A Hybridized Discontinuous Petrov-Galerkin Scheme for Compressible Flows

by

David Moro-Ludeña

Ing., Universidad Politécnica de Madrid (2007)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author .....
Department of Aeronautics and Astronautics
May 16, 2011

Certified by .....
Jaume Peraire
Professor
Thesis Supervisor

Certified by .....
Ngoc Cuong Nguyen
Research Scientist
Thesis Supervisor

Accepted by .....
Eytan H. Modiano
Associate Professor of Aeronautics and Astronautics Chair
Graduate Program Committee

A Hybridized Discontinuous Petrov-Galerkin Scheme for Compressible Flows

by

David Moro-Ludeña

Submitted to the Department of Aeronautics and Astronautics
on May 16, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The Hybridized Discontinuous Petrov-Galerkin scheme (HDPG) for compressible flows is presented. The HDPG method stems from a combination of the Hybridized Discontinuous Galerkin (HDG) method and the theory of the optimal test functions, suitably modified to enforce the conservativity at the element level. The new scheme maintains the same number of globally coupled degrees of freedom as the HDG method while increasing the stability in the presence of discontinuities or under-resolved features. The new scheme has been successfully tested in several problems involving shocks such as Burgers equation and the Navier-Stokes equations and delivers solutions with reduced oscillation at the shock. When combined with artificial viscosity, the oscillation can be completely eliminated using one order of magnitude less viscosity than that required by other Finite Element methods. Also, convergence studies in the sequence of meshes proposed by Peterson [49] show that, unlike other DG methods, the HDPG method is capable of breaking the suboptimal k+1/2 rate of convergence for the convective problem and thus achieve optimal k+1 convergence.

Thesis Supervisor: Jaume Peraire
Title: Professor

Thesis Supervisor: Ngoc Cuong Nguyen
Title: Research Scientist

Acknowledgments

Definitely, these two years at MIT have been a trip and a half; ups after downs, the whole experience has been like a roller coaster. Nobody said it was going to be easy: there have been moments of doubt and sorrow; endless working hours and that feeling that you never quite disconnect... however, there have also been times of extreme happiness, of feeling to belong to a very special community; times of success, and celebration. The former I vaguely recall, the latter I certainly do and will. And they all have in common the same thing, it is the people what makes them special.

First of all, I would like to thank the ACDL'ers for being an awesome gang. Some labs might have windows, some even views to the Muddy, but none have (or have had) you guys; thanks to Andrew, Hemant, Joel, Alejandra, Albert, Marcelo, Julie, David, Masa, Laslo, Leo, Laura, Erics, Nikhil, Matt, Huafei, Xun, JM, Tom, Emily, Chad, Mody,.... The seniors in the lab volunteered their time to prepare those of us taking Quals. I would like to thank Andrew, Hemant, Xun, Laslo and Masa for their valuable help. Without you the outcome would have probably been different.

All this outstanding minds carry out incredible research, that would not be possible without the people behind the logistics. My special thanks to Jean for her perpetual good mood and for making an exceptional job without taking any credit, and also to Laslo for all the time spent attending our computer petitions and problems.

I would like to show my most sincere gratitude to my advisors: Professor Jaume Peraire and Dr. Cuong Nguyen for their patience and availability any time I required them, regardless of how simple or silly the question might have been. They have been a continuous source of encouragement and definitely the best guidance one could ever have around this maze. Both have taught me a lot of things apart from numerical analysis (which they definitely master); the most important one: thou shalt laugh (or the importance of working hard and laughing even harder). Thank you very much

for everything.

Luckily enough there was also life outside the lab... I would like to thank my good friends Andrew (Professor) March, Hemant (Ay Mate) Chaurasia, Joel (Escandalo) Saa-Seoane, Marcelo (Fittipaldi) Buffoni, Xevi (Mojitos) Roca, Adam Conroy and Christie Klisz for being awesome. Thanks guys for the good times and for keeping me well hydrated on the weekends.

Moving countries has been hard, and leaving your loved ones behind makes it even harder. I would like to thank my family for their continuous support and specially my not-so-little-anymore brother Victor; I miss you every day "Canijo". Also, I would like to thank my friends for making it feel as if time had not passed every time I visit.

I would like to dedicate this thesis to Carmen for being the most important person in my life. Yours was the idea of coming to grad school together and I must admit it was a damn good one. During all this time we have become each others haven and best friend. You can read me like and open book, give me some slack if I need some time alone, make me smile when I am sad and laugh with me when I am happy. I love you.

Last but not least, I would like to thank the CajaMadrid Foundation for the Graduate Studies Scholarship that funded my work during the last two years. Your support enables plenty of interesting research to be carried out by young students and represents a firm example of faith in the next generations.

Contents

1 Introduction **15**

1.1 Finite Element Methods for Hyperbolic Conservation Laws 16

1.2 Discontinuous Galerkin 18

1.3 Shock Capturing 20

1.4 Hybridizable Discontinuous Petrov-Galerkin 22

2 Hybridizable Discontinuous Galerkin **25**

2.1 Notation 26

2.2 Linear Convection-Diffusion Problem 27

2.2.1 Local vs. Global Problem 27

2.2.2 Weak Formulation 29

2.2.3 The Case of Pure Convection 30

2.3 Non-linear Systems of Conservation Laws 31

2.3.1 Discretization 31

2.3.2 Solution Procedure 33

2.3.3 Non-linear Local Solver 36

2.4 Stabilization Parameter and Boundary Conditions 38

3 Optimal Test Functions **39**

3.1 General Weak Formulation 40

3.2 The Role of the Test Space 41

3.2.1 Optimal Test Space: Theoretical Result 42

3.2.2 Optimal Test Space: Discrete Approximation 43

3.3 DPG Scheme 44
3.4 Comments 47

4 Hybridizable Discontinuous Petrov-Galerkin 49
4.1 HDPG for Hyperbolic Systems 50
4.1.1 Local Problem 51
4.1.2 Imposing Conservativity 53
4.1.3 Local Problem Solution 54
4.1.4 Local Problem Sensitivities 58
4.1.5 Global Problem 58
4.2 HDPG for Elliptic Operators 61
4.2.1 Local Problem 61
4.2.2 Local Problem Sensitivities 65
4.2.3 Global Problem 66
4.3 HDPG Single Element Results 67
4.3.1 Burgers Equation in 1D 67
4.3.2 Euler Equations in 2D 69
4.4 Comments 70

5 Results 75
5.1 1D Results 76
5.1.1 Linear Convection 76
5.1.2 Burgers 1D: Steady Shock 78
5.1.3 Burgers 1D: Shock Propagation 80
5.2 2D Results 82
5.2.1 Linear Convection 82
5.2.2 Burgers 2D 85
5.2.3 Navier-Stokes 88

6 Conclusions and Future Work 97
6.1 Summary 97

6.2 Conclusion 98
6.3 Future Work 99

A HDG Method for Different Governing Equations **101**

A.1 Convection 101
A.2 Convection-Diffusion 102
A.3 Burgers 1D 104
A.4 Burgers 2D 106
A.5 Euler 107
A.6 Navier-Stokes 110

List of Figures

2-1 Support of the different solution spaces used in the HDG scheme; u_h, $\mathbf{q}_h$ and $\hat{u}_h$ on the element and the interfaces 29

4-1 Coupling introduced by the inverse Riesz mapping when HDPG or DPG are applied to an elliptic operator 63

4-2 Comparison of HDG and HDPG for the case of Burgers equation in 1D using a single element and boundary data compatible with a steady shock 68

4-3 Comparison of the trial and associated optimal test functions in the case of the Burgers equation on a single element with a steady shock 68

4-4 Comparison between HDG and HDPG for the case of an oblique shock using the Euler equations and a single element 72

4-5 HDPG solution for the case of a normal shock using the Euler equations and a single element 73

5-1 Comparison between HDG and HDPG for a linear convection case in 1D with discontinuous initial conditions 77

5-2 Comparison between HDG and HDPG for the case of Burgers equation with a steady shock 79

5-3 Comparison between HDG and HDPG for the unsteady Burgers equation in a case with shock propagation 81

5-4 Example of Peterson's mesh used to prove suboptimal converge for DG schemes 83

5-5 Converge plot for Peterson's example using HDG and HDPG with $k = 1$ 84

5-6 Solution to the Burgers equation in 2D using both HDG and HDPG on a structured mesh 86
5-7 Solution to the Burgers equation in 2D using both HDG and HDPG on an unstructured mesh 87
5-8 Unstructured mesh used to compute the flow over a supersonic wedge 90
5-9 Oblique shock over a wedge in a supersonic flow computed using HDPG 90
5-10 Structured mesh used to compute the solution in a transonic channel 91
5-11 Transonic flow inside a channel with a small bump on the lower surface computed using HDPG 92
5-12 Close-up of the solution for the transonic flow inside the channel overlapped with the grid to show the shock is being captured within one element 92
5-13 Comparison between HDG and HDPG for the case of the Trefftz airfoil at zero angle of attack and $M_\infty = 0.8$ 94
5-14 Detail of the Mach number oscillations that appear around the shock wave when HDG is used on the transonic flow over a Trefftz airfoil . . 95
5-15 Structured mesh used to compute the transonic flow over a Trefftz airfoil 95

List of Tables

4.1 Breakdown of the degrees of freedom required for HDG and HDPG . 65

4.2 States before and after the oblique shock single element case. 71

4.3 States before and after the normal shock single element case. 71

5.1 Computed error and convergence rate for Peterson's example using HDG and HDPG with $k = 1$ 84

5.2 Comparison of maximum relative oscillation (%) at the shock as a function of the viscosity between HDG and HDPG using a Burgers 2D problem 88

Chapter 1

Introduction

During the last twenty years, the field of Aeronautics has experienced the raise of Computational Fluid Dynamics (CFD) as a tool for many day-to-day design decisions. This development has been mostly leveraged by the continuous increase in computational power available on workstations and personal computers. However, many of the methods used nowadays in the industry are robust low order algorithms developed before the age of fast and affordable computers even started. Examples of this include panel methods (used for low to moderate speed aeroelastic and steady aerodynamic analysis) or Finite Volume Methods (used mostly for Euler or Reynolds Averaged Navier-Stokes (RANS) calculations of compressible flows).

At this point in time, the industry has realized that these tools are not enough, or do not take full advantage of the virtues of the hardware, and is trying to push the development into industrial stage of new, faster and more accurate methods suitable to their application (e.g. the European ADIGMA project [35]). What industry is looking for is high-order adaptive methods on unstructured/hybrid grids that can deal with compressible aerodynamics (typical of high speed configurations such as cruise) as well as separated flows (typical of high lift configurations such as landing); possibly combined with some turbulence modeling through Reynolds-Averaged Navier-Stokes (RANS) or Large Eddy Simulation (LES).

There exist several candidate algorithms to solve the problem but only some appear to meet all the requirements. Two of the most promising approaches are the WENO Finite Volume Method (WENO-FVM) and the Discontinuous Galerkin method (DG). A good review and comparison between them can be found in [53]. In this thesis, the later will be extended to a new method named Hybridizable Discontinuous Petrov-Galerkin method (HDPG) to deal with situations in which underresolution affects stability and prevents the convergence; more precisely, the objective is to enhance stability in the presence of discontinuities (shock waves) that appear naturally in the system of equations that governs compressible flows (Euler and Navier-Stokes equations).

1.1 Finite Element Methods for Hyperbolic Conservation Laws

The DG scheme, that represents the point of departure of the new method proposed here, can be classified as a Finite Element Method (FEM) with special approximation spaces. For a long time, and long before DG became a popular method to solve conservation laws, researchers in the field had tried to apply the framework of the Continuous Galerkin FEM (CG) to these problems with mixed success. In the most common CG framework, the approximation spaces are continuous across interfaces of the mesh and hence, degrees of freedom between neighboring elements are connected. Not only that, but it is common to assume that trial and test spaces, that represent the approximation space for the solution and the weighting space respectively, are the same, making the method easy to implement but lacking stability for the convective operator. Indeed, it is well known, that finite element methods are equivalent to centered differences when the approximation space is composed of piecewise linear functions. Despite this, when looking at CG in the context under which it was developed; coercive, elliptic and symmetric operators, this choice for the spaces makes all

the sense and is the reason why CG is unbeatable for certain problems such as structural analysis. This popularity partially leveraged the development of CG methods for hyperbolic problems.

All the challenges that CG methods find when dealing with compressible flows can be traced back to the nature of the Partial Differential Equations (PDEs) that describe the phenomena. Namely:

1. Conservation laws (in differential form) are derived from integral principles using the divergence theorem and certain assumptions in the conserved fluxes such as differentiability. Hence, in the presence of discontinuities, they lack all validity from a mathematical point of view even though the integral principle still applies.

2. The PDEs that govern compressible flows usually present hyperbolic character in most of the physical domain, that is, there exists transport of information along privileged directions in the space-time domain.

with this is mind one can argue that the struggles that CG schemes face when discontinuities appear are due to the fact that the discretization is not locally conservative (since conservation cannot be guaranteed element-wise) and the hyperbolic character is not preserved (since the domain of dependence of the numerical solution includes regions that are not physically meaningful).

While the first of these issues, strongly related to conservation across shock waves, cannot be addressed unless the approximation spaces are modified (this is precisely what DG does), the second one has been subject of extensive research in the CG community. It is a well known fact that, CG methods applied to linear convection-diffusion operators, present oscillations when the Peclet number $Pe = h|a|/\nu$ (that measures convective vs. diffusive effects) is greater than $\mathcal{O}(1)$. Very good examples and analysis on this can be found in [11], together with the motivation behind the idea of upwinding the weighting functions to introduce the directionality inherent

to the problem. Amongst others, the most successful CG methods in this context are the so-called stabilized Finite Element methods such as the Streamline-Upwinded Petrov-Galerkin (SUPG) method [11] and the Galerkin/Least-Squares (GLS) method [29]; these two schemes have been thoroughly applied to fluid dynamics problems with reasonable success, however, they are rarely high order. Other methods, such as the Variational Multi-Scale method [28] or stabilized bubble methods [9] rely on consistent artifacts to capture the small scales of the problem (not resolved by the mesh and approximation space) since these are the ones blamed for causing the oscillation. A good unified approach to all these methods can be found in [27].

1.2 Discontinuous Galerkin

The DG method is a FEM first introduced by Reed and Hill in 1973 [50] to solve convection-reaction laws. Unlike CG, DG was directly devised in the context of hyperbolic problems. It took some time for the advantages of the method to be realized by the numerical analysis community; as the first sharp error estimates came a decade later (see [31, 49]) and the extension to non-linear systems had to wait yet another decade until the appearance of the Runge-Kutta DG scheme (RKDG) [19]. At that point, attention was drawn to the extension to elliptic operators and its associated issues; under this hood several schemes were devised: Bassi-Rebay (BR2) [4], Local DG (LDG) [18], Compact DG (CDG) [47] and others. These were all discussed under a unified framework in [2]. The method was further developed into the Hybridizable Discontinuous Galerkin scheme (HDG) (see [16] for the inception and [40, 41, 42, 43, 46] for extension and applications to different systems) which involves less degrees of freedom than the original DG scheme amongst other advantages.

As its name indicates, DG involves spaces that are discontinuous. These discontinuities are aligned with the edges of the triangulation. When dealing with conservation laws, these discontinuities generate new terms in the weak formulation that account for the integral of the fluxes along the edges of the domain. Since the solu-

tion is discontinuous along edges, the flux across them has to be approximated using available information about the solution inside the elements. This can be done in several ways, leading to different DG schemes. In principle, the only requirement is that the approximation is consistent (reproduces the original flux when the exact solution is introduced) and is what makes DG a very powerful scheme:

1. In order to account for directionality, the approximated fluxes on the boundaries (also referred to as numerical traces) can be chosen so that the information is taken consistently with the characteristic lines; this is nothing but an upwinding-like effect. Furthermore, DG methods allow the numerical traces to be computed using Riemann Solvers inspired in the FVM (see [54] for a deep review), so that the solution is entropy-satisfying and presents other desirable features.

2. In order to deal with elliptic operators (like the viscous terms in the Navier-Stokes equations), the problem is written as a system of first order PDEs and the numerical traces can be chosen such that the degrees of freedom that represent the gradients of the solution are eliminated element-wise in favor of the solution itself, without affecting the well-possedness of the system (see LDG [18] and CDG [47]). This implies that the discretization of elliptic operators does not penalize the overall size of the system even though the number of equations is increased.

Apart from the usual properties of FEM such as the ability to deal with complex geometries (using unstructured grids) or the simple implementation of h-adaptivity, the DG method presents some extra advantages. First, the discontinuous nature of the spaces implies local conservativity (provided the constant mode belongs to the test space) which is a highly desirable property when shocks are present as it guarantees the proper shock propagation. It is also due to the special choice of the approximation spaces that high-order can be achieved easily and hp-adaptivity can be carried out almost trivially even when hanging nodes are present. Finally, the implementation of boundary conditions in DG is usually simple and straightforward compared to other methods such as FVM or Finite Differences.

As expected, DG methods also present some drawbacks. First of all, the number of degrees of freedom on the interfaces is doubled, hence, the memory requirement and operation count are increased with respect to CG; a workaround for this is the HDG method that will be discussed in the next chapter.

Second, the approximation spaces (specially the test space) are usually chosen to be standard polynomials, hence ignoring the hyperbolic nature of the problem at the element level and the presence of characteristic directions. As mentioned above, some CG methods such as SUPG deal with this by upwinding the test space. The approach discussed in this thesis will have this flavor but will be derived from other principles, namely the theory of the Optimal Test Functions (Chapter 3).

1.3 Shock Capturing

As the title of this thesis indicates, the objective is to develop an algorithm for compressible flows (or convection-dominated conservation laws in general). One of the main characteristics of these flows is the presence of discontinuities, that represent lower-dimensional regions in the domain (more precisely points in 1D, lines in 2D or surfaces in 3D) where information from different characteristic lines intersects hence generating non-uniqueness in the solution. It is well know that shock waves act as sinks of information on the $(\mathbf{x}, t)$ space, propagating at a speed given by the Rankine-Hugoniot conditions [36]. This propagation speed relies only on a conservativity argument and hence the interest in a conservative discretization. More discussion about the virtues of conservative schemes can be found in many FVM textbooks such as [37].

While local conservativity is an issue that DG deals with gracefully, the stability of the scheme around discontinuities, that can be measured in terms of the oscillation (or the total variation) of the solution, is not adequate in the sense that non-physical oscillations might appear and eventually prevent convergence. This makes standard

DG methods not as competitive as their low/moderate order FVM counterparts and is certainly a concern within the DG community.

This lack of stability can be partly explained by the high order approximation spaces used in DG. It seems like the mere idea of approximating a discontinuous solution using high order polynomials seems daunting as it contradicts the well studied Gibbs phenomenon. In order to get smooth solutions across shocks, some sort of numerical artifact is mandatory. The most popular choices are:

- Artificial Viscosity: relies on introducing enough dissipation so that $Pe = h|a|/(k\nu) = \mathcal{O}(1)$. In this situation, the solution gets regularized down to a scale (that depends on the mesh size h and approximation order k) where the elliptic operator dominates and the solution can be resolved [51]. In order to identify the elements where shocks are present several strategies can be followed; of very wide application are the polynomial coefficient decay [48] or the inter-elemental jump monitoring [34]. When using this approach the validity of the solution is at stake since the artificial viscosity introduced might well be way over the physical one, modifying the solution in an unpredictable manner. In order to be consistent, some h-adaptivity has to be used to properly resolve the flow to a sufficiently small length scale.

- Limiters: based on non-linear limiting techniques (related to Godunov's theorem [37]) inspired in the Total Variation Diminishing principle combined with explicit time integration. See for example the RKDG method [19].

However, not only the trial space is to be blamed for the oscillation. As previously mentioned, characteristic lines intersect at the shock, hence, before and after it the upwind direction might change. If this is not accounted for in the test space, stability would be compromised. In the case of FVM, since the solution is defined as the average in each cell, the volumetric terms disappear and all the upwinding-like strategies are applied on the fluxes across elements. This is not the case in standard DG since the solution inside the element is a high order polynomial instead of a constant.

In most implicit DG schemes, the stabilization of oscillations across shocks is achieved through a combination of both the diffusion associated to the jumps in the solution across interfaces and some extra artificial viscosity applied to the problem using a non-linear discontinuity sensor. In this thesis, the goal is to rely less on dissipation and more on the suitable choice of the test functions to achieve the desired upwinding inside the element.

1.4 Hybridizable Discontinuous Petrov-Galerkin

The HDPG method, relies on the computation of the test functions on the fly in order to achieve stability in a natural way. The stability of a trial-test space combination, for a general weak formulation, can be measured in terms of the so called inf-sup constant [3, 10]; if this constant is bounded away from zero as the element size decreases, the scheme is deemed stable. A practical example of this is the different interpolation spaces for velocity and pressure used in the Stokes system (see [10]). The idea, in simple words, is to set up a dual problem to find the optimal test functions so as to maximize the inf-sup. As will be discussed later in Chapter 3, this associated problem is related with the adjoint operator (hence the upwinding) and yields a symmetric positive definite system to solve for. This new scheme was recently introduced in [21, 22, 24] and, as described there, yields a system with more unknowns than the original DG and more globally coupled degrees of freedom, hence, hard to implement and solve.

The approach taken here is to combine the optimal test space with the HDG framework to stabilize the problem inside the elements while letting the numerical fluxes take care of the transfer of information across interfaces. It turns out that the computation of the test space on the fly might yield a method that is non-conservative since the constant mode is not guaranteed to belong to it; this is taken care of in HDPG by using an extra constraint in the associated optimal test space computation.

The structure of this thesis is the following. In Chapter 2 the HDG scheme, that will serve as the skeleton for the new method, will be presented for the case of a general hyperbolic-elliptic operator. Following, in Chapter 3 the theory of the optimal test functions will be discussed together with the Discontinuous Petrov-Galerkin scheme (DPG) which is the first method in which they were applied. In Chapter 4, the Hybridizable Discontinuous Petrov-Galerkin scheme (HDPG) will be described as an application of the optimal test functions to HDG with some modifications in order to ensure local conservation. Then, in Chapter 5 some 1D and 2D results will be shown to asses the HDPG method. Finally, Chapter 6 will go over some conclusions and future work.

Chapter 2

Hybridizable Discontinuous Galerkin

Despite the significant advantages of Discontinuous Galerkin (DG) methods when compared to other methods such as Finite Differences, Finite Volumes or Spectral methods, there is one important disadvantage that can outweigh them all: the high computational cost associated to DG. Such high computational cost has to do with the duplication of degrees of freedom across edges that the discontinuous spaces introduce.

The Hybridizable Discontinuous Galerkin scheme (HDG) is a variation of the DG scheme designed to reduce the overall number of degrees of freedom of the problem. This method was initially developed by Cockburn et al. [16] for elliptic operators and later extended to other problems by Nguyen et al. [40, 41, 42, 43, 46]. In this chapter, the HDG method will be first applied to a single component linear convection-diffusion problem and later extended to general time dependent non-linear systems of hyperbolic/elliptic conservation laws.

2.1 Notation

As usual in the Finite Element context, approximation (or **trial**) spaces and weighting (or **test**) spaces have to be introduced in order to derive the weak statement. Let $\mathcal{T}_h$ represent a partition of Ω composed of disjoint regular elements and let $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$ represent the set of element faces. Let $\mathcal{E}_h^i$ represent the set of internal faces counted only once (notice internal faces are counted twice in $\partial\mathcal{T}_h$) and let $\mathcal{E}_h^\partial$ denote the set of boundary faces. Let $\mathcal{E}_h$ represent the union of the internal and boundary faces: $\mathcal{E}_h := \mathcal{E}_h^i \cup \mathcal{E}_h^\partial$.

The discontinuous spaces based on this triangulation are defined as:

$$\mathbf{V}_h^k = \{\mathbf{v} \in (L^2(\Omega))^m : \mathbf{v}|_K \in (\mathcal{P}^k(K))^m \quad \forall K \in \mathcal{T}_h\} \tag{2.1}$$

$$\mathbf{W}_h^k = \{\mathbf{v} \in (L^2(\Omega))^{m\times d} : \mathbf{v}|_K \in (\mathcal{P}^k(K))^{m\times d} \quad \forall K \in \mathcal{T}_h\} \tag{2.2}$$

$$\mathbf{M}_h^k = \{\mathbf{v} \in (L^2(\mathcal{E}_h))^m : \mathbf{v}|_e \in (\mathcal{P}^k(e))^m \quad \forall e \in \mathcal{E}_h\} \tag{2.3}$$

where $\mathcal{P}^k(D)$ represents the space of polynomials of degree k in the domain D, d represents the number of space dimensions of the problem and m represents the number of components of the system. The subscript h follows the usual convention that indicates the space is finite dimensional and associated to a certain triangulation of characteristic element size h.

