# An adaptive high order Reynolds-averaged Navier-Stokes solver with transition prediction

by

David Moro-Ludeña

Ing., Universidad Politécnica de Madrid (2007)
S.M., Massachusetts Institute of Technology (2011)

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

Doctor of Philosophy in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015



Author .....
Department of Aeronautics and Astronautics
December 10, 2014

Certified by .....
Jaume Peraire
H. N. Slater Professor of Aeronautics and Astronautics
Thesis supervisor

Certified by .....
Ngoc Cuong Nguyen
Senior Research Scientist
Thesis supervisor

Certified by .....
Mark Drela
Terry J. Kohler Professor of Fluid Dynamics
Thesis supervisor

Certified by .....
David Darmofal
Professor of Aeronautics and Astronautics
Thesis committee member

Accepted by .....
Paulo Lozano
Associate Professor of Aeronautics and Astronautics
Graduate Committee Chair

# An adaptive high order Reynolds-averaged Navier-Stokes solver with transition prediction

by

David Moro-Ludeña

Submitted to the Department of Aeronautics and Astronautics
on December 10, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

## Abstract

The use of simulation techniques in applied aerodynamics has increased dramatically in the last three decades fostered by the growth in computational power. However, the state of the art discretization in industrial solvers remains nominally second order accurate, which makes them unfeasible to resolve multi-scale phenomena such as turbulence or acoustics, and limits their efficiency in terms of the error per degree of freedom. In recent years, the CFD community has put significant effort into the development of high order methods for fluid dynamics, with the goal of overcoming these barriers.

This dissertation focuses on the application of high order hybridizable discontinuous Galerkin schemes to solve the equations that govern compressible turbulent flows. In particular, this thesis describes a novel methodology to adapt the boundary layer mesh to the solution "on the fly", based on a measure of the boundary layer thickness that drives the position of the nodes in the mesh, without changing its topology. The proposed algorithm produces accurate solutions with a reduced number of degrees of freedom, by leveraging the combination of mesh adaptivity with the high order of convergence of the discretization. In addition, the active tracking of the boundary layer reduces the nonlinear stiffness and improves the robustness of the numerical solution.

A new shock capturing technique based on the addition of artificial viscosity is developed to handle shocks. The model is driven by a non-dimensional form of the divergence of the velocity, designed so that sub-cell shock resolution is achieved when a high order discretization is used, independently of the element size.

The approach is extended to include the effect of transition to turbulence using an envelope $e^N$ method. This takes advantage of the structure of the mesh and requires the solution of a surface PDE for the transition criterion, which is discretized using a novel surface hybridizable discontinuous Galerkin scheme. The resulting method can simulate transition to turbulence in attached and separated flows, and can also accommodate long-scale unsteadiness in which the transition location evolves in time.

Thesis Supervisor: Jaume Peraire
Title: H. N. Slater Professor of Aeronautics and Astronautics

Thesis Supervisor: Ngoc Cuong Nguyen
Title: Senior Research Scientist

Thesis Supervisor: Mark Drela
Title: Terry J. Kohler Professor of Fluid Dynamics

# Acknowledgments

First of all, I would like to thank my advisors, Prof. Jaume Peraire, Dr. Cuong Nguyen and Prof. Mark Drela for taking me in as a student and being an endless source of wisdom and knowledge from which I have tried to learn as much as possible. I would also like to express my gratitude to Prof. David Darmofal for being part of the thesis committee and providing feedback and guidance during the course of this research project. Similarly, I would like to show my appreciation to Dr. Steven Allmaras and Prof. Per-Olof Persson for acting as readers of this dissertation, and helping shape the final form of this document through detailed comments and questions.

Secondly, I would like to thank the ACDL family for being the most laid-back, hard-working, all-around-smart group of people that the department of Aeronautics and Astronautics has. This also goes for the people that keep it running on the background. I would also like to express my gratitude to Beth Marois for all her help throughout these years in trying to navigate my way around this maze. My special thanks go to Jean Sofronas and Joyce Light at the department headquarters for being so helpful and cheerful and supportive and great; I can certainly say I have always left that office with a smile in my face.

Life outside of MIT has been all ups, and that is thanks to many people. I would like to thank friends in and out of the lab for being the best company one could have four seasons around. Being around you made these five years go by really fast. I will never forget the time I have shared with you that includes a healthy dose of cultural travelling, fine homemade dining, and sports, as well as spring-break-like partying and everything required to kill the gains.

I cannot miss this opportunity for thanking my family and friends back in Spain, who have been extremely supportive of my endeavors and genuinely shared my ups and downs an ocean away. I would like to thank my sister for making me an uncle twice in the last years, my mother for moving mountains for me every time I needed her, my father for teaching me that you will loose many times in life but you always

have to stand back up (like Atletico de Madrid football club, of which we are both hardcore fans) and my brother for coming here every summer to spend some quality time together. Apart from him, we have been delighted with hosting plenty of visitors to Boston over these years, and hope to keep on doing it wherever we end up.

Finally, I would like to thank my soul mate Carmen for these wonderful years together.

This thesis is dedicated to my grandparents, who never set foot on a school, but taught me some of the most important lessons I have ever taken. You sacrificed many things so I am here today and for that I will be forever grateful.

# Contents

**1 Introduction** **19**

1.1 Motivation . . . . . 19

1.2 Objectives . . . . . 20

1.3 Background . . . . . 21

1.3.1 High order schemes for fluids . . . . . 21

1.3.2 Shock capturing for high order methods . . . . . 25

1.3.3 Application of high order methods to the simulation of turbulent flows . . . . . 29

1.3.4 Transition prediction in RANS solvers . . . . . 34

1.3.5 Mesh generation and adaptation techniques for boundary layer flows . . . . . 44

1.4 Thesis overview . . . . . 48

**2 Methodology** **51**

2.1 Governing Equations . . . . . 51

2.1.1 Navier-Stokes equations . . . . . 51

2.1.2 Spalart-Allmaras turbulence model . . . . . 53

2.1.3 Arbitrary Lagrangian-Eulerian formulation . . . . . 57

2.2 Flow discretization . . . . . 58

2.2.1 Spatial discretization: the hybridizable discontinuous Galerkin scheme . . . . . 59

2.2.2 Time discretization . . . . . 65

2.3 Flow discretization module . . . . . 66

**3 Shock capturing with dilatation-based artificial viscosity 67**

3.1 Formulation . . . . . 69

3.1.1 Original formulation by Nguyen and Peraire . . . . . 69

3.1.2 Modifications to the original model . . . . . 70

3.1.3 Summary of the artificial viscosity model . . . . . 79

3.2 One dimensional studies . . . . . 80

3.2.1 Comparison between $c$ and $c^*$ as velocity scale . . . . . 81

3.2.2 Choice of length scale correction factor . . . . . 83

3.2.3 Validation cases . . . . . 85

3.3 Two dimensional studies . . . . . 90

3.3.1 Supersonic flow in a duct with a ramp . . . . . 92

3.3.2 Supersonic flow inside a Scramjet geometry . . . . . 96

3.3.3 Flow around a forward facing step . . . . . 101

3.3.4 Effect of the artificial viscosity on the convergence of the subsonic flow over a smooth bump . . . . . 104

3.4 Conclusions and future work . . . . . 109

**4 $r$-adaptivity for viscous flows 111**

4.1 Introduction . . . . . 111

4.2 Formulation . . . . . 113

4.2.1 Motivation . . . . . 113

4.2.2 Normal scaling equation . . . . . 118

4.2.3 Mesh deformation . . . . . 121

4.2.4 Numerical evaluation of integral boundary layer quantities . . . 127

4.2.5 Discussion of the coupling . . . . . 129

4.2.6 Solution procedure . . . . . 129

4.3 Results . . . . . 131

4.3.1 Laminar boundary layer over a flat plate . . . . . 131

4.3.2 Laminar separation bubble over a smooth bump . . . . . 134
4.3.3 Turbulent flow over a flat plate . . . . . 138
4.3.4 Turbulent flow around a NACA 0012 Airfoil . . . . . 142
4.3.5 Viscous high-speed flow over a cylinder . . . . . 147
4.3.6 Viscous high-speed flow over a blunt wedge . . . . . 150
4.3.7 Shock wave boundary layer interaction . . . . . 156
4.4 Conclusions and future work . . . . . 161

**5 Transition prediction on the $r$-adaptive solver 165**
5.1 Introduction . . . . . 165
5.2 Formulation . . . . . 167
5.2.1 Governing equations . . . . . 169
5.2.2 High order surface HDG . . . . . 171
5.2.3 Coupling with the Spalart-Allmaras model . . . . . 175
5.3 Solution procedure . . . . . 178
5.4 Results . . . . . 180
5.4.1 Natural transition over a NACA 0012 airfoil . . . . . 180
5.4.2 Separation induced transition over an Eppler 387 airfoil . . . . . 185
5.4.3 Unsteady transition over a NACA 0012 airfoil . . . . . 192
5.5 Conclusions and future work . . . . . 196

**6 Concluding Remarks 199**
6.1 Summary and conclusions . . . . . 199
6.2 Future work . . . . . 200

**A Stability analysis of the normal scaling equation 205**
A.1 Simplified equation . . . . . 205
A.2 Stability conditions . . . . . 206
A.3 Stability analysis . . . . . 207
A.4 Parameter choice . . . . . 209

A.4.1 Initialization of $\delta$ . . . . . 211
A.4.2 Selection of $\tau_\delta$ . . . . . 212
A.5 Summary of the study . . . . . 212

**B On the transition to turbulence of the Spalart-Allmaras model at low Reynolds numbers 215**

# List of Figures


1-1 Evolution of the CFD tools available at The Boeing Company. . . . . 19

3-1 Sketch of the procedure to extract $h_e$ for triangles and quadrilaterals, based on identifying the smallest altitude in the polygon. . . . . 72

3-2 Estimates for $\tilde{s}_1$, $\tilde{s}_2$ and $\tilde{s}^*$ as a function of the incident Mach number $M_{1n}$. . . . . 76

3-3 One dimensional shock wave simulations computed using $c$ or $c^*$ as the velocity scale. . . . . 82

3-4 One dimensional shock wave simulations computed using different values of the length scale correction factor $k_h$. . . . . 84

3-5 One dimensional shock wave simulations for different values of the upstream Mach number $M_{1n}$. . . . . 86

3-6 One dimensional shock wave simulations for different values of the approximation order $p$. . . . . 87

3-7 One dimensional shock wave simulations for different locations of the shock center. . . . . 89

3-8 Sketch of the supersonic ramp geometry and boundary conditions. . . . 92

3-9 Evolution of the mesh with the anisotropic adaptivity iteration, for the case of the supersonic compression ramp. . . . . 93

3-10 Evolution of the shock sensor with the anisotropic adaptivity iteration, for the case of the supersonic compression ramp. . . . . 94

3-11 Mach number field and entropy field around a supersonic ramp computed using polynomials of order $p = 4$ on the final adapted mesh. . . . 95
3-12 Pressure signature on the top and bottom wall of the supersonic compression ramp. . . . . 95
3-13 Sketch of the scramjet geometry and boundary conditions. . . . . 96
3-14 Evolution of the mesh for the scramjet flow as a function of the number of anisotropic adaptivity iterations. . . . . 98
3-15 Flow inside the scramjet geometry computed on the finest mesh (12 adaptivity cycles) using polynomials of order $p = 4$. . . . . 99
3-16 Detail of the flow around the nozzle throat of the scramjet. . . . . 100
3-17 Detail of the flow around the trailing edge of the inner body of the scramjet. . . . . 100
3-18 Sketch of the geometry and boundary conditions for the forward facing step problem. . . . . 101
3-19 Evolution of the mesh for the forward facing step flow as a function of the number of anisotropic adaptivity iterations. . . . . 102
3-20 Flow around the forward facing step geometry computed in the finest mesh (12 refinement iterations) using polynomials of order $p = 4$. . . . . 103
3-21 Sketch of the geometry and boundary conditions for the case of the smooth subsonic bump. . . . . 105
3-22 Sample mesh and solution for the subsonic flow over a smooth bump. . 107
3-23 Convergence in the entropy error for different polynomial orders using structured triangular meshes. . . . . 108

4-1 Definition of the boundary layer integrals $\delta_k^*$ and $\theta_k$. . . . . 114
4-2 Sketch of the mesh topology for different kinds of geometries. . . . . 117
4-3 Extraction of the high order surface mesh from the volume mesh. . . . . 119
4-4 Reconstruction of the high order geometry of an element in a given layer of the stack. . . . . 123
4-5 Diagram of the mappings that define the external mesh deformation. . 124

4-6 Sketch of the correction procedure to compute the boundary layer integrals. . . . . 128
4-7 Flow chart of the adaptive solver showing the interdependence of the different modules and variables. . . . . 130
4-8 Sketch of the laminar flat plate geometry and boundary conditions. . . 132
4-9 Comparison of the solutions obtained on a sequence of meshes with equivalent resolution against the analytical solution by Blasius. . . . . 133
4-10 Normal scaling and mesh at convergence for the case of the laminar flat plate. . . . . 133
4-11 Sketch of the laminar bump geometry and boundary conditions. . . . . 134
4-12 Flow field and mesh obtained for the case of a laminar separation bubble. 136
4-13 Comparison of pressure and friction coefficients computed on a sequence of meshes for the laminar bump case. . . . . 137
4-14 Evolution of the shape parameter $H_k$ with different approximation orders for the laminar bump case. . . . . 137
4-15 Sketch of the turbulent flat plate geometry and boundary conditions. . 138
4-16 Friction coefficient along the flat plate compared to experimental data, empirical correlations, as well as grid converged results for the CFL3D solver. . . . . 139
4-17 Horizontal velocity profiles measured in wall units ($y^+$ vs. $u^+$) at different stations along the flat plate. . . . . 140
4-18 Comparison of $y_1^+$ and $\delta$ along the flat plate. . . . . 141
4-19 Sketch of the geometry and boundary conditions for the turbulent flow around a NACA 0012 airfoil. . . . . 142
4-20 Flow field and mesh obtained for the case of a NACA 0012 at $M_\infty = 0.15$, $Re_c = 6 \cdot 10^6$ and $\alpha = 15°$. . . . . 144
4-21 Comparison of pressure and friction coefficients computed using the $r$-adaptive solver versus experimental data, CFL3D and Xfoil. . . . . 145

4-22 Comparsion of the computed values of lift and drag versus experimental data, CFL3D and Xfoil. . . . . 146
4-23 Chord-wise evolution of the normal scaling $\delta$ and the distance to the first degree of freedom off the wall $(y_1^+)$ for three different angles of attack. 146
4-24 Sketch of the geometry and boundary conditions for the case of the high-speed cylinder. . . . . 147
4-25 Meshes used to compute the high-speed flow around a cylinder. . . . . 148
4-26 High-speed flow field around a cylinder computed on a refined mesh after 5 iterations of anisotropic refinement. . . . . 149
4-27 Evolution of the pressure and friction coefficient around the cylinder with the adaptation cycle. . . . . 149
4-28 Sketch of the geometry and boundary conditions for the case of the blunt wedge. . . . . 150
4-29 Evolution of the mesh with the anisotropic adaptation for the case of the blunt wedge. . . . . 151
4-30 Pressure and friction coefficient over the blunt wedge. . . . . 152
4-31 Rendering of the Mach number and entropy around the leading edge of the blunt wedge. . . . . 153
4-32 Entropy and velocity profiles at different stations along the blunt wedge. 154
4-33 Evolution of the entropy at the edge of the boundary layer domain as well as the normal scaling along the wedge. . . . . 155
4-34 Sketch of the different flow features in a shock wave boundary layer interaction with separation. . . . . 156
4-35 Sketch of the geometry and boundary conditions for the case of the shock wave boundary layer interaction. . . . . 157
4-36 Evolution of the mesh around the separation bubble in the shock wave boundary layer interaction case. . . . . 158
4-37 Mach number field and pressure field around the flat plate. . . . . 159

4-38 Velocity and density profiles around the laminar separation bubble extracted from the boundary layer domain. . . . . 159
4-39 Evolution of the stresses at the wall with the external mesh adaptation cycle. . . . . 160
4-40 Pressure signature at the edge of the boundary layer domain as a function of the external mesh adaptation cycle. . . . . 160

5-1 Sketch of the evolution of the intermittency before, during and after transition. . . . . 178
5-2 Flow chart of the adaptive solver with transition prediction showing the interdependence of the different modules and variables. . . . . 179
5-3 Sketch of the geometry and boundary conditions for the case of a transitional flow over a NACA 0012 airfoil. . . . . 181
5-4 Sequence of meshes generated for the simulation of the transitional flow over a NACA 0012 airfoil. . . . . 182
5-5 Rendering of the solution on the finest mesh for the case of the transitional flow over a NACA 0012 at $M_\infty = 0.15$, $Re_c = 6 \cdot 10^6$ and $\alpha = 6°$. . . 183
5-6 Evolution of the amplification factor $\tilde{n}$ and distance of the first node off the wall $y_1^+$ along the chord. . . . . 183
5-7 Comparison of the pressure and friction coefficient as the mesh is refined for the transitional flow around a NACA 0012 airfoil. . . . . 184
5-8 Sketch of the geometry and boundary conditions for the case of separation induced transition over an Eppler 387 airfoil. . . . . 187
5-9 Summary of results obtained for the case of an Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$ and $\alpha = 4°$. . . . . 188
5-10 Summary of results obtained for the case of an Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$ and $\alpha = 7.5°$. . . . . 189
5-11 Summary of results obtained for the case of an Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$ and $\alpha = 9°$. . . . . 190

5-12 Computed locus of separation, transition and reattachment as a function of the angle of attack for the Eppler 387 at $Re_c = 2 \cdot 10^5$. . . . . 191
5-13 Computed $c_d$ vs. $c_l$ for the case of the Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$, compared to experiments and other numerical tools. . . . . 191
5-14 Sketch of the geometry and boundary conditions for the case of the unsteady transitional flow over a NACA 0012 airfoil oscillating in pitch. 192
5-15 Evolution of the forces on the airfoil with time for the case of $\bar{k} = 0.2$. . 194
5-16 $c_d$ vs. $c_l$ as a function of the reduced frequency and compared to the steady state polar. . . . . 194
5-17 Comparison of the friction coefficient along the upper surface of the airfoil over one period for two different reduced frequencies. . . . . 195
5-18 Comparison of the amplification factor $\tilde{n}$ along the upper surface of the airfoil over one period for two different reduced frequencies. . . . . 196
5-19 Comparison of the evolution of the normal scaling $\delta_h$ along the upper surface of the airfoil over one period for two different reduced frequencies. 196

A-1 Sketch of the dependency of the edge velocity on $\delta$ and its effect on $\delta^*_{\mathrm{k}}$ and $\theta_{\mathrm{k}}$. . . . . 207
A-2 Analysis of the stability of the thickness indicator equation for the case of a laminar separated flow and a turbulent attached flow. . . . . 209
A-3 Lower stability limit $\delta^-/\bar{\delta}$, computed using different families of velocity profiles (Falkner-Skan, Whitfield and Swafford). . . . . 210

B-1 Flow around an Eppler 387 airfoil at $Re_c = 10^5$ and $\alpha = 6°$. . . . . 216
B-2 Evolution of $c_p$, $c_f$, $H_{\mathrm{k}}$ and $\tilde{n}$ on the upper surface of the airfoil as a function of the trip constant $c_0$. . . . . 217
B-3 Flow around an Eppler 387 airfoil at $Re_c = 10^5$ and $\alpha = 6°$ when the trip term constant is set to $c_0 = 2000$. . . . . 218

# List of Tables

2.1 Parameters of the Spalart-Allmaras model . . . . . 55

5.1 Averaged force coefficients on the airfoil for different values of the reduced frequency. . . . . 195

# Chapter 1

# Introduction

## 1.1 Motivation

The growth of computational power available in commodity clusters and workstations has revolutionized the way computers are used in aircraft design. According to Johnson *et al.* [109], Computational Fluid Dynamics (CFD) has come to replace wind tunnel testing to the point where, for some configurations like cruise, only the final design is built and tested in the wind tunnel (see Figure 1-1).

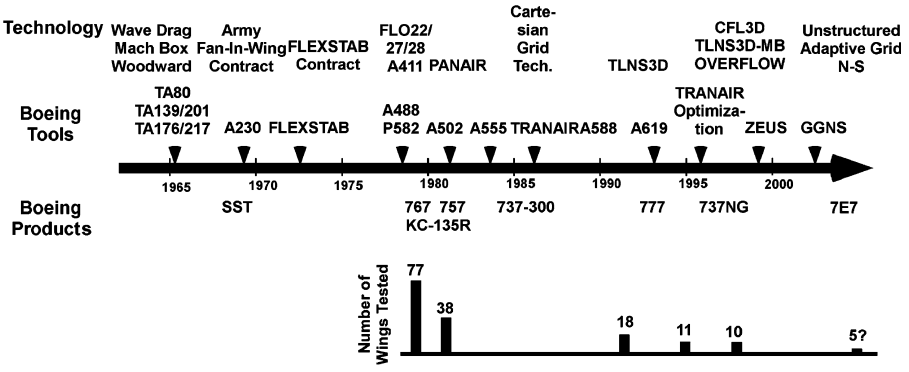



Figure 1-1: Evolution of the CFD tools available at The Boeing Company versus number of wings tested in the wind tunnel. Extracted from Johnson *et al.* [109].

According to the final report of the NASA CFD Vision 2030 Study [220], this

increase in the use of CFD as a design tool has been fostered by the increasing capabilities of current computer clusters, but still requires a significant amount of user intervention and know how to run robustly in general geometries and flow conditions.

Furthermore, the same study points out that the applicability of the current state of the art solvers to flows with moderate separation is totally inadequate, proposing several fronts in which further work is required, such as:

- the use of more advanced turbulence models (e.g. Hybrid RANS-LES or wall modeled LES) and the inclusion of transition prediction in them to reduce the mesh requirements [220] and to improve the agreement with experiments [241, 185],
- the development of solvers based on high order discretization techniques combined with advanced mesh adaptivity, and,
- efforts towards automation of the process such as better mesh generation tools or more robust nonlinear solvers.

As expected, all these thrusts are somewhat connected, and may prompt extra research items at a lower level. Of particular interest is the relationship between high order methods and robustness, which has been identified as one of the barriers that prevents the widespread adoption of the former [251].

## 1.2 Objectives

This thesis pursues some of these venues through the development of a novel high order adaptivity algorithm for the RANS equations with transition prediction. In particular, the proposed algorithm targets the boundary layer region as the solver evolves to the solution and ensures that the resolution there is always close to optimal. In this way, the solver:

- produces accurate solutions with limited amount of degrees of freedom in an automated way,

- tracks the boundary layer along the nonlinear iteration, reducing the stiffness of the problem, and hence, increasing the robustness of the solver, and,
- simplifies the task of extracting accurate boundary layer data that is required to drive a stability analysis model for transition prediction.

In the case of compressible flows, the robustness of the proposed methodology is affected by the presence of shock waves. In order to solve this problem, this thesis also touches upon shock capturing techniques for high order methods, by proposing a simple yet effective strategy based on artificial viscosity.

The particular details of all these models and algorithms will be described in the coming chapters, and justified with the help of theoretical and numerical results in 1-D and 2-D.

## 1.3 Background

The contributions of this thesis build upon a vast body of literature on a variety of topics. Of particular importance here are high order methods, mesh adaptivity, shock capturing, turbulence modeling and transition prediction.

The coming sections contain an account of the seminal papers in each relevant topic, as well as works that have served as direct inspiration for the approach proposed here.

### 1.3.1 High order schemes for fluids

The solution of fluid flow equations using first and second order accurate Finite Differences (FD), Finite Volumes (FV) or Finite Elements (FE), is nowadays a mature technology. These schemes have been implemented with great success in industrial codes thanks to their inherent robustness, which can be linked to a strong numerical dissipation. The later is the main reason why these schemes are rarely used in other communities like computational aero-acoustics, or the direct numerical simula-

tion of turbulent flows, where localized flow structures need to be accurately convected without artificial decay [251].

In the field of applied aerodynamics, most of the codes that solve the steady Euler or Navier-Stokes equations are either FV (for unstructured meshes) or FD (for multi-block, structured meshes) [109, 207]. These codes have been tested on relevant flight configurations using very fine meshes that yield good agreement with experiments [244, 139, 209]. However, this comes at the expense of tremendous computational resources.

In recent years, there has been an increasing interest in the use of high order methods for applied aerodynamics, in an effort to leverage the higher convergence rates in the adaptation process, reducing the cost required to achieve a certain error in the solution. These efforts have been supported by a variety of funding agencies, e.g. the European Union (ADIGMA project [125], IDIHOM project [1]) or the AFOSR [251]. An international group of leaders in the high order methods community has taken a step forward and put together a yearly meeting (AIAA High Order Methods Workshop [251]) as a common ground to discuss the advances and pacing items required to make high order CFD attractive to industry. One of the first tasks of this group was to come to an agreement as to what does high order mean: the unanimous consensus being third order or higher [251].

There exist a wide variety of algorithms that fit this definition. With the exception of spectral methods, most other high order schemes are an extension of a low order counterpart or a mixture of them. In what follows, a list containing the most representative schemes applied with success to fluid flows is presented.

- Spectral methods; based on the representation of the solution on a very high order orthogonal expansion that has support over the whole domain and achieves spectral convergence ("infinite order accurate") for smooth enough solutions [36, 86]. These are usually restricted to a single Cartesian domain although extensions to multiple domains have been devised, e.g. the staggered grid multidomain spectral method by Kopriva [119].

- Continuous Galerkin finite element methods; based on the use of piecewise continuous polynomials with compact support on the elements of the mesh [17]. An example of this kind would be the spectral element method (SE) introduced by Patera [179] for the incompressible Navier-Stokes equations, based on a standard Galerkin discretization. Other high order implementations based on stabilized formulations [103] like the Streamline-Upwinded Petrov-Galerkin (SUPG)[31] or the Galerkin Least Squares (GLS)[102] can also be found in the literature [217, 254].

- High order finite volume methods; based on a cell-average representation of the solution combined with a reconstruction procedure to generate a high order approximation. An example of these would be the Essentially Non-Oscillatory (ENO) scheme developed by Harten and Osher [89], based on an adaptive stencil that ignores oscillatory reconstructions around discontinuities. An extension of ENO known as Weighted-ENO (WENO) was later introduced by Liu, Osher and Chan [142] to fix some of the robustness issues of ENO by means of a convex combination of all the stencils using the appropriate weights. The extension of these schemes to unstructured meshes was developed by Hu and Shu [98] and applied to a variety of compressible cases with shocks waves.

- Finite difference methods; based on the discretization of the differential operator by means of a finite difference approximation, that can be made arbitrarily high order. An example of these would be the compact difference formulations (CD) (see Lele [138] or Visbal [246] ), and the summation-by-parts operators (SBP) (see Strand and others [231, 147]), both of which have been the source of significant research for their reduced memory footprint and increased stability, respectively.

- Correction procedure via reconstruction; based on satisfying the differential form of the equations inside each element, combined with a high order correction function to accommodate the interface fluxes, which are computed using Rie-

mann solvers. Examples of this would be the Flux Reconstruction scheme by Huynh [104] or Vincent *et al.* [245] for structured meshes as well as the work in Lifting Collocation Penalty methods by Wang and Gao [252] for unstructured meshes.

- Discontinuous Galerkin finite element methods (DG); based on a weak formulation of the problem using piecewise discontinuous polynomials over the mesh, combined with Riemman solvers borrowed from the FV method to stabilize the convective operator. The origins of the DG method can be traced back to the work of Reed and Hill in 1973 [202], however, it was not picked up by the CFD community until Allmaras [9] developed the first DG scheme for the Euler equations, followed by the Runge-Kutta DG scheme by Cockburn and Shu [48, 50], and the related work by Bassi and Rebay [26]. At that point, the applicability of DG to elliptic equations was not a solved problem and several discretization options flourished (e.g BR2 scheme [25], LDG scheme [49]). At the same time, the so-called interior penalty methods (IP) where rediscovered, and a proper unifying framework was developed by Arnold *et al.* [15]. Since then, the development of DG schemes for elliptic operators has been focusing on the computational cost, yielding techniques like compact DG (CDG) [182] or the hybridizable discontinuous Galerkin (HDG) formulation [47].

In this thesis, a DG scheme is used to discretize the equations that govern the fluid flow due to several reasons; in particular: DG can be made arbitrarily high order (provided a properly conditioned basis is constructed), it does not require a cumbersome stabilization for convection, it can deal with complex geometries using simplices (although Cartesian meshes or hybrid meshes can also be used), it has a compact stencil, and, due to its variational formulation, strong theoretical results for error estimates are available [92]. Furthermore, DG schemes are suitable for *hp*-adaptation [250, 39] and can easily deal with hanging nodes [92, 39].

This thesis uses an HDG scheme as a basis for the flow discretization. This par-

ticular form of DG scheme was originally proposed by Cockburn *et al.* [47] for the solution of elliptic equations and scalar convection-diffusion-reaction equations [46], and recently extended to a variety of problems by Nguyen, Peraire, Cockburn and some of their collaborators. These include: linear and nonlinear scalar convection-diffusion [167, 168], time-harmonic Maxwell's equations [171], continuum mechanics and elastodynamics [166, 170], incompressible Navier-Stokes [169], compressible Euler and Navier-Stokes [180], shock capturing [165] and Reynolds-averaged Navier-Stokes equations combined with the Spalart-Allmaras model [158].

The HDG scheme is of mixed-hybrid type and hence three sets of variables are solved for: the solution, the gradient of the solution, and an approximation to the value of the solution at the interfaces between elements (usually referred to as the trace). Despite the fact that HDG introduces extra unknowns, the special structure of the problem, that can be written as a parametrization of the solution and gradient as a function of the trace, allows for a static condensation step in which only the traces communicate with the neighboring elements yielding less degrees of freedom and sparser matrices than other DG schemes [100].

### 1.3.2 Shock capturing for high order methods

Many cases of interest in applied aerodynamics deal with compressible flows in which shock waves appear. The simulation of these flows using low order methods is currently a mature technology, based on well established strategies like the use of Riemann solvers (see the monograph by Toro [237]) or artificial viscosity (see the JST scheme by Jameson, Schmidt and Turkel [108]). Unfortunately, these strategies cannot be directly extended to high order methods, and more sophisticated techniques are required. These can be grossly categorized in three groups as discussed next.

#### High Order Limiting strategies

These schemes are based on extending the idea of limiters to high order polynomials. The first efforts in this front correspond to the ENO and WENO discretization

schemes [89, 142, 98] (see Section 1.3.1), where the stencil of a FV reconstruction is chosen to minimize oscillations around discontinuities. In parallel to this work, Cockburn and Shu developed their Runge-Kutta discontinuous Galerkin (RKDG) scheme for hyperbolic conservation laws [48, 50], based on a limiting of the high order representation of the solution inside each element. Building upon this work, Krivodonova [124] proposed a way to detect shocks based on jumps at the interfaces between elements, so that limiting only takes place in the vicinity of the shock. Later, the same author presented a limiting strategy based on monitoring the high order coefficients of the solution [123]. The main disadvantage of limiting has to do with the fact that most limiters are non-smooth and hence hard to combine with implicit time-stepping techniques. Furthermore, they are designed so that high order modes are suppressed when the limiter is active, which can be troublesome in the case of finite element discretizations, as this effectively diminishes the dependency of the residual on the high order coefficients.

#### Stabilized formulations for DG

A stabilized DG scheme based on a modified test space has recently been proposed by Demkowicz and Gopalakrishnan, who named it the discontinuous Petrov-Galerkin method (DPG) [62, 63, 64]. The idea behind DPG is to compute the test functions amongst a space of candidates so that the highest possible stability (measured by the inf-sup constant) is achieved. Their original intent was to recover optimal convergence for convection dominated problems, however, it took little time for it to be applied to shock capturing by the same authors [42, 41]. Based on this, Moro, Nguyen and Peraire reformulated the DPG framework as an optimization problem in which local conservation can be enforced by means of a constraint, and applied it to the local problem of the HDG scheme. This yielded the hybridizable discontinuous Petrov-Galerkin scheme (HDPG) [159, 160].

The complimentary approach of using the trial space to achieve stability has also been proposed in the literature. In particular, Huerta, Casoni and Peraire [101] de-

veloped a scheme in which the space of the solution is a blend between a high order polynomial and a piecewise constant representation in a sub-mesh inside each element. Stability is achieved in this case by means of the jumps introduced at the interfaces of the sub-mesh. The blending is driven by the resolution indicator of Persson and Peraire [188]. A more general approach has been recently presented by Nguyen *et al.* [173] in which the sub-mesh and the order of approximation within it can be arbitrarily chosen.

### Artificial Viscosity

The use of artificial viscosity to stabilize the numerical computation of shocks can be traced back to the work of Von Neumann and Richtmeyer [247] in the 1950's, who proposed to add enough viscosity in the discretization so that discontinuities are spread out and can be represented within the available resolution.

One of the first applications of artificial viscosity to DG was due to Bassi and Rebay [24]. It consisted of an extra term that resembled the integration by parts of an elliptic operator, scaled by a viscosity dependent on the residual of the Euler equations. No inter-element contributions for the elliptic operator were included, which helped maintain the stencil small. Similar approaches with different scalings for the viscosity were later used by Baumann and Oden [27], Hartmann and Houston [92] and Hartmann [90]. Later, Bassi and Rebay revisited their original idea and proposed a more advanced strategy in which artificial dissipation is only introduced in the direction of the pressure gradient, controlled element-wise by the jump between the internal flux and the common interface flux on a face [22]. Only very recently, Hartmann proposed a similar approach to take the anisotropy of the mesh into account [91].

In general, the aforementioned strategies can pose serious trouble for the convergence and consistency of the scheme. This motivated the use of other techniques based on the proper discretization of the viscous fluxes, for which an indicator of the shock location was required. To this end, Persson and Peraire developed their resolution indicator [188], based on measuring the decay of the higher order terms in the repre-

sentation of the solution, and using an analogy to Fourier coefficients as a criterion to select shock affected regions. Later, Klockner *et al.* [116] carefully analyzed the behavior of this sensor for a variety of cases and proposed a couple of strategies to filter the decay coefficients and avoid spurious shock detection.

In their original work, Persson and Peraire [188] applied a constant amount of viscosity in each flagged cell to stabilize the solution, and showed results for sub-cell shock resolution. A later study by Barter and Darmofal [20, 21] proved how a piecewise constant viscosity field can introduce spurious oscillations on the gradient of the solution, as well as the solution itself. To alleviate this, they proposed to use a linear reaction-diffusion PDE to smoothly propagate the viscosity field, using the sensors by Persson and Peraire [188] or Krivodonova [124] as a source term. A simpler way to achieve this was proposed by Klockner *et al.* [116] based on a continuous piecewise linear reconstruction of the viscosity in 1-D and by Persson [186] in 2-D and 3-D. When marched in time using an implicit scheme, this reconstruction widens the stencil of the discretization increasing the computational cost. Nevertheless, it is a reasonable choice for time-accurate shock propagation simulations [186] in which explicit time-stepping schemes can be used.

The resolution indicator of Persson and Peraire [188], or the jump indicator of Krivodonova [124] are based on the mathematical properties of the DG scheme. However, other strategies based on the physics of the Euler or Navier-Stokes equations have also been proposed in the literature.

The quintessential example of physics based shock capturing would be the JST scheme [108], that uses the pressure gradient to drive the application of viscosity. In the context of LES simulations of compressible flows, Ducros *et al.* [72] modified the JST scheme to prevent the application of viscosity to vortical structures using a sensor that relates dilatation ($\nabla \cdot \mathbf{v}$) to vorticity ($\nabla \times \mathbf{v}$) and is zero when the second dominates. Later Bhagatwala and Lele [28] used a sensor for the dilatation alone combined with the sensor by Ducros *et al.* to control the bulk viscosity term in the shock capturing formulation by Kawai and Lele [113]. The later is based on filtering

high order derivatives of the solution field to smooth out high frequency oscillations in compressible DNS simulations.

Based on this work, Premasuthan and Jameson [194] proposed a shock capturing strategy for the Euler equations combined with a spectral difference discretization, using only the dilatation part of the sensor proposed by Bhagatwala and Lele [28]. Inspired by this, Nguyen and Peraire [165] took advantage of the availability of gradients of the solution in HDG, and proposed an analytical function of the dilatation as a measure of artificial viscosity.

Very recently, Guermond *et al.* [87] proposed an artificial viscosity method that can be applied to any conservation law for which an entropy pair exists, applying viscosity in regions of entropy production. Their implementation requires a comparison with the average entropy in the domain, this being a non-local operation that requires the use of explicit time-stepping. This strategy was recently applied to high order methods by Abbassi *et al.* [3].

In this thesis, an artificial viscosity model is proposed that represents a modification of the one by Nguyen and Peraire [165] to ensure grid independence and symmetry across the shock wave.

### 1.3.3 Application of high order methods to the simulation of turbulent flows

Most of the flows of interest in the aerospace industry contain turbulent regions that can be the leading order effect in the performance of the system under study. In short, the main effect of turbulence is an enhanced mixing of transported properties, which can be beneficial or detrimental depending on the application. In the field of applied aerodynamics, turbulence is usually associated to higher viscous drag, which is generally considered a detrimental effect, however, it also delays flow separation due to adverse pressure gradients, which can be beneficial in certain situations. Several different techniques are available to simulate turbulent flows with different levels of

fidelity that correlate positively with the computational cost.

### Turbulence modeling

The highest fidelity simulations of turbulent flows are the so-called Direct Numerical Simulation techniques (DNS), where all the relevant length scales are resolved using the Navier-Stokes equations, and the only modeling comes through the constitutive equations for the gas. Even in current supercomputers, DNS simulations take an enormous amount of time and storage [106] and are only used to support fundamental research on very simple geometries [156]. The most successful DNS codes use a high order spatial discretization suitable for Cartesian domains such as compact differences [138] or spectral methods [86] combined with a high order explicit Runge-Kutta time-stepping scheme or fractional step methods [115].

The next level of fidelity would be the so-called Large Eddy Simulation techniques (LES), pioneered by Smagorinsky for weather prediction [221]. The assumption behind LES, formally proposed by Kolmogorov [118], is that the largest eddies are the ones that depend on the geometry and carry most of the energy, while eddies below a certain scale can be considered isotropic, hence, amenable to modeling. The LES formulation is based on two steps: a filter step in which sub-grid scales are removed from the unsteady Navier-Stokes equations and an evaluation of a sub-grid scale model that accounts for the dissipation of the small eddies. A comprehensive description of LES can be found in [210]. A variety of spatial discretizations are used for LES, favoring high order techniques such as Compact Differences [246], Flux Reconstruction [245] and DG [240]. Time discretization is usually carried out by an explicit or implicit high-order Runge-Kutta scheme. In the implicit case, the dissipation associated to the time discretization plays the role of the sub-grid model, and the filtering step can be ommitted.

The lowest level of fidelity in turbulent modeling is the so-called Reynolds-averaged Navier-Stokes equations [203] (RANS), based on the assumption that turbulence is a chaotic fluctuation of the state of the flow that can be decomposed in a steady

component ($\bar{\mathbf{u}}$) and a zero mean fluctuating component ($\mathbf{u}'$). A formal averaging procedure in time then yields the governing equations for the mean state, that contain some extra terms composed of averages of the fluctuation: $-\overline{\mathbf{u}'\mathbf{u}'^T}$, usually referred to as the Reynolds stress tensor, that needs to be modeled in some way to close the system.

Some models for this compute all terms in the tensor either through an algebraic relationship [248], or by solving an associated PDE representing a transport equation [135] or some other operator [73]. These are usually referred to as Reynolds stress models or RSM. It is common for these models to solve an extra transport equation for the turbulent kinetic energy or dissipation rate in order to properly scale the magnitude of the turbulent stresses [135]. RSM are considered to be the most complete (and expensive) RANS models and are still the subject of academic research [75].

The modeling can be simplified using Bousinnesq's hypothesis [29]. This states that the Reynolds stresses are proportional to the strain rate tensor of the averaged flow, with a constant of proportionality $\nu_t$, also known as the eddy viscosity. The use of this assumption reduces the modeling requirements from the elements of a tensor to a scalar. To compute the eddy viscosity, several alternatives exist, ranging from simple algebraic equations (Baldwin-Lomax model [19], Johnson-King model [19]) to transport PDEs for one or two quantities that are combined in an algebraic way to yield $\nu_t$ (e.g. the SA model by Spalart and Allmaras [223], the $k-\epsilon$ model by Launder and Sharma [136], the $k-\omega$ model by Willcox [256] or the SST model by Menter [153]). The later group is by far the most popular in industrial applications because it reduces the number of extra unknowns in the problem compared to RSM while being rich enough to model most of the turbulence effects. A very thorough description of all these models can be found in the monograph by Willcox [257].

The mesh resolution required for RANS is orders of magnitude smaller than for DNS or LES. Hence, the RANS simulation of complete geometries is feasible with modest computational resources. The discretization is usually based on second order finite volumes on unstructured grids or equivalent finite difference formulations using

overset grids.

The RANS equations are steady in nature, however, due to the inherent nonlinearity of the closure model, time-stepping is usually required to obtain a solution. For this, implicit or explicit schemes can be used combined with techniques to accelerate the evolution to steady state, like dual time-stepping or pseudo transient continuation.

### RANS solutions using High Order Methods

The solution of classical turbulence models like SA or $k-\omega$ using high order methods has been identified by the high order community as one of the key indicators for the maturity of these technologies [251]. However, this is not a trivial task for several reasons. First, RANS models are highly nonlinear and can be hard to converge in certain regimes, even using low order schemes. Also, RANS models present fairly stiff features such as a thin turbulent/laminar interface in SA or the near wall behavior in $k-\omega$ that do not favor the approximation with a high order polynomial. Finally, most RANS models are unstable for negative values of the eddy viscosity, which can produce a sudden blowup of the computations.

For all these reasons, the solution of RANS models using high order methods lacks robustness. The usual failure mode for these computations is an oscillation of the high order representation of the solution, that induces negative values of the eddy viscosity, which in turn makes the flow and the model unstable. Said oscillation is usually triggered by lack of resolution and whether or not it yields a negative value cannot be controlled *a priori*. This is not the case for low order schemes where clipping can be easily implemented. The use of such clipping might be one of the reasons why the issue of negative eddy viscosity is seldom discussed in the papers where the model is presented.

In spite of this, several groups have combined high order methods and RANS models by proposing changes to the models and the solvers. In particular, the first application of a high order method to RANS was due to Bassi and Rebay in 2005 [23], and consisted of a logarithmic formulation of the equation for $\omega$ in the $k-\omega$ model

(inspired in the work by Ilinca [105]) combined with a realizability constraint in terms of $\omega$.

Later, Nguyen, Peraire and Persson [172] used the resolution indicator of Persson and Peraire [188] to apply artificial viscosity in the SA equation. This sensor naturally targeted the oscillations in the SA working variable at the edge of the boundary layer and introduced enough viscosity to ensure a smooth transition to freestream values. A similar approach was later introduced by Oliver and Darmofal [176] who used the PDE based artificial viscosity of Barter and Darmofal [20, 21] and an indicator based on the jump in the SA working variable at the interface between elements. This reduced the dependency of their adaptive grid refinement strategy on the resolution at the edge of the boundary layer, and also improved the convergence.

Almost at the same time, Landmann *et al.* [132] implemented SA and $k-\omega$ on a DG solver using a cumbersome limiting strategy based on searching for local minima and clipping with a linear approximation if negative eddy viscosity was detected.

At that point, it seemed like classical RANS models would need to be reformulated if high order methods were to be used. Several steps in that direction were subsequently taken by a few different groups. The first modification of the SA model for high order methods was developed by Allmaras and first appeared in the PhD thesis of Oliver [175, 177]. It consisted in a new definition of the closure functions in the production term, the destruction term as well as the turbulent diffusion coefficient to ensure that patches with $\tilde{\nu} < 0$ were not energy stable. A revised version of this model was later published by Allmaras, Johnson and Spalart [10].

In the meantime, Moro, Nguyen and Peraire [158] proposed a modification of the SA equations based on a smooth clipping using an entropic regularization function, which made the SA working variable behave like a passive scalar subject to advection and diffusion when $\tilde{\nu} < 0$. These modifications were combined by Burgess and Mavriplis [34] who claimed a strong improvement in robustness compared to the standard SA model [223] or the modification by Oliver and Darmofal [177] alone.

Only very recently, another implementation of the SA model using DG has been

proposed by Crivellini *et al.* [55, 54]. Their modified model is based on a hard clipping of the eddy viscosity as well as the source terms in the SA equation for negative values of the working variable, plus a reassessment of the definition of ratio of turbulent length scale to distance to a wall. Their results show a very significant reduction in the number of Newton steps to achieve convergence, despite the fact that the proposed modifications are non-differentiable. A less aggressive approach has been followed by Drosson *et al.* [71, 70] in which the hard clipping is applied to the working variable in SA before entering the different sources and closure functions.

Finally, Chaurasia [44] took a step towards unifying all these versions into one that contained the most important findings. This version is the one that will be used for all the results presented in this thesis.

As mentioned in the beginning of this section, one of the challenges of solving RANS models with high order methods is their nonlinearity, for which an iterative procedure (usually a variant of Newton's method with time relaxation) is required. This iteration might fail during the transient, even for cases in which the mesh is adequate or the model has been properly modified. To prevent this, most groups use Pseudo-Transient Continuation (PTC) [92, 54, 117] to embed the nonlinear solve in a time relaxation. As shown by Modisette [155], PTC alone might not be enough for RANS and a line search should be implemented. Very recently, Ceze and Fidkowski [40] have revisited the topic and proposed the so-called Constrained-PTC (CPTC), in which physicality constraints can be included in the search for an update direction, avoiding those that lead to non-physical states.

In this thesis however, the combination of adaptivity and a high order version of the model produces a significant reduction of the stiffness of the system, which can then be tackled using a simple time-stepping scheme based on a BDF1 formula.

### 1.3.4 Transition prediction in RANS solvers

All the RANS models available in the literature have been devised with fully turbulent flows in mind. In those cases, RANS can produce accurate results for engineering

standards, all the way to moderate separation at a small fraction of the cost of other options such as LES or DNS.

The story is different when transition to turbulence is an important effect, since RANS models are not designed to take it into account, and hence it has to be prescribed in one way or another. This process is critical to obtain good predictions in the flow over wings at low Reynolds number [240], the design of laminar airfoils [199], or the analysis of high lift wing sections [241, 185], to name a few cases.

### Transition to turbulence

For the kinds of wall-bounded shear flows of interest here, it is generally accepted that transition happens when perturbations enter the laminar boundary layer (receptivity) and grow past the neutral point (linear amplification regime) until they reach a certain size and break down into chaotic structures (secondary/nonlinear instability) that yield turbulence. This process is usually referred to as *natural* transition and has been thoroughly documented and reviewed in the literature [214, 200, 12, 13, 213]. In some instances, the receptivity and linear amplification regime can be skipped, provided the external perturbations are big enough to produce the nonlinear breakdown, a phenomenon known as *bypass* transition [157, 225].

The boundary layer receptivity, first described by Morkovin [157], concerns the mechanism by which an external perturbation induces an unstable wave in the boundary layer. In principle, perturbations can be split into acoustic waves traveling at the speed of sound (relative to the fluid) and vorticity/entropy perturbations that travel with the fluid. Since their associated wavenumber does not need to be tuned to the unstable wavenumber from the linear stability analysis, some short of "wavelength conversion mechanism" is required [85, 212]. The conversion mechanism can depend of many factors like Reynolds number, wall curvature, sweep angle and roughness, as well as the perturbation strength and shape, which can be optimized to extract the most unstable case: e.g., using adjoint methods [260]. The understanding of receptivity is still an open question [13, 212] and is usually ignored in most practical

transition prediction methods despite its importance in setting the initial conditions for the subsequent stages [13].

The linear amplification regime deals with the growth of a given perturbation past the neutral point. After 50 years of development, a very solid theory is available to predict it, mostly based on local linear stability theory [12, 13, 145, 201]. The point of departure of local stability theory is the assumption that perturbations ($u'$) are complex exponentials in both time ($t$) and the coordinates parallel to the wall $(x, z)$ while modulated by an unknown function in the normal direction ($y$):

$$u' = u(y) \exp\left(i(\alpha x + \beta z - \omega t)\right). \tag{1.1}$$

This perturbation is introduced in the linearized Navier-Stokes operator assuming parallel flow to yield a system for the eigenvector $u(y)$ and an eigenvalue, provided enough parameters are fixed, e.g., combinations of real and imaginary parts of $\omega$, $\alpha$ or $\beta$. In boundary layer stability analysis, it is customary to assume disturbances grow in space instead of time (the so-called spatial theory), which forces $\omega$ to be real. In the particular case of 2-D flows, the local stability analysis equations reduce to the well-known Orr-Sommerfeld equation, that describes the evolution of Tollmien-Schlichting (TS) waves [214]. In 3-D flows, another unstable wave can appear due to the crossflow (CF) profile that is almost orthogonal to the streamline direction [213, 201] and admits solutions with zero frequency, also known as stationary waves or CF0, that behave differently than regular traveling CF waves (see discussion in Crouch and Ng [57]).

Due to the assumption of parallel flow, local stability theory does not take into account certain effects (e.g. curvature) that can be significant in some cases. To include these in the analysis, the parabolized stability equations (PSE) are solved [96], which include dependencies on $x$ in both $\alpha$ and $u$. When this modified ansatz is introduced in the linearized Navier-Stokes equations, one obtains a PDE that can be marched in $x$ from initial conditions. The results by Herbert [96] showing a better agreement with experiments than local theory, at the expense of solving the PSE,

motivated the development of the NASA's LASTRAC code [43] to efficiently solve the PSE equations.

Once the perturbations have grown to a certain value, nonlinear effects start to come into play [95]. So far, only the initial stages of the nonlinear interaction are properly understood. The study of these is based on nonlinear-PSE or DNS [213, 59, 14]. The later stages of transition are beyond the reach of current theories and can only be tackled using DNS. Despite this, some authors argue that for wall bounded flows in low disturbance freestream, the linear amplification region takes most of the distance before transition occurs [13, 163], hence, nonlinear breakdown can be safely assumed to occur instantaneously.

### Transition prediction using the $e^N$ method

The consensus amongst the experts in the field is that a simple and universal model for transition to turbulence that unifies all the phases of the phenomenon is unlikely to be developed [13, 43, 95, 59, 14, 243]. In addition, this panel of experts unanimously suggests that the best approach for practical applications would be the use of linear theory for natural transition of TS and CF waves combined with empirical criteria for other phenomena such as bypass transition or attachment line transition.

In particular, regarding natural transition, most of the experts sanction the use of the $e^N$ method by Smith and Gamberoni [222] and Van Ingen [242] to predict the location of transition. The main assumption of the method is that transition occurs when a certain amplification factor ($N = N_{\text{crit}}$) is reached for the most amplified wave:

$$N = \log(A/A_0) = \max_{\omega,\beta}\left[\int_{x_0}^{x} -\alpha_i(\omega,\beta,x)dx\right], \tag{1.2}$$

where $x$ is assumed to be parallel to the edge streamlines, $x_0$ is the first point where the wave is amplified (the neutral point) and $N_{\text{crit}}$ can be computed using Mack's criterion as a function of the turbulence levels [144]. The evaluation of Eq. 1.2 is not straightforward due to the dependency of the amplification rate $\alpha_i$ on the wave param-

eters $\omega$ and $\beta$. To circumvent this problem, several techniques have been proposed. As summarized by Arnal [14], these are:

- Envelope method: based on switching the maximization operator and the integral so that integration is carried out using the most unstable wave at each location along the streamline:

$$N = \int_{x_0}^{x} \max_{\omega,\beta} \left[-\alpha_i(\omega, \beta, x)\right] dx. \tag{1.3}$$

  This method tends to generate higher levels of $N$, thus, predicting transition in a conservative manner.

- Envelope of Envelopes method: based on tracking the amplification factor of a discrete number of frequencies $\omega$, and spanwise wavenumbers $\beta$ and reconstructing the maximization operator.

- $N_{\mathrm{TS}} - N_{\mathrm{CF}}$ Criterion: based on tracking two amplification factors separately, one for TS waves and another one for CF waves, and using an empirical curve in the $N_{\mathrm{TS}} - N_{\mathrm{CF}}$ plane to determine transition.

$$N_{\mathrm{TS}} = \max_{\omega} \left[ \int_{x_0}^{x} -\alpha_i(\omega, 0, x) dx \right], \qquad N_{\mathrm{CF}} = \max_{\omega,\beta\in\mathbb{R}} \left[ \int_{x_0}^{x} -\alpha_i(\omega, \beta, x) dx \right]. \tag{1.4}$$

Independently of this, the computation of $\alpha_i$ inside the integral in Eq. 1.2 or the maximum growth rate that appears in Eq. 1.3 can be carried out using the local or PSE approach. This in turn implies the solution of an eigenvalue system or a PDE along the integration trajectory, which can be costly and cumbersome. To avoid this, several authors have proposed simplified methods in which these quantities are computed with the help of a database. These methods are usually referred in the literature as approximate or database $e^N$ methods. A non-exhaustive list of them would include:

- Envelope method of Drela and Giles for TS waves [69]: an extension of the work by Gleyzes *et al.* [84], based on a linear fit to the spatial amplification envelopes

of self-similar flows. This method provides the amplification rate and the critical Reynolds number as a function of $Re_\theta$ and the shape parameter $H$.

- Two parabola approximation by Arnal for TS waves [11]: based on approximating the growth rate as a function of $Re_{\delta^*}$ using two parabolas whose parameters depend on $\omega$, $M_{\text{edge}}$ and $H$.

- Database of Stock and Degenhart for TS waves [227]: based on a database with three inputs; $Re_{\delta^*}$, $H$ and $\omega$, to recover the growth rate.

- Database method of Drela for TS waves [68]: based on an an interpolation procedure using splines to obtain the growth rate as a function of $Re_\theta$, $H$ and $\omega$, combined in a judicious way to limit the bounds of the database. This model was extended to compressible boundary layers by Sturdza [232].

- Two parabola approximation by Casalis and Arnal for CF waves [38]: which is an extension of the work by Arnal [11] where the parameters in the parabolas depend on the mean velocity and the shear stress at the inflection point.

- Database of Dagenhart for Stationary CF waves [58]: based on a table of growth rates as a function of three parameters: the crossflow Reynolds number, the crossflow shape factor and the crossflow velocity ratio (defined in a special way based on pointwise quantities [58]). This model was extended by Sturdza [232] to add compressibility effects.

- Machine learning algorithms for mixed CF and TS waves: like the ones proposed by Couch *et al.* [56] based on training a neural network using the Falkner-Skan-Cooke profiles and linear stability analysis, or the recent developments by Rajnarayan and Sturdza [198] based on kriging a database composed of swept wing simulations of a variety of airfoils.

In addition to this, the $e^N$ method can be combined with empirical criteria for bypass transition (e.g., the Abu-Ghannam/Shaw criterion [4]) and attachment line

transition (e.g., Poll's criterion [88]) to augment the range of applicability of the transition prediction module.

In practice, a properly calibrated $e^N$ method can be very accurate, even though certain processes like receptivity and nonlinear breakdown are not taken into account. Furthermore, the cost is almost negligible compared to PSE or more advanced tools like LES or DES.

Of particular interest in this dissertation is the coupling of an $e^N$ method with a RANS solver to predict transition, thus combining the reduced costs of the former with the generality of the later, and producing a scheme with similar capabilities as LES at a reduced fraction of the cost.

### The $e^N$ method in RANS solvers

By the end of the 90's, finite volume technology was mature enough so that fully turbulent RANS solutions were routinely run to completion in the industry [109]. However, transition prediction capabilities in such solvers were non-existent, which made them unsuitable for certain tasks like natural laminar flow design or high lift configurations.

The very first attempt at this was due to Radespiel *et al.* [197] from the German DLR in 1991, who showed results based on extracting the boundary layer profiles from a RANS solver to compute integral boundary layer quantities that drove a database $e^N$ method for transition. In that work, the authors identified the following three challenges as critical for the success of the methodology: 1) the accurate representation of the boundary layer profiles, 2) the identification of the boundary layer edge, and, 3) the iterative scheme used to couple the transition prediction with the RANS solver.

Three years later, two papers by Schneider and Ewald [215] from the Technical University of Darmstadt and Kusunose and Cao [131] from The Boeing Company, continued this line of research. This pioneering work was followed five years later by a seemingly independent paper due to Stock and Hasse [229], from DLR. In it, the authors highlighted the role of adaptation in tackling the first two aforementioned

challenges. Their approach was based on the extension of their diagnostic function for turbulent flows [228] to measure the boundary layer thickness in laminar flows (although this had already been proposed by Schneider and Ewald [215]). Based on this measure of the boundary layer thickness, an adapted grid was generated once and used for the whole iterative process. As in the previous paper by Radespiel *et al.* [197], the authors found a very strong interaction between the $e^N$ method and the RANS solver that required a variety of numerical tricks like an under-relaxation of the transition point update or the initialization at laminar separation and extrapolation of $N$ past separation. In a subsequent paper, the same authors proposed a way to alleviate this based on the use of intermittency functions [230].

From then on, a significant body of literature was produced mostly from groups in Europe (DLR and ONERA) trying to add transition capabilities to a variety of codes (TAU, FLOWer, elsA) used at Airbus Industries. In America, most of the published results are linked to academia[1]. Going through this literature it is easy to notice a dichotomy in the workflow to predict transition between groups that use an intermediate boundary layer solver and groups that do not.

The idea of using the pressure from the RANS solver to drive a boundary layer code that generates the boundary layer profiles was first proposed by Kusunose and Cao [131], and also mentioned in the conclusion sections of the seminal paper by Stock and Hasse [229]. It took three years before the authors published any advance in this front [226]. In the meantime, Brodeur and van Dam [30], from UC Davis, implemented transition in an incompressible Navier-Stokes code using a boundary layer code to compute amplification factors into the turbulent regions. The 3-D extension of these ideas started with the efforts of the EUROLIFT Consortium [185, 238] in which a variety of boundary layer codes (conical flow, infinite swept wing or fully 3-D) where tested on high-lift devices without coupling to RANS. In parallel, Krumbein, from

[1] The small number of publications about this topic coming from US manufacturers indicates that most of these efforts are proprietary. In particular, in the conclusions section of the preliminary report of the RAATraP project [59] by the Lockheed Martin Corporation, the coupling with a RANS solver is briefly discussed as ongoing work without showing any results.

DLR, coupled a swept wing boundary layer code, a database $e^N$ method and the TAU solver in 2-D [126] and later extended the FLOWer solver in a similar fashion in 3-D using a $N_{\mathrm{TS}} - N_{\mathrm{CF}}$ criterion [127]. The combination of this methodology with adjoint techniques for optimization of natural laminar flow wings was presented by Lee and Jameson [137].

The use of an intermediate boundary layer code enables the computation on non-adapted grids and also accelerates the iterative process since pressure convergence is usually less affected by grid resolution. Furthermore, no identification of the boundary layer edge is required. As expected, there are also some drawbacks, mainly, the limitation of the boundary layer validity to attached flow regions, which forces the point of laminar separation to be treated as turbulent [226, 126, 127]. This feature ultimately prevents the simulation of laminar separation bubbles using this methodology.

At the same time, other efforts stayed close to the original approach by Radespiel [197] and used the boundary layer quantities provided by the RANS solver directly. An early example of this would be the work by Nebel *et al.* [163] using the TAU code and a database method for CF and TS waves, followed by the work of Krimmelbein *et al.* [122]. In a similar spirit, Mayda [149] coupled an $e^N$ criterion with a 2-D compressible code, in a thesis supervised by Prof. van Dam at UC Davis. Later, Cliquet, Houdeville and Arnal [45] proposed another criterion to detect the edge of the boundary layer based on shear and vorticity, combined with an analytical criterion for transition of TS waves, and implemented it in ONERA's elsA code. Only very recently, Rashad and Zingg have applied a similar approach to an in-house solver wrapped around an optimization loop applied to natural laminar flow design [199].

The extraction of the boundary layer profiles from the RANS solution simplifies the coupling between modules and avoids the task of implementing a variety of boundary layer codes (for conical flow, swept wing, etc.). Furthermore, it allows computations past separation, which increases the robustness in the initial iterations. As a drawback, the measure of the edge of the boundary layer is required using one of the indicators described above (shear-vorticity [45], diagnostic function [215, 228] or isentropic re-

construction [197]). Also, fairly strict requirements are imposed on the initial mesh in terms of resolution (between 40 and 70 points in the boundary layer depending on the reference [197, 184]) and structure so that accurate boundary layer profiles can be easily extracted.

Recently, a significant effort by the groups in Germany was devoted to unifying both approaches into the TAU code [128, 129, 121], wrapping the transition prediction capabilities into a module that can use the RANS solution to either extract velocity profiles or drive a boundary layer code to get them, and pass them along to either a local stability analysis tool (LILO[2]) or a database method to get the growth rates. This tool has been used by Krimmelbein and Krumbein [120] to compute transitional flows over transport configurations as well as by Probst and Radespiel [196] to compute low Reynolds number airfoils and laminar nacelles with significant success.

In this thesis, the transition prediction is incorporated in the solver using a novel simultaneous coupling of the $e^N$ module and the RANS discretization, that takes advantage of the adaptivity and structure of the mesh.

### Other strategies

Needless to say that the $e^N$ method is not the only way to include natural transition in a RANS solver, as other tools have been developed in the literature.

