

Mesh adaptation strategies for compressible flows using a high-order spectral/ hp element discretisation

by

Dirk Ekelschot

Supervisors: Dr. Joaquim Peiró and Prof. Spencer Sherwin

Department of Aeronautics
Imperial College London
Prince Consort Road
London SW7 2AZ

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

2016

Declaration of originality

This thesis presents the work that I have carried out at Imperial College London. It has not been submitted to any other institute and all sources used in this work are properly referenced in the bibliography. Proper acknowledgement is given to figures and data that are not my own work.

Dirk Ekelschot

Copyright declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgements

Coming from a background in experimental aerodynamics, I dived into the realm of computational fluid dynamics when I started my PhD at Imperial College London. The learning curve was steep in the beginning but luckily I received great guidance from the very start of the project.

In particular, I would like to thank my supervisor Dr. Joaquim Peiró for guiding me during my research over the last few years. I very much enjoyed the weekly meetings that we had either over coffee or in his office. I really value his technical insight, his critical approach and enthusiasm to help improve the understanding and capabilities of his students. Furthermore, he always made a significant amount of time to discuss both research and other activities which I appreciated very much.

Furthermore I would like to thank Prof. Spencer Sherwin for giving me the opportunity to do my research in the framework of Nektar++ and for helping me together with Dr. Joaquim Peiró to overcome the difficult start of the project.

I would also like to thank the entire Nektar++ group and in particular Dr. David Moxey for his help during the very lengthy debugging sessions. Although at times these sessions were stressful, it was always a pleasure to work with him and I have learnt a great deal about how to tackle certain problems related to logic in coding.

This work was supported by the Seventh Framework Programme of the European Commission as part of the ITN project called *Advances in Numerical and Analytical tools for DETached* (ANADE) flow prediction. Therefore, I would like to thank the ANADE fellows that I have gotten to know during the visits and secondments at Universidad Politécnica de Madrid and the University of Cambridge. Furthermore, I would also like to thank Dr. Carlos Lozano for providing me with data that enabled me to compare my adjoint solutions. To my office mates Bastien, Thibault, Justin, Jean, Olga, Laurent, thanks for all the great times in the office, at the Union and outside of the Imperial College campus.

Finally, I would like to thank Tatiana for supporting me over the last three years and for allowing me to take this opportunity. I am so happy to see that we considered both Delft and London to be our home for this period of time.

Contents

List of Figures	10
List of Tables	11
List of Symbols	16
1 Introduction	18
1.1 Mesh adaptation for compressible flows	20
1.1.1 Adjoint-based mesh adaptation	23
1.1.2 Mesh deformation	25
1.2 Project objective	27
1.3 Thesis outline	28
2 The governing and adjoint equations	29
2.1 Governing equations	29
2.1.1 Compressible Euler equations	29
2.1.2 Compressible Navier-Stokes equations	30
2.1.3 Boundary conditions	31
2.2 Adjoint equations	36
2.2.1 Adjoint Euler equations	40
2.2.2 Boundary conditions for the adjoint Euler equations	42
2.2.3 Adjoint Navier-Stokes equations	43
2.2.4 Boundary conditions for the adjoint Navier-Stokes equations	45
3 Discontinuous Galerkin spectral/<i>hp</i> element method	49
3.1 Spatial discretisation	49
3.2 Discretisation of the governing equations	54
3.2.1 Compressible Euler equations	55
3.2.2 Compressible Navier-Stokes equations	57
3.2.3 Boundary conditions	58
3.3 Artificial diffusion	62
3.4 Variable polynomial order	65

3.5	Discretisation of the adjoint equations	68
3.5.1	Adjoint Euler equations	68
3.5.2	Adjoint Navier-Stokes equations	69
3.5.3	Adjoint boundary conditions	70
3.6	Verification of the adjoint solver	71
3.6.1	Quasi one-dimensional subsonic inviscid flow through a converging-diverging nozzle ($Ma = 0.12$).	72
3.6.2	Two-dimensional transonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$)	75
4	p-adaptation for compressible flows	80
4.1	p -adaptation using the discontinuity sensor	81
4.1.1	Three-dimensional subsonic inviscid flow past a sphere ($Ma = 0.5$).	82
4.2	p -adaptation using goal-based error estimation	85
4.2.1	Two-dimensional subsonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5.0^\circ$).	88
4.2.2	Two-dimensional subsonic laminar flow past a NACA 0012 aerofoil ($Ma = 0.1$, $Re = 5000$, $\alpha = 2.0^\circ$).	91
4.2.3	Two-dimensional transonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).	93
4.2.4	Three-dimensional subsonic inviscid flow past an ellipsoid of revolution ($Ma = 0.5$, $\alpha = 3.0^\circ$).	97
5	Mesh deformation using a thermo-elastic analogy	101
5.1	The linear thermo-elastic equations	103
5.2	Mesh deformation for mesh quality control	104
5.2.1	The Jacobian as an isotropic temperature term	104
5.2.2	The metric tensor as an anisotropic temperature term	105
5.2.3	Rotating circular cylinder	107
5.2.4	Moving sphere in a cube	109
5.3	r -adaptation for shock waves	110
5.3.1	Formulation of the thermal stress term	113
5.3.2	Two-dimensional transonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).	115
5.3.3	Two-dimensional supersonic inviscid flow past a forward facing step ($Ma = 3.0$).	118
6	Conclusions and recommendations for future work	124
6.1	Adjoint-based p -adaptation	124

6.1.1	Conclusions	125
6.1.2	Recommendations	127
6.2	Mesh deformation and r -adaptation using a thermo-elastic analogy	128
6.2.1	Conclusions	128
6.2.2	Recommendations	129
Bibliography		130
A Jacobians of the convective and diffusive fluxes		140
A.1	Linearisation of the convective fluxes	140
A.2	Linearisation of the diffusive fluxes	141
B Adjoint Roe-Riemann flux function		143
C Adjoint boundary conditions for converging-diverging nozzle flow		145
C.1	Adjoint boundary condition at the inflow	145
C.2	Adjoint boundary condition at the outflow	147
D Journal publications and conference proceedings		149
D.1	Journal publications	149
D.2	Conference proceedings	150
D.3	Attendance to workshops	150

List of Figures

1.1	Comparison of the error convergence between h - and p -refinement. . .	21
2.1	Characteristic analysis of subsonic and supersonic inflow.	34
2.2	Characteristic analysis of subsonic and supersonic outflow.	36
2.3	Sketch of the flow domain around a two-dimensional aerofoil	37
3.1	The mapping that relates the local coordinates (x, y) to the reference coordinates, (ξ_1, ξ_2)	50
3.2	The one-dimensional modified modal basis functions for $P = 5$	52
3.3	The modified modal basis functions for quadrilateral and triangular elements.	53
3.4	Comparison between the high-order continuous and discontinuous Galerkin discretisation.	54
3.5	Illustration of the flux for an interface for a two-dimensional case. .	56
3.6	Illustration of the imposition of the wall boundary conditions for the governing equations	59
3.7	Supersonic laminar flow past a NACA 0012 aerofoil ($Ma = 1.2$, $Re = 1000$, $\alpha = 0^\circ$).	64
3.8	Detail of the solution for supersonic laminar flow past a NACA 0012 aerofoil ($Ma = 1.2$, $Re = 1000$, $\alpha = 0^\circ$).	64
3.9	Supersonic inviscid flow past a sphere ($Ma = 3.0$).	66
3.10	Illustration of a variable polynomial order distribution where the higher order edge is used to calculate the advective fluxes and the result is projected onto the lower order adjacent edge.	67
3.11	The solution for the Mach number obtained using Nektar++ with $N_{el} = 24$ and $P = 8$	73
3.12	Comparison between the numerical and exact adjoint solution for quasi one-dimensional subsonic inviscid flow ($Ma = 0.12$). The Nektar++ solution was obtained using $N_{el} = 24$ elements and a uniform $P = 8$	75

3.13	Comparison between the DLR/TAU code and Nektar++ of two-dimensional transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).	76
3.14	Comparison between the DLR/TAU code (47×10^3 degrees of freedom) and Nektar++ (90×10^3 degrees of freedom) of the two-dimensional adjoint solution for transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).	77
3.15	Comparison between the DLR/TAU code, Nektar++ and the analytical condition of the boundary condition values. The comparison data was provided by Dr. C. Lozano from the National Institute for Aerospace Technology (INTA) in Madrid.	78
3.16	Comparison between the DLR/TAU code and Nektar++ of the ψ_2 wall profile. The comparison data was provided by Dr. C. Lozano from the National Institute for Aerospace Technology (INTA) in Madrid.	79
4.1	Detail of the hexahedral mesh with $P = 2$ surrounding a sphere.	82
4.2	Comparison between a $P = 2$ and a $P = 6$ solution of the entropy generation for inviscid flow past a sphere ($Ma = 0.5$).	84
4.3	An example of the new polynomial order distribution based on the discontinuity sensor for inviscid flow past a sphere ($Ma = 0.5$).	85
4.4	The flow solution (left) and the adjoint solution (right) for inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5^\circ$). This is the reference solution which is obtained using a uniform $P = 9$.	89
4.5	An example of the adjoint-based error indicator (left) and the derived polynomial order distribution (right) for inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5^\circ$).	90
4.6	The flow solution (left) and the adjoint solution (right) for compressible laminar flow past a NACA 0012 aerofoil ($Ma = 0.1$, $Re = 5\,000$, $\alpha = 2^\circ$). This is the reference solution which is obtained using a uniform $P = 9$.	92
4.7	Comparison between goal-based p -adaptation and uniform polynomial refinement for compressible laminar flow past a NACA 0012 aerofoil ($Ma = 0.1$, $Re = 5\,000$, $\alpha = 2^\circ$).	93
4.8	The flow solution (left) and the adjoint solution (right) for inviscid transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).	95

4.9	Comparison between the polynomial distributions obtained for the inviscid transonic flow case using the goal-based error indicator without restriction of the polynomial order at shocks (left) and with restriction of the polynomial order to $P = 3$ (right).	96
4.10	The tetrahedral mesh around an ellipsoid of revolution shown here with the degrees of freedom corresponding to a constant polynomial order $P = 4$	98
4.11	The solution for the velocity components in each Cartesian direction (x to z from top to bottom in the left-hand side column) and for their corresponding adjoint variable (ψ_2 to ψ_4 from top to bottom in the right-hand side column) for the flow past an ellipsoid of revolution at an incidence of 3°	99
4.12	The new polynomial distribution based on the goal-based error indicator for the inviscid flow past an ellipsoid of revolution.	100
5.1	Mesh deformation induced by a displacement of the boundary. . . .	101
5.2	Definition of the linear reference element and the corresponding mapping.	104
5.3	A two-dimensional representation of the local deformation generated by the mapping.	107
5.4	Test case used to illustrate the effect of pseudo-thermal stresses. . .	108
5.5	Comparing the performance of the anisotropic and anisotropic pseudo-thermal stress term.	108
5.6	Visualization of the pseudo-thermal stresses for two different values of rotational angle. The lengths of the axes of an ellipse in the plots represent the eigenvalues of the pseudo-thermal stress tensor and the axes are orientated along the direction of its orthogonal eigenvectors.	109
5.7	Configuration of the sphere in a cube example.	110
5.8	Sketch of the velocity distribution in the vicinity of a shock wave with a linear approximation of the shock thickness δ_s	112
5.9	A sketched example of r -adaptation for shock waves where the original mesh with the desired displacement field $\mathbf{d} = \{d_1, \dots, d_N\}^t$ is given on the left and the r -adapted mesh on the right and the shock wave is represented by the dashed line.	114
5.10	Visualisation of the r -adaptation procedure for inviscid transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).	116
5.11	Detailed comparison between the original mesh and the mesh after r -adaptation at the suction side of the NACA 0012.	117

5.12	Comparison between the original mesh (black dots) and the r -adapted mesh of the Mach distribution.	118
5.13	General overview of the required steps to perform r -adaptation for supersonic inviscid flow past a forward facing step ($Ma = 3.0$). . . .	120
5.14	The contours for the pseudo-temperature (left) and the resulting contours for d_x (right) obtained using the original mesh and the two r -adaptation iterations for supersonic inviscid flow past a forward facing step ($Ma = 3.0$).	122
5.15	Comparison between the original and the r -adapted mesh of the Mach number at two locations: near the bow shock ($1.4 \leq x \leq 2$) and near the normal shock at the wall ($3 \leq x \leq 3.8$).	123

List of Tables

4.1	Comparison between uniform polynomial refinement with sensor-based p -adaptation of entropy error for inviscid flow past a sphere ($Ma = 0.5$).	84
4.2	Comparison between uniform polynomial refinement and goal-based p -adaptation of the error in lift coefficient for subsonic inviscid flow. The error is calculated with respect to the solution obtained with constant polynomial order $P = 9$	91
4.3	Comparison between uniform polynomial refinement and goal-based p -adaptation for subsonic laminar flow. The error is calculated with respect to the solution obtained using constant $P = 9$	94
4.4	Comparison between uniform polynomial refinement and goal-based p -adaptation of the error in lift coefficient for transonic inviscid flow. Furthermore, data obtained using restricted and unrestricted goal-based p -adaptation is given. The error is calculated with respect to the value $c_l^{\text{ref}} = 0.35619$ evaluated in references [99, 95].	96
4.5	Comparison between uniform polynomial refinement and goal-based p -adaptation of the error in lift coefficient for three-dimensional inviscid flow past an ellipsoid. The error is calculated with respect to the reference lift coefficient corresponding to $P = 4$	98
5.1	Values of the maximum displacement Δz_m , for the three-dimensional example of figure 5.7, versus the scaling coefficient for the anisotropic pseudo-thermal stress terms controlled through the parameter ζ . . .	111

List of Symbols

Jacobians of the fluxes

$\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}}$	Jacobian matrix obtained by linearising the i^{th} component of the convective flux vector with respect to $\mathbf{u}$, page 35
$\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}}$	Jacobian matrix obtained by linearising the normal component of the convective flux vector with respect to $\mathbf{u}$, page 35
$\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}}$	Jacobian matrix obtained by linearising the i^{th} component of the diffusive flux vector with respect to $\mathbf{u}$, page 38
$\frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}_{x_i}}$	Jacobian matrix obtained by linearising the normal component of the diffusive flux vector with respect to $\frac{\partial \mathbf{u}}{\partial x_i}$, page 39
$\frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}}$	Jacobian matrix obtained by linearising the normal component of the diffusive flux vector with respect to $\mathbf{u}$, page 39
$\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_j}}$	Jacobian matrix obtained by linearising the i^{th} component of the diffusive flux vector with respect to $\frac{\partial \mathbf{u}}{\partial x_j}$, page 38

Greek symbols

γ	Ratio of specific heats, page 25
ϵ_{c_d}	Error in drag coefficient, page 88
ϵ_{c_l}	Error in lift coefficient, page 85
ϵ_e	goal-based error indicator, page 83
ϵ_{L_2}	Entropy error, page 78
ζ	Anisotropic thermal coefficient, page 101
$\vec{\theta}_d$	Directional vector of the drag force, page 35
$\vec{\theta}_l$	Directional vector of the lift force, page 35

Λ	Diagonal matrix filled with eigenvalues, page 64
λ	Lamé's first constant, page 98
μ	Dynamic viscosity, page 26
μ_a	Artificial viscosity coefficient, page 57
$\tilde{\mu}$	Lamé's second constant, page 98
ν	Poisson ratio, page 98
ξ	Standard element coordinates, page 45
ξ_i	i^{th} component of the vector of standard element coordinates, page 45
ρ	Density, page 25
τ_{ij}	Cartesian components of the shear stress tensor, page 26
ϕ_{pq}	General representation of a multidimensional modal expansion basis, page 48
φ_p	General representation of the one-dimensional modal expansion basis, page 48
ψ_e^δ	Discrete approximation of the adjoint solution within element e , page 63
ψ_{ex}^δ	External adjacent interface solution with respect to the e^{th} element, page 64
ψ_{in}^δ	Internal adjacent interface solution with respect to the e^{th} element, page 64
Ω	Computational domain, page 24
Ω_e	Elemental domain defined in elemental coordinates, page 45
Ω_{st}	Standard elemental domain defined in standard elemental coordinates, page 45

Operators

$\mathcal{H}(\circ)$	Riemann flux function, page 52
$\mathcal{M}_e(\circ)$	Coordinate mapping $\Omega_{st} \rightarrow \Omega^e$, page 45
$\mathcal{R}(\circ)$	Shorthand for the governing equations, page 25
$\hat{\mathcal{R}}(\circ)$	Shorthand for the adjoint equations, page 36

Roman symbols

c	Speed of sound, page 56
-----	-------------------------

c_d	Drag coefficient, page 83
c_l	Lift coefficient, page 71
c_p	Specific heat for a constant pressure, page 25
c_v	Specific heat for a constant volume, page 25
$\mathbf{d}$	Displacement vector, page 98
$\tilde{E}$	Young's modulus, page 98
E	Total energy, page 25
$\mathbf{f}_i^c$	i^{th} component of the convective flux vector, page 25
$\mathbf{f}_i^v$	i^{th} component of the diffusive flux vector, page 25
$\mathbf{g}$	Vector of auxiliary variables, page 52
$\bar{h}_e$	Average distance between the vertices defining the e^{th} element, page 58
H	Total enthalpy, page 25
$\mathbf{G}$	Metric tensor, page 100
$\mathbf{J}$	Jacobian of the mapping from the reference to the physical element, page 99
J_s	Scaled Jacobian, page 100
k	Thermal conductivity, page 26
Ma	Mach number, page 35
$\vec{n}$	Normal vector, page 27
N	Number of spatial dimensions, page 45
n_i	i^{th} component of the normal vector, page 27
N_Q	Number of degrees of freedom, page 16
N_{el}	Number of elements, page 45
$N_e^{\xi_i}$	Number of integration points in ξ_i direction for an elemental solution depending on the approximation order P , page 85
P	Order of the approximating polynomial, page 15
p	Pressure, page 25

P_e	Order of the approximating polynomial in the e^{th} element, page 57
Pr	Prandtl number, page 42
$\mathbf{q}$	Vector of primitive state variables ($\mathbf{q} = \{u, v, T\}^t$), page 52
R	Gas constant, page 27
Re	Reynolds number, page 58
$\mathbf{S}_e$	Stress tensor, page 98
$\mathbf{S}_t$	Thermal stress tensor, page 100
s	Entropy, page 30
S_e	Element-wise discontinuity sensor value, page 57
S_{ref}	Refence surface, page 35
$\mathbf{T}$	Matrix with eigenvectors as columns, page 64
T	Temperature, page 27
T_p	Pseudo-temperature, page 100
T_w	Temperature at the wall, page 27
t_{CPU}	CPU time per time step normalised with the CPU time per time step for the reference case, page 86
$T_{p,0}$	Pseudo-temperature of the stress-free state, page 100
$\mathbf{u}$	Vector of conservative state variables, page 25
$\mathbf{u}_e^\delta$	Discrete approximation of the flow solution within element e , page 46
$\mathbf{u}_{ex}^\delta$	External adjacent interface solution with respect to the e^{th} element, page 51
$\mathbf{u}_{in}^\delta$	Internal adjacent interface solution with respect to the e^{th} element, page 51
$\mathbf{u}_p$	Vector of primitive state variables, page 28
$\vec{v}$	Velocity vector, page 25
v	Test function, page 45
v_e	Test function within element e , page 46
v_i	i^{th} component of the velocity vector $\vec{v}$, page 25

v_n	Normal component of the velocity at the boundary, page 56
$\boldsymbol{x}$	Cartesian coordinates, page 25
x_i	i^{th} component of the vector of Cartesian coordinates, page 25

Abstract

An accurate calculation of aerodynamic force coefficients for a given geometry is of fundamental importance for aircraft design. High-order spectral/ hp element methods, which use a discontinuous Galerkin discretisation of the compressible Navier-Stokes equations, are now increasingly being used to improve the accuracy of flow simulations and thus the force coefficients. To reduce error in the calculated force coefficients whilst keeping computational cost minimal, I propose a p -adaptation method where the degree of the approximating polynomial is locally increased in the regions of the flow where low resolution is identified using a goal-based error estimator. We initially calculate a steady-state solution to the governing equations using a low polynomial order and use a goal-based error indicator to identify parts of the computational domain that require improved solution accuracy and increase the approximation order there. We demonstrate the cost-effectiveness of our method across a range of polynomial orders by considering a number of examples in two- and three-dimensions and in subsonic and transonic flow regimes. Reductions in both the number of degrees of freedom required to resolve the force coefficients to a given error, as well as the computational cost, are both observed in using the p -adaptive technique.

In addition to the adjoint-based p -adaptation strategy, I propose a mesh deformation strategy that relies on a thermo-elastic formulation. The thermal-elastic formulation is initially used to control mesh validity. Two mesh quality indicators are proposed and used to illustrate that by *heating up* (expanding) or *cooling down* (contracting) the appropriate elements, an improved robustness of the classical mesh deformation strategy is obtained. The idea is extended to perform shock wave r -adaptation (adaptation through redistribution) for high Mach number flows. The mesh deformation strategy keeps the mesh topology unchanged, contracts the elements that cover the shock wave, keeps the number of elements constant and the computation as efficient as the unrefined case. The suitability of r -adaptation for shock waves is illustrated using internal and external compressible flow problems.

Chapter 1

Introduction

The improvement in computational resources has led to the rapid development of the numerical tools that are essential in engineering analysis. The aerospace industry is particularly interested in accurately determining the aerodynamic forces of a given geometry while keeping the computational cost as low as possible. These aerodynamic forces are highly dependent on the surrounding flow properties. Hence, the ability of accurately simulating complex processes around sophisticated geometries, while using the computational resources as efficiently as possible, becomes one of the key requirements in the development of computational fluid dynamics (CFD) codes.

The high-order spectral/*hp* element method demonstrates attractive properties like low numerical diffusion and dispersion that can be exploited to perform CFD analysis. They also employ high-order meshes to improve the geometry description. Nevertheless, they are still considered to be computationally costly. This is why high-order methods are not yet very popular in the industry since it becomes computationally expensive to use calculate solution of industrial flows that exhibit turbulent behaviour. Due to the development of adaptation strategies, the aerospace industry becomes more and more interested in using these types of methods to perform their fluid analysis. Currently, the majority of the aerodynamic simulation tools in industry use a finite volume discretisation, are mostly second-order accurate and require an enormous amount of mesh points ($\mathcal{O}(10^7 - 10^8)$) for complex flow problems. The low numerical diffusion and dispersion characteristics and the capability of highly-scalable parallel implementations are features that make the high-order spectral/*hp* element discretisation method very attractive.

The high-order spectral/*hp* element approach relies on the same mathematical background as the conventional finite element approach. Hence, the computational domain is decomposed in non-overlapping elements and the solution within each element is approximated using basis functions. However, instead of using linear or cubic basis functions, the high-order spectral/*hp* element approach uses polynomial

functions of high-order.

Often, a discontinuous Galerkin (DG) discretisation is used to discretise the compressible fluid equations. The DG approach allows the solution between elements to be discontinuous. This enables the combination of the beneficial features of the classical finite element method, its flexibility and the ability to use high-order polynomial test functions, with the beneficial features of the finite volume method which is the local statement of the governing equations. The original idea of using piecewise polynomial trial functions in the framework of a Galerkin discretisation was first proposed by Reed and Hill in 1973 [78]. They worked on a discrete approximation of the neutron transport equation. This is typically a hyperbolic problem and explicit methods were used on a triangular mesh.

Shu [82] introduced the combination of a DG spatial discretisation with a Runge-Kutta (RK) time integration scheme and this technique is referred to as RKDG. From this article, a series of articles followed that illustrated the potential of this method. First, Cockburn and Shu presented the application of RKDG for discontinuous one-dimensional problems [18, 19]. This was followed by two articles in which they showed results for multi-dimensional problems. First for simplified two-dimensional conservation laws [17] and later on for two-dimensional compressible flow problems with strong shocks [21].

The compressible fluid equations are non-linear and the speed of propagation of flow disturbances depends on the solution. As a result, the solution is characterised by local discontinuities in state properties. Hence, the DG approach is a very suitable spatial discretisation scheme for such problems since it allows the solution to be discontinuous between the elements. However, up to this point, the high-order DG discretisation was mainly used to solve compressible inviscid flows.

In 1997, Bassi and Rebay [8] presented results for compressible laminar flow problems with strong shocks using a DG discretisation scheme. Shortly after, Lomtev and Karniadakis [57] presented an improved approach for the compressible Navier-Stokes equations by using a hierarchical spectral basis. As a result, the mass matrix becomes diagonal due to the orthogonality of the basis functions and the exact quadrature and no inversion of the mass matrix is required. Considering that the compressible Navier-Stokes equations require an appropriate discretisation for the diffusive operator, a significant amount of research was dedicated to find the appropriate diffusive numerical flux functions [9, 20, 83, 100] which will be discussed in more detail in chapter 3.

A clear overview of the current status of high-order methods in CFD was presented in the framework of the ADIGMA project in [53] and in the framework of the IDIHOM project in [95]. Wang et al. [95] present strong arguments why the aerospace industry is currently moving towards the development of high-order

methods and why they are becoming more attractive to the aeronautical industry. The main arguments are the more efficiently used computational resources and the development of suitable mesh adaptation strategies.

1.1 Mesh adaptation for compressible flows

Achieving high predictive accuracy to reduce (computational) cost, during a design cycle for example, is becoming of great importance to the aerospace industry. Adaptive methods play a crucial role in obtaining a given engineering output at a pre-defined level of accuracy in the most optimal way.

Using the spectral/ hp element approach, we can improve the accuracy of the solution by either decreasing the mesh size, h , or increasing the approximation order, P , within an element. We will refer to these two types of mesh adaptation as h - and p -adaptation respectively. A DG discretisation allows the solution to be discontinuous between the elements meaning that the solution at the interfaces is defined twice. On one hand, this makes the method computationally expensive but on the other hand it makes the handling of different polynomial orders within different elements relatively easy.

For sufficiently smooth solutions, adapting the polynomial order within the domain leads to a significantly faster rate of convergence than the linear rate obtained when uniformly decreasing the element size. This is illustrated by considering an unsteady advection problem in a cube. This test case has also been used to verify the variable polynomial order implementation. We solve the unsteady advection equation

$$\frac{\partial u}{\partial t} + \vec{a} \cdot \nabla u = f \quad (1.1)$$

in a cube of dimensions $[-1, 1]^3$. The initial condition is given by

$$u(\mathbf{x}, 0) = \sin(\pi x_1) \sin(\pi x_2) \sin(\pi x_3) \quad (1.2)$$

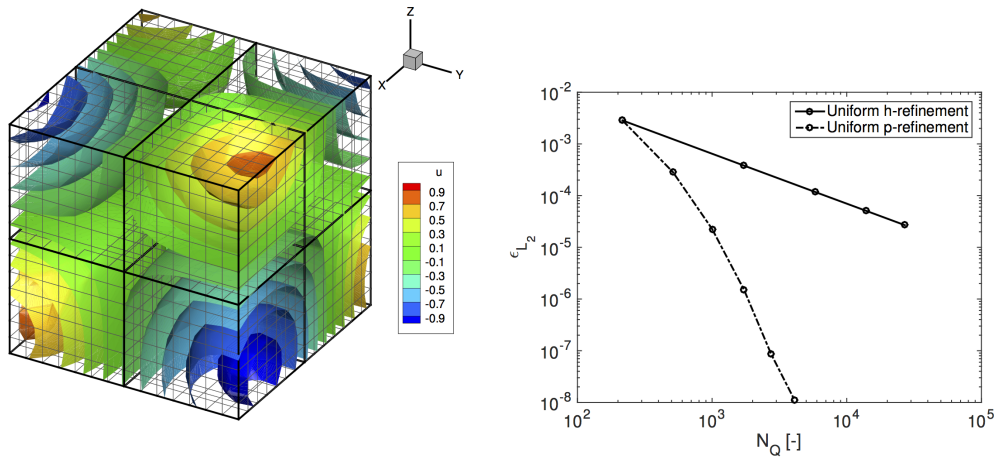
where $\mathbf{x} = (x_1, x_2, x_3)^t$ are the Cartesian coordinates. The advection vector is taken to be $\vec{a} = \{1.0, 1.5, 2.0\}^t$. We consider a time-dependent boundary condition

$$u(\mathbf{x}_b, t) = \sin(\pi(x_{1,b} - 1.0t)) \sin(\pi(x_{2,b} - 1.5t)) \sin(\pi(x_{3,b} - 2.0t)) \quad (1.3)$$

where $\mathbf{x}_b = (x_{1,b}, x_{2,b}, x_{3,b})^t$ are the coordinates at the boundary of the cube. The exact solution is

$$u(\mathbf{x}, t) = \sin(\pi(x_1 - 1.0t)) \sin(\pi(x_2 - 1.5t)) \sin(\pi(x_3 - 2.0t)) \quad (1.4)$$

When discretising equation (1.1), we can either decrease the element size and apply so called h -refinement or we can increase the approximation order within the element which we refer to as p -refinement. Figure 1.1(b) shows a comparison between uniform h - and p -refinements. We started by discretising the cube using eight hexahedral elements and we performed uniform h -refinement. We performed uniform p -refinement by using again eight hexahedral elements and increasing the polynomial order by one order per computation. Since we have an exact solution for this test case, we can plot the L_2 error against the number of degrees of freedom (N_Q) for both refinement approaches. Figure 1.1(b) shows that the error converges significantly faster with uniform p -refinement compared to uniform h -refinement.



(a) Example solution of the introduced unsteady advection problem in a cube at $t = 0.3$ obtained using eight hexahedral elements with $p = 7$. (b) Comparison of error convergence using uniform h - and p -refinement.

Figure 1.1: Comparison of the error convergence between h - and p -refinement.

We would like to use this attractive property of the spectral/ hp element approach to produce a mesh that is optimal. Babuska and Suri [4] show that, for a spectral element discretisation, the discretisation error exhibits exponential convergence as the polynomial order is increased as long as the solution does not display local discontinuities. Similar behaviour is observed in figure 1.1(b). This is one of the features of the spectral/ hp element approach that we would like to exploit in this work. The aim is to identify regions of the computational domain that require a more accurate solution using an error indicator. We increase the polynomial order locally in these regions while preserving a low polynomial order in the domains that are sufficiently resolved. To perform p -adaptation, we need an error indicator or estimator that dictates which parts of the computational domain require refinement.

Error estimation techniques can be categorised as *a priori* and *a posteriori* error estimation techniques. For *a priori* error estimators, the error bound on the solution is obtained based on known properties of the solution that are available before the solution is computed. However, in most cases, this information about the solution is not available and therefore an estimate of the error has to be found.

A posteriori error estimators are characterised by a post-processing step of the computed approximate solution to obtain an improved solution. The improved solution is compared with the original approximate solution and an error estimate is found. This estimate serves as an indicator for an adaptive algorithm to locally improve the accuracy of the discrete solution. In the following we focus on *a posteriori* error indicators.

One option is to estimate the truncation error locally by interpolating or extrapolating the solution in an element. The local truncation error can be approximated by comparing the residuals on a sequence of grids with different spacings. This is referred to as τ -estimation and has been applied to low-order methods [13, 29] and more recently to high-order spectral methods [79]. For high-order/*hp* spectral element methods, the coefficients corresponding to high-order modes are found by extrapolation and the local truncation error can be expressed in terms of these extrapolated high-order coefficients as described by Mavriplis [64, 93].

Examples of the application of a local error indicator to perform *p*-adaptation are presented in the work of Giorgiani et al. [36]. They studied the propagation of waves in a harbour for which they proposed a *p*-adaptive hybridizable discontinuous Galerkin (HDG) approach. They show that this approach outperforms the classical continuous Galerkin (CG) approach. Another example is presented by Burbeau and Sagaut [14]. They proposed a sensor that detects the regularity of the solution in the framework of a high-order DG discretisation. The polynomial order was increased or decreased based on the sensor solution. In this thesis, a similar approach is adopted to test the effectiveness of using a variable polynomial order distribution.

Demkowicz et al. [22] proposed a fully automatic *h*- and *p*-adaptivity strategy that was initially applied to elliptic problems where the projection based interpolation error of a fine reference solution was minimised. This was extended by incorporating sensitivity information that follows from an adjoint problem in the work of Solín and Demkowicz [84].

A comparison between *h*- and *p*-adaptation for external flow problems in the framework of the spectral differences method was made by Li and Jameson [56]. They found that *p*-adaptation provides higher accuracy compared to *h*-adaptation for smooth solutions both when considering the number of degrees of freedom and CPU time.

The main advantage of these local error indicators is that they are relatively easy to implement and these approaches are computationally cheap. However, these error indicators only detect the local error and do not take into account the domain of dependence of the local solution. This might result in unnecessary refinement. This issue is highlighted using a numerical example in section 4.1.1.

The aerospace industry is generally interested in determining a particular output quantity from the simulations. Examples of output quantities are aerodynamic force coefficients like the lift or drag coefficient. These coefficients are functions of the wall pressure distribution and the viscous shear stresses. Hence, we would like to determine how the aerodynamic forces will change when we locally adjust the flow properties at an arbitrary location of the surrounding flow domain. In other words, we would like to determine the sensitivity of the aerodynamic forces with respect to the state variables and use this information to perform p -adaptation.

1.1.1 Adjoint-based mesh adaptation

The following question is often posed when performing computational fluid dynamics analysis: how is the engineering output changing when the surrounding flow is perturbed at a given location? Such a perturbation can be induced by a source of momentum or a small change in geometry for example. Answering this question becomes rather difficult when dealing with a large number of degrees of freedom influencing the defined functional.

Jameson [47] used control theory to relate the change in a target functional to the gradient in design variables which in his case were variations in shape of the geometry. This was done by considering a constraint optimisation problem where the drag needed to be minimised through the variation in shape while being subject to the constraint that the governing equations need to be satisfied. Hence, the design process of an airfoil was treated as a control problem where the shape of the geometry is the control and the target functional is a quantity evaluated at the aerofoil surface and the fact that the governing equations need to be satisfied serves as a constraint to this optimisation problem.

From this constraint optimisation problem, a set of adjoint equations is derived that prescribe the sensitivity of the target functional with respect to the control variables. The solution to the adjoint problem tells us which control variables have most influence on the target functional. There is a large amount of literature available regarding the application of adjoint methodologies in CFD ranging from the earlier works by Jameson [47, 48, 71], by Anderson and Venkatakrishnan [3] and Giles and Pierce [34] for example, to the more recent works where this technique is applied to more complicated test cases [15, 49, 25].

This approach is very popular in the field of computational fluid dynamics and is currently used in a wide range of applications from flow control, flow stability analysis and mesh adaptation. In flow stability analysis and flow control, the sensitivity information is used to investigate the origin and propagation of disturbances in the flow and to determine which parts of the domain have most influence on damping these perturbations [80, 32, 63].

In this thesis, the adjoint solution is used to perform mesh adaptation in order to improve the accuracy of a specific target quantity. The discretisation error is considered to be a perturbation on the state variables which affects the error in target quantity. By considering a similar constraint optimisation problem as discussed in the papers of Jameson [47] or Giles and Pierce [34] for example, we determine the sensitivity of the error in a target functional with respect to the discretisation error in state variables that describe the surrounding flow field. Solving the derived adjoint problem is equivalent to applying a perturbation on each state variable, at each position in the computational domain and determine what the effect of this perturbation is on the target functional. To obtain the sensitivity information using this approach, one performs a computation for each perturbation on a state variable at each location in the computational domain which results in an enormous amount of computations. The same information can be obtained by solving an adjoint problem once at a cost roughly equivalent to that of solving the governing equations.

The idea of using the adjoint approach to formulate an error indicator was first applied in the finite element framework by Becker and Rannacher [10, 11]. This was then extended to DG by Hartmann [41]. Since the DG approach is very suitable to solve for non-linear hyperbolic problems, the adjoint-based adaptation strategy was applied for inviscid compressible flow problems [45, 46, 93, 61, 90] and later on for compressible laminar flow problems [42, 54, 55, 91] where mostly the adjoint-based error indicator served to drive h -adaptation. An overview of adjoint-based mesh adaptation is given in the review of Fidkowski and Darmofal [28].

High-order methods have been previously used in combination with goal-based error estimation and h -adaptation by Hartmann and Houston [46] for example. An alternative method is p -adaptation, where the polynomial order within a given element is increased or decreased as dictated by an error indicator. By varying the polynomial order throughout the domain, we can obtain additional resolution only where it is required. In this thesis, a formulation of a p -adaptive goal-based error estimation scheme for the compressible Euler and Navier-Stokes equations is proposed.

The proposed p -adaptation strategy is suitable for sufficiently smooth flow solutions. However, in the vicinity of shock waves, it is desirable to decrease the ele-

ment size instead of increasing the polynomial order to avoid numerical oscillations. Instead of using h -adaptation, it is proposed here to perform mesh deformation instead so that the addition of extra degrees of freedom is avoided by redistributing the available degrees of freedom. This is often referred to as r -adaptation.

1.1.2 Mesh deformation

One of the attractive features of high-order methods is that we can use relatively coarse meshes for smooth geometries to achieve a reasonable solution accuracy. The curvature of the geometry is taken into account by allowing the elements to be curved. Usually, for low-order methods, linear straight sided elements are used and refinement near the geometry is required to approximate the curvature of the geometry accurately. However, when using curvilinear elements, the curvature in geometry is approximated within each element and this is one of the advantages of using high-order spectral/ hp elements.

In general, a linear undeformed mesh is obtained and the curvature of the geometry is incorporated in the high-order mesh by fitting additional degrees of freedom. Normally, the location of the surrounding nodes is fixed and this can result in self-intersection for very coarse meshes when a relatively high curvature of the geometry is projected on the adjacent side of the element. Therefore, we would like to move the surrounding degrees of freedom away from the geometry to allow enough space to project the nodes onto the curved boundary while preserving the validity and quality of the mesh.

One way to do this is to use Laplacian [38, 39, 73] or Winslow smoothing techniques [50, 52]. The linear elasticity equations, often used in solid mechanics, have also been used to perform mesh deformation. The mesh is considered to be an elastic solid and subject to a prescribed boundary displacement to accommodate the curvature of geometry. The application of mesh deformation using an elastic analogy is used for fluid-structure interaction problems for example as presented in [24, 44, 85]. Karman [50] used a linear-elastic mesh deformation technique to generate unstructured meshes to solve for viscous flows. Several structured layers of elements are generated near the geometry to capture the viscous part of the domain and the existing elements are pushed away from the geometry using the linear elastic relations. A non-linear elastic analogy was applied to generate curved boundary conforming meshes from a linear mesh by Persson and Peraire [77]. A similar approach was adopted by Hassan et al. [98] who used a linear elastic analogy to improve the efficiency of the process.

In this work, we consider a mesh deformation strategy that relies on a linear-elastic formulation to ensure a valid high-order boundary conforming mesh. To

improve its robustness, we propose to incorporate pseudo-thermal stresses in the formulation. Based on a local mesh quality indicator, we can decide whether to *heat up* (expand) or *cool down* (contract) certain elements in order to avoid self intersecting elements. We propose two quality indicators and demonstrate the effectiveness of this method using two- and three-dimensional examples.

Another contribution of this work is the use of mesh deformation to accurately represent shock waves. A low-order approximation near the discontinuity is preferred to avoid numerical oscillations (Gibbs phenomenon). To increase the accuracy we need to decrease the element size in the vicinity of the shock wave. Shock waves are anisotropic flow features which preferably require anisotropic mesh adaptation. This means that the refinement takes place in perpendicular direction of the shock wave and that the elements are typically stretched and aligned parallel to the shock wave.

Examples of adaptive remeshing for shocks in supersonic flow are presented by Peraire et al. [75]. A remeshing strategy was presented that is based on a directional error indicator and produces an adapted mesh targeting shock waves, based on a pre-defined amount of nodes. This idea was later on extended for three-dimensional compressible flow problems by Peraire et al. [74]. Palemerio [72] presented mesh adaptation through mesh deformation for shock waves by looking at the directional gradient of the Mach number. This technique was demonstrated using a second-order finite element approach. Mavripilis [65] proposed to generate anisotropic elements based on a Delaunay approach in two dimensions to obtain anisotropic mesh in the wake and boundary layer regions.

George et al. [31] showed that the absolute value of the Hessian of the solution is a metric and can serve as a Riemannian metric space which can be used to generate anisotropic mesh. A mesh is obtained for which the direction and stretching (anisotropy) of the local mesh is aligned with the anisotropy of the solution by first generating a unit mesh in this defined Riemannian metric space and projecting it back onto the original Euclidean metric space. More recently, this idea has been adopted to perform anisotropic mesh adaptation where the aim is to minimize the interpolation error based on the Hessian of the solution and the local metric of the mesh [1, 58, 59, 60].

Venditti and Darmofal [92] presented anisotropic mesh adaptation for supersonic flows using an adjoint-based error indicator. The anisotropy was determined by requiring that the interpolation errors are the same along each interface for each element. More recently, anisotropic mesh adaptation in combination with p -adaptation has been applied to supersonic inviscid flows past wing profiles [6, 97]. In these works, the isotropic size of the element was determined using an adjoint-based error indicator and the choice of anisotropy of the element and the polynomial

order followed from analytic minimisation of a proposed interpolation error model.

In contrast to the aforementioned works, we propose a mesh deformation strategy for shock waves that relies on r -adaptation (adaptation through redistribution) via a thermo-elastic analogy. The shock wave detector introduced by Persson and Peraire [76] is used to calculate pseudo-thermal stresses. In this way, the elements deform so that the local element size around the shock is decreased and the representation of the shock wave is improved without adding any extra elements.

1.2 Project objective

The objective of this thesis is to investigate r - and p -adaptation strategies for compressible flows in the framework of the high-order spectral/ hp element approach. The focus of this thesis lies on two aspects: First, a goal-based error indicator is developed to perform p -adaptation in order to improve the accuracy of the aerodynamic force coefficients. The goal-based error indicator relies on a continuous adjoint formulation that is derived from the compressible flow equations. These equations are discretised and implemented in the Nektar++ library as a separate adjoint solver. One of the goals is to verify the implementation of the adjoint solver by comparing results against analytical solutions and other codes. Furthermore, the application of the adjoint-based p -adaptation strategy is demonstrated using a variety of two- and three-dimensional test cases.

Secondly, we present a mesh deformation strategy that uses a thermo-elastic analogy. The proposed mesh deformation capability is first applied to control mesh validity. This idea is then extended to adapt the local element size in order to approximate shock waves sufficiently accurate. Instead of adding extra degrees of freedom in the vicinity of the shock, we decrease the element size. In this way we keep the number of elements constant while increasing the local accuracy near shock waves.

The starting point is a compressible code that is used to simulate subsonic inviscid and laminar flows. Initially, this code is extended to handle a variable polynomial order distribution throughout the computational domain. Furthermore, the capability of modeling flows with shock waves is added. The main objectives of this thesis are:

- To develop a high-order adjoint solver that can be used to calculate the sensitivity information for aerodynamic force coefficients.
- To perform p -adaptation for compressible flows using a goal-based error indicator in order to improve the estimate of aerodynamic force coefficients.

- To apply mesh deformation using a thermo-elastic analogy to control mesh validity.
- To extend the idea of mesh deformation using a thermo-elastic analogy to perform r -adaptation (adaptation through redistribution) for shocks waves.

1.3 Thesis outline

The outline of this thesis is as follows: The governing equations are introduced in chapter 2. Some general theory regarding adjoint operators is given and the continuous adjoint equations are derived from the governing equations. This is followed by a brief overview of the DG spectral/ hp element method given in chapter 3. This discretisation method is used for both the governing and the adjoint equations and this chapter concludes with the verification of the implemented adjoint solver.

Chapter 4 starts with the application of sensor-based p -adaptation and the example that is shown serves as a motivation to use goal-based p -adaptation. This section is then followed by a description of the adjoint-based p -adaptation strategy. The efficacy of adjoint-based p -adaptation is illustrated using a range of two- and three-dimensional test cases.

We present the mesh deformation strategy that uses a thermo-elastic analogy in chapter 5. It begins with a brief description of the thermo-elastic formulation, followed by a discussion of the application of this technique for mesh quality control and to perform r -adaptation for shock waves. Both applications are illustrated using a variety of numerical examples. Based on the results obtained from adjoint-based p -adaptation and mesh deformation for mesh quality control and r -adaptation, we draw conclusions in chapter 6.

Chapter 2

The governing and adjoint equations

The conservation laws describing the dynamics of compressible fluids are introduced in this chapter. The compressible Navier-Stokes equations are derived from the principles of conservation of mass, momentum and energy. For flows where the Reynolds number is large, the compressible Euler equations are considered by neglecting the effects of viscosity and heat conduction. The compressible Navier-Stokes and Euler equations are described in sections 2.1.1 and 2.1.2 respectively.