In order to derive a weak formulation, the following inner products are introduced:

$$(v, w)_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (v, w)_K \tag{2.4}$$

$$(\mathbf{v}, \mathbf{w})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \sum_{i=1}^{d} (v_i, w_i)_K \tag{2.5}$$

$$(\mathbf{V}, \mathbf{W})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \int_K \mathrm{tr}(\mathbf{V}^T\mathbf{W}) \tag{2.6}$$

$$\langle v, w\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle v, w\rangle_{\partial K} \tag{2.7}$$

$$\langle \mathbf{v}, \mathbf{w} \rangle_{\partial \mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \langle \mathbf{v}, \mathbf{w} \rangle_{\partial K} \tag{2.8}$$

where

$$(v, w)_K = \int_K vw, \qquad \langle v, w \rangle_{\partial K} = \int_{\partial K} vw \tag{2.9}$$

2.2 Linear Convection-Diffusion Problem

The point of departure of the formulation is the linear convection-diffusion equation:

$$\begin{aligned} \nabla \cdot (\mathbf{c}u) - \nabla \cdot (\kappa \nabla u) &= f \quad \text{in} \quad \Omega & (2.10) \\ b(u, \nabla u) &= g \quad \text{on} \quad \partial\Omega & (2.11) \end{aligned}$$

that represents the distribution of a single scalar u inside domain Ω under the action of an advection field given by $\mathbf{c}$ and a homogeneous diffusion proportional to κ subject to boundary conditions on u (Dirichlet), its gradient ∇u (Neumann) or the flux $\mathbf{f} = \mathbf{c}u - \kappa \nabla u$ (Robin), encoded in the operator $b(u, \nabla u)$. Despite its simplicity, this equation is important because it represents the linearized version of any non-linear hyperbolic-elliptic operator and is thus a building block for any Newton-based iterative solver. The 2^{nd} order equation can be written as a 1^{st} order system by introducing an extra equation for the kinematic variables $\mathbf{q}$:

$$\begin{aligned} \nabla \cdot (\mathbf{c}u) - \nabla \cdot (\kappa \mathbf{q}) &= f \quad \text{in} \quad \Omega & (2.12) \\ \nabla u - \mathbf{q} &= 0 \quad \text{in} \quad \Omega & (2.13) \\ b(u, \mathbf{q}) &= g \quad \text{on} \quad \partial\Omega & (2.14) \end{aligned}$$

2.2.1 Local vs. Global Problem

The HDG method stems naturally from a very basic observation. Let K be a given element of the triangulation, and let λ be a function with support on ∂K. The

problem:

$$\nabla \cdot (\mathbf{c}u^\lambda) - \nabla \cdot (\kappa \mathbf{q}^\lambda) = f \qquad \text{in} \quad K \tag{2.15}$$

$$\nabla u^\lambda - \mathbf{q}^\lambda = 0 \qquad \text{in} \quad K \tag{2.16}$$

$$u^\lambda = \lambda \qquad \text{on} \quad \partial K \tag{2.17}$$

is well posed $\forall \kappa$. Furthermore,

$$\lambda = u|_{\partial K} \Rightarrow \quad \mathbf{q}^\lambda = \mathbf{q}, \quad u^\lambda = u \tag{2.18}$$

The idea behind HDG is to introduce the new unknown λ so that the solution inside each element K is parametrized by λ. This yields a Dirichlet-to-Neumann mapping $T : \lambda \mapsto \mathbf{F}^\lambda$ where $\mathbf{F}^\lambda = (\mathbf{c}u^\lambda - \kappa \mathbf{q}^\lambda)|_{\partial K}$ represents the flux of u at the boundaries of element K. The name local problem comes from the locality of this mapping that only depends on the value of λ at the boundaries of element K.

The choice of λ, is then dictated by the original conservative character of the problem. Namely, λ has to be such that the flux is conserved across interfaces between elements:

$$(\mathbf{F}^\lambda)^+ \cdot \mathbf{n}^+ + (\mathbf{F}^\lambda)^- \cdot \mathbf{n}^- = 0 \qquad \text{on} \quad I, \quad \forall I \in \mathcal{E}_h^i \tag{2.19}$$

$$b(\lambda, \mathbf{q}^\lambda) = g \qquad \text{on} \quad \partial\Omega \tag{2.20}$$

where $\mathbf{n}^+$ and $\mathbf{n}^-$ represent the outward-pointing normal of the element to the right and left of face I respectively. Notice that $\mathbf{n}^+ = -\mathbf{n}^-$ by definition. Notice also that the boundary conditions are now imposed on the global problem. The conservation of the fluxes represented by the global problem is responsible of coupling the solution from element to element.

2.2.2 Weak Formulation

Once the problem (Equations 2.15-2.20) has been described, the usual Finite Element procedure can be applied to derive a weak formulation for the system. For that, the system is weighted against suitable test spaces, integrated by parts and summed over all the elements (see [15, 40, 41] for details on the derivation). The weak statement then reads; find $(u_h, \mathbf{q}_h, \hat{u}_h) \in \mathbf{V}_h^k \times \mathbf{W}_h^k \times \mathbf{M}_h^k$ such that:

$$-(\mathbf{c}u_h + \kappa\mathbf{q_h}, \nabla v)_{\mathcal{T}_h} + \langle(\widehat{\mathbf{c}u_h} + \widehat{\kappa\mathbf{q}_h}) \cdot \mathbf{n}, v\rangle_{\partial\mathcal{T}_h} = (f, v)_{\mathcal{T}_h} \qquad \forall v \in \mathbf{V}_h^k \qquad (2.21)$$

$$-(u_h, \nabla \cdot \mathbf{w})_{\mathcal{T}_h} + \langle\hat{u}_h, \mathbf{w} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h} + (\mathbf{q}_h, \mathbf{w})_{\mathcal{T}_h} = 0 \qquad \forall \mathbf{w} \in \mathbf{W}_h^k \qquad (2.22)$$

$$\langle(\widehat{\mathbf{c}u_h} + \widehat{\kappa\mathbf{q}_h}) \cdot \mathbf{n}, \mu\rangle_{\partial\mathcal{T}_h \setminus \partial\Omega} + \langle b(\hat{u}_h, \mathbf{q}_h) - g, \mu\rangle_{\partial\Omega} = 0 \qquad \forall \mu \in \mathbf{M}_h^k \qquad (2.23)$$

where $\widehat{\mathbf{c}u}$ and $\widehat{\kappa\mathbf{q}}$ represent the approximation to the fluxes on the faces of $\mathcal{T}_h$ (also referred to as numerical fluxes) that appear after the integration by parts due to the discontinuous nature of the approximation spaces. Notice that in Equation 2.22, the approximation of u_h on the faces is taken to be $\hat{u}_h$ ($\hat{u}_h$ is nothing but a discrete version of λ described in §2.2.1). Figure 2-1 shows a picture of the different spaces just described.

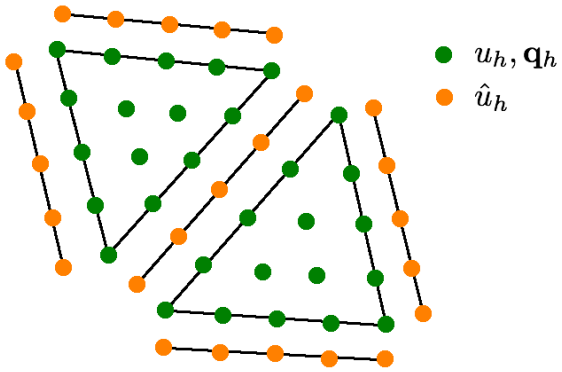


Figure 2-1: Support of the different solution spaces used in the HDG scheme; u_h (green), $\mathbf{q}_h$ (green) and $\hat{u}_h$ (orange). Notice how the traces ($\hat{u}_h$) are only defined on the faces of the triangulation and are non-unique at the vertices while the internal degrees of freedom (u_h and $\mathbf{q}_h$) are defined inside each element and duplicated at the edges.

The challenge now is to define $\widehat{\mathbf{c}u}$ and $\widehat{\kappa\mathbf{q}}$ in such a way as to render the system solvable and consistent. Following the definition of the local problem, the fluxes at the interfaces are chosen as:

$$\widehat{\mathbf{c}u_h} + \widehat{\kappa\mathbf{q}_h} = \mathbf{c}\hat{u}_h + \kappa\mathbf{q}_h + \tau(u_h - \hat{u}_h)\cdot\mathbf{n} \tag{2.24}$$

here, τ represents a stabilization parameter that has to be chosen appropriately and will be described in §2.3.2. A more detailed description of other choices for the numerical fluxes can be found in [41].

2.2.3 The Case of Pure Convection

The HDG scheme can similarly be applied to convective operators. In that case, the kinematic variables $\mathbf{q}$ are no longer needed and the local problem reads:

$$\nabla\cdot(\mathbf{c}u^\lambda) = f \quad \text{in} \quad K \tag{2.25}$$

$$u^\lambda = \lambda \quad \text{on} \quad \partial K \tag{2.26}$$

This problem is well posed provided u is defined on the inflow boundary ($\mathbf{c}\cdot\mathbf{n} < 0$). As written, the system might appear over-specified (since u is defined on the whole ∂K), however, this is not an issue provided the fluxes at the boundaries discern between incoming (inflow $\mathbf{c}\cdot\mathbf{n} < 0$) and outcoming (outflow $\mathbf{c}\cdot\mathbf{n} > 0$) information. As a consequence, once λ is set to match the solution and the conservation of fluxes across faces is imposed using the Dirichlet-to-Neumann map, the system will be well posed. As for the case of the convection-diffusion equation, the weak formulation can be derived by integration by parts and summation over all the elements. The system to solve then reads: find $(u_h, \hat{u}_h) \in \mathbf{V}_h^k \times \mathbf{M}_h^k$ such that

$$-(\mathbf{c}u_h, \nabla v)_{\mathcal{T}_h} + \langle(\widehat{\mathbf{c}u_h})\cdot\mathbf{n}, v\rangle_{\partial\mathcal{T}_h} = (f, v)_{\mathcal{T}_h} \qquad \forall v \in \mathbf{V}_h^k \tag{2.27}$$

$$\langle(\widehat{\mathbf{c}u_h})\cdot\mathbf{n}, \mu\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \langle b(\hat{u}_h) - g, \mu\rangle_{\partial\Omega} = 0 \qquad \forall \mu \in \mathbf{M}_h^k \tag{2.28}$$

where now the numerical flux is defined as:

$$\widehat{\mathbf{c}u_h} = \mathbf{c}\hat{u}_h + \tau(u_h - \hat{u}_h)\cdot\mathbf{n} \tag{2.29}$$

notice that the choice $\tau = |\mathbf{c}\cdot\mathbf{n}|$ yields the desired upwinding effect. The choice of τ is discussed in more detail in Appendix A.

2.3 Non-linear Systems of Conservation Laws

The HDG method presented above can also be applied to unsteady systems of PDEs written in conservative form:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{F}(\mathbf{u}) + \mathbf{G}(\mathbf{u},\nabla\mathbf{u})) = \mathbf{f} \qquad \text{in} \quad \Omega\times(0,T] \tag{2.30}$$

$$\mathbf{u} = \mathbf{u}_0 \qquad \text{in} \quad \Omega\times\{t=0\} \tag{2.31}$$

$$\mathbf{b}(\mathbf{u},\nabla\mathbf{u}) = \mathbf{g} \qquad \text{on} \quad \partial\Omega\times(0,T] \tag{2.32}$$

where $\mathbf{u}$ represents the vector of unknowns (conserved quantities), $\mathbf{F}$ represents the inviscid (hyperbolic) fluxes of each conserved quantity and $\mathbf{G}$ represents the viscous (elliptic) fluxes, that depend on $\mathbf{u}$ as well as its gradient. The initial conditions are set by $\mathbf{u}_0$ and the boundary conditions are imposed through the operator $\mathbf{b}$. Several problems of interest can be written in this form, in particular, the Euler and Navier-Stokes equations that describe compressible flows. It is worth noticing that the linear problems described above can also be cast into this form, hence, Equations 2.30-2.32 will be the reference problem from now onwards in this manuscript.

2.3.1 Discretization

In order to derive the HDG scheme for Equations 2.30-2.32, first, the system has to be written as a first order system by introducing the kinematic variables $\mathbf{Q}$. Then, the

time derivative has to be discretized in a Method of Lines fashion. For that, the time dependent solution is assumed to belong to a discrete space $(\mathbf{u}_h(t), \mathbf{Q}_h(t)) \in \mathbf{V}_h^k \times \mathbf{W}_h^k$ and a similar procedure to the one described earlier for the single equation is carried out in order to derive the weak formulation, namely, introduce the space for the traces on the faces ($\hat{\mathbf{u}}_h(t) \in \mathbf{M}_h^k$), integrate by parts and sum over the elements. The weak formulation then reads; find $(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h) \in \mathbf{V}_h^k \times \mathbf{W}_h^k \times \mathbf{M}_h^k$ such that:

$$\left(\frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v}\right)_{\mathcal{T}_h} - (\mathbf{F} + \mathbf{G}, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} - \langle (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial \mathcal{T}_h} = (\mathbf{f}, \mathbf{v})_{\mathcal{T}_h} \tag{2.33}$$

$$-(\mathbf{u}_h, \nabla \cdot \mathbf{E})_{\mathcal{T}_h} - (\mathbf{Q}_h, \mathbf{E})_{\mathcal{T}_h} + \langle \hat{\mathbf{u}}_h, \mathbf{E} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} = 0 \tag{2.34}$$

$$\langle (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h \setminus \partial \Omega} + \langle \mathbf{b}(\hat{\mathbf{u}}_h, \mathbf{Q}_h) - \mathbf{g}, \mu \rangle_{\partial \Omega} = 0 \tag{2.35}$$

$\forall (\mathbf{v}, \mathbf{E}, \mu) \in \mathbf{V}_h^k \times \mathbf{W}_h^k \times \mathbf{M}_h^k$. Where $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{G}}$ represent the numerical fluxes and follow the usual choice in HDG:

$$\widehat{\mathbf{F}} + \widehat{\mathbf{G}} = \mathbf{F}(\hat{\mathbf{u}}_h) + \mathbf{G}(\hat{\mathbf{u}}_h, \mathbf{Q}_h) + \mathbf{S}(\mathbf{u}_h - \hat{\mathbf{u}}_h) \tag{2.36}$$

the main difference being that the stabilization parameter $\mathbf{S}$ is now a matrix of dimensions $m \times m$, defined as a function of $\mathbf{u}_h$ and $\hat{\mathbf{u}}_h$.

The system of Equations 2.33-2.35 is of differential-algebraic nature since only Equation 2.33 presents time derivatives. The treatment of these time derivatives can be done in several ways. In this work, only implicit solution techniques will be considered since they naturally suit the differential-algebraic character of the system. Also, implicit techniques have several advantages from the point of view of time-step restriction due to the CFL condition and absolute stability that makes them very attractive. In particular, the case of a backwards in time single step discretization (BDF1) is presented; the only change to the system consists on the discretization of the time derivative:

$$\frac{\partial \mathbf{u}_h}{\partial t} \approx \frac{\mathbf{u}_h - \mathbf{u}_h^-}{\Delta t} \tag{2.37}$$

where $\mathbf{u}_h$ is the solution currently being seeked and $\mathbf{u}_h^-$ represents the solution at the previous time step $t - \Delta t$. When introduced in Equation 2.33, the final system to be solved at each iteration reads; find $(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h) \in \mathbf{V}_h^k \times \mathbf{W}_h^k \times \mathbf{M}_h^k$ such that:

$$\left(\frac{\mathbf{u}_h}{\Delta t}, \mathbf{v}\right)_{\mathcal{T}_h} - (\mathbf{F} + \mathbf{G}, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} - \langle(\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mathbf{v}\rangle_{\partial\mathcal{T}_h} = (\mathbf{f}, \mathbf{v})_{\mathcal{T}_h} + \left(\frac{\mathbf{u}_h^-}{\Delta t}, \mathbf{v}\right)_{\mathcal{T}_h} \tag{2.38}$$

$$-(\mathbf{u}_h, \nabla \cdot \mathbf{E})_{\mathcal{T}_h} - (\mathbf{Q}_h, \mathbf{E})_{\mathcal{T}_h} + \langle\hat{\mathbf{u}}_h, \mathbf{E} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h} = 0 \tag{2.39}$$

$$\langle(\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mu\rangle_{\partial\mathcal{T}_h \setminus \partial\Omega} + \langle\mathbf{b}(\hat{\mathbf{u}}_h, \mathbf{Q}_h) - \mathbf{g}, \mu\rangle_{\partial\Omega} = 0 \tag{2.40}$$

$\forall(\mathbf{v}, \mathbf{E}, \mu) \in \mathbf{V}_h^k \times \mathbf{W}_h^k \times \mathbf{M}_h^k$. If, instead of a BDF1 scheme, a higher order implicit multistep method [12] had been used, the system would have been modified in the same way by moving the terms associated to the value of the solution at previous steps to the right hand side. Furthermore, if the time integration had been carried out using a diagonally implicit Runge-Kutta scheme (DIRK)[1], each sub-iteration of the time stepping would have solved a system very similar to the one just described. It is for this reason that only the BDF1 discretization is described here.

For the case in which the conservation law does not include viscous (or elliptic) fluxes $\mathbf{G}$, Equation 2.34 can be omitted as well as the kinematic variables $\mathbf{Q}$. The space and time discretization follows the same principles and the final system to be solved will be smaller and only involve the variables $\mathbf{u}_h$ and $\hat{\mathbf{u}}_h$.

2.3.2 Solution Procedure

After the problem has been discretized in space and time, a non-linear algebraic system of equations has to be solved. To do so, a Newton iterative method is applied to the system; this relies on an initial guess for the solution and a linearization of the system so that an update of the guess can be computed. For a general problem: $\mathbf{f}(\mathbf{x}) = 0$, with initial guess $\mathbf{x}^i = \mathbf{x}^0$, the Newton iterate is:

$$\mathbf{f}(\mathbf{x}^i) + \frac{\partial \mathbf{f}(\mathbf{x}^i)}{\partial \mathbf{x}} \delta\mathbf{x} = 0 \rightarrow \mathbf{x}^{i+1} = \mathbf{x}^i - \left(\frac{\partial \mathbf{f}(\mathbf{x}^i)}{\partial \mathbf{x}}\right)^{-1} \mathbf{f}(\mathbf{x}^i) \tag{2.41}$$

the convergence of the method depends on the characteristics of the function $\mathbf{f}$ as well as the initial guess. Also, the method is sensitive to the step size and in practice some sort of line-search is required in order to achieve convergence. Despite all this, Newton's method success is largely due to the quadratic convergence that it exhibits. For detailed descriptions of the method, implementation techniques and convergence proofs see [32, 44, 45].

To solve Equations 2.33-2.35 at each time step, first an equivalent residual from has to be derived; $\mathbf{r}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h; \mathbf{u}_h^-) = 0$. For that, each equation is written in residual form and the basis for the test space ϕ_j is used to generate a residual vector:

$$\begin{aligned}\mathbf{r}_{\mathbf{u}}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h, \mathbf{v}; \mathbf{u}_h^-) = &\left(\frac{\mathbf{u}_h}{\Delta t}, \mathbf{v}\right)_{\mathcal{T}_h} - (\mathbf{F} + \mathbf{G}, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} - \\ &- \langle (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial \mathcal{T}_h} - (\mathbf{f}, \mathbf{v})_{\mathcal{T}_h} - \left(\frac{\mathbf{u}_h^-}{\Delta t}, \mathbf{v}\right)_{\mathcal{T}_h} \end{aligned} \tag{2.42}$$

$$\mathbf{r}_{\mathbf{Q}}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h, \mathbf{E}) = -(\mathbf{u}_h, \nabla \cdot \mathbf{E})_{\mathcal{T}_h} - (\mathbf{Q}_h, \mathbf{E})_{\mathcal{T}_h} + \langle \hat{\mathbf{u}}_h, \mathbf{E} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} \tag{2.43}$$

$$\mathbf{r}_{\hat{\mathbf{u}}}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h, \mu) = \langle (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h \setminus \partial \Omega} + \langle \mathbf{b}(\hat{\mathbf{u}}_h, \mathbf{Q}_h) - \mathbf{g}, \mu \rangle_{\partial \Omega} \tag{2.44}$$

$$\mathbf{r}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h; \mathbf{u}_h^-) = \left\{ \begin{array}{c} \vdots \\ \mathbf{r}_{\mathbf{u}}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h, \phi_j; \mathbf{u}_h^-) \\ \vdots \\ \mathbf{r}_{\mathbf{Q}}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h, \psi_j) \\ \vdots \\ \mathbf{r}_{\hat{\mathbf{u}}}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h, \chi_j) \\ \vdots \end{array} \right\} \tag{2.45}$$

where j simply represents a dummy index to denote the test against all the basis functions ϕ_j, ψ_j or χ_j of the spaces $\mathbf{V}_h^k$, $\mathbf{W}_h^k$ or $\mathbf{M}_h^k$ respectively. The linearization then follows by taking the derivatives with respect to $\mathbf{u}_h$, $\mathbf{Q}_h$ (if applicable) and $\hat{\mathbf{u}}_h$ of the various functions present in the different terms ($\mathbf{F}$, $\mathbf{G}$, $\widehat{\mathbf{F}}$, $\widehat{\mathbf{G}}$, $\mathbf{b}$, etc.); since these functions are known in analytical form, the derivatives can be computed using

the chain rule together with the expansion of the solution in terms of the basis, e.g.:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{u}_{h,kj}} = \frac{\partial \mathbf{F}}{\partial \mathbf{u}_k}\frac{\partial \mathbf{u}_k}{\partial \mathbf{u}_{h,kj}} = \frac{\partial \mathbf{F}}{\partial \mathbf{u}_k}\phi_j \tag{2.46}$$

where $k = 1, ..., m$ (number of conserved magnitudes), $j = 1, ..., N$ (number of degrees of freedom of the solution) and ϕ_j represents the j-th basis function.

The resulting linear system to solve at each iteration i can be written as:

$$\begin{bmatrix} [\mathbf{A}^i]_K & \mathbf{B}^i \\ \mathbf{C}^i & \mathbf{D}^i \end{bmatrix} \begin{pmatrix} \{\delta \mathbf{s}_h^i\}_K \\ \delta \hat{\mathbf{u}}_h^i \end{pmatrix} = - \begin{pmatrix} \{\mathbf{r}_{\mathbf{s}}^i\}_K \\ \mathbf{r}_{\hat{\mathbf{u}}}^i \end{pmatrix} \tag{2.47}$$

$$\left\{\delta \mathbf{s}_h^i\right\}_K = \begin{Bmatrix} \delta \mathbf{u}_h^i \\ \delta \mathbf{Q}_h^i \end{Bmatrix}_K \qquad \left\{\mathbf{r}_{\mathbf{s}}^i\right\}_K = \begin{Bmatrix} \mathbf{r}_{\mathbf{u}}^i \\ \mathbf{r}_{\mathbf{Q}}^i \end{Bmatrix}_K \tag{2.48}$$

where the vector subscripted as $\{\cdot\}_K$ denotes the element local unknowns and residuals. Similarly, the matrix $[\cdot]_K$ denotes the linearization of the local problem with respect to the local unknowns. Using this ordering, that is inspired by the mathematical structure of the local problem, yields a matrix $\mathbf{A}$ that is block diagonal. This allows for $\delta \mathbf{u}_h$ and $\delta \mathbf{Q}_h$ to be solved as a function of $\delta \hat{\mathbf{u}}_h$ element-wise and later inserted into the the global problem to solve a system for $\delta \hat{\mathbf{u}}_h$ only:

$$\mathbf{K}^i \delta \hat{\mathbf{u}}_h = -\mathbf{r}_{\hat{\mathbf{u}}}^* \tag{2.49}$$

where,

$$\mathbf{K}^i = \mathbf{D}^i - \mathbf{C}^i (\mathbf{A}^i)^{-1} \mathbf{B}^i \tag{2.50}$$

$$\mathbf{r}_{\hat{\mathbf{u}}}^* = \mathbf{r}_{\hat{\mathbf{u}}}^i - \mathbf{C}^i (\mathbf{A}^i)^{-1} \mathbf{r}_{\mathbf{s}}^i \tag{2.51}$$

Since $\delta \hat{\mathbf{u}}_h$ is single valued on the element faces, the resulting matrix $\mathbf{K}^i$ is smaller than that associated to other DG methods. Furthermore, the matrix is compact in the sense that, for a given face, the only coupling occurs with the degrees of freedom on the faces of the two elements that share that face, yielding a block structured

matrix $\mathbf{K}^i$ with a fixed number of blocks per row (3 in 1D, 5 in 2D and 7 in 3D when simplices are used). These two properties can be exploited during iterative solution processes. In practice, $\mathbf{K}^i$ is computed using the usual assembly procedure in FEM (see [56]) for which $\mathbf{A}^i$, $\mathbf{B}^i$, $\mathbf{C}^i$ and $\mathbf{D}^i$ are never formed explicitly.

2.3.3 Non-linear Local Solver

The implementation described in §2.3.2 relies on a linearization of both the local and the global problem at once. The results presented in [41, 46], that follow this approach, show that this is a consistent linearization adequate for non-linear systems; however, this is not the only way to obtain the solution.

An alternative way to solve Equations 2.33-2.35 is motivated by the definition of the local problem itself (the Dirichlet-to Neumann mapping). The idea is that, given a value for $\hat{\mathbf{u}}_h$ on the boundaries of a given element K, the solution for $\mathbf{u}_h$ and $\mathbf{Q}_h$ can be computed from it by solving the system of equations:
Find $(\mathbf{u}_h, \mathbf{Q}_h) \in (\mathcal{P}^k(K))^m \times (\mathcal{P}^k(K))^{m\times d}$ such that:

$$\left(\frac{\mathbf{u}_h}{\Delta t}, \mathbf{v}\right)_K - (\mathbf{F}+\mathbf{G}, \nabla\cdot\mathbf{v})_K - \langle(\widehat{\mathbf{F}}+\widehat{\mathbf{G}})\cdot\mathbf{n}, \mathbf{v}\rangle_{\partial K} = (\mathbf{f}, \mathbf{v})_K + \left(\frac{\mathbf{u}_h^-}{\Delta t}, \mathbf{v}\right)_K \tag{2.52}$$

$$-(\mathbf{u}_h, \nabla\cdot\mathbf{E})_K - (\mathbf{Q}_h, \mathbf{E})_K + \langle\hat{\mathbf{u}}_h, \mathbf{E}\cdot\mathbf{n}\rangle_{\partial K} = 0 \tag{2.53}$$

$\forall(\mathbf{v}, \mathbf{E}) \in (\mathcal{P}^k(K))^m \times (\mathcal{P}^k(K))^{m\times d}$. Where $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{G}}$ are defined above (Equation 2.36) and $\hat{\mathbf{u}}_h \in (\mathcal{P}^k(\partial K))^m$. The system might be written in residual notation:

$$\begin{cases} \mathbf{r}_{\mathbf{u}K}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h) = 0 \\ \mathbf{r}_{\mathbf{Q}K}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h) = 0 \end{cases} \tag{2.54}$$

where $\mathbf{r}_{\mathbf{u}K}$ and $\mathbf{r}_{\mathbf{Q}K}$ are derived from Equations 2.52 and 2.53. This system parametrizes $\mathbf{u}_h$ and $\mathbf{Q}_h$ as functions of $\hat{\mathbf{u}}_h$; $\mathbf{u}_h = \mathbf{u}_h(\hat{\mathbf{u}}_h)$ and $\mathbf{Q}_h = \mathbf{Q}_h(\hat{\mathbf{u}}_h)$. Once the solution to each local problem has been obtained, the sensitivities with respect to $\hat{\mathbf{u}}_h$ can be computed using the implicit function theorem and later introduced into the global

problem in order to update $\hat{\mathbf{u}}_h$. In essence, this procedure relies on updating $\hat{\mathbf{u}}_h$ only, assuming that the solution for the local problem and its sensitivities are always available (as if the local problem had an analytical solution available).