In particular, early studies on the subject suggested the use of the so-called analytical transition criteria, based on the assumption that transition happens once a certain algebraic inequality, that is a function of the local boundary layer properties, is satisfied. In essence, this is equivalent to the assumption that the amplification factor (rather than the growth rate) can be correlated to the local properties of the boundary layer. A variety of such models exist, such as the GH criterion for laminar separation, the C1 criterion for CF waves or the AHD criterion for TS waves, just to name a few. A careful discussion of their limitation as well as further references can be found in the review paper by Arnal [14].

[2]Propietary code of Airbus Deutschland GmbH. No references available in the open literature.

Another possibility is to use some advanced form of RANS models that are sensitive to transition, usually through the introduction of extra conservation laws to predict it. An example of these would be the laminar kinetic energy model by Walters and Leylek [249] or the $\gamma - Re_{\theta_t}$ model by Langtry and Menter [133, 134], which was combined with the SA model by Medida and Baeder [151]. Based on these ideas, Coder and Maughmer [51] proposed a method based on the intermittency and amplification factor which can be thought of as a database $e^N$ in the whole fluid domain. The same model was later refined and reduced to a single PDE for the amplification factor by the same authors [52]. Very recently, a correlation for crossflow in the $\gamma - Re_{\theta_t}$ model has been proposed by Medida and Baeder [152] which might enable 3-D computations, however, at the expense of non-local operations.

The solver presented here does not make use of any of these techniques but rather focuses on the well stablished $e^N$ method. However, the architecture of the solver would easily accommodate them if required.

### 1.3.5 Mesh generation and adaptation techniques for boundary layer flows

One of the most important tasks in CFD analysis is the development of suitable meshes for the problem of interest. In some instances, this can be a very labor-intensive task that requires a significant amount of user intervention and time. This trend is reinforced by the growth in computer power, that enables the analysis of more complex geometries, for which meshing becomes harder and requires more expertise.

The efforts of the meshing community towards reducing this burden have been phenomenal. A detailed account of all the techniques proposed is far beyond the scope of this document and can be found in monographs about the topic [236], as well as review papers like the one by Baker on meshing for CFD [18].

In what follows, the particular case of mesh generation for high Reynolds number flows is briefly discussed. One peculiarity of these flows is the presence of very thin

boundary layers in which the solution changes much faster away from a solid wall that along it. This disparity of length scales is due to the boundary conditions and roughly scales with $Re^{1/2}$ for laminar flows and $Re^{4/5}$ for turbulent ones, hence in the order of $10^3 - 10^5$ for typical flight configurations. The consensus in the meshing community is that these cases require anisotropic meshes where the elements are oriented so that their approximation properties are maximized with respect to the solution. A variety of tests that confirm this can be found in the results section of most of the papers discussed next.

The choice of mesh topology and element type is a more controversial one, especially in the case of finite volume and finite element discretizations, which are flexible enough to admit a variety of them. More precisely, the big argument is between structured meshes based on quadrilaterals and hexahedra and unstructured meshes based on simplices. The generation of one kind or another requires different techniques.

In the case of unstructured anisotropic mesh generation, some of the algorithms are based on modifications to the corresponding isotropic version. Examples of these would be the mapped Delaunay algorithm by Mavriplis [148] and some variants of the advancing front algorithm like the ones used by Peraire *et al.* [181] and Hassan *et al.* [93]. These algorithms require a preprocessing step in which a background mesh that contains the anisotropy information is generated. Another example of this kind would be the advancing-layers algorithm by Pirzadeh [192, 193] and the improved versions by Garimella and Shephard [82] or Martineau *et al.* [146], which also require a preprocess step to generate the growth direction in the boundary layer as well as the boundary layer thickness. In a completely different spirit, other authors have proposed algorithms that start with an isotropic mesh and apply operations like node insertion, node reconnection or node movement to evolve towards the desired anisotropy. Examples of these have been proposed by Park and Darmofal [178], Loiseille and Löhner [143], and Michal and Krakos [154].

In the case of structured meshes, two different techniques are commonly used according to Baker [18]. On the one hand, there is the multi-block approach in which the

user manually breaks the domain into a series of big quadrilaterals or cubes, each of which contains a structured Cartesian mesh. Said Cartesian mesh can be refined in the direction normal to the surface to achieve boundary layer anisotropy in a straightforward manner. On the other hand, there is the Chimera overset method (see Chapter 11 in Thompson [236]), based on individual grids for each element, e.g. wing, fuselage, payloads, etc., that overlap with each other. In Chimera, all the computational complexity is associated to geometric operations that detect intersection, clean the overlap geometry and compute the interpolation stencil to transfer the information from one grid to another. This technology was heavily used in the strand mesh generation by Katz *et al.* [112] based on a layer of prisms for the boundary layer and a Cartesian overset mesh for the external geometry. Only very recently, the Chimera approach has been extended to DG by Galbraith *et al.* [80].

In his review paper, Barker [18] argues that the best choice might be in the middle point, by using hybrid prism/simplex meshes. A variety of algorithms have been proposed to generate those kinds of meshes such as the ones by Nakahashi [161, 162], Kallinderis and Wardt [110], Sharov and Nakahasi [218] or Sahni *et al.* [211]. In general, all these algorithms start from a surface tessellation, extrude a mesh of prisms from it and fill the rest of the domain using any of the unstructured strategies presented above. The main difference between them is the way they deal with geometric singularities such as corners or intersections that can affect the robustness of the procedure. A slightly different approach is taken by Alauzet and Marcum [5] and involves the generation of the tetrahedral mesh before actually growing the prism layers.

As mentioned above, the goal of anisotropy is to maximize the approximation properties of the mesh with respect to the solution. Since the solution is not known a priori, a one shot anisotropic mesh might be overly conservative and involve too many degrees of freedom. It is thus common to use a partial solution as the driver for the mesh generation, in what is known as mesh adaptation. Without loss of generality, mesh adaptation can be divided intro three kinds of techniques: $h$-adaptation, $p$-adaptation and $r$-adaptation, depending on the kind of control over the discretization.

In the case of $h$-adaptation, the control is exercised over the element size and shape. For this, most $h$-adaptation strategies are based on a sequential approach in which: 1) a fully or partially converged solution on the current mesh is computed, 2) said solution is analyzed to yield anisotropy requirements (usually in the form of a metric), 3) the metric is passed along to the mesh generation, which produces a new mesh, that is used to repeat the process.

In this procedure, the critical step is the analysis of the solution which is commonly based on some measure of the error. A common choice for the later is to use derivative reconstructions to minimize the interpolation error either alone [183, 181, 234, 77] or combined with output error estimates as proposed by Fidkowski [76]. Only very recently, a different approach based on output error estimates (in particular Dual Weighted Residual or DWR) and element splitting and sampling has been proposed by Yano [259] for unstructured meshes or Ceze [39] for structured ones.

The process is similar in the case of $p$-adaptivity, except that now the approximation order inside each element can be varied depending on the smoothness of the solution there. In general, however, $p$-adaptivity is seldom found alone but rather in the form of $hp$-adaptivity [16]. This technique has seen heavy use in the DG community, where the implementation is relatively simple compared to CG schemes. In any case, the algorithm requires an error indicator to drive the change of approximation order, that can be based on extrapolation in a sequence of meshes [16], a resolution indicator [250], a post-processed solution [83] or a DWR type error estimate [39], amongst others.

Finally, in the case of $r$-adaptivity (also known as moving mesh methods), the control is exercised on the location of the nodes of the mesh, keeping its topology fixed. This process relies on the definition of a monitor function that is filtered through a PDE to yield the nodes location. The particular choice of these is discussed at length in the reviews by Budd, Huang and Russell [32, 99]. In the particular case of CFD, $r$-adaptivity has been applied to unsteady inviscid compressible flow problems, as well as free shear flows in 2-D by Tang and his collaborators [235, 65]. As common to $h-$

or $p-$ adaptivity, the update of the mesh is produced in between iterations of the flow solution.

All these instances of $r$-adaptivity are designed to be oblivious to the location of the flow features that they are trying to adapt to. This makes them reasonably general, but, at the same time, overly complicated for the case of wall bounded flows, in which the location (but not the extent) of the boundary layer is known ahead of time. However, there exist examples of $r$-adaptive methods in which this feature is taken advantage of, like the differential boundary layer solvers of Drela [67] and Allmaras [8].

Inspired by this line of work, this thesis presents a novel $r$-adaptation algorithm to generate conformal grids, in which the boundary layer thickness is the driver of the whole process. One of the novelties of the proposed approach is the simultaneous solution of the mesh adaptation and the RANS equations. This feature of the algorithm, that had never been applied outside the context of boundary layer solvers [67, 8], is crucial to increase the robustness of the iteration and the quality of the final solution, and ultimately enables the prediction of transition.

## 1.4 Thesis overview

This thesis dissertation describes the efforts in the development of an adaptive high order numerical scheme with transition prediction capabilities, as well as its extension to compressible flows that present shock waves. From these, the novel contributions of this thesis can be summarized into:

- A shock capturing strategy for high order methods based on the physics of shock waves that ensures the proper scaling of the artificial viscosity model.

- The development of an adaptive solver where the mesh is treated as an extra unknown of the problem that is driven by an indicator of the boundary layer thickness, and solved simultaneously with the flow equations.

- The analysis of the stability of the adaptivity in terms of convergence to the correct solution in the case of separated flows.

- The coupling of the adaptive solver to a transition prediction module, by means of a novel high order surface PDE solver for the amplification factor.

- The exercise of the limits of the solver in turbulent and transitional flows, with and without separation, as well as a preliminary demonstration of the capabilities of the solver to simulate unsteady transitional flows.

This thesis dissertation is divided as follows. First, Chapter 2 contains a partial description of the methodology focusing on the governing equations for the flow as well as the high order scheme used to discretize them, both of which are taken from the literature and do not represent a contribution of this thesis. Continuing, Chapter 3 focuses on the shock capturing method developed as part of this dissertation, focusing on the details of the formulation and its verification using 1-D and 2-D results. Next, Chapter 4 presents the adaptive method with special emphasis on the initial mesh requirement and what are the equations that govern its evolution, followed by a variety of verification cases in 2-D. The novel transition prediction module is discussed in Chapter 5, where the high order surface PDE discretization is described, and its coupling with the rest of the solver is tested in natural, separation induced, and unsteady transitional flows in 2-D. To finish, Chapter 6 presents some conclusions and future work.

# Chapter 2

# Methodology

This chapter presents the PDEs used to model the flow as well as the techniques used to discretize them in both space and time. In particular, the goal here is to briefly discuss the flow discretization procedure without entering in much detail, as this is a standard approach that has been extensively discussed in the literature review.

## 2.1 Governing Equations

First, the equations that govern the evolution of the fluid flow are described. For this it is necessary to distinguish between laminar and turbulent flows.

### 2.1.1 Navier-Stokes equations

In this thesis, the Navier-Stokes equations are used to simulate laminar viscous flows. In conservative form, the Navier-Stokes system reads:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_{\text{inv}} = \nabla \cdot \mathbf{F}_{\text{visc}}, \tag{2.1}$$

where:

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v_i \\ \rho E \end{bmatrix}, \qquad \mathbf{F}_{\text{inv}} = \begin{bmatrix} \rho v_i \\ \rho v_i v_j + P\delta_{ij} \\ \rho v_i H \end{bmatrix}, \qquad \mathbf{F}_{\text{visc}} = \begin{bmatrix} 0 \\ \tau_{ij} \\ \sum_{j=1}^{d} \tau_{ij} u_j + q_i \end{bmatrix}, \tag{2.2}$$

$$\tau_{ij} = \mu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\delta_{ij} \frac{\partial v_k}{\partial x_k} \right), \qquad q_i = \kappa \frac{\partial T}{\partial x_i}. \tag{2.3}$$

Here, $\rho$ represents the density, $v_i$ is the $i$-th component of the velocity, $E$ is the total specific energy and $H = E + P/\rho$ is the total specific enthalpy. The pressure ($P$), the density and the temperature ($T$) obey the ideal gas law:

$$P = \rho R T, \tag{2.4}$$

where $R$ is the specific gas constant of air. The coefficients $\mu$ and $\kappa$ are the dynamic viscosity and heat conductivity respectively. Here, $\mu = \mu(T)$ according to Sutherland's law:

$$\mu = \mu_{\text{ref}} \frac{T_{\text{ref}} + C}{T + C} \left( \frac{T}{T_{\text{ref}}} \right)^{3/2}, \tag{2.5}$$

where $C = 120K$ and the pair $(T_{\text{ref}}, \mu_{\text{ref}})$ denotes a reference temperature and viscosity. In this work, these are taken equal to the freestream values: $(T_\infty, \mu_\infty)$. The thermal conductivity $\kappa$ is related to the viscosity through the Prandtl number, that is assumed to be constant:

$$Pr = \frac{c_p \mu}{\kappa} = 0.72. \tag{2.6}$$

The Navier-Stokes equations are written in non-dimensional form using the following reference states:

$$\rho_{\text{ref}} = \rho_\infty, \tag{2.7}$$

$$\rho u_{\text{ref}} = \rho_\infty \|\mathbf{v}_\infty\|, \tag{2.8}$$

$$\rho E_{\text{ref}} = \rho_\infty \|\mathbf{v}_\infty\|^2. \tag{2.9}$$

Dropping the viscous terms on the right hand side of Eq. 2.1 yields the Euler equations that describe the evolution of inviscid compressible flows. These will be used in Chapter 3 to study the shock capturing model proposed as part of this thesis.

To close the problem, the correct boundary conditions need to be prescribed on the limits of the domain. These usually entail the proper prescription of inflow and outflow characteristics, as well as solid walls with and without slip. The specific details of these are discretization dependent and will be discussed in Section 2.2.1 once the numerical scheme is introduced.

### 2.1.2 Spalart-Allmaras turbulence model

In this thesis, the effect of turbulence on the flow field is modeled using the Reynolds-averaged Navier-Stokes equations together with Boussinesq's analogy for the eddy viscosity. In this way, the system described by Eq. 2.1-2.3 is only modified in terms of the viscosity and thermal conductivity that enter Eq. 2.3 so that the stresses are given by:

$$\tau_{ij} = (\mu + \mu_{\text{t}}) \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial v_k}{\partial x_k} \right), \qquad q_i = \left( \frac{\mu}{Pr} + \frac{\mu_{\text{t}}}{Pr_{\text{t}}} \right) \frac{\partial h}{\partial x_i}. \tag{2.10}$$

Here $\mu_{\text{t}}$ is the eddy viscosity, $Pr_{\text{t}}$ is the turbulent Prandtl number (take to be a constant $Pr_{\text{t}} = 0.9$) and $h = H - \frac{1}{2}\mathbf{v}^2$ is the thermodynamic enthalpy. Notice the vector of unknowns $\mathbf{u}$ now denotes the average density, momentum and total energy fields. The averaging process leading to these can be interpreted in a variety of ways depending on whether the average occurs in time (with finite or infinite time window)

or as an ensemble over a space of realizations (see Willcox [257]). Most of the cases computed in this thesis deal with stationary ergodic turbulence, in which case these averaging processes are equivalent. In any case, retaining the time dependent terms in the turbulence model as well as the RANS equations is useful for convergence purposes.

The use of Boussinesq's analogy requires a model for the eddy viscosity $\mu_t$ that appears in Eq. 2.10. This work uses a version of the Spalart-Allmaras turbulence model [223], which contains modifications [158, 10] that avoid stability problems when a high order discretization is used. The resulting model, as presented by Chaurasia [44], is based on the following PDE for the compressible form of the so-called working variable $\rho\tilde{\nu}$:

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \nabla\cdot(\rho\mathbf{v}\tilde{\nu}) = \rho(s_P - s_D) + \frac{1}{\sigma}\nabla\cdot\left[\rho(\nu+\psi\nu)\nabla\tilde{\nu}\right] + \frac{c_{b2}}{\sigma}\rho(\nabla\tilde{\nu})^2 + \frac{1}{\sigma}(\nu+\psi\nu)\nabla\rho\cdot\nabla\tilde{\nu}, \tag{2.11}$$

where the different terms on the right hand side of the equation are defined next. In particular, let:

$$\chi = \frac{\rho\tilde{\nu}}{\mu}, \quad \psi = \chi\left(\frac{\arctan(b\chi)}{\pi} + \frac{1}{2}\right) + c, \quad c = \frac{1}{2} - \frac{\arctan(b)}{\pi}. \tag{2.12}$$

Here, $\chi$ denotes a non-dimensional version of the working variable and $\psi$ denotes a regularized version of it, that approximates the non-differentiable operation $\max(0,\chi)$. Such regularization is controlled by the constant $b$ and the associated term $c(b)$. In addition, the production ($s_P$) and destruction ($s_D$) terms in Eq. 2.11 are defined as:

$$s_P = c_{b1}\tilde{S}\psi\nu\ , \qquad s_D = c_{w1}f_w\left(\frac{\psi\nu}{d}\right)^2\ , \tag{2.13}$$

and require the following auxiliary relationships:

$$S = \sqrt{2\Omega_{ij}\Omega_{ij}}, \qquad \bar{S} = \frac{\psi\nu}{(\kappa d)^2}f_{v2}, \qquad f_{v1} = \frac{\psi^3}{\psi^3 + c_{v1}^3}\ , \qquad f_{v2} = 1 - \frac{\psi}{1+\psi f_{v1}}, \tag{2.14}$$

$$\tilde{S} = 0.1S + (\bar{S} + 0.9S)\left(\frac{\arctan\left(b(\bar{S}/S + 0.9)\right)}{\pi} + \frac{1}{2}\right) + cS, \tag{2.15}$$

$$\bar{r} = \frac{\psi\nu}{\tilde{S}(\kappa d)^2}, \qquad r = r_{\lim} - (r_{\lim} - \bar{r})\left(\frac{\arctan(b(r_{\lim} - \bar{r}))}{\pi} + \frac{1}{2}\right) - c, \tag{2.16}$$

$$f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}, \qquad g = r + c_{w2}(r^6 - r)\ . \tag{2.17}$$

Once these are defined, the eddy viscosity that approximates the Reynolds stresses in Eq. 2.10 is given by:

$$\mu_t = \mu\psi f_{v1}\ . \tag{2.18}$$

To close the model, several constants need to be prescribed. In this case, the values used in the original SA model [223] are retained. These can be found in Table 2.1.

Table 2.1: Parameters of the Spalart-Allmaras model. Here $\kappa$ denotes the Von Karman constant, that should not be confused with the thermal conductivity.

| $c_{b1}$ | $c_{b2}$ | $c_{v1}$ | $\sigma$ | $c_{w1}$ | $c_{w2}$ | $c_{w3}$ | $\kappa$ | $r_{\lim}$ |
|---|---|---|---|---|---|---|---|---|
| 0.1355 | 0.622 | 7.1 | 2/3 | 3.2391 | 0.3 | 2 | 0.41 | 10 |

In addition, the regularization parameter $b$ needs to be defined. For all the results in this work, a value of $b = 100$ is used, as originally suggested by Chaurasia [44].

The system of PDEs that governs the flow is obtained by appending the modified version of the SA model (Eq. 2.11) to the RANS equations (Eq. 2.1-2.3 with the Reynolds stresses in Eq. 2.10). The system is written in non-dimensional form with the help of the same reference magnitudes as the Navier-Stokes equations, plus a reference value for the working variable equal to:

$$\rho\tilde{\nu}_{\text{ref}} = \mu_\infty. \tag{2.19}$$

The closure of the SA model is given by the boundary conditions, which depend on the character of the boundary. On a solid wall, the SA model requires the eddy viscosity to be zero, which is satisfied by setting $\rho\tilde{\nu} = 0$. Similarly, in the case of an inflow or far-field, it is customary to assume that the eddy viscosity asymptotes to a constant value of $\rho\tilde{\nu} = 3\mu_\infty$ for fully turbulent solutions and $\rho\tilde{\nu} = 0.1\mu_\infty$ for cases in which transition to turbulence is present. Finally, in cases where the flow is leaving the domain or an inviscid wall is simulated, it is customary to extrapolate the value of $\rho\tilde{\nu}$. The specific details of how these boundary conditions are implemented within the HDG framework are discussed in Section 2.2.1.

**Transition to turbulence**

The applicability of the SA model to transitional flows is possible with the addition of the appropriate mechanisms to prevent the growth of $\rho\tilde{\nu}$ (and hence the Reynolds stress tensor) in the laminar regions of the flow.

In the original SA model [223], such behavior is introduced through the so-called $f_{t2}$ term that modifies the equations so that $\rho\tilde{\nu} = 0$ is a local attractor of the problem. In the early stages of this thesis, a modified version of the $f_{t2}$ term proposer by Coder and Maughmer [52] was used which was found to be very unstable in combination with a high order discretization. The usual failure mode of this approach was a streamwise oscillation of $\rho\tilde{\nu}$ around the transition point that effectively took the solution out of the attractor and made the transition point creep upstream. In most instances, this instability induced the transition of the whole boundary layer.

In order to circumvent this issue, it seems mandatory to reduce the dependency of the suppression term on $\rho\tilde{\nu}$. For this, an intermittency factor $\gamma$ is used that effectively turns off the productions of eddy viscosity in regions where the flow should remain laminar. More precisely, the production term is modified as follows:

$$s_P = \gamma c_{b1} \tilde{S} \psi \nu. \tag{2.20}$$

This way of imposing transition is inspired in the work of Medida and Baeder [151]. More details on this can be found in Section 5.2.3.

### 2.1.3 Arbitrary Lagrangian-Eulerian formulation

One of the peculiarities of the adaptive solver developed in this thesis is the fact that the mesh and the flow are marched in time simultaneously. This prompts the use of an Arbitrary Lagrangian-Eulerian description (ALE) to reduce the conservation errors, avoid unphysical states due to the mesh movement, and decouple the mesh deformation from the flow field as much as possible.

In this thesis, the ALE formulation follows the notation introduced by Persson *et al.* [187]. In particular, the ALE formulation requires a mapping $\mathcal{G}$ from a reference domain ($\Omega_{\mathbf{r}} \in \mathbb{R}^d$) to the physical domain ($\Omega_{\mathbf{x}} \in \mathbb{R}^d$) denoted by:

$$\mathbf{x} = \mathcal{G}(\mathbf{r}, t),\ \mathbf{r} \in \Omega_{\mathbf{r}},\ \mathbf{x} \in \Omega_{\mathbf{x}}, \tag{2.21}$$

which is differentiable in the arguments. In particular, let:

$$\mathbf{G} = \frac{\partial \mathcal{G}}{\partial \mathbf{r}}\ , \qquad \mathbf{v}_G = \frac{\partial \mathcal{G}}{\partial t}\ , \tag{2.22}$$

denote the gradients of the mapping.

With this, a generic conservation law written as a first order system in the physical space as:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{A}(\mathbf{u}, \mathbf{Q}) = \mathbf{s}\ , \forall \mathbf{x} \in \Omega_{\mathbf{x}}, \tag{2.23}$$

$$\mathbf{Q} - \nabla \mathbf{u} = 0\ , \forall \mathbf{x} \in \Omega_{\mathbf{x}}, \tag{2.24}$$

can be converted to an equivalent conservation statement in the reference space of the

form:

$$\frac{\partial g\mathbf{u}}{\partial t} + \nabla_{\mathbf{r}} \cdot \left\{ g\mathbf{G}^{-1} \left( \mathbf{A}(\mathbf{u}, \mathbf{G}^{-1}\mathbf{Q}_{\mathbf{r}}) - \mathbf{u} \otimes \mathbf{v}_G \right) \right\} = g\mathbf{s} \ , \forall \mathbf{r} \in \Omega_{\mathbf{r}}, \tag{2.25}$$

$$\mathbf{Q}_{\mathbf{r}} - \nabla_{\mathbf{r}}\mathbf{u} = 0 \ , \forall \mathbf{r} \in \Omega_{\mathbf{r}}, \tag{2.26}$$

where $g = \det(\mathbf{G})$ and the divergence and gradient operators work on the reference coordinates $\mathbf{r}$. Here, the flux $\mathbf{A}$ agglomerates both inviscid and viscous contributions:

$$\mathbf{A} = \mathbf{F}_{\text{inv}}(\mathbf{u}) - \mathbf{F}_{\text{visc}}(\mathbf{u}, \mathbf{Q}). \tag{2.27}$$

One significant difference between this work and previous uses of this ALE formulation [187, 111, 78] is the fact that the mesh deformation is solved for together with the flow discretization inside a Newton-Raphson iteration. This implies that the velocity of the mapping, its gradient, as well as the Jacobian of them with respect to the degrees of freedom that govern the mesh geometry have to computed in a consistent manner. In the particular case of the Jacobian of geometric quantities like $\mathbf{G}$ or $g$, this requires the use of the isoparametric description of the geometry (see Section 4.2.3) and the adequate matrix identities [191].

In some instances, the formulation can be augmented with the use of a Geometric Conservation Law (or GCL) to correct for possible errors in the integration in time of the geometry [187]. However this only makes a small difference in unsteady flows [187, 111, 78] and is irrelevant when steady state solutions are sought after. For this reason, the GCL was not implemented in this work.

## 2.2 Flow discretization

The use of an ALE formulation yields a system of PDEs that are written on a fixed reference mesh and can be discretized using a standard technique. In this work, the spatial discretization is based on the hybridizable discontinuous Galerkin (HDG) Scheme by Nguyen *et al.* [168, 167, 166], combined with a Backward Euler scheme

in time. This kind of discretization has been tested before in the context of the Euler/Navier-Stokes [180] and RANS-SA system [158, 44], and delivers high order accuracy at a reasonable cost in unstructured and hybrid meshes.

### 2.2.1 Spatial discretization: the hybridizable discontinuous Galerkin scheme

The starting point of the spatial discretization is a system of partial differential equations written as a first order system. The system consists of the ALE formulation in Eq. 2.25-2.26 and the boundary conditions represented by a boundary operator:

$$\frac{\partial g\mathbf{u}}{\partial t} + \nabla_{\mathbf{r}} \cdot \mathbf{A}_{\mathrm{ALE}} = g\mathbf{s} \ , \quad \text{in } \Omega_{\mathbf{r}}, \tag{2.28}$$

$$\mathbf{Q}_{\mathbf{r}} - \nabla_{\mathbf{r}}\mathbf{u} = 0 \ , \quad \text{in } \Omega_{\mathbf{r}}, \tag{2.29}$$

$$\mathbf{f}_{\mathrm{bou}}(\mathbf{u}, \mathbf{G}^{-1}\mathbf{Q}_{\mathbf{r}}, g\mathbf{G}^{-1}\mathbf{n}_{\mathbf{r}}) = 0, \quad \text{on } \partial\Omega_{\mathbf{r}}, \tag{2.30}$$

where $\mathbf{n}_{\mathbf{r}}$ denotes the normal to the boundary in the reference domain. Here, the ALE fluxes have been abbreviated into:

$$\mathbf{A}_{\mathrm{ALE}} = g\mathbf{G}^{-1}\left(\mathbf{A}(\mathbf{u}, \mathbf{G}^{-1}\mathbf{Q}_{\mathbf{r}}) - \mathbf{u} \otimes \mathbf{v}_G\right). \tag{2.31}$$

Before proceeding with the discretization, some definitions are due. Let $\mathcal{T}_h$ denote a triangulation of $\Omega_{\mathbf{r}}$ composed of disjoint regular elements $K$ that partitions $\Omega_{\mathbf{r}}$. Let $\partial K$ denote the boundary of each element and $\partial\mathcal{T}_h = \cup_{\mathcal{T}_h}\partial K$ denote the union of all of them. In addition, let $\mathcal{E}_h$ denote the edges (in 2-D, faces in 3-D) of the triangulation counted only once. The following discontinuous spaces are supported on

these geometric entities:

$$\boldsymbol{\mathcal{V}}_h^p = \{\mathbf{v} \in (L^2(\Omega))^m : \mathbf{v}|_K \in (\mathcal{P}^p(K))^m \quad \forall K \in \mathcal{T}_h\}, \tag{2.32}$$

$$\boldsymbol{\mathcal{W}}_h^p = \{\mathbf{W} \in (L^2(\Omega))^{m\times d} : \mathbf{W}|_K \in (\mathcal{P}^p(K))^{m\times d} \quad \forall K \in \mathcal{T}_h\}, \tag{2.33}$$

$$\boldsymbol{\mathcal{M}}_h^p = \{\mathbf{m} \in (L^2(\mathcal{E}_h))^m : \mathbf{m}|_e \in (\mathcal{P}^p(e))^m \quad \forall e \in \mathcal{E}_h\}, \tag{2.34}$$

where $\mathcal{P}^p(D)$ represents the space of polynomials of degree $p$ in the domain $D$, $d$ represents the number of space dimensions of the problem and $m$ represents the number of conservation laws in the system.

In addition, the following inner products in $L^2$ need to be defined. In particular, let

$$(a,b)_{\mathcal{T}_h} \coloneqq \sum_{K\in\mathcal{T}_h} \int_K ab, \qquad \text{and} \qquad \langle a,b\rangle_{\partial\mathcal{T}_h} \coloneqq \sum_{K\in\mathcal{T}_h} \int_{\partial K} ab, \tag{2.35}$$

denote the duality pairing between scalar functions over the volume and faces, respectively. Similarly, let

$$(\mathbf{a},\mathbf{b})_{\mathcal{T}_h} \coloneqq \sum_i (a_i,b_i)_{\mathcal{T}_h}, \qquad \text{and} \qquad \langle \mathbf{a},\mathbf{b}\rangle_{\partial\mathcal{T}_h} \coloneqq \sum_i \langle a_i,b_i\rangle_{\partial\mathcal{T}_h}, \tag{2.36}$$

denote the duality pairing between vector valued functions on the volume and faces, respectively. Finally, let

$$(\mathbf{A},\mathbf{B})_{\mathcal{T}_h} \coloneqq \sum_{K\in\mathcal{T}_h} \int_K \mathrm{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{B}), \tag{2.37}$$

denote the inner product between tensor (matrix) valued functions on the volume.

The HDG discretization of the system follows from the integration by parts of Eq. 2.25-2.26 against elements of the test space, combined with a constraint that enforces conservation across faces in a weak sense. The problem reads: find $(\mathbf{u}_h, \mathbf{Q}_{\mathbf{r}h}, \hat{\mathbf{u}}_h) \in$

$\boldsymbol{\mathcal{V}}_h^p \times \boldsymbol{\mathcal{W}}_h^p \times \boldsymbol{\mathcal{M}}_h^p$, such that:

$$\left(\frac{\partial g\mathbf{u}_h}{\partial t}, \mathbf{v}\right)_{\mathcal{T}_h} - (\mathbf{A}_{\mathrm{ALE}}, \nabla\mathbf{v})_{\mathcal{T}_h} + \langle\widehat{\mathbf{A}_{\mathrm{ALE}}} \cdot \mathbf{n}_\mathbf{r}, \mathbf{v}\rangle_{\partial\mathcal{T}_h} - (g\mathbf{s}, \mathbf{v})_{\mathcal{T}_h} = 0, \tag{2.38}$$

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

$$\langle\widehat{\mathbf{A}_{\mathrm{ALE}}} \cdot \mathbf{n}_\mathbf{r}, \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h \setminus \partial\Omega_\mathbf{r}} + \langle\mathbf{f}_{\mathrm{bou}}, \boldsymbol{\mu}\rangle_{\partial\Omega_\mathbf{r}} = 0, \tag{2.40}$$

$\forall(\mathbf{v}, \mathbf{E}, \boldsymbol{\mu}) \in \boldsymbol{\mathcal{V}}_h^p \times \boldsymbol{\mathcal{W}}_h^p \times \boldsymbol{\mathcal{M}}_h^p$. Here, the function:

$$\begin{aligned}\widehat{\mathbf{A}_{\mathrm{ALE}}} \cdot \mathbf{n}_\mathbf{r} = {} & g\mathbf{G}^{-1}\left(\mathbf{A}(\hat{\mathbf{u}}_h, \mathbf{G}^{-1}\mathbf{Q}_{\mathbf{r}h}) - \hat{\mathbf{u}}_h \otimes \mathbf{v}_G\right) \cdot \mathbf{n}_\mathbf{r} \\ & + \mathbf{S}(\mathbf{u}_h - \hat{\mathbf{u}}_h),\end{aligned} \tag{2.41}$$

is the normal component of the numerical flux on the boundaries of each element. In it, the second term on the right hand side represents the stabilization of the scheme, where $\mathbf{S} = \mathbf{S}(\hat{\mathbf{u}}_h, \mathbf{v}_G, g\mathbf{G}^{-1}\mathbf{n}_\mathbf{r})$ is the so-called stabilization matrix.

### Eigendecomposition of the inviscid ALE fluxes

The choice of the stabilization matrix as well as the definition of some of the boundary conditions requires an explicit eigenvalue decomposition of the Jacobian of the normal inviscid flux with respect to the state variables.

In the case of the standard fluxes without ALE, said decomposition has already been studied for both the Navier-Stokes/Euler system (see Roe [205] and Toro [237]) as well as the RANS-SA system (see Burgess [33]). In a generic form such decomposition reads:

$$\frac{\partial(\mathbf{F}_{\mathrm{inv}} \cdot \mathbf{n})}{\partial\mathbf{u}} = \mathbf{K}\boldsymbol{\Lambda}\mathbf{K}^{-1}, \tag{2.42}$$

where $\mathbf{K}$ represents the eigenvectors and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_\mathrm{i})$ denotes a diagonal matrix that contains the eigenvalues of the problem or, equivalently, the wave speeds across the interface.

For the ALE formulation, the eigendecomposition depends also on the velocity of the mesh as well as its deformation. In particular, these produce a shifting effect on the eigenvalues and a scaling of the whole Jacobian [187], without affecting the eigenvectors. In this way, the Jacobian can be written as:

$$\frac{\partial(\mathbf{A}_{\mathrm{ALE}} \cdot \mathbf{n_r})}{\partial \mathbf{u}} = \|g\mathbf{G}^{-1}\mathbf{n_r}\|\mathbf{K}(\mathbf{\Lambda} - v_{Gn}\mathbf{I})\mathbf{K}^{-1}, \tag{2.43}$$

where $v_{Gn}$ is the normal velocity of the mesh in the physical space given by:

$$v_{Gn} = \mathbf{v}_G \cdot \frac{g\mathbf{G}^{-1}\mathbf{n_r}}{\|g\mathbf{G}^{-1}\mathbf{n_r}\|}. \tag{2.44}$$

To simplify the notation, let $\tilde{\mathbf{\Lambda}} = \mathbf{\Lambda} - v_{Gn}\mathbf{I}$ denote the matrix of shifted eigenvalues, and let $\mathbf{n_x} = g\mathbf{G}^{-1}\mathbf{n_r}$ denote the transformed normal vector in the physical space.

### Stabilization matrix

The choice of stabilization matrix follows the work by Peraire and Nguyen [180] on the Navier-Stokes system, in which three different strategies were proposed. In the case of the ALE formulation, these same strategies can be applied, but care has to be taken in order to be consistent with the information from the eigendecomposition.

The first choice is inspired in the Riemann solver of Roe [205] and is based on using the absolute value of the wave speed in the reconstruction of the eigensystem. In this way, the stabilization matrix reads:

$$\mathbf{S} = \|\mathbf{n_x}\|\mathbf{K}|\tilde{\mathbf{\Lambda}}|\mathbf{K}^{-1}, \tag{2.45}$$

where $|\tilde{\mathbf{\Lambda}}| = \mathrm{diag}(|\lambda_i - v_{Gn}|)$ and the eigendecomposition is computed using the trace state $\hat{\mathbf{u}}_h$ such that: $\tilde{\mathbf{\Lambda}} = \tilde{\mathbf{\Lambda}}(\hat{\mathbf{u}}_h)$ and $\mathbf{K} = \mathbf{K}(\hat{\mathbf{u}}_h)$.

The second choice is the Local Lax-Friedrich flux in which only the fastest wave

speed is considered. In this way, the stabilization matrix reads:

$$\mathbf{S} = ||\mathbf{n_x}|| \max(|\lambda_i - v_{Gn}|)\mathbf{I}. \tag{2.46}$$

Again, the fastest wave speed is computed with the help of $\hat{\mathbf{u}}_h$ as $\lambda_i = \lambda_i(\hat{\mathbf{u}}_h)$.

The last choice is usually referred to as the Global Lax-Friedrich flux and is based on using a constant value to approximate the wave speed across the interface. In that way, the stabilization matrix reads:

$$\mathbf{S} = ||\mathbf{n_x}||(1 + M_\infty)\mathbf{I}. \tag{2.47}$$

Unless otherwise stated, the results presented in this thesis were computed using the Global Lax-Friedrich stabilization matrix. This is motivated by its simplicity and by the fact that, according to Nguyen and Peraire [166], the effect of the stabilization matrix on the solution is minimal for high order discretizations.

Finally, notice that the stabilization matrix does not depend on the viscous terms of the governing equations, which is consistent with the findings by Nguyen *et al.* [167] for scalar convection-diffusion laws, in spite of the fact that the viscous fluxes in the Navier-Stokes system are not strictly positive definite.

### Boundary conditions

The boundary conditions for the problem are implemented in weak form through the last term on the left hand side of Eq. 2.40. In what follows, the ones used in this thesis are briefly described:

- Far-field, subsonic inflow and supersonic inflow/outflow:

$$\mathbf{f}_{\text{bou}} = ||\mathbf{n_x}|| \left( \mathbf{K}(|\tilde{\mathbf{\Lambda}}| + \tilde{\mathbf{\Lambda}})\mathbf{K}^{-1}(\mathbf{u}_h - \hat{\mathbf{u}}_h) + \mathbf{K}(|\tilde{\mathbf{\Lambda}}| - \tilde{\mathbf{\Lambda}})\mathbf{K}^{-1}(\mathbf{u}_\infty - \hat{\mathbf{u}}_h) \right), \tag{2.48}$$

  where $\mathbf{u}_\infty$ denotes a freestream state that includes the SA variable if required, $\tilde{\mathbf{\Lambda}} = \tilde{\mathbf{\Lambda}}(\hat{\mathbf{u}}_h)$ and $\mathbf{K} = \mathbf{K}(\hat{\mathbf{u}}_h)$. This boundary condition decomposes the normal

flux into incoming and outgoing waves and either imposes or extrapolates the correct eigenvector.

- Subsonic outflow at pressure $P_{\text{out}}$:

$$\mathbf{f}_{\text{bou}} = \|\mathbf{n}_{\mathbf{x}}\| \left( \mathbf{K}(|\tilde{\mathbf{\Lambda}}| + \tilde{\mathbf{\Lambda}})\mathbf{K}^{-1}(\mathbf{u}_h - \hat{\mathbf{u}}_h) + \mathbf{K}(|\tilde{\mathbf{\Lambda}}| - \tilde{\mathbf{\Lambda}})\mathbf{K}^{-1}(\mathbf{u}_{\text{out}} - \hat{\mathbf{u}}_h) \right), \qquad (2.49)$$

where:

  - $\mathbf{u}_{\text{out}} = (\rho_h, \rho\mathbf{v}_h, \frac{P_{\text{out}}}{\gamma-1} + \frac{1}{2}\rho_h \mathbf{v}_h \cdot \mathbf{v}_h)$, for the Navier-Stokes equations, and,
  - $\mathbf{u}_{\text{out}} = (\rho_h, \rho\mathbf{v}_h, \frac{P_{\text{out}}}{\gamma-1} + \frac{1}{2}\rho_h \mathbf{v}_h \cdot \mathbf{v}_h, \rho\tilde{\nu}_h)$, for the RANS-SA system.

As in the previous case, $\tilde{\mathbf{\Lambda}} = \tilde{\mathbf{\Lambda}}(\hat{\mathbf{u}}_h)$ and $\mathbf{K} = \mathbf{K}(\hat{\mathbf{u}}_h)$. This particular definition of $\mathbf{u}_{\text{out}}$ ensures that all eigenvectors are extrapolated except for the pressure wave that is fixed by $P_{\text{out}}$.

- Adiabatic wall:

$$\mathbf{f}_{\text{bou}} = \begin{Bmatrix} \rho_h - \hat{\rho}_h \\ \widehat{\rho\mathbf{v}}_h \\ \widehat{\mathbf{A}_{\text{ALE}}} \cdot \mathbf{n}_{\mathbf{r}}|_{\text{energy}} \end{Bmatrix}, \quad \text{or,} \quad \mathbf{f}_{\text{bou}} = \begin{Bmatrix} \rho_h - \hat{\rho}_h \\ \widehat{\rho\mathbf{v}}_h \\ \widehat{\mathbf{A}_{\text{ALE}}} \cdot \mathbf{n}_{\mathbf{r}}|_{\text{energy}} \\ \widehat{\rho\tilde{\nu}}_h \end{Bmatrix}, \qquad (2.50)$$

for the Navier-Stokes or RANS-SA equations, respectively. Here $\widehat{\mathbf{A}_{\text{ALE}}} \cdot \mathbf{n}_{\mathbf{r}}|_{\text{energy}}$ denotes the ALE flux corresponding to the conservation of energy, which is forced to be zero through the 4th component of $\mathbf{f}_{\text{bou}}$. Similarly, the velocity (and the SA variable when appropriate) are also set to zero.

- Inviscid wall:

$$\mathbf{f}_{\text{bou}} = \widehat{\mathbf{A}_{\text{ALE}}} \cdot \mathbf{n}_{\mathbf{r}} - \begin{Bmatrix} 0 \\ \widehat{P}_h \mathbf{n}_{\mathbf{x}} \\ \widehat{P\mathbf{v}}_h \cdot \mathbf{n}_{\mathbf{x}} \end{Bmatrix}, \quad \text{or,} \quad \mathbf{f}_{\text{bou}} = \widehat{\mathbf{A}_{\text{ALE}}} \cdot \mathbf{n}_{\mathbf{r}} - \begin{Bmatrix} 0 \\ \widehat{P}_h \mathbf{n}_{\mathbf{x}} \\ \widehat{P\mathbf{v}}_h \cdot \mathbf{n}_{\mathbf{x}} \\ 0 \end{Bmatrix}, \tag{2.51}$$

  for the Navier-Stokes or RANS-SA equations, respectively. This boundary condition states that the fluxes of mass, momentum and total energy have to be zero.

### 2.2.2 Time discretization

The spatial discretization using the HDG scheme turns the PDE into a system of Differential-Algebraic Equations (DAE) for the evolution of the solution in time. In particular, the differential terms in time appear on the first term in Eq. 2.38 as well as implicitly in the ALE fluxes through the velocity of the mapping $\mathbf{v}_G$.

In order to march the solution in time, these need to be discretized in a consistent manner. In this work, this is accomplished by means of a simple BDF1 scheme (Backward Euler). In this way, the time derivatives are approximated using the formulae:

$$\frac{\partial g \mathbf{u}_h}{\partial t} \approx \frac{g\mathbf{u}_h - g^0 \mathbf{u}_h^0}{\Delta t}, \tag{2.52}$$

$$\mathbf{v}_G = \frac{\partial \mathcal{G}}{\partial t} \approx \frac{\mathcal{G}(\mathbf{r}, t) - \mathcal{G}(\mathbf{r}, t^0)}{\Delta t}, \tag{2.53}$$

where $\Delta t$ is the time step length and the superscript $(\,)^0$ denotes values computed from the previous time step. Notice that the mapping $\mathcal{G}$ is defined explicitly as a function of the location of the high order nodes of the mesh, hence, the reconstruction of the time derivatives requires the storage of the mesh configuration at the previous time step.

All in all, this time discretization turns the DAE system into a nonlinear system

of algebraic equations that can be solved sequentially to approximate the evolution in time or until steady state is reached.

## 2.3 Flow discretization module

In this chapter, the equations that govern the flow and the scheme used to discretize them have been introduced. The flow discretization module is closed up to the definition of the mesh mapping $\mathcal{G}$ and the intermittency factor $\gamma$, that are given by the adaptivity module and the transition prediction module, respectively. This relationship is multi-lateral as these will depend on one another. The particular details of the coupled solver will be presented in the coming chapters, however, for cases where the mesh is fixed in time and there is no need to simulate transition to turbulence effects (e.g. all the instances in Chapter 3), the discretization presented here all that is required.

# Chapter 3

# Shock capturing with dilatation-based artificial viscosity

In this chapter, the shock capturing strategy proposed as part of this thesis is presented and tested for the case of inviscid flows.

As already mentioned in Section 1.3.2, the most common failure mode of high order simulations of compressible flow is associated with the appearance of shock waves. The physical nature of these waves is such that a finite jump in the flow variables happens in a thickness of the order of a few mean free paths [140]. Translated to the discrete mathematical setting, this is equivalent to a discontinuity in the field, which, in principle, cannot be smoothly represented in the space of the discrete solution. This feature is independent of whether such solution is obtained through collocation or projection, and manifests itself in the form of oscillations (or "wiggles") around the shock. In certain instances, such wiggles might be strong enough to induce an unphysical state in the flow field (e.g.: negative pressure, negative speed of sound, etc.), which prevents the convergence of the solver.

To remedy this, a variety of approaches have been proposed, that are grossly categorized into two groups: strategies that modify the numerical scheme, and strategies that augment the equations with extra dissipation. In both cases the ultimate goal is to control the appearance and intensity of such oscillations. Furthermore, in some

instances, these two strategies can be related by means of careful numerical analysis.

The approach proposed in this thesis belongs to the second group, commonly dubbed as artificial viscosity. In short, the main idea is to augment the conservative fluxes ($\mathbf{F}$) with artificial viscosity fluxes ($\mathbf{G}$) that depend on the solution:

$$\nabla \cdot \mathbf{F} = 0 \rightarrow \nabla \cdot \mathbf{F} - \nabla \cdot \mathbf{G}(\mathbf{u}, \nabla \mathbf{u}) = 0. \tag{3.1}$$

To be successful, the artificial viscosity fluxes have to induce dissipation. This is guaranteed if these can be written as:

$$\mathbf{G}(\mathbf{u}, \nabla \mathbf{u}) = \epsilon \mathbf{A}^{\epsilon} \nabla \mathbf{u}, \tag{3.2}$$

with $\epsilon > 0$ and $\mathbf{A}^{\epsilon} \geq 0$. Here $\epsilon \in \mathbb{R}$ is the amount of viscosity itself and $\mathbf{A}^{\epsilon}(\mathbf{u}) \in \mathbb{R}^{mxm}$ is the Jacobian of the artificial viscous fluxes with respect to the gradient of the solution, following notation by Barter [21]. The problem of the design of an artificial viscosity model boils down to the selection of $\epsilon$ and the matrix $\mathbf{A}^{\epsilon}$.

For $\mathbf{A}^{\epsilon}$, the three most common choices are:

- $\mathbf{A}^{\epsilon} = \mathbf{I}$, usually referred to as the Laplacian flux,
- $\mathbf{A}^{\epsilon}$ taken from the viscous portion of the Navier-Stokes equations, with or without heat conduction [188], and,
- $\mathbf{A}^{\epsilon}$ such that $\mathbf{A}^{\epsilon} \nabla \mathbf{u} = \nabla \mathbf{u}_{\mathrm{AV}}$, with $\mathbf{u}_{\mathrm{AV}} = (\rho, \rho \mathbf{v}, \rho H)$ as proposed by Jameson [107] to preserve total enthalpy across a shock.

As for $\epsilon$, it can be governed by the solution $\mathbf{u}$ (and derivatives) [195], the residual of the equations [24, 27], a filter of the solution [188], and others. A detailed list of references is given in Section 1.3.2. In this work, the first approach is taken.

The structure of this chapter is as follows. First, the formulation is introduced for the general case. Next, a one dimensional problem is investigated in order to validate the approach. To continue, several results for inviscid compressible 2-D flows

are presented and discussed. Finally, the virtues and drawbacks of the model are discussed together with potential future developments.

## 3.1 Formulation

The starting point of the proposed scheme is the shock sensor by Nguyen and Peraire [165], that took inspiration from the work of Premasuthan and Jameson [195], which itself was based on the ideas by Bhagatwala and Lele [28]. The focus in all these cases is the proper identification of a shock using the divergence of the velocity ($\nabla \cdot \mathbf{v}$, or dilatation) as a sensor to drive the amount of artificial viscosity $\epsilon$ added to the system.

### 3.1.1 Original formulation by Nguyen and Peraire

In the original paper, Nguyen and Peraire [165] proposed to take advantage of the optimal convergence of the gradients of the solution in the hybridizable discontinuous Galerkin (HDG) method to devise a pointwise indicator based on the dilatation of the flow. Their strategy consisted in solving the augmented system:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = \nabla \cdot \epsilon \nabla \mathbf{u}_{\mathrm{AV}}, \tag{3.3}$$

where $\mathbf{u}$ are the conserved variables in the Euler or Navier-Stokes equations, $\mathbf{F}$ are the fluxes of the original governing equations, and the terms on the right hand side take the form:

$$\tilde{s} = -\frac{l \nabla \cdot \mathbf{v}}{c}, \tag{3.4}$$

$$\epsilon = \epsilon_0 f\left(\tilde{s}\right), \tag{3.5}$$

$$\mathbf{u}_{\mathrm{AV}} = (\rho, \mathbf{v}, \rho H). \tag{3.6}$$

Here, $\tilde{s}$ is the shock indicator, $c = \sqrt{\gamma P/\rho}$ is the speed of sound and $l$ is a length scale that has to be precomputed using the expression $l = \min\left(h_0, 10d\right)$, where $d$ is the

distance to the closest wall. Finally, the function $f$ is defined as:

$$f(x) = \frac{\log\left(1 + \exp\left(\alpha(x-\beta)\right)\right)}{\alpha}, \tag{3.7}$$

$$\alpha = 10,\ \beta = 0.5, \tag{3.8}$$

and plays the role of a smooth surrogate for a switch function: $f(x) \approx \max(0, x-\beta)$. In essence, this strategy uses a non-dimensional version of the dilatation to drive the application of artificial viscosity in those regions where the dilatation is negative. The model is closed with the definition of the pair $(\epsilon_0, h_0)$. Typical values found in the original paper are spread in a range of two orders of magnitude, and have to be tuned on a case by case basis.

This approach is simple enough to implement in any solver where gradients are available or can be computed, yet successful at capturing a variety of flows, especially when combined with adaptivity as the results in the original paper indicated [165].

### 3.1.2 Modifications to the original model

A careful examination of the model (Eq. 3.3-3.7) reveals room for improvement in a variety of aspects, such as:

- the need for two tuning parameters, $h_0$ and $\epsilon_0$, that in general depend on the particular flow of interest through the geometry and the flow conditions, as well as the mesh,

- the jump in the speed of sound across the shock, that might yield a non-symmetric sensor profile around the midpoint of the shock,

- the use of a switch function to calculate $l$, that uses the distance to the closest wall $d$ to effectively turn off the artificial viscosity close to the boundaries,

- the choice of parameters $\alpha$ and $\beta$ in the function $f(x)$, that dictate the aggressiveness in the application of the viscosity as well as the residual values of

viscosity in the incompressible and expansion regimes ($\nabla \cdot \mathbf{v} \geq 0$).

In what follows, these points are discussed one by one.

**Choice of the tuning parameters**

The first set of modifications is related to the choice of $h_0$ and $\epsilon_0$. The starting point being the choice of the former, as it directly affects the definition of the latter. The role of $h_0$ is to provide a length scale for the shock thickness. In low order schemes, such length scale is usually of the order of the element size, denoted by $h$. For high order finite element methods, as described by Persson and Peraire [188], such length scale is of the order of $h/p$. In general, $h$ is not constant in the whole domain but rather varies from element to element, and therefore, it needs to be defined at every point in the domain as a scalar field $h = h(\mathbf{x})$.

The simplest approach would be to assign a certain element size $h_e$ to every element in the mesh, and assume it is constant within each particular element. This way, the resulting element size field $h(\mathbf{x})$ would be piecewise constant, which would produce a discontinuous artificial viscosity field even if the solution was free from oscillations. According to Barter and Darmofal [21], such jumps in the artificial viscosity field degrade the accuracy of the solution and might lead to instabilities.

For this reason, a piecewise linear reconstruction like the one proposed by Barter and Darmofal is used instead [21]. This reconstruction is based on averaging out the element size $h_e$ for all the elements surrounding a vertex and propagating this value inside the elements through a continuous, piecewise linear approximation.

This process requires the definition of $h_e$ as a function of the element shape and size, which is crucial when anisotropic adaptation is performed and elements align with the discontinuities, ideally reaching very high stretching factors. In that instance, the natural length scale for the shock is the element length measure in the direction of the gradient of the solution. A good surrogate for this is the smallest altitude of the element, defined as the minimum orthogonal distance from any vertex of the polygonal (triangle or quadrangle) to any opposite side, as depicted in Figure 3-1.

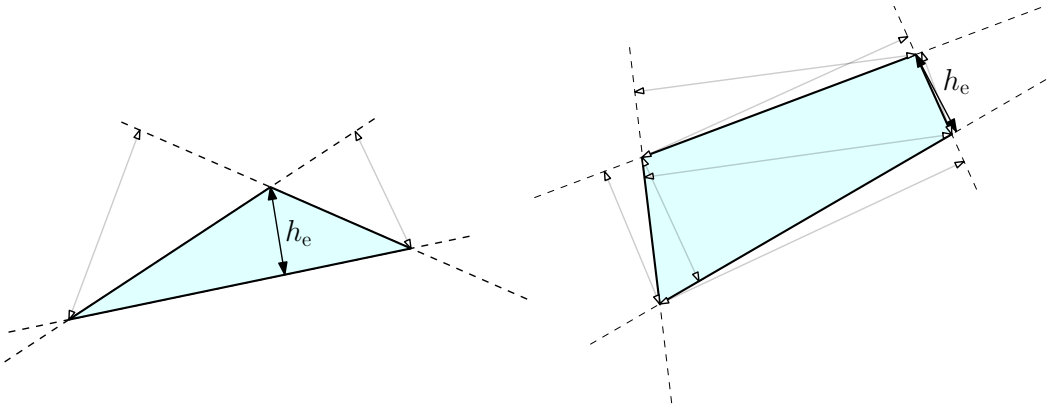



Figure 3-1: Sketch of the procedure to extract $h_e$ for triangles and quadrilaterals, based on identifying the smallest altitude in the polygon. All the altitudes are colored in light gray except for the smallest one, colored in solid black. The use of the smallest altitude ensures that the correct length scale is used under the assumption that anisotropic meshes align with shocks.

Finally, using the definition of the element size field, the parameter $h_0$ in the original model is substituted by the proper length scale:

$$h_0 \to k_h \frac{h(\mathbf{x})}{p}, \tag{3.9}$$

where $k_h = \mathcal{O}(1)$ is the length scale correction factor, whose purpose is to fine tune the model so that the shock profiles are free from oscillations. The particular choice of $k_h$ will be discussed later in this chapter with the help of 1-D results.

The new definition of $h_0$ simplifies the task of choosing a scale for $\epsilon_0$. In particular, $\epsilon_0$ is related to the viscous length scale of the shock, and has to be such that there is a smooth transition of the solution from the pre-shock state to the post-shock state, within one element. Dimensional consistency dictates that it has to have units of velocity times length. The latter is already defined by $h(\mathbf{x})$; hence, only the velocity scale requires attention. A natural choice for this would be the fastest wave across the shock, given by $\lambda_{\max} = |\mathbf{v} \cdot \mathbf{n}| + c$; however, this requires the extraction of the vector normal to the shock front $\mathbf{n}$. A simpler choice is to take $||\mathbf{v}||$ instead of $|\mathbf{v} \cdot \mathbf{n}|$. With

this, the value of $\epsilon_0$ is substituted by:

$$\epsilon_0 \to \left(k_h \frac{h}{p}\right)(\|\mathbf{v}\| + c)\,. \tag{3.10}$$

The above definition of the velocity scale might produce unphysical values when the oscillations in the solution generated by strong shocks are present. This is due to the dependency of the speed of sound on the pressure and can be prevented by regularizing the latter so that it is positive before entering the computation of $c$.

$$P \to \max(10^{-16} P_\infty, P), \tag{3.11}$$

which yields:

$$\epsilon_0 \to \left(k_h \frac{h}{p}\right)\left(\|\mathbf{v}\| + \sqrt{\frac{\gamma \max(10^{-16} P_\infty, P)}{\rho}}\right). \tag{3.12}$$

Here $P_\infty$ is the freestream pressure and the maximization is introduced to ensure that the term inside the square root is always strictly positive (although it can get very close to zero thanks to the $10^{-16}$ factor in front of it). It is worth noting that the clip is mostly a cautionary measure that is seldom active in steady state solutions, but serves to prevent a blowup during transients. A similar clip could be introduced in the denominator, however, as opposed to the pressure, it is hard to find a negative value of the density (even during transients) in the cases run here.

Notice that, provided the shock sensor is of order one, this definition of $\epsilon_0$ yields a cell Peclet number $Pe_{\text{cell}} = \mathcal{O}(1)$, which reconciles well with linear theory for convection-diffusion and the limit of $Pe_{\text{cell}} < 2$ for oscillation free solutions.

The new definitions for $h_0$ and $\epsilon_0$ ensure that the problem is dimensionally correct and also that both the sensor and the artificial viscosity itself are properly scaled with the mesh size and anisotropy.

### Choice of the velocity scale for the sensor

The second modification to the model is related to the choice of a velocity scale for the sensor. In particular, the speed of sound $c$ in the denominator of Eq. 3.4, whose presence raises two flags that are discussed next.

On the one hand, the relationship between the speed of sound and the pressure suggest that $c$ can turn imaginary whenever the pressure becomes negative. Unfortunately, this situation is common when the solution oscillates in the vicinity of strong shocks, or during transients and was the reason argued by Nguyen and Peraire to regularize the speed of sound for high Mach number cases [165]. The ideal choice of velocity scale would be one that while being relevant to the shock phenomenon does not depend so strongly on oscillatory quantities like the pressure.

On the other hand, even if the solution was free from oscillations, there exists a jump in temperature across the shock, which produces a jump in the speed of sound that translates directly into an asymmetry in the shock sensor. This bias is studied using shock wave theory as follows. Let 1 and 2 denote the state of the flow upstream and downstream of the shock, respectively. The starting point is the assumption that the shock is captured within one element so that the divergence of the velocity can be approximated by:

$$\nabla \cdot \mathbf{v} \approx \frac{\Delta v_n}{k_h h/p}, \tag{3.13}$$

where $\Delta v_n = v_{2\mathrm{n}} - v_{1\mathrm{n}}$ is the jump in normal velocity across the shock. Multiplying the estimate for $\nabla \cdot \mathbf{v}$ by the element size field and the speed of sound before and after the shock yields:

$$\tilde{s}_1 = -\frac{(k_h h/p)\nabla \cdot \mathbf{v}}{c_1} \approx -\frac{v_{2\mathrm{n}} - v_{1\mathrm{n}}}{c_1} = \left(1 - \frac{v_{2\mathrm{n}}}{v_{1\mathrm{n}}}\right) M_{1\mathrm{n}}, \tag{3.14}$$

$$\tilde{s}_2 = -\frac{(k_h h/p)\nabla \cdot \mathbf{v}}{c_2} \approx -\frac{v_{2\mathrm{n}} - v_{1\mathrm{n}}}{c_2} = \left(1 - \frac{v_{2\mathrm{n}}}{v_{1\mathrm{n}}}\right) M_{1\mathrm{n}} \frac{c_1}{c_2}. \tag{3.15}$$

Where $\tilde{s}$ denotes the approximate value of the shock sensor. It is thus clear that the

shock sensor is skewed by a factor of:

$$\frac{\tilde{s}_1}{\tilde{s}_2} \approx \frac{c_2}{c_1} = \sqrt{\frac{T_2}{T_1}} = \sqrt{\frac{\left(2\gamma M_{1n}^2 - (\gamma-1)\right)\left((\gamma-1)M_{1n}^2 + 2\right)}{(\gamma+1)^2 M_{1n}^2}}, \tag{3.16}$$

which grows linearly with the incident Mach number, $M_{1n}$. To avoid this asymmetry, the velocity scale should be one with none or minimal variation across the shock. An effective approach would be to use a velocity scale that depends on a conserved quantity across the shock. Such is the case of the critical speed of sound ($c^*$), that depends only on the total temperature ($T_0$) in the following manner:

$$c^* = \sqrt{\gamma R T^*} = \sqrt{\gamma R \left(\frac{2}{\gamma+1}\right) T_0}. \tag{3.17}$$

Notice that the value of $T_0$ across a shock is constant in the frame of reference moving with the discontinuity. This implies that $T_0$ is constant across a shock in a steady state flow, which makes the velocity scale constant over the transition region around the shock.

The use of $c^*$ as velocity scale yields:

$$\tilde{s}^* = -\frac{(k_h h/p)\nabla \cdot \mathbf{v}}{c^*} \approx -\frac{v_{2n} - v_{1n}}{c^*} = \left(1 - \frac{v_{2n}}{v_{1n}}\right) M_{1n} \frac{c_1}{c^*}, \tag{3.18}$$

that can be further expanded into:

$$\tilde{s}^* \approx \frac{2M_{1n}^2 - 2}{(\gamma+1)M_{1n}} \sqrt{\frac{\gamma+1}{2 + (\gamma-1)M_{1n}^2}}, \tag{3.19}$$

which in the limit of high Mach number asymptotes to a constant. In principle, this is desirable as it reduces the complexity of the model and decouples the shock sensor from the strength of the shock. The three estimates for the shock sensor: $\tilde{s}_1$, $\tilde{s}_2$ and $\tilde{s}^*$ are compared in Figure 3-2.