The corresponding continuous adjoint equations for the compressible Euler and Navier-Stokes equations are also discussed in this chapter. The adjoint equations prescribe the sensitivity of the flow domain with respect to a given target functional. The general theory of the adjoint formulation is given in section 2.2. The derivation of the set of adjoint equations and corresponding boundary conditions for both the compressible Euler equations and Navier-Stokes equations are given in sections 2.2.1 and 2.2.3 respectively.

2.1 Governing equations

We consider the system of governing equations for compressible flow problems in a bounded domain, Ω , where the edge of the domain is denoted by Γ . First the set of partial differential equations are introduced. This is directly followed by a description of the boundary conditions. To limit notation, we consider two-dimensional flows in this discussion.

2.1.1 Compressible Euler equations

The compressible Euler equations describe the dynamics of compressible, isentropic flows. Hence, the effects of viscosity, heat conduction and external forces are

not considered for now. The flow is dominated by convection and therefore only the convective fluxes are considered. The steady compressible Euler equations in the domain Ω are written in conservative form as

$$\mathcal{R}(\mathbf{u}) = \sum_{i=1}^2 \frac{\partial \mathbf{f}_i^c}{\partial x_i} = 0; \quad \mathbf{u} \in \Omega \quad (2.1)$$

in a two-dimensional Cartesian frame of reference with coordinates (x_1, x_2) . The vector of conservative state variables is given by $\mathbf{u} = \{u_1, u_2, u_3, u_4\}^t = \{\rho, \rho v_1, \rho v_2, \rho E\}^t$ where ρ is the density, v_1 and v_2 are the Cartesian components of the velocity, and E is the total energy. The convective fluxes are denoted by

$$\mathbf{f}_1^c = \begin{Bmatrix} \rho u \\ p + \rho v_1^2 \\ \rho v_1 v_2 \\ \rho v_1 H \end{Bmatrix} \quad \mathbf{f}_2^c = \begin{Bmatrix} \rho v \\ \rho v_1 v_2 \\ p + \rho v_2^2 \\ \rho v_2 H \end{Bmatrix} \quad (2.2)$$

Here, p is the pressure. The total enthalpy is denoted by H and defined as

$$H = E + \frac{p}{\rho} \quad (2.3)$$

The relation between the total specific energy, E , and the pressure is required to close the system and is given by

$$p = (\gamma - 1)\rho \left(E - \frac{1}{2}(v_1^2 + v_2^2) \right) \quad (2.4)$$

where $\gamma = \frac{c_p}{c_v}$ represents the ratio of specific heats. Usually for air $\gamma = 1.4$.

2.1.2 Compressible Navier-Stokes equations

The compressible Navier-Stokes equations are considered when the effects of viscosity and heat conduction are taken into account. The compressible Navier-Stokes equations are written in conservative form as

$$\mathcal{R}(\mathbf{u}, \nabla \mathbf{u}) = \sum_{i=1}^2 \frac{\partial}{\partial x_i} \{ \mathbf{f}_i^c(\mathbf{u}) - \mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u}) \} = \mathbf{0}; \quad \mathbf{u} \in \Omega \quad (2.5)$$

The convective fluxes are the same as the ones presented in equation (2.2). The Cartesian components of the viscous fluxes are given in equation (2.6).

$$\mathbf{f}_1^v = \begin{pmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ v_1\tau_{11} + v_2\tau_{12} - k\frac{\partial T}{\partial x_1} \end{pmatrix} \quad \mathbf{f}_2^v = \begin{pmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ v_1\tau_{21} + v_2\tau_{22} - k\frac{\partial T}{\partial x_2} \end{pmatrix} \quad (2.6)$$

where the components of the shear stress tensor are given by

$$\tau_{11} = 2\mu \frac{\partial v_1}{\partial x_1} - \frac{2}{3}\mu \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} \right) \quad (2.7)$$

$$\tau_{12} = \tau_{21} = \mu \left(\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} \right) \quad (2.8)$$

$$\tau_{22} = \frac{\partial v_2}{\partial x_2} - \frac{2}{3}\mu \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} \right) \quad (2.9)$$

Usually the viscosity coefficient, μ , is prescribed by Sutherland's law but out of simplicity this is taken to be constant in all the cases in this thesis.

2.1.3 Boundary conditions

The governing equations are solved within a bounded domain Ω . We consider external flows meaning that the computational domain is bounded by a wall boundary that defines a geometry around which the fluid is flowing. Depending on the governing equations, we consider a slip or no-slip wall boundary condition. Since we are dealing with a finite computational domain, we need to apply an artificial boundary condition at the far-field that mimics the flow conditions far away from the geometry. An overview of the types of boundary conditions is given in the next section.

Wall boundary conditions

We consider external compressible flow problems and the flow properties around a given body shape are determined. The solid wall of the body, denoted by Γ_w , is modelled by applying wall boundary conditions. Depending on whether we taken viscous and thermal effects into account, we use either slip or no-slip boundary conditions.

When the effects of viscosity and heat conduction are neglected and we consider the compressible Euler equations, we assume that the flow passes a frictionless wall. Therefore, the normal component is set to zero and we consider a slip boundary

condition. It is assumed that for inviscid flows, the flow follows the body and therefore we impose

$$\vec{v} \cdot \vec{n} = 0; \quad \mathbf{u} \in \Gamma_w \quad (2.10)$$

at the wall where the velocity vector is denoted by $\vec{v} = \{v_1, v_2\}^t$ and the normal vector pointing outwards of the wall boundary is written as $\vec{n} = \{n_1, n_2\}^t$.

When considering the Navier-Stokes equations, we state that the friction at the wall forces the tangential velocity component to zero. It is assumed to have a no-slip wall boundary and that the velocity at the wall is zero, i.e.

$$v_1 = v_2 = 0; \quad \mathbf{u} \in \Gamma_w \quad (2.11)$$

In the case of compressible viscous flow, we need to make a distinction between an isothermal and an adiabatic wall. For an isothermal wall boundary, we impose a fixed temperature at the wall defined as

$$T_w = \frac{\gamma - 1}{R} \left(E - \frac{1}{2}(v_1^2 + v_2^2) \right) = \frac{\gamma - 1}{R} E; \quad \mathbf{u} \in \Gamma_{\text{iso}} \quad (2.12)$$

where T_w is a prescribed wall temperature of the solid wall and R is the perfect gas constant. Equation (2.12) is written in this way since this condition is imposed by setting the total energy, E , at the wall depending on the desired wall temperature. If an adiabatic wall is considered, the normal gradient of the temperature with respect to the wall is zero and therefore we write

$$\nabla T \cdot \vec{n} = 0; \quad \mathbf{u} \in \Gamma_{\text{adia}} \quad (2.13)$$

where $\nabla T = \left\{ \frac{\partial T}{\partial x_1}, \frac{\partial T}{\partial x_2} \right\}^t$.

Far-field boundary conditions

Usually, a finite computational domain is used to model flows past wing bodies in free-flight. The values at the far-field are not known beforehand since they rely on the internal solution that is evolving in time. The far-field boundary conditions should avoid the introduction of spurious reflections that could influence the accuracy of the solution.

Non-reflective far-field boundary conditions for hyperbolic problems have been extensively discussed in the literature like the works by Thompson for example [87, 88]. It is assumed that the far-field boundary is sufficiently far away so that the flow is considered to be inviscid. The far-field boundary treatment is then based on the characteristic analysis that describes the propagation of the Riemann invariants along the characteristic lines. This characteristic analysis is based on

rewriting the compressible Euler equations in terms of a new set of quantities and show that these quantities are constant along characteristic lines and travel with characteristic velocities $d\mathbf{x}/dt = \boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ are the eigenvalues of the system. We start by writing the compressible Euler equations in non-conservative form in terms of the primitive variables $\mathbf{u}_p = \{\rho, u, v, p\}^t$ as

$$\frac{\partial \mathbf{u}_p}{\partial t} + \sum_{i=1}^2 \mathbf{A}_{p,i} \frac{\partial \mathbf{u}_p}{\partial x_i} = 0 \quad (2.14)$$

To simplify notation, we write the Jacobian of the fluxes as $\mathbf{A}_i = \partial \mathbf{f}_i^c / \partial \mathbf{u}_p$. We can now introduce the following change of variables

$$\mathbf{u}_p = \mathbf{S}\mathbf{w} \rightarrow \mathbf{w} = \mathbf{S}^{-1}\mathbf{u}_p \quad (2.15)$$

where $\mathbf{S}$ represents the transformation matrix and $\mathbf{w}$ is the new set of characteristic variables that we are seeking. Equation (2.14) can be written as

$$\mathbf{S} \frac{\partial \mathbf{w}}{\partial t} + \sum_{i=1}^2 \mathbf{A}_{p,i} \mathbf{S} \frac{\partial \mathbf{w}}{\partial x_i} = 0 \quad (2.16)$$

Multiplying this equation again with $\mathbf{S}^{-1}$ gives

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{i=1}^2 \boldsymbol{\Lambda}_i \frac{\partial \mathbf{w}}{\partial x_i} = 0 \quad (2.17)$$

where $\boldsymbol{\Lambda}_i = \mathbf{S}^{-1} \mathbf{A}_{p,i} \mathbf{S}$ is a diagonal matrix that has the eigenvalues $\boldsymbol{\lambda}$ as diagonal entries which describe the velocities at which the perturbations move along the characteristic lines $d\mathbf{x}/dt$. Furthermore, $\mathbf{w}$ represents the vector of characteristic variables that is defined by $\mathbf{w} = \mathbf{S}^{-1}\mathbf{u}_p$.

We simplify the problem by considering the flow to be normal with respect to the far-field boundary so that we can consider a Riemann problem normal to the far-field boundary. We then calculate the eigenvalues and corresponding eigenvectors for the system as

$$\boldsymbol{\lambda} = \begin{bmatrix} v_n \\ v_n \\ v_n + c \\ v_n - c \end{bmatrix}, \quad \mathbf{S}^{-1} = \frac{1}{2\rho c^2} \begin{bmatrix} 2\rho c^2 & 0 & 0 & -2\rho \\ 0 & 0 & 2\rho c^2 & 0 \\ 0 & c\rho & 0 & 1 \\ 0 & -c\rho & 0 & 1 \end{bmatrix} \quad (2.18)$$

The characteristic variables are then written in differential form as $d\mathbf{w} = \mathbf{S}^{-1}d\mathbf{u}_p$

and given by

$$d\mathbf{w} = \begin{Bmatrix} d\rho - \frac{dp}{c^2} \\ dv_t \\ (\rho c)dv_n + dp \\ -(\rho c)dv_n + dp \end{Bmatrix} = \begin{Bmatrix} d\rho - \frac{dp}{c^2} \\ dv_t \\ \rho c \left(dv_n + \frac{dp}{c\rho} \right) \\ -\rho c \left(dv_n - \frac{dp}{c\rho} \right) \end{Bmatrix} \quad (2.19)$$

where the normal velocity component is given by $v_n = \vec{v} \cdot \vec{n}$ and the tangential velocity component by $v_t = \vec{v} - v_n \vec{n}$. The two last equations in (2.19) are associated with the largest and smallest eigenvalue. Integrating these terms gives that

$$J_{\pm} = v_n \pm \frac{2}{\gamma - 1} c \quad (2.20)$$

which are the Riemann invariants that are associated with the characteristic lines $\frac{\partial \mathbf{x}}{\partial t} = u \pm c$. The second equation in (2.19) represents the propagation of vorticity waves and the first equation corresponds to the entropy being constant. Below, this analysis is applied to discuss the inflow and outflow boundary condition. During this analysis, we consider the flow to be normal to the far-field boundary with the normal vector pointing outwards.

Inflow $v_n < 0$

At subsonic inflow, the normal vector is pointing outwards in opposite direction as the flow direction, hence $v_n < 0$. The correct values of the state variables at inflow are determined based on the propagation of the previously discussed Riemann invariants. A schematic representation of the characteristics for subsonic inflow is given in figure 2.1(a).

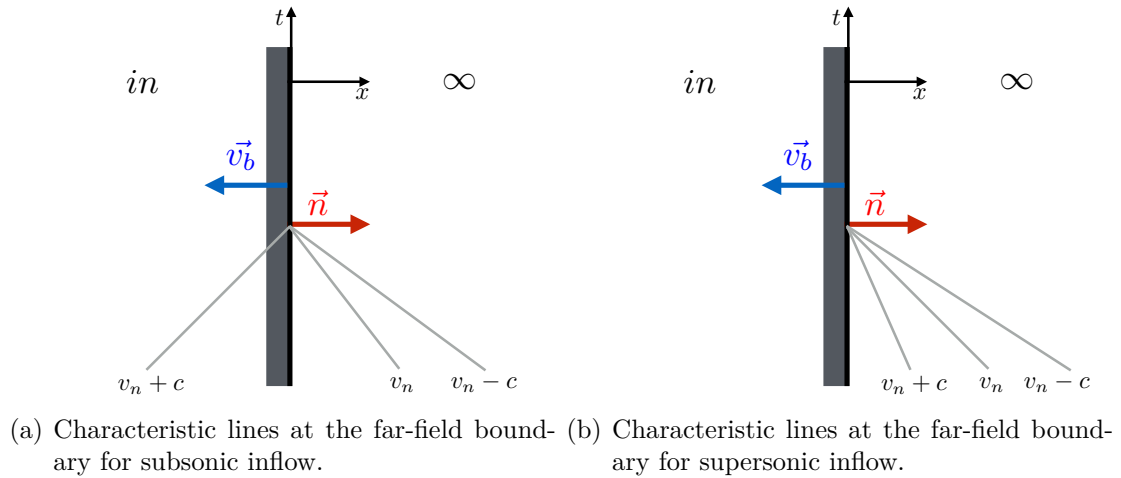


Figure 2.1: Characteristic analysis of subsonic and supersonic inflow.

For inflow holds that J_- , which is the Riemann invariant that corresponds to the eigenvalue $v_n - c$, is described by the far-field values of the state variables $\mathbf{u}_\infty = \{\rho_\infty, (\rho u)_\infty, (\rho v)_\infty, (\rho E)_\infty\}^t$ while the Riemann invariant J_+ is associated to the internal state variables $\mathbf{u}_{in} = \{\rho_{in}, (\rho u)_{in}, (\rho v)_{in}, (\rho E)_{in}\}^t$. The normal velocity and the speed of sound at the boundary should satisfy the following two relations at inflow

$$J_{-,b} = v_{n,b} - \frac{2c_b}{\gamma-1} = v_{n,\infty} - \frac{2c_\infty}{\gamma-1} \quad (2.21)$$

$$J_{+,b} = v_{n,b} + \frac{2c_b}{\gamma-1} = v_{n,in} + \frac{2c_{in}}{\gamma-1} \quad (2.22)$$

where $v_{n,b} = \vec{v}_b \cdot \vec{n}$, $v_\infty = \vec{v}_\infty \cdot \vec{n}$ and $v_{in} = \vec{v}_{in} \cdot \vec{n}$ are the normal components of the velocity at the boundary, free-stream and internal conditions respectively. Furthermore, c_b , c_∞ and c_{in} represent the boundary, free-stream and interior speed of sound respectively. From this, the normal velocity and speed of sound at the inflow boundary are retrieved using

$$v_{b,n} = \frac{J_{+,in} + J_{-, \infty}}{2}, \quad c_b = \frac{\gamma-1}{4} (J_{-, \infty} - J_{+, in}) \quad (2.23)$$

The two Cartesian components of the velocity at the boundary are calculated using

$$\vec{v}_b = \vec{v}_\infty + (v_{n,b} - \vec{v}_{n,\infty})\vec{n} \quad (2.24)$$

where $\vec{v}_b = \{u_b, v_b\}^t$. Assuming that the solution is smooth and no shock waves are present, we state that the entropy at the boundary is equal to the far-field entropy. Hence,

$$s_b = \frac{c_\infty^2}{\gamma \rho_\infty^{\gamma-1}} \quad (2.25)$$

Using equations (2.23) and (2.28), we write the density at the inflow as

$$\rho_b = \frac{c_b^2}{\gamma s_b} \quad (2.26)$$

and the pressure at the inflow as

$$p_b = \frac{\rho_b c_b^2}{\gamma} \quad (2.27)$$

The conservative state variables are now determined based on the expressions found for ρ_b , $v_{b,n}$, c_b , and p_b . In the above case, we assumed the inflow to be subsonic ($Ma_n = |v_{b,n}|/c < 1$). However when supersonic inflow ($Ma_n = |v_{b,n}|/c \geq 1$) is considered, we have no outgoing characteristic wave and J_+ is then as well calculated using the far-field state variables as depicted in figure 2.1(b).

Outflow $v_n > 0$

The subsonic outflow conditions are treated in a similar fashion. However, now the incoming Riemann invariant is associated to the free-stream state variables while the outgoing Riemann invariant is associated to the internal state variables. This is illustrated in figure 2.2(a).

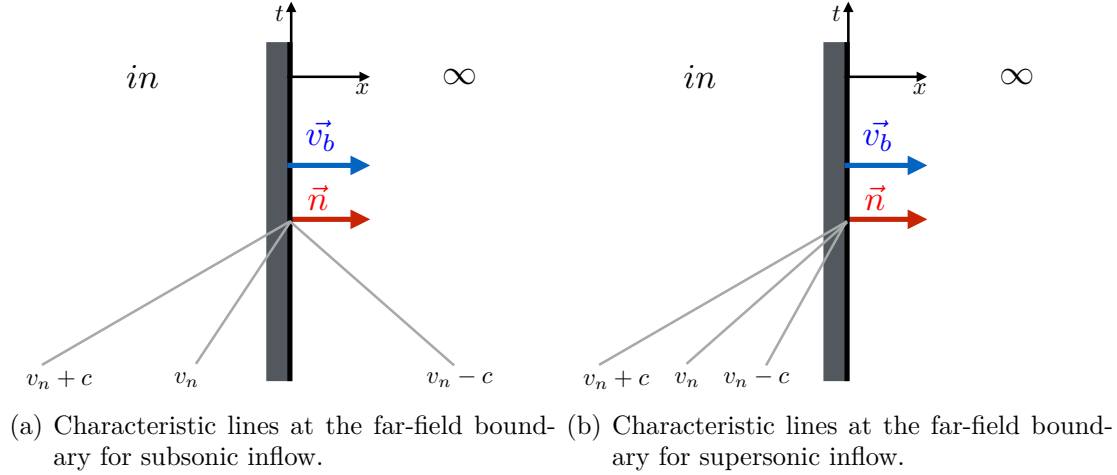


Figure 2.2: Characteristic analysis of subsonic and supersonic outflow.

The normal component of the velocity at the wall and the speed of sound are calculated again using equations (2.21) and (2.22). Again using the same assumption that there are no shock waves present, we extrapolate the entropy from the interior state using

$$s_b = \frac{c_{in}^2}{\gamma \rho_{in}^{\gamma-1}} \quad (2.28)$$

Finally, the density and the pressure are again calculated using equations (2.26) and (2.27). This characteristic analysis is used in section 3.2.3 to discretise the far-field boundary. For now, we consider the continuous governing equations and a more detailed description is given of the far-field boundary treatment during the discrete approximation that is discussed in section 3.2.3.

For viscous external flows, we can also assume the boundary to be sufficiently far away such that the far-field flow can be considered to be inviscid. However when the boundary layer is considered at outflow, we typically extrapolate the velocity components and fix the outer total pressure to a free-stream pressure value.

2.2 Adjoint equations

As mentioned in the introduction of this thesis, we are interested in determining the sensitivity of a given target functional, in this case the aerodynamic force

coefficients, with respect to the surrounding flow variables. The goal here is to identify the areas in the computational domain where the local error in flow properties have the most influence on the error in aerodynamic force coefficients. The discretisation error can be seen as a variation in flow quantities with respect to the exact solution which on its turn has an effect on the discrete solution of the aerodynamic force coefficients. We therefore would like to link the change in target functional to the change of the surrounding flow properties which can be done through the solution of an adjoint problem. The solution to this adjoint problem provides sensitivity information that can be used for mesh adaptation purposes to reduce the error in aerodynamic force coefficient which is demonstrated in chapter 4. In this section, we derive a set of continuous adjoint equations that describe the sensitivity of aerodynamic force coefficients with respect to the surrounding flow variables.

In the remainder of this chapter, we consider a two-dimensional compressible flow past a wing profile as sketched in figure 2.3.

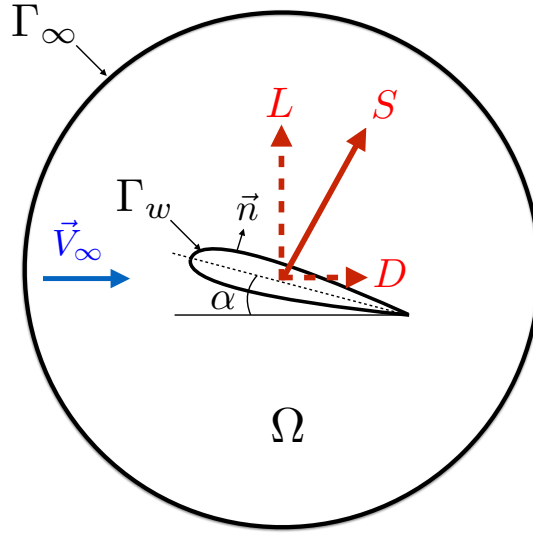


Figure 2.3: Sketch of the flow domain around a two-dimensional aerofoil

The angle of attack of the aerofoil is indicated by α , Γ_w and Γ_∞ indicate the wall and far-field boundaries respectively. The resultant force acting on the wing geometry is indicated by S which is decomposed in a lift and drag force, indicated by L and D respectively. The lift or drag coefficient, denoted by $\mathcal{J}(\mathbf{u})$, is taken to be the target functional and typically depends on the surrounding flow properties $\mathbf{u}$. We can approximate the variation in $\delta\mathcal{J}$ that is induced by a variation on the state variables, $\delta\mathbf{u}$, as

$$\delta\mathcal{J} = \mathcal{J}(\mathbf{u} + \delta\mathbf{u}) - \mathcal{J}(\mathbf{u}) \approx \left\{ \frac{\partial\mathcal{J}}{\partial\mathbf{u}} \right\}^t \delta\mathbf{u} \quad (2.29)$$

In equation (2.29) we have that $\frac{\partial \mathcal{J}}{\partial \mathbf{u}}$ is a column vector with four entries

$$\left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}_i = \frac{\partial \mathcal{J}}{\partial u_i} \quad i = 1, \dots, 4 \quad (2.30)$$

We can also approximate the variation of $\delta \mathcal{R}$ using a Taylor series expansion truncated to first order [30]. Hence, we write

$$\delta \mathcal{R} = \mathcal{R}(\mathbf{u} + \delta \mathbf{u}) - \mathcal{R}(\mathbf{u}) \approx \left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right] \delta \mathbf{u} \quad (2.31)$$

For two dimensional compressible flows, we solve for four state variables and therefore $\left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right]$ is a 4×4 matrix of the linearised governing equations around a given base flow solution. Hence, the entries of this matrix are given by

$$\left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right]_{i,j} = \frac{\partial \mathcal{R}_i}{\partial u_j}; \quad i, j = 1, \dots, 4 \quad (2.32)$$

To facilitate the identification of the nature of the various terms involved in the analysis that follows, we will use $[\circ]$ to denote a 4×4 matrix and $\{\circ\}$ for a column vector with four entries. Equation (2.32) can be rewritten such that

$$\delta \mathbf{u} = \left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right]^{-1} \delta \mathcal{R} \quad (2.33)$$

Substituting this expression into equation (2.29) gives

$$\delta \mathcal{J} = \left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}^t \left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right]^{-1} \delta \mathcal{R} \quad (2.34)$$

We consider the term $\boldsymbol{\psi}^t = \left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}^t \left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right]^{-1}$ to be a vector that has the same length as $\mathbf{u}$ which links the variation of the local residual to the variation in target functional. Hence, we write,

$$\delta \mathcal{J} = \boldsymbol{\psi}^t \delta \mathcal{R} \quad (2.35)$$

where $\boldsymbol{\psi}^t = \{\psi_1, \psi_2, \psi_3, \psi_4\}^t$ for a two-dimensional case. Substituting equations (2.29) and (2.31) back into equation (2.35) we obtain

$$\left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}^t \delta \mathbf{u} = \boldsymbol{\psi}^t \left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right] \delta \mathbf{u} \quad (2.36)$$

Equation (2.36) can be rewritten as

$$\left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}^t - \boldsymbol{\psi}^t \left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right] = 0 \quad (2.37)$$

Equation (2.37) can be used as a starting point to either obtain the adjoint vector, $\boldsymbol{\psi}$, using the *discrete* or *continuous* adjoint approach. Following the discrete adjoint approach, the non-linear governing equations are first discretised and solved and the resulting equations are then linearised. The adjoint vector $\boldsymbol{\psi}$ is determined by rewriting equation (2.37). One could as well solve the linearised governing equations and determine the linearised discrete target functional and once these terms are known, they can be plugged into equation (2.37) to find the final expression for $\boldsymbol{\psi}$.

The continuous adjoint approach uses a similar form as presented in equation (2.37) to derive a set of equations that describe $\boldsymbol{\psi}$. Once the set of continuous adjoint equations is derived, they can be discretised using similar discretisation techniques as the ones used for the governing equations.

In this thesis, it was chosen to use the continuous approach since most of the relevant architecture of the code that is used to solve the governing equations can be reused to solve the derived adjoint equations. The implementation becomes easier in this way. Furthermore, we will weakly impose the boundary conditions for the governing equations. The literature has shown that the discrete adjoint approach may lead to inconsistent adjoint solutions once the boundary conditions for the governing equations are imposed weakly, depending on which type of approximate Riemann flux function is used (see chapter 3). The paper by Hartmann [43] shows this for the compressible Euler equations. It was shown that depending on the flux function that is used to impose the boundary conditions for the governing equations, a consistent or inconsistent adjoint solution was obtained meaning that for the inconsistent adjoint solution, large numerical oscillations occur near the boundary [23, 43, 62].

Equation (2.37) is often used when the discrete adjoint approach is discussed. In that case, the term $\partial \mathcal{J} / \partial \mathbf{u}$ is an array of the discretised linearised target functional and the $\boldsymbol{\psi}^t [\partial \mathcal{R} / \partial \mathbf{u}]$ is a vector-matrix multiplication of the discrete adjoint and the linearised residuals. Considering that the conservation laws need to be satisfied within a computational domain Ω , we can write equation (2.37) in a slightly different way to start the derivation of the continuous adjoint equations. The adjoint vector is multiplied with the residuals and integrated over the computational

domain Ω . Hence, we write

$$\left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}^t - \int_{\Omega} \boldsymbol{\psi}^t \left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right] d\Omega = 0 \quad (2.38)$$

The target functional is typically an integral quantity as we will see in the following sections. Generally, the spatial derivatives that occur in the governing equations are moved onto the adjoint variables, $\boldsymbol{\psi}$, by applying Gauss' divergence theorem on the second term on the left hand side of equation (2.38). In this way, a set of adjoint equations is derived that prescribe $\boldsymbol{\psi}$. This derivation generally depends on two things: the formulation of the target functional, $\mathcal{J}(\mathbf{u})$, and the governing equations $\mathcal{R}(\mathbf{u})$. In the following two sections, the adjoint equations for the compressible Euler and Navier-Stokes equations are derived with the aerodynamic force coefficient as an objective functional.

2.2.1 Adjoint Euler equations

The continuous adjoint equations for compressible inviscid flow problems are derived in this section. For compressible inviscid flows, the drag or lift coefficient is determined by the integral of the pressure along the wall of the geometry.

$$\mathcal{J}(\mathbf{u}) = \int_{\Gamma_w} p(\mathbf{u}) (\vec{\theta} \cdot \vec{n}) d\Gamma \quad (2.39)$$

This expression is usually multiplied by $C_{\infty} = \frac{2}{\gamma S_{ref} Ma_{\infty}^2}$, where S_{ref} denotes a reference length of the body and Ma_{∞} defines the free stream Mach number, to obtain a non-dimensional aerodynamic force coefficient. In equation (2.39), the vector $\vec{\theta}$ determines the direction of interest. For the lift force $\vec{\theta} = \{-\sin \alpha, \cos \alpha\}^t$ while for the drag force, $\vec{\theta}$ becomes $\theta = \{\cos \alpha, \sin \alpha\}^t$. The variation in target functional, defined in equation (2.29), is approximated as

$$\left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}^t \delta \mathbf{u} = \int_{\Gamma_w} (\vec{\theta} \cdot \vec{n}) \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\}^t \delta \mathbf{u} d\Gamma \quad (2.40)$$

For a two-dimensional problem, the term $\frac{\partial p}{\partial \mathbf{u}}$ is defined as

$$\left\{ \frac{\partial p}{\partial \mathbf{u}} \right\}_i = \frac{\partial p}{\partial u_i} \quad i = 1, \dots, 4 \quad (2.41)$$

The variation $\delta \mathcal{R}$ is determined using (2.31) and for the compressible Euler equations, this becomes

$$\left[\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right] \delta \mathbf{u} = \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right] \delta \mathbf{u} \quad (2.42)$$

where $\left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}}\right]$ to $\left[\frac{\partial \mathbf{f}_2^c}{\partial \mathbf{u}}\right]$ are 4×4 matrices that represent the Jacobian of the convective fluxes. More details on their entries are given in Appendix A. Substituting equations (2.40) and (2.42) into equation (2.38) and applying Gauss' divergence theorem on the resulting second term on the left hand side of equation (2.38), we obtain

$$\begin{aligned} \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} (\vec{\theta} \cdot \vec{n}) \right\} d\Gamma &= \int_{\Gamma} \delta \mathbf{u}^t \left\{ \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t n_i \psi \right\} d\Gamma \\ &- \int_{\Omega} \delta \mathbf{u}^t \left\{ \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t \frac{\partial \psi}{\partial x_i} \right\} d\Omega \end{aligned} \quad (2.43)$$

Equation (2.43) shows one area and one boundary integral on the right hand side which added together have to match the boundary integral on the left hand side. To derive the adjoint equations and appropriate boundary conditions, it is helpful to split the boundary, Γ , into two components: the wall boundary, Γ_w , and the rest, i.e. the far-field boundary for external flows, which is denoted by Γ_f .

$$\begin{aligned} \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} \vec{\theta} \cdot \vec{n} \right\} d\Gamma &= \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t n_i \psi \right\} d\Gamma \\ &+ \int_{\Gamma_f} \delta \mathbf{u}^t \left\{ \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t n_i \psi \right\} d\Gamma \\ &- \int_{\Omega} \delta \mathbf{u}^t \left\{ \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t \frac{\partial \psi}{\partial x_i} \right\} d\Omega \end{aligned} \quad (2.44)$$

Since there is no volume integral present on the left hand side of equation (2.44) and it is assumed that the variation on the state variables, here denoted by $\delta \mathbf{u}$, are arbitrary, we state that the term within curly brackets in the volume integral on the right hand side of equation (2.43) needs to be zero. This results in

$$\hat{\mathcal{R}}(\psi) = - \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t \frac{\partial \psi}{\partial x_i} = 0; \quad \mathbf{u} \in \Omega \quad (2.45)$$

which are the *adjoint equations* corresponding to the compressible Euler equations. Examining equation (2.44) leaves us with two boundary integrals at the wall which need to be equal and one far-field boundary integral term that needs to vanish. This provides the appropriate boundary conditions for the derived adjoint equations presented in equation (2.45).

2.2.2 Boundary conditions for the adjoint Euler equations

As mentioned in the previous section, we generally consider two boundary contributions here, the wall boundary and the far-field boundary, which are discussed individually in this section.

Adjoint wall boundary conditions

Equation (2.44) shows two boundary integrals over Γ_w which need to be equal. Therefore, we write

$$\int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} (\vec{\theta} \cdot \vec{n}) \right\} d\Gamma = \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi} \right\} d\Gamma \quad (2.46)$$

where

$$\left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right] = \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right] n_i \quad (2.47)$$

Using again the argument that $\delta \mathbf{u}$ is arbitrary, we write that the integrands in equation (2.46) should be equal. Hence, we obtain

$$\left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi} \Big|_{\Gamma_w} = \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} (\vec{\theta} \cdot \vec{n}) \Big|_{\Gamma_w} \quad (2.48)$$

For compressible inviscid flow past a body, the wall boundary conditions prescribe that the normal velocity component is zero (see equation (2.10)). Substituting this boundary condition gives a normal convective flux at the wall

$$\mathbf{f}_n^c(\mathbf{u}) = \begin{Bmatrix} 0 \\ p\vec{n} \\ 0 \end{Bmatrix}; \quad \mathbf{u} \in \Gamma_w \quad (2.49)$$

Differentiating expression (2.49) with respect to $\mathbf{u}$ and substituting the result into equation (2.48), provides us with the boundary condition at the wall when solving the adjoint problem for compressible inviscid flows. Defining the adjoint vector $\vec{\psi} = [\psi_2, \psi_3]^t$, the adjoint boundary conditions at the wall are defined as

$$\vec{\psi} \cdot \vec{n} = \vec{\theta} \cdot \vec{n}; \quad \boldsymbol{\psi} \in \Gamma_w \quad (2.50)$$

For the adjoint variables ψ_1 and ψ_4 holds that they are not prescribed at the boundary. Therefore, these adjoint variables are not changed and extrapolated from the adjoint field solution.

Adjoint far-field boundary conditions

As for the volume integral in equation (2.44), we do not find a contribution of the far-field boundary on the left hand side of equation (2.44), therefore

$$\int_{\Gamma_f} \delta \mathbf{u}^t \left\{ \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^e}{\partial \mathbf{u}} \right]^t n_i \boldsymbol{\psi} \right\} d\Gamma = \mathbf{0} \quad (2.51)$$

This is achieved by setting the boundary conditions at the far-field

$$\boldsymbol{\psi} = \mathbf{0}; \quad \boldsymbol{\psi} \in \Gamma_f \quad (2.52)$$

2.2.3 Adjoint Navier-Stokes equations

The same methodology as discussed in the previous section is used to derive the adjoint equations for compressible viscous flows. In this section, the compressible Navier-Stokes equations are considered meaning that the value of the aerodynamic force coefficient is governed by the pressure and the viscous shear stresses at the wall of the geometry. Hence, the target functional is defined as

$$\mathcal{J}(\mathbf{u}) = \int_{\Gamma_w} (p(\mathbf{u})\vec{n} - \boldsymbol{\tau}(\mathbf{u}, \nabla \mathbf{u})\vec{n}) \cdot \vec{\theta} d\Gamma \quad (2.53)$$

where $\boldsymbol{\tau}(\mathbf{u}, \nabla \mathbf{u})$ is the tensor of viscous stresses, defined as

$$\boldsymbol{\tau}(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \end{bmatrix} \quad (2.54)$$

The objective functional now also depends on the first order derivatives of $\mathbf{u}$ denoted by $\nabla \mathbf{u} = \{\mathbf{u}_{x_1}, \mathbf{u}_{x_2}\}^t$ where $\mathbf{u}_{x_1} = \frac{\partial \mathbf{u}}{\partial x_1}$ and $\mathbf{u}_{x_2} = \frac{\partial \mathbf{u}}{\partial x_2}$. In this case, $\mathbf{u}_{x_1}$ and $\mathbf{u}_{x_2}$ are both column vectors with four entries. Each of these derivative vectors have four entries for two-dimensional flow. The linearised target functional is expressed as

$$\begin{aligned} \delta \mathcal{J} &\approx \left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \right\}^t \delta \mathbf{u} + \sum_{i=1}^2 \left\{ \frac{\partial \mathcal{J}}{\partial \mathbf{u}_{x_i}} \right\}^t \left\{ \frac{\partial}{\partial x_i} \delta \mathbf{u} \right\} \\ &= \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} (\vec{\theta} \cdot \vec{n}) - \delta \mathbf{u}^t \left\{ \frac{\partial}{\partial \mathbf{u}} ((\boldsymbol{\tau} \vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \\ &\quad - \int_{\Gamma_w} \sum_{i=1}^2 \delta \mathbf{u}_{x_i}^t \left\{ \frac{\partial}{\partial \mathbf{u}_{x_i}} ((\boldsymbol{\tau} \vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \end{aligned} \quad (2.55)$$

Linearising the compressible Navier-Stokes equations yields

$$\begin{aligned}\delta\mathcal{R} &\approx \left[\frac{\partial\mathcal{R}}{\partial\mathbf{u}}\right]\delta\mathbf{u} + \sum_{i=1}^2 \left[\frac{\partial\mathcal{R}}{\partial\mathbf{u}_{x_i}}\right]\delta\mathbf{u}_{x_i} \\ &= \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[\frac{\partial\mathbf{f}_i^c}{\partial\mathbf{u}} - \frac{\partial\mathbf{f}_i^v}{\partial\mathbf{u}}\right]\delta\mathbf{u} - \sum_{i,j=1}^2 \frac{\partial}{\partial x_j} \left[\frac{\partial\mathbf{f}_i^v}{\partial\mathbf{u}_{x_j}}\right]\delta\mathbf{u}_{x_i}\end{aligned}\quad (2.56)$$

where the linearised convective fluxes are the same as the ones described in the previous section. The differentiation of the diffusive fluxes with respect to the first order derivatives of $\mathbf{u}$ results in a 4×4 Jacobian matrix for each spatial derivative in each spatial direction. These linearised diffusive fluxes are denoted by $\frac{\partial\mathbf{f}_1^v}{\partial\mathbf{u}_{x_1}}$, $\frac{\partial\mathbf{f}_2^v}{\partial\mathbf{u}_{x_1}}$, $\frac{\partial\mathbf{f}_1^v}{\partial\mathbf{u}_{x_2}}$ and $\frac{\partial\mathbf{f}_2^v}{\partial\mathbf{u}_{x_2}}$. These matrices are defined in full in Appendix A. We can now substitute the expressions (2.55) and (2.56) into equation (2.38) which gives

$$\begin{aligned}&\int_{\Gamma_w} \delta\mathbf{u}^t \left\{ \frac{\partial p}{\partial\mathbf{u}} \right\} \vec{\theta} \cdot \vec{n} - \delta\mathbf{u}^t \left\{ \frac{\partial}{\partial\mathbf{u}} ((\boldsymbol{\tau}\vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \\ &- \int_{\Gamma_w} \sum_{i=1}^2 \delta\mathbf{u}_{x_i}^t \left\{ \frac{\partial}{\partial\mathbf{u}_{x_i}} ((\boldsymbol{\tau}\vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \\ &= \int_{\Omega} \psi^t \left\{ \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[\frac{\partial\mathbf{f}_i^c}{\partial\mathbf{u}} - \frac{\partial\mathbf{f}_i^v}{\partial\mathbf{u}} \right] \delta\mathbf{u} + \sum_{i,j=1}^2 \frac{\partial}{\partial x_j} \left[\frac{\partial\mathbf{f}_i^v}{\partial\mathbf{u}_{x_j}} \right] \delta\mathbf{u}_{x_j} \right\} d\Omega\end{aligned}\quad (2.57)$$

The derivative terms are moved onto ψ by applying Gauss' divergence theorem on the volume integral on the right hand side of equation (2.57) so that we obtain

$$\begin{aligned}&\int_{\Gamma_w} \delta\mathbf{u}^t \left\{ \frac{\partial p}{\partial\mathbf{u}} \right\} \vec{\theta} \cdot \vec{n} - \delta\mathbf{u}^t \left\{ \frac{\partial}{\partial\mathbf{u}} ((\boldsymbol{\tau}\vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \\ &- \int_{\Gamma_w} \sum_{i=1}^2 \delta\mathbf{u}_{x_i}^t \left\{ \frac{\partial}{\partial\mathbf{u}_{x_i}} ((\boldsymbol{\tau}\vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \\ &= \int_{\Omega} \delta\mathbf{u}^t \left\{ - \sum_{i=1}^2 \left[\frac{\partial\mathbf{f}_i^c}{\partial\mathbf{u}} - \frac{\partial\mathbf{f}_i^v}{\partial\mathbf{u}} \right]^t \frac{\partial\psi}{\partial x_i} - \sum_{i,j=1}^2 \left[\frac{\partial\mathbf{f}_j^v}{\partial\mathbf{u}_{x_i}} \right]^t \frac{\partial^2\psi}{\partial x_i^2} \right\} d\Omega \\ &+ \int_{\Gamma} \delta\mathbf{u}^t \left\{ \left[\frac{\partial\mathbf{f}_n^c}{\partial\mathbf{u}} - \frac{\partial\mathbf{f}_n^v}{\partial\mathbf{u}} \right]^t \psi + \sum_{i=1}^2 \left[\frac{\partial\mathbf{f}_n^v}{\partial\mathbf{u}_{x_i}} \right]^t \frac{\partial\psi}{\partial x_i} \right\} d\Gamma \\ &- \int_{\Gamma} \sum_{i=1}^2 \delta\mathbf{u}_{x_i}^t \left\{ \left[\frac{\partial\mathbf{f}_n^v}{\partial\mathbf{u}_{x_i}} \right]^t \psi \right\} d\Gamma\end{aligned}\quad (2.58)$$

By matching the boundary integrals with the integral terms that define the target functional, we derive a set of equations that describe ψ , and the corresponding boundary conditions. Similarly to the derivation of the adjoint equations for compressible inviscid flows, we split the boundary, Γ , into the wall boundary,

Γ_w , and the far-field boundary Γ_f . Equation (2.58) now reads

$$\begin{aligned}
& \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} \vec{\theta} \cdot \vec{n} - \delta \mathbf{u}^t \left\{ \frac{\partial}{\partial \mathbf{u}} ((\boldsymbol{\tau} \vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \\
& - \int_{\Gamma_w} \sum_{i=1}^2 \delta \mathbf{u}_{x_i}^t \left\{ \frac{\partial}{\partial \mathbf{u}_{x_i}} ((\boldsymbol{\tau} \vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \\
& = \int_{\Omega} \delta \mathbf{u}^t \left\{ - \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}} \right]^t \frac{\partial \psi}{\partial x_i} - \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \frac{\partial^2 \psi}{\partial x_i^2} \right\} d\Omega \\
& + \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}} \right]^t \psi + \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_j}} \right]^t \frac{\partial \psi}{\partial x_i} n_i \right\} d\Gamma \\
& - \int_{\Gamma_w} \sum_{i=1}^2 \delta \mathbf{u}_{x_i}^t \left\{ \left[\frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}_{x_i}} \right]^t \psi \right\} d\Gamma \\
& + \int_{\Gamma_f} \delta \mathbf{u}^t \left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}} \right]^t \psi + \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_j}} \right]^t \frac{\partial \psi}{\partial x_i} n_i \right\} d\Gamma \\
& - \int_{\Gamma_f} \sum_{i=1}^2 \delta \mathbf{u}_{x_i}^t \left\{ \left[\frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}_{x_i}} \right]^t \psi \right\} d\Gamma \tag{2.59}
\end{aligned}$$

From (2.59), we identify terms to cancel given that the perturbations $\delta \mathbf{u}$, $\delta \mathbf{u}_{x_1}$ and $\delta \mathbf{u}_{x_2}$ are arbitrary. The first integral term on the right-hand side of equation (2.59) vanishes and therefore we state that the integrand within the curly brackets needs to be zero.

$$\hat{\mathcal{R}}(\psi) = - \sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}} \right]^t \frac{\partial \psi}{\partial x_i} - \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \frac{\partial^2 \psi}{\partial x_i^2} = 0; \quad \psi \in \Omega \tag{2.60}$$

Equation (2.60) presents the *adjoint equations* to the compressible Navier-Stokes equations. The boundary conditions for the adjoint equations are derived in the next section using the remaining boundary integral terms in equation (2.59).

2.2.4 Boundary conditions for the adjoint Navier-Stokes equations

The boundary conditions for the adjoint equations are derived by matching the boundary terms in equation (2.59) and by considering that the perturbations $\delta \mathbf{u}$, $\delta \mathbf{u}_{x_1}$ and $\delta \mathbf{u}_{x_2}$ are arbitrary.

