

A Chimera Simulation Method and Detached Eddy Simulation for Vortex-Airfoil Interactions

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades

”Doctor rerum naturalium”

der Georg-August-Universität Göttingen

vorgelegt von
Christoph Wolf
aus Karlsruhe

Göttingen 2010

Referent: Prof. Dr. Gert Lube

Koreferent: Prof. Dr. Robert Schaback

Tag der mündlichen Prüfung: 20. 12. 2010

Contents

1	Introduction	4
1.1	Motivation and objective of the thesis	4
1.2	Overview of the literature	6
1.3	Structure and main results of the thesis	15
1.4	New aspects of the thesis	20
I	Mathematical methods	22
2	Numerical methods in the DLR TAU-code	23
2.1	The mathematical model for compressible flow	23
2.2	Spatial discretisation	27
2.3	Time discretisation	35
2.4	Preconditioning methods	38
2.5	Multigrid methods	42
3	Turbulence modeling	47
3.1	Reynolds averaging and Favre averaging	47
3.2	The Favre averaged Navier-Stokes equations	49
3.3	URANS models	51
3.4	Hybrid RANS/LES models	55
4	Chimera: a domain decomposition method	58
4.1	An introduction into domain decomposition methods	59
4.2	Moving subdomains	64
4.3	The Chimera technique	67
II	Numerical applications	71
5	Flow over a backward facing step	72
5.1	Description of the testcase	73
5.2	Numerical results	74
5.3	Examination of two sensors	80

6	An HGR01 airfoil at stall	86
6.1	Description of the testcase	87
6.2	An examination of the reliability of SA-DDES at $\alpha = 12^\circ$	88
6.3	Results with SA-DDES ₁₆ of the HGR01 airfoil at stall	98
7	Transport and collision of vortices	106
7.1	Description of the testcases	107
7.2	Examinations of vortex transport	107
7.3	Vortex interaction with a NACA 0012 airfoil at $\alpha = 0^\circ$	110
7.4	Vortex interaction with an ONERA-A airfoil at $\alpha = 13.3^\circ$	115
8	Applications of the simulation method	118
8.1	Two-dimensional (U)RANS simulations of an FNG airfoil	120
8.2	Two-dimensional URANS simulations	127
8.3	A three-dimensional SA-URANS simulation	133
8.4	A three-dimensional SA-DDES simulation	136
9	Conclusion	144
9.1	Summary of the results	144
9.2	Future work	148
A	A global existence result	150
B	Nomenclature	155

Chapter 1

Introduction

1.1 Motivation and objective of the thesis

The modeling and simulation of large-scale vortices in the onflow conditions and their interaction with airplanes pose an immense challenge in today's air traffic. Such vortices – whose length scales range from microscopically small up to several magnitudes of an airplane – can evolve naturally in the atmosphere due to certain weather conditions. Another source for the creation of dangerous disturbances lies in the airplanes themselves: During take-off and landing huge counter-rotating wake vortices evolve. Both types of disturbances pose a great danger for oncoming airplanes and can cause fatal accidents. The status quo of avoiding the risks of wake vortices lies in waiting several minutes (depending on the size of the planes and other factors) before allowing the next airplane to take-off or land on the same runway. In times of worldwide increasing air traffic, this limitation becomes more and more problematic.

The aim of this thesis lies on the development and application of a numerical simulation method that allows to generate a realistic vortex, to transport it towards an airfoil and to simulate the vortex-airfoil interaction to predict the forces and moments acting on the wing. Especially the question whether airfoil stall occurs due to the vortex-airfoil interaction is of major interest. Airfoil stall is hereby defined as massive flow separation at the wing resulting in a loss of lift, which forces the aircraft to drop and possibly makes control of the airplane impossible.

In order to develop and evaluate the simulation method, several preliminary investigations are performed. One examination deals with the turbulence model used: As it is well-known that standard URANS-methods are not capable of resolving small-scale structures (which occur at the wing during a vortex-airfoil interaction), the use of a hybrid RANS/LES method is advisable. Therefore the performance of a hybrid RANS/LES model is examined in Chapters 5 and 6, where the model is applied to a testcase including massive flow separation and respectively used to simulate the flow around an airfoil at stall, where only a mild trailing edge separation occurs. Whether the approach is capable of transporting vortices over large distances without losing them due to numerical dissipation is investigated in Chapter 7. At the end of this chapter also two vortex-airfoil interactions are presented.

Having performed the preliminary investigations, the simulation method can be introduced and applied in Chapter 8. We propose the following approach, which makes use of the so-called Chimera technique and is illustrated in Fig. 1.1. The upper picture of Fig. 1.1 shows the initial setting at the beginning of the computation. Embedded within a background grid ("orange") lie a "vortex generation grid" ("red") containing an airfoil to generate the vortex, a "vortex transport grid" ("green") to preserve and transport the vortex and a "vortex interaction grid" ("blue") containing an airfoil that in the end interacts with the vortex. While the three local grids should be chosen sufficiently fine to preserve the vortex, the background grid covering most of the computational domain can be relatively coarse, thus minimising the numerical costs. First the vortex generation grid is flapped upwards relative to the background grid, which is indicated in Fig. 1.1 (middle). As a result, the realistic vortex evolves from the trailing edge of the airfoil and moves with the free stream velocity onto the vortex transport grid. Having been interpolated onto the latter, both the vortex and the vortex transport grid are moved simultaneously with the free stream velocity towards the vortex interaction grid. As the vortex transport grid and the airfoil of the vortex interaction grid must not overlap, the vortex transport grid is stopped sufficiently far away from the airfoil, which can be seen in Fig. 1.1 (lower). The vortex continues to move to the right and is finally interpolated onto the vortex interaction grid, where the vortex-airfoil interaction eventually takes place.

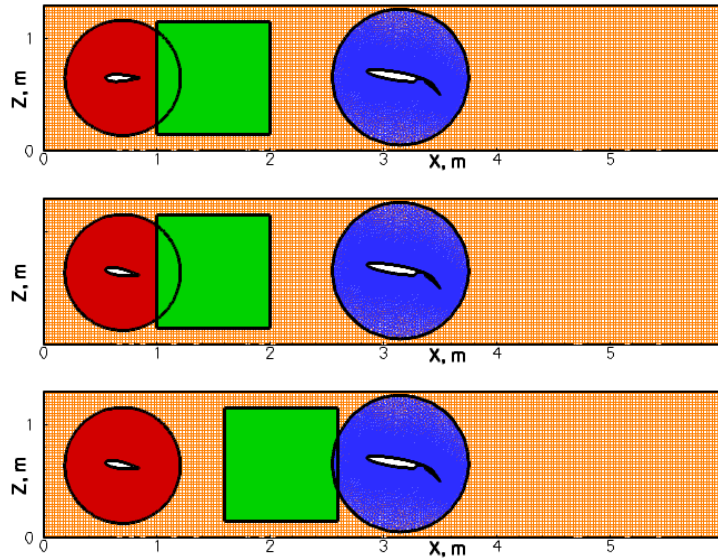


Figure 1.1: Relative position of the four Chimera grids at the beginning of the computation (upper picture), after the red "vortex generation grid" has been flapped upwards (middle) and after the green "vortex transport grid" has been stopped (lower).

The work presented in this thesis was performed in the framework of a subproject of the DFG PAK 136 project and of the DFG FOR 1066 Forschungsguppe. The DFG PAK 136 project, which consisted of five subprojects, was started in september 2006

and had a project duration of two years. The subsequent Forschergruppe, which was initiated in december 2008, consists of eight subprojects and is supposed to run for a total of six years. The major aim is the numerical investigation of wing and nacelle stall caused by disturbed onflow conditions and the experimental validation of the numerical results. Institutions in the Forschergruppe are the TU Braunschweig, the university of the armed forces in München, the German Aerospace Center in Braunschweig and Göttingen, the LU Hannover, the University Tübingen, Rolls-Royce Germany in Berlin-Dahlewitz and Airbus in Bremen. During the performance of this thesis, several cooperations with other subprojects have been realised: One such cooperation can be found in the backward facing step computations shown in Chapter 5 and of the HGR01 airfoil simulations presented in Chapter 6, which were realised in close cooperation with Axel Probst from DLR Braunschweig (see [89] for a comparison of the results shown here and the ones obtained by Axel Probst). David Hahn and Peter Scholz from TU Braunschweig currently perform the experimental validation of the numerical results shown in Chapter 8 - an FNG airfoil near stall with disturbed onflow conditions. Silvia Reuss from the German Aerospace Center in Göttingen performed DES simulations of the starting procedures of a F15 wing in [92]. Finally Torsten Auerswald from the University Tübingen applies the numerical simulation method presented in this thesis to transport a synthetically generated atmospheric turbulence field towards an airfoil, where the interaction eventually takes place (see [4]). He later plans to generate realistic turbulence by applying the tool *Meteogen*, which in combination with the simulation method would allow to numerically simulate realistic vortex-airfoil interactions.

1.2 Overview of the literature

This section provides an overview of the relevant literature for this thesis. As important topics air vortices and their interaction with solid bodies, hybrid RANS/LES methods and the Chimera technique have been identified. The major goal is to determine, whether a similar approach as the proposed simulation method has been introduced before or whether this is a truly new approach that has not been considered in literature yet. This question is addressed at the end of this section.

1.2.1 Air vortices and vortex-body interactions

At first literature considering experimental and numerical investigations of air vortices is presented. Fields of interest are particularly creation, merging and decay of vortices. Also papers concerned with vortex-body interactions are discussed at the end of Subsection 1.2.1.

Introductions

The following three sources provide good introductions into the topic of wake vortices and are therefore recommended as starting points. Gerz et al. [37] present a

”state of the art” in the field of wake vortices caused by commercial aircrafts. Important aspects are the characterisation and control of wake vortices, prediction and monitoring of vortex decay, vortex detection and warning and wake vortex safety assessment. In Holzäpfel et al. [49] a comprehensive overview of all relevant aspects of aircraft wake vortices can be found. Important issues are how these disturbances can be minimised, predicted, characterised, observed, detected and attenuated. Another introduction into the field of wake vortices and the resulting flight safety problems is given by Bobylev et al. [12]. The authors also introduce a mathematical model of wake vortices in the turbulent atmosphere, which accounts for wake evolution and destruction.

Experimental investigations

In this paragraph literature considering experimental vortex examination is provided. Whale et al. [125] experimentally investigate the vortex wake behind a two-blade wind turbine rotor by means of Particle Image Velocimetry (PIV). It is shown that the identified wake structures are in good agreement with additionally performed numerical simulations. Streamwise vortex structures in the wake of a stack, which is standing normal to the ground, are examined experimentally using a seven-hole pressure probe by Adaramola et al. [2]. Depending on the velocity, up to three pairs of counter-rotating vortex structures can be identified in the wake of the stack. How multiple-vortex systems in aircraft wakes can be detected experimentally is described by Carmer et al. [17]. The authors use PIV to obtain time-resolved velocity fields normal to the wake axis and use several vortex identification criteria for the detection of co-rotating and counter-rotating vortex pairs. Williamson [129] experimentally examines three-dimensional vortex structures in the wake of bluff bodies. Several vortex dynamics phenomena including vortex shedding, phase shocks, phase expansions, vortex dislocations, vortex loops and streamwise vortices are detected. Turbulent instabilities in the near- and far-wake of a delta wing are experimentally investigated by Miller et al. [80]. The experiments, which are performed both in a water tunnel and in a wind tunnel, reveal that in the near-wake small-scale structures evolve, which trigger large-scale vortices far downstream of the wing. Allen et al. [3] study vortex generation and evolution in the wake of a wing-tail configuration by means of hot-wire anemometry. They show that a four vortex system consisting of counter-rotating neighbored vortices evolves behind the generic airfoil model. Wu et al. [133] perform digital PIV measurements to detect shear layer vortices and longitudinal vortices in the wake of a circular cylinder. The experiments, which are carried out both in a wind tunnel and in a water tunnel, indicate that both types of disturbances evolve independently from each other. Additionally, longitudinal vortices appear already at much lower Reynolds numbers than shear layer vortices.

Numerical simulations

We proceed by papers dealing with numerical simulations of vortices. Ehrenstein et al. [29] compute a co-rotating vortex pair as equilibrium state of the Euler equations

to examine the influence of the two vortices on each other. By comparing to the case of a single vortex in an external deformation field it is shown that the influence of one vortex on the relative other can be approximated by an external rotating strain field whose characteristics are given by a point-vortex model, if both vortex centers are far enough away from each other. The merging of two analytical co-rotating vortices is examined numerically by means of Direct Numerical Simulation by Josser et al. [58]. By using a large range of Reynolds numbers (from $Re = 10^3$ to $Re = 10^6$) the effect of the Reynolds number on each of the three stages of merging is illustrated. The interaction of a two-dimensional trailing vortex-pair with a shear layer of surface air is examined numerically in [73]. While one vortex intrudes into the shear layer, the other one rebounds. Frech et al. [33] present and examine a probabilistic prediction scheme for wake vortex evolution in a convective boundary layer. A wake vortex warning system for Frankfurt Airport is introduced by Frech et al. [34]. As the runways of this airport lie relatively close to each other, the scheme especially aims to predict the horizontal winds caused by wake vortices of landing and starting airplanes. The last papers of this "numerics" paragraph are all provided by Holzäpfel et al. In [51] the evolution and decay mechanisms of wake vortices in the atmosphere are investigated numerically. While different initial conditions are applied, in all cases coherent secondary vortex structures evolve, which deform and weaken the primary vortex and in some cases lead to its rapid decay. [52] deals with methodologies, benefits and limitations of numerical simulations of evolution and decay of aircraft wake vortices in idealized and real environments. Several LES-results show that complex real scenarios can be accurately predicted. The decay of wake vortex pairs of a B-747 aircraft in an evolving and convectively driven atmospheric boundary layer is examined by means of LES-simulations in [50]. The authors especially propose an approach to adequately represent both the wake vortices and the convective boundary layer - despite the difficulty that their relatively characteristic length scales vary by a factor of 1000. In [48] the influence of stable stratifications on wake vortices are examined numerically in two dimensions, where the variations in time of descent height and speed, vortex spacing and circulation are of interest. The results provide an overview and detailed insight in two-dimensional wake vortex physics. In [47] a probabilistic real-time wake vortex decay and transport model is presented, which takes the effects of wind, turbulence, stable stratification and ground proximity into account.

Vortex-body interactions

At the end of Section 1.2.1 we present literature addressing vortex-body interactions, because this phenomenon is of major importance for this thesis. First investigations concerning a vortex-airfoil interaction are presented by Abate [1], where the collision of a single sinusoidal gust and a NACA 0012 airfoil is examined numerically in two dimensions. As this simulation requires a globally fine grid, the extension to a complex three-dimensional configuration seems hard to achieve. Svard et al. [115] numerically simulate the interaction of an analytical vortex with a NACA 0012 airfoil by means of high-order finite difference methods. The vortex is inserted into

the flow field at the left domain boundary and then transported towards the airfoil, where the vortex-airfoil interaction leads to a significant change in lift, drag and moment. The interaction of vortices emerging from the main and tail rotors of a helicopter with the flight vehicle are studied both experimentally and numerically by Coton et al. [22]. The interaction of a prismatic bluff body with a vortex wake is investigated theoretically, experimentally and numerically by Leclercq et al. [66]. The aim is to predict the lift and drag forces and the noise caused by the collision. A numerical simulation of the interaction of a pair of counter-rotating vortices and a rigid sphere is performed by Kim et al. [61], where both the vortices and the body are of comparable size. It is shown that the change in lift and drag of the sphere caused by the interaction with a single vortex is almost identical to the one due to the interaction with a vortex pair, if both vortex centers are located far enough away from each other.

1.2.2 Hybrid RANS/LES methods

The idea of hybrid RANS/LES methods is to combine the robustness and low numerical costs of RANS models with the high-resolution capabilities of LES methods. Ideally, the result should be comparably good as the outcome of an LES simulation, but with significantly reduced costs. While a variety of hybrid RANS/LES methods exist, we want to focus on Detached Eddy Simulation (DES, see also Subsection 3.4), because it is arguably the most popular approach and because it is used in this thesis. At the end of Subsection 1.2.2 also several zonal hybrid RANS/LES methods are presented.

Detached Eddy Simulation (DES)

DES is a nonzonal hybrid RANS-LES model that can be based on all commonly used RANS models. The idea is to compute attached boundary layers in RANS-mode and switch to LES as soon as the flow separates. The term "nonzonal" implies that the RANS-LES distribution is determined internally by the model and not by the user. As the "Detached" in "DES" already suggests, this model (and its modifications) has been designed to simulate flows containing a vast separation region. While applied to these testcases the model performs well, problems arise for cases with no or only mild separation. This model behaviour could also be observed within this thesis: As the testcase considered in Chapter 5 includes a large detached part, the use of DES leads to a convincing outcome. In contrast, the only small separation region in the example shown in Chapter 6 results in an inferior behaviour of DES.

The original concept of DES has been introduced 1997 by Spalart et al. [110], where the underlying RANS model is the Spalart-Allmaras model (see Subsection 3.3.1). One major drawback of the original DES-formulation turned out to be that the RANS-LES distribution is solely based on grid properties but not on the resulting flow field. As the attached boundary layer thickness is highly dependent on the Reynolds number and the Mach number, DES can only be reasonably used within a certain range of Reynolds and Mach number, if the underlying grid remains

unchanged. To overcome this flaw, the concept of Delayed DES (DDES) has been introduced by Spalart et al. [109] in 2006. As in DDES the RANS-LES distribution is also based on flow properties as the viscosity, the strain rate and the wall distance, this model can be applied for a wide range of Mach numbers and Reynolds numbers even if the grid remains the same. In 2008 Mikhail et al. [79] introduced a further modification called Improved DDES (IDDES). This model either operates as normal DDES or alternatively switches to a wall modelled LES (WMLES)-behaviour. IDDES was developed, because for certain testcases the use of DES or DDES leads to a mismatch between the modelled log layer and the resolved log layer, resulting in a wrong skin friction prediction. The concept of IDDES is to treat the large outer part of the attached boundary layer in LES-mode and only model a thin near-wall region by RANS.

An informative introduction into DES is given by Squires [111], where also several applications are shown. As all testcases considered include a vast region of separated flow, the results obtained by DES are superior to the ones obtained by URANS-computations. Spalart [107] provides a review of the first ten years of DES-methods. The author examines weaknesses and strengths of DES and its modifications and provides insight into the concepts of DES, DDES, IDDES and zonal DES. The response of DES to ambiguous grids, in which the wall-parallel grid spacing is of the order of the boundary layer thickness, is stated as main model weakness. The problem of high grid sensitivity is further discussed by Spalart [106], where basic principles of appropriate DES grid construction are provided. The use of DES can in particular lead to problems in the "grey area" of the grid, which is located in the interface of the RANS part and the LES part. Useful guidelines of how to implement DES methods based on different RANS models within a numerical code is described by Bunge et al. [15]. Morton [82] applies DES to compute vortex breakdown over a 70-degree delta wing. As the outcome is highly sensitive to the grid density, an adaptive mesh refinement is performed until good agreement with experimental data is obtained. The weakly nonlinear $k - \omega$ RANS model "WD+", DES and DDES (where both are based on WD+) are utilised by Fu et al. [36] to simulate wing-body junction flows. Only DDES is capable of resolving the large eddies detaching from the leading edge of the wing.

At this point we want to mention that the DLR AS-C²A²S²E-department in Göttingen performed and took part in several DES-examinations of one- and multielement airfoils with beginning trailing edge separation: After an extensive and time-consuming investigation the author of this thesis and Axel Probst from the German Aerospace Center in Braunschweig showed in [89] that problems can arise when using SA-DDES applied to the flow around a stalling HGR01 airfoil including a strong adverse pressure gradient. These limitations of SA-DDES, which are also subject to Chapter 6 of this thesis, are an important and new observation. In the projects ATAAC (Advanced turbulence simulation for aerodynamic application challenges) and GARTEUR (Group for aeronautical research and technology in Europe) AG 49, in which the DLR AS-C²A²S²E-department takes part, DES is applied to multielement airfoils. The need for these projects can be seen in the existence of many open questions when applying DES methods to multielement airfoils. Jakubek [55],

who performed her diploma thesis in the DLR AS-C²A²S²E-department, applied DDES to a three-element F15 airfoil. It was hypothesized that turbulent three-dimensional structures evolved in the slat-cove, which then moved along the main wing and intruded into the boundary layer, which distorted the RANS-part at the wing. The same probably occurred in the wake of the main wing, where turbulent structures distorted the attached boundary layer of the flap. Indeed this hypothesis still requires a much deeper investigation.

To sum it up, it can be stated that the use of nonzonal DES methods applied to multielement (and partly even one-element) airfoils near stall has not been fully understood yet even in the case of undisturbed onflow conditions. This task can therefore be considered as an open problem in literature that still contains many challenges. One idea to overcome these problems is the use of zonal DES methods (e. g. Deck [24] successfully performed a zonal DES to simulate the flow around a high-lift three-element configuration), which will be presented next.

Zonal hybrid RANS/LES approaches

Besides nonzonal DES methods, several zonal hybrid RANS/LES approaches have been introduced, in which the RANS/LES domains are explicitly prescribed. Fröhlich et al. [35] provide an extensive overview of different kinds of hybrid RANS/LES models including DES and its derivatives, two-layer models (in which the region between the wall and the LES domain is computed by a RANS-layer) and other zonal hybrid RANS/LES methods. As the various approaches are tested upon several testcases, an assessment of relative advantages and disadvantages of the methods can be obtained. A zonal hybrid RANS/LES method is applied to predict the flow over a high-lift configuration of a two-element airfoil at large Reynolds numbers by Zhang et al. [135]. Compared to a full LES computation, this approach allows a reduction of almost 50% of computational time. Richez et al. [93] investigate the stall mechanism of the flow around a helicopter fan blade profile by means of a zonal hybrid RANS/LES coupling method. The stall behaviour turns out to be highly sensitive to the correct prediction of a laminar separation bubble near the leading edge. While the hybrid approach minimises the numerical costs, the results are comparably good to the ones obtained by an LES simulation. Turbulent flow separation in a three-dimensional diffuser is modelled both by LES and a zonal hybrid RANS/LES method by Jakirlic et al. [56]. While the grid used for the hybrid computation contains only half the number of grid points as the one of the LES simulation, both results agree equally well with experimental data. Li et al. [67] apply a zonal hybrid RANS/LES method to simulate free surface flow through a vegetated channel. At the interface of the upstream RANS region and the downstream LES region artificial turbulent velocity fluctuations are superimposed to trigger instabilities. Benarafa et al. [8] present a zonal hybrid RANS/LES coupling approach, in which a forcing term is applied on the LES mean velocity field. By testing this method upon several examples, the authors show that this model performs well even if the Reynolds number is large and the grid relatively coarse.

We want to finish this subsection with the mentioning of the paper by Sanchez-Rocha

et al. [97], where a formal derivation of the compressible governing equations for the hybrid RANS/LES model is performed. As in most papers hybrid RANS/LES models are motivated by the need to simulate high Reynolds number flows with reasonable computing resources, the models are mostly constructed and examined as engineering models. Therefore this formal mathematical examination is uncommon and of great value.

1.2.3 The Chimera technique

This part considers literature dealing with the Chimera (or overset grid) technique (see Section 4.3), which is a type of domain decomposition method. When performing numerical simulations by means of Chimera, the computation is carried out not on a single but on several grids that can overlap and may be moved relative to each other. The flow data must be exchanged between the grids by an appropriate interpolation.

After a brief overview of papers first mentioning the overset grid technique, literature dealing with the two major fields of application are presented: While in most works the Chimera technique is used to simplify grid generation for complex geometries, more seldom overset grids are utilised to move solid objects or flow entities (such as vortices) relative to each other during unsteady simulations.

The first Chimera works

Composite grids were first used by Volkow [122, 123] (1966 and 1968) to solve the Laplace equation on regions with piecewise smooth boundaries separated by corners. Polar grids are fitted around every corner to handle potential singularities. It was not before 1977 and 1980 that the overset grid technique is again mentioned in literature, where Starius [112, 113] applied it to solve elliptic and respectively hyperbolic problems. The term "Chimera" was probably introduced by Benek et al. [9] in 1983, where the method was used to compute two-dimensional flow around a two-element airfoil. The name hereby refers to the mythical Chimera beast consisting of a human face, a lion's mane and legs, a goat's body and a dragon's tail and was chosen because the Chimera approach allows to combine highly differently shaped grids to a single overset grid. 1985 Benek et al. [10] used the Chimera technique to simplify grid generation by independently meshing wings, fuselage and elevator of a simplified airplane configuration. The component grids are finally embedded into a background grid.

Chimera applied to complex configurations

Next we consider literature, in which the Chimera technique is used to simplify grid generation for complex geometries. While the first three papers contain methods how to efficiently generate high-quality Chimera grids, the next sources adress solver issues. At the end of this "complex configuration"- part we present literature, in which the Chimera technique is applied to several testcases.

How the quality of Chimera grids for complex geometries can be improved by making use of genetic algorithms is stated by Carlsson et al. [16]. The approach is validated upon the outcome of a simple two-dimensional Poisson problem. Zheng et al. [136] provide an alternative to the standard Chimera grid generation for complex geometries, where large parts of the computational domain are covered by unstructured grids. The idea of the "DRAGON methodology" is to use structured grids in the main part of a domain and only fill small remaining gaps with unstructured Chimera grids. Overall this approach minimises the required memory and allows the use of more efficient solvers on the increased structured grid part. Chan et al. [19] describe an efficient approach to generate surface grids for complex geometries. The method, which makes use of hyperbolic partial differential equations and an algebraic marching scheme, significantly reduces the grid generation time.

The following sources provide insight into questions regarding the flow solver. Schwarz [99] introduces and implements a block structured method to simulate viscous flow around complex configurations. Validation testcases include a three-element airfoil, a helicopter fuselage and an airplane in landing configuration. It is shown that the grid generation by means of Chimera is easier and results in less deformed grid cells compared to the creation of a single grid. Banks et al. [6] present a high-resolution Godunov method for high-speed multi-material flow including shocks. While the Chimera technique is utilised to describe complex geometries, also a block structured adaptive mesh refinement algorithm is used to capture fine flow structures such as shocks and material interfaces. A method to efficiently parallelize complex three-dimensional flow problems, which include overlapping grids and an adaptive mesh refinement is presented by Henshaw et al. [45]. Several applications of an overset-adaptive grid method used to compute compressible flow around complex geometries are given by Matsuno et al. [70]. Banks et al. [5] introduce a flux-corrected transport algorithm for high-speed flows on structured overlapping Chimera grids. The latter are required to represent complex geometries as well as ensure an appropriate mesh regularity. The approach is successfully tested and validated upon several testcases such as a shock impingement on a stationary cylinder and irregular Mach reflection on an inclined ramp. A theoretical investigation and two applications of a high-order overlapping grid method that allows to couple cartesian and curvilinear grids are performed by Desquesnes et al. [25]. The method is based on Lagrange interpolation polynomials and is successfully validated upon the simulation of noise generated by vortex shedding behind a cylinder and of a three-element high-lift airfoil. Henshaw et al. [43] give a robust numerical method to solve the reactive Euler equations, which allows to consider high-speed reactive flows including generation, propagation and failure of detonation waves. Overset grids and a block structured adaptive mesh refinement are used to represent the complex geometry and locally increase the resolution in the thin reaction part of the domain. Tang et al. [118] employ overset grids to discretise highly complex geometries for three-dimensional unsteady incompressible flows. Also a new interface algorithm that simplifies data exchange between neighboring Chimera grids and that guarantees global mass conservation is introduced.

In the next five papers overset grids are used to compute flow around complex

configurations. As both the wing-body-aileron-spoiler configurations examined by Mertins et al. [78] and the satellite launcher configuration used by Basso et al. [7] are highly sophisticated, the generation of adequate single grids seems almost impossible. Therefore in both papers a Chimera approach is used to account for the complex geometries. The resulting numerical outcomes agree well with experimental data. The time-dependent Maxwell's equations in a complex geometry are solved using a high-order accurate parallel solver by means of overlapping grids by Henshaw [42]. The interface between materials with different electric and magnetic properties are accounted for by using Chimera grids that align with the interface. Liao et al. [68] generate block structured Chimera grids to compute the flow around sophisticated three-element airfoils and wing-body configurations. Numerical simulations of high-lift multi-element airfoils are presented by Rogers [94]. The Chimera technique is here utilised to simplify the grid generation of a three-element airfoil, which is either in takeoff or landing configuration. Simonsen et al. [102] simulate manoeuvres of the tanker "Esso Osaka", where overset grids are used to simplify the grid generation and the realisation of a grid convergence study.

Movement of bodies or flow entities

As mentioned earlier, the Chimera technique is less frequently used to move bodies or flow entities relative to each other. While this task is mostly performed by translating or rotating Chimera grids, in the last two mentioned sources the positions of all Chimera grids remain unchanged throughout the complete simulation.

Dietz et al. [26] conserve and transport tip vortices of a helicopter main rotor by means of moving vortex-adapted overset grids that lie embedded within a background grid containing the helicopter. The Chimera technique is used to simulate the separation of the Apollo Launch Escape Vehicle from a rocket booster by Pandya et al. [85]. An overview of the development of overset grid technologies at NASA Ames Research Center to model complex geometries and simulate multiple bodies in relative motion is presented by Chan [18]. Henshaw et al. [44] utilise overset grids in combination with adaptive mesh refinement to solve the chemical reactive and non-reactive Euler equations to model high-speed flow. This approach is required because the geometry evolves in time and also fine-scale structures such as shocks and detonations can occur. The four main possibilities to simulate flow around moving bodies - the "sliding mesh" technique, the "fictitious domain" method, the "arbitrary-Lagrangian-Eulerian" technique and the Chimera approach - are illustrated and compared by Houzeau et al. [54]. In Section 4.2 of this thesis these alternatives are examined in more detail. Togashi et al. [119] perform a simulation of a rocket booster separation from a supersonic airplane using overset grids. An unstructured grid representing the rocket booster is hereby moved in time relative to a stationary airplane grid. Zhang et al. [134] introduce and apply a pressure-correction method to compute the incompressible flow around rotating circular and elliptic cylinders. The rotation is realised by moving a Chimera grid containing the body relative to a steady unstructured background grid. A method to parallelly solve large aerodynamic problems containing Chimera grids to track moving objects

is presented by Prewitt et al. [88]. The approach is successfully validated upon a complex testcase, in which three bombs are released from an airfoil. Pahlke et al. [84] perform Chimera computations of multibladed helicopter rotors in high-speed forward flight by the DLR codes S4 and FLOWer. To simulate the circular rotor movement, four grids that each contain one of the four blades are embedded and rotated within a cartesian background grid. Numerical simulations of flows around helicopters at DLR and ONERA are shown by Sides et al. [101]. While the ONERA method uses a deforming grid strategy to accomplish the rotational movement of the blades, the DLR approach is realised by overset blade grids that are moved relative to a background grid. Both approaches provide comparably good results compared to experimental data. Sitaraman et al. [103] introduce a module to support overset grid assembly to perform time-dependent and adaptive moving body simulations of external aerodynamic flows. The approach is validated upon a testcase of a three-element airfoil, where the full retraction of both slat and flap is realised by means of the Chimera technique. Hariharan [41] captures and transports the tip vortex of a NACA 0015 airfoil within the wing wake. To resolve the vortex both high-order accurate spatial discretisations and a wing grid - vortex grid overset system are used. The fine vortex grid, which extends 18 chord lengths behind the airfoil, is required because the use of structured grids prohibits a grid adaptation in the wake region. It is important to note that - contrary to the simulations in Chapters 7 and 8 of this thesis - the vortex grid does not move. It therefore has to cover the complete vortex transport distance, which is highly inefficient.

We want to finish with the mentioning of the paper by Struijs et al. [114], where a similar experimental setup to the one presented in Chapter 8 of this thesis is examined. First a wake vortex generating airplane model is used to create a pair of counter-rotating wing tip vortices, which are transported with the freestream velocity to the right. The vortices finally interact with a model airplane, which is approximately half the size of the upstream model. To compare the results of this experiment, which has been performed in the German-dutch wind tunnel large low-speed facility (DNW-LFF) with a numerical simulation, the vortices are modelled by simple analytical functions that are used as inflow boundary conditions. The simulation therefore only consists of a stationary Chimera grid containing the downstream airplane which lies embedded within a background grid. The results of the simulation agree comparably well with the experimental data, although the lift is found to be overestimated. The authors constitute this discrepancy to the use of an inviscid solver as well as an unphysical vortex dissipation caused by too coarse grids.

1.3 Structure and main results of the thesis

This subsection provides an overview of the structure of this thesis, which is divided into two parts: While Chapters 2 - 4 consider mathematical methods, in Chapters 5 - 8 these methods are applied to numerical applications. Also the main results of this thesis are addressed in Section 1.3.

Subject of Chapter 2 are numerical methods in the DLR TAU-code. First the mathematical model consisting of the governing partial differential equations describing compressible flow - the compressible Navier-Stokes equations - and the boundary conditions are introduced. When performing numerical simulations these equations must be discretised both in space and time, which is examined next. Finally two convergence acceleration techniques - preconditioning methods and multigrid methods - are described. Throughout the chapter a testcase of a decaying analytical vortex at rest is presented several times, where the spatial refinement, the time step size, the use of low Mach number preconditioning and the application of different multigrid methods are varied. The aim is to study the effects of choosing different parameter settings on the vortex dissipation and the convergence rate.

For high Reynolds numbers the compressible Navier-Stokes equations become turbulent. This makes a direct numerical simulation (DNS) of these equations unaffordable in terms of computer resources, because in a DNS all flow scales have to be resolved, which requires an extremely fine grid and a very small timestep to capture all small turbulent structures. To avoid this problem, the concept of turbulence modeling is introduced, which is subject of Chapter 3. The idea lies in performing a scale separation into small and large scales, which is addressed at the beginning of this Chapter. Then the following three approaches are possible:

- The compressible Navier-Stokes equations can be averaged, which results in the Unsteady Reynolds Averaged Navier Stokes (URANS) equations. In this approach no scales are resolved, but all are modeled by turbulence models.
- The compressible Navier-Stokes equations can be filtered, which leads to the concept of Large Eddy Simulation (LES). While the large scales are resolved, the small scales are modeled by a subgrid scale model.
- Hybrid RANS/LES models, which are addressed at the end of Chapter 3, are a combination of URANS and LES.

As in this thesis only URANS and hybrid RANS/LES models are applied, LES models are not explicitly addressed in this chapter.

One major component of the simulation method considered in this thesis is the Chimera technique, which is a sort of domain decomposition (DD) method. In Chapter 4 first a general introduction into DD methods is provided. The concept lies in dividing a computational domain into several subdomains and perform the solution algorithm independently on each part. The solution must then be adequately exchanged at the domain interfaces, where the latter can be overlapping or non-overlapping. As in this thesis solid objects as well as vortices have to be transported relative to each other, next the four main alternatives to perform this task are presented. At the end of this Chapter the implementation of the Chimera technique in the DLR TAU-code is considered.

In Chapter 5 the three-dimensional flow over a backward facing step is examined by means of the Spalart-Allmaras Delayed Detached Eddy Simulation (SA-DDES) model, which is a hybrid RANS/LES turbulence model. As this setup comprises a large region of small-scale detached flow, this is a standard testcase for hybrid

RANS/LES models. A grid convergence study is performed, in which three SA-DDES simulations on a coarse, medium and respectively fine grid are carried out. The results are compared to experimental data and a steady two-dimensional SA-RANS computation on the medium grid. One important variable is the skin friction coefficient, because it determines the reattachment point behind the step. Figure 1.2 (left) shows that both the SA-DDES simulation on the medium and fine grid (indicated by letters "M" and respectively "F") lie in good agreement with experimental data, while the SA-RANS result and especially the coarse SA-DDES outcome ("C") are far from being accurate. Another important question is whether the SA-DDES model is capable of resolving three-dimensional small-scale structures and which grid refinement is required to obtain them. It turns out that both the SA-DDES computations on the medium and on the fine grid result in three-dimensional structures, while the SA-DDES computation on the coarse grid does not. Figure 1.2 (right), which shows isosurfaces of the instantaneous Q-criterion at a value of $Q = 200\frac{1}{s^2}$ based on the fine SA-DDES computation, visualises this observation. In the second part of Chapter 5 two sensors are introduced and evaluated for this testcase. While the first sensor indicates, in which regions the flow tends to become unsteady, the second sensor shows, if the grid is fine enough to resolve a sufficient amount of turbulent kinetic energy in the LES-region.

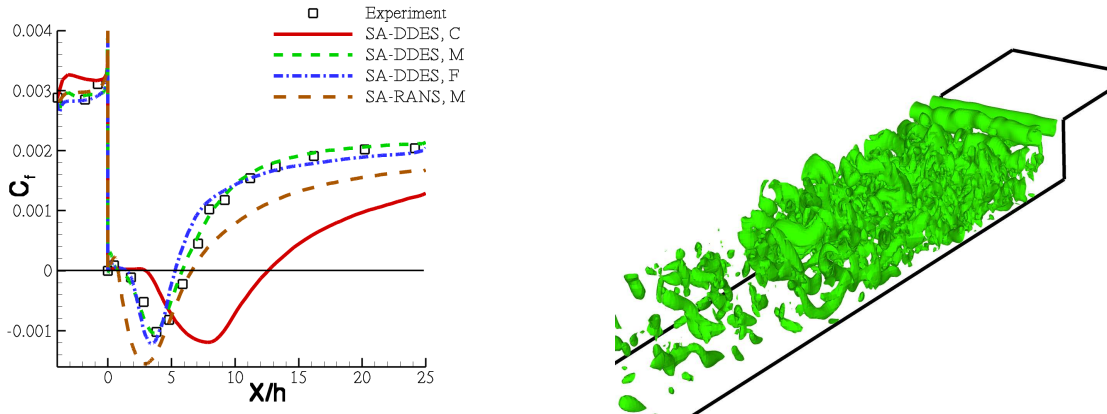


Figure 1.2: Backward facing step testcase in Chapter 5: Skin friction coefficients of all performed computations and comparison with experimental data (left) and isosurfaces of the instantaneous Q-criterion at $Q = 200\frac{1}{s^2}$ based on the fine SA-DDES computation (right).

In Chapter 6 an HGR01-airfoil in the range of stall is considered. As in this testcase only a small trailing edge separation occurs, first the reliability of the SA-DDES-model for the flow at $\alpha = 12^\circ$ is tested. When using the standard SA-DDES model, the RANS-part at the wing does not cover the complete attached boundary layer, leading to modeled stress depletion and grid induced separation. Therefore a model modification called SA-DDES₁₆ is performed, which results in a larger RANS-part. Figure 1.3 (left) shows the resulting RANS-LES distribution and the velocity streamlines of the SA-DDES₁₆ computation, where "red" indicates RANS-mode,

while in the "blue" area LES-mode is activated. Compared to the boundary layer thickness, which is indicated by the Mach number in Fig. 1.3 (right), one can see that only the outer boundary layer part is computed in LES-mode. As a result the velocity streamlines are almost fully attached, which is desirable. The SA-DDES₁₆ model is applied to the whole range of stall angles at $\alpha = 12^\circ - 16^\circ$ and the results are compared both to steady two-dimensional SA-RANS simulations and experimental data.

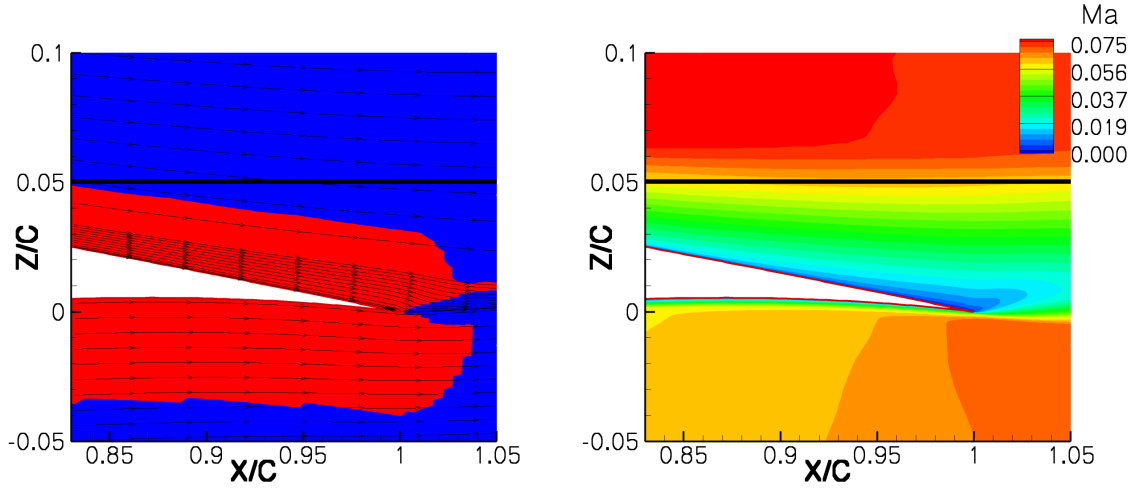


Figure 1.3: HGR01 testcase in Chapter 6: RANS-LES distribution and velocity streamlines (left) and boundary layer thickness indicated by Mach number (right). Both pictures are based on the SA-DDES₁₆ model at $\alpha = 12^\circ$.

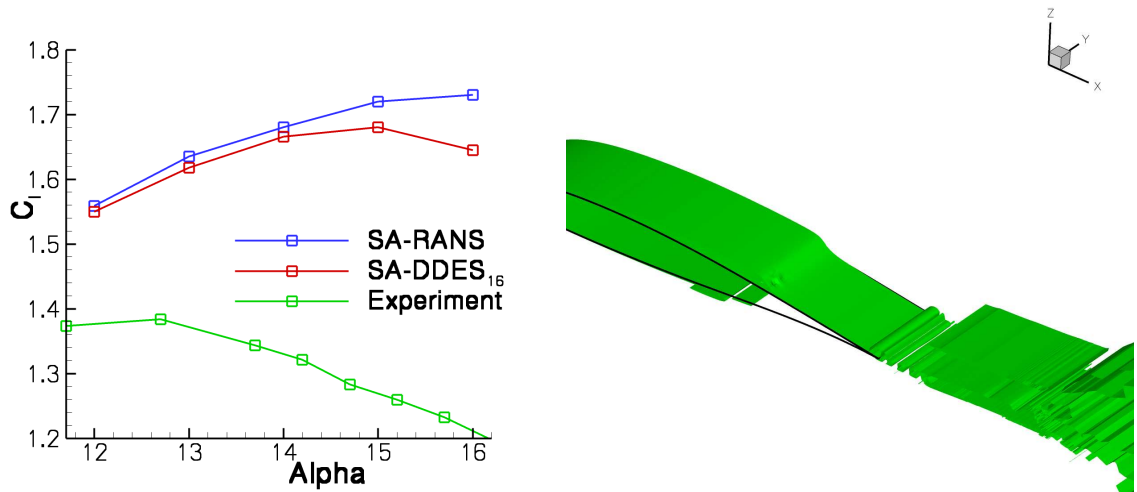


Figure 1.4: HGR01 testcase in Chapter 6: Lift coefficients (left) and isosurfaces of the instantaneous Q-criterion at $Q = 10^{-5} \frac{1}{s^2}$ of SA-DDES₁₆ computation at $\alpha = 12^\circ$ (right).

Figure 1.4 (left), which shows the lift coefficient of both models and of the experimental data, reveals that both models behave similar and clearly overestimate C_l .

As in this testcase the separation region is relatively small, no three-dimensional turbulent structures can be obtained. This is indicated in Fig. 1.4 (right), in which isosurfaces of the instantaneous Q -criterion at $Q=10^{-5}\frac{1}{s^2}$ based on the SA-DDES₁₆ simulation at $\alpha = 12^\circ$ can be seen.

In Chapter 7 vortex transports and vortex-airfoil interactions are simulated by means of two-dimensional URANS computations. In all examples first one or more analytically defined vortices are interpolated onto a fine cartesian vortex transport grid, which is then moved relative to a background grid by means of the Chimera technique. In the first example it is tested, if a single vortex can be transported over a large distance when both a fine vortex transport grid and a fine background grid are combined. In the second and third example it is examined whether the vortex dissipation significantly increases, if a coarse background is used and when not one but three vortices are transported. Only the fourth and fifth example contain a vortex-airfoil interaction, where a vortex collides with a NACA 0012 airfoil at $\alpha = 0^\circ$ and an ONERA-A airfoil at $\alpha = 13.3^\circ$, respectively. Figure 1.5 (left) shows the local velocity (defined as the absolute value of the velocity relative to the freestream velocity) of the vortex and around the airfoil before the vortex-airfoil interaction of the ONERA-A testcase. Also the boundary of the vortex transport grid is displayed. One can see in Fig. 1.5 (right) that the subsequent vortex-airfoil interaction has a massive effect on C_l , C_d and C_m . The immense breakdown in C_l indicates wing stall.

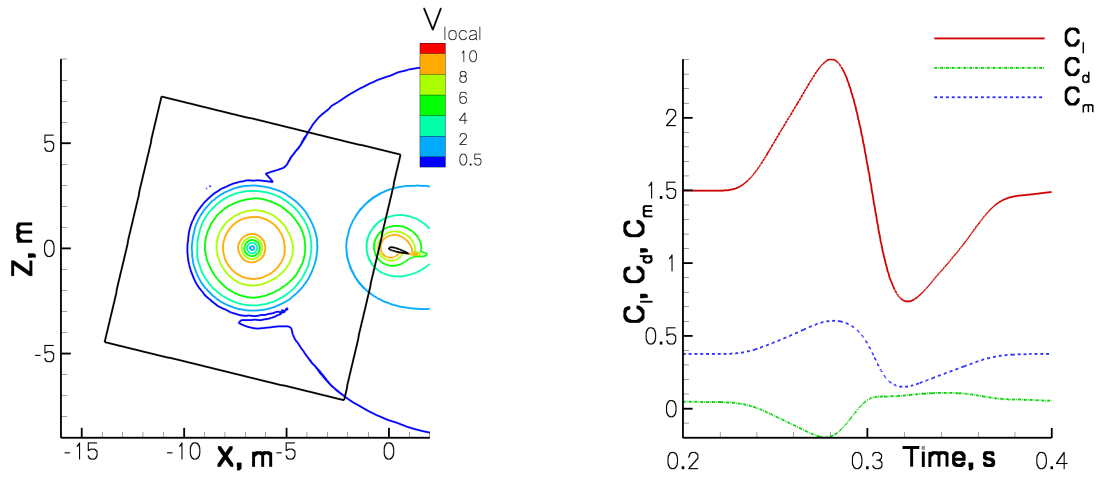


Figure 1.5: ONERA-A testcase in Chapter 7: Vortex transport grid and local velocity before vortex-airfoil interaction (left) and effect of vortex interaction on C_l , C_d and C_m (right).

Applications of the complete simulation method introduced at the end of Section 1.1 are given in Chapter 8. While a NACA 0021 airfoil is used to generate the vortex, the latter eventually interacts with a two-element FNG airfoil at $\alpha = 10^\circ$, which is close to stall. Beforehand, stationary two-dimensional RANS simulations of the FNG airfoil at undisturbed onflow conditions are shown, where both the effect of the turbulence model and the angle of attack resulting in stall are examined. Also

a grid convergence study is performed. In Section 8.2 four two-dimensional URANS simulations of vortex generation, vortex transport and vortex-FNG interaction by means of the simulation method are presented. Both the effect of varying the turbulence model and the use of a small or alternatively large airfoil-to-airfoil distance is analysed. While it turns out that in the two "small distance" computations the breakdown in C_l caused by the vortex interaction is significantly larger than in the two "large distance" simulations, in neither case stall occurs. The two turbulence models result in qualitatively similar results. The last two Subsections each contain a three-dimensional application of the simulation model, where an SA-URANS and alternatively SA-DDES model are used to perform a "small distance" computation, respectively. Figure 1.6 (left), which is based on the SA-DDES result, shows the three-dimensional setup before the vortex-airfoil interaction. Also the isosurfaces of the vorticity at a value of 100, which are additionally colored by the Mach number, are shown. One can identify the vortex located on the cuboid vortex transport grid. Figure 1.6 (right) displays the isosurfaces of the Q-criterion at a value of $100\frac{1}{s^2}$ during the vortex-airfoil interaction at the trailing edge of the FNG airfoil. While a large separation region containing three-dimensional turbulent structures has evolved, it remains unclear whether this is due to the vortex interaction. As in the two-dimensional testcases of Section 8.2 and in the three-dimensional SA-URANS simulation, no stall caused by the vortex-airfoil interaction can be verified.

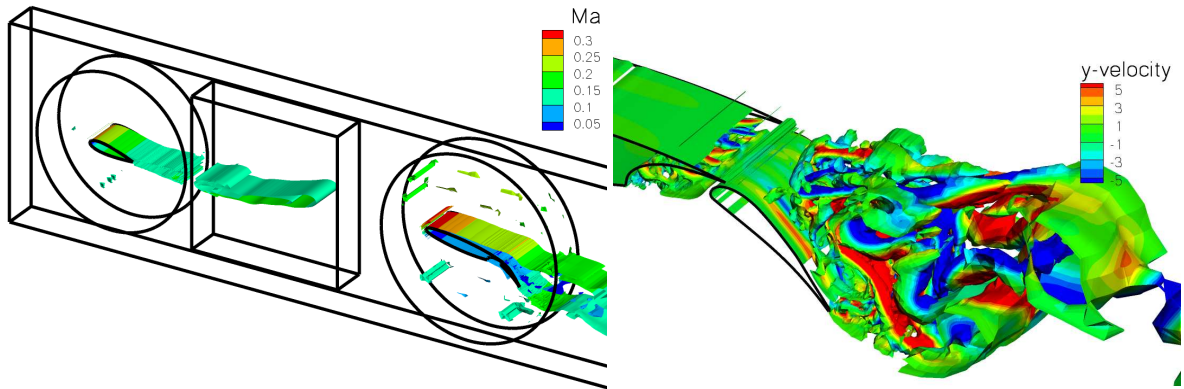


Figure 1.6: SA-DDES testcase in Chapter 8: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) before vortex-airfoil interaction (left) and isosurfaces of the instantaneous Q-criterion at a value of $100\frac{1}{s^2}$ (colored with y-velocity) during vortex-airfoil interaction (right).

At the end of this thesis a summary of the results and open questions for future work are provided in Chapter 9.

1.4 New aspects of the thesis

At this point we want to compare the topics covered by the literature introduced in Section 1.2 with the methods and results shown in this thesis. The aim is to determine, what has been done before in a similar way and what is truly new.

Regarding Chapter 5 it is stated that similar SA-DDES results of the flow over a backward facing as the ones shown in Section 5.2 are manifold in literature; see for example the paper by Spalart et al. [109]. Turning to Section 5.3 it can be said that the idea to use the von Kármán length scale as sensor to detect flow instationarities is similarly applied in the SST-SAS turbulence model introduced by Menter et al. [76]. The second sensor *Indicator* has been introduced and tested by Knopp et al. [64]. Overall, no truly new content can be found in Chapter 5.

In Chapter 6 some new aspects are present. While DES models have been applied before to simulate flow around airfoils at stall (see for example the papers by Spalart et al. [107, 109]), we are not aware of any sources in literature adresssing the observed problems of the standard SA-DDES model. Also the proposed modified SA-DDES₁₆ model is a new approach, although this modification is admittedly rather an ad-hoc fix, which can not expected to be valid in general. The results, the observed problems of the standard SA-DDES model and the model modification applied within this testcase have been published in [89].

Almost all results shown in Chapters 7 and 8 are based on truly new approaches. To the authors's knowledge no comparable simulation method to generate and transport realistic vortices and to model their interaction with solid bodies has been introduced and applied in literature yet. Also the combination of the simulation method and a hybrid RANS/LES model, which is performed in Section 8.4, has not been done before.

Part I

Mathematical methods

Chapter 2

Numerical methods in the DLR TAU-code

This chapter considers the numerical methods of the DLR TAU-code [98], which is a compressible flow solver based on the finite volume method for the three-dimensional flow equations in integral form. Besides a variety of one- and two-equation turbulence models to solve the Reynolds Averaged Navier-Stokes (RANS) equations, also Large Eddy Simulation (LES) and Detached Eddy Simulation (DES) can be applied. As all flow variables are stored at the cell vertices (or respectively at the cell centers with respect to the dual grid cells) the code is independent of the type of grid cells, which allows the use of unstructured, structured or hybrid grids. For the discretization of the convective fluxes, several upwind or central discretization methods are available. Multigrid methods and preconditioning can be used for convergence acceleration.

2.1 The mathematical model for compressible flow

The mathematical model for compressible flow consists of the governing equations, which are valid in the interior of a domain Ω , and boundary conditions that have to be fulfilled at the boundary of Ω . We begin by considering the governing equations.

2.1.1 The compressible Navier-Stokes equations

The starting point are the compressible Navier-Stokes equations, which can be written in conservative form as follows. Note, that we use the Einstein notation, in which a summation is performed, if two indices are identical in an expression.

Definition 2.1.1 (Compressible Navier-Stokes equations)

For a bounded domain $\Omega \subset \mathbb{R}^3$ and time $t \in (0, \tau)$ we seek density $\rho(x, t)$, velocity $\vec{u}(x, t)$, and temperature $T(x, t)$ such that:

$$\frac{\partial}{\partial t} \rho + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial}{\partial x_i}p + \frac{\partial}{\partial x_j}t_{ji} \quad (2.2)$$

$$\frac{\partial}{\partial t}E + \frac{\partial}{\partial x_j} \left[\rho u_j \left(h + \frac{1}{2} u_i u_i \right) \right] = \frac{\partial}{\partial x_j} (u_i t_{ij}) - \frac{\partial}{\partial x_j} q_j \quad (2.3)$$

Equations (2.1), (2.2) and (2.3) are the *continuity equation*, *momentum equation* and *energy equation*.

For a Newtonian fluid the viscous stress tensor is given by:

$$t_{ij} = 2\mu S_{ij} - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \delta_{ij}, \quad (2.4)$$

where μ is the molecular viscosity, δ_{ij} denotes the Kronecker delta and the strain-rate tensor S_{ij} is given by:

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \quad (2.5)$$

$p(\rho, T)$ denotes the pressure and $E(x, t)$ is the energy density defined by:

$$E = \rho \left(e + \frac{1}{2} u_i u_i \right) \quad (2.6)$$

We are assuming that the fluid is calorically perfect, meaning that its specific-heat coefficients are constant, such that the specific internal energy e and the specific enthalpy h are given by:

$$e = c_v T \quad \text{and} \quad h = c_p T = e + \frac{p}{\rho}, \quad (2.7)$$

where c_v and c_p are the specific-heat coefficients for constant volume and pressure processes, respectively.

Using Fourier's law, the heat flux q_j can be written as:

$$q_j = -\kappa \frac{\partial}{\partial x_j} T \quad (2.8)$$

where κ is the thermal conductivity.

To close Eqs. (2.1) – (2.3), we also need an equation of state, which is typically the perfect gas law:

$$p = \rho R T \quad (2.9)$$

where $R = c_p - c_v$ is the perfect gas constant.

2.1.2 Boundary conditions

This part considers boundary conditions that have to be fulfilled at the boundary $\partial\Omega$ of domain Ω . Note, that we focus on those conditions that are actually used in part II of this thesis. As the boundary condition within the Chimera technique will

be explained in Section 4.3, we will not consider it at this point. Besides the ones shown here, a variety of additional boundary conditions to choose from is available in the DLR TAU-code. As the exact definition of each boundary condition varies for different turbulence models, the description provided here is far from complete and should rather be seen as an overview.

Viscous wall

We want to start with the **viscous wall** condition, which is applied at the boundary of a solid body or a wall, where both viscous and inviscid effects are accounted for. For temperature and density the adiabatic wall condition is chosen. Overall, the following conditions for velocity, density and temperature are prescribed when choosing the viscous wall condition:

$$\vec{u} = 0, \quad (\vec{\nabla} T) \cdot \vec{n} = 0, \quad (\vec{\nabla} \rho) \cdot \vec{n} = 0$$

Inviscid/Euler wall

When utilising the **inviscid/Euler wall** condition, only the inviscid parts are considered, while all viscous effects are neglected. This condition is achieved by setting:

$$\vec{u} \cdot \vec{n} = 0, \quad (\vec{\nabla} T) \cdot \vec{n} = 0, \quad (\vec{\nabla} \rho) \cdot \vec{n} = 0$$

Farfield

The **farfield** condition (see [126]) should be used in the case of undisturbed onflow. Here one has to distinguish between the inflow part and the outflow part. According to the theory of characteristics for hyperbolic partial differential equations, four variables have to be specified at the inflow boundary and the remaining variable has to be taken from the first inner field point. At the outflow boundary it is the other way around. To illustrate this, we consider Fig. 2.1, which shows a simple computational domain. For simplicity we assume a flow direction from left to right and set $\vec{u} \cdot \vec{n} = 0$ both at the upper and lower boundary C and D . When choosing the farfield condition at boundary A and B , then A belongs to the inflow part, whereas B corresponds to the outflow part.

At the inflow plane A we prescribe:

$$\vec{u} = \vec{u}_\infty, \quad T = T_\infty$$

and ρ gets a value extrapolated from the first inner grid point of the domain.

At the outflow plane B we set:

$$\rho = \rho_\infty$$

and both \vec{u} and T get values extrapolated from the first inner grid point of the domain.

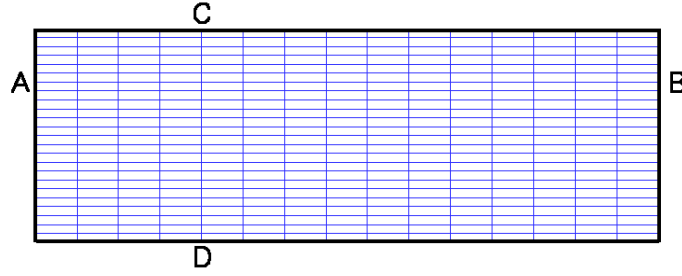


Figure 2.1: Simple computational domain. For simplicity we consider a flow direction from left to right and prescribe $\vec{u} \cdot \vec{n} = 0$ at C and D .

Dirichlet

The **Dirichlet** condition can be used at the inflow to prescribe user-defined flow variables \vec{u}_u and T_u . Analogous to the farfield condition, the Dirichlet condition (and also the following subsonic outflow condition) is based on the theory of characteristics, which implies that four variables must be set and one is taken from the interior of the domain. The prescribed values, which have to be obtained from experimental data, may vary along the boundary but should be constant in time. Within this thesis the Dirichlet condition is used for the backward facing step computations shown in Chapter 5 to guarantee a correct boundary layer thickness at the step.

If in Fig. 2.1 the Dirichlet condition is chosen at the inflow plane A , then the following must be specified:

$$\vec{u} = \vec{u}_u, \quad T = T_u$$

and ρ is extrapolated from the interior of the domain.

Subsonic outflow

When using the Dirichlet boundary condition at the inflow, the **subsonic outflow** condition should be used at the outflow.

Referring to Fig. 2.1, this is realised by setting

$$\rho = \rho_u$$

at all points of plane B . \vec{u} and T are extrapolated from the interior of the domain. ρ_u is a user-prescribed density, which can vary for each grid point along the boundary but is constant in time. Within the unsteady backward facing step computations ρ_u was taken from the result of a previously performed steady computation.

Symmetry planes

When using the **symmetry plane** condition, all fluxes in wall-normal direction are zero, i. e.

$$\vec{u} \cdot \vec{n} = 0, \quad \rho \cdot \vec{n} = 0, \quad T \cdot \vec{n} = 0, \quad (\vec{\nabla} T) \cdot \vec{n} = 0, \quad (\vec{\nabla} \rho) \cdot \vec{n} = 0$$

Periodic planes

Using the **periodic plane** condition is only possible for two (axis or point) symmetric boundary planes simultaneously. The principle is illustrated in Fig. 2.2: If the periodic plane condition is applied for boundaries A and B , artificial planes A_{-1} and B_{-1} are introduced. All primitive variables and gradients at A_{-1} are copied from plane B_1 , which is the first inner plane left of plane B . Similarly the values from plane A_1 are copied to the new plane B_{-1} . By performing this procedure the original boundary planes A and B must not be treated as boundaries anymore, but can be considered as part of the interior of the domain.

