Université de Liège
Faculté des sciences appliquées

Development of a high-order interior penalty discontinuous Galerkin method for compressible turbulent flows

Application to Reynolds-averaged Navier-Stokes and large eddy simulations

Marcus Drosson

Thèse présentée en vue de l'obtention du
grade de docteur en sciences de l'ingénieur

Composition du jury
Prof. Olivier Léonard (président)
Prof. Jean-André Essers (promoteur)
Dr. Koen Hillewaert
Prof. Claus-Dieter Munz
Prof. Ludovic Noels
Prof. Jean-François Remacle
Prof. Vincent Terrapon

Janvier 2013

Acknowledgments

To begin, I would like to express my gratitude to my advisor, Prof. Jean-André Essers. Working under his supervision was a fruitful experience and greatly influenced my development as a researcher. In addition, I would like to thank my committee members for their criticism and feedback, which led to many improvements in this thesis.

I would like to particularly thank Geoffrey Deliège (University of Liège) and Koen Hillewaert (Cenaero). Their guidance, support and suggestions from the initial to the final level enabled me to develop an understanding of the subject. Without their help, this dissertation would not have been possible.

I am also indebted to my many colleagues at the University of Liège and at Cenaero. Without being exhaustive I would like to thank Didier Vigneron for the many instructive discussions about CFD, Bastien Gorissen who generated most of the meshes used in this thesis, Corentin de Wiart for the reference DNS data and his help concerning LES, as well as all the other colleagues - Sebastien Borguet, Jean-Philippe Thomas, Wenhai Du... - thanks to which this work has become an unforgettable time in my life.

At a more personal level, I would like to thank my parents, Véronique Mercier as well as Klaus Reinartz who all supported and encouraged me to bring this work to a successful end.

Finally, I gratefully acknowledge Cenaero for providing Argo, the CFD code that is at the origin of all numerical developments of this thesis, as well as the Belgian National Fund for Scientific Research FNRS for the financial support I have received.

Contents

Chapter 1

Introduction

1.1 Motivation and background

Computational Fluid Dynamics (CFD) are today a regular feature of many engineering applications. On the one hand the numerical simulations constitute a fast and cheap alternative to experimental prototypes at the early stages of design processes. On the other hand they offer a deep insight into phenomena which are difficultly observable in practice (insight of an oven, re-entry of space vehicles...). Thereby CFD simulations no longer restrict to classical aerodynamics such as aircraft design but more and more extend to a wide range of applications including e.g. chemical reactions, combustion, acoustics, multi-physics. The tremendous success of CFD over the last decade was enabled by the fast growing computer power and the development of efficient numerical algorithms. So diverse these applications may seem, they have one thing in common: most flows of practical interest are *turbulent*. It is the unsteady, irregular and seemingly random motion of the eddies that makes turbulence one of the biggest challenges in CFD. The difficulty is not only caused by the chaotic motion of the flow but also by the wide spectrum of wavenumbers that have to be taken into account.

Turbulence modelling. To overcome the complexity of turbulent flows a multitude of methodologies have been proposed in the literature. The most straightforward and conceptually easiest approach is the *Direct Numerical Simulation (DNS)*. Following this method all turbulent scales (from the microscopic dissipation range to the large eddies) are simulated and hence no modelling effort is needed. However given the extremely high requirement in terms of mesh resolution, DNS may be helpful to study some fundamental aspects of turbulence but remains difficult for most industrial applications at moderate or high Reynolds numbers. A completely different approach is followed by the class of *Reynolds-averaged Navier-Stokes (RANS)* models. Here the effect on the mean flow of all turbulent scales (from micro- to macroscales) is mod-

elled, which results in a regular, smooth velocity field. Today most turbulence models used in CFD belong to the RANS approach. As a consequence of the demand for higher accuracy and favored by the growing computer resources intense research has been devoted over the last decade to the development of *Large Eddy Simulation (LES)*. Situated between RANS and DNS only the small turbulent eddies are modelled whereas the large structures are resolved by the computational mesh. The price to pay is a non-negligible decrease of the mesh size compared to the one habitually used for RANS computations. Furthermore, LES simulations are inevitably unsteady. Both factors limit the use of LES to moderate Reynolds numbers. Because close to a solid wall the grid requirement of LES becomes comparable to the one of DNS, hybrid RANS-LES approaches recently have growing success. The idea consists in taking advantage of the relative strengths of each approach, i.e. the cost-effectiveness of RANS methods to compute (attached) boundary layers and the high accuracy of LES for flows in which large-scale unsteadiness is significant. Since the *Detached Eddy Simulation (DES)*, the first hybrid RANS-LES method, several newly developed variants have successively been applied to the prediction of massively separated flows.

Requirements on modern flow solvers. CFD is no longer used by only a handful of experts but has become a mainstream tool. Flow solvers should thus not only by efficient (in terms of memory and CPU time) but above all the numerical algorithms must be robust, avoiding as far as possible user-defined control parameters. Furthermore, besides an efficient parallelization in order to take advantage of modern computer architecture, the request for "*guaranteed mesh convergence*" such that results are less dependent on the engineer becomes more and more important.

A first example where CFD could (at least partially) replace expensive and time-consuming wind tunnel experiments is the design of new aircrafts. Related to the continuously growing cost of kerosene and in order the reach the stringent guidelines on air pollution, intense research is made in the fields of fuel consumption and noise emission. Both objectives are closely related to an accurate wake prediction which is an extremely challenging task in turbulence modelling. In particular, we generally observe a large spread of the computed drag estimation. The AIAA drag prediction workshops [106, 113] are good illustrations, suggesting that the poor quality of the results is caused not only by shortcomings of the current turbulence models but also by the inaccuracy of today's second-order methods with currently used grids.

Another example of computations requiring highly accurate schemes are LES simulations. Since turbulence implies a complex energy transfer between the large eddies and the dissipation range scales it is of utmost importance that the numerical scheme preserves the energy. High-order discretization methods characterized by a small numerical dissipation and dispersion are therefore

recommended. Similarly wave propagation problems such as aero-acoustics demand accurate solutions in the farfield and over a long time period. As weak signals that are often hardly distinguishable from numerical noise have to be captured, the discretization scheme should ideally be free of dispersion errors; otherwise information in the high-frequency range will be lost. The issue of highly accurate methods for aero-acoustic problems has been discussed by Lighthill [114].

The above examples are certainly not exhaustive but clearly illustrate the demand for higher accuracy. As will be shown in this thesis and as reported by other authors, high-order accurate discretization schemes require significantly less degrees of freedom than low-order methods to reach the same level of accuracy. For real geometries, the high mesh resolution needed by today's second-order finite volume schemes causes an excessive computational cost. Accordingly, there is a consensus that despite impressive results obtained with current solvers, grid converged solutions are still out of reach for many applications, e.g. vortex dominated flows. Because high-Reynolds number flows furthermore imply highly stretched anisotropic meshes, which are hard to construct on complex geometries, a key feature of future flow solvers is the possibility to provide accurate results even on poor quality unstructured meshes.

High-order spatial discretization schemes: DGFEM. On the one hand, due the increasing complexity of the simulations there is a clear demand for high-order accurate methods. On the other hand, finite volume (FV) methods which are the most popular approach within commercial CFD solvers are at best second order accurate. Although different attempts have been made to design high-order FV schemes, the increasing stencil size and difficulties to define stable algorithms on highly stretched unstructured meshes have limited their application for industrial problems. Other discretization approaches such as finite difference (FD) or spectral methods are quite efficient computationally. In addition, they more easily attain the desired level of accuracy. However, their inability to handle complex geometries and unstructured meshes prevented their spread into commercial CFD. Accordingly, FD and spectral methods generally restrict to rather academic applications. For an extensive review of high-order methods we refer to [56, 179].

Recently, the ADIGMA project [105] has identified two methods showing the highest potential for efficient high-order discretisations: the *discontinuous Galerkin finite element methods* (DGFEM) [140] and the *continuous residual distribution* (CRD) schemes [1, 46]. This thesis is dedicated to the application of the discontinuous Galerkin / Interior Penalty (IP) formulation for turbulent flows.

Despite the success of DGFEM in solving advection problems, the method was relatively neglected over more than 30 years for diffusion problems. The

breakthrough finally came at the end of the 1990's when the two approaches were combined in order to extend DGFEM to the Navier-Stokes equations. Since then, a lot of work has been done to develop a compact and robust discretization of elliptic operators. In this thesis, the viscous terms are handled using an interior penalty (IP) formulation. As FD and spectral methods, the DGFEM-IP approach allows to easily reach a high level of accuracy by simply increasing the order of the polynomial approximation p. Thereby, the discretization scheme conserves its local character and furthermore naturally applies to unstructured meshes. Although DG methods have recently been identified as having the biggest potential for future flow solvers, further research is needed to improve the computational efficiency and the robustness of these schemes. Both issues are faced in this thesis.

1.2 Main achievements of this thesis

This work contributes to the development of a high-order discontinuous Galerkin solver for the simulation of compressible, turbulent flows around complex three-dimensional geometries. Platform for the numerical developments is the "Argo" software package, a research code which arose from a collaboration between the Belgian research center Cenaero and the department of Applied Mechanics and Mathematics (MEMA) at the Université catholique de Louvain. Before this thesis, Argo has successively been applied to laminar flows. The two- and three-dimensional Navier-Stokes equations had been implemented using a DGFEM-IP approach and the discretized system of equations was solved by a (parallel) Jacobian-free Newton-GMRES solver. Furthermore, different time integration schemes were available for unsteady simulations. The focus of this work is to extend Argo to turbulent applications. Thereby, the present thesis analyses for the first time the use of the interior penalty discretization to solve a system of RANS equations.

The major part of the efforts has been devoted to Reynolds-averaged Navier-Stokes methods. To this end the one-equation Spalart-Allmaras turbulence model [165] has been implemented. Thereby, the whole system of RANS equations is handled as a unique block which enables a quadratic convergence rate at the last stages of the non-linear Newton solver. Although the DGFEM-IP approach is known for its excellent dispersion and dissipation properties - and hence probably better suited to more direct turbulence modelling approaches (e.g. LES or DES) - an advantage over standard FV methods is also to be expected for RANS models. The reasons to investigate RANS models are

- they provide satisfying results for a number of applications that are not subject to flow separation;
- they can be used to generate comparative results for future LES compu-

tations;

- hybrid approaches such as DES are based on RANS models. Even RANS and DES are very different from a conceptual point of view, Spalart *et al.* [163] have shown that a minor modification of the S-A model results in a hybrid RANS-LES computation.

Stability issues. Whilst high-order accurate discretization schemes require less degrees of freedom to reach a given level of accuracy, particular care must be taken especially in the case of turbulent flows. This can be explained by the sharp gradients (e.g. at the outer boundary layer) on the one hand, and the coarse meshes typically used by DGFEM on the other hand. Given the low numerical dissipation of the scheme, Gibbs oscillations may cause the breakdown of the iterative solving procedure. A major contribution of this thesis is therefore related to improving the stability of the discrete method.

A first cause of instabilities is due to the definition of most RANS models which become unstable as negative values of the turbulent viscosity are encountered. For that reason, we compare several *modifications of the S-A turbulence model* to improve the robustness of the high-order RANS solver in under-resolved mesh regions. Numerical simulations of a turbulent flat plate and a NACA 0012 aerofoil illustrate the influence of these modifications.

Within the same context, we also investigate a *hybrid scheme* that combines distinct polynomial interpolations of the mean flow equations and the turbulence model.

Moreover, the coercivity of the interior penalty method is investigated. Compared to other popular DG formulations like the second Bassi-Rebay scheme, the simpler IP formulation is computationally less expensive, reducing by up to 30 % the computational effort to evaluate the non-linear residual. However, the method suffers from the dependency on a seemingly more or less arbitrary (i.e. "large enough") stability parameter. The latter is probably one of the main reasons for the relative unpopularity of the IP formulation. Whereas in the past, nearly optimal values of the *penalty parameter* have been derived for a Poisson equation with constant viscosity, we propose a generalization of this parameter in order to take into account mesh anisotropy on the one hand, and large variations of the diffusion coefficient on the other hand.

Finally, a closer look is taken to the *role of the transpose term* that introduces a strong coupling between the continuity equation and the turbulence model. Because this coupling was found to deteriorate the stability of the method, a modification of the transpose term is proposed in order to make the IP formulation suitable for RANS applications.

Computational efficiency. One of the major drawbacks of DGFEM is the computational cost. In the past, Argo used an approximate Jacobian matrix for the BILU preconditioner of the GMRES solver. Despite a satisfying

performance for laminar flows, the simplification turned out to be inadequate for turbulent computations. We therefore have extended the formerly approximate linearization of the (laminar) Navier-Stokes equations to the exact linearization of the coupled RANS system.

Turbulent boundary layer resolution. In use for decades, the mesh requirements for a resolved FV boundary layer computations are well known. For instance there is a general agreement that the first node should lie within the viscous sublayer at $y^+(1) \approx 1$. However, as the polynomial degree p of the approximation changes, the question arises in which measure this affects the required mesh size. Based on multiple grid convergence analyses this thesis provides clear guidelines on the choice for boundary layer resolution for practical applications. Besides grid spacings and stretchings as a function of interpolation order and element type, several other aspects like the effect of curved boundaries or the optimal position of the interpolation nodes are discussed. As an example, our numerical studies have shown that a proper choice of the inner-element nodes allows to improve the accuracy of the computed shear friction, whilst conserving the total number of unknowns. Given the close relation between RANS and DES models, these results provide precious information for future detached eddy simulations.

Large eddy simulation. The last part of this thesis is devoted to large eddy simulations. Therefore two subgrid scale models - the Smagorinsky and the WALE model - have been implemented into Argo. Because turbulence modelling is limited to the small (subgrid) scales, LES is expected to be more accurate and reliable than RANS models for flows characterized by large, unsteady structures. A further improvement is achieved by the so called Variational Multiscale Method (VMS). By introducing an explicit filter operation this variant provides a much better control of the turbulent viscosity than standard LES. Since the discretization of the large structures remains consistent with DNS, VMS significantly improves the accuracy for e.g. flows with laminar-to-turbulent transition. In addition, because with increasing p each computing cell contains a large number of degrees of freedom, filters can be defined elementwise. In contrast to widely used low order methods (finite volume,...), the filtering conserves thus the local character of the discretization scheme. Elementwise inversion of a global operator, such as a filter, in addition to low dispersion and dissipation errors make DGFEM-IP ideally suited for parallel LES computations.

One objective of this thesis was again to determine the resolution capability of high-order polynomials. To this end different convergence analyses of homogeneous isotropic turbulence and the Taylor-Green vortex are performed. While attention is again focused on resolution requirements, different subgrid filters are compared and the choice of the Smagorinsky constant in the case of

filtered LES is briefly examined.

1.3 Outline

This thesis is organized as follows.

Chapter 2 recalls the general *governing equations* of compressible tridimensional flows. After a short summary of the Navier-Stokes equations (laminar case), the focus lies on turbulence modelling. The approaches followed in this work are Reynolds-averaged Navier-Stokes (RANS) on the one hand, and large eddy simulation (LES) on the other. Given the vast literature on turbulence modelling, only the final set of equations together with the corresponding turbulence/subgrid scale models are presented. Note that no effort has been made neither to improve existing nor to develop new turbulence models. The aim was rather to adapt these models to the particular features of a high-order DGFEM discretization.

In the case of the RANS system turbulence closure is achieved using the one-equation Spalart-Allmaras (S-A) model. Given the stability issues caused by negative values of the S-A variable, several modifications of the model are discussed.

For the LES computations the static Smagorinsky model and the WALE model have been tested. After a short description of these models, two basic filters needed for the variational multiscale approach are introduced. Both filters rely on a polynomial projection between distinct function spaces and are thus well suited for the DGFEM discretization.

Whilst chapter 2 is devoted to the physical aspects of turbulence modelling, **chapter 3** focuses on the *spatial discretization*. Starting with a brief overview of the historical evolution of discontinuous Galerkin methods, their principal advantages/drawbacks are compared to standard discretization schemes. Being at the origin of all numerical developments realized within this thesis, the IP-DGFEM formulation is derived in details. A major contribution are the stability and performance improvements achieved by i) a modification of the transpose term and ii) several new proposals for the penalty parameter. Different numerical examples illustrate these findings.

Time integration and the *resolution of (non-linear) systems* of (differential) equations are the subject of **chapter 4**. Whereas large eddy simulations are inevitably time-dependent, most RANS applications (and all examples treated here) aim for a steady solution. In the first part of this chapter different explicit and implicit time integration schemes which are used for our LES computations are shortly described. The most important are the Euler, the Runge-Kutta and the three-point backward schemes. Next iterative algorithms to solve the (steady) RANS system are outlined in the second part of the chapter. In this work we have used the available matrix-free Newton-GMRES solver. The latter was found to be highly efficient for large systems

of equations and is commonly used in CFD. In order to obtain an optimal order of convergence and to increase the stability of the iterative scheme, the Navier-Stokes equations and turbulence model are solved simultaneously as a single block. Although the Newton-GMRES solver does not require the explicit construction of the Jacobian matrix, at least an approximate linearisation is needed for preconditioning. Taking into account the particularities of the IP discretization which behaves as a structured method inside each computing cell, the Jacobian can be build by adding precomputed parametric subblocks with on-the-fly calculated weights. Thereby the large number of unknowns per element leads to computationally extremely efficient matrix-vector operations. Finally the influence of an approximate linearization and of a frozen Jacobian are investigated.

In contrast to the previous chapters which discuss different numerical aspects related to either turbulence modelling, spatial discretization or solving procedures, **chapter 5** and **chapter 6** present *several RANS applications*. In chapter 5 we focus on turbulent boundary layer resolution. The principal aim is to provide detailed information on the allowable element height in function of e.g. element type and polynomial order. Further topics of interest are the influence of curved boundary representation, the role of the penalty parameter or the optimal position of the interpolation nodes. To this end, multiple convergence studies are performed for the flow along a turbulent flat plate and a NACA 0012 aerofoil. In chapter 6 we consider two more complex applications, namely the turbulent flow around the L1T2 three-element aerofoil at take-off configuration as well as a 3D highly loaded compressor cascade. On the one hand these examples are considered as validation test cases for our RANS model, on the other hand they demonstrate the applicability of the high-order RANS solver to "industrial" applications.

In **chapter 7** two *large eddy simulations* of the Taylor-Green vortex and the decay of homogeneous isotropic turbulence are presented. This last chapter should be regarded as a first step towards the use of discontinuous Galerkin methods for large eddy simulations. Besides a validation study for the Smagorinsky and the WALE subgrid scale model, the resolution requirements of resolved LES are determined. Furthermore, different subgrid filters are compared and the choice of the Smagorinsky constant in case of filtered LES is briefly examined.

Finally, the *conclusions* of this thesis and *perspectives* for future work are summarized in **chapter 8**.

Chapter 2

Governing equations

Contents

This chapter presents the general governing equations of compressible tridimensional laminar and turbulent flows. It begins with a brief review of the Navier-Stokes equations which are recalled in section 2.2.

Next, the concept of turbulence modelling is introduced in section 2.3. After a short summary of Reynolds and Favre averaging in section 2.3.1, the Reynolds-averaged Navier-Stokes (RANS) equations are derived and the main corresponding closure assumptions are discussed (see section 2.3.2).

In section 2.3.3 the one-equation Spalart-Allmaras (S-A) turbulence model is presented. Stability issues related to negative values of the turbulence working variable are considered and different modifications to the Spalart-Allmaras model are proposed in section 2.3.4.

Finally the concept of the Variational MultiScale method (VMS), a variant of standard Large Eddy Simulation (LES), is introduced in section 2.4, and the related SGS turbulence models and filtering techniques are presented in section 2.4.1 and 2.4.2, respectively.

2.1 The turbulence problem

While in principle the compressible Navier-Stokes equations (see section 2.2) govern both, laminar and turbulent flows, the "turbulence problem" remains one of the most challenging research fields in Computational Fluid Dynamics (CFD) [138, 167, 182]. The difficulty in simulating turbulent flows resides in the wide spectrum of length and time scales that are involved. According to Kolmogorov [101] the ratio between the energy-bearing scales l (largest eddies often referred to as integral scale) and the Kolmogorov microscales η is $\frac{l}{\eta} \sim Re_t^{3/4}$ where $Re_t = k^{1/2} l/\nu$ denotes the turbulent Reynolds number with k the turbulent energy and ν the kinematic viscosity. A rough estimate of the grid requirements indicates that approximately $\mathcal{O}(Re_t^{3/4})$ grid cells are necessary per space direction to resolve all turbulent scales properly. As turbulence is mainly a three-dimensional, unsteady phenomenon, the total computational cost roughly scales as $\sim Re_t^3$. Clearly, the requirements for the Direct Numerical Simulation (DNS) of aerodynamic flows at moderate and high[1] Reynolds numbers exceeds by far current computer resources.

To overcome the prohibitive high cost of DNS, different approaches based on statistical methods have been proposed to take into account turbulent effects. The strongly rotational character of the velocity field and the intrinsic three-dimensionality, time-dependency and random fluctuations make it impossible to define a simple analytic theory. From a mathematical point of view, turbulence develops as an instability of laminar flows, due to a complex interaction between non-linear inertial terms and viscous terms. Depending on the degree of modelling, we distinguish between *Reynolds-averaged Navier-Stokes* (RANS) and *Large Eddy Simulation* (LES). Although these strategies are based on different concepts, their common feature is to increase the diffusivity of the Navier-Stokes equations. Because the principal role of turbulence is to dissipate kinetic energy, a supplementary viscous term is added to the Navier-Stokes equations, which generally is computed together with its laminar counterpart. Thereby, the repartition of the turbulent diffusion to the different spatial scales is extremely important.

2.2 Compressible Navier-Stokes equations

In this section we recall the Navier-Stokes equations, which describe the conservation of mass, momentum and energy of viscous continuum flows. In the absence of body forces and volume supply of energy, these equations can be written in conservative form for a compressible time-dependent flow as follows

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{F}}^c(\mathbf{U}) = \nabla \cdot \boldsymbol{\mathcal{F}}^v(\mathbf{U}, \nabla \mathbf{U}), \tag{2.1}$$

[1] Reynolds numbers of several million are frequently encountered in practical applications such as airplanes,...

where $\mathbf{U} = [\rho, \rho u, \rho v, \rho w, \rho E]^T$ is the vector of conservative variables. Here, ρ denotes the density of the fluid, $\mathbf{u} = [u, v, w]^T$ is the velocity vector and E is the total energy per unit mass which is related to the specific internal energy e by $E = e + \frac{1}{2}\|\mathbf{u}\|^2$.

A detailed expression of the inviscid flux tensor $\boldsymbol{\mathcal{F}}^c = \left(\boldsymbol{\mathcal{F}}^c_x, \boldsymbol{\mathcal{F}}^c_y, \boldsymbol{\mathcal{F}}^c_z\right)$ and the viscous flux tensor $\boldsymbol{\mathcal{F}}^v = \left(\boldsymbol{\mathcal{F}}^v_x, \boldsymbol{\mathcal{F}}^v_y, \boldsymbol{\mathcal{F}}^v_z\right)$ in x, y and z directions is given by

$$\boldsymbol{\mathcal{F}}^c_x = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{bmatrix}, \quad \boldsymbol{\mathcal{F}}^c_y = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{bmatrix}, \quad \boldsymbol{\mathcal{F}}^c_z = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \end{bmatrix} \tag{2.2}$$

and

$$\boldsymbol{\mathcal{F}}^v_x = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x \end{bmatrix}, \quad \boldsymbol{\mathcal{F}}^v_y = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y \end{bmatrix},$$

$$\boldsymbol{\mathcal{F}}^v_z = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z \end{bmatrix}, \tag{2.3}$$

with p the static pressure, $\mathbf{q}$ the heat flux and $\boldsymbol{\tau}$ the stress tensor. As the conservation laws (2.1) solely are not sufficient to describe the behaviour of the fluid, additional constitutive equations are necessary.

In this work, the thermally perfect gas assumption is made and the static pressure can be determined by the equation of state which reads

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

where T is the temperature and R the specific gas constant. R has a unique value for every gas and is related to the universal gas constant $\bar{R} = 8.314472$ J/mol K by the relation $R = \frac{\bar{R}}{M}$, where M denotes the molar mass of the gas. Furthermore, the specific internal energy e and the specific internal enthalpy h are homogeneous functions of the temperature

$$de = c_v dT \qquad \text{and} \qquad dh = c_p dT. \tag{2.5}$$

c_p and c_v are the specific heat capacities at constant pressure and constant volume respectively. They are related to the specific gas constant R and the ratio of specific heats γ, or adiabatic exponent, by

$$R = c_p - c_v, \tag{2.6}$$

$$\gamma = \frac{c_p}{c_v}. \tag{2.7}$$

In our applications, calorically perfect gases are considered, which means that c_p and c_v are constant, so that (2.5) can be rewritten as

$$e = c_v T, \qquad h = c_p T. \tag{2.8}$$

For air at low speeds and ambiant temperatures, γ equals 1.4. Combining the expressions (2.4) and (2.8), the static pressure can be expressed in terms of the conservative variables $\mathbf{U}$

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\frac{\|\rho \mathbf{u}\|^2}{\rho}\right). \tag{2.9}$$

Since this work is restricted to Newtonian fluids for which Stokes' hypothesis is valid, the stress tensor $\boldsymbol{\tau}$ appearing in the viscous (diffusive) fluxes $\boldsymbol{\mathcal{F}}^v$ is symmetric and can be written as a linear function of the velocity gradients

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{2.10}$$

with μ the dynamic viscosity and δ_{ij} the Kronecker delta. For completeness, it should be mentioned that the variation of μ as a function of temperature can be described by the semi-empirical formula of Sutherland

$$\frac{\mu(T)}{\mu_0} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}} \frac{T_0 + S}{T + S}, \tag{2.11}$$

where the subscript "0" denotes a reference state and S is a constant which depends on the gas under consideration. However, in all computations presented in this work, a constant dynamic viscosity will be assumed.

Finally, the heat flux $\mathbf{q}$ which appears in the energy equation is modelled according to Fourier's law

$$q_j = -\lambda \frac{\partial T}{\partial x_j}. \tag{2.12}$$

Based on empiricism, the thermal conductivity λ can be linked to the dynamic viscosity through the non-dimensional Prandtl number $Pr = \frac{\mu c_p}{\lambda}$. For gases, it hardly varies with a change in temperature and as we are mainly interested in air flows, a constant Prandtl number of 0.72 is assumed. This value also applies to many other gases.

2.3 Reynolds-Averaged Navier-Stokes equations and closure models

2.3.1 Reynolds and Favre averaging

The concept of averaging to describe incompressible flows has been introduced for the first time by Reynolds [142] in 1895. Given the difficulty to characterize

a turbulent flow at all points in time and space, he proposed to decompose every quantity $\phi(\mathbf{x}, t)$ into a mean $\bar{\phi}(\mathbf{x})$ and a fluctuating part $\phi'(\mathbf{x}, t)$, i.e.

$$\phi = \bar{\phi} + \phi'. \tag{2.13}$$

Formally, the mean is defined by

$$\bar{\phi} \equiv \int_{-\infty}^{+\infty} \phi^* P(\phi^*) d\phi^*, \tag{2.14}$$

where $P(\phi^*)$ is the so-called *probability density function* and ϕ^* denotes an integration variable. The probability that ϕ lies between $\phi^* - \frac{1}{2}d\phi^*$ and $\phi^* + \frac{1}{2}d\phi^*$ is given by $P(\phi^*)d\phi^*$. Similarly to (2.14) any function of ϕ can be averaged as:

$$\bar{f(\phi)} \equiv \int_{-\infty}^{+\infty} f(\phi^*) P(\phi^*) d\phi^*. \tag{2.15}$$

Finally, the *Reynolds equations* for an incompressible fluid are obtained by using the Reynolds decomposition (2.13) for all flow variables and by applying the above averaging to the continuity, momentum and energy equations (2.1).

Whereas for incompressible fluids, Reynolds averaging leads to a set of equations for the mean properties that is formally almost identical to the Navier-Stokes equations, they take a very complex form for compressible fluids. The reason are density fluctuations ρ' that introduce multiple additional terms. As these terms must be modelled in function of known quantities, it is convenient to simplify the problem by using a density-weighted average $\tilde{\phi}$, known as Favre's average [59], which is defined by

$$\tilde{\phi} \equiv \frac{\overline{\rho\phi}}{\bar{\rho}}. \tag{2.16}$$

Similar to (2.13), the decomposition in terms of mean and fluctuating parts now reads

$$\phi = \tilde{\phi} + \phi'', \tag{2.17}$$

where the double primes denote fluctuations with respect to the density-weighted average. Substituting (2.17) into the compressible Navier-Stokes equations (2.1), the averaging procedure followed by some mathematical op-

erations gives

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0, \tag{2.18a}$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{u}_i) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{u}_i + \bar{p}\delta_{ij}) = \frac{\partial}{\partial x_j}(\bar{\tau}_{ij} - \overline{\rho u_j'' u_i''}), \tag{2.18b}$$

$$\begin{aligned}\frac{\partial}{\partial t}\left[\bar{\rho}\tilde{E} + \frac{1}{2}\overline{\rho u_i'' u_i''}\right] = &-\frac{\partial}{\partial x_j}\left[\bar{\rho}\tilde{u}_j\left(\tilde{E} + \frac{\bar{p}}{\bar{\rho}}\right) + \tilde{u}_j\frac{1}{2}\overline{\rho u_i'' u_i''}\right] \\ &+ \frac{\partial}{\partial x_j}\left[-\bar{q}_j - \overline{\rho u_j'' h''} + \overline{\tau_{ji}u_i''} - \overline{\rho u_j''\frac{1}{2}u_i'' u_i''}\right] \\ &+ \frac{\partial}{\partial x_j}\left[\tilde{u}_i\left(\bar{\tau}_{ij} - \overline{\rho u_i'' u_j''}\right)\right].\end{aligned} \tag{2.18c}$$

This set of equations is called *mass-weighted Reynolds-averaged Navier-Stokes* (RANS) equations. So far, no modelling assumption has been made, which means that the system (2.18) is exact. Note also that all variables are "Favre averaged", except density and pressure which are always expressed as Reynolds averages.

2.3.2 Closure Approach

Comparing (2.1) and (2.18), we notice five additional quantities that have been introduced by the averaging procedure, namely $\overline{\rho u_i'' u_j''}$, $\frac{1}{2}\overline{\rho u_i'' u_i''}$, $\overline{\rho u_j'' h''}$, $\overline{\tau_{ij}u_i''}$ and $\overline{\rho u_j''\frac{1}{2}u_i'' u_i''}$. As they depend on the turbulent fluctuations, these terms are unknown and the resulting system is not fully specified. To close the set of RANS equations, auxiliary relationships that model them in terms of the mean flow variables must be found.

A closer look shows that the continuity equations are formally identical. As for the momentum equations, one extra term appears. The Reynolds stress tensor

$$\tau_{ij}^R \equiv -\overline{\rho u_i'' u_j''} = -\bar{\rho}\,\widetilde{u_i'' u_j''} \tag{2.19}$$

results from the non-linearity of the inviscid fluxes (2.2) and causes momentum transfer between mean flow and turbulent fluctuations. It can be interpreted as an apparent stress through the flow. To relate τ_{ij}^R to known flow properties, a range of models which can roughly be divided into two categories have been proposed. The first group of "rather simple" models is based on the widely applied Boussinesq hypothesis [16] which states that the Reynolds stress can be computed as the product of an eddy viscosity μ_t and the mean strain-rate tensor, i.e.

$$\tau_{ij}^R = -\overline{\rho u_i'' u_j''} = \mu_t\left(2\tilde{s}_{ij} - \frac{2}{3}\tilde{s}_{kk}\delta_{ij}\right) - \frac{2}{3}\bar{\rho}k\delta_{ij}, \tag{2.20}$$

with $\tilde{s}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right)$ being the mean strain-rate tensor and $\bar{\rho}k \equiv \frac{1}{2}\overline{\rho u_i'' u_i''}$ the turbulent kinetic energy per unit volume. The addition of $-2\bar{\rho}k/3$ in

(2.20) guarantees the correct trace of the Reynolds stress, which yields $\tau_{ii}^R = -\overline{\rho u_i'' u_i''} = -2\bar{\rho}k$. A detailed description of how the eddy viscosity is calculated is given in the next section.

While the Boussinesq eddy-viscosity approximation has successfully been applied to a wide range of flows - e.g. boundary layer, channel flow, mixing layer, round jet - it fails for flows with sudden changes in the mean strain-rate. The reason is the intrinsic hypothesis that turbulent eddies act on the momentum in almost the same manner as laminar viscous stresses caused by molecular interactions. However in a simple laminar shear flow, molecular time scales related to the mean free path are significantly shorter than shear time scales. Hence, any variation in the molecular interactions is instantaneously felt by the laminar stresses, which justifies the definition of a local (laminar) viscosity.

In contrast, turbulent eddies interact over long distances for which the mean motion of the flow may change. As a consequence, turbulent and shear time scales become comparable and the assumption of a local relationship between Reynolds stresses and mean strain-rate as stated by (2.20) is no longer justified [54,138]. Furthermore, in many complex flows, turbulent stresses and mean rate of deformation are not aligned and the underlying assumption of an isotropic viscosity μ_t fails [175].

To overcome the limitations of the Boussinesq assumption, more sophisticated models known as *Reynolds stress models* (RSM) have been developed (e.g. Launder [109]). Instead of a simple algebraic relationship between τ_{ij}^R and $\tilde{s}_{ij}$, transport equations for the six independent components of the Reynolds stress tensor are solved. Thereto, successively higher moments of the Navier-Stokes equations are written. Because this procedure is of purely mathematical nature but does not describe any new physical concept, every higher moment introduces additional unknowns that must be modeled. Given the computational expense and the stiffness of the RSM model, Rodi [145] proposed an approximation for the "turbulent transport terms" that reduces the model to a set of algebraic equations. The resulting *algebraic stress model* (ASM) implicitly determines the Reynolds stresses as a function of the turbulent kinetic energy k, the dissipation rate ϵ and the mean velocity gradients. The first explicit solution has been derived by Pope [137]. Over the years different *explicit algebraic stress models* (EASM) have been developed by several independent groups. Despite the intrinsic simplicity of the algebraic formulation, EASM models can be successful in calculating flows with strong mean rotation, a feature where turbulent viscosity models fail [138].

Besides the Reynolds stress tensor, closure terms also appear in the energy equation. In general, for a compressible flow, one must provide models for four additional quantities: the turbulent kinetic energy $\bar{\rho}k \equiv \frac{1}{2}\overline{\rho u_i'' u_i''}$, (i); the turbulent heat flux vector $\overline{\rho u_j'' h''}$, (ii); the molecular diffusion term $\overline{\tau_{ij} u_i''}$, (iii); and the turbulent transport term $\overline{\rho u_j'' \frac{1}{2} u_i'' u_i''}$, (iv). According to Wilcox [182],

the effects of compressibility on the Reynolds-averaged energy equation are small whenever $k \ll \tilde{h}$ which is the case for most flows up to the supersonic range. If the turbulent kinetic energy k is not available (e.g. in algebraic models or some one-equation turbulence models), it is therefore common practice to simply ignore the terms (i), (iii) and (iv) (e.g. [18,71,107,129]). The same simplification is made by some researchers using higher-order models. As a consequence, only two closure approximations are required: for the Reynolds stress tensor and for the turbulent heat flux vector. In contrast, at hypersonic speeds, the effects of compressibility become important and models must be found for all terms (i) to (iv). For further details about the simulation of hypersonic turbulent flows, we refer to [154].

Neglecting k in (2.20), the Reynolds stress tensor used in this work reads

$$\tau_{ij}^R = -\overline{\rho u_i'' u_j''} = \mu_t \left(2\tilde{s}_{ij} - \frac{2}{3}\tilde{s}_{kk}\delta_{ij} \right). \tag{2.21}$$

Following the Reynolds analogy, the turbulent heat flux vector is expressed in terms of the mean temperature gradient as

$$q_j^t \equiv \overline{\rho u_j'' h''} = -\frac{\mu_t c_p}{Pr_t}\frac{\partial \tilde{T}}{\partial x_j}, \tag{2.22}$$

where Pr_t denotes the turbulent Prandtl number which is supposed to be constant and is set to $Pr_t = 0.9$ in the present work. Likewise in a laminar flow, it is related to the turbulent thermal conductivity λ_t by $\lambda_t = c_p \mu_t / Pr_t$.

Finally, by substituting (2.21) and (2.22) into (2.18), the RANS equations applied in this work take the same form as the compressible Navier-Stokes equations (2.1). The only difference is that (i) the working variables $\mathbf{U}$ now represent Favre-averaged mean quantities $\tilde{\mathbf{U}} = [\bar{\rho}, \tilde{\rho u}, \tilde{\rho v}, \tilde{\rho w}, \tilde{\rho E}]^T$; and (ii) the laminar stress tensor $\boldsymbol{\tau}$ and the laminar heat flux $\mathbf{q}$ are replaced by

$$\tau_{ij}^{eff} \equiv \bar{\tau}_{ij} + \tau_{ij}^R = (\mu + \mu_t)\left(2\tilde{s}_{ij} - \frac{2}{3}\tilde{s}_{kk}\delta_{ij} \right), \tag{2.23}$$

$$q_j^{eff} \equiv \bar{q}_j + q_j^t = -(\lambda + \lambda_t)\frac{\partial \tilde{T}}{\partial x_j}, \tag{2.24}$$

respectively. The effective stress tensor and the effective heat flux combine laminar and turbulent effects. Note that at this point, μ_t and λ_t are still unknown; their calculation is subject of the following sections.

For the sake of simplicity and to improve the readability of the equations we henceforth omit the $\bar{(\cdot)}$ and $\tilde{(\cdot)}$ notation. All flow variables used in the following chapters refer to appropriately averaged mean flow quantities.

2.3.3 One-equation Spalart-Allmaras turbulence model

Within the scope of this thesis, turbulence closure is accomplished by the Spalart-Allmaras model [164] (referred to as "S-A model" throughout). Based on empiricism, dimensional analysis and selective dependence on molecular viscosity, the S-A model was primarily developed for aerodynamic flows like those around aerofoils and wings. This new model was motivated on the one hand by the lack of accuracy and the incompleteness of algebraic models, and on the other hand by the complexity and the numerical stiffness of two-equation models. Unlike early one-equation models which are based on the turbulent kinetic energy k, and which require an explicit specification of the length scale, the S-A model describes a single transport equation for the eddy viscosity ν_t. The mixing length is inherently defined in terms of available variables, and the model is thus complete. Furthermore, the S-A model involves only local information which makes it particularly suitable for unstructured grids. Accurate results have been obtained for attached and mildly separated aerodynamic flows [30, 72, 150].

Since its first publication in 1992, a number of changes to improve the robustness and convergence properties of the SA-model have been proposed. These include new constants for the trip term advocated by Spalart and Allmaras [165] (not used here) and a modified source term reported by Ashford [6]. For a detailed description of the SA-model used in this work, we refer to [71].

In the full form of the compressible S-A model, a transport equation for the intermediate working variable $\tilde{\mu}$ is defined as follows:

$$\frac{\partial \tilde{\mu}}{\partial t} + \frac{\partial (u_j \tilde{\mu})}{\partial x_j} - \frac{1}{\sigma}\frac{\partial}{\partial x_j}\left[(\mu + \tilde{\mu})\frac{\partial (\frac{\tilde{\mu}}{\rho})}{\partial x_j}\right] = Q^{SA}, \tag{2.25}$$

where the source term is given by

$$\begin{aligned} Q^{SA} = &+ \underbrace{c_{b1}(1 - f_{t2})\tilde{\omega}\tilde{\mu}}_{production} - \underbrace{\rho\left(c_{w1} f_w - \frac{c_{b1}}{c_\kappa^2} f_{t2}\right)\left(\frac{\tilde{\mu}}{\rho d}\right)^2}_{destruction} \\ &+ \underbrace{\frac{1}{\sigma}\left[\rho c_{b2}\frac{\partial (\frac{\tilde{\mu}}{\rho})}{\partial x_j}\frac{\partial (\frac{\tilde{\mu}}{\rho})}{\partial x_j}\right]}_{diffusion} + \underbrace{f_{t1}\|\mathbf{u} - \mathbf{u}_t\|^2}_{transition}. \end{aligned} \tag{2.26}$$

Then, the dynamic turbulent/eddy viscosity μ_t is calculated from $\tilde{\mu}$ via

$$\mu_t = \tilde{\mu} f_{v1}, \qquad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad \chi = \frac{\tilde{\mu}}{\mu}. \tag{2.27}$$

The production term is expressed in function of the modified vorticity $\tilde{\omega}$

$$\tilde{\omega} = \omega f_{v3} + \frac{\tilde{\mu}}{\rho c_\kappa^2 d^2} f_{v2}, \qquad \omega = \sqrt{2\Omega_{ij}\Omega_{ij}}. \tag{2.28}$$

Here, $\Omega_{ij} = \frac{1}{2}(\partial u_i/\partial x_j - \partial u_j/\partial x_i)$ represents the rotation tensor, whereas c_κ denotes the von Karman constant and d the distance to the closest solid surface. The two additional functions f_{v2} and f_{v3} are given by

$$f_{v2} = \left(1 + \frac{\chi}{c_{v2}}\right)^{-3}, \qquad f_{v3} = \frac{(1+\chi)(1-f_{v2})}{\chi}. \tag{2.29}$$

As for the destruction term, the wall function f_w is defined as

$$f_w = g\left(\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right)^{1/6}, \qquad g = r + c_{w2}(r^6 - r), \qquad r = \frac{\tilde{\mu}}{\tilde{\omega}\rho c_\kappa^2 d^2}. \tag{2.30}$$

Similarly to algebraic models, the mixing length of the S-A model is defined by $l_m^{SA} = \sqrt{\mu_t/\rho\omega}$, and the quantity r can be interpreted as the squared ratio of the model mixing lenth to the mixing length in the logarithmic region.

The definition of the model is completed by the trip functions

$$\begin{aligned} f_{t1} &= c_{t1} g_t \exp\left(-c_{t2}\frac{\omega_t^2}{\|\mathbf{u}-\mathbf{u}_t\|^2}(d^2 + g_t^2 d_t^2)\right), \\ f_{t2} &= c_{t3}\exp\left(-c_{t4}\chi^2\right), \\ g_t &= \min\left(0.1, \frac{\|\mathbf{u}-\mathbf{u}_t\|^2}{\omega_t \triangle x_t}\right) \end{aligned} \tag{2.31}$$

which are calibrated to represent laminar-to-turbulent transition at prescribed positions. Using the trip term requires the a priori knowledge of the transition points, either by an educated guess, or by experimental measurements. The trip functions f_{t1} and f_{t2} are expressed in terms of the distance to the nearest transition point d_t, the vorticity at the trip point ω_t and the grid spacing at this point $\triangle x_t$. $\|\mathbf{u}-\mathbf{u}_t\|^2$ is the difference between the velocity at the field point and the velocity at the trip (on the wall).

Finally, the closure constants of the model used here are $c_{v1} = 7.1$, $c_{v2} = 5.0$, $c_{b1} = 0.1355$, $c_{b2} = 0.622$, $c_{w1} = 3.2391$, $c_{w2} = 0.3$, $c_{w3} = 2.0$, $c_\kappa = 0.41$, $\sigma = 2/3$, $c_{t1} = 1.0$, $c_{t2} = 2.0$, $c_{t3} = 1.2$ and $c_{t4} = 0.5$.

Remarks

Some clarifying remarks are in order.

Laminar-to-turbulent transition. The original S-A model makes use of a "trip term" to force turbulent transition at a desired location. The major drawback is that the transition point must be known a priori, which, especially for complex flows, is generally not the case. As the model is most often employed for fully turbulent applications, many researchers do not include the trip term. According to Rumsey [148], even for fully turbulent computations, a grid dependent transition region can occur if the farfield turbulence level

is too low ($\tilde{\mu}_{farfield} < 3\mu_\infty$). The appearance of a laminar region is further increased by the presence of the f_{t2} term but has no significant impact in practice. Indeed, because the transition region is small, the overall solution remains globally consistent. In this work, no effort to model/force transition has been made and we simply omitted the trip term by setting $f_{t1} = f_{t2} = 0$. For all applications considered here, we let the transition take place due to numerical reasons.

Compressibility effects. As most turbulence models, the S-A model was initially developped for incompressible flows. The presented form of the model shown in (2.25) to (2.30) is a straightforward generalization to compressible flows. Besides the definition of the production term, it differs from the original model in the choice of the working variable $\tilde{\mu} = \rho\tilde{\nu}$ and the definition of $\chi \equiv \tilde{\mu}/\mu$. Again, we made no effort to improve the model for flows where compressibility effects are highly important - e.g. hypersonic flows, flows with a significant heat transfer or flows involving combustion. Probably, another form would be more suitable for such cases. For example, Spalart [163] proposes a correction to improve the behaviour of the model in compressible mixing layers. Alternatively, Catris and Aupoix [29] suggest to replace the diffused quantity by $\sqrt{\rho}\tilde{\nu}$.

Rotation/Curvature correction. Shur *et al.* [160] worked out a correction to take into account system rotation and streamline curvature. A less capable but far simpler alternate has been proposed by Dacles-Mariani *et al.* [43]. However, none of these modifications is implemented in our code.

2.3.4 Negative $\tilde{\mu}$ modifications

Despite the popularity of the S-A model, the discretization of the turbulence transport equation remains a challenging task. The difficulties arise from the stiffness of the source term and the appearance of negative turbulent quantities $\tilde{\mu}$. Whereas the exact solution of (2.25) is such that $\tilde{\mu} \geq 0$, negative values can occur, wherever the grid resolution is insufficient to resolve the turbulent field. Often, this is the case in the outer boundary layer where $\tilde{\mu}$ decreases rapidly to the free-stream value. The sharp gradients in this zone cause Gibbs oscillations which, if no attention is payed, can deteriorate the iterative convergence, up to the divergence of the algorithm in some cases. To illustrate this phenomenon, let us consider the L_2-projection of the following piecewise function

$$f(\xi) = \begin{cases} -10(\xi - \frac{1}{2})^3 - (\xi - \frac{1}{2}) & \text{if } -1 \leq \xi \leq \frac{1}{2}, \\ 0 & \text{if } \frac{1}{2} < \xi \leq 1 \end{cases} \tag{2.32}$$

onto Lagrange interpolants of the polynomial order $p = 0$ to $p = 3$; see Fig. 2.1. Expression (2.32) is typical of the viscosity profile at the end of the log-layer. Characterized by large values inside the boundary layer, $\tilde{\mu}$ rapidly decreases to the free-stream value which is comparable to the molecular viscosity. Although

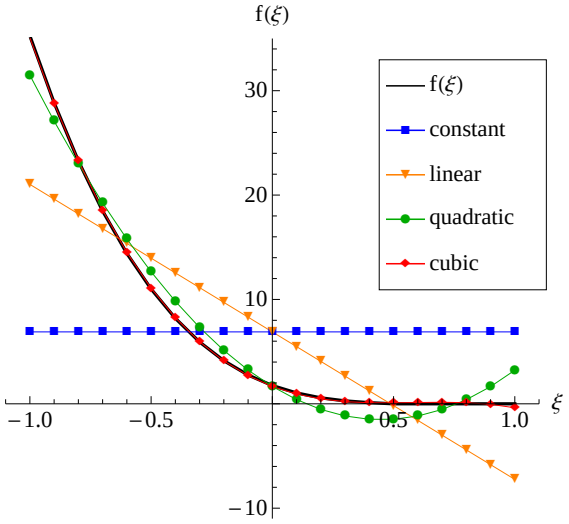


Figure 2.1: L_2-projection of the cubic function (2.32) onto Lagrange interpolants, shown for the polynomial order $p = 0$ to $p = 3$.

the exact function $f(\xi)$ is positive for all ξ in the interval $[-1, 1]$, the finite element (FE) approximations partially fall below zero, except for P^0-elements. Furthermore, regions where $f(\xi) < 0$ become less important, by increasing the polynomial order p. According to these observations, two strategies can be pursued to avoid/limit the appearance of negative values of $\tilde{\mu}$.

The first, consists in using constant ansatz functions, because P^0-elements are (obviously) better suited to ensure the positivity[2]. This approach is equivalent to a cell-centered first-order finite volume (FV) method. For the latter, many researchers combine a first-order upwinding of the convective terms in the turbulence model with a second-order discretization of the Navier-Stokes equations in order to improve the stability of the scheme. As an example, Geuzaine used this approach in his thesis [71] with a finite volume method. The same strategy but based on a finite difference (FD) discretization has been followed by Chisholm and Zingg [35].

Whereas a constant reconstruction of the turbulence model seems sufficient in combination with a second-order scheme, we expect that it will lead to poor results with higher-order interpolants ($p \geq 3$) [3]. The reason is that the latter are intended for the use with much coarser grids than those encountered for "standard" FV or FD codes. In order to not spoil the overall accuracy of the method, we discretize all equations (turbulence model and Navier-Stokes equations) by the same high-order interpolants. As seen before, this approach

[2]However, as the derivative of a constant function equals zero, P^0-elements will inevitably lead to an inconsistent FE discretization for elliptic equations.

[3]This will be examined in the results chapter.

helps to reduce the amount of negative $\tilde{\mu}$ but does not guarantee its positivity. We therefore have to define a number of modifications to handle negative values of the turbulence variable. These modifications are described below.

Eddy viscosity

In order to improve the robustness, there is agreement to neglect negative values in the momentum equation. A standard change is therefore the definition of the eddy viscosity μ_t in the following way

$$\mu_t = \begin{cases} \tilde{\mu} f_{v1} & \text{if } \chi > 0, \\ 0 & \text{if } \chi \leq 0. \end{cases} \tag{2.33}$$

This modification avoids a blow-up of the momentum equation due to non-physical negative viscosities. F. Bassi and S. Rebay [11] introduced a similar definition in the first DG implementation with the $k-\omega$ model. In his thesis, Landmann [107] refers to this approach as "soft limiting".

Modification of the Spalart-Allmaras model

Given the definitions (2.27) and (2.29) it is easy to see that the Spalart-Allmaras model itself becomes unstable for negative $\tilde{\mu}$. Indeed,

$$\begin{aligned} \lim_{\chi \to -c_{v1}} f_{v1} &= \infty, \\ \lim_{\chi \to -c_{v2}} f_{v2} &= \infty. \end{aligned} \tag{2.34}$$

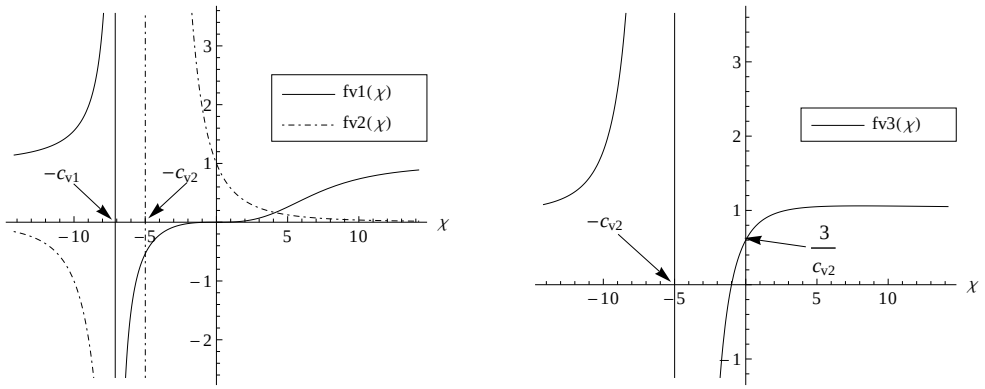
Furthermore, the function f_{v3} becomes ill-conditioned in the limit as $\chi \to 0$, i.e.

$$\lim_{\chi \to 0} f_{v3} = \text{“}\frac{0}{0}\text{”} = \frac{3}{c_{v2}}. \tag{2.35}$$

The behaviour of the S-A model for small turbulent viscosities is displayed in Fig. 2.2.

Different approaches have been proposed in the literature to improve the robustness of the S-A model.

Clipping. The easiest and most straightforward way to handle negative viscosities is "clipping". A very intrusive way is proposed by Landmann *et al.* [108]: they use a "hard limitation technique" which consists in correcting the solution vector after each time. Thereto, the global minimum of $\tilde{\mu}$ is computed for every cell. If this minimum undershoots zero, two cases have to be considered. If the integral cell average $\langle \tilde{\mu} \rangle$ is positive, $\tilde{\mu}$ is approximated by a linear function, that (i) conserves the cell average $\langle \tilde{\mu} \rangle$, and (ii) whose minimum is limited to zero. Otherwise (if $\langle \tilde{\mu} \rangle < 0$), the cell average is set



(a) f_{v1} and f_{v2} closure functions.

(b) f_{v3} closure function.

Figure 2.2: Behaviour of the S-A turbulence model for small values of the non-dimensional working variable $\chi = \tilde{\mu}/\mu$. The model itself becomes unstable for negative χ.

to a small positive value. In this way, Landmann *et al.* try to improve the robustness of the code, while keeping the conservativity of the S-A model discretization. However, this approach clearly interferes with the convergence of the Spalart-Allmaras equation. Furthermore, for higher polynomial approximations ($p \geq 4$), the search of a global minimum proves extremely costly and may become a dominant part of the overall scheme. A much more subtle, yet as effective way, is to allow for negative values of $\tilde{\mu}$ but to neglect the negative values when evaluating the diffusive and source term. This second approach is depicted as "*clipped S-A model*" in the following.

Tailored diffusion. Based on the method used for sub-cell shock capture [133], Nguyen *et al.* [125] introduce an artificial viscosity for the turbulence model. Thereto, they compare the total turbulent "energy" $E = \tilde{\nu} M \tilde{\nu}$ to the energy in the high modes $E_H = \tilde{\nu} M_H \tilde{\nu}$. Here, M and M_H represent the element mass matrix and the mass matrix using only the high Koornwinder [103] coefficients, respectively. If the turbulence variable is found to deviate from a smooth function, in other words if too much energy is contained in the high modes, an artificial diffusion term is activated, which flattens the transition layer to a thickness that can be resolved by the grid. Based on numerical tests, they found that the energy in the high modes should approximately decay as $(1/p)^4$, where p denotes the polynomial order of the interpolation.

Modified diffusive and source term operators. The major drawback of clipping is that the S-A model is not C^1 continuous, which can hamper the convergence to steady state. In his thesis, Oliver [129] describes an alternative form for the diffusive and source term operators to ameliorate the behaviour

of the model in the presence of negative $\tilde{\mu}$ values. In contrast to clipping, the new form is continuous and has continuous first derivatives at $\tilde{\mu} = 0$. In the present implementation, we have adapted these modifications to the S-A model described in section 2.3.3. The latter differ in the definition of the production term - in particular, the modified vorticity $\tilde{\omega}$ and the functions f_{v1}, f_{v2} and f_{v3}. Furthermore, we use the working variable $\tilde{\mu}$ instead of $\tilde{\nu} = \tilde{\mu}/\rho$.

The changes to the turbulence model proposed by Oliver are based on the "energy" of the working variable $\tilde{\nu}$. We therefore first rewrite (2.25) in function of $\tilde{\nu}$

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \frac{\partial u_j \rho\tilde{\nu}}{\partial x_j} = \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\mu + \rho\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\rho\frac{\partial \tilde{\nu}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j}\right] + (P_{\tilde{\nu}} - D_{\tilde{\nu}}), \quad (2.36)$$

where $P_{\tilde{\nu}} = c_{b1}\tilde{\omega}\rho\tilde{\nu}$ is the production and $D_{\tilde{\nu}} = \rho c_{w1} f_w(\tilde{\nu}/d)^2$ the destruction term[4]. Multiplying (2.36) by $\tilde{\nu}$ and taking into account the continuity equation, the turbulent "energy" $e_{\tilde{\nu}} \equiv \frac{1}{2}\tilde{\nu}^2$ can be computed as

$$\frac{\partial \rho e_{\tilde{\nu}}}{\partial t} + \frac{\partial \rho u_j e_{\tilde{\nu}}}{\partial x_j} = \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left(\eta\frac{\partial e_{\tilde{\nu}}}{\partial x_j}\right) + (c_{b2}\rho\tilde{\nu} - \eta)\frac{\partial \tilde{\nu}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j}\right] + \tilde{\nu}(P_{\tilde{\nu}} - D_{\tilde{\nu}}) \quad (2.37)$$

with the scalar diffusion parameter $\eta = \mu + \rho\tilde{\nu}$. We further define the integrated "energy" $E^-_{\tilde{\nu}}$ corresponding to negative turbulent quantities as

$$E^-_{\tilde{\nu}}(t) = \int_{T^-} \rho\, e_{\tilde{\nu}}(\mathbf{x}, t)\, dV \quad (2.38)$$

with $T^- = \{\mathbf{x} \in T \mid \tilde{\nu}(\mathbf{x}, t) < 0\}$. Bearing in mind that $\tilde{\nu}|_{\partial T^-} = 0$, the application of the divergence theorem finally leads to

$$\frac{dE^-_{\tilde{\nu}}}{dt} = \int_{T^-}\left[\frac{(c_{b2}\rho\tilde{\nu} - \eta)}{\sigma}\frac{\partial \tilde{\nu}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j} + \tilde{\nu}(P_{\tilde{\nu}} - D_{\tilde{\nu}})\right] dV. \quad (2.39)$$

According to (2.39), a sufficient condition to ensure the decrease in time of the negative turbulent energy $E^-_{\tilde{\nu}}$ is

$$\begin{cases} c_{b2}\rho\tilde{\nu} - \eta & < 0, \\ \tilde{\nu}\,(P_{\tilde{\nu}} - D_{\tilde{\nu}}) & < 0. \end{cases} \quad (2.40)$$

To satisfy these conditions, the diffusion parameter η is re-defined as follows:

$$\eta = \begin{cases} \mu\,(1 + \chi) & \chi \geq 0, \\ \mu\,(1 + \chi + \frac{1}{2}\chi^2) & \chi < 0. \end{cases} \quad (2.41)$$

As for the source term, a slightly amended modification is proposed. Differences are due to the definition of f_{v1}, f_{v2}, f_{v3} and $\tilde{\omega}$. The idea consists

[4]Note that we have omitted the f_{t2} function in the definition of $P_{\tilde{\nu}}$ and $D_{\tilde{\nu}}$ as it is not used in this work.

in modifying $(P - D)$ in such a way that it is positive definite for $\chi < 0$ with continuous first derivatives at $\chi = 0$. Starting from the definition of the production term, one gets

$$P_{\tilde{\mu}} = c_{b1}\,\tilde{\omega}\tilde{\mu}, \tag{2.42}$$

$$\begin{aligned}
\Rightarrow \frac{\partial P_{\tilde{\mu}}}{\partial \tilde{\mu}}\Big|_{\tilde{\mu}=0} &= c_{b1}\left[\frac{\partial \tilde{\omega}}{\partial \tilde{\mu}}\tilde{\mu} + \tilde{\omega}\right]_{\tilde{\mu}=0} \\
&= c_{b1}\left[\left(\frac{\omega f'_{v3}}{\mu} + \frac{f_{v2}}{\rho c_\kappa^2 d^2} + \frac{\tilde{\mu} f'_{v2}}{\rho \mu c_\kappa^2 d^2}\right)\tilde{\mu} + \omega f_{v3} + \frac{\tilde{\mu} f_{v2}}{\rho c_\kappa^2 d^2}\right]_{\tilde{\mu}=0} \\
&= \frac{3 c_{b1}\,\omega}{c_{v2}}
\end{aligned} \tag{2.43}$$

with $f_{v1}(0) = 0$, $f'_{v1}(0) = 0$, $f_{v2}(0) = 1$, $f'_{v2}(0) = -3/c_{v2}$, $f_{v3}(0) = 3/c_{v2}$ and $f'_{v3}(0) = -12/c_{v2}^2$. We therefore suggest the following modified production term

$$P_{\tilde{\mu}} = \begin{cases} c_{b1}\tilde{\omega}\tilde{\mu} & \tilde{\mu} \geq 0, \\ \frac{3\,c_{b1}\omega\tilde{\mu}}{c_{v2}} g_n & \tilde{\mu} < 0, \end{cases} \tag{2.44}$$

where

$$g_n = 1 - \frac{1000\chi^2}{1+\chi^2}. \tag{2.45}$$

Note that except a factor $3/c_{v2}$ due to a different definition of the S-A model, (2.44) is identical to the expression proposed by Oliver. The function g_n is chosen such that $P_{\tilde{\mu}} > 0$ without affecting the derivative at $\tilde{\mu} = 0$. However, for $-\sqrt{1/999} < \chi < 0$, small negative values of $P_{\tilde{\mu}}$ must be accepted. In a similar way, the new destruction term $D_{\tilde{\mu}}$ is given by

$$D_{\tilde{\mu}} = \begin{cases} c_{w1} f_w \frac{\tilde{\mu}^2}{\rho d^2} & \tilde{\mu} \geq 0, \\ -c_{w1} \frac{\tilde{\mu}^2}{\rho d^2} & \tilde{\mu} < 0. \end{cases} \tag{2.46}$$

It is easily seen that with the definitions (2.41), (2.44) and (2.46) the Spalart-Allmaras model is continuous and has continuous first derivatives at $\tilde{\mu} = 0$.

2.4 Variational MultiScale method

The *Variational MultiScale (VMS) method*, a variant of the classical *Large-Eddy Simulation (LES)*, has known a growing interest over the last decade since it has been proposed in 2000 by Hughes *et al.* [90–92]. The motivation for the development of LES/VMS is the need for more accurate methods to simulate turbulent flows. Although most RANS models (cf. section 2.3) deliver satisfying results for flows they have been designed/calibrated for, they generally fail to accurately predict complex cases involving e.g. flow separation or laminar-to-turbulent transition. The primary reason is the intrinsic

assumption that all turbulent scales - from the small dissipation scales to the large energy containing scales - can be represented by a unique turbulence model. Whereas this assumption is reasonable for the small eddies, which have an almost uniform behaviour, large eddies depend on the geometry. Hence, their simulation by a "general" turbulence model seems at least questionable.

A *Direct Numerical Simulation (DNS)* requires to resolve the small dissipation-range eddies and hence still exceeds by far the resources of modern supercomputers if industrial applications at moderate or high Reynolds numbers are considered. The high cost of DNS can be overcome by the use of LES methods. As the name suggests, the idea consists in cutting the energy spectrum: large, energy-containing scales are resolved by the computational mesh whilst small scales are modelled. Because the large eddies contain most of the energy, the cutting off does not significantly affect the energy spectrum. This is not true for the vorticity spectrum, whose primary role is to dissipate energy. To compensate for the absence of small eddies, the dissipation process is controlled by a turbulence model. The latter is much simpler than corresponding RANS models, because only the dissipation-range scales have to be modelled.

In practice, the energy spectrum is cut in the inertial range, which is responsible for the energy transfer from the large to the dissipation scales. Because the latter are generally at least one order of magnitude smaller than the largest scales, LES significantly reduces the cost compared with a DNS simulation. For instance, increasing the element size by a factor 5 reduces by 125 the number of mesh nodes of a 3D computation [54].

Today, LES has been applied to a wide range of turbulent flows, e.g. isotropic, free-shear, wall-bounded, separated, rotating, multiphase. For an overview of the history of LES and its applications we refer to the textbook of Lesieur *et al.* [112].

Standard LES formulation. In the standard LES formulation, the separation of turbulence scales is achieved by applying a low-pass filter to the Navier-Stokes equations (2.1) yielding the following transport equations for the resolved fields [120] (assuming filtering commutes with differentiation):

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0, \tag{2.47a}$$

$$\frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{u}_i + \bar{p}\delta_{ij} - \tilde{\tau}_{ij}) = -\frac{\partial \tau_{ij}^{sgs}}{\partial x_j} + \frac{\partial}{\partial x_j}(\bar{\tau}_{ij} - \tilde{\tau}_{ij}), \tag{2.47b}$$

$$\frac{\partial \bar{\rho}\tilde{E}}{\partial t} + \frac{\partial}{\partial x_j}\left[\tilde{u}_j(\bar{\rho}\tilde{E} + \bar{p}) + \tilde{q}_j - \tilde{u}_i\tilde{\tau}_{ij}\right] = -\frac{\partial q_j^{sgs}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\mathcal{J}_i - \mathcal{D}_j - (\bar{q}_j - \tilde{q}_j)\right]. \tag{2.47c}$$

In a similar way as for the Reynolds equations, Favre's average (cf. equation (2.17)) has been used to avoid the introduction of additional subgrid terms

in the continuity equation. Hence, we distinguish between filtered/resolved quantities $(\bar{\cdot})$ and Favre averaged quantities $(\tilde{\cdot})$.

The effect of the small/subgrid scales (SGS) on the large/resolved scales appears on the right-hand side of (2.47). In the present work, only the SGS stress tensor $\tau_{ij}^{sgs} = \bar{\rho}(\widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j)$ and the SGS heat flux $q_j^{sgs} = \bar{\rho}(\widetilde{u_j e} - \tilde{u}_j \tilde{e})$ are modelled. Following the results of Vreman *et al.* [178] for a mixing layer at Mach numbers in the range of 0.2-0.6, the nonlinearities due to the diffusion terms, i.e. $(\bar{\tau}_{ij} - \tilde{\tau}_{ij})$, the SGS turbulent diffusion $\mathcal{J}_j = (\bar{\rho}\widetilde{u_j u_i u_i} - \bar{\rho}\tilde{u}_j\widetilde{u_i u_i})/2$ and the SGS viscous diffusion $\mathcal{D}_j = \overline{\tau_{ij} u_i} - \tilde{\tau}_{ij}\tilde{u}_i$, are neglected in the momentum and energy equation. Similarly, we neglect the last term $(\bar{q}_j - \tilde{q}_j)$ in (2.47c). Several models for the (remaining) unclosed terms in the energy equation can be found in [120]. For an extensive overview of compressible LES, including formulations for e.g. the filtered internal energy or enthalpy equation, we refer to the textbook of Garnier *et al.* [63].

It is important to note that although the LES equations (2.47) are formally similar to the RANS equations (2.18), the concept is quite distinct. Whereas the filter acts as a local smoother that leaves an irregular, turbulent velocity field; the Reynolds average sums over an ensemble, creating a regular smooth velocity field [54].

Variational multiscale method. The variational multiscale method [90] is motivated by the observation that the interaction between turbulent scales mainly involves eddies having a similar size. In other words, the turbulence theory assumes that large scales are produced by the main shear. Their energy is then transferred through the cascade involving successively smaller scales and finally dissipated in the small scales by viscous effects.

In contrast to the classical LES formulation, where scales are separated into resolved and unresolved scales by filtering the Navier-Stokes equation, the VMS method distinguishes three scale groups. The resulting scale separation is written as

$$u_i = \underbrace{\bar{\tilde{u}}_i + \tilde{u}_i^s}_{\text{resolved}} + \underbrace{u_i'}_{\text{unresolved}} \quad , \tag{2.48}$$

where $\bar{\tilde{u}}_i$ denotes the *large resolved* scales, $\tilde{u}_i^s$ the *small resolved* scales and u_i' the *unresolved* scales respectively. By splitting the resolved scales, the VMS method allows to restrict the effect of the unresolved (modelled) scales to the small eddies. Since energy is extracted only from the small (resolved) eddies, the large scales are - in contrast to standard LES - not affect by the turbulence modelling and thus remain consistent with DNS, if an adequate resolution is achieved by the large-scale space.

To derive the VMS formulation, scales are separated a priori. By substituting (2.48) into the Navier-Stokes equations and by applying the DG variational formulation described in chapter 3, two coupled systems are written: a *large-scale equation* and a *small-scale equation*. Thereby, the unresolved

scales are neglected in the large-scale equation, because their mutual influence is assumed to be of minor relevance. For a detailed review of the VMS method applied to incompressible flows we refer to Gravemeier [73] and the therein cited literature. An extension to compressible flows in a mixed finite element/finite volume context has been proposed by Koobus and Farhat [102].

In the literature, the following solution strategies for the VMS formulation are distinguished:

- *Explicit solving of the large- and the small-scale equation.* Following this approach, the flow field is expressed as the sum of two contributions, which are obtained by separately solving the large-scale and the small-scale equation. Hence, a coupled system of non-linear equations has to be solved. Since the residual of the large-scale equation acts as a "driving force" on the small-scale momentum equation, this formulation is generally referred to as *residual-based VMS*, e.g. [14, 19, 20, 62, 74].

- *Solving a monolithic system of equations.* In this thesis, the large-scale and the small-scale equation are reunified into a single expression, which is formally nearly identical to the standard LES formulation, cf. equation (2.47). The difference lies in the subgrid viscosity term, which - in case of VMS - depends only on the smallest resolved scales, leading to the desired scale separation. Computing the subgrid term τ_{ij}^{sgs} using $\tilde{u}_i^s$ instead of $\tilde{u}_i$ agrees with the assumption that the unresolved scales should behave similarly to the smallest resolved scales. The same strategy has been used in e.g. [21, 53, 91, 102].