Adjoint wall boundary conditions

Equation (2.59) illustrates that the second integral term on the left hand side has to match the third integral term on the right hand side.

$$\sum_{i=1}^2 \left\{ \left[\frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}_{x_i}} \right]^t \boldsymbol{\psi} \right\} \Big|_{\Gamma_w} = \sum_{i=1}^2 \left\{ \frac{\partial}{\partial \mathbf{u}_{x_i}} ((\boldsymbol{\tau} \vec{n}) \cdot \vec{\theta}) \right\} \Big|_{\Gamma_w} \quad (2.61)$$

For compressible viscous flow problems, we impose a no-slip wall boundary condition. We can also impose two types of wall boundary conditions for the temperature: either the isothermal wall condition presented in equation (2.12) or the adiabatic wall boundary condition presented in (2.13). The boundary conditions for the adjoint problem vary depending on which wall condition is set for the temperature. We start by writing the viscous flux normal to the wall as

$$\mathbf{f}_n^v|_{\Gamma_w} = \left\{ \begin{array}{c} 0 \\ \boldsymbol{\tau} \vec{n} \\ k \left(\sum_{i=1}^2 \frac{\partial T}{\partial x_i} n_i \right) \end{array} \right\} \Big|_{\Gamma_w} \quad (2.62)$$

Linearising this flux normal to the wall with respect to the first-order derivatives of $\mathbf{u}$ and substituting the result into equation (2.61) leads to the boundary conditions for the adjoint problem at the wall

$$\psi_2 = \theta_1 \quad \psi_3 = \theta_2 \quad \psi_4 = 0; \quad \boldsymbol{\psi} \in \Gamma_{\text{iso}} \quad (2.63)$$

The variable ψ_1 is not prescribed at the wall. To ensure that the condition for ψ_4 is correct, we consider the integral term at the wall that is multiplied by $\delta \mathbf{u}$ on the right hand side of equation (2.59) which should be equal to the integral term at the wall that is multiplied by the perturbation $\delta \mathbf{u}$ on the left hand side of equation (2.59).

$$\begin{aligned} \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi} + \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_j}} \right]^t \frac{\partial \psi}{\partial x_i} n_i \right\} d\Gamma \\ = \int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} \vec{\theta} \cdot \vec{n} - \delta \mathbf{u}^t \left\{ \frac{\partial}{\partial \mathbf{u}} ((\boldsymbol{\tau} \vec{n}) \cdot \vec{\theta}) \right\} d\Gamma \end{aligned} \quad (2.64)$$

Knowing that the no-slip boundary conditions for the governing equations are satisfied at the wall and considering equation (2.63), equation (2.64) simplifies to

$$\int_{\Gamma_w} \delta \mathbf{u}^t \left\{ \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_j}} \right]^t \frac{\partial \psi}{\partial x_i} n_i \right\} d\Gamma = 0 \quad (2.65)$$

When considering a no-slip boundary, we know that the perturbations on the velocity components are zero ($\delta u_2 = \delta u_3 = 0$). Using the definition of the Jacobians related to the viscous flux terms given in Appendix A and imposing the no-slip boundary condition ($\vec{v} = 0$), most of the entries of these Jacobians vanish and the integrand in equation (2.65) can be rewritten as

$$\begin{aligned} \delta \mathbf{u}^t \left\{ \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_j}} \right]^t \frac{\partial \psi}{\partial x_j} n_i \right\} \Big|_{\Gamma_w} &= \sum_{i=1}^2 \left[\delta u_1 \left[\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_i}} \right]_{1,4}^t + \delta u_4 \left[\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_{x_i}} \right]_{4,4}^t \right] \frac{\partial \psi_4}{\partial x_i} n_i \Big|_{\Gamma_w} \\ &= \frac{\mu}{\rho} \frac{\gamma}{Pr} (\delta u_4 - E \delta u_1) \sum_{i=1}^2 \frac{\partial \psi_4}{\partial x_i} n_i \Big|_{\Gamma_w} = \mathbf{0} \end{aligned} \quad (2.66)$$

The subscripts in equation (2.66) denote the entry of the corresponding Jacobian matrix. For an isothermal wall condition, the wall temperature is fixed. Hence, the variation of the temperature at the wall for an isothermal boundary conditions requires that $\delta T_w = 0$ since this quantity is fixed. Using equation (2.12), we approximate the variation of the wall temperature as

$$\delta T_w \approx \frac{\partial T_w}{\partial \mathbf{u}} \delta \mathbf{u} = \frac{\gamma - 1}{R \rho^2} (\delta u_4 - E \delta u_1) = 0 \quad (2.67)$$

This confirms that the isothermal wall conditions automatically satisfies equation (2.66). However, when adiabatic wall boundary conditions are considered, the term presented in equation (2.66) is not equal to zero. In order to ensure the cancelation of the term presented in equation (2.66), we state that the normal gradient of ψ_4 with respect to the wall needs to be zero. Hence, the following adjoint boundary conditions need to be considered when dealing with an adiabatic wall.

$$\psi_2 = \theta_1 \quad \psi_3 = \theta_2 \quad \sum_{i=1}^2 \frac{\partial \psi_4}{\partial x_i} n_i = 0; \quad \psi \in \Gamma_{\text{adia}} \quad (2.68)$$

Adjoint far-field boundary conditions

For deriving the far-field boundary conditions, we apply the same approach. Again, there are no terms on the left hand side of equation (2.59) that are defined at the far-field boundary. In this case, the various terms related to the arbitrary perturbations $\delta \mathbf{u}$ and its first order derivatives need to vanish. From equation (2.59) follows

$$\int_{\Gamma_f} \sum_{i=1}^2 \delta \mathbf{u}_{x_i}^t \left\{ \left[\frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}_{x_i}} \right]^t \psi \right\} d\Gamma = \mathbf{0} \quad (2.69)$$

To satisfy equation (2.69), we impose

$$\boldsymbol{\psi} = \mathbf{0}; \quad \mathbf{u} \in \Gamma_f \quad (2.70)$$

at the far-field boundary. Furthermore, we have a term at far-field boundary that is multiplied by the perturbations $\delta \mathbf{u}$

$$\int_{\Gamma_f} \delta \mathbf{u}^t \left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi} + \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_i^v}{\mathbf{u}_{x_j}} \right]^t \frac{\partial \boldsymbol{\psi}}{\partial x_i} n_i \right\} d\Gamma = \mathbf{0} \quad (2.71)$$

The fact that $\boldsymbol{\psi} = \mathbf{0}$ at the far-field together with

$$\sum_{i=1}^2 \frac{\partial \boldsymbol{\psi}}{\partial x_i} n_i = \mathbf{0}; \quad \mathbf{u} \in \Gamma_f \quad (2.72)$$

results in the cancelation of the terms multiplied by $\delta \mathbf{u}$ at the far-field boundary.

Chapter 3

Discontinuous Galerkin spectral/ hp element method

This chapter gives an overview of the high-order spectral element discretisation that is used to solve for the governing and adjoint equations that have been introduced in the previous chapter. A general description of the spatial discretisation is given. In this chapter, we solely discuss the spatial discretisation since we only consider steady flow problems. We use an explicit time integration scheme, typically a four-stage Runge Kutta method, to calculate the steady state of the flow and the adjoint. At the end of this chapter, the implemented high-order adjoint solver is verified. This is done initially by comparing results with an analytical solution for a quasi one-dimensional problem. Furthermore, two-dimensional lift adjoint solutions for transonic flow past a wing are compared with lift adjoint solutions obtained using the DLR/TAU code.

3.1 Spatial discretisation

The high-order spectral element approach essentially relies on the same mathematical foundation as the standard finite element approach. The classical finite element approach originates from the fact that an exact solution is numerically approximated by replacing an infinite expansion by a finite representation. Generally, we are seeking a solution, denoted by $\mathbf{u}$, which is governed by conservation laws, like the ones described in the previous chapter. In this chapter, we consider steady state solutions. The steady state to a general set of conservation laws is written as

$$\mathcal{R}(\mathbf{u}) = 0 \tag{3.1}$$

where $\mathcal{R}(\circ)$ is a differential operator prescribing the transport of the solution $\mathbf{u}$. If we multiply this differential operator with a general test function, v , the method

of weighted residuals dictates that

$$\int_{\Omega} \mathcal{R}(\mathbf{u}) v \, d\Omega = 0 \quad (3.2)$$

where Ω represents the computational domain and equation (3.2) is true for arbitrary v . Equation (3.2) is the fundamental mathematical statement which the spatial discretisation in the current work relies on. Using the same arguments as for the standard finite element method, we decompose the solution domain into small subdomains, also referred to as elements, so that the mathematical operations can take place on a local elemental basis. However, instead of using a linear approximation like the classical finite element approach, we now use higher-order polynomial functions. A sketch of a typical two-dimensional high-order spectral/ hp discretisation is depicted in figure 3.1.

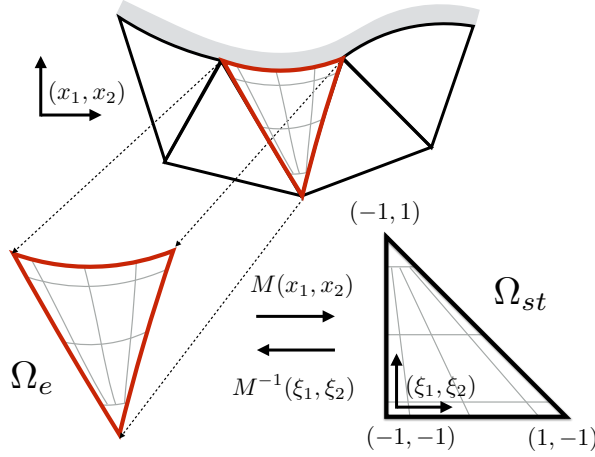


Figure 3.1: The mapping that relates the local coordinates (x, y) to the reference coordinates, (ξ_1, ξ_2) .

For the discretisation, we subdivide the computational domain, Ω , into N_{el} finite non-overlapping elements so that $\Omega = \bigcup_{e=1}^{N_{el}} \Omega_e$ where $\bigcap_{e=1}^{N_{el}} \Omega_e = \emptyset$. Here, Ω_e refers to the part of the computational domain that is covered by the e^{th} element. Using the finite element approach, the domain is discretised into elements and the local element coordinates, defined as $\mathbf{x} = (x_1, \dots, x_N)$ where $\mathbf{x} \in \Omega_e$ and N indicates the number of spatial dimensions, are mapped onto standard element coordinates, defined as $\boldsymbol{\xi} = (\xi_1, \dots, \xi_N)$ where $\boldsymbol{\xi} \in \Omega_{st}$ for which holds that

$$\Omega_{st} = -1 \leq \xi_1, \dots, \xi_N \leq 1 \quad (3.3)$$

Using a standard element, we define a parametric mapping $\mathcal{M}_e(\boldsymbol{\xi})$ and its inverse $\mathcal{M}_e^{-1}(\mathbf{x})$ so that we map the solution from the standard element Ω_{st} to any

elemental domain Ω_e and vice versa.

$$\mathbf{x} = \mathcal{M}_e(\boldsymbol{\xi}) \quad \boldsymbol{\xi} = \mathcal{M}_e^{-1}(\mathbf{x}) \quad (3.4)$$

The integral over the domain Ω , given in equation (3.2), is now expanded into the sum of integrals over a collection of N_{el} non-overlapping elements. we write equation (3.2) as

$$\sum_{e=1}^{N_{el}} \int_{\Omega_e} \mathcal{R}(\mathbf{u}_e^\delta) v_e^\delta d\Omega_e = 0 \quad (3.5)$$

where $\mathbf{u}_e^\delta$ and v_e^δ are the representation of the solution and the test function respectively within element e . The solution, $\mathbf{u}_e^\delta$, and test function, v_e^δ , are approximated using a set of polynomial functions.

There are generally two types of expansions: the nodal and modal type. The nodal coefficients represent the approximate solution at a given set of nodes, unlike the collocation method for which the equations are being solved exactly at the collocation points. For modal expansions holds that the expansion set of $P - 1$ is contained within the expansion set of order P . Therefore the modal expansion is also referred to as the hierarchical expansion [51].

The choice for the appropriate expansion set is mainly driven by two factors. First, the computational cost of creating the matrix system from the defined expansion set which involves numerical integration and differentiating and second, the computational cost of inverting this matrix system. Typically the latter one is dominant and therefore the conditioning of the matrix system is very important.

A set of semi-orthogonal basis functions is chosen which we refer to as the modified basis. The modified basis is constructed by splitting the orthogonal basis functions in an internal and boundary contribution where the internal basis functions are orthogonal and zero on the boundary and the boundary contributions are non-zero on the boundary [51].

This modified semi-orthogonal basis functions were initially proposed to preserve C^0 continuity so that it could be used as a CG discretisation. The orthogonality of the global expansion was not completely destroyed by just changing the boundary expansions to linear functions in order to enforce C^0 continuity. Here we consider a DG discretisation in this thesis so this is not requirement anymore. However, a fully orthogonal expansion basis has P^2 contributions on the boundary while the modified expansion basis only has P contributions on the boundary. This makes the modified expansion basis more efficient when evaluating the boundary terms. Based on this argument, it was chosen to use a modified version of a modal expansion set [12].

We start by denoting that the space of all polynomials of degree P are defined

on the standard element Ω_{st} . This set of polynomials is used to approximate both $\mathbf{u}_e^\delta$ and v_e^δ . In this way, all mathematical operations are performed in Ω_{st} and we use the previously introduced parametric mapping to obtain the representation of the solution in physical space. The chosen modal expansion basis is constructed from a family of orthogonal Jacobi polynomials generally denoted by $\mathcal{P}_P^{\alpha,\beta}$. Jacobi polynomials are solutions to the singular Sturm-Liouville problem as outlined in [51]. The Jacobi polynomials are orthogonal which is an important property that can be exploited to construct an efficient high-order discretisation. They satisfy

$$\int_{-1}^1 (1-\xi)^\alpha (1+\xi)^\beta \mathcal{P}_p^{\alpha,\beta}(\xi) \mathcal{P}_q^{\alpha,\beta}(\xi) d\xi = C \delta_{pq} \quad (3.6)$$

where p and q indicate the orders of the polynomial, C is a function of α , β and the order P and δ_{pq} represents the Kronecker delta.

In this thesis, we are considering two- and three-dimensional unstructured meshes. The multidimensional expansions are constructed by the product of the one-dimensional tensors defined in each Cartesian direction. Therefore, we first define the one-dimensional modal C^0 expansion basis.

$$\varphi_p^a(\xi) = \begin{cases} \frac{1-\xi}{2} & p = 0 \\ \frac{1-\xi}{2} \frac{1+\xi}{2} \mathcal{P}_{p-1}^{1,1}(\xi) & 1 \leq p < P \\ \frac{1+\xi}{2} & p = P \end{cases} \quad (3.7)$$

The modified principle basis functions for $P = 5$ are depicted in figure (3.2).

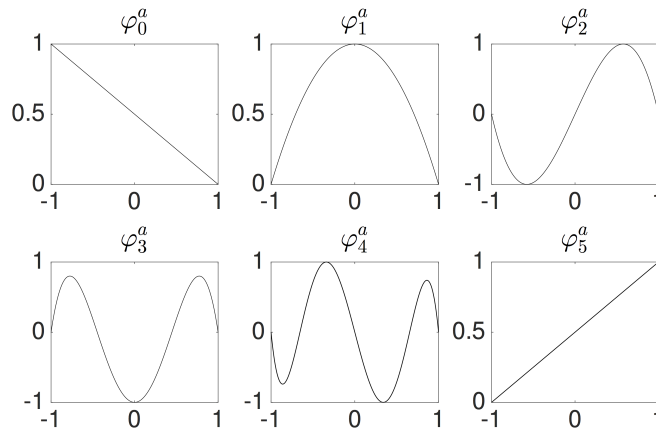


Figure 3.2: The one-dimensional modified modal basis functions for $P = 5$.

For a triangular element, the coordinates in the standard region are collapsed [51]. Therefore we define a different expansion basis in vertical direction for the

triangular element.

$$\varphi_{pq}^b(\xi) = \begin{cases} \varphi_q^a(\xi) & p = 0, 0 \leq q \leq Q \\ \left(\frac{1-\xi}{2}\right)^{p+1} & 1 \leq p \leq P, q = 0 \\ \left(\frac{1-\xi}{2}\right)^{p+1} \frac{1+\xi}{2} P_{q-1}^{2p+1,1}(\xi) & 1 \leq p < P, 1 \leq q < Q \\ \varphi_q^a(\xi) & p = P, 0 \leq q \leq Q \end{cases} \quad (3.8)$$

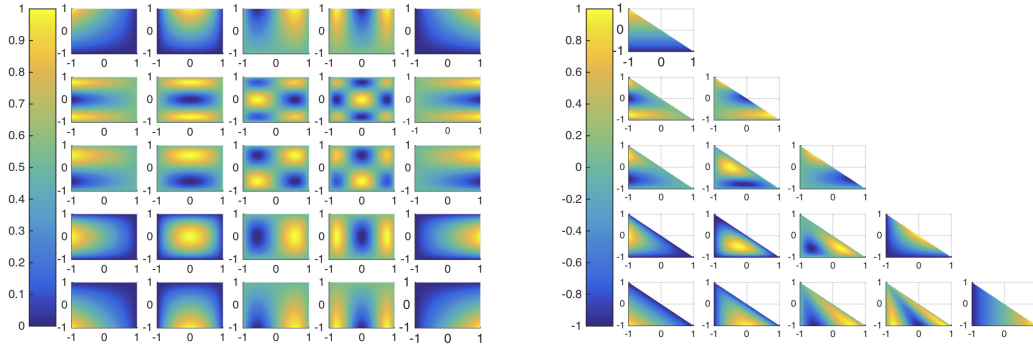
We use modal basis functions for each element type which allows us to exploit a tensor product expansion of the basis functions. The expansion basis for a quadrilateral and triangular elements becomes the product of the previously introduced one-dimensional modified modal expansion modes. For a quadrilateral element we obtain

$$\phi_e^{pq}(\xi_1, \xi_2) = \varphi_p^a(\xi_1) \varphi_q^a(\xi_2) \quad (3.9)$$

while for a triangular element we write

$$\phi_e^{pq}(\xi_1, \xi_2) = \varphi_p^a(\xi_1) \varphi_{pq}^b(\xi_2) \quad (3.10)$$

The modified modal expansions for both quadrilateral and triangular elements for $P = 4$ are depicted in figure 3.3.



(a) The modified modal basis functions ($P = 4$) for a quadrilateral element. (b) The modified modal basis functions ($P = 4$) for a triangular element.

Figure 3.3: The modified modal basis functions for quadrilateral and triangular elements.

For the Galerkin method, the solution $\mathbf{u}_e^\delta$, and test function v_e^δ within each element are approximated using the same set of basis functions. Hence, the solution within each element is represented in terms of the local expansion functions $\phi_e^{pq}(\boldsymbol{\xi})$ by

mapping the standard element Ω_{st} to each elemental domain Ω_e . In this way, the mathematical operations are carried out in a more efficient way. For a two-dimensional problem, the solution is written as

$$u_e^\delta(\mathbf{x}(\xi_1, \xi_2)) = \sum_{p=0}^P \sum_{q=0}^Q \phi_e^{pq}(\xi_1, \xi_2) \tilde{u}_e^{pq} \quad (3.11)$$

and the test function as

$$v_e^\delta(\mathbf{x}(\xi_1, \xi_2)) = \phi_e^{pq}(\xi_1, \xi_2) \quad (3.12)$$

The degrees of freedom that we are solving for are given by $\tilde{u}_e^{pq}$ for each element e .

3.2 Discretisation of the governing equations

The compressible flow equations are non-linear and the speed of propagation depends on the solution. As a results, the solution has local discontinuities in flow properties (shocks and contact discontinuities) once the Mach number is sufficiently high. Generally, we will obtain a piecewise smooth solution. Therefore it seems appropriate to use a function space that allows for discontinuous functions. The difference between the normal (continuous) and the discontinuous Galerkin approach is that for the discontinuous Galerkin method it is allowed to have a discontinuity at the elemental interfaces. A sketch is given in figure 3.4.

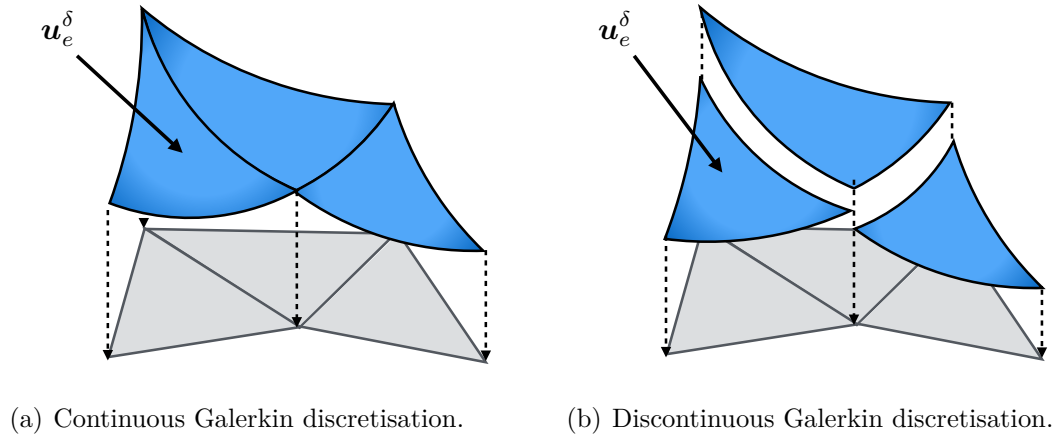


Figure 3.4: Comparison between the high-order continuous and discontinuous Galerkin discretisation.

Figure 3.4 depicts a general element-wise polynomial approximation of an arbitrary solution, $\mathbf{u}_e^\delta$, using a two-dimensional mesh (grey triangles). In figure 3.4(a), the solution is continuous between the elements while in figure 3.4(b) the solution

is discontinuous.

The discontinuous Galerkin formulation has some benefits. Coupling between the elements is ensured by computing the fluxes on each element interface by solving a local Riemann problem based on the solution points in the local element and the adjacent element. In this way, information propagates from one element to the other element and the solution is determined within a space of bounded variation. Furthermore, it is relatively easy to implement the discontinuous Galerkin scheme in parallel since the only connectivity occurs at the elemental interfaces.

The discontinuous Galerkin finite element method is designed to combine the beneficial features of local high-order approximation and high flexibility, which are typical features of the finite element method, with the local statement of the governing equations, which is a feature of the finite volume method.

The discontinuous Galerkin discretisation is applied to the compressible Euler and Navier-Stokes equations and also to their corresponding adjoint equations. A description of the DG discretisation for the flow and adjoint equations is given in the following sections.

3.2.1 Compressible Euler equations

In this section, the high-order discontinuous Galerkin discretisation for the compressible Euler equations is discussed. First, using the standard finite element approach we apply the method of weighted residuals (see equation (3.5)) and we evaluate the integral term using Gauss' divergence theorem

$$\int_{\Omega} v \sum_{i=1}^2 \frac{\partial}{\partial x_i} \mathbf{f}_i^c(\mathbf{u}) d\Omega = - \int_{\Omega} \sum_{i=1}^2 \frac{\partial v}{\partial x_i} \mathbf{f}_i^c(\mathbf{u}) d\Omega + \int_{\Gamma} v \mathbf{f}_n^c(\mathbf{u}) d\Gamma = 0 \quad (3.13)$$

where the two-dimensional convective flux normal to the boundary is given by

$$\mathbf{f}_n^c(\mathbf{u}) = \sum_{i=1}^2 \mathbf{f}_i^c(\mathbf{u}) n_i \quad (3.14)$$

We decompose the computational domain in non-overlapping elements and we approximate the test function and solution within each element using equation (3.12) and (3.11). We obtain the discontinuous Galerkin discrete form of the governing equations when we write

$$\sum_{e=1}^{N_{el}} \left\{ - \int_{\Omega_e} \sum_{i=1}^2 \frac{\partial \phi_e}{\partial x_i} \mathbf{f}_i^c(\mathbf{u}_e^\delta) d\Omega + \int_{\Gamma_e} \phi_e \mathbf{f}_n^c(\mathbf{u}_e^\delta) d\Gamma \right\} = 0 \quad (3.15)$$

where Ω_e represents the volume inside an element and Γ_e represents the surface boundary of the element. The solution is allowed to be discontinuous at the inter-

faces between the elements. As a result, the normal flux at an internal interface, $\mathbf{f}_n^c(\mathbf{u}_e^\delta)$, is defined twice. The boundary fluxes will be appropriately calculated using a suitable convective numerical flux function.

Convective numerical flux

We use the inviscid characteristic analysis [89] to calculate the appropriate convective flux at the elemental interfaces. The correct values for $\mathbf{f}_n^c(\mathbf{u})$ are calculated using a one-dimensional Riemann problem, in normal direction, at the interface between two elements. When solving for a multi-dimensional problem, the local solution is rotated in the normal direction with respect to the given interface. We then solve the one-dimensional Riemann problem at the interface and rotate the solution back into the original reference frame. In figure 3.5, a sketch is shown of the nomenclature that is used during the discussion of the convective and diffusive fluxes. Figure 3.5 depicts an interface edge that has two adjacent elements. At

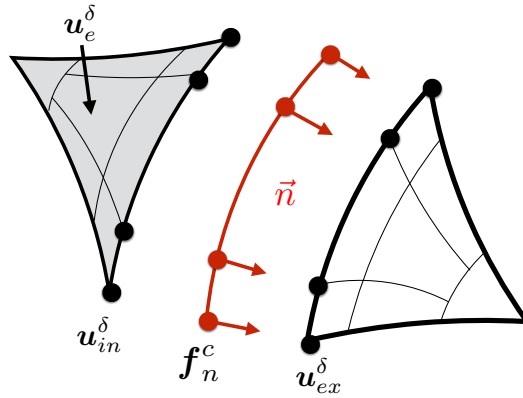


Figure 3.5: Illustration of the flux for an interface for a two-dimensional case.

this interface, we have two states indicated by the subscripts *in* and *ex*. The grey element is the e^{th} element for which the boundary integral is evaluated in order to determine $\mathbf{u}_e^\delta$. Its own adjacent interface solution is denoted by $\mathbf{u}_{in}^\delta$. The other adjacent interface solution is denoted by $\mathbf{u}_{ex}^\delta$. The normal flux at the interface $\mathbf{f}_n^c(\mathbf{u})$ is calculated by solving a Riemann problem for the points located on the interface. For this particular problem, we find an exact or an approximate solution using a Riemann solver. An extensive overview of the available Riemann solvers is given by Toro in [89]. The boundary term given in equation (3.15) is replaced by a Riemann flux function that depends on the two states and the normal vector.

$$\mathbf{f}_n^c(\mathbf{u}_e^\delta)|_{\Gamma_e} = \mathcal{H}_c(\mathbf{u}_{ex}^\delta, \mathbf{u}_{in}^\delta; \vec{n}) \quad (3.16)$$

where $\mathbf{u}_{ex}^\delta$ and $\mathbf{u}_{in}^\delta$ represent the solution at the external and internal solutions with respect to each interface as shown in figure 3.5. The numerical flux function at the interfaces allows for the propagation of information from one element to the other. A set of different Riemann flux solvers like the approximate Roe Riemann solver, the Harten-Lax-van Leer-contact (HLLC) solver and the exact Riemann flux solver are implemented in Nektar++ [12, 67].

3.2.2 Compressible Navier-Stokes equations

To derive the DG discretisation of the compressible Navier-Stokes equations, we follow the same approach as Bassi and Rebay [8]. The following mixed formulation is introduced

$$\begin{aligned} \mathbf{g} - \nabla \mathbf{q} &= \mathbf{0} \\ \sum_{i=1}^2 \frac{\partial}{\partial x_i} \{ \mathbf{f}_i^c(\mathbf{u}) - \mathbf{f}_i^v(\mathbf{u}, \mathbf{g}) \} &= \mathbf{0} \end{aligned} \quad (3.17)$$

where we first calculate the first order derivatives of the auxiliary variables $\mathbf{g} = \nabla \mathbf{q} = \left\{ \frac{\partial \mathbf{q}}{\partial x_1}, \frac{\partial \mathbf{q}}{\partial x_2} \right\}$ where $\mathbf{q} = \{v_1, v_2, T\}^t$ since the diffusive fluxes are defined by $\nabla \mathbf{q}$. Discretising the first equation of the coupled system gives

$$\sum_{e=1}^{N_{el}} \left\{ \int_{\Omega_e} \phi_e \mathbf{g}_e^\delta d\Omega - \int_{\Gamma_e} \phi_e \mathbf{q}_e^\delta \vec{n} d\Gamma + \int_{\Omega_e} \nabla \phi_e \mathbf{q}_e^\delta d\Omega \right\} = \mathbf{0} \quad (3.18)$$

Once a solution for $\mathbf{g}_e^\delta$ is computed, we substitute it into the discretised form of the second equation of the coupled system which is given by

$$\begin{aligned} & - \sum_{e=1}^{N_{el}} \int_{\Omega_e} \sum_{i=1}^2 \frac{\partial \phi_e}{\partial x_i} \{ \mathbf{f}_i^c(\mathbf{u}_e^\delta) - \mathbf{f}_i^v(\mathbf{u}_e^\delta, \mathbf{g}_e^\delta) \} d\Omega \\ & + \sum_{e=1}^{N_{el}} \int_{\Gamma_e} \phi_e \{ \mathbf{f}_n^c(\mathbf{u}_e^\delta) - \mathbf{f}_n^v(\mathbf{u}_e^\delta, \mathbf{g}_e^\delta) \} d\Gamma = \mathbf{0} \end{aligned} \quad (3.19)$$

We use a Riemann flux function for the discretisation of the convective fluxes on the interfaces. The discretisation of the diffusive fluxes on the interfaces (see integrand $\mathbf{u}_e^\delta \vec{n}$ in equation (3.18) and integrand $\mathbf{f}_n^v(\mathbf{u}_e^\delta, \mathbf{g}_e^\delta)$ in equation (3.19)) is discussed below.

Diffusive numerical flux

The integrands $\mathbf{q}_e^\delta \vec{n}$ in equation (3.18) and $\mathbf{f}_n^v(\mathbf{u}_e^\delta, \mathbf{g}_e^\delta)$ in equation (3.19) that follow from applying Gauss' divergence theorem, need to be replaced by numerical flux functions. In the literature, there are several options available to deal with

these terms.

There are several methods that were proposed over the years to calculate the viscous fluxes on the interfaces. One is the Baumann-Oden flux [9], which is a consistent version of the numerically stable but inconsistent flux term that is proposed by Shu [83] and by Zhang and Shu [100], the Bassi and Rebay diffusive fluxes function [8] takes the average of the internal and external values and the result is used for the flux values. We have adopted the local discontinuous Galerkin (LDG) flux function, introduced by Cockburn and Shu [20] because of its uniform convergence rate for all polynomial orders [51] and its computational efficiency and accuracy in $\mathbf{q}$ and $\mathbf{g}$.

This approach alternates the left and right states to calculate the viscous flux at the interface. Hence, following the mixed formulation of the LDG, we use the external state $\mathbf{q}_{ex}$ to calculate the integrand

$$\mathbf{q}_e^\delta \vec{n} \Big|_{\Gamma_e} = \mathbf{q}_{ex}^\delta \vec{n} \Big|_{\Gamma_e} \quad (3.20)$$

and the internal state to calculate $\mathbf{f}_n^v(\mathbf{u}_e^\delta, \mathbf{g}_e^\delta)$ at the interfaces. Hence,

$$\mathbf{f}_n^v(\mathbf{u}_e^\delta, \mathbf{g}_e^\delta) \Big|_{\Gamma_e} = \mathbf{f}_n^v(\mathbf{u}_{in}^\delta, \mathbf{g}_{in}^\delta) \Big|_{\Gamma_e} \quad (3.21)$$

In addition to these advantages, the LDG flux function is very suitable for a discontinuous Galerkin discretisation since the boundary conditions can be imposed weakly. The values of the first order derivatives are not defined for the ghost state when a Dirichlet boundary condition is considered. However, this is not required when evaluating the diffusive flux at a boundary of the computational domain since $\mathbf{g}$ is taken from the internal values $\mathbf{g}_{in}$. The state variables $\mathbf{u}$ are generally known at the external state at boundary and we use the external values $\mathbf{u}_{ex}$ since they are defined by the defined boundary condition.

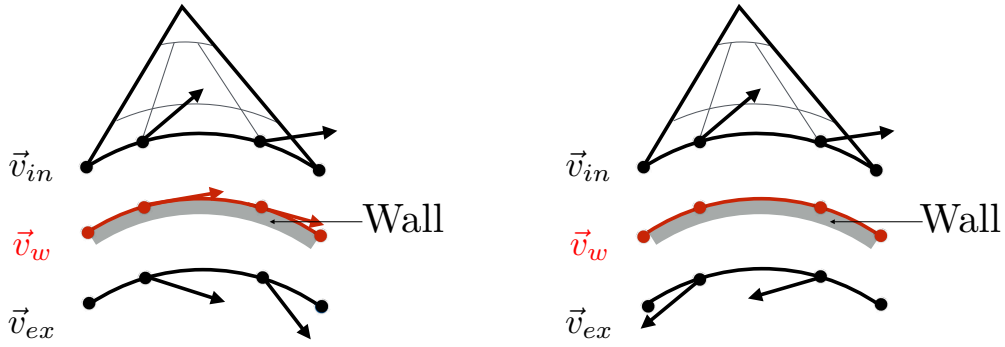
3.2.3 Boundary conditions

In this section, we discuss the discretisation of the boundary conditions that were introduced in section 2.1.3. For the discontinuous Galerkin discretisation of the governing equations, the boundary conditions are imposed weakly through the solution of the Riemann problem. As mentioned in the previous section, during the discussion of the numerical fluxes, we have two sets of solution points at the interface between two elements. However at the physical boundary we only have one adjacent element. Therefore, we use a ghost interface that resembles an exterior elemental interface. The values at the ghost interface are adjusted so that the correct boundary fluxes are evaluated when applying the Riemann solver.

Wall boundary conditions

When using the compressible Euler equations, we need to impose a slip boundary condition at the wall meaning that the normal component of the velocity vector should be equal to zero. In the following analysis, the ghost interface is equivalent to the external state. We impose a velocity vector $\vec{v}_{ex}$ at the ghost interface so that the resulting velocity vector, which is obtained after evaluating the Riemann solver, has a normal component that is zero ($\vec{v} \cdot \vec{n} = 0$).

In figure 3.6(a), a sketch is given of the boundary treatment for a slip boundary condition for a two-dimensional problem. We use equation (3.16) and apply the



(a) Illustration of the imposition of the weak slip boundary conditions for inviscid compressible flow. (b) Illustration of the imposition of the weak no-slip boundary conditions for viscous compressible flow.

Figure 3.6: Illustration of the imposition of the wall boundary conditions for the governing equations

same analogy for the wall boundary. In this work, we copy the values of the density and total energy from the elemental interface onto the ghost interface. To ensure that the tangential component is unchanged and the wall normal component of the velocity is zero, we set the velocity vector on the ghost interface equal to

$$\vec{v}_{ex} = \vec{v}_{in} - 2(\vec{v}_{in} \cdot \vec{n}) \vec{n} \quad (3.22)$$

and therefore, the external state is defined as

$$\mathbf{u}_{ex}^\delta = \left\{ \begin{array}{c} \rho \\ \rho(v_1 - 2(\vec{v} \cdot \vec{n})n_1) \\ \rho(v_2 - 2(\vec{v} \cdot \vec{n})n_2) \\ \rho E \end{array} \right\}_{in} \quad (3.23)$$

The same is done for the no-slip boundary condition. However, now the velocity

at the wall should be zero and therefore, the velocity vector at the ghost interface are the exact opposite of the velocity vectors at the element interface as depicted in figure 3.6(b). We write

$$\mathbf{u}_{ex}^\delta = \left\{ \begin{array}{c} \rho \\ -\rho v_1 \\ -\rho v_2 \\ \rho E \end{array} \right\}_{in} \quad (3.24)$$

Depending on whether we are dealing with a slip or no-slip boundary conditions, we construct the ghost interface as shown in equation (3.23) and (3.24). We substitute the internal and external fluxes and calculate the flux at the boundary using equation (3.16).

Far-field boundary conditions

To impose the far-field boundary condition, we consider again two states at the far-field boundary, one inside and one outside (ghost state) of the computational domain and the flux at the far-field is determined using a Riemann flux function. Based on the desired far-field conditions, $\mathbf{u}|_{\Gamma_\infty}$, the correct values for the state variables at the ghost interface is determined. The solution at a subsonic outflow boundary of the computational domain is not known *a priori* and therefore we use the inviscid characteristic analysis (Riemann invariants) to determine the values at the outflow. Based on the propagation of the Riemann invariants, we determine the correct values of the state variables at the inflow and outflow boundary. A short overview of the characteristic analysis for inflow and outflow boundary conditions is given in section 2.1.3 and used here to set the appropriate values in the ghost state at the far-field boundary. For more detail on this, the reader is referred to [89].

To perform the characteristic analysis at the far-field boundary, we project the local velocity vector onto a reference frame normal to the local boundary. This allows us to consider a one-dimensional Riemann problem normal to the far-field boundary as mentioned before during the discussion of the convective flux term. In figures 2.1.3 and 2.1.3, the characteristic lines corresponding to subsonic inflow and outflow are given. The Riemann invariant (J_+) corresponding to the eigenvalue $v_n + c$ is written as

$$J_+ = v_n + \frac{2c}{\gamma - 1} \quad (3.25)$$

and the Riemann invariant corresponding to the $v_n - c$ eigenvalue, which is denoted by J_- , and is defined as

$$J_- = v_n - \frac{2c}{\gamma - 1} \quad (3.26)$$

Out of simplicity, we consider only outflow here since the inflow condition is treated in a similar fashion. For subsonic outflow (see figure 2.1.3), the J_+ Riemann invariant is associated to the computational domain, indicated by the subscript in ,

$$J_{+,in} = v_{n,in} + \frac{2c_{in}}{\gamma - 1} \quad (3.27)$$

where c_{in} indicates the speed of sound within the computational domain. Considering the negative characteristic from outside the computational domain, indicated by subscript ∞ , we have that

$$J_{-,\infty} = v_{n,\infty} - \frac{2c_\infty}{\gamma - 1} \quad (3.28)$$

where c_∞ indicates the speed of sound at the far-field just outside of the computational domain. Along the characteristic lines $d\mathbf{x}/dt = v_n + c$ and $d\mathbf{x}/dt = v_n - c$ we have that J_+ and J_- are constant and therefore the following two relations need to be satisfied at the boundary interface

$$J_{+,b} = v_{n,b} + \frac{2c_b}{\gamma - 1} = J_{+,in} \quad (3.29)$$

$$J_{-,b} = v_{n,b} - \frac{2c_b}{\gamma - 1} = J_{-,\infty} \quad (3.30)$$

Substituting equations (3.27) and (3.28) into equations (3.29) and (3.30), and solving for $v_{n,b}$ and c_b , we obtain

$$v_{n,b} = -\frac{J_{-,\infty} + J_{+,in}}{2} \quad (3.31)$$

and

$$c_b = \frac{\gamma - 1}{4} (J_{-,\infty} - J_{+,in}) \quad (3.32)$$

Based on $v_{n,b}$, we construct $\vec{v}_b$ assuming that the tangential component of the velocity remains unchanged. Depending on the sign of $v_{n,b}$, we are dealing with a subsonic inflow or outflow condition. Based on the outcome of equations (3.31) and (3.32) and considering that the boundary is taken sufficiently far away enough from the geometry so that we can assume that the entropy at the boundary is equal to the free-stream entropy, we can use the same expressions as the ones given in equations (2.28)-(2.27) to calculate the s_b , ρ_b and p_b . The total energy at the far-field boundary is calculated using

$$E_b = \frac{p_b}{\rho_b(\gamma - 1)} + \frac{|\vec{v}_b|^2}{2} \quad (3.33)$$

For supersonic inflow, the values at the boundary are determined by the characteristics from the far-field. Hence, considering a supersonic inlet, we have

$$\begin{pmatrix} \rho \\ v_1 \\ v_2 \\ \rho E \end{pmatrix}_b = \begin{pmatrix} \rho \\ v_1 \\ v_2 \\ \rho E \end{pmatrix}_\infty \quad (3.34)$$

while for supersonic outflow holds that the internal characteristics determine the values at the boundary. Hence,

$$\begin{pmatrix} \rho \\ v_1 \\ v_2 \\ \rho E \end{pmatrix}_b = \begin{pmatrix} \rho \\ v_1 \\ v_2 \\ \rho E \end{pmatrix}_{in} \quad (3.35)$$

3.3 Artificial diffusion

Compressible flows are characterised by local discontinuities in flow properties, also referred to as shock waves. These discontinuities result in numerical oscillations (Gibbs phenomenon) which tend to make the numerical scheme unstable. To avoid this problem, we use a stabilising artificial diffusion term. To shorten the notation, we consider the compressible Euler equations that are given in equation (2.1) and augment them by an artificial diffusion operator to account for numerical oscillations as shown below

$$\sum_{i=1}^2 \left\{ \frac{\partial \mathbf{f}_i^c}{\partial x_i} - \mu_a(S_e) \frac{\partial^2 \mathbf{u}}{\partial x_i^2} \right\} = \mathbf{0} \quad (3.36)$$

The scaled artificial viscosity coefficient is denoted by μ_a which is dependent on an element-wise discontinuity sensor S_e . For the high-order spectral element method, Person and Peraire [76] proposed a discontinuity sensor that quantifies the smoothness of the solution within an element by comparing the solution at two different polynomial orders. The sensor is defined in the following way

$$S_e = \log_{10} \left(\frac{\|f(\mathbf{u}_e^P) - f(\mathbf{u}_e^{P-1})\|_{L_2}}{\|f(\mathbf{u}_e^P)\|_{L_2}} \right) \quad (3.37)$$

where for $f(\mathbf{u}_e^P)$ and $f(\mathbf{u}_e^{P-1})$ we choose a state variable directly or a combination of state variables like the Mach number or pressure for example. Low absolute values of S_e are associated with discontinuities and high absolute values indicate that the solution is smooth. Since we are using a modal/hierarchical expansion

basis, we determine the solution of $f(\mathbf{u}_e^{P-1})$ from the available solution $f(\mathbf{u}_e^P)$. Based on the value of the sensor, we determine the artificial viscosity coefficient μ_a which then scales the amount of artificial diffusion that is added to account for the numerical oscillations in the solution. Typically we want the artificial viscosity to vanish when we move away from the shock and therefore we use the following definition for the artificial viscosity coefficient

$$\mu_a(S_e) = \begin{cases} 0 & \text{if } S_e < s_0 - \kappa \\ \frac{\mu_{a,0}}{2} \left(1 + \sin \left(\frac{\pi(S_e - s_0)}{2\kappa} \right) \right) & \text{if } s_0 - \kappa < S_e < s_0 + \kappa \\ \mu_{a,0} & \text{if } S_e > s_0 + \kappa \end{cases} \quad (3.38)$$

where the maximum artificial viscosity coefficient is denoted by $\mu_{a,0} \sim h/p$ and $s_0 \sim 1/P^4$ and κ define the range of elements that require artificial diffusion. Biotto [12] presented a modified formulation for $\mu_{a,0}$ which is given by

$$\mu_{a,0} = \frac{\bar{h}_e}{p} \lambda(\boldsymbol{\xi})_{max}^e \quad (3.39)$$

where $\bar{h}_e$ is the average distance between the vertices defining the e^{th} element. In this way, the artificial viscosity coefficient resembles physical viscosity in terms of its dimensions.

To illustrate the application of the additional artificial diffusion term, we consider a supersonic laminar flow ($Ma = 1.2$, $Re = 1000$) past a NACA 0012 aerofoil at $\alpha = 0^\circ$. For this case, we define inlet conditions at the left boundary and a pressure outflow boundary condition on the right boundary. For the outflow, the pressure is defined by p_∞ and the velocity is extrapolated from the computational domain. Symmetry conditions are applied at the upper and lower boundary in order to avoid the application of non-reflective far-field boundary conditions which are not suitable once artificial diffusion is applied to damp the shock waves since we cannot assume the flow to be fully inviscid anymore. At the wall of the NACA 0012, we define an isothermal no-slip boundary condition.