Despite the fact that this last assumption is far from true, the system defined by Equation 2.54 can be solved efficiently using Newton's iteration for a fixed $\hat{\mathbf{u}}_h$. The Newton's iterate for the local problem is:

$$\left[\begin{array}{cc} \frac{\partial \mathbf{r}_{\mathbf{u}K}}{\partial \mathbf{u}_h} & \frac{\partial \mathbf{r}_{\mathbf{u}K}}{\partial \mathbf{Q}_h} \\ \frac{\partial \mathbf{r}_{\mathbf{Q}K}}{\partial \mathbf{u}_h} & \frac{\partial \mathbf{r}_{\mathbf{Q}K}}{\partial \mathbf{Q}_h} \end{array} \right]_i \left\{ \begin{array}{c} \delta \mathbf{u}_h^i \\ \delta \mathbf{Q}_h^i \end{array} \right\} = - \left\{ \begin{array}{c} \mathbf{r}_{\mathbf{u}K} \\ \mathbf{r}_{\mathbf{u}K} \end{array} \right\}_i \tag{2.55}$$

$$\left\{ \begin{array}{c} \mathbf{u}_h^{i+1} \\ \mathbf{Q}_h^{i+1} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{u}_h^{i} \\ \mathbf{Q}_h^{i} \end{array} \right\} + \left\{ \begin{array}{c} \delta \mathbf{u}_h^{i} \\ \delta \mathbf{Q}_h^{i} \end{array} \right\} \tag{2.56}$$

where the matrix on the left hand side represents the matrix of derivatives of the residual (Equation 2.54) with respect to the local degrees of freedom. Convergence of this iteration will be quadratic provided the initial guess is close to the solution (as will be the case if the previous solution plus the first linear correction to account for $\delta\hat{\mathbf{u}}_h$ is used) and some other extra conditions hold (single root, bounded Hessian, etc. see [32, 44, 45]).

Once the local problem is solved, the sensitivities $\partial \mathbf{u}_h / \partial \hat{\mathbf{u}}_h$ and $\partial \mathbf{Q}_h / \partial \hat{\mathbf{u}}_h$ have to be computed in order to proceed with the iteration on the global problem. The computation of the sensitivities can be carried out using the implicit function theorem [33]. The system to solve for the sensitivities reads:

$$\left[\begin{array}{cc} \frac{\partial \mathbf{r}_{\mathbf{u}K}}{\partial \mathbf{u}_h} & \frac{\partial \mathbf{r}_{\mathbf{u}K}}{\partial \mathbf{Q}_h} \\ \frac{\partial \mathbf{r}_{\mathbf{Q}K}}{\partial \mathbf{u}_h} & \frac{\partial \mathbf{r}_{\mathbf{Q}K}}{\partial \mathbf{Q}_h} \end{array} \right] \left[\begin{array}{c} \frac{\partial \mathbf{u}_h}{\partial \hat{\mathbf{u}}_h} \\ \frac{\partial \mathbf{Q}_h}{\partial \hat{\mathbf{u}}_h} \end{array} \right]_K = - \left[\begin{array}{c} \frac{\partial \mathbf{r}_{\mathbf{u}K}}{\partial \hat{\mathbf{u}}_h} \\ \frac{\partial \mathbf{r}_{\mathbf{u}K}}{\partial \hat{\mathbf{u}}_h} \end{array} \right] \tag{2.57}$$

Once the sensitivities have been computed, the linearized version of the global problem (Equation 2.44) is solved using the chain rule to account for the dependence

of $\mathbf{u}_h$ and $\mathbf{Q}_h$ on $\hat{\mathbf{u}}_h$:

$$\left(\frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \mathbf{u}_h}\frac{\partial \mathbf{u}_h}{\partial \hat{\mathbf{u}}_h} + \frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \mathbf{Q}_h}\frac{\partial \mathbf{Q}_h}{\partial \hat{\mathbf{u}}_h} + \frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \hat{\mathbf{u}}_h}\right)\delta\hat{\mathbf{u}}_h = -\mathbf{r}_{\hat{\mathbf{u}}} \tag{2.58}$$

here, as in the case of the whole linearized system, the matrices involved are not computed explicitly but assembled in an element by element fashion.

2.4 Stabilization Parameter and Boundary Conditions

So far, the HDG scheme has been presented for a general system of conservation laws without giving much detail on how to choose the stabilization parameter (τ or $\mathbf{S}$) or how to implement the boundary conditions. The stabilization parameter is responsible for generating boundary terms that penalize the jumps between $\mathbf{u}_h$ and $\hat{\mathbf{u}}_h$ and render the system solvable. These terms can be regarded as dissipation, hence, the higher τ (or $\mathbf{S}$), the more stable the method. The existence and unicity of the solution rely on discrete energy inequalities (see [15]) and can be found in [40, 41] for the particular case of HDG.

Regarding the boundary conditions, DG in general and HDG in particular, present an important advantage over other methods in that these can be imposed through the fluxes in a very natural way. Simple cases, such as Dirichlet or Neumann boundary conditions, are trivial to implement and more complicated cases, such as far-field/non-reflecting boundary conditions with multiple waves entering and leaving the domain, can be dealt with gracefully, thanks to the flexibility that $\hat{\mathbf{u}}_h$ provides to set states on the boundary.

The different equations used as validation tests in this work, together with the choice of the stabilization parameter for each case and the boundary conditions more frequently encountered are described in Appendix A.

Chapter 3

Optimal Test Functions

Most finite element formulations employ the so-called Galerkin approach whereby the test and trial spaces are the same. Here a more general Petrov-Galerkin formulation, in which the trial and test spaces are different, will be described. The objective is to enhance stability and convergence by using a modified test space that accounts for upwinding. Earlier schemes such as the SUPG method [11] already exploited this idea by adding consistent terms to the weak formulation.

The approach described here was recently proposed by Demkowicz and Gopalakrishnan [21, 22] and relies on the computation of the test space on the fly, by solving an associated dual problem. This dual problem aims at computing the optimal test space that endows the problem with maximum stability and optimal error estimates. In order to describe it, first the general variational framework will be introduced, together with an important result about existence and uniqueness of the solution. Then, the optimal test functions will be introduced and described at a continuum level. To continue, the Discontinuous Petrov-Galerkin (DPG) scheme will be presented as a first attempt to apply this concept. Finally, a few comments on DPG will help motivate the Hybridizable Discontinuous Petrov-Galerkin (HDPG) scheme, that will be the subject of Chapter 4.

3.1 General Weak Formulation

The point of departure of the theory described here will be the general abstract variational formulation of a boundary value problem; find $u \in U$ s.t.

$$\mathcal{B}(u, v) = \langle f, v \rangle \qquad \forall v \in V \tag{3.1}$$

where $\mathcal{B}(\cdot, \cdot) : U \times V \mapsto \mathbb{R}$ is a continuous bilinear form on its arguments:

$$\mathcal{B}(u, v) \leq M \|u\|_U \|v\|_V \tag{3.2}$$

U and V are **different** Hilbert spaces (with norms denoted by $\|\cdot\|_U$ and $\|\cdot\|_V$) and $f \in V^*$ is an element of the dual space of V.

The Problem 3.1 is well posed if and only if the following condition holds:

$$\inf_{u \in U} \sup_{v \in V} \frac{\mathcal{B}(u, v)}{\|u\|_U \|v\|_V} \geq \gamma > 0 \tag{3.3}$$

this condition is referred to in the literature as the inf-sup condition, and was derived by Babuška [3]. It can be shown to be equivalent to the Brezzi condition [10] for mixed formulations (see [20, 55]) and for that reason, it is also known as the Ladyzhenskaya-Babuška-Brezzi (LBB) condition.

As written, Equation 3.3 states the condition for the well-posedness of the infinite dimensional weak formulation. However, in general, the interest lies in discrete versions of the weak formulation in which the spaces have finite dimensionality. The problem then reads: find $u_h \in U_h$ such that

$$\mathcal{B}(u_h, v_h) = \langle f, v_h \rangle \qquad \forall v_h \in V_h \tag{3.4}$$

where $U_h \subseteq U$, $V_h \subseteq V$ and $\dim U_h = \dim V_h$. The well-posedness (or stability) of Equation 3.4 is associated with the discrete version of Equation 3.3; namely, the

problem has solution and this solution is unique if and only if:

$$\inf_{u_h \in U_h} \sup_{v_h \in V_h} \frac{\mathcal{B}(u_h, v_h)}{\|u_h\|_U \|v_h\|_V} \geq \gamma_h > 0 \tag{3.5}$$

furthermore, if Equation 3.5 holds, the following error estimate holds too:

$$\|u - u_h\|_U \leq \frac{M}{\gamma_h} \inf_{w_h \in U_h} \|u - w_h\|_U \tag{3.6}$$

see [3, 20, 55] for more details.

It is important to notice that, by construction, the finite dimensionality of the spaces implies that $\gamma_h \leq \gamma$. Hence, in certain situations, while the infinite dimensional (or continuous) weak formulation might be well posed, the finite dimensional counterpart, for certain choices of the test and trial spaces, might not; a well known example of this would be the different interpolation spaces required for the treatment of incompressible flows with mixed formulations [10].

3.2 The Role of the Test Space

The idea, first proposed in the series of papers [21, 22, 24], consists on using trial spaces with good approximation properties (so that the right hand side on Equation 3.6 can be properly bounded) while letting the test space take care of the constant M/γ_h. Indeed, at a certain level, sharp error bounds are strongly related with stability of the solution since they are associated to the behavior of M/γ_h as h tends to zero. Provided γ_h is bounded away from zero as h decreases, optimal error estimates are expected. The optimal test space is defined here as the one that minimizes M/γ_h or maximizes γ_h.

3.2.1 Optimal Test Space: Theoretical Result

From an abstract point of view, the construction of the optimal test space requires two ingredients. The first one is the definition of an alternative (or energy) norm

$$\|u\|_E := \sup_{v \in V} \frac{\mathcal{B}(u,v)}{\|v\|_V} \tag{3.7}$$

that is equivalent to the norm on U provided the continuous inf-sup condition for the weak formulation holds ($\gamma > 0$) [22].

The second ingredient is a mapping $T : U \mapsto V$ that for every element of U associates an element $Tu \in V$ defined as:

$$(Tu, v)_V = \mathcal{B}(u,v) \qquad \forall v \in V \tag{3.8}$$

This mapping is well defined thanks to the applicability of the Riesz representation theorem to the bounded linear operator $\mathcal{B}(u,\cdot)$.

Combining the two definitions, it is straightforward to prove that the energy norm of a given element of U can be written as:

$$(u,u)_E := (Tu, Tu)_V \tag{3.9}$$

provided V is a Hilbert space (so that the Cauchy-Schwartz inequality holds). Hence, given a discrete trial space $U_h \subseteq U$ of finite dimensionality (N) and an associated linearly independent basis for it:

$$U_h = \text{span}\,\{e_j : \quad j = 1, ..., N\} \tag{3.10}$$

the optimal tests space for it is defined as

$$V_h = \text{span}\,\{Te_j : \quad j = 1, ..., N\} \tag{3.11}$$

Now it is easy to check that this test space gives the best approximation properties when U is normed using $\|\cdot\|_E$ since, on the one hand,

$$\mathcal{B}(u_h, v_h) = (Tu_h, v_h)_V \leq \|u\|_E \|v_h\|_V \Rightarrow M = 1 \tag{3.12}$$

while on the other

$$\sup_{v_h \in V_h} \frac{\mathcal{B}(u_h, v_h)}{\|v_h\|_V} = \sup_{v_h \in V_h} \frac{(Tu_h, v_h)_V}{\|v_h\|_V} \geq \left(Tu_h, \frac{Tu_h}{\|Tu_h\|_V} \right)_V = \|u_h\|_E \Rightarrow \gamma_h = 1 \tag{3.13}$$

hence the error estimate in Equation 3.6 holds with constant $M/\gamma_h = 1$. Not only that, but the discrete operator becomes symmetric positive definite; given two elements of the trial and test space: u_{hi} and Tu_{hj}, the bilinear form is equivalent to:

$$\mathcal{B}(u_{hi}, Tu_{hj}) = (Tu_{hi}, Tu_{hj})_V = (Tu_{hj}, Tu_{hi})_V = \mathcal{B}(u_{hj}, Tu_{hi}) \tag{3.14}$$

hence the system can be solved using well developed iterative techniques such as Conjugate Gradients [26].

3.2.2 Optimal Test Space: Discrete Approximation

As written above, the optimal test space can be computed by just solving the inverse Riesz mapping (Equation 3.8). However, this task is not trivial since it involves inverting a continuous operator. For the sake of computability, what Demkowicz et al. propose in [22] is to assume that the optimal test space does not live in an infinite dimensional space of functions V but a discrete subspace of it $\tilde{V}_h$. The approximate optimal test functions are then extracted from inverting the discrete mapping: find $T_h e_i \in \tilde{V}_h$

$$\mathcal{B}(e_i, v) = (T_h e_i, v) \quad \forall v \in \tilde{V}_h \tag{3.15}$$

It is expected that as $\tilde{V}_h$ is enriched, the approximate test functions $(T_h e_i)$ converge towards the exact optimal ones (Te_i) so that the problem inherits the stability prop-

erties of the original inf-sup maximization.

The choice of $\tilde{V}_h$ is only restricted by dimensionality; it is required that $\dim \tilde{V}_h > \dim U_h$ in order to allow for improvement in the discrete inf-sup constant. For the sake of simplicity, polynomials are used for $\tilde{V}_h$ since they are complete, easy to compute and can be made well conditioned. In particular, if the trial space is associated to polynomials of a certain order k, $U_h \in \mathcal{P}^k$, the test space will belong to polynomials of a higher order $k + \Delta k$, $V_h \in \mathcal{P}^{k+\Delta k}$. Other than that, any set of functions that satisfies the regularity requirements imposed by the bilinear form $\mathcal{B}(\cdot, \cdot)$ is equally valid.

3.3 DPG Scheme

The method proposed by Demkowicz et al. [21, 22, 24] consists on applying this approximate optimal test space to the Discontinuous Petrov-Galerkin scheme (DPG) introduced by Bottasso et al. [7, 8]. The DPG scheme is constructed in 3 steps:

1. Write the governing equations as a system of first order PDEs by introducing new unknowns for the derivatives of the solution and extra equations to define these new unknowns.

2. Assume discontinuous test and trial spaces associated to a triangulation $\mathcal{T}_h$ as described in Chapter 2.

3. Derive weak formulations by integrating by parts. Given the discontinuous nature of the spaces, new terms will appear at the interfaces between elements. Unlike general DG methods, these new terms will be regarded as new unknowns and solved for together with the degrees of freedom for the solution inside each element. See [7, 8] for details on the discretization of convective-diffusive systems.

The key step in the original DPG is the definition of the test space so that the final discrete system of equations is solvable. This choice relies on a counting argument (for the system to be square) together with an elaborated construction of the test functions (so that the inf-sup condition is satisfied and the system is non-singular). A more detailed explanation can be found in [7, 8, 13]. The modified DPG scheme avoids this inconvenience by letting the definition of the test space to be carried out on the fly by means of the discrete inverse mapping (Equation 3.15). From an implementation point of view, the weak formulation and the computation of the test space can be combined resulting in a simplified structure of the problem.

To this end, first, let $\mathbb{e}_i$ denote the vector of coefficients of an element of the basis of U_h and let $\mathbb{t}_{hi}$ denote the vector of coefficients of its associated approximate test function. Similarly, let $\tilde{\mathbb{v}}_h$ denote the vector of coefficients of a general element of $\tilde{V}_h$. Since every element of V_h (the test space) belongs to $\tilde{V}_h$ (the search space), both vectors $\mathbb{t}_{hi}$ and $\tilde{\mathbb{v}}_h$ have the same length, more precisely, $\mathbb{t}_{hi}, \tilde{\mathbb{v}}_h \in \mathbb{R}^n$ where n represents the number of degrees of freedom of $\tilde{V}_h$. Also, let $\mathbb{u}_h$ denote the vector of coefficients of an element of U_h, $\mathbb{u}_h \in \mathbb{R}^m$, where m denotes the number of degrees of freedom of U_h. Now, both problems can be written in matrix form as:

$$\tilde{\mathbb{v}}_h^T X_V \mathbb{t}_{hi} = \tilde{\mathbb{v}}_h^T B \mathbb{e}_{hi} \quad \forall \tilde{\mathbb{v}}_h \in \mathbb{R}^m, \qquad i = 1, ..., m \tag{3.16}$$

$$\mathbb{t}_{hi}^T B \mathbb{u}_h = \mathbb{t}_{hi}^T F \qquad i = 1, ..., m \tag{3.17}$$

where $X_V \in \mathbb{R}^{n \times n}$ represents the inner product of the space $\tilde{V}_h$, $B \in \mathbb{R}^{n \times m}$ represents the matrix associated to the bilinear form $\mathcal{B} : U_h \times \tilde{V}_h \mapsto \mathbb{R}$ and $F \in \mathbb{R}^{n \times 1}$ represents the usual duality pairing on the right hand side $\langle f, v_h \rangle$.

Equation 3.16 can be inverted for each i to obtain the approximated test spaces thanks to the invertibility of X_V (it is the metric of an inner product, hence, symmetric

positive definite):

$$\mathbb{t}_{hi} = X_V^{-1} B \mathbb{e}_{hi} \quad i = 1, ..., m \tag{3.18}$$

combining Equation 3.18 with Equation 3.17 yields the variational form to be solved for $\mathbb{u}_h$:

$$\mathbb{w}^T B^T X_V^{-1} B \mathbb{u}_h = \mathbb{w}^T B^T X_V^{-1} F \quad \forall \mathbb{w} \in \mathbb{R}^m \tag{3.19}$$

where $\mathbb{w}$ represents a general variation in the trial space U_h.

As mentioned in the previous section, the resulting system is symmetric positive definite, which implies it can be derived from a minimization statement, namely;

$$\mathbb{u}_h = \arg \min_{\mathbb{w}_h} R^T X_V^{-1} R \tag{3.20}$$

where $R = B\mathbb{w}_h - F$ represents the residual vector. This minimization statement provides an alternative point of departure for the extension of this method to the non-linear problem. Furthermore, this problem is equivalent to:

$$\mathbb{u}_h = \arg \min_{\mathbb{w}_h \in \mathbb{R}^m} \max_{\mathbb{v}_h \in \mathbb{R}^n} \frac{\mathbb{v}_h^T R}{\|\mathbb{v}_h\|_V} \tag{3.21}$$

which shows the connection between the optimal test function theory and the inf-sup (or min-max) condition. Notice that the system above (Equation 3.20) can only be solved efficiently when the matrix for the inner product (X_V) is easy to invert; in the Continuous Galerkin context, this would not be the case which explains the choice of a Discontinuous Galerkin scheme (in this case DPG) as a basis.

3.4 Comments

As presented, the modified DPG scheme of Demkowicz et al. seems to be a suitable framework to deal with the stability issues that more general FEM present in several instances such as, for example, convection-diffusion problems. Indeed, the method has been applied to the well known Peterson's example [49] in order to asses how the extra stability affects convergence. This test case was tailored to take advantage of the error layers that DG presents in the interfaces parallel to the flow and confirm the theoretical result that the order of convergence in the pure convection regime can only be $k + 1/2$ [31]. The results obtained using DPG (see [22] p.84) indicate convergence with optimal order $k + 1$. In the same spirit, the DPG method can be applied to a convection case with a forcing term that generates a sharp gradient (imitating an underresolved boundary layer) using one single element. The results (see [21] p.1567) show how the oscillation is strongly reduced with respect to a general DG scheme by at least an order of magnitude. So far the scheme has been extended to the wave equation [57], the Poisson equation [23] and the Burgers equation in 1D [14].

Despite these good results, DPG presents several weaknesses that will hinder its application to more complicated problems, e.g. unsteady compressible flows, unless properly addressed. The most relevant would be:

- The DPG is not locally (nor globally) conservative in the sense that the constant mode might not belong to the test space, hence, guaranteeing accurate shock propagation is not straightforward. This will be an issue that needs to be addressed for several practical problems.

- Even though the modified DPG enhances the stability of the original DPG scheme of Bottasso et al., it is harder to implement and solve due to the structure of the system to invert ($B^T X_V^{-1} B$). On the one hand, the static condensation as a function of the traces, that could be carried out in the original DPG (much like the local solver described for HDG) is no longer available. On the other hand, the sparsity pattern of B will be changed by the $B^T X_V^{-1} B$ operation,

hence generating a non-compact stencil. Finally, the condition number of the system is squared, thus, diffusive operators might generate serious issues when iterative solvers are used.

- In [14] the authors apply DPG to the Burgers and Navier-Stokes equations in 1D and find convergence problems as viscosity vanishes. Unless this is fixed, the method will be useless in the 2D or 3D context for the solution of high Reynolds number flows.

The objective of the rest of this thesis is to describe a new method; the Hybridizable Discontinuous Petrov-Galerkin scheme (HDPG), as an application of some of the ideas exposed in this chapter to the HDG framework, suitably modified/augmented in order to render a method with similar stability properties that avoids the issues just mentioned.

Chapter 4

Hybridizable Discontinuous Petrov-Galerkin

In previous chapters, both the Hybridizable Discontinuous-Galerkin (HDG) and Discontinuous Petrov-Galerkin (DPG) schemes where introduced and described. From what was said there, it can be concluded that these schemes complement each other; while the HDG scheme is conservative and involves less globally coupled degrees of freedom (only the trace $\hat{u}_h$ on interfaces), the DPG scheme is stable and optimally convergent. Hence, the question is, can both concepts be combined in such a way as to generate a scheme that incorporates as many of the advantages as possible with the minimum drawbacks? The answer proposed in this thesis is the Hybridizable Discontinuous Petrov-Galerkin scheme (HDPG) [39] that will be discussed in this chapter.

The HDPG scheme stems from the following observation: the local problem in HDG represents a paradigm to break a conservation law into subdomains, e.g. elements of a triangulation, and glue them all together through a conservativity argument across edges. Hence, why not apply the DPG tools to the local problem inside an element so that the Dirichlet-Neumann mapping gets stabilized while the degrees of freedom remain local to the element? This is the main idea behind the HDPG scheme.

The structure of this chapter is as follows. First, the HDPG local solver will be described for the case of a hyperbolic system and complemented with an additional constraint to account for conservativity. Then, different strategies to solve the system will be described. Next, the elliptic case will be discussed. Finally, some results for single element problems will be presented to demonstrate the enhanced stability of the proposed approach.

4.1 HDPG for Hyperbolic Systems

The point of departure for the HDPG formulation is a general unsteady hyperbolic system of conservation laws:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{f} \qquad \text{in} \quad \Omega \times (0,T] \tag{4.1}$$

$$\mathbf{u} = \mathbf{u}_0 \qquad \text{in} \quad \Omega \times \{t=0\} \tag{4.2}$$

$$\mathbf{b}(\mathbf{u}) = \mathbf{g} \qquad \text{on} \quad \partial\Omega \times (0,T] \tag{4.3}$$

that, given a triangulation $\mathcal{T}_h$ over Ω, can be split into a local and global problem by introducing the traces of $\mathbf{u}$ on the faces of the triangulation $\mathcal{E}_h$. The continuous local problem for the system of interest is:

$$\frac{\partial \mathbf{u}^\lambda}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}^\lambda) = \mathbf{f} \qquad \text{in} \quad K \times (0,T] \tag{4.4}$$

$$\mathbf{u}^\lambda = \lambda(t) \qquad \text{on} \quad \partial K \times (0,T] \tag{4.5}$$

while the global problem reads:

$$\mathbf{F}(\mathbf{u}^\lambda)^+ \cdot \mathbf{n}^+ + \mathbf{F}(\mathbf{u}^\lambda)^- \cdot \mathbf{n}^- = 0 \qquad \text{on} \quad I, \quad \forall I \in \mathcal{E}_h^i \tag{4.6}$$

$$\mathbf{b}(\lambda) - \mathbf{g} = 0 \qquad \text{on} \quad \partial\Omega \tag{4.7}$$

In HDG, the solution is assumed to belong to a test space ($\mathbf{V}_h^k \times \mathbf{M}_h^k$) and the algebraic system of equations to solve is formed by weighting against the same space after

introducing the appropriate numerical fluxes (Equations 2.33-2.35). In HDPG, the global problem is treated in the same way as in HDG, this is, forcing the conservation of fluxes, but using DPG on the local problem to compute $\mathbf{u}_h = \mathbf{u}_h(\hat{\mathbf{u}}_h)$ and its derivatives (the exact local solver introduced in §2.3.3).

4.1.1 Local Problem

The first ingredient to consider here will be the discrete optimal test functions introduced in §3.2.2 and §3.3; the objective is to apply them to the local problem (Equation 4.4-4.5) in such a way that the stability of the solution is increased and the internal degrees of freedom for $\mathbf{u}$ are still parametrized by the traces (λ in continuous sense or $\hat{\mathbf{u}}_h$ in the discrete one) and can be eliminated element-wise.