2.4.1 Subgrid-scale modelling

In the present work, the unclosed subgrid terms are modelled based on a generalization to compressible flows of the eddy-viscosity assumption, cf. section 2.3.2. Accordingly, the SGS stress tensor is given as

$$\tau_{ij}^{sgs} = 2\bar{\rho}\mu_t \left(\tilde{s}_{ij} - \frac{1}{3}\tilde{s}_{ll}\delta_{ij} \right) \underbrace{+\frac{1}{3}\tau_{kk}^{sgs}\delta_{ij}}_{\text{neglected here}} . \tag{2.49}$$

where $\tilde{s}_{ij}$ is the resolved strain rate tensor. The SGS heat flux vector q_j^{sgs} is modelled using the eddy viscosity hypothesis and a turbulent Prandtl number Pr_t (cf. section 2.3.2)

$$q_j^{sgs} = \frac{\mu_t c_p}{Pr_t} \frac{\partial \tilde{T}}{\partial x_j}. \tag{2.50}$$

Two studies have shown that at low Mach numbers neglecting the last term in (2.49) has either no effect on the LES results (homogeneous isotropic turbulence, Squires [166]) or even leads to a better agreement with DNS (mixing layer flow, Vreman *et al.* [177]). The last study furthermore revealed that the

simulations were less stable if τ_{kk}^{sgs} was taken into account. For these reasons, we neglect this term in the present work.

Since the closure problem for the SGS stress tensor (2.49) is by far less demanding than the closure problem for the Reynolds stress tensor (2.19), attempts have been made to use dissipative numerical algorithms without any subgrid model. In the literature, such an approach is commonly known as *implicit LES*, e.g. Grinstein *et al.* [75].

Depending on whether the strain rate tensor $\tilde{s}_{ij}$ and the eddy viscosity μ_t are computed using all resolved scales $\tilde{\mathbf{u}}$ or only the small-resolved scales $\tilde{\mathbf{u}}^s$, four possible combinations are distinguished (Hughes *et al.* [90, 91]):

$$\text{all-all:} \qquad \tau_{ij}^{sgs} = 2\mu_t(\tilde{\mathbf{u}}) \left(\tilde{s}_{ij} - \frac{1}{3}\tilde{s}_{ll}\delta_{ij} \right) \tag{2.51a}$$

$$\text{small-small:} \qquad \tau_{ij}^{sgs} = 2\mu_t(\tilde{\mathbf{u}}^s) \left(\tilde{s}_{ij}^s - \frac{1}{3}\tilde{s}_{ll}^s\delta_{ij} \right) \tag{2.51b}$$

$$\text{small-all:} \qquad \tau_{ij}^{sgs} = 2\mu_t(\tilde{\mathbf{u}}^s) \left(\tilde{s}_{ij} - \frac{1}{3}\tilde{s}_{ll}\delta_{ij} \right) \tag{2.51c}$$

$$\text{all-small:} \qquad \tau_{ij}^{sgs} = 2\mu_t(\tilde{\mathbf{u}}) \left(\tilde{s}_{ij}^s - \frac{1}{3}\tilde{s}_{ll}^s\delta_{ij} \right) \tag{2.51d}$$

Here, $s_{ij}^s = \frac{1}{2}\left(\frac{\partial \tilde{u}_i^s}{\partial x_j} + \frac{\partial \tilde{u}_j^s}{\partial x_i} \right)$ denotes the strain rate tensor of the small scale field whilst $\tilde{s}_{ij}$ is the strain rate tensor of the complete LES field. Definition (2.51a) corresponds to the standard LES formulation; equation (2.51b) is obtained using the a priori scale separation of the VMS method. The last two expressions (2.51c) and (2.51d) are given for completeness only. Hence, (2.51c) constitutes an improvement over standard LES in laminar regions of transitional flows, but does not improve the spectral behaviour of the model - in contrast to (2.51b) [53]. Note also, that according to Vreman [176], the application of an additional high-pass filter to (2.51b) does not significantly change the spectral behaviour of the model, given the low large-scale content of (2.51b), i.e.

$$2\mu_t(\tilde{\mathbf{u}}^s) \left(\tilde{s}_{ij}^s - \frac{1}{3}\tilde{s}_{ll}^s\delta_{ij} \right) \approx \left[2\mu_t(\tilde{\mathbf{u}}^s) \left(\tilde{s}_{ij}^s - \frac{1}{3}\tilde{s}_{ll}^s\delta_{ij} \right) \right]^s . \tag{2.52}$$

The superscript "s" indicates a projection onto the small-scale space. Of course, the VMS approach does not only apply to the momentum (2.47b) but also the energy equation (2.47c). Hence, in the current work, the same choice (all or small) is made for the whole LES system.

In this thesis, two subgrid-scale models have been implemented, namely the *Smagorinsky model* and the *WALE model*.

Smagorinsky model

One of the most popular subgrid-scale models has been proposed by Smagorinsky [161]. Based on the hypothesis of local equilibrium between turbulence

production and dissipation, the eddy viscosity is written as

$$\mu_t = \bar{\rho} C_s \Delta^2 |\tilde{\mathbf{S}}^s| \quad \text{with} \quad |\tilde{\mathbf{S}}^s| = \sqrt{2 \tilde{s}^s_{ij} \tilde{s}^s_{ij}}. \tag{2.53}$$

The model coefficient C_s (Smagorinsky's constant) has been calibrated to 0.027 on isotropic homogeneous turbulence at high Reynolds numbers, cf. Lilly [115]. Different values of C_s are sometimes preferred in order to improve the behaviour of the model for free-shear and wall-bounded flows, see Lesieur *et al.* [112]. The grid spacing used in this work is taken as $\Delta = (\Delta V)^{1/3}/p$, where V is the cell volume and p denotes the degree of the polynomial interpolation of the solution, cf. chapter 3.

Note that (2.53) represents a filtered version of the Smagorinsky model since only the small-scales are included in the strain rate tensor. Whereas the filtered model significantly reduces the over-dissipative behaviour of the original model in laminar regions, this is not true in the viscous sublayer. Indeed, near to the wall, the subgrid shear stress (and the eddy viscosity) should approach zero as y^3. However, the original and the filtered models both lead to $\mu_t \approx \mathcal{O}(1)$, cf. Duponcheel [53]. To cure this problem, early attempts resorted either to damping functions or to wall models. An alternative approach to recover the correct near-wall behaviour is the use of the dynamic procedure proposed by Germano *et al.* [69]. In addition, the latter also presents the advantage that the model coefficient C_s is no longer to be chosen a priori, but is computed in the course of the simulation. Within the scope of this thesis, only "static" subgrid scale models have been tested. The implementation of the dynamic procedure is subject to future work.

WALE model

Since neither the standard nor the filtered Smagorinsky model lead to the correct near wall behaviour, we have implemented a second SGS model: *the Wall-adapting local eddy viscosity (WALE) model*. In order to provide the correct scaling of $\mu_t \approx \mathcal{O}(y^3)$ close to the wall, Nicoud and Ducros [126] devised the following expression

$$\mu_t = \bar{\rho} C \Delta^2 \frac{\left(\tilde{\mathscr{S}}^d_{ij}(\tilde{\mathbf{u}}^s)\tilde{\mathscr{S}}^d_{ij}(\tilde{\mathbf{u}}^s)\right)^{3/2}}{\left(\tilde{s}^s_{ij}\tilde{s}^s_{ij}\right)^{5/2} + \left(\tilde{\mathscr{S}}^d_{ij}(\tilde{\mathbf{u}}^s)\tilde{\mathscr{S}}^d_{ij}(\tilde{\mathbf{u}}^s)\right)^{5/4}}. \tag{2.54}$$

Here, $\tilde{\mathscr{S}}^d_{ij}(\tilde{\mathbf{u}}^s)$ denotes the deviatoric part of the square of the velocity gradient tensor,

$$\tilde{\mathscr{S}}_{ij}(\tilde{\mathbf{u}}^s) = \frac{1}{2}\left(\frac{\partial \tilde{u}^s_i}{\partial x_k}\frac{\partial \tilde{u}^s_k}{\partial x_j} + \frac{\partial \tilde{u}^s_j}{\partial x_k}\frac{\partial \tilde{u}^s_k}{\partial x_i}\right) = \tilde{s}^s_{ik}\tilde{s}^s_{kj} + \tilde{\Omega}^s_{ik}\tilde{\Omega}^s_{kj}, \tag{2.55}$$

$\tilde{\Omega}^s_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}^s_i}{\partial x_j}\frac{\partial \tilde{u}^s_j}{\partial x_i}\right)$ is the small scale part of the rotation tensor. In this work, the model constant is taken as $C = 0.25$. This value provided good results

for homogeneous isotropic turbulence at moderate Reynolds numbers and for a turbulent pipe flow at $Re_D = 10^4$ [126]. Again, the projection onto small scales guarantees that the model is active only if the LES field contains high wavenumbers.

Validation of the numerical model

To validate the numerical implementation of the SGS models, we consider the following vortex, cf. Bricteux *et al.* [21]:

$$\omega(r) = \frac{\Gamma_0}{\pi}\frac{r_c^2}{(r^2+r_c^2)^2}, \quad \Gamma(r) = \Gamma_0\frac{r^2}{r^2+r_c^2}, \quad U_\theta(r) = \frac{\Gamma(r)}{2\pi r}. \tag{2.56}$$

Here, Γ_0 is the total vortex circulation and r_0 is the radius corresponding to the maximum tangential velocity u_θ. Similar vortices are generally encountered as part of a two-vortex structure that occurs after roll-up of the wake generated by a wing. Given the simple algebraic expression of the vortex, we can calculate the analytical expression of the SGS viscosity in order to compare it with the numerical results. For the Smagorinsky model, one obtains

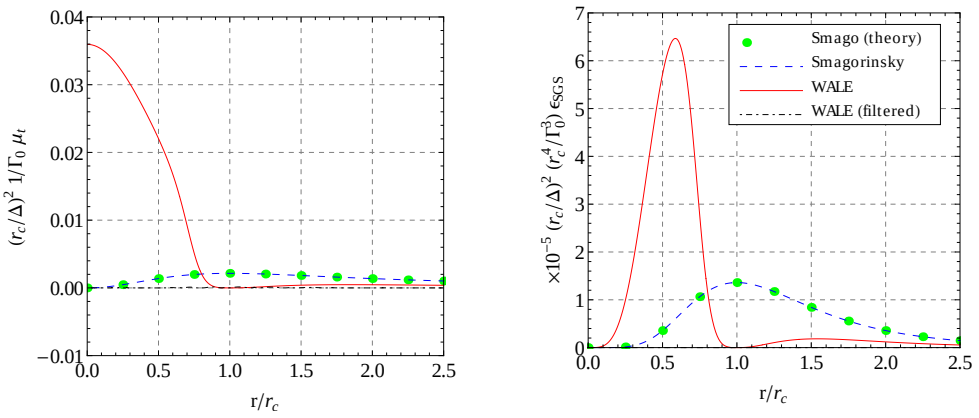
$$\frac{\mu_t^{Smago}}{\Gamma_0}\left(\frac{r_c}{\Delta}\right)^2 = \frac{C_s}{\pi}\frac{\left(\frac{r}{r_c}\right)^2}{\left[\left(\frac{r}{r_c}\right)^2+1\right]^2}. \tag{2.57}$$

The computed eddy viscosity μ_t and SGS dissipation ϵ_{SGS} using a $p = 4$ interpolation of the solution are shown in Fig. 2.3. We observe that the numerical values for the Smagorinsky model are in perfect agreement with the analytical expression. Furthermore, a comparison with the results in [21] (cf. FIG. 1 and FIG. 2 respectively) suggests that the WALE model has also been correctly implemented.

Despite the simplicity of the algebraic vortex, it clearly illustrates the over-dissipative behaviour of the standard Smagorinsky and WALE model in the presence of vortical structures. Indeed, as the flow is laminar, the SGS model should be inactive. As expected, the application of a high-pass filter significantly improves this situation. Although the integrated dissipation is even slightly higher for the Smagorinsky model [21], the latter leads to a much smaller peak viscosity. As a consequence, the WALE model suffers from a severe time step restriction, if explicit time integration schemes are used.

2.4.2 Scale separation

Within the DGFEM context (see chapter 3), scale separation can easily be achieved by a variational projection of the resolved scales $\tilde{u}$ onto distinct function spaces. Furthermore, given the large bandwith of scales included in each element, if a high-order polynomial approximation is used, the filtering can



(a) SGS viscosity μ_t (b) SGS dissipation ϵ_{SGS}

Figure 2.3: Normalized μ_t and ϵ_{SGS} profiles for an algebraic model vortex. The perfect agreement with the analytical expression suggests that the Smagorisnky model has been correctly implemented.

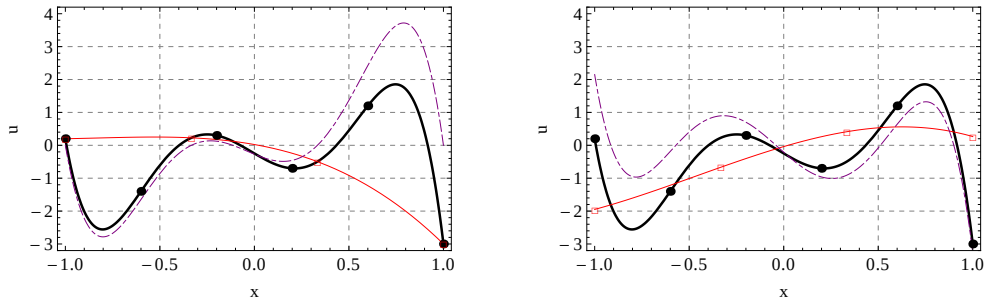
be realized on an element level. This constitutes an important advantage over standard low-order methods implying e.g. cell agglomeration techniques [102]. As the flow field is filtered locally, no data exchange between neighbouring computing cells is required. This makes the scale separation suitable for parallel computations. In addition, the technique applies as well to structured as to unstructured meshes.

If a *modal basis*[5] is used, scales can be separated by a simple restriction of the polynomial approximation to a subset of shape functions associated to the low/high order modes. This straightforward filtering does not require any additional operation and is thus computationally very efficient. A more sophisticated technique providing a better control of the resolved spectrum by filtering in the spectral domain is the *discrete polynomial transform (DPT)*. For a detailed description of DPT filters we refer to Blackburn and Schmidt [15].

If - as in the current work - a *nodal basis* is used (see section 3.2.1), an *interpolant projection* is generally preferred because otherwise the solution must first be transformed to a modal basis before the DPT filter can be applied. Following the interpolant projection, scales are separated by projecting the resolved field back and forth to a lower-order polynomial approximation. Within this thesis, two distinct interpolation filters, namely a *Lagrange filter* and a L^2*-filter*, have been implemented, see Fig. 2.4.

Hereafter, we briefly describe the projection mechanism used in this thesis. According to the DGFEM approach the (unfiltered) resolved scales $\tilde{u}(x)$ and

[5] In the case of a modal basis, the shape/basis functions form a hierarchical set, i.e. each additional shape function implies progressively higher spatial frequencies.



(a) Lagrange filter (b) L2 filter

Figure 2.4: Illustration of the filtering via the interpolant projection ($p = 5$, $q = 3$). Complete field (bold solid line); large resolved scales (solid line); small-resolved scales (dashed line); high-order interpolation nodes (•); low-order interpolation nodes (□).

the large resolved scales $\bar{\tilde{u}}(x)$ are written as

$$\tilde{u}(x) = \sum_{i=0}^{N} \mathbf{u}_i^p \phi_i^p(x) \quad \text{and} \quad \bar{\tilde{u}}(x) = \sum_{i=0}^{N} \bar{\mathbf{u}}_i^p \phi_i^p(x). \tag{2.58}$$

Here, the shape functions ϕ_i are the Lagrange interpolation polynomials of degree at most p. $\mathbf{u}_i^p$ and $\bar{\mathbf{u}}_i^p$ denote the corresponding expansion weights. Once the low-pass filter has been applied, the small scales are computed as

$$\tilde{u}^s = \tilde{u} - \bar{\tilde{u}}. \tag{2.59}$$

Lagrange filter. In a first step, the resolved scales are interpolated by a polynomial approximation of lower degree $q < p$, i.e.

$$\bar{\tilde{u}}(x) = \sum_{i=0}^{N} \bar{\mathbf{u}}_i^q \phi_i^q(x) \quad \text{with} \quad \bar{\mathbf{u}}^q = \mathbf{A}^{MN} \mathbf{u}^p. \tag{2.60}$$

For the nodal Lagrange basis, the projection operator $\mathbf{A}^{MN}$ is given by

$$\mathbf{A}_{ij}^{MN} = \phi_j^p(x_i^q) \tag{2.61}$$

with x_i^q the interpolation node corresponding to ϕ_i^q. Equation (2.61) results from the following property, $\phi_i(x_j) = \delta_{ij}$.

In a second step, the low order approximation (2.60) is projected back to the polynomial space p. The corresponding expansion weights of the large scales are finally computed as

$$\bar{\mathbf{u}}^p = \mathbf{A}^{NM} \mathbf{A}^{MN} \mathbf{u}^p. \tag{2.62}$$

L^2-filter. As for the Lagrange filter, the scale separation is realized in two steps. First, the resolved scales are projected to a lower order polynomial approximation of degree q. In order to minimize the weighted residual over the element T, the projection operator is defined as

$$\int_T \left(\sum_{k=0}^{N} \mathbf{u}_k^p \phi_k^p(x) - \sum_{l=0}^{M} \bar{\mathbf{u}}_l^q \phi_l^q(x) \right) \phi_i^q(x) dV = 0 \quad i = 0, \ldots, M. \tag{2.63}$$

In a matrix-vector form, equation (2.63) reads

$$\bar{\mathbf{u}}^q = \left(\mathbf{M}^{MM} \right)^{-1} \mathbf{M}^{MN} \mathbf{u}^p. \tag{2.64}$$

Similarly to the mass matrix in chapter 4, the matrix operator $\mathbf{M}^{MN}$ is given by

$$\mathbf{M}_{ij}^{MN} = \int_T \phi_i^q \phi_j^p dV. \tag{2.65}$$

Again, the low-order approximation is projected back to the original (complete) function space, leading the the following expansion weights

$$\bar{\mathbf{u}}^p = \left(\mathbf{M}^{NN} \right)^{-1} \mathbf{M}^{NM} \left(\mathbf{M}^{MM} \right)^{-1} \mathbf{M}^{MN} \mathbf{u}^p. \tag{2.66}$$

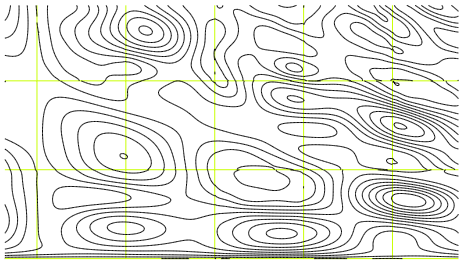
In Fig. 2.5, the Lagrange and L^2 filtering of a sinusoidal field ($p = 5$) are compared. In contrast to the Lagrange interpolation, the L^2 filter breaks the C^0 continuity of the low-order field. However, according to the work of Blackburn and Schmidt [15], the loss of C^0 continuity does not cause problems, if the filtered field is used to construct eddy viscosity estimates.

2.5 Conclusions

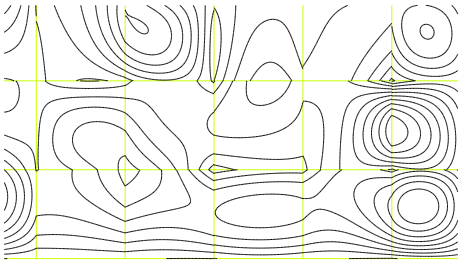
The Reynolds-averaged Navier-Stokes equations have been introduced and turbulence closure using the one-equation Spalart-Allmaras model has been presented. We have seen that a poor mesh resolution leads to negative values of the turbulence working variable $\tilde{\mu}$ in the outer boundary layer, causing the instability of the computation.

To overcome this problem different modifications to the S-A model have been compared. In particular we have adapted the changes proposed by Oliver [129] to a different version of the S-A model. The aim is to improve the stability of the method by reducing the undershoots of $\tilde{\mu}$ while conserving the C^1-continuity of the model at $\tilde{\mu} = 0$. The resulting implementation has been compared to a simple clipped model similar to but less intrusive than the one used Landmann *et al.* [108].

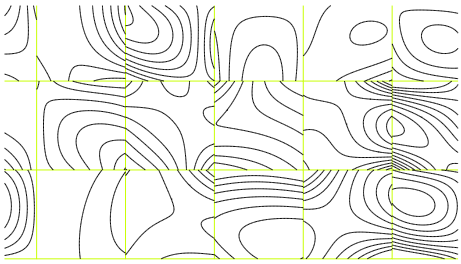
Finally, motivated by the need for more accurate methods to simulate turbulent flows, the basic features of the *variational multiscale method* VMS have been presented. Although the Smagorinsky and the WALE SGS turbulence



(a) Unfiltered field ($p = 5$)



(b) Lagrange filter ($q = 2$)



(c) L2 filter ($q = 2$)

Figure 2.5: Comparison of filtering methods for a sinusoidal field ($p = 5$). In contrast to the Lagrange interpolation, the L^2 filtering breaks the C^0 continuity.

model are similar to standard LES, improved results are expected by a second scale separation into large resolved and small resolved eddies. In this thesis, the SGS filtering is achieved explicitly using either a Lagrange or a L2 projection.

Chapter 3

The Discontinuous Galerkin method

Contents

Given the many attractive numerical properties of the discontinuous Galerkin finite element methods (DGFEM), it may surprise that they remained relatively neglected over more than 30 years since their first appearance in the early 1970s. In section 3.1 we briefly summarize the evolution of the DG approach whereby attention will be focused on the comparison of the advantages/drawbacks with regard to other popular discretization schemes. Next

the main concepts of the discontinuous Galerkin method - and in particular of the interior penalty discretization - are outlined in section 3.2. The weak imposition of the boundary conditions associated to the RANS system is presented in section 3.3. Finally, we discuss some important discretization aspects of the RANS equations in section 3.4. On the one hand we had to modify the transpose term in order to ensure the stability of the IP method in the case of a fully coupled resolution of the RANS system. On the other hand different expressions of the penalty parameter are proposed leading to a significant improvement of the nonlinear solver. Several numerical examples are presented in section 3.5 to illustrate the effect of (i) changes of the S-A model, and (ii) the definition of the penalty coefficient on the robustness of the interior penalty RANS solver.

3.1 Historical overview

Originally introduced in 1973 by Reed and Hill [140] for solving the neutron transport equation, the discontinuous Galerkin method has been applied over the last years to a wide range of problems of practical engineering interest. A detailed review of the development of the DG method from its beginning in the early 1970s through the year 2000 is provided by Cockburn *et al.* [38]. Only the most relevant advances are summarized here.

While the DG method has known a fast growing success for solving (linear) advection problems, a field where the continuous (conforming) finite element method does not work well, its application to elliptic equations is relatively new. In 1974, one year after the introduction of the DG method, LeSaint and Raviart [111] studied its mathematical aspects. They were the first to provide a priori error estimates of the newly invented method. Assuming that the exact solution is smooth, they proved a rate of convergence of $\mathcal{O}(h^p)$ in the L^2-norm for general triangulations. Later, Johnson and Pitkäranta [94] proved a rate of convergence of $\mathcal{O}(h^{p+1/2})$, while Richter showed that a rate of $\mathcal{O}(h^{p+1})$ can be achieved in L^2, if the characteristic direction is almost aligned with the grid. The theoretical results for general triangulations have been confirmed numerically by Peterson [135]. Since then, a number of convergence proofs assuming less severe conditions on the smoothness of the exact solution have been proposed.

Given its increasing popularity for solving linear advection problems, the discontinuous Galerkin method has been extended to non-linear hyperbolic systems in the beginning of the 1980s. Inspired by the finite volume method, Chavent and Salzano [31] incorporated for the first time in 1982 an approximate Riemann solver into a DG space discretization to handle the discontinuities at cell interfaces.

The breakthrough of the DG method came at the end of the 1980s. Up to this time, the main difficulty was to find a time discretization that would

result in a stable, efficient and formally high-order accurate method [38]. To overcome the limitations of previous DG approaches, Cockburn and Shu introduced, in 1989, the Runge-Kutta Discontinuous Galerkin (RKDG) method. Initially constructed for the piecewise linear discretization of one-dimensional scalar equations, the RKDG method has been progressively generalized to a high-order accurate scheme for multi-dimensional hyperbolic systems. Thereto, the method combines a DG-space discretization with a high-order accurate explicit TVD Runge-Kutta scheme and a generalized slope limiter.

Somewhat independent of the development of DG methods for advection problems, its application to diffusion problems received little attention over many years. The origin of the penalty formulation traces back to the late 1960s when Lions [116] used a penalty formulation to weakly impose Dirichlet boundary conditions. Based on the observation that the same approach could be used to weakly enforce inter-element continuity, different authors used (interior) penalty (IP) methods in the 1970s to solve elliptic problems, e.g. [7,50]. Probable reasons why the IP methods fell into oblivion in the 1980s are: (i) they were never proved to perform better than conforming FE methods for elliptic problems; (ii) the stability of the IP methods depends strongly on the choice of the penalty parameters [4].

As a consequence of the popularity of the DG methods for advection problems and given the attractive properties they exhibit (see below), tremendous efforts were made since the 1990s to extend these methods to advection-diffusion equations. Within this context, Bassi and Rebay were one of the first to use a mixed formulation to handle diffusion operators (first Bassi-Rebay (BR1) scheme). In this approach, the problem is reformulated as a system of first order PDEs by applying a DG-space discretization to both $\mathbf{U}$ and $\nabla\mathbf{U}$ which are considered as independent unknowns [38]. A stabilization of this approach, the so-called *Local Discontinuous Galerkin (LDG)* method, was introduced in 1998 by Cockburn and Shu [39]. Since then, several modifications have been proposed to eliminate the additional unknowns, to improve the coercivity and to reduce the stencil of the scheme, e.g. the *second Bassi-Rebay (BR2)* scheme [12] or the *Compact Discontinuous Galerkin (CDG)* method of Peraire and Persson [130].

In parallel to the appearance of the mixed formulation, we observe a regained interest in the interior penalty methods for solving the Navier-Stokes equations, e.g. [49,68,77,82]. A comparison between several mixed and penalty formulations has been provided by Arnold *et al.* [4].

Whereas in the last decade discontinuous Galerkin methods became quite popular for solving laminar aerodynamic flows, their application to the RANS equations is rather new. At the time of this writing, only four examples were known to the author. Bassi and Rebay [8] considered the BR2 scheme to discretize the RANS equations coupled with the $k-\omega$ turbulence model. Nguyen, Persson and Peraire [125] used the CDG scheme while turbulence was described by the S-A model. Landmann [107] applied both LDG and

BR2 scheme to the discretization of the RANS equations using either the $k-\omega$ or the S-A turbulence model. Oliver [129] examined the application of the BR2 scheme using three different discretizations of the source term of the S-A model.

The reason for the increasing interest in the discontinuous Galerkin methods in recent years, is that they inherit the attractive numerical properties of both finite element and finite volume methods without suffering from their principal limitations [36]:

- *High-order accuracy.* As in the case of conforming finite element methods, high-order accurate schemes are easily obtained by increasing the polynomial order of the interpolants. In contrast, high-order finite volume or finite difference methods suffer generally from an enlarged computation stencil that deteriorates the efficiency and the stability.

- *Upwinding.* Like finite volume methods, the DG methods offer a natural way to introduce the necessary upwinding to ensure the convective stability. Furthermore, in contrast to continuous Galerkin discretizations, which satisfy only a global mass balance over the whole computational domain, the DG methods are locally conservative [143]. This makes them ideally suited for applications where convection is important.

- *Hybrid between structured/unstructured method.* The method can be considered as a hybrid between a structured and an unstructured approach. On the one hand, as the attention is focused on the computations in an individual element, the DG methods are better suited than finite difference schemes to handle unstructured grids and complicated geometries. On the other hand, each element behaves as a structured block allowing a computationally efficient implementation using dense matrix/vector operations.

- *Boundary conditions.* Boundary conditions are imposed in the same way as inter-element fluxes. Unlike finite differences no particular numerical treatment is necessary in order to ensure uniform high-order accuracy. Note that the weak imposition of boundary conditions generally encountered with the DG methods is a matter of taste. Indeed, nothing forbids to impose the boundary conditions strongly as in the case of continuous finite elements.

- *Block-diagonal mass matrix.* As a result of the element-wise discontinuous function space, the DG methods are characterized by a block-diagonal mass matrix. Since the block size is related to the number of unknowns inside each element, they can be inverted once and for all. Moreover, the block-diagonal structure of the mass matrix reduces

p	Triangle	Quadrilateral	Tetrahedron	Hexahedron
1	6	4	20	8
2	3	2.25	6.25	3.38
3	2.22	1.78	3.70	2.37
4	1.88	1.56	2.73	1.95
p	$\frac{(p+2)(p+1)}{p^2}$	$\frac{(p+1)^2}{p^2}$	$\frac{5(p+1)(p+2)(p+3)}{6p^3}$	$\frac{(p+1)^3}{p^3}$

Table 3.1: Ratio of the number of unknowns in a discontinuous to the number of unknowns in a continuous finite element method, assuming structured grids. In the limit as $p \to \infty$, the ratio tends to one.

the communications between processors to a strict minimum[1]. The DG methods become thus highly parallelizable.

- *hp-adaptation.* Since no continuity requirement is imposed on neighbouring elements, the DG method provides an easy and efficient way to deal with adaptation strategies and an arbitrary number of hanging nodes. In contrast, continuous finite element discretizations allow at most one hanging node per edge/face, in which case special continuous basis functions have to be used [143]. Furthermore, because the interface jumps are proportional to the local discretization error, adaptation criteria can be based on a simple evaluation of the jumps amplitudes.

Although the discontinuous Galerkin finite element approach presents a lot of attractive properties, it has its drawbacks, like any other numerical methods. Currently, the main disadvantages of high-order DG schemes are related to the robustness and the computational efficiency.

- *Increased number of unknowns.* Since neighbouring elements do not share common degrees of freedom at their interface, the discontinuous discretization leads, for an identical number of elements, to a larger number of variables than a FE method using continuous basis functions. Table 3.1 illustrates the additional computational effort of the DG methods for some commonly used two- and three-dimensional elements as a function of the polynomial order p. Note that the computational surplus decreases for increasing p. It follows that for low-order polynomial degrees such as commonly used by continuous FE methods, the DG methods seem inefficient [32]. Compared to classical finite volume methods, high-order DG approximations require the solution of significantly more discrete equations [105].

[1] As we will see later, the bandwidth of the global matrix principally depends on the discretization of the viscous terms.

- *Discretization of elliptic operators.* In contrast to the continuous FE methods, where second order spacial terms (diffusion) are handled rather easily, their discretization becomes more complex if discontinuous test functions are utilized. While mixed methods considerably enlarge the number of unknowns, the choice of the somewhat arbitrary penalty parameter strongly influences the stability of the penalty methods. Although this choice can be automated by the use of lifting operators, the latter are quite expensive and significantly increase the computational cost of the method. The discretization of the viscous terms is discussed in details in section 3.2.3.

3.2 The Discontinuous Galerkin-space discretization

In this section, we introduce the basic concept of the Discontinuous Galerkin method. Thereto, we consider a general system of convection-diffusion-source equations:

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\mathcal{L}}(\mathbf{U}, \nabla \mathbf{U}) = \boldsymbol{\mathcal{S}}(\mathbf{U}, \nabla \mathbf{U}), \tag{3.1}$$

with

$$\boldsymbol{\mathcal{L}}(\mathbf{U}, \nabla \mathbf{U}) = \nabla \cdot \boldsymbol{\mathcal{F}}^c(\mathbf{U}) - \nabla \cdot \boldsymbol{\mathcal{F}}^v(\mathbf{U}, \nabla \mathbf{U}), \tag{3.2}$$

where $\mathbf{U}$ is the conservative state vector[2]. The operator $\boldsymbol{\mathcal{L}}$ regroups the convective (inviscid) and diffusive (viscous) fluxes defined in chapter 2, $\boldsymbol{\mathcal{F}}^c$ and $\boldsymbol{\mathcal{F}}^v$, respectively. $\boldsymbol{\mathcal{S}}$ represents the source term. The equations are defined on the domain $\Omega \subset \mathbb{R}^d$ (with $d = 2$ or 3), and appropriate boundary conditions are prescribed on its boundary $\partial\Omega$.

3.2.1 The weak formulation

To discretize in space, we proceed as follows. First, we define a tessellation $\mathcal{T}$ of the domain Ω into a finite number of non-overlapping elements T. In 2D, we usually choose triangles or quadrilaterals as (discretization) elements. In 3D, T can be e.g. tetrahedra, hexahedra or prisms. By multiplying (3.1) by an arbitrary, smooth "test" or "weighting" function $\mathbf{v}$ and integrating over the domain Ω, we obtain the weighted residual formulation

$$\begin{aligned}
&\int_\Omega \mathbf{v} \frac{\partial \mathbf{U}}{\partial t}\, d\Omega + \int_\Omega \mathbf{v} \left(\boldsymbol{\mathcal{L}}(\mathbf{U}, \nabla \mathbf{U}) - \boldsymbol{\mathcal{S}}(\mathbf{U}, \nabla \mathbf{U}) \right)\, d\Omega \\
&\quad = \sum_{T \in \mathcal{T}} \int_T \mathbf{v} \frac{\partial \mathbf{U}}{\partial t}\, d\Omega + \sum_{T \in \mathcal{T}} \int_T \mathbf{v} \left(\boldsymbol{\mathcal{L}}(\mathbf{U}, \nabla \mathbf{U}) - \boldsymbol{\mathcal{S}}(\mathbf{U}, \nabla \mathbf{U}) \right)\, d\Omega \\
&\quad = 0.
\end{aligned} \tag{3.3}$$

[2] In the case of the Navier-Stokes equations, the conservative state vector reads $\mathbf{U} = [\rho, \rho u, \rho v, \rho w, \rho E]^T$. If the S-A model is considered, we have $\mathbf{U} = [\rho, \rho u, \rho v, \rho w, \rho E, \tilde{\mu}]^T$.

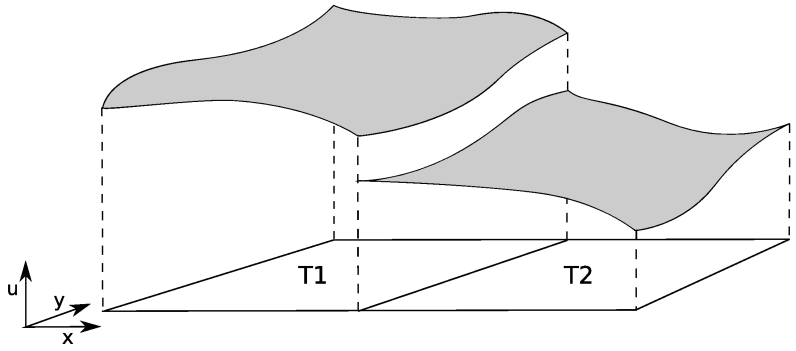


Figure 3.1: Discontinuous approximation on elements T_1 and T_2.

Next, in order to derive a discrete analogous of (3.3), we replace the smooth weighting function $\mathbf{v}$ by $\mathbf{v}_h \in \mathbf{\Phi}_h^v$ and the exact solution $\mathbf{U}(\mathbf{x}, t)$ by $\mathbf{u}_h \in \mathbf{\Phi}_h^u$, where $\mathbf{\Phi}_h^v$ and $\mathbf{\Phi}_h^u$ denote appropriate function spaces supported by the elements T of the tessellation. In this work, a standard finite element method that uses the same function space $\mathbf{\Phi}_h$ for both, the approximate solution and the weighting functions, is considered. Thereto, we define on each element a set of linearly independent (scalar) shape functions ϕ_i^T, such that

$$\begin{aligned} \mathbf{u}_h(\mathbf{x}, t)|_T &= \sum_{i=1}^{N_\phi} \mathbf{U}_i(t)\phi_i^T(\mathbf{x}), \\ \mathbf{v}_h(\mathbf{x})|_T &= \sum_{i=1}^{N_\phi} \mathbf{V}_i\phi_i^T(\mathbf{x}) \qquad \forall \mathbf{x} \in T, \end{aligned} \tag{3.4}$$

where the expansion weights $\mathbf{U}_i(t)$ and $\mathbf{V}_i$ are the degrees of freedom of the numerical solution and of the test function. N_ϕ denotes the number of shape functions.

A particularity of the Discontinuous Galerkin method is that no global continuity is required for $\mathbf{u}_h$ and $\mathbf{v}_h$. In contrast to the continuous finite element method, the discrete solution is continuous only inside the elements but discontinuous at the interfaces; as illustrated by Fig. 3.1. Hence, we have very few limitations on the shape functions ϕ_i. Note that computational efficiency is the only reason why in practice we choose shape functions having a local support, i.e.

$$\phi_i^T(\mathbf{x}) = \begin{cases} \phi_i^T(\mathbf{x}) & \text{if } \mathbf{x} \in T, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

Otherwise, not only the shape functions defined on T but also those defined on other elements would have to be taken into account in the evaluation of the volume integrals. To avoid confusion, the superscript T indicating the element on which ϕ_i^T is defined, is dropped for the remainder of this thesis.

Furthermore, it is perfectly possible to define distinct shape functions ϕ_{im} not only with respect to the interpolation point i but also with respect to the physical variable m. Such an approach is for instance used in the case of incompressible flows where the pressure is discretized by a function space which is one order lower than the one of the remaining variables. For simplicity we introduce the discontinuous Galerkin method assuming, without loss of generality, that the same shape functions are used for all equations, i.e. $\phi_{im} = \phi_i \; \forall m$. The generalization to distinct function spaces is straightforward.

In this work, a Lagrangian function space is used. Denoting by $\mathbb{P}^p$ the space of polynomials of order p, we seek an approximation $\mathbf{u}_h(\cdot, t) \in \mathbf{\Phi}^p$ with

$$\mathbf{\Phi}^p \equiv \{\mathbf{v} \in [L^2(\Omega)]^{N_v} \; : \; \mathbf{v}|_T \circ f_T \in [\mathbb{P}^p(T_{ref})]^{N_v}, \forall T \in \mathcal{T}\}, \tag{3.6}$$

such that

$$\sum_{T \in \mathcal{T}} \int_T \mathbf{v}_h \frac{\partial \mathbf{u}_h}{\partial t} \, d\Omega + \mathbf{R}_h(\mathbf{u}_h, \mathbf{v}_h) = 0, \qquad \forall \mathbf{v}_h \in \mathbf{\Phi}^p, \tag{3.7}$$

where

$$\mathbf{R}_h(\mathbf{u}_h, \mathbf{v}_h) = \mathbf{R}_{h,I}(\mathbf{u}_h, \mathbf{v}_h) + \mathbf{R}_{h,V}(\mathbf{u}_h, \mathbf{v}_h) + \mathbf{R}_{h,S}(\mathbf{u}_h, \mathbf{v}_h). \tag{3.8}$$

Here, $\mathbf{R}_{h,I}$, $\mathbf{R}_{h,V}$ and $\mathbf{R}_{h,S}$ denote the discretization of the inviscid, the viscous and the source term, respectively. Furthermore, f_T defines a smooth bijective mapping, i.e. the relation between the local coordinate system in the reference space $\boldsymbol{\xi}$ and the global (physical) coordinates $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$. N_v is the length of the state vector and $L^2(\Omega)$ represents the space of functions which are square-integrable over Ω.

Note that equation (3.7) must be satisfied for all elements $T \in \mathcal{T}$ and for every weighting function $\mathbf{v}_h|_T \in \mathbf{\Phi}^p$. As $\mathbf{v}_h|_T$ is a linear combination of the shape functions ϕ_i (having a local support only), (3.7) defines a system of N_ϕ equations for each element

$$\int_T \phi_i \frac{\partial \mathbf{u}_h}{\partial t} \, d\Omega + \mathbf{R}_h|_T(\mathbf{u}_h, \phi_i) = 0, \qquad i = 1, \dots N_\phi, \; \forall T \in \mathcal{T}. \tag{3.9}$$

Notations

Before looking in detail at the discretization of the different terms, we first introduce some notations that are used in the remainder of this chapter.

Let T be an element of the tessellation $\mathcal{T}$ of the domain Ω, we use ∂T to denote the edges / faces of T. Furthermore, we define the union Γ of the boundaries of the elements T, i.e.

$$\Gamma = \bigcup_{T \in \mathcal{T}} \partial T, \tag{3.10}$$

as well as the union of the interior faces $\Gamma^0 = \Gamma \setminus \partial\Omega$. Finally, the notation $(\cdot)^-$ and $(\cdot)^+$ refers to the left and right states at the interface between two neighbouring elements T^- and T^+ and $\mathbf{n}$ is the unit normal vector pointing from $+$ to $-$.

Local mapping

As stated in the introduction one feature of the finite element method is that the computation is focused on individual elements. This allows not only to easily handle complex geometries but also significantly improves the computational efficiency. Since the direct integration in the physical space would be by far too costly, a widespread approach is to define a *local coordinate system* in each element. The *assembly* of all individual contributions to the global system is achieved by the *mapping*. The mapping describes the relation between the global (physical) coordinates $\mathbf{x}(x, y, z)$ and the local (reference) coordinates $\boldsymbol{\xi}(\xi, \eta, \zeta)$. It can be written as

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_i \hat{\phi}_i(\boldsymbol{\xi}) \mathbf{X}_i. \tag{3.11}$$

In this work, the geometry is represented using a nodal Lagrange basis (cf. section 3.2.1) for the geometric interpolants $\hat{\phi}_i(\boldsymbol{\xi})$.

Fig. 3.2 illustrates the affine transformation between local and global coordinate systems in the case of a forth order quadrangle and a third order triangle. For instance, we map each (possibly curved) quadrangle onto a canonical square taken as $(\xi, \eta) \in [-1, 1] \times [-1, 1]$.

The Jacobian of the transformation is

$$\mathbf{J} = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{bmatrix} \partial_\xi x & \partial_\xi y & \partial_\xi z \\ \partial_\eta x & \partial_\eta y & \partial_\eta z \\ \partial_\zeta x & \partial_\zeta y & \partial_\zeta z \end{bmatrix}. \tag{3.12}$$

Accordingly, the solution gradient in physical space is computed by a simple application of the chain rule

$$\begin{aligned} \nabla U &= \sum_i U_i \nabla_{\boldsymbol{\xi}} \phi_i \cdot \nabla_{\mathbf{x}} \boldsymbol{\xi} \\ &= \sum_i U_i \nabla_{\boldsymbol{\xi}} \phi_i \cdot \mathbf{J}^{-1}. \end{aligned} \tag{3.13}$$

Obviously, in order to conserve the global accuracy, high order methods need a high order representation of the boundaries. Otherwise the discrete solution will resolve exactly the physics of the polygonal boundary surface causing entropy production or local drops in density; see Fig. 3.3. This not only deteriorates the accuracy but also affects the stability of the method if the density is reduced to values close to zero in some grid points. Bassi

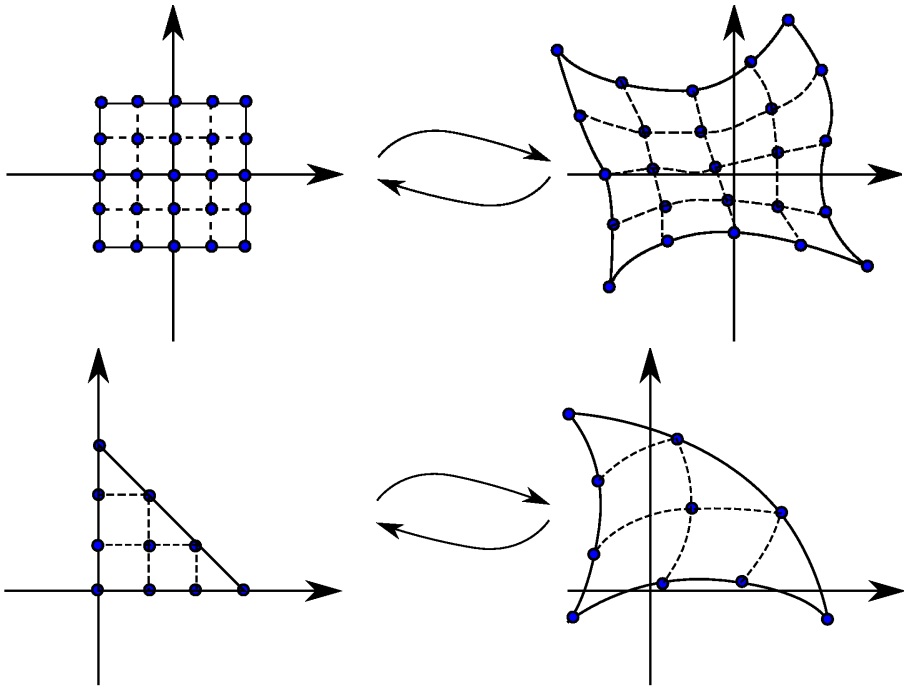


Figure 3.2: Mapping between reference space $\boldsymbol{\xi}$ and physical space $\mathbf{x}(\boldsymbol{\xi})$.

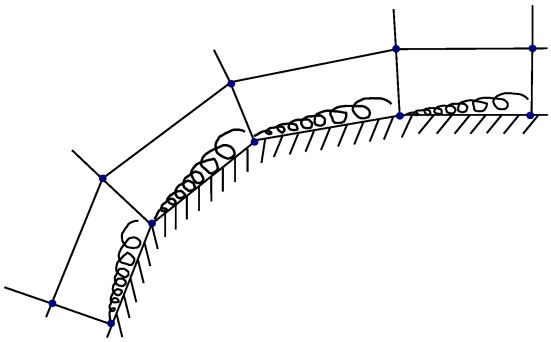


Figure 3.3: Entropy production induced by discontinuities of the surface normals in the case of a polygonal boundary representation.

and Rebay [10] were one of the first to show the sensitivity of high-order DG methods to the boundary representation.

We should however note that a simple increase of the polynomial order q of the geometric basis functions $\hat{\phi}$ may not always lead to the expected results. Studying a circular channel flow Oliver [128] observed a drop of the order of accuracy when using $q = 3$ instead of $q = 2$ for the geometry representation. He suspected oscillations in the geometry interpolation at the origin of this phenomenon. An appealing alternative to more standard Lagrange interpolations would therefore be the use of Bézier splines (see Luo *et al.* [118]) since they are less prone to oscillations [87].

Finally, we should remember that the basis functions are defined on the canonical element. Accordingly, except in the case of straight elements, polynomials of order p in the reference space lead to an approximate solution $\mathbf{u}_h$ of order less than p in the physical space.

Function space

To solve the interpolation problem (3.4) we set up a nodal Lagrange basis on each element[3]. Accordingly, the shape function $\phi_i(\boldsymbol{\xi})$ in (3.5) is a polynomial of degree p which satisfies the following relations

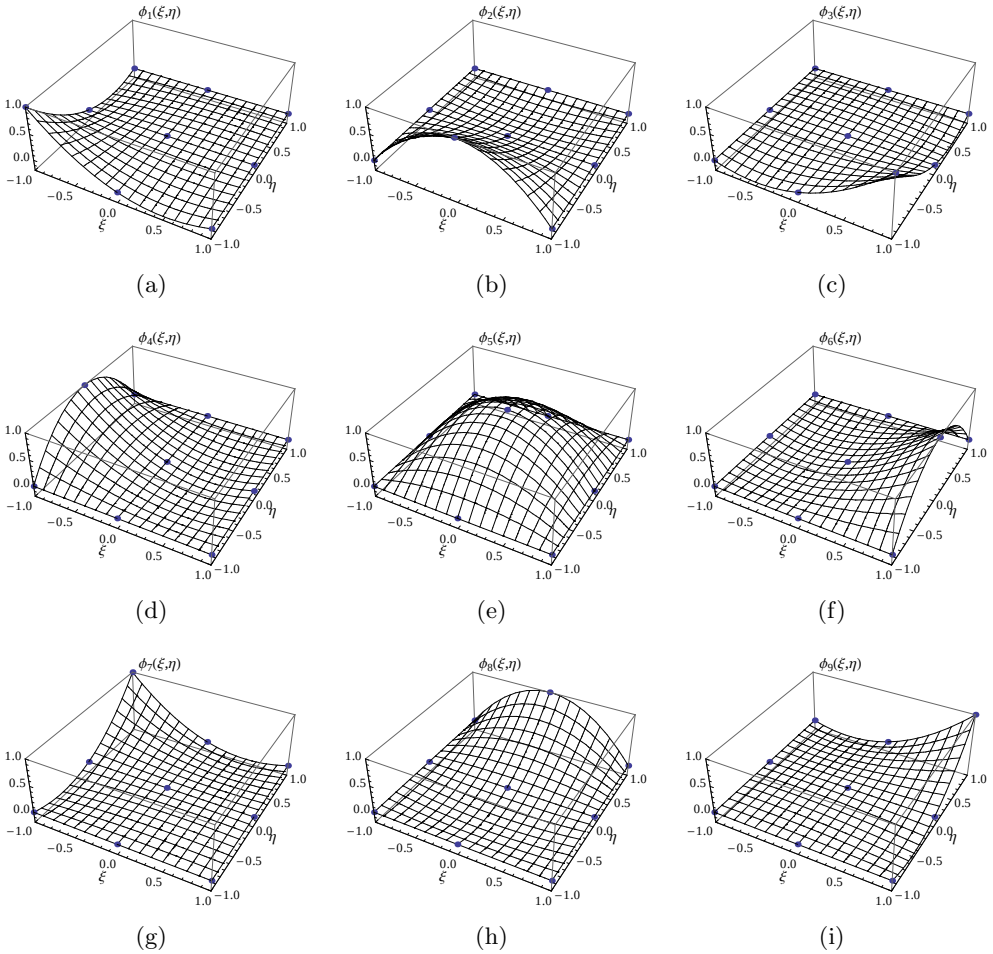
$$\phi_i(\boldsymbol{\xi}_j) = \begin{cases} 1 & \text{if} \quad \boldsymbol{\xi}_j = \boldsymbol{\xi}_i, \\ 0 & \text{if} \quad \boldsymbol{\xi}_j \neq \boldsymbol{\xi}_i, \end{cases} \qquad 1 \leq i, j \leq N_\phi, \tag{3.14}$$

with $\boldsymbol{\xi}_i$ the control or interpolation point associated to ϕ_i. In this work we use standard interpolation points which are uniformly spaced in the reference element; cf. Fig. 3.2 and 3.4. In 1D, an explicit expression for the Lagrange basis functions is given by

$$\phi_i(\xi) = \prod_{\substack{0 \leq j \leq N_\phi \\ j \neq i}} \frac{(\xi - \xi_j)}{(\xi_i - \xi_j)}. \tag{3.15}$$

An appealing property of the resulting "closed" Lagrangian interpolation formula is the efficient evaluation of face integrals since only basis functions associated to control points belonging to this face have to be considered [87]. Fig. 3.4 illustrates this property in the case of $p = 2$ Lagrangian interpolants defined on the canonical square. If we consider for instance the edge $e\colon \eta = -1$, we notice that only ϕ_1, ϕ_2 and ϕ_3 are active, whilst all the other shape functions, whose corresponding control point is not located on e, are identically zero on e. It follows that on e, the discrete solution (3.4) restricts to

$$\mathbf{u}_h^e(\boldsymbol{\xi}, t) = \sum_{i=1}^{3} U_i(t)\phi_i(\boldsymbol{\xi}). \tag{3.16}$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.4: Lagrangian interpolation of degree $p = 2$ on a canonical square. The only shape functions that are not identically zero on any edge e are those associated to an interpolation point located on e. This property allows a very efficient evaluation of face integrals.

The two main disadvantages of the present Lagrange basis are

- *Poor interpolation quality for large p.* This behaviour known as Runge's phenomenon is related to the largely oscillatory character (usually near the edges of the element) of high-order polynomials when using equally spaced control points[4]. A possible remedy to improve the interpolation quality for high-order polynomials would be the use of (i) a different net of interpolation points in order to minimize the Runge's phenomenon (e.g. Tchebycheff nodes), or (ii) a modal basis of orthogonal polynomials (e.g. Legendre or Gram interpolation). For an overview of polynomial interpolation methods we refer to Dahlquist *et al.* [45]. We should however mention that for polynomials up to degree $p \leq 4$ as used in this work, Lagrangian shape functions provide an excellent interpolation quality.

- *Absence of a hierarchical order.* Contrary to the family of modal bases, the Lagrangian shape functions do not form a hierarchical set. As a consequence, if one wants to increase the order of the polynomial approximation one can not simply add new functions to the existing basis. Since changing the polynomial degree affects each shape function, the whole basis must be redefined. Compared to modal bases the Lagrangian interpolation is thus less suited for p-adaptation, p-filtering or p-multigrid acceleration techniques. Note that p-filtering is particularly useful in LES because it allows an inexpensive elementwise separation of the flow scales.

Numerical integration: Gauss quadrature

All integrals are computed numerically by means of Gauss quadrature performed in the reference element. To illustrate the numerical integration, we consider the diffusive volume term (3.60) discussed further in this chapter (cf. section 3.2.3). Approximating the integral over element T by a weighted sum, we obtain

$$\int_T \nabla\phi_i \cdot \boldsymbol{\mathcal{F}}_m^v \, d\Omega \approx \sum_{q=1}^{N_q} w_q \left(|\mathbf{J}_T| \sum_{k=1}^{d} \sum_{l=1}^{d} \frac{\partial \phi_i}{\partial \xi_l} \frac{\partial \xi_l}{\partial x_k} \boldsymbol{\mathcal{F}}_m^v \right)_{\nu_q}, \tag{3.17}$$

[3] Remember that the shape functions ϕ_i are not directly defined in the physical coordinate system but in the reference space associated to the canonical element.

[4] Let $f(x) = 1/(1+25x^2)$ on the interval $x \in [-1,1]$ and $p_n(x)$ a polynomial of order n. By exploring the error of polynomial interpolations, Runge [151] has found that when using equally spaced interpolation points, the error difference $|f(x) - p_n(x)|$ grows without bound as $n \to \infty$.

where w_q and ν_q define the weight and position of the q-th quadrature point respectively. N_q represents the number[5] of quadrature points and $|\mathbf{J}_T|$ is the determinant of the mapping Jacobian.

To preserve the accuracy of the DG discretization, the integration order $\mathcal{O}$ must be chosen depending on (i) the polynomial degree p of the discrete solution $\mathbf{u}_h$, (ii) the (possible) non-linearity of the state equations and (iii) the grid curvature. Note that in the case of curved elements the inverse of the mapping Jacobian $\mathbf{J}^{-1} \equiv \partial\xi_l/\partial x_k$ is a rational function. As a consequence the quadrature formula (3.17) does not allow to compute the exact integral. A handbook of practical integration rules for various element types is provided by Solin *et al.* [162].

In practice for *laminar flows on straight meshes* we choose the number of quadrature points such as to exactly integrate a polynomial of order

$$\mathcal{O} \equiv 2p + 1 \quad \text{(laminar case)}. \tag{3.18}$$

In this way the quadrature formula should not deteriorate the performance of the DG method. Indeed, considering a polynomial interpolation of degree p, we have for the inertia term in equation (3.7)

$$\phi \cdot U \sim \left(h^p + \mathcal{O}(h^{p+1})\right) \cdot \left(h^p + \mathcal{O}(h^{p+1})\right) = h^{2p} + \mathcal{O}(h^{2p+1}) \tag{3.19}$$

with h a measure of the local mesh size. Similarly, if we assume that the leading term of the convective flux $\boldsymbol{\mathcal{F}}^c$ is proportional to the state vector $\mathbf{U}$, the discretization error of the convective volume term is given by

$$\nabla\phi \cdot \boldsymbol{\mathcal{F}}^c(U) \sim \left(h^{p-1} + \mathcal{O}(h^p)\right) \cdot \left(h^p + \mathcal{O}(h^{p+1})\right) = h^{2p-1} + \mathcal{O}(h^{2p}) \tag{3.20}$$

while for the viscous[6] (volume) term the error is proportional to $\mathcal{O}(h^{2p-1})$. It follows that by imposing $\mathcal{O} = 2p + 1$ the truncation error of the Gauss quadrature is one order of magnitude lower than the discretization error.

It turned however out according to our numerical experiments that the above integration rule is insufficient in the case of *turbulent flows*, even when straight meshes are used. On the one hand, the source term of the Spalart-Allmaras turbulence model is a highly non-linear function of the solution vector, on the other hand, the diffusive stress tensor is scaled by the eddy viscosity. The latter is a rational function of the S-A working variable $\tilde{\mu}$. As a consequence we suggest to use an integration rule that allows to exactly integrate a polynomial of degree

$$\mathcal{O} \geq 3p \quad \text{(turbulent case)}. \tag{3.21}$$

[5] In 1D, N_q quadrature points are required to exactly integrate a polynomial of degree $p = N_q - 1$.

[6] Since in this work only Newtonian fluids are considered, the viscous flux vector is a linear function of the velocity gradients. Assuming that the density varies slowly, we have $\boldsymbol{\mathcal{F}}^v(\mathbf{U}, \nabla\mathbf{U}) = h^{p-1} + \mathcal{O}(h^p)$.

3.2.2 Discretization of the inviscid term

Considering a generic element T, the discretization of the inviscid term reads, after integration by parts,

$$\begin{aligned}\mathbf{R}_{h,I}|_T(\mathbf{u}_h,\phi_i) &\equiv \int_T \phi_i \left(\nabla\cdot\boldsymbol{\mathcal{F}}^c(\mathbf{u}_h)\right)\, d\Omega \\ &= \oint_{\partial T} \phi_i \boldsymbol{\mathcal{F}}^c(\mathbf{u}_h)\cdot\mathbf{n}\, dS - \int_T \nabla\phi_i\cdot\boldsymbol{\mathcal{F}}^c(\mathbf{u}_h) d\Omega. \end{aligned} \tag{3.22}$$

In contrast to continuous finite element methods, where the discrete solution is single-valued at the interface between two neighbouring elements T^- and T^+, the DG formulation requires no global continuity. As a consequence, the flux vector $\boldsymbol{\mathcal{F}}^c(\mathbf{u}_h)$ takes different values depending on whether $\mathbf{u}_h^-$ or $\mathbf{u}_h^+$ is considered. Thus, as in the case of finite volume methods, the discretization of the inviscid flux leads to a Riemann problem[7]. To ensure the information transfer between adjacent elements, which otherwise would be completely decoupled, we have to introduce a numerical flux function $\boldsymbol{\mathcal{H}}(\mathbf{u}_h^-,\mathbf{u}_h^+,\mathbf{n})$ which depends on both, the left and the right state. This function determines the accuracy and the robustness of the scheme. However, according to numerical experiments of Cockburn [36], the choice of the numerical flux becomes less important as the polynomial order p of the shape functions is increased. This is to be expected, since the solution jump across cell interfaces is directly related to the integral of the residual over these cells [37]. In other words, if the differential equations are well approximated by the function space $\boldsymbol{\Phi}^p$, the jumps are small and $\boldsymbol{\mathcal{H}}(\mathbf{u}_h^-,\mathbf{u}_h^+,\mathbf{n})$ tends to the exact flux (provided that the numerical flux function is consistent).

Roe's approximate Riemann solver [146], the preference of choice in this work, is presented in the next section.

Finally, by summing (3.22) over all elements of the domain, the global residual of the inviscid term is given by

$$\begin{aligned}\mathbf{R}_{h,I}(\mathbf{u}_h,\phi_i) = &\sum_{b\in\partial T\setminus\partial\Omega}\oint_b \phi_i\boldsymbol{\mathcal{H}}(\mathbf{u}_h^-,\mathbf{u}_h^+,\mathbf{n})\, dS + \sum_{b\in\partial T\cap\partial\Omega}\oint_b \phi_i\boldsymbol{\mathcal{H}}^b(\mathbf{u}_h^-,\mathbf{u}^b,\mathbf{n})\, dS \\ &- \sum_{T\in\mathcal{T}}\int_T \nabla\phi_i\cdot\boldsymbol{\mathcal{F}}^c(\mathbf{u}_h)\, d\Omega, \end{aligned} \tag{3.23}$$

where the superscript b refers to appropriate boundary values defined in section 3.3. Note that, given the local support of the shape functions ϕ_i, (3.23)

[7] We call a *Riemann problem* the initial-value problem for a hyperbolic system of conservation laws, i.e. $\partial\mathbf{u}/\partial t + \partial\mathbf{F}(\mathbf{u})/\partial x = 0$, presenting a single discontinuity defined by $\mathbf{u}(x,0)\equiv\mathbf{u}_L(x<0)$, $\mathbf{u}(x,0)\equiv\mathbf{u}_R(x>0)$.

can alternatively be written as

$$\mathbf{R}_{h,I}(\mathbf{u}_h, \phi_i) = \oint_{\Gamma^0} (\phi_i^+ - \phi_i^-) \boldsymbol{\mathcal{H}}(\mathbf{u}_h^-, \mathbf{u}_h^+, \mathbf{n})\, dS + \oint_{\partial\Omega} \phi_i^- \boldsymbol{\mathcal{H}}^b(\mathbf{u}_h^-, \mathbf{u}^b, \mathbf{n})\, dS - \sum_{T\in\mathcal{T}} \int_T \nabla\phi_i \cdot \boldsymbol{\mathcal{F}}^c(\mathbf{u}_h)\, d\Omega. \tag{3.24}$$

Roe flux difference splitting

Henceforth, we summarize some of the basic features of Roe's approximate Riemann solver, one of the most popular advection schemes.

As shown previously, the discretization of the three-dimensional Euler equations leads to a Riemann problem along the direction normal to the cell interface. Denoting by ξ the local coordinate in this direction, the Euler equations along ξ written in quasi-linear form are given by

$$\frac{\partial \mathbf{U}}{\partial t} + \underbrace{\left(\sum_{j=1}^{3} \frac{\partial \boldsymbol{\mathcal{F}}_{x_j}^c}{\partial \mathbf{U}} n_j \right)}_{\boldsymbol{A}_n(\mathbf{U})} \cdot \frac{\partial \mathbf{U}}{\partial \xi} = 0, \tag{3.25}$$

where $\mathbf{A}_n(\mathbf{U})$ is the projected Jacobian matrix. Because solving the exact (non-linear) equations (3.25) is very expensive since iterative methods are generally required, Roe's idea consists in computing the exact solution to an approximate (linearized) Riemann problem, i.e.

$$\frac{\partial \mathbf{U}}{\partial t} + \tilde{\boldsymbol{A}}_n(\mathbf{U}_L, \mathbf{U}_R) \cdot \frac{\partial \mathbf{U}}{\partial \xi} = 0. \tag{3.26}$$

Thereto, the Jacobian matrix $\mathbf{A}_n(\mathbf{U})$ is replaced by a specially constructed constant matrix $\tilde{\boldsymbol{A}}_n(\mathbf{U}_L, \mathbf{U}_R)$ whose value depends on the left and the right states at the interface. To ensure the monotonicity of the scheme, the numerical normal flux $\boldsymbol{\mathcal{H}}(\mathbf{U}_L, \mathbf{U}_R, \mathbf{n})$ must be Lipschitz continuous, consistent and monotone. These requirements are fulfilled, if the linearized Jacobian matrix satisfies the following properties:

1. $\tilde{\boldsymbol{A}}_n(\mathbf{U}_L, \mathbf{U}_R)$ has real eigenvalues $\tilde{\lambda}_k$ and linearly independent eigenvectors $\tilde{\mathbf{r}}_k$. This condition preserves the hyperbolic nature of the linearized Euler equations.

2. The linearized Jacobian $\tilde{\boldsymbol{A}}_n$ is consistent with the exact Jacobian, i.e.

$$\tilde{\boldsymbol{A}}_n(\mathbf{U}, \mathbf{U}) = \boldsymbol{A}_n(\mathbf{U}).$$

3. Conservation across discontinuities

$$\boldsymbol{\mathcal{F}}_n^c(\mathbf{U}_L) - \boldsymbol{\mathcal{F}}_n^c(\mathbf{U}_R) = \tilde{\boldsymbol{A}}_n(\mathbf{U}_L - \mathbf{U}_R).$$

Henceforth, Δ represents the difference between right and left states. Denoting by $\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_k)$ the diagonal matrix of eigenvalues and by $\tilde{\boldsymbol{R}}$ the matrix of right eigenvectors $\mathbf{r}_k$, the numerical normal inviscid flux through the cell interface is calculated from

$$\boldsymbol{\mathcal{H}}(\mathbf{U}_L, \mathbf{U}_R, \mathbf{n}) = \underbrace{\frac{1}{2}\left[\boldsymbol{\mathcal{F}}_n^c(\mathbf{U}_L) + \boldsymbol{\mathcal{F}}_n^c(\mathbf{U}_R)\right]}_{\text{central scheme}} - \underbrace{\frac{1}{2}|\tilde{\boldsymbol{A}}_n|\Delta\mathbf{U}}_{\text{dissipation}}, \tag{3.27}$$

with

$$|\tilde{\boldsymbol{A}}_n| = \tilde{\boldsymbol{R}}|\tilde{\boldsymbol{\Lambda}}|\tilde{\boldsymbol{R}}^{-1}. \tag{3.28}$$

From (3.27) it follows that Roe's flux formula is equivalent to a central scheme plus an artificial dissipation term (e.g. Delanaye [47] for the amount of dissipation introduced in the case of a 1D advection equation).

It has been shown [89] that the linearized Jacobian can be expressed as

$$\tilde{\boldsymbol{A}}_n(\mathbf{U}_L, \mathbf{U}_R) = \boldsymbol{A}_n(\tilde{\mathbf{U}}), \tag{3.29}$$

where $\tilde{\mathbf{U}}(\mathbf{U}_L, \mathbf{U}_R)$ represent appropriate Roe-averaged values. For the 3D Euler equations, these values are

$$\tilde{\rho} = \sqrt{\rho_L \rho_R}, \tag{3.30a}$$

$$\tilde{u} = \frac{\sqrt{\rho_L}u_L + \sqrt{\rho_R}u_R}{\sqrt{\rho_L + \rho_R}}, \tag{3.30b}$$

$$\tilde{v} = \frac{\sqrt{\rho_L}v_L + \sqrt{\rho_R}v_R}{\sqrt{\rho_L + \rho_R}}, \tag{3.30c}$$

$$\tilde{w} = \frac{\sqrt{\rho_L}w_L + \sqrt{\rho_R}w_R}{\sqrt{\rho_L + \rho_R}}, \tag{3.30d}$$

$$\tilde{H} = \frac{\sqrt{\rho_L}H_L + \sqrt{\rho_R}H_R}{\sqrt{\rho_L + \rho_R}}. \tag{3.30e}$$

For detailed information about the construction of $\tilde{\boldsymbol{A}}$ and the properties of the Roe flux we refer to [89].

Numerical implementation of the RANS equations

In order to achieve a Newton-type convergence at the last stage of the time integration, the turbulence model is solved together with the main flow equations in a strong coupled way. Thereto, Geuzaine [71] proposes in his thesis an extension of Roe's approximate Riemann solver to take into account the turbulence model. However, motivated by the following considerations a simplified implementation of Roe's solver has been adopted in this work.

1. As stated previously, the choice of the numerical flux becomes less important as the polynomial order of the shape functions is increased; see Cockburn [36].

2. To improve the robustness of finite volume methods, it is common practice to increase the numerical dissipation, by using a constant reconstruction for the turbulence model, while higher order reconstructions are used for the mean flow equations (e.g. [71, 139]). Since we use the same approximation for all variables, including the turbulence working variable, a slightly increased dissipation by means of a simpler flux function for the turbulence model might be beneficial for the stability of the method. Furthermore, as it will be seen in chapter 5, the interpolation of the S-A turbulence model has only a negligible effect on the overall accuracy, even for coarse meshes, which justifies the above approach.

3. Given the relative high complexity of the Roe solver and in order to take full advantage of the C++ object-oriented programming, it is preferable to reuse as much as possible existing routines/code segments.

Argo, the code developed within this thesis, is written in modern C++. A natural way to handle multiple conservation laws such as Euler, Navier-Stokes, RANS equations... consists in defining a basic class which provides a number of general functions (e.g. initialization of common variables, finite-difference implementation of Jacobian functions...). All conservation laws are derived from this basic class. Moreover, the class of RANS equations inherits the class of the Navier-Stokes system.

If one wants to treat the whole set of RANS equations as one single block to solve the Riemann problem, the entire Roe solver (and its linearization) must be rewritten for each new turbulence model. This would make the implementation of new models quite complex, especially since different routines are required for 2D and 3D computations.

In this work, we have opted for a simplified method, which consists in separating main flow variables and turbulence model for the evaluation of the inviscid term. Henceforth, we denote by

$$\boldsymbol{\mathcal{H}}(\mathbf{U}_L, \mathbf{U}_R, \mathbf{n}) = \begin{bmatrix} \boldsymbol{\mathcal{H}}^{NS} \\ \ldots\ldots \\ \boldsymbol{\mathcal{H}}^{SA} \end{bmatrix} \tag{3.31}$$

the complete numerical inviscid flux vector, whereas $\boldsymbol{\mathcal{H}}^{NS} = [\mathcal{H}^{\rho}, \mathcal{H}^{\rho u}, \mathcal{H}^{\rho v}, \mathcal{H}^{\rho w}, \mathcal{H}^{\rho E}]^T$ and $\boldsymbol{\mathcal{H}}^{SA}$ represent the flux vectors for the Euler/Navier-Stokes equations and the Spalart-Allmaras model, respectively.

The computation of the numerical inviscid flux occurs in two stages. In a first step, we simply neglect the presence of a turbulence model and $\boldsymbol{\mathcal{H}}^{NS}$

is computed by applying a "standard" Roe solver to the Euler equations. Thereto, the normal inviscid flux

$$\boldsymbol{\mathcal{F}}_n^c(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \\ \cdots\cdots\cdots \\ u\tilde{\mu} \end{bmatrix} n_x + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \\ \cdots\cdots\cdots \\ v\tilde{\mu} \end{bmatrix} n_y + \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \\ \cdots\cdots\cdots \\ w\tilde{\mu} \end{bmatrix} n_z = \begin{bmatrix} \boldsymbol{\mathcal{F}}_n^{c,NS} \\ \cdots\cdots \\ \boldsymbol{\mathcal{F}}_n^{c,SA} \end{bmatrix} \tag{3.32}$$

has been restricted to $\boldsymbol{\mathcal{F}}_n^{c,NS}$ in the Roe flux formula (3.27).