In figure 3.7, the Mach contours are shown for the steady state solution together with the definition of the boundary conditions. This solution is characterised by a strong bow shock approximately 0.5 chord lengths upstream of the NACA 0012 profile which is shown in more detail in figure 3.8(a). The bow shock reflects both on the upper and lower symmetry boundary since the symmetry condition resembles the slip condition presented in equation (2.10). The representation of the shock near the NACA 0012 is significantly better than at the symmetry boundaries since the element size is larger at the far-field. The resolution of the solution is therefore poor at the symmetry boundaries and therefore results in a relatively thick shock

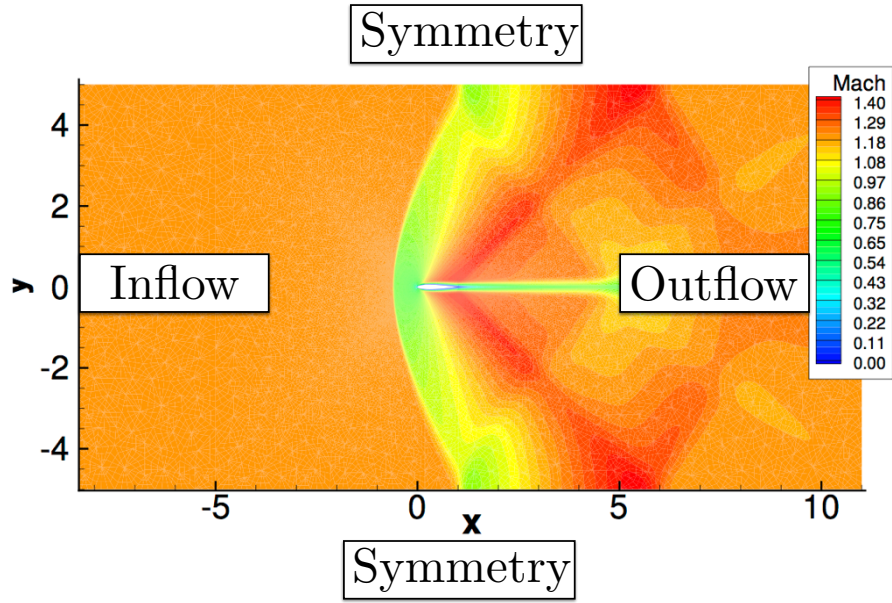
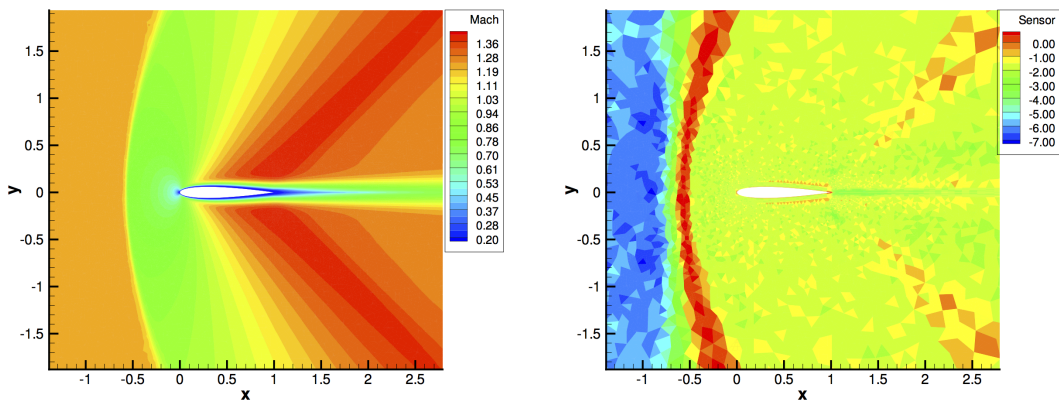


Figure 3.7: Supersonic laminar flow past a NACA 0012 aerofoil ($Ma = 1.2$, $Re = 1000$, $\alpha = 0^\circ$).

representation. Furthermore, there is a boundary layer present near the wall of the geometry.

The strong bow shock is detected using the sensor defined in expression (3.37) and the distribution of S_e is presented in figure 3.8(b). By setting the appropriate



(a) Mach solution for supersonic laminar flow past a NACA 0012 aerofoil. (b) Example of discontinuity sensor (S_e) distribution.

Figure 3.8: Detail of the solution for supersonic laminar flow past a NACA 0012 aerofoil ($Ma = 1.2$, $Re = 1000$, $\alpha = 0^\circ$).

threshold values for s_0 and κ , we select the range of elements that require artificial diffusion to damp the numerical oscillations. Hence, we use the element-wise discontinuity sensor, shown in figure 3.8(b) and equation (3.38) to obtain the correct artificial viscosity μ_a .

As an additional example to illustrate the shock capturing technique for three dimensional flows, we consider a supersonic inviscid flow ($Ma = 3.0$) past a sphere which is presented in [86]. For this case, we apply an inflow boundary condition on the curved outer boundary and wall boundary conditions at the surface of the sphere. The solution is extrapolated at the outflow. We use a structured mesh of hexahedra to calculate the steady state solution. Results are presented in figure 3.9. In figures 3.9(a) and 3.9(b), an overview of the Mach number and the artificial dissipation is given. In figures 3.9(c) and 3.9(d), the Mach solution and artificial dissipation distribution is given for the (x, y) plane. Finally, we compare the Mach profiles for different z locations with the reference data presented in [86]. In figure 3.9(e), the reference data is plotted and in figure 3.9(f), the obtained solution using Nektar++ is given. The Mach profiles correspond nicely for each z .

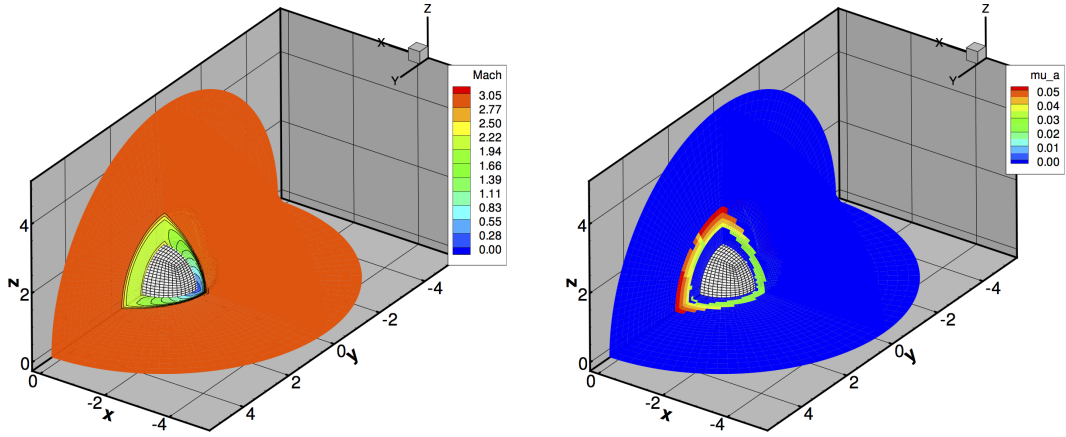
3.4 Variable polynomial order

In order to accommodate a variable polynomial order distribution across the mesh, we need to appropriately evaluate the fluxes on the elemental interfaces, since they are expressed twice using a different polynomial approximation. One needs to ensure that the fluxes are calculated using the highest number of integration points of any two adjacent elements, as shown in figure 3.10.

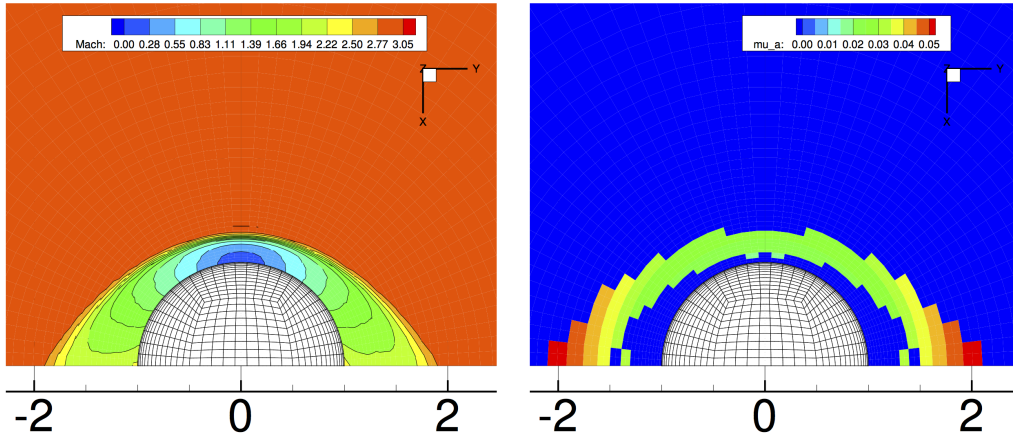
The number of integrations points is directly related to the polynomial order that is used. Numerical oscillations may occur when trying to integrate the high-order solution using the lowest number of integration points. This is explained by the fact that the element with the highest order will have unconstrained degrees of freedom [81]. Hence, to ensure conservation and stability, the continuity of the total flux is required between two adjacent elements and therefore

$$\int_{\Gamma_{fr}} \mathbf{f}_n^c(\mathbf{u}_{ex}^\delta) d\Gamma = \int_{\Gamma_{fi}} \mathbf{f}_n^c(\mathbf{u}_{in}^\delta) d\Gamma \quad (3.40)$$

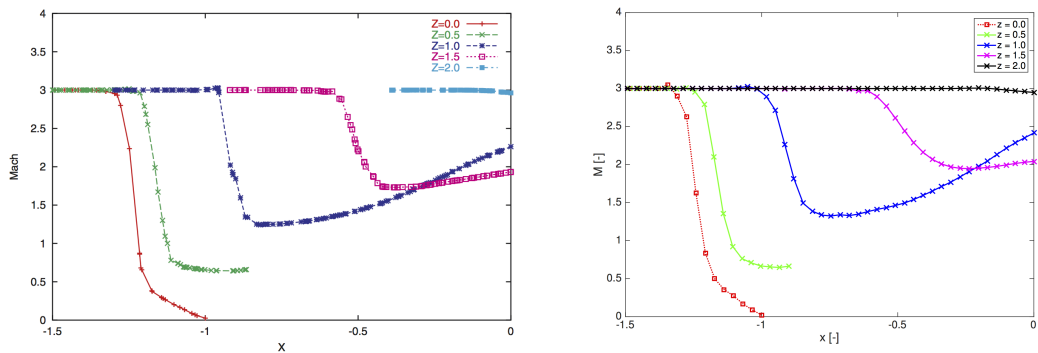
Here $\mathbf{f}_n^c(\mathbf{u}_{ex}^\delta)$ and $\mathbf{f}_n^c(\mathbf{u}_{in}^\delta)$ represent the external and internal numerical fluxes which are each approximated using a different polynomial order. Once it is determined that the order is different, the coefficients are copied directly onto the higher resolved side, but fewer coefficients have to be set on the other side. Since the modified basis defined in the previous section is not orthogonal, we instead project the higher order interface onto a space of orthogonal polynomials and filter in this



(a) Three-dimensional contour lines for the Mach solution. (b) Three-dimensional distribution of μ_a .



(c) Mach contours in the (x, y) plane for $z = 0$. (d) μ_a contours in the (x, y) plane for $z = 0$.



(e) The reference Mach number plotted against the stream wise coordinate, x , for different z positions at $y = 0$. This figure is taken from [86]. (f) The Mach number plotted against the stream wise coordinate, x , for different z positions at $y = 0$.

Figure 3.9: Supersonic inviscid flow past a sphere ($Ma = 3.0$).

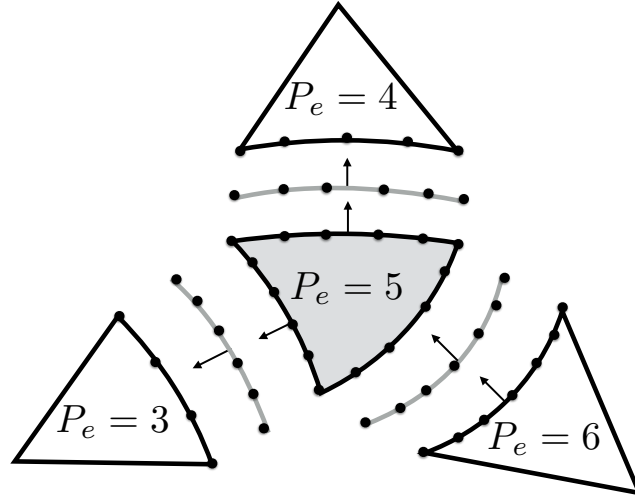


Figure 3.10: Illustration of a variable polynomial order distribution where the higher order edge is used to calculate the advective fluxes and the result is projected onto the lower order adjacent edge.

space to remove the high-order frequencies. This is done because the coefficients in orthogonal space are not coupled and in this way the high-order frequencies are removed without changing the mean solution by setting the higher order coefficients equal to zero. Once the degree of the orthogonal expansion is decreased, a reverse projection is carried out and the coefficients in the original modified basis are obtained. The newly calculated flux values are then used to determine the boundary integral for the lower order element.

Here, the mathematical background of the projection procedure is given. For now we discard the subscript e since all operations in the following analysis take place within the element. The solution within an element, as given in equation (3.11), is expressed using a matrix formulation.

$$\mathbf{u}^\delta = B\bar{\mathbf{u}} \quad (3.41)$$

where B is the matrix of the modified expansion basis and the vector $\bar{\mathbf{u}}$ represents the corresponding modified coefficients. The solution can also be expressed using an orthogonal basis and the corresponding coefficients $\bar{\mathbf{u}}_o$ as

$$\mathbf{u}^\delta = B_o\bar{\mathbf{u}}_o \quad (3.42)$$

and the array of orthogonal coefficients is calculated from

$$\bar{\mathbf{u}}_o = B_o^{-1}B\bar{\mathbf{u}} \quad (3.43)$$

Once we have calculated the solution in terms of the orthogonal coefficients, we can set the coefficients corresponding to the high-order frequencies equal to zero and determine $\bar{\mathbf{u}}_{o,f}$. The filtered coefficients that belong to the modified basis are obtained using

$$\bar{\mathbf{u}}_f = B^{-1} B_o \bar{\mathbf{u}}_{o,f} \quad (3.44)$$

where $\bar{\mathbf{u}}_f$ represents the array of filtered coefficients corresponding to the modified basis functions. In this way, the information contained in the high frequency components is removed without altering the mean value and the boundary integral for the lower order element. This capability to use a variable polynomial order distribution for a discontinuous Galerkin discretisation is implemented in parallel for two- and three-dimensional flow problems and will be demonstrated in chapter 4.

3.5 Discretisation of the adjoint equations

Using the argument of consistency, we discretise the adjoint equations using the same high-order discontinuous Galerkin scheme as the flow equations.

3.5.1 Adjoint Euler equations

In section 2.2.1, we have derived the continuous adjoint equations based on the compressible Euler equations. The adjoint equations are given in equation (2.45) which we discretise using a high-order spectral discontinuous Galerkin approach. Hence, using the same approach for obtaining equation (3.15), we write the following weak form of the adjoint equations corresponding to the compressible Euler equations.

$$\begin{aligned} & \sum_{e=1}^{N_{el}} \int_{\Omega_e} \sum_{i=1}^2 \frac{\partial \phi_e}{\partial x_i} \left\{ \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t \psi_e^\delta \right\} + \phi_e \left\{ \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} \right]^t \psi_e^\delta \right\} d\Omega \\ & + \sum_{e=1}^{N_{el}} \int_{\Gamma_e} \phi_e \left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right]^t \psi_e^\delta \right\} d\Gamma = 0 \end{aligned} \quad (3.45)$$

where

$$\left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right]^t \psi_e^\delta = \left[\sum_{i=1}^2 \frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} n_i \right]^t \psi_e^\delta \quad (3.46)$$

From integrating by parts, we obtain again a volume and a boundary integral. For the adjoint equations, we use the discontinuous Galerkin approach as well and therefore we need to replace the boundary integral with a numerical flux function. In this work, we modify Roe's approximate Riemann solver [89] in order to obtain

an equivalent numerical flux function for the adjoint equations.

Adjoint Roe-type Riemann flux

Using the same analogy as for the governing equation, we define a numerical flux function to replace the $\left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}}\right]^t \boldsymbol{\psi}_e^\delta$ term on the boundary of the elements. We also rotate the two- or three-dimensional problem in normal direction with respect to the interface and solve a similar one-dimensional Riemann-type problem in normal direction as discussed in section 3.2.1 to evaluate the boundary fluxes. This flux function is based on the Roe-type Riemann solver [89] which was used to calculate the boundary fluxes for the governing equations. Hence an equivalent adjoint Roe Riemann solver is derived.

$$\hat{\mathcal{H}}(\boldsymbol{\psi}_{ex}, \boldsymbol{\psi}_{in}) = -\frac{1}{2} \left(\left\{ \left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta \right\}_{ex} - \left\{ \left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta \right\}_{in} \right) - \frac{1}{2} |\boldsymbol{\Lambda}| \mathbf{T}^{-1} \quad (3.47)$$

For the derivation of expression (3.47) and of $\boldsymbol{\Lambda}$ and $\mathbf{T}$, the reader is referred to Appendix B.

3.5.2 Adjoint Navier-Stokes equations

The adjoint equations corresponding to the compressible Navier-Stokes equations are given in equation (2.60). Again we apply the approach described by Bassi and Rebay [8] by introducing the auxiliary term $\mathbf{g} = \nabla \boldsymbol{\psi} = \left\{ \frac{\partial \boldsymbol{\psi}}{\partial x_1}, \frac{\partial \boldsymbol{\psi}}{\partial x_2} \right\}$ that represents in this case the first order derivatives of the adjoint variables. We compose a similar coupled system, as shown in equation (3.17), to accommodate for the second order derivative terms.

$$\begin{aligned} \mathbf{g} - \nabla \boldsymbol{\psi} &= \mathbf{0} \\ -\sum_{i=1}^2 \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}} \right]^t \frac{\partial \boldsymbol{\psi}}{\partial x_i} - \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \frac{\partial \mathbf{g}_i}{\partial x_i} &= \mathbf{0} \end{aligned} \quad (3.48)$$

Hence, similar to equation (3.18), we obtain the discrete form to solve for the first order derivatives of the adjoint variables.

$$\sum_{e=1}^{N_{el}} \left\{ \int_{\Omega_e} \phi_e \mathbf{g}_e^\delta d\Omega - \int_{\Gamma_e} \phi_e \boldsymbol{\psi}_e^\delta \vec{n} d\Gamma + \int_{\Omega_e} \nabla \phi_e \boldsymbol{\psi}_e^\delta d\Omega \right\} = \mathbf{0} \quad (3.49)$$

Once a solution is found for $\mathbf{g}_e^\delta$, we then substitute this in the discretised form of the second equation of the coupled system. The discretised second equation of the

coupled system is given by

$$\begin{aligned}
& \sum_{e=1}^{N_{el}} \int_{\Omega_e} \sum_{i=1}^2 \frac{\partial \phi^e}{\partial x_i} \left\{ \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta \right\} + \sum_{i=1}^2 \phi_e \left\{ \frac{\partial}{\partial x_i} \left[\frac{\partial \mathbf{f}_i^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta \right\} d\Omega \\
& - \sum_{e=1}^{N_{el}} \int_{\Gamma_e} \phi^e \left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} - \frac{\partial \mathbf{f}_n^v}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta \right\} d\Gamma \\
& + \sum_{e=1}^{N_{el}} \int_{\Omega_e} \sum_{i,j=1}^2 \frac{\partial \phi^e}{\partial x_i} \left\{ \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \mathbf{g}_{i,e}^\delta \right\} + \sum_{i,j=1}^2 \phi_e \left\{ \frac{\partial}{\partial x_i} \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \mathbf{g}_{i,e}^\delta \right\} d\Omega \\
& - \sum_{e=1}^{N_{el}} \int_{\Gamma_e} \phi^e \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \mathbf{g}_e^\delta n_i d\Gamma = \mathbf{0}
\end{aligned} \tag{3.50}$$

For the adjoint boundary flux related to the convective terms, $\left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta$, we use the same derived adjoint Roe-type Riemann solver that was introduced in section 3.5.1. For the flux terms that follow from integrating the viscous terms by parts, we use again the LDG method in order to be consistent with the discretisation of the governing equations. Hence, the term $\boldsymbol{\psi}_e^\delta n_i$ is replaced by the external state and is therefore written as

$$\boldsymbol{\psi}_e^\delta \vec{n} \Big|_{\Gamma_e} = \boldsymbol{\psi}_{ex}^\delta \vec{n} \Big|_{\Gamma_e} \tag{3.51}$$

The LDG approach prescribes that the internal state is taken for the first order derivatives and therefore

$$\sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \mathbf{g}_{i,e}^\delta n_i \Big|_{\Gamma_e} = \sum_{i,j=1}^2 \left[\frac{\partial \mathbf{f}_j^v}{\partial \mathbf{u}_{x_i}} \right]^t \mathbf{g}_{i,in}^\delta n_i \Big|_{\Gamma_e} \tag{3.52}$$

3.5.3 Adjoint boundary conditions

In this section, the discretisation of the derived adjoint boundary conditions, given in sections 2.2.2 and 2.2.4, is given.

Adjoint boundary conditions for the compressible Euler equations

The continuous formulation of the adjoint boundary conditions for the compressible Euler equations is given in expression (2.50). Note that the boundary conditions are determined by the aerodynamic force of interest. Hence, the force projection vector, denoted by $\vec{\theta}$, determines the values of the adjoint variables at the wall. Considering equation (2.50), we need to ensure that the tangential component of $\vec{\psi} = \{\psi_2, \psi_3\}^t$ is unchanged while its wall normal component is equal to $(\vec{\theta} \cdot \vec{n})\vec{n}$. Hence, we need to set the appropriate conditions on the ghost interface ($\boldsymbol{\psi}_{ex}$) in order to impose this boundary condition. Using the same analogy applied

to derive (3.22), we write the weak form of the boundary condition in equation (2.50) as

$$\vec{\psi}_{in} = \vec{\psi}_{ex} - 2(\vec{\psi} \cdot \vec{n} - \vec{\theta} \cdot \vec{n})\vec{n} \quad (3.53)$$

Hence, at the ghost interfaces for the adjoint problem related to inviscid flows, we impose the boundary values

$$\psi_{ex}^\delta = \left\{ \begin{array}{c} \psi_1 \\ \psi_2 - 2(\vec{\psi} \cdot \vec{n} - \vec{\theta} \cdot \vec{n})n_1 \\ \psi_3 - 2(\vec{\psi} \cdot \vec{n} - \vec{\theta} \cdot \vec{n})n_2 \\ \psi_4 \end{array} \right\}_{in} \quad (3.54)$$

Adjoint boundary conditions for the compressible Navier-Stokes equations

The adjoint boundary conditions for the compressible Navier-Stokes equations are given in equation (2.63). In this thesis, we consider isothermal walls and therefore the external state becomes

$$\psi_{ex}^\delta = \left\{ \begin{array}{c} \psi_{1,in} \\ -\psi_{2,in} + \theta_1 \\ -\psi_{3,in} + \theta_2 \\ 0 \end{array} \right\} \quad (3.55)$$

Adjoint boundary conditions at the far-field

According to expression (2.52), the values of the adjoint variables at the far-field boundary should vanish. Therefore, we impose a zero condition on the ghost interface when we are dealing with a far-field boundary. Hence, we write

$$\psi_{ex}^\delta = 0 \quad (3.56)$$

3.6 Verification of the adjoint solver

To verify the adjoint solver, we have chosen to consider the adjoint field as a flow field which we compare with reference data. First we investigate a quasi one-dimensional flow for which the adjoint solution can be calculated analytically [35]. Next to that, adjoint solutions for a two-dimensional flow case is compared with reference data that is obtained using the DLR/TAU code. The DLR/TAU code uses a second-order finite volume discretisation and is used by AIRBUS.

3.6.1 Quasi one-dimensional subsonic inviscid flow through a converging-diverging nozzle ($Ma = 0.12$).

To verify the implementation of the adjoint solver in Nektar++, we examine the quasi one-dimensional test case of subsonic inviscid flow through a converging-diverging nozzle. This quasi one-dimensional test case is chosen since it retains complex flow features like non-linearities and shocks but it is simple enough to provide exact solutions for both the flow and adjoint equations. In the work of Giles and Pierce [35], exact adjoint solutions are presented for quasi one-dimensional inviscid compressible flow problems which serve as a useful set of reference solutions for researchers that develop adjoint codes. The quasi one-dimensional flow equations are given for the computational domain $\Omega = [-1, 1]$ by

$$\mathcal{R}(\mathbf{u}) = \frac{\partial \mathbf{f}_1^c}{\partial x_1} - \frac{dh}{dx_1} \mathbf{p} = 0; \quad \mathbf{u} \in \Omega \quad (3.57)$$

where $\mathbf{u} = \{\rho, \rho u, \rho E\}^t$ is the one-dimensional state vector of conservative variables, $\mathbf{f}_1^c = \{\rho u, p + \rho u^2, \rho u H\}^t$ is the one-dimensional convective flux vector, $\mathbf{p}$ is defined as $\mathbf{p} = \{0, p, 0\}^t$ and h represents the cross sectional distribution of the converging-diverging nozzle. The value of h is dependent on x_1 and is given by

$$h(x_1) = \begin{cases} 2 & \text{if } -1 \leq x_1 \leq -0.5 \\ 1 + \sin \pi x_1^2 & \text{if } -0.5 < x_1 < 0.5 \\ 2 & \text{if } 0.5 \leq x_1 \leq 1 \end{cases} \quad (3.58)$$

At the inlet we prescribe a total enthalpy $H_{in} = 4.0$ and a total pressure $p_{t,in} = 2.0$. The pressure at the outlet is prescribed by $p_{out} = 1.98$, and velocity and density are extrapolated. The numerical solution for the Mach number calculated using Nektar++ and the analytical solution from the area-Mach relation [2] are compared in figure 3.11.

Figure 3.11 shows a good agreement between the numerical and analytical flow solution. In order to derive the continuous adjoint equations, we need to specify a target functional. In the work of Giles and Pierce [35], the integral of the pressure along the channel was chosen to mimic integral target quantities like the lift or drag coefficient. The target functional is written as

$$\mathcal{J}(\mathbf{u}) = \int_{-1}^1 p \, dx_1 \quad (3.59)$$

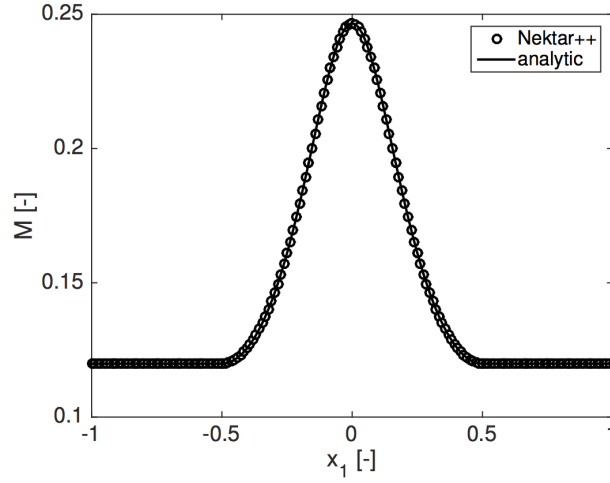


Figure 3.11: The solution for the Mach number obtained using Nektar++ with $N_{el} = 24$ and $P = 8$.

Using this definition of the target functional and the quasi one-dimensional Euler equations presented in equation (3.57) as a constraint, we write the Lagrangian as

$$\mathcal{L}(\mathbf{u}, \psi) = \int_{-1}^1 p - \psi^t \left\{ h \frac{\partial \mathbf{f}_1^c}{\partial x_1} - \frac{dh}{dx_1} \mathbf{p} \right\} dx_1 \quad (3.60)$$

Again we are seeking an extremum of this functional by setting its variation equal to zero.

$$\delta \mathcal{L} = \int_{-1}^1 \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\}^t \delta \mathbf{u} - \psi^t \left\{ h \frac{\partial}{\partial x_1} \left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} \right] \delta \mathbf{u} - \frac{dh}{dx_1} \left[\frac{\partial \mathbf{p}}{\partial \mathbf{u}} \right] \delta \mathbf{u} \right\} dx_1 = 0 \quad (3.61)$$

where for a quasi one-dimensional case $\left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} \right]$ and $\left[\frac{\partial \mathbf{p}}{\partial \mathbf{u}} \right]$ are 3×3 matrices. Evaluating the second integral term using integration by parts, we obtain

$$\delta \mathcal{L} = - \int_{-1}^1 \delta \mathbf{u}^t \left\{ -h \left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} \right]^t \frac{\partial \psi}{\partial x_1} - \frac{dh}{dx_1} \left[\frac{\partial \mathbf{p}}{\partial \mathbf{u}} \right]^t \psi - \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} \right\} dx_1 \quad (3.62)$$

$$- h \delta \mathbf{u}^t \left[\frac{\partial \mathbf{f}_1^c}{\partial x_1} \right] \psi \Big|_{-1}^1 = 0. \quad (3.63)$$

Using the argument that the variation $\delta \mathbf{u}$ is arbitrary, we state that

$$- h \left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} \right]^t \frac{d\psi}{dx_1} - \frac{dh}{dx_1} \left[\frac{\partial \mathbf{p}}{\partial \mathbf{u}} \right]^t \psi - \left\{ \frac{\partial p}{\partial \mathbf{u}} \right\} = 0; \quad \psi \in \Omega \quad (3.64)$$

where the inlet boundary condition is given by

$$h\delta\mathbf{u}^t \left[\frac{\partial \mathbf{f}_1^c}{\partial x_1} \right] \boldsymbol{\psi} \Big|_{x_1=-1} = 0 \quad (3.65)$$

and the outlet boundary condition is given by

$$h\delta\mathbf{u}^t \left[\frac{\partial \mathbf{f}_1^c}{\partial x_1} \right] \boldsymbol{\psi} \Big|_{x_1=1} = 0 \quad (3.66)$$

Hence, the adjoint problem is now defined by expressions (3.64)-(3.66). In the work of Giles and Pierce [35], the analytical solution to this problem is determined using a Green's function approach. This approach is adopted here as well to calculate an analytical solution against which we can compare the numerically obtained solution that we calculate using Nektar++. The same DG approach is used as discussed in the previous sections. The flow equations presented in expression (3.57) are written in terms of the conservative variables $\mathbf{u}$.

The same boundary conditions are adopted as the ones presented by Giles and Pierce [35]. For subsonic inflow, they fixed the total enthalpy and total pressure while at the outflow they fix the static pressure.

Lozano and Ponsin [62] propose a change of variables at each boundary and perform a characteristic analysis to derive the appropriate boundary conditions for the continuous adjoint problem. Expressions for $\boldsymbol{\psi}_{x_1=-1}$ and $\boldsymbol{\psi}_{x_1=1}$ are derived by expanding equation (3.65) in terms of H , p_t and Ma and equation (3.66) in terms of ρ , u and p and setting the appropriate variations in the newly defined state variables equal to zero. For a more detailed description of the derivation of the correct adjoint boundary conditions, the reader is referred to [62]. For completeness the derivation is also given in Appendix C.

In figure 3.12, the solution that is calculated using the adjoint solver that is implemented in Nektar++ is compared with the analytical solution that is determined using the Green's function approach proposed by Giles and Pierce [35]. The numerical solution of the adjoint problem corresponds well with the analytical solution.

Duivesteijn et al. [23] used the discrete adjoint approach for this particular test case and investigated the effect on the adjoint solution of imposing boundary conditions either in a strong or a weak form. Duivesteijn et al. [23] concluded that, when an adjoint variable is not defined by boundary conditions, the numerical adjoint solution near that boundary strongly depends on the flux function definition used in the flow solver. This issue has been thoroughly addressed for a DG discretisation by Hartmann [43] for example. Lozano and Ponsin [62] compared the continuous and discrete adjoint approach for subsonic inviscid quasi one-dimensional flow. They

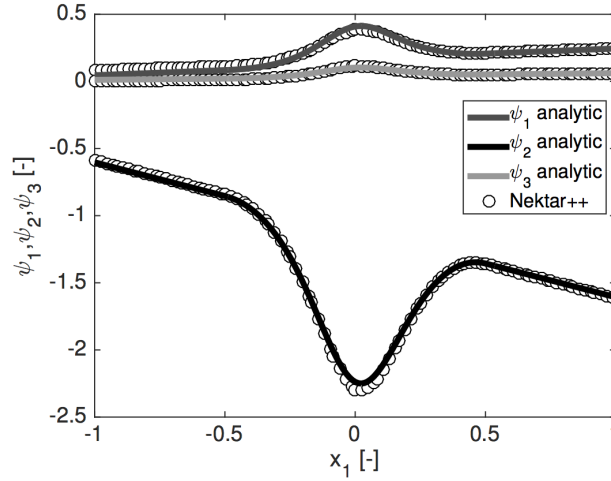


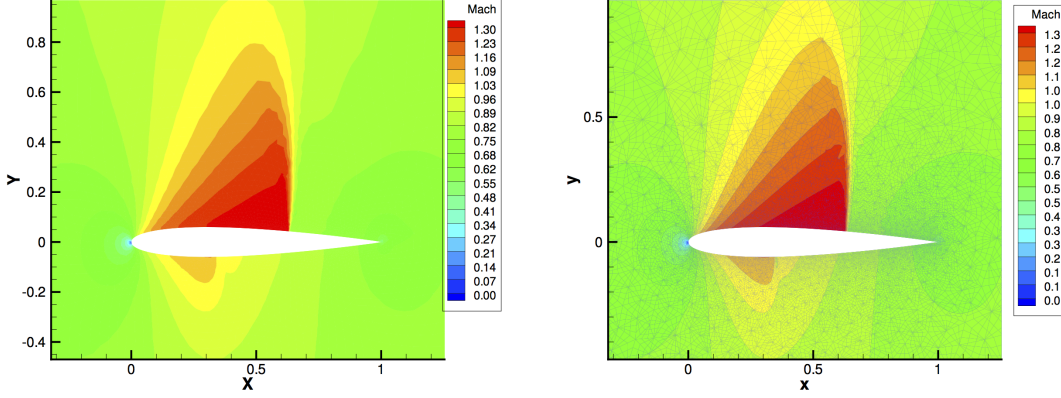
Figure 3.12: Comparison between the numerical and exact adjoint solution for quasi one-dimensional subsonic inviscid flow ($Ma = 0.12$). The Nektar++ solution was obtained using $N_{el} = 24$ elements and a uniform $P = 8$.

show that there is a shift in adjoint solution with respect to the analytical solution when the discrete adjoint approach is used. This shift is reduced by increasing the resolution and it is shown that the solution converges to the analytical solution. However, in the continuous adjoint approach, the correct adjoint solution is obtained within numerical accuracy similar to the flow solution and the adjoint solutions shown in figure 3.12. The conclusions of Duijvesteijn et al. [23] and Lozano and Ponsin [62] provide evidence that the continuous adjoint approach gives the best result since an adjoint consistent discretisation is guaranteed. The results presented in figure 3.12 and the continuous adjoint solutions presented by Lozano and Ponsin [62] confirm this statement. However, Lozano and Posin [62] show that the discrete adjoint approach can result in local oscillations due to inconsistent flux treatment. Similar effects have been presented for two-dimensional flow cases by Hartmann [43]. Furthermore, this numerical test helps detecting inaccuracies in the solution or coding flaws in the adjoint solver.

3.6.2 Two-dimensional transonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$)

In this section, we consider the test case of two-dimensional transonic inviscid flow ($Ma = 0.8$) past a NACA 0012 aerofoil at an angle of attack of 1.25° and we are interested in determining the adjoint solution for the lift coefficient. The flow has a strong shock on the suction side and a weak shock on the pressure side. The flow and adjoint solutions obtained using Nektar++ are compared with

reference data that is calculated using the DLR/TAU code. The flow solutions that are calculated using both the DLR/TAU code and Nektar++ are shown in figure 3.13(a) and 3.13(b) respectively.



(a) Contour lines for Ma obtained using the DLR/TAU code (2^{nd} order finite volume). (b) Contour lines for Ma obtained using the Nektar++ code with $P = 3$.

Figure 3.13: Comparison between the DLR/TAU code and Nektar++ of two-dimensional transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).

The steady solution shown in figure (3.13) serves as a base flow around which we linearise. To compare the adjoint solutions, we need to linearise around the same base flow and we need to determine the same target functional, in this case c_l . The DLR/TAU code calculates the adjoint solution based on a normalised base flow where the state variables are defined as follows

$$\bar{\rho} = \frac{\rho}{\rho_\infty}, \quad \bar{u} = M \sqrt{\frac{\gamma p_\infty}{\rho_\infty}} \cos \alpha \sqrt{\frac{\rho_\infty}{p_\infty}}, \quad \bar{v} = M \sqrt{\frac{\gamma p_\infty}{\rho_\infty}} \sin \alpha \sqrt{\frac{\rho_\infty}{p_\infty}}, \quad \bar{p} = \frac{p}{p_\infty}$$

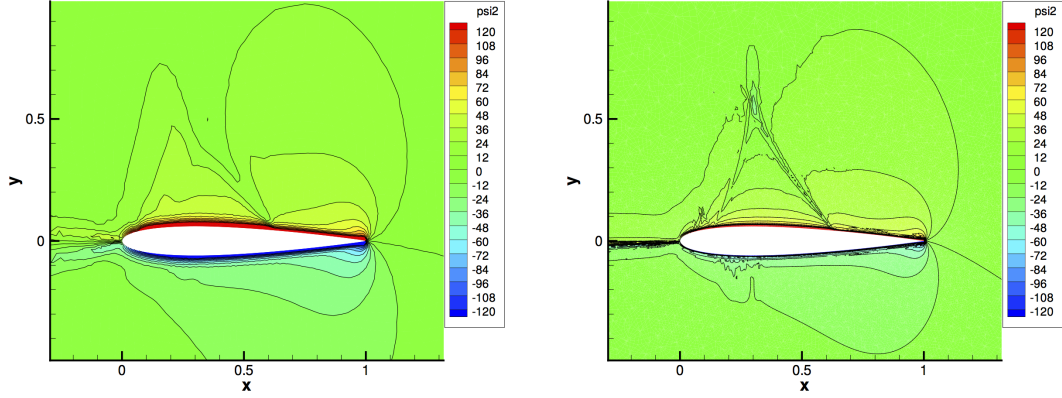
As a result, the same normalisation procedure is applied for the base flow obtained using Nektar++. Furthermore, the DLR/TAU code determines the adjoint solution based on the non-dimensional lift force coefficient which is defined as

$$c_l = \frac{2L}{S_{ref} \gamma M^2} \quad (3.67)$$

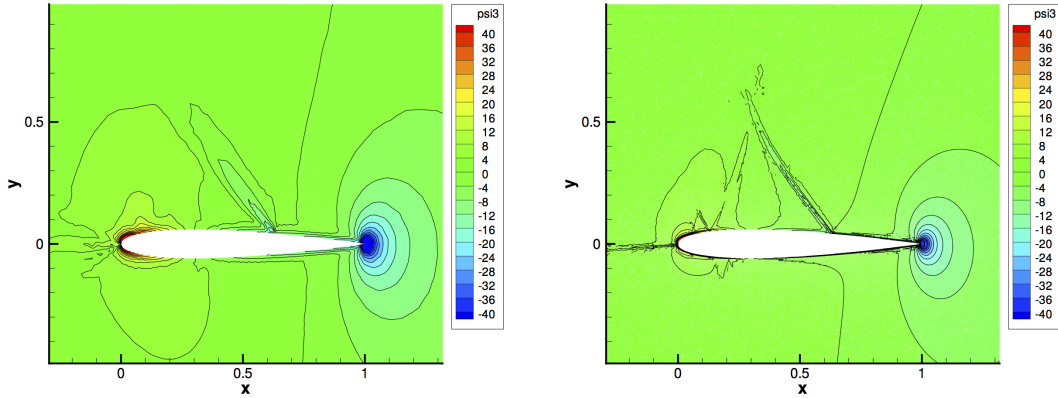
Substituting $\alpha = 1.25 \text{ deg}$, $S_{ref} = 1.0$, $\gamma = 1.4$ and $Ma = 0.8$ and the force projection vector for lift, $\vec{\theta} = \{-\sin \alpha, \cos \alpha\}^t$, into the boundary condition expression that is given in equation (2.50) gives

$$n_1 \psi_2 + n_2 \psi_3 = \frac{2}{S_{ref} \gamma M^2} (-n_1 \sin \alpha + n_2 \cos \alpha) \quad (3.68)$$

In figure 3.13 we see that the propagation of information varies as the eigenvalues vary within the domain and this is a typical feature which is highlighted by the adjoint solution. The results for the adjoint flow variables ψ_2 and ψ_3 obtained using DLR/TAU and Nektar++ are given in figure 3.14.



(a) Contour lines for ψ_2 obtained using the DLR/TAU code. (b) Contour lines for ψ_2 obtained using the Nektar++ code with $P = 3$.



(c) Contour lines for ψ_3 obtained using the DLR/TAU code. (d) Contour lines for ψ_3 obtained using the Nektar++ code with $P = 3$.

Figure 3.14: Comparison between the DLR/TAU code (47×10^3 degrees of freedom) and Nektar++ (90×10^3 degrees of freedom) of the two-dimensional adjoint solution for transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).

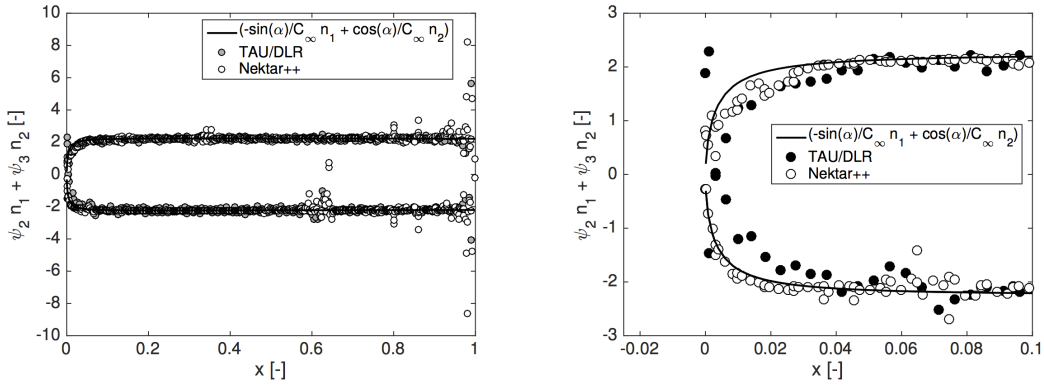
We interpret the adjoint solution presented in figure 3.14 as follows: A positive change in one of the momentum components results in an increase in lift where the corresponding adjoint component is positive and it decreases the lift where the corresponding adjoint component is negative. Hence, a local source of x -momentum on the suction side of the NACA 0012 results in an increase in lift. However, when

a similar source of x -momentum is added on the lower side of the airfoil, the lift will decrease.

By comparing the adjoint fields obtained using DLR/TAU and Nektar++ we see that similar features are captured. For example, the characteristic line that leads to the origin of the shock on the suction side of the airfoil is visible both in the ψ_2 and ψ_3 solutions. Upstream of the shock at the suction side, the value of ψ_2 is higher than downstream of the shock indicating that the lift is more sensitive to the flow upstream of the shock. At the leading edge of the NACA 0012, we see that a source of y -momentum results in an increase in lift.

The plots given in figure 3.14 also generally show that the solution obtained using the DLR/TAU code is significantly more diffusive, than the solution obtained using Nektar++. This is due to the fact that the DLR/TAU code uses a second order finite volume scheme. The shape of the solution looks very similar.

Equation (3.68) dictates the values of ψ_2 and ψ_3 at the boundary. As a results, a comparison is made between Nektar++ and the DLR/TAU code to show that the boundary condition at the wall is sufficiently satisfied. Next to the numerical solutions, we plot the theoretical value given in equation (3.68). The results are shown in figure 3.15.



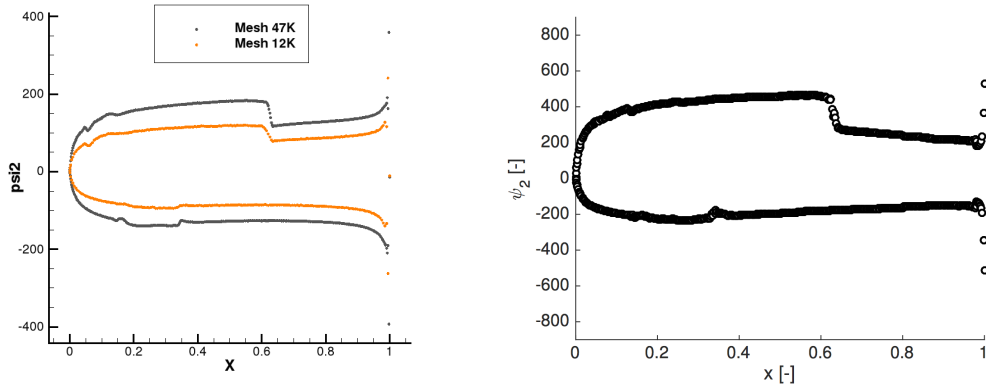
(a) Comparison between DLR/TAU and Nektar++ of the discrete boundary condition values at the wall. (b) Detail of the comparison at the leading edge between DLR/TAU and Nektar++ of the discrete boundary condition values at the wall.

Figure 3.15: Comparison between the DLR/TAU code, Nektar++ and the analytical condition of the boundary condition values. The comparison data was provided by Dr. C. Lozano from the National Institute for Aerospace Technology (INTA) in Madrid.

