T High-Order Hybridizable Discontinuous Galerkin Method for Viscous Compressible Flows

by

Mostafa Javadzadeh Moghtader

Universitat Politècnica de Catalunya
Programa de Doctorat de Ciència i Tecnologia Aeroespacial

Departament d'Enginyeria Civil i Ambiental

High-Order Hybridizable Discontinuous Galerkin Method for Viscous Compressible Flows

by

Mostafa Javadzadeh Moghtader

Doctoral Thesis
Advisors: Sonia Fernández-Méndez
Barcelona, September 2016

Abstract

High-Order Hybridizable Discontinuous Galerkin Method for Viscous Compressible Flows

Mostafa Javadzadeh Moghtader

Computational Fluid Dynamics (CFD) is an essential tool for engineering design and analysis, specially in applications like aerospace, automotive and energy industries. Nowadays most commercial codes are based on Finite Volume (FV) methods, which are second order accurate, and simulation of viscous compressible flow around complex geometries is still very expensive due to large number of low-order elements required. One the other hand, some sophisticated physical phenomena, like aeroacoustics, vortex dominated flows and turbulence, need very high resolution methods to obtain accurate results. High-order methods with their low spatial discretization errors, are a possible remedy for shortcomings of the current CFD solvers. Discontinuous Galerkin (DG) methods have emerged as a successful approach for non-linear hyperbolic problems and are widely regarded very promising for next generation CFD solvers. Their efficiency for high-order discretization makes them suitable for advanced physical models like DES and LES, while their stability in convection dominated regimes is also a merit of them. The compactness of DG methods, facilitate the parallelization and their element-by-element discontinuous nature is also helpful for adaptivity.

This PhD thesis focuses on the development of an efficient and robust high-order Hybridizable Discontinuous Galerkin (HDG) Finite Element Method (FEM) for compressible viscous flow computations. HDG method is a new class of DG family which enjoys from merits of DG but has significantly less globally coupled unknowns compared to other DG methods. Its features makes HDG a possible candidate to be investigated as next generation high-order tools for CFD applications.

The first part of this thesis recalls the basics of high-order HDG method. It is presented for the two-dimensional linear convection-diffusion equation, and its accuracy and features are investigated. Then, the method is used to solve compressible viscous flow problems modelled by non-linear compressible Navier-Stokes equations; and finally a new linearized HDG formulation is proposed and implemented for that problem, all using high-order approximations, which is $p > 2$ in this thesis. The accuracy and efficiency of high-order HDG method to tackle viscous compressible flow

problems is investigated, and both steady and unsteady solvers are developed for this purpose.

The second part is the core of this thesis, proposing a novel shock-capturing method for HDG solution of viscous compressible flow problems, in the presence of shock waves. The main idea is to utilize the stabilization of numerical fluxes, via a discontinuous space of approximation inside the elements, to diminish or remove the oscillations in the vicinity of discontinuity. This discontinuous nodal basis functions, leads to a modified weak form of the HDG local problem in the stabilized elements. First, the method is applied to convection-diffusion problems with Bassi-Rebay and LDG fluxes inside the elements, and then, the strategy is extended to the compressible Navier-Stokes equations using LDG and Lax-Friedrichs fluxes. Various numerical examples, for both convection-diffusion and compressible Navier-Stokes equations, demonstrate the ability of the proposed method, to capture shocks in the solution, and its excellent performance in eliminating oscillations is the vicinity of shocks to obtain a spurious-free high-order solution.

Acknowledgments

It all started in a summer day, with a Skype interview with Antonio Huerta and Sonia Fernández-Méndez. Thanks to them, and Marie-Curie Early Stage Research Fellowships, I've started this long journey and moved from Iran to Barcelona. I want to use the opportunity to express my gratitude to all those people who helped me in these years through my PhD, more than anybody, Sonia, for being much more then a supervisor and helping me with all types of my problems, either small or big. Every part of this thesis, enjoyed her in-depth support and omnipresent participation in the research, and I deeply thank her for the guidance and being always accessible. I also want to thank Giorgio Giorgiani to help me start learning HDG method and provide me with some basic examples and suggestions.

All these years in LaCàN, I have enjoyed the accompany of many wonderful people. The HDG people, Giorgio, Aleks and Ceren were there to share the knowledge as friends. At the beginning, I didn't speak Spanish and slowly, I learnt the language and integrated, partially thanks to other PhD students in the lab; Cristina, David, Elena, Abel and Francesc. I also had the chance to share the lab with Miquel, Raul, Raquel, Dani, Drhuba, Magnus, Daobing, Albert, Boyee and Olga in later years and we have lots of good memories; from our not very great Futsal team to lunches at the terrace of the building C2. I should also thank Imma who helped me through bureaucratic works in UPC.

When I came first to Barcelona, I knew nobody and I quite didn't know what to expect, but this generous and beautiful city surprised me with its people and live atmosphere. The long list of all the friends I've made here is too long, but I enjoyed friendship and accompany of a big group of people from a very diverse backgrounds and countries. Arash and Ali were just two of the best ones I've met here and I hope we stay in touch for many years to come. Overall, I can say that I feel *Barcelonés*, and the years spent here will stay with me.

Finally, I have to thank my parents, Reza and Zahra, and my family who have been supporting me unconditionally during these years; without the support of my parents I wouldn't be here now.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past decades, Computational Fluid Dynamics (CFD), has been established as an essential tool for engineering design and analysis. This is mainly due to developments in computational power and algorithms available to workstations for scientists and engineers. One of the fields in which CFD has been extensively used, is aerospace industry. Improved solution strategies, combined with sophisticated physical models, have made CFD a key technology in all stages of design and development in aerospace and aeronautics, see Wang and Anderson [2012], Kroll [2006], Oliver [2008].

Current CFD solvers used in the industry, are mainly based on Finite Volume (FV) Methods for Euler, Navier-Stokes or Reynolds Averaged Navier-Stokes (RANS) and nominally second order accurate. After decades of development, these methods have become robust and affordable for RANS simulations on small CPU clusters.

However, in most of real engineering applications, the degree of accuracy on irregular and highly stretched meshes, falls between one and two. As a result, lots of nodes are needed to capture flow around complex geometries, and large aerodynamic simulations of viscous compressible flows around such configurations are still very expensive, see Hartmann and Houston [2009]. On the other hand, difficulties in some applications, like vortex dominated flows or aeroacoustics, lead to complex multi-scale problems, like Large Eddy Simulation (LES) of turbulent flow, which require very high resolution to obtain accurate results, see Burgess [2011]. High-order methods are more efficient than linear methods, see Huerta et al. [2013], and in recent years, there have been significant efforts to design and develop them to reduce the spatial discretization error, and to compare them with industrial solvers.

Such an effort can be seen, for instance, in the EU Framework 6 project called ADIGMA (adaptive high-order variational methods for aerospace applications), in which, a group of universities, research centres and industrial partners collaborated to investigate on innovative adaptive higher-order methods for the compressible flow equations, enabling reliable, numerical solutions for large-scale aerodynamic applications in aircraft design. Their results prove the competitiveness of high-order methods compared to standard finite volume solvers for airfoil computations and for 3D inviscid and laminar flow around simple geometries. Besides the potential and capabilities of high-order methods, the limitations were also identified: further research on high-order adaptive methods for turbulent flows and better memory-efficient strategies, as well as improvement in generation of coarse high-order meshes were recommended. For more information see Hartmann and Houston [2009], Kroll [2006], Chalot and Normand [2010], Kroll et al. [2010], Kroll [2010].

In another study, NASA CFD vision 2030 report by Slotnick et al. [2014], also put a light on inadequate ability of current solvers for simulations of complex flow phenomenon, like turbulent flows with significant regions of separations. It also recognized mesh generation and adaptivity as significant bottlenecks for CFD simulations and required revolutionary algorithmic improvement as a requisite for future advanced simulations.

Overall, the required ingredients of the possible next generation CFD tools for aerospace applications are not limited to, but include:

• high-order/low-dispersion discretizations for sophisticated physical phenomenon (especially for DES, LES and transition to turbulence), and for higher computational efficiency

• ability to handle complex geometries on unstructured meshes, as a necessary requirement for real engineering problems

• error estimation and adaptation techniques to minimize computational efforts and provide reliable solution

• efficient solution strategies, robustness and suitability for parallelization

Some of the most promising methods for this purpose are Discontinuous Galerkin (DG) Finite Element Methods (FEM). These methods are finite element methods in which the solution is approximated by means of element-by-element polynomial functions, without global continuity requirement, and therefore the numerical solution is discontinuous at element interfaces, see for example Bassi et al. [2005]. DG methods

were first introduced by Reed and Hill [1973] for the neutron transport equation. In the last decades, after successfully being applied to the solution of non-linear hyperbolic problems, they have emerged as an alternative to FV solvers on unstructured meshes. The high-order accuracy in DG methods can be obtained by using higher order polynomials for approximation. On the other hand, their element-by-element definition provides compactness, hence makes them ideal for parallelization. They are locally conservative and because of their built-in stabilization mechanism, they are inherently stable in convection dominated regimes. In addition, their discontinuous nature, facilitates the implementation of hp-adaptive schemes.

Besides all advantages, for the same mesh and order of approximation, the number of globally coupled degrees of freedom of classical DG methods is significantly higher than their Continuous Galerkin (CG) counterparts. As a result, classical DG methods are computationally expensive for steady or implicit solvers and the big challenge is to make them competitive and robust for realistic problems.

In recent years, a new DG method has been developed by Cockburn and Gopalakrishnan [2004] called Hybridizable Discontinuous Galerkin (HDG) method. Among all DG methods, HDG method outstands for steady and implicit schemes, mainly due to its reduced number of degrees of freedom. Hence, while HDG maintains the advantages of DG methods, it is computationally much less expensive. In fact for high degrees of approximations, HDG can be as efficient as traditional CG methods on triangles and quadrilaterals, see for instance Kirby et al. [2011], Giorgiani et al. [2014]. In addition, HDG also has other promising feature; its superconvergence properties, which is possible through a cheap element-by-element procedure. This is due to optimal convergence rate for both primal variables and their derivatives, which makes it somehow unique in DG families because of optimal convergence rate of viscous fluxes in multi-dimensions, see Peraire et al. [2010].

Apart from discretization aspect, complex aerodynamic flow fields exhibit a wide range of phenomena, like thin boundary layers, high streamlined curvature regions and shock waves. This latter one puts one of challenges of high-order methods: Gibbs phenomena, see Barter [2008]. That is, numerical oscillations appear when non-smooth or discontinuous solutions are approximated with polynomials. HDG method posses a built-in stabilization, which is enough for highly convective regimes, but it is not enough to capture sharp gradients in high-order methods, hence needs further treatments. These oscillations can be eliminated or damped with shock-

capturing techniques which introduce additional dissipation in the vicinity of shocks. The best stabilization method to smooth out the spurious solution, but not damp the discontinuity, is a current field of research for high-order methods, and main goal of this thesis.

1.1 Objectives and outline

The overall goal of this thesis is to develop an efficient and robust high-order HDG method for compressible Navier-Stokes equations modelling compressible viscous flows. To deal with the shocks in transonic or subsonic regimes, a novel shock-capturing technique is developed, which uses flux stabilization inside the elements to capture the shocks in large high-order elements. The main achievements of this thesis are:

• **Implementation of a high-order HDG solver for compressible viscous flow.** The HDG discretization is implemented for compressible Navier-Stokes equations, either steady or unsteady. The weak form of the problem is recalled in section 2.2 and several examples are used to investigate the accuracy and convergence of the method for a problem with known analytical solution. In addition, efficiency of high-order elements versus low-order elements is investigated and proved for laminar flow around NACA 0012 airfoil. Development of this part is based on the work of Peraire et al. [2010], and more details of implementation aspects of the method are presented in appendix C.2. Overall, high-order approximation allows usage of larger elements, reducing the computational cost, and providing low spatial discretization error, which is suitable for applications that need very high-resolutions like acoustics or LES. The high-order HDG method, provides such a high-resolution we seek for both convection-diffusion and compressible Navier-Stokes equations.

• **Derivation and implementation of a linearized HDG method for compressible viscous flow.** Aiming to have a more robust solver, a new linearization method for the unsteady compressible Navier-Stokes equations is considered. It is expected to reduce the computational cost and improve the robustness in unsteady problems, and facilitate the convergence to the steady state solution, which is a challenge for high-order methods because of much reduced numerical dissipation associated with these methods, see Wang et al. [2013]. This linearization method is based on a linear extrapolation of solution from previous time steps, and leads to a linear system of equations to be solved in each time step. This linearization improves

the robustness of the solver and can be used interchangeably with Newton-Raphson non-linear solver for HDG simulation of compressible viscous flow. The procedure is explained in section 2.3 and successfully implemented. The applicability of the proposed method to compute the solution of viscous compressible flow is shown in results, and critical time-step size and computational cost of the method has been studied. The details of the implementation are presented in appendix C.3.

• **Development of a novel shock-capturing technique for HDG method.** The presence of shocks, as a common phenomena in compressible flows, or sharp fronts in convection-diffusion problems, requires some kind of stabilization techniques in such numerical simulations. One big challenge for high-order methods is preserving high accuracy of approximations in the vicinity of the shocks, to properly resolve the shock; because, in some cases, the low order of accuracy can pollute the solution, even away from discontinuities, see Casoni et al. [2013]. An important goal of this thesis is to reach highly accurate solution even when shocks appear in the flow field. Our strategy, as explained in section 3.1, is to exploit the stabilization induced by DG numerical fluxes to capture sharp fronts of the solution inside high-order elements. To do so, a discontinuity sensor developed by Persson and Peraire [2006] is used to detect the elements affected by sharp fronts. Based on the smoothness of the solution, the approximation space inside each element is modified, from a standard continuous representation of the solution to a piecewise constant approximation. As a result of using this new space of approximation, the HDG weak form is modified to account for discontinuities inside the elements. In the presence of shock, the new discontinuities inside an element introduce the sufficient amount of stabilization because of the numerical fluxes, and shock can be captured in one large high-order element. In section 3.1 the method is developed for convection-diffusion equations and in section 3.2 it is extended to compressible Navier-Stokes equations.

The outline of this thesis is as follows. First, literature review is presented for both DG methods for viscous compressible flows and for shock-capturing for DG methods, in sections 1.2.1 and 1.2.2, respectively. Chapter 2 is about high-order HDG method, and it presents the method for convection-diffusion equation in section 2.1, which includes formulation and numerical examples. Later, high-order HDG for compressible Navier-Stokes equations is discussed in section 2.2, which also contains the formulation and numerical examples. And then, in section 2.3, the development of a new linearization for compressible Navier-Stokes equations is covered, and the

corresponding HDG formulation and numerical tests are also included. Chapter 3 develops and studies the performance of new shock-capturing strategy for HDG, which starts with section 3.1, presenting the HDG with shock-capturing for convection-diffusion problems with internal layers. This section includes the discretization space and discontinuous shape functions, modified HDG formulation, discontinuity sensor, and numerical examples. Then, section 3.2, presents high-order HDG with shock-capturing for compressible Navier-Stokes equations with numerical simulations of compressible viscous flows with shocks. Finally, the summary of the work is presented in chapter 4 with ideas for future developments in section 4.1.

1.2 State of the art

In the last decades, the research in DG FEM for the numerical simulation of compressible viscous flow problems has been a very active field in computational mechanics. These problems have a wide range of applications, from aerodynamics and combustion simulations, to climate modellings.

In the following sections, a survey of the DG approaches, for solution of compressible Navier-Stokes equations, and shock-capturing techniques for compressible flow, is presented.

1.2.1 Discontinuous Galerkin methods for viscous compressible flows

At first, DG methods were proposed by Reed and Hill [1973] and analysed by Lasaint and Raviart [1974]. Later, DG methods were extended to non-linear hyperbolic conservation laws by Cockburn et al. [1989, 1990], Cockburn and Shu [1989]. In next decades, DG emerged as a family of powerful high-order accurate methods for the solution of the different non-linear conservation laws and convection-dominated problems, see for instance Bassi and Rebay [1997a], Arnold et al. [2002], Cockburn [2003, 2004], Bassi et al. [2005], Cockburn et al. [2000], Cockburn and Shu [2001], Nguyen et al. [2009a].

DG methods are suitable to construct robust and stable high-order schemes on unstructured and non-conforming grids, and their compactness makes them ideal for parallelization and adaptivity. As a result, there has been an increasing interest in

DG methods in various disciplines of numerical modelling of physical phenomenon, including compressible and viscous flow in aerodynamics, see for instance Hartmann [2005a], Bassi and Rebay [1997a,b], Baumann and Oden [2000, 1999], Dolej [2004], Hartmann and Houston [2002], Hartmann et al. [2010], Fidkowski et al. [2005], Drozo et al. [1998], Klaij et al. [2006b,a], Lomtev and Karniadakis [1999], Cockburn [2004]. Despite all advantages of DG methods, their main challenge to enter the practical and industrial application is the high computational cost in comparison with FEM, FVM and FDM, see Nguyen et al. [2011c]. This is because of much higher number of globally coupled degrees of freedom of classical DG methods, in comparison with CG methods for the same mesh and same polynomial degree.

The new class of DG method, HDG method, offers a way to address the issue of computational cost. Meanwhile HDG maintains the merits of DG, it uses a significantly reduced number of degrees of freedom. Hence, it is less computationally costly and needs less memory. HDG method, like other methods of DG family, enforces the equations in an element-by-element manner, hence it is suitable for parallelization and gives rise to a locally conservative method. HDG can handle meshes of different element shapes, sizes and orders of approximation, so it is ideal for hp-adaptivity. Its built-in stabilization does not degrade its high-order accuracy, and can be applied to different systems of partial differential equations (PDE). Among all DG methods, HDG stands for its reduced number of globally coupled degrees of freedom, and its superconvergence. All of these interesting features make HDG an interesting high-order alternative to current low-order solvers, particularly for applications in fluid dynamics.

HDG was first developed for elliptic problems in a series of works by Cockburn and Gopalakrishnan [2004, 2005], Cockburn et al. [2009c] and was used and analysed extensively for steady state diffusion equation in works of Cockburn et al. [2008, 2009e,d, 2014]. It was extended to time-dependent diffusion problem by Chabaud and Cockburn [2012] and then, to convection-diffusion problems, by Cockburn et al. [2009b], Nguyen et al. [2009a,b], Egger and Schberl [2010], Rhebergen and Cockburn [2013], Oikawa [2014]. HDG method was applied to the wave equation by works of Nguyen et al. [2011a], Griesmaier and Monk [2014], Feng and Xing [2013], Giorgiani et al. [2013a,b] and it was developed for linear and non-linear elasticity, see Soon et al. [2009], Kikuchi et al. [2009], Nguyen and Peraire [2012], and also for Timoshenko beams and biharmonic problems, by Celiker et al. [2012, 2010], Cockburn

et al. [2009a]. HDG was developed and analysed for flow problems, first for solving the Stokes equations, in works of Carrero et al. [2006], Cockburn and Gopalakrishnan [2009], Nguyen et al. [2010], Cockburn et al. [2011], Cockburn and Cui [2012]. Then, HDG was extended to incompressible Navier-Stokes equations, see Nguyen et al. [2011b], Rhebergen and Cockburn [2012], Rhebergen et al. [2013], Giorgiani et al. [2014], Montlaur and Giorgiani [2015], Qiu and Shi [2015]. And finally, the Euler and compressible Navier-Stokes equations of gas dynamics are getting attention of HDG researchers in recent years in works of Peraire et al. [2010], Nguyen et al. [2010], Nguyen and Peraire [2011], Schütz et al. [2012], Moro et al. [2012], Nguyen and Peraire [2012], Roca et al. [2013], Woopen et al. [2014b], Jaust and Schtz [2014], Woopen et al. [2014a], Woopen and May [2015].

Overall, HDG is still a novel DG method and its development and application for CFD is an open and interesting field of study. The main task of this thesis is to develop a robust and efficient HDG solver for compressible viscous flow problems in presence of shocks and investigate its properties.

1.2.2 Shock-capturing for discontinuous Galerkin methods

Shock waves in fluids, are a type of propagation of disturbance that cause almost abrupt changes in characteristics of the medium, see Anderson [2010]. Large amplitude compression waves, such as that produced by an explosion, or by supersonic or near supersonic motion of a body in a medium, are quite common phenomenon in many problems of interest in compressible flows. In transonic, supersonic and hypersonic flows the presence of normal, oblique or bow shocks have significant influence on the lift, drag and heat conduction loads of the aircraft, and correct modelling and simulation of shocks is an important issue in computational aero-thermodynamics.

After years of research, low-order methods for simulations of flow with shocks are a mature field of study; however this is not the case for high-order methods. In case of HDG method, like other DG methods, some inherent stability due to numerical fluxes can be seen near discontinuities or sharp fronts; like the case of a linear convection-diffusion equation with discontinuous boundary conditions, or compressible Navier-Stokes equations with weak shocks on coarse meshes. As reported by Hartmann [2005a], these types of problems can be discretized and solved without any additional stabilization. However, solutions suffer from spurious oscillations near discontinuities or very sharp front, with high-order approximation. This polluted solution may

even affect the convergence of steady state solvers, and additional shock-capturing methods have to be implemented for high-order HDG method to further stabilize the numerical discretization, and to overcome local numerical oscillations (overshoots and undershoots) in the vicinity of sharp fronts and shocks for high-order computations. Hence, dealing with discontinuities is one of the big challenges of high-order methods and although it has been an active field of research still there is great debate on the most effective approach to compute flow with shocks in CFD community, see Burgess [2011], Vincent and Jameson [2011].

In recent years, shock-capturing techniques for high-order methods have become a rich and intense area of research and have been investigated by many authors, see for instance Cockburn and Shu [2001], Hartmann and Houston [2002], Krivodonova et al. [2004], Bassi et al. [2009], Wang and Mavriplis [2009], Hartmann [2006], Casoni et al. [2013]. Some of traditional shock-capturing techniques have been developed in the past decades; and many researchers have extended classical shock-capturing methodologies of finite differences (FD) and finite volume (FV) schemes to high-order DG and HDG methods, see for instance Zhu and Qiu [2009], Persson and Peraire [2006], Nguyen and Peraire [2011], Barter [2008], Barter and Darmofal [2010], Burgess and Mavriplis [2012].

One of the main techniques to capture the shock is artificial viscosity, which uses additional dissipation. One may think of explicitly adding the dissipation term to the equations, see Anderson [1995], Burgess [2011]. To ensure the consistency of the method, artificial viscosity in the perturbed partial differential equation must vanish as mesh size goes to zero and also in the regions of smooth solution. Artificial diffusion was first introduced by von Neumann and Richtmyer [1950] and further developed later by Jameson et al. [1981], Baldwin and MacCormack [1975]. It became popular in context of Streamline Upwind Petrov-Galerkin (SUPG) finite element methods, see Hughes et al. [1986], Hughes and Mallet [1986a,b], Hughes et al. [1987] and in the last decades, it has been used in DG simulations of compressible flow, see for instance Bassi and Rebay [1995], Baumann and Oden [2000], Hartmann and Houston [2002], Hartmann [2005a, 2006], Bassi et al. [2009], Hartmann [2013]. As described by Barter and Darmofal [2010], "artificial viscosity expands the thickness of the shock layer so that it safely exceeds the resolution length scales of the numerical method and eliminates the spurious oscillations". Hence, the amount of artificial viscosity is of great importance and some authors propose a sub-cell based artificial viscosity

to increase the accuracy of the solution in the vicinity of shocks, and impose as less amount of artificial viscosity as possible, see for instance Persson and Peraire [2006], Casoni et al. [2013]. Artificial diffusion techniques can capture shocks in a robust and accurate manner, however, the amount of artificial viscosity is not straightforward due to its non-linearity, and also it is quite difficult to incorporate directionality.

Some other classical shock-capturing methods are limiting techniques. The Total Variation Diminishing (TVD) methods, bound the variations in solution so that no new local extrema forms in the domain, see LeVeque [1992]. These methods were initially designed in the context of FD and FV by van Leer [1974, 1977a,b, 1979], and have been successfully developed in context of DG methods, namely Runge-Kutta Discontinuous Galerkin (RKDG) methods by Cockburn et al. [1989, 1990], Cockburn and Shu [1998, 2001]. RKDG, which are a class of explicit RK schemes, use a combination of slope limiters and approximate Riemann solvers, and as a result, they are Total Variation Bounded in the Means (TVBM). These limiters are specially developed to damp the oscillations, but the order of the approximation is reduced to linear or constant. As a result of this order reduction, accuracy can only be improved by mesh refinement in the vicinity of the shock.

An alternative to maintain high-order accuracy, is adding degrees of freedom to capture sharper shock transition. These methods are Essentially Non-Oscillatory (ENO), developed by Harten et al. [1987]. An improvement to this method is Weighted Essentially Non-Oscillatory (WENO) schemes, developed by Liu et al. [1994]. These methods use a single or a non-linearly weighted multiple stencil to reconstruct a high-order polynomial representation from a set of local cell average values while eliminating spurious oscillations. ENO methods have been developed for FD and FV methods and have been extended to DG methods, see for instance Qiu and Shu [2004, 2005], Zhu and Qiu [2009, 2013], Jiang et al. [2012], Luo et al. [2007]; However, there are still difficulties in these approaches specially because an implicit time marching to steady state solution has not been developed yet. The other issues are robustness for high-order approximation and extension of the method to multiple dimensions. Overall the compact implicit WENO methods for DG is an active research field.

In a different approach, Huerta et al. [2012] proposed to use the stabilization induced by DG numerical fluxes to capture sharp fronts of the solution inside high-order elements for Euler equations of gas dynamics. By means of this shock-capturing

technique, the order of the approximation is reduced only in the elements where the solution is not smooth. Thus, the high-order accuracy of order $p+1$ in the large majority of the domain, is locally decreased to order h/p only in the elements where the shock is contained, being p the degree of approximation, and h the element size. As a results, no mesh adaptation is needed, and sharp fronts can be captured without modifying the DOFs or mesh topology. In this thesis, this main idea is developed for HDG method for convection-diffusion and Navier-Stokes equations. The shock-capturing technique proposed for HDG method is inspired by the ideas of Huerta et al. [2012] and uses DG fluxes inside elements to stabilize the method, which give rise to a modification of the variational form of the local HDG problem in the vicinity of shocks.

Chapter 2

High-order hybridizable discontinuous Galerkin method

In this chapter, HDG method for two-dimensional linear convection-diffusion equation is introduced according to the work of Nguyen et al. [2009a], and then the method is used to solve the compressible viscous flow problems for non-linear compressible Navier-Stokes equations. Finally, a new linearized HDG method is proposed and implemented for compressible Navier-Stokes equations.

In first part of this chapter, section 2.1, the model convection-diffusion equation is presented, and the approximation spaces for the solution, derivation of formulations of HDG discretization and local post-processing of the solution to obtain superconvergence are also described. Convergence and accuracy of HDG method for solution, gradient of solution and post-processed solution are investigated through numerical tests, and the expected high-order accuracy is achieved. In section 2.2, HDG is used to solve the compressible Navier-Stokes equations. The governing equations and the variational form of the problem are presented. Again, accurate solutions are obtained using high-order elements, and the efficiency of high-order HDG is also demonstrated in comparison with low-order approximations. Then, in section 2.3, a new linearized HDG method for compressible Navier-Stokes equations is developed and implemented successfully for steady and unsteady test cases, and its accuracy is compared with both steady and unsteady non-linear solver.

2.1 The HDG method for convection-diffusion equation

A linear convection-diffusion model is considered here as a model problem, because it has both diffusive and convective operators like Navier-Stokes equations. In addition, there are numerous applications in science and technology, e.g., transport of a contaminant in air or water, oil reservoir flow, electro-hydrodynamics or concentration of electrons in semiconductor devices, see for instance Egger and Schberl [2010]. HDG for convection-diffusion equation has been already developed by Nguyen et al. [2009a], here we try to describe the main ideas and features of HDG method through it.

Let $\Omega \in \mathbb{R}^{sd}$ be an open bounded domain, where sd is the space dimension, with boundary $\partial\Omega$ split in the Dirichlet, $\partial\Omega_D$, and Neumann, $\partial\Omega_N$, boundaries. The steady convection-diffusion equation can be written as

$$\begin{aligned} \boldsymbol{\nabla}\cdot(\boldsymbol{c}u) - \boldsymbol{\nabla}\cdot(k\boldsymbol{\nabla}u) &= f \qquad \text{in } \Omega, \\ u &= g_D \qquad \text{on } \partial\Omega_D, \\ (-k\boldsymbol{\nabla}u + \boldsymbol{c}u)\cdot\boldsymbol{n} &= g_N \qquad \text{on } \partial\Omega_N, \end{aligned} \tag{2.1}$$

where u is the unknown, k is a positive diffusion coefficient, $\boldsymbol{c}$ is a smooth convection velocity field, f is a source term, and g_D and g_N are given values for essential and natural boundary conditions respectively.

For DG approach the domain Ω is partitioned in n_{el} disjoint elements, Ω_e, with boundaries $\partial\Omega_e$, such that

$$\bar{\Omega} = \bigcup_{e=1}^{n_{el}} \bar{\Omega}_e, \quad \bar{\Omega}_l \cap \bar{\Omega}_m = \emptyset \quad \text{for} \quad l \neq m, \tag{2.2}$$

and the union of all n_{fc} faces, Γ_f, is denoted as

$$\Gamma = \bigcup_{e=1}^{n_{el}} \partial\Omega_e = \bigcup_{f=1}^{n_{fc}} \Gamma_f \tag{2.3}$$

The discontinuous setting now induces a new problem, equivalent to (2.1) corresponding to a system of first order partial differential equations (mixed form) stated element-by-element

$$\left.\begin{aligned} \boldsymbol{q} + k\boldsymbol{\nabla}u &= \boldsymbol{0} \\ \boldsymbol{\nabla}\cdot(\boldsymbol{c}u + \boldsymbol{q}) &= f \end{aligned}\right\} \qquad \text{in } \Omega_e, \quad e = 1, \cdots, n_{el} \tag{2.4}$$

and some global equations imposing continuity between elements and boundary conditions

$$[\![u\boldsymbol{n}]\!] = \mathbf{0} \qquad \text{on } \Gamma \setminus \partial\Omega, \tag{2.5}$$

$$[\![(\boldsymbol{c}u + \boldsymbol{q}) \cdot \boldsymbol{n}]\!] = 0 \qquad \text{on } \Gamma \setminus \partial\Omega, \tag{2.6}$$

$$u = g_D \qquad \text{on } \Omega_D, \tag{2.7}$$

$$(\boldsymbol{c}u + \boldsymbol{q}) \cdot \boldsymbol{n} = g_N \qquad \text{on } \Omega_N, \tag{2.8}$$

In equation (2.4), a new variable $\boldsymbol{q}$ is introduced, proportional to the gradient of the solution (actually it is equal to $-k\boldsymbol{\nabla}u$). The *jump* $[\![\cdot]\!]$ and *mean* $\{\cdot\}$ operators are defined at internal faces, i.e. on $\Gamma \setminus \partial\Omega$ using the values from the elements on the right and left of the face, Ω^+ and Ω^-,

$$[\![a]\!] = a^+ + a^-, \tag{2.9}$$

$$\{a\} = (a^+ + a^-)/2, \tag{2.10}$$

The main difference between the two is that the jump always involves the normal to the interface, see more details in Giorgiani et al. [2014]. Note that equations (2.5) and (2.6) impose the continuity of the solution and normal component of the flux through element boundaries.

A major feature of the HDG method is that, unknowns are finally reduced to the skeleton of the mesh, that is, to the union of all faces Γ. This is done via introduction of a new variable $\hat{u}$, corresponding to the trace of the solution Γ. The new variable $\hat{u}$ allows to state the so-called *local problem* in each element, corresponding to the convection-diffusion equations (2.4) with Dirichlet boundary conditions,

$$\begin{aligned} \boldsymbol{q} + k\boldsymbol{\nabla}u &= \mathbf{0} && \text{in } \Omega_e, \\ \boldsymbol{\nabla} \cdot (\boldsymbol{c}u + \boldsymbol{q}) &= f && \text{in } \Omega_e, \\ u &= \hat{u} && \text{on } \partial\Omega_e, \end{aligned} \tag{2.11}$$

Note that the trace of the solution $\hat{u}$ acts as boundary condition in the local problem (2.11) in each element, and as unknown for the so-called global problem, that corresponds to the continuity condition (2.6), and the boundary conditions (2.7) and (2.8). Note that the continuity of u in (2.5) is ensured by the fact that $\hat{u}$ is single-valued on Γ, that is, in the local problems, the same value $\hat{u}$ is imposed on the face shared by two elements. The approximation spaces for u, $\boldsymbol{q}$ and $\hat{u}$, the HDG weak form for local problems, and the weak form for the HDG global problem are detailed in next sections.

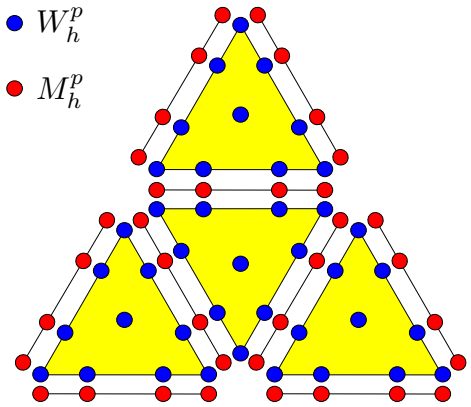


Figure 2.1: Spaces of solutions

HDG formulation

For HDG discretization, two types of finite element spaces are defined, one for functions in the elements interior and another for trace functions, as

$$\begin{aligned} \mathscr{W}_h^p &= \{v \in \mathcal{L}^2(\bar{\Omega}) : v|_{\Omega_e} \in \mathcal{P}^p(\Omega_e), \text{ for } \quad e = 1, ..., n_{el}\}, \\ \mathscr{M}_h^p &= \{\mu \in \mathcal{L}^2(\Gamma) : \mu|_{\Gamma_f} \in \mathcal{P}^p(\Gamma_f), \text{ for } \quad f = 1, ..., n_{fc}\}, \end{aligned} \tag{2.12}$$

where $\mathcal{P}^p(\bar{\Omega})$ is the set of polynomials of degree at most p on the elements, and $\mathcal{P}^p(\Gamma)$ is the space of polynomials of degree at most p on the faces. $\mathscr{M}_h^p$ is the space for the approximation of trace function on Γ, defined at the faces.

The HDG discretization of the local problem (2.11) in each element, is stated as: given $\hat{u}_h$ on $\partial\Omega_e$, find $u_h \in \mathscr{W}_h^p$ and $\boldsymbol{q}_h \in [\mathscr{W}_h^p]^{sd}$ such that

$$\begin{aligned} &(k^{-1}\boldsymbol{q}_h, \boldsymbol{z})_{\Omega_e} - (u_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e} + \langle \hat{u}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e} = 0, \\ &-(\boldsymbol{c}u_h + \boldsymbol{q}_h, \boldsymbol{\nabla} r)_{\Omega_e} + \langle \widehat{(\boldsymbol{c}u_h + \boldsymbol{q}_h)}\cdot\boldsymbol{n}, r\rangle_{\partial\Omega_e} = (f, r)_{\Omega_e}, \end{aligned} \tag{2.13}$$

for all $r \in \mathscr{W}_h^p$ and all $\boldsymbol{z} \in [\mathscr{W}_h^p]^{sd}$, where $(\cdot,\cdot)_{\Omega_e}$ denotes the $\mathcal{L}^2$ scalar product in the element Ω_e and $\langle\cdot,\cdot\rangle_{\partial\Omega_e}$ denotes the $\mathcal{L}^2$ scalar product in the element boundary $\partial\Omega_e$. Numerical traces $\hat{u}_h$ and $\widehat{\boldsymbol{c}u_h + \boldsymbol{q}_h}$ are approximations to u_h and $\boldsymbol{c}u_h + \boldsymbol{q}_h$ over $\partial\Omega_e$ respectively. The trace $\hat{u}_h$ will be an unknown in the global problem, but the trace $\widehat{\boldsymbol{c}u_h + \boldsymbol{q}_h}$ is set as

$$\widehat{\boldsymbol{c}u_h + \boldsymbol{q}_h} = \boldsymbol{c}\hat{u}_h + \boldsymbol{q}_h + \tau(u_h - \hat{u}_h)\boldsymbol{n} \quad \text{on} \quad \Gamma, \tag{2.14}$$

where τ is the local stabilization parameter, which has a strong effect on the stability and accuracy of the method, see Nguyen et al. [2009a] and remark 2.2. Replacing the

definition of the numerical flux $\hat{u}_h$, (2.14), the weak form of the local problem for each element is: find $u_h \in \mathscr{W}_h^p$ and $\boldsymbol{q}_h \in [\mathscr{W}_h^p]^{sd}$ such that

$$\begin{aligned}(k^{-1}\boldsymbol{q}_h, \boldsymbol{z})_{\Omega_e} - (u_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e} + \langle \hat{u}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e} = 0,\\ (\boldsymbol{\nabla}\cdot\boldsymbol{q}_h, r)_{\Omega_e} - (\boldsymbol{c}u_h, \boldsymbol{\nabla}r)_{\Omega_e} + \langle \tau u_h, r\rangle_{\partial\Omega_e} + \langle (\boldsymbol{c}\cdot\boldsymbol{n} - \tau)\hat{u}_h, r\rangle_{\partial\Omega_e} = (f, r)_{\Omega_e},\end{aligned} \tag{2.15}$$

for all $r \in \mathscr{W}_h^p$ and all $\boldsymbol{z} \in [\mathscr{W}_h^p]^{sd}$.