Now, based on the numerical flux obtained for the continuity equation $\mathcal{H}^\rho$, a simple upwind scheme is used, in a second step, to discretize the advective term of the turbulence model. The latter is calculated from

$$\boldsymbol{\mathcal{H}}^{SA}(\mathbf{U}_L, \mathbf{U}_R, \mathbf{n}) = \begin{cases} \frac{\tilde{\mu}^-}{\rho^-}\mathcal{H}^\rho & \mathcal{H}^\rho \geq 0, \\ \frac{\tilde{\mu}^+}{\rho^+}\mathcal{H}^\rho & \mathcal{H}^\rho < 0. \end{cases} \tag{3.33}$$

3.2.3 Discretization of the viscous term

In contrast to the convective flux, the viscosity of the Spalart-Allmaras model depends not only on the solution $\mathbf{U}$ but also on the gradient $\nabla\mathbf{U}$ of the solution. Consequently, the divergence of the viscous flux introduces second order derivatives into the governing equations, that cannot be handled directly in a weak formulation using a discontinuous function space [9]. This makes the discretization of the viscous term much more challenging than the construction of a DGFEM scheme for hyperbolic equations. An overview of various DGFEM approaches for elliptic problems can be found in Arnold *et al.* [4].

The first approach to solve elliptic equations using DGFEM goes back on a generalization of Nitsche's method [127] in the 1970's. The idea of this penalty approach consists in handling connections between elements in the same weak manner as Dirichlet boundary conditions.

Later, a different strategy has become popular leading to several new discretization schemes. In a first step, appropriate auxiliary variables are defined in order to rewrite the second order partial differential equation into a system of first order equations. The coupled system is then discretized by applying the Galerkin variational formulation, together with the introduction of suitable numerical flux functions. The resulting discrete form is commonly referred to as *flux* or *mixed formulation*. However, given the high memory requirements due to the additional degrees of freedom, especially for an implicit time scheme, it is generally preferable to eliminate the auxiliary variables in order to reduce the size of the underlying system. This can be achieved by lifting operators, cf. [4, 8, 9], and leads to the so-called *primal formulation*.

The properties of the resulting DGFEM scheme highly depend on the definition of the lifting operator. In particular, the use of global lifting operators can lead to suboptimal convergence rates for piecewise polynomial approximations of odd order [8, 39]. Furthermore, some lifting operators are unsuitable for purely elliptic problems. It is also worth noting that the computation of the lifting operators presents a non negligible cost. Indeed, for a non-linear problem, up to 30 % of the computational effort to evaluate the non-linear residual is related to the lifting operator [84].

In this work, we have opted for the family of *Interior Penalty (IP)* methods. As it will be seen later in this section, the IP-DGFEM has a local support, implying only direct neighbours. This allows a very efficient parallelization of the method and makes it particularly appealing for large scale CFD applications. In addition, the IP method defines an explicit expression of the lifting operator, which compared to the second Bassi-Rebay scheme (BR2) [8] considerably reduces the computational effort.

As stated previously, the starting point in order to adopt a similar technique as for the discretization of the convective term is to reformulate problem (3.1)-(3.2) as a system of first order PDE[8]

$$\boldsymbol{\sigma} = \boldsymbol{\mathcal{D}}(\mathbf{U})\nabla\mathbf{U}, \tag{3.34}$$

$$\nabla \cdot \boldsymbol{\sigma} = \text{RHS}, \tag{3.35}$$

with the auxiliary variables[9] $\boldsymbol{\sigma} \in \mathbb{R}^{N_v \otimes d}$, N_v denoting the length of the state vector and d is the number of space dimensions ($d \leq 3$ in practice). Furthermore, we have introduced the following first order expansion of the diffusive flux $\boldsymbol{\mathcal{F}}^v$ with respect to the solution gradients:

$$\boldsymbol{\mathcal{F}}^v_{k,m}(\mathbf{U}, \nabla\mathbf{U}) = \sum_{n=1}^{N_v} \sum_{l=1}^{d} D^{kl}_{mn}(\mathbf{U}) \frac{\partial U_n}{\partial x_l}, \quad m = 1 \dots N_v, \tag{3.36}$$

$$\boldsymbol{\mathcal{D}}(\mathbf{U}) \equiv \partial\boldsymbol{\mathcal{F}}^v(\mathbf{U}, \nabla\mathbf{U})/\partial\nabla\mathbf{U}, \tag{3.37}$$

where k and l denotes space directions and the subscript m refers to the m-th equation of the underlying system. Note that (3.36) is exact, because in the particular case of a Newtonian fluid, $\boldsymbol{\mathcal{F}}^v$ is a linear function of the velocity gradients. RHS ("right-hand side") groups all remaining terms of the governing equations such as time dependency, inviscid and source terms. Their discretization will be discussed in another section. To improve the readability of the formulae, we henceforth use the Einstein summation convention.

[8] Although the IP method has initially been introduced in the primal form, the use of the mixed formulation has become quite popular because it facilitates a comparison with other discretization schemes.

[9] Note that in the remainder of this section, $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ denote auxiliary variables which will be eliminated later; and should not be confused with the stress tensor introduced in chapter 2. We have conserved this notation because it is the standard notation in the literature.

An alternative approach to the system (3.34)-(3.35) would be to rewrite the model problem in function of $\nabla\mathbf{U} \in \mathbb{R}^{N_v \otimes d}$, resulting in the following first order system

$$\begin{aligned} \boldsymbol{\sigma} &= \nabla\mathbf{U}, \\ \nabla \cdot (\boldsymbol{\mathcal{D}}\boldsymbol{\sigma}) &= \text{RHS}, \end{aligned} \tag{3.38}$$

with

$$\nabla\mathbf{U} = \begin{bmatrix} \partial_x\rho & \partial_y\rho & \partial_z\rho \\ \partial_x\rho u & \partial_y\rho u & \partial_z\rho u \\ \partial_x\rho v & \partial_y\rho v & \partial_z\rho v \\ \partial_x\rho w & \partial_y\rho w & \partial_z\rho w \\ \partial_x\rho E & \partial_y\rho E & \partial_z\rho E \\ \partial_x\tilde{\mu} & \partial_y\tilde{\mu} & \partial_z\tilde{\mu} \end{bmatrix}. \tag{3.39}$$

For DGFEM schemes based on the system (3.38) we refer to [8, 9, 107].

The next step in order to derive a discretization of the viscous term consists in applying the Galerkin variational formulation to the coupled system (3.34) and (3.35). Thereto, in addition to the scalar[10] shape functions $\phi_i \in \Phi^p$ already introduced in section 3.2.1, a vector function space $\boldsymbol{\Sigma}_h$ is defined as

$$\begin{aligned} \Phi^p &\equiv \{\phi \in L^2(\Omega) \mid \phi \circ f_T \in \mathbb{P}^p(T_{ref}), \forall T \in \mathcal{T}\}, \\ \boldsymbol{\Sigma}_h &\equiv \{\boldsymbol{\tau} \in [L^2(\Omega)]^d \mid \boldsymbol{\tau} \circ f_T \in [\mathbb{P}^p(T_{ref})]^d, \forall T \in \mathcal{T}\}. \end{aligned} \tag{3.40}$$

Multiplying (3.34) with $\boldsymbol{\tau} \in \boldsymbol{\Sigma}_h$ and (3.35) with $\phi_i \in \Phi^p$ respectively, and summing over all elements, we obtain after integration by parts

$$\int_\Omega \boldsymbol{\sigma}_{h_m} \cdot \boldsymbol{\tau}\, d\Omega = \sum_{T\in\mathcal{T}} \left[\oint_{\partial T} \hat{u}_{h_n} D^{kl}_{mn} \tau_k n_l \, dS - \int_T u_{h_n} \frac{\partial}{\partial x_l} \left(D^{kl}_{mn} \tau_k \right) \, d\Omega \right], \tag{3.41}$$

$$\int_\Omega \phi_i \nabla \cdot \boldsymbol{\sigma}_{h_m} \, d\Omega = \sum_{T\in\mathcal{T}} \left[\oint_{\partial T} \phi_i \hat{\boldsymbol{\sigma}}_{h_m} \cdot \mathbf{n} \, dS - \int_T \nabla\phi_i \cdot \boldsymbol{\sigma}_{h_m} \, d\Omega \right] = \text{RHS}^*. \tag{3.42}$$

Here, similarly to the inviscid numerical flux $\boldsymbol{\mathcal{H}}(\cdot,\cdot,\cdot)$ in section 3.2.2, $\hat{\mathbf{u}}_h$ and $\hat{\boldsymbol{\sigma}}_h$ represent appropriate vector and tensor numerical fluxes, respectively. For clarity and in order to avoid confusion with tensor products, we prefer an index notation.

A common way to write the discretization of the viscous term in a compact form is to use jump $[\![\cdot]\!]$, and average $\langle\cdot\rangle$ operators. Let q be a scalar and $\boldsymbol{\varphi}$ a

[10] Whereas $\boldsymbol{\Phi}^p$ represents a vector space that contains the discrete solution $\mathbf{u}_h$, we use Φ^p to denote the scalar space formed by the shape functions ϕ_i.

vector, these operators are defined as

$$\begin{aligned} [\![q]\!] &= q^+\mathbf{n}^+ + q^-\mathbf{n}^-, & \langle q\rangle &= (q^+ + q^-)/2, \\ [\![\boldsymbol{\varphi}]\!] &= \boldsymbol{\varphi}^+ \otimes \mathbf{n}^+ + \boldsymbol{\varphi}^- \otimes \mathbf{n}^-, & \langle \boldsymbol{\varphi}\rangle &= (\boldsymbol{\varphi}^+ + \boldsymbol{\varphi}^-)/2. \end{aligned} \tag{3.43}$$

Furthermore, a straightforward computation shows that

$$\sum_{T\in\mathcal{T}} \oint_{\partial T} q_T \boldsymbol{\varphi}_T \cdot \ \mathbf{n}\, dS = \oint_\Gamma [\![q]\!] \cdot \langle \boldsymbol{\varphi}\rangle \ dS + \oint_{\Gamma^0} \langle q\rangle\, [\![\boldsymbol{\varphi}]\!]\, dS. \tag{3.44}$$

A new integration by part of the volume integrals in (3.41) and (3.42), followed by the application of the above identity (3.44) leads to

$$\begin{aligned} \int_\Omega \boldsymbol{\sigma}_{h_m} \cdot \boldsymbol{\tau}\, d\Omega &= \sum_{T\in\mathcal{T}} \left[\int_T \tau_k D^{kl}_{mn} \frac{\partial u_{h_n}}{\partial x_l}\, d\Omega + \oint_{\partial T} (\hat{u}_{h_n} - u_{h_n}) D^{kl}_{mn} \tau_k n_l\, dS \right] \\ &= \sum_{T\in\mathcal{T}} \int_T \tau_k D^{kl}_{mn} \frac{\partial u_{h_n}}{\partial x_l}\, d\Omega + \oint_\Gamma [\![\hat{u}_{h_n} - u_{h_n}]\!] \cdot \left\langle D^{kl}_{mn}\tau_k \right\rangle \ dS \\ &\quad + \oint_{\Gamma^0} \langle \hat{u}_{h_n} - u_{h_n}\rangle\, [\![D^{kl}_{mn}\tau_k]\!]\, dS, \end{aligned} \tag{3.45}$$

$$\begin{aligned} \int_\Omega \phi_i \nabla\cdot\boldsymbol{\sigma}_{h_m}\, d\Omega &= -\sum_{T\in\mathcal{T}} \int_T \nabla\phi_i \cdot \boldsymbol{\sigma}_{h_m}\, d\Omega + \oint_\Gamma [\![\phi_i]\!] \cdot \langle \hat{\boldsymbol{\sigma}}_{h_m}\rangle \ dS \\ &\quad + \oint_{\Gamma^0} \langle \phi_i\rangle\, [\![\hat{\boldsymbol{\sigma}}_{h_m}]\!]\, dS. \end{aligned} \tag{3.46}$$

If we assume that appropriate numerical fluxes have been defined for $\hat{\mathbf{u}}_h$ and $\hat{\boldsymbol{\sigma}}_h$, system (3.45)-(3.46) can be solved using a two-step method. First, at the beginning of each time step, the auxiliary variables $\boldsymbol{\sigma}_h$ are computed from (3.45). This can be done very efficiently, thanks to the block diagonal structure of the mass matrix. The updated gradients are then used to solve the main flow equations (3.46).

Due to the absence of privileged propagation directions for elliptic operators, a natural choice consists in defining the numerical flux functions as the average between the two interface states, i.e. $\hat{\mathbf{u}}_h = \langle \mathbf{u}_h\rangle$ and $\hat{\boldsymbol{\sigma}}_h = \langle \boldsymbol{\sigma}_h\rangle$. The resulting central scheme is known in literature as the *first Bassi-Rebay (BR1)* scheme [9]. The large amount of degrees of freedom, suboptimal convergence for odd order polynomials and its ill-posedness for purely elliptic problems are the principal drawbacks of this method. To overcome these limitations, Cockburn and Shu [39] proposed in 1998 a generalization of the BR1 approach, the *Local Discontinuous Galerkin (LDG)* method. Here, the numerical fluxes do not only depend on the averages $\langle\cdot\rangle$ but also on the interface jumps $[\![\cdot]\!]$. For detailed information on these schemes, we refer to the cited literature. A comparison between various DGFEM approaches including stability analysis and error estimates can be found in Arnold *et al.* [4].

As already mentioned, the major drawback of the BR1 and LDG approaches is their important memory requirement due to the additional degrees of freedom. It is therefore common practice to eliminate the auxiliary variables. This can be achieved by integrating by parts the volume integral $\int_T \nabla\phi_i \cdot \boldsymbol{\sigma}_{h_m}\, d\Omega$ in (3.46), taking $\boldsymbol{\tau} = \nabla\phi_i$ and substituting (3.45) into (3.46). The resulting discrete form for the m-th equation of the viscous term then reads:

$$\begin{aligned}\mathbf{R}_{h,V_m}(\mathbf{u}_h, \phi_i) = &- \sum_{T\in\mathcal{T}} \int_T \frac{\partial \phi_i}{\partial x_k} D^{kl}_{mn} \frac{\partial u_{h_n}}{\partial x_l}\, d\Omega \\ &- \oint_\Gamma \left([\![\hat{\boldsymbol{u}}_{h_n} - \boldsymbol{u}_{h_n}]\!] \cdot \left\langle D^{kl}_{mn} \frac{\partial \phi_i}{\partial x_l} \right\rangle - [\![\phi_i]\!] \cdot \langle \hat{\boldsymbol{\sigma}}_{h_m} \rangle \right) dS \\ &- \oint_{\Gamma^0} \left(\langle \hat{u}_{h_n} - u_{h_n} \rangle [\![D^{kl}_{mn} \frac{\partial \phi_i}{\partial x_l}]\!] - \langle \phi_i \rangle [\![\hat{\boldsymbol{\sigma}}_{h_m}]\!] \right) dS. \end{aligned} \tag{3.47}$$

In the literature, (3.47) is commonly known as *primal formulation*.

To complete the definition of the discrete scheme, we must specify the remaining numerical fluxes. In this work, we have chosen an *Interior Penalty* method. The properties of this family of DGFEM approaches have been studied extensively by Georgoulis [68] for a convection-diffusion-reaction equation, and by Hartmann and Houston [82–84] in the case of the compressible Navier-Stokes equations. For an edge $\partial T \in \Gamma^0$ which lies inside the domain Ω, we take

$$\hat{\mathbf{u}}_h = \langle \mathbf{u}_h \rangle \qquad \text{and} \qquad \hat{\boldsymbol{\sigma}}_{h_m} = \langle \boldsymbol{\mathcal{F}}^v_m(\mathbf{u}_h, \nabla \mathbf{u}_h) \rangle - \delta [\![\mathbf{u}_{h_m}]\!], \tag{3.48}$$

while for boundary edges $\partial T \in \partial\Omega$ we write

$$\hat{\mathbf{u}}_h = \mathbf{u}^b_h(\mathbf{u}^-_h) \qquad \text{and} \qquad \hat{\boldsymbol{\sigma}}_{h_m} = \boldsymbol{\mathcal{F}}^{v,b}_m(\mathbf{u}^-_h, \nabla \mathbf{u}^-_h) - \delta [\![\mathbf{u}^b_{h_m} - \mathbf{u}^-_{h_m}]\!]. \tag{3.49}$$

where the superscript b refers to boundary values, and δ is a penalty weighting function.

Finally, by substituting the numerical fluxes (3.48) and (3.49) into (3.47), the IP-DGFEM formulation of the viscous term is: find $u_h \in \boldsymbol{\Phi}^p$ such that $\forall m$, $\forall \phi_i$

$$\begin{aligned}\mathbf{R}_{h,V_m}(\mathbf{u}_h, \phi_i) = &- \sum_{T\in\mathcal{T}} \int_T \nabla\phi_i \cdot \boldsymbol{\mathcal{F}}^v_m\, d\Omega + \oint_{\Gamma^0} \delta [\![u_{h_m}]\!] [\![\phi_i]\!]\, dS \\ &+ \oint_{\Gamma^0} [\![\phi_i]\!]_k \left\langle D^{kl}_{mn} \frac{\partial u_{h_n}}{\partial x_l} \right\rangle dS + \theta \oint_{\Gamma^0} [\![u_{h_n}]\!]_k \left\langle D^{kl}_{nm} \frac{\partial \phi_i}{\partial x_l} \right\rangle dS \\ &+ DBD + DBN + DBP. \end{aligned} \tag{3.50}$$

Here, DBD, DBN and DBP denote appropriate boundary terms that will be further discussed in section 3.3. The sign of the 4^{th} term is determined by θ, which equals 1 for the *Symmetric (SIPDG)* and -1 for the *Non-Symmetric Interior Penalty Discontinuous Galerkin Method (NIPDG)*.

3.2.4 Discretization of the source term

The source term is discretized using a straightforward test function weighted approach. Multiplying the source term by the test function and integrating over the domain, we get

$$\mathbf{R}_{h,S}(\mathbf{u}_h, \phi_i) = \sum_{T \in \mathcal{T}} \int_T \phi_i \mathbf{S}(\mathbf{u}_h, \nabla \mathbf{u}_h) \, d\Omega. \tag{3.51}$$

In his thesis, Oliver [129] refers to this method as "standard weighting approach". He has shown that, contrary to the BR2 [8] and LDG [39] scheme, where an auxiliary variable is solved for the state gradients $\nabla \mathbf{u}_h$, formulation (3.51) leads to an *adjoint inconsistent* discretization, see section 3.2.5. Note however that adjoint consistency can be obtained by including additional interface terms, which are derived from the source operator.

3.2.5 Properties of the interior penalty method

Before summarizing the main features of the presented interior penalty method, some definitions currently used within the discontinuous Galerkin context are in order. The definitions given here are reproduced from [78, 129, 143] and transcribed to our notations.

Definitions

To introduce the concept of consistency/adjoint consistency, we consider the non-linear steady problem

$$R(\mathbf{u}) = 0 \quad \text{in } \Omega, \qquad B(\mathbf{u}) = \mathbf{g} \quad \text{on } \partial\Omega, \qquad \mathbf{u} \in \boldsymbol{\mathcal{V}}, \tag{3.52}$$

on the open bounded domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$, where R is a non-linear differential operator, and B is a boundary operator. $\boldsymbol{\mathcal{V}}$ denotes an appropriate function space.

Then, let $\boldsymbol{\mathcal{V}}_h$ be a discrete function space on a tessellation $\mathcal{T}$ of Ω and consider a general discontinuous Galerkin approximation of the *primal problem* (3.52): find $\mathbf{u}_h \in \boldsymbol{\mathcal{V}}_h$ such that

$$R_h(\mathbf{u}_h, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in \boldsymbol{\mathcal{V}}_h, \tag{3.53}$$

where $R_h : \boldsymbol{\mathcal{V}}_h \times \boldsymbol{\mathcal{V}}_h \to \mathbb{R}$ is a semi-linear form derived from the weak formulation of (3.52).

Definition 1 *The discretization* (3.53) *defined by the semi-linear form, R_h, is said to be consistent if the exact solution* $\mathbf{u}$ *to the primal problem* (3.52) *satisfies following equation:*

$$R_h(\mathbf{u}, \mathbf{v}) = 0 \quad \forall \mathbf{v} \in \boldsymbol{\mathcal{V}}. \tag{3.54}$$

Next, let $J(\cdot) : \boldsymbol{\mathcal{V}} \to \mathbb{R}$ be a general functional of interest

$$J(\mathbf{u}) = \int_{\Omega} j_{\Omega}(u)\, d\Omega + \oint_{\partial\Omega} j_{\Gamma}(\mathbf{u})\, dS, \tag{3.55}$$

where in the context of aerodynamic applications, the boundary target functional $j_{\Gamma}(\mathbf{u})$ typically represents the lift or drag coefficient, whereas $j_{\Omega}(\mathbf{u})$ defines weighted mean-values, such as the mean density/energy inside Ω.

Furthermore, let $J_h(\cdot) : \boldsymbol{\mathcal{V}}_h \to \mathbb{R}$ be the discrete version of the functional of interest (3.55), with the Frechét derivative given by

$$J_h'[\mathbf{u}_h](\mathbf{v}_h) = \int_{\Omega} j_{\Omega}'[\mathbf{u}_h]\mathbf{v}_h\, d\Omega + \oint_{\partial\Omega} j_{\Gamma}'[\mathbf{u}_h]\mathbf{v}_h\, dS. \tag{3.56}$$

Here, $'$ denotes the Fréchet derivative and the square bracket $[\cdot]$ indicates the state about which linearization is evaluated. Similarly, we denote by $R_h'[\mathbf{u}_h](\mathbf{v}_h, \mathbf{w}_h)$ the Frechét derivative of the discrete primal problem. Then, the discrete *adjoint/dual problem* is defined by: find $\mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h$ such that

$$R_h'[\mathbf{u}_h](\mathbf{v}_h, \mathbf{w}_h) = J_h'[\mathbf{u}_h](\mathbf{v}_h), \quad \forall \mathbf{v}_h \in \boldsymbol{\mathcal{V}}_h. \tag{3.57}$$

Definition 2 *The discretization defined by the semi-linear form R_h and the target functional J_h is said to be adjoint/dual consistent if the exact solution* $\mathbf{w}$ *of the continuous adjoint problem satisfies the following equation:*

$$R_h'[\mathbf{u}](\mathbf{v}, \mathbf{w}) = J_h'[\mathbf{u}](\mathbf{v}), \quad \forall \mathbf{v} \in \boldsymbol{\mathcal{V}}. \tag{3.58}$$

Consistency and adjoint consistency play an important role to ensure optimal accuracy of a discontinuous Galerkin method. Indeed, it has been shown (see e.g. Hartmann and Houston [84])that an adjoint inconsistent discretization such as the non-symmetric interior penalty method (NIPDG) leads to suboptimal convergence rates if the error is measured in terms of:

- the L^2-norm of the primal solution[11];
- the adjoint target functional (e.g. wall friction, lift/drag coefficients,...).

Harriman *et al.* [76, 77] were amongst the first who studied the influence of adjoint consistency on the accuracy of DG discretizations of elliptic PDEs. According to them, the suboptimal convergence is caused by a lack of smoothness of the adjoint problem. Later, an adjoint consistency analysis of the interior penalty DG discretization of the compressible Euler equations has been realized by Lu [117] who demonstrated the impact of adjoint consistent boundary conditions, and Hartmann [78, 80] who provided a general framework for analyzing adjoint consistency of DG discretizations. This analysis has been extended to the compressible Navier-Stokes equations in [83, 84]. Recently, Oliver [129] proposed an adjoint consistent discretization of the Reynolds-averaged Navier-Stokes equations.

[11] For even polynomial degrees p, the convergence rate in L^2-norm was found to be one order lower than the optimal rate of $\mathcal{O}(h^{p+1})$.

Definition 3 *A numerical flux function (see section 3.2.2 and 3.2.3) is said to be conservative, if it is single-valued at the inter element boundaries.*

Definition 4 *A bilinear form R_h defined on a normed linear space $\mathcal{V}$ with norm $\|\cdot\|_{\mathcal{V}}$ is said to be coercive if there is a positive constant κ such that [143]*

$$\forall u \in \mathcal{V}, \quad \kappa \|u\|_{\mathcal{V}}^2 \leq R_h(u,u). \tag{3.59}$$

Summary

Gathering the discretization of the inviscid, viscous and source terms given by (3.23), (3.50) and (3.51) respectively, the interior penalty discontinuous Galerkin method used in this works reads: find $\mathbf{u}_h \in \mathbf{\Phi}^p$ such that $\forall m$, $\forall \phi_i$

$$\begin{aligned} r_{im} = & \underbrace{-\sum_{T\in\mathcal{T}} \int_T \nabla\phi_j \cdot \boldsymbol{\mathcal{F}}_m^c d\Omega}_{CV} \underbrace{-\sum_{T\in\mathcal{T}} \int_T \nabla\phi_i \cdot \boldsymbol{\mathcal{F}}_m^v \, d\Omega}_{DV} \underbrace{-\sum_{T\in\mathcal{T}} \int_T \phi_i \mathbf{S}_m \, d\Omega}_{SV} \\ & + \underbrace{\oint_{\Gamma^0} [\![\phi_i]\!] \boldsymbol{\mathcal{H}}_m(\mathbf{u}_h^-, \mathbf{u}_h^+, \mathbf{n}) \, dS}_{CI} + \underbrace{\oint_{\Gamma^0} [\![\phi_i]\!]_k \left\langle D_{mn}^{kl} \frac{\partial u_{h_n}}{\partial x_l} \right\rangle \, dS}_{DI} \\ & + \underbrace{\oint_{\Gamma^0} \delta [\![u_{h_m}]\!] [\![\phi_i]\!] \, dS}_{DP} + \theta \underbrace{\oint_{\Gamma^0} [\![u_{h_n}]\!]_k \left\langle D_{nm}^{kl} \frac{\partial \phi_i}{\partial x_l} \right\rangle \, dS}_{DT} \\ & + CBD + DBD + DBN + DBP. \end{aligned} \tag{3.60}$$

Here, CV, DV and SV denote the volume terms for the convective, diffusive and source fluxes, respectively. The Dirichlet boundary terms for the convective (CBD), the diffusive (DBD) and the penalty terms (DBP) are found using the same formulation as for internal faces (CI, DI, DP), and imposing the boundary conditions by providing an appropriate external state $\mathbf{u}^b$. Neumann boundary conditions (DBN) are imposed by correcting the diffusive flux. The boundary specification is discussed in more details in section 3.3. The sign of the transpose term is determined by θ, which equals 1 for the *Symmetric (SIPDG)* and -1 for the *Non-Symmetric Interior Penalty Discontinuous Galerkin Method (NIPDG)*.

We conclude this section with a summary of the most important properties of the presented IP-DGFEM approach:

- An approximate Riemann problem providing the necessary upwinding to ensure the convective stability is solved for the inviscid interface term, whereas the impact of the choice of a particular Riemann flux $\boldsymbol{\mathcal{H}}_m$ decreases rapidly as the polynomial order of the interpolant is increased.

- The diffusive interface fluxes are discretized using a central scheme explicitly stabilized by the transpose term DT (if $\theta = -1$) and the penalty term DP. Note that if the latter is omitted, the method is referred to as Baumann-Oden [13] approach, which is known to be only weakly stable.

- From (3.60) it is easily seen that the IP-DGFEM has a local support, implying only direct neighbours. This allows a very efficient parallelization of the method. Furthermore, all interface terms are defined explicitly, which reduces the computational effort by about 30% compared to the BR2 scheme. Both properties make the interior penalty approach particularly attractive for large CFD applications.

- The SIPDG as well as the NIPDG variant are both consistent and of optimal order $\mathcal{O}(h^p)$ if measured in H^1-norm. However, while the presented SIPDG scheme is adjoint consistent without the source term, the NIPDG method leads to an adjoint inconsistent discretization. This lack of adjoint consistency results in a suboptimal order of convergence $\mathcal{O}(h^p)$ when the error is measured in terms of the L^2-norm for even polynomial degrees p. For odd p, both variants achieve the optimal convergence rate of $\mathcal{O}(h^{p+1})$ in L^2-norm. Similarly, if measured in terms of target functionals $J(\cdot)$, a so-called *error doubling* $\mathcal{O}(h^{2p})$ has been observed with the symmetric SIPDG method. In contrast to this, the target functionals behave like $\mathcal{O}(h^p)$ in case of the NIPDG variant [79].

- In this work, jumps of the state variables between neighbouring elements are penalized by the scalar weighting function δ defined as

$$\delta = C_{IP}\frac{\nu}{h}, \tag{3.61}$$

  with ν the maximum viscosity at the interface, h a measure of the local mesh size (cf. section 3.4.2) and C_{IP} a positive "constant", that depends on the interpolation order p, the dimension d and the element type. For stability reasons this parameter must be chosen sufficiently large. Note that for very large values of C_{IP} the DG method tends to behave like a standard conforming finite element method. This *super-penality* [4, 144] allows the NIPDG to recover optimal order of convergence for even polynomial degrees because the consistency error can be driven down to an arbitrarily small value. However, a large penalty parameter significantly increases the condition number of the stiffness matrix, which in our experience increases the computing time and, in some cases, prevents the algorithm converging to machine accuracy.

3.3 Boundary conditions

In order to obtain a well-posed problem, the Navier-Stokes/RANS equations must be completed by additional restraints prescribed on the boundary $\partial\Omega$.

A particular feature of the discontinuous Galerkin methods is that boundary fluxes are handled in exactly the same weak manner as inter-element fluxes. Thus, no particular mathematical treatment is necessary to ensure high-order accuracy close to the boundaries. This constitutes an important advantage over finite difference schemes when complex geometries are considered.

From a practical point of view, the inviscid flow boundary conditions are weakly imposed via the Riemann solver. Similarly, Dirichlet boundary conditions for the diffusive and penalty terms are found by defining an appropriate external state and applying the same formulation as for internal faces. Neumann boundary conditions are prescribed by correcting the diffusive flux.

As for internal cells, a local outward pointing normal vector is considered. This implies that the left state $\mathbf{u}_h^-$ refers to values inside the domain, while $\mathbf{u}_h^b(\mathbf{u}_h^-, \mathbf{u}_{BC})$ and $\mathbf{u}_h^+(\mathbf{u}_h^-, \mathbf{u}_{BC})$ denote user defined external states. The latter are function of the interior values $\mathbf{u}_h^-$ and the known physical boundary data $\mathbf{u}_{BC}$. As we separately treat inviscid and viscous boundary terms, different (but compatible) exterior boundary states may be defined for each of them. In the following, $\mathbf{u}_h^b$ refers to the exterior state of the convective fluxes, while we use $\mathbf{u}_h^+$ for the diffusive and penalty terms. Details about different types of boundary conditions implemented in our code are given below.

Farfield boundary conditions: $\mathbf{U}_\infty$. Assuming that the boundary is located sufficiently far from the obstacle, this boundary condition prescribes user defined free stream values

$$\mathbf{u}_h^b = [\rho_\infty, \rho u_\infty, \rho v_\infty, \rho w_\infty, \rho E_\infty, \tilde{\mu}_\infty]^T \tag{3.62}$$

for the whole state vector. To ensure the well-posedness of the problem[12], the inviscid normal flux is computed using a Riemann solver $\boldsymbol{\mathcal{F}}_n^{c,b} = \boldsymbol{\mathcal{H}}(\mathbf{u}_h^-, \mathbf{u}_h^b)$. In addition, the viscous terms are handled by extrapolating the solution for the evaluation of the penalty term, i.e. $\mathbf{u}_h^+ = \mathbf{u}_h^- \Longleftrightarrow [\![\mathbf{u}_h]\!] = 0$, and by enforcing $\boldsymbol{\mathcal{F}}_n^{v,b} = 0$.

Subsonic inflow: T_t, p_t, α, β, $\tilde{\mu}_\infty$. According to the theory of characteristics only one characteristic wave leaves the computational domain, whereas all the remaining are incoming waves. It follows that in the case of convection-dominated three-dimensional RANS-SA equations five quantities must be specified at subsonic inflow boundaries. In this work, we use a classical approach which consists in imposing the total temperature T_t, the total pressure p_t, the inflow angles of the velocity α and β, and the turbulence working variable $\tilde{\mu}_\infty$. This choice is particularly interesting for internal flows such as turbomachinery applications because it corresponds to available physical quantities. Next, the information is combined with the velocity magnitude, extrap-

[12] For a hyperbolic system of equations, the number of boundary conditions to be imposed is equal to the number of waves entering the domain.

olated/reconstructed from the interior values, in order to define an external state $\mathbf{u}_h^b$. Then, the inviscid flux $\boldsymbol{\mathcal{F}}_n^{c,b}$ is given by

$$\boldsymbol{\mathcal{F}}_n^{c,b} = \boldsymbol{\mathcal{H}}(\mathbf{u}_h^-, \mathbf{u}_h^b), \tag{3.63}$$

while for the viscous terms, we impose

$$\mathbf{u}_h^+ = \mathbf{u}_h^- \Longleftrightarrow [\![\mathbf{u}_h]\!] = 0 \qquad \text{and} \qquad \boldsymbol{\mathcal{F}}_n^{v,b} = 0. \tag{3.64}$$

Subsonic outflow: p_∞. Under the assumption of a locally Euler flow, only one physical boundary condition is required at a subsonic outlet. A widespread choice also followed in this work is to specify the static pressure p_∞ while the density, the velocity and turbulent viscosity are extrapolated. Combining these quantities with the outflow pressure, we define the following boundary state

$$\mathbf{u}_h^b = \left[\rho^-, \rho u^-, \rho v^-, \rho w^-, \rho E^b, \tilde{\mu}^-\right]^T, \tag{3.65}$$

with the total boundary energy computed as

$$\rho E^b = \frac{p_\infty}{\gamma - 1} + \frac{1}{2}\frac{\|\boldsymbol{\rho u}^-\|^2}{\rho^-}. \tag{3.66}$$

Then, the inviscid flux $\boldsymbol{\mathcal{F}}_n^{c,b}$ is given by the Riemann solver

$$\boldsymbol{\mathcal{F}}_n^{c,b} = \boldsymbol{\mathcal{H}}(\mathbf{u}_h^-, \mathbf{u}_h^b), \tag{3.67}$$

while for the viscous terms, we impose

$$\mathbf{u}_h^+ = \mathbf{u}_h^- \Longleftrightarrow [\![\mathbf{u}_h]\!] = 0 \qquad \text{and} \qquad \boldsymbol{\mathcal{F}}_n^{v,b} = 0. \tag{3.68}$$

Isothermal, adiabatic wall. As in the case of subsonic inlet/outlet boundary conditions, we seperately treat the inviscid and the viscous boundary terms. While the advective fluxes are chosen to ensure the impermeability of the wall, the no-slip condition is imposed through the diffusive terms.

Let $\mathbf{u}_w = [u_w, v_w, w_w]^T$ be the wall velocity, the no-penetration condition is satisfied if the normal convection velocity through the boundary face is zero, i.e.

$$(\mathbf{u}^- - \mathbf{u}_w) \cdot \mathbf{n} = 0. \tag{3.69}$$

This can be achieved imposing a velocity field that is symmetric with respect to the boundary face and resulting in the following boundary state vector,

$$\mathbf{u}_h^b = \begin{bmatrix} \rho^- \\ \rho u^- - 2n_x\left[(\boldsymbol{\rho u}^- - \rho^-\mathbf{u}_w)\cdot\mathbf{n}\right] \\ \rho v^- - 2n_y\left[(\boldsymbol{\rho u}^- - \rho^-\mathbf{u}_w)\cdot\mathbf{n}\right] \\ \rho w^- - 2n_z\left[(\boldsymbol{\rho u}^- - \rho^-\mathbf{u}_w)\cdot\mathbf{n}\right] \\ \rho E^- \\ 0 \end{bmatrix}. \tag{3.70}$$

Then, the inviscid flux $\boldsymbol{\mathcal{F}}_n^{c,b}$ is given by the Riemann solver $\boldsymbol{\mathcal{F}}_n^{c,b} = \boldsymbol{\mathcal{H}}(\mathbf{u}_h^-, \mathbf{u}_h^b)$.

Taking into account the viscous effects, an additional "no-slip" boundary condition expressing the adherence of the fluid particles to the wall, i.e. $\mathbf{u} = \mathbf{u}_w$, must be satisfied. Furthermore, zero eddy viscosity at the wall and a thermal condition have to be enforced. In this regard, we distinguish between adiabatic and isothermal walls.

- *Adiabatic wall.* First, Dirichlet boundary conditions are prescribed through the penalty term providing an appropriate right state vector $\mathbf{u}_h^+$,

$$\mathbf{u}_h^+ = \left[\rho^-, \rho^- u_w, \rho^- v_w, \rho^- w_w, \rho E^+, 0\right]^T, \tag{3.71}$$

  where the density is extrapolated from the interior value and where the turbulence working variable $\tilde{\mu}$ is set to zero. The total energy is given by

$$\rho E^+ = \rho E^- - \frac{1}{2}\frac{\|\boldsymbol{\rho u}^-\|^2}{\rho^-} + \frac{1}{2}\rho^- \|\boldsymbol{u}_w\|^2. \tag{3.72}$$

  The viscous direct and transpose terms are evaluated based on the left state $\mathbf{u}_h^-$, i.e. $\boldsymbol{\mathcal{F}}_n^{v,b}(\mathbf{u}_h^-, \nabla\mathbf{u}_h^-)$ and $\boldsymbol{\mathcal{D}}^T(\mathbf{u}_h^-)\cdot\nabla\mathbf{u}_h^-$, respectively. An alternative approach would be to use the right state $\mathbf{u}_h^+$ or the average state $\langle \mathbf{u}_h^-, \mathbf{u}_h^+ \rangle$. However, in our experience, there are only negligible differences between these implementations.

  Next, the Neumann type adiabatic condition requiring a zero heat transfer is imposed by correcting the viscous flux of the energy equation. The resulting diffusive flux is computed as

$$\boldsymbol{\mathcal{F}}_n^{v,b}(\mathbf{u}_h^-, \nabla\mathbf{u}_h^-) = \left[0, \tau_{xk}^- n_k, \tau_{yk}^- n_k, \tau_{zk}^- n_k, 0, (\mu + \tilde{\mu}^-)/\sigma \nabla(\tilde{\mu}/\rho)_k^- n_k\right]^T, \tag{3.73}$$

  with τ_{ij} the viscous stress tensor introduced in section 2.2 (2.3.2) for the Navier-Stokes (RANS) equations.

- *Isothermal wall.* At isothermal walls, instead of requiring zero heat transfer, a prescribed wall temperature T_w is applied, resulting in the following right state vector

$$\mathbf{u}_h^+ = \left[\rho^+, \rho^+ u_w, \rho^+ v_w, \rho^+ w_w, \rho E^+, 0\right]^T. \tag{3.74}$$

  Here, the density at the wall ρ^+ is computed assuming a perfect gas while the total energy is taken from the interior data, i.e.

$$\begin{aligned} \rho^+ &= \frac{p^-}{RT_w} \\ \rho E^+ &= \rho E^- - \frac{1}{2}\frac{\|\boldsymbol{\rho u}^-\|^2}{\rho^-} + \frac{1}{2}\rho^+ \|\boldsymbol{u}_w\|^2. \end{aligned} \tag{3.75}$$

Then, Dirichlet boundary conditions are again enforced through the penalty term, whereas in contrast to the adiabatic wall, no Neumann boundary condition has to be imposed. Thus, the viscous flux is given by

$$\boldsymbol{\mathcal{F}}_n^{v,b}(\mathbf{u}_h^-, \nabla \mathbf{u}_h^-). \tag{3.76}$$

Symmetry plane. As the name implies, this boundary condition requires the flow state to be symmetric about the boundary face. This is achieved by (i) enforcing the flow to be tangent to the plane and (ii) ensuring that the fluxes normal to the boundary vanish. Similarly to impermeable walls, the boundary state is calculated with twice the normal velocity,

$$\mathbf{u}_h^b = \mathbf{u}_h^+ = \begin{bmatrix} \rho^- \\ \rho u^- - 2n_x \left(\boldsymbol{\rho u}^- \cdot \mathbf{n}\right) \\ \rho v^- - 2n_y \left(\boldsymbol{\rho u}^- \cdot \mathbf{n}\right) \\ \rho w^- - 2n_z \left(\boldsymbol{\rho u}^- \cdot \mathbf{n}\right) \\ \rho E^- \\ \tilde{\mu}^- \end{bmatrix}. \tag{3.77}$$

Then, the numerical fluxes are computed as

$$\boldsymbol{\mathcal{F}}_n^{c,b} = \boldsymbol{\mathcal{H}}(\mathbf{u}_h^-, \mathbf{u}_h^b) \qquad \text{and} \qquad \boldsymbol{\mathcal{F}}_n^{v,b} = 0. \tag{3.78}$$

The latter condition ensures that the flow gradients vanish in the boundary normal direction.

Adaptive wall function. CFD simulations constitute today a standard design tool for many engineering applications. However, the important computational cost of three-dimensional simulations and in particular their application in optimization processes requires the development of fast algorithms.

A huge part of the computational effort needed for wall bounded viscous flows is due to the strong gradients close to the wall requiring a large number of computational cells to resolve the boundary layer. For instance, to perform the simulation of a turbulent flow using a finite volume method, the first grid point off the wall must lie within the viscous sublayer, at about $y^+(1) \approx 1$. Thus, assuming a typical grid stretching of 1.15, around 40 grid cells are generally needed in the wall normal direction [95]. Although the mesh resolution can be reduced by increasing the polynomial order of the shape functions (see chapter 5), the high cost still remains a limiting factor in many large scale industrial CFD applications. Similarly, the automatic generation of highly curved meshes around complex geometries as needed by the DGFEM method is anything other than trivial.

One approach to circumvent the high cost of CFD simulations is based on the almost *universal character of the flow close to the wall* in quasi-equilibrium

boundary layers. In fact, if an appropriate scaling is used, the velocity profiles collapse into a single curve in regions without (i) laminar-to-turbulent transition, (ii) strong (adverse) pressure gradients and (iii) separation or reattachment. This allows under many flow conditions and without a significant loss in accuracy to replace the original problem by decomposing the computational domain into two fully overlapping regions [171]. While the *global flow* problem based on the RANS equations is solved in the outer domain, a simplified one-dimensional model is used in the *near-wall region*. In practice, this can be done by a simple change of the boundary condition. Instead of the no-slip condition (cf. section 3.3), the wall-shear stress and the no-penetration at the wall are prescribed on Γ_w. Since the flow profile in the vicinity of the wall is modeled, significantly coarser meshes can be used.

It is worth noting that the computational efficiency of wall functions is not only due to the smaller number of grid nodes, but also to the larger time steps allowed by the increased height of the elements close to the wall.

While the first wall functions provided accurate results only inside the log-layer, their range of validity has progressively been extended to the entire near-wall region. This new family of wall functions is commonly known as *adaptive* or *low y^+ wall functions*.

In this work, we have implemented an adaptive wall function method similar to the one presented by Knopp [100]. Assuming a quasi-equilibrium flow, derivatives in the streamwise direction are negligible and the flow variables only depend on the coordinate y (measured in the wall normal direction). Then the starting point in the development of any wall function is to write the turbulence variables in non-dimensional form scaling them by the wall shear stress $\tau_w = \mu(\partial u/\partial y)_w = \rho u_\tau^2$. In plus units, the flow variables are defined as

$$y^+ = \frac{y u_\tau}{\nu}, \quad u^+ = \frac{u}{u_\tau}, \quad \mu^+ = \frac{\mu_t}{\mu}, \tag{3.79}$$

where u_τ is the so-called friction velocity.

To improve the accuracy of the model over the entire boundary layer (from the log-layer down to the viscous sublayer), the presented method combines Spalding's (inverse) formula

$$F_{Sp,N}^{-1} \equiv y^+ = u^+ + \exp(-5.2\kappa)\left(\exp(\kappa u^+) - \sum_{n=0}^{N} \frac{(\kappa u^+)^n}{n!}\right) \tag{3.80}$$

with Reichardt's law [141] of the wall

$$F_{Rei}(y^+) \equiv u^+ = \frac{\ln(1+0.4y^+)}{\kappa} + 7.8\left(1 - \exp(-\frac{y^+}{11.0}) - \frac{y^+}{11.0}\exp(-\frac{y^+}{3.0})\right). \tag{3.81}$$

The latter is blended with the classical log-law $F_{log} \equiv u^+ = \ln(y^+)/\kappa + 5.1$,

$$F_{Rei}^{log}(y^+) = (1-\phi_b)F_{Rei} + \phi_b F_{log}, \quad \phi_b = \tanh(\arg^4), \quad \arg = y^+/27, \tag{3.82}$$

where κ denotes the von Karmann constant $\kappa = 0.41$. The final expression of the wall function is then defined by

$$F_{SA} = (1-\phi)F_{Sp,5} + \phi F_{Rei}^{log}, \quad \phi = \tanh(\arg^3), \quad \arg = y^+/24. \tag{3.83}$$

Because the wall distance y is determined by the height of the first cell layer, and u is taken as the slip velocity at the boundary ($y = 0$), the wall function (3.83) defines an implicit expression for the wall shear stress. To solve this non-linear equation we proceed in several steps:

- First, starting from an initial guess for the friction velocity u_τ^0, we use a Newton method to seek $u_{\tau,Rei}$ as solution of the blended law of Reichardt (3.82), i.e. $u/u_\tau = F_{Rei}^{log}(yu_\tau/\nu)$.
- Next, using $u_{\tau,Rei}$ as new "initial" guess, we seek $u_{\tau,Sp}$ as solution of Spalding's inverse formula (3.80), i.e. $yu_\tau/\nu = F_{Sp,5}^{-1}(u/u_\tau)$.
- We then set $\phi = \phi(u_{\tau,Sp})$ and compute the combined friction velocity $u_\tau = (1-\phi)u_{\tau,Sp} + \phi u_{\tau,Rei}$.
- Finally, we compute the wall shear stress as $\tau_w = \rho u_\tau^2$.

A Dirichlet boundary condition for the turbulence working variable $\tilde{\mu}$ of the S-A model can be defined as follows. Given an estimation of the eddy viscosity μ_t computed as

$$\frac{\mu_t}{\mu} = \kappa \exp(-5.5\kappa)\left(\exp(\kappa u^+) - 1 - \kappa u^+ - 0.5(\kappa u^+)^2\right), \tag{3.84}$$

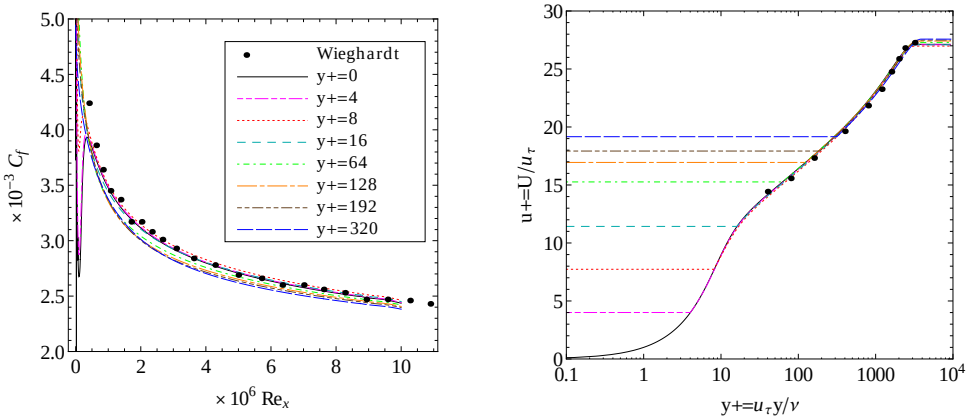
we seek $\tilde{\mu}$ as solution of

$$\mu_t = f_{v1}(\tilde{\mu}/\mu). \tag{3.85}$$

Finally, the numerical fluxes are computed in the following way. As in the case of a symmetry plane the boundary state for the convective term is extrapolated from interior data (cf. equation (3.77)). The inviscid flux is then computed by the Riemann solver while the viscous flux is prescribed by the Neumann boundary condition

$$\boldsymbol{\mathcal{F}}_n^{v,b} = \begin{bmatrix} 0 \\ -\tau_w u_t/\|\mathbf{u}_t\| \\ -\tau_w v_t/\|\mathbf{u}_t\| \\ -\tau_w w_t/\|\mathbf{u}_t\| \\ 0 \\ (\mu + \tilde{\mu}^-)/\sigma \nabla(\tilde{\mu}/\rho)_k^- n_k \end{bmatrix}. \tag{3.86}$$

with $\mathbf{u}_t = [u_t, v_t, w_t]^T$ the wall tangential velocity. The velocity profiles and shear friction distributions obtained with the combined Reichardt-Spalding wall law are presented in Fig. 3.5.



(a) Shear friction

(b) Velocity profiles

Figure 3.5: Shear friction distribution and velocity profiles for the turbulent flat plate using the combined Reichardt-Spalding law of the wall, $p = 4$. y^+ indicates the height of the first grid cell.

3.4 Discretization aspects of the Reynolds-averaged Navier-Stokes equations

As shown in chapter 2, in spite of the popularity of the Spalart-Allmaras model the development of a robust RANS solver remains a challenging task, and many variants to resolve the issues have been proposed. While in section 2.3.4 different attempts to improve the intrinsic stability of the S-A model have been compared, this section focuses on the particular discretization aspects of the Reynolds-averaged Navier-Stokes equations.

3.4.1 The transpose term

Whereas in general a fully coupled RANS solver significantly reduces the number of Newton iterations, we found that - in combination with the IP formulation - the strong coupling between Navier-Stokes equations and SA turbulence model deteriorates the stability of the solver. In contrast to what could be expected, the breakdown of the solver was not caused by Gibbs oscillations of the turbulent viscosity (usually encountered in the under resolved regions of the outer boundary layer). The instabilities are rather related to (unphysical) negative values of the density which systematically appeared in the vicinity of the leading and the trailing edge. A detailed analysis traced the transpose term DT to be at the origin of this phenomenon.

Since similar issues were not encountered for laminar computations, we concluded that they are linked to the diffusive terms that the turbulence model adds to the continuity equation via the transpose term. As all equations are

solved together, it seems that the rapid increase of μ_t during the start-up may result in negative densities at some interpolation points. A similar effect has been observed for the coupling between the continuity and the energy equation. This second instability can be explained by the relationship between the thermal conductivity and the eddy viscosity. To remedy this problem, we propose to *partially decouple the continuity equation by neglecting certain entries of the transpose term.* Indeed, the latter constitutes the only place in the discrete continuity equation where the turbulence variable appears explicitly.

In order to clarify the origin of the coupling, we first rewrite the transpose term as a matrix-vector product of the projected 4th order diffusive tensor D^{kl}_{nm} (cf. Equ. (3.37)) and the interface jump of the solution vector $[\![\mathbf{u}_h]\!]$.

$$
\begin{aligned}
DT_{im} &\equiv \theta \oint_\Gamma \sum_{k=1}^{d}\sum_{l=1}^{d}\sum_{n=1}^{N_v} [\![u_{h_n}]\!]^k \left\langle D^{kl}_{nm} \cdot \frac{\partial \phi_i}{\partial x^l} \right\rangle dS \\
&= \theta \oint_\Gamma \sum_{k,l=1}^{d} \begin{pmatrix} \left\langle n_k\, D^{kl}_{11}\, \frac{\partial \phi_i}{\partial x^l} \right\rangle & \cdots & \cdots & \left\langle n_k\, D^{kl}_{N_v 1}\, \frac{\partial \phi_i}{\partial x^l} \right\rangle \\ \vdots & \ddots & \vdots & \vdots \\ \left\langle n_k\, D^{kl}_{1 N_v}\, \frac{\partial \phi_i}{\partial x^l} \right\rangle & \cdots & \cdots & \left\langle n_k\, D^{kl}_{N_v N_v}\, \frac{\partial \phi_i}{\partial x^l} \right\rangle \end{pmatrix} \begin{bmatrix} u^+_{h_1} - u^-_{h_1} \\ \vdots \\ u^+_{h_{N_v}} - u^-_{h_{N_v}} \end{bmatrix} dS.
\end{aligned}
\tag{3.87}
$$

Taking into account the definition of the diffusive fluxes (2.3) and (3.37), the contribution of the transpose term to the continuity equation is, in 2D, given by

$$
DT_{i\rho} = \theta \frac{\mu + \mu^t}{\rho} \begin{bmatrix} 0 \\ (-\frac{4}{3} u n_x - v n_y)\frac{\partial \phi_i}{\partial x} + (\frac{2}{3} v n_x - u n_y)\frac{\partial \phi_i}{\partial y} \\ (-\frac{4}{3} v n_y - u n_x)\frac{\partial \phi_i}{\partial y} + (\frac{2}{3} u n_y - v n_x)\frac{\partial \phi_i}{\partial x} \\ DT_{i\rho,\rho E} \\ -\frac{\mu \langle \mu + \mu_t \rangle}{\sigma \rho \langle \mu + \mu_t \rangle}(n_x \frac{\partial \phi_i}{\partial x} + n_y \frac{\partial \phi_i}{\partial y}) \end{bmatrix}^T \cdot \begin{bmatrix} \rho^+ - \rho^- \\ \rho u^+ - \rho u^- \\ \rho v^+ - \rho v^- \\ \rho E^+ - \rho E^- \\ \tilde{\mu}^+ - \tilde{\mu}^- \end{bmatrix}
\tag{3.88}
$$

with

$$
\begin{aligned}
DT_{i\rho,\rho E} = &- \frac{\langle \lambda + \lambda_t \rangle}{\langle \mu + \mu_t \rangle} \frac{E - \|\mathbf{u}\|^2}{c_v} \left(n_x \frac{\partial \phi_i}{\partial x} + n_y \frac{\partial \phi_i}{\partial y} \right) \\
&- \left(n_x (\frac{4}{3} u^2 + v^2) + n_y \frac{uv}{3} \right) \frac{\partial \phi_i}{\partial y} - \left(n_y (\frac{4}{3} v^2 + u^2) + n_x \frac{uv}{3} \right) \frac{\partial \phi_i}{\partial x}.
\end{aligned}
\tag{3.89}
$$

Note that (3.88) simply corresponds to the first line of the transpose term evaluated at the average state $\langle \mathbf{u}_h \rangle$. In the case of the velocity and the total energy, a density weighted average is calculated as $\langle \mathbf{u} \rangle \equiv \frac{\rho \mathbf{u}^- + \rho \mathbf{u}^+}{\rho^- + \rho^+}$ and $\langle E \rangle \equiv \frac{\rho E^- + \rho E^+}{\rho^- + \rho^+}$. λ_t represents the turbulent heat conductivity calculated as $\lambda_t = \mu_t c_p / Pr_t$ (cf. section 2.3.2). Pr_t is the turbulent Prandtl number.

Since instabilities mainly appeared in the vicinity of the leading or the trailing edge where adiabatic wall boundary conditions are imposed, we next investigate their influence on (3.88). Denoting by $\Gamma_w \subset \partial\Omega$ the subset of solid wall boundaries, the following Dirichlet conditions are prescribed on Γ_w: $\rho^+ = \rho^-$, $\rho\mathbf{u}^+ = 0$, $\rho E^+ = \rho E^- - 0.5\|\mathbf{u}^-\|^2$ and $\tilde{\mu}^+ = 0$. Substituting these expressions in (3.88) and (3.89), and taking into account that $\mathbf{u}^- \approx \mathbf{u}^+ = 0$ on Γ_w, one immediately remarks that the terms related to the momentum (and continuity) equation are negligible compared to the contribution of the energy equation or the turbulence model. Hence the transpose term simplifies to

$$\begin{aligned} DT_{i\rho} \approx & -\theta \frac{k}{\rho c_v} E \left(n_x \frac{\partial \phi_i}{\partial x} + n_y \frac{\partial \phi_i}{\partial y} \right) \left[\rho E^+ - \rho E^- \right] \\ & - \theta \frac{\mu^2}{\rho^2 \sigma} \left(n_x \frac{\partial \phi_i}{\partial x} + n_y \frac{\partial \phi_i}{\partial y} \right) \left[\tilde{\mu}^+ - \tilde{\mu}^- \right], \qquad \text{on } \Gamma_w. \end{aligned} \tag{3.90}$$

Note that (3.90) is exact if the diffusive tensor is computed using the right state $\mathbf{u}_h^+$. However, we did not observe any influence on the stability in the case the diffusive tensor is evaluated based on the left $\mathbf{u}_h^-$, the right $\mathbf{u}_h^+$ or the average $\langle \mathbf{u}_h \rangle$ state.

In section 3.4.2 we will see that the penalty term plays an important role for the stability of the diffusive operator. In particular, it must be chosen large enough in order to ensure the coercivity of the SIPDG method. However, close to the boundaries the transpose term $DT_{i\rho}$ does not vanish and, since $[\![\rho]\!] = 0$ on Γ_w, it is not counterbalanced by the penalty term. Furthermore, one should bear in mind that coercivity only guarantees the boundedness of the solution but does not forbid negative values of ρ. We therefore propose the following modification which consists in neglecting the components (1,N_v) and (1,N_v-1) of the projected diffusive tensor:

$$DT_{im}^* \equiv \theta \oint_\Gamma \sum_{k,l=1}^{d} \begin{pmatrix} \left\langle n_k \, D_{11}^{kl} \, \frac{\partial \phi_i}{\partial x^l} \right\rangle & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \left\langle n_k \, D_{1N_v}^{kl} \, \frac{\partial \phi_i}{\partial x^l} \right\rangle & \cdots & \cdots & \left\langle n_k \, D_{N_v N_v}^{kl} \, \frac{\partial \phi_i}{\partial x^l} \right\rangle \end{pmatrix} \begin{bmatrix} \rho^+ - \rho^- \\ \vdots \\ \tilde{\mu}^+ - \tilde{\mu}^- \end{bmatrix} dS. \tag{3.91}$$

This hypothesis is equivalent to consider that the diffusive fluxes of the energy equation and the turbulence model do not depend on the gradient of ρ, i.e.

$$D_{N_v-1,1}^{kl} = 0, \quad D_{N_v 1}^{kl} = \frac{\partial \boldsymbol{\mathcal{F}}_k^{v,SA}}{\partial \left(\frac{\partial \rho}{\partial x_l} \right)} = 0, \quad \forall k, \forall l. \tag{3.92}$$

As we were not able to get converged results with the original (full) transpose term, all applications shown in this thesis have been computed with the new proposal (3.91).

Hartmann [82] and Dolejší [49] both applied the interior penalty method to the laminar Navier-Stokes equations. Whereas no particular treatment of

the transpose term has been reported by Hartmann, similar stability issues as in the turbulent case have been encountered by Dolejší. Instead of neglecting some components of the transpose diffusive tensor, he proposes to replace $\frac{\partial \phi_1}{\partial x^l}$ by $\frac{\partial \rho}{\partial x^l}$ in (3.87). For completeness, it should be mentioned that with definition (3.91), the method is strictly speaking no longer symmetric nor anti-symmetric. That means that in contrast to the original IP formulation, the direct and the transpose diffusive term no longer satisfy the following property: $DI_{im} = \theta\ DT_{mi}$.

3.4.2 The penalty term

The numerical properties of the interior penalty method are closely related to the definition of the penalty parameter δ. On the one hand, the coercivity and hence the stability of the SIPDG method is only guaranteed for sufficiently large δ. On the other hand, Castillo [28] proved that the conditioning number of the resulting set of equations grows linearly with δ. Its definition is hence of utmost importance.

Based on the work of Warburton and Hesthaven [180], Shahbazi [158] has derived an explicit expression for δ leading to *optimal* values for the Poisson equation (assuming a constant viscosity parameter ν) on simplex meshes. Epshteyn *et al.* [57] generalized this definition providing scaling rules in case of variable and anisotropic viscosity, whilst recently Hartmann and Houston [84] considered matrix penalties. Instead of the scalar weighting function (3.61), they proposed a matrix variant (transcribed to our notation)

$$\boldsymbol{\delta}(\mathbf{u}_h) = C_{IP}^* \frac{p^2}{h} \langle \boldsymbol{\mathcal{D}}(\mathbf{u}_h) \rangle \tag{3.93}$$

scaled by the average of the diffusive Jacobian tensor (cf. section (3.2.3)) and with C_{IP}^* a constant of around 10-20. Contrary to (3.61), where all components of the systems are scaled by the same value, the matrix penalty (3.93) introduces a tailored jump penalization thereby leading to a superior convergence rate - compared to the standard (scalar) IP method - for some applications [84]. A very similar approach but with the Jacobian tensor evaluated at the average of the numerical solution, i.e. $\boldsymbol{\mathcal{D}}(\langle \mathbf{u}_h \rangle)$ instead of $\langle \boldsymbol{\mathcal{D}}(\mathbf{u}_h) \rangle$, has been considered by Gassner *et al.* [64].

In [52] we have discussed the influence of the scaling length h on the stability of the RANS equations. Very recently we have extended Shahbazi's analysis to provide optimal scalar values which account not only for mesh anisotropy and hybrid meshes [52], but also for highly variable viscosity; see Drosson and Hillewaert [51]. Both issues are encountered in turbulent (boundary layer) computations.

In the present section we briefly summarize the main findings of the stability analysis presented in [51]. For a detailed description, we refer to the cited literature.

Starting point of the coercivity analysis. To simplify the discussion, we will consider a simple Poisson problem with variable viscosity ν

$$\nabla(\nu\nabla u) = \mathcal{S}. \tag{3.94}$$

As the discretization of the source term $\mathcal{S}$ is independent of the IP formulation, it will be neglected here. Furthermore, only the symmetric case ($\theta = 1$) is considered because the NIPDG method ($\theta = -1$) trivially satisfies the coercivity condition [143]. By applying the IP formulation (3.60) to (3.94) and using Young's inequality[13], the bilinear form $a(\cdot,\cdot)$ of the Poisson problem can be written as

$$\begin{aligned} a(u_h, u_h) \geq & \sum_T \int_T \nu \left(\nabla u_h\right)^2 dV - \sum_F \frac{1}{\epsilon_F} \int_F \langle \nu \nabla u_h \rangle^2 \, dS \\ & + \sum_F \left(\delta_F - \epsilon_F\right) \int_F [\![u_h]\!]^2 dS \quad \forall u_h \in \Phi^p, \end{aligned} \tag{3.95}$$

with ϵ_F a positive yet undefined constant (see equation (8) in [51]). Next, to derive a lower bound of the bilinear form the following trace inverse inequality is introduced

$$\int_F u_h^2 dS \leq C_{\mathfrak{T},\mathfrak{F}}(q) \frac{\mathcal{A}(F)}{\mathcal{V}(T)} \int_T u_h^2 dV, \quad \forall u_h \in \Phi^q. \tag{3.96}$$

Here $C_{\mathfrak{T},\mathfrak{F}}(q)$ is a coefficient that depends on the polynomial degree q of the discrete solution and on the canonical type of face $\mathfrak{F}$ and element $\mathfrak{T}$ in the reference space. $\mathcal{A}(F)$ denotes the "area" of face F and $\mathcal{V}(T)$ the "volume" of element T. In a recent work Hillewaert *et al.* [86] propose sharp values of $C_{\mathfrak{T},\mathfrak{F}}(q)$ for any type of element present in unstructured hybrid meshes. Thereby, they generalize previous results of Warburton and Hesthaven [180] for Lagrangian interpolation on simplices.

Based on (3.95) and (3.96) six different variants of the penalty parameter have been investigated, resulting from two choices for the length scale and three for the viscosity scale.

Choice of the length scale. Assuming a constant viscosity ν and using the trace inverse inequality followed by some mathematical manipulations, one finds [51]

$$\begin{aligned} & a(u_h, u_h) \geq \sum_F \left(\delta_F - \epsilon_F\right) \int_F [\![u_h]\!]^2 dS \\ & + \sum_T \int_T \nu \left(\nabla u_h\right)^2 \left(1 - \sum_{\substack{F \in T \\ F \notin \partial\Omega}} \frac{\nu C_{\mathfrak{T},\mathfrak{F}}(p)}{2\epsilon_F} \frac{\mathcal{A}(F)}{\mathcal{V}(T)} - \sum_{\substack{F \in T \\ F \in \partial\Omega}} \frac{\nu C_{\mathfrak{T},\mathfrak{F}}(p)}{\epsilon_F} \frac{\mathcal{A}(F)}{\mathcal{V}(T)} \right) dV. \end{aligned} \tag{3.97}$$

[13] Young's inequality: $\forall \epsilon > 0, \quad \forall a, b \in \mathbb{R}, 2ab \leq \epsilon a^2 + b^2/\epsilon$.

A straightforward generalization of Shahbazi's proposal to hybrid meshes shows that $a(\cdot,\cdot)$ is coercive, by choosing

$$\delta_F > \epsilon_F > \max_{T \ni F} \left(\nu \sum_{g \in T} \frac{C_{\mathfrak{T},\mathfrak{g}}(p)}{2} \frac{\mathcal{A}(g)}{\mathcal{V}(T)} \right), \quad F \notin \partial\Omega \tag{3.98}$$

for internal faces, whilst the penalty parameter should be twice as large for boundary faces. g is a local index running on the faces of the element and $\mathfrak{g}$ denotes the corresponding canonical element. Since (3.98) is roughly proportional to the inverse of the element radius, we refer to this definition as "*minimum radius*". Furthermore, as δ_F is similar for all faces of the element, this choice is useful in case of (nearly) isotropic elements.

In case of the highly anisotropic meshes as encountered in turbulent (boundary layer) computations, the following choice might be more appropriate

$$\delta_F > \epsilon_F > \max_{T \ni F} \left(\nu \frac{C_{\mathfrak{T},\mathfrak{F}}(p)}{2N_T} \frac{\mathcal{A}(F)}{\mathcal{V}(T)} \right), \quad F \notin \partial\Omega. \tag{3.99}$$

where N_T is the number of faces of the element T. The resulting penalty parameter is inversely proportional to the element "height" normal to the face and enhances thus the penalty contribution in the wall normal direction for boundary layer meshes. We refer to this choice as "*distance to the opposing node*". It is easy to check that both definitions respect the coercivity condition.

Choice of the viscosity scale. The basic idea of turbulence modelling is to simulate the effect of the small scales by increasing the effective viscosity. As a consequence, the latter can - in contrast to most laminar computations - no longer be considered constant, and the question arises which value we should use for the viscosity scale. Because this analysis is independent of the choice of the length scale, the anisotropic variant is preferred here as it leads to the most simple expression.

In a recent study [51] we have compared three different definitions of the viscosity scale. These are:

- *Facewise maximum viscosity*

$$\delta_F > \epsilon_F > \max_{T \ni F} \left(\tilde{M}_F \frac{C_{\mathfrak{T},\mathfrak{F}}(p)}{2N_T} \frac{\mathcal{A}(F)}{\mathcal{V}(T)} \right), \quad F \notin \partial\Omega; \tag{3.100}$$

- *Facewise relative viscosity*

$$\delta_F > \epsilon_F > \max_{T \ni F} \left(\frac{M_F^2}{m_T} \frac{C_{\mathfrak{T},\mathfrak{F}}(p)}{2N_T} \frac{\mathcal{A}(F)}{\mathcal{V}(T)} \right), \quad F \notin \partial\Omega; \tag{3.101}$$

- *Elementwise maximum viscosity*

$$\delta_F > \epsilon_F > \max_{T \ni F} \left(M_T \frac{C_{\mathfrak{T},\mathfrak{F}}(p+q)}{2N_T} \frac{\mathcal{A}(F)}{\mathcal{V}(T)} \right), \quad F \notin \partial\Omega. \tag{3.102}$$

Here $M_F \equiv \max_F(\nu)$ and $M_T \equiv \max_T(\nu)$ are the maximum viscosity on face F and inside element T, respectively, whilst $m_T = \min_T(\nu)$ is the minimum viscosity inside element T. The $\tilde{(\cdot)}$ indicates an average related viscosity, i.e. $\tilde{M}_F \equiv \max_F(\langle\nu\rangle) = \max_F((\nu^- + \nu^+)/2)$.

The first definition (3.100) is a naive extension of the constant viscosity case. It is the *standard choice in this thesis* and except mentioned differently all presented numerical applications are performed using this choice. The second proposition (3.101) is similar to the one of Epshteyn and Rivière [57, 143]. However, a sharper bound is obtained by using the recent trace inverse inequalities. It is further referred to as the *maxMin* scale. Finally, the third proposal (3.102) results from the observation that the (RANS) eddy viscosity μ_t is bounded by the S-A variable $\tilde{\mu}$.

From Equ. (2.27) it is easily seen that for any positive value of the intermediate variable $\tilde{\mu}$, an upper bound of the eddy viscosity μ_t is given by

$$|\mu_t| \leq |\tilde{\mu}|, \quad \forall \tilde{\mu} > 0. \tag{3.103}$$

Hence,

$$\int_F (\nu \nabla u_h)^2 dS \leq C_{\mathfrak{T},\mathfrak{F}}(p+q) \frac{\mathcal{A}(F)}{\mathcal{V}(T)} \int_T (\nu \nabla u_h)^2 dV. \tag{3.104}$$

Inequality (3.103) is illustrated in Fig. 3.6 and has an important consequence on the choice of the penalty parameter. Note that for $\tilde{\mu} < 0$, Equ. (3.103) is no longer satisfied since $f_{v1}(\chi) \to \infty$ as $\chi \to -c_{v1}$. However, for negative values of $\tilde{\mu}$ the turbulence model itself becomes unstable and several modifications have been discussed in chapter 2 to avoid $\tilde{\mu} < 0$. In turn, $\tilde{\mu}$ varies with the order p of the interpolation space, since $\tilde{\mu}$ is one of the variables. Hence choosing $q = p$ in equation (3.102) is an adequate and strictly stable choice, although it becomes rather conservative when approaching the boundary; see Fig. 3.6. Although our standard choice, the *facewise maximum scale*, does not guarantee the coercivity of the bilinear form, it leads to satisfying results - in terms of stability and non-linear convergence - for most applications. In contrast, both the *maxMin* and the *elementwise* scale respect the coercivity condition.

