Adjoint-Based *hp*-Adaptivity on Anisotropic Meshes for High-Order Compressible Flow Simulations

Adjungierte *hp*-Adaption auf anisotropen Gittern für Strömungssimulationen höherer Ordnung

Von der Fakultät für Maschinenwesen der
Rheinisch-Westfälischen Technischen Hochschule Aachen zur
Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Aravind Balan

Berichter: Univ.-Prof. Georg May, Ph.D.

apl. Prof. Dr. rer. nat. Siegfried Müller

Tag der mündlichen Prüfung: 02.05.2016

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Zusammenfassung

Numerische Verfahren höherer Ordnung, wie zum Beispiel Discontinuous Galerkin, Spectral Difference und Flux Reconstruction, bei denen die Lösung durch stückweise definierte Polynome mit keinerlei Stetigkeitsvorgaben zwischen Elementen dargestellt wird, werden immer mehr für die Simulation von Strömungsproblemen mit sehr hohen Geschwindigkeiten verwendet. Dies liegt vor allem daran, dass diese Verfahren das Potential besitzen, für eine gegebene Genauigkeit effizienter zu sein als klassische Finite Volumen Methoden. Die meisten Anwendungen im Ingenieurwesen haben nicht unbedingt die Berechnung des gesamten Strömungsfeldes als Ziel sondern vielmehr eine daraus resultierende Größe. In der Aerodynamik können dies zum Beispiel Auftriebs- oder Widerstandsbeiwerte sein. Die Kombination aus adjungierten Fehlerschätzern und Methoden höherer Ordnung stellt sich als sehr vielversprechend heraus, um diese Größen möglichst genau zu berechnen. Mit Hilfe der Fehlerschätzer können Elemente identifiziert werden, die am meisten zu dem globalen Fehler beisteuern und die es sich lohnen würde zu adaptieren. Letzteres kann entweder durch die lokale Verfeinerung des Rechengitters (h-Adaption), die Anpassung des lokalen Polynomgrades (p-Adaptation) oder aber einer Kombination aus beidem realisiert werden (hp-Adaption). Von diesen drei Strategien bietet die hp-Adaption den effizientesten Weg, da frei gewählt werden kann, was am besten geeignet ist, um lokale Lösungsmerkmale abzubilden.

Wir präsentieren adjungierte hp-adaptive Methoden auf isotropen und anisotropen Gittern in Kombination mit der in letzter Zeit entwickelten hybridisierten Discontinuous Galerkin Methode [86] für nichtlineare Konvektion-Diffusions-Probleme (inklusive der kompressiblen Euler und Navier-Stokes Gleichungen). hp-Adaption auf isotropen Gittern basiert auf einem lokalisierten adjungiertem Fehlerschätzer für ein gegebenes Ziel-funktional und einem Schätzer für die Regularität der Lösung. Im Falle von anisotropen Gittern erweitern wir die Verfeinerungsstrategie basierend auf Interpolationsfehlerschätzern von Dolejsi [30] um den Einsatz des adjungierten Fehlerschätzers. Mit Hilfe dieser beiden Fehlerschätzer werden sowohl die Größe als auch die Form der Dreieckselemente des Gitters für die nächsten Adaptationsschritte bestimmt. Hierbei wird das Konzept der Dualität zwischen einem Rechengitter und der zugehörigen Metrik benutzt. Hierbei werden Information über die einzelnen Elemente in einem Metrikfeld kodiert, welches dann an einen metrik-basierten Gittergenerator weitergegeben werden kann. Als Resultat erhält man ein Gitter mit der geforderten Anisotropie. Wir demonstrieren die Effektivität dieser Adaptationsmethode anhand numerischer Resultate für ein skalares Konvektion-Diffusions-Problem mit einer dünnen Grenzschicht,

reibungsfreien subsonischen, transsonischen und supersonischen und viskosen subsonischen Strömungen um ein NACA0012 Profil.

Abstract

High-order numerical methods such as Discontinuous Galerkin, Spectral Difference, and Flux Reconstruction etc, which use polynomials that are local to each mesh element to represent the solution field, are becoming increasingly popular in solving convection-dominated flows. This is due to their potential in giving accurate results more efficiently than lower order methods such as the classical Finite Volume methods. In most engineering applications, we are more interested in some specific scalar quantities rather than the full flow details. In the case of aerodynamic flow simulations, these quantities can be lift or drag coefficient. To get accurate values for such target functional quantities, adjoint-based error estimators, along with a high-order solver, have been found to be quite useful. They can identify the mesh elements that contribute the most to the error, and adapting these elements should result in a more accurate target functional. To adapt a mesh element, one can either do mesh refinement (h -adaptation) or polynomial space enrichment (p -adaptation) or both (hp -adaptation). Of these, hp -adaptation offers the most efficient way for adaptation, since one can locally choose between mesh refinement or polynomial space enrichment based on what is more efficient in resolving the local solution features.

We present efficient adjoint-based hp -adaptation methodologies on isotropic and anisotropic meshes for the recently developed high order Hybridized Discontinuous Galerkin scheme [86] for (nonlinear) convection-diffusion problems, including the compressible Euler and Navier-Stokes equations. hp -adaptation on isotropic meshes is based on the spatial error distribution for a given target functional given by the adjoint error estimator and the solution regularity given by a regularity indicator. For anisotropic meshes, we extend the refinement strategy based on an interpolation error estimate, due to Dolejsi [30], by incorporating an adjoint-based error estimate. Using the two error estimates we determine the size and the shape of the triangular mesh elements on the desired mesh to be used for the subsequent adaptation steps. This is done using the concept of mesh-metric duality, where the metric tensors can encode information about mesh elements, which can be passed to a metric-conforming mesh generator to generate the required anisotropic mesh. The effectiveness of the adaptation methodology is demonstrated using numerical results: for a scalar convection-diffusion case with a strong boundary layer; inviscid subsonic, transonic and supersonic flows and viscous subsonic flow around a NACA0012 airfoil.

Acknowledgements

First and foremost, I would like to thank my advisor Prof. Georg May for his supervision, kindness and encouragement throughout my time at AICES Graduate School. I consider my association with Georg as one of the most rewarding and formative experiences in my career. Apart from imparting his knowledge in numerical methods, Georg has always shown considerable patience while guiding me through various phases of my research. I am sure his students will continue to be inspired by his work ethic, knowledge, the rigour with which he pursues his research, and his sheer professionalism.

I would like to thank Prof. Siegfried Müller for being part of my doctoral committee. I really appreciate his painstaking review of my thesis. His insightful feedback was indispensable in having the thesis in its current refined form. I am also grateful to Prof. Herbert Olivier for taking the time and effort to chair the doctoral committee.

Many thanks are due to my colleague Michael Woopen with whom I did much of my doctoral research. Michael would always come to my aid whenever I had any issue with the code or any technical or theoretical query that needed clarification. His support was nothing short of significant during the course of my research.

I would also like to thank my colleagues Francesca and Jochen who were promptly obliging whenever I sought their help. I am also grateful to Ajay for our many stimulating, mutually beneficial discussions.

I consider it an honor to be a part of the research at AICES graduate school, and I appreciate all those who were part of my life here. A debt of gratitude is owed to the AICES service team led by Dr. Nicole Faber who did a wonderful job in supporting all of us. Nicole's administrative and organisational abilities deserve a special mention.

Finally, I would like to express my sincere thanks to my family and friends for being there for me always.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Anisotropic Adaptation (h and hp)	3
1.3	Contributions	7
1.3.1	Extending the HDG code to handle hp -adaptivity on isotropic mesh elements	7
1.3.2	Extending the HDG code to handle h - and hp -adaptivity on anisotropic mesh elements	7
1.4	Thesis Organization	8
2	Governing Equations	9
2.1	Introduction	9
2.2	Hyperbolic Conservation Laws	9
2.2.1	Weak solutions	11
2.2.2	Rankine-Hugoniot condition	12
2.2.3	Entropy solutions	12
2.3	Linear Advection Equation	13
2.4	The Euler Equations	14
2.4.1	Boundary conditions	14
2.5	The Navier-Stokes Equations	15
2.5.1	Boundary conditions	17
3	Numerical Discretization	19
3.1	Introduction	19
3.2	Discontinuous Galerkin Methods	19
3.2.1	Formulation	21
3.3	Hybridized Discontinuous Galerkin Methods	24
3.3.1	Relaxation	27
3.3.2	Hybridization	28
4	Adjoint-Based hp-Adaptation on Isotropic Meshes	31

4.1	Introduction	31
4.2	Adaptation Strategy	32
4.3	Adjoint-Based Error Estimation	32
4.4	Adjoint-Based hp -Adaptation	35
4.5	Numerical Results	37
4.5.1	Scalar convection-diffusion equation	37
4.5.2	Inviscid, subsonic flow	42
4.5.3	Inviscid, transonic flow	45
4.5.4	Viscous, subsonic flow	47
5	Adjoint-Based hp-Adaptation on Anisotropic Meshes	53
5.1	Introduction	53
5.2	Anisotropic h -Adaptation	55
5.2.1	Algorithm for h -adaptation	57
5.2.2	Finding the area of the desired mesh elements	57
5.2.3	Patch reconstruction	59
5.2.4	Finding the anisotropy of the desired mesh	60
5.3	Anisotropic hp -Adaptation	63
5.3.1	Algorithm for hp -adaptation	64
5.4	Verification of the Interpolation Error Bound	65
5.5	Numerical Results	71
5.5.1	Scalar convection-diffusion equation	75
5.5.2	Viscous, subsonic flow	82
5.5.3	Inviscid, subsonic flow	93
5.5.4	Inviscid, transonic flow	94
5.5.5	Inviscid, supersonic flow	100
6	Conclusions and Outlook	107
	Appendix	108
	Appendix	109
A	Appendix	109
A.1	High Order Derivatives of Dubiner Basis Functions on Triangles	109
A.1.1	Dubiner polynomials	109

A.1.2	Evaluation of the n -th derivatives of the Dubiner polynomials	110
	Index	113
	Bibliography	113

List of Figures

1.1	Anisotropic mesh adaptation for viscous subsonic (top) and inviscid supersonic (bottom) flows around the NACA0012 airfoil. Drag coefficient is used as the target functional.	5
1.2	Element-wise polynomial degrees on an adapted mesh for an inviscid supersonic flow case around the NACA0012 airfoil. Drag coefficient is used as the target functional.	6
4.1	Scalar convection-diffusion test case, $\epsilon = 0.005$, hp -adaptation, total weighted boundary flux as target functional.	39
4.2	Scalar convection-diffusion test case, $\epsilon = 0.005$, hp -adaptation, total weighted boundary flux as target functional, 9th adapted step with 1724 elements.	39
4.3	Scalar convection-diffusion test case, $\epsilon = 0.005$, hp -adaptation, total weighted boundary flux as target functional, polynomial map near the top right corner of the domain for different adaptation steps.	40
4.4	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of h - and hp -adaptivity, total weighted boundary flux as the target functional. . .	41
4.5	Scalar convection-diffusion test case, $\epsilon = 0.005$, 8th hp -adapted step, L^2 -norm of solution error as target functional.	43
4.6	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of h - and hp -adaptation, L^2 -norm of solution error as the target functional. . . .	44
4.7	Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, Mach contours on initial mesh.	44
4.8	Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, Mach contours on hp adapted mesh (2553 elements), 9th adaptation step.	45
4.9	Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, hp -adaptation ($p = 1, \dots, 6$), 2553 elements, 9th adaptation step.	46
4.10	Inviscid subsonic flow around NACA 0012 airfoil, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$. Comparison of h - and hp -adaptation. For hp -adaptation, $p = 1, \dots, 6$. . .	46
4.11	Inviscid transonic flow, $\text{Ma}_\infty = 0.8$, $\alpha = 1.25^\circ$, 14th adaptation step, 3552 elements.	47

4.12	Inviscid transonic flow around the NACA0012 airfoil, $\text{Ma}_\infty = 0.8$, $\alpha = 1.25^\circ$, comparison of h - and hp -adaptation, drag coefficient as the target functional.	48
4.13	Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, hp -adaptation, drag coefficient as the target functional.	49
4.14	Viscous subsonic flow around the NACA0012 airfoil, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, hp -adaptation, drag coefficient as the target functional. . .	50
4.15	Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of h - and hp -adaptation. For hp -adaptation, $p = 1, \dots, 6$	51
5.1	Ellipse which encloses the triangular mesh element	56
5.2	Patch for the element k	59
5.3	Scalar boundary layer case, $\epsilon = 0.1$, Patch reconstruction.	60
5.4	Verification of the inequality Eq. (5.20) on uniformly refined meshes and varying polynomial approximation degrees. (---): $\ E_\Omega\ _{L^2(\Omega)}$, (—■—): $\ \tilde{B}_\Omega\ _{L^2(\Omega)}$	66
5.5	Verification of the inequality Eq. (5.20) on adjoint-based adapted anisotropic meshes and varying polynomial approximation degrees. (---): $\ E_\Omega\ _{L^2(\Omega)}$, (—■—): $\ \tilde{B}_\Omega\ _{L^2(\Omega)}$, (—○—): $\ B_\Omega\ _{L^2(\Omega)}$	68
5.6	Element-wise verification of inequality Eq. (5.20) on adapted anisotropic meshes. The numbering of the elements is sorted with respect to the L^2 -norm of the error bound.	69
5.7	Element-wise verification of inequality Eq. (5.20) on an adapted anisotropic mesh. The numbering of the elements is sorted with respect to the L^2 -norm of the error bound. $\ E_\Omega\ _{L^2(\Omega_k)}$ is evaluated using approximate derivatives.	70
5.8	Initial mesh for the compressible flow test cases around the NACA0012 airfoil	72
5.9	Scalar convection-diffusion test case, $\epsilon = 0.005$, L^2 -norm of solution error as the target functional, comparison of error convergence for various parameter settings for anisotropic hp -adaptivity	74
5.10	Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of error convergence for various parameter settings for anisotropic hp -adaptivity.	75
5.11	Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic h -adaptation, $p = 2$	77

5.12	Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic h -adaptation, $p = 2$	78
5.13	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of error convergence for isotropic and anisotropic h -adaptation. 'iso' denotes isotropic refinement, $p = 2$, total weighted boundary flux as target functional.	78
5.14	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of error convergence for isotropic and anisotropic hp -adaptation, 'iso' denotes isotropic refinement, total weighted boundary flux as target functional.	79
5.15	Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic hp -adaptation, total weighted boundary flux as target functional.	80
5.16	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of anisotropic h - and hp -adapted mesh which give the same error level of around 10^{-10} , total weighted boundary flux as target functional, magnified view near the boundary layer.	81
5.17	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of isotropic and anisotropic h -adaptation, L^2 -norm of solution error as the target functional, 'iso' denotes isotropic refinement.	82
5.18	Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic hp -adaptation, 5th adapted step, 1231 elements, L^2 -norm of solution error as the target functional.	83
5.19	Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic hp -adaptation, 5th adapted step, 1231 elements, L^2 -norm of solution error as the target functional, magnified view near the boundary layer.	83
5.20	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of isotropic and anisotropic hp -adaptation, L^2 -norm of solution error as the target functional, 'iso' denotes isotropic refinement.	84
5.21	Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of isotropic and anisotropic h - and hp -adaptation, L^2 -norm of solution error as the target functional, 'iso' denotes isotropic refinement.	84
5.22	Viscous subsonic flow around the NACA0012, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, $p = 2$, h -adaptation with drag coefficient as target.	87
5.23	Viscous subsonic flow around the NACA0012, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, $p = 2$, h -adaptation with drag coefficient as target.	88
5.24	Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of error convergence for isotropic and anisotropic h -adaptation. 'iso' denotes isotropic refinement, $p = 2$	88

5.25	Viscous subsonic flow around the NACA0012, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, hp -adaptation with drag coefficient as target. $p = 1, \dots, 6$. . .	89
5.26	Viscous subsonic flow around the NACA0012, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, hp -adaptation with drag coefficient as target. $p = 1, \dots, 6$. . .	90
5.27	Viscous subsonic flow, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, comparison of error convergence for isotropic and anisotropic hp -adaptation. 'iso' denotes isotropic refinement, $p = 1, \dots, 6$	90
5.28	Viscous subsonic flow, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic refinement. For hp -adaptation, $p = 1, \dots, 6$	91
5.29	Viscous subsonic flow, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, comparison of error convergence with respect to the work units for isotropic and anisotropic adaptations. 'iso' denotes isotropic refinement.	93
5.30	Inviscid subsonic flow, $Ma_\infty = 0.5$, $\alpha = 2^\circ$, h -adaptation, $p = 3$	94
5.31	Inviscid subsonic flow, $Ma_\infty = 0.5$, $\alpha = 2^\circ$, comparison of error convergence for isotropic and anisotropic h -adaptation. 'iso' denotes isotropic adaptation. $p = 3$	95
5.32	Inviscid subsonic flow, $Ma_\infty = 0.5$, $\alpha = 2^\circ$, hp -Adaptation, 1674 mesh elements, $p = 1, \dots, 6$	95
5.33	Inviscid subsonic flow, $Ma_\infty = 0.5$, $\alpha = 2^\circ$, hp -Adaptation, 1674 mesh elements, $p = 1, \dots, 6$	96
5.34	Inviscid subsonic flow, $Ma_\infty = 0.5$, $\alpha = 2^\circ$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic adaptation. For hp -adaptation, $p = 1, \dots, 6$	96
5.35	Inviscid transonic flow around the NACA0012, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, $p = 2$, h -adaptation with drag coefficient as target.	98
5.36	Inviscid transonic flow, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, comparison of error convergence for isotropic and anisotropic h -adaptation, 'iso' denotes isotropic adaptation, $p = 2$	99
5.37	Inviscid transonic flow around the NACA0012, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, hp -adaptation, $p = 1, \dots, 6$	100
5.38	Inviscid transonic flow, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic adaptation, For hp -adaptation, $p = 1, \dots, 6$	101
5.39	Inviscid supersonic flow around the NACA0012, $Ma_\infty = 1.5$, $\alpha = 0^\circ$, $p = 2$, h -adaptation with drag coefficient as target.	102

5.40	Inviscid supersonic flow, $Ma_\infty = 1.5$, $\alpha = 0^\circ$, comparison of error convergence for isotropic and anisotropic adaptations, 'iso' denotes isotropic h -refinement. $p = 3$	103
5.41	Inviscid supersonic flow around the NACA0012, $Ma_\infty = 1.5$, $\alpha = 0^\circ$, $p = 2$, h -adaptation with drag coefficient as target.	103
5.42	Inviscid supersonic flow around the NACA0012, $M_\infty = 1.5$, $\alpha = 0^\circ$, hp -adaptation with drag coefficient as target, $p = 1, \dots, 6$	104
5.43	Inviscid supersonic flow, $M_\infty = 1.5$, $\alpha = 0^\circ$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic adaptation, for hp -adaptation, $p = 1, \dots, 6$	105

1 Introduction

1.1 Background and Motivation

The advancement in computational power and increased research on numerical algorithms in recent decades have enabled widespread industrial application of numerical simulations. The design cycle costs, across diverse fields, have been economized as a result. The present work is concerned with the numerical simulation of fluid flows - compressible flows, in particular. For compressible flow simulations, the classical Finite Volume scheme [12] is the most popular and widely used numerical method. With proven reliability, stability and robustness, it has been the method of choice for industrial applications for quite a long time.

For flows where high accuracy is needed, as turbulent, aeroacoustic, and vortex dominated flows, methods of first and second order of consistency are not sufficient. Henceforth we make the distinction that such methods are referred to as "low order methods", and those higher than second order of consistency as "high order methods". For flows where high accuracy is needed, the use of low order methods will result in very high mesh resolution which makes the computation prohibitive. It is in this context that higher order methods have started gaining attention in academia [81]. Higher order methods are potentially more efficient than low order methods [82]. In case of smooth flows, it has been shown that high order methods give results of the same accuracy level as that of lower order methods, but with fewer degrees of freedom [39]. This feature of high order methods in case of smooth flows, can be exploited even in flows with discontinuities by using hp-adaptivity [54, 50, 29, 15]. In hp-adaptivity, the solution approximation space can be locally changed based on the local regularity of the solution.

Various high order methods have been developed to solve compressible flows. The most popular high order method used to solve convection dominated flows is arguably the discontinuous Galerkin (DG) method [24, 18, 26, 23, 25]. In DG methods, the solution variables are represented as polynomials that are local to each mesh element. The local order can be arbitrarily increased by increasing the polynomial approxima-

tion degree. There are other similar DG-like methods which use locally discontinuous polynomials, such as Spectral Difference [62, 7], Flux Reconstruction [51, 79], Correction Procedure via Reconstruction [74], etc. Each of them, having evolved in a short time, has shown potential in solving compressible flows with high order of accuracy. As the viability and competence of high order methods are now established in academic problems, efforts have been underway for their industrial application. One such major effort was the European Union project ADIGMA - Adaptive Higher-Order Variational Methods for Aerodynamic Applications in Industry [55], which started in 2006 and ended in 2009. The main aim of the project was to develop next generation CFD tools with significantly improved efficiency, accuracy and reliability for advanced aerodynamic applications. Several universities and industries in Europe were part of this project. Significant progress was made in the development of adaptive high order methods such as the Discontinuous Galerkin method and Continuous Residual Distribution schemes [2]. The project also identified limitations for further industrialization of higher order methods, and it seemed that the the development of efficient solvers for industrially relevant applications still remained a major challenge. As a follow-up of the ADIGMA project, IDIHOM -Industrialization of High-Order Methods, has started in 2011, due to the increased demand from the European aerospace industries to advance their CFD-aided design procedures. The aim was to further advance high order methods and apply them to complex industrial flows. Significant progress has been made in the development of high order methods and in their applications to aerodynamic flow cases [1]. There was an extension of high order methods to multidisciplinary applications such as aero-acoustics and aero-elastics. Despite the progress achieved, challenges exist which prohibit incorporation of these tools into routine industrial process. Also, there is still room for improvement in solver efficiency and robustness.

One disadvantage of the Discontinuous Galerkin high-order method is the large number of degrees of freedom resulting from relaxing the continuity constraint of the solution across element interfaces. Recently, a technique called hybridization has been found useful in reducing the number of globally coupled degrees of freedom and hence the size of the global system that has to be solved at each iteration step. Here, the coupled unknowns, also known as the hybrid variables, have support only on the element interfaces. Hybridized DG discretizations were presented by Nguyen et al. for linear [63] and nonlinear [64] convection-diffusion equations. Peraire et al. [67] have extended the method for the Euler and Navier-Stokes equations and have shown its viability in aerodynamic flow simulations. In a slightly different approach, Egger et al. [33] have presented the hybridized formulation for the linear convection-diffusion equation,

where the convection term is discretized with a DG method and the diffusion term with a hybrid mixed method. Schütz and May [73] have extended this for compressible Euler and Navier-Stokes equations with promising results.

In most engineering applications, one may not be interested in full flow details, but rather in some specific quantities. In external aerodynamics, these may be lift or drag coefficients. With the aim of getting accurate values for such functional quantities in the most efficient way, target-based error control methods have been developed. One such method is based on the adjoint solution of the original governing equations. In this method, an additional linear system is solved which then gives an estimate on the spatial error distribution contributing to the error in the target functional. This estimate can be used as a criterion for local adaptation - either h -adaptation (mesh refinement) or for p -adaptation (polynomial space enrichment by increasing the local polynomial degree)- so that the error in the target functional is expected to get reduced in the next simulation run. Obviously, this local refinement strategy is more efficient than a uniform refinement where even the cells that contribute little to the error in the target functional get adapted thereby increasing the computational cost unnecessarily. The most common adaptation based on an adjoint formulation is h -adaptivity (mesh refinement alone). Schütz and May [72] and Woopen et al. [86] devised and implemented an adjoint-based mesh adaptation procedure based on a class of adjoint-consistent hybridized finite element methods. The method has also been extended to solve 3D problems and turbulent flows using the RANS approach by Woopen et al. [83, 84]. For aerodynamic flows, which have diverse flow features including shocks, arguably the best approach is hp -adaptivity - varying both mesh size and polynomial degree - since we have the choice of determining mesh refinement or polynomial enrichment based on the flow features. Adjoint-based hp -adaptivity has already been found efficient in error reduction in the context of discontinuous Galerkin methods [45, 44, 80] and also in the CPR methods [74].

1.2 Anisotropic Adaptation (h and hp)

Many flows, especially aerodynamic flows, exhibit highly anisotropic features like boundary layers, shocks, etc. To efficiently resolve these features, one should have a mesh with very high resolution across the region of high gradient or the derivatives of the solution or even a discontinuity, and lesser resolution along the direction where the solution does not change much. Anisotropic mesh elements (highly skewed elements

in one direction) are perfect in such situations and reduce the degrees of freedom substantially. Fig. 1.1 shows such meshes with highly anisotropic elements which align with the flow features. Figs. 1.1a, 1.1b are for a viscous subsonic flow test case around the NACA0012 airfoil. Anisotropic elements stretched along the boundary layer can resolve the flow more efficiently than isotropic mesh elements. Figs. 1.1c, 1.1d are for an inviscid supersonic case, where there is a bow shock in front of the airfoil. It is evident that the anisotropic elements could resolve the shock quite well.

One of the ways to generate the desired anisotropic meshes, is to use metric conforming mesh generators. The key idea is to encode the desired mesh element information such as the size, orientation and stretching ratio in a field of symmetric metric tensors. The metric conforming mesh generator will then generate meshes which conform to the input metric. When used in the adaptation process, at each step a new mesh is generated for the entire computational domain (re-meshing), which allows the flexibility of coarsening or refining in the regions we want. Along with adjoint error estimation, metric-based remeshing would provide an efficient way to distribute the degrees of freedom in order to get an accurate value for the target functional. There have been many attempts so far in using adjoint error estimate with metric based adaptations [78, 66, 36, 59, 87]. The adjoint-based anisotropic refinement algorithm was first devised by Venditti and Darmofal [78], which was applied to a finite volume scheme with piecewise linear reconstruction. This was extended to high order schemes by Fidkowski [34]. The basic idea is to construct a mesh where the size of the elements is determined by the adjoint error estimator and the anisotropy is determined by requiring that the interpolation errors are the same along each edge for each mesh element. In a slightly different approach of Dolejsi [30], the anisotropy is chosen in such a way that the interpolation error in the L^q -norm ($q \in [1, \infty)$) is minimized. In the present work, we extend the adaptation strategy of Dolejsi [30] to incorporate an adjoint-based error estimate to perform h -adaptation on anisotropic elements. The same strategy is also used for hp -adaptivity, where we use both the adjoint-based error estimator and the interpolation error estimate to determine the desired mesh and the polynomial degree distribution on the desired mesh.

Fig. 1.2 shows the polynomial degree distribution on an anisotropic mesh generated using our methodology for one the test cases (inviscid supersonic flow test case around NACA0012) which we present in detail in Chapter 5.