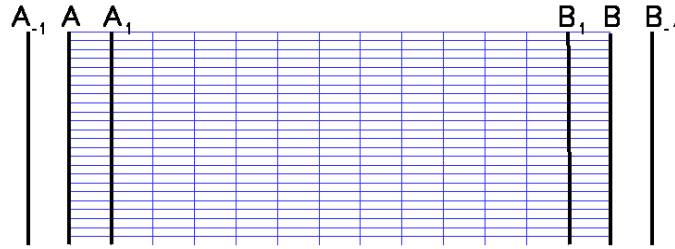


Figure 2.2: Simple computational domain to illustrate the principle of the periodic plane condition.

A global existence result of the Navier-Stokes-Fourier system

Having introduced the model, we want to state an important existing result that has been published by Feireisl et al. [31] in 2009. The existence of a weak solution globally in time for the Navier-Stokes-Fourier system was proven. While the theorem and complete proof can be found in [31] (theorem 3.1 on p. 50 ff.), we present the theorem in Appendix A. As the Navier-Stokes-Fourier system is a generalisation of the compressible Navier-Stokes equations defined in Eqs. (2.1) – (2.3) (this is shown in Appendix A), this existence result is also valid for the system of equations used in the DLR TAU-code.

2.2 Spatial discretisation

The next step to perform numerical simulations lies in discretising Eqs. (2.1)-(2.3) both in space and in time. We first consider the spatial discretisation.

2.2.1 Primary grid and dual grid

First a suitable discretisation of the computational domain Ω is needed, the so-called primary grid. It consists of a set of n grid points x_i ($i = 0, \dots, n - 1$). The primary grids of the DLR TAU-code can contain four different types of volume elements: tetrahedrons, pyramids, prisms and hexahedrons. These are obtained by connecting the grid points x_i by means of faces, such that the unification of all elements form a non-overlapping decomposition of Ω .

Figure 2.3 shows a two-dimensional sketch of a primary grid consisting of triangles, which are the analogon to tetrahedrons in two dimensions. The faces are indicated by solid lines. Equations (2.1) – (2.3) are then solved on the so-called dual grid, which is established during the preprocessing routine of the DLR TAU-code. By connecting the center points of the cells of the primary grid, one obtains the dual grid consisting of dual cells (or control volumes) V_i around grid point x_i of the primary grid. The shape of two dual cells V_0 and V_1 around grid points x_0 and x_1 are indicated by dashed lines in Figure 2.3.

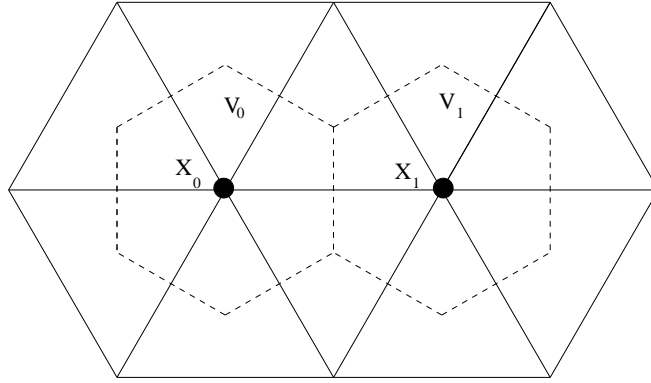


Figure 2.3: Two-dimensional sketch of primary grid cells (solid lines) and the corresponding dual grid cells (dashed lines).

Figure 2.4 shows a larger cutout of a primary grid (red lines) and the corresponding dual grid (grey lines). By using the primary and the corresponding dual grid, geometries of unrestricted complexity can be discretised.

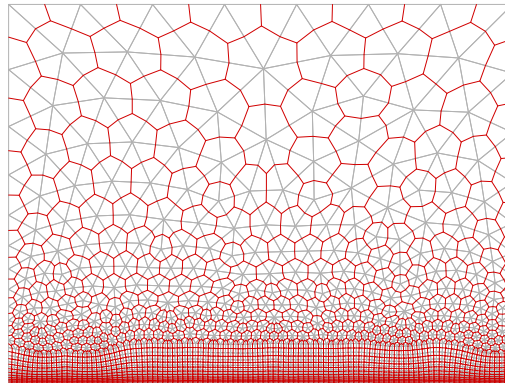


Figure 2.4: Primary grid (red lines) and corresponding dual grid (grey lines).

2.2.2 Upwind scheme

Equations (2.1)–(2.3) can now be discretised in space on the dual grid. Depending on the order of the discretisation this is realised by an upwind scheme or a central scheme. Note, that for simplicity both schemes are only explained for cartesian grids. To illustrate the principle of upwind schemes, we consider the continuity equation (2.1), where the indices are omitted for better readability:

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho u) = 0$$

If the velocity u is regarded constant, Eq. (2.1) becomes a linear convection equation, which describes the transport of mass ρ by a flow of velocity u :

$$\frac{\partial}{\partial t}\rho + u\frac{\partial}{\partial x}\rho = 0 \text{ or abbreviated : } \rho_t + u\rho_x = 0 \quad (2.10)$$

Now let V_i be the control volume around grid point x_i . Then ρ_i is given by:

$$\rho_i = \frac{1}{|V_i|} \int_{V_i} \rho \, dV \quad (2.11)$$

When now applying a backward (or upwind) differencing scheme in space, one obtains the following first order upwind scheme for V_i :

$$\frac{\partial}{\partial t}\rho_i = -\frac{u}{\Delta x}(\rho_i - \rho_{i-1}) \quad (2.12)$$

This procedure is valid, if $u > 0$ and constant.

Note, that the upwind scheme is not applied in the examples shown in part II of this thesis, because the central scheme is much more accurate.

2.2.3 Central scheme with artificial dissipation

While the upwind scheme is a spatial discretisation of first order, the central scheme [71] includes a second order spatial discretisation. We again start with Eq. (2.10). When now applying a second order difference formula for the spatial derivative on V_i , one obtains the following second-order central scheme in space:

$$\frac{\partial}{\partial t}\rho_i = -\frac{u}{2\Delta x}(\rho_{i+1} - \rho_{i-1}) \quad (2.13)$$

For the momentum equation (2.2) and energy equation (2.3) the spatial discretisation can be done similarly either with an upwind scheme or a central scheme.

Artificial dissipation

Unfortunately the just described central scheme leads to unphysical oscillations and hereby makes the scheme numerically unstable. Therefore a modification is required, which lies in adding an additional term to the flow equations, the so-called artificial dissipation [116]. To illustrate this procedure, consider the one-dimensional compressible Navier-Stokes equations (2.1)–(2.3) in semi-discrete form:

$$\frac{\partial}{\partial t}\vec{w}_j + L\vec{w}_j = 0, \quad (2.14)$$

where

$$\vec{w}_j = \frac{1}{|V_j|} \int_{V_j} \vec{W} dV, \quad \vec{W} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} \quad (2.15)$$

\vec{w}_j denotes the averaged conservative variables at the dual cell V_j . The spatial discretisation operator L can be split into:

$$L = L_C + L_D + L_{AD} \quad (2.16)$$

where the indices stand for "convection", "diffusion" and "artificial dissipation". If the spatial discretisation is performed by an upwind scheme, $L_{AD} \equiv 0$. In the case of a central scheme, one has to choose between scalar dissipation and matrix dissipation. The first one is numerically more stable, while the latter results in more accurate results.

Scalar dissipation

In the scalar dissipation scheme (see [59]) L_{AD} is given by:

$$L_{AD} \vec{w}_j = -(D^2 - D^4) \vec{w}_j, \quad (2.17)$$

where

$$D^2 \vec{w}_j = \nabla [(\lambda_{j+\frac{1}{2}} \epsilon_{j+\frac{1}{2}}^{(2)}) \Delta] \vec{w}_j \quad (2.18)$$

$$D^4 \vec{w}_j = \nabla [(\lambda_{j+\frac{1}{2}} \epsilon_{j+\frac{1}{2}}^{(4)}) \Delta \nabla \Delta] \vec{w}_j \quad (2.19)$$

Here ∇ and Δ denote the forward- and backward differencing operator, respectively. $\lambda_{j+\frac{1}{2}}$ is given by

$$\lambda_{j+\frac{1}{2}} = \frac{1}{2} [\lambda_j + \lambda_{j+1}], \quad (2.20)$$

where $\lambda_j = u_j + a$ with speed of sound a . The coefficients $\epsilon_{j+\frac{1}{2}}^{(2)}$ and $\epsilon_{j+\frac{1}{2}}^{(4)}$ contain a pressure-based sensor for large gradients:

$$\epsilon_{j+\frac{1}{2}}^{(2)} = \kappa^{(2)} \max(\nu_{j-1}, \nu_j, \nu_{j+1}, \nu_{j+2}), \quad (2.21)$$

where

$$\nu_j = \left| \frac{p_{j-1} - 2p_j + p_{j+1}}{p_{j-1} + 2p_j + p_{j+1}} \right| \quad (2.22)$$

$$\epsilon_{j+\frac{1}{2}}^{(4)} = \max \left[0, \kappa^{(4)} - \epsilon_{j+\frac{1}{2}}^{(2)} \right] \quad (2.23)$$

$\kappa^{(2)}$ and $\kappa^{(4)}$ are constants, which must be suitably chosen. Standard values are $\kappa^{(2)} = \frac{1}{2}$ and $\kappa^{(4)} = \frac{1}{64}$.

The sensor ν in Eq. (2.22) works as follows: In regions of smooth flow the pressure gradient is near zero, making ν small. This results in $\epsilon_{j+\frac{1}{2}}^{(4)}$ being much larger than

$\epsilon_{j+\frac{1}{2}}^{(2)}$ (unless $\kappa^{(4)}$ is chosen very small, which is not advisory). Overall the linear fourth order dissipation term D^4 dominates the operator L_{AD} , which results in a damping of high frequencies that are not damped by the original unmodified central scheme. This behaviour of L_{AD} is helpful for achieving a steady state solution. Contrarily ν becomes large in regions of shocks, resulting in a dominant position of D^2 . This is necessary to introduce an entropy condition to reduce overshoots near discontinuities and to choose the correct shock relationships.

Matrix dissipation

When using matrix dissipation, the scalar values λ_i in Eq. (2.20) are replaced by the matrix

$$A = \text{diag}(|\lambda_i^{(1)}|, |\lambda_i^{(2)}|, |\lambda_i^{(3)}|), \quad \lambda_i^{(1)} = u_i - a, \quad \lambda_i^{(2)} = u_i, \quad \lambda_i^{(3)} = u_i + a, \quad (2.24)$$

The scalar $\lambda_{j+\frac{1}{2}}$ in Eqs. (2.18) and (2.19) hereby also becomes a matrix.

2.2.4 Testcase in space and with varying dissipation type

In order to compare scalar and matrix dissipation when applying the central scheme, a numerical testcase has been performed. We consider the decay over time of a two-dimensional vortex resting in the center of the computational domain. At the same time these simulations can be regarded as preliminary investigations for the simulations shown in Chapter 7 of this thesis.

Derivation of the finite vortex

First we require an initial analytical vortex that is only defined on a finite domain is required. As all analytical vortices found in literature [32] spread infinitely wide in space, first the derivation of such a finite vortex is presented.

We start with the **potential vortex**, which is one of the simplest structured ones. Its tangential velocity v_ϕ is defined by:

$$v_\phi = \frac{\Gamma_0}{2 \cdot \pi \cdot r} \quad (2.25)$$

where Γ_0 describes the constant circulation of the vortex and r is its radius.

Unfortunately, the potential vortex is highly unphysical because of two reasons: Firstly, v_ϕ becomes unbounded as r tends to zero. Secondly, the vortex ranges infinitely wide, which can - due to dissipation - not be expected in reality.

One idea to overcome the first problem is to split the vortex into two zones, which is done by introducing a core radius r_c . While for $r > r_c$ the formular of the potential vortex remains unchanged, the fluid is supposed to rotate as a solid body if $r \leq r_c$. The result is the so-called **Rankine vortex**:

$$\begin{aligned} v_\phi &= \frac{\Gamma_0 \cdot r}{2 \cdot \pi \cdot r_c^2} \quad \text{if } r < r_c \\ v_\phi &= \frac{\Gamma_0}{2 \cdot \pi \cdot r} \quad \text{if } r_c \leq r \end{aligned} \quad (2.26)$$

The **Lamb-Oseen vortex** [30] goes one step further. Not only the first unphysical behaviour of the potential vortex is corrected, but also the second one is addressed. This is achieved by multiplying the formular of the potential vortex with a function β , which both bounds the values of v_ϕ in the center of the vortex and at the same time damps the tangential velocity, if the radius becomes large. Choosing $\beta = (1 - \exp(-\frac{r^2}{r_c^2}))$ turns out to be appropriate for these purposes. The result looks therefore as follows:

$$v_\phi = \frac{\Gamma_0}{2 \cdot \Pi \cdot r} \cdot \beta \quad (2.27)$$

Despite all efforts, the Lamb-Oseen vortex still spreads infinitely wide in space. In order to overcome this remaining obstacle, we introduce a new vortex model, which uses the ideas of all three models shown above. As this vortex only covers a finite area, we call it the **finite vortex** from this point on. It consists of four zones, which are defined by choosing a core radius r_c , a medium radius r_m and an outer radius r_o . While the finite vortex consists of a Rankine vortex in the two inner zones, the principle of the Lamb-Oseen vortex is applied in zone three. As the original function β damps the values of v_ϕ too slowly for increasing radius, we replace it by a function β^* , that meets our demands better. In zone four the values of v_ϕ are prescribed as zero, which finally results in a vortex of finite size. The complete finite vortex looks herewith:

Definition 2.2.1 (Finite vortex)

Choose core radius r_c , middle radius r_m , outer radius r_o , circulation Γ_0 and decay constant δ and let $\beta = e^{-\frac{(r-r_m)^2}{\delta}}$. Then the corresponding finite vortex is defined by:

$$\begin{aligned} \text{zone 1 : } v_\phi &= \frac{\Gamma_0 \cdot r}{2 \cdot \Pi \cdot r_c^2} & \text{if } r < r_c \\ \text{zone 2 : } v_\phi &= \frac{\Gamma_0}{2 \cdot \Pi \cdot r} & \text{if } r_c \leq r < r_m \\ \text{zone 3 : } v_\phi &= \frac{\Gamma_0}{2 \cdot \Pi \cdot r} \cdot \beta & \text{if } r_m \leq r < r_o \\ \text{zone 4 : } v_\phi &= 0 & \text{if } r_o \leq r \end{aligned}$$

Performance of the testcase

Having derived the finite vortex, the simulation to compare scalar and matrix dissipation can be described as follows: First a single finite vortex is initialised in the center of a structured quadratic grid. The size of the computational domain equals $100m \times 100m$. We consider four different cartesian grids, which contain 51^2 , 101^2 , 201^2 and 301^2 grid points. The grids are adapted both along the x- and the z-axis around the origin. Figure 2.5 shows the grid containing 201^2 grid points.

A time step size of $\Delta t = 10^{-2}$ s is chosen. The finite vortex is defined by the properties $\Gamma_0 = 60$, $r_c = 2m$, $r_m = 4m$, $r_o = 10m$, $\delta = 10m^2$ and its center is located at $(x, z) = (0m, 0m)$. As v_∞ is prescribed as zero, the vortex rests in the center of the grid throughout the whole computation, which is performed until 10s of physical time have elapsed.

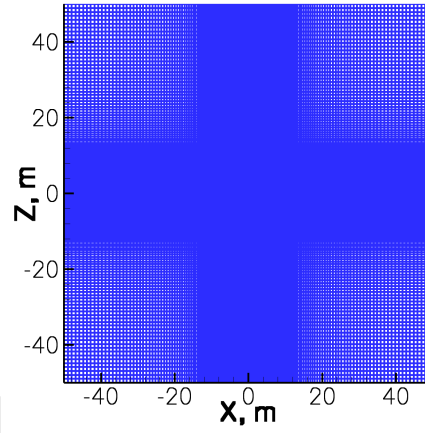
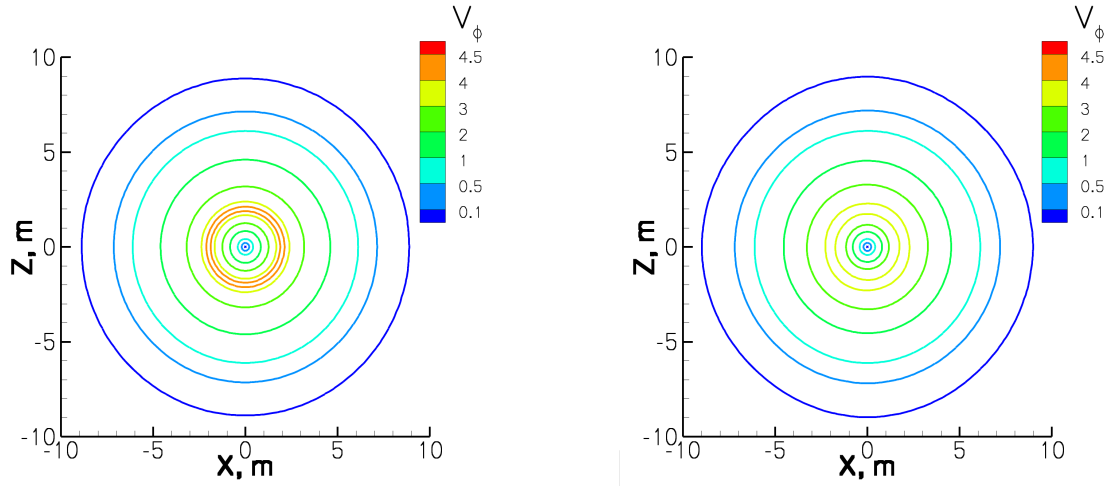
Figure 2.5: Structured adapted grid containing 201^2 points.Figure 2.6: Tangential velocity at $t=0s$ (left) and $t=10s$ (right) on the grid containing 201^2 points.

Figure 2.6 (left) shows v_ϕ at $t = 0s$ on a cutout of the computational domain. This is the result of the computation on the grid containing 201^2 points. One can clearly identify the core radius at $r = 2m$, where v_ϕ reaches its peak. Figure 2.6 (right) shows the same cutout at $t = 10s$. Size and magnitude of the vortex remain almost unchanged.

Figures 2.7-2.10 show v_ϕ at times $t = 0s, 1s, 3s$ and $10s$ on the various grids, where only the values at $(x, z) \in [0m, 10m] \times \{0m\}$ are displayed. While on the left hand sides the results obtained by applying scalar dissipation are shown, the right hand sides illustrate the decay of v_ϕ when using matrix dissipation.

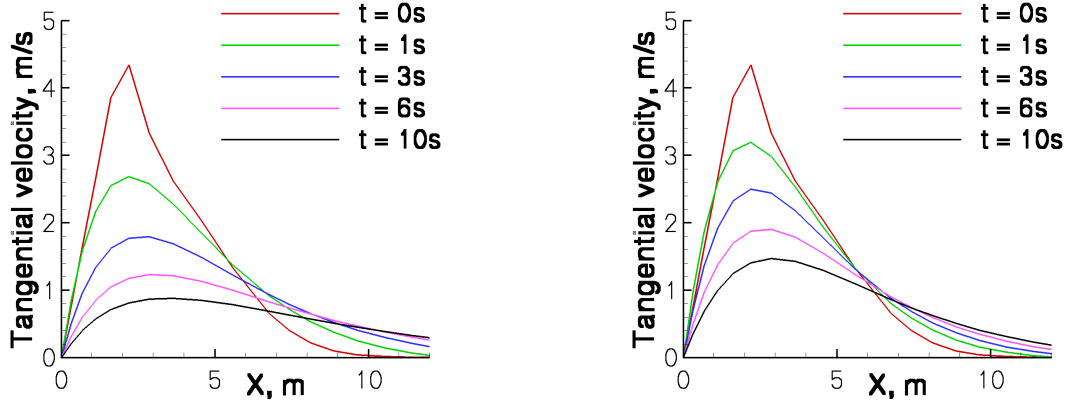


Figure 2.7: Decay of tangential velocity using scalar dissipation (left) and matrix dissipation (right) on the grid containing 51^2 points.

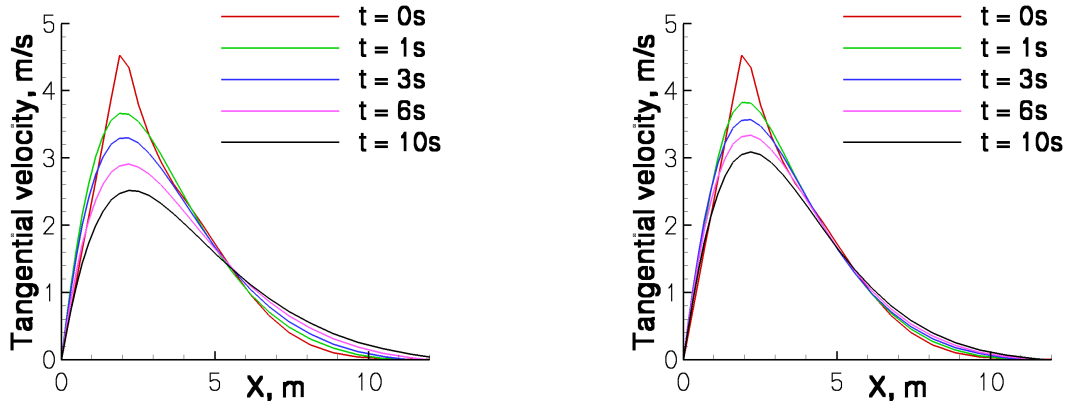


Figure 2.8: Decay of tangential velocity using scalar dissipation (left) and matrix dissipation (right) on the grid containing 101^2 points.

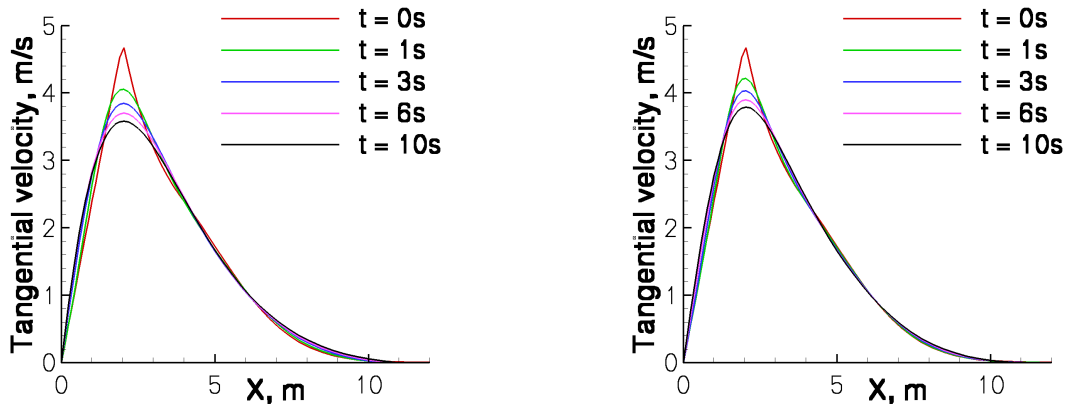


Figure 2.9: Decay of tangential velocity using scalar dissipation (left) and matrix dissipation (right) on the grid containing 201^2 points.

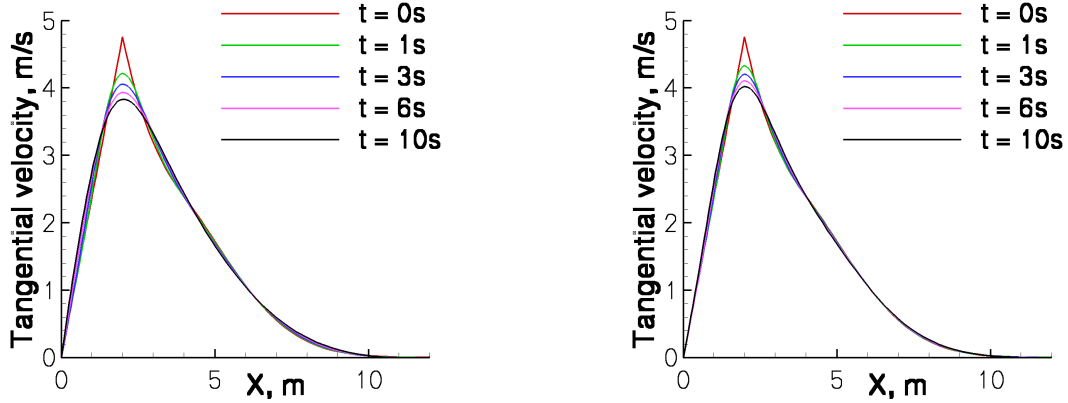


Figure 2.10: Decay of tangential velocity using scalar dissipation (left) and matrix dissipation (right) on the grid containing 301^2 points.

The following two conclusions can be made:

- As expected the vortex sustains the longer the finer the grid is chosen. While the two coarsest grids result in rapid vortex dissipation, the grid containing 201^2 points seems to be sufficiently fine.
- Matrix dissipation is superior to scalar dissipation. While the differences are not as obvious as the ones due to the choice of different grids, they are still clearly visible.

To sum it up, the result obtained by matrix dissipation on the grid containing 201^2 grid points seems to be the most convincing one regarding the trade-off of an adequate vortex preservation and numerical costs.

2.3 Time discretisation

Having successfully completed the spatial discretisation within one physical time step, one is left with a system of coupled ordinary differential equations of the form

$$\frac{\partial}{\partial t} \vec{w}_i + \vec{R}(\vec{w}_i) = 0 \quad (2.28)$$

The residual $\vec{R}(\vec{w}_i)$ contains all terms arising due to the spatial discretisation and \vec{w}_i is given in Eq. (2.15), where the extension of the definition to three dimensions is canonically.

In the case of steady state problems $\frac{\partial}{\partial t} \vec{w}_i = 0$ and Eq. (2.28) becomes:

$$\vec{R}(\vec{w}_i) = 0 \quad (2.29)$$

This problem is then solved by considering the corresponding time-dependent problem with fictitious pseudo time t^* and seek its steady-state solution:

$$\frac{\partial}{\partial t^*} \vec{w}_i + \vec{R}(\vec{w}_i) = 0 \quad (2.30)$$

In the case of time-accurate computations, one can choose between global and dual time stepping.

2.3.1 Dual time stepping

When using dual time stepping a backward difference formula (BDF) is applied to Eq. (2.28). In the DLR TAU-code this BDF can be chosen of order one, two or three. The second order BDF is given by:

$$\frac{3}{2\Delta t}\vec{w}_i^{n+1} - \frac{4}{2\Delta t}\vec{w}_i^n + \frac{1}{2\Delta t}\vec{w}_i^{n-1} + \vec{R}(\vec{w}_i^{n+1}) = 0 \quad (2.31)$$

Here \vec{w}_i^ν denotes the solution at time t_ν on dual cell V_i . $n+1$ denotes the current time level and n and $n-1$ the two previous levels. For each time step the nonlinear steady-state problem from Eq. (2.31) is solved by using the following dual-time stepping (DTS) scheme for pseudo time t^* :

$$\begin{aligned} \frac{\partial}{\partial t^*}\vec{w}_i^{n+1} + \vec{R}^{DTS}(\vec{w}_i^{n+1}) &= 0, \\ \vec{R}^{DTS}(\vec{w}_i^\nu) &= \vec{R}(\vec{w}_i^\nu) + \frac{3}{2\Delta t}\vec{w}_i^\nu - \frac{4}{2\Delta t}\vec{w}_i^n + \frac{1}{2\Delta t}\vec{w}_i^{n-1} \end{aligned} \quad (2.32)$$

One possibility of solving Eq. (2.30) or Eq. (2.32) lies in applying the following k-step low storage Runge-Kutta scheme [57, 59]:

$$\begin{aligned} \vec{w}_i^{(0)} &= \vec{w}_i^n \\ \vec{w}_i^{(1)} &= \vec{w}_i^{(0)} - \alpha_1 \Delta t \vec{R}(\vec{w}_i^{(0)}) \\ &\vdots \\ \vec{w}_i^{(k)} &= \vec{w}_i^{(0)} - \alpha_k \Delta t \vec{R}(\vec{w}_i^{(k-1)}) \\ \vec{w}_i^{n+1} &= \vec{w}_i^{(k)} \end{aligned} \quad (2.33)$$

where $\alpha_1, \dots, \alpha_k$ are the Runge-Kutta coefficients.

Alternatively a lower-upper symmetric Gauss-Seidel (LUSGS, [28, 60]) scheme can be used to solve Eq. (2.30) or Eq. (2.32). By means of the generalized trapezoid scheme one obtains the following A-stable scheme, which is first order accurate for $\beta = 1$ and second order accurate for $\beta = 0.5$:

$$\frac{\vec{w}_i^{n+1} - \vec{w}_i^n}{\Delta t} = -\beta \vec{R}(\vec{w}_i^{n+1}) - (1 - \beta) \vec{R}(\vec{w}_i^n) \quad (2.34)$$

Equation (2.34) is then linearised:

$$\begin{aligned} \vec{R}(\vec{w}_i^{n+1}) &= \vec{R}(\vec{w}_i^n) + \frac{\partial \vec{R}(\vec{w}_i^n)}{\partial t} \Delta t + O(\Delta t^2) \\ &= \vec{R}(\vec{w}_i^n) + \sum_{j \in N(i)} \frac{\partial \vec{R}(\vec{w}_i^n)}{\partial \vec{w}_j} \frac{\partial \vec{w}_j}{\partial t} \Delta t + O(\Delta t^2) \end{aligned} \quad (2.35)$$

where $N(i)$ is the set of grid points in the stencil of $\vec{R}(\vec{w}_i^n)$. Furthermore, the relation

$$\frac{\partial \vec{w}_i}{\partial t} = \frac{\vec{w}_i^{n+1} - \vec{w}_i^n}{\Delta t} + O(\Delta t^2) \quad (2.36)$$

is used, which altogether results in the following linear system:

$$\sum_{j \in N(i)} \left(\frac{1}{\Delta t} \delta_{ij} + \beta \frac{\partial \vec{R}(\vec{w}_i^n)}{\partial \vec{w}_j} \right) \Delta \vec{w}_j^n = -\vec{R}(\vec{w}_i^n) \quad (2.37)$$

where $\Delta \vec{w}_j^n = \vec{w}_j^{n+1} - \vec{w}_j^n$.

Instead of solving Eq. (2.37) directly, the aim of the LUSGS scheme is to provide an explicit formula for $\Delta \vec{w}_j^n$. For this purpose the matrix $A_{ij} = \frac{1}{\Delta t} \delta_{ij} + \beta \frac{\partial \vec{R}(\vec{w}_i^n)}{\partial \vec{w}_j}$ is approximately factorised by:

$$A' = (L + D)D^{-1}(U + D) \approx (L + D)D^{-1}(U + D) - LD^{-1}U = A \quad (2.38)$$

where U,D,L are the block lower triangular, diagonal and upper triangular parts of matrix A, respectively. This procedure allows to solve the modified Eq. (2.37) by two Gauss-Seidel sweeps [72], which is equivalent to two triangular systems solves, one lower and one upper. The advantage of the LUSGS scheme lies in the fact that the underlying system matrix needs not to be stored explicitly and hereby reducing the memory requirements to that of an explicit scheme.

Both in the Runge-Kutta scheme and in the LUSGS scheme the pseudo time step Δt is determined by the CFL-number.

2.3.2 Global time stepping

In the case of global time stepping the original system (2.28) is solved by applying the just described Runge-Kutta or LUSGS scheme. The global time step Δt is then chosen as the minimum time step of all dual cells. When applying the Runge-Kutta scheme in combination with global time stepping, the number of Runge-Kutta steps must be 1 or 2, because a higher number of steps would violate the time accuracy. Also the multigrid cycle must be set to "sg" in the DLR TAU-code because multigrid is not time-accurate in TAU.

2.3.3 Testcase in time

As in global time stepping the chosen minimum time step of all dual cells can become extremely small, the use of global time stepping is in general to expensive. Therefore only dual time stepping is used in this testcase and in all computations within in this thesis.

To investigate the effect of different time step sizes, the same testcase as in Subsection 2.2.4 has been performed, where all computations are realised on the grid with 201^2 grid points and matrix dissipation activated. While in Subsection 2.2.4 a fixed time step size $\Delta t = 10^{-2}s$ was used, now additionally the values $\Delta t = 10^{-1}s, 10^{-3}s$ and $10^{-4}s$ are tested.

Figures 2.11-2.12 show v_ϕ at times $t = 0s, 1s, 3s$ and $10s$ using the various time step sizes, where again only the values at $(x, z) \in [0m, 10m] \times \{0m\}$ are shown. As the computation using a time step size of $\Delta t = 10^{-1}s$ crashed after approximately 5s of

physical time, Fig. 2.11 (left) does not contain a result at $t = 10s$. Besides that the simulation shown in Fig. 2.11 (left) turned out to be unstable, also the dissipation is much larger compared to the other results. While the outcome shown in Fig. 2.11 (right), which is based on $\Delta t = 10^{-2}s$, looks satisfactory, the use of $\Delta t = 10^{-3}s$ even minimises the dissipation slightly more. Comparing the results utilising $\Delta t = 10^{-3}s$ and $10^{-4}s$ no improvement regarding the vortex preservation can be noticed. To sum it up, the result obtained by using a time step size of $\Delta t = 10^{-2}s$ is advisable with regard to both minimising vortex dissipation and numerical costs.

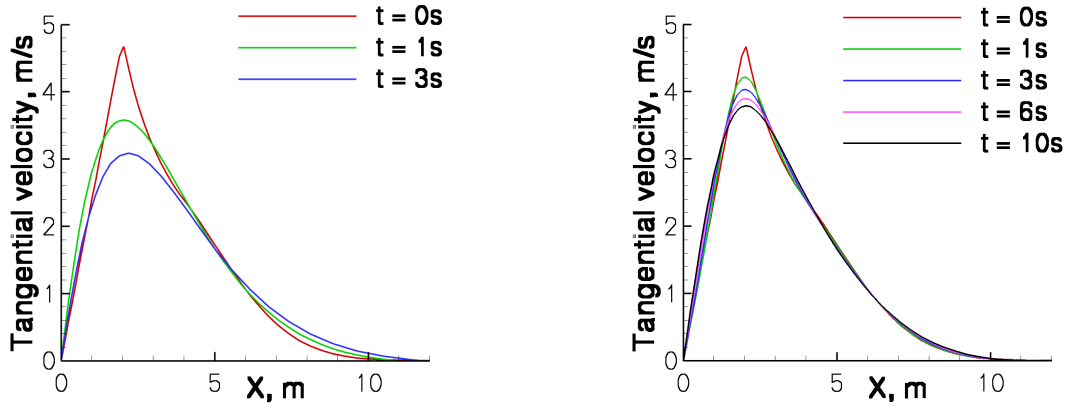


Figure 2.11: Decay of tangential velocity using a time step size of $\Delta t = 10^{-1}s$ (left) and $10^{-2}s$ (right).

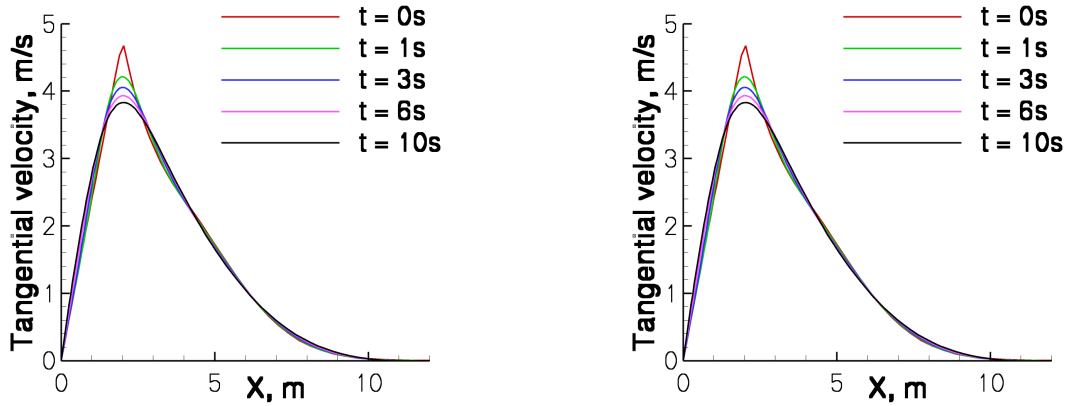


Figure 2.12: Decay of tangential velocity using a time step size of $\Delta t = 10^{-3}s$ (left) and $10^{-4}s$ (right).

2.4 Preconditioning methods

The aim of preconditioning methods (see e.g. [11, 121, 65]) is both to achieve convergence acceleration and to minimise the numerical dissipation.

2.4.1 Motivation: The idea of preconditioning methods

To get a basic idea of the principle of these methods, we first consider the following problem:

Definition 2.4.1 ($n \times n$ system of linear equations)

For $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ find $x \in \mathbb{R}^n$ such that:

$$Ax = b$$

If A is symmetric and positive definite, the condition $\text{cond}(A)$ of matrix A is given by:

$$\text{cond}(A) := \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

where λ_{\max} and λ_{\min} are the largest and respectively smallest eigenvalues of A . If $\text{cond}(A)$ becomes large, problem (2.4.1) is *ill posed*, meaning that small variations in A can result in strongly varying solutions of x . As a result problem (2.4.1) cannot be easily solved numerically because of the naturally involved rounding errors. To avoid this problem, we instead consider the following equivalent problem to (2.4.1):

Definition 2.4.2 (Preconditioned $n \times n$ system of linear equations)

For $M, A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ find $x \in \mathbb{R}^n$ such that:

$$MAx = Mb$$

The goal is then to find a matrix M , such that $\text{cond}(MA)$ becomes small. The perfect choice is obviously $M = A^{-1}$, resulting in $\text{cond}(MA) = 1$. Unfortunately, the exact determination of A^{-1} is too costly for most relevant problems. The aim of preconditioning methods lies in choosing matrices M that are good approximations of A^{-1} .

2.4.2 Low Mach number preconditioning

To illustrate the principle of low Mach number preconditioning, consider the three-dimensional compressible Euler equations in primitive variables $w := (p, u_1, u_2, u_3, T)$, which can be written as:

$$\frac{\partial w}{\partial t} + \sum_{i=1}^3 \frac{\partial f^{(i)}(w)}{\partial x_i} = 0 \quad (2.39)$$

As the exact definitions of functions $f^{(i)}$ are space-consuming and not strictly required in this context, they are omitted here. For low Mach numbers system (2.39) is stiff (i.e. its condition is large), which can be shown by a spectral analysis. As a result the admissible time step size to compute an approximate steady state solution of Eqs. (2.39) is extremely small, thus making numerical simulations virtually impossible.

Therefore the concept of low Mach number preconditioning is introduced, which is based on the following idea: We seek a steady state solution w of Eq. (2.39):

$$\sum_{i=1}^3 \frac{\partial f^{(i)}(w)}{\partial x_i} = 0 \quad (2.40)$$

Hence, we can multiply the second summand of (2.39) with a matrix valued preconditioner $P(w) \in \mathbb{R}^{5 \times 5}$, which aims to reduce the stiffness. Instead of Eqs. (2.39) we then must solve the following equations:

$$\frac{\partial w}{\partial t} + P(w) \sum_{i=1}^3 \frac{\partial f^{(i)}(w)}{\partial x_i} = 0 \quad (2.41)$$

While several preconditioners can be found in literature, we want to focus on $P(w) := P(\alpha, \beta, \delta, w) \in \mathbb{R}^{5 \times 5}$, which has been introduced in [91] and is a generalization of a large range of preconditioners. α, β , and δ are parameters that have yet to be specified. $P(w)$ is given by:

$$P(w) := \begin{pmatrix} m^2 g & 0 & 0 & 0 & -m^2 \frac{\gamma p}{T} \delta \\ -\frac{\alpha u_1 g}{\rho a^2} & 1 & 0 & 0 & \frac{\alpha u_1}{\rho a^2} \frac{\gamma p}{T} \delta \\ -\frac{\alpha u_2 g}{\rho a^2} & 0 & 1 & 0 & \frac{\alpha u_2}{\rho a^2} \frac{\gamma p}{T} \delta \\ -\frac{\alpha u_3 g}{\rho a^2} & 0 & 0 & 1 & \frac{\alpha u_3}{\rho a^2} \frac{\gamma p}{T} \delta \\ \frac{(\gamma-1)}{\gamma \rho} (m^2 g - 1) & 0 & 0 & 0 & 1 - (\gamma - 1) m^2 \delta \end{pmatrix}, \quad (2.42)$$

where

$$m^2 := \frac{\beta}{a^2} \quad \text{and} \quad g := 1 + (\gamma - 1) \delta$$

and γ denotes the ratio of specific heats and a is the speed of sound. As a necessary condition that $P(w)$ remains nonsingular we prescribe $\beta \neq 0$. As the formulation of a sufficient condition is impossible for most problems, $P(w)$ becoming singular cannot be prevented in general. Therefore one has to use experience-based ad-hoc methods.

One can show that the eigenvalues of $P(w) \frac{\partial f^{(i)}(w)}{\partial x_i}$ ($i=1,2,3$) are given by:

$$\Lambda_0^{(i)} = \text{diag} \left(u_i, u_i, u_i, \mu_+^{(i)}, \mu_-^{(i)} \right) \quad (2.43)$$

where

$$\mu_{+,-}^{(i)}(\alpha, \beta) := \frac{1}{2} \left((1 - \alpha + m^2) u_i \pm \sqrt{(1 - \alpha + m^2)^2 u_i^2 + 4\beta \left(1 - \frac{u_i^2}{a^2} \right)} \right)$$

Ideal choices of α and β would result in a clustering of the eigenvalues, i.e.:

$$|\mu_+^{(i)}(\alpha, \beta)| \approx |u_i| \approx |\mu_-^{(i)}(\alpha, \beta)| \quad (i = 1, 2, 3) \quad (2.44)$$

To satisfy (2.44) the following optimisation problem must be solved:

$$\frac{1}{2} \sum_{i=1}^3 \left[\left(\mu_+^{(i)}(\alpha, \beta) - u_i \right)^2 + \left(\mu_-^{(i)}(\alpha, \beta) + u_i \right)^2 \right] = \min_{\alpha, \beta} ! \quad (2.45)$$

Note that the parameter δ does not occur in (2.44) and (2.45), because it is not part of the eigenvalues but of the corresponding eigenvectors. As we have six conditions for only two unknowns and because the eigenvalues depend on the velocities u_i , which are functions of space and time, it is unrealistic to find perfect values of α and β in reasonable time. Therefore in practice only heuristic parameter approximations are used, such that the following three conditions are met:

- a) For low Mach numbers the parameters α, β and δ should be chosen such that (2.44) is fulfilled as good as possible.
- b) For transonic and supersonic flows the parameters α, β and δ should be chosen such that the preconditioner $P(w)$ is mostly switched off.
- c) To ensure that the preconditioner $P(w)$ is nonsingular and to avoid numerical instabilities, m^2 should be significantly larger than zero, i.e.

$$m^2 = \frac{\beta}{a^2} > \varepsilon > 0, \quad \varepsilon > 0$$

At the end of Subsection 2.4 the parameter choices in the DLR TAU-Code are given. As the effect of α in general is hard to predict and also not well understood, $\alpha = 0$ is prescribed. β is determined by the following formula, which was introduced in [91]:

$$\beta(u) = \min \left\{ \max \left\{ \|u\|_2^2, K \|u_\infty\|_2^2 \right\}, a^2 \right\} \quad (2.46)$$

where $\|u_\infty\|_2$ is the speed of the inflowing fluid. In the supersonic case Eq. (2.46) results in $\beta = a^2$, thus precondition is switched off. The parameter K is a cut-off value that has to be set a-priori such that the numerical scheme does not become unstable. For most flow problems, $K \in [1, 4]$ should be chosen.

Two alternatives for δ are available:

- a) $\delta = 1$. While this choice works well for most flows, numerical instabilities can occur for transonic and supersonic cases.

- b) $\delta := \delta(Ma)$ where

$$\delta(Ma) := \begin{cases} 0, & Ma^2 \geq 1, \\ 1, & Ma^2 < 1. \end{cases}$$

and Ma is the Mach number. While this definition of δ is identical to a) in the case of subsonic flow, it results in a numerically more stable algorithm for transonic and supersonic problems.

2.4.3 Testcase with regard to low Mach number preconditioning

To illustrate the effect of low Mach number preconditioning on the dissipation, the testcase shown in Subsections 2.2.4 and 2.3.3 is performed again.

While all simulations in Subsections 2.2.4 and 2.3.3 were realised with low Mach number preconditioning activated, now additionally a simulation with low Mach number preconditioning deactivated is presented. Based on the best-practice results obtained in Subsections 2.2.4 and 2.3.3, both shown computations are based on a time step size of $\Delta t = 10^{-2}$ s on the grid with 201^2 grid points and matrix dissipation activated.

Figures 2.13 (left) and (right) show that the decay of the tangential velocity is much faster, if preconditioning is deactivated. Therefore the use of low Mach number preconditioning is clearly advisable.

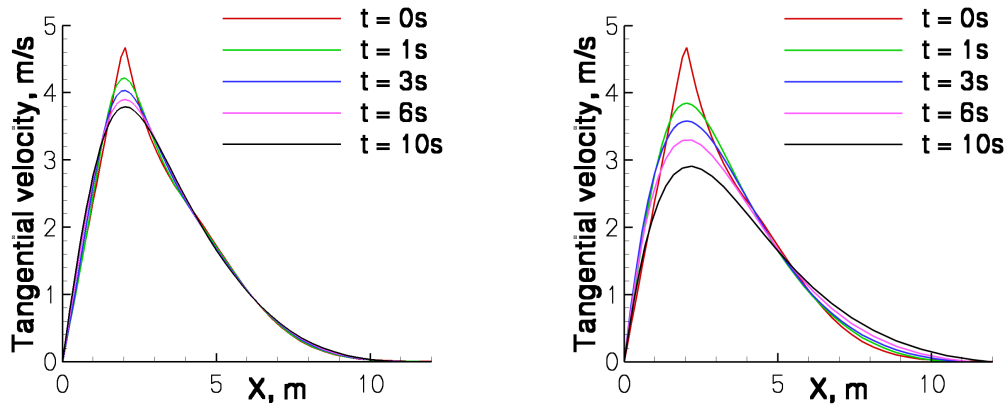


Figure 2.13: Decay of tangential velocity with low Mach number preconditioning activated (left) and deactivated (right).

2.5 Multigrid methods

In multigrid methods the flow solutions are obtained by using a hierarchy of discretisations. The main idea is to accelerate the convergence by correcting the solution globally by solving a problem on a set of coarse grids. This is achieved by applying restriction operators (allowing the transfer of solutions and residuals from a fine grid to the next coarser grid), prolongation operators (transfer from a coarse grid to the next finer grid) and smoothing operators (which perform a smoothing of the residuals and corrections to improve convergence).

2.5.1 Motivation: Correction scheme for linear problems

As a motivation for the multigrid scheme for nonlinear problems used in the DLR TAU-code (the so-called full approximation scheme), the well-known multigrid scheme for linear problems referred to as correction scheme (see [13, 63]) is introduced first. Consider the linear problem

$$L w = f \quad (2.47)$$

We introduce the following notation: Solving (2.47) on grid level k is written as

$$L^k w^k = f^k \quad (2.48)$$

We seek w^1 s.t. on the finest grid level 1

$$L^1 w^1 = f^1 \quad (2.49)$$

Next restriction and prolongation operators have to be specified. The restriction operator I_k^{k+1} restricts a solution on level k to the next coarser level $k+1$. The prolongation operator I_{k+1}^k prolongates a solution from level $k+1$ to the next finer level k .

On each coarse grid level $k > 1$ and given an approximation \tilde{w}^{k-1} from the next finer level, we seek the correction c^k s.t.

$$L^k(I_{k-1}^k \tilde{w}^{k-1} + c^k) = f^k \Leftrightarrow L^k c^k = r^k \equiv I_{k-1}^k(f^{k-1} - L^{k-1} \tilde{w}^{k-1}) \quad (2.50)$$

where the approximation \tilde{w}^{k-1} is from the next finer level and the linearity of L is used. After having solved Eq. (2.50), c^k is used as correction to obtain a new fine-grid solution

$$w^{k-1} = \tilde{w}^{k-1} + I_k^{k-1} c^k \quad (2.51)$$

When performing N relaxation smoothing steps on level k , starting with w_0^k and with right hand side term being f^k , we write $J_N(w_0^k, f^k)$. Additionally, $w_N^k = J_N(w_0^k, f^k)$ indicates the smoothed solution.

The correction scheme then reads

INPUT: w_0^k

OUTPUT: $MGC(k, w_0^k, f^k)$ is defined by the recursion

IF $k = I_C$ **THEN**

- (1) Determine (exact) solution c^k of $L^k c^k = f^k$
- (2) **RETURN** c^k

ELSE

- (1) Initial guess : $w_0^k = 0$
- (2) Perform N_1 relaxation/smoothing steps : $w_a^k = J_{N_1}(w_0^k, f^k)$
- (3) DO Recursion : Compute correction on next coarser grid
 $c^{k+1} = MGC(k+1, w_0^{k+1}, I_k^{k+1}(f^k - L^k w_a^k)).$
- (4) Correction : $w_b^k = w_a^k + I_{k+1}^k c^{k+1}$
- (5) Perform N_2 relaxation/smoothing steps : $w_c^k = J_{N_2}(w_b^k, f^k)$
- (6) **RETURN** w_c^k

2.5.2 The full approximation scheme

The full approximation scheme (see [13, 63]) does not require the operator L in Eq. (2.47) to be linear. It is therefore directly applicable to Eqs. (2.1)–(2.3) and can be described as follows: On the finest grid level 1 we seek w^1 s.t.

$$L^1 w^1 = f^1 \quad (2.52)$$

On each coarse grid level $k > 1$ and given an approximation \tilde{w}^{k-1} from the finer level, we seek $\hat{w}^k = I_{k-1}^k \tilde{w}^{k-1} + c^k$ s.t.

$$L^k \hat{w}^k = \hat{f}^k \equiv L^k(I_{k-1}^k \tilde{w}^{k-1}) + I_{k-1}^k(f^{k-1} - L^{k-1} \tilde{w}^{k-1}) \quad (2.53)$$

where the approximation \tilde{w}^{k-1} is given from the finer level and L needs not to be linear. Having solved Eq. (2.53), the new fine-grid solution is obtained by

$$w^{k-1} = \tilde{w}^{k-1} + I_k^{k-1}(\hat{w}^k - I_{k-1}^k \tilde{w}^{k-1}) \quad (2.54)$$

Note, that the full approximation scheme reduces to the correction scheme, if L is a linear operator. To see this, we rearrange (2.53) as

$$\begin{aligned} L^k \hat{w}^k - L^k(I_{k-1}^k \tilde{w}^{k-1}) &= I_{k-1}^k(f^{k-1} - L^{k-1} \tilde{w}^{k-1}) \\ \Leftrightarrow L^k c^k &= I_{k-1}^k(f^{k-1} - L^{k-1} \tilde{w}^{k-1}), \quad c^k \equiv \hat{w}^k - I_{k-1}^k \tilde{w}^{k-1} \end{aligned}$$

given that L is linear. Moreover, the fine grid updated solution becomes

$$w^{k-1} = \tilde{w}^{k-1} + I_k^{k-1} c^k. \quad (2.55)$$

The recursion form of the full approximation scheme can be written as follows:

INPUT: w_0^k

OUTPUT: $MGC(k, w_0^k, f^k)$ is defined by the following recursion

IF $k = I_C$ **THEN**

(1) Determine exact solution \hat{w}^k of $L^k \hat{w}^k = f^k$

(2) **RETURN** \hat{w}^k

ELSE

(1) Initial guess : $w_0^k = 0$

(2) Perform N_1 relaxation/smoothing steps : $w_a^k = J_{N_1}(w_0^k, f^k)$

(3) DO Recursion : Compute solution \hat{w}^{k+1} on next coarser grid
 $\hat{w}^{k+1} = MGC(k+1, w_0^{k+1}, L^{k+1}(I_k^{k+1} w_a^k) + I_k^{k+1}(f^k - L^k w_a^k)).$

(4) Correction : $w_b^k = w_a^k + I_{k+1}^k(\hat{w}^{k+1} - I_k^{k+1} w_a^k)$

(5) Perform N_2 relaxation/smoothing steps : $w_c^k = J_{N_2}(w_b^k, f^k)$

(6) **RETURN** w_c^k

Note, that (2.53) can be rewritten as

$$L^k \hat{w}^k = f^k + \tau_{k-1}^k, \quad \tau_{k-1}^k \equiv L^k(I_{k-1}^k \tilde{w}^{k-1}) - I_{k-1}^k(L^{k-1} \tilde{w}^{k-1}) \quad (2.56)$$

with $f^k = I_{k-1}^k(f^{k-1})$.

Finally the relaxation/smoothing operation $J_N(w_0^k, f^k)$ and the solution operation on the coarsest grid level I_C have to be specified.

In principle, each iterative solver of problem (2.47) can be used as a smoother for a multigrid scheme, if it damps the high frequency components of the error on the corresponding grid level. In the DLR TAU-code, smoothing is performed by applying an explicit K -stage low storage Runge-Kutta scheme as iterative solver to the pseudo-instationary problem given by

$$\frac{\partial}{\partial t^*} w + Lw = f \quad (2.57)$$

with pseudo time t^* . Only one smoothing step before the recursion is performed, i.e. $N_1 = 1$, and moreover $N_2 = 0$ is used. Thus $J_1(w_0^k, f^k)$ is defined as follows:

Smoothing/ relaxation $J_1(w_0^k, f^k)$

INPUT: w_0^k

OUTPUT: $J_1(w_0^k, f^k)$

FOR $j = 1, \dots, K$ **DO** $w_j^k = w_0^k - \alpha_j \Delta t^* (L(w_{j-1}^k) - f^k)$

RETURN $J_1(w_0^k, f) = w_K^k$

On the coarsest grid level $k = I_C$, we define $MGC(I_C, w_0^{I_C}, f^{I_C})$ also by applying one step of an explicit K -stage low storage Runge-Kutta scheme, i.e.

Solution on coarsest grid level $MGC(I_C, w_0^{I_C}, f^{I_C})$

INPUT: $w_0^{I_C}$

OUTPUT: $MGC(I_C, w_0^{I_C}, f^{I_C})$

FOR $j = 1, \dots, K$ **DO** $w_j^{I_C} = w_0^{I_C} - \alpha_j \Delta t^* (L(w_{j-1}^{I_C}) - f^{I_C})$

RETURN $MGC(I_C, w_0^{I_C}, f^{I_C}) = w_K^{I_C}$

2.5.3 Testcase with regard to multigrid

While the use of different multigrid methods has no effect on the vortex dissipation, the same testcase already shown in Subsections 2.2.4, 2.3.3 and 2.4.3 is nevertheless performed at this point, where now the influence of varying multigrid methods on the convergence rate is examined.

The computations were all realised using a time step size of $\Delta t = 10^{-2}$ s on the grid with 201^2 grid points and matrix dissipation and low Mach number preconditioning activated. Three computations have been performed: While the green line in Fig. 2.14 shows the convergence rate of the computation using singlegrid, the blue and red line indicate the result when a multigrid method with two and alternatively three grid levels is activated.

For better comparison only 200 inner iterations have been performed in each time step within all three computations. Note additionally, that the results shown in Fig. 2.14 are only based on the first five time steps of each simulation and that the residual is renormalised to one at the beginning of each time step.

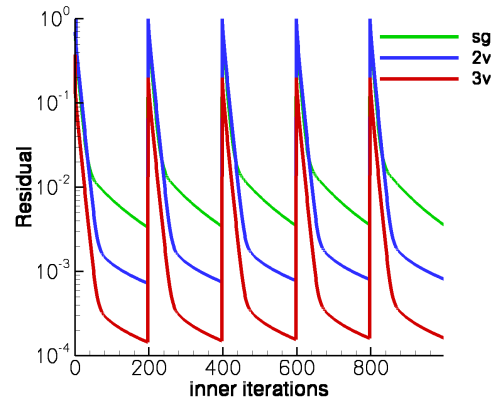


Figure 2.14: Convergence rates when using singlegrid (sg), multigrid with two grid levels (2v) and multigrid with three grid levels (3v).

Looking at Fig. 2.14 it can be stated that the use of multigrid clearly improves the convergence acceleration. The use of multigrid is therefore advisable in order to minimise the numerical costs.

Chapter 3

Turbulence modeling

Recall the compressible Navier-Stokes equations defined by Eqs. (2.1) – (2.3):

$$\begin{aligned}\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x_i}(\rho u_i) &= 0 \text{ in } \Omega \\ \frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) &= -\frac{\partial}{\partial x_i}p + \frac{\partial}{\partial x_j}t_{ji} \text{ in } \Omega \\ \frac{\partial}{\partial t}E + \frac{\partial}{\partial x_j}\left[\rho u_j\left(h + \frac{1}{2}u_i u_i\right)\right] &= \frac{\partial}{\partial x_j}(u_i t_{ij}) - \frac{\partial}{\partial x_j}q_j\end{aligned}$$

When the Reynolds number gets large, this system becomes turbulent, meaning that the solution is three-dimensional, time-dependent and includes a large range of timescales and lengthscales. As a Direct Numerical Simulation (DNS) to solve Eqs. (2.1) – (2.3) must take all occurring scales into account, realistic flow problems with high Reynolds numbers would exceed today's computer power resources.

One approach to overcome this limitation is to suitably average the original equations. In the case of compressible flow this averaging procedure is performed by applying so-called Reynolds averaging and Favre averaging.

3.1 Reynolds averaging and Favre averaging

Definition 3.1.1 (Reynolds averaging)

Let $f(x, t)$ be any instantaneous flow variable. f can then be decomposed into

$$f(x, t) = \overline{f(x)} + f'(x, t)$$

where the Reynolds-Averaged (or time-averaged) variable $\overline{f(x)}$ is given by:

$$\overline{f(x)} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} f(x, t) dt$$

and f' is the fluctuating part. This decomposition is also referred to as "Reynolds decomposition".

Reynolds averaging fulfills the following properties:

Lemma 3.1.2 (Properties of Reynolds averaging)

Let $f(x, t)$ and $g(x, t)$ be instantaneous flow variables and let $c, d \in \mathbb{R}$. Then the following identities hold:

$$\begin{aligned} \text{Linearity} & : \overline{c \cdot f + d \cdot g} = c \cdot \overline{f} + d \cdot \overline{g} \\ \text{Commutativity of differentiation w.r.t. } x & : \overline{\frac{\partial}{\partial x} f} = \frac{\partial}{\partial x} \overline{f} \\ \text{Commutativity of differentiation w.r.t. } t & : \overline{\frac{\partial}{\partial t} f} = \frac{\partial}{\partial t} \overline{f} \\ \text{Projection identity} & : \overline{\overline{f}} = \overline{f} \end{aligned}$$

Due to the projection identity, $\overline{f'(x, t)}$ equals zero.

When dealing with incompressible flow, only velocity and pressure need to be averaged, which can be realised by the just described Reynolds averaging. In the case of a compressible fluid, also density and temperature must be considered. Unfortunately Reynolds averaging Eqs. (2.1) – (2.3) would then lead to the creation of additional terms, which have no counterpart in the laminar equations.

To understand this problem in more detail, consider the Reynolds decomposition of the velocity $u_i(x, t)$ and density ρ :

$$u_i(x, t) = \overline{u_i(x)} + u'_i(x, t) \quad (3.1)$$

$$\rho(x, t) = \overline{\rho(x)} + \rho'(x, t) \quad (3.2)$$

Substituting (3.1) and (3.2) into continuity equation (2.1) results in:

$$\frac{\partial}{\partial t} (\overline{\rho} + \rho') + \frac{\partial}{\partial x_i} (\overline{\rho} \overline{u_i} + \rho' \overline{u_i} + \overline{\rho} u'_i + \rho' u'_i) = 0 \text{ in } \Omega \quad (3.3)$$

Reynolds averaging Eq. (3.3) yields the Reynolds averaged continuity equation for compressible flow:

$$\frac{\partial}{\partial t} \overline{\rho} + \frac{\partial}{\partial x_i} (\overline{\rho} \overline{u_i} + \overline{\rho' u'_i}) = 0 \text{ in } \Omega \quad (3.4)$$

Compared to the original Eq. (2.1), Eq. (3.4) now contains the additional term $\overline{\rho' u'_i}$, resulting in additional variables. This leads to the problem, that additional approximations would have to be found to model these variables.

