

Stellingen
behorende bij het proefschrift

An Industrially Applicable Solver
for Compressible, Turbulent Flows

van

JOHAN KOK

De nauwkeurigheid in grenslagen van de numerieke methode van Jameson voor compressibele stromingen kan sterk verbeterd worden door het toepassen van een specifieke matrix-vorm van artificiële diffusie – waarbij de artificiële diffusie enkel in de richting loodrecht op de wand verminderd wordt – zonder dat dit een toename van de rekenkosten tot gevolg heeft (zie dit proefschrift).

★

Bij de ontwikkeling van numerieke methoden voor complexe, niet-lineaire vergelijkingen, zoals de Reynolds-gemiddelde Navier-Stokes vergelijkingen, is het uitvoeren van numerieke experimenten essentieel.

★

Voor zeer complexe geometriën vergt het uitvoeren van een stromingssimulatie m.b.v. een multi-block flow solver een dergelijk grote inspanning, dat het gevaar dreigt dat men het verkrijgen van een oplossing al als een succes beschouwt, onafhankelijk van de nauwkeurigheid van deze oplossing.

★

Indien men een stelsel vergelijkingen dimensieloos maakt, dient het aantal schalingsgrootheden gelijk te zijn aan het aantal fysische dimensies in het probleem. De vorm van de dimensieloze vergelijkingen is dan gelijk aan die van de oorspronkelijke vergelijkingen. Van deze regel wordt vaak afgeweken waardoor de dimensieloze variabelen en vergelijkingen niet meer uniek gedefinieerd zijn en als gevolg daarvan de kans op fouten toeneemt.

★

Aangezien de stad het domein is van voetgangers en fietsers, en derhalve in het verkeer hun veiligheid voorop hoort te staan, dient het snelverkeer óf volledig gescheiden te worden van het langzaam verkeer, óf zich volledig aan te passen aan het langzaam verkeer. Zolang dit niet het geval is, is het terecht dat de automobilist, als veroorzaker van deze onveilige situatie, volledig aansprakelijk wordt gesteld voor eventuele ongelukken.

★

Mensen die over alles een mening en over iedereen een oordeel hebben, hebben over niets nagedacht.

★

Niets is zo verwonderlijk als het feit dat er iets is in plaats van niets.

3104

2395113

TR 3105

697017/100000

7000000000

An Industrially Applicable Solver for Compressible, Turbulent Flows

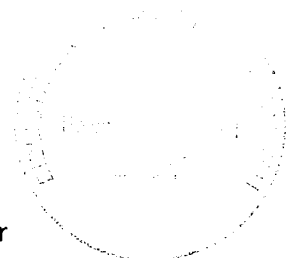
PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,
op maandag 23 maart 1998 te 13.30 uur

door

JOHANNES CHRISTIAAN KOK

wiskundig ingenieur
geboren te Leeuwarden



Dit proefschrift is goedgekeurd door de promotoren:

Prof. dr. ir. J.W. Boerstael

Prof. dr. ir. A.J. Hermans

Samenstelling promotiecommissie:

Rector Magnificus,

Prof. dr. ir. J.W. Boerstael,

Prof. dr. ir. A.J. Hermans,

Prof. dr. A.E.P. Veldman,

Prof. dr. ir. F.T.M. Nieuwstadt,

Prof. dr. ir. H.W.M. Hoeijmakers,

Prof. ir. J.W. Slooff,

Prof. dr. H. Deconinck,

voorzitter

Technische Universiteit Delft, promotor

Technische Universiteit Delft, promotor

Rijksuniversiteit Groningen

Technische Universiteit Delft

Technische Universiteit Delft

Technische Universiteit Delft en

Nationaal Lucht- en Ruimtevaartlaboratorium

Université Libre de Bruxelles en

Von Karman Institute for Fluid Dynamics

Gepresenteerde resultaten zijn verkregen onder contracten toegekend door het Nederlands Instituut voor Vliegtuigontwikkeling en Ruimtevaart, contractnummers 01105N, 03301N en 03401N.

Copyright ©1998 by J.C. Kok, Amsterdam, the Netherlands. All rights reserved.
Cover design by Ronald Nonnekens.

SUMMARY

In this thesis, a flow solver for the steady, compressible, Reynolds-averaged Navier-Stokes equations is discussed. This solver is intended for industrial applications, in particular for simulating compressible, turbulent flows around transport-type aircraft to support the integration of propulsion systems with wing-body configurations. Starting point for the development of the flow solver is an existing solver for the steady, compressible Euler equations. Viscous and turbulent effects are incorporated by means of the thin-layer Reynolds-averaged Navier-Stokes equations and algebraic turbulence models. Thus, the wing boundary layer (and its effect on the wing pressure distribution) is captured, while growth potential to more general viscous flows is ensured.

Since the existing Euler flow solver has been accepted and used extensively by industry, a maximum reuse of methods and software is required. This solver was based on the use of multi-block structured grids, so that arbitrary complex geometries could be considered. The Euler equations were discretized in space by a cell-centred finite-volume scheme using central differences and explicit scalar artificial diffusion. The discrete equations were solved by a pseudo-time integration method using explicit Runge-Kutta schemes and implicit residual averaging.

Considering at first a straightforward extension of the numerical scheme for the Euler equations to the Navier-Stokes equations, further improvements are needed. A matrix artificial-diffusion scheme is employed in order to reduce the grid dependency of the numerical solution in boundary layers, without a significant reduction of convergence speed or increase of computation time compared to scalar artificial diffusion. An efficient solution procedure is obtained by using a multi-grid scheme in order to accelerate the pseudo-time integration method. A robust scheme for multi-block grids with cells of high aspect ratio is obtained by using the concept of multi-block inside multi-grid, a W-cycle multi-grid scheme with five pre- and five post-relaxations, and high-aspect-ratio scaling of artificial diffusion and residual averaging. Also, attention is given to a robust (numerical) implementation of the algebraic turbulence models. In particular, for the Johnson-King model, attention is given to detailed problems related to the extension of the model from 2D to 3D and to its implicit algebraic relations.

Finally, an assessment is made of the numerical accuracy and efficiency of the Navier-Stokes solver, and its applicability to typical transport-type aircraft configurations is demonstrated.

ACKNOWLEDGEMENTS

The research presented in this thesis as well as the writing of the thesis itself could not have been done without the support and contributions of a great number of people.

In particular, I would like to thank Prof. Hermans and Prof. Boerstael for their supervision and guidance in writing this thesis; Bert Kassies and Stefan Spekreijse, with whom I've worked together with much pleasure in the ENFLOW-project for many years; Marcello Amato, who made essential contributions to the development of ENSOLV in the first years; Ronny Groothuizen and Wout Loeve, for their never-ceasing encouragement to write this thesis and their critical review of draft versions; Bas Oskam, for his general support of the ENFLOW-project; Rob Hagmeijer, with whom I've enjoyed working in several projects in which ENSOLV was used; Martin Laban, for sharing some of his knowledge on aerodynamics and as a very critical user of the ENFLOW-system; and further all those who made some contribution to the development of the ENFLOW-system or as user really put the system to the test.

I am indebted to the management of the National Aerospace Laboratory NLR for giving me the opportunity to write this thesis and for allowing me to use research conducted at NLR.

In this thesis, results are presented that have been obtained under a contract awarded by the Netherlands Agency for Aerospace Programs NIVR, within the framework of the ARP and VTP research programs, for which the NIVR kindly gave permission.

CONTENTS

SUMMARY	III
ACKNOWLEDGEMENTS	V
1 INTRODUCTION	1
1.1 General overview of Euler flow solver	2
1.2 Modelling of turbulent boundary layers and wakes	3
1.3 Requirements and project constraints	4
2 FLOW MODEL	9
2.1 Introduction	9
2.2 The Reynolds-averaged Navier–Stokes equations	10
2.3 The thin-layer Reynolds-averaged Navier–Stokes equations	13
2.4 The Euler equations	15
2.5 Boundary Conditions	16
2.5.1 Solid surface	17
2.5.2 Far-field boundary	17
2.5.3 Symmetry plane	19
2.5.4 Plane of infinite continuation	19
2.6 Flow parameters and dimensionless groups	19
2.7 Summary	21
3 SPACE DISCRETIZATION	23
3.1 Introduction	23
3.2 Literature review	24
3.3 Multi-block approach and domain discretization	26
3.4 Basic scheme	28
3.4.1 Convective fluxes	30
3.4.2 Diffusive fluxes for the TLNS equations	31
3.4.3 Scalar artificial diffusion model	32
3.5 Boundary conditions	34
3.5.1 Solid surface	35
3.5.2 Far-field boundary	37
3.5.3 Symmetry plane	38
3.5.4 Plane of infinite continuation	38
3.5.5 Internal faces	38
3.6 Extensions needed for the Navier–Stokes equations	42
3.6.1 Matrix artificial diffusion model	42
3.6.2 High-aspect-ratio scaling of the artificial diffusion	45
3.7 Results	47

4	SOLUTION PROCEDURE	53
4.1	Introduction	53
4.2	Literature review	53
4.3	Basic solution procedure	55
4.3.1	Runge-Kutta time integration	55
4.3.2	Local time stepping	57
4.3.3	Implicit residual averaging	58
4.4	Extensions needed for the Navier-Stokes equations	60
4.4.1	Multi-grid scheme	60
4.4.2	Treatment of high-aspect-ratio grid cells	68
4.5	Results	69
5	TURBULENCE MODELLING	81
5.1	Introduction	81
5.2	Review of turbulence models	81
5.3	The Cebeci-Smith turbulence model	84
5.4	The Baldwin-Lomax turbulence model	86
5.5	The Johnson-King turbulence model	88
5.6	Numerical implementation	96
5.6.1	Computation of boundary-layer and displacement thickness	96
5.6.2	Computation of maxima and their locations	97
5.6.3	Transition	99
5.6.4	Numerical method for the Johnson-King model	99
5.7	Results	108
6	VERIFICATION AND DEMONSTRATION	113
6.1	Verification of numerical accuracy	113
6.2	Verification of efficiency	116
6.3	Verification of robustness	117
6.4	Demonstration	118
6.4.1	VTP airfoil	118
6.4.2	Aerospatiale AS28G wing-body-nacelle configuration	122
6.4.3	VTP wing-body-engine configuration	124
7	CONCLUSIONS AND FINAL REMARKS	129
A	FLUX JACOBIAN FOR THE EULER EQUATIONS	133
B	STABILITY CONSIDERATIONS	137
	BIBLIOGRAPHY	149
	NOMENCLATURE	159
	SAMENVATTING (SUMMARY IN DUTCH)	165
	CURRICULUM VITAE	167

INTRODUCTION

Between 1985 and 1990, a flow solver for the steady, compressible Euler equations was developed at NLR. This solver was intended for the simulation of steady, compressible, inviscid flow around transport-type aircraft in subsonic or transonic cruise conditions, and included means for the simulation of the effect of propulsion systems on the flow. The flow solver was part of a flow-simulation system called ENFLOW, which included tools for grid generation and flow visualization. This system was based on the use of multi-block structured grids, so that arbitrary complex geometries could be considered.

The main application of the Euler flow solver has been to support the integration of propulsion systems with wing-body configurations. One is then primarily interested in the effect that the presence of the propulsion system has on the wing pressure distribution. A strong distortion of this pressure distribution is undesired, since, generally, the wing has been carefully designed to meet certain aerodynamic requirements, usually in terms of lift, drag, and geometric characteristics. However, for the application region considered, large differences may still exist between the wing pressure distribution predicted by the Euler equations and the measured pressure distribution (e.g. shock locations may easily be off by 15% of the local chord for a given angle of attack in transonic cruise). The main cause for this difference is, of course, that the displacement effect of the wing boundary layer has been neglected in the Euler flow solver.

Around 1990, it was decided that the Euler flow solver should be extended with means for simulating the effect of the wing boundary layer on the wing pressure distribution. For transport aircraft in cruise conditions, these boundary layers are generally turbulent. The choice was made to model the wing boundary layer using the thin-layer Reynolds-averaged Navier-Stokes (TLNS) equations, and simple algebraic turbulence models, under the assumption that for the application region, the boundary layers are generally attached or only weakly separated. Although only the wing boundary layer was taken into account and only simple algebraic turbulence models were considered, growth potential towards more complex viscous flows had to be ensured.

The actual extension of the Euler flow solver with the thin-layer Navier-Stokes equations and algebraic turbulence models, resulting in the Euler/Navier-Stokes flow solver ENSOLV, is the topic of this thesis. Since the flow solver is aimed at industrial applications, it is important that the computational method has sufficient numerical accuracy, efficiency, and robustness so that the method may be used reliably within its region of application. This region of application, as for the Euler flow solver, consists

of the simulation of compressible flow around transport-type aircraft in transonic cruise conditions, in particular to support the integration of propulsion systems with wing-body configurations.

Before going into a detailed discussion of the extension, an overview is given of the original Euler flow solver, a short description is given of how the turbulent boundary layers and wakes are modelled, and requirements and project constraints for the extension are formulated.

1.1 GENERAL OVERVIEW OF EULER FLOW SOLVER

A short overview is given of the concepts that were used to develop the existing NLR Euler flow solver (see also Boerstoele et al. [11, 12]). These concepts include a number of major design choices which in general must be made when developing a computational method for flow simulation. These choices are outlined here (without an extensive discussion or justification) as they form the starting point for the development of the Navier-Stokes flow solver ENSOLV:

- The Euler flow solver was aimed at the simulation of steady, compressible, inviscid flow around transport-type aircraft in subsonic or transonic cruise conditions.
- The flow solver was based on the Euler equations. Inviscid flow may also be modelled using potential-flow theory. However, in case of strong shocks, not negligible generation of vorticity (e.g. due to propulsion systems), or complex vortex sheets (for complex configurations), potential-flow theory is difficult to be applied.
- The choice was made to deal with arbitrarily complex geometries, instead of a limited number of simple geometries (e.g. only wing-body configurations). This allowed for the simulation of the flow around complete aircraft, including propulsion systems.
- Boundary conditions were included to simulate the effect of propulsion systems on the flow.
- In order to deal with complex geometries, the solver was based on multi-block structured grids. Such a grid consists of a division of an arbitrary domain into a number of non-overlapping blocks, each with the topology of a cube. Within each block, a structured grid is present, made up of hexahedral cells. An alternative approach could have been the use of unstructured grids, usually existing of tetrahedral cells. In general, for structured grids it is easier to develop accurate and efficient flow-solution algorithms, but at the cost of a more man-hour consuming grid-generation task, compared to unstructured grids. In order to minimize these costs, significant investments have been made in the development of efficient (interactive) means for the generation of multi-block structured grids [104, 105], parallel to the development of the flow solver: ENDOMO for the generation of the multi-block topology, and ENGRID for the generation of the grids within the blocks.

- The equations were discretized by a cell-centred finite-volume scheme using central differencing and explicit artificial diffusion. Following the work of Jameson [52], this artificial diffusion consisted of a blending of second- and fourth-order differences. The finite-volume concept ensures conservation, needed for correct prediction of shocks. A cell-centre scheme was chosen over a cell-vertex scheme, because it results in a simpler scheme at block boundaries (in particular at edges and vertices). For non-smooth grids, cell-vertex schemes can be more accurate than cell-centre schemes. However, the structured grids are required to be smooth within blocks. The Jameson-type scheme, developed for transonic flows, was chosen for its robustness and its cheapness compared to upwind schemes (factor 2 to 3). Upwind schemes, however, are generally more accurate at flow-field discontinuities, resulting in sharper shocks, and are in particular preferred for supersonic and hypersonic flows.
- The discrete equations were solved using an explicit Runge-Kutta time-integration scheme, accelerated by local time stepping, implicit residual averaging, and enthalpy damping. It was argued that such an explicit scheme has a similar efficiency as implicit time-integration schemes for the considered applications (the slower convergence counterbalanced by the lower cost of one time step), while being simpler to implement, in particular for the multi-block concept.

1.2 MODELLING OF TURBULENT BOUNDARY LAYERS AND WAKES

The flow of air around transport aircraft at cruise conditions occurs at high Reynolds numbers, implying that the viscous effects are mostly limited to (thin) boundary layers and wakes, and that the flow in these regions is generally turbulent. The extension of the Euler flow solver is in first instance aimed at capturing the turbulent boundary layer on the wing surface as well as the near wake downstream of the wing, while ensuring growth potential towards capturing all boundary layers and wakes, or towards more complex flows with viscous effects not only limited to such thin layers. At cruise conditions, these wing boundary layers are mostly attached or only weakly separated, while the cross flow in the boundary layer is weak across the largest part of the wing.

The turbulent boundary layers and wakes are here essentially described by the Reynolds-averaged Navier-Stokes (RANS) equations. In this model, the flow is considered to be the sum of a mean flow and turbulent fluctuations. The mean flow is described by the RANS equations, which are obtained by applying an appropriate averaging procedure to the Navier-Stokes equations [20]. Here, only steady mean flow is considered. The RANS equations contain unknown terms (the so-called Reynolds stresses and the turbulent heat flux) which describe the effect of the turbulent fluctuations on the mean flow. A turbulence model is required to model these terms, and thus close the system of equations. The choice was made to use initially simple algebraic turbulence models (in particular the model of Baldwin and Lomax [6]). For attached or weakly separated boundary layers on transport wings, standard algebraic turbulence models give results which are usually hardly distinguishable from those obtained with more generally applicable turbulence models (such as two-equation

models), see for instance Holst [46]. For boundary layers with shock-induced separation, or strong adverse pressure gradients, turbulence models are required which take non-equilibrium effects into account. For this reason, it was also decided to use the model of Johnson and King [56] (which is an algebraic model extended with a partial differential equation along configuration surfaces). Algebraic turbulence models can be easily separated from the other parts of the flow solver, thus allowing them to be replaced by different models in the future.

The thin-layer Reynolds-averaged Navier–Stokes (TLNS) equations are a simplified form of the RANS equations, retaining in the viscous terms only derivatives in the direction normal to configuration surfaces (a reduction of the computational effort in the order of 50% in 3D). As shown by Blottner [10], all terms of the Euler equations, and of the first-order boundary-layer equations are contained in the TLNS equations, but some curvature terms are neglected, compared to the second-order boundary-layer equations. For the TLNS equations, grids may be used with high aspect ratios of the grid cells (small mesh sizes in the direction normal to the configuration surface, and larger mesh sizes in the tangential directions). In order to accurately represent the terms of the full RANS equations that were neglected in the TLNS equations, grids would be required with significantly lower aspect ratios (several orders of magnitude), thus increasing the computational effort considerably. Future extension of the TLNS equations to the full RANS equations is very well possible, and similar (if not identical) numerical techniques may be used for both systems of equations.

Thus, the TLNS equations, together with algebraic turbulence models, may be considered as the least expensive method for simulating the wing boundary layer, while retaining growth potential towards the simulation of the boundary layer on all aircraft components, or the simulation of more complex viscous flows (e.g. with strong separation). Methods based on coupling the Euler equations to a boundary-layer method, although possibly cheaper than solving the TLNS equations, generally lack this growth potential.

1.3 REQUIREMENTS AND PROJECT CONSTRAINTS

In this section, we formulate the main project constraints within which the new flow solver ENSOLV had to be developed, as well as the requirements which the new solver had to satisfy. The project constraints essentially have been described in the two previous sections, and can be summarized as follows.

1. *Region of applicability:* The flow solver shall be used to simulate compressible, turbulent flow (steady in the mean) around transport-type aircraft in subsonic and transonic cruise conditions. The region of applicability is limited to flows for which the following assumptions hold:
 - Viscous and turbulent effects are limited to thin boundary layers and wakes.
 - Wing boundary layers are attached or weakly separated.
 - No strong cross flow occurs in the wing boundary layers, apart from the wing tip region.

For these flows, the main interest concerns the wing pressure distribution, and thus it is in particular the wing boundary layer that must be captured.

2. *Flow model*: The flow solver shall be based on the steady thin-layer Reynolds-averaged Navier–Stokes (TLNS) equations in combination with standard algebraic turbulence models. This allows for the separate simulation of the wing boundary layer and wake, while the flow over other aircraft components (such as the nacelle) is modelled as inviscid.
3. *Starting point*: Starting point for the development of the flow solver shall be an existing solver for the compressible, Euler equations. Since this solver has been accepted and used extensively by industry, a maximum reuse of methods and software is required. Further, the functionality of the existing Euler solver must be retained.
4. *Geometrical complexity*: The flow solver must be able to deal with complex geometries, typically complete (transport) aircraft, as far as the topology is concerned. In fact, since the concept of multi-block structured grids has been adopted to deal with complex geometries, this means that the flow solver must be able to deal with arbitrary multi-block topologies. (Note that this does not imply that the flow solver shall be able to deal with arbitrary (complex) flows.)

If a computational method is to be applied in a routine manner (as part of a design process), it is important that such a method is sufficiently robust and efficient, as well as sufficiently accurate, or at least that its accuracy has been assessed.

For assessment of accuracy, it is important to identify separate error sources. This is done by distinguishing the subsequent phases of the physical and mathematical modelling that are present in developing a numerical simulation method [93, 101]:

1. physical reality,
2. physical model: a conceptual model stating which physical effects are taken into account, and which effects are neglected,
3. continuous mathematical model, introducing representation errors with respect to the physical model,
4. discrete mathematical model, introducing discretization errors,
5. solution procedure, introducing solution errors, and
6. software implementation.

In the context of accuracy assessment, validation is defined as the assessment of the physical relevance of a simulation, i.e. the assessment of the physical modelling errors, consisting of the neglected physical effects and the representation errors. For validation it is necessary to relate to the ‘outside world’: physical simulations or experiments, such as a wind-tunnel measurement (with its own error sources), and other numerical simulation methods. Verification is defined as the assessment of the numerical accuracy of a simulation, i.e. the assessment of the numerical discretization errors, for which reference to the outside world is not needed. Verification presupposes negligible solution errors – i.e. the solution procedure should be sufficiently converged (ideally to machine accuracy) – as well as the absence of implementation errors. Validation presupposes that the discretization errors have been assessed.

The current work focuses on the numerical scheme, which consists of the discrete mathematical model and the solution procedure. The physical and continuous mathematical model are required before hand. Together they are named flow model, and are presented in the following chapter. A general validation of these models lies outside the scope of this thesis. We will, however, perform comparisons with experimental results and results of some other computational methods for a few individual test cases in chapter 6. The software implementation (and the assessment of implementation errors) also lies outside the scope of this thesis. For recent research conducted at NLR into an appropriate software structure for CFD software see Vogels [117].

The following requirements are imposed with respect to the numerical scheme:

1. *Numerical accuracy:* It is required that for practical applications, the numerical errors are at least smaller than the physical modelling errors. For 2D airfoils, the physical accuracy for the lift and drag coefficients for methods based on the RANS equations with algebraic turbulence models lie in the order of 3 to 5% (Holst [46], Haase [40]) for subsonic and transonic conditions with attached or weakly separated boundary layers. For 3D cases, for more complex geometries, and for more difficult flow conditions, the accuracy may certainly be expected to be less. Thus, it is required that it must be possible to keep numerical errors in lift and drag within 5%, while maintaining an acceptable computation time (as specified in the subsequent requirement).
For validation of physical models (in particular turbulence models), it should be possible to obtain numerical errors that are an order of magnitude smaller than the physical modelling errors. Thus, it is also required that it must be possible to keep numerical errors in lift and drag within 0.5%, with a considerable computation time (order of one day at 400 Mflop/s).
2. *Efficiency:* In order for numerical simulations to be useful during a design process, computation times (excluding grid generation) must be of the order of several hours for a complete aircraft configuration (at most ten to allow an overnight computation). For this requirement, a computational speed of 400 Mflop/s is considered, which should be obtainable on a modern supercomputer. Modern supercomputers have computation speeds in the order of several hundred Mflop/s to several Gflop/s (e.g. the NEC-SX4/16 parallel/vector supercomputer has a peak performance of 2 Gflop/s on a single processor).
3. *Robustness:* The computational method must be robust. This means that within the application region converged results must be obtainable without any user interaction (tuning of parameters) as far as the flow solver is concerned. It is not required that the solver shall give converged results for arbitrary grids which may be excessively skew or non-smooth. Instead, parallel to the development of the flow solver, tools for the generation of multi-block structured grids have been developed which deliver smooth grids that are orthogonal near configuration surfaces (i.e. within boundary layers).

In the following chapters, we will first present the continuous flow model. After that, the space discretization, the solution procedure, and turbulence modelling will

be discussed, keeping in mind the three main numerical requirements just formulated. Finally, in chapter 6, a verification of these requirements will be performed, and the application of the flow solver to a few typical aircraft configurations will be demonstrated.

FLOW MODEL

2.1 INTRODUCTION

Consider the compressible, subsonic or transonic flow around a transport-type aircraft. In the introduction, it has been stated that the turbulent boundary layers and wakes of the wing will be modelled by the Reynolds-averaged Navier–Stokes (RANS) equations, and in particular by the thin-layer approximation thereof (TLNS equations). Other regions of the flow will be modelled as inviscid and are described by the Euler equations, consistent with the existing Euler flow solver. In this chapter, all these equations will be described, including boundary conditions and the scaling to make the equations dimensionless. As will be seen later, the RANS equations are a set of unclosed equations, with terms (the Reynolds stresses and the turbulent heat flux) which need to be defined by a so-called turbulence model. As stated in the introduction, algebraic turbulence models are used, which will be discussed in chapter 5.

The three sets of equations considered (RANS, TLNS, and Euler) can be written in the same conservation form, consisting of five partial differential equations describing the conservation of mass, momentum, and total energy. Corresponding to these equations, the density ρ , the momentum vector per unit volume or mass flux $\rho\vec{u}$, and the total energy per unit volume ρE are taken as the basic dependent variables (where $\vec{u} \in \mathbb{R}^3$ is the velocity vector, and E is the total energy per unit mass). These variables form the components of the flow-state vector $U \in \mathbb{R}^5$:

$$U = \begin{pmatrix} \rho \\ \rho\vec{u} \\ \rho E \end{pmatrix}. \quad (2.1)$$

The general conservation form of the equations is then given by

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathcal{F} = 0, \quad (2.2)$$

where the flux matrix $\mathcal{F} \in \mathbb{R}^{5 \times 3}$ depends on the precise equations considered (RANS, TLNS, or Euler) and ∇ is the gradient operator. Since the interest only concerns steady (mean) flows, equation (2.2) reduces to

$$\nabla \cdot \mathcal{F} = 0. \quad (2.3)$$

The independent variables are the time $t \in [0, \infty)$, and the position vector $\vec{x} \in D \subset \mathbb{R}^3$, where D is the flow domain. The boundary ∂D of the flow domain consists of a configuration surface ∂D_c and an outer boundary ∂D_o . Considering a Cartesian reference frame with unit base vectors \vec{e}_i ($i \in \{1, 2, 3\}$), the position vector may be expressed as $\vec{x} = x_i \vec{e}_i$ (using the Einstein summation convention). The Cartesian coordinates (x_1, x_2, x_3) are also sometimes denoted as (x, y, z) . Similarly, the velocity vector and the gradient operator are expressed as $\vec{u} = u_i \vec{e}_i$ and $\nabla = \vec{e}_i \partial / \partial x_i$, respectively. Furthermore, it is assumed that the flow domain is described by a boundary-conforming curvilinear coordinate system $(\xi, \eta, \zeta) \in [0, 1]^3$ (or separate parts of the flow domain can be described by such a system). This coordinate system can be considered to correspond to the grid which is used to discretize the equations.

In order to render the equations dimensionless, at least four reference quantities must be defined to scale all variables, since the equations contain four basic dimensions (time, length, mass, and temperature). When exactly four appropriate reference quantities are used, the scaling of every variable is uniquely defined, and the dimensionless equations have a form identical to the dimensional equations. Here, the four reference quantities are chosen as the free-stream values of density ρ_∞ , pressure p_∞ , and temperature T_∞ , and a relevant length scale L (usually a representative length of the aerodynamic body such as the wing chord). Considering all variables and equations from here on to be dimensionless, this is equivalent to setting these four variables (but now scaled) equal to one:

$$\rho_\infty = 1, \quad p_\infty = 1, \quad T_\infty = 1, \quad L = 1. \quad (2.4)$$

In the following sections, the specific form of the general continuous equation (2.2) are defined for the RANS equations (section 2.2), the TLNS equations (section 2.3), and the Euler equations (section 2.4). To complete the continuous flow model, boundary conditions must be added, which are described in section 2.5. Initial conditions are not needed in the continuous flow model, since only steady flows are considered. Finally, in section 2.6 flow parameters are identified and suitable dimensionless groups are derived.

2.2 THE REYNOLDS-AVERAGED NAVIER-STOKES EQUATIONS

Generally, a turbulent flow can be considered to be the sum of a mean flow and turbulent fluctuations around this mean. The smallest time and length scales of the fluctuations can be very small compared to the scales of the mean flow, so that generally less resolution is required to describe the mean flow than to describe the complete turbulent flow. (See for an order of magnitude of these different scales e.g. Tennekes and Lumley [113].) Furthermore, one is often only interested in mean properties of the flow, and not in all (small-scale) details. Therefore, through an appropriate averaging procedure, equations describing the mean flow have been derived from the general Navier-Stokes equations [14, 20, 84, 113, 123]. The mean-flow equations are called the Reynolds-averaged Navier-Stokes equations, and for a compressible, turbulent

flow are given by (using the Einstein convention)

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} &= 0, \\ \frac{\partial \rho u_i}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} &= \frac{\partial \tau_{ij}}{\partial x_j}, \\ \frac{\partial \rho E}{\partial t} + \frac{\partial(\rho E u_j)}{\partial x_j} + \frac{\partial(p u_j)}{\partial x_j} &= \frac{\partial(u_i \tau_{ij} - Q_j)}{\partial x_j},\end{aligned}\tag{2.5}$$

where p is the static pressure, τ_{ij} are the Cartesian components of the total stress tensor τ , and Q_i are the Cartesian components of the total heat-flux vector \vec{Q} . In these equations, already some simplifying assumptions have been made (see for a discussion [14, 20]). All dependent variables in the RANS equations are actually averaged quantities (time-averaged or Favre-averaged) describing the mean flow.

The RANS equations may also be written in the form of equation (2.2),

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathcal{F}^c = \nabla \cdot \mathcal{F}^d,\tag{2.6}$$

in which the convective flux matrix $\mathcal{F}^c \in \mathbb{R}^{5 \times 3}$ and the diffusive flux matrix $\mathcal{F}^d \in \mathbb{R}^{5 \times 3}$ are given by

$$\mathcal{F}^c = \begin{pmatrix} \rho \vec{u}^T \\ \rho \vec{u} \vec{u}^T + p \mathbf{I} \\ \rho E \vec{u}^T + p \vec{u}^T \end{pmatrix}, \quad \mathcal{F}^d = \begin{pmatrix} 0 \\ \tau \\ \vec{u} \cdot \tau - \vec{Q}^T \end{pmatrix}.\tag{2.7}$$

The term $\vec{u} \vec{u}^T$ is a tensor of order 2 with Cartesian components $u_i u_j$, and the term \mathbf{I} is the unit tensor of order 2 with Cartesian components δ_{ij} (the Kronecker delta).

In order to close the equations, equations of state are needed, expressing the pressure (and internal energy) in terms of the basic dependent variables. Also, the stress tensor and the heat-flux vector must be related to the basic variables.

Assuming a calorically perfect gas, the internal energy e and the pressure p are given by

$$\begin{aligned}e &= c_v T, \\ p &= R \rho T,\end{aligned}\tag{2.8}$$

with T the temperature, c_v the specific heat at constant volume, and R the gas constant. Since the total energy E is the sum of the internal energy and the kinetic energy,

$$E = e + \frac{1}{2} \|\vec{u}\|^2\tag{2.9}$$

(with $\|\cdot\|$ the Euclidean norm), the pressure is directly related to the basic dependent variables by

$$p = (\gamma - 1)\rho e = (\gamma - 1) \left(\rho E - \frac{1}{2} \rho \|\vec{u}\|^2 \right),\tag{2.10}$$

where $\gamma = c_p/c_v$ is the ratio of specific heats at constant pressure (c_p) and constant volume (c_v), and where the relation $R = c_p - c_v$ has been used.

The total stress tensor τ consists of the sum of the viscous stress tensor τ^v and the Reynolds-stress tensor τ^R ,

$$\tau_{ij} = \tau_{ij}^v + \tau_{ij}^R. \quad (2.11)$$

For a Newtonian fluid, employing the Stokes' hypothesis, the viscous term is given by

$$\begin{aligned} \tau_{ij}^v &= 2\mu \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \\ S_{ij} &= \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right), \end{aligned} \quad (2.12)$$

with S the rate-of-strain tensor, and with μ the molecular dynamic viscosity. The Reynolds-stress tensor is an unknown term in the RANS equations, representing the effect on the mean flow of turbulent fluctuations in the velocity field. In an eddy-viscosity turbulence model, the Reynolds stresses are modelled using the Boussinesq hypothesis [84],

$$\tau_{ij}^R = 2\mu_t \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij}, \quad (2.13)$$

with μ_t the so-called eddy-viscosity coefficient and with k the turbulent kinetic energy.

The total heat-flux vector consists of the sum of contributions due to heat conduction and due to turbulence, which are modelled using, respectively, the law of Fourier and a gradient hypothesis,

$$Q_i = -(\kappa_h + \kappa_t) \frac{\partial T}{\partial x_i}, \quad (2.14)$$

with κ_h the thermal conductivity coefficient, and κ_t the turbulent conductivity coefficient. Similar to the Reynolds stress tensor, the turbulent heat-flux vector represents the effect on the mean flow of turbulent fluctuations in the temperature and velocity fields.

The dynamic viscosity coefficient μ is assumed to be given by the Sutherland law [64],

$$\frac{\mu}{\mu_\infty} = \left(\frac{T}{T_\infty} \right)^{3/2} \frac{T_\infty + T_S}{T + T_S}, \quad (2.15)$$

with μ_∞ its free-stream value, and T_S a constant. The thermal conductivity coefficient κ_h is assumed to be proportional to the dynamic viscosity,

$$\frac{\kappa_h}{\mu} = \frac{\kappa_{h,\infty}}{\mu_\infty}. \quad (2.16)$$

The eddy-viscosity coefficient μ_t is defined by one of the turbulence models given in chapter 5. In the algebraic models employed there, the turbulent kinetic energy is usually neglected in the Boussinesq relation (or assumed to be contained in the pressure). Similar to the thermal conductivity coefficient, the turbulent conductivity coefficient κ_t is assumed to be proportional to the eddy-viscosity coefficient,

$$\frac{\kappa_t}{\mu_t} = \text{constant}. \quad (2.17)$$

Integrating equation (2.6) over an arbitrary fixed volume segment $\Omega \subset D$ and applying Gauss' divergence theorem, the integral form of the RANS equations may be obtained,

$$\frac{d}{dt} \iiint_{\Omega} U dV + \iint_{\partial\Omega} \mathcal{F}^c \cdot \vec{m} dA = \iint_{\partial\Omega} \mathcal{F}^d \cdot \vec{m} dA, \quad (2.18)$$

where $\partial\Omega$ is the boundary of Ω , and where \vec{m} is the unit vector normal to $\partial\Omega$ and pointing outward of Ω . This integral form is the starting point for the discretization of the equations by a finite-volume method (see chapter 3). Note that a steady, uniform flow satisfies the integral equations, because

$$\iint_{\partial\Omega} \vec{m} dA \equiv \vec{0}. \quad (2.19)$$

For the numerical discretization of the equations and for the derivation of the TLNS equations, it is useful to write the RANS equations in conservation form in the curvilinear coordinate system (ξ, η, ζ) ,

$$\begin{aligned} J \frac{\partial U}{\partial t} + \frac{\partial}{\partial \xi} (\mathcal{F}^c \cdot J \nabla \xi) + \frac{\partial}{\partial \eta} (\mathcal{F}^c \cdot J \nabla \eta) + \frac{\partial}{\partial \zeta} (\mathcal{F}^c \cdot J \nabla \zeta) \\ = \frac{\partial}{\partial \xi} (\mathcal{F}^d \cdot J \nabla \xi) + \frac{\partial}{\partial \eta} (\mathcal{F}^d \cdot J \nabla \eta) + \frac{\partial}{\partial \zeta} (\mathcal{F}^d \cdot J \nabla \zeta), \end{aligned} \quad (2.20)$$

with J the determinant of the Jacobian,

$$J = \det \left(\frac{d(x, y, z)}{d(\xi, \eta, \zeta)} \right) = \frac{\partial \vec{x}}{\partial \xi} \cdot \left(\frac{\partial \vec{x}}{\partial \eta} \times \frac{\partial \vec{x}}{\partial \zeta} \right), \quad (2.21)$$

and with

$$J \nabla \xi = \frac{\partial \vec{x}}{\partial \eta} \times \frac{\partial \vec{x}}{\partial \zeta}, \quad J \nabla \eta = \frac{\partial \vec{x}}{\partial \zeta} \times \frac{\partial \vec{x}}{\partial \xi}, \quad J \nabla \zeta = \frac{\partial \vec{x}}{\partial \xi} \times \frac{\partial \vec{x}}{\partial \eta}. \quad (2.22)$$

For the discretization, J is usually identified with the volume of a grid cell, and $J \nabla \xi$ is identified with an area vector belonging to a face of a grid cell (see chapter 3). For a discussion of the conservation form of the equations in curvilinear coordinates see e.g. Vinokur [115]. Also, see the derivation in appendix A.

2.3 THE THIN-LAYER REYNOLDS-AVERAGED NAVIER–STOKES EQUATIONS

The thin-layer Reynolds-averaged Navier–Stokes equations are used to describe compressible, turbulent flows in which the viscous and turbulent effects are limited to thin layers, such as thin boundary layers and wakes. The TLNS equations are a simplification of the full RANS equations, where in the diffusive terms only derivatives in the so-called thin-layer normal direction are retained [10]. This normal direction is orthogonal to a solid surface or a wake-centre surface. One of the curvilinear coordinates ξ , η , or ζ is assumed to coincide with the thin-layer normal direction. In the case

of two solid surfaces (or wakes) making an intersection, also more than one thin-layer normal direction is allowed. This allows the simulation of e.g. the boundary layers on both the wing and the fuselage of an aircraft with the TLNS equations. However, the interaction region of the two boundary layers is not accurately modelled, since cross derivatives which may be significant in that region have been neglected in the TLNS approximation.

Using the curvilinear coordinates (ξ, η, ζ) , the TLNS equations may be written as

$$\begin{aligned} J \frac{\partial U}{\partial t} + \frac{\partial}{\partial \xi}(\mathcal{F}^c \cdot J \nabla \xi) + \frac{\partial}{\partial \eta}(\mathcal{F}^c \cdot J \nabla \eta) + \frac{\partial}{\partial \zeta}(\mathcal{F}^c \cdot J \nabla \zeta) \\ = \frac{\partial F_\xi^d}{\partial \xi} + \frac{\partial F_\eta^d}{\partial \eta} + \frac{\partial F_\zeta^d}{\partial \zeta}. \end{aligned} \quad (2.23)$$

This equation is equivalent to the RANS equations in the form of equation (2.20), with the diffusive flux vector $F_\xi^d \in \mathbb{R}^5$ given by

$$F_\xi^d = \mathcal{F}^d \cdot J \nabla \xi = \begin{pmatrix} 0 \\ \vec{\tau}^\xi \\ \vec{u} \cdot \vec{\tau}^\xi - Q^\xi \end{pmatrix} J \|\nabla \xi\|, \quad (2.24)$$

with $\vec{\tau}^\xi$ a shear stress vector and Q^ξ a heat flux. However, in the TLNS equations, the diffusive flux vector F_ξ^d is only taken into account when the ξ direction corresponds to a thin-layer normal direction. Otherwise it is neglected.

Expressions for the shear stress vector $\vec{\tau}^\xi$ and the heat flux Q^ξ are obtained using equations (2.11) to (2.14), but retaining only derivatives in the ξ direction,

$$\begin{aligned} \vec{\tau}^\xi &= \boldsymbol{\tau} \cdot \vec{n}^\xi = (\mu + \mu_t) \left(\frac{\partial \vec{u}}{\partial n^\xi} + \frac{1}{3} \left(\vec{n}^\xi \cdot \frac{\partial \vec{u}}{\partial n^\xi} \right) \vec{n}^\xi \right), \\ Q^\xi &= \vec{Q} \cdot \vec{n}^\xi = -(\kappa_h + \kappa_t) \frac{\partial T}{\partial n^\xi}, \end{aligned} \quad (2.25)$$

with $\vec{n}^\xi = \nabla \xi / \|\nabla \xi\|$ the unit vector in the direction normal to the surfaces $\xi = \text{constant}$ and with n^ξ the distance in the same direction, $dn^\xi = d\xi / \|\nabla \xi\|$.

As for the RANS equations, one also wants to write the TLNS equations in integral form. Define the co-variant vectors \vec{s}_ξ , \vec{s}_η , and \vec{s}_ζ in the ξ , η , and ζ directions, respectively, as

$$\vec{s}_\xi = \frac{1}{J} \frac{\partial \vec{x}}{\partial \xi}, \quad \vec{s}_\eta = \frac{1}{J} \frac{\partial \vec{x}}{\partial \eta}, \quad \vec{s}_\zeta = \frac{1}{J} \frac{\partial \vec{x}}{\partial \zeta}. \quad (2.26)$$

Recombining the flux vectors F_ξ^d , F_η^d , and F_ζ^d into a flux matrix \mathcal{F}^d ,

$$\mathcal{F}^d = F_\xi^d \vec{s}_\xi^T + F_\eta^d \vec{s}_\eta^T + F_\zeta^d \vec{s}_\zeta^T, \quad (2.27)$$

and considering that the vector \vec{s}_ξ is orthogonal to the contra-variant vectors $\nabla \eta$ and $\nabla \zeta$, equation (2.23) can be rewritten in the form of equation (2.20). Thus, the

integral form of equation (2.18) also applies, which may be written as

$$\begin{aligned} \frac{d}{dt} \iiint_{\Omega} U dV + \iint_{\partial\Omega} \mathcal{F}^c \cdot \vec{m} dA = \\ \iint_{\partial\Omega} F_{\xi}^d(\vec{s}_{\xi} \cdot \vec{m}) dA + \iint_{\partial\Omega} F_{\eta}^d(\vec{s}_{\eta} \cdot \vec{m}) dA + \iint_{\partial\Omega} F_{\zeta}^d(\vec{s}_{\zeta} \cdot \vec{m}) dA. \end{aligned} \quad (2.28)$$

2.4 THE EULER EQUATIONS

The Euler equations for a compressible, inviscid flow are obtained by neglecting the terms due to viscosity, heat conduction, and turbulence from the RANS equations (in the form of equations (2.5) and (2.6)), resulting in

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} &= 0, \\ \frac{\partial \rho u_i}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} &= 0, \\ \frac{\partial \rho E}{\partial t} + \frac{\partial(\rho E u_j)}{\partial x_j} + \frac{\partial(p u_j)}{\partial x_j} &= 0, \end{aligned} \quad (2.29)$$

or equivalently,

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathcal{F}^c = 0. \quad (2.30)$$

Note that for a steady inviscid flow, the equations for the conservation of mass and energy (equations (2.29a) and (2.29c)) may be combined into

$$u_j \frac{\partial H}{\partial x_j} = 0, \quad (2.31)$$

with $H = E + p/\rho$ the total enthalpy per unit mass. Thus, for a steady inviscid flow the enthalpy will be constant along streamlines or even throughout the flow domain if the same value is specified at all inflow boundaries (and no recirculation occurs). Similarly, it can be shown that the entropy $S = c_v \ln(p/\rho^\gamma)$ is constant along path lines (the paths followed by material elements) in unsteady inviscid flows, and thus constant along streamlines in steady inviscid flows.

The Euler equations may also be written in integral form,

$$\frac{d}{dt} \iiint_{\Omega} U dV + \iint_{\partial\Omega} \mathcal{F}^c \cdot \vec{m} dA = 0, \quad (2.32)$$

or using the curvilinear coordinate system (ξ, η, ζ) ,

$$J \frac{\partial U}{\partial t} + \frac{\partial}{\partial \xi} (\mathcal{F}^c \cdot J \nabla \xi) + \frac{\partial}{\partial \eta} (\mathcal{F}^c \cdot J \nabla \eta) + \frac{\partial}{\partial \zeta} (\mathcal{F}^c \cdot J \nabla \zeta) = 0. \quad (2.33)$$

The integral form of the Euler equations allows solutions in which the flow state is discontinuous across a surface (weak solutions). In steady flows, requiring the flux vector $\mathcal{F}^c \cdot \vec{m}$ to be continuous at the surface of discontinuity (with \vec{m} normal to the surface), two types of solutions are obtained (see e.g. [65, 103]):

- shock waves, for which the normal velocity component ($\vec{u} \cdot \vec{m}$) and the pressure are discontinuous, while the tangential velocity components and the total enthalpy are continuous, and
- contact discontinuities, for which the normal velocity component equals zero, and the pressure is continuous, while the tangential velocity components and the total enthalpy are possibly discontinuous.

For these weak solutions, the total enthalpy is still constant along streamlines, and thus isenthalpic flow will still occur if the same value is specified at all inflow boundaries. Across a shock wave, the entropy is discontinuous, and must increase in the flow direction for the solution to be physically valid (the so-called entropy condition). As a consequence, the normal Mach number (ratio of normal velocity component and speed of sound) must decrease in the flow direction, being larger than one upstream of the shock (supersonic flow) and smaller than one downstream of the shock (subsonic flow for shock waves normal to the flow), while the pressure increases. For transonic flows around aircraft, a supersonic-flow region is commonly present on the upper side of the wing, starting near the leading edge, and ending in a shock wave further downstream on the wing. Slip surfaces, which are contact discontinuities for which the tangential velocity is discontinuous, are typically present downstream of the wing trailing edge.

For the RANS equations, discontinuous solutions are not allowed. Slip surfaces would be immediately smeared out by viscous (and turbulent) diffusion, resulting in wakes of finite thickness. Shock waves, however, are distributed over regions of such small dimensions (of the order of magnitude of the mean free path of the gas molecules for strong shocks [65]) that they may still be seen as discontinuities. For the TLNS equations, viscous diffusion is only taken into account in the thin-layer normal direction, so that slip surfaces are allowed if the surfaces are tangential to this direction.

2.5 BOUNDARY CONDITIONS

The flow model must be completed by appropriate boundary conditions, which are described in this section. Conditions are considered for both the TLNS and the Euler equations. For the flow around a (transport-type) aircraft, the following boundaries may occur:

- solid surface,
 - far-field boundary,
 - symmetry plane, and
 - propulsion-system boundaries (inlets and outlets of jets, propeller disks),
- while, in order to perform 2D simulations with a 3D method, also the
- plane of infinite continuation

is considered. Boundary conditions for the propulsion systems were already developed for the Euler solver. A boundary condition for a jet-engine inlet suitable for the Navier–Stokes equations was recently developed at NLR. Main idea of such boundary conditions is to specify general properties (such as mass flux, total temperature, etc.), so that the main effect of the propulsion system on the flow is taken into account. These propulsion-system boundary conditions are not considered further here.

In the following subsections, the vector \vec{m} will be the unit vector normal to the boundary of the flow domain and pointing outward of the flow domain.

2.5.1 Solid surface

At solid surfaces, viscous effects cause the flow to ‘stick’ to the surface, thus implying that the velocity must be equal to zero (no-slip condition). Furthermore, adiabatic flow is assumed, i.e. the heat flux through solid surfaces is required to be equal to zero. Thus, for the TLNS equations, the following boundary conditions are applied at solid surfaces:

$$\begin{aligned}\vec{u} &= \vec{0}, \\ \vec{m} \cdot \nabla T &= 0.\end{aligned}\tag{2.34}$$

In the case of the Euler equations, due to the absence of viscous effects, the flow is allowed to slip along the surface, so that only the normal velocity component must be equal to zero (slip condition),

$$\vec{u} \cdot \vec{m} = 0.\tag{2.35}$$

When for the TLNS equations, only the wing boundary layers are modelled, and not the boundary layer on e.g. the fuselage, then the no-slip condition is applied at the wing surface, while the slip condition is applied at the fuselage surface.

2.5.2 Far-field boundary

It is assumed that at infinity, a uniform flow is present, given by

$$\begin{aligned}\rho &= \rho_\infty, \\ \vec{u} &= \vec{u}_\infty, \\ p &= p_\infty.\end{aligned}\tag{2.36}$$

In the discrete model, the far-field boundary will be located at a finite distance from the configuration. Assuming the flow in the far field to be inviscid, the same boundary condition may be applied for the TLNS equations as for the Euler equations.

At the far field, a boundary condition can be derived by applying local characteristic theory. The main idea is to require characteristic variables of incoming characteristics to be equal to their free-stream values, while, in the discrete model, the other characteristic variables are extrapolated to the far-field boundary from within the flow domain [51]. This boundary condition belongs to the class of non-reflective boundary conditions. This means that waves are not reflected at the far-field boundary back into the domain, which could otherwise slow down the convergence of the flow computation considerably.

Consider the Euler equations in curvilinear coordinates (equation (2.33)). Let the far-field boundary be given by $\xi = \text{constant}$, with the ξ direction pointing outwards of the flow domain. Assume that at the far field the curvilinear coordinate system reduces to a rectilinear system (thus J and $\nabla \xi$ are locally constant, and we may also assume that $\|\nabla \xi\| = 1$). Neglecting the derivatives of the flow state to η and ζ , the

Euler equations may be written in the one-dimensional characteristic form derived in appendix A,

$$\frac{\partial V}{\partial t} + \Lambda \frac{\partial V}{\partial \xi} = 0, \quad (2.37)$$

with $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_5)$ a real diagonal matrix and $V \in \mathbb{R}^5$ the state vector of characteristic variables or Riemann invariants (see e.g. Courant and Hilbert [27]). The characteristic variables are propagated along the corresponding characteristics $d\xi/dt = \lambda_i$, which are incoming if $\lambda_i < 0$ and outgoing if $\lambda_i > 0$. The diagonal elements λ_i are the eigenvalues of the Jacobian of the convective flux in ξ direction, $A^\xi = d(\mathcal{F}^c \cdot \vec{m})/dU$ (also derived in appendix A), and are given by

$$\begin{aligned} \lambda_i &= \vec{u} \cdot \vec{m}, & i \in \{1, 2, 3\}, \\ \lambda_4 &= \vec{u} \cdot \vec{m} + c, \\ \lambda_5 &= \vec{u} \cdot \vec{m} - c, \end{aligned} \quad (2.38)$$

in which $\vec{m} = \nabla \xi$ is the outward unit vector normal to the far-field boundary, and in which c is the local speed of sound,

$$c = \sqrt{\frac{\gamma p}{\rho}}. \quad (2.39)$$

The characteristic variables are defined by the differential relation $dV = \mathcal{L}dU$, with \mathcal{L} the matrix with the left eigenvectors of A^ξ as its row vectors. Using the primitive flow variables ρ , \vec{u} , and p , this relation can be written as (with the row vectors of \mathcal{L} suitably scaled)

$$\begin{aligned} dV_1 &= -c_p d\rho/\rho + c_v dp/p, \\ dV_2 &= \vec{s} \cdot d\vec{u}, \\ dV_3 &= \vec{t} \cdot d\vec{u}, \\ dV_4 &= \vec{m} \cdot d\vec{u} + (c/\gamma) dp/p, \\ dV_5 &= \vec{m} \cdot d\vec{u} - (c/\gamma) dp/p, \end{aligned} \quad (2.40)$$

with $(\vec{m}, \vec{s}, \vec{t})$ orthonormal. Consider the entropy $S = c_v \ln(p/\rho^\gamma)$, so that $dV_1 = dS$. In order to obtain simple algebraic expressions for V_4 and V_5 , it is assumed that the flow at the far field is isentropic ($dS = 0$, which implies $\gamma d\rho/\rho = dp/p$), resulting in the following expressions for the characteristic variables (apart from additive constants):

$$\begin{aligned} V_1 &= S, \\ V_2 &= \vec{u} \cdot \vec{s}, \\ V_3 &= \vec{u} \cdot \vec{t}, \\ V_4 &= R_{\text{out}} = \vec{u} \cdot \vec{m} + 2c/(\gamma - 1), \\ V_5 &= R_{\text{in}} = \vec{u} \cdot \vec{m} - 2c/(\gamma - 1), \end{aligned} \quad (2.41)$$

where the differential relation $2dc/c = dp/p - d\rho/\rho = (\gamma - 1)dp/(\gamma p)$ has been used (which follows from equation (2.39)).

Finally, the boundary conditions are defined as follows. We only consider subsonic flow at the far field, meaning that the Mach number ($M_\infty = \|\vec{u}_\infty\|/c_\infty$) is smaller

than one. In case of inflow ($\vec{u}_\infty \cdot \vec{m} < 0$), there are four incoming characteristics (1, 2, 3, and 5) and the corresponding boundary conditions are

$$\begin{aligned} S &= c_v \ln(p_\infty / \rho_\infty^\gamma), \\ \vec{u} \cdot \vec{s} &= \vec{u}_\infty \cdot \vec{s}, \\ \vec{u} \cdot \vec{t} &= \vec{u}_\infty \cdot \vec{t}, \\ R_{\text{in}} &= \vec{u}_\infty \cdot \vec{m} - (2c_\infty)/(\gamma - 1). \end{aligned} \quad (2.42)$$

In case of outflow ($\vec{u}_\infty \cdot \vec{m} > 0$), there is only one incoming characteristic (5) with corresponding boundary condition,

$$R_{\text{in}} = \vec{u}_\infty \cdot \vec{m} - (2c_\infty)/(\gamma - 1). \quad (2.43)$$

2.5.3 Symmetry plane

In the case of a symmetric problem, it is sufficient to consider only half of the flow domain. In that case a symmetry boundary condition must be prescribed at the symmetry plane. This condition consists of requiring the normal velocity component as well as the normal gradients of density, pressure, and the tangential velocity components to be equal to zero,

$$\begin{aligned} \vec{m} \cdot \vec{u} &= 0, \\ \vec{m} \cdot \nabla \rho &= 0, \\ \vec{m} \cdot \nabla (\vec{u} \cdot \vec{s}) &= 0, \\ \vec{m} \cdot \nabla (\vec{u} \cdot \vec{t}) &= 0, \\ \vec{m} \cdot \nabla p &= 0, \end{aligned} \quad (2.44)$$

with \vec{s} and \vec{t} unit vectors tangential to the symmetry plane.

2.5.4 Plane of infinite continuation

In order to do a 2D computation (with a 3D method) two parallel planes can be defined, so-called planes of infinite continuation (usually given by $y = \text{constant}$), between which the 2D configuration is extended. At the two planes, the normal gradient of the basic variables are required to be equal to zero,

$$\begin{aligned} \vec{m} \cdot \nabla \rho &= 0, \\ \vec{m} \cdot \nabla (\rho \vec{u}) &= \vec{0}, \\ \vec{m} \cdot \nabla (\rho E) &= 0. \end{aligned} \quad (2.45)$$

This boundary condition may also be employed for simulating an infinite sheared wing, by giving the free-stream velocity also a component in the direction normal to the plane of infinite continuation. (The 2D airfoil must then be defined as a cross section of the wing normal to the leading edge.)

2.6 FLOW PARAMETERS AND DIMENSIONLESS GROUPS

Now that the general equations and boundary conditions describing the compressible flow around an aircraft configuration have been defined, we must identify suitable

parameters for defining a particular flow. The solution of the TLNS equations depends on the following:

- the aircraft configuration in terms of its geometric form and an associated characteristic length scale L ,
- the free-stream density ρ_∞ , velocity \vec{u}_∞ , and temperature T_∞ ,
- the gas constant R and the specific heat at constant pressure c_p ,
- the free-stream dynamic viscosity μ_∞ and thermal conductivity $\kappa_{h,\infty}$,
- the constant of the Sutherland law T_S , and
- the constant ratio of turbulent conductivity and eddy viscosity κ_t/μ_t .

Thus, the solution of the TLNS equations depends on twelve parameters, considering the geometric form of the aircraft configuration to be given. Since the equations contain four basic dimensions (time, length, mass, and temperature), eight dimensionless groups can be derived by applying dimensional analysis (see e.g. [98]):

- the angles of attack α and side slip β , specifying the direction of the free-stream velocity with respect to the Cartesian coordinate system as defined in figure 2.1,
- the free-stream Mach number $M_\infty = u_\infty/c_\infty$,
- the ratio of specific heats $\gamma = c_p/c_v$,
- the free-stream Reynolds number $Re_\infty = \rho_\infty u_\infty L/\mu_\infty$,
- the Prandtl number $Pr = c_p \mu_\infty/\kappa_{h,\infty}$,
- the turbulent Prandtl number $Pr_t = c_p \mu_t/\kappa_t$, and
- the ratio of the constant of the Sutherland law and the free-stream temperature $\theta_S = T_S/T_\infty$,

with $u_\infty = \|\vec{u}_\infty\|$, and with c_∞ the free-stream speed of sound. (For the Euler equations the latter four parameters are not needed.) Note that it is customary in aerodynamics to define the xz plane parallel to the symmetry plane or root plane of the aerodynamic configuration, with the x axis parallel to the reference axis of the body (e.g. the chord-wise direction) and pointing downstream, and with the z axis pointing upwards. Figure 2.1 also gives the direction of the three components (lift, drag, and side force) of the aerodynamic force working on the aircraft and the direction of the three components (pitch, roll, and yaw) of the moment exerted by the flow on the aircraft (with respect to a certain reference point, e.g. its centre of gravity).

