

**Compressible Flow Simulation on
Unstructured Grids using
Multi-dimensional Upwind
Schemes**

Compressible Flow Simulation on Unstructured Grids using Multi-dimensional Upwind Schemes

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 donderdag 12 november 1998 te 13.30 uur

door Edwin Theodorus Antonius VAN DER WEIDE
ingenieur luchtvaart en ruimtevaart
geboren te Oudewater

Dit proefschrift is goedgekeurd door de promotoren:

Prof.dr.ir. P.G. Bakker

Prof.dr.ir. H. Deconinck

Samenstelling promotiecommissie:

Rector Magnificus

Prof.dr.ir. P.G. Bakker

Prof.dr.ir. H. Deconinck

Prof.dr.ir. G. Degrez

Prof.dr. B. van Leer

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

Prof.dr. A.E.P. Veldman

Prof.dr.ir. P. Wesseling

voorzitter

Technische Universiteit Delft, promotor

Université Libre de Bruxelles, promotor

Université Libre de Bruxelles

University of Michigan

Technische Universiteit Delft

Rijks Universiteit Groningen

Technische Universiteit Delft

Contents

1	Introduction	1
1.1	Space discretization techniques	2
1.1.1	The finite-difference method (FDM)	2
1.1.2	The finite-volume method (FVM)	3
1.1.3	The finite-element method (FEM)	5
1.2	The problem with the central discretization	7
1.3	Upwind discretizations	10
1.3.1	The first-order upwind scheme	10
1.3.2	The hybrid upwind/central scheme	11
1.4	Motivation for multi-dimensional upwinding	13
1.5	Structured and unstructured grids	18
1.6	Objectives and structure of the thesis	20
2	Review of the scalar schemes	23
2.1	General concepts	23
2.2	Properties of the distribution schemes	27
2.2.1	Multi-dimensional upwind (\mathcal{MU})	28
2.2.2	Positivity (\mathcal{P})	28
2.2.3	Linearity Preservation (\mathcal{LP})	30
2.2.4	Continuity (\mathcal{C})	32
2.3	Numerical schemes	32
2.3.1	The N-scheme ($\mathcal{MU}, \mathcal{P}, \mathcal{C}$)	32
2.3.2	The PSI or limited N-scheme ($\mathcal{MU}, \mathcal{LP}, \mathcal{P}, \mathcal{C}$)	34
2.3.3	The upwind finite-volume scheme (\mathcal{P}, \mathcal{C})	36
2.3.4	The LDA-scheme ($\mathcal{MU}, \mathcal{LP}, \mathcal{C}$)	37
2.3.5	The Galerkin-scheme ($\mathcal{LP}, \mathcal{C}$)	37
2.3.6	The Lax-Wendroff-scheme ($\mathcal{LP}, \mathcal{C}$)	38
2.3.7	The SUPG-scheme ($\mathcal{LP}, \mathcal{C}$)	38
2.4	Treatment of non-constant advection speeds	41
2.5	The advection-diffusion equation	42
2.6	Advection schemes in 3D	45
2.6.1	The N-scheme ($\mathcal{MU}, \mathcal{P}, \mathcal{C}$)	46
2.6.2	The PSI or limited N-scheme ($\mathcal{MU}, \mathcal{LP}, \mathcal{P}, \mathcal{C}$)	47
2.6.3	The upwind finite-volume scheme (\mathcal{P}, \mathcal{C})	49

2.6.4	The LDA-scheme ($\mathcal{M}\mathcal{U}, \mathcal{L}\mathcal{P}, \mathcal{C}$)	50
2.6.5	The Lax-Wendroff scheme ($\mathcal{L}\mathcal{P}, \mathcal{C}$)	51
2.6.6	The SUPG-scheme ($\mathcal{L}\mathcal{P}, \mathcal{C}$)	51
2.7	Summary of the advection schemes	51
3	System schemes	53
3.1	General concepts	53
3.2	Properties of the system schemes	55
3.3	System schemes	56
3.3.1	The system N-scheme	56
3.3.2	The system PSI-scheme	58
3.3.3	The upwind finite-volume scheme	59
3.3.4	The system LDA-scheme	60
3.3.5	The system Lax-Wendroff scheme	60
3.3.6	The system SUPG-scheme	60
4	The time integration method	63
4.1	Runge-Kutta methods	64
4.2	The backward Euler method	64
4.2.1	Jacobian computation	66
4.2.2	Linear iterative solvers	69
4.2.3	Memory requirements	73
4.3	The parallelization technique	73
5	The Euler equations	77
5.1	Conservative linearization	78
5.2	Boundary conditions	79
5.2.1	Supersonic inlet	80
5.2.2	Supersonic outlet	80
5.2.3	Subsonic inlet	80
5.2.4	Subsonic outlet	83
5.2.5	Far field	83
5.2.6	Impermeable wall	83
5.2.7	Symmetry plane	84
5.2.8	Implicit boundary conditions	85
5.3	Preconditioning	86
5.3.1	The two-dimensional case	86
5.3.2	The three-dimensional case	90
5.4	Results	92
5.4.1	Subsonic flow over a NACA-0012 airfoil	92
5.4.2	Supersonic channel flow	97
5.4.3	M6 wing	102

6	The Navier-Stokes equations	109
6.1	Walls with no-slip boundary conditions	112
6.2	Turbulence models	112
6.2.1	Discretization of the turbulent transport equations	113
6.2.2	The Spalart-Allmaras model	114
6.2.3	The $k - \omega$ model	116
6.2.4	The BSL model	118
6.2.5	The SST model	119
6.3	Results	119
6.3.1	Hyperboloid flare	120
6.3.2	Transonic turbulent channel, Détery's Experiment C	123
6.3.3	Rocket model with exhaust nozzle	131
6.3.4	Ogive cylinder	137
7	Conclusions and future prospects	149
7.1	Conclusions	149
7.1.1	The spatial discretization	149
7.1.2	Time integration and parallelization	150
7.1.3	Euler results	150
7.1.4	Navier-Stokes results	152
7.2	Prospects	154
7.2.1	The spatial discretization	154
7.2.2	The nonlinear iterative solver	154
7.2.3	Euler/Navier-Stokes solutions	155
A	Existence proof of $\left(\sum_m K_m^-\right)^{-1}$	157
B	Energy-stability of the N-scheme	159
C	Symmetrizability of the Euler equations	163
	Summary	177
	Samenvatting	179
	Acknowledgments	181
	Curriculum Vitae	183

Chapter 1

Introduction

Fluid dynamics plays an important role in today's society, from industrial processes, such as the production of steel and coatings, to means of transport like cars, trains, ships and aircraft. Historically fluid dynamics was divided into a theoretical and an experimental branch. While the former attempts to (approximately) solve the governing equations by means of known mathematical methods such as small disturbance theory, asymptotic analysis, hodograph transformation etc., the latter performs experiments and develops experimental techniques, which lead to a better understanding of the physical phenomena involved in the flow field. Due to the enormous growth in computer power, both in speed and memory, over the last 30 years, to these two branches has been added a third one, computational fluid dynamics (CFD). Like the theoretical branch, it tries to solve the governing equations, but now with numerical methods rather than analytical ones.

In the aerospace industry, CFD has gained an important place in the design cycle of a new aircraft. Currently simulations of the three-dimensional Reynolds-Averaged Navier-Stokes (RANS) equations over complete aircraft configurations are being used in the final design phase, reducing the number of even more expensive and time consuming wind-tunnel tests. Although the belief (in the early eighties) that everything could be done numerically and experiments would no longer be needed much, has turned out to be over-optimistic (mainly because of the very difficult task of closing the set of RANS equations by a suitable turbulence model), CFD has led to a shift of the development procedure in the fluid dynamics world: experiments are now more often used to validate the numerical results, rather than as the only way to obtain accurate information about the flow field. Despite this success of CFD, progress can still be made and hopefully this thesis contributes a small part to this.

1.1 Space discretization techniques

Many physical problems can be modeled by a conservation law, a set of partial differential equations of the form

$$\frac{\partial U(\vec{x})}{\partial t} + \frac{\partial F_i(U(\vec{x}))}{\partial x_i} = S(U(\vec{x})) \quad (1.1)$$

on a certain spatial domain Ω . Here $U(\vec{x})$ is the vector of conserved variables, $F_i(U)$ the flux vector in the x_i direction, $S(U)$ the source term and $\vec{x} = x_i \vec{1}_{x_i}$ the position vector. Use is made of the Einstein summation convention and, unless stated otherwise, this convention will be used in the entire thesis.

The set of equations (1.1) are completed by appropriate boundary conditions on Γ , the boundaries of Ω . It must be noted that equations of the type (1.1) have been derived under some simplifying assumptions, for example: the medium is a continuum, or: the stresses are linear functions of the strain rates (Newtonian fluid) etc.

It is the task of the discretization technique to transform the differential equation (1.1) and corresponding boundary conditions into a set of algebraic equations, which can be solved numerically. As the number of "points" where the solution is known changes from infinite in the original problem to a finite number in the discrete problem, it is inevitable that during this transformation an error is introduced, the discretization error. In the following sections the three most well-known discretization techniques are presented, the finite-difference method, the finite-volume method and the finite-element method.

1.1.1 The finite-difference method (FDM)

As the name implies, in the finite-difference method the derivatives occurring in equation (1.1) are approximated by finite differences. Consider, for example, the one-dimensional steady advection equation:

$$u \frac{d\phi}{dx} = 0 \quad (1.2)$$

on a Cartesian grid with step size h , see figure 1.1, with advection speed u . A

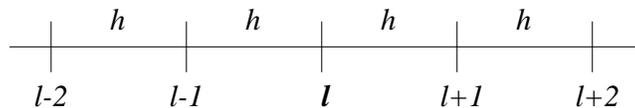


Figure 1.1: One-dimensional Cartesian finite-difference grid

Taylor series expansion in the neighborhood of point x_l gives:

$$\phi_{l+1} = \phi_l + \left(\frac{d\phi}{dx} \right)_l h + \frac{1}{2} \left(\frac{d^2\phi}{dx^2} \right)_l h^2 + O(h^3), \quad (1.3)$$

$$\phi_{l-1} = \phi_l - \left(\frac{d\phi}{dx}\right)_l h + \frac{1}{2} \left(\frac{d^2\phi}{dx^2}\right)_l h^2 + O(h^3). \quad (1.4)$$

Subtracting both equations and dividing the result by h results in:

$$\left(\frac{d\phi}{dx}\right)_l = \frac{\phi_{l+1} - \phi_{l-1}}{2h} + O(h^2), \quad (1.5)$$

and thus the first term in the right hand side is a second-order approximation of $\frac{d\phi}{dx}$ in point x_l . Substituting the expression (1.5) into the differential equation (1.2) for every grid point, results in a set of algebraic equations, which can be solved if the boundary conditions are taken into account.

The approximation (1.5) is called a central discretization of the first-order derivative, because it is symmetric with respect to the neighbors that occur in the expression for $(\frac{d\phi}{dx})_l$. Alternatively either equation (1.3) or (1.4) can be used separately to approximate $(\frac{d\phi}{dx})_l$, which leads to:

$$\left(\frac{d\phi}{dx}\right)_l = \frac{\phi_{l+1} - \phi_l}{h} + O(h), \quad (1.6)$$

$$\left(\frac{d\phi}{dx}\right)_l = \frac{\phi_l - \phi_{l-1}}{h} + O(h), \quad (1.7)$$

respectively. These types of differences are called one-sided differences, because only one neighbor is involved in the expressions. They are also known as upwind differences if they are taken in the appropriate direction, *i.e.* if the advection speed u is positive, equation (1.7) is used, otherwise (1.6) is needed. Equations (1.6) and (1.7) only give a first-order approximation to $(\frac{d\phi}{dx})_l$ and consequently the discretization error will be larger than for the central discretization (1.5). Despite this, upwind-differences play an important role in the numerical treatment of advection-dominated problems, see section 1.2. Higher order, either central or upwind, approximations can be obtained by taking into account the nodes $l - 2$ and $l + 2$, see figure 1.1.