In order to avoid this problem, an alternative averaging procedure is introduced, the so-called Favre averaging.

Definition 3.1.3 (Favre averaging)

Let $f(x, t)$ be any instantaneous flow variable. f can then be decomposed into

$$f(x, t) = \widetilde{f(x)} + f''(x, t)$$

where the Favre averaged (or mass averaged) variable $\widetilde{f(x)}$ is given by:

$$\widetilde{f(x)} = \frac{1}{\overline{\rho(x)}} \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \rho(x, t) f(x, t) dt$$

and $f''(x, t)$ is the corresponding fluctuating part. This decomposition is also referred to as "Favre decomposition".

Note that Favre averaging yields the same properties as the ones for Reynolds averaging described in lemma 3.1.2 except for the projection identity. As a consequence, also $\widetilde{f''(x, t)}$ does not vanish in general.

In order to realise, why Favre averaging is useful in the current context, consider the following identity:

$$\overline{\rho \widetilde{u}_i} = \overline{\rho u_i} = \overline{\rho} \overline{u_i} + \overline{\rho' u_i'} \quad (3.5)$$

Inserting Ey. (3.5) into Eq. (3.4) results in

$$\frac{\partial}{\partial t} \overline{\rho} + \frac{\partial}{\partial x_i} (\overline{\rho} \widetilde{u}_i) = 0 \text{ in } \Omega \quad (3.6)$$

which is clearly a simplification compared to Eq. (3.4).

3.2 The Favre averaged Navier-Stokes equations

As now all required mathematical techniques have been introduced, the compressible Navier-Stokes equations (2.1) – (2.3) can be averaged. First the following decompositions are performed:

$$\begin{aligned} u_i &= \widetilde{u}_i + u_i'' & \rho &= \overline{\rho} + \rho' & p &= \overline{p} + p' & h &= \widetilde{h} + h'' \\ e &= \widetilde{e} + e'' & T &= \widetilde{T} + T'' & q_j &= q_{L_j} + q_j'' \end{aligned} \quad (3.7)$$

where q_{L_j} is the laminar mean heat-flux vector. Inserting the decompositions from (3.7) into Eqs. (2.1), (2.2), (2.3) and (2.4) and performing the Favre averaging operations results in the following *Favre averaged mean conservation equations* and the equation of state:

Definition 3.2.1 (Favre averaged mean Navier-Stokes equations)

$$\frac{\partial}{\partial t} \overline{\rho} + \frac{\partial}{\partial x_i} (\overline{\rho \widetilde{u}_i}) = 0 \text{ in } \Omega \quad (3.8)$$

$$\frac{\partial}{\partial t} (\overline{\rho \widetilde{u}_i}) + \frac{\partial}{\partial x_j} (\overline{\rho \widetilde{u}_j \widetilde{u}_i}) = - \frac{\partial}{\partial x_i} \overline{P} + \frac{\partial}{\partial x_j} [\overline{t_{ji}} + \overline{\rho \tau_{ij}}] \text{ in } \Omega \quad (3.9)$$

$$\frac{\partial}{\partial t} (\overline{\rho \widetilde{E}}) + \frac{\partial}{\partial x_j} (\overline{\rho \widetilde{u}_j \widetilde{H}}) =$$

$$\frac{\partial}{\partial x_j} \left[-q_{L_j} - q_{T_j} + \overline{t_{ji}u_i''} - \overline{\rho u_j'' \frac{1}{2} u_i'' u_i''} \right] + \frac{\partial}{\partial x_j} [\tilde{u}_i (\overline{t_{ij}} + \overline{\rho \tau_{ij}})] \quad (3.10)$$

$$P = \overline{\rho} R \tilde{T} \quad (3.11)$$

Again, these equations are completed by Eq. (3.11). The turbulent mean heat-flux vector q_{T_j} is given by:

$$q_{T_j} = \overline{\rho u_j'' h''} \quad (3.12)$$

The Favre averaged Reynolds-stress tensor is defined by:

$$\overline{\rho \tau_{ij}} = -\overline{\rho u_i'' u_j''} \quad (3.13)$$

\tilde{E} and \tilde{H} are the total energy and total enthalpy:

$$\tilde{E} = \tilde{e} + \frac{\tilde{u}_i \tilde{u}_i}{2} + k \quad (3.14)$$

$$\tilde{H} = \tilde{h} + \frac{\tilde{u}_i \tilde{u}_i}{2} + k \quad (3.15)$$

At this point it is important to note that Eqs. (3.8) - (3.10) and (3.11) are not closed, meaning that more variables than equations exist. In order to close this system of equations, additional approximations for the mean heat-flux vector q_{T_j} , the term $\overline{t_{ji}u_i''} - \overline{\rho u_j'' \frac{1}{2} u_i'' u_i''}$, the Favre averaged Reynolds-stress tensor $\overline{\rho \tau_{ij}}$ and the turbulent kinetic energy k must be found.

q_{T_j} can be modeled by a gradient-diffusion approximation, meaning that the variable is proportional to the mean temperature gradient:

$$q_{T_j} = -\frac{\mu_T c_p}{Pr_T} \frac{\partial}{\partial x_j} \tilde{T} = -\frac{\mu_T}{Pr_T} \frac{\partial}{\partial x_j} \tilde{h} \quad (3.16)$$

where the turbulent Prandtl number Pr_T is usually considered as constant. Depending on the type of flow, a value between 0.5 and 0.9 is chosen. μ_T denotes the eddy-viscosity.

$\overline{t_{ji}u_i''}$ and $\overline{\rho u_j'' \frac{1}{2} u_i'' u_i''}$ are the molecular diffusion and the turbulent transport, respectively. In most turbulence models, these terms are simply ignored. Alternatively, they can be modeled by:

$$\overline{t_{ji}u_i''} - \overline{\rho u_j'' \frac{1}{2} u_i'' u_i''} = \left(\mu + \frac{\mu_T}{\sigma_k} \right) \frac{\partial}{\partial x_j} k \quad (3.17)$$

Here μ denotes the molecular viscosity. In almost all turbulence models $\overline{\rho \tau_{ij}}$ is modeled by the famous Boussinesq approximation for compressible flow:

$$\overline{\rho \tau_{ij}} = 2\mu_T \left(S_{ij} - \frac{1}{3} \frac{\partial}{\partial x_k} \tilde{u}_k \delta_{ij} \right) - \frac{2}{3} \overline{\rho} k \delta_{ij} \quad (3.18)$$

3.3 URANS models

By using the approximations from Section 3.2, the only elements missing to close Eqs. (3.8) - (3.10) and (3.11) are models to obtain μ_T and k . How these variables are obtained depends on the type of turbulence model used.

Before introducing them we want to note, that the commonly used abbreviation for the Favre averaged Navier-Stokes equations is denoted by "(U)RANS". The letter "U" stands hereby for "unsteady" and is used, if the time derivatives are included in the equations; in case of steady-state equations (i.e. the time derivatives are omitted) also the "U" is excluded. Note furthermore, that the letter "R" ("Reynolds") and not "F" ("Favre") is used in the literature even in the case of compressible flow.

3.3.1 The Spalart-Allmaras (SA(O)) model

In the Spalart-Allmaras Favre averaged Navier-Stokes (abbreviated by "SA(0)" where the sometimes used "0" stands for "original") model k is set to zero, such that only one transport equation for the eddy viscosity μ_T is required. The model was introduced by Spalart et al. [108] and calibrated and validated upon free shear flow, near-wall flow at high Reynolds numbers, near-wall flow at finite Reynolds numbers, laminar flow of shear layers and zero-pressure gradient boundary layer flow.

μ_T is obtained by:

$$\mu_T = \rho f_{v1} \tilde{\nu}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu} \quad (3.19)$$

A sketch of f_{v1} can be seen in Fig. 3.1 (left). ν is the kinematic viscosity and $\tilde{\nu}$ satisfies the following transport equation:

$$\frac{\partial}{\partial t}(\rho \tilde{\nu}) + \frac{\partial}{\partial x_i}(\rho u_i \tilde{\nu}) - \frac{\partial}{\partial x_k} \left[\frac{\rho(\nu + \tilde{\nu})}{\sigma} \frac{\partial}{\partial x_k} \tilde{\nu} \right] - \rho \frac{c_b2}{\sigma} \left(\frac{\partial}{\partial x_k} \tilde{\nu} \right)^2 = P - D \quad (3.20)$$

The production and destruction terms P and D are given by

$$P = c_{b1} \rho \tilde{S} \tilde{\nu}, \quad D = c_{w1} f_w \rho \left(\frac{\tilde{\nu}}{d} \right)^2 \quad (3.21)$$

with wall distance d and

$$\tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad (3.22)$$

The shape of f_{v2} is illustrated in Fig. 3.1 (left) and the magnitude of the vorticity Ω is defined by:

$$\Omega = \sqrt{2\Omega_{ij}\Omega_{ij}}, \quad \Omega_{ij} = \frac{1}{2} \left(\frac{\partial}{\partial x_j} u_i - \frac{\partial}{\partial x_i} u_j \right) \quad (3.23)$$

The function f_w (see Fig. 3.1 (right)) equals

$$f_w = g \left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{\frac{1}{6}}, \quad g = r + c_{w2} (r^6 - r), \quad r = \frac{\tilde{\nu}}{\tilde{S} \kappa^2 d^2} \quad (3.24)$$

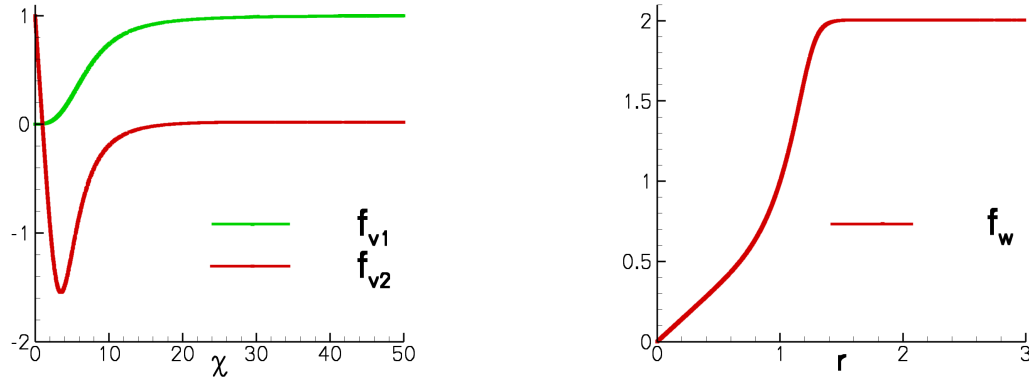


Figure 3.1: Sketch of the functions $f_{v1}(\chi)$, $f_{v2}(\chi)$ (left) and $f_w(r)$ (right) used in the Spalart-Allmaras model.

The constants are given by:

$$c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad c_{v1} = 7.1, \quad \sigma = \frac{2}{3} \quad (3.25)$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad \kappa = 0.41 \quad (3.26)$$

3.3.2 k - ω models

The Wilcox (2006) k - ω model

The Wilcox (2006) k - ω model is a modification of the original Wilcox k - ω model introduced by Wilcox [127]. While the testcases upon which the 2006-version was calibrated are not explicitly stated by Wilcox [128], it is mentioned that this new model version improves the accuracy for free shear flows and strongly separated flows. The eddy viscosity is obtained by

$$\mu_T = \frac{\bar{\rho}k}{\tilde{\omega}}, \quad \tilde{\omega} = \max \left(\omega, C_{lim} \sqrt{\frac{2\bar{S}_{ij} \bar{S}_{ij}}{\beta^*}} \right), \quad \bar{S}_{ij} = S_{ij} - \frac{1}{3} \frac{\partial}{\partial x_k} \tilde{u}_k \delta_{ij}, \quad C_{lim} = \frac{7}{8} \quad (3.27)$$

The turbulent kinetic energy k and the specific dissipation rate ω are the solutions of the following two coupled nonlinear transport equations:

$$\frac{\partial}{\partial t} (\bar{\rho}k) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j k) = \bar{\rho} \tau_{ij} \frac{\partial}{\partial x_j} \tilde{u}_i - \beta^* \bar{\rho} k \omega + \frac{\partial}{\partial x_j} \left[\left(\mu + \sigma^* \frac{\bar{\rho}k}{\omega} \right) \frac{\partial}{\partial x_j} k \right] \quad (3.28)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\bar{\rho}\omega) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j \omega) &= \alpha \frac{\omega}{k} \bar{\rho} \tau_{ij} \frac{\partial}{\partial x_j} \tilde{u}_i - \beta \bar{\rho} \omega^2 \\ &+ \sigma_d \frac{\bar{\rho}}{\omega} \frac{\partial}{\partial x_j} k \frac{\partial}{\partial x_j} \omega + \frac{\partial}{\partial x_j} \left[\left(\mu + \sigma \frac{\bar{\rho}k}{\omega} \right) \frac{\partial}{\partial x_j} \omega \right] \end{aligned} \quad (3.29)$$

The closure constants are given by:

$$\alpha = \frac{13}{25}, \quad \beta = \beta_0 f_\beta, \quad \beta^* = \frac{9}{100}, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{3}{5}, \quad \sigma_{d0} = \frac{1}{8}, \quad (3.30)$$

$$\beta_0 = 0.0708, \quad Pr_T = \frac{8}{9}, \quad \sigma_d = \begin{cases} 0 & \text{if } \frac{\partial}{\partial x_j} k \frac{\partial}{\partial x_j} \omega \leq 0 \\ \sigma_{d0} & \text{else} \end{cases} \quad (3.31)$$

$$f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega}, \quad \chi_\omega = \left| \frac{\Omega_{ij}\Omega_{jk}\widetilde{S}_{ki}}{(\beta^*\omega)^3} \right|, \quad \widetilde{S}_{ki} = S_{ki} - \frac{1}{2}\partial\partial x_m \widetilde{u}_m \delta_{ki} \quad (3.32)$$

The linearized explicit algebraic stress (LEA) model

The k - ω turbulence model LEA, which was introduced by Rung et al. [96], is a modification of the Wilcox k - ω model. k and ω are the results of Eqs. (3.28) and (3.29). The eddy viscosity is then obtained by:

$$\mu_T = \frac{C_\mu^* \bar{\rho} k}{C_\mu \omega} \quad (3.33)$$

$$C_\mu^* = \frac{\beta_1}{1 - \frac{2}{3}\eta^2 + 2\zeta^2}, \quad C_\mu = 0.09 \quad (3.34)$$

where

$$\eta^2 = \frac{\beta_3^2 T^2}{8}, \quad \zeta^2 = \frac{\beta_2^2 G^2}{2}, \quad \beta_1 = \frac{\frac{4}{3} - C_2}{2g}, \quad \beta_2 = \frac{2 - C_4}{2g}, \quad \beta_3 = \frac{2 - C_3}{g} \quad (3.35)$$

$$g = f_1(C_1 - 1) + \frac{T^2}{4 + 1.83\sqrt{0.8G^2 + 0.2T^2}} \quad (3.36)$$

$$f_1 = 1 + 0.95 \left(1 - \tanh \left(\frac{T^2}{4.6625} \right) \right) \quad (3.37)$$

$$C_1 = 2.6, \quad C_2 = \max \left(0.4, \frac{1.5T^{1.7}}{17.1 + 1.875T^{1.7}} \right), \quad C_3 = 1.25, \quad C_4 = 0.45 \quad (3.38)$$

$$T = \frac{1}{C_\mu \omega} \sqrt{2tr(S^2)}, \quad G = \frac{1}{C_\mu \omega} \sqrt{-2tr(\Omega^2)} \quad (3.39)$$

S is the strain-rate tensor defined in Eq. (2.5), Ω is given in Eq. (3.23) and tr denotes the trace of a tensor.

The Menter-SST model

The idea of the Menter-SST ("shear stress transport") model, which was introduced and reviewed by Menter et al. [74, 77], is to use a $k - \omega$ -formulation in the inner part of a boundary layer and to switch to a $k - \epsilon$ -like behaviour farther away from the wall. By doing so, the accuracy in the near-wall region of the Wilcox $k - \omega$ model is guaranteed. On the other hand the commonly known problem of $k - \omega$ models of being too sensitive to inlet-free-stream turbulence properties can be avoided. The Menter-SST model is especially useful when dealing with adverse pressure gradients and separated flows. According to Menter [74] the model was originally calibrated and validated upon the following testcases: flat plate boundary layer flow, free shear layers, adverse pressure gradient flows and the backward facing step. The equations for k and ω are given by:

$$\frac{\partial}{\partial t} (\bar{\rho}k) + \frac{\partial}{\partial x_j} (\bar{\rho}\tilde{u}_j k) = \bar{\rho}P_k - \beta^* \bar{\rho}k\omega + \frac{\partial}{\partial x_j} \left[\left(\mu + \sigma_k \frac{\bar{\rho}k}{\omega} \right) \frac{\partial}{\partial x_j} k \right] \quad (3.40)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\bar{\rho}\omega) + \frac{\partial}{\partial x_j} (\bar{\rho}\tilde{u}_j \omega) &= \gamma \frac{\omega}{k} \bar{\rho}\tau_{ij} \frac{\partial}{\partial x_j} \tilde{u}_i - \beta_\omega \bar{\rho}\omega^2 + \\ \frac{\partial}{\partial x_j} \left[\left(\mu + \sigma_\omega \frac{\bar{\rho}k}{\omega} \right) \frac{\partial}{\partial x_j} \omega \right] &+ 2\sigma_{\omega 2}(1 - F_1) \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \end{aligned} \quad (3.41)$$

where

$$P_k = \min(\bar{\rho}\tau_{ij} \frac{\partial}{\partial x_j} \tilde{u}_i, 10\beta^*k\omega)$$

The last term in in Eq. (3.41) is the so-called *cross-diffusion* term, which controls the blending between $k - \omega$ and $k - \epsilon$. The coefficients $\phi \in \{\sigma_k, \sigma_\omega, \gamma, \beta_\omega\}$ of the model are interpolated using the blending function

$$\phi = F_1\phi_1 + (1 - F_1)\phi_2$$

between those of the $k - \omega$ model, $\phi_1 \in \{\sigma_{k1}, \sigma_{\omega 1}, \gamma_1, \beta_{\omega 1}\}$ (inner layer), and those of the k/ϵ model, $\phi_2 \in \{\sigma_{k2}, \sigma_{\omega 2}, \gamma_2, \beta_{\omega 2}\}$ (outer layer), which are given by:

$$\begin{aligned} \text{Inner layer : } \quad \sigma_{k1} &= 0.85, \quad \sigma_{\omega 1} = 0.5, \quad \gamma_1 = 0.555556, \quad \beta_{\omega 1} = \beta_k \left(\gamma_1 + \frac{\sigma_{\omega 1} \kappa^2}{\sqrt{\beta_k}} \right) \\ \text{Outer layer : } \quad \sigma_{k2} &= 1.0, \quad \sigma_{\omega 2} = 0.857, \quad \gamma_2 = 0.44, \quad \beta_{\omega 2} = \beta_k \left(\gamma_2 + \frac{\sigma_{\omega 2} \kappa^2}{\sqrt{\beta_k}} \right) \end{aligned}$$

with $\kappa = 0.41$, $\beta_k = 0.09$.

The blending function F_1 (see Fig. 3.2) has to be one in the near-wall region and in the logarithmic layer and has to become zero rapidly at the outer edge of the boundary layer in order to prevent the freestream dependence of the $k - \omega$ model. F_1 is therefore defined by:

$$F_1 = \tanh(arg_1^4) , \quad arg_1 = \min \left(\max \left(\frac{\sqrt{k}}{\beta_k \omega y} ; \frac{500\nu}{y^2 \omega} \right) ; \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega} y^2} \right)$$

$$CD_{k\omega} = \max \left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} , 10^{-20} \right)$$

where $\beta_{\omega 1} = 0.07522$ and $\beta_{\omega 2} = 0.08282$.

Finally μ_t is obtained by introducing the so-called *shear-stress correction*:

$$\mu_t = \min \left(\frac{\rho k}{\omega} ; \frac{a_1 \rho k}{\Omega F_2} \right) \quad (3.42)$$

with

$$F_2 = \tanh(arg_2^2) , \quad arg_2 = \max \left(2 \frac{\sqrt{k}}{\beta_k \omega y} ; \frac{500\nu}{y^2 \omega} \right) \quad (3.43)$$

$$\overline{\Omega} = \sqrt{2\Omega : \Omega} , \quad \Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (3.44)$$

where $a_1 = 0.31$ is the Bradshaw constant and the blending function F_2 is illustrated in Fig. 3.2. The underlying idea of limiting μ_t is to restrict the tendency of two-equation models to overestimate the shear stress, which can especially be observed for flows including adverse pressure gradients.

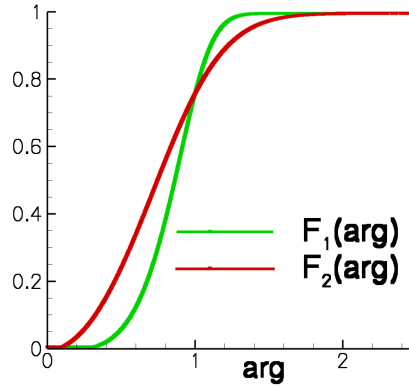


Figure 3.2: Sketch of the blending functions F_1 and F_2 in the Menter-SST model. Note, that the arguments "arg" of the two functions are not identical but are abbreviations for arg_1 and alternatively arg_2 .

3.4 Hybrid RANS/LES models

3.4.1 Spalart-Allmaras Detached Eddy Simulation (SA-DES)

Detached Eddy Simulation is a nonzonal hybrid RANS/LES model and was introduced and reviewed by Spalart et al. [110, 107]. The idea is to use the model as a

RANS-model in boundary layers and to switch to a Large Eddy Simulation (LES, [87])-like behaviour in the rest of the computational domain. If it is based on the SA-RANS model described in Subsection 3.3.1, the starting point is Eq. (3.20):

$$\frac{\partial}{\partial t}(\rho\tilde{\nu}) + \frac{\partial}{\partial x_i}(\rho u_i \tilde{\nu}) - \frac{\partial}{\partial x_k} \left[\frac{\rho(\nu + \tilde{\nu})}{\sigma} \frac{\partial}{\partial x_k} \tilde{\nu} \right] - \rho \frac{c_b 2}{\sigma} \left(\frac{\partial}{\partial x_k} \tilde{\nu} \right)^2 = P - D$$

where again production and destruction P and D are given by

$$P = c_{b1} \rho \tilde{S} \tilde{\nu}, \quad D = c_{w1} f_w \rho \left(\frac{\tilde{\nu}}{d} \right)^2$$

The SA-DES model is then obtained by replacing the original length scale d in the destruction term by \tilde{d} :

$$\tilde{d} = \min(d, C_{\text{DES}} \Delta), \quad \Delta = \max[\Delta x, \Delta y, \Delta z], \quad C_{\text{DES}} = 0.65 \quad (3.45)$$

Within a boundary layer $\tilde{d} = d$, resulting in the normal SA-RANS model introduced in Subsection 3.3.1.

Farther away from a wall $\tilde{d} = C_{\text{DES}} \Delta$. Now P and D are balanced:

$$c_{b1} \rho \tilde{S} \tilde{\nu} = c_{w1} f_w \rho \left(\frac{\tilde{\nu}}{\tilde{d}} \right)^2 \quad (3.46)$$

If for simplicity $f_w = \text{const}$ is assumed, one obtains

$$\tilde{\nu} \sim \tilde{S} \tilde{d}^2 \quad (3.47)$$

This is analogous to the Smagorinsky model, which is an LES model, in which the kinematic eddy viscosity is given by:

$$\nu_T = (C_S \Delta)^2 \sqrt{\mathbb{S}_{ij} \mathbb{S}_{ij}} \quad (3.48)$$

Here the resolved strain-rate tensor \mathbb{S}_{ij} is defined by:

$$\mathbb{S}_{ij} = \frac{1}{2} \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) - \frac{1}{3} \frac{\partial \langle u_k \rangle}{\partial x_k} \quad (3.49)$$

and C_s is a constant which can be problem-dependent.

The operator $\langle \cdot \rangle$ denotes a filtering procedure applied in LES. The idea lies in computing the resolvable large eddies, while the smaller subgrid-scale eddies are modeled.

As both the filtering procedure used in LES and Favre averaging applied in RANS contains the same properties regarding linearity and commutativity, the governing equations used in LES take the same form as Eqs. (3.8) - (3.10) - with the only difference that the equations contain the Favre averaged and filtered variables, respectively.

3.4.2 SA-Delayed Detached Eddy Simulation (SA-DDES)

One problem of the just described SA-DES is, that the filter Δ is only based on grid properties but not on the actual flow. If a fine grid is used, the LES-part of the SA-DES model extends into the boundary layer (this is called "underresolved LES"), which can lead to grid induced flow separation. Another problem of the SA-DES model lies in the fact, that the RANS-LES partition differs for varying grids. A grid convergence study is therefore almost impossible, because different grids then operate in different turbulence models on parts of the computational domain.

In order to adress these problems, the SA-DDES model, which is an adjustment of the SA-DES model, has been introduced by Spalart et al. [109]. SA-DDES was originally calibrated and validated upon flat plate boundary layer flow, flow around circular cylinders, flow around a single airfoil and the flow of a backward facing step. The length scale \tilde{d} is here defined by:

$$\tilde{d} = d - f_d \cdot \max(0, d - \Psi C_{\text{DES}} \Delta), \quad \Delta = \max[\Delta x, \Delta y, \Delta z] \quad (3.50)$$

$$\boxed{f_d = 1 - \tanh(8r_d^3), \quad r_d = \frac{\nu + \nu_t}{S_{ij} \kappa^2 d^2}, \quad \kappa = 0.41} \quad (3.51)$$

As the function f_d defined in Eq. (3.51) is subject to a detailed research in Section 6, it's definition is highlighted by a surrounding box.

Within a boundary layer $f_d \equiv 0$, leading to $\tilde{d} = d$ and hereby ensuring normal SA-RANS-mode. Farther away from the wall $f_d \equiv 1$, resulting in $\tilde{d} = \min(d, C_{\text{DES}} \Delta)$, which is the original SA-DES model.

The term Ψ in Eq. (3.50) is the so-called low-Reynolds modification, which is required to obtain an Smagorinsky-like behaviour at locally low eddy-viscosity levels. It is defined by:

$$\Psi^2 = \frac{1 - c_{b1} f_{v2} / (c_{w1} \kappa^2 f_w^*)}{f_{v1}} \quad (3.52)$$

Here f_{v1} and f_{v2} are functions already used in the SA-RANS model (they are defined in Eqs. (3.19) and (3.22), respectively), and $f_w^* = 0.427$.

Chapter 4

The Chimera technique: a domain decomposition method

A major focus of this thesis lies on the Chimera technique, which is a special kind of domain decomposition (DD) method. The aim of this chapter is to examine both the underlying ideas and classifications of DD methods in general and take a closer look at the Chimera technique in more detail. This chapter is organised as follows:

- Firstly an introduction into DD methods is given in Section 4.1 by considering a simple one-dimensional Poisson problem. The aim is to provide an understanding of the basic principles applied in these techniques.
- The main reason, why the Chimera technique is used within this thesis lies in its ability to move grids relative to each other during a computation. Therefore secondly the different possibilities to position domains (not necessarily grids) relative to each other are examined in Section 4.2. As we will see, the Chimera technique will turn out to be the most useful method for the applications within this thesis.
- Thirdly the Chimera technique is presented in detail in Section 4.3. At the beginning an algorithm to solve the one-dimensional Poisson problem from Section 4.1 using the Chimera technique is shown. Then the implementation of the Chimera technique in the DLR TAU-code is examined for unsteady two-dimensional problems.

Within this chapter, especially the papers of Houzeau et al. [53, 54] and Toselli et al. [120] have been extensively used within Sections 4.1 and 4.2. DD methods are farther examined by Smith et al. [105]. The description of the implementation in the DLR TAU-code in Section 4.3 is based on the work by Schwarz [99].

4.1 An introduction into domain decomposition methods

The very basic idea of DD methods is to split a boundary value problem defined on a domain $\Omega \subset \mathbb{R}^n$, $n = 1, 2, 3$, into smaller boundary value problems defined on subdomains $\Omega_i \subset \Omega$, $i = 1, \dots, n$, where $\cup_i \Omega_i = \Omega$. To do so in a consistent way, adequate boundary conditions must be prescribed at the interfaces of the subdomains.

Having solved the local problems, the resulting local solutions u_i on subdomains Ω_i must then be united appropriately to obtain the global solution u of the original boundary value problem defined on Ω .

The major advantage of DD methods is that the local problems on subdomains Ω_i can be solved independently from each other. This enables four major fields of applications for DD methods:

Firstly DD methods can be used as an easy and straightforward way to parallelize a numerical code.

Secondly the meshing of complex geometries can be simplified by the help of DD methods: Instead of generating one single grid for the complete configuration (which can be very hard to achieve and often leads to strongly deformed grid cells), the domain is split into several subdomains such that the meshing can be performed independently on each part. This can simplify the meshing process immensely.

Thirdly DD methods can be used within zonal methods, meaning that different models are used on the various subdomains. While in the SA-DDES model described in Subsection 3.4.2 the RANS-LES switch is based on the length scale \tilde{d} (implying that SA-DDES is a nonzonal method), in zonal DES (or ZDES) [93] the RANS-zone and the LES-zone are defined explicitly, where at the interface of the two zones the flow variables of one zone are used as boundary condition for the relatively other zone. ZDES can therefore be considered as a DD method. Another example of zonal methods is the so-called GOALS (Goal oriented adaptive local solution)-algorithm [83, 131]. Here firstly a *homogenised problem* is solved globally on Ω , where the underlying grid can be relatively coarse. Then another model, the so-called *fine-scale model*, is solved on a subdomain $\Omega_L \subset \Omega$ by using the homogenised solution as Dirichlet boundary condition at the boundary of Ω . For the fine-scale problem a very fine grid must be chosen within Ω_L . The principle of the GOALS-algorithm is to use the cheap but less accurate homogenised model in regions where it is sufficient and only to switch to the more expensive but at the same time more accurate fine-scale model, where it is necessary in order to minimise a prescribed local error.

The fourth field of DD methods is the most important one for the applications of this thesis: DD methods allow to comfortably move solid bodies or flow properties such as vortices relative to each other within Ω .

Despite the variety of DD methods, they can be divided into only two classes:

1) In *nonoverlapping DD methods* the interior of all subdomains Ω_i are disjoint, i.e. $\mathring{\Omega}_i \cap \mathring{\Omega}_j = \emptyset \forall i, j$, where $\mathring{\Omega}_k$ denotes the interior of Ω_k . A special class of non-overlapping DD methods are the so-called *Mortar methods*, which are discretization methods for partial differential equations, where different discretizations on the var-

ious Ω_i are used. As the grids of the subdomains in general do not match at the interface, Lagrange multipliers are used in that region, which enforces equal solutions on all adjacent subdomains.

2) In *overlapping DD methods* neighbouring subdomains are not disjoint, i.e. $\exists i, j : \Omega_i \cap \Omega_j \neq \emptyset$. This class of DD methods include the *additive Schwarz method* (see Subsection 4.1.3) and the *alternating Schwarz method* [62].

To present a more detailed examination of these two different types of DD methods we regard the following simple one-dimensional Poisson problem:

$$\left. \begin{aligned} -\frac{\partial u^2}{\partial x^2} &= f & \text{if } x \in \Omega := (-1, 1) \\ u &= 0 & \text{if } x = -1, 1 \end{aligned} \right\} \quad (4.1)$$

Next Ω is divided into subdomains Ω_1 and Ω_2 (disjoint or overlapping) by setting $\Omega_1 = (-1, \delta)$ and $\Omega_2 = (-\delta, 1)$, where $0 \leq \delta \leq 1$.

In order to construct well-defined local boundary value problems with unique solutions on Ω_1 and Ω_2 , an additional consistent boundary condition at $x = \delta$ and $x = -\delta$, respectively, must be prescribed. Note that in general this is a necessary but not sufficient condition to recover the global solution u from the local solutions u_i . The following three choices of boundary conditions are possible: Either the variable u itself can be prescribed (which is a so-called *Dirichlet* condition), its first derivative can be set (which is called *Neumann* condition) or most generally a linear combination of both can be chosen (*Robin* condition).

By using a Robin condition, one obtains the following boundary value problem on subdomain Ω_1 :

$$\left. \begin{aligned} -\frac{\partial u_1^2}{\partial x^2} &= f & \text{if } x \in \Omega_1 \\ u_1 &= 0 & \text{if } x = -1 \\ \alpha_1 u_1 + \beta_1 \frac{\partial u_1}{\partial x} &= g_1 & \text{if } x = \delta \end{aligned} \right\} \quad (4.2)$$

where α_1 and β_1 are constants such that $\alpha_1 \neq 0$ or $\beta_1 \neq 0$. Similarly, the following boundary value problem is considered on subdomain Ω_2 :

$$\left. \begin{aligned} -\frac{\partial u_2^2}{\partial x^2} &= f & \text{if } x \in \Omega_2 \\ u_2 &= 0 & \text{if } x = 1 \\ \alpha_2 u_2 + \beta_2 \frac{\partial u_2}{\partial x} &= g_2 & \text{if } x = -\delta \end{aligned} \right\} \quad (4.3)$$

where α_2 and β_2 are constants such that $\alpha_2 \neq 0$ or $\beta_2 \neq 0$. The coefficients α_1 and α_2 are called *Robin coefficients*. The aim of decomposition methods lies in recovering the global solution u of problem (4.1) from solutions u_1 and u_2 . For this we must specify functions g_1 and g_2 such that:

$$u_1 = u|_{\Omega_1}$$

$$u_2 = u|_{\Omega_2}$$

We first consider the case of nonoverlapping subdomains Ω_1 and Ω_2 , i.e. $\delta = 0$:

4.1.1 Nonoverlapping subdomains

Let $\delta = 0$. Then for $f \in \mathbb{L}^2(\Omega)$, problem (4.1) can be transformed into the following coupled problem:

$$\left. \begin{aligned} -\frac{\partial u_1^2}{\partial x^2} &= f & \text{if } x \in \Omega_1 \\ u_1 &= 0 & \text{if } x = -1 \\ u_1 &= u_2 & \text{if } x = 0 \\ \frac{\partial u_1}{\partial x} &= \frac{\partial u_2}{\partial x} & \text{if } x = 0 \\ -\frac{\partial u_2^2}{\partial x^2} &= f & \text{if } x \in \Omega_2 \\ u_2 &= 0 & \text{if } x = 1 \end{aligned} \right\} \quad (4.4)$$

It is now straightforward to see that problem (4.4) is equivalent to problems (4.2) and (4.3) for the following choices of g_1 and g_2 :

$$g_1 = \alpha_1 u_2 + \beta_1 \frac{\partial u_2}{\partial x} \quad (4.5)$$

$$g_2 = \alpha_2 u_1 + \beta_2 \frac{\partial u_1}{\partial x} \quad (4.6)$$

where the coefficients $\alpha_1, \alpha_2, \beta_1, \beta_2$ must be chosen such that

$$|\alpha_1 \beta_2 - \alpha_2 \beta_1| \neq 0 \quad (4.7)$$

Note, that in system (4.4) both the continuity of u and of $\frac{\partial u}{\partial x}$ is strictly required at $x = 0$. Imposing only the continuity of u by setting $\alpha_1 = \alpha_2 = 1$ and $\beta_1 = \beta_2 = 0$ may lead to $\frac{\partial u_1}{\partial x} \neq \frac{\partial u_2}{\partial x}$ at $x = 0$, which is indicated in Fig. 4.1 (left). If on the other hand only the continuity of $\frac{\partial u}{\partial x}$ is prescribed at $x = 0$ (which is achieved by choosing $\alpha_1 = \alpha_2 = 0$ and $\beta_1 = \beta_2 = 1$), u in general is not continuous at $x = 0$, which is shown in Fig. 4.1 (right). In both cases u can not be recovered from the local solutions u_1 and u_2 , i.e. $u_1 \neq u|_{\Omega_1}$ and $u_2 \neq u|_{\Omega_2}$.

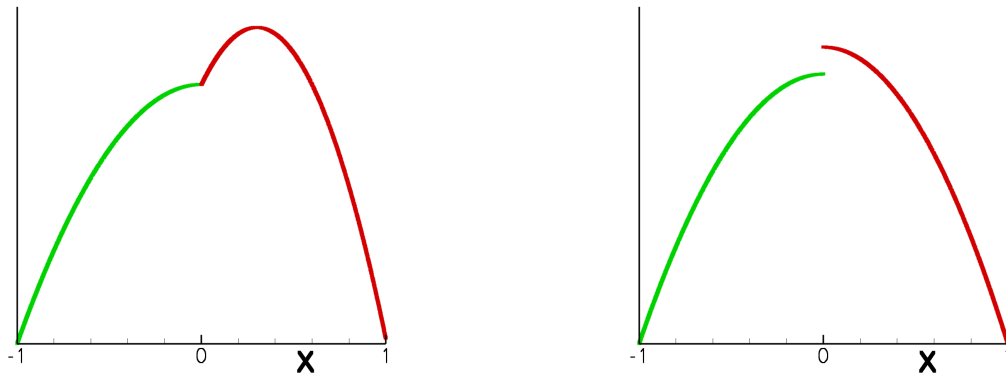


Figure 4.1: Nonoverlapping subdomains: Local solutions u_1 ("green") and u_2 ("red") if only $u_1 = u_2$ is imposed at $x = 0$ (left) and if only $\frac{\partial u_1}{\partial x} = \frac{\partial u_2}{\partial x}$ is imposed at $x = 0$ (right).

Only if both boundary conditions are fulfilled at $x = 0$, the global solution u of problem (4.1) can be regained by setting:

$$u = \begin{cases} u_1 & \text{in } \Omega_1 \\ u_2 & \text{in } \Omega_2 \end{cases} \quad (4.8)$$

which is shown in Fig. 4.2.

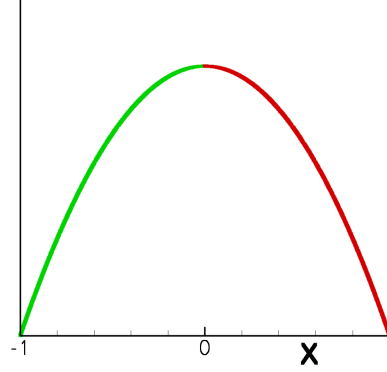


Figure 4.2: Nonoverlapping subdomains: Local solutions u_1 ("green") and u_2 ("red") if both $u_1 = u_2$ and $\frac{\partial u_1}{\partial x} = \frac{\partial u_2}{\partial x}$ are imposed at $x = 0$.

In particular we have shown in this subsection that neither a Dirichlet/Dirichlet coupling nor a Neumann/Neumann coupling is admissible in order to recover u from u_1 and u_2 .

4.1.2 Overlapping subdomains

Now the case of overlapping subdomains is considered, i.e. $\delta > 0$. We will show that choosing the same functions g_1 and g_2 as in the case of nonoverlapping subdomains is sufficient to recover the global solution u of problem (4.1) by the local solutions u_1 and u_2 . We first prove that u_1 and u_2 are identical on the overlap of Ω_1 and Ω_2 .

Lemma 4.1.1 (Equality of u_1 and u_2 in $\Omega_1 \cap \Omega_2$)

For $\delta > 0$ let u_1 and u_2 be the solutions to problems (4.2) and (4.3), respectively, where g_1 and g_2 are given by Eqs. (4.5) and (4.6) such that condition (4.7) holds. Then $u_1 \equiv u_2$ in $\Omega_1 \cap \Omega_2$.

Proof: Let $w = u_1 - u_2$. By subtracting Eq. (4.2)₁ from Eq. (4.3)₁ and Eq. (4.2)₃ from Eq. (4.3)₃ using Eqs. (4.5) and (4.6) we obtain the following boundary value problem for w on $\Omega_1 \cap \Omega_2$:

$$\left. \begin{aligned} \frac{\partial w^2}{\partial x^2} &= 0 & \text{if } x &\in \Omega_1 \cap \Omega_2 = (-\delta, \delta) \\ \alpha_2 w + \beta_2 \frac{\partial w}{\partial x} &= 0 & \text{if } x &= -\delta \\ \alpha_1 w + \beta_1 \frac{\partial w}{\partial x} &= 0 & \text{if } x &= \delta \end{aligned} \right\} \quad (4.9)$$

Due to condition (4.7) (which in particular guarantees either $\alpha_1 \neq 0$ or $\alpha_2 \neq 0$) problem (4.9) is well-defined and therefore has a unique solution $w \equiv 0$. \square

As u_1 and u_2 are the solutions to problems (4.2) and (4.3), respectively, the continuity of the derivatives of u_1 at $x = -\delta$ as well as u_2 at $x = \delta$ is furthermore guaranteed. We therefore can recover the global solution u of problem (4.1) by setting:

$$u = \begin{cases} u_1 & \text{in } \Omega_1 \\ u_2 & \text{in } \Omega \setminus \Omega_1 \end{cases} \quad (4.10)$$

or alternatively by

$$u = \begin{cases} u_1 & \text{in } \Omega \setminus \Omega_2 \\ u_2 & \text{in } \Omega_2 \end{cases} \quad (4.11)$$

which is indicated in Fig. 4.3.

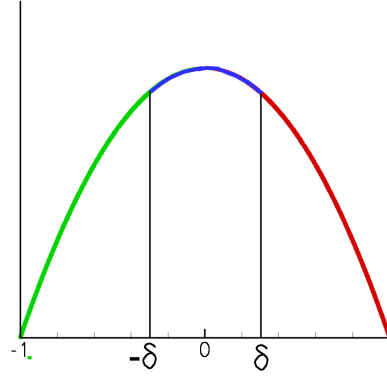


Figure 4.3: Overlapping subdomains: Local solutions u_1 ("green"), u_2 ("red") and identical solution on $\Omega_1 \cap \Omega_2$ ("blue").

Note that in the case of $\alpha_1 = \alpha_2 = 0$, $\beta_1 \neq 0$ and $\beta_2 \neq 0$ (which is an overlapping Neumann/Neumann coupling) the solution w of problem (4.9) is defined up to an additive constant.

4.1.3 Relation between nonoverlapping and overlapping DD methods

We again consider problems (4.2) and (4.3), where $\delta > 0$. The *additive Schwarz method* is then obtained by choosing the following boundary conditions:

$$\begin{aligned} u_1 &= u_2 & \text{if } x &= \delta \\ u_2 &= u_1 & \text{if } x &= -\delta \end{aligned} \quad (4.12)$$

When now performing a forward and backward Taylor-expansion at $x = 0$, these two Dirichlet conditions can be transformed to the following Robin/Robin coupling

at $x = 0$:

$$\begin{aligned} u_1 + \delta \frac{\partial u_1}{\partial x} &= u_2 + \delta \frac{\partial u_2}{\partial x} \\ u_2 - \delta \frac{\partial u_2}{\partial x} &= u_1 - \delta \frac{\partial u_1}{\partial x} \end{aligned} \quad (4.13)$$

It is hereby shown that a DD method on overlapping subdomains using a Dirichlet/Dirichlet coupling is equivalent to a DD method on nonoverlapping subdomains using a Robin/Robin coupling. The latter DD method is therefore also referred to as *fictitious overlapping method* [117].

4.1.4 An iterative algorithm to solve DD methods

To solve local problems (4.2) and (4.3) numerically, the following iterative algorithm can be used:

Definition 4.1.2 (Iterative algorithm for DD methods)

Choose initial guesses u_1^0 and u_2^0 as solutions for the local problems (4.2) and (4.3). Then for all $k \neq 0$ successively solve the following two problems:

$$\left. \begin{aligned} -\frac{\partial^2 u_1^{k+1}}{\partial x^2} &= f & \text{if } x \in \Omega_1 \\ u_1^{k+1} &= 0 & \text{if } x = -1 \\ \alpha_1 u_1^{k+1} + \beta_1 \frac{u_1^{k+1}}{\partial x} &= \alpha_1 u_2^k + \beta_1 \frac{u_2^k}{\partial x} & \text{if } x = \delta \end{aligned} \right\} \quad (4.14)$$

$$\left. \begin{aligned} -\frac{\partial^2 u_2^{k+1}}{\partial x^2} &= f & \text{if } x \in \Omega_2 \\ u_2^{k+1} &= 0 & \text{if } x = 1 \\ \alpha_2 u_2^{k+1} + \beta_2 \frac{u_2^{k+1}}{\partial x} &= \alpha_2 u_1^{k'} + \beta_2 \frac{u_1^{k'}}{\partial x} & \text{if } x = -\delta \end{aligned} \right\} \quad (4.15)$$

The index k' is hereby given by:

$$k' = \begin{cases} k & \text{if problems (4.14) and (4.15) are solved parallelly} \\ k+1 & \text{if problems (4.14) and (4.15) are solved sequentially} \end{cases} \quad (4.16)$$

This iterative procedure is performed until $\|u_1^{k+1} - u_1^k\| \leq tol$ and $\|u_2^{k+1} - u_2^k\| \leq tol$, where $tol > 0$ is an error tolerance parameter and $\|\cdot\|$ a suitably chosen norm.

4.2 Moving subdomains

In several applications presented in part II of this thesis solid bodies or vortices have to be moved within the computational domain Ω . This subsection aims to examine the different ways to move domains relative to each other.

Mainly four different approaches are possible:

1) In the *sliding mesh technique* different meshes can be moved along their common interfaces, which is illustrated in Fig. 4.4. This technique is mainly applied when

Mortar methods are combined with moving subdomains, where Mortar methods use separate finite element discretizations on nonoverlapping subdomains. As the different grids by definition are disjoint, this approach can not be comfortably used when more than one entity (which can be a solid object or a vortex) has to be tracked. As in the numerical applications in this thesis up to three entities are moved relative to each other, the use of the sliding mesh technique is not advisable here.

The sliding mesh technique is a nonoverlapping DD method.

2) The *fictitious domain technique* [38, 39] enables to track solid bodies inside a single background mesh, which is shown in Fig. 4.5. The boundary conditions at the body are hereby enforced directly into the flow equations with the help of Lagrange multipliers. As the fictitious domain technique can only be applied to track solid bodies but not vortices, this method obviously cannot be used for the applications of this thesis.

The fictitious domain method is no DD method, because the computation is only performed on one domain.

3) The *arbitrary-Lagrangian-Eulerian (ALE) technique* [40] combined with an adaptive remeshing technique adapts the mesh in time to track the specified entity (see Fig. 4.6). This technique works well, if the displacements are small, because then only a minor grid adaptation is required. While the ALE-approach in principle also allows to move both solid objects and vortices over large distances, in practice this leads to several problems. The main drawback of this technique is that the complete mesh has to be remeshed after every timestep, which is numerically very expensive, if the adaptation includes large parts of the domain. Additionally the adaption can lead to a deformed grid with badly shaped grid cells, leading to an increased numerical error. If the underlying numerical code furthermore does not allow the use of hanging grid nodes (which holds true for the DLR TAU-code), the ALE approach can only be performed on unstructured grids. In the case of tracking a vortex this can result in an increased vortex dissipation rate, which is undesirable.

The ALE technique is no DD method, because the computation is only performed on one domain.

4) The *Chimera technique*, which is also called *overset grid technique*, uses multiple grids for each moving component and couples them by interpolation. This technique, which is illustrated in Fig. 4.7, comfortably allows to move both solid objects and vortices relative to each other. A farther advantage is that the use of structured subgrids is problem-free. One disadvantage of the Chimera technique lies in the overlapping part of two grids: In this area the underlying equations have to be solved twice, which increases the computational cost. Nevertheless the Chimera technique seems to be the most useful and flexible of all presented methods for the performed applications in this thesis.

The Chimera technique is an overlapping DD method.

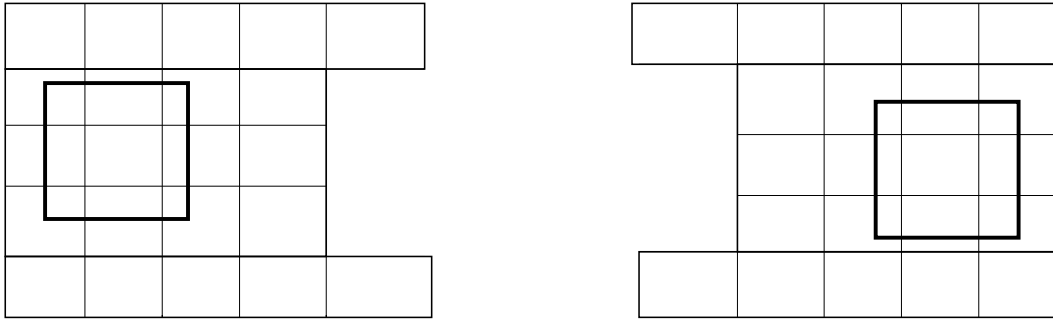


Figure 4.4: Visualisation of the sliding mesh technique

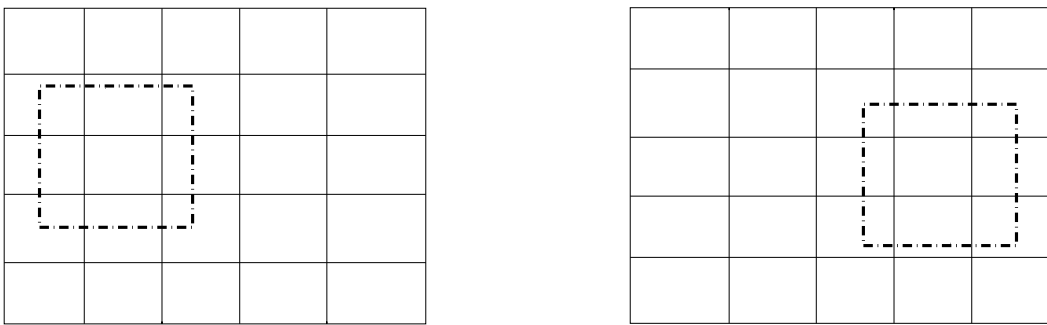


Figure 4.5: Visualisation of the fictitious domain technique

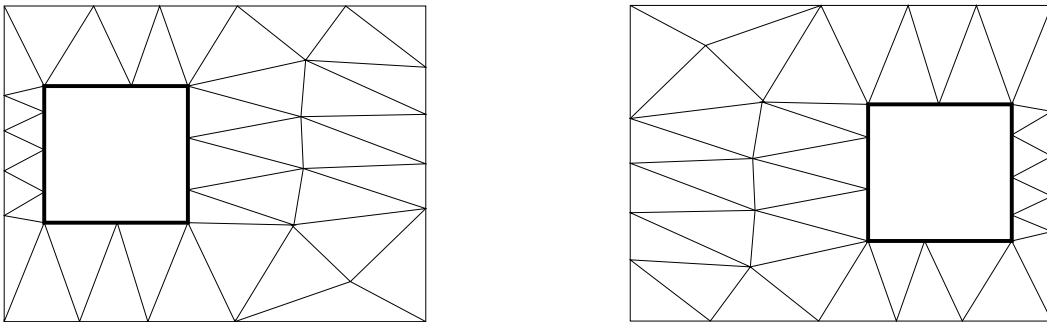


Figure 4.6: Visualisation of the ALE technique

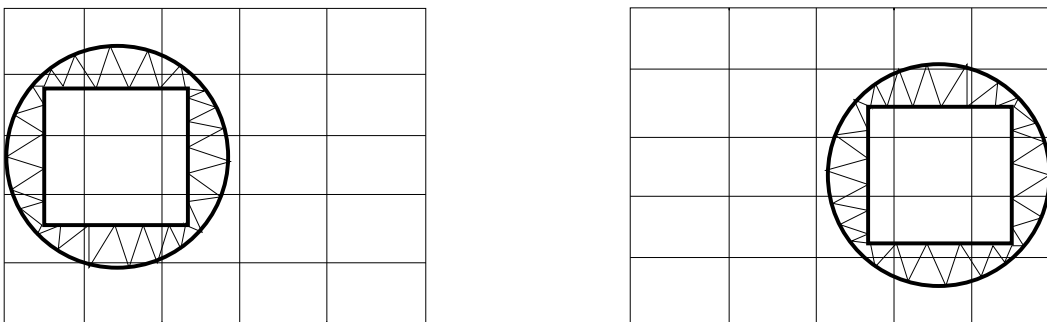


Figure 4.7: Visualisation of the Chimera technique

4.3 The Chimera technique

The Chimera technique allows flow solutions on overlapping grids, where the grids can be in relative motion to each other during a simulation.

First we want to demonstrate how the Chimera technique can be used to numerically solve the one-dimensional Poisson problem defined in equations (4.1). Identical as in Subsection 4.1.2 Ω is divided into two overlapping subdomains $\Omega_1 = (-1, \delta)$ and $\Omega_2 = (-\delta, 1)$, where $\delta > 0$. Figure 4.8 shows the Chimera setting, where the indicated green points belong to a grid g_1 that covers Ω_1 , while the red points are part of a second grid g_2 covering Ω_2 .

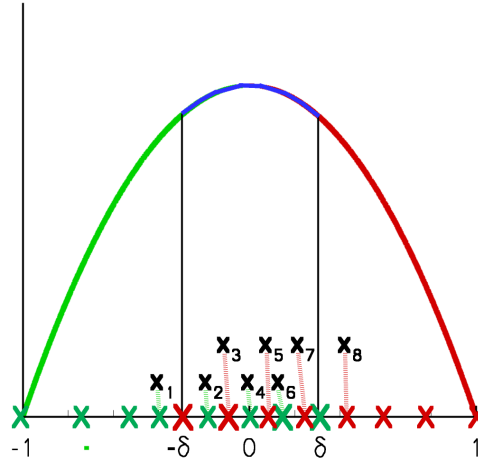


Figure 4.8: Overlapping subdomains using the Chimera technique: Local solutions u_1 ("green") on grid g_1 (indicated by green points) and u_2 ("red") on grid g_2 (indicated by red points). The local solution on $\Omega_1 \cap \Omega_2$ ("blue") is identical on both g_1 and g_2 except for a second order spatial error, if the algorithm from definition 4.3.1 has converged.

If a second order spatial discretisation scheme (e.g. the central scheme introduced in Subsection 2.2.3) is applied, a linear Dirichlet interpolation on respectively two grid points is required on both grids to maintain the scheme's second order consistency in space. In the setting shown in Fig. 4.8 the two grid points at $x = x_6$ and $x = \delta$, which belong to grid g_1 , and the two points at $x = -\delta$ and $x = x_3$, which are part of g_2 , are interpolation points that obtain their values by a linear interpolation from the relative other grid. The modified iterative algorithm to solve problem (4.1) is then given by:

Definition 4.3.1 (Modified iterative algorithm using Chimera)

Choose initial guesses u_1^0 and u_2^0 as solutions for the local problems (4.2) and (4.3). Then for all $k \neq 0$ successively solve the following two problems:

$$\left. \begin{aligned} -\frac{\partial^2 u_1^{k+1}}{\partial x^2} &= f & \text{if } x &\in \Omega_1 \\ u_1^{k+1} &= 0 & \text{if } x &= -1 \\ u_1^{k+1} &= u_2^k(x_7) + \frac{x_7-x_6}{x_7-x_5}(u_2^k(x_5) - u_2^k(x_7)) & \text{if } x &= x_6 \\ u_1^{k+1} &= u_2^k(x_8) + \frac{x_8-\delta}{x_8-x_7}(u_2^k(x_7) - u_2^k(x_8)) & \text{if } x &= \delta \end{aligned} \right\} \quad (4.17)$$

$$\left. \begin{aligned} -\frac{\partial^2 u_2^{k+1}}{\partial x^2} &= f & \text{if } x &\in \Omega_2 \\ u_2^{k+1} &= 0 & \text{if } x &= 1 \\ u_2^{k+1} &= u_1^{k'}(x_2) + \frac{x_2-(-\delta)}{x_2-x_1}(u_1^{k'}(x_1) - u_1^{k'}(x_2)) & \text{if } x &= -\delta \\ u_2^{k+1} &= u_1^{k'}(x_4) + \frac{x_4-x_3}{x_4-x_2}(u_1^{k'}(x_2) - u_1^{k'}(x_4)) & \text{if } x &= x_3 \end{aligned} \right\} \quad (4.18)$$

The index k' is hereby given by:

$$k' = \begin{cases} k & \text{if problems (4.17) and (4.18) are solved parallelly} \\ k+1 & \text{if problems (4.17) and (4.18) are solved sequentially} \end{cases} \quad (4.19)$$

This iterative procedure is performed until $\|u_1^{k+1} - u_1^k\| \leq \text{tol}$ and $\|u_2^{k+1} - u_2^k\| \leq \text{tol}$, where $\text{tol} > 0$ is an error tolerance parameter and $\|\cdot\|$ a suitably chosen norm.

As in Chapters 7 and 8 of this thesis unsteady two- and three-dimensional applications of the Chimera technique are presented, we next consider the implementation of the Chimera technique in the DLR TAU-code for these cases. The major difference to the one-dimensional Poisson problem just shown is that now additionally a hole must be cut into one of the grids. Hole cutting is required, if one of the grids contains a solid body, if a grid lies fully embedded within another grid or if the overlapping region of two grids is large.

To explain the principle of the implementation it is sufficient to examine the two-dimensional case of a small cartesian grid (SG) that lies embedded within a large cartesian grid (LG). The extension to three dimensions, the use of more than two grids, unstructured or hybrid grids and grids that contain solid bodies is then straightforward.

The algorithm can be divided into grid generation, hole cutting, interpolation & solution computation and grid movement and can be stated as follows:

1. Grid generation

Generate the SG and the LG and specify their initial relative position. Figure 4.9 (left) shows the SG (solid blue lines) placed within the LG (dashed red lines). The solid black lines belong to a so-called "hole definition grid (HDG)", which is required in step 2.

2. Hole cutting

As the SG lies fully embedded within the LG, those grid points of the LG that lie

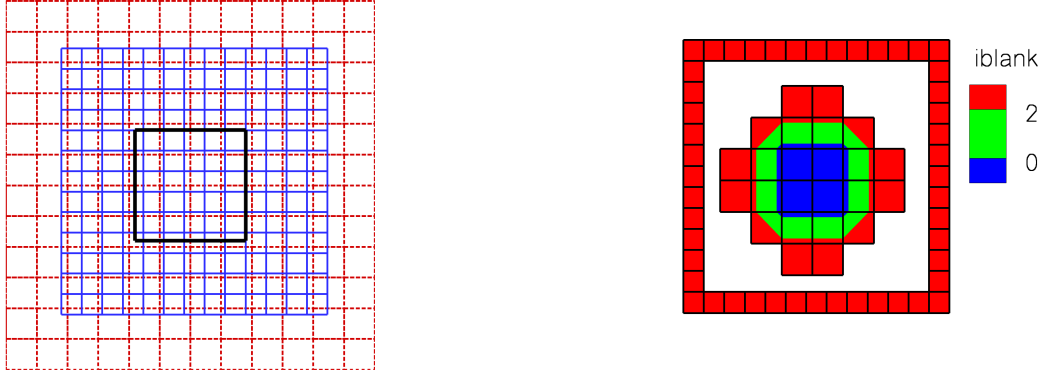


Figure 4.9: Small cartesian grid ("SG", solid blue lines) and boundary of a hole definition grid ("HDG", solid black lines) placed within the large cartesian grid ("LG", dashed red lines) (left) and corresponding grid points with *iblack*-values 0 or 2 (right)

within the region covered by the center of the SG must be excluded from the computation. This action is commonly referred to as "hole cutting". For this purpose first an Integer-variable "iblack" is assigned to each grid point of both the SG and the LG. If "*iblack_i*" denotes the value of grid point *i*, then this variable can take the following values:

- *iblack_i* = 0: Grid point *i* is located in the interior of a hole and is therefore excluded from the computation.
- *iblack_i* = 1: On grid point *i* the computation is performed normally.
- *iblack_i* = 2: Grid point *i* belongs to a Chimera boundary and obtains its data by interpolation from the respectively other grid.