Before moving on to the next point, it is necessary to comment on the assumptions used in this simple analysis, in particular the validity of Eq. 3.13 as an approximation

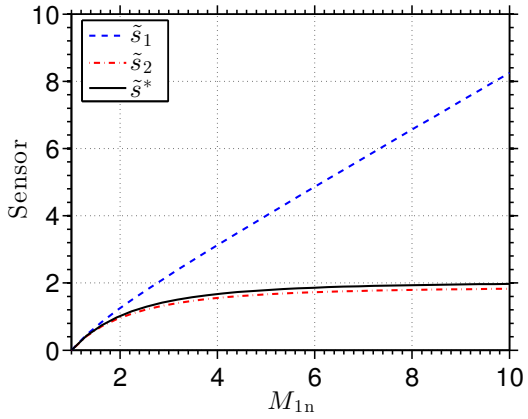



Figure 3-2: Estimates for $\tilde{s}_1$, $\tilde{s}_2$ and $\tilde{s}^*$ as a function of the incident Mach number $M_{1n}$, based on the assumption that the shock is one-dimensional and captured within the available resolution. Notice how $\tilde{s}^*$ is very close to $\tilde{s}_2$, and both of them asymptote to a constant independently of the shock strength.

to the divergence of the velocity field. The main concern here is whether or not the actual divergence is symmetric across the shock, and, if not, how does it affect the analysis. Unfortunately, the shock profile does not have a analytical solution to answer this question. An empirical answer is provided in a later section of this chapter by means of a numerical experiment.

### Interaction with the boundary condition

The next modification to the model affects the definition of $l$:

$$l = \min(h_0, 10d), \tag{3.20}$$

which effectively turns off the shock sensor close to solid walls. As privately communicated by Nguyen [164], the reason to do this was a strong interaction between the shock sensor and the inviscid wall boundary conditions, that generated a strong spurious entropy layer at the wall that interfered with the convergence of the solver. A careful review of the literature on artificial viscosity models for DG schemes did not yield any explanation for this, as other authors seemed to be able to compute a variety of inviscid compressible flows without any special treatment on the boundary.

A careful look at the boundary conditions revealed the source of the problem. In the original implementation, the inviscid wall boundary condition was written as:

$$\rho = \widehat{\rho}, \tag{3.21}$$

$$\rho\mathbf{v} - (\rho\mathbf{v}\cdot\mathbf{n})\mathbf{n} = \widehat{\rho\mathbf{v}}, \tag{3.22}$$

$$\rho E = \widehat{\rho E}, \tag{3.23}$$

which effectively forced the extrapolation of density, tangential component of the momentum (by dotting Eq. 3.22 with any tangential vector) and total energy, as well as zero normal mass flux (by dotting Eq. 3.22 with the normal vector). Here $\widehat{(\cdot)}$ denotes trace variables following standard HDG notation.

This boundary condition automatically satisfies the inviscid wall conditions for the Euler equations (namely, zero flux of mass, momentum and energy) in the case where there is no artificial viscosity flux. However, it clearly fails in the case where a shock ends or reflects in a wall. The solution to this is to explicitly impose the physically correct boundary conditions by means of the fluxes rather than the states, using the boundary operator defined in Eq. 2.51. Notice that in this case, the ALE flux also contains a term due to the artificial viscosity.

The use of this particular boundary condition does not appear to be novel. Unfortunately, this is hard to verify as many implementation details are seldom discussed in the literature. In any case, the use of the correct boundary condition for an inviscid wall is enough to discard the switch $l$, and allows setting $l = h$.

### Parameters for the ramp

The last point of study is related to the choice of the parameters $\alpha$ and $\beta$ that govern the shape of the function $f(x)$ in Eq. 3.7 used to apply artificial viscosity in the regions where it is required. A simple inspection of $f(x)$ reveals that the roles of $\alpha$ and $\beta$ are somewhat decoupled. On the one hand, $\alpha$ controls how close $f(x)$ is to $\max(0, x-\beta)$, for which the bound

$$0 < f(x) - \max(0, x-\beta) < (\log 2)/\alpha, \tag{3.24}$$

holds. This bound makes the distance between the functions shrink as $1/\alpha$. On the other hand, $\beta$ controls the value of the sensor $\tilde{s}$ at which the model starts to add artificial viscosity.

The choice of the particular values of $\alpha$ and $\beta$ needs to be driven by some design principle. A reasonable criterion was proposed as a result of some tests carried out with this model in the Project X code [79]. The idea consists in using the value of $f(x)$ at the incompressible point ($\nabla \cdot \mathbf{v} = 0$ or $\tilde{s}^* = 0$) to establish a constraint between $\alpha$ and $\beta$. More precisely, let $C$ denote the value of $f(0)$, then:

$$\frac{\log\left(1 + \exp\left(\alpha(-\beta)\right)\right)}{\alpha} = C \rightarrow \beta = -\frac{\log\left(e^{C\alpha} - 1\right)}{\alpha}. \tag{3.25}$$

In this work the value of $C$ is taken to be small enough ($C \approx 10^{-16}$) so that the artificial viscosity is below the numerical noise for $\nabla \cdot \mathbf{v} \geq 0$. Once $C$ is fixed, a direct relationship between $\alpha$ and $\beta$ is available.

For all the results shown in this thesis, $\alpha = 10^4$ and $\beta = 0.01$. This combination of parameters is used to introduce a small gap in which $\tilde{s}^* > 0$, yet the applied artificial viscosity is small. This is required to recover optimal asymptotic convergence in cases where there are no shocks, but, there exist regions of compression. This claim is verified later in Section 3.3.4 using the convergence in entropy over a smooth subsonic bump.

### 3.1.3 Summary of the artificial viscosity model

The artificial viscosity model reads:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_{\text{Euler}} = \nabla \cdot \epsilon \nabla \mathbf{u}_{\text{AV}}, \tag{3.26}$$

$$\mathbf{u}_{\text{AV}} = (\rho, \rho\mathbf{v}, \rho H)^T, \tag{3.27}$$

$$\tilde{s}^* = -\frac{(k_h h/p)\nabla \cdot \mathbf{v}}{c^*} \tag{3.28}$$

$$\epsilon = \left(k_h \frac{h}{p}\right)\left(\|\mathbf{v}\| + \sqrt{\frac{\gamma \max(10^{-16} P_\infty, P)}{\rho}}\right) f\left(\tilde{s}^*\right), \tag{3.29}$$

$$f(x) = \frac{\log\left(1 + \exp\left(\alpha(x-\beta)\right)\right)}{\alpha}, \tag{3.30}$$

where $p$ denotes the polynomial order and $h = h(\mathbf{x})$ denotes the element size field computed using the linear reconstruction procedure described before. The model is closed by defining the constants $\alpha = 10^4$, $\beta = 0.01$ and $k_h = 1.5$. The particular choice of $k_h$ is determined by some numerical experiments shown next.

The extension of the model to the Navier-Stokes equations is simple and only requires a modification of the left hand side of Eq. 3.26. In particular, $\mathbf{F}_{\text{Euler}}$ needs to be replaced by the Navier-Stokes fluxes $\mathbf{F}_{\text{NS}}$. The boundary conditions in the case of Navier-Stokes do not require any special treatment compared.

The artificial viscosity model can also be applied to the RANS equations. In particular, for the case of the Spalart-Allmaras model, the only change required is the extension of $\mathbf{u}_{\text{AV}}$ to account for the turbulence model equation: $\mathbf{u}_{\text{AV}} = (\rho, \rho\mathbf{v}, \rho H, \rho\tilde{\nu})^T$. It is worth noting that this particular form of artificial viscosity ensures that the jump in $\tilde{\nu}$ is zero across a shock, according to the analysis by Allmaras *et al.* [10].

## 3.2 One dimensional studies

The goal of this section is to perform a series of parameter sweeps (e.g. shock strength, polynomial order, etc.) on a simple 1-D case and use the results to guide the choice of tuning parameters and to validate the artificial viscosity model.

The base problem is that of a stationary shock wave in a tube modeled using the 1-D Euler equations. The setup of the problem is simple: given a Mach number upstream of the shock, the corresponding state behind the shock can be computed using standard normal shock relationships. These two states dictate the boundary conditions, namely, supersonic inflow and subsonic outflow. In between those two conditions, the flow solution can be initialized to a smooth profile that steepens in time until a steady shock wave is formed. For this, the problem is marched in time using a backward Euler formula with increasingly bigger time steps until a steady state solution is found.

In principle, this problem is ill posed due to the ambiguity in location of the shock wave, that could be placed it any point along the 1-D domain and still be a weak solution to the problem. This manifests itself numerically as a divergence of the simulation for very big time steps. The remedy to this is to constrain the shock position in some way. Here, the solver is modified so that the value of the density at the center of the element where the shock profile is initialized is set to the average density across the shock.

The discretization of the problem is performed using a 1-D version of the HDG algorithm. The basis functions for the test and trial space are Lagrange polynomials with equally spaced nodes. All integrals are evaluated using Gaussian quadrature with enough points to ensure exact integration up to order $4p$.

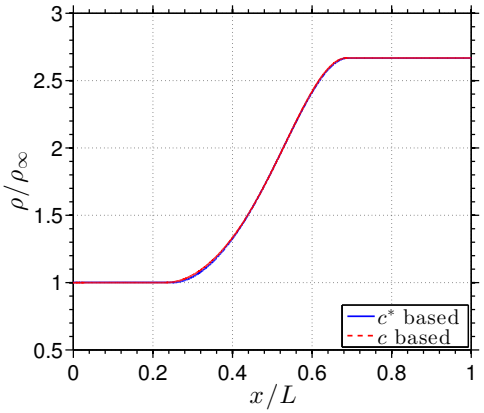
HDG requires the definition of a stabilization parameter for the fluxes at the interfaces. In all the 1-D results presented in this section, Roe's Riemann solver is used. The boundary conditions are imposed using the standard far-field decomposition in 1-D (see Section 2.2.1).

### 3.2.1 Comparison between $c$ and $c^*$ as velocity scale

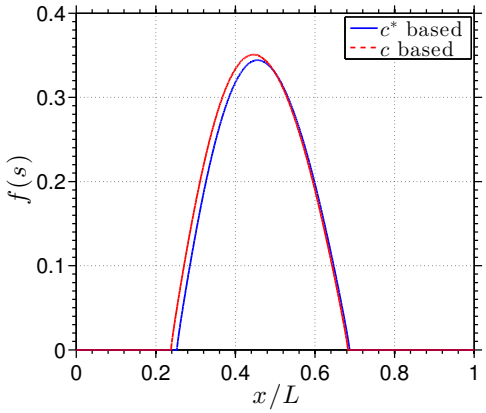
The choice of $c^*$ over $c$ as a suitable scale for the velocity in the shock sensor was motivated by the desire to avoid the variation of $c$ across a shock wave. This has ramifications beyond the fact that $c$ can oscillate and become unphysical. Of particular concern is the sensor bias towards high Mach number regions that in principle grows linearly with the strength of the shock (see Eq. 3.16). In this first 1-D example, the goal is to study this phenomenon by means of a numerical approximation to the actual shock profile computed using both velocity scales.

To this end, a high order solution with polynomials of order $p = 4$ is computed on a fine mesh composed of 600 uniform elements in the domain $x \in [0, 1]$. The length scale field $k_h h(x)/p$ is set to be significantly bigger than the actual one so that the analytical shock profile can be properly approximated. In particular, $k_h h/p = 0.05$, which is equivalent to 30 elements per length scale. The computations are performed for three cases: $M_{1\mathrm{n}} = \{2, 5, 10\}$. The only difference between these cases is the way the shock sensor is computed, namely whether: $s^*$ or $s$ is used in Eq. 3.26-3.30.

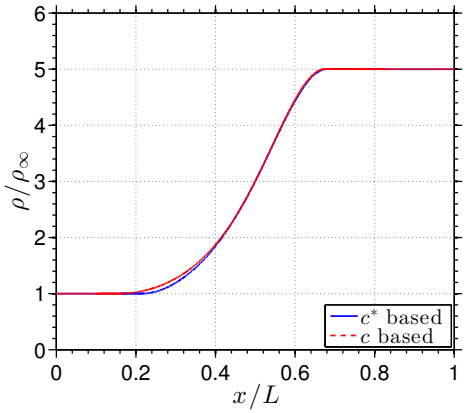
The results are summarized in Figure 3-3 and confirm the hypothesis of the bias. In particular, for strong shocks, the use of $\tilde{s}$ induces an asymmetry in the viscosity that manifests itself in the form of a wider shock profile due to the effect of the PDE. For the higher Mach number tested here ($M_{1\mathrm{n}} = 10$) this effect widens the shock around a 20% compared to the result using $\tilde{s}^*$. The gains of using $c^*$ are thus marginal when measured in these terms. However, they add up to other factors that might come into play. Of significant importance is the sensitivity of $c$ to oscillations in the thermodynamic variables, which requires a regularization of $c$ (as suggested by Nguyen and Peraire [165]), and contributes to the nonlinearity of the problem without a clear benefit.



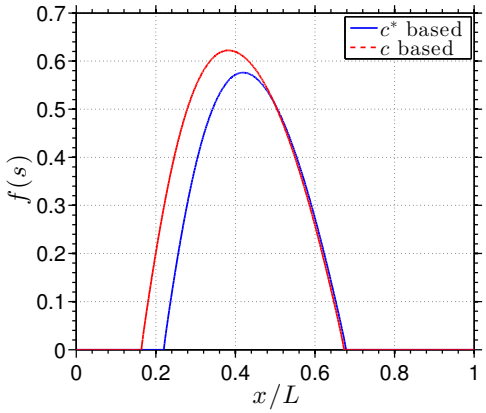

(a) Density, $M_{1n} = 2$



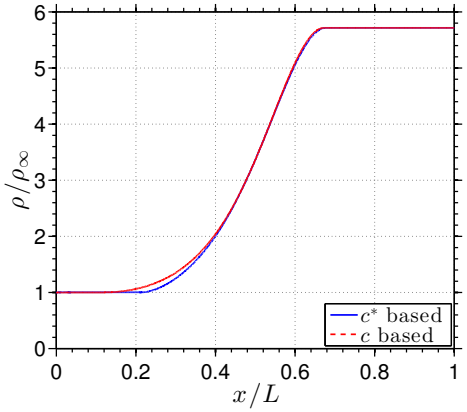

(b) Shock sensor, $M_{1n} = 2$



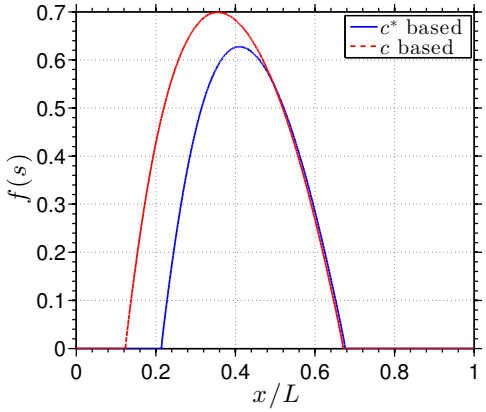

(c) Density, $M_{1n} = 5$




(d) Shock sensor, $M_{1n} = 5$




(e) Density, $M_{1n} = 10$




(f) Shock sensor, $M_{1n} = 10$

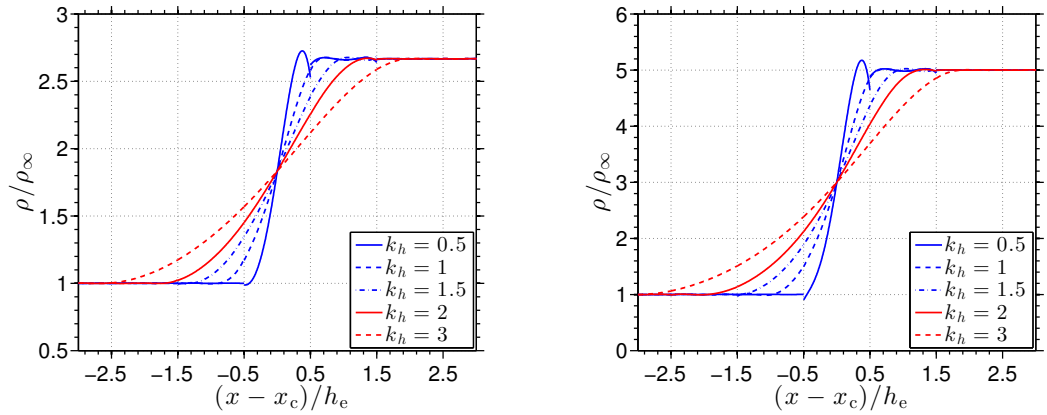
Figure 3-3: Density and sensor profile for different shock strengths: $M_{1n} = \{2, 5, 10\}$, computed using $c$ or $c^*$ as the velocity scale, and a very fine discretization. The results confirm the bias of the $c$-based sensor ($\tilde{s}$) upstream of the shock as it gets stronger.

### 3.2.2 Choice of length scale correction factor

In the previous example, the particular choice of the length scale did not matter much, as the goal was to numerically approximate the analytical shock profile using many degrees of freedom. In contrast, for more realistic cases, the goal is to capture the shock with as few degrees of freedom as possible. In the proposed model, the parameter that controls this property is the length scale correction factor $k_h$. The following results provide a guideline for the choice of $k_h$ using a parametric sweep.

The test cases are based on the simulation of a 1-D steady shock in the domain $x \in [0, 1]$, using 40 elements of order $p = 4$. For this study, only two parameters are varied: $M_{1\mathrm{n}} = \{2, 5\}$ and $k_h = \{0.5, 1, 1.5, 2, 3\}$. The results for all these cases are summarized in Figure 3-4. For the purpose of clarity, the $x$-axis in that Figure is transformed to non-dimensional units around the center of the shock, using the element size $h_\mathrm{e}$ as reference length. Also, the grid lines in the horizontal axis are chosen to align with the actual mesh.

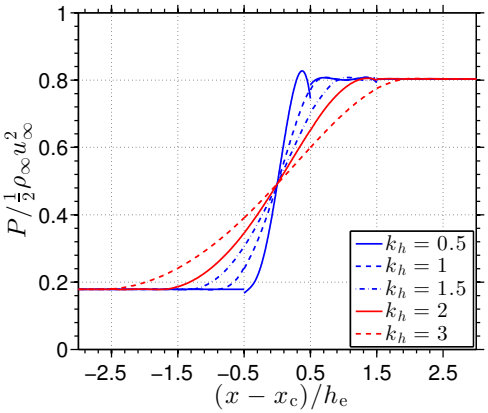
In both cases, the shock profiles agree with the expected behavior in that the shock gets wider as $k_h$ grows. One can readily identify $k_h < 1$ as an unsuitable choice due to the strong oscillations in both pressure and density, as well as the shock sensor itself. Similarly, values in the region $k_h > 2$ are not interesting as they do not show improvement over the $k_h = 2$ solution. The conclusion is that $k_h \in [1, 2]$ is a reasonable choice that only affects the width of the shock.



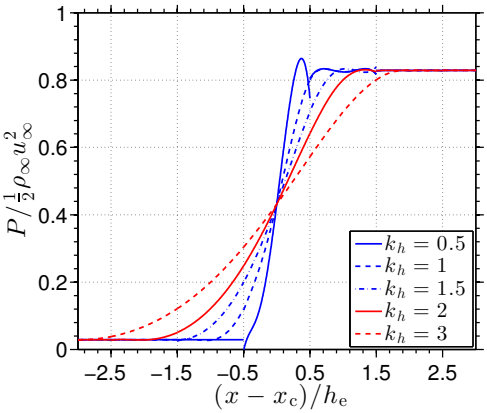

(a) Density, $M_{1n} = 2$



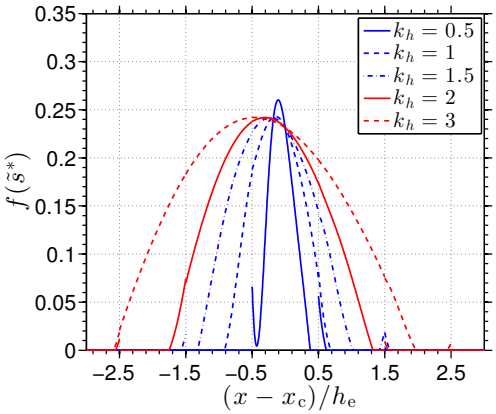

(b) Density, $M_{1n} = 5$



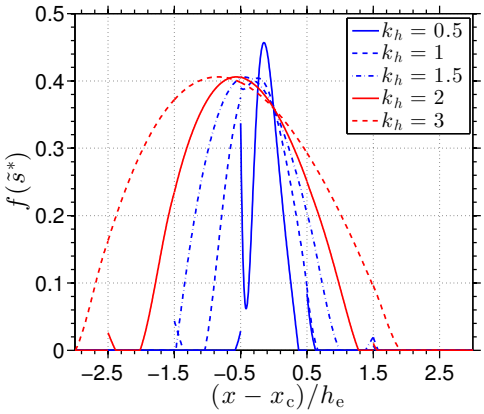

(c) Pressure, $M_{1n} = 2$




(d) Pressure, $M_{1n} = 5$




(e) Shock sensor, $M_{1n} = 2$




(f) Shock sensor, $M_{1n} = 5$

Figure 3-4: Density, pressure and sensor profile around a 1-D shock with $M_{1n} = \{2, 5\}$, computed using different values of the length scale correction factor: $k_h = \{0.5, 1, 1.5, 2, 3\}$ and polynomials of order $p = 4$. For $k_h \in [1, 2]$ the shock profiles are clean of major oscillations.

### 3.2.3 Validation cases

To conclude this section, the artificial viscosity model is applied to a variety of cases in order to assess its behavior in terms of shock strength, polynomial order of the approximation and shock location. This study would ideally explore the whole parameter space. However, to reduce the number of cases and also isolate the different effects, the parameters are independently varied around the reference case: $M_{1n} = 5$, $p = 4$ and shock midpoint located at the center of the element. As in the previous example, the 1-D domain spans $x \in [0, 1]$ and is composed of 40 uniform elements. For all the cases presented here, $k_h = 1.5$.

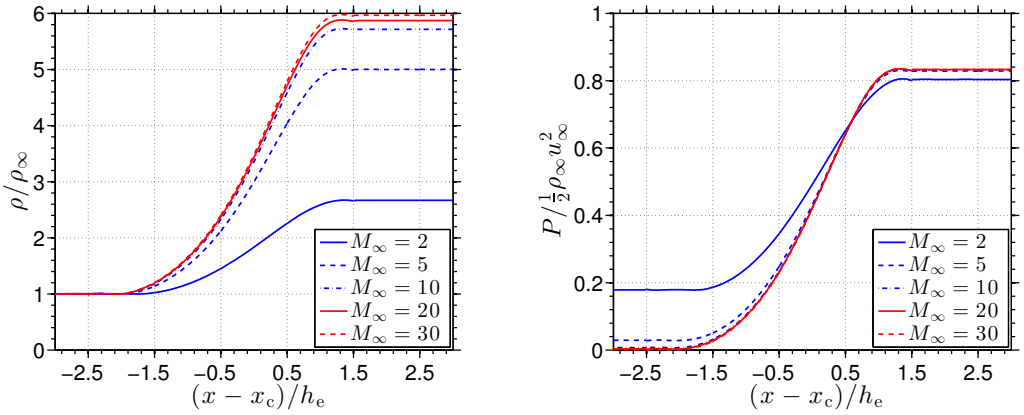
#### Mach number study

The first set of results represents a variation of the shock strength through a variation of $M_{1n}$. In particular, the cases of $M_{1n} = \{2, 5, 10, 20, 30\}$ are computed. The results, which are summarized in Figure 3-5, indicate that the shock thickness is independent of the strength of the shock, which is a desirable feature. More importantly, these results verify the claim that the shock indicator asymptotes to a constant profile as the shock gets stronger, which was one of the reasons to use $c^*$ as velocity scale. Notice that for the strongest shocks, the sensor presents a spurious spike outside the shock profile at the interface with the neighboring elements. This is due to the fact that the analytical sensor profile (shown in Figure 3-3) presents a discontinuity in the slope that triggers small oscillations in the sensor itself.

#### Polynomial order study

All the results presented up to this point were computed using polynomials of order $p = 4$. While this value of $p$ is representative of most of the cases used in this thesis, it is by no means an upper bound on how high the solver or the model can go. To prove it, this section contains results for a range of polynomial orders ($p = \{2, 3, 4, 5, 6, 7, 8\}$) on the standard case ($M_{1n} = 5$ and shock centered in the mid element). The results are summarized in Figure 3-6 and show how the solution is clean and free from oscillations

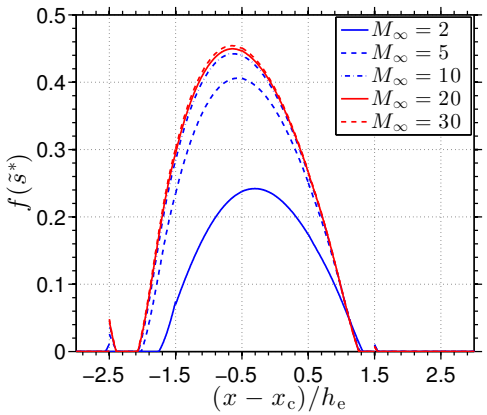
up to $p = 8$. Furthermore, the shock thickness scales like $1/p$ as expected from the definition of the length scale (see Eq. 3.9). In that respect, the value $k_h \in [1, 2]$ derived using solutions with $p = 4$ seems to be independent of the approximation order.




(a) Density

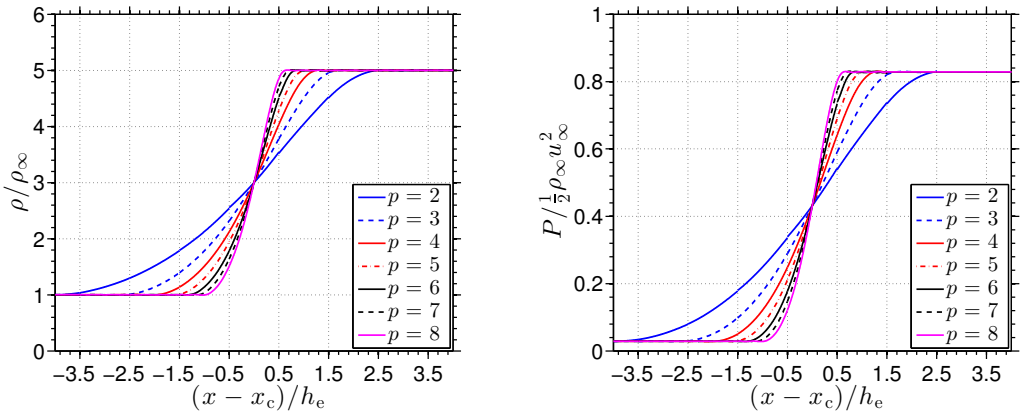



(b) Pressure



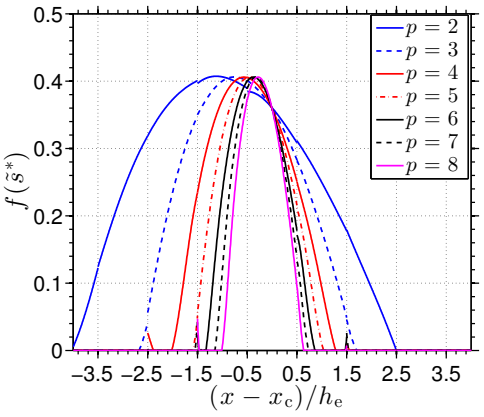

(c) Shock sensor

Figure 3-5: Density, pressure and sensor profile computed using different values of the upstream Mach number: $M_{1n} = \{2, 5, 10, 20, 30\}$. Here, $k_h = 1.5$ and $p = 4$. The shock indicator asymptotes to a constant profile that is independent of the strength of the shock as intended.




(a) Density

(b) Pressure



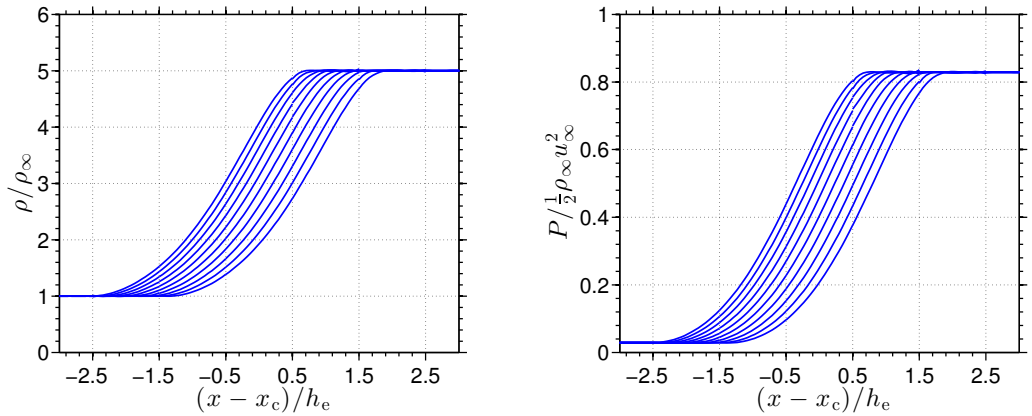

(c) Shock sensor

Figure 3-6: Density, pressure and sensor profile computed using different polynomial orders: $p = \{2, 3, 4, 5, 6, 7, 8\}$. In all these runs $M_{1n} = 5$ and $k_h = 1.5$. The results are clean of oscillations in the primal variables for all the orders of approximation.

### Shock center study

All the cases shown previously require a numerical fix to avoid ill-posedness. Namely, that the value of the density at one point in the domain is equal to the average value across the shock. This removes the degree of freedom associated to the location of the shock and makes the steady state problem solvable. For simplicity, this point has been made to coincide with the center of an element, which is not generally true in more complex cases.

The purpose of the next case is to address this concern within the limitations of this simple 1-D case. For this, the reference case ($x \in [0, 1]$, $M_{1n} = 5$, $p = 4$) is modified so that the midpoint of the shock is no longer placed in the center of an element, but rather can be moved within it. The results in Figure 3-7 indicate that the shock midpoint location does not affect the shock profile, which is translated smoothly without any major oscillation.




(a) Density profile

(b) Pressure profile

(c) Shock sensor

Figure 3-7: Density, pressure and sensor profile computed using different locations for the shock midpoint along an element. For these cases, $M_{1n} = 5$, $p = 4$ and $k_h = 1.5$. The results indicate that the shock profile is not affected by the particular location of the shock midpoint within one element.

## 3.3 Two dimensional studies

This section is composed of a few numerical experiments in 2-D using the proposed artificial viscosity. The goal is to prove that the model can be extended to multiple spatial dimensions using both isotropic and anisotropic meshes. All the cases presented here are inviscid, hence modeled using the Euler equations. The use of this shock capturing strategy in viscous flows is postponed to the next chapter where boundary layer adaptivity is discussed.

The discretization of the problem in 2-D is based on a standard version of the HDG solver for triangular unstructured meshes [180]. In it, the basis functions for the test and trial spaces are nodal (interpolant) polynomials of order $p$. Due to the moderate order of approximation used here ($p \leq 4$), equispaced nodes are used, although higher order discretizations would benefit from more sophisticated node placement [97]. All the integrals in the problem are evaluated using Gaussian quadrature with enough points to ensure exact integration up to order $4p$. The stabilization matrix at the interfaces is based on the global Lax-Friedrichs scheme. Following the results presented in the previous section, $k_h = 1.5$ is used.

All the results presented here are converged to steady state by advancing a time dependent solution using a backward Euler formula. At each time step, a system of nonlinear equations is solved using Newton's method combined with a backtracking line search algorithm. The time step selection rule proceeds as follows: if the number of Newton iterations required to converge the current step is less than or equal to the number of Newton iterations required to converge the previous step, then the time step is increased by a factor of 2. Otherwise, it is reduced by the same factor. In addition, if an exception is found (singular Jacobian, divide by zero, etc.), the nonlinear solver is aborted and the time step is reduced by 10. Finally, when the time step grows beyond a given value (usually 20 convective times for the geometry of interest), a steady state nonlinear solve is performed.

Despite its simplicity, this time step selection rule has proved to be fairly robust.

As expected, there is a price to pay in that it is not optimal and might require extra computational time compared to more advanced time relaxation schemes like pseudo transient continuation [114].

### Anisotropic adaptation

In most practical applications, once a steady state solution has been obtained, the mesh is adapted to maximize the approximation properties of the scheme. In the case of compressible flows, such adaptation introduces anisotropic elements in the mesh that align with shocks and other flow features.

This possibility was taken into account in the current model through the way $h_e$ is related to the lower height of the element; however, it remains to be verified in multiple dimensions. For this, the model is tested on anisotropic meshes that are generated with the help of the BAMG mesh generator [94].

In particular, BAMG can take as input a continuous piecewise linear solution on a given mesh, and by reconstructing the Hessian of such solution, generate a new mesh that minimizes the interpolation error, subject to constraints like the number of vertices of the triangulation, the maximum anisotropy in the mesh or the minimum element size.

The input data for BAMG is generated from a high order solution by extracting the solution at the corners of each element (ignoring the rest of the nodal values) and averaging out the contributions from all the elements that share a vertex of the triangulation. This procedure yields a continuous piecewise linear approximation to the solution that approximates the sharp gradients around shocks as desired. Needless to say that other procedures could be used to generate the input data like a projection onto a linear basis functions or a refinement of the mesh to take into account all the nodal values of the solution.

In any case, the proposed methodology for adaptation is enough to generate anisotropic meshes in which the artificial viscosity model can be tested.

### 3.3.1 Supersonic flow in a duct with a ramp

The first case of study deals with the supersonic inviscid flow at $M_{\text{in}} = 1.5$ in a duct with straight sided compression ramp of 5 degrees inclination. In this case, both a shock wave and an expansion fan appear with limited interaction between them. The flow is constrained by the ramp itself as well as an upper wall that makes the waves reflect. A sketch of the problem setup is depicted in Figure 3-8.

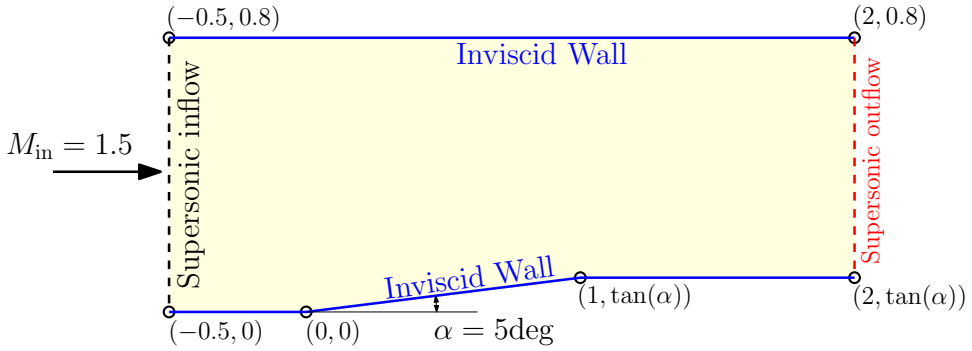



Figure 3-8: Sketch of the supersonic ramp geometry and boundary conditions.

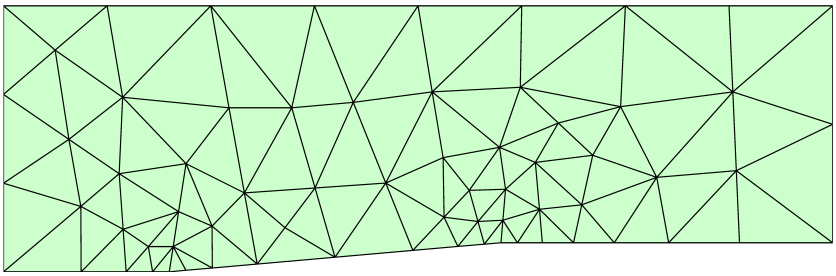
The simulation starts from an isotropic mesh with minor refinement at the corners of the ramp (see Figure 3-9a). At each step, a steady state solution is computed using polynomials of order $p = 4$, then BAMG is used to generate a new mesh. The final mesh, obtained after 9 iterations, is clearly adapted to capture the shock bouncing off the walls as well as the expansion wave at the second corner of the ramp (see Figure 3-9c).

The evolution of the shock sensor with the mesh refinement is depicted in Figure 3-10. The fact that the maximum sensor value asymptotes to a constant confirms that it is grid-independent. A plot of the Mach number and the entropy computed on the finest mesh is contained in Figure 3-11. Notice how the shock waves and expansion fans are captured in a clean fashion, especially when looking at the Mach number.

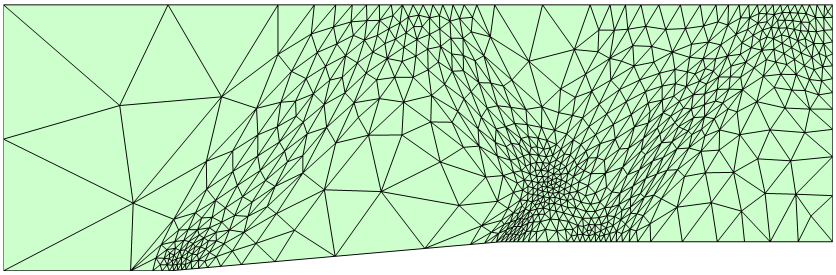
Finally, the pressure signature in the upper and lower wall for both the initial and final mesh is plotted in Figure 3-12. As expected, an adapted mesh with stretched elements yields sharper results.

Albeit simple, this test case highlights the importance of adaptivity and anisotropy,

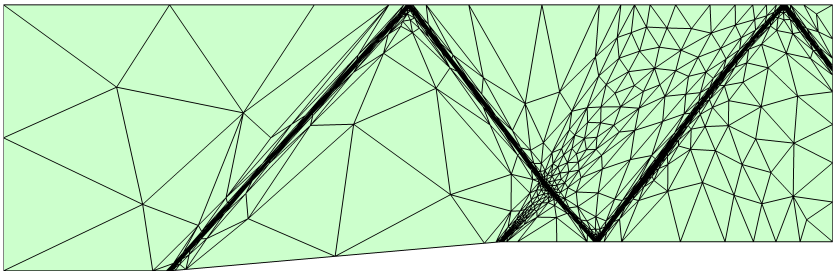
and how the proposed artificial viscosity model can perform well in that environment.



(a) Initial mesh, 102 elements

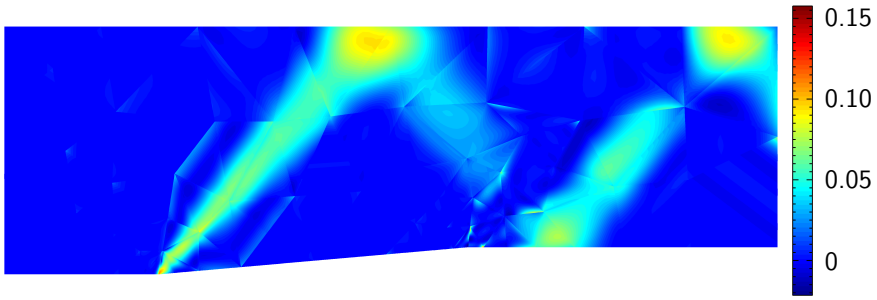


(b) Adapted mesh after 2 iterations, 1324 elements

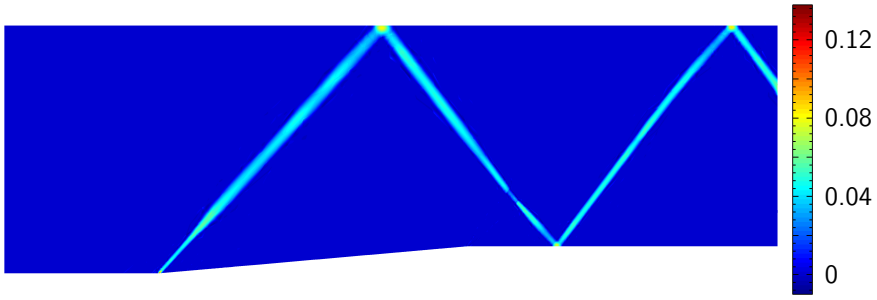


(c) Adapted mesh after 9 iterations, 2439 elements

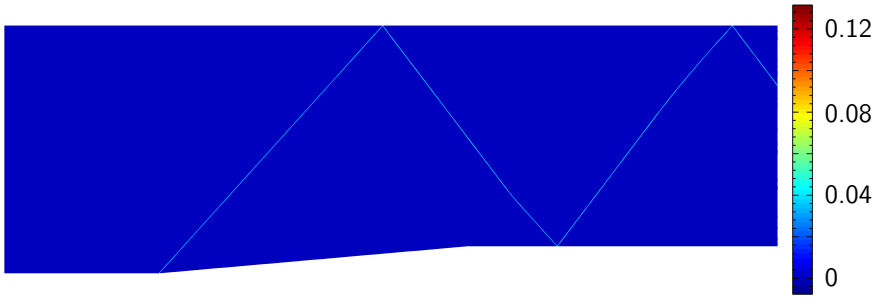
Figure 3-9: Evolution of the mesh with the anisotropic adaptivity iteration, for the case of the supersonic compression ramp.




(a) Shock sensor on initial mesh

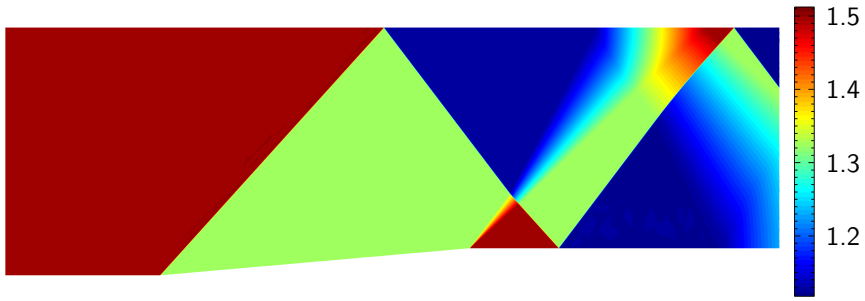



(b) Shock sensor after 2 iterations of anisotropic refinement

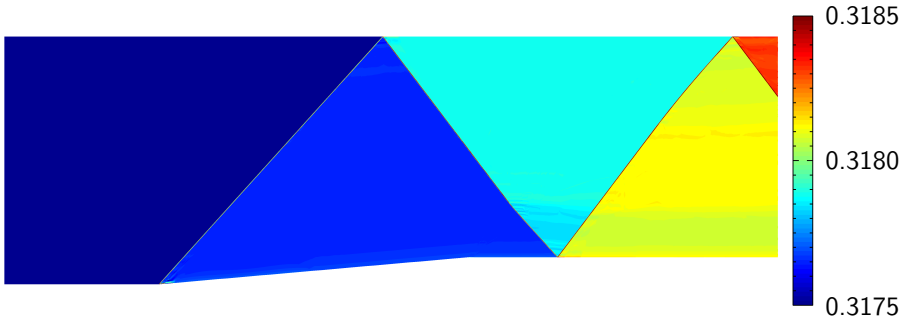



(c) Shock sensor after 9 iterations of anisotropic refinement

Figure 3-10: Evolution of the shock sensor with the anisotropic adaptivity iteration. The proper scaling of the shock removes the dependency of the sensor on the grid.

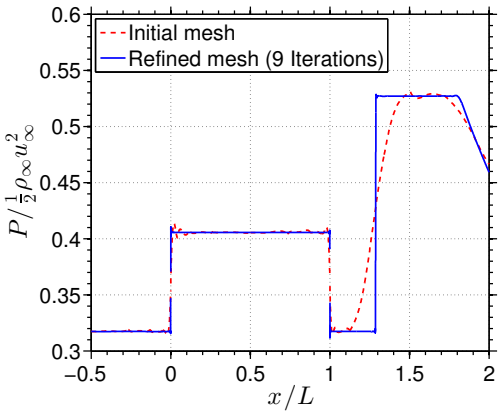



(a) Mach number

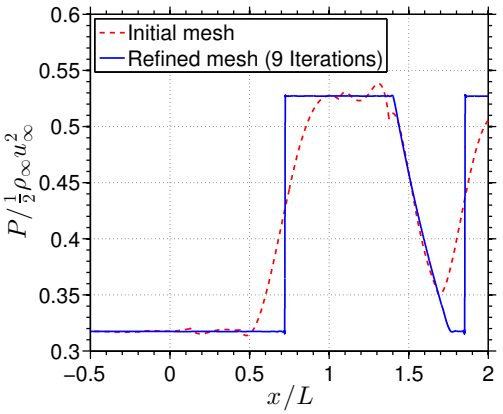



(b) Entropy

Figure 3-11: Mach number field and entropy field around a supersonic ramp computed using polynomials of order $p = 4$ on the final mesh. The combination of anisotropic adaptivity and artificial viscosity yields sharp shocks with a limited use of degrees of freedom.




(a) Pressure signature on the bottom wall

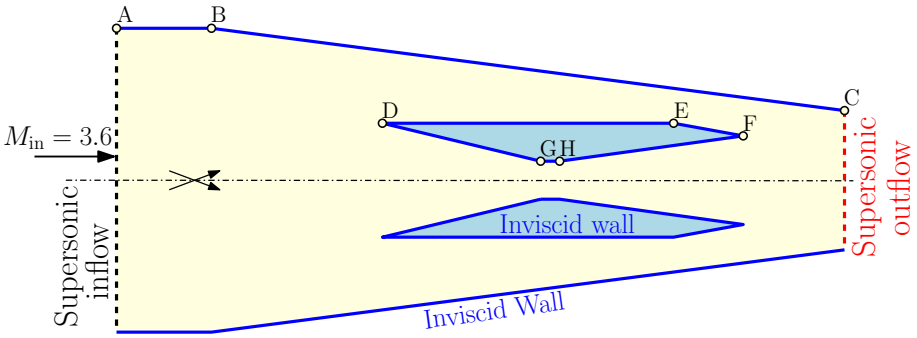



(b) Pressure signature on the top wall

Figure 3-12: Comparison of the pressure signature in the top and bottom wall. The results stress the sub-cell shock resolution of the scheme and the importance of adaptivity.

### 3.3.2 Supersonic flow inside a Scramjet geometry

The next case of study deals with the supersonic flow inside a Scramjet at $M_\infty = 3.6$. The surrogate 2-D geometry of the scramjet is composed of two inner bodies inside a duct, all of which are straight sided [130]. The geometry is symmetric across the horizontal axis although this is not enforced in the meshes. A detailed description of the geometry is contained in Figure 3-13. This configuration produces a variety of shock waves and expansion waves that interact with each other to form a complex pattern.




(a) Sketch of the scramjet geometry and boundary conditions.

| Point | $x$ | $y$ |
|---|---|---|
| A | -0.5 | 0.8650 |
| B | 0 | 0.8650 |
| C | 3.9899 | 0.3986 |
| D | 1.0880 | 0.3460 |
| E | 2.9410 | 0.3460 |
| F | 3.3940 | 0.2811 |
| G | 2.1071 | 0.1211 |
| H | 2.1998 | 0.1211 |

(b) Location of the vertices of the geometry in space (for lower half, assume symmetry).

Figure 3-13: Problem description for the case of the scramjet flow. Notice the geometry is symmetric along the horizontal axis. This condition is not enforced in the mesh nor the flow.

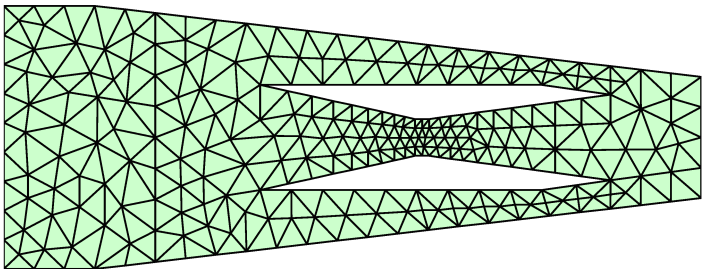
As in the previous case, the initial coarse mesh is isotropic with a moderate refinement at the corners. At every step, a steady state solution is computed that is passed

along to BAMG that returns a new mesh. All the intermediate solutions required in this process are computed directly using polynomials of order $p = 4$ without any continuation in the polynomial order. A total of 12 adaptivity cycles are performed. A sample of the mesh evolution is summarized in Figure 3-14. The flow field computed on the finest mesh is depicted in Figure 3-15. The results are satisfactory for several reasons.

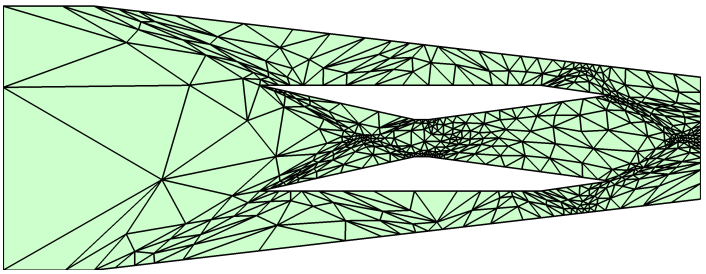
First, they prove that the flow field remains symmetric even if the mesh is not. This is a desirable property that helps avoid mesh dependency on the final solution and also implies that the scheme is robust at handling meshes that do not need to be aligned with the shocks.

Secondly, they show that the artificial viscosity is properly targeting the shocks and not other flow features like contacts discontinuities or expansion waves. One place where this is visible is the region around the wall in the nozzle throat. A detail of the flow there is found in Figure 3-16. Notice how the shock sensor is only active in the shock wave and is turned off when this intersects the expansion fan (which being so close to the corner is the dominant term in the dilatation). Something similar happens at the trailing edge of the inner body, where the shock sensor is not activated by the presence of a slip line (see Figure 3-17).

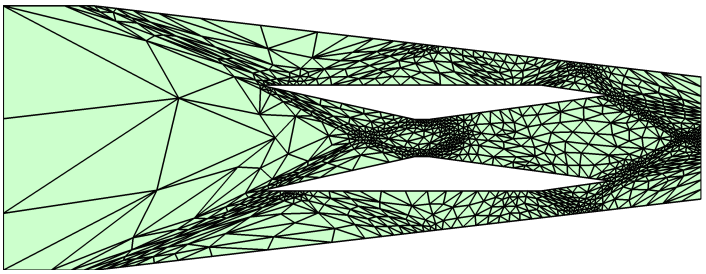
Finally, these results highlight how flexible the artificial viscosity is in terms of mesh resolution, being able to compute results in very coarse meshes as well as refined anisotropic ones. This means that the scales that govern this shock capturing strategy are well balanced and follow the resolution available in the mesh.



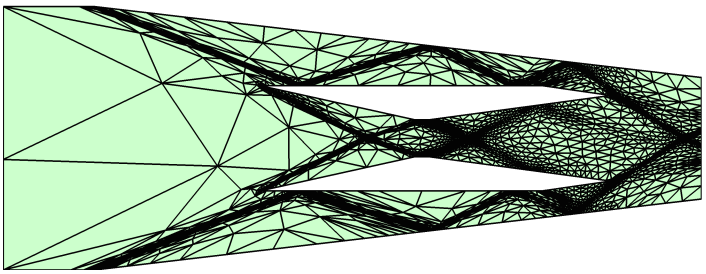
(a) Initial mesh, 461 elements



(b) Adapted mesh after 4 cycles, 898 elements

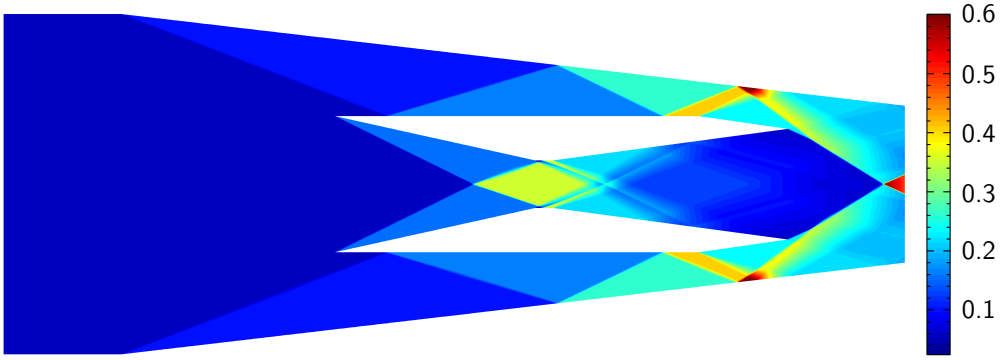


(c) Adapted mesh after 8 cycles, 1873 elements

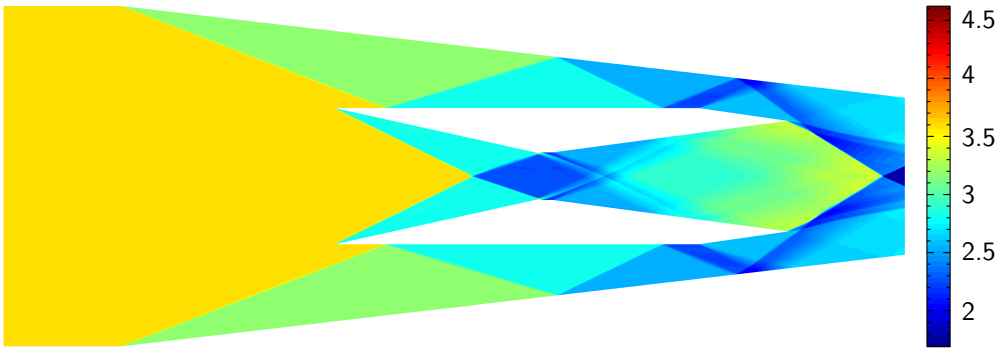


(d) Adapted mesh after 12 cycles, 6509 elements

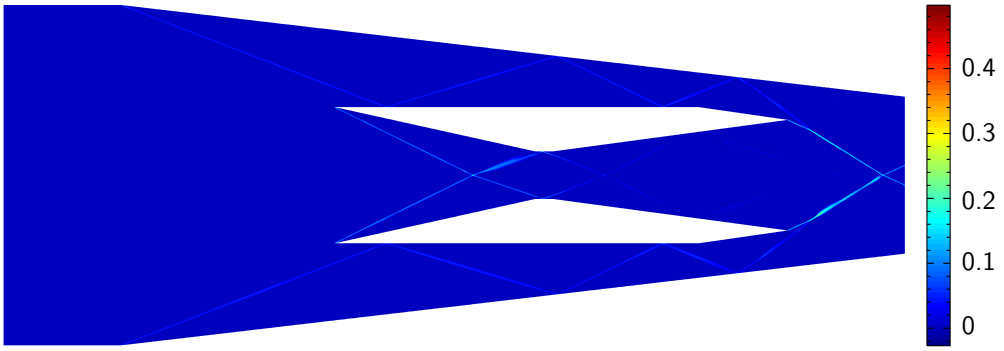
Figure 3-14: Evolution of the mesh for the scramjet flow as a function of the number of anisotropic adaptivity iterations.




(a) Pressure

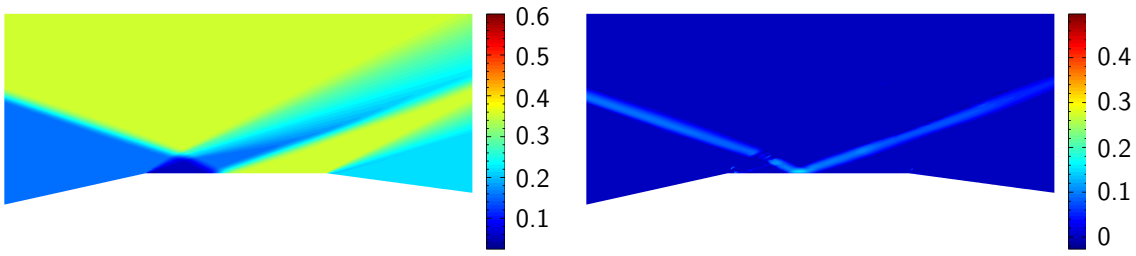



(b) Mach number



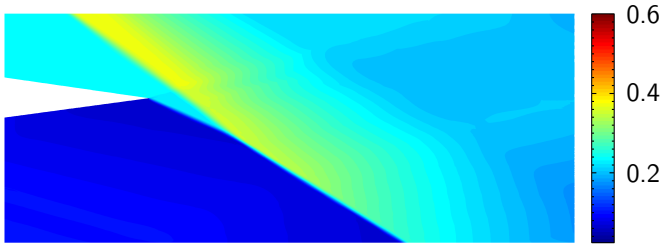

(c) Shock sensor

Figure 3-15: Flow inside the scramjet geometry computed on the finest mesh (12 adaptivity cycles) using polynomials of order $p = 4$. The combination of adaptivity and the sub-cell shock resolution of the proposed scheme can separate very fine details in the flow. Furthermore, there is no loss of symmetry despite the fact that the meshes are not forced to be symmetric. This indicates that the method is robust to misalignments of the mesh.

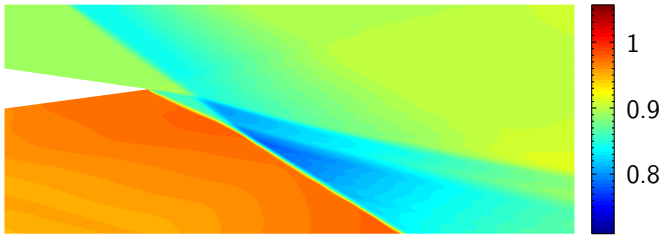



(a) Pressure (b) Shock sensor

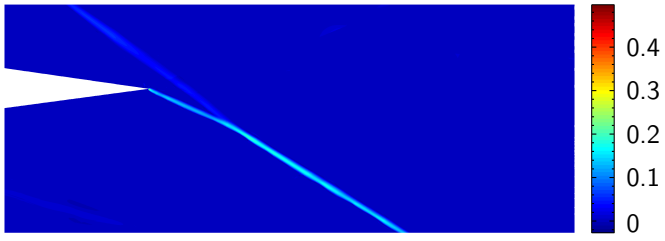
Figure 3-16: Detail of the flow around the nozzle throat. The artificial viscosity is only applied in the shock region and is turned off at the intersection with the expansion fan.




(a) Pressure




(b) Magnitude of the velocity




(c) Shock sensor

Figure 3-17: Detail of the flow around the trailing edge of the inner body. The presence of a contact discontinuity, which is a purely vortical feature, does not trigger the shock indicator.

### 3.3.3 Flow around a forward facing step

In this test case, the shock capturing scheme is used to simulate the supersonic flow at $M_\infty = 3$ around a forward facing step. This is a well-known validation case that was extensively used by Woodward and Collela [258] to compare different finite volume schemes. While originally intended as an unsteady test case, the flow reaches a steady state that contains very rich physics. This will help assess the combination of the artificial viscosity with anisotropic adaptivity in a different setting. The particular geometry used here is described in Figure 3-18 and has been slightly modified with respect to the original one. In particular, the corner of the step, that represents the vertex of a very strong expansion fan, has been rounded to a radius of 1% of the total height so that the singularity is removed. This is similar in spirit (although not in form) to the modified discretization that Woodward and Collela used in the corner [258], or the viscosity that some schemes based on a resolution indicator add there [101, 186].

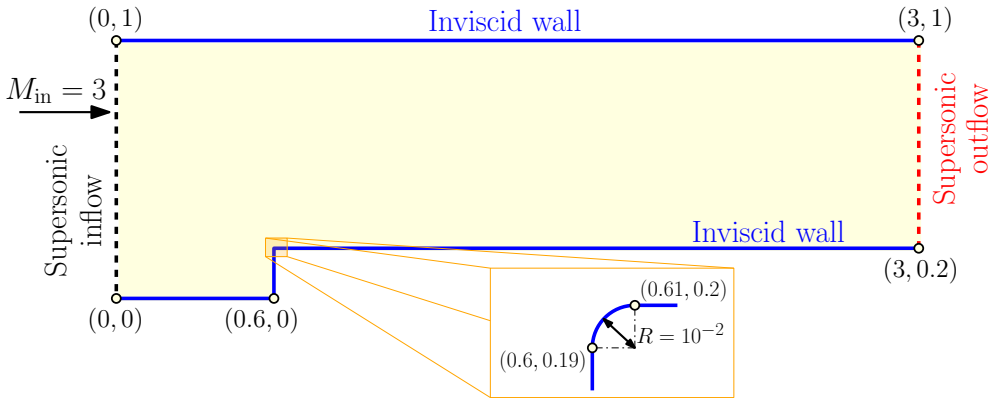
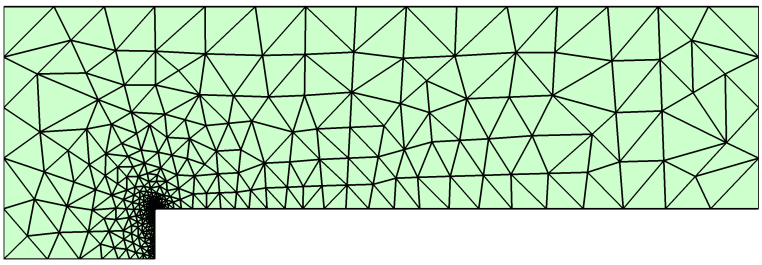



Figure 3-18: Sketch of the geometry and boundary conditions for the forward facing step problem.