Both Nektar++ and the DLR/TAU code provide a fairly good correspondence with the theoretical value of the boundary conditions. The results obtained using Nektar++ show two peaks, one at approximately 35% of the chord and one at

approximately 60% of the chord which correspond to the location of the weak shock on the lower side and the strong shock on the upper side respectively. However, the DLR/TAU code shows only a peak at the 60% chord location. Hence, there is no effect visible related to the weak shock on the pressure side which can be ascribed to the more diffusive nature of the discretisation scheme used by the DLR/TAU code.

Both numerical solutions show high fluctuations at the trailing edge ($x = 1$). A detailed comparison of the discrete boundary values at the leading edge of the wing is plotted in figure 3.15(b). This figure illustrates that the solution obtained using Nektar++ corresponds better with the analytical solution compared to the solution obtained using the DLR/TAU code. However, it is noted that we use more degrees of freedom to approximate the adjoint solution compared to the DLR/TAU solution.



(a) The ψ_2 solution at the wall of the NACA 0012 calculated using the DLR/TAU code. (b) The ψ_2 solution at the wall of the NACA 0012 calculated using Nektar++.

Figure 3.16: Comparison between the DLR/TAU code and Nektar++ of the ψ_2 wall profile. The comparison data was provided by Dr. C. Lozano from the National Institute for Aerospace Technology (INTA) in Madrid.

Figure 3.16(a) depicts two ψ_2 adjoint solutions at different resolutions which were both obtained using the DLR/TAU code. The coarse mesh has 12×10^3 nodes while the fine mesh has 47×10^3 nodes. As a comparison, a similar plot is shown for the Nektar++ solution in figure 3.16(b). The solution obtained using the finer mesh differs by a fixed factor from the coarse solution. Similar issues have been noted by others [33, 97] and this has been ascribed to the geometrical singularity induced by the cusp trailing edge.

Chapter 4

p -adaptation for compressible flows

Adaptive methods in the field of computation fluid dynamics aim to approximate flow features as accurate as possible while maintaining a low computational cost. By identifying the local accuracy within the computational domain, we determine whether an improvement is required either by decreasing the local element size (h -adaptation) or by increasing the approximation order P within the elements (p -adaptation). In this chapter, we will focus on the latter one.

We consider the high-order spectral/ hp element discretisation that uses a discontinuous Galerkin (DG) approach which was discussed in chapter 3. High-order spectral/ hp methods combine the accuracy and rapid rate of convergence with the flexibility of low order finite element methods. When using the DG discretisation scheme, the solution is allowed to be discontinuous between the elements which brings as an advantage that the code becomes relatively easy to parallelise. Furthermore, it becomes significantly easier to implement the capability of using a variable polynomial order distribution within the computational domain as shown in section 3.4.

We would like to optimally use this attractive property of spectral/ hp element methods by highlighting the parts of the computational domain that require a more accurate solution using an error indicator. Based on the error indicator, we increase the polynomial order locally while preserving a low polynomial order in parts of the computational domain that are sufficiently resolved. This technique is also referred to as p -adaptation and is the topic of this chapter.

As mentioned before in the introduction chapter of this thesis, there are two different types of error indicators: the *a priori* and the *a posteriori* error indicator. In this work we present a local and a global *a posteriori* error indicator. Local error indicators quantify the truncation error based on some form of local interpolation

or extrapolation. Global error indicators on the other hand rely mostly on the solution of an adjoint problem which highlights the domain of dependence of a pre-defined target quantity for which the error needs to be reduced. The adjoint solution determines the importance of the local truncation error with respect to the error in a target functional and this adjoint solution is used to weigh the local truncation error to define an adjoint-based error indicator. In this chapter, we perform p -adaptation using both a local error indicator based on interpolation (see section 4.1) and an adjoint-based error indicator (see section 4.2).

4.1 p -adaptation using the discontinuity sensor

In the previous chapter, we discussed briefly the capability of varying the polynomial order across the mesh in the framework of a DG discretisation. In this chapter, we would like to apply this to solve for the compressible Euler and Navier-Stokes equations. Therefore, we need an error indicator that quantifies how well the governing equations are resolved locally.

In the thesis of Biotto [12], the discontinuity sensor that was first introduced by Persson and Perraire [76] was proposed as an error indicator for two-dimensional flow problems. In this section, this idea is extended to three dimensions. The discontinuity sensor determines the local smoothness of the solution by comparing the solution within an element using two different polynomial orders. The sensor value is calculated by taking the L_2 norm of the difference between the two approximate solutions as shown in equation (3.37). A converged solution is calculated and we define a set of threshold values based on the determined sensor value. An example of a typical sensor distribution was presented in figure 3.8(b). Based on a set of empirically defined sensor threshold values, we identify which elements required an increase in P . For a range of $P_e \leq P \leq P_e + 3$ for example, where P_e represents the initial polynomial order which is typically low ($P_e = 2$), we need three different threshold values namely, s_1 , s_2 and s_3 .

$$P_e^{\text{new}} = \begin{cases} P_e + 3 & \text{if } s_e > s_1 \\ P_e + 2 & \text{if } s_2 < s_e \leq s_1 \\ P_e + 1 & \text{if } s_3 < s_e \leq s_2 \\ P_e & \text{if } s_e \leq s_3 \end{cases} \quad (4.1)$$

The number of required sensor values is dependent on the range of polynomial orders that one wants to use. These threshold values are determined by looking at the steady state sensor distribution. Based on these threshold values, a new polynomial distribution is obtained and this procedure is performed iteratively.

The expression given in equation 4.1 is just an example. The sensor value is typically high when a shock is detected. Hence, it is preferred to keep P relatively low near the shock wave to prevent numerical oscillations. It is straightforward to incorporate the capability to keep the polynomial order low above a certain value. This local error indicator is used to verify that the capability to vary P across the mesh is implemented correctly for three-dimensional flows. We consider a three-dimensional test case of inviscid flow ($Ma = 0.5$) past a sphere. This test case also serves as a motivation for using an adjoint-based error indicator which is explained in the next section.

4.1.1 Three-dimensional subsonic inviscid flow past a sphere ($Ma = 0.5$).

To illustrate the sensor-based p -adaptation procedure, we consider the case of compressible inviscid flow ($Ma = 0.5$) past a sphere. A mesh is generated of a sphere inside a bigger sphere. For the inner sphere diameter holds that $d = 0.5$ and for the outer sphere holds that $D = 40$. In this way we impose far-field boundary conditions on the outer sphere and wall boundary conditions on the inner sphere. The mesh is structured and consists of $N_{el} = 864$ hexahedral elements. Initially, we obtain a low-order ($P = 2$) steady state solution. The mesh with $P = 2$ near the wall of the sphere is given in figure 4.1.

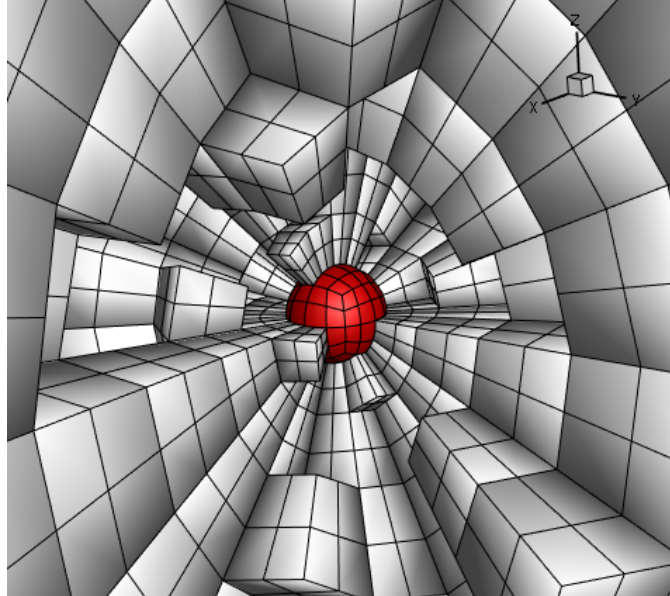


Figure 4.1: Detail of the hexahedral mesh with $P = 2$ surrounding a sphere.

Figure 4.1 depicts the coarseness of the mesh near the wall particularly when considering $P = 2$. For this case we have that $(x_1, x_2, x_3) = (x, y, z)$ and the flow is

moving in positive x -direction.

We know, that for compressible subsonic inviscid flow, the entropy in the computational domain should be constant. Hence, we are sure that when entropy is observed, this is ascribed to the numerical discretisation of the continuous equations and particularly due to the discrete geometry approximation. The entropy generation in this case is an indicator of the overall solution accuracy. Therefore, we use the following definition as a measure of accuracy

$$\epsilon_{L_2} = \sqrt{\frac{\int_{\Omega} \frac{p/\rho^\gamma - p_\infty/\rho_\infty^\gamma}{p_\infty/\rho_\infty^\gamma} d\Omega}{\int_{\Omega} d\Omega}} \quad (4.2)$$

The accuracy of this solution is improved by means of the previously discussed sensor-based p -adaptation strategy. In figure 4.2, a low-order solution ($P = 2$) is compared with a high-order solution ($P = 6$). The solution is supposed to be symmetric like the picture shown for the $P = 6$ result. The solution in the wake of the sphere looks very similar to the solution upstream of the sphere. However for the $P = 2$ solution, shown in figure 4.2(a), we see that the solution in the wake of the sphere is significantly different from the solution upstream of the sphere. This is caused by the numerical entropy that is generated due to the discretisation as shown in figure 4.2(c). The error is convected downstream causing this entropy wake that is observed in figure 4.2(c). Based on the calculated sensor distribution, we determine a set of threshold values and we obtain a new polynomial distribution. An example of the sensor-based polynomial distribution for the case of inviscid flow past a sphere is given in figure 4.3.

We compare the value ϵ_{L_2} and the number of degrees of freedom (N_Q) for each case. The value for N_Q is obtained for a mesh of hexahedra using the following relation

$$N_Q = \sum_{e=1}^{N_{el}} N_{P_e}^{\xi_1} N_{P_e}^{\xi_2} N_{P_e}^{\xi_3} \quad (4.3)$$

where $N_{P_e}^{\xi_1}$, $N_{P_e}^{\xi_2}$ and $N_{P_e}^{\xi_3}$ represent the number of integration points in standard element space in the directions ξ_1 , ξ_2 and ξ_3 respectively. We are using a modified expansion basis and for hexahedral elements holds that $N_{P_e}^{\xi_1} = N_{P_e}^{\xi_2} = N_{P_e}^{\xi_3} = P_e + 2$. The data for ϵ_{L_2} and N_Q is collected in the table 4.1.

Table 4.1 illustrates that a numerical entropy reduces when a variable p -distribution is used. A similar reduction is achieved compared to a uniform $P = 6$ and $P = 7$ solution while only using half of the number of degrees of freedom. This numerical example serves as a motivation to explore new error indicators. It shows that the error is convected downstream. Hence, once the polynomial order is increased at the source of error we see that it is unnecessary to increase the polynomial order

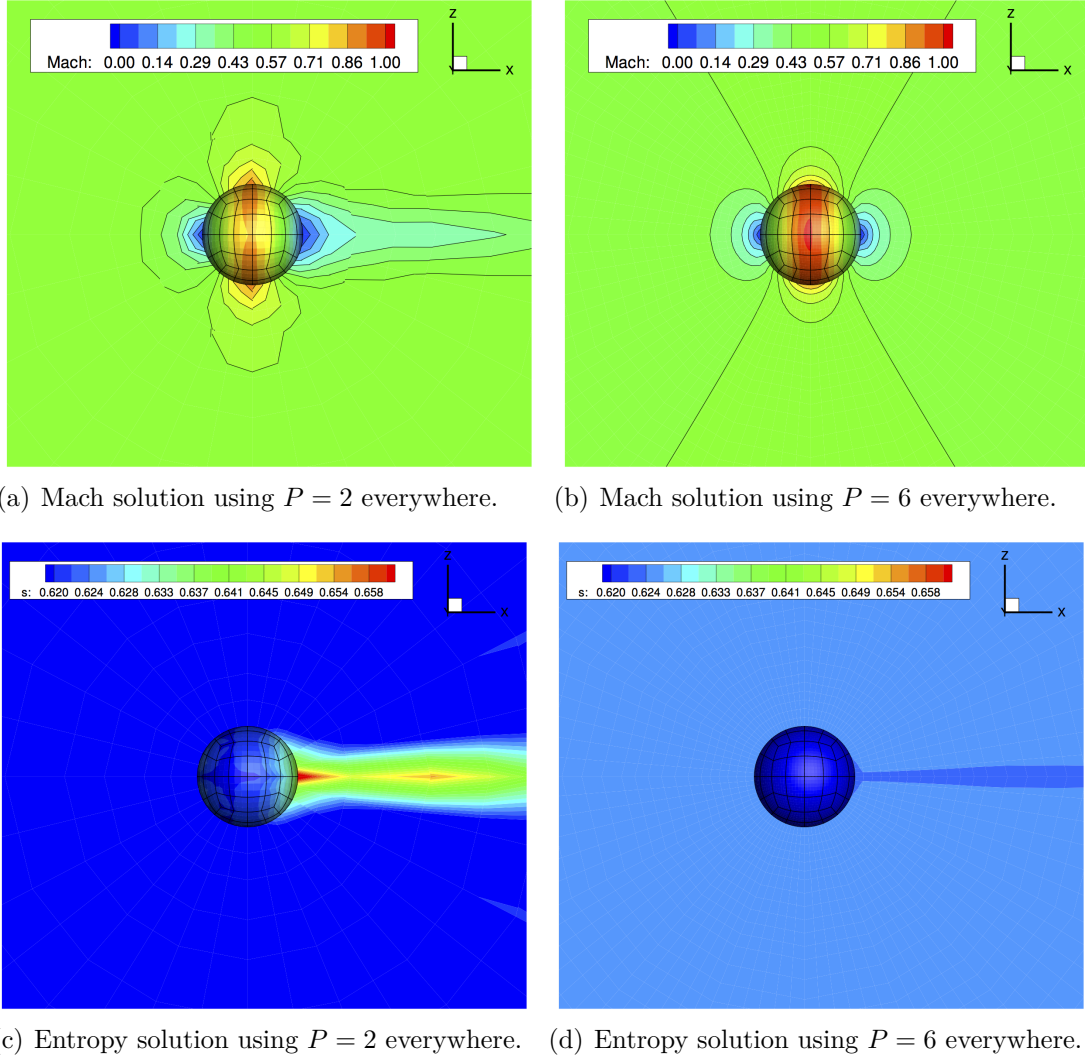
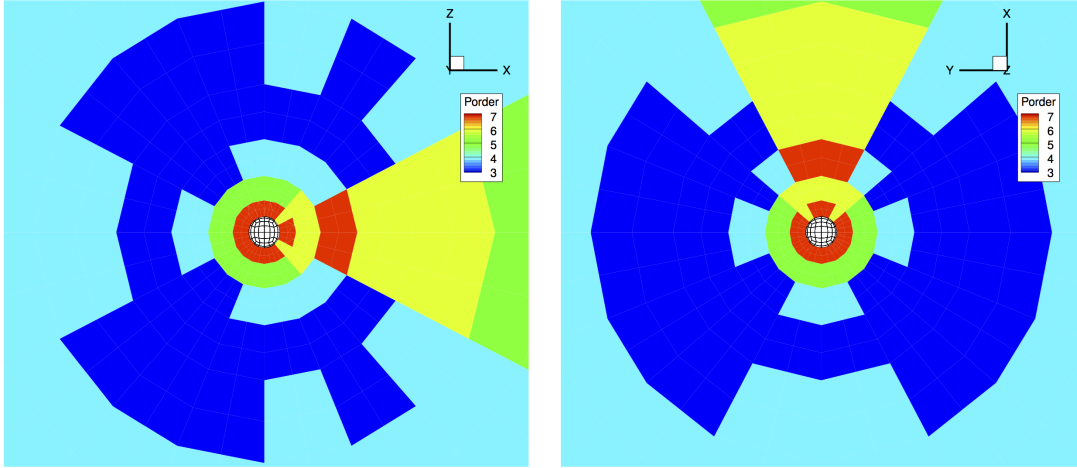


Figure 4.2: Comparison between a $P = 2$ and a $P = 6$ solution of the entropy generation for inviscid flow past a sphere ($Ma = 0.5$).

Uniform P	$P = 2$	$P = 6$	$P = 7$
N_Q	55296	442368	629856
ϵ_{L_2}	2.1×10^{-3}	3.17×10^{-7}	1.10×10^{-7}
Variable P	$2 \leq P \leq 6$	$3 \leq P \leq 7$	
N_Q	160588	255700	
ϵ_{L_2}	1.6×10^{-5}	1.15×10^{-6}	

Table 4.1: Comparison between uniform polynomial refinement with sensor-based p -adaptation of entropy error for inviscid flow past a sphere ($Ma = 0.5$).



(a) The new polynomial order distribution in the (x, z) plane based on the sensor value. (b) The new polynomial order distribution in the (x, y) plane based on the sensor value.

Figure 4.3: An example of the new polynomial order distribution based on the discontinuity sensor for inviscid flow past a sphere ($Ma = 0.5$).

downstream of this region.

Sensor-based p -adaptation is an easy to implement p -adaptation strategy that is computationally cheap. However, this technique only considers the local error and does not incorporate the dependency of the local error to other parts of the computational domain. This results in unnecessary refinement particularly in the wake of a body. Next to that, the elements at the far-field are usually larger causing a less accurate approximation of the local solution. It has been noticed that this is picked up by the sensor causing again unnecessary refinement at the far-field boundary. In the next section, we introduce a p -adaptation strategy that avoids this. In order to improve the accuracy of target functionals like aerodynamic force coefficients, we determine the sensitivity map based on an adjoint solution and define a goal-based error indicator.

4.2 p -adaptation using goal-based error estimation

The adaptation approach adopted here is to increase the polynomial order of the elemental approximation in parts of the computational domain where the presence of numerical errors affecting the value of a quantity of interest, such as lift or drag, is highlighted by a goal-based error indicator. In this section, we present the formulation of the goal-based error indicator.

Goal-based error estimation is a technique that links the error in a pre-defined

target functional to the local solution error in the domain by using the concept of duality. This technique relies on an adjoint solution that provides us with the sensitivity of the state variables with respect to the target functional as explained in section 2.2. In other words, the adjoint solution, tells us how the value of the target functional will change when a small perturbation is added to the solution at an arbitrary point in the domain. In this section, we consider the discretisation error to be a small perturbation. The discrete solution is essentially the exact solution, denoted by $\mathbf{u}$, with an additional variation due to the discretisation, denoted by $\delta\mathbf{u}$. Therefore we write the discrete approximation of $\mathbf{u}$ as $\mathbf{u} + \delta\mathbf{u}$. Hence, the sensitivity information is used to weigh the local discretisation error with respect to the error in the pre-defined target functional. By defining an adjoint-based error indicator, we decide whether we would like to improve the local accuracy by increasing the polynomial order.

For aerodynamic problems, we are typically interested in determining the lift (c_l) or drag (c_d) coefficient as accurate as possible. Depending on whether we consider inviscid or viscous laminar flow, we use (2.39) or (2.53) as our target functional. The aim of this section is to link the error in $\mathcal{J}(\mathbf{u})$ to the solution error and define an element-wise error indicator that drives p -adaptation. During this discussion we assume that we have a discrete flow and adjoint solution.

The error in target functional is determined both by the solution error and by the truncation error that is introduced when approximating the target functional $\mathcal{J}(\mathbf{u})$. As a result, we write the error in target functional, denoted by ϵ_J , as

$$\begin{aligned} \epsilon_J = \{\mathcal{J}_\delta(\mathbf{u} + \delta\mathbf{u}) - \mathcal{J}(\mathbf{u})\} &= \{\mathcal{J}_\delta(\mathbf{u} + \delta\mathbf{u}) - \mathcal{J}_\delta(\mathbf{u}(\mathbf{x}_\delta))\} \\ &+ \{\mathcal{J}_\delta(\mathbf{u}(\mathbf{x}_\delta)) - \mathcal{J}(\mathbf{u})\} \end{aligned} \quad (4.4)$$

where $\mathcal{J}(\mathbf{u})$ represents the exact solution for the target functional. We use the subscript δ to indicate that we are dealing with a discrete operator hence, $\mathcal{J}_\delta(\mathbf{u} + \delta\mathbf{u})$ is the discrete target functional operator evaluated using the discrete solution. The term $\mathcal{J}_\delta(\mathbf{u}(\mathbf{x}_\delta))$ represents the discrete operator of the target functional determined by evaluating the exact solution at the discrete coordinates $(\mathbf{u}(\mathbf{x}_\delta))$. The right hand side of equation (4.4) is written in this particular form to illustrate the two contributions of error in the defined target functional. The first contribution $\{\mathcal{J}_\delta(\mathbf{u} + \delta\mathbf{u}) - \mathcal{J}_\delta(\mathbf{u}(\mathbf{x}_\delta))\}$, is due to the error in the discrete solution ($\delta\mathbf{u}$) and the second contribution $\{\mathcal{J}_\delta(\mathbf{u}(\mathbf{x}_\delta)) - \mathcal{J}(\mathbf{u})\}$ is the truncation error in approximating the target functional [34]. In this work, we are interested in the first contribution to the error and we consider that the variation in $\mathcal{J}$ is written as

$$\delta\mathcal{J} = \mathcal{J}_\delta(\mathbf{u} + \delta\mathbf{u}) - \mathcal{J}_\delta(\mathbf{u}(\mathbf{x}_\delta)) \quad (4.5)$$

Furthermore, the discretisation error leads to a non-zero residual vector, denoted by $\delta\mathcal{R}$, which is a column vector with 4 entries. Mathematically, we write the effect of the discretisation error on the governing equations in a similar way as equation (4.5) and therefore

$$\delta\mathcal{R} = \mathcal{R}_\delta(\mathbf{u} + \delta\mathbf{u}) - \mathcal{R}_\delta(\mathbf{u}(\mathbf{x}_\delta)) \quad (4.6)$$

We approximate the variation $\delta\mathcal{R}$ using the same formulation as given in equation (2.31) and (2.32). The variation in the target functional, $\delta\mathcal{J}$, is approximated using equations (2.29) and (2.30). Rewriting equation (2.31) so that we obtain the fluctuation $\delta\mathbf{u}$ in terms of $\delta\mathcal{R}$ we get

$$\delta\mathbf{u} = \left[\frac{\partial\mathcal{R}_\delta}{\partial\mathbf{u}} \right]^{-1} \delta\mathcal{R} \quad (4.7)$$

We then substitute this expression in equation (2.29) to link the variation in target functional $\delta\mathcal{J}$ to the residual $\delta\mathcal{R}$

$$\delta\mathcal{J} \approx \left\{ \frac{\partial J_\delta}{\partial\mathbf{u}} \right\}^t \left[\frac{\partial\mathcal{R}_\delta}{\partial\mathbf{u}} \right]^{-1} \delta\mathcal{R} \quad (4.8)$$

Here, the term $\left\{ \frac{\partial J_\delta}{\partial\mathbf{u}} \right\}^t \left[\frac{\partial\mathcal{R}_\delta}{\partial\mathbf{u}} \right]^{-1}$ represents a vector of four entries which we generally write as

$$\boldsymbol{\psi}^t = \left\{ \frac{\partial J_\delta}{\partial\mathbf{u}} \right\}^t \left[\frac{\partial\mathcal{R}_\delta}{\partial\mathbf{u}} \right]^{-1} \quad (4.9)$$

Here, $\boldsymbol{\psi}^t$ consists of the sensitivities of the local residual with respect to the error in target functional, $\delta\mathcal{J}$, and we refer to this as the adjoint vector. Hence, the first component of $\boldsymbol{\psi}^t$ is the sensitivity of $\mathcal{J}(\mathbf{u})$ to changes in the first component of the residuals. This vector is obtained by solving the continuous adjoint equations that were derived in section 2.2.

Since the exact solution is not known, we are estimating the error in target functional using two discretisation levels as presented in [27]. In our case, we are using the same mesh but two different polynomial orders. We denote the fine solution obtained using a high polynomial order by $\mathbf{u}_H$ and the solution obtained using a low polynomial order by $\mathbf{u}_L$. In order to estimate the error in target functional, we interpolate the lower order solution, $\mathbf{u}_L$, onto the high order solution using the following notation

$$\mathbf{u}_H^L \equiv \mathbf{I}_H^L \mathbf{u}_L \quad (4.10)$$

where $\mathbf{I}_H^L$ represents the coarse-to-fine extrapolation operator. We write the target functional on the fine mesh as $\mathcal{J}(\mathbf{u}_H)$. Furthermore, we use the interpolated

solution $\mathbf{u}_H^L$ to measure the output error in the coarse solution relative to the fine solution by

$$\delta\mathcal{J} \approx \mathcal{J}_\delta(\mathbf{u}_H^L) - \mathcal{J}_\delta(\mathbf{u}_H) \quad (4.11)$$

In a similar fashion, we approximate $\delta\mathcal{R}$. However, $\mathcal{R}_\delta(\mathbf{u}_H) = \mathbf{0}$ so that

$$\delta\mathcal{R} \approx \mathcal{R}_\delta(\mathbf{u}_H^L) - \mathcal{R}_\delta(\mathbf{u}_H) = \mathcal{R}_\delta(\mathbf{u}_H^L) \quad (4.12)$$

we substitute equations (4.11) and (4.12) into equation (4.8) to link the approximate output error to the local residual induced by the extrapolated solution $\mathbf{u}_H^L$ and obtain

$$\mathcal{J}_\delta(\mathbf{u}_H^L) - \mathcal{J}_\delta(\mathbf{u}_H) = \boldsymbol{\psi}^t \mathcal{R}_\delta(\mathbf{u}_H^L) = \sum_{e=1}^{N_{el}} \left(\boldsymbol{\psi}^t \mathcal{R}_\delta(\mathbf{u}_H^L) \right)_e \quad (4.13)$$

where N_{el} is the number of elements in the mesh. We consider the inner product of the sensitivities with the residuals due to the projection of the coarse solution onto the fine mesh for each element individually and sum over the number of elements. To define the goal-based error indicator, we take the absolute value of the inner product of the sensitivities with the corresponding residuals [27, 28].

$$\epsilon_e = |\boldsymbol{\psi}^t \mathcal{R}_\delta(\mathbf{u}_H^L)|_e \quad (4.14)$$

To illustrate the implemented adjoint-based p -adaptation strategy, we use a range of two- and three-dimensional numerical examples. First we show error convergence in lift coefficient, c_l , for subsonic inviscid ($Ma = 0.5$) flow. This is followed by the error convergence in drag coefficient, c_d , for subsonic laminar ($Ma = 0.1$, $Re = 1000$) flow. As a third numerical example we consider transonic flow past a NACA 0012 aerofoil and we determine the effects of shock waves. We conclude with applying the adjoint-based p -adaptation strategy for three-dimensional inviscid flow. The results presented in the following sections are also published in [26].

4.2.1 Two-dimensional subsonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5.0^\circ$).

As a first test case, we consider subsonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5.0^\circ$). We are interested in reducing the error in lift coefficient c_l . The flow solution under these conditions is smooth and we expect an exponential decay of the numerical error as the polynomial order is increased. This case is well suited to illustrate the ability of p -adaptation to increase numerical resolution whilst keeping CPU time costs low.

A steady-state flow solution is calculated using a structured mesh of 874 triangles with a uniform polynomial order of $P = 3$. This is followed by calculating a steady-state adjoint solution using the same mesh but with a uniform polynomial order of $P = 5$. We assume that a steady-state solution has been reached when the lowest of the values of the L_2 norm of the residuals of the conservative variables is smaller than 10^{-8} . Both the steady state flow and adjoint solution are given in figure 4.4.

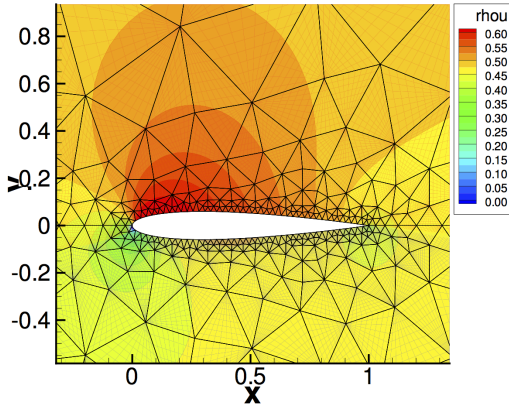
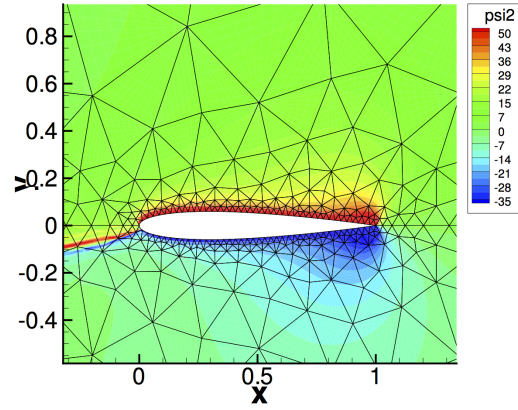
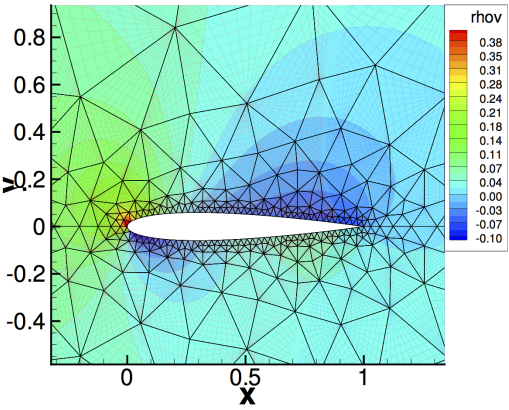
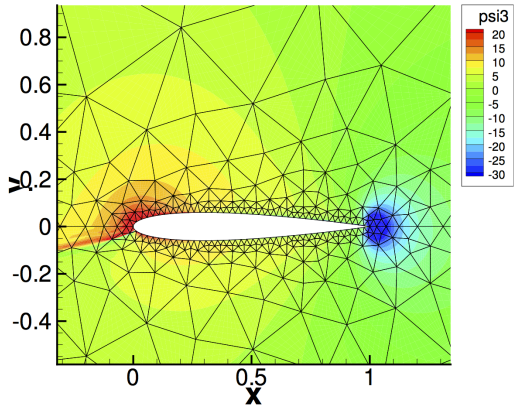
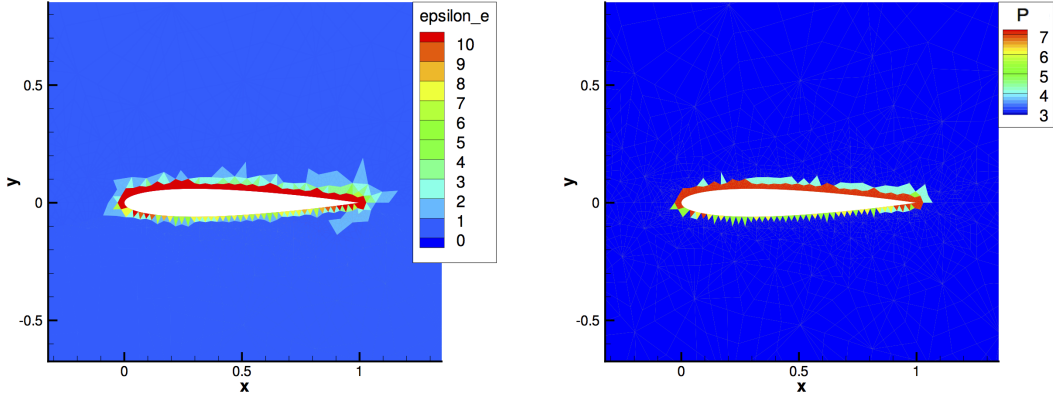
(a) Solution for x -momentum (ρu).(b) Adjoint variable ψ_2 .(c) Solution for y -momentum (ρv).(d) Adjoint variable ψ_3 .

Figure 4.4: The flow solution (left) and the adjoint solution (right) for inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5^\circ$). This is the reference solution which is obtained using a uniform $P = 9$.

Figure 4.4(a) shows the x -momentum contours and figure 4.4(b) shows the contours of the corresponding adjoint solution ψ_2 . The adjoint solution presented in figures 4.4(b) and 4.4(d) can be interpreted in the same way as described in section 3.6.2. A positive perturbation in one of the momentum components where the

corresponding adjoint component is positive, results in an increase in lift while a positive perturbation in momentum in the parts of the domain where the corresponding adjoint component is negative results in a decrease in lift.

The residuals are calculated by interpolating the $P = 3$ solution onto the $P = 5$ mesh and subtracting the interpolated result from the fine result. Furthermore, the adjoint solution is obtained using the $P = 5$ mesh in order to determine the importance of the local residual with respect to the lift coefficient. An element-wise error indicator is calculated using expression (4.14). A new polynomial distribution is determined based on ϵ_e and preset threshold values that are based on the absolute values of the adjoint weighted residuals. An example of the residual and the new polynomial distribution is given in figure 4.5(a) and figure 4.5(b) respectively.



(a) The goal-based error indicator ϵ_e .

(b) The distribution of polynomial orders derived from ϵ_e .

Figure 4.5: An example of the adjoint-based error indicator (left) and the derived polynomial order distribution (right) for inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5^\circ$).

The reference c_l is computed using a uniform $P = 9$ and we define the error as $\epsilon_{c_l} = |c_l - c_l^{\text{ref}}|$. The effectiveness of the proposed goal-based p -adaptation strategy is accessed by monitoring the total number of degrees of freedom (N_Q). For a mesh that consists of solely triangular elements, we calculate N_Q as follows

$$N_Q = \sum_{e=1}^{N_{el}} N_e^{\xi_1} N_e^{\xi_2} \quad (4.15)$$

where N_{el} is the number of elements and $N_e^{\xi_1}$ and $N_e^{\xi_2}$ represent the number of integration points in the ξ_1 and ξ_2 direction respectively. In this case we use a modified expansion basis and therefore $N_e^{\xi_1} = P_e + 2$ and $N_e^{\xi_2} = P_e + 1$ [51].

Furthermore, as a measure of computational cost, we record the CPU time per time step and normalise this with the CPU time per time step for the reference case, in this case when $P = 9$. This quantity is denoted by t_{CPU} . Although both the flow and adjoint solver are designed to run in parallel, in this instance we determine the value of t_{CPU} by running the simulations on a single CPU. In this way, we avoid potential load balancing issues and give a fair assessment of the performance of each simulation.

In table 4.2, the error in lift coefficient is shown obtained using a uniform p -refinement and adjoint-based p -refinement.

Uniform P	$P = 2$	$P = 3$	$P = 5$	$P = 7$	$P = 9$
N_Q	10488	17480	36708	62928	96140
t_{CPU}	0.25	0.27	0.48	0.70	1
c_l	0.6447	0.6574	0.659605	0.65972	0.65972
ϵ_{c_l}	1.5×10^{-2}	2.3×10^{-3}	6×10^{-5}	2.0×10^{-6}	—
Variable P	$2 \leq P \leq 4$	$3 \leq P \leq 6$	$3 \leq P \leq 9$		
N_Q	17844	21422	23892		
t_{CPU}	0.35	0.40	0.54		
c_l	0.659584	0.65969	0.65971		
ϵ_{c_l}	5.00×10^{-3}	8.25×10^{-4}	4.0×10^{-6}		

Table 4.2: Comparison between uniform polynomial refinement and goal-based p -adaptation of the error in lift coefficient for subsonic inviscid flow. The error is calculated with respect to the solution obtained with constant polynomial order $P = 9$.

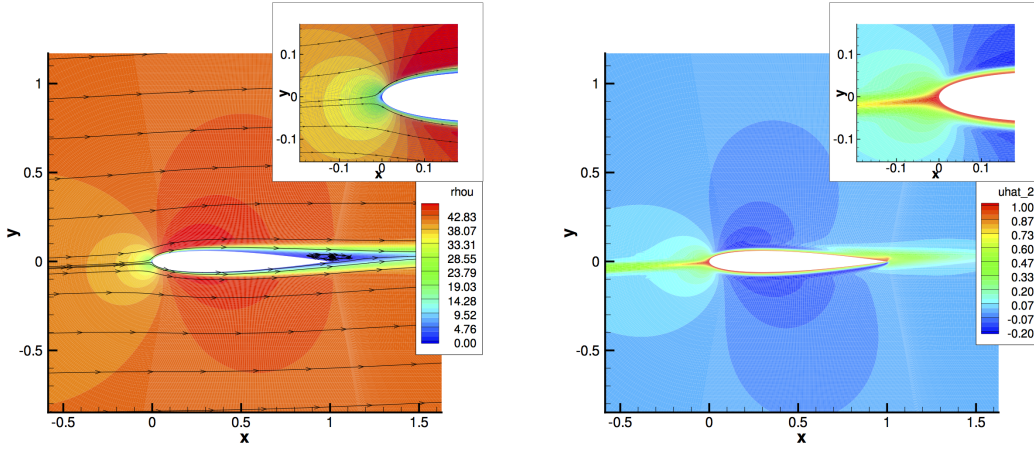
Table 4.2 compares the error in lift coefficient obtained using uniform polynomial refinement, i.e. a sequence of solutions with increasing constant polynomial order, and goal-based p -adaptation. Table 4.2 indicates that for the adaptive simulation with $3 \leq P \leq 9$, we obtain an error between the $P = 5$ and $P = 7$ simulations, but with a reduction of 35% and 62% fewer degrees of freedom respectively. The CPU time per time step is reduced by 46% for the $3 \leq P \leq 9$ case compared to the reference case while achieving a similar level of accuracy. The adjoint-based p -adaptation strategy, using a variable P distribution of $3 \leq P \leq 8$, achieves a bigger reduction in error than when one would use a mesh with uniformly $P = 5$.

4.2.2 Two-dimensional subsonic laminar flow past a NACA 0012 aerofoil ($Ma = 0.1$, $Re = 5000$, $\alpha = 2.0^\circ$).

As a second test case, we consider subsonic laminar flow ($Ma = 0.1$, $Re = 5000$) past a NACA 0012 aerofoil at an incidence of $\alpha = 2^\circ$. For this example, we are interested in improving the estimate for the drag coefficient. We adopt the

same approach as described in the previous numerical example however for this example, the adjoint solution represents the local sensitivity within the domain with respect to c_d . Initially, the steady-state solution for the compressible Navier-Stokes equations is determined. Based on the flow solution, we determine the adjoint solution by linearising around the obtained steady-state solution. For this test case, we solve the adjoint equations presented in expression (2.60). We consider an isothermal wall boundary condition for the flow solution. Hence, equation (2.63) dictates the boundary conditions for the adjoint problem in order to determine the sensitivity with respect to the drag coefficient.

For this case, we use a structured mesh of 3012 quadrilateral shaped elements. The steady-state solution for the governing equations and the corresponding adjoint solution are given in figure 4.6(a) and 4.6(b) respectively. The adjoint solution,



(a) The x -momentum solution (ρu).

(b) The corresponding adjoint variable ψ_2 .

Figure 4.6: The flow solution (left) and the adjoint solution (right) for compressible laminar flow past a NACA 0012 aerofoil ($Ma = 0.1$, $Re = 5\,000$, $\alpha = 2^\circ$). This is the reference solution which is obtained using a uniform $P = 9$.

given in figure 4.6(b), shows a high sensitivity at the leading edge of the airfoil and in the boundary layer which, as expected, indicates that the viscous drag is dominant in this particular case. The high sensitivity at the leading edge indicates that the drag will increase when a positive perturbation in x -momentum occurs which is very intuitive.

A comparison was made between uniform p -refinement and adjoint-based p -refinement. Again, the reference solution is computed using a uniform $P = 9$ and therefore the error is defined as $\epsilon_{c_d} = |c_d^{\text{ref}} - c_d|$. A comparison between uniform p -refinement and adjoint-based p -refinement is given in figure 4.7 in which ϵ_{c_d} is plotted against N_Q . Starting with a low-order ($P = 3$) steady state solution, the error in c_d is decreased by two orders of magnitude by increasing N_Q by a factor of

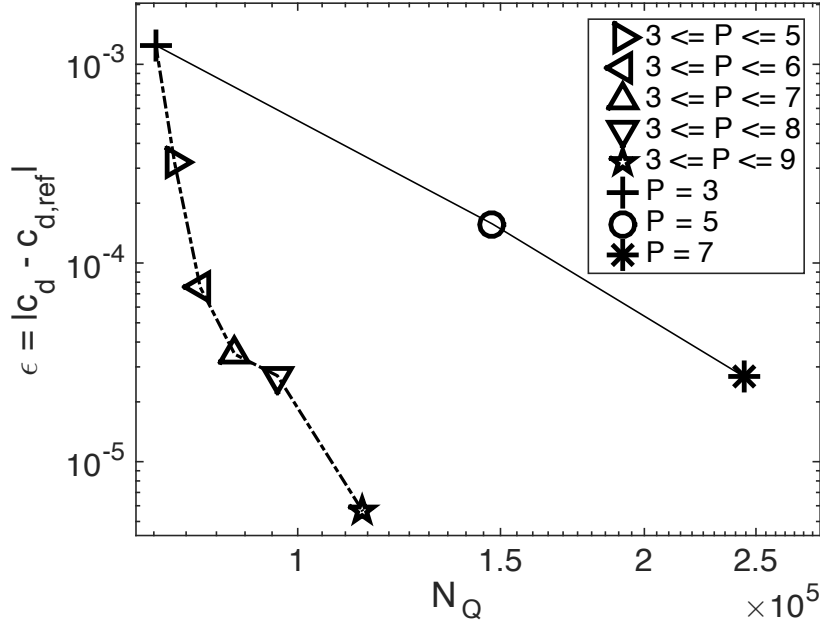


Figure 4.7: Comparison between goal-based p -adaptation and uniform polynomial refinement for compressible laminar flow past a NACA 0012 aerofoil ($Ma = 0.1$, $Re = 5000$, $\alpha = 2^\circ$).

1.4 using adjoint-based p -adaptation. However, for uniform polynomial refinement, the same error reduction is obtained by an increase of N_Q by a factor of 3.0.

The CPU time per time step is reduced as well using a goal-based p -adaptive strategy. Table 4.3 confirms this, by showing the data of figure 4.7 alongside the recorded CPU time. For example the CPU time per time step is reduced by 54% for the $3 \leq P \leq 9$ case compared to the reference case while a similar level of accuracy is achieved as the reference case. The results show a broad increase in performance against the inviscid case considered in the previous section, due to both the sensitivity of c_d in a more localised region near the boundary layer, and the additional computational cost of evaluating the viscous tensor terms for constant polynomial order simulations. However, the results give a clear indication of the increase in performance: the $3 \leq P \leq 8$ simulation exhibits the same error as the solution obtained using constant $P = 7$ but at half of the computational cost.

4.2.3 Two-dimensional transonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).

This case corresponds to a transonic flow, with $Ma = 0.8$ past a NACA 0012 aerofoil at an incidence $\alpha = 1.25^\circ$. Salient features of this flow are a strong shock

Uniform P	$P = 3$	$P = 5$	$P = 7$	$P = 9$	
N_Q	75300	147588	243972	364452	
t_{CPU}	0.28	0.29	0.64	1	
c_d	0.01507	0.01616	0.01629	0.01632	
ϵ_{c_d}	1.20×10^{-3}	1.57×10^{-4}	2.69×10^{-5}	—	
Variable P	$3 \leq P \leq 5$	$3 \leq P \leq 6$	$3 \leq P \leq 7$	$3 \leq P \leq 8$	$3 \leq P \leq 9$
N_Q	78183	82286	88165	96013	113909
t_{CPU}	0.31	0.32	0.34	0.36	0.45
c_d	0.01599	0.01624	0.01628	0.01629	0.01631
ϵ_{c_d}	3.19×10^{-4}	7.55×10^{-5}	3.47×10^{-5}	2.71×10^{-5}	5.63×10^{-6}

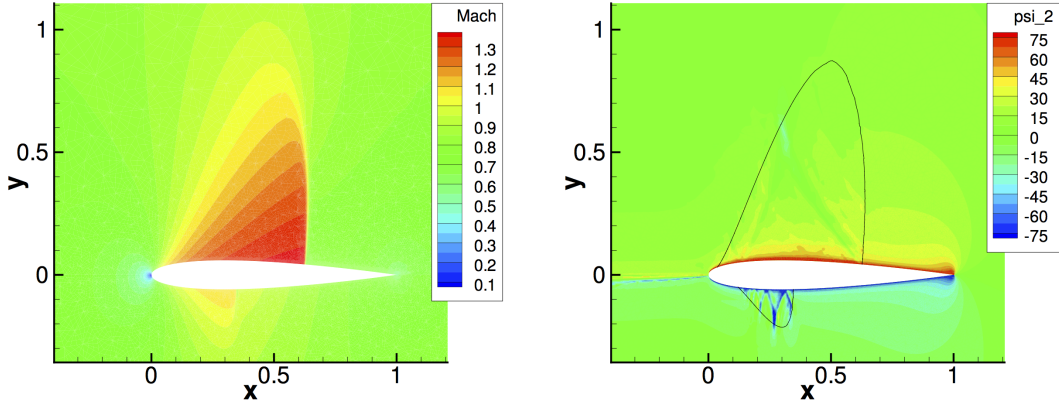
Table 4.3: Comparison between uniform polynomial refinement and goal-based p -adaptation for subsonic laminar flow. The error is calculated with respect to the solution obtained using constant $P = 9$.

on the suction side of the aerofoil at 60% of the chord and a weaker shock on the pressure side at approximately 35% of the chord, both measured from the aerofoil's leading edge.