The eight dimensionless groups, together with the four variables of equation (2.4) (i.e. ρ_∞ , p_∞ , T_∞ , and L), now completely define the twelve flow parameters. Thus, to define a specific flow case, it is sufficient to give the dimensionless groups. Four of these groups are given specific values, which are considered to be suitable for the applications of interest [14]:

$$\gamma = 1.4, \quad Pr = 0.72, \quad Pr_t = 0.9, \quad \theta_S = 0.4. \quad (2.46)$$

For completeness, the dependence of the flow parameters on the dimensionless groups and on ρ_∞ , p_∞ , T_∞ , and L is specified. The free-stream velocity is defined as in figure 2.1,

$$\vec{u}_\infty = \begin{pmatrix} \cos \alpha \cos \psi \\ \sin \beta \\ \sin \alpha \end{pmatrix} u_\infty, \quad (2.47)$$

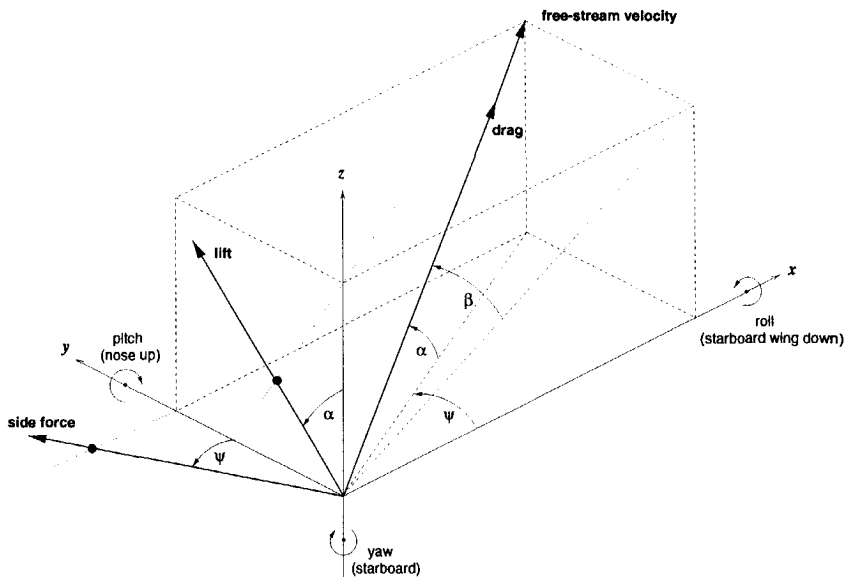


Figure 2.1: Definition of free-stream velocity and of forces and moments

with $\psi = \arcsin(\sin \beta / \cos \alpha)$ the orthogonal projection of β on the xy plane, and with the magnitude of velocity (and the speed of sound c_∞) given by

$$u_\infty = c_\infty M_\infty, \quad c_\infty = \sqrt{\frac{\gamma p_\infty}{\rho_\infty}}. \quad (2.48)$$

The other parameter are given by

$$\begin{aligned} R &= \frac{p_\infty}{\rho_\infty T_\infty}, & c_p &= \frac{\gamma}{\gamma - 1} R, & \mu_\infty &= \frac{\rho_\infty u_\infty L}{Re_\infty}, \\ \kappa_{h,\infty} &= \frac{c_p \mu_\infty}{Pr}, & T_S &= \theta_S T_\infty, & \frac{\kappa_t}{\mu_t} &= \frac{c_p}{Pr_t}. \end{aligned} \quad (2.49)$$

2.7 SUMMARY

In this chapter, we have described the continuous equations by which the compressible, turbulent flow around transport-type aircraft will be modelled. In particular, the turbulent boundary layers and wakes of the aircraft wing are described by the thin-layer Reynolds-averaged Navier-Stokes equations. Other regions of the flow will be modelled as inviscid and are described by the Euler equations. The integral form of these continuous equations, given by equations (2.28) and (2.32), respectively, will be the starting point for the numerical method presented in the subsequent two chapters. The continuous equations are complemented by the boundary conditions of section 2.5. For a given aircraft configuration, the solution of the equations is determined by eight dimensionless groups of which four may be chosen freely (the free-stream Mach

and Reynolds numbers, and the angles of attack and side slip), while the remaining four have been given specific values (equation (2.46)).

3

SPACE DISCRETIZATION

3.1 INTRODUCTION

In the previous chapter, we have described the flow model in terms of a set of continuous equations (in particular the TLNS equations) complemented by a collection of boundary conditions. A numerical method for approximating such a set of equations generally consists of first deriving a set of discrete equations, and second developing a procedure for solving these discrete equations. Since, in our case, the flow model consists of steady equations, only discretization with respect to spatial directions is needed. However, one may also consider the unsteady equations, discretize them both in space and time, and integrate in time until a steady solution is obtained. In that case, one should ensure that the final solution is independent of the method used to perform the time integration, and is in fact the solution of the discretized steady equations. In this chapter, the space discretization of the unsteady equations will be discussed. This discretization will be equivalent to a direct discretization of the steady equations. In the next chapter, the solution procedure will be discussed, which will be based on the idea of integration in time.

The starting point for the space discretization of the TLNS equations is the discretization of the Euler equations as used for the existing Euler flow solver. As outlined in chapter 1, the main concepts of this discretization are:

- multi-block structured grids,
- a cell-centred finite-volume scheme, and
- central differencing with explicit artificial diffusion.

First, a literature review will be presented, discussing these concepts and the possibilities for extending this discretization to the TLNS equations. After briefly describing multi-block structured grids in section 3.3, a direct extension of the methods for the Euler equations to the TLNS equations is described in section 3.4, resulting in the so-called basic scheme, followed by the discretization of the boundary conditions in section 3.5. However, it follows from the literature review that this basic scheme will not satisfy the requirements formulated in chapter 1 (in particular the requirement on numerical accuracy). Therefore, the method is further extended in section 3.6 with a matrix artificial diffusion scheme, which reduces the grid dependency of the numerical solution in boundary layers. Also, in that section, a further extension of the space discretization is described (scaling of the artificial diffusion for high-aspect-ratio grid cells) which is needed in order to obtain an efficient solution procedure, as will be discussed in chapter 4.

3.2 LITERATURE REVIEW

The three main concepts of the space discretization are major design concepts that influence the structure of the complete flow solver. Switching to other concepts, e.g. changing from multi-block structured grids to unstructured grids, would be a considerable effort and pose a serious risk, and in fact would almost amount to developing a completely new flow solver. Thus, when extending an existing flow solver, one should evaluate if the objectives can be met maintaining these main concepts.

Multi-block structured grids are probably the most popular type of grids for aerodynamic flow simulations [12, 100], and many solutions of the Euler and Navier–Stokes equations for complete aircraft using this type of grids may be found in the literature [18, 35, 45, 62, 63, 91, 96]. The main drawback of this approach is the man-hour consuming grid-generation task. An advantage is that accurate and efficient flow-solution algorithms may be more easily developed than for alternative approaches. The main alternatives for using multi-block structured grids are:

- Chimera approach (e.g. [9, 38]): also a multi-block approach, but one in which the blocks are allowed to overlap. The Chimera approach simplifies the grid generation task, but at the expense of an increased complexity of the flow solver, in particular with respect to the coupling of the blocks.
- Unstructured grids (e.g. [25, 75]): grids in which no structure is assumed, and that usually consist of tetrahedral cells, although also prismatic or hexahedral cells are sometimes used. If not automatic, grid generation is certainly less time consuming. There are however difficulties for application to the Navier–Stokes equations. Apparently there seems to be some need for local structure of the grid (e.g. through using prismatic cells) to properly describe boundary layers.
- Cartesian grids (e.g. [89]): grids which are defined as the intersection of constant Cartesian coordinate planes (as opposed to the multi-block structured grids being curvilinear body-fitted grids) usually with local refinement of grid cells. For Cartesian grids, the difficulty lies in the treatment of the intersection with the geometry surface and in the treatment of the boundary conditions at this surface.

All three approaches, but some more than others, lighten the grid generation task compared to the multi-block structured-grid approach. Unstructured grids are particularly gaining popularity. Their main strength lies in faster grid generation for complex geometries. Also, adaptation of grids to the flow solution (to improve the accuracy) may simply be done through adding and deleting grid points. However, grid adaptation, though perhaps somewhat limited in its potential compared to unstructured grids, is certainly also possible for structured grids through grid-point movement [41, 42, 43].

Boundary layers are easily captured using body-fitted curvilinear grids (such as multi-block structured grids) with grid cells of high aspect ratio. A high grid resolution in boundary-layer normal direction can be obtained without needlessly increasing the resolution in tangential directions. The alternatives do not offer any advantages

in this respect, if not making the capturing of boundary layers more difficult. Numerical methods will, in general, not be more efficient or accurate. Thus, although the alternatives to multi-block structured grids do offer some advantages, these advantages are not relevant for our purposes (extension from Euler to Navier–Stokes), and therefore we do not change the grid approach. The alternatives could be considered, for example, when one wants to reduce the total turn-around time of the complete flow simulation (including grid generation).

For compressible flows, the main alternative to using a cell-centre scheme (storage of flow variables at cell centres) is the cell-vertex approach (storage at cell vertices) [50, 73, 109]. For incompressible flows, also frequently the staggered grid approach is used (storage of pressure at cell centres and of velocity components at the appropriate cell faces), e.g. [126], the advantages of which are related to the particular form of the incompressible equations. Cell-vertex schemes are very similar to cell-centre schemes. They are in particular used for unstructured grids, since for such grids a cell-vertex scheme is in general argued to be more accurate than a cell-centre scheme, see e.g. [50]. For structured grids, in practice similar accuracies are obtained for both schemes (e.g. for 2D transonic airfoils see [73, 109]). It has been shown theoretically that for central differencing of first- and second-order derivatives, cell-vertex schemes require less smoothness of the grid than cell-centre schemes to obtain the same order of accuracy [80, 109], but this effect may be lost when artificial diffusion (or upwinding) is considered. We do not consider the choice between cell-centre and cell-vertex schemes to be decisive for the question whether a sufficiently accurate Navier–Stokes solver can be obtained, and thus stick to the cell-centre approach. Moreover, for multi-block grids, a disadvantage of cell-vertex schemes is that they require the coupling of blocks at block faces, edges, and vertices, increasing the complexity of the flow solver compared to cell-centre schemes, which only require the coupling of blocks at block faces (as long as the TLNS approximation is used).

The differencing scheme of the existing Euler solver (central differencing with artificial diffusion) is basically the method developed by Jameson et al. [48, 51, 52]. In combination with a Runge–Kutta time-integration scheme (possibly accelerated by residual averaging, enthalpy damping, and multi grid), it is a highly popular scheme mainly due to its simplicity, robustness, and cheapness.

An important property of any differencing scheme for the compressible flow equations is the correct treatment of flow discontinuities, in particular shock waves (occurring in transonic and supersonic flows) and contact discontinuities (only for inviscid flow). For the Jameson scheme, with the artificial diffusion consisting of a blending of 2nd and 4th-order differences, oscillation-free shocks distributed over three to four grid cells can be obtained (when the shock is aligned with the grid). However, no special facility is included for treating contact discontinuities, which may be strongly smeared.

The main alternative for the Jameson-type scheme consists of the large collection of upwind schemes (see e.g. [95, 118]). Two important classes of upwind schemes for hyperbolic systems of conservation laws are the approximate Riemann solvers (ARS), e.g. the schemes of Roe [94], Osher [86], and Harten [44], and the flux-vector splitting (FVS) schemes, e.g. Steger–Warming [108] and Van Leer [68]. The ARS schemes are generally capable of sharply capturing flow discontinuities (both shocks and contact

discontinuities) usually with only one or two interior points (when the discontinuity is aligned with the grid). The FVS schemes are also generally capable of sharply capturing shocks, but tend to smear out contact discontinuities due to significant cross-wind diffusion. In their original form, these upwind schemes are only first-order accurate. Several approaches exist to obtain higher-order accurate, oscillation-free upwind schemes from the first-order schemes, such as the MUSCL approach of Van Leer [69] and the addition of limited anti-diffusive fluxes [95].

Basing on an analogy with upwind schemes, Swanson and Turkel [111] have improved the Jameson scheme following the notion of matrix artificial diffusion. Their scheme resembles upwind schemes, and is claimed to be only slightly more diffusive at shocks (compared to the Roe scheme). Since all characteristic wave speeds are taken into account, it seems the matrix scheme should in principle be able to sharply capture contact discontinuities, although Swanson and Turkel do not show this.

The important question is now how these different types of differencing schemes will behave when applied to the Navier-Stokes equations, and in particular for high-Reynolds-number boundary layers. Van Leer et al. [71] have shown that schemes that are not capable of sharply capturing contact discontinuities for the Euler equations (i.e. schemes containing serious cross-wind diffusion) tend to diffuse boundary layers unless very fine grids are used. The Jameson-type scheme and the FVS schemes fall into this class. With the flux-difference splitting scheme of Roe, for example, boundary-layer solutions with considerably less grid dependency can be obtained. Swanson and Turkel show [112] that their matrix scheme (provided some parameters are given appropriate values) has a grid-convergence behaviour for boundary layers far better than the Jameson scheme, and in fact quite similar to the Roe scheme.

In conclusion, it may be expected that the requirements formulated in chapter 1 can be met, maintaining the main concepts of the space discretization of the existing Euler solver. The Jameson-type scheme, however, must be improved along the lines of the matrix artificial-diffusion scheme of Swanson-Turkel to reduce the grid dependency of the numerical solution in boundary layers.

3.3 MULTI-BLOCK APPROACH AND DOMAIN DISCRETIZATION

In order to define a finite set of discrete equations, first the flow domain must be discretized. A bounded flow domain is considered by setting the far-field boundary at a finite distance from the configuration. The discretization of this bounded domain consists of dividing the domain in a finite number of volume segments called cells, which together form a grid. In a structured 3D grid, these cells are hexahedral shaped, are packed face-to-face, and can be numbered in an ordered way using a triple index (i, j, k) (with neighbouring cells having subsequent index values). If the flow domain can be described by a curvilinear coordinate system $(\xi, \eta, \zeta) \in [0, 1]^3$, then a structured grid can be seen as the image of a uniform mesh in the unit cube by the mapping of (ξ, η, ζ) to (x, y, z) (see figure 3.1).

Around a complex aeronautical configuration, a boundary-conforming structured grid can, in general, only be generated using the multi-block approach [106]. In such an approach, the flow domain is divided in a set of non-overlapping blocks. Each block has the topology of a cube, thus having six block faces, and can be described

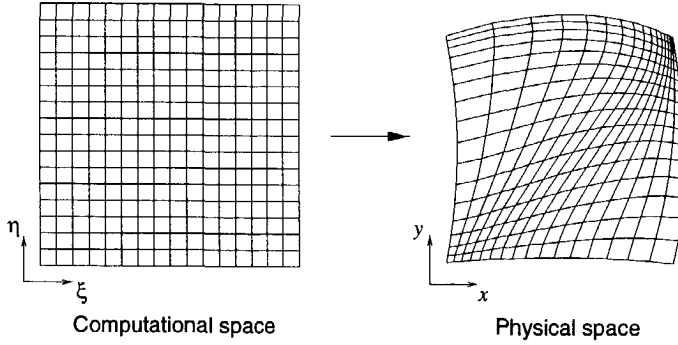
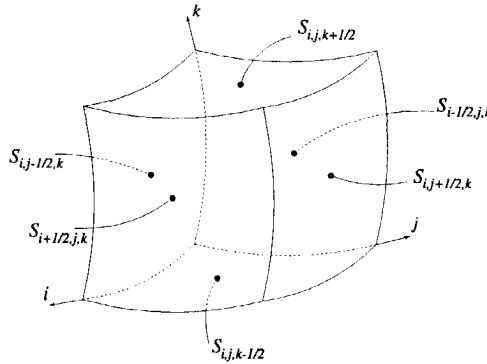


Figure 3.1: Mapping from computational to physical domain in 2D

by a curvilinear coordinate system $(\xi, \eta, \zeta) \in [0, 1]^3$, so that a structured grid can be defined in each block separately. In order to reduce the complexity of multi-block domains and the number of blocks, blocks are not required to be packed block-face-to-block-face. Instead, block faces are divided into subfaces (called elementary faces), and two distinct blocks may border only at one (or a few) of those elementary faces. The collection of blocks, block faces, elementary faces (as well as edges and vertices) and their relations is called a multi-block topology. The definition of a multi-block topology is here constrained by the requirement that the grids are continuous across the elementary faces.

For each block, the structured grid divides the block into the grid cells $\Omega_{i,j,k}$ with $i = 1 \dots N_i$, $j = 1 \dots N_j$, and $k = 1 \dots N_k$. The grid cell indices i , j , and k correspond to the curvilinear coordinates ξ , η , and ζ , respectively. Each grid cell has the topology of a cube, and thus its boundary surface $\partial\Omega_{i,j,k}$ is the union of six cell faces denoted by $S_{i-1/2,j,k}$, $S_{i+1/2,j,k}$, $S_{i,j-1/2,k}$, $S_{i,j+1/2,k}$, $S_{i,j,k-1/2}$ and $S_{i,j,k+1/2}$ (see figure 3.2).

Figure 3.2: Cell faces of cell $\Omega_{i,j,k}$

3.4 BASIC SCHEME

In a cell-centred finite-volume approach, cell-averaged values of the flow state (U) are thought of as being located at the centres of the grid cells, and the discrete equations are obtained by applying the continuous equations in integral form (equation (2.28) for the TLNS equations) to every grid cell $\Omega_{i,j,k}$. Define the cell volume $V_{i,j,k}$,

$$V_{i,j,k} = \iiint_{\Omega_{i,j,k}} dV. \quad (3.1)$$

and the cell-face area vectors $\vec{A}_{i+1/2,j,k}^{(i)}$, $\vec{A}_{i,j+1/2,k}^{(j)}$, and $\vec{A}_{i,j,k+1/2}^{(k)}$,

$$\begin{aligned} \vec{A}_{i+1/2,j,k}^{(i)} &= \iint_{S_{i+1/2,j,k}} \vec{m} dA, \\ \vec{A}_{i,j+1/2,k}^{(j)} &= \iint_{S_{i,j+1/2,k}} \vec{m} dA, \\ \vec{A}_{i,j,k+1/2}^{(k)} &= \iint_{S_{i,j,k+1/2}} \vec{m} dA, \end{aligned} \quad (3.2)$$

where \vec{m} is the unit vector normal to the face $S_{i+1/2,j,k}$, $S_{i,j+1/2,k}$, or $S_{i,j,k+1/2}$, and pointing in the positive i , j , or k direction, respectively. The computation of these geometric quantities is done such that the cell-face area vectors are exact if the edges of the grid cells are straight lines (which may be freely assumed), and such that the summation of the area vectors of one cell equals zero, so that equation (2.19) is satisfied.

Considering in equation (2.18) the volume segment Ω to be a grid cell $\Omega_{i,j,k}$, this equation becomes

$$\begin{aligned} \frac{dU_{i,j,k}}{dt} + R_{i,j,k} &= 0, \\ R_{i,j,k} &= \frac{D_{i,j,k}}{V_{i,j,k}}, \end{aligned} \quad (3.3)$$

with the discrete flow state $U_{i,j,k}$ defined by taking the average of the continuous flow state U over the grid cell $\Omega_{i,j,k}$,

$$U_{i,j,k} = \frac{1}{V_{i,j,k}} \iiint_{\Omega_{i,j,k}} U dV, \quad (3.4)$$

and with the residual $R_{i,j,k}$ depending on the flux balance $D_{i,j,k}$ given by

$$D_{i,j,k} = \int \int_{\partial\Omega_{i,j,k}} (\mathcal{F}^c(U) - \mathcal{F}^d(U, \nabla U)) \cdot \vec{m} dA, \quad (3.5)$$

with the convective flux matrix (\mathcal{F}^c) depending on the flow state vector only, and with the diffusive flux matrix (\mathcal{F}^d) depending on the flow state vector and on its gradient through the stress tensor and the heat-flux vector (equation (2.25)). For steady flows, the discrete equations reduce to

$$R_{i,j,k} = 0. \quad (3.6)$$

The flux balance $D_{i,j,k}$ is defined by summing the different flux contributions from the six faces of cell $\Omega_{i,j,k}$. To obtain a conservative discretization, a unique flux must be defined for each cell face. The convective flux across cell $S_{i,j,k+1/2}$, for example, is approximated by

$$\begin{aligned} F_{i,j,k+1/2}^c &= \iint_{S_{i,j,k+1/2}} \mathcal{F}^c(U) \cdot \vec{m} \, dA \\ &\approx \mathcal{F}^c(U_{i,j,k+1/2}) \cdot \vec{A}_{i,j,k+1/2}^{(k)}. \end{aligned} \quad (3.7)$$

To obtain the flow state $U_{i,j,k+1/2}$ at cell faces from the flow states assigned to the cell centres, a central scheme using simple averaging is employed,

$$U_{i,j,k+1/2} = \frac{1}{2} (U_{i,j,k} + U_{i,j,k+1}). \quad (3.8)$$

The gradient of the flow state at cell faces (needed for the diffusive flux) is evaluated by a finite-difference approximation. The precise discretization of the convective and diffusive fluxes is presented in sections 3.4.1 and 3.4.2.

As is well known, this central difference discretization of the convective fluxes leads to so-called odd-even decoupling for the Euler equations, which means in 1D that the solution in grid cells with even indices is independent from the solution in cells with odd indices (not considering the boundary conditions). In other words, there is no diffusion present and thus point-to-point oscillations will not be suppressed. This will generally prevent any solution method from converging. To resolve this problem, a form of artificial diffusion is added, as presented in section 3.4.3. Furthermore, this artificial diffusion is defined such that shock waves are represented in an acceptable manner, i.e. such that the entropy condition is satisfied (section 2.4) and no strong oscillations around the shocks occur. Unique fluxes are defined at the cell faces for the artificial diffusion, so that the conservative discretization is retained.

Summarizing, the discrete mathematical model consists of the set of equations given by equation (3.6) for each grid cell ($i = 1 \dots N_i$, $j = 1 \dots N_j$, and $k = 1 \dots N_k$) in each block. The residual $R_{i,j,k}$ is the sum of the convective, diffusive, and artificial diffusive flux balances, divided by the cell volume,

$$R_{i,j,k} = (D_{i,j,k}^c - D_{i,j,k}^d - D_{i,j,k}^a) / V_{i,j,k}. \quad (3.9)$$

The flux balances are found by summing the appropriate fluxes across the six faces of each cell,

$$\begin{aligned} D_{i,j,k}^c &= F_{i+1/2,j,k}^c - F_{i-1/2,j,k}^c + F_{i,j+1/2,k}^c - F_{i,j-1/2,k}^c + F_{i,j,k+1/2}^c - F_{i,j,k-1/2}^c, \\ D_{i,j,k}^d &= F_{i+1/2,j,k}^d - F_{i-1/2,j,k}^d + F_{i,j+1/2,k}^d - F_{i,j-1/2,k}^d + F_{i,j,k+1/2}^d - F_{i,j,k-1/2}^d, \\ D_{i,j,k}^a &= F_{i+1/2,j,k}^a - F_{i-1/2,j,k}^a + F_{i,j+1/2,k}^a - F_{i,j-1/2,k}^a + F_{i,j,k+1/2}^a - F_{i,j,k-1/2}^a. \end{aligned} \quad (3.10)$$

(The diffusive flux balance D^d equals zero for the Euler equations.)

The space discretization of equation (2.18) that now has been obtained has the following properties:

- The discretization is conservative, a necessary requirement for the correct representation of shock waves (see Lax and Wendroff [66]).
- The discretization is uniform-flow consistent, i.e. a constant uniform flow satisfies the equations exactly. This is a consequence of the summation of area vectors of one grid cell being equal to zero. Generally, this allows coarse and highly stretched grids in the far-field domain, where the flow solution is a small perturbation of a uniform flow.
- If the grid is sufficiently smooth (i.e. it is considered to be the image under a twice differentiable mapping of a uniform grid in the unit cube), the discretization is second-order accurate in regions where the flow is smooth (thus, outside flow discontinuities such as shock waves). This is a consequence of using central differences, and of using fourth-order artificial diffusion in smooth-flow regions (see section 3.4.3).
- The finite-volume discretization can also be seen as a finite-difference discretization of the equations in curvilinear coordinates (equation (2.23) or (2.33)), identifying J with the cell volume V and identifying $J\nabla\xi$ with the cell-face area vector $\vec{A}^{(i)}$ (see e.g. [116]). Sometimes the space discretization is more easily understood from this point of view.

3.4.1 Convective fluxes

The expression for the discrete convective flux across a certain cell face, is obtained by taking the product of the convective flux matrix \mathcal{F}^c (given by equation (2.7)) with the cell-face area vector. Let (I, J, K) be a cyclic permutation of (i, j, k) . For all cell faces $S_{I,J,K+1/2}$, the following expression results:

$$F_{I,J,K+1/2}^c = \begin{pmatrix} \rho q \\ (\rho \vec{u}) q + p \vec{A}^{(K)} \\ (\rho E + p) q \end{pmatrix}_{I,J,K+1/2}, \quad (3.11)$$

with $q = \vec{u} \cdot \vec{A}^{(K)}$. The values at the cell face of the flow-state vector U and of the pressure p are found by averaging,

$$\begin{aligned} U_{I,J,K+1/2} &= \frac{1}{2}(U_{I,J,K} + U_{I,J,K+1}), \\ p_{I,J,K+1/2} &= \frac{1}{2}(p_{I,J,K} + p_{I,J,K+1}). \end{aligned} \quad (3.12)$$

The volume flux $q_{I,J,K+1/2}$ is calculated by

$$q_{I,J,K+1/2} = \frac{(\rho \vec{u})_{I,J,K+1/2} \cdot \vec{A}_{I,J,K+1/2}^{(K)}}{\rho_{I,J,K+1/2}}. \quad (3.13)$$

As indicated by Vinokur [116], this is a reasonably efficient method for evaluating the convective flux in line with the finite-volume concept. It gives better results near solid

surfaces than computing the convective fluxes first at cell centres and subsequently averaging the fluxes (instead of the flow states) to the cell faces.

It can be shown that constant total enthalpy is an exact solution of the discrete Euler equations, consistent with the continuous equations, as follows. In case of constant total enthalpy, the balance for one grid cell of the discrete convective fluxes in the energy equation equals the total enthalpy times the balance of convective fluxes in the continuity equation. If the same holds true for the balance of the artificial diffusive fluxes (see section 3.4.3), then satisfaction of the discrete continuity equation implies that the discrete energy equation is satisfied if the total enthalpy is constant.

3.4.2 Diffusive fluxes for the TLNS equations

For the discretization of the diffusive fluxes of the TLNS equations, we consider the equations in curvilinear coordinates (equations (2.23)). The expression for the discrete diffusive flux across a certain cell face is then given by equation (2.24), identifying $J\nabla\xi$ with the cell-face area vector as indicated above. Let (I, J, K) be a cyclic permutation of (i, j, k) . If the K direction corresponds to a thin-layer normal direction, then for all cell faces $S_{I,J,K+1/2}$, the diffusive flux is given by the following expression:

$$F_{I,J,K+1/2}^d = \begin{pmatrix} 0 \\ \vec{\tau}^n \\ \vec{u} \cdot \vec{\tau}^n - Q^n \end{pmatrix}_{I,J,K+1/2} \left\| \vec{A}_{I,J,K+1/2}^{(K)} \right\|. \quad (3.14)$$

For the other directions, the diffusive fluxes are equal to zero.

The velocity at the cell face is found from averaged cell-centre values of density and momentum,

$$\vec{u}_{I,J,K+1/2} = \frac{(\rho\vec{u})_{I,J,K} + (\rho\vec{u})_{I,J,K+1}}{\rho_{I,J,K} + \rho_{I,J,K+1}}. \quad (3.15)$$

which is consistent with the definition of the volume flux in the expression for the convective fluxes (equation (3.13)).

The normal vector component $\vec{\tau}^n$ of the total stress tensor and the heat-flux component in normal direction Q^n are given by equation (2.25), and thus are given in discretized form by

$$\begin{aligned} \vec{\tau}_{I,J,K+1/2}^n &= (\mu + \mu_t)_{I,J,K+1/2} \left(\frac{\partial \vec{u}}{\partial n} + \frac{1}{3} \left(\vec{n} \cdot \frac{\partial \vec{u}}{\partial n} \right) \vec{n} \right)_{I,J,K+1/2}, \\ Q_{I,J,K+1/2}^n &= -(\kappa_h + \kappa_t)_{I,J,K+1/2} \left(\frac{\partial T}{\partial n} \right)_{I,J,K+1/2}, \end{aligned} \quad (3.16)$$

with $\vec{n} = \vec{A}^{(K)} / \|\vec{A}^{(K)}\|$ the unit vector normal to the cell face. The normal derivatives of velocity $(\partial \vec{u} / \partial n)$ and temperature $(\partial T / \partial n)$ at cell face $S_{I,J,K+1/2}$ are approximated

by central differences as is usually done for the TLNS equations [116],

$$\begin{aligned} \left(\frac{\partial \vec{u}}{\partial n} \right)_{I,J,K+1/2} &= \frac{\vec{u}_{I,J,K+1} - \vec{u}_{I,J,K}}{(\Delta n)_{I,J,K+1/2}}, \\ \left(\frac{\partial T}{\partial n} \right)_{I,J,K+1/2} &= \frac{T_{I,J,K+1} - T_{I,J,K}}{(\Delta n)_{I,J,K+1/2}}, \\ (\Delta n)_{I,J,K+1/2} &= \frac{V_{I,J,K+1} + V_{I,J,K}}{2 \left\| \vec{A}_{I,J,K+1/2}^{(K)} \right\|}. \end{aligned} \quad (3.17)$$

The dynamic- and eddy-viscosity coefficients are evaluated at cell centres [73, 110], according to equation (2.15) and chapter 5, respectively. Values at cell faces are found by averaging,

$$\begin{aligned} \mu_{I,J,K+1/2} &= \frac{1}{2}(\mu_{I,J,K} + \mu_{I,J,K+1}), \\ (\mu_t)_{I,J,K+1/2} &= \frac{1}{2}((\mu_t)_{I,J,K} + (\mu_t)_{I,J,K+1}). \end{aligned} \quad (3.18)$$

3.4.3 Scalar artificial diffusion model

In the basic scheme, the artificial diffusive flux is defined according to the scalar diffusion model. This model is basically the model developed by Jameson et al. [48, 52]. The artificial-diffusive divergence is defined as a blending of second-order differences to obtain physically acceptable representations of shock waves and fourth-order differences to damp high-frequency modes (preventing odd-even decoupling). This results in first-order and third-order differences for the fluxes at cell faces. Let (I, J, K) be a cyclic permutation of (i, j, k) . For all cell faces $S_{I,J,K+1/2}$, the artificial diffusive flux is given by

$$F_{I,J,K+1/2}^a = \lambda_{I,J,K+1/2}^{(K)} (f_{I,J,K+1/2}^{(2)} - f_{I,J,K+1/2}^{(4)}), \quad (3.19)$$

with $f^{(2)}$ and $f^{(4)}$ the first- and third-order differences, and λ a scaling factor which ensures that the artificial diffusion has the correct magnitude compared to the convective fluxes.

The first-order difference $f^{(2)}$ is evaluated as

$$f_{I,J,K+1/2}^{(2)} = \epsilon_{I,J,K+1/2}^{(2)} (U_{I,J,K+1}^* - U_{I,J,K}^*), \quad (3.20)$$

with $U^* = (\rho, \rho \vec{u}, \rho H)^T$. The enthalpy is used instead of the energy, so that constant enthalpy is an exact solution of the discrete Euler equations. For a scalar convection equation with a wave speed λ , it may easily be seen that taking $\epsilon^{(2)} = 1/2$ (and $f^{(4)} \equiv 0$) would lead to a first-order accurate upwind scheme. Near shock waves, such a first-order formulation ensures an oscillation-free solution (see e.g. [95]). In order to obtain a second-order accurate discretization away from shock waves, the first-order difference is switched on only near shock waves. This may be done by letting the factor $\epsilon^{(2)}$ depend on a shock sensor ν ,

$$\epsilon_{I,J,K+1/2}^{(2)} = \min \left\{ \frac{1}{2}, k^{(2)} \max \{ \nu_{I,J,K}, \nu_{I,J,K+1} \} \right\}, \quad (3.21)$$

with $k^{(2)}$ a numerical parameter (taken equal to 1). The shock sensor is calculated using the second-order difference of the pressure [52],

$$\nu_{I,J,K} = \frac{|p_{I,J,K+1} - 2p_{I,J,K} + p_{I,J,K-1}|}{p_{I,J,K+1} + 2p_{I,J,K} + p_{I,J,K-1}}. \quad (3.22)$$

In regions where the pressure is smooth (thus away from shock waves), the shock sensor is of $O(h^2)$ (with h the mesh size and considering smooth grids), and thus the artificial-diffusive residual ($R_{i,j,k} = D_{i,j,k}^a/V_{i,j,k}$ with $f^{(4)} \equiv 0$) becomes of $O(h^3)$. For transonic flows, the shock sensor is usually about 0.1 near shock waves. Thus, $\epsilon^{(2)}$ has a value that is considerably smaller than the value it would have for the first-order accurate upwind scheme which would ensure an oscillation-free solution. However, in practice oscillation-free shocks are also obtained with this lower value, at least for transonic flows. This is partly due to the fact that a scalar scaling factor λ being equal to the maximum wave speed is used (as will be seen later on), which increases the artificial diffusion for the characteristic equations with lower wave speeds.

The third-order difference $f^{(4)}$ is evaluated as

$$f_{I,J,K+1/2}^{(4)} = \epsilon_{I,J,K+1/2}^{(4)} (U_{I,J,K+2}^* - 3U_{I,J,K+1}^* + 3U_{I,J,K}^* - U_{I,J,K-1}^*). \quad (3.23)$$

To avoid oscillations, the third-order difference must be turned off near shock waves, which is done by setting the coefficient $\epsilon^{(4)}$ as

$$\epsilon_{I,J,K+1/2}^{(4)} = k^{(4)} \max \left\{ 0, \frac{1}{64} - k^{(s)} \epsilon_{I,J,K+1/2}^{(2)} \right\}, \quad (3.24)$$

with $k^{(4)}$ and $k^{(s)}$ numerical parameters. Considering only the third-order differences ($f^{(2)} \equiv 0$), the artificial-diffusive residual can be seen to be of $O(h^3)$. The numerical parameter $k^{(4)}$ influences the convergence speed of the solution method, and is limited by stability requirements for the explicit Runge–Kutta time integration schemes described in the next chapter (or conversely, for a given value of $k^{(4)}$, one should choose a Runge–Kutta scheme with a sufficiently large stability region), see appendix B. We take $k^{(4)} = 2$, consistent with the Runge–Kutta schemes that will be chosen in the next chapter. The parameter $k^{(s)}$ influences the switching of the fourth-order diffusion near shocks. If chosen too small, the third-order difference may not be switched off sufficiently near shocks, thus causing oscillations, while if chosen too large, the convergence speed may be strongly affected. In practice, $k^{(s)} = 1/2$ has been found to be suitable.

As stated before, the scaling factor λ is defined such that the artificial diffusion has the correct magnitude compared to the convective fluxes. Based on an analogy with the first-order accurate upwind scheme, it is seen that the scaling factor must be proportional to the convective wave speed. For a set of equations, the wave speeds are equal to the eigenvalues of the Jacobian of the convective flux dF^c/dU (see appendix A). At cell centres, the scalar scaling factor is defined as the spectral radius of the convective Jacobian in K direction (multiplied by the area vector),

$$\lambda_{I,J,K}^{(K)} = \left| \vec{u}_{I,J,K} \cdot \vec{A}_{I,J,K}^{(K)} \right| + c_{I,J,K} \left\| \vec{A}_{I,J,K}^{(K)} \right\|, \quad (3.25)$$

in which $\bar{A}_{I,J,K}^{(K)}$ is the average of the area vectors in K direction,

$$\bar{A}_{I,J,K}^{(K)} = \frac{1}{2}(\bar{A}_{I,J,K+1/2}^{(K)} + \bar{A}_{I,J,K-1/2}^{(K)}), \quad (3.26)$$

and $c_{I,J,K}$ is the local speed of sound ($c = \sqrt{\gamma p / \rho}$). The scaling factors at cell faces are obtained by averaging,

$$\lambda_{I,J,K+1/2}^{(K)} = \frac{1}{2}(\lambda_{I,J,K}^{(K)} + \lambda_{I,J,K+1}^{(K)}). \quad (3.27)$$

3.5 BOUNDARY CONDITIONS

The space discretization of the basic scheme must be completed by discretizing the boundary conditions. Not only conditions at the flow domain boundary are considered, but also conditions at faces between blocks (so-called internal faces). These latter conditions deal with the coupling of the equations in different blocks. Boundary conditions are applied at the elementary faces of the multi-block grid, so that at one block face (containing several elementary faces) different boundary conditions can be applied. They are used to evaluate the convective, diffusive, and artificial diffusive fluxes across cell faces that are located at an elementary face.

For the discretization of the boundary conditions, the dummy-cell concept is used. At internal faces, values of the flow-state vector are needed from the two blocks adjacent to the face. Also, the artificial diffusive flux at a cell face just inside a certain block, having a 4-point stencil, requires values of the flow-state vector of the adjacent block. In order to keep the evaluation of the fluxes for a certain block as much as possible local to that block (i.e. using only data belonging to that block, which simplifies both the data management and the computational algorithm), one layer of so-called dummy cells is introduced at each block face (see figure 3.3). The layer of dummy cells outside a certain block face is indicated by indices $i = 0$, $i = N_i + 1$, $j = 0$, $j = N_j + 1$, $k = 0$, or $k = N_k + 1$ depending on its location. Values of the state-vector in the dummy cells are obtained by using the boundary or coupling conditions. Thus, for each block the flow-state vector $U_{i,j,k}$ is defined for

$$(i, j, k) \in \{0 \dots N_i + 1\} \times \{0 \dots N_j + 1\} \times \{0 \dots N_k + 1\}. \quad (3.28)$$

Then, fluxes across block faces can be evaluated in the same manner as the fluxes at grid-cell faces internal to the block, using the dummy-cell values.

To keep the evaluation of the artificial diffusive fluxes at block faces completely local, a second layer of dummy cells would be required, due to the 4-point stencil. However, for certain boundary conditions (e.g. at internal faces where the grid is not smooth across the face [60]) the definition of flow-state values at this second dummy cell would be computationally complex, if possible at all. Therefore, such a second layer of dummy cells is not employed, and the artificial diffusive fluxes at internal faces must be evaluated using data from the adjacent block directly.

As stated at the beginning of this chapter, the solution procedure will be based on the time-integration concept. Complete time steps will be performed for each block successively. At the beginning of a time step in a certain block, data will be

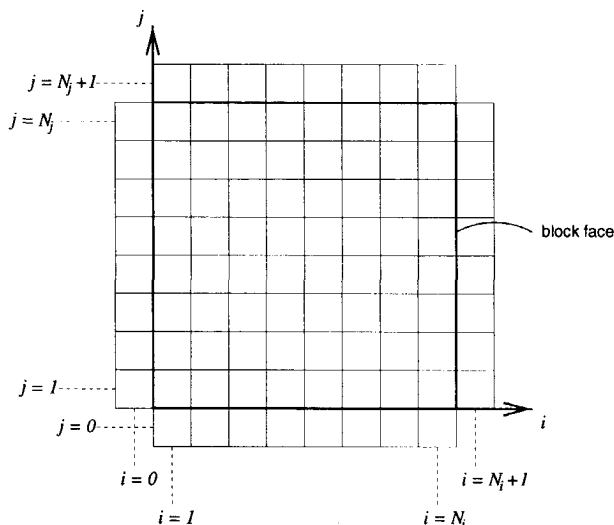


Figure 3.3: Example of block with dummy cells (2D)

exchanged between that block and its neighbours. This exchange of data will consist of applying the coupling conditions at internal faces, or in other words, of setting the dummy-cell values at internal faces.

The discretization of the boundary and coupling conditions now consists of specifying the method by which the values of the flow variables are obtained in the dummy cells. Furthermore, also discrete boundary conditions are needed for the artificial diffusion. In the following sections, the discrete boundary conditions will be described, considering for simplicity the row of dummy cells at $i = 0$.

3.5.1 Solid surface

Let the block face be a solid surface. In the case of the TLNS equations, the dummy-cell values must be evaluated such that the no-slip condition is satisfied. This can be done by setting the momentum vector as follows:

$$(\rho \vec{u})_{0,j,k} = -(\rho \vec{u})_{1,j,k}. \quad (3.29)$$

In this way, the velocity at the solid surface is identically zero, and thus the discrete convective flux at the surface is exactly zero apart from the pressure contribution (see equation (3.11)). Furthermore, the adiabatic wall condition for the temperature is discretized as,

$$T_{0,j,k} = T_{1,j,k}, \quad (3.30)$$

assuming that the grid lines are locally orthogonal to the solid surface. To complete the definition of the flow-state vector in the dummy cell, a fifth dependent variable is needed. Here, the pressure is linearly extrapolated to the dummy cell in computa-

tional space from the values internal to the block,

$$p_{0,j,k} = 2p_{1,j,k} - p_{2,j,k}. \quad (3.31)$$

In the case of the Euler equations, the dummy-cell values must be evaluated such that the slip condition is satisfied. This can be done by reflecting the normal component of the momentum vector across the solid surface. The tangential components (given by the vector $(\rho\vec{u})_{\text{tan}}$) are linearly extrapolated in the computational space from the values internal to the block. Thus, the following expression for the momentum in the dummy cell results

$$\begin{aligned} (\rho\vec{u})_{0,j,k} &= (\rho u)_m \vec{m} + (\rho\vec{u})_{\text{tan}}, \\ (\rho u)_m &= -(\rho\vec{u})_{1,j,k} \cdot \vec{m}, \\ (\rho\vec{u})_{\text{tan}} &= (\rho\vec{u})_E - ((\rho\vec{u})_E \cdot \vec{m}) \vec{m}, \\ (\rho\vec{u})_E &= 2(\rho\vec{u})_{1,j,k} - (\rho\vec{u})_{2,j,k}, \end{aligned} \quad (3.32)$$

with \vec{m} the unit vector normal to the solid surface. Furthermore, the pressure and the total enthalpy per unit mass are linearly extrapolated to the dummy cell,

$$\begin{aligned} p_{0,j,k} &= 2p_{1,j,k} - p_{2,j,k}, \\ H_{0,j,k} &= 2H_{1,j,k} - H_{2,j,k}. \end{aligned} \quad (3.33)$$

Again, only the pressure contribution to the discrete convective flux at the surface is not equal to zero, since the normal velocity component at the surface is identically zero.

For viscous-flow computations, it is important that the artificial diffusion does not interfere with the physical diffusion. For inviscid-flow computations, too much artificial diffusion near the solid surface may create an artificial boundary layer. Thus, the artificial diffusive fluxes in normal direction should not be too large near the surface. A suitable boundary condition [33, 112] is to set the artificial diffusive fluxes at the surface equal to zero,

$$F_{1/2,j,k}^a = 0. \quad (3.34)$$

For inviscid-flow computations, the numerical accuracy may be improved by modifying the artificial diffusive fluxes at the cell faces $i = 3/2$, just inside the block, such that these fluxes do not depend on the dummy-cell values. Essentially, this may be done by writing the general formulation of these fluxes as

$$F_{3/2,j,k}^a = \lambda_{3/2,j,k} \left(f_{3/2,j,k}^{(2)} - f_{3/2,j,k}^{(4)} \right), \quad (3.35)$$

with

$$f_{3/2,j,k}^{(4)} = \epsilon_{3/2,j,k}^{(4)} \left((\delta_i^2 U^*)_{2,j,k} - (\delta_i^2 U^*)_{1,j,k} \right), \quad (3.36)$$

and setting the following terms to zero:

$$\begin{aligned} (\delta_i^2 U^*)_{1,j,k} &= 0, \\ f_{3/2,j,k}^{(2)} &= 0. \end{aligned} \quad (3.37)$$

Since there is no first-order term remaining in these fluxes, the coefficient $\epsilon^{(4)}$ is computed as

$$\epsilon_{3/2,j,k}^{(4)} = \frac{1}{64} k^{(4)}. \quad (3.38)$$

A similar modification for viscous flows is not appropriate, since it reduces the order of accuracy from two to one, as shown in [112]. Note that now the artificial-diffusive flux balances in the first two cells interior to the block are given by

$$\begin{aligned} D_{1,j,k}^a &= F_{3/2,j,k}^a, \\ D_{2,j,k}^a &= F_{5/2,j,k}^a - F_{3/2,j,k}^a. \end{aligned} \quad (3.39)$$

3.5.2 Far-field boundary

At a far-field boundary, the boundary condition using characteristic variables, as described in section 2.5.2, is applied. It consists of requiring the characteristic variables of incoming characteristics to be equal to their free-stream values.

The discrete equations require a specification of all five basic flow variables in the dummy cells. These variables are derived from the dummy-cell values of the characteristic variables, which are set to free-stream values for incoming characteristics (consistent with the continuous boundary condition), and are extrapolated from the inside of the flow field for outgoing characteristics.

Let \vec{m} be the outward unit vector normal to the far-field boundary. The variables R_{in} and R_{out} always belong to an incoming and an outgoing characteristic, respectively, and thus their dummy-cell values are set as

$$\begin{aligned} (R_{\text{in}})_{0,j,k} &= \vec{u}_{\infty} \cdot \vec{m} - 2c_{\infty}/(\gamma - 1), \\ (R_{\text{out}})_{0,j,k} &= \vec{u}_{1,j,k} \cdot \vec{m} + 2c_{1,j,k}/(\gamma - 1), \end{aligned} \quad (3.40)$$

with c the local speed of sound. The dummy-cell values of the other three characteristic variables (the two tangential velocity components, given by the vector \vec{u}_{tan} , and the entropy S) and the total enthalpy H are set depending on whether the flow is coming in through the boundary, or going out. In case of inflow ($\vec{u}_{\infty} \cdot \vec{m} < 0$) they are set to their free-stream values,

$$\begin{aligned} (\vec{u}_{\text{tan}})_{0,j,k} &= \vec{u}_{\infty} - (\vec{u}_{\infty} \cdot \vec{m})\vec{m}, \\ S_{0,j,k} &= c_v \ln(p_{\infty}/\rho_{\infty}^{\gamma}), \\ H_{0,j,k} &= H_{\infty}, \end{aligned} \quad (3.41)$$

while in case of outflow ($\vec{u}_{\infty} \cdot \vec{m} \geq 0$) they are given by

$$\begin{aligned} (\vec{u}_{\text{tan}})_{0,j,k} &= \vec{u}_{1,j,k} - (\vec{u}_{1,j,k} \cdot \vec{m})\vec{m}, \\ S_{0,j,k} &= c_v \ln(p_{1,j,k}/\rho_{1,j,k}^{\gamma}), \\ H_{0,j,k} &= H_{1,j,k}. \end{aligned} \quad (3.42)$$

By treating total enthalpy separately in the numerical scheme (not computing it from the characteristic variables), constant enthalpy becomes an exact solution of the discrete equations in case of inviscid flow.

To define a boundary condition for the artificial diffusion (to compute the artificial diffusive flux $F_{1/2,j,k}^a$), imagine a second row of dummy-cell values $U_{-1,j,k}^*$. Following [33], the boundary condition consists of setting the second-order differences of the flow-state vector at $i = 0$ equal to zero,

$$U_{1,j,k}^* - 2U_{0,j,k}^* + U_{-1,j,k}^* = 0, \quad (3.43)$$

which is equivalent to setting the term $f^{(4)}$ as

$$f_{1/2,j,k}^{(4)} = \epsilon_{1/2,j,k}^{(4)} (U_{2,j,k}^* - 2U_{1,j,k}^* + U_{0,j,k}^*). \quad (3.44)$$

3.5.3 Symmetry plane

At a symmetry plane, the continuous boundary condition is discretized by setting the dummy-cell values of pressure and density equal to the values inside the block (thus, normal gradient equals zero if the grid lines are assumed orthogonal to the symmetry plane),

$$\begin{aligned} \rho_{0,j,k} &= \rho_{1,j,k}, \\ p_{0,j,k} &= p_{1,j,k}, \end{aligned} \quad (3.45)$$

while the momentum vector is reflected with respect to the symmetry plane,

$$\begin{aligned} (\rho \vec{u})_{0,j,k} &= (\rho \vec{u})_{1,j,k} - 2(\rho u_m) \vec{m}, \\ (\rho u_m) &= (\rho \vec{u})_{1,j,k} \cdot \vec{m}, \end{aligned} \quad (3.46)$$

with \vec{m} the unit vector normal to the symmetry plane. The artificial diffusive fluxes at the symmetry plane are set to zero.

3.5.4 Plane of infinite continuation

The plane of infinite continuation has been defined in order to enable a 2D calculation with a 3D method (section 2.5.4). In the direction perpendicular to the 2D domain, i.e. between two planes of infinite continuation, only one grid cell is defined. Since the 2D solution is independent of the third direction, the dummy cell values at the two planes must be set equal to the values internal to the block,

$$\begin{aligned} \rho_{0,j,k} &= \rho_{1,j,k}, \\ (\rho \vec{u})_{0,j,k} &= (\rho \vec{u})_{1,j,k}, \\ (\rho E)_{0,j,k} &= (\rho E)_{1,j,k}. \end{aligned} \quad (3.47)$$

The artificial diffusive fluxes at the two planes are set equal to zero.

3.5.5 Internal faces

Consider the elementary face F that forms an interface between block B^1 and block B^2 . We will discuss how for this face the dummy-cell values of both blocks, as well as the convective, diffusive, and artificial diffusive fluxes through the face can be calculated. The definition of the dummy-cell values depends on the grid properties at the internal face. The following cases are discerned (see figure 3.4):

- faces with C^1 -continuity of grids (thus, the grid is smooth across the internal face),
- faces with C^0 -continuity of grid lines (thus, both the normal mesh size and the slope of the grid lines may jump across the internal face), and
- faces with partial continuity of grid lines (only a subset of the grid lines are C^0 -continuous across the internal face).

Allowing for grids which are only C^0 -continuous simplifies the multi-block grid generation task. It is further allowed to locally refine the grid in certain blocks by e.g. a factor 2 or 4 (thus locally increasing the numerical accuracy), so that only partial continuity of the grid lines across the internal faces may result. For the multi-grid scheme on a multi-block grid that will be discussed in section 4.4.1, partial continuity may also occur on coarse grid levels.

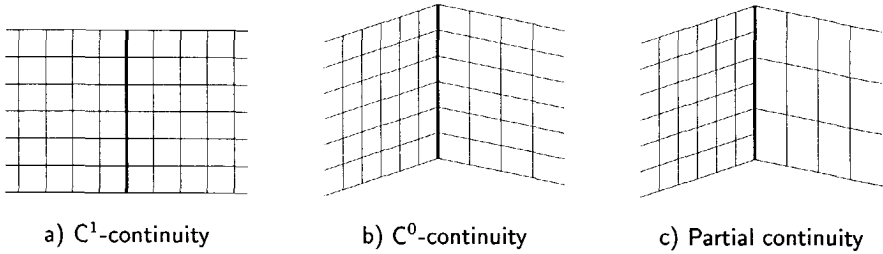


Figure 3.4: Grid properties at an internal face

With respect to face F , define the indices (i^1, j^1, k^1) for block B^1 and the indices (i^2, j^2, k^2) for block B^2 . Face F is given by $i^1 = 1/2$ and by $i^2 = 1/2$, and thus the dummy cells of block B^1 at face F are given by $i^1 = 0$. The variables related to block B^1 will be indicated by a superscript (1) and are functions of (i^1, j^1, k^1) , while those related to block B^2 will be indicated by a superscript (2) and are functions of (i^2, j^2, k^2) . For simplicity, it is assumed that the indices $j^1 = j^2 = j$ and $k^1 = k^2 = k$.

Requirements on the dummy-cell values

The convective and physical diffusive fluxes at the internal face F are computed in the same manner as the fluxes at grid-cell faces internal to the blocks, using the dummy-cell values. In order to obtain a conservative discretization, these fluxes at the internal face have to be defined uniquely. For the convective flux, this means that the values of density, momentum, total energy, and pressure at the block face must be unique (see equation (3.12)). This results in the following requirement:

$$W_{0,j,k}^{(1)} + W_{1,j,k}^{(1)} = W_{0,j,k}^{(2)} + W_{1,j,k}^{(2)}, \quad (3.48)$$

with W the flow-state vector U extended with pressure, $W = (\rho, \rho \vec{u}, \rho E, p)^T$.

When a physical diffusive flux is evaluated at the internal face, then the normal

derivatives of velocity and temperature must be unique (see equation (3.17)):

$$\begin{aligned} \frac{\vec{u}_{1,j,k}^{(1)} - \vec{u}_{0,j,k}^{(1)}}{\Delta n_{1/2,j,k}^{(1)}} &= \frac{\vec{u}_{0,j,k}^{(2)} - \vec{u}_{1,j,k}^{(2)}}{\Delta n_{1/2,j,k}^{(2)}}, \\ \frac{T_{1,j,k}^{(1)} - T_{0,j,k}^{(1)}}{\Delta n_{1/2,j,k}^{(1)}} &= \frac{T_{0,j,k}^{(2)} - T_{1,j,k}^{(2)}}{\Delta n_{1/2,j,k}^{(2)}}. \end{aligned} \quad (3.49)$$

Requirements (3.48) and (3.49) cannot be satisfied at the same time when the grids are not C^1 -continuous across the internal face. In these cases, the dummy-cell values will be set satisfying requirement (3.48). This means that the physical diffusive fluxes cannot be calculated using these dummy-cell values, but the values from the adjacent block have to be used directly, taking the grid properties into account. In the remainder of this section, we will deal with the computation of the dummy-cell values and of the artificial diffusive fluxes (which are not computed using the dummy-cell values).

Faces without local grid refinement

First, we consider the case when blocks are not locally refined. When the grids are smoothly connected across face F , the dummy-cell values of block B^1 can be simply copied from block B^2 , and vice versa,

$$\begin{aligned} W_{0,j,k}^{(1)} &= W_{1,j,k}^{(2)}, \\ W_{0,j,k}^{(2)} &= W_{1,j,k}^{(1)}. \end{aligned} \quad (3.50)$$

Boundary conditions for grids which are non-smooth across the internal face (only C^0 -continuous) were developed by Kassies et al. [60]. The main idea of these boundary conditions is to compute a gradient of the flow-state vector W at each cell face on face F , and subsequently use this gradient to compute a unique value of the flow-state vector W itself at each cell face (see [60] for the details). Dummy-cell values are then simply obtained by linear extrapolation, satisfying requirement (3.48). The artificial diffusive fluxes at the internal face are computed using the flow-state values of the two adjacent blocks in exactly the same way as they are computed inside blocks (irrespective of the grid properties).

Faces with local grid refinement

We now consider the case when there are blocks present that are locally refined. Thus, it is assumed that at face F the grid lines are only partly continuous. For the Euler equations, the boundary condition at this type of face has been extensively described in reference [60]. In case of local grid refinement, the actual grid (in all blocks) has been obtained from a basic grid which is at least C^0 -continuous everywhere. In a certain block, the grid dimensions may have been increased by an integer factor (possibly different in each direction) with respect to the basic grid, while maintaining the same grid distribution.

Consider face F with adjacent blocks B^1 and B^2 . In these two blocks, the actual grid has dimensions (N_i^1, N_j^1, N_k^1) and (N_i^2, N_j^2, N_k^2) , respectively, while the basic grid has dimensions (N_i^{b1}, N_j^b, N_k^b) and (N_i^{b2}, N_j^b, N_k^b) . (Note that the basic grid has the same dimensions in the j and k directions, i.e. along face F , for both blocks.) Then, the refinement factors of B^2 with respect to B^1 in the i , j , and k directions are defined as

$$M_i = \frac{N_i^2/N_i^{b2}}{N_i^1/N_i^{b1}}, \quad M_j = \frac{N_j^2}{N_j^1}, \quad M_k = \frac{N_k^2}{N_k^1}. \quad (3.51)$$

We require that one of the two blocks is the fine block for all directions, and assume that this is block B^2 , i.e.

$$M_i \geq 1, \quad M_j \geq 1, \quad M_k \geq 1. \quad (3.52)$$

The grid in block B^2 can be reduced to a grid that is C^0 -continuous with the grid of block B^1 across face F by fusing grid cells. This reduced grid has one grid cell normal to F (in the i direction) and the same number of grid cells as block B^1 along face F (N_j^1 by N_k^1 in the j and k directions). A grid cell (j^1, k^1) of the reduced grid is formed by fusing the grid cells (i^2, j^2, k^2) in block B^2 given by

$$\begin{aligned} i^2 &= di, & di &= 1 \dots M_i, \\ j^2 &= M_j(j^1 - 1) + dj, & dj &= 1 \dots M_j, \\ k^2 &= M_k(k^1 - 1) + dk, & dk &= 1 \dots M_k. \end{aligned} \quad (3.53)$$

On this reduced grid, cell-centred values of the flow state are obtained by averaging,

$$W_{j^1, k^1}^{(a)} = \frac{1}{M_i M_j M_k} \sum_{di=1}^{M_i} \sum_{dj=1}^{M_j} \sum_{dk=1}^{M_k} W_{i^2, j^2, k^2}^{(2)}. \quad (3.54)$$

Now, the dummy-cell values are computed by first determining a unique wall value at all cell faces $S_{1/2, j^1, k^1}^{(1)}$ of block B^1 on face F . This value is determined using the flow-state values of block B^1 and of the reduced grid in block B^2 , and depends on the grid properties of the basic grid. For non-smooth basic grids, this is simply a matter of applying the same method as for faces without local grid refinement. If the basic grid is C^1 -continuous, then the wall value is given by simple averaging,

$$W_{j^1, k^1}^w = \frac{1}{2}(W_{1, j^1, k^1}^{(1)} + W_{j^1, k^1}^{(a)}), \quad (3.55)$$

Note that a C^0 -continuous grid could also have been obtained without fusing grids in the i direction. However, if the basic grid is C^1 -continuous, this could lead to a large jump in normal mesh size across face F . Taking this jump in mesh size into account in determining the wall value W^w could lead to a form of downwinding (if the flow direction is from the large to the small mesh size), which is unstable.