It is important to notify that the local problem can be solved in an element-by-element manner, to express the solution at each element, u_h and $\boldsymbol{q}_h$, in terms of the trace of the solution, $\hat{u}_h$. Thus, the actual unknown will be $\hat{u}_h$ on Γ.

The problem is then closed with the discretization of the conservativity condition, (2.6), using (2.14), and the Neumann boundary condition (2.8): find $\hat{u}_h \in \mathscr{M}_h^p$ such that

$$\sum_{e=1}^{n_{el}} \langle (\boldsymbol{c}\hat{u}_h + \boldsymbol{q}_h)\cdot\boldsymbol{n} + \tau(u_h - \hat{u}_h), \mu\rangle_{\partial\Omega_e} = g_N, \quad \forall \mu \in \mathscr{M}_h^p, \tag{2.16}$$

where u_h and $\boldsymbol{q}_h$ are solutions of the local problem, (2.15). Note that the solution of the local problems (2.15) can be replaced in the global problem (2.16), leading to global system of equations that only involves $\hat{u}_h$, as

$$\mathbb{K}\hat{u}_h = \mathbb{F}, \tag{2.17}$$

with an important reduction in DOFs.

Remark 2.1. *In HDG method, the Dirichlet boundary condition can be enforced weakly, setting $\hat{u}_h = \mathbb{P}(g_D)$ on Dirichlet part of boundary, $\partial\Omega_D$, where $\mathbb{P}(g_D)$ is the L^2 projection of g_D on the approximation space for the traces on $\partial\Omega_D$. On the other hand, Neumann boundary condition is enforced on numerical flux in* (2.16)*. This methods proves to be an easy and effective way to impose the boundary condition in convection-diffusion problems as used in different examples.*

Once the linear system (2.17) is solved, the solution, u_h and $\boldsymbol{q}_h$, in each element can be computed from u_h, with the local problem (2.15). A second element-by-element post-process can be performed to obtain a super-convergent solution, u_h^*: find $u_h^* \in \mathscr{W}_h^{p+1}$ such that

$$\begin{aligned}(-k\boldsymbol{\nabla}u_h^*, \boldsymbol{\nabla}r)_{\Omega_e} &= (\boldsymbol{q}_h, \boldsymbol{\nabla}r)_{\Omega_e}\\ (u_h^*, 1)_{\Omega_e} &= (u_h, 1)_{\Omega_e}\end{aligned} \tag{2.18}$$

for all $w \in \mathscr{W}_h^{p+1}$ and $e = 1, ..., n_{el}$. It should be noted that this post-process leads to superconvergence of the solution due to optimal rate of convergence for both solution and its gradient which is unique to HDG method between DG methods. For more details, see Cockburn et al. [2008].

Remark 2.2. *The local stabilization parameter, τ, is shown to have important effects on the accuracy and convergence of the HDG method. Inappropriate choice of τ may lead to significant deviation from accurate solution or loss of optimal convergence rates for scalar variable or flux or superconvergence. Following Nguyen et al. [2009a], here the stabilization parameter is considered as*

$$\tau = \tau_d + \tau_c \tag{2.19}$$

where τ_d and τ_c represent the local stabilization parameter related to the diffusion and convection, respectively. A good expression for τ_d and τ_c may be stated as

$$\tau_d = \frac{k}{\ell}, \qquad \tau_c = |\boldsymbol{c} \cdot \boldsymbol{n}|, \tag{2.20}$$

where ℓ is a representative value of diffusive length scale, typically of the order of unity and independent of mesh size h. It is shown by Nguyen et al. [2009a] that the above choices of the local stabilization parameter are useful and the optimal convergence rate of $p+1$ can be obtained for both approximate scalar variable and the flux, and also superconvergence rate of $p+2$ can be gained for post processed solution as shown in next section.

Numerical results

In this section, numerical results for two-dimensional steady state convection-diffusion problems are presented. The accuracy and convergence of the scalar solution and flux are investigated, and also the superconvergence of the post-processed solution is demonstrated. Computational meshes, obtained by splitting a regular Cartesian grid of n^2 quadrilaterals into a grid of $2n^2$ triangular elements, see Figure 2.2. The stabilization parameter, τ, is defined according to (2.19), with $l = 1$, and the nodes are distributed according to Fekete distribution in every element.

Example 1: Smooth convection-diffusion problem

The first example, from Cockburn et al. [2009b], is the solution of a steady convection-diffusion problem in $\Omega = (0,1) \times (0,1)$ with Dirichlet boundary condition $g_D = 0$ on

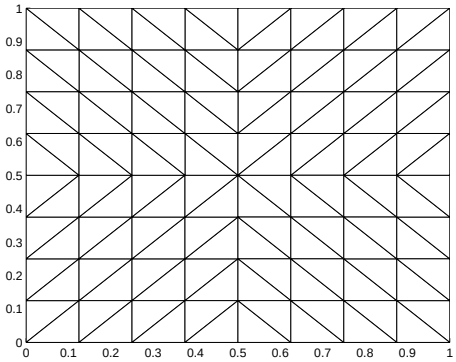


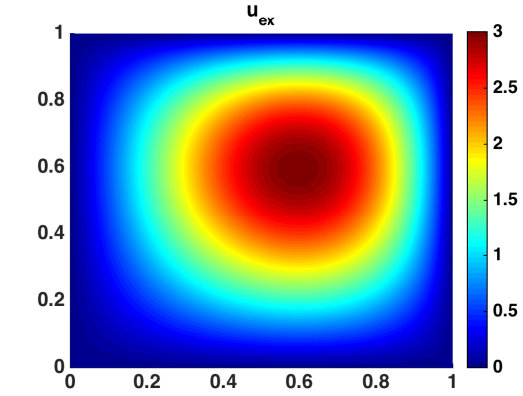
Figure 2.2: A typical triangular mesh

$\partial\Omega$. Convection and diffusion coefficients are $\boldsymbol{c} = (1, 1)$ and $k = 1$, respectively and the source term f is chosen such that the exact solution is

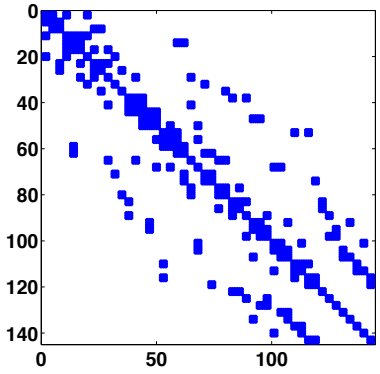
$$u(x, y) = \exp(x + y)\sin(\pi x)\sin(\pi y) \tag{2.21}$$

Figure 2.3(a) shows the smooth exact solution of the problem, while Figure 2.3(b) shows the sparsity pattern of the HDG matrix on a coarse mesh with $n = 8$ and polynomial of degree $p = 3$. The block structure of the HDG matrix can be seen in this Figure.

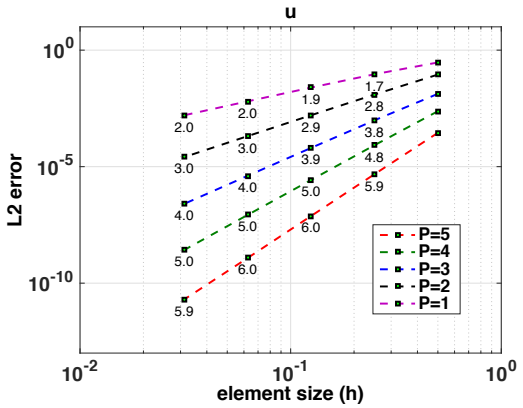
Five different meshes and five different orders of approximation are used to compute the approximated solution and the $L2$ norm of the error for entire domain. These errors for the solution u_h and for the post-process solution u_h^* are presented in Figures 2.3(c) and 2.3(d) respectively. These results demonstrate that the method is capable of achieving optimal convergence rate for solution and superconvergence for post-processed solution. The solution u_h converges optimally with order $p + 1$ for $p = 1, 2, 3, 4, 5$ and the post-processed solution u_h^* converges at the rate of $p + 2$ for polynomials of same degrees, proving the excellent agreement with expectations. The decrement of $L2$-error with increase of degrees of freedom of the problem are shown for both solution and post-processed solution in Figures 2.3(e) and 2.3(f). For the same level of error, high-order approximations need less DOFs, hence are more efficient than low-order approximations.



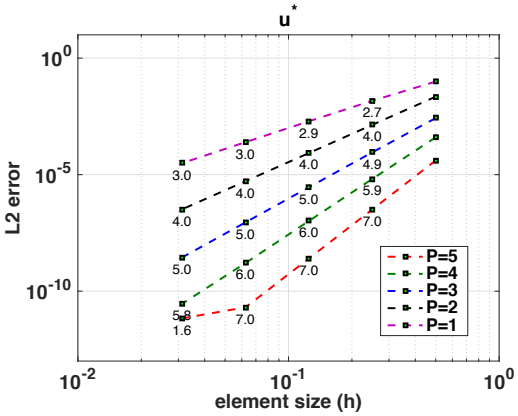
(a) Exact solution



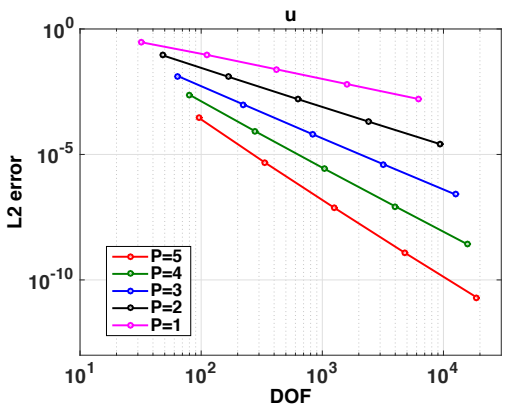
(b) The sparsity pattern of HDG matrix



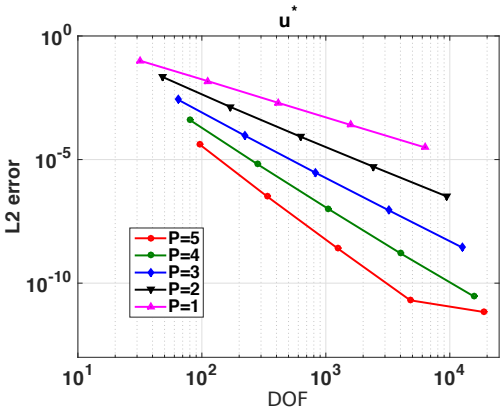
(c) Convergence of u_h



(d) Convergence of u_h^*



(e) Convergence of u_h



(f) Convergence of u_h^*

Figure 2.3: Example 1: Smooth convection-diffusion problem

Example 2: Convection-diffusion problem with boundary layer

Second example, from Cockburn et al. [2009b], is the solution of a steady convection-diffusion problem in $\Omega = (0,1) \times (0,1)$ with Dirichlet boundary conditions $g_D = 0$ on $\partial\Omega$. Convection and diffusion coefficients are $\boldsymbol{c} = (25, 25)$ and $k = 1$, respectively and the source term f is chosen such that the exact solution is

$$u(x,y) = xy\frac{(1 - e^{(x-1)c_x})(1 - e^{(y-1)c_y})}{(1 - e^{-c_x})(1 - e^{-c_y})} \tag{2.22}$$

where c_x and c_y are components of convection velocity. A boundary layer forms

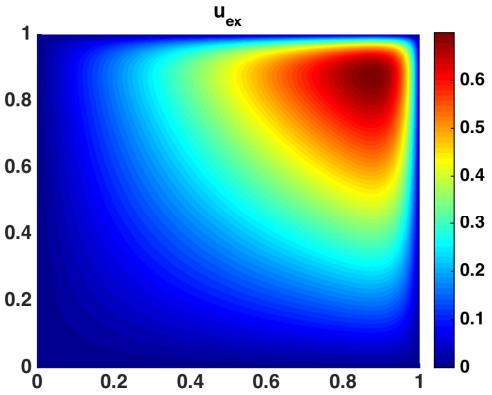
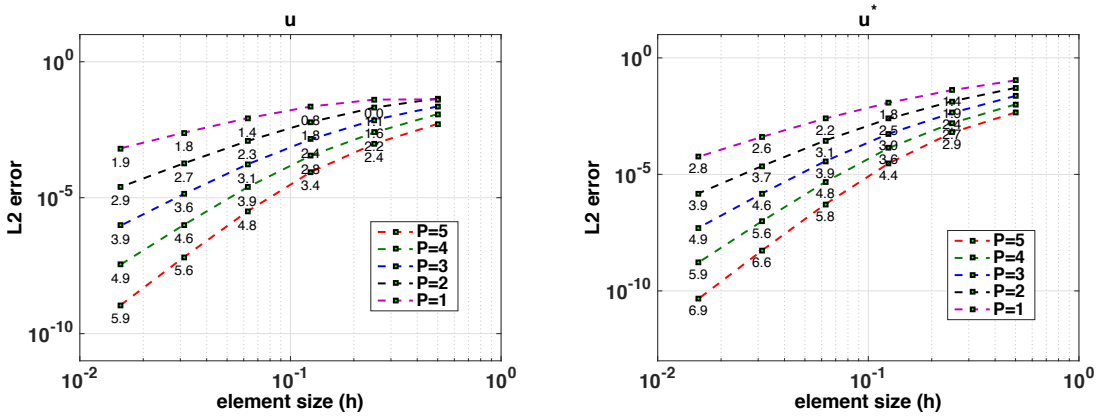


Figure 2.4: Example 2: Solution



(a) Convergence of u_h (b) Convergence of u_h^*

Figure 2.5: Example 2: Convergence plots for the solution u_h and post-process solution u_h^*

near top right corner, see Figure 2.4, which may not be captured correctly for coarse

meshes. The convergence plots in Figure 2.5 (produced using the same five mesh and orders of approximation as example 2.1), show that, although $L2$-norm errors of the solution u_h and post-processed solution u_h^* are high for coarse meshes, for finer meshes, the method is capable of obtaining optimal convergence rates of $p+1$ for u_h and superconvergence of order $p+2$ for post-processed solution u_h^*.

2.2 High-order HDG for compressible Navier-Stokes equations

Navier-Stokes equations are the most fundamental equations of fluid dynamics and aerodynamics. These equations describe the flow of Newtonian liquids and gases, see Riedmann [2009], Cao [2005], and they describe the physics of so many problems from weather and ocean currents, to flows in arteries and engines. In particular, solving the compressible Navier-Stokes equations, is of great importance in lots of engineering fields, like aerospace applications. Although known for more then 100 years, in their complete form, theses equations are very difficult to solve, see White [2005]. On the other hand, from mathematical point of view, the existence of solution and smoothness of these equations is one of most important open problems of mathematics. Therefore, there is a big interest in finding and developing more efficient, more robust, more accurate and faster methods, to solve them numerically.

In this section, the HDG formulation for unsteady compressible Navier-Stokes equations in conservative form is presented and applied to several problems of compressible viscus flow regime and its suitability to perform numerical simulations of such physical phenomena is investigated.

Let $\Omega \in \mathbb{R}^{sd}$ be an open bounded domain, where sd is space dimension, with boundary $\partial\Omega$. The dimensionless form of unsteady compressible Navier-Stokes equations, in conservative form and without body force in the domain $\Omega \times]0,T[$, is

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) &= 0, \\ \frac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v}) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} &= 0, \\ \frac{\partial \rho E}{\partial t} + \boldsymbol{\nabla} \cdot (\rho E \boldsymbol{v}) - \boldsymbol{\nabla} \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{v} - \boldsymbol{q}) &= 0, \end{aligned} \tag{2.23}$$

The equations above are conservation of mass (continuity), momentum (Newton's second law) and energy (first law of thermodynamics) in which ρ, $\boldsymbol{v}$ and E are den-

sity, velocity and total energy density of the fluid respectively. $\boldsymbol{\sigma}$ and $\boldsymbol{q}$ are the Cauchy viscous stress tensor and heat flux vector. To close the system, the following constitutive equations are added

$$\begin{aligned}
\boldsymbol{\sigma} &= -p\boldsymbol{I} + \boldsymbol{\tau}, && \text{Cauchy stress tensor} \\
p &= (\gamma - 1)\rho(E - \|\boldsymbol{v}\|^2/2), && \text{Eq. of state for perfect gas} \\
\boldsymbol{\tau} &= \frac{1}{Re_\infty}[\mu(\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T) + \lambda(\boldsymbol{\nabla} \cdot \boldsymbol{v})\boldsymbol{I}], && \text{Viscous part of stress} \\
\boldsymbol{q} &= -\frac{\mu}{Re_\infty Pr(\gamma - 1)M_\infty^2}\boldsymbol{\nabla} T, && \text{Fourier law of heat conduction} \\
T &= \frac{1}{C_v}(E - \|\boldsymbol{v}\|^2/2), && \text{Calorically perfect gas}
\end{aligned} \tag{2.24}$$

where μ, λ, γ, C_v, Re, Pr and M are viscosity coefficient, bulk viscosity coefficient, specific heats ratio, specific heat in constant volume, Reynolds number, Prandtl number and Mach number respectively, and ∞ denotes free-stream conditions; see White [2005], Anderson [2010] for more information. The compressible Navier-Stokes equations in our preferred applications (mainly external flows) have non-slip boundary condition at wall, and inflow/outflow boundary conditions at far fields, see appendix B for details.

2.2.1 HDG formulation

The time-dependent compressible Navier-Stokes, equations (2.23) with source term, can be written in vector form as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F}_c(\boldsymbol{U}) - \boldsymbol{\nabla} \cdot \boldsymbol{F}_d(\boldsymbol{U}, \boldsymbol{\nabla U}) = \boldsymbol{f}, \text{ in } \Omega \times]0, T[\,, \tag{2.25}$$

where $\boldsymbol{U}$ is the vector of conserved variables and $\boldsymbol{F}_c$ and $\boldsymbol{F}_d$ are the convective and diffusive parts of flux respectively

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho\boldsymbol{v} \\ \rho E \end{bmatrix}, \qquad \boldsymbol{F}_c = \begin{bmatrix} \rho\boldsymbol{v} \\ \rho\boldsymbol{v} \otimes \boldsymbol{v} \\ \rho E\boldsymbol{v} \end{bmatrix}, \qquad \boldsymbol{F}_d = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\sigma} \\ \boldsymbol{\sigma} \cdot \boldsymbol{v} - \boldsymbol{q} \end{bmatrix} \tag{2.26}$$

and $\boldsymbol{f}$ is the possible body force. As one may notice, $\boldsymbol{F}_c$ is just function of solution $\boldsymbol{U}$, while $\boldsymbol{F}_d$ is a function of both solution $\boldsymbol{U}$ and it gradient $\boldsymbol{\nabla U}$.

Following the DG approach, the equations (2.25) can be rewritten as a system of first order PDEs, introducing the new variable $\boldsymbol{Q}$ corresponding to the gradient of the

vector of conserved variables $\boldsymbol{U}$. Taking into the account the discontinuities of the approximation spaces between elements, the compressible Navier-Stokes equations, in a *discrete domain* with elements Ω_e, can be expressed as

$$\left.\begin{aligned} \boldsymbol{Q} - \boldsymbol{\nabla} \boldsymbol{U} &= 0 \\ \frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F}_c(\boldsymbol{U}) - \boldsymbol{\nabla} \cdot \boldsymbol{F}_d(\boldsymbol{U}, \boldsymbol{Q}) &= \boldsymbol{f} \end{aligned}\right\} \qquad \text{in } \Omega_e \times]0, T[\,, \quad e = 1, \cdots, n_{el} \tag{2.27}$$

$$[\![\boldsymbol{U} \otimes \boldsymbol{n}]\!] = 0 \qquad \text{on } \Gamma \setminus \partial\Omega \times]0, T[\,, \tag{2.28}$$

$$[\![(\boldsymbol{F}_c - \boldsymbol{F}_d) \cdot \boldsymbol{n}]\!] = 0 \qquad \text{on } \Gamma \setminus \partial\Omega \times]0, T[\,, \tag{2.29}$$

where $\boldsymbol{n}$ is unitary outward normal vector. Equations (2.28) and (2.29) impose the continuity of the conservative variables and the normal component of the flux across the interior faces.

Again, the main idea of the HDG method for compressible Navier-Stokes equations is the introduction of an approximation of the trace of conserved variables $\boldsymbol{U}$ on the mesh skeleton Γ, which is

$$\hat{\boldsymbol{U}} = \begin{bmatrix} \hat{\rho} \\ \widehat{\rho \boldsymbol{v}} \\ \widehat{\rho E} \end{bmatrix} \tag{2.30}$$

This new variable allows to express a local problem in each element Ω_e, corresponding to the compressible Navier-Stokes equations (2.27) with Dirichlet boundary conditions

$$\boldsymbol{U} = \hat{\boldsymbol{U}} \text{ on } \partial\Omega_e \tag{2.31}$$

According to the procedure for convection-diffusion equation in section 2.1, and using the same finite element spaces defined in (2.12), the HDG discretization of the local problem (2.27), becomes: find $(\boldsymbol{Q}_h, \boldsymbol{U}_h) \in [\mathscr{W}_h^p]^{(n_{sd}+2)n_{sd}} \times [\mathscr{W}_h^p]^{(n_{sd}+2)}$ such that

$$\begin{aligned} (\boldsymbol{Q}_h, \boldsymbol{z})_{\Omega_e} + (\boldsymbol{U}_h, \boldsymbol{\nabla} \cdot \boldsymbol{z})_{\Omega_e} - \langle \hat{\boldsymbol{U}}_h, \boldsymbol{z} \cdot \boldsymbol{n} \rangle_{\partial\Omega_e} &= 0, \\ (\frac{\partial \boldsymbol{U}_h}{\partial t}, \boldsymbol{r})_{\Omega_e} - (\boldsymbol{F}_c(\boldsymbol{U}_h) - \boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h), \boldsymbol{\nabla} \boldsymbol{r})_{\Omega_e} + \langle (\boldsymbol{F}_c(\boldsymbol{U}_h) \widehat{-\boldsymbol{F}_d}(\boldsymbol{U}_h, \boldsymbol{Q}_h)) \cdot \boldsymbol{n}, \boldsymbol{r} \rangle_{\partial\Omega_e} \\ = (\boldsymbol{f}, \boldsymbol{r})_{\Omega_e}, \end{aligned} \tag{2.32}$$

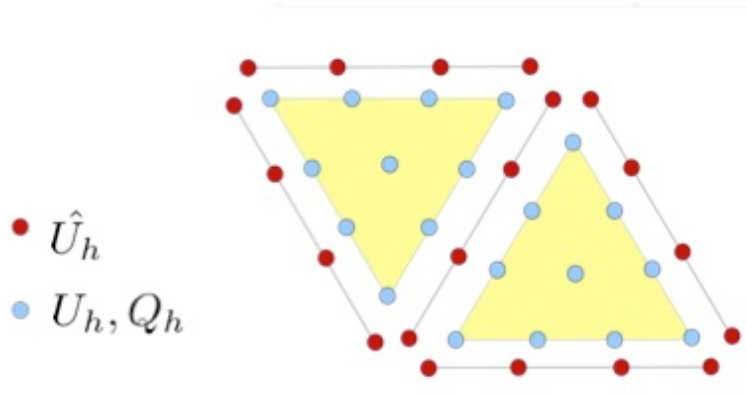


Figure 2.6: Spaces of solutions

for all $(\boldsymbol{z}, \boldsymbol{r}) \in [\mathscr{W}_h^p]^{(n_{sd}+2)n_{sd}} \times [\mathscr{W}_h^p]^{(n_{sd}+2)}$, for $e = 1, ..., n_{el},$. In these equations, the continuity of vector of conserved variables $\boldsymbol{U}$, is weakly imposed by the fact that trace variable $\hat{\boldsymbol{U}}$ is single valued on each face in the mesh skeleton Γ. The definition of numerical flux is taken from the work of Peraire et al. [2010], as

$$(\boldsymbol{F}_c(\boldsymbol{U}_h) \widehat{-\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h)}) \cdot \boldsymbol{n} = (\boldsymbol{F}_c(\hat{\boldsymbol{U}}_h) - \boldsymbol{F}_d(\hat{\boldsymbol{U}}_h, \boldsymbol{Q}_h)) \cdot \boldsymbol{n} + \boldsymbol{S}(\boldsymbol{U}_h - \hat{\boldsymbol{U}}_h), \tag{2.33}$$

where $\boldsymbol{S}$ is the stabilization matrix, considered as

$$\boldsymbol{S} = \begin{bmatrix} 0 & & & \\ & \frac{1}{Re} & & \\ & & \frac{1}{Re} & \\ & & & \frac{1}{(\gamma-1)M_\infty^2 RePr} \end{bmatrix} \tag{2.34}$$

Replacing the definition of the numerical flux (2.34), in the local problem (2.32), and in the weak form of conservativity condition (2.29), the HDG discretization of the compressible Navier-Stokes equations leads to the following problem: find $(\boldsymbol{Q}_h, \boldsymbol{U}_h, \hat{\boldsymbol{U}}_h) \in [\mathscr{W}_h^p]^{(n_{sd}+2)n_{sd}} \times [\mathscr{W}_h^p]^{(n_{sd}+2)} \times [\mathscr{M}_h^p]^{(n_{sd}+2)}$ such that

$$\left.\begin{aligned} (\boldsymbol{Q}_h, \boldsymbol{z})_{\Omega_e} + (\boldsymbol{U}_h, \boldsymbol{\nabla} \cdot \boldsymbol{z})_{\Omega_e} - \langle \hat{\boldsymbol{U}}_h, \boldsymbol{z} \cdot \boldsymbol{n} \rangle_{\partial\Omega_e} &= 0, \\ (\frac{\partial \boldsymbol{U}_h}{\partial t}, \boldsymbol{r})_{\Omega_e} - (\boldsymbol{F}_c(\boldsymbol{U}_h), \boldsymbol{\nabla} \boldsymbol{r})_{\Omega_e} + (\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h), \boldsymbol{\nabla} \boldsymbol{r})_{\Omega_e} + \langle \boldsymbol{F}_c(\hat{\boldsymbol{U}}_h) \cdot \boldsymbol{n}, \boldsymbol{r} \rangle_{\partial\Omega_e} & \\ - \langle \boldsymbol{F}_d(\hat{\boldsymbol{U}}_h, \boldsymbol{Q}_h) \cdot \boldsymbol{n}, \boldsymbol{r} \rangle_{\partial\Omega_e} + \langle \boldsymbol{S}\boldsymbol{U}_h, \boldsymbol{r} \rangle_{\partial\Omega_e} - \langle \boldsymbol{S}\hat{\boldsymbol{U}}_h, \boldsymbol{r} \rangle_{\partial\Omega_e} &= (\boldsymbol{f}, \boldsymbol{r})_{\Omega_e}, \end{aligned}\right\} \tag{2.35}$$

for $e = 1, \cdots, n_{el}$, and

$$\sum_{e=1}^{n_{el}} \langle (\boldsymbol{F}_c(\hat{\boldsymbol{U}}_h) - \boldsymbol{F}_d(\hat{\boldsymbol{U}}_h, \boldsymbol{Q}_h)) \cdot \boldsymbol{n} + \boldsymbol{S}(\boldsymbol{U}_h - \hat{\boldsymbol{U}}_h), \boldsymbol{\mu} \rangle_{\partial\Omega_e \setminus \partial\Omega} = 0, \tag{2.36}$$

for all $(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{\mu}) \in [\mathscr{W}_h^p]^{(n_{sd}+2)n_{sd}} \times [\mathscr{W}_h^p]^{(n_{sd}+2)} \times [\mathscr{M}_h^p]^{(n_{sd}+2)}$.

Implementing different boundary conditions for HDG discretization of compressible Navier-Stokes equations can be tricky, and some decisions are made in this work. In general, the boundary conditions are computed using the interior solution $\boldsymbol{U}_h$, gradient of solution $\boldsymbol{Q}_h$, and boundary condition information (free steam condition $_\infty$), and they are imposed on the trace of the solution on the boundary. For the purpose of this thesis, different boundary conditions are implemented for HDG discretization; inflow, outflow, no-slip wall and symmetry plane boundary conditions, and the details are presented in appendix B.

The HDG discrete problem defined by (2.35) and (2.36) is a system of Differential Algebraic Equations (DAE), which can be discretized in time with an implicit time integrator, such as backward Euler or Crank-Nicolson method. Time discretization of (2.35) and (2.36) leads to *a non-linear system of equations* at each time step, to compute the solution at time t^{n+1} from the solution of time t^n. Here, the non-linear system has been linearized using the Newton-Raphson method.

To implement the Newton-Raphson method the residual form of the equations are considered. To do so, the discretization of local problem (2.35), and global problem (2.36), are written as

$$R(\boldsymbol{Q}_h, \boldsymbol{U}_h, \hat{\boldsymbol{U}}_h) = \begin{bmatrix} R_Q(\boldsymbol{Q}_h, \boldsymbol{U}_h, \hat{\boldsymbol{U}}_h) \\ R_U(\boldsymbol{Q}_h, \boldsymbol{U}_h, \hat{\boldsymbol{U}}_h) \\ R_{\hat{U}}(\boldsymbol{Q}_h, \boldsymbol{U}_h, \hat{\boldsymbol{U}}_h) \end{bmatrix} = \boldsymbol{0} \tag{2.37}$$

In every time step, the initial guess is the solution of the previous time, that is

$$\begin{bmatrix} {}^0\boldsymbol{Q}_h^{n+1} \\ {}^0\boldsymbol{U}_h^{n+1} \\ {}^0\hat{\boldsymbol{U}}_h^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_h^n \\ \boldsymbol{U}_h^n \\ \hat{\boldsymbol{U}}_h^n \end{bmatrix} \tag{2.38}$$

Then, given the approximation $({}^k\boldsymbol{Q}_h^{n+1}, {}^k\boldsymbol{U}_h^{n+1}, {}^k\hat{\boldsymbol{U}}_h^{n+1})$, the new approximation is computed as

$$({}^{k+1}\boldsymbol{Q}_h^{n+1}, {}^{k+1}\boldsymbol{U}_h^{n+1}, {}^{k+1}\hat{\boldsymbol{U}}_h^{n+1}) = ({}^k\boldsymbol{Q}_h^{n+1}, {}^k\boldsymbol{U}_h^{n+1}, {}^k\hat{\boldsymbol{U}}_h^{n+1}) + (\delta\boldsymbol{Q}^{n+1}, \delta\boldsymbol{U}^{n+1}, \delta\hat{\boldsymbol{U}}^{n+1}) \tag{2.39}$$

where the increments in the values of unknowns, $\delta\boldsymbol{Q}, \delta\boldsymbol{U}$ and $\delta\hat{\boldsymbol{U}}$ are are the solution of the system

$$\begin{bmatrix} \frac{\partial R_{\boldsymbol{Q}}}{\partial \boldsymbol{Q}} & \frac{\partial R_{\boldsymbol{Q}}}{\partial \boldsymbol{U}} & \frac{\partial R_{\boldsymbol{Q}}}{\partial \hat{\boldsymbol{U}}} \\ \frac{\partial R_{\boldsymbol{U}}}{\partial \boldsymbol{Q}} & \frac{\partial R_{\boldsymbol{U}}}{\partial \boldsymbol{U}} & \frac{\partial R_{\boldsymbol{U}}}{\partial \hat{\boldsymbol{U}}} \\ \frac{\partial R_{\hat{\boldsymbol{U}}}}{\partial \boldsymbol{Q}} & \frac{\partial R_{\hat{\boldsymbol{U}}}}{\partial \boldsymbol{U}} & \frac{\partial R_{\hat{\boldsymbol{U}}}}{\partial \hat{\boldsymbol{U}}} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{Q}^{n+1} \\ \delta\boldsymbol{U}^{n+1} \\ \delta\hat{\boldsymbol{U}}^{n+1} \end{bmatrix} = \begin{bmatrix} -R_{\boldsymbol{Q}}({}^{k}\boldsymbol{Q}_h^{n+1}, {}^{k}\boldsymbol{U}_h^{n+1}, {}^{k}\hat{\boldsymbol{U}}_h^{n+1}) \\ -R_{\boldsymbol{U}}({}^{k}\boldsymbol{Q}_h^{n+1}, {}^{k}\boldsymbol{U}_h^{n+1}, {}^{k}\hat{\boldsymbol{U}}_h^{n+1}) \\ -R_{\hat{\boldsymbol{U}}}({}^{k}\boldsymbol{Q}_h^{n+1}, {}^{k}\boldsymbol{U}_h^{n+1}, {}^{k}\hat{\boldsymbol{U}}_h^{n+1}) \end{bmatrix} \tag{2.40}$$

Note that superscripts k and n, correspond to the iteration of non-linear solver and time steps, respectively.

In this linear system, the two first blocs of equations, corresponding to (2.35), can be solved element-by-element to express the solution at each element Ω_e in terms of the increment of trace variable, $\delta\hat{\boldsymbol{U}}_h$. Then, replacing in (2.36) yields a global system of equations involving only $\delta\hat{\boldsymbol{U}}_h$ as

$$\mathbb{K}\delta\hat{\boldsymbol{U}}_h = \mathbb{F} \tag{2.41}$$

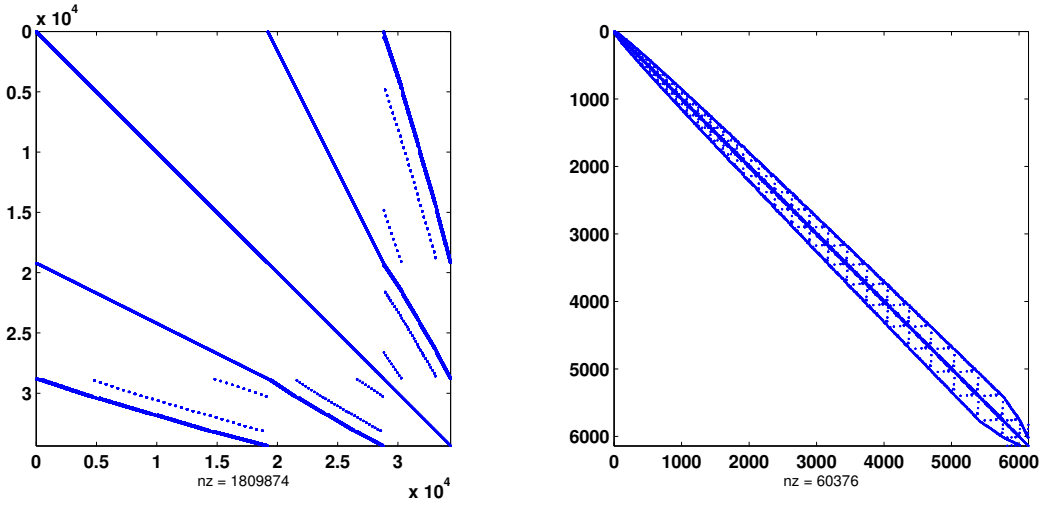
So the final system only includes the DOFs of $\hat{\boldsymbol{U}}$ and the system is compact and with usual block structure of HDG. Then the equations corresponding to the local problem, can be used for an element-by-element reconstruction of the vector of conserved variables and its gradient. Figure 2.7 shows the effect of hybridization on the size and structure of final linear system.

Some details of the implementation of HDG method and the boundary condition for compressible Navier-Stokes can be found in appendices C.2 and B, respectively.