1.1.2 The finite-volume method (FVM)

In the finite-volume method the differential form of the conservation law is transformed into the integral form by integrating equation (1.1) over a control volume V . Using Gauss's theorem for the spatial derivatives results in:

$$\int_V \frac{\partial U}{\partial t} dV + \oint_{\partial V} F_i \cdot n_i^{\text{ext}} d\partial V = \int_V S dV. \quad (1.8)$$

Again U is the vector of conserved variables, F_i the flux vector in the x_i direction and S the source term. Additionally n_i^{ext} is the component of the unit outward normal in the x_i direction, see figure 1.2. The one-dimensional advection equa-

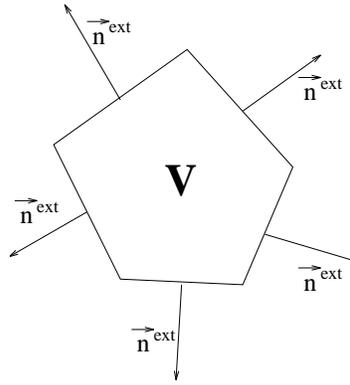


Figure 1.2: Control volume V and its unit outward normals.

tion (1.2) is written in conservation form, assuming a constant advection speed u :

$$\frac{du\phi}{dx} = \frac{df}{dx} = 0 \implies \oint_{\partial V} f n_x d\partial V = 0, \quad (1.9)$$

where $f = u\phi$ is the flux function. For the finite-volume grid given in figure 1.3 this gives at point l :

$$f_{l+\frac{1}{2}} - f_{l-\frac{1}{2}} = 0. \quad (1.10)$$

The accuracy of the discretization is determined by the way in which the flux functions at the interfaces, $f_{l+\frac{1}{2}}$ and $f_{l-\frac{1}{2}}$, are computed. The central approximation:

$$f_{l+\frac{1}{2}} = \frac{1}{2}(f_{l+1} + f_l) + O(h^2), \quad f_{l-\frac{1}{2}} = \frac{1}{2}(f_l + f_{l-1}) + O(h^2), \quad (1.11)$$

leads to a second-order accurate discretization. Moreover on a Cartesian grid and for a constant advection speed u it is obvious that, apart from the boundary condition treatment, the resulting set of algebraic equations is identical to the one obtained with the finite-difference method in combination with the central discretization (1.5).

The upwind discretization is obtained by the following approximations:

$$f_{l+\frac{1}{2}} = f_l + O(h), \quad f_{l-\frac{1}{2}} = f_{l-1} + O(h) \quad (1.12)$$

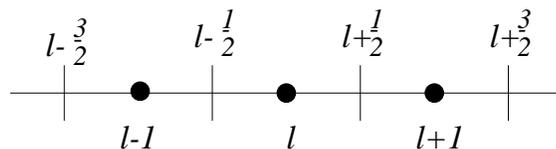


Figure 1.3: One-dimensional finite-volume grid

for a positive u , and:

$$f_{l+\frac{1}{2}} = f_{l+1} + O(h), \quad f_{l-\frac{1}{2}} = f_l + O(h) \quad (1.13)$$

in case the advection speed is negative. Again these approximations are only first-order accurate and so is the resulting discretization based on them.

Note that in two and three space dimensions not only the definition of the flux function but also the numerical integration rule used in equation (1.8) determines the accuracy of the discretization. The overall accuracy is given by the lowest approximation of the two and therefore it is pointless to use for example a fourth-order integration rule, while the flux functions are only second-order accurate and vice versa.

1.1.3 The finite-element method (FEM)

The philosophy behind the finite-element method is somewhat different than the previously discussed discretization techniques. The solution of the problem (1.1) is assumed to have the following form:

$$U(\vec{x}, t) = \sum_{l=1}^{\#\text{nodes}} N_l(\vec{x}) U_l(t), \quad (1.14)$$

where $\vec{x} = x_i \vec{1}_{x_i}$ is the position vector, U_l the value of U in node l and $N_l(\vec{x})$ the nodal basis functions of U_l which must have the property:

$$N_l(\vec{x}_k) = \delta_{kl}, \quad (1.15)$$

where δ_{kl} is the Kronecker delta function. In principle any function which obeys condition (1.15) can be used. In practice these functions are piecewise-continuous polynomials. An example is seen in figure 1.4.

The second step consists of transforming the conservation law (1.1) into an equivalent integral form, also called a weighted residual or variational formulation. This is done by multiplying equation (1.1) by weight functions $w_l(\vec{x})$ and integrating the result over the domain Ω , which leads to the following set of equations:

$$\iint_{\Omega} w_l \frac{\partial}{\partial t} \left(\sum_{k=1}^{\#\text{nodes}} N_k U_k \right) d\Omega + \iint_{\Omega} w_l \frac{\partial F_i}{\partial x_i} d\Omega = \iint_{\Omega} w_l S d\Omega, \quad (1.16)$$

where the counter l varies from 1 to the number of nodes, *i.e.* there is one relation (1.16) for each node. The reason why equation (1.16) must hold for all w_l is that (1.1) and (1.16) are equivalent if and only if equation (1.16) holds for all possible choices of the weight function. Practically this is not feasible and therefore this choice is limited to all w_l . It can be proved [56] that equation (1.16) is the best possible approximation to (1.1) in the finite-dimensional function space with the w_l 's as basis. The weight functions w_l must obey certain conditions; both

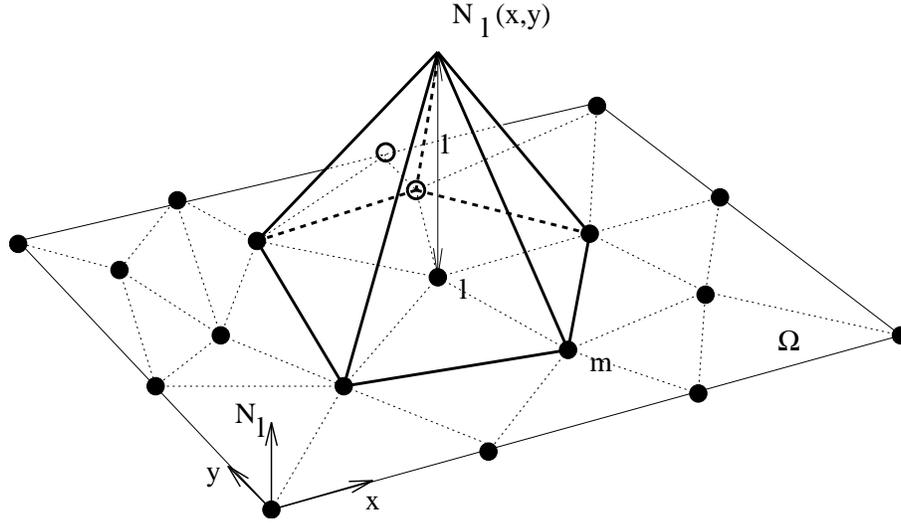


Figure 1.4: Two-dimensional piecewise linear interpolation function for node l .

w_l and its first-order derivatives must be square integrable, see [56]. A function is square integrable if the integral of the square of that function over a certain domain exists. More formally this means w_l must belong to $H^1(\Omega)$, the Hilbert space of square integrable functions with square integrable first-order derivatives. Note that for systems of equations w_l is a matrix. If (parts of) the flux functions F_i contain derivatives of U , for example the viscous fluxes in the Navier-Stokes equations, these terms in equation (1.16) are rewritten as:

$$\iint_{\Omega} w_l \frac{\partial F_i}{\partial x_i} d\Omega = - \iint_{\Omega} \frac{\partial w_l}{\partial x_i} F_i d\Omega + \oint_{\partial\Omega} w_l F_i \cdot n_i^{\text{ext}} d\partial\Omega, \quad (1.17)$$

where use is made of the Gauss theorem. Equations (1.16) and (1.17) lead to a set of coupled, ordinary differential equations in time, which can be solved by standard time-integration methods. Like the FDM and FVM the steady problem is given by a set of algebraic equations. If the weight functions w_l are identical to the basis functions N_l , the classical Galerkin FEM is obtained. Otherwise the method is called a Petrov-Galerkin FEM.

Again consider the one-dimensional advection equation (1.2) on the Cartesian grid of figure 1.1. The piecewise linear basis function for nodes $l-1$, l and $l+1$, are shown in figure 1.5. The weighted residual formulation becomes:

$$\int_{\Omega} w_l(x) u \sum_{k=1}^{\#\text{nodes}} \frac{dN_k(x)}{dx} \phi_k d\Omega = 0. \quad (1.18)$$

Usually the weight function $w_l(x)$ is chosen to be zero if $N_l(x)$ vanishes and consequently only the nodes $l-1$, l and $l+1$ give a nonzero contribution in the sum

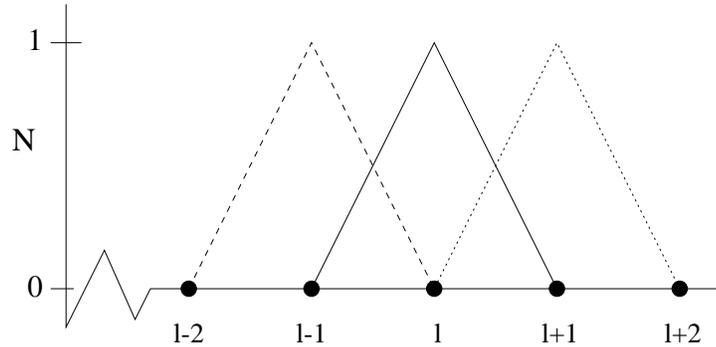


Figure 1.5: Piecewise-linear, tent-shaped, one-dimensional basis functions for nodes $l - 1$ (dashed), l (solid) and $l + 1$ (dotted).

of equation (1.18). For the Galerkin method, $w_l(x) = N_l(x)$, on a uniform grid the weighted residual formulation (1.18) for node l becomes:

$$\frac{u}{2} (\phi_{l+1} - \phi_{l-1}) = 0, \quad (1.19)$$

which is identical to the relations obtained with the central discretization (1.5) for the FDM and the central approximation (1.11) for the FVM.

1.2 The problem with the central discretization

Despite the fact that the central discretization of the first-order derivative, equation (1.5), results in a second-order approximation, it cannot be used in practice, because of stability reasons. This explains the use of upwind schemes for advection (-dominated) problems. The instability of the central discretization will be analyzed for the one-dimensional steady advection-diffusion equation

$$u \frac{d\phi}{dx} - \nu \frac{d^2\phi}{dx^2} = 0, \quad (1.20)$$

with boundary conditions

$$\begin{aligned} \phi &= \phi_{\text{in}} & x &= 0 \\ \phi &= \phi_{\text{out}} & x &= 1 \end{aligned} \quad (1.21)$$

on the domain $\Omega = [0, 1]$. For simplicity the advection speed u and the diffusion coefficient ν are taken constant and positive. The grid is the Cartesian grid of figure 1.1 with step size h . As explained in the previous sections, under these conditions the three discretization techniques all lead to the same result for the advective part. Similarly this can be shown for the diffusive part as well and therefore it suffices to work out the stability analysis only for the finite-difference method.

The exact solution of equations (1.20) and (1.21) is:

$$\phi(x) = \phi_{\text{in}} + (\phi_{\text{out}} - \phi_{\text{in}}) \frac{\exp(2Pe x) - 1}{\exp(2Pe) - 1}, \quad Pe = \frac{u \times 1}{2\nu}, \quad (1.22)$$

where the 1 in the numerator of the expressions for Pe has been added to indicate a unit length. The non-dimensional Peclet number Pe is the ratio between the advection and diffusion terms. In figure 1.6 two solutions are shown for different values of this non-dimensional quantity. Clearly the diffusion-dominated case, low Pe , is totally different from the advection dominated case, high Pe . While the former shows a nearly linear variation between the inflow and outflow boundary (the limiting case for $u = 0$), the latter has a constant solution in almost the entire domain in combination with a thin outflow boundary layer. A similar behavior is seen in high Reynolds number flow fields and therefore this model equation is well suited to demonstrate the problem of the central discretization for advection dominated flows.

The central discretization of the advection term at point x_l is given in equation (1.5). A similar analysis can be done for the second-order derivative, which results in:

$$\left(\frac{d^2\phi}{dx^2} \right)_l = \frac{\phi_{l+1} - 2\phi_l + \phi_{l-1}}{h^2} + O(h^2). \quad (1.23)$$

Consequently the discrete version of equation (1.20) at point x_l becomes:

$$u \frac{\phi_{l+1} - \phi_{l-1}}{2h} - \nu \frac{\phi_{l+1} - 2\phi_l + \phi_{l-1}}{h^2} = 0. \quad (1.24)$$

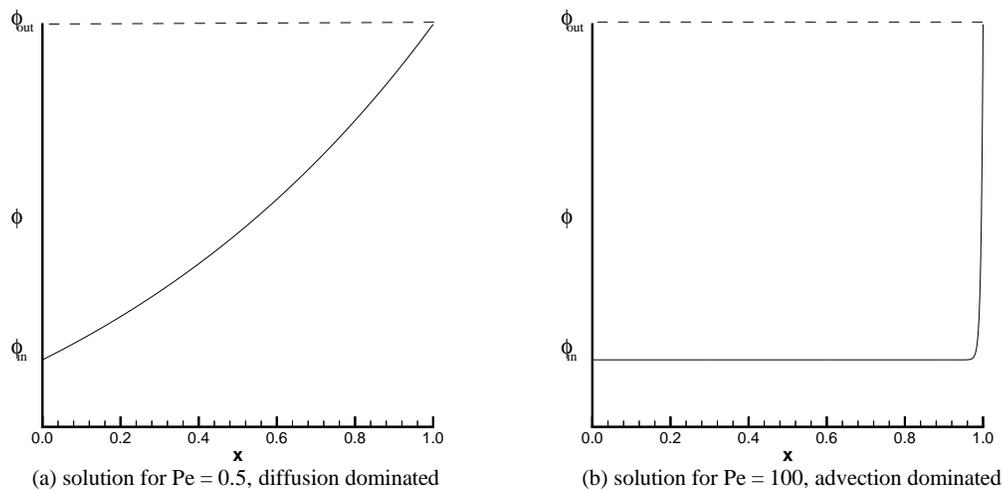


Figure 1.6: Exact solutions of the advection-diffusion equation: (a) low Pe number, (b) high Pe number.