3.4.3 Wall distance

As can be seen from equations (2.27) to (2.30) the Spalart-Allmaras model requires the knowledge of the distance to the nearest wall. Since the integration of the different terms in (3.60) is performed numerically, this implies to determine the smallest wall distance for each Gaussian quadrature point.

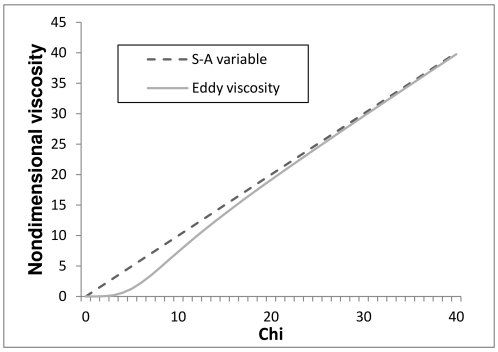


Figure 3.6: Comparison between the (non-dimensional) eddy viscosity $\mu_t/\mu = \chi f_{v1}$ and the Spalart-Allmaras variable $\tilde{\mu}/\mu$ for different values of χ. Note that $|\mu_t| < |\tilde{\mu}|$, $\forall \chi \geq 0$. In particular, $\mu_t \to \tilde{\mu}$ as $\chi \to \infty$.

In practice, the computation of these distances occurs in two stages at the beginning of the simulation.

Let Q be a point whose distance to Γ_w, the union of all (solid) walls, must be calculated. We first identify the nearest element to Q lying on Γ_w. Thereto, we discretize each boundary element by a certain number[14] of points. Next, the nearest point P belonging to the boundary is searched using the *Approximate Nearest Neighbour Searching (ANN)* library [123]. This open source library employs a couple of different tree algorithms allowing to efficiently process very large data sets.

Since the use of high-order polynomial approximations goes most of the time together with the use of coarse meshes, the nearest neighbour search may result in an inaccurate estimation of the wall distance; see Fig. 3.7 (case 2). To improve the accuracy, the exact distance is computed in a second step by projecting the query point Q on the previously identified element. Denoting by (ξ, η) the local coordinates in the reference space associated to the boundary element, the projection algorithm reads: Minimize the objective function

$$d(\xi,\eta) = \left(x_P(\xi,\eta) - x_Q\right)^2 + \left(y_P(\xi,\eta) - y_Q\right)^2 + \left(z_P(\xi,\eta) - z_Q\right)^2, \quad (3.105)$$

subject to the linear constraints

$$-1 \leq \xi, \eta \leq 1 \quad \text{(quadrangle)} \quad (3.106)$$

or

$$0 \leq \xi, \quad 0 \leq \eta, \quad \text{and} \quad \eta \leq 1 - \xi \quad \text{(triangle)}. \quad (3.107)$$

[14] A compromise between a sufficiently accurate representation of the boundary elements on the one hand and the size of the resulting data set on the other hand must be found. Indeed, the search of the nearest neighbour could identify the wrong element, if the number of boundary points is insufficient. Furthermore, the better the estimation of the wall distance, the faster the ensuing projection algorithm. In practice, we represent each element by 10 points per space direction.

Here, the constraints (3.106) and (3.107) express that the projection P must lie inside the element. The optimization problem (3.105) to (3.107) can be solved very efficiently using a quasi-Newton method. In this work, we build an approximate Hessian matrix that neglects the curvature $\partial^2\mathbf{x}/\partial\boldsymbol{\xi}^2$ of the boundary elements. The resulting algorithm leads to the exact distance in only one iteration in the case of straight boundary elements.

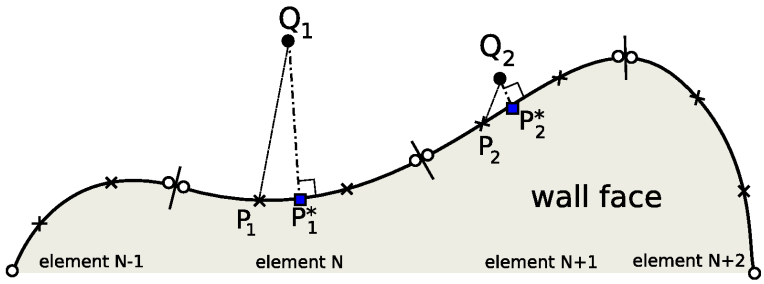
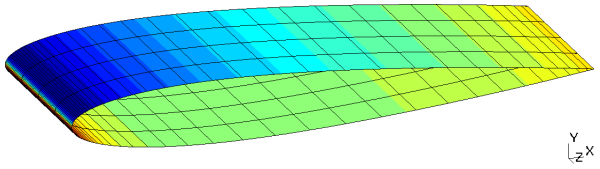


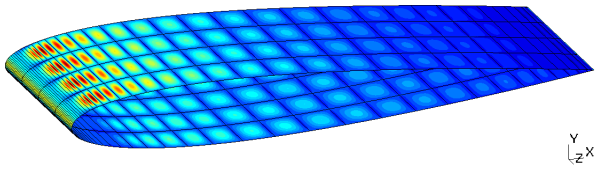
Figure 3.7: Minimum wall distance: '•' query quadrature point Q, '○' element vertex, '×' auxiliary discretization points, '■' orthogonal projection. Especially if the quadrature point Q is located close to the wall, a simple nearest neighbour search leads to a very inaccurate wall distance.

According to our experience the Spalart-Allmaras model turned out to be extremely sensitive to the accuracy of the computed wall distance. Thus, especially for three-dimensional simulations small fluctuations in the distances close to the wall are amplified by the model and result in an inaccurate estimation of the skin friction. Other quantities such as the velocity, pressure... seem however less affected.

The impact of the wall distance on the computed flow fields is illustrated by the NACA 0012 aerofoil; Fig. 3.8. Thereto, we extrude the two-dimensional profile by 4 layers in the spanwise direction, imposing symmetry boundary conditions on z-plans. Since the resulting stationary flow is two-dimensional, no fluctuation should be observed in the spanwise direction. However, Fig. 3.8 shows a situation that we typically observed if the wall projection is solved with an insufficient accuracy. While the pressure coefficient (Fig. 3.8(a)) is perfectly homogeneous in z-direction, the skin friction (Fig. 3.8(b)) oscillates inside elements. Note that in the present case, the norm of the non-linear residual vector has been reduced by a factor 10^{-9}. Finally, this example clearly demonstrates the importance of the boundary representation, since oscillations in the (curved) boundary elements have a similar effect than an inaccurate computation of the wall distance.



(a) Pressure coefficient C_p



(b) Friction coefficient C_f

Figure 3.8: NACA 0012 aerofoil ($Re = 1.86 \times 10^6$, $M_\infty = 0.3$, $\alpha = 4°$). While almost no influence on the pressure distribution is observed, an inaccurate computation of the wall distance strongly affects the skin friction.

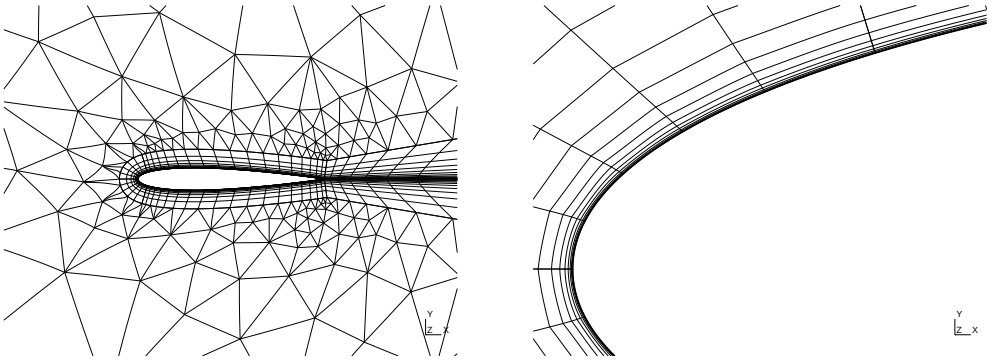
3.5 Numerical results: Robustness of the RANS solver

In the remainder of this section, different numerical examples are presented to investigate the robustness of the interior penalty RANS solver. The focus of the first part (section 3.5.1) lies on the inherent stability of the S-A model. Thereto, two versions of the S-A model are compared in terms of predicted eddy viscosity and algorithm stability. The corresponding results have been submitted for publication in Drosson *et al.* [52]. In the second part (section 3.5.2), the role of the penalty coefficient is analysed in greater detail. In particular, three choices of the viscosity scale and two choices of the length scale are discussed. Finally, we consider a hybrid discretization of the RANS system in section 3.5.3.

3.5.1 Clipped versus modified S-A model

Henceforth, we compare the two versions of the S-A turbulence model, i.e. the "*clipped*" and the "*modified*" one[15], by means of two standard test cases: a flat plate ($Re_x = 5 \times 10^6$ (at $x = 1$), $M_\infty = 0.2$) with zero pressure gradient and a NACA 0012 aerofoil at zero incidence ($Re = 1.85 \times 10^6$, $M_\infty = 0.3$). While

[15]The "*modified*" S-A turbulence model refers to the changes proposed by Oliver resumed in section 2.3.4.



(a) Near aerofoil (b) Leading edge

Figure 3.9: Mixed structured/unstructured NACA 0012 grid, $\alpha = 0°$ (~ 2080 elements). The boundary layer is discretized by several layers of double sided 4th order curved quadrangles.

the present section only focuses on the effect of the changes to the turbulence model, both flows are studied in more details in chapter 5.

The system of RANS equations is solved using a Jacobian-free Newton-Krylov solver with ILU(1) preconditioner. The definition of the penalty length scale h (see section 3.2.3) is based on either the "distance to the opposing node" or the quotient $\mathcal{V}(T)/\mathcal{A}(\partial T)$ referred to as "minimum radius", where $\mathcal{V}(T)$ is the area / volume of element T and $\mathcal{A}(\partial T)$ is the circumference of element T. As for the viscosity scale, all computations have been performed using the *facewise* definition which constitutes the default choice in this thesis. The influence of the viscosity scale is subject of the next section.

All results shown here are computed with $p = 4$ polynomials for the flow over the flat plate and $p = 2$ polynomials for the NACA 0012 aerofoil. Furthermore, the simulations are accelerated by an order sequencing: we first start wit P^1-elements. Once the residual has converged by a factor 10^{-6}, the solution is projected onto P^2 and eventually P^3 and P^4 function spaces.

The computation of the flow around the NACA 0012 aerofoil is performed on a hybrid C-type mesh extending to 30 chord lengths. The mesh is composed of 2080 elements with 2×23 elements on the chord. Grid points are clustered near the ends of the aerofoil. The boundary layer is discretized by several layers of double sided 4th order curved quandrangles; see Fig. 3.9.

For the flat plate, the analysis is made on a relatively fine unstructured grid (744 triangles) as well as on two coarse structured grids composed of 204 / 408 quadrangles / triangles, respectively. The height of the first cell off the wall is set to $y^+(1) = 16$. The larger cell size is possible due to the high-order interpolation compared to classical FV methods where $y^+(1) \approx 1$ is required.

Case	Grid	Penalty	CPU time (sec)		Gain (%)
			Clipped	Modified	
Flat plate	Quads	OppNod	36	40	-11.8
	Quads	MinRad	35	37	-7.5
	Triangles	OppNod	34	39	-13.8
	Triangles	MinRad	n.c.	n.c.	-
	Unstructured	OppNod	307	347	-13.0
NACA0012	Hybrid	OppNod	244	252	-3.4
	Hybrid	MinRad	180	196	-9.0

Table 3.2: Summary of the setups used for the comparison between "clipped" and "modified" S-A turbulence model. The results are computed with $p = 4$ polynomials for the flow over the flat plate and $p = 2$ polynomials for the NACA 0012 foil. The penalty parameter is based on a *facewise* definition of the viscosity scale. ("n.c." indicates that the computation did not converge.)

The setups used for the comparison between the clipped and the modified S-A turbulence model are summarized in table 3.2 which also lists the computational cost (in seconds) necessary to decrease the non-linear residual norm by 10 orders of magnitude. The corresponding timings were obtained on a single Intel core i7-960 processor (3.20 GHz).

Effect on the computed solution

Before analysing the stability of the resulting RANS-SA systems, we first concentrate on the effect on the Spalart-Allmaras working variable $\tilde{\mu}$ on the one hand, and on the remaining flow variables (ρ, $\mathbf{u}$,...) on the other hand.

Fig. 3.10 compares the regions of negative turbulent viscosity in the boundary layer of the flat plate using either the modified or the clipped S-A model. For representation purposes, the results are scaled by a factor 40 in the vertical direction. We first observe that negative values of $\tilde{\mu}$ are found only in the outer boundary layer, where the grid resolution is insufficient to resolve the sharp gradients of the viscosity profile. Although both models predict negative values of the turbulence working variable, these undershoots are reduced by the modifications (2.41) to (2.46) described in the previous section, whereby the amount depends on the grid resolution. For fine meshes, nearly no difference is observed between the two models. This is not surprising since the Gibbs oscillations tend to disappear as the mesh is refined. Moreover, the profiles of the non-dimensional turbulent viscosity $\chi = \tilde{\mu}/\mu$ (see Fig. 3.11) show that the changes to the turbulence model have only a local impact. Except at the

interface between boundary layer and free stream both profiles are identical.

Most important, at least from an engineering point of view, is to know whether the two models lead to similar mean flow solutions. According to the flat plate computations, the modifications do not influence the solution of the Navier-Stokes equations, neither the direct variables (density, momentum,...), see Fig. 3.12(a), nor the derived fields (skin friction, pressure coefficient,...); see Fig. 3.12(b). These observations are confirmed by the flow around the NACA aerofoil and can be explained by the following two arguments: (i) the very local impact of the modifications, and (ii) a certain insensitivity of the flow solution to the accuracy of the turbulence model. The latter will be examined further in chapter 5.

Effect on the stability

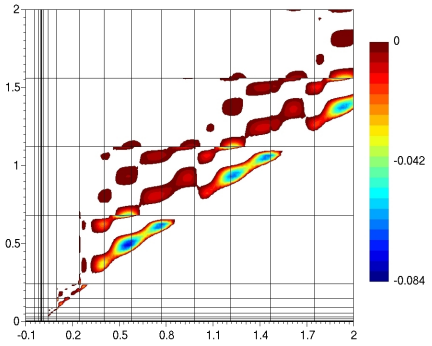
Regarding the results in table 3.2, we notice that despite the changes intended to improve the behaviour of the S-A model,

- clipping is approximately 10% faster than the "modified" model;
- both models diverged for the flat plate computation on the triangular grid with a penalty coefficient based on the "minimum radius".

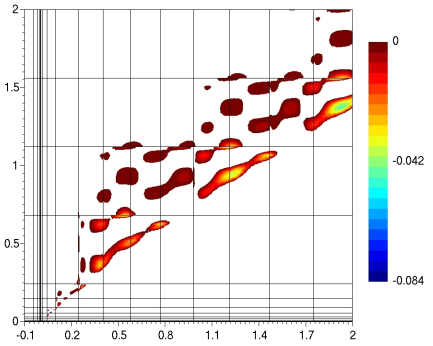
Even though these observations may surprise at first glance, at least a partial explanation can be given looking to the convergence histories of the flat plate and the NACA aerofoil shown in Fig. 3.13. Remember that one of the aims of the modifications is to conserve the C^1-continuity of the model at $\tilde{\mu} = 0$ in order to not hamper the convergence to steady state. However, the new definition of the diffusion and the source terms results in a huge increase of the residual in elements with negative viscosity. Since the CFL number is related to the residual norm by an exponential law, a higher residual leads to smaller time steps and as a consequence increases the computing time. Conversely, even in the case of ill-conditioned unstructured meshes, we did not observe any disturbing impact of the clipping procedure, neither on the non-linear convergence nor on the iterative linear solver.

Even worse, since the whole set of RANS-SA equations is solved as a single coupled system, the sudden increase of the turbulence model residual strongly perturbs the remaining flow variables. It follows according to our numerical experiments that the modifications proposed by Oliver slightly reduce the stability of the method, although they should ensure a decrease in time of the negative turbulent energy.

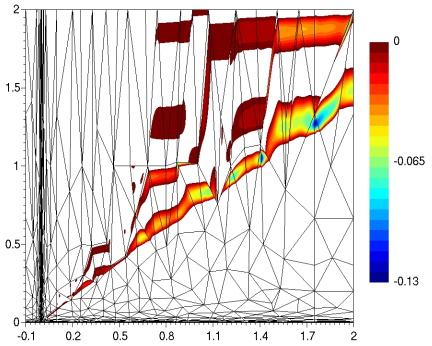
Given the non-linear character of the RANS equations, it is hard to explain the exact mechanisms leading to the breakdown of the computation in some cases. Though, a second factor that may influence the stability of the algorithm is the precision with which the (linear) Newton system is solved. The standard convergence criterion for the iterative linear solver used in this



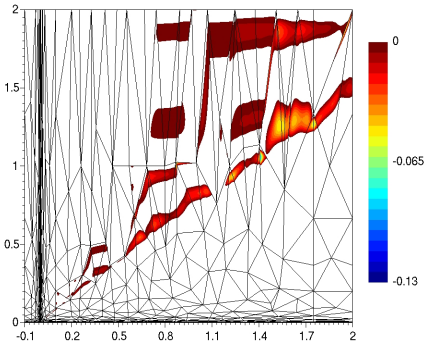
(a) Clipped S-A model



(b) Modified S-A model

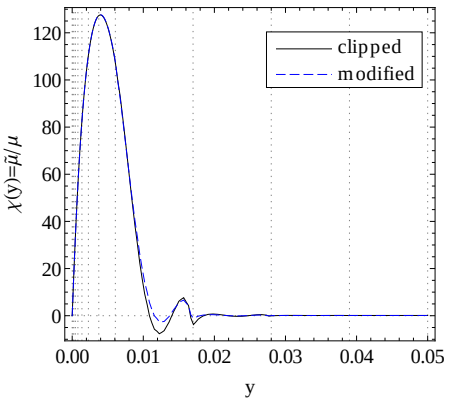


(c) Clipped S-A model

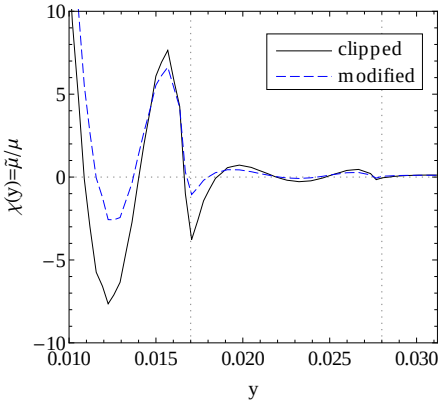


(d) Modified S-A model

Figure 3.10: Regions of negative turbulent viscosity $\bar{\mu} = \tilde{\mu}/(100\mu)$ for a flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) using P^4-elements: *top*, structured quadrangular grid ($(4 + 13) \times 12$ elements); *bottom*, unstructured triangular grid (744 elements). For representation purposes, the results are scaled by a factor 40 in the y-direction. As expected, the "modified S-A model" reduces the regions where $\bar{\mu}$ has significant negative values.

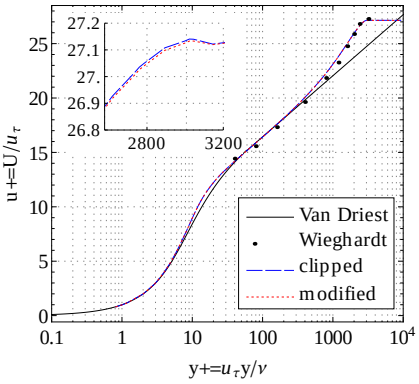


(a) Turbulent viscosity profil $\chi(y)$

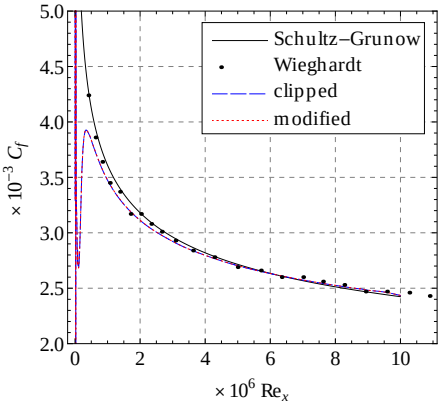


(b) Zoom on the outer boundary layer

Figure 3.11: Non-dimensional viscosity profiles at station $x = 0.55$ for the flow over the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) computed with p=4 polynomials on a coarse structured grid ($(4 + 13) \times 12$ quadrangles). Both versions of the S-A turbulence model lead to identical viscosity profiles except in the outer boundary layer.

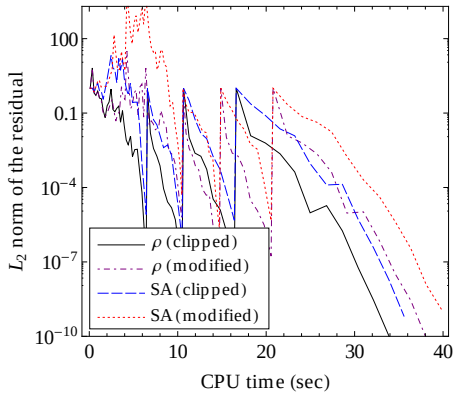


(a) Non-dimensional velocity profile at section $x = 0.97$

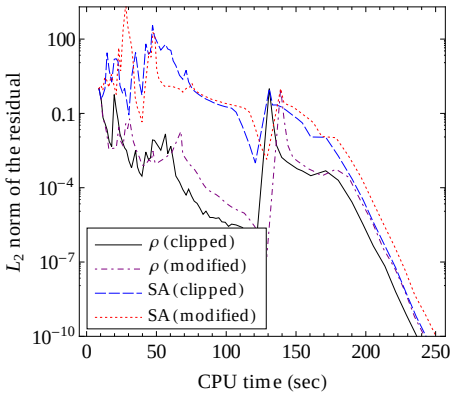


(b) Skin friction

Figure 3.12: Comparison between the clipped and the modified S-A model for the flow over the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) computed with $p = 4$ polynomials on a coarse structured quadrangular grid, $h \equiv$ distance to the opposing node. Both versions of the turbulence model lead to nearly identical solutions of the Navier-Stokes equations.



(a) Flat plate ($Re = 10^7$, $M_\infty = 0.2$): p=4 computed on a coarse quadrangular grid



(b) NACA 0012 ($Re = 1.85 \times 10^6$, $M_\infty = 0.3$, $\alpha = 0°$): p=2 computed on a coarse hybrid grid

Figure 3.13: Comparison between the clipped and the modified S-A turbulence model: convergence histories, distance to the opposing node. The huge increase of the residual observed for the modified S-A model leads to smaller time steps and hence can increase the total computing time.

work requires a reduction by one order of magnitude of the residual norm[16]. While most of the time this tolerance leads to satisfying results, it may be insufficient if the turbulence working variable drops below zero. Indeed, because the residual is much higher for elements with a negative viscosity, the reduction in the largest component of the residual can be sufficient to achieve the linear convergence tolerance whereas the small components of the residual can even have increased [35].

To illustrate the impact of the linear convergence tolerance $\|\mathbf{R}_0\|/\|\mathbf{R}\|$ (with $\mathbf{R}$ the residual vector[17] of the linearized problem), we compute the flow over the flat plate using the modified S-A model and imposing a higher CFL number. The latter is necessary because for small CFL values, all computations remain stable. As illustrated by Fig. 3.14, the instabilities generally appear during the first Newton iterations, which are characterized by a fast growing of the turbulence working variable. We furthermore see that solving the Newton system more accurately can, at least in the present case, avoid the breakdown of the computation. In practice however, it may be very difficult to meet such severe linear convergence requirements for large, ill-conditioned problems. A much better and even more stable approach in order to avoid "oversolving" would be to simply reduce the CFL number. Even worth in terms of memory and CPU resources is the use of direct solvers such as the Gauss method. Although the total number of Newton iterations has been re-

[16] For details about the iterative solution of the time dependent PDEs we refer to chapter 4.
[17] The subscript "0" refers to the first iteration of the linear solver.

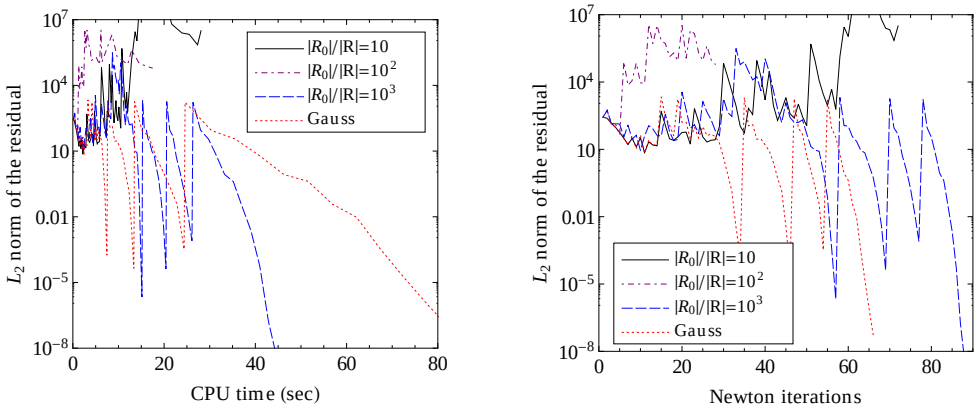


Figure 3.14: Impact of the linear convergence tolerance $\|\mathbf{R}_0\|/\|\mathbf{R}\|$ on the stability of the modified S-A model for the flow over the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) computed with $p = 4$ polynomials on a coarse structured grid, $h \equiv$ distance to the opposing node.

duced by $\sim 25\%$, the computing time has more than doubled; see Fig. 3.14. Finally, since the instabilities are strongly related to the grid resolution, it sometimes proves advantageous to start the computation on P^2- (rather than P^1-) elements. Although this slows down the calculation on fine meshes, it may avoid the breakdown on very coarse or highly curved meshes.

We end the comparison between the clipped and the modified S-A model with a few words about the influence of the penalty coefficient. The latter will be further discussed in section 3.5.2. Regarding the results in table 3.2, we notice that with a scaling length h based on the "minimum radius" computations are (i) generally faster but (ii) sometimes less stable than by choosing the "distance to the opposing node". Instabilities appeared especially in combination with the modified S-A model. The second observation may surprise because - as seen in section 3.4.2 - both definitions lead to a coercive discretization of the RANS-SA system. The stabilizing effect of choosing the distance to the opposing node can probably be explained as follows. Given the strong anisotropy of the meshes currently used in the boundary layer, this choice leads to a more stringent jump penalization in the wall normal direction. However, it is in this direction where the sharpest flow gradients are observed. Hence, depending on the aspect ratio of the grid, the penalty coefficient on faces parallel to the wall can be several orders of magnitude higher than with the "minimum radius". In contrast to this, as the flow changes slowly in the streamline direction, smaller penalty coefficients on faces perpendicular to the streamlines should only have a minor impact on the algorithm stability.

3.5.2 Influence of the penalty coefficient

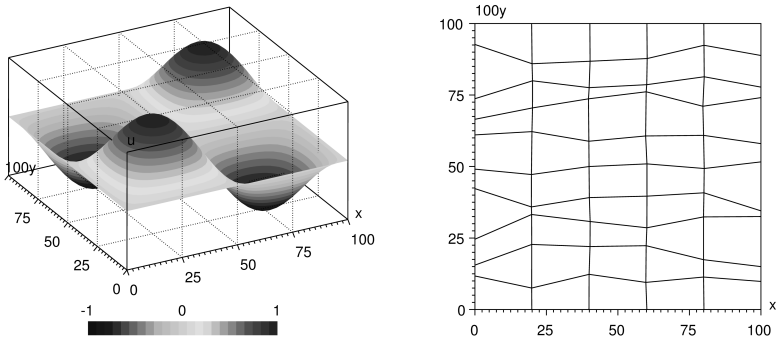
In the following we shortly summarize the main numerical results presented in [51]. The first example is a simple Poisson equation with a prescribed (variable) viscosity. Next the method is applied to the system of RANS equations. The considered test cases are a turbulent flat pate with zero pressure gradient and a NACA 0012 aerofoil at an incidence angle of $\alpha = 3.59°$. Both applications are studied in detail in chapter 5. As the influence of the length scale h has already been discussed in section 3.5.1, we will focus here on the choice of the viscosity scale. The aim is to determine the best strategy for the penalty parameter δ to ensure the stability during the iterative solving procedure. The transient is generally characterized by large interface jumps, especially when the calculation is accelerated by an order sequencing procedure. For the *elementwise* definition of the viscosity, we will assume that the μ_t is satisfactorily approximated in Φ^p.

Poisson equation with variable viscosity

The first application consists of a Poisson equation to be solved on the rectangular domain $[0, 100] \times [0, 1]$. The method of *manufactured solutions*, i.e. using an appropriate value of the source term, is used to provide the reference solution

$$u(x, y) = \sin(2\pi x/100) \sin(2\pi y). \tag{3.108}$$

The computational domain is discretized by 50 quadrilaterals with a mean aspect ratio of 200. The exact solution and the mesh are shown in Fig. 3.15.

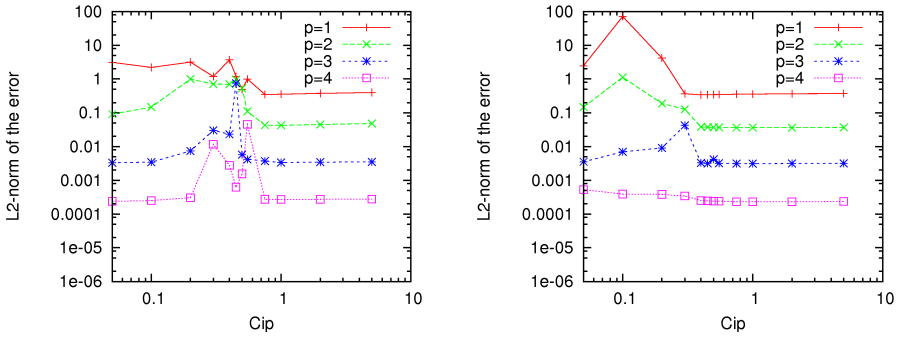


(a) Exact solution (b) Quadrilateral mesh

Figure 3.15: The computational domain $[0, 100] \times [0, 1]$ used for the Poisson problem is partitioned into 50 quadrilaterals. For visualization purposes the mesh is scaled by a factor 100 in the y-direction. The mean aspect ratio of the grid cells equals 200.

Constant viscosity. To emphasize the effect of the length scale we first consider a Poisson problem with unit viscosity. The stability of the method is analyzed by computing for each polynomial orders $p = 1 \dots 4$, the L_2-norm of the nodal error versus the penalty parameter. Since the three definitions of the viscosity scale are equivalent in case of a constant viscosity, only the facewise scale is considered here. A systematic variation of δ indicates that the six critical values δ^* proposed in section 3.4.2 yield a stable solution; see Fig. 3.16. (Whereas for $\delta > \delta^*$, the nodal error remains almost constant, a small variation of the penalty parameter causes a large variation of u if $\delta < \delta^*$.)

Despite of the high aspect ratio of the grid cells, almost no difference is found between the isotropic and the (new) anisotropic choice of the length scale. In particular both formulations lead to a penalty parameter that is approximately two times larger than the critical stable value observed in practice.



(a) Isotropic (b) Anisotropic

Figure 3.16: Influence of the penalty parameter on the L_2-norm of the nodal error for different interpolation orders $1 \leq p \leq 4$ in case of a Poisson equation with *constant* viscosity parameter. The computational domain is partitioned into 50 quadrilaterals with a mean aspect ratio of 200.

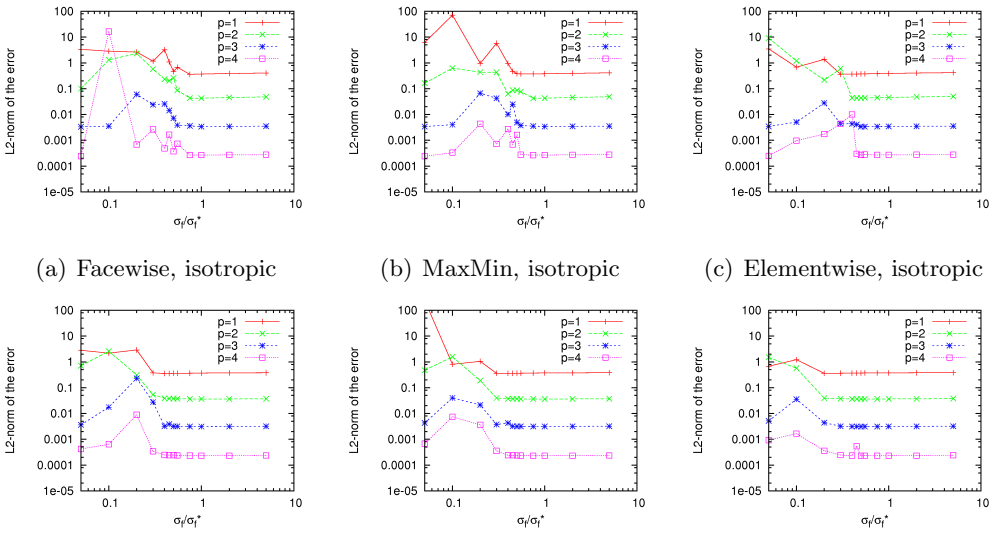
Variable viscosity. Whereas the previous paragraph was dedicated to a Poisson problem with constant viscosity, we now investigate the effect of strong variations of the diffusivity parameter. The latter is characteristic of turbulent boundary layers where the eddy viscosity rapidly reaches values that are several orders of magnitude higher than the molecular viscosity. In the remainder of this section, the following polynomial expression of the viscosity will be considered

$$\nu(x, y) = 1 + 1000 \left[\left(\frac{x}{100} \right)^q + y^q \right].$$

with $q = 1$, 2 or 4. Since q is known, the elementwise formulation can be based on the exact polynomial order of the viscosity. Furthermore the source term

$\mathcal{S}(x, y)$ is adapted in order to conserve the same analytic solution as for the Poisson problem with unit viscosity; cf. Equ. (3.108).

Calculating the L^2-norm of the nodal error against the penalty parameter a similar situation as for the Poisson problem with constant viscosity is observed; see Fig. 3.17, 3.18 and 3.19. In particular, all formulations yield a stable discretization if $\delta > \delta^*/2$, and this almost independently of the order q of the viscosity. It is worth noting that although the (naive) facewise scale does not formally guarantee coercivity, it defines a very sharp bound of the penalty parameter, that furthermore is close to the maxMin formulation. One reason is probably the smoothness of the imposed viscosity that is continuous through the entire domain. Moreover, the viscosity does not include any Gibbs oscillations (with possibly negative values) as currently encountered for RANS computations in the underresolved regions of the outer boundary layer. For the same reason it might be not surprising that the elementwise scale results in the most conservative estimation of δ. This is particularly true if a high order viscosity ($q = 4$) is combined with low p.

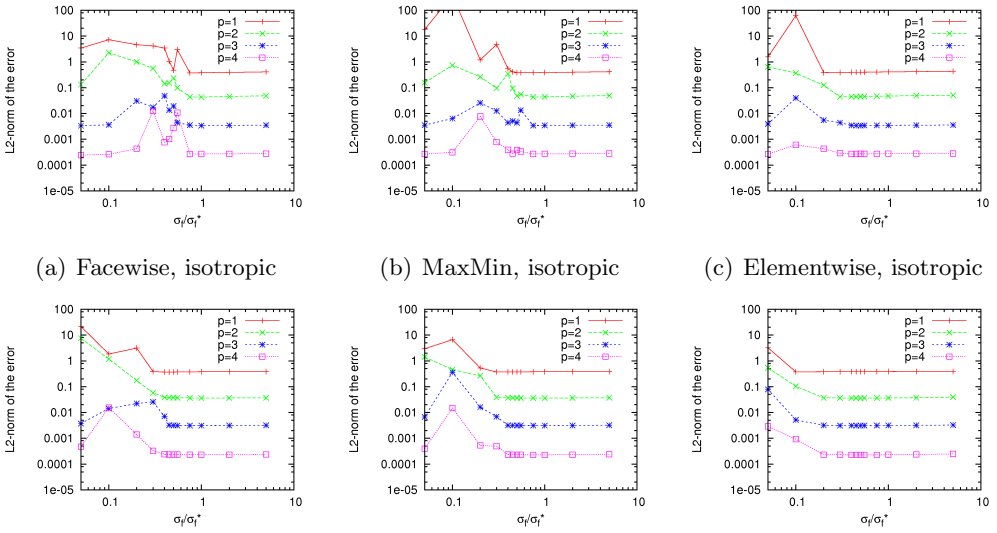


(a) Facewise, isotropic (b) MaxMin, isotropic (c) Elementwise, isotropic

(d) Facewise, anisotropic (e) MaxMin, anisotropic (f) Elementwise, anisotropic

Figure 3.17: Influence of the penalty parameter on the L_2-norm of the nodal error for different interpolation orders $1 \leq p \leq 4$ for a Poisson equation with *linear* viscosity parameter, $\nu(x, y) = 1+1000((x/100)+y)$. The computational domain is partitioned into 50 quadrilaterals with a mean aspect ratio of 200.

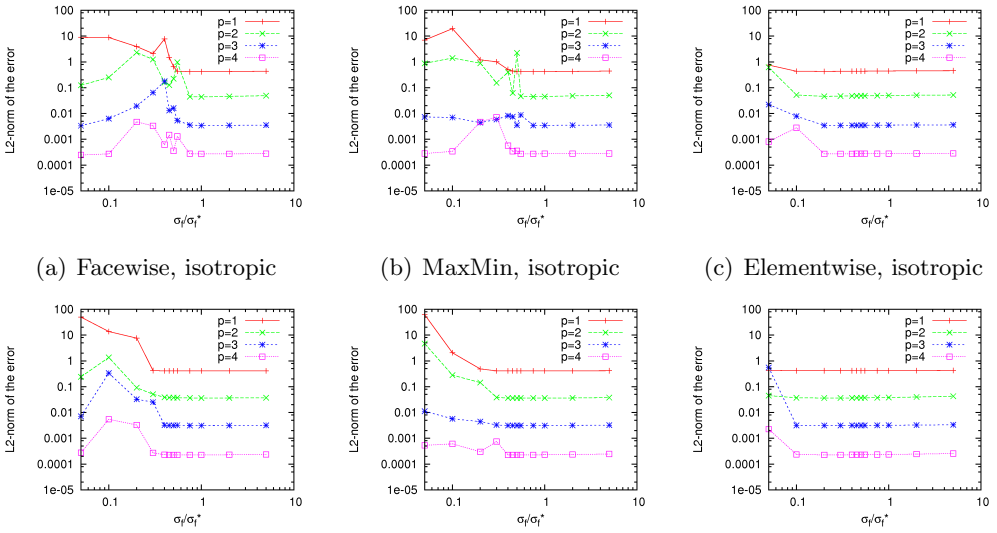
Although the previous results suggest that the anisotropic length scale does not result in a sharper definition of the penalty parameter, it apparently produces less stiff discretizations. In fact, we found that compared to the isotropic length scale, it reduces the number of GMRES iterations required to



(a) Facewise, isotropic (b) MaxMin, isotropic (c) Elementwise, isotropic

(d) Facewise, anisotropic (e) MaxMin, anisotropic (f) Elementwise, anisotropic

Figure 3.18: Influence of the penalty parameter on the L_2-norm of the nodal error for different interpolation orders $1 \leq p \leq 4$ for a Poisson equation with *quadratic* viscosity parameter, $\nu(x, y) = 1 + 1000((x/100)^2 + y^2)$. The computational domain is partitioned into 50 quadrilaterals with a mean aspect ratio of 200.

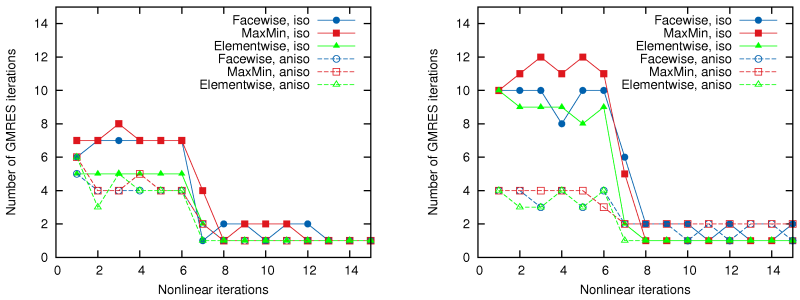


(a) Facewise, isotropic (b) MaxMin, isotropic (c) Elementwise, isotropic

(d) Facewise, anisotropic (e) MaxMin, anisotropic (f) Elementwise, anisotropic

Figure 3.19: Influence of the penalty parameter on the L_2-norm of the nodal error for different interpolation orders $1 \leq p \leq 4$ for a Poisson equation with *4th order* viscosity parameter, $\nu(x, y) = 1 + 1000((x/100)^4 + y^4)$. The computational domain is partitioned into 50 quadrilaterals with a mean aspect ratio of 200.

solve the linearized system; see Fig. 3.20. Thereby, the higher the aspect ratio of the grid cells, the higher the influence on the stiffness.



(a) Mean aspect ratio of 200 (b) Mean aspect ratio of 320

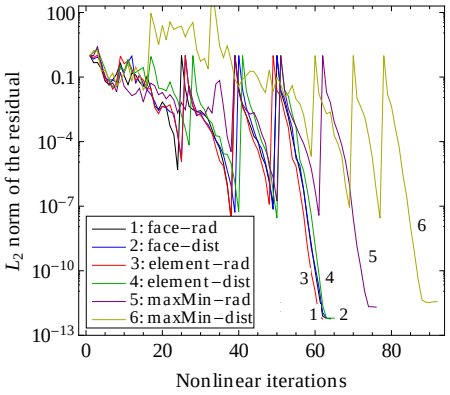
Figure 3.20: Influence of the length scale on the stiffness of the discretization for a Poisson problem with 4th order viscosity.

RANS system

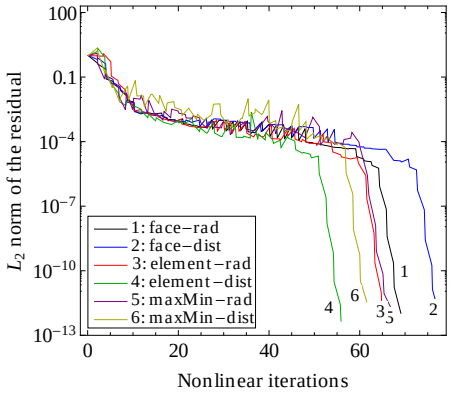
As shown in Fig. 3.21(a), no significant difference was found between our standard *facewise* and the new *elementwise* definition of δ for the turbulent flat plate computations on very resolved meshes. Similarly, the choice of the length scale has almost no influence on the convergence. However, the situation changes if the mesh resolution is reduced. In this case, the new *elementwise* definition clearly outperforms the other ones. Whilst in the absence of a sequencing procedure, all simulations remain stable, the standard and the *maxMin* definition often diverge during the start-up with $p = 1$ polynomials. Furthermore, as already observed in chapter 2 the length scale definition for anisotropic meshes also seems to improve the stability of the non-linear solver. Note that all computations presented in Fig. 3.21 are resolved with regard to the skin friction and the velocity profile; see section 5.1.

The differences between the three definitions of δ are even more drastic if we consider the flow around the NACA 0012 aerofoil. Again we have chosen a large grid normal spacing ($y^+(1) \approx 50$) which - although coarse - was found delivering acceptable results by a mesh convergence study in section 5.3.

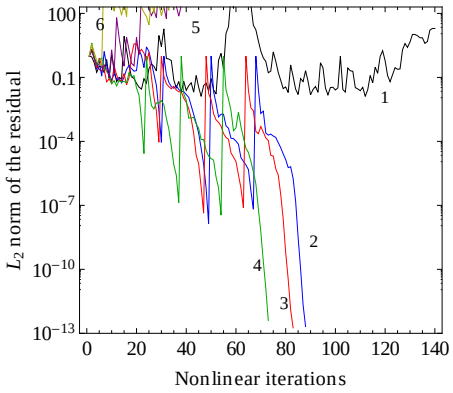
In contrast to the turbulent flat plate, where by a systematic variation of δ its value was shown to be nearly optimal, the *facewise* and the *maxMin* definition of δ lead to a breakdown of the computation, if large off-wall spacings are used. The only way to avoid the breakdown is then to increase the penalty parameter by a factor of at least 2, if $p = 4$ polynomials are used, and by a factor even higher (15 to 20), if the computation is initialized with $p = 2$ polynomials. It is worth noting that even with a significant reduction of the CFL number, converged results could not be obtained. The only alternative



(a) $y^+(1) = 24$, 4 levels



(b) $y^+(1) = 64$, 1 level

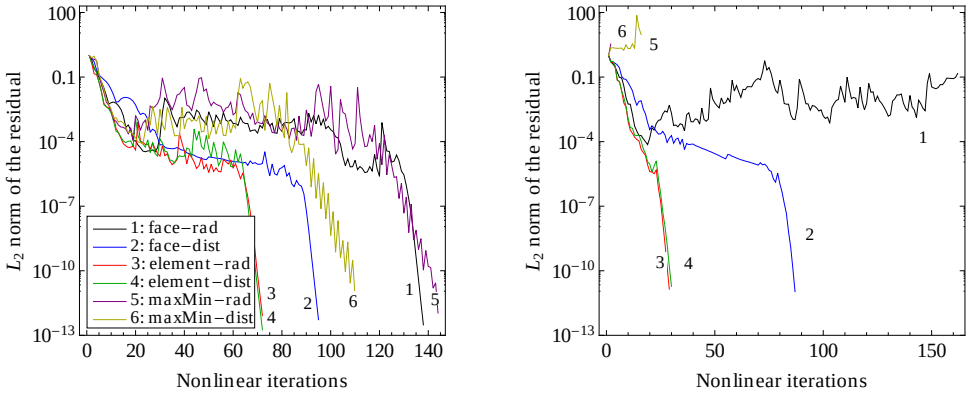


(c) $y^+(1) = 64$, 4 levels

Figure 3.21: Non-linear convergence histories for the flow over the turbulent flat plate ($Re_x = 5 \times 10^6$ at $x = 1\,\mathrm{m}$, $M = 0.2$) on a quadrilateral mesh using $p = 4$. "rad" and "dist" refer to the element radius (isotropic meshes) and the distance to the opposing node (anisotropic meshes) respectively.

to the increase of δ is the use of finer grids presenting a smaller element height, e.g. $y^+ \approx 20$. In contrast, the *elementwise* definition is always stable.

Convergence histories of the NACA aerofoil for different initial values of the eddy viscosity are displayed in Fig. 3.22. Especially for a high initial value of the turbulent viscosity μ_t, the new definition of δ spectacularly reduces the number of non-linear iterations. In contrast, the *maxMin* scale - although formally coercive - is clearly not very stable. As it is proportional to the ratio of the square of the maximum to the minimum value, the *maxMin* scale is highly sensitive to spatial and transient variations of μ_t. For RANS computations where the eddy viscosity is currently 1000 higher than the molecular viscosity this dependency is troublesome.



(a) Initial eddy viscosity: $\mu_t/\mu = 500$ (b) Initial eddy viscosity: $\mu_t/\mu = 5000$

Figure 3.22: Convergence histories for the flow around the NACA 0012 aerofoil ($Re_c = 1.86 \times 10^6$, $M = 0.3$) on a hybrid mesh using 3^{rd} order curved quadrilaterals in the boundary layer ($y^+(1) \approx 50$), $p = 4$.

3.5.3 Low-order interpolation of the Spalart-Allmaras variable

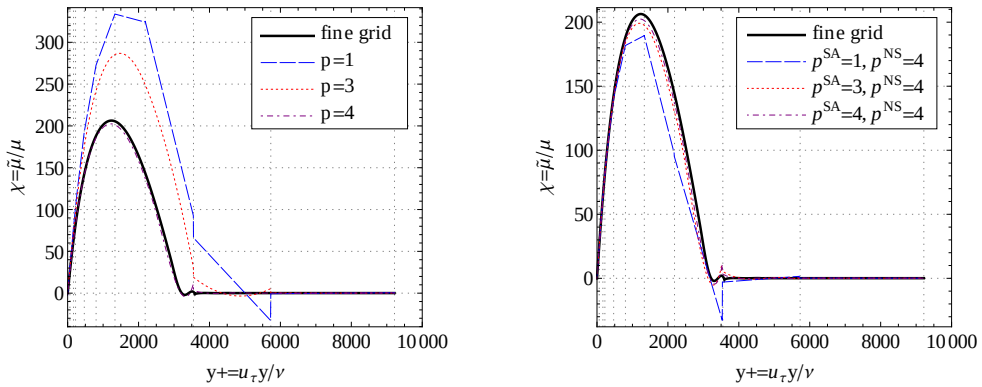
In the remainder of this section we aim to answer the following questions:

- Can a hybrid discretization improve the robustness of the DG-IP solver?
- How does a low-order interpolation of the SA variable influence the overall accuracy of the method?

Although the previous examples have been computed using the same polynomials of degree p for all state variables, this is not required by the IP formulation. The hybrid discretization is obtained by reducing the polynomial order of the turbulence model discretization p^{SA} while keeping unchanged the high-order interpolation p^{NS} of the continuity, momentum and energy equations. A similar approach is commonly utilized within the context of finite volume methods, where a constant reconstruction of the turbulence model is combined with a linear or quadratic discretization of the remaining equations. However, in contrast to standard finite volume schemes, the presented high-order DG method allows the use of significantly coarser meshes, calling into question the accuracy obtained with a reduced order turbulence variable.

To analyse the effect of the hybrid discretization, we again compute the flow around the flat plate. The employed coarse grid has a wall normal spacing of the first layer equal to $y^+(1) = 64$. The main flow equations are interpolated using P^4-elements while the interpolation order p^{SA} of the SA model is progressively reduced. As will be seen in section 5.1.1, the corresponding 5th order accurate DGM discretization is sufficiently resolved on this grid.

The resulting turbulent viscosity profiles for the standard ("homogeneous") and the hybrid discretization are compared in Fig. 3.23. In contrast to what



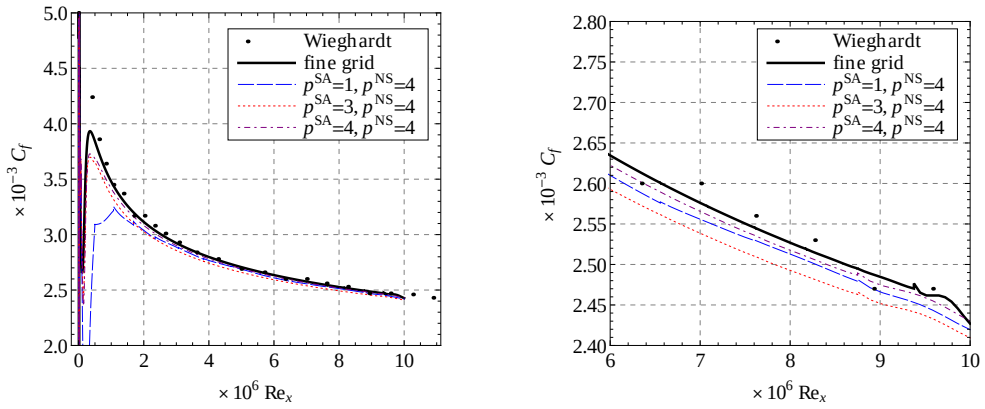
(a) "Global" interpolation order

(b) SA-variable interpolation order

Figure 3.23: Comparison of the non-dimensional viscosity profiles $\chi = \mu_t/\mu$ ($y^+(1) = 64$). The dotted vertical lines represent the mesh. To facilitate the comparison, all results have been non-dimensionalized by the same friction velocity, i.e. $u_\tau(p = 4)$. *Left*: All equations are discretized by the same function space. As the viscosity peak is mainly represented by three elements only, important differences are observed between 1^{st}, 2^{nd} and 4^{th} order polynomials. *Right*: Given a sufficiently high interpolation of the Navier-Stokes equations (here, $p^{NS} = 4$), the interpolation order p^{SA} of the turbulence variable has only little effect.

could be expected intuitively, the hybrid discretization does almost not affect the maximum value of the SA variable. The latter is nearly unchanged and principally depends on the interpolation order p^{NS} of the Navier-Stokes equations. The same observation holds for the spread of the computed boundary layer. It is therefore not surprising that except close to the leading edge, all formulations lead to nearly identical skin friction distributions; see Fig. 3.24. These results are confirmed by P^3-elements on a finer grid ($y^+(1) = 32$), not presented here.

Nevertheless, some remarks are in order. It is worth noting that we were not able to get converged results on grid E combining $p^{SA} = 2$ with $p^{NS} = 4$. This can probably be explained by an important undershoot at the outer boundary layer as observed for a linear interpolation of the turbulence variable. Remember that without special treatment, the SA model becomes unstable in the limit as $\chi = \mu_t/\mu \to -c_{v2} = -5$. Thus, we may conclude from the above considerations that the use of higher-order interpolations of the Spalart-Allmaras variable seems not indispensable to guarantee an overall high-order accuracy but is strongly recommended for stability reasons. In a future work, these results should be confirmed for more complex flow situations. Furthermore, these results may be very specific to the Spalart-Allmaras model.



(a) Whole plate (b) Zoom

Figure 3.24: Effect of low-order interpolations of the Spalart-Allmaras variable on the computed skin friction C_f for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) using P^4-elements to discretize the Navier-Stokes equations. Despite the important off-wall spacing ($y^+(1) = 64$), the interpolation order of the turbulence model has only little impact on the overall accuracy.

3.6 Concluding remarks

In this chapter the basic features of the interior penalty discontinuous Galerkin approach have been introduced. Although this method presents many attractive properties, it was found to be unstable if the system of RANS equations is solved in a fully coupled way. A coupled resolution is however desired in order to reach a Newton type convergence at the end of the iterative solving procedure. Indeed, Burgess and Mavriplis [25] have shown in a recent study that a decoupled model results in a significant deterioration of the non-linear convergence and even can prevent the solver to reach machine accuracy.

To overcome the stability issues different approaches have been considered. The first is related to the definition of the S-A model which becomes unstable for negative values of the turbulent viscosity. It turns out that while changes of the diffusive and the source term allow to reduce the regions of negative $\tilde{\mu}$, they hardly affect the computed solution. However, despite the loss of C^1-continuity the clipped model was found to be approximately 10% faster and even slightly more stable than the modified model.

The second type of stability improvements is related to the interior penalty formulation. First, we propose a modification of the transpose term, that partially decouples the continuity equation from the S-A turbulence model. According to our numerical experiments this modification significantly improves the stability of the RANS solver, whilst still allowing a quadratic convergence rate.

Another factor of paramount importance for the performance and the sta-

bility of the IP method is the penalty parameter δ. Using recent sharp trace inverse inequalities we have compared six different definitions of δ - resulting from two choices of the length scale and three choices of the viscosity scale. Whilst the length scale was found to have a minor influence, the anisotropic definition always outperformed the isotropic one, in case differences were observed. In contrast, the numerical applications have shown that the efficiency of the IP method is highly sensitive to the choice of the viscosity scale. On the one hand, the naive facewise choice does not guarantee the coercivity. It performs well on overresolved meshes but (often) fails if the mesh resolution is reduced. On the other hand, the *maxMin* and the *elementwise* definitions are both formally coercive. By including the recent trace inverse inequalities, the former choice is a sharper variant of the proposal of Epshteyn [57]. However, its extremely high sensitivity to local and transient variations of the viscosity makes the *maxMin* scale unsuitable for RANS computations. Conversely, by accounting - at least approximately - for the variation of the viscosity, the newly proposed *elementwise* scale proved to be much more robust, as it only depends on the maximum value.

Finally, we have considered a hybrid discretization of the RANS system. According to the numerical tests for a flat plate the use of higher-order interpolations of the Spalart-Allmaras variable seems not indispensable to guarantee an overall high-order accuracy but is strongly recommended for stability reasons.

Chapter 4

Time integration and iterative solution methods

Contents

In chapter 3 we have introduced and discussed the spatial discretization of the Navier-Stokes / RANS equations with the Discontinuous Galerkin method. So far however, we have left aside the question of how to advance in time the resulting system of ordinary differential equations. This is the subject of the present chapter.

In the first part of this chapter (section 4.1), we briefly summarize the most important *time integration schemes* currently used within this thesis. Although explicit schemes such as the backward Euler or the Runge-Kutta method are easy to implement and highly parallelizable, they suffer from severe time step restrictions. These restrictions are overcome by the use of implicit schemes, at the cost, however, of an increased computational complexity. In particular, the solution of a large system of equations is required at each time step. In this work, the solution is based on a matrix-free Newton-Krylov method.

In the second part (section 4.2), we focus on *iterative algorithms* to solve the linear system of equations arising from the Newton linearization. After a review of the main features of the GMRES solver in section 4.2.1, two widely used preconditioning techniques are considered in section 4.2.2, namely the block Jacobi and the incomplete LU-factorization (ILU).

Finally, the implementation of the *Jacobian matrix* is presented in section 4.3. Furthermore, different aspects to enhance the efficiency of the Jacobian-free Newton-GMRES solver are discussed in section 4.3.1.

Note that as well the mentioned time integration schemes as the in-house developed iterative solvers were already available in Argo. The main contribution of this chapter consists in extending a formerly approximate linearization of the (laminar) Navier-Stokes equations to the exact linarization of the coupled RANS system. The resulting improvement of the iterative solver is illustrated by several numerical examples.

4.1 Time integration

In chapter 3 we have derived the so-called *semi-discrete* problem associated to the Discontinuous Galerkin formulation. Recalling equations (3.7) to (3.9), the following expression

$$\int_T \phi_i \frac{\partial \mathbf{u}_h}{\partial t}\, d\Omega + \mathbf{R}_h|_T(\mathbf{u}_h, \phi_i) = 0 \tag{4.1}$$

must hold for every cell $T \in \mathcal{T}$ and for every shape function $\phi_i \in \Phi^p$ ($i = 1, \ldots, N_\phi$). Taking into account the definition (3.4) of the discrete solution $\mathbf{u}_h$, equation (4.1) can be written as a system of ordinary (first order) differential equations in time, i.e.

$$\mathbf{M}\frac{\partial \mathbf{U}}{\partial t} = \mathbf{R}(\mathbf{U}), \tag{4.2}$$

with $\mathbf{U}(t) = [\mathbf{U}_1(t), \mathbf{U}_2(t), \ldots, \mathbf{U}_k(t), \ldots]^T$ the global vector of degrees of freedom, $\mathbf{M}$ the (global) mass matrix operator and $\mathbf{R}$ the vector residual function. $\mathbf{U}_k(t)$ denotes the degrees of freedom associated to the k-th element.

Due to the local support of the shape functions, the global mass matrix has a block-diagonal structure with block size $n = N_\phi N_v$[1]. Furthermore, assuming that the unknowns are organized by grouping physical variables, i.e. $\mathbf{U}_k = [\rho_1^k, \ldots, \rho_{N_\phi}^k, \rho u_1^k, \ldots, \rho u_{N_\phi}^k, \ldots, \tilde{\mu}_1^k, \ldots, \tilde{\mu}_{N_\phi}^k]^T$, each entry of the global matrix has itself a block-diagonal structure and is composed of N_v element mass matrices $\mathbf{M}_T$ of size $N_\phi \times N_\phi$. The element mass matrices are computed in the reference space as

$$(\mathbf{M}_T)_{ij} = \int_{T_{Ref}} \phi_i \phi_j |\mathbf{J}_T|\, d\xi d\eta d\zeta, \qquad 1 \le i, j \le N_\phi, \tag{4.3}$$

[1] N_ϕ and N_v denote the number of shape functions and the length of the state vector, respectively (cf. section 3.2.1).

with $\mathbf{J}$ the mapping Jacobian defined in section 3.2.1. Given the relative small block size, the mass matrix can easily be inverted once and for all at the beginning of the computation. It is worth noting that in the case of straight sided elements, $\mathbf{J}_T$ is constant and the element mass matrix simplifies to $(\mathbf{M}_T)_{ij} = |\mathbf{J}_T| \int \phi_i \phi_j \, d\xi d\eta d\zeta = |\mathbf{J}_T| \boldsymbol{\mathcal{M}}_{ij}$. As a consequence, the parametric matrix $\boldsymbol{\mathcal{M}}$ has to be inverted only once.

Equation (4.2) defines the time-continuous Discontinuous Galerkin problem. To integrate (4.2) by marching in time from an initial solution $\mathbf{U}(0) = \mathbf{U}_0$, a variety of time discretization schemes have been proposed in the literature. The choice of the appropriate discretization scheme depends on the dynamics of the problem under consideration. On the one hand, for fast dynamic problems requiring an accurate time integration, the time step is often constraint by the physics and hence, explicit schemes are most adapted. On the other hand, if the flow changes slowly in time, numerical constraints on the stability become the limiting factor and implicit schemes providing an enhanced stability are generally preferred. Sections 4.1.1 and 4.1.2 present the basic features of the currently used explicit and implicit integration schemes that are available in Argo.

4.1.1 Explicit time integration schemes

As their name implies, since the right-hand-side of equation (4.2) is evaluated at the *previous/old* time level $t^n = n\Delta t$, these discretization schemes provide an explicit expression of the unknowns $\mathbf{U}^{n+1}$ at the *new* time level t^{n+1}. Here, Δt is the time step and the superscript n denotes the time level.

The two explicit time discretization schemes implemented in DGArgo are the *explicit Euler scheme*

$$\begin{aligned} \mathbf{M}\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} &= \mathbf{R}(\mathbf{U}^n) \\ \Longrightarrow \mathbf{U}^{n+1} &= \mathbf{U}^n + \Delta t \mathbf{M}^{-1}\mathbf{R}(\mathbf{U}^n) \end{aligned} \tag{4.4}$$

and the *(explicit) m-stage Runge-Kutta (RK) scheme* which can be written

$$\begin{aligned} \mathbf{k}_i &= \mathbf{M}^{-1}\mathbf{R}(\mathbf{U}^n + \alpha_i \Delta t \mathbf{k}_{i-1}), \quad \text{with} \quad \mathbf{k}_0 = \mathbf{M}^{-1}\mathbf{R}(\mathbf{U}^n), \\ \mathbf{U}^{(0)} &= \mathbf{U}^n, \\ \mathbf{U}^{(i)} &= \mathbf{U}^{(i-1)} + \beta_i \Delta t \mathbf{k}_i \quad 1 \leq i \leq m, \\ \mathbf{U}^{n+1} &= \mathbf{U}^{(m)}. \end{aligned} \tag{4.5}$$

The superscript enclosed in parentheses indicates the RK stage; α_i and β_i are the RK stage coefficients. The initial aim of the Runge-Kutta method was to develop a high-order time accurate discretization scheme by combining the solution vectors at several intermediate RK stages between t^n and t^{n+1}.

Over the last decades a variety of RK coefficients have been proposed optimizing either the time accuracy, the stability or the convergence to steady state (cf. [58, 93, 168, 169]).

The main advantages of explicit schemes are the straightforward implementation, the low storage requirements and the high parallelization efficiency since data exchanges between neighbouring domains is reduced to a strict minimum. Unfortunately the computational time step of explicit schemes is limited by the *Courant-Friedrichs-Levy (CFL)* stability condition [40]. Due to this severe restriction explicit schemes are most of the time inappropriate for wall bounded flows or steady state computations. Therefore, all RANS computations in this thesis are carried out using an implicit time integration method.

4.1.2 Implicit time integration schemes

The implicit time integration schemes are based on the following two-parameter family

$$\frac{\mathbf{M}}{\Delta t}\left((1+\varphi)\mathbf{U}^{n+1}+\frac{1}{\varphi-1}\mathbf{U}^{n}+\varphi\mathbf{U}^{n-1}\right)=\vartheta\mathbf{R}(\mathbf{U}^{n+1})+(1-\vartheta)\mathbf{R}(\mathbf{U}^{n}). \tag{4.6}$$

As can be seen from (4.6) the right-hand-side combines residual vectors taken at the previous but also at the new time level. The latter is necessary in order to overcome the time step limitation of explicit schemes (see Dahlquist [44]).

Choosing $\varphi = 0$ and $\vartheta = 1$, the above expression simplifies to the *implicit Euler scheme*

$$\mathbf{M}\frac{\mathbf{U}^{n+1}-\mathbf{U}^{n}}{\Delta t}=\mathbf{R}(\mathbf{U}^{n+1}) \tag{4.7}$$

which is first order accurate in time. If time accuracy is important, selecting $\varphi = 1/2$ and $\vartheta = 1$ results in the so-called *three point backward scheme*

$$\mathbf{M}\frac{\frac{3}{2}\mathbf{U}^{n+1}-2\mathbf{U}^{n}+\frac{1}{2}\mathbf{U}^{n-1}}{\Delta t}=\mathbf{R}(\mathbf{U}^{n+1}). \tag{4.8}$$

The latter is widely used for unsteady viscous computations because of its accuracy[2]and its robustness. Note that neither (4.7) nor (4.8) require the storage of the residual vector $\mathbf{R}$ at previous time levels.

Newton's method

Whatever the choice of the implicit scheme a large system of non-linear (algebraic) equations has to be solved at each time level n. A classical algorithm

[2]The three point backward scheme is second order accurate in time.

to solve these systems is Newton's method. Thereto, we linearize the residual around the global vector of degrees of freedom

$$\mathbf{R}(\mathbf{U}^{(k)}) \approx \mathbf{R}(\mathbf{U}^{(k-1)}) + \left.\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right|^{(k-1)} \left(\mathbf{U}^{(k)} - \mathbf{U}^{(k-1)}\right). \tag{4.9}$$

The superscript enclosed in parentheses indicates the Newton step.

Starting from an initial estimate $\mathbf{U}^{(0)} = \mathbf{U}^n$ given by the previous time level, the iterative Newton method for the implicit Euler scheme can be written

$$\begin{aligned}
&\mathbf{U}^{(0)} = \mathbf{U}^n, \\
&\underbrace{\left[\frac{\mathbf{M}}{\Delta t} - \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]}_{\mathbf{A}} \underbrace{\Delta \mathbf{U}^{(k)}}_{\mathbf{x}} = \underbrace{\mathbf{R}(\mathbf{U}^{(k-1)}) - \frac{\mathbf{M}}{\Delta t}\left(\mathbf{U}^{(k-1)} - \mathbf{U}^n\right)}_{\mathbf{b}}, \\
&\mathbf{U}^{(k)} = \mathbf{U}^{(k-1)} + \Delta \mathbf{U}^{(k)}, \quad k = 1, \dots m, \\
&\mathbf{U}^{n+1} = \mathbf{U}^{(m)},
\end{aligned} \tag{4.10}$$

where $\mathbf{A}$ is the Jacobian matrix of the time dependent system (4.7), including contributions of both the spatial and the temporal discretization. m denotes the number of (non-linear) Newton iterations.

An appealing property of Newton's method is the quadratic convergence to the exact solution, provided a sufficiently good initial guess. However, the major drawback of algorithm (4.10) is the necessity to solve a linear system of equations at each Newton iteration, which makes the method expensive for large applications. Before summarizing in section 4.2 how we solve this system in practice, some remarks allowing to significantly reduce the computational cost for steady state computations are in order.

Steady state computations: inexact Newton's method

Inexact Newton methods constitute one of the most effective tools for solving systems of non-linear equations. In the previous section we have seen that a considerable part of the computational effort of Newton's method results from solving the linearized Newton equation. However, if we are only interested in the stationary solution, time accuracy is not required and the question arises to which level of accuracy the non-linear equations (4.7) or (4.8) must be solved. Especially at the beginning of the computation when the iterate is still far from steady flow conditions, the exact solving of the non-linear system seems questionable.

Considering a general system of non-linear equations $\mathbf{F}(\mathbf{x}) = 0$, Dembo, Eisenstat and Steighaug [48] were among the first to analyze the convergence of inexact Newton methods. In particular they have proved that these methods are locally convergent provided that

$$\frac{\| \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_k \Delta \mathbf{x}^k + \mathbf{F}(\mathbf{x}_k)\|}{\|\mathbf{F}(\mathbf{x}_k)\|} \leq \eta_k, \tag{4.11}$$

where the forcing term η_k is a constant smaller than unity. Furthermore, they have studied the order of convergence in terms of η_k. Since then, several authors have proposed criteria to determine the forcing term in order to improve the efficiency and the robustness of inexact Newton methods, e.g. [3,48,55,131]. Note that in some cases (see [157, 172]), oversolving ($\eta_k \ll$) the linearized Navier-Stokes equations in early iterations may cause inadequate Newton updates which even deteriorate the (global) convergence.

In practice we proceed as follows:

- *time-dependent flows.* Several Newton iterations must be performed at each time level until a sufficient drop in the norm of the residual vector is achieved. However, since the computational time step is small, the old time level provides a good initial guess. Hence Newton's method generally converges in a few iterations only. Furthermore, a small time step improves the conditioning number of the linear system which reduces the cost of solving this system (see section 4.2).

- *stationary flow.* Since time accuracy is not required, we use an implicit Euler scheme and only one Newton iteration is performed per time level. This approach is commonly used in the literature, e.g. [18,107,110,136].