2.2.2 Numerical results

This section presents numerical results for the solution of steady compressible Navier-Stokes equations with HDG. To solve the steady state problem, either a steady solver, or an unsteady one (time relaxation) can be used. For the steady solver, the temporal term in (2.35) is neglected and Newton-Raphson iterative solver is used. To reach the solution far from initial guess, i.e. high Reynolds and high Mach numbers, usually a continuation method is needed. On the other hand, the unsteady solver uses a gradual change until it reaches the final solution, which needs more steps but is more robust.

First, accuracy and convergence of the solver is studied and the ability of the non-linear solver to reach of the solution is demonstrated. Then, a laminar viscous flow around a NACA 0012 airfoil is considered in subsonic and transonic regimes and



(a) Matrix before condensation (b) HDG matrix

Figure 2.7: Condensation of HDG matrix

the efficiency of high-order methods is investigated. These test cases have been used in literature to verify the solution of compressible viscous laminar flow, for instance see Bassi and Rebay [1997a], Wang and Anderson [2012], Luo et al. [2010], Borrel and Ryan [2012], Hartmann [2005b], Hartmann and Houston [2009]. Finally, in a Carter plate example, the solutions from steady and unsteady solvers are compared to assure their accuracy.

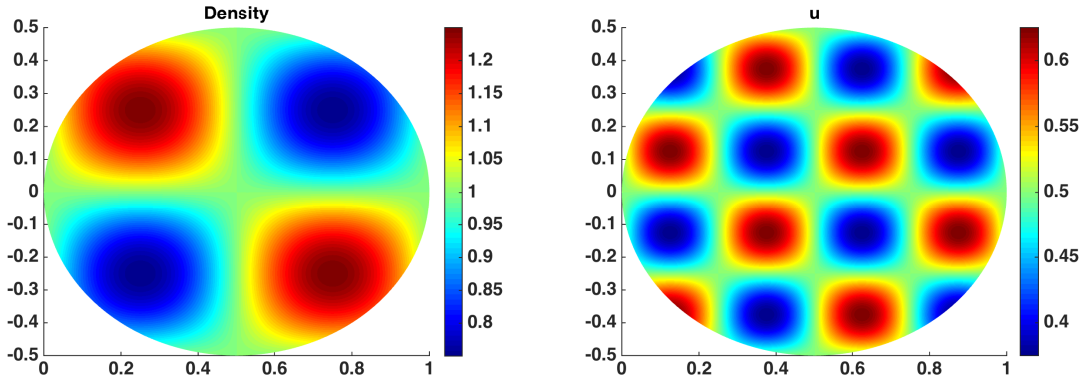
Viscous flow in a circle

The goal of this example, from Wang and Anderson [2012], is to verify the accuracy of the HDG solver through convergence of error with respect to mesh size. To do so, the two-dimensional steady compressible Navier-Stokes equations are solved in a circular computational domain of radius 0.5, centred at $(0.5, 0)$. An inhomogeneous source term, and Dirichlet boundary conditions are set so that density, velocity and total energy are

$$\begin{aligned} \rho_{ex} &= \rho_0(1 + \sin(\pi x)\cos(\pi x)\sin(\pi y)\cos(\pi y)) \\ u_{ex} &= u_0(1 + \sin(\kappa\pi x)\cos(\kappa\pi x)\sin(\kappa\pi y)\cos(\kappa\pi y)) \\ v_{ex} &= v_0(1 + \sin(\kappa\pi x)\cos(\kappa\pi x)\sin(\kappa\pi y)\cos(\kappa\pi y)) \\ (\rho E)_{ex} &= E_0(1 + \sin(\pi x)\sin(\pi x)\sin(\pi y)\sin(\pi y)) \end{aligned} \tag{2.42}$$

where κ is the frequency of the velocity, here considered $\kappa = 2$. The coefficients (ρ_0, u_0, v_0, E_0) are set to $(1, 0.5, 0, 5, 3)$ and Reynolds number is $Re_\infty = 1$.

Figures 3.9(a) and 3.9(b) show density and x-component of velocity, respectively, in a computational mesh of 1916 unstructured triangular elements of size $h = 0.0317$ and polynomials of degree $p = 4$. As expected, the frequency of the velocity is twice of the frequency of density. Four different meshes of $28, 122, 484$ and 1916 triangular elements, with order of approximations $p = 1, 2, 3$ and 4 are computed, and evolution is shown in Figure 2.9(a), showing optimal convergence rates are obtained. Figure 2.9(b), shows the evolution of error vs. square root of DOF for the same meshes and degree of approximations. Here, again the efficiency of high-order approximation can be seen: to reach the same accuracy, much less DOF are needed for $p = 4$ in comparison with $p = 2,\ 3$, let alone $p = 1$.

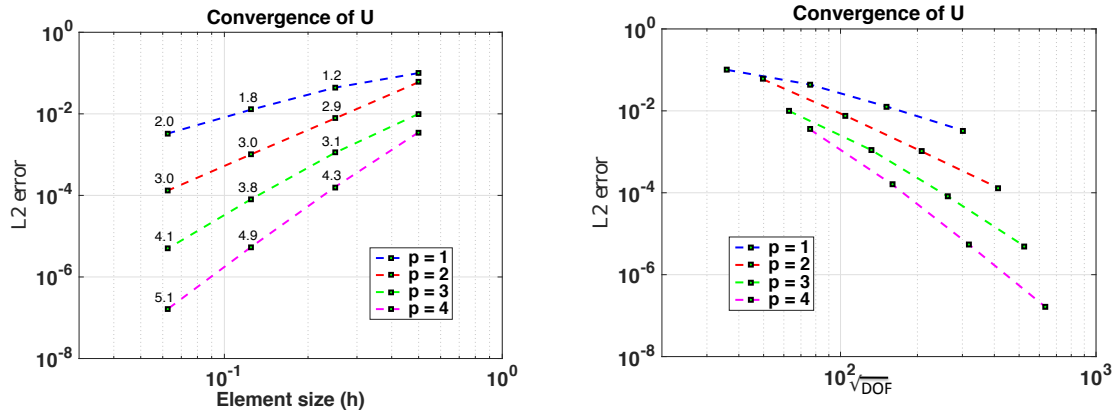


(a) density (b) x-velocity

Figure 2.8: HDG solution of a viscous flow in a circle, 1916 elements and $p = 4$

Laminar flow around NACA 0012 airfoil

In this numerical test, from Hartmann [2005b], the goal is to investigate the ability of HDG method to compute the solution for compressible viscous external flows using large high-order elements. The steady state viscous laminar flow around NACA 0012 airfoil at the angle of attack $\alpha = 0$ is computed with Reynolds number $Re_\infty = 5000$, (which is quite high for laminar range) and free stream Mach number of $M_\infty = 0.5$. The airfoil surface is no-slip boundary with adiabatic wall condition (zero heat flux). The far-field boundary condition is implemented as subsonic inflow/outflow boundary conditions of Euler equations. The rationale for using Euler far-field boundary



(a) Error vs. mesh size

(b) Error vs. DOF

Figure 2.9: Evolution of error

conditions is the uniformity of the flow characteristics far from the airfoil, which leads to zero gradient of the conserved variables, hence to zero shear stress and heat conduction. The schematic drawing of the problem is shown in Figure 2.10.

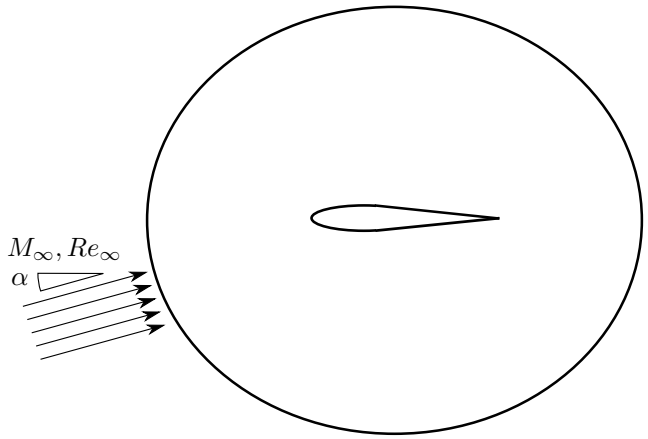


Figure 2.10: Schematic drawing of airfoil problem

The analytical expression for parametrization of upper and lower part of symmetric NACA 0012 airfoil is

$$y = \pm\frac{t_k}{0.2}(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4), \text{ for } x \in [0,1] \tag{2.43}$$

where $t_k = 0.12$ is the maximum thickness of the airfoil. The computational meshes are defined following the procedure in Giorgiani et al. [2014], which leads to meshes

refined around the airfoil surface to capture boundary layer, and also refined near leading edge and trailing edge of the airfoil.

First, Figure 2.11 shows the distribution of flow parameters over the airfoil with a zoom of the mesh with 640 elements of degree $p = 6$, used for the computations in Figure 2.11(a). The computational domain is a circle with radius of 10 times the airfoil cord. The steady solver is used here and the convergence of non-linear solver is demonstrated in Figure 2.11(b), with the expected quadratic convergence. It should be noted that for a steady problem, $|d\boldsymbol{U}|/|\boldsymbol{U}|$ denotes the normalized increment of the vector of conserved variable in each iteration of non-linear solver. Distribution of Mach number and pressure are shown in Figures 2.11(c) and 2.11(d) respectively. The flow around the leading edge and trailing edge are shown in Figures 2.11(e) and 2.11(f), respectively, showing the detachment of flow from airfoil surface.

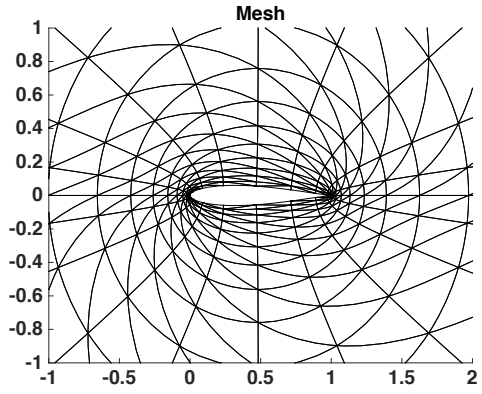
Distribution of pressure coefficient on the airfoil is shown in Figure 2.12. It is in good agreement with a reference solution of Hartmann [2005b] with even much coarser mesh of 640 elements with degree $p = 6$.

Table 2.1, shows a comparison of errors of aerodynamic coefficients (lift and drag coefficients) for four different meshes and degree of approximation: three meshes of 640 elements with degree $p = 2$, $p = 4$ and $p = 6$, and a mesh of 1944 elements with degree $p = 2$. Lift and Drag coefficients of the airfoil are computed from distribution of pressure and shear stress over the upper and lower wall of airfoil, and then compared with reference values from Hartmann [2005b]. It is important noting that increasing the DOFs of the problem increases the accuracy of the method, but high-order approximation of $p = 6$ produces more accurate solution than low-order approximation of $p = 2$, despite having less DOFs than quadratic approximation on a finer mesh.

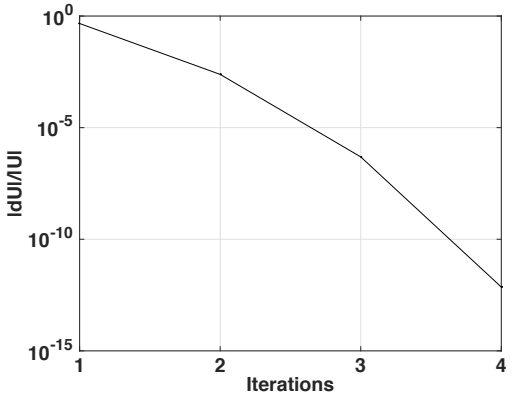
Mesh NofEl	**Approx. p**	**DOF**	**Error Cl**	**Cd**
640	2	11760	9.14e-03	2.53e-02
640	4	19600	5.83e-04	2.50e-03
640	6	27440	3.36e-05	7.34e-04
1944	2	35424	1.56e-04	3.27e-03

Table 2.1: Comparison of error of aerodynamics coefficients with HDG, $Re = 5000, M_\infty = 0.5, \alpha = 0$

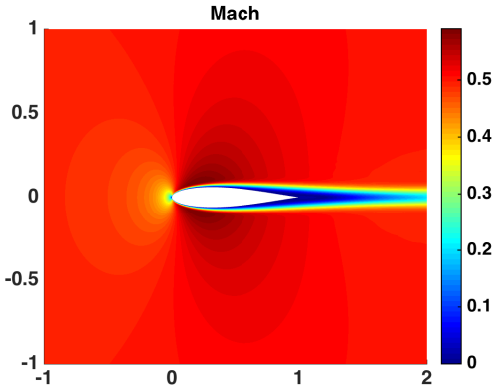
To test the code for the steady state solution of *transonic* viscous laminar flow, The far-field condition is changed to the angle of attack $\alpha = 5^o$, Reynolds number



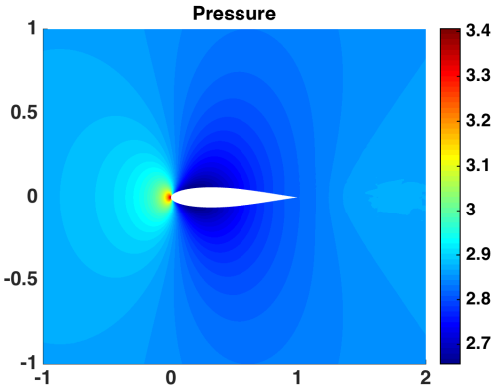
(a) Computational mesh: 640 elements p = 6



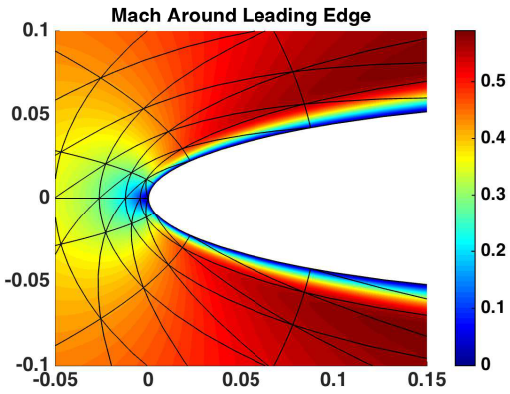
(b) Convergence



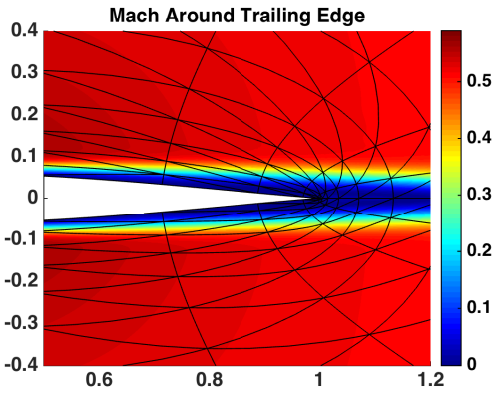
(c) Mach



(d) pressure



(e) Mach around leading edge



(f) Mach around trailing edge

Figure 2.11: Laminar subsonic flow around NACA 0012, $M_\infty = 0.5$, $Re_\infty = 5000$, $\alpha = 0$

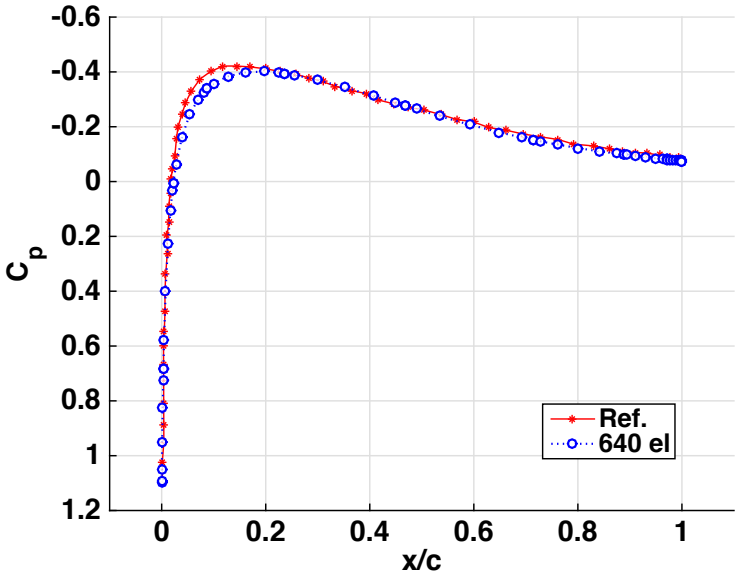


Figure 2.12: Distribution of pressure coefficient over a subsonic airfoil

$Re_\infty = 1000$ and Mach number $M = 0.85$. This high Mach number and angle of attack leads to formation of some low supersonic flow regions specially over the airfoil. The same NACA 0012 and the same geometry is considered with the computational mesh of 640 elements with degree $p = 6$.

The Mach number distribution is shown in Figure 2.13 and it can be observed that a supersonic region is starting to appear above the front part of the airfoil. The pressure coefficient distribution is presented in Figure 2.14. Note that the angle of attack leads to non-symmetric flow and higher pressure on the lower part of the airfoil to produce the lift, as expected. In conclusion, numerical experiments in NACA 0012 airfoil example, demonstrate the applicability of high-order HDG for external laminar viscous compressible flow computations.

Flow passing over Carter plate

In this test case, compressible viscous flow passing over an infinitely thin flat plate, at zero angle of attack, is modelled. The goal of this test case is to compare the solution of steady and unsteady solver for the same steady problem. The domain is a rectangle of $[-1, 1] \times [0, 1.25]$, and the first part, $\{x \in [-1, -0.8], y = 0\}$, is a slip wall, and the rest of the surface, $\{x \in [-0.8, 1], y = 0\}$ is the plate, where the boundary is no-slip. This change of boundary condition causes a boundary layer formation,

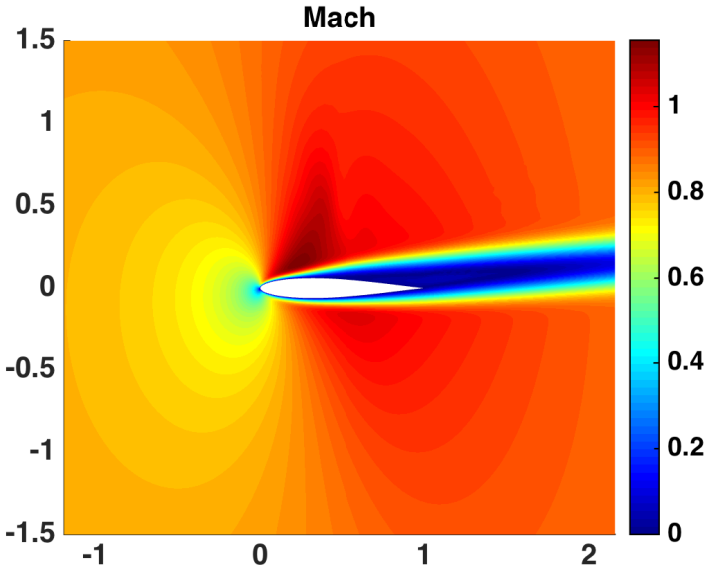


Figure 2.13: Mach over a transonic airfoil

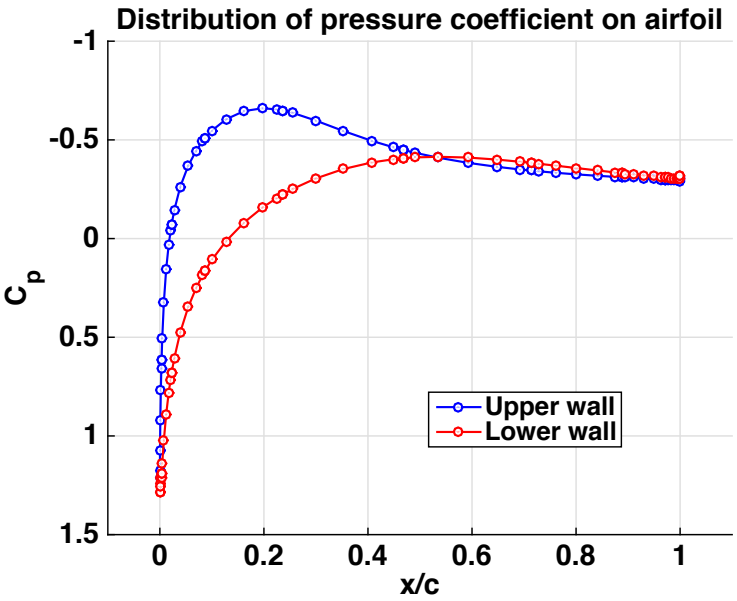


Figure 2.14: Distribution of pressure coefficient over a transonic airfoil

which develops further downstream. The top part of horizontal the domain, $y = 1.2$, is a symmetry boundary, and inflow is at the left, $x = 0$, and outflow is on the right side of domain, $x = 1$. A mesh of 480 element and cubic approximation is used to calculate the solution from both steady and unsteady solver. The Mach number is 0.75 and Reynolds number is 800 for this problem. For the unsteady solver a time step of $\Delta t = 2.5 \times 10^{-2}$ is considered.

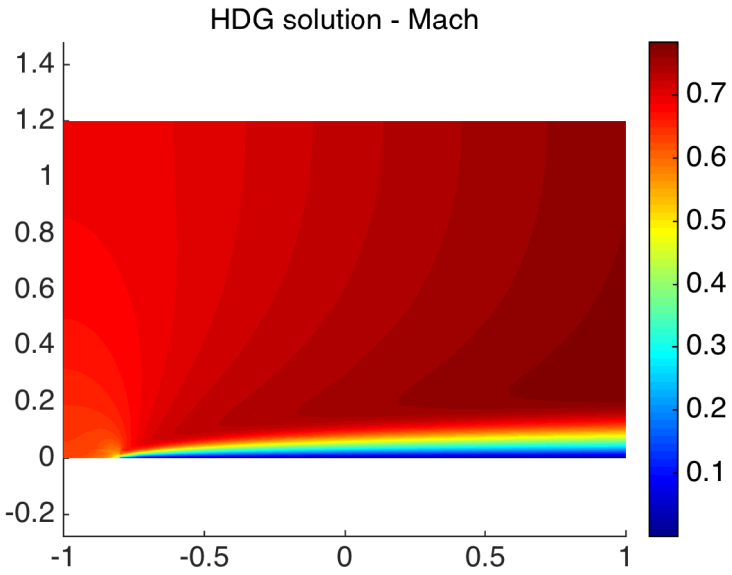
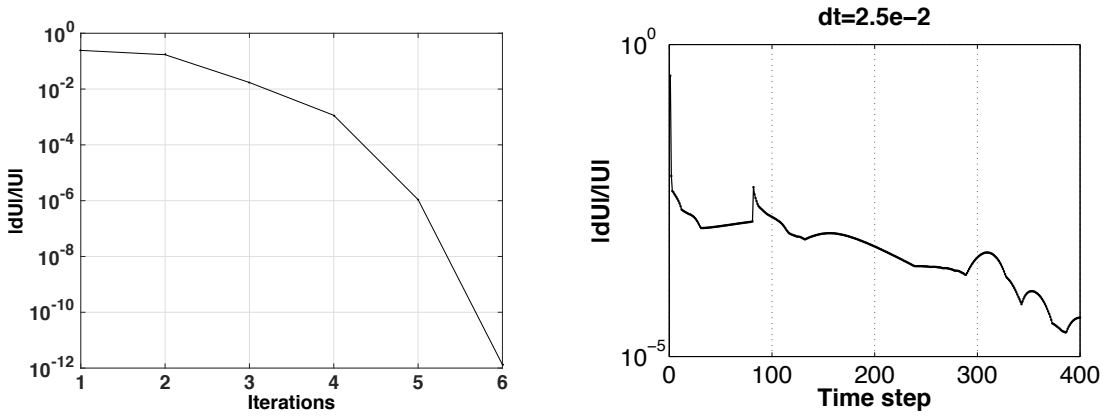


Figure 2.15: flow passing over Carter plate

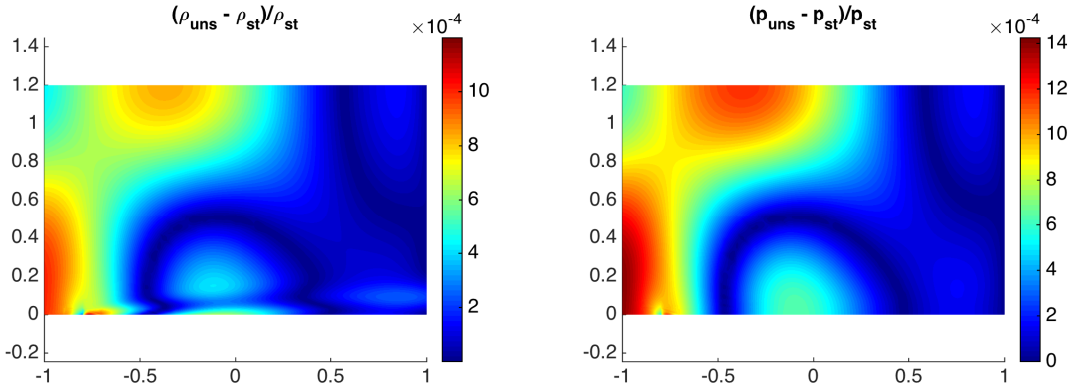
Figure 2.15 shows the distribution of Mach number, and the boundary layer over the flat plate is clear in the solution. Figure 2.16 shows the comparison between steady solver and unsteady one. The convergence of the solution is presented for steady solver in Figure 2.16(a) and for unsteady solver in Figure 2.16(b), and it can be seen that steady state solver needs just few iterations to converge, while using the transient solver although more robust, needs much more CPU time to reach the steady state solution. In the plots, $|d\boldsymbol{U}|/|\boldsymbol{U}|$ represents the normalized increment in each iteration or time step, for steady or unsteady solver, respectively. Then, the normalized difference in the solution for density and pressure are shown in Figures 2.16(c) and 2.16(d) respectively, between two solutions of steady or unsteady solvers. As expected, the solution is the same for the level of accuracy of the solutions, with repetitive error of 10^{-4}.

2.3 A new linearization for compressible Navier-Stokes equations

The compressible Navier-Stokes equations contain several non-linear terms. The Newton-Raphson non-linear solver presented in section 2.2 has quadratic convergence, but fro steady solutions, it is very sensitive to initial guess, and may diverge



(a) Convergence of the solution - steady solver (b) Convergence of the solution - unsteady solver



(c) Density difference (d) Pressure difference

Figure 2.16: Comparison of steady and unsteady solver

for initial guess far from the solution. Thus, to solve the problem at high Reynolds numbers and high Mach numbers, a continuation method must be used, i.e. several intermediate problems are solved by increasing the Reynolds and Mach number, step by step to reach the final solution. However, a more robust approach would be very welcomed to reach final solution, and there are open issues towards development of stable, robust and feasible methods to approximate the solution for compressible Navier-Stokes equations.

Other usual strategy is to use a relaxation method, e.g. to solve the transient problem, to reach the steady state solution. But as seen in numerical examples of section 2.2.2, it may be costly and non-linear solver in each time step may diverge. By solving a transient problem with linearized HDG for compressible Navier-Stokes to avoid solving the non-linear system in each time step, a better approach may be

achieved.

In this section, a method to linearize the unsteady compressible Navier-Stokes equations is proposed, aiming to implement a more robust solver. The strategy is mainly to use an approximated solution from previous time steps in order to reach a linear set of equations in current time step. Even for steady problems, the transient solver is utilized, which relaxes in time, and finally converges to the steady state solution. The idea has been first proposed by Baker [1976] and has been applied to incompressible Navier-Stokes, see for instance Baker et al. [1982], Yang et al. [2009], He and Sun [2007], Liu and Hou [2010], Labovsky et al. [2009] and is known as Linearly Extrapolated Crank-Nicolson time-stepping methods (CNLE). Here the same idea is applied to the solution of compressible Navier-Stokes equations, taking into the account that now the non-linearity is due to convective term, but also due to non-linear terms in both momentum and energy equations. The goal of this section, is to develop such a solver for HDG spatial discretization of compressible Navier-Stokes equations and investigate its ability to perform compressible viscous flow computations.

Let $\Omega \in \mathbb{R}^{sd}$ be an open bounded domain, where sd is spatial dimension, with boundary $\partial\Omega$. We start by replacing the constitutive equations (2.24), in the Navier-Stokes equations (2.23), leading to the following form of the equations for unsteady compressible viscous flow

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) = 0, \\
&\frac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v} + (\gamma - 1)(\rho E - \frac{1}{2}\rho \boldsymbol{v} \cdot \boldsymbol{v})\boldsymbol{I}) - \boldsymbol{\nabla} \cdot (\frac{\lambda}{Re_\infty}(\boldsymbol{\nabla} \cdot \boldsymbol{v})\boldsymbol{I} + 2\frac{\mu}{Re_\infty}\boldsymbol{\nabla}^s \boldsymbol{v}) = 0, \\
&\frac{\partial \rho E}{\partial t} + \boldsymbol{\nabla} \cdot (\gamma \rho E \boldsymbol{v} - (\gamma - 1)\frac{1}{2}(\rho \boldsymbol{v} \cdot \boldsymbol{v})\boldsymbol{v}) \\
&\qquad - \boldsymbol{\nabla} \cdot (\frac{\lambda}{Re_\infty}(\boldsymbol{\nabla} \cdot \boldsymbol{v})\boldsymbol{v} + 2\frac{\mu}{Re_\infty}\boldsymbol{\nabla}^s \boldsymbol{v} \cdot \boldsymbol{v} + \frac{k}{c_v}\boldsymbol{\nabla} E - \frac{k}{c_v}(\boldsymbol{v}\boldsymbol{\nabla}) \cdot \boldsymbol{v}) = 0,
\end{aligned}
\tag{2.44}
$$

in domain $\Omega \times]0, T[$, where $\boldsymbol{\nabla}^s \boldsymbol{v}$ is the strain rate tensor, i.e. $\boldsymbol{\nabla}^s \boldsymbol{v} = \frac{1}{2}(\boldsymbol{\nabla}\boldsymbol{v} + (\boldsymbol{\nabla}\boldsymbol{v})^T)$, and $\boldsymbol{I}$ is identity matrix. Following the HDG approach, new variables, $\boldsymbol{L}$ and E are introduced in order to rewrite the system into a system of first order PDEs. However, note that, here, gradient of velocity and total energy are considered as new variables, instead of the gradient of whole vector of conserved variables, which is used

in standard HDG method. The equations are then

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) = 0, \\
&\frac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v} + (\gamma - 1)(\rho E - \frac{1}{2}\rho \boldsymbol{v} \cdot \boldsymbol{v})\boldsymbol{I}) - \boldsymbol{\nabla} \cdot (\frac{\lambda}{Re_\infty} tr(\boldsymbol{L})\boldsymbol{I} + \frac{\mu}{Re_\infty}(\boldsymbol{L} + \boldsymbol{L}^T)) = 0, \\
&\frac{\partial \rho E}{\partial t} + \boldsymbol{\nabla} \cdot (\gamma \rho E \boldsymbol{v} - (\gamma - 1)\frac{1}{2}(\rho \boldsymbol{v} \cdot \boldsymbol{v})\boldsymbol{v}) \\
&\qquad - \boldsymbol{\nabla} \cdot (\frac{\lambda}{Re_\infty} tr(\boldsymbol{L})\boldsymbol{v} + \frac{\mu}{Re_\infty}(\boldsymbol{L} + \boldsymbol{L}^T) \cdot \boldsymbol{v} + \frac{k}{c_v}\boldsymbol{w} - \frac{k}{c_v}\boldsymbol{L}^T \cdot \boldsymbol{v}) = 0, \\
&\boldsymbol{L} - \boldsymbol{\nabla} \boldsymbol{v} = 0, \\
&\boldsymbol{w} - \boldsymbol{\nabla} E = 0,
\end{aligned} \tag{2.45}$$

in domain $\Omega \times]0, T[$. In equations (2.45) the non-linearities of the equations in terms of the solution can be detected easily and equations (2.45) can be rewritten as

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) = 0, \\
&\frac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v} \otimes \bar{\boldsymbol{v}} + (\gamma - 1)(\rho E - \frac{1}{2}\rho \boldsymbol{v} \cdot \bar{\boldsymbol{v}})\boldsymbol{I}) - \boldsymbol{\nabla} \cdot (\frac{\lambda}{Re_\infty} tr(\boldsymbol{L})\boldsymbol{I} + \frac{\mu}{Re_\infty}(\boldsymbol{L} + \boldsymbol{L}^T)) = 0, \\
&\frac{\partial \rho E}{\partial t} + \boldsymbol{\nabla} \cdot (\gamma \rho E \bar{\boldsymbol{v}} - (\gamma - 1)\frac{1}{2}(\rho \boldsymbol{v} \cdot \bar{\boldsymbol{v}})\bar{\boldsymbol{v}}) \\
&\qquad - \boldsymbol{\nabla} \cdot (\frac{\lambda}{Re_\infty} tr(\boldsymbol{L})\bar{\boldsymbol{v}} + \frac{\mu}{Re_\infty}(\boldsymbol{L} + \boldsymbol{L}^T) \cdot \bar{\boldsymbol{v}} + \frac{k}{c_v}\boldsymbol{w} - \frac{k}{c_v}\boldsymbol{L}^T \cdot \bar{\boldsymbol{v}}) = 0, \\
&\bar{\rho}\boldsymbol{L} - \boldsymbol{\nabla}(\rho \boldsymbol{v}) + \bar{\boldsymbol{v}} \boldsymbol{\nabla} \rho^T = \boldsymbol{0}, \\
&\bar{\rho}\boldsymbol{w} - \boldsymbol{\nabla}(\rho E) + \bar{E} \boldsymbol{\nabla} \rho^T = \boldsymbol{0},
\end{aligned} \tag{2.46}$$

where $\bar{\rho}$,$\bar{\boldsymbol{v}}$ and $\bar{E}$ are the new variables introduced, for which an explicit approximation with values from previous time steps will be used. Using a discretization in time, this approximation will lead to a linear system of equations, (2.46), and no need for non-linear solver. From now on these equations are called *linearized compressible Navier-Stokes* equations.

$$\bar{a}^{n+1} = \frac{3}{2}a^n - \frac{1}{2}a^{n-1} \tag{2.47}$$

2.3.1 HDG formulation

The linearized compressible Navier-Stokes equations (2.46), with body force, can be written in the vector form on continuous domain as

$$\begin{aligned} \frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U} + \boldsymbol{\nabla}\cdot\boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}})\boldsymbol{L} + \boldsymbol{\nabla}\cdot\boldsymbol{A}_{fd_2}\boldsymbol{w} = \boldsymbol{f}, \quad &\text{in } \Omega\times]0,T[\\ \bar{\rho}\boldsymbol{L} - \boldsymbol{\nabla}(\rho\boldsymbol{v}) + \bar{\boldsymbol{v}}\boldsymbol{\nabla}\rho^T = \boldsymbol{0}, \quad &\text{in } \Omega\times]0,T[\\ \bar{\rho}\boldsymbol{w} - \boldsymbol{\nabla}(\rho E) + \bar{E}\boldsymbol{\nabla}\rho^T = \boldsymbol{0}, \quad &\text{in } \Omega\times]0,T[\end{aligned} \tag{2.48}$$

where $\boldsymbol{U}$ is vector of conserved variables, $\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U}$, $\boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}})\boldsymbol{L}$ and $\boldsymbol{A}_{fd_2}\boldsymbol{w}$ are linearized versions of compressible Navier-Stokes fluxes and are linearized using the $\bar{\boldsymbol{v}}$, which is approximated from previous time steps with (2.47). Note that here, the equations are solved for $\boldsymbol{U}$, $\boldsymbol{L}$ and $\boldsymbol{w}$, so vector of conserved variables, gradient of velocity and gradient of total energy are the unknowns in the equations (2.48). In 2D, the fluxes for the linearized equations are given by

$$\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}}) = \begin{bmatrix} \bar{\boldsymbol{v}}_1 & 0 & 0 & 0 \\ \bar{\boldsymbol{v}}_2 & 0 & 0 & 0 \\ 0 & \bar{\boldsymbol{v}}_1 - \frac{\gamma-1}{2}\bar{\boldsymbol{v}}_1 & -\frac{\gamma-1}{2}\bar{\boldsymbol{v}}_2 & \gamma-1 \\ 0 & \bar{\boldsymbol{v}}_2 & 0 & 0 \\ 0 & 0 & \bar{\boldsymbol{v}}_1 & 0 \\ 0 & -\frac{\gamma-1}{2}\bar{\boldsymbol{v}}_1 & \bar{\boldsymbol{v}}_2 - \frac{\gamma-1}{2}\bar{\boldsymbol{v}}_2 & \gamma-1 \\ 0 & -\frac{\gamma-1}{2}\bar{\boldsymbol{v}}_1\cdot\bar{\boldsymbol{v}}_1 & -\frac{\gamma-1}{2}\bar{\boldsymbol{v}}_2\cdot\bar{\boldsymbol{v}}_1 & \bar{\boldsymbol{v}}_1 + \frac{\gamma-1}{2}\bar{\boldsymbol{v}}_1 \\ 0 & -\frac{\gamma-1}{2}\bar{\boldsymbol{v}}_1\cdot\bar{\boldsymbol{v}}_2 & -\frac{\gamma-1}{2}\bar{\boldsymbol{v}}_2\cdot\bar{\boldsymbol{v}}_2 & \bar{\boldsymbol{v}}_2 + \frac{\gamma-1}{2}\bar{\boldsymbol{v}}_2 \end{bmatrix}$$

$$\boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{4}{3}\kappa_1 & 0 & 0 & -\frac{2}{3}\kappa_1 \\ 0 & \kappa_1 & \kappa_1 & 0 \\ 0 & \kappa_1 & \kappa_1 & 0 \\ -\frac{2}{3}\kappa_1 & 0 & 0 & \frac{4}{3}\kappa_1 \\ (\frac{4}{3}\kappa_1+\kappa_2)\bar{\boldsymbol{v}}_1 & \kappa_1\bar{\boldsymbol{v}}_2 & (\kappa_1+\kappa_2)\bar{\boldsymbol{v}}_2 & -\frac{2}{3}\kappa_1\bar{\boldsymbol{v}}_1 \\ -\frac{2}{3}\kappa_1\bar{\boldsymbol{v}}_2 & (\kappa_1+\kappa_2)\bar{\boldsymbol{v}}_1 & \kappa_1\bar{\boldsymbol{v}}_1 & (\frac{4}{3}\kappa_1+\kappa_2)\bar{\boldsymbol{v}}_2 \end{bmatrix}, \qquad \boldsymbol{A}_{fd_2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\kappa_2 & 0 \\ 0 & -\kappa_2 \end{bmatrix} \tag{2.49}$$

where $\kappa_1 = -\frac{\mu}{Re_\infty}$ and $\kappa_2 = \frac{\mu}{Re_\infty Pr(\gamma-1)M_\infty^2 C_v}$.