Figure 1.1: Anisotropic mesh adaptation for viscous subsonic (top) and inviscid supersonic (bottom) flows around the NACA0012 airfoil. Drag coefficient is used as the target functional.

Figure 1.2: Element-wise polynomial degrees on an adapted mesh for an inviscid supersonic flow case around the NACA0012 airfoil. Drag coefficient is used as the target functional.

1.3 Contributions

The starting point of the present work is an HDG code that had the capability of doing adjoint-based h -adaptation on isotropic triangular mesh elements [86].

Two main contributions can be attributed to the present thesis work.

1.3.1 Extending the HDG code to handle hp -adaptivity on isotropic mesh elements

The code was modified to handle variable polynomial degrees on the mesh elements. The work also involved trying out various smoothness sensors, available in literature [80, 69, 65], to choose between h - or p -refinement, and obtaining the most robust sensor. Results using one such sensor, based on the decay of expansion coefficients, were presented in an AIAA conference [8]. In the present work, a more robust sensor which is based on a measure of the inter-element jumps in the solution [29] is employed to obtain the presented results. The effectiveness of the hp -adaptation methodology is demonstrated using numerical experiments, involving a scalar model problem with a strong boundary layer, as well as various compressible flow test cases.

1.3.2 Extending the HDG code to handle h - and hp -adaptivity on anisotropic mesh elements

The main part of the thesis was the extension of the code to perform h - and hp -adaptations on anisotropic triangular elements. The work mainly involved the generation of suitable metric tensors that encode the details of the desired anisotropic mesh for the subsequent iteration. A metric-conforming anisotropic mesh generator was then used to generate the desired mesh. In this process, a completely new mesh is generated (remeshing) unlike the local mesh refinement by element subdivision done in the case of isotropic adaptation. The main novelty of the present work is the use of adjoint error estimator in a metric-based hp -adaptation methodology proposed by Dolejsi [30], that was using only the interpolation error estimate in generating the metric tensors. To the best of our knowledge, this is the first attempt to combine adjoint-based hp -refinement into a metric-based mesh-adaptation framework. Another novelty of the work is the explicit numerical verification of the interpolation error model of Dolejsi. We could demonstrate that both h - and hp -adaptations on anisotropic meshes are significantly more efficient than the adaptations on isotropic meshes in terms of the number of degrees of freedom required to achieve the same level of error. The results of the work has

been presented in an AIAA conference [9] and the extended version has been published in a journal [10].

1.4 Thesis Organization

The rest of the thesis is organized as follows. The governing equations (Eulers and Navier-Stokes equations) and associated basic concepts of hyperbolic conservations laws are introduced in Chapter 2. The details of the numerical scheme, which includes the spatial discretization (Hybridized discontinuous Galerkin), time relaxation, etc is given in Chapter 3. This Chapter also briefly reviews the standard discontinuous Galerkin method, from which the HDG method inherits basic properties. Together with a short introduction on adjoint error estimator, the hp -adaptation methodology on isotropic meshes and the numerical results are given in Chapter 4. In Chapter 5, the whole adaptation (h and hp) strategy on anisotropic meshes is described, and numerical results for various test cases are provided to demonstrate the effectiveness of the adaptation methodology. Conclusions follow in Chapter 6.

2 Governing Equations

2.1 Introduction

The equations that are of interest for solving compressible flow problems are the Euler equations in case of inviscid flows, and the Navier-Stokes equations for viscous flows. Both set of equations are systems of conservation laws that model conservation of such physical quantities as mass, momentum, energy, and are nonlinear in nature. The Euler equations are hyperbolic conservation laws while the Navier-Stokes contain both hyperbolic and parabolic components, but hyperbolic part can be dominated for high Reynolds number flows. Hence, the concept of hyperbolicity is important to understand the nature of the flow and to devise suitable numerical methods to solve the equations. In this chapter, basic concepts and features of hyperbolic conservation laws are reviewed. The concept of weak solutions in case of nonlinear hyperbolic conservation laws, and conditions to get unique entropy solutions are also dealt with. Finally, the complete Euler and the Navier-Stokes equations, to be solved, numerically, along the associated boundary conditions, are provided.

2.2 Hyperbolic Conservation Laws

Consider a Cauchy problem for the variable w , i.e.

$$\begin{aligned}\frac{\partial w}{\partial t} + \nabla \cdot f(w) &= 0, \\ w(0, x) &= w_0(x), \quad x \in \mathbb{R}^d,\end{aligned}\tag{2.1}$$

where $w = (w_1, w_2, \dots, w_n)$ is a vector in $\mathbb{R}^n$ and f defines a mapping from $\mathbb{R}^n$ into $\mathbb{R}^{n \times d}$, where d is the number of spatial dimensions, and $w_0(x)$ gives the initial value of w . Equations that can be written in the form of Eq. (2.1) are called conservation laws, where w is the vector of conserved quantities and f is the vector of flux functions. The flux functions, in general, can be nonlinear in the state variable w . The fact that the above equation represents a conservation of certain quantity is manifested in the

integral form of the equation given as

$$\frac{d}{dt} \int_{\Omega} w dx + \int_{\partial\Omega} f \cdot n ds = 0, \quad (2.2)$$

where Ω is any given control volume, and $\partial\Omega$ is its boundary. The physical meaning of this equation is that the integral of the quantity w in the control volume Ω can change only due to the flux f across the boundary $\partial\Omega$.

Eq. (2.1) can also be written in the quasi-linear form as

$$\frac{\partial w}{\partial t} + \sum_{i=1}^d A_i \frac{\partial w}{\partial x_i} = 0, \quad (2.3)$$

where $A_i := \frac{\partial f_i}{\partial w}$ is the flux Jacobian.

The system of conservation laws given by Eq. (2.1) is said to be hyperbolic if any linear combination of the Jacobian matrices A_i 's is diagonalizable with n real eigenvalues, and strictly hyperbolic if the eigenvalues are distinct. This means that for any $w \in \mathbb{R}^n$ and any $\eta \in \mathbb{R}^d \setminus \{0\}$, the linear combination of the Jacobian matrices,

$$A := \sum_{i=1}^d \eta_i A_i, \quad (2.4)$$

will have complete set of n linearly independent eigenvectors and the corresponding n real eigenvalues. In this case, the matrix A can be diagonalized as $A = Y\Lambda Y^{-1}$, where Λ is a diagonal matrix with the eigenvalues of A as its entries, and Y is the matrix with the corresponding eigenvectors as its columns. For the one-dimensional case, in particular, with A considered as a constant matrix, the equation system Eq. (2.3) can be decoupled into a system of n independent scalar advection equations via transformation into the characteristic variables $\tilde{w} = Y^{-1}w$, to give

$$\frac{\partial \tilde{w}_j}{\partial t} + \lambda_j \frac{\partial \tilde{w}_j}{\partial x} = 0, \quad (2.5)$$

$$\tilde{w}_j(x, 0) = \tilde{w}_{j,0}(x), \quad j = 1, \dots, n. \quad (2.6)$$

These can be solved exactly as

$$\tilde{w}_j(x, t) = \tilde{w}_{j,0}(x - \lambda_j t), \quad j = 1, \dots, n. \quad (2.7)$$

Thus, each characteristic solution $\tilde{w}_j$ is transported in the $x-t$ plane along the straight line $(x - \lambda_j t)$, called a characteristic. The conservative solution w is found as the linear

combination of the characteristic variables as $w = Y\tilde{w}$. Thus, the solution consists of a combination of n waves, each being a constant along the straight line with slope $\frac{dx}{dt} = \lambda_j$ in the $x-t$ plane. Since the λ_j 's are constant values, the characteristic lines emanating from different points in the x axis will be parallel. But, for the general nonlinear case, where f is a nonlinear function of w , the slope of the lines may not be constant. The characteristic lines may intersect, leading to the formation of discontinuities, even if the initial solution is smooth. The function w is said to be a classical solution to Eq. (2.1) if w is a C^1 function that satisfies the Eq. (2.1) pointwise. But a unique classical solution for nonlinear hyperbolic conservation laws cannot be guaranteed due to the intersection of the characteristics. This motivates the introduction of the weak formulation of the governing equations.

2.2.1 Weak solutions

To derive the weak formulation, we multiply the conservation law (Eq. (2.1)) with smooth test functions $\varphi \in C_0^1(\mathbb{R}^d \times \mathbb{R}^+)^n$, where $\mathbb{R}^+ := [0, \infty)$, to get

$$\int_{\mathbb{R}^d \times \mathbb{R}^+} \left(\frac{\partial w}{\partial t} + \nabla \cdot f \right) \varphi \, dx dt = 0. \quad (2.8)$$

The above equation, upon integration by parts, gives the weak formulation of Eq. (2.1),

$$\int_{\mathbb{R}^d \times \mathbb{R}^+} \left(w \frac{\partial \varphi}{\partial t} + f \cdot \nabla \varphi \right) dx dt - \int_{\mathbb{R}^d} w_0(x) \varphi(x, 0) \, dx = 0, \quad (2.9)$$

for all $\varphi \in C_0^1(\mathbb{R}^d \times \mathbb{R}^+)^n$, the continuously differentiable functions with compact support. Note that a function φ has compact support if it vanishes outside some bounded set. It is clear from the above equation that the solution is not required to be differentiable and hence the weak formulation admits discontinuous solutions. We can thus look for weak solutions in a broader class of functions unlike the restricted set of solutions for the strong form of the equation. If the initial state is locally bounded, i.e., $w_0 \in L_{loc}^\infty(\mathbb{R}^d)^n$, then $w \in L_{loc}^\infty(\mathbb{R}^d \times \mathbb{R}^+)^n$ is said to be a weak solution of Eq. (2.1) if w satisfies Eq. (2.9) for any $\varphi \in C_0^1(\mathbb{R}^d \times \mathbb{R}^+)^n$. If a weak solution w is in $C^1(\mathbb{R}^d \times \mathbb{R}^+)^n$, then it is also a classical solution, i.e. it satisfies Eq. (2.1) point-wise. Note that a classical solution is also a weak solution by construction. The class of weak solutions is too large to ensure uniqueness of solutions. An important class of functions are piecewise classical solutions with discontinuities separating the smooth regions. However, there are restrictions imposed on these discontinuities such that the solution is a weak solution. The following Rankine-Hugoniot condition states

the condition for which a solution with discontinuities forms a weak solution.

2.2.2 Rankine-Hugoniot condition

Given a surface of discontinuity on the $x-t$ plane, if we define $w_{\pm} = \lim_{\epsilon \rightarrow 0+} (u(x \pm \epsilon n, t))$, where $n = (n_{x_1}, n_{x_2}, \dots, n_{x_d}, n_t)$ is the normal on the surface of discontinuity, then w is a weak solution of Eq. (2.9) if and only if w is a classical solution in region where $w \in C^1$ and w satisfies the algebraic expression

$$(w_+ - w_-)n_t + \sum_{j=1}^d (f_j(w_+) - f_j(w_-))n_{x_j} = 0 \quad (2.10)$$

across jump discontinuities. Eq. (2.10) is known as the Rankine-Hugoniot condition [42].

2.2.3 Entropy solutions

The weak formulation, even with the Rankine-Hugoniot condition, does not give a unique solution and one should identify the physically admissible solution among the weak solutions. Such admissible solutions turn out to be the entropy solutions which take into account the fact that entropy does not decrease with time. To give a mathematical definition of the entropy solutions, the entropy functions of the conservative laws need to be defined. A convex function $E(w)$ is called an entropy function of Eq. (2.1), if there exist functions F_i , $i = 1, \dots, d$, called entropy fluxes, such that

$$\nabla E(w)A_i = \nabla F_i(w), \quad i = 1, \dots, d, \quad (2.11)$$

where the gradients are taken with respect to w , and $A_i = \frac{\partial f_i}{\partial w}$ is the Jacobian matrix of the i^{th} flux function. A weak solution to Eq. (2.9) is an entropy solution if w satisfies for all entropy functions $E(w)$ the condition

$$\int_{\mathbb{R}^d \times \mathbb{R}^+} \left(E(w) \frac{\partial \varphi}{\partial t} + \sum_{i=1}^d F_i \frac{\partial \varphi}{\partial x_i} \right) dx dt + \int_{\mathbb{R}^d} E(w_0) \varphi(x, 0) dx \geq 0, \quad (2.12)$$

for all test functions $\varphi \in C_0^1(\mathbb{R}^d \times \mathbb{R}^+)$, $\varphi \geq 0$. It is possible to deduce that w is an entropy solution to Eq. (2.1), if the following conditions hold:

- w is a classical solution that satisfies Eq. (2.1) pointwise in regions where $w \in C^1$,
- w satisfies the Rankine-Hugoniot condition Eq. (2.10),

- w satisfies the entropy jump inequality,

$$(E(w_+) - E(w_-))n_t + \sum_{j=1}^d (F_j(w_+) - F_j(w_-))n_{x_j} \leq 0. \quad (2.13)$$

Finding entropy functions associated with nonlinear system of conservation laws is in general not trivial. For a scalar conservation law, any convex function is an entropy function and it can be shown that the entropy solutions are unique [42]. Another way of finding the physically admissible solution is by using the concept of vanishing viscosity. Here the unique weak solution has been identified as the limit $\lim_{\epsilon \rightarrow 0^+} w^\epsilon$, where w^ϵ is the solution of the viscous equation (adding a viscous term to the conservation law)

$$\frac{\partial w^\epsilon}{\partial t} + \nabla \cdot f(w^\epsilon) = \epsilon \nabla^2 w^\epsilon, \quad (2.14)$$

with $\epsilon > 0$. For this parabolic problem, in the scalar case, one can show that a unique smooth solution w_ϵ exists. It can also be shown that if $w_\epsilon \rightarrow w$ as $\epsilon \rightarrow 0^+$, then w is an entropy solution to the conservation law Eq. (2.1). The concept of vanishing viscosity has an important role in shock-capturing using numerical methods, where it helps to get non-oscillatory solutions. The key idea is to add an artificial viscosity (diffusion) term to the original partial differential equation. The viscosity coefficient ϵ is chosen in such a way that the term is active only in non-smooth regions, thereby regularizing the solution, and goes to zero as the mesh size goes to zero.

2.3 Linear Advection Equation

One of the simplest hyperbolic conservation laws is the scalar advection equation which models the transport of a scalar conserved variable, w , as:

$$\frac{\partial w}{\partial t} + \nabla \cdot f(w) = 0, \quad (2.15)$$

where $f = cw$, and $c = (c_1, c_2, \dots, c_d)$ is the advection velocity. Given initial conditions at $t = 0$, the analytical solution of this equation is given as

$$w(x, t) = w(x - ct, 0) = w_0(x - ct) \quad (2.16)$$

From the above equation, it is clear that the solution is constant on $\xi = x - ct$, the characteristics. The equation is hyperbolic as it has c_i as real eigenvalue of each flux

Jacobian $\frac{\partial f_i}{\partial u}$.

2.4 The Euler Equations

Inviscid compressible flows, in which viscous stresses and heat conduction are neglected, are modeled by the Euler equations given in 2D in the conservative form as

$$\partial_t w + \nabla \cdot f_c(w) = 0. \quad (2.17)$$

w is the state vector of conservative variables and $f_c = (f_{c,1}, f_{c,2})$ is the vector of convective flux functions given by

$$w = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad f_{c,1} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{bmatrix} \quad f_{c,2} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{bmatrix}, \quad (2.18)$$

where ρ is the density of the fluid, u is the x -velocity, v is the y -velocity, p is the pressure, and E is the total energy. This set of four coupled and nonlinear equations models the conservation of mass, x -momentum, y -momentum, and total energy, respectively.

To close the system of equations, we assume the fluid to be an ideal gas for which the relation between pressure and energy is

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2) \right), \quad (2.19)$$

where γ is the specific heat capacity ratio, which is 1.4 for air. It is possible to show under the ideal gas law assumption that the flux Jacobians $\frac{\partial f_c}{\partial w}$ have real eigenvalues, which means Eq. (2.17) together with Eq. (2.18) form a hyperbolic system of partial differential equations. It can be shown that in one dimension, the Euler equations are strictly hyperbolic whereas in two and three dimensions, they are hyperbolic [77].

2.4.1 Boundary conditions

The two main boundary conditions we use when we solve the Euler equations are slip boundary conditions at wall boundaries and far-field boundary conditions based on characteristic upwinding at far-field. Slip boundary means that the flow is parallel to the wall and there is no flow penetrating the wall. Hence the normal component of the

velocity is zero, i.e.

$$(u, v) \cdot (n_x, n_y) = 0. \quad (2.20)$$

A boundary function w^b can be defined for the conservative variable w which satisfies the above slip condition as

$$w^b = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 - n_x^2 & -n_x n_y & 0 \\ 0 & -n_x n_y & 1 - n_y^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} w. \quad (2.21)$$

At the far field, the boundary conditions can be defined by using characteristic upwinding. Here, the normal component of the flux Jacobian is decomposed as

$$f'(w) \cdot n = Q(w, n) \cdot \Lambda(w, n) \cdot Q^{-1}(w, n), \quad (2.22)$$

where Λ is a diagonal matrix with entries as the eigenvalues of $f'(w) \cdot n$, Q is the matrix with the column vectors as the corresponding eigenvectors, and n is the outward pointing normal. The solution variables are also transformed to their characteristic variables as $(w_c)_i := (Q^{-1}w)_i$, which propagate with the characteristic velocities $\Lambda(w, n)_{i,i}$. Now the boundary conditions are prescribed based on the direction of the characteristics. If the characteristics enter the domain, then the far-field values, w_∞ , are taken as the boundary values, and if they flow out of the domain, then the values from the interior are considered as the boundary values. In other words, the far field values are imposed as boundary values only to those variables for which there holds $\Lambda(w, n)_{i,i} < 0$. The boundary state is defined as

$$w^b = Qw_c^b, \quad (2.23)$$

where the boundary state for the characteristic variable, w_c^b , is defined as

$$(w_c^b)_i = \begin{cases} (w_c)_i, & (\Lambda(w, n))_{i,i} \geq 0 \\ (w_{c,\infty})_i, & (\Lambda(w, n))_{i,i} < 0. \end{cases} \quad (2.24)$$

2.5 The Navier-Stokes Equations

The Navier-Stokes equations are the viscous complement of the Euler equations for a Newtonian fluid. In conservative form they are given as

$$\partial_t w + \nabla \cdot (f_c(w) - f_v(w, \nabla w)) = 0. \quad (2.25)$$

The convective part of the Navier-Stokes equations coincides with the Euler equations. Thus, we only state the viscous flux in the following.

$$f_{v,1} = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{11}u + \tau_{12}v + kT_x \end{bmatrix} \quad f_{v,2} = \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{21}u + \tau_{22}v + kT_y \end{bmatrix}. \quad (2.26)$$

The above expression assumes that the Fourier's law of conduction [60], which connects the heat flux and the temperature gradient using the thermal conductivity coefficient k , is valid. The temperature is defined via the ideal gas law

$$T = \frac{\mu\gamma}{k \cdot \text{Pr}} \left(\frac{E}{\rho} - \frac{1}{2} (u^2 + v^2) \right) = \frac{1}{(\gamma - 1)c_v} \frac{p}{\rho}, \quad (2.27)$$

where $\text{Pr} = \frac{\mu c_p}{k}$ is the Prandtl number, which has a value of around $\text{Pr} = 0.72$ for air at moderate conditions. c_p and c_v are specific heat at constant pressure and constant volume, respectively. They are related via the specific heat capacity ratio, γ as $\gamma = \frac{c_p}{c_v}$. For a Newtonian fluid, the stress tensor is defined as

$$\tau = \mu \left(\nabla \hat{w} + (\nabla \hat{w})^T - \frac{2}{3} (\nabla \cdot \hat{w}) \text{Id} \right), \quad (2.28)$$

where $\hat{w} := (u, v)^T$ and Id is the identity matrix of size 2×2 . The variation of dynamic viscosity μ as a function of temperature is determined by Sutherland's law [76] as

$$\mu = \frac{C_1 T^{3/2}}{T + C_2}, \quad (2.29)$$

where $C_1 = 1.458 \times 10^{-6} \frac{\text{kg}}{\text{ms}\sqrt{\text{K}}}$ and $C_2 = 110.4 \text{ K}$. The equations above are in dimensional form, and in practice, they are non-dimensionalised before solving. In the non-dimensional form, the flow can be fully described by the Mach number Ma , the Reynolds number Re , and the constants, Prandtl number Pr , and specific heat capacity ratio γ . The Reynolds number is defined as

$$\text{Re} = \frac{U_0 L \rho_0}{\mu_0}, \quad (2.30)$$

where U_0 is the reference speed, L is the reference length, ρ_0 is the reference density and μ_0 is the reference viscosity.

2.5.1 Boundary conditions

While solving the Navier-Stokes equations, we impose no-slip boundary condition on the wall boundaries. This means both the velocity components vanish at the wall:

$$(u, v) = 0. \quad (2.31)$$

The boundary function w^b then becomes

$$w^b = (\rho, 0, 0, E)^T, \quad (2.32)$$

with ρ and E values taken from inside the element. A boundary condition for the temperature needs to be specified as well. In the present work we use the adiabatic condition, i.e.

$$\nabla T \cdot n = 0. \quad (2.33)$$

Combining both no-slip and adiabatic wall boundary conditions, the viscous flux at the boundary can be written as

$$f_v^b = \begin{pmatrix} 0 & \tau_{11} & \tau_{21} & 0 \\ 0 & \tau_{12} & \tau_{22} & 0 \end{pmatrix}^T. \quad (2.34)$$

3 Numerical Discretization

3.1 Introduction

To discretize the equations presented in Chapter 2, the hybridized discontinuous Galerkin (HDG) method is used [63, 64, 67, 86]. HDG is derived from the standard DG method, but uses hybridization as a technique to rewrite the DG equations in terms of the trace of the solution on the edges. An additional unknown, say λ , approximating the trace of the solution is introduced to the DG framework, along with an additional equation to (weakly) satisfy the continuity in the normal components of the fluxes. The advantage is that, now the globally coupled degrees of freedom are that of the new variable λ which lies on a space with one dimension lower than that of the solution variables. While it inherits all the advantages of the DG methods - high order of accuracy on unstructured grids, local conservation, etc - it has the additional advantage that it has significantly smaller global system to solve at each step for iterative type of solvers. Like the standard DG methods, due to local polynomial approximation of the solution, the HDG methods also offer complete flexibility for *hp*-adaptivity. For compressible aerodynamic flows which exhibit diverse flow features such as shocks, *hp*-adaptivity is expected to give the best results.

In this chapter, first, a brief review and the mathematical formulation of the discontinuous Galerkin method for general nonlinear convection-diffusion equations are given. This is followed by the derivation of the HDG method. Since we are considering only steady-state problems, the derivations are also given for the same. It is emphasized that the development of DG/HDG methods were not part of the present work and their derivations are provided only for the sake of completeness.

3.2 Discontinuous Galerkin Methods

The most popular high order method used to solve convection dominated flows is arguably the discontinuous Galerkin (DG) method, which combines the advantages of both Finite Volume and Finite Element methods. DG method uses locally discontin-

uous polynomials to represent the solution field inside a mesh element. Arbitrarily high formal order of accuracy can be obtained by appropriately choosing the polynomial degree. The discontinuous nature of the approximation space makes the method highly parallelizable. Furthermore, the mass matrix is block-diagonal and can be inverted cheaply. The DG method can also handle complex geometries and the associated boundary conditions without compromising the high-order accuracy. Another important advantage of the DG method is the ease at which one can perform adaptivity - both h - and hp -adaptivity. h -adaptivity is easier since refinement and coarsening can be done without taking into account the continuity restrictions like in the case of continuous Finite Element methods. For hp -adaptivity, appropriate element-wise polynomial degrees can be chosen easily, without any additional requirement.

The DG method was first introduced by Reed and Hill for the numerical treatment of the neutron transport equation [70]. A major step in the development of DG methods was due to the works of Cockburn and Shu who, in a series of papers [20, 23, 22, 25], developed TV-stable DG methods by using DG space approximation with limiting techniques and TV-stable high order Runge-Kutta explicit schemes for time discretization, for the numerical simulation of time-dependent nonlinear conservation laws. Thus they introduced the ideas from the Finite Volume - numerical flux, slope limiter etc, into the Finite Element framework to produce high order accurate and nonlinearly stable schemes.

Researchers had also been trying independently to solve elliptic and parabolic problems using discontinuous Finite Element spaces. These were generally called Interior Penalty (IP) methods. Penalties were first introduced for imposing Dirichlet boundary conditions weakly rather than incorporating them into the Finite Element space. Later they observed that the inter-element continuity could be attained in a similar fashion and that this makes it possible to use discontinuous spaces for elliptic problems. Douglas and Dupont have analysed such a method which uses interior penalty to enforce continuity for linear elliptic and parabolic problems [31]. The interior penalty method for non-linear parabolic problems is analyzed by Arnold [6]. We refer to the paper of Arnold et al. [27] for an extensive review of DG methods for parabolic and elliptic problems.

In 1996, Bassi and Rebay solved the compressible Euler equations with DG method [14]. They have also extended the DG method to the compressible Navier-Stokes equations [13], where the viscous term was treated with a mixed formulation — gradient of the solution as an additional auxiliary unknown variable. Cockburn and Shu, generalized the approach of Bassi and Rebay, and called the method Local Discontinuous Galerkin (LDG) method [24]. They also proved stability and error estimates for the

method. LDG methods have the drawback of having non-compact stencil for viscous discretizations, and this was rectified by Peraire and Person in their method called the Compact Discontinuous Galerkin (CDG) method [68].

The rigorous analysis of DG method for the general nonlinear case seemed to be quite difficult, and not much results exist. In the one-dimensional scalar case with convex flux function, Jiang and Shu showed that if the approximate solution, for any degree of the approximation space, converges, then it converges to the entropy solution [53]. For the space-time DG method for scalar conservation laws in multidimension, Jaffre, Johnsson and Szepessy [52] showed that for arbitrary polynomial approximation degree on unstructured meshes, solution converges to the entropy solution if one includes a streamline diffusion and an artificial viscosity term in the formulation. In 1996, for the DG method in the scalar case in multidimension with shock capturing, Cockburn and Gremaud proved convergence to the entropy solution with a rate of at least $h^{1/8}$, in the $L^\infty(L^1)$ -norm [19].

For an extensive review on the evolution of DG methods, and its theoretical and algorithmic aspects in a wide variety of applications, we refer to the paper of Cockburn et al [21].