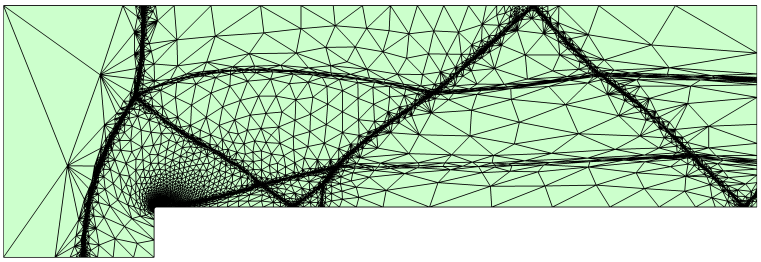
The solution to this flow is computed on a series of meshes generated with BAMG using polynomials of order $p = 4$. In total, 12 cycles of adaptation are enough to produce a final mesh like the one shown on the bottom of Figure 3-19. For reference purposes, the initial mesh is plotted in the same figure. The solution field for the pressure, Mach number and shock sensor are plotted in Figure 3-20. Notice how the

different flow features are properly described within the resolution available in the mesh. This is clearly the case for shock waves as well as contact discontinuities. In particular, one can readily observe the presence of a triple point in the bow shock that leads to a contact discontinuity as well as a reflected shock. The latter then interacts with the over-expansion around the corner to generate a complex flow pattern that includes a weak normal shock, and two additional triple points.

Two out of the three contact discontinuities are picked up by the anisotropic refinement along the adaptation cycle as can be seen in the final meshes (see Fig. 3-19b). However, the weakest one, associated to the merging of the reflected shock and the weak normal shock close to the wall is not. This problem depends on the initial mesh used and could be fixed by carefully tuning the parameters that govern the mesh generation. However, this is not within the scope of this thesis.

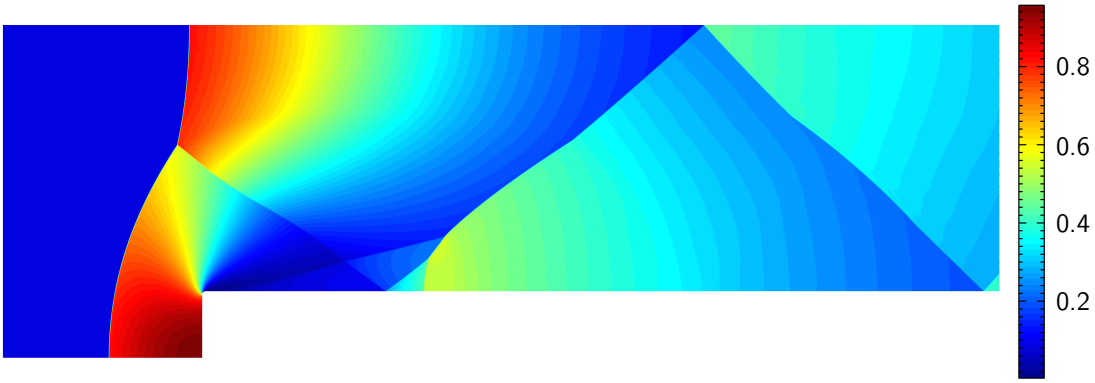


(a) Initial mesh, 677 elements

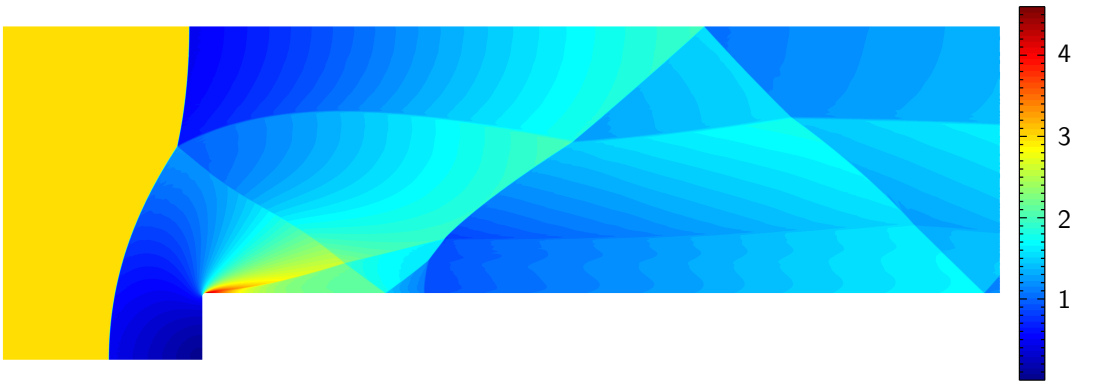


(b) Adapted mesh after 12 cycles, 9584 elements

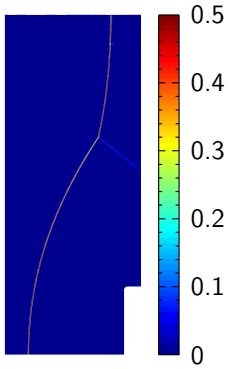
Figure 3-19: Evolution of the mesh for the forward facing step flow as a function of the number of anisotropic adaptivity iterations.



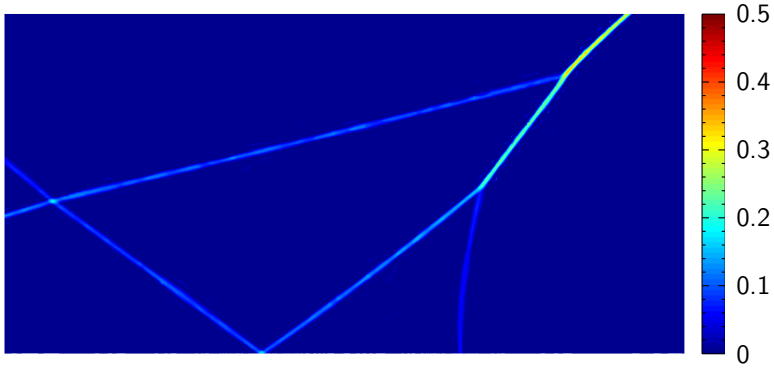

(a) Pressure




(b) Mach number




(c) Detail of the shock sensor




(d) Detail of the shock sensor

Figure 3-20: Flow solution for the forward facing step problem computed using polynomials of order $p = 4$ on the finest mesh (after 12 refinement iterations). Sharp flow features (e.g. shock waves, slip lines, etc) are properly captured by the mesh adaptivity and the artificial viscosity model, while smooth regions benefit from the high order approximation.

As described before, these cases were run to steady state using a BDF1 discretization in time and an aggressive time step selection rule. For that reason, the Kelvin-Helmholtz instability that usually appears in the upper contact discontinuity is totally suppressed. To compare with the unsteady results by Woodward and Collela [258] would require finer isotropic meshes as well as a high order time-stepping scheme (e.g. higher order BDF schemes or DIRK schemes [6]), with smaller time steps in the order of $CFL = 1$. In principle, and given the steady state results used $CFL \gg 1$, this should not be a problem for this model.

### 3.3.4 Effect of the artificial viscosity on the convergence of the subsonic flow over a smooth bump

The previous results have shown that the artificial viscosity model captures shocks as intended. However, the effect of the model in the smooth regions of the flow (which generally represent the majority of the domain) still needs to be assessed. For this, the method is applied to the C1.1 case of the 1st International Workshop on High-Order CFD Methods [251]. The case deals with the subsonic inviscid flow in a channel with a smooth bump on the lower surface. The geometry and boundary conditions are described in Figure 3-21. For this particular case, the analytical solution remains isentropic in the whole domain; however, the numerical approximation tends to introduce small oscillations in the entropy close to the lower wall. Once generated, these oscillations persist downstream and induce a non-negligible error in the entropy norm, which is measured as:

$$||s - s_\infty||_2 = \sqrt{\frac{\int_\Omega \left(1 - \frac{P/\rho^\gamma}{P_\infty/\rho_\infty^\gamma}\right)^2 dV}{\int_\Omega dV}} \tag{3.31}$$

The interest here is on quantifying how the artificial viscosity affects this error. For this, the solution on a sequence of structured isoparametric meshes using polynomials of order $p = \{1, 2, 3, 4\}$, with and without the artificial viscosity model are computed

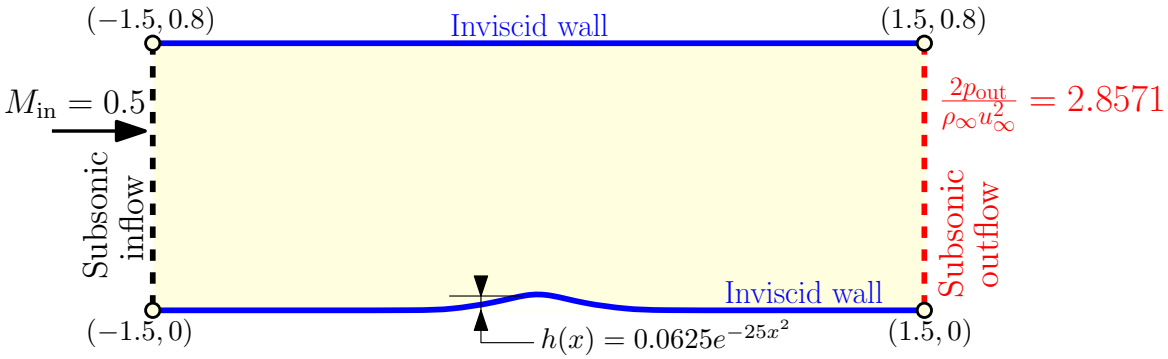



Figure 3-21: Sketch of the geometry and boundary conditions for the case of the smooth subsonic bump.

and compared in terms of convergence in the entropy norm.

The mesh generation at each stage of refinement starts from a structured mesh of quads for the bounding rectangle that is broken along the SE-NW diagonal to generate a structured mesh of triangles. The high order nodes are placed on the rectangular mesh and then transformed analytically by means of a linear blending between the functions that describe the lower and upper wall so that their new coordinates obey:

$$x' = x \tag{3.32}$$

$$y' = y + (1 - y/0.8)0.0625e^{-25x^2}. \tag{3.33}$$

Using this construction, a sequence of meshes starting from a $6 \times 2$ mesh can be constructed by just doubling the number of elements in each direction at each stage.

The solutions are computed using the standard HDG solver. Given the simplicity of the flow, a direct steady state solve is performed without the need for time-stepping. For the cases with artificial viscosity, the standard parameters are used, namely $\alpha = 10^4$, $\beta = 0.01$ and $k_h = 1.5$. A sample of the mesh together with the solution (Mach number and entropy) is compiled in Figure 3-22. Notice how the Mach number field looks clean, however, there exist some oscillations in the entropy. These are the target of the present study.

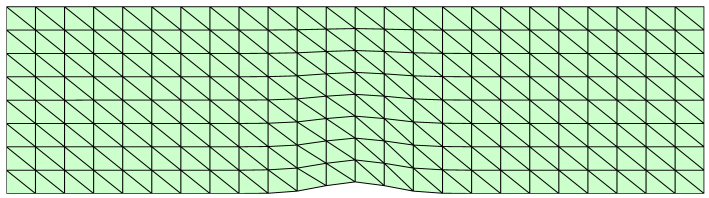
The convergence in the entropy error for all the mesh sequences is plotted in

Figure 3-23. In there, the horizontal scale measures the square root of the number of elements in the mesh, which is equivalent to a measure of the element size. The solid lines denote results computed using artificial viscosity while the dashed lines denote results without it. Notice that for the coarse meshes (right half of the plot) there is an effect of the artificial viscosity on the entropy production, which is less important as the polynomial order is reduced. In particular, for the case of $p = 1$, the effect of adding an artificial viscosity term of order $h^2$ barely affects the entropy error. This suggests that the equivalent dissipation of the original HDG scheme is order $h^2$ or lower for $p = 1$, and greater than $h^2$ for $p \geq 2$.

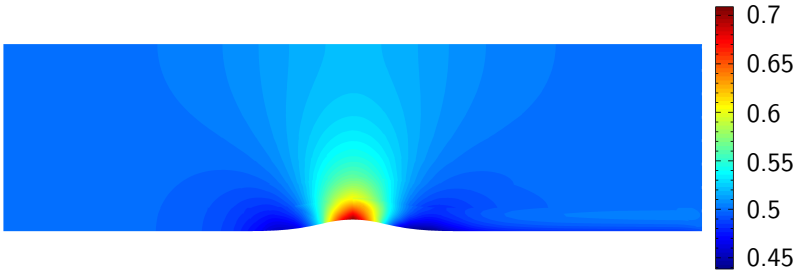
As the mesh is refined, there is a point in which the gap introduced by the choice of $\alpha$ and $\beta$ (see Section 3.1.2) takes effect and the results with and without artificial viscosity coincide. This is due to the fact that the flow might contain smooth regions where dilatation is negative but finite. In these regions, $\tilde{s}^*$ decreases as the mesh is refined, and, at some point, is effectively turned off. This is clearly seen on the left half of Figure 3-23. A similar pre-asymptotic behavior was documented by Barter and Darmofal for another bump geometry using a variety of shock indicators with and without their PDE smoothing [251].

All in all, these convergence tests have been performed using other stabilization terms (in particular the local Lax-Friedrichs and Roe's scheme) as well as other boundary conditions for the inviscid wall and the subsonic inflow and outflow, without any noticeable difference.

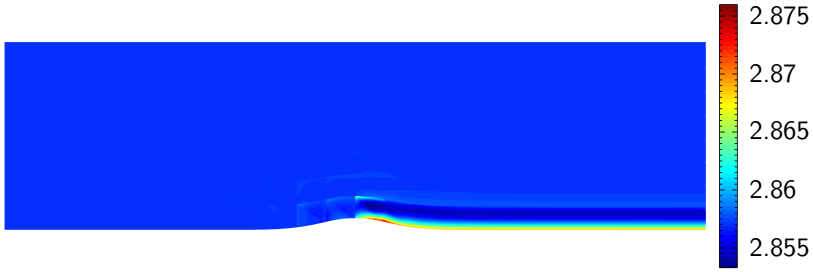
This study indicates that the proposed scheme converges to the inviscid solution for flows without discontinuities. For practical cases, however, there exists a pre-asymptotic range that is governed by the finite value of $\nabla \cdot \mathbf{v}$ in certain parts of the flow. This pre-asymptotic behavior can be reduced by carefully adjusting $\beta$.



(a) Mesh




(b) Mach number




(c) Entropy

Figure 3-22: Sample mesh and solution for the subsonic flow over a smooth bump. The numerical solution shows oscillations in the entropy around the bump that can be quantified to assess the effect of the artificial viscosity on a smooth flow.

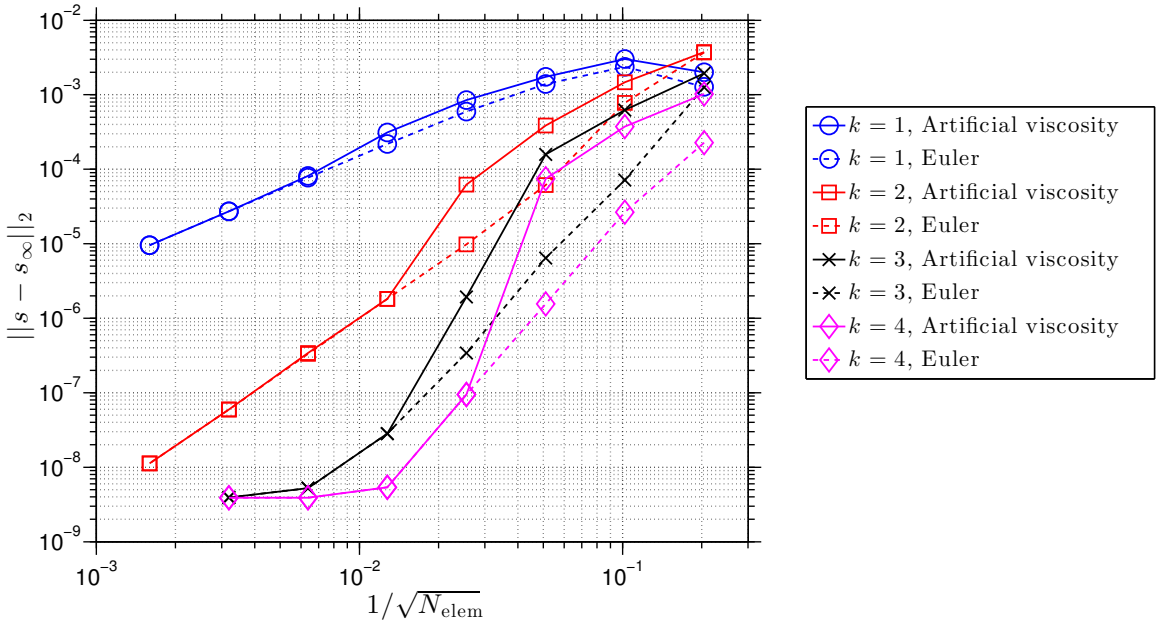



Figure 3-23: Comparison of the convergence in the entropy error for different polynomial orders using structured triangular meshes. The results using artificial viscosity show a pre-asymptotic behavior that seems to limit convergence between 2nd and 3rd order. This effect disappears once the element size is small enough.

## 3.4 Conclusions and future work

This chapter described a shock capturing strategy for high order methods based on artificial viscosity and the dilatation of the flow as a shock indicator. The model has been extensively tested in 1-D and 2-D in combination with anisotropic adaptive refinement. The results indicate that the model can capture shocks in a variety of flows and does not interact with other flow features such as expansion fans or contact discontinuities, which in theory do not require stabilization.

In comparison with the state of the art, the proposed model has several advantages. First of all, it does not require the use of an extra equation to solve for the artificial viscosity field [21] since it produces artificial viscosity fields that are essentially $\mathcal{C}^0$ (up to the approximation space of the solution), thus in agreement with the conclusions of Barter and Darmofal [21] and without the need for reconstructions that either limit the $CFL$ number when treated explicitly, or widen the stencil when treated implicitly.

Indeed, the proposed model is fully analytical and can be implemented in any solver where gradients of the solution are available or can be computed. This is crucial to deliver steady state solutions at a reasonable cost, especially in the case where viscous effects (either laminar or turbulent) are taken into account.

Of course, the model also has its weaknesses. For example, the need for gradients of the solution that can be of low quality (suboptimal order of convergence) for certain discretization schemes and incur an extra cost in the computation. Neither of these is the case in the HDG scheme, which makes the pairing with this artificial viscosity model very attractive.

Also, there is the problem of the nonlinearity of the model, which might have a strong impact on the convergence of the solver. In this work, the use of a BDF1 to march to steady state has proved very robust, however, the implementation in other relaxation schemes (e.g. PTC or variations of it) might require further study.

Last but not least is the pre-asymptotic behavior found in smooth regions, that might degrade the accuracy of the scheme in regions where the mesh is coarse, e.g.

when using anisotropic adaptivity. The solution for this in the case of steady state flows would be to post-process the solution in a similar fashion to what Persson proposed [186], so that the artificial viscosity is turned off in regions away from shocks by setting $h(\mathbf{x}) = 0$ there. The sensing of the shock could be based on the non-dimensional dilatation $\tilde{s}^{\star}$, the resolution indicator of Persson and Peraire [188] or a combination of both. A more cynical view would be to totally disregard this issue as the overall error is determined by the presence of the shock at least in the $L^1$ norm.

The model could be enhanced in a variety of ways. First, other definitions of $\mathbf{u}_{\mathrm{AV}}$ or $\mathbf{G}$ could yield sharper shocks or a better behavior at high Mach number [188]. Also, the whole definition of the shock length scale could be revisited to try other reconstructions for the $h$ field as well as more sophisticated approaches based on a tensor field rather than a scalar one. The later would make sure that the dilatation is measured in the metric induced by the element so that misalignments with the shock could be better identified. All these subjects will be the focus of future research.

# Chapter 4

# $r$-adaptivity for viscous flows

## 4.1 Introduction

This chapter describes the adaptivity strategy for viscous flows proposed as part of this thesis.

As with any adaptive strategy, the goal is to produce a mesh where the degrees of freedom are placed to minimize the error in the solution or in an output of interest (e.g. lift, drag, etc.). In general terms, adaptive strategies can be categorized into three groups: $h$-adaptivity, $p$-adaptivity and $r$-adaptivity, depending on whether the element size (and shape), the approximation order or the vertex location (but not the topology) are controlled, respectively. In addition, the first two can be combined in what is known as $hp$-adaptivity [250, 39].

In many cases, the adaptivity module and the solver are called iteratively in a staggered fashion until a certain stopping criterion is met. In the case of $h$- and $p$-adaptivity, the stopping criterion is usually related to computational cost (e.g. CPU time or maximum number of degrees of freedom). On the contrary, in the case of $r$-adaptivity, the stopping criterion is usually dictated by the solver reaching the end of the simulation.

This staggering between flow solution and adaptivity has two main purposes. Firstly, it serves as a firewall between both so that the particular data structures

and requirements of one do not interfere with the inner workings of the other one. Secondly, it potentially saves computation in the case of $h$- or $p$-adaptivity by using as few degrees of freedom as possible in the early iterations.

This shortcut can be a source of problems when nonlinear systems of PDEs are simulated. In the case of the Reynolds-averaged Navier-Stokes equations, the root of the problem can be usually traced back to the extremely thin flow features in the boundary layer, that are misrepresented in a coarse mesh and end up producing spurious oscillations that ultimately prevent convergence. This behavior has prompted a wide body of research ranging from modified turbulence models [23, 158, 10] to advanced nonlinear solvers [155, 40], all of which aim to converge to a solution on a mesh without the adequate resolution.

In this work, this problem is tackled by means of an $r$-adaptive formulation that evolves the mesh "on the fly", and is inspired in the differential boundary layer solvers of Drela [67] and Allmaras [8]. In essence, the procedure turns the mesh generation into a set of equations, that are solved simultaneously with the flow equations. The former are designed so that the mesh tracks the viscous layers closely, thereby reducing the risk of divergence of the simulation and ensuring that, at convergence, the mesh is adapted to the boundary layer. An interesting feature of this procedure is that boundary layer profiles are very accurate and can drive a module that predicts the transition location together with the flow. The discussion on this topic is postponed to Chapter 5,

This chapter is organized as follows. First, the mesh adaptivity is thoroughly described starting from a high level discussion of the method. Next, the equations that govern the process are explained in detail. Then, a suite of 2-D results for laminar and turbulent flows is presented as a validation of the solver. Finally, a critical discussion of the method as well as some future developments are presented.

## 4.2 Formulation

### 4.2.1 Motivation

The proposed scheme is based on a high order discretization of the RANS equations using an $r$-adaptivity approach where the mesh generation and the fluid discretization are coupled in a strong way and solved simultaneously "on the fly". This is enabled by a series of high level decisions on matters like the structure of the mesh, the high order representation of it and the way that boundary layers are gauged, that are discussed next.

#### Measuring the boundary layer thickness

One of the advantages of adapting to the boundary layer versus other flow features like shock waves or wakes is the fact that the location of the boundary layer is known. In particular, boundary layers are present close to solid walls in most flows of interest in aerodynamics, provided the flow only separates moderately. This property effectively removes the location of the boundary layer as an unknown leaving its extension (or thickness) as the only variable of the problem. In general, the boundary layer thickness is not uniform and can range from virtually zero near the stagnation point to the order of the airfoil thickness close to the trailing edge. Thus, the first technical challenge to address is that of determining the boundary layer thickness.

Strictly speaking, the boundary layer ends where the viscous effects are negligible, or when a certain percentage of the external velocity is recovered [214]. However, this is of little help since it requires an exhaustive search in the direction normal to the wall.

When it comes to implementation, especially in the context of adaptivity, it is more convenient to consider other ways to measure the boundary layer thickness. Of particular interest here are those based on an algebraic relationship between integral boundary layer quantities. In this work, the following relationship proposed by

Drela [69] is used:

$$\delta_{\mathrm{BL}} = \theta_{\mathrm{k}} \left( 3.15 + \frac{1.72}{H_{\mathrm{k}} - 1} \right) + \delta^*_{\mathrm{k}}. \tag{4.1}$$

Here, $\delta_{\mathrm{BL}}$ is an estimate for the measure of the boundary layer and $\delta^*_{\mathrm{k}}$, $\theta_{\mathrm{k}}$ and $H_{\mathrm{k}}$ represent the displacement thickness, momentum thickness and shape parameter, respectively, which are defined as:

$$\delta^*_{\mathrm{k}} = \int_0^{y_{\mathrm{e}}} \left( 1 - \frac{u}{u_{\mathrm{e}}} \right) dy, \tag{4.2}$$

$$\theta_{\mathrm{k}} = \int_0^{y_{\mathrm{e}}} \left( 1 - \frac{u}{u_{\mathrm{e}}} \right) \frac{u}{u_{\mathrm{e}}} dy, \tag{4.3}$$

$$H_{\mathrm{k}} = \frac{\delta^*_{\mathrm{k}}}{\theta_{\mathrm{k}}}. \tag{4.4}$$

Here, $u$ denotes the velocity in the direction tangent to the wall, $u_{\mathrm{e}}$ denotes the velocity at the edge in the direction tangent to the wall, and $dy$ is just a formalism to denote integration in the direction normal to the wall. In a rigorous setting, the integrals described in Eq. 4.2 and 4.3 should be extended to infinity in the upper limit, however, give $u$ quickly asymptotes to the inviscid solution, it is customary to truncate them to a distance $y_{\mathrm{e}}$ of the order of the thickness.

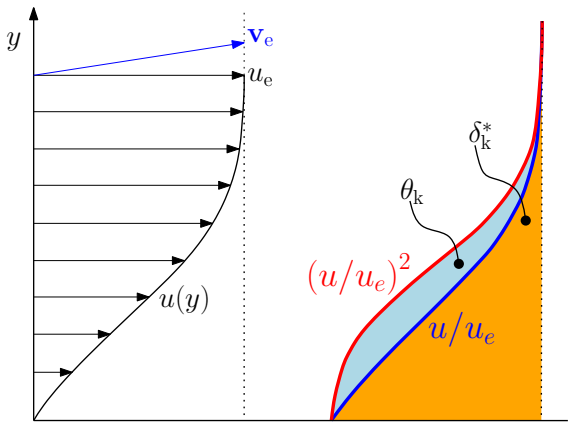



Figure 4-1: Graphical definition of the boundary layer integrals $\delta^*_{\mathrm{k}}$ and $\theta_{\mathrm{k}}$.

Other ways of measuring the boundary layer thickness in a RANS solver have been proposed in the literature. These range from the use of a stand-alone solver for the

boundary layer equations coupled through the pressure at the wall [131] to a variety of algorithms that search for the edge of the boundary layer by means of a diagnostic function [197, 229]. However, none of these have any advantage over the one used here when the mesh is adapted as part of the solution.

By using Eq. 4.1, the complexity of measuring the boundary layer thickness has been transferred to performing a series of integrals across the boundary layer, which requires: 1) an easy way to extract the boundary layer profiles, and, 2) an unambiguous definition of the state at the edge of the boundary layer. Both of these are satisfied by the mesh structure proposed in the next section. With these, the numerical evaluation of Eq. 4.1 is straightforward as explained in detail in Section 4.2.4.

### Mesh topology

The solution to the challenge of tracking the boundary layer using $r$-adaptivity lies on allocating structure to the region near solid walls, so that the two aforementioned requirements are met. More precisely, the proposed solver uses a hybrid mesh composed of quads and triangles that satisfies two design principles:

- that the region near the wall is meshed using stacks of quadrilaterals extruded from the surface of the geometry of interest, and,
- that the number of elements in each stack is fixed.

In this way, extracting the boundary layer profiles within each stack is a trivial task once the right data structures are put together. Similarly, the state at the edge of the boundary layer can be taken from the edge of the stack. From now on, the part of the mesh composed of the stacks adjacent to the wall will be referred to as the boundary layer mesh (or domain), while the remainder will be referred to as the external mesh (or domain).

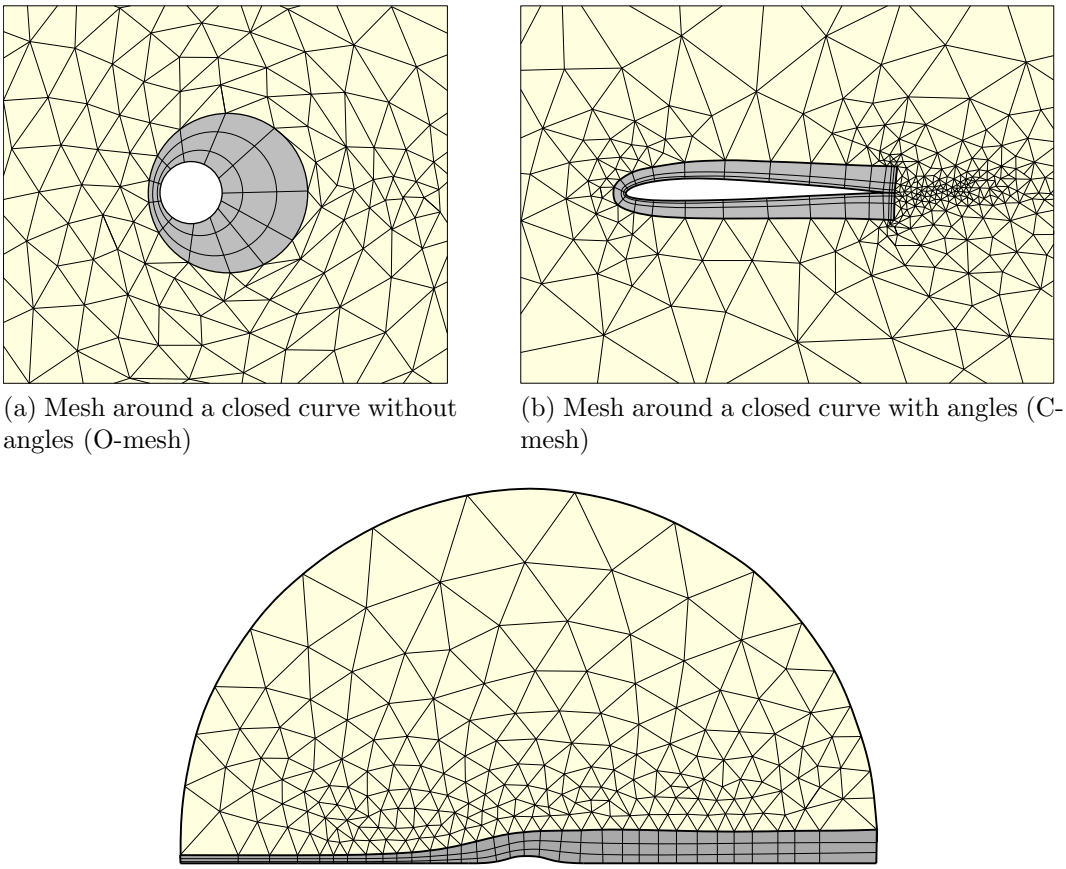
Notice that the two design principles only affect the boundary layer domain and do not impose any restriction on the topology of the external mesh other than it being conformal to the rest of the geometry. In all the results presented in this work, the

external mesh is unstructured and composed of triangles, which simplifies the task of generating it using standard mesh generation tools. In particular, the 2-D meshes used in all the results presented here are constructed in three steps. The first step is the generation of the boundary layer mesh by extruding it from a suitable surface mesh for the wall. Once this is done, the edge geometry is collected along with the rest of the features (e.g. far-field, symmetry plane, etc.) and used to call BAMG [94] or Distmesh [190], that return the external mesh. In the final step, the connectivities are computed and the degrees of freedom in the boundary layer mesh are reorganized to facilitate the extraction of the boundary layer profiles. This simplified process is only performed once and serves to fix the topology of the mesh. More sophisticated techniques could also be used [18, 5] provided that they abide to the design principles stated above.

For illustration purposes, a sketch of the possible meshes obtained using this process is shown in Figure 4-2. The final topology of the mesh depends on the geometry of the solid wall. For example, in the case of a closed curve without angles (e.g. a cylinder or a turbine blade) the resulting mesh resembles an O-mesh in the boundary layer. On the contrary, if the geometry has a trailing edge, the boundary layer mesh will look like a C-mesh. Finally, in the case where the geometry is not closed (e.g. flat plate), the boundary layer mesh will be close to a rectangular structured mesh.

### High order geometry

The use of a structured quadrilateral mesh on the boundary layer domain has advantages beyond the convenience of the associated data structure. In particular, it has been recently proved that the combination of high order methods and quadrilateral meshes in the vicinity of the wall delivers solutions of higher quality in terms of overall force prediction as well as pointwise stresses [71]. Furthermore, the use of a high order discretization makes the boundary layer profiles less sensitive to quantities like grid stretching in the case of turbulent flows [71] and produce the same level of accuracy with fewer degrees of freedom. This reason alone is enough to justify the use of a



(a) Mesh around a closed curve without angles (O-mesh)

(b) Mesh around a closed curve with angles (C-mesh)

(c) Mesh around an open curve (Rectangular mesh)

Figure 4-2: Sketch of the mesh topology for different kinds of geometries. For visualization purposes the boundary layer mesh and the external mesh are colored differently.

high order method on the meshes described above, especially when transition is also computed (see Chapter 5).

The use of a high order discretization in space requires the use of a mesh that represents the geometry with the same order or more [26, 217]. In this work, the geometry is represented using isoparametric elements by means of a set of high order interpolating polynomials in the reference space, along with the associated interpolation nodes (also referred to as high order nodes). The latter are the degrees of freedom that the method can re-locate to adapt the mesh to the boundary layer. The particular equations that govern such a process are described in detail in the coming sections.

### 4.2.2 Normal scaling equation

The first set of equations of the system is associated with the so-called normal scaling $\delta$, which is an approximation to the thickness of the boundary layer mesh. Ideally, $\delta$ should be a scalar field on the manifold defined by the wall geometry, that targets the value of $\delta_{\mathrm{BL}}$ in Eq. 4.1. For this, the surface PDE developed by Allmaras [8] in his studies of an adaptive differential boundary layer code is used. This reads:

$$\frac{\partial \delta}{\partial t} = \frac{k_\delta \delta_{\mathrm{BL}} - \delta}{\tau_\delta} + \mu_\delta \Delta_\Gamma \delta. \tag{4.5}$$

Here, $k_\delta > 1$ is a safety factor that ensures that the boundary layer mesh is thicker than the fluid boundary layer itself, $\tau_\delta$ is a constant that controls the response time of the system and $\mu_\delta$ is a constant that governs the smoothing of the solution along the surface. In this notation, $\Delta_\Gamma$ represents the surface Laplacian, also known as the Laplace-Beltrami operator. Notice that this surface PDE relates $\delta$ to the flow quantities through the definition of the normal scaling indicator $\delta_{\mathrm{BL}}$.

Furthermore, in the limit of steady state solutions and small diffusion, Eq. 4.5 is equivalent to $\delta = k_\delta \delta_{\mathrm{BL}}$, which ensures that the solution follows the indicator in Eq. 4.1, up to the safety factor $k_\delta > 1$.

#### Discretization

The discretization of $\delta$ is based on a standard finite element technique, adapted to the fact that Eq. 4.5 is a PDE on a surface, hence, $\delta$ is a variable that lives on a sub-manifold of the domain. The first step in the discretization is to define a mesh for such manifold. A simple solution for this is to use the hybrid mesh as a donor from which the boundary mesh is extracted. This procedure is graphically explained in Figure 4-3.

The resulting mesh, $\Gamma_h$, is a conformal, high order approximation to the solid wall manifold. In it, each element $K^\Gamma \in \Gamma_h$ is defined through an isoparametric mapping (or chart) from a reference domain in Euclidian space $\boldsymbol{\xi} \in \mathbb{R}^n$ to the physical space

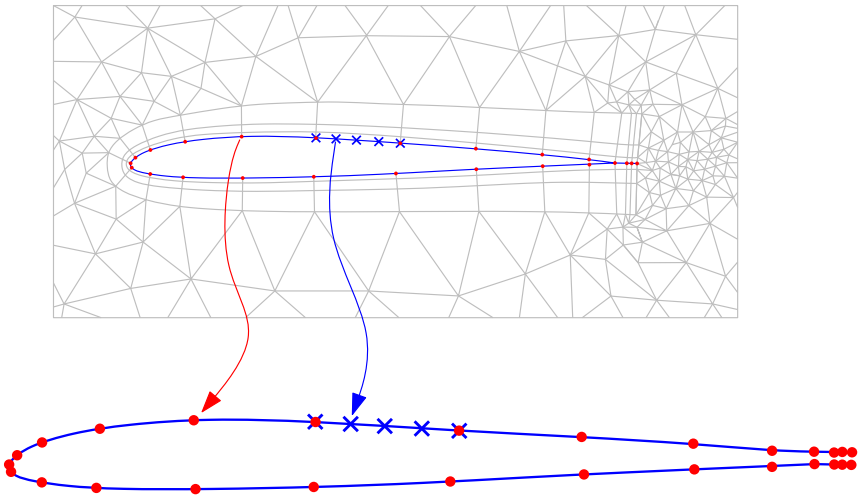


Figure 4-3: Extraction of the high order surface mesh from the volume mesh. The solid dots (•) denote the vertices of the mesh, while the crosses (×) denote the high order nodes that define the isoparametric mapping.

$\mathbf{x}(\boldsymbol{\xi}) \in \mathbb{R}^{n+1}$. Here, $n$ is the parametric dimension of the surface (e.g. $n = 1$ in 2-D flows). The union of all these mappings defines an atlas of the approximate manifold. Notice the approximate manifold is not differentiable at the boundaries between elements since continuity in the tangent is not enforced there. However, this does not affect the calculus inside each element, where all the tools from differential geometry are properly defined, such as the covariant and contravariant metric tensors:

$$g_{ij} = \frac{\partial \mathbf{x}}{\partial \xi_i} \cdot \frac{\partial \mathbf{x}}{\partial \xi_j}, \quad g^{ij} = (g_{ij})^{-1}, \tag{4.6}$$

and determinant

$$g = \det(g_{ij}), \tag{4.7}$$

as well as the covariant gradient of a scalar field:

$$\nabla_i f = \frac{\partial f}{\partial \boldsymbol{\xi}}. \tag{4.8}$$

The discretization of Eq. 4.5 on $\Gamma_h$ follows the Surface Finite Element Method

(or SFEM) by Dziuk [74] with minor modifications to accommodate the high order geometry. The starting point is the definition of the functional spaces and inner products. Let, $\mathcal{U}_h$ denote the trial space for the solution, consisting of piecewise linear functions on $\Gamma_h$:

$$\mathcal{U}_h = \{v \in (C^0(\Gamma_h)) : v|_{K^\Gamma} \in (\mathcal{P}^1(K^\Gamma)), \forall K^\Gamma \in \Gamma_h\}. \tag{4.9}$$

Also, let:

$$(u, v)_{K^\Gamma} = \int_{K^\Gamma} uv d\Omega, \quad \text{and} \quad \langle u, v\rangle_{\partial K^\Gamma} = \int_{\partial K^\Gamma} uv ds, \tag{4.10}$$

denote the inner product between two functions integrated in the interior of the element or the boundary, respectively.

To derive the weak formulation, the original surface PDE is integrated by parts against the test space $\mathcal{V}_h$. Assuming that the test and trial space coincide: $\mathcal{U}_h = \mathcal{V}_h$, the semi-discrete weak form for the evolution of $\delta_h$ in time reads: find $\delta_h \in \mathcal{U}_h$ such that:

$$\sum_e \left(\frac{\partial \delta_h}{\partial t}, v\right)_{K^\Gamma} - \sum_e \left(\frac{k_\delta \delta_{\mathrm{BL}} - \delta_h}{\tau_\delta}, v\right)_{K^\Gamma} + \sum_e \left(\mu_\delta g^{ij} \nabla_i \delta_h, \nabla_j v\right)_{K^\Gamma} - \\ - \sum_e \langle \mu_\delta g^{ij} \nabla_i \delta_h \mathbf{n_i}, v\rangle_{\partial K^\Gamma} = 0, \quad \forall v \in \mathcal{V}_h. \tag{4.11}$$

Here $\mathbf{n_i}$ represents the covariant components of the normal to the element boundary. Notice that Eq. 4.11 looks like a standard discretization of a parabolic reaction-diffusion equation, except for the term in the second line involving the integrals in the contours of the elements, which is due to the non-differentiability of the approximate manifold at the element borders as discussed by Cantwell *et al.* [35].

As written, the discretization imposes natural boundary conditions on the boundaries of the manifold, provided such boundaries exist. Otherwise, the problem is still well posed due to the reaction term.

To close the problem, the constants $k_\delta$, $\mu_\delta$ and $\tau_\delta$ need to be defined. The first

of them is set empirically to $k_\delta = 1.5$ for laminar flows and $k_\delta = 2$ for turbulent or transitional flows, independently of the discretization. On the contrary, $\mu_\delta$ and $\tau_\delta$ depend on the time step $\Delta t$ and the element size $\Delta x$ in the following fashion: $\tau_\delta = 2\Delta t$ and $\mu_\delta = 0.1\Delta x^2/\tau_\delta$. This particular choice is motivated by the stability limit of the equation, which is discussed at length in Section A.4.

### 4.2.3 Mesh deformation

The second set of nonlinear equations is provided by the mesh deformation algorithm, that depends only on the normal scaling $\delta_h$. In particular, the mesh deformation proceeds independently on the boundary layer mesh and on the external mesh as discussed next.

#### Geometry of the boundary layer domain

The geometry of the boundary layer mesh is defined analytically as a function of $\delta_h$, with the help of the following three geometric entities:

- the surface mesh $\Gamma_h$: that was introduced in the previous section (see Figure 4-3) and will be the base for the extrusion,
- the extrusion direction $\hat{\mathbf{n}}$: that is a continuous vector field on $\Gamma_h$ that approximates the normal to the surface, and,
- the stack distribution $\{h_i\}$: that sets the relative thickness of the different layers in the boundary layer mesh.

Of these, the last two require a formal introduction.

The purpose of $\hat{\mathbf{n}}$ is to set the direction in which the extrusion will happen. Formally, $\hat{\mathbf{n}}$ is a piecewise linear vector field on the surface triangulation, or, using the same notation as in the previous section: $\hat{\mathbf{n}} \in (\mathcal{U}_h)^{n+1}$. In this work, $\hat{\mathbf{n}}$ is pre-computed and fixed throughout the computation. The process is very simple and involves computing an average of the normals of the different elements that contain a given vertex

of $\Gamma_h$, normalized so that $\|\hat{\mathbf{n}}\| = 1$ at the vertices. With this information, the propagation of $\hat{\mathbf{n}}$ to the interior of the element is based on a linear interpolation in the parametric space. Notice that the main purpose of the normalization of $\hat{\mathbf{n}}$ at the vertices is to decouple the thickness (measured by $\delta_h$) from the extrusion direction as much as possible.

The role of the stack distribution $\{h_i\}$ is to assign a certain fraction of the whole extrusion to each layer of the boundary layer mesh, hence controlling the growth rate of the mesh away from the wall. Here, $h_i$ denotes the relative thickness of element $i = 1, 2, ..., n_{\text{norm}}$ in the stack, with $i = 1$ being the closest to the wall and $n_{\text{norm}}$ being the total number of them. The sum of all relative thicknesses is normalized so that $\sum_i h_i = 1$. The process to construct the stack is very simple. Starting from $i = 1$, generate the sequence $\{\alpha_h^{i-1}\}$ where $\alpha_h$ is the relative growth between adjacent stacks. With this, the stack distribution follows from the normalization condition:

$$\{h_i\} = \frac{1}{\sum_1^{n_{\text{norm}}} \alpha_h^{i-1}} \{\alpha_h^{i-1}\}. \tag{4.12}$$

The parameter $\alpha_h$ controls the level of packing of the mesh close to the wall. In this thesis, it is set to $\alpha_h = 1.4$ for laminar cases and $\alpha_h = 1.6$ for turbulent ones as recently proposed by Drosson *et al.* [71]. In particular, these authors showed that for high order solutions ($p \geq 2$), the convergence of the friction coefficient mainly depends on the distance of the first node off the wall rather than the stretching, hence the choice of a rather high value of $\alpha_h$ that ensures the viscous sublayer is properly represented with only a few elements across the boundary layer. Needless to say that this is not the only possible way to define $\{h_i\}$.

With all these at hand, the geometry of a given element $i$ of the stack is defined analytically by:

$$\mathbf{x}(\boldsymbol{\xi}, \eta) = \mathbf{x}(\boldsymbol{\xi})|_{K^\Gamma} + \left( \sum_{j=1}^{i-1} h_j + \eta h_i \right) \left( \delta_h(\boldsymbol{\xi})|_{K^\Gamma} \right) \hat{\mathbf{n}}(\boldsymbol{\xi})|_{K^\Gamma}. \tag{4.13}$$

The discrete high order representation of the geometry follows a collocation of the isoparametric mapping on it. This is as simple as sampling Eq. 4.13 on the high order nodes of the reference space $(\boldsymbol{\xi}, \eta)$, which yields the position of the high order nodes in the boundary layer domain, denoted by $\mathbf{x}^{\mathrm{BL}}$. For illustrative purposes, the procedure is depicted in Figure 4-4.

Notice that the total extrusion, given by the term $(\delta_h(\boldsymbol{\xi})|_{K^\Gamma})\,\hat{\mathbf{n}}(\boldsymbol{\xi})|_{K^\Gamma}$, is a polynomial of order 2 in $\boldsymbol{\xi}$. This implies that the collocation of the geometry is exact provided $\mathbf{x}(\boldsymbol{\xi})|_{K^\Gamma}$ is a polynomial of order 2 or more. This requirement is automatically satisfied for all the high order solutions presented in this work.

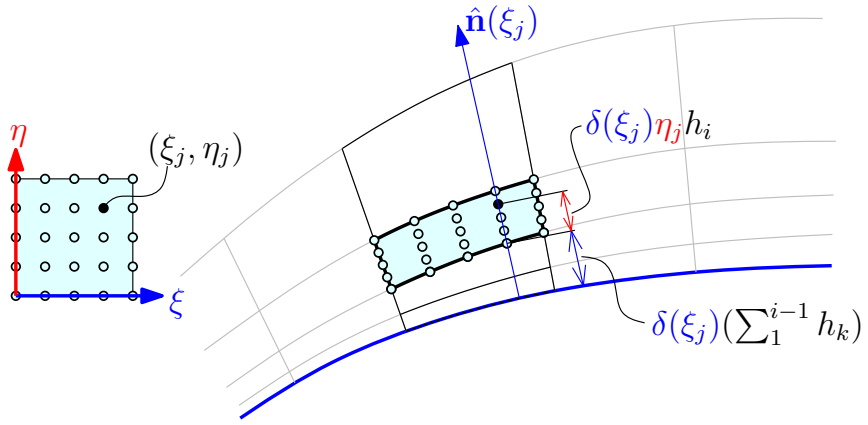



Figure 4-4: Reconstruction of the high order geometry of an element in the i-th layer of the stack.

### Deformation of the external mesh

The external mesh has to conform to the evolution of the boundary layer geometry during the solution process. For this, the mesh deformation algorithm by Roca *et al.* [204] and Gargallo-Peiro *et al.* [81] is used. This algorithm is based on an optimization procedure that minimizes the distortion of the mesh and penalizes inverted elements, hence reducing the risk of finding tangled elements at convergence.

The algorithm requires its own notation. Given an element of the external mesh $e$, let $\varphi_0 : \boldsymbol{\chi} \mapsto \mathbf{x}_0$ denote the map from a reference element to the initial (non-inverted) configuration of the element returned by the external mesh generation algorithm.

Similarly, let $\varphi : \boldsymbol{\chi} \mapsto \mathbf{x}$ denote the map from the reference element to the current configuration of the mesh. Both these maps are isoparametric and can be written explicitly in terms of the coordinates of the high order nodes of the mesh on the target space. Also, let $\phi : \mathbf{x}_0 \mapsto \mathbf{x}$ denote the map from the initial configuration of the element to the current one, which can be written as

$$\phi = \varphi \circ \varphi_0^{-1}. \tag{4.14}$$

For the sake of clarity, all these are sketched in Figure 4-5.

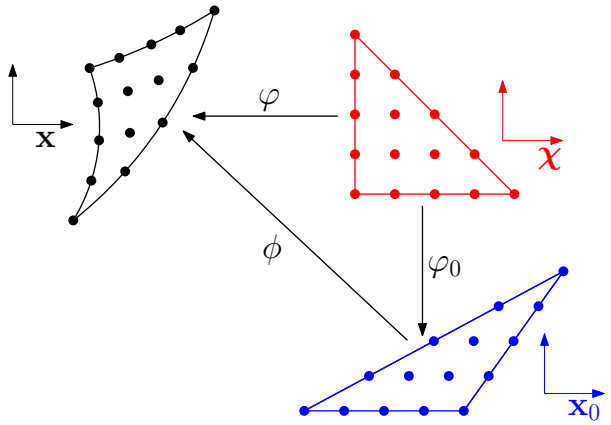



Figure 4-5: Diagram of the mappings that are relevant to the external mesh deformation.

The distortion at any point of the element is defined with the help of the gradient of the mapping $\mathbf{D}\phi$ in the following way [81]:

$$\zeta = \frac{\|\mathbf{D}\phi\|_{\mathrm{Fr}}^2}{2\sigma^*}, \tag{4.15}$$

$$\sigma^* = \frac{1}{2}\left(\det(\mathbf{D}\phi) + \sqrt{\det(\mathbf{D}\phi)^2 + 10^{-2}\det(\mathbf{D}\varphi_0)^2}\right). \tag{4.16}$$

Notice that $\zeta$ is a positive quantity such that $\zeta \in [1, \infty)$, with the lower bound corresponding to no distortion at all. This implies that for an ideal mesh $\zeta \approx 1$, or, equivalently, the distance $\zeta - 1$ is small. Rewriting this condition in least squares form and integrating over the element in the initial configuration yields an indicator for the

distortion of the element:

$$\zeta_e = \frac{\int_{e_0}(\zeta - 1)^2 dV_0}{\int_{e_0} dV_0}. \tag{4.17}$$

Notice that the value of $\zeta_e$ depends exclusively on the location of the high order nodes of the element at the current configuration $\mathbf{x}^{\text{ext}}_{\{\mathcal{J}(e)\}}$, which represent a subset $\{\mathcal{J}(e)\}, e \in \mathcal{T}_h$ of all the high order nodes of the external geometry, denoted by $\mathbf{x}^{\text{ext}}$.

The element-wise indicator can be reduced to a single scalar for the whole mesh by means of a summation over the elements. In that way, the problem of deforming the mesh is equivalent to that of minimizing the overall measure of distortion, given by:

$$\min_{\mathbf{x}^{\text{ext}}_j} \quad \sum_{e \in \mathcal{T}_h} \zeta_e(\mathbf{x}^{\text{ext}}_{\{\mathcal{J}(e)\}}), \tag{4.18}$$

$$\text{s.t.} \quad \mathbf{x}^{\text{ext}}_{\{\mathcal{J}_{\text{bou}}\}} = \mathbf{x}^{\text{edge}}(\delta_h). \tag{4.19}$$

Here, $\{\mathcal{J}_{\text{bou}}\}$ is the subset of nodes that lie on the interface between the eternal mesh and the boundary layer mesh and $\mathbf{x}^{\text{edge}}(\delta_h)$ is shorthand for the evaluation of Eq. 4.13 in those same boundaries. Notice the explicit dependency of the later on $\delta_h$, which is the driver for the whole mesh deformation. In essence, the constraint in Eq. 4.19 states that both meshes have to be conformal at the interface.

At optimality, the solution needs to satisfy the Karush-Kuhn-Tucker (or KKT) conditions. In this particular case, these are equivalent to the stationary point conditions since the constraints depend only on $\delta_h$. More precisely, the KKT conditions for the problem read:

$$\frac{\partial \sum_{e \in \mathcal{T}_h} \zeta_e}{\partial \mathbf{x}^{\text{ext}}_j} = 0, \quad \forall j \in \{\mathcal{J}\} \backslash \{\mathcal{J}_{\text{bou}}\}, \tag{4.20}$$

$$\mathbf{x}^{\text{ext}}_j = \mathbf{x}^{\text{edge}}(\delta_h), \quad \forall j \in \{\mathcal{J}_{\text{bou}}\}. \tag{4.21}$$

Where again, abusing notation, $\mathbf{x}^{\text{edge}}(\delta_h)$ is an explicit function of $\delta_h$ for each degree

of freedom on the boundary $j \in \{\mathcal{J}_{\text{bou}}\}$.

In the solver proposed here, the minimization statement is solved using a Sequential Quadratic Programming approach[174], which is equivalent to applying the Newton-Raphson iteration to the KKT conditions. The use of SQP as a minimization algorithm has several advantages, the most important being that it uses the same nonlinear iteration as the rest of the equations of the problem. This way, the simultaneous solution of the flow and the mesh can be recast as a Newton-Raphson iteration on a larger system of equations without the need to worry about details such as the order of the iteration between mesh and flow.

The external mesh could have been deformed using another method, for example, radial basis functions [187], or an elasticity analogy [5]. But, these were discarded in favor of the present approach for several reasons. In particular, the use of radial basis functions for the mesh deformation requires a wide stencil that can only be efficiently treated using an implicit-explicit scheme like the one proposed by Froehle and Persson [78]. This opens the door to all sorts of potential stability problems without adding much to the contributions of this thesis, hence it was never considered as an option. On the other hand, a nonlinear elasticity analogy [189] was implemented in the early stages of this work but was quickly dropped in favor of the current approach due to stability issues for large deformations.

As a final remark, it is important to highlight that the mesh generation technique just described is capable of producing a mesh that is parametrized only by $\delta_h$ and uses the same stencil as a Continuous Galerkin discretization, which could be considered the minimum coupling to represent an unstructured conformal high order mesh. In that sense, the method has the same exact algebraic cost as an elastic analogy, without the shortcomings of the physical model, like the reduced control over inversion of the elements.

### 4.2.4 Numerical evaluation of integral boundary layer quantities

The use of a hybrid mesh combined with $r$-adaptivity for the boundary layer simplifies the task of extracting the boundary layer profiles to that of evaluating the velocity profiles in the structured boundary layer mesh only. This way, the discretization of the integrals in Eq. 4.2 and 4.3 (required to evaluate $\delta_{\mathrm{BL}}$) can be simplified to a piecewise 1-D integral over the stack. For example, in the case of $\delta_{\mathrm{k}}^*$, this integral reads:

$$\delta_{\mathrm{k}}^* = \sum_{i=1}^{n_{\mathrm{norm}}} \int_{y_i}^{y_{i+1}} \left(1 - \frac{u}{u_{\mathrm{e}}}\right) dy, \tag{4.22}$$

where $y_i$ are just generic limits of integration for each element, and $dy$ denotes integration in the direction normal to the wall $\mathbf{n}$. In general, the latter does not need to coincide exactly with the extrusion direction $\hat{\mathbf{n}}$, which, on the other hand, is the natural direction for sampling the boundary layer profile since it coincides with the $\boldsymbol{\xi}$ = constant lines. Nevertheless, both can be reconciled if the extrusion direction is projected on the normal, as depicted in Figure 4-6.

This way, the discrete integral for $\delta_{\mathrm{k}}^*$ reads:

$$\delta_{\mathrm{k}}^* \approx \delta_h \times (\mathbf{n} \cdot \hat{\mathbf{n}}) \times \sum_{i=1}^{n_{\mathrm{norm}}} h_i \int_{\eta=0}^{\eta=1} \left(1 - \frac{u}{u_{\mathrm{e}}}\right) d\eta. \tag{4.23}$$

Similarly, the discrete version of $\theta_{\mathrm{k}}$ reads:

$$\theta_{\mathrm{k}} \approx \delta_h \times (\mathbf{n} \cdot \hat{\mathbf{n}}) \times \sum_{i=1}^{n_{\mathrm{norm}}} h_i \int_{\eta=0}^{\eta=1} \left(1 - \frac{u}{u_{\mathrm{e}}}\right) \frac{u}{u_{\mathrm{e}}} d\eta. \tag{4.24}$$

The only term missing a formal definition is the normalized tangential velocity profile $u/u_e$, which depends on the velocity vector $\mathbf{v}$ and the normal to the wall $\mathbf{n}$ in the following way:

$$\frac{u}{u_e} = \frac{\|\mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}\|}{\|\mathbf{v}_{\mathrm{e}} - (\mathbf{v}_{\mathrm{e}} \cdot \mathbf{n})\mathbf{n}\|}. \tag{4.25}$$

Here $\mathbf{v}_\mathrm{e}$ denotes the equivalent inviscid velocity profile which represents the extension of the inviscid flow into the boundary layer. For a general case with curvature at the wall, $\mathbf{v}_\mathrm{e} = \mathbf{v}_\mathrm{e}(y)$, which prevents the computation of $u/u_e$ directly. This issue could be circumvented if the velocity profile was substituted by a pseudo-velocity profile obtained from the integration of the vorticity away from the wall [240]. Unfortunately, this would introduce a double integral in the computation of $\delta^*_\mathrm{k}$ and $\theta_\mathrm{k}$ that would complicate the implementation. In this work, $\mathbf{v}_\mathrm{e}$ is assumed to be constant across the boundary layer and equal to the velocity at the edge of the boundary layer domain. This simplifies the implementation at the expense of some accuracy in the computation of the boundary layer properties. When the problem is discretized using an HDG approach, the approximation to $\mathbf{v}_\mathrm{e}$ is taken from the corresponding trace variable $\hat{\mathbf{u}}$.

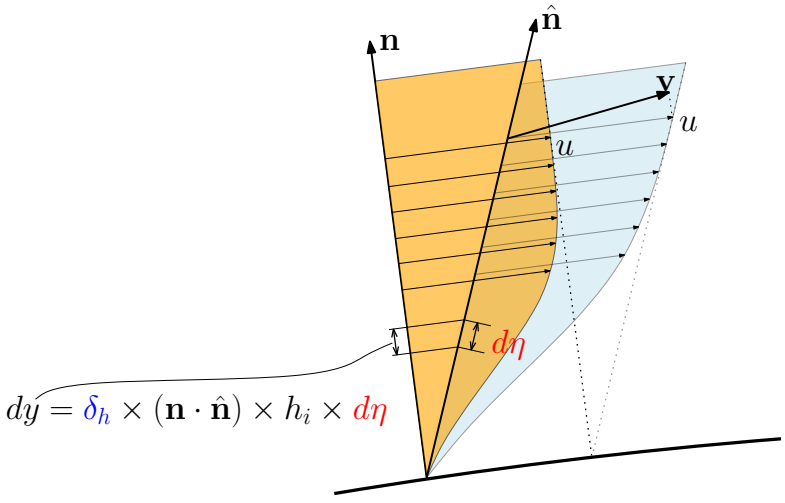



Figure 4-6: Sketch of the correction procedure to compute the boundary layer integrals. The distance to the wall is corrected using the inner product between $\mathbf{n}$ and $\hat{\mathbf{n}}$.

### 4.2.5 Discussion of the coupling

The adaptive solver is composed of three sets of discrete nonlinear equations:

- the discrete equation that governs the evolution of the boundary layer thickness (Eq. 4.11), discretized using a continuous Galerkin Surface FEM,
- the discrete equations that govern the mesh deformation for the external domain (Eq. 4.20 - 4.21), that are discretized using an SQP approach, and,
- the discrete equations for the fluid flow (see Chapter 2), that are discretized using an HDG method with ALE to account for the mesh deformation.

These three systems govern the evolution of three corresponding sets of variables:

- the normal scaling for the boundary layer $\delta_h$,
- the location of the high order nodes of the external mesh $\mathbf{x}^{\text{ext}}$, and,
- the high order approximation of the solution $\mathbf{u}_h$, its gradient $\mathbf{Q}_h$ and the traces $\hat{\mathbf{u}}_h$.

All of these together are enough to fully describe the flow solution as well as the mesh that supports it. The fact that said mesh is adapted to the solution on the fly comes at a price in that the variables are coupled throughout the system. Some of these couplings are one-way while others are more convoluted. For clarification purposes, a high level flow chart of all these relationships is depicted in Figure 4-7.

### 4.2.6 Solution procedure

All the results presented in this chapter represent simulations of steady state flows. To reach such state, a relaxation in time is performed using a BDF1 formula for the discretization of the different time dependent terms that appear in the system. The selection rule for $\Delta t$ follows the one from Chapter 3 (see pp. 90). The velocity of the mesh, that is required in the ALE mapping, is obtained through the same BDF1

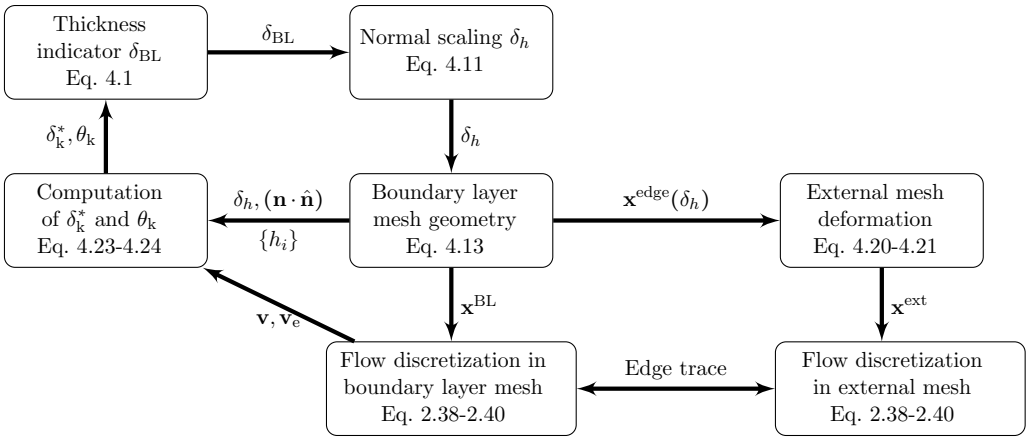



Figure 4-7: Flow chart of the adaptive solver showing the interdependence of the different modules (boxes) as well as the variables (arrows).

discretization. All this yields a system of nonlinear equations at each time step that is solved using a Newton-Raphson iteration.