Given the wall value at cell face $S_{1/2, j^1, k^1}^{(1)}$, linear extrapolation (consistent with requirement (3.48)) is used to compute the dummy-cell value at $(0, j^1, k^1)$ of block B^1 ,

$$W_{0, j^1, k^1}^{(1)} = 2W_{j^1, k^1}^w - W_{1, j^1, k^1}^{(1)}, \quad (3.56)$$

and similarly to compute the dummy-cell values at all cells $(0, j^2, k^2)$ (with j^2 and k^2 given by equation (3.53)) of block B^2 ,

$$W_{0,j^2,k^2}^{(2)} = 2W_{j^1,k^1}^w - W_{1,j^2,k^2}^{(1)}, \quad (3.57)$$

The convective fluxes for both the coarse block (B^1) and the fine block (B^2) are calculated using the corresponding dummy-cell values. Consider a coarse-block cell face $S_{1/2,j^1,k^1}^{(1)}$ at the face F and the corresponding fine-block cell faces $S_{1/2,j^2,k^2}^{(2)}$. At these cell faces, a unique wall value W_{j^1,k^1}^w has been defined. Furthermore, if the area vector of the coarse-block cell face is taken equal to the sum of the area vectors of the fine-block cell faces, then the convective flux across the coarse-block cell face will be equal to the sum of the fluxes across the fine-block cell faces,

$$F_{1/2,j^1,k^1}^{c(1)} = - \sum_{dj=0}^{M_j-1} \sum_{dk=0}^{M_k-1} F_{1/2,j^2,k^2}^{c(2)}, \quad (3.58)$$

with the indices (j^2, k^2) of the fine cells defined by equation (3.53). (The minus sign is due to the fact that i^1 and i^2 increase in opposite directions.) Thus, a conservative discretization is obtained.

The artificial diffusive flux at a cell face of the fine block is computed using values from the corresponding coarse cells in the coarse block instead of dummy-cell values. The artificial diffusive flux through a cell face of the coarse block is calculated explicitly by the sum of the fluxes through the corresponding fine-block cell faces,

$$F_{1/2,j^1,k^1}^{a(1)} = - \sum_{dj=0}^{M_j-1} \sum_{dk=0}^{M_k-1} F_{1/2,j^2,k^2}^{a(2)}, \quad (3.59)$$

with the indices (j^2, k^2) of the fine cells defined by equation (3.53).

3.6 EXTENSIONS NEEDED FOR THE NAVIER-STOKES EQUATIONS

3.6.1 Matrix artificial diffusion model

As explained in section 3.2, the basic scheme with Jameson-type scalar artificial diffusion must be improved along the lines of the matrix artificial-diffusion model of Swanson-Turkel [111, 112]. Comparing the matrix model to the scalar model, two effects can be distinguished: the matrix model captures shocks more sharply, and it reduces the grid dependency in boundary layers, e.g. [3, 92, 112]. For the extension of the existing Euler solver to the Navier-Stokes equations, we are only interested in the second effect. For industrial applications, a reduced grid dependency in boundary layers is important, since it is not always feasible (due to limited computer resources) or simple to define the very fine grids required by the scalar model. The first effect (sharper shocks) comes at a cost. A different, stronger shock sensor must be used, which may have side effects such as e.g. an increase of artificial diffusion in stagnation regions, while several coefficients of the artificial diffusion may have to be

retuned. Further, it results in a moderate loss of convergence speed and an increase of computation time (in the order of 15%) per iteration [112].

We will define a modified form of the matrix-diffusion model. As will be shown, it gives the same reduction of the grid dependency in boundary layers as the original matrix model, but it does not result in a loss of convergence speed, nor does it require a significant increase in computation time compared to the scalar model. Also with this modified matrix model, the basic, robust Jameson-type shock sensor can be retained.

First, the original matrix model of Swanson–Tukel will be described, and subsequently the modified form will be presented. The main idea of the matrix model is to scale the artificial diffusion by a matrix depending on the Jacobian matrix of the convective fluxes instead of scaling by the spectral radius of the Jacobian. In this way, for each characteristic equation, the artificial diffusion is scaled with the corresponding wave speed (the eigenvalues of the convective Jacobian) so that the appropriate amount of artificial diffusion is employed, instead of using the maximum required amount for all characteristic equations. Thus, the scheme has similarities with upwind schemes [111, 112].

Let (I, J, K) be a cyclic permutation of (i, j, k) . For all cell faces $S_{I,J,K+1/2}$, the artificial diffusive flux is given by

$$F_{I,J,K+1/2}^a = |\mathcal{A}^{(K)}|_{I,J,K+1/2} (f_{I,J,K+1/2}^{(2)} - f_{I,J,K+1/2}^{(4)}), \quad (3.60)$$

with $\mathcal{A}^{(K)}$ the Jacobian matrix of the convective flux in K direction (given in appendix A), with $|\mathcal{A}^{(K)}|$ the absolute-value Jacobian matrix (defined below), and with

$$\begin{aligned} f_{I,J,K+1/2}^{(2)} &= \epsilon_{I,J,K+1/2}^{(2)} (U_{I,J,K+1} - U_{I,J,K}), \\ f_{I,J,K+1/2}^{(4)} &= \epsilon_{I,J,K+1/2}^{(4)} (U_{I,J,K+2} - 3U_{I,J,K+1} + 3U_{I,J,K} - U_{I,J,K-1}). \end{aligned} \quad (3.61)$$

Note that now the total energy may not be replaced by the total enthalpy, as was done for the scalar model.

Let Λ be the diagonal matrix with the eigenvalues of $\mathcal{A}^{(K)}$ along its diagonal,

$$\begin{aligned} \Lambda &= \text{diag}(\lambda_1, \lambda_1, \lambda_1, \lambda_4, \lambda_5), \\ \lambda_1 &= \vec{u} \cdot \vec{A}^{(K)}, \\ \lambda_4 &= \vec{u} \cdot \vec{A}^{(K)} + c \|\vec{A}^{(K)}\|, \\ \lambda_5 &= \vec{u} \cdot \vec{A}^{(K)} - c \|\vec{A}^{(K)}\|, \end{aligned} \quad (3.62)$$

with c the speed of sound and with $\vec{A}^{(K)}$ the area vector in K direction. Let the matrix \mathcal{Q} have the right eigenvectors of $\mathcal{A}^{(K)}$ as its column vectors. Then, the absolute-value Jacobian matrix is defined by taking the absolute value of its eigenvalues,

$$|\mathcal{A}^{(K)}| = \mathcal{Q} |\Lambda| \mathcal{Q}^{-1}, \quad (3.63)$$

where the matrix $|\Lambda|$ is defined by taking the absolute value of the elements of Λ . The complete expression for $|\mathcal{A}^{(K)}|$ is given in appendix A.

In order to avoid that the eigenvalues may become zero (which would lead to zero artificial diffusion for some characteristic equations and therefore could lead to unphysical solutions as well as to odd-even decoupling) a lower bound is defined using the spectral radius,

$$\lambda_{\max} = |\vec{u} \cdot \vec{A}^{(K)}| + c \|\vec{A}^{(K)}\|, \quad (3.64)$$

so that the diagonal elements of $|\Lambda|$ become

$$\begin{aligned} \tilde{\lambda}_1 &= \max\{|\lambda_1|, \varepsilon^L \lambda_{\max}\}, \\ \tilde{\lambda}_4 &= \max\{|\lambda_4|, \varepsilon^N \lambda_{\max}\}, \\ \tilde{\lambda}_5 &= \max\{|\lambda_5|, \varepsilon^N \lambda_{\max}\}. \end{aligned} \quad (3.65)$$

Swanson and Turkel [111] at first chose ε^L and ε^N to be equal to 0.25. In order to obtain sharp oscillation-free shocks, as stated before, they also needed to modify the shock sensor of equation (3.22), replacing it by a stronger shock sensor given by

$$\nu_{I,J,K} = \frac{|p_{I,J,K+1} - 2p_{I,J,K} + p_{I,J,K-1}|}{\delta D_{\text{TVD}} + (1 - \delta) D_{\text{old}}}, \quad (3.66)$$

with

$$\begin{aligned} D_{\text{old}} &= p_{I,J,K+1} + 2p_{I,J,K} + p_{I,J,K-1}, \\ D_{\text{TVD}} &= |p_{I,J,K+1} - p_{I,J,K}| + |p_{I,J,K} - p_{I,J,K-1}|, \end{aligned} \quad (3.67)$$

and with $\delta = 0.5$.

However, to improve the accuracy in boundary layers, Swanson and Turkel found it necessary to take ε^L equal to 0.01 for the boundary-layer normal direction [112]. In this way, the amount of artificial diffusion in the normal direction is strongly reduced for the entropy and shear waves, so that no interference occurs with the physical diffusion. Sufficient damping of the high-frequency modes in normal direction is provided by the physical diffusion. In practice, convergence is not affected with this choice for ε^L , as will be shown in chapter 6.

We define a reduced form of the matrix diffusion model, with as main purpose the reduction of the artificial diffusion for the normal direction in boundary layers and wakes. For the tangential directions, the two factors ε^L and ε^N are set to one, or, similarly, the scalar diffusion model is applied for these directions (which is computationally more efficient). For the normal direction the two factors are set as

$$\begin{aligned} \varepsilon^L &= 0.01, \\ \varepsilon^N &= 1. \end{aligned} \quad (3.68)$$

Taking ε^N equal to one, implies that $\tilde{\lambda}_4$ and $\tilde{\lambda}_5$ are equal to the spectral radius λ_{\max} , so that the standard Jameson shock capturing scheme can be used. Thus, the factors $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are defined in the same way as in the scalar model (equations (3.21) and (3.24)).

The modified matrix model is less expensive than the original model for two reasons: it is only applied in the boundary-layer normal direction, and the expression

for the scaling matrix $|\mathcal{A}^{(K)}|$ (given fully in appendix A) is simplified due to taking $\tilde{\lambda}_4 = \tilde{\lambda}_5$. This simplified form of the scaling matrix is given by

$$|\mathcal{A}^{(K)}|_{I,J,K+1/2} = e_0 \mathcal{J} + e_1 A C^T + e_2 B D^T, \quad (3.69)$$

with \mathcal{J} the unit 5×5 matrix, and with the coefficients e_i given by

$$\begin{aligned} e_0 &= (\tilde{\lambda}_1)_{I,J,K+1/2}, \\ e_1 &= \frac{\gamma - 1}{c_{I,J,K+1/2}^2} (\tilde{\lambda}_{4,5} - \tilde{\lambda}_1)_{I,J,K+1/2}, \\ e_2 &= -\frac{(\tilde{\lambda}_{4,5} - \tilde{\lambda}_1)_{I,J,K+1/2}}{\|\vec{A}^{(K)}\|_{I,J,K+1/2}^2}, \end{aligned} \quad (3.70)$$

where the scaling factors $\tilde{\lambda}_1$ and $\tilde{\lambda}_{4,5} = \lambda_{\max}$ at cell faces are obtained by averaging cell-centre values. The column vectors A , B , C , and D are given by

$$\begin{aligned} A &= \begin{pmatrix} 1 \\ \vec{u} \\ H \end{pmatrix}_{I,J,K+1/2}, & B &= \begin{pmatrix} 0 \\ \vec{A}^{(K)} \\ \vec{u} \cdot \vec{A}^{(K)} \end{pmatrix}_{I,J,K+1/2}, \\ C &= \begin{pmatrix} \frac{1}{2} \|\vec{u}\|^2 \\ -\vec{u} \\ 1 \end{pmatrix}_{I,J,K+1/2}, & D &= \begin{pmatrix} \vec{u} \cdot \vec{A}^{(K)} \\ -\vec{A}^{(K)} \\ 0 \end{pmatrix}_{I,J,K+1/2}, \end{aligned} \quad (3.71)$$

and the velocity vector, the total enthalpy, and the speed of sound at the cell face are defined by

$$\begin{aligned} \vec{u}_{I,J,K+1/2} &= \frac{(\rho \vec{u})_{I,J,K+1} + (\rho \vec{u})_{I,J,K}}{\rho_{I,J,K+1} + \rho_{I,J,K}}, \\ H_{I,J,K+1/2} &= \frac{(\rho H)_{I,J,K+1} + (\rho H)_{I,J,K}}{\rho_{I,J,K+1} + \rho_{I,J,K}}, \\ c_{I,J,K+1/2}^2 &= (\gamma - 1) \left(H_{I,J,K+1/2} - \frac{1}{2} \|\vec{u}_{I,J,K+1/2}\|^2 \right). \end{aligned} \quad (3.72)$$

The product of the absolute value of the Jacobian with the column vector $E = f^{(2)} - f^{(4)}$ can be efficiently computed by

$$|\mathcal{A}^{(K)}|E = e_0 E + e_1 (C \cdot E) A + e_2 (D \cdot E) B, \quad (3.73)$$

without fully computing the Jacobian itself.

3.6.2 High-aspect-ratio scaling of the artificial diffusion

In grids used for Navier–Stokes computations, very large aspect ratios of the grid cells can appear near solid surfaces. If the same scaling of the artificial diffusion is used as for the Euler equations, the artificial diffusion in tangential directions along

the solid surface will be insufficient to obtain strong damping of high-frequency modes in these directions. This will result in a poor convergence speed, in particular for the multi-grid scheme discussed in chapter 4 (see also section 4.4.2).

Consider a scaling factor λ of either the scalar or the matrix artificial diffusion model. This scaling factor determines the strength of the damping of high-frequency modes by the artificial diffusion (see e.g. appendix B). The scaling factor for the fluxes in K direction (as defined by equation (3.25) for the scalar model) is proportional to the area of the cell face normal to the K direction, or equivalently, it is proportional to the cell volume divided by the mesh size in K direction. Thus, if a cell has a large aspect ratio, the scaling factor, and therefore the damping of high-frequency modes, will be small in the direction of the largest mesh size.

In order to obtain an adequate amount of artificial diffusion in the direction of the largest mesh size, the scaling is modified following Martinelli [73]:

$$\begin{aligned}\tilde{\lambda}_{I,J,K} &= v_{I,J,K}^{(K)} \lambda_{I,J,K}, \\ v_{I,J,K}^{(K)} &= \max \left\{ 1, \alpha^{(K)} + \alpha^{(I)} \left(r_{I,J,K}^{(IK)} \right)^{k^{(h)}} + \alpha^{(J)} \left(r_{I,J,K}^{(JK)} \right)^{k^{(h)}} \right\},\end{aligned}\quad (3.74)$$

in which $r^{(IK)}$ is a measure of the cell aspect ratio:

$$r_{I,J,K}^{(IK)} = \frac{\left\| \tilde{A}_{I+1/2,J,K}^{(I)} + \tilde{A}_{I-1/2,J,K}^{(I)} \right\|}{\left\| \tilde{A}_{I,J,K+1/2}^{(K)} + \tilde{A}_{I,J,K-1/2}^{(K)} \right\|}. \quad (3.75)$$

The coefficients $\alpha^{(K)}$ are chosen such that for cells with an aspect ratio of one the scaling has no effect:

$$\alpha^{(K)} = \begin{cases} \frac{1}{3}, & \text{3D calculations,} \\ \frac{1}{2}, & \text{2D calculations in } (I, K) \text{ or } (J, K) \text{ plane,} \\ 0, & \text{2D calculations in } (I, J) \text{ plane.} \end{cases} \quad (3.76)$$

The parameter $k^{(h)}$ determines the extent to which the high-aspect-ratio scaling is applied. With the value $k^{(h)} = 0$, no scaling is applied. With the value $k^{(h)} = 1$, the effect of multiplying the scaling factor λ with $v^{(K)}$ is to replace the cell-face area of the K direction in the expression for the scaling factor with the average of the cell-face areas in I , J , and K directions. Thus, the scaling of the artificial diffusive fluxes, and therefore also the damping of high-frequency modes, is of comparable magnitude in the different computational directions.

Increasing the artificial diffusion strongly by setting $k^{(h)} = 1$, however, will also strongly reduce the numerical accuracy. A fair compromise between numerical accuracy and convergence speed is given by the values $k^{(h)} = 1/2$ for 3D calculations, and $k^{(h)} = 2/3$ for 2D calculations, as employed by many authors [73, 109, 114]. (For 3D calculations, generally larger cell aspect ratios will be present, e.g. due to large mesh sizes in the span direction on a wing, and therefore a smaller value of $k^{(h)}$ is needed to avoid too strong a reduction of the numerical accuracy.)

3.7 RESULTS

In this section we will verify that the space discretization gives results as expected. In particular, the grid dependency of the numerical solution obtained with the scalar and (modified) matrix artificial-diffusion models will be evaluated.

Two test cases will be considered: the laminar flow over a flat plate, and the transonic, turbulent flow around the RAE2822 airfoil. The first case was used by Swanson and Turkel to show the reduced grid dependency of their matrix model, and thus this case is used to show that the modified matrix model gives a similar result. The second case is a two-dimensional flow that is representative of the intended application range of the flow solver.

Consider the laminar flow over a flat plate of length L at a Mach number $M_\infty = 0.2$ and a Reynolds number $Re_\infty = 5 \cdot 10^3$. Let x be the coordinate along and y the coordinate normal to the flat plate, with corresponding velocity components u and v . Applying boundary-layer theory, a similarity solution may be derived (for the incompressible equations) where u/u_∞ is a function of $\eta = Re_x^{1/2}y/x$ with the local Reynolds number $Re_x = \rho u_\infty x / \mu$ (see e.g. Batchelor [7]). This so-called Blasius solution is given in figure 3.5 together with the numerical solutions at $x = 0.5L$ using the scalar model and the modified matrix model. A range of grids has been used,

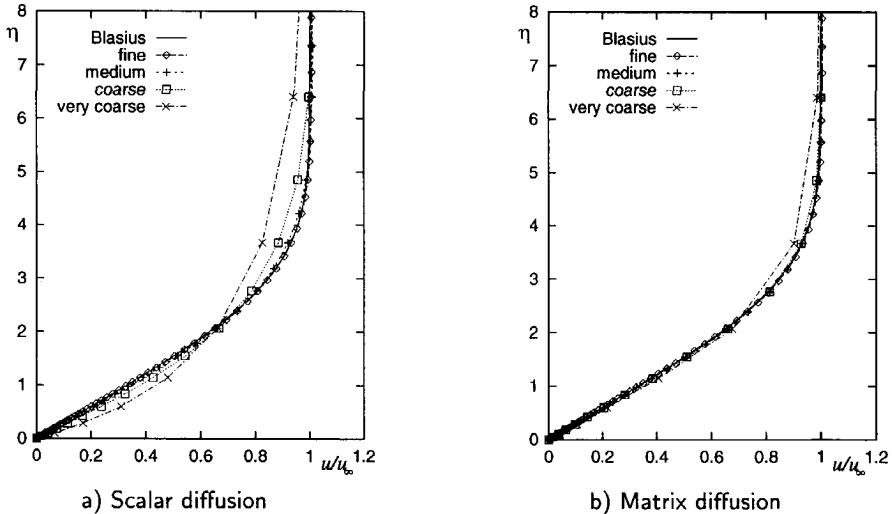


Figure 3.5: Velocity distribution at $x = 0.5L$ for the laminar flow over a flat plate at different grid levels ($M_\infty = 0.2$, $Re_\infty = 5 \cdot 10^3$, fine grid: 56 points in boundary layer)

with the finest grid having approximately 56 grid points in the boundary layer (at $x = 0.5L$). As can be seen, for the scalar model, only the finest-grid solution is indistinguishable from the Blasius solution, while for the matrix model, all but the coarsest-grid solution (with only 7 points in the boundary layer) lie on the Blasius solution. This weak grid dependency for the modified matrix model is consistent with the solutions of the original matrix model of Swanson and Turkel [112]. The

reduced grid dependency of the matrix model can be seen perhaps even better for the skin-friction coefficient $C_f = \tau_w / (\frac{1}{2} \rho_\infty u_\infty^2)$ (with τ_w the wall shear stress) given in figure 3.6, where none of the solutions with the matrix model on the different grids can be distinguished from the Blasius solution.

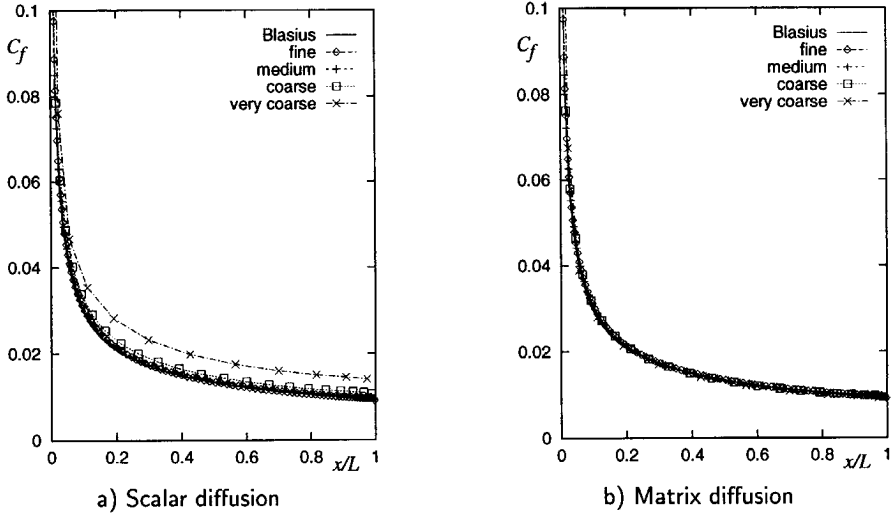
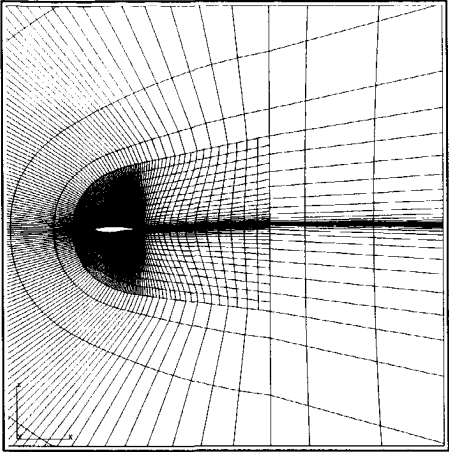


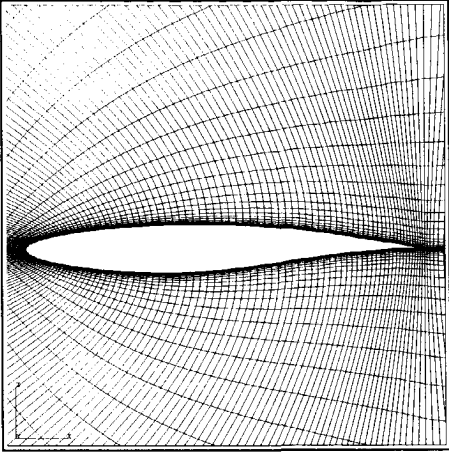
Figure 3.6: Skin-friction distribution for the laminar flow over a flat plate at different grid levels ($M_\infty = 0.2$, $Re_\infty = 5 \cdot 10^3$, fine grid: 56 points in boundary layer at $x = 0.5L$)

As a second test case, we consider the turbulent, transonic flow around the RAE2822 airfoil with as flow conditions $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$, and with transition at 3% of the chord from the leading edge (so-called case 9 of [26]). An 8-block C-type grid is employed, with 528×96 grid cells (circumferential \times normal) of which an impression is given in figure 3.7 (medium grid). Note that the four outer blocks have been coarsened by a factor two. In figure 3.8, contour lines of constant Mach number are shown of the numerical solutions (with scalar and matrix artificial diffusion) on the fine grid. (The Baldwin-Lomax model has been used to compute the eddy viscosity, see chapter 5.) Clearly visible are the boundary layer on the upper and lower sides of the airfoil and the shock wave on the upper side. Upon close inspection, it can be seen that with the scalar model overshoots appear in the Mach-number distribution at the edge of the boundary layers (in particular visible on the lower side), which are not present in the solution obtained with the matrix model.

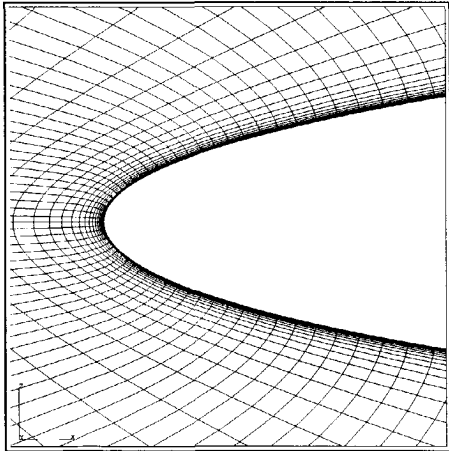
In figure 3.9, a detailed comparison is made of the velocity profiles at $x = 0.4c$ on the upper side (with c the chord length) for three grid levels (medium and coarse grids having been obtained by doubling the mesh size). The tangential velocity component (u) and the normal wall distance (y) have been made dimensionless according to the law-of-the-wall scaling: $u^+ = u/u_\tau$ and $y^+ = \rho u_\tau y / \mu$ with $u_\tau = (\tau_w / \rho_w)^{1/2}$ the friction velocity. Further, the theoretical solutions for the viscous sublayer ($u^+ = y^+$,



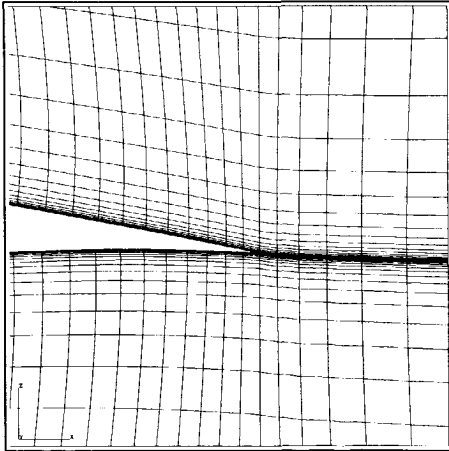
a) Local grid refinement



b) Near airfoil



c) Near leading edge



d) Near trailing edge

Figure 3.7: Medium (264×48) 8-block grid around RAE2822 airfoil

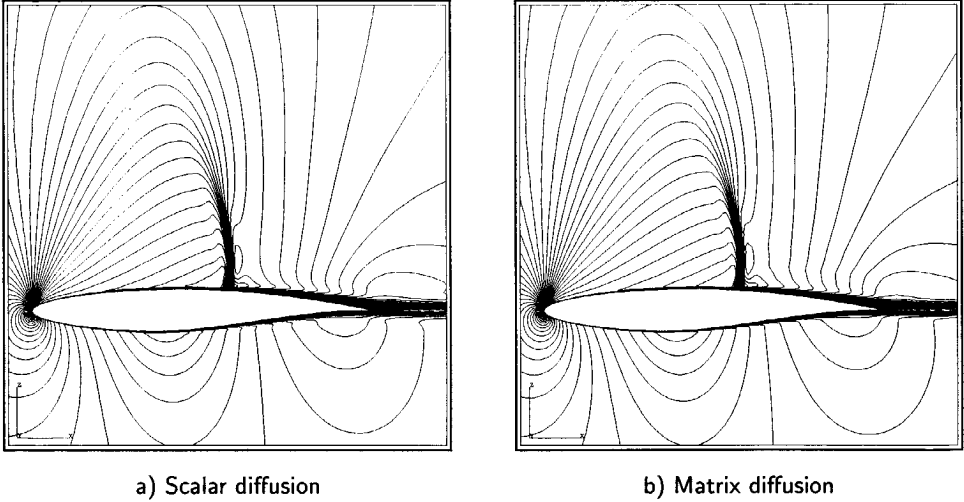


Figure 3.8: Lines of constant Mach number for RAE2822 case 9 ($M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$, 8-block C-type 528×96 grid)

valid for $y^+ < 5$) and for the log-layer ($u^+ = \ln(y^+)/\kappa + C$ with $\kappa = 0.40$ and $C = 5$, valid for $y^+ > 30$) have been included as reference, see e.g. [84, 98]. With the matrix model, the grid dependency is less than with the scalar model, although the difference is not as strong as for the flat plate. However, the reduced grid dependency with the matrix model becomes better visible if integral quantities are considered: in particular the skin-friction coefficient, figure 3.10, and the displacement thickness δ^* , figure 3.11. On the lower side, the improved numerical accuracy is clearly visible for both quantities. On the upper side, the results are obscured by the smearing of the shock wave on the medium and coarse grid (revealing the dependency on the grid resolution in tangential rather than in normal direction).

Finally, also the pressure coefficient $C_p = (p - p_\infty)/(\frac{1}{2}\rho_\infty u_\infty^2)$ is given in figure 3.12, together with the experimental results of [26] as a reference. Although less pronounced, also here the reduced grid dependency can be seen for the matrix model, in particular on the upper side upstream of the shock, and on the aft part of the lower side.

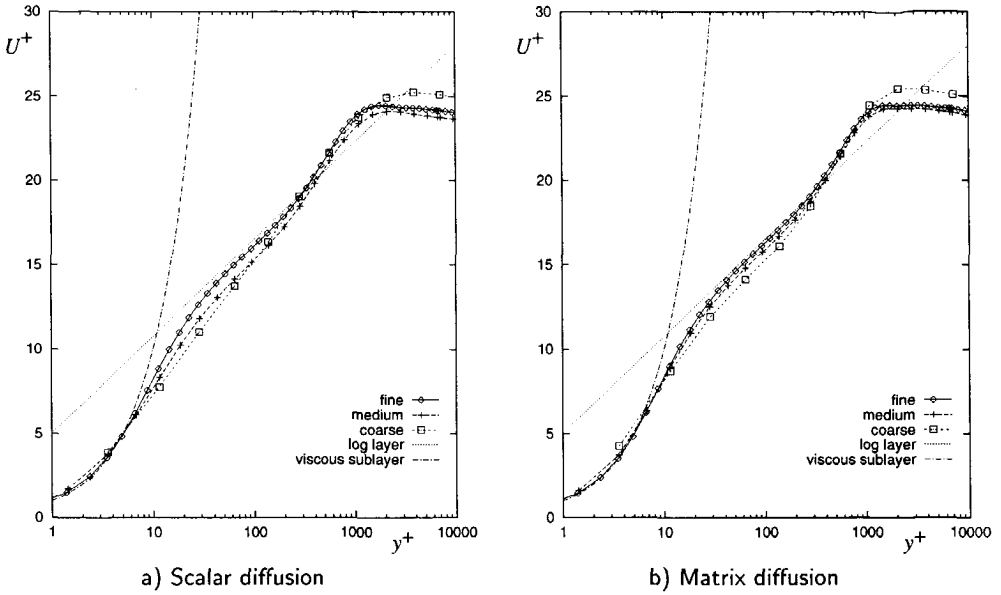


Figure 3.9: Velocity profile in law-of-the-wall coordinates on upper side at $x/c = 0.4$ for RAE2822 case 9 ($M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) at different grid levels (fine: 8-block C-type 528×96 grid)

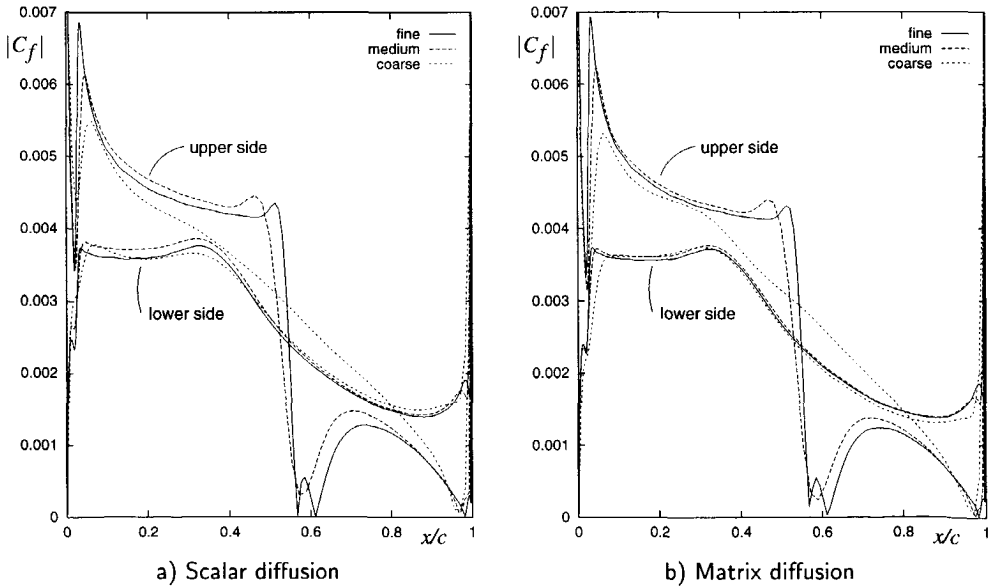


Figure 3.10: Skin-friction distribution for RAE2822 case 9 ($M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) at different grid levels (fine: 8-block C-type 528×96 grid)

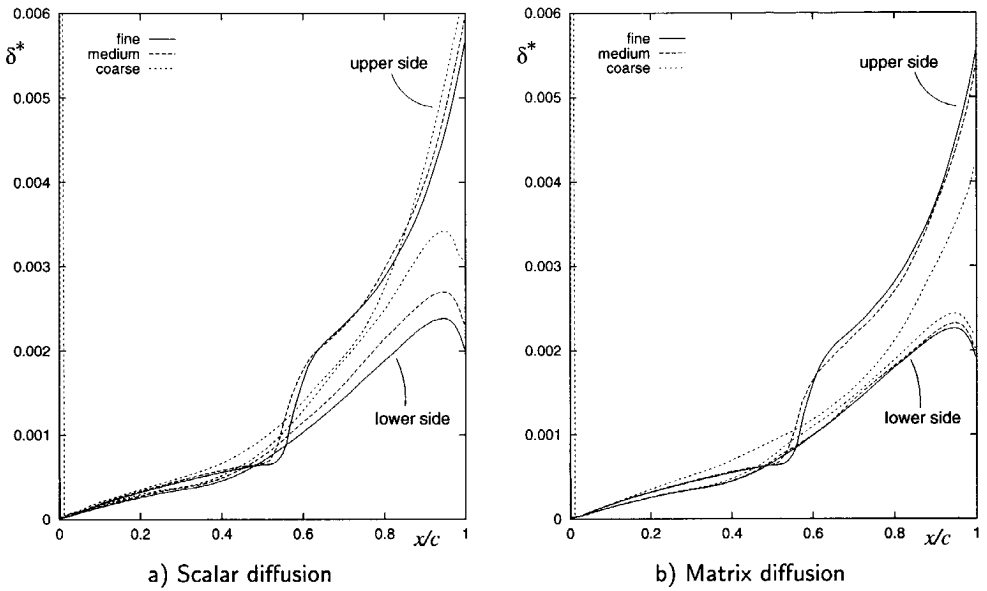


Figure 3.11: Displacement thickness for RAE2822 case 9 ($M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) at different grid levels (fine: 8-block C-type 528×96 grid)

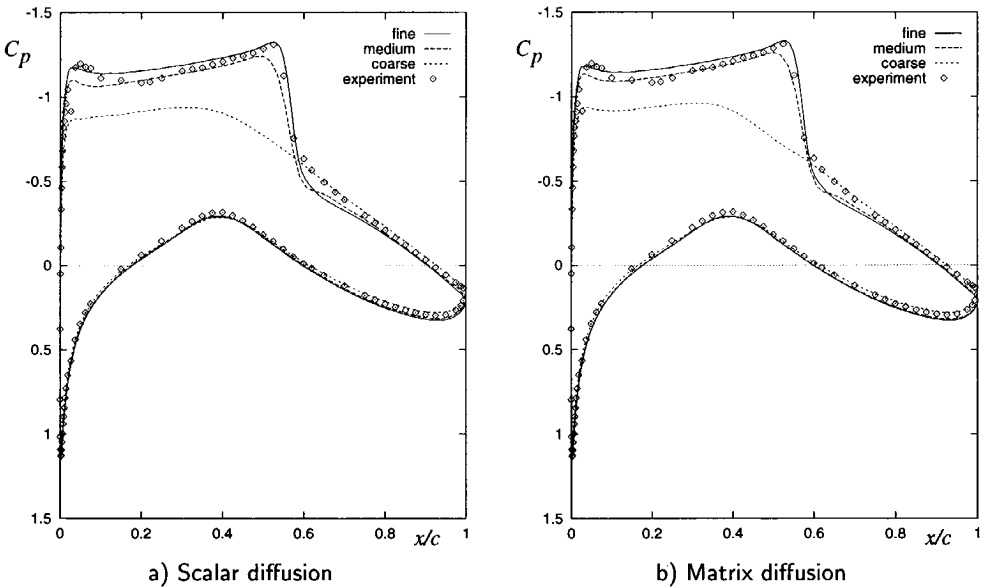


Figure 3.12: Surface pressure coefficient for RAE2822 case 9 ($M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) at different grid levels (fine: 8-block C-type 528×96 grid)

SOLUTION PROCEDURE

4.1 INTRODUCTION

In the previous chapter, the TLNS equations have been discretized in space as a first step in deriving a numerical method for approximating the continuous equations presented in chapter 2. For the steady-flow equations, this space discretization has resulted in a set of non-linear, algebraic equations. In this chapter, the procedure by which these equations are solved will be discussed. The solution procedure will be based on the time-integration idea: the unsteady equations, which have been discretized in space only (semi-discretized) and form a set of ordinary differential equations, are integrated in time until a steady solution is obtained.

As for the space discretization, the starting point for developing a solution procedure for the TLNS equations is formed by the solution procedure used for the Euler equations. This solution procedure consists of explicit time integration using Runge-Kutta schemes, accelerated by local time stepping, implicit residual averaging, and enthalpy damping. In section 4.2, a literature review will be presented, and the possibilities for extending this solution procedure to the TLNS equations will be considered. A direct extension of the method for the Euler equations to the TLNS equations is described in section 4.3 resulting in the so-called basic scheme. However, it is well-known that the convergence speed of this basic scheme is insufficient (in the light of the requirement formulated in chapter 1). Therefore, the method is further extended in section 4.4 with a multi-grid scheme in combination with a variable-coefficient implicit residual averaging scheme. In particular, multi-block aspects of these schemes will be considered. Also, the scaling of the artificial diffusion for high-aspect-ratio grid cells, presented in the previous chapter, will be discussed.

4.2 LITERATURE REVIEW

There are many different solution procedures for solving the type of non-linear, discrete equations considered here. A first major distinction can be made between schemes which are based on the time-integration concept, and schemes which are not, such as Newton-iteration schemes (e.g. [37, 103]). Time-integration schemes may be divided into explicit and implicit schemes, where implicit schemes sometimes reduce to a Newton-iteration scheme for infinite time steps. For the compressible Euler and Navier-Stokes equations, both explicit and implicit schemes are frequently used in the literature (the choice between strategies perhaps often depending on personal preferences).

In traditional explicit schemes (e.g. Lax–Wendroff [67]), the discretization of time and space derivatives is interdependent, so that steady-state solutions depend on the time step. To avoid this problem, semi-discretization methods, with separate space discretization and time integration as considered here, were developed. In particular, an efficient explicit scheme for the Euler equations was developed by Jameson et al. [52], based on Runge–Kutta type schemes. This efficiency was obtained by accelerating the time integration by local time stepping, implicit residual averaging, and enthalpy damping.

Implicit time discretization of the unsteady Euler or Navier–Stokes equations results in a set of non-linear discrete equations to be solved at each time step. Newton-linearization is generally applied to reduce these equations to a set of linear equations. Solving this set of equations directly is too expensive (both in computation time and memory requirement), particularly in three dimensions, and therefore either approximate factorization methods, such as the alternating direction implicit (ADI) scheme [8, 88] and approximate LU-factorization [85], or iterative methods [21], such as Gauss–Seidel iterations, are used.

An important method for improving the convergence speed of both explicit and implicit schemes is the so-called multi-grid scheme, first developed by Fedorenko [34], and made popular by Brandt [16]. For explicit schemes, multi-grid methods have been developed e.g. by Ni [83] and Jameson [51] for the Euler equations, and by Martinelli [73], Vatsa–Wedan [114] and Swanson–Radespiel [109] for the Navier–Stokes equations. For implicit schemes, multi-grid methods have been developed e.g. by Caughey [19] and Spekrijse [103] for the Euler equations, and by Yoon [125] for the Navier–Stokes equations. Multi-grid theory (in particular for elliptic equations) indicates that convergence speeds independent from the number of grid points are obtainable. It is then important that the integration scheme, used as relaxation operator in the multi-grid method, strongly damps high-frequency modes.

The direct extension of the Jameson-type explicit time-integration scheme to the Navier–Stokes equations does not give a satisfactory convergence speed, even when employing a multi-grid acceleration [110]. This is mainly caused by the severe limit on the time steps as a consequence of the small mesh sizes needed to represent the boundary layers; typically, for a Reynolds number of $1 \cdot 10^6$, the height of the first grid cell on the wing surface must be of the order of 10^{-5} chord lengths. Furthermore, no such resolution is required in the directions along the wing surface, resulting in mesh sizes in these directions that are several orders larger; typically, cell aspect ratios of the order of 10^3 to 10^4 are obtained. As a result, high frequency modes in the tangential direction are hardly damped by one explicit time step, so that the explicit scheme is not suitable as relaxation operator in a multi-grid scheme.

Several approaches may be considered for improving the convergence speeds for the Navier–Stokes equations, starting from the Jameson-type explicit scheme. These approaches are in particular aimed at dealing with grid cells with high aspect ratios.

- The Jameson-type scheme, including multi-grid, was first successfully extended to the Navier–Stokes equations in two dimensions by Martinelli [73]. Both in the implicit residual averaging and in the artificial diffusion, coefficients dependent on the aspect ratios were introduced. The main effect was an improved damping

of high frequencies in directions along configuration surfaces as a consequence of increased artificial diffusion.

- In the semi-implicit scheme [72], the explicit treatment of the computational directions along a configuration surface is maintained, but the computational direction normal to the surface is treated implicitly. Thus, a strong increase of the time step is obtained, without the expense of going fully implicit, and as a consequence the damping of high frequencies in tangential direction is strengthened.
- The semi-coarsening multi-grid scheme [82, 90] also appears to be suitable for high-aspect-ratio grid cells. On each multi-grid level several coarse grids are considered, varying the directions in which the grids have been coarsened.

The Jameson-type scheme with multi-grid and with the improvements of Martinelli has been successfully applied in many three-dimensional flow computations [45, 50, 62, 63, 97, 114]. Although the ideal multi-grid convergence is not attained with this approach, computation speeds as required in chapter 1 are attainable. The increase of the artificial diffusion may reduce the numerical accuracy to some extent, for example at trailing edges (see e.g. [109]). For the other two approaches, although possibly having more potential, there is less evidence yet of successful applications. Therefore, we adopt a scheme along the lines of Jameson and Martinelli, thus relying on well-established methods.

4.3 BASIC SOLUTION PROCEDURE

The basic solution procedure, described in this section, consists of integrating the unsteady equations in time until a steady solution is obtained, using a Runge-Kutta scheme accelerated by local time stepping and implicit residual averaging. This scheme was already employed in the Euler solver and is here directly extended to the TLNS equations by including the viscous time-step limit in the local time step. Enthalpy damping, a procedure used in the Euler solver to improve the convergence rate further, is not considered here since it is based on the assumption that the enthalpy is constant, which is in general not true for viscous flows.

4.3.1 Runge-Kutta time integration

Consider the unsteady TLNS (or Euler) equations, discretized in space as described in chapter 3. These semi-discretized equations form a set of ordinary differential equations, given by

$$\frac{dU}{dt} + R(U) = 0, \quad (4.1)$$

which holds for all grid cells in all blocks (i.e. equation (3.3), dropping the subscripts i , j , and k for simplicity). The residual R in a grid cell depends on the flow-state vector in neighbouring cells, which may be dummy cells. This residual is computed following the discrete mathematical model as described in the previous chapter, including the boundary conditions which determine the dummy-cell values.

Following Jameson [48], a general class of Runge–Kutta (RK) schemes, with optionally frozen diffusive terms, is used to integrate the set of semi-discretized equations in time. For time-accurate computations, a RK scheme may be chosen with a certain required order of accuracy in time. When the time integration is only used as a means for finding a steady solution, as is the case here, a RK scheme may be chosen so as to obtain fast convergence (or to reduce the memory requirement). The time integration is started from a uniform flow, with all flow variables set to their free-stream values.

One time step for a general m -stage RK scheme, going from time level n to level $n + 1$, is sketched in figure 4.1, where the flow state at time level n is indicated by a superscript n . In this RK scheme, the intermediate flow states ($U^{(q)}$) do not need to be stored, thus saving memory compared to traditional RK schemes. Note that the calculation of the flow states at dummy cells, the calculation of the time step, and the application of residual averaging have been included, which will be discussed further on.

1. Save the present flow state at time level n ,

$$U^{(0)} = U^n.$$

2. Calculate the time step Δt .
3. Calculate the dummy-cell values of $U^{(0)}$ at internal faces.
4. For stage $q = 1$ to m do:

- (a) Calculate the dummy-cell values of $U^{(q-1)}$ at external faces.

- (b) Calculate the flow correction,

$$\Delta U = \alpha_q \Delta t R^{(q-1)}.$$

- (c) Optionally apply implicit residual averaging to the flow correction ΔU .

- (d) Determine the flow state at the intermediate stage q ,

$$U^{(q)} = U^{(0)} - \Delta U.$$

5. Determine the new flow state at time level $n + 1$,

$$U^{n+1} = U^{(m)}.$$

Figure 4.1: One Runge–Kutta time step

The residual R consists of the summation of the different flux balances, given by equation (3.9). For a certain Runge–Kutta stage the residual may be calculated using flux balances of solutions of all previous stages. Here, the convective flux balance is

always computed using the most recent solution, while the diffusive flux balances can be frozen for a certain number of stages. Thus, the residual at stage q is given by

$$R_{i,j,k}^{(q)} = \left(D^c(U^{(q)})_{i,j,k} - D^d(U^{(l_q)})_{i,j,k} - D^a(U^{(l_q)})_{i,j,k} \right) / V_{i,j,k}, \quad (4.2)$$

with l_q the most recent stage before stage q at which the diffusive terms have been computed.

A specific RK scheme is now defined by the number of stages m and the coefficients α_q and l_q . For the Euler equations, a popular scheme has been a four-stage scheme (RK4) with one evaluation of the diffusive terms,

$$\begin{array}{llll} \alpha_1 = 1/4, & \alpha_2 = 1/3, & \alpha_3 = 1/2, & \alpha_4 = 1, \\ l_0 = 0, & l_1 = 0, & l_2 = 0, & l_3 = 0, \end{array} \quad (4.3)$$

(This scheme has a Courant number, defined in the subsequent section, of $C_{FL} = 2.6$.) For the TLNS equations, this scheme is less suitable due to the single evaluation of the diffusive terms resulting in a small diffusive time-step limit. A suitable scheme for the TLNS equations (as well as for the Euler equations) is defined in section 4.4.1 in relation to the multi-grid scheme.

A relevant aspect is how the Runge–Kutta scheme is applied on the multi-block grid. A complete time step is performed for each block successively, thus introducing a time lag between the blocks. An alternative is to update all the blocks at each stage of a time step, which would avoid any time lag and therefore would seem to be the safest approach from the point of view of stability and convergence speed. This requires significantly more memory (e.g. the time step and the residuals must be stored for all blocks, instead of for one block at a time), while in practice it has no significant advantage in stability or convergence speed [36, 96]. The loop over the blocks is done in a Gauss–Seidel manner. This means that when updating one block, the most recent solution in the adjacent blocks is used for computing the dummy-cell values at internal faces (faces between blocks). These dummy-cell values only need to be computed at the beginning of a time step, as indicated in figure 4.1, since during a complete Runge–Kutta time step in a block the solution in the other blocks is kept frozen. The dummy-cell values at external faces (solid surface, far field, etc.) are evaluated at each stage.

4.3.2 Local time stepping

The Runge–Kutta scheme is accelerated by evaluating the time step locally, i.e. for each grid cell separately [52]. In this way, larger time steps can be used where possible, thus expelling disturbances faster. Since the semi-discretization approach has been followed, the local evaluation of the time step does not affect the steady-state solution.

The time step, which is constant for all stages of one Runge–Kutta time step in any grid cell, must be chosen such that the scheme is still stable. Two time-step limits can be defined for each grid cell, one related to the convective terms and the other related to the diffusive terms. Relations for these limits can be found by stability considerations. The derivation of these relations is presented in appendix B.

The convective time-step limit Δt_c is evaluated by the following relation:

$$(\Delta t_c)_{i,j,k} = \frac{C_{\text{FL}} V_{i,j,k}}{\lambda_{i,j,k}^{(i)} + \lambda_{i,j,k}^{(j)} + \lambda_{i,j,k}^{(k)}}, \quad (4.4)$$

with C_{FL} the Courant (or Courant–Friedrichs–Lewy) number. The maximum eigenvalues $(\lambda^{(i)}, \lambda^{(j)}, \lambda^{(k)})$ of the convective Jacobians in i , j , and k directions (multiplied by the area vectors) are given by equation (3.25).

The diffusive time-step limit Δt_v is evaluated by

$$(\Delta t_v)_{i,j,k} = \frac{R_K V_{i,j,k}^2}{4 \left(\|\vec{A}_{i,j,k}^{(i)}\|^2 + \|\vec{A}_{i,j,k}^{(j)}\|^2 + \|\vec{A}_{i,j,k}^{(k)}\|^2 \right) \lambda_{i,j,k}^v}, \quad (4.5)$$

with R_K a stability limit similar to the Courant number (see appendix B). The area vector for each computational direction $(\vec{A}^{(i)}, \vec{A}^{(j)}, \text{ and } \vec{A}^{(k)})$, respectively) is only included (and given by equation (3.26)) if the relevant direction corresponds to a thin-layer normal direction. The maximum eigenvalue λ^v of the diffusive Jacobian is given by

$$\lambda_{i,j,k}^v = \max \left\{ \frac{1}{c_v} \frac{(\kappa_h + \kappa_t)_{i,j,k}}{\rho_{i,j,k}}, \frac{4}{3} \frac{(\mu + \mu_t)_{i,j,k}}{\rho_{i,j,k}} \right\}. \quad (4.6)$$

Instead of taking the minimum of the two time-step limits, a conservative approach is followed [73] to determine the final time step,

$$\Delta t_{i,j,k} = \frac{1}{1/\Delta t_c + 1/\Delta t_v}. \quad (4.7)$$

For this approach, it is not necessary to include a third time-step limit related to the artificial diffusion, if a suitable value of the artificial-diffusion coefficient $k^{(4)}$ is chosen, as shown in appendix B.

Values for the constants C_{FL} (Courant number) and R_K depend on the particular integration scheme employed (type of Runge–Kutta scheme, application of residual averaging), and are given in section 4.4.1.

4.3.3 Implicit residual averaging

A second method for accelerating the Runge–Kutta scheme is to apply implicit residual averaging [48]. With this method, stability for a model equation can be obtained for any Courant number C_{FL} (and any diffusive stability limit R_K) as long as a sufficient amount of averaging is employed [110]. For explicit time-integration schemes, the convergence speed is generally determined by (slow converging) low-frequency modes, while the time-step limit is determined by (fast converging) high-frequency modes. The idea of residual averaging is to smooth the high-frequency components present in the flow correction, thus allowing the use of a larger time step, which results in an increased convergence rate of the low-frequency modes. At the same time, however, the damping of high-frequency modes in the flow solution is reduced. (In the extreme case that the high-frequency modes are completely filtered out of

the flow correction, then the high-frequency modes in the flow solution will not be damped at all.) See appendix B for more details. Thus, residual averaging will at first improve the convergence speed, but applying it too strongly will on the other hand reduce the convergence speed. In practice, the optimal convergence rate is obtained for a Courant number of about two times the Courant number of the scheme without residual averaging.

Consider a single-block grid. At a certain stage q the flow correction (or residual) $(\Delta U)_{i,j,k}$ (as computed in step (4b) of the RK scheme) is replaced by a smoothed residual $S_{i,j,k}$. The latter is computed as solution of the following difference equation:

$$\left((1 - \epsilon^{(i)} \delta_i^2)(1 - \epsilon^{(j)} \delta_j^2)(1 - \epsilon^{(k)} \delta_k^2) S \right)_{i,j,k} = (\Delta U)_{i,j,k}, \quad (4.8)$$

for $i = 1 \dots N_i$, $j = 1 \dots N_j$, and $k = 1 \dots N_k$, with δ_i^2 the second-order difference operator in i direction, $(\delta_i^2 f)_{i,j,k} = f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k}$. (Note that $(\Delta U)_{i,j,k}$ and $S_{i,j,k}$ are both vectors with 5 components.) At the block boundaries, we follow a conservative approach by applying the Dirichlet boundary condition

$$S_{i,j,k} = 0, \quad (4.9)$$

at all dummy cells. This boundary condition will tend to reduce the flow correction at block faces, which we expect to be more robust than applying a Neumann condition.

For a multi-block grid, a Runge-Kutta time step is performed for each block successively. At each stage (or optionally at alternate stages) of a RK time step, we apply the residual-averaging operator for the considered block only. One could in theory apply the residual averaging operator over the complete domain, but as for the RK scheme, this would result in a significant increase of required memory, as well as an increase of computational cost. Further, in practice it is found that applying the averaging operator per block does not reduce the convergence speed, as long as the block dimensions are not too small.

In the basic scheme, the coefficients $\epsilon^{(i)}$, $\epsilon^{(j)}$, and $\epsilon^{(k)}$ are taken equal to a constant value ϵ . (In section 4.4.2, it will be seen that variable coefficients are more suitable when dealing with high-aspect-ratio grid cells.) Stability analysis (appendix B) shows that a stable scheme is obtained by taking this constant coefficient equal to

$$\epsilon = \frac{1}{4} \left(\left(\frac{C_{FL}}{C_{FL}^*} \right)^2 - 1 \right), \quad (4.10)$$

with C_{FL} the actual Courant number and C_{FL}^* the Courant number for the scheme without residual averaging. As stated before, the ratio of these Courant numbers is generally taken equal to 2, implying $\epsilon = 0.75$.

The difference equation (4.8) can be split into three sets of difference equations (one for each computational direction), which have the following general form, here written for the k -direction:

$$\left((1 - \epsilon^{(k)} \delta_k^2) X \right)_{i,j,k} = Y_{i,j,k} \quad (k = 1 \dots N_k), \quad (4.11)$$

with the boundary conditions

$$X_{i,j,0} = 0, \quad X_{i,j,N_k+1} = 0, \quad (4.12)$$

for $i = 1 \dots N_i$ and $j = 1 \dots N_j$.

For each i and j , this results in a tridiagonal linear system which can be written in the following matrix form (dropping subscripts i and j):

$$\mathcal{B}X = Y, \quad (4.13)$$

with the left-hand-side and right-hand-side vectors given by $X = (X_1, \dots, X_{N_k})^T$ and $Y = (Y_1, \dots, Y_{N_k})^T$, and the matrix \mathcal{B} of dimension $N_k \times N_k$ given by

$$\mathcal{B} = \begin{pmatrix} 1 + 2\epsilon_1 & -\epsilon_1 & & & \\ -\epsilon_2 & 1 + 2\epsilon_2 & -\epsilon_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -\epsilon_{N_k-1} & 1 + 2\epsilon_{N_k-1} & -\epsilon_{N_k-1} \\ & & & -\epsilon_{N_k} & 1 + 2\epsilon_{N_k} \end{pmatrix}. \quad (4.14)$$

Since \mathcal{B} is a tridiagonal matrix, the system can be solved in a straightforward manner by a direct method, using an LU-decomposition (triangular factorization) of \mathcal{B} (i.e. the Thomas algorithm).