3.2.1 Formulation

We can rewrite general steady-state convection-diffusion equations as a first-order system by introducing an additional unknown representing the gradient of the solution, i.e.

$$\begin{aligned} q &= \nabla w \\ \nabla \cdot (f_c(w) - f_v(w, q)) &= s(w, q). \end{aligned} \tag{3.1}$$

We tessellate the domain Ω into a collection of non-overlapping elements, denoted as $\mathcal{T}_h$, such that $\bigcup_{\Omega_k \in \mathcal{T}_h} \overline{\Omega}_k = \overline{\Omega}$. We define the discontinuous Finite Element spaces to look for the Finite Element approximated solution $w_h \in V_h$ and for the gradient $q_h \in W_h$ as

$$\begin{aligned} V_h &:= \{ \varphi_h \in L^2(\Omega) : \varphi_h|_{\Omega_k} \in P^p(\Omega_k) \} \\ W_h &:= \{ \tau_h \in L^2(\Omega) \times L^2(\Omega) : \tau_h|_{\Omega_k} \in P^p(\Omega_k) \times P^p(\Omega_k) \}, \end{aligned}$$

where P^p is the space of polynomials up to a total degree of p . To get the discontinuous Galerkin method, we first multiply Eq. (3.1) with test functions from the above spaces,

integrate by parts, and sum over all the elements. This yields

$$\begin{aligned} \sum_{\Omega_k \in \mathcal{T}_h} \left(\int_{\Omega_k} q_h \cdot \tau_h dx + \int_{\Omega_k} (\nabla \cdot \tau_h) w_h dx - \int_{\partial\Omega_k} (\tau_h \cdot n) w_h \right) ds &= 0, \quad \forall \tau_h \in W_h \\ \sum_{\Omega_k \in \mathcal{T}_h} \left(\int_{\Omega_k} -(f_c - f_v) \cdot \nabla \varphi_h dx + \int_{\partial\Omega_k} \varphi_h (f_c - f_v) \right) ds &= \sum_{\Omega_k \in \mathcal{T}_h} \int_{\Omega_k} s \varphi_h dx, \quad \forall \varphi_h \in V_h \end{aligned}$$

But now, since w_h and q_h are discontinuous across elements, the quantities that are to be evaluated to get the surface integrals - w_h , $\widehat{f}_c$ and $\widehat{f}_v$ - are not uniquely defined on the element edges. So they have to be replaced by the numerical fluxes $\widehat{w}_h$, $\widehat{f}_c$ and $\widehat{f}_v$, which are approximations of w_h , $f_c \cdot n$ and $f_v \cdot n$, respectively on the edges. On the boundary of the domain, they are replaced by the boundary functions w^b , f_c^b and f_v^b , that have to be determined from the desired boundary conditions and the characteristics. Now, with these, we get the discontinuous Galerkin discretization in the mixed formulation. Find $(q_h, w_h) \in W_h \times V_h$ s.t. $\forall (\tau_h, \varphi_h) \in W_h \times V_h$

$$\begin{aligned} \sum_{\Omega_k \in \mathcal{T}_h} \left(\int_{\Omega_k} q_h \cdot \tau_h dx + \int_{\Omega_k} (\nabla \cdot \tau_h) w_h dx - \int_{\partial\Omega_k \setminus \partial\Omega} (\tau_h \cdot n) \widehat{w}_h ds - \int_{\partial\Omega_k \cap \partial\Omega} (\tau_h \cdot n) w^b ds \right) \\ = 0 \\ \sum_{\Omega_k \in \mathcal{T}_h} \left(\int_{\Omega_k} -(f_c - f_v) \cdot \nabla \varphi_h dx + \int_{\partial\Omega_k \setminus \partial\Omega} \varphi_h (\widehat{f}_c - \widehat{f}_v) ds + \int_{\partial\Omega_k \cap \partial\Omega} \varphi_h (f_c^b - f_v^b) ds \right) \\ = \sum_{\Omega_k \in \mathcal{T}_h} \int_{\Omega_k} s \varphi_h dx. \end{aligned} \quad (3.2)$$

The numerical trace and the numerical fluxes should be chosen in such a way that the resulting scheme is stable and consistent. For the convective numerical flux $\widehat{f}_c$, one can use the Lax-Friedrich flux defined as

$$\widehat{f}_c = \{f_c(w_h) \cdot n\} - \frac{1}{2} \alpha \llbracket w_h n \rrbracket, \quad (3.3)$$

where α is the maximum eigen value of $\frac{\partial f_c(w)}{\partial w} \cdot n$. And for $\widehat{w}$ and $\widehat{f}_v$, one can use the LDG proposed by Cockburn [24] as given below.

$$\widehat{w} = \{w_h\} - C_{12} \cdot \llbracket w_h n \rrbracket \quad (3.4)$$

$$\widehat{f}_v = \{f_v(w_h, q_h) \cdot n\} + C_{11} \llbracket w_h n \rrbracket + C_{12} \llbracket f_v(w_h, q_h) \cdot n, \rrbracket \quad (3.5)$$

where, $\{ \}$ and $\llbracket \rrbracket$ denote the average and jump operators across the element interface. C_{11} is a positive constant and C_{12} is a vector which is to be determined for each interior face. The LDG method has the drawback that the scheme is non-compact. This is rectified with the compact discontinuous Galerkin (CDG) scheme proposed by Peraire and Persson [68]. Here, the LDG scheme is modified with "edge" fluxes q_h^e instead of q_h in finding viscous numerical flux in Eq. (3.5). Now, the equation becomes

$$\widehat{f}_v = \{f_v(w_h, q_h^e) \cdot n\} + C_{11} \llbracket w_h n \rrbracket + C_{12} \llbracket f_v(w_h, q_h^e) \cdot n \rrbracket. \quad (3.6)$$

The "edge" fluxes q_h^e are computed by solving the equation

$$\int_{\Omega_k} q_h^e \cdot \tau_h dx = - \int_{\Omega_k} w_h \nabla \cdot \tau_h dx + \int_{\partial\Omega_k} \widehat{w}_h \tau_h \cdot n ds \quad \forall \tau_h \in W_h. \quad (3.7)$$

It is also possible to discretize the equations in such a way that one gets a system in w_h only, using lifting operations. This is done in a popular method called the BR2 discretization of Bassy and Rebay [5]. Here the approximation for the gradient, q_h , is rewritten in terms of the lifting operator as

$$q_h|_{\Omega_k} = \nabla w_h + R(\llbracket w_h \rrbracket) \quad (3.8)$$

$$q_h|_{\partial\Omega_k} = \nabla w_h + \eta_e R_e(\llbracket w_h \rrbracket), \quad (3.9)$$

where R and R_e are global and local lifting operators, respectively. For an edge e , R_e is defined as

$$\int_{\Omega_h} R_e(\llbracket w_h \rrbracket) \cdot \tau_h dx = - \int_e \{\tau_h\} \cdot \llbracket w_h \rrbracket ds \quad (3.10)$$

and

$$R(\llbracket w_h \rrbracket) = \sum_e R_e(\llbracket w_h \rrbracket). \quad (3.11)$$

Using the BR2 scheme, we get

$$\begin{aligned} \sum_{\Omega_k \in \mathcal{T}_h} \left(\int_{\Omega_k} -(f_c - f_v) \cdot \nabla \varphi_h dx + \int_{\partial\Omega_k \setminus \partial\Omega} \varphi_h (\widehat{f}_c - \widehat{f}_v) ds + \int_{\partial\Omega_k \cap \partial\Omega} \varphi_h (f_c^b - f_v^b) ds \right) \\ = \sum_{\Omega_k \in \mathcal{T}_h} \int_{\Omega_k} s \varphi_h dx \end{aligned} \quad (3.12)$$

where,

$$f_v = f_v(w_h, \nabla w_h + R(\llbracket w_h \rrbracket)) \quad (3.13)$$

$$\widehat{f}_v = \{f_v(w_h, \nabla w_h + \eta_e R_e(\llbracket w_h \rrbracket)) \cdot n\} \quad (3.14)$$

$$s = s(w_h, \nabla w_h + \eta_e R_e(\llbracket w_h \rrbracket)). \quad (3.15)$$

The DG methods have become quite popular for convection-dominated flows over the last decade. They give high order of accuracy on unstructured meshes and allows one to choose local polynomial degrees which is quite helpful for *hp*-adaptivity. However, this flexibility for DG methods comes at the cost of increased degrees of freedom compared to continuous Galerkin methods. The extra degrees of freedom are due to relaxing the continuity constraint which is there in the continuous Galerkin methods. To overcome this deficiency of having too many degrees of freedom for the DG methods, a technique called hybridization has been developed. The resulting method is called the Hybridized Discontinuous Galerkin (HDG) method. With hybridization we can reduce the size of the globally coupled system that has to be solved and this helps to get faster solution convergence. In addition to inheriting the geometric flexibility and the arbitrary high order accuracy of Discontinuous Galerkin methods, the HDG method offers a significant reduction in the computational cost due to the lower size of the system matrix. Woopen et.al [85] have compared the performance of the standard DG method and the HDG method for the target-based adaptivity for steady-state compressible flow simulations and have found that the HDG methods perform better than the DG methods in terms of runtime and memory requirements from polynomial approximation degree $p = 2$ for a broad range of test cases. We have used the HDG scheme for all our simulations - both isotropic and anisotropic adaptations. In the next section, the HDG formulation is discussed.

3.3 Hybridized Discontinuous Galerkin Methods

For the DG method, we have seen that the numerical trace $\widehat{w}$ should be defined uniquely on the edges using the solution w_h from the elements on the either sides of the edges. In the Hybridized DG method, one treats $\widehat{w}$ as a separate unknown, say λ , which lies only on the mesh edges. But additionally, we need to ensure the continuity of the numerical fluxes in a weak sense. This allows the solution variables defining the discontinuous Galerkin solution to be locally condensed element-wise, resulting in a system which involves only the degrees of freedom of the approximate traces of the

solution. In addition to the spaces defined in Eq. (3.2), a Finite Element space for the additional variable λ needs to be defined. The space for λ is given as

$$M_h := \{\mu_h \in L^2(\Gamma_h) : \mu_h|_e \in P^p(e), e \in \Gamma_h\}, \quad (3.16)$$

where Γ_h is the set of all the interior edges, e , of the triangulation.

By including the continuity constraint for the numerical flux (normal component of the convective and the viscous flux) in the weak sense, we now get the Hybridized Discontinuous Galerkin formulation as, find $(q_h, w_h, \lambda_h) \in X_h := (W_h, V_h, M_h)$ s.t. $\forall (\tau_h, \varphi_h, \mu_h) \in X_h$

$$\begin{aligned} \sum_{\Omega_k \in \mathcal{T}_h} \left(\int_{\Omega_k} q_h \cdot \tau_h dx + \int_{\Omega_k} (\nabla \cdot \tau_h) w_h dx - \int_{\partial\Omega_k \setminus \partial\Omega} (\tau_h \cdot n) \lambda_h ds \right. \\ \left. - \int_{\partial\Omega_k \cap \partial\Omega} (\tau_h \cdot n) w^b ds \right) = 0 \end{aligned} \quad (3.17)$$

$$\begin{aligned} \sum_{\Omega_k \in \mathcal{T}_h} \left(\int_{\Omega_k} -(f_c - f_v) \cdot \nabla \varphi_h dx + \int_{\partial\Omega_k \setminus \partial\Omega} \varphi_h (\hat{f}_c - \hat{f}_v) ds + \int_{\partial\Omega_k \cap \partial\Omega} \varphi_h (f_c^b - f_v^b) ds \right. \\ \left. - \int_{\Omega_k} s \varphi_h dx \right) = 0 \end{aligned} \quad (3.18)$$

$$\sum_{\Omega_k \in \mathcal{T}_h} \int_{\partial\Omega_k \setminus \partial\Omega} (\hat{f}_c - \hat{f}_v) \mu_h ds = 0. \quad (3.19)$$

The last equation (Eq. (3.19)) ensures that the numerical fluxes across the edges are continuous in the weak sense. Note that λ_h does not occur in the boundary terms as it is only defined on interior edges. The boundary conditions should be incorporated in such a way that the scheme is so-called adjoint-consistent [72]. The convective numerical flux $\hat{f}_c$ is chosen to be the one similar to the Lax-Friedrich flux and viscous numerical flux $\hat{f}_v$ is similar to the LDG flux, i.e.

$$\begin{aligned} \hat{f}_c(\lambda_h, w_h) &= f_c(\lambda_h) \cdot n - \alpha_c(\lambda_h - w_h) \\ \hat{f}_v(\lambda_h, w_h, q_h) &= f_v(\lambda_h, q_h) \cdot n + \alpha_v(\lambda_h - w_h). \end{aligned}$$

These can be grouped together with a new scalar constant, $\alpha = \alpha_c + \alpha_v$, which seems

to be sufficient for a wide range of test cases.

$$\widehat{f}_c(\lambda_h, w_h) - \widehat{f}_v(\lambda_h, w_h, q_h) = (f_c(\lambda_h) - f_v(\lambda_h, q_h)) \cdot n - \alpha(\lambda_h - w_h).$$

The first two HDG equations (Eqns. (3.17) and (3.18)) are called the local solvers. This is because the solution w_h and q_h on each element, according to these equations, are not coupled to the solutions w_h and q_h on neighboring elements, but only on the trace variable λ_h that lies on the edges. The significance is that, using the local solvers, we can eliminate w_h and q_h from the last equation and solve only for the trace variable λ_h . Thus the hybridization formulation allows us to get a smaller system to solve at the end. Once λ_h is known, one can solve for w_h and q_h using local solvers element-wise. The procedure will be explained more in the section on Hybridization (Section 3.3.2). Adding the three equations, Eqns. (3.17), (3.18), and (3.19), and defining $\mathbf{x}_h := (q_h, w_h, \lambda_h)$ and $\mathbf{y}_h := (\tau_h, \varphi_h, \mu_h)$ we can rewrite them in a compact form

$$\mathcal{N}_h^{\text{HDG}}(\mathbf{x}_h; \mathbf{y}_h) = 0, \quad \forall \mathbf{y}_h \in X_h. \quad (3.20)$$

To handle discontinuities or shocks, we also need to incorporate a shock-capturing term to the equation. We have used the artificial diffusion term $\nabla \cdot (\epsilon \nabla w)$, that was added to the original PDE. The viscosity coefficient ϵ , taken as a constant on each element, is defined as

$$\epsilon = \frac{\epsilon_0 \tilde{h}_k^2}{|\Omega_k|} \int_{\Omega_k} \|\nabla \cdot f_c\|_{L^1} dx, \quad (3.21)$$

where $\tilde{h} := \frac{h}{p}$, is the effective mesh size, ϵ_0 is a user defined constant (we have taken it as 0.2), and $\|\cdot\|_{L^1}$, the L^1 norm is taken over the solution components. As mentioned, the term $\nabla \cdot (\epsilon \nabla w)$ is added to the PDE, but in the corresponding variational form we neglect the edge integral and use only the volume integral. Now, the artificial viscosity term in the variational form becomes

$$\mathcal{N}_{h,\text{sc}} = \sum_k \int_{\Omega_k} \epsilon \nabla w_h \cdot \nabla \phi_h dx. \quad (3.22)$$

With the artificial viscosity term, the final weak form is given as

$$\mathcal{N}_h(\mathbf{x}_h; \mathbf{y}_h) := \mathcal{N}_h^{\text{HDG}}(\mathbf{x}_h; \mathbf{y}_h) + \mathcal{N}_{h,\text{sc}}(q_h, w_h; \varphi_h) = 0, \quad \forall \mathbf{y}_h \in X_h. \quad (3.23)$$

Note that the artificial viscosity term will be switched on only for the transonic and the supersonic test cases, where we expect shocks. Since the viscosity coefficient ϵ depends on the strong form of the residual, the artificial viscosity term vanishes for

smooth solutions up to level of accuracy. Thus the artificial viscosity acts only on the non-smooth parts of the solution where the residuals are large, and almost vanishing in smooth parts of the solution.

3.3.1 Relaxation

Eq. (3.23), in general, forms a nonlinear system of equations. We solve them using a modified version of the Newton-Raphson method. Here an initial guess $\mathbf{x}_h^0$ is provided and then a linearized system is solved to get the update for the solution $\delta\mathbf{x}$. This process of solving the linear system and updating the solution at the n -th step via the equation,

$$\mathbf{x}_h^{n+1} = \mathbf{x}_h^n + \delta\mathbf{x}_h^n, \quad (3.24)$$

is continued until the residual $\mathcal{N}_h(\mathbf{x}_h^n; \mathbf{y}_h)$ drops below a certain threshold. If the initial guess $\mathbf{x}_h^0$ is chosen sufficiently close to the solution $\mathbf{x}_h$, then the original Newton-Raphson method can be used where the update is obtained by

$$\mathcal{N}'_h[\mathbf{x}_h^n](\delta\mathbf{x}_h^n; \mathbf{y}_h) = -\mathcal{N}_h(\mathbf{x}_h^n; \mathbf{y}_h), \quad \forall \mathbf{y}_h \in X_h, \quad (3.25)$$

where $\mathcal{N}'_h$ denotes the Fréchet derivative of $\mathcal{N}_h$ with respect to $\mathbf{x}_h$. However, in most cases the initial guess is not close enough to the solution, so that a modified version of Newton's method - called as Damped Newton's method, has been used. Now, the equation system becomes

$$\mathcal{N}'_h[\mathbf{x}_h^n](\delta\mathbf{x}_h^n; \mathbf{y}_h) + \frac{1}{\Delta t^n} \text{Id} \delta\mathbf{x}_h^n = -\mathcal{N}_h(\mathbf{x}_h^n; \mathbf{y}_h), \quad \forall \mathbf{y}_h \in X_h. \quad (3.26)$$

Please note, that Newton's method is formally recovered for $\Delta t \rightarrow \infty$. Otherwise this method can be interpreted as implicit Euler method

$$\mathbf{x}_h^{n+1} = \mathbf{x}_h^n + \Delta t \mathcal{N}_h(\mathbf{x}_h^{n+1}; \mathbf{y}_h), \quad \forall \mathbf{y}_h \in X_h, \quad (3.27)$$

where one does not completely solve the nonlinear system of equations, but approximates the term $\mathcal{N}_h(\mathbf{x}_h^{n+1}; \mathbf{y}_h)$ by its first order Taylor expansion

$$\mathcal{N}_h(\mathbf{x}_h^{n+1}; \mathbf{y}_h) \approx \mathcal{N}_h(\mathbf{x}_h^n; \mathbf{y}_h) + \mathcal{N}'_h[\mathbf{x}_h^n](\mathbf{x}_h^{n+1} - \mathbf{x}_h^n; \mathbf{y}_h), \quad \forall \mathbf{y}_h \in X_h.$$

At the initial steps since the iterative solution $\mathbf{x}_n$ may be far from the actual solution $\mathbf{x}_h$, time steps of finite values are used. This phase is called the ramping phase. After the ramping phase, $\mathbf{x}_n$ is expected to be closer to $\mathbf{x}_h$, and we let the time step go

towards infinity to recover the Newton's method. We choose a sequence $(\Delta t^n)_n$ to consist of the ramping phase with n_0 number of steps, followed by a very fast increase. Since we are interested only in computing the steady-state solutions, we can use local time-stepping to accelerate the computation. On each element k , we define the time step,

$$\Delta t_k^n = \text{CFL}(n) \frac{|\Omega_k|}{\lambda_c + 4\lambda_v}, \quad (3.28)$$

where λ_c and λ_v represent approximations to the maximum eigenvalue of the convective and viscous flux, respectively [61]. The CFL number at the n -th step is defined as

$$\text{CFL}(n) = \begin{cases} c_0 \left(3 \left(\frac{n}{n_0} \right)^2 - 2 \left(\frac{n}{n_0} \right)^3 \right) & n \leq n_0 \\ \text{CFL}(n-1) \left(1 + c_1 \max \left(0, \log \left(\frac{\|\mathcal{N}_h^{n-1}\|_2}{\|\mathcal{N}_h^n\|_2} \right) \right) \right) & \text{otherwise} \end{cases} \quad (3.29)$$

where both c_0 and c_1 are usually around 100. Strictly speaking, these parameters depend on the actual problem; experiments, however, showed that they can be reliably tuned for a wide range of problems. For the simulations we performed, the Newton iterations are stopped when the L_2 norm of the residual is smaller than 10^{-10} . Details of the linear solver are given in the next section.

3.3.2 Hybridization

Eq. (3.26) is a large linear system of equations for the three unknown solution quantities $(\delta q_h, \delta w_h, \delta \lambda_h)$. But, since the local variables $\delta q_h, \delta w_h$ can be written in terms of $\delta \lambda_h$, one can reduce the global system of equations to the one on the hybrid variable $\delta \lambda_h$ as follows. First, using an appropriate polynomial expansion for $\delta q_h, \delta w_h$ and $\delta \lambda_h$, the linearized global system can be written in matrix form as

$$\begin{bmatrix} A & B & R \\ C & D & S \\ L & M & N \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \\ \delta \Lambda \end{bmatrix} = \begin{bmatrix} F \\ G \\ H \end{bmatrix} \quad (3.30)$$

where the vector $[\delta Q, \delta W, \delta \Lambda]^T$ contains the expansion coefficients of $\delta \mathbf{x}_h = (\delta q_h, \delta w_h, \delta \lambda_h)$ with respect to the chosen basis. Note that this vector is organized in such a manner that the degrees of freedom related to δq_h and δw_h are grouped together element-wise. Now we split Eq. (3.30) into

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix} - \begin{bmatrix} R \\ S \end{bmatrix} \delta \Lambda \quad (3.31)$$

and

$$\begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \end{bmatrix} + N\delta\Lambda = H. \quad (3.32)$$

Since, on each element, $(\delta q_h, \delta w_h)$ depend only on solution trace $\delta\lambda_h$, the matrices A, B, C , and D are block-diagonal and hence the first system Eq. (3.31) can be easily inverted element-wise to substitute for $[\delta Q, \delta W]^T$ in the second system (Eq. (3.32)). This substitution yields the hybridized system

$$\left(N - \begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} R \\ S \end{bmatrix} \right) \delta\Lambda = H - \begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} F \\ G \end{bmatrix} \quad (3.33)$$

This is now the global system that has to be solved at each relaxation step. The system depends only on $\delta\lambda$ and the size of the system is much smaller than that of the original system as given in Eq. (3.30). The linear system of equations (Eq. (3.33)) is solved with a restarted GMRES solver (30 search directions) with ILU preconditioning. The ILU depth is chosen to be 3. The systems are solved with a relative tolerance of 10^{-4} . We use the GMRES solver that is available through the PETSc library [11]. Once we solve for $\delta\Lambda$, we can solve for δQ and δW element-wise using Eq. (3.31).

4 Adjoint-Based hp -Adaptation on Isotropic Meshes

4.1 Introduction

In most engineering applications, one may not be interested in full flow details but rather in some specific functional quantities. In the case of external aerodynamics, these may be lift or drag coefficients of airplanes. Adjoint-based techniques can be used to estimate the error in such output functionals, and also to provide local indicators for adaptation. The adjoint is defined as the sensitivity of the output functional with respect to residual source perturbations. Using the adjoint, one can get an estimate for the spatial error distribution contributing to the error in the target functional. This estimate can be used as a criterion for local adaptation — either h -adaptation (mesh refinement) or for p -adaptation (polynomial space enrichment) — so that the error in the target functional is expected to decrease in the next iteration. This local adaptation is more efficient than a global refinement where even the cells that contribute little to the error in the target functional get refined thereby increasing computational cost unnecessarily. Becker and Rannacher [16] have given a thorough review of the adjoint-weighted residual method in finite element discretizations. Giles and Pierce [41] have dealt with finite element methods applied to PDEs relevant to CFD applications. An overview of the application of the discontinuous Galerkin method to target-based adaptation is given by Hartmann and Houston [46]. For aerospace CFD applications and various mesh adaptation techniques Fidkowski [37] has provided relevant work.

In adjoint-based h -adaptivity, the mesh is refined or coarsened based on the error estimate, keeping the polynomial order constant. In adjoint-based hp -adaptivity, both the mesh size and the local polynomial degree can be varied. When the flow is smooth, p -refinement is more efficient as it gives better reduction in the error per cost. Near to discontinuities, h -refinement is a better choice since p -refinement may introduce additional Gibbs oscillations. In methods which use discontinuous function spaces for the solution variable, like Discontinuous Galerkin (DG), HDG, etc., hp -adaptivity is particularly simple to implement because of the local nature of the polynomial ap-

proximation. Adjoint-based hp -adaptivity has already been found efficient in error reduction in the context of discontinuous Galerkin methods [45, 44, 80] and also in the CPR methods [74]. Schütz and May [72] and Woopen et al. [86] devised and implemented an adjoint-based mesh adaptation procedure based on a class of adjoint-consistent hybridized finite element methods. In the present work, hp -adaptivity is added to our hybridized finite element solver.

4.2 Adaptation Strategy

The adaptation methodology is based on an adjoint error estimator that gives a spatial error distribution based on a target functional. This helps identify the flow regions that need to be resolved more to reduce the error in the target functional. Having found the element-wise error distribution, the fixed mesh fraction strategy is used where the elements are first sorted with respect to the error (in absolute value), and a user defined mesh fraction f_h , typically in the range of 5-10 %, is used to flag the elements which belong to the top f_h fraction of the elements with the largest errors. A very low value of f_h would result in too many iterations to reach an error tolerance, and a very high value would result in excessive mesh refinement. Values in the range of 5-10 % were found to be a good compromise for the simulations considered. In the case of h -adaptation, the flagged elements are subdivided to form new elements. In the case of hp -adaptation, the flagged elements either undergo a h -refinement or a p -refinement depending on the solution regularity. The solution regularity is quantified using a smoothness sensor. Details about the sensor used herein and the criterion of choosing h - and p -refinement are given in Section 4.4. The h -adapted mesh along with the polynomial degree distribution in the case of hp -adaptation is then used for the next iteration. The process is repeated till the target functional is within the given error tolerance. Note that there is no coarsening done either for h -adaptations or for hp -adaptations presented in this Chapter. The adjoint-based error estimation is described briefly in the following.