Each step of Newton-Raphson requires the assembly of the residual of the equations as well as the Jacobian of the residual through the evaluation of a variety of integrals. In this solver, all these are approximated using Gaussian quadrature with enough points to integrate polynomials up to order $4p$, paying particular attention to ensure that these are uniform amongst the different discretizations, e.g. that the 1-D quadrature rules on the boundary of a triangle, on the boundary of a quad or on the surface FEM discretization are exactly the same. In the same fashion, the value of $\delta_{\text{BL}}$ at all the Gauss points on the surface is generated through an independent integration in the extrusion direction for every Gauss point independently. The resulting residual and the Jacobian are stored in a sparse format to minimize the memory footprint. No effort is devoted to optimize the linear algebra (static condensation, unknown reordering, etc.) in this early prototype of the solver.

The update of the solution involves inverting a linear system with the Jacobian on the left hand side, which is done using a sparse direct solver. The size of the update of the solution at each Newton-Raphson iteration is chosen using a rule that combines a backtracking line-search with a check to ensures that $\delta_h$ remains positive. The latter is very simple to implement thanks to the piecewise linear nature of $\delta_h$. The nonlinear

iteration ends when the norm of the update vector or the residual is below $10^{-9}$ in absolute value.

The solution process is started from uniform freestream conditions for the flow and a constant value for the normal scaling: $\delta_h/L = 3.1619 k_\delta \frac{1}{\sqrt{Re_L}} \sqrt{\frac{\Delta t u_\infty}{L}}$ (see Appendix A.4). The latter depends on $\Delta t$ at the first time step. Common values for it are $\Delta t u_\infty/L = 10^{-3}$ for laminar flows and $\Delta t u_\infty/L = 10^{-5}$ for turbulent flows. This particular choice of initialization for the flow makes $\delta^*_{\mathrm{k}}$ and $\theta_{\mathrm{k}}$ zero at the first iteration, which causes an indefinite value of $\delta_{\mathrm{BL}}$. For that reason, during the first time step, the mesh is frozen by forcing $\frac{\partial \delta_h}{\partial t} = 0$.

## 4.3 Results

This section contains a collection of 2-D results computed using the $r$-adaptivity described above. The goal here is to verify and validate the solver using a variety of laminar and turbulent cases, with and without shock waves.

### 4.3.1 Laminar boundary layer over a flat plate

This section starts with the simulation of the laminar flow over a flat plate at zero degrees angle of attack, $M_\infty = 0.1$ and $Re_L = 10^5$. This test case will serve to verify the correct implementation of the $r$-adaptivity as well as to partially validate the solver. The geometry used in this simulation is described in Figure 4-8 together with the boundary conditions. Notice how the leading edge of the flat plate is explicitly embedded on the lower boundary of the domain. This is achieved by a simple change of boundary condition past a certain point along the boundary. This way of setting up the leading edge introduces a square root type singularity at that location that is alleviated by refining the mesh around it (see Figure 4-10).

In this and other cases involving flat-plate-like geometries, there exists a portion of the boundary layer domain that lies over an inviscid wall boundary condition. In those instances, the boundary layer thickness indicator yields $\delta_{\mathrm{BL}} = 0$ which generates

an infeasible mesh. The solution for this is to override the value of $\delta_{\mathrm{BL}} = 0$ with a small constant as indicated in Figure 4-8.

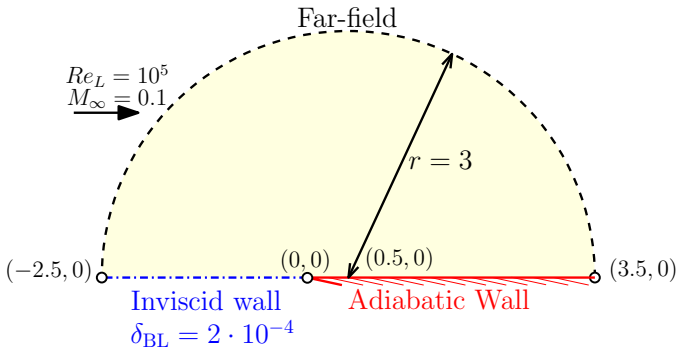


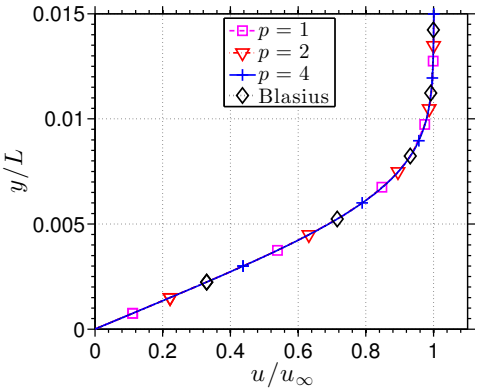

Figure 4-8: Sketch of the laminar flat plate geometry and boundary conditions.

For comparison purposes, the simulation is performed on a sequence of polynomial orders and meshes such that the resolution is approximately constant. For this particular case, this is achieved by uniformly refining the mesh and dividing the approximation order by two at the same time. Notice that this refinement is merely topological, since the $r$-adaptivity is re-run at each time to adapt to the boundary layer. The coarsest case is computed using polynomials of order $p = 4$ with a total of 10 elements across the boundary layer.

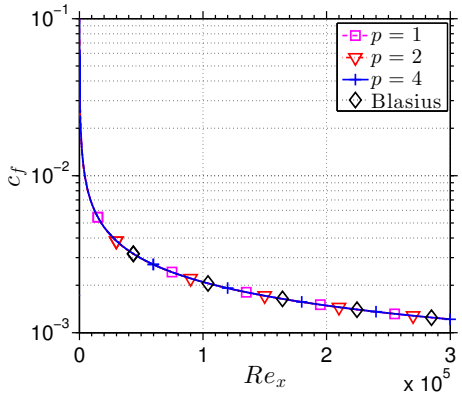
A summary of the results obtained on this sequence of meshes together with the analytical solution by Blasius is plotted in Figure 4-9. The numerical results for the flow field (i.e. velocity profiles or $c_f$) lie on top of the analytical curves for any approximation order. In addition, other derived quantities like the shape parameter $H_{\mathrm{k}}$ also show a very good agreement with the theoretical result.

In addition to this, the solver returns a mesh which is adapted to the boundary layer along the wall. This is plotted in Figure 4-10 for the case of the coarsest mesh. Notice how the normal scaling $\delta$ approximately follows a square root law as is expected from Blasius' theory, and asymptotes to the prescribed value of $\delta_{\mathrm{BL}}$ to the left of the stagnation point.

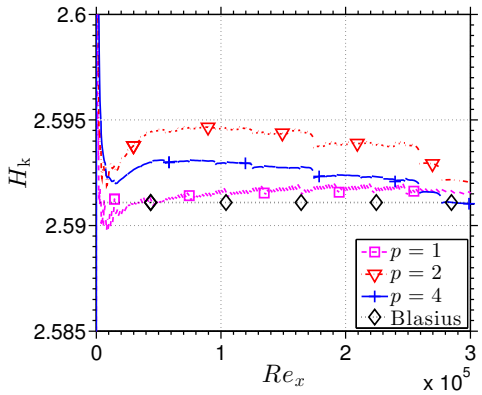
This simple test case serves as the starting point of the validation of the method, that is extended to more complex cases in what follows.




(a) Velocity profiles at $Re = 5 \cdot 10^4$

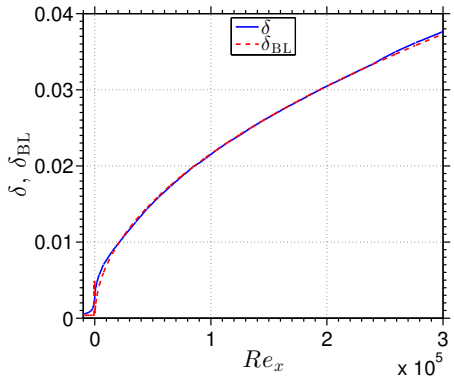



(b) Friction coefficient

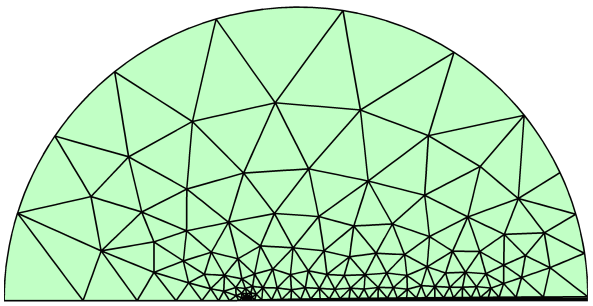



(c) Shape parameter

Figure 4-9: Comparison of the solutions obtained on a sequence of meshes with equivalent resolution against the analytical solution by Blasius. By properly adapting the boundary layer, the results become insensitive to the polynomial order.




(a) Normal scaling and thickness indicator



(b) Mesh

Figure 4-10: Mesh and normal scaling at convergence for the case of the laminar flat plate. Notice how the normal scaling follows the thickness indicator and can avoid the leading edge singularity thanks to the diffusive terms in the governing PDE.

### 4.3.2 Laminar separation bubble over a smooth bump

The goal of the next test case is to assess the behavior of the $r$-adaptivity when the boundary layer is subject to a generic pressure gradient, with particular emphasis on configurations where the flow separates. For this the laminar flow at $M_\infty = 0.3$ and $Re_L = 10^4$ over a smooth sine-squared bump of height $h/L = 0.06$ is simulated. A detailed description of the geometry and boundary conditions is found in Figure 4-11.

The focus here will be on the behavior at separation (since this might be troublesome according to Appendix A) as well as the assessment of the importance of the order of approximation in the quality of the solution when the resolution is limited. In all the runs performed here, the flow is assumed to remain laminar. While this might not be physically correct, it does not affect the conclusions of this study.

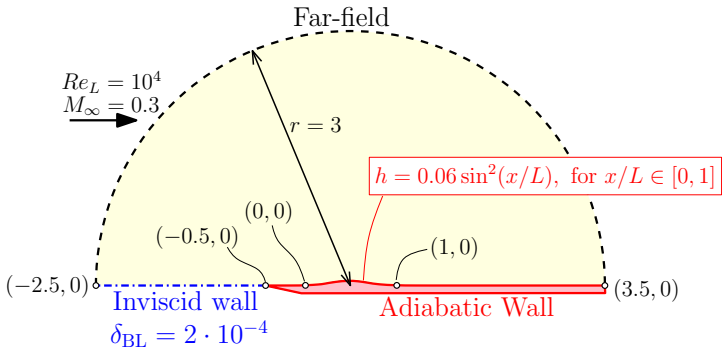



Figure 4-11: Sketch of the laminar bump geometry and boundary conditions.

As in the previous test case, the study is based on a sequence of mesh/$p$ pairs that are constructed so that the resolution is approximately the same in all the runs. In this case, however, uniform refinement will not do the trick, due to geometrical incompatibilities for $p = 1$. Instead, the following sequence is used: $(p, n_{\text{norm}}) = \{(2,10),(3,7),(4,5)\}$, which guarantees approximately the same resolution across the boundary layer mesh. To match the resolution in the external mesh, the element size function passed along to the external mesh generator is modified accordingly depending on $p$. The results obtained on this sequence of meshes is then compared to a high order solution on the finest mesh, that can be considered grid converged. Notice how the coarsest mesh only uses 5 elements of order $p = 4$ across the boundary layer. This

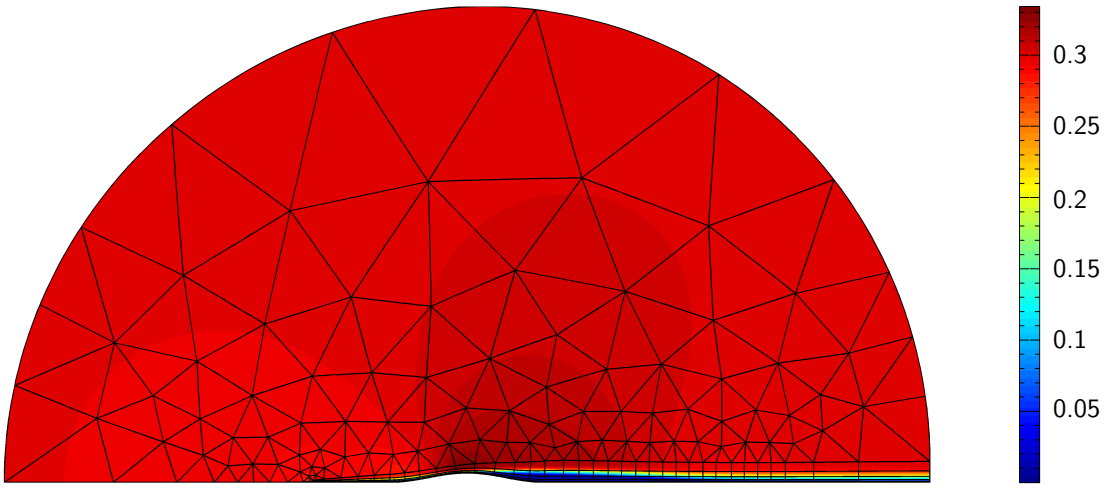
represents half the resolution used for the case of the laminar flat plate, and will serve to highlight the benefits of using a high order discretization.

The results from the run on the coarsest mesh are summarized in Figure 4-12. In particular, a general view of the mesh overlapped with the Mach number is contained in Figure 4-12a. Notice how the structured mesh is growing along the wall from the leading edge onwards. A zoom of this region starting at the beginning of the bump (Figs. 4-12b and 4-12c) reveals that the flow separates past the top of the bump and reattaches a certain distance behind it. This process is properly captured by the normal scaling that grows accordingly and maintains the shear layer within the boundary layer domain.

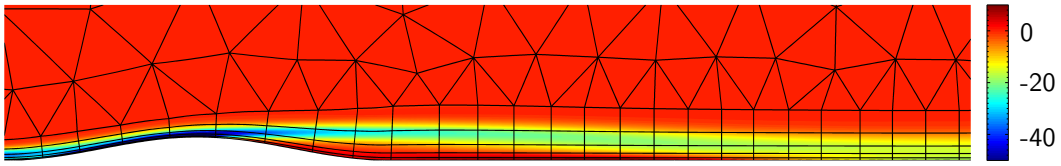
Similar results are obtained using the proposed mesh sequence with negligible differences between them. In particular, the pressure and friction coefficient coincide with one another as well as with the high order solution on the fine mesh, according to the plots in Figure 4-13. These results indicate that a variety of laminar boundary layer profiles can be accurately represented using a limited number of degrees of freedom (here, around 20), provided they are properly placed. This compares favorably with the 70-80 degrees of freedom required by standard finite volume solvers to get accurate boundary layer profiles (see Radespiel [197]). This property alone is a strong point in favor of the use of the proposed adaptivity.

Another quantity of interest is the evolution of the integral boundary layer properties along the separation bubble, since these are key in the transition prediction module that will be introduced in the next chapter. For this case, the evolution of the shape parameter $H_{\mathrm{k}}$ along the wall is depicted in Figure 4-14. These plots show a clear spread of $H_{\mathrm{k}}$ in the laminar separation bubble, that grows linearly from the separation point to peak at around 10%. This discrepancy can have a serious impact on the computed growth rates, that translates directly into errors in the transition location. Under this criterion, the departure from the actual value (assume to the solution on the finest mesh) is reduced if the polynomial order is increased.

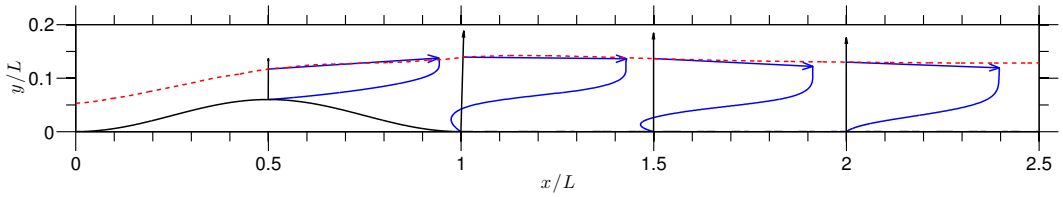
In summary, the results presented above have served three purposes. First and




(a) Mach number field and mesh.




(b) Vorticity field in the vicinity of the bump.

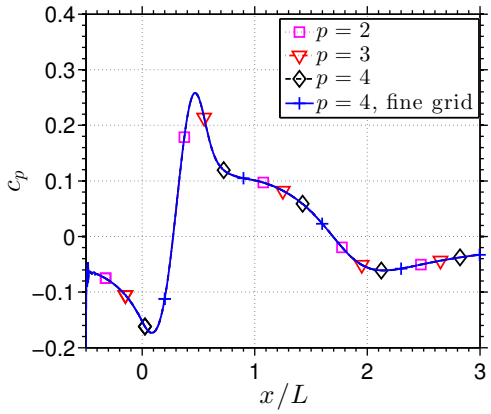



(c) Hodograph of the velocity at different stations along the wall. Dashed line represents the edge of the boundary layer domain.

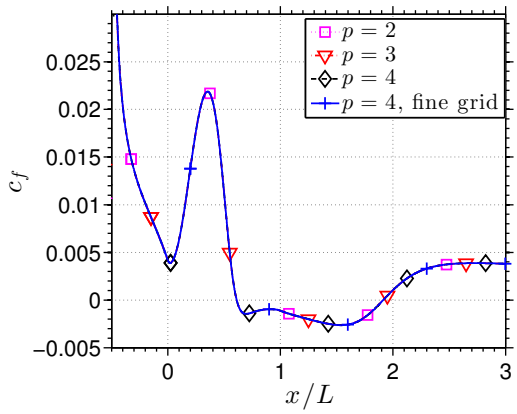
Figure 4-12: Flow field and mesh computed using a mesh with $n_{\text{norm}} = 5$ and polynomials of order $p = 4$. The results indicate that the $r$-adaptivity can also adapt to laminar separation bubbles.

foremost, they prove that the proposed $r$-adaptivity also works for separated flows when the parameters of the model are properly selected, as discussed in Appendix A. Secondly, the results confirm that the generation of the mesh on the fly can produce very accurate results with a limited number of degrees of freedom, independently of the details of the boundary layer like the profile shape or its thickness. Finally, the results indicate that the convergence of the integral boundary layer quantities might favor

the use of a high order discretization, even in cases where the flow field is essentially grid converged.

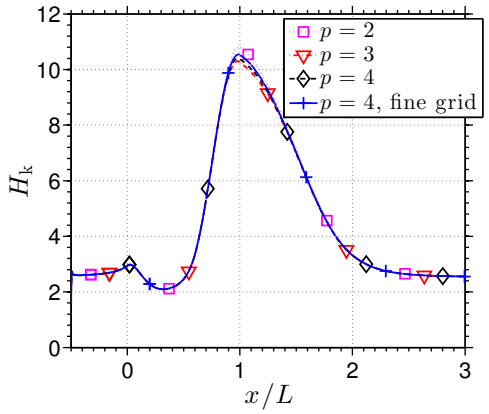



(a) Pressure coefficient

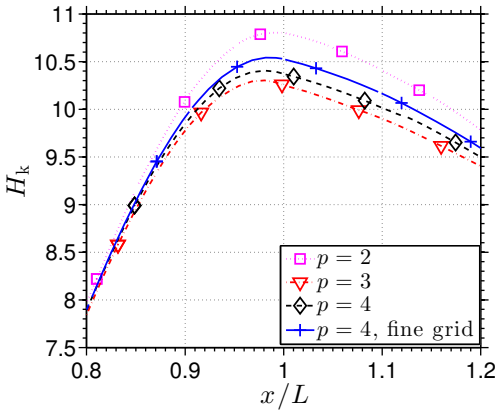



(b) Friction coefficient

Figure 4-13: Comparison of pressure and friction coefficients computed on a sequence of meshes with different order of approximation but the same overall resolution.




(a) Evolution of $H_k$ along the bump




(b) Detail of $H_k$ at the peak of separation

Figure 4-14: Evolution of the shape parameter $H_k$ with different approximation orders. The solutions coincide when the flow is attached, but show discrepancies in the separation bubble.

### 4.3.3 Turbulent flow over a flat plate

The extension of the proposed method to turbulent flows is studied next, starting with the simple case of a flat plate at zero degrees angle of attack, modeled using the RANS equations with the modified version of the Spalart-Allmaras model introduced in Chapter 2. For all the runs presented next, $M_\infty = 0.2$ and $Re_L = 5 \cdot 10^6$. The geometry used here is very similar to the one used for the laminar flat plate, with minor modifications. A sketch of the problem setup is found in Figure 4-15. The goal of this test case is to validate the solver against experimental data (hence the choice of Mach number and Reynolds number) and to assess the effect of using a high order approximation.

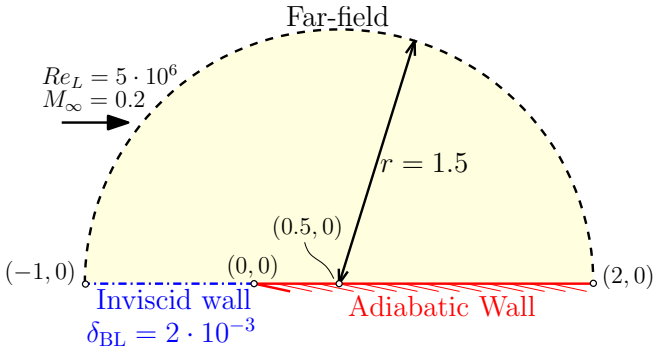



Figure 4-15: Sketch of the turbulent flat plate geometry and boundary conditions.

The results presented next are based on runs on a sequence of meshes constructed using the same procedure as in the laminar bump case. The main difference with the latter is in the definition of the $(p, n_{\text{norm}})$ sequence, that here is given by $(p, n_{\text{norm}}) = \{(2, 20), (3, 13), (4, 10)\}$. This combination represents twice as much resolution, and responds to the need to capture the complexity of the turbulent velocity profiles, especially close to the wall.

The skin friction coefficient obtained from these runs is plotted in Figure 4-16 together with the experimental results by Wieghart and Tillman [255], as well as the correlation by Schultz-Grunow [216]. For comparison purposes, the grid converged results obtained with the CFL3D solver that are available for download at NASA's Turbulence Modeling Resource database [208] are also included there. The agree-

ment amongst the numerical solutions is excellent, and very close to the experimental data, except around the leading edge. This seems to be an effect of the turbulence model rather than the numerics as other results on the Turbulence Modeling Resource database indicate.

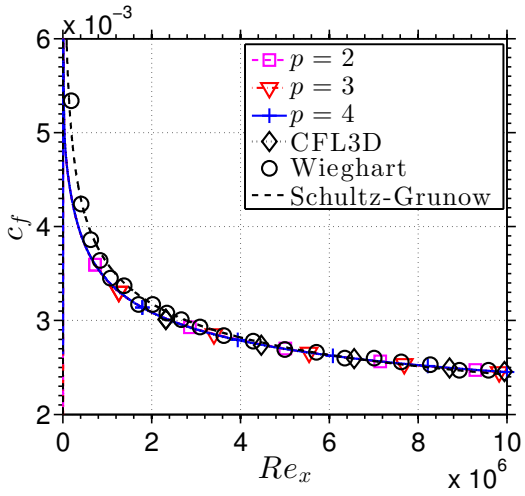
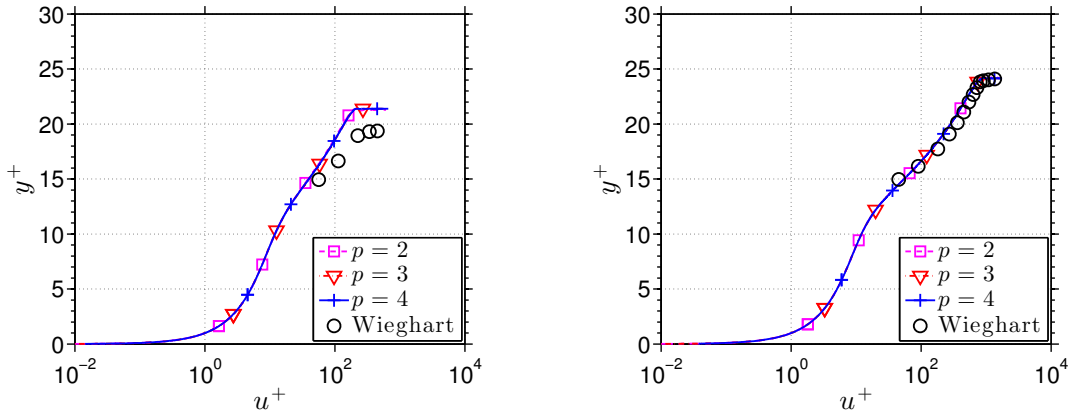


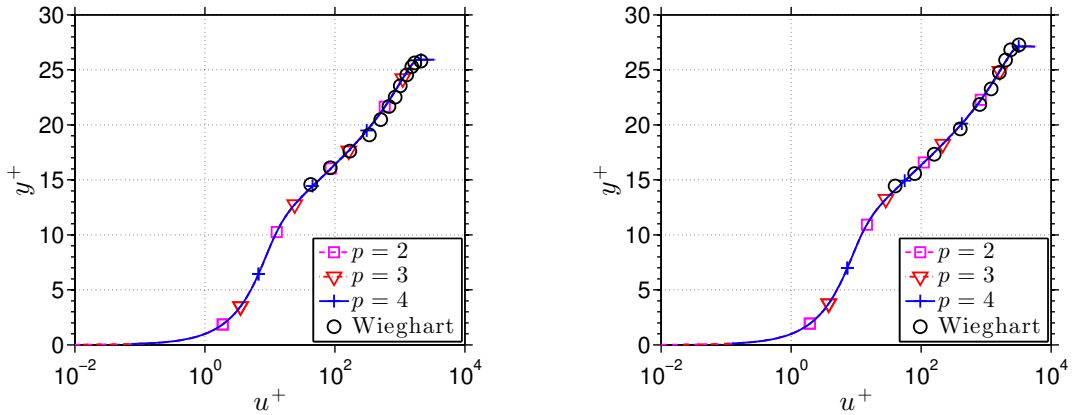

Figure 4-16: Friction coefficient along the flat plate compared to experimental data, empirical correlations, as well as grid converged results for the CFL3D solver.

To continue, Figure 4-17 contains a summary of the computed velocity profiles at different stations along the flat plate, compared with the experimental results by Wieghart [2]. The results show a very strong agreement between the different runs and also with the experimental data, except for the first station at $Re_x = 1.9071 \cdot 10^5$ (Figure 4-17a). The discrepancies found there are also present in other validation studies (e.g. the NPARC alliance database [219]) and can be partly attributed to the errors in $c_f$ around the leading edge mentioned above.

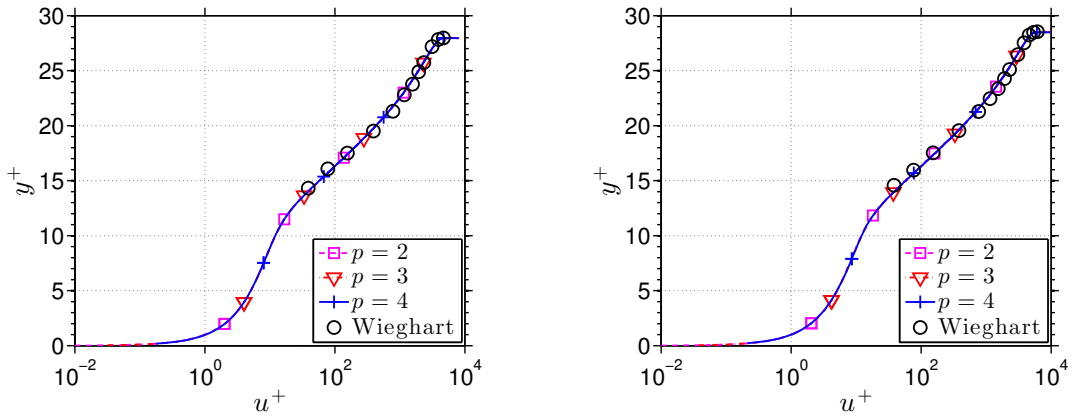
One of the advantages of adapting to the boundary layer thickness is the possibility to control the location of the first degree of freedom off the wall. This quantity of interest, denoted by $y_1^+$ in wall units, dictates the resolution in the viscous sublayer of the turbulent boundary layer and hence the accuracy of the computed friction coefficient. In the proposed $r$-adaptive scheme, the value of $y_1^+$ is partially controlled by $\delta$, hence, it is interesting to compare them side-by-side. This is done in Figure 4-18. The comparison reveals that adapting to the boundary layer thickness is enough




(a) $Re_x = 1.9071 \cdot 10^5$ (b) $Re_x = 1.0643 \cdot 10^6$




(c) $Re_x = 2.7034 \cdot 10^6$ (d) $Re_x = 4.9981 \cdot 10^6$

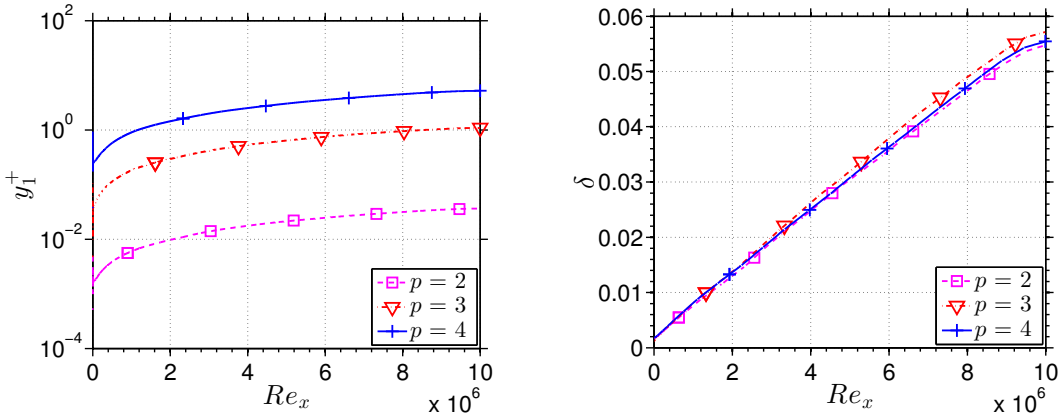



(e) $Re_x = 7.6206 \cdot 10^6$ (f) $Re_x = 1.0274 \cdot 10^7$

Figure 4-17: Horizontal velocity profiles measured in wall units ($y^+$ vs. $u^+$) at different stations along the flat plate. The agreement with experiments is excellent except for the case closest to the leading edge.

to keep $y_1^+$ under control, at least for the case of a zero pressure gradient flow.

All in all, this test case has served to verify the implementation of the solver for turbulent flows as well as to prove that adapting to the boundary layer thickness indirectly controls the resolution close to the wall. However, the simplicity of the test case makes all these conclusions preliminary, pending the application to a more general case. That is the goal of the next section.




(a) $y_1^+$ (b) Normal scaling $\delta$

Figure 4-18: Side-by-side comparison of $y_1^+$ and $\delta$ along the flat plate. The results indicate that the stretched grid in the normal direction as well as the adaptivity in the boundary layer thickness are enough to control the growth of $y_1^+$.

### 4.3.4 Turbulent flow around a NACA 0012 Airfoil

The study of the applicability of the solver to turbulent flows is expanded by looking at the flow around an airfoil. In particular, this test case deals with the flow around a NACA 0012 airfoil at $M_\infty = 0.15$ and $Re_c = 6 \cdot 10^6$. This particular configuration was chosen due to the availability of experimental and numerical data [208] for a variety of angles of attack in the range $\alpha \in [0°, 15°]$. The geometry of the problem as well as the boundary conditions are described in Figure 4-19.

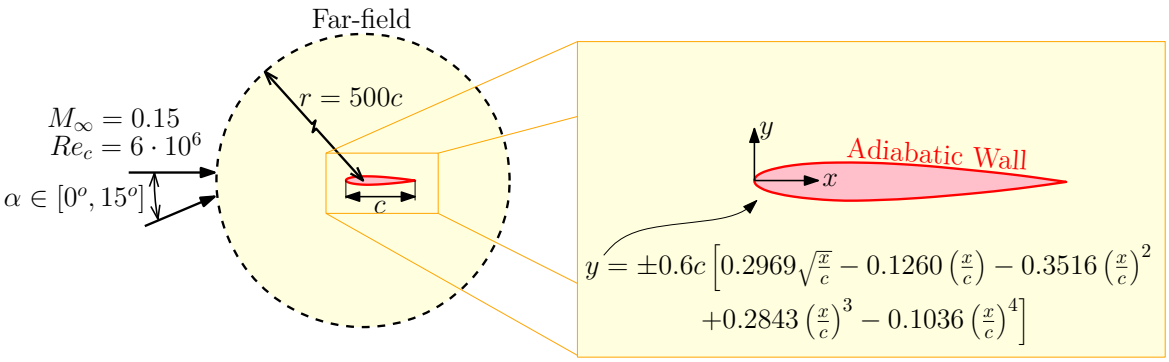



Figure 4-19: Sketch of the geometry and boundary conditions for the turbulent flow around a NACA 0012 airfoil.

This study compares runs at different angles of attack using the same polynomial order $p = 4$ and the same topological mesh composed of $n_{\text{surf}} \times n_{\text{norm}} = 49 \times 10$ elements in the boundary layer domain. An example of the flow field and mesh returned by the solver for the case of $\alpha = 15°$ is depicted in Figure 4-20. The validation proceeds by comparing the pressure and friction coefficients at three angles of attack ($\alpha = \{0°, 10°, 15°\}$) to experimental data (available only for $c_p$) as well as grid converged results computed with CFL3D and Xfoil. The results, that are summarized on Figure 4-21, show an excellent agreement between the numerical solutions and the experiments.

A similar behavior is found in terms of integrated forces like lift or drag, as depicted in Figure 4-22. Here, the $r$-adaptive solver is run at intervals of 1° ($\alpha = 0°, 1°, 2°, ...$)

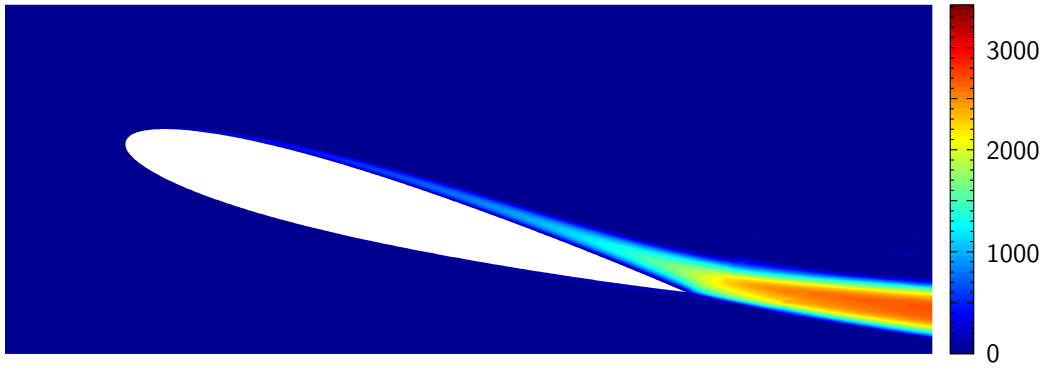
and compared against experimental data by Ladson, the results from CFL3D for $\alpha = \{0°, 10°, 15°\}$, and a polar computed with Xfoil.

In all the aforementioned, the results labeled as CFL3D were downloaded directly from the turbulence modeling resource webpage [208]. According to this source, these simulations were run using a structured C-mesh of $897 \times 257$ nodes ($\approx 230,000$ vertices). In comparison, the mesh used for the $r$-adaptive solver was composed of 38,500 high order nodes. This represents six times fewer degrees of freedom in the problem, before taking into account the static condensation of HDG that further reduces the number of coupled variables by a factor of around 2. These savings alone are enough to motivate the use of a high order approximation combined with $r$-adaptivity.

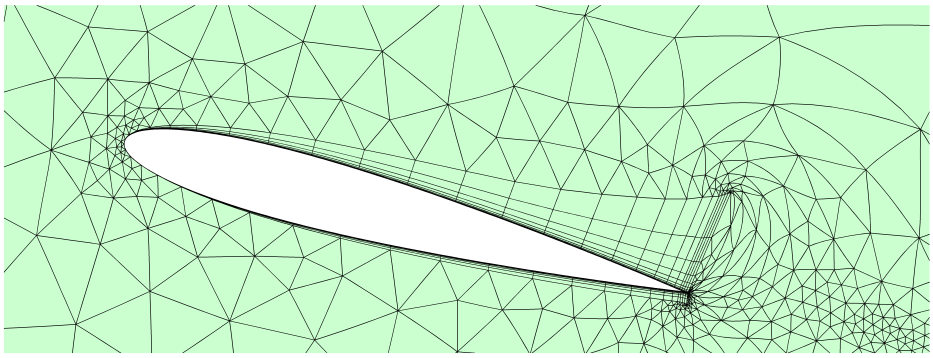
The effect of $r$-adaptivity shows clearly on the meshes obtained as part of the solution (see Figure 4-20c), but can be quantified by plotting $\delta$ and $y_1^+$ side-by-side. This is done in Figure 4-23 for the three reference values of $\alpha$, namely $\alpha = \{0°, 10°, 15°\}$. The most important thing to notice is how the $r$-adaptive strategy can produce meshes in which the thickness of the boundary layer domain grows up to three orders of magnitude from the stagnation point (where it is minimal) to the trailing edge. The adaptivity happens automatically as part of the solution process and is oblivious to the existence or not of a stagnation point in the flow. Furthermore, this aggressive growth of the normal scaling $\delta$ has a clear effect on the distribution of $y_1^+$, that, as in the case of the turbulent flat plate, remains of $\mathcal{O}(1)$ along the surface.

In summary, these results serve to validate the solver for low Mach number flows and also highlight the capabilities of the proposed scheme. In particular, the method presented in this thesis can produce meshes that are adapted to the different features of the flow (e.g. stagnation points, separation, etc.) in a way that increases the accuracy of the stresses at the wall. Furthermore, the mesh generation is totally independent of the parameters of the problem such as Reynolds number, Mach number or angle of attack, which allows reusability of the mesh topology.


(a) Mach number field

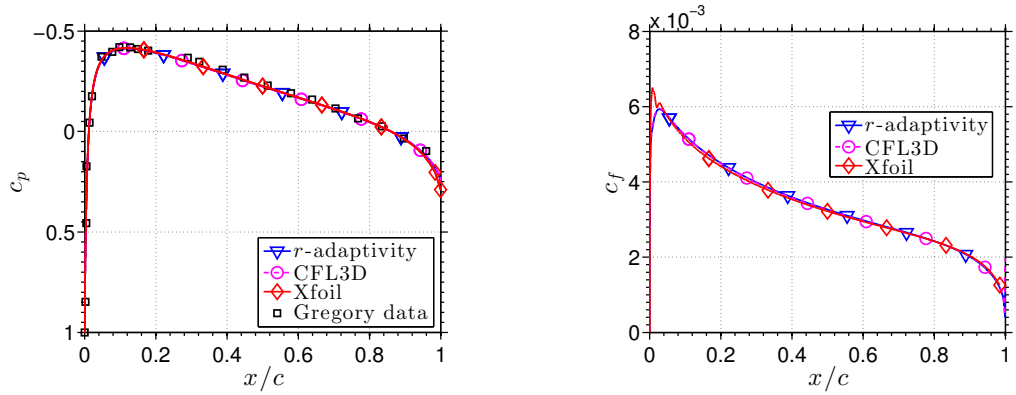



(b) Eddy viscosity $\mu_t/\mu_\infty$



(c) Mesh

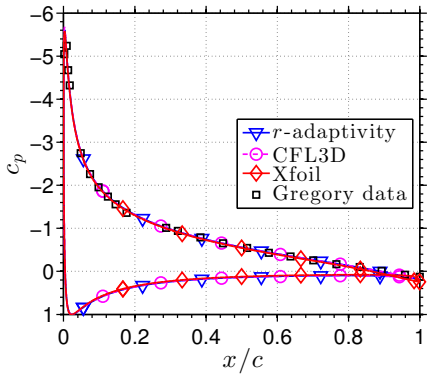
Figure 4-20: Flow field and mesh obtained for the case of a NACA 0012 at $M_\infty = 0.15$, $Re_c = 6 \cdot 10^6$ and $\alpha = 15°$. Notice the variation of the thickness of the mesh from the leading edge to the trailing edge.



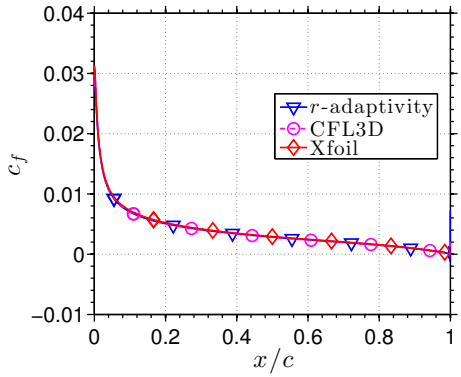

(a) $c_p$ at $\alpha = 0°$



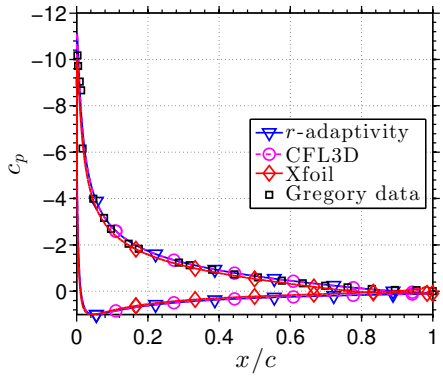

(b) $c_f$ at $\alpha = 0°$



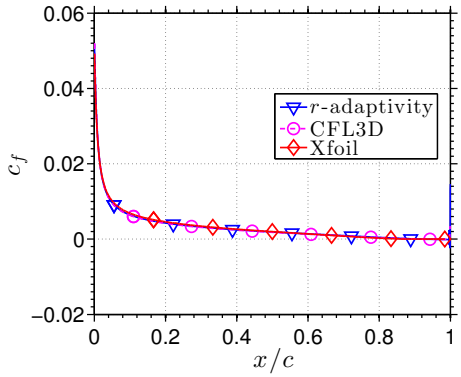

(c) $c_p$ at $\alpha = 10°$




(d) $c_f$ at $\alpha = 10°$

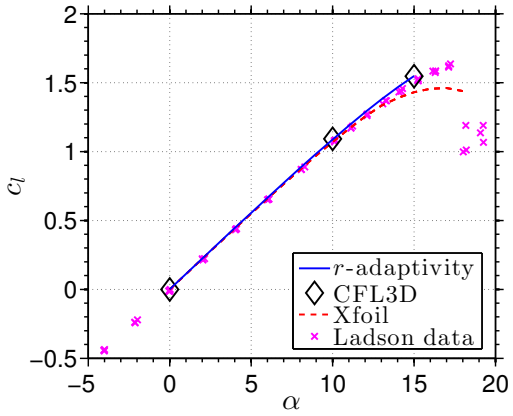



(e) $c_p$ at $\alpha = 15°$

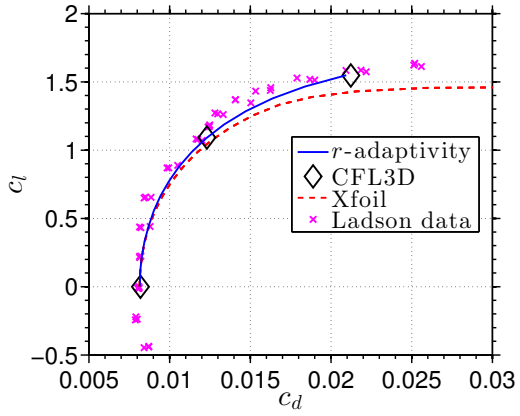



(f) $c_f$ at $\alpha = 15°$

Figure 4-21: Comparison of pressure and friction coefficients computed using the $r$-adaptive solver versus experimental data, CFL3D and Xfoil.

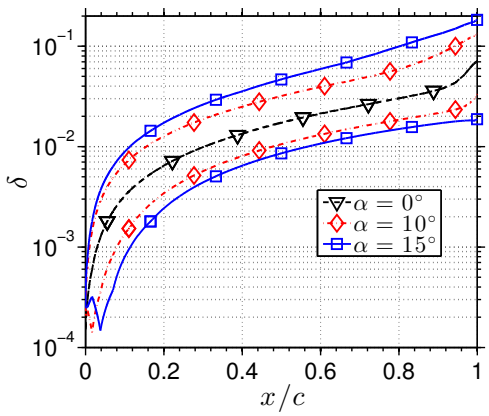



(a) Lift versus angle of attack

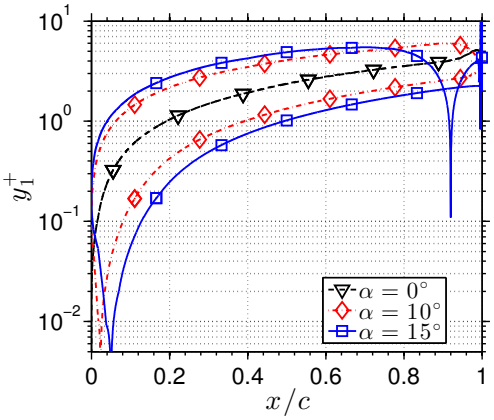



(b) Lift versus drag

Figure 4-22: Comparsion of the computed values of lift and drag versus experimental data, CFL3D and Xfoil.




(a) Normal scaling $\delta$




(b) Distance of first node off the wall

Figure 4-23: Chord-wise evolution of the normal scaling $\delta$ and the distance to the first degree of freedom off the wall ($y_1^+$) for three different angles of attack.

### 4.3.5 Viscous high-speed flow over a cylinder

The previous test cases have served to showcase the benefits of $r$-adaptivity for low Mach number flows, both in the laminar and turbulent regime. In what comes next, this study is extended to laminar compressible flows with the help of the shock capturing model proposed in Chapter 3.

The first test case in this series deals with the simulation of the flow around the bow of a cylinder in a supersonic stream at $M_\infty = 5$ with a Reynolds number of $Re_R = 4 \cdot 10^4$. In this configuration, the flow exhibits a detached shock wave and a very thin boundary layer. The details of the geometry and the boundary conditions are contained in Figure 4-28.

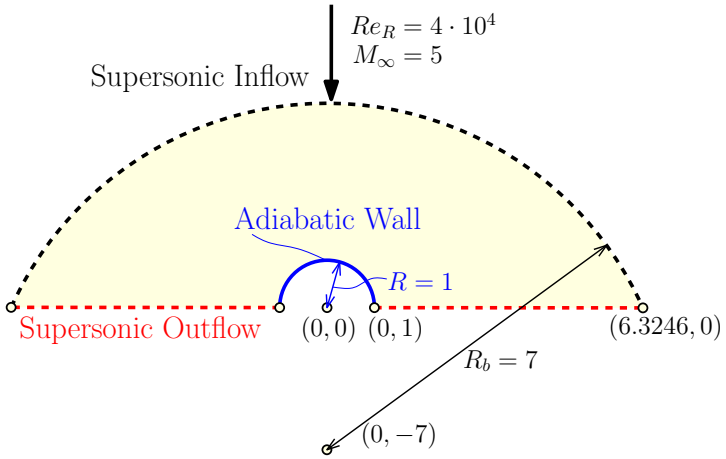
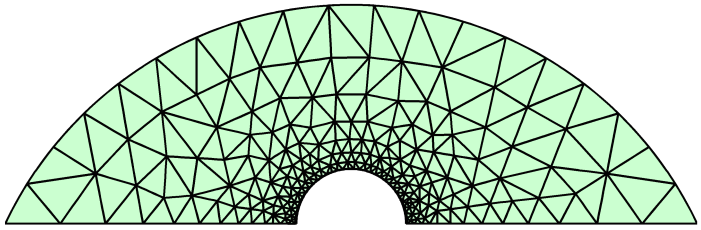


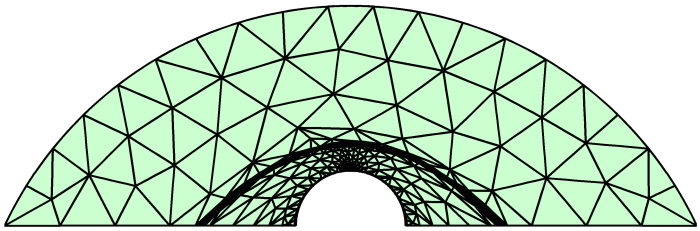

Figure 4-24: Sketch of the geometry and boundary conditions for the case of the high-speed cylinder.

As discussed in Chapter 3, the presence of the shock wave requires some extra adaptivity mechanisms besides the ones provided by the boundary layer adaptivity. In that same spirit, this is done here using BAMG [94] in an outer refinement loop, with the $r$-adaptive solver inside it. The mesh returned by the process is adapted to both the boundary layer and the shock wave. An example of this can be found in Figure 4-25.

A sample of the flow solution obtained after 5 iterations of anisotropic refinement is shown in Figure 4-26. For all these runs, $p = 3$ and $n_{\text{norm}} = 7$. As expected, the



(a) Initial mesh

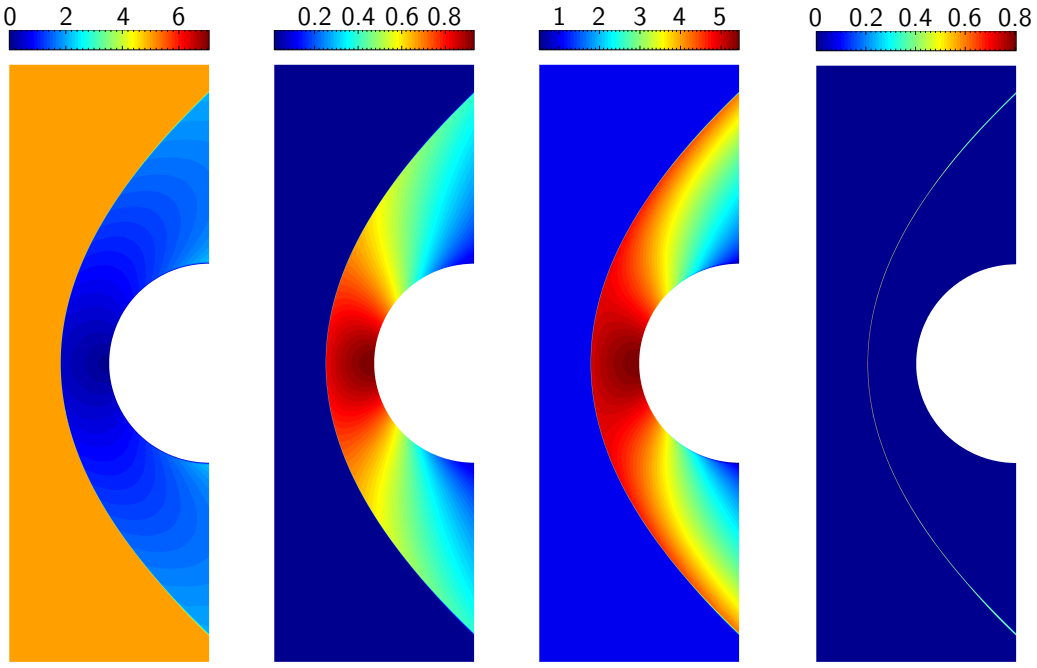


(b) After 5 iterations of anisotropic refinement on the external mesh

Figure 4-25: Meshes used to compute the high-speed flow around a cylinder. The combination of anisotropic refinement and $r$-adaptivity yields meshes that are adapted to the boundary layer and the shock wave using different mechanisms.

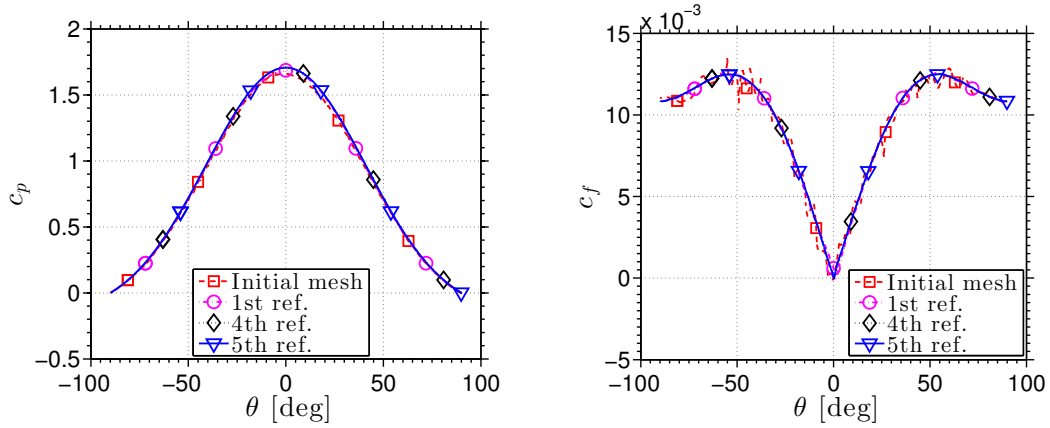
use of adaptivity on the external domain yields sharper shocks and also removes the oscillations in the flow behind the shock, which are a direct cause for inaccuracies in the boundary layer. This is clearly visible in Figure 4-27 where the pressure coefficient is compared to the friction coefficient as the refinement evolves. Notice how this effect is significantly more pronounced in viscous quantities (e.g. $c_f$) that in the inviscid ones (e.g. $c_p$). This is so despite of the fact that the boundary layer mesh is computed on the fly using $r$-adaptivity and there is enough degrees of freedom in the boundary layer thickness to ensure a proper resolution of the flow there.

All this comes to say that the success of this $r$-adaptive methodology in the case of high Mach number flows is constrained by the shock capturing capabilities of the scheme, in addition to the proper adaptation mechanisms around the shock waves.




(a) Mach number (b) Pressure (c) Density (d) Shock indicator

Figure 4-26: High-speed flow field around a cylinder computed on a refined mesh after 5 iterations of anisotropic refinement on the external mesh.




(a) Pressure coefficient (b) Friction coefficient

Figure 4-27: Evolution of the pressure and friction coefficient around the cylinder with the adaptation cycle.

### 4.3.6 Viscous high-speed flow over a blunt wedge

In the previous case, the leading order effect in the interaction between the shock and the boundary layer is the propagation of oscillations at the shock into the boundary layer, which can be minimized by properly resolving the shock wave. In other instances, however, such interaction has a physical basis. In the following section, the attention turns towards the study of the interaction between an entropy layer and a boundary layer, specifically, how the former can interfere with $\delta_{\mathrm{BL}}$ and affect the $r$-adaptivity.

The numerical setting for this case is based on a blunt wedge of 8° half angle in a flow at Mach $M_\infty = 5$ with $Re_R = 10^3$. The details of the geometry as well as boundary conditions are found in Figure 4-28. This geometry features two independent length-scales that stem from different physical phenomena. On the one hand, the blunt nose produces a bow shock that generates a non-uniform entropy profile close to the wedge, also known as an entropy layer. The thickness of said layer remains almost constant downstream of the leading edge and is proportional to the radius of curvature of the nose. On the other hand, the presence of the wall produces the usual boundary layer that grows in thickness along the wedge driven by the pressure gradient.

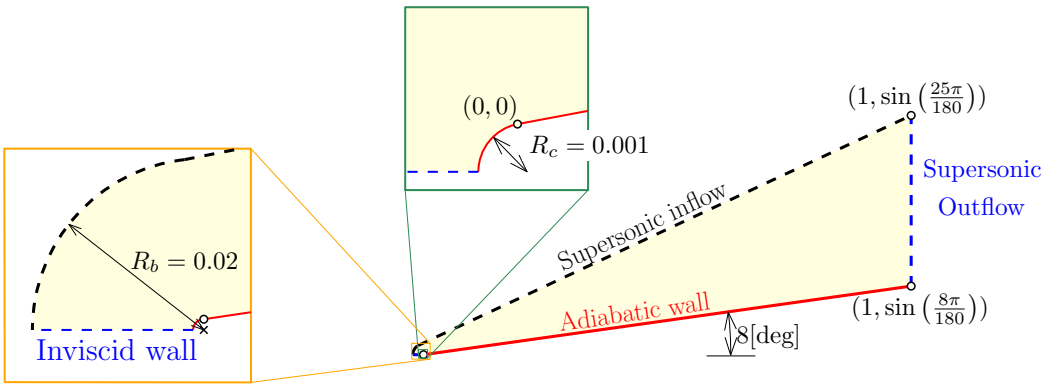
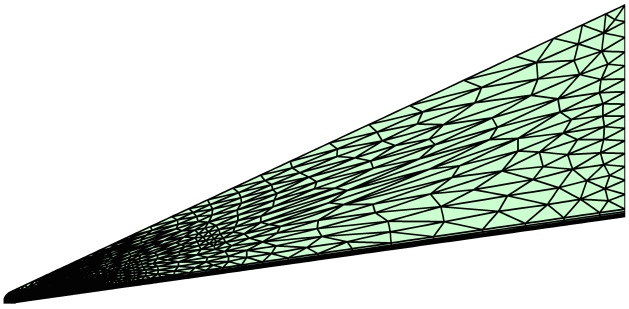



Figure 4-28: Sketch of the geometry and boundary conditions for the case of the blunt wedge.

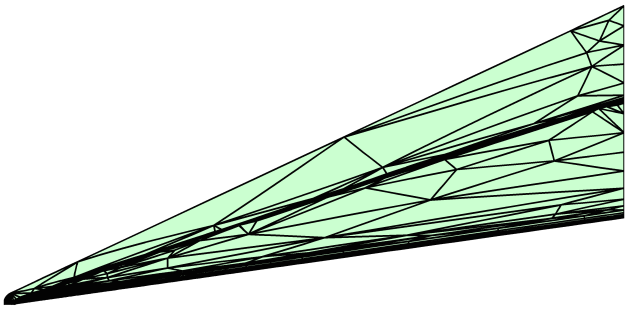
The interaction is due to the fact that the boundary layer is growing inside the entropy layer. From an engineering standpoint, the entropy layer increases the heating

rate close to the leading edge, which is a very important design quantity for hypersonic vehicles. However, rather that focusing on the heating rates, this study is centered on the effect of the entropy layer on the adaptivity, more precisely, what is the effect of the vorticity associated with the entropy layer on the evolution of $\delta$.

For these runs, the same strategy is used to adapt the external mesh and ensure that the numerical oscillations on the bow shock are minimized. The procedure is identical to the one used in the previous test case, and produces meshes like the ones depicted in Figure 4-29 after 5 iterations of BAMG. All the results presented here were run using $p = 3$ and $n_{\text{norm}} = 7$. The use of anisotropic adaptivity on the external mesh yields cleaner solutions for the pressure and friction coefficient, as is reflected in Figure 4-30.

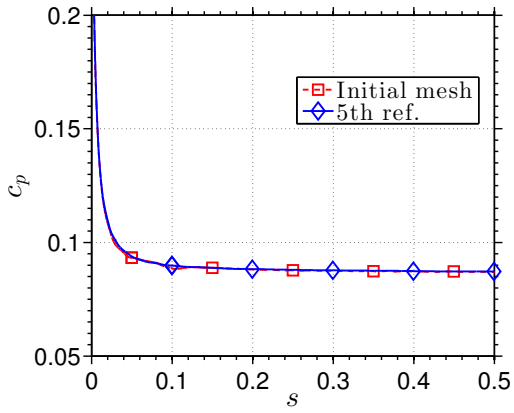


(a) Initial mesh

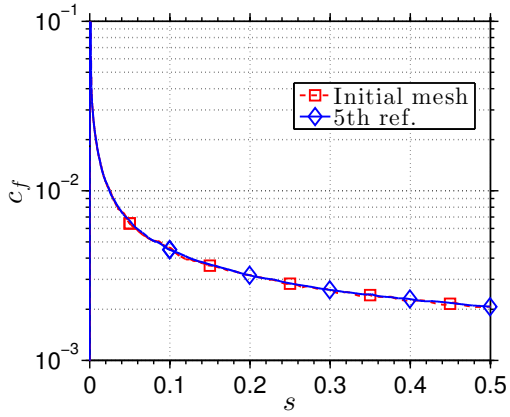


(b) Mesh after 5 iterations of anisotropic refinement on the external mesh

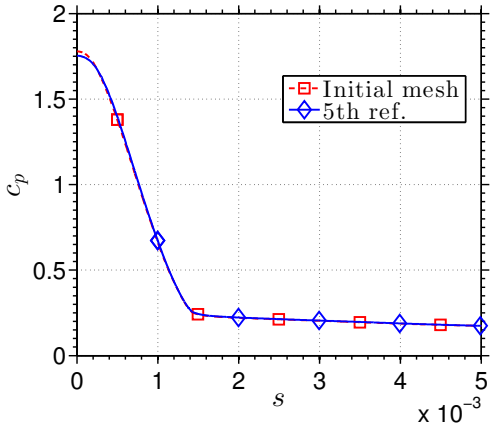
Figure 4-29: Evolution of the mesh with the anisotropic adaptation for the case of the blunt wedge.