Initially all *iblack*-values are set to 1. Next a HDG is generated and placed in the middle of the SG, see Fig. 4.9 (left). A HDG usually consists of only one or very few cells (in the case of cutting a hole around a complex geometry, one is possibly forced to use a more elaborated HDG) and should be significantly smaller than the SG to ensure a sufficiently large overlapping region of the SG and the active part of the LG. Now the hole is cut: If a grid point of the LG lies within the domain covered by the HDG, its *iblack*-value is set to 0. How these points can be determined even in complex geometries is described by Petersson [86].

3. Interpolation and solution computation

Next the boundary conditions both at the inner boundary of the LG and at the outer boundary of the SG have to be specified. The former boundary consists of those points of the LG, whose spatial discretisation operator would access data from points with an *iblack*-value of 0. In the case of using a central scheme for the spatial

discretisation, this boundary is composed of a two point layer around the hole. Due to the same argumentation, the latter boundary is also comprised of a layer of two points. The points lying in either of these two boundaries are marked by setting their *iblack*-values to 2. Figure 4.9 (right) shows all grid points containing an *iblack*-value of 0 or 2, where the points with a value of 1 are omitted for better visibility. In the inner block - which is made up of points of the LG - the hole consisting of 9 points (*iblack*-value 0) is surrounded by the points of the inner boundary of the LG (*iblack*-value 2). The points in the outer frame (*iblack*-value 2) belong to the outer boundary of the SG.

The concept of the boundary condition introduced lies in interpolating data from the respectively other grid, while it is a priori not clear, which data should be chosen. As Wu [132] shows that the interpolation of the numerical fluxes leads to problems, the conservative variables are used instead. The accuracy of the interpolation is prescribed as trilinear (for a 3D-computation), as this is sufficient for maintaining the numerical scheme's second order consistency in space according to Chesshire [20]. The task of finding appropriate donor grid points, from which the data can be obtained for interpolation, is discussed in detail in by Schwarz [99] and Holst [46]. When using dual time stepping, the original k -step low storage Runge-Kutta scheme defined in Eq. (2.33) is replaced by the following modified Runge-Kutta scheme:

IF (*iblack*_{*i*} \equiv 2)

Obtain \vec{w}_i^{n+1} by interpolation

ELSE

$$\begin{aligned} \vec{w}_i^{(0)} &= \vec{w}_i^n \\ \vec{w}_i^{(1)} &= \vec{w}_i^{(0)} - \text{iblack}_i \cdot \alpha_1 \Delta t \vec{R}(\vec{w}_i^{(0)}) \\ &\vdots \\ \vec{w}_i^{(k)} &= \vec{w}_i^{(0)} - \text{iblack}_i \cdot \alpha_k \Delta t \vec{R}(\vec{w}_i^{(k-1)}) \\ \vec{w}_i^{n+1} &= \vec{w}_i^{(k)} \end{aligned}$$

where $\alpha_1, \dots, \alpha_k$ are the normal Runge-Kutta coefficients.

4. Grid movement

If the computation has not terminated yet, optionally move simultaneously the SG and HDG relative to the LG and go back to 2. It is important to emphasize that the HDG remains attached to the SG during the complete computation, meaning that their relative position never changes.

Part II

Numerical applications

Chapter 5

Flow over a backward facing step

As the flow over a backward facing step comprises a large region of massively separated flow, it is considered a standard testcase for DES models. For this testcase the separation point is determined by the geometry. The aim of this section is to show that the use of DES leads to reliable results for setups with massive separation. This model validation is necessary with regard to Chapter 8 of this thesis, where the flow around a two-element FNG airfoil near stall both at undisturbed and disturbed onflow conditions is considered. As one can see in Figs. 5.1 (left) and (right), where the velocity streamlines of the backward facing step testcase and alternatively of the flow around the FNG airfoil at undisturbed onflow conditions is shown, both testcases include a vast separation region and can in this spirit be regarded as similar. Both results are based on steady two-dimensional SA-RANS simulations and the angle of attack of the FNG-airfoil is $\alpha = 10^\circ$, which is close to stall.

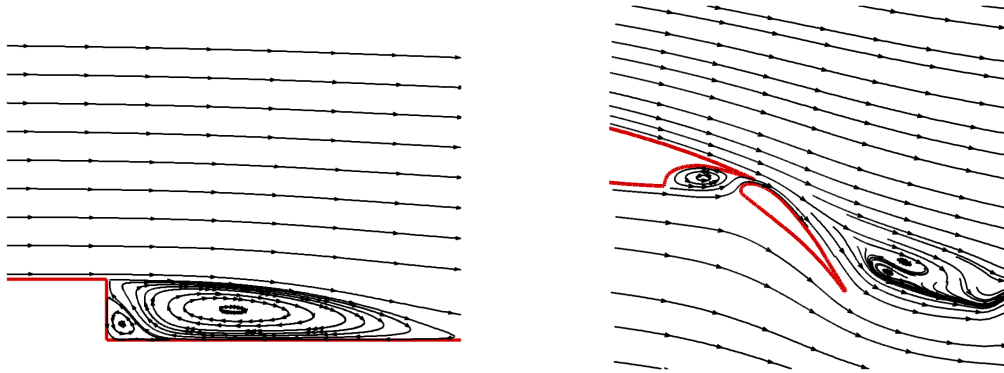


Figure 5.1: Comparison of velocity streamlines obtained by steady two-dimensional SA-RANS simulations of backward facing step testcase (left) and of flow around two-element FNG airfoil at $\alpha = 10^\circ$ (right).

5.1 Description of the testcase

Due to the availability of a variety of experimental data to compare with, the testcase of Driver et al. [27] is considered in this chapter.

Let us first define the computational domain: Based on a step height of $h = 0.0127\text{m}$, a length of $4h$ is prescribed before the step; behind the step the domain covers $25h$. The domain height is given by $8h$ upstream, and $9h$ downstream of the step, respectively. In the three-dimensional computations a spanwise extension of $4h$ is applied. Figure 5.2 shows a sketch of the three-dimensional domain, where additionally the main flow direction in the center of the domain is indicated by an arrow. Viscous walls are chosen both at the top and the bottom of the domain.

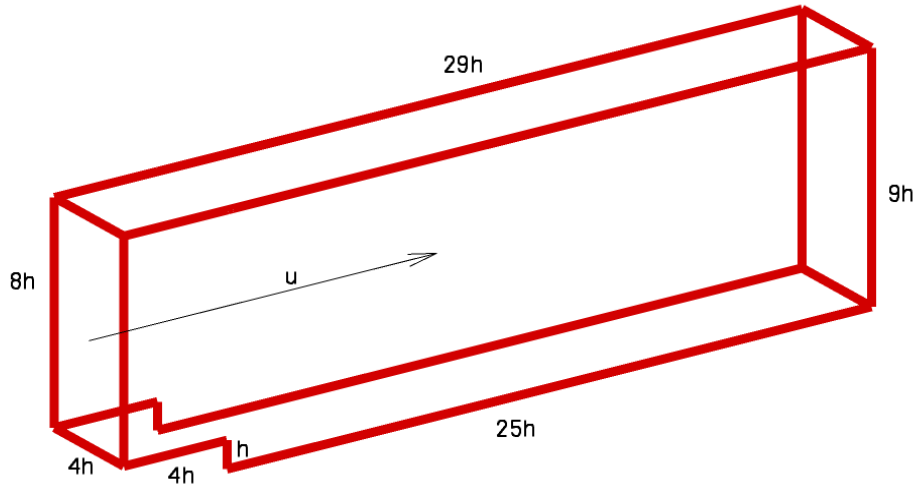


Figure 5.2: Three-dimensional computational domain.

As inflow condition a mean velocity profile based on experimental data is prescribed by using the Dirichlet boundary condition (see Fig. 5.3 (left)). This guarantees a correct boundary layer thickness at the complete upper wall and at the lower wall before the step, which is essential to obtain reliable results in the region below the step. At the outflow the subsonic outflow condition is chosen, where in the unsteady computations the prescribed pressure is taken from the result of previously performed steady computations. Periodic planes are assigned in spanwise direction in the three-dimensional cases.

Results based on three structured grids are presented: While a two-dimensional plane of the coarse grid (which is illustrated in Fig. 5.3 (right)) contains 6617 grid points, the number of grid points of the medium grid is twice that of the coarse grid both in x - and in z -direction, resulting in 25168 points. The fine grid is obtained by again uniformly doubling the number of points both in x - and in z -direction (so an x - z -plane of the fine grid contains 101861 points). All grids are adapted in z -direction both at the upper and the lower wall to resolve the area containing the boundary layer. Starting from the step the grids are also adapted in x -direction

both to the left and the right in order to guarantee a sufficiently fine mesh in the region before and after the step, which is the main region of interest. In spanwise direction, both the coarse and the medium grid contain 33 equidistant planes, while the fine grid is comprised of 65 equidistant planes.

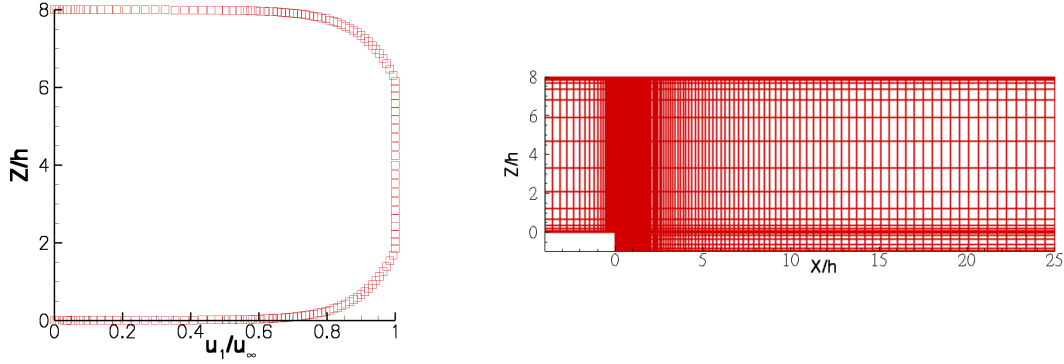


Figure 5.3: Mean velocity inflow profile (left) and x-z-plane of the coarse grid (right).

The Reynolds number Re_h , which is based on the the step height h and the mean inflow velocity $u_\infty = 44.2 \frac{m}{s}$, equals 37500, and a time step size of $\Delta t = 1e - 5s$ is chosen. From similar BFS-computations (see [64]) it could be concluded, that a further time step reduction does not yield any accuracy improvement even on the fine grid. Therefore the chosen time step is regarded as sufficient.

Four computations have been performed: One steady two-dimensional SA-RANS simulation on the medium grid and three SA-DDES computations on all three grids. The SA-DDES computations have been computed for a minimum of 10 convective time units first. Then the flow variables u/u_∞ , C_f and C_p , which are presented in Section 5.2 and the two sensors, which are subject to Section 5.3, have been averaged in time for not less than another 10 convective time units. Finally these variables have also been averaged in homogenous spanwise direction, such that they can be regarded as statistically converged.

5.2 Numerical results

To get a first impression on the basic shape of the flow within this testcase, Figs. (5.4) - (5.7) show the resulting velocity streamlines of the steady two-dimensional SA-RANS computation on the medium grid and the three SA-DDES simulations on the coarse grid, the medium grid and the fine grid, respectively. In the case of the three latter results the streamlines are based on the time-averaged x- and z-velocity. Coming from the left, the flow separates at the step and reattaches further downstream at the lower wall. With the exception of the SA-DDES simulation on the coarse grid, which predicts reattachment too far downstream, the other three results look quite similar and all seem to be in relatively good agreement with the experimentally given reattachment point of $x/h = 6.38$.

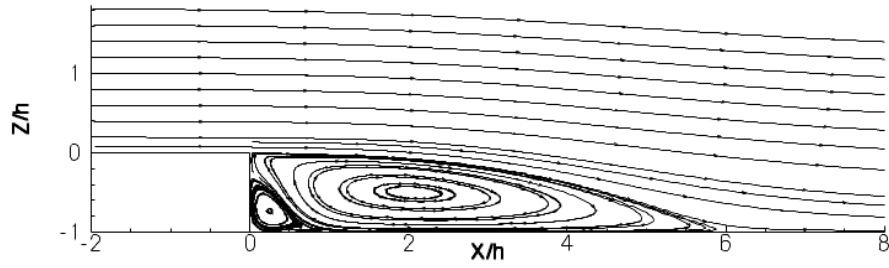


Figure 5.4: Velocity streamlines of the steady two-dimensional SA-RANS computation on the medium grid.

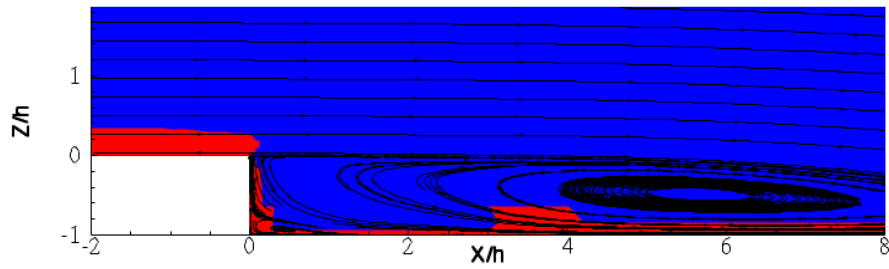


Figure 5.5: RANS-LES distribution and mean velocity streamlines of the SA-DDES computation on the coarse grid. Blue indicates LES-mode and red RANS-mode.

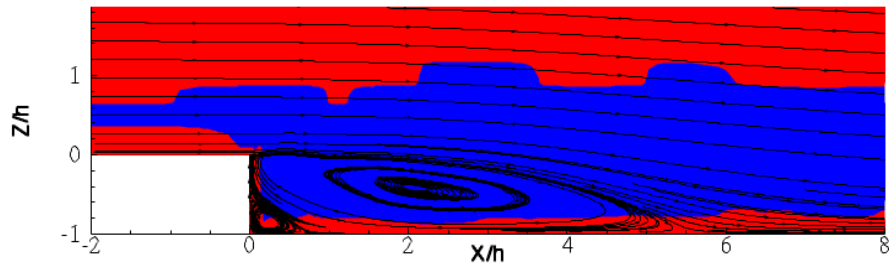


Figure 5.6: RANS-LES distribution and mean velocity streamlines of the SA-DDES computation on the medium grid. Blue indicates LES-mode and red RANS-mode.

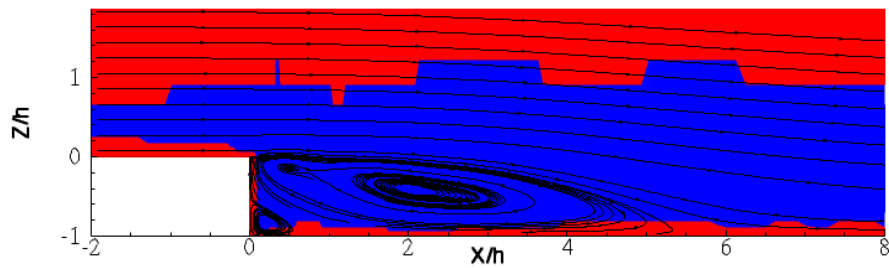


Figure 5.7: RANS-LES distribution and mean velocity streamlines of the SA-DDES computation on the fine grid. Blue indicates LES-mode and red RANS-mode.

In Figs. (5.5) - (5.7) additionally the RANS-LES distributions are illustrated, where red indicates that the model operates in RANS-mode, while blue stands for an LES-like behaviour. In all three figures the model operates as expected: While the boundary layer before the step is computed in RANS-mode, the SA-DDES model switches to LES just behind the step - which is desired, because here the flow separation takes place. Almost the complete area behind the step is then computed in LES-mode - which is proof that the SA-DDES model correctly detects the separation region. The reason why the SA-DDES model switches back to a RANS-like behaviour farther away from the wall in the SA-DDES computations on the medium grid and on the fine grid is due to the usage of the low-Reynolds modification defined in Eq. (3.52).

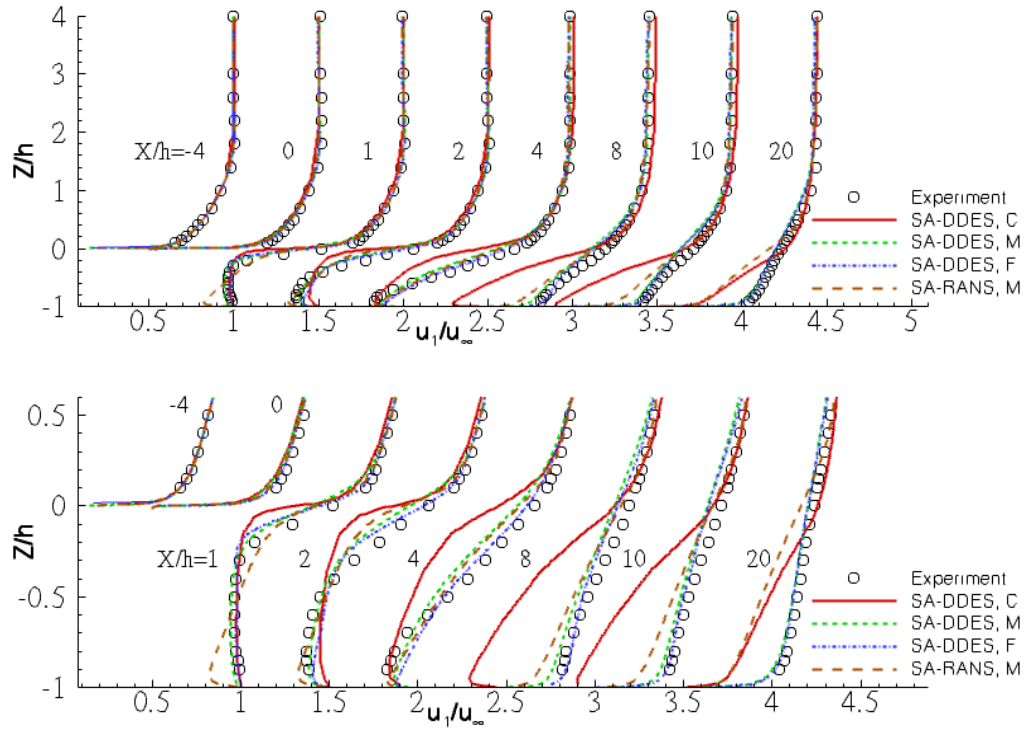


Figure 5.8: Mean velocity profiles of all performed computations at various x -coordinates and comparison with experimental data. The letters C, M, F indicate computations on the coarse, medium and fine grid, respectively. The lower picture shows a zoom on the region below the step.

Next the prediction of the nondimensionalised mean velocity $\overline{u_1}/u_\infty$ in x -direction is considered at various x -coordinates of the computational domain. The letters C, M, F indicate computations on the coarse, medium and fine grid, respectively. Figures 5.8 (upper) and (lower) show the results of all computations in comparison with experimental data with special emphasis on the region below the step in the latter figure. Most obvious is the insufficient velocity prediction of the SA-DDES computation on the coarse grid. This grid is clearly too coarse to enable a realistic flow prediction and can therefore not be recommended for further usage. The SA-

DDES computations on the medium and on the fine grid result in similar outcomes, where the fine SA-DDES computation is slightly superior (this is especially visible at $x/h = 4$ and $x/h = 8$). Nevertheless a slight difference to the experiment is still obvious even in the case of the fine SA-DDES computation. While the steady two-dimensional SA-RANS computation on the medium grid clearly results in more realistic mean velocity profiles than the SA-DDES computation on the coarse grid, it is inferior to the SA-DDES simulation on the medium grid (and hereby also to the one on the fine grid). This is already a hint, that when simulating on the same grid the SA-DDES model is a better choice than SA-RANS within this testcase.

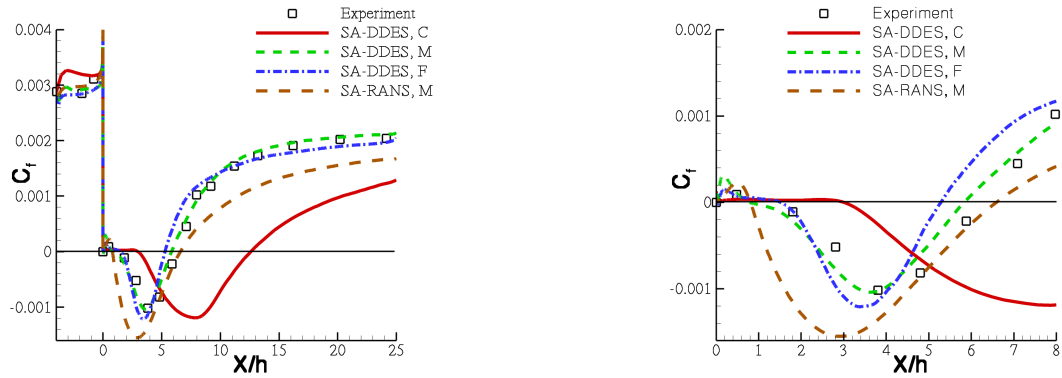


Figure 5.9: Skin friction coefficients of all performed computations and comparison with experimental data. The letters C, M, F indicate computations on the coarse, medium and fine grid, respectively. The right picture shows a zoom on the region below the step.

One of the most important aspects within this testcase is the correct prediction of the reattachment point, which the experiment sets at $x/h = 6.38$. Figures 5.9 (left) and (right) show the skin friction coefficient on the whole domain and in the region below the step, respectively. The SA-DDES computation on the coarse grid predicts reattachment too far downstream, which is due to the fact that this simulation fails to generate enough turbulent content. The SA-RANS model sees reattachment at $x/h = 6.6$, which is a convincing result, although the rest of the c_f -distribution does not match the experiment as good as the SA-DDES simulations on the medium grid and on the fine grid. Comparing the two latter simulations, a rather surprising observation can be made: The SA-DDES computation on the medium grid seems to be superior to the SA-DDES computation on the fine grid. Also the prediction of the reattachment point worsens when regarding the fine computation, which gives reattachment at about $x/h = 5.2$, while in the medium SA-DDES simulation the reattachment point is predicted at about $x/h = 5.9$.

In order to give a possible explanation for this at first glance unexpected supremacy of the medium SA-DDES computation over the fine SA-DDES simulation, the pressure distribution is illustrated in Fig. 5.10. Here one can see that the SA-DDES computation on the medium grid underperforms compared to the SA-DDES simulation on the fine grid: Firstly in the region starting from the step ($x/h = 0$) to

about $x/h = 4$ the medium SA-DDES computation differs vastly from the experiment and secondly oscillations in the area ranging from $x/h = 9$ to $x/h = 25$ are visible. The unexpectedly good result of SA-DDES on the medium grid regarding c_f should therefore be regarded with caution -the good prediction of the reattachment point might be due to the wrong reasons, namely differences to the experiment just behind the step. With regard to the pressure distributions of the other two computations shown in Fig. 5.10, one can see that the SA-DDES computation on the coarse grid once more differs immensely compared to the experiment, while the SA-RANS computation here surprisingly seems to be even slightly superior to SA-DDES on the fine grid.

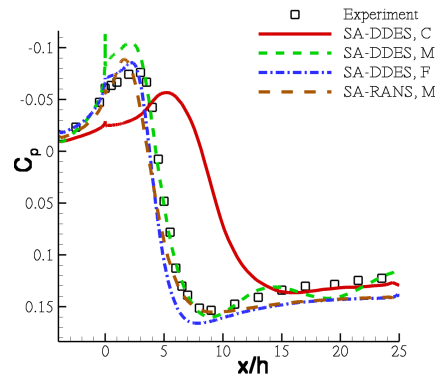


Figure 5.10: Pressure coefficients of all performed computations and comparison with experimental data. The letters C, M, F indicate computations on the coarse, medium and fine grid, respectively.

The last variable to look at is the instantaneous Q-criterion, which is an indicator for turbulent structures. In Figs. 5.11 (upper), (middle) and (lower) isosurfaces of the SA-DDES computations at a value of $Q = 200\frac{1}{s^2}$ on the respectively coarse grid, medium grid and fine grid are plotted. Figure 5.11 (upper) again reveals that the coarse grid is too coarse to resolve turbulence: Almost no three-dimensional structures can be seen and the resulting two-dimensional rolls are similar to the typical outcome of a three-dimensional SA-RANS computation. In the case of the SA-DDES computations on the medium grid and on the fine grid clearly three-dimensional structures can be observed, while the scales seem to be smaller in the fine SA-DDES computation. This outcome is hereby as expected: The finer the grid, the smaller the resolvable turbulent structures.

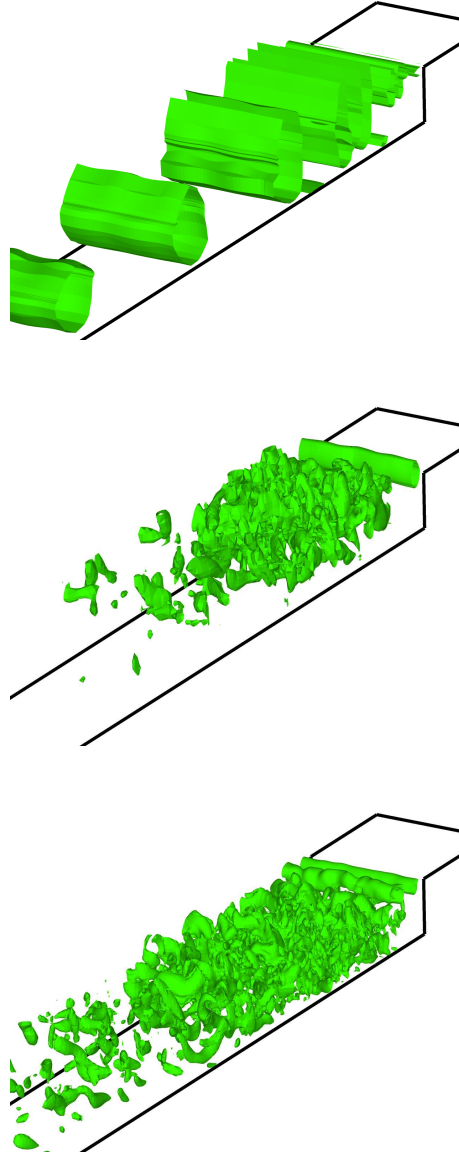


Figure 5.11: Isosurfaces of the instantaneous Q-criterion at a value of $Q = 200\frac{1}{s^2}$. Shown are the results of SA-DDES on the coarse grid (upper), on the medium grid (middle) and on the fine grid (lower).

Overall the results of the backward facing step computations can be summarized as follows: Despite some numerical instabilities observed in the pressure distribution of the medium SA-DDES computation and a reattachment prediction too far upstream of the fine SA-DDES simulation, both computations agree comparably well with the experimental data. This is expected, because the SA-DDES model (and other DES models likewise) has originally been designed for testcases with massive flow separation. The SA-DDES computation on the coarse grid performs less accurately, because not enough turbulent content is produced, resulting in too late reattachment and overall large differences to the experimental data. The SA-

RANS model predicts reattachment comparably well, while it is overall inferior to the SA-DDES computation on the medium grid and especially to the SA-DDES simulation on the fine grid. As the fine SA-DDES computation additionally results in the greatest amount of turbulent small-scale structures, this computation overall can be regarded as the most reliable and accurate one.

5.3 Examination of two sensors

Additionally to the results presented in Section 5.2 two sensors have been implemented and examined within this testcase. Both sensors have been averaged in time and also in homogenous spanwise direction at the end of the computations to guarantee statistically converged mean values.

The aim of the first sensor, which is the von Kármán length scale L_{vk} , is to detect small-scale turbulent structures. The second sensor *Indicator* can be used to guarantee a sufficiently high grid refinement in LES-regions.

5.3.1 First sensor: The von Kármán length scale

Based on Rotta's equation for the integral length scale [95], Menter and Egorov deduce a formulae for the von Kármán length scale [75]. They then use L_{vk} to modify the Menter-SST URANS model, which results in the so-called SST-based Scale-Adaptive Simulation (SST-SAS) model [76]. In contrast to the original Menter-SST model the SST-SAS model is capable of adjusting to resolvable small-scale structures, resulting in an LES-like behaviour in unsteady flow regions.

Unless higher order methods are used, the resolvable structures within a computation cannot be smaller than the grid cell sizes. Therefore L_{vk} should scale with a certain grid cell size Δ in regions with small turbulent structures, namely flow separation regions. As the backward facing step includes a vast region of separated flow, this testcase is considered to be ideal with regard to test the performance of L_{vk} . L_{vk} is given by:

$$L_{vK} = \kappa \frac{\mathbb{S}}{\mathbb{U}''}, \quad \kappa = 0.41 \quad (5.1)$$

$$\mathbb{S} = \sqrt{2S_{ij}S_{ij}}, \quad \mathbb{U}'' = \sqrt{\sum_{i=1}^d \sum_{j,k=1}^d \frac{\partial^2 u_i}{\partial^2 x_j} \frac{\partial^2 u_i}{\partial^2 x_k}} \quad (5.2)$$

where the strain-rate tensor S_{ij} has been defined in Eq. (2.5) and $d=1,2,3$ denotes the spatial dimension.

Figures 5.12 (upper), (middle) and (lower) show L_{vk} of the SA-DDES computations on the relatively coarse, medium and fine grid, where especially the region below the step is significant, because here flow separation and small-scale structures occur. L_{vk} behaves as desired: The finer the grid the smaller the structures and L_{vk} become. It hereby can be stated that L_{vk} truly is capable of detecting small-scale structures.

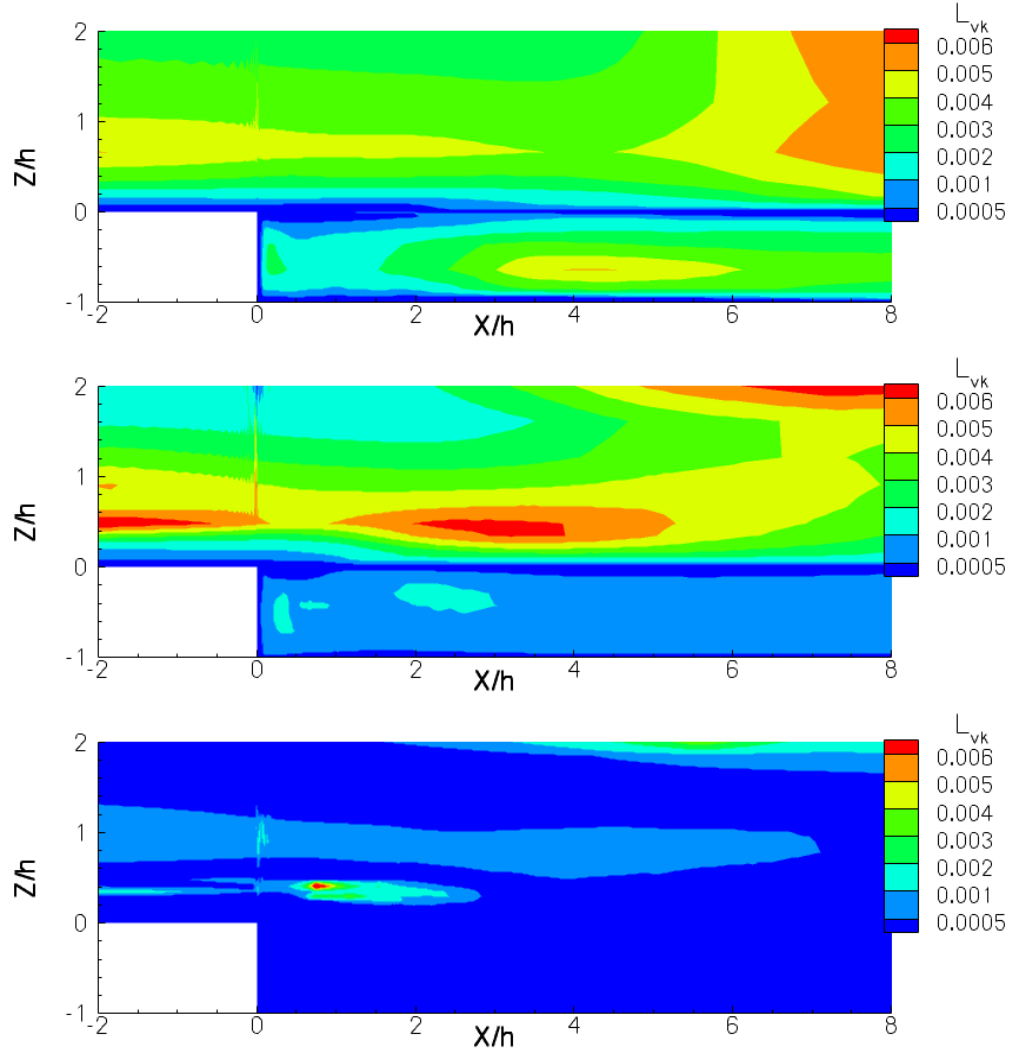


Figure 5.12: Von Kármán length scale L_{vk} of the SA-DDES computations on the coarse grid (upper), medium grid (middle) and fine grid (lower).

In order to use L_{vk} as an enhancement of the SA-DDES model (by manually switching to LES-mode in regions with massive flow separation), an adequate scaling of L_{vk} with a grid parameter Δ is required. Ideally $\frac{L_{vk}}{\Delta}$ should look similar for a set of different grids with varying fineness.

By dx, dy, dz we denote the grid cell sizes in respectively x -, y -, and z - direction. In [69] it was shown that $\Delta := (dx dz)^{\frac{1}{2}}$ leads to similar results of $\frac{L_{vk}}{\Delta}$ for grids of different refinement in the case of isotropic grids, where the testcase considered was an LES-simulation of a turbulent channel flow.

Unfortunately the grids used within the backward facing step computations are highly anisotropic in large parts of the domain (see Fig. 5.3 (right)). $\Delta = (dx dz)^{\frac{1}{2}}$ therefore does not seem to be the correct scaling parameter. This is indicated in Figs. 5.13 (upper), (middle) and (lower), which show $\frac{L_{vk}}{(dx dz)^{\frac{1}{2}}}$ on the relatively coarse, medium and fine grid. Clearly, the values of the three computations vary immensely.

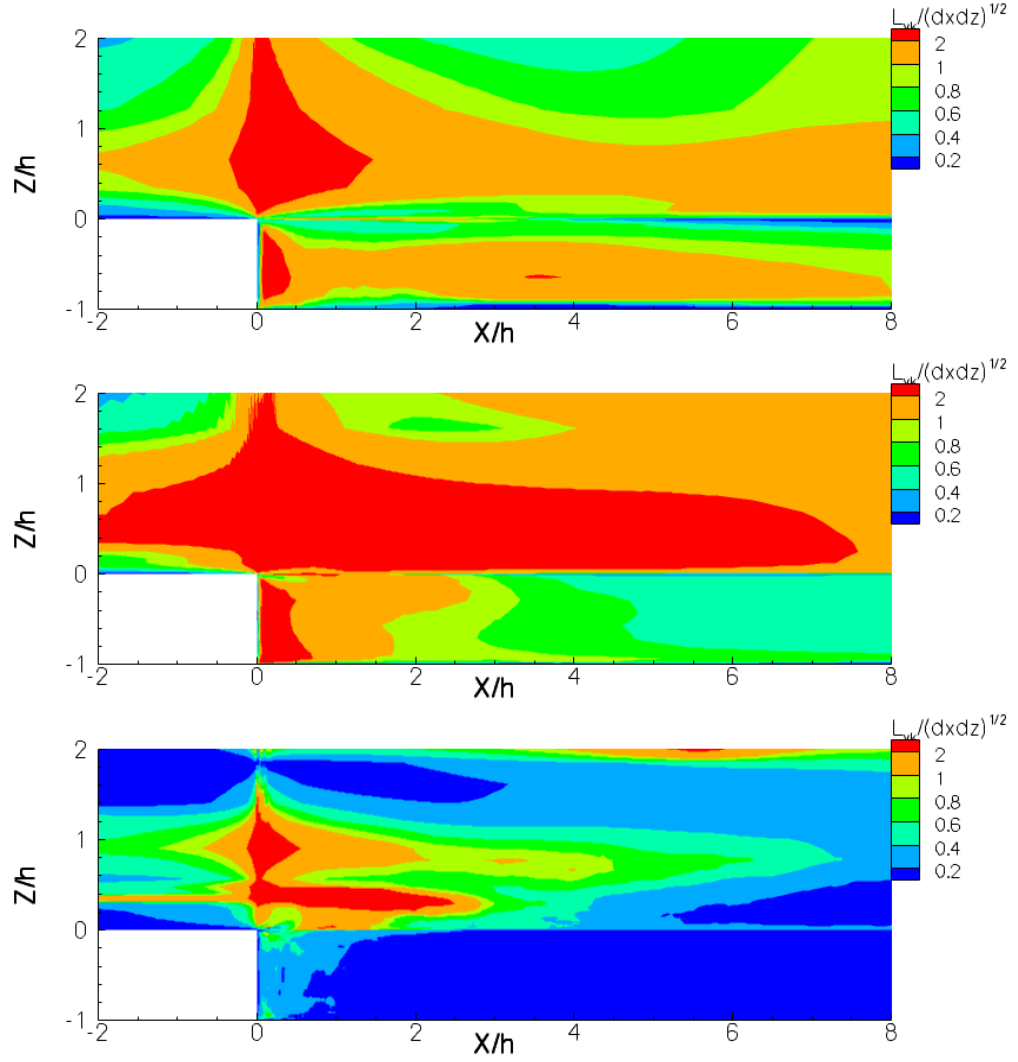


Figure 5.13: $\frac{L_{vk}}{(dx dz)^{\frac{1}{2}}}$ of the SA-DDES computations on the coarse grid (upper), medium grid (middle) and fine grid (lower).

Another approach is to choose $\Delta := (Vol)^{\frac{1}{3}}$, where Vol denotes the volume of a grid cell given by $Vol = dx \cdot dy \cdot dz$. As one can see in Figs. 5.14 (upper), (middle) and (lower) also in this case $\frac{L_{vk}}{\Delta}$ does not look similar below the step when comparing the different computations.

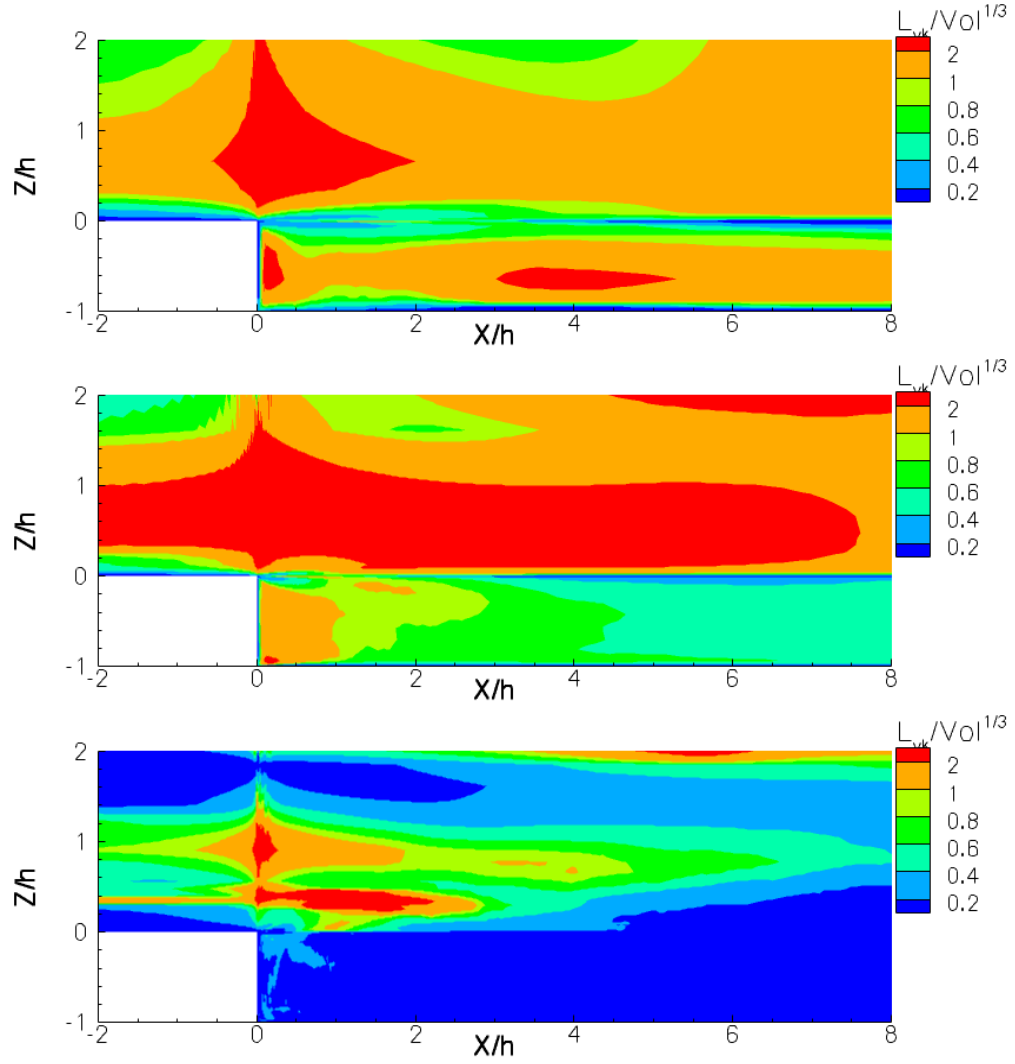


Figure 5.14: $\frac{L_{vk}}{Vol^{1/3}}$ of the SA-DDES computations on the coarse grid (upper), medium grid (middle) and fine grid (lower).

As a summary regarding the reliability of the first sensor L_{vk} it can be stated that this sensor truly is able to detect small-scale turbulent structures. Unfortunately no scaling parameter Δ could be identified, such that the computations on the different grids result in comparable outcomes of $\frac{L_{vk}}{\Delta}$. Although it can not be ruled out that such a parameter Δ exists, it seems unlikely due to the use of strongly stretched grid cells: If dx , dy and dz vary immensely within a region it is by no means straightforward to decide which size of a turbulent structure can be resolved. It therefore is also unclear how small L_{vk} becomes. As the answers regarding these issues are also highly testcase dependent, **it cannot be expected to find a universal Δ such that $\frac{L_{vk}}{\Delta}$ looks similar for computations on differently refined anisotropic grids for a vast set of testcases.**

This leads to two possible solutions when one wants to utilize L_{vk} : Either mainly isotropic grids are used or one has to find an appropriate scaling parameter Δ

independently for each testcase. As it has been shown, finding such a Δ can be hard even when restricting to a single testcase.

5.3.2 Second sensor: *Indicator*

Another major issue when performing LES or DES computations lies in guaranteeing a sufficiently fine grid within the LES-part. This is not only of major importance in order to resolve small-scale structures but also with regard to the problem of so-called modelled stress depletion, which can occur due to the use of a too coarse grid, leading to unphysical flow separation. The phenomena of modelled stress depletion and grid induced separation are explained and examined in detail in Section 6.2.

To overcome the just mentioned flaws, a sensor *Indicator* was proposed by Knopp et al. [64], which estimates the amount of resolved turbulent kinetic energy relative to the overall kinetic energy. If this value is sufficiently high, the grid is considered fine enough to allow for a fully resolved LES-computation in that region. If not, the grid has to be refined there. *Indicator* is defined by:

$$\text{Indicator} = \frac{k_{res}}{k_{res} + k_{sgs}} \quad (5.3)$$

where

$$k_{res} = 0.5 \cdot \overline{(u_i - \bar{u}_i)^2}, \quad k_{sgs} = 0.5 \cdot \overline{(u_i - u_i^\Delta)^2}$$

$$u_i^\Delta(x, t) = \int_{\mathbb{R}} g_\Delta(x - y) u_i(y, t) dy$$

Here \bar{f} again denotes the average of f in time and g_Δ is the top hat filter function. Due to its construction, *Indicator* can only take values within $[0, 1]$.

In [64] it has been proposed that *Indicator* > 0.9 suggests a sufficiently fine grid, while *Indicator* < 0.8 indicates that the grid is too coarse. Values between 0.8 and 0.9 do not allow an unambiguous decision regarding the grid quality.

Figures 5.15 (upper), (middle) and (lower) show that *Indicator* behaves grid convergent, i.e. the finer the grid the higher the values of *Indicator*. One can especially see that *Indicator* highlights the coarse grid as insufficient for an LES-computation in the region below the step - an assumption that is clearly supported by the results presented in Section 5.2. While the medium grid is regarded as adequate in the area just behind the step, *Indicator* shows that it needs refinement farther downstream. The fine grid seems to be sufficiently fine everywhere. Overall the behaviour of *Indicator* lies in very good agreement with the results shown in Section 5.2. Nevertheless *Indicator* should still be tested for different testcases to evaluate its reliability in a more general context.

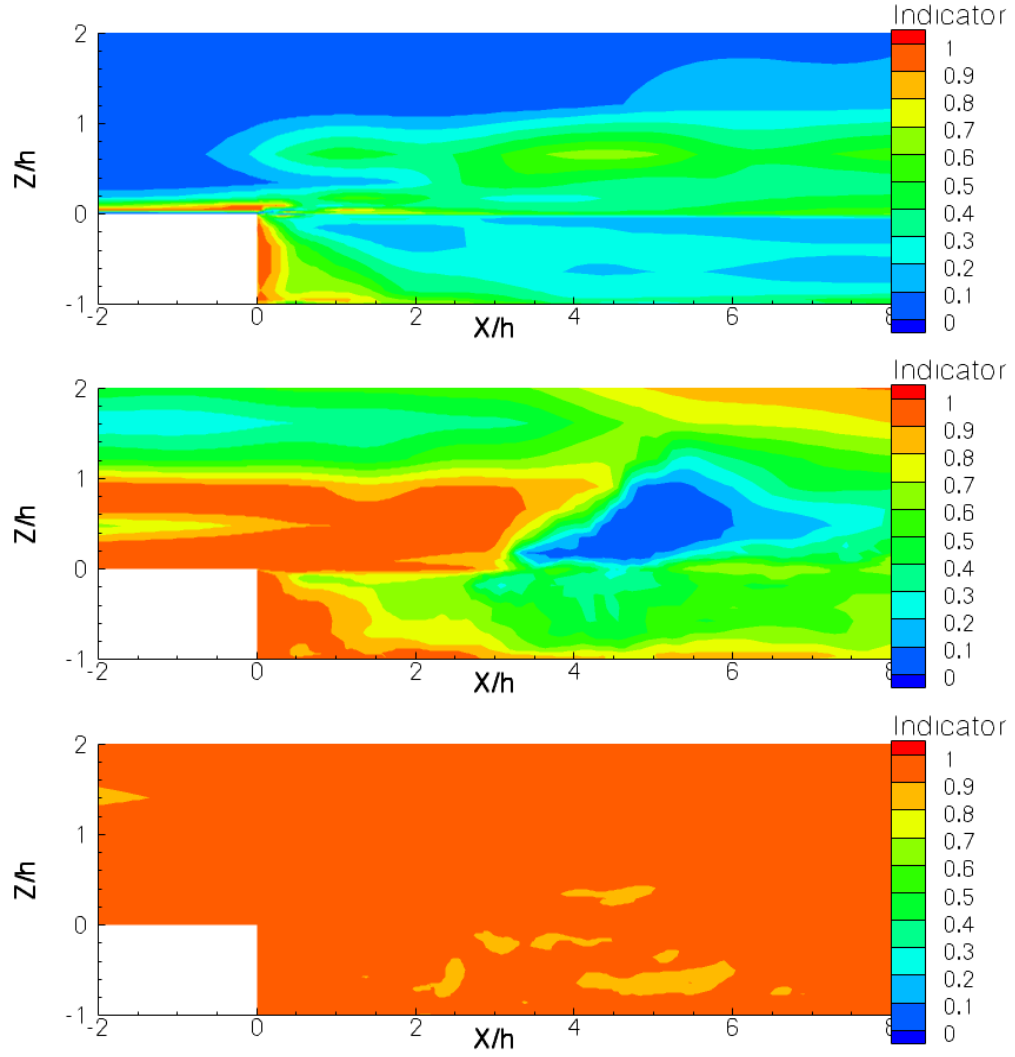


Figure 5.15: Sensor *Indicator* of the SA-DDES computations on the coarse grid (upper), medium grid (middle) and fine grid (lower).

Summary of the chapter and implications on the simulation method

At the end of this and of the next two chapters we want to give a short summary of the relatively results, where especially the implications on the applications of the complete simulation method in Chapter 8 shall be investigated.

It has been shown that the use of the SA-DDES model leads to reliable results with regard to mean velocity profiles, prediction of reattachment point and development of three-dimensional turbulent structures, if the grid is fine enough. As expected, SA-DDES is superior to SA-RANS because of the large separation region. While the first sensor L_{vk} shows an overall reasonable behaviour, its practical use is limited because no correct grid scaling parameter Δ could be identified. The second sensor *Indicator* behaves as expected and even grid convergence could be achieved.

Based on the results in this chapter, the use of the SA-DDES turbulence model can be strongly recommended for the applications in Chapter 8.

Chapter 6

An HGR01 airfoil at stall

As the testcase considered in Chapter 5 includes a vast region of massively separated ("detached") flow, the SA-DDES model performed satisfactory. This could be expected, because DES models have been designed for these kinds of flow scenarios. In the case of mainly attached flow around a stalling airfoil, the performance of SA-DDES is not so straightforward to predict. As in the final numerical example shown in Section 8.4 SA-DDES is applied to the flow around a two-element airfoil at stall, the stalling behaviour of an HGR01 airfoil is subject of this chapter. For this testcase the separation point is not determined by the geometry. As one can see in Figs. 5.1 (left) and (right), where the velocity streamlines of the flow around the HGR01 airfoil and alternatively around the FNG airfoil at undisturbed onflow conditions is shown, the separation region in the latter example is larger than in the former testcase. Nevertheless, both testcases can be considered as similar in large parts of the domain, because of the mainly attached flows at the relatively main wings. Both results are based on steady two-dimensional SA-RANS simulations, where the angle of attack of the HGR01 airfoil is $\alpha = 14^\circ$, while the incidence angle of the FNG-airfoil is $\alpha = 10^\circ$. In both cases the airfoils are close to stall.

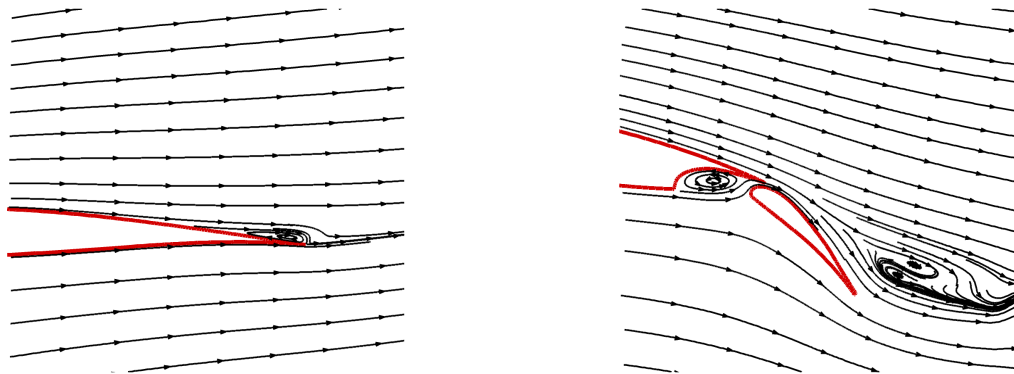


Figure 6.1: Comparison of velocity streamlines obtained by steady two-dimensional SA-RANS simulations of flow around HGR01 airfoil at $\alpha = 14^\circ$ (left) and of flow around two-element FNG airfoil at $\alpha = 10^\circ$ (right).

6.1 Description of the testcase

The HGR01 is a research airfoil which has been designed to investigate the stall process of a typical tail plane airfoil at relatively low Reynolds numbers. It has been experimentally investigated at $Re = 0.65 \cdot 10^6$ and $Ma = 0.073$ in the low-speed wind tunnel of the Institute of Fluid Mechanics, TU Braunschweig, using static pressure probes, oil visualization and PIV measurement techniques [130]. Although the stall is of mixed type, i.e. a combination of laminar leading-edge separation and turbulent separation from the trailing edge, the experiments indicate that the flow is dominated by the trailing edge separation up to angles of attack clearly beyond maximum lift ($\alpha = 16^\circ$), which is the range of interest within this simulation.

In accordance with the experiments, the SA-DDES simulations are performed at $Re = 0.65 \cdot 10^6$ and $Ma = 0.073$ and cover the stall process from almost fully attached flow at $\alpha = 12^\circ$ up to clear lift breakdown induced by trailing edge separation at $\alpha = 16^\circ$. The computational grid for the three-dimensional computations (which is called "original grid" from this point on) is based on a two-dimensional hybrid RANS grid obtained from a mesh convergence study [130]. To meet the requirements of a DES it has been further refined in the expected separation region above the trailing-edge and the wake, yielding 650×112 points in the near-wall structured part of the x-z plane, where x denotes the streamwise variable, y is associated with the spanwise direction and z points in wallnormal direction.

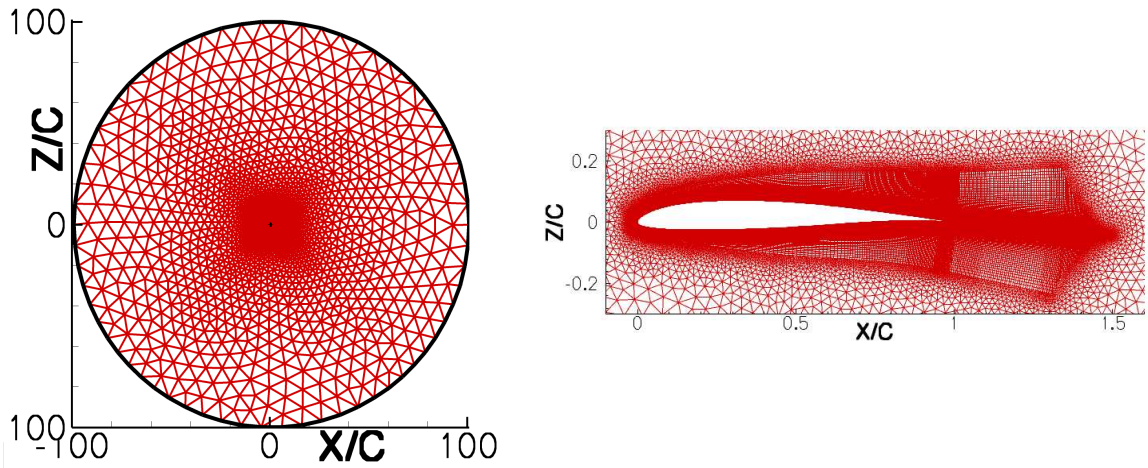


Figure 6.2: Plot of the computational domain and the hybrid grid (which is also called "original grid") used in the HGR01 simulation. The right picture shows a zoom at the center, where the airfoil is located.

This two-dimensional grid, which is shown in Figs. 6.2 (left) and (right), is uniformly extruded in spanwise direction with an extent of 20 % chord length using 33 two-dimensional planes, resulting in $3.5 \cdot 10^6$ grid points. In spanwise direction periodic planes are applied and at the airfoil viscous walls are chosen. In- and outflow is modeled by the farfield boundary condition in a distance of 100 chord

lengths away from the airfoil, thus neglecting the effects of wind tunnel walls and installations throughout this study. At the beginning of the computation a steady three-dimensional SA-RANS simulation at $\alpha = 12^\circ$ is performed, which is then used as restart solution for the following SA-DDES computations. After having reached a convergent behaviour of the aerodynamic coefficients c_l , c_d and c_m , the grid is rotated to increase the angle of attack by 1° . This procedure is performed repeatedly until the maximal angle of $\alpha = 16^\circ$ has been reached. The chosen physical timestep $\Delta t = 1 \cdot 10^{-4} s$ within the dual-timestepping method is based on the expected maximum velocity and the target grid resolution in the LES region according to [106]. The results are compared to SA-RANS computations and experimental data.

6.2 An examination of the reliability of SA-DDES at $\alpha = 12^\circ$ and necessary model modifications

Since large parts of the boundary layer are expected to remain attached even at higher angles of attack, it is advisable to verify, if the problem of modelled stress depletion and grid-induced separation is actually cured by the use of SA-DDES. Modelled stress depletion can be observed frequently in SA-DES computations and occurs when the underlying grid is fine enough too force the model into LES-mode in large parts of the boundary layer but still is too coarse to enable a fully resolved LES (this is also called "underresolved" LES). As a result the eddy viscosity and hereby also the Reynolds stresses are depleted in this part of the boundary layer: The modelled part of them is lower compared to computing in RANS-mode and the resolved part is too small to compensate since the grid is too coarse. The vanishing of the Reynolds stresses then leads to unphysical grid induced flow separation.

An appropriate test case is the flow at $\alpha = 12^\circ$, since the trailing edge separation computed with SA-RANS is so small here, that it is safe to expect the LES region in SA-based DDES computations to be limited to the wake region. Therefore, the results obtained with SA-DDES should be very similar to SA-RANS along the airfoil.

6.2.1 f_d functions and RANS-LES distributions

Recall, that the RANS-LES distribution in SA-DDES is based upon the function f_d , where $f_d \equiv 0$ results in RANS-mode, while $f_d \equiv 1$ forces the model to operate in LES-mode. f_d has been defined in Eq. (3.51):

$$f_d = 1 - \tanh(8r_d^3), \quad r_d = \frac{\nu + \nu_t}{S_{ij}\kappa^2 d^2}, \quad \kappa = 0.41$$

f_d function of SA-RANS computation

As it is used as restart for the subsequent SA-DDES computations, the SA-RANS solution is taken a closer look at first. Although f_d is not used within the SA-RANS simulation, it still can be computed and analysed. This allows to determine

a priori, if the complete boundary layer will stay in RANS-mode when switching to SA-DDES.

Figure 6.3 (left) and Fig. 6.4 (left) show f_d of the SA-RANS simulation in the middle of the upper surface and at the trailing edge, respectively. For comparison, Fig. 6.3 (right) and Fig. 6.4 (right) illustrate the boundary layer thickness indicated by the Mach number (recall that the freestream Mach number equals 0.073). In the last picture mentioned, the approximate boundary layer thickness is additionally indicated by a black line. One can clearly see, that f_d already becomes 1 well inside the boundary layer, which is a hint that problems may arise when switching to SA-DDES.

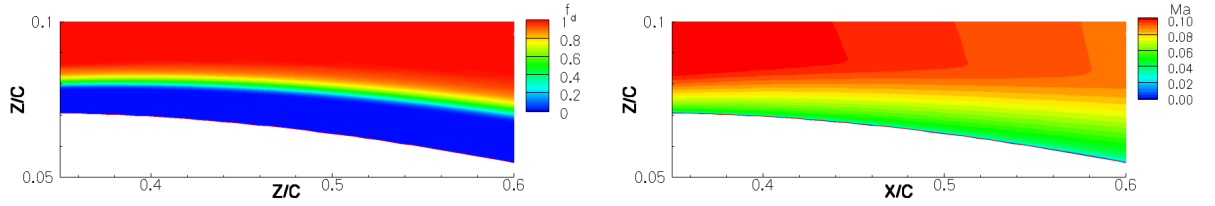


Figure 6.3: Plot of f_d based on the SA-RANS computation on the original grid (left) and boundary layer thickness indicated by Mach number (right) in the middle of the upper surface.

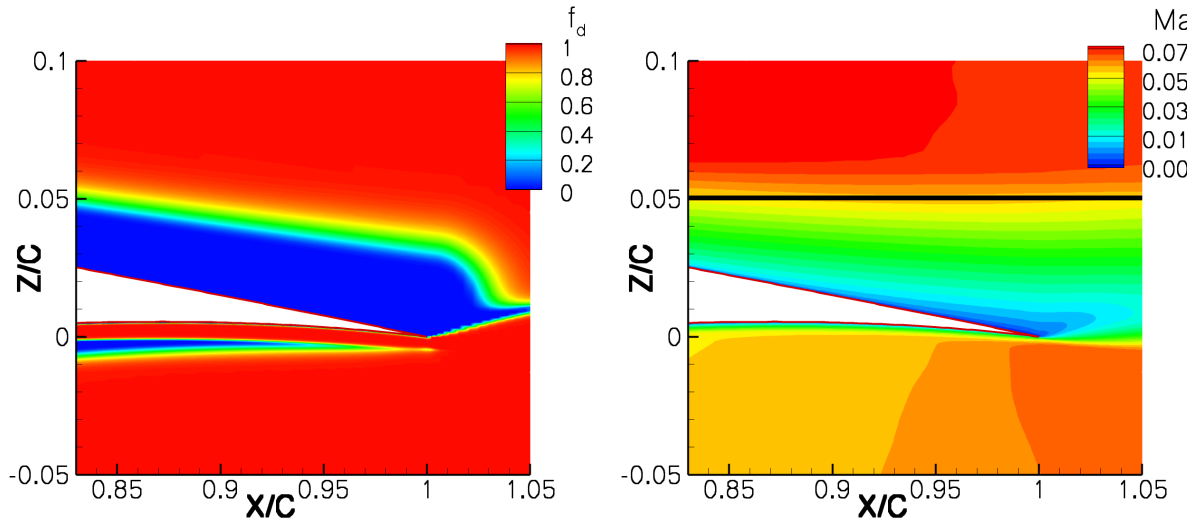


Figure 6.4: Plot of f_d based on the SA-RANS computation on the original grid (left) and boundary layer thickness indicated by Mach number (right) at the trailing edge.