4.3 Adjoint-Based Error Estimation

Consider a general semi-linear form in a variational framework : Find $\mathbf{x}_h \in X_h$ such that

$$\mathcal{N}_h(\mathbf{x}_h; \mathbf{y}_h) = 0, \quad \forall \mathbf{y}_h \in X_h. \quad (4.1)$$

Given a scalar target functional $J_h(\mathbf{x}_h)$, the associated adjoint function $\mathbf{z}_h$ is defined as the sensitivity of the target functional with respect to the perturbations in the residual $\mathcal{N}_h$, i.e.

$$\delta J_h = J_h(\mathbf{x}_h + \delta \mathbf{x}_h) - J_h(\mathbf{x}_h) = \delta \mathcal{N}_h(\mathbf{z}_h) \quad (4.2)$$

The infinitesimal state $\delta \mathbf{x}_h$ and the residual perturbation are related as

$$\mathcal{N}'_h[\mathbf{x}_h](\delta \mathbf{x}_h; \mathbf{y}_h) + \delta \mathcal{N}_h(\mathbf{y}_h) \approx 0, \quad (4.3)$$

where the prime denotes the Fréchet linearization with respect to the argument in the square bracket. Now when the output functional is linearized as

$$J'_h[\mathbf{x}_h](\delta \mathbf{x}_h) + \delta J_h \approx 0, \quad (4.4)$$

using Eq. (4.2), Eq (4.3) and Eq. (4.4), we get

$$\mathcal{N}'_h[\mathbf{x}_h](\delta \mathbf{x}_h; \mathbf{z}_h) = J'_h[\mathbf{x}_h](\delta \mathbf{x}_h).$$

In order for the above equation to be satisfied for any general perturbation, then the adjoint function $\mathbf{z}_h$ should satisfy the so-called adjoint equation,

$$\mathcal{N}'_h[\mathbf{x}_h](\mathbf{y}_h; \mathbf{z}_h) = J'_h[\mathbf{x}_h](\mathbf{y}_h), \quad \forall \mathbf{y}_h \in X_h. \quad (4.5)$$

For the derivation of the adjoint-based error estimate for a finite perturbation, the target functional and the residual are expanded in Taylor series as follows:

$$J_h(\mathbf{x}) - J_h(\mathbf{x}_h) = J'_h[\mathbf{x}_h](\mathbf{x} - \mathbf{x}_h) + O(\|\mathbf{x} - \mathbf{x}_h\|^2), \quad (4.6)$$

$$\mathcal{N}_h(\mathbf{x}; \mathbf{y}_h) - \mathcal{N}_h(\mathbf{x}_h; \mathbf{y}_h) = \mathcal{N}'_h[\mathbf{x}_h](\mathbf{x} - \mathbf{x}_h; \mathbf{y}_h) + O(\|\mathbf{x} - \mathbf{x}_h\|^2). \quad (4.7)$$

Now using the adjoint equation, Eq. (4.5), an estimate of the error in the target functional is obtained as

$$e_h = J_h(\mathbf{x}) - J_h(\mathbf{x}_h) \quad (4.8)$$

$$= J'_h[\mathbf{x}_h](\mathbf{x} - \mathbf{x}_h) + O(\|\mathbf{x} - \mathbf{x}_h\|^2) \quad (4.9)$$

$$= \mathcal{N}'_h[\mathbf{x}_h](\mathbf{x} - \mathbf{x}_h; \mathbf{z}_h) + O(\|\mathbf{x} - \mathbf{x}_h\|^2) \quad (4.10)$$

$$= \mathcal{N}_h(\mathbf{x}; \mathbf{z}_h) - \mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h) + O(\|\mathbf{x} - \mathbf{x}_h\|^2) \quad (4.11)$$

$$= -\mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h) + O(\|\mathbf{x} - \mathbf{x}_h\|^2). \quad (4.12)$$

Note that the term $\mathcal{N}_h(\mathbf{x}; \mathbf{y}_h)$ vanishes as the discretization is consistent. The error estimate is now approximated as

$$e_h \approx \eta := -\mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h), \quad (4.13)$$

which can then be restricted to a single element Ω_k to yield a local indicator for mesh refinement, i.e.

$$\eta_k := \left| \mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h) \right|_{\Omega_k} \quad (4.14)$$

so that $\eta \leq \sum_{k \in \mathcal{T}_h} \eta_k$ holds. The functionals $\mathcal{N}_h$ and J_h and their Jacobians have to be evaluated in a somewhat richer space than X_h , namely $\tilde{X}_h \supset X_h$. Otherwise the weighted residual $\mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h)$ would be identical zero since

$$\mathcal{N}_h(\mathbf{x}_h; \mathbf{y}_h) = 0 \quad \forall \mathbf{y}_h \in X_h. \quad (4.15)$$

Enriching the space can be achieved by either mesh refinement or a higher polynomial degree for the ansatz functions. In the current setting, especially when using a hierarchical basis, the latter is advantageous with respect to implementational effort and efficiency. With the richer space $\tilde{X}_h$, the adjoint equation is now rewritten as

$$\mathcal{N}'_h[\mathbf{x}_h](\mathbf{y}_h; \mathbf{z}_h) = J'_h[\mathbf{x}_h](\mathbf{y}_h) \quad \forall \mathbf{y}_h \in \tilde{X}_h. \quad (4.16)$$

An important property that has to be respected while using adjoint-based error estimation is the adjoint-consistency of the underlying numerical discretization. The discrete adjoint approach, where the adjoint is calculated using the discretization that was used to obtain the primal solution $\mathbf{x}_h$ is used here. The adjoint equations produced in this manner may not be consistent with the exact adjoint differential equations, unless the discretization of the primal problem is adjoint consistent. Adjoint consistency makes sure that the adjoint solution using the discrete adjoint procedure is a consistent approximation to the exact adjoint. If the exact primal solution $\mathbf{x} \in X$ satisfies a weak formulation of Eq. (3.1), i.e.

$$\mathcal{N}(\mathbf{x}; \mathbf{y}) = 0 \quad \forall \mathbf{y} \in X. \quad (4.17)$$

and the exact adjoint z satisfies

$$\mathcal{N}''[\mathbf{x}](\mathbf{y}, \mathbf{z}) = J'[\mathbf{x}](\mathbf{y}) \quad \forall \mathbf{y} \in X, \quad (4.18)$$

then the discrete semi-linear form $\mathcal{N}_h$ is said to be adjoint consistent if the exact adjoint

z satisfies

$$\mathcal{N}'_h[\mathbf{x}](\mathbf{y}, \mathbf{z}) = J'_h[\mathbf{x}](\mathbf{y}) \quad \forall \mathbf{y} \in X. \quad (4.19)$$

An adjoint inconsistent discretization can make the adjoint solution oscillatory, pollute the error estimate and can lead to wrong adaptations. It can also affect the convergence of both primal and adjoint solutions. Adjoint consistency has been proved for a class of Hybrid mixed schemes including the one we use here - the HDG method, by Schütz and May [72]. To make the scheme adjoint consistent, there could be restrictions on the form of the target functionals and on the interior and the boundary discretizations. However, normally, an adjoint inconsistent scheme can be made adjoint consistent by modifying the semi-linear form and the target functional [43].

4.4 Adjoint-Based *hp*-Adaptation

The aim is to perform *hp*-refinement judiciously such that the error in the target functional is reduced with the least computational cost. Increasing the polynomial order on smooth flow regions and refining the cells on non-smooth flow regions are expected to resolve the flow in the best way. The adjoint-based error estimator gives the spatial error distribution and the cells which contribute the most to the error are flagged for refinement. Among these cells, those which are further flagged by the smoothness sensor would undergo an *h*-refinement, whereas in the remaining cells the polynomial degree is increased. HDG schemes offer full flexibility like DG schemes to have different polynomial approximations locally, and also to refine the elements without any continuity restrictions. However, care must be taken when approximating the hybrid variable, λ , which sits on the edges of the mesh elements. When there are neighbouring cells having different polynomial approximations, due to stability reasons, the hybrid variable on the edge connecting those cells has to be approximated with the maximum of the two degrees.

There are various ways of quantifying smoothness of a solution and hence the associated smoothness sensors. One of the popular sensors used both in shock detection and for *hp*-adaptivity is the smoothness sensor based on the decay of the expansion coefficients [80, 69]. The key idea is that, for smooth flows the expansion coefficients will have a similar decay rate as the Fourier coefficients, whereas for non-smooth solutions the decay rates get deteriorated. This can be exploited to create a local smoothness sensor. Although it works very well, we have found that we cannot use a unique criteria to choose between *h*- and *p*-refinement for different test cases [8]. This means,

for different test cases, we need to find the best criteria using trial and error method, which is a severe drawback. The smoothness sensor we use in the present work is the one proposed by Dolejsi [29] to choose between h - and p -refinement in the context of residual based hp -adaptivity using discontinuous Galerkin methods. This has the advantage that we can use a unique criteria valid for all the test cases we considered, without resorting to any trial and error method. The approach is based on a measure of the inter-element jumps in the solution. It has been shown in [28] for the case of scalar convection-diffusion equation that the following a priori error estimate is satisfied

$$\sum_k \int_{\partial\Omega_k} \llbracket w_h \rrbracket ds \leq C \sum_k h_k^{2\mu_k-1} |w|_{H^{s_k}(\Omega_k)}^2, \quad (4.20)$$

where $C > 0$ is a constant independent of h and $\mu_k = \min(p_k + 1, s_k)$, s_k is the local regularity of the solution w , i.e. $w|_k \in H^{s_k}(\Omega_k)$, the Sobolev space of index s_k . The above a priori error estimate implies that for a sufficiently regular flow, increasing the polynomial degree p_k is more efficient than decreasing the mesh size h_k in decreasing the error. For each element Ω_k , this is then written as

$$\int_{\partial\Omega_k} \llbracket w_h \rrbracket ds \approx C h_k^{2\mu_k-1} |w|_{H^{s_k}(\Omega_k)}^2. \quad (4.21)$$

Based on the above relation, the sensor is defined as follows

$$S_k = \frac{1}{|\Omega_k| h_k^{2p_k-2}} \int_{\partial\Omega_k} \llbracket w_h \rrbracket ds. \quad (4.22)$$

To arrive at a criterion to choose between h - and p -refinement based on this sensor, we look at the order of magnitude of the sensor for smooth and non-smooth flows. Since the mesh is quasi-uniform, the volume of triangular element Ω_k is $|\Omega_k| \approx \mathcal{O}(h_k^2)$. If the solution is sufficiently smooth, then $\mu_k = p_k + 1$, and

$$S_k \approx \mathcal{O}\left(\frac{h_k^{2p_k+1}}{h_k^2 h_k^{2p_k-2}}\right) = \mathcal{O}(h_k^1). \quad (4.23)$$

If the solution is not smooth enough, then $\mu_k = s_k$, and

$$S_k \approx \mathcal{O}\left(\frac{h_k^{2s_k-1}}{h_k^2 h_k^{2p_k-2}}\right) = \mathcal{O}(h_k^{2(s_k-p_k)-1}). \quad (4.24)$$

Since, for non-smooth flows $s_k < p_k$, the sensor S_k will take high values. If $S_k \leq 1$, the solution is assumed to be sufficiently regular, and p -refinement is performed. Otherwise,

h -refinement is done, i.e.

$$S_k \leq 1 \Rightarrow p\text{-refinement} \quad (4.25)$$

$$S_k > 1 \Rightarrow h\text{-refinement}. \quad (4.26)$$

A slightly modified version of the sensor is also given which, according to Dolejsi, performs better than the sensor S_k Eq. (4.22). The modified one is given as

$$\tilde{S}_k = \frac{1}{|\Omega_k| h^{2p_k-4}} \int_{\partial\Omega_k} \llbracket w_h \rrbracket ds \quad (4.27)$$

Based on numerical experiments, the sensor S_k is used if $p_k < 2$, and $\tilde{S}_k$ if otherwise.

4.5 Numerical Results

In this section, hp -adaptations are performed for various test cases and the results are compared with pure h -adaptations. The governing equations are discretized using the HDG method as discussed in Section 3.3. The discretized equations are solved using the damped Newton method (Section 3.3.1) and the resulting linear sub-problems are solved using a restarted GMRES with ILU(3) preconditioner. The same GMRES method is also used for the discrete adjoint problems. The Newton iterations are stopped when the L_2 norm of the residual is smaller than 10^{-10} . At first, a scalar convection-diffusion is considered, where there is a strong boundary layer on the top right of the domain. Then we solve the Euler and the Navier-Stokes equations over the NACA0012 airfoil. For the scalar test case, the number of Newton iterations required to attain the residual tolerance was around 2 – 4, and for the other cases it was in the range of 8 – 12. For h -adaptations, we use fixed fraction marking strategy, and those elements which are marked are subdivided to form child elements. For hp -adaptation, the marked elements either undergo a h -refinement or a p -refinement depending on the value given by the smoothness sensor. Both for h - and hp -adaptations, the meshes involved are isotropic, and there is no coarsening of the mesh.

4.5.1 Scalar convection-diffusion equation

The scalar convection-diffusion equation is considered:

$$\begin{aligned} \nabla \cdot (w, w) - \epsilon \Delta w &= s & (x, y) \in \Omega = [0, 1]^2 \\ w(x, y) &= 0 & (x, y) \in \partial\Omega. \end{aligned}$$

The solution is taken as

$$w(x, y) = \left(x + \frac{e^{x/\epsilon} - 1}{1 - e^{1/\epsilon}} \right) \cdot \left(y + \frac{e^{y/\epsilon} - 1}{1 - e^{1/\epsilon}} \right),$$

which forms a boundary layer for lower values of ϵ . Substituting this solution into the convection-diffusion equation gives a source function which is then used to solve the equation. We have chosen $\epsilon = 0.005$, in which case there is a strong boundary layer on the top right corner of the domain. The presence of such a strong gradient in the solution makes it a suitable candidate for hp -adaptation. Two cases are considered, in the first case a weighted total boundary flux is taken as the target functional, and in the second case the L^2 -norm of the solution error is taken as the target functional. A uniform mesh with 512 elements has been used as the initial mesh. In both cases, hp -adapted simulations are performed and the results are compared with those from pure h -adaptation.

4.5.1.1 Target functional - total weighted boundary flux

The target functional is taken as the weighted total boundary flux i.e.

$$J = \int_{\partial\Omega} \psi(w - \epsilon n \cdot \nabla w) d\sigma, \quad (4.28)$$

where the weighting function is $\psi = \cos(2\pi x) \cos(2\pi y)$. For hp -adaptation, a fixed fraction of 10% is used to mark the elements based on the adjoint-based error estimator. Among the marked elements, the smoothness sensor decides if an element would undergo mesh refinement or polynomial space enrichment. For the initial mesh, a polynomial degree of 2 has been used uniformly. Fig. 4.1 shows the solution contours both on the initial mesh (uniform) and on the (9th) hp -adapted mesh (isotropically refined). Fig. 4.2 shows the corresponding polynomial map of the hp -adapted mesh. The solution is well-resolved with hp -adaptivity. As expected, the boundary layer got both h - and p -refined and the region outside the layer got neither h -refined nor p -refined. Polynomial map near the boundary layer (top right corner of the domain) at different hp -adaptation steps are shown in Fig. 4.3. During the initial adaptation steps, the boundary layer region got h -refined and once the solution has been resolved well, the region got p -refined on further adaptations. Fig. 4.4 and table 4.1 compare the convergence of the error in the target functional with that of pure h -adaptations ($p = 2$ and $p = 3$). As expected, the hp -adaptation outperforms the h -adaptation in reducing the target error in terms of the number of degrees of freedom.

(a) Solution on initial mesh of 512 elements (b) Solution on hp -adapted mesh of 1724 elements

Figure 4.1: Scalar convection-diffusion test case, $\epsilon = 0.005$, hp -adaptation, total weighted boundary flux as target functional.

(a) Polynomial map on an hp -adapted mesh

Figure 4.2: Scalar convection-diffusion test case, $\epsilon = 0.005$, hp -adaptation, total weighted boundary flux as target functional, 9th adapted step with 1724 elements.

Figure 4.3: Scalar convection-diffusion test case, $\epsilon = 0.005$, hp -adaptation, total weighted boundary flux as target functional, polynomial map near the top right corner of the domain for different adaptation steps.

(a) Error in target functional vs. degrees of freedom

Figure 4.4: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of h - and hp -adaptivity, total weighted boundary flux as the target functional.

$p = 1$		$p = 2$		$p = 3$		hp	
ndof	$ J(w) - J(w_h) $	ndof	$ J(w) - J(w_h) $	ndof	$ J(w) - J(w_h) $	ndof	$ J(w) - J(w_h) $
1536	6.842752e-04	3072	2.334958e-05	5120	1.202522e-06	3072	2.334958e-05
2244	6.502062e-05	3864	9.659463e-06	7440	1.690367e-07	4030	1.055007e-05
3450	4.463755e-05	5124	5.414750e-07	11380	9.864631e-09	5272	1.770566e-07
5193	1.374114e-06	7434	2.959092e-07	18700	1.896111e-10	6356	2.004500e-08
8634	9.831071e-07	10296	9.420152e-08	30820	3.372015e-12	6919	4.694267e-08
13230	5.723201e-07	13344	1.041343e-08			7917	2.421866e-09
23268	3.192402e-07	18150	5.760613e-09			8628	3.722019e-09
38694	1.539292e-07	24774	2.376638e-09			10553	2.371469e-09
62895	1.733895e-08	33942	1.460914e-10			12488	4.645180e-10
						15213	5.519287e-11
						19060	6.535844e-12

Table 4.1: Scalar convection-diffusion test case, $\epsilon = 0.005$, total weighted boundary flux as target functional, h - and hp -adaptation results.

4.5.1.2 Target functional - L^2 norm of solution error

The target functional is taken as the L^2 -norm of solution error i.e.

$$J = \|w - w_h\|_{L^2} = \left(\int_{\Omega} (w - w_h)^2 dx \right)^{\frac{1}{2}}. \quad (4.29)$$

For hp -adaptation, a fixed fraction of 30% is used to mark the elements based on the adjoint-based error estimator. In this case, more aggressive refinement (h or p) is needed to reduce the target error to a low value. Fig. 4.5 shows the solution on the hp -adapted mesh (8th adapted step) and the corresponding polynomial map. Similar to the previous test case, the boundary layer first gets h -refined and once the layer is

resolved, it gets higher polynomial space enrichment. The figure also shows a magnified picture where higher polynomial approximations near the boundary layer can be seen. Fig. 4.6 and table 4.2 compare the convergence in the error for hp -adaptation and pure h -adaptation. With increasing polynomial degree approximations, the convergence rate gets better, and overall hp -adaptation performs the best in terms of number of degrees of freedom.

$p = 1$		$p = 2$		$p = 3$		hp	
ndof	$\ w - w_h\ _{L^2}$	ndof	$\ w - w_h\ _{L^2}$	ndof	$\ w - w_h\ _{L^2}$	ndof	$\ w - w_h\ _{L^2}$
1536	2.596999e-01	3072	9.512229e-02	5120	3.784952e-02	3072	9.512229e-02
2220	8.028988e-02	4452	2.227440e-02	7420	6.010695e-03	4460	2.228048e-02
3186	1.875204e-02	6384	3.788012e-03	10580	7.806488e-04	6382	4.065254e-03
4617	5.486098e-03	9438	1.035054e-03	16160	1.837911e-04	9064	8.507883e-04
7143	2.687164e-03	15678	3.371452e-04	27560	3.467194e-05	15304	1.444789e-04
11916	1.100552e-03	24936	1.006331e-04	42800	9.717754e-06	30235	2.935554e-05
18084	6.031232e-04	40098	4.625256e-05	72380	3.331454e-06	43828	7.582200e-06
29466	3.968404e-04	64524	2.074920e-05	113260	1.071605e-06	61641	1.704193e-06
43830	2.462242e-04	101418	9.366562e-06	181130	4.248541e-07	83533	4.980376e-07
68874	1.522312e-04	159546	4.629941e-06			128518	1.477030e-07
						205014	4.349079e-08
						326197	1.016255e-08
						489835	1.965720e-09

Table 4.2: Scalar convection-diffusion test case, $\epsilon = 0.005$, L^2 -norm of solution error as the target functional, h - and hp -adaptation results.

4.5.2 Inviscid, subsonic flow

The steady state Euler equations are solved around the NACA0012 airfoil with a free stream Mach number $Ma_\infty = 0.5$ and an angle of attack $\alpha = 2^\circ$. The drag coefficient, C_d , has been used as the target functional for the adjoint-based adaptations. hp -adaptive simulations were done using the adjoint-based estimator and the smoothness sensor. A fixed fraction of 10% is used to mark the elements for hp -adaptation. Fig. 4.7 shows the Mach number contours obtained on the initial mesh (2155 elements). Fig. 4.8 shows the Mach number contours around the airfoil, near the trailing and the leading edges on an hp -adapted mesh (2553 elements). Compared to the initial mesh, the hp -adapted mesh has h -refined regions near the trailing edge where there is a singularity. At the stagnation point near the leading edge, the elements get p -refined. The other regions around the airfoil, which are marked by the adjoint error estimate, were p -refined as well (Fig. 4.9). This is the reason why we get smoother contours around the airfoil compared to that on the initial mesh even though the mesh remains almost the same as that of the initial mesh. Fig. 4.10 and table 4.3 compare the convergence of the error in the drag coefficient for h - and hp -adaptation. In order to compute the error in the drag coefficient, a reference value of $C_{d,\text{ref}} = 4.686105 \times 10^{-6}$ is used, which was

Figure 4.5: Scalar convection-diffusion test case, $\epsilon = 0.005$, 8th hp -adapted step, L^2 -norm of solution error as target functional.

(a) Error in L^2 -norm vs. degrees of freedom

Figure 4.6: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of h - and hp -adaptation, L^2 -norm of solution error as the target functional.

obtained from the simulation using solution approximation degree of $p = 4$ on a mesh which was isotropically refined to have around 2.8×10^5 degrees of freedom. As seen in the figure, for higher p the convergence rates become better. hp -adaptation works quite well and outperforms h -adaptation in reducing the error in terms of number of degrees of freedom.

(a) Mach number contours on initial mesh (2155 elements)

Figure 4.7: Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, Mach contours on initial mesh.

(a) Mach number contours

(b) Smooth contours on leading edge due to p -refinement

(c) Mesh refined region near trailing edge

Figure 4.8: Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, Mach contours on hp adapted mesh (2553 elements), 9th adaptation step.

4.5.3 Inviscid, transonic flow

Inviscid transonic flow is computed over the NACA0012 airfoil with a free stream Mach number $\text{Ma}_\infty = 0.8$ and an angle of attack $\alpha = 1.25^\circ$. Target functional is the drag coefficient, C_d . A fixed fraction of 10% is used to mark the elements for hp -adaptation. Due to the presence of shocks, artificial viscosity term is switched on to stabilise the numerical scheme. Fig. 4.11 shows the Mach number contours around the airfoil and the polynomial degree distribution on the corresponding region at the 14th adaptation step. The upper and the lower shock region, the trailing and the leading edge got h -adapted at first. After getting h -refined and the solution is resolved to a certain

Figure 4.9: Inviscid subsonic flow, $Ma_\infty = 0.5$, $\alpha = 2^\circ$, hp -adaptation ($p = 1, \dots, 6$), 2553 elements, 9th adaptation step.

Figure 4.10: Inviscid subsonic flow around NACA 0012 airfoil, $Ma_\infty = 0.5$, $\alpha = 2^\circ$. Comparison of h - and hp -adaptation. For hp -adaptation, $p = 1, \dots, 6$.

level, they get p -refined. Fig. 4.12 and table 4.4 compare the convergence in the drag coefficient for the h - and hp -adaptation. To compute the error in the drag coefficient, a reference value of $C_{d,\text{ref}} = 2.265319 \times 10^{-2}$ obtained from the adapted mesh with around 7.6×10^4 degrees of freedom and solution approximation degree of $p = 2$ was used. Increasing the polynomial degrees in pure h -refinements shows no significant

$p = 1$		$p = 2$		hp	
ndof	$ C_d - C_{d,ref} $	ndof	$ C_d - C_{d,ref} $	ndof	$ C_d - C_{d,ref} $
6465	5.60401e-04	12930	5.99641e-05	6465	5.60402e-04
8376	2.61872e-04	16434	3.10772e-05	7113	2.41623e-04
10896	1.53879e-04	21636	1.43644e-05	7770	1.13112e-04
14511	9.48622e-05	28434	6.63281e-06	8668	6.29776e-05
19854	5.80417e-05	37086	3.07335e-06	9821	3.56347e-05
26469	3.52866e-05	48702	1.42405e-06	11157	1.98297e-05
35049	2.22988e-05	64638	6.55497e-07	12290	1.04927e-05
46761	1.40881e-05	85146	2.95740e-07	13595	5.68423e-06
62208	9.11151e-06			14882	3.02475e-06
				16504	1.60674e-06
				17900	8.36980e-07
				19280	4.27327e-07
				20762	2.07803e-07
				22242	9.26130e-08
				24196	3.51180e-08

Table 4.3: Inviscid subsonic flow, $M_\infty = 0.5$, $\alpha = 2^\circ$, h - and hp -adaptation results.

improvement in the convergence rate. This must be due to the strong shock on the upper side and the singularity on the trailing edge. However, hp -adaptation seems to work better overall compared to pure h -adaptation in terms of degrees of freedom.

Figure 4.11: Inviscid transonic flow, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, 14th adaptation step, 3552 elements.

4.5.4 Viscous, subsonic flow

As the final test case, viscous subsonic flow around the NACA0012 airfoil with a free stream Mach number $Ma_\infty = 0.5$, an angle of attack $\alpha = 1^\circ$ and a Reynolds number $Re = 5000$ is considered. Adaptations are again performed on the drag coefficient.

(a) Error in drag coefficient vs. degrees of freedom

 Figure 4.12: Inviscid transonic flow around the NACA0012 airfoil, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, comparison of h - and hp -adaptation, drag coefficient as the target functional.

$p = 1$		$p = 2$		hp	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	2.4545e-03	12930	1.0472e-03	6465	2.4543e-03
7329	2.0275e-03	14646	7.1941e-04	6789	1.8502e-03
8508	1.7376e-03	16818	5.0755e-04	7136	1.5308e-03
9966	1.4121e-03	19590	3.9122e-04	7883	1.0821e-03
11631	1.0541e-03	22458	2.9451e-04	9161	9.0369e-04
13734	7.8286e-04	26034	2.3233e-04	10870	7.4150e-04
16206	6.0869e-04	30552	1.6632e-04	12460	5.0659e-04
19074	4.6232e-04	35370	1.2069e-04	13848	4.0003e-04
22233	3.6892e-04	41784	8.3090e-05	15423	3.2277e-04
25845	2.7682e-04	48870	5.5030e-05	17230	2.0706e-04
30285	2.2027e-04			18729	1.5895e-04
35412	1.7503e-04			20412	1.3085e-04
				21781	8.1610e-05
				23902	7.1240e-05
				26264	5.1330e-05

 Table 4.4: Inviscid transonic flow, $M_\infty = 0.8$, $\alpha = 1.25^\circ$, h - and hp -adaptation results.

A fixed fraction of 10% is used to mark the elements for hp -adaptation. To compute the error in the drag coefficient, a reference value of $C_{d,\text{ref}} = 5.531683 \times 10^{-2}$ was taken from a fine adapted mesh with 5.1×10^5 degrees of freedom and the solution approximation degree $p = 4$. Fig 4.13 and Fig 4.14 show the Mach number contours around the airfoil and corresponding polynomial map at different adaptation steps. With adaptation steps, more and more regions around the airfoil get p -refined and the contours become smoother. Fig. 4.15 and table 4.5 show the convergence of the error in the drag coefficient against the number of degrees of freedom for both h - and hp -adaptation. With increasing polynomial degree, the convergence becomes better and

hp-adaptation outperforms pure *h*-adaptation.