4.4 EXTENSIONS NEEDED FOR THE NAVIER-STOKES EQUATIONS

The solution procedure as described in the previous section is found to require too much computation time when used for the Navier-Stokes equations, in particular for the 3D applications of interest here. As indicated in section 4.2, reasonable computation times may be obtained by applying a multi-grid scheme as was developed by Jameson [51] and Martinelli [73], together with the treatment of high-aspect-ratio grid cells of Martinelli. Here, we discuss this multi-grid scheme and in particular consider its application for multi-block structured grids.

4.4.1 Multi-grid scheme

The multi-grid scheme, as applied here, can be considered to be an acceleration of a time-like integration scheme. In this case, the time-like integration scheme is an explicit Runge-Kutta scheme, with local time stepping and residual averaging. The acceleration by the multi-grid scheme is obtained by performing relaxations (i.e. Runge-Kutta steps) on coarser grid levels. On these coarser levels, the allowed time step is larger, while the amount of computational work is smaller than on the finest level.

The basic multi-grid scheme used is the Full Approximation Storage (FAS) multi-grid algorithm [16, 17, 120]. In order to precondition the initial solution on the finest grid level, the Full Multi-Grid (FMG) scheme [17, 103, 120] is used.

For the multi-grid algorithm, a sequence of coarse grid levels must be defined below a given fine grid on which the solution is desired. For a structured grid, these

grid levels are obtained by dropping every other grid point in each computational direction from the next-finer grid. In other words, a coarse grid is obtained by fusing eight grid cells (in 3D) of the next-finer grid into one grid cell. We number the grid levels from $l = l_{\text{coar}}$ (coarsest grid) to $l = l_{\text{fine}}$ (finest grid), with $l_{\text{coar}} < l_{\text{fine}}$. Further, all variables at level l will be indicated by a superscript l .

The Full Approximation Storage multi-grid scheme

On the finest grid level $l = l_{\text{fine}}$, the discrete steady-state equations must be solved, given by

$$R^l(U^l) = 0, \quad (4.15)$$

in which U^l is the collection of flow states of all grid cells in all blocks, and R^l is the residual (as defined in equation (3.9), dropping the subscripts i , j , and k for simplicity). This equation may be solved by applying the Runge-Kutta time-integration scheme as described before. In general, in such a scheme the low-frequency modes are slowly converging. The idea of a multi-grid scheme is to treat those low frequency modes on coarser grids, where they are modes of higher frequency, and where they can be expected to converge faster (at smaller cost).

The Full Approximation Storage (FAS) multi-grid scheme, which is a scheme suitable for non-linear equations, may be derived as follows. Consider at a grid level l the following discrete equation,

$$R^l(U^l) + P^l = 0, \quad (4.16)$$

with P^l a given forcing function ($P^l = 0$ for $l = l_{\text{fine}}$). Let W^l be an approximate solution to this equation (obtained e.g. by several RK time steps) such that the errors in the defect

$$r^l = R^l(W^l) + P^l \quad (4.17)$$

are of low frequency. The equation for the exact solution U^l can then be written as

$$R^l(U^l) - R^l(W^l) + r^l = 0. \quad (4.18)$$

Since r^l contains only low frequencies, this equation can be approximated on the coarse grid level $l-1$ by

$$R^{l-1}(U^{l-1}) - R^{l-1}(T_{l-1,l}(W^l)) + Q_{l-1,l}(r^l) = 0, \quad (4.19)$$

with $T_{l-1,l}$ and $Q_{l-1,l}$ restriction operators, transferring the approximate solution and the defect to the coarse grid level. This equation can be written in the form of equation (4.16) with the forcing function given by

$$P^{l-1} = Q_{l-1,l}(r^l) - R^{l-1}(T_{l-1,l}(W^l)). \quad (4.20)$$

Solving equation (4.19), a coarse-grid correction $U^{l-1} - T_{l-1,l}(W^l)$ for the low-frequency components in the solution is obtained, which is prolonged to the fine grid by

$$W^l = W^l + I_{l,l-1} (U^{l-1} - T_{l-1,l}(W^l)), \quad (4.21)$$

1. Perform N_{pre} (pre)relaxations on grid level l (resulting in an updated solution U^l).

2. If level l is not the coarsest level ($l \neq l_{\text{coar}}$), then:

(a) Define an initial solution U_0^{l-1} on the coarser level $l-1$,

$$U_0^{l-1} = T_{l-1,l} U^l,$$

by approximating the current solution on level l , using the restriction operator $T_{l-1,l}$.

(b) Define a residual forcing function P^{l-1} on level $l-1$,

$$P^{l-1} = Q_{l-1,l} (R^l(U^l) + P^l) - R^{l-1}(U_0^{l-1}),$$

with R^l the residual operator on level l and $Q_{l-1,l}$ another restriction operator.

(c) Perform N_{cgc} FAS cycles (coarse grid corrections) on level $l-1$, resulting in an updated solution U^{l-1} .

(d) Update the current solution on level l by interpolating the correction from the next-coarser level,

$$U^l = U^l + I_{l,l-1} (U^{l-1} - U_0^{l-1}),$$

where $I_{l,l-1}$ is a prolongation operator.

3. Perform N_{post} (post)relaxations on grid level l (resulting in an updated solution U^l).

Figure 4.2: Recursive definition of a FAS cycle

with $I_{l,l-1}$ a prolongation operator. Suitable restriction and prolongation operators will be defined later on.

A solution to the steady-state equation (4.15) can now be obtained by alternately taking RK time steps (in the context of multi-grid called relaxation steps) and determining coarse-grid corrections as just described. A coarse-grid correction can be obtained in the same manner: performing RK time steps for the discrete equation (4.19), and determining coarse-grid corrections using the next-coarser grid level. This process can be continued until the coarsest grid level is reached. This results in the FAS multi-grid scheme for solving equation (4.16) on a certain grid level l , and in particular on the finest grid level l_{fine} . One iteration step or so-called cycle of the FAS scheme consist of the steps given in figure 4.2, which define the cycle recursively.

The relaxation step on level l actually consists of applying one RK time step, as defined in figure 4.1. This time step must be based on the steady-state equation (4.16) and thus in step (4b) of the RK scheme, the forcing function P^l must be added

1. Set the initial solution U_0^l on grid level $l = l_{\text{init}}$ to uniform flow.
2. For $l = l_{\text{init}}$ to $l_{\text{fine}} - 1$ do:
 - (a) Perform N_{coar} FAS cycles on level l , resulting in an updated solution U^l .
 - (b) Define an initial solution U_0^{l+1} on the next-finer grid level $l + 1$ by interpolating the current solution U^l ,

$$U_0^{l+1} = J_{l+1,l} U^l, \quad (4.22)$$

where $J_{l+1,l}$ is a prolongation operator.

3. Perform N_{fine} FAS cycles on level $l = l_{\text{fine}}$, resulting in the final solution.

Figure 4.3: The FMG scheme

to the residual.

The form of the FAS cycle depends on the number of pre-relaxations N_{pre} , the number of coarse grid corrections N_{cgc} and the number of post-relaxations N_{post} . In principle, these numbers can be chosen differently for each grid level. A V-cycle is obtained by applying one coarse-grid correction ($N_{\text{cgc}} = 1$) and a W-cycle by applying two ($N_{\text{cgc}} = 2$). Specific choices will be made later on.

The Full Multi-Grid scheme

The FAS scheme as just described needs an initial solution on the finest grid level to start. One can simply start from a uniform flow, but a more robust scheme is obtained by first determining solutions on the coarse grid levels, and subsequently prolonging these solutions to the next-finer grid levels where they form the initial solution. This will also improve the initial (but not the asymptotic) convergence speed. In the context of multi-grid, this process of grid sequencing is called the Full Multi-Grid (FMG) scheme.

Consider an initial grid level l_{init} , with $l_{\text{coar}} \leq l_{\text{init}} \leq l_{\text{fine}}$. The Full Multi-Grid (FMG) scheme consists of applying the FAS scheme on the grid levels l_{init} through l_{fine} , successively, as specified in figure 4.3. The solution is transferred from a certain grid level to the next finer level by the prolongation operator J , which will be described later on.

Restriction and prolongation operators

In a multi-grid scheme the restriction and prolongation operators must satisfy the following requirement so as not to ruin the convergence speed [17, 120]. Let m be the order of the differential equation considered; for the Euler equations $m = 1$, and for the Navier-Stokes equations $m = 2$. Further, let m_r be the order of accuracy of

the restriction operator, and m_p the order of accuracy of the prolongation operator. Then, the requirement is

$$m_r + m_p > m. \quad (4.23)$$

For the Euler equations, both the restriction and the prolongation operator are allowed to be first order. For the Navier–Stokes equations, at least one of the operators must be second order (or higher). Here, a first-order restriction operator is defined, based on averaging, while a second-order prolongation operator is defined, based on linear interpolation.

The operator $T_{l-1,l}$, for the restriction of the basic variables, is defined by volume-weighted averaging [47, 73] of $U^* = (\rho, \rho \vec{u}, \rho H)^T$,

$$U_{i,j,k}^{*l-1} = \frac{1}{\tilde{V}_{i,j,k}^{l-1}} \sum_{\Delta i=-1}^0 \sum_{\Delta j=-1}^0 \sum_{\Delta k=-1}^0 (V^l U^{*l})_{2i+\Delta i, 2j+\Delta j, 2k+\Delta k}, \quad (4.24)$$

for $i = 1 \dots N_i^{l-1}$, $j = 1 \dots N_j^{l-1}$, and $k = 1 \dots N_k^{l-1}$, with

$$\tilde{V}_{i,j,k}^{l-1} = \sum_{\Delta i=-1}^0 \sum_{\Delta j=-1}^0 \sum_{\Delta k=-1}^0 V_{2i+\Delta i, 2j+\Delta j, 2k+\Delta k}^l. \quad (4.25)$$

Note that $N_i^l = 2N_i^{l-1}$. The volume-weighted averaging is applied to total enthalpy, because this makes it possible to apply enthalpy damping on all levels for the Euler equations ($H^l \equiv H_\infty$ implies now $H^{l-1} \equiv H_\infty$).

The restriction operator $Q_{l-1,l}$ for the restriction of residuals is defined by a simple summation of the total flux balance [47, 73]. Since the residual is equal to the total flux balance divided by the cell volume, the following definition results:

$$(Q_{l-1,l} R^l)_{i,j,k} = \frac{1}{V_{i,j,k}^{l-1}} \sum_{\Delta i=-1}^0 \sum_{\Delta j=-1}^0 \sum_{\Delta k=-1}^0 (V^l R^l)_{2i+\Delta i, 2j+\Delta j, 2k+\Delta k}, \quad (4.26)$$

for $i = 1 \dots N_i^{l-1}$, $j = 1 \dots N_j^{l-1}$, and $k = 1 \dots N_k^{l-1}$.

Both the prolongation operator $I_{l,l-1}$, for the transfer of the correction $\Delta U^{l-1} = U^{l-1} - U_0^{l-1}$, as well as the operator $J_{l,l-1}$, for prolongating the solution U^{l-1} , are defined by trilinear interpolation in the computational space. Thus, the prolonged solution U^l (or similarly the prolonged correction ΔU^l) is obtained by linearly interpolating first in the i direction,

$$\begin{cases} A_{2i-1,j,k} &= \frac{3}{4} U_{i,j,k}^{l-1} + \frac{1}{4} U_{i-1,j,k}^{l-1}, \\ A_{2i,j,k} &= \frac{3}{4} U_{i,j,k}^{l-1} + \frac{1}{4} U_{i+1,j,k}^{l-1}, \end{cases}, \quad (4.27)$$

then in the j direction,

$$\begin{cases} B_{i_f, 2j-1, k} &= \frac{3}{4} A_{i_f, j, k} + \frac{1}{4} A_{i_f, j-1, k}, \\ B_{i_f, 2j, k} &= \frac{3}{4} A_{i_f, j, k} + \frac{1}{4} A_{i_f, j+1, k}, \end{cases}, \quad (4.28)$$

and finally in the k direction,

$$\begin{cases} U_{i_f, j_f, 2k-1}^l &= \frac{3}{4}B_{i_f, j_f, k} + \frac{1}{4}B_{i_f, j_f, k-1}, \\ U_{i_f, j_f, 2k}^l &= \frac{3}{4}B_{i_f, j_f, k} + \frac{1}{4}B_{i_f, j_f, k+1}, \end{cases}, \quad (4.29)$$

with $i = 1 \dots N_i^{l-1}$, $j = 1 \dots N_j^{l-1}$, and $k = 1 \dots N_k^{l-1}$, and with $i_f = 1 \dots N_i^l$, $j_f = 1 \dots N_j^l$, and $k_f = 1 \dots N_k^l$.

For the trilinear interpolation, boundary conditions are required, in particular values are needed in the dummy cells for the flow solution U^{l-1} or the flow correction ΔU^{l-1} . The dummy-cell values for the flow solution are obtained by applying the boundary conditions as defined in section 3.5. The dummy-cell values of the flow correction are obtained by first determining the dummy-cell values of the flow solutions U^{l-1} and U_0^{l-1} (by the boundary conditions of section 3.5) and subsequently taking their difference.

Choices for multi-grid and Runge–Kutta schemes

Finally, we must choose the type of multi-grid cycle that will be used, as well as suitable Runge–Kutta schemes.

For the Runge–Kutta scheme, when used within a multi-grid scheme, an important property is the strong damping of high-frequency modes. For the Euler equations, we will use a three-stage scheme RK3 with a single evaluation of the diffusive terms, following Jameson [51]:

$$\begin{aligned} \alpha_1 &= 0.6, & \alpha_2 &= 0.6, & \alpha_3 &= 1, \\ l_0 &= 0, & l_1 &= 0, & l_2 &= 0, \\ C_{FL} &= 1.5, & k^{(4)} &= 2. \end{aligned} \quad (4.30)$$

The specified value of the fourth-order artificial-diffusion coefficient $k^{(4)}$ ensures the maximum damping of high-frequency modes for this scheme (see appendix B). The RK4 scheme as specified before, is also found to be reasonably efficient for the Euler equations.

For the TLNS equations, a large stability limit for the diffusive terms is desired. The following five-stage scheme (RK5) with three evaluations of the diffusive terms is frequently used [73, 114]:

$$\begin{aligned} \alpha_1 &= 1/4, & \alpha_2 &= 1/6, & \alpha_3 &= 3/8, & \alpha_4 &= 1/2, & \alpha_5 &= 1, \\ l_0 &= 0, & l_1 &= 0, & l_2 &= 2, & l_3 &= 2, & l_4 &= 4, \\ C_{FL} &= 3, & R_K &= 3, & k^{(4)} &= 2. \end{aligned} \quad (4.31)$$

Sometimes the diffusion at the third and fifth stages is calculated by a combination of the diffusion at the first, third and fifth stages [109]. However, this option is not part of the present scheme.

Finally, the type of multi-grid cycle should be defined. For the Euler equations, the V-cycle with one pre-relaxation and one post-relaxation works well [47],

$$N_{\text{pre}} = 1, \quad N_{\text{cgc}} = 1, \quad N_{\text{post}} = 1, \quad (4.32)$$

while for Navier–Stokes computations the W-cycle is commonly used [73, 109, 114], also with one pre-relaxation and one post-relaxation,

$$N_{\text{pre}} = 1, \quad N_{\text{cgc}} = 2, \quad N_{\text{post}} = 1. \quad (4.33)$$

We found that the application of more pre and post-relaxations ($N_{\text{pre}} = N_{\text{post}} = 5$) generally leads to a more robust scheme. The increased robustness may be due to the increased number of relaxations performed between two prolongations, so that possible high-frequency errors induced by the prolongation operator are damped more strongly. For this purpose, other authors [109, 114] employed smoothing of the prolonged flow correction (by an operator similar to the implicit residual averaging operator). The increase of the number of relaxations per cycle is also slightly more efficient if in total the same number of fine-grid relaxations is performed, due to relatively less restrictions and prolongations.

Multi-block aspects

For the multi-grid algorithm just described, we must consider how it is applied on a multi-block structured grid. Two main strategies may be considered:

- multi-block inside multi-grid; meaning that on each grid level of a multi-grid cycle a loop over all blocks is performed,
- multi-grid inside multi-block; meaning that a complete multi-grid cycle is performed for each block successively.

This resembles the two possible strategies for the Runge–Kutta scheme, where either all the blocks are updated at each stage of a time step, or a complete time step is performed for each block successively. For the RK scheme, we chose the second strategy since it requires less memory, while in practice it does not deteriorate the convergence speed. For the multi-grid scheme, however, we choose the first strategy (multi-block inside multi-grid). The second strategy has as effect that so much work is done in a block before other blocks are updated (thus a large time lag between the blocks results) that the convergence speed may be significantly reduced or the scheme may even become unstable [96]. The memory advantage of the second strategy (coarse-grid data only needs to be stored one block at a time) is limited. Other authors have employed the first strategy successfully as well [4, 36, 96].

On a multi-block grid, the prolongation of the flow correction from level $l - 1$ to level l poses a problem. Consider a block at grid level $l - 1$. At internal block faces, dummy-cell values of the flow correction must be obtained from the adjacent block. The flow correction consists of the difference between the current solution U^{l-1} and the initial solution U_0^{l-1} . The current solution is available in the adjacent block, but the initial solution is not (to save memory) and should be redetermined by restricting the solution U^l . We found the implementation of this algorithm to be needlessly complex. Instead, we simplified the algorithm by setting the dummy-cell values of the correction to zero. This results in an underrelaxation of the correction near block interfaces, which also improves the robustness, while in practice no reduction of convergence speed is found. (See section 4.5, where single-block and multi-block convergence histories are compared.)

For the definition of the coarse grid levels two options may be considered: global coarsening and local coarsening. Global coarsening means that the same number of grid levels is used in each block. For a grid with a large number of blocks, there will usually be some block with small grid dimensions which then limits the number of grid levels in all other blocks. Therefore, we also allow the option of local coarsening which means that the maximum number of grid levels is used in each block separately. In figure 4.4, 1D examples are given for both these coarsening options.

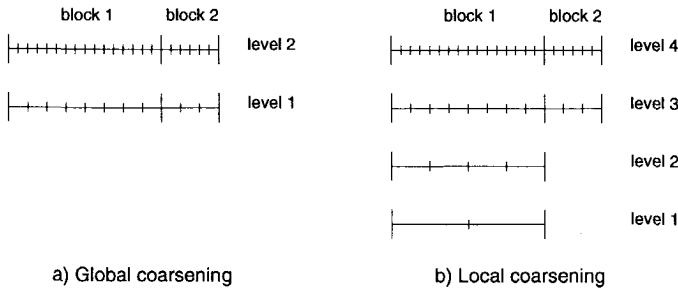


Figure 4.4: Coarsening options for two blocks with different dimensions

A consequence of local coarsening is that at a certain coarse grid level, there may not be a grid defined for each block. We can make this grid complete, however, by considering the coarsest available grid to be present in the blocks that could not be further coarsened, as sketched in figure 4.5. The solution in these blocks can then just be used to determine the dummy-cell values of the other blocks, or optionally one could also perform relaxations in these blocks. At block interfaces, the grid may then be partially continuous, and thus appropriate boundary conditions, as described in section 3.5.5, must be used. We expect the local coarsening scheme to be successful when the blocks that can be further coarsened are in regions of the flow field where convergence is usually the slowest (such as in the boundary layers). Results with local coarsening are shown in section 4.5. (A concept related to local coarsening is described by Mavriplis [74] for a multi-grid scheme applied to the Euler equations on adaptively-refined unstructured grids.)

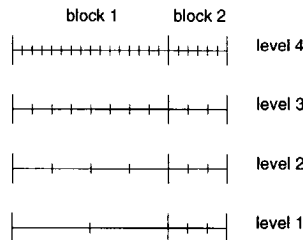


Figure 4.5: Complete locally coarsened grid for two blocks with different dimensions

4.4.2 Treatment of high-aspect-ratio grid cells

The multi-grid scheme that has been defined in the previous section is known to give poor performance on grids with high-aspect-ratio cells. The main problem is that high-frequency modes in the direction of the large mesh size are poorly damped. To resolve this problem, two modifications of the scheme are important that were introduced by Martinelli [73]: using varying coefficients in the residual averaging scheme, and scaling the artificial diffusion for high-aspect-ratio cells. (For some analysis of the effect of high-aspect-ratio cells on the high-frequency damping, see appendix B.)

Residual averaging with varying coefficients

For a grid cell with a high aspect ratio (of the order 10^4), the time step is limited by the smallest mesh size, usually in boundary-layer normal direction. For the larger mesh sizes, usually in tangential directions, the time step is thus several orders of magnitude too small for obtaining a good damping of high-frequency modes. Applying residual averaging allows larger time steps, but at the same time reduces the high-frequency damping even further. Therefore, residual averaging is only applied in the direction of the smallest mesh size by defining the residual-averaging coefficients dependent on the cell aspect ratio. For the direction of the largest mesh size, this means a larger time step and no residual averaging, and therefore improved damping of the high-frequency modes.

The appropriate time step for a computational direction does not just depend on the related mesh size, but more general it depends on the spectral radius of the related convective flux Jacobian (as in equation (4.4)). Thus, the residual-averaging coefficients are defined as a function of the ratio of the spectral radii of the different computational directions. Let (I, J, K) be a cyclic permutation of (i, j, k) . The coefficients $\epsilon^{(K)}$ are defined as follows:

$$\epsilon_{I,J,K}^{(K)} = \max \left\{ \frac{1}{4} \left(\left(\varepsilon \frac{\tilde{\lambda}_{I,J,K}^{(K)}}{\lambda_{I,J,K}^{(I)} + \lambda_{I,J,K}^{(J)} + \lambda_{I,J,K}^{(K)}} \right)^2 - 1 \right), 0 \right\}, \quad (4.34)$$

with ε a constant coefficient, $\lambda^{(K)}$ given by equation (3.25), and $\tilde{\lambda}^{(K)}$ evaluated similarly as in equation (3.74),

$$\begin{aligned} \tilde{\lambda}_{I,J,K}^{(K)} &= \left(1 + \left(r_{I,J,K}^{(IK)} \right)^{k^{(h)}} + \left(r_{I,J,K}^{(JK)} \right)^{k^{(h)}} \right) \lambda_{I,J,K}^{(K)}, \\ r_{I,J,K}^{(IK)} &= \lambda_{I,J,K}^{(I)} / \lambda_{I,J,K}^{(K)}. \end{aligned} \quad (4.35)$$

With residual averaging, we have found that a reasonably efficient and robust scheme is obtained for $\varepsilon = 1.5$. For the three- and five-stage schemes, residual averaging is best applied every stage, while for the four-stage scheme, residual averaging is best applied every odd stage. Appropriate stability limits are generally obtained by scaling C_{FL} with ε and R_K with ε^2 (see appendix B). For the RK5 scheme, used for the TLNS equations, the stability limits are then given by $C_{FL} = 4.5$ and $R_K = 1.5C_{FL}$. For the RK3 and RK4 schemes, used for the Euler equations in

combination with enthalpy damping, the stability limits are given by $C_{FL} = 3.2$ and $C_{FL} = 4.2$, respectively.

High-aspect-ratio scaling of the artificial diffusion

To further improve the damping of high-frequency modes in the direction of the largest mesh size for a grid cell with high aspect ratio, the artificial diffusion is scaled as described in section 3.6.2. The high-frequency damping becomes most effective, if the parameter $k^{(h)}$ is taken equal to one, with the damping being the same in all directions. In this case, the perfect multi-grid convergence, independent from the number of grid cells, may be obtained. However, this could seriously reduce the numerical accuracy, since the artificial diffusion may then easily be increased by a factor of the order 10^4 . As stated before, a lower value of $k^{(h)} = 1/2$ for 3D flow calculations seems to be a reasonable compromise between convergence speed and numerical accuracy.

4.5 RESULTS

In this section, we will verify that the solution procedure behaves as expected. In particular, the convergence speeds of the basic scheme (single grid), and of the improved scheme (multi-grid scheme and treatment of high-aspect-ratio grid cells) will be evaluated, and we will consider whether the computation times are sufficient to satisfy the efficiency requirement stated in chapter 1.

Two-dimensional inviscid flow

We will first evaluate the multi-grid scheme on a grid without cells of high aspect ratio, which is typically used for inviscid flows. On such a grid it must be possible to obtain a convergence speed near to the ‘ideal’ multi-grid convergence, i.e. a convergence speed independent of the number of grid points. As a test case, we take the inviscid, transonic flow around the NACA0012 airfoil with the flow conditions $\alpha = 1.25^\circ$ and $M_\infty = 0.8$. An O-type single-block grid is used with 256×128 grid cells. Figure 4.6 gives an impression of this grid and of the flow solution obtained on it.

Since for this case, the Euler equations are to be solved, we use as basic solution method the RK3 scheme with enthalpy damping and constant-coefficient residual averaging. Figure 4.7 gives the convergence history in terms of the residual of the mass equation (root-mean-square value over the complete grid) and the lift coefficient versus the number of relaxations. For the multi-grid scheme, the number of relaxations performed on the finest grid is considered, which is equal to the number of multi-grid cycles times the number of pre- and post-relaxations. The multi-grid scheme, using a V-cycle with $N_{pre} = N_{post} = 1$, gives a significant increase of the convergence speed compared to the basic scheme. Using varying-coefficient residual averaging gives a similar convergence speed. The ‘ideal’ multi-grid convergence, however, is not yet reached (figure 4.8). We approach the ideal more closely, if the number of pre- and post-relaxations is increased to $N_{pre} = N_{post} = 5$, in which case the convergence on the finest of the four grid levels is only slightly slower than on the other grid levels. A

possible explanation is that if not enough relaxations are applied, then high-frequency errors introduced by the prolongation operator are not sufficiently damped and reduce the convergence speed. Generally, one could state that a single explicit time step is not a strong enough smoother (of high frequencies) as required by the multi-grid theory as relaxation operator; five explicit time steps apparently do form a strong enough smoother (see also appendix B).

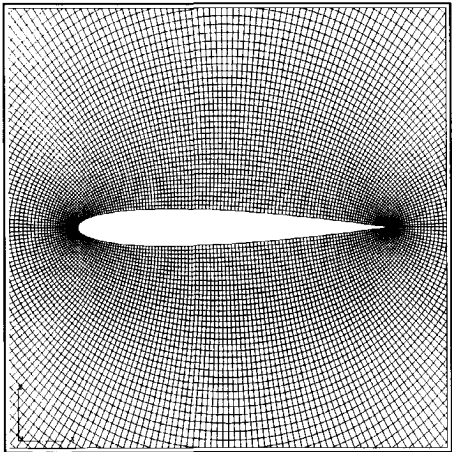
To quantify the increased convergence speed with the multi-grid scheme, we consider the number of fine-grid relaxations needed to obtain a convergence of the lift, drag, and pitch coefficient within 10^{-4} or 10^{-5} of the fully-converged value (i.e. within 1 or 0.1 count for the drag coefficient), or to obtain a root-mean-square residual of the mass equation of 10^{-5} , as given in table 4.1. Also, the needed CPU-time is given (on the NEC-SX4). With $N_{\text{pre}} = N_{\text{post}} = 1$, the multi-grid scheme is 2.1 times as expensive per fine-grid relaxation as the basic scheme, while with $N_{\text{pre}} = N_{\text{post}} = 5$ it is only 1.6 times as expensive (due to the reduced number of restrictions and prolongations). In terms of the convergence of the residual, the multi-grid scheme is 6 to 9 times as efficient as the basic scheme in terms of number of relaxations, and 3 to 6 times in terms of the CPU-time. However, the multi-grid scheme gives a stronger improvement of efficiency for the force coefficients: factors 20 to 50 (relaxations) or 10 to 25 (CPU-time).

Whether the multi-grid convergence speed is maintained for a multi-block grid is verified for the same test case, where the grid is divided into four blocks of equal size in the circumferential direction. Basically the same scheme is employed as for the single-grid case (with the treatment of the blocks following the part on multi-blocks aspects of section 4.4.1). Using a V-cycle with $N_{\text{pre}} = N_{\text{post}} = 1$, the convergence completely breaks down on the third grid level (figure 4.9). Again, if the number of pre- and post-relaxations is increased to $N_{\text{pre}} = N_{\text{post}} = 5$, then the convergence speed improves, and in fact becomes practically equal to the convergence speed for the single-block case (figure 4.10). Thus, we may conclude that for the considered multi-grid scheme, increasing the number of pre- and post-relaxations does not only improve the efficiency but also the robustness of the scheme.

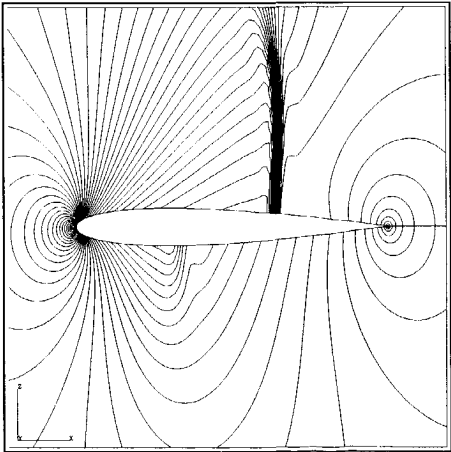
	converged within 10^{-4}			mass residual of 10^{-5}
	lift	drag	pitch	
SG	2670(194)	864(62.7)	1570(114)	1810(131)
MG-1	56(8.57)	18(2.75)	64(9.79)	318(48.7)
MG-5	90(10.3)	30(3.42)	50(5.70)	200(22.8)

	converged within 10^{-5}		
	lift	drag	pitch
SG	>3200(232)	2290(166)	3100(225)
MG-1	150(23.0)	132(20.2)	162(24.8)
MG-5	120(13.7)	60(6.84)	100(11.4)

Table 4.1: Number of fine-grid relaxations (CPU seconds) performed to obtain a certain level of convergence for inviscid flow around NACA0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$) on finest grid level (256×128), comparing single grid (SG) with multi grid with 1 or 5 pre- and post-relaxations (MG-1 resp. MG-5)



a) Grid



b) Pressure distribution

Figure 4.6: Impression of O-type single-block grid (256×128 grid cells) and inviscid flow solution around NACA0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$)

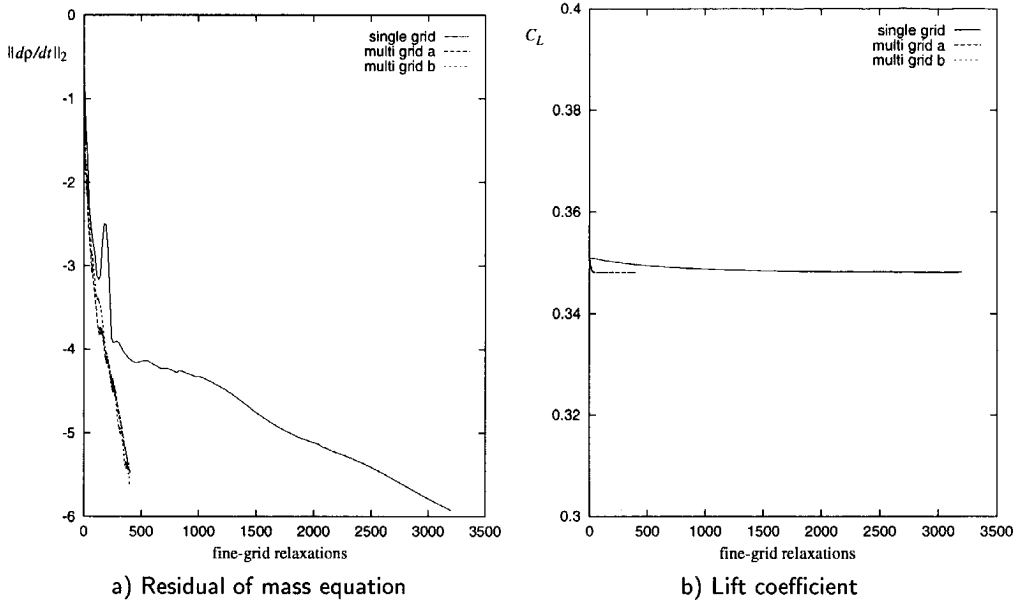


Figure 4.7: Convergence history for inviscid flow around NACA0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$) on finest grid level (256×128) comparing the basic method (single grid), multi grid with constant-coefficient residual averaging (multi grid a), and multi grid with varying-coefficient residual averaging (multi grid b)

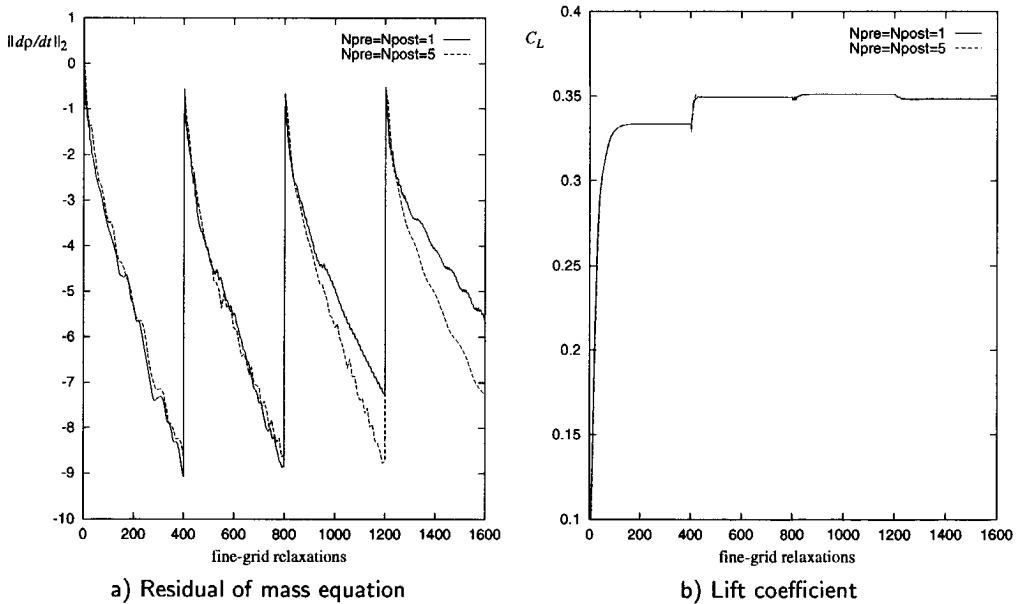


Figure 4.8: Convergence history for inviscid flow around NACA0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$) on four subsequent grid levels using V-cycle multi grid, comparing $N_{pre} = N_{post} = 1$ with $N_{pre} = N_{post} = 5$

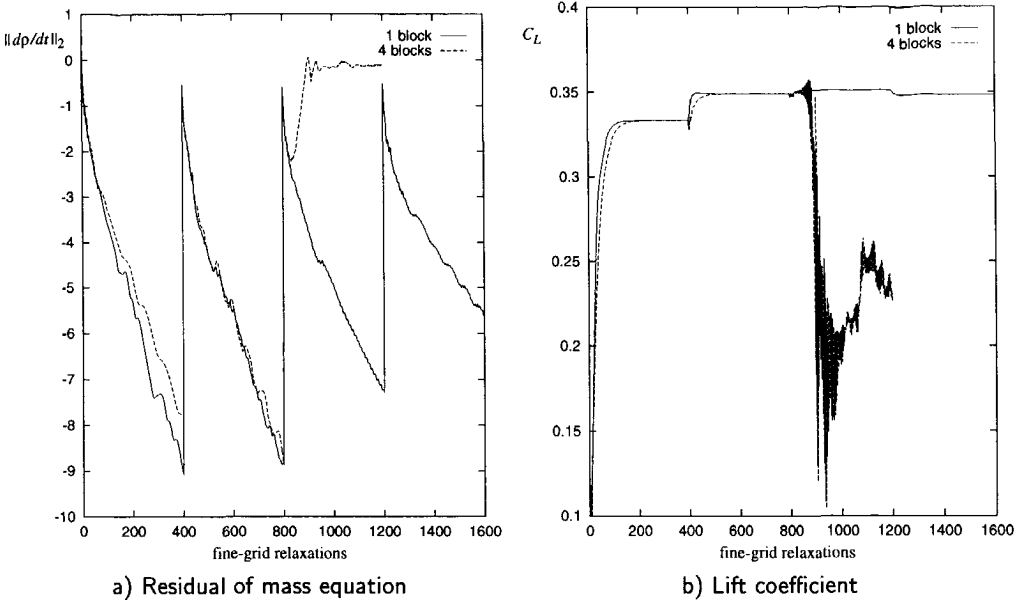


Figure 4.9: Convergence history for inviscid flow around NACA0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$) on four subsequent grid levels using V-cycle multi grid with $N_{pre} = N_{post} = 1$, comparing results for single-block grid with those for four-block grid

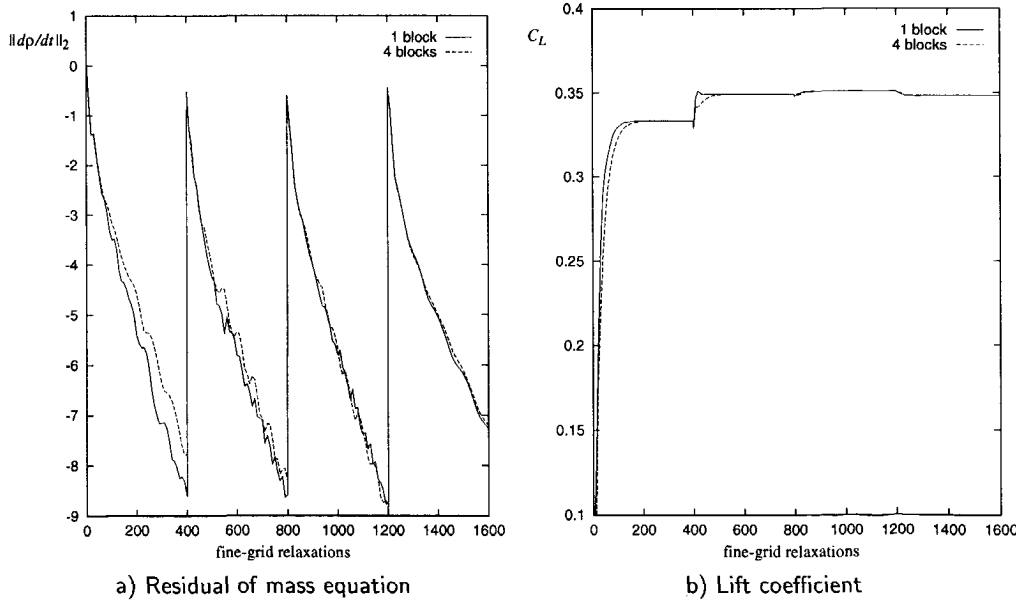


Figure 4.10: Convergence history for inviscid flow around NACA0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$) on four subsequent grid levels using V-cycle multi grid with $N_{pre} = N_{post} = 5$, comparing results for single-block grid with those for four-block grid

Two-dimensional turbulent flow

Next, we will evaluate the multi-grid scheme on a grid with cells of high aspect ratio, typically used for viscous flows. On such a grid, we do not expect to reach the ideal multi-grid convergence completely, as explained in section 4.4.2. As a test case, we take the turbulent, transonic flow around the RAE2822 airfoil at the flow conditions $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$, and transition at 3% (case 9). This case was also used in the previous chapter, but here we first employ a somewhat coarser, single-block grid (C-type with 256×64 grid cells). With the basic solution method, consisting of the RK5 scheme with constant-coefficient residual averaging, the convergence is very slow, in particular for the force coefficients (figure 4.11). The multi-grid scheme, using a W-cycle with $N_{\text{pre}} = N_{\text{post}} = 5$, improves the initial convergence speed, but seems to give the same asymptotic convergence speed as the single-grid computation. As anticipated, the high cell aspect ratios are the problem. Using varying-coefficient residual averaging and high-aspect-ratio scaling of the artificial diffusion (with the parameter $k^{(h)} = 2/3$), a convergence speed is obtained on this grid which is close to the ideal multi-grid convergence. The scaling of the artificial diffusion, however, also modifies the numerical solution, reducing the lift coefficient by 0.004 (0.5%) and increasing the drag coefficient by 2 counts (1%) (with the friction drag only increased by 0.2 counts) on the finest grid. As can be seen from the surface pressure and skin-friction distributions, figure 4.13, the high-aspect-ratio scaling appears to smear out the shock and to create a peak in skin friction at the trailing edge, both involving regions with high gradients in tangential direction. (The mild flow separation at the foot of the shock and at the trailing edge are also suppressed.) To reduce these effects, either the number of cells in tangential direction should be increased (as done for this flow case in the previous chapter) or the high-aspect-ratio scaling should be reduced (e.g. by setting $k^{(h)} = 1/2$, as we will do for 3D flow computations).

Again, the increased convergence speed with the multi-grid scheme is quantified by considering the number of fine-grid relaxations and the CPU-time needed to obtain a convergence of the lift, drag, and pitch coefficient within 10^{-4} , or to obtain a root-mean-square residual of the mass equation of $10^{-2.2}$, as given in table 4.2. With high-aspect-ratio scaling, the efficiency of the multi-grid scheme, compared to that of single grid, is increased by factors 44 (relaxations) or 18 (CPU-time) for the drag coefficient, and by factors 13 (relaxations) or 6 (CPU-time) for the mass residual. (Note that, for single-grid computation, the lift and pitch coefficients were not yet converged within 10^{-4} after 2000 relaxations.)

So far in this section, the scalar model has been used for the artificial diffusion. We will also consider the effect on the convergence speed of the matrix artificial-diffusion model. Since matrix diffusion reduces the artificial diffusion in boundary-layer normal direction to improve the numerical accuracy, we may expect that it also reduces the convergence speed. This reduction is minimized by applying the matrix diffusion only on the finest grid level of a multi-grid cycle. As can be seen from figure 4.12, the asymptotic convergence speed is lowered, but the initial convergence speed is not affected by matrix diffusion, in particular on the finest grid level. For most practical computations, the computations will not be converged to levels at which the reduction of the convergence speed by matrix diffusion occurs. From table 4.2, it can be seen

	converged within 10^{-4}			mass residual of $10^{-2.2}$
	lift	drag	pitch	
SG	>2000(200)	1740(174)	>2000(200)	1990(199)
MG-a	600(142)	170(40.1)	390(92.0)	440(104)
MG-b	110(26.0)	40(9.44)	80(18.9)	150(35.4)
MG-c	140(33.3)	40(9.52)	90(21.4)	150(35.7)

Table 4.2: Number of fine-grid relaxations (CPU seconds) performed to obtain a certain level of convergence for turbulent flow around RAE2822 airfoil (case 9, $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) on finest grid level (256×64), comparing results for single grid (SG) with those for multi grid without and with high-aspect-ratio scaling (MG-a and MG-b, resp.) and multi grid with matrix diffusion (MG-c)

	converged within 10^{-4}			mass residual of 10^{-3}
	lift	drag	pitch	
MG-b	110(159)	50(72.5)	70(101)	340(493)
MG-c	80(116)	40(58.4)	60(87.5)	360(525)

Table 4.3: Number of fine-grid relaxations (CPU seconds) performed to obtain a certain level of convergence for turbulent flow around RAE2822 airfoil (case 9, $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) on finest grid level (512×96) of the 8-block grid, using multi grid with scalar and with matrix diffusion (MG-b and MG-c, resp.)

that the number of relaxations, as well as the CPU-time, to obtain force coefficients converged within 10^{-4} is practically the same for scalar and matrix diffusion.

Finally, we also present the convergence for this flow case on the fine multi-block grid (512×96 grid cells) employed in the previous chapter (figure 4.14 and table 4.3). The multi-grid scheme on this 8-block grid uses the local coarsening strategy explained in section 4.4.1: 4 grid levels are used in the 4 inner blocks (near the airfoil), and only 3 grid levels in the 4 outer blocks. After 300 fine-grid relaxations, the mass residual is converged to 10^{-3} on the finest grid, and up to that level scalar and matrix diffusion give practically identical convergence speeds. All force coefficient are then converged within 10^{-6} . Converging the force coefficients within 10^{-4} only takes about 100 fine-grid relaxations, which is equivalent to 10 multi-grid cycles and takes about 140 CPU seconds.

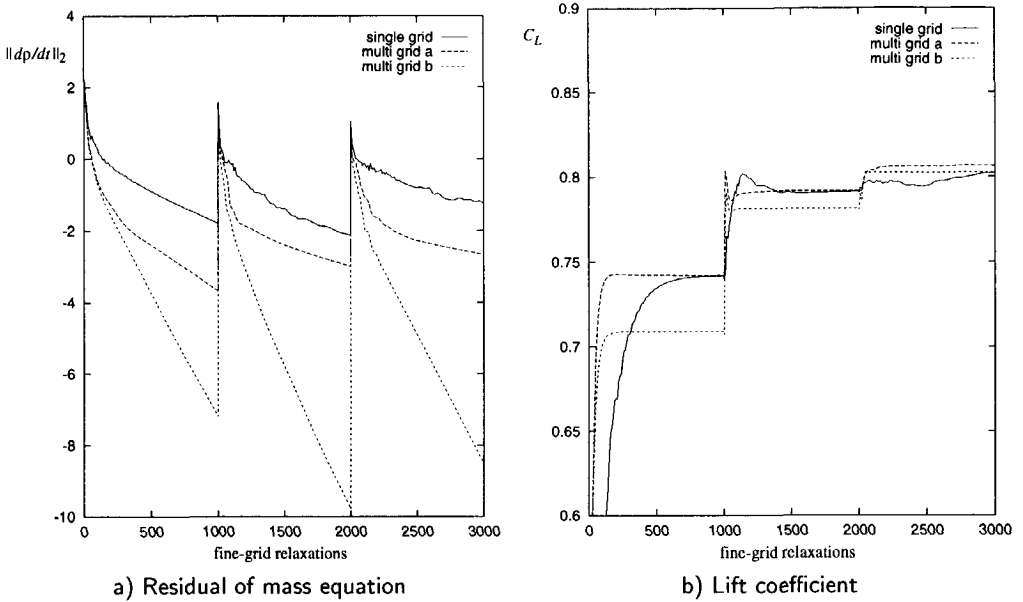


Figure 4.11: Convergence history for turbulent flow around RAE2822 airfoil (case 9, $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) on three subsequent grid levels (fine: 256×64), comparing the basic method (single grid), multi grid without high-aspect-ratio scaling (multi grid a), and multi grid with high-aspect-ratio scaling (multi grid b)

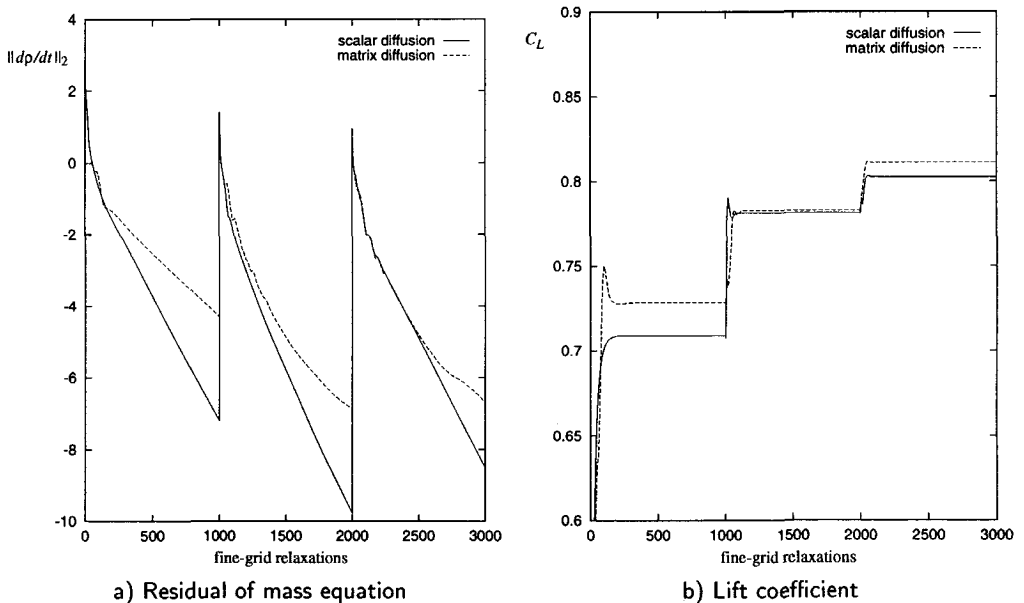


Figure 4.12: Convergence history for turbulent flow around RAE2822 airfoil (case 9, $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) on three subsequent grid levels (fine: 256×64), comparing scalar artificial diffusion with matrix artificial diffusion

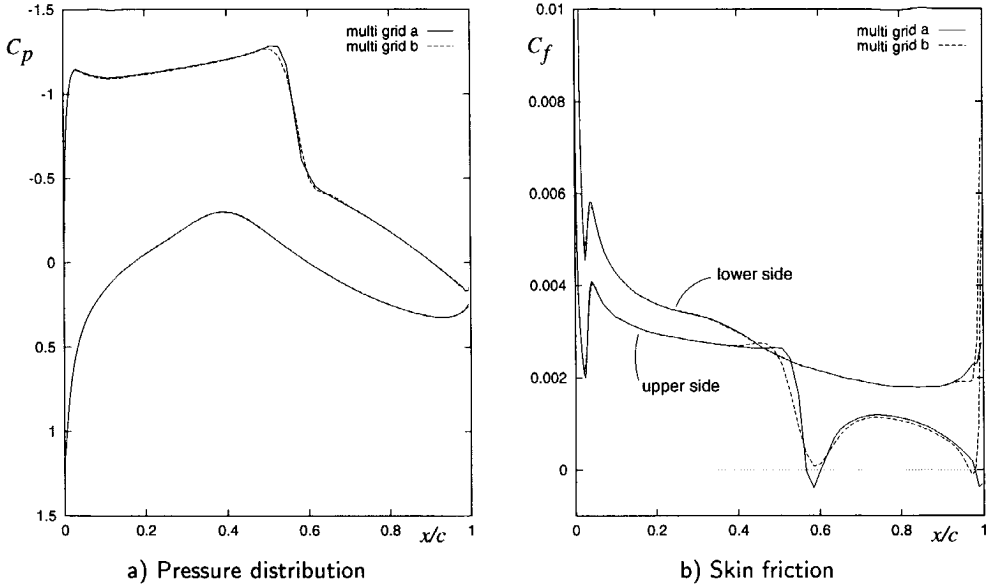


Figure 4.13: Turbulent flow around RAE2822 airfoil (case 9, $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) on finest grid level (256×64), comparing the solution without high-aspect-ratio scaling of the artificial diffusion (multi grid a) with the solution with high-aspect-ratio scaling (multi grid b)

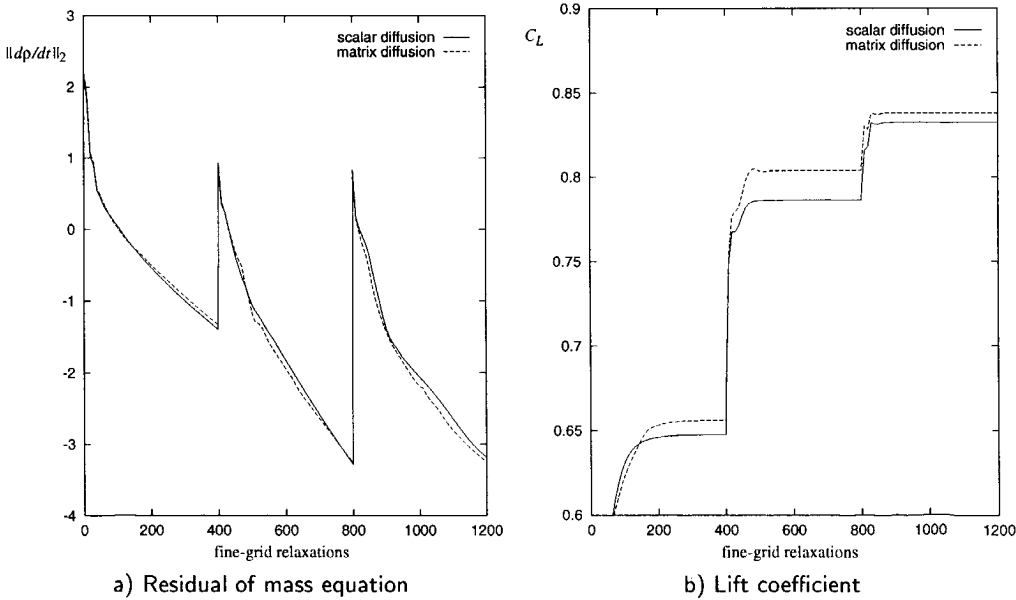


Figure 4.14: Convergence history for turbulent flow around RAE2822 airfoil (case 9, $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$) on three subsequent grid levels of the 8-block fine grid (512×96 cells)

converged within 10^{-4}						mass residual of $10^{-2.5}$
lift	drag	side	pitch	roll	yaw	
80(655)	50(409)	60(491)	80(655)	80(655)	50(409)	980(8020)

Table 4.4: Number of fine-grid relaxations (CPU seconds) performed to obtain a certain level of convergence for turbulent flow around ONERA M6 wing ($M_\infty = 0.84$, $\alpha = 3.06^\circ$, $Re_\infty = 11.8 \cdot 10^6$) on finest grid level ($256 \times 64 \times 48$) using multi grid with matrix diffusion

Three-dimensional turbulent flow

As a 3D test case, we consider the turbulent flow around the ONERA M6 wing with the flow conditions $M_\infty = 0.84$, $\alpha = 3.06^\circ$, $Re_\infty = 11.8 \cdot 10^6$, and transition at 3% of the local chord. A 12-block CO-type grid is employed with $256 \times 64 \times 48$ grid cells (chordwise, normal, and spanwise directions). Figure 4.16 gives an impression of the grid in the half space and of the flow solution (obtained with the Cebeci–Smith turbulence model). The same multi-grid scheme as for the 2D turbulent flow case is employed (W-cycle, $N_{\text{pre}} = N_{\text{post}} = 5$, RK5, high-aspect-ratio scaling), but with the high-aspect-ratio scaling parameter reduced to $k^{(h)} = 1/2$. The convergence speed (figure 4.15) appears to decrease on finer grids (thus the ideal multi-grid convergence is not fully reached). Again, this can be attributed to the high cell aspect ratios, which are generally larger in 3D than in 2D (due to large mesh sizes in spanwise direction). For practical computations, however, we consider this convergence speed to be sufficient. The force coefficients are converged within 10^{-4} in 80 fine-grid relaxations (8 multi-grid cycles) which takes less than 11 CPU minutes, while after 980 fine-grid relaxations (about 2 CPU hours) the mass residual has dropped to $10^{-2.5}$ (table 4.4) and the force coefficients are converged well within 10^{-6} .

Computations for wing–body configurations including a propulsion system may require about 4 times as many grid cells (in the order of 3 million) if only the boundary layer on the wing is simulated. Computation times are increased by the same factor, and thus 3/4 to 8 CPU hours will then be needed, depending on the required level of convergence. This falls within the efficiency requirement stated in chapter 1.

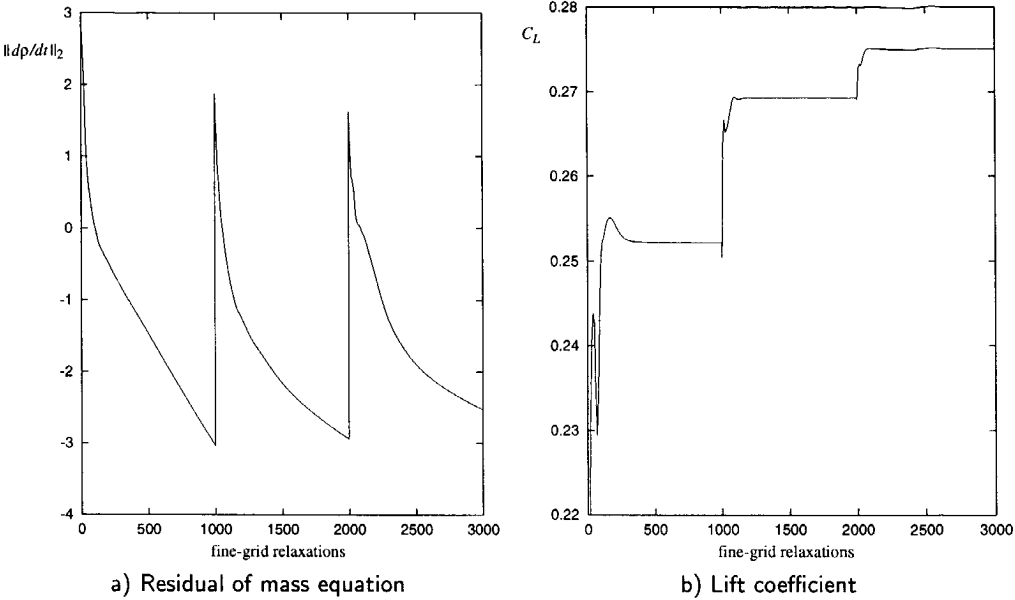


Figure 4.15: Convergence history for turbulent flow around ONERA M6 wing ($M_\infty = 0.84$, $\alpha = 3.06^\circ$, $Re_\infty = 11.8 \cdot 10^6$) on three subsequent grid levels (fine: $256 \times 64 \times 48$)

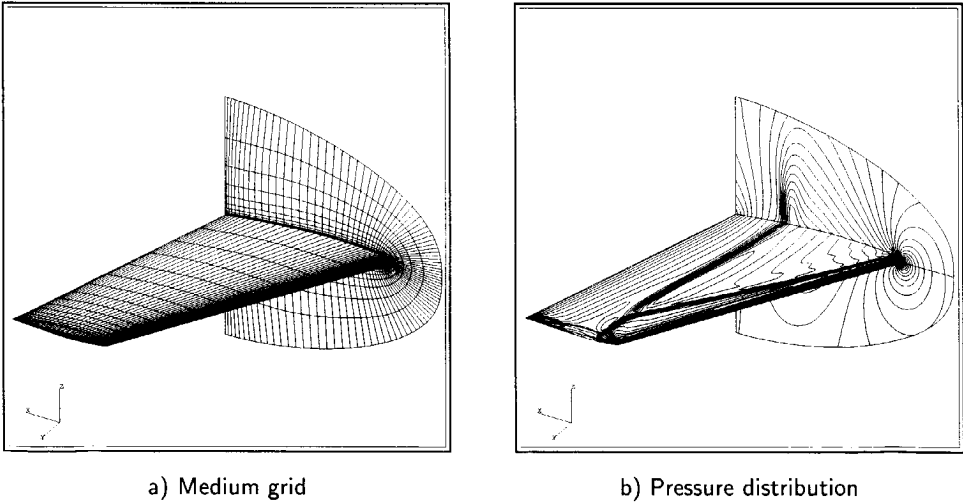


Figure 4.16: Impression of 12-block grid ($256 \times 64 \times 48$ grid cells) and pressure distribution on the surface of ONERA M6 wing as well as in part of the symmetry plane ($M_\infty = 0.84$, $\alpha = 3.06^\circ$, $Re_\infty = 11.8 \cdot 10^6$)

TURBULENCE MODELLING

5.1 INTRODUCTION

In the description of the flow model, as well as in the space discretization and in the solution procedure, we have left one issue open: the modelling of turbulence. In this chapter, we will discuss some turbulence models, and focus mainly on the robust implementation of these models.