To that end, the solution is assumed to belong to the usual polynomial spaces of order k: $(\mathbf{u}_h, \hat{\mathbf{u}}_h) \in \mathbf{V}_h^k \times \mathbf{M}_h^k$, and the test space is assumed to belong to polynomials of Δk order higher: $\mathbf{v}_h \in \mathbf{V}_h^{k+\Delta k}$. The local problem residual is then obtained through integration by parts:

$$r_{\mathbf{u}K}(\mathbf{u}_h, \mathbf{v}_h; \hat{\mathbf{u}}_h) = \left(\frac{\mathbf{u}_h - \mathbf{u}_h^-}{\Delta t}, \mathbf{v}_h\right)_K - (\mathbf{F}, \nabla \cdot \mathbf{v}_h)_K - \langle \widehat{\mathbf{F}} \cdot \mathbf{n}, \mathbf{v}_h \rangle_{\partial K} - (\mathbf{f}, \mathbf{v}_h)_K \quad (4.8)$$

where the time derivative has already been discretized using Backward Euler (as described for HDG in §2) and can be easily extended to other time stepping schemes. The only terms to be determined are the fluxes at the boundaries, that will follow the same definition as in HDG:

$$\widehat{\mathbf{F}} = \mathbf{F}(\hat{\mathbf{u}}_h) + \mathbf{S}(\mathbf{u}_h - \hat{\mathbf{u}}_h) \quad (4.9)$$

The problem is then written in the DPG min-max fashion (see Equation 3.21):

$$\mathbf{u}_h = \arg \min_{\mathbf{u}_h \in \mathbf{V}_h^k} \max_{\mathbf{v}_h \in \mathbf{V}_h^{k+\Delta k}} \frac{r_{\mathbf{u}K}(\mathbf{u}_h, \mathbf{v}_h; \hat{\mathbf{u}}_h)}{\|\mathbf{v}_h\|_V} \quad (4.10)$$

To simplify this statement, the residual is written in vector form by integrating

against the basis for the test space ($\phi_i \in \mathbf{V}_h^{k+\Delta k}$);

$$\mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h) = \left\{ \begin{array}{c} \dots \\ r_{\mathbf{u}K}(\mathbf{u}_h, \phi_i; \hat{\mathbf{u}}_h) \\ \dots \end{array} \right\} \tag{4.11}$$

where the index $i = 1, \dots, N$ runs in the number of basis functions for $\mathbf{V}_h^{k+\Delta k}$. Defined this way, the residual for the local problem can be written as:

$$r_{\mathbf{u}K}(\mathbf{u}_h, \mathbf{v}_h; \hat{\mathbf{u}}_h) = \mathbb{v}_h^T \cdot \mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h) \tag{4.12}$$

where $\mathbb{v}_h^T \in \mathbb{R}^n$ represents the vector of coefficients of $\mathbf{v}_h$ expanded in the basis ϕ_i. The local problem then reads:

$$\mathbf{u}_h = \arg \min_{\mathbf{u}_h \in \mathbf{V}_h^k} \max_{\mathbb{v}_h \in \mathbb{R}^n} \frac{\mathbb{v}_h^T \cdot \mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h)}{\sqrt{\mathbb{v}_h^T X_V \mathbb{v}_h}} \tag{4.13}$$

where X_V represents the inner product matrix for the space $\mathbf{V}_h^{k+\Delta k}$ associated to the basis ϕ_i. Now, the maximization can be solved explicitly to yield the following local problem:

$$\mathbf{u}_h = \arg \min_{\mathbf{u}_h \in \mathbf{V}_h^k} \quad \frac{1}{2} \mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h)^T X_V^{-1} \mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h) \tag{4.14}$$

Applying the first order optimality conditions to the local problem [6], the non-linear system of equations that has to be solved to obtain $\mathbf{u}_h$ as a function of $\hat{\mathbf{u}}_h$ reads:

$$\left[\frac{\partial \mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h)}{\partial \mathbf{u}_h} \right]^T X_V^{-1} \mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h) = 0 \tag{4.15}$$

where the term $[\partial \mathbf{r}(\mathbf{u}_h; \hat{\mathbf{u}}_h)/\partial \mathbf{u}_h]$ represents the Jacobian of the local residual and plays the same role as the discrete bilinear form B in §3. Similarly, the Jacobian matrix together with the inner product X_V describe the inverse of the Riesz mapping (homologue to Equation 3.18). This non-linear system can be solved using Newton's

method or another iterative technique.

4.1.2 Imposing Conservativity

As written in Equation 4.15, the HDPG formulation is non conservative since the inverse of the Riesz mapping is not guaranteed to contain the constant mode. The importance of conservative schemes has been highlighted several times throughout this manuscript. Without conservativity, shocks are not propagated at the right speed and results are meaningless.

In order to fix this issue, the problem has to be treated in a different manner. Instead of looking at the way to impose conservativity in Equation 4.15, it is better to take one step back and have a look at Equation 4.14. The problem is a general minimization statement, and can be constrained to impose the integration against the constant mode. Since the test space consists of polynomials of degree $k+\Delta k$, the constant mode ($\mathbf{v}_h = 1$) can be exactly represented in that basis. The constrained problem then reads:

$$\mathbf{u}_h = \arg\min_{\mathbf{u}_h \in \mathbf{V}_h^k} \quad \frac{1}{2}\mathbf{r}(\mathbf{u}_h;\hat{\mathbf{u}}_h)^T X_V^{-1}\mathbf{r}(\mathbf{u}_h;\hat{\mathbf{u}}_h) \tag{4.16}$$

$$\text{s.t.} \quad r_{u_i K}(\mathbf{u}_h, 1;\hat{\mathbf{u}}_h) = 0 \qquad i = 1, ..., m \tag{4.17}$$

where the last m equations represent the conservativity condition for each of the m conservation laws that compose the system and can be written in vector notation as:

$$r_{u_i K}(\mathbf{u}_h, 1;\hat{\mathbf{u}}_h) = \mathbb{c}_i^T \mathbf{r}(\mathbf{u}_h;\hat{\mathbf{u}}_h) \qquad i = 1, ..., m \tag{4.18}$$

where $\mathbb{c}_i$ is the vector of coefficients of the constant mode for component i.

Now, instead of the first order optimality conditions, the local problem has to satisfy the Karush-Kuhn-Tucker conditions (or KKT conditions [6]) that represent optimality for the Lagrangian function together with primal feasibility. Namely, the

solution to the local problem has to satisfy the following system of equations:

$$\left[\frac{\partial \mathbf{r}(\mathbf{u}_h;\hat{\mathbf{u}}_h)}{\partial \mathbf{u}_h}\right]^T X_V^{-1}\mathbf{r}(\mathbf{u}_h;\hat{\mathbf{u}}_h) + \sum_{i=1}^{m}\lambda_i \mathbb{c}_i^T\left[\frac{\partial \mathbf{r}(\mathbf{u}_h;\hat{\mathbf{u}}_h)}{\partial \mathbf{u}_h}\right] = 0 \tag{4.19}$$

$$\mathbb{c}_i^T\mathbf{r}(\mathbf{u}_h;\hat{\mathbf{u}}_h) = 0 \qquad i = 1,...,m \tag{4.20}$$

where λ_i are the Lagrange multipliers (or dual variables) that are also part of the solution. This system is similar to the unconstrained case (Equation 4.15), however it has m more equations (the conservativity conditions) and m more unknowns (the Lagrange multipliers). Even though this makes the system more expensive to solve, the effect is small, since only one extra unknown is required per component regardless of the polynomial order; when the polynomial order is high, the associated cost is negligible.

4.1.3 Local Problem Solution

Once the system has been modified to make it conservative, the remaining task is to solve the constrained minimization statement to obtain the $\mathbf{u}_h = \mathbf{u}_h(\hat{\mathbf{u}}_h)$ dependence. Being this a non-linear problem, iterative solvers are mandatory in order to find a solution through a series of iterations $\mathbf{u}_h^{i+1} = \mathbf{u}_h^i + \delta\mathbf{u}_h$. Here, in particular, two approaches that rely on linearization steps will be used, the only difference between them being the order in which the linearization and optimality steps are taken.

Linearization before KKT: Constrained Gauss-Newton (CGN) Algorithm

For this approach, the linearization step is taken before posing the first order optimality conditions. Let $\mathbf{u}_h^i$ be the current iterate, and let $\delta\mathbf{u}_h$ denote the solution update, then the first order perturbation of the residual is:

$$\mathbf{r}(\mathbf{u}_h^i + \delta\mathbf{u}_h;\hat{\mathbf{u}}_h) \approx \mathbf{r}(\mathbf{u}_h^i;\hat{\mathbf{u}}_h) + \left[\frac{\partial \mathbf{r}(\mathbf{u}_h^i;\hat{\mathbf{u}}_h)}{\partial \mathbf{u}_h}\right]\delta\mathbf{u}_h \tag{4.21}$$

If this linearization is plugged into the constrained minimization statement of the HDPG method (Equations 4.16-4.17):

$$\delta\mathbf{u}_h = \arg\min_{\delta\mathbf{u}_h \in \mathbf{V}_h^k} \quad \frac{1}{2}\left(\mathbf{r} + \frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\delta\mathbf{u}_h\right)^T X_V^{-1}\left(\mathbf{r} + \frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\delta\mathbf{u}_h\right) \tag{4.22}$$

$$\text{s.t.} \quad \mathbb{c}_i^T\left(\mathbf{r} + \frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\delta\mathbf{u}_h\right) = 0 \qquad i = 1, ..., m \tag{4.23}$$

where $\mathbf{r}$ and its Jacobian matrix are assumed to be evaluated at the current iterate $\mathbf{u}_h^i$ and $\hat{\mathbf{u}}_h$. In order to obtain the $\delta\mathbf{u}_h$ that minimizes the linearized problem, the KKT conditions can be invoked. All in all, the system to solve is:

$$\left[\begin{array}{c|ccc} \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T X_V^{-1}\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] & \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T \mathbb{c}_1 & \cdots & \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T \mathbb{c}_m \\ \hline \mathbb{c}_1^T\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{c}_m^T\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] & 0 & \cdots & 0 \end{array}\right] \left\{\begin{array}{c} \delta\mathbf{u}_h \\ \hline \lambda_1 \\ \vdots \\ \lambda_m \end{array}\right\} =$$

$$= -\left\{\begin{array}{c} \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T X_V^{-1}\mathbf{r} \\ \hline \mathbb{c}_1^T\mathbf{r} \\ \vdots \\ \mathbb{c}_m^T\mathbf{r} \end{array}\right\} \tag{4.24}$$

This solution method is also known as the constrained Gauss-Newton algorithm and is used to solve non-linear least squares problems like the one HDPG poses. It is easy to see that the system formed at each iteration is symmetric and definite under general assumptions for the rank of the Jacobian matrix (easy to prove by taking the Schur complement of the system), hence invertible. One of its disadvantages is that the convergence depends on the conditioning of the squared Jacobian and also on the value of the cost function at convergence, that might be different from zero. For more details about this algorithm see [44].

KKT before Linearization: Sequential Quadratic Programming (SQP) Algorithm

The second approach proposed here is based on the application of Newton's method to the KKT conditions stated in Equations 4.19 and 4.20. For this, the KKT conditions are linearized around the current iterate $(\mathbf{u}_h, \lambda_i)^i$ and the update is computed out of it, namely:

$$\left[\begin{array}{c|ccc} \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T X_V^{-1}\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] + \left[\frac{\partial^2 \mathbf{r}}{\partial \mathbf{u}_h^2}\right]\cdot\left(X_V^{-1}\mathbf{r} + \sum_{i=1}^m \lambda_i \mathbb{C}_i\right) & \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T \mathbb{C}_1 & \cdots & \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T \mathbb{C}_m \\ \hline \mathbb{C}_1^T\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}_m^T\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] & 0 & \cdots & 0 \end{array}\right] \times$$

$$\times \left\{\begin{array}{c} \delta \mathbf{u}_h \\ \hline \delta\lambda_1 \\ \vdots \\ \delta\lambda_m \end{array}\right\} = -\left\{\begin{array}{c} \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T X_V^{-1}\mathbf{r} + \sum_{i=1}^m \lambda_i \mathbb{C}_i^T\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] \\ \hline \mathbb{C}_1^T\mathbf{r} \\ \vdots \\ \mathbb{C}_m^T\mathbf{r} \end{array}\right\} \tag{4.25}$$

where the term $\partial^2\mathbf{r}/\partial\mathbf{u}_h^2$ represents the second derivative of the residual, and is a third order tensor. This method is also known as the Sequential Quadratic Programming (SQP) algorithm in the optimization community, and benefits from the convergence properties of Newton's method for systems, this is, quadratic convergence when the system is not singular and the iterate is close enough.

CGN vs. SQP

Both algorithms have been implemented and tested and both converge to the same solution. This has to be the case since the system that both algorithms satisfy at convergence is the same one (Equations 4.19-4.20). While for the SQP approach this assertion is straightforward to prove (it is enough to see that the KKT conditions are precisely the right hand side of the system) for the CGN case it is enough to notice that at convergence $\delta\mathbf{u}_h = 0$ while $\lambda_i \neq 0$ in general, hence, moving the Lagrange multiplier terms to the right hand side produces again the KKT conditions.

Even though both converge to the same solution, they present different behavior during the iteration. While the CGN algorithm is robust in the sense that it continuously reduces the value of the cost function, it usually reaches a point in which the convergence rate is slow, hence, requiring more iterations than desirable. On the other hand, the SQP algorithm converges quadratically when close enough to the solution, however it is not guaranteed to converge. Also, it is more expensive per iteration than CGN due to the second order derivatives and the associated tensor contraction.

Therefore, the strategy proposed is based on using the CGN iterate on the first steps and switching to the SQP iterate to finally converge the solution. The switch relies on monitoring the relative change in the solution $\|\delta \mathbf{u}_h\|/\|\mathbf{u}_h\|$ and has to be defined a priori. A value of $\|\delta \mathbf{u}_h\|/\|\mathbf{u}_h\| = \mathcal{O}(1)$ was found to work for most of cases tested. The iteration is stopped when the relative update in the solution is below a certain user defined value, in all the cases presented here this was set to $\|\delta \mathbf{u}_h\|/\|\mathbf{u}_h\| = \mathcal{O}(10^{-6}) - \mathcal{O}(10^{-8})$.

4.1.4 Local Problem Sensitivities

Once the local problem has been solved, both the solution ($\mathbf{u}_h = \mathbf{u}_h(\hat{\mathbf{u}}_h)$) and the sensitivities ($\partial\mathbf{u}_h/\partial\hat{\mathbf{u}}_h$) have to be transferred to the global problem. This requires one extra computation for the sensitivities using the implicit function theorem as in the HDG method [33]. For this, the KKT conditions (Equations 4.19 and 4.20) are linearized in both $\mathbf{u}_h$ and $\hat{\mathbf{u}}_h$ and the sensitivities are extracted from the system:

$$\left[\begin{array}{c|ccc} \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T X_V^{-1}\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] + \left[\frac{\partial^2 \mathbf{r}}{\partial \mathbf{u}_h^2}\right]\cdot\left(X_V^{-1}\mathbf{r} + \sum_{i=1}^m \lambda_i \mathbb{C}_i\right) & \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T \mathbb{C}_1 & \cdots & \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T \mathbb{C}_m \\ \hline \mathbb{C}_1^T\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}_m^T\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right] & 0 & \cdots & 0 \end{array}\right] \times$$

$$\times \left\{\begin{array}{c} \frac{\partial \mathbf{u}_h}{\partial \hat{\mathbf{u}}_h} \\ \hline \frac{\partial \lambda_1}{\partial \hat{\mathbf{u}}_h} \\ \vdots \\ \frac{\partial \lambda_m}{\partial \hat{\mathbf{u}}_h} \end{array}\right\} = -\left\{\begin{array}{c} \left[\frac{\partial^2 \mathbf{r}}{\partial \mathbf{u}_h \partial \hat{\mathbf{u}}_h}\right]^T \cdot \left(X_V^{-1}\mathbf{r} + \sum_{i=1}^m \lambda_i \mathbb{C}_i\right) + \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_h}\right]^T X_V^{-1}\left[\frac{\partial \mathbf{r}}{\partial \hat{\mathbf{u}}_h}\right] \\ \hline \mathbb{C}_1^T\left[\frac{\partial \mathbf{r}}{\partial \hat{\mathbf{u}}_h}\right] \\ \vdots \\ \mathbb{C}_m^T\left[\frac{\partial \mathbf{r}}{\partial \hat{\mathbf{u}}_h}\right] \end{array}\right\} \tag{4.26}$$

Notice this is a linear system with multiple right hand sides (one for each degree of freedom in $\hat{\mathbf{u}}_h$), thus the system matrix only has to be inverted once. Furthermore, the matrix contains the same terms as in the SQP case and can be reused from the last SQP iteration.

4.1.5 Global Problem

In the same spirit as the HDG scheme introduced in §2.3.3, the global problem in HDPG is defined by the conservation of fluxes across interfaces, assuming that the mapping $\mathbf{u}_h = \mathbf{u}_h(\hat{\mathbf{u}}_h)$ and the sensitivities are computed exactly at each step of the

iteration. The global problem then reads; find $\hat{\mathbf{u}}_h \in \mathbf{M}_h^k$ such that,

$$\langle \widehat{\mathbf{F}} \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h \setminus \partial\Omega} + \langle \mathbf{b}(\mathbf{u}_h, \hat{\mathbf{u}}_h) - \mathbf{g}, \mu \rangle_{\partial\Omega} = 0 \qquad \forall \mu \in \mathbf{M}_h^k \tag{4.27}$$

where

$$\widehat{\mathbf{F}} = \mathbf{F}(\hat{\mathbf{u}}_h) + \mathbf{S}(\hat{\mathbf{u}}_h)(\mathbf{u}_h - (\hat{\mathbf{u}}_h)) \cdot \mathbf{n} \tag{4.28}$$

The weak formulation can be written as an algebraic system by weighting against all the basis functions of the test space $\phi_i \in \mathbf{M}_h^k$:

$$\mathbf{r}_{\hat{\mathbf{u}}}(\mathbf{u}_h, \hat{\mathbf{u}}_h) = \left\{ \begin{array}{c} \dots \\ \langle \widehat{\mathbf{F}} \cdot \mathbf{n}, \phi_i \rangle_{\partial \mathcal{T}_h \setminus \partial\Omega} + \langle \mathbf{b}(\mathbf{u}_h, \hat{\mathbf{u}}_h) - \mathbf{g}, \phi_i \rangle_{\partial\Omega} \\ \dots \end{array} \right\} \tag{4.29}$$

and can be solved using Newton's method. Each update in the solution can be computed by solving the system:

$$\left(\frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \mathbf{u}_h} \frac{\partial \mathbf{u}_h}{\partial \hat{\mathbf{u}}_h} + \frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \hat{\mathbf{u}}_h} \right) \delta \hat{\mathbf{u}}_h = -\mathbf{r}_{\hat{\mathbf{u}}} \tag{4.30}$$

where the first term has been simplified using the chain rule and the sensitivities of the local problem with respect to the numerical traces presented in §4.1.4. In order to clarify how the local and global solvers interact, how the solutions are updated, etc., the HDPG scheme is described in the form of an algorithm on the next page.

Algorithm 4.1 HDPG Scheme for Hyperbolic Systems

Given some initial condition, set the initial values...
$\hat{\mathbf{u}}_h^1 \leftarrow \hat{\mathbf{u}}_h^0$
$\mathbf{u}_h^1 \leftarrow \mathbf{u}_h^0$

...Take T time steps...
for $i = 1 \rightarrow T$ **do**

...Solve the system using Newton's iteration...
while $\|\delta\hat{\mathbf{u}}_h\|/\|\hat{\mathbf{u}}_h^i\| \geq \epsilon$ **do**

...First solving the local problem for each element of $\mathcal{T}_h$...
for $j = 1 \rightarrow N$ **do**

...Using HDPG and Newton's method...
while $\|\delta\mathbf{u}_{hj}\|/\|\mathbf{u}_{hj}^i\| \geq \epsilon$ **do**
if $\|\delta\mathbf{u}_{hj}\|/\|\mathbf{u}_{hj}^i\| \geq \mathcal{O}(1)$ **then**
Solve the CGN system (Equation 4.24) $\rightarrow \delta\mathbf{u}_{hj}$
else
Solve the SQP system (Equation 4.25) $\rightarrow \delta\mathbf{u}_{hj}$
end if
$\mathbf{u}_{hj}^i \leftarrow \mathbf{u}_{hj}^i + \delta\mathbf{u}_{hj}$
end while
Solve the senstivities system (Equation 4.26) $\rightarrow \frac{\partial\mathbf{u}_{hj}^i}{\partial\hat{\mathbf{u}}_h}$
end for

...Then solving the global problem...
Solve the global problem (Equation 4.30) $\rightarrow \delta\hat{\mathbf{u}}_h$

...And updating using damped Newton with a suitable α...[32]
$\hat{\mathbf{u}}_h^i \leftarrow \hat{\mathbf{u}}_h^i + \alpha\delta\hat{\mathbf{u}}_h$
$\mathbf{u}_h^i \leftarrow \mathbf{u}_h^i + \alpha\frac{\partial\mathbf{u}_h^i}{\partial\hat{\mathbf{u}}_h}\delta\hat{\mathbf{u}}_h$
end while
$\hat{\mathbf{u}}_h^{i+1} \leftarrow \hat{\mathbf{u}}_h^i$
$\mathbf{u}_h^{i+1} \leftarrow \mathbf{u}_h^i$
end for

4.2 HDPG for Elliptic Operators

As presented above, the HDPG scheme can be readily applied to elliptic operators provided the definition of the local problem residual is augmented accordingly like it was done in §2. The equations of interest, written as a first order system are:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{F}(\mathbf{u}) + \mathbf{G}(\mathbf{u}, \mathbf{Q})) = \mathbf{f} \qquad \text{in} \quad \Omega \times (0, T] \tag{4.31}$$

$$\mathbf{Q} - \nabla \mathbf{u} = 0 \qquad \text{in} \quad \Omega \times (0, T] \tag{4.32}$$

$$\mathbf{u} = \mathbf{u}_0 \qquad \text{in} \quad \Omega \times \{t = 0\} \tag{4.33}$$

$$\mathbf{b}(\mathbf{u}, \mathbf{Q}) = 0 \qquad \text{on} \quad \partial\Omega \times (0, T] \tag{4.34}$$

where $\mathbf{Q}$ represents the kinematic variables, related to the gradients of the solution $\mathbf{u}$ through Equation 4.32.

4.2.1 Local Problem

The local problem is then defined by Equations 4.31 and 4.32 on each element, together with the Dirichlet boundary conditions:

$$\frac{\partial \mathbf{u}^\lambda}{\partial t} + \nabla \cdot \big(\mathbf{F}(\mathbf{u}^\lambda) + \mathbf{G}(\mathbf{u}^\lambda, \mathbf{Q}^\lambda)\big) = \mathbf{f} \qquad \text{in} \quad K \times (0, T] \tag{4.35}$$

$$\mathbf{Q}^\lambda - \nabla \mathbf{u}^\lambda = 0 \qquad \text{in} \quad K \times (0, T] \tag{4.36}$$

$$\mathbf{u}^\lambda = \lambda(t) \qquad \text{on} \quad \partial K \times (0, T] \tag{4.37}$$

The weak formulation is obtained by integrating against the usual test space with local support. In residual form, it reads:

$$r_{\mathbf{u}K}(\mathbf{u}_h, \mathbf{Q}_h, \mathbf{v}_h; \hat{\mathbf{u}}_h) = \left(\frac{\mathbf{u}_h - \mathbf{u}_h^-}{\Delta t}, \mathbf{v}_h\right)_K - (\mathbf{F} + \mathbf{G}, \nabla \cdot \mathbf{v}_h)_K - \\ - \langle (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mathbf{v}_h \rangle_{\partial K} - (\mathbf{f}, \mathbf{v}_h)_K \tag{4.38}$$

$$r_{\mathbf{Q}K}(\mathbf{u}_h, \mathbf{Q}_h, \mathbf{E}_h; \hat{\mathbf{u}}_h) = (\mathbf{u}_h, \nabla \cdot \mathbf{E}_h)_K + (\mathbf{Q}_h, \mathbf{E}_h)_K - \langle \hat{\mathbf{u}}_h, \mathbf{E}_h \cdot \mathbf{n} \rangle_{\partial K} \tag{4.39}$$

where, as usual, $\hat{\mathbf{u}}_h$ is an approximation of λ on the boundaries of the element and

$\widehat{\mathbf{F}}+\widehat{\mathbf{G}}$ is an approximation to the fluxes on the boundaries, defined as in Equation 2.36.

It is easy to see that Equation 4.39 is linear in all the arguments ($\mathbf{u}_h$,$\mathbf{Q}_h$,$\hat{\mathbf{u}}_h$) since it simply represents a kinematic relationship for the unknowns. If a stress formulation, in which $\mathbf{Q}$ represents the viscous stresses, was used, this would not be the case. The comparison of both formulations in the case of the Stokes system can be found in [17] and shows that even though both share the same implementation advantages, the gradient formulation is better from an accuracy point of view. This has been confirmed to carry on to the incompressible Navier-Stokes equations [43] and is the approach taken here.

Now, if the original plan of applying the HDPG scheme to the residual (defined by Equation 4.38-4.39) is carried out, coupling between the different components of $\mathbf{Q}_h$ will be introduced due to the inverse Riesz mapping. An illustration of the situation can be found in Figure 4-1. In it, the inverse Riesz mapping is symbolically computed (without the conservativity constraint) for the case of a local problem with viscous effects proportional to the non-dimensional parameter ϵ. As illustrated, terms proportional to ϵ^2 couple the gradient along x and y. This would not be a problem provided ϵ is small enough, however, $\epsilon = \mathcal{O}(1)$ for resolved flows (ϵ is basically the cell Peclet number) hence the coupling might be strong.

In addition, if HDPG is applied to the residual as it is, an extra set of constraints has to be included in order to enforce a conservativity-like condition on the kinematic relation, namely, the constant mode has to belong to the test space so that the integral of $\hat{\mathbf{u}}_h$ along the boundaries equals the volume integral of $\mathbf{Q}_h$.

Since the initial goal was to stabilize the conservation law and not the equivalent first order system, the approach proposed here consists on taking advantage of the minimization statement and as before, constrain it. In this case, the constraints will provide the conservativity condition, along with the linear kinematic relations. For

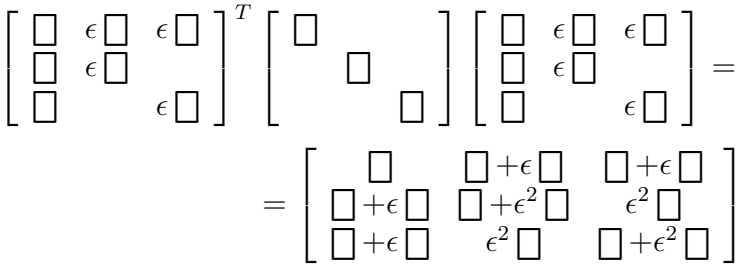


Figure 4-1: Symbolic computation of the effect of the inverse Riesz mapping on an elliptic operator when HDPG (or DPG) is used. Notice how the gradients along the x and y direction (2^{nd} and 3^{rd} row and columns) are coupled. Here ϵ represents a non-dimensional viscosity coefficient.

this, the test space for the weak formulation Equation 4.38 will be the usual polynomials of order $k + \Delta k$ while the test space for Equation 4.39 will be polynomials of order k. This way, the kinematic relations become a square solvable system in $\mathbf{Q}_h$ that will be used to define $\mathbf{Q}_h$ as a function of $\mathbf{u}_h$ and $\hat{\mathbf{u}}_h$. By including this system as a constraint, the effective minimization statement acts on $\mathbf{u}_h$ and $\hat{\mathbf{u}}_h$ and treats $\mathbf{Q}_h$ as a middle-step to compute viscous fluxes.