Taking into the account the discontinuities of the approximation spaces between elements, the linearized compressible Navier-Stokes equations, in a *discrete domain*,

can be expressed as

$$\left.\begin{aligned} \frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U} + \boldsymbol{\nabla}\cdot\boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}})\boldsymbol{L} + \boldsymbol{\nabla}\cdot\boldsymbol{A}_{fd_2}\boldsymbol{w} &= \boldsymbol{f}, \\ \bar{\rho}\boldsymbol{L} - \boldsymbol{\nabla}(\rho\boldsymbol{v}) + \bar{\boldsymbol{v}}\boldsymbol{\nabla}\rho^T &= \boldsymbol{0}, \\ \bar{\rho}\boldsymbol{w} - \boldsymbol{\nabla}(\rho E) + \bar{E}\boldsymbol{\nabla}\rho^T &= \boldsymbol{0}, \end{aligned}\right\} \quad \text{in } \Omega_e, \quad e = 1, \cdots, n_{el} \tag{2.50}$$

$$\begin{aligned} [\![\boldsymbol{U}\otimes\boldsymbol{n}]\!] &= \boldsymbol{0}, \qquad \text{on } \Gamma\setminus\partial\Omega, \\ [\![(\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U} + \boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}})\boldsymbol{L} + \boldsymbol{A}_{fd_2}\boldsymbol{w})\cdot\boldsymbol{n}]\!] &= \boldsymbol{0}, \qquad \text{on } \Gamma\setminus\partial\Omega, \end{aligned} \tag{2.51}$$

The equations (2.50) are defined in each element, and the equations (2.51) impose the continuity of the conserved variables and normal component of the flux across the interior faces.

Following the HDG approach, an approximation of the trace of conserved variables $\boldsymbol{U}$ on the mesh skeleton Γ is introduced, which is $\hat{\boldsymbol{U}} = [\hat{\rho}, \widehat{\rho\boldsymbol{v}}, \widehat{\rho E}]^T$. This new variable allows to express a local problem in each element Ω_e, corresponding to the compressible Navier-Stokes equations (2.50) with Dirichlet boundary conditions precisely as trace variable $\hat{\boldsymbol{U}}$. According to the procedure we did for convection-diffusion equation in section 2.1, and compressible Navier-Stokes in 2.2, and using the same finite element spaces as in (2.12), the HDG discretization of the local problem (2.50) and global problem (2.51) leads to the following problem: to find approximation $(\boldsymbol{U}_h, \boldsymbol{L}_h, \boldsymbol{w}_h, \hat{\boldsymbol{U}}_h) \in [\mathscr{W}_h^p]^{sd+2} \times [\mathscr{W}_h^p]^{sd+2} \times [\mathscr{W}_h^p]^{sd} \times [\mathscr{M}_h^p]^{sd+2}$ such that

$$\left.\begin{aligned} (\frac{\partial \boldsymbol{U}_h}{\partial t}, \boldsymbol{r})_{\Omega_e} - (\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U}_h, \boldsymbol{\nabla}\boldsymbol{r})_{\Omega_e} - (\boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}})\boldsymbol{L}_h, \boldsymbol{\nabla}\boldsymbol{r})_{\Omega_e} - (\boldsymbol{A}_{fd_2}\boldsymbol{w}_h, \boldsymbol{\nabla}\boldsymbol{r})_{\Omega_e} \\ + \langle \boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\hat{\boldsymbol{U}}_h\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial\Omega_e} + \langle \boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}})\boldsymbol{L}_h\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial\Omega_e} + \langle \boldsymbol{A}_{fd_2}\boldsymbol{w}_h\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial\Omega_e} \\ + \langle \boldsymbol{S}\boldsymbol{U}_h, \boldsymbol{r}\rangle_{\partial\Omega_e} - \langle \boldsymbol{S}\hat{\boldsymbol{U}}_h, \boldsymbol{r}\rangle_{\partial\Omega_e} = (\boldsymbol{f}, \boldsymbol{r})_{\Omega_e}, \\ (\bar{\rho}\boldsymbol{L}_h, \boldsymbol{z})_{\Omega_e} + (\rho_h\boldsymbol{v}_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e} - \langle \widehat{\rho_h\boldsymbol{v}_h}, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e} + (\bar{\boldsymbol{v}}\boldsymbol{\nabla}\rho_h, \boldsymbol{z})_{\Omega_e} = \boldsymbol{0}, \\ (\bar{\rho}\boldsymbol{w}_h, \boldsymbol{g})_{\Omega_e} + (\rho_h E_h, \boldsymbol{\nabla}\cdot\boldsymbol{g})_{\Omega_e} - \langle \widehat{\rho_h E_h}, \boldsymbol{g}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e} + (\bar{E}\boldsymbol{\nabla}\rho_h, \boldsymbol{g})_{\Omega_e} = \boldsymbol{0}, \end{aligned}\right\} \tag{2.52}$$

for $e = 1, \cdots, n_{el}$, and

$$\sum_{e=1}^{n_{el}} \langle \Big(\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U}_h + \widehat{\boldsymbol{A}_{fd_1}\boldsymbol{L}_h} + \boldsymbol{A}_{fd_2}(\bar{\boldsymbol{v}})\boldsymbol{w}_h \Big)\cdot\boldsymbol{n}, \boldsymbol{\mu}\rangle_{\partial\Omega_e\setminus\partial\Omega} = 0, \tag{2.53}$$

for all $(\boldsymbol{r}, \boldsymbol{z}, \boldsymbol{g}, \boldsymbol{\mu}) \in [\mathscr{W}_h^p]^{sd+2} \times [\mathscr{W}_h^p]^{sd+2} \times [\mathscr{W}_h^p]^{sd} \times [\mathscr{M}_h^p]^{sd+2}$, and as usual, $(.,.)_{\Omega_e}$ denotes the $\mathcal{L}^2$ scalar product in the element Ω_e and $\langle .,. \rangle_{\partial\Omega_e}$ denotes the $\mathcal{L}^2$ scalar product in the element boundary $\partial\Omega_e$. Note that $\boldsymbol{S}$ is the stabilization matrix and the definition of numerical flux is

$$\left(\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U}_h + \widehat{\boldsymbol{A}_{fd_1}(\bar{\boldsymbol{v}})}\boldsymbol{L}_h + \boldsymbol{A}_{fd_2}\boldsymbol{w}_h\right)\cdot\boldsymbol{n} = \left(\boldsymbol{A}_{fc}\hat{\boldsymbol{U}}_h + \boldsymbol{A}_{fd_1}\boldsymbol{L}_h + \boldsymbol{A}_{fd_2}\boldsymbol{w}_h\right)\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{U}_h - \hat{\boldsymbol{U}}_h), \tag{2.54}$$

The HDG discrete problem defined by (2.52) and (2.53) is a system of equations, which can be efficiently discretized in time with an implicit time integrator, such as backward Euler or Crank-Nicolson method. Time discretization of (2.52) and (2.53), combined with extrapolation of new terms ($\bar{\boldsymbol{v}}$, $\bar{\rho}$ and $\bar{E}$), leads to a *linear system of equations* at each time step. In this linear system, the equations corresponding to local problem (2.52) can be solved element-by-element to express the solution at each element Ω_e in terms of the trace of conserved variables, $\hat{\boldsymbol{U}}$. Then these expressions are replaced in global problem (2.53) yielding the global system of equations that only involves $\hat{\boldsymbol{U}}$, with an important reduction in DOFs. More details are presented in appendix C.3. Steady state computations are also considered here and they follow the same procedure, relaxing in time toward the steady state solution.

2.3.2 Numerical results

In this part, the solver for linearized compressible Navier-Stokes with HDG spatial discretization and Crank-Nicolson time discretization is used to test the ability of the method for computations of compressible viscous flow problems. It should be noted that backward Euler time marching is also possible and it is implemented in the code. However, the examples are computed with Crank-Nicolson time marching.

Three numerical examples are considered here. First, the unsteady viscous compressible flow in a circle is considered to test the accuracy of the method. The second numerical example is the steady laminar flow around airfoil, and the ability of the method to reach the steady state solution and critical time step size are investigated and the results are also compared with non-linear steady solver. Finally in a vortex convection problem, the performance of the method is tested for an unsteady problem, and this time a comparison with non-linear unsteady solver is presented.

Unsteady viscous flow in a circle

In first example, two-dimensional unsteady compressible Navier-Stokes equations are solved in a circular computational domain of radius 0.5. This test case is an unsteady version of the example in section 2.2.2. An inhomogeneous source term has been imposed on right hand side of the equations to ensure the analytical solution of the form

$$U(x,y,t) = U_{st}(x,y)[1 - e^{-\varkappa(t+t_0)}], \tag{2.55}$$

$$U_{st}(x,y) = \begin{bmatrix} \rho_{st} \\ \rho_{st}\boldsymbol{v}_{st} \\ \rho_{st}E_{st} \end{bmatrix} \tag{2.56}$$

where

$$\begin{aligned} \rho_{st} &= \rho_0(1 + \sin(\pi x)\cos(\pi x)\sin(\pi y)\cos(\pi y)) \\ u_{st} &= u_0(1 + \sin(\kappa\pi x)\cos(\kappa\pi x)\sin(\kappa\pi y)\cos(\kappa\pi y)) \\ v_{st} &= v_0(1 + \sin(\kappa\pi x)\cos(\kappa\pi x)\sin(\kappa\pi y)\cos(\kappa\pi y)) \\ \rho_{st}E_{st} &= E_0(1 + \sin(\pi x)\sin(\pi x)\sin(\pi y)\sin(\pi y)) \end{aligned} \tag{2.57}$$

in which κ is the frequency of the velocity solution and U_{st} is the solution at steady state. Like the example in section 2.2.2, the coefficients (ρ_0, u_0, v_0, E_0) are set to be $(1, 0.5, 0, 5, 3)$, Reynolds number is set to $Re = 1$, and on the boundary, Dirichlet boundary condition is imposed according to the exact solution. Parameters $\varkappa$ and t_0 determine the rate of movement toward steady state solution and initial solution respectively. As stated before, a Crank-Nicolson time marching is used for the discretization in time.

First, to linearize the equations, the exact solution is used, i.e. $\bar{\rho} = \rho_{ex}$, $\bar{\boldsymbol{v}} = \boldsymbol{v}_{ex}$ and $\bar{E} = E_{ex}$. This is done to eliminate the extrapolation error for linearization of parameters from previous time steps. Remember that here, the error includes spatial discretization error, temporal discretization error and extrapolation error. A computational mesh of 1916 elements and polynomial degree $p = 4$ is used with big time step sizes of $\Delta t = 0.5, 1, 2, 4$s, so temporal discretization error dominates the error. The convergence of error in time is shown in Figure 2.17. It is second order accurate, as expected for Crank-Nicolson method.

Then, in Figure 2.18, density and x-component of velocity are shown for the unsteady solutions on a mesh of 122 elements with degree $p = 3$, at $t = 4$s. It can

be seen that the frequency of the velocity is twice of the frequency of density. These solutions are obtained with extrapolation $\bar{a}^{n+1} = \frac{3}{2}a_n - \frac{1}{2}a_{n-1}$ for $\bar{\rho}, \bar{\boldsymbol{v}}$ and $\bar{E}$ and $\Delta t = 1 \times 10^{-3}$. The proposed method is able to solve the compressible Navier-Stokes problems without a non-linear solver.

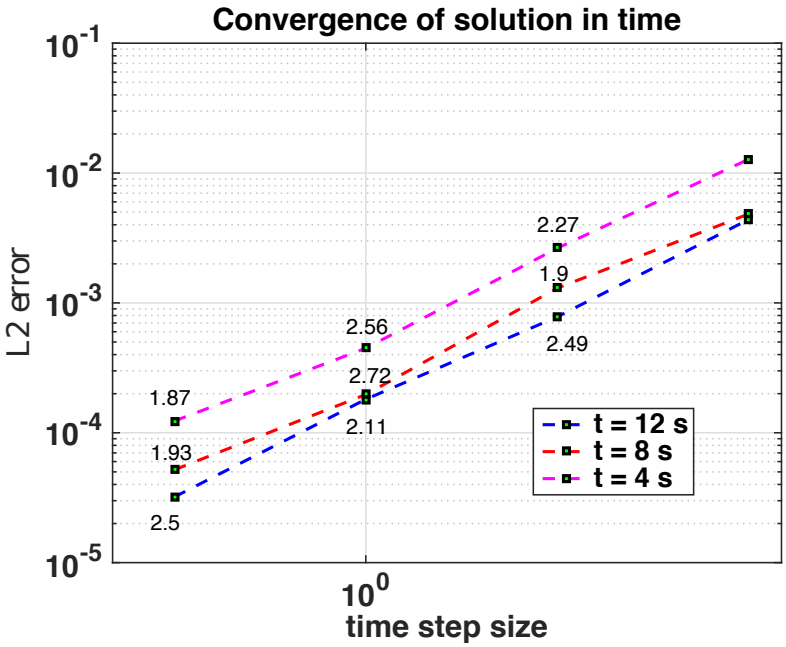
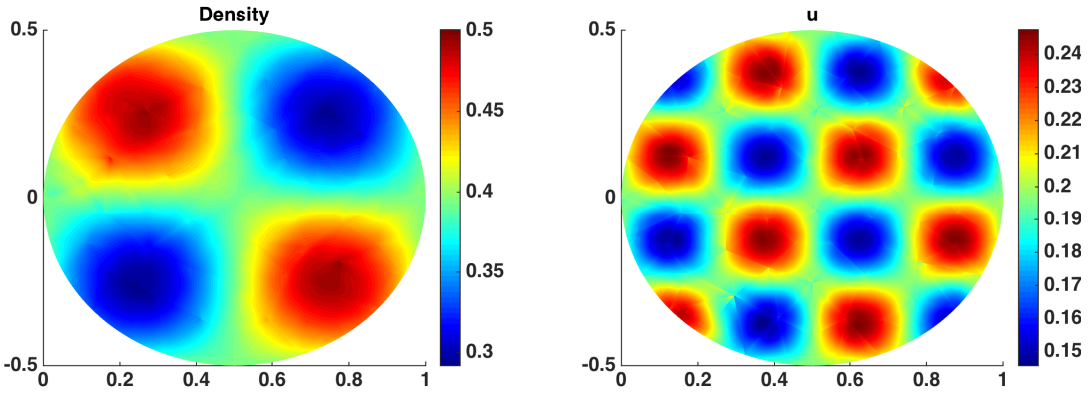


Figure 2.17: Convergence of error in time for linearized HDG method with exact solution for extrapolated values, $\bar{\rho}$, $\bar{\boldsymbol{v}}$ and $\bar{E}$



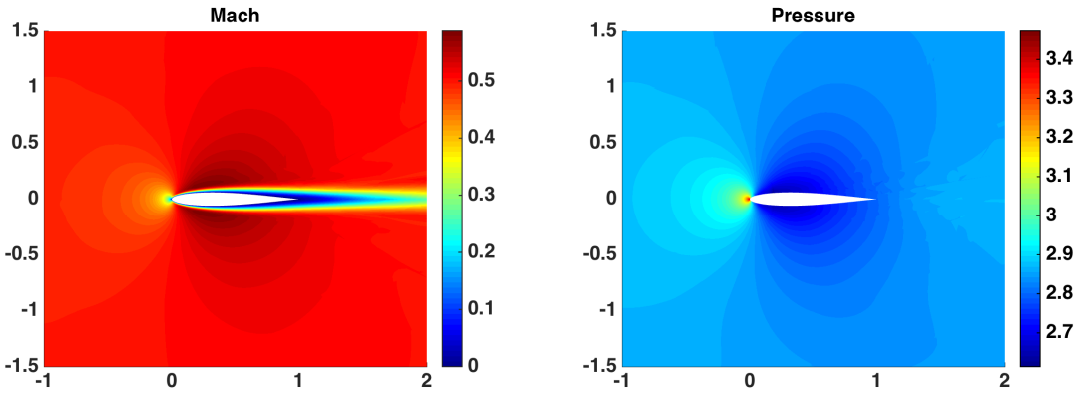
(a) density (b) x-velocity

Figure 2.18: Solution with the linearized HDG method for a viscous flow in a circle for mesh of 122 elements degree $p = 3$, at $t = 4$s

Laminar flow over NACA 0012 airfoil

In this part, the steady state solution of viscous compressible flow over NACA 0012 airfoil is presented. The angle of attack is $\alpha = 0$, Reynolds number is $Re_\infty = 1500$ and Mach number is $M = 0.5$. The far-field boundary condition is Euler inflow/outflow boundary condition for subsonic flow and the airfoil surface is no-slip boundary with adiabatic wall condition. The computational mesh is coarse and the computational domain is a circle with radius 10 times the airfoil cord. Here extrapolation from previous time steps is used for linearization, and a Crank-Nicolson time marching is used for temporal discretization.

Figures 2.19(a) and 2.19(b) show the Mach number and pressure distribution for mesh of 448 cubic elements and Figure 2.20 shows the time history of the convergence of solution toward steady state solution. $\frac{\delta \boldsymbol{U}}{\boldsymbol{U}}$ is the normalized increment in vector of solution, at each time step. The critical time step is $\Delta t_{cr} = 1 \times 10^{-4}$ and any time step size bigger than this leads to divergence of the code. This example shows that the proposed method is able to reach the steady state solution of compressible Navier-Stokes equations with sufficiently small time step size. Given that the time integration is done with Crank-Nicolson, that is unconditionally stable, the critical time step size is forced by the extrapolation.



(a) Mach (b) Pressure

Figure 2.19: Laminar subsonic flow over NACA 0012, mesh of 448 elements $p = 3$

The distribution of pressure coefficient of the airfoil is presented in Figure 2.21, for solutions from linearized solver, as well as non-linear solver in section 2.2. A good agreement between solutions of two different solvers for steady problems can be

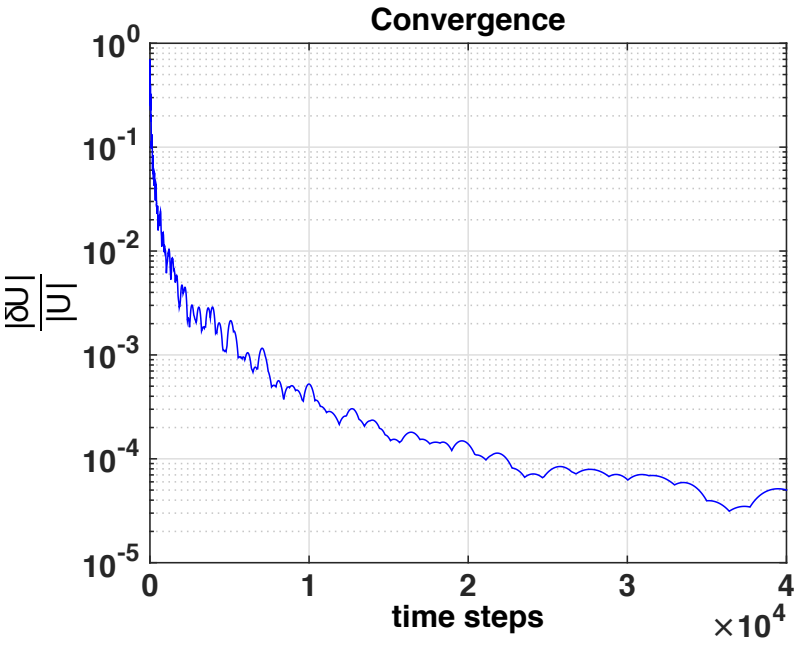


Figure 2.20: Convergence of solution

observed, even on a coarse mesh, so non-linear solver and linearized solver with time relaxation can be used interchangeably.

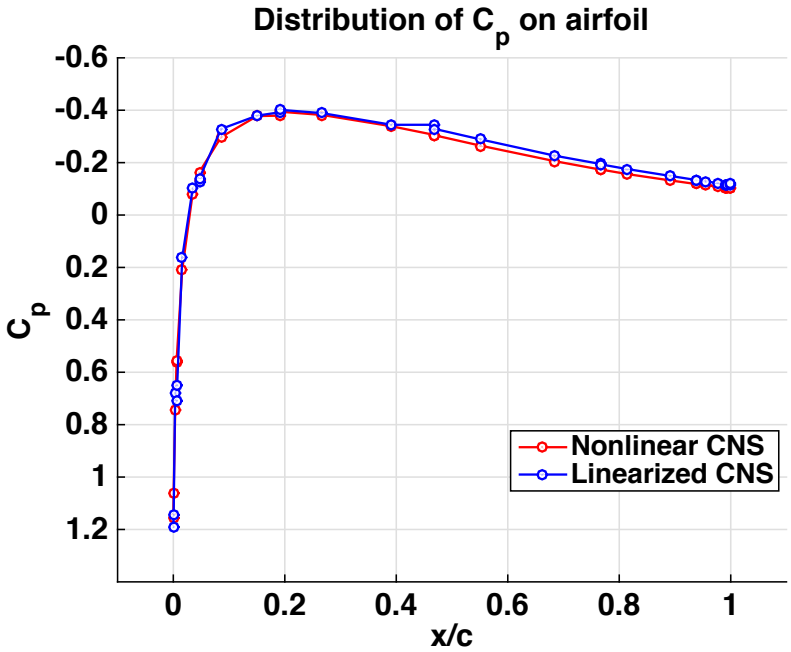


Figure 2.21: Comparison of solutions of linear unsteady and non-linear steady solvers, mesh of 448 elements $p = 3$

One of motivations to develop a linearization method for compressible Navier-Stokes is to improve the robustness of the solver, and avoid continuation method. The non-linear Newton-Raphson solver is quadratic and converges in a few iterations but if the initial guess is far from the solution, it may diverge and small steps in continuation method may be needed, while using the linearized solver the solution of compressible problems in high Reynolds and high Mach numbers can be achieved easier (without intermediary steps). The linear solver uses data from previous time steps to linearize the compressible Navier-Stokes equations, and given the high number of non-linearities to deal with, it is only natural that the time step size should be small enough to ensure the stability of the method.

Vortex convection problem

In this section, an unsteady laminar problem is addressed in order to evaluate the performance of the linearized HDG method for compressible Navier-Stokes equations in time-dependent problems, and to compare with the transient non-linear solver. The test is a 2D vortex convection in a laminar flow regime from Birken et al. [2012]. The initial condition is the free stream condition $(\rho_\infty, v_{1_\infty}, v_{2_\infty}, T_\infty) = (1, v_{1_\infty}, 0, 1)$ with a perturbation at time t_0. The perturbation is a vortex of $(\delta v_1, \delta v_2, \delta T)$ centred at $(\bar{x}_1, \bar{x}_2)$, with

$$\begin{aligned} \delta v_1 &= -\frac{\varepsilon}{2\pi}(x_2 - \bar{x}_2)e^{\phi(1-r^2)}, \\ \delta v_2 &= \frac{\varepsilon}{2\pi}(x_1 - \bar{x}_1)e^{\phi(1-r^2)}, \\ \delta T &= -\frac{\varepsilon^2(\gamma-1)}{16\phi\gamma\pi^2}(x_1 - \bar{x}_1)e^{2\phi(1-r^2)}, \end{aligned} \tag{2.58}$$

where ε and ϕ are parameters which determine the tweak and size of the speed of the flow in vortex, respectively; and r is the distance from center of vortex. For this test, the center of vortex is $(\bar{x}_1, \bar{x}_2) = (0, 0)$, in the domain $[-7, 7] \times [-3.5, 3.5]$. Parameters are chosen as $\varepsilon = 4$, $\phi = 1$, $v_{1_\infty} = 0.5$ and $t_{end} = 4.0$, while the Reynolds number is set to $Re = 100$.

Figures 2.22(a) and 2.22(b) show the pressure and density of the flow, respectively, after 4 seconds. These results are computed on a mesh of 196 elements of degree $p = 4$ and time step size of $\Delta t = 10^{-3}$ is used. For this example, critical time step for the linearized HDG method is found out to be around $\Delta t_{cr} = 2 \times 10^{-3}$ and bigger time step size leads to divergence of the solution.

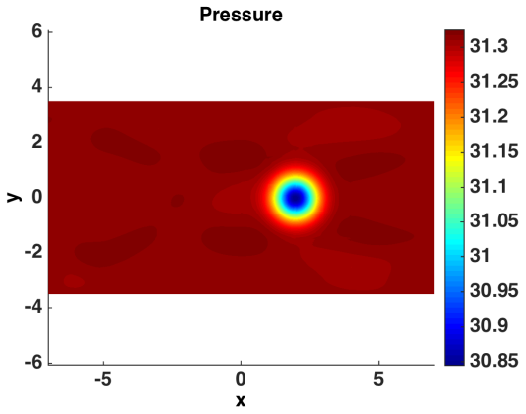
The non-linear unsteady solver, presented in section 2.2, is employed to compute the solution with the same mesh and degree of approximation. It should be noted that non-linear solver utilize a backward Euler time marching, hence, it is stable and we can use big time step sizes. Here for this test case, two time step sizes of $\Delta t = 10^{-2}$ and $\Delta t = 10^{-1}$ are considered. To compare the results, the normalized difference between the solution of linear solver and non-linear solver is calculated; first for $\Delta t = 10^{-2}$, pressure and density differences are shown in Figures 2.22(c) and 2.22(d), respectively, and then for $\Delta t = 10^{-1}$, pressure and density differences are presented in Figures 2.22(e) and 2.22(f), respectively.

These results show that there is a great agreement between the solution of linear and non-linear solver for unsteady problems. For time step sizes of $\Delta t = 10^{-2}$, the differences are of the order 10^{-4}, while for time step sizes of $\Delta t = 10^{-1}$, the differences are of the order 10^{-3}. As expected, using bigger time step size leads to less accurate solution. The main conclusion is both linear and non-linear solvers are capable of computing the unsteady compressible Navier-Stokes problem, however, while non-linear solver can utilize bigger time step sizes and produce relatively accurate results with less computational time (here 40 and 400 time steps have been calculated for $\Delta t = 10^{-1}$ and $\Delta t = 10^{-2}$ respectively), the linear solver needs much smaller time step sizes, due to extrapolation, hence, it has bigger computational time (here 4000 time steps were required to reach the final time).

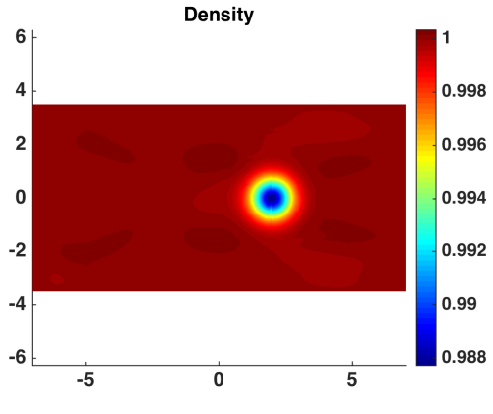
2.4 Conclusions

The HDG method is presented for 2D steady linear convection-diffusion equation. The basic features of the method, like optimal convergence of the solution and super-convergence of the post-processed solution are investigated through numerical results. So the HDG method for convection-diffusion is tested and proved numerically to be optimal for solution, and super-optimal for post-processed solution.

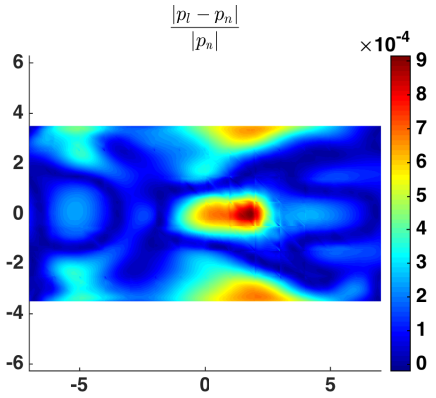
Then, HDG method is applied to both steady and unsteady compressible Navier-Stokes equations, with a Newton-Raphson non-linear solver. The optimal accuracy of the method is studied for a synthetic problem with exact solution. The ability of the method, in computing the viscous compressible flow around NACA 0012 airfoil is investigated by comparing distribution of pressure coefficient on the airfoil, and the error of aerodynamic coefficients have been utilized to check the efficiency



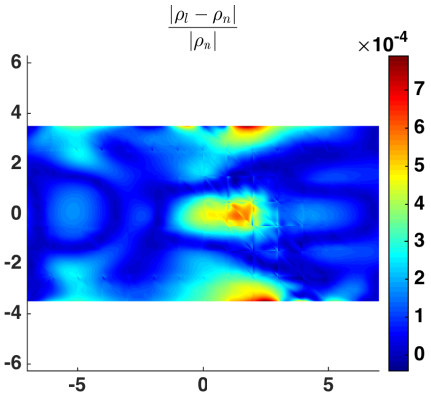
(a) pressure (linear solver with $\Delta t = 1 \times 10^{-3}$)



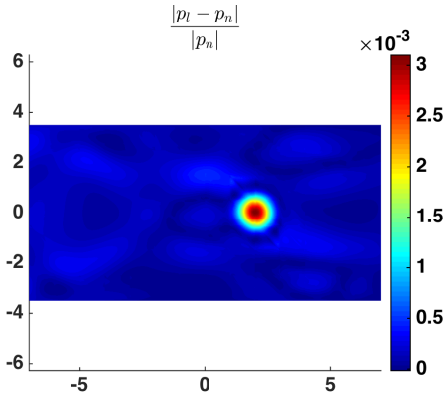
(b) density (linear solver with $\Delta t = 1 \times 10^{-3}$)



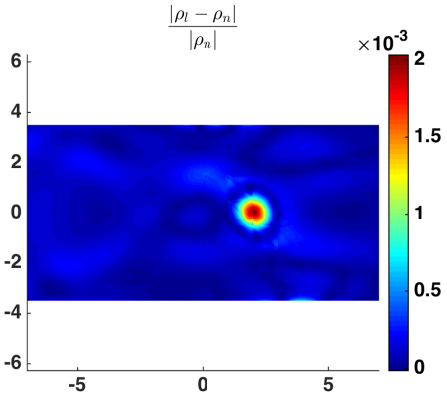
(c) pressure difference (non-linear solver with $\Delta t = 1 \times 10^{-2}$)



(d) density difference (non-linear solver with $\Delta t = 1 \times 10^{-2}$)



(e) pressure difference (non-linear solver with $\Delta t = 1 \times 10^{-1}$)



(f) density difference (non-linear solver with $\Delta t = 1 \times 10^{-1}$)

Figure 2.22: Vortex convection, mesh of 196 elements, $p = 4$, after 4 seconds

of high-order approximation versus low-order one. Implicit time integration is also implemented in straightforward manner, and solutions of both steady and transient solvers are also compared for flow over flat plate to ensure the accuracy of both solvers. Thus, the HDG method is proved to be accurate and capable for compressible viscous flow problems, and it also demonstrated its high-order efficiency to compute accurate solutions in comparison with low-order approximations, since it can obtain more accurate solution with less DOFs.

A linear solver for compressible Navier-Stokes is developed based on linear extrapolation from previous time steps. The accuracy and convergence of the method, are studied, and its ability to obtain solution of laminar compressible flow problems are proved. The linear solver is compared to non-linear solver, for steady NACA 0012 airfoil problem, and for unsteady vortex convection problem. In both cases, the accuracy of the linear solver is in good agreement with non-linear solver. Overall the linearized HDG for compressible Navier-Stokes equations is proved to be a novel approach for computation of viscous compressible flow problems, and it is proved to be accurate and capable, however, using the extrapolation from previous time steps, takes its tools on the limitation it imposes on the time step size, and as a results, for steady problem it may be too computationally costly.

Overall in this chapter, the HDG method is proved to be a good candidate for computations of highly convective flows and it is demonstrated to be both accurate and high-order efficient for these problems.

Chapter 3

A new shock-capturing strategy for HDG method

This chapter deals with the presence of shocks and sharp fronts in the solution, for both convection-diffusion and Navier-Stokes equations. Here, based on the idea in work of Huerta et al. [2012], the strategy is to exploit the stabilization induced by DG numerical fluxes to capture sharp fronts of the solution inside high-order elements. First, a discontinuity sensor developed by Persson and Peraire [2006] is used to detect the elements affected by sharp fronts. This discontinuity sensor is based on the rate of decay of the coefficients of the approximated solution, and quantifies the smoothness of the solution with an elemental scalar. Based on the smoothness of the solution, the approximation space inside each element is modified, with nodal basis functions, to a discontinuous approximation. The basis of discontinuous shape functions inside the elements is based on a division of the element in non-overlapping sub-cells, such that each sub-cell contains one elemental node. The weak form corresponding to the HDG local problem in the element is modified to take into account the discontinuities inside element, introducing DG numerical fluxes across sub-cells boundaries. As a result, the numerical fluxes inside elements provide additional stabilization with no addition to DOFs. By means of this shock-capturing technique, the order of the approximation is reduced only in the elements where the solution is not smooth. Thus, the high-order accuracy, of order $p+1$ in the large majority of the domain, is locally decreased to order h/p only in the elements where the shock is contained,

being p the degree of approximation, and h the element size. As a result, no mesh adaptation is needed, and sharp fronts can be captured without modifying the DOFs or mesh topology.

The outline of this chapter is as follows. First, section 3.1, presents the stabilized HDG method for convection-diffusion equations. The continuous and discontinuous nodal basis for standard and stabilized elements respectively, are presented in section 3.1.1. The weak form of the HDG local problem, for standard and stabilized elements, and for the HDG global problem are stated in section 3.1.2. The shock sensor is detailed in section 3.1.3. Finally, numerical examples in section 3.1.4, demonstrate the ability of the method to capture shocks in the solution, and its excellent performance in damping oscillations is the vicinity of shocks to obtain a spurious-free high-order solution of two dimensional steady convection-diffusion equations. Next, in section 3.2 the strategy is extended to the compressible Navier-Stokes equations. First in section 3.2.1, the HDG local problem and modified HDG local problem are developed for standard and stabilized elements respectively, and the proper DG flux for inter-sub-cell stabilization is presented. Finally in section 3.2.2, numerical examples show the ability of the method to capture shocks in compressible viscous flow problems.