As stated in chapter 2, simple algebraic models are used. These models were developed for two-dimensional, attached turbulent boundary layers and wakes, making use of a priori knowledge of the flow physics. Mainly due to their simplicity and cheapness, they are often used as a first step in developing a Navier–Stokes solver for turbulent flows around wings, and as such have become a standard. These models are not generally applicable, but reasonable results have been obtained for attached or mildly separated boundary layers not deviating strongly from essentially 2D flows [46, 97]. Further, for these types of flow, there was no evidence that more sophisticated models (such as two-equation models) would give significantly better results [14, 46], at the time when the development of this Navier–Stokes solver started.

First, a short review of turbulence models is given (section 5.2), followed by descriptions of the algebraic models considered here: the Cebeci–Smith model, the Baldwin–Lomax model, and the Johnson–King model (sections 5.3 to 5.5), and finally a discussion of the numerical implementation of these models (section 5.6).

5.2 REVIEW OF TURBULENCE MODELS

For the Reynolds-averaged Navier–Stokes equations, a turbulence model is needed to model the so-called Reynolds-stress tensor and the turbulent heat-flux vector, which describe how the turbulent fluctuations influence the mean flow. Two main classes of turbulence models may be considered: eddy-viscosity models and Reynolds-stress transport models.

In the eddy-viscosity turbulence models, the Reynolds-stress tensor is usually assumed to be proportional to the rate-of-strain of the mean flow (Boussinesq’s hypothesis). The resulting scalar coefficient of proportionality is called the eddy-viscosity coefficient. This coefficient can be expressed as the product of a turbulence length and a turbulence velocity scale. Based on the number of additional partial differential equations (PDE’s) that is employed to model these scales, a further classification can be made ranging from algebraic (or zero-equation) models to two-equation models.

Reynolds-stress transport models (see e.g. [107]), also referred to as second-order

turbulence-closure models, usually introduce seven additional PDE's to the RANS equations: one for each of the six components of the (symmetric) Reynolds-stress tensor, and one for the turbulent dissipation rate. These models are more sophisticated than the eddy-viscosity models, in the sense that they are in principle capable of dealing with history effects (important e.g. in separated flows) and with more complex turbulent flows (e.g. including strong streamline curvature). However, Reynolds-stress transport models strongly increase the computational cost and the complexity of the flow solver, and they are still in a research stage, at least from the point of view of applicability to flow computations in aircraft industry. Therefore, these models are not further considered here.

Algebraic or zero-equation models

Algebraic models, also called zero-equation models, express the turbulence scales in terms of the mean flow and the distance to the wall (or to the wake-centre line) using algebraic relations. Standard models in CFD literature are the Cebeci-Smith (CS) model [20] and the Baldwin-Lomax (BL) model [6]. Also the Johnson-King (JK) model (sometimes called a half-equation model) may be mentioned here [55, 56]. This model combines an algebraic eddy-viscosity distribution with an ordinary differential equation (in 2D) or a partial differential equation (in 3D) along the configuration surface for the maximum Reynolds shear stress. For these models, the following observations can be made:

- Algebraic models are not 'complete', i.e. they require a priori knowledge of the (turbulence) properties of the flow field [123]. The definition of the turbulence scales will have to be tuned for different types of flow. In fact, the three models here referred to were designed for 2D boundary layers and to some extent for wakes.
- These models have a so-called strong dependency on the geometry, since they use the normal distance to the wall to define the turbulence scales.
- Difficulties in applying these models to more complex flows or configurations are encountered when the boundary layer is no longer well-defined, due to the fact that they use integral properties of boundary layers (CS), maxima within the boundary layer (BL), or even both (JK), to represent turbulence length and velocity scales.
- All three models give reasonable results for flows with attached boundary layers, with weak shocks, and without strong 3D effects (i.e. with approximately 2D boundary layers) (see e.g. [97]).
- For boundary-layer flows with strong adverse pressure gradients or separation, the BL and CS models fail to predict sufficiently accurate pressure distributions. For this type of flow, the JK model gives a strong improvement over these models (e.g. a better shock prediction for the difficult case 10 of the RAE2822 airfoil), but still the results are not fully satisfactory [2, 46, 97].

One-equation models

Traditional one-equation models (e.g. [13, 39]) employ one additional PDE for the transport of the turbulent kinetic energy (k) to define the turbulence velocity scale, while the turbulence length scale is defined by algebraic relations. Thus, these models still have similar deficiencies (in terms of incompleteness, geometry dependence, and difficulties for complex flows or configurations) as algebraic models.

Recent models, in particular the models of Baldwin–Barth [5] and of Spalart–Allmaras [102], however, employ a PDE directly for the eddy viscosity, without introducing additional algebraic relations (for the length scale). Thus, they may be more easily applied to complex configurations, although they still depend on the normal distance to the configuration surface. It seems that these models may be included in RANS flow solvers without a significant increase of numerical stiffness, or computational complexity, compared to algebraic models [57, 97]. Concerning the accuracy of these models, varying results have been reported in the literature. They seem to be successful for e.g. boundary layers with shock-induced separation (both in 2D and 3D) (e.g. [57, 97]), but are less successful for more simple flows [123].

Two-equation models

Two-equation models employ two additional PDE's to define the two turbulence scales, and are therefore in principle complete. The two most popular classes of models are the k - ϵ models and the k - ω models.

In k - ϵ models the turbulent kinetic energy (k) and the turbulent dissipation rate (ϵ) are used as dependent variables of the PDE's. The standard Jones–Launder high-Reynolds-number k - ϵ model [58] has an incorrect near-wall behaviour for boundary layers. Many low-Reynolds number variants have been introduced to correct this behaviour (see for an overview e.g. [87, 122]), usually with wall functions depending on the distance to the wall, but with only limited success. Like the algebraic CS and BL models, k - ϵ models generally give acceptable results for quasi-2D flows with attached boundary layers and weak shocks, but fail for boundary layers with adverse pressure gradients and separation (e.g. [122]).

In k - ω models the turbulent kinetic energy (k) and the specific turbulent dissipation rate ($\omega = \epsilon/k$) are used as dependent variables of the PDE's. The standard k - ω model of Wilcox [121] has two main advantages over k - ϵ models (see [121, 122, 123]):

- Correct near-wall behaviour in boundary layers is obtained without the introduction of wall functions (no geometry dependence).
- Good results have been reported for boundary layers with favourable and adverse pressure gradients.

An important disadvantage of the standard k - ω model is its dependence on the free-stream value of ω [76]. Modifications to resolve this problem have been proposed by Menter [78] and by Wilcox [124], with Menter's modification dependent on the wall distance. Furthermore, although the k - ω model gives better results than the k - ϵ model for boundary layers with adverse pressure gradients, its success for flows with strong shock/boundary-layer interaction, in particular shock-induced separation, is

limited [77, 78]. An important extension of the k - ω model is the so-called shear-stress transport (SST) model of Menter [78]. This extension has been shown to give good results for a number of cases, both 2D and 3D, with strong shock-induced boundary layer separation, competitive with results of the JK model [79]. Another interesting variant of the k - ω model is the k - g model ($g = 1/\omega^2$) which recently has been applied successfully to complete aircraft configurations [59].

Concluding remarks

When we started the extension of the Euler flow solver to a Navier–Stokes solver, it was decided that algebraic turbulence models would be used, in particular the model of Baldwin and Lomax. For turbulent boundary layers around airfoils and wings, this model formed a standard and thus was a good starting point, also considering its cheapness compared to e.g. two-equation models. At that time, more sophisticated turbulence models did not seem to give better results for these types of flow. However, for boundary-layer flows with strong adverse pressure gradients or (mild) shock-induced separation, the prediction of pressure distributions with the BL model was insufficient. Therefore, we later also included the JK model, which gives a strong improvement for this type of flow, as stated before. The mathematical modelling as well as the numerical implementation of the JK model, however, are not straightforward (in particular for 3D boundary layers) as will be seen in section 5.6.

The Navier–Stokes flow solver was mainly developed for simulating the effect of, mostly attached, wing boundary layers on the wing pressure distribution for transport aircraft in subsonic or transonic cruise condition. For this application, algebraic turbulence models can be used. When the flow solver is further extended for more general applications, including e.g. the interaction of boundary layers on separate aircraft components, more sophisticated turbulence models are required. In particular the k - ω models (including the k - g model), which have no explicit geometry dependence, are then good candidates.

5.3 THE CEBECI–SMITH TURBULENCE MODEL

The three algebraic models described here were originally developed for (2D) boundary layers, and therefore explicitly make use of the theory of turbulent boundary layers (e.g. [23, 98]). The boundary layer may be divided into a region next to the wall, called the inner layer, where the turbulence is influenced by the nearby presence of the wall, and an outer layer or defect layer, where this is not the case. These two layers have their own characterizing scales, which may be used to obtain universal velocity profiles (for boundary layers with zero pressure gradients and smooth walls). The outer layer is characterized by the friction velocity $u_\tau = \sqrt{\tau_w/\rho_w}$ (with τ_w the wall shear stress) and the boundary-layer thickness δ , and the tangential velocity u is given by the so-called defect law,

$$\frac{u - u_e}{u_\tau} = f\left(\frac{n}{\delta}\right), \quad (5.1)$$

with n the distance to the wall and u_e the velocity at the edge of the boundary layer. The inner layer is characterized by the friction velocity and the (molecular) kinematic

viscosity $\nu = \mu/\rho$, and the velocity is given by the so-called law of the wall,

$$u^+ = f(n^+), \quad (5.2)$$

with $u^+ = u/u_\tau$ and $n^+ = nu_\tau/\nu$. In the inner layer, a thin viscous sublayer very close to the wall can be distinguished where viscous effects dominate over turbulence effects. In this sublayer, the shear stress $\tau = \mu\partial u/\partial n$ is constant, from which it follows that the velocity must vary linearly as $u^+ = n^+$. The inner and outer layers have an overlap region called the log layer. Matching the defect law and the law of the wall results in a log law for the velocity given by $u^+ = \ln(n^+)/\kappa + 5$ with $\kappa = 0.4$ the Von Kármán constant. The viscous-sublayer and log-layer solutions are valid for $n^+ < 5$ and $n^+ > 30$, respectively, and already have been depicted in figure 3.9.

All three algebraic models use separate formulations for the eddy-viscosity coefficient in the inner layer and in the outer layer. Also, all models use the normal distance n to the wall. Assume that the curvilinear coordinates (ξ, η, ζ) of a block are defined such that the ζ direction is approximately orthogonal to the wall, which is given by $\zeta = 0$. The normal distance is then given by

$$n(\xi, \eta, \zeta) = \int_0^\zeta \frac{d\zeta}{\|\nabla\zeta\|}, \quad (5.3)$$

integrated along a line $\xi = \text{constant}$ and $\eta = \text{constant}$.

In the Cebeci-Smith model [20], the eddy viscosity in the inner and outer layers is determined as follows. Consider the kinematic eddy viscosity ($\nu_t = \mu_t/\rho$) to be the product of a velocity scale (u') and a length scale or 'mixing length' (l'), analogous to the molecular kinematic viscosity ν (i.e. $\nu_t = u' l'$). In the region near the wall (i.e. the inner layer with exception of the viscous sublayer) the velocity scale is determined according to the Prandtl mixing-length hypothesis: $u' = l' |\partial u / \partial n|$, with u the tangential velocity component. Experiments indicate that the mixing length varies linearly with the wall distance in this inner layer, $l' = \kappa n$, with $\kappa = 0.4$ the Von Kármán constant. Assuming the resulting Reynolds shear stress $\tau^R = \mu_t \partial u / \partial n$ to be constant, again the log law is obtained. In the CS model, the inner eddy viscosity is now expressed as

$$\nu_t^{\text{in}}(\xi, \eta, \zeta) = (\kappa n D_d)^2 \|\vec{\omega}\|, \quad (5.4)$$

with the velocity gradient replaced by the magnitude of the vorticity vector $\vec{\omega} = \nabla \times \vec{u}$, and with the mixing length multiplied with the Van Driest damping term D_d . This damping term ensures that the model is also valid in the viscous sublayer, and is defined by

$$\begin{aligned} D_d(\xi, \eta, \zeta) &= 1 - \exp\left(-\frac{n^+}{A^+}\right), \\ n^+(\xi, \eta, \zeta) &= \frac{\rho_w u_\tau n}{\mu_w}, \\ u_\tau(\xi, \eta) &= \sqrt{\frac{\tau_w}{\rho_w}}, \end{aligned} \quad (5.5)$$

with $A^+ = 26$. The subscript w refers to values at the wall ($n = 0$ or $\zeta = 0$). In order to avoid that, in case of (slightly) separated boundary layers, the eddy viscosity

becomes identically equal to zero, the wall shear stress is replaced by the maximum shear stress, $\tau_w(\xi, \eta) = \max_{\zeta} \{\mu \|\vec{\omega}\|\}$. (The maximum shear stress equals the wall shear stress for most attached boundary layers.)

In the outer part of the boundary layer, the Clauser formulation is used to determine the eddy viscosity, which assumes that the velocity scale u' is proportional to the magnitude of velocity at the edge of the boundary layer and that the mixing length l' is proportional to the displacement thickness,

$$\nu_t^{\text{out}}(\xi, \eta, \zeta) = K u_e \delta^* \gamma_K, \quad (5.6)$$

with $K = 0.0168$, $u_e(\xi, \eta) = \|\vec{u}_e\|$ the velocity magnitude at the edge of the boundary layer (given by $n = \delta$ or $\zeta = \zeta_e$), and δ^* the (incompressible) displacement thickness,

$$\delta^*(\xi, \eta) = \int_0^\delta \left(1 - \frac{\|\vec{u}\|}{u_e}\right) dn, = \int_0^{\zeta_e} \left(1 - \frac{\|\vec{u}\|}{u_e}\right) \frac{d\zeta}{\|\nabla \zeta\|}, \quad (5.7)$$

where $\delta(\xi, \eta)$ is the boundary-layer thickness. The Klebanoff function γ_K is included to represent the intermittent behaviour of turbulence at the edge of the boundary layer,

$$\gamma_K(\xi, \eta, \zeta) = \frac{1}{1 + 5.5 (n/\delta)^6}. \quad (5.8)$$

The outer-viscosity formulation depends on the boundary-layer thickness. The determination of this quantity is problematic, especially in a (general) 3D flow, and is discussed in section 5.6.

Finally, the two eddy viscosity formulations must be combined. In the original CS model the eddy viscosity is defined as the minimum of the inner and outer values, but here the following smooth blending is employed [14],

$$\mu_t(\xi, \eta, \zeta) = \rho \nu_t^{\text{out}} \tanh \left(\frac{\nu_t^{\text{in}}}{\nu_t^{\text{out}}} \right). \quad (5.9)$$

For the CS model, but also for the other two algebraic turbulence models, transition from laminar to turbulent flow is obtained by simply setting the eddy viscosity to zero upstream of a specified transition line, and switching the turbulence model instantaneously on at the transition line. In the numerical implementation (section 5.6), this discontinuous eddy-viscosity distribution is smoothed out over a few grid cells.

5.4 THE BALDWIN–LOMAX TURBULENCE MODEL

The Baldwin–Lomax model [6] was developed from the Cebeci–Smith model, with as main aim to avoid the (explicit) calculation of the boundary-layer thickness. This has made the BL model fairly robust and simple to implement, and therefore it is probably the most popular algebraic turbulence model, used in many Navier–Stokes flow solvers. The BL model also includes an eddy-viscosity formulation for wake flows.

Above a solid surface, the eddy viscosity is defined using inner and outer formulations in the same way as in the CS model. However, the outer formulation, containing

the boundary-layer thickness, is modified. For wake flows, no inner layer (influenced by the presence of a solid surface) is present, and thus only an outer-like formulation is used.

The general formulation of the outer viscosity, both for boundary layers and wakes, is given by

$$\nu_t^{\text{out}}(\xi, \eta, \zeta) = KC_{\text{cp}} f_w \gamma_K, \quad (5.10)$$

with $C_{\text{cp}} = 1.6$. The quantity f_w , representing the product of the velocity and length scales, is given by

$$f_w(\xi, \eta) = \min \left\{ f_{\text{max}} n_{\text{max}}, C_{\text{wk}} u_{\text{dif}}^2 \frac{n_{\text{max}}}{f_{\text{max}}} \right\}, \quad (5.11)$$

with $C_{\text{wk}} = 0.25$. The first argument (of the minimum operator) is in particular intended for boundary layers and the second argument for wake flows. The velocity scale f_{max} is the maximum in normal direction of the so-called 'Baldwin-Lomax function' f , given by

$$f(\xi, \eta, \zeta) = n \|\vec{\omega}\| D_d \quad (5.12)$$

(with D_d the Van Driest damping term, which is taken identical to one in wakes). The length scale n_{max} is the location of this maximum. The normal distance n represents either the normal distance to the wall or the normal distance to the centre line of the wake. The velocity scale u_{dif} is defined as the difference between the maximum and the minimum, in normal direction, of the velocity magnitude,

$$u_{\text{dif}}(\xi, \eta) = \max_{\zeta} \{\|\vec{u}\|\} - \min_{\zeta} \{\|\vec{u}\|\}. \quad (5.13)$$

The Klebanoff intermittency function γ_K , equation (5.8), requires the boundary-layer thickness, which is estimated by Baldwin and Lomax as $\delta = n_{\text{max}}/C_{\text{kleb}}$, with $C_{\text{kleb}} = 0.3$.

In a wake flow, the values of f_{max} , n_{max} , and u_{dif} are obtained by taking the maxima and minima along a line passing through the complete wake (thus on both sides of the wake centre). This, together with the absence of an inner viscosity in the wake, may cause the eddy viscosity to be discontinuous near trailing edges. Therefore, the eddy viscosity in the wake is reformulated as a blending of the local eddy viscosity and the trailing-edge eddy viscosity [14]. Let the trailing edge be given by $\xi = 0$, let there be an eddy viscosity $\mu_t^{\text{te}}(\eta, \zeta)$ defined at the trailing edge (i.e. in the plane $\xi = 0$) by the solid-surface formulation, and let $\mu_t^l(\xi, \eta, \zeta)$ be the local eddy viscosity in the wake. The final eddy viscosity is then defined as

$$\mu_t(\xi, \eta, \zeta) = \mu_t^l(\xi, \eta, \zeta) + (\mu_t^{\text{te}}(\eta, \zeta) - \mu_t^l(\xi, \eta, \zeta)) e^{-d(\xi, \eta, \zeta)/B}, \quad (5.14)$$

with d the distance to the trailing edge, and B proportional to a rough estimate of the boundary-layer thickness at the trailing edge,

$$B = 8Re_{\infty}^{-1/5} L. \quad (5.15)$$

5.5 THE JOHNSON-KING TURBULENCE MODEL

The Johnson-King turbulence model was developed to treat two-dimensional, turbulent boundary layers with separation due to strong adverse pressure gradients (such as shocks) [53, 56]. For this type of turbulent boundary layers it is important to take history effects into account. This was achieved through formulating an ordinary differential equation (ODE) along the solid surface for the maximum Reynolds shear stress, which was derived from the equation for turbulent kinetic energy.

In the JK model, a distribution of the eddy viscosity μ_t is defined similar to the CS and BL models (a blending of inner and outer viscosities based on algebraic relations). Such a distribution assumes the turbulence in the boundary layer to be in local equilibrium, so that the eddy viscosity can be directly related to the local mean-flow quantities. Roughly, local equilibrium means that locally the production and dissipation of turbulence (i.e. of the turbulent kinetic energy or of the Reynolds shear stress) are in balance, while convection and diffusion of turbulence are negligible. However, when strong pressure gradients in stream-wise direction are present, this assumption of local equilibrium is no longer valid so that history effects (i.e. convection and diffusion) must be taken into account and the eddy viscosity distribution must be modified accordingly. Johnson and King have done this by requiring that the maximum through the boundary layer of the Reynolds shear stress, modelled as $\tau^R = \mu_t ||\vec{\omega}||$, satisfies a certain ODE along the solid surface. This requirement can be fulfilled by scaling the outer viscosity by an appropriate scaling factor σ . If this scaling factor is equal to one, then we can state that the boundary layer is in equilibrium, and thus this factor indicates to what extent the turbulence deviates from local equilibrium.

The extension of the JK model to three dimensions is not straightforward. Usually, the ODE for the maximum Reynolds shear stress is replaced by a 3D partial differential equation (PDE) covering the complete flow domain (see e.g. [1, 2]). However, the form of this PDE is not accurately defined in the literature, and quite some details remain unclear. We have chosen to use a 2D PDE along solid surfaces, for which the relation with the original ODE is more clear. The numerical solution method for this equation will be described in section 5.6.

In the literature, different modifications to the original JK model can be found. Here, the improvements of Johnson and Coakley [55] are employed. For the outer viscosity, sometimes the Baldwin-Lomax formulation is used, in particular in 3D (as done by Abid et al. [1, 2]), but we use the original Clauser-type formulation with the boundary-layer thickness evaluated as suggested by Johnson [54].

Basically, the JK model introduces five additional dependent variables, which are a function of the curvilinear coordinates (ξ, η) along the solid surface (which is assumed to be given by $\zeta = 0$):

- the variable $g = (\tau_\rho^R)_m^{-1/2}$, with $(\tau_\rho^R)_m$ the maximum in ζ direction of the Reynolds shear stress τ^R divided by density,
- the location of the maximum Reynolds shear stress n_m ,
- the scaling factor of the outer viscosity, σ ,
- the variable $g_{eq} = (\tau_{\rho,eq}^R)_m^{-1/2}$, with $(\tau_{\rho,eq}^R)_m$ the maximum Reynolds shear stress assuming an equilibrium boundary layer, and

- the location of the equilibrium maximum Reynolds shear stress $n_{\text{eq},m}$.

The eddy-viscosity distribution will depend on the first three variables. In particular, the maximum Reynolds shear stress is used as a velocity scale, $u'_m(\xi, \eta) = 1/g$, for the inner eddy viscosity, while as stated before, the outer eddy viscosity is scaled by the factor σ . The last two variables (g_{eq} and $n_{\text{eq},m}$) are used in the PDE for the maximum Reynolds shear stress.

For the additional dependent variables, five relations are needed to close the system of equations. The most important one is the PDE for the maximum Reynolds shear stress (i.e. for g). Four algebraic relations are used to define implicitly the other four variables, one of which states that the factor σ should have such a value that the maximum of the Reynolds shear stress satisfies the PDE.

The JK model will now be described in three parts:

- the eddy-viscosity distribution (defining μ_t , given u'_m , n_m , and σ),
- the PDE for the maximum Reynolds shear stress, and
- the four algebraic relations closing the model.

In the subsequent description, a subscript m will refer to values evaluated at the location of the maximum Reynolds shear stress ($n = n_m$), and a subscript w will refer to values evaluated at the wall ($n = 0$).

The eddy-viscosity distribution

Similar to the CS and BL models, the eddy-viscosity distribution in the JK model consists of an inner and an outer formulation.

The original inner viscosity formulation of the JK model is given by,

$$\nu_{t(\text{JK})}^{\text{in}}(\xi, \eta, \zeta) = \kappa n D_{\text{JK}}^2 \sqrt{\frac{\rho_m}{\rho}} u'_m, \quad (5.16)$$

in which, instead of using the Prandtl mixing-length hypothesis, the velocity scale u'_m is used, which depends on the maximum Reynolds shear stress ($u'_m = 1/g$). The term $\sqrt{\rho_m/\rho}$ is included for compressible flows as suggested in [55]. Also, the definition of the Van Driest damping term follows [55],

$$\begin{aligned} D_{\text{JK}}(\xi, \eta, \zeta) &= 1 - \exp\left(-\frac{\rho_w u_T}{A^+ \mu_w} n\right), \\ u_T(\xi, \eta) &= \max\left\{u_\tau, \sqrt{\frac{\rho_m}{\rho_w}} u'_m\right\}, \end{aligned} \quad (5.17)$$

with $A^+ = 17$. (A value for A^+ different from the CS model is needed, because the inner viscosity behaves as n^3 instead of n^4 .) The friction velocity u_τ is defined such that it is negative in case of (2D) separation,

$$u_\tau(\xi, \eta) = \text{sign}(\tau_w) \sqrt{\frac{\|\tau_w\|}{\rho_w}}, \quad (5.18)$$

where τ_w is the component of the wall shear stress in the direction of the velocity at the edge of the boundary layer,

$$\tau_w(\xi, \eta) = \mu_w \left(\frac{\partial \vec{u}}{\partial n} \right)_w \cdot \frac{\vec{u}_e}{\|\vec{u}_e\|}, \quad (5.19)$$

with the subscript e indicating the edge of the boundary layer (defined as in the Cebeci–Smith model).

In order to improve the JK model for boundary layers with zero or favourable pressure gradients (which was a well-known deficiency of the model, see e.g. Holst [46]), Johnson and Coakley [55] redefined the inner viscosity as a blending of the original formulation and a formulation using the Prandtl mixing-length hypothesis $\nu_{t(\text{ML})}^{\text{in}}$,

$$\begin{aligned} \nu_t^{\text{in}}(\xi, \eta, \zeta) &= (1 - \Psi) \nu_{t(\text{ML})}^{\text{in}} + \Psi \nu_{t(\text{JK})}^{\text{in}}, \\ \Psi(\xi, \eta, \zeta) &= \tanh \left(\frac{n}{L_c} \right), \\ L_c(\xi, \eta) &= \max \left\{ \frac{\sqrt{\rho_w} u_\tau}{\sqrt{\rho_w} |u_\tau| + \sqrt{\rho_m} u'_m}, 0.005 \right\} n_m. \end{aligned} \quad (5.20)$$

The mixing-length inner eddy viscosity ($\nu_{t(\text{ML})}^{\text{in}}$) is equal to the inner viscosity of the Cebeci–Smith model, given by equation (5.4), but with the actual wall shear stress $\tau_w = \mu_w \|\partial \vec{u} / \partial n\|_w$ used in the Van Driest damping term D_d .

For the outer viscosity, two options are considered. In both options, the outer viscosity is scaled by the factor σ . In the first option, the outer viscosity is given by the Clauser-type formulation (as in the Cebeci–Smith model, equation (5.6)),

$$\nu_t^{\text{out}}(\xi, \eta, \zeta) = \sigma K u_e \delta^* \gamma_K, \quad (5.21)$$

with the boundary-layer thickness (needed e.g. to define the displacement thickness, but also for the subsequent PDE) evaluated as suggested by Johnson [54]. This procedure for evaluating the boundary-layer thickness is described in section 5.6. In the second option, the outer viscosity is given by the Baldwin–Lomax type formulation (equation (5.10)),

$$\nu_t^{\text{out}}(\xi, \eta, \zeta) = \sigma K C_{\text{cp}} f_w \gamma_K. \quad (5.22)$$

In this case, the boundary-layer thickness is estimated by $\delta = 1.9 n_{\text{max}} [1]$, where n_{max} is the location of the maximum of the Baldwin–Lomax function. In the original Johnson–King model, the first option (Clauser) was used, which we will also do here. The second option was mainly introduced by Abid et al. [1, 2] to avoid the calculation of the boundary-layer thickness in 3D. However, this formulation was shown by Johnson [54] to give worse results for 2D airfoils.

Finally, the inner and outer eddy viscosity formulations must be combined. A blending function is defined which is slightly different from the one used in the CS and BL models,

$$\mu_t(\xi, \eta, \zeta) = \rho \nu_t^{\text{out}} \left(1 - \exp \left(- \frac{\nu_t^{\text{in}}}{\nu_t^{\text{out}}} \right) \right). \quad (5.23)$$

The PDE for the maximum Reynolds shear stress

The eddy-viscosity distribution as given by the previous equations depends on the maximum Reynolds shear stress through the variable g . In the original Johnson–King model, formulated in 2D, this variable was required to satisfy an ODE along the solid surface, given by

$$u_m \frac{dg}{ds} = RHS, \quad (5.24)$$

with u_m the tangential velocity component at the location $n = n_m$, with s the arc length along the solid surface, and with the right-hand side equal to

$$RHS = \frac{a_1}{2L_m} \left(1 - \frac{g}{g_{eq}}\right) + \frac{C_{dif}}{2(0.7\delta - n_m)} \max\{\sqrt{\sigma} - 1, 0\}, \quad (5.25)$$

in which $a_1 = 0.25$, $C_{dif} = 0.5$ and

$$L_m(\xi, \eta) = \min\{\kappa n_m, 0.09\delta\}. \quad (5.26)$$

The ODE for the maximum Reynolds shear stress had been derived by Johnson and King from the equation for the turbulent kinetic energy k , which under boundary-layer assumptions is given by (at the location $n = n_m$),

$$u_m \frac{dk_m}{ds} = (\tau_\rho^R)_m \left(\frac{\partial u}{\partial n} \right)_m - \epsilon_m + \mathcal{D}_m, \quad (5.27)$$

with the left-hand side representing convection of turbulent kinetic energy, the first term on the right-hand side representing production, ϵ_m representing dissipation, and \mathcal{D}_m representing diffusion. To arrive at equation (5.24), the following modelling steps were made (see in particular [56] for a full derivation):

- assuming a constant ratio of maximum Reynolds shear stress and maximum turbulent kinetic energy, $(\tau_\rho^R)_m/k_m = a_1$, following Bradshaw [13],
- modelling the turbulent dissipation rate using the velocity scale $u'_m = (\tau_\rho^R)_m^{1/2}$ and the length scale L_m as $\epsilon_m = (u'_m)^3/L_m$,
- modelling the normal velocity gradient using a velocity scale based on the equilibrium maximum Reynolds shear stress $(\tau_{\rho,eq}^R)_m$ and the length scale L_m as $(\partial u/\partial n)_m = (\tau_{\rho,eq}^R)_m^{1/2}/L_m$, and
- modelling the diffusion term as

$$\mathcal{D}_m = \frac{C_{dif}(u'_m)^3}{a_1(0.7\delta - n_m)} \max\{\sqrt{\sigma} - 1, 0\}$$

(also a velocity scale to the third power divided by a length scale; for more detail see [53]).

The third step ensures that when production and dissipation are in balance, i.e. in case of equilibrium, then the maximum Reynolds shear stress equals the equilibrium maximum Reynolds shear stress, or equivalently $g = g_{eq}$. The relation in the third step can be related to the Prandtl mixing-length hypothesis, and therefore the length

scale L_m has been modelled analogous to the mixing length, i.e. varying linearly with the wall distance in the inner layer, and proportional to the boundary-layer thickness in the outer layer.

When applying the JK model in three dimensions, the ODE is usually extended to a 3D PDE defined on the complete flow domain [1, 2, 97]. However, the rationale and the precise details of this extension are not made clear. In particular, the maximum Reynolds shear stress, and thus the variable g , is independent of the direction normal through the boundary layer (since it is a maximum in this direction). The same is true for the velocity component u_m . Thus, the introduction of a PDE in which these variables are dependent on the normal direction does not seem logical. Furthermore, when this 3D PDE is discretized in space, the small mesh sizes in boundary-layer normal direction will render the equation more stiff than necessary. On the other hand, a 3D PDE could be solved together with the five basic flow equations as one system of equations, simplifying the numerical method, but this does not seem to be the practice.

To avoid the uncertainties that accompany a 3D PDE, we extend the original ODE to a 2D PDE along the solid surface. Including the time derivative, to facilitate a solution procedure based on time integration, this PDE is given by

$$\frac{\partial g}{\partial t} + (\vec{u}_m \cdot \nabla \xi^*) \frac{\partial g}{\partial \xi} + (\vec{u}_m \cdot \nabla \eta^*) \frac{\partial g}{\partial \eta} = RHS, \quad (5.28)$$

with \vec{u}_m the velocity vector at the location $n = n_m$, and with the same right-hand side as for the ODE. The metric vectors $\nabla \xi^*$ and $\nabla \eta^*$ are independent vectors tangent to the wall (i.e. contravariant base vectors of the tangent space), and are taken as

$$\begin{aligned} \nabla \xi^* &= \frac{1}{J^*} \frac{\partial \vec{x}}{\partial \eta} \times \vec{n}, & \nabla \eta^* &= \frac{1}{J^*} \vec{n} \times \frac{\partial \vec{x}}{\partial \xi}, \\ \vec{n} &= \frac{1}{J^*} \frac{\partial \vec{x}}{\partial \xi} \times \frac{\partial \vec{x}}{\partial \eta}, & J^* &= \left\| \frac{\partial \vec{x}}{\partial \xi} \times \frac{\partial \vec{x}}{\partial \eta} \right\|. \end{aligned} \quad (5.29)$$

Numerical discretization of these equations will only involve differences along the solid surface. Note that the 2D PDE of equation (5.28) combined with equation (5.29) will be valid irrespective of the ζ direction (since ζ does not appear in equation (5.29)).

The first-order hyperbolic PDE of equation (5.28) must be supplemented with boundary conditions. The PDE is defined on a two-dimensional domain (the solid surface) and thus the boundary of this domain is a curved line. Let \vec{m} be the unit vector in the 2D domain that is normal to this curved line, and that is pointing outward of the domain of the PDE. In case of inflow ($\vec{u}_m \cdot \vec{m} < 0$), a Dirichlet condition is applied,

$$g = g_{eq}, \quad (5.30)$$

while in case of outflow ($\vec{u}_m \cdot \vec{m} \geq 0$), no boundary condition should be given. In general, inflow will occur at the transition line and outflow will occur at the trailing edge.

Closure of the model

To close the Johnson–King model, four algebraic relations are needed as was stated at the beginning of this section. First, assume that the Reynolds shear stress is given by $\tau^R = \mu_t \|\vec{\omega}\|$, and consider the maximum of the Reynolds shear stress divided by density,

$$(\tau_\rho^R)_m = \max_\zeta \left\{ \frac{\tau^R}{\rho} \right\}. \quad (5.31)$$

From the definition of the variable g it then follows that the relation

$$(\tau_\rho^R)_m = g^{-2}, \quad (5.32)$$

must be satisfied, where g satisfies the PDE of equation (5.28). This algebraic relation implicitly defines σ , since both μ_t and the PDE depend on σ . However, in practice we found that this relation does not always give realistic solutions for σ . Considerations on this matter are made in section 5.6.4, where equation (5.32) is replaced by equation (5.67).

The second algebraic relation is given by the statement that n_m is by definition the location of the maximum Reynolds shear stress (equation (5.31)). This is also an implicit relation, since the eddy viscosity depends on n_m .

Finally, two relations are needed to define the equilibrium variables g_{eq} and $n_{eq,m}$. Consider an equilibrium eddy-viscosity distribution $\mu_{t,eq}$ given by equations (5.16) through (5.23), with $u_m = 1/g_{eq}$, $n_m = n_{eq,m}$, and $\sigma = 1$, and consider the related equilibrium Reynolds shear stress $\tau_{eq}^R = \mu_{t,eq} \|\vec{\omega}\|$. Then, g_{eq} is defined similarly to equation (5.31) as

$$g_{eq}^{-2} \equiv (\tau_{\rho,eq}^R)_m = \max_\zeta \left\{ \frac{\tau_{eq}^R}{\rho} \right\}, \quad (5.33)$$

while $n_{eq,m}$ is by definition the location of the maximum equilibrium Reynolds shear stress. Again, these are two implicit algebraic relations.

Summary

For the Johnson–King model, we can summarize the complete flow model, consisting of the basic flow equations as well as the PDE and the four implicit algebraic relations, as follows. Consider as the dependent variables the flow-state vector U , and the JK variables g , n_m , σ , g_{eq} , and $n_{eq,m}$. The eddy viscosity depends on the first four variables: $\mu_t(U, g, \sigma, n_m)$. The dependent variables are defined by six equations, consisting of two time-dependent PDE's,

$$\begin{aligned} U &: \frac{\partial U}{\partial t} + R(U, \mu_t(U, g, \sigma, n_m)) = 0, \\ g &: \frac{\partial g}{\partial t} + R^{JK}(U, g, \sigma, g_{eq}, n_m) = 0, \end{aligned} \quad (5.34)$$

and four implicit algebraic equations,

$$\begin{aligned}
 n_m &: n_m &= \tilde{n}_m(U, \mu_t(U, g, \sigma, n_m)), \\
 \sigma &: g^{-2} &= \tau_m(U, \mu_t(U, g, \sigma, n_m)), \\
 n_{eq,m} &: n_{eq,m} &= \tilde{n}_m(U, \mu_t(U, g_{eq}, 1, n_{eq,m})), \\
 g_{eq} &: g_{eq}^{-2} &= \tau_m(U, \mu_t(U, g_{eq}, 1, n_{eq,m})),
 \end{aligned} \tag{5.35}$$

where τ_m and \tilde{n}_m are functions to evaluate the maximum Reynolds shear stress and its location. (For shortness, τ_m denotes $(\tau_\rho^R)_m$.) The solution method for the Johnson–King model in section 5.6.4 will be based on this form of the equations.

Illustration

At this point, we will illustrate some aspects of the functioning of the JK model that will be useful in the subsequent discussion on the numerical implementation. We consider the transonic, turbulent flow around the ONERA M6 wing at the conditions $M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, and $\alpha = 5.06^\circ$ (with transition at 3% of the local chord). The pressure distribution on the upper side of the wing features a lambda shock structure (figure 5.1a). Near the tip, the boundary layer separates due

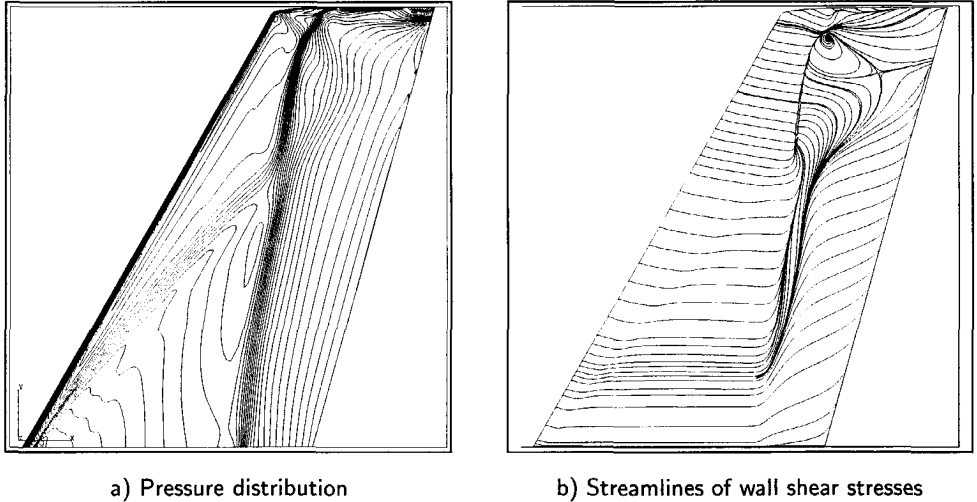
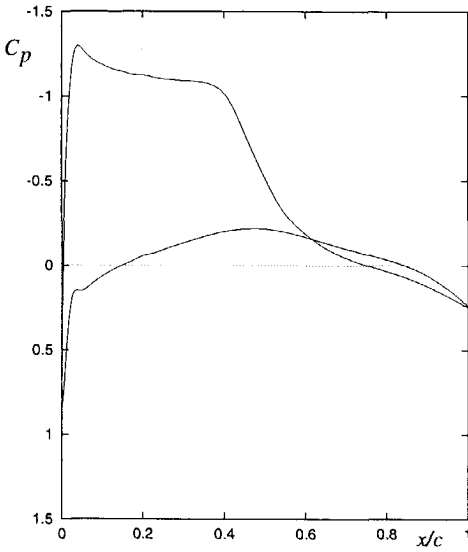
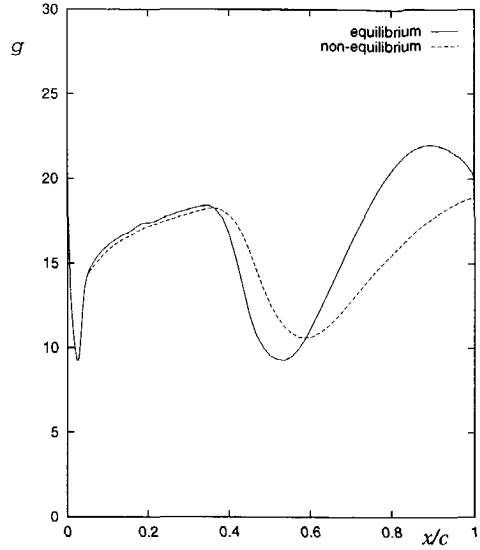


Figure 5.1: Flow solution on upper side of ONERA M6 wing ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$)

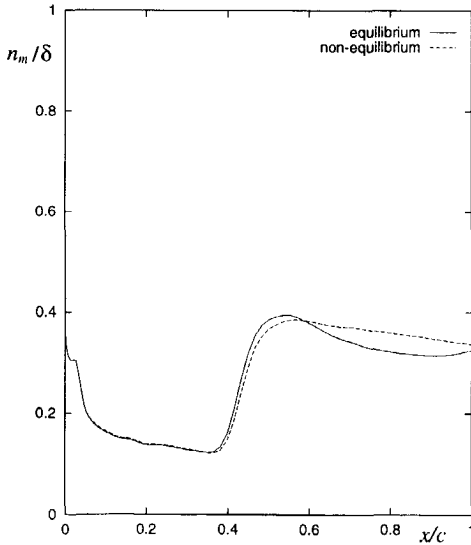
to the presence of a single strong shock as can be seen from the pattern of surface streamlines obtained from the wall shear stresses (figure 5.1b). Figure 5.2 gives the pressure distribution and the distribution of several Johnson–King variables at the span-wise station $\eta = 0.65$ (η being the distance to the root divided by the semi-span). Included are both the non-equilibrium values (g , n_m , and σ), as well as the equilibrium values (g_{eq} , $n_{m,eq}$, while $\sigma = 1$), where the locations of the maximum Reynolds shear



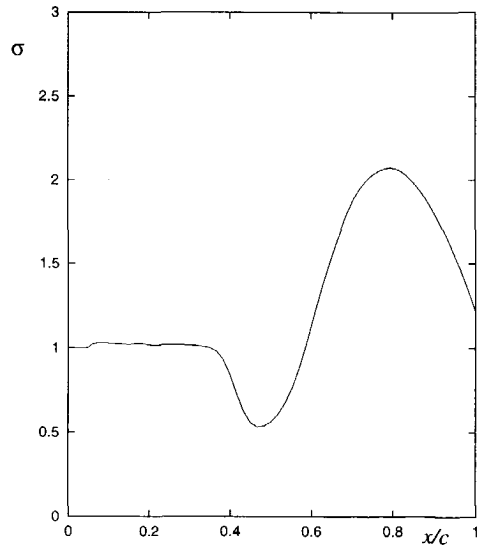
a) Pressure coefficient



b) Johnson–King variables g and g_{eq}



c) Locations of maximum Reynolds-shear stresses



d) Scaling factor of outer eddy-viscosity coefficient

Figure 5.2: Surface pressure distribution and Johnson–King variables on upper side of ONERA M6 wing at span-wise station $\eta = 0.65$ ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$)

stresses have been scaled by the boundary-layer thickness. Upstream of the shock, the Johnson–King model indicates that the boundary layer is in equilibrium ($\sigma \approx 1$), so that the maximum Reynolds shear stress equals the equilibrium value ($g \approx g_{eq}$), while the stream-wise variation is only weak. The maximum stress is here located near the inner layer. Downstream of the shock, there is a strong increase of the maximum stresses (decrease of g and g_{eq}), while the location of the stresses moves more towards the outer part of the boundary layer. These effects are moderated in the non-equilibrium values, due to the transport effects taken into account in the JK model. This moderation can also be seen from the scaling factor σ , which is smaller than one downstream of the shock, reducing the outer eddy viscosity, and thus reducing the maximum stress relative to the equilibrium value. Further aft, the maximum stresses decrease, while again this effect is moderated for the non-equilibrium value, now related to a value of σ larger than one.

5.6 NUMERICAL IMPLEMENTATION

The numerical implementation of the turbulence models described here may at first seem straightforward, since these models consist for a large part of algebraic relations. However, a number of problems are encountered in practice, mainly due to the fact that these models strongly depend on the (2D) boundary-layer concept. Even for a wing with attached boundary layers, there may be regions (such as the tip) where the models are applied outside their application range and may fail to function properly. There, one cannot require more than that the models give reasonable values for the eddy viscosity that do not cause the computation to break down. (Of course, this does not guarantee that the solutions obtained are accurate in such regions.) The Johnson–King model was originally designed for 2D problems, and the extension towards 3D (although formally described in the previous section) also poses a number of problems.

5.6.1 Computation of boundary-layer and displacement thickness

The first problem we will consider is the evaluation of boundary-layer integral quantities, in particular the boundary-layer and displacement thicknesses. These quantities are needed for both the CS and the JK model. The standard method for computing the boundary-layer thickness consists of searching along a line through the boundary layer normal to the surface for the location where the velocity magnitude equals e.g. 95 or 99 percent of the maximum velocity. However, spurious oscillations in the numerical solution can easily cause this method to fail. To avoid this problem, Johnson [54] proposed the following procedure. An estimate of the boundary-layer thickness is obtained by first finding the maximum f_{max} (located at $n = n_{max}$) in ζ direction (the direction normal to the surface) of the ‘Baldwin–Lomax function’ given by equation (5.12), after which the boundary-layer thickness is estimated by

$$\delta(\xi, \eta) = 1.2 n_{1/2}, \quad (5.36)$$

with $n_{1/2}$ the first location beyond $n = n_{max}$ where $f = 0.5 f_{max}$ and $n_{1/2} > n_{max}$.

In the numerical implementation, one must make sure that the boundary-layer and displacement thicknesses are continuous functions of the velocity profile. A smooth change in the velocity profile is not allowed to cause a discontinuous change in these boundary-layer quantities. Violation of this requirement will generally prevent the flow solver from converging. Thus, we compute several quantities by appropriate interpolations between grid cells. Let k be the grid index in normal direction. Then, the distance $n_{1/2}$ is computed by

$$n_{1/2} = n_{i,j,k_{1/2}-1} + \frac{0.5f_{\max} - f_{i,j,k_{1/2}-1}}{f_{i,j,k_{1/2}} - f_{i,j,k_{1/2}-1}}(n_{i,j,k_{1/2}} - n_{i,j,k_{1/2}-1}), \quad (5.37)$$

with $k_{1/2}$ the index of the first grid cell away from the wall with $f \leq 0.5 f_{\max}$,

$$f_{i,j,k_{1/2}-1} > 0.5f_{\max} \geq f_{i,j,k_{1/2}}. \quad (5.38)$$

Similarly, the velocity magnitude at the edge of the boundary layer is computed by

$$u_e = \|\vec{u}_{i,j,k_e-1}\| + \frac{\delta_{i,j} - n_{i,j,k_e-1}}{n_{i,j,k_e} - n_{i,j,k_e-1}}(\|\vec{u}\|_{i,j,k_e} - \|\vec{u}\|_{i,j,k_e-1}), \quad (5.39)$$

with k_e the index of the first grid cell away from the wall with $n \geq \delta$,

$$n_{i,j,k_e-1} < \delta_{i,j} \leq n_{i,j,k_e}. \quad (5.40)$$

Finally, the (incompressible) displacement thickness $\delta_{i,j}^*$ can be computed by

$$\delta_{i,j}^* = \sum_{k=1}^{k_e-1} \left(1 - \frac{\|\vec{u}_{i,j,k}\|}{u_e}\right) (\Delta n)_{i,j,k}, \quad (5.41)$$

with Δn the mesh size in normal direction.

5.6.2 Computation of maxima and their locations

In the BL and JK models, maxima are evaluated along the direction normal through the boundary layer (the ζ direction). Moreover, the locations of these maxima are needed, in particular the location n_{\max} of the maximum of the Baldwin–Lomax function and the location n_m of the maximum Reynolds shear stress. These locations must be defined carefully, for example through linear interpolations as described in the previous section, so that discontinuous jumping of these locations does not prevent the solver from converging. However, such a measure is not sufficient, because in general, the location of a maximum is not unique. For example, it may occur that two local maxima with values close to each other cause an oscillation in the location of the maximum during the time-integration procedure. For 2D boundary layers, for which these turbulence models were designed, generally one unique maximum exists, but there are exceptions, for example local peaks may occur in the BL function close to the wall [22]. Furthermore, we found that in 3D the occurrence of non-unique locations of maxima was so frequent, in particular for the JK model, that in practice the solution procedure would not even start converging.

To resolve this problem, we approximated the location of a maximum by an integral formulation that will always give a unique value, even when the location itself is not unique. The integral formulation for the location n_{\max} of the maximum of the Baldwin–Lomax function, or similarly for the location of the maximum Reynolds shear stress, is given by

$$n_{\max}(\xi, \eta) = \frac{\int_{\zeta=0}^{\zeta_i} \left(\frac{f}{f_{\max}} \right)^q n \, dn}{\int_{\zeta=0}^{\zeta_i} \left(\frac{f}{f_{\max}} \right)^q \, dn}. \quad (5.42)$$

When a unique maximum exists, this formulation will converge to the exact location of the maximum for $q \rightarrow \infty$. It was verified that for $q = 8$, the correct location was obtained for a 2D boundary layer with a clear unique maximum (as illustrated in figure 5.3), while the method was robust in case of non-unique maxima. The upper

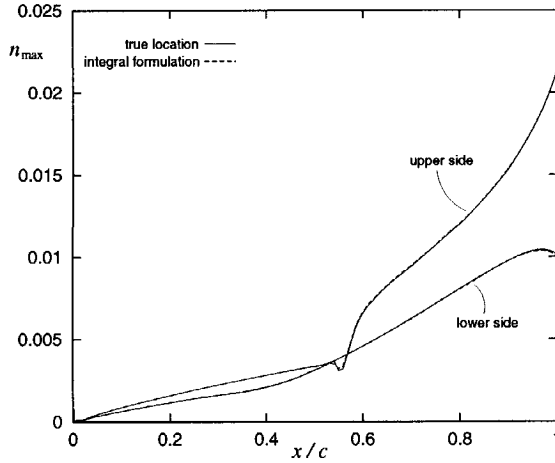


Figure 5.3: Location of maximum of Baldwin–Lomax function for RAE2822 airfoil case9 ($M_\infty = 0.73$, $Re_\infty = 6.5 \cdot 10^6$, $\alpha = 2.8^\circ$)

limit of integration ζ_i is a rough estimate of the boundary-layer edge or simply the boundary of the computational domain. A related idea was used in [22] in which an algebraic turbulence model was derived from both the BL and the CS models using integral formulations based on the BL function.

A further well-known problem for the BL model is that an incorrect location of the maximum of the BL function may be found outside the boundary layer, for example due to the presence of a vortex. Several measures may be taken to resolve this problem. When the flow is dominated by a large vortex structure, as for delta wings, the Degani–Schiff criterion [28] is generally used, which ensures that the correct local maximum nearest to the wall is taken. However, this criterion has some arbitrariness in it (a local maximum is only considered when the function drops to at least 90% of

this local maximum in the direction away from the wall) and is not very robust (e.g. an occasional peak in the BL function very close to the wall may give rise to a low amount of eddy viscosity, destabilizing the computation). For transport-type wings, with only a relatively small tip vortex, the Degani–Schiff criterion has no significant effect on the flow solution away from the tip. Thus, we do not employ this criterion (although it is optionally available in the solver). We do employ a method of [29] for determining a rough estimate of the edge of the boundary layer that is used as an upper bound for the search for the maximum of the BL function. This method consists of determining the boundary-layer edge for a velocity profile corrected for inviscid flow effects.

5.6.3 Transition

The turbulence models are switched on instantaneously at a specified transition line, as stated in section 5.3. In the numerical implementation, however, this discontinuous eddy-viscosity distribution may be problematic, and therefore the eddy viscosity is smoothed out over approximately four grid cells [14],

$$\mu_t = \mu_t^d \left(1 - \exp \left(\frac{-C_{im}(\Delta x)^2}{(\Delta x)_{tran}^2} \right) \right), \quad (5.43)$$

with μ_t^d the discontinuous eddy-viscosity distribution, with the constant $C_{im} = 2.78$, with Δx the distance from the transition line, and with $(\Delta x)_{tran}$ the length of a fixed number of cells ($N_{tran} = 4$) downstream of the transition line.

5.6.4 Numerical method for the Johnson–King model

The Johnson–King model is not a purely algebraic model. It contains a partial differential equation along solid surfaces. The numerical method for approximating this equation will follow the same approach as that followed for the basic flow equations. The unsteady PDE (equation (5.28)) is first discretized in space and subsequently integrated in time until the steady solution is reached. The main differences with the numerical method used for the basic flow equations follow from the facts that the PDE has a non-conservation form and that the PDE is defined along a 2D curved space (the solid surface).

Further, the JK model contains four implicit algebraic relations, for which also a proper solution procedure must be defined.

Space discretization of the PDE

The PDE for the maximum Reynolds shear stress (equation (5.28)) is a linear, first-order, advection equation with variable coefficients and a source term. It has a non-conservation form, so that it cannot be discretized by a finite-volume method as used for the basic flow equations. Instead, we resort to a finite-differencing method. First, we followed a similar approach as for the basic equations: central differencing with explicit fourth-order artificial diffusion. For 2D attached or mildly separated flows, where the advection velocity (u_m) is always of one sign, this resulted

in smooth solutions. For 3D flows around wings, however, the advection velocity field ($\vec{u}_m \cdot \nabla \xi^*$, $\vec{u}_m \cdot \nabla \eta^*$) may contain separation lines, in particular in regions where the 3D flow field separates strongly (e.g. at the tip vortex). At such a separation line, the solution of the PDE will in general be discontinuous, since information is transported to the separation line from opposite directions. The central differencing scheme will then give highly oscillatory solutions. To resolve this problem, we discretized the equations by a second-order fully-upwind scheme. This scheme does not ensure oscillation-free solutions (see e.g. [118]), but in practice we found it to be sufficiently robust for the simple linear advection equation considered here, since the stencil of the fully-upwind scheme does not reach across the separation lines.

Figures 5.4 and 5.5, give an impression of a typical advection velocity field on a wing surface (ONERA M6 wing), and of the corresponding solution g of the Johnson–King PDE. The corresponding pressure distribution on the upper surface was already given in figure 5.1. The velocity field has mainly the same direction as the free-stream velocity, except near the leading edge and near the wing tip. The stagnation line can be recognized on the lower side near the leading edge. The solutions of g on the upper and lower sides are coupled at the leading edge and at the tip, but not at the trailing edge, where the flow leaves the wing. Near the tip, the velocity is directed from the lower side to the upper side, except near the trailing edge. There the velocity is directed towards the tip on both sides (due to the presence of a tip vortex), and therefore a discontinuous solution is found. High gradients in g are present downstream of the transition lines (at 3% of the local chord) and downstream of the shock waves on the upper side, as well as near the tip where shock-induced separation and a tip vortex are present.

Let k be the grid index in the direction normal to the wall (thus, corresponding to the ζ direction). The solid surface is then described by a 2D grid, with the indices (i, j) indicating the cell centers of this grid. The five additional dependent variables (g , n_m , σ , g_{eq} , and $n_{eq,m}$) defined along the solid surface are then functions of the indices (i, j) (corresponding to ξ and η , respectively) for $i = 1 \dots N_i$ and $j = 1 \dots N_j$.