In order to write the problem, some notation is required. Let:

$$\mathbf{r}_{\mathbf{u}}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h) = \left\{ \begin{array}{c} \dots \\ r_{\mathbf{u}K}(\mathbf{u}_h, \mathbf{Q}_h, \phi_i; \hat{\mathbf{u}}_h) \\ \dots \end{array} \right\} \tag{4.40}$$

denote the conservation law residual vector obtained by integrating against all the basis functions ϕ_i of $\mathbf{V}_h^{k+\Delta k}$. Let also:

$$\mathbf{r}_{\mathbf{Q}}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h) = \left\{ \begin{array}{c} \dots \\ r_{\mathbf{Q}K}(\mathbf{u}_h, \mathbf{Q}_h, \phi_j; \hat{\mathbf{u}}_h) \\ \dots \end{array} \right\} \tag{4.41}$$

denote the kinematic variables residual vector, also obtained by integration against all the basis functions ϕ_j of $\mathbf{E}_h^k$. Again notice this **is not** order $k+\Delta k$, but simply or-

der k in order to render a square system. Also, notice $\mathbf{r_Q}$ is linear in all the unknowns.

All in all, the minimization statement reads:

$$(\mathbf{u}_h, \mathbf{Q}_h) = \arg \min_{(\mathbf{u}_h, \mathbf{Q}_h) \in \mathbf{V}_h^k \times \mathbf{E}_h^k} \quad \frac{1}{2} \mathbf{r_u}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h)^T X_V^{-1} \mathbf{r_u}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h) \tag{4.42}$$

$$\text{s.t.} \quad \mathbb{c}_i^T \mathbf{r_u}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h) = 0 \qquad i = 1, ..., m \tag{4.43}$$

$$\mathbf{r_Q}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h) = 0 \tag{4.44}$$

where as in the case of the hyperbolic problem, $\mathbb{c}_i$ is the vector of coefficients of the constant mode for component i and X_V is the inner product matrix for the basis of $\mathbf{V}_h^{k+\Delta k}$. The optimality conditions for this problem read:

$$\left[\frac{\partial \mathbf{r_u}}{\partial \mathbf{u}_h}\right]^T X_V^{-1} \mathbf{r_u} + \sum_{i=1}^{m} \lambda_i \mathbb{c}_i^T \left[\frac{\partial \mathbf{r_u}}{\partial \mathbf{u}_h}\right] + \mu^T \left[\frac{\partial \mathbf{r_Q}}{\partial \mathbf{u}_h}\right] = 0 \tag{4.45}$$

$$\left[\frac{\partial \mathbf{r_u}}{\partial \mathbf{Q}_h}\right]^T X_V^{-1} \mathbf{r_u} + \sum_{i=1}^{m} \lambda_i \mathbb{c}_i^T \left[\frac{\partial \mathbf{r_u}}{\partial \mathbf{Q}_h}\right] + \mu^T \left[\frac{\partial \mathbf{r_Q}}{\partial \mathbf{Q}_h}\right] = 0 \tag{4.46}$$

$$\mathbb{c}_i^T \mathbf{r_u} = 0 \qquad i = 1, ..., m \tag{4.47}$$

$$\mathbf{r_Q} = 0 \tag{4.48}$$

where $\mathbf{r_u} = \mathbf{r_u}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h)$, $\mathbf{r_Q} = \mathbf{r_Q}(\mathbf{u}_h, \mathbf{Q}_h; \hat{\mathbf{u}}_h)$ and μ is a vector of Lagrange multipliers for the kinematic constraints.

The solution of this system can be computed through an iterative scheme identical to the one described in the hyperbolic case (Algorithm 4.1); using first a Constrained Gauss-Newton (§4.1.3) approach and, once close enough to the solution, a Sequential Quadratic Programming (§4.1.3) iteration. The results indicate that this approach works, however, it involves too many degrees of freedom since the Lagrange multipliers for the kinematic relations have to be computed too. For an m component system in d space dimensions with polynomial order k, HDG requires $m \times (d+1) \times f(k)$

unknowns while HDPG needs $m \times (2d+1) \times f(k) + m$ (see Table 4.1 for a detailed breakdown).

Table 4.1: Comparison of the local degrees of freedom between HDG and HDPG for each involved unknown.

Method	$\mathbf{u}_h$	$\mathbf{Q}_h$	λ	μ
HDG	$m \times f(k)$	$m \times d \times f(k)$	-	-
HDPG	$m \times f(k)$	$m \times d \times f(k)$	m	$m \times d \times f(k)$

This represents roughly 40% more unknowns but implies significantly more work to solve each iteration of the local problem. A workaround for this consists on using the linearity of $\mathbf{r_Q}$ to locally compute $\mathbf{Q}_h = \mathbf{Q}_h(\mathbf{u}_h; \hat{\mathbf{u}}_h)$;

$$\mathbf{Q}_h = \left[\frac{\partial \mathbf{r_Q}}{\partial \mathbf{Q}_h}\right]^{-1} \left(\left[\frac{\partial \mathbf{r_Q}}{\partial \mathbf{u}_h}\right] \mathbf{u}_h + \left[\frac{\partial \mathbf{r_Q}}{\partial \hat{\mathbf{u}}_h}\right] \hat{\mathbf{u}}_h \right) \tag{4.49}$$

and redefine the local solve as:

$$\mathbf{u}_h = \arg \min_{\mathbf{u}_h \in \mathbf{V}_h^k} \quad \frac{1}{2} \mathbf{r_u}(\mathbf{u}_h, \mathbf{Q}_h(\mathbf{u}_h; \hat{\mathbf{u}}_h); \hat{\mathbf{u}}_h)^T X_V^{-1} \mathbf{r_u}(\mathbf{u}_h, \mathbf{Q}_h(\mathbf{u}_h; \hat{\mathbf{u}}_h); \hat{\mathbf{u}}_h) \tag{4.50}$$

$$\text{s.t.} \quad \mathbb{c}_i^T \mathbf{r_u}(\mathbf{u}_h, \mathbf{Q}_h(\mathbf{u}_h; \hat{\mathbf{u}}_h); \hat{\mathbf{u}}_h) = 0 \qquad i = 1, ..., m \tag{4.51}$$

which presents the same number of degrees of freedom as the HDPG scheme presented in §4.1.2 for a hyperbolic system and a very similar structure that can be solved using exactly the same iterative scheme as in §4.1.3. The only difference being that now $\mathbf{Q}_h$ is an intermediate variable, hence, derivatives have to be taken using the chain rule. All in all, this boils down to a rearrangement of the equations and the evolution of the iteration is the same in both cases, as would be expected.

4.2.2 Local Problem Sensitivities

Once the problem has been solved, the sensitivities of the solution inside the elements to the degrees of freedom on the boundaries ($\partial \mathbf{u}_h / \partial \hat{\mathbf{u}}_h$ and $\partial \mathbf{Q}_h / \partial \hat{\mathbf{u}}_h$) have to

be computed. To obtain $\partial \mathbf{u}_h / \partial \hat{\mathbf{u}}_h$, the system described by Equation 4.26 is solved. As in the case of the hyperbolic problem, the matrix from the last iteration of the local solver can be re-used.

Once $\partial \mathbf{u}_h / \partial \hat{\mathbf{u}}_h$ has been computed, $\partial \mathbf{Q}_h / \partial \hat{\mathbf{u}}_h$ follows from Equation 4.49 using the chain rule:

$$\frac{\partial \mathbf{Q}_h}{\partial \hat{\mathbf{u}}_h} = \frac{\partial \mathbf{Q}_h}{\partial \hat{\mathbf{u}}_h} + \frac{\partial \mathbf{Q}_h}{\partial \mathbf{u}_h} \frac{\partial \mathbf{u}_h}{\partial \hat{\mathbf{u}}_h} \tag{4.52}$$

4.2.3 Global Problem

For elliptic problems, the same steps are followed as in the hyperbolic case. Namely, the goal is to solve the problem; find $\hat{\mathbf{u}}_h \in \mathbf{M}_h^k$ such that,

$$\langle (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h \setminus \partial \Omega} + \langle \mathbf{b}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h) - \mathbf{g}, \mu \rangle_{\partial \Omega} = 0 \qquad \forall \mu \in \mathbf{M}_h^k \tag{4.53}$$

where

$$\widehat{\mathbf{F}} = \mathbf{F}(\hat{\mathbf{u}}_h) + \mathbf{S}(\hat{\mathbf{u}}_h)(\mathbf{u}_h - (\hat{\mathbf{u}}_h)) \cdot \mathbf{n} \tag{4.54}$$

$$\widehat{\mathbf{G}} = \mathbf{G}(\hat{\mathbf{u}}_h, \mathbf{Q}_h) + \mathbf{S}_v(\mathbf{u}_h - (\hat{\mathbf{u}}_h)) \cdot \mathbf{n} \tag{4.55}$$

As usual, Newton's iteration will be applied to solve it using the local problem as an exact relationship for $\mathbf{u}_h$ and $\mathbf{Q}_h$ and their sensitivities as a function of $\hat{\mathbf{u}}_h$. Namely, at each iteration:

$$\left(\frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \mathbf{u}_h} \frac{\partial \mathbf{u}_h}{\partial \hat{\mathbf{u}}_h} + \frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \mathbf{Q}_h} \frac{\partial \mathbf{Q}_h}{\partial \hat{\mathbf{u}}_h} + \frac{\partial \mathbf{r}_{\hat{\mathbf{u}}}}{\partial \hat{\mathbf{u}}_h} \right) \delta \hat{\mathbf{u}}_h = -\mathbf{r}_{\hat{\mathbf{u}}} \tag{4.56}$$

where $\mathbf{r}_{\hat{\mathbf{u}}}$ is defined, as usual, by testing against all the basis functions ϕ_i of $\mathbf{M}_h^k$

$$\mathbf{r}_{\hat{\mathbf{u}}}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h) = \left\{ \begin{array}{c} \cdots \\ \langle (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n}, \phi_i \rangle_{\partial \mathcal{T}_h \setminus \partial \Omega} + \langle \mathbf{b}(\mathbf{u}_h, \mathbf{Q}_h, \hat{\mathbf{u}}_h) - \mathbf{g}, \phi_i \rangle_{\partial \Omega} \\ \cdots \end{array} \right\} \tag{4.57}$$

The whole scheme follows the same flow-diagram as Algorithm 4.1 but taking into account the update in $\mathbf{Q}_h$ at each iteration of the local solver and the dependency of the different residuals on it.

4.3 HDPG Single Element Results

The rest of this chapter will deal with some examples to show how stability is enhanced in the non-linear case when discontinuities arise. For this, it is enough to compare how the HDG and HDPG schemes behave in the case of a single element with fixed $\hat{\mathbf{u}}_h$ on the boundary. The prescribed value of $\hat{\mathbf{u}}_h$ is extracted from analytical solutions that present discontinuities both in 1D and 2D.

4.3.1 Burgers Equation in 1D

The first test will be carried out using Burgers equation in 1D on a single element with boundary conditions such that a steady shock is a feasible solution to the problem. For that, the boundary conditions have to correspond to compressive data (left end solution has to be greater than right end solution) and equal in absolute value at both ends (so that the Rankine-Hugoniot condition yields zero propagation speed [37]). A case compatible with this requirement reads:

$$\frac{\partial u^2/2}{\partial x} = 0 \quad \text{in} \quad x \in (0,1) \tag{4.58}$$

$$u = 1 \quad \text{at} \quad x = 0 \tag{4.59}$$

$$u = -1 \quad \text{at} \quad x = 1 \tag{4.60}$$

In order to apply HDG and HDPG to this problem, the numerical fluxes and boundary conditions described in §A.3 have been used. A sample solution for polynomial order $k = 5$ is shown in Figure 4-2. As expected, HDPG reduces the oscillations in the solution with an enriched space only two orders higher ($\Delta k = 2$). If higher Δk is used, the only change in the solution occurs at both ends (getting slightly closer to

the exact one) while the interior solution barely changes. This indicates that $\Delta k = 2$ is enough to represent the optimal test space for this problem.

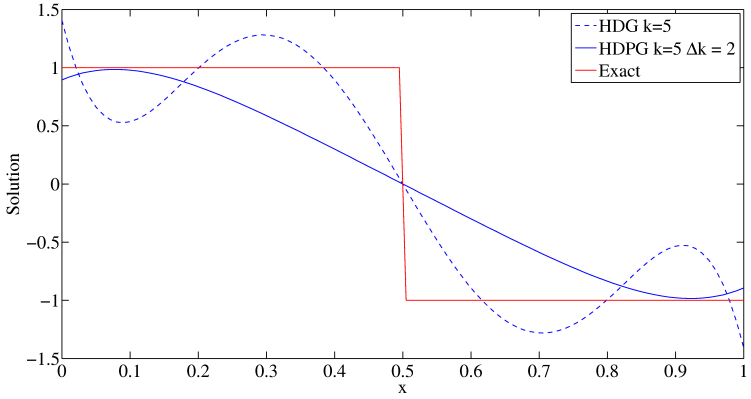
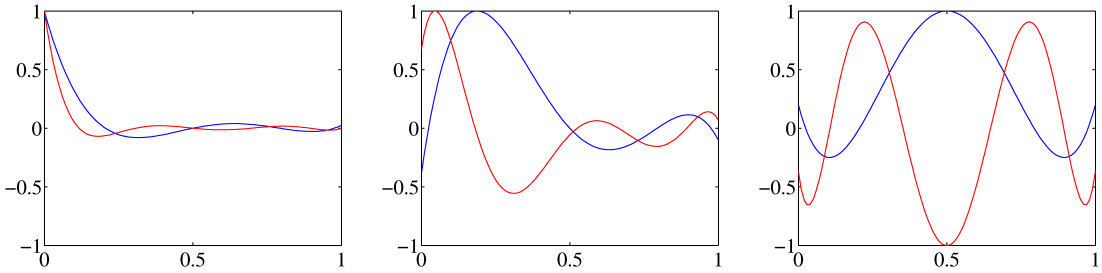


Figure 4-2: Comparison of HDG and HDPG for the case of Burgers equation in 1D using a single element and boundary data compatible with a steady shock. The polynomial order is $k = 5$ and the enriched space for HDPG is computed using $\Delta k = 2$. The result indicates that HDPG is less oscillatory in this instance.

In order to shed some light on how HDPG works, the trial functions and the associated optimal test functions can be compared in the 1D case in which the simple geometry helps the understanding. Figure 4-3 shows each Lagrange polynomial $l_j(x)$ of order $k = 4$ (computed using Chebysev nodes [30]) together with its associated optimal test function in an enriched space with $\Delta k = 2$.



(a) $l_1(x)$ (b) $l_2(x)$ (c) $l_3(x)$

Figure 4-3: Lagrange polynomials $l_j(x)$ (blue) and associated optimal test function (red) when $k = 4$ and $\Delta k = 2$. The polynomials $l_4(x)$ and $l_5(x)$ are symmetric to $l_2(x)$ and $l_1(x)$ respectively and hence have been omitted. Notice the upwinding effect introduced in the test space when HDPG is used.

It is easy to argue that the optimal test space is upwinded since the trial functions are somehow displaced opposite to the propagation velocity (in this problem $\partial F/\partial u =$

u); that is, away from the discontinuity. Indeed, this intuitive conclusion is somehow linked to the action of the inverse Riesz mapping ($[\partial \mathbf{r}/\partial u]^T X_V^{-1}$) that just represents the adjoint operator ($[\partial \mathbf{r}/\partial u]^T$) preconditioned by X_V^{-1}.

4.3.2 Euler Equations in 2D

The next step after the simple 1D case is to test HDPG in a more complicated 2D setting such as the Euler equations to further confirm the enhanced stability. For that, a solution presenting a shock will be computed using the adequate $\hat{\mathbf{u}}_h$ as boundary data. To this end, compressible flow theory [38] can be used to calculate simple straight shocks inside the element that can be later transferred to the boundaries. Since $\hat{\mathbf{u}}_h$ is assumed to belong to a certain polynomial space, some projection has to be carried out to initialize its value. For that, simple collocation is used, that may generate oscillations if the polynomial order is too high because the shock cuts the boundaries and hence the trace of the solution is not continuous on them. To minimize this effect, the comparison between HDG and HDPG will be limited to polynomial order $k = 3$.

Oblique Shock

The first test case in 2D will be the supersonic flow across an oblique shock, that is equivalent (up to viscous effect) to the supersonic flow over a small angled wedge. In this case, the inflow is set to $M_1 = 2$ and the wedge angle is set to $20°$. That way, an attached oblique shock appears $34°$ away from the wall that sets the Mach number behind it to $M_2 = 1.18$; the corresponding states are described in Table 4.2. Figure 4-4 represents the solution for $k = 3$ using both HDG and HDPG (with $\Delta k = 2$) for the Mach number, the pressure and the entropy. Both cases were initialized with the same uniform $M = 2$ flow solution and converged in less than 6 iterations.

The difference between both schemes is noticeable. As expected, the HDPG solution is more stable and hence less oscillatory. Indeed, if the polynomial order is

increased to $k = 4$, HDG does not converge while HDPG works up to $k = 7$ (before having problems with the basis conditioning). However, at that point, the interpolation of $\hat{\mathbf{u}}_h$ at the boundaries might introduce non-physical oscillations and the validity of the test itself is questionable.

Normal Shock

So far, HDPG has proved to be less oscillatory and more stable than HDG in the two previous single element examples. The purpose of this last one is to further confirm this by looking at a more complicated case in which there is a mixture of up-running and down-running waves in some parts of the element (equivalent to subsonic flow), together with a strong shock. The problem to solve will be the case of a normal shock with $M_1 = 2$ that leaves subsonic flow behind at $M_2 = 0.58$; the corresponding states on each side of the shock are described in Table 4.3 and were obtained using the Rankine-Hugoniot conditions [38]. The results using HDPG with $k = 3$ and $\Delta k = 2$ are plotted in Figure 4-5. Unlike the supersonic wedge case, HDG **did not** converge for this example even when the initial solution was interpolated from the exact solution.

4.4 Comments

All the previous results clearly indicate that the HDPG local solver is substantially more stable than HDG in cases where shocks appear and there are different propagation directions inside the domain. Furthermore, this is achieved without any sort of artificial viscosity and just relying on the enriched test space to approximate the optimal test functions and introduce the required stabilization.

The choice of the enriched space $k + \Delta k$ is, in general, a problem dependent rule; theory says the higher Δk the better because the discrete optimal test space will be close to the real one, however, in all the cases shown here, $\Delta k = 2$ seemed to be enough to get significant improvement with respect to HDG. All the cases did work

with $\Delta k = 1$, but the results where not as outstanding and hence were not reproduced here. All in all, $\Delta k = 2$ seems to be a good candidate to start from, at least in the transonic to moderate supersonic regime.

It is worth mentioning that no artificial viscosity was used in any of these examples. Similar cases have been run using the Navier-Stokes equations and the approach described in §4.2 to deal with the elliptic terms, and for them, provided the viscosity is high enough, both HDG and HDPG work. The upper limit in the viscosity is associated to the so-called cell Peclet number $Pe = \frac{h}{k}\frac{u}{\nu}$, that relates the resolution of the scheme $(\frac{h}{k})$ to the scale of the problem $(\frac{u}{\nu})$. For general DG schemes, it is well known that $Pe = \mathcal{O}(1)$ is required to capture the shock within an element. In the case of HDPG, some preliminary results (shown in the next chapter) indicate that $Pe = \mathcal{O}(10)$ is enough to capture the shock smoothly, however, this limit is still to be explored in detail.

Table 4.2: States before and after the oblique shock single element case.

	State 1	State 2
M	2	1.18
ρ	1	2.04
ρu	0.94	1.44
ρv	-0.34	0
ρE	0.95	2.32

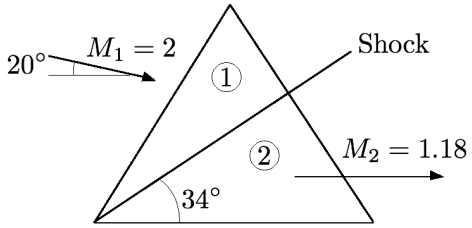
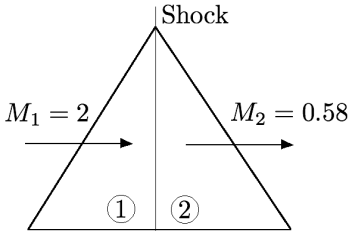
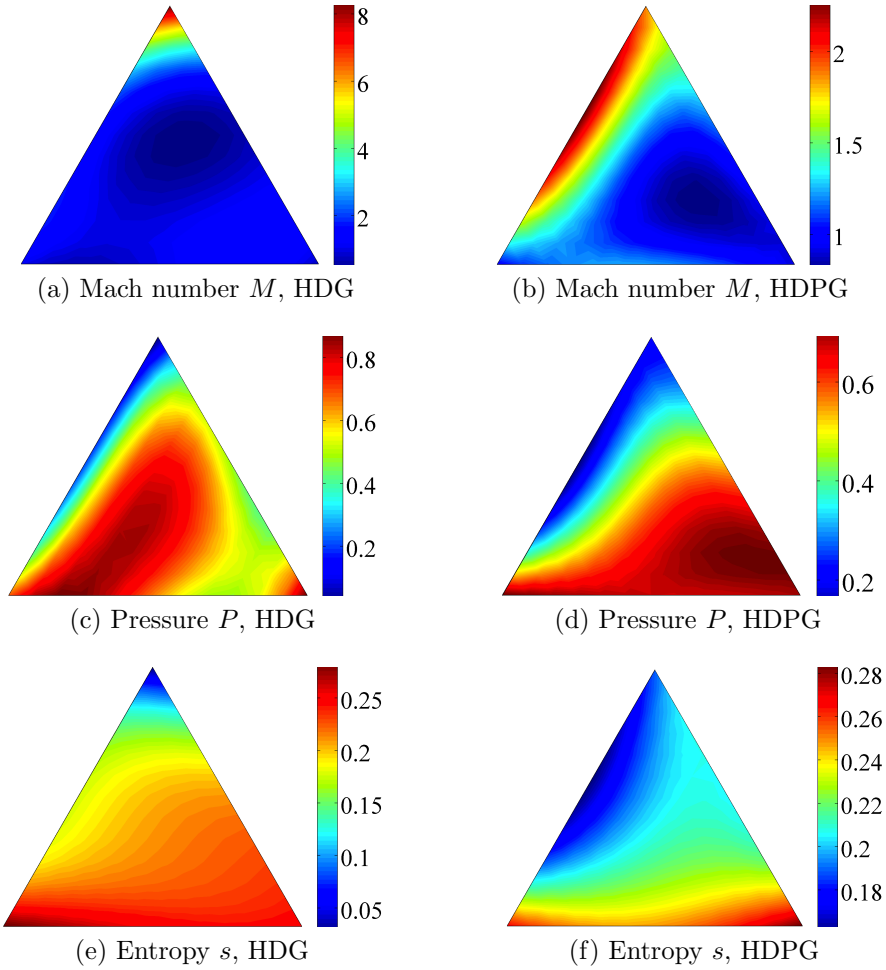


Table 4.3: States before and after the normal shock single element case.

	State 1	State 2
M	2	0.58
ρ	1	4.5
ρu	1	1
ρv	0	0
ρE	0.95	2.20



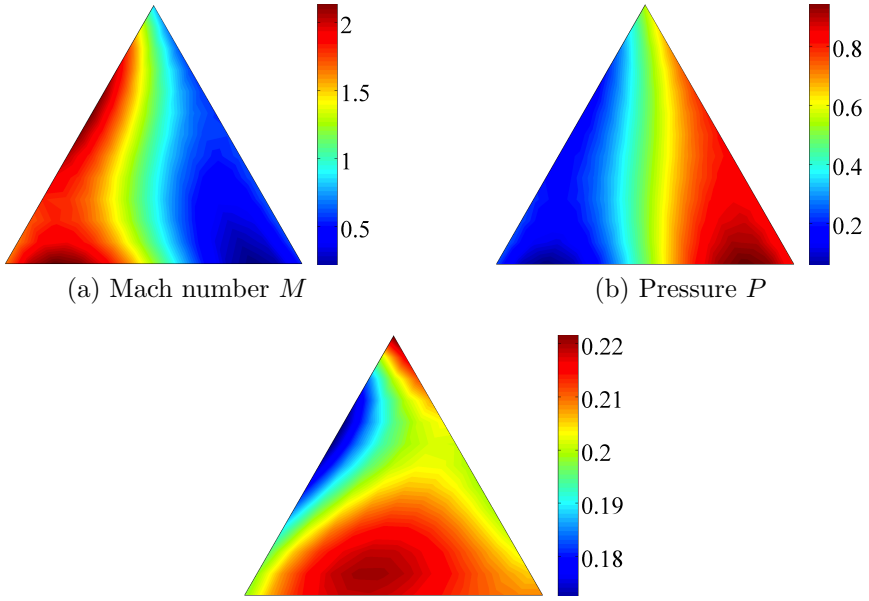


(a) Mach number M, HDG (b) Mach number M, HDPG

(c) Pressure P, HDG (d) Pressure P, HDPG

(e) Entropy s, HDG (f) Entropy s, HDPG

Figure 4-4: Results for the Euler equations on a single element with boundary data ($\hat{\mathbf{u}}_h$) extracted from an oblique shock condition using HDG (left) and HDPG (right). In this case $k = 3$ and $\Delta k = 2$. HDPG delivers a reasonable solution while HDG is close to divergence.



(a) Mach number M (b) Pressure P

(c) Entropy s

Figure 4-5: Results for the Euler equations on a single element with boundary data ($\hat{\mathbf{u}}_h$) extracted from a normal shock condition using HDPG. In this case $k = 3$ and $\Delta k = 2$. No solution for HDG was included since convergence was not achieved.