3.1 High-order HDG with shock-capturing for convection-diffusion problems

The convection-diffusion equation, and its HDG formulation are presented in section 2.1 and here the same definitions and notations are used. The convection-diffusion equation in a discrete domain, presented by (2.4) is considered as a starting point and the difference between the standard HDG, presented in section 2.1 and modified HDG for shock-capturing is described in the following sections.

3.1.1 Discretization space and discontinuous shape functions

For HDG discretization, two types of finite element spaces must be defined; one for functions in the elements interior and another for trace functions on the mesh

skeleton, Γ. The approximation space for elemental variables u, $\boldsymbol{q}$ is

$$\begin{aligned} \mathscr{W}_h^p = \{v \in \mathcal{L}^2(\Omega) : v|_{\Omega_e} \in \mathcal{P}^p(\Omega_e) \qquad & \text{for} \quad e \notin \mathscr{E}_{\mathscr{S}}, \\ v|_{\Omega_e} \in \widetilde{\mathcal{P}^p}(\Omega_e) \qquad & \text{for} \quad e \in \mathscr{E}_{\mathscr{S}}\} \end{aligned} \tag{3.1}$$

where $\mathscr{E}_{\mathscr{S}}$ is the set of index of the elements detected by shock detector, i.e. stabilized elements.

At standard elements, $e \notin \mathscr{E}_{\mathscr{S}}$, the standard p-th degree polynomial approximation space $\mathcal{P}^p(\Omega_e)$ is considered, with nodal basis

$$\mathcal{P}^p(\Omega_e) = \text{span}\{N_1^e(\boldsymbol{x}), N_2^e(\boldsymbol{x}), ..., N_{n_{en}}^e(\boldsymbol{x})\} \tag{3.2}$$

where n_{en} is number of element nodes.

For stabilized elements, $e \in \mathscr{E}_{\mathscr{S}}$, which are detected by the discontinuity sensor, a *discontinuous space of approximation*, is considered inside the element and on its faces. The detected element is arbitrarily divided into n_{en} non-overlapping partitions $\{V_e^k\}_{k=1}^{n_{en}}$ such that

$$\bar{\Omega}_e = \bigcup_{k=1}^{n_{en}} \bar{V}_e^k, \qquad V_e^k \cap V_e^m = \emptyset \quad \text{for} \quad k \neq m, \tag{3.3}$$

and each sub-cell V_e^k contains only one elemental node, see an example in Figure 3.1. Then new shape functions $\tilde{N}_i^e(\boldsymbol{x};\beta)$ are defined as a convex combination with parameter $\beta \in [0,1]$ of standard polynomials, $N_i^e(\boldsymbol{x})$, and a set of piecewise constant functions (constant within each sub-cell of the element), $\phi_i^e(\boldsymbol{x})$. That is the approximation space in an stabilized element is

$$\widetilde{\mathcal{P}^p}(\Omega_e) = \text{span}\{\tilde{N}_1^e(\boldsymbol{x};\beta), \tilde{N}_2^e(\boldsymbol{x};\beta), ..., \tilde{N}_{n_{en}}^e(\boldsymbol{x};\beta)\} \tag{3.4}$$

with

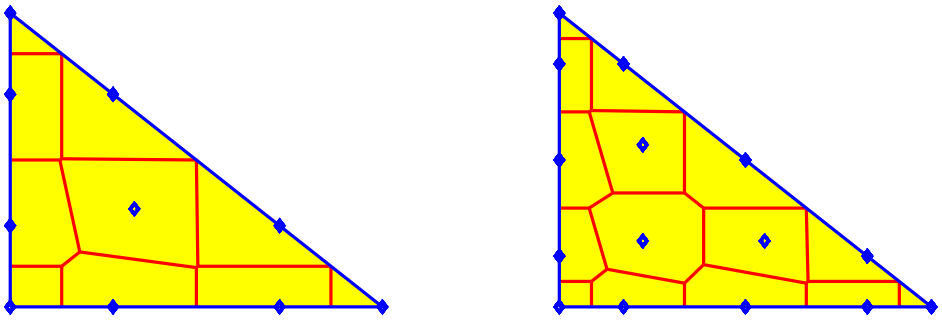
$$\tilde{N}_i^e(\boldsymbol{x};\beta) := (1-\beta)N_i^e(\boldsymbol{x}) + \beta\phi_i^e(\boldsymbol{x}), \tag{3.5}$$

$$\phi_i^e(\boldsymbol{x}) = \phi_i^k \quad \text{for} \quad \boldsymbol{x} \in V_e^k, \qquad k = 1, ..., n_{en}, \tag{3.6}$$

for $i = 1, ..., n_{en}$, where parameter β characterizes the smoothness of the approximation, and it is given by shock sensor, and $\phi_i(\boldsymbol{x})$ are the piecewise constant functions. The constants ϕ_i^k are here defined as

$$\phi_i^k = \frac{1}{\text{meas}(V_e^k)} \int_{V_e^k} N_i(\boldsymbol{x}) dV \qquad , k = 1, ..., n_{en} \tag{3.7}$$

Other definitions for ϕ_i^k can be used such as, for instance, $\phi_i^k = \delta_{ik}$, provided that they always lead to a partition of unity, see Huerta et al. [2012] for more details. The numerical experiments show that, non-delta functions work better for convergence of non-linear solver, while using delta functions may cause divergence of the solution for non-linear problems. It is worth noting that in the limit case of $\beta = 1$, a piecewise constant approximation is obtained over the cub-cells, $\tilde{N}_i(\boldsymbol{x}; 1) = \phi_i$.



(a) Sub-cells for p=3 (b) Sub-cells for p=4

Figure 3.1: Cell partitions for cubic and fourth order approximations

The parameter β in equation (3.5) controls the magnitude of the jump across the sub-element interfaces. This parameter benefits from a major feature of DG methods in flexibility in modifying the approximation space element by element. The value of β is adapted to adequately capture the different discontinuities in the solution. Thus, without modifying the mesh topology, nor the number of DOFs, nor the node position, nor the structure of the matrices, the space of the approximation is adapted in space and time to the regularity of the solution.

As it is clear in equation (3.5), the extreme values of β ($\beta = 0$ or 1) can lead to totally continuous or discontinuous solutions inside the element. Intermediate values of β allow to calibrate the amount of stabilization introduced in the solution. Therefore, β is a function of discontinuity sensor, S_e, which is indicator of the smoothness of the solution, and will be introduced in section 3.1.3.

The expression of β in term of S_e can take different forms. The simplest choice is to model it as a switch function. That is, $\beta = 0$ if discontinuity sensor does not detect a discontinuity and $\beta = 1$ when a discontinuity is detected. Thus, functions $\tilde{N}_i$ are prescribed to be either piecewise constant or smooth p^{th}-order approximations. Here a linear variation is proposed to introduce a smooth transition between piecewise

constant or smooth p^{th}-order approximations. In this manner when the discontinuity sensor clearly detects a discontinuity, the parameter is $\beta = 1$, and when the solution is clearly smooth, the parameter is $\beta = 0$, and p^{th}-order approximation is recovered. In between, a linear variation progressively introduces larger stabilization.

Similarly, the approximation space for trace variable $\hat{u}$, is

$$\begin{aligned} \mathscr{M}_h^p = \{\mu \in \mathcal{L}^2(\Gamma) : \mu|_{\Gamma_f} \in \mathcal{P}^p(\Gamma_f) \qquad & \text{for} \quad f \notin \mathscr{F}_{\mathscr{S}}, \\ \mu|_{\Gamma_f} \in \widetilde{\mathcal{P}^p}(\Gamma_f) \qquad & \text{for} \quad f \in \mathscr{F}_{\mathscr{S}}, \} \end{aligned} \tag{3.8}$$

where $\mathscr{F}_{\mathscr{S}}$ is the set of index of the faces shared by two stabilized elements; and

$$\begin{aligned} \mathcal{P}^p(\Gamma_f) &= \text{span}\{N_1^f(\boldsymbol{x}), N_2^f(\boldsymbol{x}), ..., N_{n_{fn}}^f(\boldsymbol{x})\}, \\ \widetilde{\mathcal{P}^p}(\Gamma_f) &= \text{span}\{\tilde{N}_1^f(\boldsymbol{x};\beta), \tilde{N}_2^f(\boldsymbol{x};\beta), ..., \tilde{N}_{n_{fn}}^f(\boldsymbol{x};\beta)\}, \end{aligned} \tag{3.9}$$

where n_{fn} is number of face nodes, $\{N_i^f(\boldsymbol{x})\}_{i=1}^{n_{fn}}$ are standard polynomial nodal basis functions on the face Γ_f and $\{\tilde{N}_i^f(\boldsymbol{x})\}_{i=1}^{n_{fn}}$ are discontinuous basis functions obtained the same way as (3.5) and (3.6).

Remark 3.1. *If face Γ_f is shared by two elements, Ω_L and Ω_R, with different β, the smaller β is considered at the face.*

$$\beta_f = \min(\beta_L, \beta_R) \qquad on \quad \Gamma_f = \Omega_L \cap \Omega_R \tag{3.10}$$

Note that standard elements correspond to $\beta = 0$; thus, for the faces between continuous and discontinuous elements, a continuous representation of basis functions is used as illustrated in Figure 3.2.

With the defined finite element spaces (3.1) and (3.8), the HDG local and global problem can be discretized for both standard and stabilized elements, as explained in next section.

3.1.2 HDG formulation with shock-capturing

This section presents the novel modified HDG weak form for the local problem in stabilized elements. For standard elements, $e \notin \mathscr{E}_{\mathscr{S}}$, standard HDG discretization of the local problem in each element (2.15) is used here. This standard HDG formulation is not appropriate for the stabilized elements, because of the discontinuities across sub-cells boundaries, hence there is a need for modified HDG formulation.

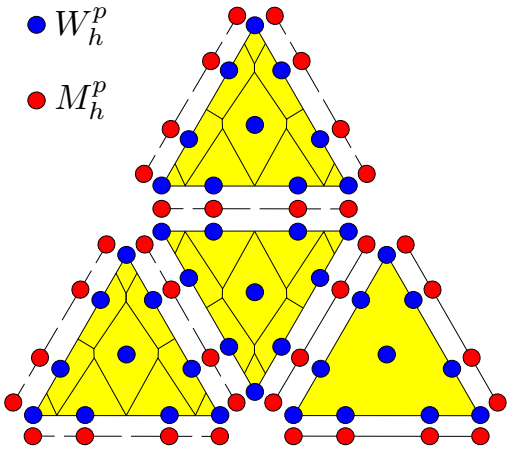


Figure 3.2: Representation of spaces for elements with and without sub-cells for degree $p = 3$

For stabilized elements, $e \in \mathscr{E}_{\mathscr{S}}$, the modified approximation space with discontinuities inside the element, between sub-cells (3.1) is considered, see section 3.1.1. Discretization of the local problem in k^{th} sub-cell of e^{th} element is: given $\bar{u}_h$ on ∂V_e^k, find $u_h \in \widetilde{\mathcal{P}^p}(V_e^k)$ and $\boldsymbol{q}_h \in [\widetilde{\mathcal{P}^p}(V_e^k)]^{sd}$ such that

$$\begin{aligned}(k^{-1}\boldsymbol{q}_h, \boldsymbol{z})_{V_e^k} - (u_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{V_e^k} + \langle \bar{u}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial V_e^k \setminus \partial\Omega_e} + \langle \bar{u}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial V_e^k \cap \partial\Omega_e} = 0,\\ (\boldsymbol{\nabla}\cdot(\boldsymbol{c}u + \boldsymbol{q}_h), r)_{V_e^k} + \langle \boldsymbol{c}\cdot\boldsymbol{n}(\bar{u}_h - u_h) + (\tilde{\boldsymbol{q}}_h - \boldsymbol{q}_h)\cdot\boldsymbol{n}, r\rangle_{\partial V_e^k \setminus \partial\Omega_e}\\ + \langle(\tau - \boldsymbol{c}\cdot\boldsymbol{n})(u_h - \bar{u}_h), r\rangle_{\partial V_e^k \cap \partial\Omega_e} = (f, r)_{V_e^k},\end{aligned} \tag{3.11}$$

for all $r \in \widetilde{\mathcal{P}^p}(V_e^k)$ and all $\boldsymbol{z} \in [\widetilde{\mathcal{P}^p}(V_e^k)]^{sd}$. It should be noted that here the boundary of sub-cell is divided into interior, $\partial V_e^k \setminus \partial\Omega_e$, and exterior part, $\partial V_e^k \cap \partial\Omega_e$. On the exterior part, which is part of the boundary of the element as well, the trace $\bar{u}_h$ is set to the standard trace of the HDG, $\hat{u}_h$ (which is considered as a variable in the global problem), while on the interior part, new fluxes, $\bar{u}_h = \tilde{u}_h$ and $\tilde{\boldsymbol{q}}_h$ are introduced. These numerical fluxes transfer information across the inter-sub-cells boundaries and can be defined as a function of elemental unknowns, u_h and $\boldsymbol{q}_h$, as in standard DG methods.

Summing equation (3.11) over sub-cells of the element, leads to a modified weak form for the local problem: given $\hat{u}$ in $\partial\Omega_e$, find $u_h \in \widetilde{\mathcal{P}^p}(\Omega_e)$ and $\boldsymbol{q}_h \in [\widetilde{\mathcal{P}^p}(\Omega_e)]^{sd}$

such that

$$\begin{aligned}(k^{-1}\boldsymbol{q}_h,\boldsymbol{z})_{\Omega_e}-(u_h,\boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e}+\langle\tilde{u}_h,[\![\boldsymbol{z}\cdot\boldsymbol{n}]\!]\rangle_{\Gamma_e}+\langle\hat{u}_h,\boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e}=0,\\ -(\boldsymbol{q}_h,\boldsymbol{\nabla}r)_{\Omega_e}+\langle\boldsymbol{q}_h\cdot\boldsymbol{n},r\rangle_{\partial\Omega_e}+\langle\tilde{\boldsymbol{q}}_h,[\![r\boldsymbol{n}]\!]\rangle_{\Gamma_e}-(\boldsymbol{c}u_h,\boldsymbol{\nabla}r)_{\Omega_e}+\langle\tau u_h,r\rangle_{\partial\Omega_e}\\ +\langle\tilde{u}_h,[\![r(\boldsymbol{c}\cdot\boldsymbol{n})]\!]\rangle_{\Gamma_e}+\langle(\boldsymbol{c}\cdot\boldsymbol{n}-\tau)\hat{u}_h,r\rangle_{\partial\Omega_e}=(f,r)\Omega_e,\end{aligned}\tag{3.12}$$

for all $r\in\widetilde{\mathcal{P}^p}(\Omega_e)$ and all $\boldsymbol{z}\in[\widetilde{\mathcal{P}^p}(\Omega_e)]^{sd}$, where the set of inter-sub-cells boundaries inside element Ω_e is defined as

$$\Gamma_e=\left[\bigcup_{k=1}^{n_{en}}\partial V_e^k\right]\setminus\partial\Omega_e\tag{3.13}$$

Comparing equations (3.12) and (2.15), one can see that, there are three additional terms in the stabilized weak form. All additional terms, which add extra stabilization, contain jumps, and therefore if a continuous basis of shape functions is used, then standard HDG formulation is recovered. Those jumps appear because in summing over sub-cells of the element, the interior boundary of sub-cell is involved twice, once from left, and once from right, hence, results in a jump over inter-sub-cell boundary.

In this section, for $\tilde{u}_h$ and $\tilde{\boldsymbol{q}}_h$, the definitions for numerical flux of the Bassi-Rebay method, from Bassi and Rebay [1997a], and of the LDG method, from Cockburn et al. [2002], are considered and presented in Table 3.1, where the coefficients C_{11}, $\boldsymbol{C}_{12}$ and C_{22} have effects on the stability and the convergence of the methods, for more details see Cockburn et al. [2002], Castillo et al. [2001].

Method	$\tilde{u}_h$	$\tilde{\boldsymbol{q}}_h$
BR	$\{u_h\}$	$\{\boldsymbol{q}_h\}$
LDG	$\{u_h\}-\boldsymbol{C}_{12}\cdot[\![u_h\boldsymbol{n}]\!]$	$\{\boldsymbol{q}_h\}-C_{11}[\![u_h\boldsymbol{n}]\!]+\boldsymbol{C}_{12}[\![\boldsymbol{q}_h\cdot\boldsymbol{n}]\!]$

Table 3.1: Some possible definition for inter-sub-cells fluxes

The local problems (2.15) and (3.12), allow to express the approximation of solutions u_h and $\boldsymbol{q}_h$ element-by-element, in the whole domain in terms of approximation of the trace of the solution $\hat{u}_h$. The solution is fully determined using the global equations defined on Γ, (2.6). The first one (2.5) is already imposed and the remaining global condition which must be imposed is (2.6). So the global problem is the same as (2.16), and it imposes the continuity of normal component of the numerical flux. It is important to know that while the weak form of global problem is the same for standard HDG and stabilized HDG, the difference is that, basis of shape functions change to discontinuous ones in stabilized elements and also on their faces.

The HDG discrete problem defined by either (2.15) or (3.12), and (2.16), is a system of equations. In this system, the equations corresponding to (2.15) or (3.12) can be solved element-by-element to express the solution at each element, Ω_e in terms of the trace of the solution, $\hat{u}_h$. Then, these expressions are replaced in (2.16), yielding the global system of equations that only involves $\hat{u}_h$, keeping the standard HDG DOFs.

3.1.3 Discontinuity sensor and parameter β

Detecting the sharp front is necessary to switch from continuous shape functions to discontinuous ones in a small region close to the sharp gradients. Here a shock sensor based on the rate of decay of the approximated solution is considered, which is proposed by Persson and Peraire [2006]. It is based on an element-by-element non-linear projection leading to a single scalar measure of the smoothness of the numerical approximation, $S_e(s) : \Omega_e \to \mathbb{R}$, depending only on the sensing variable s, which is here the approximated solution u_h.

The solution is expressed in terms of hierarchical orthonormal polynomials

$$s(x) = \sum_{i=1}^{n_{en}(p)} s_i P_i(x) \tag{3.14}$$

where P_i is a set of orthonormal polynomials of degree p and n_{en} is the number of element nodes. For smooth functions the coefficients in the expansion, s_i, are expected to decay rapidly, while in the regions of sharp gradients the rate of decay of the expansion coefficients lower. A truncated expansion of the same solution without the highest order of approximation is considered.

$$\hat{s}(x) = \sum_{i=1}^{n_{en}(p-1)} s_i P_i(x) \tag{3.15}$$

Then for each element Ω_e, an smoothness indicator can be defined as

$$S_e = 2\log_{10}\left(\frac{\| s - \hat{s} \|_2}{\| s \|_2}\right) \tag{3.16}$$

where $\| \cdot \|_2$ is the standard $L_2(\Omega_e)$ norm.The smoothness indicator S_e is a normalized measure of the highest frequencies in the approximation.

The relationship between smoothness indicator, Se, and parameter β is not unique. One may easily think of a switch between $\beta = 0$ and $\beta = 1$, to turn it off in the regions of smooth solution, and enable it in the detected elements. However, a more practical approach is to smoothly change from standard p^{th}-order continuous shape functions to piecewise constant ones. Here, a linear relation between parameter β and smoothness indicator is chosen, which is shown in figure 3.3.

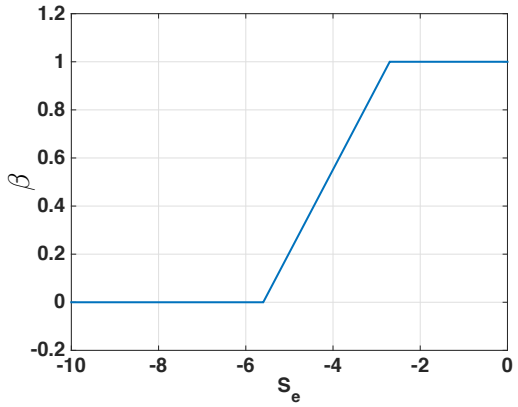


Figure 3.3: Variations of parameter β with smoothness indicator, S_e

3.1.4 Numerical results

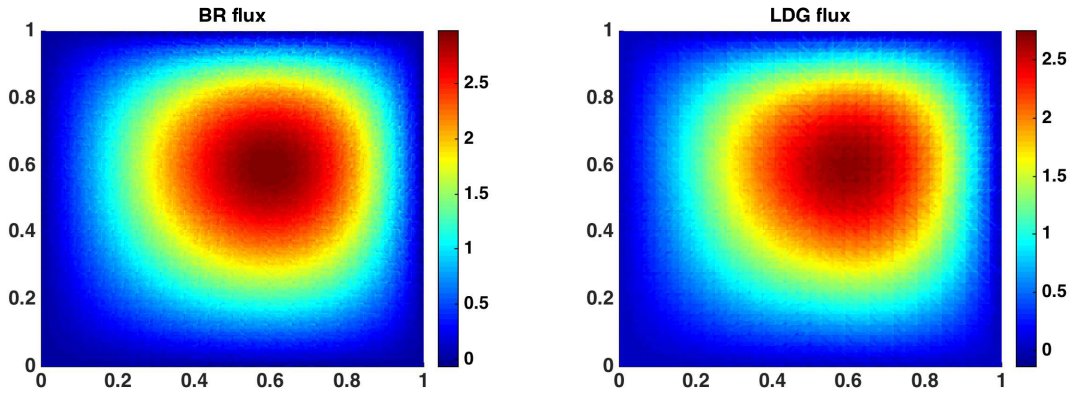
Here some numerical results are presented for several test cases, all two dimensional linear convection-diffusion problems in $\Omega = (0, 1)^2$. First, a problem with smooth solution is considered to check the ability of modified approximation space to converge toward the solution and the convergence rate of the solution with different values of β. Then, a convection-diffusion problem with sharp front is solved to check the ability of the method to eliminate spurious regions of the solution in the vicinity of the layers. Finally, a complex non-constant convection dominated problem with two inner layers is presented to check the ability of the method to deal with sophisticated oblique shocks.

Diffusion dominated problem with a source term: convergence test

In this section, according to an example in Nguyen et al. [2009a], the Dirichlet boundary condition and source term are such that exact solution is

$$u_{ex}(x, y) = \exp(x + y) \sin(\pi x) \sin(\pi y) \tag{3.17}$$

The convection velocity is $\boldsymbol{c} = (1, 1)^T$, and diffusion coefficient is $k = 1$, hence the solution is diffusion dominated. The numerical experiments are conducted for uniform distribution of β with values $\beta = 1, 0.5, 0.1$ over all elements, to test the ability of modified approximation space to converge toward the solution. Note that using the *discontinuous shape functions inside elements* means that the solution is not approximated with standard polynomials of degree p, hence reducing the order of approximations. Figure 3.4 shows the solution, with Bassi-Rebay and LDG flux, approximated with $p = 3$ on uniform mesh of $2n^2$ triangle elements, where $n = 32$. Tables 3.2 and 3.3 show the convergence rate of the errors for discontinuous shape functions with both options for the numerical flux between the sub-cells. The stabilized formulation with Bassi-Rebay flux converges with rate of order $\mathcal{O}(h)$. This rate is optimal for the stabilized formulation, because the approximation can reproduce only constant function exactly. On the other hand, the LDG flux under-performs with slightly lower rates due to 0^{th}-order convergence of gradient of solution in LDG with piecewise constant approximations. It is worth noting that, for smaller stabilization, $\beta \approx 0$, better convergence is achieved in comparison with higher stabilization, $\beta \approx 1$.



(a) Solution with Bassi-Rebay flux (b) Solution with LDG flux

Figure 3.4: Diffusion dominated problem: approximated solution for uniform $\beta = 1$, $p = 3$ and $h = 0.031$

Overall, both choices of fluxes between the sub-cells lead to stable and reasonable results, and both demonstrate ability of modified approximation space to converge toward the solution.

mesh		$\beta = 0.1$		$\beta = 0.5$		$\beta = 1$	
h	**p**	**L2 error**	**order**	**L2 error**	**order**	**L2 error**	**order**
0.5	2	1.22e-01		4.05e-01		6.61e-01	
0.25	2	3.29e-02	**1.89**	1.48e-01	**1.45**	2.69e-01	**1.29**
0.125	2	1.54e-02	**1.09**	7.26e-02	**1.03**	1.32e-01	**1.03**
0.0625	2	7.50e-03	**1.04**	3.54e-02	**1.04**	6.60e-02	**1.00**
0.03125	2	3.71e-03	**1.01**	1.77e-02	**1.00**	3.42e-02	**0.95**
0.5	3	4.56e-02		2.34e-01		5.01e-01	
0.25	3	1.98e-02	**1.20**	9.88e-02	**1.25**	1.98e-01	**1.34**
0.125	3	1.03e-02	**0.95**	5.10e-02	**0.95**	9.98e-02	**0.99**
0.0625	3	5.10e-03	**1.01**	2.55e-02	**1.00**	5.00e-02	**1.00**
0.03125	3	2.55e-03	**1.00**	1.29e-02	**0.99**	2.60e-02	**0.95**

Table 3.2: Diffusion dominated problem: rates of convergence using discontinuous shape functions with Bassi-Rebay flux

mesh		$\beta = 0.1$		$\beta = 0.5$		$\beta = 1$	
h	**p**	**L2 error**	**order**	**L2 error**	**order**	**L2 error**	**order**
0.5	2	1.38e-01		5.09e-01		7.43e-01	
0.25	2	4.50e-02	**1.61**	2.76e-01	**0.88**	4.19e-01	**0.82**
0.125	2	2.23e-02	**1.01**	1.85e-01	**0.57**	2.83e-01	**0.57**
0.0625	2	1.16e-02	**0.94**	1.28e-01	**0.54**	2.02e-01	**0.49**
0.03125	2	6.48e-03	**0.84**	9.86e-02	**0.37**	1.60e-01	**0.34**
0.5	3	6.70e-02		3.81e-01		6.97e-01	
0.25	3	3.14e-02	**1.09**	1.98e-01	**0.94**	3.98e-01	**0.81**
0.125	3	1.70e-02	**0.89**	1.28e-01	**0.63**	2.83e-01	**0.49**
0.0625	3	9.20e-03	**0.89**	8.39e-02	**0.61**	2.12e-01	**0.42**
0.03125	3	5.28e-03	**0.80**	6.15e-02	**0.45**	1.79e-01	**0.25**

Table 3.3: Diffusion dominated problem: rates of convergence using sub-cell discontinuous shape functions with LDG flux

Convection dominated problem with an inner layer: stabilization test

This is a problem with convection skew to the mesh, with unidirectional $\|\boldsymbol{c}\| = 1$ and angle θ, and a very low diffusion of $k = 10^{-7}$, hence a highly convection-dominated regime, without a source term. The inflow boundary condition is discontinuous which introduces a sharp front into the domain, and outflow boundary condition is a homogeneous natural boundary condition, e.g. $\partial u/\partial \boldsymbol{n} = 0$. This problem is taken from Brooks and Hughes [1982] in order to evaluate the ability of the continuous-discontinuous shape functions approach to capture a oscillation free solution in the vicinity of the shock.

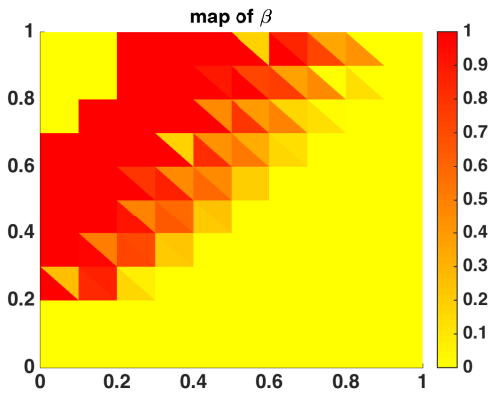
For a solution with angle $\theta = \pi/4$, the problem is solved on a 10 by 10 mesh

of equal size triangular elements with Peclet number $Pe = 10^6$. In the results, HDG solutions with shock-capturing, with different fluxes, are approximated with quartic polynomials and presented in Figure 3.5. The solutions at outflow ($y = 1$) are presented in Figure 3.6. The solutions in Figure 3.5 show that the LDG flux is more successful than Bassi-Rebay flux in removing the overshoots and undershoots of the spurious oscillations of high-order solutions in the vicinity of sharp fronts. However, the LDG flux is seen to be over diffusive and it is not able to accurately capture the sharp front. In order to address this issue, a less discontinuous shape function is used; e.g. instead of $\beta \in [0, 1]$ for the discontinuous shape function, $\beta \in [0, 1/p]$ is used. This option yields the best solution, and as it can be seen in 3.5(d), the overshoots and undershoots are diminished while the sharp front is captured. The cross section of the solutions at outflow and at $y = 0.5$ in Figure 3.6, also show the ability of the method to significantly improve the solution of the HDG method. Again, the best solution is obtained with LDG flux $\beta \in [0, 1/p]$, and from now on this flux is used if no other flux is mentioned.

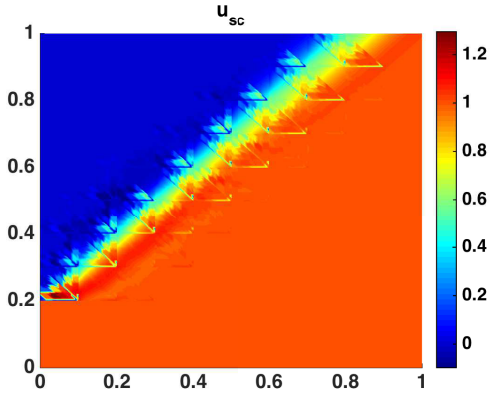
In the second part of the test, an angle $\theta = 3\pi/8$ is considered. The standard HDG solutions at outflow have been compared with HDG solutions *with shock-capturing* in order to further investigate the effect of continuous-discontinuous shape functions approach. The undershoots and overshoots of the shock at outflow are presented in table 3.4 for different orders of approximation and for a coarse and a fine mesh. For these numerical investigations, $k = 10^{-7}$ is used, therefore a very convection-dominated regime is studied. The results demonstrate the capabilities of the method to eliminate the oscillations in the vicinity of shocks and spurious parts of the solutions are diminished significantly.

mesh		**HDG**		**HDG-SC**	
h	**p**	**undershoot**	**overshoot**	**undershoot**	**overshoot**
0.1	2	1.92e-02	5.51e-02	0	4.00e-03
0.1	3	3.95e-02	1.16e-01	0	3.55e-05
0.1	4	4.31e-02	9.25e-02	2.40e-05	1.25e-05
0.1	5	8.91e-02	4.74e-02	3.88e-04	8.16e-06
0.0312	2	5.70e-02	8.69e-02	2.04e-04	1.92e-02
0.0312	3	5.86e-02	7.32e-02	1.00e-03	2.30e-03
0.0312	4	3.51e-02	1.04e-01	9.68e-05	5.01e-06
0.0312	5	6.38e-02	8.99e-02	8.75e-05	2.62e-05

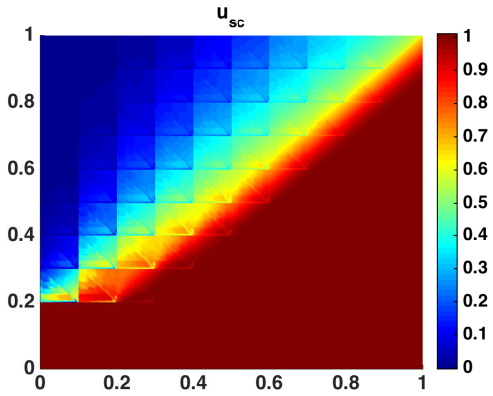
Table 3.4: Convection dominated problem with $\theta = 3\pi/8$: comparison of undershoots and overshoots before and after shock-capturing at outflow



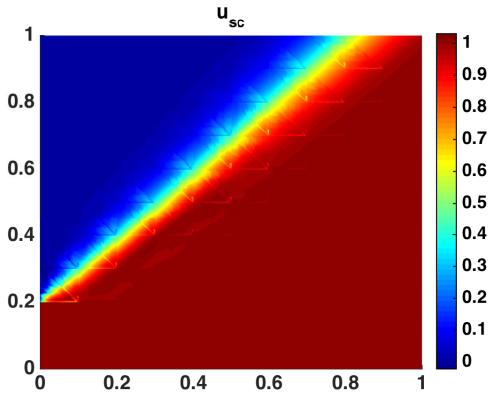
(a) Map of β



(b) u_h, sub-cell flux of Bassi-Rebay



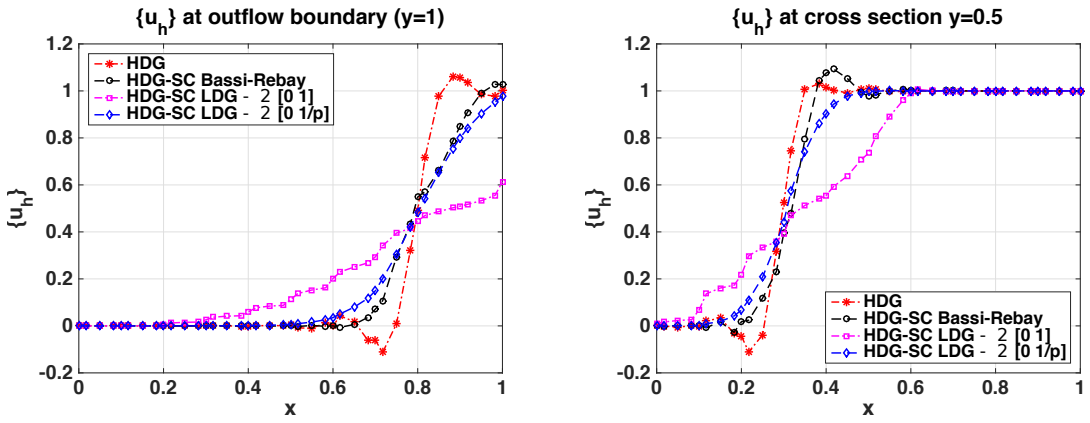
(c) u_h, sub-cell flux of LDG



(d) u_h, sub-cell flux of LDG, $\beta \in [0, 1/p]$

Figure 3.5: Convection dominated problem: HDG with shock-capturing, $\theta = \pi/4$, $p = 4$, $h = 0.1$

In Figure 3.7, a comparison between the results of Brooks and Hughes [1982] and our results is presented. The HDG solutions for $\theta = \pi/8$, $\pi/4$ and $3\pi/8$ are obtained with 4^{th}-degree approximations with LDG flux and $\beta \in [0, 1/p]$. Like the reference, a mesh of 10 by 10 is used and $Pe = 10^6$, hence a very convective regime. The HDG solution with shock-capturing presents the best approximation of the inner layer without typical oscillations of high-order methods. The ability of the continuous-discontinuous shape functions approach to capture the discontinuity is fully demonstrated.



(a) (b)

Figure 3.6: Convection dominated problem: standard HDG vs. stabilized HDG, $\theta = \pi/4$, $p = 4$, $h = 0.1$

Variable convection problem with 2 inner layers

This sophisticated test case for linear convection-diffusion equations is an example from John and Knobloch [2007]. Here the convection velocity is $\boldsymbol{c}(x, y) = (-y, x)$ while $k = 10^{-4}$ and $f = 0$. Dirichlet boundary condition is prescribed to $u(x, y) = 1$ for $1/3 \leq x \leq 2/3$,$y = 0$ and $u(x, y) = 0$ on the remaining parts of lower boundary as well as right and upper boundaries. And on the left boundary a homogeneous natural boundary condition is imposed, e.g. $\partial u/\partial \boldsymbol{n} = 0$. In this example the LDG flux with $\beta \in [0, 1/p]$ is used with mesh size of $h = 0.002$ and 3^{rd} order approximation.

The solution in Figure 3.8 shows that the method is able to capture the oblique discontinuity with variable convection velocity for this problem and two inner layers are approximated accurately. Figure 3.8 shows the comparison between our solution and the reference solution from John and Knobloch [2007]. Note that DOFs for our method is 2 998 000 and the reference solution is a $p2$ FEM with 16 785 409 DOFs. Thus, high-order HDG with the proposed strategy provides an accurate solution with less DOFs. Finally, table 3.5 compares the maximum overshoot and undershoot in the solution, with and without shock-capturing, and it demonstrates the reduction of the spurious overshoots and undershoots in the whole domain as a result of shock-capturing.