Why we solve time-dependent equations for stationary flows. Assume one wants to perform a steady state computation. Why not simply discard the temporal derivative in (4.2) and apply Newton's method directly to the stationary equation $\mathbf{R}(\mathbf{U}) = 0$? The reasons we prefer to keep a "pseudo-time dependency" are the following: (i) *globalization*, to ensure the convergence of Newton's method, the initial iterate must lie in the vicinity of the expected root; (ii) *stability*, the underrelaxation introduced by the time stepping avoids non-physical intermediate states; (iii) *improved conditioning*, the temporal term $\mathbf{M}/\Delta t$ enforces the diagonal dominance of the corresponding linear system [88].

Moreover, the convergence of the method to the stationary solution is accelerated by *local time-stepping*. Therefore, we choose for each computing cell T_k the maximum allowable time step $\Delta\tau_k$ such that a constant CFL number is maintained across the domain.

4.1.3 Choice of the time step size

For stationary computations, a local time step $\Delta\tau_k$ is determined for each computing cell T_k based on the following relations

$$\begin{aligned}\Delta\tau_k^n &= \frac{CFL^n}{\lambda_c + \lambda_v},\\ \lambda_c &= \frac{4f_1(p)U_{max}}{h},\\ \lambda_v &= \frac{3df_1(p)|C_{IP}|\mu_{max}}{h^2}.\end{aligned} \tag{4.12}$$

As illustrated by the above expressions (4.12), the time step includes contributions from the convective λ_c and the viscous λ_v terms. U_{max} and μ_{max} denote the maximum wave speed and the maximum viscosity, respectively; h is a measure of the element size and d represents the number of space dimensions. Furthermore, we take into account the dependency of the characteristic length on the interpolation order p by the function $f_1(p) = 1/3p^2 + 7/6p + 1$. For time-dependent computations, a global time step $\Delta\tau^n$ is defined as the minimum time step of all computing cells, i.e.

$$\Delta\tau^n = \min_k(\Delta\tau_k^n). \tag{4.13}$$

In order to reach a quadratic convergence at the last Newton iterations, the CFL number is progressively increased according an exponential law

$$CFL^n = \max\left(CFL^0\left(\frac{\|\hat{\mathbf{R}}^0\|_2}{\|\hat{\mathbf{R}}^{n-1}\|_2}\right)^{\alpha}, CFL^{\infty}\right), \tag{4.14}$$

where $\hat{\mathbf{R}}^n$ denotes the residual vector at time level n including contributions of the spatial and the temporal discretization. CFL^0, CFL^{∞} and α are user defined control parameters. Typical[3] values for steady RANS computations are $CFL^0 \sim 100$, $CFL^{\infty} \sim 10^8$ and $\alpha \in [0.5, 1.5]$. For a comparison of different CFL evolution strategies namely "predetermined", "residual-based" and "locally optimal" strategies we refer to the thesis of B. Pollul [136]. Numerical experiments carried out by Pollul have however shown that there is no clear winner among the different strategies.

To improve the robustness of the RANS equations, several strategies intended to prevent the appearance of negative values of the Spalart-Allmaras working variable have been proposed in the literature (cf. chapter 2). In particular Spalart and Allmaras [165] suggested to modify the Jacobian matrix

[3] Because of the important storage requirement of ILU preconditioners in the case of three-dimensional computations, we had to use a block-Jacobi preconditioner instead. Given the poor conditioning of the resulting linear system, we limited the maximum CFL number to $CFL^{\infty} \sim 10^5$.

of the turbulence model. The principal drawbacks of this approach are the significant deterioration of the Newton method and its incompatibility with a matrix-free Krylov solver, see section 4.2. For these reasons we did not implement this approach in our code.

Based on the observation that the first Newton iterations are the most critical for the stability, we generally limit the CFL number during the start-up of the computation. Thereto, a second set with smaller CFL control parameters is defined[4]. After the turbulence model is "fully" active (which is supposed to happen if the residual norm has decreased by ~ 2 to 3 orders of magnitude) Newton type convergence is achieved using a second set of larger CFL parameters. A similar approach has been reported by Chisholm [34] and Chisholm and Zingg [35].

In contrast to Chisholm who defines a different time step for the turbulence model, we use the same (local) time step to integrate simultaneously the Navier-Stokes equations and the turbulence model.

If nonetheless the computations breaks down[5], the time integration from level $n-1$ to level n is automatically repeated with a CFL number divided by two. This can be done several times till convergence is reached. The restart turned out to be useful especially during the start-up for complex three-dimensional flows, where a "good choice" of CFL parameters can sometimes be quite tricky.

4.2 Iterative methods for solving linear systems

In the previous section we have seen that a linear system of equations has to be solved at each Newton iteration. The present section outlines how we solve this system in practice.

Given the large number of degrees of freedom, *direct solution methods* such as Gaussian elimination are by far too costly for "real world applications", as well in terms of storage requirements as in terms of CPU time[6]. In these cases, *iterative projection techniques* based on Krylov subspaces are currently considered to be the most efficient methods for solving large linear systems [152].

4.2.1 Jacobian-free GMRES

A popular method for solving sparse linear systems $\mathbf{Ax} = \mathbf{b}$ which is applicable to non-symmetric matrices is the *Generalized Minimum RESidual (GMRES)*

[4] During the start up, we typically limit the CFL number to values between 10^3 and 10^4 while $\alpha \in [0.5, 0.7]$.

[5] The breakdown test simply verifies if the L^2-norm of the residual vector is a positive, real number.

[6] Approximately $\mathcal{O}(n^3)$ operations are needed to solve a system of n equations by Gaussian elimination.

algorithm [153]. It is the method of choice in this thesis.

As for any Krylov subspace method, we seek an approximate solution $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m$ such that the Petrov-Galerkin condition is satisfied

$$\mathbf{A}\mathbf{x}_m - \mathbf{b} \perp \mathcal{L}_m, \quad \mathbf{A} \in \mathbb{R}^{n\times n}. \tag{4.15}$$

The Krylov subspaces $\mathcal{K}_m$ and $\mathcal{L}_m$ (of dimension m) are characterized as

$$\mathcal{K}_m(\mathbf{A}, \mathbf{v}) = \text{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \mathbf{A}^2\mathbf{v}, \ldots, \mathbf{A}^{m-1}\mathbf{v}\}, \quad \text{and} \quad \mathcal{L}_m = \mathbf{A}\mathcal{K}_m \tag{4.16}$$

with $\mathbf{v} = \hat{\mathbf{R}}^0/\|\hat{\mathbf{R}}^0\| \in \mathbb{R}^n$. The size of the Krylov subspaces is increased by one at each iteration step m and an orthogonal basis is built using Arnoldi's method [5]. For detailed information about practical implementation issues we refer to [152].

Probably the most important drawback of the GMRES algorithm is its high memory requirement since all previous basis vectors have to be stored. Furthermore, the computational cost of the projection method increases with the size of the Krylov subspace. This renders the algorithm impractical for large m. To overcome these limitations, a common remedy consists in restarting the algorithm after m^* iterations. However, a well known difficulty of the restarted GMRES(m^*) algorithm is that it can stagnate when the matrix is not positive definite [152].

Alternative Lanczos-based methods such as the *stabilized Bi-Conjugate Gradient algorithm (BICGSTAB)* [174] have the significant advantage to not require the storage of the entire Krylov subspace. Unfortunately, there are more potential situations causing the breakdown of these algorithms. Comparative studies [124] have shown that there is no clear winner between the various iterative methods for solving linear systems and that the "best choice" is problem dependent.

A significant part of the memory requirement of the iterative solver is related to the storage of the Jacobian matrix $\mathbf{A}$ which is large for DGFEM. Based on the observation that to construct the Krylov subspace $\mathcal{K}_m$ only matrix-vector products of the Jacobian are needed (but not individual elements of the Jacobian), these products are commonly formed using a *Jacobian-free* approach. A first order Taylor series expansion leads to

$$\mathbf{A} \cdot \mathbf{v} \approx \frac{\hat{\mathbf{R}}(\mathbf{U} + \epsilon\mathbf{v}) - \hat{\mathbf{R}}(\mathbf{U})}{\epsilon}, \tag{4.17}$$

where ϵ denotes a small perturbation parameter. The choice of this parameter results from a compromise between an accurate approximation of the derivative on the one hand and (numerical) round-off errors on the other hand. In practice we take

$$\epsilon = \sqrt{\psi}\frac{\|\mathbf{U}\|_2}{\|\mathbf{v}\|_2}, \tag{4.18}$$

where ψ is the relative round-off error associated to double precision [88].

The convergence properties of the resulting Jacobian-free GMRES method have been studied in [22–24]. For a general review of existing Jacobian-free Newton-Krylov solvers we refer to the paper of Knoll and Keyes [97]. Here, we just want to remember that the Jacobian-free approach (4.17) is known to suffer from convergence/stability problems in the presence of (i) sharp non-linear solution structures (e.g. shocks) and (ii) discontinuities in the non-linear residual function (e.g. clipping[7]) [97].

4.2.2 Preconditioning techniques

The reliability and the robustness of iterative Krylov solvers highly depends on the conditioning of the linear system. As a consequence, the performance of the linear solver is often more affected by the preconditioning technique than by the choice of a particular Krylov method, e.g., GMRES, BICGSTAB [152].

The aim of preconditioning is to improve the convergence of the linear solver by clustering the eigenvalues of the iterative matrix, were a compromise must be found between the additional computational cost of the preconditioner and the acceleration of the Krylov method. In the present work, we use a *right preconditioning*[8] technique which transforms the original linear system $\mathbf{Ax} = \mathbf{b}$ into

$$(\mathbf{AP})\mathbf{y} = \mathbf{b} \quad \text{and} \quad \mathbf{x} = \mathbf{Py}. \tag{4.19}$$

Note that in practice it is not necessary to explicitly compute the intermediate variable $\mathbf{y}$. Details about the implementation of preconditioned linear solvers can be found in [152].

In principal, any (simple) operator/iterative method $\mathbf{P}$ can be used as a preconditioner, provided that

$$\mathbf{Px} \approx \mathbf{A}^{-1}\mathbf{x}. \tag{4.20}$$

In other words, the preconditioner $\mathbf{P}$ should approximate $\mathbf{A}^{-1}$, while being reasonably cheap to compute. Remark that since at least an approximation of the Jacobian is needed to precondition the linear system, the resulting "Jacobian-free GMRES" solver is strictly speaking no longer Jacobian-free. The two preconditioners used within this work are the *block Jacobi* and the *BILU* preconditioner.

[7] However, for the test cases examined in this thesis, no convergence problems related to the clipping of the turbulence model have been observed, cf. chapter 2.

[8] Alternative approaches are the *left preconditioning* $(\mathbf{PA})\mathbf{x} = \mathbf{Pb}$ and the *double-sided preconditioning* $(\mathbf{P}_1\mathbf{AP}_2)(\mathbf{P}_2^{-1}\mathbf{x}) = \mathbf{P}_1\mathbf{b}$. While right-preconditioned GMRES results in a solution vector which minimizes the norm of the residual vector $\|\hat{\mathbf{R}}\|$, left-preconditioning minimizes $\|\mathbf{P}\hat{\mathbf{R}}\|$. For most practical applications no significant difference is however observed between left and right preconditioning, except when $\mathbf{P}$ is ill-conditioned, see [152].

The block Jacobi or block-diagonal preconditioner can be written setting all blocks except the diagonal blocks to zero, i.e.

$$\mathbf{P}_{ij} = \begin{cases} (\mathbf{A}_{ij})^{-1} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.21}$$

It is computationally cheap and except of the diagonal blocks no further storage is required. Since it performs poorly for general problems, e.g. [132, 134], we use this preconditioner mainly for time dependent applications or three-dimensional problems where the storage of the Jacobian becomes the limiting factor.

The BILU preconditioner is based on a block incomplete lower-upper factorization of the Jacobian leading to a well invertible matrix, i.e. $\tilde{\mathbf{A}}^{ILU} = \tilde{\mathbf{L}}\tilde{\mathbf{U}} \approx \mathbf{A}$. The preconditioner is then computed as $\mathbf{P} = (\tilde{\mathbf{A}}^{ILU})^{-1} = \tilde{\mathbf{U}}^{-1}\tilde{\mathbf{L}}^{-1}$. Starting point for the approximate factorization is a Gaussian elimination where some off-diagonal blocks are dropped according to a predefined *fill-in* strategy. The resulting algorithm is commonly known as BILU(p); p denoting the level of fill-in [152]. In particular, for the BILU(0) preconditioner, only those blocks initially present in the sparsity pattern of $\mathbf{A}$ are allowed during the approximate factorization. A better control of the addition of new (off-diagonal) entries is provided by the BILU(p,τ) approach, where p and τ are the number of new entries per row and a threshold respectively.

Despite the relative high cost of factorization and inversion, block incomplete factorizations are considered to be among the most efficient preconditioning techniques. The reason is the significant reduction of the number of Krylov iterations compared to e.g. Jacobi preconditioned solvers. The main disadvantage of BILU is its large memory footprint which becomes the limiting factor for three-dimensional applications using high-order polynomials. For two-dimensional applications we usually prefer a BILU(1) preconditioner, while for three-dimensional computations with low order polynomials ($p \leq 2$) BILU(0) provides a good compromise between storage requirements and preconditioning quality.

Finally, since the efficiency of the preconditioner is strongly influenced by the ordering of the unknowns, we use the Reverse Cuthill-McKey reordering algorithm [41, 42, 65] to reduce the bandwidth of $\mathbf{A}$.

4.3 Jacobian evaluation

In the previous section the main characteristics of the "Jacobian-free GMRES" solver have been discussed. In particular we have seen that although the construction of the Krylov space is based on a finite difference approach (4.17) at least an approximation of the Jacobian/tangent matrix is need as a preconditioner. Since in this work all terms are handled by an implicit time integration, the Jacobian includes contributions from the inviscid, the viscous

and the source term. Similarly to the computation of the mass matrix or the residual vector the linearization is performed on the element level; the contributions from each element are then assembled to the global matrix.

Because of the compactness of the present Interior Penalty discretization the resulting (global) Jacobian matrix has a block-sparse structure, with N_n+1 blocks of size $n = N_\phi N_v$ per row. N_n denotes the number of neighbouring cells of the element under consideration. While the off-diagonal blocks are related to the coupling between adjacent elements, the diagonal blocks describe the coupling between degrees of freedom inside the same element. A detailed description of the data structure and the assembly of the Jacobian matrix can be found in [33, 88].

Except of the inviscid boundary conditions which are linearized by finite differences, all the other terms are linearized analytically. For three-dimensional RANS computations the analytical derivation of the viscous terms is in our experience approximately 20 to 30 % faster than a first order finite difference linearization.

In [88] Hillewaert *et al.* describe the efficient implementation of a BILU preconditioner based on an approximate derivation of the viscous terms; the inviscid terms are linearized exactly. While this method leads to a high-performance algorithm for laminar flows, a suboptimal convergence is achieved in the case of turbulent flows. Since the inviscid terms are treated in the same manner as in [88], only the linearization of the viscous terms is discussed here.

The viscous volume term. According to chapter 3, the viscous volume term (3.50) is computed by Gaussian quadrature leading to the following expression

$$DV_{im} = \int_T \nabla\phi_i \cdot \boldsymbol{\mathcal{F}}^v_m \, d\Omega = \sum_{q=1}^{N_q} w_q \left(|\mathbf{J}_T| \sum_{k=1}^{d}\sum_{u=1}^{d} \frac{\partial\phi_i}{\partial\xi_u}\frac{\partial\xi_u}{\partial x_k}\boldsymbol{\mathcal{F}}^v_{k,m} \right)_{\nu_q} . \quad (4.22)$$

The linearization of (4.22) reads

$$\begin{aligned}\frac{\partial DV_{im}}{\partial \boldsymbol{u}_{jn}} = &\underbrace{\sum_{q=1}^{N_q}\sum_{u=1}^{d}\sum_{v=1}^{d} \left(w_q \frac{\partial\phi_i}{\partial\xi_u}\frac{\partial\phi_j}{\partial\xi_v} \right)_{\nu_q} \cdot \left(|\mathbf{J}_T| \sum_{k=1}^{d}\sum_{l=1}^{d} \frac{\partial\xi_u}{\partial x_k} D^{kl}_{mn} \frac{\partial\xi_v}{\partial x_l} \right)_{\nu_q}}_{\text{term I}} \\ &+ \underbrace{\sum_{q=1}^{N_q}\sum_{u=1}^{d} \left(w_q \frac{\partial\phi_i}{\partial\xi_u}\phi_j \right)_{\nu_q} \cdot \left(|\mathbf{J}_T| \sum_{k=1}^{d}\sum_{l=1}^{d}\sum_{o=1}^{N_v} \frac{\partial\xi_u}{\partial x_k}\frac{\partial D^{kl}_{mo}}{\partial \boldsymbol{u}_n}\frac{\partial \boldsymbol{u}_o}{\partial x_l} \right)_{\nu_q}}_{\text{term II}} .\end{aligned} \quad (4.23)$$

Notice that the above linearization (4.23) is valid since the viscous flux can be written as a linear function of the solution gradients, cf. equations (2.3), (2.10) and (3.37).

In [88] only the first term of the derivative (4.23) has been implemented, while the second term has been neglected. In other words, the viscous Jacobian tensor D_{mn}^{kl} was supposed to be constant. A straightforward calculation shows that the linearization error committed by this approach is proportional to the solution gradients $\nabla \mathbf{U}$. This explains the different behaviour of the original preconditioner for the laminar and the turbulent case because much stronger gradients are observed for turbulent flows.

An important aspect for the efficient implementation is the separation of shape functions ϕ_i and physical contributions (state functions, local-to-global mapping,...). This allows to build the Jacobian matrix by adding, for each quadrature point q and variable combination (m,n), precomputed parametric subblocks $\mathfrak{D}_q^{uv}$ and $\mathfrak{C}_q^u$ with respective weights $\delta_{q,mn}^{1,uv}$ and $\delta_{q,mn}^{2,u}$, i.e.,

$$\frac{\partial DV_{im}}{\partial \boldsymbol{u}_{jn}} = \sum_{q=1}^{N_q} \sum_{u=1}^{d} \sum_{v=1}^{d} \mathfrak{D}_q^{uv} \cdot \delta_{q,mn}^{1,uv} + \sum_{q=1}^{N_q} \sum_{u=1}^{d} \mathfrak{C}_q^u \cdot \delta_{q,mn}^{2,u}. \tag{4.24}$$

Since the parametric matrix $\mathfrak{C}_q^u$ is the same as for the linearization of the convective volume term, their respective weights are added before assembling the Jacobian subblocks.

Diffusive interface terms. Remember that for the closed Lagrangian interpolation used in this work, see section 3.2.1, only a subset of the element shape functions has to be considered for the interface terms. In practice, we compute these terms defining a set of local shape functions ψ_k, which are the restrictions of ϕ_k on $\mathcal{F} \in \partial T$, together with the associated quadrature rules (β_q, v_q). β_q and v_q denote the quadrature point and weight respectively $(1 \leq q \leq N_\beta)$.

According to (3.60), the (direct) diffusive interface term DI_{im}^- and the transpose term DT_{im}^- can be expanded as

$$\begin{aligned}
DI_{im}^- &= \oint_{\mathcal{F}} [\![\phi_i]\!]_k \left\langle D_{mn}^{kl} \frac{\partial u_{h_n}}{\partial x_l} \right\rangle \, dS \\
&= \sum_{j=1}^{N_\phi} \sum_{q=1}^{N_\beta} \frac{v_q}{2} \left[\sum_{k=1}^{d} \sum_{l=1}^{d} \phi_i^- n_k^- D_{mn}^{kl} \left(\frac{\partial \phi_j^-}{\partial x_l} \boldsymbol{u}_{jn}^- + \frac{\partial \phi_j^+}{\partial x_l} \boldsymbol{u}_{jn}^+ \right) |\mathbf{J}_{\mathcal{F}}| \right]_{\beta_q}, \\
DT_{im}^- &= \theta \oint_{\mathcal{F}} [\![u_{h_n}]\!]_k \left\langle D_{nm}^{kl} \frac{\partial \phi_i}{\partial x_l} \right\rangle \, dS \\
&= \theta \sum_{j=1}^{N_\phi} \sum_{q=1}^{N_\beta} \frac{v_q}{2} \left[\sum_{k=1}^{d} \sum_{l=1}^{d} \left(\phi_j^- \boldsymbol{u}_{jn}^- n_k^- + \phi_j^+ \boldsymbol{u}_{jn}^+ n_k^+ \right) D_{nm}^{kl} \frac{\partial \phi_i^-}{\partial x_l} |\mathbf{J}_{\mathcal{F}}| \right]_{\beta_q}
\end{aligned} \tag{4.25}$$

and similar expressions are obtained for the right state $\mathcal{F}^+$ (remember that $\mathbf{n}^- = -\mathbf{n}^+$). As an example, only one of the (off-diagonal) Jacobian subblocks

is detailed here. The linearization of the remaining three blocks can be realized in a similar fashion. For the diffusive interface term, we get

$$\frac{\partial DI_{im}^{-}}{\partial \boldsymbol{u}_{jn}^{+}} = \underbrace{\sum_{q=1}^{N_\beta}\sum_{u=1}^{d}\left(v_q\phi_i^{-}\frac{\partial\phi_j^{+}}{\partial\xi_u}\right)_{\beta_q}\cdot\left(\frac{|\mathbf{J}_{\mathcal{F}}|}{2}\sum_{k=1}^{d}\sum_{l=1}^{d}n_k^{-}D_{mn}^{kl}\left(\frac{\partial\xi_u}{\partial x_l}\right)^{+}\right)_{\beta_q}}_{\text{term I}}$$
$$+\underbrace{\sum_{q=1}^{N_\beta}\left(v_q\phi_i^{-}\phi_j^{+}\right)_{\beta_q}\cdot\left(\frac{|\mathbf{J}_{\mathcal{F}}|}{2}\sum_{k=1}^{d}\sum_{l=1}^{d}\sum_{o=1}^{N_v}n_k^{-}\frac{\partial D_{mo}^{kl}}{\partial\boldsymbol{u}_n}\left(\frac{\partial\boldsymbol{u}_o^{-}}{\partial x_l}+\frac{\partial\boldsymbol{u}_o^{+}}{\partial x_l}\right)\right)_{\beta_q}}_{\text{term II}}. \tag{4.26}$$

Here the derivation of the viscous Jacobian tensor has been computed according to the chain rule

$$\frac{\partial D_{mo}^{kl}}{\partial \boldsymbol{u}_{jn}^{+}} = \frac{\partial D_{mo}^{kl}}{\partial \boldsymbol{u}_n}\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{u}_{jn}^{+}} = \frac{\partial D_{mo}^{kl}}{\partial \boldsymbol{u}_n}\phi_j^{+}. \tag{4.27}$$

The linearization of the transpose term reads

$$\frac{\partial DT_{im}^{-}}{\partial \boldsymbol{u}_{jn}^{+}} = \underbrace{\theta\sum_{q=1}^{N_\beta}\sum_{u=1}^{d}\left(v_q\phi_i^{+}\frac{\partial\phi_j^{-}}{\partial\xi_u}\right)_{\beta_q}\cdot\left(\frac{|\mathbf{J}_{\mathcal{F}}|}{2}\sum_{k=1}^{d}\sum_{l=1}^{d}n_k^{+}D_{nm}^{kl}\left(\frac{\partial\xi_u}{\partial x_l}\right)^{-}\right)_{\beta_q}}_{\text{term I}}$$
$$+\underbrace{\theta\sum_{q=1}^{N_\beta}\sum_{u=1}^{d}\left(v_q\phi_i^{+}\frac{\partial\phi_j^{-}}{\partial\xi_u}\right)_{\beta_q}\cdot\left(\frac{|\mathbf{J}_{\mathcal{F}}|}{2}\sum_{k=1}^{d}\sum_{l=1}^{d}\sum_{o=1}^{N_v}[\![\boldsymbol{u}_n]\!]_k\frac{\partial D_{mo}^{kl}}{\partial\boldsymbol{u}_n}\left(\frac{\partial\xi_u}{\partial x_l}\right)^{-}\right)_{\beta_q}}_{\text{term II}}, \tag{4.28}$$

where we use the shorthand notation $[\![\boldsymbol{u}_n]\!]_k = \sum_{j=1}^{d}\left(\phi_j^{-}\boldsymbol{u}_{jn}^{-}n_k^{-}+\phi_j^{+}\boldsymbol{u}_{jn}^{+}n_k^{+}\right)$.

As for the diffusive volume term, only the first contributions ("term I") have been implemented in [88] while the linearization of the diffusive stress tensor ("term II") has been neglected.

Again, the Jacobian is assembled adding precomputed parametric matrices $\mathfrak{d}_{q,ij}^{ab,u}$ and $\mathfrak{m}_{q,ij}^{ab}$ with their respective weights, i.e.

$$\begin{aligned}\frac{\partial DI_{im}^{a}}{\partial \boldsymbol{u}_{jn}^{b}} &= \sum_{q=1}^{N_\beta}\sum_{u=1}^{d}\mathfrak{d}_{q,ij}^{ab,u}\cdot\gamma_{q,mn}^{u,b}+\sum_{q=1}^{N_\beta}\mathfrak{m}_{q,ij}^{ab}\cdot\delta_{q,mn}^{ab},\\ \frac{\partial DT_{im}^{a}}{\partial \boldsymbol{u}_{jn}^{b}} &= \theta\sum_{q=1}^{N_\beta}\sum_{u=1}^{d}\mathfrak{d}_{q,ji}^{ba,u}\cdot\gamma_{q,nm}^{u,a}+\theta\sum_{q=1}^{N_\beta}\sum_{u=1}^{d}\mathfrak{d}_{q,ji}^{ba,u}\cdot\epsilon_{q,nm}^{a,u}.\end{aligned} \tag{4.29}$$

The superscripts "a" and "b" are wild card characters which may denote either left or right states. To improve the assembly efficiency, $\mathfrak{d}_{q,ij}^{ab,u}$ and $\mathfrak{m}_{q,ij}^{ab}$ contain only contributions of shape functions which are not identically zero on $\mathcal{F}$. Note that one parametric matrix is stored for every possible relative orientation of two neighbouring elements.

The efficiency is further improved by taking into account the symmetries that exist between direct and transpose term. A closer look to equation (4.29) shows for instance that for the first term of the linearization, we have

$$\theta \left.\frac{\partial DI_{im}^{a}}{\partial \boldsymbol{u}_{jn}^{b}}\right|_{\text{term I}} = \left.\frac{\partial DT_{jn}^{b}}{\partial \boldsymbol{u}_{im}^{a}}\right|_{\text{term I}} . \tag{4.30}$$

Accordingly, these contributions have to be computed only once. Attention must however be paid to the system of RANS equations. Since we have partially decoupled the continuity equation from the Spalart-Allmaras turbulence model, see section 3.4, some blocks of the transpose term Jacobian are set to zero during the assembly of the (global) Jacobian matrix.

Penalty term and source term. In contrast to the laminar case, where a constant dynamic viscosity μ and thermal conductivity λ are assumed, the diffusivity, and hence the penalty parameter, increases with the eddy/turbulent viscosity μ_t. To keep the linearization of the penalty term DP exact, we have added a second contribution which takes into account the dependency of the penalty parameter on the S-A working variable. The linearization of the penalty term is processed together with the convective flux CI and the diffusive flux DI ("term II").

Moreover we have implemented the exact linearization of the S-A source term (SV). In contrast to [18] where only the production and the destruction terms are treated in an implicit manner, we also have included the diffusion (source) term in the linearization process. Furthermore, in addition to the derivative with respect to $\tilde{\mu}$, the Jacobian takes into account the dependency of the source term on the main flow variables ρ, ρu...

The assembly of the source term Jacobian slightly differs from the viscous DV and the inviscid CV volume terms described previously. Although we still use precomputed parametric matrices, the block size of the respective weights is smaller. Since for the test cases studied in this work only the turbulence model includes a (non-constant) source term, we prefer to add row vectors to the lines related to the S-A Jacobian, rather than large subblocks containing a lot of zeros.

4.3.1 Efficiency of the enhanced BILU preconditioner

The enhancement of the preconditioner resulting from the exact linearization of the governing equations is illustrated by our two standard test cases, namely

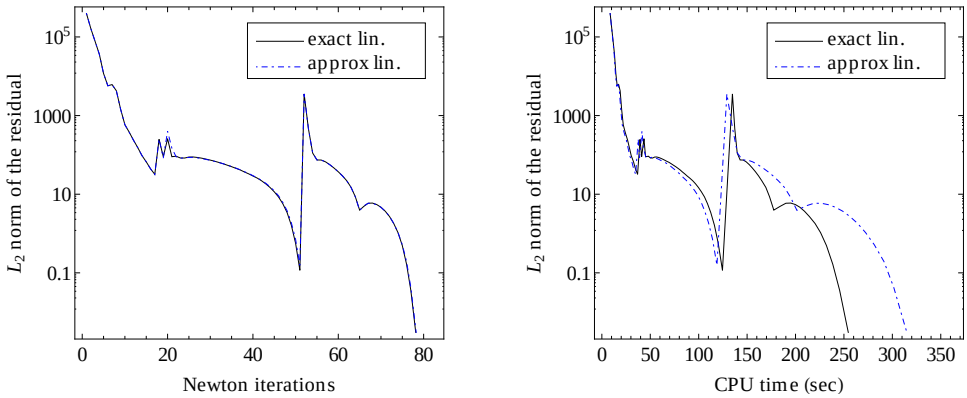
the turbulent flat plate ($Re_x = 10^7$ (at $x = 2$), $M_\infty = 0.2$) and the NACA 0012 aerofoil at zero incidence ($Re = 1.85 \times 10^6$, $M_\infty = 0.3$), cf. section 3.5.1. For the results presented here, turbulence modelling is based on the "clipped" version of the Spalart-Allmaras model. We found however that the same conclusions hold for the "modified" model. A BILU(1) algorithm is used as preconditioner and the linear system is solved with a precision of 10^{-1}.

In the case of small/moderate CFL numbers the quality of the preconditioner has no effect on the non-linear/Newton convergence; see. Fig. 4.1(a). In both cases, exact and approximate linearization, the same number of Newton iterations is needed by the computation. This observation was to be expected since (i) both approaches use a Jacobian-free GMRES solver resulting hence in the same linear system; (ii) the small time step size reduces the impact of the spatial discretization on the Jacobian matrix. Nevertheless, as a consequence of the smaller number of GMRES iterations the enhanced preconditioning reduces the CPU time by $\sim 20\%$, despite the additional cost due to the exact linearization, cf. "terms II" (Fig. 4.1(b)).

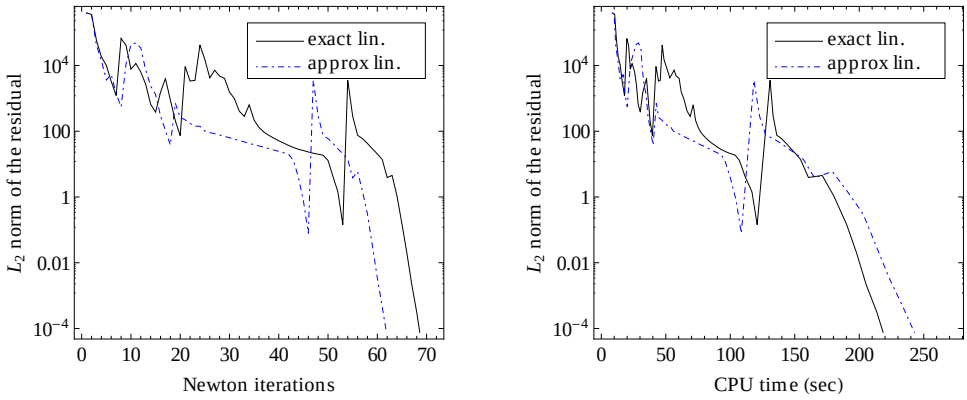
Next, we consider the impact of the linearization quality in the case of a more "aggressive" CFL strategy by increasing the CFL exponent α and the initial CFL number; see Fig. 4.1(c) and 4.1(d). To ensure the stability of the non-linear solver we furthermore limit the CFL number during the start-up. Although the convergence is less smooth than with smaller CFL values, the computation is faster. As an example the CPU time is reduced by $\sim 18\%$ (passing from 294 to 243 seconds, $\|\mathbf{R}\|_2 = 10^{-6}$) if the exact Jacobian is used.

An important difference to the previous case is that for large CFL numbers the preconditioner also affects the non-linear convergence; see Fig. 4.1(c). The reason is that the linear system is solved only approximately, hence resulting in different Newton iterates. Although (here) the total number of Newton iterations is even larger in case of the exact linearization, the improvement of the preconditioner is sufficient to reduce the total computing cost. These observations are confirmed by the flow along the flat plate; Fig. 4.2.

We furthermore observe that the impact of the quality of the preconditioner (exact/approximate linearizaton) increases with the polynomial order. While for 1-st order polynomials, the approximate linearization is even slightly faster, the computing time is more than trebled for 4-th order polynomials; cf. Fig. 4.2(b). This is in accordance with the fact that the linearization error committed by neglecting "term II" is proportional to the solution gradients. Because for the underresolved $p = 1$ computation, the gradients are rather small, taking into account the additional linearization terms does not significantly improve the quality of the preconditioner.



(a) Moderate CFL (b) Moderate CFL



(c) Large CFL (d) Large CFL

Figure 4.1: Because the linearization error is proportional to the gradients $\nabla \mathbf{U}$, the exact Jacobian significantly improves the quality of the BILU(1) preconditioner, resulting in a reduction of the computing time for turbulent flows. Here, NACA 0012 aerofoil at zero incidence, $Re = 1.85 \times 10^6$, $M_\infty = 0.3$, $p = 2$, Jacobian-free GMRES solver. The mesh is the same as in section 3.5.1; cf. Fig. 3.9.

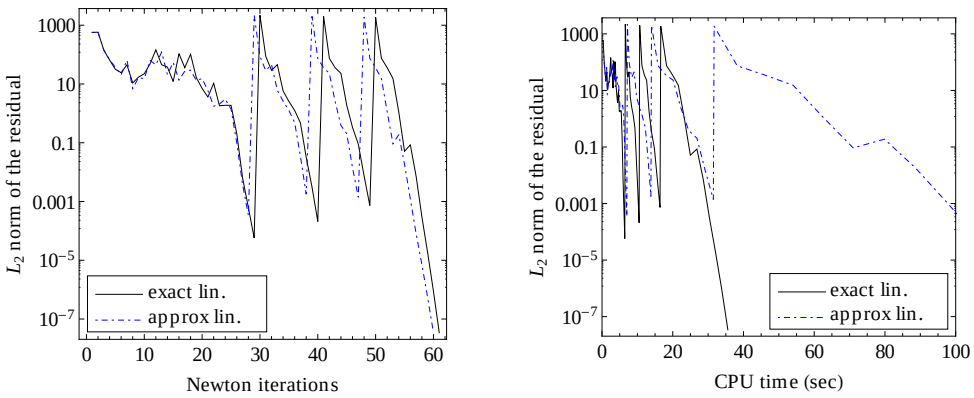


Figure 4.2: The impact of the linearization error increases with the polynomial order. Flow along a flat plate at $Re_x = 10^7$ (at $x = 2$) and $M_\infty = 0.2$, structured quadrangular mesh with $y^+(1) = 16$, $p = 4$, Jacobian-free GMRES solver.

Remarks

- *Single precision Jacobian.* To reduce the cost of the solver, the Jacobian matrix is stored in single precision. A straightforward calculation shows that especially for 3D applications the memory needed for the Jacobian exceeds by far the memory requirement of all the other data (solution vector,...). As a consequence storing the Jacobian in single precision not only significantly reduces the memory footprint of the method; it also should halve the CPU time to compute and inverse the Jacobian[9]. Moreover, since the Jacobian is only needed for preconditioning, a single precision approximation does not significantly affect the Newton-GMRES convergence [88].

- *Stale/frozen Jacobian.* Since the assembly and the inversion of the Jacobian can represent a significant computational cost attempts have been made to employ a "frozen" Jacobian calculated at a previous time level as preconditioner, e.g. [98, 99].

  The effect of a frozen Jacobian on the Newton-GMRES convergence is illustrated by two test cases: (i) a two-dimensional cascade flow[10] ($Re = 8.4 \times 10^5$ and $M_{inlet} = 0.68$), see Fig. 4.3; and (ii) a three-dimensional (extruded) NACA 0012 aerofoil. The cascade flow is initialized with $p = 3$ before projecting the solution on $p = 4$ polynomials. For the aerofoil, only one computational level has been used. The former is

[9] Single precision floating point operations are at least theoretically two times faster than double precision operations on modern CPUs.

[10] We have chosen the cascade flow because it is (somewhat) more complex than the NACA aerofoil without being too resources demanding which allows to easily test .

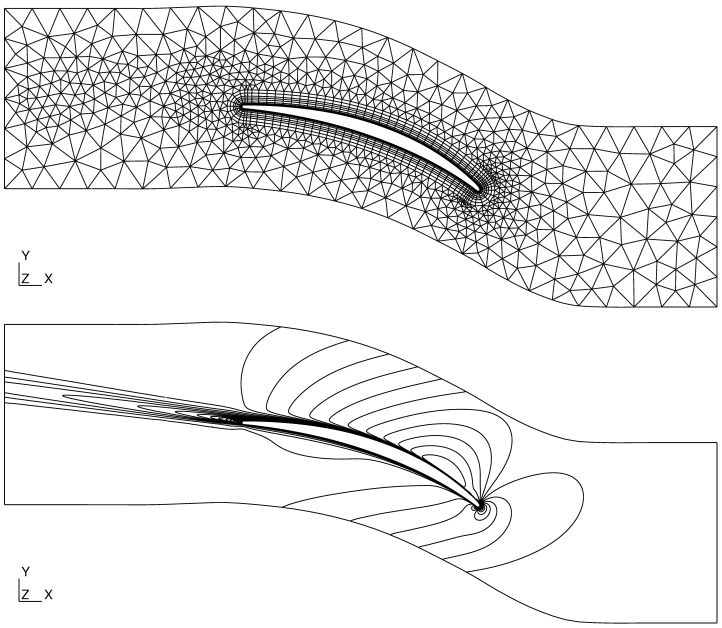


Figure 4.3: Two-dimensional cascade flow ($Re = 8.4 \times 10^5$ and $M_{inlet} = 0.68$): computational grid and iso-Mach contours ($p = 4$). The mesh is composed of 1293 triangles and 1183 (curved) quadrangles. Subsonic inflow boundary conditions ($T_t = 293.15K$, $p_t = 1.2 \times 10^5 Pa$, $\alpha = -37.67°$) are imposed on the right and the static pressure on the left ($p = 1.01 \times 10^5 Pa$). The flow is periodic in the y-direction.

discretized by 4 layers leading to a total of 8280 (curved) hexahedra and 6372 prisms; the computation is performed on 20 processors.

We observe that for both applications the CPU time can be reduced if the preconditioner is not computed at each Newton iteration, see Fig. 4.4 and 4.5, where the gain mainly depends on the relative cost of the preconditioner compared to the GMRES iterations.

Given the "moderate" memory footprint of the 2D cascade flow, a BILU preconditioner with different levels of fill-in has been tested. We found that for the present flow conditions, the Jacobian should be updated after 2 or 3 Newton steps, regardless of the level of fill-in. On the one hand, the increased number of GMRES iterations cancels out the savings related to the preconditioner. On the other hand, the computation systematically breaks down during the start-up if the preconditioner is frozen for more than 3 Newton steps. Fig. 4.4 also clearly demonstrates the advantage of a BILU preconditioner with a higher level of fill-in. Compared to the BILU(0) preconditioning the use of BILU(1) reduces the total computing time by $\sim 25\%$.

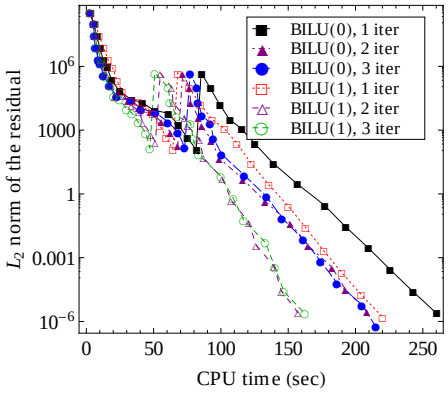


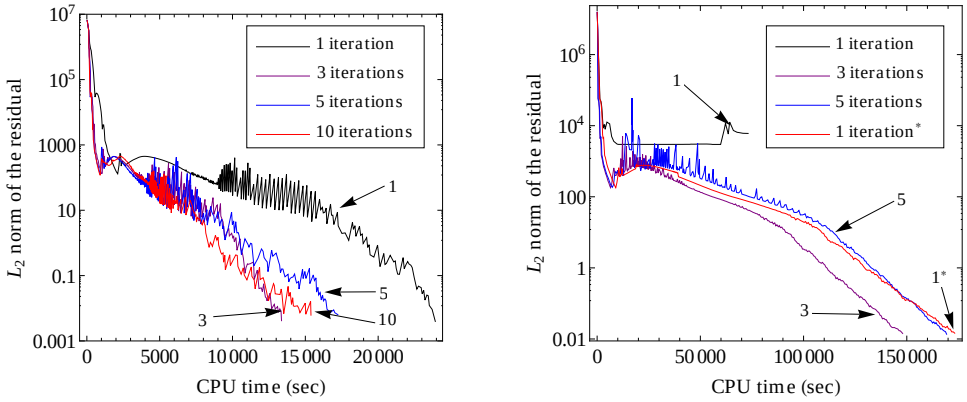
Figure 4.4: Two-dimensional cascade flow ($p = 4$). Significant savings of the CPU time can be achieved when the preconditioner is not evaluated at each Newton iteration. However, this approach may cause the breakdown of the Jacobian-free Newton-GMRES solver.

Computations of the NACA aerofoil have been performed for two different configurations: (i) $p = 2$ polynomials with a BILU(0) preconditioner; and (ii) $p = 3$ polynomials with a block-Jacobi preconditioner. Comparing figures 4.4 and 4.5(a) we observe that for the BILU(0) preconditioner the savings are even more important than for the two-dimensional application, i.e. the computing time can be halved by freezing the Jacobian. Although a similar situation is observed for P^3-elements, the resulting gain in CPU time is smaller, Fig. 4.5(b). One reason is the poor quality of the Jacobi preconditioner. Despite a reduced CFL number many GMRES iterations[11] are needed to solve the linear system which shrinks the relative cost of the preconditioner. Furthermore, because of the smaller CFL number, the solution and hence the Jacobian changes slowly during Newton steps. Accordingly, the linear solver is less affected by freezing the preconditioner[12].

Although there is no general rule for updating the preconditioner, a good choice seems, according to our numerical experiments, to evaluate the Jacobian every 3 to 4 Newton steps.

[11] In the present case ($CFL = 5 \times 10^4$, $p = 3$), 300 to 400 GMRES iterations are needed at the last stage of the computation, in order to reduce the residual norm of the linear system by one order of magnitude. For comparison, only 70 to 90 iterations are required in case of the BILU(0) preconditioner ($CFL = 10^5$, $p = 2$).

[12] Note that the computation diverged if the Jacobian is evaluated at each Newton iteration. Stable results could however be obtained by increasing the penalty parameter by a factor two.



(a) 3D NACA 0012: $p = 2$, BILU(0)

(b) 3D NACA 0012: $p = 3$, Jacobi

Figure 4.5: Whereas freezing the Jacobian can reduce the CPU time by up to $\sim 50\%$ in case of the BILU(0) preconditioner, similar savings as for the two-dimensional cascade flow are observed with the Jacobi preconditioner ($\sim 20\%$). The superscript '$*$' indicates that in order to avoid the breakdown of the computation, we had to double the penalty parameter.

4.4 Conclusions

Explicit and implicit time integration schemes have been introduced and the beasic features of the Generalized Minimum RESidual algorithm (GMRES) have been presented. As for any iterative Krylov method the performance of the GMRES solver strongly depends on the conditioning of the linear system. Although in this thesis a "matrix free" variant of the Newton-GMRES solver is used in order to reduce the rather large memory footprint of the DG method, at least an approximation of the Jacobian matrix is needed for preconditioning purposes.

Based on the work of Hillewaert *et al.* [88] we have implemented the exact linearization of the Navier-Stokes/RANS equations in a computationally efficient way separating shape functions and physical terms. Our numerical experiments have clearly shown a significant reduction of the total computing time for turbulent flows, if the preconditioner is based on the exact linearization. We found that the benefit increases with the polynomial order. As an example, for the flat plate with 4-th order polynomials the exact Jacobian leads to more than three times faster results.

Finally, the computing time can be reduced further by freezing the preconditioner during some Newton steps. Depending on the type of the preconditioner, the space dimension and the polynomial interpolation order, savings between 20 and 50 % have been observed. In our experience best performance is obtained if the Jacobian is evaluated every 3 to 4 Newton steps.

Chapter 5

Turbulent boundary layer resolution

Contents

This chapter focuses on the grid resolution that is needed to correctly compute a turbulent boundary layer. Based on grid convergence analyses in function of interpolation order and element type (triangle and quadrangle) on the one hand, and grid spacing and stretching on the other, clear guidelines on the choice for boundary layer resolution for practical applications are provided.

First straight boundaries are considered in section 5.1 by means of the turbulent flow along a flat plate with zero pressure gradient. The corresponding results have been submitted for publication in [52].

Next the influence of the position of the interpolation nodes on the accurate estimation of the shear friction is investigated in section 5.2.

Finally the effect of boundary curvature is studied for the flow around the NACA 0012 aerofoil in section 5.3.

5.1 Straight sided elements: the turbulent flat plate

In this section we perform a grid convergence study for the computation of a flat plate turbulent boundary layer, using different interpolation orders p and different types of straight sided elements. The Reynolds number at the center of the plate ($x = 1$) is $Re_x = 5 \times 10^6$ whilst the Mach number is given by $M = 0.2$. The plate leading edge is preceded by a symmetry plane, and constant pressure is imposed on all boundaries, except at the inlet, where free-stream conditions are imposed, in particular $\tilde{\mu}_\infty = 1/10\mu$.

In the following, both triangular and quadrangular grids are considered. Seven grids with different off-wall spacings and a stretching coefficient of $\alpha = 1.6$ have been generated (here, α represents the ratio between the heights of two adjacent elements). The off-wall spacing has been chosen in order to get a prescribed non-dimensional height $y^+(1) = y(1)u_\tau/\nu$ at x=1, where $y(1)$ denotes the (geometrical) height of the first element; see Table 5.1. The reference friction velocity $u_\tau \equiv \sqrt{\tau_w/\rho}$ is estimated by the Schultz-Grunow formula [155]

$$C_f \equiv \frac{\tau_w}{\frac{1}{2}\rho U_\infty^2} = 0.370(\log_{10} Re_x)^{-2.584}, \tag{5.1}$$

where C_f and τ_w denote the friction coefficient and the wall shear stress respectively.

Grid	y^+	nb elements	leading edge spacing
A	4	(5+16) × 15	$4.9\ 10^{-3} \times 2.2\ 10^{-5}$
B	8	(5+16) × 14	$4.9\ 10^{-3} \times 4.2\ 10^{-5}$
C	16	(5+16) × 12	$4.9\ 10^{-3} \times 8.7\ 10^{-5}$
D	32	(5+16) × 11	$4.9\ 10^{-3} \times 1.7\ 10^{-4}$
E	64	(5+16) × 10	$4.9\ 10^{-3} \times 3.5\ 10^{-4}$
F	96	(5+16) × 9	$4.9\ 10^{-3} \times 5.2\ 10^{-4}$
G	128	(5+16) × 8	$4.9\ 10^{-3} \times 6.9\ 10^{-4}$

Table 5.1: Grid specifications, (x_1+x_2) × y : with x_1 and x_2 the number of elements in x-direction upstream and on the plate respectively and y the number of elements in y-direction. The stretching coefficient for all grids is set to $\alpha = 1.6$.

The present section is divided into three parts. In the first part (section 5.1.1), we investigate the grid resolution needed for an accurate computation of the boundary layer as a function of the interpolation order p. Furthermore, we propose a correction of the computed wall friction which allows the use of considerable coarser meshes. In the second part (section 5.1.2), we compare the solution accuracy in terms of p for a given problem size[1]. Finally, the

[1]The total number of degrees of freedom is kept constant.

influence of a hybrid discretization, that combines a low-order interpolation of the S-A model with a high-order interpolation of the Navier-Stokes equations, is discussed in section 3.5.3.

5.1.1 Underresolution effects

An accurate prediction of the friction coefficient requires the resolution of strong gradients of the solution very close to the wall. For a standard finite volume method this implies that the first element must lie in the viscous sublayer ($0 \leq y^+ \leq 5$) with $y^+(1) \approx 1$ and that the grid normal stretching α is in the order of 1.05 to 1.15. These severe restrictions not only lead to an important computing cost but also increase the numerical stiffness.

In the following, we will see that high-order DG methods allow to correctly compute a turbulent boundary layer using grids with $y^+(1)$ up to 64 ($p = 4$). Hence, the first interpolation point is located 16 times farther from the wall than for standard low-order methods. A comparison of the results is performed in terms of wall friction as well as velocity and viscosity profiles.

Physical versus numerical friction

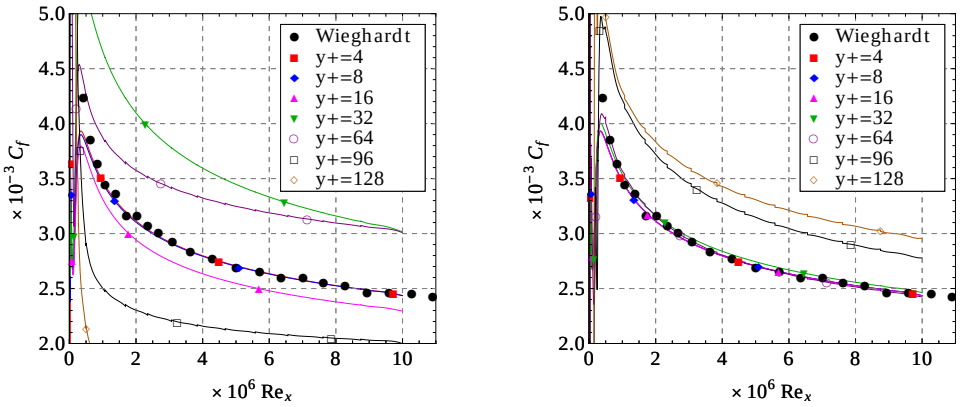
According to Fig. 5.1(a), a perfect agreement is observed between the experimental data of Wieghardt [181] and the calculated diffusive flux ($p = 4$, quadrangles) as long as $y^+(1) \leq 8$. This means that at least half of the element must lie within the viscous sublayer. As discussed before, one can consider that the 4^{th} order polynomial (or equivalently 5^{th} order accurate) interpolation on the quadrangle/hexahedron provides an equivalent resolution, corresponding to a subdivision of the element by 4 in all directions. Hence, the off-wall spacing of the first interpolation point remains comparable to the one used by 2^{nd} order accurate finite volume methods. For $y^+(1) = 16$, the friction coefficient is already underestimated by $\sim 4\%$.

However, a much better estimate of the skin friction can be obtained by considering the numerical flux $\hat{\boldsymbol{\mathcal{F}}}^v_{k,m} = \left\langle D^{kl}_{mn} \frac{\partial \mathbf{u}_{hn}}{\partial x_l} \right\rangle + \delta [\![\mathbf{u}_{hm}]\!]_k$ which is the sum of the *physical* or *consistent* diffusive flux $\boldsymbol{\mathcal{F}}^v$ and the penalty term $\delta [\![\mathbf{u}_h]\!]$, see Fig.5.1(b). A comparison of the physical flux, Fig. 5.1(a), and the numerical flux, Fig. 5.1(b), shows that

(i) the solution jump vanishes when the grid resolution is increased, and

(ii) the penalty term corrects the diffusive flux in such a way that the numerical flux is in equilibrium with the external flow even for large off-wall spacings.

Observation (i) was expected as it has been demonstrated by Cockburn [37] that the jump $[\![\mathbf{u}_h]\!]$ is directly related to the residual over the element integral T. Observation (ii) is a consequence of the conservativity of the present



(a) Physical flux: diffusive term

(b) Numerical flux: diffusive + penalty term

Figure 5.1: Skin friction for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$): comparison between the physical flux $\boldsymbol{\mathcal{F}}^v = \langle \mathbf{D}\nabla \mathbf{u}_h \rangle$ and the numerical flux $\hat{\boldsymbol{\mathcal{F}}}^v = \langle \mathbf{D}\nabla \mathbf{u}_h \rangle + \delta [\![\mathbf{u}_h]\!]$ for different structured quadrangular grids, h≡ distance to the opposing node. Taking into account the penalty term considerably improves the skin friction.

DG method. By definition a numerical flux $\hat{\boldsymbol{\psi}}(\mathbf{u}_h, \nabla \mathbf{u}_h)$ is conservative if it is single-valued on the element interfaces Γ. An important property of a conservative scheme is that for any union Ω^* of elements, the following relation holds

$$\int_{\Omega^*} \mathbf{S}\, dV - \oint_{\partial\Omega^*} \hat{\boldsymbol{\psi}} \mathbf{n}\, dS = 0, \tag{5.2}$$

where $\mathbf{S}$ is a source term. In particular ϕ_i can be chosen to be constant in the IP-DGFEM residual (3.60), so we get

$$\int_{\Omega} \mathbf{S}\, dV = \oint_{\partial\Omega} \boldsymbol{\mathcal{H}}\, dS + \oint_{\partial\Omega} \left(\mathbf{n} \cdot \langle \mathbf{D}\nabla \mathbf{u}_h \rangle + \delta [\![\mathbf{u}_h]\!] \right) dS \tag{5.3}$$

since $[\![\phi_i]\!]_{\phi_i=1} = 0$ except on $\partial\Omega$. From (5.2) and (5.3) it follows that the numerical diffusive flux is given by the sum of the diffusive and the penalty term. Note that a similar definition has been used by Hartmann *et al.* [81] within the context of adjoint-based error estimation, leading to the following expression for the enhanced target quantity $J(\mathbf{u}_h)$:

$$\tilde{J}(\mathbf{u}_h) = J(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}}_h). \tag{5.4}$$

Here, $\tilde{\mathbf{z}}_h$ is the discrete adjoint solution.

Influence of the penalty coefficient

In chapter 3, the influence of the penalty coefficient $\delta = C_{IP}\nu/h$ on the algorithm stability has been discussed. In particular, we have seen that the

new definition of the penalty length for anisotropic meshes (cf. "distance to the opposing node") improves the stability of the (modified) S-A turbulence model. However, the penalty parameter does not only affect the stability of the method but also might have an influence on the steady solution, especially on the skin friction. This will be examined in the present section.

First, we focus on the definition of the *characteristic length scale* h. Comparing the results for the different types of grids, it turned out that the choice of h has no effect on fine grids ($y^+(1) \leq 16$). This is not surprising since the solution jump vanishes in these cases. For large off-wall spacings, the (consistent) diffusive flux computed with h equal to the "distance to the opposing node" is slightly closer to the experimental results. However, the difference between the two approaches becomes almost insignificant if we consider the numerical flux; see Fig.5.2.

Next, in order to demonstrate the reliability of the numerical friction as well as its independence from the choice of the penalty coefficient, we progressively increase the *penalty constant* C_{IP}. As illustrated by Fig. 5.3(a), the higher the jump penalization, the closer the consistent contribution of the diffusive flux to the exact solution. The principal drawback of such an approach is the important increase of the condition number of the underlying system, which considerably deteriorates the convergence of the method. In contrast to the consistent flux, the numerical friction defined in section 5.1.1 is almost independent of C_{IP}; cf. Fig. 5.3(b). Finally, important oscillations are observed, when the penalty constant approaches the stability limit. This is the case if the C_{IP} is divided by a factor ~ 2.

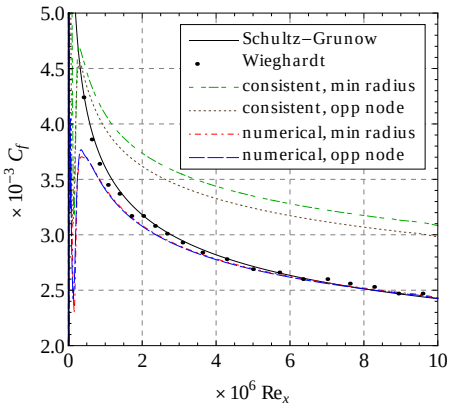
Influence of the element type on the numerical friction

Fig. 5.4 compares the friction coefficient of the turbulent flat plate obtained with triangular and quadrangular grids for different off-wall spacings and interpolation orders. For clarity of the figures only friction coefficients with $y^+(1) \leq 32$ are presented for triangles. Probably the most important difference between triangles and quadrangles is that the latter lead to a much smoother skin friction, whereas the use of triangles causes oscillations which rapidly grow if the grid is coarsened. Separating out the effects of the different fluxes shows that these oscillations are mainly due to the penalty term.

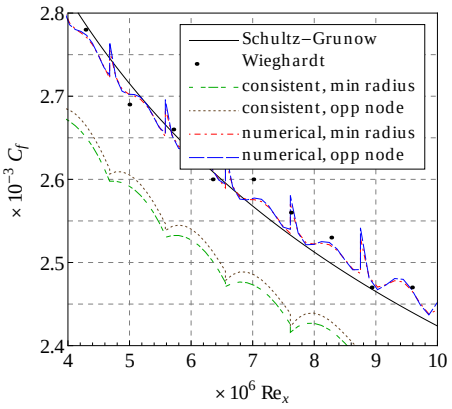
In particular, comparing the results obtained with first order elements ($p = 1$), we notice that the jumps of C_f along the plate almost disappear for quadrangles. The reasons for this different behaviour are

(i) Since C_f is proportional to the gradient normal to the wall $\mu\ \partial u/\partial y$, quadrangles will always provide a skin friction which is one order above the one provided by triangles;

(ii) While neighbouring quadrangles along the plate typically share a common face, adjacent triangles on the plate usually only have one node in

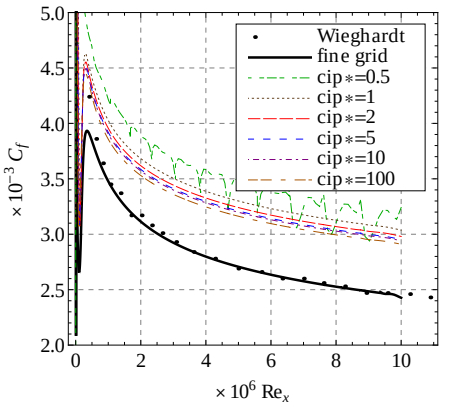


(a) Whole plate: $y^+(1)$=64, quadrangles

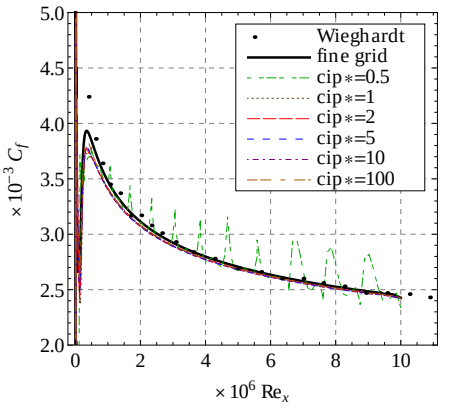


(b) Trailing edge: $y^+(1)$=16, triangles

Figure 5.2: Influence of the type of the penalty term $\delta = C_{IP}\,\nu/h$ on the skin friction along the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) with 4^{th} order polynomials. Left, structured quadrangular grid ($y^+(1)$=64, ref. "grid E"); right, structured triangular grid ($y^+(1)$=16, ref. "grid C"). Choosing the "distance to the opposing node" slightly reduces the impact of the penalty term on the numerical flux.



(a) Consistent friction: $y^+(1)$=64, p=4



(b) Numerical friction: $y^+(1)$=64, p=4

Figure 5.3: Influence of the penalty constant C_{IP} on the skin friction along the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) with 4^{th} order polynomials ($y^+(1)$=64, ref. "grid E"). Left, consistent friction; right, numerical friction. Increasing the jump penalization increases the contribution of the consistent diffusive flux but has virtually no influence on the computed numerical friction.

common. As a consequence, they are not considered to be neighbours and jumps between them are not penalized.

(iii) the penalty term is proportional to the solution, and hence is much richer in content than the physical skin friction.

One of the consequences is that in the case of quadrangles larger off-wall spacings can be used.

Velocity profiles

The non-dimensional velocity profiles $u^+(y^+)$ are shown in Fig. 5.5. When analyzing these results, we should first of all consider that boundary conditions are imposed weakly. This results in the non-compliance to the no-slip condition for underresolved computations. Furthermore, the reference velocity u_τ is based on the *computed* skin friction and hence the non-dimensional free-stream velocity depends on the resolution.

As a function of resolution, we can clearly see that at equal number of degrees of freedom, the solution drastically improves with the interpolation order. The mesh resolutions leading to a good correlation with the experiments furthermore clearly correspond to the ones based on the *numerical* friction as discussed in the previous section. This means that although it contains an important jump contribution, one can probably rely on the numerical value, and a very strict compliance with the no-slip boundary condition is not required.

For underresolved computations we furthermore clearly see an increase of the boundary layer thickness, which at first sight may be attributed to the underresolution, but also to a larger spread of the eddy viscosity as illustrated in following section.

Non-dimensional viscosity profiles

The non-dimensional viscosity profiles $\bar{\mu}(y^+) = \tilde{\mu}(y^+)/(100\mu)$ at section $x = 0.97$ are presented in Fig. 5.6.

We observe that independently of the interpolation order the element size has no effect on the behaviour of $\bar{\mu}$ *close to the wall*. The reason is that the working variable $\tilde{\mu}$ of the Spalart-Allmaras model has been designed to vary almost linearly from the wall to the log-layer. The slope in this region is approximately equal to the von Karman constant $\kappa = 0.41$.

In contrast, the accuracy of the viscosity profile *in the log layer* is strongly improved by high-order elements. Even for grids with a relative small off-wall spacing, imposing a constant grid stretching of about 1.6 will inevitably lead to large elements in the log-layer, which means that the viscosity peak is discretized by a few elements only. An illustration of the element size through the boundary layer for a quadrangular grid with $y^+(1) = 64$ ("grid E") is given

in Fig. 3.23. The dotted vertical lines represent the mesh, the grid stretching in the wall normal direction equals 1.6. We find that, although this mesh is extremely coarse (the viscosity peak is mainly discretized by three elements), good results are obtained with 4^{th} order polynomials, whereas the problem is clearly underresolved for $p = 1$ or $p = 2$.

Influence of the grid normal stretching

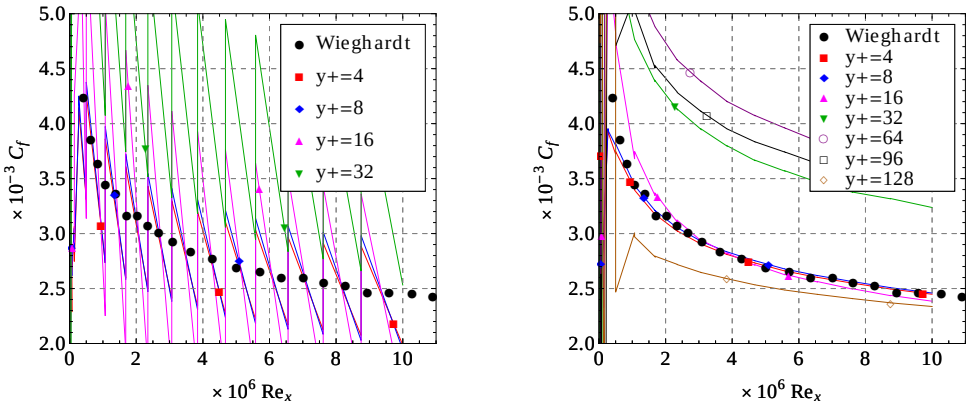
So far, all meshes have been generated with a grid normal stretching of $\alpha = 1.6$. Note that for standard finite volume methods, α commonly is in the order of 1.05 to 1.15. Although it is likely that higher-order methods allow higher grid stretchings, the value of 1.6 has been chosen somewhat randomly. The influence of α on the grid resolution is the subject of the present section.

First, we try to single out if there is a relation between the grid normal stretching and the maximum allowed off-wall spacing. Thereto, we compare the results obtained with 2^{nd} order polynomials and an off-wall spacing of $y^+(1) = 32$ for different stretching coefficients between 1.1 and 2. As shown in the previous sections, this computation is clearly underresolved for P^2-elements and a stretching of 1.6. The corresponding skin friction distributions along the plate are displayed in Fig. 5.7(a). According to these results, the grid normal stretching has only a minor effect on the skin friction. In particular, reducing the stretching does not improve the computed friction, which mainly depends on the height of the first cell close to the wall; see Fig. 5.7(b).

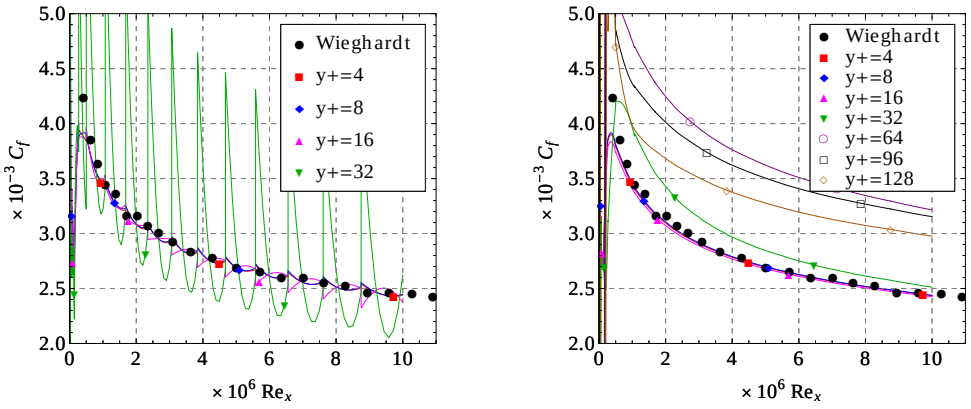
Next, we want to determine the maximum grid stretching for a given off-wall spacing. Thereto, we have chosen P^4-elements and an off-wall spacing of $y^+(1) = 64$, which approximately corresponds to the minimum required mesh resolution for this interpolation order. The stretching coefficient varies between 1.1 and 2. Again, as illustrated by Fig. 5.7(c) and 5.7(d), there is virtually no impact on the friction coefficient C_f. These observations are confirmed by the velocity profiles (see Fig. 5.8), which in the case of $\alpha = 2$, present a small acceleration at the outer boundary of only $\sim 0.2\%$ compared to the freestream velocity.

5.1.2 Convergence analyses with equal number of degrees of freedom

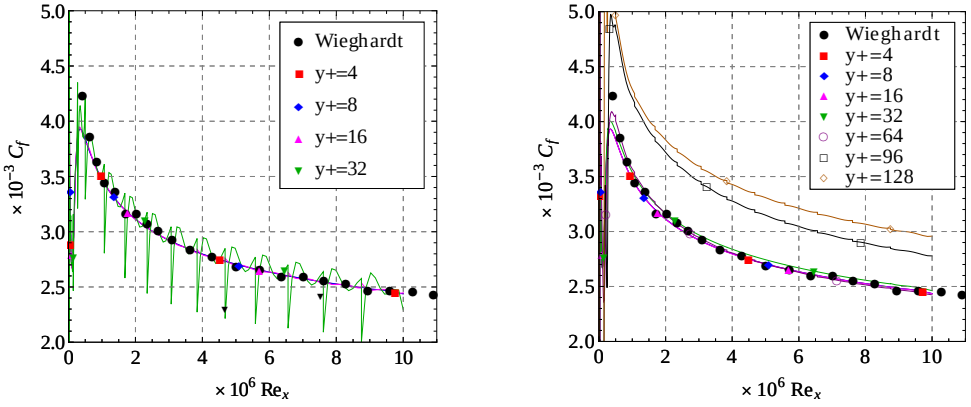
In the preceding section we have compared the velocity/viscosity profiles and the skin friction coefficient for a large number of grids. The main focus was (i) to demonstrate the improvement of the estimation of the skin friction by the penalty term and (ii) to determine the maximal off-wall spacing in function of the interpolation order p and the element type. Therefore, the same grids were used for P^1-, P^2- and P^4-elements. Now, we want to compare interpolation orders in terms of accuracy for an equal number of degrees of freedom, and evaluate the mesh convergence for a given order.