Chapter 5

Results

In the previous chapter together with the definition of the HDPG scheme, some preliminary, single element results were included to point out the enhanced stability achieved by the method. The objective in this chapter is to apply this method to more realistic geometries and confirm that HDPG is more robust than HDG in the presence of discontinuities. To that end, the same model equations will be used (Burgers, Euler, Navier-Stokes) but, this time, using a multi-element mesh, where the solution at the edges is not prescribed but computed using the global problem introduced in §4.1.5.

The structure of this chapter is as follows. First, 1D cases will be discussed for the linear convection-diffusion equation and Burgers equation both with and without discontinuity propagation. Next, Peterson's example in 2D will be used to asses the convergence properties of the method. Finally, some non-linear 2D cases will be presented to test the method on solutions with shocks.

5.1 1D Results

The purpose of the 1D test cases is to confirm that the HDPG scheme is well posed in general multi-element cases, when the global problem is used to solve for the coupling between elements.

5.1.1 Linear Convection

The first problem to discuss will be the propagation of discontinuities in the linear setting. For that, the unsteady linear convection equation in 1D, described by:

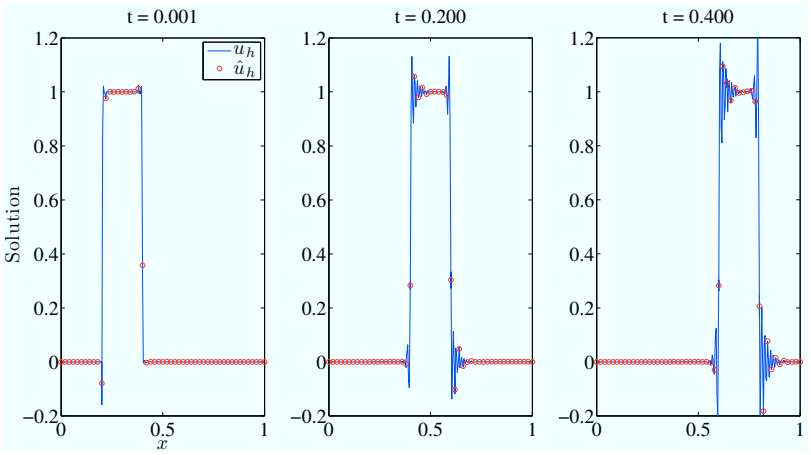
$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \quad \text{in} \quad \Omega = (0, 1) \tag{5.1}$$

is used, where u is a certain scalar quantity convected in time along the x axis with unit velocity. The initial condition in this case has been chosen as a hat function:

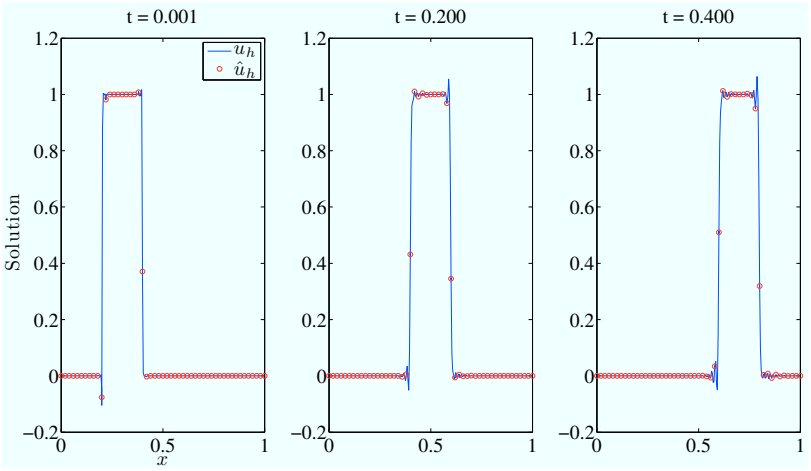
$$u(x, t = 0) = H(0.2) - H(0.4) \tag{5.2}$$

where H represents the Heaviside step function. The boundary condition for this case is set to homogeneous Dirichlet at both ends of the domain and the time integration is carried out using a simple backwards Euler scheme with time step $\Delta t = 10^{-3}$. Spatial discretization is carried out using 50 elements and polynomial order $k = 5$.

The results obtained with both HDG and HDPG are included in Figure 5-1. Notice the initial condition is challenging enough to generate small wiggles already in the first time step on both schemes (see leftmost frame of Figure 5-1), however, the evolution of such oscillations is completely different. In the HDG case, these are significantly amplified, while in the HDPG case they barely grow.



(a) HDG, $k = 5$



(b) HDPG, $k = 5$, $\Delta k = 5$

Figure 5-1: Results obtained with HDG (top) and HDPG (bottom) for the case of linear convection with discontinuous initial conditions consisting of a hat function on a mesh of 50 elements. Time integration was carried out using a Backward Euler formula with $\Delta t = 10^{-3}$. HDPG oscillation is noticeably smaller than HDG.

5.1.2 Burgers 1D: Steady Shock

The next case to try is the unsteady Burgers equation in 1D described by:

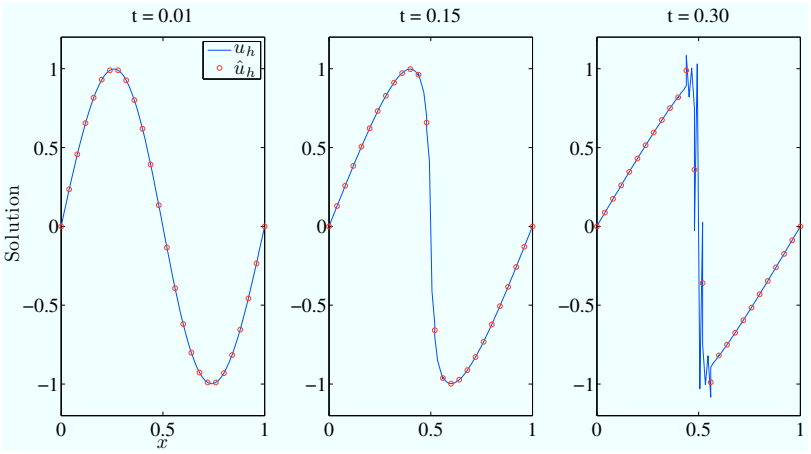
$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0 \quad \text{in} \quad \Omega = (0,1) \tag{5.3}$$

that represents a non-linear conservation law that can develop discontinuities in finite time even if the initial condition is smooth. In this case, the initial condition is set to be a sinusoidal profile

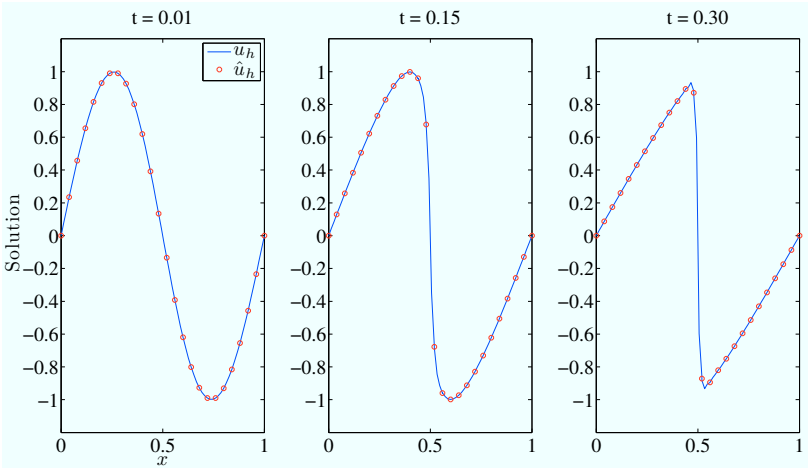
$$u(x, t = 0) = \sin(2\pi x) \tag{5.4}$$

and the boundary conditions are homogeneous Dirichlet at both ends of the domain. This set of conditions yields a solution that initially steepens into a steady shock due to the initial condition and dies off in time because of the boundary condition influence. The time integration is carried out using a BDF3 scheme (that is third order accurate in time) with $\Delta t = 10^{-2}$ while the spatial discretization consists on 25 elements and polynomials of order $k = 3$.

The results computed using both HDG and HDPG are included in Figure 5-2 and show the benefit of using HDPG. While the solution is smooth (two leftmost frames), the solution that HDG and HDPG deliver is basically the same, however, once the shock forms (rightmost frame) the difference is manifest. While the solution computed using HDPG **does not** oscillate at the shock, its HDG homologue produces strong oscillations in the element at the shock and its neighbors. This behavior would ultimately prevent convergence if a higher polynomial order (say $k = 5$) was used.



(a) HDG, $k = 3$



(b) HDPG, $k = 3$, $\Delta k = 4$

Figure 5-2: Results obtained using HDG and HDPG for the Burgers equation with initial conditions that develop a steady shock. Space discretization consisted on 25 elements. Time discretization was carried out using a BDF3 scheme with $\Delta t = 10^{-2}$. The HDPG solution captures the shock within an element.

5.1.3 Burgers 1D: Shock Propagation

The last case in the 1D setting will be the viscous Burgers equation:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = \epsilon \frac{\partial^2 u}{\partial x^2} \quad \text{in} \quad \Omega = (0,1) \tag{5.5}$$

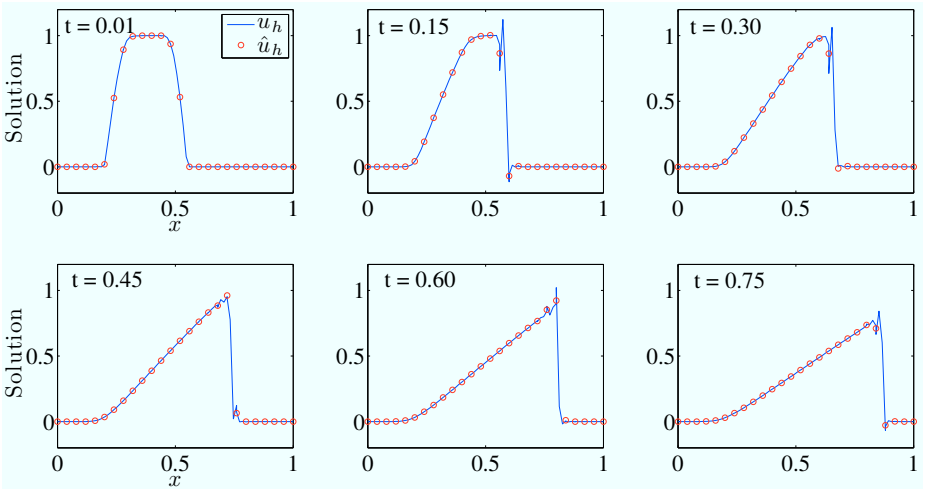
that represents the same conservation law as the previous case augmented to consider a dissipation mechanism (through the elliptic operator on the right hand side). The initial condition for this case is a smoothed hat function and the boundary conditions are homogeneous Dirichlet. The inviscid solution consists on a shock wave that forms at the right edge of the hat function and travels to the right, eventually merging with a rarefaction wave that is formed at the left edge of the hat function. When viscosity is added (through the coefficient ϵ), the solution follows the same pattern, however, discontinuities are spread over a length $l = \mathcal{O}(\epsilon/|u|)$; for these cases ϵ is set so that the cell Peclet number defined as

$$Pe|_{\text{cell}} = \frac{h}{k}\frac{|u|}{\epsilon} \tag{5.6}$$

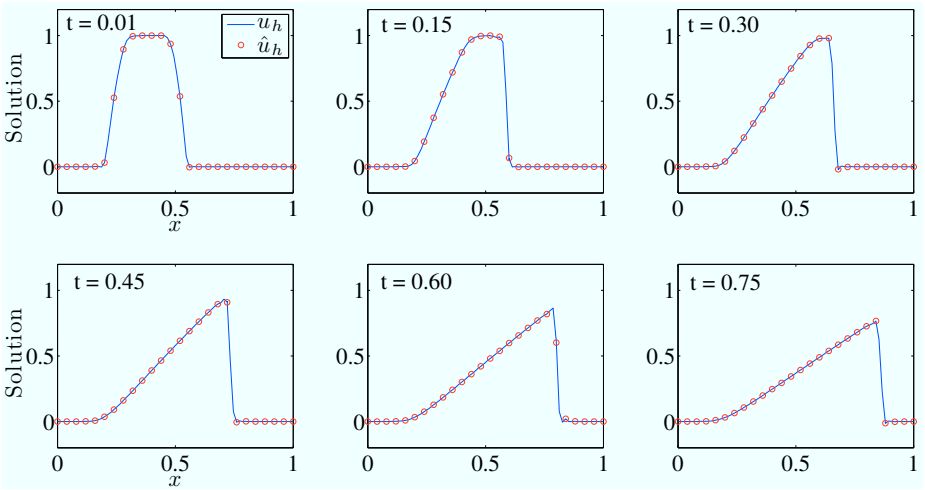
has a prescribed value $Pe|_{\text{cell}} = 10$. The time integration is carried out using a BDF3 scheme with $\Delta t = 10^{-2}$ that is third order accurate in time, while the spatial discretization consists on 25 elements and polynomials of order $k = 3$.

The results in Figure 5-3 show the solution obtained using both HDG and HDPG schemes. First of all, notice how both schemes propagate the shock at the same speed; given HDG is conservative, this implies the conservativity condition on HDPG is properly enforced. Secondly, notice how the prescribed viscosity is capable of capturing the shock within one element with HDPG but not with HDG. The reason for this has to do with the $Pe|_{\text{cell}}$ used; this non-dimensional parameter measures the ratio of discretization resolution (h/k) to viscous length scale ($\epsilon/|u|$) and has to satisfy $Pe|_{\text{cell}} = \mathcal{O}(1)$ for general Finite Element formulations to be non-oscillatory. The result here indicates that HDPG requires one order of magnitude less viscosity,

at least in this 1D setting.



(a) HDG, $k = 3$, $Pe|_{\text{cell}} = 10$



(b) HDPG, $k = 3$, $\Delta k = 4$, $Pe|_{\text{cell}} = 10$

Figure 5-3: Results obtained using HDG and HDPG for the unsteady Burgers equation with initial conditions that develop a propagating shock. Space discretization consisted on 25 elements. Time discretization was carried out using a BDF3 scheme with $\Delta t = 10^{-2}$. Viscosity was set so that $Pe|_{\text{cell}} = 10$. The HDPG scheme propagates the shock at the right speed and produces less oscillation even in this under-resolved setting.

5.2 2D Results

Even though the 1D results just described show how the new HDPG scheme delivers more stable solutions in the presence of discontinuities and under-resolution than HDG, they have limited validity as to draw any final conclusion about which method is better in practical compressible flows. To shed some light into this, some 2D results are presented in this section.

5.2.1 Linear Convection

The first case to consider is the linear convection example proposed by Peterson [49] in order to show the sub-optimal convergence of DG methods for pure convective operators. The governing equation for this problem is:

$$\frac{\partial u}{\partial y} = 0 \quad \text{in} \quad \Omega = (0,1) \times (0,1) \tag{5.7}$$

which implies that the solution is constant along y-lines. The boundary conditions are Dirichlet everywhere except at the outflow ($y = 1$) and prescribed by the function $u_0(x)$.

$$u(x,0) = u_0(x) \qquad u(0,y) = u_0(0) \qquad u(0,y) = u_0(1) \tag{5.8}$$

Peterson's strategy consisted on taking advantage of the error that appears on the edges of the triangulation that are parallel to the characteristic lines. For linear elements, this error is of order $\mathcal{O}(h^{1.5})$ point-wise in a layer of order $\mathcal{O}(h^{0.75})$ along the edges. What Peterson proposed was a sequence of meshes that would accumulate the errors and hence make the method loose half an order of convergence $(k + 1/2)$ in the L^2 norm. A detailed description of the construction of these meshes can be found in [49] and a sample one is included in Figure 5-4.

This problem cannot assess how HDPG will behave in compressible flows because it is linear, however, the convergence rate in this case is strongly related to the sta-

bility of the solution and hence it is interesting to explore. Indeed, the DPG scheme of Demkowicz et al. is claimed to be optimal due to the "enhanced stability" introduced by the optimal test functions [21, 22], and, given HDPG is based on DPG, it is worth check if optimal convergence is also achieved. For that, two different boundary conditions will be used: the first one sets $u_0(x) = x^2$ and was the one described by Peterson in the original work [49]. The second one sets $u_0(x) = \sin(6x)$ and was introduced by Demkowicz et al. to obtain the results shown in [21, 22].

In order to be consistent with both studies, both problems were solved using HDPG and compared against HDG, in all cases with polynomial order $k = 1$. The convergence plots are included in Figure 5-5 and the data for such plots is contained in Table 5.1. As expected, the result using HDG converges with the suboptimal order $k + 1/2$. What really stands out in these results is the nearly optimal convergence achieved by HDPG when $\Delta k \geq 3$ that indeed indicates that the HDPG scheme can also break Peterson's barrier. Below that value, either the scheme diverges ($\Delta k = 1$) or it does not show any enhanced convergence ($\Delta k = 2$). The first phenomenon still lacks a sound explanation but might be related with an incompatibility between the conservativity constraint and the min-max local problem.

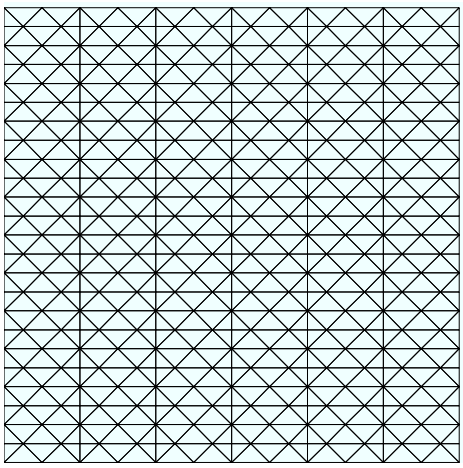


Figure 5-4: Example of Peterson's mesh used to prove suboptimal converge for DG schemes. In this case $h = 1/16$. Notice the vertical edges that are used every two horizontal edges to accumulate the error and produce the sub-optimal convergence.

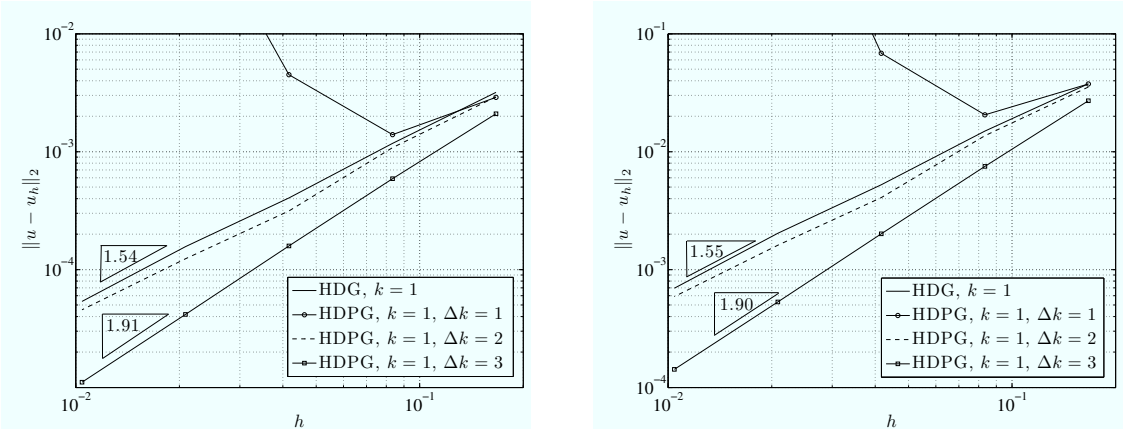
Table 5.1: Computed error and convergence rate for Peterson's example using HDG and HDPG with $k = 1$. Notice how the solution using HDPG converges with nearly optimal order when $\Delta k \geq 3$. Below such value, either the solution diverges ($\Delta k = 1$) or it yields sub-optimal convergence ($\Delta k = 2$).

(a) $u_0 = x^2$

Method	Δk	h	$\|u - u_h\|_2$	Order
HDG	–	0.167	3.19×10^{-3}	–
HDG	–	0.083	1.18×10^{-3}	1.44
HDG	–	0.042	4.04×10^{-4}	1.54
HDG	–	0.021	1.56×10^{-4}	1.37
HDG	–	0.010	5.36×10^{-5}	1.54
HDPG	1	0.167	2.89×10^{-3}	–
HDPG	1	0.083	1.40×10^{-3}	–
HDPG	1	0.042	4.49×10^{-3}	–
HDPG	1	0.021	1.68×10^{-1}	–
HDPG	1	0.010	1.95×10^{4}	–
HDPG	2	0.167	2.92×10^{-3}	–
HDPG	2	0.083	1.08×10^{-3}	1.43
HDPG	2	0.042	3.14×10^{-4}	1.78
HDPG	2	0.021	1.24×10^{-4}	1.35
HDPG	2	0.010	4.55×10^{-5}	1.44
HDPG	3	0.167	2.10×10^{-3}	–
HDPG	3	0.083	5.90×10^{-4}	1.83
HDPG	3	0.042	1.59×10^{-4}	1.90
HDPG	3	0.021	4.16×10^{-5}	1.93
HDPG	3	0.010	1.11×10^{-5}	1.91

(b) $u_0 = \sin(6x)$

Method	Δk	h	$\|u - u_h\|_2$	Order
HDG	–	0.167	3.77×10^{-2}	–
HDG	–	0.083	1.49×10^{-2}	1.33
HDG	–	0.042	5.24×10^{-3}	1.51
HDG	–	0.021	2.04×10^{-3}	1.37
HDG	–	0.010	6.95×10^{-4}	1.55
HDPG	1	0.167	3.75×10^{-2}	–
HDPG	1	0.083	2.05×10^{-2}	–
HDPG	1	0.042	6.83×10^{-2}	–
HDPG	1	0.021	5.56×10^{0}	–
HDPG	1	0.010	9.12×10^{5}	–
HDPG	2	0.167	3.53×10^{-2}	–
HDPG	2	0.083	1.37×10^{-2}	1.37
HDPG	2	0.042	4.08×10^{-3}	1.74
HDPG	2	0.021	1.61×10^{-3}	1.34
HDPG	2	0.010	5.91×10^{-4}	1.44
HDPG	3	0.167	2.71×10^{-2}	–
HDPG	3	0.083	7.51×10^{-3}	1.85
HDPG	3	0.042	2.01×10^{-3}	1.90
HDPG	3	0.021	5.32×10^{-4}	1.92
HDPG	3	0.010	1.43×10^{-4}	1.90



(a) $u_0 = x^2$ (b) $u_0 = \sin(6x)$

Figure 5-5: Converge plots for Peterson's example using HDG and HDPG with $k = 1$. While HDG delivers the expected suboptimal convergence rate (k+1/2), HDPG yields nearly optimal convergence for $\Delta k \geq 3$.

5.2.2 Burgers 2D

To continue, the HDPG scheme will be applied to Burgers equation in 2D in order to compare HDG and HDPG in a simple, single component, non-linear case where shocks may appear. The formulation of the problem is:

$$\nabla \cdot \left[\begin{Bmatrix} u^2/2 \\ u \end{Bmatrix} - \epsilon \nabla u \right] = 0 \quad \text{in} \quad \Omega = (0,1) \times (0,1) \tag{5.9}$$

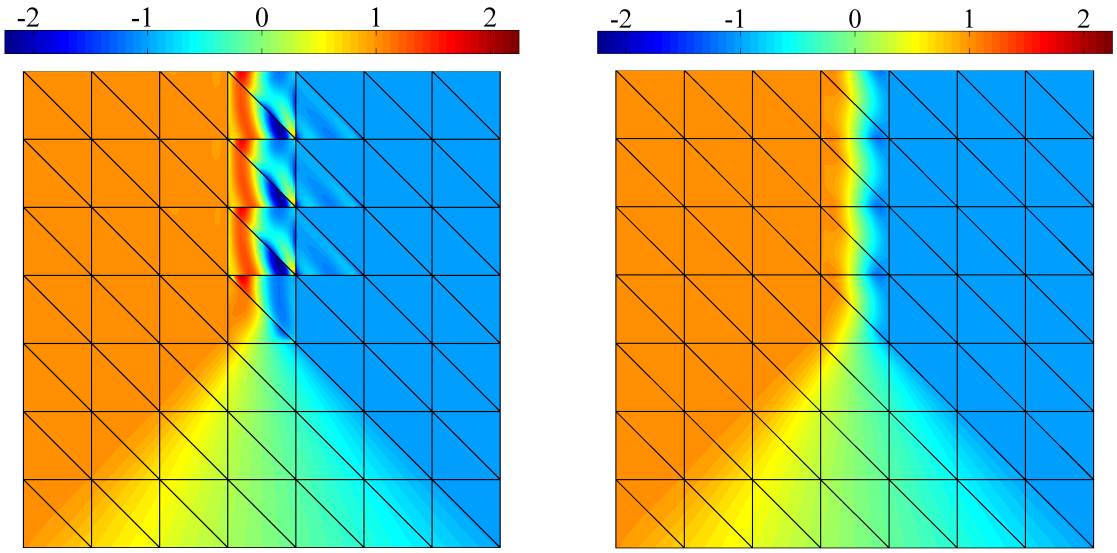
where the boundary conditions are Dirichlet everywhere except at the outflow ($y = 1$) and prescribed using the function $u_0(x)$:

$$u(x,0) = u_0(x) \qquad u(0,y) = u_0(0) \qquad u(0,y) = u_0(1) \tag{5.10}$$

In the cases presented here, $u_0(x) = 1 - 2x$ is used. The solution expected is a compression fan that eventually generates a shock in the domain parallel to the y direction.

The first set of results consists on the solution of the problem on a structured grid using high order polynomials ($k = 4$) and no viscosity ($\epsilon = 0$, $Pe|_{cell} = \infty$). The solution to this problem using both HDG and HDPG (with $\Delta k = 2$) is plotted in Figure 5-6. It is patent how the HDPG solution is significantly less oscillatory than the HDG one. Indeed, the HDG oscillations are strongly aligned with the shock, which indicates the convergence of the non-linear solver might be favored by the mesh, since the edges are bounded away from the shock direction.