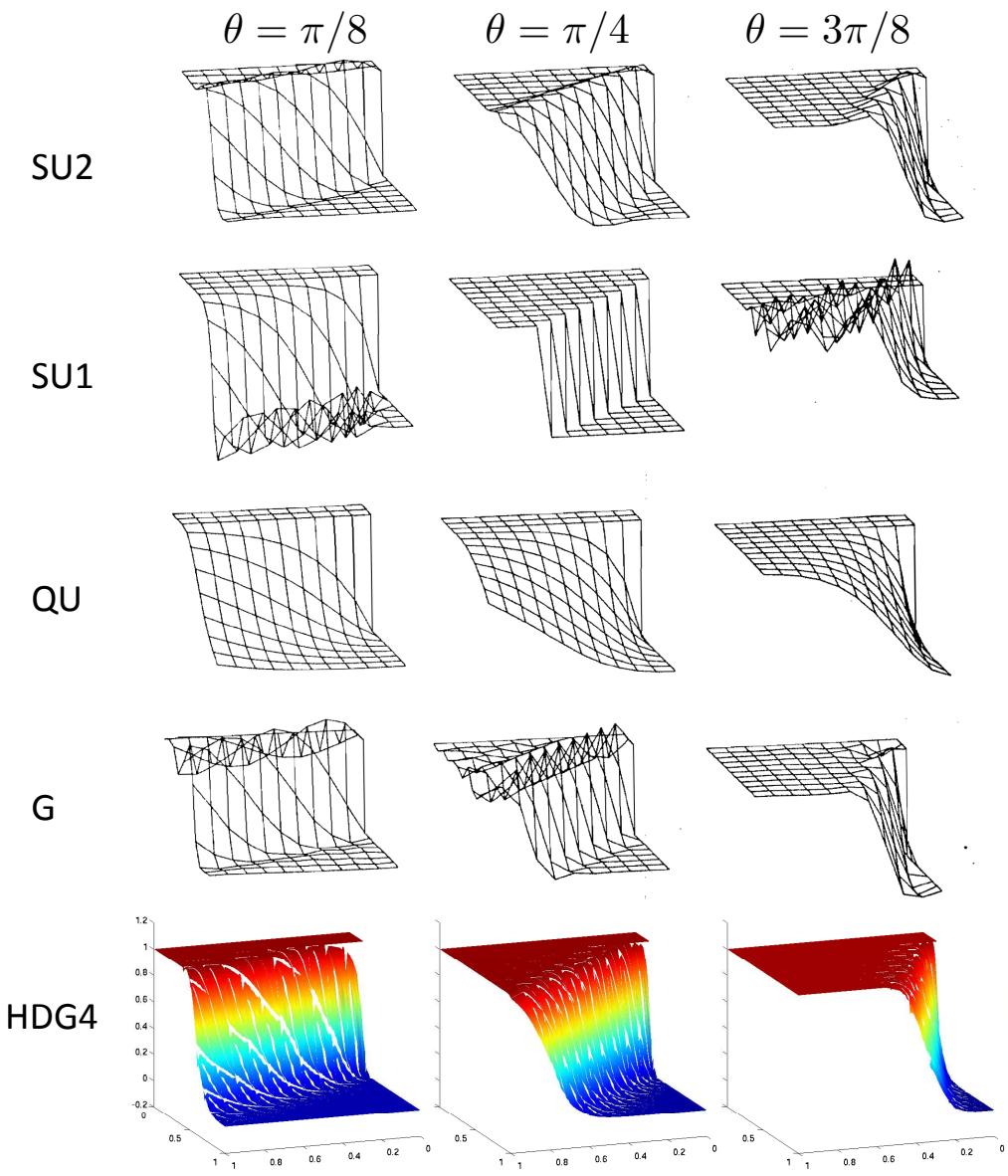


Figure 3.7: Convection dominated problem: Comparison between Reference [Brooks and Hughes, 1982] and 4^{th} order HDG and with $h = 0.1$, shock-capturing with LDG flux $\beta \in [0, 1/p]$

HDG		**HDG-SC**	
undershoot	**overshoot**	**undershoot**	**overshoot**
0.1424	0.1417	0.0525	0.0522

Table 3.5: Variable convection problem: comparison of undershoot and overshoot

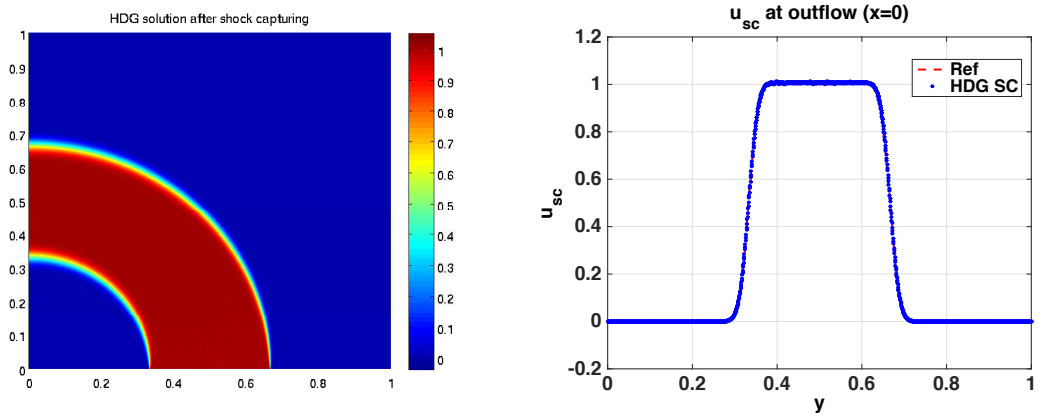


Figure 3.8: Variable convection problem: The HDG solution with shock-capturing and cross-section of the solution at outflow, for $p = 3$ approximation

3.2 High-order HDG with shock-capturing for compressible Navier-Stokes

In this section, the shock-capturing technique is extended to viscous compressible flow problems. The compressible Navier-Stokes equations, and the HDG formulation for them are presented in section 2.2 and here the same definitions and notations are used. The compressible Navier-Stokes equations in a discrete domain, stated in (2.27) are considered as a starting point here. The difference between the standard HDG, presented in section 2.2 and modified HDG for shock-capturing is described in the following sections. The same spaces of approximations and shape functions as in 3.1.1 are used to develop the shock-capturing for compressible Navier-Stokes equations.

3.2.1 HDG formulation with shock-capturing

Following the procedure in section 3.1.2, we need two different weak forms for HDG local problem: standard HDG weak form in standard elements, where no shock is detected, and modified HDG weak form in stabilized elements, detected by shock sensor. First, for standard elements, $e \notin \mathscr{E}_{\mathscr{S}}$, standard HDG discretization of the local problem in each element is presented in (2.35) in section 2.2.1.

For stabilized elements, $e \in \mathscr{E}_{\mathscr{S}}$, the modified approximation space with discontinuities inside the element (3.1) is considered, see section 3.1.1. Discretization of the

local problem in k^{th} sub-cell of e^{th} element is: given $\bar{\boldsymbol{U}}_h$ on ∂V_e^k, find approximation $(\boldsymbol{Q}_h, \boldsymbol{U}_h) \in [\widetilde{\mathcal{P}^p}(V_e^k)]^{(n_{sd}+2)n_{sd}} \times [\widetilde{\mathcal{P}^p}(V_e^k)]^{(n_{sd}+2)}$ such that

$$\begin{aligned}
(\boldsymbol{Q}_h, \boldsymbol{z})_{V_e^k} + (\boldsymbol{U}_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{V_e^k} - \langle \bar{\boldsymbol{U}}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial V_e^k \setminus \partial\Omega_e} - \langle \bar{\boldsymbol{U}}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial V_e^k \cap \partial\Omega_e} = 0,\\
(\frac{\partial \boldsymbol{U}_h}{\partial t}, \boldsymbol{r})_{V_e^k} - (\boldsymbol{F}_c(\boldsymbol{U}_h), \boldsymbol{\nabla}\boldsymbol{r})_{V_e^k} + (\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h), \boldsymbol{\nabla}\boldsymbol{r})_{V_e^k} +\\
\langle (\boldsymbol{F}_c(\bar{\boldsymbol{U}}_h) - \boldsymbol{F}_d(\bar{\boldsymbol{U}}_h, \boldsymbol{Q}_h))\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial V_e^k \cap \partial\Omega_e} + \langle \boldsymbol{S}(\boldsymbol{U}_h - \bar{\boldsymbol{U}}_h), \boldsymbol{r}\rangle_{\partial V_e^k \cap \partial\Omega_e} +\\
\langle \widetilde{(\boldsymbol{F}_c(\boldsymbol{U}_h) - \boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h))}\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial V_e^k \setminus \partial\Omega_e} = (\boldsymbol{f}, \boldsymbol{r})_{V_e^k},
\end{aligned} \tag{3.18}$$

for all $(\boldsymbol{z}, \boldsymbol{r}) \in [\widetilde{\mathcal{P}^p}(V_e^k)]^{(n_{sd}+2)n_{sd}} \times [\widetilde{\mathcal{P}^p}(V_e^k)]^{(n_{sd}+2)}$. It should be noted that here the boundary of sub-cell is divided into interior, $\partial V_e^k \setminus \partial\Omega_e$, and exterior part, $\partial V_e^k \cap \partial\Omega_e$. On the exterior part, which is part of the boundary of the element as well, the trace $\bar{\boldsymbol{U}}_h$ is set to the standard trace of the HDG solution $\hat{\boldsymbol{U}}_h$, (which is considered as a variable in the global problem), and the definition of numerical flux (2.32) is utilized. On the interior part, the trace $\bar{\boldsymbol{U}}_h$ is set to a new flux $\tilde{\boldsymbol{U}}_h$, and also a new total flux, $\widetilde{(\boldsymbol{F}_c(\boldsymbol{U}_h) - \boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h))}$ is introduced. These numerical fluxes transfer information across the inter-sub-cells boundaries and can be defined as a function of elemental unknowns, $\boldsymbol{U}_h$ and $\boldsymbol{Q}_h$, which will be discussed later.

Equation (3.18) is written for each sub-cell, and it should be summed over all sub-cells of the element to represent the local problem at each element. Summing equation (3.18) over sub-cells of the element, leads to a modified weak form for the local problem: given $\hat{\boldsymbol{U}}_h$ on $\partial\Omega_e$, find approximation $(\boldsymbol{Q}_h, \boldsymbol{U}_h) \in [\widetilde{\mathcal{P}^p}(\Omega_e)]^{(n_{sd}+2)n_{sd}} \times [\widetilde{\mathcal{P}^p}(\Omega_e)]^{(n_{sd}+2)}$ such that

$$\begin{aligned}
(\boldsymbol{Q}_h, \boldsymbol{z})_{\Omega_e} + (\boldsymbol{U}_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e} - \langle \tilde{\boldsymbol{U}}_h, [\![\boldsymbol{z}\cdot\boldsymbol{n}]\!]\rangle_{\Gamma_e} - \langle \hat{\boldsymbol{U}}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e} = 0,\\
(\frac{\partial \boldsymbol{U}_h}{\partial t}, \boldsymbol{r})_{\Omega_e} - (\boldsymbol{F}_c(\boldsymbol{U}_h), \boldsymbol{\nabla}\boldsymbol{r})_{\Omega_e} + (\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h), \boldsymbol{\nabla}\boldsymbol{r})_{\Omega_e} + \langle \boldsymbol{F}_c(\hat{\boldsymbol{U}}_h)\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial\Omega_e}\\
- \langle \boldsymbol{F}_d(\hat{\boldsymbol{U}}_h, \boldsymbol{Q}_h)\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial\Omega_e} + \langle \boldsymbol{S}\boldsymbol{U}_h, \boldsymbol{r}\rangle_{\partial\Omega_e} - \langle \boldsymbol{S}\hat{\boldsymbol{U}}_h, \boldsymbol{r}\rangle_{\partial\Omega_e}\\
+ \langle \widetilde{\boldsymbol{F}_c(\boldsymbol{U}_h)}, [\![\boldsymbol{n}\otimes\boldsymbol{r}]\!]\rangle_{\Gamma_e} - \langle \widetilde{\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h)}, [\![\boldsymbol{n}\otimes\boldsymbol{r}]\!]\rangle_{\Gamma_e} = (\boldsymbol{f}, \boldsymbol{r})_{\Omega_e},
\end{aligned} \tag{3.19}$$

for $e = 1, \cdots, n_{el}$, and for all $(\boldsymbol{z}, \boldsymbol{r}) \in [\widetilde{\mathcal{P}^p}(\Omega_e)]^{(n_{sd}+2)n_{sd}} \times [\widetilde{\mathcal{P}^p}(\Omega_e)]^{(n_{sd}+2)}$. It should be remembered that Γ_e is defined in (3.13). Like for the convection-diffusion equation, utilizing the stabilized HDG formulation for compressible Navier-Stokes equations leads to some additional terms which can be detected by comparing equations (2.35)

and (3.19). All additional terms, which add extra stabilization, contain jumps, and if a continuous basis of shape functions is used, then standard HDG formulation, (2.35) is recovered. Those jumps appear because in summing over sub-cells of the element, the interior boundary of sub-cell is involved twice, once from left, and once from right, hence, results in a jump over inter-sub-cell boundary.

Here, the definition for numerical flux of the LDG method from Persson [2012] is considered for diffusive part of the flux, while a Lax-Friedrichs flux is used for the convective part of the flux, see Atkins and Helenbrook [2009]. These definitions are presented in (3.20), where the coefficients C_{11}, $\boldsymbol{C}_{12}$ and C_{22} have effects on the stability and the convergence of the methods, for more details see Cockburn et al. [2002], Castillo et al. [2001]. The parameter λ in Lax-Friedrichs flux is the maximum absolute eigenvalue of the Jacobian of the convective part of flux.

$$\begin{aligned}
\tilde{\boldsymbol{U}}_h &= \{\boldsymbol{U}_h\} - \boldsymbol{C}_{12} \cdot [\![\boldsymbol{U}_h \otimes \boldsymbol{n}]\!] \\
\widetilde{\boldsymbol{F}_c(\boldsymbol{U}_h)} &= \{\boldsymbol{F}_c(\boldsymbol{U}_h)\} - \lambda [\![\boldsymbol{U}_h \otimes \boldsymbol{n}]\!]/2 \\
\widetilde{\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h)} &= \{\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h)\} - C_{11}[\![\boldsymbol{U}_h \otimes \boldsymbol{n}]\!] + \boldsymbol{C}_{12} \otimes [\![\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h) \cdot \boldsymbol{n}]\!]
\end{aligned} \tag{3.20}$$

It is important to note that, the weak form of the global problem is the same for standard elements and for discontinuous elements, which is presented by equation (2.36). The only difference is that basis of shape functions on the faces change from continuous ones to discontinuous ones and for continuous approximations, it is exactly the same as standard HDG method.

The HDG discrete problem, defined by either (2.35) or (3.19), and (2.36) is a system of Differential Algebraic Equations (DAE), with the same structure as standard HDG and, as usual, it can be efficiently discretized in time with an implicit time integrator, such as backward Euler or Crank-Nicolson method. Time discretization of local and global problems leads to *a non-linear system of equations* at each time step, that can be solved with an iterative scheme. Here, the non-linear system is solved using the Newton-Raphson method, like in 2.2.1.

3.2.2 Numerical results

In this section, numerical examples are presented for the HDG solution of compressible Navier-Stokes equations with the proposed shock-capturing technique. First, a

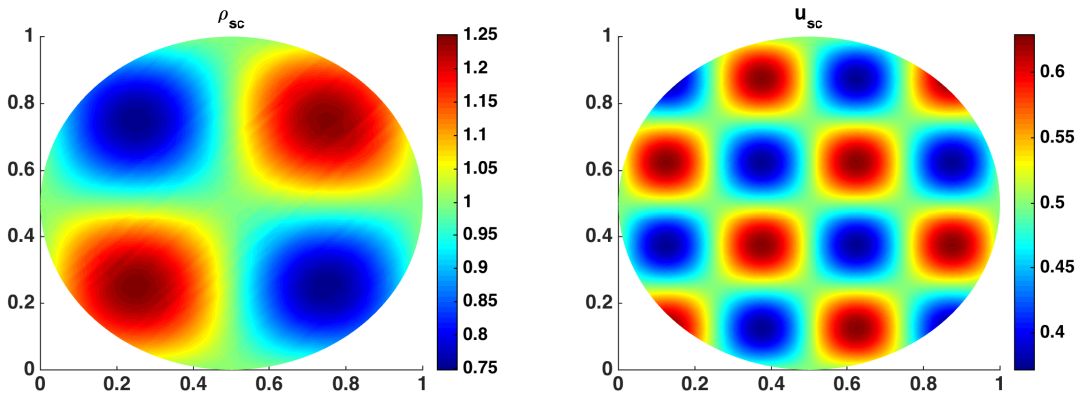
problem with smooth solution is considered, to check the ability of the shape functions to capture the solution and the convergence rate. Then, in a supersonic viscous flow around NACA 0012 airfoil, presence of shocks is studied and the ability of the method in stabilizing the oscillations in the vicinity of shocks is investigated.

Viscous flow in a circle: convergence test

The goal of this example is to investigate the ability of the discontinuous shape function approximation inside stabilized elements, and the corresponding modified HDG formulation, to capture the solution of compressible viscous flows. In a circular computational domain of radius 0.5, centred at $(0.5, 0)$, two-dimensional steady state solution of compressible Navier-Stokes equations is computed, with an inhomogeneous source term on right hand side of the equations, so that the analytical solution is provided by (2.42), with $\kappa = 2$.

The coefficients are set to be $\rho_0 = 1$, $u_0 = 0.5$, $v_0 = 0.5$ and $E_0 = 3$. Reynolds number is set to $Re = 1$ and on the boundary, Dirichlet boundary condition is imposed according to the exact solution, for more information see Wang and Anderson [2012].

The density and x-component of velocity are shown in Figure 3.9. For this numerical simulations, a computational mesh with 1916 elements of characteristic size $h = 0,032$ and polynomials of degree $p = 4$ is used. It can be seen that the frequency of the velocity is twice of the frequency of density.



(a) HDG-SC solution: density

(b) HDG-SC solution: x-velocity

Figure 3.9: HDG solution with LDG inter-sub-cell flux of a viscous flow in a circle, $h = 0.0317$ and $p = 4$

To check the convergence of the stabilized formulation, LDG flux is used on inter-

sub-cells boundaries with uniform $\beta = 1/p$ all over the domain. The L_2 norm of the error of the solution is compared for five different meshes, with degree $p = 3$ and 4. Table 3.6 shows the convergence of the solution and the convergence rate of HDG method with LDG inter-sub-cell flux. The results show that convergence is achieved and discontinuous shape functions are able to approximate the solution of non-linear compressible Navier-Stokes equations correctly.

mesh		$\boldsymbol{U}_h$	
h	**p**	**L2 error**	**order**
0.33	3	4.17e-01	
0.17	3	1.01e-02	**5.65**
0.10	3	4.40e-03	**1.53**
0.05	3	1.30e-03	**1.81**
0.03	3	3.43e-04	**1.93**
0.33	4	2.46e-02	
0.17	4	3.00e-03	**3.20**
0.10	4	1.40e-03	**1.40**
0.05	4	4.23e-04	**1.77**
0.03	4	1.16e-04	**1.88**

Table 3.6: L2 errors and rates of convergence using discontinuous shape functions with LDG flux with $\beta = 1/p$ for Navier-Stokes

Supersonic viscous flow around a NACA 0012 airfoil

In this section, supersonic viscous flow around a NACA 0012 airfoil is considered, hence, the presence of a bow shock in the solution can be studied. Like former examples of NACA 0012, in sections 2.2.2 and 2.3.2, for computational mesh, a circular domain with the radius ten times the cord of the airfoil is subdivided into triangles.

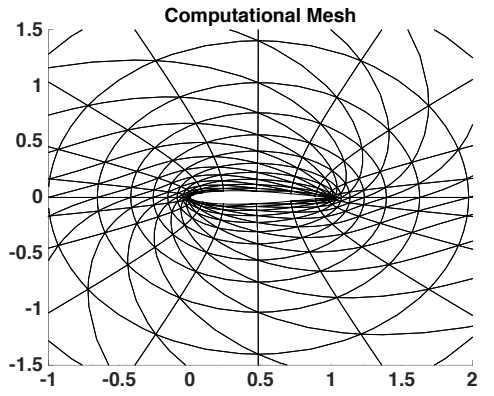
First, for $Re_\infty = 1000$, $M_\infty = 1.2$ and $\alpha = 0$, as the test case in Hartmann [2005a], the shock-capturing is used to calculate the supersonic viscous flow on the computational mesh of 640 elements with degree $p = 4$. The unsteady code is used to solve this problem, with backward Euler time marching and time step size of $\Delta t = 2.5 \times 10^{-2}$. A subsonic steady state solution of $Re_\infty = 1000$ and $M_\infty = 0.9$ is employed as initial condition. Then, the unsteady solver is marched for 20s in time, with a linearly increasing Mach number and one iteration of non-linear solver per time steps, to compute the final supersonic solution. The mesh is shown in Figure 3.10(a), and the convergence of the solution is demonstrated in Figure 3.10(b), where $|d\boldsymbol{U}|/|\boldsymbol{U}|$

represents the normalized increment solution at each time step. The distribution of Mach number and density are presented in Figures 3.10(e) and 3.10(f) respectively. The shock in the solution can be detected at a distance in front of the airfoil, in both Mach and density. It should be noted that, to compute this solution, discontinuous shape functions are used in detected elements, as presented in Figure 3.10(d), which is based on the smoothness of the solution, as presented in 3.10(c).

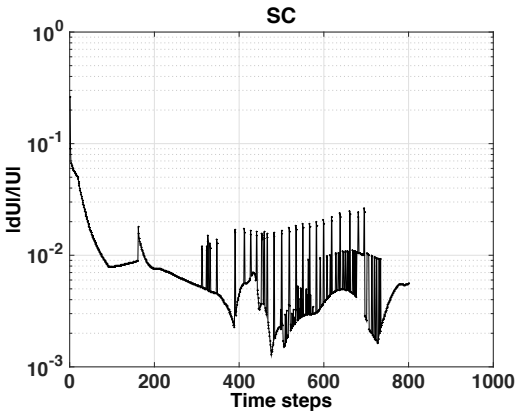
In the work of Hartmann [2005a], which uses DG for compressible Navier-Stokes with shocks, it is reported that DG discretization for this problem can be solved even without any shock-capturing as long as numerical dissipation is sufficiently large. So an effort has been done to compute the solution without shock-capturing, and as expected, HDG can also be applied to solve this problem without shock-capturing, but not for very strong shocks. Figure 3.11, shows the cross section of the pressure and density, at line $y = 0$, in front of airfoil, where the bow shock is located. Figures 3.11(a) and 3.11(b), show the pressure and density, respectively, for mesh of 640 elements of degree $p = 4$ and Figures 3.11(c) and 3.11(d), show the same for mesh of 560 elements of degree $p = 5$. It can be noted that in both cases, employing the shock-capturing reduces the oscillations in the vicinity of the shock.

The computation of supersonic viscous flow around NACA 0012 on the same domain is repeated for different far-field conditions. This time, $Re_\infty = 2000$, $M_\infty = 1.2$ and and $\alpha = 0$ is chosen to see a sharper, stronger shock. The unsteady code is used to solve this problem, with backward Euler time marching and time step size of $\Delta t = 2.5 \times 10^{-2}$. A supersonic steady state solution of $Re_\infty = 1000$, $M_\infty = 1.2$ and and $\alpha = 0$ is employed as initial condition. Then, the unsteady solver is marched for 10s in time, with a linearly increasing Mach number and one iteration of non-linear solver per time step, to compute the final supersonic solution. This time, HDG without shock-capturing is not able to capture the solution and utilization of shock-capturing is mandatory. Figures 3.12(a) and 3.11(d) show the Mach number and density around the airfoil and the presence of strong shock is clear in both solutions.

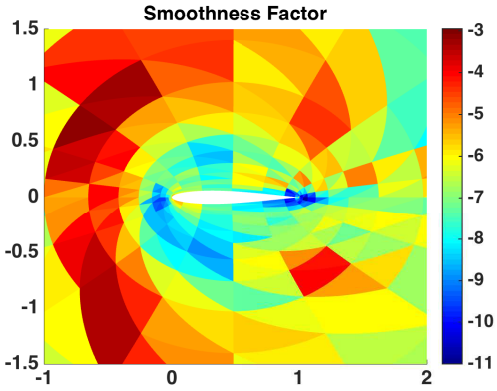
This example of NACA 0012 airfoil shows that, the proposed method is able to capture the shock in supersonic viscous flow, and it either improves the solution on shocks by reducing the oscillations, or provides extra stabilization to capture the sharp solution in case of strong shocks.



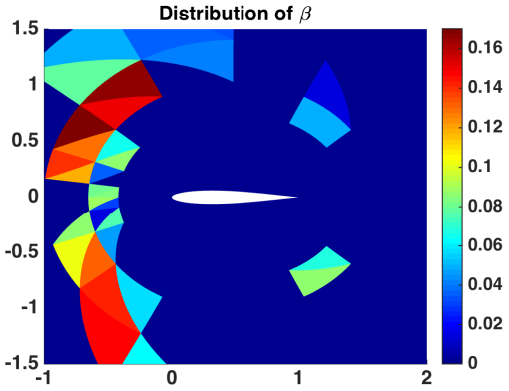
(a) Computational mesh



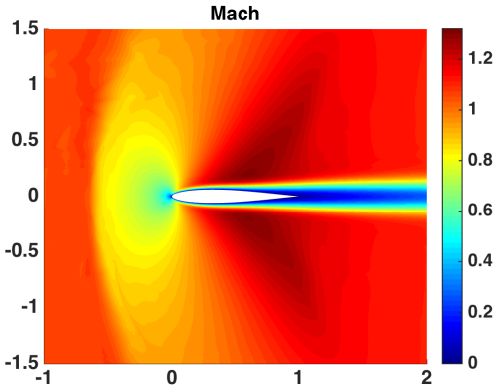
(b) Convergence of solution



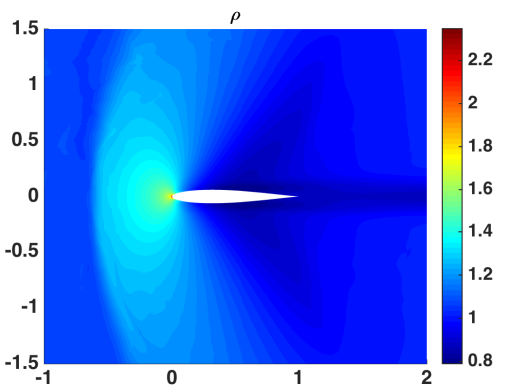
(c) Smoothness factor



(d) distribution of β

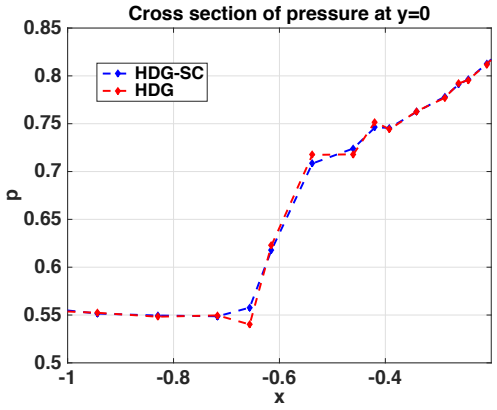


(e) Mach

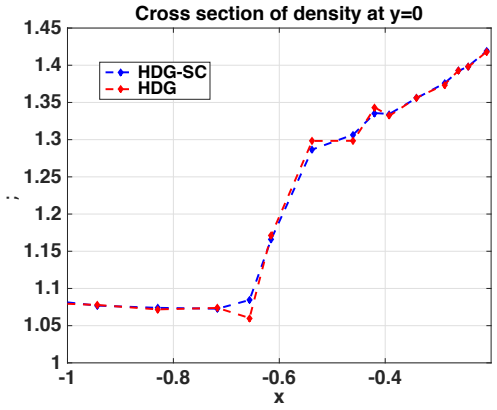


(f) Density

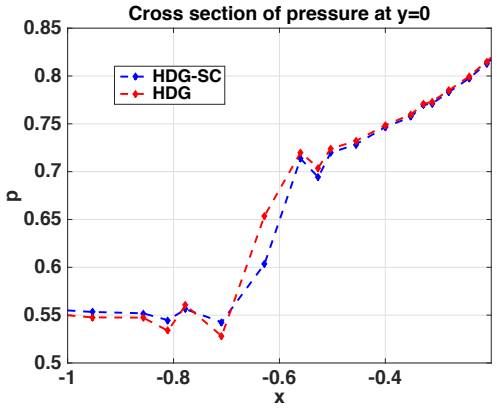
Figure 3.10: Supersonic flow around a NACA 0012 airfoil, $M_\infty = 1.2$, $Re_\infty = 1200$ and $\alpha = 0$, computational mesh of 640 elements of degree $p = 4$



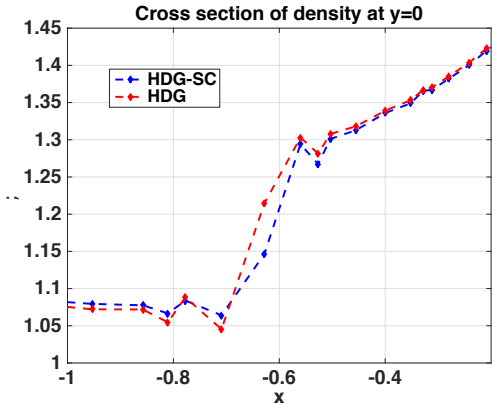
(a) Cross section of pressure, mesh of 640 elements of degree $p = 4$



(b) Cross section of density, mesh of 640 elements of degree $p = 4$



(c) Cross section of pressure, mesh of 560 elements of degree $p = 5$



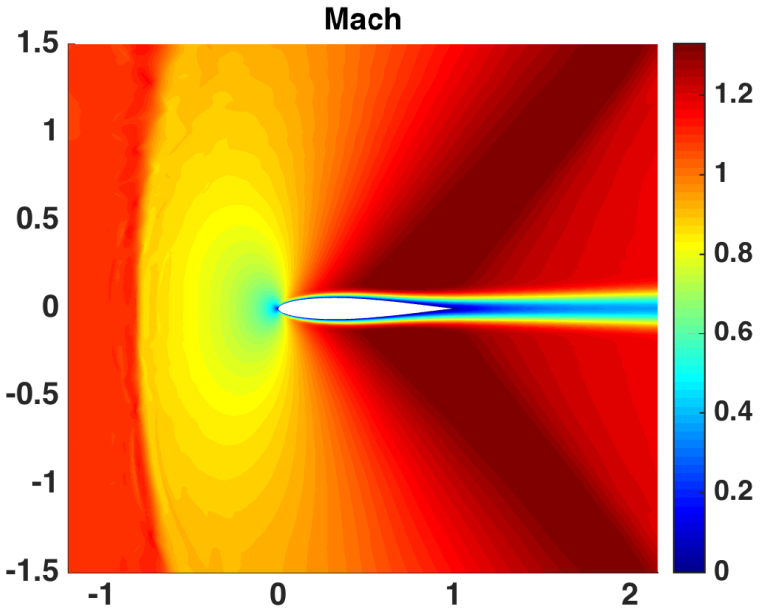
(d) Cross section of density, mesh of 560 elements of degree $p = 5$

Figure 3.11: Comparison of solution with and without shock-capturing

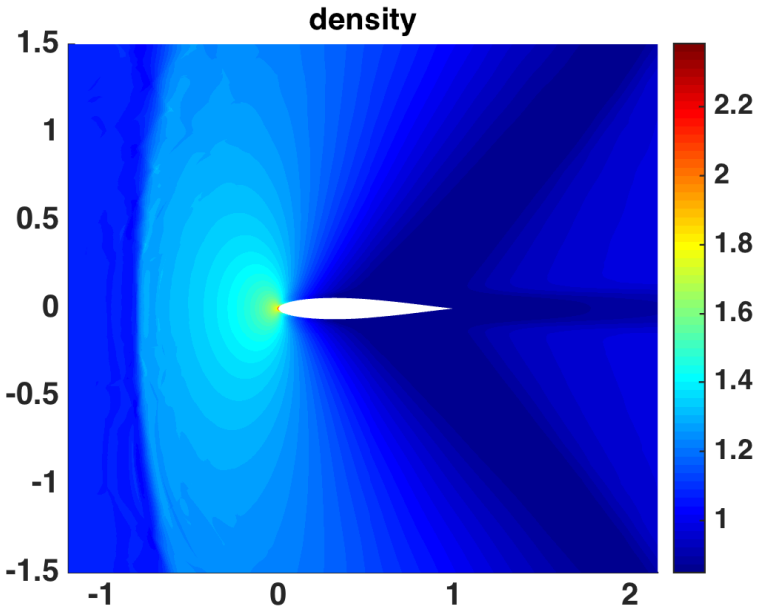
3.3 Conclusions

In this chapter, a new shock-capturing technique is proposed for the HDG method. Based on the work of Huerta et al. [2012], a new space of approximation is considered for discretization, in stabilized elements and their faces, which requires a proper modified weak form of HDG local problem at stabilized elements. This new space of approximation allows utilization of discontinuous shape functions, hence, imposing additional stabilization inside high-order elements. The extra stabilization is in form of inter-sub-cell fluxes, and it varies with level of smoothness of the local solution, and works via a shock detector.

First, the method is applied to linear convection-diffusion equations with inner



(a) Mach



(b) Density

Figure 3.12: Supersonic flow around a NACA 0012 airfoil, $M_\infty = 1.2$, $Re_\infty = 2000$ and $\alpha = 0$, computational mesh of 2560 elements of degree $p = 3$

layers, and the numerical results show that shock-capturing with LDG type of inter-sub-cell flux works very well in reduction of oscillations in the vicinity of sharp fronts, for different meshes and degree of approximations. Thus, the method proves to enable usage of high-order approximations on coarse meshes, even for convection-diffusion problems with sharp fronts, without need for adaptation of mesh or additional DOFs.

Then, the method is extended to compressible Navier-Stokes equations, in which we may encounter shocks in transonic or supersonic regimes. The proposed method is proved to be able to capture the shock in supersonic viscous flow problems. In the presence of not so sharp fronts, for which HDG without shock-capturing converges, HDG with shock-capturing improves the results and reduces the oscillations in the vicinity of shocks. In the presence of strong shocks, when HDG without shock-capturing can not converge, HDG with proposed shock-capturing technique is able to provide a stable solution. Overall, again, the novel shock-capturing strategy enables usage of coarse high-order elements without additional change in mesh topology, for compressible viscous flow computations.

Chapter 4

Summary and future developments

This thesis presents the HDG method for compressible viscous flow computations with shocks. The HDG discretization is applied to convection-diffusion and compressible Navier-Stokes equations in sections 2.1 and 2.2 respectively, and in both cases, the accuracy of the method, and efficiency of high-order approximation versus low-order one are studied. For compressible Navier-Stokes equations, both steady and unsteady solvers are developed using a Newton-Raphson non-linear solver, and a continuation method or time relaxation can be employed to reach the steady state solution at high Reynolds numbers. Development of the code for these parts of the work, is a necessary platform for the contributions of this thesis.

In addition, a linearization method, based on the linear extrapolation from solutions of previous time steps, is proposed. It is developed for unsteady compressible Navier-Stokes equations with HDG discretization in space, and Crank-Nicolson discretization in time, as presented in section 2.3. The numerical results for NACA 0012 problem and unsteady vortex convection show that, the proposed method is capable of computing accurate solution of compressible viscous flow, for both steady and transient problems, and the results are compared to the non-linear steady and transient solvers. However, the extrapolation from previous time steps, imposes a limitation on time step size, and for bigger time step sizes, convergence of the method may be lost. Thus, even using an implicit time discretization can not grantee freedom of

choice for time marching, and high number of time steps are necessary, and hence, computational cost is considerably high, specially for steady problems.

The main contributions of this thesis can be summarized as follows.