To further study the reliability of f_d , the velocity profiles of u_1/u_∞ (indicating the boundary layer thickness), the eddy viscosity ratio ν_t/ν and f_d are plotted in Fig. 6.5, where the profiles are taken in wall-normal direction at $x/c = 0.9$ on the upper airfoil surface. Again, it becomes obvious that the LES-part (where $f_d \equiv 1$) will intrude deeply into the boundary layer when switching to SA-DDES. As expected

the values of ν_t/ν coincide strongly with the boundary layer and vanish outside of it.

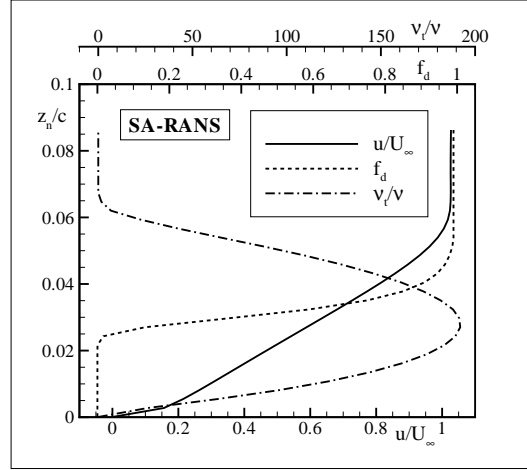


Figure 6.5: Profiles in wall-normal direction of u_1/u_∞ , ν_t/ν and f_d of the SA-RANS computation at $x/c = 0.9$ on the upper surface.

f_d functions of SA-DDES and SA-DDES₁₆ computations

When now activating the SA-DDES model, f_d should take a value of zero in the complete attached boundary layer. Instead, only about 50 % of the strongly decelerated boundary layer is covered, apparently not enough to retain the required eddy viscosity level to preserve the boundary layer shape. On the other hand, the grid in the outer boundary-layer region is too coarse to possibly compensate the lacking Reynolds-stress by directly resolved turbulent fluctuations.

Moreover, as f_d itself depends on ν_t , the “shielded” part of the boundary layer that is computed in RANS-mode becomes smaller and smaller with reduced ν_t , which hereby reduces ν_t even further. A stable solution with fixed RANS-LES distribution therefore cannot be reached even after a very long computational time. Figure 6.6 (left) shows the result after the simulation has been performed for three convective time units in SA-DDES mode, where the SA-RANS solution has been taken as restart. Due to the dysfunctional behaviour of f_d , ν_t/ν already has been vastly diminished and the shape of the boundary layer also has changed compared to Fig. 6.5.

As these observations are consistent with the concept of “ambiguous” grids, which lead to modelled stress depletion in SA-DES, it is clearly illustrated that the counter-measures introduced with SA-DDES are not sufficient with regard to this testcase. In particular, the empirical factor 8 in Eq. (3.50), which mainly governs the extent of f_d inside the boundary layer, is derived from a calibration based on the flat-plate flow [109]. It therefore seems not too surprising that the original formulation partly fails its purpose in flows, which strongly deviate from a constant pressure flow, such as the adverse-pressure boundary layer on the HGR01 airfoil at stall.

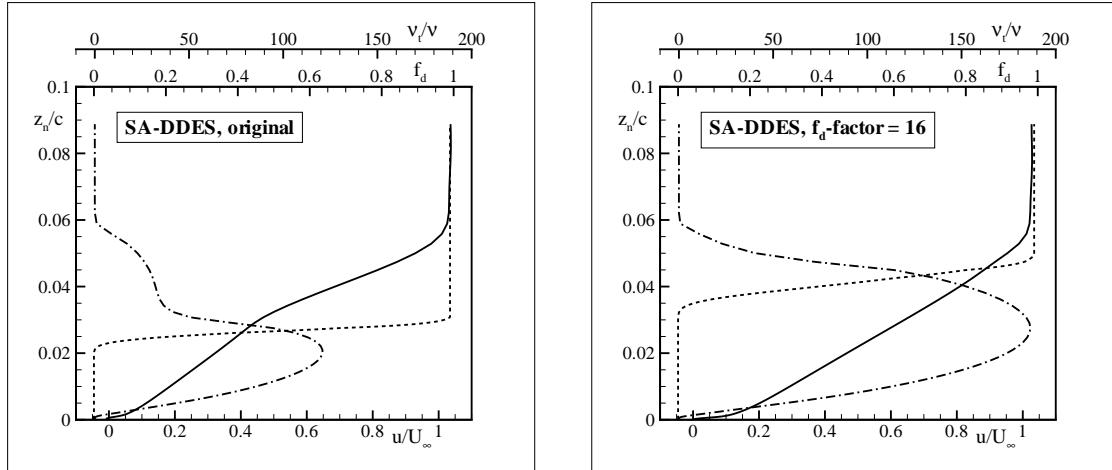


Figure 6.6: Profiles in wall-normal direction of u_1/u_∞ , ν_t/ν and f_d of the SA-DDES computation (left) and of the SA-DDES₁₆ computation (right) at $x/c = 0.9$ on the upper surface. Both simulations were started with the same SA-RANS restart and have then been further computed for three convective time units.

Since applying a grid coarse enough to avoid modelled stress depletion by itself is not considered an option, a modification of f_d seems inevitable. It is found that raising the factor 8 to 16 is necessary to shield enough of the boundary layer to obtain a solution close to SA-RANS at $\alpha = 12^\circ$. This modified version of SA-DDES will be referred to as "SA-DDES₁₆" henceforth and has been confirmed by application to higher angles of attack, which are subject of the next section. Note, that the standard SA-DDES model (i. e. a factor of 8 is used in the f_d -formulation) is unchanged called "SA-DDES" and not "SA-DDES₈".

Figure 6.6 (right) illustrates the results obtained by SA-DDES₁₆ after three convective time units. The range, where $f_d \equiv 0$ is notably increased compared to the left picture and now covers most of the boundary layer. As a result, the conservation of ν_t/ν is comparably good as in the SA-RANS computation. Also the shape of the boundary layer seems not to be affected by the use of SA-DDES₁₆.

Effect of grid modification on f_d

Although the first result presented by using SA-DDES₁₆ seems to be promising, the increase of the factor from 8 to 16 in the f_d -formulation is rather an ad-hoc fix than a general solution to the failure of SA-DDES. Therefore also another approach has been tested, which lies in using an alternative grid. This grid will be called "modified grid" from now on.

The idea of using a second grid evolved from the observation, that f_d is also based on the velocity gradients. An accurate gradient approximation is numerically a difficult task, especially when performed on unstructured grids. A modified grid could allow an improved gradient approximation, which might then result in a correct behaviour of f_d and hereby leading to RANS-mode within the complete boundary layer.

Based on the original grid, the following modifications have been performed:

In the structured part the number of wall-normal lines was increased firstly, resulting in more quadratic cells. Secondly, the transition from the structured part to the unstructured part – which is a highly sensitive area – was improved. Thirdly, the unstructured part was globally refined. All of the performed modifications were the result of discussion with experts of grid generation and of SA-DDES within the DLR- $C^2A^2S^2E$ -department and also the final modified grid was approved by them. Figure 6.7 shows a cutout of the original grid (left) and of the modified grid (right).

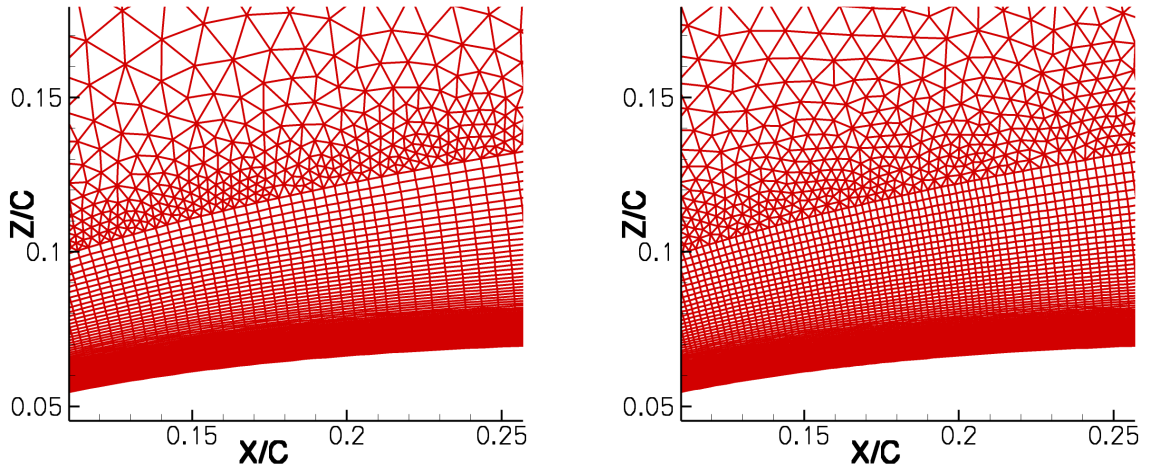


Figure 6.7: Cutout of the original grid and of the modified grid.

To save computational time, all performed computations on the modified grid were performed two-dimensionally. Again, first a steady SA-RANS simulation was obtained that was then used as restart for the forthcoming SA-DDES computations, which were performed for another three convective time units. Although operating the SA-DDES model two-dimensionally does not make sense from a physical point of view - turbulence is always three-dimensional - the results still allow to decide, if the change in grids improves the shape of f_d .

To examine the impact of the two modifications closer, Fig. 6.8 (left) and Fig. 6.9 (left) show f_d resulting from SA-DDES₁₆ on the original grid, while Fig. 6.8 (right) and Fig. 6.9 (right) are based on the two-dimensional SA-DDES computation on the modified grid. While the area, where $f_d \equiv 0$ is clearly thicker and covers most of the boundary layer (see Fig. 6.3 (right) and Fig. 6.4 (right)) in the SA-DDES₁₆ simulation, f_d based on the two-dimensional SA-DDES computation on the modified grid does not seem to be changed compared to Fig. 6.3 (left) and Fig. 6.4 (left). It therefore can already be concluded at this point, that the grid modification does not solve the problems caused by the use of SA-DDES.

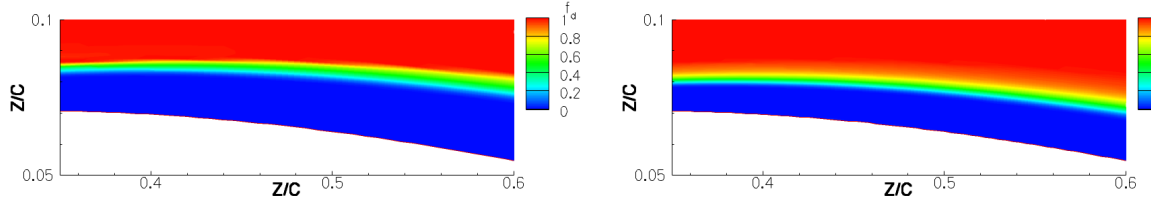


Figure 6.8: Plot of f_d of the SA-DDES₁₆ computation on the original grid (left) and of the two-dimensional SA-DDES simulation on the modified grid (right) in the middle of the upper surface. Both computations were started with an SA-RANS restart and have then been further computed for three convective time units.

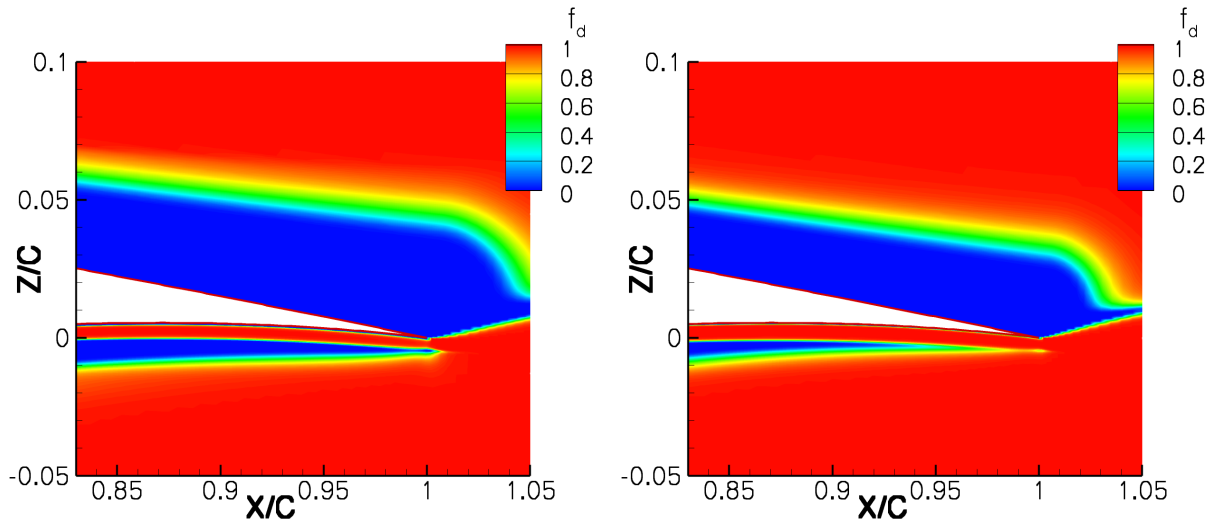


Figure 6.9: Plot of f_d of the SA-DDES₁₆ computation on the original grid (left) and of the two-dimensional SA-DDES simulation on the modified grid (right) at the trailing edge. Both computations were started with an SA-RANS restart and have then been further computed for three convective time units.

RANS-LES distributions of various computations

Figures 6.10 (left) and (right) and Fig. 6.11 show the resulting RANS-LES distributions and velocity streamlines of the SA-DDES computation on the original grid, the SA-DDES₁₆ solution on the original grid and the SA-DDES computation on the modified grid, respectively, in the middle of the upper surface. All of these computations were started with an SA-RANS restart and have then been further computed for another three convective time units. While the change in grids does not yield any improvement, the use of SA-DDES₁₆ clearly enlarges the RANS-part near the airfoil. In each case the reason for the model switching back into RANS-mode farther away from the airfoil (which can be seen at the relative upper part of the figures) is due to the use of the low-Reynolds modification.

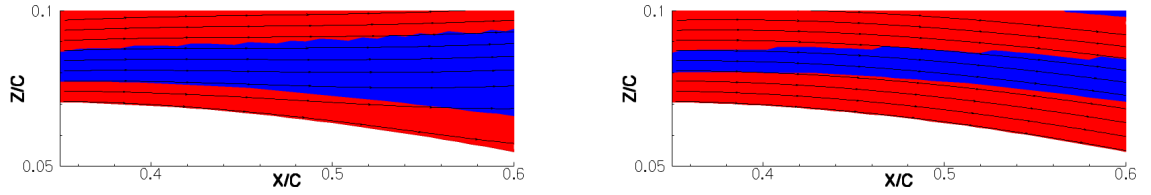


Figure 6.10: RANS-LES distribution and streamlines of SA-DDES (left) and SA-DDES₁₆ (right) on the original grid in the middle of the upper surface. Both computations were started with an SA-RANS restart and have then been further computed for three convective time units.

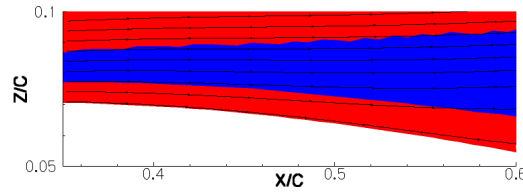


Figure 6.11: RANS-LES distribution and streamlines of the two-dimensional SA-DDES simulation on the modified grid in the middle of the upper surface. The computation was started with an SA-RANS restart and has then been further computed for three convective time units.

In Figs. 6.12–6.13 the RANS-LES distributions and streamlines of the same computations as in Figs. 6.10–6.11 can be seen, where the focus now lies on the trailing edge. Again, Fig. 6.12 (left) and Fig. 6.13 (left) look almost identical, which further provides evidence that the modified grid does not improve matters. Although the flow separation is still moderate in both computations, it already has visibly grown larger than the one resulting from the SA-DDES₁₆ computation on the original grid, where the larger RANS-part prevents grid induced separation. Figure 6.13 (right) shows that the separation even grows much larger when the SA-DDES computation on the original grid is performed for 8 convective time units. Here one can also see that the turbulent structures on the upper surface already have vastly diminished the RANS-part, while ν_t on the lower surface has become so small, that the low-Reynolds modification is active, resulting in a thickening of the RANS-part there.

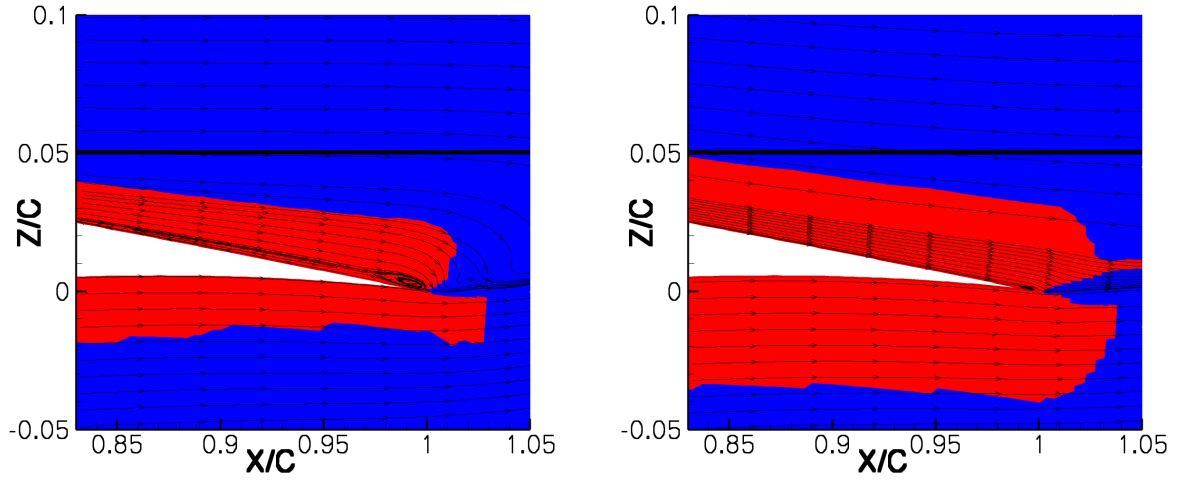


Figure 6.12: RANS-LES distribution and streamlines of SA-DDES (left) and SA-DDES₁₆ (right) on the original grid at the trailing edge. Both computations were started with an SA-RANS restart and have then been further computed for three convective time units.

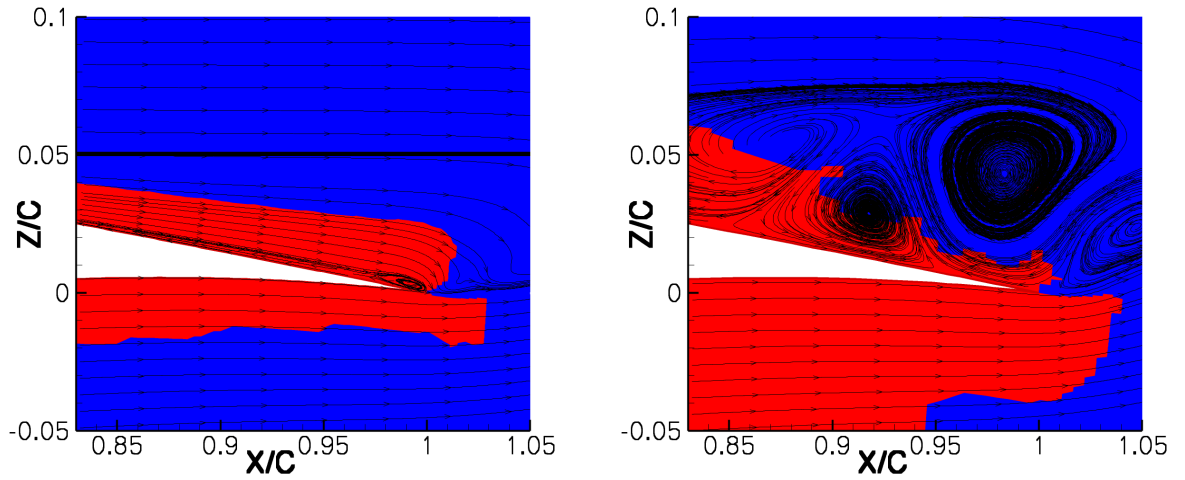


Figure 6.13: RANS-LES distribution and streamlines of the two-dimensional SA-DDES computation on the modified grid (left) and SA-DDES on the original grid (right) in the middle of the upper surface. While the left figure shows the result after three convective time units in SA-DDES mode, the right figure is the outcome after 8 convective time units were performed with the SA-DDES model.

6.2.2 Examination of Reynolds stress distribution

The final investigation in this section concentrates on the non-dimensionalised Reynolds stress UW/U_∞^2 resulting from the various computations. Within this testcase this

variable ideally should be similar in the SA-DDES and SA-DDES₁₆ computations compared to the one obtained by SA-RANS.

As UW cannot be obtained directly, it is approximated as follows:

$$UW = -\sigma\tau_{13}, \quad \tau_{13} = \tau_{13}^{modelled} + \tau_{13}^{resolved} \quad (6.1)$$

$$\begin{aligned} \tau_{13}^{modelled} &= -\langle \nu_t (\frac{\partial}{\partial x} u_3 + \frac{\partial}{\partial z} u_1) \rangle \\ &\approx -\langle \nu_t \rangle \langle (\frac{\partial}{\partial x} u_3 + \frac{\partial}{\partial z} u_1) \rangle \approx -\langle \nu_t \rangle (\frac{\partial}{\partial x} \langle u_3 \rangle + \frac{\partial}{\partial z} \langle u_1 \rangle) \end{aligned} \quad (6.2)$$

$$\tau_{13}^{resolved} = \begin{cases} -\frac{\langle u_1' u_3' \rangle}{\sigma} & \text{if SA - DDES or SA - DDES}_{16} \text{ computation} \\ 0 & \text{if SA - RANS computation} \end{cases} \quad (6.3)$$

The operator $\langle . \rangle$ denotes averaging in time, u_i is the i -th component of the velocity, τ_{13} is one of the components of the Reynolds stress tensor, $\tau_{13}^{modelled}$ is its modelled part, $\tau_{13}^{resolved}$ its resolved part and u_i' is the fluctuation of the i -th velocity component according to Eq. (3.1).

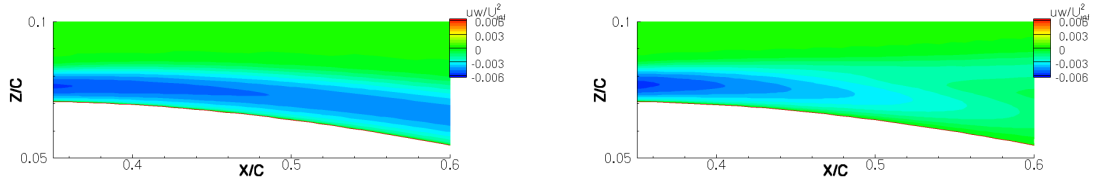


Figure 6.14: Normalised Reynolds stress UW/U_∞^2 of SA-RANS (left) and SA-DDES (right) on the original grid in the middle of the upper surface. The SA-DDES computation was started with the SA-RANS restart and has then been further computed for three convective time units.

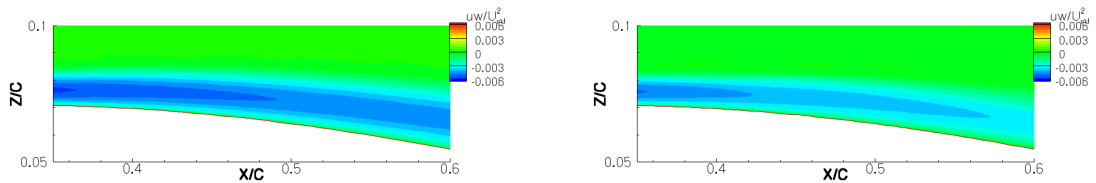


Figure 6.15: Normalised Reynolds stress UW/U_∞^2 of SA-DDES₁₆ on the original grid (left) and of the two-dimensional SA-DDES computation on the modified grid (right) in the middle of the upper surface. Both computations were started with the SA-RANS restart and have then been further computed for three convective time units.

Figures 6.14 – 6.15 show UW/U_∞^2 based on the steady SA-RANS computation, the SA-DDES simulation on the original grid, the SA-DDES₁₆ simulation on the original grid and the SA-DDES simulation on the modified grid, respectively, in the

middle of the upper surface. Note, that Fig. 6.15 (right) must be treated with caution, because a two-dimensional SA-DDES computation can not be expected as physically correct. Besides the SA-RANS simulation, all three figures are the outcome after three convective time units. Although Fig. 6.14 (right) and Fig. 6.15 (right) differ slightly, in both cases a loss of UW/U_∞^2 starting from about $x/c = 0.4$ can be observed. Only the SA-DDES₁₆ simulation conserves the Reynolds stress comparably well as SA-RANS and hereby avoids modelled stress depletion.

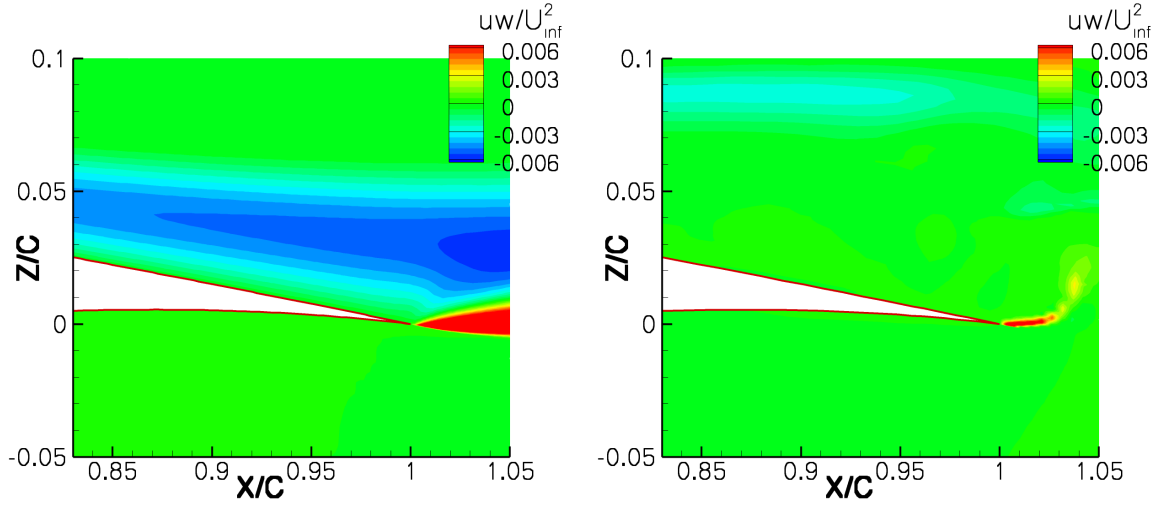


Figure 6.16: Normalised Reynolds stress UW/U_∞^2 of SA-RANS (left) and SA-DDES (right) on the original grid at the trailing edge. The SA-DDES computation was started with the SA-RANS restart and has then been further computed for three convective time units.

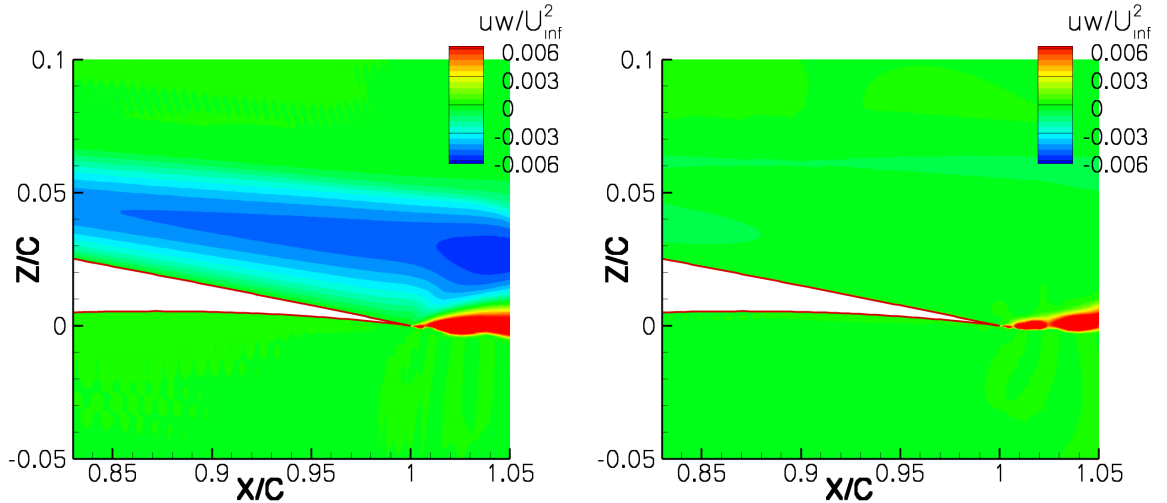


Figure 6.17: Normalised Reynolds stress UW/U_∞^2 of SA-DDES₁₆ on the original grid (left) and of the two-dimensional SA-DDES computation on the modified grid (right) at the trailing edge. Both computations were started with the SA-RANS restart and have then been further computed for three convective time units.

This observation is consistent with the results presented in Figs. 6.16 – 6.17, where UW/U_∞^2 of the same computations as in Figs. 6.14 – 6.15 are shown with focus on the trailing edge. Again, Fig. 6.17 (right) must be treated with caution. While the values indicated in Fig. 6.16 (right) and Fig. 6.17 (right) are almost zero, the SA-DDES₁₆ computation is comparably good as the one obtained by SA-RANS.

Summary of the section

Overall the results presented in this section have proven that within this testcase the standard version of SA-DDES leads to modelled stress depletion and a subsequent grid induced separation. Also the use of a modified grid did not overcome this flaw of the model. As only the SA-DDES₁₆ modification enabled a stable and reasonable computation, only SA-DDES₁₆ was used for the further simulations at higher angles of attack within this testcase. The results are subject to the next section.

6.3 Results with SA-DDES₁₆ of the HGR01 airfoil at stall

In this section the results of the SA-DDES₁₆ computation on the original grid at $\alpha = 12^\circ - 16^\circ$ are presented.

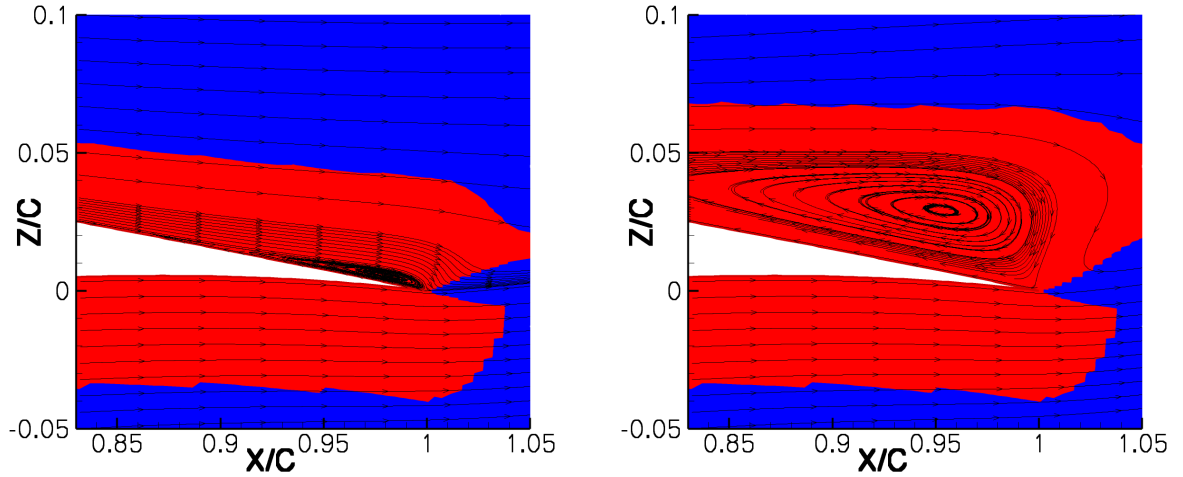


Figure 6.18: RANS-LES distribution and streamlines of SA-DDES₁₆ on the original grid at $\alpha = 14^\circ$ (left) and $\alpha = 16^\circ$ (right) at the trailing edge.

As already described at the beginning of this chapter, first a steady three-dimensional SA-RANS simulation at $\alpha = 12^\circ$ is performed, which is then used as restart solution for the following SA-DDES₁₆ computations. After having reached a convergent behaviour of the aerodynamic coefficients c_l , c_d and c_m , the grid is rotated to increase the angle of attack by 1° . This procedure is performed repeatedly until the maximal

angle of $\alpha = 16^\circ$ has been reached. Figures 6.18 (left) and (right) show the RANS-LES distributions and velocity streamlines at $\alpha = 14^\circ$ and $\alpha = 16^\circ$, respectively. Compared to Fig. 6.12 (right) it is obvious, that the trailing edge separation becomes larger with increased angle of attack. As a result also the RANS-part thickens such that it completely contains the grown separation bubble.

6.3.1 Pressure and skin friction distributions

In Figs. 6.19 – 6.23 the pressure distributions and skin friction distributions of various computations at different angles of attack are presented. The variables resulting from SA-DDES₁₆ have been averaged both in time and in spanwise direction such that the shown results are statistically converged mean values.

For comparison, steady two-dimensional SA-RANS computations were performed at each angle of attack. While in all figures both the results of SA-RANS and SA-DDES₁₆ are shown, Fig. 6.19 (left) and Fig. 6.19 (right), which are based on the results at $\alpha = 12^\circ$, also contain the outcome of SA-DDES both on the original grid and on the modified grid. Here the former is based on the result after 8 convective time units and the latter adverts to the computation after three convective time units. As it was shown in Chapter 6.2 that these computations did not lead to a stable and reasonable outcome, they were not perpetuated at higher angles of attack. As no statistically converged solution could be obtained, the shown values are based on the spanwise average of the instantaneous flow fields at the end of the two simulations.

To evaluate the quality of the computations, also experimental data of the pressure distributions are included in the respectively left figures. The first four figures of the skin friction distributions additionally contain experimental data regarding the separation point. As in the experiments this point was not fixed - it varied both in time and in spanwise direction – both the minimum and maximum values are shown.

In Figs. 6.19 (left) and (right) one can see that both SA-RANS and SA-DDES₁₆ lead to similar results (the red and blue line lie on top of each other), whereas the two SA-DDES computations predict a larger flow separation due to modelled stress depletion. While in Fig. 6.19 (left) both SA-RANS and SA-DDES₁₆ agree well with the experimental data throughout most of the airfoil, some slight differences can be observed at the upper surface near the leading edge, where the laminar separation bubble is located. Figure 6.19 (right) again shows that SA-RANS and SA-DDES₁₆ lead to the same outcome, while the two SA-DDES computations differ. Especially the SA-DDES computation on the original grid, which has been performed for much longer than all other simulations, includes a large region of flow separation. Although the separation point predicted by SA-DDES on the original grid seems to be in good agreement with the experimentally obtained values of the minimum and maximum separation, it must be emphasized that this is only coincidence and should not be regarded as a sign of model soundness.

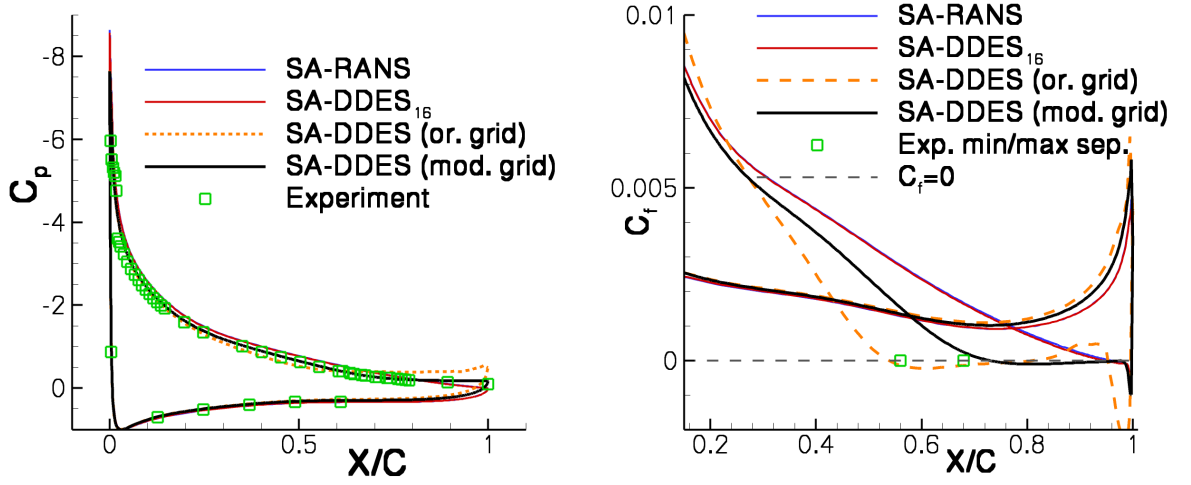


Figure 6.19: Pressure (left) and skin friction (right) distributions of SA-RANS, SA-DDES₁₆, SA-DDES on the original grid, SA-DDES on the modified grid and comparison with experimental data at $\alpha = 12^\circ$.

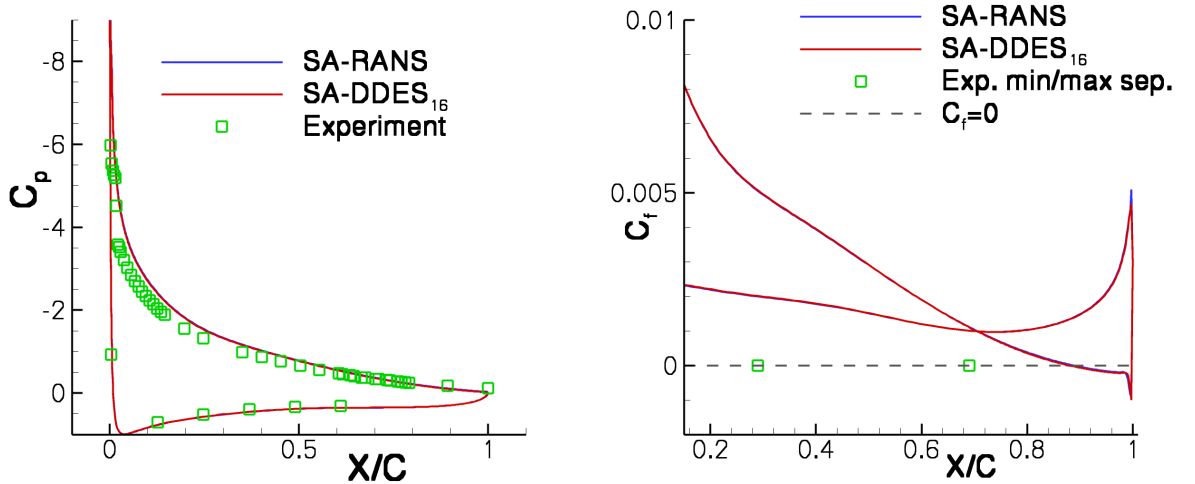


Figure 6.20: Pressure (left) and skin friction (right) distributions of SA-RANS and SA-DDES₁₆ on the original grid and comparison with experimental data at $\alpha = 13^\circ$.

In Fig. 6.20 and Fig. 6.21 it is revealed that SA-RANS and SA-DDES₁₆ also produce almost the same results at higher angles of attack, while the differences to the experimental data increase: Especially when looking at Fig. 6.21 (left), the experimentally obtained C_p -values flatten out much earlier on the upper surface than indicated by the numerical computations, which is a sign that the trailing edge separation within the experiment is larger than in the simulations. This assumption is further supported by Fig. 6.20 (right) and Fig. 6.21 (right): While the separation points predicted by the numerical simulations are shifted towards the leading edge

with increasing angle of attack, they still are located much too far downstream the airfoil compared to the experiment.

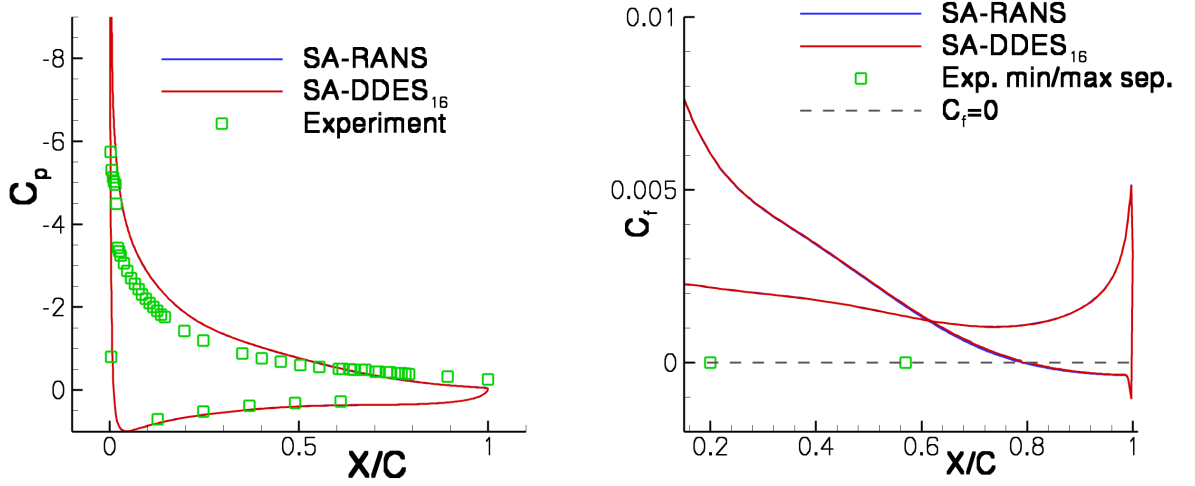


Figure 6.21: Pressure (left) and skin friction (right) distributions of SA-RANS and SA-DDES₁₆ on the original grid and comparison with experimental data at $\alpha = 14^\circ$.

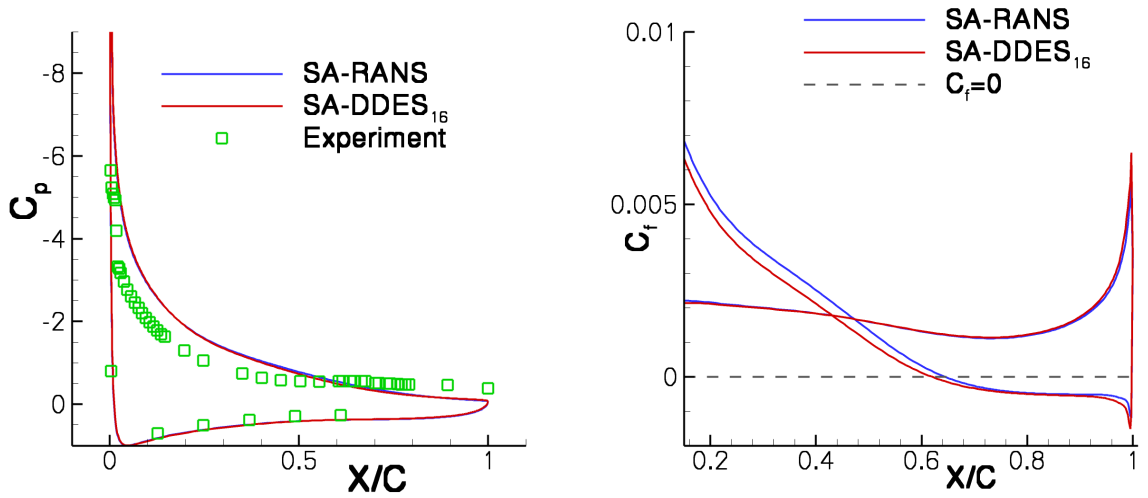


Figure 6.22: Pressure (left) and skin friction (right) distributions of SA-RANS and SA-DDES₁₆ on the original grid and comparison with experimental data at $\alpha = 15^\circ$.

When looking at even higher angles of attack, SA-RANS and SA-DDES₁₆ start to behave in a slightly different way. While the pressure distributions of the two models still look very similar for $\alpha = 15^\circ$ and $\alpha = 16^\circ$ (see Fig. 6.22 (left) and Fig. 6.23 (left)), the skin friction distributions shown in Fig. 6.22 (right) and Fig. 6.23 (right) differ slightly. This unequal behaviour of the two models is due to the fact that the trailing edge separation increases vastly at higher angles of attack, while the RANS-part resulting from the SA-DDES₁₆ computation does not thicken enough to contain

the complete boundary layer. This leads to a partially modelled stress depletion. Compared to the experiment, both models again fail to reproduce the increased flow separation.

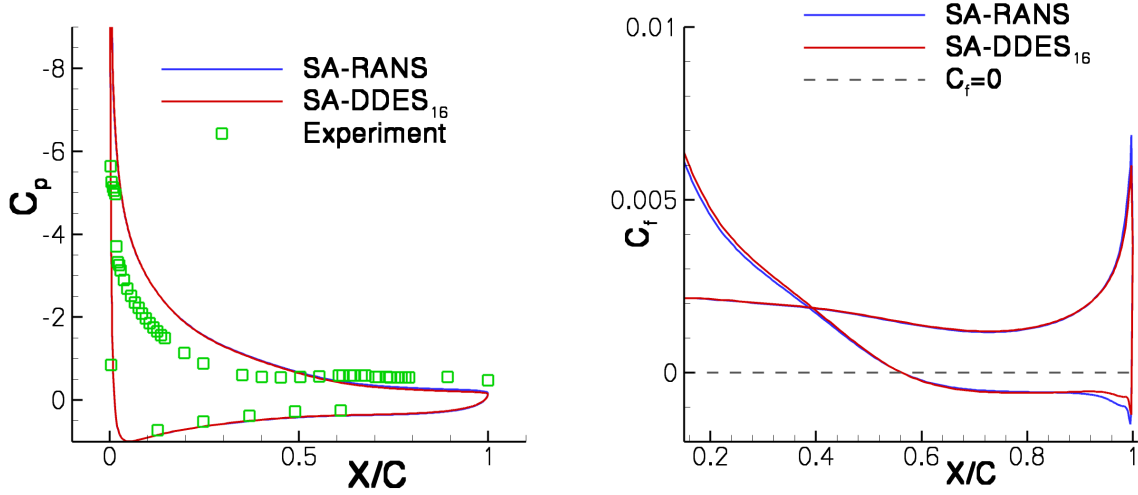


Figure 6.23: Pressure (left) and skin friction (right) distributions of SA-RANS and SA-DDES₁₆ on the original grid and comparison with experimental data at $\alpha = 16^\circ$.

6.3.2 Detection of three-dimensional structures

The last investigation within this chapter is concerned with the question, whether the use of SA-DDES₁₆ leads to the creation of three-dimensional structures at the trailing edge and in the wake, which is highly desirable when using DES models. As indicator to detect such structures, the instantaneous Q-criterion is referred to. Figures 6.24 (left) and (right) and Fig. 6.25 (left) show isosurfaces of the Q-criterion at a value of $Q = 10^{-5} \frac{1}{s^2}$ at $\alpha = 12^\circ$, $\alpha = 14^\circ$ and $\alpha = 16^\circ$, respectively. Although the illustrated value is extremely small, still only two-dimensional rolls can be seen, which is typical for RANS computations. It therefore can be concluded that the SA-DDES₁₆ model does not lead to three-dimensional structures within this testcase. As in literature it is well-known that the use of a too coarse grid (especially in streamwise direction), of a too small spanwise extension and a too large timestep size can suppress the evolving of three-dimensional structures within DES simulations, all of these three parameters have been varied to examine their influence. Recall, that all SA-DDES₁₆ computations shown in this chapter so far were based on a spanwise extension of $0.2c$ and contained 33 planes in spanwise direction. Firstly, three simulations including a spanwise extension of $0.2c$, $0.4c$ and $0.8c$ and 65 planes in spanwise extension, respectively, were performed for 10 convective time units at $\alpha = 14^\circ$ with fixed timestep size $\Delta t = 1 \cdot 10^{-4}s$. As restart for these simulations the SA-DDES₁₆ result of the original computation at $\alpha = 14^\circ$ was doubled in spanwise extension and - if necessary- stretched to meet the respectively spanwise extension. Figure 6.25 (right) and Figs. 6.26 (left) and (right) show the resulting isosurfaces of the three computations at a value of $Q = 10^{-5} \frac{1}{s^2}$. As still no three-dimensional

structures can be observed, it can be concluded that neither an increase of the spanwise extension nor a spanwise grid refinement lead to the creation of three-dimensional structures. Additionally, all three computations were performed for three more convective time units at a decreased timestep size of $\Delta t = 2.5 \cdot 10^{-5}s$, which also did not result in structures. As the figures of the instantaneous Q-criterion look identical to the ones obtained at a timestep size of $\Delta t = 1 \cdot 10^{-4}s$, they are not illustrated here.

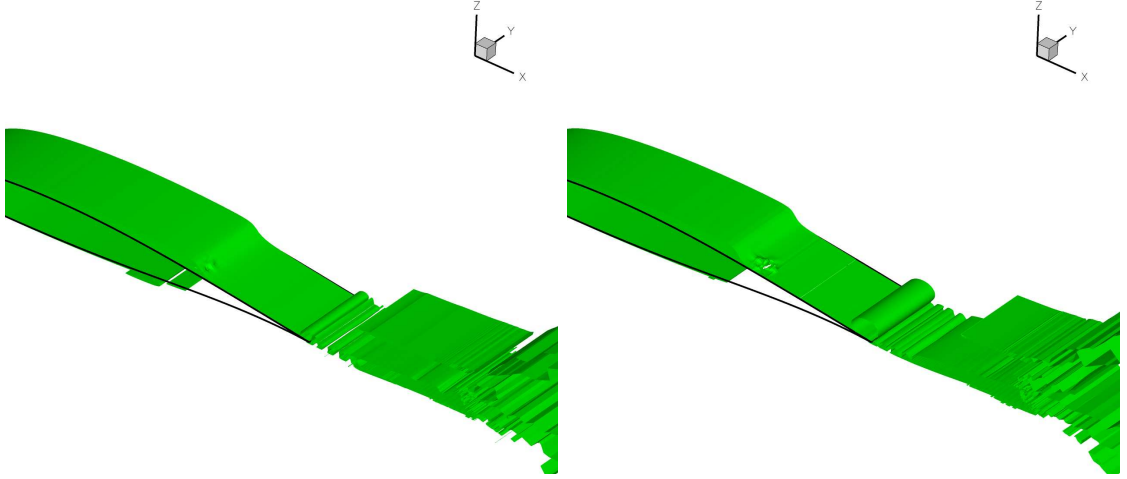


Figure 6.24: Isosurfaces of the instantaneous Q-criterion of SA-DDES₁₆ at a value of $10^{-5} \frac{1}{s^2}$ at $\alpha = 12^\circ$ (left) and $\alpha = 14^\circ$ (right).

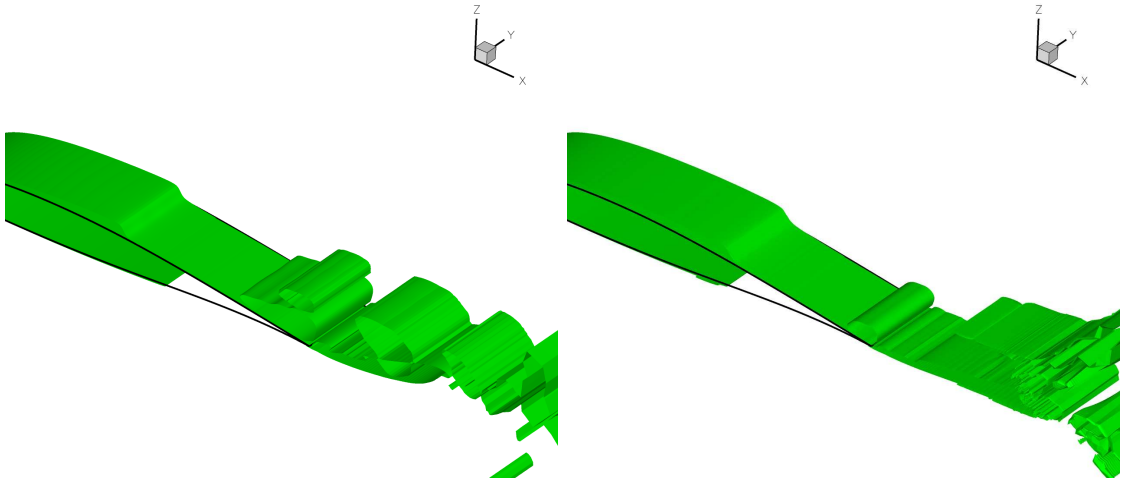


Figure 6.25: Isosurfaces of the instantaneous Q-criterion of SA-DDES₁₆ at a value of $10^{-5} \frac{1}{s^2}$ at $\alpha = 16^\circ$ (left) and $\alpha = 14^\circ$ (right). The right figure is based on the grid with a spanwise extension of $0.2c$ and 65 planes in spanwise extension.

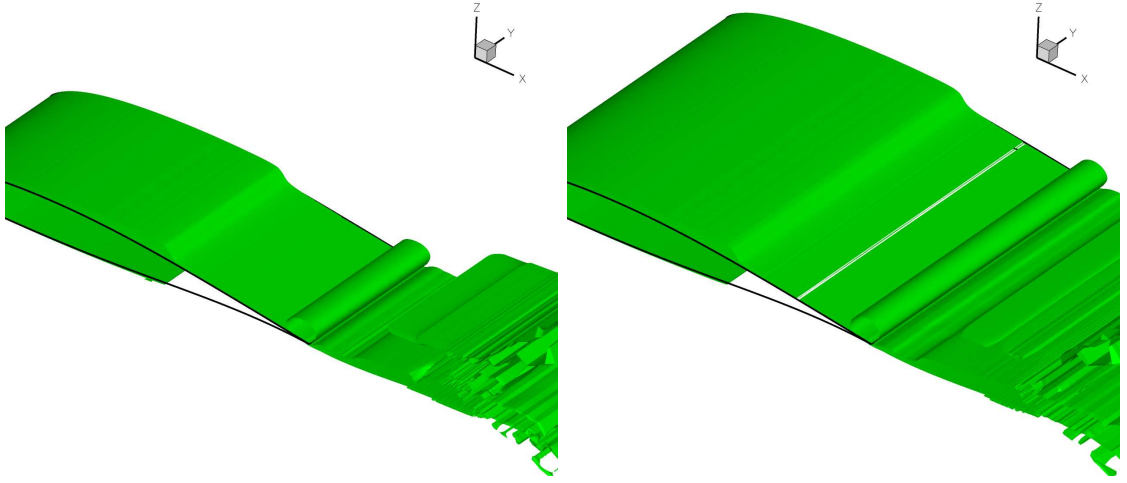


Figure 6.26: Isosurfaces of the instantaneous Q-criterion of SA-DDES₁₆ at a value of $10^{-5} \frac{1}{s^2}$ at $\alpha = 14^\circ$. Shown are the results based on the grid with a spanwise extension of $0.4c$ and 65 planes in spanwise extension (left) and of the grid with a spanwise extension of $0.8c$ and 65 planes in spanwise extension (right).

Figure 6.27, which shows the instantaneous C_f distribution of the SA-DDES₁₆ computation at $\alpha = 16^\circ$ on the upper airfoil surface, implies that even at the highest angle of attack considered, the separation point does not vary in spanwise direction. (The result is based on the grid with a spanwise extension of $0.2c$ and 65 planes in spanwise extension.) This is one more proof that no three-dimensional structures have evolved.

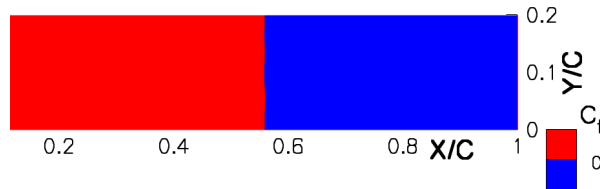


Figure 6.27: Instantaneous C_f distribution of SA-DDES₁₆ at $\alpha = 16^\circ$ based on the grid with a spanwise extension of $0.2c$ and 65 planes in spanwise extension.

At this point we want to mention that an alternative scheme to discretise the convectives fluxes has been implemented into the DLR TAU-code (see [124]), which leads to the creation of turbulent three-dimensional structures in similar testcases as the one shown here. As this scheme has not been included into the official TAU-release at the time the HGR01 simulations were performed, it has not been used here. Nevertheless, in the future this scheme should be applied for the flow around the HGR01 airfoil to test, if this approach can trigger turbulent structures.

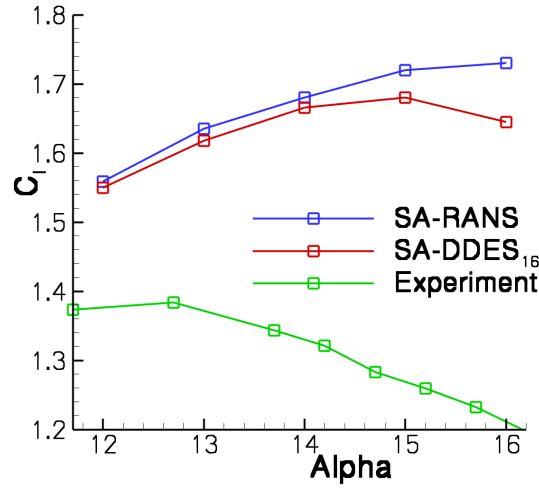


Figure 6.28: Comparison of the lift coefficients of SA-RANS, SA-DDES₁₆ and experimental data.

Figure 6.28 shows the C_l values given by SA-RANS, SA-DDES₁₆ and the experiment. While SA-RANS and SA-DDES₁₆ predict similar values at $\alpha = 12^\circ - 14^\circ$, the use of SA-DDES₁₆ leads to a C_l breakdown at higher angles of attack, which is due to the already mentioned beginning modelled stress depletion. As it was already shown in Figs. 6.19 – 6.23 that the trailing edge separation is vastly underpredicted by both models compared to the experiment, it comes with no surprise that the experimental C_l values strongly differ from the ones resulting from the two numerical models.

Summary of the chapter and implications on the simulation method

As conclusion of Chapter 6 it has been shown that the use of the standard SA-DDES model applied to the HGR01 airfoil at stall leads to a diminished RANS part. As a consequence, modelled stress depletion and grid induced flow separation occurs, which prevents the performance of a stable and reasonable computation. Also a modification of the underlying grid was not capable of eliminating these flaws in the model. Only the use of a modified version of SA-DDES ("SA-DDES₁₆") prevented the loss of the Reynolds stresses and resulted in similar outcomes as SA-RANS computations. Despite the variation of several parameters- spanwise extension, spanwise grid refinement and timestep size - the use of the SA-DDES₁₆ model did not result in desirable three-dimensional structures in the region of the trailing edge separation and the wake.

Based on the results in this chapter, the use of the SA-DDES turbulence model can be restrictedly recommended for the applications in Chapter 8. If the RANS part is ensured to cover the complete attached boundary layer at the wing, the model can be expected to perform satisfactory.

Chapter 7

Transport and collision of vortices

The aim of this chapter is to verify and validate the Chimera technique applied to two-dimensional simulations of transported large-scale vortices and vortex-airfoil interactions. The principle can be described as follows: Firstly, one or more analytical vortices are interpolated onto a fine cartesian vortex transport grid, which lies embedded within a background grid. Both the vortex and the vortex transport grid are then simultaneously moved with the freestream velocity to the right. In the case of a subsequent vortex-airfoil interaction the vortex transport grid is stopped just before reaching the airfoil, while the vortex continues to move to the right. The vortex-airfoil interaction eventually takes place on the background grid.