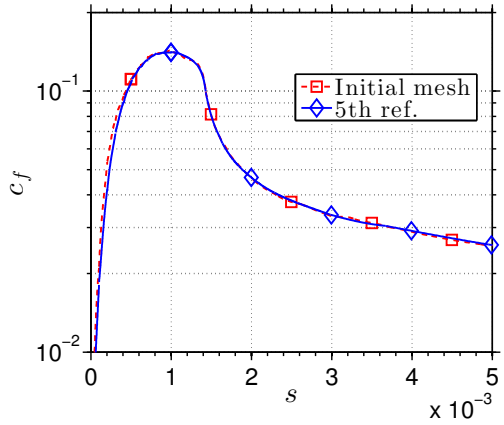

(a) Pressure coefficient




(b) Friction coefficient




(c) Pressure coefficient around the leading edge

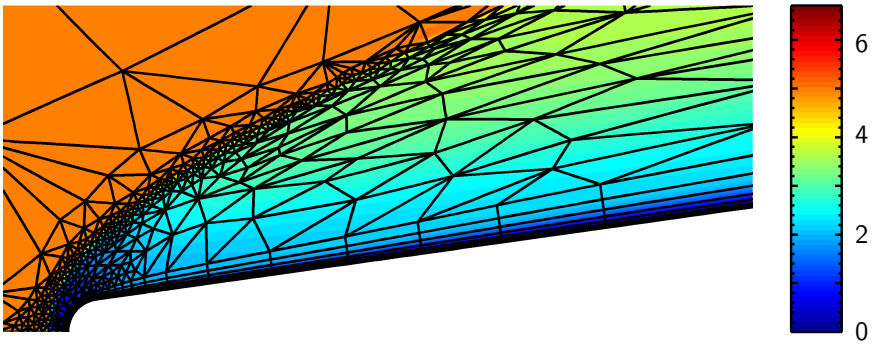



(d) Friction coefficient around the leading edge

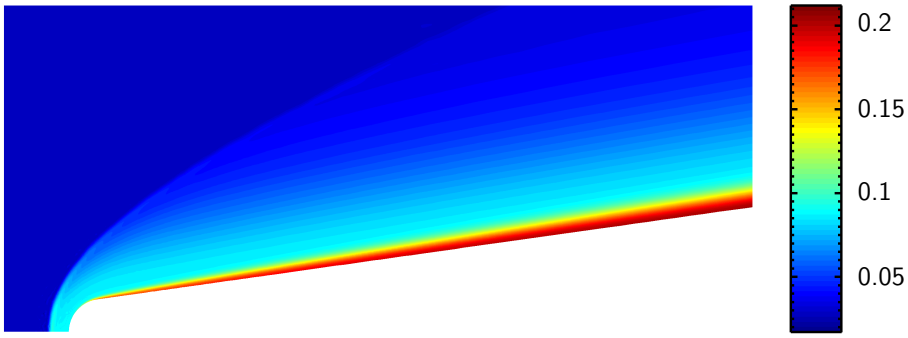
Figure 4-30: Pressure and friction coefficient over the blunt wedge. The variable $s$ denotes the arc parameter from the stagnation point. For clarification purposes, a detail of both around the leading edge is included in the bottom row.

The resulting flow field around the leading edge is rendered in Figure 4-31. Notice how the boundary layer evolves independently of the entropy layer in the vicinity of the nose. This is so until both length scales are comparable, at which point $\delta$ jumps the edge of the entropy layer. Downstream of this point, the growth of the boundary layer eventually entrains the entropy layer. This is visualized in Figure 4-32, where the entropy and velocity profiles are plotted at three different locations along the surface of the wedge. This phenomenon is also visible in Figure 4-33, that contains plots of the trace of the entropy at the edge of the boundary layer domain, as well as the evolution

of $\delta$ itself. In principle, this behavior does not affect the mesh adaptivity around the leading edge, that is still properly addressed by the combination of BAMG and the $r$-adaptivity. However, this is merely coincidental with the parameters of the problem, as a lower value of $Re_R$ might move the interaction point closer to the leading edge.

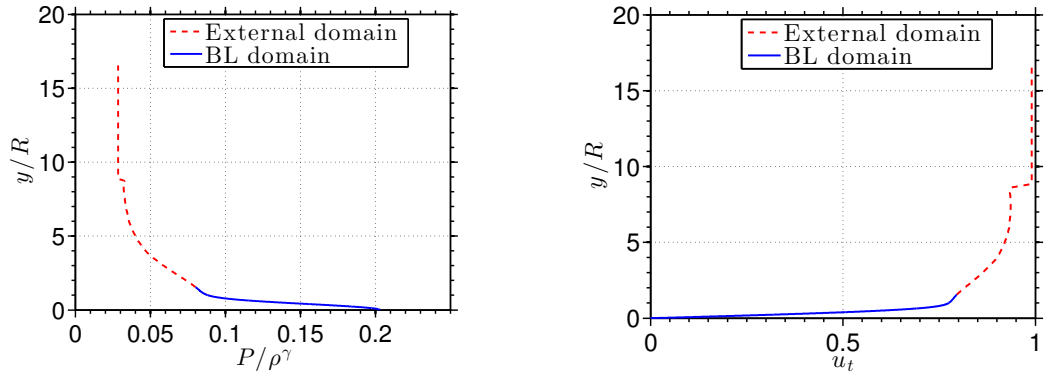



(a) Mach number and mesh

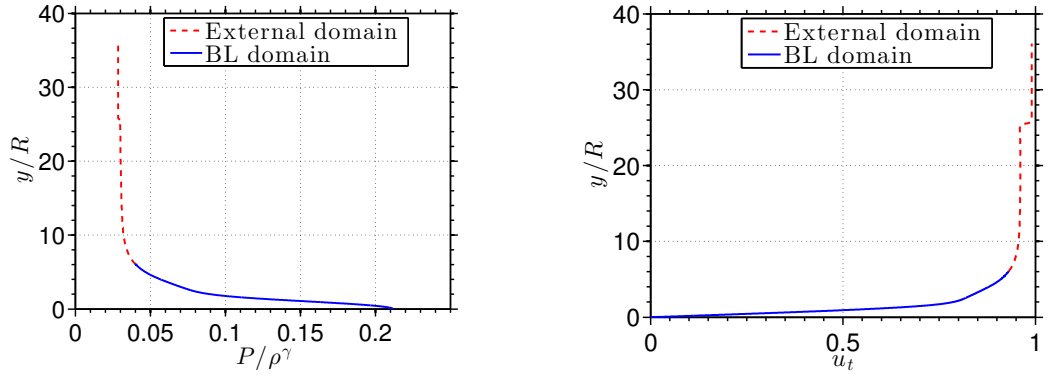



(b) Entropy

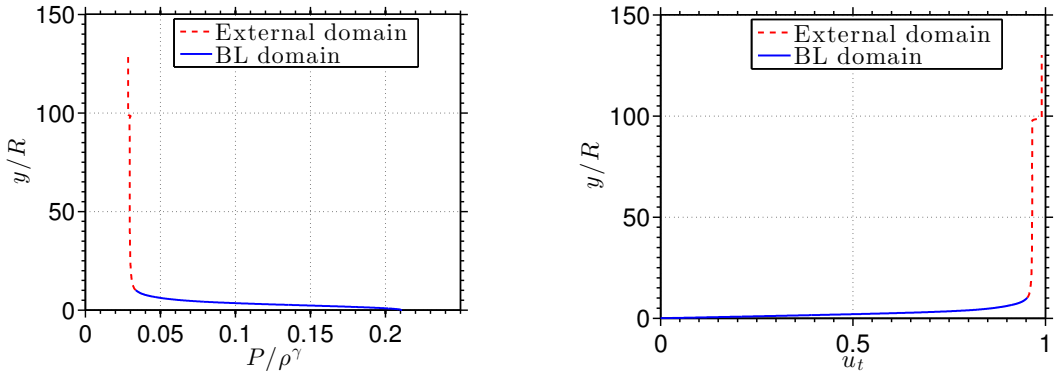
Figure 4-31: Rendering of the Mach number and entropy around the leading edge of the blunt wedge. The characteristic thickness of the boundary layer and the entropy layer are clearly visible in both quantities.




(a) Entropy profile at $s = 0.02$ (b) Velocity profile at $s = 0.02$



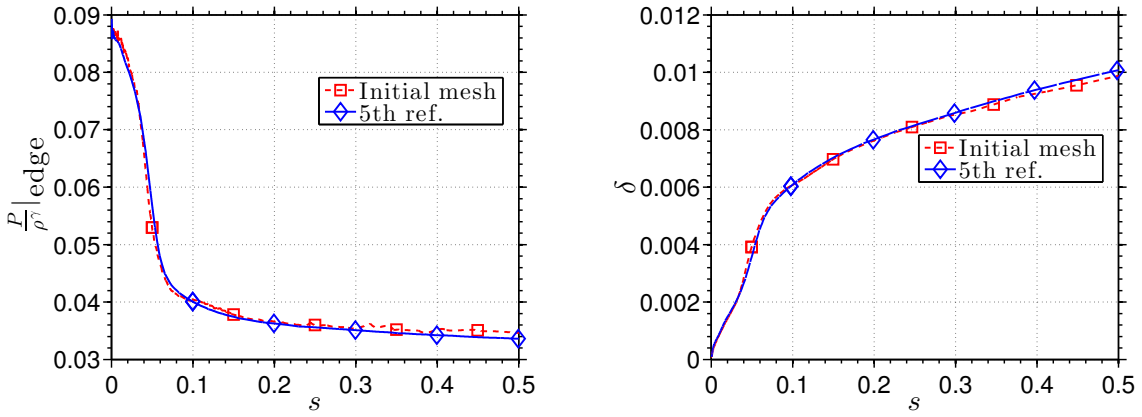

(c) Entropy profile at $s = 0.1$ (d) Velocity profile at $s = 0.1$




(e) Entropy profile at $s = 0.5$ (f) Velocity profile at $s = 0.5$

Figure 4-32: Entropy and velocity profiles at different stations along the blunt wedge. The entropy layer as well as the associated vorticity layer are visible at $s = 0.02$ but the details fade away as the boundary layer grows, being almost unnoticeable at $s = 0.5$.

In short, the numerical evidence suggests that there is an interaction. However, the effect of the entropy layer on the $r$-adaptivity is minor for Reynolds numbers over




(a) Entropy at the edge of the boundary layer domain

(b) Normal scaling

Figure 4-33: Plot of the entropy at the edge of the boundary layer domain as well as the normal scaling along the wedge. The variable $s$ in the $x$-axis denotes the arc length from the stagnation point of the wedge. The jump of the normal scaling over the entropy layer is visible on both around $s = 0.05$.

$Re_R = 10^3$. Extending it below this limit would first require a careful assessment of what the limit actually is, and probably a more sophisticated way of computing $\delta_{\text{BL}}$ that takes into account the entropy layer. A possible way of doing this would be to reconstruct the entropy layer profile using the vorticity of the flow at the edge of the boundary layer mesh, and subtract it from the actual flow to get a better approximation of the effects due to the boundary condition versus the effects due to the curvature of the shock upstream. This study will not be undertaken in this thesis, but instead, is proposed as a future extension.

### 4.3.7 Shock wave boundary layer interaction

In the previous test cases, the interaction between the shock wave and the boundary layer happens at a distance through the convection of the state behind the shock. This makes the interaction predominantly one way. In many instances though, the shock wave impinges on the boundary layer directly, which opens the door for a stronger feedback mechanism between them. Of particular interest here is the so-called strong shock wave boundary layer interaction [61] which happens when a shock hits the boundary layer and is strong enough to produce separation in it. When this happens, the shock does not reflect off the wall as it would in the inviscid case (or if the boundary layer does not separate), but rather turns into an expansion fan at the edge of the boundary layer, plus two compression waves around the separation point and the reattachment point. A sketch of the flow pattern in this case is shown in Figure 4-34.

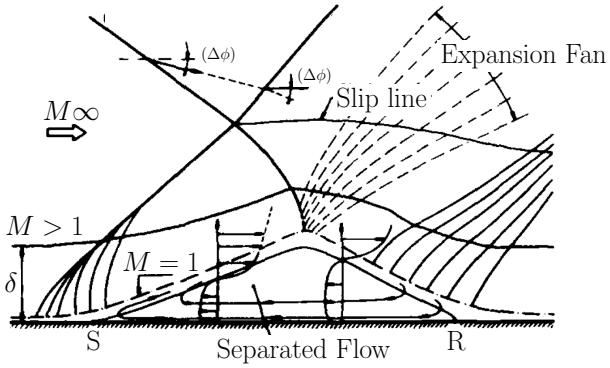



Figure 4-34: Sketch of the different flow features in a shock wave boundary layer interaction with separation. Drawing adapted from Delery and Marvin [61].

The present section focuses on the applicability of the $r$-adaptivity to such flows. For this, the test case by Degrez *et al.* [60] is reproduced here. The geometry is composed of a flat plate and a shock generator inside a stream at $M_\infty = 2.15$, with a Reynolds number of $Re_L = 10^5$. A sketch of the geometry together with the boundary conditions can be found in Figure 4-35. The rounding of the geometry at the leading edge of the flat plate was introduced to avoid the boundary condition singularity (also

known as Carter's flat plate problem [37]). In any case the magnitude of the rounding is the same as the one measured by Degrez *et al.* [60] in the physical experiment.

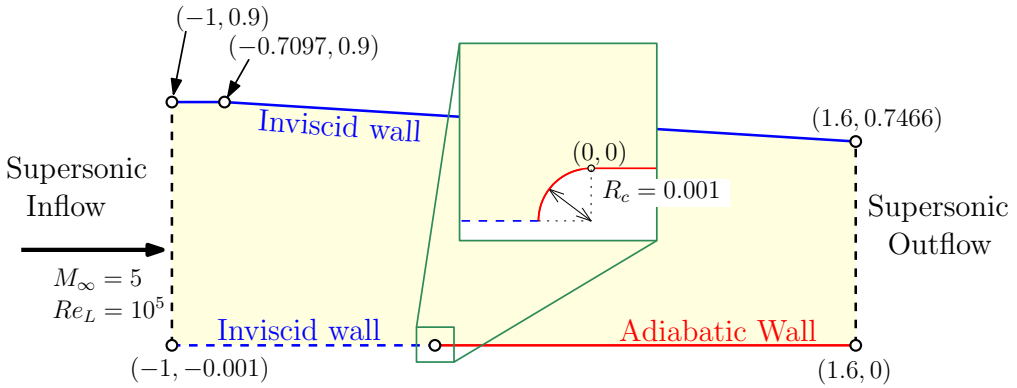
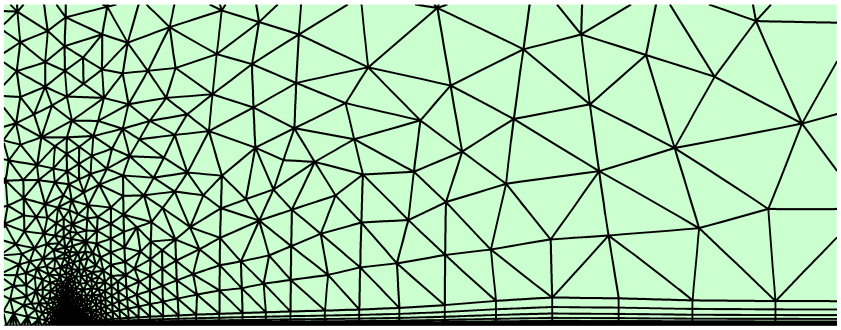


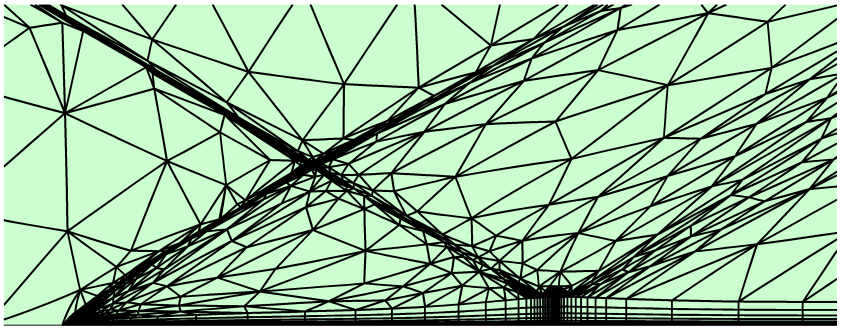

Figure 4-35: Sketch of the geometry and boundary conditions for the case of the shock wave boundary layer interaction.

As in the previous cases, the external mesh is refined iteratively using BAMG in an outer loop, and the $r$-adaptive solver between iterations. All the runs presented here were computed using polynomials of order $p = 3$ and a fixed number of elements across the boundary layer of $n_{\text{norm}} = 7$. For comparison purposes, the meshes obtained in the first and last refinement iteration are plotted in Figure 4-36. A rendering of the flow field obtained on the latter can be found in Figure 4-37. In addition, the boundary layer profiles are plotted in Figure 4-38. Notice how all the flow features sketched in Figure 4-34, like the separation bubble, or the various wave patterns at the edge of the boundary layer, can be easily identified in these plots.

The role of adaptivity in these results can be assessed by looking at the evolution of the stresses at the wall with each iteration of BAMG, as plotted in Figure 4-39. These seem to be fairly insensitive to the refinement of the external mesh except at the region immediately adjacent to the point where the shock hits the wall (see right column of Figure 4-39). This is so even though the pressure signature at the edge of the boundary layer becomes increasingly sharp as the region is refined, as plotted in Figure 4-40. This behavior can be explained by the presence of the laminar separation bubble, and how it reacts to smear out any sharp pressure gradient, as explained by Delery and Marvin [61]. These results highlight that for the case of shock induced



(a) Initial mesh

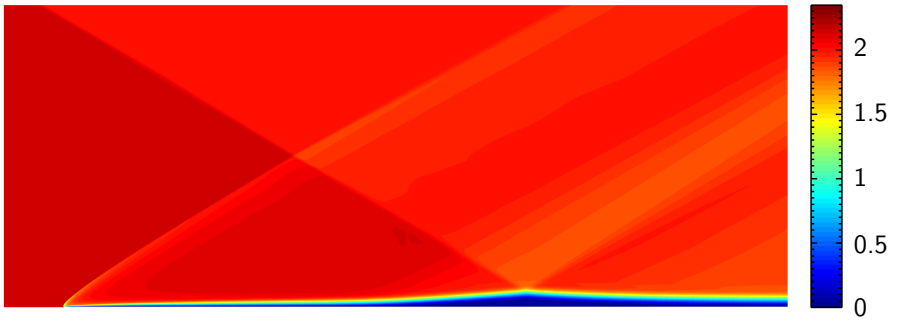


(b) Final mesh after 9 iterations of anisotropic refinement on the external mesh

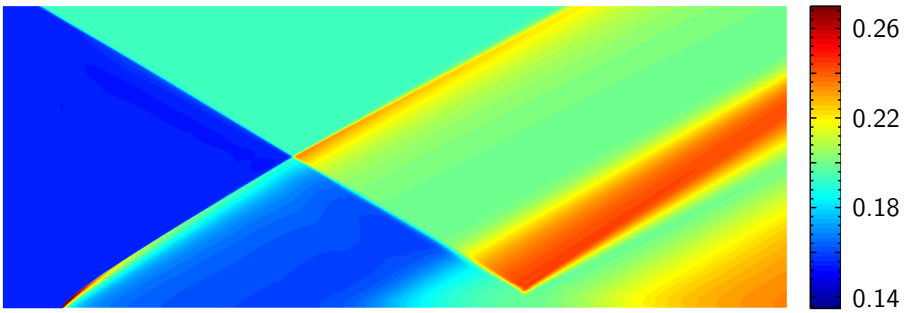
Figure 4-36: Evolution of the mesh around the separation bubble in the shock wave boundary layer interaction case.

separation, adapting to the thickness of the boundary layer is at least as important as producing thin shock structures.

In qualitative terms, the computed results compare well with the experimental curve by Degrez *et al.* [60] for the pressure coefficient, but strongly disagree with the numerical simulations presented in the same paper. According to the authors, this disagreement might be due to experimental errors, but the results presented here seem to indicate that it might be more of an issue with the numerics. In the absence of better data, this question is not further investigated.

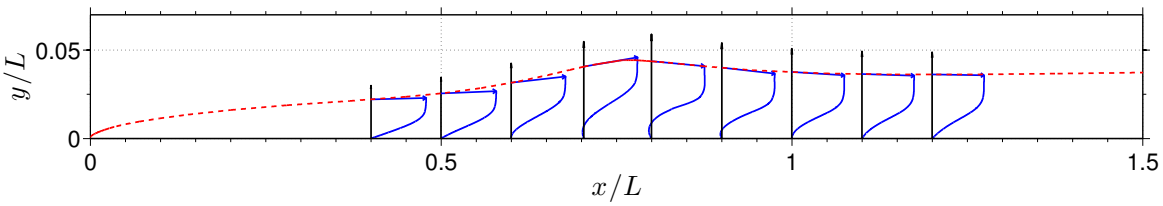



(a) Mach number

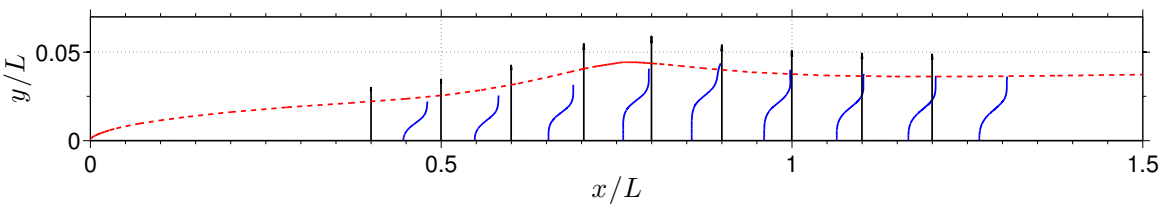



(b) Pressure

Figure 4-37: Mach number field and pressure field around the flat plate. The separation and reattachment of the boundary layer due to the shock is visible as compression waves in the pressure field.

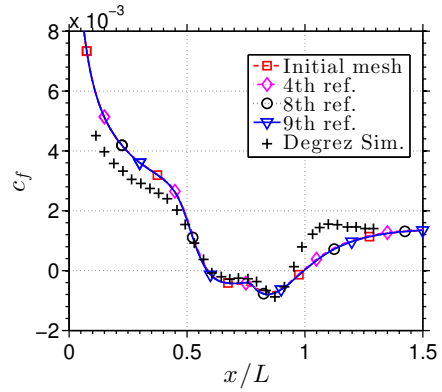



(a) Hodograph of the velocity

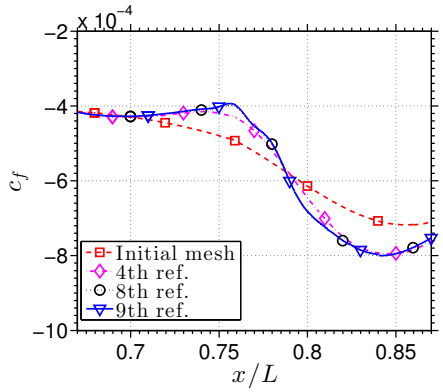



(b) Density profiles

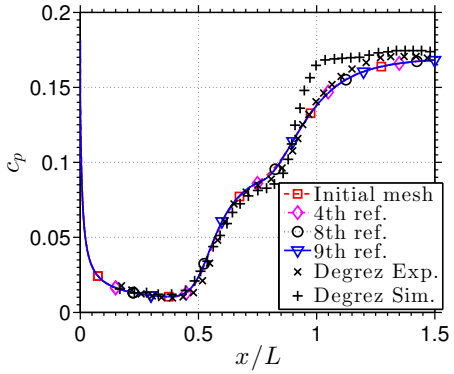
Figure 4-38: Velocity and density profiles around the laminar separation bubble extracted from the boundary layer domain. The dashed line denotes the edge of the boundary layer domain.



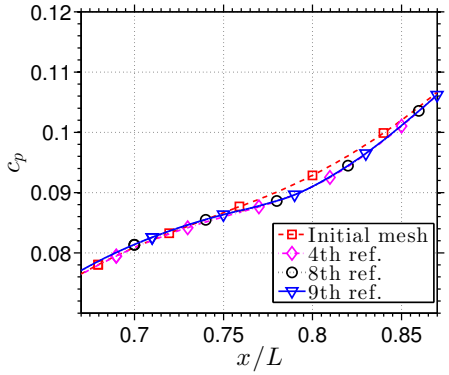

(a) Friction coefficient




(b) Detail of the friction coefficient

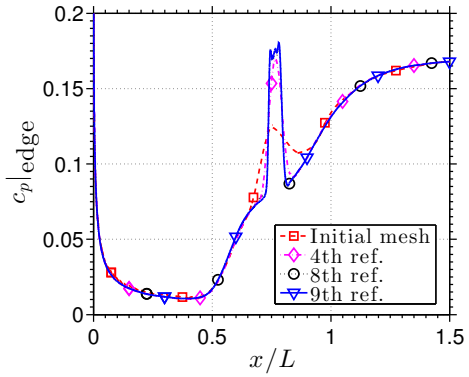



(c) Pressure coefficient

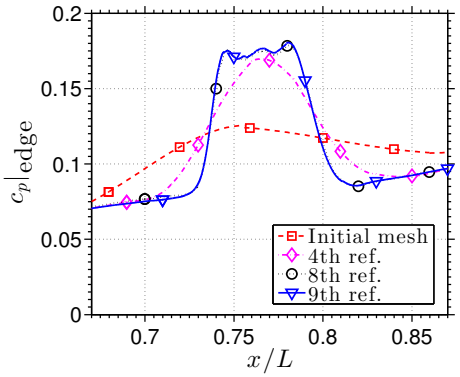



(d) Detail of the pressure coefficient

Figure 4-39: Evolution of the stresses at the wall with the external mesh adaptation cycle. The effect of sharper shock profiles can only be seen in the vicinity of the region where the shock impinges (right column).




(a) Pressure coefficient at the edge of the boundary layer domain




(b) Detail of the pressure coefficient at the edge of the boundary layer domain

Figure 4-40: Pressure signature at the edge of the boundary layer domain as a function of the external mesh adaptation cycle.

All in all, these results have served to prove that the $r$-adaptivity proposed as part of this thesis also works in situations where there is a strong interaction between the shock wave and the boundary layer.

## 4.4 Conclusions and future work

In this chapter, a novel $r$-adaptive method for viscous flows has been presented. This strategy is based on a one-shot approach to solve the flow field together with a mesh that is adapted to the boundary layer.

The mechanics of the adaptivity relies on a mesh that has a fixed hybrid topology, and is deformed according to a measure of the boundary layer thickness. Such quantity is derived through the boundary layer profiles, that are easily extracted from the mesh thanks to its topology. This circular dependence is one of the keys to the success of the algorithm, as well as a possible liability. The latter is associated to the existence of more than one solution for separated flows, and has been carefully studied with the help of a surrogate problem in Appendix A. The outcome of that exercise has been that the problem can be avoided by means of a judicious choice of initial condition and parameters for the mesh generation. In addition, following these guidelines ensures that the mesh tracks the boundary layer throughout the steady state relaxation, hence reducing the stiffness of the nonlinear systems that need to be solved.

The results produced for a variety of 2-D flows (attached, separated, laminar, turbulent, with and without shocks) have served to evaluate the performance of the scheme in metrics such as accuracy per degree of freedom or grid control (through $y_1^+$) close to the wall. Also, the results compare favorably to general anisotropic strategies in the case of high Mach number flows, and, without loss of generality, can be combined with them to produce meshes that are adapted to the boundary layer as well as other features like shock waves. As expected, the solver can be improved in a variety of ways.

First of all, notice that method is general enough to treat a variety of mesh topolo-

gies provided that they abide to the requirements in Section 4.2.1. This includes cases in which there is more than one boundary layer domain (e.g. a multi-element airfoil) or there exist discontinuities in the geometry that prevent the extrusion of the boundary layer mesh (e.g. a flap cove). In principle, the extension to these kinds of problems is a mere implementation task, that is left open as future development of the code. The same applies to the meshes obtained near the trailing edge (see Figure 4-20), that despite not being a source of trouble in the cases presented before, could be handled in a seamless way using other mesh generation algorithms.

In addition to this, the mesh generation module could be augmented in a variety of ways, for example, by adding an extra surface PDE to control $y_1^+$ through the stack, or by deforming the boundary layer mesh instead of prescribing it analytically. A particularly interesting line of research would be the combination of this $r$-adaptivity with some form of $hp$-adaptivity to generate the external mesh and guide the choice of approximation order $p$, number of elements $n_{\text{norm}}$ across the boundary layer, as well as the stack distribution $\{h_i\}$.

In terms of applicability to other flow regimes, notice that the scheme readily extends to unsteady turbulent boundary layers under the assumption that these can be modeled using the Unsteady RANS equations. An example of this can be found in the next chapter. Furthermore, it could also be used in LES-type simulations, provided the equation for $\delta$ was filtered in time to reflect the average boundary layer thickness rather than the instantaneous one.

Finally, notice that solving the mesh generation and the flow discretization in a simultaneous manner incurs an overhead with respect to the cost of solving the same flow on a fixed mesh. A lower bound for such overhead is the increase in degrees of freedom of the problem, which is dominated by the unknowns associated with the location of the high order nodes that define the external mesh. Unfortunately such estimate is overly optimistic since it ignores the increase in memory required to store the Jacobian as well as the cost of solving the associated linear systems. While this has not been a concern for the prototype solver developed in this thesis, it will

require a careful study in the future. This could be tackled with the use of some form of staggered update to algebraically decouple the mesh generation from the flow discretization.

# Chapter 5

# Transition prediction on the $r$-adaptive solver

## 5.1 Introduction

The novelty of the $r$-adaptive solver presented in the previous chapter resides in the way the mesh is generated "on the fly" to adapt to the boundary layer. At the heart of this adaptive process is the ability to extract the velocity profiles in the vicinity of solid walls thanks to the inherent structure of the mesh. In this chapter, the availability of such high quality boundary layer profiles is used to predict transition to turbulence in the RANS equations, thereby extending the range of applications of the solver.

This approach represents a departure from the classical one and circumvents some of the existing problems reported in the literature such as the treatment of separated flow or the mesh requirements across the boundary layer. A detailed literature review on the topic was presented in Section 1.3.4.

The simulation of transitional flows is of considerable importance for the design and optimization of a variety of devices, like high lift appendages [238], nacelles [196] or low Reynolds number airfoils[196], for which the location of the region where the flow turns turbulent can be the leading order effect on the outputs of interest. Leaving aside the use of more costly simulation techniques like LES or DES, the focus here is

placed on predicting transition in RANS solvers. Within this subject, there is a clear distinction in the literature between approaches that model transition as an extra field in the domain, and approaches that only perform the analysis in the boundary layer region.

The strategies in the first group usually involve the use of an extra set of PDEs to model quantities such as the amplification factor [52], the intermittency [133, 151] or a pointwise approximation to certain boundary layer quantities like $Re_\theta$ [133]. In terms of cost and implementation, these models incur a small overhead compared to fully turbulent RANS solutions, which can be very attractive provided the nonlinear stiffness of the combined system is not increased. Although easy to implement, these models are largely *ad hoc* and do not take advantage of the existing body of research on boundary layer stability.

On the contrary, the models in the second group are based on the analysis of the stability characteristics of the boundary layer profiles through the linear amplification regime, which in most instances, dominates the length of the transition process [13, 163]. This makes them more theoretically sound even though there is still some level of approximation involved (see Section 1.3.4). However, they require a dedicated solver for the transition (with the appropriate handles to couple it to the RANS solver), as well as having access to the boundary layer profiles.

Regarding this issue, there are two typical approaches, namely: strategies that extract the boundary layer profiles from the discrete solution [197, 229, 163], and strategies that use an intermediate boundary layer code driven by the pressure of the RANS solver to approximate the velocity profiles [131, 127, 185]. These two strategies represent different ways of generating boundary layer data each with its own limitations. In particular, extracting the boundary layer profiles directly from the discrete solution requires structure in the mesh, a reliable way to identify the boundary layer edge, and, a minimum number of degrees of freedom in the viscous domain [197, 229]. On the other hand, using an intermediate boundary layer solver limits the applicability of the method to attached flows [131].

All these constraints stem from the assumption that the mesh is generated *a priori* and fixed throughout the solution process. However, when looking at it from the perspective of the $r$-adaptive solver presented in Chapter 4, all the aforementioned conditions are automatically satisfied. The purpose of this chapter is to take advantage of this and extend the solver to account for transition using an $e^N$ method.

The structure of the chapter is as follows. First, the particular details of the formulation and the discretization are presented in detail. Next, the coupling on the equations that govern transition with the rest of the modules of the solver is briefly discussed. Then, the method is applied to a variety of transitional flows to demonstrate its capabilities. Finally, the chapter closes with conclusions and future work.

## 5.2 Formulation

As stated above, this work will use a variant of the $e^N$ method [222, 242] to track the growth of perturbations in the boundary layer in the linear regime. In the case of 2-D flows, such perturbations take the form of Tollmien-Schlichting (TS) waves, whose amplification ($N_{\mathrm{TS}}$) can be computed by integrating the growth rate ($\alpha_i$) along the boundary layer past the neutral point ($s_0$) in the following fashion:

$$N_{\mathrm{TS}} = \log(A/A_0) = \max_{\omega}\left[\int_{s_0}^{s} -\alpha_i(\omega, \mathbf{u}(s))ds\right]. \tag{5.1}$$

Here, $\omega$ represents the temporal frequency of the perturbations, and is one of the two arguments that modulates the growth rate $\alpha_i$, the other one being the boundary layer profiles, denoted by $\mathbf{u}(s)$. In the $e^N$ method, transition is assumed to happen at the location $s_{\mathrm{tr}}$ where the amplification factor first reaches a certain threshold $N_{\mathrm{crit}}$, that can be correlated to the turbulence intensity of the freestream using Mack's relationship [144]. Mathematically, this condition reads:

$$\max_{\omega}\left[\int_{s_0}^{s_{\mathrm{tr}}} -\alpha_i(\omega, \mathbf{u}(s))ds\right] = N_{\mathrm{crit}}. \tag{5.2}$$

The presence of the maximization operator in Eq. 5.2 complicates the implementation of this criterion since it implies a search for the maximum in the range of $\omega$. This issue has been addressed in a variety of ways in the literature as discussed in Section 1.3.4, pp. 37. The model proposed by Drela and Giles [69] is used in this thesis because it retains most of the complexity of the problem while being analytical and differentiable. In this model, the explicit form of the transition criterion reads:

$$\int_{s_0}^{s_{\mathrm{tr}}} \frac{dN}{ds}(\theta_{\mathrm{k}}, H_{\mathrm{k}})ds = N_{\mathrm{crit}}, \tag{5.3}$$

where the different functions that enter the integrand are defined as:

$$\frac{dN}{ds}(\theta_{\mathrm{k}}, H_{\mathrm{k}}) = \frac{dN}{dRe_\theta}(H_{\mathrm{k}})\frac{m(H_{\mathrm{k}})+1}{2}l(H_{\mathrm{k}})\frac{1}{\theta_{\mathrm{k}}}, \tag{5.4}$$

$$\frac{dN}{dRe_\theta}(H_{\mathrm{k}}) = 0.01\sqrt{\left(2.4H_{\mathrm{k}} - 3.7 + 2.5\tanh(1.5H_{\mathrm{k}} - 4.65)\right)^2 + 0.25}, \tag{5.5}$$

$$l(H_{\mathrm{k}}) = \frac{6.54H_{\mathrm{k}} - 14.07}{H_{\mathrm{k}}^2}, \tag{5.6}$$

$$m(H_{\mathrm{k}}) = \left(0.058\frac{(H_{\mathrm{k}}-4)^2}{H_{\mathrm{k}}-1} - 0.068\right)\frac{1}{l(H_{\mathrm{k}})}, \tag{5.7}$$

and, the lower limit of integration in Eq. 5.3 is taken to be the point that satisfies:

$$Re_\theta(s_0) = Re_{\theta 0}, \tag{5.8}$$

$$\log_{10} Re_{\theta 0} = \left(\frac{1.415}{H_{\mathrm{k}}-1} - 0.489\right)\tanh\left(\frac{20}{H_{\mathrm{k}}-1} - 12.9\right) + \frac{3.295}{H_{\mathrm{k}}-1} + 0.44. \tag{5.9}$$

Note that the model described by Eq. 5.3-5.9 only depends on the distribution of the integral boundary layer quantities $\delta_{\mathrm{k}}^*$ and $\theta_{\mathrm{k}}$ along the boundary layer, which are readily available as an output of the $r$-adaptive scheme presented in Chapter 4.

### 5.2.1 Governing equations

To determine the location of transition requires finding a solution to the integral equality in Eq. 5.3. As written, this particular criterion is not amenable to numerical discretization. In what follows, Eq. 5.3 is modified into an approximate system that can be discretized using common techniques.

The starting point of this process is to re-write the equation so that the lower limit in the integral does not require an evaluation of the $Re_\theta = Re_{\theta 0}$ condition. This is easily achieved if this condition is written in the form of a Heaviside function, denoted by $H(\cdot)$, and moved into the integrand. In this way, Eq. 5.3 becomes:

$$\int_0^{s_{\mathrm{tr}}} H(Re_\theta - Re_{\theta 0}) \frac{dN}{ds}(\theta_{\mathrm{k}}, H_{\mathrm{k}}) ds = N_{\mathrm{crit}}, \tag{5.10}$$

where $s = 0$ in the lower limit of integration denotes the stagnation point. Here, the Heaviside function $H$ is not to be mistaken with the shape parameter $H_{\mathrm{k}}$. The use of $H$ in the implementation of the switch makes the integrand on the left hand side of Eq. 5.10 discontinuous at the point $Re_\theta - Re_{\theta 0}$. To alleviate this, $H(Re_\theta - Re_{\theta 0})$ is approximated by a smooth surrogate $\widetilde{H}(Re_\theta, Re_{\theta 0})$ in the form:

$$H(Re_\theta - Re_{\theta 0}) \approx \widetilde{H}(\theta_{\mathrm{k}}, H_{\mathrm{k}}) = \frac{1}{2}\left(\tanh\left(10\left(\frac{Re_\theta(\theta_{\mathrm{k}})}{Re_{\theta 0}(H_{\mathrm{k}})} - 1\right)\right) + 1\right), \tag{5.11}$$

and the new condition for the location of the transition point becomes:

$$\int_0^{s_{\mathrm{tr}}} \widetilde{H}(\theta_{\mathrm{k}}, H_{\mathrm{k}}) \frac{dN}{ds}(\theta_{\mathrm{k}}, H_{\mathrm{k}}) ds = N_{\mathrm{crit}}. \tag{5.12}$$

As written, Eq. 5.12 represents a condition to find the transition point $s_{\mathrm{tr}}$, which delimits the boundary between the laminar and turbulent regions in the flow. In most instances, however, the interest lies on whether or not a spot is turbulent. This leads to the following true or false statement represented by the binary variable $\gamma_{\tilde{n}}$, in the form:

$$\tilde{n}(s) = \int_0^s \widetilde{H}(\theta_{\mathrm{k}}, H_{\mathrm{k}}) \frac{dN}{ds}(\theta_{\mathrm{k}}, H_{\mathrm{k}}) ds, \tag{5.13}$$

$$\gamma_{\tilde{n}} = \begin{cases} 0 \text{ for } \tilde{n} < N_{\mathrm{crit}}, \\ 1 \text{ for } \tilde{n} \geq N_{\mathrm{crit}}. \end{cases} \tag{5.14}$$

Here, the intermediate variable $\tilde{n}(s)$ is an approximation to the amplification factor along the boundary layer.

Notice that the integral in Eq. 5.13 requires an explicit knowledge of the location of the stagnation point since it represents the lower limit of integration $s = 0$. From the point of view of implementation, this is a serious inconvenience as the location might be changing along with the iteration to steady state or at every time step if the flow is unsteady. To circumvent this issue, Drela [66, 239] proposed to cast the integral as a hyperbolic equation on the surface, using the edge velocity as the convection field. In this work this same approach is used, with the exception that the convection field is normalized to have almost unit norm. In this way, Eq. 5.13 is turned into a PDE for $\tilde{n}$ that reads:

$$\nabla_\Gamma \cdot \left( \frac{\mathbf{v}_\Gamma^{\mathrm{e}}}{\widetilde{|\mathbf{v}_\Gamma^{\mathrm{e}}|}} \tilde{n} \right) = \widetilde{H}(Re_\theta, Re_{\theta 0}) \frac{dN}{ds}(\theta_{\mathrm{k}}, H_{\mathrm{k}}), \tag{5.15}$$

where $\nabla_\Gamma \cdot ()$ represents the surface divergence operator, $\mathbf{v}_\Gamma^{\mathrm{e}}$ represents the edge velocity projected in the surface direction, and $\widetilde{|\mathbf{v}_\Gamma^{\mathrm{e}}|}$ is a regularized version of the norm, according to the following expression:

$$\widetilde{|\mathbf{v}_\Gamma^{\mathrm{e}}|} = \|\mathbf{v}_\infty\| \sqrt{\frac{\log\left(1 + \exp\left(a \frac{\mathbf{v}_\Gamma^{\mathrm{e}} \cdot \mathbf{v}_\Gamma^{\mathrm{e}}}{\|\mathbf{v}_\infty\|^2}\right)\right)}{a}}. \tag{5.16}$$

Here, the parameter $a$ governs how close the convection field $\mathbf{v}_\Gamma^{\mathrm{e}} / \widetilde{|\mathbf{v}_\Gamma^{\mathrm{e}}|}$ is to a sign function with respect to the arc parameter in the vicinity of the stagnation point. In all the results presented in this chapter: $a = 10^2$.

The model needs to be closed by defining the appropriate boundary conditions. Given the convective character of the differential operator, these are set to homogeneous Dirichlet ($\tilde{n} = 0$) for inflow and extrapolation of $\tilde{n}$ for outflow. Here the inflow/outflow character is given by the direction of the convective field at the boundaries.

As written, the model does not include any explicit mechanism for the evolution of $\tilde{n}$ in time, which makes the solution react immediately to the convective field and the forcing term on the right hand side. For convergence purposes, however, some level of damping in $\tilde{n}$ is desired. This is achieved by the simple addition of a time dependent term to the left hand side of Eq. 5.15[1]:

$$\frac{\partial}{\partial t}\left(\frac{\tilde{n}}{\widetilde{|\mathbf{v}_\Gamma^e|}}\right) + \nabla_\Gamma \cdot \left(\mathbf{v}_\Gamma^e \frac{\tilde{n}}{\widetilde{|\mathbf{v}_\Gamma^e|}}\right) = \widetilde{H}(Re_\theta, Re_{\theta 0})\frac{dN}{ds}(\theta_k, H_k). \tag{5.17}$$

In summary, the $e^N$ method of Drela and Giles has been transformed here into an equivalent system composed of a PDE (Eq. 5.17) and an intermittency condition (Eq. 5.14) that can be used to segregate the laminar and turbulent regions once the governing equation for $\tilde{n}$ is solved.

### 5.2.2 High order surface HDG

The first step in the solution of Eq. 5.17 is to define the domain in which $\tilde{n}$ is supported. In this case, $\tilde{n}$ as well as the governing PDE are defined on the manifold that serves as basis for the boundary layer, which is usually associated with the solid wall. This is equivalent to the geometry that supports $\delta$, and hence can be discretized accordingly. In particular, the geometry is approximated using the high order surface mesh $\Gamma_h$ defined in Section 4.2.2.

The next step in the process is to discretize Eq. 5.17 on said mesh using a numerical scheme that can handle hyperbolic surface PDEs and produce a high order approxi-

[1] As written, the model is not frame invariant since $\nabla_\Gamma \cdot \mathbf{v}_\Gamma^e \neq 0$, however, it could be easily modified to account for this by adding $(\nabla_\Gamma \cdot \mathbf{v}_\Gamma^e)\frac{\tilde{n}}{\widetilde{|\mathbf{v}_\Gamma^e|}}$ as a source term.

mation to $\tilde{n}$. The last condition is a requirement to avoid the suboptimal convergence caused by a mismatch in resolution between the different variables that enter the system. While there are examples in the literature that satisfy the first condition (e.g. the finite volume scheme by Roshmanith, Bale and LeVeque [206]), no reference was found on any discretization scheme that fulfills both. This observation triggered the development of a novel surface hybridizable discontinuous Galerkin scheme (SHDG), that is presented next.

The starting point of the SHDG formulation is a hyperbolic surface PDEs of the form:

$$\frac{\partial u}{\partial t} + \nabla_\Gamma \cdot \mathbf{f}(u) = s, \text{ in } \Gamma. \tag{5.18}$$

Here, $u$ denotes the unknown of the problem, $\nabla_\Gamma \cdot ()$ denotes the surface divergence operator, $\mathbf{f}$ denotes a generic conservative flux and $s$ is a generic source term. Notice that the PDE is defined in the continuous sense on a manifold $\Gamma$.

Before proceeding with the details of the discretization, some notation is introduced. Let $\Gamma_h$ denote a collection of disjoint elements that approximate the manifold $\Gamma$, of boundary $\partial\Gamma$, and let $\partial\Gamma_h$ denote the set of boundaries of all the elements: $\partial\Gamma_h = \{\partial K^\Gamma : K^\Gamma \in \Gamma_h\}$. Here, each element $K^\Gamma \in \Gamma_h$ is defined through an isoparametric mapping from the parameter space $\boldsymbol{\xi} \in \mathbb{R}^n$ to the physical space $\mathbf{x}(\boldsymbol{\xi}) \in \mathbb{R}^{n+1}$, where $d$ is the dimension of the manifold (e.g. $n = 1$ in 2-D flows). Similarly, let $\boldsymbol{\mathcal{E}}_h$ denote the set of all element boundaries counted only once.

These geometrical entities serve as support for the following function spaces:

$$\mathcal{U}_h^p = \{u \in L^2(\Gamma_h) : u|_{K^\Gamma} \in \mathcal{P}^p(K^\Gamma), \forall K^\Gamma \in \Gamma_h\}, \tag{5.19}$$

$$\mathcal{V}_h^p = \{v \in L^2(\Gamma_h) : v|_{K^\Gamma} \in \mathcal{P}^p(K^\Gamma), \forall K^\Gamma \in \Gamma_h\}, \tag{5.20}$$

$$\mathcal{M}_h^p = \{\mu \in L^2(\boldsymbol{\mathcal{E}}_h) : \mu|_F \in \mathcal{P}^p(F), \forall F \in \boldsymbol{\mathcal{E}}_h\}. \tag{5.21}$$

For 2-D flows, the boundaries (or edges) of the elements degenerate to a point in space,

so that the last function space can be written as:

$$\mathcal{M}_h = \{\mu \in \mathbb{R}(\boldsymbol{\mathcal{E}}_h) : \mu|_F \in \mathbb{R}, \forall F \in \boldsymbol{\mathcal{E}}_h\}. \tag{5.22}$$

The differentiability of the mapping $\mathbf{x}(\boldsymbol{\xi})$ enables the extension of the usual rules of vector calculus onto each element. However, this requires the shorthand used in differential geometry to account for the geometry of the manifold. In particular, let:

$$\mathbf{e}_i = \frac{\partial \mathbf{x}}{\partial \xi_i}, \tag{5.23}$$

denote the tangent (or covariant) basis on the manifold, that can be used to generate the metric (covariant) tensor and its inverse (contravariant) by means of the expression

$$g_{ij} = \frac{\partial \mathbf{x}}{\partial \xi_i} \cdot \frac{\partial \mathbf{x}}{\partial \xi_j}, \quad g^{ij} = (g_{ij})^{-1}, \quad g = \det(g_{ij}). \tag{5.24}$$

Here, $\cdot$ denotes the Euclidean inner product in $\mathbb{R}^{n+1}$. The existence of a metric induces the associated dual (or contravariant basis) $\mathbf{e}^j$ given by $\mathbf{e}^i = g^{ij}\mathbf{e}_j$. Here, and for the rest of this section, Einstein's notation is implied. In this way, any vector $\mathbf{v}$ can be expressed in either basis for the tangent space by means of the following identities:

$$\mathbf{v} = v^i \mathbf{e}_i = v_i \mathbf{e}^i, \qquad v^j = g^{ij} v_i, \qquad v_i = \mathbf{v} \cdot \mathbf{e}_i, \tag{5.25}$$

where $v_i$ and $v^i$ denotes the covariant and contravariant components of $\mathbf{v}$, respectively. In addition, the metric plays a key role in the definition of the inner product between two vectors $\mathbf{v}$ and $\mathbf{w}$ on the surface. Namely:

$$\mathbf{v} \cdot \mathbf{w} = v_i w^i = v^i w_i = v_i g^{ij} w_j = v^i g_{ij} w^j. \tag{5.26}$$

Based on this, a variety of differential operators on the surface can be defined. Of particular interest here are the divergence of a vector $\mathbf{v}$ along the surface, that can be

expressed as:

$$\nabla_{K^\Gamma} \cdot \mathbf{v} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi_i} (\sqrt{g} v^i), \tag{5.27}$$

or, the covariant components of the gradient of a scalar field along the surface, that are given by:

$$(\nabla f)_i = \frac{\partial f}{\partial \boldsymbol{\xi}}. \tag{5.28}$$

In addition, the following inner products need to be defined:

$$(u, v)_{\Gamma_h} := \sum_{K^\Gamma \in \Gamma_h} \int_{K^\Gamma} uv, \qquad \langle u, v \rangle_{\partial K^\Gamma} := \sum_{K^\Gamma \in \Gamma_h} \int_{\partial K^\Gamma} uv, \tag{5.29}$$

where $\int_{K^\Gamma}$ or $\int_{\partial K^\Gamma}$ are computed in the reference space with the help of the determinant of the metric $g$.

With all these definitions at hand, the SHDG discretization follows the same three steps described in Chapter 2 for the standard HDG method, namely: 1) integration by parts of the governing equation in every element, 2) introduction of a numerical flux at the interface that depends on the trace unknown (and the solution at each side), and, 3) an integral statement across each interface to ensure conservation of fluxes.

In an algebraic sense, the discrete weak form for the SHDG scheme is composed of two sets of equations that are related to conservation statements at the element level and across interfaces. These read: find $(u_h, \hat{u}_h) \in (\mathcal{U}_h^p, \mathcal{M}_h^p)$ such that

$$\left( \frac{\partial u_h}{\partial t}, v \right)_{\Gamma_h} - \left( f^i, \nabla v_i \right)_{\Gamma_h} + \langle \hat{f}^i n_i, v \rangle_{\partial \Gamma_h} - (s, v)_{\Gamma_h} = 0, \qquad \forall v \in \mathcal{V}_h^p, \tag{5.30}$$

$$\langle \hat{f}^i n_i, \mu \rangle_{\partial \Gamma_h \setminus \partial \Gamma} + \langle f_{\text{bou}}^\Gamma(u_h, \hat{u}_h), \mu \rangle_{\partial \Gamma} = 0, \qquad \forall \mu \in \mathcal{M}_h^p, \tag{5.31}$$

where:

$$\hat{f}^i n_i = f^i(\hat{u}_h) n_i + \tau(u_h - \hat{u}_h). \tag{5.32}$$

In this notation, $f^{\Gamma}_{\text{bou}}(u_h, \hat{u}_h)$ encodes the boundary conditions of the PDE. Also, $\tau$ is the usual stabilization constant in the HDG scheme, that in the case of a hyperbolic equation is commonly set to a fraction of the convection velocity.

As written, the system formed by Eq. 5.30-5.31 has a special algebraic structure by which the degrees of freedom for $u_h$ in a given element $K^{\Gamma} \in \Gamma_h$, only depend on themselves and on the value of $\hat{u}_h$ on the boundary $\partial K^{\Gamma}$. This is common to all HDG discretizations [167] and has been used at the assembly level to reduce the complexity of the problem.

In the particular case of the model for the evolution of $\tilde{n}$ (Eq. 5.17), the SHDG discretization follows naturally from Eq. 5.30-5.31 up to the definition of $\tau$ and $f^{\Gamma}_{\text{bou}}(u_h, \hat{u}_h)$. For all the result presented in this work:

$$\tau = \widetilde{|\mathbf{v}^{\text{e}}_{\Gamma}|}, \tag{5.33}$$

and,

$$f^{\Gamma}_{\text{bou}}(u_h, \hat{u}_h) = (v^{\text{e}\,i}_{\Gamma} n_i - |v^{\text{e}\,i}_{\Gamma} n_i|)(u_h - \hat{u}_h) + (v^{\text{e}\,i}_{\Gamma} n_i + |v^{\text{e}\,i}_{\Gamma} n_i|)(\hat{u}_h). \tag{5.34}$$

Notice that the model can accommodate any approximation order, provided the basis functions for the discrete spaces $\mathcal{U}^p_h$, $\mathcal{V}^p_h$ and $\mathcal{M}^p_h$ are properly defined. However, following the reasoning that led to the development of the scheme in the first place, the order of approximation is matched to the one used in the flow solver.

### 5.2.3 Coupling with the Spalart-Allmaras model

The use of SHDG to discretize Eq. 5.17 yields a high order approximation to the amplification factor $\tilde{n}$ that can be used to flag a given point along the boundary layer as either laminar or turbulent. This information needs to be propagated into the turbulence model so that the different regions of the boundary layer are treated accordingly.

In the end, the previous problem boils down to that of preventing the growth of the turbulent variables ($k$, $\omega$, $\tilde{\nu}$, etc.) in the laminar regions of the flow. In the case of the SA model, this could be achieved in two ways, either through the original provision based on the $f_{t1}$ and $f_{t2}$ terms [223, 52], or through the use of an intermittency factor $\gamma$ in front of the source terms [151, 51]. Each one of these strategies has their own advantages and disadvantages.

In particular, the use of the original $f_{t2}$ term introduces a nonlinearity in the model that makes the laminar solution stable, without the need for an extra equation (except for $\tilde{n}$ or an equivalent variable). While this saves computational cost, it also makes the model very sensitive to the way the system is taken out of this equilibrium. This is especially true for high Reynolds number flows where the growth of eddy viscosity at the transition location is very aggressive and might induce oscillations in $\tilde{\nu}$ that effectively make the transition point creep upstream.

This instability is not present when the intermittency factor $\gamma$ is used instead, since the production of eddy viscosity is decoupled from the eddy viscosity itself. This certainly simplifies the process of converging to a solution, but requires a proper definition of $\gamma$ in the whole fluid domain. This can be done in a variety of ways depending on the mesh and the complexity of the geometry. In particular, for geometries that can be meshed using a structured grid, $\gamma$ can be written as an algebraic function of $\tilde{n}$ that is then propagated away from the body along the corresponding grid lines. This approach was used successfully in most of the early examples of transition prediction in RANS solvers [197, 131, 229]. Unfortunately, this simple procedure cannot be applied to more complex cases that use multi-block or unstructured grids. For these cases, the classical solution has been to use a PDE to turn $\gamma$ into an unknown on the whole domain, as originally proposed by Langtry and Menter [133] and recently adopted by Medida and Baeder [151], and, Coder and Maughmer [51]. Obviously, this incurs a penalty in terms of cost and also requires a careful design of the PDE that governs $\gamma$.

In this work, the transition to turbulence is handled through an algebraic relation for $\gamma$ as a function of $\tilde{n}$. In this case, the choice of intermittency over the $f_{t2}$ terms is

based on the appearance of the aforementioned instability, that only worsens as the order of approximation is increased. Once this is set, the particular choice of model for $\gamma$ is not that important since both the PDE and algebraic models could be easily implemented within the framework of the $r$-adaptive solver. Nevertheless, the use of an algebraic model has been favored in this thesis due to its simplicity and reduced cost.

The starting point of the description of $\gamma$ is a piecewise linear scalar field $\gamma_0$ over the discrete mesh $\Gamma_h$, that will serve to tag points of the mesh where the flow is forced to turn turbulent, such as trailing edges, kinks in the geometry, etc. For these, $\gamma_0 = 1$. In the rest of the surface, however, $\gamma_0 = 0$. The value of $\gamma_0$ is pre-computed *a priori* and remains constant throughout the iteration.

The second component is a function $\gamma_n$ that modulates the intermittency across the boundary layer domain. Here, $\gamma_n$ is defined as a function of the element in the stack $i$ and the reference coordinate $\eta$ in the following way:

$$\gamma_n(i,\eta) = \max\left(0, \frac{\left(\sum_{j=1}^{i-1} h_j + h_i\eta\right) - c_0}{1 - c_0}\right), \tag{5.35}$$

where $c_0$ is a cutoff value that controls the extent of the $\gamma_n = 0$ region. For all the results presented in this chapter: $c_0 = 0.8$. As in the case of $\gamma_0$, the distribution of $\gamma_n$ is precomputed and kept constant throughout the solution process.

The last relevant term is a relationship between the intermittency and $\tilde{n}$ that acts as a smooth surrogate for Equation 5.14. In this work, this is defined as:

$$\gamma_{\tilde{n}} = \frac{\tanh\left(\frac{\tilde{n} - N_{\text{crit}}}{\Delta\tilde{n}}\right) + 1}{2}. \tag{5.36}$$

Here, the parameter $\Delta\tilde{n}$ controls the width of the ramp, and hence is a suitable variable to mimic the evolution of secondary instabilities in certain flows [199]. In this work, the focus is not on modeling the physics to that detail, and a constant value $\Delta\tilde{n} = 0.01$ is used instead.

With all these at hand, the intermittency in the whole domain is defined as a scalar field by means of the following expression:

$$\gamma = \begin{cases} 1, \text{ in the external domain,} \\ 1 - (1-\gamma_n)(1-\gamma_0)(1-\gamma_{\tilde{n}}), \text{ in the boundary layer domain,} \end{cases} \tag{5.37}$$

that enters the source term of the SA model as described in Chapter 2. This particular definition of $\gamma$ guarantees that the intermittency is zero in the viscous region when both $\tilde{n} < N_{\text{crit}}$ and $\gamma_0 = 0$, thus suppressing the growth of eddy viscosity there. Notice that this is not the case in the external domain where $\gamma = 1$ and the eddy viscosity can grow driven by shear. This was done on purpose to recover the behavior of the SA model in the wake. A sketch of this is found in Figure 5-1.

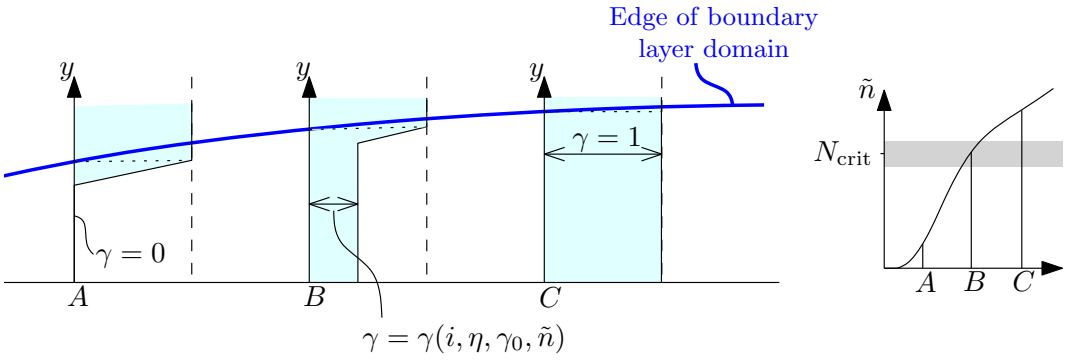



Figure 5-1: Sketch of the evolution of the intermittency before (A), during (B) and after (C) transition. For the external domain, beyond the boundary layer edge, $\gamma = 1$ independently of $\tilde{n}$.

## 5.3 Solution procedure

The extension of the $r$-adaptive solver for transitional flows consists of:

- an extra set of equations associated to the SHDG discretization (Eq. 5.30-5.31) of the governing equation for $\tilde{n}$ (Eq. 5.17),
- the extra unknowns $\tilde{n}_h$ and $\hat{\tilde{n}}_h$ associated with them, and,

- the proper ways to compute $\gamma$, $\delta_{\mathrm{k}}^*$, $\theta_{\mathrm{k}}$, etc. as well as its derivatives.

These quantities, combined with the rest of the equations and unknowns of the system (see Section 4.2.5) are enough to define the laminar/turbulent regions of the domain, the flow solution as well as the mesh. However, the inclusion of the transition module in the solver induces a few extra couplings with respect to the ones described in Figure 4-7. These are denoted by solid arrows in the updated diagram contained in Figure 5-2.

The solution of the resulting system of nonlinear equations follows the exact same strategy as the $r$-adaptive solver. A careful description of the details of the procedure can be found in Section 4.2.6. In addition, the variables associated with the amplification factor need to be initialized. For all the results presented here, $\tilde{n} = 0$ and $\hat{\tilde{n}} = 0$ at $t = 0$.