To further explore this phenomenon, an unstructured mesh with roughly the same element size was used with the same polynomial order ($k = 4$) and zero viscosity ($\epsilon = 0$, $Pe|_{cell} = \infty$). In this setting, HDPG did converge while HDG diverged. In order to obtain a solution, the viscosity coefficient had to be increased so that dissipation would take care of the instability. The results in Figure 5-7 show the solution computed using HDG ($Pe|_{cell} = 10$) and HDPG (with both $Pe|_{cell} = 10$ and



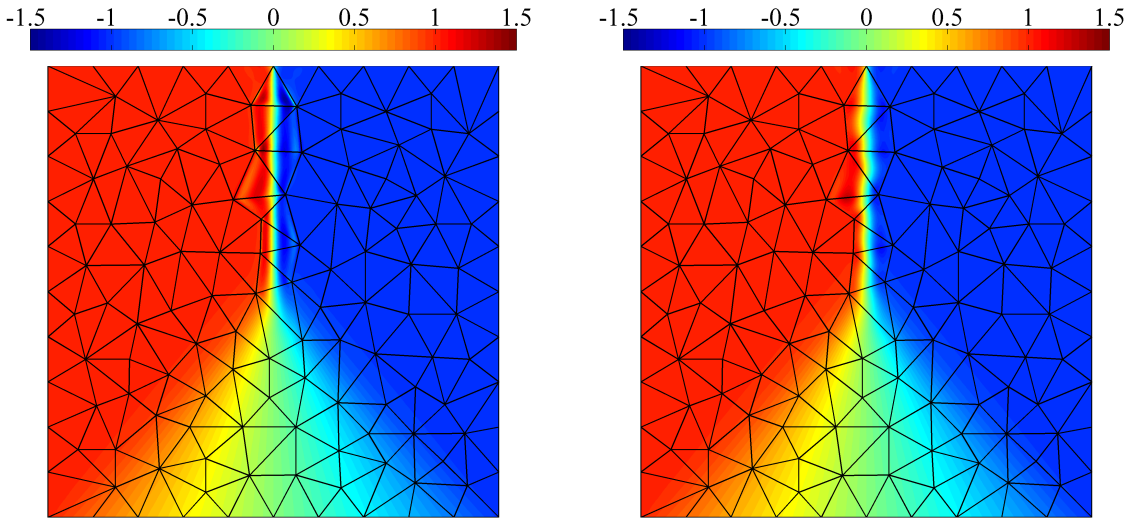
(a) HDG, $k = 4$, $Pe|_{cell} = \infty$ (b) HDPG, $k = 4$, $\Delta k = 2$, $Pe|_{cell} = \infty$

Figure 5-6: Solution to the Burgers equation in 2D using both HDG and HDPG on a structured mesh. Notice the reduced oscillation that HDPG introduces compared to HDG at the shock location.

$Pe|_{cell} = \infty$). The viscosity was chosen so that the overshoot in the solution was roughly the same.

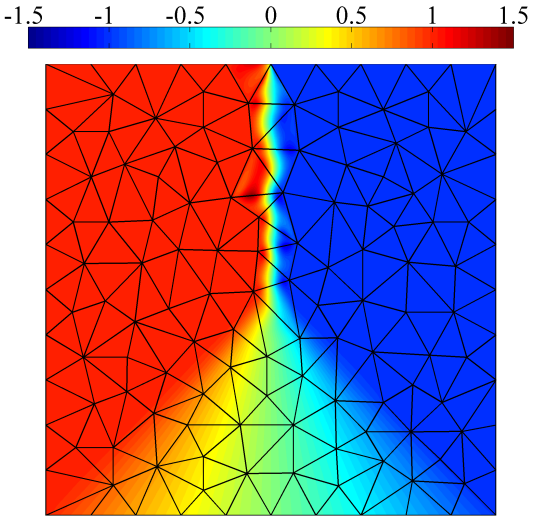
These results show that the HDG scheme with viscosity produces a straight shock line (still with some oscillation due to under-resolution) while the HDPG scheme slightly bends the shock. This is specially patent in the $Pe|_{cell} = \infty$ case and indicates that the stabilization mechanism in both cases obeys different principles. While the viscosity stabilizes in the direction of the gradient of the solution (basically the x direction, hence, the straight shock line), the optimal test functions stabilize in a global sense inside the element. In addition to this bending, HDPG seems to spread the shock in a wider layer compared to HDG.

It is clear that the HDPG scheme is more stable than HDG in under-resolved situations, not only since it is able to converge, but also because it delivers less oscillatory solutions for this problem. In order to measure this, a possible metric to



(a) HDG, $k = 4$, $Pe|_{cell} = 10$

(b) HDPG, $k = 4$, $\Delta k = 2$, $Pe|_{cell} = 10$



(c) HDPG, $k = 4$, $\Delta k = 2$, $Pe|_{cell} = \infty$

Figure 5-7: Solution to the Burgers equation in 2D using both HDG and HDPG on an unstructured mesh in order to investigate the effect of edge alignment with the shock. The HDG scheme without viscosity did not converge in this mesh.

look at would be how big the oscillation is with respect to the exact solution, that is known to be bounded by $u \in [-1, 1]$. The results obtained in the same unstructured mesh with polynomials of order $k = 4$ and different $Pe|_{cell}$ are included in Table 5.2 and show how HDPG is less oscillatory than HDG in all the cases tested. It is worth mention that, once the viscous effects are small enough, the solution basically

behaves like the $Pe|_{cell} = \infty$ case were all the stabilization comes from the optimal test functions on the local problem.

Table 5.2: Comparison of the maximum relative oscillation (%) for a Burgers 2D case between HDG and HDPG in the same unstructured mesh as Figure 5-7, using polynomials of order $k = 4$ and different $Pe|_{cell}$. The oscillation obtained with HDPG seems to level off once the viscous effects are negligible.

$Pe\|_{cell}$	HDG Oscillation (%)	HDPG Oscillation (%)
2	0	0
10	44	30
50	90	42
100	110	43
1000	–	44
∞	–	44

5.2.3 Navier-Stokes

So far, the HDPG scheme has been compared against HDG in several 1D and 2D cases in order to demonstrate its robustness and convergence properties. In this final part of the chapter, the ultimate goal of this thesis will be achieved by applying the new method to the equations of compressible flow.

The objective is then to show how HDPG behaves in different fluid flows that present shocks. It is a well known fact that accurate solutions for these cases require mesh adaptation, specially around sharp features such as shocks or boundary layers in order to resolve them, however, this will not be the focus here and fairly coarse, nearly isotropic meshes will be used to compute the results. The conclusions drawn here will just concern the convergence of the scheme as well as the stability around the shocks.

In all these cases, the problem will be modeled using the Navier-Stokes equations (see §A.6) and setting the reference viscosity coefficient μ_0 that appears in the viscosity law to be such that a certain $Pe|_{cell}$ is prescribed according to some reference

length of the mesh. Because the meshes used are coarse, this viscosity will be high everywhere in the computational domain which is equivalent to a low Reynolds number. This, combined with the fact that shocks only appear in transonic or supersonic flows where $M > 1$, makes the solution of little physical interest. In any case, regardless of the feasibility of the problem, the mathematical structure is the same and shocks still appear and trigger non-linear divergence if not stabilized properly.

Supersonic Wedge

The first example presented will be the case of a supersonic flow at $M_\infty = 2$ over a 20° wedge. The angle of the wedge is low enough for an attached oblique shock to appear that deflects the flow to be parallel to the wedge. Notice this is the same situation as the single element case described in §4.3.2. The solution using HDPG was computed without viscosity first (solving the Euler equations) and convergence was achieved, however, the results are not clean after the shock because the oscillations generated there propagate downstream. It seems necessary then to generate oscillation free shocks, and for that, some artificial viscosity is required.

A sample of the results obtained using HDPG on an unstructured grid (Figure 5-8) with polynomials of order $k = 4$, enriched test space of order $\Delta k = 2$ and viscosity such that $Pe|_{cell} = 10$, is contained in Figure 5-9 and shows the shock captured within an element with no oscillation past it. Notice how the element at the tip of the wedge shows some distortion due to the singularity present there (as the results shown in §4.3.2 for the single element), however, this is moderate and does not contaminate the rest of the solution. Notice also, the viscosity used is one order of magnitude smaller than the one required by other DG schemes.

Transonic Channel

As discussed in §4.3.2, the oblique shock case should be an easy one, despite of the shock, because the flow remains supersonic everywhere in the domain. The next ex-

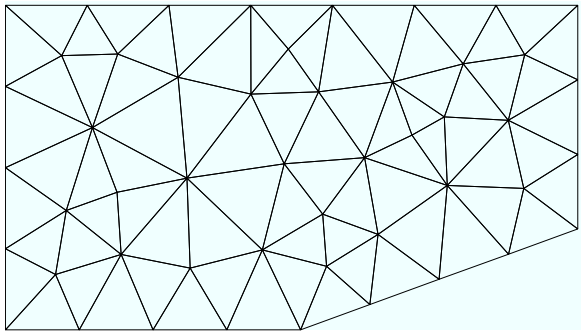
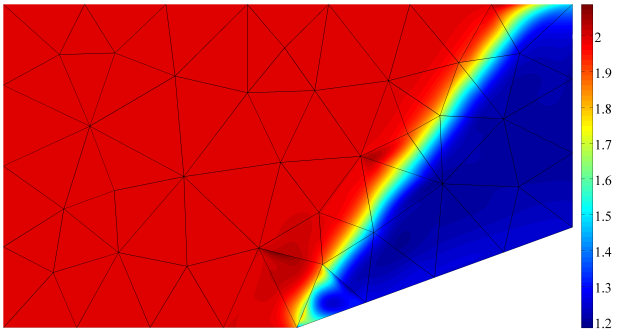
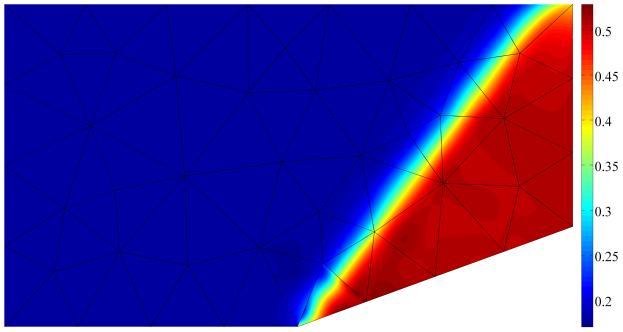


Figure 5-8: Unstructured mesh used to compute the flow over a supersonic wedge.



(a) Mach number M



(b) Pressure P

Figure 5-9: 20° wedge in a supersonic flow at $M_\infty = 2$ computed using HDPG with $k = 4$, $\Delta k = 2$ and $Pe|_{cell} = 10$. The solution is clean of oscillations and the shock is captured within an element.

ample will pose a harder challenge and consists on a transonic flow inside a channel with a small bump on the lower surface (5 % of the total height). The Mach number ($M_{\text{inlet}} = 0.8$) is high enough for a supersonic region to appear over the bump that ends in a normal shock to accommodate the flow to the pressure at the outlet. As in the previous case, some viscosity is required in order to yield oscillation free solutions.

The Mach number and pressure distribution obtained using HDPG on a structured grid (Figure 5-10), with polynomials of order $k = 3$, enriched space of order $\Delta p = 2$ and $Pe|_{cell} = 10$ is depicted in Figure 5-11. As can be observed, the HDPG scheme captures the normal shock within one element (see Figure 5-12 for the solution overlapped with the mesh) with little oscillation in the Mach number or the pressure. This indicates the scheme is robust to multiple propagation directions within an element. As in the previous case, this is achieved with an order of magnitude less viscosity than other DG schemes.

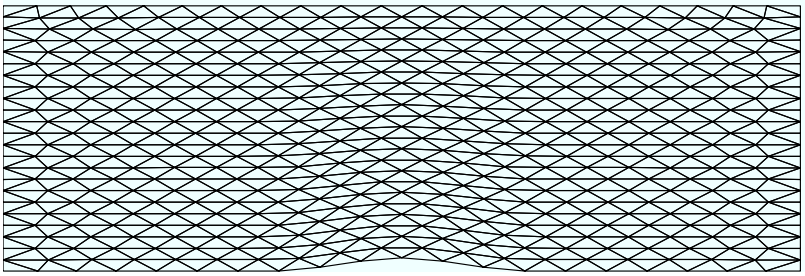
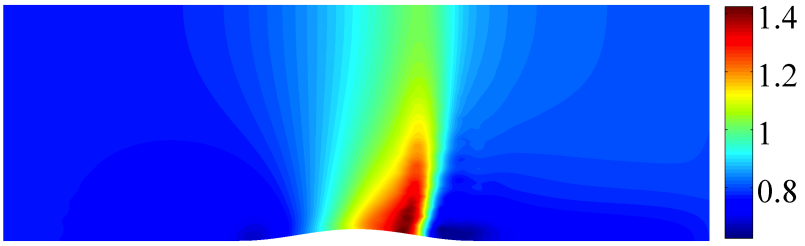
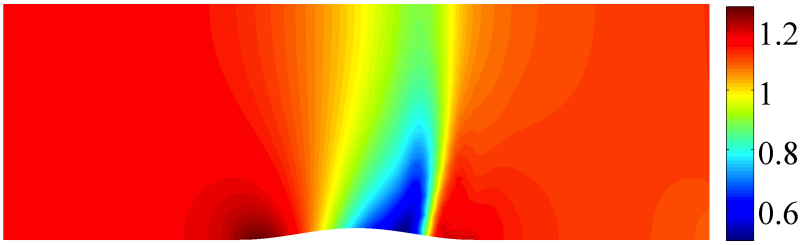


Figure 5-10: Structured mesh used to compute the solution in a transonic channel with a bump.



(a) Mach number M



(b) Pressure P

Figure 5-11: Transonic flow at $M_{\text{inlet}} = 0.8$ inside a channel with a small bump on the lower surface (5 % of height) computed using HDPG with $k = 3$, $\Delta k = 2$ and $Pe|_{cell} = 10$. The solution shows the usual supersonic region over the bump with a normal shock captured within one element.

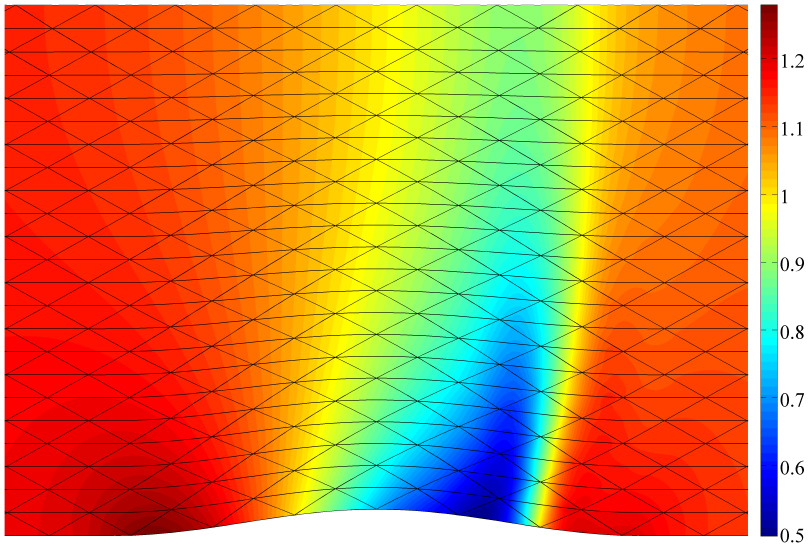


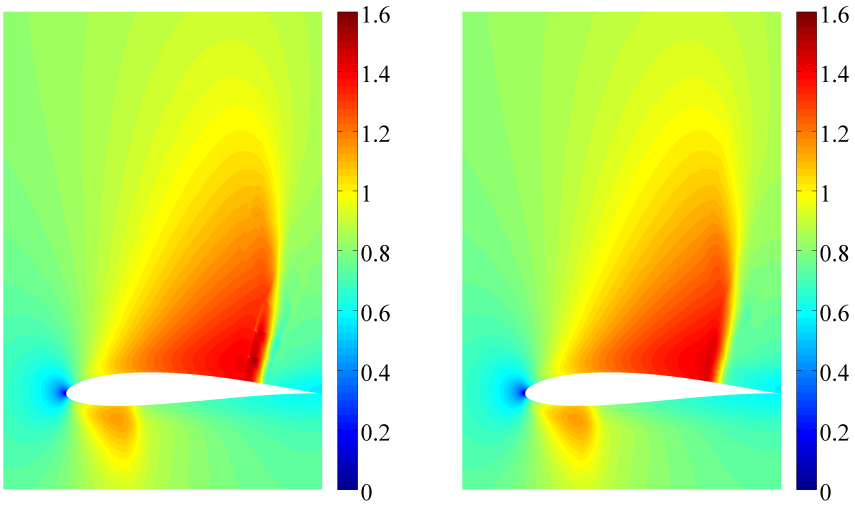
Figure 5-12: Close-up of the solution for the transonic flow inside the channel overlapped with the grid to show the shock is being captured within one element.

Trefftz Airfoil

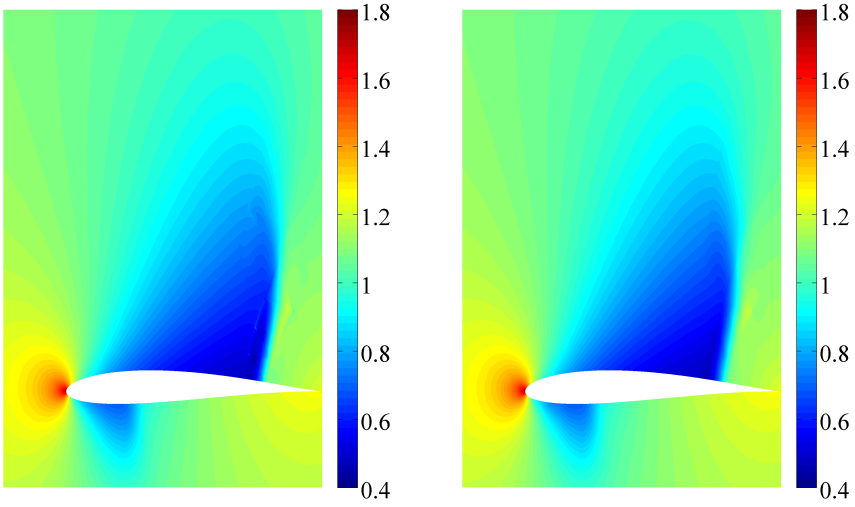
The last example will describe the transonic flow $M_\infty = 0.8$ over a Trefftz airfoil, at zero angle of attack, that is an analogue to the transonic channel in an external flow

configuration. The solution obtained using both HDG and HDPG on a structured grid (see Figure 5-15) using polynomials of order $k = 3$, enriched space $\Delta k = 2$ and $Pe|_{cell} = 10$ is plotted in Figure 5-13. The result that HDPG delivers is patently better in the sense that oscillation at the shock (that appears in HDG due to under-resolution since $Pe|_{cell} = \mathcal{O}(10)$) is not present in the HDPG solution; this is specially noticeable in the Mach number plot, (see detail on Figure 5-14).

All in all, these results seem to indicate HDPG is a feasible alternative to solve compressible flows.



(a) HDG, Mach number M (b) HDPG, Mach number M



(c) HDG, Pressure P (d) HDPG, Pressure P

Figure 5-13: Comparison between HDG and HDPG for the case of the Trefftz airfoil at zero angle of attack and $M_\infty = 0.8$. The solution was computed using polynomials of order $k = 3$, enriched test space of order $\Delta k = 2$ and $Pe|_{cell} = 10$ (based on the element size close to the airfoil). As can be appreciated, the HDPG solution is less oscillatory than HDG in this under-resolved setting.

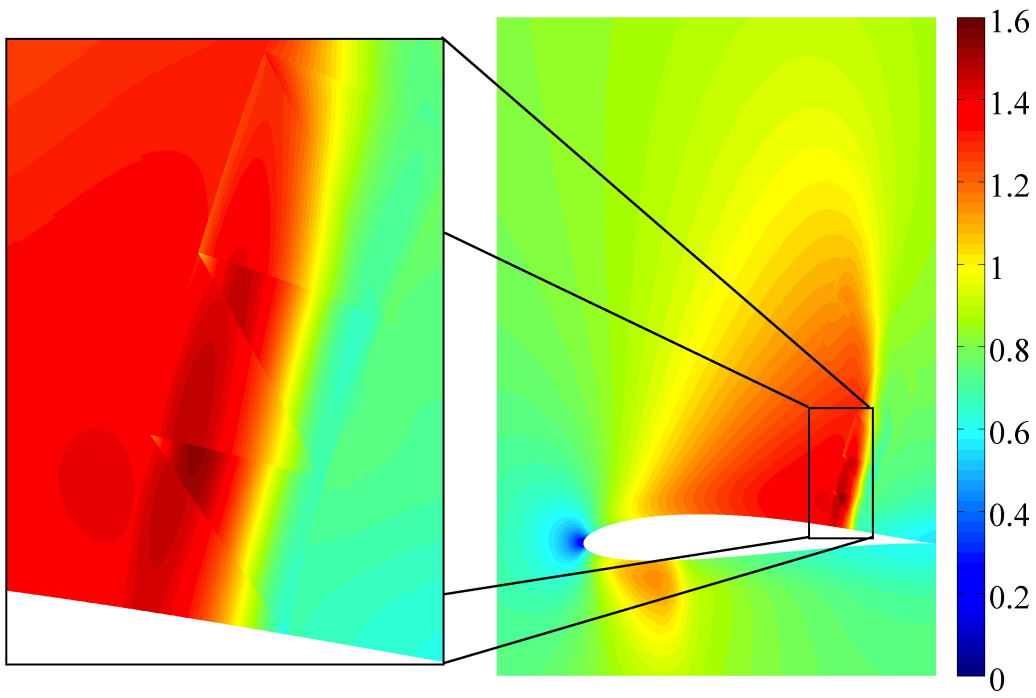


Figure 5-14: Detail of the Mach number oscillations that appear when HDG is used on the transonic flow over a Trefftz airfoil. The parameters are the same as in Fig. 5-13. The solution that HDPG delivers does not present such oscillation.

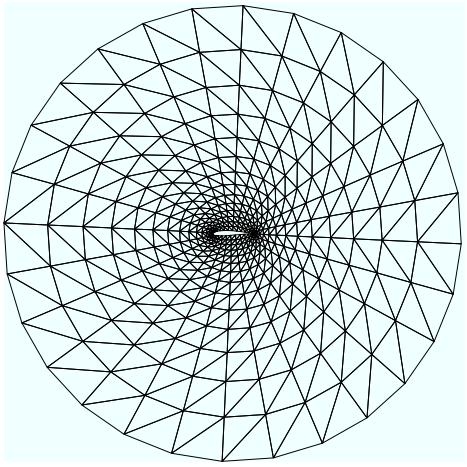


Figure 5-15: Structured mesh used to compute the transonic flow over a Trefftz airfoil.

Chapter 6

Conclusions and Future Work

6.1 Summary

In this thesis, the Hybridizable Discontinuous Petrov-Galerkin (HDPG) scheme has been devised. This scheme represents a new method to deal with hyperbolic systems of conservation laws, in particular non-linear problems that develop discontinuities or shock waves. The two main ingredients used to derive it were:

1. **The Hybridizable Discontinuous Galerkin Scheme (HDG)**; introduced in Chapter 2, that represents a domain decomposition paradigm for the solution of conservation laws.

2. **The Approximate Optimal Test Functions**; introduced in Chapter 3, that represent a methodology to compute the test functions that maximize the stability condition (or inf-sup constant) within a certain space.

In short, the HDPG scheme, introduced in Chapter 4, breaks the formulation into a local, element-wise Dirichlet problem (the local problem) and a conservativity condition at the interfaces between elements (the global problem) as the HDG scheme does, but, re-defining the local problem in order to maximize the stability of the solution. This is shown to be equivalent to a non-linear least squares minimization statement and may yield a non-conservative problem. To prevent this from happen-

ing, the local minimization statement is constrained to be conservative explicitely.

The scheme has been successfully applied to several standard problems, in the linear and non-linear setting, and compared against HDG in order to draw some conclusions (see Chapter 5).

6.2 Conclusion

The results indicate that the HDPG scheme is more stable than HDG in several instances; with particular emphasis on solution around shocks in non-linear conservation laws, where HDPG was proved more robust and less prone to non-physical oscillations. Also, in the linear case, HDPG was proved to break the sub-optimality barrier in the pathological case of Peterson's mesh.

These results also indicate that the stabilization mechanism that HDPG introduces is different from the one associated with artificial viscosity. The combination of both yields non-oscillatory solutions with artificial viscosity an order of magnitude smaller than what general DG schemes require. This allows HDPG to reproduce shock waves in under-resolved situations with bigger elements, being this a desirable feature for adaptivity purposes.

From an implementation point of view, HDPG represents a modification to HDG at the element level, hence, it does not affect the structure and number of degrees of freedom of the global problem. This modification converts the local problem into an optimization problem and hence can be further constrained to satisfy physical conditions.

6.3 Future Work

To the best of the author's knowledge, this manuscript together with [39] represent the inception of HDPG, hence, plenty of details are still missing. The following would be a list of topics that would have to be addressed in order to further understand or validate the method:

- Analysis of the well-posedness of HDPG.
- Analysis of the stabilization mechanism introduced to confirm the relationship to the adjoint operator and local upwinding.
- Benchmark against other Finite Element and Finite Volume methods in compressible flow problems.
- Exploration of the optimal convergence envelope for Peterson's example.
- Extension of the local problem to include non-negativity constraints for certain quantities such as pressure.
- Combination of HDPG to a discontinuity sensor in order to apply artificial viscosity selectively.
- ...

This is certainly not a closed list but rather what the author considers the most interesting questions still pending a formal answer/proof, some of them being crucial for the HDPG scheme to take off as a suitable algorithm to address the challenging problems found in the Aerospace engineering practice.

Appendix A

HDG Method for Different Governing Equations

In this annex the different equations used as validation tests in this work will be presented, together with the choice of the stabilization parameter for each case and the boundary conditions more frequently encountered. The notation used here obeys the one introduced in §2.3.

A.1 Convection

The convection equation represents the time evolution of a single scalar quantity u under a convective field $\mathbf{c}(\mathbf{x})$, given an initial distribution of u. Once discretized, the fluxes that enter the system are

$$\mathbf{F} = \mathbf{c}u_h \tag{A.1}$$

$$\widehat{\mathbf{F}} = \mathbf{c}\hat{u}_h + \tau(u_h - \hat{u}_h)\mathbf{n} \tag{A.2}$$

The system is hyperbolic which implies there is propagation of information along characteristic lines; these lines are driven by the convective field $\mathbf{c}(\mathbf{x})$.