In combination with the boundary conditions (1.21) this gives a set of linear algebraic equations, which can easily be solved:

$$\phi_l = \phi_{\text{in}} + (\phi_{\text{out}} - \phi_{\text{in}}) \frac{r^l - 1}{r^N - 1}, \quad (1.25)$$

where

$$r = \frac{1 + Pe^h}{1 - Pe^h}, \quad Pe^h = \frac{uh}{2\nu} = Pe \times h. \quad (1.26)$$

The node $l = 0$ corresponds with the inflow and $l = N$ with the outflow boundary. On the Cartesian grid use can be made of the relation:

$$\exp(2Pe x_l) = \exp\left(2Pe^h \frac{x_l}{h}\right) = \exp(2Pe^h l) = (\exp(2Pe^h))^l, \quad (1.27)$$

and the exact solution (1.22) at point x_l can be written as:

$$\phi_l = \phi_{\text{in}} + (\phi_{\text{out}} - \phi_{\text{in}}) \frac{(\exp(2Pe^h))^l - 1}{(\exp(2Pe^h))^N - 1}. \quad (1.28)$$

Compared to the solution of the central discretization (1.25) it is clear that r , equation (1.26), must be an approximation to $\exp(2Pe^h)$. Substituting the power series

$$\frac{1}{1 - Pe^h} = \sum_{k=0}^{\infty} (Pe^h)^k \quad (1.29)$$

into equation (1.26) results in:

$$\begin{aligned} r &= (1 + Pe^h) \sum_{k=0}^{\infty} (Pe^h)^k = 1 + 2 \sum_{k=1}^{\infty} (Pe^h)^k \\ &= 1 + 2Pe^h + 2(Pe^h)^2 + 2(Pe^h)^3 + \dots \end{aligned} \quad (1.30)$$

Comparison with the power series of $\exp(2Pe^h)$,

$$\exp(2Pe^h) = \sum_{k=0}^{\infty} \frac{1}{k!} (2Pe^h)^k = 1 + 2Pe^h + 2(Pe^h)^2 + \frac{4}{3}(Pe^h)^3 + \dots, \quad (1.31)$$

shows that r is a third-order approximation to $\exp(2Pe^h)$. However, the power series (1.29) has a radius of convergence $|Pe^h| < 1$ and consequently the approximation (1.26) is only valid for $|Pe^h| < 1$. For $|Pe^h| > 1$, r^l in equation (1.25) changes sign from grid point to grid point and the solution becomes oscillatory, as can be seen in figure 1.7, which illustrates a stable ($|Pe^h| < 1$) and an unstable ($|Pe^h| > 1$) solution.

To give an idea about the number of points needed to obtain a stable solution with the central discretization for a high Reynolds number flow, consider the

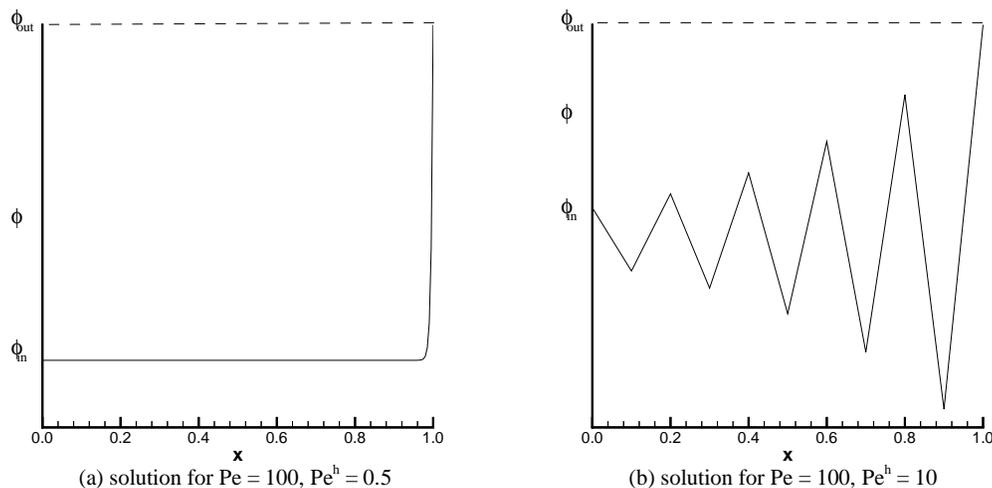


Figure 1.7: Numerical solutions of the centrally-discretized advection-diffusion equation: (a) stable ($|Pe^h| < 1$), (b) unstable ($|Pe^h| > 1$).

following example. A low estimate of the typical Reynolds number for a modern transonic airliner in cruise conditions is $O(10^7)$ and consequently the number of points per coordinate direction is of the same magnitude in order to have $Re^h < 1$. This leads to a total of $O(10^{21})$ points in three space dimensions, which is far beyond the capabilities of current supercomputers, $O(10^7)$ points, and even more than required for a direct numerical simulation of turbulence, $O(10^{16})$ points. Therefore the central discretization is not practical, unless artificial dissipation terms are added, see [55]. These schemes are robust and relatively cheap, but the quality of the solution is not as good as what modern upwind schemes give.

1.3 Upwind discretizations

1.3.1 The first-order upwind scheme

An alternative for the central difference is to use the first-order upwind difference (1.7) for the advective term, which leads to the following discrete approximation of equation (1.20):

$$u \frac{\phi_l - \phi_{l-1}}{h} - \nu \frac{\phi_{l-1} - 2\phi_l + \phi_{l+1}}{h^2} = 0. \quad (1.32)$$

The solution to this problem is also given by equation (1.25), but with a different definition of r :

$$r = 1 + 2Pe^h. \quad (1.33)$$

This is only a second-order approximation to $\exp(2Pe^h)$ and consequently an order of accuracy is lost compared to the central discretization. This is to be expected,

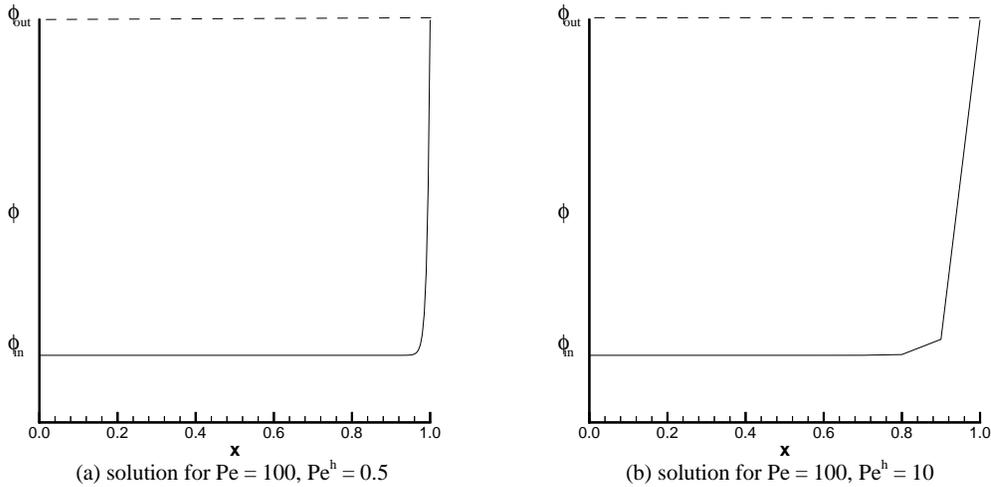


Figure 1.8: Numerical solutions of the first-order upwind discretized advection-diffusion equation: (a) ($|Pe^h| < 1$), (b) ($|Pe^h| > 1$). Both are stable.

as equation (1.7) is only a first-order approximation to $(\frac{d\phi}{dx})_l$. However, equation (1.33) is valid for any Pe^h and remains positive. Therefore a stable scheme is obtained independent of the cell Peclet number, as illustrated for two solutions with different Pe^h , figure 1.8.

1.3.2 The hybrid upwind/central scheme

The first-order upwind scheme is indeed stable, but its drawback is that it is too diffusive. This can be seen by rewriting equation (1.32) as:

$$u \frac{\phi_{l+1} - \phi_{l-1}}{2h} - (\nu + \hat{\nu}) \frac{\phi_{l-1} - 2\phi_l + \phi_{l+1}}{h^2} = 0, \quad (1.34)$$

where $\hat{\nu} = \frac{1}{2}|u|h$ is the numerical diffusion coefficient. Equation (1.34) is a central discretization of a modified problem, with ν replaced by $(\nu + \hat{\nu})$. Hence upwinding of the advection term is identical to adding a (centrally discretized) artificial dissipation term, $\hat{\nu} \frac{d^2\phi}{dx^2}$, to the central discretization. Consequently sharp gradients are smeared out, especially when $Pe^h \gg 1$, for the ratio $\frac{\hat{\nu}}{\nu} = Pe^h$. As this is the normal situation for advection dominated problems, this is unacceptable in terms of accuracy and a remedy must be found.

One possibility is to use high-order upwind approximations to the advection term, which have to be nonlinear to assure monotonicity [44, 98], see e.g. [96] or in a slightly different context [129]. This technique is usually used for the finite-difference and finite-volume method and very good results have been obtained. Its drawback is that the computational stencil is increased, *i.e.* also the neighbors of the neighbors are needed, which becomes rather complicated on unstructured grids in multiple space dimensions [10, 133]. An alternative, which keeps the compact stencil (only immediate neighbors are involved), was discovered by Spalding [114]

and is mainly used in the finite-element method, namely a blending between the upwind approximation (1.7) and the central one (1.5). This leads to the discretization:

$$\zeta u \frac{\phi_l - \phi_{l-1}}{h} + (1 - \zeta) u \frac{\phi_{l+1} - \phi_{l-1}}{2h} - \nu \frac{\phi_{l-1} - 2\phi_l + \phi_{l+1}}{h^2} = 0, \quad (1.35)$$

where the blending function $\zeta = \zeta(Pe^h)$ is a function of the cell Peclet number Pe^h . A nodally exact solution of this model problem can be obtained by choosing, see [48]:

$$\zeta(Pe^h) = \coth(Pe^h) - \frac{1}{Pe^h}. \quad (1.36)$$

This function has been plotted in figure 1.9 together with two other choices for $\zeta(Pe^h)$, which do not contain the expensive hyperbolic cotangent function,

- Mizukami's approximation [78]:

$$\zeta = \frac{Pe^h}{1 + Pe^h} \quad (1.37)$$

- Johnson's approximation [57]:

$$\zeta = \max\left(0, 1 - \frac{1}{2Pe^h}\right) \quad (1.38)$$

Note that for $Pe^h > 1$ both functions are above the optimal blending function, see figure 1.9. This is essential, because otherwise not enough diffusion is added to guarantee stability. Solutions of these two alternatives are shown in figure 1.10. Mizukami's blending function gives a slightly better solution for $Pe^h = 10$, while both solutions are clearly better than the first-order upwind solution, see figure 1.8.

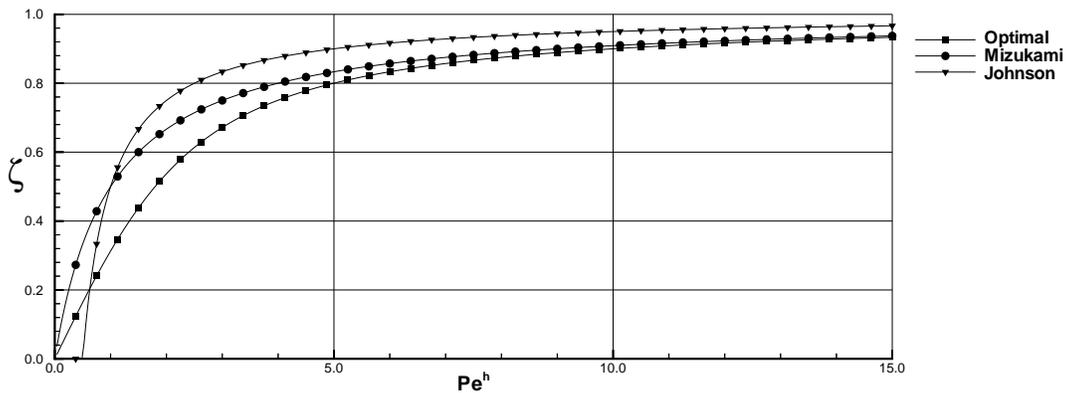


Figure 1.9: Three blending functions for hybrid upwind/central differencing.