The investigations in this chapter are especially significant with regard to Chapter 8, where a vortex has to be transported through a coarse background grid. Also the vortex-airfoil interactions are an important part of the simulations in Chapter 8. As one can see in Figs. 7.1 (left) and (right), where the vorticity of the NACA 0012 testcase in Section 7.3 and alternatively of the testcase in Section 8.4 is shown, both setups are highly similar.

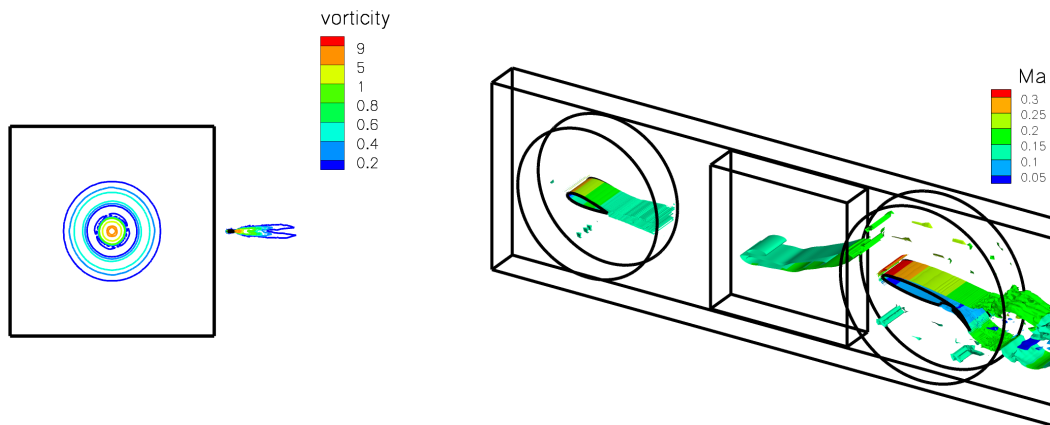


Figure 7.1: Vorticity of NACA 0012 testcase in Section 7.3 (left) and isosurfaces of the vorticity at a value of 100 of testcase in Section 8.4 (right).

7.1 Description of the testcases

In this section a short overview of the presented testcases is provided. The first three examples in Section 7.2 consider the transport of analytical finite vortices due to an external convection field, where especially the vortex decay in time is examined. Firstly, a vortex is resolved on a fine vortex transport grid that is moved together with the vortex relative to a background grid by applying the Chimera technique. In the second example a much coarser background grid is used instead. In the third example the convection of three finite vortices (who are almost not interacting due to their sufficiently large mutual distance) is examined numerically. In Section 7.3 the collision of a finite vortex and a NACA 0012 airfoil at $\alpha = 0^\circ$ is simulated. The final example shown in Section 7.4 examines the interaction of a vortex and an ONERA-A airfoil at $\alpha = 13.3^\circ$, leading to flow separation at the trailing edge.

In all cases the farfield boundary condition is applied at the inflow and outflow. Farfield is also chosen at the upper and lower wall in the first three examples. Viscous wall are prescribed at the two bodies in the two final testcases.

In all numerical examples shown one or more finite vortices, which were defined in definition 2.2.1, are initialised onto a vortex transport grid and then moved together with the latter through the background grid. The velocity relative to the background grid is denoted by v and its x- and z-components by v_x and v_z , respectively.

To be precise, one has to distinguish between v_∞ and the velocity v_{vg} by which the vortices are transported through the domain. As in all examples both velocities are equal in magnitude and direction, only v_∞ will be specified in order to keep the notation simple. Also the so-called local velocity

$$v_{local} = ((v_x - v_{\infty,x})^2 + (v_z - v_{\infty,z})^2)^{\frac{1}{2}} \quad (7.1)$$

is helpful when dealing with moving grids, where $v_{\infty,x}$ and $v_{\infty,z}$ are the x- and z-components of v_∞ , respectively. Note, that v_{local} is identical to v_ϕ on the vortex transport grid (with respect to the local coordinates x_{local} and z_{local} of the vortex transport grid), if v_{vg} equals v_∞ . This holds always true with the exception of the final parts of the two examples in Section 7.3 and 7.4, where the vortex transport grid is stopped, while v_∞ remains nonzero.

7.2 Examinations of vortex transport

This section considers two-dimensional simulations of transported analytical vortices. The aim is to determine, whether the vortices can be preserved without loosing them due to numerical dissipation. Also the influence of using a coarse background grid and the simultaneous transport of more than one vortex is subject of this section.

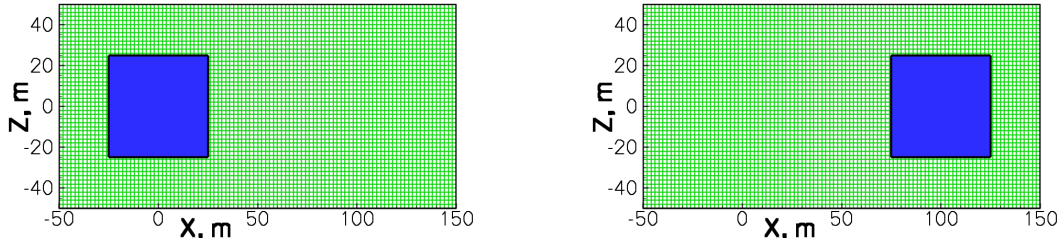


Figure 7.2: Relative position of the background grid and the vortex transport grid at beginning (left) and end (right) of examples 7.2.1, 7.2.2 and 7.2.3.

7.2.1 Transport of a finite vortex within a fine background grid

In this first Chimera example the transport of a finite vortex is simulated, where both the background grid and the vortex transport grid are fine. The equidistant background grid covers the rectangular area given by $(x, z) \in [-50m, 150m] \times [-50m, 50m]$ and it contains 601×201 grid points. The vortex transport grid, which is adapted similar as in Fig. 2.5, covers the domain $(x_{local}, z_{local}) \in [-25m, 25m] \times [-25m, 25m]$ and contains 201×201 grid points. The cells of the background grid and of the outer edge of the vortex transport grid are of approximately equal size, which allows a reasonable interpolation. At the beginning of the computation x_{local} equals x , which results in a relative position of both grids at $t = 0s$ as shown in Fig. 7.2 (left). The hole definition grid covers the area $(x_{local}, z_{local}) \in [-23m, 23m] \times [-23m, 23m]$ and the finite vortex is defined by the properties $\Gamma_0 = 60$, $r_c = 2m$, $r_m = 4m$, $r_o = 10m$, $\delta = 10m^2$ and its center is located at $(x_{local}, z_{local}) = (0m, 0m)$. Both $v_{\infty, x}$ and $v_{vg, x}$ are prescribed as 50 m/s. A time step size of $\Delta t = 10^{-2}s$ is chosen and the computation is performed until 2s of physical time have elapsed. At the end of the simulation both the vortex transport grid and the vortex have moved 100m to the right. Figure 7.2 (right) shows the relative position of both grids at the end of the computation.

Figure 7.3 (left) shows v_{local} at various times, where again only the values at points $(x_{local}, z_{local}) \in [0m, 10m] \times \{0m\}$ are displayed. Comparing Fig. 7.3 (left) with the results obtained in Subsections 2.2.4, 2.3.3 and 2.4.3 v_{local} seems to sustain longer in the Chimera example shown here.

So the most important result of this first Chimera example is the observation that the use of the Chimera technique does not increase the dissipation of the vortex in an unphysical way.

7.2.2 Transport of a finite vortex within a coarse background grid

The goal of this example is to examine whether the good resolution in example 7.2.1 can be retained when choosing a background grid that is much coarser than the vortex transport grid. The equidistant background grid now contains

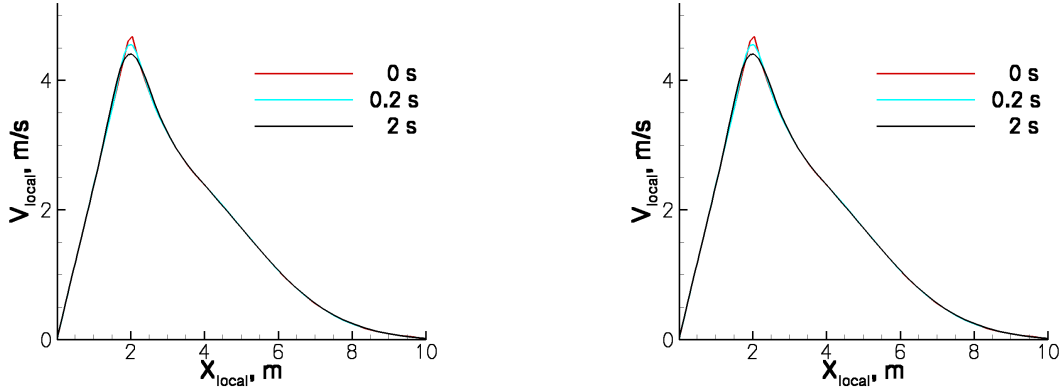


Figure 7.3: Local velocity at various times and $(x_{local}, z_{local}) \in [0m, 10m] \times \{0m\}$ in example 7.2.1 using a fine background grid (left) and example 7.2.2 using a coarse background grid (right).

101×51 grid points. The area covered by the hole definition grid is reduced to $(x_{local}, z_{local}) \in [-18m, 18m] \times [-18m, 18m]$, which is necessary to ensure a sufficiently large overlapping region of the vortex transport grid and the active part of the background grid. All other data remain unchanged compared to example 7.2.1. Figure 7.3 (right) shows v_{local} displayed at the same time and space coordinates as in Fig. 7.3 (left). Comparing both figures, no significant differences can be noticed. This shows that a coarse background grid does not noticeably increase the numerical dissipation.

7.2.3 Transport of three finite vortices within a coarse background grid

The aim of this third Chimera example is to demonstrate that the method can be used for the transport of more than one vortex. Compared to example 7.2.2, only two changes are accomplished. Firstly, three finite vortices are initialised at the beginning. The first vortex is defined by the properties $\Gamma_0 = 160, r_c = 3m, r_m = 6m, r_o = 12m, \delta = 10m^2$ and its center is located at $(x_{local}, z_{local}) = (-8m, 8m)$, the second vortex is defined by $\Gamma_0 = 63, r_c = 2m, r_m = 4m, r_o = 10m, \delta = 10m^2$ and its center is located at $(x_{local}, z_{local}) = (10m, 10m)$, whereas the third vortex is defined by $\Gamma_0 = 37, r_c = 1.6m, r_m = 3.2m, r_o = 8m, \delta = 10m^2$ and its center is located at $(x_{local}, z_{local}) = (3m, -10m)$.

As an adaption of the vortex transport grid does not seem reasonable when transporting more than one vortex, the vortex transport grid that again consists of 201×201 grid points is secondly chosen as equidistant.

Figures 7.4 (left) and (right) show v_{local} at start and end of the computation, respectively, on a cutout of the computational domain. One can see that the three vortices have been conserved satisfactory. Noticeable is also the very limited interaction between the vortices due to their relatively large mutual distance.

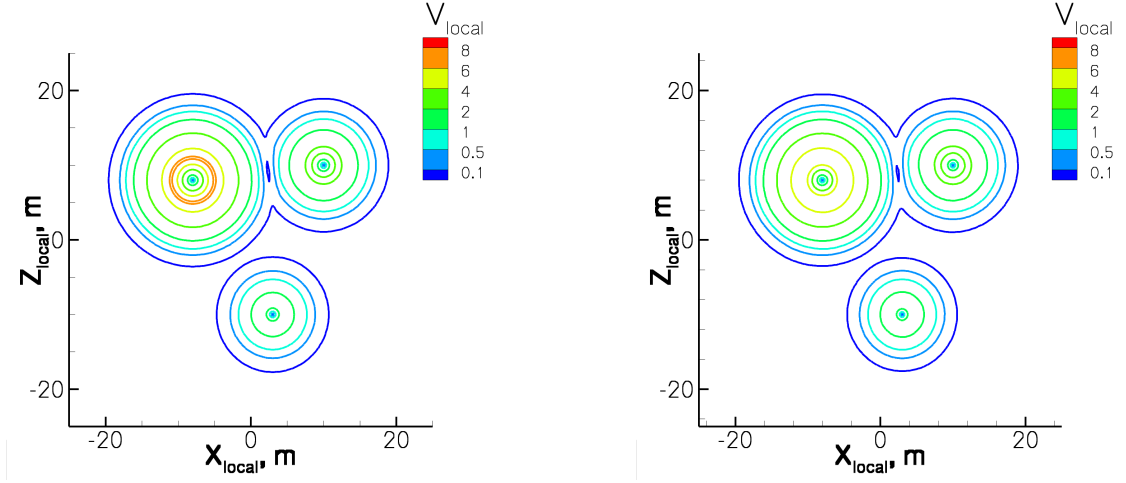


Figure 7.4: Local velocity at beginning (left) and end (right) of example 7.2.3.

Overall this third Chimera example shows that the transport of more than one finite vortex does not cause additional problems.

To sum up Section 7.2 we have shown that the Chimera technique allows to preserve transported vortices. Also the use of a coarse background grid and the transport of more than one vortex does not result in an increased numerical dissipation.

7.3 Vortex interaction with a NACA 0012 airfoil at $\alpha = 0^\circ$

In this example the interaction between a finite vortex and a NACA 0012 airfoil is simulated.

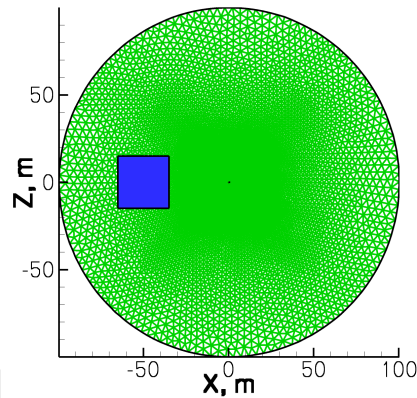


Figure 7.5: Relative position of the background grid, the vortex transport grid and the airfoil at $t = 0$ s in example 7.3.

As one can see in Fig. 7.5 the background grid describes a circular area of ra-

dius 100m. It consists of 55558 grid points and contains the airfoil in its center. The chord length is prescribed as 1m and the angle of attack is $\alpha = 0^\circ$. The Reynolds number and Mach number are $Re = 3.8 \times 10^6$ and $Ma = 0.15$. $v_{\infty,x}$ and $v_{vg,x}$ are prescribed as 50 m/s. The background grid is of hybrid type and is composed of prismatic elements in the boundary layer and tetrahedral cells in the outer domain. At $t = 0s$ the vortex transport grid, which contains 151×151 points, covers the domain $(x, z) \in [-65m, -35m] \times [-15m, 15m]$ or accordingly $(x_{local}, z_{local}) \in [-15m, 15m] \times [-15m, 15m]$. Due to reasons explained in the third part of this example an equidistant vortex transport grid is chosen. Figure 7.5 shows the relative position of the vortex transport grid and the background grid at the beginning of the computation. The airfoil can hardly be seen in the center of the background grid.

As the background grid is finer towards the airfoil, the hole definition grid is not placed in the center of the vortex transport grid, but is shifted 1m to the right. This results in a hole definition grid covering the area $(x_{local}, z_{local}) \in [-9m, -11m] \times [-10m, 10m]$.

The computation is divided into three parts. First a steady computation is performed in order to obtain a stable restart solution. Note that at this stage the finite vortex has not been inserted into the flow field yet.

Only at the beginning of the second part of the computation the finite vortex is initialised in the center of the vortex transport grid. It is defined by the properties $\Gamma_0 = 60, r_c = 1.5m, r_m = 3m, r_o = 7m, \delta = 5m^2$ and its center is located at $(x_{local}, z_{local}) = (0m, 0m)$.

Figures 7.6 (left) and (right) show v_{local} on the vortex transport grid and around the airfoil, respectively, after the initialisation of the vortex. Note that the distance of the finite vortex and the airfoil is chosen sufficiently large so that initially the presence of the vortex does not influence the flow around the airfoil.

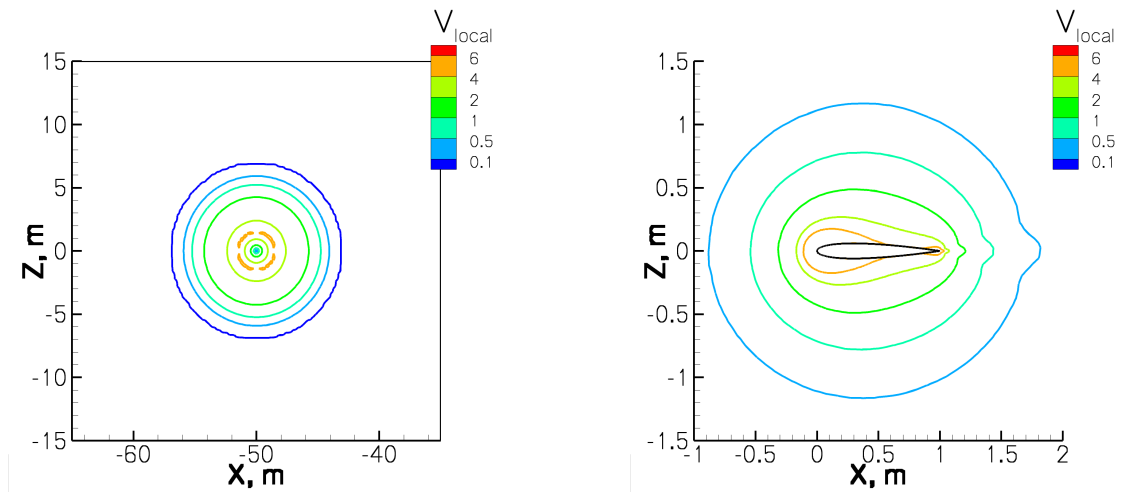


Figure 7.6: Local velocity after the initialisation of the finite vortex at $t = 0s$ in example 7.3.

A time step size of $\Delta t = 10^{-4}$ s is chosen. The simulation is then performed until 0.66s of physical time have passed, which marks the end of the second part of the computation. At this time the distance of the right boundary of the vortex transport grid to the airfoil is only 2m. Figure 7.7 (left) shows v_{local} on a domain of interest at $t = 0.66$ s; the boundary of the vortex transport grid is displayed, too. In Fig. 7.7 (right) v_{local} is presented in more detail around the airfoil. Comparing Fig. 7.7 (right) with Fig. 7.6 (right), no major differences can be noticed except to the right of the trailing edge. This indicates that the flow fields of the vortex and around the airfoil have not interacted directly yet.

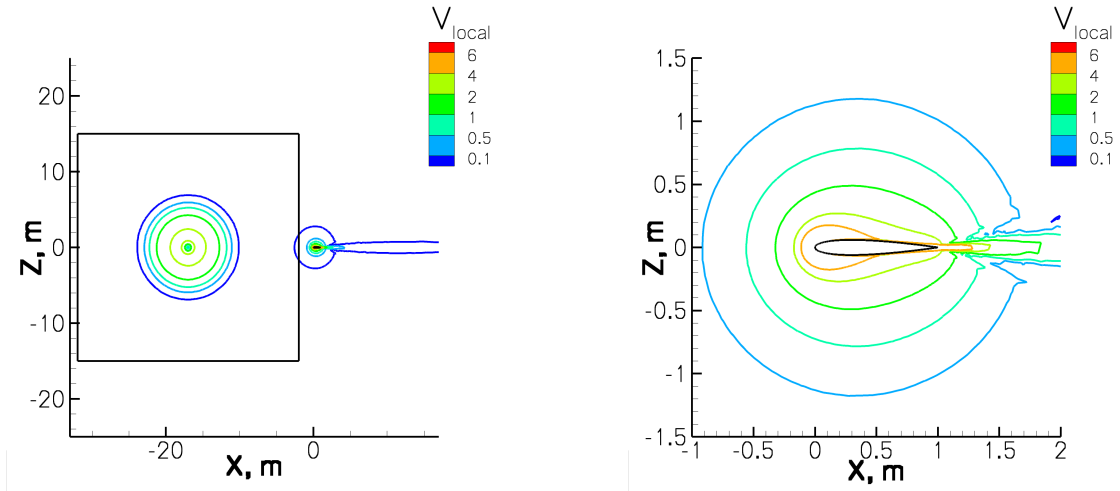


Figure 7.7: Local velocity at $t = 0.66$ s in example 7.3. The vortex transport grid is stopped at this stage.

While the vortex continues to move to the right, the vortex transport grid is stopped at the beginning of the third part of the computation. This is due to two reasons: Firstly the vortex transport grid and the airfoil must not overlap, as in this case no donor grid cells which are required for the Chimera interpolation onto the vortex transport grid, are existent in the region covered by the airfoil. Secondly the grid cells of the background grid close to the airfoil are much finer than the cells of the vortex transport grid. Therefore the vortex is better resolved on the background grid in this region.

As the computation proceeds, the vortex passes the area covered by the hole definition grid and reaches the artificial boundary around the hole of the background grid. Here it is then successively interpolated completely onto the background grid. At this stage the reason for choosing an equidistant vortex transport grid becomes obvious: Whereas an adapted vortex transport grid, as it is shown in Fig. 2.5, is advantageous when using a single vortex resting in the center, such a grid is not appropriate in the case of a vortex moving across the coarser grid part at the boundary. The interpolation is further improved by increasing the hole definition grid 2.5m towards the right at the beginning of the third part of the computation, which leads to finer grid cells in the right part of the artificial boundary of the background grid. The new hole definition grid covers the area $(x_{local}, z_{local}) \in [-9m, -13.5m] \times [-10m, 10m]$.

This approach is mathematically correct, because it is uncritical to increase the set of grid points that are excluded from the computation.

Figure 7.8 (left) shows v_{local} after 1s of physical time has elapsed. The major part of the vortex has already been interpolated onto the background grid, which is emphasized by showing the right boundary of the vortex transport grid (the interpolation is performed somewhat left of this boundary). Comparing Fig. 7.8 (right), which shows v_{local} near the airfoil, with Fig. 7.7 (right), one can see that the vortex is intensely interacting with the flow around the airfoil. The vortex core has split into two halves.

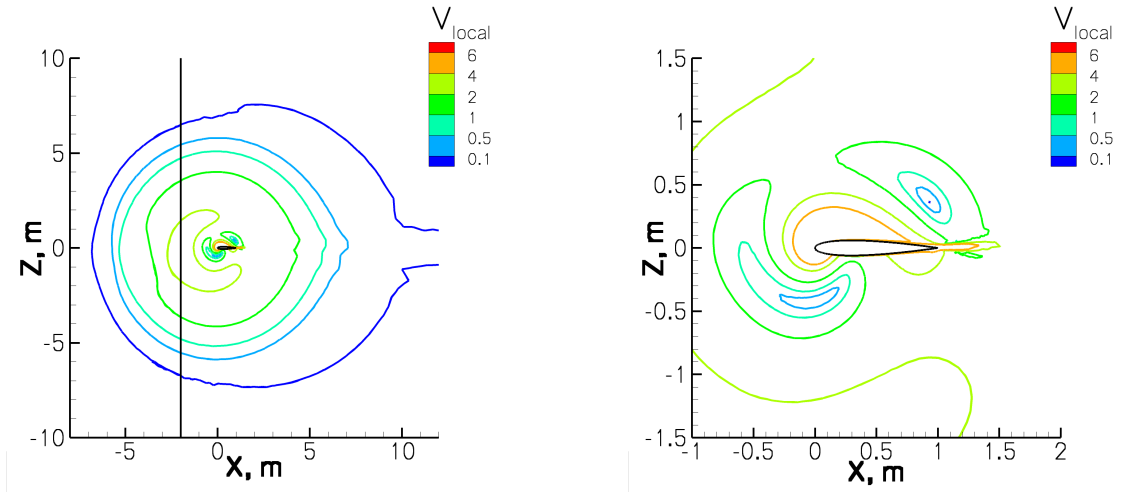


Figure 7.8: Local velocity at $t = 1s$ in example 7.3.

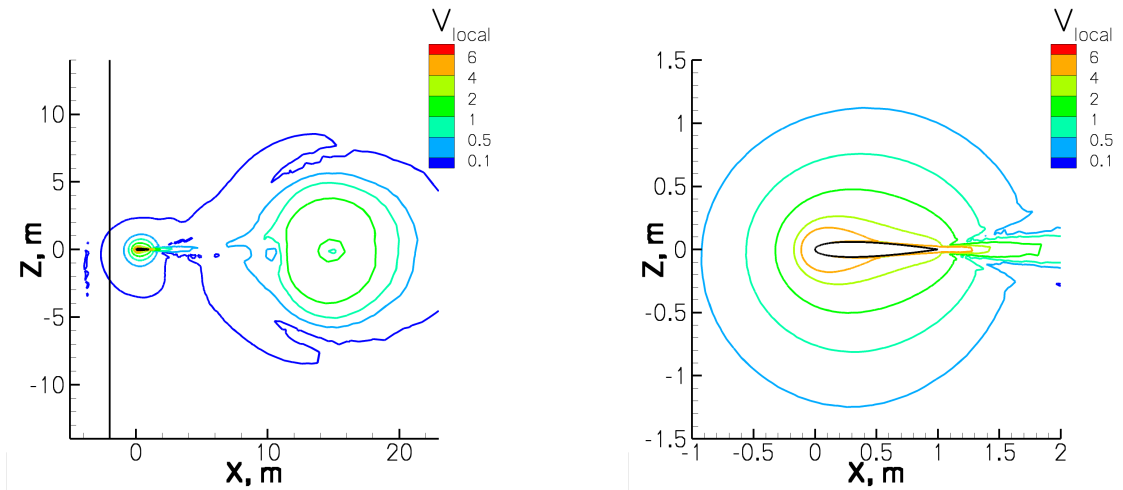


Figure 7.9: Local velocity at $t = 1.3s$ in example 7.3.

The computation is continued until 1.3s of physical time have passed. Figure 7.9 (left) shows that the vortex core, which is located approximately 15m to the right of the airfoil, has reunited. Although the shape of the vortex has been mostly recovered, the magnitude of v_{local} has noticeably decreased. Comparing the flow

around the airfoil before (see Fig. 7.7 (right)) and after the interaction of the vortex and the airfoil (see Fig. 7.9 (right)), one can see that both velocity distributions look very similar. This indicates that the flow around the airfoil has normalised again after 1.3s. As continuing the computation would move the vortex onto cells too coarse to reasonably resolve it, the simulation is stopped at this stage.

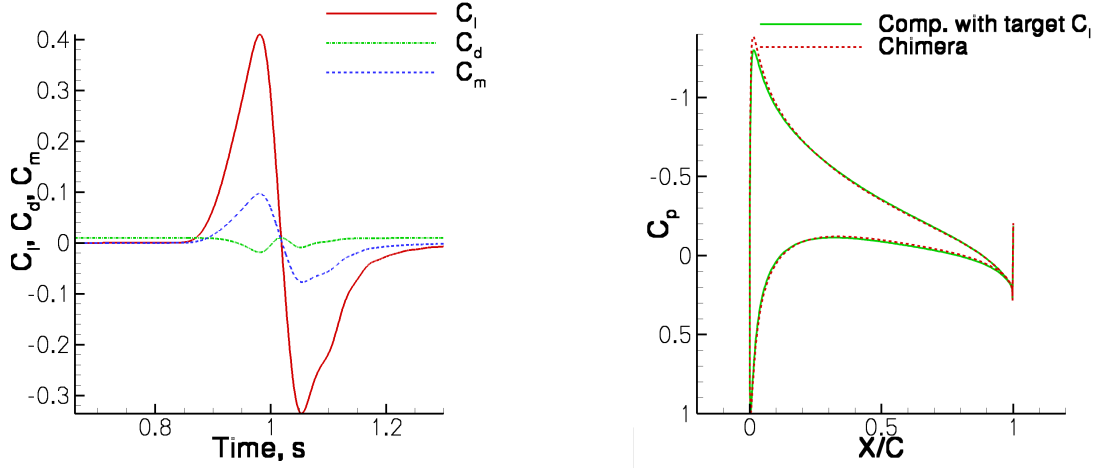


Figure 7.10: Time history of force, drag and moment coefficients (left) and comparison of C_p distributions (right) in example 7.3.

Figure 7.10 (left) shows the time history of C_l , C_d and C_m . The results can be interpreted as a change in the effective incidence angle. During the first half period the effective α is increased, which results in positive values of C_l and C_m . Then α is effectively decreased in the second half period, leading to negative C_l and C_m -values. At $t = 0.981$ s the lift coefficient reaches its maximal value of $C_l = 0.4097$ and the associated moment coefficient equals $C_m = 0.0966$.

Comparison to steady computation and flapping wing simulation

At this point it is interesting to study which values of α and C_m correspond to a C_l -value of 0.4097 in the case of undisturbed onflow conditions. For this purpose the first part of example 7.3 – a steady computation without vortex initialisation – is performed again with the difference that a target C_l -value of 0.4097 is prescribed. Starting with $\alpha = 0^\circ$, the angle of attack is then iteratively increased by rotating the wing upwards around the quarter point until the target C_l -value has been obtained. The resulting values are $\alpha = 3.704^\circ$ and $C_m = 0.1012$, which is a good agreement with the C_m -value in example 7.3. Another analogy of both computations can be seen when looking at the pressure distributions, which are shown in Fig. 7.10 (right). The solid lines belong to the computation with a prescribed C_l -value of 0.4097, whereas the dashed lines correspond to example 7.3 at $t = 0.981$ s. Except at the leading edge of the airfoil, both curves almost coincide.

Finally a simulation of a flapping NACA 0012 airfoil is performed with an amplitude of the pitching motion of $\Delta\alpha = 3.704^\circ$ and a period of 0.25s. Figure 7.11, which

shows the C_l time histories of the Chimera simulation and of the flapping airfoil, indicates that the two curves vary by about 25% in magnitude and that the flapping wing computation results in a phase transition compared to the Chimera outcome. From an engineering point-of-view the flapping wing simulation can be regarded as satisfactory because of the simpler computational setup and the less computational time required. In contrast, the Chimera simulation is more complex and time-consuming, but also leads to more reliable and detailed results.

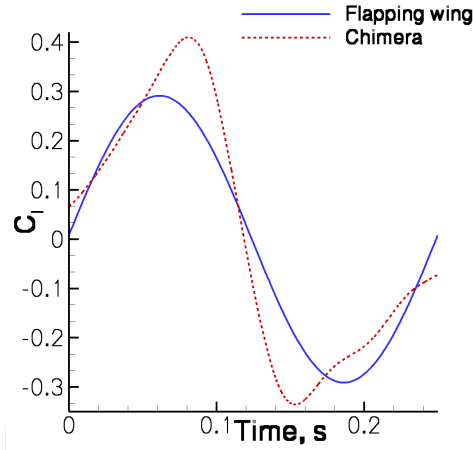


Figure 7.11: Time history of lift coefficient of Chimera simulation and of flapping wing computation in Section 7.3.

7.4 Vortex interaction with an ONERA-A airfoil at $\alpha = 13.3^\circ$

In this final example within this section the collision of a vortex and an ONERA-A airfoil with chord length $c = 1m$ at $\alpha = 13.3^\circ$ is simulated. At this angle of attack the airfoil is already close to stall even at undisturbed onflow conditions. The subsequent collision with a vortex is therefore expected to result in stall. The setting is described by $Re = 2.0 \times 10^6$, $Ma = 0.15$, $v_{\infty,x} = v_{vg,x} = 51.5m/s$ and $\Delta t = 2.5 \cdot 10^{-4}s$. The finite vortex is defined by exactly the same properties as in example 7.3 and also an equidistant vortex grid is chosen here. The background grid constitutes of 38668 grid points. As the previous examples have shown that the vortex transport over a large distance does not have a significant impact on the simulation, the distance is shortened in order to reduce CPU-time. The computation is therefore only performed for 0.42s.

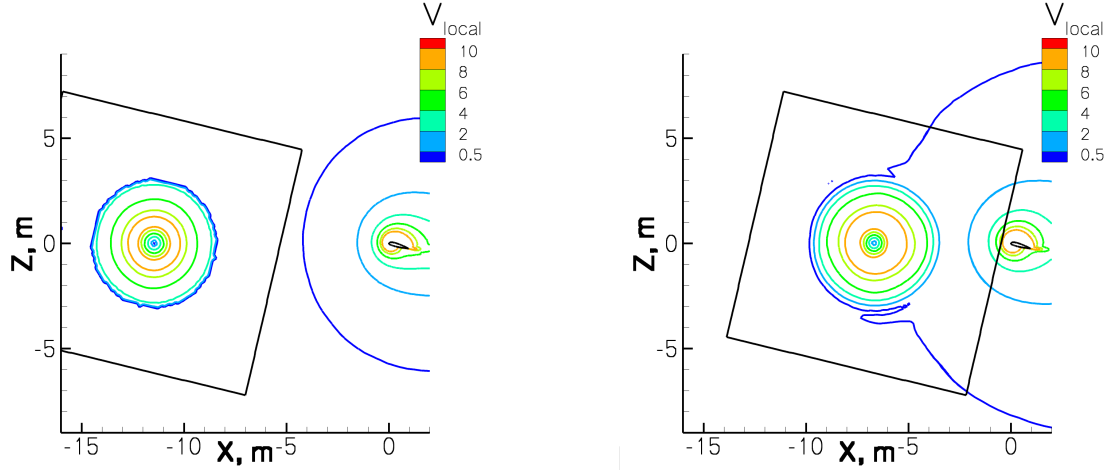


Figure 7.12: Vortex grid and v_{local} at $t = 0s$ (left) and $t = 0.16s$ (right) in example 7.4.

Figures 7.12 (left) and (right) show the relative position of the vortex grid, the vortex and the airfoil at $t = 0s$ and $t = 0.16s$, respectively. The vortex grid is halted then. Figure 7.13 (left) shows v_{local} after 0.3s of physical time has elapsed. The vortex has been mostly interpolated onto the background grid and is intensely interacting with the airfoil flow. As after 0.42s the vortex has passed the airfoil (see Fig. 7.13 (right)), the computation is stopped.

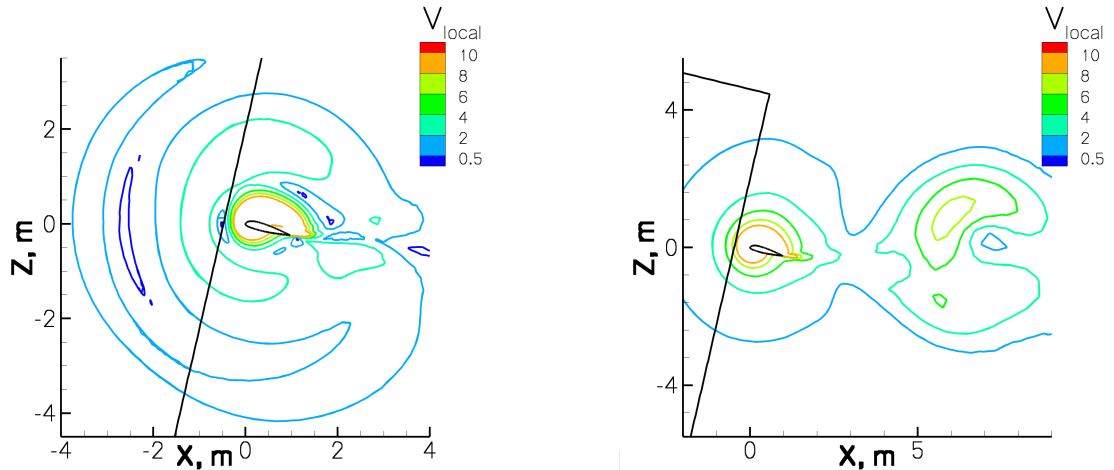


Figure 7.13: Vortex grid and v_{local} at $t = 0.3s$ (left) and $t = 0.42s$ (right) in example 7.4.

Contrary to the previous example, flow separation occurs at the trailing edge of the airfoil due to the vortex interaction. This is indicated in Fig. 7.14 (left), which shows the pressure distribution at $t = 0.3s$. Figure 7.14 (right), which illustrates the time history of the force and moment coefficients, shows that the vortex interaction can again be interpreted as a change of the effective incidence angle.

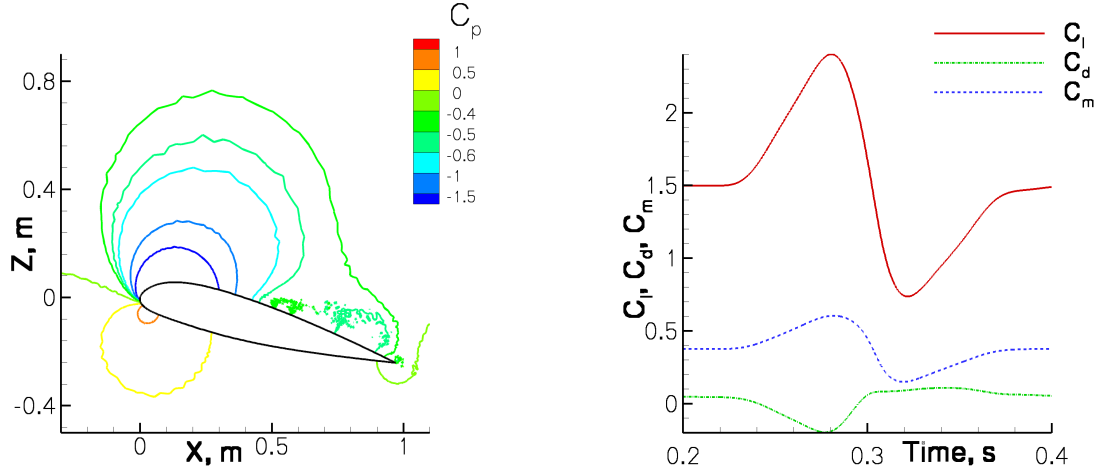


Figure 7.14: Pressure distribution at $t = 0.3s$ (left) and time history of force and moment coefficients (right) in example 7.4.

Summary of the chapter and implications on the simulation method

As conclusion of Chapter 7 we have shown in examples 7.2.1-7.2.3 that the Chimera technique minimises the numerical dissipation of transported vortices, even in the case of a coarse background grid or the simultaneous transport of more than one vortex. The subsequent vortex-airfoil interactions in Sections 7.3 and 7.4 performed problem-free and in both cases a large impact on C_l could be verified due to the collision of the vortex. In contrast to example 7.3, flow separation at the trailing edge could be observed in example 7.4 due to the vortex interaction. Both the NACA 0012 and the ONERA-A testcase have shown that the vortex-airfoil interaction can be interpreted as a change of the effective angle of attack. While the computation of a flapping NACA 0012 airfoil at the end of Section 7.3 was capable of predicting the correct trend of a vortex-airfoil interaction, the Chimera simulation in Section 7.3 lead to more accurate results. Overall, this proves that the Chimera technique is essential for the regarded applications.

Based on the results in this chapter, the use of the Chimera technique can be strongly recommended for the applications in Chapter 8.

Chapter 8

Applications of the simulation method

The aim of this last chapter of the numerical applications is to combine both the SA-DDES model and the Chimera technique to simulate the interaction of a realistic vortex with a two-element FNG(“Flügel neuer Generation”–“new generation airfoil”) airfoil near stall.

The principle is illustrated in Fig. 8.1: Situated within a wind tunnel grid (“orange”) lies a NACA0021-airfoil grid (“red”) just behind the inflow boundary. The wing is flapped upwards, resulting in a realistic vortex evolving from the trailing edge, which is interpolated onto a fine equidistant cartesian vortex transport grid (“blue”). Both the vortex and the vortex transport grid are then moved with the freestream velocity towards the FNG grid (“green”). Just before reaching the latter the vortex transport grid is stopped and the vortex is interpolated onto the FNG grid, where the interaction can be studied. As the incidence angle of the FNG airfoil before the arrival of the vortex is already chosen close to stall, the effect of the vortex-FNG interaction is expected to result in detectable flow separation at the FNG airfoil. The three-dimensional setup shown in Fig. 8.1 is used both in Section 8.3 and in Section 8.4, where the SA-URANS and alternatively SA-DDES model are applied.

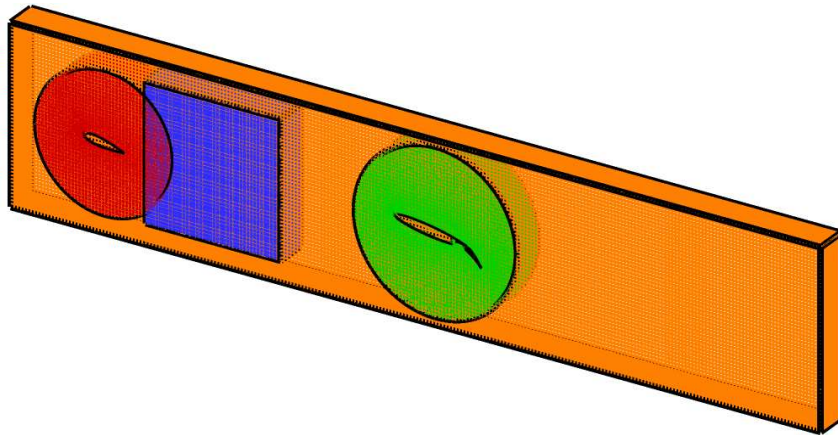


Figure 8.1: Three-dimensional setup including grids to create and transport a realistic vortex and to simulate its interaction with a two-element airfoil near stall.

Beforehand, several preexaminations are presented in this chapter:

In Section 8.1 steady two-dimensional RANS computations of an FNG airfoil situated in a wind tunnel are shown. The simulations have been performed at various angles of attack at undisturbed onflow conditions and both the SAO model and the Menter-SST model have been applied. The aim is twofold: Firstly the influence of the turbulence model is examined and secondly the angle of attack leading to stall is detected. Additionally, a grid convergence study is performed.

In Section 8.2 two-dimensional URANS simulations to generate and transport a realistic vortex and to simulate the vortex-FNG interaction are shown, where the principle is identical to the one just described. Again, both the SAO model and the Menter-SST model have been applied. Additionally, both a small and a large airfoil-to-airfoil distance (defined as distance between the trailing edge of the NACA0021 airfoil and the leading edge of the FNG airfoil) of 2m and alternatively 4m has been used, resulting in an overall of four simulations. Besides identifying the most promising numerical setting for the three-dimensional simulations performed in Section 8.3 and 8.4, the major goal of Section 8.2 is to check whether stall occurs due to the vortex-airfoil interaction.

In Fig. 8.2 and Fig. 8.3 two-dimensional cutouts of the four Chimera grids used in this chapter are shown. For better comparison of the varying grid refinements, all cutouts are of an equal size of $0.4m \times 0.4m$. The NACA0021 grid (containing approximately 33000 grid points per two-dimensional plane) and the FNG grid (45000 points) cover a circular area of radius 0.5m and respectively 0.6m and are of hybrid type. The two airfoils are positioned in the center of their relative grid, where the NACA0021 airfoil has a chord length of 0.3m and the FNG airfoil one of 0.6m. (The chord length of the FNG airfoil corresponds to cruise configuration, whereas in this chapter take-off configuration is considered. Therefore the actual length of the FNG airfoil is larger than its chord length.) While the quadratic vortex transport grid ($151^2 \approx 23000$ points) of size $1m \times 1m$ looks relatively coarse in comparison, it is considered fine enough to conserve the vortex, because the grid is equidistant and cartesian and the transport distance is much shorter than in all previous simulations shown in Chapter 7. The three Chimera grids lie embedded in the coarse wind tunnel grid (12000 points), which is also equidistant and cartesian and covers an area of $1.3m \times 6m$. Overall a two-dimensional plane containing all four Chimera grids (as is used in Section 8.2) contains approximately 114000 grid points.

In all simulations in this chapter $Re = 2.0 \times 10^6$ (with respect to $c=0.6m$) and $Ma = 0.15$ have been used. At the in- and outflow of the wind tunnel the farfield boundary condition is applied, while at the upper and lower end of the wind tunnel inviscid walls are utilised, thus neglecting the viscous effects of the wind tunnel walls. At the two airfoils viscous walls are chosen. For the three-dimensional computations a two-dimensional plane containing the four Chimera grids has been uniformly extruded in spanwise direction with an extent of 40 % chord length using 49 two-dimensional planes, resulting in $5.6 \cdot 10^6$ grid points. As the periodic plane condition can not be combined with the Chimera technique in the DLR TAU-code, the symmetry plane condition has been used instead for the boundaries in spanwise direction.

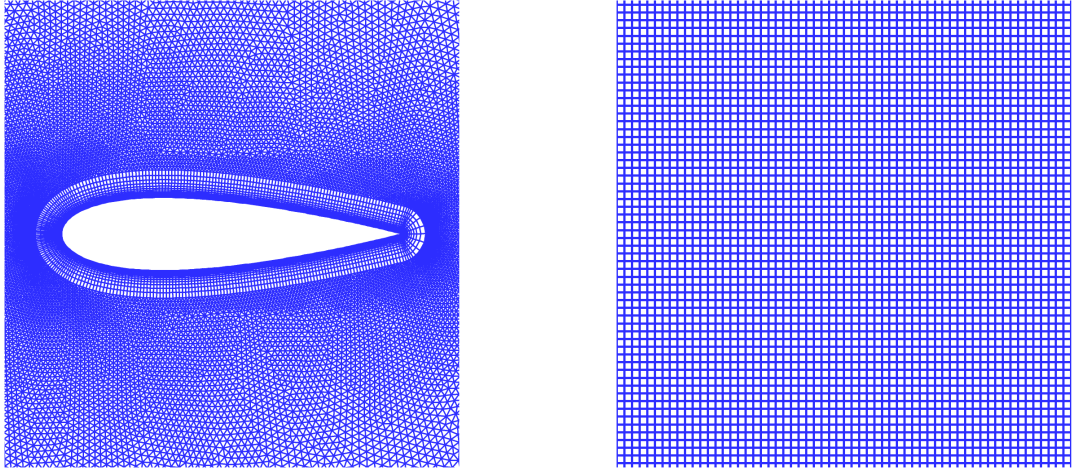


Figure 8.2: Cutout of size $0.4m \times 0.4m$ of NACA0021 grid (left) and of vortex transport grid (right)

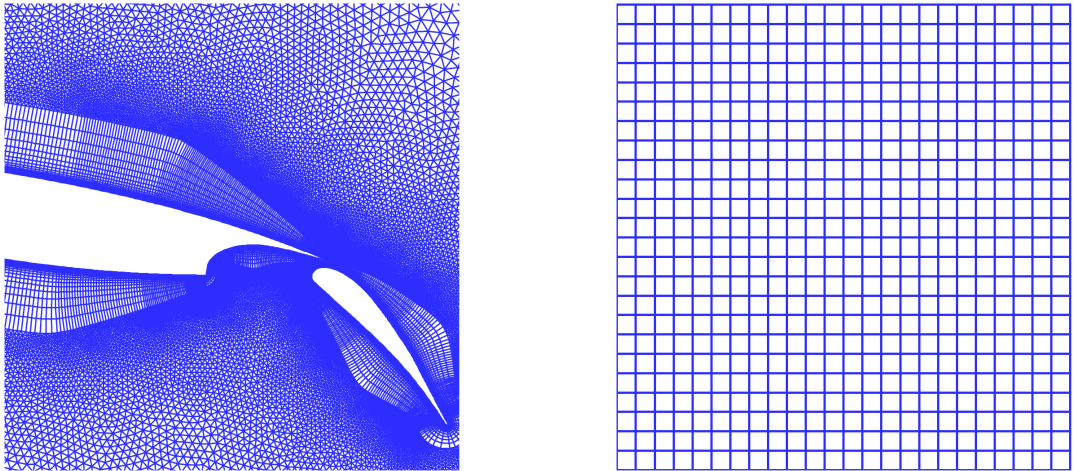


Figure 8.3: Cutout of size $0.4m \times 0.4m$ of FNG grid (left) and of wind tunnel grid (right)

8.1 Two-dimensional (U)RANS simulations of an FNG airfoil at undisturbed onflow conditions

The aim of this section is to determine the angle of attack leading to stall at the FNG airfoil at undisturbed onflow conditions. Based on this information an angle

just before stall is chosen in Sections 8.2- 8.4, such that the subsequent vortex-airfoil interaction should lead to massive flow separation at the FNG airfoil. Additionally, the effect of different turbulence models is examined in this section by using both the SAO and the Menter-SST model.

Figure 8.4 shows the numerical setup utilised in this section, where the origin regarding the x-z coordinates is set at the lower left wind tunnel corner. At $\alpha = 0^\circ$ the left boundary of the FNG grid is located at an x-position of 2.85m and the lower boundary at a z-position of 0.05m. All simulations have been performed stationary and both multigrid and low Mach number preconditioning have been used.

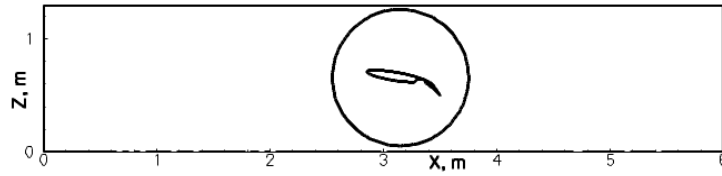


Figure 8.4: Numerical setup used in Section 8.1

Figures 8.5–8.9 show the pressure distributions of both the SAO and the Menter-SST model at various angles of attack, where the results at the wing and the flap are colored differently to enable distinction. Besides slight differences in the height of the suction peak, in the levels on the upper wing surface and the behaviour on the upper side of the flap, both turbulence models result in very similar C_p -distributions in the range of $\alpha = 0^\circ - 8^\circ$. The differences of the two models become more significant at 10° and 11° . At 12° the Menter-SST model predicts complete flow separation on the wing and also looks somewhat unsteady on the upper flap side, where it must be stated that this computation was the only one in this section that did not fully converge to a steady state solution. In contrast, the SAO model still results in fully attached flow at 12° . To determine the incidence angle leading to stall when using SAO, α is further increased to 13° . This finally results in stall as can be seen in Fig. 8.9.

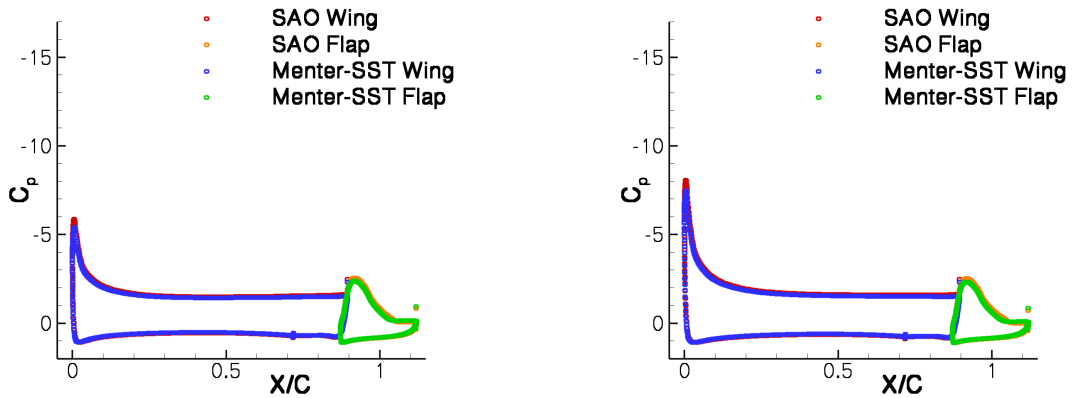


Figure 8.5: Pressure distributions of the SAO model and the Menter-SST model at $\alpha = 0^\circ$ (left) and $\alpha = 2^\circ$ (right).

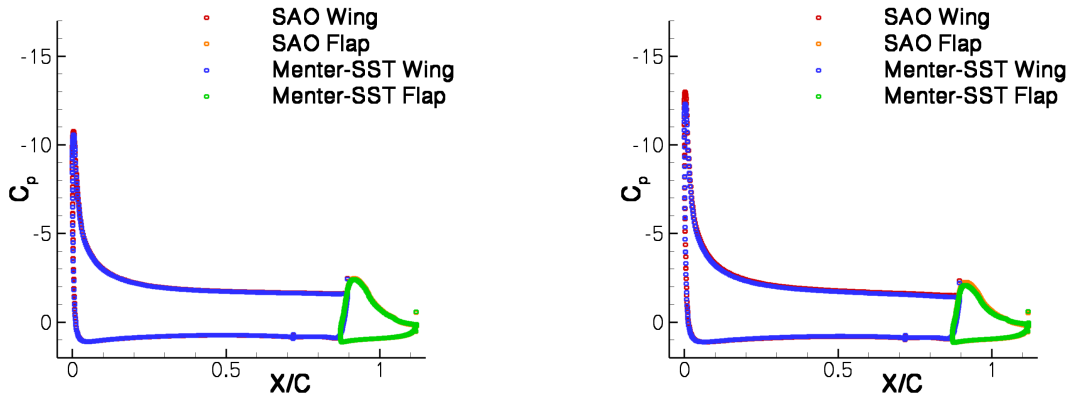


Figure 8.6: Pressure distributions of the SAO model and the Menter-SST model at $\alpha = 4^\circ$ (left) and $\alpha = 6^\circ$ (right).

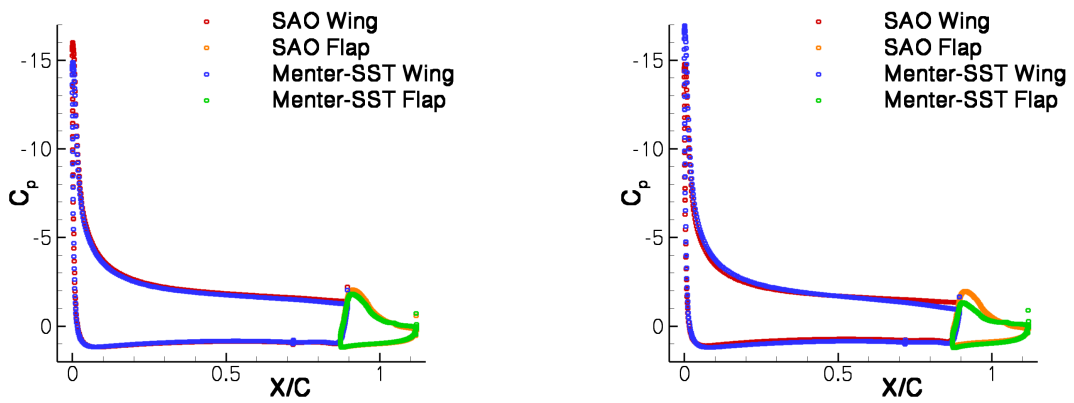


Figure 8.7: Pressure distributions of the SAO model and the Menter-SST model at $\alpha = 8^\circ$ (left) and $\alpha = 10^\circ$ (right).

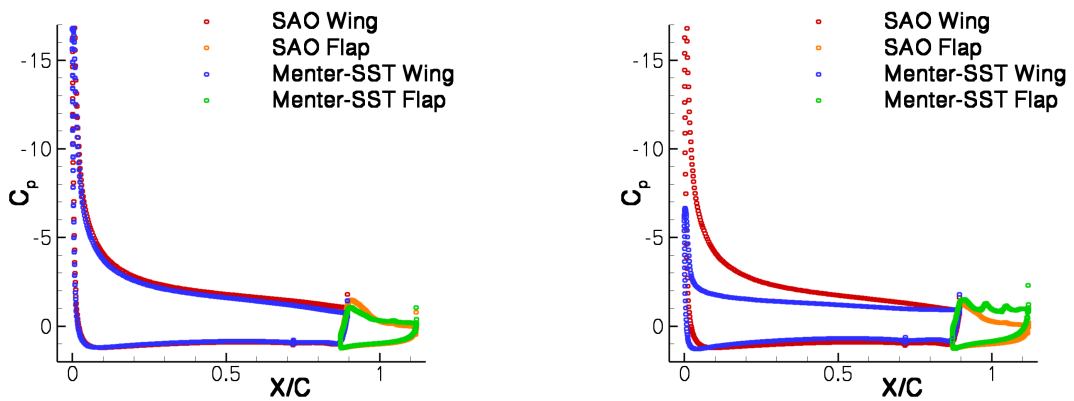


Figure 8.8: Pressure distributions of the SAO model and the Menter-SST model at $\alpha = 11^\circ$ (left) and $\alpha = 12^\circ$ (right).

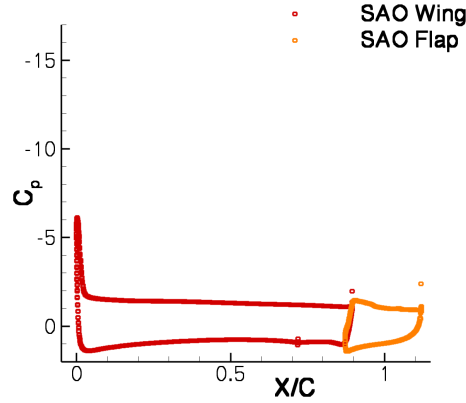


Figure 8.9: Pressure distribution of the SAO model at $\alpha = 13^\circ$.

To further investigate the phenomena occurring at stall, Fig. 8.10 shows the skin friction coefficients of both models in the case of their relative "stall" angles. Independently of the model the wing completely detaches, while the flow at the flap remains attached.

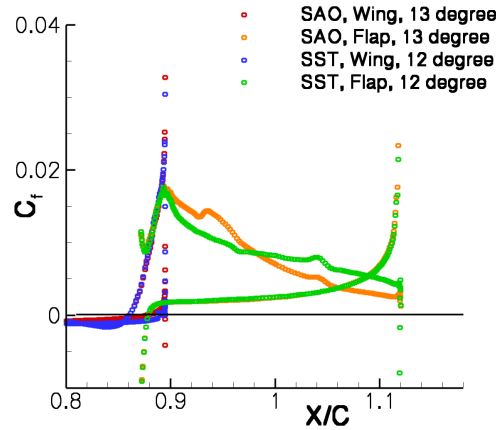


Figure 8.10: Skin friction distributions of the SAO model and the Menter-SST model at the angles of attack leading to stall.

Figure 8.11, which shows the lift coefficients of the two models at all investigated angles of attack, supports the observations made from the pressure and skin friction distributions. Despite slight differences in the level of C_l , both models show a similar behaviour in the range of $\alpha = 0^\circ - 8^\circ$. At higher angles of attack the differences become more obvious and $C_{l,max}$ is reached at $\alpha = 8^\circ$ when using the Menter-SST model, whereas only at $\alpha = 10^\circ$ with the SAO model. Due to stall a large lift breakdown can be observed at 12° with the Menter-SST model and at 13° with SAO.

Based on the results obtained so far we choose – independently of the turbulence model – an angle of attack of 10° before the arrival of the vortex in Sections 8.2- 8.4. This incidence angle is considered close enough to stall, such that the vortex-airfoil

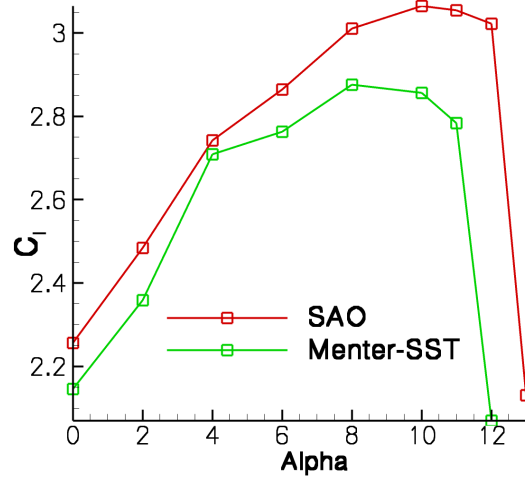


Figure 8.11: Comparison of the lift coefficients in the stationary RANS simulations.

interaction should result in massive flow separation. It is also easier to compare the performance of both models, if the same original angle is selected.

Grid convergence study

When using a converged steady solution as restart for a subsequent simulation utilising dual time stepping, it is expected that the solution and therefore also C_l does not change in time. Surprisingly, this proved to be wrong in this testcase for $\alpha = 10^\circ$ when using a time step size of $10^{-4}s$. Independently of the turbulence model, strong C_l -oscillations occurred almost immediately after switching to dual time stepping. The resulting C_l based on SAO is shown as solid black line ("Coarse w. Pr." in the legend) in Fig. 8.12.