The target functional is the lift coefficient and it is evaluated following the same methodology used in the previous two cases. The computational domain is discretised into an unstructured mesh consisting of 8 446 triangles. The direct and adjoint steady-state solutions calculated using constant polynomial order $P = 6$ are shown in figures 4.8(a) and 4.8(b), respectively. This is the reference solution for this case. The presence of dissipation terms that filter the high-order frequencies of the solution in the vicinity of shocks to ensure stability, does not justify the use solutions at higher polynomial orders as reference.

The contour map of the adjoint variable ψ_2 in figure 4.8(b) shows clear traces of the characteristics carrying acoustic perturbations, i.e. flow compressions and expansions, throughout the supersonic regions which are enclosed by the Mach lines ($Ma = 1$) delineated in black in the figure. This is a clear illustration of the ability of the adjoint formulation to capture the mathematical essence of the problem. The adjoint solution also highlights the region at the foot of the strong shock on the suction side of the aerofoil as the major contributor to drag. In general, the sensitivity is highest in the vicinity of the shock but reduces significantly away from it.

The presence of shocks in the flow field requires shock capturing to stabilise the simulations. This is accomplished here through the introduction of appropriate solution-dependent dissipative terms as shown in section 3.3. We use the discontinuity sensor proposed in [76] which identifies the presence of a shock by quantifying the smoothness of the solution within an element through comparison of solutions at two different polynomial orders. As mentioned before, the discontinuity sensor is



(a) Mach number

(b) Adjoint variable for ψ_2 .

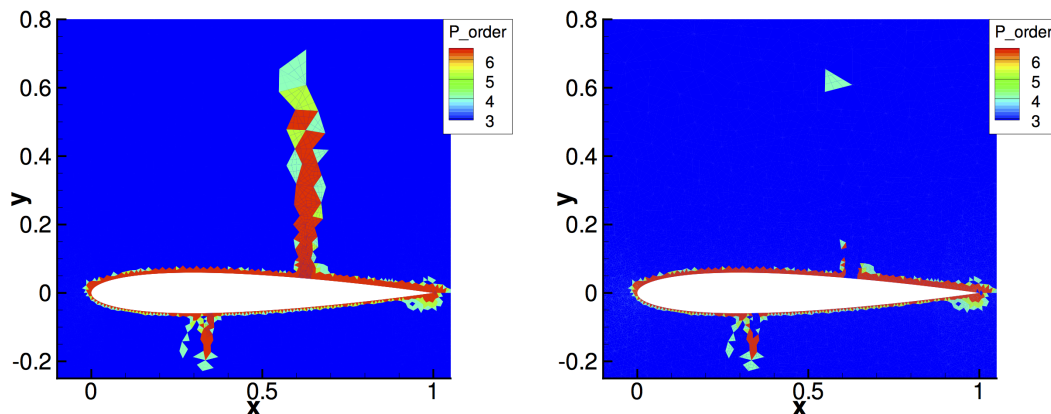
Figure 4.8: The flow solution (left) and the adjoint solution (right) for inviscid transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).

used to selectively apply the dissipative terms that dump numerical oscillations by effectively lowering the polynomial order approximation in those elements affected by the shock.

To assess the performance of the p -adaptive method in this context, we perform two simulations using constant order polynomials $P = 3$ and $P = 6$, and two p -adaptive simulations with polynomial orders in the range $3 \leq P \leq 6$. In the first p -adaptive simulation we apply the adaptation criterion as before in the whole domain without any special treatment of the regions affected by the shock. For the second p -adaptive simulation, we use the sensor (3.37) to identify the shock location and keep the polynomial order low (typically $P = 3$) in those elements within the vicinity of the shock. The resulting distribution of polynomial orders from the p -adaptive simulations without and with polynomial order restriction are shown in figures 4.9(a) and 4.9(b), respectively.

Table 4.4 presents the errors in the lift coefficients computed in these four cases with respect to a reference solution published by Yano and Darmofal [99, 95] that gives a value $c_l^{\text{ref}} = 0.35619$. Analysing the values of the relative error in table 4.4, it can be concluded that not much is gained by using solely p -adaptation using the goal-based error indicator. The comparison between the two polynomial distributions with and without restrictions in the polynomial order of the elements in the vicinity of the shock shows that the decrease in error is roughly comparable.

Since we are using a shock capturing scheme that effectively reduces the order of the polynomial in the vicinity of a discontinuity through the addition of artificial viscosity, these results indicate that we achieve little gain in accuracy by allowing



(a) Unrestricted polynomial order.

(b) Polynomial order restricted to $P = 3$ at shocks.

Figure 4.9: Comparison between the polynomial distributions obtained for the inviscid transonic flow case using the goal-based error indicator without restriction of the polynomial order at shocks (left) and with restriction of the polynomial order to $P = 3$ (right).

	$P = 3$	$3 \leq P \leq 6$	$3 \leq P \leq 6$ (shock filter)	$P = 6$
N_Q	168920	193460	189798	472976
t_{CPU}	0.65	0.72	0.70	1
c_l	0.333	0.336	0.334	0.336
ϵ_{c_l}	0.0232	0.0202	0.0222	0.0202

Table 4.4: Comparison between uniform polynomial refinement and goal-based p -adaptation of the error in lift coefficient for transonic inviscid flow. Furthermore, data obtained using restricted and unrestricted goal-based p -adaptation is given. The error is calculated with respect to the value $c_l^{\text{ref}} = 0.35619$ evaluated in references [99, 95].

the order of the polynomial to increase in those elements near the shock.

The reference solution, c_l^{ref} , is merely used here to indicate that the potential gain of using solely p -adaptation is limited. The accuracy of the solution in the vicinity of the shock wave on the upper side of the airfoil is of dominant importance when determining the drag coefficient. As a result, the combination of goal-based p -adaptation with some other form of mesh adaptation that allows the reduction in size of the elements is preferred for transonic flows. Therefore, these results also suggest that it would be preferable to increase the resolution there by keeping the polynomial degree fixed and decreasing the element size.

4.2.4 Three-dimensional subsonic inviscid flow past an ellipsoid of revolution ($Ma = 0.5$, $\alpha = 3.0^\circ$).

Finally we apply the goal-based p -adaptation technique to obtain improved estimates of lift for a three-dimensional inviscid subsonic flow past an ellipsoid of revolution with a free-stream Mach number $Ma = 0.5$. The lengths of the semi-axes of the ellipsoid are $0.2r$, r and $0.2r$, with $r = 0.25$. The flow considered is symmetric, the symmetry plane contains the major semi-axis which is at an incidence of 3° with the free-stream velocity vector. We use an unstructured mesh of 23 366 tetrahedra surrounding the ellipsoid. The ellipsoid is located in the centre of a $[0, 10]^3$ box. The semi-major axis of the ellipsoid is aligned with the y -axis. In this case, the z -axis is taken to be pointing upwards with respect to the ellipsoid and the x -axis pointing sideways as shown in figure 4.10. In this case $(x_1, x_2, x_3) = (x, y, z)$. In this particular example, we take the lift coefficient as a target quantity meaning that for this three-dimensional flow we have that $\vec{\theta}_l = \{-\sin \alpha, 0, \cos \alpha\}^t$. The reference value of the lift coefficient, c_l^{ref} , is calculated from a steady-state solution obtained using a constant polynomial order $P = 4$. This permits us to afford the computational cost of this three-dimensional simulation. The solutions for the velocity components together with their corresponding adjoint variables are depicted in figure 4.11.

As in the previous examples, we compute a goal-based error indicator using equation (4.14). The absolute value of this error indicator is used to define a set of threshold values according to which we decide whether to increase the polynomial order or not at an element. After applying this strategy, the corresponding variable polynomial order distribution is shown in figure 4.12. Since this case uses three-dimensional curvilinear elements, the choice of quadrature order can affect stability as shown in [68]. We therefore increase the number of integration points with a factor of 2, compared to the previous two-dimensional simulations, in each direction in order to avoid aliasing effects. The number of quadrature points, N_Q , in three

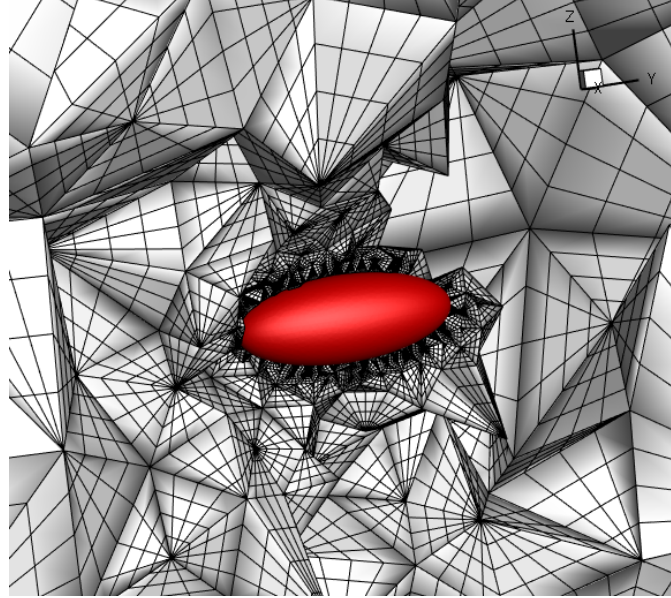


Figure 4.10: The tetrahedral mesh around an ellipsoid of revolution shown here with the degrees of freedom corresponding to a constant polynomial order $P = 4$.

	$P = 2$	$2 \leq P \leq 4$	$P = 4$
N_Q	5.9×10^6	12.8×10^6	25.7×10^6
t_{CPU}	0.25	0.66	1.0
c_l	0.9658	1.0006	1.0005
ϵ_{c_l}	0.0347	10^{-4}	-

Table 4.5: Comparison between uniform polynomial refinement and goal-based p -adaptation of the error in lift coefficient for three-dimensional inviscid flow past an ellipsoid. The error is calculated with respect to the reference lift coefficient corresponding to $P = 4$.

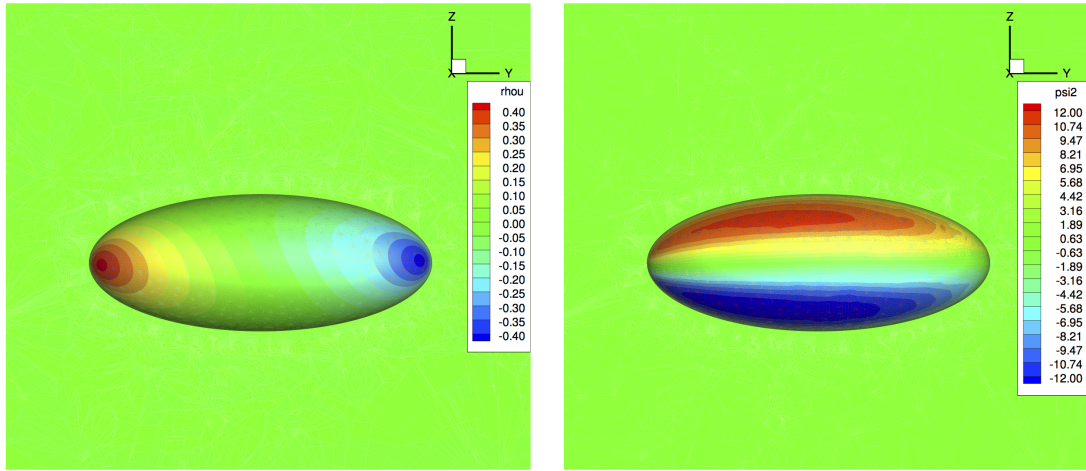
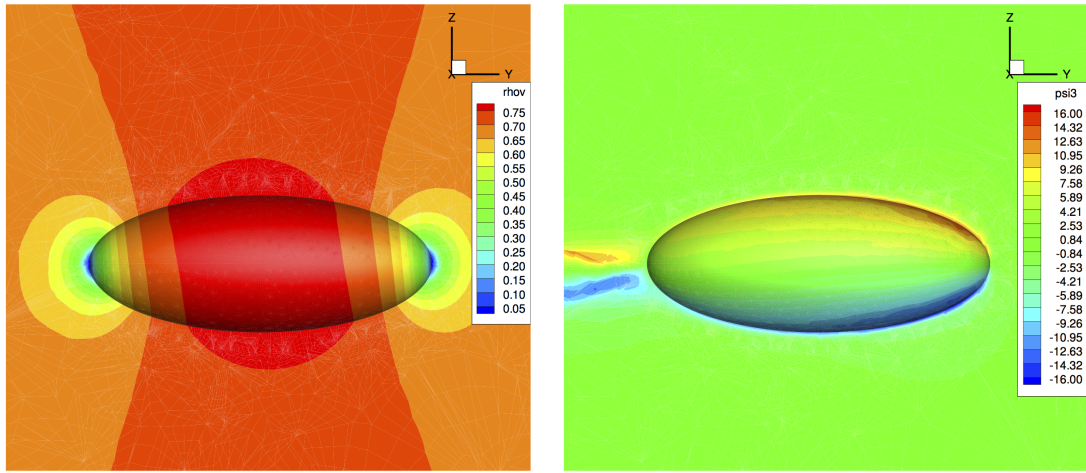
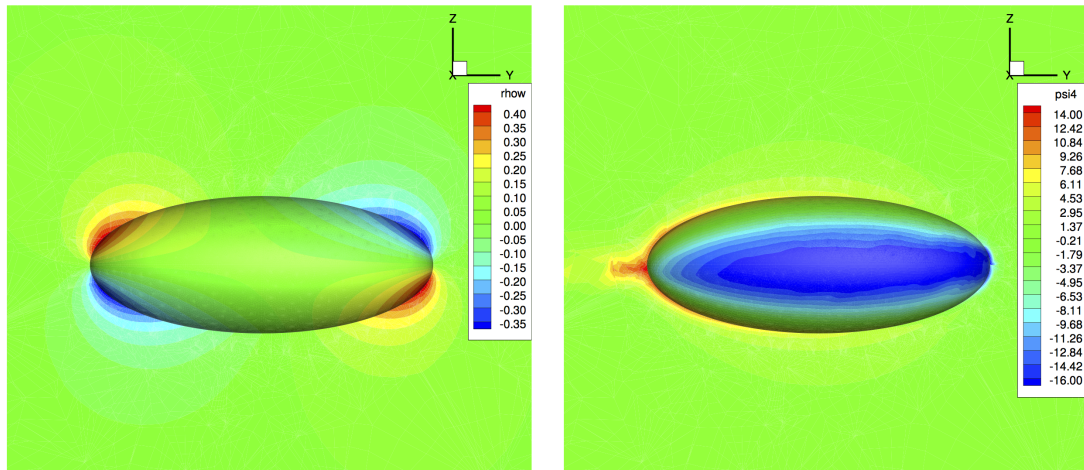
(a) The x -momentum solution (ρu).(b) Adjoint variable ψ_2 .(c) The y -momentum solution (ρv).(d) Adjoint variable ψ_3 .(e) The z -momentum solution (ρw).(f) Adjoint variable ψ_4 .

Figure 4.11: The solution for the velocity components in each Cartesian direction (x to z from top to bottom in the left-hand side column) and for their corresponding adjoint variable (ψ_2 to ψ_4 from top to bottom in the right-hand side column) for the flow past an ellipsoid of revolution at an incidence of 3° .

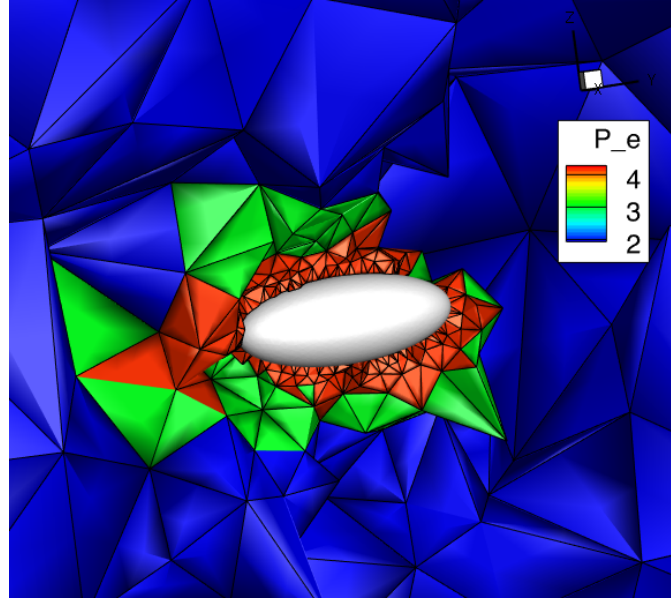


Figure 4.12: The new polynomial distribution based on the goal-based error indicator for the inviscid flow past an ellipsoid of revolution.

dimensions is

$$N_Q = \sum_{e=1}^{N_{el}} 2^3 N_e^{\xi_1} N_e^{\xi_2} N_e^{\xi_3}, \quad (4.16)$$

where $N_e^{\xi_i}$ is the number of quadrature points, in the parametric direction ξ_i , within element e , and depends on P . Table 4.5 shows a comparison of the error in lift coefficient calculated using goal-based p -adaptation and constant polynomial orders $P = 2$ and $P = 4$. The difference between the value of c_l obtained using goal-based p -adaptation and that using constant $P = 4$ is very small. However the goal-based p -adaptation strategy requires less than half of the resolution compared to the constant $P = 4$ solution and, in terms of CPU time per time step, it is decreased by approximately 30%. The error in the lift coefficient, c_l , is also reduced by two orders of magnitude compared to the $P = 2$ solution.

Chapter 5

Mesh deformation using a thermo-elastic analogy

To calculate solutions using a high-order spectral element discretisation, a high-order mesh is required too. Usually, low-order methods demand significant refinement at the wall to accurately approximate the curvature of the geometry. For boundary conforming high-order methods, the curvature of the geometry is embedded in the mesh meaning that adjacent elements to a wall geometry are curved. As a result, the mesh is allowed to be coarser than for the classical low-order methods. One way of generating a high-order mesh is to start with an initial linear mesh and project the adjacent element edge onto the curvature of the geometry as shown in figure 5.1.

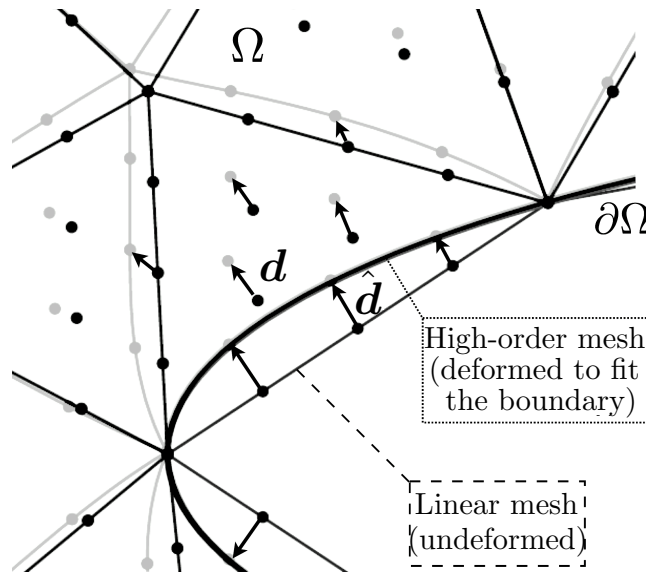


Figure 5.1: Mesh deformation induced by a displacement of the boundary.

By treating the mesh as an elastic solid, we apply external loading that forces the mesh to deform. To prevent elements near the boundary to self-intersect, we calculate a local displacement of the surrounding nodes that is based on a linear elastic formulation. Figure 5.1 illustrates how the degrees of freedom surrounding the boundary edge are moving when mesh deformation is applied based on the curvature of the geometry. This technique is well documented in the literature and has been applied to generate curved boundary conforming meshes from linear meshes by [50, 77, 98]. Furthermore, the linear elastic analogy for mesh deformation has been used to generate meshes when dealing with aeroelastic flow problems as shown in [24, 85, 44] for example.

In this chapter, we incorporate an additional pseudo-thermal stress term into the linear elasticity equations initially intended to improve robustness of the classical linear elastic mesh deformation technique. By considering a thermo-elastic formulation for mesh deformation, we can *heat-up* or *cool down* certain elements to expand or contract them depending on the local quality of the mesh. Hence, we apply the pseudo-thermal stress term to control mesh quality meaning that once an element tends to become invalid, the element is expanded or contracted depending on what is necessary to prevent the element from becoming invalid. This requires the additional pseudo-thermal stresses to depend on a local mesh quality indicator. In this section, we investigate two types of mesh quality indicators namely an isotropic and an anisotropic quality indicator.

The idea of using a thermo-elastic formulation for mesh deformation is extended. In addition to applying this formulation to control mesh validity, we also perform mesh refinement for shock waves. We refer to this technique as *r*-adaptation (adaptation through redistribution) for shock waves. Transonic or supersonic flows are characterised by shock waves which represent local discontinuities in flow properties. Usually, the local accuracy is improved by means of mesh adaptation to approximate these high gradients in flow properties. Typically when mesh adaptation is applied for shocks, the elements in the vicinity of the shock are split into smaller elements (*h*-adaptation). The polynomial order is kept low near the shock and the mesh size is decreased in order to avoid numerical oscillations (Gibbs phenomenon). However this makes the problem computationally more expensive since generally more elements are added to the problem. To prevent this, we shrink the elements in the vicinity of shock waves by *cooling* the elements that cover the shock waves and redistribute the nodes using the thermo-elastic analogy. Next to the fact that no extra degrees of freedom are added, the *r*-adaptation strategy prevents the problem of having to repartition the newly obtained mesh since the connectivity of the degrees of freedom is preserved.

This chapter is structured as follows: First, the general formulation of the

linear thermo-elastic equations is introduced. This is followed by the description of two different formulations of the pseudo-thermal stresses that are used for mesh quality control. An isotropic and anisotropic formulation are described and the two different approaches are compared for a set of simple test cases. Finally, r -adaptation for shock waves using the thermo-elastic analogy is introduced and illustrated for the test case of transonic flow ($Ma = 0.8$) past a NACA 0012 aerofoil at an angle of attack of 1.25° and supersonic inviscid flow ($Ma = 3.0$) past a forward facing step.

5.1 The linear thermo-elastic equations

To illustrate the linear thermo-elastic formulation for mesh deformation, we again consider the deformation of straight sided high-order elements to accommodate curved boundaries as shown in figure 5.1. Hence, we are looking for the displacement vector field $\mathbf{d} = \{d_1, \dots, d_N\}^t$ that is illustrated in figure 5.1. The linear thermo-elastic equations are given by

$$\nabla \cdot (\mathbf{S}_e + \mathbf{S}_t) + \mathbf{f} = 0; \quad \mathbf{d} \in \Omega \quad (5.1)$$

where $\mathbf{f}$ represents an arbitrary force in the domain and $\mathbf{S}_e$ is the stress tensor that is given by

$$\mathbf{S}_e = \lambda \text{Tr}(\mathbf{E}) \mathbf{I} + \tilde{\mu} \mathbf{E} \quad (5.2)$$

$\mathbf{I}$ denotes the identity matrix and the strain tensor $\mathbf{E}$ depends on the displacement vector $\mathbf{d}$ and is given by

$$\mathbf{E} = \frac{1}{2} (\nabla \mathbf{d} + (\nabla \mathbf{d})^t) \quad (5.3)$$

The Lamé constants λ and $\tilde{\mu}$ are described by the Young's modulus $\tilde{E}$ and the Poisson ratio ν and are written as

$$\lambda = \frac{\nu \tilde{E}}{(1 + \nu)(1 - 2\nu)}, \quad \tilde{\mu} = \frac{\tilde{E}}{2(1 + \nu)} \quad (5.4)$$

In equation (5.1), the pseudo-thermal stress tensor is denoted by $\mathbf{S}_t$. Depending on the formulation of $\mathbf{S}_t$, we can use it either to apply mesh quality control or to perform r -adaptation. Note that in the remainder of this chapter, the application of the thermal stress tensor in certain parts of the domain is often referred to as *heating* or *cooling down* the elements.

5.2 Mesh deformation for mesh quality control

For mesh quality control, the pseudo-thermal stress tensor $\mathbf{S}_t$ depends on a quantity that indicates the level of deformation for each element. In this chapter, we discuss two quality measures which can be used in the introduced thermo-elastic formulation. One isotropic and one anisotropic formulation. The results shown in this section are also presented in [70].

5.2.1 The Jacobian as an isotropic temperature term

The Jacobian relates the elemental area ($N = 2$) or volume ($N = 3$) in reference space to the elemental area or volume in physical space. The Jacobian of an element is determined using the mapping of the defined reference element, where the coordinates are given by $\mathbf{x} \in \Omega_e^r$, to the physical element where the coordinates are given by $\mathbf{y} \in \Omega_e$. Here, the reference element is defined by the linear approximation of the physical element as shown in figure 5.2.

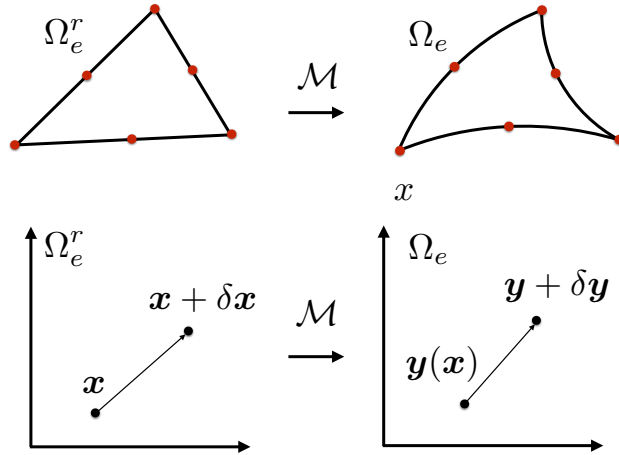


Figure 5.2: Definition of the linear reference element and the corresponding mapping.

We can also write

$$\mathbf{y} + \delta \mathbf{y} = \mathbf{y}(\mathbf{x} + \delta \mathbf{x}) \approx \mathbf{y}(\mathbf{x}) + \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \delta \mathbf{x} \rightarrow \delta \mathbf{y} \approx \mathbf{J} \delta \mathbf{x} \quad (5.5)$$

where the Jacobian matrix $\mathbf{J}$ of the mapping $\mathcal{M}$ is defined as

$$[\mathbf{J}]_{i,j} = \frac{\partial y_i}{\partial x_j} \quad (5.6)$$

The Jacobian is normalised by the area of the element by considering the reference element to be a linear approximation of the physical element. This prevents having

a large range of absolute values for the Jacobians in the domain since usually the domain is discretised using elements that significantly vary in size. In this way we only take the curvature of the element into account when determining the Jacobians.

The element is inverted with negative or zero area when $\det(\mathbf{J}) \leq 0$. This condition is therefore an appropriate condition that indicates the validity of the element. To measure the effect of the isotropic temperature term of the solution, we determine a scaled Jacobian that is given by

$$J_s = \frac{\min J(\xi)}{\max J(\xi)} \quad (5.7)$$

where ξ is the coordinate in the standard element reference frame, $J(\xi)$ is the Jacobian of the mapping $\mathcal{M}(\circ)$ evaluated at ξ and the maximum and minimum is evaluated at a tensor product of $(P+1)^n$ quadrature point within Ω_{st} . By assuming that the pseudo-thermal stresses are modelled using a linear isotropic material, the form of the pseudo-thermal stress tensor becomes

$$\mathbf{S}_t = \beta (T_p - T_{p,0}) \mathbf{I} \quad (5.8)$$

where T_p is the pseudo-temperature, $T_{p,0}$ is the pseudo-temperature of the stress-free state and β is a thermal coefficient that scales the applied temperature. One of the first obvious choices is to let the temperature T depend on the Jacobian, $\mathbf{J}$, of the element.

5.2.2 The metric tensor as an anisotropic temperature term

The previously introduced Jacobian does not take directional information into account and therefore anisotropic elemental distortion is not accounted for. By considering the metric tensor of the element, we determine anisotropic pseudo-thermal stresses and expand or contract elements in a certain direction. The ratio of lengths in the physical and reference spaces is given by the metric tensor which is defined as follows

$$\|\delta \mathbf{y}\|^2 = \delta \mathbf{y}^t \delta \mathbf{y} = \delta \mathbf{x}^t \mathbf{J}^t \mathbf{J} \delta \mathbf{x} = \delta \mathbf{x}^t \mathbf{G} \delta \mathbf{x} \rightarrow \|\delta \mathbf{y}^2\| = \|\delta \mathbf{x}\|^2 \mathbf{d}^t \mathbf{G} \mathbf{d} \quad (5.9)$$

where $\mathbf{d}$ is a unit vector parallel to $\delta \mathbf{x}$. The metric tensor, $\mathbf{G}$ is defined as the right Cauchy-Green deformation tensor and, based on the mapping of the previously described reference element to the physical element, it is given by

$$\mathbf{G} = \mathbf{J}^t \mathbf{J} \quad (5.10)$$

The metric tensor tells us in which direction the deformation of the element takes place. The principal directions of the strains can be determined by formulating an eigenvalue problem. By seeking the extrema of the ratio $\hat{\rho} = \frac{\|\delta \mathbf{y}\|^2}{\|\delta \mathbf{x}\|^2}$ subject to $\|\mathbf{d}\| = 1$. Using Lagrange multipliers, we minimise the functional

$$\hat{\rho} = \mathbf{d}^t \mathbf{G} \mathbf{d} + \lambda (1 - \|\mathbf{d}\|^2) \quad (5.11)$$

Setting the derivatives of $\hat{\rho}$ with respect to $\mathbf{d}$ equal to zero leads to the eigenvalue problem

$$\frac{\partial \hat{\rho}}{\partial d_i} = 2 \sum_{j=1}^N G_{i,j} d_j - 2\lambda d_i = 0, \quad i = 1, \dots, N \quad \rightarrow \quad (\mathbf{G} - \lambda \mathbf{I}) \mathbf{d} = 0 \quad (5.12)$$

The metric tensor is symmetric and positive definite when $\det(\mathbf{J}) > 0$. As a result, the eigenvalues are positive and correspond to the maximum and minimum values of $\hat{\rho}$. The metric tensor can be written as

$$\mathbf{G} = \mathbf{T} \mathbf{\Lambda} \mathbf{T}^{-1} \quad (5.13)$$

where $\mathbf{T}$ and $\mathbf{\Lambda}$ are the matrices of eigenvectors and eigenvalues of $\mathbf{G}$ respectively. To interpret the eigenvalues of the metric tensor we use a two-dimensional example. Figure 5.3 shows a circle with a radius L which is transformed into an ellipse with semi-axes L_1 and L_2 . These semi-axis are aligned with the eigenvectors of the metric tensor $\mathbf{G}$. The eigenvalues can be interpreted as a scaling of the strains in these principal directions. Hence by appropriately modifying the eigenvalues following from the eigendecomposition, we counteract the deformation. The strains in the principal directions are given by

$$\lambda_n = \frac{L_i - L}{L} = \sqrt{\lambda_i} - 1, \quad i = 1, \dots, N \quad (5.14)$$

We use these newly defined eigenvalues to counteract the mapping deformation. Therefore we define the pseudo-thermal stress tensor based on the modified metric tensor as follows

$$\mathbf{S}_t = \mathbf{T} \mathbf{\Lambda}_t \mathbf{T}^{-1}, \quad D_{t,i} = \zeta \lambda_n \quad (5.15)$$

where ζ is the anisotropic thermal coefficient. The effect of both the isotropic and anisotropic pseudo-thermal stress terms are demonstrated using two numerical examples.

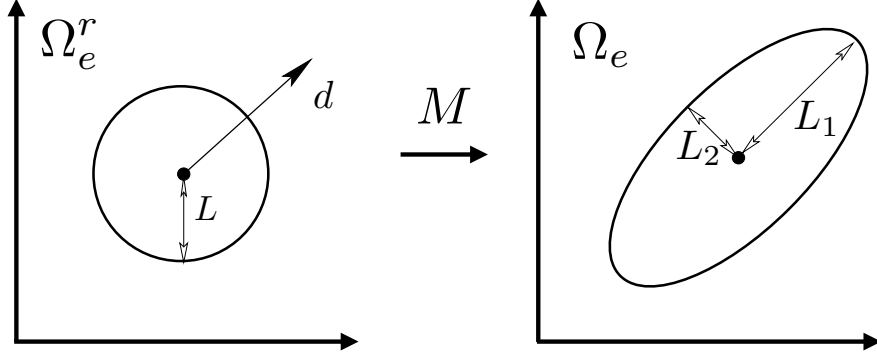


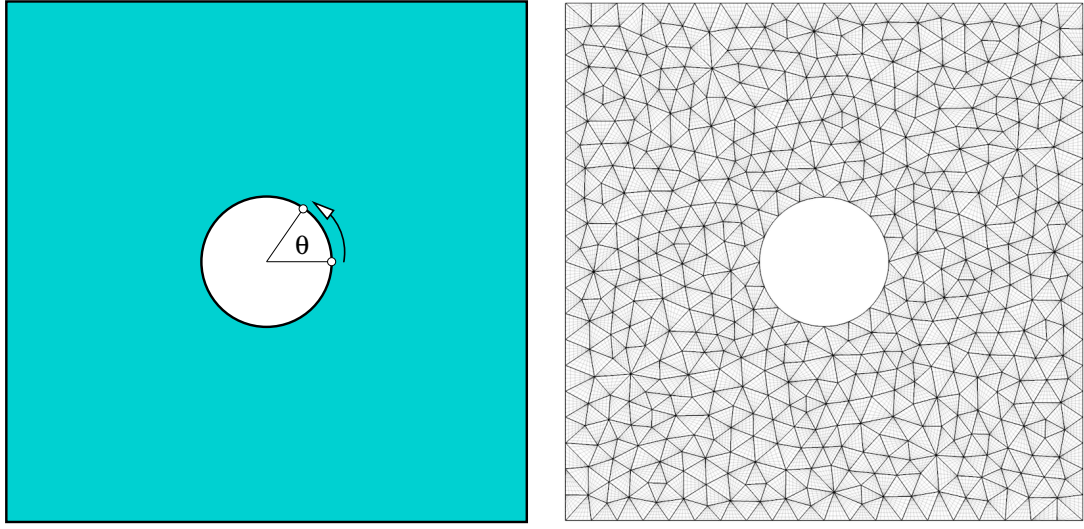
Figure 5.3: A two-dimensional representation of the local deformation generated by the mapping.

5.2.3 Rotating circular cylinder

To illustrate the effectiveness of the additional pseudo-thermal stress term in the linear elastic formulation, we use a rotating circular cylinder of $r = 0.1$. An initial square mesh is generated around the cylinder in the region $[-1, 1]^2$. The wall of the circular cylinder is rotated an angle θ and we monitor the maximum rotational angle, θ_m , until the determinant of the Jacobian becomes negative in the mesh. The computational domain of this test case is given in figure 5.4.

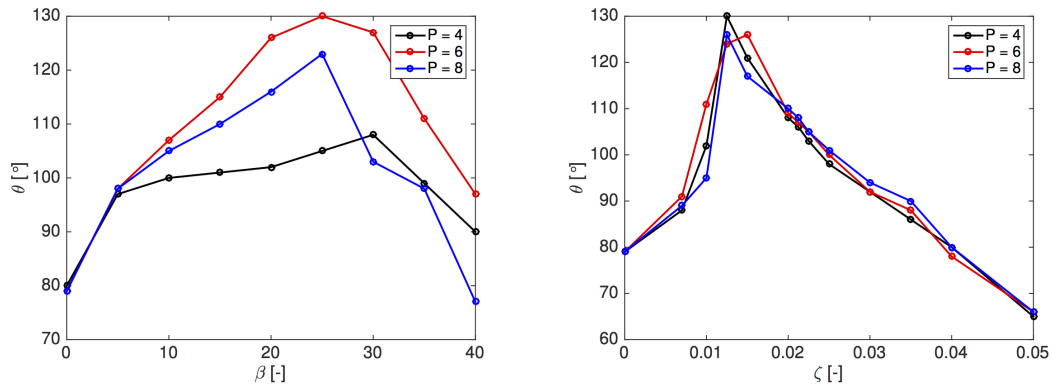
We first consider the isotropic pseudo-thermal stress formulation that was introduced in section 5.2.1. Hence we use equation (5.8) with $T_p = J_i$ where J_i is the value of the Jacobian at the i^{th} node. In figure 5.5(a), the maximum rotation angle θ_m is plotted against the thermal coefficient β for three different polynomial orders. Figure 5.5(a) shows that once the isotropic control temperature is applied, a higher rotational angle is achieved for all polynomial orders however, there is a clear sensitivity with respect to the polynomial order. From figure 5.5(a), can be concluded that for $20^\circ \leq \beta \leq 25^\circ$ we find an optimal rotational angle of $120^\circ \leq \theta_m \leq 130^\circ$ which is a significant improvement compared to when we do not apply any thermal control ($\beta = 0, \theta = 79^\circ$).

We apply the anisotropic pseudo-thermal stress term in a similar way as the isotropic one. Instead of using equation (5.8), we now use equation (5.15). In figure 5.5(b), we plotted θ_m against the anisotropic thermal coefficient ζ . Again we see



(a) The computational domain is a square plate with a circular hole and the points in the circle are rotated about its center by an angle θ . (b) Initial mesh consisting of 1031 high-order triangles (thicker outline) with interior points shown (thin outline).

Figure 5.4: Test case used to illustrate the effect of pseudo-thermal stresses.



(a) Maximum rotation angle θ_m plotted against the isotropic thermal coefficient β for the isotropic temperature term. (b) Maximum rotation angle θ_m plotted against the anisotropic thermal coefficient ζ for the anisotropic temperature term.

Figure 5.5: Comparing the performance of the anisotropic and anisotropic pseudo-thermal stress term.

that once we apply the pseudo-thermal stress term, we are able to increase the rotational angle. Now we also see that there is little dependency on the polynomial order since the lines for different polynomial orders collapse nicely with each other. From figure 5.5(b) can be concluded that the maximum rotation angle is achieved for $\zeta \approx 0.015$ where $\theta_m \approx 130^\circ$. Based on the difference between figure 5.5(a) and figure 5.5(b), we conclude that both pseudo-thermal stress formulations can achieve a similar rotational angle however for the anisotropic formulation this is independent from the polynomial order that is used.

To illustrate the anisotropic behaviour of this pseudo-thermal stress term, we have plotted the stress tensor field for two different rotational angles in figure 5.6. This figure illustrates how the thermal stresses are going to counteract the stretching of each element at each mesh point.

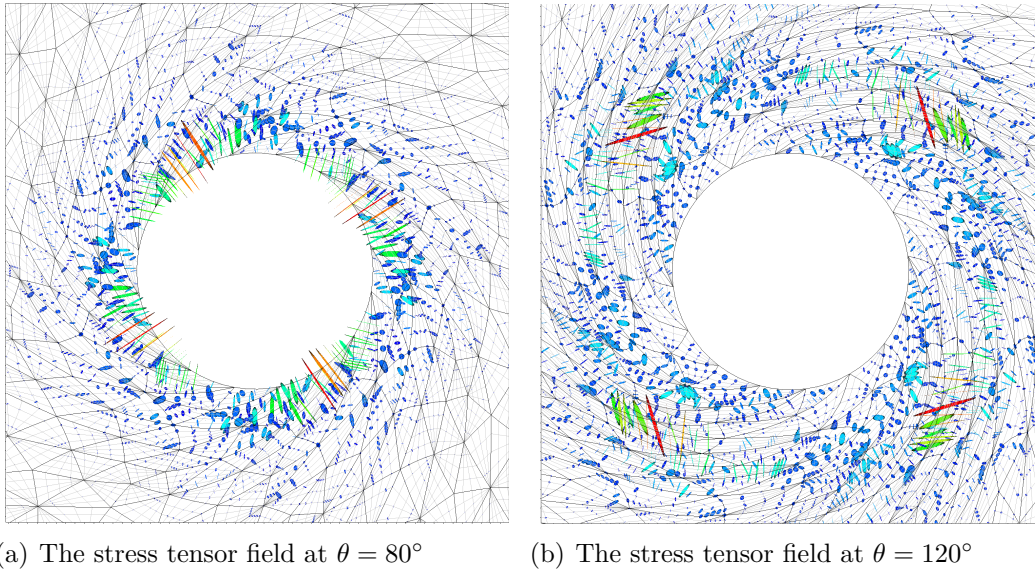


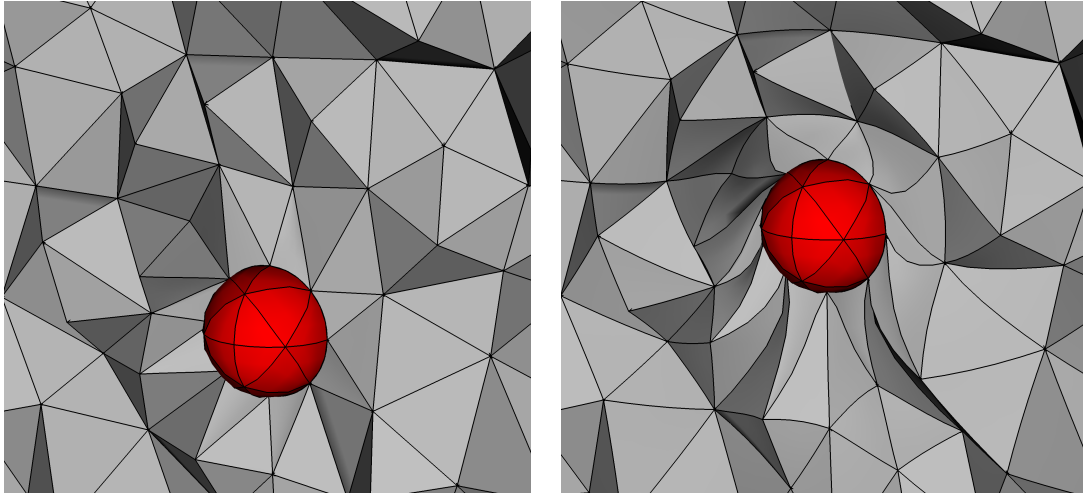
Figure 5.6: Visualization of the pseudo-thermal stresses for two different values of rotational angle. The lengths of the axes of an ellipse in the plots represent the eigenvalues of the pseudo-thermal stress tensor and the axes are orientated along the direction of its orthogonal eigenvectors.

5.2.4 Moving sphere in a cube

As a second example, we consider a three-dimensional problem to show that the pseudo-thermal stress term also has a beneficial effect in three dimensions. Instead of considering a rotational deformation, we consider a translational deformation. A high-order mesh is generated for a sphere that has a radius of 0.25 which is situated in the middle of a cube of dimensions $[-2, 2]^3$. The mesh has 8347 curvilinear tetrahedra elements at $P = 4$. A cut through the mesh is shown in figure 5.7(a).

This figure shows the faces of tetrahedra in the interior of the domain through a $y - z$ plane. The sphere is repeatedly translated in z -direction by steps of $\Delta z = 0.01$. This procedure is carried out until we detect that the determinant of the Jacobian is negative and we determine the maximum translation denoted by Δz_m . In figure 5.7(b), the deformed mesh is presented. The mesh is deformed due to a translation in z direction of $\Delta z = 0.40$. We use the anisotropic pseudo-thermal stress terms and we recorded Δz_m corresponding to different values of thermal coefficient ζ in table 5.1. Note that for $\zeta = 0$ we obtain $\Delta z = 0.43$ which resembles the classical linear elastic formulation. Using the appropriate ζ parameter, we can increase the maximum translation with 16%. However, the effects are not as clear as for the two-dimensional test cases shown in the previous section.