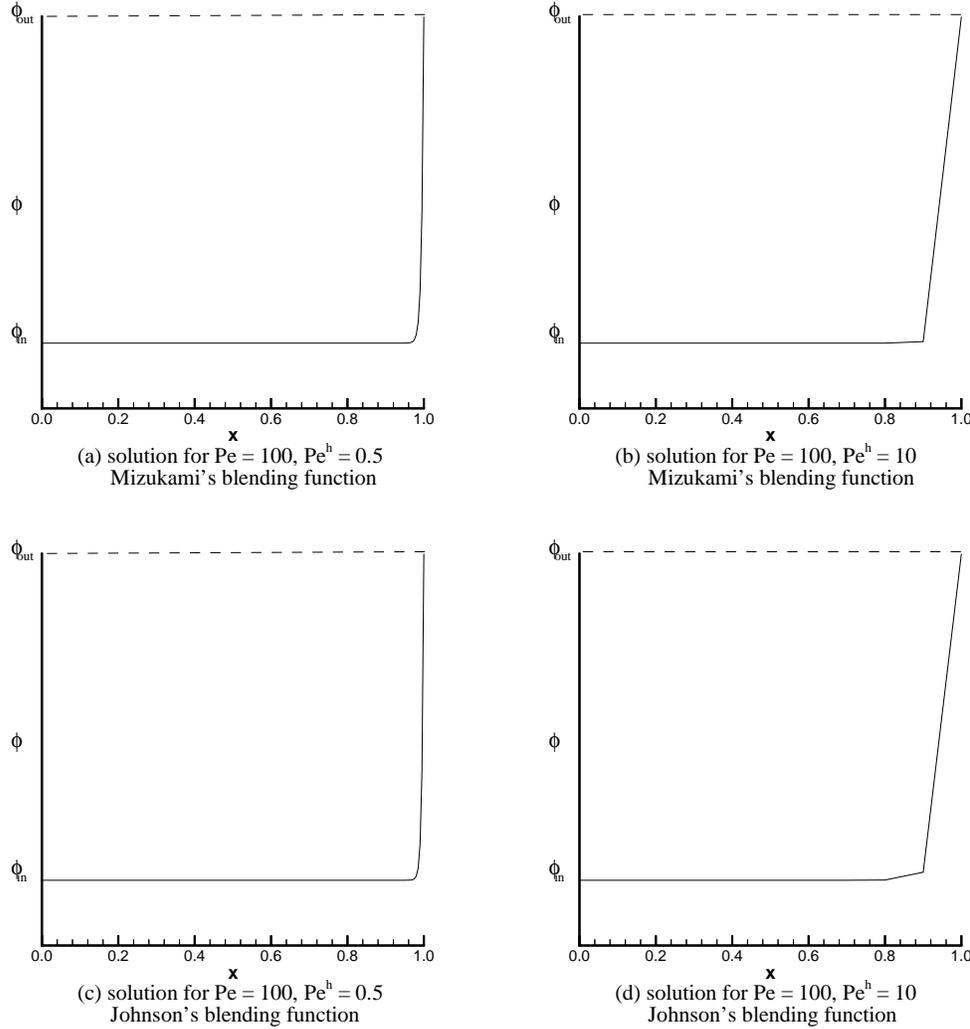


Figure 1.10: Numerical solutions of the hybrid upwind/central discretized advection-diffusion equation, (a) and (c) ($|Pe^h| < 1$), (b) and (d) ($|Pe^h| > 1$).

1.4 Motivation for multi-dimensional upwinding

In the previous section it has been shown that the upwind discretization is a very robust tool for the numerical approximation of the one-dimensional model problem. In this section the extension to two space-dimensions is made and the motivation for the use of multi-dimensional upwind schemes is given.

Consider the two-dimensional, steady advection equation:

$$u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} = 0, \quad (1.39)$$

where, for the sake of convenience, the advection speeds u, v are assumed positive and constant. For reasons of clarity the summation convention is not used here. The most common method to discretize equation (1.39) is the dimensionally split

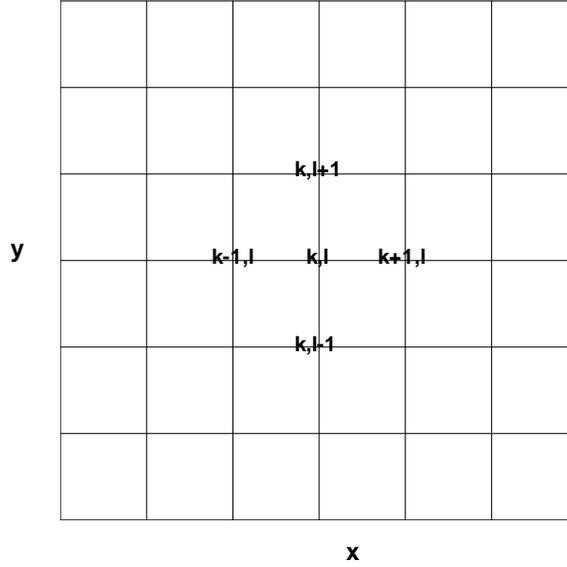


Figure 1.11: Uniform 2D Cartesian grid.

approach, *i.e.* the x and y derivatives are treated in a one-dimensional manner. On the Cartesian grid, shown in figure 1.11, the first-order, dimensionally-split upwind discretization at point (x_k, y_l) becomes for $u, v > 0$:

$$u \frac{\phi_{k,l} - \phi_{k-1,l}}{h} + v \frac{\phi_{k,l} - \phi_{k,l-1}}{h} = 0. \quad (1.40)$$

A simple Taylor series analysis shows that

$$u \frac{\phi_{k,l} - \phi_{k-1,l}}{h} = u \left(\frac{\partial \phi}{\partial x} \right)_{k,l} - \hat{\nu}_{xx} \left(\frac{\partial^2 \phi}{\partial x^2} \right)_{k,l} + O(h^2), \quad (1.41)$$

$$v \frac{\phi_{k,l} - \phi_{k,l-1}}{h} = v \left(\frac{\partial \phi}{\partial y} \right)_{k,l} - \hat{\nu}_{yy} \left(\frac{\partial^2 \phi}{\partial y^2} \right)_{k,l} + O(h^2), \quad (1.42)$$

with the numerical diffusion coefficients:

$$\hat{\nu}_{xx} = \frac{uh}{2}, \quad \hat{\nu}_{yy} = \frac{vh}{2}. \quad (1.43)$$

Consequently the first-order dimensionally split upwind discretization is a central second-order approximation to:

$$u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} - \hat{\nu}_{xx} \frac{\partial^2 \phi}{\partial x^2} - \hat{\nu}_{yy} \frac{\partial^2 \phi}{\partial y^2} = 0. \quad (1.44)$$

To analyze the numerical diffusion coefficients (1.43), equation (1.44) is written in a streamline coordinate system (ξ, η) , see figure 1.12:

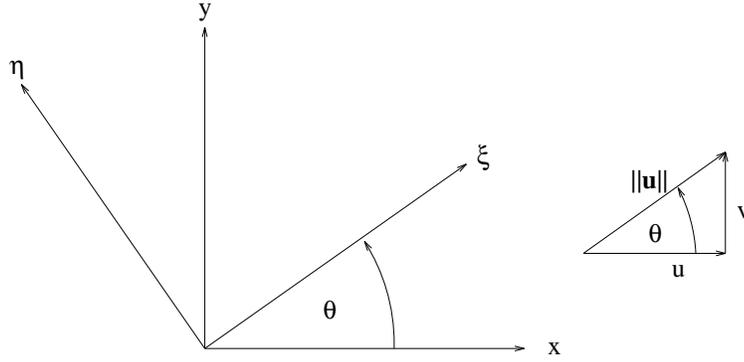


Figure 1.12: Relation between the Cartesian (x, y) coordinate system and the streamline (ξ, η) coordinate system.

$$\|\mathbf{u}\| \frac{\partial \phi}{\partial \xi} - \hat{\nu}_{\xi\xi} \frac{\partial^2 \phi}{\partial \xi^2} - \hat{\nu}_{\xi\eta} \frac{\partial^2 \phi}{\partial \xi \partial \eta} - \hat{\nu}_{\eta\eta} \frac{\partial^2 \phi}{\partial \eta^2} = 0, \quad (1.45)$$

where $\|\mathbf{u}\| = \sqrt{u^2 + v^2}$,

$$\cos \theta = \frac{u}{\|\mathbf{u}\|}, \quad \sin \theta = \frac{v}{\|\mathbf{u}\|}, \quad (1.46)$$

see figure 1.12, and the diffusion coefficients are:

$$\hat{\nu}_{\xi\xi} = \hat{\nu}_{xx} \cos^2 \theta + \hat{\nu}_{yy} \sin^2 \theta = \frac{\|\mathbf{u}\| h}{2} (\cos^3 \theta + \sin^3 \theta), \quad (1.47)$$

$$\hat{\nu}_{\xi\eta} = 2(\hat{\nu}_{yy} - \hat{\nu}_{xx}) \sin \theta \cos \theta = \frac{\|\mathbf{u}\| h}{2} \sin 2\theta (\sin \theta - \cos \theta), \quad (1.48)$$

$$\hat{\nu}_{\eta\eta} = \hat{\nu}_{xx} \sin^2 \theta + \hat{\nu}_{yy} \cos^2 \theta = \frac{\|\mathbf{u}\| h}{2} \cos \theta \sin \theta (\sin \theta + \cos \theta). \quad (1.49)$$

In figure 1.13 the values of these three coefficients as a function of the flow angle θ are depicted. Equation (1.45) shows that the overall effect on the numerical solution is determined by the product of the diffusion coefficient and its corresponding second-order derivative. For advection-dominated flows the derivatives normal to the streamline are much larger than those in the streamwise direction (for pure advection, $\vec{u} \cdot \vec{\nabla} \phi = 0$, ϕ is even constant in the streamwise direction (u, v)). Therefore the cross-wind diffusion coefficient $\hat{\nu}_{\eta\eta}$ is of much more concern than the streamline, $\hat{\nu}_{\xi\xi}$, and the mixed, $\hat{\nu}_{\xi\eta}$ diffusion coefficients. As can be seen in figure 1.13, $\hat{\nu}_{\eta\eta}$ is large if the flow is not aligned with the grid lines and consequently flow features are smeared out considerably for these cases.

An examination of the computational stencil (1.40) of the dimensionally split upwind scheme shows that the point $(k-1, l-1)$ is not used, although from a physical point of view there is no reason not to use it, see figure 1.15. This was the motivation for the use of multi-dimensional upwind schemes, in which the upwind

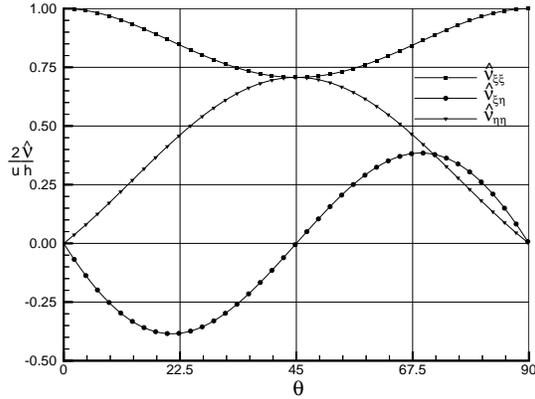


Figure 1.13: Numerical diffusion coefficients for the first-order dimensionally-split upwind scheme as a function of the flow angle θ .