Convection: Boundary Conditions

The boundary conditions can be categorized in two groups depending on whether information propagates from the boundary into the domain (inflow, $\mathbf{c} \cdot \mathbf{n} < 0$) or information leaves the domain (outflow, $\mathbf{c} \cdot \mathbf{n} > 0$). In order to impose them, a switch can be used to discern between inflow and outflow, based on the value of $\mathbf{c} \cdot \mathbf{n}$. Namely, the boundary operator will be:

$$b(\hat{u}_h, u_h) - g = (|\mathbf{c} \cdot \mathbf{n}| - \mathbf{c} \cdot \mathbf{n})(\hat{u}_h - g) + (|\mathbf{c} \cdot \mathbf{n}| + \mathbf{c} \cdot \mathbf{n})(\hat{u}_h - u_h) \tag{A.3}$$

it is easy to see that this definition imposes a Dirichlet condition $\hat{u}_h = g$ at the inflow and an extrapolation condition $\hat{u}_h = u_h$ at the outflow.

Convection: Stabilization Parameter

The stabilization parameter for the convection equation is obtained from an energy identity (see [40]) and has to satisfy:

$$\tau > \frac{1}{2}|\mathbf{c} \cdot \mathbf{n}| \tag{A.4}$$

in addition, for the system to be well posed, the convective field is required to be non-compressible ($\nabla \cdot \mathbf{c} \geq 0$).

A.2 Convection-Diffusion

The convection-diffusion equation represents the time evolution of a scalar quantity u as it is advected under the action of a field $\mathbf{c}(\mathbf{x})$ and dispersed homogeneously due

to gradients in the solution. The fluxes that define the conservation law are:

$$\mathbf{F} = \mathbf{c}u_h \tag{A.5}$$

$$\mathbf{G} = -\kappa\nabla u_h = -\kappa\mathbf{q} \tag{A.6}$$

$$\widehat{\mathbf{F}} = \mathbf{c}\hat{u}_h + \tau_c(u_h - \hat{u}_h)\mathbf{n} \tag{A.7}$$

$$\widehat{\mathbf{G}} = -\kappa\mathbf{q} + \tau_d(u_h - \hat{u}_h)\mathbf{n} \tag{A.8}$$

The system is elliptic which implies that solutions are smoother than in the pure convection case. However, the hyperbolic character can still be present in regions where the gradients are small.

Convection-Diffusion: Boundary Conditions

In this case, due to the presence of the elliptic operator, the boundary conditions can also depend on the gradient of the solution. The discrete boundary operators for the most commonly found boundary conditions are:

- **Dirichlet** This boundary condition imposes the value $u = g$ on the boundary. In HDG it is imposed on the numerical trace:

  $$b(\hat{u}_h, u_h, \mathbf{q}_h) - g = \hat{u}_h - g \tag{A.9}$$

  and is equivalent to setting $\hat{u}_h = g$ on the boundary.

- **Neumann** This boundary condition imposes the value of the fluxes normal to the boundary $(\mathbf{F} + \mathbf{G}) \cdot \mathbf{n} = g$. In HDG it is set through the numerical fluxes:

  $$b(\hat{u}_h, u_h, \mathbf{q}_h) - g = (\widehat{\mathbf{F}} + \widehat{\mathbf{G}}) \cdot \mathbf{n} - g \tag{A.10}$$

  and forces the numerical flux (which is itself an approximation to the flux at the boundaries) to match the prescribed value.

- **Extrapolation** As in the pure convection case, in certain parts of the boundary, information might be leaving the domain, hence, the solution is extrapolated:

$$b(\hat{u}_h, u_h, \mathbf{q}_h) - g = \hat{u}_h - u_h \tag{A.11}$$

hence the numerical fluxes $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{G}}$ are set to be equal to the interior ones.

Convection-Diffusion: Stabilization Parameter

The stabilization parameter used in the convection-diffusion case can be broken into a part corresponding to the convective flux (τ_c) and a part corresponding to the diffusive flux (τ_d). The former is defined in the same way as in the pure convective case while the later just has to be positive.

$$\tau_c > \frac{1}{2}|\mathbf{c} \cdot \mathbf{n}| \tag{A.12}$$

$$\tau_d > 0 \tag{A.13}$$

a usual choice for τ_d is driven by dimensional consistency and reads:

$$\tau_c = |\mathbf{c} \cdot \mathbf{n}| \tag{A.14}$$

$$\tau_d = \kappa / l \tag{A.15}$$

where l is a typical length scale of the problem. For more details see [40].

A.3 Burgers 1D

The Burgers equation in 1D represents the convection of a scalar quantity u with a velocity proportional to u itself. This is an example of a non-linear system that might develop discontinuities for certain initial and boundary conditions. As mentioned, a usual way to deal with such discontinuities is to introduce artificial viscosity, hence

an elliptic operator is also required. The fluxes that define this conservation law are:

$$\mathbf{F} = \frac{u_h^2}{2} \tag{A.16}$$

$$\mathbf{G} = -\kappa \nabla u_h = -\kappa \mathbf{q} \tag{A.17}$$

$$\widehat{\mathbf{F}} = \frac{\hat{u}_h^2}{2} + \tau_c (u_h - \hat{u}_h)\mathbf{n} \tag{A.18}$$

$$\widehat{\mathbf{G}} = -\kappa \mathbf{q} + \tau_d (u_h - \hat{u}_h)\mathbf{n} \tag{A.19}$$

In the inviscid case both $\mathbf{G}$ and $\widehat{\mathbf{G}}$ together with the definition of the kinematic variables can be discarded.

Burgers 1D: Boundary Conditions

In the general case where artificial viscosity is present, the boundary conditions can be set like in the convection-diffusion case §A.2. However, when artificial viscosity is ignored, the system is purely hyperbolic and the convection of u is driven by u itself, the boundary conditions then depend on the solution and cannot be set a priori. As in the linear convection case §A.1, a switch can be used to set the value of $\hat{u}_h$ to either a Dirichlet boundary condition or extrapolation; namely:

$$b(\hat{u}_h, u_h) - g = (|\hat{u}_h \cdot \mathbf{n}| - \hat{u}_h \cdot \mathbf{n})(\hat{u}_h - g) + (|\hat{u}_h \cdot \mathbf{n}| + \hat{u}_h \cdot \mathbf{n})(\hat{u}_h - u_h) \tag{A.20}$$

Burgers 1D: Stabilization parameter

In this particular case, the problem can be proved to be well posed if the stabilization parameter satisfies:

$$\tau_c > \frac{1}{2} \sup_{\{u_h, \hat{u}_h\}} |u| \tag{A.21}$$

$$\tau_d > 0 \tag{A.22}$$

or simply

$$\tau_c = |\hat{u}_h| \tag{A.23}$$

$$\tau_d = \kappa / l \tag{A.24}$$

As defined, the stabilization parameter is equivalent to the linear convective case with $\mathbf{c}$ equal to be the biggest absolute propagation speed between u_h and $\hat{u}_h$. For more details on how to choose τ for a general case see [41].

A.4 Burgers 2D

The previous equation can be extended to 2D by just treating the time derivative as a derivative along the y direction in a space-time FEM fashion. The fluxes for this case are then:

$$\mathbf{F} = \left(\frac{u_h^2}{2}, u_h \right) \tag{A.25}$$

$$\mathbf{G} = -\kappa \nabla u_h = -\kappa \mathbf{q} \tag{A.26}$$

$$\widehat{\mathbf{F}} = \left(\frac{\hat{u}_h^2}{2}, \hat{u}_h \right) + \tau_c (u_h - \hat{u}_h) \mathbf{n} \tag{A.27}$$

$$\widehat{\mathbf{G}} = -\kappa \mathbf{q} + \tau_d (u_h - \hat{u}_h) \mathbf{n} \tag{A.28}$$

Burgers 2D: Boundary Conditions

As in the 1D case, the boundary conditions in §A.2 are valid when there is viscosity. For the hyperbolic case, the boundary conditions are set in a similar manner:

$$\begin{aligned} b(\hat{u}_h, u_h) - g = & \left(\left| \left(\frac{\hat{u}_h^2}{2}, \hat{u}_h \right) \cdot \mathbf{n} \right| - \left(\frac{\hat{u}_h^2}{2}, \hat{u}_h \right) \cdot \mathbf{n} \right) (\hat{u}_h - g) + \\ & + \left(\left| \left(\frac{\hat{u}_h^2}{2}, \hat{u}_h \right) \cdot \mathbf{n} \right| + \left(\frac{\hat{u}_h^2}{2}, \hat{u}_h \right) \cdot \mathbf{n} \right) (\hat{u}_h - u_h) \end{aligned} \tag{A.29}$$

Burgers 2D: Stabilization Parameter

Again, following [41], the stabilization parameter can be defined by

$$\tau_c > \frac{1}{2} \sup_{\{u_h, \hat{u}_h\}} \sqrt{1+u^2} \tag{A.30}$$

$$\tau_d > 0 \tag{A.31}$$

however, in order to avoid the non-smoothness associated to the sup operation, the following convective stabilization parameter, inspired in the linearized equation, is used:

$$\tau_c = |\,(\hat{u}_h, 1) \cdot \mathbf{n}| \tag{A.32}$$

$$\tau_d = \kappa / l \tag{A.33}$$

A.5 Euler

The Euler equations represent the conservation of mass, momentum and energy in a compressible fluid flow under the assumption that viscosity and heat conduction are negligible. In conservative form, the variables and the fluxes read:

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix} \qquad \mathbf{F} = \begin{bmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho uv & \rho v^2 + p \\ \rho uH & \rho vH \end{bmatrix} \qquad \widehat{\mathbf{F}} = \mathbf{F}(\hat{\mathbf{u}}_h) + \mathbf{S}(\mathbf{u}_h - \hat{\mathbf{u}}_h)\mathbf{n} \tag{A.34}$$

$$H = E + p/\rho, \qquad E = e + \frac{1}{2}(u^2+v^2), \qquad p = (\gamma - 1)\rho e \tag{A.35}$$

where the last line contains the definition of Total Enthalpy (H) and Total Energy (E) together with the ideal gas equation of state.

Euler: Boundary Conditions

For the cases of interest in this thesis, two kinds of boundary conditions are considered: either a far-field boundary condition or a wall boundary condition. The discrete boundary operators for these are:

- **Far-field** This boundary condition is employed when the value of the solution u_∞ is known to be close a given state g. This would be equivalent to an in-flow/outflow boundary condition in the case of the linear convection equation, however, given that the problem has multiple components, the process is more elaborated. From a mathematical point of view, the system of equations can be written in quasi-linear form as:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \cdot \nabla \mathbf{u} = \mathbf{f} \tag{A.36}$$

  where $\partial \mathbf{F}/\partial \mathbf{u}$ represents the Jacobian of the Euler fluxes. By taking the normal component of the Jacobian ($\mathbf{A}_n = \partial \mathbf{F}/\partial \mathbf{u} \cdot \mathbf{n}$) the problem can be diagonalized and incoming/outcoming waves can be separated (see [25] for details on general non-reflecting boundary conditions). The diagonalization can be carried out using the auxiliary parameter vector introduced by Roe [52]. Let $\mathbf{A}_n = \mathbf{L}\Lambda\mathbf{R}$ denote such diagonal decomposition, then, in the same fashion as in the linear convection case, the boundary operator reads:

$$\begin{aligned} \mathbf{b}(\hat{\mathbf{u}}_h, \mathbf{u}_h) - \mathbf{g} = &\left(\mathbf{A}_n(\hat{\mathbf{u}}_h) + |\mathbf{A}_n|(\hat{\mathbf{u}}_h)\right)(\hat{\mathbf{u}}_h - \mathbf{u}_h) - \\ &- \left(\mathbf{A}_n(\hat{\mathbf{u}}_h) - |\mathbf{A}_n|(\hat{\mathbf{u}}_h)\right)(\hat{\mathbf{u}}_h - \mathbf{g}) \end{aligned} \tag{A.37}$$

  where $|\mathbf{A}_n| = \mathbf{L}|\Lambda|\mathbf{R}$. It is easy to see that this operator separates the solution into incoming and outcoming waves and sets the right combination of the components (through the eigenvectors) to either far-field condition $\mathbf{u}_\infty$ or extrapolation respectively. For more details about this see [46].

- **Wall** Another common boundary condition found in compressible flow calculations (either internal or external) is the flow tangency to a wall. In this case, the boundary operator reads:

$$\mathbf{b}(\hat{\mathbf{u}}_h, \mathbf{u}_h) - \mathbf{g} = \left\{ \begin{array}{c} \hat{\rho}_h - \rho_h \\ \widehat{\rho u}_h - \rho u_h + ((\rho u_h, \rho v_h) \cdot \mathbf{n}) n_x \\ \widehat{\rho v}_h - \rho v_h + ((\rho u_h, \rho v_h) \cdot \mathbf{n}) n_y \\ \widehat{\rho E}_h - \rho E_h \end{array} \right\} \tag{A.38}$$

  and implies the extrapolation of density, energy and tangential component of the momentum while enforcing a zero normal component.

Euler: Stabilization Matrix

Since the Euler equations represent a system of conservation laws, the stabilization terms are added through the matrix $\mathbf{S}$. The most common choices for this are based again on the linearization of the problem and the identification of the wave speeds. The first approach relies on using the diagonalized form described for the case of far-field boundary conditions:

$$\mathbf{S} = \mathbf{L}|\Lambda|\mathbf{R}(\hat{\mathbf{u}}_h) \tag{A.39}$$

so that each wave across the interface between elements gets properly stabilized.

A less cumbersome approach relies on using the fastest wave speed [46, 52] that corresponds to:

$$\mathbf{S} = |\lambda|_{max}(\hat{\mathbf{u}}_h)\mathbf{I} = (|\hat{\mathbf{u}}_h \cdot \mathbf{n}| + a(\hat{\mathbf{u}}_h))\mathbf{I} \tag{A.40}$$

where a represents the speed of sound.

In both cases, the stabilization matrix only depends on $\hat{\mathbf{u}}_h$, which implies the

definition of $\widehat{\mathbf{F}}$ is linear in the degrees of freedom inside the element.

A.6 Navier-Stokes

The Navier-Stokes equations are similar to the Euler equations but including both viscosity and heat conduction effects. The inviscid flux definition is the same while the viscous flux reads:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 \\ \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \\ \tau_{xx}u + \tau_{xy}v + \kappa\partial T/\partial x & \tau_{yx}u + \tau_{yy}v + \kappa\partial T/\partial y \end{bmatrix} \tag{A.41}$$

where τ represents the stress tensor, that, under Stokes assumption [5], can be written as:

$$\tau_{ij} = 2\mu\left(\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\sum_i \frac{\partial u_i}{\partial x_i}\delta_{ij}\right) \tag{A.42}$$

In order to close the system, μ is defined using Sutherland's law and κ is derived from the constant Prandtl number assumption: $Pr = \mu C_p/\kappa$. More details about the derivation of the Navier-Stokes equations can be found in [5]. Notice that in this case the viscous fluxes are still linear in the gradient ($\mathbf{Q}$) but non-linear in the solution itself ($\mathbf{u}$), since, for example:

$$\frac{\partial u}{\partial x} = \frac{\partial \rho u/\rho}{\partial x} = \frac{1}{\rho}\frac{\partial \rho u}{\partial x} - \frac{\rho u}{\rho^2}\frac{\partial \rho}{\partial x} \tag{A.43}$$

The numerical viscous fluxes are defined as usual:

$$\widehat{\mathbf{G}} = \mathbf{G}(\hat{\mathbf{u}}_h, \mathbf{Q}_h) + \mathbf{S}_v(\mathbf{u}_h - \hat{\mathbf{u}}_h)\mathbf{n} \tag{A.44}$$

Navier-Stokes: Boundary Conditions

For the Navier-Stokes system, most of the cases of interest present only one of the following three types of boundary conditions:

- **Far-field** The Navier-Stokes system, being in a sense the origin of the Euler system, shares with the later some properties. In particular, in regions of the domain where the viscous effects are negligible, the far-field boundary conditions can be imposed in the same way, hence, the boundary operator defined by Equation A.37 is still valid.

- **Wall** Unlike the Euler equations, the Navier-Stokes system requires the definition of the velocity at the wall (zero-slip condition) and either the assumption of a certain temperature at the wall $T = T_w$ or a certain heat flow $\kappa \partial T/\partial x = q_w$. All in all, the boundary operator reads:

$$\mathbf{b}(\hat{\mathbf{u}}_h, \mathbf{u}_h, \mathbf{Q}_h) - \mathbf{g} = \left\{ \begin{array}{c} \hat{\rho}_h - \rho_h \\ \widehat{\rho u}_h - \hat{\rho}_h u_{wall} \\ \widehat{\rho v}_h - \hat{\rho}_h v_{wall} \\ \widehat{T}(\hat{\mathbf{u}}_h) - T_w \end{array} \right\} \quad \text{or} \quad \left\{ \begin{array}{c} \hat{\rho}_h - \rho_h \\ \widehat{\rho u}_h - \hat{\rho}_h u_{wall} \\ \widehat{\rho v}_h - \hat{\rho}_h v_{wall} \\ \kappa \nabla T(\hat{\mathbf{u}}_h, \mathbf{Q}_h) \cdot \mathbf{n} - q_w \end{array} \right\} \tag{A.45}$$

  where, $\nabla T(\hat{\mathbf{u}}_h, \mathbf{Q}_h) = \frac{\partial T(\hat{\mathbf{u}}_h)}{\partial \hat{\mathbf{u}}_i} \mathbf{Q}_{hi}$.

- **Wall without stress** In certain cases, the Reynolds number of the flow is high enough so that viscous terms can be dropped and the Euler system represents a good physical model. If shock waves appear and artificial viscosity has to be used, thick boundary layers (caused by the artificial viscosity being several orders of magnitude bigger than the real viscosity) might appear in the walls due to the non-slip condition that may interact with the shock waves. In order to prevent this from happening, special treatment has to be given to the wall when no viscous effects are desired in the direction normal to it. The boundary

operator for this condition reads:

$$\mathbf{b}(\hat{\mathbf{u}}_h, \mathbf{u}_h, \mathbf{Q}_h) - \mathbf{g} = \left\{ \begin{array}{c} \hat{\rho}_h - \rho_h \\ \widehat{\rho u}_h n_x + \widehat{\rho v}_h n_y \\ \mathbf{t}^T \cdot \tau(\hat{\mathbf{u}}_h, \mathbf{Q}_h) \cdot \mathbf{n} \\ \kappa \nabla T(\hat{\mathbf{u}}_h, \mathbf{Q}_h) \cdot \mathbf{n} \end{array} \right\} \tag{A.46}$$

which implies extrapolation of the density, tangency of the velocity, zero tangent stress and zero normal heat conduction.

Navier-Stokes: Stabilization Matrix

The stabilization matrix that enters $\widehat{\mathbf{G}}$ is defined based on the dimensional consistency of the equations and reads:

$$\mathbf{S}_v = \begin{bmatrix} 0 & & & \\ & \frac{1}{Re} & & \\ & & \frac{1}{Re} & \\ & & & \frac{1}{(\gamma-1)M_\infty^2 RePr} \end{bmatrix} \tag{A.47}$$

where Re, M_∞ and Pr are defined as usual and $\gamma = 1.4$ for the case of interest (as in air).

Bibliography

[1] R. Alexander. Diagonally Implicit Runge-Kutta methods for stiff ODE's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

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

[3] I. Babuška. Error-bounds for Finite Element method. *Numerische Mathematik*, 16(4):322–333, 1971.

[4] F. Bassi and S. Rebay. A high-order accurate Discontinuous Finite Element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131(2):267–279, 1997.

[5] G.K. Batchelor. *An introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, UK, 2000.

[6] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, USA, 1999.

[7] C.L. Bottasso, S. Micheletti, and R. Sacco. The Discontinuous Petrov-Galerkin method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 191(31):3391–3409, 2002.

[8] C.L. Bottasso, S. Micheletti, and R. Sacco. A multiscale formulation of the Discontinuous Petrov-Galerkin method for advective-diffusive problems. *Computer Methods in Applied Mechanics and Engineering*, 194(25-26):2819–2838, 2005.

[9] F. Brezzi, M.O. Bristeau, L.P. Franca, M. Mallet, and G. Rogé. A relationship between stabilized Finite Element methods and the Galerkin method with bubble functions. *Computer Methods in Applied Mechanics and Engineering*, 96(1):117–129, 1992.

[10] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element methods*. Springer-Verlag New York, Inc., New York, USA, 1991.

[11] A.N. Brooks and T.J.R. Hughes. 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(1-3):199–259, 1982.

[12] J.C. Butcher and J. Wiley. *Numerical methods for ordinary differential equations*. Wiley Online Library, 2008.

[13] P. Causin and R. Sacco. A Discontinuous Petrov-Galerkin method with Lagrangian multipliers for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 43(1):280–302, 2006.

[14] J. Chan, L. Demkowicz, R. Moser, and N. Roberts. A class of Discontinuous Petrov-Galerkin methods. Part V: solution of 1D Burgers and Navier-Stokes equations. Technical Report 10-29, ICES, 2010.

[15] B. Cockburn. Discontinuous Galerkin methods. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 83(11):731–754, 2003.

[16] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of Discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47(2):1319–1365, 2009.

[17] B. Cockburn, N. Nguyen, and J. Peraire. A comparison of HDG Methods for Stokes flow. *Journal of Scientific Computing*, 45:215–237, 2010.

[18] B. Cockburn and C.W. Shu. The local Discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.

[19] B. Cockburn and C.W. Shu. The Runge-Kutta Discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[20] L. Demkowicz. Babuska $\Longleftrightarrow$ Brezzi. Technical Report 06-8, ICES, 2006.

[21] L. Demkowicz and J. Gopalakrishnan. A class of Discontinuous Petrov-Galerkin methods. Part I: the transport equation. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1558 – 1572, 2010.

[22] L. Demkowicz and J. Gopalakrishnan. A class of Discontinuous Petrov-Galerkin methods. Part II: optimal test functions. *Numerical Methods for Partial Differential Equations*, 27(1):70–105, 2010.

[23] L. Demkowicz and J. Gopalakrishnan. Analysis of the DPG method for the Poisson equation. Technical Report 10-10, ICES, 2010.

[24] L. Demkowicz, J. Gopalakrishnan, and A. Niemi. A class of Discontinuous Petrov-Galerkin methods. Part III: adaptivity. Technical Report 10-1, ICES, 2010.

[25] M. Giles. Nonreflecting boundary conditions for Euler equation calculations. *AIAA Journal*, 28(12):2050–2058, 1990.

[26] G.H. Golub and C.F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[27] T.J.R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1-4):387–401, 1995.

[28] T.J.R. Hughes, G.R. Feijoo, L. Mazzei, and J.B. Quincy. The variational multiscale method–a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2):3–24, 1998.

[29] T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational Fluid Dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73(2):173–189, 1989.

[30] E. Isaacson and H.B. Keller. *Analysis of Numerical Methods*. Dover Publications, Mineola, NY, USA, 1994.

[31] C. Johnson and J. Pitkäranta. An analysis of the Discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 46(173):1–26, 1986.

[32] C.T. Kelley. *Solving nonlinear equations with Newton's method*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[33] S.G. Krantz and H.R. Parks. *The implicit function theorem: history, theory, and applications*. Birkhauser, 2002.

[34] L. Krivodonova, J. Xin, J.F. Remacle, N. Chevaugeon, and J.E. Flaherty. Shock detection and limiting with Discontinuous Galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics*, 48(3-4):323–338, 2004.

[35] N. Kroll. The ADIGMA Project. In *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 1–9. Springer Berlin / Heidelberg, 2010.

[36] P.D. Lax. *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1973.

[37] R.J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge University Press, Cambridge, UK, 2002.

[38] H.W. Liepmann and A. Roshko. *Elements of Gasdynamics*. Dover Publications, Mineola, NY, USA, 2001.

[39] D. Moro, N.C. Nguyen, J. Peraire, and J. Gopalakrishnan. A hybridized Discontinuous Petrov-Galerkin method for compressible flows (AIAA Paper 2011-197). In *Proceedings of the 49th AIAA Aerospace Sciences Meeting*, Orlando, FL, USA, Jan 2011.

[40] N.C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable Discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228(9):3232–3254, 2009.

[41] N.C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable Discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics*, 228(23):8841–8855, 2009.

[42] N.C. Nguyen, J. Peraire, and B. Cockburn. A hybridizable Discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12):582 – 597, 2010.

[43] N.C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable Discontinuous Galerkin method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 230(4):1147 – 1170, 2011.

[44] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer series in operations research. Springer, 1999.

[45] J.M. Ortega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[46] J. Peraire, NC Nguyen, and B. Cockburn. A hybridizable Discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations (AIAA Paper 2010-363). In *Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, FL, USA, Jan 2010.

[47] J. Peraire and P.-O. Persson. The compact Discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.

[48] P.O. Persson and J. Peraire. Sub-cell shock capturing for Discontinuous Galerkin methods (AIAA Paper 2006-112). In *Proceedings 44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, Jan 2006.

[49] Todd E. Peterson. A note on the convergence of the Discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal on Numerical Analysis*, 28(1):133–140, 1991.

[50] NH Reed and TR Hill. Triangle mesh methods for the neutron transport equation. Technical Report LA2 UR-73-479, Los Alamos Scientific Laboratory, 1973.

[51] R.D. Richtmyer and K.W. Morton. *Difference methods for initial-value problems*. Wiley-Interscience, 1967.

[52] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

[53] C.W. Shu. High-order Finite Difference and Finite Volume WENO schemes and Discontinuous Galerkin methods for CFD. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, 2003.

[54] E.F. Toro. *Riemann solvers and numerical methods for Fluid Dynamics: a practical introduction*. Springer Verlag, 2009.

[55] Xu, J. and Zikatanov, L. Some observations on Babuška and Brezzi theories. *Numerische Mathematik*, 94(1):194–202, 2003.

[56] O.C. Zienkiewicz and R.L. Taylor. *The finite element method, vol. 1–3*. Butterworth-Heinemann, Oxford, UK, 2000.

[57] J. Zitelli, I. Muga, L. Demkowicz, J. Gopalakrishnan, D. Pardo, and V.M. Calo. A Class of Discontinuous Petrov-Galerkin methods. Part IV: the optimal test norm and time-harmonic wave propagation in 1D. *Journal of Computational Physics*, 230(7):2406–2432, 2011.