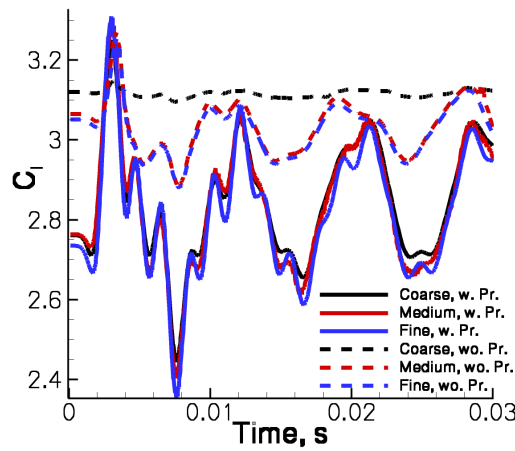


Figure 8.12: Lift coefficients of various unsteady SAO simulations at $\alpha = 10^\circ$ in the grid convergence study.

This unsteadiness causes serious problems when performing simulations including a vortex-airfoil interaction, because the C_l -oscillations superimpose the effect of the vortex. One should therefore aim to dampen C_l , but only if the unsteadiness is due to numerical reasons. If on the other hand the C_l -oscillations are based on an actual physical phenomenon, one has to accept C_l as it is. The rest of this section is devoted to examine this question, whereas it is obvious that a solid proof of such a physical phenomenon can only be provided by performing wind tunnel experiments and not by means of numerical simulations. Nevertheless, a numerical investigation can still give hints regarding the nature of the C_l -oscillations by systematically investigating all possible numerical reasons causing the unsteadiness. As both turbulence models result in similar magnitudes and periods regarding the C_l -oscillations, only the SAO model will be used from this point on.

The following variations in the numerical setup had no significant impact on C_l :

- Instead of inviscid walls at the upper and lower ends of the wind tunnel grid the farfield boundary condition was chosen to check whether reflections at these walls are the source of the unsteadiness.
- The time step size was reduced to $10^{-5}s$.
- The angle of attack was varied to 2° and 6° .
- To check, whether reflections from the outflow boundary are responsible, the outflow boundary condition was changed to "exit pressure outflow".
- To furthermore exclude the outflow boundary as source of the oscillations, a simulation using a wind tunnel grid enlarged by 100m to the right was performed. If the outer boundary, where again farfield was utilised, was the cause of the unsteadiness, the period of the oscillations should have increased.

One parameter that did turn out to have a huge impact on C_l was low Mach number preconditioning (see Subsection 2.4).

While all computations performed to this stage included preconditioning, the dashed black line (indicated as "Coarse wo. Pr.") in Fig. 8.12 shows the behaviour of C_l when this parameter is switched off. As restart a steady-state solution without preconditioning was used at $t = 0s$. While still not resulting in a constant C_l -value, the oscillations have decreased immensely.

How can this strong effect of low Mach number preconditioning be explained? It is well-known that the use of preconditioning reduces the numerical dissipation. Turning off preconditioning may therefore lead to a dissipation of the flow structures responsible for the C_l -oscillations.

To further investigate this assumption, a grid convergence study of the FNG grid has been performed. As the magnitude of numerical dissipation decreases with finer grids, also the two C_l -curves should coincide the better the finer the grid when comparing the solutions with and without using preconditioning.

Based on the so-far used "coarse" FNG grid (45000 points per two-dimensional plane) a medium and a fine grid have been built by performing the following refinements:

- While the coarse grid contains 30 structured prismatical layers at the body, both the medium and the fine grid comprise 45. As additionally the stretching ratio in wall-normal direction was drastically reduced, the domain covered by the structured grid part is nevertheless smaller than in the coarse grid.
- The number of surface points has been increased, resulting in a higher resolution in streamwise direction.
- The unstructured part in the area starting from the middle of the main wing to just behind the flap was refined both below and above the body, because this is the region where instationarities are expected to evolve.

A two-dimensional plane of the medium FNG grid contains 84000 grid points, while the fine grid is made up of 128000 points. Figure 8.13 shows the same cutouts of the medium and respectively fine grid that was already illustrated in Fig. 8.3 (left) for the coarse grid. Especially the reduced domain size of the structured parts and the refined unstructured areas are clearly visible.

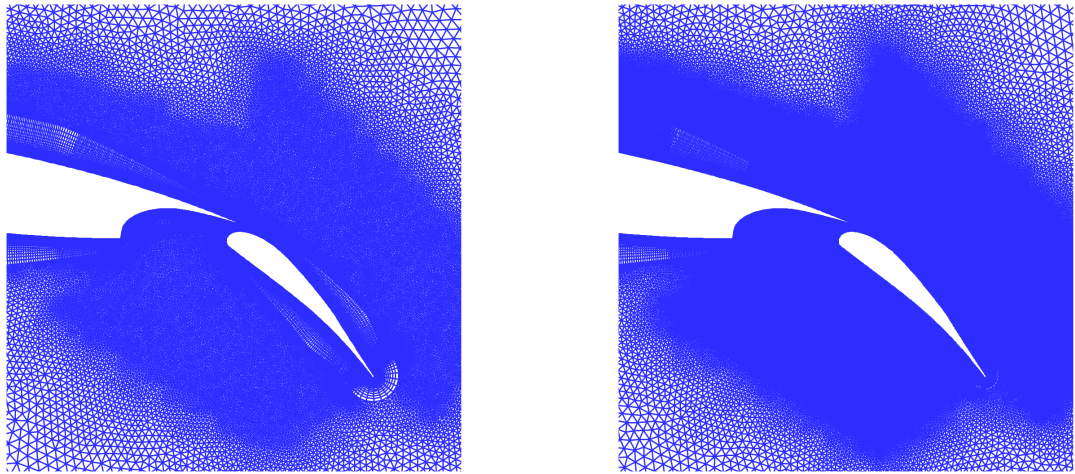


Figure 8.13: Cutout of size $0.4m \times 0.4m$ of the medium FNG grid (left) and of the fine FNG grid (right) used in the grid convergence study.

The position of both the medium and the fine FNG grid relative to the wind tunnel grid is identical to the one shown in Fig. 8.4. In all computations the incidence angle of the FNG airfoil is 10° . Due to the use of three grids and utilising or alternatively switching off low Mach number preconditioning, a total of six simulations was performed. Before changing to dual time-stepping, a steady-state restart solution was computed at the beginning of all simulations.

Figure 8.12 shows the resulting C_l -distributions, which look surprising at first glance. The three solid lines, which are based on the results utilising preconditioning, almost coincide and show a large variation in C_l . As already mentioned above, the almost

constant dashed black curve is the result of the coarse grid when preconditioning is switched off. The dashed red and blue curve, which are based on the computations on the medium and respectively fine grid without usage of preconditioning, are almost identical and vary stronger than the dashed black line but less than the three solid curves.

The results are interpreted as follows:

- As the computation on the fine grid with preconditioning switched on should be the most accurate one, the C_l -oscillations are considered numerically correct and should not be damped out.
- When using low Mach number preconditioning, the coarse grid is sufficient.
- The computation on the coarse grid without utilising preconditioning is too dissipative.

Nevertheless, the simulations in Section 8.2 are performed utilising the coarse grid with preconditioning switched off. The reason for deactivating preconditioning is, that these two-dimensional simulations should be considered as preexaminations for the forthcoming three-dimensional computations and not as valid simulations aiming to accurately predict the vortex-airfoil interaction. The main goal is hereby to determine, whether the interaction does have a significant effect on C_l and possibly leads to stall. This is easier to examine when the C_l -curve does not oscillate before the arrival of the vortex; therefore preconditioning is switched off.

In contrast, the three-dimensional computations shown in Section 8.3 and 8.4 are performed using the coarse FNG grid and low Mach number preconditioning activated. This choice is considered as both sufficiently accurate and numerically cheap.

8.2 Two-dimensional URANS simulations

In this section two-dimensional URANS simulations to generate and transport a realistic vortex and to simulate the vortex-FNG interaction are shown, where both the SAO model and the Menter-SST model have been applied. Both a small and a large airfoil-to-airfoil distance of 2m and alternatively 4m has been used, resulting in an overall of four simulations.

Figure 8.14 and Fig. 8.15 show the small and respectively large numerical setup. The origin of the x-z coordinates is set at the lower left wind tunnel corner. At the beginning of all simulations the left boundary of the NACA0021 grid of radius 0.5m is located at an x-position of 0.25m and the lower one at a z-position of 0.15m, where originally $\alpha = 0^\circ$ is chosen at the NACA0021 airfoil. The left edge of the quadratic vortex transport grid of size 1m is located at x=1m, while the lower edge is situated at z=0.15m. The only difference between the small and the large setup is due to the location of the FNG grid of radius 0.6m. Based on $\alpha = 0^\circ$ the lower boundary of the FNG grid is situated at z=0.05m in both setups, while the left boundary is located at x=2.55m in the small setup and at x=4.55m in the large setup. The incidence angle of the FNG airfoil is held constant at $\alpha = 10^\circ$

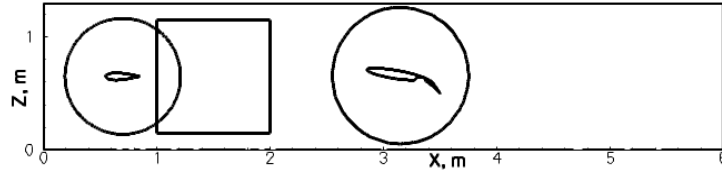


Figure 8.14: Numerical setup including the small airfoil-to-airfoil distance of 2m.

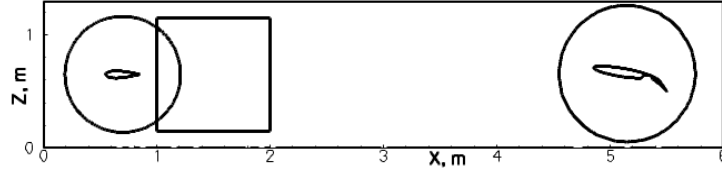


Figure 8.15: Numerical setup including the large airfoil-to-airfoil distance of 4m.

throughout all simulations. As already mentioned at the beginning of this chapter, a two-dimensional plane including all four Chimera grids contains 114000 grid points. For convergence acceleration, multigrid is applied. During the upward rotation of the NACA0021 airfoil a time step size of $\Delta t = 1.5 \cdot 10^{-5}$ s is chosen; afterwards it is increased to $\Delta t = 5 \cdot 10^{-5}$ s.

One goal of this section is to identify the most promising numerical setting regarding the turbulence model and the airfoil-to-airfoil distance for the three-dimensional simulations performed in Sections 8.3 and 8.4.

The second aim is to check whether the vortex-airfoil interaction has a significant effect on C_l and leads to stall. As the vortex impact on C_l is easier to examine when C_l does not oscillate, low Mach number preconditioning is switched off, even though this results in less exact solutions. The computations shown in this section should therefore be regarded as preliminary investigations, while only the three-dimensional results in Sections 8.3 and 8.4 aim to obtain accurate physical solutions.

Figures 8.16–8.22 show the vorticity based on the SAO-computations at various times of the simulations, where on the left the small numerical setup is considered, while the figures on the right are based on the large numerical setup. As the vorticity looks almost identical with Menter-SST, only the SAO-results are shown.

Before switching to dual time stepping, steady restart solutions have been performed at the beginning of all four simulations. Figure 8.16 shows the vorticity on cutouts of the small and relatively large setup based on these restart solutions at $t=0$ s. Focusing on the area of the NACA0021 grid and the vortex transport grid, both results look almost identical. At the right boundary of the vortex transport grid the vorticity vanishes, which is a hint that the wind tunnel grid is too coarse to conserve it, while both the NACA0021 grid and the vortex transport grid are sufficiently fine.

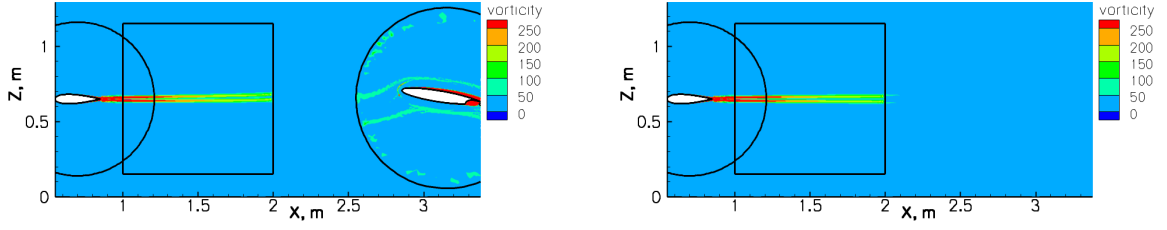


Figure 8.16: Vorticity based on SAO computations at $t = 0s$ in the small numerical setup (left) and in the large numerical setup (right).

Next the NACA0021 airfoil is rotated 10° upwards within $1.5 \cdot 10^{-2}s$. The resulting vorticity plots are shown in Fig. 8.17. In both simulations the vortex has evolved from the trailing edge and has already been interpolated partly onto the vortex transport grid. Furthermore, the vortex has started to split up into several subvortices. This effect is caused by discretisation errors including the ones due to the Chimera interpolation. In physics this phenomenon can also be observed behind an airfoil, where it emerges due to geometrical unevenness.

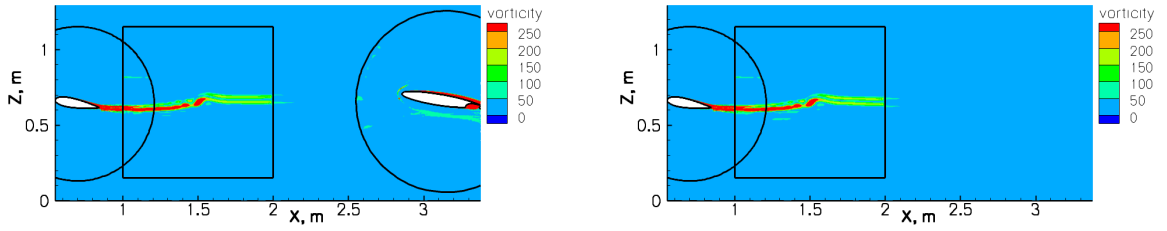


Figure 8.17: Vorticity based on SAO computations at $t = 1.5 \cdot 10^{-2}s$ in the small numerical setup (left) and in the large numerical setup (right).

After $1.9 \cdot 10^{-2}s$ the vortex has been interpolated onto the vortex transport grid, which is shown in Fig. 8.18. Now the vortex transport grid is moved with the freestream velocity of $50 \frac{m}{s}$ to the right.

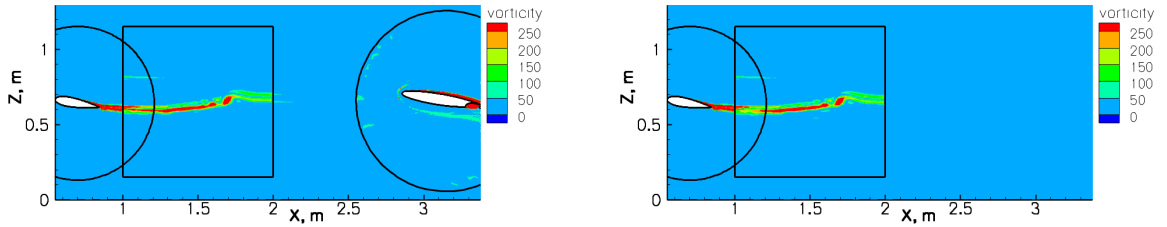


Figure 8.18: Vorticity based on SAO computations at $t = 1.9 \cdot 10^{-2}s$ in the small numerical setup (left) and in the large numerical setup (right).

At $t = 2.5 \cdot 10^{-2}s$ the vortex transport grid has moved 0.3m towards the FNG airfoil, which is indicated in Fig. 8.19. The results based on the small and the large setup still look identical and the vortex has continued to split into subvortices.

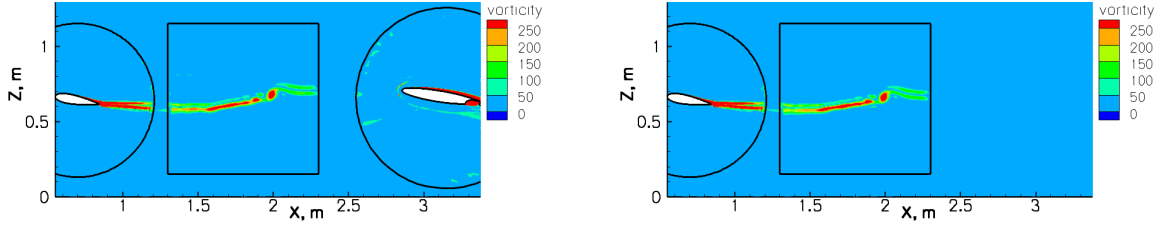


Figure 8.19: Vorticity based on SAO computations at $t = 2.5 \cdot 10^{-2}s$ in the small numerical setup (left) and in the large numerical setup (right).

The vortex transport grid is further transported to the right, until it reaches its final position at $t = 3.1 \cdot 10^{-2}s$ in the small setup and at $t = 7.1 \cdot 10^{-2}s$ in the large one. At this stage, which is indicated in Fig. 8.20, the vortex transport grid is stopped, while the vortex continues to move to the right. Due to the longer physical time passed in the simulation of the large setup, the vortex has dissipated more than in the small setup.

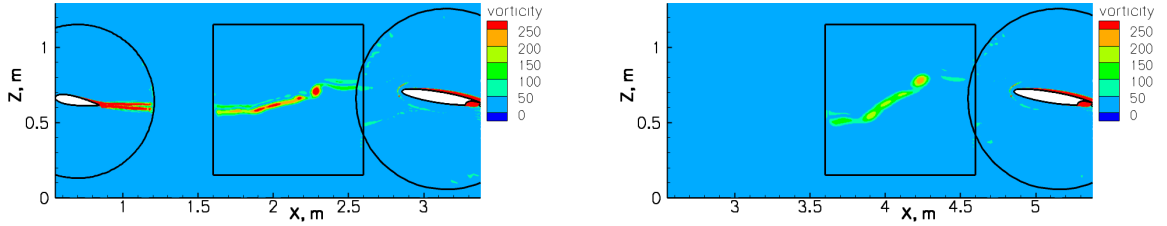


Figure 8.20: Vorticity based on SAO computations at $t = 3.1 \cdot 10^{-2}s$ in the small numerical setup (left) and at $t = 7.1 \cdot 10^{-2}s$ in the large numerical setup (right).

After $t = 3.9 \cdot 10^{-2}s$ and alternatively $t = 7.9 \cdot 10^{-2}s$ the vortex has been partly interpolated onto the FNG grid, which is shown in Fig. 8.21. While the vortex in the small setup still contains higher maximal vorticity values than in the large setup, it has also become weaker. One possible explanation for this relatively fast vortex decay might be, that the vortex is interpolated several times onto different Chimera grids. In combination with low Mach number preconditioning switched off, this can lead to an increase in dissipation.

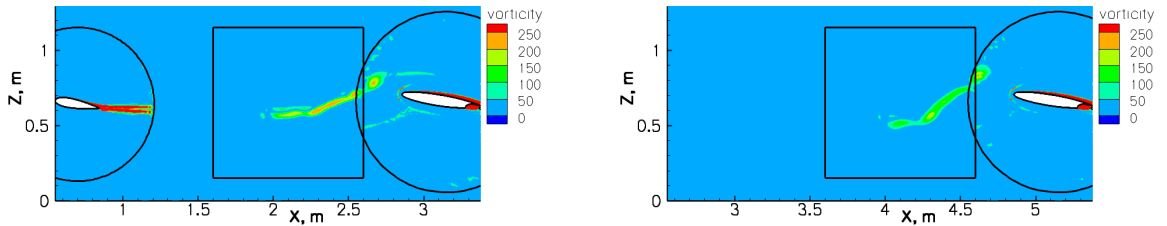


Figure 8.21: Vorticity based on SAO computations at $t = 3.9 \cdot 10^{-2}s$ in the small numerical setup (left) and at $t = 7.9 \cdot 10^{-2}s$ in the large numerical setup (right).

While the simulations were continued for an even longer time, Fig. 8.22 shows the vorticity plots at $t = 4.7 \cdot 10^{-2}s$ and respectively $t = 8.7 \cdot 10^{-2}s$. In both cases a main vortex has evolved at the right of the longitudinal vortex above the FNG airfoil.

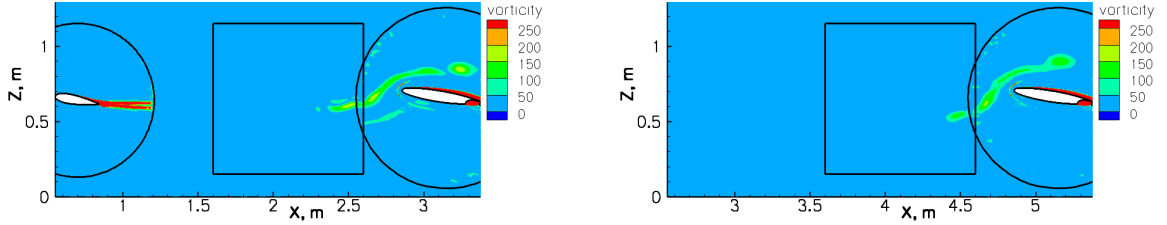


Figure 8.22: Vorticity based on SAO computations at $t = 4.7 \cdot 10^{-2}s$ in the small numerical setup (left) and at $t = 8.7 \cdot 10^{-2}s$ in the large numerical setup (right).

Judging by the vorticity plots, all simulations performed satisfactory, whereas the vortex dissipation seems to be more severe in the large setup due to the longer physical time.

Next the effect of the vortex-FNG interaction on C_l is examined. Figure 8.23 shows the C_l -distributions of all four computations, where in the left picture the curves are based on the two simulations on the small setup, while in Fig. 8.23 (right) the large setup is considered. Note the different time scales in both pictures.

The following observations can be made:

Firstly, the effect of the vortex-airfoil interaction on C_l is much stronger in the small setup, independently of the turbulence model. Only in the small setup a lift breakdown occurs.

Secondly, both turbulence models result in similar C_l -curves, but the values based on the SAO computations are larger by $\Delta C_l \approx 0.2$.

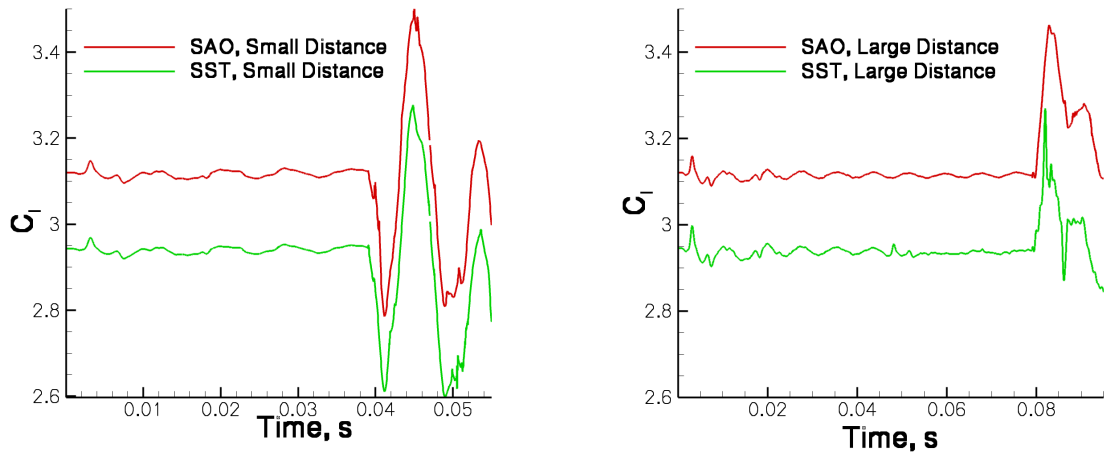


Figure 8.23: Time-dependent lift coefficients based on the SAO and Menter-SST models on the small numerical setup (left) and on the large numerical setup (right).

The final question examined in this section is whether stall occurs due to the vortex-airfoil interaction. This is obviously not the case in the two simulations using the large setup, because no lift breakdown can be observed.

As furthermore both turbulence models behave qualitatively similar, it is sufficient to look at the results obtained by SAO on the small setup. (The results of Menter-SST on the small setup were also examined and turned out to be similar to SAO.) Figure 8.24 (left) and (right) and Fig. 8.25 show C_f at $t = 4.1 \cdot 10^{-2}s$, $t = 4.5 \cdot 10^{-2}s$ and $t = 4.9 \cdot 10^{-2}s$. These are the points in time, when C_l reaches a local minimum, a local maximum and again a local minimum.

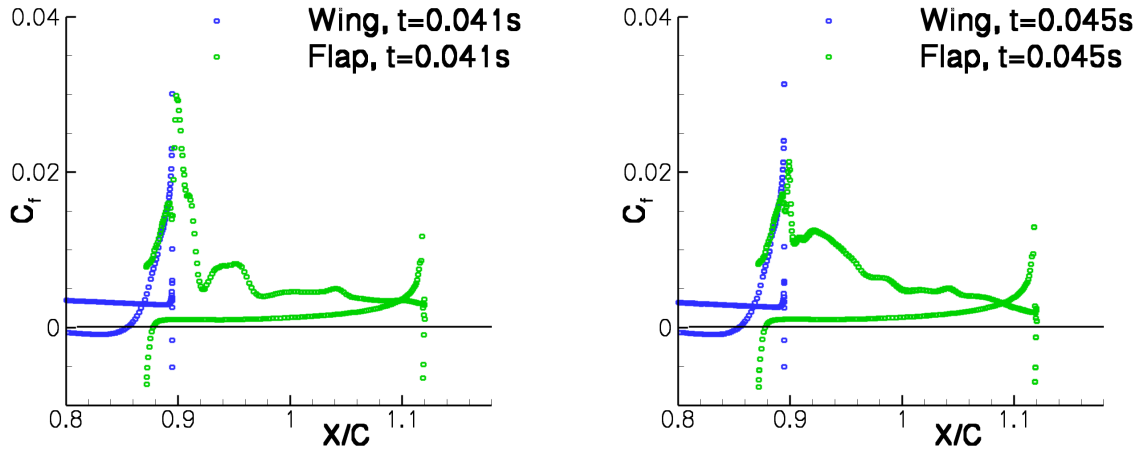


Figure 8.24: Skin friction coefficients based on SAO computation on the small numerical setup at $t = 4.1 \cdot 10^{-2}s$ (left) and $t = 4.5 \cdot 10^{-2}s$ (right).

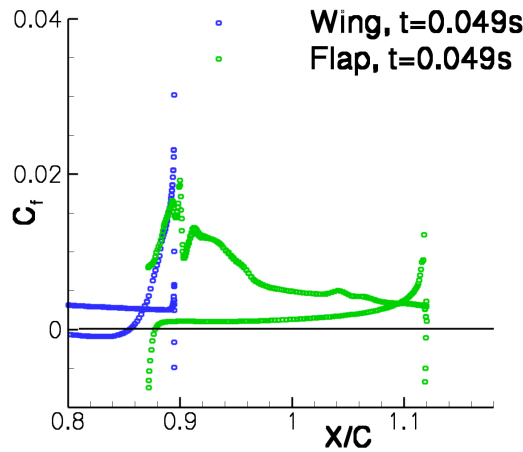


Figure 8.25: Skin friction coefficient based on SAO computation on the small numerical setup at $t = 4.9 \cdot 10^{-2}s$.

The flap remains fully attached in all cases. As additionally also the flow at the main wing does not detach, it is hereby proven that in this computation no stall occurs.

This is consistent with the SAO results shown in Section 8.1, where the "stall angle" was identified at $\alpha = 13$, causing a lift breakdown to $C_l = 2.1$ (compare with Fig. 8.23 (left)). In contrast, the vortex-airfoil interaction in the SAO simulation on the small setup only results in a much higher minimum C_l -value of 2.8, which clearly contradicts the occurrence of stall.

Based on the results in this section the small setup is chosen for the three-dimensional computations performed in Sections 8.3 and 8.4. As neither SAO nor Menter-SST proved to be superior, only the SAO model and SA-DDES will be used, because these models are numerically more stable and the author of this work is also more experienced in them.

8.3 A three-dimensional SA-URANS simulation

Based on the results in the previous sections a three-dimensional SA-URANS simulation is performed, where now low Mach number preconditioning is activated to minimise the numerical dissipation. The three-dimensional grid is obtained by uniformly extruding a two-dimensional plane of the small numerical setup (see Fig. 8.14) in spanwise direction with an extent of 40 % chord length using 49 two-dimensional planes, resulting in $5.6 \cdot 10^6$ grid points.

While the applied boundary conditions have been specified at the beginning of this chapter, all other data like time step size, Mach number, Reynolds number and exact procedure of the computation remain unchanged with regard to the simulations performed in Section 8.2.

Figures 8.26-8.29 show the results at various times, where in the relatively left picture isosurfaces of the vorticity at a value of 100 are presented, while the right pictures show isosurfaces of the instantaneous Q-criterion at a value of $100 \frac{1}{s^2}$ in the area of the FNG airfoil. The isosurfaces in the left pictures are additionally colored with the Mach number and the ones in the right pictures show the spanwise velocity. While the vorticity, the Q-criterion and the y-velocity contain important information and are examined in detail, the illustration of the Mach number is only added to show that this variable looks similar in the three-dimensional SA-URANS computation of this section and in the three-dimensional SA-DDES simulation presented in Section 8.4.

After $t = 1.5 \cdot 10^{-2}s$ the upward rotation of the NACA0021 airfoil has been finished and the vortex has been partly interpolated onto the vortex transport grid. Fig. 8.26 (left) shows that the vortex is completely homogeneous in spanwise direction. No three-dimensional structures have evolved at the FNG airfoil and the spanwise velocity remains small except for the region in the cove and directly at the upper flap surface (see Fig. 8.26 (right)). This lack of three-dimensional structures both at the vortex and at the FNG airfoil, which will remain throughout the whole simulation, is expected due to the use of SA-URANS.

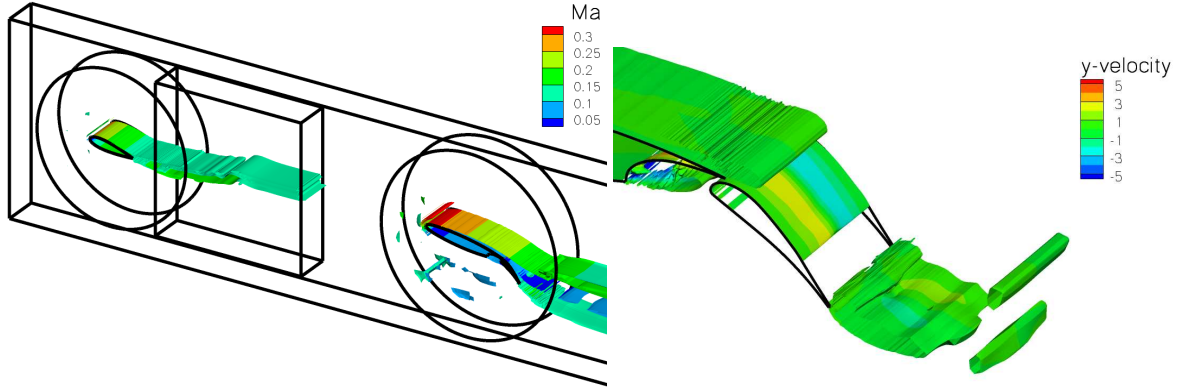


Figure 8.26: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q-criterion at a value of $100\frac{1}{s^2}$ (colored with y-velocity) (right) at $t = 1.5 \cdot 10^{-2}s$.

At $t = 1.9 \cdot 10^{-2}s$ the vortex has been fully interpolated onto the vortex transport grid, which is then transported to the right (not shown).

After $t = 2.5 \cdot 10^{-2}s$ the vortex transport grid has been moved 0.3m towards the FNG airfoil, which is indicated in Fig. 8.27 (left). Figure 8.27 (right) looks almost identical to Fig. 8.26 (right).

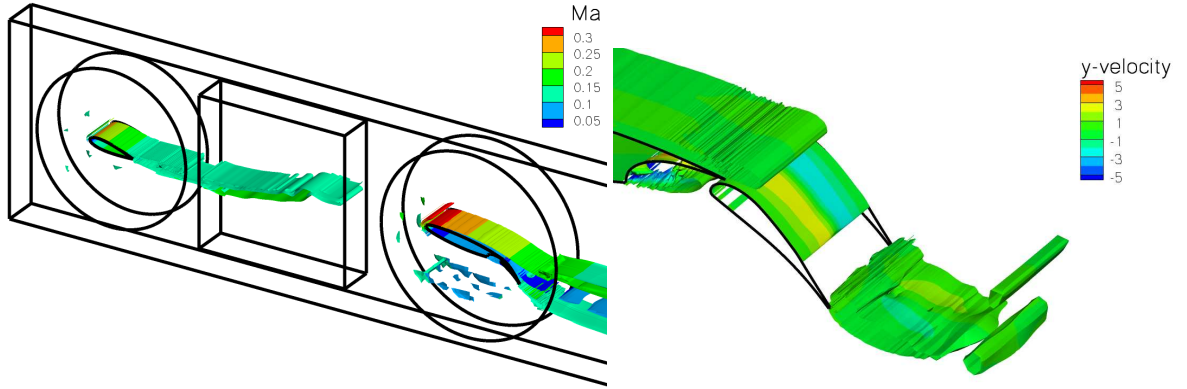


Figure 8.27: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q-criterion at a value of $100\frac{1}{s^2}$ (colored with y-velocity) (right) at $t = 2.5 \cdot 10^{-2}s$.

At $t = 3.1 \cdot 10^{-2}s$ the vortex transport grid has reached its final position and is stopped (not shown), while the vortex continues to move to the right.

After $t = 3.9 \cdot 10^{-2}s$ the vortex has been partly interpolated onto the FNG grid. Figure 8.28 shows that similar to the two-dimensional simulations of Section 8.1 the vortex splits into subvortices and both the vortex and the flow around the FNG airfoil remain two-dimensional.

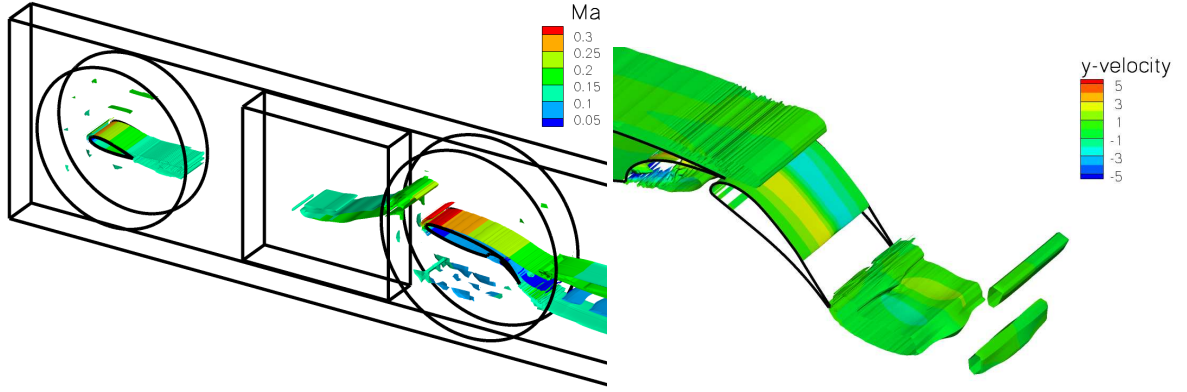


Figure 8.28: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q -criterion at a value of $100\frac{1}{s^2}$ (colored with y -velocity) (right) at $t = 3.9 \cdot 10^{-2}s$.

After $t = 4.7 \cdot 10^{-2}s$ the vortex has moved above the FNG airfoil, where a main vortex has demerged from the rest of the longitudinal vortex, see Fig. 8.29 (left). No effect of the vortex-airfoil interaction regarding stall can be identified in Fig. 8.29 (right).

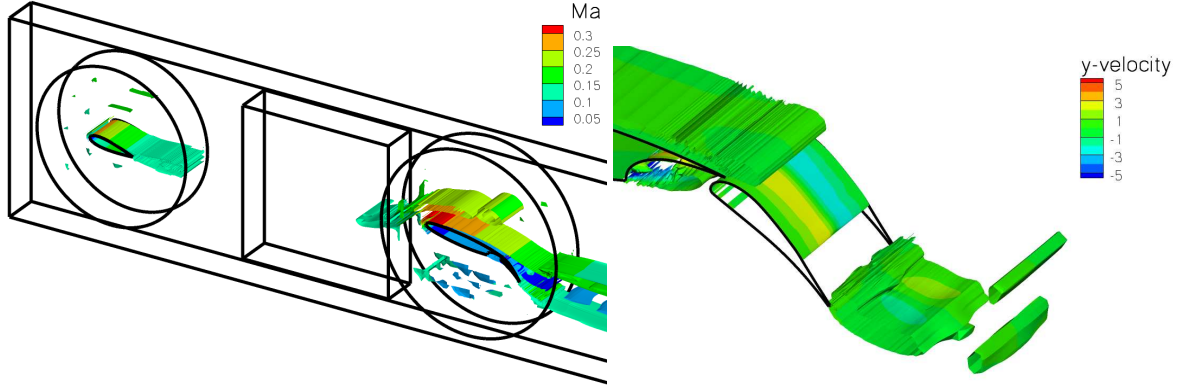


Figure 8.29: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q -criterion at a value of $100\frac{1}{s^2}$ (colored with y -velocity) (right) at $t = 4.7 \cdot 10^{-2}s$.

The C_l -curve, which is shown in Fig. 8.30, oscillates significantly less than in the two-dimensional simulations utilising low Mach number preconditioning, which were examined in the grid convergence study in Section 8.1. Still, only a small effect on C_l due to the vortex-airfoil interaction can be observed and no lift breakdown takes place. The times just before and during the vortex interaction are additionally highlighted in Fig. 8.30 by solid lines.

Again no stall occurs, which can be seen in Fig. 8.31, where the skin friction coefficients at two times before the arrival of the vortex ($t = 3.9 \cdot 10^{-2}s$) and during the vortex-airfoil interaction ($t = 4.7 \cdot 10^{-2}s$) are presented. As both C_f -curves are almost identical, the effect of the vortex-airfoil interaction seems to be small with respect to flow separation at the FNG airfoil.

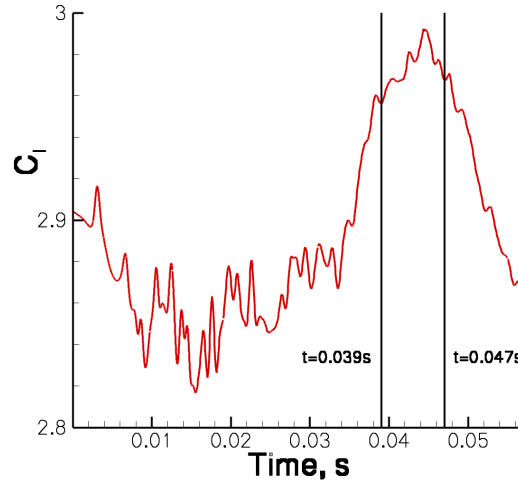
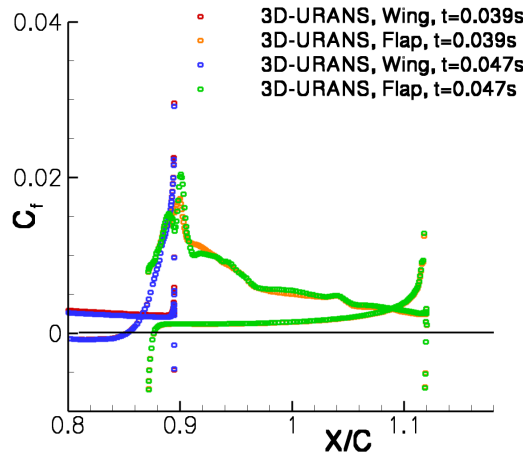


Figure 8.30: Time-dependent lift coefficient.

Figure 8.31: Skin friction coefficients at $t = 3.9 \cdot 10^{-2}s$ and $t = 4.7 \cdot 10^{-2}s$.

As a summary of this section it can be stated, that the simulation performed successfully. Also a small effect of the vortex-airfoil interaction could be verified, although just as in the two-dimensional simulations no stall occurred. As expected, the results remained almost fully two-dimensional because of the choice of the turbulence model.

8.4 A three-dimensional SA-DDES simulation

The final example of the numerical applications consists of a three-dimensional SA-DDES computation.

We want to emphasize that the simulation shown in this section should only be regarded as first attempt to apply the Chimera technique in combination with a

DES model to the creation, transport and interaction of realistic vortices with an airfoil. The result should therefore not be considered as final statement regarding the reliability of DES models for this kind of setup. The aim of this section is rather to compare the outcomes obtained with SA-DDES to the ones by SA-URANS, which possibly allows conclusions regarding necessary model modifications for forthcoming simulations. As the SA-DDES computation shown in this section has been computed for four months on 48 processors of the computer cluster ENIGMA of the DLR Göttingen, it becomes obvious that the development and validation of these model modifications would vastly exceed the scope of this thesis. After all, the focus of this thesis lies on the application and validation of the simulation method and not on an improved DES modeling.

Note, that we use the standard SA-DDES model and not the modified SA-DDES₁₆ version introduced in Chapter 6. Before looking at the simulation in detail, we want to briefly discuss the reasons for and against this choice. While it is shown at the end of this section that the RANS-part is slightly too thin to contain the complete attached boundary layer, the occurring separation at the wing is modest compared to the HGR01-testcase. As at some points in time the flow reattaches, it is unclear, if this small separation is caused by modeled stress depletion at all. Therefore the use of the original SA-DDES turbulence model seems acceptable for this testcase. The reason against using the SA-DDES₁₆ version is, that a thickened RANS-part at the FNG airfoil might suppress three-dimensional turbulent structures. As a consequence the SA-DDES₁₆ outcome could be similar to the result obtained by SA-URANS from Section 8.3, which is undesirable. Nevertheless, it would be interesting to compare the result shown in this section with the outcome of an SA-DDES₁₆ computation. The performance of an SA-DDES₁₆ simulation is therefore suggested as an important next step in Section 9.2.

Besides of the change in the turbulence model, the complete flow setup (Mach number, Reynolds number, boundary conditions, three-dimensional grid, time step size during the NACA0021 rotation, usage of low Mach number preconditioning and procedure of the computation) remains unchanged compared to the three-dimensional SA-URANS simulation in Section 8.3 – with the exception of the following two modifications:

Firstly, the time step size after the upward rotation of the NACA0021 airfoil is further decreased to $2.5 \cdot 10^{-5} s$ to meet the requirements of an SA-DDES computation. Secondly, the vortex transport grid is kept $0.2 \cdot 10^{-2} s$ longer at its original position, before it is transported towards the FNG airfoil. This turned out to be necessary, because at $t = 1.9 \cdot 10^{-2} s$ the vortex had not been fully interpolated onto the vortex transport grid.

Similar to the three-dimensional SA-URANS simulation, the relatively left pictures of Fig. 8.32-8.35 show the isosurfaces of the vorticity at a value of 100 colored by the Mach number, while in the relatively right pictures isosurfaces of the instantaneous Q -criterion at a value of $100 \frac{1}{s^2}$ in the area of the FNG airfoil can be seen, which additionally contain the streamwise velocity.

After $t = 1.5 \cdot 10^{-2} s$ the NACA0021 airfoil has been rotated upwards by 10° and the vortex, which is homogeneous in spanwise extension, has evolved from the trailing

edge (see Fig. 8.32 (left)). Compared to the result of the three-dimensional SA-URANS simulation shown in Fig. 8.26 (left), the vortex seems to be moving slower to the right. It turned out impossible to determine, whether this vortex deceleration really takes place or if alternatively the right vortex part seen in Fig. 8.32 (left) is actually not part of the vortex but belongs to the wake flow of the NACA0021 airfoil, which in the three-dimensional DDES simulation has vanished due to dissipation. The vorticity at the trailing edge of the wing looks more detached than in the three-dimensional SA-URANS computation, which hints to an increased flow separation. Figure 8.32 (right) reveals that no three-dimensional structures have evolved yet, although the two-dimensional rolls at the flap of the FNG airfoil are more numerous than in the three-dimensional SA-URANS simulation.

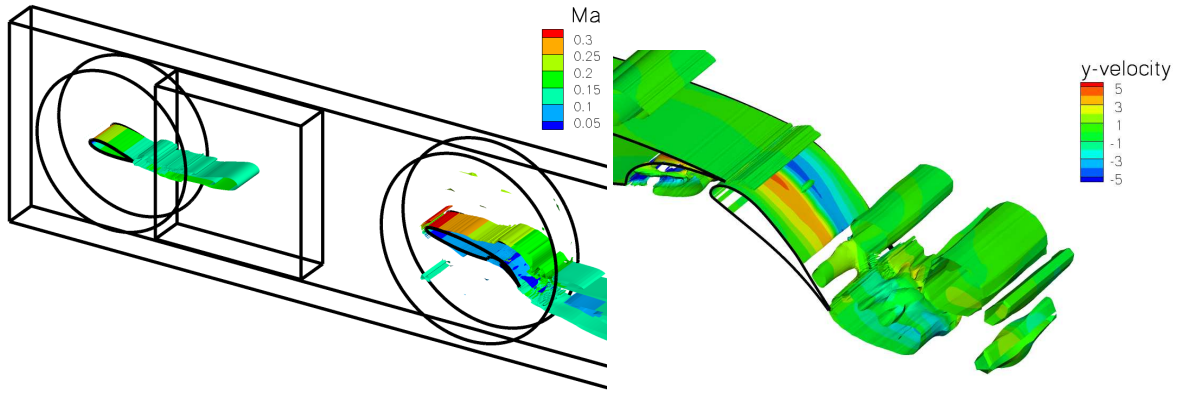


Figure 8.32: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q-criterion at a value of $100 \frac{1}{s^2}$ (colored with y-velocity) (right) at $t = 1.5 \cdot 10^{-2}s$.

At $t = 2.1 \cdot 10^{-2}s$ the vortex interpolation onto the vortex transport grid has been finished and the vortex transport grid is transported to the right (not shown). After $t = 2.6 \cdot 10^{-2}s$ the vortex transport grid has moved 0.25m towards the FNG airfoil, which is seen in Fig. 8.33 (left). While the vortex remains homogenous in spanwise extension and contains no small-scale turbulent structures, the flow at the FNG airfoil looks more attached than at $t = 1.5 \cdot 10^{-2}s$. Figure 8.33 (right) proves that three-dimensional structures have emerged, which is indicated both by the shape of the Q-isosurfaces and by the increased spanwise velocities.

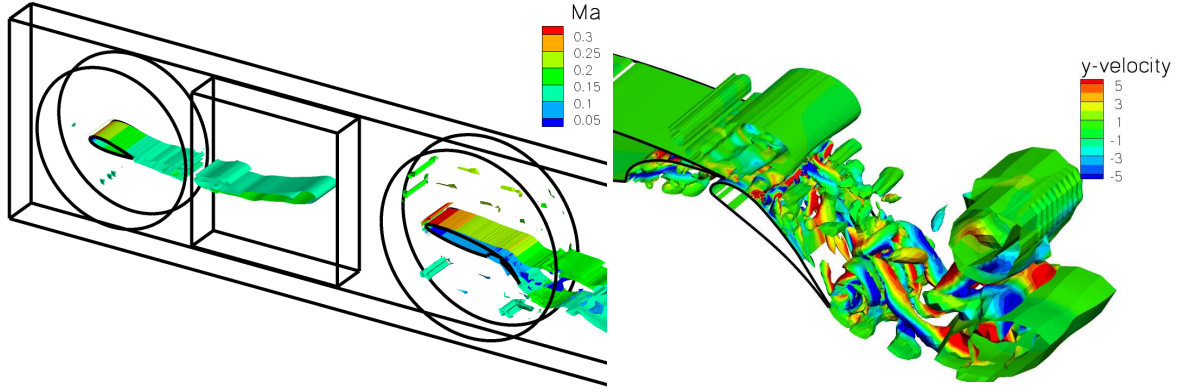


Figure 8.33: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q -criterion at a value of $100\frac{1}{s^2}$ (colored with y -velocity) (right) at $t = 2.6 \cdot 10^{-2}s$.

At $t = 3.3 \cdot 10^{-2}s$ the vortex transport grid has reached its final position and is stopped (not shown), while the vortex continues to move to the right.

The vortex interpolation onto the FNG grid has started at $t = 3.7 \cdot 10^{-2}s$, which is indicated in Fig. 8.34 (left) and intense three-dimensional structures with large spanwise velocities have evolved in the cove and behind the FNG flap, see Fig. 8.34 (right).

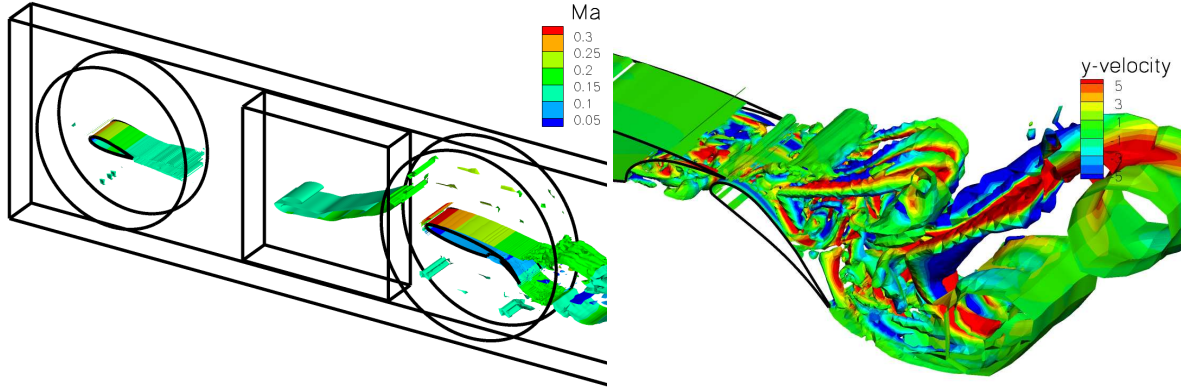


Figure 8.34: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q -criterion at a value of $100\frac{1}{s^2}$ (colored with y -velocity) (right) at $t = 3.7 \cdot 10^{-2}s$.

At $t = 4.7 \cdot 10^{-2}s$ the vortex interpolation has been finished and the vortex has moved above the FNG airfoil, see Fig. 8.35 (left). Contrary to the three-dimensional SA-URANS simulation at this time, no visible main vortex has split from the longitudinal vortex. While Fig. 8.35 (right) reveals that the area containing three-dimensional content is still very large and includes large spanwise velocities, it is hard to tell, whether the vortex-airfoil interaction has a significant impact on the separation region at the flap.

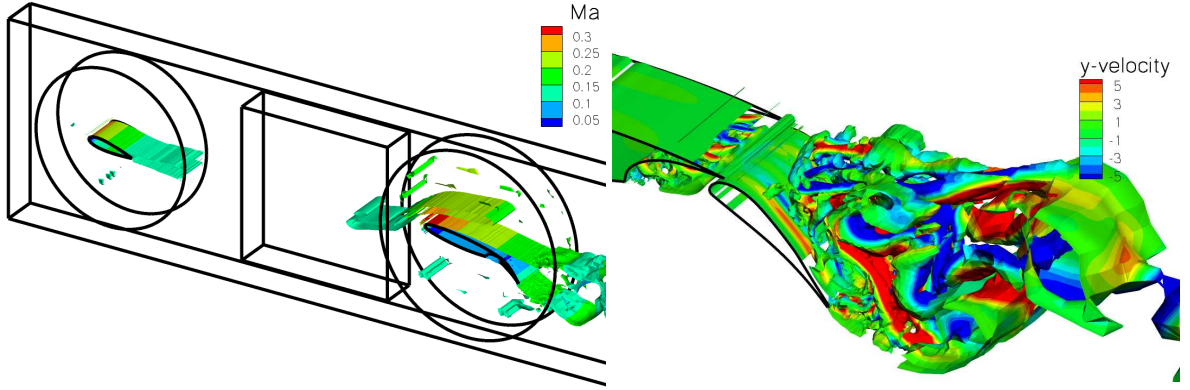


Figure 8.35: Isosurfaces of the vorticity at a value of 100 (colored with Mach number) (left) and isosurfaces of the instantaneous Q -criterion at a value of $100 \frac{1}{s^2}$ (colored with y -velocity) (right) at $t = 4.7 \cdot 10^{-2} s$.

Next we want to investigate, if the vortex has an effect on C_l . Figure 8.36 shows that a significant lift breakdown long before the arrival of the vortex takes place. Also no clear vortex effect on C_l can be identified (the times just before and during the vortex interaction are again marked by horizontal lines). The straightforward assumption regarding the cause of the lift breakdown is - identical as in the HGR01-testcase examined in Chapter 6 - a too thin RANS-part at the FNG airfoil, leading to modeled stress depletion and grid induced separation. To check whether this holds true, Fig. 8.37 shows the RANS-LES distribution and the streamlines at $t = 3.7 \cdot 10^{-2} s$, which is the time just before the arrival of the vortex. While the RANS-part at the trailing edge of the wing looks relatively thin, only a modest flow separation at the end of the wing can be observed. As desired, both the region in the cove and behind the flap are treated in LES-mode. As a consequence, turbulent structures have evolved here.

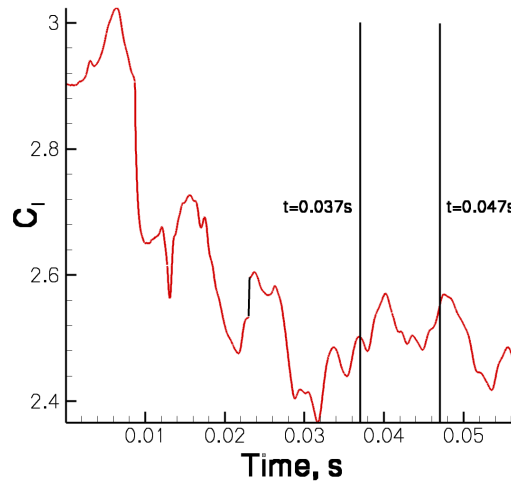


Figure 8.36: Time-dependent lift coefficient.

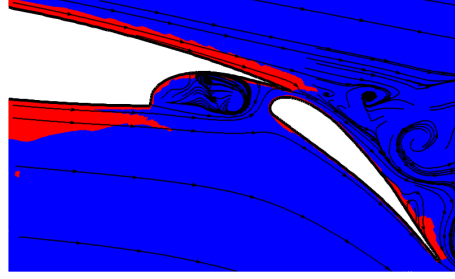


Figure 8.37: RANS-LES distribution at $t = 3.7 \cdot 10^{-2}s$.

Figure 8.38 (left) shows the Mach number of the three-dimensional SA-DDES computation at $t = 3.7 \cdot 10^{-2}s$ to further check, whether the thickness of the RANS-part is sufficient. Compared to Fig. 8.37 it is obvious, that the RANS-part does not contain the complete attached boundary layer. Also at the trailing edge of the wing a small separation can be seen. For comparison, Fig. 8.38 (right) shows the Mach number of the three-dimensional SA-URANS computation from Section 8.3 at the same time of the simulation. Clearly, the flow at the wing is more attached than in the three-dimensional SA-DDES computation.

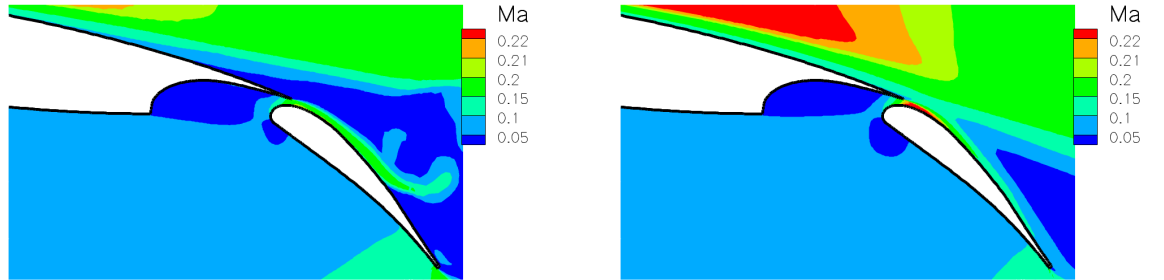


Figure 8.38: Mach number of three-dimensional SA-DDES simulation (left) and of three-dimensional SA-URANS computation (right) at $t = 3.7 \cdot 10^{-2}s$.

Figure 8.39 shows the skin friction coefficients both at $t = 3.7 \cdot 10^{-2}s$ and $t = 4.7 \cdot 10^{-2}s$, where the right picture focuses on the trailing edge of the wing. While at $t = 4.7 \cdot 10^{-2}s$ (during the vortex-airfoil interaction) no flow separation neither at the wing nor at the flap can be observed, surprisingly the flow has slightly detached at $t = 3.7 \cdot 10^{-2}s$. This leads to the following two conclusions:

Firstly, it is unclear whether grid induced separation occurs or not. While the skin friction coefficient at $t = 3.7 \cdot 10^{-2}s$ and the lift breakdown suggest this being the case, the reattachment at $t = 4.7 \cdot 10^{-2}s$ stands in contrast to this assumption. Secondly, no stall occurs due to the vortex-airfoil interaction.

Furthermore it is unclear, why the two skin friction coefficients vary that immensely. It is assumed that this is due to instationarities independent of the vortex.

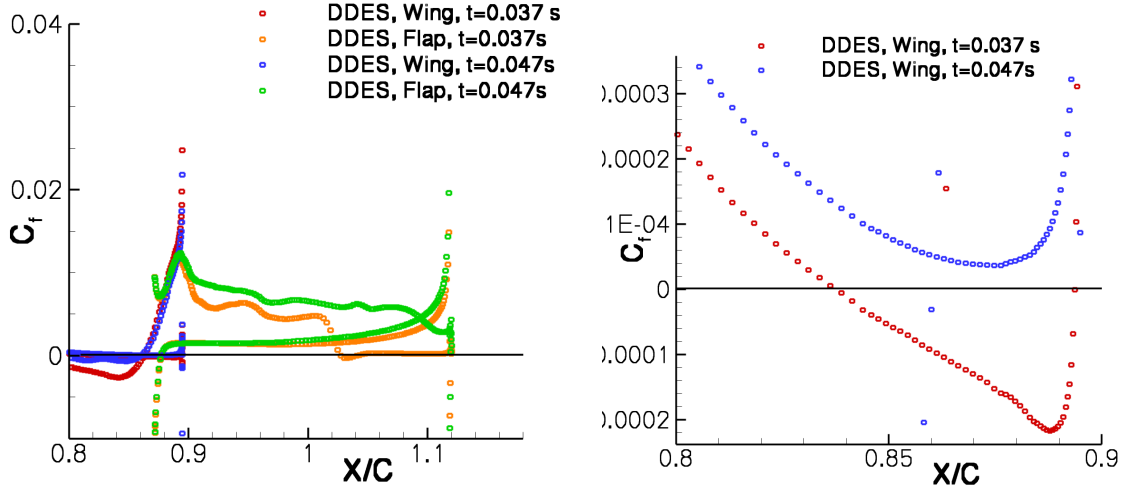


Figure 8.39: Skin friction coefficients at $t = 3.7 \cdot 10^{-2} s$ and $t = 4.7 \cdot 10^{-2} s$ at the trailing edge of the wing and the flap (left) and with zoom on the trailing edge of the wing (right).

As a summary of Section 8.4 it can be stated that the three-dimensional SA-DDES computation performed only partly successful. On the positive side it can be noted that the vortex was successfully transported towards the FNG airfoil without losing it by means of numerical dissipation. Also strong three-dimensional structures in the cove and behind the flap evolved for the first time due to the use of SA-DDES. Still, the following open questions arose, which must be addressed in the future:

- Has modeled stress depletion at the FNG airfoil occurred, resulting in grid induced separation and lift breakdown?
While the RANS-part at the main wing was relatively thin and at some times during the simulation a slight detachment occurred, the flow reattached at subsequent points in time. The question can therefore not be answered definitively at this stage.
- How can the C_l -breakdown (see Fig. 8.36) long before the arrival of the vortex be explained?
One explanation could be modeled stress depletion, but as just explained it is unclear if this phenomenon actually occurs.
- Have turbulent structures from the cove and the wake of the main wing intruded into the RANS-part at the attached boundary layer of the flap, resulting in a depletion of the RANS-part there? (A similar hypothesis was constructed by Jakubek [55], where SA-DDES simulations of a three-element F15 airfoil possibly lead to these problems.)
While this question can not be negated for sure, we want to mention that the flow directly at the flap remains attached throughout the whole computation, which can be seen when looking at the C_f distributions. As - in contrast to

the outcome shown in [55] - the resulting flow shown in this thesis contains a large recirculation region between the attached boundary layer of the FNG flap and the wake of the main wing, both flows are highly different and should therefore not be compared directly.