(a) triangle: p=1 (b) quadrangle: p=1

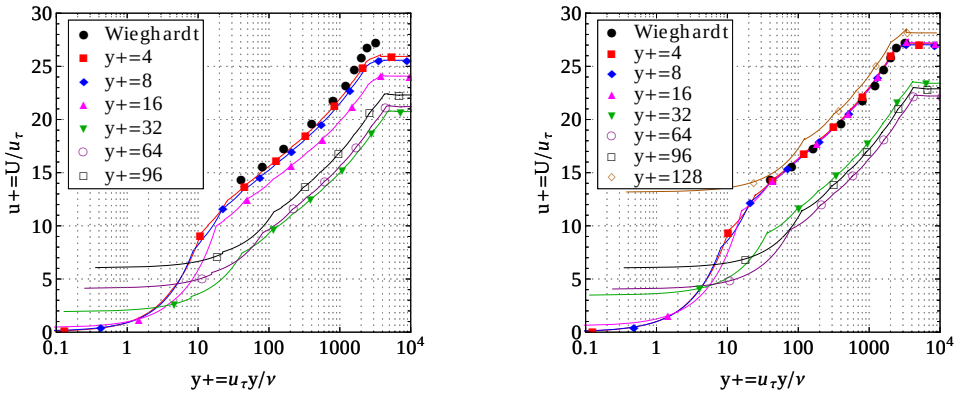


(c) triangle: p=2 (d) quadrangle: p=2



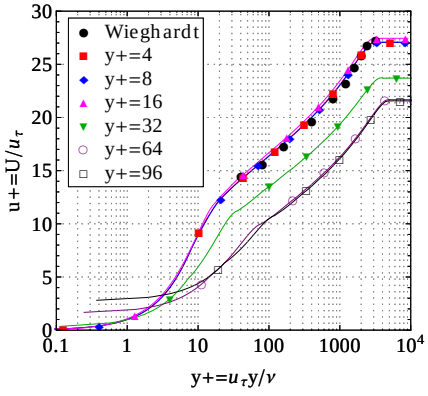
(e) triangle: p=4 (f) quadrangle: p=4

Figure 5.4: Skin friction based on the numerical flux $\hat{\boldsymbol{\mathcal{F}}}^v$ for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$): comparison between triangles and quadrangles, h≡ distance to the opposing node. Regardless of the interpolation order, quadrangular grids lead to smoother skin friction than the equivalent triangular grids. As a consequence, off-wall spacings, which are 2 to 3 times larger, can be used.

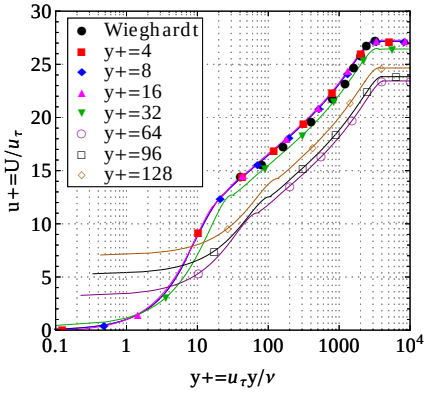


(a) triangle: p=1

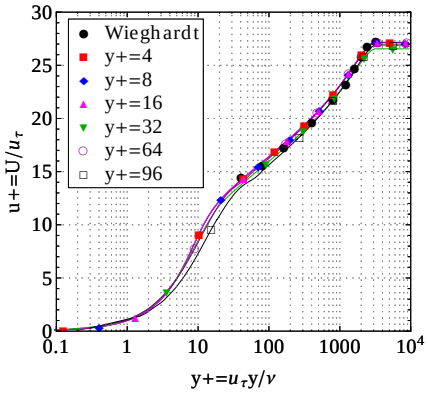
(b) quadrangle: p=1



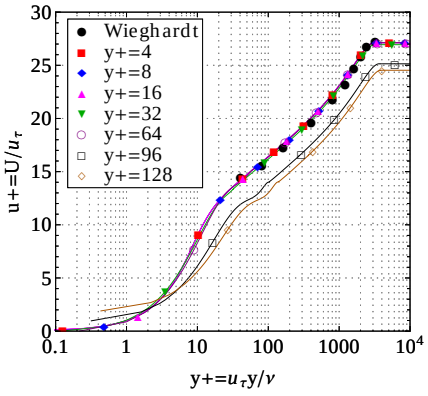
(c) triangle: p=2



(d) quadrangle: p=2

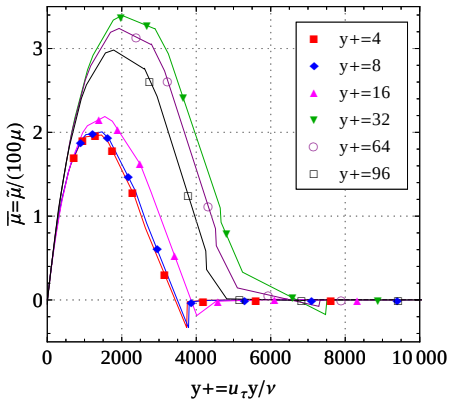


(e) triangle: p=4

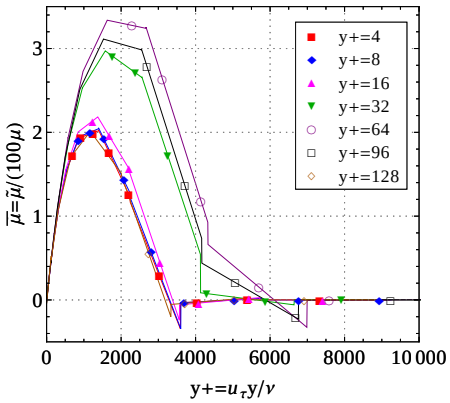


(f) quadrangle: p=4

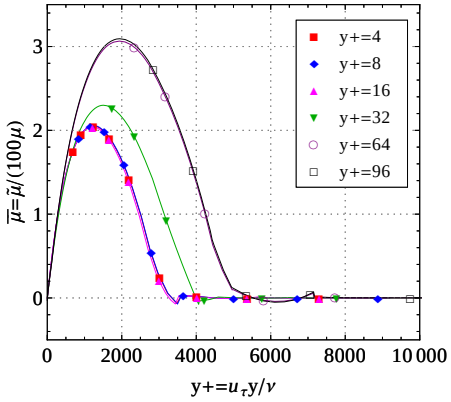
Figure 5.5: Normalized velocity profile for the flat plate ($Re_x = 5\times10^6$ at x=1, $M_\infty = 0.2$): comparison between triangles and quadrangles. The skin friction used to normalize the velocity is based on the numerical flux, h≡ distance to the opposing node. As boundary conditions are imposed weakly, the no-slip condition is not satisfied by all the grids. This is particularly true for coarse quadrangular grids.



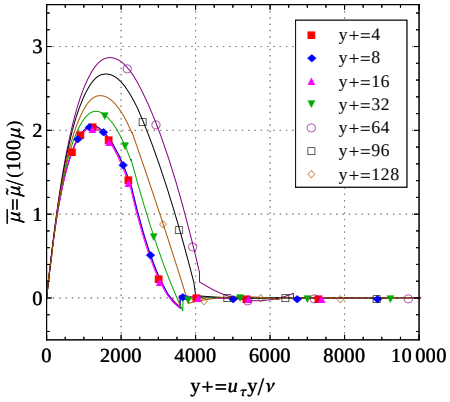
(a) triangle: p=1



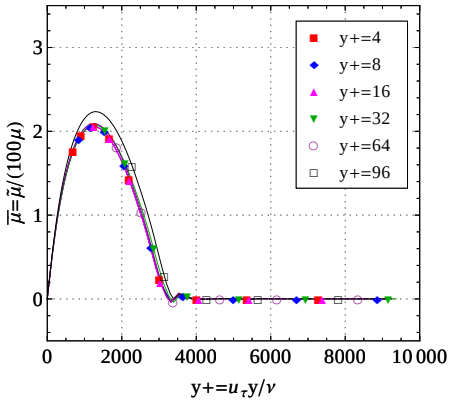
(b) quadrangle: p=1



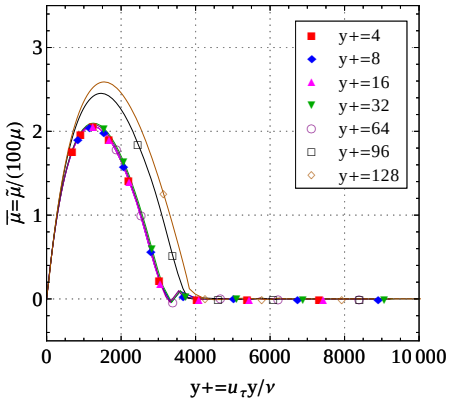
(c) triangle: p=2



(d) quadrangle: p=2

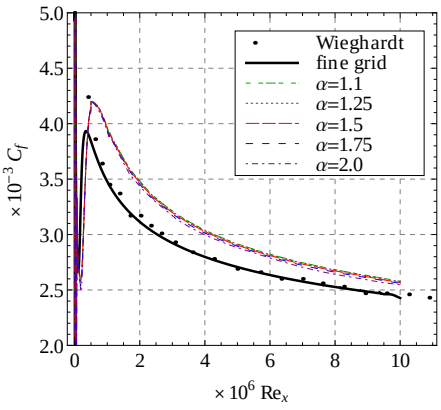


(e) triangle: p=4

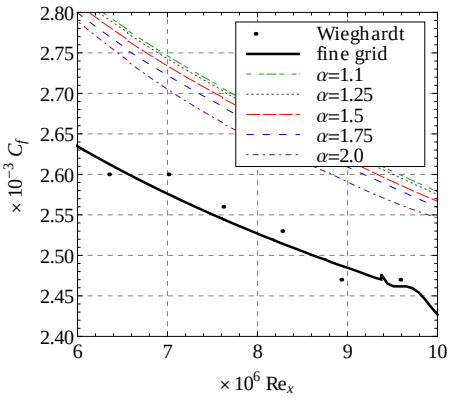


(f) quadrangle: p=4

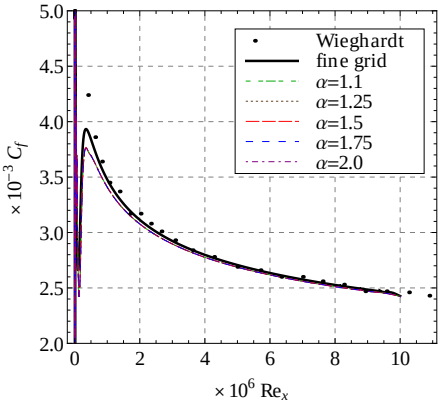
Figure 5.6: Non-dimensional viscosity profile $\bar{\mu} = \mu_t/(100\mu)$ at section x=0.97 for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$): comparison between triangles and quadrangles.



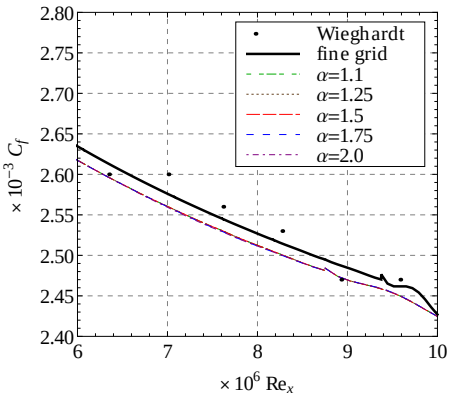
(a) $y^+(1)$=32, p=2



(b) $y^+(1)$=32, p=2 (zoom)



(c) $y^+(1)$=64, p=4



(d) $y^+(1)$=64, p=4 (zoom)

Figure 5.7: Influence of the normal grid stretching α on the skin friction along the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) with quadrangular grids.

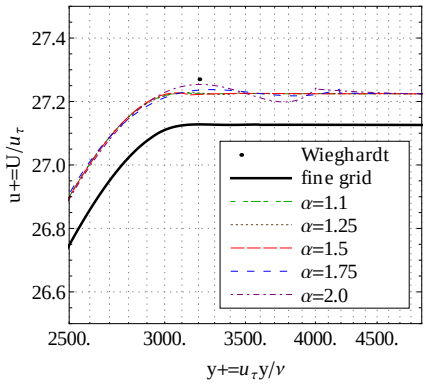


Figure 5.8: Influence of the normal grid stretching α on the skin friction along the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) with quadrangular grids.

To both ends, each grid of table 5.1 is refined two times by dividing every element face in half. This leads to a sequence of grids suitable for convergence studies. Henceforth, we call "level 1" the non-refined grid, "level 2" the same grid refined once and "level 3" the grid refined two times.

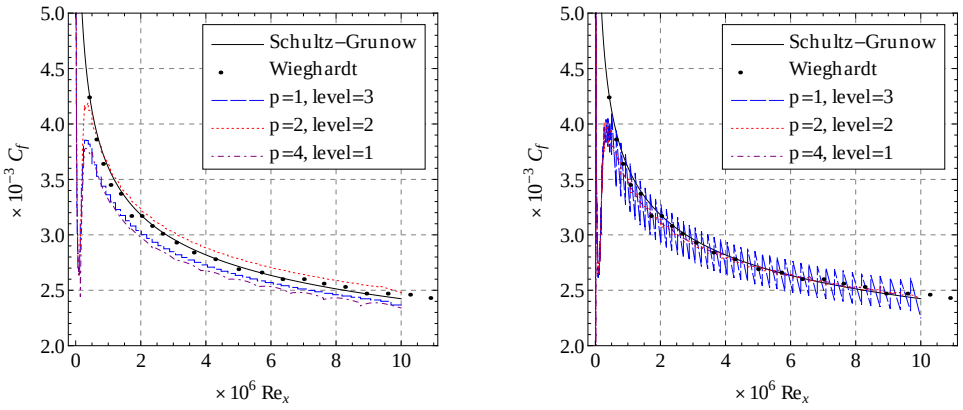
Convergence in function of interpolation orders

In order to compare different orders, whilst keeping computational cost constant (in terms of degrees of freedom), the turbulent plate is computed on refinement "level 1" for $p = 4$, "level 2" for $p = 2$ and "level 3" for $p = 1$ respectively.

Triangles. In the case of grids A to C ($y^+(1) \leq 16$ at level 1) an excellent agreement is observed between the different interpolation orders on the one hand, and the experimental/theoretical results of Wieghardt/Van Driest (resp. Schultz-Grunow) on the other hand. This is particularly true for the normalized velocity profile $u^+(y^+)$ (not represented here), whereas the influence of the interpolation order is more important on the skin friction C_f. Regarding the (physical) diffusive flux only (Fig. 5.9(a)), all three curves slightly differ from the experimental measurements. Notably P^1-elements lead to less smooth results. This is due to the fact that in contrast to the velocity profile, the skin friction does not only depend on the solution variables but also on their gradients which are constant for P^1-elements. Taking into account the penalty term (Fig. 5.9(b)) improves the results for p=2 and p=4 but causes important oscillations for p=1, reflecting the linear variation associated to the friction.

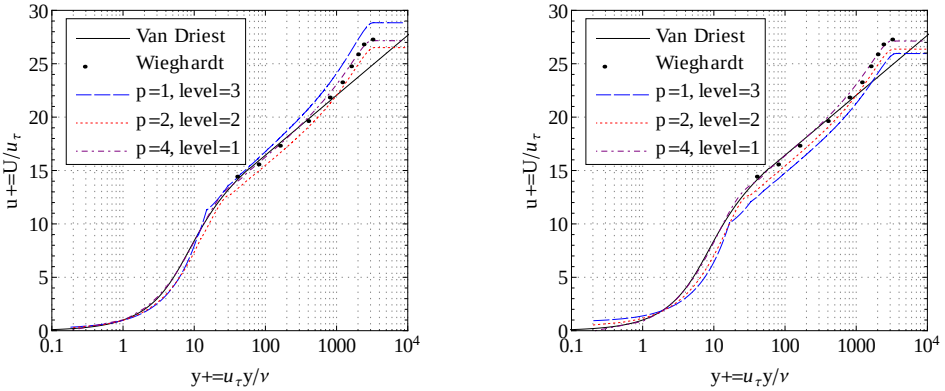
A further increase of the off-wall spacing ($y^+(1) \leq 64$) hardly affects the velocity profile but blows up the oscillations of the friction coefficient. This explains the different values of u^+ far from the wall.

Quadrangular grids. As already observed before, the main difference between triangles and quadrangles is that the latter lead to a smoother skin friction. Regarding the diffusive term we notice that the element type particularly affects P^1-results which no longer behave stepwise. But also the oscillations of the penalty term which constitute the principal limitation for a further increase of the off-wall spacing vanish (compare Fig. 5.9(b) and 5.11(a)). The superiority of high order polynomials is illustrated by Fig. 5.11(b) and Fig. 5.10(b). A comparison with Fig. 5.10(a) reveals however that coarse grids triangles imply a better respect of the no-slip boundary condition. An additional argument for the use of quadrangles is the memory consumption. As nodes are doubled at the interface between adjacent elements, the ratio of the total number of degrees of freedom between triangles and quadrangles is given by $(p+2)/(p+1)$. This ratio tends to 1 as the interpolation order is increased. Furthermore, taking into account that flop rates of the matrix-vector products should increase



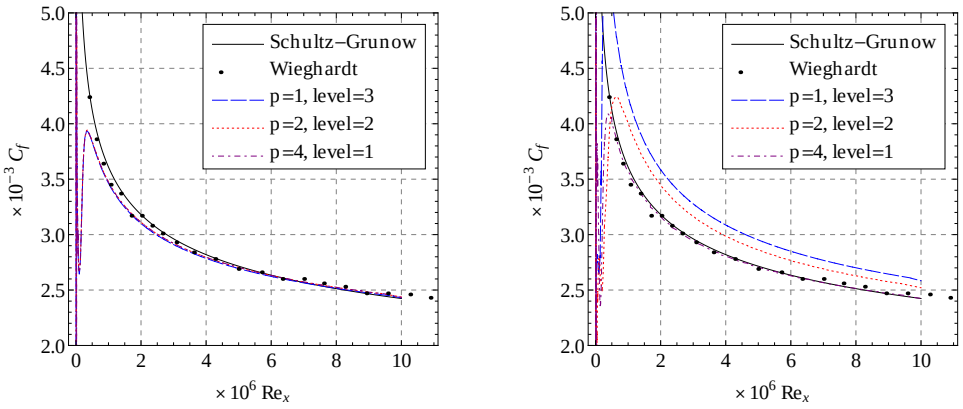
(a) Consistent flux: $y^+(1)$=16 at x=1 (b) Numerical flux: $y^+(1)$=16 at x=1

Figure 5.9: Skin friction for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) with structured triangles (grid C): comparison between physical and numerical flux, h≡ distance to the opposing node. Low order triangles suffer from oscillations of the penalty term. These disappear in the case of quadrangular grids or higher polynomials.



(a) Triangle (grid E: $y^+(1)$=64 at x=1) (b) Quadrangle (Grid E: $y^+(1)$=64 at x=1)

Figure 5.10: Normalized velocity profile for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) with structured triangular grids. The skin friction used to normalize the velocity is based on the numerical flux, h≡ distance to the opposing node.



(a) Grid C: $y^+(1)$=16 at x=1

(b) Grid E: $y^+(1)$=64 at x=1

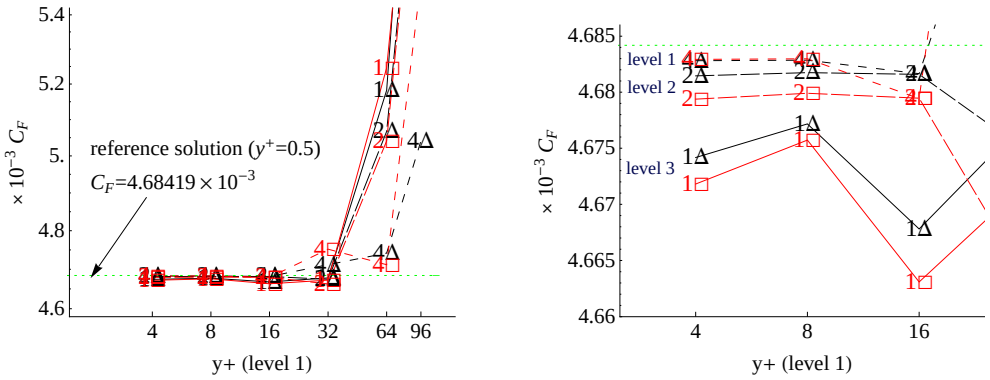
Figure 5.11: Skin friction (based on the numerical flux) for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$) with structured quadrangles, h≡ distance to the opposing node. Especially for coarse grids, better results are obtained by increasing the interpolation order rather than the number of elements.

with the number of nodes per element, quadrangles are expected to give faster results.

Convergence in function of mesh resolution

Finally, a convergence study based on the total skin friction (diffusive flux + penalty term) is realized; see Fig. 5.12. In order to avoid the singularity at the leading edge of the plate as well as boundary condition effects, the skin friction is integrated for $0.1 \leq x \leq 1.75$. The reference solution is composed of (40+128) × 160 triangles ($p = 4$). Its accuracy is improved using a Richardson extrapolation. Note that in the absence of the exact solution, only the convergence of the skin friction to the asymptotic value is presented here.

The advantage of high order methods becomes obvious when comparing the friction error for different interpolation orders. Except in the case of very coarse grids with a large off-wall spacing, high order elements on level i give better results than lower order elements on level $i + 1$ and that, as already mentioned, for a lower memory cost. in the case of 1^{st} and 2^{nd} order elements the formal convergence order is achieved for all the grids (not shown here). Remember that the order of accuracy of a scheme is guaranteed only at the limit of grid refinement. This is also true for 4^{th} order elements on coarse grids (for instance grid D where $y^+(1)$=32). As for rather fine grids, the convergence rate is underestimated. A likely reason is that the reference solution is not fine enough and the error on C_F becomes comparable to the error on C_F^{Ref}.



(a) all grids (b) zoom

Figure 5.12: Integrated skin friction C_F with $0.1 \leq x \leq 1.75$ for the flat plate ($Re_x = 5 \times 10^6$ at x=1, $M_\infty = 0.2$), h≡ distance to the opposing node. The reference solution is computed with (40+128) × 160 triangles ($p = 4$), which corresponds to $y^+(1) = 0.5$. In order to get comparative results, the friction cofficients are calculated on refinement "level 1" (not refined) for p=4, "level 2" (refined once) for p=2 and "level 3" (refined twice) for p=1 respectively. △, structured triangles; □, structured quadrangles;· · · , reference solution.

5.2 Quasi-straight elements

In the previous section, we have determined the required mesh resolution as a function of the polynomial degree ($p \leq 4$) in order to accurately compute a turbulent boundary layer. Given the simple geometry of the flat plate, the simulations have been carried out with straight sided elements and a linear mapping has been defined between the parametric and the physical function spaces. As in this work, the interpolation points are equally spaced in the canonical/reference element, a linear mapping also implies an equidistant distribution of the interpolation points in the physical space.

However, the question arises if for a given problem size, one cannot improve the accuracy of the computed results by clustering the interpolation points in the vicinity of the wall, where the strongest gradients are observed. Note that such an approach does not affect the formal order of accuracy of the DG method, which only depends on p. The optimal *position of the interpolation points* and its influence on the wall friction will be investigated in the present section.

5.2.1 Position of the interpolation points

A simple but efficient way to move the interpolation points closer to the wall, which does not require to modify the existing shape functions ϕ_i, consists in stretching the computing cell in the wall normal direction. This can be

achieved by using a non-linear geometrical representation of the cell ($q \geq 2$) and by applying a stretching α_i to the internal (high-order) nodes, in the same way as the inter-element stretching α defined in the previous section; cf. section 5.1.1. In the remainder of this thesis, we distinguish between the interpolation order p and the order of the geometrical representation (the mapping) q.

To illustrate this concept, we consider the second order mapping of the physical edge $x \in [0, L]$ to the segment $\xi \in [-1, 1]$, which is the reference element in 1D, i.e.

$$x(\xi) = c_0 + c_1\xi + c_2\xi^2. \tag{5.5}$$

The constants c_0, c_1 and c_2 are determined by imposing the desired position of the interpolation points in the physical space,

$$\begin{aligned} x(-1) &= x_0 = 0, \\ x(0) &= x_1 = L/(1+\alpha_i), \\ x(1) &= x_2 = L, \end{aligned} \tag{5.6}$$

leading to the final expression of the mapping

$$x(\xi) = \frac{L}{1+\alpha_i} + \frac{L}{2}\xi + \frac{L(\alpha_i - 1)}{2(1+\alpha_i)}\xi^2. \tag{5.7}$$

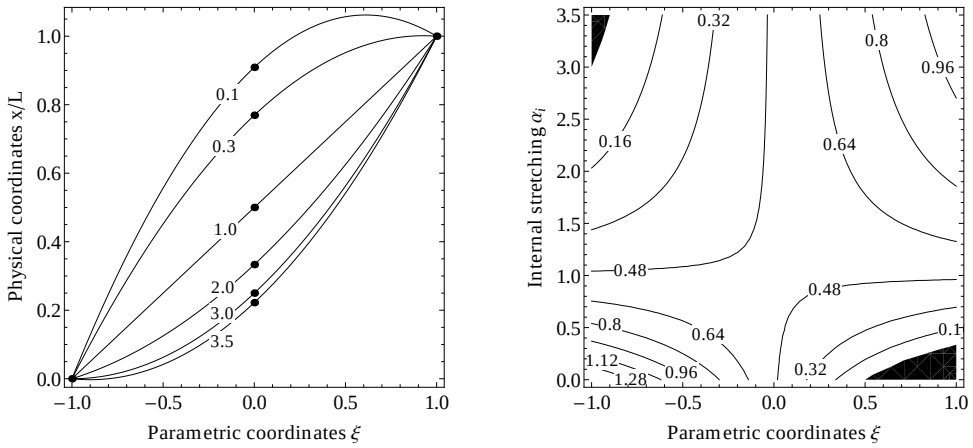
It is easily seen that with the above conditions (5.6), the following relation holds $(x_2 - x_1)/(x_1 - x_0) = \alpha_i$. Furthermore, setting α_i equal to 1 results in the standard linear element (with equally spaced interpolation points), whereas x_1 approaches the wall for $\alpha_i > 1$; see Fig. 5.13(a). The wall is supposed to be located at $x = 0$.

While increasing α_i moves the first interpolation point closer to the wall, we must ensure that (5.7) still defines a unique mapping between x and ξ. Thereto, α_i must be chosen such that

$$J^{-1} \equiv \partial x/\partial\xi \neq 0 \quad \forall\xi \in [-1, 1]. \tag{5.8}$$

In the case of the above second order mapping, a straightforward computation shows that the internal stretching must lie in the range of $\alpha_i \in [0.3, 3]$; see Fig. 5.13(b). Black regions indicate a negative Jacobian and thus an invalid mapping. Figure 5.13(b) also demonstrates that by stretching the computing cell, an invalid mapping first appears at the ends of the segment ($\xi = -1$ if $\alpha_i > 1$). Similarly, a numerical analysis leads to a maximal allowable stretching of $\alpha_i^{max} \approx 4.88$ ($q = 3$) and $\alpha_i^{max} \approx 2.62$ ($q = 4$).

Finally, as the first interpolation point in the reference space is located at $\xi_1 = -1 + 2/p$ and by choosing $\alpha_i = 3$ in equation (5.7), one finds that x_1 can not lie closer than L/p^2 off from the wall; compared to $x_1 = L/p$ in the case of straight (linear) elements. As a consequence, if one wants to keep the position of the first node unchanged, the element height can at most be coarsened by



(a) $x(\xi)$ for $\alpha_i \in [0.1, 3.5]$ (b) Inverse of the mapping Jacobian J^{-1}

Figure 5.13: In order to ensure a unique second order mapping ($q = 2$) between the physical and the parametric spaces, the internal stretching α_i must lie in the range of $[0.3, 3]$. Black regions indicate an invalid mapping. '•' denotes the interpolation nodes.

a factor p. Bearing in mind that an off-wall spacing of $y^+(1) \approx 64$ has been found for straight-sided P^4-elements, a theoretical mesh resolution of at best $y^+(1) \leq 256$ is to be expected.

5.2.2 Convergence analyses

The effect of the position of the interpolation points on the wall friction and in particular the consequences on the mesh resolution are studied by means of the flow along the flat plate (cf. section 5.1). To facilitate comparison with linear elements ($q = 1$), we use the same meshes as in the previous section, i.e. the same number of grid points in $x-$ and $y-$direction, and an element stretching of $\alpha = 1.6$. The analysis is limited to quadrilateral elements.

Stretching approaches. Before looking into details, which off-wall spacing is needed, different stretching approaches are compared, namely

(i) *uniform stretching*: a stretching of $\alpha_i = \alpha$ is applied to the internal nodes of all layers of elements;

(ii) *first layer stretching*: only the first layer of elements next to the wall is stretched imposing $\alpha_i = \alpha$;

(iii) *equidistant stretching*: since imposing $\alpha_i = \alpha$ inevitably leads to a situation, where the distance between the last internal nodes x^n_{M-1} and x^n_M at layer n is larger than the distance between the first internal nodes

x_0^{n+1} and x_1^{n+1} at the next layer $n+1$, we compute $\alpha_i^* < \alpha$ such that $x_M^n - x_{M-1}^n = x_1^{n+1} - x_0^{n+1}$.

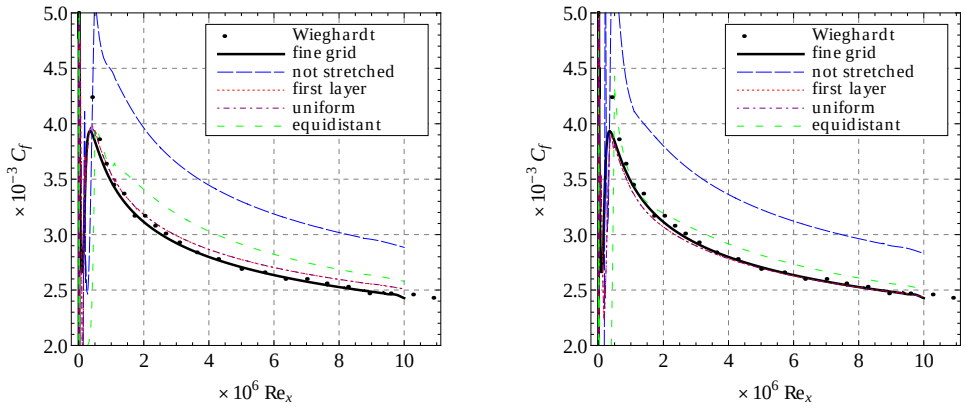
To investigate the influence of the above approaches, we have carried out a large number of simulations including several mesh resolutions $y^+(1)$ as well as degrees of polynomial and geometrical interpolation $2 \leq p, q \leq 4$. Furthermore, to exclude numerical integration errors, results have been compared for different rules of Gaussian quadrature ranging from $\mathcal{O}(3p)$ to $\mathcal{O}(5p)$.

As already observed in section 5.1.1, where we have analysed the influence of the grid normal stretching, it results from the present study that the friction coefficient mainly depends on the height of the first grid cell, and in particular on the position of the first interpolation point x_1 off the wall. Two representative examples are shown in Fig. 5.14. Both cases clearly demonstrate the potential/interest of approaching the interpolation points to the wall, leading to a significant improvement of the skin friction. The dependency on the position of the interpolation points becomes even more obvious if we compare the three strategies. Thereto, we should remember that the "uniform" stretching and the "first layer stretching" both lead to the same location of x_1. Since the two approaches result in (exactly) the same friction, whereas slightly distinct values are observed for the "equidistant" stretching ($\alpha_i^* < \alpha_i \Longleftrightarrow x_1^* > x_1$), we conclude that the difference is related to the location of x_1. As a consequence, in the remainder of this section, only the first layer next to the wall will be stretched because a displacement of the interpolation points in the farfield would require a higher Gaussian quadrature rule while no improvement of the accuracy is to be expected in this region.

Influence of the internal stretching on the off-wall spacing. To analyse the impact of the high-order nodes stretching, α_i has been adapted to the element height such that for enlarging computing cells, the first interpolation point x_1 remains at a fixed distance to the wall. In particular, based on the results for straight sided elements (cf. section 5.1.1), we have chosen $y^+(x_1) = 8$ for $p = 2$, $y^+(x_1) = 12$ for $p = 3$ and $y^+(x_1) = 16$ for $p = 4$.

It follows from the numerical experiments, that no clear improvement of the wall friction is observed if a *low-order polynomial interpolation* ($p \leq 2$) is used. More precisely, compared to the non-stretched case, the computed friction is in somewhat better agreement with the reference solution as long as the stretching remains small ($\alpha_i \lesssim 1.4$); although this eventually implies that $y^+(x_1) > 8$. However, as the stretching becomes important (in order to respect the desired location of the interpolation point), the accuracy of the computed friction rapidly deteriorates. In particular, for $y^+(1) = 24$, even better results are obtained by (simple) linear elements, provided that a Gaussian quadrature rule of $\mathcal{O}(4p)$ is employed.

Two likely explanations for this behaviour are (i) the increased numerical errors as α_i tends to α_i^{max}; and (ii) the difference between $\mathbf{\Phi}^p$ and $\tilde{\mathbf{\Phi}}^p$,



(a) $p = 3$, $q = 4$, $y^+(1) = 80$ (b) $p = 4$, $q = 3$, $y^+(1) = 128$

Figure 5.14: The wall friction mainly depends on the position of the first interpolation point. Hence, no difference is observed between a "uniform" stretching and a stretching of the "first element layer" only. To avoid numerical integration errors, the Gaussian quadrature has been chosen to exactly integrate a polynomial of degree $4p$.

$y^+(1^{st}$ cell)	16	24	32	64
$\alpha_i(q=2)$	1.0	2.0000	3.0000	–
$\alpha_i(q=3)$	1.0	1.5840	2.0613	3.4495
$\alpha_i(q=4)$	1.0	1.4142	1.7321	2.6458

Table 5.2: Stretching to be imposed to the high-order (mesh) nodes in the case of $p = 2$ such that the first interpolation point x_1 is located at $y^+(x_1) = 8$. Note that the second order mapping $q = 2$ becomes invalid as the element height exceeds $y^+(1) = 32$, cf. section 5.2.1.

cf. Rivière [143]. Whereas the former denotes the space of polynomials of degree less than p defined on the reference element T_{ref} (see section 3.2.1), the later denotes the spaces of polynomials defined on the physical space. Although the approximation results for $\mathbf{\Phi}^p$ and $\tilde{\mathbf{\Phi}}^p$ are the same on linear triangles/parallelograms, they differ in the case of general quadrilaterals ($q \geq 2$). For instance, a flow field, which is linear in the physical space, becomes of order q in the parametric space.

The values of the stretching coefficient are resumed in table 5.2. Since in the case of 2^{nd} order polynomials, only one interpolation point lies inside the computing cell, important stretchings are necessary, which explains the poor results.

In contrast to the low-order P^2-elements, the effect of the internal stretching changes, if a *high-order polynomial interpolation* ($p \geq 3$) is used; see

$y^+(1^{st}$ cell)	64	80	96	128	192
$\alpha_i(q=2)$	1.0	1.3077	1.5714	2.0000	–
$\alpha_i(q=4)$	1.0	1.1509	1.2782	1.4883	1.8089

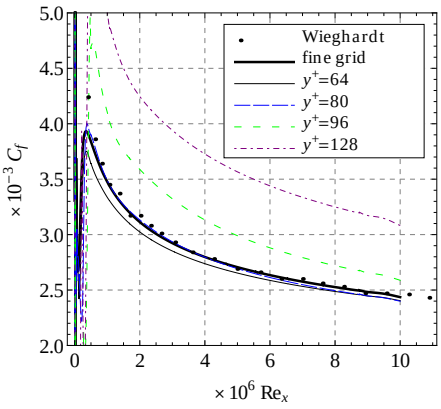
Table 5.3: Stretching to be imposed to the high-order (mesh) nodes in the case of $p = 4$ such that the first interpolation point x_1 is located at $y^+(x_1) = 16$.

Fig. 5.15. It is worth noting that even in the case of straight quadrilaterals, cf. Fig. 5.15(a) and 5.15(b), the influence of the Gaussian quadrature grows as the element size is increased. This behaviour is however to be expected because the eddy viscosity is defined by a rational function[2] (close to the wall); see section 2.3.3. Accordingly, it cannot be integrated exactly by numerical quadrature. Nevertheless, we observe that the default quadrature rule used within this thesis $\mathcal{O}(3p)$ leads to an acceptable compromise between accuracy and computational cost even for coarse meshes up to $y^+(1) \approx 64$ $(p = 4)$.

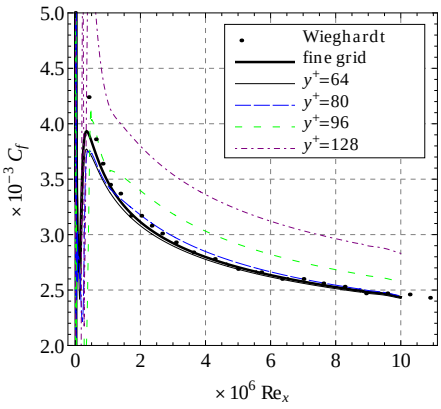
A comparison of figures 5.15(a) to 5.15(f) clearly demonstrates the interest of moving the interpolation nodes toward the wall. Indeed, almost grid independent results are obtained up to an element height which is more than doubled compared to the linear quadrilaterals ($q = 1$, $y^+(1) = 64$), leading to a significant reduction of the memory footprint of $\approx 30\%$. In addition, the non-dimensional velocity profile is almost not affected by moving the interpolation nodes toward the wall, although for the coarsest mesh, a huge part of the boundary layer ranging from the viscous sublayer up to the log-layer is discretized by one element only; see Fig. 5.16. For completeness, we should however mention that due to the poor mesh resolution in the outer boundary layer we had to increase the penalty constant by a factor two, in order to avoid a breakdown of the computation in the case of $y^+(1) = 192$. The corresponding stretching coefficients are summarized in table 5.3.

Finally, numerical experiments using $p = 3$ polynomials (not shown here) have confirmed the above considerations, whereby the use of a third order mapping ($q = 3$) has turned out to be the most efficient, leading to grid independent results up to $y^+(1) = 96$. While the quality of the second order mapping ($q = 2$) suffers from large values of α_i close to α_i^{max}, it seems that choosing $q > p$ also deteriorates the accuracy of the results.

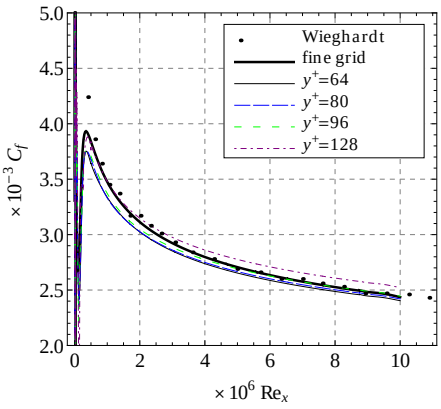
[2]The eddy viscosity μ_t is related to the working variable $\tilde{\mu}$ of the S-A turbulence model by $\mu_t = \tilde{\mu}(\tilde{\mu}/\mu)^3/[(\tilde{\mu}/\mu)^3 + c_{v1}^3]$. Hence, although μ_t is a rational function, it behaves like a polynomial of degree p sufficiently far from the wall.



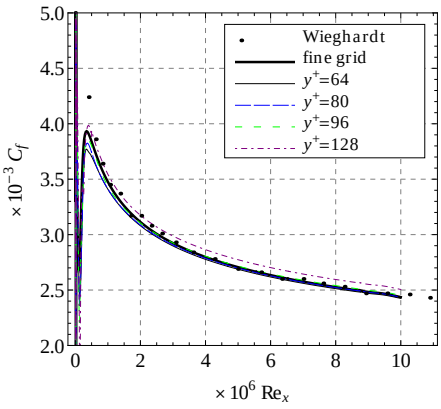
(a) $q = 1$, $\mathcal{O}(3p)$



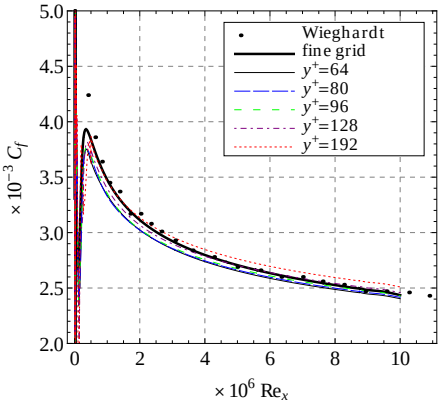
(b) $q = 1$, $\mathcal{O}(4p)$



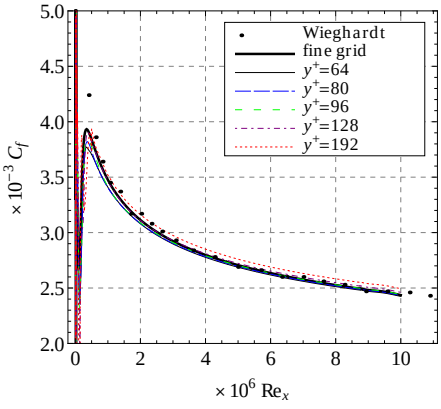
(c) $q = 2$, $\mathcal{O}(3p)$



(d) $q = 2$, $\mathcal{O}(4p)$



(e) $q = 4$, $\mathcal{O}(3p)$



(f) $q = 4$, $\mathcal{O}(4p)$

Figure 5.15: Skin friction coefficient ($p = 4$, $y^+(x_1) = 16$): Moving the interpolation points toward the wall, significantly improves the computed skin friction and allows to more than double the height of the first computing cell compared to standard linear quadrilaterals.

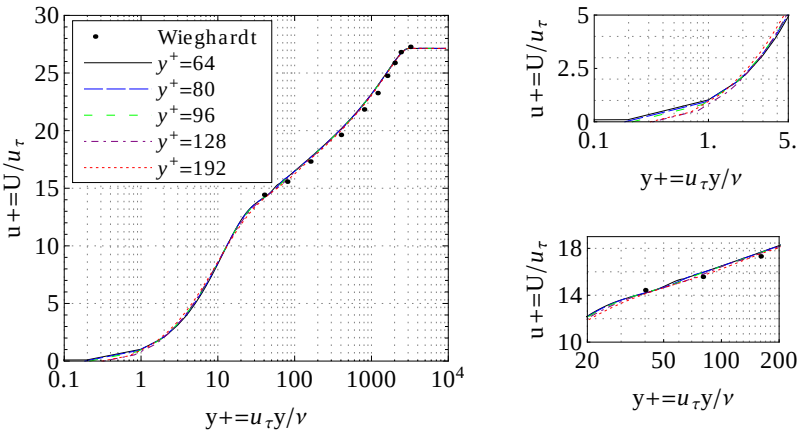


Figure 5.16: Non-dimensional velocity profiles ($p = 4$, $q = 4$, $y^+(x_1) = 16$, Gaussian quadrature $\mathcal{O} = 4p$): Grid independent velocity profiles are obtained up to an element of $y^+(1^{st}cell) \approx 192$, in which case a significant part of the boundary layer ranging from the viscous sublayer to the log-layer is discretized by a single element only.

5.3 Curved boundary cells: turbulent NACA 0012 aerofoil

In this section, we investigate the effect of curved boundary cells by means of the flow around the NACA 0012 aerofoil at an angle of attack of $\alpha = 3.59°$ and a Reynolds number of 1.86×10^6. The free-stream Mach number is set to $M_\infty = 0.3$. This standard test case is taken from the "Experimental Data Base for Computer Program Assessment" [2]. Note that the experiments in the wind tunnel have been realized for an angle of attack of $\alpha = 4°$. However, by taking into account the wall effect, the corrected angle of incidence is given by 3.59°, which corresponds to the value used for the numerical simulations.

The traditional definition of the aerofoil has been modified such that the trailing edge ($\bar{x} = 1$) has zero thickness, leading to the following analytical description

$$y(x) = \pm\frac{12}{20}\left(0.2969\sqrt{\bar{x}} - 0.126\bar{x} - 0.3516\bar{x}^2 + 0.2843\bar{x}^3 - 0.1015\bar{x}^4\right), \quad (5.9)$$

where $\bar{x} = x/(1.00893c)$ is the normalized chord length. The leading edge of the aerofoil is located at $\bar{x} = 0$.

In order to determine if similar mesh resolutions in terms of grid spacing and stretching can be achieved as for linear straight sided cells, two convergence studies are performed.

5.3.1 Chordwise resolution and high-order boundaries

An important aspect of high-order methods is the generation of curved grids because even a high-order polynomial approximation will inevitably lead to poor results, if the mesh is too coarse to correctly represent the geometry. Hence, in order to take full advantage of these methods, curved boundaries must be used and it is therefore not surprising that the generation of high aspect ratio curved meshes is still a subject of intense research.

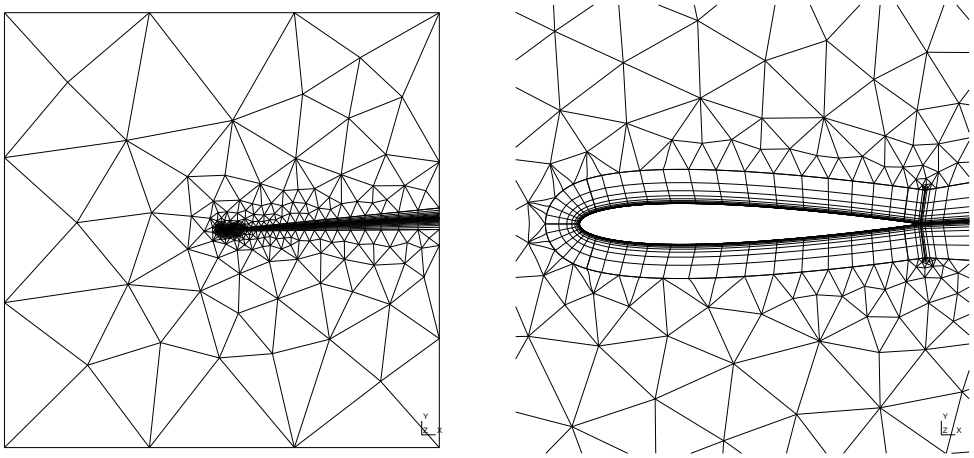
In this section, attention is focused on the discretization of curved boundaries. Thereto, we generate a family of five hybrid meshes which differ in the number of elements along the chord c. Grid points are clustered near the ends of the aerofoil leading to a chordwise spacing $\Delta x/c$ (measured along the profile) which varies between 1.7×10^{-2} (*grid I*) and 1.7×10^{-3} (*grid V*) at the leading edge, and between 5.5×10^{-3} (*grid I*) and 1.9×10^{-3} (*grid V*) at the trailing edge, respectively. To eliminate all eventual perturbations resulting from an insufficient discretization of the boundary layer, the grid spacing in the wall normal direction is set to 1.5×10^{-5} times the aerofoil chord, which corresponds to a small non-dimensional height of $y^+(1) \approx 1$. The outer boundary is located 30 chords away from the aerofoil; unstructured triangles are used as discretization elements in the farfield, whereas the wake and the near aerofoil regions are discretized by several layers of double sided curved quadrangles. The stretching parameter in the grid normal direction equals 1.6 for all grids. The grid specifications are summarized in table 5.4.

To ensure the positivity of the Spalart-Allmaras variable $\tilde{\mu}$, all simulations have been performed using the "clipped" S-A model. Except mentioned differently, the penalty coefficient is based on the "distance to the opposing node". On the aerofoil, the no-slip adiabatic boundary condition is imposed, whereas at the inlet plane (left), the whole state vector is defined (cf. "farfield boundary condition"). Along the rest of the boundary, the pressure is set to the freestream value.

To investigate the influence of the geometrical representation, computations have been performed using a 2^{nd}, 3^{rd} and 4^{th} order polynomial interpolation of the aerofoil. As an example, a $q = 4$ interpolation of *grid II* is shown at Fig. 5.17(a) and 5.17(b). The corresponding iso-Mach contours using a $p = 4$ approximation of the solution are displayed in Fig. 5.17(c).

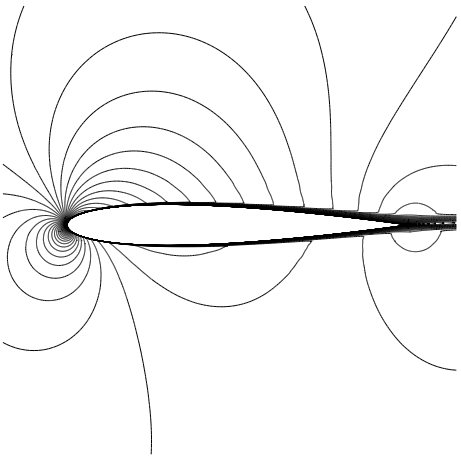
The computed lift and drag coefficients, C_l and C_d, respectively, for each grid (I to V with $q \in [2, 4]$) and interpolation order p are compared in Fig. 5.18. Both coefficients include the physical/consistent flux as well as the penalty contribution. In the present case, the latter is however negligible which is due to the small normal grid spacing of $y^+(1) \approx 1$. The fine mesh resolution in the wall normal direction also explains the little differences between $p = 2$, $p = 3$ and $p = 4$ results.

For clarity of the figures, the aerodynamic coefficients obtained using a $p = 1$ approximation are not shown here. Note however, that despite impor-



(a) Full grid ($q = 4$)

(b) Near aerofoil region ($q = 4$)



(c) Iso-Mach contours ($p = 4$)

Figure 5.17: NACA 0012 aerofoil mesh (*grid II*) with $\approx$ 2500 cells and iso-Mach number contours (0-0.4788, $\Delta = 0.0106$). The aerofoil is dicretized by 2×19 fourth order quadrangles ($q = 4$).

Grid	$\Delta_{leading}$	$\Delta_{trailing}$	nb elmts chord†	total nb elmts‡
I	1.68×10^{-2}	5.50×10^{-3}	3+9+3	1462 + 730
II	6.62×10^{-3}	1.97×10^{-3}	6+14+4	1802 + 766
III	5.17×10^{-3}	1.97×10^{-3}	7+19+4	2006 + 798
IV	2.10×10^{-3}	1.61×10^{-3}	11+24+5	2380 + 836
V	1.71×10^{-3}	1.94×10^{-3}	17+34+9	3570 + 1080

† Number of elements close to the leading edge $x \leq 0.05c$, in the middle of the aerofoil and close to the trailing edge $x \geq 0.95c$, respectively.
‡ Total number of curved quadrangles and triangles, respectively.

Table 5.4: Grid specifications for the NACA 0012 aerofoil ($Re_c = 1.86 \times 10^6$, $M_\infty = 0.3$, $\alpha = 3.59°$). The grid spacing in the wall normal direction is set to 1.5×10^{-5} which corresponds to $y^+(1) \approx 1$.

tant jumps in the streamwise direction, which are related to an insufficient discretization of the aerofoil chord, even the 1^{st} order interpolation leads to a satisfying estimation of the integrated lift and drag. Depending on the grid, the latter lie in the range of $C_l^{p=1} \in [0.390, 0.396]$ and $C_d^{p=1} \in 10^{-2} \times [1.1, 1.5]$. Not surprising are the instabilities encountered on most grids if we use a $p = 1$ interpolation together with 4^{th} order discretization of the aerofoil. Since the aerofoil geometry is described by a 4^{th} order interpolation, our standard quadrature rule for turbulent flows, which allows to exactly integrate polynomials up to degree $\mathcal{O}(3p)$, is insufficient on highly curved mesh cells. Increasing the integration order to $\mathcal{O}(4p)$ avoids the aforementioned instability. Furthermore, one should bear in mind that all computations in this chapter use the *facewise* definition of the penalty parameter (see chapter 3). The latter was found to be less stable than the recently proposed *elementwise* viscosity scale.

Highly curved grid cells. According to Fig. 5.18, not only the order of the polynomial approximation p but also the geometrical discretization q seem to have only a minor effect on the aerodynamics coefficients, provided a sufficiently small normal grid spacing. A closer look at the skin friction distribution reveals however important oscillations of the computed friction in the case of $q = 2$. The oscillations are mainly observed at the first part of the aerofoil, where the laminar-to-turbulent transition takes place[3]. Boundary cells in this region are characterized by the highest curvature. Note that the oscillations of C_f disappear as a higher order geometrical representation ($q \geq 3$) is used; see Fig. 5.19. Hence, $p = 4$ polynomials provide satisfying results even on the

[3]Remember that the trip term has been omitted in the current implementation of the S-A model. Accordingly, the laminar-to-turbulent transition is caused by purely numerical effects. The exact location of the transition might thus be mesh dependent.

coarsest mesh (cf. *grid I* with 2×15 elements along the aerofoil) if $q = 4$. For low-order $p = 2$ polynomials, at least *grid III* is required in order to ensure a smooth skin friction distribution, where it is again preferable to choose $q \geq 3$.

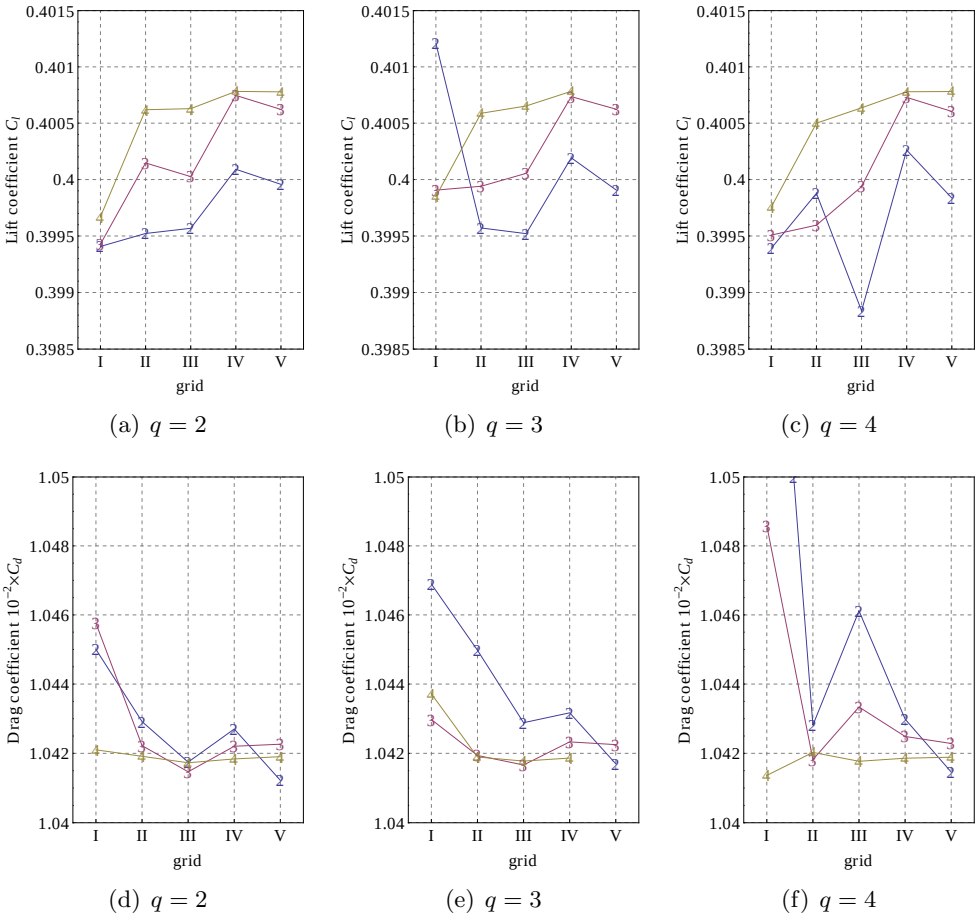
A comparison of Fig. 5.19 and 5.20 clearly demonstrates the dependency of the wall friction on the accurate discretization of the aerofoil geometry. The deviations of the interpolated aerofoil from the exact geometry are displayed in Fig. 5.20(a). Not surprisingly, best results are achieved by the 4^{th} order interpolation, which leads to a smooth wall friction, even on *grid I*. The corresponding 1^{st} and 2^{nd} order derivatives for the different geometry interpolations are calculate as

$$\begin{aligned} \frac{\partial y(\xi)}{\partial x} &= \frac{\partial y/\partial \xi}{\partial x/\partial \xi} \\ \frac{\partial^2 y(\xi)}{\partial x^2} &= \left(\frac{\partial^2 y}{\partial \xi^2} \frac{\partial x}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial^2 x}{\partial \xi^2} \right) \left(\frac{\partial x}{\partial \xi} \right)^{-3} . \end{aligned} \tag{5.10}$$

Details of the third grid cell ($x/c \in [0.05, 0.12]$), where the largest friction jumps are observed, are given in Fig. 5.20(b) and 5.20(c). Although the C^1 continuity of the discrete aerofoil is not guaranteed, it is significantly improved as q is increased.

In contrast to previous works, e.g. [107, 125], where curved grid cells have been obtained by an ad hoc meshing tool, the meshes used within this thesis have been generated in two steps by the GMSH grid generator [70]. Starting from a straight sided mesh, we set up a Lagrangian interpolation with uniformly distributed nodes in both the reference and the physical space. On the domain boundaries, the high-order nodes are placed so as to match the aerofoil geometry. Next, linear elastic material properties are applied to the grid cells and an iterative solver is used in order to minimize the deformation energy of the mesh. Here, the higher the aspect ratio of the grid cells, the more layers of grid cells must be curved to avoid singular mappings. Treating the mesh as an elastic material inherently results in a final grid where the curvature is progressively reduced, leading to linear elements in the farfield. So far, no effort has been made to optimize the position of the high-order mesh nodes. The latter could be used to minimize differences between the interpolated and the exact geometry, or to ensure continuous normal vectors between adjacent grid cells. Furthermore, the inner-element stretching described in section 5.2 is not yet implemented for complex geometries.

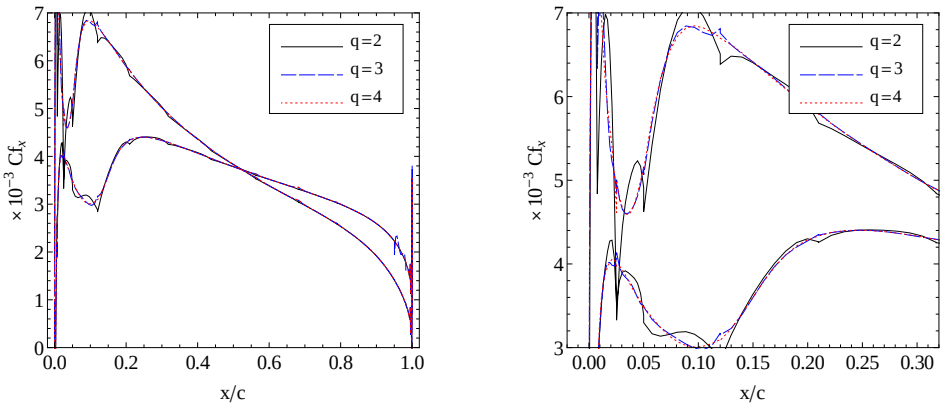
High aspect ratio grid cells. The previous section has revealed that a high-order representation significantly improves the computed friction, in regions where the grid curvature is important. Surprisingly, the $q = 4$ interpolation causes small oscillations of the wall friction at the middle of the aerofoil, which are not observed with $q = 2$; see Fig. 5.21(a). Contrary to the leading edge, grid cells in this region are characterized by a high aspect ratio, whereas



(a) $q = 2$ (b) $q = 3$ (c) $q = 4$

(d) $q = 2$ (e) $q = 3$ (f) $q = 4$

Figure 5.18: NACA 0012 ($Re_\infty = 1.86 \times 10^6$, $M_\infty = 0.3$, $\alpha = 3.59°$, $p = 4$): Given the small element height in the wall normal direction, $y^+(1) = 1$, a good estimation of the integrated lift and drag coefficients is obtained, even on *grid I* (2×15 elements along the aerofoil chord).

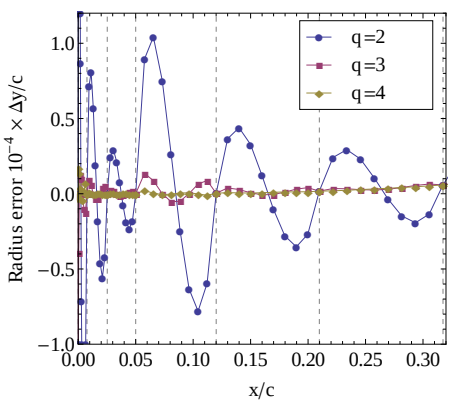


(a) Full aerofoil (b) Leading edge

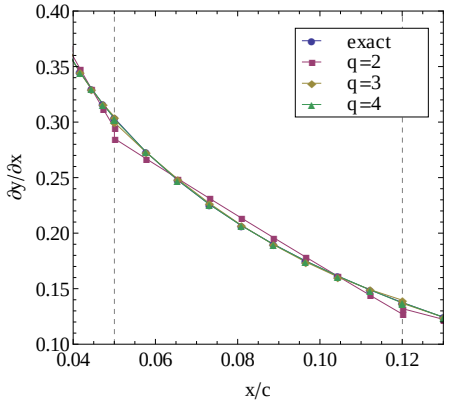
Figure 5.19: NACA 0012 ($Re_\infty = 1.86 \times 10^6$, $M_\infty = 0.3$, $\alpha = 3.59°$, $p = 4$, *grid I*): The low 2^{nd} order interpolation of the aerofoil geometry causes important oscillations of the skin friction distribution near the leading edge, where the grid curvature is important.

the curvature is rather unimportant. Although the amplitude of these fluctuations decreases rapidly as the the mesh is refined, the same behaviour is found on grids *I*, *II* and *III* ($p = 4$). On *grid III*, the relative local error is less than $\|\Delta C_f\|/\|C_f\| < 0.25\%$.

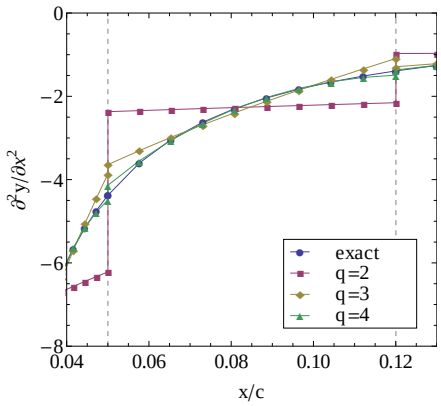
The origin of the local friction error is all the more mysterious as the 4^{th} order representation is very close to the exact aerofoil geometry. As illustrated by Fig. 5.21(b), even the second order derivative is almost continuous at the interface between adjacent boundary cells ($q = 4$), whereas large jumps are observed for $q = 2$. To exclude a poor element quality as the origin of the loss of accuracy, we have computed the inverse of the mapping Jacobian $\mathbf{J}^{-1} \equiv (\partial \mathbf{x}/\partial \boldsymbol{\xi})^{-1}$ for the boundary cell presenting the largest deviations, i.e. $x/c \in [0.68, 0.79]$; see Fig. 5.22. The presented non-dimensional values are obtained by dividing the Jacobian by two times the volume of the associated straight-sided element. Fig. 5.22(a) and 5.22(b) clearly show that, independently of the geometrical interpolation order, the Jacobian is nearly constant in the wall normal direction. This indicates a (desired) uniform grid height along the profile and suggests a "correct" curvature of the 2^{nd}, 3^{rd}... element layers. In addition, despite a decrease at the ends of the grid cell, the variations of the Jacobian remain small in the chordwise direction and in particular, the mapping is far from being singular; see Fig. 5.22(c) and 5.22(d). Although the causes for these oscillations are not yet understood, we can exclude numerical integration errors, since the same behaviour is observed with an extremely accurate Gaussian quadrature rule of $\mathcal{O}(5p)$. Assuming that the friction error is not due to deviations from the aerofoil geometry, another explanation could be a reduction of the formal order of accuracy of the DG method, if curved



(a) Deviations from the exact geometry

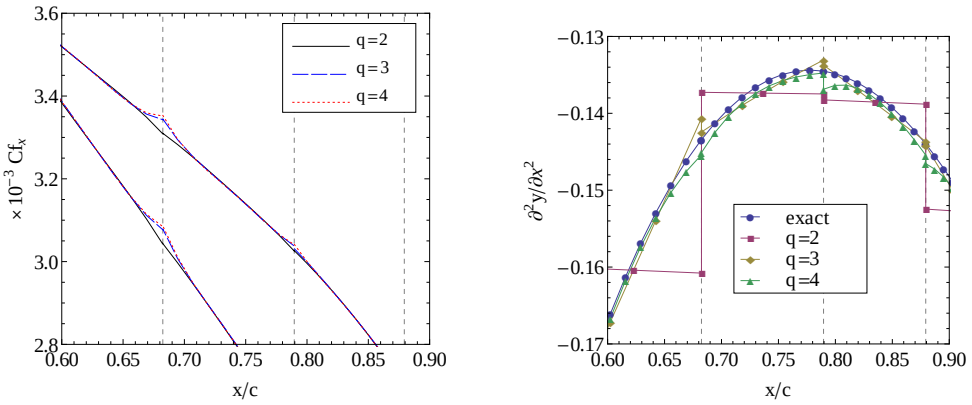


(b) 1^{st} order derivative



(c) 2^{nd} order derivative

Figure 5.20: Details of the leading edge (*grid I*): The significant improvement of the geometrical representation by increasing q results in a much smoother skin friction in regions, where the grid curvature is important.



(a) Skin friction (*grid I*, $p = 4$)

(b) Second order derivative

Figure 5.21: Despite a nearly continuous second order derivative of the interpolated geometry, small oscillations of the skin friction appear at the middle of the aerofoil as q is increased. The grid cells in this region are characterized by a high aspect ratio. The dashed vertical lines represent the element boundaries.

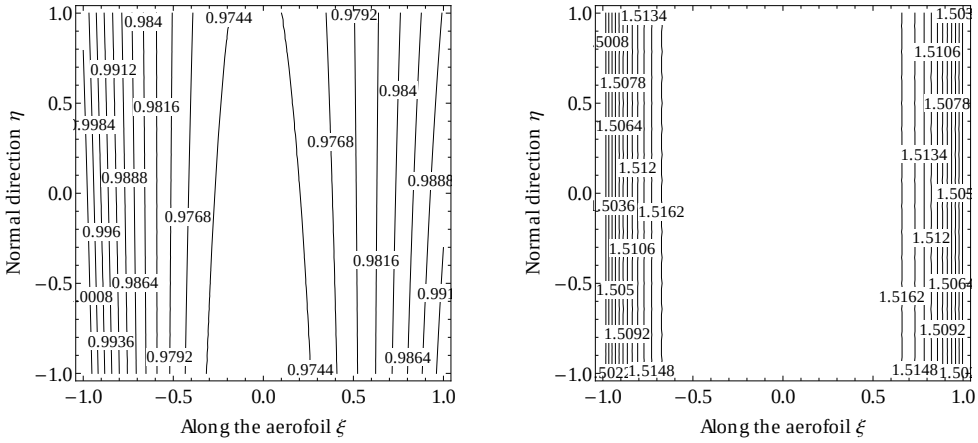
elements are used; cf. section 5.2.2. A reduction of the order of accuracy has for instance been reported by Todd *et al.* [128], who studied the laminar flow around a cylinder using a $q = 2$ and $q = 3$ interpolation of the geometry.

5.3.2 Influence of the grid curvature on the boundary resolution

We finish our analysis of the flow around the NACA 0012 aerofoil with a few words about the possible influence of the element curvature on the grid spacing in the wall normal direction. The latter has been studied extensively for straight-sided meshes at the beginning of this chapter. The goal of this section is to determine whether similar grid spacings can be achieved as in the case of the flow along the turbulent flat plate. Based on the previous results, see section 5.3.1, we have chosen *grid II* as reference grid for computations using $p = 4$ and *grid III* for $p = 2$. The normal spacing Δy of the first node off the wall varies between $5 \times 10^{-6}c$ and $7.5 \times 10^{-4}c$. The aerofoil is discretized by a 3^{rd} order polynomial interpolation.

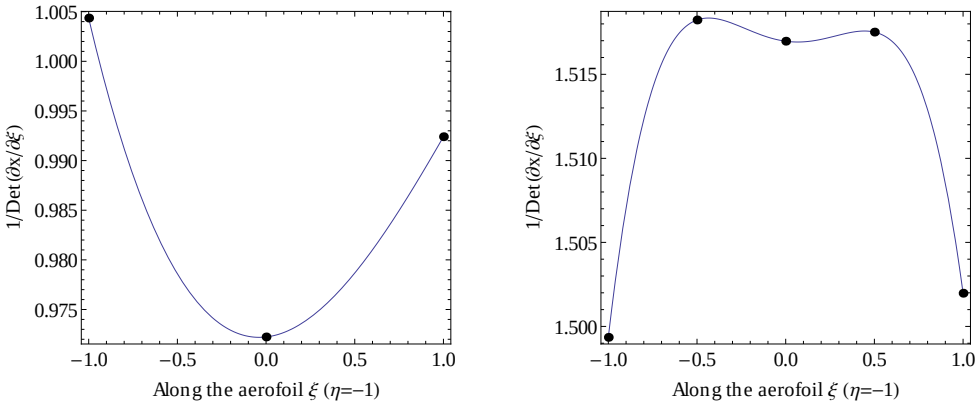
The influence of the off-wall spacing on the skin friction using a $p = 4$ approximation is illustrated in Fig. 5.23. All results are in good agreement and a discernible difference in the peak friction appears for $\Delta/c \geq 50 \times 10^{-5}$ only; see Fig. 5.23(a). According to Fig. 5.23(b), this value corresponds to a non-dimensional height of roughly $y^+(1) \approx 60$, roughly the same upper limit we already encountered for the turbulent flat plate.

A second similarity to previous flat plate results is that for triangular grids



(a) Entire element ($p = 2$)

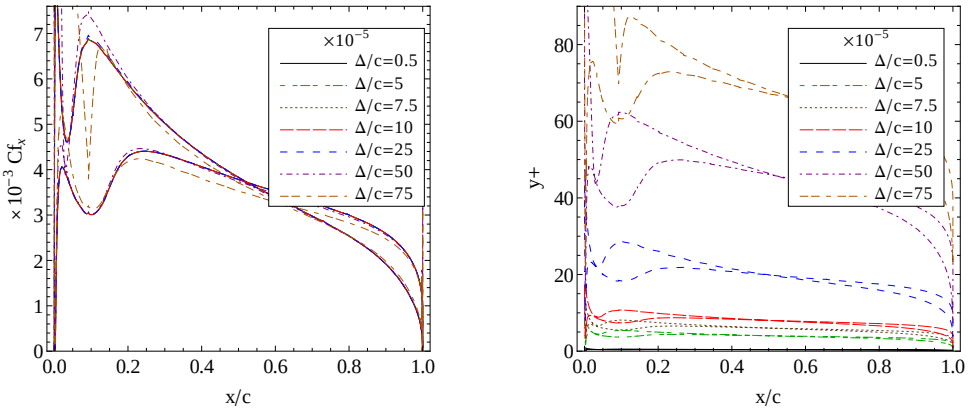
(b) Entire element ($p = 4$)



(c) Boundary face ($p = 2$)

(d) Boundary face ($p = 4$)

Figure 5.22: Inverse of the mapping Jacobian for the boundary cell presenting the largest oscillations of the skin friction, i.e. $x/c \in [0.68, 0.79]$. The nearly constant Jacobian in the normal direction indicates a uniform element height and suggests a "correct" curvature of the 2^{nd}, 3^{rd}... boundary layers. Despite small variations at the ends of the element, the Jacobian changes little in the chordwise direction. • indicates the interpolation nodes.