Figure 4.13: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, *hp*-adaptation, drag coefficient as the target functional.

Figure 4.14: Viscous subsonic flow around the NACA0012 airfoil, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, hp -adaptation, drag coefficient as the target functional.

(a) Error in drag coefficient vs. degrees of freedom

Figure 4.15: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of h - and hp -adaptation. For hp -adaptation, $p = 1, \dots, 6$.

$p = 1$		$p = 2$		hp	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	7.3133e-03	12930	9.9055e-03	6465	7.3133e-03
8088	5.6375e-04	16188	3.9648e-04	7113	5.9789e-03
10434	1.0846e-03	21534	3.2206e-04	7863	9.9875e-04
14310	1.5075e-03	29826	1.5752e-04	9392	1.2526e-03
19773	6.6313e-04	41196	6.3990e-05	11104	8.7986e-04
26913	3.8112e-04	56592	4.1740e-05	12868	7.0765e-04
36948	2.6112e-04	77172	2.1470e-05	14624	4.0869e-04
50592	1.6325e-04			16633	2.6899e-04
68670	1.1841e-04			18648	1.7451e-04
				21831	9.7050e-05
				26322	6.2100e-05
				30018	4.7860e-05
				33088	3.6050e-05
				38440	2.7240e-05
				45416	1.6870e-05
				51683	1.1270e-05

Table 4.5: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, h - and hp -adaptation results.

5 Adjoint-Based hp -Adaptation on Anisotropic Meshes

5.1 Introduction

Aerodynamic flows typically exhibit highly anisotropic features like boundary layers and shocks. In order to resolve these solution features efficiently, the mesh elements should also possess anisotropy (stretching in some direction) to align with the flow features. Anisotropic mesh elements along with adjoint-based error estimation would provide an efficient distribution of the degrees of freedom for accurate computation of solution-dependent target functionals. In the context of aerodynamic flow simulations using high order discontinuous Galerkin methods, Leicht and Hartmann [56] have demonstrated the potential of adjoint based error estimator and anisotropic mesh refinement. In their method, the mesh elements are split in certain directions, given by anisotropy indicators, to achieve the required anisotropy along flow features. A monograph by Hartmann [46] provides a detailed theoretical formulation of adjoint-based error estimation and adaptive mesh refinement for aerodynamic flows using discontinuous Galerkin methods. Apart from splitting the mesh in the required direction, another way of generating anisotropic simplex mesh elements, which has been becoming popular recently, is by using the concept of mesh-metric duality [87, 30, 4, 35, 3], where metric tensors encode information about the mesh elements. In the adaptation procedure, metric tensors for the desired mesh are evaluated using the solution from the current simulation, and these metric tensors are then passed to a metric-conforming mesh generator to generate the required anisotropic mesh. The main advantage of using metric-based mesh adaptation is the flexibility it provides in manipulating the size and the shape of mesh elements. Metric-based mesh generation, along with an adjoint-based error estimator, was first devised by Venditti and Darmofal [78]. They combined an adjoint-based error estimate and a Hessian-based method for the metric calculation to a Finite Volume scheme with piecewise linear reconstruction. This was extended to high order schemes by Fidkowski [34]. The basic idea is to construct a mesh in which the area of each element is determined by the adjoint error estimator and the anisotropy is determined by requiring that the interpolation errors be the same

along each edge of a given mesh element. Loseille et. al [59] has demonstrated the potential of metric based anisotropic mesh adaptation using several test cases in three dimensions. An overview of adjoint-based adaptation on anisotropic meshes for CFD applications in aerospace engineering is briefly given by Fidkowski and Darmofal [38]. In addition to pure mesh adaptation (h -adaptivity), one can also vary the polynomial degree on each element (hp -adaptivity). hp -adaptivity has been proven to be quite efficient since the error converges at an exponential rate in the number of degrees of freedom for a smooth enough solution [50, 49]. Consequently, in smooth regions of the flow, p -refinement is likely to be more efficient, and in region of discontinuities, h -refinement is expected to give better results. Leicht and Hartmann [57] have shown the potential of adjoint-based hp -adaptation using several numerical examples. In their approach, a local criterion is used to choose between h -subdivision and p -enrichment combined with the adjoint error estimator to yield an hp -adaptive algorithm. Ceze and Fidkowski [17] uses a selection process, based on a merit function, to choose between the h -refinement and p -enrichment. The choices include elements subdivided in different directions, and elements with increased polynomial order.

The present work aimed to incorporate adjoint based error estimator into the metric-based hp -adaptation methodology proposed by Dolejsi [30], which uses an interpolation error model in determining both the size and the shape of mesh elements, and also the polynomial degree. In the presented method, the area of the triangular mesh elements is set using the adjoint error estimate, and then we seek the mesh element and the polynomial degree which gives the smallest interpolation error bound in the L^q -norm ($q \in [1, \infty)$). The work also includes a numerical verification of the interpolation error model, presented in [30], which forms the basis of our adaptation algorithm.

The rest of the chapter is organized as follows. In Section 5.2 we describe how pure h -adaptation on anisotropic meshes is done. This includes a short introduction on Riemannian metric and metric-based adaptation, finding the area of the mesh elements using adjoint-based error estimator, and finding the anisotropy of the mesh using directional derivatives. In the following section (Section 5.3), we describe the algorithm for hp -adaptation on anisotropic meshes. This is followed by the section (Section 5.4) where we numerically verify the interpolation error estimation which forms the basis of the adaptation algorithm. The last section (Section 5.5) is for the numerical experiments, where we show a scalar case, and flow cases - inviscid and viscous, to demonstrate the efficiency of the adaptation method.

5.2 Anisotropic h -Adaptation

The methodology is an extension of the mesh adaptation strategy of Dolejsi [30], incorporating the adjoint-based error estimate in finding the desired area of the mesh element. The original method of Dolejsi tries to generate anisotropic meshes such that the interpolation error in the L^q -norm ($q \in [1, \infty)$) is under a given tolerance, and the number of degrees of freedom is the smallest possible. Equivalently, one can also set the area of the element and find the anisotropy which gives the minimum interpolation error in the L^q -norm.

The fundamental concept used in the mesh adaptation process is the mesh-metric duality [58]. Given a triangulation $\mathcal{T}_h = \{\Omega_k\}$, each of its elements can be characterized by a 2×2 (in 2D) symmetric, positive definite matrix. For an element Ω_k , we call such a matrix as its implied metric $\mathcal{M}_k$ given as,

$$\mathcal{M}_k = \begin{pmatrix} a & b \\ b & c \end{pmatrix}. \quad (5.1)$$

The implied metric $\mathcal{M}_k$ is constructed in such a way that each of the three edges, $e_j, j = 1, 2, 3$, of the triangular element Ω_k has a constant length $C > 0$ with respect to the metric, i.e.

$$\sqrt{e_j^T \mathcal{M}_k e_j} = C, \quad j = 1, 2, 3. \quad (5.2)$$

The above equation (Eq. (5.2)) represents a linear system for the elements a, b and c of the metric $\mathcal{M}_k$. The triangle which is defined by the metric $\mathcal{M}_k$ can be inscribed in an ellipse, as shown in Fig. 5.1, with the origin as the centroid of the triangle, h_1 and h_2 as the principal lengths and θ as the orientation. These quantities h_1, h_2 and θ can be extracted from the metric $\mathcal{M}_k$ by its spectral decomposition given as

$$\mathcal{M}_k = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}^T \begin{pmatrix} \frac{1}{h_1^2} & 0 \\ 0 & \frac{1}{h_2^2} \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}. \quad (5.3)$$

Hence the eigenvectors give the direction of the principal axes of the ellipse, and the eigenvalues give the inverse of the square of the principal lengths h_1 and h_2 . For $C = \sqrt{3}$, the area of the inscribed triangle can be found as

$$I = \frac{3\sqrt{3}}{4} h_1 h_2. \quad (5.4)$$

Figure 5.1: Ellipse which encloses the triangular mesh element

One can also define the aspect ratio, $\beta > 1$, of the mesh element as

$$\beta := h_2/h_1. \quad (5.5)$$

Now the set of three quantities (I, β, θ) uniquely defines the metric and hence the mesh element Ω_k . The aspect ratio β and the orientation θ together is defined as the anisotropy of the element, and I is the area of the element. In our adaptation methodology, the implied metric of each mesh element of the current mesh is evaluated and is updated to get a new metric which corresponds to the desired mesh element. Finding the new metric consists of finding the desired size I and the desired anisotropy (β, θ) for each element. These are explained in Section 5.2.2 and Section 5.2.4, respectively. Once the desired metric is evaluated at all the elements of the current mesh, a metric-based mesh generator can be used to create a mesh that conforms with the desired metric.

The metric explained so far is discrete, in the sense that it is only defined on each mesh element of the current mesh. A metric-based mesh generator operates on a continuous metric field $\mathcal{M}(x, y)$, which may be generated by a suitable interpolation of the discrete metric. In the continuous setting, the length of a segment $\vec{ab}$ with respect to the metric, $\mathcal{M}$, is computed using a straight line parameterization given as

$$l_{\mathcal{M}}(\vec{ab}) = \int_0^1 \sqrt{\vec{ab}^T \mathcal{M}(a + s\vec{ab}) \vec{ab}} \, ds. \quad (5.6)$$

In a metric-conforming triangulation, the mesh will have edges that satisfy

$$l_{\mathcal{M}}(e) \approx C \quad (5.7)$$

where e is any edge in the triangulation and C is the constant in Eq. 5.2. For a given metric, $\mathcal{M}$, the metric-conforming triangulation is not unique. However, they will have similar approximation properties since the edges have to meet the length constraints given by the metric. Usually, we have to give an initial mesh and the metric evaluated at each element or the vertices, as input to the mesh generator and it creates the new mesh which conforms with the given metric. We have used BAMG mesh generator [47] for which the metric should be evaluated at the vertices of the mesh.

5.2.1 Algorithm for h -adaptation

The adaptation step consists of the following.

Algorithm A

- (1) For each mesh element $\Omega_k \in \mathcal{T}_h$,
 - (a) Find the implied metric using Eq. (5.2).
 - (b) Determine the desired area I , using an adjoint-based error estimate and the current area, as explained in Section 5.2.2.
 - (c) Determine the desired anisotropy (β, θ) , using the concept of the minimization of interpolation error, as explained in Section 5.2.4.
 - (d) Find the desired metric $\mathcal{M}$ from the new triplet (I, β, θ) using Eqns. (5.3), (5.4), and (5.5).
- (2) The new metric $\mathcal{M}$ evaluated at all the elements of the current mesh, and the current mesh are used as the inputs to the mesh generator to produce the desired mesh, that conform to the given metric $\mathcal{M}$. This new mesh is used for the next computation.

This process is repeated till we get the mesh which keeps the target functional within the given error tolerance. Note that to find the desired anisotropy, we need to find the $(p + 1)$ -st derivative of the solution variable. Since the solution is only of degree p , we reconstruct the solution to polynomial degree $p + 1$. This is achieved using a patch reconstruction, as given in Section 5.2.3

5.2.2 Finding the area of the desired mesh elements

To reduce the error in the target functional, we need to ensure that the elements with high (adjoint-based) error estimate, are refined. Since our anisotropic adaptation

involves remeshing, those elements with low errors can be coarsened as well. For each element Ω_k , the desired area I_k can be found from the error estimate η_k (Eq. (4.14)) and the current area of the element I_k^c as given below :

$$I_k = \alpha_k I_k^c, \quad (5.8)$$

where α_k decides the factor by which the element Ω_k is refined or coarsened. Any value greater than one for α_k would mean coarsening and lower than one would mean refinement of the element Ω_k . The elements are first sorted in the descending order with respect to the error (in absolute value). Then a user-specified mesh fraction f_h is used to define the reference error level η_{ref} . A certain value of f_h signifies f_h fraction of elements have errors higher than the reference error η_{ref} . Next, those elements with errors higher than η_{ref} are refined and those with errors lower than η_{ref} are coarsened. The refinement or the coarsening factor is found in the following heuristic way:

$$\alpha_k = \begin{cases} ((r_{max} - 1)\xi_k^2 + 1)^{-1}, & \eta_k \geq \eta_{ref} \\ ((c_{max} - 1)\xi_k^2 + 1), & \eta_k < \eta_{ref} \end{cases} \quad (5.9)$$

where r_{max} and c_{max} are the user-specified maximum refinement and maximum coarsening factors, respectively, and ξ_k is based on a logarithmic scaling of the error estimate as

$$\xi_k = \begin{cases} \frac{\log_{10}(\eta_k) - \log_{10}(\eta_c)}{\log_{10}(\eta_{max}) - \log_{10}(\eta_c)}, & \eta_k \geq \eta_{ref} \\ \frac{\log_{10}(\eta_k) - \log_{10}(\eta_c)}{\log_{10}(\eta_{min}) - \log_{10}(\eta_c)}, & \eta_k < \eta_{ref}. \end{cases} \quad (5.10)$$

η_{max} and η_{min} are the maximum and the minimum of the error estimates η_k , i.e.

$$\eta_{max} = \max_{\Omega_k \in \mathcal{T}_h} \eta_k \quad (5.11)$$

$$\eta_{min} = \min_{\Omega_k \in \mathcal{T}_h} \eta_k. \quad (5.12)$$

Eq. (5.9) gives for α_k a quadratic fit between r_{max} and 1 for the case $\eta_k \geq \eta_{ref}$, and another quadratic fit between c_{max} and 1 for the case $\eta_k < \eta_{ref}$. In this way, the element with the highest error gets refined by a factor of r_{max} times, and the one with the lowest error gets coarsened by a factor of c_{max} times. All the intermediate elements between the one with the highest error and the one with the reference error get refined, and all those between the one with the lowest error and the reference error get coarsened, by the factors given by the quadratic fit, Eq. (5.9). In short, to evaluate the new area of the mesh elements we need to specify three parameters — the maximum refinement and

coarsening factors r_{max} , and c_{max} , respectively, and the mesh fraction f_h . Details on choosing values for these parameters for the simulations performed are given in Section 5.5. It is worth mentioning that the simulations are pretty robust with respect to these parameters. For the test cases we considered, we could observe that the convergence rates stay the same over a range of parameter values.

5.2.3 Patch reconstruction

To find the $(p + 1)$ -st derivative of the solution variable we need to reconstruct the solution from polynomial degree p to $p + 1$. For this we use an H^1 patch reconstruction in which a patch is defined around each element and the solution is reconstructed in this patch. Each element will have an associated patch, and the reconstructed solution that is restricted to that element. For hp -adaptivity, we also need to reconstruct the solution to degree $p + 2$. For each element Ω_k , the patch, defined as $D(\Omega_k)$, consists of the element and all the neighboring elements as given in Fig. 5.2. For the solution variable u , we define the reconstructed solution $\tilde{u} \in P^{(p+r)}(D(\Omega_k))$, $r = 1, 2$ as the solution to the equation,

$$(\tilde{u}, \phi)_{H^1(D(\Omega_k))} = (u, \phi)_{H^1(D(\Omega_k))} \quad \forall \phi \in P^{(p+r)}(D(\Omega_k)). \quad (5.13)$$

Fig. 5.3 compares the original solution and the reconstructed solutions for the boundary

Figure 5.2: Patch for the element k .

layer test case defined in Section 5.5.1. Here, the original solution which is in $p = 1$ is reconstructed to both $p + 1 = 2$ and $p + 2 = 3$ discretization space. We observe that the solution gets smoother with higher order reconstructions.

Figure 5.3: Scalar boundary layer case, $\epsilon = 0.1$, Patch reconstruction.

5.2.4 Finding the anisotropy of the desired mesh

We use the methodology proposed by Dolejsi [30] to find the anisotropy of the triangle. The basic concept is to find the anisotropy of the triangle, with a given area, which minimizes an interpolation error function in the L_q -norm, $q \in [1, \infty)$. We briefly reproduce here some of the essential details that are required to find the anisotropy of the triangle, as described in the paper of Dolejsi [30]. Those details include the definition of an interpolation error function and finding a set of parameters which can be used to derive an estimate for the interpolation error function. The set of parameters is called the anisotropy of the interpolation error function, and they are

in turn used to find the anisotropy of the triangle for which the interpolation error function is minimum in the L_q -norm. At first we define the directional derivative and the maximum directional derivative of a function u , which is needed to find the anisotropy of the interpolation error function of the same function u .

5.2.4.1 Directional derivative

For a sufficiently smooth function u , the k -th directional derivative of u along the direction $(\cos \phi, \sin \phi)$ is defined as

$$u^{(k,\phi)} = \sum_{l=0}^k \binom{k}{l} \frac{\partial^k u}{\partial x^l \partial y^{k-l}} (\cos \phi)^l (\sin \phi)^{k-l}. \quad (5.14)$$

We define $\tilde{u}^{(p+1,\tilde{\phi}_p)}$ as the maximum $(p+1)$ st directional derivative of the (reconstructed) anisotropy variable $\tilde{u}$, $\tilde{\phi}_p$ as the corresponding angle and $\tilde{u}_{\perp}^{(p+1,\tilde{\phi}_p)}$ as the value of the directional derivative perpendicular to $\tilde{\phi}_p$. We also define $\tilde{A}_p$ as the scaled maximum directional derivative of u , and $\tilde{\rho}_p$ as the ratio of the maximum derivative and the derivative along its perpendicular direction as given below:

$$\tilde{A}_p = \frac{1}{(p+1)!} \tilde{u}^{(p+1,\tilde{\phi}_p)}, \quad \tilde{\rho}_p = \frac{\tilde{u}^{(p+1,\tilde{\phi}_p)}}{\tilde{u}_{\perp}^{(p+1,\tilde{\phi}_p)}}. \quad (5.15)$$

Note that to get the high order directional derivatives using Eq. (5.14) in the physical space, we first evaluate the derivatives in the reference space, and then they are transformed into the physical space. To evaluate the derivatives in the reference space, we need the derivatives of the basis functions. The evaluation of high order derivatives of Dubiner polynomial functions in the reference space has been done analytically and is described in Appendix A.

5.2.4.2 Interpolation error function

Given a sufficiently smooth function u , a point $\bar{\mathbf{x}} = (\bar{x}, \bar{y}) \in \Omega$, and the polynomial degree, p , then a projector operator $\pi_{\bar{\mathbf{x}},p}$ is defined such that

$$\frac{\partial^k (\pi_{\bar{\mathbf{x}},p} u(\bar{\mathbf{x}}))}{\partial x^l \partial y^{k-l}} = \frac{\partial^k u(\bar{\mathbf{x}})}{\partial x^l \partial y^{k-l}} \quad \forall l = 0, \dots, k \quad \forall k = 0, \dots, p. \quad (5.16)$$

This means the projection $\pi_{\bar{\mathbf{x}},p}u$, and the function u have the same partial derivatives up to order p . Using Taylor series expansion around $\bar{\mathbf{x}}$, for any point $\mathbf{x} = (x, y)$,

$$u(\mathbf{x}) = \sum_{k=0}^{p+1} \frac{1}{k!} \left(\sum_{l=0}^k \binom{k}{l} \frac{\partial^k u(\bar{\mathbf{x}})}{\partial x^l \partial y^{k-l}} (x - \bar{x})^l (y - \bar{y})^{k-l} \right) + \mathcal{O}(|\mathbf{x} - \bar{\mathbf{x}}|^{p+2}). \quad (5.17)$$

Using the projection operator $\pi_{\bar{\mathbf{x}},p}$ defined in Eq. (5.16), the above equation can be rewritten as

$$u(\mathbf{x}) - \pi_{\bar{\mathbf{x}},p}u(\mathbf{x}) \approx E_{\bar{\mathbf{x}},p}(\mathbf{x}), \quad (5.18)$$

where $E_{\bar{\mathbf{x}},p}$ is the interpolation error function of degree p located at $\bar{\mathbf{x}}$, defined as

$$E_{\bar{\mathbf{x}},p}(\mathbf{x}) = \frac{1}{(p+1)!} \sum_{l=0}^{p+1} \binom{p+1}{l} \frac{\partial^{p+1} u(\bar{\mathbf{x}})}{\partial x^l \partial y^{p+1-l}} (x - \bar{x})^l (y - \bar{y})^{p+1-l}. \quad (5.19)$$

The aim is to obtain a triangle with a given area and having centroid at $\hat{\mathbf{x}}$, which has the least interpolation error $E_{\hat{\mathbf{x}},p}(\mathbf{x})$, in the L_q -norm $q \in [1, \infty)$. A set of parameters called the anisotropy of the interpolation error function needs to be defined as done in the next section.

5.2.4.3 Anisotropy of the interpolation error function

An estimate for the interpolation error function $E_{\hat{\mathbf{x}},p}(\mathbf{x})$ can be written in terms of three parameters (A_p, ρ_p, ϕ_p) as

$$|E_{\hat{\mathbf{x}},p}(\mathbf{x})| \leq A_p \left((\mathbf{x} - \hat{\mathbf{x}})^T Q_{\phi_p} D_{\rho_p} Q_{\phi_p}^T (\mathbf{x} - \hat{\mathbf{x}}) \right)^{\frac{p+1}{2}} \forall \mathbf{x} \in \Omega, \quad (5.20)$$

where $A_p > 0$, Q_{ϕ_p} is the rotation through angle ϕ_p and D_{ρ_p} is a diagonal matrix as given below:

$$Q_{\phi_p} = \begin{bmatrix} \cos \phi_p & -\sin \phi_p \\ \sin \phi_p & \cos \phi_p \end{bmatrix}, \quad D_{\rho_p} = \begin{bmatrix} 1 & 0 \\ 0 & \rho_p^{-\frac{2}{p+1}} \end{bmatrix}. \quad (5.21)$$

The triplet (A_p, ρ_p, ϕ_p) is called as the anisotropy of the interpolation error function $E_{\hat{\mathbf{x}},p}(\mathbf{x})$. They are to be defined in such a way that the estimate (Eq. (5.20)) is as sharp as possible. With the choice $(A_p, \rho_p, \phi_p) = (\tilde{A}_p, \tilde{\rho}_p, \tilde{\phi}_p)$, which are defined in Section 5.2.4.1, it can be proved that the equality in Eq. (5.20) holds for the special case of $p = 1$ [30]. But for $p > 1$, the bound is not guaranteed always. Provided that the exact partial derivatives are known, a procedure is proposed by Dolejsi in [30] to

modify the coefficients $(\tilde{A}_p, \tilde{\rho}_p, \tilde{\phi}_p)$ such that a sharp bound is obtained. In practice, since the exact derivatives are not known and numerical approximations are involved, using the coefficients $(\tilde{A}_p, \tilde{\rho}_p, \tilde{\phi}_p)$ directly without modification can often be acceptable. This will also be numerically demonstrated in Section 5.4.

5.2.4.4 Anisotropy of the desired triangular element

The aim is to obtain a triangle, with a given area I , which gives the least interpolation error in the L_q -norm $q \in [1, \infty)$. In other words, given the area I , the anisotropy (β, θ) of the triangle for which $\|E_{\tilde{\mathbf{x}},p}\|_{L^q(\Omega_k)}$ is minimum is to be found. Such an anisotropy (β, θ) for the triangle for which the interpolation error is minimum can be found using the anisotropy of the interpolation error function, (A_p, ρ_p, ϕ_p) , defined in the previous section. Strictly speaking the minimization is done over the ellipse Ω_E which encloses the triangle Ω_k . Since $\Omega_k \subset \Omega_E$, the bound $\|E_{\tilde{\mathbf{x}},p}\|_{L^q(\Omega_k)} \leq \|E_{\tilde{\mathbf{x}},p}\|_{L^q(\Omega_E)}$ holds. For a triangle Ω_k defined by (I, β, θ) , if (β, θ) is evaluated as,

$$\beta = \rho_p^{\frac{1}{p+1}}, \quad \theta = \phi_p - \pi/2, \quad (5.22)$$

then it can be proved that $\|E_{\tilde{\mathbf{x}},p}\|_{L^q(\Omega_E)}$ is the minimum [30]. Furthermore, the bound for the interpolation error function $E_{\tilde{\mathbf{x}},p}$ on element Ω_k , is given as follows:

$$\|E_{\tilde{\mathbf{x}},p}\|_{L^q(\Omega_k)} \leq \omega \quad (5.23)$$

where, for any $q \in [1, \infty)$

$$\omega = c_{p,q} A_p \rho_p^{-0.5} I^{\left(\frac{p+1}{2} + \frac{1}{q}\right)} \quad (5.24)$$

$$c_{p,q} = \left(\frac{2\pi}{q(p+1)+2} \left(\frac{1}{\pi} \right)^{\frac{q(p+1)+2}{2}} \right)^{1/q}, \quad (5.25)$$

and for $q = \infty$,

$$\omega = A_p \rho_p^{-0.5} I^{\left(\frac{p+1}{2}\right)}, \quad (5.26)$$

and I is the area of the mesh element. In the present case, we find the area using the adjoint error estimate as given in Section 5.2.2.

5.3 Anisotropic hp -Adaptation

The hp -adaptation strategy of Dolejsi [30] is extended to incorporate adjoint error estimate to choose the area of the triangular mesh elements. The original method

generates anisotropic meshes such that the interpolation error in the L^q -norm ($q \in [1, \infty)$) is under a given tolerance, and the number of degrees of freedom in minimum. Equivalently, one can also set the area of the element and find the anisotropy and the polynomial degree which gives the minimum interpolation error. In the current method, we set the area of the element using the adjoint error estimate like in the case of anisotropic h -refinement and choose the configuration (anisotropy and polynomial degree) which gives the least interpolation error estimate.

5.3.1 Algorithm for hp -adaptation

The adaptation step consists of the following.

Algorithm B

- (1) For each mesh element $\Omega_k \in \mathcal{T}_h$,
 - (a) Find the implied metric using Eq. (5.2).
 - (b) Determine the desired area I , using an adjoint-based error estimate and the current area, as explained in Section 5.2.2.
 - (c) If the anisotropy variable $u_h \in P^{p_k}$, then we consider three cases: the reconstructed variable $\tilde{u}_h$ to be polynomial of degree $p \in \{p_k - 1, p_k, p_k + 1\}$. In all the three cases, determine the anisotropy triplet (A_p, ρ_p, ϕ_p) of the interpolation function $E_{\hat{x}, p}$ as given in Section 5.2.4, and the corresponding interpolation error bound ω using Eq. (5.24) and Eq. (5.25) if the bound is measured in L_q , $q \in [1, \infty)$, or by using Eq. (5.26) for L_q , $q = \infty$.
 - (d) Choose the polynomial degree p and the corresponding triplet (I, β, θ) using Eq. (5.22) which gives the smallest error bound ω .
 - (e) Find the desired metric $\mathcal{M}$ from the new triplet (I, β, θ) using Eqns. (5.3), (5.4), and (5.5).
- (2) The new metric $\mathcal{M}$ evaluated at all the elements of the current mesh, and the current mesh are used as the inputs to the mesh generator to produce the desired mesh, that conform to the given metric $\mathcal{M}$. The new polynomial orders are treated as real variables, and are interpolated to the nodes of the new mesh. Subsequently, they are averaged to the cells of the new mesh and rounded to integers.