- Why does the vortex from the NACA 0021 airfoil remain two-dimensional?
It is assumed that the short distance of $0.6m$ between the two airfoils is too small to allow three-dimensional structures to evolve. Nevertheless, it remains an open task to show that these turbulent structures do occur, if the airfoil-to-airfoil distance is increased.
- Why does the vortex-airfoil interaction not result in stall?
This question is especially hard to answer, because the behaviour of DES models applied to the flow around stalling multielement airfoils is so little understood. Possible ideas are that the original angle of attack of $\alpha = 10^\circ$ at the FNG airfoil is still too small or that the oncoming vortex already has dissipated too much. Also the longitudinal vortex structure and the observation that the vortex does not actually collide with the FNG airfoil, but instead moves above the wing, may play a role.

Summary of the chapter

As conclusion of Chapter 8 we have shown that the simulation method allows to generate and transport a realistic vortex and to model its interaction with an airfoil near stall. While in the two-dimensional simulations the use of low Mach number preconditioning led to strong C_l -oscillations, this problem was less severe when switching to three-dimensional. Section 8.2 showed that a small airfoil-to-airfoil distance of $2m$ is superior to a large distance of $4m$ with respect to vortex preservation and its effect on C_l . While in the three-dimensional SA-URANS computation in Section 8.3 the vortex-airfoil interaction still resulted in a small effect on C_l , no clear impact on the lift could be observed in the three-dimensional SA-DDES simulation in Section 8.4. While strong turbulent three-dimensional structures evolved at the FNG airfoil, the SA-DDES computation possibly included a mild form of modeled stress depletion and grid induced separation.

Chapter 9

Conclusion

9.1 Summary of the results

At the end of this thesis we want to give an overview of the results. As stated in Section 1.1 the objective of this thesis lies on the development and application of a numerical simulation method that allows to generate a realistic vortex, to transport it towards an airfoil and to simulate the vortex-airfoil interaction to predict the forces and moments acting on the wing. The following classes of testcases have been performed:

- flow over a backward facing step
- an HGR01 airfoil at stall
- transport and airfoil interaction of vortices
- applications of the simulation method

Flow over a backward facing step

The aim of the backward step testcase in Chapter 5 is to test the performance of the SA-DDES turbulence model for a problem with massive flow separation. Also the required grid refinement to obtain reliable results is investigated by means of a grid convergence study. By comparing to experimental data and a two-dimensional SA-RANS simulation it can be said that the use of the SA-DDES model leads to good results on the medium and fine grid. Both the reattachment point indicated by the skin friction coefficient and the pressure distribution agree well with the experimental data and also three-dimensional small-scale turbulent structures evolve. In contrast, the simulation on the coarse grid is not capable of resolving a sufficient amount of turbulent content, leading to a poor outcome. Overall it is verified that the SA-DDES model performs well for this testcase, if the grid is sufficiently fine.

While they are not further used within the simulation method, Chapter 5 additionally examines the reliability of two sensors obtained from the SA-DDES computations. The first sensor L_{vk} shows, in which regions the flow tends to become

unsteady, whereas the aim of the second sensor *Indicator* is to guarantee a sufficiently fine grid in the LES-region. While the performances of both sensors are convincing, it remains an open problem to find an appropriate grid scaling parameter Δ for L_{vk} .

As a summary of Chapter 5 it can be said that the SA-DDES model performs well for this testcase, if the grid is chosen sufficiently fine.

An HGR01 airfoil at stall

In Chapter 6 the three-dimensional flow around an HGR01 airfoil at stall is simulated by means of the SA-DDES turbulence model. As in this testcase only a mild trailing edge separation occurs, the standard SA-DDES model is not capable of predicting the correct attached boundary layer thickness. This results in a too thin RANS-part, leading to modeled stress depletion and grid induced separation.

To overcome this flaw, a model modification named SA-DDES₁₆ is introduced, in which the RANS-part at the airfoil is thick enough to allow for stable simulations. The full range of stall angles of $\alpha = 12^\circ - 16^\circ$ is simulated by means of SA-DDES₁₆. It is shown that the SA-DDES₁₆ results are comparably good as stationary SA-RANS simulations regarding prediction of skin friction and pressure coefficients. On the other hand, both SA-DDES₁₆ and SA-RANS strongly overestimate the experimentally determined lift coefficient. The reason for this mismatch lies in a strong underprediction of the trailing edge separation by both models. The SA-DDES₁₆ model is also not capable of producing three-dimensional turbulent structures in the wake, because the separation region is too small.

As a summary of Chapter 6 it is stated that the SA-DDES model can be used to perform stable computations of the flow around an stalling airfoil, if the RANS-layer at the wing covers a sufficient part of the attached boundary layer (which is guaranteed by using SA-DDES₁₆ in this testcase). As the trailing edge separation is too small to trigger turbulent structures, the SA-DDES₁₆ results are not superior to SA-RANS outcomes.

Transport and airfoil interaction of vortices

The goal of Chapter 7 is to determine, if the proposed simulation method allows to transport analytical vortices over large distances without losing them due to numerical dissipation. Additionally it is tested, if the simulation method can be used to simulate vortex-airfoil interactions. All problems are two-dimensional and always URANS turbulence models are utilised.

At the beginning three testcases of varying complexity are shown, in which one or more analytical vortices are interpolated onto a fine cartesian vortex transport grid, which is moved relative to a background grid. In all three simulations the vortices can be maintained. Also the use of more than one initial vortex or a coarse background grid are problem-free.

The last two testcases prove that the simulation method can be applied to simulate vortex-airfoil interactions. First the collision of a vortex and a NACA 0012 airfoil at $\alpha = 0^\circ$ is examined, which results in a massive change in lift, drag and

moment. While the vortex collision can be interpreted as a change in the incidence angle, it is shown that the simulation of a flapping NACA 0012 airfoil at undisturbed onflow conditions is an insufficient approximation of the vortex-airfoil interaction. The use of the Chimera technique is therefore proven to be mandatory. In the last example, in which a vortex interacts with an ONERA-A airfoil at $\alpha = 13.3^\circ$, wing stall is successfully simulated.

Chapter 7 proves that the simulation method allows to transport vortices over large distances and to model vortex-airfoil interactions. All simulations are very robust and also the use of coarse or hybrid background grids are unproblematic.

Applications of the simulation method

In Chapter 8 applications of the complete simulation method are presented. While a NACA 0021 airfoil is used to generate the vortex, the latter eventually interacts with a two-element FNG airfoil at $\alpha = 10^\circ$, which is close to stall. A fine cartesian vortex transport grid is used to transport and preserve the vortex and both airfoils are located within a windtunnel.

At first, stationary two-dimensional RANS simulations of the FNG airfoil at undisturbed onflow conditions are shown, where both the effect of the turbulence model and the angle of attack resulting in stall are examined. It turns out that the Menter-SST turbulence model more accurately predicts the correct stall angle than the SA-RANS model, which is validated by comparing to experimental data. For both turbulence models the incidence angle of $\alpha = 10^\circ$ is close to stall, such that this angle of attack is used within all subsequent computations.

Next four two-dimensional URANS simulations including vortex generation, vortex transport and vortex-FNG interaction by means of the simulation method are presented. Both the effect of varying the turbulence model and the use of a small or alternatively large airfoil-to-airfoil distance is analysed. While it turns out that in the two "small distance" computations the breakdown in C_l caused by the vortex interaction is significantly larger than in the two "large distance" simulations, in neither case stall occurs. The two turbulence models result in qualitatively similar results.

In the following example a three-dimensional SA-URANS simulation of the simulation method is performed using the small airfoil-to airfoil distance of 2m. As expected, no three-dimensional turbulent structures evolve. While a small effect on C_l due to the vortex-airfoil interaction is visible, no stall can be verified.

The final testcase consists of a three-dimensional SA-DDES computation, where again a small distance between the airfoils is utilised. A slight form of grid induced separation at the FNG main wing is assumed, because the RANS-part looks relatively thin compared to the attached boundary layer. As at some points in time the flow detaches and reattaches again, it is impossible to decide, if the thickness of the RANS-part is actually the cause of the separation, or if physically correct flow instationarities are present. While a large separation region with high spanwise velocities and small-scales structures in the cove and the wake of the FNG-airfoil evolves, it remains unclear whether this is due to the vortex interaction. Again, no

clear stall behaviour can be seen.

To sum up Chapter 8 we have shown that the proposed simulation method allows to generate and transport a realistic vortex and to model its interaction with an airfoil near stall. In all computations the vortex could be preserved without losing it due to numerical dissipation. Nevertheless, no stall could be verified in the simulations. While in the three-dimensional SA-URANS computation the vortex-airfoil interaction has an effect on C_l , no clear impact on the lift coefficient can be observed in the three-dimensional SA-DDES simulation. The use of the latter results in the development of strong turbulent three-dimensional structures at the FNG airfoil and also possibly a mild form of grid induced separation.

An overview of the computational time to perform the simulations

At the end of the summary we want to give a quick overview of the computational time required to perform the most expensive (three-dimensional) simulations shown in this thesis. While the two-dimensional simulations were mostly realised using a single processor, all three-dimensional computations have been performed on the computer cluster ENIGMA of the DLR Göttingen.

- Chapter 5:

While the computation of the backward facing step on the coarse grid (2.2×10^5 points) was performed for approximately 10 days using 16 CPUs on ENIGMA, the computation on the medium grid (8.8×10^5 points) took 40 days utilising the same number of CPUs. By far the most expensive computation was the simulation on the fine grid (6.6×10^6 points), which had to be performed for 120 days using 32 CPUs on ENIGMA.

For comparison: When using the incompressible solver THETA of the DLR TAU-code an LES simulation using wall-functions on a grid, which in the middle of the domain was comparably fine as the fine grid used within this thesis, only took 21 days on a single processor (see [64]).

- Chapter 6:

The performance of the complete SA-DDES₁₆ simulation shown in Section 6.3 required approximately seven months using 64 processors on ENIGMA.

- Chapter 8:

While the realisation of the SA-URANS computation from Section 8.3 took four months on 32 processors of ENIGMA, the SA-DDES simulation shown in Section 8.4 even had to be performed for four months on 48 processors.

Overall, the computations turned out to be comparably expensive, which was due to a large number of up to 300 inner iterations per timestep and the choice of a small timestep size in many simulations. This was often necessary to achieve convergence and allow for stable simulations.

It also can be stated that the DLR TAU-code is much slower than the incompressible solver THETA of the DLR TAU-code. It seems that a pressure-based method to

solve the incompressible Navier-Stokes equations is more efficient than solving the compressible equations, which is done in TAU.

9.2 Future work

There is still work that is worth considering in the future:

- The backward facing step computations from Chapter 5 can be performed on hybrid grids, which simplifies the grid adaptation.
- The two sensors shown in Section 5.3 can be utilised to improve the simulation method. While the first sensor L_{vk} may be applied to switch to LES-mode in regions with massive flow separation, the second sensor *Indicator* guarantees a sufficiently fine grid in the LES-part. When using hybrid grids the grids can easily be refined in regions, where *Indicator* suggests that the grid is locally too coarse. A main issue is indeed to find a correct grid scaling parameter Δ for the first sensor.
- The flow around the HGR01 airfoil can be performed using an alternative scheme to discretise the convective fluxes, which might trigger three-dimensional turbulent structures.
- The simulation method can be applied in combination with SA-DDES₁₆.
- The simulation method can be performed by means of the Menter-SST URANS model and the SST-DES model. To do so successfully, a deep investigation of the SST-DES model is required.
- Instead of considering a vortex-FNG interaction, the FNG airfoil can be substituted by another aerodynamical object, e. g. a nacelle.
- When performing the simulation method, the FNG airfoil could be set at a slightly higher initial incidence angle (e. g. $\alpha = 11^\circ$). This "closer-to-stall-angle" might finally result in stall due to the vortex-airfoil interaction. Yet it remains unclear, if this setup allows for a robust computation, because simulations that close to stall are more sensitive to small changes and tend therefore to become numerically unstable.
- The numerical outcomes shown in this thesis will be validated upon several experiments of the DFG 1066 Forschergruppe, which are performed by other subprojects.
- The numerical results shown in this thesis should be compared to computations performed with the incompressible solver THETA of the DLR TAU-code. Especially a comparison of the computational times of both codes applied to comparable setups should be performed.

- The impact of the underlying discretisation on the development of three-dimensional turbulent small-scale structures must be investigated. This would allow to decide, if a forcing procedure is required for turbulent structures to evolve.

Appendix A

A global existence result of the Navier-Stokes-Fourier system

The existence result shown in this section has been presented by Feireisl et al. [31] (theorem 3.1 on p. 50). The starting point is the Navier-Stokes-Fourier system, which can be written as follows, where all definitions are identical to the ones defined in Section 2.1:

Definition A.0.1 (Navier-Stokes-Fourier system)

For a bounded domain $\Omega \subset \mathbb{R}^3$ and time $t \in (0, \tau)$ we seek density $\rho(x, t)$, velocity $\vec{u}(x, t)$, and temperature $T(x, t)$ such that:

$$\frac{\partial}{\partial t} \rho + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \quad (\text{A.1})$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_j u_i) + \frac{\partial}{\partial x_i} p = \frac{\partial}{\partial x_j} t_{ji} + \rho f_i \quad (\text{A.2})$$

$$\frac{\partial}{\partial t} E + \frac{\partial}{\partial x_j} (E u_j) + \frac{\partial}{\partial x_j} (q_j - u_i t_{ij} + p u_j) = \rho f_i u_i + \rho \Theta \quad (\text{A.3})$$

Here $f(x, t)$ denotes the volume force acting on a fluid and $\Theta(x, t)$ is the production of internal energy.

Besides Eqs. (A.1), (A.2) and (A.3) the following *entropy production equation* shall be valid: Find specific entropy $s(\rho, T)$ such that :

$$\frac{\partial}{\partial t} (\rho s) + \frac{\partial}{\partial x_j} (\rho s u_j) + \frac{\partial}{\partial x_j} \left(\frac{q_j}{T} \right) = \sigma + \frac{\rho}{T} \Theta \quad (\text{A.4})$$

with entropy production rate σ given by:

$$\sigma = \frac{1}{T} \left(\frac{\partial}{\partial x_j} \cdot u_i t_{ij} - \frac{q_j \frac{\partial}{\partial x_j} T}{T} \right) \quad (\text{A.5})$$

Also Gibbs' equation shall be fulfilled, which is given by:

$$T Ds = De + p D \left(\frac{1}{\rho} \right) \quad (\text{A.6})$$

and D denotes the differential with respect to the variables ρ, T .
Next we introduce the weak formulation of Eqs. (A.1)-(A.3) and (A.4):

Continuity equation

The weak formulation of Eq. (A.1) is given by:

$$\begin{aligned} & \int_0^\tau \int_\Omega \rho B(\rho) \left(\frac{\partial}{\partial t} \varphi + u \nabla_x \varphi \right) dx dt \\ &= \int_0^\tau \int_\Omega b(\rho) \nabla_x \cdot u \varphi dx dt - \int_\Omega \rho_0 B(\rho_0) \varphi(\cdot, 0) dx \end{aligned} \quad (\text{A.7})$$

where $\rho_0 = \rho(\cdot, 0)$ and for the admissible test functions we demand:

$$b \in L^\infty \cap C[0, \infty), B(\rho) = B(1) + \int_1^\rho \frac{b(z)}{z^2} dz \quad (\text{A.8})$$

$$\varphi \in C_c^1(\overline{\Omega} \times (0, \tau); \mathbb{R}) \quad (\text{A.9})$$

As minimum regularity of the solutions we require:

$$\rho \geq 0, \rho \in L^1(\Omega \times (0, \tau); \mathbb{R}), \quad (\text{A.10})$$

$$\rho u \in L^1(\Omega \times (0, \tau); \mathbb{R}^3), \nabla_x \cdot u \in L^1(\Omega \times (0, \tau); \mathbb{R}) \quad (\text{A.11})$$

$$u \cdot n|_{\partial\Omega} = 0 \text{ (impermeability of the boundary } \partial\Omega) \quad (\text{A.12})$$

Momentum equation

The weak formulation of Eq. (A.2) is given by:

$$\begin{aligned} & \int_0^\tau \int_\Omega \left(\rho u \cdot \frac{\partial}{\partial t} \varphi + \rho(u \otimes u) : \nabla_x \varphi + p \nabla_x \cdot \varphi \right) dx dt \\ &= \int_0^\tau \int_\Omega (t : \nabla_x \varphi - \rho f \cdot \varphi) dx dt - \int_\Omega (\rho u)_0 \cdot \varphi(\cdot, 0) dx \end{aligned} \quad (\text{A.13})$$

where $(\rho u)_0 = (\rho u)(\cdot, 0)$ and for the admissible test functions we demand:

$$\varphi \in C_c^1(\overline{\Omega} \times (0, \tau); \mathbb{R}^3) \quad (\text{A.14})$$

and either

$$\varphi \cdot n|_{\partial\Omega} = 0 \text{ (in the case of slip boundary conditions)} \quad (\text{A.15})$$

or

$$\varphi|_{\partial\Omega} = 0 \text{ (in the case of no-slip boundary conditions)} \quad (\text{A.16})$$

As minimum regularity of the solutions we require:

$$\rho u \in L^1(\Omega \times (0, \tau); \mathbb{R}^3), \rho|u|^2 \in L^1(\Omega \times (0, \tau); \mathbb{R}) \quad (\text{A.17})$$

$$p \in L^1(\Omega \times (0, \tau); \mathbb{R}), t \in L^1(\Omega \times (0, \tau); \mathbb{R}^{3 \times 3}), \rho f \in L^1(\Omega \times (0, \tau); \mathbb{R}^3) \quad (\text{A.18})$$

$$\nabla_x u \in L^1((0, \tau); L^q(\Omega; \mathbb{R}^{3 \times 3})) \text{ for a } q > 1 \quad (\text{A.19})$$

and either

$$u \cdot n|_{\partial\Omega} = 0 \text{ (in the case of slip boundary conditions)} \quad (\text{A.20})$$

or

$$u = 0 \text{ (in the case of no-slip boundary conditions)} \quad (\text{A.21})$$

Energy equation

The weak formulation of Eq. (A.3) is given by:

$$\begin{aligned} & \int_0^\tau \int_\Omega \left(E(t) dx \cdot \frac{\partial}{\partial t} \psi(t) \right) dt \\ &= - \int_0^\tau \int_\Omega \left(\rho u \cdot f(t) + \rho \Theta(t) \right) \psi(t) dx dt - \psi(0) E_0 \end{aligned} \quad (\text{A.22})$$

where $E_0 = \int_\Omega E(\cdot, 0) dx$ and for the admissible test functions we demand:

$$\psi \in C_c^1([0, \tau]; \mathbb{R}) \quad (\text{A.23})$$

As minimum regularity of the solutions we require:

$$E, \rho u \cdot f, \rho \Theta \in L^1(\Omega \times (0, \tau); \mathbb{R}) \quad (\text{A.24})$$

Entropy production equation

The weak formulation of Eq. (A.4) is given by:

$$\begin{aligned} & \int_0^\tau \int_\Omega \rho s \left(\frac{\partial}{\partial t} \varphi + u \cdot \nabla_x \varphi \right) dx dt + \int_0^\tau \int_\Omega \left(\frac{q}{T} \nabla_x \varphi \right) dx dt + \langle \sigma; \varphi \rangle_{M^+, C} \\ &= - \int_\Omega (\rho s)_0 \varphi(\cdot, 0) dx - \int_0^\tau \int_\Omega \left(\frac{\rho}{T} \Theta \varphi \right) dx dt \end{aligned} \quad (\text{A.25})$$

where $(\rho s)_0 = (\rho s)(\cdot, 0)$, $\langle \cdot; \cdot \rangle_{X^*, X}$ is the duality pairing between a vector space X and its dual space X^* and M^+ denotes the positive measures. Furthermore, $\sigma \in M^+(\overline{\Omega} \times [0, \tau]; \mathbb{R})$ such that

$$\sigma \geq \frac{1}{T} \left(t : \nabla_x u - \frac{q}{T} \nabla_x T \right) \quad (\text{A.26})$$

For the admissible test functions we demand:

$$\varphi \in C_c^1(\overline{\Omega} \times (0, \tau); \mathbb{R}) \quad (\text{A.27})$$

As minimum regularity of the solutions we require:

$$T > 0, \quad T \in L^q(\Omega \times (0, \tau); \mathbb{R}), \quad \nabla_x T \in L^q(\Omega \times (0, \tau); \mathbb{R}^3) \quad \text{for a } q > 1 \quad (\text{A.28})$$

$$\rho s \in L^1(\Omega \times (0, \tau); \mathbb{R}), \quad \rho s u, \frac{q}{T} \in L^1(\Omega \times (0, \tau); \mathbb{R}^3) \quad (\text{A.29})$$

$$\frac{\rho}{T} \Theta \in L^1(\Omega \times (0, \tau); \mathbb{R}) \quad (\text{A.30})$$

$$\frac{1}{T} (t : \nabla_x u), \frac{q}{T^2} \nabla_x T \in L^1(\Omega \times (0, \tau); \mathbb{R}) \quad (\text{A.31})$$

Having introduced the weak formulations of Eqs. (A.1)–(A.3) and (A.4) we are now in the position to state the following global existence result. Note, that the domain Ω , the initial data ρ_0 , $(\rho u)_0$, E_0 and $(\rho s)_0$ as well as the source terms f and Θ have to fulfill a number of additional constraint qualifications. Also p , e and s and additionally introduced transport coefficients must obey several structural hypotheses. While all of these conditions are stated by Feireisl [31], it is important to note that none of these conditions impose any restriction on the time t .

Theorem A.0.2 (Global existence of the Navier-Stokes-Fourier system)

Let all aboved mentioned constraint qualifications be fulfilled. Then for any time $\bar{\tau}$ the Navier-Stokes-Fourier system has a weak solution (ρ, u, T) on $\overline{\Omega} \times (0, \bar{\tau})$, i. e. (ρ, u, T) fulfill Eqs. (A.7) - (A.31).

The complete proof can be found in [31], p 50 ff., and is based on the following concept: Firstly, the original weak formulation is replaced by an approximate system, which is then proven to possess a weak solution locally in time. By using uniform estimates this result can be extended to the complete time intervall $(0, \bar{\tau})$. Finally, the limit of the approximate system is performed, which enables to recover the original weak formulation.

Validity of theorem A.0.2 for the compressible Navier-Stokes equations

We want to note, that theorem A.0.2 is also valid for the compressible Navier-Stokes equations defined in Eqs. (2.1) – (2.3) in combination with the equation of state given in Eq. (2.9): By choosing both the volume force f and the production of internal energy Θ as zero, it is easy to see that Eqs. (A.1) – (A.3) and Eqs. (2.1) – (2.3) are identical, i. e. the Navier-Stokes-Fourier system is a generalisation of the compressible Navier-Stokes equations. One can also show (see Feireisl et al. [31], p. 14 ff.) that the perfect gas law from Eq. (2.9) is a special case of the Gibbs' equation defined in Eq. (A.6).

As overall the Navier-Stokes-Fourier system with Gibbs' equation is a generalisation of the compressible Navier-Stokes equations closed with the perfect gas law, theorem A.0.2 is also valid for the latter system of equations used in the DLR TAU-code.

Appendix B

Nomenclature

Symbols and notations

Ω	bounded domain $\Omega \subset \mathbb{R}^d$
$\partial\Omega$	boundary of Ω
t	time, viscous stress
ρ	density
\vec{u}	velocity
T	temperature
p	pressure
E	energy density
S	strain-rate tensor
δ_{ij}	Kronecker delta
μ	molecular viscosity
e	specific internal energy
h	specific enthalpy
q	heat-flux
R	molecular gas constant
$\nabla u(x)$	gradient of a function u
$\nabla_x u(x, y)$	gradient of u with respect to x
$\nabla \cdot u(x)$	divergence of u
$\nabla_x \cdot u(x, y)$	divergence of u with respect to x
\vec{n}	normal vector
\vec{u}_∞	reference freestream velocity
T_∞	reference freestream temperature
ρ_∞	reference freestream density
a	speed of sound
v_ϕ	tangential velocity
Γ_0	constant circulation of a vortex
r	radius of a finite vortex
r_c	core radius of a finite vortex
r_m	middle radius of a finite vortex
r_o	outer radius of a finite vortex
δ	decay constant of a finite vortex

β	damping function of a finite vortex
Δt	time step size
Ma	Mach number
Re	Reynolds number
c	chord length of an airfoil
$\overline{f(x)}$	Reynolds average of instantaneous $f(x, t)$
$f'(x, t)$	fluctuating part of instantaneous $f(x, t)$
$\widetilde{f(x)}$	Favre average of instantaneous $f(x, t)$
$f''(x, t)$	(Favre) fluctuating part of instantaneous $f(x, t)$
μ_T	eddy viscosity
ν	kinematic viscosity
$\tilde{\nu}$	variable of the transport equation in the SA0 model
P	production term in the SA0 model
D	destruction term in the SA0 model
d	wall distance
Ω	magnitude of the vorticity
κ	von Kármán constant, $\kappa = 0.41$
k	turbulent kinetic energy
ω	specific dissipation rate
\tilde{d}	modified length scale used in DES
C_{DES}	DES constant
Δ	grid spacing
f_d	blending function used in SA-DDES
Ψ	low-Reynolds modification
α	angle of attack
h	step height of the BFS
C_d	drag coefficient
C_l	lift coefficient
C_m	moment coefficient
C_p	pressure coefficient
C_f	skin friction coefficient
Q	instantaneous Q-criterion
L_{vk}	von Kármán length scale
<i>Indicator</i>	sensor examined in the BFS computations
Δ	grid scaling parameter to scale <i>Indicator</i>
k_{res}	resolved part of the turbulent kinetic energy
k_{res}	modeled part of the turbulent kinetic energy
UW/U_∞^2	non-dimensionalised Reynolds stress
$\langle \cdot \rangle$	operator to perform time averaging
v	velocity relative to a background grid
v_x, v_z	x - and z -components of v
v_∞	freestream velocity
$v_{\infty, x}, v_{\infty, z}$	x - and z -components of v_∞
v_{local}	local velocity, $v_{local} := ((v_x - v_{\infty, x})^2 + (v_z - v_{\infty, z})^2)^{\frac{1}{2}}$
v_{vg}	velocity of a vortex grid relative to a background grid
$v_{vg, x}, v_{vg, z}$	x - and z -components of v_{vg}

$A : B$	scalar product of tensors A and B
$a \otimes b$	tensor product of vectors a and b
$\langle a; b \rangle$	scalar product between vector a and b
L^q	Lebesgue function space
C_c^1	space of continous functions with compact support

Abbreviations

(U)RANS	(Unsteady) Reynolds averaged Navier-Stokes
LES	Large Eddy Simulation
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
SA(O)	Spalart-Allmaras (Original) model
LES	Linearized Explicit Algebraic Stress model
Menter-SST	Menter-Shear Stress Transport model
SA-DES	Spalart-Allmaras Detached Eddy Simulation
SA-DDES	Spalart-Allmaras Delayed Detached Eddy Simulation
SA-DDES ₁₆	modified SA-DDES model
SA-IDDES	Spalart-Allmaras Improved Delayed Detached Eddy Simulation
DD	Domain Decomposition
ZDES	zonal DES
ALE	arbitrary-Lagrangian-Eulerian technique
SG	small grid
LG	large grid
HDG	hole definition grid
FNG	Flügel neuer Generation (new generation airfoil)
BFS	backward facing step

Bibliography

- [1] Abate, M.: "Gust boundary condition for the DLR TAU-Code," Technical Report IB 124-2007/9, DLR, Brunswick, Germany, 2007.
- [2] Adaramola, M. S., Sumner, D., Bergstrom, D. J.: "Effect of velocity ratio on the streamwise vortex structures in the wake of a stack", *Journal of Fluids and Structures*, Vol. 26, No. 1, pp. 1-18, 2010.
- [3] Allen, A., Breitsamter, C.: "Experimental investigation of counter-rotating four vortex aircraft wake", *Aerospace Science and Technology*, Vol. 13, No. 2-3, pp. 114-129, 2009.
- [4] Auerswald, T., Bange, J., Weinman, K., Knopp, T., Radespiel, R., Weinreis, C., Raasch, S.: "Large-Eddy Simulations of the atmospheric boundary layer initialised with synthetic turbulence", 19th Symposium on Boundary Layers and Turbulence in Keystone, Colorado, USA, August 2010.
- [5] Banks, J. W., Henshaw, W. D., Shadid, J. N.: "An evaluation of the FCT method for high-speed flows on structured overlapping grids", submitted for publication, 2008.
- [6] Banks, J. W., Schwendeman, D. W., Kapila, A. K., Henshaw, W. D.: "A high-resolution Godunov method for compressible multi-material flow on overlapping grids", *Journal of Computational Physics*, Vol. 223, pp. 262-297, 2007.
- [7] Basso, E., Antunes, A. P., Azevedo, J. L. F.: "Chimera simulations of supersonic flows over a complex satellite launcher configuration", *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, pp. 345-355, 2003.
doi: 10.2514/2.3969
- [8] Benarafa, Y., Cioni, O., Ducros, F., Sagaut, P.: "RANS/LES coupling for unsteady turbulent flow simulation at high Reynolds number on coarse meshes", *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 23-24, pp. 2939-2960, 2006.
- [9] Benek, J. A., Buning, P. G., Steger, J. L.: "A 3-D Chimera grid embedding technique", AIAA paper 83-1944, American Institute of Aeronautics and Astronautics, Reston, VA, pp. 373-382, 1983.

- [10] Benek, J. A., Steger, J. L., Dougherty, F. C.: "A flexible grid embedding technique", AIAA paper 85-1523, American Institute of Aeronautics and Astronautics, Reston, VA, pp. 322-331, 1985.
- [11] Birken, P., Meister, A.: "Stability of preconditioned finite volume schemes at low mach numbers", BIT Numerical Mathematics, Vol. 45, No. 3, pp. 463-480, Springer Netherlands, 2005.
- [12] Bobylev, A., Vyshinsky, V., Soudakov, G. Yaroshevsky, V.: "Aircraft vortex wake and flight safety problems", Journal of Aircraft, Vol. 47, No. 2, pp. 663-674, 2010.
doi: 10.2514/1.46432
- [13] Brandt, A.: "Multigrid techniques 1984 guide with applications to fluid dynamics", VKI lecture series 1984-04, 1984.
- [14] Bull, M. K., Blazewicz, A. M., Pickles, J. M, Bies, D. A.: "Interaction between a vortex wake and an immersed rectangular plate", Experimental Thermal and Fluid Science, Vol. 12, No. 2, pp. 209-220, 1996.
- [15] Bunge, U., Mockett, C., Thiele, F.: "Guidelines for implementing Detached-Eddy Simulation using different models", Aerospace Science and Technology, Vol. 11, No. 5, pp. 376-385, 2007.
- [16] Carlsson, L., Petersson, N. A.: "Optimizing Chimera grids using genetic algorithms", Submitted to International Journal of Numerical Methods, 2001.
- [17] Carmer, C. F. v., Konrath, R., Schröder, A., Monnier, J.-C.: "Identification of vortex pairs in aircraft wakes from sectional velocity data", Experiments in Fluids, Vol. 44, pp. 367-380, 2008.
- [18] Chan, W. M.: "Overset grid technology development at NASA Ames Research Center", Computers and Fluids, Vol. 24, No. 3, pp. 496-503, 2009.
- [19] Chan, W. M., Bunin, P. G.: "Surface grid generation methods for overset grids", Computers and Fluids, Vol. 24, No. 5, pp. 509-522, 1995.
- [20] Chesshire, G., Henshaw, W. D.: "Composite overlapping meshes for the solution of partial differential equations", Journal of Computational Physics, Vol. 90, pp. 1-64, 1990.
- [21] Cho, K. W., Kwon, J. H., Lee, S.: "Development of a fully systemized Chimera methodology for steady/unsteady problems", Journal of Aircraft, Vol. 36, No. 6, pp. 973-980, 1999.
- [22] Coton, F. N., Marshall, J. S., Galbraith, R. A. McD., Green, R. B.: "Helicopter tail rotor orthogonal blade vortex interaction", Progress in Aerospace Sciences, Vol. 40, No. 7, pp. 453-486, 2004.

- [23] Davidson, L.: "Evaluation of the SST-SAS model: channel flow, asymmetric diffuser and axi-symmetric hill", ECCOMAS CFD 2006, Egmond aan Zee, The Netherlands, September 5-8, 2006.
- [24] Deck, S., : "Zonal-Detached-Eddy Simulation of the Flow Around a High-Lift Configuration", AIAA Journal, Vol. 43, No. 11, pp. 2372-2384, 2005.
- [25] Desquesnes, G., Terracol, M., Manoha, E., Sagaut, P.: "On the use of a high order overlapping grid method for coupling in CFD/CAA", Journal of Computational Physics, Vol. 220, No. 1, pp. 355-382, 2006.
- [26] Dietz, M., Kessler, M., Krämer E., Wagner, S.: "Tip vortex conservation on a helicopter main rotor using vortex-adapted Chimera grids", AIAA Journal, Vol. 45, No. 8, 2007.
- [27] Driver, D. M., Seegmiller, H. L.: "Features of a reattaching turbulent shear layer in divergent channel flow", AIAA Journal, Vol. 23, No. 2, pp. 163-171, 1985.
- [28] Dwight, R.: "Time-accurate Navier-Stokes calculations with approximately factored implicit schemes", Computational Fluid Dynamics 2004, edited by Groth, C. and Zingg, D. W., Springer, pp. 211-218, 2004.
- [29] Ehrenstein, U., Le Dizes, S.: "Relationship between corotating vortex-pair equilibria and a single vortex in an external deformation field", Physics of Fluids, Vol. 17, pp. 074103.1-074103.7, 2005.
- [30] Fabre, D., Sipp, D., Jacquin, L.: "Kelvin waves and the singular modes of the Lamb-Oseen vortex", Journal of Fluid Mechanics, Cambridge University Press, Cambridge, England, Vol. 551, pp. 235-274, 2006.
doi: 10.1017/S0022112005008463
- [31] Feireisl, E., Novotny, A.: "Singular limits in thermodynamics of viscous fluids", Advances in Mathematical Fluid Mechanics, Birkhäuser, Basel, 2009.
- [32] Fischer, T.: "Wavelet-Transformation von instationären Wirbeln und turbulenten Strömungsvorgängen", diploma thesis, Institute for aerodynamics and gas dynamics of the university Stuttgart, 1997.
- [33] Frech M., Holzäpfel F.: "A probabilistic prediction scheme for wake vortex evolution in a convective boundary layer", Air Traffic Control Quarterly, Vol. 10, No. 1, pp. 23-41, 2002.
- [34] Frech M., Holzäpfel F., Gerz T., Konopka, J.: "Short-term prediction of the horizontal wind vector within a wake vortex warning system", Meteorol. Appl., Vol. 9, No. 1, pp. 9-20, 2002.
- [35] Fröhlich, J., von Terz, D.: "Hybrid LES/RANS methods for the simulation of turbulent flows", Progress in Aerospace Sciences, Vol. 44, No. 5, pp. 349-377, 2008.

- [36] Fu, S., Xiao, Z., Chen, H., Zhang, Y., Huang, J.: "Simulation of wing-body junction flows with hybrid RANS/LES methods", *International Journal of Heat and Fluid Flow*, Vol. 28, No. 6, pp. 1379-1390, 2007.
- [37] Gerz T., Holzäpfel F., Darracq D.: "Commercial aircraft wake vortices", *Progress in Aerospace Sciences*, Vol. 38, No. 3, pp. 181-208, 2002.
- [38] Glowinski, R., Pan, T.-W., Periaux, J.: "Fictitious domain methods for the Dirichlet problem and its generalisation to some flow problems", In: *Finite Element in Fluids, New Trends and Applications*, pp. 347-368, Pineridge Press, Barcelona, 1993.
- [39] Glowinski, R., Pan, T.-W., Periaux, J.: "A fictitious domain method for Dirichlet problems and applications", *Comp. Meth. Appl. Mechan. Eng.*, Vol. 111, pp. 203-303, 1994.
- [40] Hagland, B., Skaflestad, B.: "A survey of some methods for moving grid and grid adaptation", *Numerics No. 2*, Norwegian University of Science and Technology, 2002.
- [41] Hariharan, N.: "Rotary-wing wake capturing: high-order schemes towards minimizing numerical vortex dissipation", *Journal of Aircraft*, Vol. 39, No. 5, 2002.
- [42] Henshaw, W. D.: "A high-order accurate parallel solver for Maxwell's equations on overlapping grids", *SIAM Journal of Scientific Computing*, Vol. 28, pp. 1730-1765, 2006.
- [43] Henshaw, W. D., Schwendeman, D. W.: "An adaptive numerical scheme for high-speed reactive flow on overlapping grids", *Journal of Computational Physics*, Vol. 191, pp. 420-447, 2003.
- [44] Henshaw, W. D., Schwendeman, D. W.: "Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow", *Journal of Computational Physics*, Vol. 216, pp. 744-779, 2006.
- [45] Henshaw, W. D., Schwendeman, D. W.: "Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement", *Journal of Computational Physics*, Vol. 227, pp. 7469-7502, 2008.
- [46] Holst, T. L.: "Chimera donor cell search algorithm suitable for solving the full potential equation", *Journal of Aircraft*, Vol. 37, No. 1, pp. 76-84, 2000.
- [47] Holzäpfel F.: "A probabilistic two-phase wake vortex decay and transport model", *Journal of Aircraft*, Vol. 40, No. 2, pp. 323-331, 2003.
- [48] Holzäpfel F., Gerz, T.: "Two-dimensional wake vortex physics in the stably stratified atmosphere", *Aerospace Science and Technology*, Vol. 3, No. 5, pp. 261-270, 1999.

- [49] Holzäpfel, F. (editor), Gerz, T. (project Manager): "The DLR project Wirbelschlepe - detecting, characterizing, controlling, attenuating, understanding, and predicting aircraft wake vortices", DLR-Forschungsbericht 2008-15, Oberpfaffenhofen, Berlin, Braunschweig und Göttingen, ISSN 1434-8454, 139 p., 2008.
- [50] Holzäpfel, F., Gerz, T., Frech, M., Dörnbrack, A.: "Wake vortices in a convective boundary layer and their influence on following aircraft", *Journal of Aircraft*, Vol. 37, No. 6, pp. 1001-1007, 2000.
- [51] Holzäpfel F., Hofbauer T., Darracq D., Moet H., Garnier F., Ferreira Gago C.: "Analysis of wake vortex decay mechanisms in the atmosphere", *Aerospace Science and Technology*, Vol. 7, No. 4, pp. 263-275, 2003.
- [52] Holzäpfel, F., Hofbauer, T., Gerz, T., Schumann, U.: "Aircraft wake vortex evolution and decay in idealized and real environments: methodologies, benefits and limitations", *Fluid Mechanics and its Applications*, Vol. 65, Kluwer Academic Publishers, Dordrecht, pp. 293-309, 2002.
- [53] Houzeaux, G.: "A geometrical domain decomposition method in computational fluid dynamics", PhD thesis, 2008.
- [54] Houzeaux, G., Codina, C.: "A Chimera method based on a Dirichlet/Neumann(Robin) coupling for the Navier-Stokes equations", *Computer Methods in Applied Mechanics and Engineering*, Vol. 192, No. 31-32, pp. 3343-3377, 2003.
- [55] Jakubek, D. "Parameterstudien für numerische RANS- und DES-Verfahren für ein Drei-Element Profil in Hochauftriebskonfiguration mit dem DLR TAU-code", diploma thesis, RWTH Aachen, 2010.
- [56] Jakirlic, S., Kadavelil, G., Kornhaas, M., Schäfer, M., Sternel, D. C. Tropea, C.: "Numerical and physical aspects in LES and hybrid LES/RANS of turbulent flow separation in a 3-D diffuser", *International Journal of Heat and Fluid Flow*, 2010.
- [57] Jameson, A.: "Transonic flow calculations", MAE Report 1651, Princeton University, Princeton, New Jersey, 1983.
- [58] Jossier, C., Rossia, M.: "The merging of two co-rotating vortices: a numerical study", *European Journal of Mechanics - B/Fluids*, Vol. 26, No. 6, pp. 779-794, 2007.
- [59] Jameson, A., Schmidt, W., Turkel, E.: "Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes", *AIAA Paper* 81-1259, 1981.
- [60] Jameson, A., Turkel, E.: "Implicit Scheme and LU-decompositions", *Mathematics of Computation*, Vol. 37, pp. 385-397, 1981.

- [61] Kim, I., Elghobashi, S., Sirignano, W. A.: "Unsteady flow interactions between a pair of advected vortex tubes and a rigid sphere", *International Journal of Multiphase Flow*, Vol. 23, No. 1, pp. 1-23, 1997.
- [62] Kimm, J.-H.: "Additive Schwarz & multilevel methods", Department of Mathematics, Louisiana State University.
- [63] Knopp, T., "Validation of the turbulence models in the DLR TAU code for transonic flows - a best practice guide", *DLR-Forschungsbericht 2006-01*, 107 p., ISSN 1434-8454, 2006.
- [64] Knopp, T., Zhang, X., Kessler, R., Lube, G.: "Enhancement of an industrial finite-volume code for Large-Eddy-Type Simulation of incompressible high-Reynolds number flow using near-wall modelling", *Computer Methods in Applied Mechanics and Engineering*, 2009.
- [65] Langer, S.: "Investigation of preconditioning techniques for the iteratively regularized Gauss-Newton method for exponentially ill-posed problems", *SIAM Journal on Scientific Computing*, Vol. 32, No. 5, pp. 2543-2559, 2010.
- [66] Leclercq, D. J. J., Doolan, C. J.: "The interaction of a bluff body with a vortex wake", *Journal of Fluids and Structures*, Vol. 25, No. 5, pp. 867-888, 2009.
- [67] Li, C. W., Yu, L. H.: "Hybrid LES/RANS modelling of free surface flow through vegetation", *Computers and Fluids*, Vol. 39, No. 9, pp. 1722-1732, 2010.
- [68] Liao, W., Cia, W., Tsai, H. M.: "A multigrid overset grid flow solver with implicit hole cutting method", *Computer Methods in Applied Mechanics and Engineering*, No. 196, Vol. 9-12, pp. 1701-1715, 2007.
- [69] Linke, A.: "Verification and validation of von Kármán length scale for identification of turbulent structures", *German Aerospace Center, IB 224-2009 A 34*, diploma thesis, 2009.
- [70] Matsuno, K., Yamakawa, M., Satofuka, N.: "Overset adaptive-grid method with applications to compressible flows", *Computers and Fluids*, Vol. 27, No. 5-6, pp. 599-610, 1998.
- [71] Mavriplis, D. J., Jameson, A., Martinelli, L.: "Multigrid solution of the Navier-Stokes equations on triangular meshes", *Technical Report, ICASE-report No. 89-35*, 1989.
- [72] Meister, A.: "Numerik linearer Gleichungssysteme", Second edition, Vieweg, 2005.

- [73] Meleshko V. V., Gurzhi A. A., Dörnbrack A., Gerz T., Holzäpfel F., Hofbauer T.: "Interaction of two-dimensional trailing vortex pair with a shear layer", *International Applied Mechanics*, Vol. 37, No. 7, pp. 948-957, Russian original: *Prikladnaya Mekhanika*, Vol. 37, No. 7, pp. 128-136, 2001.
- [74] Menter, F.: "Zonal two equation $k-\omega$ turbulence models for aerodynamic flows", AIAA 93-2906, 1993.
- [75] Menter, F., Egorov, Y.: "Revisiting the turbulent scale-equation", in: *IUTAM Symposium on One Hundred Years of Boundary Layer Research*, Vol. 129, pp. 279-290, Springer Netherlands, 2006.
- [76] Menter, F., Egorov, Y.: "Development and application of SST-SAS turbulence model in the DESIDER project", in: *Advances in Hybrid RANS-LES Modelling*, Vol. 97, pp. 261-270, Springer, 2008.
- [77] Menter, F., Kuntz, M., Langtry, R.: "Ten years of industrial experience with the SST turbulence model", in: Hanjalic, K., Nagano, Y., Tummers, M. (Eds.): *Turbulence, Heat and Mass Transfer 4*, Begell House, pp. 625-632, 2003.
- [78] Mertins, M., Elsholz, E., Barakat, S., Colak, B.: "3D viscous flow analysis on wing-body-aileron-spoiler configurations", *Aerospace Science and Technology*, Vol. 9, No. 6, pp. 476-484, 2005.
- [79] Mikhail L., Spalart, P. R., Strelets, M. K., Travin, A. K.: "A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities", *International Journal of Heat and Fluid Flow*, Vol. 29, No. 6, pp. 1638-1649, 2008.
- [80] Miller, G. C., Williamson, C. H. K.: "Turbulent structures in the trailing vortex wake of a delta wing", *Experimental Thermal and Fluid Science*, Vol. 14, No. 1, pp. 2-8, 1997.
- [81] Moin, P., Squires, K., Cabot, W., Lee, S.: "A dynamic subgrid-scale model for compressible turbulence and scalar transport", *Physics of Fluids A: Fluid Dynamics*, Vol. 3, No. 7, pp. 1760-1765, 1991.
- [82] Morton, S.: "Detached-Eddy Simulations of vortex breakdown over a 70-degree Delta wing", *Journal of Aircraft*, Vol. 46, No. 3, pp. 746-755, 2009.
doi: 10.2514/1.4659
- [83] Oden, J. T., Vemaganti, K.: "Estimation of local modeling error and goal-oriented adaptive modeling of heterogeneous materials part I: error estimates and adaptive algorithms", *Journal of Computational Physics*, Vol. 164, pp. 22-47, 2000.
- [84] Pahlke, K., van der Wal, B. G.: "Chimera simulations of multibladed rotors in high-speed forward flight with weak fluid-structure-coupling", *Aerospace Science and Technology*, Vol. 9, No. 5, pp. 379-389, 2005.

- [85] Pandya, S., Onufer, J., Chan, W., Klopfer, G.: "Capsule abort recontact simulation", AIAA Paper 2006-3324, 24th AIAA Applied Aerodynamics Conference, San Francisco, USA, 2006.
- [86] Petersson, N. A.: "Hole-cutting for three-dimensional overlapping grids", SIAM Journal on Scientific Computing, Vol. 21, No. 2, pp. 646-665, 1999.
- [87] Pope, S.: "Turbulent flows", Cambridge University Press, reprint, 2007.
- [88] Prewitt, N. C., Belk, D. M., Shyy, W.: "Parallel computing of overset grids for aerodynamic problems with moving objects", Progress in Aerospace Sciences, Vol. 36, No. 2, pp. 117-172, 2000.
- [89] Probst, A., Radespiel, R., Wolf, C., Knopp, T., Schwamborn, D.: "A comparison of Detached-Eddy Simulation and Reynolds-Stress modelling applied to the flow over a backward-facing step and an airfoil at stall", AIAA Conference, 4th-7th Jan. 2010, Orlando, Florida,
- [90] Prosser, R.: "Improved boundary conditions for the Direct Numerical Simulation of turbulent subsonic flows. I. Inviscid flows", Journal of Computational Physics, Vol. 207, pp. 736-768, 2005.
- [91] Radespiel, R., Turkel, E., Kroll, N.: "Assessment of preconditioning methods", DLR Forschungsbericht, 1995.
- [92] Reuss, S.: "Investigation of starting procedures for DES computations on the F15 geometry", 17. DGLR-Fach-Symposium der STAB, 9th-10th Nov. 2010, Berlin.
- [93] Richez, F., Mary, I., Gleize, V., Basdevant, C.: "Near stall simulation of the flow around an airfoil using zonal RANS/LES coupling method", Computers and Fluids, Vol. 37, No. 7, pp. 857-866, 2008.
- [94] Rogers, S. E.: "Progress in high-lift aerodynamic calculations", Journal of Aircraft, Vol. 31, No. 6, pp. 1244-1251, 1994.
doi: 10.2514/3.46642
- [95] Rotta, J. C.: "Turbulente Strömungen. Eine Einführung in die Theorie und ihre Anwendung", Stuttgart, 179 p., B. G. Teubner, 1972.
- [96] Rung, T., Lübecke, H., Franke, M., Xue, L., Fu, S.: "Assessment of explicit algebraic stress models in transonic flow", Engineering Turbulence Modeling and Experiments 4, Elsevier, Amsterdam, pp. 659-668, 1999.
- [97] Sanchez-Rocha, M., Menon, S.: "The compressible hybrid RANS/LES formulation using an additive operator", Journal of Computational Physics, Vol. 228, No. 6, pp. 2037-2062, 2009.

- [98] Schwamborn, D., Gerhold, T., Heinrich, R.: "The DLR TAU-code: recent applications in research and industry", ECCOMAS CFD 2006, edited by Wesseling, P., Onate, E. and Periaux, J., Technical University Delft, The Netherlands, 2006.
- [99] Schwarz, T.: "Ein blockstrukturiertes Verfahren zur Simulation der Umströmung komplexer Konfigurationen", DLR-Forschungsbericht 2005-20, 122 p., ISSN 1434-8454, 2005.
- [100] Schwarz, T.: "The overlapping grid technique for the time-accurate simulation of rotorcraft flows", 31st European Rotorcraft Forum, Royal Aeronautical Society, London, 2005.
- [101] Sides, J., Pahlke, K., Costes, M.: "Numerical simulation of flows around helicopters at DLR and ONERA", Aerospace Science and Technology, Vol. 5, No. 1, pp. 35-53, 2001.
- [102] Simonsen, C. D., Stern, F.: "Verification and validation of RANS manoeuvring simulation of Esso Osaka: effects of drift and rudder angle on forces and moments", Computers and Fluids, Vol. 32, No. 10, pp. 1325-1356, 2003.
- [103] Sitaraman, J., Floros, M., Wissink, A., Potsdam, M.: "Parallel domain connectivity algorithm for unsteady flow computations using overlapping and adaptive grids", Journal of Computational Physics, Vol. 229, No. 12, pp. 4703-4723, 2010.
- [104] Smagorinsky, J.: "General circulation experiments with the primitive equations, I. the basic experiment", Monthly Weather Review, Vol. 91, No. 3, pp. 99-164, 1963.
- [105] Smith, B., Bjorstad, P., Gropp, W.: "Domain decomposition, parallel multi-level methods for elliptic partial differential equations", Cambridge University Press, 1996.
- [106] Spalart, P. R.: "Young person's guide to Detached-Eddy Simulation grids", NASA CR-2001-211032, 2001.
- [107] Spalart, P. R.: "Detached-Eddy Simulation", Annual Review of Fluid Mechanics, Vol. 41, pp. 181-202, 2009.
doi: 10.1146/annurev.fluid.010908.165130
- [108] Spalart, P. R., Allmaras, S. R.: "A one-equation turbulence model for aerodynamic flows", AIAA Paper 92-0439, 1992.
- [109] Spalart, P. R., Deck, S., Shur, M. L.: "A new Version of Detached-Eddy Simulation, resistant to ambiguous grid densities", Theoretical Computational Fluid Dynamics, Vol. 20, pp. 181-195, 2006.

- [110] Spalart, P. R., Jou, W.-H., Strelets, M., Allmaras, S. R.: "Comments on the feasibility of LES for wings, and on hybrid RANS/LES approach", in: Proceedings of first AFOSR international conference on DNS/LES, Rustion, Louisiana, Greyden Press, 1997.
- [111] Squires, K. D.: "Detached-Eddy Simulation: current status and perspectives", Direct and Large-Eddy Simulation, pp. 465-480, Kluwer Academic Publishers, 2004.
- [112] Starius, G.: "Composite mesh difference methods for elliptic boundary value problems", Numer. Math., Vol. 28, pp. 243-258, Springer, 1977.
- [113] Starius, G.: "On composite mesh difference methods for hyperbolic boundary value problems", Numer. Math., Vol. 35, pp. 241-255, Springer, 1980.
- [114] Struijs, R., Jonville, G., Darracq, D., Heinrich, R.: "Inviscid computation of effect of wake vortices on a scale-model airplane", Journal of Aircraft, Vol. 40, No. 1, pp. 100-109, 2003.
doi: 10.2514/2.3063
- [115] Svard, M., Lundberg, J., Nordström, J.: "A computational study of vortex-airfoil interaction using high-order finite difference methods", Computers and Fluids, Vol. 39, No. 8, pp. 1267-1274. 2010.
- [116] Swanson, R. C., Radespiel, R., Turkel, E.: "On some numerical dissipation schemes", Journal of Computational Physics, Vol. 147, pp. 518-544, 1998.
- [117] Le Tallec, P.: "Domain decomposition methods in computational mechanics", in: J. Tinsley Oden (editor): Computational Mechanics Advances, Vol. 1, No. 2, pp. 121-220, NorthHolland, 1994.
- [118] Tang, H. S., Jones, S. C., Sotiropoulos, F.: "An overset-grid method for 3D unsteady incompressible flows", Journal of Computational Physics, Vol. 191, No. 2, pp. 567-600, 2003.
- [119] Togashi, F., Nakahashi, K., Ito, Y., Iwamiya, T., Shimbo, Y.: "Flow simulation of NAL experimental supersonic airplane/booster separation using overset unstructured grids", Computers and Fluids, Vol. 30, No. 6, pp. 673-688, 2001.
- [120] Toselli, A., Widlund, O.: "Domain decomposition methods-algorithms and theory", Vol. 34, Springer series in computational mathematics, Springer, 2005.
- [121] Turkel, E., Vatsa, V. N., Radespiel, R.: "Preconditioning methods for low-speed flows", AIAA-paper 96-2460-CP, 14th AIAA Applied Aerodynamics Conference, New Orleans, USA, June 18-20, 1996.
- [122] Volkow, E. A.: "A finite difference method for finite and infinite regions with piecewise smooth boundaries", Doklady, Vol. 168, No. 6, pp. 744-747, 1966.

- [123] Volkow, E. A.: "The method of composite meshes for finite and infinite regions with piecewise smooth boundaries", *Proc. Steklov Inst. Math*, Vol. 96, pp. 145-185, 1968.
- [124] Weinman, K.: personal communication, 2010.
- [125] Whale, J., Anderson, C. G., Bareiss, R., Wagner, S.: "An experimental and numerical study of the vortex structure in the wake of a wind turbine", *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 84, No. 1, pp. 1-21, 2000.
- [126] Whitfield, D. L., Janus, J. M.: "Three-dimensional unsteady Euler equations solution using flux vector splitting", *AIAA-Paper 84-1552*, 1984.
- [127] Wilcox, D. C.: "Reassessment of the scale determining equation for advanced turbulence models", *AIAA Journal*, Vol. 26, No. 11, pp. 1299-1310, 1988.
- [128] Wilcox, D. C.: "Turbulence modeling for CFD", Third edition, DCW Industries, 2006.
- [129] Williamson, C. H. K.: "Three-dimensional vortex dynamics in bluff body wakes", *Experimental Thermal and Fluid Science*, Vol. 12, No. 2, pp. 150-168, 1996.
- [130] Wokoeck, R., Krimmelbein, N., Ortmanns, J., Ciobaca, C., Radespiel, R., Krumbein, A.: "RANS simulation and experiments on the stall behaviour of an airfoil with laminar separation bubbles", in: *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, 2006.
- [131] Wolf, C.: "Lokal adaptive Mehrskalenalgorithmen für Einphasenströmungen in porösen Medien", Freiburg, diploma thesis, 2005.
- [132] Wu, Z.-N.: "Uniqueness of steady-state solutions for difference equations on overlapping grids", *SIAM Journal on Numerical Analysis*, Vol. 33, No. 4, pp. 1336-1357, 1996.
- [133] Wu, J., Sheridan, J., Hourigan, K., Soria, J.: "Shear layer vortices and longitudinal vortices in the near wake of a circular cylinder", *Experimental Thermal and Fluid Science*, Vol. 12, No. 2, pp. 169-174, 1996.
- [134] Zhang, X., Ni, S., He, G.: "A pressure-correction method and its applications on an unstructured Chimera grid", *Computers and Fluids*, Vol. 37, No. 8, pp. 993-1010, 2008.
- [135] Zhang, Q., Schröder, W., Meinke, M.: "A zonal RANS-LES method to determine the flow over a high-lift configuration", *Computers and Fluids*, Vol. 39, No. 7, pp. 1241-1253, 2010.

- [136] Zheng, Y., Liou, M.-S.: "A novel approach of three-dimensional hybrid grid methodology: Part 1. Grid generation", *Computer Methods in Applied Mechanics and Engineering*, Vol. 192, No. 37-38, pp. 4147-4171, 2003.

Acknowledgements

I want to thank many people who helped me directly or indirectly with the realisation of this thesis. Firstly, I am deeply grateful to my doctoral adviser Prof. Dr. Gert Lube for giving me the opportunity to pursue a PhD in mathematics. The supervision can only be described as exemplary and I would always again choose him as my adviser. Moreover I am very thankful to Dr. Tobias Knopp who strongly assisted me during the time of this thesis. Without his guidance, motivation and ideas this thesis would not have been possible.

Special thanks is dedicated to Axel Probst from TU Braunschweig for the close and highly efficient cooperation regarding the testcases shown in Sections 5 and 6. I also want to thank Axel's boss and head of the DFG FOR 1066 research group Prof. Dr. Rolf Radespiel who provided many useful ideas and suggestions. I want to thank Dr. Thorsten Schwarz, Dr. Ralf Heinrich and Axel Raichle (all from DLR Braunschweig) for the immense help with the Chimera technique - both regarding understanding, implementation and practical use.

I am very thankful to the complete DLR AS-C²A²S²E group in Göttingen to enable me working within this interesting and sophisticated field of research. Besides the already mentioned Dr. Tobias Knopp I especially want to thank Dr. Keith Weinman for his numerous best practice hints of how to use the DLR TAU-Code, Dr. Dieter Schwamborn for giving me the possibility to learn very much and also for his support and guidance and my former office roommate Dr. Mariafrancesca Valentino for many valuable discussions and practical advises.

Finally I want to thank my family and especially my wife Anna and our two wonderful children Milena and Bruno. Without their support I would not have managed to accomplish this thesis.

Lebenslauf

Persönliche Daten

Name: Christoph Wolf
Geburtsdatum: 06.12.1977
Geburtsort: Karlsruhe, Deutschland
Familienstand: verheiratet, zwei Kinder
Nationalität: deutsch

Schulbildung

1982 - 1986 Grundschole in Winkelhaid bei Nürnberg
1986 - 1995 Gymnasium Lauf a. d. Pegnitz
1995 - 1996 Livingston High School in Texas, USA
1996 - 1998 Gymnasium Lauf a. d. Pegnitz
Mai 1998 Abitur

Zivildienst

1998 - 1999 Zivildienst im Blindeninstitut in Rückersdorf bei Nürnberg

Studium und praktische Arbeitserfahrung

1999 - 2001 Studium an der Albert-Ludwigs-Universität Freiburg
Hauptfach: Informatik, abgeschlossen mit Vordiplom

2000 - 2006 Studium an der Albert-Ludwigs-Universität Freiburg
Hauptfach: Mathematik, Studienschwerpunkt: Numerik
Abgeschlossen mit Diplom

2003 - 2006 Mitarbeiter am C++-Programmiersprojekt DUNE (Distributed and Unified Numerics Environment)
(siehe auch <http://dune.mathematik.uni-freiburg.de>)

2006- 2010 wissenschaftlicher Mitarbeiter beim Deutschen Zentrum für Luft- und Raumfahrt e. V. in Göttingen. Insbesondere Arbeiten im Rahmen der DFG FOR 1066 "Simulation des Überziehens von Tragflügeln und Triebwerksgondeln"
(siehe auch <http://www.for1066.tu-bs.de>)

2008 - 2010 Promotion am Institut für numerische und angewandte Mathematik der Georg-August-Universität Göttingen bei Prof. Dr. G. Lube