• **Shock-capturing HDG for convection-diffusion problems:** High-order HDG for steady convection-diffusion problems with sharp fronts is developed in section 3.1. The discretization space and discontinuous shape functions, for both standard elements and stabilized elements are developed, and those two types of elements are distinguishable by employment of a shock detector, which is based on the work of Persson and Peraire [2006]. Discontinuous shape functions are produced by utilization of non-overlapping sub-cells, resulting in piecewise constant functions over sub-cells, ϕ_i^e. Then, a linear combination of these piecewise constant functions, and standard p^{th} order continuous shape functions (with p being the order of approximation), N_i^e, over the element, leads to a modified shape functions, $\tilde{N}_i^e$. This combination, depends on a scalar elemental parameter, β, which is based on the local smoothness of the solution, the higher the β, the more discontinuous shape functions, and more stabilization inside the element. For shape functions on faces, the minimum value of β of the two neighbouring element are considered, hence, less stabilization is added. The employment of these new shape functions, leads to a modified weak form of the HDG local problem, with new terms on the inter-sub-cells boundaries, which contain numerical fluxes and jumps. For convection-diffusion problems, Bassi-Rebay and LDG fluxes are used for these fluxes inside the elements.

The discontinuous shape functions are tested for a smooth convection-diffusion problem, and convergence is obtained for uniform distribution of discontinuous shape functions. Then, problems with one and tow inner layers in very convective regimes are studied. The results show that, the proposed shock-capturing technique is able to substantially reduce the oscillations of high-order approximations in the vicinity of sharp layers, without increasing DOFs of the problem or changing the mesh topology. It worth mentioning that, the LDG flux with $\beta \in [0, 1/p]$ yields best results, while higher values lead to over-stabilization and damping the sharp front.

• **Shock-capturing HDG for compressible Navier-Stokes equations:** The strategy of utilizing discontinuous shape functions for extra stabilization, is extended to the compressible Navier-Stokes equations in the presence of shocks,

and it is developed in section 3.2. Again, discontinuous shape functions are built based on the local smoothness of the solution and employment of these discontinuous shape functions leads to a modified weak form of the HDG local problem, while the global problem remains the same as standard HDG. New terms in the local problem are on the inter-sub-cells boundaries, and contain numerical fluxes and jumps. Based on the DG fluxes used in the literature, for compressible Navier-Stokes, here we consider LDG type flux for diffusive part of the flux, and Lax-Friedrichs type for convective part.

A test case of smooth viscous compressible flow with exact solution is used to study the possible convergence of discontinuous shape functions. The result of this example demonstrates that, convergence can be obtained for uniform distribution of β over all domain, while the accuracy of the proposed method can beat first order accuracy of slope limiting techniques. Then, the supersonic viscous flow around NACA 0012 airfoil is studied, and the shock-capturing technique is proved to be able to produce non-oscillatory solutions in presence of the bow shock in front of the airfoil. It should be noted that the inherent stability of HDG method provides enough dissipation to capture not so strong shocks in lower Reynolds numbers and on coarser meshes. Even in these cases, the proposed shock-capturing can improve the solution and reduce the oscillations in the vicinity of shocks. Finally, for higher Reynolds numbers, when HDG method can not capture the solution in the presence of a strong shock, the shock-capturing technique obtains the solution and stabilizes the method. This novel approach, doesn't need any tuning, and choice of shape functions is automatically performed. On the other hand, no mesh adaptivity is required and large high-order elements can be utilized even in the vicinity of strong shocks and they can be captured inside the element.

4.1 Future developments

• **Study of piecewise linear shape functions inside the sub-cells:** In order to add extra stabilization inside the elements, piecewise constant shape functions are combined with normal continuous ones. The numerical tests for both convection-diffusion and Navier-Stokes equations show that the extreme case of piecewise constant shape functions are not needed, and instead less

stabilization of $\beta \in [0, 1/p]$ is better fitted for the purpose of shock-capturing. *Piecewise linear shape functions* in the sub-cells, instead of piecewise constant shape functions, provide another option to introduce extra stabilization inside elements, which can even increase the accuracy of the method in the vicinity of shocks. The idea can be implemented quite easily, and the changes will be minimum, hence, worth being considered.

• **Extension to turbulent compressible flows:** The compressible viscous flow problems, tackled in this thesis, can be extended to more realistic engineering problems of high-speed high-Reynolds flows, where strong shocks are more frequent. In order to do that, including turbulence will be a vital step to accurately capture important flow features, like shocks and boundary layers. The RANS models are the first candidates, which can be handled by adding extra equations to the system. Some work has been done to include Spalart-Allmaras model in the work of Moro et al. [2011], but this is a new field of research and many more models can be employed, and also one can consider LES and DES, specially for strong separations or off-design situations.

• **Parallelization and Extension to 3D:** Another important step, which can enhance the current work is 3D implementation of the method. This will enable more complex geometries of real engineering problems, like 3D wings and bodies to be considered. The codes for this thesis are written in Matlab and they are optimized substantially, however, further boost of performance needs a change of the existing codes from Matlab to Fortaran or C++. The parallelization of codes will allow to compute the compressible viscous flow problems within a suitable computational cost.

• **Adaptivity:** Complex flow features in compressible viscous flow problems, like shocks, boundary layers and wakes require different mesh sizes and order of approximations for efficient numerical simulations. The computational cost of complex simulations can be optimized by adapting the mesh or polynomial degrees, in different regions of solution, which is easier to achieve in DG families, because discontinuous space of approximation between the elements allows local adaptivity with ease.

On the other hand, goal oriented adaptivity by defining quantities of interests, can be implemented to improve the performance of the method and reduce user defined parameters. In compressible Navier-Stokes problems, out-

puts of interests, for instance aerodynamics coefficients of an airfoil, which are functions of the solution, are used to estimate the error of numerical simulations and providing bound for it. In context of DG, works of Hartmann [2005a] is an example which can be extended to HDG.

Bibliography

J. D. Anderson. *Computational Fluid Dynamics*. McGraw-Hill, 1 edition, 1995.

J. D. Anderson. *Fundamentals of Aerodynamics*. Anderson series. McGraw-Hill Education, 2010.

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.

H. L. Atkins and B. Helenbrook. Super-convergence of discontinuous Galerkin method applied to the Navier-Stokes equations. *AIAA Paper*, 3787:2009, 2009.

G. A. Baker. Galerkin approximation for the Navier-Stokes equations. Technical report, Harvard university, 1976.

G. A. Baker, V. A. Dougalis, and O. A. Karakashian. On a higher order accurate fully discrete Galerkin approximation to the Navier-Stokes equations. *Mathematics of Computation*, 39(160):339–375, 1982.

B. Baldwin and R. MacCormack. Interaction of strong shock wave with turbulent boundary layer. In *Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics*, volume 35 of *Lecture Notes in Physics*. 1975.

G. Barter. *Shock capturing with PDE-based artificial viscosity for an adaptive, higher-order discontinuous Galerkin finite element method*. PhD thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2008.

G. E. Barter and D. L. Darmofal. Shock capturing with pde-based artificial viscosity for dgfem: Part I. formulation. *Journal of Computational Physics*, 229(5):1810 – 1827, 2010.

F. Bassi and S. Rebay. Accurate 2d Euler computations by means of a high order discontinuous finite element method. In *Fourteenth International Conference on Numerical Methods in Fluid Dynamics*, pages 234–240. Springer, 1995.

F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131(2), 1997a.

F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics*, 138(2):251 – 285, 1997b.

F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the reynolds-averaged Navier-Stokes and k- turbulence model equations. *Computers & Fluids*, 34(4-5):507 – 540, 2005.

F. Bassi, A. Crivellini, and A. G. S. Rebay. High-order discontinuous Galerkin discretization of transonic turbulent flows. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, Junuary 2009.

C. Baumann and J. T. Oden. An adaptive-order discontinuous Galerkin method for the solution of the Euler equations of gas dynamics. *International Journal for Numerical Methods in Engineering*, 47(1-3):61–73, 2000.

C. E. Baumann and J. T. Oden. A discontinuous hp finite element method for the Euler and Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 31(1):79–95, 1999.

P. Birken, G. Gassner, M. Haas, and C.-D. Munz. Efficient time integration for discontinuous Galerkin methods for the unsteady 3d Navier-Stokes equations. In *European Congress on Computational Methods and Applied Sciences and Engineering (ECCOMAS 2012), number Eccomas*, 2012.

M. Borrel and J. Ryan. The elastoplast discontinuous Galerkin (EDG) method for the Navier-Stokes equations. *J. Comput. Phys.*, 231(1):1–22, 2012.

A. Brooks and T. 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.

N. Burgess. *An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows*. PhD thesis, Department of Mechanical Engineering, University of Wyoming, 2011.

N. K. Burgess and D. J. Mavriplis. Computing shocked flows with high-order accurate discontinuous Galerkin methods. In *42nd AIAA Fluid Dynamics Conference and Exhibit*, number AIAA-2012-2715, New Orleans, Louisiana, June 2012.

J. Cao. Application of a posteriori error estimation to finite element simulation of compressible Navier-Stokes flow. *Computers & Fluids*, 34(8):991 – 1024, 2005.

J. Carrero, B. Cockburn, and D. Schötzau. Hybridized globally divergence-free ldg methods. part i: The stokes problem. *Mathematics of computation*, 75(254):533–563, 2006.

E. Casoni, J. Peraire, and A. Huerta. One-dimensional shock-capturing for high-order Discontinuous Galerkin methods. *Int. J. Numer. Methods Fluids*, 71(6):737–755, 2013.

P. Castillo, B. Cockburn, I. Perugia, and D. Schtzau. An a priori error analysis of the local discontinuous Galerkin method for elliptic problems. *SIAM Journal on Numerical Analysis*, 38(5):pp. 1676–1706, 2001.

F. Celiker, B. Cockburn, and K. Shi. Hybridizable discontinuous Galerkin methods for timoshenko beams. *Journal of Scientific Computing*, 44(1):1–37, 2010.

F. Celiker, B. Cockburn, and K. Shi. A projection-based error analysis of HDG methods for timoshenko beams. *Mathematics of Computation*, 81(277):131–151, 2012.

B. Chabaud and B. Cockburn. Uniform-in-time superconvergence of HDG methods for the heat equation. *Mathematics of Computation*, 81(277):107–129, 2012.

F. Chalot and P.-E. Normand. Towards high-fidelity industrial cfd. In *5th ECCOMAS European Conference on Computational Fluid Dynamics, Lisbon, Portugal*, 2010.

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

B. Cockburn. Discontinuous Galerkin methods for computational fluid dynamics. *Encyclopedia of Computational Mechanics*, 2004.

B. Cockburn and J. Cui. An analysis of HDG methods for the vorticity-velocity-pressure formulation of the stokes problem in three dimensions. *Mathematics of Computation*, 81(279):1355–1368, 2012.

B. Cockburn and J. Gopalakrishnan. A characterization of hybridized mixed methods for second order elliptic problems. *SIAM J. Numer. Anal*, 42(1):283–301, 2004.

B. Cockburn and J. Gopalakrishnan. Error analysis of variable degree mixed methods for elliptic problems via hybridization. *Mathematics of computation*, 74(252):1653–1677, 2005.

B. Cockburn and J. Gopalakrishnan. The derivation of hybridizable discontinuous Galerkin methods for stokes flow. *SIAM Journal on Numerical Analysis*, 47(2): 1092–1125, 2009.

B. Cockburn and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. general framework. *Mathematics of Computation*, 52(186):411–435, 1989.

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

B. Cockburn and C. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16:173–261, 2001.

B. Cockburn, S. Lin, and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.

B. Cockburn, S. Hou, and C. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. the multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.

B. Cockburn, G. Karniadakis, and C.-W. Shu. The development of discontinuous Galerkin methods. In B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 3–50. Springer Berlin Heidelberg, 2000.

B. Cockburn, G. Kanschat, I. Perugia, and D. Schtzau. Superconvergence of the local discontinuous Galerkin method for elliptic problems on cartesian grids. *SIAM Journal on Numerical Analysis*, 39(1):pp. 264–285, 2002.

B. Cockburn, B. Dong, and J. Guzmán. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Mathematics of Computation*, 77(264):1887–1916, 2008.

B. Cockburn, B. Dong, and J. Guzmán. A hybridizable and superconvergent discontinuous Galerkin method for biharmonic problems. *Journal of Scientific Computing*, 40(1-3):141–187, 2009a.

B. Cockburn, B. Dong, J. Guzmán, M. Restelli, and R. Sacco. A hybridizable discontinuous Galerkin method for steady-state convection-diffusion-reaction problems. *SIAM J. Sci. Comput.*, 31:3827–3846, 2009b.

B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 47(2):1319–1365, 2009c.

B. Cockburn, J. Guzmán, and H. Wang. Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Mathematics of Computation*, 78(265): 1–24, 2009d.

B. Cockburn, J. Guzmn, S.-C. Soon, and H. Stolarski. An analysis of the embedded discontinuous Galerkin method for second-order elliptic problems. *SIAM J. Numerical Analysis*, 47(4):2686–2707, 2009e.

B. Cockburn, J. Gopalakrishnan, N. Nguyen, J. Peraire, and F.-J. Sayas. Analysis of HDG methods for stokes flow. *Mathematics of Computation*, 80(274):723–760, 2011.

B. Cockburn, O. Dubois, J. Gopalakrishnan, and S. Tan. Multigrid for an HDG method. *IMA Journal of Numerical Analysis*, 34(4):1386–1425, 2014.

V. Dolej. Discontinuous Galerkin finite element method for the numerical solution of viscous compressible flows. In *Numerical Mathematics and Advanced Applications*, pages 260–268. Springer Berlin Heidelberg, 2004.

C. Drozo, M. Borrel, and A. Lerat. Discontinuous Galerkin schemes for the compressible Navier-Stokes equations. In *Sixteenth International Conference on Numerical Methods in Fluid Dynamics*, volume 515 of *Lecture Notes in Physics*, pages 266–271. Springer Berlin Heidelberg, 1998.

H. Egger and J. Schberl. A hybrid mixed discontinuous Galerkin finite-element method for convection-diffusion problems. *IMA Journal of Numerical Analysis*, 30(4):1206–1234, 2010.

X. Feng and Y. Xing. Absolutely stable local discontinuous Galerkin methods for the helmholtz equation with large wave number. *Mathematics of Computation*, 82 (283):1269–1296, 2013.

K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 207(1):92–113, July 2005.

G. Giorgiani, S. Fernández-Méndez, and A. Huerta. Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems. *International Journal for Numerical Methods in Fluids*, 72(12):1244–1262, 2013a.

G. Giorgiani, D. Modesto, S. Fernández-Méndez, and A. Huerta. High-order continuous and discontinuous Galerkin methods for wave problems. *International Journal for Numerical Methods in Fluids*, 73(10):883–903, 2013b.

G. Giorgiani, S. Fernández-Méndez, and A. Huerta. Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible Navier-Stokes equations. *Computers & Fluids*, 98(0):196 – 208, 2014.

R. Griesmaier and P. Monk. Discretization of the wave equation using continuous elements in time and a hybridizable discontinuous Galerkin method in space. *Journal of Scientific Computing*, 58(2):472–498, 2014.

A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 131(1):3 – 47, 1987.

R. Hartmann. Discontinuous Galerkin methods for compressible flows:higher order accuracy, error estimation and adaptivity. In *VKI LS 2006-01: CFD-Higher Order Discretization Methods*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2005a.

R. Hartmann. Discontinuous Galerkin methods for compressible flows: higher order accuracy, error estimation and adaptivity. In H. Deconinck and M. Ricchiuto, editors, *VKI LS 2006-01: CFD-Higher Order Discretization Methods, Nov. 14-18, 2005*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2005b.

R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 51(9-10):1131–1156, 2006.

R. Hartmann. Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow. *International Journal for Numerical Methods in Fluids*, 72(8):883–894, 2013.

R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183(2):508–532, 2002.

R. Hartmann and P. Houston. Error estimation and adaptive mesh refinement for aerodynamic flows. In H. Deconinck, editor, *VKI LS 2010-01: 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.

R. Hartmann, J. Held, T. Leicht, and F. Prill. Discontinuous Galerkin methods for computational aerodynamics 3D adaptive flow simulation with the {DLR} {PADGE} code. *Aerospace Science and Technology*, 14(7):512 – 519, 2010.

Y. He and W. Sun. Stabilized finite element method based on the Crank-Nicolson extrapolation scheme for the time-dependent Navier-Stokes equations. *Mathematics of Computation*, 76(257):115–136, 2007.

A. Huerta, E. Casoni, and J. Peraire. A simple shock-capturing technique for high-order Discontinuous Galerkin methods. *Int. J. Numer. Methods Fluids*, 69(10): 1614–1632, 2012.

A. Huerta, A. Angeloski, X. Roca, and J. Peraire. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, 96(9):529–560, 2013.

T. J. R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: IV. a discontinuity-capturing operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 58(3):329 – 336, 1986a.

T. J. R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: III. the generalized streamline operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 58(3):305 – 328, 1986b.

T. J. R. Hughes, L. P. Franca, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II beyond supg. *Comput. Methods Appl. Mech. Eng.*, 54:341–355, 1986.

T. J. R. Hughes, L. P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: VI. convergence analysis of the generalized SUPG formulation for linear time-dependent multi-dimensional advective-diffusive systems. *Comput. Methods Appl. Mech. Eng.*, 63(1):97–112, 1987.

A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. In *AIAA, Fluid and Plasma Dynamics Conference*, number AIAA 81-1259, 1981.

A. Jaust and J. Schtz. A temporally adaptive hybridized discontinuous Galerkin method for time-dependent compressible flows. *Computers & Fluids*, 98:177 – 185, 2014.

Z.-H. Jiang, C. Yan, J. Yu, and W. Yuan. Hermite WENO-based limiters for high order discontinuous Galerkin method on unstructured grids. *Acta Mechanica Sinica*, 28(2):241–252, 2012.

V. John and P. Knobloch. On the performance of sold methods for convection-diffusion problems with interior layers. *International Journal of Computing Science and Mathematics*, 1(2-4):245–258, 2007.

F. Kikuchi, K. Ishii, and I. Oikawa. Discontinuous Galerkin fem of hybrid displacement type - development of polygonal elements -. *Theoretical and Applied Mechanics Japan*, 57:395–404, 2009.

R. Kirby, S. Sherwin, and B. Cockburn. To CG or to HDG: A comparative study. *Journal of Scientific Computing*, pages 1–30, 2011.

C. Klaij, J. van der Vegt, and H. van der Ven. Pseudo-time stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 219(2):622 – 643, 2006a.

C. Klaij, J. van der Vegt, and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 217(2):589 – 611, 2006b.

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. *Appl. Numer. Math.*, 48(3-4):323–338, Mar. 2004.

N. Kroll. ADIGMA - a european project on the development of adaptive higher order variational methods for aerospace applications. In *ECCOMAS CFD 06 - European Conference on Computational Fluid Dynamics*, TU Delft, Netherlands, 2006.

N. Kroll. The adigma project. In *ADIGMA-A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, pages 1–9. Springer, 2010.

N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sorensen. *ADIGMA–A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications: Results of a Collaborative Research Project Funded by the European Union, 2006-2009*, volume 113. Springer Science & Business Media, 2010.

A. Labovsky, W. J. Layton, C. C. Manica, M. Neda, and L. G. Rebholz. The stabilized extrapolated trapezoidal finite-element method for the Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 198(9-12):958 – 974, 2009.

P. Lasaint and P. Raviart. On a finite element method for solving the neutron transport equation. In C. d. Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89 – 123. Academic Press, 1974.

R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhuser Basel, 1992.

Q. Liu and Y. Hou. A two-level finite element method for the Navier-Stokes equations based on a new projection. *Applied Mathematical Modelling*, 34(2):383 – 399, 2010.

X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.

I. Lomtev and G. E. Karniadakis. A discontinuous Galerkin method for the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 29(5): 587–603, 1999.

H. Luo, J. B. D., and R. Löhner. A hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. *J. Comput. Phys.*, 225(1):686–713, 2007.

H. Luo, L. Luo, R. Nourgaliev, V. A. Mousseau, and N. Dinh. A reconstructed discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids. *J. Comput. Phys.*, 229(19):6961–6978, 2010.

A. Montlaur and G. Giorgiani. Numerical study of 2d vertical axis wind and tidal turbines with a degree-adaptive hybridizable discontinuous Galerkin method. In E. Ferrer and A. Montlaur, editors, *CFD for Wind and Tidal Offshore Turbines*, Springer Tracts in Mechanical Engineering, pages 13–26. Springer International Publishing, 2015.

D. Moro, N. Nguyen, and J. Peraire. Navier-Stokes solution using hybridizable discontinuous Galerkin methods. *AIAA paper*, 3407:2011, 2011.

D. Moro, N. C. Nguyen, and J. Peraire. A hybridized discontinuous petrov-Galerkin scheme for scalar conservation laws. *International Journal for Numerical Methods in Engineering*, 91(9):950–970, 2012.

N. 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, 2009a.

N. 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, 2009b.

N. 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.

N. Nguyen, J. Peraire, and B. Cockburn. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics*, 230(10):3695 – 3718, 2011a.

N. 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, 2011b.

N. C. Nguyen and J. Peraire. An adaptive shock-capturing HDG method for compressible flows. Number AIAA-2011-3060, Honolulu, HI, June 2011.

N. C. Nguyen and J. Peraire. Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. *Journal of Computational Physics*, 231(18):5955–5988, 2012.

N. C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods. In J. S. Hesthaven and E. M. Rnquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 63–84. Springer Berlin Heidelberg, 2011c.

I. Oikawa. Hybridized discontinuous Galerkin method for convection-diffusion problems. *Japan Journal of Industrial and Applied Mathematics*, 31(2):335–354, 2014.

T. A. Oliver. *A high-order, adaptive, discontinuous Galerkin finite element method for the Reynolds-Averaged Navier-Stokes equations*. PhD thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2008.

J. Peraire, N. Nguyen, and B. Cockburn. A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations. In *48th AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2010-363, Orlando, Florida, January 2010.

P.-O. Persson. High-order Navier-Stokes simulations using a sparse line-based discontinuous Galerkin method. In *Proc. of the 50th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2012-456*, 2012.

P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting*, number AIAA-2006-0112, Reno, Nevada, Junuary 2006.

J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, 193(1):115 – 135, 2004.

J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two dimensional case. *Computers & Fluids*, 34(6):642 – 663, 2005.

W. Qiu and K. Shi. A superconvergent HDG method for the Incompressible Navier-Stokes Equations on general polyhedral meshes. *ArXiv e-prints*, June 2015.

W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

S. Rhebergen and B. Cockburn. A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains. *Journal of Computational Physics*, 231(11):4185 – 4204, 2012.

S. Rhebergen and B. Cockburn. Space-time hybridizable discontinuous Galerkin method for the advection-diffusion equation on moving and deforming meshes. In C. A. de Moura and C. S. Kubrusly, editors, *The Courant-Friedrichs-Lewy (CFL) Condition*, pages 45–63. Birkhuser Boston, 2013.

S. Rhebergen, B. Cockburn, and J. J. van der Vegt. A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 233:339 – 358, 2013.

H. Riedmann. Efficient numerical treatment of the compressible Navier-Stokes equations with nodal discontinuous Galerkin methods on graphics processors. Technical Report 2009-32, Scientific Computing Group, Brown University, Providence, RI, USA, 2009.

X. Roca, N. C. Nguyen, and J. Peraire. Scalable parallelization of the hybridized discontinuous Galerkin method for compressible flow. In *21st AIAA Computational Fluid Dynamics Conference*, 2013.

J. Schütz, M. Woopen, and G. May. A hybridized dg/mixed scheme for nonlinear advection-diffusion systems, including the compressible Navier-Stokes equations. *AIAA Paper*, 729:2012, 2012.

J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. Cfd vision 2030 study: a path to revolutionary computational aerosciences. Technical Report NASA/CR-2014-218178, NF1676L-18332, NASA, Langley Research Center; Hampton, VA, United States, 2014.

S.-C. Soon, B. Cockburn, and H. Stolarski. A hybridizable discontinuous Galerkin method for linear elasticity. *International Journal for Numerical Methods in Engineering*, 80(8):1058–1092, 2009.

B. van Leer. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics*, 14(4):361 – 370, 1974.

B. van Leer. Towards the ultimate conservative difference scheme iii. upstream-centered finite-difference schemes for ideal compressible flow. *Journal of Computational Physics*, 23(3):263 – 275, 1977a.

B. van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. *Journal of Computational Physics*, 23(3):276 – 299, 1977b.

B. van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101 – 136, 1979.

P. E. Vincent and A. Jameson. Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists. *Mathematical Modelling of Natural Phenomena*, 6:97–140, 1 2011.

J. von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21:232–237, 1950.

L. Wang and W. K. Anderson. Shape sensitivity analysis for the compressible Navier-Stokes equations via discontinuous Galerkin methods. *Computers & Fluids*, 69 (Complete):93–107, 2012.

L. Wang and D. J. Mavriplis. Adjoint-based h-p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *J. Comput. Phys.*, 228(20): 7643–7661, Nov. 2009.

Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

F. M. White. *Viscous Fluid Flow*. McGraw-Hill Education, 3 edition, 2005.

M. Woopen and G. May. An anisotropic adjoint-based hp-adaptive HDG method for compressible turbulent flow. 2015.

M. Woopen, T. Ludescher, and G. May. A hybridized discontinuous Galerkin method for turbulent compressible flow. 2014a.

M. Woopen, G. May, and J. Schtz. Adjoint-based error estimation and mesh adaptation for hybridized discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 76(11):811–834, 2014b.

X. Yang, W. Wang, and Y. Duan. The approximation of a Crank-Nicolson scheme for the stochastic Navier-Stokes equations. *Journal of Computational and Applied Mathematics*, 225(1):31 – 43, 2009.

J. Zhu and J. Qiu. Hermite weno schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method, III: Unstructured meshes. *Journal of Scientific Computing*, 39(2):293–321, 2009.

J. Zhu and J. Qiu. Weno schemes and their application as limiters for RKDG methods based on trigonometric approximation spaces. *J. Sci. Comput.*, 55(3):606–644, 2013.

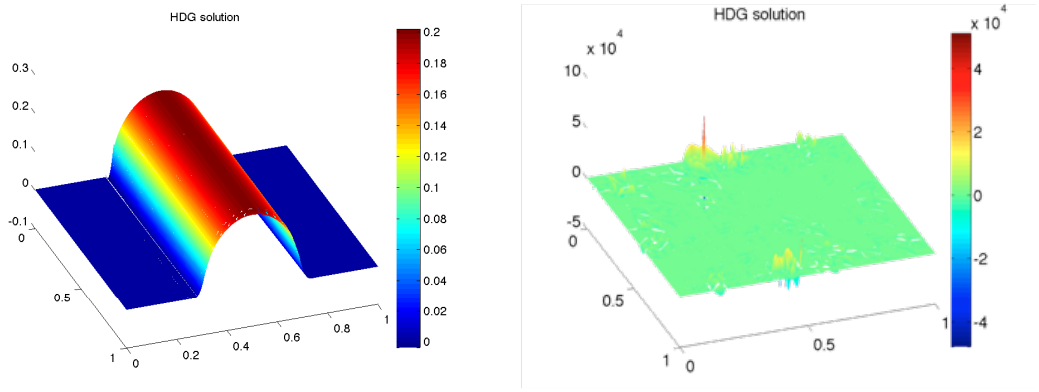
Appendix A

Importance of stabilization parameter for HDG method

The stabilization parameter, τ, is reported to be important for the stability and accuracy of the HDG method, see for instance Nguyen et al. [2009a]. Here an example for convection-diffusion is presented to demonstrate the importance of stabilization parameter. For smooth diffusion dominated problems, no significant difference is observed, but an example of convection dominated problem, with very high Peclet number, shows the effect of correct stabilization parameter on the stability of the solution.

The example considered here is the solution of steady highly convection dominated linear convection-diffusion problem in $\Omega = (0,1) \times (0,1)$, with Dirichlet boundary condition $g_D \mid_{x=0} = \sqrt{0.04 - (y-0.5)^2}$ and Neumann boundary condition $g_N \mid_{x=1} = 0$ and $g_D = 0$ elsewhere on the boundary. So the solution enters the domain from left and goes out from right. Two different stability parameters are used, $\tau_1 = \frac{k}{l} + \mid \boldsymbol{c} \cdot \boldsymbol{n} \mid$ and $\tau_2 = \mid \frac{1}{h^2} \mid_{singlefaced}$, with h being the mesh size, the latter is used in some examples and seems to work well for diffusive regions. Peclet number is $Pe = 1000$ (where normal finite element methods blow up), and cubic approximation is used on a triangular mesh of the type of Figure 2.2 with $n = 32$.

Comparison of results presented in Figures A.1(a) and A.1(b) show that, for τ_1 the method is stable and able to get the correct result, while for single-faced stabilization parameter of τ_2, the method is unstable and blows up, hence, the importance of the



(a) u_h (b) u_h

Figure A.1: Example 3: Numerical results and importance of the stability parameter

stabilization parameter is demonstrated.

Appendix B

Boundary conditions for compressible Navier-Stokes equations

This appendix describes different boundary conditions used for compressible Navier-Stokes equations in this work. Implementing different boundary conditions for HDG discretization can be tricky, and here some decisions are made which are detailed in this section. In general, the boundary conditions are computed using the interior solution $\boldsymbol{U}_h$, gradient of solution $\boldsymbol{Q}_h$, and boundary condition information. It should be noted that, in following description, the $()_{in}$ and $()_{out}$ notations, denote the trace values taken from interior and exterior of domain, respectively.

First, in HDG for compressible Navier-Stokes equations, the boundary conditions are imposed weakly, on the trace of the solution on the boundary, $\hat{\boldsymbol{U}}_{\partial\Omega}$, hence, $\hat{\boldsymbol{U}} = [\hat{\boldsymbol{U}}_{\Gamma\setminus\partial\Omega}, \boldsymbol{U}^b_{\partial\Omega}]^T$, where $\boldsymbol{U}^b_{\partial\Omega}$ is *boundary state vector*, and it is simply referred as $\boldsymbol{U}_b$ afterwards. The Figure B.1, shows this distinction between traces on the interior and boundaries.

For the purpose of this thesis, different boundary conditions are implemented for HDG discretization; inflow, outflow, no-slip wall and symmetry plane boundary conditions. The calculation of boundary state vector for various boundary conditions are presented below.

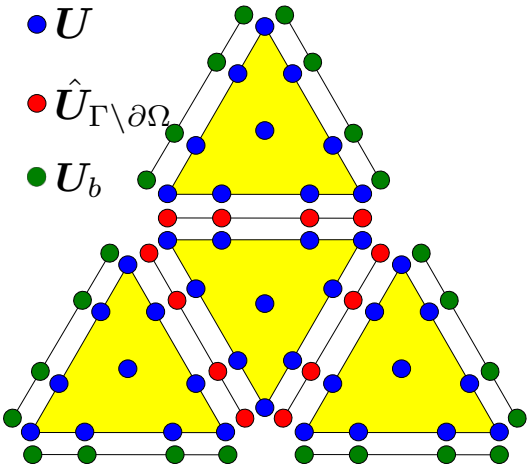


Figure B.1: Boundary condition for HDG

No-slip wall

At no-slip wall, the desired condition is that velocity of fluid is equal to velocity of the wall. For energy equation, either adiabatic or isothermal wall can be considered. In adiabatic case, gradient of temperature is zero, $\boldsymbol{\nabla} T = 0$, and for isothermal case, the wall temperature is considered for fluid at the boundary, $T = T_w$. For aerodynamic simulations, usually the first one is employed, while the second one is used for aero-thermodynamic simulations of hypersonic flows. For adiabatic wall, pressure is taken from interior and total energy is computed from it. The important issue is where to choose interior? Here we consider the value of solution $\boldsymbol{U}_h$ at the same coordinate, as interior for $\boldsymbol{U}_b$. In Figure B.1, this can be seen as choosing the blue point as interior for the boundary condition at the green point, which are at the same coordinate. So the boundary state vector is

$$\boldsymbol{U}_b = \frac{p_{in}}{\gamma - 1} \begin{bmatrix} \frac{1}{c_v T_{in}} \\ \frac{\boldsymbol{v}_w}{c_v T_{in}} \\ 1 + \frac{|\boldsymbol{v}_w|^2}{2 c_v T_{in}} \end{bmatrix} \tag{B.1}$$

if the wall is stationary, then $\boldsymbol{v}_w = 0$ and the boundary state vector is reduced to

$$\boldsymbol{U}_b = \begin{bmatrix} \frac{p_{in}}{c_v T_{in}(\gamma-1)} \\ \mathbf{0} \\ \frac{p_{in}}{\gamma-1} \end{bmatrix} \tag{B.2}$$

Symmetry

The Symmetry plane boundary condition imposes that, the solution to be symmetric about the boundary. The normal derivative of solution, at the boundary is continuous, and velocity is imposed to be tangential to the boundary. Hence, the boundary state vector is set to the values from interior and only difference is that, the normal velocity is subtracted.

$$\boldsymbol{U}_b = \begin{bmatrix} \rho_{in} \\ \rho_{in}\boldsymbol{v}_b \\ \rho_{in}E_{in} \end{bmatrix} \tag{B.3}$$

where $\boldsymbol{v}_b = \boldsymbol{v}_{in} - (\boldsymbol{v}_{in} \cdot \boldsymbol{n})\boldsymbol{n}$. So the normal components of ρ, ρE, and normal component of velocity are all zero.

Inflow/Outflow

Characteristic Euler boundary condition is used here to implement at the inflow and outflow boundaries. For most of our applications, like external flow problems, putting boundary of domain far enough from flow features leads to a uniform flow at far-fields, hence, the utilization of Euler boundary condition is justified.

In this approach, Reimann invariants are used, and at each point of boundary, the number of prescribed boundary conditions corresponds to the number of negative eigenvalues. Without entering the details of the method, just a brief description is presented here for 2D flows.

Subsonic flow

For subsonic inflow, one characteristics comes from domain, while the rests come from far-field, hence, one parameter is taken from flow field, and all others are imposed based on the free-stream conditions, that is $\boldsymbol{U}_{out} = \boldsymbol{U}_\infty$. The boundary state vector is

$$\boldsymbol{U}_b = \begin{bmatrix} \rho_b \\ \rho_b\boldsymbol{v}_b \\ \rho_b E_b \end{bmatrix} \tag{B.4}$$

where

$$\begin{aligned}\rho_b &= \frac{(\rho_{out})^\gamma(\gamma-1)^2}{16\gamma p_{out}}(w_{in}^1 - w_{out}^5)^2 \\ \boldsymbol{v}_b &= \boldsymbol{v}_{out}^t + \frac{1}{2}(w_{in}^1 + w_{out}^5)\boldsymbol{n} \\ E_b &= \frac{\gamma-1}{16\gamma}(w_{in}^1 - w_{out}^5)^2 + \frac{1}{2}|\boldsymbol{v}_b|^2\end{aligned} \tag{B.5}$$

where the Reimann invariants are

$$\begin{aligned}w_{in}^1 &= \boldsymbol{n}\cdot\boldsymbol{v}_{in} + \frac{2c_{in}}{\gamma-1} \\ w_{out}^5 &= \boldsymbol{n}\cdot\boldsymbol{v}_{out} - \frac{2c_{out}}{\gamma-1}\end{aligned} \tag{B.6}$$

where c is speed of sound, which is

$$c = \sqrt{\frac{\gamma p}{\rho}} \tag{B.7}$$

For subsonic outflow, one characteristics comes from far-field, while the rests come from domain, hence, one parameter is taken from boundary, usually pressure, and all other variables are from flow field. The boundary state vector is $\boldsymbol{U}_b = [\rho_b, \rho_b\boldsymbol{v}_b, \rho_b E_b]^T$, where

$$\begin{aligned}\rho_b &= \rho_{in}(\frac{p_{out}}{p_{in}})^{(1/\gamma)} \\ \boldsymbol{v}_b &= \boldsymbol{v}_{in} + \frac{2}{\gamma-1}(c_{in} - \sqrt{\frac{\gamma p_{out}}{\rho_b}}) \\ E_b &= \frac{p_{out}}{\rho_b(\gamma-1)} + \frac{1}{2}|\boldsymbol{v}_b|^2\end{aligned} \tag{B.8}$$

Supersonic flow

For supersonic inflow, all characteristics are from boundary, hence, it corresponds to Dirichlet boundary condition.

$$\boldsymbol{U}_b = \begin{bmatrix}\rho_{out} \\ \rho_{out}\boldsymbol{v}_{out} \\ \rho_{out}E_{out}\end{bmatrix} \tag{B.9}$$

For supersonic outflow, all characteristics come from flow field, hence, it corresponds to Neumann boundary condition.

$$\boldsymbol{U}_b = \begin{bmatrix} \rho_{in} \\ \rho_{in}\boldsymbol{v}_{in} \\ \rho_{in}E_{in} \end{bmatrix} \tag{B.10}$$

Appendix C

Implementation of HDG method

This appendix presents main indications on the implementation of the HDG method for different equations; first convection-diffusion equation, then compressible Navier-Stokes equations, and finally linearized compressible Navier-Stokes equations.