The discretization of the PDE for the maximum Reynolds shear stress with a second-order fully-upwind differencing scheme results in the following equation in each grid cell (i, j) :

$$\begin{aligned} \frac{dg_{i,j}}{dt} + R_{i,j}^{JK} &= 0, \\ R_{i,j}^{JK} &= Q_{i,j}^{JK} - RHS_{i,j}, \end{aligned} \quad (5.44)$$

with the advection term Q^{JK} given by

$$Q_{i,j}^{JK} = \lambda_{i,j}^\xi (g_{i+1/2,j}^+ - g_{i-1/2,j}^-) + \lambda_{i,j}^\eta (g_{i,j+1/2}^+ - g_{i,j-1/2}^-), \quad (5.45)$$

where the values at the cell faces are computed by linear extrapolation from the upwind direction, i.e.

$$\begin{aligned} g_{i+1/2,j}^+ &= \frac{3}{2}g_{i,j} - \frac{1}{2}g_{i-1,j}, \\ g_{i-1/2,j}^- &= \frac{3}{2}g_{i-1,j} - \frac{1}{2}g_{i-2,j}, \end{aligned} \quad (5.46)$$

if $\lambda_{i,j}^\xi \geq 0$, or

$$\begin{aligned} g_{i+1/2,j}^+ &= \frac{3}{2}g_{i+1,j} - \frac{1}{2}g_{i+2,j}, \\ g_{i-1/2,j}^- &= \frac{3}{2}g_{i,j} - \frac{1}{2}g_{i+1,j}, \end{aligned} \quad (5.47)$$

if $\lambda_{i,j}^\xi < 0$, and with similar expressions for the j direction. The values of λ^ξ and λ^η are given by

$$\begin{aligned} \lambda_{i,j}^\xi &= \vec{u}_m \cdot (\nabla \xi^*)_{i,j}, \\ \lambda_{i,j}^\eta &= \vec{u}_m \cdot (\nabla \eta^*)_{i,j}, \end{aligned} \quad (5.48)$$

where $\nabla \xi^*$ and $\nabla \eta^*$ can be easily computed according to equation (5.29).

Because we want to solve the discrete equations by a Runge-Kutta scheme with frozen numerical diffusion (similar to the basic flow equations), we rewrite the advection term Q^{JK} as the sum of central differences and numerical diffusion,

$$Q_{i,j}^{\text{JK}} = Q_{i,j}^{c\xi} + Q_{i,j}^{d\xi} + Q_{i,j}^{c\eta} + Q_{i,j}^{d\eta}, \quad (5.49)$$

with the central differences given by

$$\begin{aligned} Q_{i,j}^{c\xi} &= \lambda_{i,j}^\xi (g_{i+1,j} - g_{i-1,j} - \frac{1}{4}(g_{i+2,j} - g_{i-2,j})), \\ Q_{i,j}^{c\eta} &= \lambda_{i,j}^\eta (g_{i,j+1} - g_{i,j-1} - \frac{1}{4}(g_{i,j+2} - g_{i,j-2})), \end{aligned} \quad (5.50)$$

and with the numerical diffusion given by

$$\begin{aligned} Q_{i,j}^{d\xi} &= \frac{1}{4}|\lambda_{i,j}^\xi|(g_{i+2,j} - 4g_{i+1,j} + 6g_{i,j} - 4g_{i-1,j} + g_{i-2,j}), \\ Q_{i,j}^{d\eta} &= \frac{1}{4}|\lambda_{i,j}^\eta|(g_{i,j+2} - 4g_{i,j+1} + 6g_{i,j} - 4g_{i,j-1} + g_{i,j-2}). \end{aligned} \quad (5.51)$$

The difference equation (5.44) must be supplemented with boundary conditions. As for the basic flow equations this can be done using the dummy-cell concept. In general, the solid surface will be divided into a number of faces, belonging to different blocks. Two layers of dummy cells are added at each boundary of a face: $i = -1$, $i = 0$, $i = N_i + 1$, $i = N_i + 2$, and $j = -1$, $j = 0$, $j = N_j + 1$, and $j = N_j + 2$.

Consider the boundary given by $j = 1/2$. At this boundary, the dummy-cell values are determined depending on the type of boundary.

1. *Trailing edge*: At the trailing edge no continuous boundary condition is present (outward flow), and thus the dummy-cell values are set by extrapolation,

$$\begin{aligned} g_{i,0} &= 2g_{i,1} - g_{i,2}, \\ g_{i,-1} &= 2g_{i,0} - g_{i,1}. \end{aligned} \quad (5.52)$$

These dummy-cell values are used in the central differences and in the numerical diffusion, but cancel out in the total residual due to the upwinding.

2. *Symmetry plane*: If the boundary lies on a symmetry plane, the dummy-cell values are set by reflection,

$$\begin{aligned} g_{i,0} &= g_{i,1}, \\ g_{i,-1} &= g_{i,2}. \end{aligned} \quad (5.53)$$

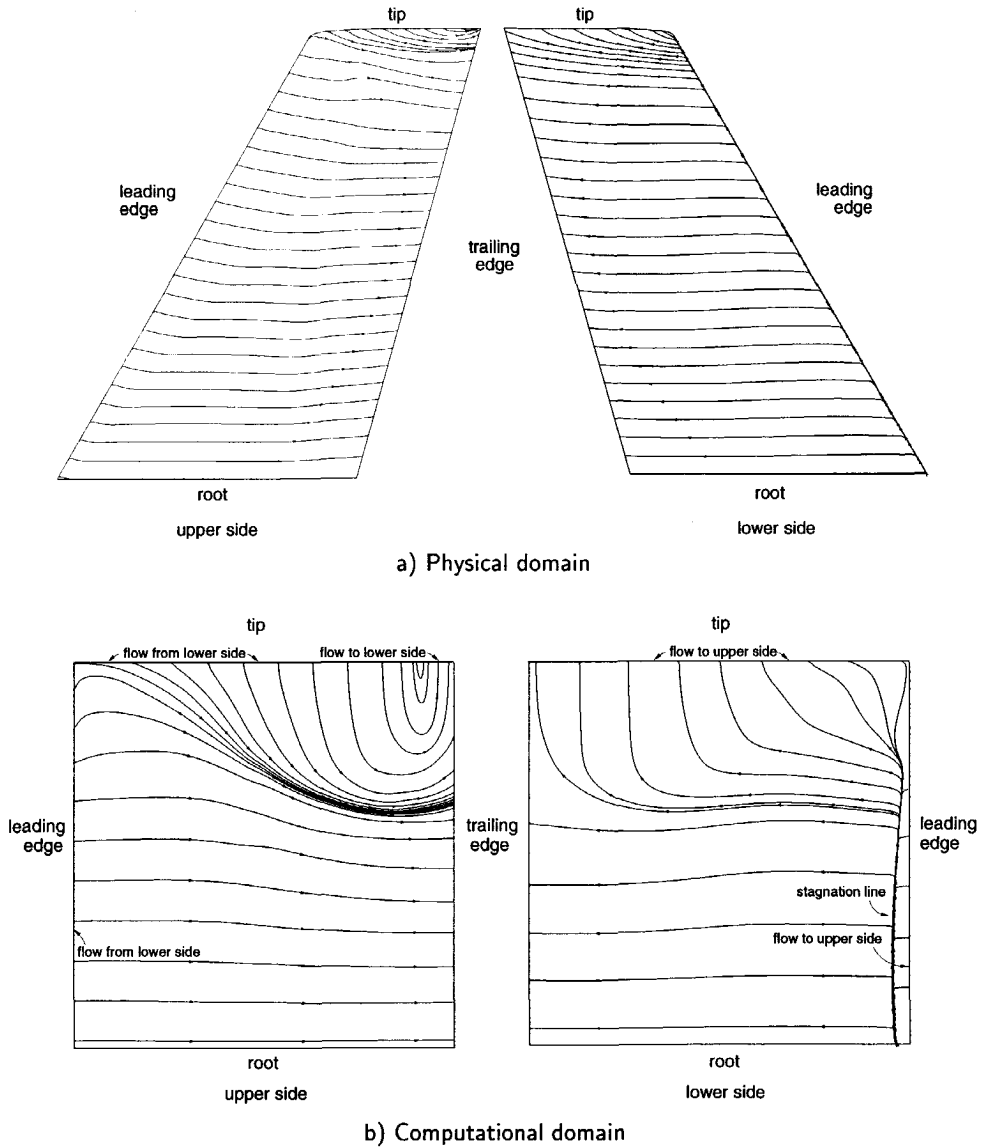
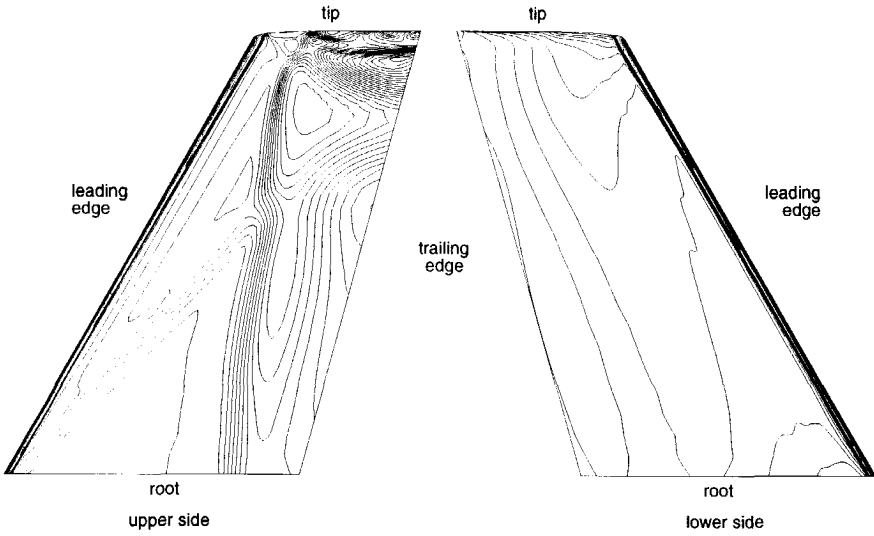
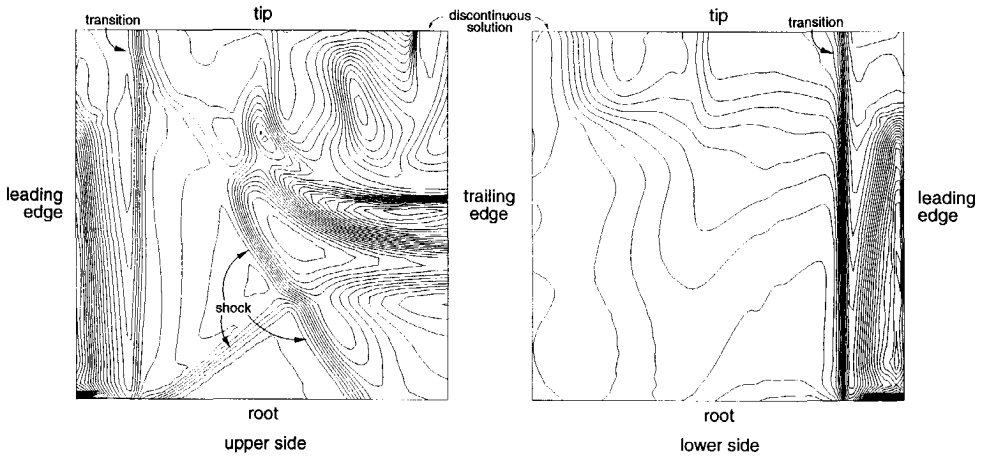


Figure 5.4: Streamlines of advection velocity field for Johnson-King PDE on surface of ONERA M6 wing ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$, 4-block CO-type $256 \times 64 \times 48$ grid)



a) Physical domain



b) Computational domain

Figure 5.5: Distribution of solution g of Johnson–King PDE on surface of ONERA M6 wing ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$, 4-block CO-type $256 \times 64 \times 48$ grid)

3. *Slip solid surface:* When only the boundary layer on the aircraft wing is taken into account, then other configuration surfaces, in particular the surface of the aircraft fuselage, have to be treated inviscidly, i.e. the slip condition for the velocity is applied there. At such boundaries, there is essentially no contribution to the advection term, since the velocity component normal to the boundary is equal to zero. The dummy-cell values are set by extrapolation as for the trailing edge.
4. *Plane of infinite continuation:* If the boundary lies on a so-called plane of infinite continuation (needed to do 2D calculations), then the dummy-cell values are set by copying,

$$\begin{aligned} g_{i,0} &= g_{i,1}, \\ g_{i,-1} &= g_{i,1}. \end{aligned} \quad (5.54)$$

5. *Internal boundary:* At the boundary $j = 1/2$, let the face F be adjacent to another face F_2 that represents a solid surface. The other face has indices i^2 and j^2 . Let the boundary for this face be given by $j^2 = 1/2$. Then, the dummy-cell values are set by copying,

$$\begin{aligned} g_{i,0} &= g_{i^2,1}^{(2)}, \\ g_{i,-1} &= g_{i^2,2}^{(2)}, \end{aligned} \quad (5.55)$$

with superscript (2) indicating values belonging to face F_2 .

Apart from the boundaries of a face, a boundary condition also has to be applied at the transition line (given as $i = i_{\text{tran}}(j)$). In the numerical model, the transition is distributed over three to four grid cells. Therefore, we start the Johnson–King model only a fixed number of grid cells $N^{\text{JK}} = 5$ downstream of the transition line to avoid a strong interference with the numerical transition. Here, g is set using the equilibrium value g_{eq} . This value is also set before this point, so that g is defined over the complete wing surface. Thus, an equilibrium solution is assumed, consisting of

$$\begin{aligned} g_{i,j} &= (g_{\text{eq}})_{i,j}, \\ \sigma_{i,j} &= 1, \end{aligned} \quad (5.56)$$

for $i \leq i_{\text{tran}}(j) + N^{\text{JK}} - 1$, and the difference equation (5.44) is applied for

$$\begin{aligned} i_{\text{tran}}(j) + N^{\text{JK}} &\leq i \leq N_i, \\ 1 &\leq j \leq N_j. \end{aligned} \quad (5.57)$$

If $i_{\text{tran}} \geq 1$ and $N^{\text{JK}} \geq 1$ then the value at the dummy cell $i = 0$ is not needed.

Solution method

For the discretized PDE of equation (5.44), we will use a solution method similar to the method used for the basic flow equations. The PDE will be integrated in time, using a Runge–Kutta scheme, until a steady solution is obtained. For the combination with the solution procedure for the basic flow equations, there are two options:

integrate all equations simultaneously as one system of equations, or integrate the basic flow equations and the Johnson–King PDE by taking separate, subsequent time steps. We have chosen the latter option, because it allows a complete separation of the basic flow solver and the turbulence models, thus simplifying the implementation and allowing an easy replacement of turbulence models. Further, there were indications in the literature [1] that such a procedure would work. If the JK model was defined with a 3D PDE equation instead of a 2D equation, then the simultaneous integration with the basic flow equations could be done more easily. The replacement of the turbulence model with e.g. a two-equation model would then still be possible if the system of equations (and the number of equations) would be defined sufficiently general. However, such an approach would be hindered by details of the Johnson–King model, such as the fact that the advection velocity field in the JK PDE is not equivalent to the velocity field of the 3D flow itself.

We will now present the complete solution method. Consider the basic flow equations, as well as the PDE and the four implicit algebraic relations of the JK model, in the form as summarized at the end of section 5.5 (equations (5.34) and (5.35)). In these equations, the basic dependent variables are the flow-state vector U , and the JK variables g , n_m , σ , g_{eq} , and $n_{eq,m}$.

Consider first only the two time-dependent equations for U and g . As time integration forward Euler is considered here for simplicity, although the equations are actually integrated by Runge–Kutta schemes. Let superscript n indicate the time level. The two equations can be integrated simultaneously by

$$\begin{aligned} \left(\frac{dg}{dt}\right)^n + R^{JK}(U^n, g^n, \sigma^n, g_{eq}^n, n_m^n) &= 0, \\ \left(\frac{dU}{dt}\right)^n + R(U^n, \mu_t(U^n, g^n, \sigma^n, n_m^n)) &= 0, \end{aligned} \quad (5.58)$$

with

$$\left(\frac{df}{dt}\right)^n = \frac{f^{n+1} - f^n}{\Delta t}. \quad (5.59)$$

However, as stated before, subsequent time steps are taken for the two equations, which can be written as

$$\begin{aligned} \left(\frac{dg}{dt}\right)^n + R^{JK}(U^n, g^n, \sigma^n, g_{eq}^n, n_m^n) &= 0, \\ \left(\frac{dU}{dt}\right)^n + R(U^n, \mu_t(U^n, g^{n+1}, \sigma^n, n_m^n)) &= 0. \end{aligned} \quad (5.60)$$

The values of the other four variables (σ , n_m , g_{eq} , $n_{eq,m}$) at time level n are obtained using the four algebraic equations. Each equation involves an eddy-viscosity coefficient, which is defined using values of the four variables at the previous time level. For the variables n_m , g_{eq} , and $n_{eq,m}$ this results in

$$\begin{aligned} n_m^n &= \tilde{n}_m(U^n, \mu_t(U^n, g^n, \sigma^{n-1}, n_m^{n-1})), \\ (g_{eq}^n)^{-2} &= \tau_m(U^n, \mu_t(U^n, g_{eq}^{n-1}, 1, n_{eq,m}^{n-1})), \\ n_{eq,m}^n &= \tilde{n}_m(U^n, \mu_t(U^n, g_{eq}^{n-1}, 1, n_{eq,m}^{n-1})). \end{aligned} \quad (5.61)$$

The evaluation of the variable σ is less straightforward, since it does not appear as left-hand or right-hand side in equation (5.35). We derive a method for evaluating σ by linearizing as follows. Generally, the maximum of the Reynolds shear stress is found in the outer part of the boundary layer, or at least in the part of the boundary layer where the eddy viscosity depends significantly on the outer eddy viscosity (see [56]). Thus, we consider for the eddy viscosity only the outer viscosity, which depends linearly on σ , so that one can write

$$\mu_t(U, g, \sigma, n_m) = \sigma \tilde{\mu}_t(U, g, n_m). \quad (5.62)$$

As a consequence, also the maximum Reynolds shear stress depends linearly on σ ,

$$\tau_m(U, \mu_t(U, g, \sigma, n_m)) = \sigma \tau_m(U, \tilde{\mu}_t(U, g, n_m)). \quad (5.63)$$

Using the second relation of equation (5.35), the variable σ is then found by

$$\sigma^n = \frac{(g^n)^{-2}}{\tau_m(U^n, \tilde{\mu}_t(U^n, g^n, n_m^n))}. \quad (5.64)$$

Since equation (5.62) is only approximately true, we replace $\tilde{\mu}_t$ by the expression $\mu_t(U, g, \sigma, n_m)/\sigma$ (which depends only weakly on σ), but with σ and n_m evaluated at the previous time level. Then, the value of σ^n is found by

$$\sigma^n = \sigma^{n-1} \frac{(g^n)^{-2}}{\tau_m(U^n, \mu_t(U^n, g^n, \sigma^{n-1}, n_m^{n-1}))}. \quad (5.65)$$

Following a somewhat different reasoning, a similar scheme for computing σ has been derived in [1, 2].

It was found that for 2D flows, this scheme for evaluating σ generally worked well. However, for 3D flows we often found that the solution scheme for σ diverged. Analysing the flow solution, we discovered that in these cases, the maximum of the Reynolds shear stress was located near the inner region of the boundary layer (see figure 5.2c of the ONERA M6 wing at station $\eta = 0.65$, where this is the case for $x/c < 0.4$). The eddy viscosity is then strongly determined by the inner formulation, which is independent from σ . Thus, unless unrealistic, high values are used for σ , it is not possible to satisfy the second relation of equation (5.35). Furthermore, the problem mainly occurred in regions where the equilibrium maximum Reynolds shear stress varied only weakly in stream-wise direction (see figure 5.2b), i.e. $dg/ds \approx 0$, suggesting that $g = g_{eq}$ and $\sigma = 1$ would be a reasonable solution for the PDE of equation (5.28). In other words, these are regions where transport effects are not important, and therefore an equilibrium boundary layer may be assumed. Thus, we solved the problem by setting $\sigma = 1$ when the maximum of the Reynolds shear stress is located in the inner layer. To locate the inner layer, the parameter $\alpha_m = (\nu_{t(ML)}^{in}/\nu_{t(eq)}^{out})_m$ is used, which is smaller than 1 in the inner layer, and larger than 1 in the outer layer (given in figure 5.6 for the ONERA M6 wing at station $\eta = 0.65$). The outer viscosity $\nu_{t(eq)}^{out}$ is computed with $\sigma = 1$. To obtain a smooth distribution

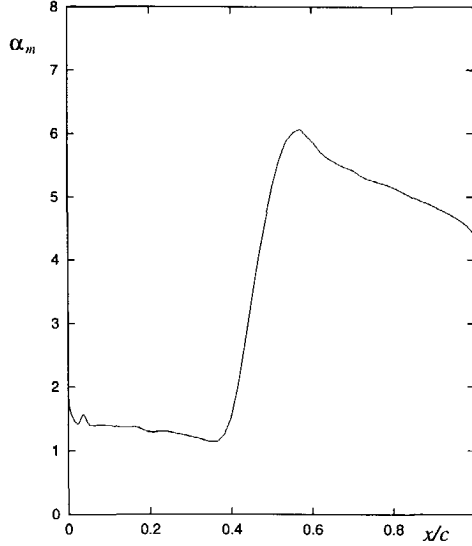


Figure 5.6: Ratio of equilibrium inner and outer viscosities on upper side of ONERA M6 wing at span-wise station $\eta = 0.65$ ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$)

of σ , we define the value of σ at time level n by taking the following two steps:

$$\begin{aligned}\sigma^* &= \sigma^{n-1} \frac{(g^n)^{-2}}{\tau_m(U^n, \mu_t(U^n, g^n, \sigma^{n-1}, n_m^{n-1}))}, \\ \sigma^n &= \frac{\tanh(\alpha_m/\sigma^*)}{\tanh(\alpha_m)} \sigma^*,\end{aligned}\tag{5.66}$$

so that σ^n is computed with equation (5.65) for $\alpha_m \gg 1$ and $\sigma^n = 1$ for $\alpha_m \ll 1$. Substituting $\sigma^n = \sigma^{n-1} = \sigma$, it is seen that at convergence, the following implicit algebraic relation for σ is satisfied,

$$(\tau_\rho^R)_m = \frac{\tanh((\tau_\rho^R)_m g^2 \alpha_m / \sigma)}{\tanh(\alpha_m)} g^{-2},\tag{5.67}$$

which replaces the algebraic relation of equation (5.32) (i.e. the second relation of equation (5.35)).

One time step of the solution method for the Johnson–King model can now be summarized by the scheme outlined in figure 5.7. Note that the eddy viscosity is computed twice: once at step 1, so that no old value of the eddy viscosity is required (and therefore does not need to be stored), and once before the basic flow equations are integrated. Also, an equilibrium eddy viscosity is computed at step 4.

Step 5 actually consists of taking several Runge–Kutta time steps (instead of simply forward Euler). A 5-stage scheme is used, given in reference [70], that was specifically tuned to obtain optimal convergence for the second-order fully-upwind scheme employed here.

$$\begin{array}{ll}
1. \quad \mu_t^+ & = \mu_t(U^n, g^n, \sigma^{n-1}, n_m^{n-1}), \\
2. \quad n_m^n & = \tilde{n}_m(U^n, \mu_t^+), \\
3. \quad \sigma^* & = \sigma^{n-1} \frac{(g^n)^{-2}}{\tau_m(U^n, \mu_t^+)}, \\
& \sigma^n = \frac{\tanh(\alpha_m/\sigma^*)}{\tanh(\alpha_m)} \sigma^*, \\
4. \quad (g_{\text{eq}}^n)^{-2} & = \tau_m(U^n, \mu_t(U^n, g_{\text{eq}}^{n-1}, 1, n_{\text{eq},m}^{n-1})), \\
& n_{\text{eq},m}^n = \tilde{n}_m(U^n, \mu_t(U^n, g_{\text{eq}}^{n-1}, 1, n_{\text{eq},m}^{n-1})), \\
5. \quad \left(\frac{dg}{dt}\right)^n & = -R^{\text{JK}}(U^n, g^n, \sigma^n, g_{\text{eq}}^n, n_m^n), \\
6. \quad \mu_t^n & = \mu_t(U^n, g^{n+1}, \sigma^n, n_m^n), \\
7. \quad \left(\frac{dU}{dt}\right)^n & = -R(U^n, \mu_t^n).
\end{array}$$

Figure 5.7: Solution method for the Johnson–King model

Finally, we must consider how the solution method for the JK model is combined with the multi-grid algorithm used for the basic flow equations. The multi-grid algorithm (combined with other measures) is used to resolve the stiffness in the basic flow equations due to the small mesh sizes in normal direction. Because the Johnson–King equation does not involve the normal direction, this equation is not as stiff as the basic flow equations, and thus there is less need to apply the multi-grid algorithm also to this equation. Therefore, we integrate the Johnson–King equation only on the finest grid level; on the other grid levels the eddy-viscosity coefficient is obtained by restriction. Typically, when 5 Runge–Kutta time steps are taken for every fine-grid relaxation of the basic flow equations, the convergence of the Johnson–King equation is fast enough, for most practical computations, as to not deteriorate the convergence for the basic flow equations.

5.7 RESULTS

In this section, we will verify that the implemented turbulence models give results consistent with results found in the literature.

First, two turbulent, transonic conditions for the RAE2822 airfoil are considered: $M_\infty = 0.73$, $Re_\infty = 6.5 \cdot 10^6$, $\alpha = 3.19^\circ$, and $M_\infty = 0.75$, $Re_\infty = 6.2 \cdot 10^6$, $\alpha = 3.19^\circ$ (resp. cases 9 and 10 of [26], both with transition at 3% of the chord). These two cases were also used in the 1987 AIAA viscous transonic airfoil workshop [24, 46, 61] as well as by other authors [55, 97]. Most authors used an angle of attack $\alpha = 2.8^\circ$ (wind-tunnel correction suggested in [26]), which is also used here. The same grid

is employed as in chapter 3 (i.e. 8-block C-type 528×96 grid), where already some results were presented for case 9 with the BL model.

For case 9, the surface pressure and skin-friction coefficients computed with all three turbulence models are given in figure 5.8, together with the experimental data. The skin-friction coefficient has been scaled by the dynamic pressure at the boundary-layer edge ($C_f = \tau_w / (\frac{1}{2} \rho_e u_e^2)$). The results with the BL and CS models are consistent

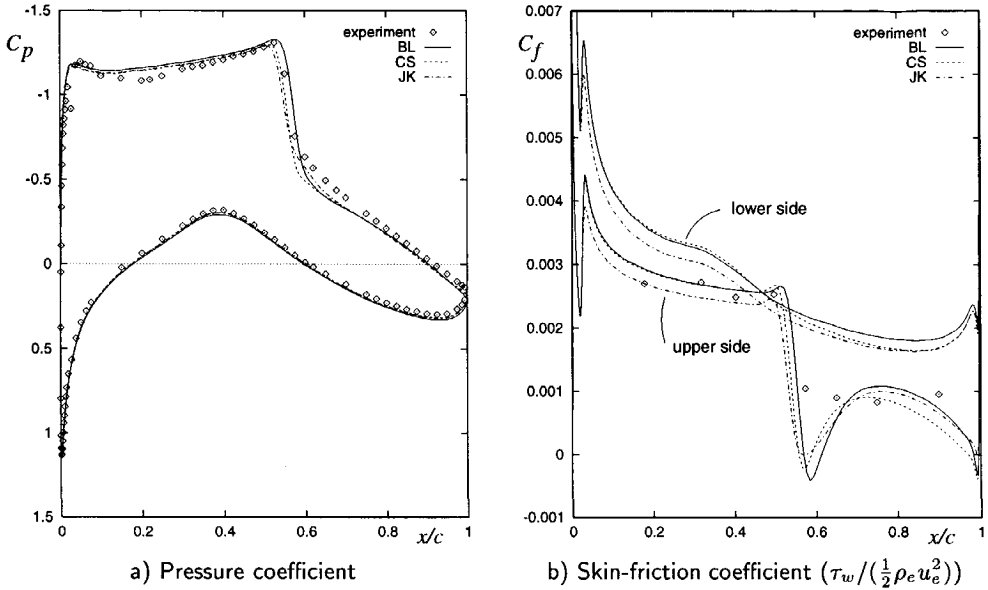


Figure 5.8: Computational and experimental results for RAE2822 case 9 with the Baldwin–Lomax (BL), Cebeci–Smith (CS), and Johnson–King (JK) turbulence models ($M_\infty = 0.73$, $Re_\infty = 6.5 \cdot 10^6$, $\alpha = 2.8^\circ$, 8-block C-type 528×96 grid)

with the results of the workshop, with the CS model predicting the shock slightly more forward than the BL model. At the workshop, it was found that the JK model predicted the shock location too far forward for cases without strong shock-induced separation (as this case 9). Johnson and Coakley [55] modified the inner eddy-viscosity formulation of the JK model, resulting for case 9 in a shock location close to that of the CS model and the experiment, and in a shock strength closer to that of the experiment than is the case for the CS model. The computational results presented here are consistent with these results.

For case 10, the surface pressure and skin-friction coefficients computed with the CS and JK turbulence models are given in figure 5.9, together with the experimental data. This case has strong shock-induced separation. For such a case, the CS model does not yield satisfactory results, predicting the shock too strong and too far downstream. The JK model was designed to deal with strong adverse pressure gradients, and as can be seen, gives a substantial improvement of the pressure distribution. For the BL model, the strong separation is too difficult. (Converged results can only be obtained if the model is modified; typically increasing the constant C_{wk} from 0.25

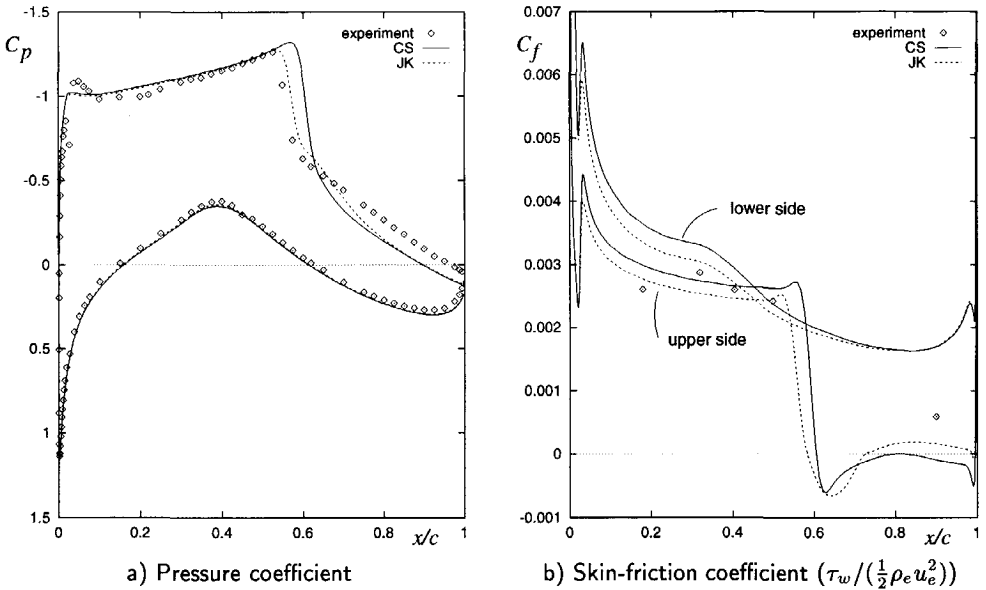


Figure 5.9: Computational and experimental results for RAE2822 case 10 with the Cebeci-Smith (CS) and Johnson-King (JK) turbulence models ($M_\infty = 0.75$, $Re_\infty = 6.2 \cdot 10^6$, $\alpha = 2.8^\circ$, 8-block C-type 528×96 grid)

to 1.0 will do the trick, resulting in a pressure distribution comparable to the CS model, but with the shock location even further aft.) These results are consistent with the workshop results, which showed that all considered turbulence models (for methods based on the RANS equations) predicted the shock location too far aft, with the exception of the JK model, while further all models did not predict the pressure level on the rear part of the upper side correctly.

The application of the turbulence models in 3D is shown for the ONERA M6 wing. A 4-block CO-type grid with $256 \times 64 \times 48$ grid cells has been used. The test case for the ONERA M6 wing ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$, transition at 3% of the local chord) has strong shock-induced separation, and has been used by several authors, e.g. [2, 97]. In the course of this chapter already several figures relating to this case have been presented. Figure 5.10 shows the surface pressure distribution at four span-wise stations (η being the distance to the root divided by the semi-span) computed using the CS and JK models, as well as the experimental data [99]. The convergence history is given in figure 5.11, where the computation with the JK model has been started from a solution obtained using the CS model. As for case 10 of the RAE2822 airfoil, the CS model predicts the shock location too far aft, in particular at the two most outboard stations, where the shock-induced separation is located. Again, the JK model clearly improves the results, giving a shock location and strength reasonably close to the ones found in the experiment. In the literature, different results can be found for this case with the JK model, sometimes similar to our results, and sometimes with the shock further upstream

(see in particular [97]). These differences may be a consequence of different versions of the JK model (e.g. with or without the Johnson–Coakley modification, with the Clauser-type or the BL-type outer-viscosity formulation), while also the extension to 3D may vary. We have chosen a variant of the JK model that gives proper results in 2D for both weak and strong shock-induced separation, while in the extension to 3D we have stayed as close as possible to the 2D model. The comparison with the experiment, and the consistency of the present results with some of the results in the literature gives confidence in our implementation of the JK model in 3D.

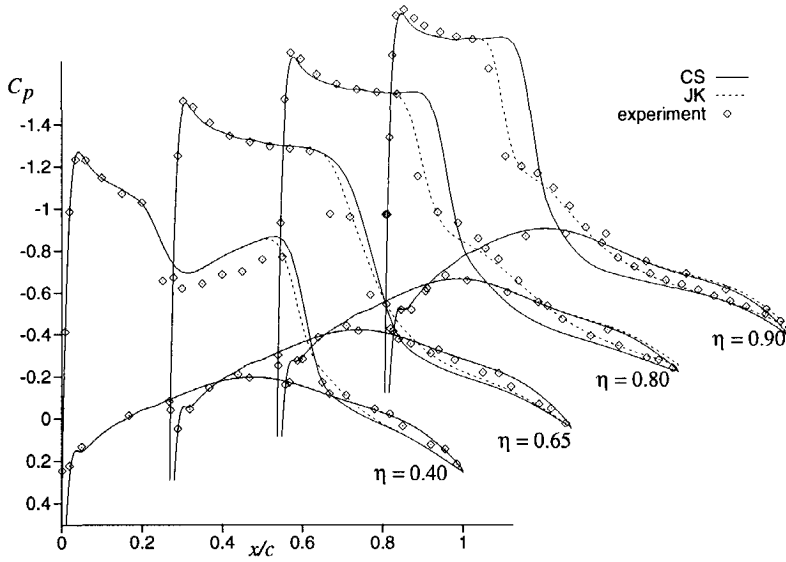


Figure 5.10: Pressure coefficient for ONERA M6 wing ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$, 4-block CO-type $256 \times 64 \times 48$ grid)

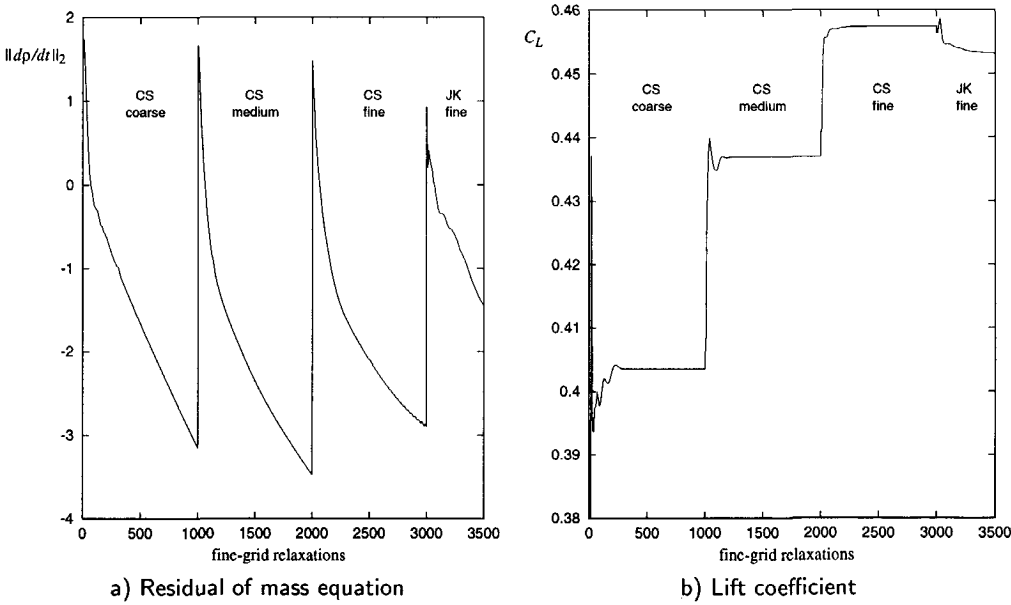


Figure 5.11: Convergence history for turbulent flow around ONERA M6 wing with the Cebeci–Smith (CS) and Johnson–King (JK) turbulence models ($M_\infty = 0.8447$, $Re_\infty = 11.78 \cdot 10^6$, $\alpha = 5.06^\circ$, 4-block CO-type $256 \times 64 \times 48$ grid)

VERIFICATION AND DEMONSTRATION

In this chapter, we will perform a verification of the requirements on numerical accuracy and on efficiency for three cases: RAE2822 airfoil, ONERA M6 wing, and Aerospatiale AS28G wing-body configuration. Further, also the robustness requirement will be considered. Finally, we will make a comparison to both experimental results and results of other computational methods for practical applications (a 2D airfoil and two wing-body configurations with nacelle) to demonstrate the application range and to give an impression of the physical accuracy, but without a full validation.

6.1 VERIFICATION OF NUMERICAL ACCURACY

As a 2D flow case to verify the numerical accuracy, we consider the RAE2822 airfoil, which we have already employed as a standard test case in previous chapters. For this airfoil, we consider the same test case as in chapter 3 ($M_\infty = 0.73$, $\alpha = 2.8^\circ$, and $Re_\infty = 6.5 \cdot 10^6$; case 9 of [26]) and use the Baldwin-Lomax model. Two fine grids are considered (identified as $h = 1$), both with 528×96 grid cells and with the far-field boundary placed at 50 chord lengths from the airfoil, but with different grid-point distributions around the airfoil and generated using different techniques. In particular, the second grid was tuned at the leading edge to improve the prediction of the drag coefficient compared to the first grid. An impression of the first grid was given in figure 3.7. Medium ($h = 2$) and coarse ($h = 4$) grids are obtained by doubling the mesh sizes once and twice, respectively.

In table 6.1, the lift, drag, and friction-drag coefficients are given for the two ranges of grids, together with values for ‘zero mesh size’ ($h = 0$) obtained by Richardson extrapolation from the fine and medium grid values. This extrapolation is done by assuming a second-order accurate scheme. Whether the scheme truly has this order of accuracy may be questioned: in particular it depends on the way in which the discrete solution converges near shock waves, where the space discretization switches locally to second-order artificial diffusion. When the scheme is second-order accurate, it is still the question whether the fine and medium grids are fine enough to let the second-order terms in the truncation error dominate the higher-order terms, and thus allow the extrapolation. The fact that the two different grid sequences give extrapolated values close to each other (a difference of at most 0.3%), however, supports the assumption of second-order accuracy. The extrapolated value itself will be of third (or possibly fourth) order accuracy [93]. In table 6.1, also the relative difference with respect to the extrapolated value is indicated (percentage) in order to assess the numerical accuracy. Thus, we see that the fine grid values have an estimated numerical error

		C_L (error)	C_D (error)	$(C_D)_{\text{fric}}$ (error)
Grid 1	$h = 4$	0.6562	0.02324	0.005564
	$h = 2$	0.8039 (5.3%)	0.01948 (8.2%)	0.005514 (1.4%)
	$h = 1$	0.8380 (1.3%)	0.01836 (2.1%)	0.005457 (0.35%)
	$h = 0$	0.8493	0.01799	0.005438
Grid 2	$h = 4$	0.6191	0.01784	0.005431
	$h = 2$	0.7974 (5.9%)	0.01745 (3.3%)	0.005585 (2.7%)
	$h = 1$	0.8346 (1.5%)	0.01790 (0.83%)	0.005474 (0.68%)
	$h = 0$	0.8471	0.01805	0.005437

Table 6.1: Force coefficients and estimated errors on two ranges of grids for turbulent flow around RAE2822 airfoil (case 9, $M_\infty = 0.73$, $\alpha = 2.8^\circ$, $Re_\infty = 6.5 \cdot 10^6$, 8-block C-type 528×96 grids ($h = 1$))

of at most 2.1%, which falls within the requirement stated in chapter 1 (at most 5%). For the second medium grid, the values have an estimated numerical error of at most 5.9%, just outside the requirement. Thus, a grid resolution between the fine and medium grids is needed to satisfy the requirement, unless further tuning of the grid or automatic adaptation of the grid is performed, in which case the medium grid might be sufficient.

As a first 3D flow case, we consider the turbulent flow around the ONERA M6 wing with the flow conditions $M_\infty = 0.84$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.8 \cdot 10^6$, using the Cebeci–Smith model; a case that also has been considered in chapter 4. A 12-block CO-type grid is employed with $256 \times 64 \times 48$ grid cells. The far field is placed at approximately 10 times the semi-span (distance from root to tip) in vertical direction, 5 times the semi-span in upstream direction, and 3 times the semi-span in downstream and span-wise directions from the wing.

For this case, the force coefficients, given in table 6.2, appear to contain only small numerical errors: less than 1% on the finest grid level. The solution on the medium grid already falls within the accuracy requirement of chapter 1. The ONERA M6 wing, however, is not very representative for transport-type aircraft. It has a low wing aspect ratio and practically no rear loading.

Grid	C_L (error)	C_D (error)	$(C_D)_{\text{fric}}$ (error)
$h = 4$	0.2522	0.01926	0.004579
$h = 2$	0.2693 (2.7%)	0.01721 (2.3%)	0.005000 (1.4%)
$h = 1$	0.2751 (0.69%)	0.01693 (0.56%)	0.005055 (0.35%)
$h = 0$	0.2770	0.01683	0.005073

Table 6.2: Force coefficients and estimated errors on a range of grids for turbulent flow around ONERA M6 wing ($M_\infty = 0.84$, $\alpha = 3.06^\circ$, $Re_\infty = 11.8 \cdot 10^6$, 12-block CO-type $256 \times 64 \times 48$ grid ($h = 1$))

As a test case representative for transport-type aircraft, we consider the AS28G wing–body configuration with the flow conditions $M_\infty = 0.80$, $\alpha = 2.20^\circ$, and $Re_\infty = 10.5 \cdot 10^6$. For this case a body is included, but we compute the boundary layer only on the wing, using the Baldwin–Lomax model. A 32-block CO-type grid is employed

with 1.2 million grid cells in the half space. The far field is placed at a similar distance as for the ONERA M6 wing in terms of the wing span.

The numerical errors in the force coefficients are clearly larger for this case, see table 6.3, in particular that in the drag coefficient, but still fall within numerical accuracy requirement. The largest errors are found for the total drag coefficient, while similar to the previous cases, relatively small errors are found in the friction drag coefficient. The error in the total drag coefficient is likely to be determined for a significant part by the fuselage to which less attention has been paid in generating the grid than to the wing. The error in the lift coefficient may be more indicative of the error in the wing pressure distribution, which is what we are mainly interested in as explained in chapter 1.

Grid	C_L (error)	C_D (error)	$(C_D)_{\text{fric}}$ (error)
$h = 4$	0.4724	0.03339	0.003746
$h = 2$	0.5270 (6.1%)	0.02516 (16%)	0.004168 (1.9%)
$h = 1$	0.5529 (1.5%)	0.02262 (3.9%)	0.004227 (0.47%)
$h = 0$	0.5615	0.02177	0.004247

Table 6.3: Force coefficients and estimated errors on a range of grids for turbulent flow around AS28G wing-body configuration ($M_\infty = 0.80$, $\alpha = 2.20^\circ$, $Re_\infty = 10.5 \cdot 10^6$, 32-block CO-type grid of $1.2 \cdot 10^6$ cells ($h = 1$))

Thus, we may conclude that for the selected cases the required numerical accuracy (less than 5% error in the force coefficients) is obtainable, in particular for a wing-body configuration with in the order of 1 million grid cells, under the assumption that the scheme is indeed second-order accurate. To verify that the scheme is truly second-order accurate, one should consider more grid levels (particularly a finer level, but also intermediate levels, i.e. $h = 3/2$). In such a grid-refinement study for one particular wing-body configuration¹, deviations from second-order accuracy were found, but no further investigation was performed into the sources for these deviations. Further research is required, to obtain more confidence in the numerical accuracy of the flow solver. Nevertheless, we believe that the current results already give a fair indication of the order of magnitude of the numerical errors. As an alternative, one could also take the difference between the medium-grid coefficients and the fine-grid coefficients as an indication for the numerical error in the fine-grid coefficients (in fact this would be the difference between the fine-grid value and the extrapolated value if first-order accuracy is assumed). This error also lies within the 5% norm, with exception of the drag coefficient on the first grid for the RAE2822 airfoil, and the drag coefficient for the AS28G wing-body configuration. The second grid for the RAE2822 airfoil was in particular tuned at the leading edge to obtain a better prediction of the drag coefficient. For the AS28G configuration it should also be possible to improve the drag coefficient through careful tuning of the grid or automatic grid adaptation.

The numerical error of 5% may certainly in some cases be considered to be too large, for example in case of validation of the continuous flow model, or in case of computations for slightly different designs (or with/without nacelle) for which systematic

¹ A study performed by M. Laban, W.J. Piers, and R. Hagmeijer at NLR

errors in the continuous model may be of no importance. From the previous exercise, however, it may be concluded that in order to obtain considerably smaller numerical errors, either very fine grids are needed (say in the order of 8 million grid cells) or very careful tuning of the grid is required (e.g. through automatic grid adaptation).

One point that still needs to be addressed is the placement of the far-field boundaries. Considering a finite domain introduces a numerical error, which decreases as the distance of the far-field boundaries to the configuration increases. In the cases discussed, the far field has been placed at a distance for which the experience is that the errors are smaller than the numerical errors shown here. To be rigorous one should also assess these far-field errors by considering a range of computations with increasing far-field distance. However, the question whether the required numerical accuracy can be obtained is not considered to critically depend on this issue, since the far-field distance can be increased at little expense.

6.2 VERIFICATION OF EFFICIENCY

In chapter 4 we have already seen that for 2D airfoils (528×96 grid), the computation times lie in the order of several minutes. For the ONERA M6 wing ($256 \times 64 \times 48$ grid), the computation time is in the order of 10 minutes if one requires the force coefficients (converged within 10^{-4}) and in the order of 2 hours if one is also interested in the details of the flow solution and therefore requires well-converged residuals. These computation times were obtained on the NEC SX-4 (single processor) at a computation speed in the order of 300 to 400 Mflop/s.

The AS28G wing-body configuration as discussed in the previous section is a more typical application for which the solver is intended, and is therefore a suitable case for verifying the efficiency requirement. Using a grid of 1.2 million grid cells, we performed 1000 fine-grid relaxations (i.e. 100 multi-grid cycles) on the fine grid, requiring 4 CPU hours, and resulting in a mass residual of 10^{-2} and force coefficients converged well within 10^{-6} . Convergence of the force coefficients up to 10^{-4} takes only 1 CPU hour (see table 6.4).

converged within 10^{-4}						mass residual of 10^{-2}
lift	drag	side	pitch	roll	yaw	
270(65)	160(39)	20(29)	280(68)	280(68)	240(58)	1000(242)

Table 6.4: Number of fine-grid relaxations (CPU minutes) performed to obtain a certain level of convergence for turbulent flow around AS28G wing-body configuration on finest grid level ($M_\infty = 0.80$, $\alpha = 2.20^\circ$, $Re_\infty = 10.5 \cdot 10^6$, 32-block CO-type grid of $1.2 \cdot 10^6$ cells)

A full application of the flow solver would consist of a wing-body configuration together with a propulsion system (typically a jet engine). As explained in chapter 1, the Navier-Stokes computation is then limited to simulating the wing boundary layer, while on the other components the slip boundary conditions is applied. In the next section, results for the AS28G wing-body configuration with an engine nacelle will be shown (but computed only on a medium grid level). Including a nacelle in this case

implied a doubling of the number of grid cells (2.4 million). Since the convergence speed is determined by the small mesh sizes inside the wing boundary layer, which remain the same, this increase of grid cells does not affect the convergence speed. Thus, in this case, the computation time is doubled, when including a nacelle, which is still within the efficiency requirement.

We may conclude that the efficiency requirement stated in chapter 1 is satisfied (at most 10 hours) if a grid with a maximum of 3 million grid cells is employed. However, as was seen in the previous section, the grid used for the AS28G configuration has really a minimum grid resolution as far as numerical accuracy is concerned. If significantly finer grids are employed (say 8 million grid cells for a wing-body configuration), the computation times may still be within the efficiency requirement if one requires the force coefficients, but not when one requires the details of the flow solutions.

6.3 VERIFICATION OF ROBUSTNESS

In chapter 1, we required that the computational method must be robust in the sense that converged results must be obtainable without any user interaction (i.e. no tuning of flow-solver parameters) for flow cases within the application range. In the previous chapters, we have presented convergence histories for selected cases (either to discuss properties of the solution procedure or to evaluate the computational efficiency), and each of these computations was performed with the same numerical scheme with the same numerical parameters (except when to demonstrate a property of the scheme). The coefficients of the artificial-diffusion scheme were not adjusted ($k^{(2)} = 1$, $k^{(4)} = 2$, $k^{(s)} = 1/2$, and $k^{(h)} = 2/3$ in 2D, $k^{(h)} = 1/2$ in 3D). For the Navier-Stokes computations, the solution procedure always consisted of a W-cycle multi-grid scheme, with 5 pre- and 5 post-relaxations, using the 5-stage Runge-Kutta scheme as relaxation operator, using variable-coefficient residual averaging with the coefficient $\varepsilon = 1.5$, and using the stability limits $C_{FL} = 4.5$, and $R_K = 1.5C_{FL}$. Also for the computations that will be presented in the next section to demonstrate the application of the flow solver, always the same numerical scheme and parameters were employed, with an exception for the stability limit, which had to be lowered slightly for some of the most complex flow cases ($C_{FL} = 4.0$).

Two remarks should be made. First, all computations were converged at least to what is sometimes called ‘engineering accuracy’ which means that the force coefficients are sufficiently converged (typically within 10^{-4}), while also the residuals have dropped significantly (typically 2 to 3 orders of magnitude compared to the residuals of the initial (free-stream) solution). Thus, it cannot be stated that all computations will converge fully to machine accuracy (which is of course not even feasible for the fine-grid 3D flow cases), and in fact for a few cases this was found not to be true. This could sometimes be attributed to the algebraic turbulence models, which as stated before depend completely on the boundary-layer concept. In regions where the boundary-layer concept may no longer be valid (such as the wing-tip region), these models might occasionally run into convergence problems, even though we have tried to minimize the chance that this will happen as discussed in chapter 5. (Also for applications at (or beyond) the edge of the application range, e.g. flows with strong

shock-induced separation, the turbulence models occasionally do not converge on fine grids.) Especially the Johnson–King model, with its implicit algebraic equations, is an intricate model which is very hard to make completely robust. These kind of convergence problems are usually very local, and it may then be questioned whether it is worth the effort to solve such problems, considering that, even if the computation is converged further, no significantly different flow solution is obtained.

The second remark concerns grid quality. It is true that for the computations presented no tuning of flow-solver parameters was required. However, user interaction was needed in tuning the multi-block grid to ensure convergence (and accuracy). For example, the flow solver will not always converge or converge only slowly for grids with excessive skewness or stretching, for singular multi-block topologies (such as a face that is collapsed to an edge), or for grids with strong changes in grid properties across block faces. Not much effort has been expended on improving the robustness of the flow solver with respect to dealing with such difficult grids. Instead concurrently with the flow solver, also suitable tools for generating high-quality multi-block structured grids have been developed at NLR. Nevertheless, considering that the grid generation is the most time-consuming part of a flow simulation, it could be worthwhile to improve the flow-solver robustness in this respect. (In fact, that is the typical approach for unstructured grids: simplifying the grid generation task at the expense of a more complex flow solver.)

6.4 DEMONSTRATION

To demonstrate the application of the flow solver, we consider three cases for which comparisons can be made to other computational methods and to experimental data.

6.4.1 VTP airfoil

As a suitable 2D flow case, a 2D airfoil is considered that is representative for transport-type wings, i.e. it has significant rear loading and a strong adverse pressure gradient on the aft part of the airfoil. For this airfoil, experimental results are available that have been obtained in the NLR HST wind tunnel, as well as computational results² obtained with the ISES Euler/boundary-layer method [32]. The computational results with the present flow solver ENSOLV have been obtained within the VTP program of the Netherlands Agency for Aerospace Programs NIVR.

A large set of experimental data is available of which 6 cases have been selected: two Reynolds numbers, with for each Reynolds number a subsonic case and two transonic cases (one with a weak and one with a strong shock). The ISES computations have been carried out with the experimental lift coefficient prescribed (with the exception of the first case where the angle of attack used in the experiment was prescribed). The angles of attack resulting from the ISES computations have been used to run ENSOLV. The flow conditions and the resulting lift and drag coefficients are given in table 6.5. For the two subsonic cases, the Baldwin–Lomax model has been used, while for the transonic cases the Johnson–King model has been used. Only for case 5 both models have been employed, in order to demonstrate the importance of using

²Kindly supplied by A.J. Broekhuizen.

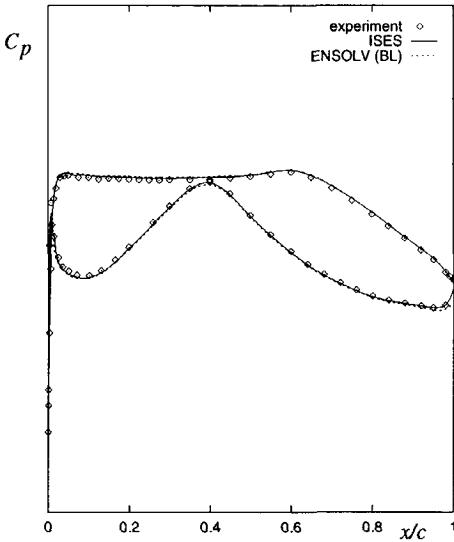
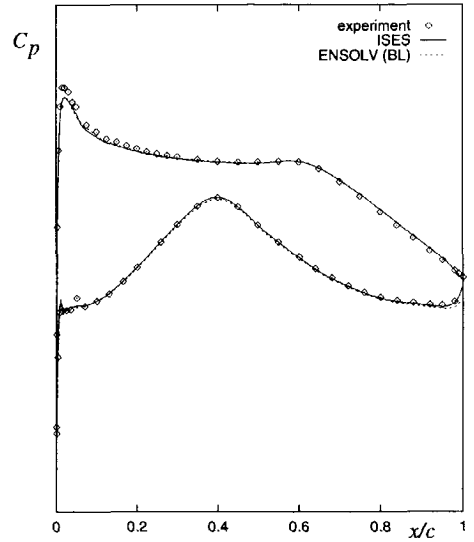
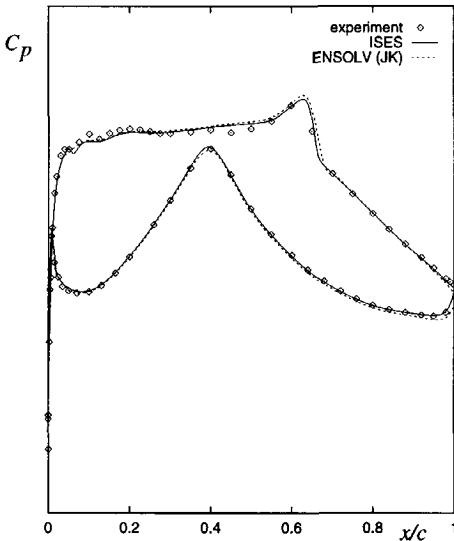
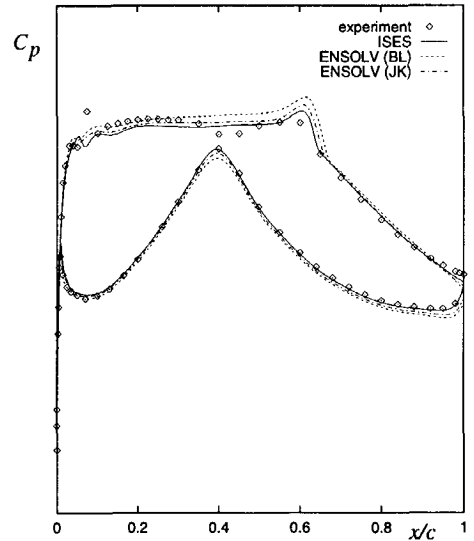
Test case	Model	M_∞	Re_∞	α ($^\circ$)	C_L	C_D
1 subsonic	experiment	0.50	$12 \cdot 10^6$	-0.58	0.345	0.0078
	ISES			-0.59	0.364	0.0076
	ENSOLV (BL)			-0.59	0.375	0.0080
2 transonic weak shock	experiment	0.74	$12 \cdot 10^6$	0.03	0.548	0.0091
	ISES			-0.18	0.545	0.0091
	ENSOLV (JK)			-0.18	0.566	0.0086
3 transonic strong shock	experiment	0.732	$12 \cdot 10^6$	1.69	0.890	0.0155
	ISES			1.56	0.887	0.0145
	ENSOLV (JK)			1.56	0.902	0.0158
4 subsonic	experiment	0.50	$3.2 \cdot 10^6$	1.24	0.536	0.0103
	ISES			0.86	0.535	0.0095
	ENSOLV (BL)			0.86	0.546	0.0103
5 transonic weak shock	experiment	0.74	$3.2 \cdot 10^6$	0.60	0.548	0.0125
	ISES			0.128	0.542	0.0113
	ENSOLV (JK)			0.128	0.574	0.0106
	ENSOLV (BL)			0.128	0.619	0.0112
6 transonic strong shock	experiment	0.772	$3.2 \cdot 10^6$	2.38	0.689	0.0401
	ISES			1.63	0.690	0.0297
	ENSOLV (JK)			1.67	0.719	0.0333

Table 6.5: Flow conditions and coefficients for VTP airfoil

the Johnson–King model for boundary layers with strong adverse pressure gradients. A similar grid as for the RAE2822 airfoil has been used (8-block C-type 528×96 grid), resulting in comparable convergence speed and numerical accuracy.