Both the numerical examples presented in section 5.2.3 and in this section serve as a proof of concept that we can apply the pseudo-thermal stress terms to control the quality of the mesh. The implementation of the linear thermo-elastic solver is rather flexible and therefore, we extended the idea of applying mesh deformation using a thermo-elastic analogy for mesh adaptation for shock waves. This is presented in the next section.



(a) Cross-section of the initial mesh through the $y - z$ plane, showing the surface mesh of the sphere. (b) Cross-section of the mesh after a displacement $\Delta z = 0.4$.

Figure 5.7: Configuration of the sphere in a cube example.

5.3 r -adaptation for shock waves

Compressible flows are characterised by shock waves which occur in the transonic or supersonic flow regime. Depending on the Mach number, the region of the

$\zeta \times 10^{-3}$	0	1	2	3
Δz_m	0.43	0.5	0.42	0.41

Table 5.1: Values of the maximum displacement Δz_m , for the three-dimensional example of figure 5.7, versus the scaling coefficient for the anisotropic pseudo-thermal stress terms controlled through the parameter ζ .

shock is very thin and usually falls within the domain of a single element causing the gradients within this element to be very large. The thickness of a shock wave is the spatial domain where the variation in flow properties is irreversible and where entropy increases. By considering the one-dimensional Navier-Stokes equations, we can estimate the thickness of a stationary shock wave. Assuming a stationary shock and by integrating the one-dimensional Navier-Stokes once in space we obtain

$$C_\rho = \rho v \quad (5.16)$$

$$C_{\rho v} = p + \rho v^2 - \frac{4}{3}\mu \frac{\partial v}{\partial x} \quad (5.17)$$

$$C_{\rho E} = \rho v H - v \frac{4}{3}\mu \frac{\partial v}{\partial x} - k \frac{\partial T}{\partial x} \quad (5.18)$$

where v in this case represents the velocity in x direction, μ represents the dynamic viscosity, k the thermal conductivity and C_ρ , $C_{\rho v}$ and $C_{\rho E}$ represent the integration constants corresponding to the conservation of mass, momentum and energy. By taking the limits far away from the shock, i.e. $x \rightarrow \pm\infty$, the first order derivatives for the velocity and temperature will vanish and equations (5.16)-(5.18) become the Rankine-Hugoniot relations. However, we use the integrated form of the one-dimensional Navier-Stokes equations given in equations (5.16)-(5.18), to find an estimate of the shock thickness [5].

In figure 5.8, a sketch of the velocity distribution in the vicinity of a stationary shock wave is shown. Here, v_u and v_d denote the velocities upstream and downstream of the shock respectively. Hence, in the following, the subscripts u and d indicate the upstream and downstream state with respect to the shock wave. Furthermore, δ_s represents the estimated shock thickness based on a linear approximation of the gradient at the midpoint of the shock. Considering the x -momentum equation given in (5.17) and evaluating $C_{\rho v}$ at $x = -\infty$ so that $C_{\rho v} = p_u + \rho_1 v_u^2$, we obtain an ordinary differential equation that prescribes the behaviour presented in figure 5.8 given by

$$\frac{4}{3}\mu \frac{\partial v}{\partial x} = p - p_u + C_\rho(v - v_u) \quad (5.19)$$

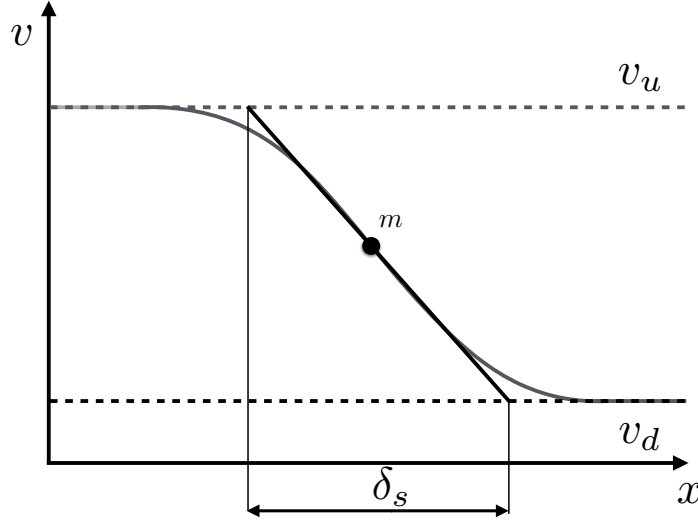


Figure 5.8: Sketch of the velocity distribution in the vicinity of a shock wave with a linear approximation of the shock thickness δ_s .

We approximate the velocity at the midpoint of the shock using

$$v_m = v_u + \frac{v_d - v_u}{2} \quad (5.20)$$

The shock thickness can now be estimated using an approximation for the velocity gradient at the midpoint of the shock

$$\left(\frac{\partial u}{\partial x} \right)_m = \frac{\Delta u}{\delta_s} \quad (5.21)$$

where $\Delta v = v_d - v_u$. In equation (5.19), the expression for the pressure is unknown. By estimating the pressure at the midpoint using a Taylor expansion since the pressure is a function of the density and the entropy we write

$$p = p_u + \left(\frac{\partial p}{\partial \rho} \right)_s (\rho - \rho_u) + \left(\frac{\partial^2 p}{\partial \rho^2} \right)_s \frac{(\rho - \rho_u)^2}{2} + \dots + \left(\frac{\partial p}{\partial s} \right)_\rho (s - s_u) + \dots \quad (5.22)$$

Assuming that this analysis only takes into account the second order terms and considering that $(s - s_u)$ is at least of order $(\rho - \rho_u)^3$ then

$$p - p_u = a_u^2 (\rho - \rho_u) + (\gamma - 1) \frac{a_u^2 (\rho - \rho_u)^2}{\rho_u} \quad (5.23)$$

Using the Rankine-Huginiot relations and the fact that $\rho_m v_m = \rho_u v_u$ we write the

density at the midpoint as

$$\rho_m - \rho_u = -\frac{\rho_u \Delta v}{2v_u + \Delta v} \quad (5.24)$$

and the pressure at the midpoint in terms of Δu as

$$p_m - p_u = -\frac{\rho_u a_u^2}{2v_u} \Delta v + \rho_u a_u^2 \frac{\gamma + 1}{2} \left(\frac{\Delta v}{2v_u} \right)^2 + \mathcal{O}(\Delta v^3) \quad (5.25)$$

Using expressions (5.21), (5.24) and (5.25), evaluating the one-dimensional momentum equation (5.17) at the midpoint of the shock wave and by considering that for the upstream Mach number holds that $M_u > 1$, we can estimate the shock width

$$\delta_s = \frac{8\mu}{3\rho_u a_u (M_u - 1)} \quad (5.26)$$

Considering supersonic flow and writing the properties in SI units, we know that for an ideal gas at sea level we have that $M_u = \mathcal{O}(10^0)$, $a_u = \mathcal{O}(10^2)$, $\rho_u = \mathcal{O}(10^0)$, and $\mu = \mathcal{O}(10^{-5})$. Hence, the order of the shock thickness will be $\delta_s = \mathcal{O}(10^{-7})$. Expression (5.26) illustrates that the shock thickness decreases when the upstream Mach number increases and vice versa.

Essentially, shock waves are anisotropic flow features and it is therefore preferred to use a mesh adaptation strategy that is anisotropic as well. In this way, the elements are allowed to be stretched in a certain direction and they are aligned with the flow features. Usually, the elements that cover the shock waves are split into smaller elements to increase the resolution in the vicinity of the shock so that the strong gradients are represented more accurately. However, this usually increases the computational cost. To alleviate this, we introduce a thermo-elastic r -adaptation strategy that shrinks the elements in the vicinity of shock waves to improve local accuracy. In this way we intend to improve the representation of shock waves without increasing computational cost as shown in figure 5.9.

5.3.1 Formulation of the thermal stress term

The idea is to apply pseudo-thermal stresses to contract the elements in the vicinity of a shock wave. The discontinuity sensor (3.37) normally used for shock capturing, is used here as an additional temperature term from which the pseudo-thermal stress term is evaluated. This mesh deformation technique allows for anisotropic adaptation since the elements will be contracted in the direction perpendicular to the shock wave.

However, from equation (3.37) we know that S_e is piecewise constant within

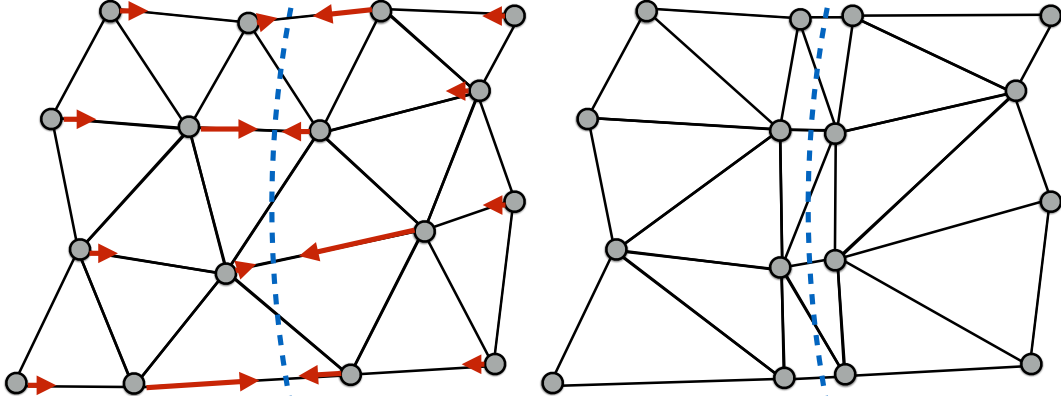


Figure 5.9: A sketched example of r -adaptation for shock waves where the original mesh with the desired displacement field $\mathbf{d} = \{d_1, \dots, d_N\}^t$ is given on the left and the r -adapted mesh on the right and the shock wave is represented by the dashed line.

the computational domain as shown for example in figure 3.8(b). As a result, the gradient between the elements is going to be very large at the interfaces and zero within each element. This causes problems once the divergence of such a piecewise constant temperature field is taken. Therefore, we seek a smooth temperature distribution. To compute a smooth distribution for T_p , we use a similar analogy as the artificial diffusion model introduced by Barter and Darmofal [7]. The following partial differential equation prescribes the diffusion of S_e and is solved along with the governing equations in order to obtain a smooth pseudo-temperature distribution.

$$\frac{\partial T_p}{\partial t} = \sum_{i=1}^2 \frac{\partial^2 T_p}{\partial x_i^2} + \mu_a(S_e) - T_p; \quad T_p \in \Omega \quad (5.27)$$

Here, μ_a is the same as presented in equation (3.38). Generally, we consider a zero normal gradient for the pseudo-temperature on each type of boundary. Hence,

$$\sum_{i=1}^2 \frac{\partial T_p}{\partial x_i} n_i = 0; \quad T_p \in \Gamma \quad (5.28)$$

Equation (5.27) is a simplified version of the smooth artificial viscosity model that was introduced by Barter and Darmofal [7]. In the thermo-elastic formulation discussed in this chapter, we substitute the solution that follows from equation (5.27) into the isotropic stress tensor formulation given in equation (5.8).

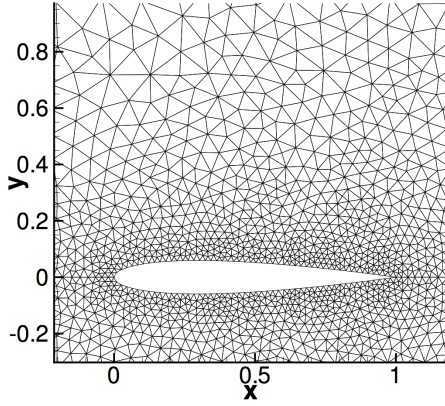
One of the drawbacks of the model introduced in equation (5.27) is that it makes the problem significantly more stiff. Hence, to calculate the smooth pseudo-temperature distribution, we first calculate the steady-state solution to the governing equations using the piecewise constant sensor distribution which is required

anyway to compute the artificial diffusion term. Once a steady-state is obtained, the computation is restarted and we use equation (5.27) to determine the smooth temperature field. This is computationally cheaper than running the simulation with the additional equation for T_p from the start. The displacement of the nodes is then calculated based on the smooth pseudo-temperature. This application of r -adaptation is illustrated using two numerical examples: a transonic flow ($Ma = 0.8$) past a NACA 0012 aerofoil at an angle of incidence $\alpha = 1.25^\circ$ and a two-dimensional supersonic inviscid flow ($Ma = 3.0$) past a forward facing step.

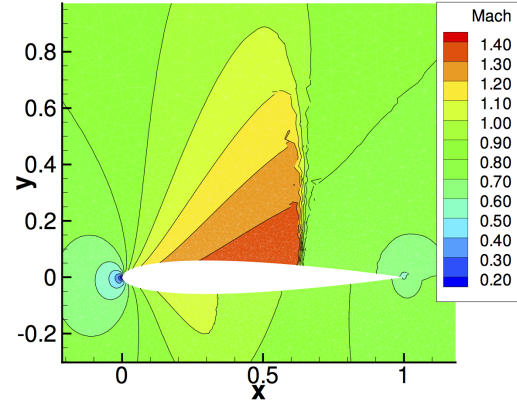
5.3.2 Two-dimensional transonic inviscid flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).

The performance of the proposed r -adaptation strategy using the thermo-elastic analogy is investigated using the well-known test case of transonic inviscid flow ($M = 0.8$) past a NACA 0012 aerofoil at an angle of attack of 1.25° . The domain around the NACA 0012 profile is discretised using an unstructured mesh of 6470 triangular elements. This is the same test case as the one presented in section 3.6.2. Hence, we obtain a strong shock on the suction side at approximately 60% of the chord and a weak shock on the pressure side at approximately 35% of the chord.

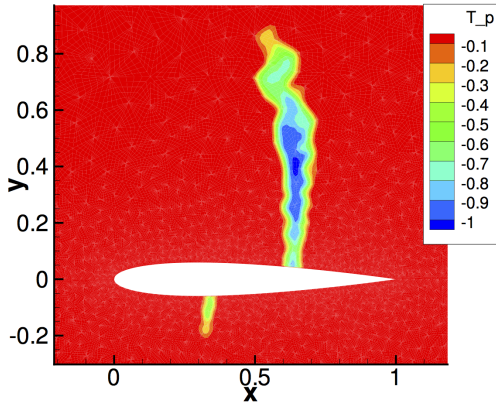
Initially, a flow solution is computed using a uniform polynomial order of $P = 2$. The original mesh and the flow solution obtained using $P = 2$ are shown in figures 5.10(a) and 5.10(b) respectively. The smooth pseudo-temperature distribution is calculated using equation (5.27) and is shown in figure 5.10(c). This pseudo-temperature field is used to contract the elements that cover the shocks. Based on this pseudo-temperature, we calculate the displacement field of the degrees of freedom according to the thermo-elastic equations. In figure 5.10(d), the streamlines are plotted which indicate how the nodes are going to move. This figure illustrates that the elements that cover the strong shock on the suction side and the elements that cover the weak shock on the pressure side are going to contract. After r -adaptation is applied, we determine the final mesh which is shown in figure 5.10(e). The solution that was obtained on the initial coarse mesh at $P = 2$ is then projected onto the r -adapted mesh and used as an initial condition. This initial condition is used to march the computation further in time so that a new steady state solution is obtained using the r -adapted mesh. This refined flow solution is given in figure 5.10(f). By comparing figure 5.10(b) with 5.10(f), we conclude that particularly the representation of the strong shock on the suction side is significantly improved. The r -adaptation procedure for this particular case takes about three to five minutes on four CPU's on a personal laptop. This is negligible compared to the time it takes to solve the governing equations, which was in the order



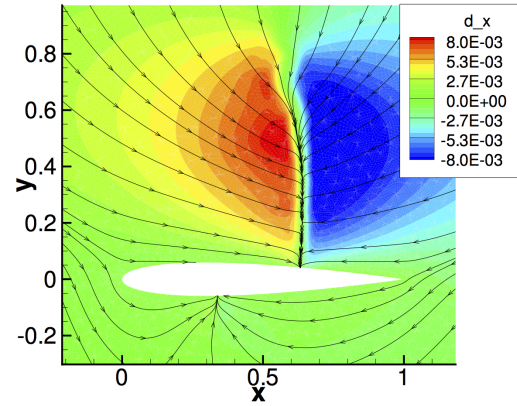
(a) Mach contours obtained using the initial mesh with a uniform $p = 2$.



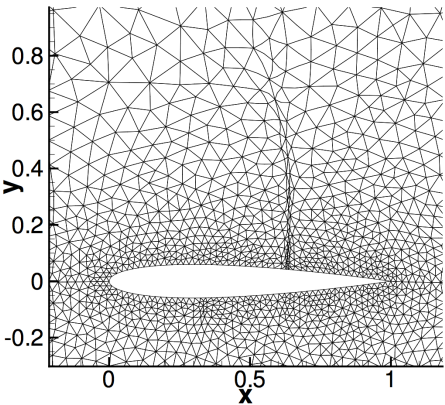
(b) Mach contours obtained using the initial mesh with a uniform $p = 2$.



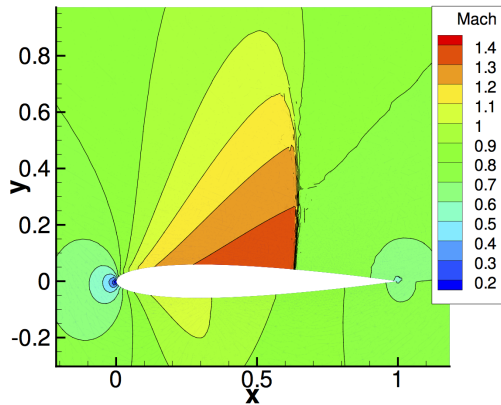
(c) T_p contours based on the steady-state sensor solution.



(d) d_x contours plotted with the streamlines that indicate how the nodes will move.



(e) Mach contours obtained using the r -refined mesh with a uniform $p = 2$.

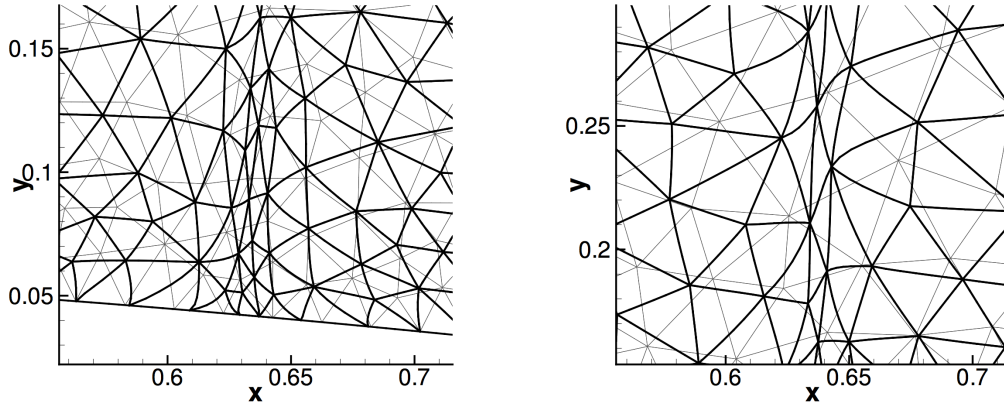


(f) Mach contours obtained using the r -refined mesh with a uniform $p = 2$.

Figure 5.10: Visualisation of the r -adaptation procedure for inviscid transonic flow past a NACA 0012 aerofoil ($Ma = 0.8$, $\alpha = 1.25^\circ$).

of a few hours since a large mesh was used.

For optimal performance of the r -adaptation strategy we need to allow the nodes to move freely along the wall boundary edge. Figure 5.11(a) shows a detailed comparison of the original mesh and the r -adapted mesh at the root of the strong shock on the suction side. Figure 5.11(a) illustrates that the nodes at the wall

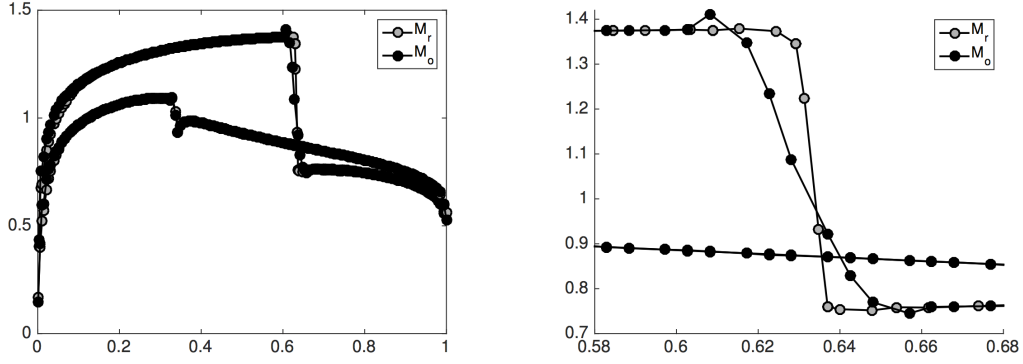


(a) Detail of the comparison between the original mesh (thin grey lines) and the r -adapted mesh (thick black lines) at the root of the strong shock at the suction side. (b) Detail of the comparison between the original mesh (thin grey lines) and the mesh after r -adaptation is applied (thick black lines) away from the geometry near the shock at the suction side.

Figure 5.11: Detailed comparison between the original mesh and the mesh after r -adaptation at the suction side of the NACA 0012.

moved from both sides towards the origin of the the strong shock wave. The elements at the root in the r -adapted mesh are approximately two times smaller than in the original mesh. Figure 5.11(b) illustrates an even further reduction in size of the elements a bit further away from the wall. The original size of these elements was bigger compared to the element size near the wall and therefore a more improved resolution was obtained. Both figures 5.11(a) and 5.11(b) demonstrate the anisotropic refinement very well. The elements are clearly stretched in the direction of the shock.

As a final comparison we plot the Mach solution at the wall for both the original (Ma_0) and r -adapted mesh (Ma_r). We used a uniform $P = 2$ to calculate both solutions. The Mach solutions are given in figure 5.12. The detailed picture of the Mach solution at the strong shock on the suction side, given in figure 5.12(b), shows particularly well the improvement that is obtained using the r -adapted mesh. Both cases use the same amount of artificial diffusion. The solution for Ma_0 shows small wiggles both upstream and downstream of the shock which are completely removed for Ma_r . Furthermore, as expected, a significantly stronger gradient at the shock



(a) Comparison of the Mach solution at wall obtained using the original and r -adapted mesh. (b) Detailed comparison of the Mach solution at the root of the strong shock on the suction side obtained using the original and r -adapted mesh.

Figure 5.12: Comparison between the original mesh (black dots) and the r -adapted mesh of the Mach distribution.

is obtained using the r -adapted mesh. This corresponds with the shock thickness estimation.

The high curvature of the elements can cause aliasing since the Jacobian of the mapping is included as a polynomial function in the integral over the elemental area. Hence, it was noticed that for the highly curved elements, for example at the top of the strong shock on the suction side, de-aliasing techniques should be applied to prevent numerical instabilities.

The slide boundary conditions work well up to point that the surrounding elements become invalid. This is prevented by implementing a combination of the previously introduced mesh validity control strategy using one of mesh quality indicators proposed section 5.2. The mesh quality control parameter is used to increase the local stiffness coefficient of the element preventing it to become invalid. However, this method requires an appropriate stopping criteria which has not been incorporated in this study.

5.3.3 Two-dimensional supersonic inviscid flow past a forward facing step ($Ma = 3.0$).

To illustrate the r -adaptation procedure, we compute the inviscid supersonic flow ($Ma = 3.0$) past a forward facing step which has been thoroughly investigated by Woodward and Colella [96]. For this problem, we consider the compressible Euler equations and we neglect the effects of viscosity. In the paper of Woodward and Colella [96], the height of the step is taken to be 20% of the height of the inlet

so $h = 0.2H$ where in this case $H = 2.0$. The conditions at the inlet are $\rho_\infty = 1.4$, $p_\infty = 1$, $u_\infty = 3.0$, $v_\infty = 0.0$. Woodward and Colella [96] presented results at $t = 4.0$ since the steady-state solution does not show any interesting features other than the bow shock that is present upstream of the step. The results presented at $t = 4.0$ serve as reference data for current CFD code developers. However, in our case, we are interested in a configuration that results in shock reflections in the steady solution. In this way we can study the suitability of r -adaptation when shock reflections and interactions occur. Therefore we have modified the height of the step from $h = 0.2H$ to $h = 0.175H$ as shown in [69]. A detail of the initial mesh is shown in figure 5.13(a).

Figure 5.13 illustrates the three main steps of the r -adaptation procedure. We first obtain a low-order steady-state solution at $P = 2$, shown in figure 5.13(b). This figure shows that there is a strong bow shock upstream of the step. Furthermore, a strong normal shock forms at the upper wall where the bow shock reflects back. At the corner of the forward facing step, an expansion wave interacts with the reflected shock wave forming another normal shock at the lower wall. We will use r -adaptation to accurately capture these flow features.

We use the discontinuity sensor to determine a smooth pseudo-temperature distribution which is shown in figure 5.13(c). Figure 5.13(c) shows that there is a significant variation in temperature because the bow shock and the normal shock at the upper wall are relatively strong compared to the weak reflection shock. Based on the pseudo-temperature distribution, we obtain a displacement field of the nodes by solving the previously introduced thermo-elastic problem. An example of the horizontal displacement component, d_x is shown in figure 5.13(d). The streamlines indicate the direction in which the degrees of freedom are moving. Figure 5.13(d) illustrates that the displacement field seems to be balanced at the top and elements are attracted from both sides to refine the normal shock at the upper wall.

Figure 5.13(e) shows the resulting mesh after r -adaptation is applied. Furthermore, figure 5.13(f) depicts the steady-state Mach contours that are obtained using the new mesh. Comparing figures 5.13(b) and 5.13(f) indicates that the representation of the bow shock and the reflecting shock waves further downstream is improved considerably.

It has been noticed that the geometry of the step and the shock pattern further downstream have a significant influence on how well the r -adaptation procedure follows the curvature of the bow shock upstream of the step. This is why two additional iterations were required to obtain the final mesh that was shown in figure 5.13(e). Initially, a pseudo-temperature field is calculated based on the sensor distribution obtained using the initial coarse mesh. Based on this pseudo-temperature field, we calculate the deformation field. The initial pseudo-temperature field to-

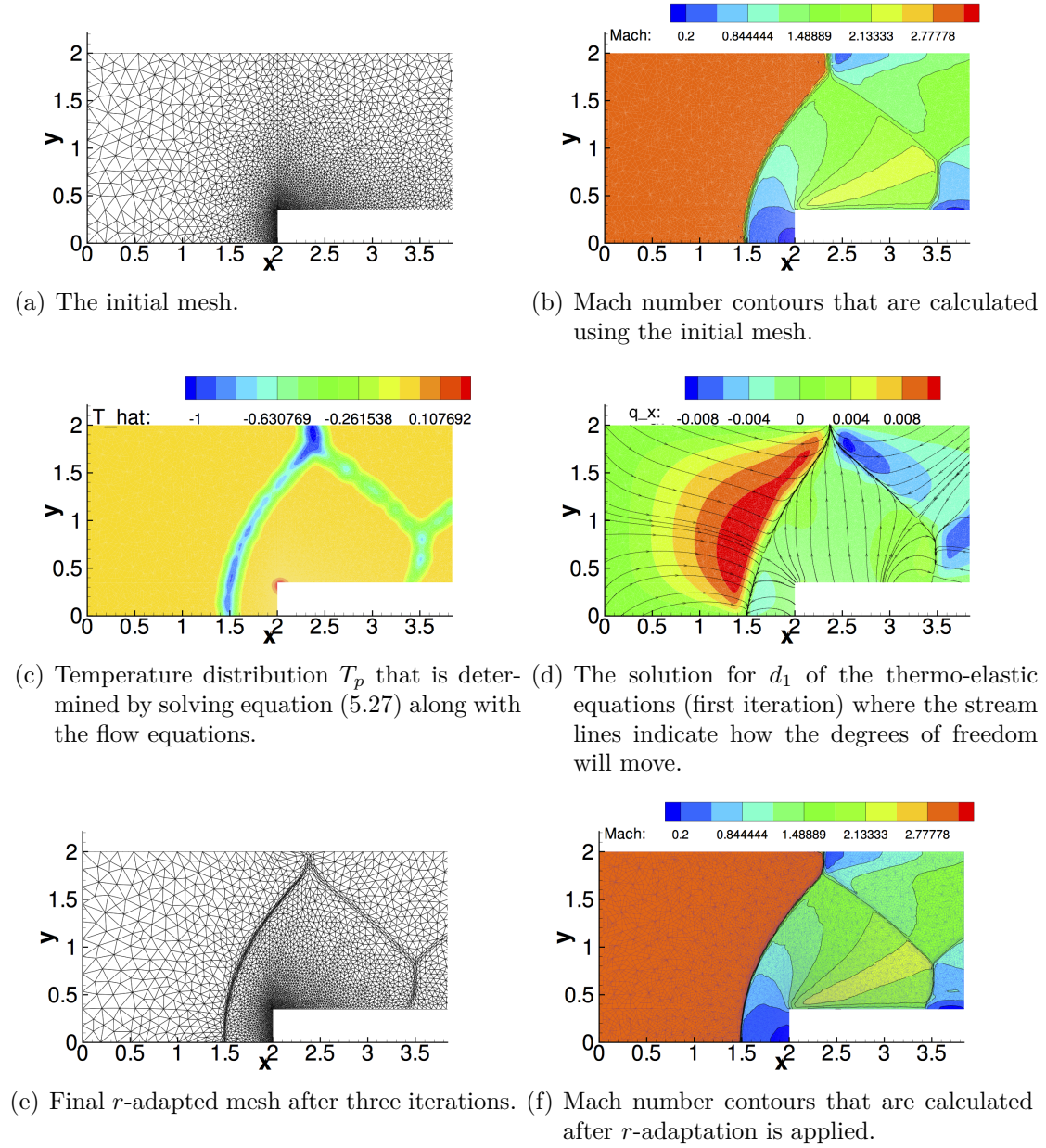


Figure 5.13: General overview of the required steps to perform r -adaptation for supersonic inviscid flow past a forward facing step ($Ma = 3.0$).

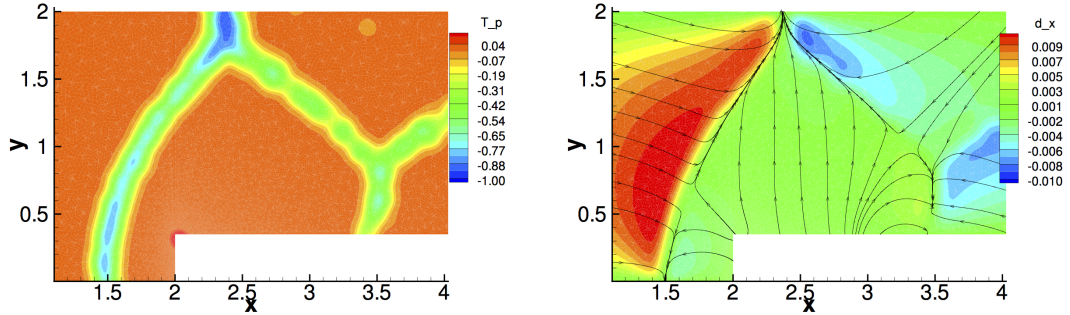
gether with the corresponding d_x contours are given in figures 5.14(a) and 5.14(b). From figure 5.14(b) can be seen that there is a strong forcing in positive x -direction. However the forcing in opposite direction downstream of the bow shock is not as strong. This causes an inaccurate overlap of the refinement and the location of the shock. This is why two additional r -adaptation iterations were performed to ensure that the refinement corresponds well with the location of the shock. Figure 5.14(c) depicts a more narrow pseudo-temperature distribution in the vicinity of the shock meaning that the location of the shock is determined more accurately. Figure 5.14(e) eventually shows that the larger elements just upstream of the shock are *cooled down* and contracted. However, in the upper part of the bow shock, as for the weaker shock just downstream of the bow shock, it is shown that the pseudo-temperature approaches zero and no extra refinement is required. Figure 5.14(f) illustrates that the forcing aligns the refinement with the physical location of the bow shock.

To get more insight on the improvement, we compare the Mach distributions along three different y positions, $y = 0.5$, $y = 0.75$ and $y = 1.0$ which are shown in figure 5.15. In figures 5.15(a) and 5.15(b) the Mach number is plotted near the bow shock and near the normal shock on the lower wall downstream of the bow shock respectively. Figure 5.15(a) shows the influence of the bow shock on each location. Figure 5.15(b) depicts the Mach number distribution further downstream of the bowshock at the normal shock. For $y = 0.5$ and $y = 0.75$ we see again a strong shock which corresponds to the normal shock located at $x \approx 3.5$ while at $y = 1.0$ we see two weak shocks which correspond with the shock reflections induced by the normal shock that occurs at approximately $(x, y) = (3.5, 0.8)$.

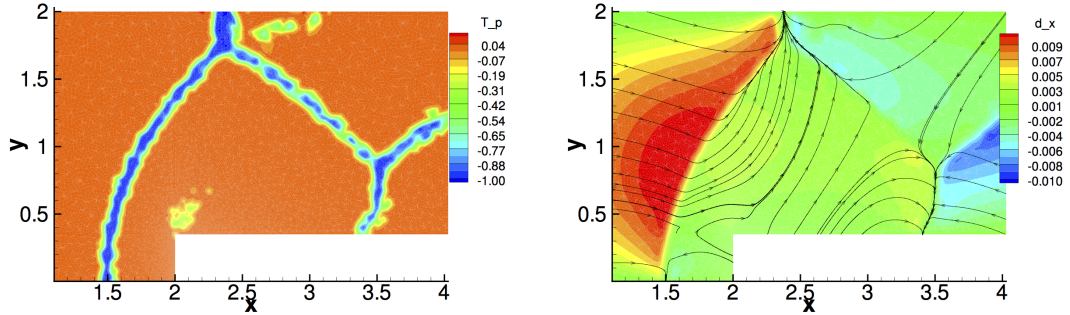
Figures 5.15(c) and 5.15(d) depict the error in Mach number that is determined using

$$\epsilon_{Ma} = \frac{|Ma_o - Ma_r|}{Ma_o} = \frac{|\Delta Ma|}{Ma_o} \quad (5.29)$$

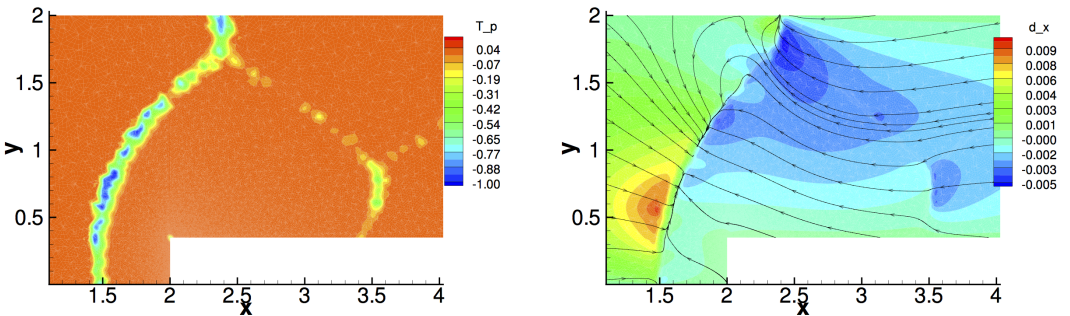
where Ma_o represent the Mach solution calculated using the original mesh and Ma_r the Mach solution calculated using the final mesh after r -adaptation is applied. At the bow shock $\Delta Ma = 0.48$ and $\Delta Ma = 0.6$ for $y = 0.5$ and $y = 0.75$ respectively while for $y = 1.0$ we find that $\Delta Ma = 0.2$. Naturally, ΔMa depends on the difference in Mach number upstream and downstream of the shock. Hence, it is expected that a large ΔMa is obtained for stronger shocks. Both figures 5.15(c) and 5.15(d) show that ϵ_{Ma} is largest when the difference between v_u and v_d is biggest as well. It is assumed that the discretisation is generally much coarser than the anticipated shock thickness. Hence, it is expected to have the biggest gain in accuracy when the difference between the Mach number upstream and downstream of the shock is large.



(a) Example of the contours for T_p determined using the initial mesh. (b) Example of the contours for d_x determined using the initial mesh.

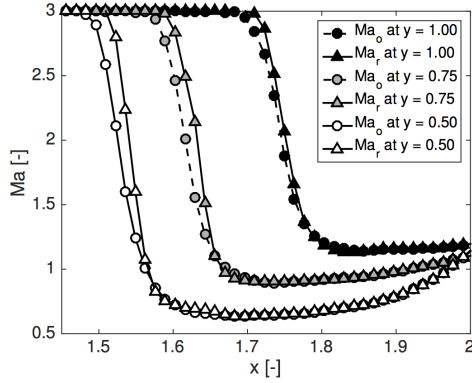


(c) Example of the contours for T_p determined after the first iteration. (d) Example of the contours for d_x determined after the first iteration.

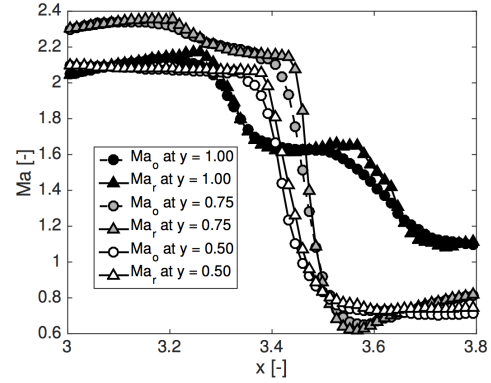


(e) Example of the contours for T_p determined after the second iteration. (f) Example of the contours for d_x determined after the second iteration.

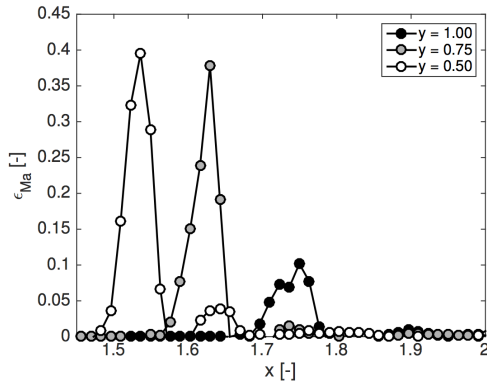
Figure 5.14: The contours for the pseudo-temperature (left) and the resulting contours for d_x (right) obtained using the original mesh and the two r -adaptation iterations for supersonic inviscid flow past a forward facing step ($Ma = 3.0$).



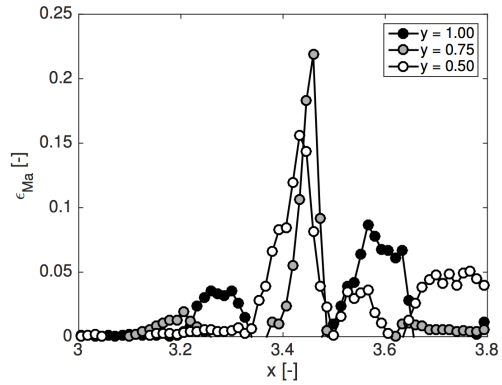
(a) Comparison of the Mach number between the original and the r -adapted mesh near the bow shock.



(b) Comparison of the Mach number between the original (circles) and the r -adapted (triangles) mesh near the normal shock.



(c) Difference in Mach number between the original (circles) and the r -adapted (triangles) mesh near the bow shock.



(d) Difference in Mach number between the original and the r -adapted mesh near the normal shock.

Figure 5.15: Comparison between the original and the r -adapted mesh of the Mach number at two locations: near the bow shock ($1.4 \leq x \leq 2$) and near the normal shock at the wall ($3 \leq x \leq 3.8$).

Chapter 6

Conclusions and recommendations for future work

The conclusions and recommendations discussed in this chapter are split up into two parts: First the conclusions and recommendations are given related to adjoint-based p -adaptation. Secondly the conclusions and recommendations regarding the proposed r -adaptation capability for shock waves are given.

6.1 Adjoint-based p -adaptation

The improvement in terms of computational efficiency and cost by using an adjoint-based p -adaptation strategy was investigated. This required the implementation of a continuous adjoint formulation within the framework of the high-order spectral/ hp element library Nektar++. Adjoint methods are currently very popular in the field of low-order and more recently high-order computational methods and are mostly used for flow stability and control, aerodynamic design and mesh adaptation. As a result, several useful references were at hand that served as guidelines during the implementation process of the high-order adjoint code.

However, due to the fact that there is a lack of reference adjoint solutions, an effort has been made to verify the high-order adjoint code. A review of the available literature showed that the discrete adjoint approach could result in inconsistent adjoint solutions, particularly near the boundary, due to the usage of inconsistent flux functions when imposing the boundary conditions weakly. To ensure adjoint consistency, the continuous adjoint approach was chosen to develop the high-order adjoint solver.

6.1.1 Conclusions

A quasi one-dimensional test case of subsonic flow through a converging-diverging nozzle has been considered to verify the implementation of the high-order adjoint solver in Nektar++. For this test case, an analytical solution can be derived following the approach presented by Giles and Pierce [35]. Hence, a comparison can be made between the adjoint solution obtained using Nektar++ with the derived analytical adjoint solution. For this particular case, the literature showed that the continuous adjoint approach provides better results than the discrete adjoint approach particularly near the boundary [23, 62]. The adjoint solution obtained using Nektar++ showed a good correspondence with the analytical solution providing the confidence that the continuous adjoint equations were implemented correctly in the high-order spectral element library Nektar++.

The initial purpose of the adjoint implementation was to investigate the influence of the local discretisation error on the error in aerodynamic force coefficient. Therefore, as an additional verification step, adjoint solutions with aerodynamic force coefficients as objective functionals were compared with other available adjoint codes that are used for the same purpose by partners within the FP7 ANADE framework.

A transonic flow past a wing geometry was considered with the lift coefficient as an objective functional and the adjoint solution obtained using Nektar++ was compared with the verified adjoint solver that is implemented in the DLR/TAU code. The results obtained indicate a good correspondence in terms of shape however the absolute values at the wall are different. This is ascribed to the cusp trailing edge of the wing. Tests have been performed both using the DLR/TAU code and Nektar++ by increasing the spatial resolution and a fixed shift in absolute value was noted in both codes. It is expected that improving the resolution results in an improved estimate of the high gradients related to the local singularity near the trailing edge resulting in an overall shift in absolute value. As a result, it was checked how well Nektar++ and the DLR/TAU code satisfy the theoretical wall boundary condition for the adjoint problem. Nektar++ showed a good correspondence with the theoretical value that lies within the numerical accuracy of the DLR/TAU code.

The comparison between Nektar++ and the analytical solution for the one-dimensional test case and the comparison between Nektar++ and the DLR/TAU code served as a thorough verification of the high-order adjoint code. This provided the confidence that the continuous adjoint formulation was the appropriate choice and was correctly implemented in the high-order spectral/ hp element framework of Nektar++. There is now a verified high-order adjoint tool for compressible

flows available in the Nektar++ framework which potentially can be used for other applications and fields of research like flow stability and control or aerodynamic design for the compressible flow regime.

In this thesis, the implemented high-order adjoint solver has been used to drive a p -adaptation algorithm that aims to approximate target quantities like the aerodynamic force coefficients as accurate as possible. The adjoint-based p -adaptation strategy is illustrated using a series of inviscid and laminar flow cases in two and three dimensions. The results for the two-dimensional inviscid test case illustrate that the goal-based p -adaptive solution requires around 30% of the degrees of freedom and 46% less CPU time per time step to achieve the same order of accuracy as the reference simulation.

The two-dimensional laminar test case also illustrates particularly well the benefits of using the sensitivity information following from the adjoint solution, since the boundary layer region is clearly highlighted in the adjoint solution. For this case, the drag was taken to be the target functional. The adjoint solution clearly identified the region downstream of the leading edge and the boundary layer as areas that mostly influence the drag coefficient. Compared to a local sensor-based adaptation procedure we see that solely these areas are refined instead of also refining the complete wake of the airfoil due to the error that gets convected downstream (see the numerical example shown in section 4.1.1).

For the three-dimensional subsonic inviscid case, we can draw similar conclusions as for the two-dimensional subsonic inviscid and laminar test cases. The error is reduced by two orders of magnitude compared to the $P = 2$ solution using goal-based p -adaptation. In this case approximately 50% fewer integration points were used compared to the reference case ($P = 4$). However, the achieved CPU time reduction of 30% is less compared to the two-dimensional cases. This can be ascribed to the de-aliasing strategy that was chosen by doubling the number of integration points in each spatial direction.