C.1 Implementation of HDG for convection-diffusion equation

We start HDG implementation of convection-diffusion equation, presented in section 2.1, by recalling the local and global problems, 2.15 and 2.16,

$$\begin{aligned} (k^{-1}\boldsymbol{q}_h, \boldsymbol{z})_{\Omega_e} - (u_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e} + \langle \hat{u}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e} = 0, \\ (\boldsymbol{\nabla}\cdot\boldsymbol{q}_h, r)_{\Omega_e} - (\boldsymbol{c}u_h, \boldsymbol{\nabla}r)_{\Omega_e} + \langle \tau u_h, r\rangle_{\partial\Omega_e} + \langle (\boldsymbol{c}\cdot\boldsymbol{n} - \tau)\hat{u}_h, r\rangle_{\partial\Omega_e} = (f, r)_{\Omega_e}, \end{aligned} \tag{C.1}$$

for $e = 1, \cdots, n_{el}$,

$$\sum_{e=1}^{n_{el}} \langle (\boldsymbol{c}\hat{u}_h + \boldsymbol{q}_h)\cdot\boldsymbol{n} + \tau(u_h - \hat{u}_h), \mu\rangle_{\partial\Omega_e} = g_N, \quad \forall \mu \in \mathscr{M}_h^p, \tag{C.2}$$

On one hand, equation C.1 is a local system for each element Ω_e, which is independent of other elements. Thus, an element-by-element procedure can be employed to express both $\boldsymbol{q}_h$ and u_h, in terms of $\hat{u}_h$. In addition, there is equation C.2, which is global, and coupled for all elements and traces. Replacing $\boldsymbol{q}_h$ and u_h, as solution

of C.1 in terms of $\hat{u}_h$, in the global problem, C.2, yields a global system on the whole mesh skeleton, Γ.

Elemental unknowns, $\boldsymbol{q}_h$ and u_h, are presented with 2D nodal shape functions N_j, as

$$\begin{aligned} \boldsymbol{q}_h &= \sum_{i=1}^{n_{se}} N_i \boldsymbol{I} \{\mathbb{Q}_e\}_i, \\ u_h &= \sum_{i=1}^{n_{se}} N_i \{\mathbb{U}_e\}_i, \end{aligned} \tag{C.3}$$

where n_{se} is number of shape functions in the element Ω_e, and $\boldsymbol{I}$ is identity matrix of $sd \times sd$, where sd is number of spatial dimensions. The vector of nodal values in each element for variable $\boldsymbol{q}_h$ and u_h are shown by $\mathbb{Q}_e$ and $\mathbb{U}_e$, respectively, which include contributions from each shape function of the elements, $\{\mathbb{Q}_e\}_i$ and $\{\mathbb{U}_e\}_i$. For the trace unknown, $\hat{u}_h$, in the face f of the element e is denoted as

$$\hat{u}_h = \sum_{i=1}^{n_{sf}} \bar{N}_i \{\hat{\mathbb{U}}_{e,f}\}_i, \tag{C.4}$$

where, n_{sf} is the number of shape functions at the face f of element Ω_e, and $\bar{N}_i$ is the i^{th} 1D basis function in that face. For the whole element, vector of nodal value of $\hat{u}_h$, is sum of the all faces of that element. It's also important to know that since each face is shared by two elements, $\hat{\mathbb{U}}_{e,f}$ has contributions from both elements and is single valued.

Summing up the local problem, C.1, over all elements, with the global problem, C.2, lead to discretization in the following form.

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{q}_h, \boldsymbol{z}) - \boldsymbol{B}(u_h, \boldsymbol{z}) + \boldsymbol{C}(\hat{u}_h, \boldsymbol{z}) &= \boldsymbol{0}, \\ \boldsymbol{B}(r, \boldsymbol{q}_h) + \boldsymbol{D}(u_h, r) + \boldsymbol{E}(\hat{u}_h, r) &= \boldsymbol{F}(r), \\ \boldsymbol{C}(\mu, \boldsymbol{q}_h) + \boldsymbol{G}(\mu, u_h) + \boldsymbol{H}(\mu, \hat{u}_h) &= \boldsymbol{0}, \end{aligned} \tag{C.5}$$

where the bilinear forms and linear functionals are given by

$$\begin{aligned}
\boldsymbol{A}(\boldsymbol{q},\boldsymbol{z}) &= (k^{-1}\boldsymbol{q},\boldsymbol{z})_{\Omega_e},\\
\boldsymbol{B}(u,\boldsymbol{z}) &= (u,\boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e},\\
\boldsymbol{C}(\hat{u},\boldsymbol{z}) &= \langle\hat{u},\boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e},\\
\boldsymbol{D}(u,r) &= -(\boldsymbol{c}u,\boldsymbol{\nabla}r)_{\Omega_e} + \langle r,\tau u\rangle_{\partial\Omega_e},\\
\boldsymbol{E}(\hat{u},r) &= \langle r,(\boldsymbol{c}\cdot\boldsymbol{n}-\tau)\hat{u}\rangle_{\partial\Omega_e},\\
\boldsymbol{G}(\mu,u) &= \langle\mu,\tau u\rangle_{\partial\Omega_e},\\
\boldsymbol{H}(\mu,\hat{u}) &= \langle\mu,(\boldsymbol{c}\cdot\boldsymbol{n}-\tau)\hat{u}\rangle_{\partial\Omega_e},\\
\boldsymbol{F}(r) &= (f,r)_{\Omega_e}
\end{aligned} \tag{C.6}$$

Now, using the definitions of unknowns, C.3 and C.4, and replacing in C.5, gives rise to the matrix form of local and global equations as

$$\begin{bmatrix}\mathcal{A} & -\mathcal{B}^T & \mathcal{C}^T\\ \mathcal{B} & \mathcal{D} & \mathcal{E}\\ \mathcal{C} & \mathcal{G} & \mathcal{H}\end{bmatrix}\begin{bmatrix}\mathbb{Q}\\ \mathbb{U}\\ \hat{\mathbb{U}}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathcal{F}\\ \mathbf{0}\end{bmatrix} \tag{C.7}$$

where matrices in C.7 correspond to the bilinear forms in C.6, in the order they appear in the equations.

As stated before, the local problem, can be used to eliminate both $\boldsymbol{q}_h$ and u_h in an element-by-element fashion and the system C.7 can be expressed as

$$\begin{bmatrix}\mathbb{Q}\\ \mathbb{U}\end{bmatrix} = \begin{bmatrix}\mathcal{A} & -\mathcal{B}^T\\ \mathcal{B} & \mathcal{D}\end{bmatrix}^{-1}\left(\begin{bmatrix}\mathbf{0}\\ \mathcal{F}\end{bmatrix} - \begin{bmatrix}\mathcal{C}^T\\ \mathcal{E}\end{bmatrix}\hat{\mathbb{U}}\right), \tag{C.8}$$

and

$$\mathcal{C}\mathbb{Q} + \mathcal{G}\mathbb{U} + \mathcal{H}\hat{\mathbb{U}} = \mathbf{0}, \tag{C.9}$$

Once computing, the system C.8 is used for each element, and it's replaced in C.9, hence, for triangular elements, each element contributes to 3 faces in the final global system which only includes the DOFs of trace of the solution and is solved for DOFs of trace of the solution, $\hat{\mathbb{U}}$.

C.2 Implementation of HDG for compressible Navier-Stokes equations

Here, some insight for implementation of HDG for compressible Navier-Stokes equations are presented. The overall idea is extension of section C.1, but the difference is compressible Navier-Stokes equations are non-linear, thus, a Newton-Raphson method is employed and the system is solved for the increments of the solution, until convergence, as explained in section 2.2.

The weak form of the HDG local and global problems for unsteady compressible Navier-Stokes equations are recalled here.

$$\left.\begin{aligned} (\boldsymbol{Q}_h, \boldsymbol{z})_{\Omega_e} + (\boldsymbol{U}_h, \boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e} - \langle \hat{\boldsymbol{U}}_h, \boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e} &= 0, \\ (\frac{\partial \boldsymbol{U}_h}{\partial t}, \boldsymbol{r})_{\Omega_e} - (\boldsymbol{F}_c(\boldsymbol{U}_h), \boldsymbol{\nabla}\boldsymbol{r})_{\Omega_e} + (\boldsymbol{F}_d(\boldsymbol{U}_h, \boldsymbol{Q}_h), \boldsymbol{\nabla}\boldsymbol{r})_{\Omega_e} + \langle \boldsymbol{F}_c(\hat{\boldsymbol{U}}_h)\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial\Omega_e} & \\ -\langle \boldsymbol{F}_d(\hat{\boldsymbol{U}}_h, \boldsymbol{Q}_h)\cdot\boldsymbol{n}, \boldsymbol{r}\rangle_{\partial\Omega_e} + \langle S\boldsymbol{U}_h, \boldsymbol{r}\rangle_{\partial\Omega_e} - \langle S\hat{\boldsymbol{U}}_h, \boldsymbol{r}\rangle_{\partial\Omega_e} &= (\boldsymbol{f}, \boldsymbol{r})_{\Omega_e}, \end{aligned}\right\} \tag{C.10}$$

for $e = 1, \cdots, n_{el}$, and

$$\sum_{e=1}^{n_{el}} \langle (\boldsymbol{F}_c(\hat{\boldsymbol{U}}_h) - \boldsymbol{F}_d(\hat{\boldsymbol{U}}_h, \boldsymbol{Q}_h))\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{U}_h - \hat{\boldsymbol{U}}_h), \boldsymbol{\mu}\rangle_{\partial\Omega_e\setminus\partial\Omega} = 0, \tag{C.11}$$

System C.10 represents a local problem for each element, Ω_e and it allows elemental unknown $\boldsymbol{Q}_h$ and $\boldsymbol{U}_h$ as a function of the trace unknown, $\hat{\boldsymbol{U}}_h$. Then, replaced in C.11, and as a result, a global system is set up in terms of just the trace unknown.

Following the procedure in C.1, unknowns are represented with nodal shape functions N_j in each element, Ω_e, or in each face, Γ_f, as

$$\begin{aligned} \boldsymbol{Q}_h &= \sum_{i=1}^{n_{se}} N_i \boldsymbol{I}\{\mathbb{Q}_e\}_i, \\ \boldsymbol{U}_h &= \sum_{i=1}^{n_{se}} N_i \boldsymbol{I}\{\mathbb{U}_e\}_i, \\ \hat{\boldsymbol{U}}_h &= \sum_{i=1}^{n_{sf}} \bar{N}_i \boldsymbol{I}\{\hat{\mathbb{U}}_{e,f}\}_i, \end{aligned} \tag{C.12}$$

where the identity matrix, $\boldsymbol{I}$ is has dimension of $(sd+2)\times(sd+2)$ for $\boldsymbol{U}_h$ and $\hat{\boldsymbol{U}}_h$, and $(sd+2)sd \times sd(sd+2)$ for $\boldsymbol{Q}_h$, while $\bar{N}_i$ is 1D basis function in face f of the

element Ω_e. For the whole element, vector of nodal value of $\hat{\boldsymbol{U}}_h$, is sum of the all faces of that element.

Summing up the local problem, C.10, over all elements, with the global problem, C.11, and using backward Euler time marching, leads to discretization in the following form.

$$\begin{aligned}
\boldsymbol{A}(\boldsymbol{Q}_h,\boldsymbol{z})+\boldsymbol{B}(\boldsymbol{U}_h,\boldsymbol{z})-\boldsymbol{C}(\hat{\boldsymbol{U}}_h,\boldsymbol{z})&=\boldsymbol{0},\\
\boldsymbol{T1}(\boldsymbol{U}_h,\boldsymbol{r})-\boldsymbol{D}(\boldsymbol{U}_h,\boldsymbol{r})+\boldsymbol{E}(\boldsymbol{U}_h,\boldsymbol{Q}_h,\boldsymbol{r})+\boldsymbol{G1}(\hat{\boldsymbol{U}}_h,\boldsymbol{r})-\boldsymbol{H1}(\hat{\boldsymbol{U}}_h,\boldsymbol{Q}_h,\boldsymbol{r})&\\
+\boldsymbol{L1}(\boldsymbol{U}_h,\boldsymbol{r})-\boldsymbol{M1}(\hat{\boldsymbol{U}}_h,\boldsymbol{r})&=\boldsymbol{O}(\boldsymbol{r})+\boldsymbol{T2}(\boldsymbol{r}),\\
\boldsymbol{G2}(\hat{\boldsymbol{U}}_h,\boldsymbol{\mu})-\boldsymbol{H2}(\hat{\boldsymbol{U}}_h,\boldsymbol{Q}_h,\boldsymbol{\mu})+\boldsymbol{L2}(\boldsymbol{U}_h,\boldsymbol{\mu})-\boldsymbol{M2}(\hat{\boldsymbol{U}}_h,\boldsymbol{\mu})&=\boldsymbol{0},
\end{aligned} \tag{C.13}$$

where the bilinear forms and linear functionals are given by

$$\begin{aligned}
\boldsymbol{A}(\boldsymbol{Q},\boldsymbol{z})&=(\boldsymbol{Q}^n,\boldsymbol{z})_{\Omega_e},\\
\boldsymbol{B}(\boldsymbol{U},\boldsymbol{z})&=(\boldsymbol{U}^n,\boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e},\\
\boldsymbol{C}(\hat{\boldsymbol{U}},\boldsymbol{z})&=\langle\hat{\boldsymbol{U}}^n,\boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e},\\
\boldsymbol{D}(\boldsymbol{U},\boldsymbol{r})&=(\boldsymbol{F}_c(\boldsymbol{U}^n),\boldsymbol{\nabla r})_{\Omega_e},\\
\boldsymbol{E}(\boldsymbol{U},\boldsymbol{Q},\boldsymbol{r})&=(\boldsymbol{F}_d(\boldsymbol{U}^n,\boldsymbol{Q}^n),\boldsymbol{\nabla r})_{\Omega_e},\\
\boldsymbol{G1}(\hat{\boldsymbol{U}},\boldsymbol{r})&=\langle\boldsymbol{F}_c(\hat{\boldsymbol{U}}^n)\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e},\\
\boldsymbol{H1}(\hat{\boldsymbol{U}},\boldsymbol{Q},\boldsymbol{r})&=\langle\boldsymbol{F}_d(\hat{\boldsymbol{U}}^n,\boldsymbol{Q}^n)\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e},\\
\boldsymbol{L1}(\boldsymbol{U},\boldsymbol{r})&=\langle S\boldsymbol{U}^n,\boldsymbol{r}\rangle_{\partial\Omega_e},\\
\boldsymbol{M1}(\hat{\boldsymbol{U}},\boldsymbol{r})&=\langle S\hat{\boldsymbol{U}}^n,\boldsymbol{r}\rangle_{\partial\Omega_e},\\
\boldsymbol{G2}(\hat{\boldsymbol{U}},\boldsymbol{\mu})&=\langle\boldsymbol{F}_c(\hat{\boldsymbol{U}}^n)\cdot\boldsymbol{n},\boldsymbol{\mu}\rangle_{\partial\Omega_e},\\
\boldsymbol{H2}(\hat{\boldsymbol{U}},\boldsymbol{Q},\boldsymbol{\mu})&=\langle\boldsymbol{F}_d(\hat{\boldsymbol{U}}^n,\boldsymbol{Q}^n)\cdot\boldsymbol{n},\boldsymbol{\mu}\rangle_{\partial\Omega_e},\\
\boldsymbol{L2}(\boldsymbol{U},\boldsymbol{\mu})&=\langle S\boldsymbol{U}^n,\boldsymbol{\mu}\rangle_{\partial\Omega_e},\\
\boldsymbol{M2}(\hat{\boldsymbol{U}},\boldsymbol{\mu})&=\langle S\hat{\boldsymbol{U}}^n,\boldsymbol{\mu}\rangle_{\partial\Omega_e},\\
\boldsymbol{O}(\boldsymbol{r})&=(\boldsymbol{f},\boldsymbol{r})_{\Omega_e},\\
\boldsymbol{T1}(\boldsymbol{U},\boldsymbol{r})&=\frac{1}{dt}(\boldsymbol{U}^n,\boldsymbol{r})_{\Omega_e},\\
\boldsymbol{T2}(\boldsymbol{r})&=\frac{1}{dt}(\boldsymbol{U}^{n-1},\boldsymbol{r})_{\Omega_e},
\end{aligned} \tag{C.14}$$

Note that, the last two terms are from backward Euler time discretization and utilization of other time marching would change these two, but the other remain untouched.

Now, using the definitions of unknowns, C.12, and replacing in C.13, give rise to the residual form of local and global equations as

$$\begin{aligned}
\boldsymbol{R}_Q(\mathbb{Q},\mathbb{U},\hat{\mathbb{U}}) &= \mathcal{A}+\mathcal{B}-\mathcal{C},\\
\boldsymbol{R}_U(\mathbb{Q},\mathbb{U},\hat{\mathbb{U}}) &= \mathcal{T}1-\mathcal{D}+\mathcal{E}+\mathcal{G}1-\mathcal{H}1+\mathcal{L}1-\mathcal{M}1-\mathcal{T}2,\\
\boldsymbol{R}_{\hat{U}}(\mathbb{Q},\mathbb{U},\hat{\mathbb{U}}) &= \mathcal{G}2-\mathcal{H}2+\mathcal{L}2-\mathcal{M}2,
\end{aligned} \tag{C.15}$$

where matrices in C.15 correspond to the bilinear forms in C.14, in the order they appear in the equations. In order to compute the Jacobian for non-linear solver, the derivatives of of residuals with respect to the nodal values are needed, and using definitions in C.15 lead to

$$\begin{aligned}
&\frac{\partial \boldsymbol{R}_Q}{\partial \mathbb{Q}} = \frac{\partial \mathcal{A}}{\partial \mathbb{Q}},\\
&\frac{\partial \boldsymbol{R}_Q}{\partial \mathbb{U}} = \frac{\partial \mathcal{B}}{\partial \mathbb{U}},\\
&\frac{\partial \boldsymbol{R}_Q}{\partial \hat{\mathbb{U}}} = -\frac{\partial \mathcal{C}}{\partial \hat{\mathbb{U}}},\\
&\frac{\partial \boldsymbol{R}_U}{\partial \mathbb{Q}} = \frac{\partial \mathcal{E}}{\partial \mathbb{Q}} - \frac{\partial \mathcal{H}1}{\partial \mathbb{Q}},\\
&\frac{\partial \boldsymbol{R}_U}{\partial \mathbb{U}} = \frac{\partial \mathcal{T}1}{\partial \mathbb{U}} - \frac{\partial \mathcal{D}}{\partial \mathbb{U}} + \frac{\partial \mathcal{E}}{\partial \mathbb{U}} + \frac{\partial \mathcal{L}1}{\partial \mathbb{U}},\\
&\frac{\partial \boldsymbol{R}_U}{\partial \hat{\mathbb{U}}} = \frac{\partial \mathcal{G}1}{\partial \hat{\mathbb{U}}} - \frac{\partial \mathcal{H}1}{\partial \hat{\mathbb{U}}} - \frac{\partial \mathcal{M}1}{\partial \hat{\mathbb{U}}},\\
&\frac{\partial \boldsymbol{R}_{\hat{U}}}{\partial \mathbb{Q}} = -\frac{\partial \mathcal{H}2}{\partial \mathbb{Q}},\\
&\frac{\partial \boldsymbol{R}_{\hat{U}}}{\partial \mathbb{U}} = \frac{\partial \mathcal{L}2}{\partial \mathbb{U}},\\
&\frac{\partial \boldsymbol{R}_{\hat{U}}}{\partial \hat{\mathbb{U}}} = \frac{\partial \mathcal{G}2}{\partial \hat{\mathbb{U}}} - \frac{\partial \mathcal{H}2}{\partial \hat{\mathbb{U}}} - \frac{\partial \mathcal{M}2}{\partial \hat{\mathbb{U}}},
\end{aligned} \tag{C.16}$$

Using C.16 and C.15, one can set up the Jacobian and residual for non-linear solver as presented in 2.40. Then, the HDG procedure to eliminate the local unknowns in terms of the trace unknown, as presented in C.1, can be repeated and as a result, system can be solved for the increments in the trace of solution, and update it until convergence.

C.3 Implementation of HDG for linearized compressible Navier-Stokes equations

This section, provides the basic details, for implementation of linearized HDG for compressible Navier-Stokes equations, which is presented in section 2.3 of this thesis.

The weak form of HDG local and global problem for linearized unsteady compressible Navier-Stokes equations, 2.52 and 2.53 are recalled here.

$$\left.\begin{aligned}(\frac{\partial \boldsymbol{U}_h}{\partial t},\boldsymbol{r})_{\Omega_e}-(\boldsymbol{A}_{fc}\boldsymbol{U}_h,\boldsymbol{\nabla r})_{\Omega_e}-(\boldsymbol{A}_{fd_1}\boldsymbol{L}_h,\boldsymbol{\nabla r})_{\Omega_e}-(\boldsymbol{A}_{fd_2}\boldsymbol{w}_h,\boldsymbol{\nabla r})_{\Omega_e}\\ +\langle \boldsymbol{A}_{fc}\hat{\boldsymbol{U}}_h\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e}+\langle \boldsymbol{A}_{fd_1}\boldsymbol{L}_h\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e}+\langle \boldsymbol{A}_{fd_2}\boldsymbol{w}_h\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e}\\ +\langle \boldsymbol{S}\boldsymbol{U}_h,\boldsymbol{r}\rangle_{\partial\Omega_e}-\langle \boldsymbol{S}\hat{\boldsymbol{U}}_h,\boldsymbol{r}\rangle_{\partial\Omega_e}=(\boldsymbol{f},\boldsymbol{r})_{\Omega_e},\\ (\bar{\rho}\boldsymbol{L}_h,\boldsymbol{z})_{\Omega_e}+(\rho_h\boldsymbol{v}_h,\boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e}-\langle\widehat{\rho_h\boldsymbol{v}_h},\boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e}+(\bar{\boldsymbol{v}}\boldsymbol{\nabla}\rho_h,\boldsymbol{z})_{\Omega_e}=\boldsymbol{0},\\ (\bar{\rho}\boldsymbol{w}_h,\boldsymbol{g})_{\Omega_e}+(\rho_h E_h,\boldsymbol{\nabla}\cdot\boldsymbol{g})_{\Omega_e}-\langle\widehat{\rho_h E_h},\boldsymbol{g}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e}+(\bar{E}\boldsymbol{\nabla}\rho_h,\boldsymbol{g})_{\Omega_e}=\boldsymbol{0},\end{aligned}\right\} \tag{C.17}$$

for $e=1,\cdots,n_{el}$, and

$$\sum_{e=1}^{n_{el}}\langle\Big(\boldsymbol{A}_{fc}(\bar{\boldsymbol{v}})\boldsymbol{U}_h+\widehat{\boldsymbol{A}_{fd_1}\boldsymbol{L}_h}+\boldsymbol{A}_{fd_2}(\bar{\boldsymbol{v}})\boldsymbol{w}_h\Big)\cdot\boldsymbol{n},\boldsymbol{\mu}\rangle_{\partial\Omega_e\setminus\partial\Omega}=0, \tag{C.18}$$

System C.17 represents a local problem for each element, Ω_e and it allows elemental unknown $\boldsymbol{U}_h$, $\boldsymbol{L}_h$ and $\boldsymbol{w}_h$ as a function of the trace unknown, $\hat{\boldsymbol{U}}_h$. Then, replaced in C.18, and as a result, a global system is set up in terms of just the trace unknown.

Following the HDG procedure in C.1, and C.2, unknowns are represented with nodal shape functions N_j in each element, Ω_e, or in each face, Γ_f, as

$$\begin{aligned}\boldsymbol{U}_h&=\sum_{i=1}^{n_{se}}N_i\boldsymbol{I}\{\mathbb{U}_e\}_i,\\ \boldsymbol{L}_h&=\sum_{i=1}^{n_{se}}N_i\boldsymbol{I}\{\mathbb{L}_e\}_i,\\ \boldsymbol{w}_h&=\sum_{i=1}^{n_{se}}N_i\boldsymbol{I}\{\mathbb{W}_e\}_i,\\ \hat{\boldsymbol{U}}_h&=\sum_{i=1}^{n_{sf}}\bar{N}_i\boldsymbol{I}\{\hat{\mathbb{U}}_{e,f}\}_i,\end{aligned} \tag{C.19}$$

where the identity matrix, $\boldsymbol{I}$ is has dimension of $(sd+2)\times(sd+2)$ for $\boldsymbol{U}_h$, $\boldsymbol{L}_h$ and $\hat{\boldsymbol{U}}_h$, and $sd\times sd$ for $\boldsymbol{w}_h$, while $\bar{N}_i$ is 1D basis function in face f of the element Ω_e. For the whole element, vector of nodal value of $\hat{u}_h$, is sum of the all faces of that element.

Summing up the local problem, C.17, over all elements, with the global problem, C.18, and using backward Euler time marching, leads to discretization in the following form.

$$\begin{aligned}
\boldsymbol{T1}(\boldsymbol{U}_h,\boldsymbol{r}) - \boldsymbol{A}(\boldsymbol{U}_h,\boldsymbol{r}) - \boldsymbol{B}(\boldsymbol{L}_h,\boldsymbol{r}) + \boldsymbol{C}(\boldsymbol{w}_h,\boldsymbol{r}) + \boldsymbol{An1}(\hat{\boldsymbol{U}}_h,\boldsymbol{r}) \\
+\boldsymbol{Bn1}(\boldsymbol{L}_h,\boldsymbol{r}) + \boldsymbol{Cn1}(\boldsymbol{w}_h,\boldsymbol{r}) + \boldsymbol{D1}(\boldsymbol{U}_h,\boldsymbol{r}) - \boldsymbol{D2}(\hat{\boldsymbol{U}}_h,\boldsymbol{r}) = \boldsymbol{S}(\boldsymbol{r}) + \boldsymbol{T2}(\boldsymbol{r}), \\
\boldsymbol{G1}(\boldsymbol{L}_h,\boldsymbol{z}) + \boldsymbol{F1}(\boldsymbol{U}_h,\boldsymbol{z}) - \boldsymbol{H1}(\hat{\boldsymbol{U}}_h,\boldsymbol{z}) + \boldsymbol{N1}(\boldsymbol{U}_h,\boldsymbol{z}) = \boldsymbol{0}, \\
\boldsymbol{G2}(\boldsymbol{w}_h,\boldsymbol{g}) + \boldsymbol{F2}(\boldsymbol{U}_h,\boldsymbol{g}) - \boldsymbol{H2}(\hat{\boldsymbol{U}}_h,\boldsymbol{g}) + \boldsymbol{N2}(\boldsymbol{U}_h,\boldsymbol{g}) = \boldsymbol{0}, \\
\boldsymbol{An2}(\hat{\boldsymbol{U}}_h,\boldsymbol{\mu}) + \boldsymbol{Bn2}(\boldsymbol{L}_h,\boldsymbol{\mu}) + \boldsymbol{Cn2}(\boldsymbol{w}_h,\boldsymbol{\mu}) + \boldsymbol{D3}(\boldsymbol{U}_h,\boldsymbol{\mu}) - \boldsymbol{D3}(\hat{\boldsymbol{U}}_h,\boldsymbol{\mu}) = \boldsymbol{0},
\end{aligned} \tag{C.20}$$

where the bilinear forms and linear functionals are given by

$\boldsymbol{T1}(\boldsymbol{U},\boldsymbol{r}) = \frac{1}{dt}(\boldsymbol{U}^n,\boldsymbol{r})_{\Omega_e}$,	$\boldsymbol{A}(\boldsymbol{U},\boldsymbol{r}) = (\boldsymbol{A}_{fc}\boldsymbol{U}^n,\boldsymbol{\nabla r})_{\Omega_e}$,
$\boldsymbol{B}(\boldsymbol{L},\boldsymbol{r}) = (\boldsymbol{A}_{fd_1}\boldsymbol{L}^n,\boldsymbol{\nabla r})_{\Omega_e}$,	$\boldsymbol{C}(\boldsymbol{w},\boldsymbol{r}) = (\boldsymbol{A}_{fd_2}\boldsymbol{w}^n,\boldsymbol{\nabla r})_{\Omega_e}$,
$\boldsymbol{An1}(\hat{\boldsymbol{U}},\boldsymbol{r}) = \langle \boldsymbol{A}_{fc}\hat{\boldsymbol{U}}^n\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e}$,	$\boldsymbol{Bn1}(\boldsymbol{L},\boldsymbol{r}) = \langle \boldsymbol{A}_{fd_1}\boldsymbol{L}^n\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e}$,
$\boldsymbol{Cn1}(\boldsymbol{w},\boldsymbol{r}) = \langle \boldsymbol{A}_{fd_2}\boldsymbol{w}^n\cdot\boldsymbol{n},\boldsymbol{r}\rangle_{\partial\Omega_e}$,	$\boldsymbol{D1}(\boldsymbol{U},\boldsymbol{r}) = \langle \boldsymbol{SU}^n,\boldsymbol{r}\rangle_{\partial\Omega_e}$,
$\boldsymbol{D2}(\hat{\boldsymbol{U}},\boldsymbol{r}) = \langle \boldsymbol{S}\hat{\boldsymbol{U}}^n,\boldsymbol{r}\rangle_{\partial\Omega_e}$,	$\boldsymbol{G1}(\boldsymbol{L},\boldsymbol{z}) = (\bar{\rho}\boldsymbol{L}^n,\boldsymbol{z})_{\Omega_e}$,
$\boldsymbol{F1}(\boldsymbol{U},\boldsymbol{z}) = (\rho\boldsymbol{v}^n,\boldsymbol{\nabla}\cdot\boldsymbol{z})_{\Omega_e}$,	$\boldsymbol{H1}(\hat{\boldsymbol{U}},\boldsymbol{z}) = \langle \widehat{\rho\boldsymbol{v}}^n,\boldsymbol{z}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e}$,
$\boldsymbol{N1}(\boldsymbol{U},\boldsymbol{z}) = (\bar{\boldsymbol{v}}\boldsymbol{\nabla}\rho^n,\boldsymbol{z})_{\Omega_e}$,	$\boldsymbol{G2}(\boldsymbol{w},\boldsymbol{g}) = (\bar{\rho}\boldsymbol{w}^n,\boldsymbol{g})_{\Omega_e}$,
$\boldsymbol{F2}(\boldsymbol{U},\boldsymbol{g}) = (\rho E^n,\boldsymbol{\nabla}\cdot\boldsymbol{g})_{\Omega_e}$,	$\boldsymbol{H2}(\hat{\boldsymbol{U}},\boldsymbol{g}) = \langle \widehat{\rho E}^n,\boldsymbol{g}\cdot\boldsymbol{n}\rangle_{\partial\Omega_e}$,
$\boldsymbol{N2}(\boldsymbol{U},\boldsymbol{g}) = (\bar{E}\boldsymbol{\nabla}\rho^n,\boldsymbol{g})_{\Omega_e}$,	$\boldsymbol{An2}(\hat{\boldsymbol{U}},\boldsymbol{\mu}) = \langle \boldsymbol{A}_{fc}\hat{\boldsymbol{U}}^n\cdot\boldsymbol{n},\boldsymbol{\mu}\rangle_{\partial\Omega_e}$,
$\boldsymbol{Bn2}(\boldsymbol{L},\boldsymbol{\mu}) = \langle \boldsymbol{A}_{fd_1}\boldsymbol{L}^n\cdot\boldsymbol{n},\boldsymbol{\mu}\rangle_{\partial\Omega_e}$,	$\boldsymbol{Cn2}(\boldsymbol{w},\boldsymbol{\mu}) = \langle \boldsymbol{A}_{fd_2}\boldsymbol{w}^n\cdot\boldsymbol{n},\boldsymbol{\mu}\rangle_{\partial\Omega_e}$,
$\boldsymbol{D3}(\boldsymbol{U},\boldsymbol{\mu}) = \langle \boldsymbol{SU}^n,\boldsymbol{\mu}\rangle_{\partial\Omega_e}$,	$\boldsymbol{D4}(\hat{\boldsymbol{U}},\boldsymbol{\mu}) = \langle \boldsymbol{S}\hat{\boldsymbol{U}}^n,\boldsymbol{\mu}\rangle_{\partial\Omega_e}$,
$\boldsymbol{S}(\boldsymbol{r}) = (\boldsymbol{f}^n,\boldsymbol{r})_{\Omega_e}$,	$\boldsymbol{T2}(\boldsymbol{r}) = \frac{1}{dt}(\boldsymbol{U}^{n-1},\boldsymbol{r})_{\Omega_e}$,

(C.21)

Now, using the definitions of unknowns, C.19, and replacing in C.20, give rise to the matrix form of local and global equations as

$$\begin{bmatrix} (\mathcal{T}1-\mathcal{A}+\mathcal{D}1) & (-\mathcal{B}+\mathcal{B}n1) & (-\mathcal{C}+\mathcal{C}n1) & (\mathcal{A}n1-\mathcal{D}2) \\ (\mathcal{F}1+\mathcal{N}1) & \mathcal{G}1 & & -\mathcal{H}1 \\ (\mathcal{F}2+\mathcal{N}2) & & \mathcal{G}2 & -\mathcal{H}2 \\ \mathcal{D}3 & \mathcal{B}n2 & \mathcal{C}n2 & (\mathcal{A}n2-\mathcal{D}4) \end{bmatrix} \begin{bmatrix} \mathbb{U}^n \\ \mathbb{L}^n \\ \mathbb{W}^n \\ \hat{\mathbb{U}}^n \end{bmatrix} = \begin{bmatrix} \mathcal{S}^{n-1}+\mathcal{T}2^{n-1} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{C.22}$$

where matrices in C.22 correspond to the bilinear forms in C.21, in the order they appear in the equations. So, the HDG procedure to eliminate the local unknowns in terms of the trace unknown as presented in C.1, can be repeated and as a result, system can be solved for the trace of solution. It should be noted that, the system C.22, is linear system, hence, no need for non-linear solver. A part from that, in the thesis, most of examples are solved using a Crank-Nicolson time discretization. One can follow the same procedure as presented above for Crank-Nicolson and the final system is

$$\begin{bmatrix} \mathcal{T}1 + \frac{(-\mathcal{A}+\mathcal{D}1)}{2} & \frac{(-\mathcal{B}+\mathcal{B}n1)}{2} & \frac{(-\mathcal{C}+\mathcal{C}n1)}{2} & \frac{(\mathcal{A}n1-\mathcal{D}2)}{2} \\ (\mathcal{F}1+\mathcal{N}1) & \mathcal{G}1 & & -\mathcal{H}1 \\ (\mathcal{F}2+\mathcal{N}2) & & \mathcal{G}2 & -\mathcal{H}2 \\ \mathcal{D}3 & \mathcal{B}n2 & \mathcal{C}n2 & (\mathcal{A}n2-\mathcal{D}4) \end{bmatrix} \begin{bmatrix} \mathbb{U}^n \\ \mathbb{L}^n \\ \mathbb{W}^n \\ \hat{\mathbb{U}}^n \end{bmatrix} = \begin{bmatrix} \mathcal{S}^{n-1/2} + \mathcal{T}2^{n-1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$
$$- \begin{bmatrix} \frac{(-\mathcal{A}+\mathcal{D}1)}{2} & \frac{(-\mathcal{B}+\mathcal{B}n1)}{2} & \frac{(-\mathcal{C}+\mathcal{C}n1)}{2} & \frac{(\mathcal{A}n1-\mathcal{D}2)}{2} \\ (\mathcal{F}1+\mathcal{N}1) & \mathcal{G}1 & & -\mathcal{H}1 \\ (\mathcal{F}2+\mathcal{N}2) & & \mathcal{G}2 & -\mathcal{H}2 \\ \mathcal{D}3 & \mathcal{B}n2 & \mathcal{C}n2 & (\mathcal{A}n2-\mathcal{D}4) \end{bmatrix} \begin{bmatrix} \mathbb{U}^{n-1} \\ \mathbb{L}^{n-1} \\ \mathbb{W}^{n-1} \\ \hat{\mathbb{U}}^{n-1} \end{bmatrix} \tag{C.23}$$