(a) Skin friction

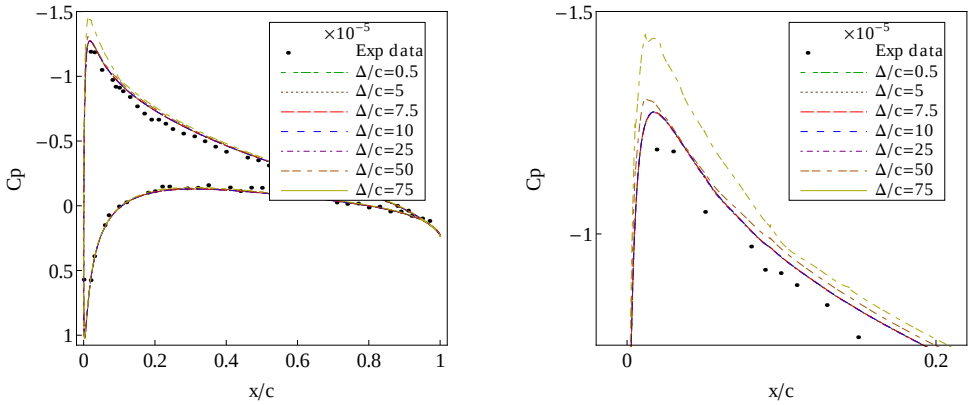
(b) Non-dimensional height

Figure 5.23: NACA 0012 ($Re_\infty = 1.86 \times 10^6$, $M_\infty = 0.3$, $\alpha = 3.59°$, $p = 4$, $q = 3$, *grid II*): Roughly the same non-dimensional height as for straight-sided boundary cells, i.e. $y^+(1) \approx 60$, is required to ensure a grid independent skin friction (consistent and penalty term).

(not shown here) the element height close to the wall should be reduced by a factor 2 to 4 in order to avoid oscillations of the skin friction. Whereas the pressure distribution is almost unaffected, we observe important jumps of the skin friction if $\Delta/c \geq 25$. The corresponding pressure distributions for hybrid grids are compared to the experimental data in Fig. 5.24. Although the measured and the computed pressure are in excellent agreement on the pressure side, the computation slightly overestimates the peak value on the suction side. The most likely reasons for this small discrepancy are the wall effect in the wind tunnel, which has been partially taken into account by correcting the flow incidence angle, the exact location of the laminar-to-turbulent transition, and the turbulence model. Remember that the present study has been realized without the trip term of the S-A turbulence model. Hence, the turbulence transition takes place due to numerical reasons which introduces a certain grid dependency of the results; cf. section 2.3.3. We also should bear in mind the minor modification of the aerofoil profile, which ensures a zero thickness of the trailing edge.

5.4 Conclusions

The high-order RANS solver has been applied to two classical test cases, namely a turbulent flat plate and a NACA 0012 aerofoil at an incidence of $\alpha = 3.59°$. Given the simplicity of the flow, the flat plate computations were used to investigate the influence of the interpolation order on the required grid resolution on the one hand, and the behaviour of different element types



(a) Whole aerofoil (*grid II*, $p = 4$, $q = 3$)

(b) Leading edge (suction side)

Figure 5.24: NACA 0012 ($Re_\infty = 1.86 \times 10^6$, $M_\infty = 0.3$, $\alpha = 3.59°$): Pressure coefficient along the aerofoil for different wall distances of the first cell layer. Despite a small discrepancy of the peak value on the suction side, the computed pressure distribution agrees well with the experimental data.

(triangles, quadrangles) on the other hand. The computation of the turbulent flow around the NACA aerofoil has been performed to verify these conclusions in the case of curved geometries.

The study has clearly shown

- an important reduction in the number of degrees of freedom with respect to a classical finite volume method, both in terms of spacing and stretching. Especially the high stretching and the smooth transition of the solution in the case of structured-unstructured connections can greatly simplify mesh generation in complex geometries;
- the importance of including the penalty contribution in the skin friction; this furthermore implies that a strict imposition of the no-slip wall boundary condition will probably degrade the solution accuracy;
- the advantage of using regular quadrilateral regions in the boundary layer. Besides a lower number of degrees of freedom, they give smoother results which allows the use of 2 to 4 times larger off-wall spacings compared to equivalent triangular meshes;
- the insensitivity of the solution with respect to the penalty parameter;
- the improved accuracy of the shear friction by approaching the interpolation nodes close to the wall. To avoid a deterioration of the solution caused by a large inner-element deformation/stretching, best results were obtained by choosing $q \leq p$;

- the high-sensitivity of the DGFEM method to the quality of the curved boundary representation. Although the required off-wall spacings are similar as in the case of straight boundaries, the C^1 continuity of the discrete geometry is of utmost importance.

Chapter 6

Numerical RANS applications

Contents

In this chapter two flow simulations are presented to demonstrate the applicability of the high-order interior penalty method to more complex applications. By that means we further validate the implementation of the Spalart-Allmaras turbulence model. The first test case is the two-dimensional *L1T2 three-element aerofoil* which has been extensively studied in the literature and for which experimental measurements are available (section 6.1). As a second test case we compute the flow in a *three-dimensional compressor cascade* featuring secondary flow and hub stall (section 6.2). A comparison with two state-of-the-art finite volume solvers is provided.

6.1 L1T2 three-element aerofoil

In this section we consider the turbulent flow around the L1T2 three-element aerofoil at take-off configuration ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and an angle of attack $\alpha = 20.18°$). The L1T2 case consists of a main element, a slat forward of the main element (deflection angle of 25°) and a Fowler flap aft of the main element (deflection angle of 20°). This application is a typical test case for high lift configurations (see e.g. [60,85]) and has recently been studied by Hartmann *et al.* [81] within the European research project ADIGMA [104]. The ADIGMA project is intended for the development of adaptive higher-order variational methods for aerospace applications. In addition to the various numerical studies that exist in the literature, experimental measurements have been performed in the early 1970s by the former British Aerospace Company (BAC) [121].

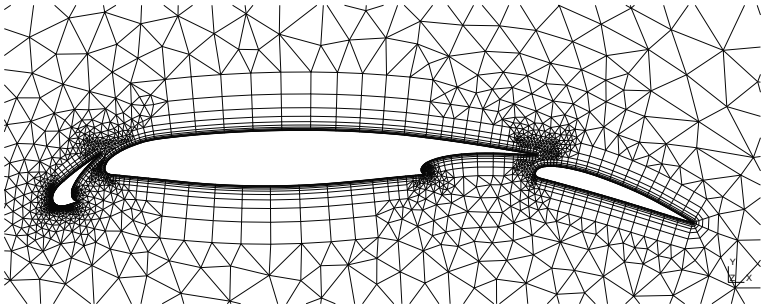
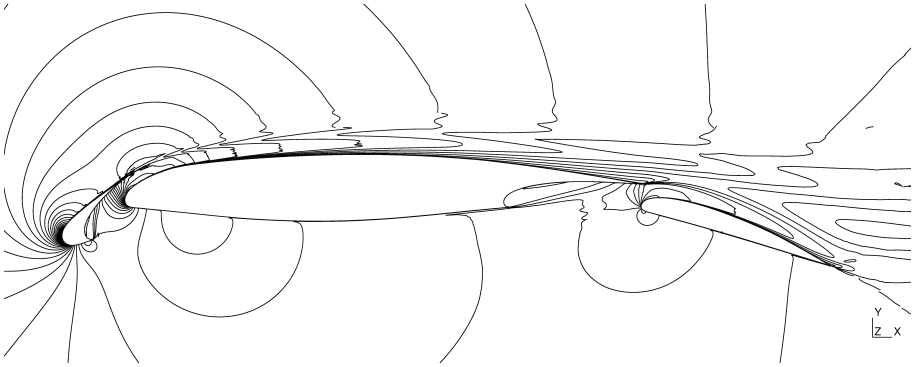


Figure 6.1: Global view of the 3^{rd} order curved hybrid mesh ($\approx$ 6000 elements) for the turbulent flow around the L1T2 three-element aerofoil ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and $\alpha = 20.18°$).

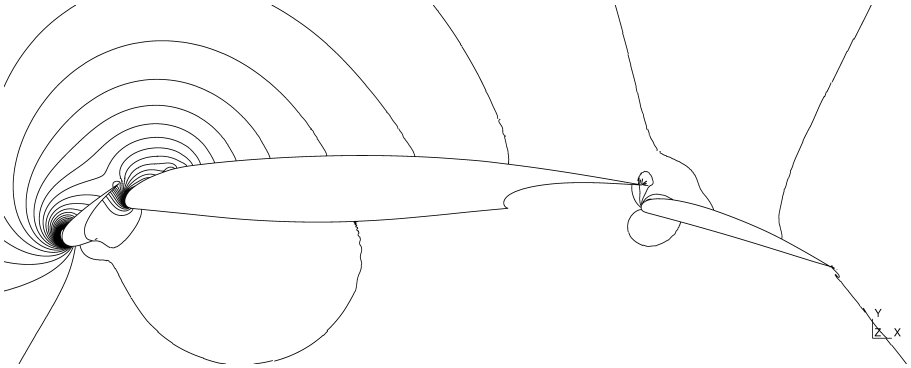
Fig. 6.1 shows the initial coarse mesh (*grid I*) generated for this test case. The hybrid computational mesh consists of 3986 (unstructured) triangles and 1970 third order curved quadrilaterals in the boundary layer. The element height of the first layer surrounding the aerofoil equals $\Delta/c = 1.5 \times 10^{-4}$ (slat), $\Delta/c = 2.5 \times 10^{-4}$ (main element) and $\Delta/c = 1.88 \times 10^{-4}$ (flap) respectively. The square outer boundary is located at 36 chords away from the aerofoil. Additionally, a second mesh (*grid II*) has been generated using an isotropic refinement. To this end, every element face is divided in half.

The $p = 4$ solution on the coarse mesh (*grid I*) is presented in Fig. 6.2. As the L1T2 aerofoil is a take-off configuration, there is no considerable flow separation, which limits turbulence modelling errors. Although we were not able to decrease the non-linear residual to machine precision on the coarse grid, the computed lift coefficient of $C_l = 4.04$ agrees well with the experimental measurements and other computations; see table 6.1. The reason we could not reach machine precision is the poor grid resolution close to the leading edge of the slat. In this region the boundary layer is extremely thin and therefore captured by a single element only. In combination with a strong wall curvature and an important acceleration of the flow (up to Mach $\approx$ 1), the high-order solution was found to be extremely oscillatory; see Fig. 6.3. Problems with the meshing tool prevented us from reducing the element height, whilst keeping unchanged the chordwise mesh resolution. To improve the stability and to better account for the high-order spatial discretization future work should therefore concentrate on curved grid generation, and adaptive mesh refinement techniques (see e.g. [81]) should be investigated.

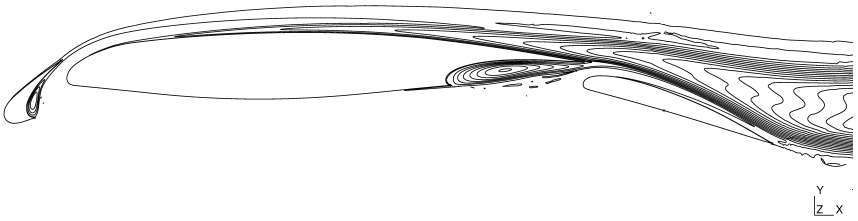
Table 6.1 summarizes the lift and drag coefficients for different interpolation degrees on *grid I* and *grid II*. Whereas C_l rapidly converges to the experimental results, all simulations overestimate C_d. However, predicting the drag in a high-lift configuration is an extremely difficult task [147]. An uncertain transition location and an insufficient mesh resolution in the wake region of the slat (cf. Fig. 6.5 and 6.6) may explain the high drag values.



(a) Mach number



(b) Pressure



(c) Turbulent eddy viscosity

Figure 6.2: $p = 4$ approximation on the coarse 3^{rd} order mesh (*grid I* with $\sim$ 6000 elements) for the L1T2 high lift configuration ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and $\alpha = 20.18°$).

Nonetheless, even on the coarse grid the $p = 4$ interpolation performs rather well and furthermore outperforms the $p = 2$ solution on the fine grid, whilst demanding less degrees of freedom.

	C_l	C_d
$p = 1$, *grid I* (coarse)	3.53	0.144
$p = 4$, *grid I* (coarse)†	4.00	0.084
$p = 1$, *grid II* (fine)	3.71	0.118
$p = 2$, *grid II* (fine)†	3.80	0.089
$p = 3$, *grid II* (fine)†	4.05	0.082
$p = 4$, *grid II* (fine)	4.04	0.077
Hartmann [81] (DGFEM)	3.96	0.071
Rumsey [149] ($k-\omega$ EARSM‡, FV)	4.08	0.068
Rudnik [147] ($k-\omega$, FV)	4.01	0.071
Experiment, corr	4.11	0.068

† The non-linear residual could not be decreased to machine precision.
‡ Explicit algebraic Reynolds stress models.

Table 6.1: Summary of the computed aerodynamic coefficients for the L1T2 high lift configuration ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and $\alpha = 20.18°$). C_l and C_d both include the conservative and the penalty contribution.

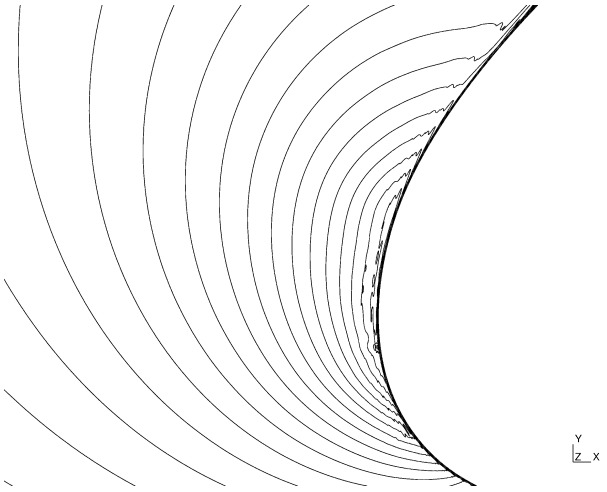


Figure 6.3: Details of the iso-Mach number contours on *grid I* close to the leading edge of the slat for the L1T2 high lift configuration ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and $\alpha = 20.18°$), $p = 4$.

The pressure and friction distributions are presented in Fig. 6.4 and 6.5. Whereas the $p = 1$ solution on the coarse grid is clearly under-resolved, the pressure distribution for $p = 4$ is very similar on both meshes. Except in the mixing region between the slat and main wing boundary layer, C_p is in good agreement with the experimental data. Not surprisingly the differences

are more visible on the shear friction distribution C_f (see Fig. 6.5), whereby the oscillations on the flap are - at least partially - a consequence of the poor quality of the available aerofoil geometry used for the mesh generation.

Finally the iso-Mach number contours around the slat are compared in Fig. 6.6. As stated previously, the region aft of the slat should be further refined to accurately capture the slat wake. In addition, the thickness of the boundary layer strongly depends on the mesh resolution.

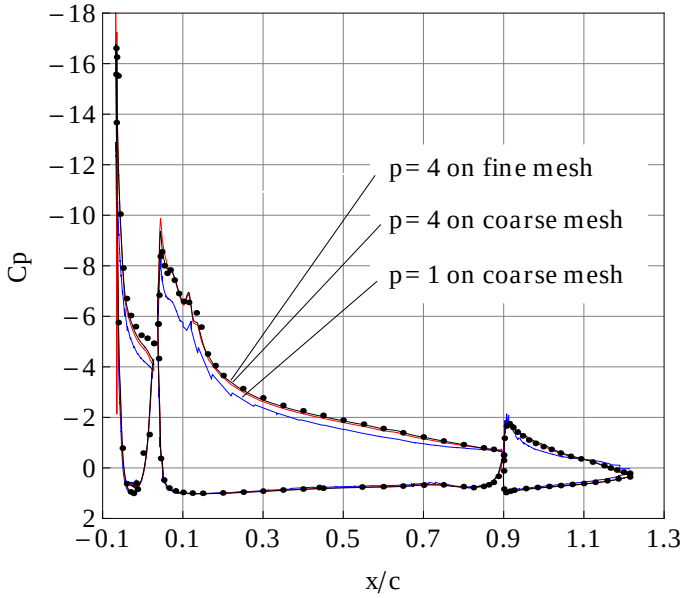


Figure 6.4: Surface pressure distribution for the L1T2 high lift configuration ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and $\alpha = 20.18°$) for different 3^{rd} order curved hybrid meshes. The single (blue) line represents the $p = 1$ solution on *grid I*. The $p = 4$ solutions on *grid I* (red) and *grid II* (black) are nearly superposed. '•' denotes the experimental data.

6.2 Turbulent three-dimensional high lift cascade flow

To validate the 3D implementation of the S-A turbulence model and to demonstrate the robustness of the DG method, the turbulent flow in a highly loaded low aspect ratio compressor cascade featuring secondary flow and hub stall is computed; see Fig 6.7.

The blade has a chord length of $c = 46.3\,\text{mm}$ and a span of $l = 100\,\text{mm}$. The pitch to chord ratio equals $t/c = 0.72$. The stagger angle is $\gamma = 19.7°$. According to the experimental measurements, a total temperature of $T_t \approx 293\,\text{K}$ and a total pressure ranging from $p_t = 93745\,\text{Pa}$ (end-wall) to $p_t =$

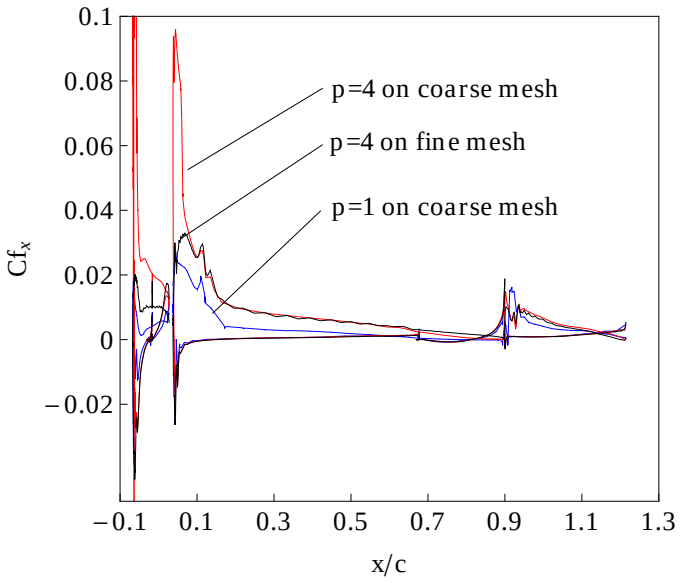
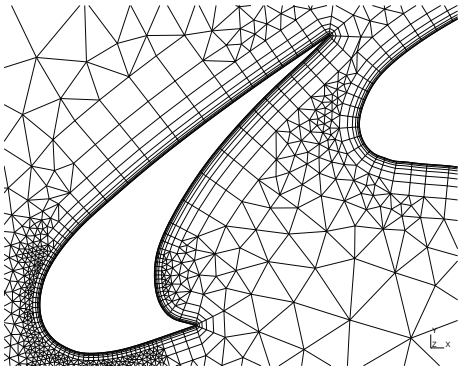


Figure 6.5: Friction distribution in $x-$direction for the L1T2 high lift configuration ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and $\alpha = 20.18°$) for different 3^{rd} order curved hybrid meshes: $p = 1$ solution on *grid I* (blue), $p = 4$ solution on *grid I* (red) and $p = 4$ solution on *grid II* (black).

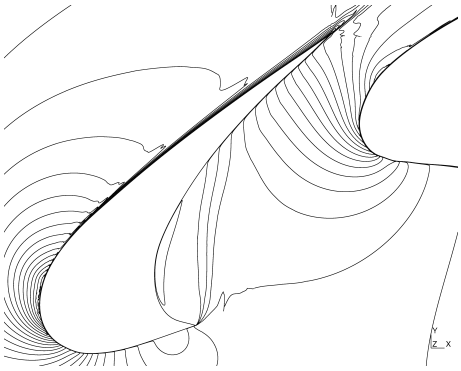
120000 Pa (half span) are imposed at the inlet boundary. The corresponding temperature and pressure profiles are displayed in Fig. 6.8(a). The inlet flow angle is $\alpha = 34.8°$. At the exit plane, located at one chord length away from the blade, the static pressure is set to $p = 101000$ Pa, leading to an inflow Mach number of $M_1 = 0.63$. The corresponding chord Reynolds number based on inlet quantities is $Re_c = 6.40 \times 10^5$. On the blade and on the end-walls, adiabatic no-slip boundary conditions are prescribed. Because the stationary flow is symmetric in the spanwise direction, only half of the blade is simulated, while symmetric boundary conditions are imposed on the mid-span plane. Furthermore, the flow is assumed to be periodic in the "circumferential" (vertical) direction. An overview of the flow parameters is given in Fig. 6.8(b).

Due to the low aspect ratio of the cascade and the rather small pitch to chord ratio, the flow through the compressor cascade is characterized by strong three-dimensional effects including secondary flow, a horseshoe vortex near the trailing edge and a separation bubble on the blade suction side.

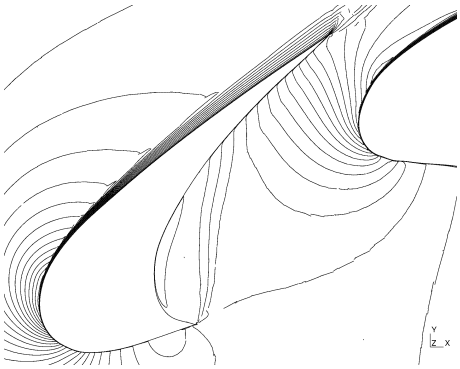
Computations have been performed using two hybrid O-type grids. *Grid I* involves 21981 (straight sided) prisms and 20111 (2^{nd} order curved) hexahedra; 1547 of them are located on the blade. The height of the first element off the end-wall / blade equals 10^{-2} mm / 2×10^{-2} mm, respectively. *Grid II* is composed of 18428 prisms and 9520 ($q = 2$) hexahedra, with 1190 cells on



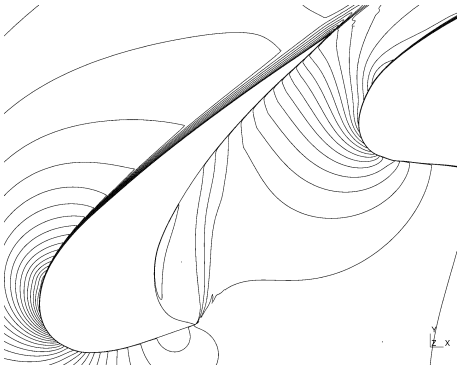
(a) Zoom on *grid I* ($\sim$ 6000 elements)



(b) Mach number ($p = 4$, *grid I*)



(c) Mach number ($p = 1$, *grid II*)



(d) Mach number ($p = 4$, *grid II*)

Figure 6.6: Comparison of the iso-Mach number contours ($M \in [0, 1]$, $\Delta = 1/30$) around the slat for the L1T2 high lift configuration ($Re_c = 3.52 \times 10^6$, $M = 0.197$ and $\alpha = 20.18°$).

the blade. The corresponding grid spacings are 4×10^{-2} mm (end-wall) and 2×10^{-2} mm (blade). Both meshes are generated by extrusion in the spanwise direction (17 layers) of a 2D mesh. Besides the element height close to the end-wall and an increased grid size in the farfield, the meshes mainly differ in the boundary layer surrounding the blade (chordwise resolution and normal extension). While in case of *grid I*, the 4 element layers close to the blade have a constant height (before increasing progressively), the stretching parameter equals 1.65 in case of *grid II*; cf. Fig. 6.9. As a comparison, the element height of the coarsest finite volume mesh equals 3×10^{-2} mm resulting in a mean y^+ of 1 to 2 with a maximum of $y^+ \approx 10$. A mesh convergence study has shown that this grid spacing delivers acceptable results. The fine FV meshes ($\Delta/c = 2.15 \times 10^{-5}$) involve approximately 3×10^6 grid points.

The non-linear convergence history for $p = 3$ polynomials on *grid II* is shown in Fig. 6.10. We notice that the residual norm of the turbulence model decreases very fast compared to the Navier-Stokes equations. This can be

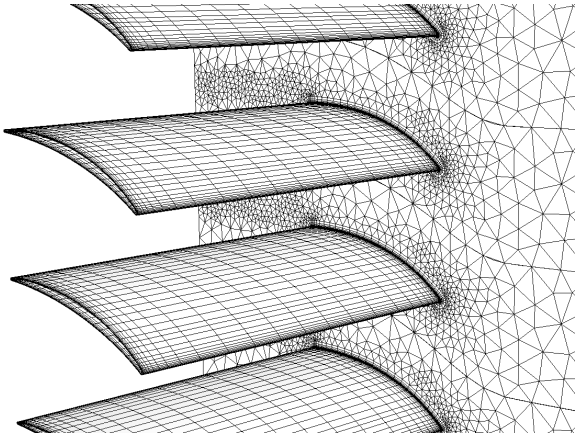


Figure 6.7: Turbulent flow in a highly loaded three-dimensional compressor cascade ($Re_c = 6.40 \times 10^5$, $M_1 = 0.63$): global view on *grid II* (27948 cells).

explained by the large initial value of the eddy viscosity $\tilde{\mu}/\mu = 233$, which significantly reduces the stiffness of the S-A source term. The computation is started with a uniform flow field and a quite small CFL number of 10. The latter improves the stability during the growing of the boundary layer along the end-wall. Otherwise, instabilities were encountered due to the large element size at the inlet boundary in addition to a flow inversion at the first Newton steps. The flow inversion is a consequence of the adverse pressure gradient because the total inlet pressure in the vicinity of the end-wall is smaller than the initial static pressure inside the cascade.

To reduce the computing time, the simulation is started with a $p = 2$ approximation, which - once sufficiently converged - is projected onto $p = 3$ polynomials. Up to $p = 3$, a BILU(0) preconditioner is used; for $p = 4$ the latter is replaced by a block-Jacobi preconditioner. For a further gain of CPU time, the preconditioner is updated every 3 Newton iterations only.

The pressure distribution on the blade using a $p = 3$ (left part of the blade)/$p = 4$ (right part of the blade) approximation is shown in Fig. 6.11 for grids *I* and *II*, respectively. Whereas no discernible difference is observed on the pressure side, the computation is slightly underresolved on the suction side if *grid II* is used with $p = 3$ polynomials.

Details of the surface pressure and axial shear stress distributions at different locations in the spanwise direction are compared to the finite volume results in Fig. 6.12. The latter have been obtained by two state-of-the-art finite volume flow solvers, namely the Argo software package from Cenaero [67] and the elsA software from Onera [26,27]. Despite the coarser grid resolution used in this work, all three methods lead to a very similar surface pressure. The small differences between Argo (finite volume) and elsA are probably due to a distinct implementation of the S-A model. A closer look to the suction side on

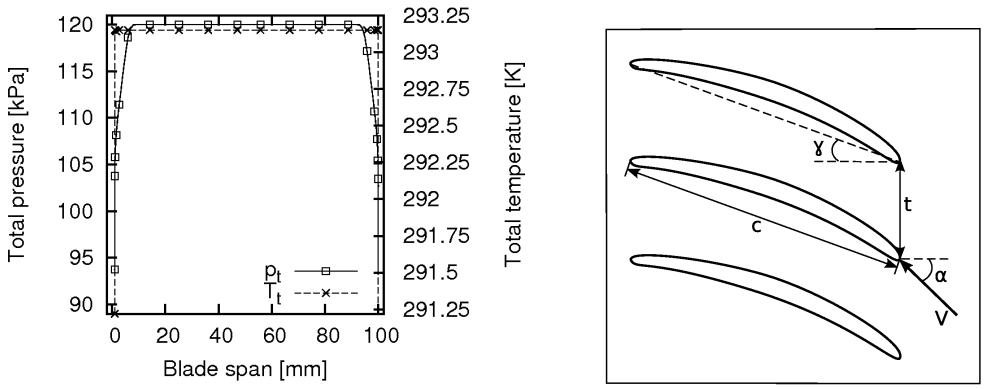


Figure 6.8: Total temperature/pressure profiles at the inlet plane and overview of the stator cascade parameters for the turbulent flow in a highly loaded three-dimensional compressor cascade ($Re_c = 6.40 \times 10^5$, $M_1 = 0.63$.) The parameters used for the computation are $c = 46.3\,\text{mm}$, $t/c = 0.72$, $\gamma = 19.7°$ and $\alpha = 34.8°$.

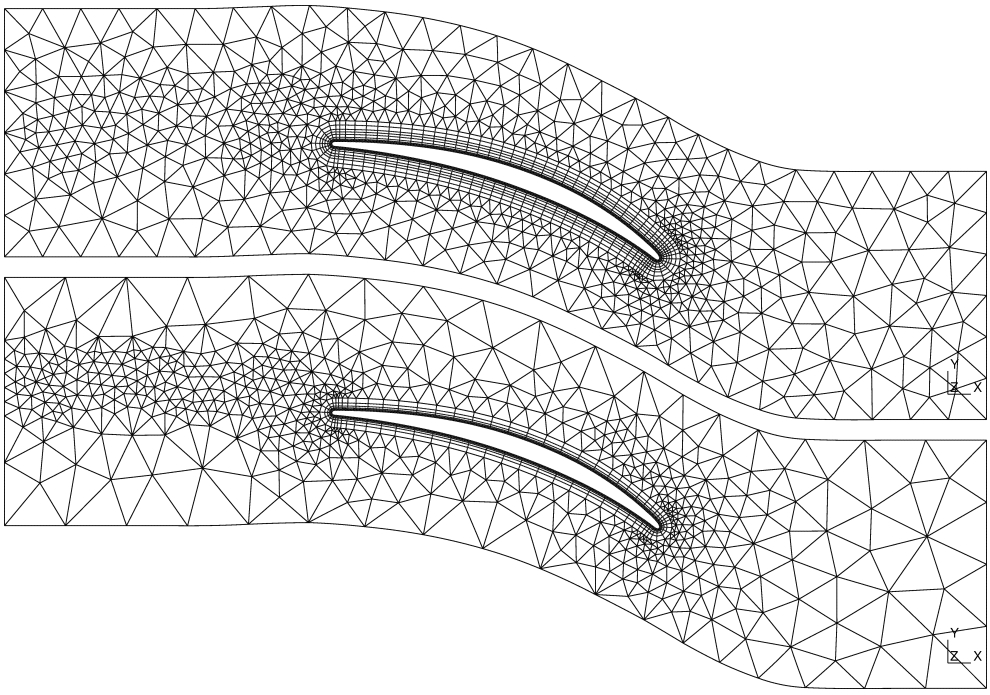


Figure 6.9: Details of the meshes used for the turbulent compressor cascade, top: *grid I* (2476 elements/layer), bottom: *grid II* (1644 elements/layer). Both meshes are extended by 17 layers in the spanwise direction; the corresponding grid normal spacings at the end-wall are $10^{-2}\,\text{mm}$ (*grid I*) and $4 \times 10^{-2}\,\text{mm}$ (*grid II*). The outlet plane is located at one chord length away from the blade.

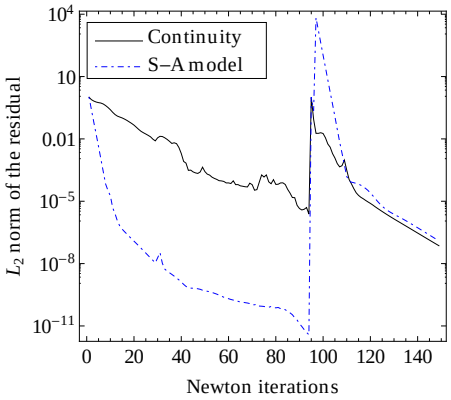
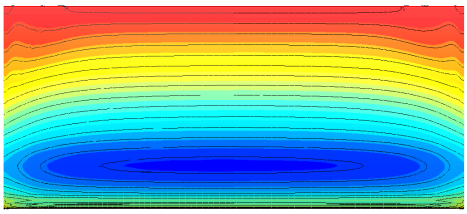
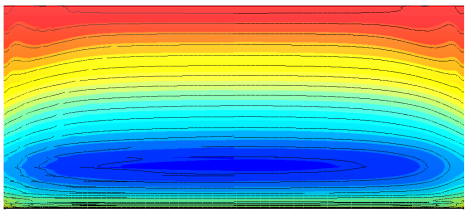


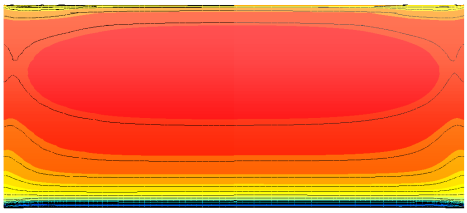
Figure 6.10: Non-linear convergence histories for the turbulent compressor cascade ($Re_c = 6.40 \times 10^5$, $M_1 = 0.63$.) using a $p = 3$ approximation on *grid II*. The fast convergence of the turbulence model can be explained by the large initial value of the eddy viscosity $\tilde{\mu}/\mu = 233$. To further reduce the computing time, the simulation is started with $p = 2$.



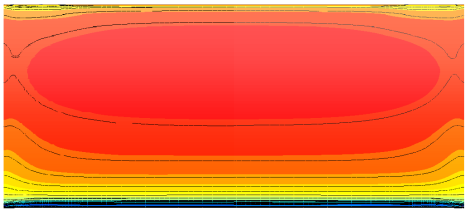
(a) Suction side (*grid I*)



(b) Suction side (*grid II*)



(c) Pressure side (*grid I*)



(d) Pressure Side (*grid II*)

Figure 6.11: Surface pressure distribution for the turbulent compressor cascade ($Re_c = 6.40 \times 10^5$, $M_1 = 0.63$.): $C_p \in [-0.28, 0]$ (suction side) and $C_p \in [-0.2, 0.05]$ (pressure side) using $p = 3$ (left part of the blade) / $p = 4$ (right part of the blade).

grid II reveals however some oscillations of the DG results near the peak value ($x/c \leq 0.4$). Although the oscillations are reduced by increasing the polynomial interpolation order p, they do not totally disappear. The perturbations are even more visible when regarding the shear friction; see Fig. 6.12(d) and

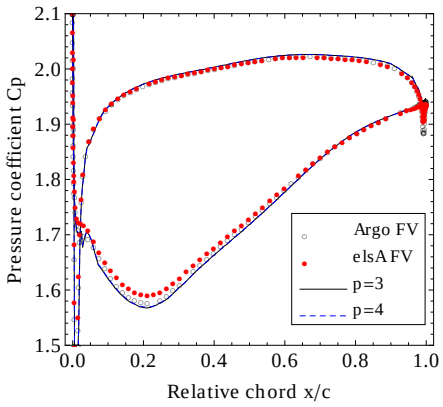
6.12(f). In particular, in addition to the numerical transition, an important peak of the shear friction occurs at $x/c \approx 0.036$ (*grid II*) and a slightly smaller at $x/c \approx 0.039$ (*grid I*). Since the height of the first element layer off the blade is the same for both meshes, they must be related to the chordwise resolution. At first glance, this may surprise, given the rather large number of elements on the blade and the high degree of p. However, the same phenomenon was already encountered for the flow around the 2D NACA 0012 aerofoil (see chapter 5). Due to difficulties to generate high-order ($q \geq 3$) meshes, the cascade geometry is only discretized by a second order polynomial representation $q = 2$. As a consequence, the continuity of the element normal vectors cannot be guaranteed and hence, as p is increased, the mesh discontinuities become resolved, leading to a local vortex generation and the pressure drops observed in Fig. 6.12. According to the previous 2D simulations, much coarser meshes can be used if a polynomial representation of at least $q = 3$ - a necessity to afford the C^1 continuity of the discrete geometry - is employed.

A second difference between both meshes concerns the prediction of the separation bubble, which is sharper for *grid I*; cf. Fig. 6.12(b) and 6.12(d). As illustrated by Fig. 6.13, a flow separation occurs at the trailing edge on the suction side. We should however bear in mind that the shear profiles displayed in Fig. 6.12(b) and 6.12(d) lie within the end-wall boundary layer. This region is characterized by important flow gradients and hence, the profiles highly depend on the spanwise location. For instance, at 5% of the span, the shear profile computed with *grid II* are nearly identical to the one shown in Fig. 6.12(b).

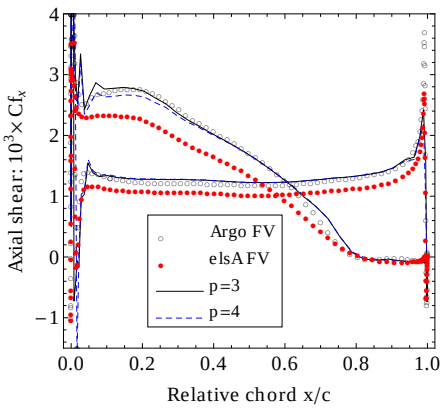
Finally, the total pressure contours at the outlet plane on *grid II* are shown in Fig. 6.14. Despite the relatively large element size, even the $p = 2$ interpolation leads to a smooth variation of the total pressure.

6.3 Concluding remarks

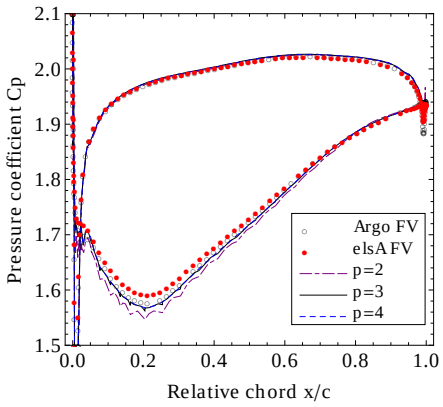
The results of both numerical applications were found to be in good agreement with either the available experimental measurements or other finite volume/element computations. Again the validation study has demonstrated the advantage of a high-order interpolation, leading to superior results than a low-order interpolation on finer meshes. However, the simulations have also illustrated the extremely high sensitivity of the DG method to the mesh quality as p increases. The C^1 continuity of the discrete geometry is therefore of utmost importance, and hence at least a $q = 3$ representation should be used. Finally, to fully benefit from the high-order interpolation, future work should concentrate on the generation of unstructured curved grids. As the compressor cascade has been simulated using extruded 2D meshes, an excessive number of degrees of freedom was required compared to a completely unstructured computation.



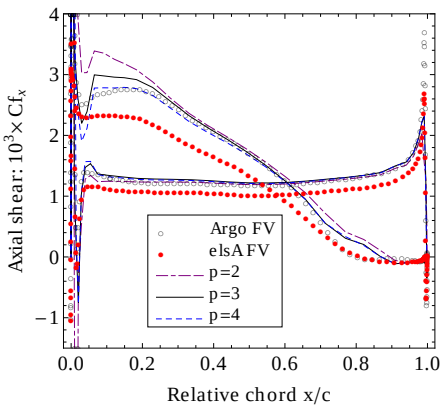
(a) 6% span (*grid I*)



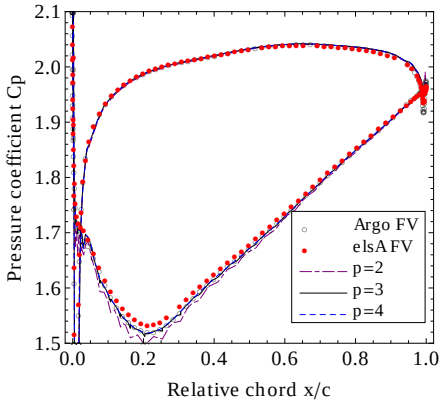
(b) 6% span (*grid I*)



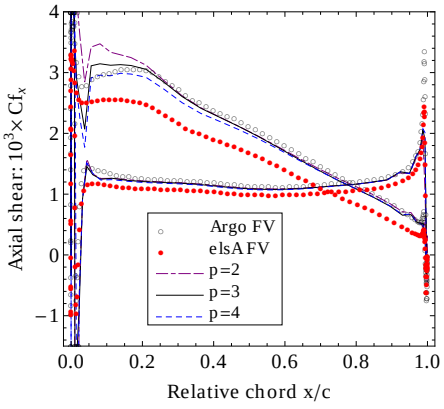
(c) 6% span (*grid II*)



(d) 6% span (*grid II*)

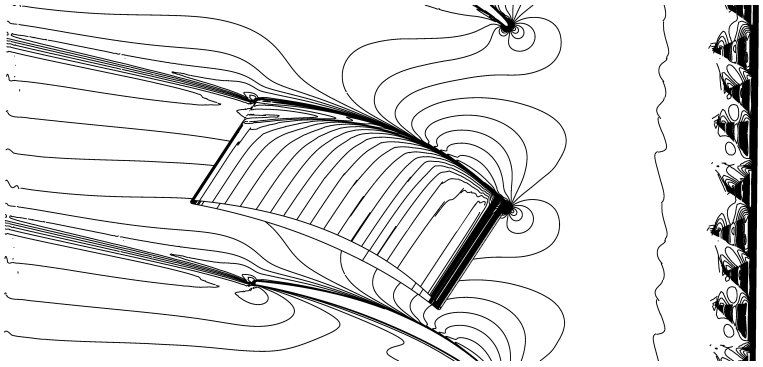


(e) 50% span (*grid II*)

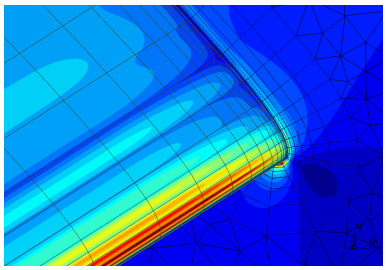


(f) 50% span (*grid II*)

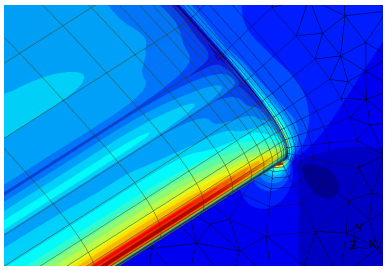
Figure 6.12: Surface pressure distribution and axial shear stress at different spanwise locations for the turbulent compressor cascade ($Re_c = 6.40 \times 10^5$, $M = 0.63$). Especially the shear stress suffers from the absence of C^1 continuity of the discrete aerofoil, as a consequence of the low-order ($q = 2$) geometrical representation.



(a) Global view of the shear stresses computed with *grid II* ($p = 4$)

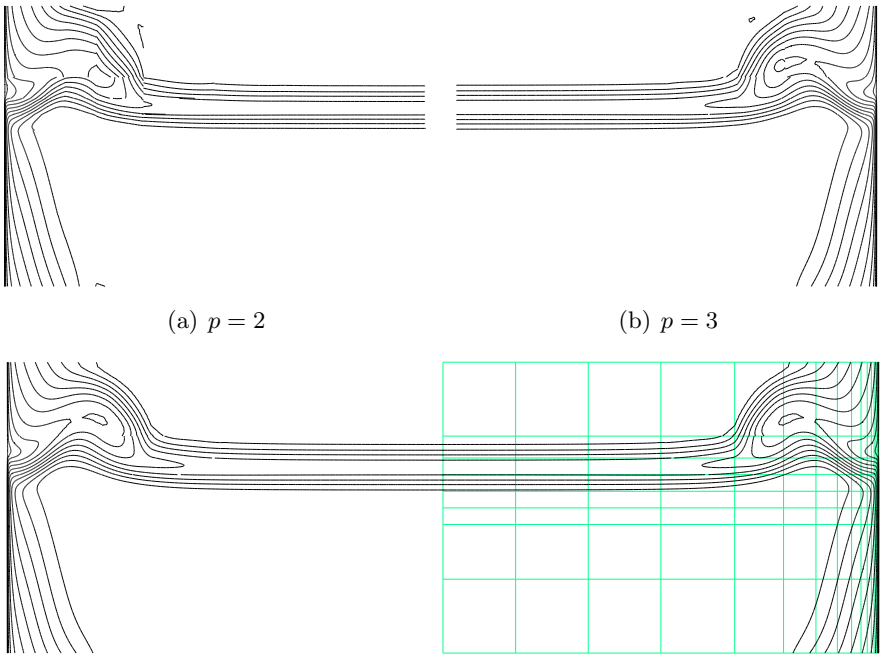


(b) $p = 3$



(c) $p = 4$

Figure 6.13: Norm of the shear stresses for the turbulent compressor cascade ($Re_c = 6.40 \times 10^5$, $M = 0.63$, *grid II*). Despite a significant improvement of the computed friction using $p = 4$ polynomials, an important jump is observed near the leading edge ($x/c = 0.036$). The friction jump is caused by the poor geometrical representation ($q = 2$) leading to discontinuous normal vectors between adjacent elements.



(a) $p = 2$ (b) $p = 3$

(c) $p = 4$

Figure 6.14: Total pressure contours ($p_{tot} \in [5.4 \times 10^4,\ 1.2 \times 10^5]$ Pa, $\Delta = 1467$ Pa) at the outlet plane on *grid II*. The green lines represent the mesh.

Chapter 7

Towards a discontinuous Galerkin LES solver

Contents

This chapter is intended as a first step towards the use of discontinous Galerkin methods for large eddy simulations. The principal aim is (i) to *validate our numerical implementation* of the subgrid-scale (SGS) models described in chapter 2 and (ii) to determine the *resolution requirements* of LES. To this end, two numerical applications are considered.

In section 7.1 we compare the LES results for the *Taylor-Green vortex* at $Re = 1600$ using either the Smagorinsky or the WALE model. The decay of kinetic energy and the dissipation rate constitute useful indicators in order to quantify the (numerical) dissipation and the resolution power of the discretization scheme.

In section 7.2 LES computations of *decaying homogeneous isotropic turbulence* are presented. Besides a mesh convergence study, subgrid filtering as well as the influence of the model coefficient and of the penalty parameter are investigated.

7.1 Nearly incompressible Taylor-Green vortex

The *Taylor-Green vortex* [170] is probably the simplest flow to investigate (i) the creation of small eddies by three-dimensional vortex stretching and (ii) the non-linear transfer of kinetic energy among distinct spatial scales. Hence, at sufficiently high Reynolds numbers, the decay phase, which follows the breakdown of the initial vortex, presents various characteristic features of fully developed turbulence [17]. The evolution of the kinetic energy and the dissipation rate are useful indicators[1] to quantify the accuracy of the numerical discretization scheme.

7.1.1 Flow conditions for the Taylor-Green vortex at $Re = 1600$

The flow is computed within a square box $-\pi \leq x, y, z \leq \pi$ and periodic boundary conditions are imposed in all three space directions. The initial velocity field is defined by a two-dimensional single-Fourier mode of the form:

$$\begin{aligned} u(\mathbf{x}, 0) &= V_0 \sin(x) \cos(y) \cos(z), \\ v(\mathbf{x}, 0) &= -V_0 \cos(x) \sin(y) \cos(z), \\ w(\mathbf{x}, 0) &= 0. \end{aligned} \tag{7.1}$$

The initial temperature field is taken uniform $T(\mathbf{x}, 0) = 1$ whilst the pressure is given by the solution of the pressure Poisson equation, i.e.

$$p(\mathbf{x}, 0) = 1 + \frac{\rho_0 V_0^2}{16} \left[\cos(2x) + \cos(2y)\right] \left[\cos(2z) + 2\right]. \tag{7.2}$$

The velocity is chosen as $V_0 = 0.1$ in order to render the flow nearly incompressible. The resulting Mach number equals $M \approx 0.085$. The Reynolds number is set to $Re = \rho_0 V_0 L / \mu = 1600$ (with $\rho_0 = 1$ and the reference length taken as $L = 1$).

7.1.2 Numerical results

Hereafter, we present the LES simulations performed using $p = 4$ polynomials and two distinct meshes composed of 16^3 and 24^3 structured hexahedra respectively. Turbulence closure is achieved by either the Smagorinsky or the WALE subgrid-scale model; cf. section 2.4.1. The system of ordinary differential equations is integrated in time by means of an explicit 4th order Runge-Kutta scheme. The results are compared to the DNS data obtained by the dealiased pseudo-spectral code developed at Université catholique de Louvain (UCL) [53, 66]. The DNS reference solution is computed on a 512^3 grid and hence captures exactly all Fourier modes up to the 256th harmonic with

[1] The Taylor-Green vortex is one of the test cases proposed by the 1^{st} *International Workshop on High-Order CFD Methods* [61].

respect to the domain length. Using a $p = 3$ interpolation, DG simulations on a 96^3 mesh were found to be already in good agreement with the spectral results; on a 128^3 hexahedra mesh full DNS resolution is reached. The breakdown of the initial vortex into successively smaller structures is illustrated in Fig. 7.1.

The effect of SGS filtering on the time evolution of the integrated[2] kinetic energy E in case of the coarse 16^3 mesh is shown in Fig. 7.2(a) . To this end the resolved flow field is projected onto low-order $p = 2$ polynomials by a Lagrange interpolation. Whereas the standard LES ("all-all") is clearly too dissipative, the filtered Smagorinsky model ("small-small") agrees well with the DNS data. Similar results have been observed for the WALE model. Compared to the coarse mesh, the fine 24^3 mesh slightly improves the prediction of the kinetic energy for $6 \leq t \leq 10$. At this phase, an important part of the energy is confined to the small structures. Note also that even an implicit LES (ILES), without any SGS model, performs quite well on the 24^3 grid, see Fig. 7.2(b).

Since no numerical scheme can - with a finite number of grid cells - capture the singularity of the Taylor- Green vortex core, the temporal evolution of the enstrophy $\mathcal{E}(t)$ constitutes a good indicator to measure the *resolving capability* of the numerical scheme [159]. The enstrophy is defined as

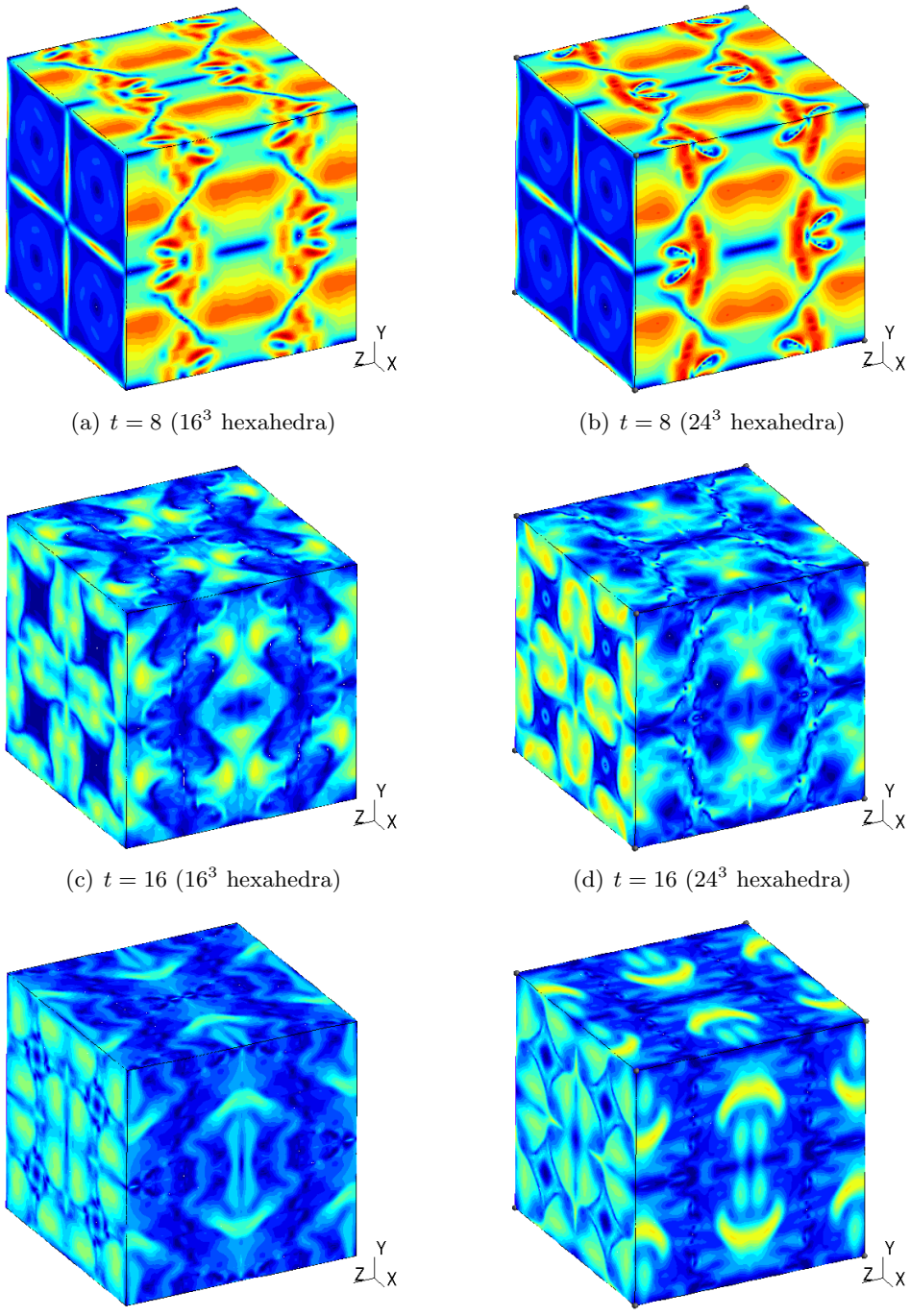
$$\mathcal{E}(t) \equiv \frac{1}{\Omega} \int_{\Omega} \boldsymbol{\omega}\boldsymbol{\omega} \, d\Omega, \tag{7.3}$$

where $\boldsymbol{\omega}$ denotes the vorticity. For incompressible flows, the enstrophy is furthermore directly related to the dissipation rate $\epsilon(t)$ of the kinetic energy by $\mathcal{E} = \epsilon/(2\mu)$. To further investigate the effect of the SGS model on the resolved scales, we compare the decay of the kinetic energy $-dE/dt$ to the dissipation rate $\epsilon(t)$ computed as the sum of a resolved $\epsilon_{resolved}$ and a SGS ϵ_{sgs} contribution, i.e.

$$\epsilon = \underbrace{\frac{1}{\Omega}\int_{\Omega} \tau_{ij} s_{ij} d\Omega - \frac{1}{\Omega}\int_{\Omega} p \nabla \mathbf{u} d\Omega}_{\epsilon_{resolved}} + \underbrace{\frac{1}{\Omega}\int_{\Omega} \tau_{ij}^{sgs} \tilde{s}_{ij}^{s} d\Omega}_{\epsilon_{sgs}}. \tag{7.4}$$

Since a low Mach number has been chosen in order to facilitate the comparison with the DNS results, compressibility effects are negligible and thus the contribution of the pressure work $p\nabla\mathbf{u}$ is small in (7.4). The relative importance of the resolved and the SGS dissipation for the unfiltered Smagorinsky model is illustrated in Fig. 7.3. A detailed comparison between the dissipation rate ϵ and the decay of the kinetic energy $-dE/dt$ for several (filtered) SGS models is shown in Fig. 7.4 (16^3 mesh) and Fig. 7.5 (24^3 mesh). At the beginning, the dissipation rate is small, because the flow is composed of a "single" large Fourier mode. The vortex stretching due to the non-linear terms generates successively smaller scales, which increases ϵ. At $t \approx 8$, an important part of

[2] $E(t) = \frac{1}{\Omega}\int_{\Omega} \frac{1}{2}\rho \|\mathbf{u}\|^2 d\Omega$



(a) $t = 8$ (16^3 hexahedra)

(b) $t = 8$ (24^3 hexahedra)

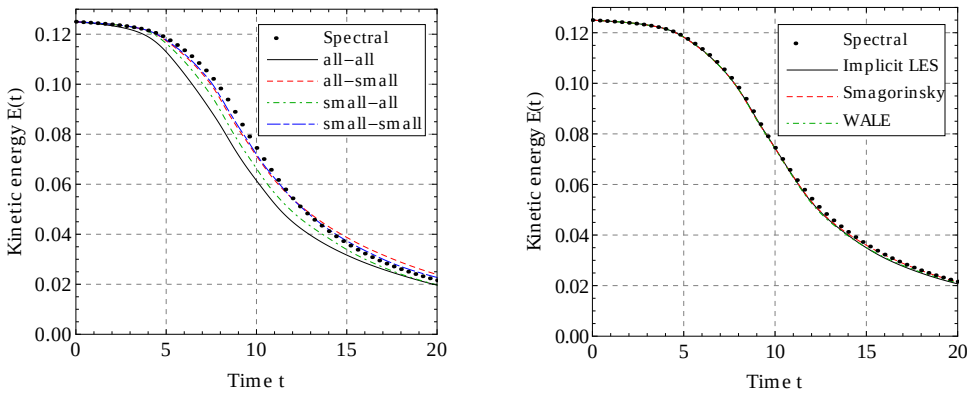
(c) $t = 16$ (16^3 hexahedra)

(d) $t = 16$ (24^3 hexahedra)

(e) $t = 20$ (16^3 hexahedra)

(f) $t = 20$ (24^3 hexahedra)

Figure 7.1: Time evolution of the velocity norm $\|\mathbf{u}\|$ for the Taylor-Green vortex at $Re = 1600$ ($p = 4$, "small-small" Smagorinsky model using a 2nd order Lagrange filter). $\|\mathbf{u}\|$ is between 0 (blue) and 0.1 (red). At $t \approx 8$, the dissipation of the kinetic energy reaches its maximum.



(a) 16^3 hexahedra (Smagorinsky model) (b) 24^3 hexahedra (filtered SGS models)

Figure 7.2: Time evolution of the normalized kinetic energy $E(t)/\Omega$ for the Taylor-Green vortex at $Re = 1600$ ($p = 4$). Whereas the standard model is clearly too dissipative, the subgrid-scale filter significantly improves the LES results, which are in good agreement with the DNS data even on coarse meshes.

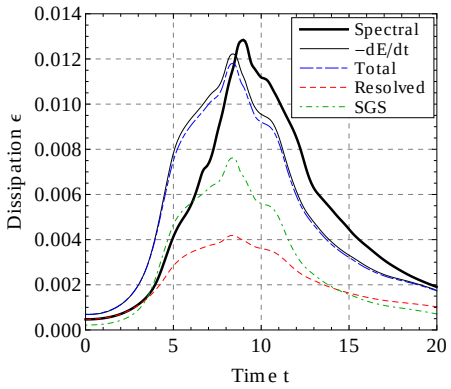


Figure 7.3: Comparison between the resolved and the subgrid-scale dissipation for the Taylor-Green vortex at $Re = 1600$ ($p = 4$, 16^3 hexahedra). The SGS stress tensor is modelled by the unfiltered Smagorinsky model. The "total" dissipation rate is computed according to (7.4) as the sum of the resolved and the SGS dissipation. The discrepancy with $-dE/dt$ is related to the numerical dissipation of the method.

the kinetic energy is contained in the small eddies, leading to a peak of the dissipation rate. The ensuing decrease of ϵ is a consequence of the growing amount of (kinetic) energy that has been removed by viscous effects.

According to our numerical experiments, the Smagorinsky and the WALE model lead to similar results for pure vortical flows such as the Taylor-Green

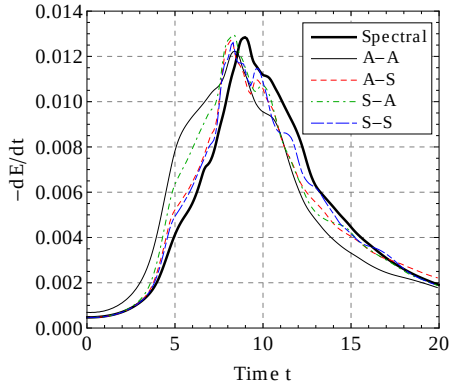
problem. In particular, the standard unfiltered variant ("all-all") of both models is too dissipative at the beginning of the simulation, when the flow is still laminar; cf. also Fig. 7.3. Separating the resolved scales significantly improves this behaviour. In addition, filtering the strain rate tensor $\tilde{s}_{ij}$ seems more effective than computing the subgrid-scale viscosity by the small scales only. Indeed nearly identical dissipation rates are observed for the "small-small" and the "all-small" models, whereas the time evolution of the "small-all" model resembles the unfiltered one.

Fig. 7.4 and Fig. 7.5 demonstrate the overall good agreement between the LES computations and the reference DNS. Whereas the peak value of dE/dt is comparable for all computations, the maximum is slightly shifted if the coarse 16^3 mesh is used. It is important to notice the discrepancy between $-dE/dt$ and the dissipation rate $\epsilon(t)$ computed by (7.4). Whereas the maxima of both quantities are nearly identical for the standard unfiltered LES (see Fig. 7.3), they differ in case of filtered and implicit LES. The differences are due to the numerical dissipation of the discretization scheme. In particular, the penalty term and the transpose term are not included in (7.4). Both increase the numerical dissipation, if the flow field is underresolved by the polynomial approximation. It is therefore not surprising that the differences are the most visible close to the peak dissipation, since at this time a significant part of the energy is contained in the badly resolved small scales. In chapter 5, we have seen that including the penalty term significantly improves the predicted shear friction. The present results suggest, that a similar approach should be applied to estimate the dissipation rate.

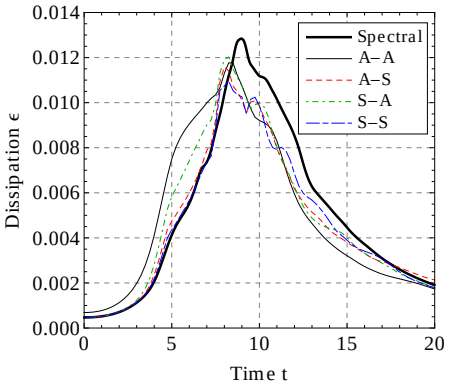
Finally, the influence of the filter order $\tilde{p}$ on the dissipation rate $\epsilon(t)$ is shown in Fig. 7.6. The computations are performed on the coarse 16^3 mesh using the filtered Smagorinsky model. The filter order ranges from $\tilde{p} = 2$ to $\tilde{p} = p = 4$. The latter case corresponds to an implicit LES, since all resolved eddies are included in the large-scale space and the SGS model - based on the high wavenumber modes - is thus inactive. As expected, the higher the filter order $\tilde{p}$, the lower the "physical" dissipation rate $\epsilon(t)$ (7.4). However, because for higher $\tilde{p}$ an increasing part of the kinetic energy is removed by the numerical dissipation of the scheme, the time derivative dE/dt remains similar in all three cases. The main drawback of implicit LES is that we cannot control how the energy is dissipated. The influence of filtering on the energy spectrum is further investigated in section 7.2, by means of the *decaying homogeneous isotropic turbulence (HIT)*.

7.2 Decay of homogeneous isotropic turbulence

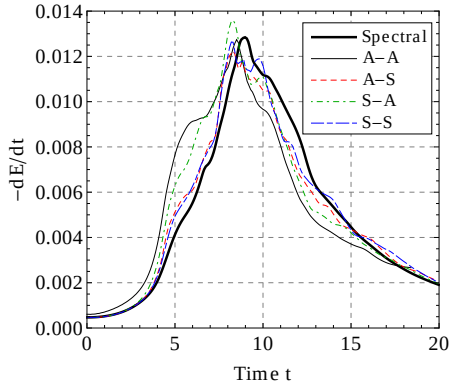
In this section we present simulations of decaying homogeneous isotropic turbulence within a periodic box of size 2π; see Fig. 7.7. An initial correlated turbulent flow field with a specified energy spectrum is generated using the



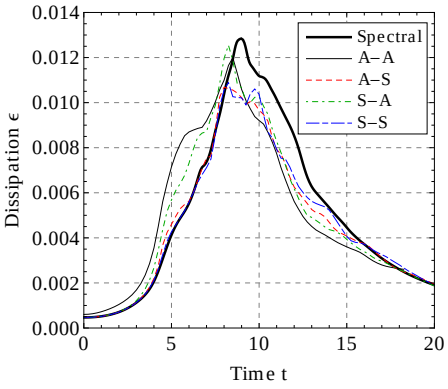
(a) Energy decay rate: Smagorinsky



(b) Dissipation: Smagorinsky

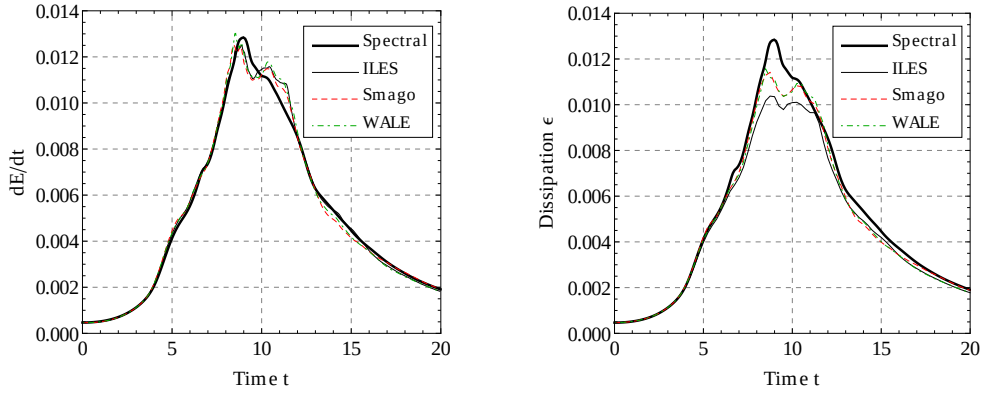


(c) Energy decay rate: WALE



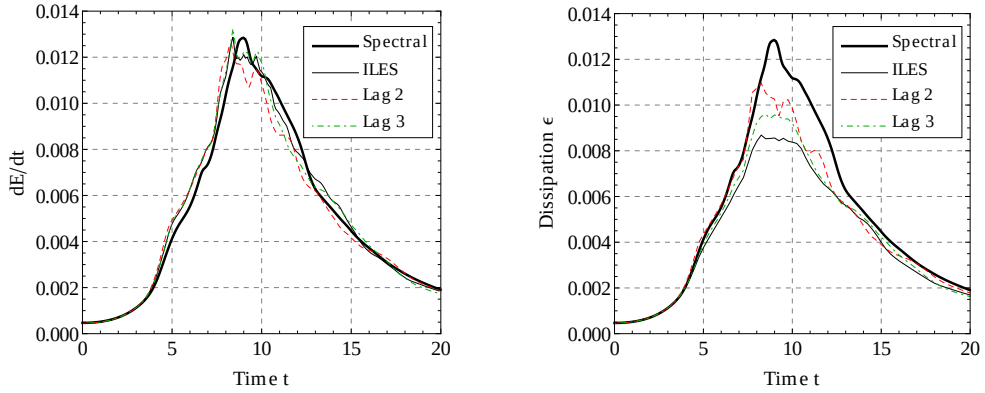
(d) Dissipation: WALE

Figure 7.4: Time evolution of the normalized energy dissipation rate for the Taylor-Green vortex at $Re = 1600$ ($p = 4$, 2nd order Lagrange filter, 16^3 *hexahedra*). The abbreviations "A" and "S" refer to filtering and denote "all" and "small", respectively. Whereas the amplitude of the (numerical) dissipation peak dE/dt is rather well predicted, the location of the maximum is shifted compared to the DNS reference.



(a) Energy decay rate (b) Dissipation

Figure 7.5: Time evolution of the normalized energy dissipation rate for the Taylor-Green vortex at $Re = 1600$ ($p = 4$, 2nd order Lagrange filter, 24^3 *hexahedra*). Compared to the coarse 16^3 mesh, the 24^3 mesh improves the prediction of the time, when the dissipation rate reaches its maximum.



(a) Energy decay rate (b) Dissipation

Figure 7.6: Comparison of different filter orders $\tilde{p}$ for the Taylor-Green vortex at $Re = 1600$ ($p = 4$, Smagorinsky model, Lagrange filter, 16^3 *hexahedra*). The higher the filter order, the more important the contribution of the numerical dissipation, which might explain a similar evolution of dE/dt independently of $\tilde{p}$.

de-alised pseudo-spectral code developed at Université catholique de Louvain [53,66]. The spectrum is purely solenoidal (divergence-free) and does not present any fluctuations of the thermodynamic variables. The corresponding Taylor microscale Reynolds number equals $Re_\lambda \approx 136$. To initialize the DGFEM simulations we first truncate the velocity field to $p \times N/2$ Fourier modes. Here, p and N are the degree of the polynomial approximation and the number of elements per coordinate direction respectively. Based on this field a truncated pressure is computed. Finally, both fields are projected on the high-order DG nodes using a Lagrangian interpolation.