The experimental and computed pressure coefficients (figures 6.1 to 6.3) correlate satisfactorily for the high-Reynolds-number cases and for the low-Reynolds-number subsonic case (cases 1 to 4), with the ISES result slightly closer to the experiment. For these cases, the differences between the ENSOLV and the ISES lift coefficients vary from 2 to 4%, while the drag coefficients of ENSOLV differ with 5 to 9% from the ISES values and with 0 to 5.5% from the experimental values. Although for a proper comparison with the experiment ENSOLV should be run at the same lift coefficient, these results seem to be in reasonable agreement with results from literature where the accuracy of lift and drag predictions for methods based on the RANS equations for 2D airfoils is reported to be in the order of 3 to 5% for attached or weakly separated boundary layers. The numerical-accuracy requirement specified in chapter 1 has been based on these results from literature.

For the low-Reynolds-number case with a weak shock (case 5) larger differences are found between the three results. The ENSOLV lift coefficient (with the JK model) is higher than the ISES lift coefficient by 6%, while the experimental drag lies considerably above the two computed values (11 and 15%). With the BL model, the ENSOLV lift coefficient is even higher (14% above the ISES value). A number of differences can be found in the pressure distributions (figure 6.2b). The pressure distribution on the upper side upstream of the shock is rather irregular for the experimental results, compared to both computational results. The JK and ISES results lie fairly close: the JK model predicts only slightly lower pressures on the forward

a) Case 1 ($Re_\infty = 12 \cdot 10^6$, $\alpha_{comp} = -0.59^\circ$)b) Case 4 ($Re_\infty = 3.2 \cdot 10^6$, $\alpha_{comp} = 0.86^\circ$)Figure 6.1: Pressure coefficients for VTP airfoil at subsonic conditions ($M_\infty = 0.50$)a) Case 2 ($Re_\infty = 12 \cdot 10^6$, $\alpha_{comp} = -0.18^\circ$)b) Case 5 ($Re_\infty = 3.2 \cdot 10^6$, $\alpha_{comp} = 0.128^\circ$)Figure 6.2: Pressure coefficients for VTP airfoil at transonic conditions with weak shocks ($M_\infty = 0.74$)

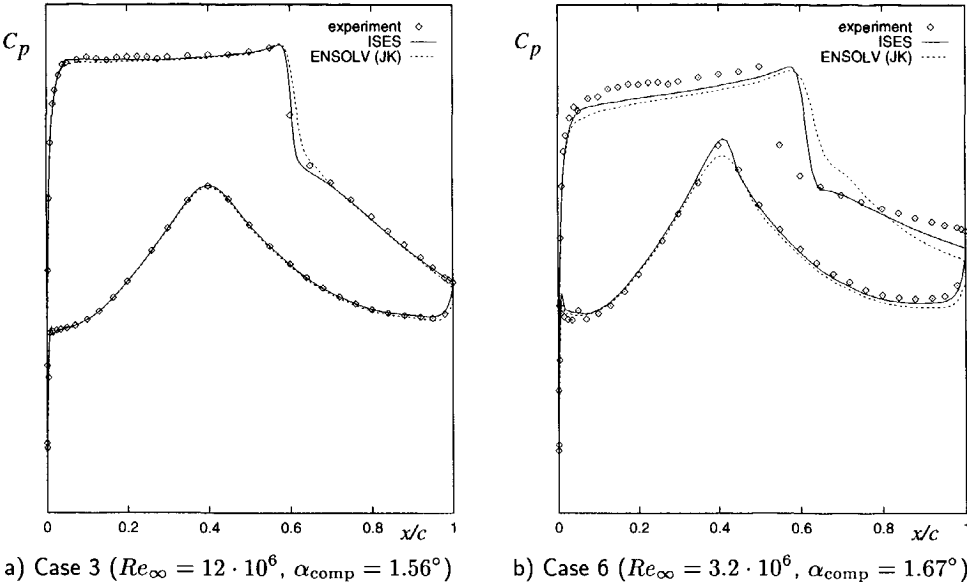


Figure 6.3: Pressure coefficients for VTP airfoil at transonic conditions with strong shocks (case 3: $M_\infty = 0.732$, case 6: $M_\infty = 0.772$)

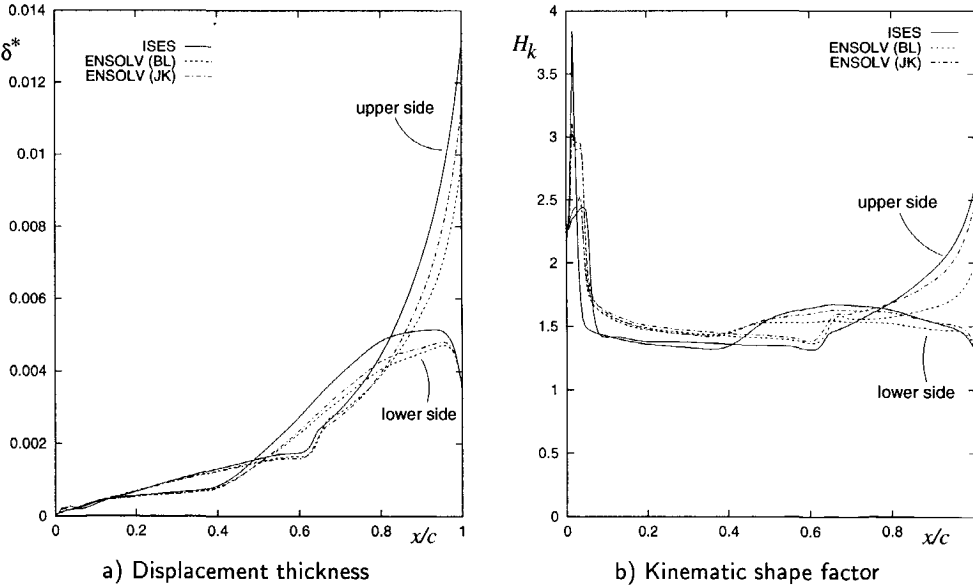


Figure 6.4: Boundary-layer properties for VTP airfoil case 5 ($M_\infty = 0.74$, $Re_\infty = 3.2 \cdot 10^6$, $\alpha_{comp} = 0.128^\circ$)

upper part of the airfoil and higher pressures on the aft lower part (resulting in a 6% higher lift coefficient); similar but less strong differences can be observed for case 2. The BL model, however, predicts different pressure levels almost completely around the airfoil, consistent with the considerably higher lift. These differences may be related to differences in the boundary-layer displacement thickness and (kinematic) shape factor (figure 6.4): on the aft part of the airfoil, the BL model predicts thinner boundary layers, while the shape factor deviates clearly from the JK and ISES results, indicating a weaker response to the strong adverse pressure gradients. The underprediction of the displacement thickness on the upper side near the trailing edge is the reason why a higher lift is found for the BL model: a thinner boundary layer increases the angle between the free-stream velocity and the velocity just downstream of the trailing edge, which corresponds to an increased circulation around the airfoil and thus an increased lift. Thus for this case, we see the importance of taking history effects into account in the turbulence modelling for boundary layers subjected to strong adverse pressure gradients, as is done in the Johnson–King model.

In the computations for the last case considered, case 6, the boundary layer separates strongly at the shock, and does not reattach before the trailing edge. Such a case lies outside the intended application range, and as can be seen from figure 6.3b is clearly too difficult for both computational methods, both predicting the shock too far aft.

6.4.2 *Aerospatiale AS28G wing-body-nacelle configuration*

As an illustration of the application of the flow solver to a transport-type aircraft configuration, the Aerospatiale AS28G wing-body configuration with and without a flow-through nacelle is considered. We have already considered the numerical accuracy and the computational efficiency for this case in the previous sections.

For the wing-body configuration, the computed pressure distributions with the Baldwin–Lomax and Johnson–King turbulence models are given in figure 6.5 together with the experimental results of Aerospatiale (at the conditions $M_\infty = 0.80$, $Re_\infty = 10.5 \cdot 10^6$, $\alpha = 2.20^\circ$, employing a 32-block grid with 1.2 million grid cells). As for the VTP airfoil, this case has a strong adverse pressure gradient on the rear part of the upper wing surface. As a consequence the two turbulence models predict different boundary-layer thicknesses at the trailing edge on the upper side, resulting in a lower circulation around the wing for the JK model. This leads to a more forward prediction of the shock location, as well as to a lower level of rear loading. Comparison with the experimental results is not fully satisfactory, in particular on the upper side of the wing. These differences are not further investigated here. Such investigations could be part of a validation of the physical accuracy of the flow solver, requiring a careful checking of the wind-tunnel conditions (and corrections) and of the geometries employed in the experiment and computation, as well as comparisons with other computational methods.

For the configuration including the nacelle, the computed and experimental pressure distributions are given in figure 6.6 (at span-wise stations located near the nacelle), together with pressures for the configuration without nacelle. In the experiments a pylon was included. The computations were performed on medium grids

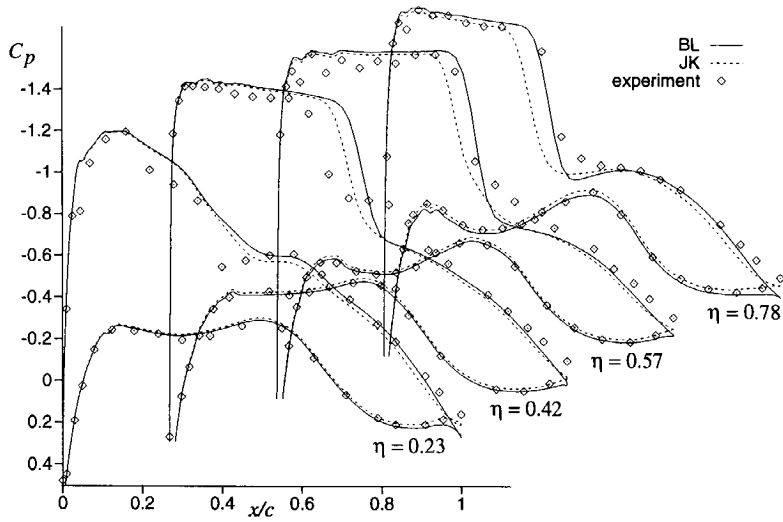


Figure 6.5: Pressure coefficients for AS28G wing-body configuration with Baldwin-Lomax (BL) and Johnson-King (JK) turbulence models ($M_\infty = 0.80$, $Re_\infty = 10.5 \cdot 10^6$, $\alpha = 2.20^\circ$)

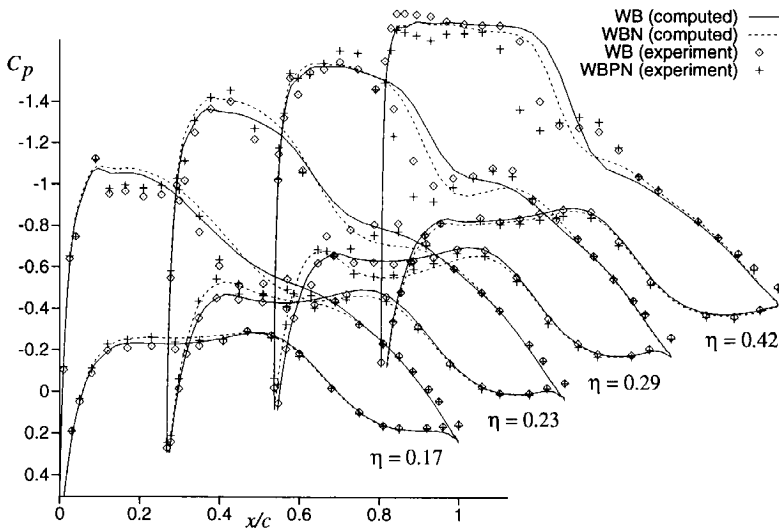


Figure 6.6: Pressure coefficients for AS28G wing-body and wing-body-nacelle configurations ($M_\infty = 0.80$, $Re_\infty = 10.5 \cdot 10^6$, $\alpha = 2.20^\circ$)

(with nacelle: 101 blocks with 0.28 million grid cells, and without nacelle: 32 blocks with 0.15 million grid cells) and with the BL turbulence model. An impression of the grid is given in figure 6.7. The effect of the nacelle on the wing pressure distribution

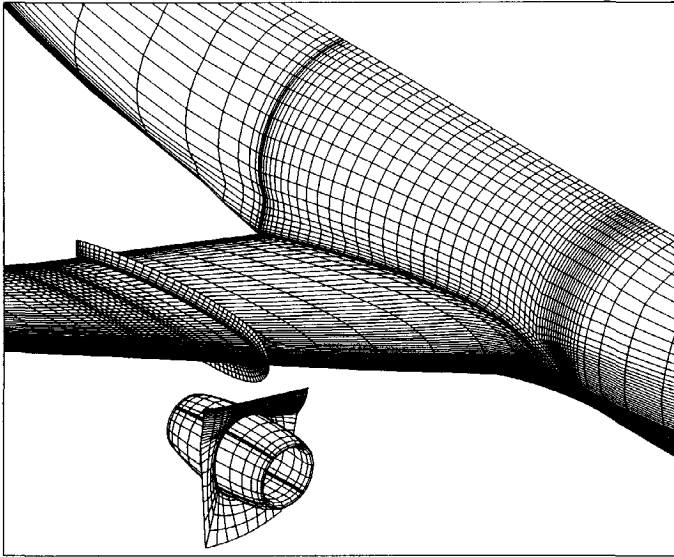


Figure 6.7: Impression of grid around AS28G wing-body-nacelle configuration (medium grid: 101 blocks, 0.28 million grid cells)

is predicted in a qualitatively similar matter by the experiments and computations: in particular the modification of the lower-side pressure distribution and the forward movement of the shock. This, despite the fact that only medium grids are employed and despite the differences between computed and experimental results on the upper wing as noted before. Although more computations for similar cases would be required to be certain, the results give the impression that for a qualitative evaluation of the effect of for example the nacelle position on the wing pressure distribution, one may accept significant numerical errors in the computations, as well as uncertainties in the physical accuracy, suggesting that these errors are of a systematic nature (and thus can be corrected for).

6.4.3 VTP wing-body-engine configuration

Finally, we present some results³ for a wing-body transport-aircraft configuration at design conditions, obtained within the VTP program of the NIVR. In this case not only the engine nacelle is taken into account, but also the boundary conditions for simulating the effect of the jet-engine inlet and outlet are employed. For the computations without the engine, comparisons are made to experimental results obtained

³All computations presented in this section (both ENSOLV and MATRICS-V) were performed by M. Laban, NLR.

in the NLR HST wind tunnel and to computational results obtained with the NLR MATRICS-V full-potential/boundary-layer method [81, 119] at the flow conditions $M_\infty = 0.78$, $Re_\infty = 7.4 \cdot 10^6$, and $C_L = 0.57$ (transition at 5% of the local chord). Again, only the boundary layer on the wing is simulated in the ENSOLV computation, which is also the case for the MATRICS-V method. The two computed pressure distributions (no engine), presented in figure 6.8, lie reasonably close, in particular on the inboard part of the wing. On the outboard part, ENSOLV predicts the

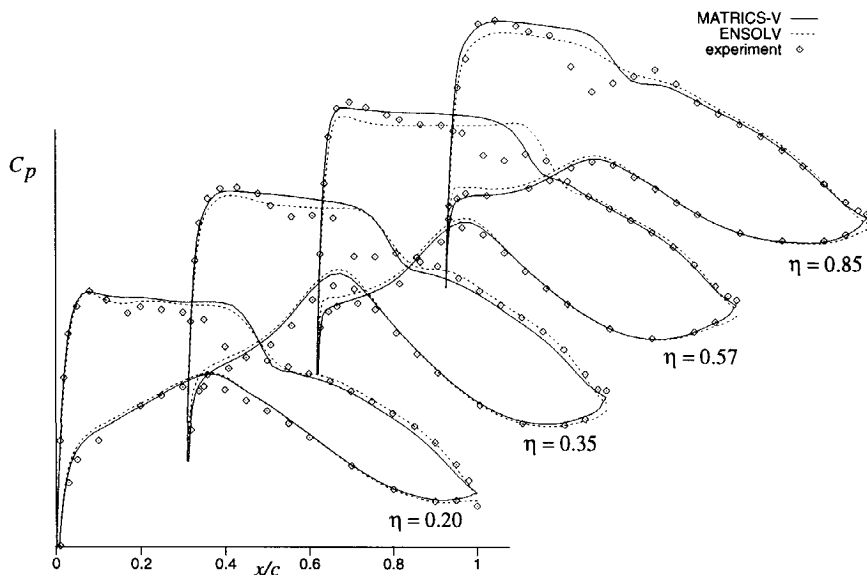


Figure 6.8: Pressure coefficients for the VTP wing-body configuration ($M_\infty = 0.78$, $Re_\infty = 7.4 \cdot 10^6$, $C_L = 0.57$)

shock slightly further downstream (which seems to resemble some of the results obtained for the VTP airfoil). As before, it was important to employ the Johnson–King turbulence model. Although not shown, the Baldwin–Lomax model predicted a significantly different pressure distribution, similar to the differences found for case 5 of the VTP airfoil. The experimental results clearly differ from the two computational results, with the shock predicted significantly further upstream. This seems to be attributable to the way in which the experimental results were corrected for tunnel-wall interference effects, which also included a correction for different fuselages.⁴

As explained in chapter 1, the intended application of ENSOLV is to assess the influence of in particular the engine location on the wing pressure distribution. Such computations can be part of a design procedure in which, given a wing-body configuration with a desired pressure distribution, one tries to restore this pressure distribution.

⁴M. Laban reported that taking into account the differences in fuselage shapes, MATRICS-V gave pressure distributions close to the experimental results.

bution for the configuration including the engine nacelle (and possibly the pylon as well) by redesigning the wing. In figure 6.9, the upper-wing pressure distributions are shown for different steps in such a design procedure for the VTP configuration: original wing-body configuration, original wing-body-nacelle configuration, and wing-body-nacelle configuration with redesigned wing. Observe that the redesigned wing with nacelle has about the same pressure distribution as the original wing without nacelle; hence, the wing-geometry correction was successful.

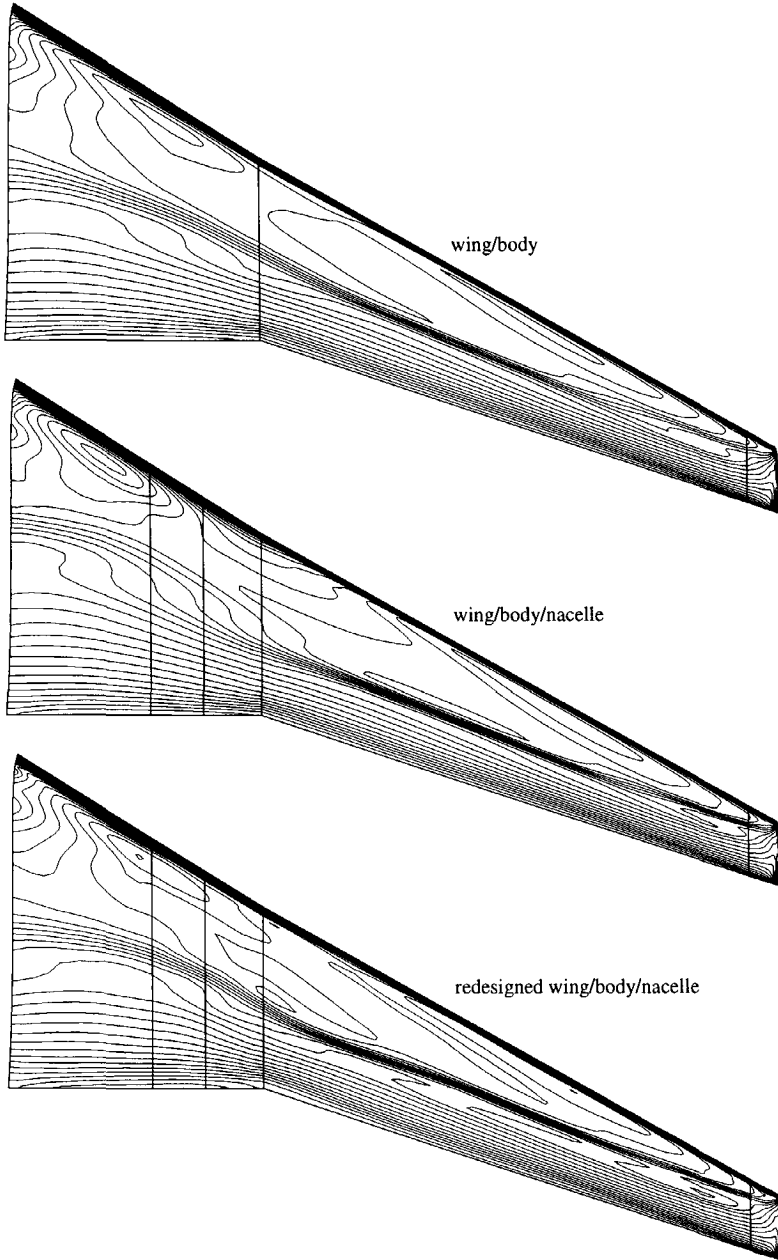


Figure 6.9: Lines of constant pressure on VTP upper wing at three subsequent steps during a design procedure to correct for nacelle effects ($M_\infty = 0.78$, $Re_\infty = 7.4 \cdot 10^6$, $C_L = 0.57$)

CONCLUSIONS AND FINAL REMARKS

In this thesis, we have discussed the development of a Navier–Stokes flow solver, called ENSOLV, intended for the simulation of steady, compressible, turbulent flow around transport-type aircraft in cruise conditions. The main application consists of the simulation of flows around wing–body configurations combined with propulsion systems such as jet engines. Of primary interest is the effect of the presence of the propulsion system on the wing pressure distribution. The Navier–Stokes flow solver was developed from an existing Euler solver. Viscous and turbulence effects were incorporated by means of the thin-layer Reynolds-averaged Navier–Stokes equations and algebraic turbulence models, aiming at the simulation of the wing boundary layer and wake only, while maintaining growth potential towards more complex viscous flows.

If a computational method is to be applied routinely in an industrial environment, it must be sufficiently efficient and robust, and have sufficient accuracy. In chapter 1, we have formulated requirements with respect to these properties, particularly focusing on the numerical aspects of the flow solver. Considering at first a straightforward extension of the Euler flow solver to the TLNS equations, further improvements were needed to satisfy these requirements. The following conclusions can be drawn:

- *Space discretization:* The basic discretization scheme consisted of a cell-centred finite-volume scheme using central differences with explicit artificial diffusion and employed multi-block structured grids. In order to reduce the strong grid dependency of the numerical solution in boundary layers, it has been necessary to replace the scalar form of artificial diffusion by a matrix form, based on the ideas of Swanson and Turkel [111]. However, we have modified the matrix artificial-diffusion scheme so that it is effective in boundary layers only, while avoiding a significant reduction of convergence speed or increase of computation time compared to the scalar form.
- *Solution procedure:* The basic solution procedure consisted of explicit time integration using Runge–Kutta schemes, accelerated by local time stepping and implicit residual averaging. To obtain an acceptable convergence speed for the Navier–Stokes equations, we have extended the solution procedure with a multi-grid scheme. Furthermore, the high aspect-ratios of the grid cells, typical for high Reynolds-number boundary layers, had to be accounted for in the residual-averaging coefficients and the scaling factors of the artificial diffusion, following the ideas of Martinelli [73]. On multi-block grids, we have obtained a robust

scheme using the concept of multi-block inside multi-grid and employing a W-cycle multi-grid scheme with five pre- and five post-relaxations. It has been shown that multi-block and single-block grids give similar convergence speeds. In particular the convergence rate of the force coefficients is strongly improved by the multi-grid scheme.

- *Turbulence modelling:* The algebraic turbulence models of Baldwin–Lomax, Cebeci–Smith, and Johnson–King have been considered. Mainly because these models strongly depend on the 2D boundary-layer concept, it is not straightforward to obtain a robust implementation. In particular for the Johnson–King model, we have analysed detailed problems, related to the extension from 2D to 3D and to its implicit algebraic relations. The investment in implementing the Johnson–King model, however, is worthwhile, considering it improves the prediction of the lift considerably for airfoils and wings with strong adverse pressure gradients compared to the other algebraic models.

In chapter 6, we have verified whether the requirements formulated in chapter 1 could be met. Considering a typical application, consisting of a transport-type wing-body configuration including an engine nacelle, sufficiently converged solutions (force coefficients well within 10^{-4}) can be obtained on grids of 2 to 3 million grid cells within the required computation time (at most 10 CPU hours assuming a computation speed of 400 Mflop/s). Employing Richardson extrapolation, it was inferred that a minimal numerical accuracy (numerical errors in the force coefficients within 5%) is attainable on such grids. A more detailed assessment of the numerical accuracy may be needed if higher accuracies are required.

With respect to robustness, we have drawn the conclusion that for typical applications, as those employed for demonstration purposes, sufficiently converged solutions (force coefficients within 10^{-4}) can be obtained without variation of the numerical scheme or its parameters, provided that a high quality grid is employed. This means that user interaction is mainly required in generating such a grid and not in fine-tuning solver parameters.

In the future, the constraints imposed on the development of the flow solver, particularly the region of applicability, may change. For transport aircraft, future applications may be accurate drag prediction, which requires higher numerical accuracy, and the simulation of high-lift devices (take-off and landing conditions), which involves more complex flows. Furthermore, other application areas are currently being considered as well. Military-aircraft applications involve highly complex configurations and flows. In particular, the flow solver has already been used for the simulation of the flow around delta wings, involving strong vortices [15]. Spacecraft applications involve flows at supersonic or hypersonic conditions, possibly leading to strong shocks, and complex flows at the base of spacecraft.

Different applications generally lead to changing requirements, such as higher numerical accuracy for accurate drag prediction. For these further developments of the flow solver, the following may be considered:

- In order to extend the flow solver to more general flows, currently the full

Reynolds-averaged Navier–Stokes equations are also included, as well as two-equation turbulence models, which are more generic than algebraic models.

- Dependency on grid resolution in boundary-layer normal direction has been reduced by employing matrix artificial diffusion. The solver still shows significant dependence on the grid resolution in tangential directions, for example at shock waves and near wing leading edges. A successful approach to improve the grid in these regions is automatic grid adaptation as developed by Hagmeijer [42]. Also, upwind-type schemes may be considered, in particular to improve the numerical accuracy at shock waves.
- The form of the current matrix artificial diffusion is suitable when there is one dominant viscous direction. If all viscous regions around an aircraft configuration are to be simulated, this may no longer be true, and the matrix diffusion would have to be reconsidered.
- The scaling of the artificial diffusion for high-aspect-ratio grid cells, which was needed to obtain acceptable convergence speeds, may locally introduce high numerical errors. If the grid is not carefully generated, this may suppress local details such as trailing-edge separation.
- For supersonic flows with strong shocks, the current artificial-diffusion scheme is not satisfactory. Either upwind schemes will have to be used, or extensions as proposed by Jameson [49] will have to be applied.

To allow for future extensions as just described, a flow solver should have an appropriate software structure. At NLR, also research has been done recently into the question how to obtain a structure of CFD software that supports characteristics as adaptability and testability, as discussed by Vogels [117].

Experience with the flow solver ENSOLV at industry, as illustrated by some of the presented computations, has shown that the solver in its current form is very well applicable in practice.

APPENDIX A

FLUX JACOBIAN FOR THE EULER EQUATIONS

Both in the boundary condition at the far field and in the matrix artificial diffusion, the Jacobian of the convective flux is used. Also, the eigenvalues and the eigenvectors of the Jacobian are used. In this appendix, these are all derived.

First, quasi-linear forms of the Euler equations in curvilinear coordinates will be derived, both in terms of the conservative variables and in terms of the primitive variables. In these equations, the flux Jacobians will appear as coefficient matrices.

A non-conservation form of the Euler equations in curvilinear coordinates can be directly derived from the basic conservation form in Cartesian coordinates (equation (2.30)) and is given by

$$\frac{\partial U}{\partial t} + \frac{\partial \mathcal{F}^c}{\partial \xi} \cdot \nabla \xi + \frac{\partial \mathcal{F}^c}{\partial \eta} \cdot \nabla \eta + \frac{\partial \mathcal{F}^c}{\partial \zeta} \cdot \nabla \zeta = 0. \quad (\text{A.1})$$

Using the identity

$$\frac{\partial(J\nabla\xi)}{\partial\xi} + \frac{\partial(J\nabla\eta)}{\partial\eta} + \frac{\partial(J\nabla\zeta)}{\partial\zeta} = 0, \quad (\text{A.2})$$

which can be verified by substituting the expressions of equation (2.22) (and is the equivalence of equation (2.19) for the integral form), the conservation form of equation (2.33) can be derived. This conservation form can also be written as

$$J \frac{\partial U}{\partial t} + \frac{\partial}{\partial \xi} (F^\xi J \|\nabla \xi\|) + \frac{\partial}{\partial \eta} (F^\eta J \|\nabla \eta\|) + \frac{\partial}{\partial \zeta} (F^\zeta J \|\nabla \zeta\|) = 0, \quad (\text{A.3})$$

in which the flow-state vector U of conservative variables and the flux vector F^ξ are given by

$$U = \begin{pmatrix} \rho \\ \rho \vec{u} \\ \rho E \end{pmatrix}, \quad F^\xi = \mathcal{F}^c \cdot \vec{m} = \begin{pmatrix} \rho u_m \\ \rho u_m \vec{u} + p \vec{m} \\ \rho H u_m \end{pmatrix}, \quad (\text{A.4})$$

with $\vec{m} = \nabla \xi / \|\nabla \xi\|$ and $u_m = \vec{u} \cdot \vec{m}$. Differentiating \mathcal{F}^c first to U , a quasi-linear form can be derived from equation (A.1)

$$\frac{\partial U}{\partial t} + \|\nabla \xi\| \mathcal{A}^\xi \frac{\partial U}{\partial \xi} + \|\nabla \eta\| \mathcal{A}^\eta \frac{\partial U}{\partial \eta} + \|\nabla \zeta\| \mathcal{A}^\zeta \frac{\partial U}{\partial \zeta} = 0, \quad (\text{A.5})$$

with $\mathcal{A}^\xi = d\mathcal{F}^c/dU \cdot \vec{m} = dF^\xi/dU$ the Jacobian of the convective flux in ξ -direction, denoted as the 'conservative' flux Jacobian.

Consider the state vector W of primitive variables

$$W = \begin{pmatrix} \rho \\ \vec{u} \\ p \end{pmatrix}. \quad (\text{A.6})$$

The quasi-linear form can then also be written as

$$\frac{\partial W}{\partial t} + \|\nabla \xi\| \mathcal{B}^\xi \frac{\partial W}{\partial \xi} + \|\nabla \eta\| \mathcal{B}^\eta \frac{\partial W}{\partial \eta} + \|\nabla \zeta\| \mathcal{B}^\zeta \frac{\partial W}{\partial \zeta} = 0, \quad (\text{A.7})$$

with the ‘primitive’ flux Jacobian given by $\mathcal{B}^\xi = dW/dU A^\xi dU/dW = dW/dU dF^\xi/dW$. Since the conservative and primitive flux Jacobians are related by a similarity transformation they have the same eigenvalues.

Now, the primitive flux Jacobian together with its eigenvalues and eigenvectors will be derived. This Jacobian has a less complex form than the conservative Jacobian, and thus the derivation of eigenvalues and eigenvectors is easier. Expressing pressure in terms of the conservative variables, and total energy and total enthalpy in terms of the primitive variables (for a calorically perfect gas),

$$\begin{aligned} p &= (\gamma - 1) \left(\rho E - \frac{1}{2\rho} \|\rho \vec{u}\|^2 \right), \\ \rho E &= \frac{1}{\gamma - 1} p + \frac{1}{2} \rho \|\vec{u}\|^2, \\ \rho H &= \frac{\gamma}{\gamma - 1} p + \frac{1}{2} \rho \|\vec{u}\|^2, \end{aligned} \quad (\text{A.8})$$

the Jacobian matrices dW/dU and dU/dW can be derived,

$$\begin{aligned} \frac{dW}{dU} &= \begin{pmatrix} 1 & \vec{0}^T & 0 \\ -\vec{u}/\rho & \mathbf{I}/\rho & \vec{0} \\ \frac{\gamma-1}{2} \|\vec{u}\|^2 & -(\gamma-1)\vec{u}^T & (\gamma-1) \end{pmatrix}, \\ \frac{dU}{dW} &= \begin{pmatrix} 1 & \vec{0}^T & 0 \\ -\vec{u} & \rho \mathbf{I} & \vec{0} \\ \frac{1}{2} \|\vec{u}\|^2 & \rho \vec{u}^T & \frac{1}{\gamma-1} \end{pmatrix}, \end{aligned} \quad (\text{A.9})$$

as well as the flux Jacobian dF^ξ/dW ,

$$\frac{dF^\xi}{dW} = \begin{pmatrix} u_m & \rho \vec{m}^T & 0 \\ u_m \vec{u} & \rho \vec{u} \vec{m}^T + \rho u_m \mathbf{I} & \vec{m} \\ \frac{1}{2} \|\vec{u}\|^2 & \rho H \vec{m}^T + \rho u_m \vec{u}^T & \frac{\gamma}{\gamma-1} u_m \end{pmatrix}. \quad (\text{A.10})$$

Multiplication of dW/dU with dF^ξ/dW gives the primitive flux Jacobian,

$$\mathcal{B}^\xi = \begin{pmatrix} u_m & \rho \vec{m}^T & 0 \\ \vec{0} & u_m \mathbf{I} & \vec{m}/\rho \\ 0 & \gamma p \vec{m}^T & u_m \end{pmatrix}. \quad (\text{A.11})$$

Let $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_5)$ be the diagonal matrix with the eigenvalues of \mathcal{B}^ξ along its diagonal, and let \mathcal{R} and $\mathcal{L} = \mathcal{R}^{-1}$ be the left and right eigenvector matrices of \mathcal{B}^ξ , respectively. It can be easily verified that the eigenvalues are given by

$$\begin{aligned}\lambda_i &= u_m, & i \in \{1, 2, 3\}, \\ \lambda_4 &= u_m + c, \\ \lambda_5 &= u_m - c,\end{aligned}\tag{A.12}$$

with $c = (\gamma p / \rho)^{1/2}$ the speed of sound, and that the eigenvector matrices are given by

$$\begin{aligned}\mathcal{R} &= \begin{pmatrix} 1 & 0 & 0 & \rho/c & -\rho/c \\ \vec{0} & \vec{s} & \vec{t} & \vec{m} & \vec{m} \\ 0 & 0 & 0 & \rho c & -\rho c \end{pmatrix}, \\ \mathcal{L} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \vec{0} & \vec{s} & \vec{t} & \frac{1}{2}\vec{m} & \frac{1}{2}\vec{m} \\ -1/c^2 & 0 & 0 & 1/(2\rho c) & -1/(2\rho c) \end{pmatrix}^T,\end{aligned}\tag{A.13}$$

with $(\vec{m}, \vec{s}, \vec{t})$ orthonormal. The Jacobian \mathcal{B}^ξ can now be diagonalized by $\mathcal{B}^\xi = \mathcal{L} \Lambda \mathcal{R}$. Defining the characteristic variables $dV = \mathcal{L} dW$, the one-dimensional Euler equations (considering only the ξ -direction) may be written in characteristic form

$$\frac{\partial V}{\partial t} + \|\nabla \xi\| \Lambda \frac{\partial V}{\partial \xi} = 0.\tag{A.14}$$

The flux Jacobian \mathcal{A}^ξ has the same eigenvalues as \mathcal{B}^ξ , as stated before, and can be diagonalized as $\mathcal{A}^\xi = \mathcal{Q} \Lambda \mathcal{Q}^{-1}$ with $\mathcal{Q} = dU/dW \mathcal{R}$ and $\mathcal{Q}^{-1} = \mathcal{L} dW/dU$. The matrix \mathcal{Q} has the right eigenvectors of \mathcal{A}^ξ as its column vectors, similarly to the matrix \mathcal{R} . The column vectors may be scaled freely by a scalar, and upon doing so the matrix \mathcal{Q} may be shown to be equal to

$$\mathcal{Q} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ \vec{u} & u_s \vec{s} & u_t \vec{t} & \vec{u} + c \vec{m} & \vec{u} - c \vec{m} \\ \frac{1}{2} \|\vec{u}\|^2 & u_s^2 & u_t^2 & H + c u_m & H - c u_m \end{pmatrix},\tag{A.15}$$

while its inverse is given by

$$\mathcal{Q}^{-1} = \begin{pmatrix} 1 - k \frac{1}{2} \|\vec{u}\|^2 & -1 & -1 & k \frac{1}{4} \|\vec{u}\|^2 - \frac{1}{2} u_m / c & k \frac{1}{4} \|\vec{u}\|^2 + \frac{1}{2} u_m / c \\ k \vec{u} & \vec{s} / u_s & \vec{t} / u_t & -k \frac{1}{2} \vec{u} + \frac{1}{2} \vec{m} / c & -k \frac{1}{2} \vec{u} - \frac{1}{2} \vec{m} / c \\ -k & 0 & 0 & \frac{1}{2} k & \frac{1}{2} k \end{pmatrix}^T,\tag{A.16}$$

with $k = (\gamma - 1)/c^2$.

Finally, the absolute-value Jacobian matrix (as used in section 3.6.1) is defined by $|\mathcal{A}| = \mathcal{Q} |\Lambda| \mathcal{Q}^{-1}$. Let C_i be the column vectors of \mathcal{Q} and let R_i be the row vectors of \mathcal{Q}^{-1} . Then the absolute-value Jacobian matrix can be written as

$$|\mathcal{A}| = \sum_{i=1}^5 C_i |\lambda_i| R_i^T.\tag{A.17}$$

After considerable manipulation, this can be shown to be equivalent to

$$|A| = e_0 J + e_1 (kAC^T - BD^T) + e_2 (-AD^T + (\gamma - 1)BC^T), \quad (\text{A.18})$$

with J the unit 5×5 matrix, with the coefficients e_0 , e_1 , and e_2 given by

$$\begin{aligned} e_0 &= |\lambda_1|, \\ e_1 &= \frac{1}{2}(|\lambda_4| + |\lambda_5|) - |\lambda_1|, \\ e_2 &= \frac{|\lambda_4| - |\lambda_5|}{2c}, \end{aligned} \quad (\text{A.19})$$

and with the column vectors A , B , C , and D given by

$$A = \begin{pmatrix} 1 \\ \vec{u} \\ H \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \vec{m} \\ u_m \end{pmatrix}, \quad C = \begin{pmatrix} \frac{1}{2}\|\vec{u}\|^2 \\ -\vec{u} \\ 1 \end{pmatrix}, \quad D = \begin{pmatrix} u_m \\ -\vec{m} \\ 0 \end{pmatrix}. \quad (\text{A.20})$$

The Jacobian matrix as used in section 3.6.1 is actually defined for the convective flux $F_A = F^\epsilon \|\vec{A}\|$ across a cell face with area vector \vec{A} . Thus, the absolute-value Jacobian matrix presented in this section should be multiplied by $\|\vec{A}\|$, while the unit vector \vec{m} should be replaced by $\vec{A}/\|\vec{A}\|$.

APPENDIX B

STABILITY CONSIDERATIONS

In this appendix, we will present a stability analysis for a simple model equation, with as purpose to derive the expression for the time step used in section 4.3.2 and to derive suitable values for the stability limits C_{FL} (Courant number) and R_K , and for the numerical parameter $k^{(4)}$ of the artificial diffusion. These values depend on the specific Runge–Kutta time-integration scheme used and on the amount of implicit residual averaging. Besides stability, attention is also paid to the damping of high-frequency modes, which is important when the RK scheme is used as relaxation operator in a multi-grid scheme.

Consider the following 3D scalar model equation:

$$\frac{\partial u}{\partial t} + \lambda^\xi \frac{\partial u}{\partial \xi} + \lambda^\eta \frac{\partial u}{\partial \eta} + \lambda^\zeta \frac{\partial u}{\partial \zeta} = \lambda^v \frac{\partial^2 u}{\partial \eta^2}, \quad (\text{B.1})$$

with constant coefficients $\lambda^\xi, \lambda^\eta, \lambda^\zeta, \lambda^v \in \mathbb{R}^+$ (representing the eigenvalues of the convective and viscous flux Jacobians). The independent variables are the spatial coordinates $(\xi, \eta, \zeta) \in \Omega = [0, 1]^3$ and the time coordinate $t \in [0, \infty)$. The dependent variable is a scalar function $u : \Omega \times [0, \infty) \rightarrow \mathbb{R}$, which is considered to be periodic.

Let the spatial domain Ω be discretized by a uniform grid of dimension $N^\xi \times N^\eta \times N^\zeta$ as follows:

$$\begin{aligned} (\xi, \eta, \zeta) &\in \Omega_h \subset \Omega, \\ \Omega_h &= \{ (ih^\xi, jh^\eta, kh^\zeta) \mid i = 0 \dots N^\xi, j = 0 \dots N^\eta, k = 0 \dots N^\zeta \}, \\ h^\xi &= 1/N^\xi, \quad h^\eta = 1/N^\eta, \quad h^\zeta = 1/N^\zeta, \end{aligned} \quad (\text{B.2})$$

and let u be approximated by the grid function given by

$$u_{i,j,k} : \Omega_h \times [0, \infty) \rightarrow \mathbb{R}. \quad (\text{B.3})$$

Equation (B.1) is discretized in space using central differencing and addition of fourth-order artificial diffusion,

$$\begin{aligned} \tau \frac{du_{i,j,k}}{dt} + R_{i,j,k} &= 0, \\ \frac{1}{\tau} R_{i,j,k} &= \frac{1}{2} A^\xi (\delta_\xi^1 u)_{i,j,k} + B^\xi (\delta_\xi^4 u)_{i,j,k} \\ &+ \frac{1}{2} A^\eta (\delta_\eta^1 u)_{i,j,k} + B^\eta (\delta_\eta^4 u)_{i,j,k} \\ &+ \frac{1}{2} A^\zeta (\delta_\zeta^1 u)_{i,j,k} + B^\zeta (\delta_\zeta^4 u)_{i,j,k} - \frac{1}{4} C^\eta (\delta_\eta^2 u)_{i,j,k}, \end{aligned} \quad (\text{B.4})$$

with τ the time step and $R_{i,j,k}$ the residual. The first-, second-, and fourth-order differences are given by

$$\begin{aligned}(\delta_\xi^1 u)_{i,j,k} &= u_{i+1,j,k} - u_{i-1,j,k}, \\(\delta_\xi^2 u)_{i,j,k} &= u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}, \\(\delta_\xi^4 u)_{i,j,k} &= (\delta_\xi^2 u)_{i+1,j,k} - 2(\delta_\xi^2 u)_{i,j,k} + (\delta_\xi^2 u)_{i-1,j,k},\end{aligned}\tag{B.5}$$

while the constant coefficients are given by

$$A^\xi = \frac{\lambda^\xi}{h^\xi}, \quad A^\eta = \frac{\lambda^\eta}{h^\eta}, \quad A^\zeta = \frac{\lambda^\zeta}{h^\zeta}, \quad C^\eta = \frac{4\lambda^v}{(h^\eta)^2}.\tag{B.6}$$

The coefficients of the artificial diffusion (B^ξ , B^η , and B^ζ) will be defined below. Define the ratios $\rho = A^\xi/A^\eta$, $\sigma = A^\xi/A^\zeta$, and $v = A^\zeta/A^\eta$. When the convective eigenvalues are equal ($\lambda^\xi = \lambda^\eta = \lambda^\zeta$), these ratios are the cell aspect ratios.

In order to analyse the stability properties of this equation and of the time-integration schemes, we apply Fourier or local mode analysis. The grid function $u_{i,j,k}$ may be represented by a discrete Fourier series given by

$$u_{i,j,k} = \sum_{l=0}^{N^\xi-1} \sum_{m=0}^{N^\eta-1} \sum_{n=0}^{N^\zeta-1} \hat{u}_{l,m,n} \exp\left(i(\theta_l^\xi i + \theta_m^\eta j + \theta_n^\zeta k)\right),\tag{B.7}$$

with $\hat{i}^2 = -1$ and with the wave numbers given by

$$\theta_l^\xi = \frac{2\pi l}{N^\xi}, \quad \theta_m^\eta = \frac{2\pi m}{N^\eta}, \quad \theta_n^\zeta = \frac{2\pi n}{N^\zeta}.\tag{B.8}$$

Due to linearity, we may consider for simplicity only one Fourier mode,

$$u_{i,j,k} = \hat{u} \exp\left(i(\theta^\xi i + \theta^\eta j + \theta^\zeta k)\right),\tag{B.9}$$

with $\theta^\xi, \theta^\eta, \theta^\zeta \in [-\pi, \pi]$. Substitution in equation (B.4) gives for the residual

$$R_{i,j,k} = -z \hat{u} \exp\left(i(\theta^\xi i + \theta^\eta j + \theta^\zeta k)\right),\tag{B.10}$$

with $z \in \mathbb{C}$ the Fourier symbol of the residual,

$$\begin{aligned}z &= x + iy = x^d + x^v + iy^c, \\x^d &= -4\tau B^\xi(1 - \cos \theta^\xi)^2 - 4\tau B^\eta(1 - \cos \theta^\eta)^2 - 4\tau B^\zeta(1 - \cos \theta^\zeta)^2, \\x^v &= -\frac{1}{2}\tau C^\eta(1 - \cos \theta^\eta), \\y^c &= -\tau A^\xi \sin \theta^\xi - \tau A^\eta \sin \theta^\eta - \tau A^\zeta \sin \theta^\zeta.\end{aligned}\tag{B.11}$$

The semi-discrete equation (B.4) can now be written as an ODE for the Fourier symbol \hat{u} :

$$\tau \frac{d\hat{u}}{dt} = z\hat{u}.\tag{B.12}$$

Note that this ODE is (marginally) stable if $x \leq 0$, which is true if $B^\xi, B^\eta, B^\zeta, C^\eta \geq 0$.

Time integration and time step

One time step of any numerical time-integration scheme for the ODE of equation (B.12) can be represented by an amplification factor $G(z)$,

$$\hat{u}^{n+1} = G(z) \hat{u}^n, \quad (\text{B.13})$$

with the superscript n indicating the solution at time $t^n = n\tau$. For example, for the standard three-stage Runge–Kutta scheme the amplification factor is given by

$$G(z) = 1 + \alpha_3 z + \alpha_3 \alpha_2 z^2 + \alpha_3 \alpha_2 \alpha_1 z^3, \quad (\text{B.14})$$

while if the artificial diffusion is frozen at the first stage (the RK3 scheme defined in section 4.4.1), the amplification factor is given by

$$G(z) = 1 + \alpha_3 x + i\alpha_3 y (1 + \alpha_2 x + i\alpha_2 y (1 + \alpha_1 x + i\alpha_1 y)). \quad (\text{B.15})$$

The time integration is stable if the following requirement is satisfied:

$$\forall \theta^\xi, \theta^\eta, \theta^\zeta \in [-\pi, \pi] : |G(z)| \leq 1, \quad (\text{B.16})$$

which typically means that z must lie in a (stability) domain as depicted in figures B.1 to B.3 for the RK3, RK4, and RK5 schemes (as defined in section 4.4.1). In general, it is sufficient to define in the z -plane a stability limit C_{FL} along the imaginary axis and a stability limit R_K along the negative real axis (see figure B.1), i.e. to require that

$$\begin{aligned} \max |y| &\leq C_{\text{FL}}, \\ \max(-x) &\leq R_K. \end{aligned} \quad (\text{B.17})$$

Not considering artificial diffusion (i.e. assuming $x^d = 0$), this leads to the following requirements:

$$\begin{aligned} \tau &\leq \frac{C_{\text{FL}}}{A^\xi + A^\eta + A^\zeta} = \frac{C_{\text{FL}}}{\lambda^\xi/h^\xi + \lambda^\eta/h^\eta + \lambda^\zeta/h^\zeta}, \\ \tau &\leq \frac{R_K}{C^\eta} = R_K \frac{(h^\eta)^2}{4\lambda^\eta}. \end{aligned} \quad (\text{B.18})$$

This requirement can be satisfied by defining the time step by the relation

$$\frac{1}{\tau} = \frac{1}{C_{\text{FL}}} \left(\frac{\lambda^\xi}{h^\xi} + \frac{\lambda^\eta}{h^\eta} + \frac{\lambda^\zeta}{h^\zeta} \right) + \frac{4\lambda^\eta}{R_K (h^\eta)^2}, \quad (\text{B.19})$$

which is essentially the relation used in section 4.3.2.

Standard artificial diffusion

We will now show that the time step just defined is also appropriate when the artificial diffusion is included, provided that the artificial-diffusion coefficients B^ξ , B^η , and

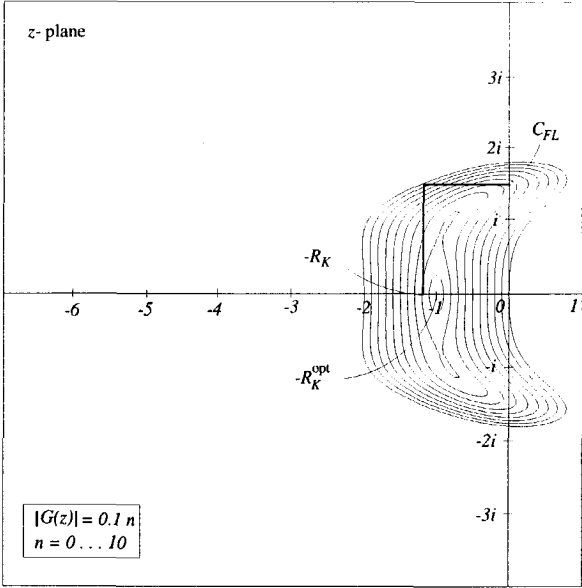


Figure B.1: Stability region in z -plane for RK3 scheme

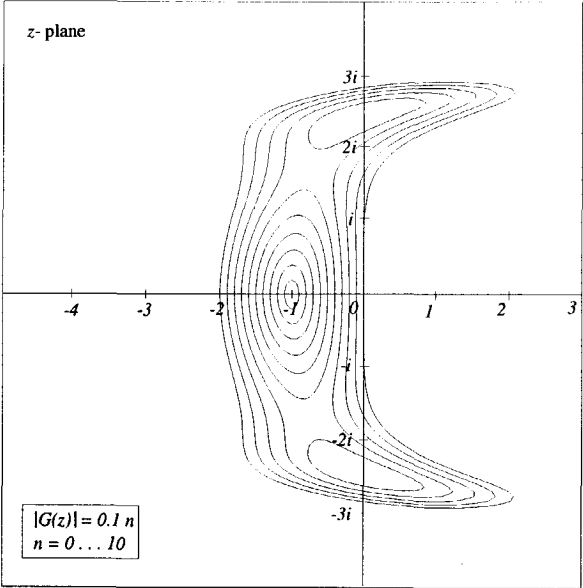


Figure B.2: Stability region in z -plane for RK4 scheme

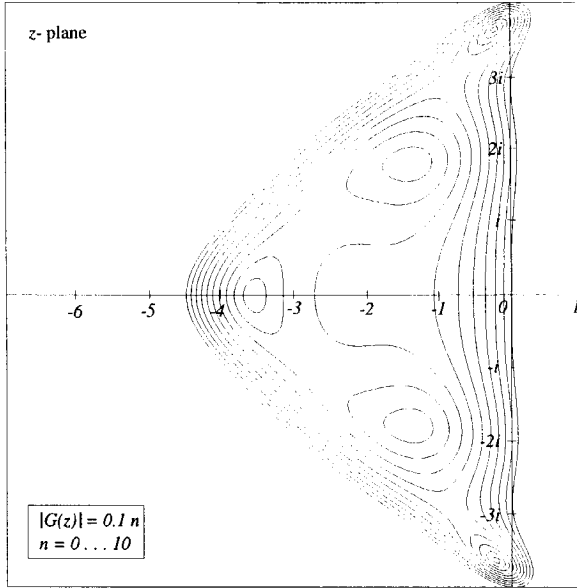


Figure B.3: Stability region in z -plane for RK5 scheme

B^ζ have appropriate values. Following section 3.4.3, the standard values for these coefficients are given by

$$B^\xi = \frac{1}{64} k^{(4)} A^\xi, \quad B^\eta = \frac{1}{64} k^{(4)} A^\eta, \quad B^\zeta = \frac{1}{64} k^{(4)} A^\zeta. \quad (\text{B.20})$$

The parameter $k^{(4)}$ should be defined such that high-frequency modes are strongly damped, while the time integration remains stable.

First, we consider the inviscid equations (i.e. $\lambda^v = 0$). From figure B.1, it can be seen that a small amplification factor, and therefore a strong damping, occurs at the location $x = -R_K^{\text{opt}}$ and $y = 0$. Thus, a strong damping of high-frequency modes is obtained if

$$x^d(\theta^\xi = \pi, \theta^\eta = \pi, \theta^\zeta = \pi) = -R_K^{\text{opt}}. \quad (\text{B.21})$$

Using the time step given by equation (B.19), one finds that

$$-x^d(\theta^\xi = \pi, \theta^\eta = \pi, \theta^\zeta = \pi) = 16\tau\Sigma B = \frac{1}{4} k^{(4)} \tau \Sigma A = \frac{1}{4} k^{(4)} C_{\text{FL}}, \quad (\text{B.22})$$

with $\Sigma A = A^\xi + A^\eta + A^\zeta$ and $\Sigma B = B^\xi + B^\eta + B^\zeta$, so that an appropriate choice for $k^{(4)}$ is

$$k^{(4)} = \frac{4R_K^{\text{opt}}}{C_{\text{FL}}}. \quad (\text{B.23})$$

Next, we consider the viscous equations (i.e. with λ^v non-zero). Let $k^{(4)}$ be given by equation (B.23) and let the time step be defined by equation (B.19). If we require that $R_K^{\text{opt}} < R_K$ (as in figure (B.1)), then it follows that

$$\max(-x^d - x^v) = \tau(16\Sigma B + C^\eta) = \frac{R_K^{\text{opt}} \Sigma A / C_{\text{FL}} + C^\eta}{\Sigma A / C_{\text{FL}} + C^\eta / R_K} \leq R_K, \quad (\text{B.24})$$

and thus the stability requirement (B.17) is satisfied.

Implicit residual averaging and scaled artificial diffusion

We will now consider the time-integration scheme together with implicit residual averaging (section 4.3.3) as well as with the scaling for high-aspect-ratio cells (section 4.4.2) of both artificial diffusion and residual averaging. It will be seen that with residual averaging the time-step limits can be increased (R_K stronger than C_{FL}), which is the reason for applying residual averaging, while the value of $k^{(4)}$ can be maintained.