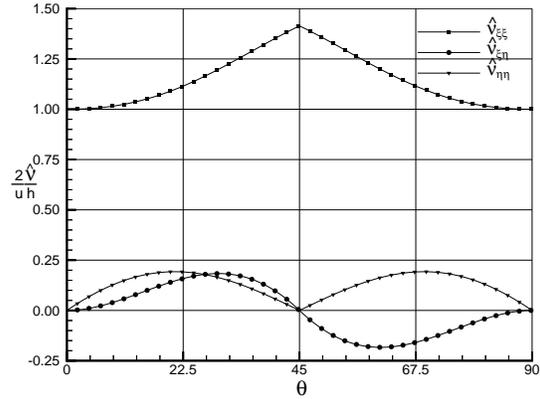


Figure 1.14: Numerical diffusion coefficients for the first-order multi-dimensional upwind scheme as a function of the flow angle θ .

direction is not determined by the grid lines, but by the streamline [93, 92, 94]. Equation (1.39) is discretized directly rather than being first split into its Cartesian components, followed by the discretization of the one-dimensional subproblems. The result of this multi-dimensional approach is:

$$\|\mathbf{u}\| \left(\frac{\partial \phi}{\partial \xi} \right)_{k,l} = \|\mathbf{u}\| \frac{\phi_{k,l} - \phi_{in}}{\Delta \xi} + \|\mathbf{u}\| O(\Delta \xi) \tag{1.50}$$

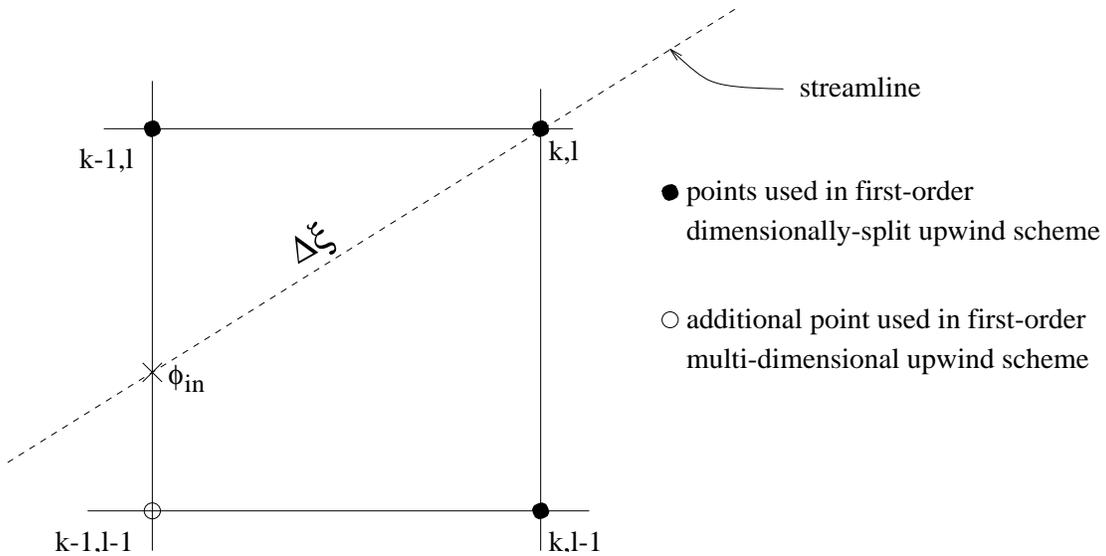


Figure 1.15: Computational stencil for the dimensionally-split upwind scheme and multi-dimensional upwind scheme, both first order.

where ϕ_{in} is a linear interpolated value, see figure 1.15:

$$\phi_{\text{in}} = \frac{\min(u, v)\phi_{k-1, l-1} + \max(0, u - v)\phi_{k-1, l} + \max(0, v - u)\phi_{k, l-1}}{\max(u, v)} \quad (1.51)$$

and

$$\Delta\xi = \frac{\|\mathbf{u}\|}{\max(u, v)}h. \quad (1.52)$$

The min and max functions are used in equations (1.51) and (1.52) to obtain a formula which is valid for both $u > v$, corresponding to figure 1.15, and $u < v$. Note that the interpolation is either between $\phi_{k-1, l-1}$ and $\phi_{k-1, l}$ ($u > v$), or between $\phi_{k-1, l-1}$ and $\phi_{k, l-1}$ ($u < v$). A similar argument holds for the length $\Delta\xi$.

Substituting (1.51) and (1.52) into equation (1.50) gives:

$$\begin{aligned} \|\mathbf{u}\| \left(\frac{\partial\phi}{\partial\xi} \right)_{k, l} &= \|\mathbf{u}\| O(\Delta\xi) + \\ &\frac{\max(u, v)\phi_{k, l} - \min(u, v)\phi_{k-1, l-1} - \max(0, u - v)\phi_{k-1, l} - \max(0, v - u)\phi_{k, l-1}}{h} \\ &= \underbrace{\|\mathbf{u}\| O(\Delta\xi)}_{\text{Discr. error}} + \underbrace{\frac{(u + v)\phi_{k, l} - u\phi_{k-1, l} - v\phi_{k, l-1}}{h}}_{\text{Dim. split scheme}} \\ &\quad - \underbrace{\frac{h \min(u, v) (\phi_{k, l} - \phi_{k-1, l} - \phi_{k, l-1} + \phi_{k-1, l-1})}{h^2}}_{\text{Correction term}}. \end{aligned} \quad (1.53)$$

So the first-order multi-dimensional upwind discretization equals the first-order dimensionally-split scheme minus a correction term, in which $\frac{\phi_{k, l} - \phi_{k-1, l} - \phi_{k, l-1} + \phi_{k-1, l-1}}{h^2}$ is a second-order approximation to $\frac{\partial^2\phi}{\partial x\partial y}$ at point $(k - \frac{1}{2}, l - \frac{1}{2})$. As

$$\left(\frac{\partial^2\phi}{\partial x\partial y} \right)_{k-\frac{1}{2}, l-\frac{1}{2}} = \left(\frac{\partial^2\phi}{\partial x\partial y} \right)_{k, l} + O(h), \quad (1.54)$$

and thus:

$$h \min(u, v) \left(\frac{\partial^2\phi}{\partial x\partial y} \right)_{k-\frac{1}{2}, l-\frac{1}{2}} = h \min(u, v) \left(\frac{\partial^2\phi}{\partial x\partial y} \right)_{k, l} + O(h^2), \quad (1.55)$$

the first-order multi-dimensional upwind discretization (1.53) of equation (1.39) is a second-order approximation to the equation:

$$u \frac{\partial\phi}{\partial x} + v \frac{\partial\phi}{\partial y} - \hat{\nu}_{xx}^{MD} \frac{\partial^2\phi}{\partial x^2} - \hat{\nu}_{xy}^{MD} \frac{\partial^2\phi}{\partial x\partial y} - \hat{\nu}_{yy}^{MD} \frac{\partial^2\phi}{\partial y^2} = 0. \quad (1.56)$$

The numerical diffusion coefficients $\hat{\nu}_{xx}^{MD}$ and $\hat{\nu}_{yy}^{MD}$ are identical to $\hat{\nu}_{xx}$ and $\hat{\nu}_{yy}$ of the first-order dimensionally split scheme, equation (1.43), and

$$\hat{\nu}_{xy}^{MD} = h \min(u, v). \quad (1.57)$$

By performing the transformation to the streamline coordinate system, see figure 1.12, the numerical streamline, $\hat{\nu}_{\xi\xi}^{MD}$, mixed, $\hat{\nu}_{\xi\eta}^{MD}$, and crosswind, $\hat{\nu}_{\eta\eta}^{MD}$, diffusion coefficients are obtained:

$$\hat{\nu}_{\xi\xi}^{MD} = \frac{\|\mathbf{u}\|h}{2} \{ \cos^3 \theta + \sin^3 \theta + 2 \cos \theta \sin \theta \min(\sin \theta, \cos \theta) \} \quad (1.58)$$

$$\hat{\nu}_{\xi\eta}^{MD} = \frac{\|\mathbf{u}\|h}{2} \{ \sin 2\theta (\sin \theta - \cos \theta) + 2 \cos 2\theta \min(\sin \theta, \cos \theta) \} \quad (1.59)$$

$$\hat{\nu}_{\eta\eta}^{MD} = \frac{\|\mathbf{u}\|h}{2} \cos \theta \sin \theta \{ \sin \theta + \cos \theta - 2 \min(\sin \theta, \cos \theta) \}. \quad (1.60)$$

These coefficients are plotted in figure 1.14. Comparison with the dimensionally-split scheme, figure 1.13, shows that the important crosswind diffusion coefficient $\hat{\nu}_{\eta\eta}$ is much smaller for the multi-dimensional scheme and consequently the latter is more accurate if the flow is not aligned with the grid. Moreover, as shown by Roe [97], the dimensionally-split upwind scheme is the worst possible choice among all monotonic first-order schemes and the multi-dimensional upwind scheme is the best.

The price paid for minimizing the crosswind diffusion is the increase in the streamline diffusion. As explained earlier this is not an issue for advection dominated flows, but it may have some effect for advection with strong source terms, for example the equations for turbulence modeling or chemical reacting flows, where the variables do change along the streamline. However in general this change is still less than the change normal to the streamline.

1.5 Structured and unstructured grids

In multiple space dimensions the numerical simulation of a flow field consists of three almost independent disciplines:

- 1: The geometry definition.
- 2: The grid generation: (a) surface grid.
(b) volume grid.
- 3: The flow solver.

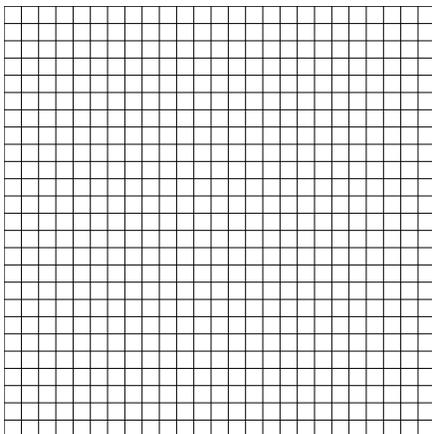
In 2D the geometry definition is a rather trivial task. Usually a set of coordinates is given, which are used to define the geometry by splines. In 3D it is more complicated and normally CAD tools are used for complex configurations.

The third discipline involves the discretization of the governing equations and their solution, the subject of this work, on the computational grid generated by the second discipline.

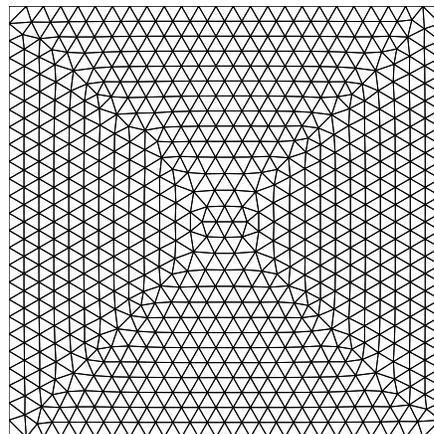
The grid generation process is split into a surface part and a volume part. In two space dimensions the surface grid generation is the creation of a point distribution on the boundaries, based on several criteria like curvature, discontinuities in the slope, proximity of other bodies etc. In 3D it is a generation of a two-dimensional grid on the given geometries, also with the possibility to take the criteria mentioned above into account. The volume grid is built starting from the surface grid.

A distinction may be made between structured and unstructured grids, see figure 1.16. For the former every point is uniquely determined by two indices in 2D and three in 3D. In an unstructured grid, as the name implies, such a structure is absent and both the nodes and the cells are numbered. In addition the cell connectivity, the node numbers of the vertices of each cell, must be given as well. This implies that for unstructured grids more information must be stored, but this is negligible compared to the storage requirements of an implicit solver, as will be shown later chapter 4.

Structured grids of the type of figure 1.16(a) are very restrictive from the geometrical point of view and usually a multi-block approach is used for more complicated geometries. This means that the domain is split into a small number of subdomains, or blocks, in an unstructured way, while the grids inside these blocks are structured. An example is shown in the left picture of figure 1.17. It is clear that this type of grids require a large human input, especially in 3D, and consequently it takes a lot of time to generate them, one of the main bottlenecks which currently prevents the use of Euler and RANS simulations in the early design stages. On the other hand, unstructured grid generation is highly automatic and thus potentially able to reduce the grid generation time considerably. For viscous



(a) structured grid



(b) unstructured grid

Figure 1.16: (a) Structured quadrilateral grid and (b) unstructured triangular grid.

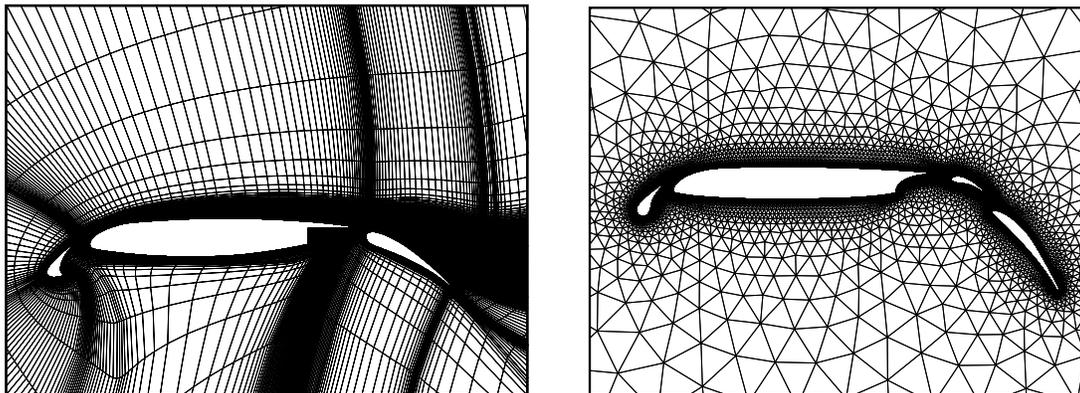


Figure 1.17: Multi-block structured grid (obtained from [104]) and unstructured grid, for multi-element airfoil configurations.

computations it is possible to use a hybrid approach [20, 4], *i.e.* a structured layer of stretched quadrilaterals in the boundary layer combined with triangles in the rest of the domain. An example is given in the right picture of figure 1.17. Hassan *et al.*[90] claim that this approach is too restrictive in 3D and advocate a fully unstructured method.