This process is repeated till the target functional is within the given error tolerance. Note that for $p = p_k$ and $p = p_k + 1$, we need to reconstruct u to the spaces P^{p_k+1}

and P^{p_k+2} to find the $(p+1)$ -st derivative. This is done using the patch reconstruction given in Section 5.2.3. Similar to the mesh fraction f_h used in h -adaptation to find a reference error, a user specified mesh fraction f_p is also used in hp -adaptation to find the reference error for p -adaptation. Only those elements with (adjoint-based) errors higher than this reference error are considered for p -adaptation. And those elements with (adjoint-based) errors lower than this reference error would retain the original polynomial degree p_k , even if the interpolation error bound suggests $p_k + 1$.

5.4 Verification of the Interpolation Error Bound

This section aims to validate the inequality given in Eq. (5.20). First, we define a function $E_\Omega : \Omega \rightarrow \mathbb{R}$ such that $E_\Omega|_{\Omega_k} = E_{\hat{\mathbf{x}}_k, p_k}|_{\Omega_k}$, where $E_{\hat{\mathbf{x}}_k, p_k}$ is the interpolation error function given in Eq. (5.19). To quantify the error for the whole domain we use the L^q norm of the function E_Ω , i.e.

$$\|E_\Omega\|_{L^q(\Omega)} = \left(\sum_{\Omega_k \in \mathcal{T}_h} \int_{\Omega_k} |E_{\hat{\mathbf{x}}_k, p_k}|^q dx \right)^{\frac{1}{q}} \quad (5.27)$$

We define another function $B_\Omega : \Omega \rightarrow \mathbb{R}$ such that

$$B_\Omega|_{\Omega_k} = \left(A_{p_k} \left((\mathbf{x} - \hat{\mathbf{x}})^T Q_{\phi_{p_k}} D_{\rho_{p_k}} Q_{\phi_{p_k}}^T (\mathbf{x} - \hat{\mathbf{x}}) \right)^{\frac{p_k+1}{2}} \right) \Big|_{\Omega_k}, \quad (5.28)$$

where the right-hand-side is the restriction to element Ω_k of the interpolation error bound as given in the right-hand-side of Eq. (5.20). If the coefficients $(\tilde{A}_p, \tilde{\rho}_p, \tilde{\phi}_p)$ are used in the right-hand-side of Eq. (5.28), then the corresponding function is denoted as $\tilde{B}_\Omega$. (Recall from Section 5.2.4.3 the difference between these two choices.) To verify the bound, the test function as given below is used:

$$\begin{aligned} w(x, y) &= \left(x + \frac{e^{x/\epsilon} - 1}{1 - e^{1/\epsilon}} \right) \cdot \left(y + \frac{e^{y/\epsilon} - 1}{1 - e^{1/\epsilon}} \right), & (x, y) \in \Omega = [0, 1]^2 \\ w(x, y) &= 0, & (x, y) \in \partial\Omega. \end{aligned} \quad (5.29)$$

This forms boundary layers near $x = 1$ and $y = 1$ for low values of ϵ . Numerical simulations have been performed on various meshes - both uniform and adapted ones. To test the bound, we evaluated the L^2 -norm of the error function E_Ω and the error-bound functions $\tilde{B}_\Omega$ and B_Ω over the domain. The error function E_Ω is evaluated with the exact derivatives and the error-bound functions $\tilde{B}_\Omega$ and B_Ω are evaluated using

the derivatives approximated using the patch reconstruction. This enables us to check how well the high order derivatives are approximated and how well the inequality for the domain $E_\Omega \leq \tilde{B}_\Omega$ holds true. We also check if the modification of the coefficients improves the safety of the error bound, i.e. we check the inequality $E_\Omega \leq B_\Omega$.

The tests are conducted for different polynomial approximation degrees ($p = 1, 2, 3, 4$) on uniform and adapted (anisotropic h -adapted) meshes. Fig. 5.4 shows the L^2 -norm of the error function E_Ω and the error-bound function $\tilde{B}_\Omega$ for four different meshes (uniformly refined) and varying polynomial approximations for $\epsilon = 0.1$. For each polynomial approximation degree and mesh, the bound seems to be fairly tight with the error function.

(a) L^2 -norm of E_Ω and $\tilde{B}_\Omega$ vs. number of elements, $\epsilon = 0.1$

Figure 5.4: Verification of the inequality Eq. (5.20) on uniformly refined meshes and varying polynomial approximation degrees. (---): $\|E_\Omega\|_{L^2(\Omega)}$, (—■—): $\|\tilde{B}_\Omega\|_{L^2(\Omega)}$.

Figs. 5.5 shows the L^2 -norm of the functions E_Ω , $\tilde{B}_\Omega$, B_Ω on a sequence of anisotropic meshes that have been obtained from the adaptation process outlined in Section 5.2, for various values of ϵ and polynomial degrees. With decreasing values of ϵ the boundary layer gets stronger. For all three values of ϵ , and for varying polynomial approximations, the bounds are again found to be valid. The L^2 -norm of E_Ω is generally below that of the corresponding error-bound function $\tilde{B}_\Omega$. As expected, using the modified set of coefficients, i.e. using B_Ω , improves the safety of the bound. Note that for these adaptations, the total weighted boundary flux, as explained in Section 5.5.1.1, is used as the target functional, and not the L^2 -norm of the functions E_Ω , $\tilde{B}_\Omega$ or B_Ω . This is the reason why we do not see optimal convergence in the L^2 -norm of these functions for

various polynomial degrees. The sole purpose here is to demonstrate that the bound is safe even on non-uniform meshes.

It is also worth looking at the element-wise L^2 -norm of the error function E_Ω and the bounds B_Ω and $\tilde{B}_\Omega$. Fig. 5.6 plots these quantities for various values of ϵ . For each value of ϵ , an anisotropic h -adapted mesh is chosen to plot the values. The L^2 -norm of the element-wise bounds $\|B_\Omega\|_{L^2(\Omega_k)}$ or $\|\tilde{B}_\Omega\|_{L^2(\Omega_k)}$ have been sorted in the ascending order and these sorted values are plotted along with the corresponding L^2 -norm of the elemental error functions $\|E_\Omega\|_{L^2(\Omega_k)}$. (Consequently the curves for the bounds are seen as smooth plots, whereas the errors have high oscillations, which is thus an artifact of the postprocessing.) Note that all the plots on the left side (Figs. 5.6a, 5.6c, 5.6e) use the function $\tilde{B}_\Omega$ as the bound and those on the right side (Figs. 5.6a, 5.6c, 5.6e) use the function B_Ω as the bound. It is evident that both the bounds are fairly tight, even then there are some overshoots in the error function. With the modified set of coefficient, the overshoots are fewer, though. A strict bound is not expected here, because the bound B_Ω uses the estimated $(p+1)$ -st derivatives. If E_Ω is evaluated using the same approximate derivatives, then the bound, by construction, is strictly guaranteed for B_Ω . Such a plot is shown in Fig. 5.7. Note that in Figs. 5.6c, 5.6d, 5.6e, 5.6f, and 5.7, since the values for E_Ω are very small until around element number 400, they are skipped and not shown in the plots.

As a conclusion, we can infer from these results that the bound given by the function $\tilde{B}_\Omega$ is reasonably safe. The bound becomes safer with B_Ω , but at the expense of more computational work to modify the coefficients. We have thus used the coefficients $(A_p, \rho_p, \phi_p) = (\tilde{A}_p, \tilde{\rho}_p, \tilde{\phi}_p)$ to find the anisotropy for h -adaptation. But in case of hp -adaptation, since the interpolation error bounds are compared for choosing the polynomial degrees, we use the modified set of coefficients as proposed in [30] to get a somewhat safer estimate.

 (a) L^2 -norm of functions E_Ω , $\tilde{B}_\Omega$ and B_Ω vs. number of elements, $\epsilon = 0.1$

 (b) L^2 -norm of E_Ω , $\tilde{B}_\Omega$ and B_Ω vs. number of elements, $\epsilon = 0.01$

 (c) L^2 -norm of E_Ω , $\tilde{B}_\Omega$ and B_Ω vs. number of elements, $\epsilon = 0.005$

 Figure 5.5: Verification of the inequality Eq. (5.20) on adjoint-based adapted anisotropic meshes and varying polynomial approximation degrees. (---): $\|E_\Omega\|_{L^2(\Omega)}$, ($\blacksquare$): $\|\tilde{B}_\Omega\|_{L^2(\Omega)}$, ($\circ$): $\|B_\Omega\|_{L^2(\Omega)}$.

Figure 5.6: Element-wise verification of inequality Eq. (5.20) on adapted anisotropic meshes. The numbering of the elements is sorted with respect to the L^2 -norm of the error bound.

Figure 5.7: Element-wise verification of inequality Eq. (5.20) on an adapted anisotropic mesh. The numbering of the elements is sorted with respect to the L^2 -norm of the error bound. $\|E_\Omega\|_{L^2(\Omega_k)}$ is evaluated using approximate derivatives.

5.5 Numerical Results

In this section, the adaptation (both h and hp) is applied to different test cases to demonstrate the effectiveness of the anisotropic h - and hp -adaptation methods. The governing equations are discretized using the HDG method as discussed in Section 3.3. The discretized equations are solved using the damped Newton method (Section 3.3.1) and the resulting linear sub-problems are solved using a restarted GMRES with ILU(3) preconditioner. The same GMRES method is also used for the discrete adjoint problems. The Newton iterations are stopped when the L_2 norm of the residual is smaller than 10^{-10} . At first, a scalar convection-diffusion is considered, where there is a strong boundary layer which is expected to be resolved well using anisotropic meshes. Then, in order to verify the method for aerodynamic applications, we solve the Euler and the Navier-Stokes equations over the NACA0012 airfoil.

Four cases of flow over the NACA0012 airfoil are considered.

- Viscous, subsonic flow, $\text{Ma}_\infty = 0.5, \alpha = 1.0^\circ, \text{Re} = 5000$
- Inviscid, subsonic flow, $\text{Ma}_\infty = 0.5, \alpha = 2^\circ$
- Inviscid, transonic flow, $\text{Ma}_\infty = 0.8, \alpha = 1.25^\circ$
- Inviscid, supersonic flow, $\text{Ma}_\infty = 1.5, \alpha = 0^\circ$

Flows with higher Reynolds numbers are not considered due to the limitations in the computational resources. The geometry of the airfoil is defined by

$$y = \pm 0.6(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4) \quad (5.30)$$

with $x \in [0, 1]$. The initial mesh (Fig. 5.8) with 2155 triangular elements was created using the Netgen mesh generator [71]. The far field is a circle, centered at the airfoil mid chord with a radius of 1000 chords. This mesh has been used as the initial mesh for all the above test cases. In order to assess the effect of discretization orders on the error convergence rate, all the test cases are done for different solution spaces ($p = 1, 2, 3$). Mach number has been used as the anisotropy variable for all the NACA0012 test cases. In addition to the anisotropic adaptation, we also do isotropic adaptation (see Chapter 4) and the results are compared. For isotropic adaptation, we use fixed fraction marking strategy, and those elements which are marked are subdivided to form child elements. Please note that the adapted meshes are conforming, i.e. after the subdivision, the neighbouring elements are refined as well to prevent hanging nodes. We need to specify mesh fraction as a parameter for isotropic refinement. Typically

5 – 10% is used. In the convergence plots shown in the section on numerical results, 'iso' denotes isotropic refinement. For hp -adaptation on isotropic meshes, the jump sensor described in Section 4.4 is used to choose between h - and p -refinement.

(a) Initial mesh - 2155 elements, magnified view around the airfoil

Figure 5.8: Initial mesh for the compressible flow test cases around the NACA0012 airfoil

We give a short description about the parameters used for the anisotropic adaptation. For the h -adaptation, there are three parameters. They are the mesh fraction f_h that determines the reference error for refinement and coarsening, the maximum refinement factor r_{max} , and the maximum coarsening factor c_{max} . We have used $f_h = 0.3$ for all the test cases for h -adaptation. With this choice, around 30% of the total number of elements get refined and the remaining get coarsened. Although any values in the range of 0.1 – 0.4 give acceptable results, 0.3 seems to be the best choice for the test cases we considered. For the scalar test case, r_{max} is taken in the range of 16 – 20. For this strongly anisotropic test cases, aggressive refinement with high values of r_{max} proved to be beneficial. For the flow cases around the NACA0012 airfoil, aggressive refinement is not found to be optimal, and hence lower values for r_{max} in the range of 6 – 10 are considered for h -adaptation. In choosing c_{max} , one should be a bit more careful since higher values of c_{max} may lead to too much of coarsening thereby affecting the convergence. We have mostly used values in the range of 2 – 4 to be on the safer side even though higher values may still give similar convergence. For the scalar test

case with boundary flux as the target functional, we could use values of up to $c_{max} = 20$ and still obtained good convergence.

For hp -adaptation, in addition to the parameters mentioned above, there is only one more parameter - f_p , the mesh fraction that determines the reference error for p -refinement. For the scalar test case and the viscous subsonic flow around the NACA0012 airfoil, a high value of $f_p = 0.3$ gives good results. This must be due to the fact that, these test cases have regions that give a higher contribution to the error, but that can get resolved well with higher polynomial approximations thereby reducing the error. For the inviscid test cases, which have shocks or singularities, a smaller value of f_p in the range of $0.1 - 0.2$ is found to be sufficient. Compared to h -adaptation, hp -adaptation needs only a lower value of f_h since the same region may also get p -refined. So for hp -adaptation, we used f_h in the range of $0.1 - 0.2$. The factor r_{max} used is in the range of $6 - 8$, except for the scalar test case with L^2 error as the target functional where aggressive refinement is needed to reduce the error down to a small value, and hence a high value of $r_{max} = 20$ was used. Similar to the case of h -adaptation, one must be a bit careful in choosing the factor c_{max} as a very high value may lead to excessive coarsening. For the hp -adaptation, values for c_{max} in the range of $4 - 8$ have given good results.

Exact values for the parameters used are given in the sections of the corresponding test cases where the results are described. The results presented are the best ones among those we got for various values of the parameters. However, we found that the adaptation is pretty robust with respect to these parameters, in the sense that the convergence remains almost the same if the values of the parameters are changed to a certain extent. In order to substantiate this, we present here results from some of the test cases (defined in the subsequent sections) using various parameter settings. The first is for the test case defined in Section 5.5.1.2 using hp -adaptivity and various parameter settings (cf. table 5.1). Here the mesh fractions, f_p and f_h are kept constant and the refinement/coarsening ratios, r_{max} and c_{max} are varied. Fig. 5.9 shows the convergence in the target error functional using those parameter settings, and it is clear that the convergence rates are similar. The second one is for the hp -adaptivity done for the test case defined in Section 5.5.2. The various parameters used are given in table 5.2. Here we keep all the parameters but f_p as constant. The convergence plots are shown in Fig. 5.10. It is evident from the plots that, even though f_p varies significantly from a value of 20% to 50%, the convergence rates are almost the same. Overall, our experience with testing parameters indicates that for the test cases we considered, the simulations give good results even when the following wide range of values for the parameters are used: $f_p \approx 10 - 50\%$, $f_h \approx 10 - 30\%$, $r_{max} \approx 4 - 20$,

and $c_{max} \approx 2 - 10$. Values outside this range are still expected to work, but they may result in longer time for the convergence. For instance, a very low value for the mesh fractions f_p and f_h or for the refinement ratio r_{max} would naturally result in slower convergence.

Test case	f_p	f_h	r_{max}	c_{max}
hp 1	0.3	0.1	20	4
hp 2	0.3	0.1	20	10
hp 3	0.3	0.1	10	4
hp 4	0.3	0.1	10	2
hp 5	0.3	0.1	12	4

Table 5.1: Various parameter settings for the scalar convection diffusion test case defined in Section 5.5.1.2

(a) Error in target functional vs. degrees of freedom

Figure 5.9: Scalar convection-diffusion test case, $\epsilon = 0.005$, L^2 -norm of solution error as the target functional, comparison of error convergence for various parameter settings for anisotropic hp -adaptivity

Test case	f_p	f_h	r_{max}	c_{max}
hp 1	0.2	0.1	6	4
hp 2	0.3	0.1	6	4
hp 3	0.4	0.1	6	4
hp 4	0.5	0.1	6	4

Table 5.2: Various parameter settings for the viscous subsonic test case defined in Section 5.5.2

(a) Error in drag coefficient vs. degrees of freedom

Figure 5.10: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of error convergence for various parameter settings for anisotropic hp -adaptivity.

Finally, we mention that for all the cases, the interpolation error bound was estimated for $q = 2$.

5.5.1 Scalar convection-diffusion equation

We consider the scalar convection-diffusion equation

$$\begin{aligned} \nabla \cdot (w, w) - \epsilon \Delta w &= s & (x, y) \in \Omega = [0, 1]^2 \\ w(x, y) &= 0 & (x, y) \in \partial\Omega. \end{aligned}$$

We take the solution as

$$w(x, y) = \left(x + \frac{e^{x/\epsilon} - 1}{1 - e^{1/\epsilon}} \right) \cdot \left(y + \frac{e^{y/\epsilon} - 1}{1 - e^{1/\epsilon}} \right),$$

which forms a boundary layer for lower values of ϵ . Substituting this solution into the convection-diffusion equation gives a source function which we then use to solve

the equation. We have chosen $\epsilon = 0.005$, in which case there is a strong boundary layer on the top right corner of the domain. The presence of such a strong gradient in the solution makes it a suitable candidate to perform hp -adaptation. Two cases are considered where, in the first case a weighted total boundary flux is taken as the target functional, and in the second case L^2 -norm of solution error is taken as the target functional. A uniform mesh with 512 elements has been used as the initial mesh. In both cases, h and hp -adapted simulations on anisotropic meshes have been performed and the results are compared with those on isotropic meshes.

5.5.1.1 Target functional - total weighted boundary flux

The target functional is taken as the weighted total boundary flux i.e.

$$J = \int_{\partial\Omega} \psi(w - \epsilon n \cdot \nabla w) d\sigma \quad (5.31)$$

where the weighting function, $\psi = \cos(2\pi x) \cos(2\pi y)$.

Anisotropic adaptations (both h and hp) have been performed and the results are compared with that of isotropic adaptations. For the anisotropic h -adaptation, $f_h = 0.3$, $r_{max} = 16$, $c_{max} = 20$ are used as the parameters. Note that very aggressive coarsening and refining are used here. For this strongly anisotropic test case, such aggressive ratios have proved to be beneficial. Fig. 5.11 shows the initial mesh (512 elements) and the h -adapted anisotropic mesh after 3 adaptations (1559 elements) for polynomial approximation degree $p = 2$. We can see the highly skewed elements near the boundary layer on the adapted mesh. In the adapted mesh almost all the elements are in the boundary layer, while the region outside the boundary layer is resolved with very few elements. The corresponding contour plots of the solution can be seen in Fig. 5.12. Fig. 5.13 and Table 5.3 compare the convergence in error for isotropic and anisotropic h -adaptation. With anisotropic h -adaptation we could get an error of the order of 10^{-10} with just 1559 elements whereas the isotropic refinement needed 5657 elements to reach that error level.

For anisotropic hp -adaptation, the parameters used are $f_p = 0.3$, $f_h = 0.1$, $r_{max} = 8$, $c_{max} = 8$. With hp -adaptation, we could get the error down from around 10^{-5} (512 elements, 3072 degrees of freedom) to around 10^{-11} (295 elements, 4081 degrees of freedom), in just 4 adaptation steps. Fig. 5.14 and Table 5.4 show the error convergence and compares it with that of isotropic hp -adaptivity. The almost vertical convergence plot of the hp -adaptation implies that the algorithm works quite well in redistributing the degrees of freedom in the domain to reduce the error. Fig. 5.15 shows solution con-

Figure 5.11: Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic h -adaptation, $p = 2$.

Isotropic		Anisotropic	
ndof	$ J(w) - J(w_h) $	ndof	$ J(w) - J(w_h) $
3072	2.33495e-05	3072	2.33496e-05
3864	9.65946e-06	6276	1.87306e-07
5124	5.41475e-07	7596	5.61738e-09
7434	2.95909e-07	9354	1.44080e-10
10296	9.42015e-08		
13344	1.04134e-08		
18150	5.76061e-09		
24774	2.37663e-09		
33942	1.46091e-10		

Table 5.3: Scalar convection-diffusion test case, $\epsilon = 0.005$, total weighted boundary flux as target functional, isotropic and anisotropic h -adaptations, $p = 2$.

tours and the polynomial map on the corresponding meshes for different adaptation steps. With adaptations, the boundary layer region gets higher polynomial approximation degrees thereby resolving the layer in the most efficient manner. As how it should be, the region outside the boundary layer neither gets mesh refinement nor gets polynomial space enrichment. Fig. 5.16 compares the solution near the boundary layer on both an anisotropic h -adapted mesh (1559 elements) and an anisotropic hp -adapted mesh (307 elements) which give almost the same error level of $\approx 10^{-10}$. We can observe that on the hp -adapted mesh, even though its much coarser than the h -adapted mesh, we get a solution as well-resolved and of same error level as that of the h -adapted mesh because of the higher polynomial approximation in the region.

Figure 5.12: Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic h -adaptation, $p = 2$.

Figure 5.13: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of error convergence for isotropic and anisotropic h -adaptation. 'iso' denotes isotropic refinement, $p = 2$, total weighted boundary flux as target functional.

5.5.1.2 Target functional - L^2 -norm of solution error

The target functional is taken as the L^2 -norm of solution error i.e.

$$J = \|w - w_h\|_{L^2} = \left(\int_{\Omega} (w - w_h)^2 dx \right)^{\frac{1}{2}}. \quad (5.32)$$

(a) Error in target functional vs. degrees of freedom

Figure 5.14: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of error convergence for isotropic and anisotropic hp -adaptation, 'iso' denotes isotropic refinement, total weighted boundary flux as target functional.

Isotropic		Anisotropic	
ndof	$ J(w) - J(w_h) $	ndof	$ J(w) - J(w_h) $
3072	2.33495e-05	3072	2.33495e-05
4030	1.05500e-05	3134	2.02011e-06
5272	1.77056e-07	3346	4.60944e-09
6356	2.00450e-08	3757	1.83977e-10
6919	4.69426e-08	4081	5.62836e-11
7917	2.42186e-09		
8628	3.72201e-09		
10553	2.37146e-09		
12488	4.64518e-10		
15213	5.51928e-11		

Table 5.4: Scalar convection-diffusion test case, $\epsilon = 0.005$, total weighted boundary flux as target functional, isotropic and anisotropic hp -adaptations.

For anisotropic h -adaptation, $f_h = 0.3$, $r_{max} = 20$, $c_{max} = 4$ are used as the parameters. Fig. 5.17 compares the convergence in error for isotropic and anisotropic h -adaptation for solution approximated with polynomials of degree $p = 3$. It is interesting to see that, with anisotropic adaptation the error gets reduced by a factor of 6 in just 3 steps, and afterwards it seems to maintain the same convergence rate as that of isotropic refinement. The reason could be that, during the initial few adaptations the mesh reorients itself such that the elemental contributions to the target error become equal and any further adaptations can only give a convergence rate as that of an isotropic refinement. For the anisotropic hp -adaptation, $f_p = 0.3$, $f_h = 0.1$, $r_{max} = 20$, $c_{max} = 4$ are used as the parameters. Fig. 5.18 shows the solution and the polynomial degree distribution on the 5th hp -adapted mesh, and the magnified view near the boundary layer

Figure 5.15: Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic hp -adaptation, total weighted boundary flux as target functional.

Figure 5.16: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of anisotropic h - and hp -adapted mesh which give the same error level of around 10^{-10} , total weighted boundary flux as target functional, magnified view near the boundary layer.

can be seen in Fig. 5.19. Similar to the previous test case, here also the boundary layer region gets higher polynomial approximation degrees, and the region outside the layer gets neither mesh refinement nor polynomial space enrichment. Fig. 5.20 compares the error convergence of anisotropic hp -adaptation with isotropic hp -adaptation. There is a significant improvement in efficiency for anisotropic hp -adaptation over isotropic ones. In Fig. 5.21, the error convergence on both isotropic and anisotropic meshes for h -adaptation ($p = 2, 3$) and hp -adaptation together are plotted. It is evident from the plots that the anisotropic h - and hp -adaptation are significantly more efficient than isotropic h - and hp -adaptation, respectively. Tables 5.5 and 5.6 tabulates the convergence in error achieved in h -adaptations and hp -adaptations, respectively. Overall, anisotropic hp -adaptation performs the best in terms of number of degrees of freedom to reduce the error in the target functional.

(a) Error in target functional vs. degrees of freedom

Figure 5.17: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of isotropic and anisotropic h -adaptation, L^2 -norm of solution error as the target functional, 'iso' denotes isotropic refinement.

5.5.2 Viscous, subsonic flow

We consider viscous, subsonic flow around the NACA0012 airfoil with a free-stream Mach number $\text{Ma}_\infty = 0.5$, angle of attack $\alpha = 1^\circ$ and Reynolds number $\text{Re} = 5000$. Governing equations are the Navier-Stokes equations (Section 2.5). Both h -adapted and hp -adapted simulations have been performed with the drag coefficient C_d as the target functional. Initial mesh with 2155 elements is shown in Fig. 5.8. For anisotropic h -adaptation, the parameters used are $f_h = 0.3$, $r_{\max} = 6$, $c_{\max} = 4$. Fig. 5.22 shows

Figure 5.18: Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic hp -adaptation, 5th adapted step, 1231 elements, L^2 -norm of solution error as the target functional.

Figure 5.19: Scalar convection-diffusion test case, $\epsilon = 0.005$, anisotropic hp -adaptation, 5th adapted step, 1231 elements, L^2 -norm of solution error as the target functional, magnified view near the boundary layer.

(a) Error in target functional vs. degrees of freedom

Figure 5.20: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of isotropic and anisotropic hp -adaptation, L^2 -norm of solution error as the target functional, 'iso' denotes isotropic refinement.