Let residual averaging be applied to the residual at every stage of the RK scheme, according to

$$((1 - \epsilon^\xi \delta_\xi^2)(1 - \epsilon^\eta \delta_\eta^2)(1 - \epsilon^\zeta \delta_\zeta^2)S)_{i,j,k} = R_{i,j,k}, \quad (\text{B.25})$$

with $S_{i,j,k}$ the smoothed residual. Substituting the expression of equation (B.10) for both residuals, one finds the following relation for the Fourier symbol \tilde{z} of the smoothed residual:

$$f(\epsilon^\xi, \theta^\xi) f(\epsilon^\eta, \theta^\eta) f(\epsilon^\zeta, \theta^\zeta) \tilde{z} = z, \quad (\text{B.26})$$

with

$$f(\epsilon, \theta) = 1 + 2\epsilon(1 - \cos \theta). \quad (\text{B.27})$$

One time step for the ODE of equation (B.12) is now given by

$$\hat{u}^{n+1} = G(\tilde{z}) \hat{u}^n, \quad (\text{B.28})$$

i.e. the amplification factor is given by the same function G as before, but with the Fourier symbol of the residual replaced by the Fourier symbol of the smoothed residual. This means that instead of requiring z to lie in the stability domain of the RK scheme, we must require \tilde{z} to lie in the stability domain. If, as before, the Fourier symbol of the smoothed residual is split as

$$\tilde{z} = \tilde{x} + \tilde{y} = \tilde{x}^d + \tilde{x}^v + \tilde{y}^c, \quad (\text{B.29})$$

with

$$\begin{aligned} \tilde{x} &= \frac{x}{f(\epsilon^\xi, \theta^\xi) f(\epsilon^\eta, \theta^\eta) f(\epsilon^\zeta, \theta^\zeta)}, \\ \tilde{y} &= \frac{y}{f(\epsilon^\xi, \theta^\xi) f(\epsilon^\eta, \theta^\eta) f(\epsilon^\zeta, \theta^\zeta)}, \end{aligned} \quad (\text{B.30})$$

then the stability requirement given by equation (B.17) is replaced by

$$\begin{aligned} \max |\tilde{y}| &\leq C_{FL}^*, \\ \max(-\tilde{x}) &\leq R_K^*, \end{aligned} \quad (\text{B.31})$$

with C_{FL}^* and R_K^* the stability limits defined in the \tilde{z} -plane (analogous to figure B.1). The time step is still given by relation (B.19), but with the values of C_{FL} and R_K increased with respect to the values of C_{FL}^* and R_K^* (i.e. the values used for the RK scheme without residual averaging) due to the residual averaging.

Considering the scaling for high-aspect-ratio cells, the residual-averaging coefficients ϵ^ξ , ϵ^η , and ϵ^ζ are chosen as (see section 4.4.2)

$$\begin{aligned}\epsilon^\xi &= \max \left\{ \frac{1}{4} \left(\left(\epsilon \frac{1 + \rho^{-\kappa} + \sigma^{-\kappa}}{1 + \rho^{-1} + \sigma^{-1}} \right)^2 - 1 \right), 0 \right\}, \\ \epsilon^\eta &= \max \left\{ \frac{1}{4} \left(\left(\epsilon \frac{1 + \rho^\kappa + v^\kappa}{1 + \rho + v} \right)^2 - 1 \right), 0 \right\}, \\ \epsilon^\zeta &= \max \left\{ \frac{1}{4} \left(\left(\epsilon \frac{1 + \sigma^\kappa + v^{-\kappa}}{1 + \sigma + v^{-1}} \right)^2 - 1 \right), 0 \right\},\end{aligned}\tag{B.32}$$

with $\epsilon \geq 1$ and $0 \leq \kappa \leq 1$, while the artificial-diffusion coefficients B^ξ , B^η , and B^ζ are chosen as (see section 4.4.2)

$$\begin{aligned}B^\xi &= \frac{1}{64} k^{(4)} A^\xi \max \left\{ 1, \frac{1}{3} (1 + \rho^{-\kappa} + \sigma^{-\kappa}) \right\} = \frac{1}{64} k^{(4)} A^\xi \frac{1}{3} (1 + \rho^{-\kappa} + \sigma^{-\kappa}), \\ B^\eta &= \frac{1}{64} k^{(4)} A^\eta \max \left\{ 1, \frac{1}{3} (1 + \rho^\kappa + v^\kappa) \right\} = \frac{1}{64} k^{(4)} A^\eta, \\ B^\zeta &= \frac{1}{64} k^{(4)} A^\zeta \max \left\{ 1, \frac{1}{3} (1 + \sigma^\kappa + v^{-\kappa}) \right\} = \frac{1}{64} k^{(4)} A^\zeta \frac{1}{3} (1 + \sigma^\kappa + v^{-\kappa}).\end{aligned}\tag{B.33}$$

Since the η direction is the viscous direction, we have assumed small mesh sizes in this direction compared to the other two directions, in particular in equation (B.33) $\rho^\kappa \leq 1/2$ and $v^\kappa \leq 1/2$ has been used.

Let the time step be given by equation (B.19). Then the following inequalities can be derived:

$$\begin{aligned}\max |\tilde{y}| &\leq \tau \max \left(\frac{A^\xi \sin \theta^\xi}{f(\epsilon^\xi, \theta^\xi)} + \frac{A^\eta \sin \theta^\eta}{f(\epsilon^\eta, \theta^\eta)} + \frac{A^\zeta \sin \theta^\zeta}{f(\epsilon^\zeta, \theta^\zeta)} \right) \\ &= \tau \left(\frac{A^\xi}{\sqrt{1 + 4\epsilon^\xi}} + \frac{A^\eta}{\sqrt{1 + 4\epsilon^\eta}} + \frac{A^\zeta}{\sqrt{1 + 4\epsilon^\zeta}} \right) \\ &\leq \frac{C_{\text{FL}}}{\Sigma A} \cdot \left(\frac{\Sigma A}{\epsilon(1 + \rho^{-\kappa} + \sigma^{-\kappa})} + \frac{\Sigma A}{\epsilon(1 + \rho^\kappa + v^\kappa)} + \frac{\Sigma A}{\epsilon(1 + \sigma^\kappa + v^{-\kappa})} \right) \\ &= \frac{C_{\text{FL}}}{\epsilon},\end{aligned}\tag{B.34}$$

and

$$\begin{aligned}\max(-\tilde{x}) &\leq \tau \max \left(\frac{4B^\xi(1 - \cos \theta^\xi)^2}{f(\epsilon^\xi, \theta^\xi)} + \frac{4B^\eta(1 - \cos \theta^\eta)^2}{f(\epsilon^\eta, \theta^\eta)} + \right. \\ &\quad \left. \frac{4B^\zeta(1 - \cos \theta^\zeta)^2}{f(\epsilon^\zeta, \theta^\zeta)} + \frac{C^\eta(1 - \cos \theta^\eta)}{2f(\epsilon^\eta, \theta^\eta)} \right) \\ &= \tau \left(\frac{16B^\xi}{1 + 4\epsilon^\xi} + \frac{16B^\eta}{1 + 4\epsilon^\eta} + \frac{16B^\zeta}{1 + 4\epsilon^\zeta} + \frac{C^\eta}{1 + 4\epsilon^\eta} \right).\end{aligned}\tag{B.35}$$

Substituting the coefficients B^ξ , B^η , and B^ζ by their definitions (equation (B.33)), and using the inequalities

$$\begin{aligned} \frac{A^\xi(1 + \rho^{-\kappa} + \sigma^{-\kappa})}{1 + 4\epsilon^\xi} &= \frac{\Sigma A}{\epsilon} \min \left\{ \frac{1}{\epsilon} \frac{1 + \rho^{-1} + \sigma^{-1}}{1 + \rho^{-\kappa} + \sigma^{-\kappa}}, \epsilon \frac{1 + \rho^{-\kappa} + \sigma^{-\kappa}}{1 + \rho^{-1} + \sigma^{-1}} \right\} \\ &\leq \frac{1}{\epsilon} \Sigma A, \end{aligned} \quad (\text{B.36})$$

and

$$\frac{1}{1 + 4\epsilon^\eta} = \min \left\{ \frac{1}{\epsilon^2} \left(\frac{1 + \rho + v}{1 + \rho^\kappa + v^\kappa} \right)^2, 1 \right\} \leq \frac{1}{\epsilon^2}, \quad \text{if } \rho \leq 1, v \leq 1, \quad (\text{B.37})$$

equation (B.35) becomes

$$\max(-\tilde{x}) \leq \frac{\frac{2}{3} \frac{k^{(4)}}{4\epsilon} \Sigma A + \frac{k^{(4)}}{4\epsilon^2} A^\eta + \frac{1}{\epsilon^2} C^\eta}{\frac{1}{C_{\text{FL}}} \Sigma A + \frac{1}{R_K} C^\eta}. \quad (\text{B.38})$$

From equations (B.34) and (B.38) it follows that the stability requirement given by equation (B.31) can be satisfied by defining

$$\begin{aligned} C_{\text{FL}} &= \epsilon C_{\text{FL}}^*, \\ R_K &= \epsilon^2 R_K^*, \\ k^{(4)} &\leq \frac{3}{5} \frac{4R_K^*}{C_{\text{FL}}^*}. \end{aligned} \quad (\text{B.39})$$

Thus, we see that with implicit residual averaging, the stability limits C_{FL} and R_K may be increased by factors ϵ and ϵ^2 , respectively. The value of $k^{(4)}$, given by equation (B.23), can be maintained, provided that $R_K^{\text{opt}} \leq \frac{3}{5} R_K^*$.

1D model equation

For a 1D model equation, we will show the distribution of the amplification factor as a function of the wave number for standard values of the stability limits C_{FL} and R_K , for the numerical parameter $k^{(4)}$, and for the residual-averaging coefficient ϵ . Consider a 1D inviscid version of the model equation with the Fourier symbol of the residual equal to

$$\begin{aligned} z &= x^d + i y^c, \\ x^d &= -\frac{1}{16} k^{(4)} C_{\text{FL}} (1 - \cos \theta^\xi)^2, \\ y^c &= -C_{\text{FL}} \sin \theta^\xi. \end{aligned} \quad (\text{B.40})$$

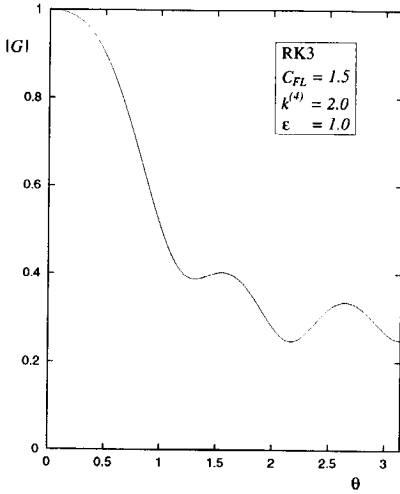
For this model equation, the amplification factor G as function of the wave number θ^ξ is given in figure B.4a for the RK3 scheme with $C_{\text{FL}} = 1.5$, in figure B.5a for the RK4 scheme with $C_{\text{FL}} = 2.6$, and in figure B.6a for the RK5 scheme with $C_{\text{FL}} = 3.0$. All three schemes show reasonable damping for high frequencies ($\theta^\xi \geq \pi/2$). Figures

B.4b to B.6b show the amplification factor when residual averaging is applied with $\varepsilon = 1.5$. The amplification factor is reduced for low frequencies, and is increased moderately for high frequencies (especially for the RK3 scheme). If stronger residual averaging is applied, then the high-frequency damping may be reduced too much, destroying the multi-grid convergence rate.

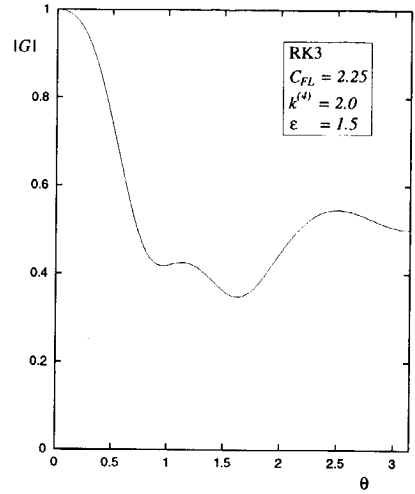
In chapter 4, a robust multi-grid scheme is obtained by applying five pre- and five post-relaxations or, in other words, by taking as relaxation operator five Runge-Kutta time steps instead of only one. Figure B.7 gives the amplification factor for the 1D model equation for five iterations of the RK5 scheme. As can be seen, strong damping of high-frequency modes is obtained, as is required in multi-grid theory.

2D model equation

Finally, for the relaxation operator considered as last for the 1D model equation (five RK5 time steps with residual averaging), we will show the amplification factor for a similar inviscid 2D model equation (thus $\lambda^\zeta = \lambda^\nu = 0$). For a cell aspect ratio of one ($\rho = 1$), the amplification factor G as function of the wave numbers θ^ξ and θ^η is given in figure B.8a. The damping of high frequency modes is similar to the 1D model equation. For a high aspect ratio of $\rho = 100$, the damping of modes which are of low frequency in the ξ direction ($\theta^\xi \approx 0$) but of high frequency in the η direction ($\theta^\eta \approx \pi/2$) is almost completely lost for the basic scheme (figure B.8b). Including the high-aspect-ratio scaling of residual averaging and artificial diffusion, the strong damping of all high-frequency modes is fully restored if $\kappa^{(h)} = 1$ (figure B.8c). However, as explained in section 3.6.2, a compromise between numerical accuracy and convergence speed is made by taking $\kappa^{(h)} = 2/3$, for which the high-frequency damping is less, but still reasonable (figure B.8d). For very high aspect ratios ($\rho = 10^4$), usually only occurring in a very small region close to solid surfaces, the damping of high frequencies of this latter scheme is rather weak (figure B.8f), but in practice reasonable convergence rates are obtained nevertheless (section 4.5).

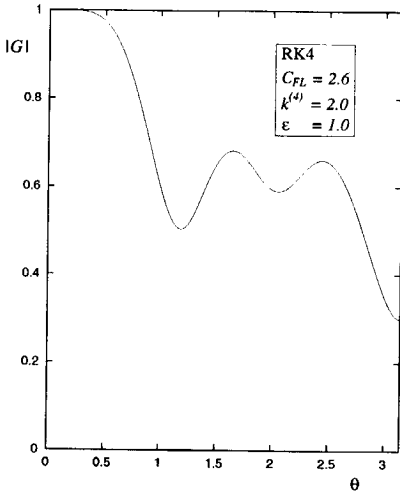


a) Without residual averaging

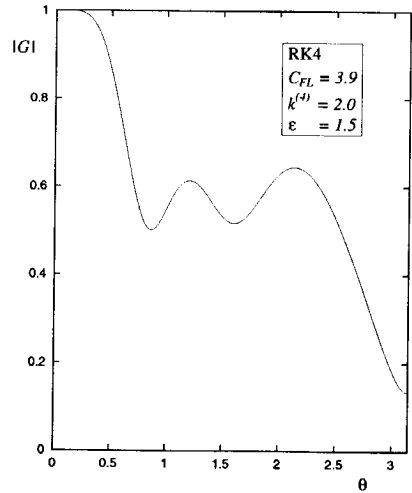


b) With residual averaging

Figure B.4: Amplification factor for one iteration of the RK3 scheme

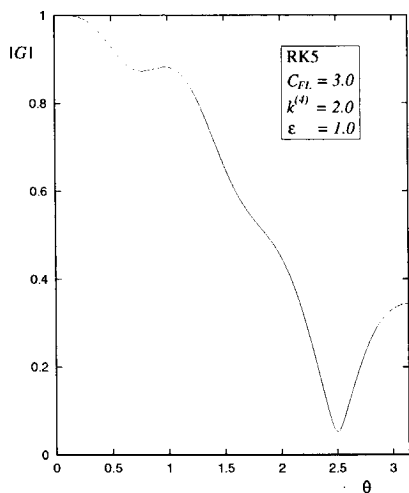


a) Without residual averaging

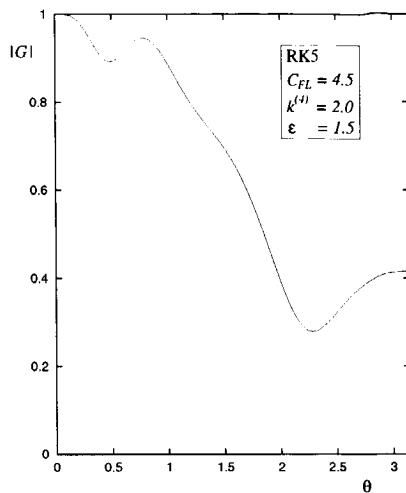


b) With residual averaging

Figure B.5: Amplification factor for one iteration of the RK4 scheme

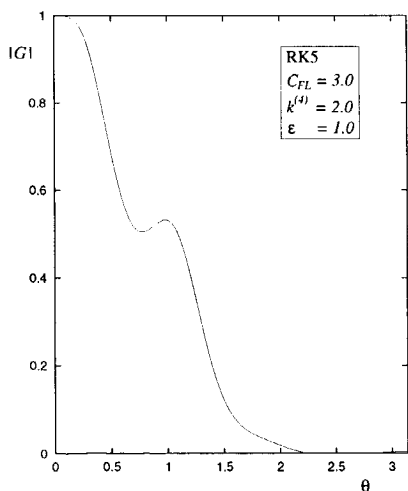


a) Without residual averaging

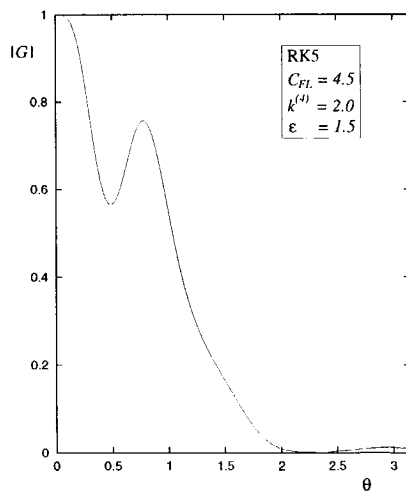


b) With residual averaging

Figure B.6: Amplification factor for one iteration of the RK5 scheme

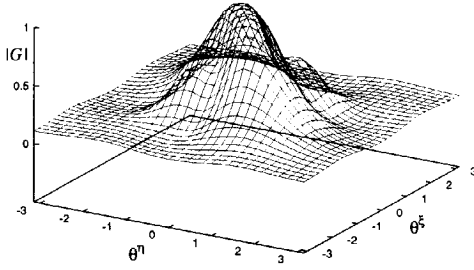


a) Without residual averaging



b) With residual averaging

Figure B.7: Amplification factor for five iterations of the RK5 scheme



a) Aspect ratio 1

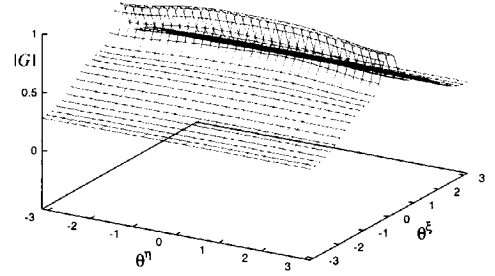
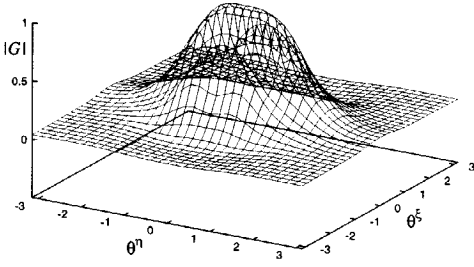
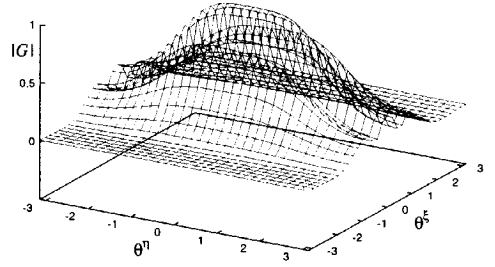
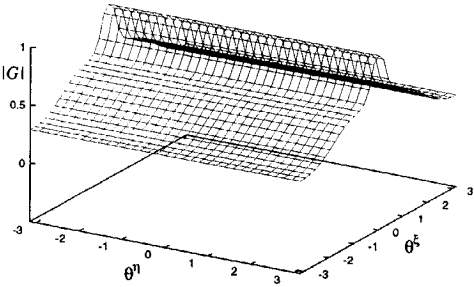
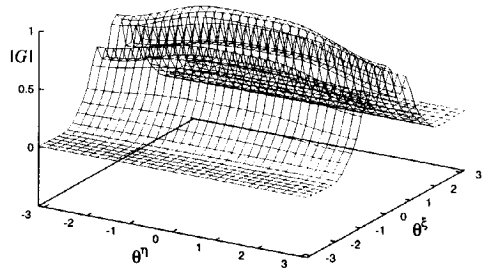
b) Aspect ratio 100, $\kappa^{(h)} = 0$ c) Aspect ratio 100, $\kappa^{(h)} = 1$ d) Aspect ratio 100, $\kappa^{(h)} = 2/3$ e) Aspect ratio 10^4 , $\kappa^{(h)} = 0$ f) Aspect ratio 10^4 , $\kappa^{(h)} = 2/3$

Figure B.8: Amplification factor for five iterations of the RK5 scheme for 2D model equation

BIBLIOGRAPHY

- [1] R. Abid and D.A. Johnson, 'Effects of turbulence models on the prediction of transonic wing flows'. AIAA-89-2224, 1989.
- [2] R. Abid, V.N. Vatsa, D.A. Johnson, and B.W. Wedan, 'Prediction of separated transonic wing flows with a non-equilibrium algebraic model'. AIAA-89-0558, 1989.
- [3] S.R. Allmaras, 'Contamination of laminar boundary layers by artificial dissipation in Navier-Stokes solutions', in *Conference on Numerical Methods in Fluid Dynamics*, (April 7-10 1992).
- [4] H.L. Atkins, 'A multi-block multigrid method for the Euler and Navier-Stokes equations for three-dimensional flows'. AIAA-91-0101, 1991.
- [5] B.S. Baldwin and T.J. Barth, 'A one-equation turbulence transport model for high Reynolds number wall-bounded flows', TM 102847, NASA, (1990).
- [6] B.S. Baldwin and H. Lomax, 'Thin-layer approximation and algebraic model for separated turbulent flows'. AIAA-78-257, 1978.
- [7] G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, Cambridge, 1967.
- [8] R.W. Beam and R.F. Warming, 'An implicit finite difference algorithm for hyperbolic systems in conservation form', *Journal of Computational Physics*, **23**, 87-110, (1976).
- [9] J.A. Benek, T.L. Donegan, and N.E. Suhs, 'Extended Chimera grid embedding scheme with applications to viscous flows'. AIAA-87-1126, 1987.
- [10] F.G. Blottner, 'Significance of the thin-layer Navier-Stokes approximation', in *Numerical and Physical Aspects of Aerodynamic Flows III*, ed., T. Cebeci, chapter 10, 184-197, Springer Verlag, New York, (1985).
- [11] J.W. Boerstael, 'Conceptual design of a computer-code system for the calculation of flows around transport aircraft', TP 92065 U, NLR, (1992).
- [12] J.W. Boerstael, A. Kassies, J.C. Kok, and S.P. Spekreijse, 'ENFLOW, a full-functionality system of CFD codes for industrial Euler/Navier-Stokes flow computations', TP 96286 L, NLR, (1996).
- [13] P. Bradshaw, D.H. Ferriss, and N.P. Atwell, 'Calculation of boundary layer development using the turbulent energy equation', *Journal of Fluid Mechanics*, **28**, 593-616, (1967).
- [14] F.J. Brandsma, 'Mathematical physics aspects of simulations based on the Navier-Stokes equations', TP 89060 L, NLR, (1989).

- [15] F.J. Brandsma, S. Bosse, H.W.M. Hoeijmakers, and J.I. van den Berg, 'The application of an Euler method and a Navier-Stokes method to the vortical flow about a cropped delta wing'. AIAA-93-3510, 1993.
- [16] A. Brandt, 'Multi-level adaptive solutions to boundary-value problems', *Mathematics of Computation*, **31**(138), 333-390, (1977).
- [17] A. Brandt, 'Guide to multigrid development', in *Multigrid Methods*, eds., W. Hackbusch and U. Trottenberg, number 960 in Lecture Notes in Mathematics, 220-312, Springer Verlag, Berlin, (1982).
- [18] F.E. Cannizzaro, A. Elmilgui, N.D. Melson, and E. von Lavante, 'A multiblock multigrid method for the solution of the three-dimensional Euler equations'. AIAA-90-0105, 1990.
- [19] D.A. Caughey, 'Diagonal implicit multigrid algorithm for the Euler equations', *AIAA Journal*, **26**(7), 841-851, (July 1989).
- [20] T. Cebeci and A.M.O. Smith, *Analysis of Turbulent Boundary Layers*, Academic Press, New York, 1974.
- [21] S.R. Chakravarthy, 'Relaxation methods for unfactored implicit upwind schemes'. AIAA-84-0165, 1984.
- [22] R.V. Chima, P.W. Giel, and R.J. Boyle, 'An algebraic turbulence model for three-dimensional viscous flows'. AIAA-93-0083, 1993.
- [23] F.H. Clauser, 'The turbulent boundary layer', *Advances in Applied Mechanics*, **4**, 1-51, (1956).
- [24] T.J. Coakley, 'Numerical simulation of viscous transonic airfoil flows'. AIAA-87-0416, 1987.
- [25] K.M.J. de Cock, 'Fully automatic Navier-Stokes algorithm for 2D high-lift devices', in *Fifteenth International Conference on Numerical Methods in Fluid Dynamics*, Monterey, USA, (June 24-28 1996). (NLR TP 96487).
- [26] P. Cook, M. McDonald, and M. Firmin, 'Airfoil RAE 2822 — pressure distributions and boundary layer wake measurements', AR 138, AGARD, (1979).
- [27] R. Courant and D. Hilbert, *Methods of Mathematical Physics II*, Wiley, New York, 1962.
- [28] D. Degani and L.B. Schiff, 'Computation of turbulent supersonic flows around pointed bodies having crossflow separation', *Journal of Computational Physics*, **66**, 173-196, (1986).
- [29] D. Degani, L.B. Schiff, and Y. Levy, 'Physical considerations governing computation of turbulent flows over bodies at large incidence'. AIAA-90-0096, 1990.

- [30] J.-A. Désidéri et al., eds. *Inv. Lect. and Spec. Techn. Sess. of the Third ECCOMAS Computational Fluid Dynamics Conference*, Paris, France, September 9-13 1996. ECCOMAS, John Wiley & Sons.
- [31] J.-A. Désidéri et al., eds. *Proceedings of the Third ECCOMAS Computational Fluid Dynamics Conference*, Paris, France, September 9-13 1996. ECCOMAS, John Wiley & Sons.
- [32] M. Drela and M.B. Giles, 'Viscous-inviscid analysis of transonic and low Reynolds number airfoils', *AIAA Journal*, **25**(10), (1987).
- [33] L.-E. Eriksson, 'Boundary conditions for artificial dissipation operators', TN 1984-53, FFA, (1984).
- [34] R.P. Fedorenko, 'The speed of convergence of one iterative process', *USSR Computational Mathematics and Mathematical Physics*, **4**, 227-235, (1964).
- [35] C. Gacherieu, E. Chaput, and L. Tourrette, 'Navier-Stokes for the design of transport aircrafts using parallel computing', In Désidéri et al. [31], pp. 471-477.
- [36] B.J. Geurts and H. Kuerten, 'Numerical aspects of a block structured compressible flow solver', *Journal of Engineering Mathematics*, **27**, 293-307, (1993).
- [37] M. Giles, M. Drela, and W.T. Thompkins, 'Newton solution of direct and inverse transonic Euler equations'. AIAA-85-1530, 1985.
- [38] J.P. Gillyboeuf, P. Mansuy, and S. Pavsic, 'Two new Chimera methods: Application to missile separation'. AIAA-95-0353, 1995.
- [39] G.S. Glushko, 'Turbulent boundary layer on a flat plate in an incompressible fluid', Tech. Trans. No. F-10080, NASA, (1966).
- [40] W. Haase, 'Flow-physics modelling — A validation process', In Désidéri et al. [30], pp. 503-508.
- [41] R. Hagmeijer, 'Grid adaption based on modified anisotropic diffusion equations formulated in the parametric domain', *Journal of Computational Physics*, **115**, 169-183, (1994).
- [42] R. Hagmeijer, *Adaptation of Structured Grids Based on Weighted Least Squares Formulations*, Ph.D. dissertation, Delft University of Technology, 1997.
- [43] R. Hagmeijer and J.C. Kok, 'Adaptive 3D single-block grids for the computation of viscous flows around wings', in *Fifth International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Starkville, USA, (April 1-5 1996). (NLR TP 96288).
- [44] A. Harten, 'High resolution schemes for hyperbolic conservation laws', *Journal of Computational Physics*, **49**, 357-393, (1983).

- [45] D.P. Hills, 'Numerical aerodynamics: Past successes and future challenges from an industrial point of view', In Désidéri et al. [30], pp. 166–173.
- [46] T.L. Holst, 'Viscous transonic airfoil workshop compendium of results'. AIAA-87-1460, 1987.
- [47] A. Jameson, 'Multigrid algorithms for compressible flow calculations', Report 1743, Dept. of Mech. and Aerosp. Eng., Princeton University, (1985). Text of lecture given at 2nd European Conference on Multigrid Methods, Cologne.
- [48] A. Jameson, 'Numerical simulation of the Euler equations for compressible inviscid fluids', Report 1643, Dept. of Mech. and Aerosp. Eng., Princeton University, (1985).
- [49] A. Jameson, 'Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flow', in *AIAA 11th Computational Fluid Dynamics Conference*, Orlando, USA, (1993).
- [50] A. Jameson, 'The present status, challenges, and future developments in computational fluid dynamics', in *AGARD Symposium on Progress and Challenges in CFD Methods and Algorithms*, Sevilla, Spain, (1995).
- [51] A. Jameson and T.J. Baker, 'Multigrid solution of the Euler equations for aircraft configurations'. AIAA-84-0093, 1984.
- [52] A. Jameson, W. Schmidt, and E. Turkel, 'Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes'. AIAA-81-1259, 1981.
- [53] D.A. Johnson, 'Predictions of transonic separated flow with an eddy-viscosity/Reynolds-shear-stress closure model'. AIAA-85-1683, 1985.
- [54] D.A. Johnson, 'Nonequilibrium algebraic turbulence modeling considerations for transonic airfoils and wings'. AIAA-92-0026, 1992.
- [55] D.A. Johnson and T.J. Coakley, 'Improvements to a non-equilibrium algebraic turbulence model', *AIAA Journal*, **28**, 2000–2003, (1990).
- [56] D.A. Johnson and L.S. King, 'A new turbulence closure model for boundary layer flows with strong adverse pressure gradients and separation'. AIAA-84-0175, 1984.
- [57] D.A. Johnson, F.R. Menter, and C.L. Rumsey, 'The status of turbulence modeling for external aerodynamics'. AIAA-94-2226, 1994.
- [58] W.P. Jones and B.E. Launder, 'The calculation of low-Reynolds-number phenomena with a two-equation model of turbulence', *International Journal of Heat and Mass Transfer*, **16**, 1119–1130, (1973).
- [59] G. Kalitzin, A.R.B. Gould, and J.J. Benton, 'Application of two-equation turbulence models in aircraft design'. AIAA-96-0327, 1996.

- [60] A. Kassies and R. Tognaccini, 'Boundary conditions for Euler equations at internal block faces of multi-block domains using local grid refinement'. AIAA-90-1590, 1990.
- [61] L.S. King, 'A comparison of turbulence closure models for transonic flows about airfoils'. AIAA-87-0418, 1987.
- [62] J.C. Kok, 'A robust multi-block Navier-Stokes flow solver for industrial applications', In Désidéri et al. [31], pp. 972-978. (NLR TP 96323).
- [63] N. Kroll, R. Radespiel, and C.-C. Rossow, 'Structured grids solvers I - Accurate and efficient flow solvers for 3D applications on structured meshes', in *Special Course on Parallel Computing in CFD*, chapter 4, AGARD R-807, (October 1995).
- [64] A.M. Kuethe and C. Chow, *Foundations of Aerodynamics: Bases of Aerodynamic Design*, Wiley & sons, New York, 1976.
- [65] L.D. Landau and E.M. Lifshitz, *Fluid Mechanics*, volume 6 of *Course of Theoretical Physics*, Butterworth-Heinemann, Oxford, second edn., 1987.
- [66] P.D. Lax and B. Wendroff, 'Systems of conservation laws', *Communications on Pure and Applied Mathematics*, **13**, 217-237, (1960).
- [67] P.D. Lax and B. Wendroff, 'Difference schemes for hyperbolic equations with high order of accuracy', *Communications on Pure and Applied Mathematics*, **7**, 381-398, (1964).
- [68] B. van Leer, 'Flux-vector splitting for the Euler equations', *Lecture Notes in Physics*, **170**, 507-512, (1982).
- [69] B. van Leer, 'Upwind-difference methods for aerodynamic problems governed by the Euler equations', *Lectures in Applied Mathematics*, **22**, 327-336, (1985).
- [70] B. van Leer, C-H. Tai, and K.G. Powell, 'Design of optimally smoothing multi-stage schemes for the Euler equations'. AIAA-89-1933-CP, 1989.
- [71] B. van Leer, J.L. Thomas, P.L. Roe, and R.W. Newsome, 'A comparison of numerical flux formulas for the Euler and Navier-Stokes equations', in *AIAA Eighth Computational Fluid Dynamics Conference*, Honolulu, USA, (June 9-11 1987).
- [72] B. Loyd, K. Lee, and E. Murman, 'Semi-implicit Navier-Stokes solver (SINSS) calculations of separated flows around blunt delta wings'. AIAA-90-0590, 1990.
- [73] L. Martinelli, *Calculations of Viscous Flows with a Multigrid Method*, Ph.D. dissertation, Princeton University, 1987.
- [74] D.J. Mavriplis, 'Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes', Report 88-40, ICASE, (1988).

- [75] D.J. Mavriplis and A. Jameson, 'Multigrid solution of the Navier-Stokes equations on triangular meshes', *AIAA Journal*, **28**(8), 1415-1425, (August 1990).
- [76] F.R. Menter, 'Influence of freestream values on k - ω turbulence model predictions', *AIAA Journal*, **30**(6), 1657-1659, (1991).
- [77] F.R. Menter, 'Performance of popular turbulence models for attached and separated adverse pressure gradient flows'. AIAA-91-1784, 1991.
- [78] F.R. Menter, 'Zonal two equation k - ω turbulence models for aerodynamic flows'. AIAA-93-2906, 1993.
- [79] F.R. Menter, 'Assessment of two-equation turbulence models for transonic flows'. AIAA-94-2343, 1994.
- [80] K.W. Morton and M.F. Paisley, 'A finite volume scheme with shock fitting for the steady Euler equations', Report 87/6, Oxford University Computing Laboratory, (1987).
- [81] J. van Muijden, A.J. Broekhuizen, A.J. van der Wees, and J. van der Vooren, 'Flow analysis and drag prediction for transonic transport wing/body configurations using a viscous-inviscid interaction type method', in *19th Congress of ICAS*, Anaheim, California, USA, (September 18-23 1994). (NLR TP 94161 U).
- [82] W.A. Mulder, 'A new multigrid approach to convection problems', *Journal of Computational Physics*, **83**, 303-323, (1989).
- [83] R.H. Ni, 'A multiple grid scheme for solving the Euler equations', *AIAA Journal*, **20**, 1565-1571, (1982).
- [84] Nieuwstadt, *Turbulentie; inleiding in de theorie en toepassingen van turbulente stromingen*, Epsilon Uitgaven, Utrecht, 1992.
- [85] S. Obayashi and K. Kuwahara, 'LU factorization of an implicit scheme for the compressible Navier-Stokes equations'. AIAA-84-1670, 1984.
- [86] S. Osher and F. Solomon, 'Upwind difference schemes for hyperbolic systems of conservation laws', *Mathematics of Computation*, **38**(158), 339-374, (1982).
- [87] V.C. Patel, W. Rodi, and G. Scheuerer, 'Turbulence models for near-wall and low Reynolds number flows: A review', *AIAA Journal*, **23**(9), 1308-1319, (1984).
- [88] T.H. Pulliam and D.S. Chaussee, 'A diagonal form of an implicit approximate-factorization algorithm', *Journal of Computational Physics*, **39**, 347-363, (1981).
- [89] J.J. Quirk, 'A Cartesian grid approach with hierarchical refinement for compressible flows', in *Inv. Lect. and Spec. Techn. Sess. of the Second ECCOMAS Computational Fluid Dynamics Conference*, eds., S. Wagner et al., pp. 200-209, Stuttgart, Germany, (September 5-8 1994). ECCOMAS, John Wiley & Sons.

- [90] R. Radespiel and R.C. Swanson, 'Progress with multigrid schemes for hypersonic flow problems', Report 91-89, ICASE, (1991).
- [91] P. Raj, C.R. Olling, and S.W. Singer, 'Applications of multizone Euler/Navier-Stokes aerodynamic methods to aircraft configurations', Report 90-6.4.4, ICAS, (1990).
- [92] R. Ramakrishnan, V. Vatsa, J. Otto, and A. Kumar, 'A detailed study of mean-flow solutions for stability analysis of transitional flows'. AIAA-93-3052, 1993.
- [93] P.J. Roache, 'Quantification of uncertainty in computational fluid dynamics', *Annual Review of Fluid Dynamics*, **29**, 123-160, (1997).
- [94] P.L. Roe, 'Approximate Riemann solvers, parameter vectors, and difference schemes', *Journal of Computational Physics*, **43**, 357-372, (1981).
- [95] P.L. Roe, 'Characteristic-based schemes for the Euler equations', *Annual Review of Fluid Mechanics*, **18**, 337-365, (1986).
- [96] C.-C. Rossow, 'Efficient computation of inviscid flow fields around complex configurations using a multi-block multigrid method', in *Fifth Copper Mountain Conference on Multigrid Methods*, (March 31 - April 5 1991).
- [97] C.L. Rumsey and V.N. Vatsa, 'A comparison of the predictive capabilities of several turbulence models using upwind and central-difference computer codes'. AIAA-93-0192, 1993.
- [98] H. Schlichting, *Boundary-Layer Theory*, McGraw-Hill, New York, 1968.
- [99] V. Schmitt and F. Charpin, 'Pressure distributions on the ONERA M6 wing at transonic Mach numbers', AR 138, AGARD, (1979).
- [100] *Computational Aerodynamics Based on the Euler Equations*, eds., J.W. Slooff and W. Schmidt, AGARD, AGARDograph 325, 1994.
- [101] M.A.M Somers and J. v.d. Vooren, 'Validation of digital simulation methods in CAE', in *Ontwikkelingen rond Industriële Automatisering; Congresbijdragen CAPE '89*, Alphen a.d. Rijn, the Netherlands, (1989). Samsom.
- [102] P.R. Spalart and S.R. Allmaras, 'A one-equation turbulence model for aerodynamic flows'. AIAA-92-0439, 1992.
- [103] S.P. Spekreijse, 'Multigrid solution of the steady Euler equations', Tract 46, CWI, (1988).
- [104] S.P. Spekreijse, 'Elliptic generation systems', TP 96735 L, NLR, (1996). (to appear in Handbook of Grid Generation, eds. J.F. Thompson et.al., CRC Press Inc., 1997).
- [105] S.P. Spekreijse and J.W. Boerstool, 'Multiblock grid generation', in *27th Computational Fluid Dynamics Course*, Rhode Saint Genèse, Belgium, (March 25-29 1996). Von Karman Institute for Fluid Dynamics. (NLR TP 96338 L).

- [106] S.P. Spekreijse, J.W. Boerstol, and P.L. Vitagliano, 'New concepts for multi-block grid generation for flow domains around complex aerodynamic configurations', TP 91046 U, NLR, (1991).
- [107] C.G. Speziale, 'Analytical methods for the development of Reynolds-stress closures in turbulence', *Annual Review of Fluid Mechanics*, **23**, 107-157, (1991).
- [108] J.L. Steger and R.F. Warming, 'Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods', *Journal of Computational Physics*, **40**, 263-293, (1981).
- [109] R.C. Swanson and R. Radespiel, 'Cell centered and cell vertex multigrid schemes', *AIAA Journal*, **29**(5), 697-703, (May 1991).
- [110] R.C. Swanson and E. Turkel, 'A multistage time-stepping scheme for the Navier-Stokes equations'. AIAA-85-0035, 1985.
- [111] R.C. Swanson and E. Turkel, 'On central-difference and upwind schemes', *Journal of Computational Physics*, **101**, 292-306, (1992).
- [112] R.C. Swanson and E. Turkel, 'Aspects of a high-resolution scheme for the Navier-Stokes equations'. AIAA-93-3372-CP, 1993.
- [113] H. Tennekes and J.L. Lumley, *A First Course in Turbulence*, MIT Press, 1972.
- [114] V.N. Vatsa and B.W. Wedan, 'Development of an efficient multigrid code for 3-D Navier-Stokes equations'. AIAA-89-1791, 1989.
- [115] M. Vinokur, 'Conservation equations of gasdynamics in curvilinear coordinate systems', *Journal of Computational Physics*, **14**, 105-125, (1974).
- [116] M. Vinokur, 'An analysis of finite-difference and finite-volume formulations of conservation laws', *Journal of Computational Physics*, **81**, 1-52, (1989).
- [117] M.E.S. Vogels, *Architecture and Construction Principles for Computational Fluid Dynamics Software for Engineering in Industry*, Ph.D. dissertation, University of Twente, 1997.
- [118] C.B. Vreugdenhill and B. Koren, *Numerical Methods for Advection-Diffusion Problems*, volume 45 of *Notes on Numerical Fluid Mechanics*, Vieweg, Braunschweig, Wiesbaden, 1993.
- [119] A.J. van der Wees and J. van Muijden, 'A fast and robust viscous-inviscid interaction solver for transonic flow about wing/body configurations on the basis of full potential theory', in *24th AIAA Fluid Dynamics Conference*, Orlando, Florida, USA, (July 6-9 1993). (AIAA-93-3026, NLR TP 93214 U).
- [120] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons Ltd., England, 1992.

-
- [121] D.C. Wilcox, 'Reassessment of the scale-determining equation for advanced turbulence models', *AIAA Journal*, **26**(11), 1299–1310, (1988).
 - [122] D.C. Wilcox, 'Comparison of two-equation turbulence models for boundary layers with pressure gradient', *AIAA Journal*, **31**(8), 1414–1421, (1993).
 - [123] D.C. Wilcox, *Turbulence Modeling for CFD*, DCW Industries Inc., 1993.
 - [124] D.C. Wilcox, 'A two-equation turbulence model for wall-bounded and free-shear flows'. AIAA-93-2905, 1993.
 - [125] S. Yoon and A. Jameson, 'Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations', *AIAA Journal*, **24**(11), 1737–1743, (November 1986).
 - [126] M. Zijlema, *Computational Modeling of Turbulent Flow in General Domains*, Ph.D. dissertation, Delft University of Technology, 1996.

NOMENCLATURE

ABBREVIATIONS

2D	two dimensional
3D	three dimensional
BL	Baldwin-Lomax turbulence model
CFD	computational fluid dynamics
CPU	central processing time
CS	Cebeci-Smith turbulence model
ENSOLV	NLR Euler/Navier-Stokes flow solver
FAS	Full Approximation Storage multi-grid scheme
FMG	Full Multi-Grid scheme
HST	NLR high-speed wind tunnel
ISES	Euler/boundary-layer method of Drela and Giles [32]
JK	Johnson-King turbulence model
MATRICES-V	NLR full-potential/boundary-layer method
MG	multi-grid computation
NIVR	Netherlands Agency for Aerospace Programs
NLR	National Aerospace Laboratory
ODE	ordinary differential equation
PDE	partial differential equation
RANS	Reynolds-averaged Navier-Stokes equations
RK	Runge-Kutta time-integration scheme
SG	single-grid computation
TLNS	thin-layer Reynolds-averaged Navier-Stokes equations

LIST OF SYMBOLS

a_1, C_{dif}	constants of the Johnson-King turbulence model
A, B, C, D	column vectors of the matrix artificial diffusion model ($A, B, C, D \in \mathbb{R}^5$)
A^+	constant of the Van Driest damping term
$\vec{A}^{(i)}, \vec{A}^{(j)}, \vec{A}^{(k)}$	cell-face area vectors in i, j , and k directions
A	Jacobian of the convective flux ($A \in \mathbb{R}^{5 \times 5}$)
B	block
B	residual-averaging matrix
c	speed of sound or local chord length, depending on context
c_p, c_v	specific heats at constant volume and constant pressure
$C_{\text{cp}}, C_{\text{kleb}}, C_{\text{wk}}$	constants of the Baldwin-Lomax outer-viscosity formula- tion

LIST OF SYMBOLS (CONTINUED)

C_D	drag coefficient ($C_D = F_D / (\frac{1}{2} \rho_\infty u_\infty^2 A_{\text{ref}})$, with F_D the drag component of the aerodynamic force and A_{ref} a reference area, typically the plan area of the wing)
C_f	skin-friction coefficient ($C_f = \tau_w / (\frac{1}{2} \rho_\infty u_\infty^2)$, unless otherwise indicated)
C_{FL}	Courant number
C_L	lift coefficient ($C_L = F_L / (\frac{1}{2} \rho_\infty u_\infty^2 A_{\text{ref}})$, with F_L the lift component of the aerodynamic force and A_{ref} a reference area, typically the plan area of the wing)
C_p	pressure coefficient ($C_p = (p - p_\infty) / (\frac{1}{2} \rho_\infty u_\infty^2)$)
D	flow domain
$D_{i,j,k}$	flux balance ($D_{i,j,k} \in \mathbb{R}^5$)
D_d, D_{JK}	Van Driest damping term
D^c, D^d, D^a	convective, diffusive, and artificial diffusive flux balances ($D^c, D^d, D^a \in \mathbb{R}^5$)
e	internal energy per unit mass
\vec{e}_i	Cartesian unit base vectors ($i \in \{1, 2, 3\}$)
e_0, e_1, e_2	coefficients of the matrix artificial diffusion model
E	total energy per unit mass
$E(\cdot)$	Entier function (nearest smaller integer)
f	Baldwin–Lomax function
$f^{(2)}, f^{(4)}$	first- and third-order difference in artificial diffusive flux
F	face
F^c, F^d, F^a	convective, diffusive, and artificial diffusive flux vectors ($F^c, F^d, F^a \in \mathbb{R}^5$)
\mathcal{F}	flux matrix ($\mathcal{F} \in \mathbb{R}^{5 \times 3}$)
g	dependent variable of the Johnson–King PDE ($g = (\tau_\rho^R)^{-1/2}$)
h	mesh size
H	total enthalpy per unit mass
(i, j, k)	grid-cell indices
(I, J, K)	grid-cell indices defined by cyclic permutation of (i, j, k)
$I_{l,l-1}$	prolongation operator (of correction)
\mathbf{I}	unit tensor of order 2 (Cartesian components δ_{ij})
\mathcal{J}	unit 5×5 matrix
J	determinant of the Jacobian of the Cartesian coordinates as a function of the curvilinear coordinates
$J_{l+1,l}$	prolongation operator (of flow state)
k	turbulent kinetic energy
$k^{(2)}, k^{(4)}, k^{(s)}, k^{(h)}$	numerical parameters of the artificial diffusion model
K	constant of the Clauser outer-viscosity formulation
l	grid level

LIST OF SYMBOLS (CONTINUED)

l_q	most recent stage before stage q of Runge–Kutta scheme at which the diffusive terms have been computed
L	reference length scale
\mathcal{L}	matrix with left-hand-side eigenvectors of \mathcal{A}^5
m	number of stages of Runge–Kutta scheme
$(\vec{m}, \vec{s}, \vec{t})$	orthonormal set of vectors with \vec{m} normal to a specified face or boundary
M	Mach number
M_i, M_j, M_k	refinement factors across a face with local grid refinement
n	normal distance to solid wall or wake-centre surface (normal to planes ξ, η , or $\zeta = \text{constant}$)
\vec{n}	unit vector in normal direction (normal to planes ξ, η , or $\zeta = \text{constant}$)
(N_i, N_j, N_k)	number of grid cells in i, j , and k directions
$N_{\text{pre}}, N_{\text{cgc}}, N_{\text{post}}$	number of pre-relaxations, coarse-grid corrections, and post-relaxations of multi-grid scheme
p	pressure
P	forcing function of multi-grid scheme
Pr, Pr_t	laminar and turbulent Prandtl numbers
Q, \vec{Q}	heat flux and heat-flux vector (Cartesian components Q_i)
$Q_{l-1,l}$	restriction operator (of defect)
Ω	eigenvector matrix of the convective Jacobian
r	defect ($r \in \mathbb{R}^5$)
R	gas constant
$R_{i,j,k}$	residual of main flow equations ($R_{i,j,k} \in \mathbb{R}^5$)
$R_{\text{in}}, R_{\text{out}}$	incoming and outgoing Riemann invariants
R_K	stability limit for diffusive terms
R^{JK}	residual of the Johnson–King PDE
Re	Reynolds number
RHS	right-hand side of the Johnson–King PDE
s	arc length
S	entropy
$S_{i-1/2,j,k}, S_{i,j-1/2,k},$ $S_{i,j,k-1/2}$	cell faces in i, j , and k directions
$S_{i,j,k}$	smoothed residual of main flow equations ($S_{i,j,k} \in \mathbb{R}^5$)
t	time
\mathbf{S}	rate-of-strain tensor
T	temperature
$T_{l-1,l}$	restriction operator (of flow state)
T_S	constant of Sutherland law
\vec{u}, u	velocity vector ($\vec{u} = u_i \vec{e}_i$) and its magnitude
u_τ	friction velocity ($u_\tau = (\tau_w / \rho_w)^{1/2}$)

LIST OF SYMBOLS (CONTINUED)

u'_m	velocity scale of Johnson–King inner-viscosity formulation ($u'_m = (\tau_\rho^R)_m^{1/2}$)
U	flow-state vector ($U = (\rho, \rho\vec{u}, \rho E)^T \in \mathbb{R}^5$)
U^*	flow-state vector ($U^* = (\rho, \rho\vec{u}, H)^T \in \mathbb{R}^5$)
V	state vector of characteristic variables ($V \in \mathbb{R}^5$)
$V_{i,j,k}$	volume of grid cell $\Omega_{i,j,k}$
W	flow-state vector ($W = (\rho, \rho\vec{u}, \rho E, p)^T \in \mathbb{R}^6$) in section 3.5.5, approximate solution ($W \in \mathbb{R}^5$) in section 4.4.1
\vec{x}	Cartesian coordinates $\vec{x} = x_i \vec{e}_i = (x, y, z)^T$
α	angle of attack
α_m	ratio of equilibrium inner and outer viscosities (Johnson–King model)
α_q	coefficient of Runge–Kutta scheme at stage q
$\alpha^{(K)}$	coefficient of high-aspect-ratio scaling
β	angle of side slip
γ	ratio of specific heats
γ_K	Klebanoff intermittency function
δ	boundary-layer thickness
δ^*	boundary-layer displacement thickness
δ_{ij}	Kronecker delta
Δn	mesh size in thin-layer normal direction
Δt	time step
ΔU	flow correction ($\Delta U \in \mathbb{R}^5$)
$\epsilon^{(2)}, \epsilon^{(4)}$	factors in formulation artificial diffusive fluxes
$\epsilon^{(i)}, \epsilon^{(j)}, \epsilon^{(k)}$	variable coefficients of implicit residual averaging
ε	numerical parameter of implicit residual averaging
$\varepsilon^L, \varepsilon^N$	numerical parameters of the matrix artificial diffusion model
η	variable of similarity equation for laminar flat-plate flow (chapter 3) span-wise coordinate (distance to the root divided by the semi-span) (chapters 5 and 6)
θ_S	ratio of the constant of the Sutherland law and the free-stream temperature
κ	Von Kármán constant
κ_h	thermal conductivity coefficient
κ_t	turbulent conductivity coefficient
λ	eigenvalue of the Jacobian of the convective flux
λ^v	eigenvalue of the Jacobian of the diffusive flux
Λ	diagonal matrix of convective eigenvalues
μ	(laminar) dynamic viscosity coefficient
μ_t	dynamic eddy-viscosity coefficient

LIST OF SYMBOLS (CONTINUED)

ν	shock sensor
ν_t	kinematic eddy-viscosity coefficient
$\nu_t^{\text{in}}, \nu_t^{\text{out}}$	inner and outer kinematic eddy-viscosity coefficient
ρ	density
σ	scaling factor of outer viscosity (Johnson–King model)
$\vec{\tau}, \tau$	shear stress vector and stress tensor (Cartesian components τ_{ij})
τ^R	Reynolds shear stress
$(\tau_\rho^R)_m$	maximum of Reynolds shear stress divided by density
(ξ, η, ζ)	boundary-conforming curvilinear coordinate system
$\vec{\omega}$	vorticity vector
Ω	volume segment (grid cell)
$\partial\Omega$	boundary of volume segment Ω
∇	gradient operator ($\nabla = \vec{e}_i \partial / \partial x_i$)
$\ \cdot \ $	Euclidean norm
subscripts:	
e	value at edge of boundary layer
eq	equilibrium value
i, j, k	value at the grid-cell indices
m	value at location of maximum Reynolds shear stress
w	wall value
∞	free-stream value
superscripts:	
l	variable at grid level l
n	value at time level n
(q)	value at stage q of the Runge–Kutta scheme
$+$	variable made dimensionless according to the law-of-the-wall scaling

SAMENVATTING (SUMMARY IN DUTCH)

EEN INDUSTRIEEL TOEPASBARE SIMULATIEMETHODE VOOR COMPRESSIBELE, TURBULENTE STROMINGEN

In dit proefschrift wordt een numerieke simulatiemethode voor de Reynolds-gemiddelde Navier-Stokes-vergelijkingen beschouwd. Deze methode is gericht op industriële toepassingen, met name de simulatie van compressibele, turbulente stromingen rond transportvliegtuigen ter ondersteuning van de integratie van voortstuwingsmechanismen met vleugel-rompconfiguraties. Uitgangspunt voor de ontwikkeling van de huidige methode is een reeds bestaande simulatiemethode voor de compressibele Euler-vergelijkingen. Viskeuze en turbulente effecten worden toegevoegd op basis van de 'thin-layer' Reynolds-gemiddelde Navier-Stokes-vergelijkingen en algebraïsche turbulentie modellen. Op deze wijze wordt de grenslaag op de vleugel gemodelleerd (en het effect van de grenslaag op de drukverdeling rond de vleugel), terwijl uitbreiding naar meer algemene viskeuze stromingen mogelijk blijft.

Aangezien de bestaande Euler-simulatiemethode geaccepteerd was en intensief gebruikt werd door de industrie, was hergebruik van methoden en software een vereiste. Deze simulatiemethode was gebaseerd op het gebruik van multi-blok gestructureerde rekenroosters, zodat willekeurig complexe geometrieën beschouwd konden worden. De ruimtediscretisatie van de Euler-vergelijkingen bestond uit een eindig-volume methode met de stromingsvariabelen gelokaliseerd in celmidden, waarbij gebruik werd gemaakt van centrale differenties en expliciete, artificiële diffusie. Als oplossingsmethode voor de discrete vergelijkingen werd een pseudo-tijdsintegratiemethode gebruikt, bestaande uit Runge-Kutta-schema's en impliciete middeling van de residuen.

Uitgaande van de meest voor de hand liggende uitbreiding van de numerieke methode voor de Euler-vergelijkingen naar de Navier-Stokes-vergelijkingen zijn verdere verbeteringen noodzakelijk. Een matrixvorm van artificiële diffusie wordt gebruikt om de roosterafhankelijkheid in grenslagen te verminderen zonder dat dit ten koste gaat van de convergentiesnelheid of de rekentijd. Een efficiënte oplossingsmethode wordt verkregen door de pseudo-tijdsintegratiemethode te versnellen met een multi-grid methode. Dit resulteert in een robuuste methode voor multi-blok roosters met slanke cellen indien de volgende concepten worden gebruikt: multi-blok binnen multi-grid, W-cyclus multi-grid met vijf pre- en postrelaxaties, en een geschikte schaling van de artificiële diffusie en van de middelingsoperator van de residuen. Verder wordt er aandacht besteed aan een robuuste implementatie van de algebraïsche turbulentie modellen, waarbij met name voor het Johnson-King-model de uitbreiding naar 3D en impliciete algebraïsche relaties van belang zijn.

Ten slotte worden de numerieke nauwkeurigheid en de efficiëntie van de simulatiemethode voor de Navier-Stokes-vergelijkingen geëvalueerd en wordt de toepasbaarheid voor typische transportvliegtuigconfiguraties gedemonstreerd.

CURRICULUM VITAE

The author was born in Leeuwarden, the Netherlands on October 20, 1967. He attended secondary school from 1979 to 1986 at the 'Samenwerkingsschool voor HAVO-Atheneum' in Waddinxveen, with the exception of the school year 1980–1981 in which he attended 'Tilden Junior High School' in Rockville, Maryland, USA. In 1986 he went to the Delft University of Technology, where he studied mathematics at the faculty of 'Technische Wiskunde en Informatica'; he graduated (cum laude) in June 1990. His graduation thesis dealt with the numerical simulation of incompressible, viscous flow around cylindrical objects using a spectral method; work which was conducted at the Maritime Research Institute of the Netherlands (Marin) and for which he received an award from the 'Universiteitsfonds Delft'. Since August 1990, the author has been working at the National Aerospace Laboratory NLR at the department of mathematical models and methods. Here, he has been involved in the ENFLOW project in which a flow-simulation system for compressible flow around aircraft (including propulsion systems) and other aerodynamic configurations is being developed. His contribution to this project was mainly concerned with the extension of a multi-block flow solver for the Euler equations to the Reynolds-averaged Navier–Stokes equations, as reflected in this thesis. Recent work concerns the further extension of the flow solver with a two-equation turbulence model.