Another advantage of unstructured grids is that points can be put where they are needed and the grid can be coarsened in less significant regions. So local refinements, e.g. near separation points, do not influence the grid in the far field. On the contrary for structured grids, all local refinements are "transported" to the far field where these points are not needed and consequently more points are used than strictly necessary. This is clearly visible in figure 1.17.

Despite these advantages of unstructured grids, the vast majority of the current RANS computations are performed on structured grids, because (1) the expertise of generating such grids for these problems is more developed than for unstructured viscous grid generation, although the situation is changing [20, 4, 90], and (2) the current dimensionally-split upwind algorithms perform very badly in the stretched triangles in the boundary layer, because the upwind directions are completely misaligned with the flow.

1.6 Objectives and structure of the thesis

The main objective of this thesis is to improve the multi-dimensional upwind discretization technique for non-commuting hyperbolic systems. The new method will be applied to the Euler and Navier-Stokes equations on unstructured grids, both in 2D and 3D. Only steady state problems will be considered and time stepping is used as an iterative tool to solve the set of nonlinear algebraic equations. Consequently accuracy in time is not important, an argument which will be used frequently.

Started by Roe [95], the present multi-dimensional upwind method was developed further in collaboration with Deconinck, Struijs, Paillère and Bonfiglioli [118, 119, 116, 84, 16]. Second-order monotonic scalar advection schemes on triangles (in 2D) and tetrahedra (in 3D) were developed. These schemes have a compact support (only the immediate neighbors are involved in the computational stencil) and are more accurate than standard finite-volume/finite-difference methods. A review of these schemes is given in chapter 2.

The extension of scalar schemes to non-commuting hyperbolic systems is straightforward for dimensionally split algorithms, where the multi-dimensional problem is split into a set of one-dimensional subproblems, usually in the direction normal to the faces of the control volume. As in one dimension there is only one Jacobian matrix, the characteristic form of the equations is a set of one-dimensional scalar equations and the scalar schemes can be applied directly in the case that the Jacobian is constant. For nonlinear conservation laws this is not the case and the generalization is given by a, usually approximate, solution to the one-dimensional Riemann problem in the direction of the normals. Attempts have been made to solve the true multi-dimensional Riemann problem [43, 3] and to use this solution as a building block for finite-volume upwind solvers. However, the solution to this problem, where three or more states interact with each other is extremely complex and expensive to compute and therefore probably not suited for practical computations.

As shown in section 1.4 the dimensional splitting introduces an unnecessary amount of numerical diffusion. In the multi-dimensional upwind approach this amount is reduced by not splitting the problem into one-dimensional subproblems. Consequently the extension to systems is more difficult, unless the Jacobians of the system commute. In this case the characteristic form is a set of two or three dimensional scalar advection equations and again what is needed is the scalar schemes. However it is well-known that the Jacobian matrices of the Euler equations do not commute and something else must be done.

One possibility, used by Roe, Deconinck and co-workers, is based on an approximate diagonalization approach, *i.e.* trying to find a characteristic form of the Euler equations which is as close as possible to a set of scalar advection equations. Then the scalar schemes are applied to this set of equations. The characteristic form can be written as:

$$\frac{\partial W}{\partial t} + \underbrace{D_i \frac{\partial W}{\partial x_i}}_{\text{Advection terms}} + \underbrace{C_i \frac{\partial W}{\partial x_i}}_{\text{Coupling terms}} = 0. \quad (1.61)$$

Here W is the set of characteristic variables, D_i are diagonal matrices and C_i off-diagonal matrices. The intention of the approximate diagonalization approach is to minimize the number of elements in the matrices C_i , also called coupling terms. The early attempts were based on the direction of flow-variable gradients [26], and stability problems inevitably occurred. Later, Paillère *et al.*[85] based their characteristic form on physical directions, streamline, Mach lines, etc., rather than

flow variable gradients at the price of more coupling terms. This cured some of the stability problems and very good results have been obtained for subsonic and transonic flows with relatively weak shocks and not too high aspect ratio cells [84]. However, as it is impossible to obtain a complete diagonalization (except for a 2D steady supersonic flow, where the characteristic set of equations is identical to the equations of the method of characteristics), coupling terms will always be present. These terms cannot be treated with monotonic schemes and consequently instabilities occur near strong discontinuities, which leads to failure of the method.

In this thesis a completely different approach is used. Following the work of Sidilkover [109, 108, 110], the scalar advection schemes are extended to system schemes for coupled hyperbolic systems. This completely solves the instability problems near strong discontinuities and a very robust discretization technique is obtained. The system schemes are discussed in general in chapter 3 and some specific issues for the Euler equations in chapter 5. The implicit time integration method and its parallelization (probably necessary to tackle large 3D problems) are given in chapter 4.

Chapter 6 deals with the (Reynolds-Averaged) Navier-Stokes equations. It will be shown that a Galerkin finite-element discretization of the viscous fluxes is consistent on linear triangles and tetrahedra, the only control volumes considered. Two types of turbulence models are used, the one-equation Spalart-Allmaras model and two-equation $k-\omega$ type models, such as Menter's SST and BSL models and the $k-\omega$ model itself. The last chapter contains the conclusions and some suggestions for future developments.

Chapter 2

Review of the scalar schemes

In this chapter numerical schemes for scalar advection are reviewed. The governing advection equation is given by

$$\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} = 0, \quad (2.1)$$

and attention is restricted to schemes on P1 elements, linear triangles in 2D and linear tetrahedra in 3D. This review includes the multi-dimensional upwind schemes developed by Roe, Deconinck, Sidilkover, Struijs, Paillère and Bonfiglioli [118, 119, 116, 84, 111, 16], the SUPG finite-element scheme [19, 57] and more classical schemes like the dimensionally-split first-order upwind scheme on the dual grid, the Galerkin finite-element scheme and the Lax-Wendroff scheme.

In the first three paragraphs the advection speed u_i is assumed to be piecewise constant. Section 2.4 introduces a local linearization of equation (2.1), which is a way to treat non-constant advection speeds.

It will be shown that all second-order schemes can be interpreted as Petrov-Galerkin finite-element schemes, which allows a straightforward extension to the discretization of the advection-diffusion equation, section 2.5. As the latest developments of these schemes have been known for some years, this chapter has been kept as brief as possible and should be seen as an introduction to the next chapter, the discretization of non-commuting hyperbolic systems. A much more detailed overview of the scalar schemes is given in chapter 2 of reference [84], on which parts of this chapter are based, see also [27], chapter 1.

For reasons of simplicity, the main part is restricted to two space dimensions. The extension to 3D is given in section 2.6 and it will become clear that the formulae describing the different schemes are completely analogous with their 2D counterparts. First some general concepts are introduced.

2.1 General concepts

Consider the scalar advection equation (2.1) on the domain Ω with (piecewise) constant advection speed u_i . This is an initial, boundary value problem, where for

well-posedness the initial conditions:

$$\phi(x_i, t = 0) = \phi_0(x_i), \quad \forall x_i \in \Omega \quad (2.2)$$

and the boundary conditions:

$$\phi(x_i, t) = \phi_{\Gamma^+}(x_i, t), \quad \forall x_i \in \Gamma^+, \quad \forall t > 0 \quad (2.3)$$

must be given [65]. Here Γ^+ is the inlet boundary of Ω , *i.e.* the part of Γ where the advection vector $\vec{u} = u_i \vec{1}_{x_i}$, which is the characteristic of equation (2.1), enters the domain. Conversely on $\Gamma^- = \Gamma \setminus \Gamma^+$, called the outlet boundary, where the characteristic leaves the domain, no boundary conditions have to be specified.

Assume that the 2D spatial domain Ω is triangulated with triangles of the type given in figure 2.1, where 1 – 2 – 3 are local node numbers. The inward scaled normals, *i.e.* normals with the length of the corresponding face, are defined for a counter clockwise numbering of the nodes by:

$$\vec{n}_1 = (y_2 - y_3) \vec{1}_x + (x_3 - x_2) \vec{1}_y, \quad (2.4)$$

$$\vec{n}_2 = (y_3 - y_1) \vec{1}_x + (x_1 - x_3) \vec{1}_y, \quad (2.5)$$

$$\vec{n}_3 = (y_1 - y_2) \vec{1}_x + (x_2 - x_1) \vec{1}_y, \quad (2.6)$$

where for convenience x and y are used to indicate the independent variables. In case the node numbering is clockwise the subscripts in the definitions (2.4) to (2.6) must be reversed. It is obvious that:

$$\vec{n}_1 + \vec{n}_2 + \vec{n}_3 = \vec{0}. \quad (2.7)$$

The Petrov-Galerkin discretization at node l , see section 1.1.3, of equation (2.1) becomes:

$$\iint_{\Omega} w_l \sum_{k=1}^{\#\text{nodes}} N_k \frac{\partial \phi_k}{\partial t} d\Omega + \iint_{\Omega} w_l u_i \sum_{k=1}^{\#\text{nodes}} \frac{\partial N_k}{\partial x_i} \phi_k d\Omega = 0. \quad (2.8)$$

Here w_l is the Petrov-Galerkin weight function at node l , which should obey certain conditions, see [56]. In practice this means that w_l and its first order derivatives must be square integrable. Consequently w_l is bounded. N_k is the basis function

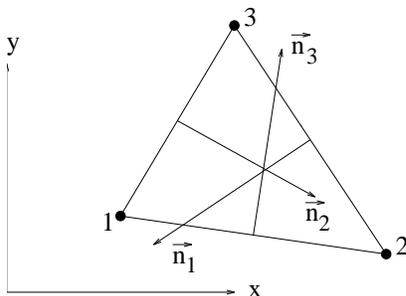


Figure 2.1: A generic triangle with inward scaled normals

of node k , see figure 1.4, and ϕ_k the value of ϕ in node k . Usually the global system (2.8) is built as a summation over the individual triangles T :

$$\sum_T \left\{ \iint_T w_l \sum_{k=1}^3 N_k \frac{\partial \phi_k}{\partial t} dT + \iint_T w_l u_i \sum_{k=1}^3 \frac{\partial N_k}{\partial x_i} \phi_k dT \right\} = 0, \quad (2.9)$$

where the sum inside every triangle runs from 1 to 3, the number of nodes of the triangle. The first integral leads to the product of the so called mass matrix M , whose element M_{lk} is defined as

$$M_{lk} = \sum_T \iint_T w_l N_k dT, \quad (2.10)$$

and the time derivatives of ϕ_k .

For time dependent problems the mass matrix must be taken into account to obtain an accuracy in time higher than first order. Note that the mass matrix (2.10) results in a relation between nodes l and k and consequently a linear system must be solved every time step, even if an explicit time integration method is used.

However in this work only steady problems are considered and therefore the w_l in the mass matrix can be chosen differently from the w_l in the spatial part of equation (2.9). It was found that the lumped Galerkin mass matrix leads to the most stable formulation. The Galerkin, $w_l = N_l$, mass matrix is given by:

$$M_{lk}^{Gal} = \sum_T \iint_T N_l N_k dT, \quad (2.11)$$

and its lumped version is obtained by summing the row elements and storing it in the diagonal entry:

$$M_{lk}^{Gal \text{ lumped}} = \sum_{T \in \Omega_l} \frac{S_T}{3} \delta_{lk} = S_l \delta_{lk}. \quad (2.12)$$

Here S_T is the area of triangle T , δ_{lk} the Kronecker delta function and S_l the area of the median dual cell of node l , see figure 2.2. Ω_l indicates all triangles which belong to neighborhood of node l , see figure 2.3.

The approximation (2.12) does not change the steady-state solution and, because it is a diagonal matrix, it avoids the solution of a linear system for explicit time integrators. Moreover the full mass matrix is usually not positive-definite and causes the positive schemes, see section 2.2.2 for the definition of positivity, to loose that property in the transient phase, which can lead to failure of the method. The approximation (2.12), a diagonal matrix with positive coefficients, which is positive-definite, does not affect the positivity.