(a) Error in target functional vs. degrees of freedom

Figure 5.21: Scalar convection-diffusion test case, $\epsilon = 0.005$, comparison of isotropic and anisotropic h - and hp -adaptation, L^2 -norm of solution error as the target functional, 'iso' denotes isotropic refinement.

the h -adapted mesh (11945 elements) and the Mach number contours on it for $p = 2$ discretization. The essential features of this flow are a laminar boundary layer around the airfoil, and a flow separation and corresponding recirculation zone near the trailing edge on the upper side of the airfoil. Both of these features have to be resolved well to get accurate values for the drag coefficient. As expected, both the boundary layer

$p = 2$, isotropic		$p = 2$, anisotropic	
ndof	$\ w - w_h\ _{L^2}$	ndof	$\ w - w_h\ _{L^2}$
3072	9.51222e-02	3072	9.51223e-02
4452	2.22744e-02	7152	1.35133e-03
6384	3.78801e-03	15084	1.12254e-05
9438	1.03505e-03	19362	1.04849e-06
15678	3.37145e-04	33210	3.35721e-07
24936	1.00633e-04	50724	1.58630e-07
40098	4.62525e-05	69342	8.81227e-08
64524	2.07492e-05	95802	5.27650e-08
101418	9.36656e-06	137142	3.08209e-08
159546	4.62994e-06		
$p = 3$, isotropic		$p = 3$, anisotropic	
ndof	$\ w - w_h\ _{L^2}$	ndof	$\ w - w_h\ _{L^2}$
5120	3.78495e-02	5120	3.78495e-02
7420	6.01069e-03	11680	2.09668e-04
10580	7.80648e-04	21400	8.44138e-07
16160	1.83791e-04	27180	4.99810e-08
27560	3.46719e-05	45510	1.21622e-08
42800	9.71775e-06	65760	4.44191e-09
72380	3.33145e-06		
113260	1.07160e-06		
181130	4.24854e-07		

Table 5.5: Scalar convection-diffusion test case, $\epsilon = 0.005$, L^2 -norm of solution error as the target functional, isotropic and anisotropic h -adaptations for $p = 2$ and $p = 3$.

Isotropic		Anisotropic	
ndof	$\ w - w_h\ _{L^2}$	ndof	$\ w - w_h\ _{L^2}$
3072	9.51222e-02	3072	9.51222e-02
4460	2.22804e-02	5352	4.57002e-04
6382	4.06525e-03	8888	1.84748e-06
9064	8.50788e-04	11623	1.45476e-07
15304	1.44478e-04	15073	1.47985e-08
30235	2.93555e-05	23734	2.25651e-09
43828	7.58220e-06	34797	3.84877e-10
61641	1.70419e-06	66134	6.40106e-11
83533	4.98037e-07		
128518	1.47703e-07		
205014	4.34907e-08		
326197	1.01625e-08		
489835	1.96572e-09		

Table 5.6: Scalar convection-diffusion test case, $\epsilon = 0.005$, L^2 -norm of solution error as the target functional, isotropic and anisotropic hp -adaptations.

and the recirculation zone are well-resolved with highly anisotropic mesh elements. We can see the highly anisotropic mesh elements at the boundary layer and the corresponding smooth contours of the Mach number in Fig. 5.22. The recirculation zone is also well-captured with anisotropic elements as shown in Fig. 5.23. The figure shows both the mesh and the streamlines near the recirculation region. It is interesting to see a thin layer of highly anisotropy elements that perfectly passes through the region where the streamlines change its direction from left to right. Fig. 5.24 compares the convergence of the error in the drag coefficient for the isotropic and the anisotropic

h -refinement for $p = 2$. To compute the error in the drag coefficient, a reference value of $C_{d,\text{ref}} = 5.531683 \times 10^{-2}$ was taken from a fine adapted mesh with around 5.1×10^5 degrees of freedom and the solution approximation degree $p = 4$. The anisotropic h -refinement seems to be significantly more efficient compared to the isotropic refinement.

For anisotropic hp -adaptation, $f_p = 0.3$, $f_h = 0.1$, $r_{\max} = 8$, $c_{\max} = 4$ are used as the parameters. Fig. 5.25 shows the adapted mesh, Mach number contours and the corresponding polynomial map for the hp -adaptive case. The boundary layer region around the airfoil, the stagnation region near the leading edge and the wake in the region downstream of the airfoil get higher polynomial degree. Similar to the pure h -adaptive case, we can also see a layer of highly anisotropic mesh elements near the trailing edge to resolve the recirculation of flow (Fig. 5.26). In the case of hp -adaptation too, the anisotropic refinement outperforms the isotropic refinement as evident from Fig. 5.27. Fig. 5.28 shows the convergence results for both h - and hp -adaptation on both isotropic and anisotropic elements. Tables 5.7 and 5.8 tabulates the error convergences achieved in h -adaptations and hp -adaptations, respectively. Clearly the anisotropic refinement is superior to the isotropic one for both h - and hp -adaptation in terms of the number of degrees of freedom. Increasing the polynomial degree makes the convergence rate better for h -adaptation. Overall, hp -adaptation works the best and its convergence rate is better than that of any of the considered h -adaptations.

The computational cost in terms of the work units (time taken for the computation normalized with the time taken for the Tau-bench computation [40]) is also plotted in Fig. 5.29. The curves follow, more or less, the same trend as that in the previously mentioned convergence plots which are plotted in terms of the degrees of freedom. As expected, since the number of degrees of freedom involved are much lower for the anisotropic adaptations, the computational time for those adaptations are also lower compared to that of isotropic adaptations. It has also to be mentioned that for the isotropic adaptations, the solutions can be easily projected to the new mesh since the adapted meshes are hierarchical. These projected solutions provide a good initial approximation for the next simulation. For the anisotropic adaptations, we can only start with a linearly extrapolated solution (given by BAMG) from the old mesh and hence the convergence takes a longer time compared to that of isotropic adaptation. But in terms of the overall computational time required to achieve a particular error level, the anisotropic adaptations are still more efficient, as evident from the plots, due to the lower number of degrees of freedom. There is also computational overhead associated with the generation of anisotropic mesh from the base mesh and the metric using the

mesh generator. But for the 2D cases we considered, the time taken for the mesh generation is relatively negligible. For the more complex 3D cases, the computational overhead associated with the mesh generation may become significant. We would like to stress that the present work concerns mostly with the algorithm optimization in terms of number of degrees of freedom and not in terms of computational cost. Hence, we skip comparisons of the latter for the subsequent test cases. A detailed study and algorithm optimization in terms of the computational cost is left for future research.

Figure 5.22: Viscous subsonic flow around the NACA0012, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, $p = 2$, h -adaptation with drag coefficient as target.

Figure 5.23: Viscous subsonic flow around the NACA0012, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, $p = 2$, h -adaptation with drag coefficient as target.

Figure 5.24: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of error convergence for isotropic and anisotropic h -adaptation. 'iso' denotes isotropic refinement, $p = 2$.

Figure 5.25: Viscous subsonic flow around the NACA0012, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, *hp*-adaptation with drag coefficient as target. $p = 1, \dots, 6$.

Figure 5.26: Viscous subsonic flow around the NACA0012, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, hp -adaptation with drag coefficient as target. $p = 1, \dots, 6$.

Figure 5.27: Viscous subsonic flow, $Ma_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, comparison of error convergence for isotropic and anisotropic hp -adaptation. 'iso' denotes isotropic refinement, $p = 1, \dots, 6$.

(a) Error in drag coefficient vs. degrees of freedom

Figure 5.28: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic refinement. For hp -adaptation, $p = 1, \dots, 6$.

$p = 1$ (iso)		$p = 2$ (iso)		$p = 3$ (iso)	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	7.3131e-03	12930	9.9054e-03	21550	6.9430e-05
8088	5.6375e-04	16188	3.9648e-04	28020	2.8912e-04
10434	1.0848e-03	21534	3.2206e-04	37570	6.4930e-05
14310	1.5074e-03	29826	1.5752e-04	51080	3.7610e-05
19773	6.6313e-04	41196	6.3990e-05	69300	1.3920e-05
26913	3.8112e-04	56592	4.1740e-05		
36948	2.6112e-04	77172	2.1470e-05		
50592	1.6325e-04				
68670	1.1841e-04				

$p = 1$		$p = 2$		$p = 3$	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	7.3131e-03	21550	6.9430e-05	21550	6.9430e-05
8214	6.8172e-04	23600	1.4704e-04	23600	1.4704e-04
10881	3.9399e-04	24870	4.8620e-05	24870	4.8620e-05
14325	1.5772e-04	28650	1.4870e-05	28650	1.4870e-05
17949	9.4940e-05	29410	6.2600e-06	29410	6.2600e-06
21063	7.7140e-05	31330	2.5400e-06	31330	2.5400e-06
24222	6.9160e-05	33940	1.0400e-06	33940	1.0400e-06
27957	5.7060e-05	37460	5.3000e-07	37460	5.3000e-07
32085	4.9410e-05	39450	2.6000e-07	39450	2.6000e-07
35502	4.3760e-05	41960	1.8000e-07	41960	1.8000e-07
44100	1.3000e-07	44100	1.3000e-07		
46260	1.1000e-07	46260	1.1000e-07		

Table 5.7: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, isotropic and anisotropic h -adaptations for various polynomial approximation degrees. 'iso' denotes isotropic adaptation.

Isotropic		Anisotropic	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	7.3131e-03	6465	7.3131e-03
7113	5.9786e-03	9573	1.3976e-03
7863	9.9875e-04	12753	1.1619e-04
9392	1.2525e-03	15924	4.1980e-05
11104	8.7986e-04	19913	1.2720e-05
12868	7.0765e-04	23223	6.3800e-06
14624	4.0869e-04	26612	3.0100e-06
16633	2.6899e-04	30820	9.6000e-07
18648	1.7451e-04	35613	5.9000e-07
21831	9.7050e-05	39316	2.2000e-07
26322	6.2100e-05	44769	1.5000e-07
30018	4.7860e-05	50416	6.0000e-08
33088	3.6050e-05	59669	3.0000e-08
38440	2.7240e-05		
45416	1.6870e-05		
51683	1.1270e-05		

Table 5.8: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, isotropic and anisotropic hp -adaptations.

(a) Error in drag coefficient vs. work units

Figure 5.29: Viscous subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$, comparison of error convergence with respect to the work units for isotropic and anisotropic adaptations. 'iso' denotes isotropic refinement.

5.5.3 Inviscid, subsonic flow

The steady state Euler equations have been solved for the flow around the NACA 0012 airfoil with a free stream Mach number $\text{Ma}_\infty = 0.5$ and an angle of attack $\alpha = 2^\circ$. The drag coefficient has been used as the target functional for the adjoint-based adaptations. Both anisotropic and isotropic adaptations are performed on an initial mesh with 2155 mesh elements (Fig. 5.8). For anisotropic h -adaptation the parameters used are $f_h = 0.3$, $r_{\max} = 10$, $c_{\max} = 2$ and for isotropic refinement, a mesh fraction of 5% was used for the refinement. Fig. 5.30 shows the Mach number contours obtained on an h -adapted mesh (2655 elements) for $p = 3$ discretization. As expected, the adapted mesh has h -refined regions near the trailing edge where there is a singularity, and near the leading edge due to the high gradient of the solution, which would result in more accurate drag coefficient values. Fig. 5.31 plots the convergence of the error in drag coefficient for both isotropic and anisotropic h -adaptation for $p = 3$. From the plot it is clear that the anisotropic h -adaptation is significantly more efficient than the isotropic one. In order to compute the error in the drag coefficient, a reference value of $C_{d,\text{ref}} = 4.686105 \times 10^{-6}$ was used which was obtained from a mesh which was isotropically refined to get a very fine mesh with around 2.8×10^5 degrees of freedom and solution approximation degree of $p = 4$. For anisotropic hp -adaptation, $f_p = 0.2$, $f_h = 0.1$, $r_{\max} = 8$, $c_{\max} = 4$ are used as the adaptation parameters. The Mach number contours and the corresponding polynomial map on an hp -adapted mesh are shown in

Fig. 5.32. As expected, the region near the leading where there is a high gradient of solution got p -refined. Fig. 5.33 compares the polynomial map we got for both isotropic and anisotropic hp -adaptation after few steps. It is interesting to note that the polynomial maps look similar even though they are determined using completely different methods. The polynomial map for anisotropic adaptation is determined using the concept of the minimization of interpolation error bound and that for isotropic adaptation is determined using the jump indicator described in Section 4.4. Fig. 5.34 compares the convergence of the error in the drag coefficient for the isotropic and anisotropic refinement for $p = 1, 2, 3$ and for the anisotropic hp -adaptation. Tables 5.9 and 5.10 tabulates the error convergences achieved in h -adaptations and hp -adaptations, respectively. We observe that the anisotropic adaptations work significantly better than isotropic adaptations in terms of the degree of freedom. Note that there are hardly any anisotropic features, such as boundary layer or shocks, in this test case. So the improved efficiency is not due to the detection of any anisotropic features, but due to remeshing, that includes coarsening, in the case of anisotropic refinement.

(a) Mach number on h -adapted mesh of 2655 elements

Figure 5.30: Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, h -adaptation, $p = 3$.

5.5.4 Inviscid, transonic flow

We consider inviscid transonic flow around the NACA0012 airfoil with a free-stream Mach number $\text{Ma}_\infty = 0.8$, angle of attack $\alpha = 1.25^\circ$. The main features of this flow are the strong shock on the upper side of the airfoil, a weak shock on the lower side and a

(a) Error in drag coefficient vs. degrees of freedom

Figure 5.31: Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, comparison of error convergence for isotropic and anisotropic h -adaptation. 'iso' denotes isotropic adaptation. $p = 3$.

(a) Mach number

(b) Polynomial map

Figure 5.32: Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, hp -Adaptation, 1674 mesh elements, $p = 1, \dots, 6$.

singularity at the trailing edge. The Euler equations are solved with shock capturing term active. Drag coefficient is used as the target functional. Initial mesh with 2155 elements is shown in Fig. 5.8.

For anisotropic h -adaptation the parameters used are $f_h = 0.3$, $r_{max} = 6$, $c_{max} = 4$ and for isotropic refinement, a mesh fraction of 2% was used for the refinement. Fig. 5.35 shows the h -adapted mesh (4394 elements) and the Mach number contours on

Figure 5.33: Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, hp -Adaptation, 1674 mesh elements, $p = 1, \dots, 6$.

Figure 5.34: Inviscid subsonic flow, $\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic adaptation. For hp -adaptation, $p = 1, \dots, 6$.

it for $p = 2$ discretization. As one would expect, both shocks (upper and lower) regions got refined with anisotropic meshes and the shocks are captured sharply. Fig. 5.36 plots the convergence of the error in drag coefficient for both isotropic and anisotropic h -adaptation for $p = 2$. There has been an improvement in convergence rate with anisotropic refinement. To compute the error in the drag coefficient, a reference value

$p = 1$ (iso)		$p = 2$ (iso)		$p = 3$ (iso)	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	5.6040e-04	12930	5.9964e-05	21550	2.6826e-05
8376	2.6187e-04	16434	3.1077e-05	27910	1.4741e-05
10896	1.5387e-04	21636	1.4364e-05	36630	6.7981e-06
14511	9.4862e-05	28434	6.6328e-06	47220	3.1350e-06
19854	5.8041e-05	37086	3.0733e-06	60460	1.4409e-06
26469	3.5286e-05	48702	1.4240e-06	77790	6.5724e-07
35049	2.2298e-05	64638	6.5549e-07		
46761	1.4088e-05	85146	2.9574e-07		
62208	9.1115e-06				

$p = 1$		$p = 2$		$p = 3$	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	5.6040e-04	12930	5.9964e-05	21550	2.6826e-05
7779	2.3122e-04	14574	2.1888e-05	22310	1.0295e-05
8253	1.8251e-04	15396	9.3293e-06	23060	3.2465e-06
8898	1.5347e-04	16296	4.5315e-06	23760	1.0682e-06
9972	1.1954e-04	17442	2.5462e-06	24390	3.7506e-07
11352	9.0938e-05	19008	1.5875e-06	25060	1.1756e-07
15813	5.1528e-05	20268	1.1789e-06	25650	4.3126e-08
18651	3.7729e-05	21594	8.9753e-07		
21708	2.9564e-05	23040	7.0132e-07		
28173	2.0419e-05				
35196	1.4577e-05				
38673	1.2851e-05				

Table 5.9: Inviscid subsonic flow, $M_\infty = 0.5$, $\alpha = 2^\circ$, isotropic and anisotropic h -adaptations for various polynomial approximation degrees. 'iso' denotes isotropic adaptation.

Isotropic		Anisotropic	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	5.6040e-04	6465	5.6040e-04
7113	2.4162e-04	7743	1.2039e-04
7770	1.1311e-04	8761	4.0610e-05
8668	6.2977e-05	9837	1.7011e-05
9821	3.5634e-05	10746	9.8766e-06
11157	1.9829e-05	12159	4.3662e-06
12290	1.0492e-05	13216	1.9148e-06
13595	5.6842e-06	14100	6.4867e-07
14882	3.0247e-06	14809	2.7170e-07
16504	1.6067e-06	15309	1.5773e-07
17900	8.3698e-07	16708	1.2468e-07
19280	4.2732e-07	18477	7.3459e-08
20762	2.0780e-07		
22242	9.2613e-08		
24196	3.5118e-08		

Table 5.10: Inviscid subsonic flow, $M_\infty = 0.5$, $\alpha = 2^\circ$, isotropic and anisotropic hp -adaptations.

of $C_{d,\text{ref}} = 2.265319 \times 10^{-2}$ was used which was obtained from the adapted (isotropic) mesh with around 7.6×10^4 degrees of freedom and solution approximation degree of $p = 2$. For anisotropic hp -adaptation, $f_p = 0.2$, $f_h = 0.1$, $r_{\max} = 6$, $c_{\max} = 4$ are used as the adaptation parameters. Fig. 5.37 shows the hp -adapted mesh, Mach number contours and the corresponding polynomial map. Fig. 5.38 compares the convergence

of the error in the drag coefficient for the isotropic and anisotropic refinement for $p = 1, 2, 3$ and for the anisotropic hp -adaptation. Tables 5.11 and 5.12 tabulates the error convergences achieved in h -adaptations and hp -adaptations, respectively. High order ($p > 1$) does not seem to perform better than the low order one ($p = 1$) for both isotropic and anisotropic adaptations. This is probably due to the presence of singularity and the strong shock attached to the airfoil. However, if we compare isotropic and anisotropic refinement, then certainly there have been improvement in convergence rate with anisotropic refinement.

Figure 5.35: Inviscid transonic flow around the NACA0012, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, $p = 2$, h -adaptation with drag coefficient as target.

(a) Error in drag coefficient vs. degrees of freedom

Figure 5.36: Inviscid transonic flow, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, comparison of error convergence for isotropic and anisotropic h -adaptation, 'iso' denotes isotropic adaptation, $p = 2$.

$p = 1$ (iso)		$p = 2$ (iso)		$p = 3$ (iso)	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	2.4545e-03	12930	1.0472e-03	21550	7.6752e-04
7329	2.0275e-03	14646	7.1941e-04	24320	4.6929e-04
8508	1.7376e-03	16818	5.0755e-04	27340	3.5426e-04
9966	1.4121e-03	19590	3.9122e-04	30840	2.6849e-04
11631	1.0541e-03	22458	2.9451e-04	34820	1.9444e-04
13734	7.8286e-04	26034	2.3233e-04	40440	1.4275e-04
16206	6.0869e-04	30552	1.6632e-04	47690	1.0192e-04
19074	4.6232e-04	35370	1.2069e-04	56440	6.6650e-05
22233	3.6892e-04	41784	8.3090e-05		
25845	2.7682e-04	48870	5.5030e-05		
30285	2.2027e-04				
35412	1.7503e-04				

$p = 1$		$p = 2$		$p = 3$	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	2.7848e-03	12930	1.0473e-03	21550	7.6750e-04
7287	1.8869e-03	14280	7.4745e-04	23560	5.0908e-04
8136	1.3891e-03	15252	5.4707e-04	25170	2.4141e-04
8979	1.1676e-03	16398	3.4332e-04	26680	3.0926e-04
9990	9.6870e-04	17724	2.6826e-04	27580	2.5976e-04
10902	8.0634e-04	18480	1.4803e-04	28480	1.7506e-04
12303	7.3463e-04	20274	9.7100e-05	29230	8.3520e-05
14280	5.3488e-04	21282	5.9500e-05		
17178	4.4201e-04	23634	5.3100e-05		
20610	2.3716e-04	26364	1.5110e-05		

Table 5.11: Inviscid transonic flow, $M_\infty = 0.8$, $\alpha = 1.25^\circ$, isotropic and anisotropic h -adaptations for various polynomial approximation degrees. 'iso' denotes isotropic adaptation.

Figure 5.37: Inviscid transonic flow around the NACA0012, $\text{Ma}_\infty = 0.8$, $\alpha = 1.25^\circ$, hp -adaptation, $p = 1, \dots, 6$.

5.5.5 Inviscid, supersonic flow

We consider inviscid supersonic flow around the NACA0012 airfoil with a free-stream Mach number $\text{Ma}_\infty = 1.5$, angle of attack $\alpha = 0^\circ$. The governing equations are the Euler equations. For this case, a strong and detached bow shock is formed in front of the airfoil. This has to be captured sharply with the help of anisotropic mesh elements. The target functional considered is again the drag coefficient, C_d . The initial mesh with 2155 elements is shown in Fig. 5.8.

For the anisotropic h -adaptation the parameters used are $f_h = 0.3$, $r_{\max} = 6$, $c_{\max} =$

(a) Error in drag coefficient vs. degrees of freedom

Figure 5.38: Inviscid transonic flow, $Ma_\infty = 0.8$, $\alpha = 1.25^\circ$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic adaptation, For hp -adaptation, $p = 1, \dots, 6$.

Isotropic		Anisotropic	
ndof	$ C_d - C_{d,ref} $	ndof	$ C_d - C_{d,ref} $
6465	2.4545e-03	6465	2.7848e-03
6789	1.8505e-03	7311	1.2586e-03
7136	1.5300e-03	8520	7.3040e-04
7883	1.0829e-03	9632	3.3684e-04
9161	9.0369e-04	10884	3.6377e-04
10870	7.4150e-04	11836	2.0213e-04
12460	5.0659e-04	12942	1.4937e-04
13848	4.0003e-04	13994	1.6829e-04
15423	3.2277e-04	15077	6.7820e-05
17230	2.0706e-04	16374	8.7390e-05
18729	1.5895e-04	17829	9.3760e-05
20412	1.3085e-04	19718	9.4730e-05
21781	8.1610e-05	22044	7.3600e-05
23902	7.1240e-05	24360	3.2310e-05
26264	5.1330e-05		

Table 5.12: Inviscid transonic flow, $M_\infty = 0.8$, $\alpha = 1.25^\circ$, isotropic and anisotropic hp -adaptations.

4 and for isotropic refinement, a mesh fraction of 2% was used for the refinement. Fig. 5.39 shows the h -adapted mesh (7380 elements) and the Mach number contours on it for $p = 2$ discretization. The whole mesh got aligned with the flow features, especially near the bow shock region. As expected, the mesh elements outside the Mach cone get coarsened since they do not contribute to the error in the drag coefficient. Fig. 5.41 shows the anisotropic mesh near the bow-shock close to the airfoil and also near the upper end of the bow-shock. We can clearly see the stretched mesh elements. Fig. 5.40 plots the convergence of the error in drag coefficient for both isotropic and anisotropic h -adaptation for $p = 3$. There has been an improvement in convergence rate

with anisotropic refinement. To compute the error in the drag coefficient, a reference value of $C_{d,\text{ref}} = 9.629886 \times 10^{-2}$ was used which was obtained from an isotropically adapted fine mesh with around 4.1×10^5 degrees of freedom for solution approximation of degree ($p = 2$). For anisotropic hp -adaptation, $f_p = 0.1$, $f_h = 0.2$, $r_{\max} = 6$, $c_{\max} = 6$ are used as the adaptation parameters. Fig. 5.42 shows the hp -adapted mesh, Mach number contours and the corresponding polynomial map. The region outside the Mach cone got coarser mesh and lower polynomial degree which is consistent with the physics of the flow. This region is not 'aware' of the presence of the airfoil due to the supersonic nature of the flow. As expected, the region between the airfoil and the shock got higher polynomial degree. Fig. 5.43 compares the convergence of the error in the drag coefficient for the isotropic and anisotropic refinement for $p = 1, 2, 3$ and for the anisotropic hp -adaptation. Tables 5.13 and 5.14 tabulates the error convergences achieved in h -adaptations and hp -adaptations, respectively. For both isotropic and anisotropic refinement, $p = 2$ gives better convergence rate than $p = 1$. With higher p ($p > 2$) the convergence does not seem to get better, probably due to the low regularity of the dominant flow features. This is also reflected in the hp -adaptation, where the overall convergence rate is not really superior to pure h -adaptation with $p = 2$.

Figure 5.39: Inviscid supersonic flow around the NACA0012, $\text{Ma}_\infty = 1.5$, $\alpha = 0^\circ$, $p = 2$, h -adaptation with drag coefficient as target.

(a) Error in drag coefficient vs. degrees of freedom

Figure 5.40: Inviscid supersonic flow, $Ma_\infty = 1.5$, $\alpha = 0^\circ$, comparison of error convergence for isotropic and anisotropic adaptations, 'iso' denotes isotropic h -refinement. $p = 3$.

(a) Mesh elements near the bow shock, h -adapted mesh - 7380 elements (b) Mesh elements near the upper end of the bow shock

Figure 5.41: Inviscid supersonic flow around the NACA0012, $Ma_\infty = 1.5$, $\alpha = 0^\circ$, $p = 2$, h -adaptation with drag coefficient as target.

Figure 5.42: Inviscid supersonic flow around the NACA0012, $M_\infty = 1.5$, $\alpha = 0^\circ$, hp -adaptation with drag coefficient as target, $p = 1, \dots, 6$.

(a) Error in drag coefficient vs. degrees of freedom

Figure 5.43: Inviscid supersonic flow, $M_\infty = 1.5$, $\alpha = 0^\circ$, comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic adaptation, for hp -adaptation, $p = 1, \dots, 6$.

$p = 1$ (iso)		$p = 2$ (iso)		$p = 3$ (iso)	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	2.4545e-03	12930	1.0472e-03	21550	7.6752e-04
7329	2.0275e-03	14646	7.1941e-04	24320	4.6929e-04
8508	1.7376e-03	16818	5.0755e-04	27340	3.5426e-04
9966	1.4121e-03	19590	3.9122e-04	30840	2.6849e-04
11631	1.0541e-03	22458	2.9451e-04	34820	1.9444e-04
13734	7.8286e-04	26034	2.3233e-04	40440	1.4275e-04
16206	6.0869e-04	30552	1.6632e-04	47690	1.0192e-04
19074	4.6232e-04	35370	1.2069e-04	56440	6.6650e-05
22233	3.6892e-04	41784	8.3090e-05		
25845	2.7682e-04	48870	5.5030e-05		
30285	2.2027e-04				
35412	1.7503e-04				

$p = 1$		$p = 2$		$p = 3$	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	1.0504e-03	21550	1.9566e-04	21550	1.9566e-04
9135	3.2842e-04	23910	7.6600e-05	23910	7.6600e-05
12099	2.0108e-04	26130	6.9340e-05	26130	6.9340e-05
16119	1.2217e-04	28990	4.1640e-05	28990	4.1640e-05
21105	6.3700e-05	31850	2.5250e-05	31850	2.5250e-05
25824	4.8070e-05	35190	1.4110e-05	35190	1.4110e-05
33024	3.1610e-05	41360	5.8099e-06	41360	5.8099e-06
39279	2.4550e-05	45920	4.8200e-06	45920	4.8200e-06
50940	2.4099e-06			50940	2.4099e-06

Table 5.13: Inviscid supersonic flow, $M_\infty = 1.5$, $\alpha = 0^\circ$, isotropic and anisotropic h -adaptations for various polynomial degree approximations. 'iso' denotes isotropic adaptation.