Furthermore, an effort is made to determine the performance of adjoint-based p -adaptation for flows with shocks by considering a transonic flow past a wing geometry. One typically wants to avoid increasing the local approximation order in the vicinity of shocks and the conventional manner to refine is to decrease the element size near the shock. However, the adjoint solution illustrates that the error in aerodynamic force coefficient mainly depends on the accuracy of the solution at the root of the strong shock on the suction side and the importance of the accuracy of the solution reduces considerably when moving away from the surface. It is therefore expected that the discretisation error further away from the geometry does not contribute much to the error in lift coefficient. However, it was noted that particularly the elements at the wall drive the error in aerodynamic force

coefficient and therefore still require size reduction by means of h -adaptation or mesh deformation. Hence, it is recommended to investigate the effects of applying h -adaptation or mesh deformation solely at the root of the shock and apply goal-based p -adaptation for the rest of the computational domain. As long as the solution is sufficiently smooth, goal-based p -adaptation is a suitable strategy to improve an initial coarse solution, both in terms of degrees of freedom and CPU time per time step.

6.1.2 Recommendations

As mentioned before, there is now a verified high-order adjoint tool available as a separate solver that can serve for other purposes than mesh adaptation like flow stability analysis or shape optimisation in the framework of design for example. This would require additional implementation of control and constraint parameters. However, the infrastructure is there now and it becomes relatively straightforward to add additional features within the framework of the solver. Hence, it is strongly recommended to start using the adjoint solver for these purposes as well. The laminar adjoint implementation could be further verified using the investigation of laminar flow past a flat plate for which an analytical adjoint expression can be derived considering that we have a Blasius profile at the wall.

As a crucial additional improvement for the adjoint solver, it is suggested to develop a driver module in Nektar++ that allows the computation of the adjoint for unsteady compressible flow problems. This can be done using a checkpointing procedure as described by Wang et al. [94]. To enable the linearisation of the solution to the governing equations, we either need to store it in memory for each time step or recalculate it from a stored solution at a checkpoint in time using a checkpointing scheme. The latter strategy is often used for engineering problems that require both large memory storage and many time steps, like the ones presented in this thesis, since the combination of these two factors makes the storage of the solution at each time step very impractical. A checkpointing scheme is proposed by Griewank et al. [37] which requires the *a priori* knowledge of the number of time steps. Wang et al. [94] introduced an optimal dynamic checkpointing algorithm for which the number of time steps does not need to be known *a priori* and only uses a fixed number of checkpoints for an arbitrary number of time steps. This dynamic checkpointing strategy is computationally more costly but it requires less storage space to capture the sensitivity fields for unsteady flows. The checkpointing scheme proposed by Wang et al. [94] is a suitable strategy for adjoint-based p -adaptation of unsteady compressible flows considering the scale of the computations that were shown in this thesis.

Finally, two recommendations regarding p -adaptation in general. Incorporating variable approximation orders in each direction within each element (anisotropic p -adaptation) would be very useful when one is interested in refining for anisotropic flow features like boundary layers or shock waves. Furthermore, the issue of appropriate load-balancing when running the p -adaptive algorithm in parallel is an issue that still requires attention. At the moment, the partitioning of the mesh happens on an element-wise basis which may result in an unequal division of the total number of degrees of freedom when the simulation is run in parallel.

6.2 Mesh deformation and r -adaptation using a thermo-elastic analogy

The proposed goal-based p -adaptation strategy relied mainly on tools that were available in the literature and the main contribution there can be ascribed to the efforts made in terms of implementation and the fact that there is now a verified high-order adjoint solver for compressible flows available. However next to that, a more conceptual contribution has been discussed in this thesis.

6.2.1 Conclusions

A proof-of-concept implementation shows an improved robustness of the classical mesh deformation strategy using an elastic formulation. High-order methods require elements to be curved particularly near a boundary edge of the computational domain if we consider for example external flow problems. This curvature can cause self intersecting elements. This problem is prevented by moving the surrounding nodes using a mesh deformation strategy. The linear elastic equations are often used to determine the redistribution of the degrees of freedom based on a boundary displacement. In the classical approach, the mesh is considered to be an elastic solid and the boundary displacement as an external force acting on the elastic solid.

Additional thermal stress terms are incorporated to locally *cool down* (contract) or *heat up* (expand) elements to ensure mesh validity once an element tends to become invalid. It is shown using both two- and three-dimensional examples that incorporating additional thermal stresses that are based on a local mesh quality indicator results in an improved robustness of the classical linear elastic mesh deformation technique. Both isotropic and anisotropic thermal stress terms were considered. Both showed similar improvements in robustness however for the anisotropic formulation, the improvement is independent from the polynomial order that is used. In this work, a linear elastic approach was adopted where the domain was

deformed using small substeps. When adopting a non-linear analogy, the number of substeps can be aligned with the number of iterations that is required to solve the non-linear system using a Newton-Raphson method for example. Therefore, future work could focus on using a non-linear elastic analogy which may prevent having to tune the thermal coefficients that are currently used depending on which thermal stress formulation is chosen.

This idea of using a thermo-elastic formulation for mesh deformation was extended in this thesis by applying the thermal forcing to adapt the mesh for shock waves. As mentioned before, p -adaptation is suitable for sufficiently smooth solutions but compressible flows are generally characterised by shock waves. To simulate flows in the transonic and supersonic flow regime, it is usually preferred to decrease the element size near the shock rather than increasing the polynomial order to avoid numerical oscillations. Instead of adding extra elements in the vicinity of the shock, it is proposed to contract the elements that cover the shock wave by *cooling them down* using the thermo-elastic analogy. The pseudo-temperature model that is proposed in this case relies on a smoothed shock capturing sensor distribution. The proposed pseudo-temperature model results in an anisotropic deformation which is particularly suitable considering the anisotropic nature of shock waves. We refer to this technique as r -adaptation (adaptation through redistribution). An additional advantage of this technique is that the connectivity is preserved which means that no repartitioning of the mesh is required.

The two-dimensional examples shown in the last chapter of this thesis illustrate the proof-of-concept implementation and suitability of r -adaptation using a thermo-elastic analogy for shock waves. Improvements are obtained in shock representation while maintaining the same number of degrees of freedom. Both internal and external compressible flows were considered. The r -adaptation strategy performed particularly well for the external flow test case. For the internal flow test case it was noticed that the geometry (sharp corners) or asymmetric flow features have a significant influence of the number of r -adaptation iterations that are required. However, running the elastic solver to obtain the new r -adapted mesh is computationally cheap and can be run on a personal laptop within three to five minutes depending on the size of the mesh.

6.2.2 Recommendations

Future work should focus on the formulation of a suitable stopping criteria that indicates that a refined flow solution should be obtained once again and that based on this new flow solution, an additional r -adaptation iteration needs to be performed. Potentially this can be done by including a driver module in Nektar++ that allows

the user to calculate intermediate meshes and compute corresponding intermediate flow solutions while running the r -adaptation procedure. Hence, general automation and improvement in robustness of the r -adaptation strategy are key elements for future work.

For both internal and external flow problems it is important that the nodes are allowed to slide along a wall boundary in order to improve the accuracy near the wall. Hence, for the numerical examples discussed in this thesis, a slip condition is imposed at the wall. However, it has been noticed that this may result in highly deformed elements at the wall due to the anisotropic nature of shock waves. Similar behaviour was noticed by McRae [66]. Therefore, it is proposed that future work should concentrate on the investigation of appropriate stopping criteria and corresponding actions that can prevent these elements from becoming invalid. Solutions to this problem can be the combination of the proposed mesh validity control strategy with r -adaptation or increasing the stiffness coefficient of elements based on a local mesh quality indicator.

It is also suggested to combine r - and p -adaptation for transonic and supersonic flows. Initially, r -adaptation could be combined with sensor-based p -adaptation. Potentially, the r -adaptation could also be combined with goal-oriented p -adaptation. Furthermore, it could be of interest to also investigate the inclusion of h -adaptation since the improvement due to r -adaptation is limited. This limitation can be seen for the case of supersonic flow past a forward facing step that was presented in section 5.3.3. For this test case, the accuracy of the solution near the shock reflections upstream of the bow shock is limited. At these locations one would typically like to apply h -adaptation instead. However, h -adaptation is currently not available in Nektar++ and requires further implementation.

Next to that, it is suggested to investigate unsteady r -adaptation. There is no need to repartition the mesh since the connectivity of the nodes is maintained. This avoids the use of complicated remeshing algorithms which often requires cumbersome human intervention during the unsteady adaptation process.

Finally, it is recommended to investigate the application of r -adaptation for shock waves using quad- or hex-dominant meshes should be investigated. These types of elements are preferably used to capture large gradients that are present in the boundary layer for example. Similar benefits are expected once quadrilateral or hexahedral shaped elements are aligned with shock waves [16, 40].

Bibliography

- [1] F. Alauzet and A. Loseille. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Computer-Aided Design*, 72:13–39, 2016.
- [2] J. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill Education, 2010.
- [3] W.K. Anderson and V. Venkatakrishnan. Aerodynamic design optimisation on unstructured grids with a continuous adjoint formulation. *AIAA Paper 97-0643*, 1997.
- [4] I. Babuska and M. Suri. The p and hp versions of the finite element method, basic principles and properties. *SIAM Review*, 36(4):578–632, 2012.
- [5] P.G. Bakker. Gas dynamics lecture notes. Delft University of Technology, 2000.
- [6] A. Balan, M. Woopen, and G. May. hp -adaptivity on anisotropic meshes for hybridized discontinuous Galerkin scheme. In *22nd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2015-2606*, Dallas, Texas, USA, 2015.
- [7] G.E. Barter and D.L. Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part 1. Formulation. *Journal of Computational Physics*, 229:1810–1827, 2010.
- [8] 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*, 131:267–279, 1996.
- [9] C.E. Baumann and T.J. Oden. A discontinuous hp finite element method for convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175:311–341, 1999.
- [10] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West Journal of Numerical Mathematics*, 4:237–264, 1996.

- [11] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.
- [12] C. Biotto. *A discontinuous Galerkin method for the solution of compressible flows*. PhD thesis, Imperial College London, 2011.
- [13] A. Brandt and O. Livne. *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*. Society for Industrial and Applied Mathematics, 2011.
- [14] A. Burbeau and P. Sagaut. A dynamic p -adaptive discontinuous Galerkin method for viscous flow with shocks. *Computers and Fluids*, 34:401–417, 2005.
- [15] C. Castro, C. Lozano, C. Palacios, and E. Zuazua. A systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. *AIAA Journal*, 45(9):2125–2139, 2007.
- [16] M. Ceze and S.M. Murman. Robust quad-dominant meshing with alignment using l_p -CVT. Submitted for Scitech, Grapevine, Texas, US, 2017.
- [17] B. Cockburn, S. Hou, and C.W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV. *Journal of Computational Physics*, 54(1):545–581, 1989.
- [18] B. Cockburn, S. Hou, and C.W. Shu. TVB Runge-Kutta local projection discontinuous galerkin finite element method for conservation laws III: One dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.
- [19] B. Cockburn and C.W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.
- [20] B. Cockburn and C.W. Shu. The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.
- [21] B. Cockburn and C.W. Shu. The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.
- [22] L. Demkowicz, W. Rachowicz, and Ph. Devloo. A fully automatic hp -adaptivity. *Journal of Scientific Computing*, 17(1):117–142, 2002.

- [23] G.F. Duivesteijn, H. Bijl, and B. Koren. On the adjoint solution of the quasi 1D Euler equations: The effect of boundary conditions and the numerical flux function. *International Journal for Numerical Methods in Fluids*, 47:987–993, 2005.
- [24] R.P. Dwight. Robust mesh deformation using the linear elasticity equations. In H. Deconinck and E. Dick, editors, *Computational Fluid Dynamics 2006*, pages 401–406. Springer, 2009.
- [25] T. D. Economon, F. Palacios, and J.J. Alonso. Unsteady aerodynamic design on unstructured meshes with sliding interfaces. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2013-0632*, Dallas, Texas, USA, 2013.
- [26] D. Ekelschot, D. Moxey, S.J. Sherwin, and J. Peiró. A p -adaptation method for compressible flow problems using a goal-based error indicator. *Computers and Structures*, 2016. doi:10.1016/j.compstruc.2016.03.004.
- [27] K.J. Fidkowski. High-order output-based adaptive methods for steady and unsteady aerodynamics. VKI Lecture Notes, 2013.
- [28] 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.
- [29] F. Fraysse, J. de Vicente, and E. Valero. The estimation of truncation error by τ -estimation revisited. *Journal of Computational Physics*, 231(9):3457 – 3482, 2012.
- [30] I. M. Gelfand and S. V. Fomin. *Calculus of variations*. Dover publications, 1963.
- [31] P.L. George, F. Hecht, and M.G. Vallet. Creation of internal points in voronoi’s type method. control adaptation. *Advances in Engineering Software and Workstations*, 13(5):303 – 312, 1991.
- [32] F. Giannetti and P. Luchini. Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics*, 581:167–197, 2007.
- [33] M. B. Giles and N. A. Pierce. Improved lift and drag estimates using adjoint euler equations. *AIAA Paper 99-3293*, 1999.
- [34] M.B. Giles and N.A. Pierce. Adjoint equations in CFD: Duality, boundary conditions and solution behaviour. *AIAA Paper 97-1950*, 1997.

- [35] M.B. Giles and N.A. Pierce. Analytic adjoint solutions for the quasi-one-dimensional Euler equations. *Journal of Fluid Mechanics*, 426:327–345, 2001.
- [36] G. Giorgiani, S. Fernández-Méndez, and A. Huerta. Goal-oriented *hp*-adaptivity for elliptic problems. *International Journal for Numerical Methods in Fluids*, 72(1):1244–1262, 2013.
- [37] A. Griewank and A. Walther. Revolve: An implementation of check-pointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26:19–45, 2000.
- [38] G. Hansen, A. Zardecki, D. Greening, and R. Bos. A finite element method for unstructured grid smoothing. *Journal of Computational Physics*, 194:611–631, 2004.
- [39] G. Hansen, A. Zardecki, D. Greening, and R. Bos. A finite element method for three-dimensional unstructured grid smoothing. *Journal of Computational Physics*, 202:281–297, 2005.
- [40] M.J. Harris and N. Qin. Using the medial axis to represent flow features for feature-aligned unstructured quad-dominant mesh generation. *Computers and Fluids*, 102:1–14, 2014.
- [41] R. Hartmann. Hyperbolic problems: theory, numerics, applications. In *Adaptive High-Order Methods in Computational Fluid Dynamics*, volume 141 of *International Series of Numerical Mathematics*, pages 67–94. 2001.
- [42] R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 51:1131–1156, 2005.
- [43] R. Hartmann. Derivation of an adjoint consistent discontinuous Galerkin discretization of the compressible Euler equations. In *Proceedings of the BAIL 2006 conference*, Göttingen, Germany, 2006.
- [44] R. Hartmann, J. Held, T. Leicht, and F. Prill. Discontinuous Galerkin methods for computational aerodynamics - 3D adaptive flow simulation with the DLR PADGE code. *Aerospace Science and Technology*, 14:512–519, 2010.
- [45] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 24:979–1004, 2002.

- [46] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183(2):508–532, 2002.
- [47] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.
- [48] A. Jameson. Optimum aerodynamic design using CFD and control theory. In *AIAA 12th Computational Fluid Dynamics Conference, AIAA Paper 97-0643*, San Diego, California, USA, June 1995.
- [49] A. Jameson. Aerodynamic shape optimization using the adjoint method. In *VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics*, Rhode St Genese, Belgium, 2003.
- [50] S.L. Karman. Unstructured viscous layer insertion using linear-elastic smoothing. *AIAA Journal*, 45(1):168–180, January 2007.
- [51] G. E. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford Science Publications, 2 edition, 2005.
- [52] P.M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. *Engineering with Computers*, 15(3):263–268, 1999.
- [53] N. Kroll, T. Leicht, C. Hirsch, F. Bassi, C. Johnston, K. Sorensen, and K. Hillewaert. Results and conclusions of the European project IDIHOM on high-order methods for industrial aerodynamic applications. In *53rd AIAA Aerospace Sciences Meeting, AIAA Paper 2015-0293*, Kissimmee, Florida, USA, 2015.
- [54] 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.
- [55] T. Leicht and R. Hartmann. Error estimation and *hp*-adaptive mesh refinement for discontinuous Galerkin methods. In Z.J. Wang, editor, *Adaptive High-Order Methods in Computational Fluid Dynamics*, volume 2 of *Advances in Computational Fluid Dynamics*, chapter 3, pages 67–94. World Science Books, 2011.
- [56] L.Y. Li, Y. Allaneau, and A. Jameson. Comparison of *h*- and *p*-adaptations for spectral difference methods. In *40th AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2010-4435*, Chicago, Illinois, USA, 2010.

- [57] I. Lomtev and G.E. Karniadakis. A discontinuous Galerkin method for the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 29:587–603, 1999.
- [58] A. Loseille and F. Alauzet. Continuous Mesh Model and Well-Posed Continuous Interpolation Error Estimation. Research Report RR-6846, INRIA, 2009.
- [59] A. Loseille and F. Alauzet. Continuous mesh framework part 1: Well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, 2011.
- [60] A. Loseille and F. Alauzet. Continuous mesh framework part 2: Validations and applications. *SIAM Journal on Numerical Analysis*, 49(1):61–86, 2011.
- [61] A. Loseille, A. Dervieux, and F. Alauzet. A 3D goal-oriented anisotropic mesh adaptation applied to inviscid flows in aeronautics. In *48th AIAA aerospace sciences meeting including the new horizons forum and aerospace Exposition, AIAA Paper 2010-1067*, Orlando Florida, 2010.
- [62] C. Lozano and J. Ponsin. Remarks on the numerical solution of the adjoint quasi-one-dimensional euler equations. *International Journal for Numerical Methods in Fluids*, 69(5):966–982, 2012.
- [63] P. Luchini and A. Bottaro. Adjoint equations in stability analysis. *Annual Review of Fluid Mechanics*, 46:493–517, 2014.
- [64] C. Mavriplis. Adaptive mesh strategies for the spectral element method. *Computer Methods in Applied Mechanics and Engineering*, 116:77–86, 1994.
- [65] D.J. Mavriplis. Adaptive mesh generation for viscous flows using triangulation. *Journal of Computational Physics*, 90(2):271 – 291, 1990.
- [66] D.S. McRae. r -refinement grid adaptation algorithms and issues. *Computer Methods in Applied Mechanics and Engineering*, 189(4):1161 – 1182, 2000.
- [67] G. Mengaldo, D. De Grazia, J. Peiró, A. Farrington, F. Witherden, P.E. Vincent, and S.J. Sherwin. A guide to the implementation of boundary conditions in compact high-order methods for compressible aerodynamics. In *7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation, AIAA Paper 2014-2923*, Atlanta, Georgia, USA, 2014.
- [68] G. Mengaldo, D. De Grazia, D. Moxey, P. E. Vincent, and S. J. Sherwin. Dealiasing techniques for high-order spectral element methods on regular and irregular grids. *Journal of Computational Physics*, 299:56–81, 2015.

- [69] R.C. Moura, A.F.C. Silva, E.D.V. Bigarella, A.L. Fazenda, and M.A. Ortega. Lyapunov exponents and adaptive mesh refinement for high-speed flows using a discontinuous galerkin scheme. *Journal of Computational Physics*, 319:9–27, 2016.
- [70] D. Moxey, D. Ekelschot, Ü. Keskin, S. J. Sherwin, and J. Peiró. High-order curvilinear meshing using a thermo-elastic analogy. *Computer-Aided Design*, 72:130–139, 2016.
- [71] S. Nadarajah and A. Jameson. Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint methods. *AIAA Journal*, 45(7):1478–1491, 2007.
- [72] B. Palmerio. A two-dimensional FEM adaptive moving-node method for steady Euler flow simulations. *Computer Methods in Applied Mechanics and Engineering*, 71(3):315 – 340, 1988.
- [73] B. Palmerio. An attraction-repulsion mesh adaption model for flow solution on unstructured grids. *Computers and Fluids*, 23(3):487–506, 1994.
- [74] J. Peraire, J. Peiró, and K. Morgan. Adaptive remeshing for three-dimensional compressible flow computations. *Journal of Computational Physics*, 103:269–285, 1992.
- [75] J. Peraire, M. Vahdati, K. Morgan, and O.C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72:449–466, 1987.
- [76] P.O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous galerkin method. In *44th AIAA Aerospace Sciences Meeting and Exhibit (AIAA Paper 2006-112)*, Reno, Nevada, USA, 2006.
- [77] P.O. Persson and J. Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *Proceedings of the 47th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2009-949*, Orlando, Florida, USA, January 2009.
- [78] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. *Los Alamos Scientific laboratory report, LA-UR-73-479*, 1973.
- [79] G. Rubio, F. Fraysse, J. de Vicente, and E. Valero. The estimation of truncation error by τ -estimation for chebyshev spectral collocation method. *Journal of Scientific Computing*, pages 146–173, 2013.

- [80] P.J. Schmid and D.S. Hennigson. *Stability and transition in shear flows*. Springer, 2001.
- [81] S.J. Sherwin. A high-order Fourier/unstructured discontinuous Galerkin method for hyperbolic conservation laws. In *Hyperbolic Problems: Theory, Numerics, Applications*, volume 130 of *International Series of Numerical Mathematics*, pages 875–884. 1999.
- [82] C.W. Shu. Total variation diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing*, 9(6):1073–1084, 1988.
- [83] C.W. Shu. Different formulations of the discontinuous Galerkin method for the viscous terms. In Z.C. Shi, M. Mu, W.M. Due, and J.Zou, editors, *Advances in Scientific Computing*, pages 144–155. Science Press, 2001.
- [84] P. Solín and L. Demkowicz. Goal-oriented hp -adaptivity for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1):449–468, 2004.
- [85] K. Stein, T. Tezduyar, and R. Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *ASME Journal of Applied Mechanics*, 70(1):58–63, 2003.
- [86] T.E. Tezduyar, M. Senga, and D. Vicker. Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and $YZ\beta$ shock-capturing. *Computational Mechanics*, 38:469–481, 2006.
- [87] K.W. Thompson. Time dependent boundary conditions for hyperbolic systems. *Journal of Computational Physics*, 68:1–24, 1987.
- [88] K.W. Thompson. Time dependent boundary conditions for hyperbolic systems, part 2. *Journal of Computational Physics*, 89:439–461, 1990.
- [89] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics : a Practical Introduction*. Springer, 1997.
- [90] D. A. Venditti and D.L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176(1):40 – 69, 2002.
- [91] D.A. Venditti and D.L. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *Journal of Computational Physics*, 164(1):204–227, 2000.

- [92] D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187(1):22–46, 2003.
- [93] 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.
- [94] Q. Wang, P. Moin, and G. Iaccarino. Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. *SIAM Journal of Scientific Computing*, 31(4):2549–2567, 2009.
- [95] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72:1–42, 2013.
- [96] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115 – 173, 1984.
- [97] 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 and Fluids*, 98:3–16, 2014.
- [98] Z.Q. Xie, R. Sevilla, O. Hassan, and K. Morgan. The generation of arbitrary order curved meshes for 3D finite element analysis. *Computational Mechanics*, 51:361–374, 2013.
- [99] M. Yano and D. L. Darmofal. Case C1.3 Flow over the NACA 0012 airfoil: subsonic inviscid, transonic inviscid, and subsonic laminar flows. Contribution to the First Workshop of High-order CFD Methods held at the 50th AIAA Aerospace Sciences Meeting, Nashville, Tennessee, January 7–8. 2012.
- [100] M. Zhang and C.W. Shu. An analysis of three different formulations of the discontinuous galerkin method for diffusion equations. *Mathematical Models and Methods in Applied Sciences*, 13:395–413, 2003.

Appendix A

Jacobians of the convective and diffusive fluxes

The compressible Euler and Navier-Stokes are linearised to derive their corresponding adjoint equations. As a result, the Jacobians of the convective and diffusive fluxes are required.

A.1 Linearisation of the convective fluxes

The convective fluxes are linearised with respect to the conservative state vector given by

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{Bmatrix} \quad (\text{A.1})$$

The convective fluxes are introduced in section 3.5. For completeness, we consider the three-dimensional Navier-Stokes equations so $N = 2$ and the velocity vector is defined as $\vec{v} = \{v_1, v_2, v_3\}^t$. Hence the Jacobian matrices for the convective fluxes, $\mathbf{f}^c = [\mathbf{f}_1^c \ \mathbf{f}_2^c \ \mathbf{f}_3^c]$ are defined as $\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{f}_2^c}{\partial \mathbf{u}}$ which are 4×4 Jacobian matrices where N indicates the spacial dimension. The matrices are given below.

$$\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(\gamma-1)q^2}{2} - v_1^2 & (3-\gamma)v_1 & -(\gamma-1)v_2 & \gamma-1 \\ -v_1v_2 & v_2 & v_1 & 0 \\ v_1 \left(\frac{(\gamma-1)q^2}{2} \right) & H - (\gamma-1)v_1^2 & -(\gamma-1)v_1v_2 & \gamma v_1 \end{bmatrix} \quad (\text{A.2})$$

$$\frac{\partial \mathbf{f}_2^c}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -v_1 v_2 & v_2 & v_1 & 0 \\ \frac{(\gamma-1)q^2}{2} - v_2^2 & -(\gamma-1)v_1 & (3-\gamma)v_2 & \gamma-1 \\ v_2 \left(\frac{(\gamma-1)q^2}{2} \right) & -(\gamma-1)v_1 v_2 & H - (\gamma-1)v_2^2 & \gamma v_2 \end{bmatrix} \quad (\text{A.3})$$

where $q^2 = v_1^2 + v_2^2$ and the total enthalpy is indicated by H .

A.2 Linearisation of the diffusive fluxes

However, it becomes more convenient to make a change of variables when linearising the diffusive fluxes using the primitive variables given by

$$\mathbf{u}_p = \begin{Bmatrix} \rho \\ v_1 \\ v_2 \\ p \end{Bmatrix} \quad (\text{A.4})$$

Since the diffusive fluxes are dependent on the first derivatives of the velocity component rather than the momentum components it becomes easier to linearise with respect to $\mathbf{u}_p$. However to have consistency, we have to apply the chain rule to obtain $\frac{\partial \mathbf{f}^v}{\partial \mathbf{u}}$. Hence, the Jacobians obtained when linearising the diffusive fluxes with respect to the conservative state vector $\mathbf{u}$ can be written as

$$\frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}} = \frac{\partial \mathbf{u}_p}{\partial \mathbf{u}} \frac{\partial \mathbf{f}_i^v}{\partial \mathbf{u}_p} \quad (\text{A.5})$$

where $\frac{\partial \mathbf{u}_p}{\partial \mathbf{u}}$ is given by

$$\frac{\partial \mathbf{u}_p}{\partial \mathbf{u}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{v_1}{\rho} & \frac{1}{\rho} & 0 & 0 \\ -\frac{v_2}{\rho} & 0 & \frac{1}{\rho} & 0 \\ \frac{(\gamma-1)q^2}{2} & -v_1(\gamma-1) & -v_2(\gamma-1) & \gamma-1 \end{bmatrix} \quad (\text{A.6})$$

Furthermore, we have that $\frac{\partial \mathbf{f}^v}{\partial \mathbf{u}_p} = \left\{ \frac{\partial \mathbf{f}_1^v}{\partial \mathbf{u}_p}, \frac{\partial \mathbf{f}_2^v}{\partial \mathbf{u}_p} \right\}$ where the entries $\frac{\partial \mathbf{f}_1^v}{\partial \mathbf{u}_p}, \frac{\partial \mathbf{f}_2^v}{\partial \mathbf{u}_p}$ are the $(N+2)$ by $(N+2)$ Jacobians of the viscous fluxes when linearising with respect to $\mathbf{u}_p$. The matrices are given below.

$$\frac{\partial \mathbf{f}_1^v}{\partial \mathbf{u}_p} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_1} \left(\frac{1}{\rho} \right) & \tau_{11} & \tau_{12} & \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_1} \left(\frac{p}{\rho^2} \right) \end{bmatrix} \quad (\text{A.7})$$

$$\frac{\partial \mathbf{f}_2^v}{\partial \mathbf{u}_p} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_2} \left(\frac{1}{\rho} \right) & \tau_{21} & \tau_{22} & \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_2} \left(\frac{p}{\rho^2} \right) \end{bmatrix} \quad (\text{A.8})$$

The linearised diffusive fluxes with respect to the first order derivatives are given by

$$\frac{\partial \mathbf{f}_1^v}{\partial \mathbf{u}_{x_1}} = \frac{\mu}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}v_1 & \frac{4}{3} & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ -\left(v_1^2 + \frac{4}{3}v_2^2 + \frac{\gamma}{Pr}(E - q^2)\right) & \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)v_1 & \left(1 - \frac{\gamma}{Pr}\right)v_2 & \frac{\gamma}{Pr} \end{bmatrix} \quad (\text{A.9})$$

$$\frac{\partial \mathbf{f}_1^v}{\partial \mathbf{u}_{x_2}} = \frac{\mu}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3}v_2 & 0 & -\frac{2}{3} & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{1}{3}v_1v_2 & v_2 & -\frac{2}{3}v_1 & 0 \end{bmatrix} \quad (\text{A.10})$$

$$\frac{\partial \mathbf{f}_2^v}{\partial \mathbf{u}_{x_1}} = \frac{\mu}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ \frac{2}{3}v_1 & -\frac{2}{3} & 0 & 0 \\ -\frac{1}{3}v_1v_2 & -\frac{2}{3}v_2 & v_1 & 0 \end{bmatrix} \quad (\text{A.11})$$

$$\frac{\partial \mathbf{f}_2^v}{\partial \mathbf{u}_{x_2}} = \frac{\mu}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{4}{3}v_2 & 0 & \frac{4}{3} & 0 \\ -\left(v_1^2 + \frac{4}{3}v_2^2 + \frac{\gamma}{Pr}(E - q^2)\right) & \left(1 - \frac{\gamma}{Pr}\right)v_1 & \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)v_2 & \frac{\gamma}{Pr} \end{bmatrix} \quad (\text{A.12})$$

Appendix B

Adjoint Roe-Riemann flux function

The Roe-averaged Riemann flux in the normal direction is given by

$$\mathcal{H}^c(\mathbf{u}_{ex}, \mathbf{u}_{in}) = \frac{1}{2} (\mathbf{f}_{n,in}^c + \mathbf{f}_{n,ex}^c) - \frac{1}{2} |\mathbf{\Lambda}| \mathbf{T}^{-1} \Delta \mathbf{u} \quad (\text{B.1})$$

where $\Delta \mathbf{u} = \mathbf{u}_{ex} - \mathbf{u}_{in}$ denotes the difference between the solution at the external and internal interface. Furthermore, $\mathbf{\Lambda}$ and $\mathbf{T}$ follow from diagonalising the Jacobian matrix of the convective flux. Hence, $\mathbf{\Lambda}$ denotes the diagonal 4×4 eigenvalue matrix and $\mathbf{T}$ denotes a 4×4 matrix containing the right eigenvectors.

We use equation (B.1) to construct an adjoint Roe-averaged Riemann solver. First we want to obtain the appropriate formulation for $\mathbf{T}^{-1}$. For the adjoint problem, the Jacobian is transposed and the sign is negative. The transposed Jacobian for the flux in the normal direction gives

$$\left[\frac{\partial \mathbf{f}_1^c}{\partial \mathbf{u}} \right]^t = \begin{bmatrix} 0 & -u^2 + \frac{\gamma-1}{2} q^2 & -uv & u \left(\frac{\gamma-1}{2} q^2 - H \right) \\ 1 & (\gamma-3)u & v & H - (\gamma-1)u^2 \\ 0 & -(\gamma-1)v & u & -(\gamma-1)uv \\ 0 & (\gamma-1) & 0 & \gamma u \end{bmatrix} \quad (\text{B.2})$$

In equation (B.2), u and v represent the velocity components normal and parallel to the interface respectively. The Riemann problem is solved in one-dimension by mapping the local quantities onto the normal of the interface. The matrix of right

eigenvectors of the transposed Jacobian, $-\left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}}\right]^t$, is

$$\mathbf{T} = \begin{bmatrix} \frac{q^2}{2} + \frac{cu}{\gamma-1} & u^2 - H & -v & \frac{q^2}{2} - \frac{cu}{\gamma-1} \\ -u - \frac{c}{\gamma-1} & -u & 0 & -u + \frac{c}{\gamma-1} \\ -v & 0 & 1 & -v \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (\text{B.3})$$

The inverse of $\mathbf{T}$ is given by

$$\mathbf{T}^{-1} = \frac{\gamma-1}{2c^2} \begin{bmatrix} 1 & (c-u) & v & (H-uc) \\ -2 & -2u & -2v & -q^2 \\ 2v & 2uv & 2v^2 + \frac{2c^2}{\gamma-1} & 2vH \\ 1 & (c+u) & v & (H+uc) \end{bmatrix} \quad (\text{B.4})$$

We substitute $\mathbf{T}$ in the definition of the Roe-averaged Riemann flux (B.1) and, using the same analogy as for the governing equations, we write an adjoint Roe-averaged numerical flux function as

$$\hat{\mathcal{H}}(\boldsymbol{\psi}_{ex}, \boldsymbol{\psi}_{in}) = -\frac{1}{2} \left(\left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta \right\}_{in} - \left\{ \left[\frac{\partial \mathbf{f}_n^c}{\partial \mathbf{u}} \right]^t \boldsymbol{\psi}_e^\delta \right\}_{ex} \right) - \frac{1}{2} |\Lambda| \mathbf{T}^{-1} \Delta \boldsymbol{\psi} \quad (\text{B.5})$$

Appendix C

Adjoint boundary conditions for converging-diverging nozzle flow

In section 3.6.1, we calculated the numerical adjoint solution corresponding to the minimisation of the pressure along a converging-diverging nozzle when considering inviscid subsonic flow. This numerical solution was verified using the analytical solution derived by Giles and Pierce [35] as a benchmark. It was mentioned that a similar boundary conditions implementation was adopted as presented by Lozano and Ponsin [62]. However, for completeness, the weakly imposed adjoint boundary conditions in the framework of a DG discretisation are described here as well. To calculate the subsonic inviscid flow solution, two boundary conditions at the inflow ($x = -1$) and one boundary condition at the outflow ($x = 1$) is imposed. The corresponding adjoint boundary conditions at $x = -1$ and $x = 1$ are discussed in the following two sections.

C.1 Adjoint boundary condition at the inflow

Going back to section 3.6.1 shows that an inviscid subsonic ($M = 0.12$) flow through a diverging-converging nozzle was considered. As inflow boundary conditions at $x = -1$, the stagnation enthalpy and the stagnation pressure were fixed ($H_{in} = 4.0$, $p_{t,in} = 1.98$). Lozano and Ponsin [62] proposed to expand the adjoint boundary conditions at the inflow in terms of a new set of inlet variable denoted by $\hat{\mathbf{u}} = \{H, p_t, M\}$. Hence, the adjoint boundary condition presented in equation 3.65 are written as

$$\psi^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial \hat{\mathbf{u}}} \right\} \delta \hat{\mathbf{u}} \Big|_{x=-1} = \psi^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial H} \right\} \Big|_{p_t, M} \delta H + \psi^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial p_t} \right\} \Big|_{H, M} \delta p_t + \psi^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial M} \right\} \Big|_{H, p_t} \delta M \quad (\text{C.1})$$

Based on the inflow boundary conditions set for the flow equations, we know that $\delta H = \delta p_t = 0$. Then the adjoint conditions follow since we need to satisfy

$$\boldsymbol{\psi}^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial M} \right\} \Big|_{H, p_t} = 0 \quad (\text{C.2})$$

As a result we are left with the dependency of the Mach number and we write the adjoint variables corresponding to H , p_t and M by introducing the following adjoint variables at the inlet boundary ($\hat{\boldsymbol{\psi}}_{x=-1} = \{\hat{\psi}_M, \hat{\psi}_H, \hat{\psi}_{p_t}\}^t$)

$$\hat{\psi}_M = \frac{M \left(1 + \frac{\gamma-1}{2} M^2\right)}{\rho u (1 - M^2)} \left\{ \frac{\partial \mathbf{f}_1^c}{\partial M} \right\}^t \Big|_{H, p_t} \quad \boldsymbol{\psi} = \psi_1 + u\psi_2 + H\psi_3 \quad (\text{C.3})$$

$$\hat{\psi}_H = \frac{2H}{\rho u} \left\{ \frac{\partial \mathbf{f}_1^c}{\partial H} \right\}^t \Big|_{M, p_t} \quad \boldsymbol{\psi} = -\psi_1 + H\psi_3 \quad (\text{C.4})$$

$$\hat{\psi}_{p_t} = p_t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial p_t} \right\}^t \Big|_{H, M} \quad \boldsymbol{\psi} = \rho u \psi_1 + (\rho u^2 + P) \psi_2 + \rho u H \psi_3 \quad (\text{C.5})$$

The same change of variables has been adopted in [35, 62]. The flow boundary conditions at the inlet imply that $\hat{\psi}_M = 0$ while $\hat{\psi}_H$ and $\hat{\psi}_{p_t}$ are extrapolated from the interior. Hence, we write this in matrix form as

$$\begin{Bmatrix} \hat{\psi}_M \\ \hat{\psi}_H \\ \hat{\psi}_{p_t} \end{Bmatrix} = \begin{bmatrix} 1 & u & H \\ -1 & 0 & H \\ \rho u & p + \rho u^2 & \rho u H \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{Bmatrix} \quad (\text{C.6})$$

By taking the inverse of the matrix given in equation (C.6) we write $\boldsymbol{\psi}_{x=-1}$ in terms of $\hat{\boldsymbol{\psi}}_{x=-1}$ as

$$\psi_1 = -\frac{1}{2} \hat{\psi}_H - \frac{u}{2p} \hat{\psi}_{p_t} \quad (\text{C.7})$$

$$\psi_2 = \frac{1}{p} \hat{\psi}_{p_t} \quad (\text{C.8})$$

$$\psi_3 = \frac{1}{2H} \hat{\psi}_H - \frac{u}{2pH} \hat{\psi}_{p_t} \quad (\text{C.9})$$

The adjoint variables corresponding to H and p_t are calculated using

$$\hat{\psi}_H = -\psi_1 + H\psi_3 \quad (\text{C.10})$$

$$\hat{\psi}_{p_t}^e = \rho u \psi_1 + (\rho u^2 + P) \psi_2 + (\rho u H) \psi_3 \quad (\text{C.11})$$

where ψ_1 , ψ_2 and ψ_3 are taken from the inner state at $x = -1$.

C.2 Adjoint boundary condition at the outflow

A similar approach is adopted to derive the adjoint boundary conditions for the outflow boundary at $x = 1$. However, for the outflow we know that the pressure is fixed and therefore the variation $\delta p = 0$. To impose this we perform again a change of variables. However, now $\hat{\mathbf{u}} = \{u, v, p\}^t$ and we write the adjoint boundary condition at $x = 1$ as

$$\boldsymbol{\psi}^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial \hat{\mathbf{u}}} \right\} \delta \hat{\mathbf{u}} \Big|_{x=1} = \boldsymbol{\psi}^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial \rho} \right\} \Big|_{u,p} \delta \rho + \boldsymbol{\psi}^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial u} \right\} \Big|_{\rho,p} \delta u + \boldsymbol{\psi}^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial p} \right\} \Big|_{\rho,u} \delta p \quad (\text{C.12})$$

As mentioned before, $\delta p = 0$. Hence the adjoint boundary conditions follow from the fact that

$$\boldsymbol{\psi}^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial \rho} \right\} \Big|_{u,p} = 0, \quad \boldsymbol{\psi}^t \left\{ \frac{\partial \mathbf{f}_1^c}{\partial u} \right\} \Big|_{\rho,p} = 0 \quad (\text{C.13})$$

We write the alternative adjoint variables, $\hat{\boldsymbol{\psi}}_{x=1}$, at the outflow boundary as

$$\hat{\psi}_\rho = \frac{1}{u} \left\{ \frac{\partial \mathbf{f}_1^c}{\partial \rho} \right\}^t \Big|_{u,p} \quad \boldsymbol{\psi} = \psi_1 + u\psi_2 + \frac{u^2 H}{2} \psi_3 \quad (\text{C.14})$$

$$\hat{\psi}_u = \frac{1}{\rho} \left\{ \frac{\partial \mathbf{f}_1^c}{\partial u} \right\}^t \Big|_{\rho,u} \quad \boldsymbol{\psi} = \psi_1 + 2u\psi_2 + (u^2 + H)\psi_3 \quad (\text{C.15})$$

$$\hat{\psi}_p = \left\{ \frac{\partial \mathbf{f}_1^c}{\partial p} \right\}^t \Big|_{\rho,u} \quad \boldsymbol{\psi} = \psi_2 + \frac{\gamma}{\gamma - 1} \psi_3 \quad (\text{C.16})$$

Writing this in matrix form gives

$$\begin{pmatrix} \hat{\psi}_\rho \\ \hat{\psi}_u \\ \hat{\psi}_p \end{pmatrix} = \begin{bmatrix} 1 & u & \frac{u^2 H}{2} \\ 1 & 2u & u^2 + H \\ 0 & 1 & \frac{\gamma}{\gamma - 1} \end{bmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} \quad (\text{C.17})$$

Again by taking the inverse of the matrix and knowing that $\hat{\psi}_\rho = \hat{\psi}_u = 0$, it allows us to write $\boldsymbol{\psi}_{x=1}$ in terms of $\hat{\boldsymbol{\psi}}_{x=1}$ as

$$\psi_1 = \frac{uH}{\beta} \psi_p^e \quad (\text{C.18})$$

$$\psi_2 = \frac{H + \frac{u^2}{2}}{\beta} \psi_p^e \quad (\text{C.19})$$

$$\hat{\psi}_p = \frac{u}{\beta} \psi_p^e \quad (\text{C.20})$$

where $\beta = \frac{\gamma}{\gamma-1}u^2 - H - \frac{u^2}{2}$ and where

$$\psi_p = \psi_2 + \frac{\gamma}{\gamma-1}u\psi_3 \quad (\text{C.21})$$

Here, ψ_2 and ψ_3 are taken from the inner state at $x = 1$.

Appendix D

Journal publications and conference proceedings

D.1 Journal publications

- **D. Ekelschot** , D. Moxey, S.J. Sherwin, and J. Peiró. *Combining r -adaptation using a thermo-elastic analogy with p -adaptation for compressible flows*, In preparation for submission to the Journal of Computational Physics.
- **D. Ekelschot** , D. Moxey, S.J. Sherwin, and J. Peiró. *p -adaptation for compressible laminar flow problems using a goal-based error estimator*, Computers and Structures, doi:10.1016/j.compstruc.2016.03.004, 2016.
- D. Moxey, **D. Ekelschot**, Ü. Keskin, S.J. Sherwin and J. Peiró. *High-order curvilinear meshing using a thermo-elastic analogy*, Computer Aided Design, 73:130-139, 2016.
- C.D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, **D. Ekelschot** , B. Jordi, H. Xu , Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R.M. Kirby, S.J. Sherwin. *Nektar++: An open-source spectral/hp element framework*, Computer Physics Communications Volume 192, pp. 205-219, 2015.

D.2 Conference proceedings

- 23rd Conference on Computational Mechanics ACME-UK 2015, Swansea, Wales, United Kingdom, April 8-10 2015. Presentation was given on *p-adaptation for compressible laminar flow problems using goal-based error estimation*.
Won the prize for best postgraduate paper.
- 67th Annual Meeting of the APS Division of Fluid Dynamics, San Francisco, California, USA, November 23-25 2014. Presentation was given on *p-adaptation for compressible laminar flow problems using goal-based error estimation*.
- 6th European Conference on Computational Fluid Dynamics (ECCOMAS ECFD VI), Barcelona, Spain, July 20-25 2014. Presentation was given on *p-adaptation for compressible inviscid flow problems using goal-based error estimation*.
- 6th International Conference on Adaptive Modelling and Simulation (ADMOS 2013), Lisbon, Portugal, June 3-5, 2013. A presentation given on *p-adaptation for compressible flows*.

D.3 Attendance to workshops

- *High-order methods for flow simulation*. Imperial College London, London, United Kingdom, January 16-18, 2013. (Organiser)
- *Progress in flow instability analysis and laminar-t Transition modeling*. Von Karman Institute for Fluid Dynamics, Brussels, Belgium, July 2-6 2014.
- *Numerical prediction of detached flows*. Universidad Politécnica de Madrid, Madrid, Spain, October 23-24 2014.
- *Receptivity and sensitivity analyses; uncertainties and mesh adaptation algorithms*. University of Cambridge, Cambridge, United Kingdom, September 23-26 2014.
- *Introduction to CFD and adjoint for aerodynamic shape optimisation with the DLR TAU code*. Universidad Politécnica de Madrid, Madrid, Spain, September 22-25 2015.