For a linear triangle it can easily be verified that the following relation holds inside triangle T :

$$\frac{\partial N_k}{\partial x_i} = \frac{n_{k_i}}{2S_T}, \quad (2.13)$$

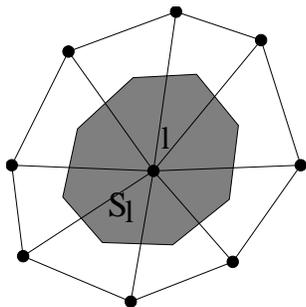


Figure 2.2: Node l , its immediate neighbors and its median dual cell S_l (shaded region)

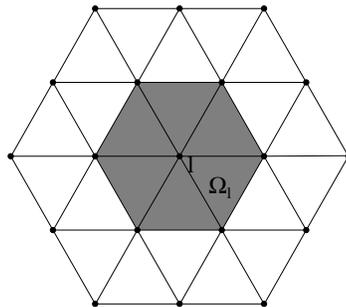


Figure 2.3: The neighborhood Ω_l , shaded region, of node l .

where n_{k_i} is the component of \vec{n}_k in the x_i direction, *i.e.* $n_{k_i} = \vec{n}_k \cdot \vec{1}_{x_i}$. Consequently the second integral of equation (2.9) for triangle T , from now on called R_l^T , simplifies to:

$$R_l^T = \left(\frac{1}{S_T} \iint_T w_l dT \right) R_T. \quad (2.14)$$

Here R_T is the cell residual:

$$R_T = \frac{1}{2} \sum_{p=1}^3 u_i n_{p_i} \phi_p = \sum_{p=1}^3 k_p \phi_p, \quad (2.15)$$

where the upwind parameters k_p are defined as:

$$k_p = \frac{1}{2} u_i n_{p_i}. \quad (2.16)$$

Note that in equation (2.15) use has been made of the assumption that u_i is (piecewise) constant.

The form (2.14) can be simplified even further by introducing of the distribution coefficient β_l^T for triangle T ,

$$\beta_l^T = \frac{1}{S_T} \iint_T w_l dT, \quad (2.17)$$

leading to

$$R_l^T = \beta_l^T R_T. \quad (2.18)$$

The β_l^T are called distribution coefficients, because they distribute parts of the cell residual R_T to the 3 nodes of triangle T . For consistency the local nodal residuals R_l^T should sum up to R_T , which is equivalent to:

$$\beta_1^T + \beta_2^T + \beta_3^T = 1. \quad (2.19)$$

Combining equations (2.9), (2.12) and (2.18) results in:

$$\frac{d\phi_l}{dt} + \frac{1}{S_l} \sum_T \beta_l^T R_T = 0. \quad (2.20)$$

The conclusion of this exercise is that any Petrov-Galerkin finite-element scheme for equation (2.1) with piecewise constant advection speed u_i on linear triangles can be written in the residual distribution form (2.18). The condition that the weight functions w_l are bounded, has as a consequence that the distribution coefficients β_l^T are bounded. This implies that every Petrov-Galerkin finite-element scheme is linearity preserving, see section 2.2.3.

2.2 Properties of the distribution schemes

The properties of the different schemes depend on the definition of the β_l^T (or equivalently w_l). The only restriction that they must obey is given in equation (2.19), and consequently degrees of freedom remain to adapt the scheme such that it has the desired properties. The following properties will be discussed in detail in the next sections:

- Multi-dimensional upwind (\mathcal{MU})
- Positivity (\mathcal{P})
- Linearity Preservation (\mathcal{LP})
- Continuity (\mathcal{C})

They correspond with different aspects, which are discussed below.

As mentioned in section 1.4, the multi-dimensional upwind scheme minimizes the amount of crosswind diffusion within the class of upwind schemes and consequently gives the most accurate results.

The positivity property guarantees that existing discontinuities are captured monotonically (if the initial solution does not contain overshoots).

Within the class of schemes considered, *i.e.* assuming a linear variation of the solution per element, at most linear varying solutions can be reproduced exactly by the numerical scheme. This ability is called Linearity Preservation.

Continuity of the schemes is required to obtain a smooth convergence to the steady-state solution.

The optimal numerical scheme, again within the class considered, for advection problems with discontinuities should have all four properties.

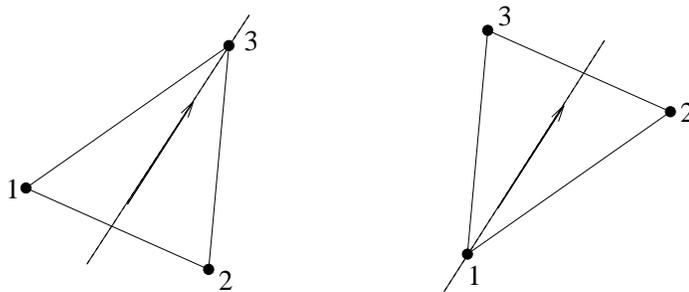


Figure 2.4: One-inflow triangle (left) and two-inflow triangle (right). In both cases the arrow indicates the direction of the streamline.

2.2.1 Multi-dimensional upwind (\mathcal{MU})

The motivation for multi-dimensional upwinding has become clear from the analysis in section 1.4. However this was carried out for the finite-difference method on a structured Cartesian grid, while the intended discretization technique is a finite-element method on an unstructured triangular grid. Consequently a condition must be found which guarantees multi-dimensional upwinding for this kind of schemes.

For scalar advection one can distinguish two types of triangles, one-inflow and two-inflow triangles, of which examples are shown in figure 2.4. For one-inflow triangles only one of the dot products of the advection vector with the inward scaled normals, the upwind parameters k_p (2.16), is positive; for two-inflow triangles two of the k_p 's are positive. As equation (2.7) holds, and thus:

$$k_1 + k_2 + k_3 = 0, \quad (2.21)$$

it is obvious that these are the only configurations possible. A scheme is now said to be multi-dimensional upwind if:

$$\mathcal{MU} \quad \beta_l^T = 0 \quad \text{if } k_l \leq 0, \quad (2.22)$$

i.e. nothing is distributed to inflow nodes. By using the expression (2.15) for R_T , one can show that this condition leads to an exact steady-state solution of equation (2.1) if the advection vector is aligned with one of the faces of the triangle. The multi-dimensional upwind finite-difference scheme on quadrilaterals, section 1.4, reproduces the exact steady solution if the advection vector is aligned with one of the faces or diagonals. This implies that the discretization technique on triangles is less accurate than on quadrilaterals [125, 88], the price paid for using unstructured grids of triangles. However, the first-order multi-dimensional upwind method on triangles is in general more accurate than the standard first-order dimensionally-split upwind method on quadrilaterals.

2.2.2 Positivity (\mathcal{P})

In a compressible flow, discontinuities like shocks and slip lines can be present. It is desirable that the numerical method captures these phenomena monotonically,

i.e. without under- and overshoots. One way to achieve this is to require the scheme to be positive. Consider the semi-discrete version of equation (2.1) at node l , in which only the spatial part has been discretized:

$$\frac{d\phi_l}{dt} + \sum_m c_{lm} (\phi_l - \phi_m) = 0. \quad (2.23)$$

Note that for the schemes considered only the immediate neighbors have non-zero coefficients c_{lm} . A scheme is said to be positive [89, 98] if:

$$\mathcal{P} \quad c_{lm} \geq 0 \quad \forall l, m, l \neq m, \quad (2.24)$$

which is identical to Jameson's LED (Local Extremum Diminishing) criterion [54, 52]. This ensures that no new extrema are created. It is clear from equation (2.23) and (2.24) that ϕ_l cannot increase/decrease for a local maximum/minimum. Clearly the positivity property does not guarantee existing overshoots to vanish. Therefore it is possible that there are several (numerical) steady-state solutions depending on the initial conditions, especially for nonlinear schemes and/or problems.

From the positivity concept it is possible to derive a time step restriction. For example, equation (2.23) integrated with the forward Euler time integrator results in:

$$\phi_l^{n+1} = \phi_l^n - \Delta t \sum_m (c_{lm} (\phi_l - \phi_m))^n, \quad (2.25)$$

where n and $n + 1$ indicate consecutive time levels. To guarantee that overshoots cannot arise at the new time level, it is sufficient to require that the new solution is a convex sum of the values at the previous level:

$$\phi_l^{n+1} = \sum_m \tilde{c}_{lm} \phi_l^n, \quad \tilde{c}_{lm} \geq 0, \quad \forall l, m. \quad (2.26)$$

For consistency one has $\sum_m \tilde{c}_{lm} = 1$. When combining equations (2.25) and (2.26), the condition $\tilde{c}_{ll} \geq 0$ leads to the restriction:

$$\Delta t_l \leq \frac{1}{\sum_{m \neq l} c_{lm}}. \quad (2.27)$$

For a general time integrator this becomes:

$$\Delta t_l \leq \frac{\text{CFL}}{\sum_{m \neq l} c_{lm}}, \quad (2.28)$$

where CFL is a typical constant for the time integration method used.

The positivity condition (2.24) is grid dependent and consequently difficult to impose. Therefore the local positivity property is introduced. This requires that condition (2.24) is obeyed for each individual element. Obviously this is more restrictive, but it has the advantage of being grid independent and thus easy to impose.

2.2.3 Linearity Preservation (\mathcal{LP})

Linearity preservation is the ability of the numerical scheme to reproduce steady linear solutions of equation (2.1) exactly. Within the class of schemes considered, linear variation of the solution over the elements, this is the highest possible order of polynomials, which can be calculated exactly. The condition imposed on the residual distribution schemes (2.18) can easily be determined.

Consider node l and its immediate neighbors, figure 2.2. The numerical steady-state solution is given by, see equation (2.20):

$$\frac{d\phi_l}{dt} = -\frac{1}{S_l} \sum_T \beta_l^T R_T = 0, \quad (2.29)$$

where S_l is the area of the median dual cell of node l , see figure 2.2, which enters the formulation due to the mass matrix approximation (2.12). One of the cells T of figure 2.2 is depicted in more detail in figure 2.5. Suppose that the exact solution is given by ϕ_l^e , ϕ_m^e and ϕ_n^e , where the subscript indicates the node number, see figure 2.5. The values of ϕ_m^e and ϕ_n^e can be computed from a Taylor series expansion around node l :

$$\begin{aligned} \phi_m^e &= \phi_l^e + \left(\frac{\partial\phi}{\partial x}\right)_l^e n_{n_y}^T - \left(\frac{\partial\phi}{\partial y}\right)_l^e n_{n_x}^T \\ &\quad + \frac{1}{2} \left(\frac{\partial^2\phi}{\partial x^2}\right)_l^e (n_{n_y}^T)^2 - \frac{1}{2} \left(\frac{\partial^2\phi}{\partial x\partial y}\right)_l^e n_{n_y}^T n_{n_x}^T + \frac{1}{2} \left(\frac{\partial^2\phi}{\partial y^2}\right)_l^e (n_{n_x}^T)^2 + O(h^3) \end{aligned} \quad (2.30)$$

$$\begin{aligned} \phi_n^e &= \phi_l^e - \left(\frac{\partial\phi}{\partial x}\right)_l^e n_{m_y}^T + \left(\frac{\partial\phi}{\partial y}\right)_l^e n_{m_x}^T \\ &\quad + \frac{1}{2} \left(\frac{\partial^2\phi}{\partial x^2}\right)_l^e (n_{m_y}^T)^2 - \frac{1}{2} \left(\frac{\partial^2\phi}{\partial x\partial y}\right)_l^e n_{m_y}^T n_{m_x}^T + \frac{1}{2} \left(\frac{\partial^2\phi}{\partial y^2}\right)_l^e (n_{m_x}^T)^2 + O(h^3) \end{aligned} \quad (2.31)$$

or in streamline coordinates (ξ, η) , see figure 1.12:

$$\phi_m^e = \phi_l^e - \left(\frac{\partial\phi}{\partial\eta}\right)_l^e n_{n_\xi}^T + \frac{1}{2} \left(\frac{\partial^2\phi}{\partial\eta^2}\right)_l^e (n_{n_\xi}^T)^2 + O(h^3), \quad (2.32)$$

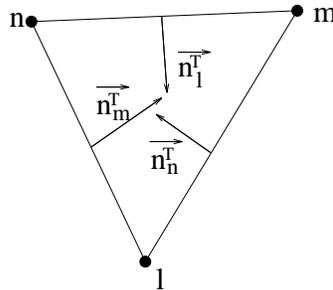


Figure 2.5: Cell T which shares node l and its inward scaled normals

$$\phi_n^e = \phi_l^e + \left(\frac{\partial\phi}{\partial\eta}\right)_l^e n_{m\xi}^T + \frac{1}{2} \left(\frac{\partial^2\phi}{\partial\eta^2}\right)_l^e (n_{m\xi}^T)^2 + O(h^3). \quad (2.33)$$

The components of the Cartesian inward scaled normals of the equations (2.30) and (2.31) are given in their local form in equations (2.4), (2.5) and (2.6), while in streamline coordinates the following expressions hold:

$$\begin{aligned} n_{m\xi}^T &= \eta_n^T - \eta_l^T \\ n_{n\xi}^T &= \eta_l^T - \eta_m^T. \end{aligned} \quad (2.34)$$

Furthermore in equations (2.32) and (2.33) use is made of the fact the exact steady state solution of equation (2.1) is only a function of η , the coordinate normal to the streamline. If the equations (2.32) and (2.33) are substituted into equation (2.15) for the cell residual R_T , after some manipulations the following expression is obtained:

$$R_T = \frac{\|\mathbf{u}\|}{4} \left(\frac{\partial^2\phi}{\partial\eta^2}\right)_l^e n_{l\xi}^T n_{m\xi}^T n_{n\xi}^T + O(h^4), \quad (2.35)$$

where $\|\mathbf{u}\| = \sqrt{u_i u_i}$. Substituting this into equation (2.29) gives:

$$\frac{\|\mathbf{u}\|}{4} \left(\frac{\partial^2\phi}{\partial\eta^2}\right)_l^e \frac{1}{S_l} \sum_T \beta_l^T n_{l\xi}^T n_{m\xi}^T n_{n\xi}^T + O(h^2) = 0. \quad (2.36)$$

If the scheme has to be exact for linear solutions, both $\left(\frac{\partial^2\phi}{\partial\eta^2}\right)_l^e$ and the terms appearing in $O(h^2)$ are zero, β_l^T must be bounded to make equation (2.36) into an identity. Consequently a sufficient condition for linearity preservation is:

$$\mathcal{LP} \quad \beta_l^T \text{ is bounded.} \quad (2.37)$$

Struijs [116] proved that linear schemes cannot have both the positivity and linearity preservation property, which is an equivalent formulation of the famous Godunov theorem [44]. A scheme is called linear if R_l^T , the part of the residual R_T of cell T which is sent to node l , is a linear function of the unknowns ϕ_m , *i.e.* the coefficients c_{lm} in equation (2.23) are independent of ϕ_m .