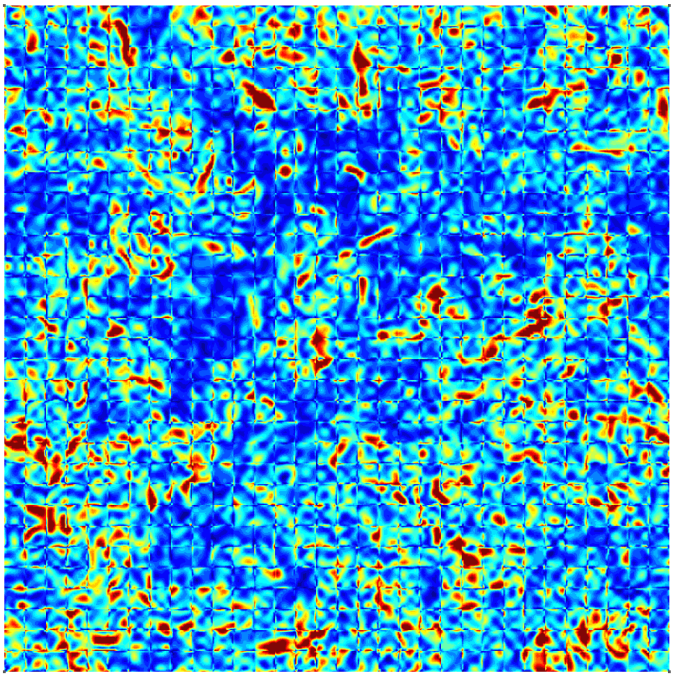


Figure 7.7: Vorticity magnitude of the initial homogeneous isotropic turbulence spectrum truncated to 64 Fourier modes ($p = 4$, 32^3 hexahedra).

Since in the absence of solid walls similar results are expected for the Smagorinsky and the WALE model (cf. section 7.1), only the former is considered here. The compared methods are implicit LES, the standard/unfiltered Smagorinsky model as well as different variants of the filtered VMS model. To validate our LES solver, the decay of the kinetic energy is benchmarked against DNS results computed with the pseudo-spectral code on a 512^3 mesh, thus representing exactly 256^3 Fourier modes. As for the Taylor-Green vortex studied in the previous section, the system of differential equations is integrated in time by an explicit 4th order Runge-Kutta scheme. All computations are performed using a CFL number of $CFL = 1$ and a maximum allowable time step of $\Delta t = 0.001$. The latter was shown to deliver time accurate results by an a

priori convergence study[3].

Furthermore, the resolution requirements are investigated by three structured meshes involving 8, 16 and 32 hexahedra per coordinate direction, respectively. The degree of the polynomial interpolation is set to $p = 4$.

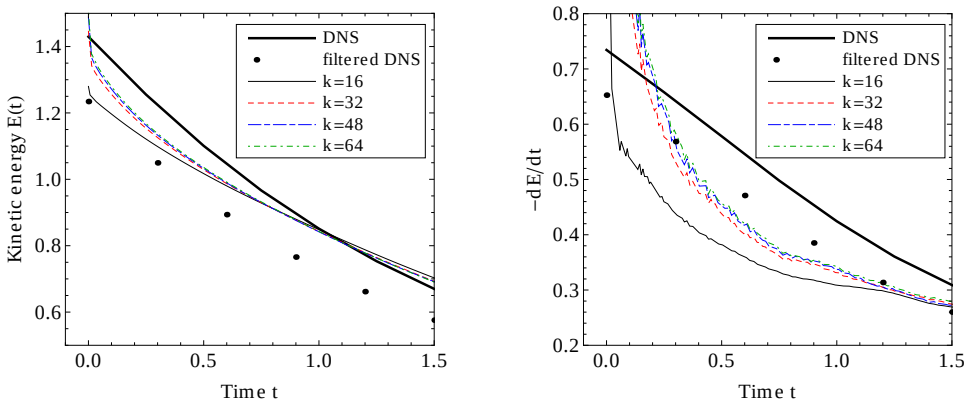
7.2.1 Implicit LES of homogeneous isotropic turbulence

Before validating the LES solver, we start with some preliminary ILES computations (without any SGS model). The aim is to determine the influence of the truncation of the initial spectrum as well as the role played by the penalty term.

Truncation of the initial spectrum. As the computational grid is too coarse to resolve all turbulent scales of the DNS spectrum, an initial solution is generated by truncating the DNS flow field (velocity and pressure) to the large Fourier modes. The resulting field is then projected onto the DG shape functions using a Lagrange interpolation. Although the Lagrange projection acts itself as a low-pass filter, the a priori cut-off in the spectral space provides a better control of the resolved scales and hence facilitates the comparison with filtered DNS. Indeed, since the Lagrange interpolation does not constitute a top-hat filter in the spectral space, the question arises of how many Fourier modes should be included in the truncated reference DNS in order to be comparable to a DG approximation of polynomial degree p .

The decay of the kinetic energy $E(t)$ and the corresponding dissipation rates dE/dt on the coarse mesh $N = 8$ using different truncated initial fields are compared in Fig. 7.8. The filtered DNS includes $k^3 = (p \times N/2)^3 = 16^3$ modes. If $k = p \times N/2$ the integrated (kinetic) energy $E(0)$ is very close to the filtered DNS results and the initial energy spectrum seems thus adequately resolved; see Fig. 7.8(a). The deficiency of the Lagrange interpolation becomes obvious by pushing the cut-off frequency towards larger wavenumbers. Indeed, if $k > 16$, we observe an important drop of the kinetic energy at the very beginning of the computation. This indicates that the resulting initial flow field is under-resolved. ILES computations with a finer $N = 16$ mesh confirmed these observations. In the reminder we therefore choose $k = p \times N/2$ as the standard initial solution. Fig. 7.8(a) also shows that the energy decays much more slowly than the one of the reference DNS. We conclude that the numerical dissipation of the discretization scheme is insufficient and that a SGS model must be used. Finally, the time derivative dE/dt is compared to the DNS data in Fig. 7.8(b). All derivatives are calculated using a 2nd order accurate finite difference approximation. Contrary to the Taylor-Green vortex (see section 7.1), the time derivatives are not smooth. We had observed a similar phenomenon if the initial pressure field of the Taylor-Green problem

[3]On the $N = 16$ mesh, 4 to 5 intermediate time iterations are necessary at each time step in order to respect the CFL condition.



(a) Decay of the kinetic energy

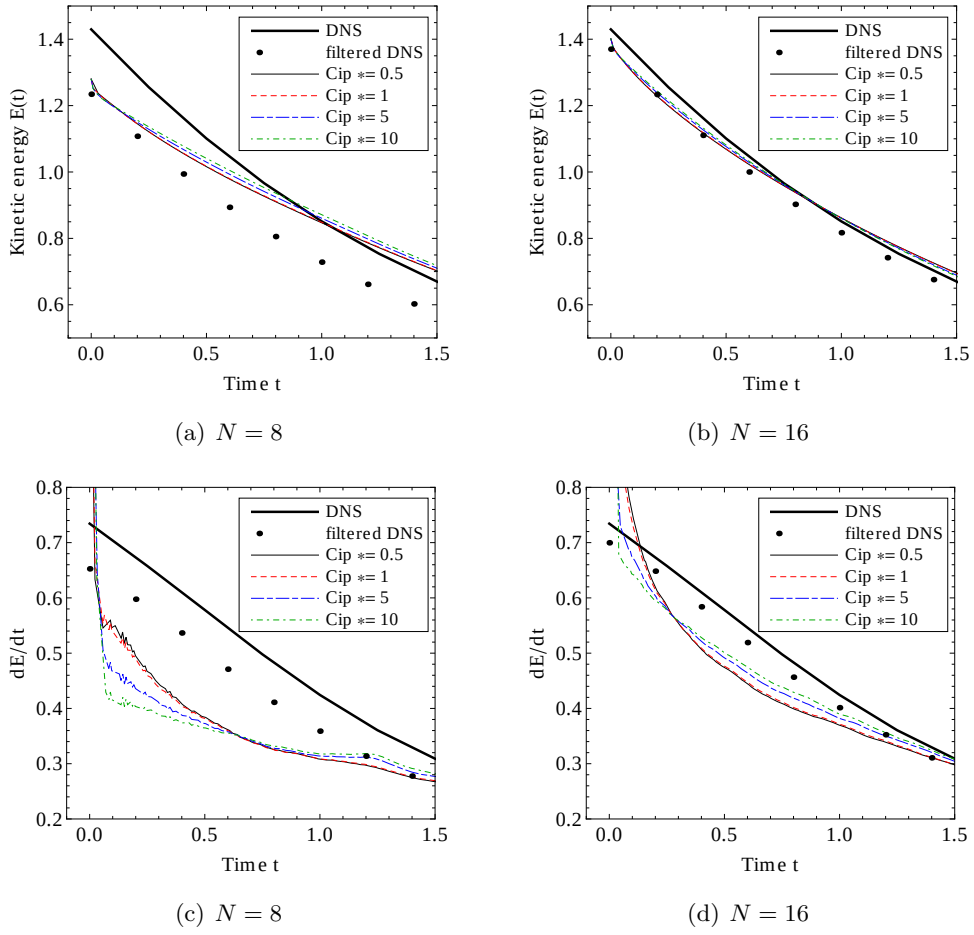
(b) Dissipation rate

Figure 7.8: Influence of the initial (truncated) solution on the decay of the kinetic energy $E(t)/\Omega$ and on the dissipation rate for the homogeneous, isotropic turbulence ($p = 4$, $N = 8$). No SGS-model is used for these simulations. The filtered DNS is truncated to $k = 16$ Fourier modes. The fluctuations in the time derivative dE/dt are probably due to an "incompatibility" between the projected velocity and pressure fields.

does not satisfy the pressure Poisson equation. We therefore suppose that the oscillations are related to an "incompatibility" between the velocity and the pressure fields. In fact, although the DNS velocity field is purely solenoidal, we cannot guarantee that the same holds for the truncated, projected velocity. In addition, there is no reason for the projected pressure to satisfy the Poisson equation. As expected, the fluctuations disappear once the flow has been sufficiently smoothed by viscous effects. Moreover, they have not been noticed on the finer meshes $N = 16$ or $N = 32$. In [173] van der Bos and Geurts made use of an alternative *face-based* projection in order to smooth the initial field compared to the *element-based* projection employed in the current work.

Influence of the penalty parameter. In chapter 5 we have investigated the influence of the penalty parameter δ within the context of two-dimensional RANS computations. Our numerical studies have shown that although the inclusion of the penalty term significantly improves the computed wall friction, the latter is insensitive to value of δ when the penalty term is included. In [173] van der Bos and Geurts have shown that the numerical dissipation has a similar effect as the SGS model (see also Grinstein *et al.* [75]). This observation has been confirmed numerically by the Taylor-Green problem in section 7.1. Since the dissipation of the IP formulation is directly related to the interface jumps, the question arises in which measure it is influenced by the penalty parameter. To answer this question we have performed several ILES computations of decaying isotropic turbulence using distinct values of δ ranging from 0.5 to 10

times the reference value; see Fig. 7.9. To better emphasize the influence of the penalty term, very coarse meshes ($N = 8$ and $N = 16$) are adopted.

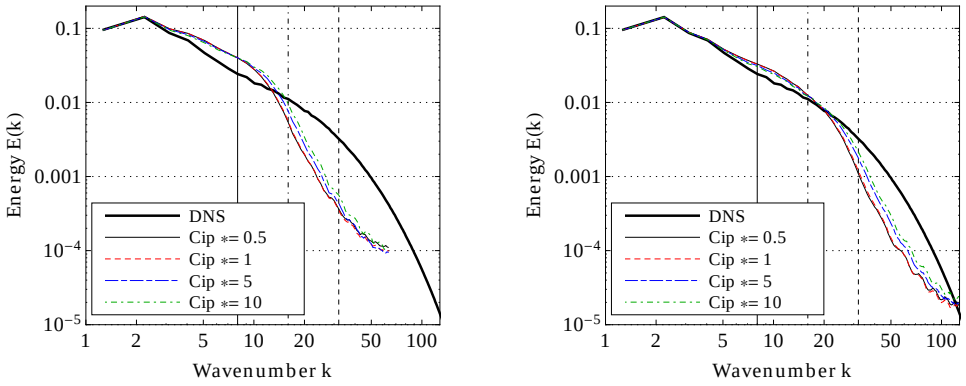


(a) $N = 8$ (b) $N = 16$

(c) $N = 8$ (d) $N = 16$

Figure 7.9: Influence of the penalty parameter δ on the decay of the kinetic energy $E(t)/\Omega$ and on the dissipation rate for homogeneous, isotropic turbulence ($p = 4$). No SGS-model is used for these simulations. The filtered DNS is truncated to $k = p \times N/2$ Fourier modes.

As illustrated by Fig. 7.9(a) and 7.9(c), increasing the penalty parameter results in a faster smoothing of the flow during the first time steps, if the simulation is poorly resolved ($N = 8$). After this short transition, the kinetic energy evolves quite similarly whereby the amount of energy remains slightly higher for larger values of δ. On the intermediate grid $N = 16$ (see Fig. 7.9(b) and 7.9(d)), the results are almost insensitive to the penalty parameter. Furthermore, as already suspected by the decay of the kinetic energy, the penalty parameter has only a minor effect on the energy spectrum; see Fig. 7.10. The latter will be further discussed in the next section. In conclusion, changing the



(a) $N = 8$ (b) $N = 16$

Figure 7.10: Influence of the penalty parameter δ on the kinetic energy spectrum for homogeneous isotropic turbulence ($p = 4$). No SGS-model is used for these simulations. The spectra are computed at $t = 1$ by a fast Fourier transform (FFT) using a structured grid with $4 \times p \times N$ points in each direction.

jump penalization does not allow - in any of the presented cases - to modify the numerical dissipation in order to recover the correct dissipation rate. The use of a SGS model is thus obligatory.

7.2.2 Explicit LES of homogeneous isotropic turbulence

The time zero initial condition is the same as for the previous ILES computations, i.e. a truncation of the developed DNS turbulence field including $p \times N/2$ Fourier modes. Figure 7.11 shows the evolution of the Reynolds number Re_λ based on Taylor microscales over the course of the simulation.

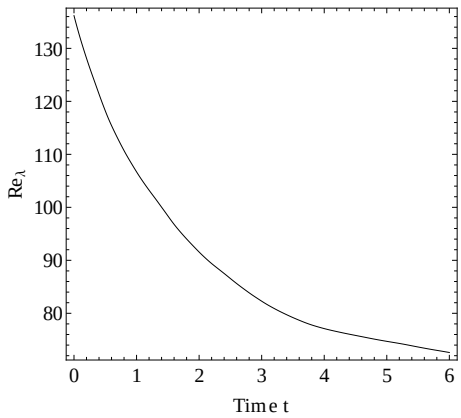


Figure 7.11: Time evolution of the Taylor microscale Reynolds number Re_λ for the decay of homogeneous isotropic turbulence.

Resolution and spectra. In the following we present the convergence study that has been realized to determine the resolution requirements for LES computations of homogeneous isotropic turbulence using a 5th order accurate ($p = 4$) DGFEM discretization. Thereto, we performed several computations using distinct structured meshes with $N = 8$, $N = 16$ and $N = 32$ hexahedra in each coordinate direction, respectively. The compared methods are the standard (unfiltered) Smagorinsky model as well as different variants of the VMS approach[4]. In the latter case, scales are separated by a Lagrange projection of the resolved field onto lower-order polynomials of degree $\tilde{p} = 2$.

The decay of kinetic energy $E(t)$ and its corresponding time derivative are presented in Fig. 7.12. It is important to notice that at $N = 32$, there is virtually no difference between the DNS and the filtered DNS obtained by a top-hat filter in the spectral space. Accordingly, any simulation, that accurately computes the first 64 Fourier modes, lies in the DNS rather than in the LES range. Even at $N = 16$, differences between filtered and unfiltered DNS are small. However, so far we only have assumed that a polynomial interpolation of degree p is able to accurately represent up to $p \times N/2$ Fourier modes. Numerical dispersion and dissipation errors inherent to the DG scheme and the Lagrange interpolation could make (at least) the $N = 16$ mesh unsuitable for DNS.

A comparison of the different SGS approaches shows that the standard Smagorinsky model ("all-all") is again the most dissipative and leads to an excessive energy decay at the beginning of the computation. Beyond $t \approx 1.5$ it matches the DNS data quite well. Although most pronounced on the coarse mesh, this behaviour is independent of the grid resolution. A similar observation has been reported by Hughes *et al.* [91]. In contrast, without any SGS model (cf. ILES) the numerical dissipation is insufficient to prevent the energy from being accumulated in the small scales. The results of the "all-small" and "small-small" VMS methods are very close and - at least at the early stages - in good agreement with the DNS data. Whilst the initial slope coincides with the one of the DNS data, it deviates beyond $t \approx 1$ for $N = 8$. Although the dissipation rate - calculated here as the time derivative of the kinetic energy $-dE/dt$ - does not compare as favorable, it globally agrees with the DNS results.

The kinetic energy spectra at $t = 1$ are presented in Fig. 7.13. Whereas the LES results are close to the DNS spectrum in the low frequency range, they systematically overestimate the energy content of the small "resolved" scales. The origin of this phenomenon is not yet fully understood but might be related to the projection of the truncated DNS field at the beginning of the computation. Indeed as illustrated by Fig. 7.14, the initial Lagrange projection slightly alters the wavenumbers beyond $k \geq N$. Although the energy excess

[4]Part of these simulations, and notably all presented $N = 32$ results, have been realized by R. Luccioni during his internship at Cenaero.

persists on the fine mesh, its amount is much smaller, which shows that the LES results (rapidly) converge to the DNS data as the mesh is refined.

Finally, it is apparent that the standard Smagorinsky model is again the most dissipative, affecting not only the small but all resolved scales. In contrast, the high-frequency spectrum of the implicit LES is essentially determined by numerical errors[5]. The error pile-up in the high-frequency range is a consequence of the small numerical dissipation of the present 5th order accurate DGFEM discretization. The "all-small" and "small-small" VMS methods lead to nearly identical energy spectra.

Filtering. In the previous study, the resolved scales were separated by projecting the polynomial space of degree $p = 4$ onto a subspace of lower-degree $\tilde{p} = 2$ using a Lagrange interpolation. Here, the *Lagrange* and the *L2* filters are compared and the influence of $\tilde{p}$ on the VMS method is investigated. Obviously, the lower $\tilde{p}$ the more wavenumbers are included in the small scale space and hence, the more important is the SGS modelling.

Fig. 7.15 shows the decay of kinetic energy and the corresponding energy spectra at $t = 1$ using the "small-small" Smagorinsky model with grid $N = 16$. On the one hand, neither the decay rate nor the energy spectrum are significantly affected by the filter type, which both lead to nearly identical results. On the other hand, increasing the filter order $\tilde{p}$ slightly improves the initial slope of $E(t)$ which matches the DNS results very well. Beyond $t \geq 0.75$, the decay rate is the same for all cases but diverges from the DNS data. Regarding the energy spectra, increasing the filter order to $\tilde{p} = 3$ shifts the cut-off frequency from $k \approx 16$ to $k \approx 20$. Whereas the tendencies are the same, the influence of the filter type/order is more pronounced as the mesh resolution is decreased ($N = 8$). According to the work of Sengupta *et al.* [156] who used a spectral multidomain method with $p \geq 8$, the choice of the filter is more visible on the pressure and density fluctuations than on the kinetic energy decay. For the influence of subgrid-scale filtering on mean velocity profiles and shear stresses we refer to the validation study of Blackburn and Schmidt [15].

Influence of the Smagorinsky constant. To finish the discussion of the homogeneous isotropic turbulence, we briefly investigate the influence of the Smagorinsky coefficient C_s. In this context, we want to recall that a constant value of $C_s = 0.027$ was used for all simulations, independently of the subgrid filter. However, there is no a priori reason why this choice should be optimal for both, filtered and unfiltered, models. Indeed, in [91], Hughes *et al.*

[5]Care should be taken by interpreting the very high frequency range. As the spectra are computed by a Fast Fourier transform using a finite number of grid points, the last wavenumbers are spoiled by numerical errors. Results in this range should therefore be viewed as qualitative rather than quantitative.

estimate the "small-small" model coefficient to be approximately 1.28 times larger than for standard LES. Furthermore, the correction of the characteristic grid spacing defined as $\Delta = (\Delta V_T)^{1/3}/p$ in order to take into account the polynomial degree p of the solution was also chosen somewhat arbitrarily[6]. Recently, van der Bos and Geurts [173] have realized a computational error analysis of the Smagorinsky constant using standard LES in combination with a low-order $p = 1$ and $p = 2$ DGFEM discretization.

In order to asses a "good" value of C_s for the "small-small" model, we performed several computations with C_s ranging from 0.018 to 0.054. The filtering is achieved by a Lagrange projection onto low-order polynomials of degree $\tilde{p} = 2$. The decay of the resolved kinetic energy and the associated spectra at $t = 1$ are presented in Fig. 7.16 and Fig. 7.17, respectively. It may be seen that increasing C_s mainly affects the initial slope but has only little effect on the further decay rate. Accordingly, the choice of C_s seems to be less critical than what could be expected. Furthermore, even the standard value $C_s = 0.027$ calibrated for unfiltered LES provides a good first estimate for filtered SGS models. Although the LES computations still overestimate the energy content of the "resolved" scales, the spectra clearly show the effectiveness of the filtering, which restricts the modelling to the small eddies. Indeed, while increasing the SGS dissipation attenuates the high-frequency spectrum, it does not affect the small wavenumbers in the range $k \leq N$.

7.3 Conclusions

To validate our implementation of the Smagorinsky and the WALE subgrid-scale model we have studied two numerical applications: the Taylor-Green vortex ($Re = 1600$) and decaying homogeneous, isotropic turbulence ($Re_\lambda \approx 136$). The compared methods are implicit LES, standard unfiltered LES and different VMS methods. All results are benchmarked against DNS data computed by a pseudo-spectral code involving 512^3 grid points.

A grid convergence study has shown that the LES results rapidly converge to the DNS data as the mesh is refined. Whereas the filtered ("small-small" and "all-small") VMS methods generally were in good agreement with the reference DNS, the unfiltered ("all-all") model was found to be too dissipative, leading to an excessive decay of the kinetic energy at the beginning of the simulation. Despite the overall good agreement, a detailed analysis of the kinetic energy spectra revealed a systematic energy excess in the small "resolved" scales. The origin of this phenomenon remains unclear and will be further investigated.

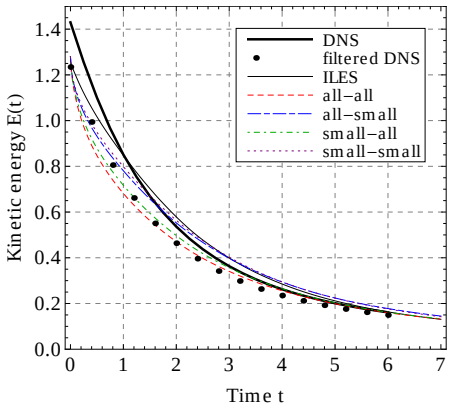
Although the penalty term has a direct influence on the numerical dissipation of the IP method, the (implicit) LES simulations are insensitive to the

[6]Here, V_T denotes the cell volume.

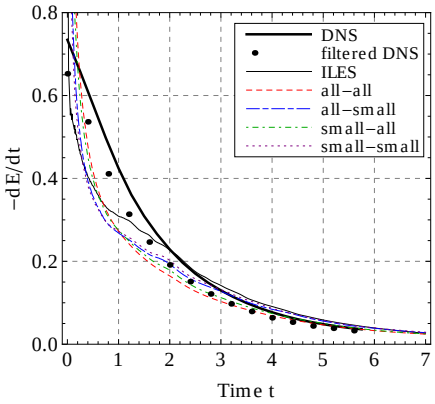
value of the penalty parameter δ. As a consequence, our numerical studies suggest that tuning δ does not replace turbulence modelling.

Whereas we have not seen any (significant) difference in the decay rate and in the energy spectra using either the Lagrange or the L2 projection, both filters effectively restrict the turbulence modelling to the small eddies, as demonstrated by changing the Smagorinsky constant C_s.

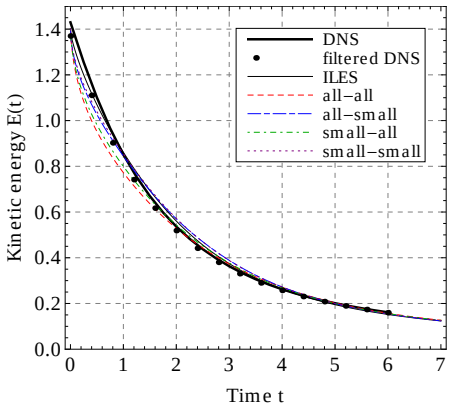
In a future work, the next validation step of the presented LES solver is its application to wall-bounded flows. One of the most popular benchmarks in DNS and LES literature is the turbulent channel flow, e.g. [96, 119, 122]. Furthermore, a major drawback of the presented LES method is the a priori definition of the model constant in the subgrid terms. Although a value of $C_s = 0.027$ turned out to be a good choice for the decay of homogeneous, isotropic turbulence ($p = 4$), a different value may be optimal for other applications, filters, interpolation order. To overcome this limitation, a dynamic procedure [69] to evaluate the model constant should be implemented. Finally, more sophisticated filters which provide a sharper cut-off in the spectral space should be investigated.



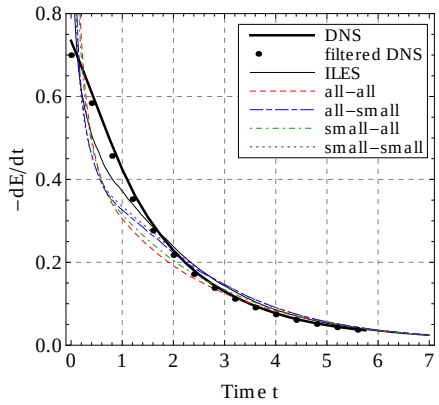
(a) $N = 8$



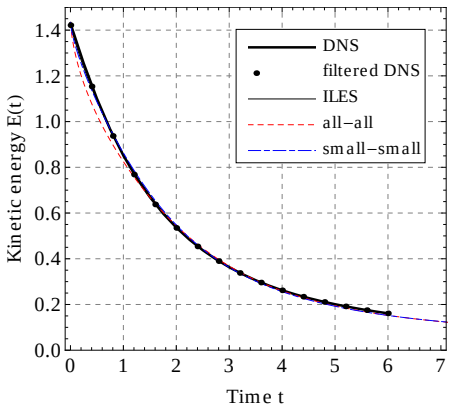
(b) $N = 8$



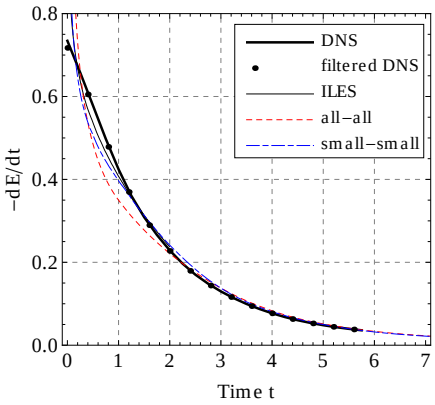
(c) $N = 16$



(d) $N = 16$

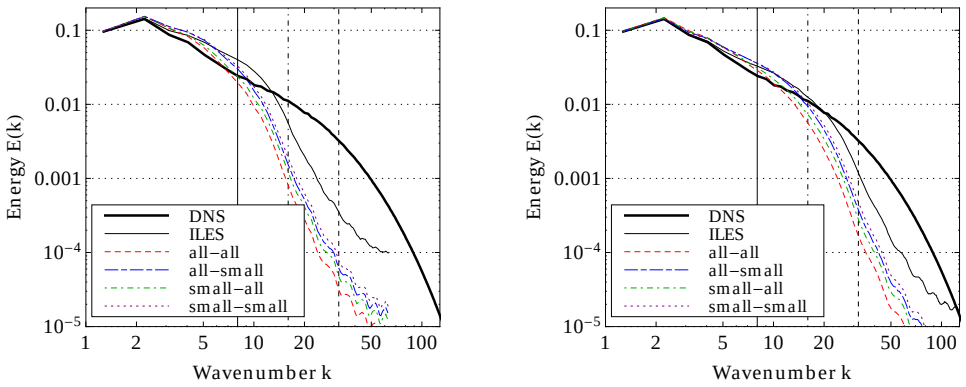


(e) $N = 32$

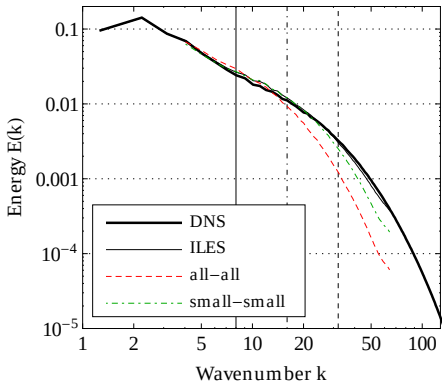


(f) $N = 32$

Figure 7.12: Time evolution of the kinetic energy $E(t)/\Omega$ and its dissipation rate for decaying homogeneous isotropic turbulence, $p = 4$. The filtered DNS is truncated to $k = p \times N/2$ Fourier modes.



(a) $N = 8$ (b) $N = 16$



(c) $N = 32$

Figure 7.13: Comparison of the kinetic energy spectra for decaying homogeneous isotropic turbulence at $t = 1$. The spectra are calculated by a fast Fourier transform. To this end, the $p = 4$ approximation is projected on a structured grid with $4 \times p \times N/2$ ($N = 8$ and $N = 16$) and $p \times N$ ($N = 32$) nodes in each direction, respectively.

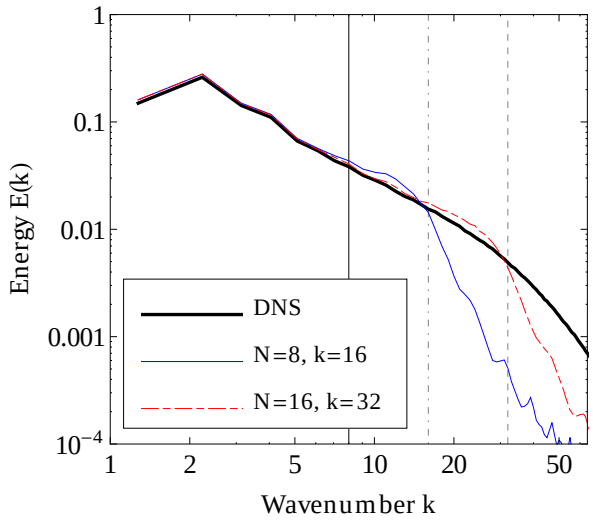
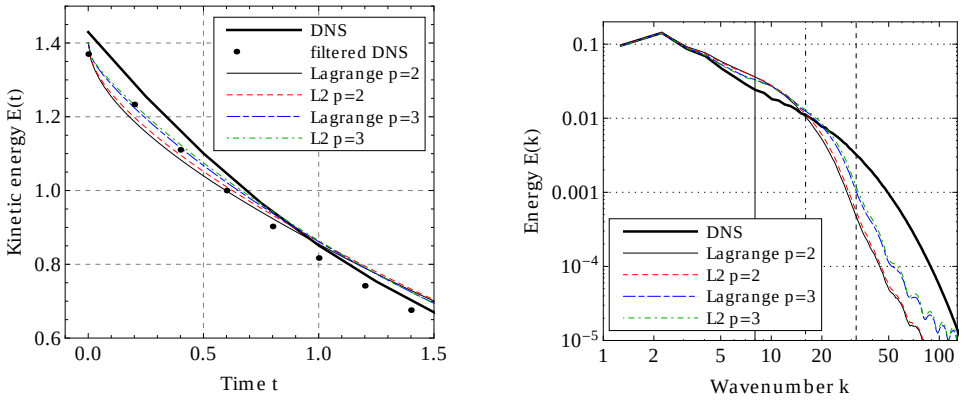
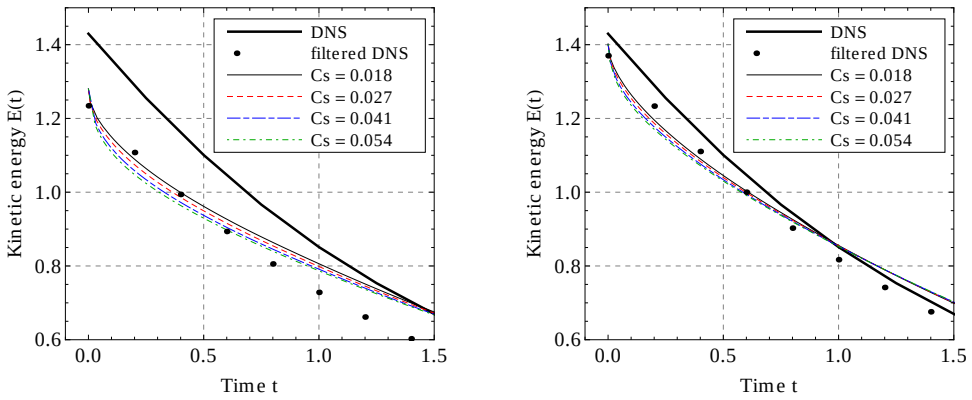


Figure 7.14: Kinetic energy spectrum for decaying homogeneous isotropic turbulence at $t = 0$. Before projection onto the computational grid, the initial DNS field is truncated to $p \times N/2$ Fourier modes ($p = 4$). The vertical lines represent $k = 8$ (solid), $k = 16$ (dot-dashed) and $k = 32$ (dashed), respectively.



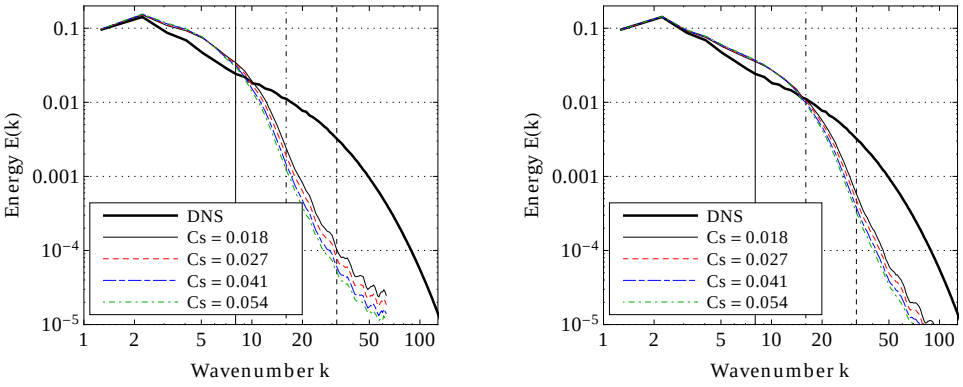
(a) $N = 16$ (b) $N = 16$

Figure 7.15: Influence of the SGS filtering on the decay of kinetic energy and the corresponding spectra at $t = 1$ for homogeneous isotropic turbulence using the "small-small" Smagorinsky model, $p = 4$.



(a) $N = 8$ (b) $N = 16$

Figure 7.16: Influence of the Smagorinsky constant C_s on the decay of kinetic energy for homogeneous isotropic turbulence using the "small-small" SGS model. Filtering is achieved by a Lagrange projection of the resolved scales ($p = 4$) onto polynomials of lower degree $\tilde{p} = 2$.



(a) $N = 8$ (b) $N = 16$

Figure 7.17: Influence of the Smagorinsky constant C_s on the kinetic energy spectra at $t = 1$ for homogeneous isotropic turbulence using the "small-small" SGS model. Filtering is achieved by a Lagrange projection of the resolved scales ($p = 4$) onto polynomials of lower degree $\tilde{p} = 2$.

Chapter 8

Conclusions and perspectives

The presented research constitutes a further step towards the use of high-order discontinuous Galerkin methods for the simulation of compressible turbulent aerodynamic flows around complex geometries. In the past the team of Jean-François Remacle (UCL) and the CFD Multi-Physics Group of CENAERO have developed a DG-based solver for laminar flows, called *Argo*. Within this thesis different RANS and LES turbulence models have been implemented into Argo with as long-term objective the capability to perform e.g. LES or DES computations of a complete fan. To this end all turbulence models have been implemented following a hybrid MPI/OpenMP parallelization strategy.

The main contributions of this work are improvements in the robustness of the coupled RANS solver necessitating several adaptations of the interior penalty discretization, the establishment of clear guidelines for resolved high-order boundary computations as well as different proposals to increase the accuracy of the estimated wall friction without affecting the problem size. Furthermore, the efficiency of the iterative Newton-Krylov solver has been significantly improved by replacing the former approximate Jacobian matrix by an exact linearization of the coupled RANS system.

8.1 Summary and conclusions

The following sections summarize the main findings of this work. More detailed concluding remarks can be found at the end of each chapter.

Interior penalty discretization. In contrast to other discontinuous Galerkin high-order RANS solvers using either LDG [107], BR2 [8, 107, 129] or CDG [125] schemes (see chapter 3), we have opted for an interior penalty approach. Since the IP formulation does not use lifting operators, the computational effort to evaluate the non-linear residual is reduced by up to 30 % compared to the popular BR2 scheme. In addition the IP formulation implies only direct neighbours of a computing cell and is thus well adapted for

massively parallel computations. Nonetheless, some stability issues had to be solved.

A first was caused by the *transpose term*, if the Navier-Stokes equations and the turbulence model are treated as a single block. We therefore propose to neglect some entries of this term in order to partially decouple the continuity equation and the turbulence model. In our experience this modification significantly improves the stability of the RANS solver.

A second difficulty is related to the definition of the *penalty parameter* δ which strongly affects the performance of the IP method. Following the same approach as Shahbazi [158] we have generalized the definition of δ in order to take into account mesh anisotropy on the one hand, and a fast changing viscosity on the other hand. Both issues are currently encountered in turbulent (boundary layer) computations. A coercivity study for the Poisson equation led to six expressions of δ resulting from two choices of the length scale and three choices of the viscosity scale. Whereas the length scale seems to have a minor influence, the choice of the viscosity scale was found to be of utmost importance. Hence, in case of merely resolved meshes, our new "elementwise" definition of the penalty parameter, which takes into account the polynomial character of the S-A variable, outperforms by far the standard "facewise" and the "maxMin" definition (cf. chapter 3).

Iterative solver. In this work we use a matrix-free Newton-GMRES algorithm to solve the non-linear system of RANS/LES equations. Although the Jacobian matrix is only needed by the BILU preconditioner, a non-negligible reduction of the total computing time was observed in the case of an exact linearization of the diffusive terms (see chapter 4). Thereby, precomputed parametric subblocks help to minimize the additional cost caused by the exact linearization (as opposed to the approximate linearization which was implemented for laminar flows). Finally, the computational efficiency has further been improved by freezing the preconditioner over some Newton steps.

Spalart-Allmaras turbulence model. High-order polynomial shape functions are prone to Gibbs oscillations in regions of sharp gradients and low mesh resolution. This situation is currently encountered at the outer boundary layer where it can cause unphysical negative values of the turbulent viscosity. To overcome this problem we have compared two adaptations of the S-A turbulence model (chapter 2 and chapter 3). The "modified" model is inspired by the work of Oliver [129]. It aims to improve the computational robustness through changes in the diffusive and source terms whilst conserving the C^1-continuity. This modification has been compared to a "clipped" model. Less intrusive than the one used by Landmann et al. [108] (since it does not affect the solution vector) the clipped model was found to be faster and even more stable, and this despite the loss of C^1-continuity. Furthermore, numerical re-

sults obtained by combining a high-order discretization of the Navier-Stokes equations with a low-order turbulent viscosity (cf. chapter 5) suggest that the clipping should not deteriorate the overall accuracy.

Turbulent boundary layer resolution. In chapter 5 and 6 several RANS applications of increasing complexity have been considered. Whereas the theoretical order of convergence of DG methods has already been investigated in the past, none of these studies provides clear guidelines for the choice of grid resolutions for practical applications. Given its simplicity we have chosen the turbulent flow along a flat plate in order to investigate the influence of the polynomial order on the required grid resolution on the one hand, and the behaviour of different types of discretization elements on the other (chapter 5). Computations of the turbulent flow around a NACA 0012 aerofoil have been performed to verify these conclusions in case of curved geometries.

All these studies clearly show that a much better estimation of the wall shear friction is obtained if it includes the penalty contribution. In addition, the accuracy can further be improved by clustering the (internal) high-order interpolation nodes in the vicinity of the wall. Thereby, only the boundary elements (in contact with the solid wall) have to be adapted.

In comparison with classical low-order methods, an important reduction of the number of degrees of freedom was found, both in terms of stretching and spacing (cf. also chapter 6). Especially the high stretching and the insensitivity of DG methods to structured-unstructured connections can greatly simplify grid generation. Nonetheless, the representation of curved boundaries has an important influence on the accuracy of the method and causes unphysical oscillations if the continuity of the normal vectors is not ensured. For that reason a geometrical representation which is at least third order ($q = 3$) is advisable.

Using regular quadrilateral regions in the boundary layer is beneficial. Not only quadrilaterals result in a shear friction which is one order more accurate, but also the jump penalization between neighbouring cells along the boundary significantly smooths the solution.

Finally, our computations suggest that the need for using a high-order interpolation for both the Navier-Stokes equations and the S-A turbulence model is more related to the robustness of the RANS solver than to the accuracy of the results.

Large-eddy simulations. To validate the implementation of the Smagorinsky and the WALE subgrid-scale model, the results for the Taylor-Green vortex and decaying homogeneous isotropic turbulence have been benchmarked against DNS data computed by a pseudo-spectral code (chapter 7).

First, we should mention the rapid convergence of the LES results to the DNS reference as the grid is refined. A comparison between standard LES and

VMS confirmed previous studies and showed that unfiltered LES ("all-all") is most often too dissipative, whereas implicit LES does not replace dedicated turbulence modelling. In contrast, the results of the filtered VMS methods were generally close to the DNS data. Despite the overall good agreement, an excess in the energy spectrum of the small resolved scales has been observed. Its origin remains unclear and should be further investigated.

According to our numerical results, no difference has been observed between the Lagrange and the L2 filter that have been implemented into Argo. The effectiveness of subgrid filtering has been tested by increasing the turbulent dissipation through the Smagorinsky constant.

Finally, as for the RANS computations, the method is insensitive to the choice of a specific value of the penalty parameter. As a consequence optimizing this coefficient does not replace turbulence modelling.

8.2 Future work

The numerical applications presented in chapter 5 and 6 clearly demonstrate the sensitivity of DG methods to a smooth representation of curved boundaries. This is a challenging task and remains subject of active research. However, in order to avoid excessive grid refinements related to geometrical representation purposes, future work should concentrate on meshing algorithms.

In addition, another way of reducing CPU and memory requirements would be the use of h/p-adaptation techniques. Especially for time-dependent large eddy or detached eddy simulations, a significant gain of degrees of freedom is to be expected, and hence adaptation strategies should be integrated into the flow solver. Automatic adaptation algorithms are for example described in [81, 129].

Within this work all RANS computations have been performed using the one-equation Spalart-Allmaras turbulence model. Although the S-A model has been shown to give acceptable results for a wide range of applications, more sophisticated models (e.g. $k-\omega$, $k-\epsilon$, Reynolds stress) should also be considered. Furthermore, it would be interesting to analyse whether a low-order interpolation of these models has a similar effect to the case of the S-A model.

Concerning the Smagorinsky and WALE models, only flows within a three-dimensional periodic box have been considered so far. The next validation should be their application to wall-bounded flows, e.g. a turbulent channel flow. Furthermore, a drawback of the presented LES method is the a priori definition of the model constant in the subgrid terms. To overcome this limitation, a dynamic procedure to evaluate the model constant should be implemented. Finally, more sophisticated subgrid filters providing a sharper cut-off in the spectral space should be investigated. Within this context it would be interesting to program the exact analytical Fourier transform. By eliminating

the numerical noise in the high wavenumber range, the exact Fourier transform allows a more accurate comparison of subgrid models and filtering techniques.

Whilst work has begun to implement detached eddy models based on the software developed in this thesis, these models have not yet been tested in practice.

Finally, since many applications of engineering interest are supersonic, shock limitation should be included into Argo.

Bibliography

[1] R. Abgrall. Residual distribution schemes: current status and future trends. *Journal of Computational Fluids*, 35:641–669, 2006.

[2] AGARD. Experimental Data Base for Computer Program Assessment. Technical Report AGARD-AR-138, 1979.

[3] Heng-Bin An, Ze-Yao Mo, and Xing-Ping Liu. A choice of forcing terms in inexact Newton method. *Journal of Computational and Applied Mathematics*, 200:47–60, 2007.

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

[5] W. E. Arnoldi. The principal of minimized iteration in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.

[6] G. A. Ashford. *An Unstructured Grid Generation and Adaptive Solution Technique for High Reynolds Number Compressible Flows.* PhD thesis, University of Michigan, USA, 1996.

[7] I. Babuška and M. Zlámal. Nonconforming elements in the finite element method with penalty. *SIAM Journal of Numerical Analysis*, 10:863–875, 1973.

[8] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-omega turblence model equations. *Computers & Fluids*, 34:507–540, 2005.

[9] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

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

[11] F. Bassi and S. Rebay. *Discontinuous Galerkin methods, theory, computation and applications*, chapter A high order discontinuous Galerkin method for compressible turbulent flows, pages 77–88. Springer, 2000.

[12] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, and M. Savini. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In R. Decuypere and G. Dibelius, editors, *2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*, pages 99–108, Antwerpen, Belgium, 1997.

[13] C. E. Baumann and J. T. Oden. A discontinuous hp finite element method for the convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175:311–341, 1999.

[14] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[15] H. M. Blackburn and S. Schmidt. Spectral element filtering techniques for large eddy simulation with dynamic estimation. *Journal of Computational Physics*, 186:610–629, 2003.

[16] J. Boussinesq. Thérie de l' écoulement tourbillonnant. *Comptes Rendus de l'Académie des Sciences*, pages 46–50, 1877.

[17] M. E. Brachet, D. Meiron, S. Orszag, B. Nickel, R. Morf, and U. Frisch. The Taylor-Green vortex and fully developed turbulence. *Journal of Statistical Physics*, 34:1049–1063, 1984.

[18] F. D. Bramkamp. *Unstructured h-adaptive finite-volume schemes for compressible viscous fluid flows*. PhD thesis, RWTH Aachen, 2003.

[19] F. Brezzi, L. P. Franca, and A. Russo. Further considerations on residual-free bubbles for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 166:25–33, 1998.

[20] F. Brezzi and A. Russo. Choosing bubbles for advection-diffusion problems. *Mathematical Models and Methods in Applied Sciences*, 4:571–587, 1994.

[21] L. Bricteux, M. Duponcheel, and G. Winckelmans. A multiscale subgrid model for both free vortex flows and wall-bounded flows. *Physics of Fluids*, 21, 2009.

[22] P. N. Brown. A local convergence theory for combined inexact-Newton/finite-difference projection methods. *SIAM Journal on Numerical Analysis*, 24:407–434, 1987.

[23] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonliner systems of equations. *SIAM Journal on Scientific and Statistical Computing*, 11:450–481, 1990.

[24] P. N. Brown and Y. Saad. Convergence theory for nonlinear Newton-Krylov algorithms. *SIAM Journal on Optimization*, 4:297–330, 1994.

[25] N. K. Burgess and D. J. Mavriplis. Robust computation of turbulent flows using a discontinuous Galerkin method. In *50th AIAA Aerospace Science Meeting*, AIAA Paper 2012-0457, January 2012.

[26] L. Cambier and M. Gazaix. elsa: an efficient object-oriented solution to cfd complexity. In *40th AIAA Aerospace Science Meeting and Exhibit*, 2002.

[27] L. Cambier and J. P. Veuillot. Status of the elsa cfd software for flow simulation and multidisciplinary applications. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, January 2008.

[28] P. Castillo. Performance of discontinuous Galerkin methods for elliptic pdes. *SIAM Journal on Scientific Computing*, 24:524–547, 2002.

[29] S. Catris and B. Aupoix. Density corrections for turbulence models. *Aerospace Science and Technology*, 4:1–11, 2000.

[30] A. Celić and E. H. Hirschel. Comparison of eddy-viscosity turbulence models in flows with adverse pressure gradients. *AIAA Journal*, 44(10):2156–2169, 2006.

[31] G. Chavent and G. Salzano. A finite element method for the 1d water flooding problem with gravity. *Journal of Computational Physics*, 45:307–344, 1982.

[32] Z. Chen. *Finite element methods and their applications*, chapter Discontinuous finite elements, pages 173–213. Springer Berlin Heidelberg, 2005.

[33] N. Chevaugeon, K. Hillewaert, X. Gallez, P. Ploumhans, and J.-F. Remacle. Optimal numerical parametrization of the discontinuous Galerkin method applied to wave propagation problems. *Journal of Computational Physics*, 223:188–207, 2007.

[34] T. T. Chisholm. *A fully coupled Newton-Krylov solver with a one-equation turbulence model*. PhD thesis, University of Toronto Institute for Aerospace Studies, 2007.

[35] T. T. Chisholm and D. W. Zingg. A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows. *Journal of Computational Physics*, 228:3490–3507, 2009.

[36] B. Cockburn. *An introduction to the discontinuous Galerkin method for convection-dominated problems*. Lecture Note Series. CIME, 1997.

[37] B. Cockburn. Discontinuous Galerkin methods. *Z. Angew. Math. Mech.*, 83(11):731–754, 2003.

[38] B. Cockburn, G. Karniadakis, and C. Shu. The development of discontinuous Galerkin methods. In *Lecture Notes in Computational Science and Engineering*, volume 11. Springer, 2000.

[39] B. Cockburn and C. W. Shur. The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM Journal of Numerical Analysis*, 35:337–361, 1998.

[40] R. Courant and K.O. Friedrichs. *Supersonic flow and shock waves*, volume 21 of *Applied Mathematical Sciences*. Springer, Berlin, 1999. reprint from 1948.

[41] E. Cuthill. Several strategies for reducing the bandtwidth of matrices. In D. J. Rose and R. A. Willoughby, editors, *Sparse matrices and their applications*, pages 157–166. New York, USA, 1972.

[42] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *24th national conference*, pages 157–172, New York, 1969.

[43] J. Dacles-Mariani, G. G. Zilliac, J. S. Chow, and P. Bradshaw. Numerical/experimental study of a wingtip vortex in the near field. *AIAA Journal*, 33(9):1561–168, 1995.

[44] G. Dahlquist. A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, 3:27–43, 1963.

[45] G. Dahlquist, Å. Björck, and N. Anderson. *Numerical methods*, chapter Approximation of functions, pages 81–131. Prentice Hall, 1974.

[46] H. Deconinck, K. Sermeus, and R. Abgrall. Status of multidimensional upwind residual distribution schemes and application in aeronautics. *AIAA paper*, 2000-2328, 2000.

[47] M. Delanaye. *Polynomial reconstruction finite volume schemes for the compressible Euler and Navier-Stokes equations on unstructured adaptive grids*. PhD thesis, University of Liège, 1998.

[48] R. S. Dembo, S. C. Eisenstat, and T. Steighaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[49] V. Dolejší. On the discontinuous Galerkin method for the numerical solution of the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 45:1083–1106, 2004.

[50] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic Galerkin methods. In *Lecture Notes in Physics*. Springer, 176.

[51] 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*, November 2012. In Press, Corrected Proof, http://dx.doi.org/10.1016/j.cam.2012.09.019.

[52] 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. submitted for publication in SIAM Journal on Scientific Computing, 2011.

[53] M. Duponcheel. *Direct and large-eddy simulation of turbulent wall-bounded flows: further development of a parallel solver, improvement of multiscale subgrid models and investigation of vortex pairs in ground effect*. PhD thesis, Université catholique de Louvain, 2009.

[54] P. A. Durbin and B. A. Pettersson Reif. *Statistical theory and modeling for turbulent flows*. Wiley & Sons, 2000.

[55] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in a inexact Newton method. *SIAM Journal on Scientific Computing*, 17:16–32, 1996.

[56] J. A. Ekaterinaris. High-order accurate. low diffusion methods for aerodynamics. *Progress in Aerospace Sciences*, 41:192–300, 2005.

[57] Y. Epshteyn and B. Rivière. Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *Journal of Computational and Applied Mathematics SIAM*, 206:843–872, 2007.

[58] J.A Essers, E. Renard, E. Chapelle, and B. Custinne. Construction d'un code de calcul pour la simulation de phénomènes complexes de convection-diffusion sur super-ordinateurs. Technical Report IT/SC/32, Université de Liège, 1992.

[59] A. Favre. Equations des gaz turbulents compressibles. *Journal de Mécanique*, 4:361–390, 1965.

[60] I. Fejtek. Summary of code validation results for a multiple element airfoil test case. In *28th AIAA Fluid Dynamics Conference*, AIAA Paper 97-1932, 1997.

[61] Fluid Dynamics TC, AFOSR, and DLR, editors. *International Workshop on High-Order CFD Methods*1st, Nashville, Tennsessee, January 7-8 2012. 50th AIAA Aerospace Sciences Meeting.

[62] L. P. Franca and C. Farhat. Bubble functions prompt unusual stabilized finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 123:299–308, 1995.

[63] E. Garnier, N. Adams, and P. Sagaut. *Large Eddy Simulation for compressible flows*, chapter LES governing equations, pages 5–39. Springer, 2009.

[64] G. Gassner, F. Lörcher, and C.-D. Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 224(2):1049–1063, 2007.

[65] A. George. Computer solution of large sparse positive definite systems. Technical Report STAN-CS-208, Stanford University, Department of Computer Science, 1971.

[66] L. Georges. *Development and validation of a LES methodology for complex wall-bounded flows: Application to high-order structured and industrial unstructured solvers.* PhD thesis, Université catholique de Louvain, June 2007.

[67] L. Georges, K. Hillewaert, R. Capart, J.-F. Thomas, T. Louagie, and P. Geuzaine. RANS-LES simulations around complex geometries using unstructured compressible flow solver. In *7th International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements*, 2008.

[68] E. H. Georgoulis. hp-version interior penalty discontinuous Galerkin finite element methods on anisotropic meshes. *International Journal of Numerical Analysis and Modeling*, 3(1):52–79, 2006.

[69] M. Germano, U. Piomelli, P. Moin, and W. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids*, 3(7):1760–1765, 1991.

[70] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and pos- processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[71] P. Geuzaine. *An implicit finite volume method for compressible turbulent flows on unstructured meshes.* PhD thesis, University of Liège, 1999.

[72] P. Godin, D. W. Zingg, and T. E. Nelson. High-lift aerodynamic computations with one- and two-equation turbulence models. *AIAA Journal*, 35(2):237–243, 1997.

[73] V. Gravemeier. The Variational Multiscale method for laminar and turbulent flows. *Archives of Computational Methods in Engineering*, 13(2):249–324, 2006.

[74] V. Gravemeier and W. A. Wall. Residual-based variational multiscale methods for laminar, transitional and turbulent variable-density flow at low mach number. *International Journal for Numerical Methods in Fluids*, 2009.

[75] F. F. Grinstein, M. G. Margolin, and W. J. Rider. *Implicit large eddy simulation*, volume I. Cambridge University Press, 2007.

[76] K. Harriman, D. Gavaghan, and E. Süli. The importance of adjoint consistency in the approximation of linear functionals using the discontinuous Galerkin finite element method. Technical Report 04/18, Oxford University Computing Laboratory, Numerical Analysis group, July 2004.

[77] K. Harriman, P. Houston, B. Senior, and E. Süli. hp-version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form. Technical Report NA 02/21, Oxford University Computing Laboratory, Numerical Analysis Group, England, 2002.

[78] R. Hartmann. Derivation of an adjoint consistent discontinuous Galerkin discretization of the compressible Euler equations. In G. Lube and G. Rapin, editors, *Conference on Boundary and Interior Layers BAIL 2006*. University of Göttingen, 2006.

[79] R. Hartmann. Adjoint consistency analysis of discontinuous Galerkin discretizations. *SIAM Journal of Numerical Analysis*, 45(6):2671–2696, 2007.

[80] R. Hartmann. Error estimation and adjoint based refinement for an adjoint consistent DG discretisation of the compressible Euler equations. *International Journal of Computing Science and Mathematics*, 1(2-4):207–220, January 2007.

[81] R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and $k-\omega$ turbulence model equations. *Journal of Computational Physics*, 230:4268–4284, 2011.

[82] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations I: method formulation. *International Journal of Numerical Analysis and Modeling*, 3(1):1–20, 2006.

[83] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier-Stokes II: goal-oriented a posteriori error estimation. *International Journal of Analysis and Modeling*, 3(2):141–162, 2006.

[84] R. Hartmann and P. Houston. An optimal interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 227(22):9670–9685, 2008.

[85] A. Hellsten. New two-equation turbulence model for aerodynamic applications. Technical Report A-21, Helsinki University of Technology, Laboratory for Aerodynamics, 2004.

[86] K. Hillewaert, M. Drosson, and J.-F. Remacle. Sharp constants in the hp-finite element trace inverse inequality for standard functional spaces on all element types in hybrid meshes. submitted for publication in SIAM Journal on Numerical Analysis.

[87] K. Hillewaert, J.-F. Remacle, N. Chevaugeon, and P. Geuzaine. Discontinuous Galerkin methods: implementation issues. In *Lecture series on CFD/ADIGMA course on very high order discretization methods*. von Karman Institute for Fluid Dynamics, 2008.

[88] K. Hillewaert, J.-F. Remacle, B. Helenbrook, and P. Geuzaine. Exploiting single precision BLAS/LAPACK for the efficient implementation of the implicit discontinuous Galerkin method. in preparation.

[89] C. Hirsch. *Numerical computation of internal and external flows, computational methods for inviscid and viscous flows*, volume 2. Wiley, 1990.

[90] T. J. R. Hughes, L. Mazzei, and K. Jansen. Large-eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3:47–59, 2000.

[91] T. J. R. Hughes, L. Mazzei, A. A. Oberai, and A. A. Wray. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of Fluids*, 13(2):505–512, 2001.

[92] T. J. R. Hughes, A. A. Oberai, and L. Mazzei. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Physics of Fluids*, 13(6):1784–1799, 2001.

[93] A. Jamseon, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA paper*, 81-1259, 1981.

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

[95] G. Kalitzin, G. Medic, G. Iaccarino, and P. Durbin. Near-wall behavior of RANS turbulence models and implications for wall functions. *Journal of Computational Physics*, 204:265–291, 2005.

[96] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 170:133–166, 1987.

[97] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and apllications. *Journal of Computational Physics*, 193:357–397, 2004.

[98] D. A. Knoll and P. R. McHugh. Newton-Krylov methods applied to a system of convection-diffusion-reaction equations. *Communications in Computational Physics*, 88:141–160, 1995.

[99] D. A. Knoll and P. R. McHugh. Enhanced nonlinear iterative techniques applied to a nonequilibrium plasma flow. *SIAM Journal on Scientific Computing*, 19:291–301, 1998.

[100] T. Knopp, T. Alrutz, and D. Schwamborn. A grid adaptive wall-function method for RANS turbulence modelling. *Journal of Computational Physics*, 220:19–40, 2006.

[101] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Dokl. Akad. Nauk SSSR*, 30:299–303, 1941.

[102] B. Koobus and C. Farhat. A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes - application to vortex shedding. *Computer Methods in Applied Mechanics and Engineering*, 193:1367–1383, 2004.

[103] T. H. Koornwinder. *Hypergeometric functions on domains of positivity, Jack polynomials, and applications*, chapter Askey-Wilson polynomials for root systems of type BC, pages 189–204. American Math. Society, 1992.

[104] N. Kroll. ADIGMA - a European project on the development of adaptive high-order variational methods for aerospace applications. In *47th AIAA Aerospace Sciences Meeting*, volume 176. AIAA, 2009.

[105] N. Kroll, H. Bieler, H. Deconinck, and V. Couaillier, editors. *ADIGMA - A European initiative on the development of adaptive high-order variational methods for aerospace applications - Results of a collaborative research project funded by the European Union*, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer, 2006-2009.

[106] K. R. Laflin, J. C. Vassberg, R. A. Wahls, J. H. Morrison, O. Brodersen, M. Rakowitz, E. N. Tinoco, and J.-L. Godard. Summary of data from

the second AIAA CFD drag prediction workshop. In *AIAA Paper 2004-0555*, 2004.

[107] B. Landmann. *A parallel discontinuous Galerkin code for the Navier-Stokes and Reynolds-averaged Navier-Stokes equations*. PhD thesis, University of Stuttgart, 2008.

[108] 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.

[109] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the development of a Reynolds stress turbulence closure. *Journal of Fluid Mechanics*, 68(3):537–566, 1975.

[110] I. Lepot. *A parallel high-order implicit finite volume method for three-dimensional inviscid compressible flow on deforming unstructured meshes*. PhD thesis, University of Liège, 2004.

[111] P. LeSaint and P. A. Raviart. On a finite element method for solving the neutron transport equation. In *Mathematical aspects of finite elements in partial differential equations*, pages 89–123. Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York, 1974.

[112] M. Lesieur, O. Métais, and P. Comte. *Large-Eddy simulations of turbulence*. Cambridge University Press, 2005.

[113] D. W. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R. A. Wahls, S. Pirzadeh, and M. J. Hemsch. Data summary from the first AIAA computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, 40(5):875–882, 2003.

[114] M. J. Lighthill. *Computational Aeroacoustics*. Springer Verlag, New York, 1993.

[115] D. K. Lilly. The representation of small-scale turbulence in numerical simulation experiments. In *IBM Scientific Symposium on Envirenmental Sciences*, pages 195–210. IBM DP Devision, 1967.

[116] J.-L. Lions. Problèmes aux limites non homogènes à données irrégulières : une méthode d'approximation. In *Numerical analysis of partial differential equations*, pages 283–292. Springer, 1968.

[117] J. Lu. *An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method*. PhD thesis, Massachusetts Institue of Technology, Cambridge, 2005.

[118] X.-J. Luo, M. Shephard, and J.-F. Remacle. p-version mesh generation issues. In *11th International Meshing Roundtable*, 2002.

[119] N. Mansour, J. Kim, and P. Moin. Reynolds-stress and dissipation-rate budgets in a turbulent channel flow. *Journal of Fluid Mechanics*, 194:15–44, 1988.

[120] M. P. Martín, U. Piomelli, and G. V. Candler. Subgrid-scale models for compressible large-eddy simulations. *Theoretical and Computational Fluid Dynamics*, 13:361–376, 2000.

[121] I. R. M. Moir. *Measurements on a two-dimensional aerofoil with high-lift devices*, volume II of *AGARD-Report AGARD-AR-303*, chapter A2. 1994.

[122] R. Moser, J. Kim, and N. Mansour. Direct numerical simulation of turbulent channel flow up to $re_\tau = 590$. *Physics of Fluids*, 11(4):943–945, 1999.

[123] D. Mount and S. Arya. http://www.cs.umd.edu/∼mount/ANN/.

[124] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen. How fast are non-symmetric matrix iterations. *SIAM Journal on Matrix Analysis and Applications*, 13:778–795, 1992.

[125] 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*, January 2007.

[126] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62:183–200, 1999.

[127] J. A. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 35:9–15, 1971.

[128] T. Oliver. Multigrid solution for high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. Master's thesis, Massachusetts Institute of Technology, 2002.

[129] 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.

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

[131] M. Pernice and H. F. Walker. NITSOL: a Newton iterative solver for nonlinear systems. *SIAM Journal on Scientific Computing*, 19:302–318, 1998.

[132] P.-O. Persson. Scalable parallel Newton-Krylov solvers for discontinuous Galerkin discretizations. In *47th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2009-606, 2009.

[133] P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, January 2006.

[134] P.-O. Persson and J. Peraire. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2733, 2008.

[135] T. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal of Numerical Analysis*, 28:133–140, 1991.

[136] B. Pollul. *Iterative solvers in implicit time integration for compressible flows*. PhD thesis, RWTH Aachen, 2008.

[137] S. B. Pope. A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*, 72:331–340, 1975.

[138] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2001.

[139] S. De Rango and D. W. Zingg. Aerodynamic computations using higher-order algorithms. *AIAA paper*, 99-0167, 1999.

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

[141] H. Reichardt. Vollst andige Darstellung der turbulent Geschwindigkeitsverteilung in glatten Leitungen. *Z. angew. Math Mech*, 31(7):208–219, 1951.

[142] 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.

[143] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations - Theory and implementation*. Society for Industrial and Applied Mathematics SIAM, 2008.

[144] B. Rivière, M. F. Wheeler, and V. Girault. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. Technical Report 99-09, TICAM, 1999.

[145] W. Rodi. A new algebraic relation for calculating the Reynolds stresses. *Z. Angew. Math. Mech.*, 56:219–221, 1976.

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

[147] R. Rudnik. *Untersuchung der Leistungsfähigkeit von Zweigleichungs-Turbulenzmodellen bei Profilströmungen.* PhD thesis, Technische Universität Berlin, 1997.

[148] C. L. Rumsey. Apparent transition behavior of widely-used turbulence models. *International Journal of Heat and Fluid Flow*, 28:1460–1471, 2007.

[149] C. L. Rumsey, T. B. Gatski, S. X. Ying, and A. Bertelrud. Prediction of high-lift flows using turbulence closure models. In *15th AIAA Applied Aerodynamics Conference*, AIAA Paper 1997-2260, 1997.

[150] C. L. Rumsey and V. N. Vatsa. A comparison of the predictive capabilities of several turbulence models using upwind and central-difference computer codes. *AIAA paper*, 93-0192, 1993.

[151] C. Runge. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik*, 46:224–243, 1901.

[152] Y. Saad. *Iterative methods for sparse linear systems.* International Thomson Publishing Inc., 1996.

[153] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[154] A. Schubert. *Grundlagen für die numerische Simulation von laminaren und turbulent Hyperschallströmungen.* PhD thesis, RWTH Aachen, 1998.

[155] F. Schultz-Grunow. New frictional resistance law for smooth plates. *NACA TM*, 17:1–24, 1941.

[156] K. Sengupta, G. B. Jacobs, and F. Mashayek. Large-eddy simulation of compressible flows using a spectral multidomain method. *International Journal for Numerical Methods in Fluids*, 61(3):311–340, September 2009.

[157] J. N. Shadid, R. S. Tuminaro, and H. F. Walker. An inexact Newton method for fully coupled solution of the Navier-Stokes equations with heat and mass trasnport. *Journal of Computational Physics*, 137:155–185, 1997.

[158] K. Shahbazi. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics*, 205:401–407, 2005.

[159] C.-W. Shu, W.-S. Don, D. Gottlieb, O. Schilling, and L. Jameson. Numerical convergence study of nearly incompressible, inviscid Taylor-Green vortex flow. *Journal of Scientific Computing*, 24(1), 2005.

[160] M. Shur, M. Strelets, A. Travin, and P. R. Spalart. Turbulence modeling in rotating and curved channels: Assessment of the spalart-shur correction term. *AIAA Journal*, 38(5):784–792, 2000.

[161] J. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–164, 1693.

[162] P. Solin, K. Segeth, and I. Dolezel. *Higher-order finite element methods*. Studies in Advanced Mathematics. Chapman and Hall, 2004.

[163] P. R. Spalart. Trends in turbulence treatments. *AIAA 2000-2306*, June 2000.

[164] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA paper*, pages 92–0439, January 1992.

[165] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aérospatiale*, 1:5–21, 1994.

[166] K. D. Squires. Dynamic subgrid scale modeling of compressible turbulence. Technical report, Annual Research Brief, Stanford University, 1991.

[167] K. D. Squires. Detached-eddy simulation of turbulent flows. In *Lecture series on Large Eddy Simulation and related techniques: theory and applications*. von Karman Institute for Fluid Dynamics, March 2008.

[168] R. C. Swanson and E. Turkel. Multistage schemes with multigrid for Euler and Navier-Stokes equations. Technical report, NASA TP-3631, 1997.

[169] C. H. Tai, J. H. Sheu, and B. van Leer. Optimal multistage schemes for Euler equations with residual smoothing. *AIAA Journal*, 33:1008–1016, 1995.

[170] G. I. Taylor and A. E. Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society A*, 151:499–521, 1935.

[171] M. D. Tidriri. Domain decomposition for compressible Navier-Stokes equations with different discretizations and formulations. *Comput. Phys.*, 119:271–282, 1995.

[172] R. S. Tuminaro, H. F. Walker, and J. N. Shadid. On backtracking failure in Newton-GMRES methods with a demonstration for the Navier-Stokes equations. *Journal of Compuational Physics*, 180:549–558, 2002.

[173] F. van der Bos and B. Geurts. Computational error-analysis of the discontinuous Galerkin discretization applied to large-eddy simulation of homogeneous turbulence. *Computer Methods in Applied Mechanics and Engineering*, 199:903–915, 2010.

[174] H. A. van der Vorst. Bi-CGSTAB: a fast and smooth converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[175] H. Versteeg and W. Malalasekra. *An Introduction to Computational Fluid Dynamics - The finite volume method*. Pearson, 2007.

[176] A. W. Vreman. The filtering analog of the variational multiscale method in large-eddy simulation. *Physics of Fluids*, 15(8):L61–L64, August 2003.

[177] A. W. Vreman, B. J. Guerts, and J. Kuerten. *Direct and large-eddy simulation*, volume I. Kluwer Academic Publishers, 1994.

[178] B. Vreman, B. Geurts, and H. Kuerten. Subgrid-modeling in LES of compressible flow. *Applied Scientific Research*, 54:191–203, 1995.

[179] Z. J. Wang. High-order methods for the Euler and Navier-Storkes equations on unstructured grids. *Progress in Aerospace Sciences*, 43:1–41, 2007.

[180] T. Warburton and J. S. Hesthaven. On the constants in hp-finite element trace inverse inequality. *Computer Methods in Applied Mechanics and Engineering*, 192:2765–2773, 2003.

[181] K. Wieghardt and W. Tillman. On the turbulent friction layer for rising pressure. *NACA TM*, 1314, 1951.

[182] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, 1993.