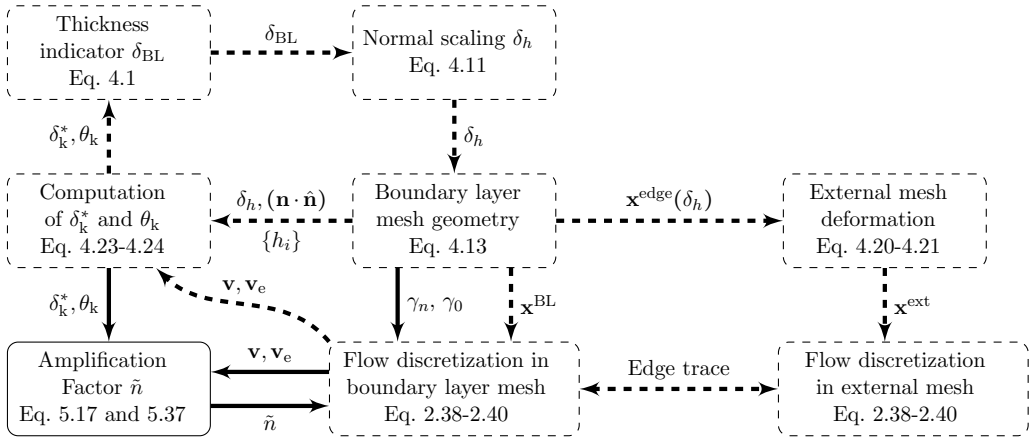



Figure 5-2: Flow chart of the solver showing the different modules (boxes) as well as the variables that are transferred between them (arrows). Dashed arrows or boxes represent the original $r$-adaptive solver from Chapter 4 while solid arrows and boxes denote the ones introduced by the transition prediction module.

## 5.4 Results

In this section, the $r$-adaptive solver augmented with the transition prediction module is tested on three different cases. The focus here will be on the behavior of the solver in a variety of flow conditions (e.g. attached flow transition, separation induced transition and unsteady transition) comparing to experimental data as well as Xfoil when possible. These comparisons should be taken with caution given the strong dependence of the results on the choice of transition database and the parameters that govern the model.

### 5.4.1 Natural transition over a NACA 0012 airfoil

The study begins with the simulation of the flow around a NACA 0012 airfoil at $\alpha = 6°$, $Re_c = 6 \cdot 10^6$ and $M_\infty = 0.15$, which, for this choice of parameters, produces natural transition on both the upper and lower surface. This particular mode of transition is very benign in terms of the coupling between the flow field and the amplification factor, which makes this test case ideal as an introductory example.

The simulation is carried out on the geometry depicted in Figure 5-3 using polynomials of order $p = 4$, $N_{\text{crit}} = 9$ and a series of mesh refinements. In particular, the meshes are constructed in such a way that the region around the transition point is uniformly refined between iterations, while keeping the element count across the boundary layer domain fixed to $n_{\text{norm}} = 10$. An example of the initial mesh and the mesh after 4 iterations of refinement is shown in Figure 5-4. Also, a sample of the Mach number field and well as the eddy viscosity field computed on the finest mesh is found in Figure 5-5.

In addition to this, the solver also returns the amplification factor as well as the normal scaling indicator (or equivalently, $y_1^+$). Both these quantities are plotted in Figure 5-6 for some of the refinement steps. Notice the disparity in the location of transition between the upper surface and the lower surface, which explains the low levels of the eddy viscosity on the lower surface as shown in Figure 5-5b. This effect

is also visible in $y_1^+$, that evolves at two different rates before and after transition happens.

To assess the grid convergence of the solver, attention is turned to the pressure and friction coefficient obtained as the mesh is refined. These are plotted in Figure 5-7 for the whole airfoil (top row) as well as a detail around the transition point on the upper surface (bottom row). The result indicate that grid convergence with respect to the transition module is observed past a certain level of refinement, that will depend in general on the aggressiveness of the transition parameter $\Delta\tilde{n}$. The choice of a constant value for $\Delta\tilde{n}$ is justified for high Reynolds number flows as transition is extremely abrupt in those cases, however, it might require some extra modeling for lower values of the Reynolds number. For reference purposes, the solution is compared to the one returned by Xfoil, which for this flow condition can be considered very close to the true solution. In general terms, both solvers agree on the transition location as well as the pressure and friction coefficient over the airfoil.

In summary, this test case serves to verify the proposed extension of the $r$-adaptive solver that, in addition to the mesh and the flow field, can produce an accurate estimate of the amplification factor to drive the location of transition to turbulence.

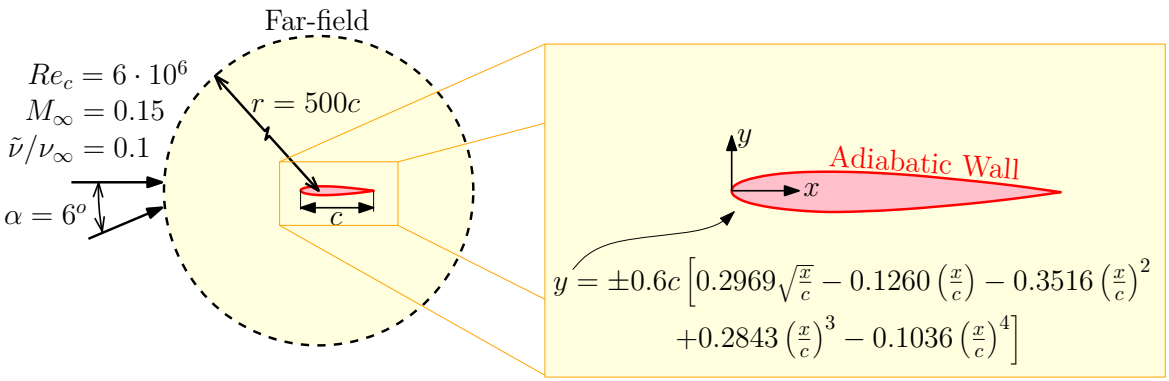
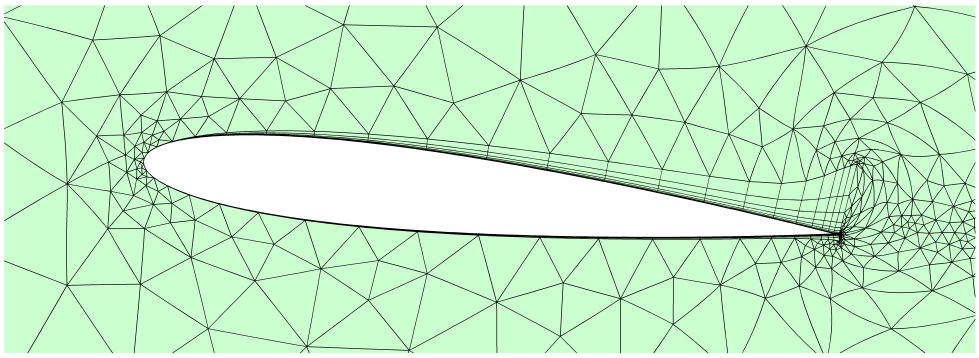
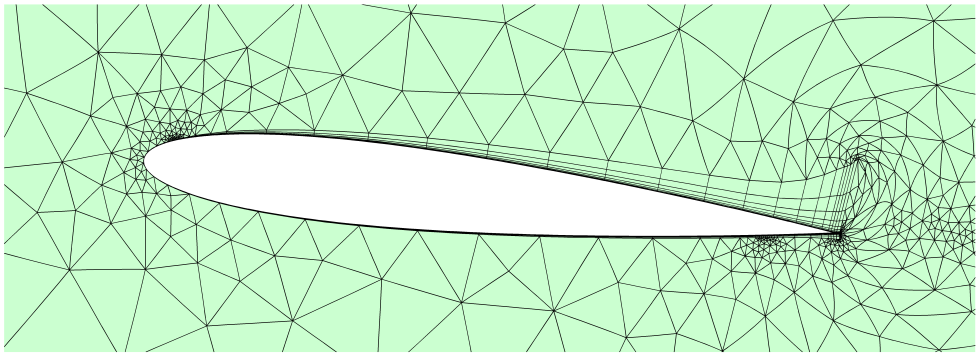


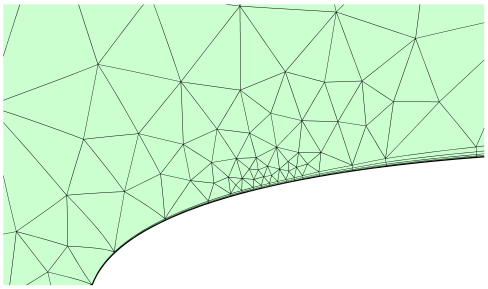

Figure 5-3: Sketch of the geometry and boundary conditions for the case of a transitional flow over a NACA 0012 airfoil.



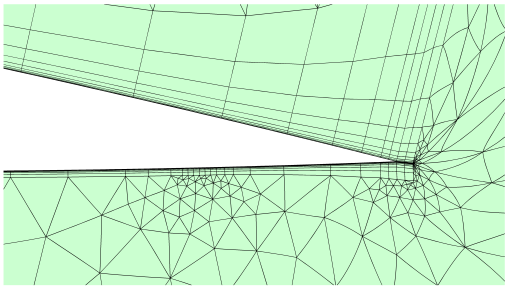
(a) Initial mesh



(b) Final mesh after 4 iterations of refinement around the transition point

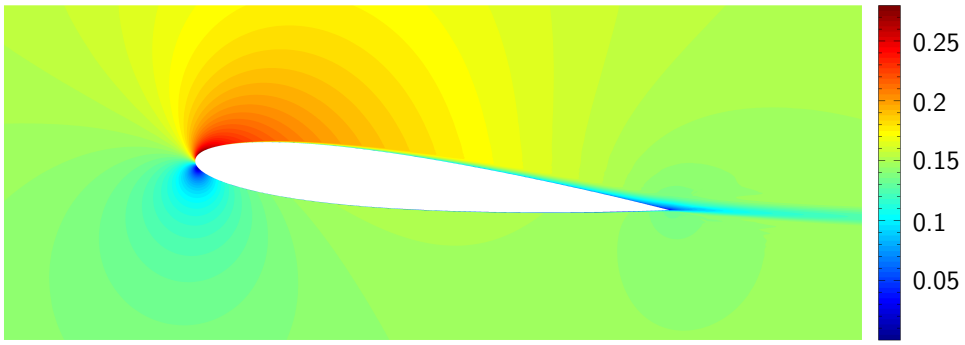


(c) Detail around the upper surface transition point

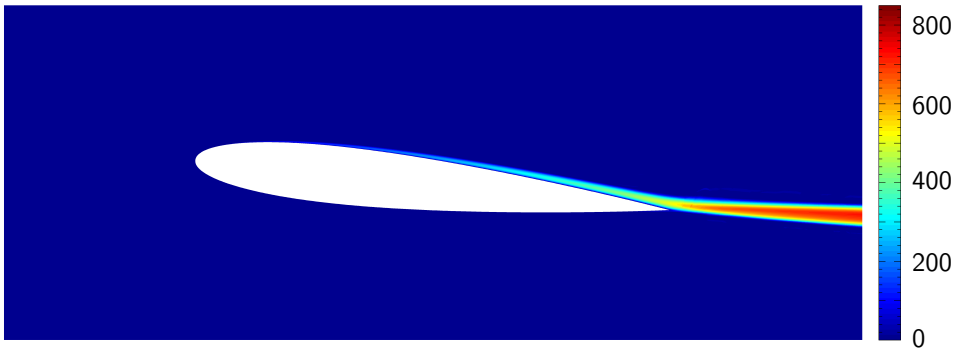


(d) Detail around the lower surface transition point

Figure 5-4: Sequence of meshes generated for the simulation of the transitional flow over a NACA 0012 airfoil, showing the refinement mechanism.

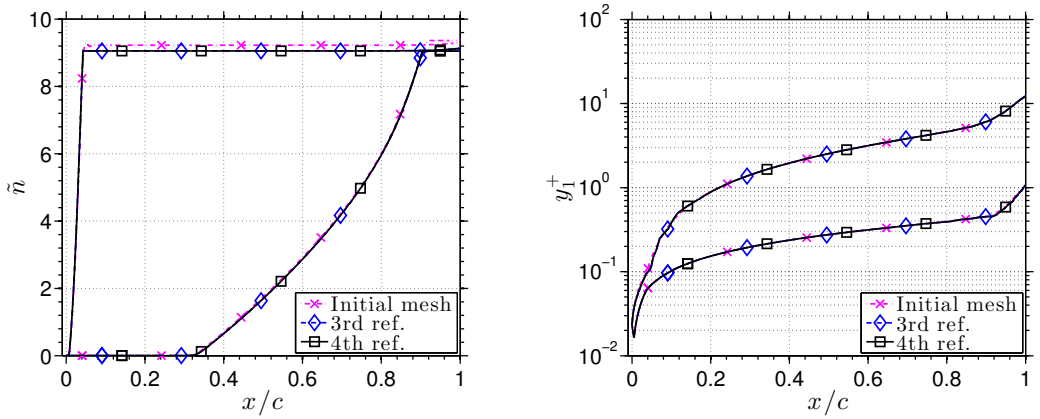



(a) Mach number field



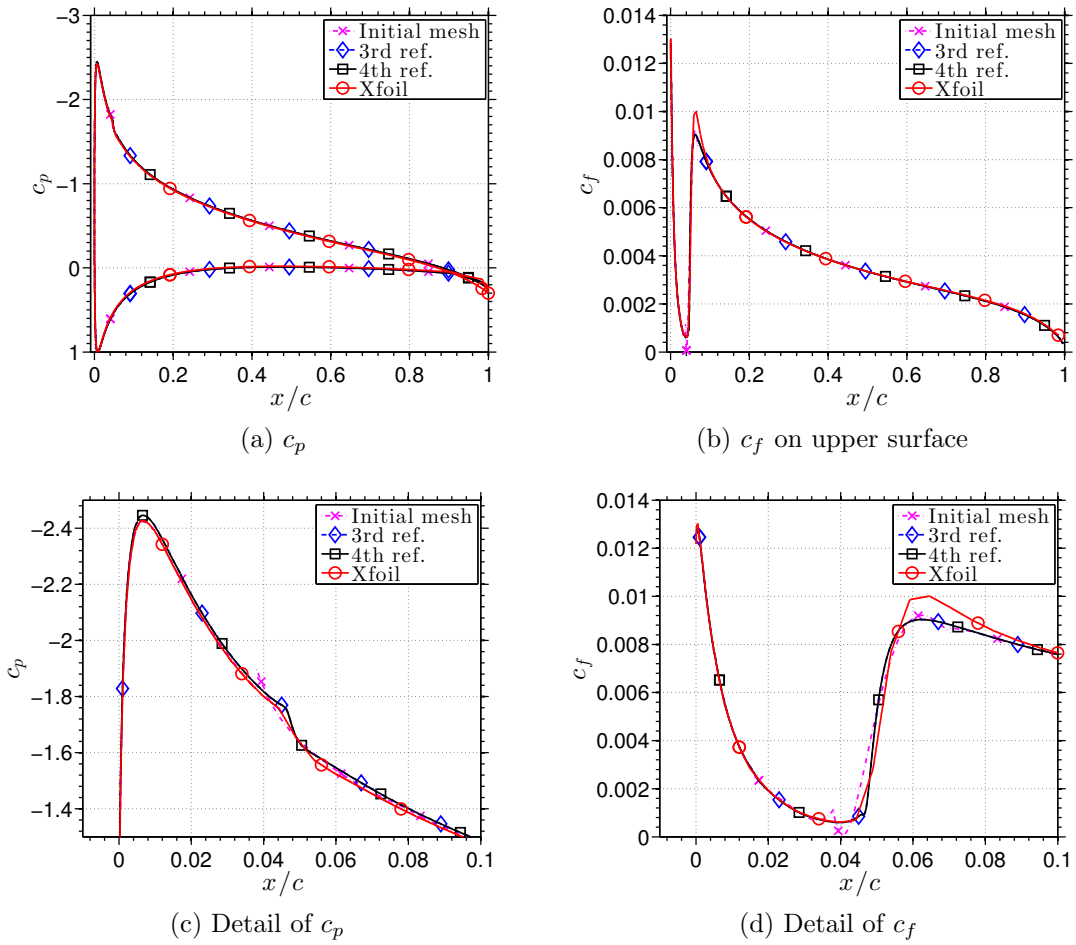

(b) $\mu_t/\mu_\infty$ field

Figure 5-5: Rendering of the solution on the finest mesh for the case of the transitional flow over a NACA 0012 at $M_\infty = 0.15$, $Re_c = 6 \cdot 10^6$ and $\alpha = 6°$. Notice the suppression of the eddy viscosity on the lower surface due to the late transition.




(a) $\tilde{n}$ (b) $y_1^+$

Figure 5-6: Evolution of the amplification factor $\tilde{n}$ and distance of the first node off the wall $y_1^+$ along the chord. The location of transition strongly correlates with a change in the growth rate of the boundary layer.




(a) $c_p$

(b) $c_f$ on upper surface

(c) Detail of $c_p$

(d) Detail of $c_f$

Figure 5-7: Comparison of the pressure and friction coefficient as the mesh is refined. The detail around the transition point on the upper surface (lower row) indicates that grid convergence is attained if the mesh is properly refined around the transition region.

### 5.4.2 Separation induced transition over an Eppler 387 airfoil

The next section considers the computation of transition in a laminar separation bubble, which is the typical transition mode in moderate to low Reynolds number flows. In particular, this mode represents a challenge for any solver due to the strong interaction between the amplification factor and the location of transition, that makes most staggered iterations fail, and has forced other authors to totally ignore this mode by prescribing transition at separation [197, 226, 126, 127]. The goal of this section is to verify that a fully coupled solution in which all the variables of the problem are computed at the same time does not suffer from this problem, as already proved by Drela and Giles in the context of viscous-inviscid interaction solvers [69].

To this end, the flow around an Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$, $M_\infty = 0.08$ and various angles of attack is simulated using the $r$-adaptive solver with the transition prediction module. The geometry and boundary conditions for this test case are described in Figure 5-8. For all the results shown here, the solution is computed using polynomials of order $p = 4$ on the same initial mesh with $60 \times 10$ elements in the boundary layer domain.

The simulation consists of an angle of attack sweep from $\alpha = 0°$ to $\alpha = 9°$, focusing on the flow characteristics on the upper surface, that develop three clearly distinct flow regimes.

In the first one, which appears at low angles of attack, the flow presents a laminar separation bubble in the rear part of the airfoil that moves upstream as the incidence is increased. Said laminar separation bubble is rather long and can extend for up to 30% of the chord (from separation to reattachment). An example of this is found in Figure 5-9, where some results for the case of $\alpha = 4°$ are compiled. In there, the laminar separation bubble is clearly visible in the velocity profiles as well as in the pressure signature. The later compares reasonably well with experiments as well as Xfoil. In this case, the reattachment is triggered by transition to turbulence, as the eddy viscosity field indicates. This last phenomenon is also visible in the velocity profiles.

In the second regime, that extends a very narrow window of $\alpha$, transition occurs before separation, which re-energizes the boundary layer and prevents separation altogether. An example of this is found in the plots contained in Figure 5-10 for the case of $\alpha = 7.5°$. This phenomenon is hard to identify in a rendering of the flow field, however, it is directly visible in the velocity profiles as well as in the pressure coefficient, where there is no trace of the plateau associated with the separation bubble. Again, the comparison with experiments and Xfoil is fairly positive.

The last of the three regimes occurs beyond a certain angle of attack and is characterized by the presence of a laminar separation bubble starting at the leading edge that transitions and reattaches in a very short distance of the order of a few percent of the chord. A sample of the results for the case of $\alpha = 9°$ is compiled in Figure 5-11. In these, the laminar separation bubble is visible in the velocity profiles around the leading edge, as well as the pressure signature. The agreement with the experiments is poor in the separation bubble, however, the comparison with Xfoil on the same discrete geometry is only marginally better (see Figure 5-11d). In any case, this discrepancy, that could be due to a variety of reasons, does not prevent the solver from reproducing the leading edge separation regime in a qualitative sense.

In order to assess the extension of each of the three aforementioned regimes, the solver was run on a finer grid in angle of attack, with special emphasis on the natural transition region. These runs are summarized on Figure 5-12, where the location of separation, transition and reattachment is plotted. The results indicate that the location of separation compares well with the experiments, however reattachment happens too late. This behavior worsens as the Reynolds number is reduced which questions not only the way transition is imposed but also the SA model itself. A possible explanation of the source of the problem can be found in Appendix B.

These same runs can be used to produce a polar for the airfoil. In particular, Figure 5-13 contains the results computed with the $r$-adaptive solver, the experimental data by McGhee *et al.* [150], the results from runs with Xfoil on the same discrete geometry, and the data published by Crivellini and D'Alessandro [53]. In global terms, the $r$-

adaptive solver can capture the three regions of the polar (separation bubble, natural transition, and leading edge separation) with a reasonable agreement with data and Xfoil, except for the leading edge separation regime, where the lift is over-predicted and the $r$-adaptive solver tends to produce results that are closer to the ones extracted from Crivellini and D'Alessandro [53].

The most important conclusion of this study is that the solver can simulate transition in a variety of separated flows, without experiencing any issues in terms of convergence. The sole enabler for this is the use of a simultaneous iteration in which the amplification factor and the flow field communicate in a closed loop. In addition, the results compare reasonably well with experiments and well proven numerical tools in the regime of interest.

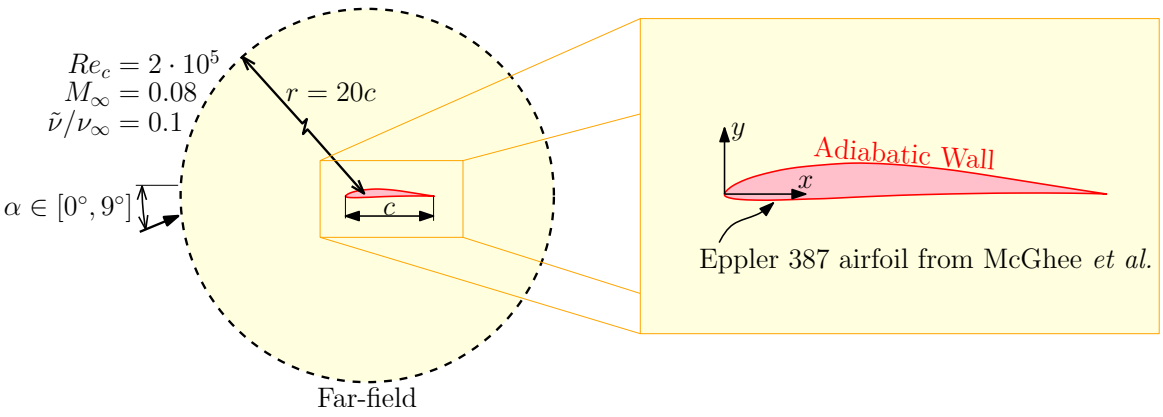
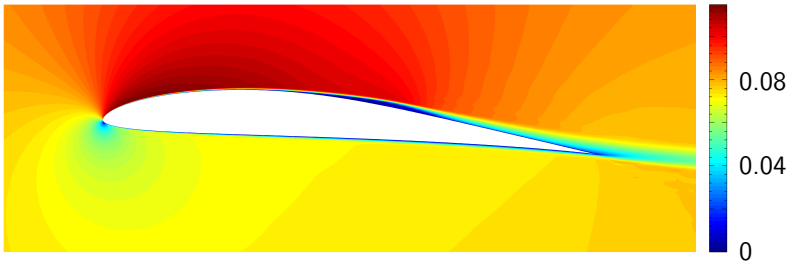
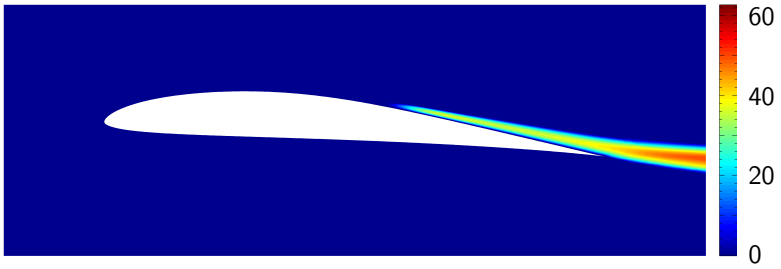


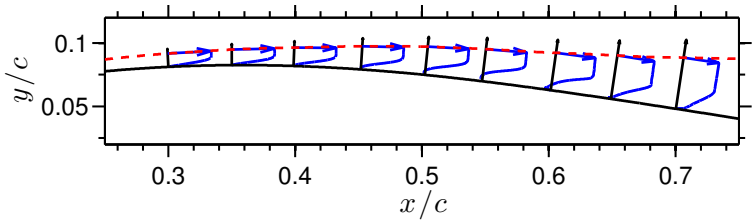

Figure 5-8: Sketch of the geometry and boundary conditions for the case of separation induced transition over an Eppler 387 airfoil.



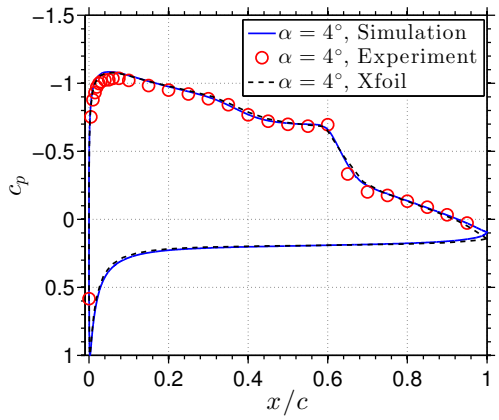

(a) Mach number field




(b) $\mu_t/\mu_\infty$ field

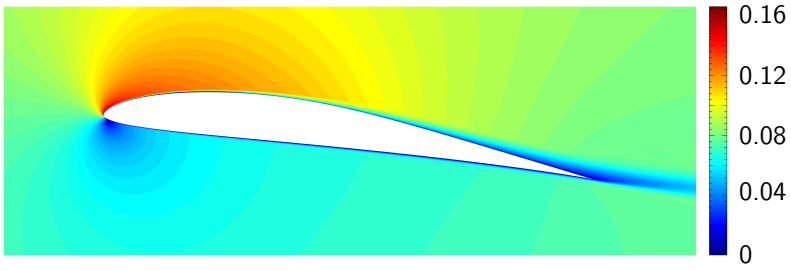



(c) Hodograph of the velocity around the transition point

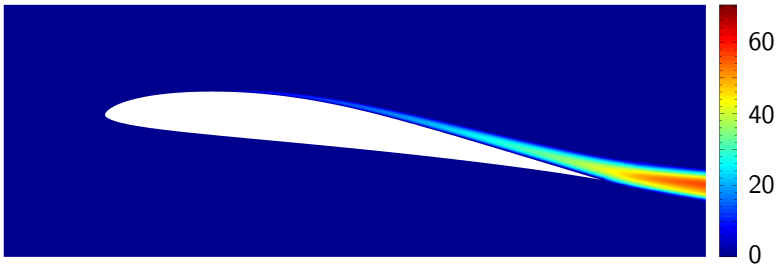



(d) Pressure coefficient

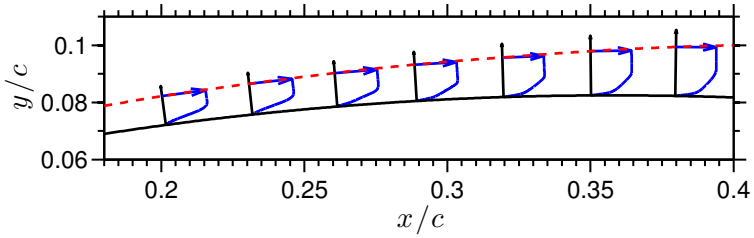
Figure 5-9: Summary of results obtained for the case of an Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$ and $\alpha = 4°$. Notice the separation bubble on the upper surface that is characteristic of this airfoil at low angle of attack.



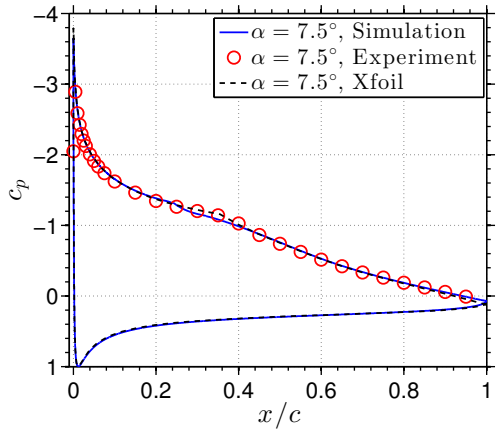

(a) Mach number field




(b) $\mu_t/\mu_\infty$ field

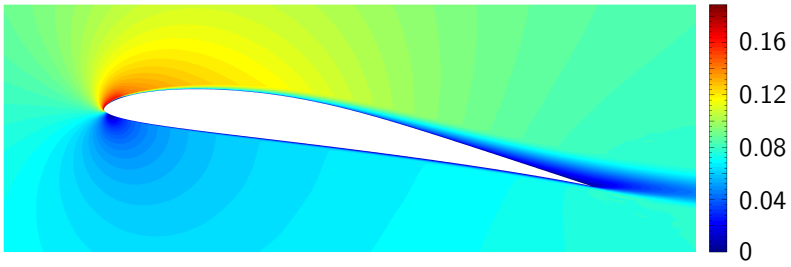



(c) Hodograph of the velocity around the transition point

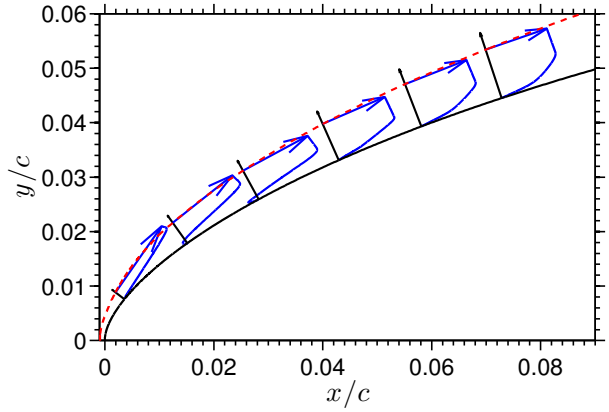



(d) Pressure coefficient

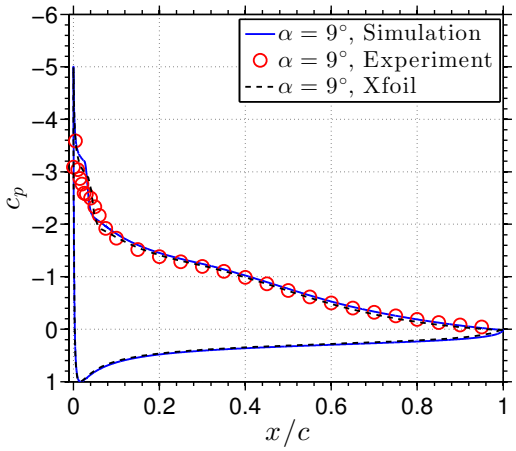
Figure 5-10: Summary of results obtained for the case of an Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$ and $\alpha = 7.5°$. In this case, transition happens before the flow has a chance to separate.



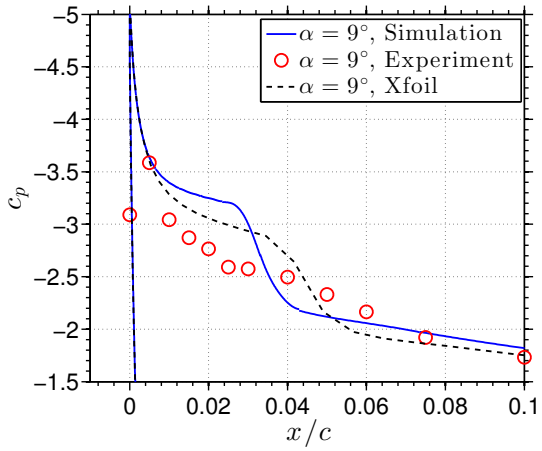

(a) Mach number field




(b) Hodograph of the velocity around the transition point




(c) Pressure coefficient




(d) Pressure coefficient around the leading edge

Figure 5-11: Summary of results obtained for the case of an Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$ and $\alpha = 9°$. At this incidence angle, the flow separates at the leading edge and quickly transitions to turbulence and reattaches. This phenomenon is hard to capture in experiments as well as numerically hence the discrepancy in the pressure coefficient.

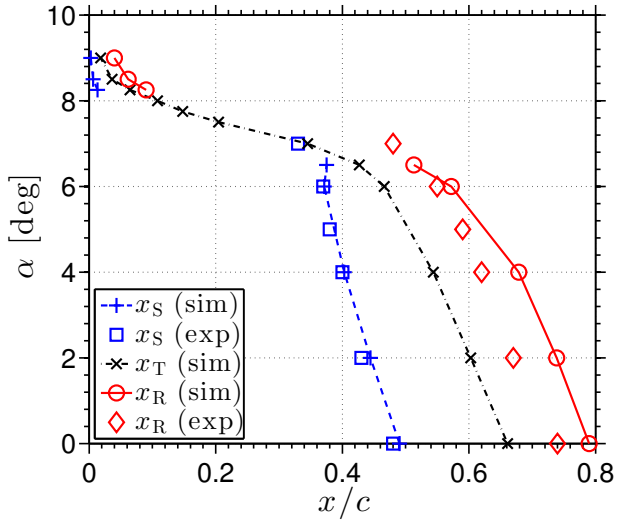



Figure 5-12: Computed locus of separation, transition and reattachment as a function of the angle of attack for the Eppler 387 at $Re_c = 2 \cdot 10^5$. The agreement with experiments is excellent for the separation point, but presents a delay in the computed reattachment.

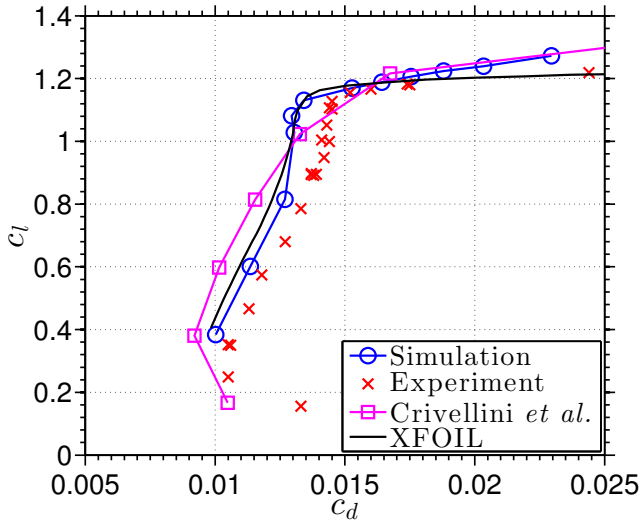



Figure 5-13: Computed $c_d$ vs. $c_l$ for the case of the Eppler 387 airfoil at $Re_c = 2 \cdot 10^5$, compared to experiments and other numerical tools. The three regimes on the airfoil (separation bubble, natural transition and leading edge separation) are visible on the computed polar.

### 5.4.3 Unsteady transition over a NACA 0012 airfoil

In the last test case, the method is applied to an unsteady flow using a simple ad-hoc modification to the steady state criterion (see Eq. 5.17). While this model is not physically sound, it will serve to prove the concept for quasi-steady flows and show the appearance of nonlinear effects as the unsteadiness is increased. Furthermore, this study proves that the solver can accommodate general transition models, as long as these can be written as a system of surface PDEs.

For this, the flow around a NACA 0012 airfoil, that oscillates in pitch around its quarter chord according to:

$$\theta = \alpha = 3.75^{\circ}(1 + \cos(\omega t)), \tag{5.38}$$

is simulated. In this particular case, $M_\infty = 0.1$ and $Re_c = 2.5 \cdot 10^5$, which ensures the presence of separation induced transition, as well as natural transition in some instances. As usual in unsteady aerodynamics, the angular velocity $\omega$ is expressed in terms of the reduced frequency $\bar{k} = (\omega c)/(2u_\infty)$.

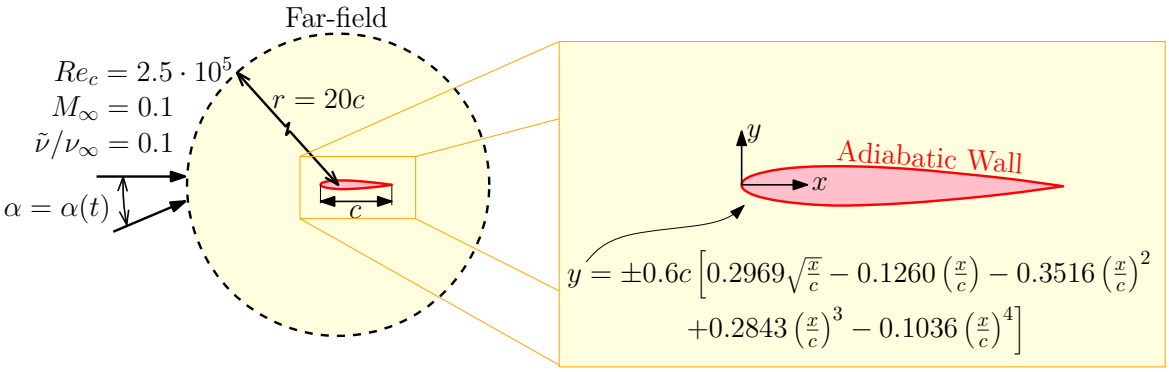



Figure 5-14: Sketch of the geometry and boundary conditions for the case of the unsteady transitional flow over a NACA 0012 airfoil oscillating in pitch.

For this problem, the solver needs to be adapted to accommodate the moving geometry. One possible way to do this is to use an ALE formulation that takes into

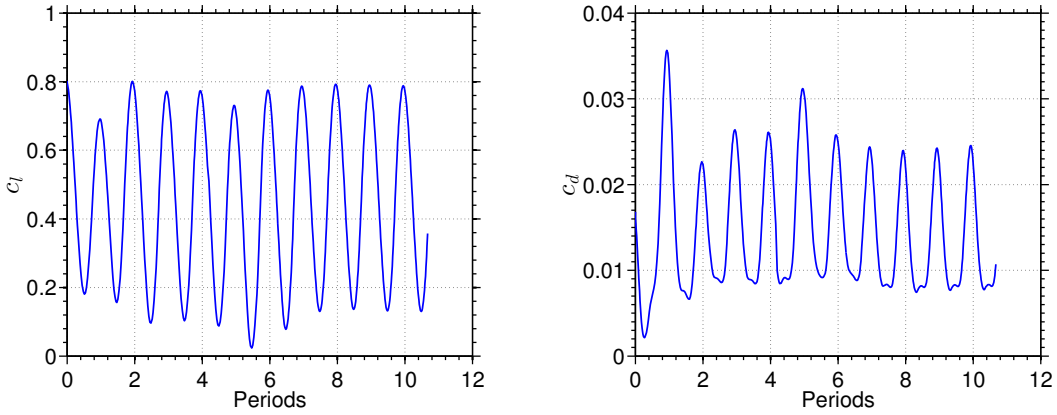
account the velocity of the airfoil. However, that requires some extra coding effort to make sure that the boundary layer profiles are defined with respect to the wall velocity, and that the surface PDEs are transformed accordingly. Given the preliminary nature of these results, this was not deemed adequate.

A simpler approach for the case of rigid solid motion is to assume that the reference frame is non inertial. In that case, the formulation is the same up to the appearance of fictitious volume forces (Coriolis, centripetal, etc.) on the right hand side of the RANS equations [141] and possibly in the turbulence model too. The later however, is not required in 2-D according to the theoretical results by Speziale [224], and, in any case, would have a very small effect for the low reduced frequencies used here.

The results presented next are computed using the non-inertial formulation on the $r$-adaptive solver using a similar numerical setting as in the previous cases. In particular, the solution is approximated in space using polynomials of order $p = 4$ and the boundary layer is resolved using $n_{\text{norm}} = 10$ elements across it. The time-stepping is based on the same BDF1 scheme used for the steady state simulations, however, with a limit on the upper value of $\Delta t$ to ensure a minimum of 60 time steps per oscillation cycle. This particular combination of high order in space and low order in time limits the accuracy of the scheme, but this is not a concern given the exploratory nature of this case.

The focus here is on the flow characteristics when the solution settles in a periodic regime. This involves marching the solution in time until a periodic pattern is identified in outputs like the lift and drag, as shown in Figure 5-15. Repeating this process for a variety of frequencies allows for a qualitative assessment of the evolution of the flow characteristics with the period of the oscillation.

In particular, it is interesting to see how the load cycle evolves with the frequency, since, in principle, there should be a departure from the quasi-steady solution as the pitching becomes more aggressive. This is confirmed by the simulations for $\bar{k} = \{0.05, 0.1, 0.4\}$ where the solver predicts a departure from the steady state polar as the reduced frequency is increased. This is depicted in Figure 5-16. A simple integration




(a) Lift versus time (b) Drag versus time

Figure 5-15: Evolution of the forces on the airfoil with time for the case of $\bar{k}$ = 0.2, showing how the system relaxes towards a periodic solution after a certain number of pitching cycles.

in time of these load cycles yields higher average values of lift and lower average values of drag than the quasi-steady approach, as summarized in Table 5.1.

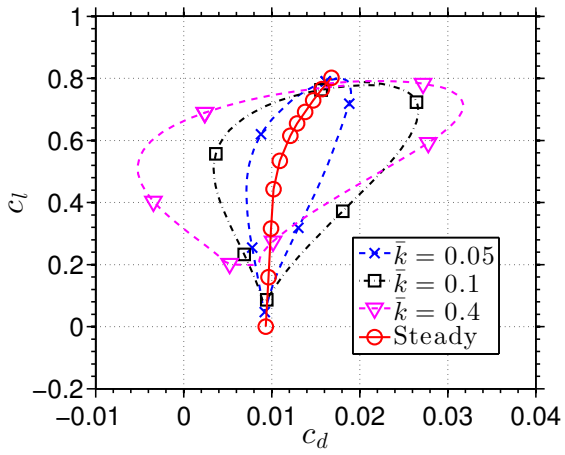



Figure 5-16: $c_d$ vs. $c_l$ as a function of the reduced frequency and compared to the steady state polar. As expected, the solution departs from the steady state case as the frequency of the oscillation is increased.

A similar hysteretic effect is also found in the evolution of the laminar separation bubble on the upper surface of the airfoil, that is totally suppressed during the down-stroke portion of the cycle if the reduced frequency is sufficiently high. This is visible on the maps of $c_f$ vs. time contained in Figure 5-17, where the low frequency solution

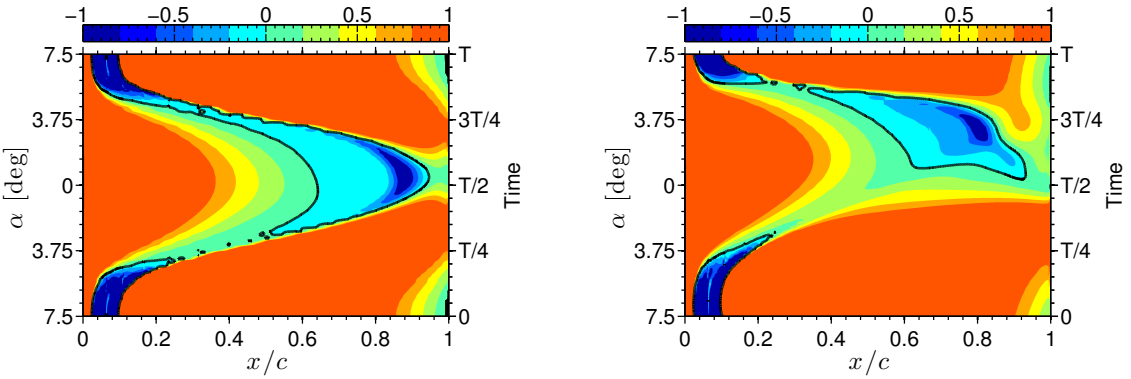
Table 5.1: Averaged force coefficients on the airfoil for different values of the reduced frequency. The quasi-steady result corresponds to an average over a cycle assuming that the lift and drag coefficients are taken from the steady state polar.

| | Quasi-steady | $\bar{k} = 0.05$ | $\bar{k} = 0.1$ | $\bar{k} = 0.4$ |
|---|---|---|---|---|
| Average $c_l$ | 0.4551 | 0.4593 | 0.4573 | 0.4929 |
| Average $c_d$ | 0.0121 | 0.0123 | 0.0134 | 0.0118 |

($\bar{k} = 0.05$, left) is compared to the high frequency one ($\bar{k} = 0.4$, right). This suppression of the separation bubble has a direct effect on the evolution of the amplification factor $\tilde{n}$ and produces a total relaminarization of the upper surface as the maps of $\tilde{n}$ vs. time in Figure 5-18 show.

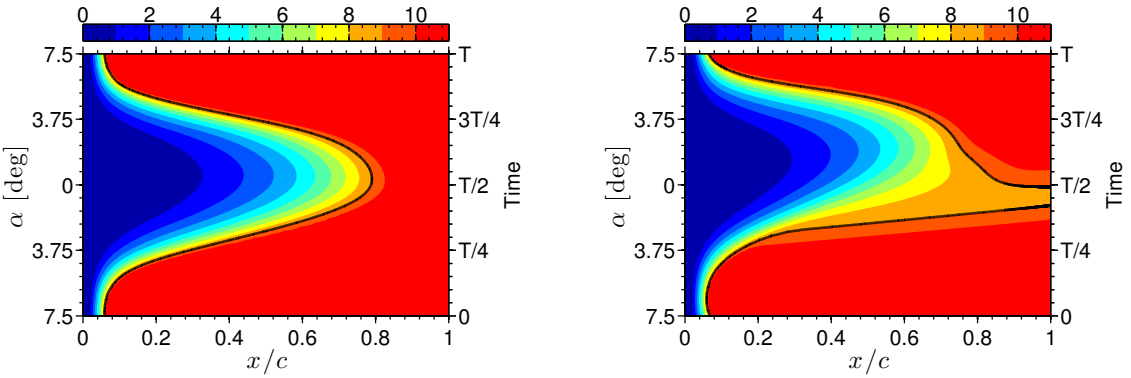
The effect of the frequency on the evolution of the mesh (or equivalently, the evolution of $\delta_h$) is more subtle and involves a small delay in time in addition to the effect of the relaminarization. This is depicted in Figure 5-19.

While these results would probably be wrong if compared to experiments, they still serve to illustrate a possible extension of the scheme into the unsteady transition regime, which is simple to incorporate in the current framework.



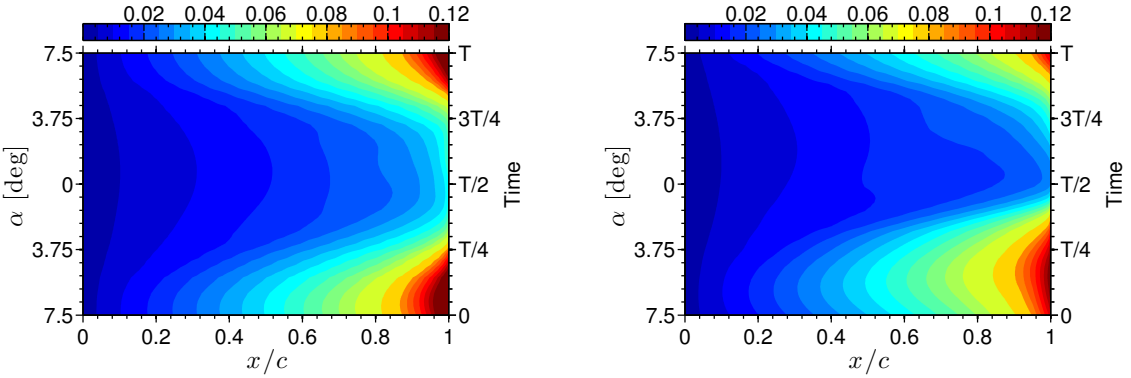

(a) $\bar{k} = 0.05$ (b) $\bar{k} = 0.4$

Figure 5-17: Comparison of the friction coefficient ($c_f \times 10^3$) along the upper surface of the airfoil over one period for two different reduced frequencies. The bottom half of each plot corresponds to the downstroke while the top half represents the upstroke. At increasing reduced frequencies, there is a suppression of the separation bubble during downstroke as the contours of zero friction (black lines) indicate.




(a) $\bar{k} = 0.05$ (b) $\bar{k} = 0.4$

Figure 5-18: Comparison of the amplification factor $\tilde{n}$ along the upper surface of the airfoil over one period for two different reduced frequencies. The bottom half of each plot corresponds to the downstroke while the top half represents the upstroke. For high reduced frequencies, the suppression of the separation bubble prevents transition to turbulence (black lines) over some portion of the downstroke.




(a) $\bar{k} = 0.05$ (b) $\bar{k} = 0.4$

Figure 5-19: Comparison of the evolution of the normal scaling $\delta_h$ along the upper surface of the airfoil over one period for two different reduced frequencies. The bottom half of each plot corresponds to the downstroke while the top half represents the upstroke. An increase in reduced frequency translates into a shift in time of the peak, in addition to the effect of the suppressed separation bubble.

## 5.5 Conclusions and future work

In this chapter, the $r$-adaptive method has been extended to account for transition to turbulence, using a surface discretization of the $e^N$ method.

The success of this strategy is based on two key ingredients. On the one hand,

there is the $r$-adaptive solver itself, that produces high quality boundary layer profiles as an intermediate step in the solution, which can be used to drive the $e^N$ method. On the other hand, there is a novel high order discretization for surface PDEs (SHDG) that enables the computation of the amplification factor in the whole surface without the need to identify stagnation points.

The solver has been applied successfully to cases with natural and separation induced transition. In the later case, the use of a simultaneous iteration allows the computation of transition past separation, followed by turbulent reattachment. This is a significant advantage over other RANS solvers that cannot simulate such flows due to instabilities in the coupling between the transition module and the flow discretization.

Furthermore, the solver has been extended to treat unsteady flows. For this, the equations have been solved using a non-inertial reference frame. A possible extension not considered here would be to use an inertial reference frame combined with the ALE formulation, which would eliminate the uncertainties associated to the correct modeling of the non-inertial terms in the turbulence model.

In addition, there is room for improvement in the transition prediction module by including other modes of transition (e.g. bypass [4]), more advanced databases for TS waves [68] and more sophisticated models for unsteady transition prediction. A similar situation arises in 3D which would require an extension of the database to include crossflow instabilities.

# Chapter 6

# Concluding Remarks

## 6.1 Summary and conclusions

This thesis presents a novel methodology to simulate viscous compressible flows with transition modeling based on a combination of adaptivity and a high order discretization. The unique feature of the proposed $r$-adaptive method is the way in which the position of the nodes of the mesh is cast into an explicit set of unknowns and solved together with the flow field. In this way, the mesh can be adapted to follow the boundary layer, which results in improved solution quality and robustness.

The success of this methodology relies heavily on the synergistic combination of a variety of techniques that are integrated in the solver. Of special importance is the use of hybrid meshes in which the boundary layer region is discretized using a structured mesh in the direction normal to the wall. This allows for boundary layer profiles to be easily extracted and to determine the thickness of the viscous layer which drives the mesh deformation algorithm. This way of treating the mesh as part of the solution enables the accurate simulation of separation bubbles as well as the simulation of high Reynolds number flows on highly anisotropic meshes.

The proposed algorithm performs adaptivity "on the fly" and hence can easily be combined with traditional a-posteriori adaptive schemes. This is particularly important in the case of compressible flows where the location of shock waves is not known

beforehand. In those cases, it is necessary to equip the solver with a shock capturing capability. In this thesis, a novel approach based on artificial viscosity has been proposed and demonstrated on a number of test cases for a wide range of Mach numbers, showing very sharp and clean shock profiles.

The $r$-adaptive scheme proposed here was purposely designed to accommodate transition prediction in addition to the boundary layer adaptivity. The capability of the solver to predict transition is based on the use of an envelope $e^N$ method discretized using a novel high order scheme for surface PDEs and coupled to the turbulent model using intermittency. One of the strengths of this approach is the monolithic or simultaneous coupling of all the governing equations (flow field, mesh and amplification factor) that shows its value in the case of separation induced transition, where this coupling is the key to maintaining robust convergence.

The development of the different modules that compose the solver has relied heavily on simple numerical exercises such as the 1-D study of the scaling of the shock indicator in Section 3.2, or the analysis of the stability of the adaptive process in Appendix A. Ultimately, the resulting algorithm has been validated through the successful simulation of a variety of 2-D flows.

Finally, note that the different contributions presented in this thesis are strongly related, in the sense that ones could not be materialized without the others. This is especially true in the case of the high order discretization, the $r$-adaptivity and the transition prediction, all of which are individual pacing items in the NASA CFD Vision 2030 Study [220]. This suggests that there is tremendous room for improvement (in terms of the goals of the CFD Vision 2030 Study) if some of the problems are tackled simultaneously, as attempted in this thesis, rather than individually.

## 6.2 Future work

Needless to say that the methodology presented in this thesis could be extended or enhanced in a variety of ways. A list of possible future research is presented as follows.

- **Enhancements on the shock capturing**

  The artificial viscosity model could be enhanced in a variety of ways. One of them would be to post-process the solution to avoid the pre-asymptotic regime in smooth flows, by reconstructing the element size field only in the elements where a shock is present. In addition, the element size field could be substituted by a metric field in order to make the method robust against a misalignment of the shock and the mesh.

- **Enhancements on the normal scaling equation**

  In the current stage of development, the whole adaptivity is driven by the normal scaling $\delta$, that controls the thickness of the boundary layer domain. For the range of Reynolds numbers tested in this work, this level of control is enough to ensure that the first node off the wall is placed within the viscous reference length ($y_1^+ = \mathcal{O}(1)$). However, this might not be the case at higher Reynolds numbers. A possible extension of this work would be to add another surface PDE to the system to measure the near wall scale and ensure that $y_1^+ = \mathcal{O}(1)$ by manipulating the spatial distribution of the nodes within the boundary layer $\{h_i\}$ point wise along the surface.

- **Loosely coupled iteration**

  In the current setting, the equations that govern the mesh, the flow field and the transition prediction are all solved together using a simultaneous iteration. While this makes the solver robust, it also increases the cost per iteration, hence the interest on staggered updates between the equations that can reduce the computational burden of the problem. Of particular interest here would be to stagger the mesh update from the rest of the unknowns, which are somewhat protected by the ALE formulation. If successful, this strategy would bring the cost per iteration close to that of the discretization of the fluid equations on a fixed mesh.

- **Use of advanced transition models**

  A simple modification of the current transition prediction module would be to use other databases for the $e^N$ method as well as some criterion to include the effects of bypass transition. Depending on the level of complexity of the model, these could be done in a trivial manner by changing the equations that govern the evolution of $\tilde{n}$. The proposed framework is very general and can propagate any number of scalar fields along the surface provided they only require data that can be extracted from the boundary layer profiles.

- **Application to complex geometries**

  In all the results presented in this thesis, the solver was exercised on relatively simple and smooth geometries. A possible extension of this would be to extend it to cases in which several boundary layer domains are present, such as for example a multi-element airfoil or internal flows with more than one viscous wall. While this is just a matter of implementation, it still requires a reasonable amount of effort to ensure that the solver is general enough to admit any number of boundary layer domains.

- **Extension to 3-D**

  The extension of the presented approach to 3-D flows would be the most important in terms of impact on the current state of the art. Unfortunately, this would also require work in several fronts. First of all, the key point in the solver presented here is the use of a hybrid mesh that is structured close to the wall. The generation of such meshes is a simple task in 2-D but not so much in 3-D, where it requires not only tools to grow the layers and mesh the external domain, but also intermediate pyramid elements to connect both. Secondly, the validity of the normal scaling indicator $\delta_{\mathrm{BL}}$ would need to be revisited to ensure that the effect of crossflow was taken into account. To continue, the solver would need to be implemented in a parallel environment, with the associated development of wrappers or custom made tools for mesh partitioning, distributed linear

solvers, preconditioners, visualization, etc. Finally, the transition prediction models should be extended to include crossflow transition as well as attachment line transition.

- **Adaptive Large Eddy Simulations**

  The whole idea of making the mesh move with the solution can also be applied in the context of implicit LES type simulations. In essence, this would require a suitable hybrid mesh and an average of the velocity profiles to compute $\delta_{\mathrm{BL}}$, dispensing with the transition prediction module. This strategy would be very attractive in combination with the loosely coupled iteration discussed above, as it would then incur no extra cost over that of solving for the flow alone. As an intermediate step, a similar strategy could also be applied in the hybrid RANS/LES context.

- **Other turbulence models**

  As the results in Appendix B show, the SA model in its standard form might not be suitable to model the flow in laminar separation bubbles at low Reynolds numbers. However, this could be circumvented if the model was modified appropriately. This task would require a recalibration of the model for separated flows. In any case, the choice of turbulence model is not unique since other models could be easily implemented within this framework.

# Appendix A

# Stability analysis of the normal scaling equation

The capabilities of the proposed solver are based on the synergies between its different modules, as summarized in Figure 4-7. Of special importance here is the feedback loop between $\delta_h$, the boundary layer mesh and the associated discretization, that represents the core of the adaptivity process. The goal of this Appendix is to study this loop using a simplified model to gain some intuition on the possible failure modes of the scheme, and how to avoid them.

## A.1 Simplified equation

The starting point of this study is Eq. 4.5, reproduced here:

$$\frac{\partial \delta}{\partial t} = \frac{k_\delta \delta_{\mathrm{BL}} - \delta}{\tau_\delta} + \mu_\delta \Delta_\Gamma \delta. \tag{A.1}$$

This time-dependent surface PDE can be classified as a nonlinear reaction-diffusion PDE, where the nonlinearity comes from the computation of $\delta_{\mathrm{BL}}$. More precisely, $\delta$ governs the location of the edge of the boundary layer domain, and hence the value of the integrals $\delta^*_{\mathrm{k}}$ and $\theta_{\mathrm{k}}$ according to the description in Section 4.2.4. In addition to this,

there exists a dependency of the velocity field $\mathbf{v}$ on time as well as on the boundary layer mesh, or equivalently $\delta$. The combination of these factors makes the theoretical analysis of the problem extremely hard, and motivates the use of a simplified version of Eq. 4.5 to gather partial evidence on the behavior of the indicator.

The simplified model is based on two assumptions. Firstly, that the diffusive operator in Eq. 4.5 has a minor effect on the final solution, hence can be safely ignored in the analysis. Secondly, that the time-scale of the flow is faster than that of $\delta$ hence the flow can be considered "frozen" in the computation of $\delta_{\mathrm{BL}}$. In this way, the surface reaction-diffusion PDE can be simplified into the following autonomous scalar ODE:

$$\frac{\partial \delta}{\partial t} = \frac{k_\delta \delta_{\mathrm{BL}}(\delta; \mathbf{u}) - \delta}{\tau_\delta}. \tag{A.2}$$

## A.2 Stability conditions

The following study focuses on the existence of stable steady state solutions to Eq. A.2 (denoted by $\bar{\delta}$) which, according to the theory of ODEs, need to satisfy:

$$k_\delta \delta_{\mathrm{BL}}(\bar{\delta}; \mathbf{u}) - \bar{\delta} = 0, \tag{A.3}$$

$$\frac{\partial}{\partial \delta}\left(\frac{k_\delta \delta_{\mathrm{BL}} - \delta}{\tau_\delta}\right)\Big|_{\delta=\bar{\delta}} < 0 \rightarrow k_\delta \left.\frac{\partial \delta_{\mathrm{BL}}}{\partial \delta}\right|_{\delta=\bar{\delta}} - 1 < 0. \tag{A.4}$$

Here $\mathbf{u}$ denotes a frozen state of the flow solution in the boundary layer.

An equivalent set of equations is obtained if the explicit relationship for $\delta_{\mathrm{BL}}$ ( Eq. 4.1 ) is introduced in the stability conditions above, which then read:

$$k_\delta \left(\overline{\theta_{\mathrm{k}}}\left(3.15 + \frac{1.72}{\overline{H_{\mathrm{k}}} - 1}\right) + \overline{\delta^*_{\mathrm{k}}}\right) - \bar{\delta} = 0, \tag{A.5}$$

$$k_\delta \left(1 - \frac{1.72}{(\overline{H_{\mathrm{k}}} - 1)^2}\right) \left.\frac{\partial \delta^*_{\mathrm{k}}}{\partial \delta}\right|_{\delta=\bar{\delta}} + k_\delta \left(3.15 + \frac{1.72}{\overline{H_{\mathrm{k}}} - 1} + \frac{1.72\overline{H_{\mathrm{k}}}}{(\overline{H_{\mathrm{k}}} - 1)^2}\right) \left.\frac{\partial \theta_{\mathrm{k}}}{\partial \delta}\right|_{\delta=\bar{\delta}} - 1 < 0. \tag{A.6}$$

Here, $\overline{\delta^*_{\mathrm{k}}} = \delta^*_{\mathrm{k}}(\bar{\delta})$, $\overline{\theta_{\mathrm{k}}} = \theta_{\mathrm{k}}(\bar{\delta})$ and $\overline{H_{\mathrm{k}}} = H_{\mathrm{k}}(\bar{\delta})$. Note that all the integral quantities are meant to be computed according to the procedure outline in Section 4.2.4. This

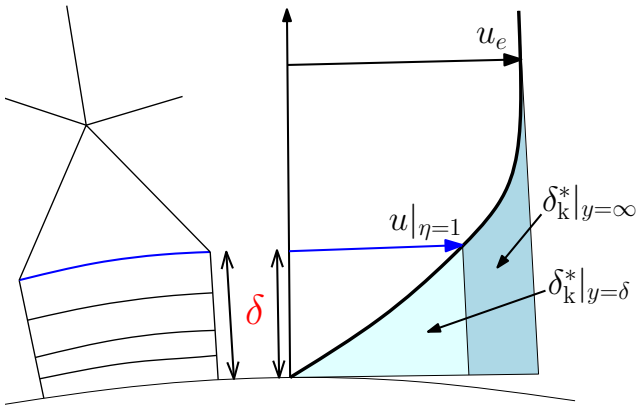



Figure A-1: The dependency of the approximate edge velocity on $\delta$ generates a discrepancy between the computed value of $\delta^*_{\mathrm{k}}$ (clear shade) and the actual value of $\delta^*_{\mathrm{k}}$ (dark shade).

means that the edge velocity is assumed to be equal to the velocity a distance $\delta$ away from the wall, independently of whether or not the viscous effects are negligible there. This is illustrated in Figure A-1.