Isotropic		Anisotropic	
ndof	$ C_d - C_{d,\text{ref}} $	ndof	$ C_d - C_{d,\text{ref}} $
6465	1.0504e-03	6465	1.0504e-03
9873	1.2803e-04	7857	1.7449e-04
15357	6.7400e-06	10013	6.4170e-05
23808	1.6460e-05	12465	3.5750e-05
36829	6.9100e-06	14929	1.5350e-05
58281	1.0600e-05	17598	1.8020e-05
96432	4.6100e-06	19902	1.3540e-05
		22901	7.9000e-06
		27792	4.9500e-06
		32202	7.6600e-06
		34674	5.3700e-06
		37910	4.2400e-06
		42917	3.3100e-06
		43843	3.3599e-06
		50414	2.6100e-06
		55448	2.6600e-06

Table 5.14: Inviscid supersonic flow, $M_\infty = 1.5$, $\alpha = 0^\circ$, isotropic and anisotropic hp -adaptations.

6 Conclusions and Outlook

Efficient adjoint-based hp -adaptation methodologies on isotropic and anisotropic meshes have been presented for the recently developed high-order Hybridized discontinuous Galerkin solver for (nonlinear) convection-diffusion problems. hp -adaptation on isotropic meshes is based on the spatial error distribution, for a given target functional, given by the adjoint error estimator and the solution regularity given by a regularity indicator. The methodology has been tested for various test cases including a scalar convection-diffusion case with a strong boundary layer, and inviscid and viscous flow cases around NACA 0012 airfoil. In all the test cases presented, hp -adaptivity was found to be more efficient than pure h -adaptivity simulations in terms of the number of degrees of freedom required to achieve same error level in the target functional.

With the aim of making the adaptations more efficient, anisotropic meshes — based on the concept of mesh-metric duality — are generated that are expected to efficiently resolve the anisotropic features of the solution. To define the desired anisotropic mesh elements, the refinement strategy based on an interpolation error estimate, due to Dolejsi [30], has been extended by incorporating an adjoint-based error estimate. Using the two error estimates the size and the shape of the triangular mesh elements on the desired mesh have been determined in the case of pure h -adaptation, and also polynomial degree distribution in the case of hp -adaptivity. The effectiveness of the adaptation methodology is demonstrated using scalar convection-diffusion test cases and various flow cases around the NACA 0012 airfoil. Both h - and hp -adaptations on anisotropic meshes are significantly more efficient than the adaptations on isotropic meshes in terms of the number of degrees of freedom required to achieve the same level of error. The anisotropic meshes could resolve the flow features like shocks and flow recirculation very well. In the case of pure h -adaptations, increasing the polynomial degree could also make the adaptations more efficient. In general, hp -adaptations on anisotropic mesh performed the best and gave promising results. However, for transonic and supersonic inviscid flows, even though the shocks and other flow features were resolved well, overall gain in efficiency in reducing the target error was not significant when polynomial degree was increased as well as when hp -refinement was used.

Future work should aim at exploring ways to tackle flows with shocks and to make hp-adaptivity to work more efficiently for such cases. In order to perform any realistic flow calculation, the methodology and the mesh generation capabilities should be extended to three dimensions. Extending the theoretical framework of the methodology to three dimensions is straightforward, but challenges are expected in the robust generation of anisotropic meshes for complex geometries. Further, handling realistic three-dimensional problems would require parallelisation of the code. Extending the framework to unsteady problems is far from being straightforward due to a number of reasons. The mesh lags behind the solution and hence the adapted mesh is no longer adapted for the subsequent time steps. At each remeshing step, the solution has to be interpolated to the new adapted mesh, and the error due to this solution transfer may get accumulated throughout the simulation. Metric-based anisotropic mesh generation, however, has demonstrated its potential and efficiency in academic problems, and there is a lot of scope for its further advancement into real world problems in engineering and industry.

A Appendix

A.1 High Order Derivatives of Dubiner Basis Functions on Triangles

A.1.1 Dubiner polynomials

If (ξ, η) is the coordinates of the reference triangle, then the Dubiner basis [32] is defined in the transformed coordinate system (a, b) given as

$$a = \frac{2\eta}{1-\xi}, \quad b = 2\xi - 1. \quad (\text{A.1})$$

For the polynomial approximation degree of m , the Dubiner basis, ψ in the coordinate system (a, b) can be written as the product of two one-dimensional functions, ψ^a and ψ^b , as given below:

$$\psi_{(q,l)} = \psi_q^a(a) \psi_{ql}^b(b), \quad (\text{A.2})$$

where $q + l \leq m$. For example, for $p = 2$, the six basis functions can be written in the form of pascal triangles as follows:

$$\begin{array}{ccccc} & & \psi_{(0,0)} & & \\ & & & & \\ & \psi_{(1,0)} & & \psi_{(0,1)} & \\ & & & & \\ \psi_{(2,0)} & & \psi_{(1,1)} & & \psi_{(0,2)} \end{array}$$

The functions ψ^a and ψ^b are given as,

$$\psi_q^a(a) = P_q^{(0,0)}(a) \quad (\text{A.3})$$

$$\psi_{ql}^b(b) = \left(\frac{1-b}{2} \right)^q P_l^{(2q+1,0)}(b), \quad (\text{A.4})$$

where $P_q^{(\alpha,\beta)}(x)$ are the Jacobi polynomials of degree q , which are orthogonal with respect to the weight $w = (1-x)^\alpha(1+x)^\beta$ on the interval $[-1, 1]$. The first two

Jacobi polynomials ($q = 0, 1$) are given as

$$P_0^{(\alpha, \beta)} = 1 \quad (\text{A.5})$$

$$P_1^{\alpha, \beta} = (1.0 + 0.5(\alpha + \beta))x + 0.5(\alpha - \beta), \quad (\text{A.6})$$

and the rest can be found from the recursive relation,

$$\begin{aligned} & 2q(q + \alpha + \beta)(2q + \alpha + \beta - 2)P_q^{(\alpha, \beta)}(x) \\ &= (2q + \alpha + \beta - 1) \left\{ (2q + \alpha + \beta)(2q + \alpha + \beta - 2)x + \alpha^2 - \beta^2 \right\} P_{q-1}^{(\alpha, \beta)}(x) \\ & - 2(q + \alpha - 1)(q + \beta - 1)(2q + \alpha + \beta)P_{q-2}^{(\alpha, \beta)}(x), \quad q = 2, 3, \dots \end{aligned} \quad (\text{A.7})$$

The k -th derivative of the Jacobi polynomial of degree q can be derived to be

$$\frac{d^k}{dx^k} P_q^{(\alpha, \beta)}(x) = \frac{(\alpha + \beta + q + 1 + k)!}{2^k(\alpha + \beta + q + 1)!} P_{q-k}^{(\alpha+k, \beta+k)}(x). \quad (\text{A.8})$$

A.1.2 Evaluation of the n -th derivatives of the Dubiner polynomials

We can derive the n -th derivatives of the Dubiner basis ψ as.

$$\begin{aligned} \frac{\partial^n \psi}{\partial \xi^r \partial \eta^{n-r}} &= \frac{\partial^r}{\partial \xi^r} \left(\frac{\partial^{n-r} \psi}{\partial \eta^{n-r}} \right) \\ &= \frac{\partial^r}{\partial \xi^r} \left(\frac{\partial^{n-r} \psi^a \psi^b}{\partial \eta^{n-r}} \right) \\ &= \frac{\partial^r}{\partial \xi^r} \sum_{k=0}^{n-r} \left[\binom{n-r}{k} \frac{\partial^{n-r-k} \psi_a}{\partial \eta^{n-r-k}} \frac{\partial^k \psi_b}{\partial \eta^k} \right] \\ &= \sum_{l=0}^r \left[\binom{r}{l} \sum_{l=0}^{n-r} \left[\binom{n-r}{k} \underbrace{\frac{\partial^{r-l}}{\partial \xi^{r-l}} \left(\frac{\partial^{n-r-k} \psi_a}{\partial \eta^{n-r-k}} \right)}_{T_1} \underbrace{\frac{\partial^l}{\partial \xi^l} \left(\frac{\partial^k \psi_b}{\partial \eta^k} \right)}_{T_2} \right] \right] \end{aligned} \quad (\text{A.9})$$

where $r = 0, 1, \dots, n$ give the $n + 1$ n -th derivatives of ψ , and

$$\binom{r}{k} := \frac{r!}{k!(r-k)!}. \quad (\text{A.10})$$

The expressions T_1 and T_2 in Eq. (A.9) can be computed as

$$\begin{aligned}
 T_1 &= \frac{\partial^{r-l}}{\partial \xi^{r-l}} \left(\frac{\partial^{n-r-k} \psi_a}{\partial \eta^{n-r-k}} \right) \\
 &= \frac{\partial^{r-l}}{\partial \xi^{r-l}} \left(\left(\frac{\partial a}{\partial \eta} \right)^{n-r-k} \frac{\partial^{n-r-k} \psi_a}{\partial a^{n-r-k}} \right) \\
 &= \sum_{p=0}^{r-l} \left[\binom{r-l}{p} \underbrace{\frac{\partial^{r-l-p}}{\partial \xi^{r-l-p}} \left(\frac{\partial a}{\partial \eta} \right)^{n-r-k}}_{T_3} \underbrace{\frac{\partial^p}{\partial \xi^p} \left(\frac{\partial^{n-r-k} \psi_a}{\partial a^{n-r-k}} \right)}_{T_4} \right], \quad (\text{A.11})
 \end{aligned}$$

$$T_2 = \frac{\partial^l}{\partial \xi^l} \left(\frac{\partial^k \psi_b}{\partial \eta^k} \right) = \left(\frac{\partial b}{\partial \xi} \right)^l \left(\frac{\partial b}{\partial \eta} \right)^k \underbrace{\frac{\partial^{l+k} \psi_b}{\partial b^{l+k}}}_{T_5}. \quad (\text{A.12})$$

From Eq. (A.1),

$$\frac{\partial a}{\partial \eta} = \frac{2}{1-\xi}. \quad (\text{A.13})$$

Using the above equation (Eq. (A.13)), the expression T_3 in Eq. (A.11) can be evaluated as

$$T_3 = \frac{\partial^{r-l-p}}{\partial \xi^{r-l-p}} \left(\frac{\partial a}{\partial \eta} \right)^{n-r-k} = 2^{n-r-k} \frac{(n-k-l-p-1)!}{(n-r-k-1)!} (1-\xi)^{-(n-l-p-k)}. \quad (\text{A.14})$$

Using Hoppe's formula [75, 48], the expression T_4 in Eq. (A.11) can be expressed as

$$T_4 = \frac{\partial^p}{\partial \xi^p} \left(\frac{\partial^{n-r-k} \psi_a}{\partial a^{n-r-k}} \right) = \sum_{i=0}^p A_p^i \frac{\partial^i}{\partial a^i} \underbrace{\left(\frac{\partial^{n-r-k} \psi_a}{\partial a^{n-r-k}} \right)}_{T_8} \quad (\text{A.15})$$

where,

$$A_p^i = \sum_{m=0}^i \frac{(-1)^{i-m}}{m!(i-m)!} a^{i-m} \underbrace{\frac{d^p a^m}{d \xi^p}}_{T_7}. \quad (\text{A.16})$$

The expression T_7 in Eq. (A.16) can be expressed, using Hoppe's formula, as

$$T_7 = \frac{d^p a^j}{d \xi^p} = \sum_{i=0}^p A_p^i \frac{d^i(a^j)}{da^i}, \quad (\text{A.17})$$

where

$$A_p^i = \sum_{k=0}^i \frac{(-1)^{i-k}}{k!(i-k)!} \phi^{i-k} \frac{d^p \phi^k}{d\xi^p}, \quad \phi = a + 1 = \frac{2\eta}{1-\xi}. \quad (\text{A.18})$$

Substituting

$$\frac{d^p \phi^k}{d\xi^p} = (2\eta)^k \frac{(k+p-1)!}{(k-1)!} (1-\xi)^{-(-k-p)} \quad (\text{A.19})$$

in Eq. (A.18), and

$$\frac{d^i(a^j)}{da^i} = \frac{j!}{(j-i)!} a^{j-i} \quad (\text{A.20})$$

in Eq. (A.17), we can evaluate the expression T_7 . The expression T_8 can be evaluated using Eq. (A.3) and Eq. (A.8). With these, we can evaluate T_4 . If Eq. (A.4) is rewritten as

$$\psi^b = G(b) \hat{P}(b), \quad (\text{A.21})$$

where,

$$G(b) := \left(\frac{1-b}{2} \right)^q, \quad (\text{A.22})$$

$$\hat{P}(b) := P_l^{(2q+1,0)}(b), \quad (\text{A.23})$$

then the expression T_5 in Eq. (A.12) can be expressed as

$$T_5 = \frac{\partial^{l+k} \psi_b}{\partial b^{l+k}} = \frac{\partial^{l+k}}{\partial b^{l+k}} \left(G(b) \hat{P}(b) \right) \quad (\text{A.24})$$

$$= \sum_{q=0}^{l+k} \left[\binom{l+k}{q} \underbrace{\frac{\partial^{l+k-q} G(b)}{\partial b^{l+k-q}}}_{T_6} \underbrace{\frac{\partial^q \hat{P}(b)}{\partial b^q}}_{T_9} \right]. \quad (\text{A.25})$$

The expression T_6 in Eq. (A.25) can be computed as

$$T_6 = \frac{\partial^{l+k-q} G(b)}{\partial b^{l+k-q}} = 0.5^N \frac{N!}{(N-l-k+q)!} (1-b)^{N-l-k+q} (-1)^{l+k-q} \quad (\text{A.26})$$

The expression T_9 in Eq. (A.25) can be evaluated using Eq. (A.23) and Eq. (A.8). With these, we can find the expressions T_1 and T_2 , and hence the n -th derivatives of the Dubiner basis, ψ using Eq. (A.9).

Bibliography

- [1] *IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach*. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, 2015.
- [2] R. Abgrall and D. De Santis. High order residual distribution scheme for Navier-Stokes equations. AIAA Conference Paper 2011-3231, American Institute of Aeronautics and Astronautics, June 2011.
- [3] L. Adrien and F. Alauzet. Continuous mesh framework part I: Well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, 2011.
- [4] L. Adrien and F. Alauzet. Continuous mesh framework part II: Validations and applications. *SIAM Journal on Numerical Analysis*, 49(1):61–86, 2011.
- [5] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Num. Anal.*, 39(5):1749–1779, 2002.
- [6] D.N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Num. Anal.*, 19(4), 1982.
- [7] A. Balan, G. May, and J. Schöberl. A stable high-order Spectral Difference method for hyperbolic conservation laws on triangular elements. *Journal of Computational Physics*, 231(5):2359–2375, March 2012.
- [8] A. Balan, M. Woopen, and G. May. Adjoint-based hp-adaptation for a class of high-order hybridized finite element schemes for compressible flows. AIAA Conference Paper 10.2514/6.2013-2938, American Institute of Aeronautics and Astronautics Conference, June 2013.
- [9] A. Balan, M. Woopen, and G. May. Hp-adaptivity on anisotropic meshes for hybridized discontinuous galerkin scheme. AIAA Conference Paper 10.2514/6.2015-2606, American Institute of Aeronautics and Astronautics Conference, 2015.

- [10] A. Balan, M. Woopen, and G. May. Adjoint-based hp-adaptivity on anisotropic meshes for high-order compressible flow simulations. *Computers and Fluids*, doi:10.1016/j.compfluid.2016.03.029, 2016.
- [11] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.
- [12] T. Barth and M. Ohlberger. Finite volume methods: Foundation and analysis. In E. Stein, R. De Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics, Part 1: Fundamentals*, chapter 15. Wiley, 2004.
- [13] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131(2):267–279, 1997.
- [14] 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, 12 1997.
- [15] 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.
- [16] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10(1):1–102, 2001.
- [17] M. A. Ceze and K. J. Fidkowski. An anisotropic hp-adaptation framework for functional prediction. *American Institute of Aeronautics and Astronautics Journal*, 51:492–509, 2013.
- [18] B. Cockburn. Discontinuous Galerkin methods. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 83(11):731–754, 2003.
- [19] B. Cockburn and P. A. Gremaud. Error estimate for finite element methods for nonlinear conservation laws. *SIAM J. Num. Anal.*, 33:522–554, 1996.
- [20] B. Cockburn, S. Hou, and S. W. Shu. The Runge–Kutta local projection Discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54(190):545–581, 1990.

- [21] B. Cockburn, G. E. Karniadakis, and C. W. Shu. The development of Discontinuous Galerkin methods. In B. Cockburn, G. E. Karniadakis, and C. W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 3–50. Springer, 2000.
- [22] B. Cockburn and S. Y. Lin. TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. *J. Comp. Phys.*, 84:90–113, 1989.
- [23] B. Cockburn and C. W. Shu. TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comp.*, 52(186):411–435, 1988.
- [24] B. Cockburn and C. W. Shu. The local Discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.
- [25] B. Cockburn and C. W. Shu. The Runge-Kutta Discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comp. Phys.*, 141:199–224, 1998.
- [26] B. Cockburn and C. W. Shu. Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comp.*, 16(3):173–261, 2001.
- [27] B. Cockburn D. N. Arnold, F. Brezzi and D. Marini. Discontinuous Galerkin method for elliptic problems. In *Discontinuous Galerkin methods. Theory, computation and applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 89–101, 2000.
- [28] V. Dolejší. Analysis and application of the IIPG method to quasilinear non-stationary convection–diffusion problems. *Journal of Computational and Applied Mathematics*, 222(2):251 – 273, 2008.
- [29] V. Dolejší. hp-dgfem for nonlinear convection-diffusion problems. *Mathematics and Computers in Simulation*, 87:87–118, 2013.
- [30] V. Dolejší. Anisotropic hp-adaptive method based on interpolation error estimates in the l_q -norm. *Applied Numerical Mathematics*, 82:80–114, 2014.

- [31] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic galerkin methods. In *Lecture Notes in Physics*, volume 58, Berlin, 1976. Springer-Verlag.
- [32] M. Dubiner. Spectral methods on triangles and other domains. *J. Sci. Comput.*, 6(4):345–390, 1991.
- [33] H. Egger and J. Schöberl. A hybrid mixed Discontinuous Galerkin finite-element method for convection–diffusion problems. *IMA Journal of Numerical Analysis*, 30(4):1206–1234, 2010.
- [34] K. J. Fidkowski. *A simplex cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations*. PhD thesis, MIT, 2007.
- [35] K. J. Fidkowski. A local sampling approach to anisotropic metric-based mesh optimization. AIAA Conference Paper 10.2514/6.2016-0835, American Institute of Aeronautics and Astronautics, 2016.
- [36] K. J. Fidkowski and D. L. Darmofal. A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations. *J. Comp. Phys.*, 225:1653–1672, 2007.
- [37] K. J. Fidkowski and D. L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.
- [38] K. J. Fidkowski and D. L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *American Institute of Aeronautics and Astronautics Journal*, 49(4):673–694, 2011.
- [39] 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. *Journal of Computational Physics*, 207(1):92 – 113, 2005.
- [40] DLR Germany. Taubench - ipacs, <http://www.ipacs-benchmark.org>.
- [41] M. B. Giles and N. A. Pierce. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. AIAA Conference Paper 97-1850, American Institute of Aeronautics and Astronautics, 1997.

-
- [42] E. Godlewski and P. A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, volume 118 of *Applied Mathematical Sciences*. Springer Verlag, 1996.
- [43] R. Hartmann. Adjoint consistency analysis of Discontinuous Galerkin discretizations. *SIAM J. Numer. Anal.*, 45(6):2671–2696, 2007.
- [44] R. Hartmann. Error estimation and adjoint based refinement for an adjoint consistent DG discretisation of the compressible Euler equations. *Int. J. Comput. Sci. Math.*, 1(2/3/4):207–220, 2007.
- [45] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183:508–532, 2002.
- [46] 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.
- [47] F. Hecht. Bamg: Bidimensional anisotropic mesh generator. Technical report, INRIA-Rocquencourt, France, 2006.
- [48] R. Hoppe. Über independente Darstellung der höheren Differentialquotienten. *Math. Ann.*, 4:85–87, 1871.
- [49] P. Houston, C. Schwab, and E. Suli. Discontinuous hp-finite element methods for advection-diffusion-reaction problems. *SIAM Journal on Numerical Analysis*, 39(6):2133–2163, 01 2002.
- [50] P. Houston and E. Süli. hp-adaptive Discontinuous Galerkin finite element methods for first order hyperbolic problems. *SIAM J. Sci. Comput.*, 23(4):1226–1252, 2001.
- [51] H. T. Huynh. A flux reconstruction approach to high-order schemes including Discontinuous Galerkin methods. AIAA Conference Paper 07-4079, American Institute of Aeronautics and Astronautics, 2007.
- [52] J. Jaffre, C. Johnson, and A. Szepessy. Convergence of the Discontinuous Galerkin finite element method for hyperbolic conservation laws. *Math. Models Meth. Appl. Sci.*, 5(3):367–386, 1995.

- [53] G. Jiang and C. W. Shu. On a cell entropy inequality for Discontinuous Galerkin methods. *Math. Comput.*, 62(206):531–538, 1994.
- [54] G. E. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2nd edition, 2005.
- [55] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Van, and K. Sorensen. *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*. Springer, 2010.
- [56] T. Leicht and R. Hartmann. Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations. *Journal of Computational Physics*, 229(19):7344 – 7360, 2010.
- [57] T. Leicht and R. Hartmann. Goal-oriented error estimation and hp-adaptive mesh refinement for aerodynamic flows. Number 2011-212. American Institute of Aeronautics and Astronautics, January 2011.
- [58] A. Loseille and F. Alauzet. Continuous mesh model and well-posed continuous interpolation error estimation. 2009 RR-6846, INRIA, France.
- [59] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3d steady Euler equations. *J. Comput. Phys.*, 229(8):2866–2897, April 2010.
- [60] P. Marinov and G. Lebon. A generalized fourier’s law in thermodynamically simple materials of the differential type. *Acta Mechanica*, 22(3):257–265, 1975.
- [61] D. J. Mavriplis and A. Jameson. Multigrid solution of the navier-stokes equations on triangular meshes. *AIAA Journal*, 28:1415–1425, 1990.
- [62] G. May and A. Jameson. A Spectral Difference method for the Euler and Navier-Stokes equations on unstructured meshes. AIAA Conference Paper 06-0304, American Institute of Aeronautics and Astronautics, 2006.
- [63] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable Discontinuous Galerkin method for linear convection–diffusion equations. *Journal of Computational Physics*, 228(9):3232 – 3254, 2009.
- [64] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable Discontinuous Galerkin method for nonlinear convection–diffusion equations. *Journal of Computational Physics*, 228(23):8841 – 8855, 2009.

-
- [65] N.C. Nguyen and J. Peraire. An adaptive shock capturing hdg method for compressible flows. *20th AIAA computational fluid dynamics conference, AIAA 2011-3060*, 2011.
- [66] T. A. Oliver. *A high-order adaptive Discontinuous Galerkin finite element method for the Reynolds-averaged Navier-Stokes Equations*. PhD thesis, MIT, 2008.
- [67] J. Peraire, N. C. Nguyen, and B. Cockburn. A hybridizable Discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations. AIAA Conference Paper 2010-363, American Institute of Aeronautics and Astronautics, 2010.
- [68] J. Peraire and P. O. Persson. The compact Discontinuous Galerkin method for elliptic problems. *SIAM J. Sci. Comput.*, 30:1806–1824, 2008.
- [69] P. O. Persson and J. Peraire. Sub-cell shock capturing for Discontinuous Galerkin methods. AIAA Conference Paper 06-0112, American Institute of Aeronautics and Astronautics, 2006.
- [70] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [71] J. Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, 1997.
- [72] J. Schütz and G. May. An adjoint consistency analysis for a class of hybrid mixed methods. *IMA Journal of Numerical Analysis*, 34(3):1222–1239, 2013.
- [73] J. Schütz and G. May. A hybrid mixed method for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 240:58–75, 2013.
- [74] L. Shi and Z. J. Wang. Adjoint-based error estimation and hp-adaptation for the high order CPR method. AIAA Conference Paper 2013-0999, American Institute of Aeronautics and Astronautics, 2013.
- [75] R. M. Slevinsky and H. Safouhi. New formulae for higher order derivatives and applications. *J. Comput. Appl. Math.*, 233(2):405–419, 2009.
- [76] W. Sutherland. The viscosity of gases and molecular force. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 36(223):507–531, 1893.

- [77] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 1999.
- [78] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187:22–46, 2003.
- [79] P. Vincent, P. Castonguay, and A. Jameson. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47(1):50–72, 2011-04-01.
- [80] L. Wang and D. J. Mavriplis. Adjoint-based hp adaptive discontinuous Galerkin methods for the 2d compressible Euler equations. *Journal of Computational Physics*, 228:7643–7661, 2009.
- [81] Z. J. Wang, editor. *Adaptive High-Order Methods in Computational Fluid Dynamics*. World Scientific Publishing Company, 2011.
- [82] Z. J. Wang. High-order CFD methods : Current status and perspective. *International Journal for Numerical Methods in Fluids*, pages 1–42, 2012.
- [83] M. Woopen, A. Balan, and G. May. A hybridized Discontinuous Galerkin method for three-dimensional compressible flow problems. AIAA Conference Paper 2014-0938, American Institute of Aeronautics and Astronautics, 2014.
- [84] M. Woopen, A. Balan, and G. May. A hybridized Discontinuous Galerkin method for turbulent compressible flow problems. AIAA Conference Paper 2014-2783, American Institute of Aeronautics and Astronautics, 2014.
- [85] M. Woopen, A. Balan, G. May, and J. Schütz. A comparison of hybridized and standard DG methods for target based hp-adaptive simulation of compressible flow. *Computers & Fluids*, 98:3–16, 2014.
- [86] M. Woopen, G. May, and J. Schütz. Adjoint-based error estimation and mesh adaptation for hybridized Discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 76(11):811–834, 2014.
- [87] M. Yano and D. L. Darmofal. An optimization-based framework for anisotropic simplex mesh adaptation. *J. Comput. Phys.*, 231(22):7626–7649, September 2012.