By looking at Eq. A.6 it is clear that any value $\bar{\delta}$ that satisfies Eq. A.5 and is big enough to contain the whole boundary layer will be a stable solution to the problem, since then: $H_{\mathrm{k}} > 1$, $\frac{\partial \delta^*_{\mathrm{k}}}{\partial \delta} \approx 0$ and $\frac{\partial \theta_{\mathrm{k}}}{\partial \delta} \approx 0$ by definition.

Unfortunately, this result does not guarantee that any initial condition will smoothly evolve towards the desirable value $\bar{\delta}$, as there are instances in which there might be more than one attractor. Furthermore, developing a general theory seems rather difficult due to the dependency of the problem on the velocity profiles of the boundary layer, which in principle can adopt any shape.

## A.3 Stability analysis

To overcome this last obstacle, the velocity in the boundary is assumed to belong to a family of parametrized velocity profiles like the ones that can be found in most integral boundary layer codes.

In this study, laminar flows are approximated using the Falkner-Skan boundary layer profiles computed in inverse mode [67] as a function of $H_{\mathrm{k}}$, that can represent

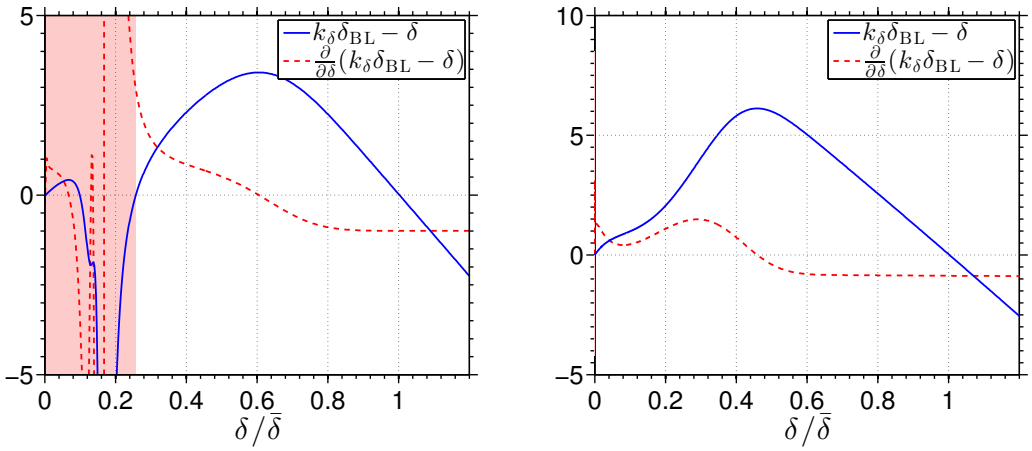
laminar flow in favorable and adverse pressure gradient, well beyond separation.

Similarly, turbulent flows are approximated using the analytical velocity profiles by Whitfield [253], and the extension by Swafford to separated flow [233]. These velocity profiles are analytical in $C_f$, $H_\mathrm{k}$ and $Re_{\theta_\mathrm{k}}$, this is: $u/u_e = f(y/\theta_\mathrm{k}; C_f, H_\mathrm{k}, Re_{\theta_\mathrm{k}})$, but can be reduced to the form $u/u_e = f(y/\theta_\mathrm{k}; H_\mathrm{k}, Re_{\theta_\mathrm{k}})$ with the help of the friction coefficient law proposed by Swafford [233], that takes the following functional form: $C_f = C_f(H_\mathrm{k}, Re_{\theta_\mathrm{k}})$.

The analysis of the stability limits for a given flow condition starts from a discretization of the velocity profile at a sufficient number of points (here in the order of 10000) using the corresponding model. This profile is then used to approximate the value of $\delta^*_\mathrm{k}$ and $\theta_\mathrm{k}$ at every station across the boundary layer (in the sense of Fig. A-1) using the trapezoidal rule. With these, the stability conditions in Eq. A.3 and A.4 are evaluated using a centered difference approximation for the derivative when required.

The limits of stability are obtained by analyzing this data. In particular, the lower stable limit (denoted by $\delta^-$) is found by looking at the unstable roots of the problem (e.g. points that satisfy A.3 and do not satisfy A.4) and taking the greatest one. This point, together with $+\infty$ brackets the region where Eq. A.2 is stable and any perturbation will evolve towards the target value $\bar{\delta}$. This process is illustrated in Figure A-2 for a laminar case with separation as well as a turbulent attached flow.

The stability curves for laminar and turbulent flows are generated using this procedure over the parameter space of the velocity family. In the case of laminar flows, this involves sampling the shape parameter $H_\mathrm{k}$ in the range $H_\mathrm{k} \in [2.2, 20]$, where the lower limit is close to the minimum attainable ($\approx 2.16$) and the upper limit is somewhat arbitrary. For turbulent flows, the sampling is carried out in the shape parameter also, with range $H_\mathrm{k} = [1.2, 3]$ for the Whitfield profiles and $H_\mathrm{k} = [2.5, 8]$ for the Swafford ones. These limits are directly taken from those used in the original papers [253, 233]. In addition, the turbulent profiles require sampling in the range of $Re_{\theta_\mathrm{k}}$, that, given the weak dependence of the solution on it, can be fairly coarse. Here, $Re_{\theta_\mathrm{k}} = \{10^3, 10^4, 10^5\}$ are used.




(a) Laminar, $H_{\mathrm{k}} = 6$ (b) Turbulent, $H_{\mathrm{k}} = 1.5$, $Re_{\theta_{\mathrm{k}}} = 10^3$

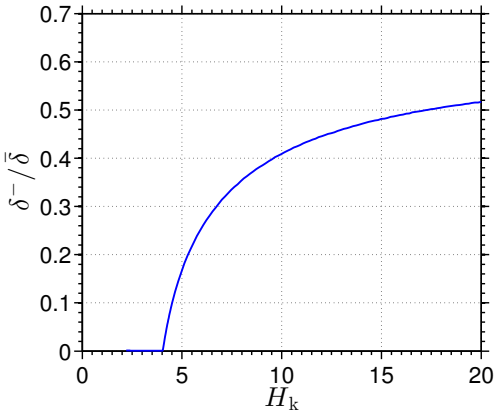
Figure A-2: Analysis of the stability of the thickness indicator equation for the case of a laminar separated flow (left) as well as a turbulent attached flow (right). The unstable region (shaded) represents the range of initial conditions for which $\delta$ evolves towards a solution other than $\bar{\delta}$.

The stability curves as a function of $H_{\mathrm{k}}$ are plotted in Figure A-3. The results indicate that the unstable region is minimal for most attached velocity profiles, but grows as the flow separates in direct correlation with the shape parameter. For the range of parameters tested here, the maximum value of $\delta^-/\bar{\delta}$ barely exceeds 0.5.

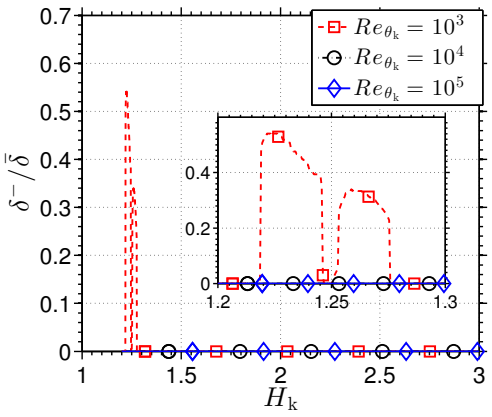
This behavior in separated flows is totally unacceptable as it might produce solutions where the boundary layer mesh leaves out some of the viscous effects, hence defeating the whole purpose of the $r$-adaptivity and possibly affecting the convergence of the solver.

## A.4 Parameter choice

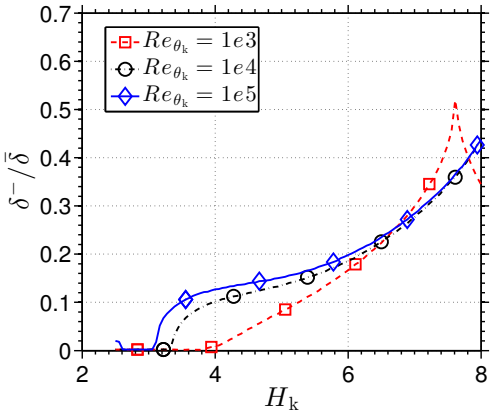
A naive remedy to this problem is to initialize $\delta_h$ to a very thick value so that equilibrium is approached from the right (see Fig. A-2). However, this would affect the stiffness of the nonlinear iteration in the early stages of the simulation and would not prevent an undershoot in the discrete solution which might bring the system into the unstable region anyways.




(a) Falkner-Skan profiles




(b) Whitfield profiles




(c) Swafford profiles

Figure A-3: Lower stability limit $\delta^-/\bar{\delta}$, computed using different families of velocity profiles (Falkner-Skan, Whitfield and Swafford) and different input parameters ($H_k$ and $Re_{\theta_k}$). This result shows a strong correlation between the lower stability limit and $H_k$ past separation for both laminar and turbulent flows.

Another possible solution would be to approximate the initial value of $\delta_{BL}$, and take the required measures to ensure that $\delta_h$ follows it closely as it evolves in time. This approach is the one advocated for in what follows.

### A.4.1 Initialization of $\delta$

The approximate value of $\delta_{\mathrm{BL}}$ by the end of the first time step $\Delta t$ can be estimated with the help of the analytical solution to Rayleigh's problem, which describes the evolution of the velocity field in a semi-infinite flow, initially at rest, when the solid wall suffers an impulse and starts to move with constant velocity $u_\infty$:

$$u = u_\infty \left(1 - \mathrm{erf}\left(\frac{y}{2\sqrt{\nu t}}\right)\right). \tag{A.7}$$

By changing the frame of reference, this solution can be adapted to the case in which the flow is moving with uniform velocity $u_\infty$ and the wall boundary conditions are suddenly imposed. This is very similar to the actual initialization of the solver, and hence can be used as a surrogate for the velocity profile at the first time step. In particular, after the proper non-dimensionalization, the velocity profile reads:

$$\frac{u}{u_\infty} = \mathrm{erf}\left(\frac{(y/L)\sqrt{Re_L}}{2}\sqrt{\frac{L}{tu_\infty}}\right), \tag{A.8}$$

which is an explicit expression that depends on the distance $y$ away from the wall, the reference length $L$, the Reynolds number and the magnitude of the velocity itself. Integrating the boundary layer thicknesses analytically yields:

$$\delta_{\mathrm{k}}^* = \frac{2L}{\sqrt{\pi Re_L}}\sqrt{\frac{tu_\infty}{L}}, \tag{A.9}$$

$$\theta_{\mathrm{k}} = (\sqrt{2} - 1)\frac{2L}{\sqrt{\pi Re_L}}\sqrt{\frac{tu_\infty}{L}}. \tag{A.10}$$

These relationships, together with Eq. 4.1, can be used to estimate $\delta_{\mathrm{BL}}$ by the end of the first time step $\Delta t$. Combining this with $k_\delta$ yields an estimate for the initial value of the normal scaling:

$$\frac{\delta}{L} \approx 3.1619 k_\delta \frac{1}{\sqrt{Re_L}}\sqrt{\frac{\Delta t u_\infty}{L}}. \tag{A.11}$$

This ensures that the edge of the boundary layer domain is placed outside the viscous region during the first time step, yet close enough to avoid problems due to lack of resolution.

### A.4.2 Selection of $\tau_\delta$

Unfortunately, tuning the initial condition is not enough to ensure that $\delta$ will remain outside the stability region at all times, since the opposite might easily happen if $\delta$ does not react fast enough to the changes in the boundary layer (or equivalently $\delta_{\mathrm{BL}}$).

To circumvent this, $\tau_\delta$ is taken to be proportional to the time step of the discretization $\Delta t$. In this way, the reaction time of the system can be crafted so that changes of the order of $k_\delta \delta_{\mathrm{BL}} - \delta$ (which represent the error in the location of the boundary layer edge) happen within the allocated time step $\Delta t$. For all the results presented in this thesis, the proportionality constant is equal to 2, hence the rule $\tau_\delta = 2\Delta t$.

## A.5 Summary of the study

The stability of the equation that governs the normal scaling $\delta$ has been studied with the help of a simplified model. The accompanying results indicate that there is only one stable solution for $\delta$ in the case of attached flows, hence, the location of the edge of the boundary layer domain is uniquely determined.

Unfortunately, this is not the case for separated flows, where two or more stable solutions might exist. Of these, only one is correct and captures the whole viscous layer, while the others are spurious and appear due the dependency of the state at the edge of the boundary layer on $\delta$ itself.

In general, no guarantee on the convergence to the right solution exists; however, a judicious choice of the initial conditions and the parameters that govern the model should alleviate this problem. Needless to say that other forms of $\delta_{\mathrm{BL}}$ might help avoid this undesirable behavior or at least help extend the limit of stability $\delta^-$.

The assumptions made for this simple study are reasonably strong and obviate

effects like the diffusivity on the governing PDE or the fact that the boundary layer profiles evolve with time too. The experience gained during the testing of the solver seems to indicate, that, if anything, these play in favor of the stability. However, this is a purely empirical observation that needs to be formally addressed.

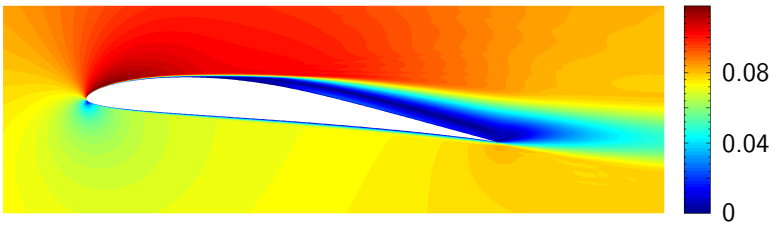
# Appendix B

# On the transition to turbulence of the Spalart-Allmaras model at low Reynolds numbers

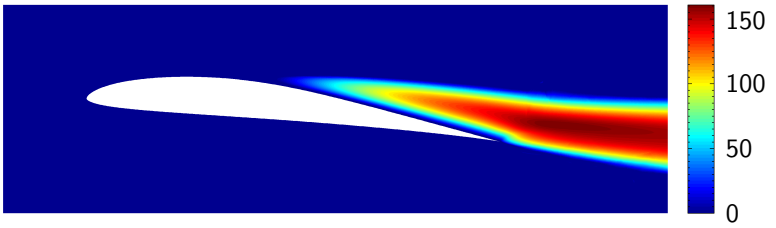
This appendix presents some findings concerning the behavior of the SA model at low Reynolds number, with special emphasis on the transition to turbulence in laminar separation bubbles. The study was triggered by some early simulations of the Eppler 387 airfoil at $Re_c = 10^5$ for which the flow transitioned due to separation but did not reattach, as shown in Figure B-1 for the case of $\alpha = 6°$.

In the beginning, these was deemed to be caused by an incorrect implementation of the SA model in the solver, which Dr. Steven Allmaras [7] quickly ruled out by suggesting to try a higher Reynolds number case. Surely enough the model behaved as expected for $Re_c > 2 \cdot 10^5$ (see results in Chapter 5), which turned the attention to the differences between the original model [223] and the one implemented here (see Section 2), mostly in terms of the modifications that affect transition.

At approximately the same time, Crivellini and D'Alessandro [53] published a paper in which they used the fully turbulent form of the SA model (without $f_{t1}$ and $f_{t2}$ terms), to predict transition in laminar separation bubbles. Amongst other things, the authors performed a careful study of the effect of $\tilde{\nu}_\infty$ on the computed results and




(a) Mach number field




(b) $\mu_t/\mu_\infty$ field

Figure B-1: Flow around an Eppler 387 airfoil at $Re_c = 10^5$ and $\alpha = 6°$. The flow remains detached over the upper surface despite the fact that transition occurs, as evidenced by the eddy viscosity field.

found that the laminar separation bubble could be totally suppressed for values as low as $\tilde{\nu}_\infty/\nu_\infty = 10^{-11}$.

Extrapolating it to the case of interest here, this result comes to say that the reattachment can depend strongly on the value of $\tilde{\nu}_\infty$ at the point where transition is predicted (hence $\gamma = 1$) and the source terms are allowed to act. In this spirit, Prof. Drela [66] suggested the use of a source term around the transition point to mimic the Reynolds stresses due to the Kelvin-Helmholtz instability of the shear layer. For this, the SA model was augmented with a source term in the boundary layer domain of the form:

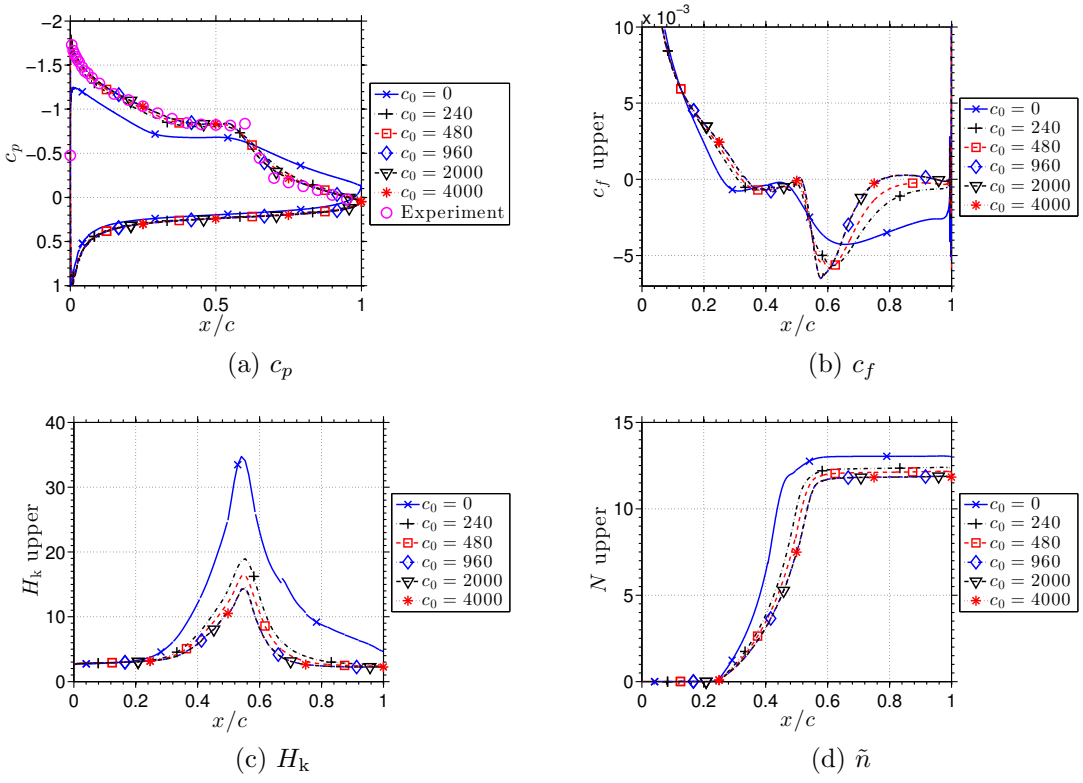
$$s_{trip} = c_0 \rho \left(\omega \frac{d}{u_e}\right)^2 \exp\left(-\left(\frac{\tilde{n} - N_{\text{crit}}}{\Delta \tilde{n}}\right)^2\right) \|\mathbf{v}\|^2, \tag{B.1}$$

where denotes $\omega$ the vorticity, and the same case was run again for increasingly higher values of $c_0$.

The results of this exercise, that are compiled in Figure B-2, show a strong depen-

dence of the solution on $c_0$, that is patent in the evolution of the stresses at the wall as well as the shape parameter $H_k$ and the amplification factor $\tilde{n}$. Furthermore, the results seem to reach a saturation point past a certain value of $c_0$. One of such cases is shown in Figure B-3.

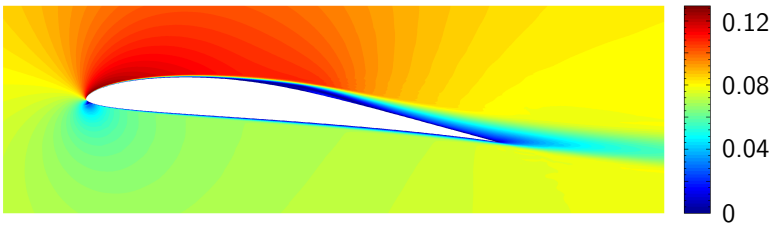
These preliminary results confirm the strong dependency of the solution on the initial condition at transition, which will need to be properly addressed if the model is to be applied in the low Reynolds number regime. However, this might not be enough as the saturation in $c_0$ happens before a good agreement with the experiments is found (see Fig. B-2a). This last point indicates that the SA model might need to be modified or re-calibrated in this regime for it to faithfully reproduce the data. These two tasks will be the subject of future research.



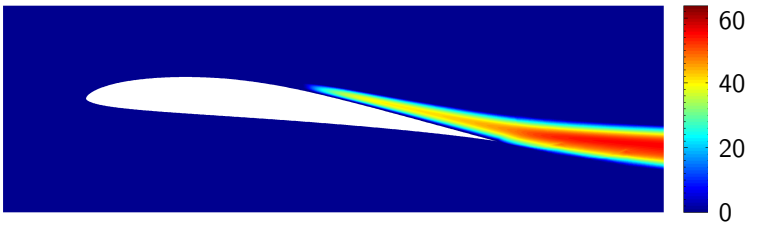

(a) $c_p$ (b) $c_f$

(c) $H_k$ (d) $\tilde{n}$

Figure B-2: Evolution of $c_p$, $c_f$, $H_k$ and $\tilde{n}$ on the upper surface of the airfoil as the constant $c_0$ is increased. The dependency on $c_0$ is very strong at first but quickly reaches a saturation point.




(a) Mach number field




(b) $\mu_t/\mu_\infty$ field

Figure B-3: Flow around an Eppler 387 airfoil at $Re_c = 10^5$ and $\alpha = 6°$ when the trip term constant is set to $c_0 = 2000$. In this case, the action of the trip term at the transition point is enough to induce the reattachment of the flow.

# Bibliography


[1] IDIHOM - Industrialisation of High-Order Methods A Top-Down Approach. `http://www.idihom.de/`.

[2] Volume II, Compiled Data. In D.E. Coles and E.A. Hirst, editors, *Computation of Turbulent Boundary Layers - 1968 AFOSR-IFP-Stanford Conference*, Stanford University, CA, 1968.

[3] H. Abbassi, F. Mashayek, and G. B. Jacobs. Shock capturing with entropy-based artificial viscosity for staggered grid discontinuous spectral element method. *Computers & Fluids*, 98:152–163, 2014.

[4] B. J. Abu-Ghannam and R. Shaw. Natural transition of boundary layers–The effects of turbulence, pressure gradient, and flow history. *Journal of Mechanical Engineering Science*, 22(5):213–228, 1980.

[5] F. Alauzet and D. Marcum. A Closed Advancing-Layer Method with Changing Topology Mesh Movement for Viscous Mesh Generation. In J. Sarrate and M. Staten, editors, *22nd International Meshing Roundtable*, pages 241–261, Orlando, FL, 2013. Springer International Publishing.

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

[7] S. R. Allmaras. Personal Communication.

[8] S. R. Allmaras. *A coupled Euler/Navier-Stokes algorithm for 2-D unsteady transonic shock/boundary-layer interaction*. PhD thesis, Massachusetts Institute of Technology, 1989.

[9] S. R. Allmaras and M. Giles. A Second Order Flux Split Scheme for the Unsteady 2 - D Euler Equations on Arbitrary Meshes. In *27th Aerospace Sciences Meeting*, 1987.

[10] S. R. Allmaras, F. T. Johnson, and P. R. Spalart. Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model. In *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, Big Island, HI, 2012.

[11] D. Arnal. Transition Prediction in Transonic Flow. In J. Zierep and H. Oertel, editors, *Symposium Transsonicum III*, pages 253–262. Springer, Berlin-Heidelberg, 1988.

[12] D. Arnal. Boundary layer transition: predictions based on linear theory. In *Special Course on Progress in transition Modelling (AGARD-R-793)*, pages 63–126, Madrid, Spain, 1994.

[13] D. Arnal and G. Casalis. Laminar-turbulent transition prediction in three-dimensional flows. *Progress in Aerospace Sciences*, 36:173–191, 2000.

[14] D. Arnal, G. Casalis, and R. Houdeville. Practical Transition Prediction Methods: Subsonic and Transonic Flows. In *AVT-151 RTO AVT/VKI Lecture Series*, pages 1–34, Rhode St. Genèse, Belgium, 2008.

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

[16] I. Babuška and B. Q. Guo. The $h$, $p$ and $h$–$p$ version of the finite element method; basis theory and applications. *Advances in Engineering Software*, 15:159–174, 1992.

[17] I. Babuška, B. A. Szabo, and I. N. Katz. The $p$-version of the Finite Element Method. *SIAM Journal on Numerical Analysis*, 18(3):515–545, 1981.

[18] T. J. Baker. Mesh generation: Art or science? *Progress in Aerospace Sciences*, 41:29–63, 2005.

[19] B. S. Baldwin and H. Lomax. Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows. In *16th AIAA Aerospace Sciences Meeting*, Huntsville, AL, 1978.

[20] G. E. Barter. *Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive, Higher-Order Discontinuous Galerkin Finite Element Method*. PhD thesis, Massachusetts Institute of Technology, 2008.

[21] G. E. Barter and D. L. Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *Journal of Computational Physics*, 229(5):1810–1827, 2010.

[22] F. Bassi and A. Crivellini. High-Order Discontinuous Galerkin Discretization of Transonic Turbulent Flows. In *47th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2009.

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

[24] F. Bassi and S. Rebay. Accurate 2D Euler Computations by means of a High Order Discontinuous Finite Element Method. In *14th International Conference on Numerical Methods in Fluids*, pages 234–240, Bangalore, India, 1994.

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

[26] F. Bassi and S. Rebay. High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations. *Journal of Computational Physics*, 138(2):251–285, 1997.

[27] C. E. Baumann and J. T. Oden. An adaptive-order discontinuous Galerkin method for the solution of the Euler equations of gas dynamics. *International Journal for Numerical Methods in Engineering*, 73:61–73, 2000.

[28] A. Bhagatwala and S. K. Lele. A modified artificial viscosity approach for compressible turbulence simulations. *Journal of Computational Physics*, 228(14):4965–4969, 2009.

[29] J. Boussinesq. *Essai sur la théorie des eaux courantes*. Imprimerie nationale, 1877.

[30] R. R. Brodeur and C. P. van Dam. Transition prediction for a two-dimensional Reynolds-averaged Navier-Stokes method applied to wind turbine airfoils. *Wind Energy*, 4(2):61–75, 2001.

[31] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[32] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111 – 241, 2009.

[33] N. K. Burgess. *An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows*. PhD thesis, University of Wyoming, 2011.

[34] N. K. Burgess and D. J. Mavriplis. Robust Computation of Turbulent flows using a Discontinuous Galerkin Method. In *50th AIAA Aerospace Sciences Meeting and Exhibit*, Nashville, TN, 2012.

[35] C.D. Cantwell, S. Yakovlev, R. M. Kirby, N. S. Peters, and S. J. Sherwin. High-order spectral/*hp* element discretisation for reaction-diffusion problems on surfaces: application to cardiac electrophysiology. *Journal of Computational Physics*, 257:813–829, 2014.

[36] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods in fluid dynamics*. Springer-Verlag, 1988.

[37] J. E. Carter. Numerical solutions of the supersonic, laminar flow over a two-dimensional compression corner. In *3rd International Conference on Numerical Methods in Fluid Mechanics*, pages 69–78, 1973.

[38] G. Casalis and D. Arnal. ELFIN II, Subtask 2.3: Database method. Development and validation of the simplified method for pure crossflow instability at low speed. Technical Report 119/5618.16, CERT/ONERA, Toulouse, France, 1996.

[39] M. Ceze and K. J. Fidkowski. An anisotropic *hp*-adaptation framework for functional prediction. *AIAA Journal*, 51(2):492–509, 2013.

[40] M. Ceze and K. J. Fidkowski. Pseudo-transient Continuation, Solution Update Methods, and CFL Strategies for DG Discretizations of the RANS-SA Equations. In *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 2013.

[41] J. Chan, L. Demkowicz, and R. Moser. A DPG method for steady viscous compressible flow. *Computers & Fluids*, 98:69–90, 2014.

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

[43] C.-L. Chang. The Langley Stability and Transition Analysis Code (LASTRAC): LST, Linear & Nonlinear PSE for 2-D, Axisymmetric, and Infinite Swept Wing Boundary Layers. In *41st Aerospace Sciences Meeting*, Reno, NV, USA, 2003.

[44] H. K. Chaurasia. *A Time-Spectral Hybridizable Discontinuous Galerkin Method for Periodic Flow Problems*. PhD thesis, Massachusetts Institute of Technology, 2014.

[45] J. Cliquet, R. Houdeville, and D. Arnal. Application of Laminar-Turbulent Transition Criteria in Navier-Stokes Computations. *AIAA Journal*, 46(5):1182–1190, 2008.

[46] B. Cockburn, B. Dong, J. Guzman, M. Restrelli, and R. Sacco. A hybridizable Discontinuous Galerkin method for steady-state convection-diffusion-reaction problems. *SIAM Journal on Scientific Computing*, 31(5):3827–3846, 2009.

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

[48] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection P1-discontinuous-Galerkin finite element method for scalar conservation laws. Technical Report 388, IMA, University of Minnesota, Minneapolis, MN, 1988.

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

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

[51] J. G. Coder and M. D. Maughmer. A CFD-Compatible Transition Model Using an Amplification Factor Transport Equation. In *51st AIAA Aerospace Sciences Meeting*, Grapevine, TX, 2013.

[52] J. G. Coder and M. D. Maughmer. Computational Fluid Dynamics Compatible Transition Modeling Using an Amplification Factor Transport Equation. *AIAA Journal*, 52(11):2506–2512, 2014.

[53] A. Crivellini and V. D'Alessandro. Spalart-Allmaras model apparent transition and RANS simulations of laminar separation bubbles on airfoils. *International Journal of Heat and Fluid Flow*, 47:70–83, 2014.

[54] A. Crivellini, V. D'Alessandro, and F. Bassi. A Spalart-Allmaras turbulence model implementation in a discontinuous Galerkin solver for incompressible flows. *Journal of Computational Physics*, 241:388–415, 2013.

[55] A. Crivellini, V. D'Alessandro, and F. Bassi. High-order discontinuous Galerkin solutions of three-dimensional incompressible RANS equations. *Computers & Fluids*, 81:122–133, 2013.

[56] J. D. Crouch, I. W. M. Crouch, and L. L. Ng. Transition Prediction for Three-Dimensional Boundary Layers in Computational Fluid Dynamics Applications. *AIAA Journal*, 40(8):1536–1541, 2002.

[57] J. D. Crouch and L. L. Ng. Variable N-Factor Method for Transition Prediction in Three-Dimensional Boundary Layers. *AIAA Journal*, 38(2):211–216, 2000.

[58] J. R. Dagenhart. Amplified Crossflow Disturbances in the Laminar Layer on Swept Wings with Suction. Technical Report NASA-TP-1902, NASA, 1981.

[59] M. B. Davis, H. L. Reed, H. Youngren, B. Smith, and E. Bender. Transition Prediction Method Review Summary for the Rapid Assessment Tool for Transition Prediction (RATTraP). Technical Report AFRL-VA-WP-TR-2005-3130, Air Force Research Lab, Wright-Patterson AFB, OH, 2005.

[60] G. Degrez, C.H. Boccadoro, and J.F. Wendt. The interaction of an oblique shock wave with a laminar boundary layer revisited . An experimental and numerical study. *Journal of Fluid Mechanics*, 177:247–263, 1987.

[61] J. Delery and J.G. Marvin. Shock-Wave Boundary Layer Interactions. Technical Report 21, AGARD-AG-280, 1986.

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

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

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

[65] Y. Di, R. Li, T. Tang, and P. Zhang. Moving Mesh Finite Element Methods for the Incompressible Navier–Stokes Equations. *SIAM Journal on Scientific Computing*, 26(3):1036–1056, 2005.

[66] M. Drela. Personal Communication.

[67] M. Drela. A new transformation and integration scheme for the compressible boundary layer equations, and solution behavior at separation. MSc Thesis, Massachusetts Institute of Technology, 1983.

[68] M. Drela. Implicit Implementation of the Full $e^N$ Transition Criterion. In *21st AIAA Applied Aerodynamics Conference*, Orlando, FL, 2003.

[69] M. Drela and M. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10):1347–1355, 1987.

[70] M. Drosson and K. Hillewaert. On the stability of the symmetric interior penalty method for the Spalart-Allmaras turbulence model. *Journal of Computational and Applied Mathematics*, 246:122–135, 2013.

[71] M. Drosson, K. Hillewaert, and J.-A. Essers. Stability and boundary resolution analysis of the discontinuous Galerkin method applied to the Reynolds–averaged Navier-Stokes equations using the Spalart-Allmaras model. *SIAM Journal on Scientific Computing*, 35(3):666–700, 2013.

[72] F. Ducros, V. Ferrand, F. Nicoud, C. Weber, D. Darracq, C. Gacherieu, and T. Poinsot. Large-Eddy Simulation of the Shock/Turbulence Interaction. *Journal of Computational Physics*, 152(2):517–549, 1999.

[73] P. A. Durbin. A Reynolds stress model for near-wall turbulence. *Journal of Fluid Mechanics*, 249:465, 1993.

[74] G. Dziuk and C. M. Elliott. Surface finite elements for parabolic equations. *Journal of Computational Mathematics*, 25(4):385–407, 2007.

[75] B. Eisfeld, A. Probst, S. Crippa, and R. Radespiel. Differential Reynolds Stress Modeling for Aeronautics. In *50th AIAA Aerospace Sciences Meeting and Exhibit*, Nashville, TN, 2012.

[76] K. J. Fidkowski. *A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2007.

[77] P.J. Frey and F. Alauzet. Anisotropic mesh adaptation for CFD computations. *Computer Methods in Applied Mechanics and Engineering*, 194:5068–5082, 2005.

[78] B. Froehle and P. O. Persson. High-Order Accurate Fluid-Structure Simulation of a Tuning Fork. *Computers & Fluids*, 98:230–238, 2013.

[79] M. C. Galbraith. Personal Communication.

[80] M. C. Galbraith, J. A. Benek, P. D. Orkwis, and M. G. Turner. A Discontinuous Galerkin Chimera Scheme. *Computers & Fluids*, 98:27–53, 2014.

[81] A. Gargallo-Peiro, X. Roca, J. Peraire, and J. Sarrate. Defining Quality Measures for Validation and Generation of High-Order Tetrahedral Meshes. In J. Sarrate and M. Staten, editors, *22nd International Meshing Roundtable*, pages 109–126, Orlando, FL, USA, 2013. Springer International Publishing.

[82] R. V. Garimella and M. S. Shephard. Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering*, 49:193–218, 2000.

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

[84] C. Gleyzes, J. Cousteix, and J.L. Bonnet. A Calculation Method of Leading-Edge Separation Bubbles. In T. Cebeci, editor, *Numerical and Physical Aspects of Aerodynamic Flows II*, pages 173–192, Long Beach, CA, 1983.

[85] M. E. Goldstein and L. S. Hultgren. Boundary-layer receptivity to long-wave free-stream disturbances. *Annual Review of Fluid Mechanics*, 21:137–166, 1989.

[86] D. Gottlieb and S. A. Orszag. *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM, Philadelphia, PA, USA, 1977.

[87] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248–4267, 2011.

[88] P. Hall, M. R. Malik, and D. I. A. Poll. On the stability of an infinite swept attachment line boundary layer. *Proc. London Math. Soc.*, 395(1809):229–245, 1984.

[89] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[90] 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, 2006.

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

[92] 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.

[93] O. Hassan, E. J. Probert, K. Morgan, and J. Peraire. Mesh generation and adaptivity for the solution of compressible viscous high speed flows. *International Journal for Numerical Methods in Engineering*, 148:1123–1148, 1995.

[94] F. Hecht. BAMG : Bidimensional Anisotropic Mesh Generator. Technical report, INRIA-Rocquencourt, 2006.

[95] T. Herbert. Secondary instability of boundary layers. *Annual Review of Fluid Mechanics*, 20:487–526, 1988.

[96] T. Herbert. Parabolized Stability Equations. *Annual Review of Fluid Mechanics*, 29:245–283, 1997.

[97] J. S. Hesthaven and T. Warburton. Beyond one dimension. In *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, chapter 6. Springer, 2007.

[98] C. Hu and C.-W. Shu. Weighted Essentially Non-oscillatory Schemes on Triangular Meshes. *Journal of Computational Physics*, 150(1):97–127, 1999.

[99] W. Huang and R. D. Russell. *Adaptive Moving Mesh Methods*. Springer Science, New York, 2011.

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

[101] A. Huerta, E. Casoni, and J. Peraire. A simple shock-capturing technique for high-order discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 69(10):1614–1632, 2012.

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

[103] T. J. R. Hughes, G. Scovazzi, and T. E. Tezduyar. Stabilized Methods for Compressible Flows. *Journal of Scientific Computing*, 43(3):343–368, 2008.

[104] H. T. Huynh. A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, FL, 2007.

[105] F. Ilinca and D. Pelletier. Positivity preservation and adaptive solution of two-equation models of turbulence. *International Journal of Thermal Sciences*, 38(7):560–571, 1999.

[106] T. Ishihara, T. Gotoh, and Y. Kaneda. Study of High–Reynolds Number Isotropic Turbulence by Direct Numerical Simulation. *Annual Review of Fluid Mechanics*, 41(1):165–180, 2009.

[107] A. Jameson. Analysis and Design of Numerical Schemes for Gas Dynamics, 2: Artificial Diffusion and Discrete Shock Structure. *International Journal of Computational Fluid Dynamics*, 5:1–38, 1995.

[108] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. In *14th AIAA Fluid and Plasma Dynamics Conference*, Palo Alto, CA, 1981.

[109] F. T. Johnson, E. N. Tinoco, and N. J. Yu. Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle. *Computers & Fluids*, 34(10):1115–1151, 2005.

[110] Y. Kallinderis and S. Wardt. Prismatic Grid Generation for Three-Dimensional Complex Geometries. *AIAA Journal*, 31(10):1850–1856, 1993.

[111] S. M. Kast and K. J. Fidkowski. Output-based mesh adaptation for high order Navier-Stokes simulations on deformable domains. *Journal of Computational Physics*, 252:468–494, 2013.

[112] A. J. Katz, A. M. Wissink, V. Sankaran, R. L. Meakin, and W. M. Chan. Application of strand meshes to complex aerodynamic flow fields. *Journal of Computational Physics*, 230(17):6512–6530, 2011.

[113] S. Kawai and S. K. Lele. Localized artificial diffusivity scheme for discontinuity capturing on curvilinear meshes. *Journal of Computational Physics*, 227(22):9498–9526, 2008.

[114] C. T. Kelley and D. E. Keyes. Convergence Analysis of Pseudo-transient Continuation. *SIAM Journal on Numerical Analysis*, 35(2):508–523, 1998.

[115] J. Kim and P. Moin. Application of a Fractional-Step Incompressible Method to Incompressible Navier-Stokes Equations. *Journal of Computational Mathematics*, 59:308–323, 1985.

[116] A. Klöckner, T. Warburton, and J. S. Hesthaven. Viscous Shock Capturing in a Time-Explicit Discontinuous Galerkin Method. *Mathematical Modelling of Natural Phenomena*, 10(10):1–27, 2011.

[117] D. A. A. Knoll and D. E. Keyes. Jacobian–free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[118] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Dokl. Akad. Nauk SSSR*, 30(4):9–13, 1941.

[119] D. A. Kopriva. A Staggered-Grid Multidomain Spectral Method for the Compressible Navier-Stokes Equations. *Journal of Computational Physics*, 143(1):125–158, 1998.

[120] N. Krimmelbein and A. M. Krumbein. Automatic Transition Prediction for Three-Dimensional Configurations with Focus on Industrial Application. *Journal of Aircraft*, 48(6):1878–1887, 2011.

[121] N. Krimmelbein and R. Radespiel. Transition prediction for three-dimensional flows using parallel computation. *Computers & Fluids*, 38(1):121–136, 2009.

[122] N. Krimmelbein, R. Radespiel, and C. Nebel. Numerical Aspects of Transition Prediction for Three-Dimensional Configurations. In *35th AIAA Fluid Dynamics Conference*, Orlando, FL, 2005.

[123] L. Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

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

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

[126] A. M. Krumbein. Automatic Transition Prediction for High-Lift Systems Using a Hybrid Flow Solver. *Journal of Aircraft*, 42(5):1362–1366, 2005.

[127] A. M. Krumbein. Automatic Transition Prediction and Application to Three-Dimensional Wing Configurations. *Journal of Aircraft*, 44(1):119–133, 2007.

[128] A. M. Krumbein, N. Krimmelbein, and G. Schrauf. Automatic Transition Prediction in Hybrid Flow Solver, Part 1: Methodology and Sensitivities. *Journal of Aircraft*, 46(4):1176–1190, 2009.

[129] A. M. Krumbein, N. Krimmelbein, and G. Schrauf. Automatic Transition Prediction in Hybrid Flow Solver, Part 2: Practical Application. *Journal of Aircraft*, 46(4):1191–1199, 2009.

[130] A. Kumar and S. N. Tiwari. Analysis of the Scramjet inlet flow field using two-dimensional Navier-Stokes equations. Technical Report NASA-CR-3562, NASA Langley Research Center, Hampton, VA, 1982.

[131] K. Kusunose and H. V. Cao. Prediction of transition location for a 2-D Navier-Stokes solver for multi-element airfoil configurations. In *25th AIAA Fluid Dynamics Conference*, Colorado Springs, CO, 1994.

[132] B. Landmann, M. Kessler, S. Wagner, and E. Krämer. A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows. *Computers & Fluids*, 37:427–438, 2008.

[133] R. B. Langtry and F. R. Menter. Transition Modeling for General CFD Applications in Aeronautics. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2005.

[134] R. B. Langtry and F. R. Menter. Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes. *AIAA Journal*, 47(12):2894–2906, 2009.

[135] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the development of a Reynolds-stress turbulence closure. *Journal of Fluid Mechanics*, 68(03):537–566, 1975.

[136] B. E. Launder and B. I. Sharma. Applications of energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in Heat and Mass Transfer*, 1(2):131–138, 1974.

[137] J.-D. Lee and A. Jameson. Natural-Laminar-Flow Airfoil and Wing Design by Adjoint Method and Automatic Transition Prediction. In *47th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2009.

[138] S. K. Lele. Compact Finite Difference Schemes with Spectral-like Resolution. *Journal of Computational Physics*, 103:16–42, 1992.

[139] D. W. Levy, K. R. Laflin, E. N. Tinoco, J. C. Vassberg, M. Mani, B. Rider, C. L. Rumsey, R. A. Wahls, J. H. Morrison, O. P. Brodersen, S. Crippa, D. J. Mavriplis, and M. Murayama. Summary of Data from the Fifth AIAA CFD Drag Prediction Workshop. In *51st AIAA Aerospace Sciences Meeting*, Grapevine, TX, 2013.

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

[141] A. C. Limache. *Aerodynamic Modeling Using Computational Fluid Dynamics and Sensitivity Equations*. PhD thesis, Virginia Polytechnic Institute and State University, 2000.

[142] X. D. Liu, S. Osher, and T. Chan. Weighted Essentially Non-oscillatory Schemes. *Journal of Computational Physics*, 115:200–212, 1994.

[143] A. Loseille and R. Löhner. On 3D anisotropic local remeshing for surface, volume and boundary layers. In *18th International Meshing Roundtable*, pages 611–630, Berlin-Heidelberg, 2009. Springer.

[144] L. M. Mack. Transition prediction and linear stability theory. In *AGARD-CP-224*, 1977.

[145] L. M. Mack. Boundary Layer Linear Stability Theory. Technical Report AD-A417-243, Jet Propulsion Laboratory, 1984.

[146] D. G. Martineau, S. Stokes, S. J. Munday, A. P. Jackson, B. J. Gribben, and N. A. Verhoeven. Anisotropic Hybrid Mesh Generation for Industrial RANS Applications. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 2006.

[147] K. Mattsson and J. Nordström. Summation by parts operators for finite difference approximations of second derivatives. *Journal of Computational Physics*, 199(2):503–540, 2004.

[148] D. J. Mavriplis. Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation. *Journal of Computational Physics*, 90:271–291, 1990.

[149] E. A. Mayda. *Boundary-layer transition prediction for Reynolds-averaged Navier-Stokes methods*. PhD thesis, University of California Davis, 2007.

[150] R. McGhee, G .S. Jones, and R. Jouty. Performance Characteristics from Wind-Tunnel Tests of a Low-Reynolds-Number Airfoil. In *26th AIAA Aerospace Science Meeting*, Reno, NV, 1988.

[151] S. Medida and J. D. Baeder. Application of the Correlation-based $\gamma - \overline{Re_{\theta_t}}$ Transition Model to the Spalart-Allmaras Turbulence Model. In *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, HI, 2011.

[152] S. Medida and J. D. Baeder. A New Crossflow Transition Onset Criterion for RANS Turbulence Models. In *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 2013.

[153] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, 1994.

[154] T. Michal and J. Krakos. Anisotropic Mesh Adaptation Through Edge Primitive Operations. In *50th AIAA Aerospace Sciences Meeting and Exhibit*, Nashville, TN, 2012.

[155] J.M. Modisette. *An Automated Reliable Method for Two-Dimensional Reynolds-averaged Navier-Stokes Simulations*. PhD thesis, Massachusetts Institute of Technology, 2011.

[156] P. Moin and K. Mahesh. Direct Numerical Simulation: A Tool in Turbulence Research. *Annual Review of Fluid Mechanics*, 30(1):539–578, 1998.

[157] M. V. Morkovin. Critical evaluation of transition from laminar to turbulent shear layers with emphasis on hypersonically traveling bodies. Technical Report AFFDL-TR-08-149, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, OH, 1969.

[158] D. Moro, N.C. Nguyen, and J. Peraire. Navier-Stokes solution using Hybridizable Discontinuous Galerkin methods. In *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, HI, 2011.

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

[160] D. Moro, N.C. Nguyen, J. Peraire, and J. Gopalakrishnan. A Hybridized Discontinuous Petrov-Galerkin Method for Compresible Flows. In *49th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2011.

[161] K. Nakahashi. A finite-element method on prismatic elements for the three-dimensional Navier-Stokes equations. In *11th International Meshing Roundtable*, pages 434–438. Springer, 1989.

[162] K. Nakahashi. Adaptive prismatic-grid method for external viscous flow computations. In *11th AIAA Computational Fluid Dynamics Conference*, pages 195–203, 1993.

[163] C. Nebel, R. Radespiel, and T. Wolf. Transition prediction for 3D flows using a Reynolds-averaged Navier-Stokes code and N-factor methods. In *33rd AIAA Fluid Dynamics Conference and Exhibit*, Orlando, FL, 2003.

[164] N.C. Nguyen. Personal Communication.

[165] N.C. Nguyen and J. Peraire. An Adaptive Shock-Capturing HDG Method for Compressible Flows. In *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, HI, 2011.

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

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

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

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

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

[171] N.C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwells equations. *Journal of Computational Physics*, 230(19):7151–7175, 2011.

[172] N.C. Nguyen, P. O. Persson, and J. Peraire. RANS solutions using high order discontinuous Galerkin methods. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2007.

[173] N.C. Nguyen, X. Roca, D. Moro, and J. Peraire. A Hybridized Multiscale Discontinuous Galerkin Method for Compressible Flows. In *51st AIAA Aerospace Sciences Meeting*, Grapevine, TX, 2013.

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

[175] T. A. Oliver. *A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations*. PhD thesis, Massachusetts Institute of Technology, 2008.

[176] T. A. Oliver and D. L. Darmofal. An Unsteady Adaptation Algorithm for Discontinuous Galerkin Discretizations of the RANS Equations. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, FL, 2007.

[177] T. A. Oliver and D. L. Darmofal. Impact of Turbulence Model Irregularity on High-Order Discretizations. In *47th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2009.

[178] M. A. Park and D. L. Darmofal. Parallel Anisotropic Tetrahedral Adaptation. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2008.

[179] A. T. Patera. A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion. *Journal of Computational Physics*, 54:468–488, 1984.

[180] J. Peraire, N.C. Nguyen, and B. Cockburn. A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations. In *48th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2010.

[181] J. Peraire, J. Peiro, and K. Morgan. Adaptive Remeshing for Three-Dimensional Compressible Flow Computations. *Journal of Computational Physics*, 103:269–285, 1992.

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

[183] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive Remeshing for Compressible Flow Computations. *Journal of Computational Physics*, 72:449–466, 1987.

[184] J. Perraud, D. Arnal, G. Casalis, J. P. Archambaud, and R. Donelli. Automatic Transition Predictions Using Simplified Methods. *AIAA Journal*, 47(11):2676–2684, 2009.

[185] J. Perraud, A. Séraudie, and F. Moens. Transition on a High-Lift Swept Wing in the European Project EUROLIFT. *Journal of Aircraft*, 41(5):1183–1190, 2004.

[186] P. O. Persson. Shock Capturing for High-Order Discontinuous Galerkin Simulation of Transient Flow Problems. In *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 2013.

[187] P. O. Persson, J. Bonet, and J. Peraire. Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198:1585–1595, 2009.

[188] P. O. Persson and J. Peraire. Sub-cell shock capturing for Discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2006.

[189] P. O. Persson and J. Peraire. Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In *47th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2008.

[190] P. O. Persson and G. Strang. A simple mesh generator in MATLAB. *SIAM review*, 46(2):329–345, 2004.

[191] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook. `https://archive.org/details/K_B_Petersen_and_M_S_Peders__The_Matrix_Cookbook`, 2012.

[192] S. Pirzadeh. Unstructured Viscous Grid Generation by the Advancing-Layers Method. *AIAA Journal*, 32(8):1735–1737, 1990.

[193] S. Pirzadeh. Three-dimensional unstructured viscous grids by the advancing-layers method. *AIAA Journal*, 34(1):43–49, 1996.

[194] S. Premasuthan, C. Liang, and A. Jameson. Computation Of Flows with Shocks Using Spectral Difference Scheme with Artificial Viscosity. In *48th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2010.

[195] S. Premasuthan, C. Liang, and A. Jameson. Computation of Flows With Shocks using the Spectral Difference method with Artificial Viscosity: Part I. *Computers & Fluids*, 98:111–121, 2013.

[196] A. Probst, R. Radespiel, and U. Rist. Linear-Stability-Based Transition Modeling for Aerodynamic Flow Simulations with a Near-Wall Reynolds-Stress Model. *AIAA Journal*, 50(2):416–428, 2012.

[197] R. Radespiel, K. Graage, and O. P. Brodersen. Transition Predictions Using Reynolds-Averaged Navier-Stokes and Linear Stability Analysis Methods. In *22nd AIAA Fluid Dynamics, Plasma Dynamics and Lasers Conference*, Honolulu, HI, 1991.

[198] D. Rajnarayan and P. Sturdza. Extensible Rapid Transition Prediction for Aircraft Conceptual Design. In *29th AIAA Applied Aerodynamics Conference*, Honolulu, HI, 2011.

[199] R. Rashad and D. W. Zingg. Toward High-Fidelity Aerodynamic Shape Optimization for Natural Laminar Flow. In *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 2013.

[200] H. L. Reed and W. S. Saric. Stability of three-dimensional boundary layers. *Annual Review of Fluid Mechanics*, 21:235–284, 1989.

[201] H. L. Reed, W. S. Saric, and D. Arnal. Linear Stability Theory. *Annual Review of Fluid Mechanics*, 28:389–428, 1996.

[202] N. H. Reed and T. R. Hill. Triangle mesh methods for the neutron transport equation. Technical Report LA2 UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.

[203] O. Reynolds. On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Philosophical Transactions of the Royal Society of London. A*, 186:123–164, 1895.

[204] X. Roca, A. Gargallo-Peiro, and J. Sarrate. Defining Quality Measures for High-Order Planar Triangles and Curved Mesh Generation. In *20th International Meshing Roundtable*, pages 365–383, Paris, 2012.

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

[206] J. A. Rossmanith, D. S. Bale, and R. J. LeVeque. A wave propagation algorithm for hyperbolic systems on curved manifolds. *Journal of Computational Physics*, 199(2):631–662, 2004.

[207] C. Rossow and L. Cambier. European Numerical Aerodynamics Simulation Systems. In E. H. Hirschel and E. Krause, editors, *Notes on Numerical Fluid Mechanics Vol. 100*, pages 189–208. Springer-Verlag, Berlin-Heidelberg, 2009.

[208] C. L. Rumsey. NASA Langley Research Center Turbulence Modeling Resource. `http://turbmodels.larc.nasa.gov`.

[209] C. L. Rumsey, M. Long, R. A. Stuever, and T. R. Wayman. Summary of the First AIAA CFD High Lift Prediction Workshop. In *49th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2011.

[210] P. Sagaut. *Large eddy simulation for incompressible flows*. Springer, Berlin-Heidelberg, 2006.

[211] O. Sahni, K. E. Jansen, M. S. Shephard, C. A. Taylor, and M. W. Beall. Adaptive boundary layer meshing for viscous flow simulations. *Engineering with Computers*, 24(3):267–285, 2008.

[212] W. S. Saric, H. L. Reed, and E. J. Kerschen. Boundary-Layer Receptivity to Freestream Disturbances. *Annual Review of Fluid Mechanics*, 34:291–319, 2002.

[213] W. S. Saric, H. L. Reed, and E. B. White. Stability and Transition of Three-Dimensional Boundary Layers. *Annual Review of Fluid Mechanics*, 35(1):413–440, 2003.

[214] H. Schlichting and K. Gersten. *Boundary layer theory*. Springer-Verlag, Berlin-Heidelberg, 2000.

[215] J. Schneider and B. Ewald. Integration of Linear Stability Methods into Navier-Stokes Solvers for Computation of Transonic Laminar Airfoils. In *12th AIAA Applied Aerodynamics Conference*, Colorado Springs, CO, 1994.

[216] F. Schultz-Grunow. New Frictional Resistance Law for Smooth Plate. Technical Report TM-986, NACA, 1940.

[217] R. Sevilla, O. Hassan, and K. Morgan. An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows. *Computer Methods in Applied Mechanics and Engineering*, 253:15–27, 2013.

[218] D. Sharov and K. Nakahashi. Hybrid prismatic/tetrahedral grid generation for viscous flow applications. *AIAA Journal*, 36(2):157–162, 1998.

[219] J.W. Slater. NPARC Alliance CFD Verification and Validation (Online). `http://www.grc.nasa.gov/WWW/wind/valid/validation.html`.

[220] J. Slotnick, A. Khodadoust, J. Alonso, D. L. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis. CFD Vision 2030 Study : A Path to Revolutionary Computational Aerosciences. Technical Report NASA-CR-2014-218178, NASA Langley, 2014.

[221] J. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–164, 1963.

[222] A. M. O. Smith and N. Gamberoni. Transition, pressure gradient and stability theory. Technical Report ES-26388, Douglas Aircraft Company, El Segundo, CA, 1956.

[223] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La recherche aérospatiale*, 1(1):5–21, 1994.

[224] C. G. Speziale. Turbulence Modeling in Noninertial Frames of Reference. *Theoretical and Computational Fluid Dynamics*, 1:3–19, 1989.

[225] J. Steelant and E. Dick. Modelling of bypass transition with conditioned Navier-Stokes equations coupled to an intermittency transport equation. *International Journal for Numerical Methods in Fluids*, 23:193–220, 1996.

[226] H. W. Stock. Airfoil Validation Using Coupled Navier–Stokes and $e^N$ Transition Prediction Methods. *Journal of Aircraft*, 39(1):51–58, 2002.

[227] H. W. Stock and E. Degenhart. A simplified $e^N$ method for transition prediction in two-dimensional, incompressible boundary layers. *Zeitschrift fuer Flugwissenschaften und Weltraumforschung*, 13:16–30, 1989.

[228] H. W. Stock and W. Haase. Determination of length scales in algebraic turbulence models for Navier-Stokes methods. *AIAA Journal*, 27(1):5–14, 1989.

[229] H. W. Stock and W. Haase. Feasibility Study of $e^N$ Transition Prediction in Navier-Stokes Methods for Airfoils. *AIAA journal*, 37(10):1187–1196, 1999.

[230] H. W. Stock and W. Haase. Navier–Stokes Airfoil Computations with $e^N$ Transition Prediction Including Transitional Flow Regions. *AIAA Journal*, 38(11):2059–2066, 2000.

[231] B. Strand. Summation by Parts for Finite difference Approximations for $d/dx$. *Journal of Computational Physics*, 110:47–67, 1994.

[232] P. Sturdza. *An aerodynamic design method for supersonic natural laminar flow aircraft*. PhD thesis, Stanford University, 2003.

[233] T. W. Swafford. Analytical Approximation of Two-Dimensional Separated Turbulent Boundary-Layer Velocity Profiles. *AIAA journal*, 21(6):923–926, 1983.

[234] A. Tam, D. Ait-ali-yahia, M. P. Robichaud, M. Moore, and V. Kozel. Anisotropic mesh adaptation for 3D flows on structured and unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 189:1205–1230, 2000.

[235] H. Tang and T. Tang. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 41(2):487–515, 2003.

[236] J. F. Thompson, B. K. Soni, and N. P. Weatherill, editors. *Handbook of Grid Generation*. CRC Press, Boca Raton, FL, 1999.

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

[238] T. Toulorge, J. Ponsin, J. Perraud, and F. Moens. Automatic Transition Prediction for RANS Computations Applied to a Generic High-Lift Wing. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 2007.

[239] A. Uranga. *Investigation of transition to turbulence at low Reynolds numbers using Implicit Large Eddy Simulations with a Discontinuous Galerkin method*. PhD thesis, Massachusetts Intitute of Technology, 2011.

[240] A. Uranga, P. O. Persson, M. Drela, and J. Peraire. Implicit Large Eddy Simulation of transition to turbulence at low Reynolds numbers using a Discontinuous Galerkin method. *International Journal for Numerical Methods in Engineering*, 87:232–261, 2011.

[241] C. P. van Dam. The aerodynamic design of multi-element high-lift systems for transport airplanes. *Progress in Aerospace Sciences*, 38(2):101–144, 2002.

[242] J. L. Van Ingen. A suggested semi-empirical method for the calculation of the boundary layer transition region. Technical Report V.T.H.-74, Delft University of Technology, Delft, The Netherlands, 1956.

[243] J. L. Van Ingen. The $e^N$ method for transition precition. Historical review of work at TU Delft. In *38th AIAA Fluid Dynamics Conference*, Seattle, WA, 2008.

[244] J. C. Vassberg, E. N. Tinoco, M. Mani, B. Rider, T. Zickuhr, D. W. Levy, O. P. Brodersen, B. Eisfeld, S. Crippa, R. A. Wahls, J. H. Morrison, D. J. Mavriplis, and M. Murayama. Summary of the Fourth AIAA CFD Drag Prediction Workshop. In *28th AIAA Applied Aerodynamics Conference*, Chicago, IL, 2010.

[245] P. E. Vincent, P. Castonguay, and A. Jameson. A New Class of High-Order Energy Stable Flux Reconstruction Schemes. *Journal of Scientific Computing*, 47(1):50–72, 2010.

[246] M. R. Visbal and D. V. Gaitonde. On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes. *Journal of Computational Physics*, 181(1):155–185, 2002.

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

[248] S. Wallin and A. V. Johansson. An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows. *Journal of Fluid Mechanics*, 403:89–132, 2000.

[249] D. K. Walters and J. H. Leylek. A New Model for Boundary Layer Transition Using a Single-Point RANS Approach. *Journal of Turbomachinery*, 126(1):193, 2004.

[250] L. Wang and D. J. Mavriplis. Adjoint-based h-p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *Journal of Computational Physics*, 228:7643–7661, 2009.

[251] Z. J. Wang, K. J. Fidkowski, 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. R. Visbal. High-Order CFD Methods : Current Status and Perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2012.

[252] Z. J. Wang and H. Gao. A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids. *Journal of Computational Physics*, 228(21):8161–8186, 2009.

[253] D. L. Whitfield. Analytical Description of the Complete Turbulent Boundary Layer Velocity Profile. In *11th AIAA Fluid and Plasma Dynamics Conference*, Seattle, WA, 1978.

[254] C. H. Whiting and K. E. Jansen. A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis. *International Journal for Numerical Methods in Fluids*, 35:93–116, 2001.

[255] K. Wieghardt and W. Tillman. On the Turbulent Friction Layer for Rising Pressure. Technical Report NACA-TM-1314, National Advisory Committee for Aeronautics, 1951.

[256] D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26(11):1299–1310, 1988.

[257] D. C. Wilcox. *Turbulence modelling for CFD*. DCW Industries Inc, La Cañada, CA, 1998.

[258] 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.

[259] M. Yano. *An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes*. PhD thesis, Massachusetts Institute of Technology, 2012.

[260] S. Zuccher, I. Shalaev, A. Tumin, and E. Reshotko. Optimal Disturbances in the Supersonic Boundary Layer Past a Sharp Cone. *AIAA Journal*, 45(2):366–373, 2007.