The relation of linearity preservation with second-order truncation error is grid dependent. On triangulated structured grids with a uniform choice of the diagonals, see figure 2.6, it can be proved that linear linearity-preserving schemes are

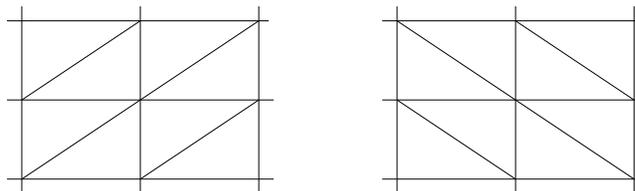


Figure 2.6: Uniformly triangulated structured grids

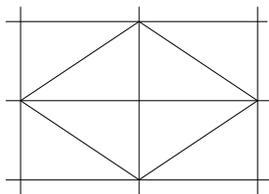


Figure 2.7: Isotropic triangulated structured grids

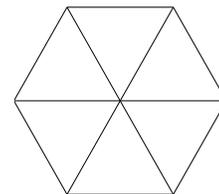
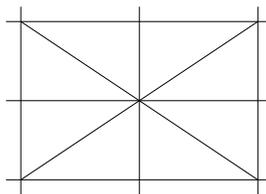


Figure 2.8: Regular unstructured grid

second-order accurate for constant advection speed u_i [84]. However a similar analysis for isotropic triangulated structured grids, figure 2.7, and a regular unstructured grid, figure 2.8, only shows first-order accuracy for linearity preserving schemes. However this local truncation analysis is too pessimistic, for Petrov-Galerkin finite-element methods, which are linearity preserving by definition, can be proved to be at least $O(h^{1.5})$ accurate, see [83, 58, 59]. In practice second-order accuracy is found for sufficiently smooth solutions on all types of grids, as long as the scheme is linear [21]. For nonlinear schemes a numerical convergence study shows results closer to the theoretical $O(h^{1.5})$, see [84].

2.2.4 Continuity (\mathcal{C})

Convergence towards steady-state solutions by means of iterative methods are hampered if the distributions to the nodes discontinuously change from one iteration to the other. Usually a cyclic behavior, known as a limit cycle, is then observed and therefore it is desirable that the schemes are continuous. Two kinds of continuity are distinguished:

- Continuity with respect to the solution ϕ_m
- Continuity with respect to the advection vector u_i .
(Only important for non-constant advection speeds.)

2.3 Numerical schemes

2.3.1 The N-scheme ($\mathcal{MU}, \mathcal{P}, \mathcal{C}$)

The N-scheme is the residual distribution formulation of the first-order multi-dimensional upwind method, see section 1.4. For the triangle shown in figure 2.1, the distributions to the nodes are defined by:

$$R_l^N = k_l^+ (\phi_l - \phi_{in}), \quad (2.38)$$

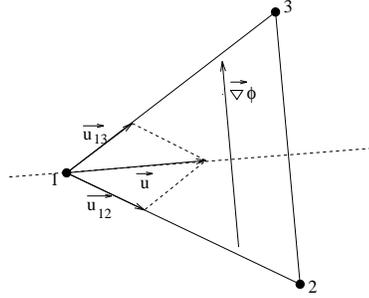


Figure 2.9: Two target case for the N-scheme with zero cell residual.

where

$$\phi_{in} = \frac{1}{\sum_{m=1}^3 k_m^-} \sum_{m=1}^3 k_m^- \phi_m \quad (2.39)$$

is the (linearly interpolated) state at the inflow point of the triangle, point 1 in figure 2.9 for a two inflow triangle, see also figure 2.4. The parameters k_l^+ and k_l^- are defined as $\max(0, k_l)$ and $\min(0, k_l)$ respectively, in which k_l is the upwind parameter, see equation (2.16). The multi-dimensional upwind property is easily checked, as only nodes with $k_l > 0$ receive a contribution due to the multiplication with k_l^+ .

If equation (2.39) is substituted in (2.38) the following alternative formulation of the N-scheme is obtained:

$$R_l^N = \sum_{m=1}^3 c_{lm} (\phi_l - \phi_m), \quad c_{lm} = \frac{k_l^+ k_m^-}{\sum_{p=1}^3 k_p^-}. \quad (2.40)$$

This form is identical to the one used in the positivity definition, equation (2.24). From the formulation (2.40) it is obvious the coefficients $c_{lm} \geq 0$ and thus the N-scheme is positive. Clearly, the scheme is linear as well.

The distribution coefficients β_l^N are not bounded. This can be seen in the case that two of the upwind parameters k_l are positive. Without loss of generality it is assumed that these are k_2 and k_3 . Furthermore assume the total cell residual $R_T = 0$, *i.e.* the advection vector is normal to $\vec{\nabla}\phi$. This situation is shown in figure 2.9. For this case the distribution of the N-scheme is equivalent to splitting the advection vector into two parts, as shown in figure 2.9, which results in $R_2^N = S_T \vec{u}_{12} \cdot \vec{\nabla}\phi$ and $R_3^N = S_T \vec{u}_{13} \cdot \vec{\nabla}\phi$. Both are clearly non-zero and consequently for this case the β_2^N and β_3^N are not bounded. This means that the N-scheme is only first order accurate, which is in agreement with Godunov's theorem. Due to this non-boundedness of β_l^N , and consequently of the weight function w_l , see

equation (2.17) , the N-scheme does not belong to the class of Petrov-Galerkin finite-element schemes.

The operations $k_l^+ = \max(0, k_l)$ and $k_l^- = \min(0, k_l)$ are continuous at $k_l = 0$ and therefore the continuity of the scheme is ensured.

2.3.2 The PSI or limited N-scheme ($\mathcal{MU}, \mathcal{LP}, \mathcal{P}, \mathcal{C}$)

The PSI (Positive Streamwise Invariant) scheme [119] was the first scheme, which had all the properties defined at the beginning of section 2.2. Earlier attempts, like the Petrov-Galerkin scheme of Mizukami-Hughes [79], the NN-scheme [117] and the Level scheme [101], all failed the continuity requirement and consequently convergence to steady-state solutions could be hampered. The original formulation of the PSI-scheme is given in [119]:

$$\beta_l^{\text{PSI}} = \frac{\max(0, \beta_l^{\text{N}})}{\sum_{m=1}^3 \max(0, \beta_m^{\text{N}})}, \quad (2.41)$$

where

$$\beta_l^{\text{N}} = \frac{R_l^{\text{N}}}{R_T}. \quad (2.42)$$

Later Sidilkover and Roe [111] showed that a linearity-preserving, positive scheme could be constructed from any first-order, positive scheme by applying a symmetric limiter function (with some additional constraints) to the distributions of the first-order scheme.

If the first-order scheme is the N-scheme, the limiter must be applied only if two inflow parameters k_l are positive, as in the one-target case the N-scheme contributions are both positive and linearity preserving. Again assume that k_2 and k_3 are the positive inflow parameters and the corresponding N-scheme distributions are given by R_2^{N} and R_3^{N} respectively. Then the limited N-scheme is defined as:

$$R_2^{\text{LN}} = R_2^{\text{N}} \left[1 - \Psi \left(\frac{-R_3^{\text{N}}}{R_2^{\text{N}}} \right) \right], \quad (2.43)$$

$$R_3^{\text{LN}} = R_3^{\text{N}} \left[1 - \Psi \left(\frac{-R_2^{\text{N}}}{R_3^{\text{N}}} \right) \right], \quad (2.44)$$

where $\Psi(r)$ is a limiter function with the following properties:

- consistency: $\Psi(1) = 1$
- symmetry: $\Psi \left(\frac{1}{r} \right) = \frac{\Psi(r)}{r}$

The consistency property ensures linearity preservation, for the argument r of the limiter Ψ equals 1 if $R_2^N = -R_3^N$ which is the case if the cell residual $R_T = 0$. If the limiters are applied, see equations (2.43) and (2.44), it is clear that R_2^{LN} and R_3^{LN} are zero only if $\Psi(1) = 1$. The symmetry property ensures conservation, *i.e.* $R_2^{LN} + R_3^{LN} = R_T$.

From equations (2.43) and (2.44) it can be seen that the N-scheme distribution is multiplied by a factor. To ensure that the limited scheme is locally positive, this factor must be positive. In combination with the consistency and symmetry condition, this gives an additional condition for the limiter function:

$$0 \leq \Psi(r) \leq 1. \quad (2.45)$$

Sidilkover and Roe [111] showed that this condition can be relaxed somewhat for global positivity, for example for linear advection with a constant advection speed on a uniformly triangulated structured grid, see figure 2.6, the global positivity condition becomes:

$$0 \leq \Psi(r), \frac{\Psi(r)}{r} \leq 2. \quad (2.46)$$

However in general this is not the case and in practice condition (2.45) is used.

If the minmod limiter is chosen,

$$\Psi(r) = \max[0, \min(r, 1)], \quad (2.47)$$

the expressions (2.43) and (2.44) can be simplified, if use is made of the typical property of the minmod limiter:

$$\Psi(r) + \Psi(1-r) = 1. \quad (2.48)$$

The distribution to node 2 then becomes:

$$\begin{aligned} R_2^{LN} &= \left[1 - \Psi\left(\frac{-R_3^N}{R_2^N}\right) \right] R_2^N \\ &= \left\{ 1 - \left[1 - \Psi\left(\frac{R_2^N + R_3^N}{R_2^N}\right) \right] \right\} R_2^N = \Psi\left(\frac{R_T}{R_2^N}\right) R_2^N \\ &= \Psi\left(\frac{R_2^N}{R_T}\right) R_T = \Psi(\beta_2^N) R_T \end{aligned} \quad (2.49)$$

A similar analysis for R_3^{LN} gives:

$$R_3^{LN} = \Psi\left(\frac{R_3^N}{R_T}\right) R_T = \Psi(\beta_3^N) R_T. \quad (2.50)$$

This form of the limited N-scheme shows that a linearity preserving, positive scheme can be obtained by simply limiting the distribution coefficients of the N-scheme. Moreover, if the expressions (2.49) and (2.50) are compared with the original formulation of the PSI-scheme, equation (2.41), one can see both schemes are identical. However, the formulation with a limiter function is more general, because other limiters than minmod can be chosen.

Since both the limiter and its arguments are continuous functions, the PSI-scheme, or limited N-scheme, is continuous.

2.3.3 The upwind finite-volume scheme $(\mathcal{P}, \mathcal{C})$

The upwind finite-volume scheme is the scheme used in finite-volume upwind solvers on unstructured grids, see e.g. [11, 13]. Usually it is combined with a nonlinear reconstruction phase to obtain second or higher-order accuracy and to maintain the positivity property [15, 12, 133, 33]. There are several variants of the scheme, for example cell-centered, vertex-centered on the median dual and vertex-centered on the centroid dual. Here only the residual distribution formulation of the vertex-centered scheme on the median dual cell is discussed. It is easy to verify that this scheme is identical to the finite-volume scheme on the median dual grid.

Consider the cell T and the scaled normals of the dual grid, see figure 2.10. These normals are given by:

$$\vec{n}_{12} = \frac{1}{6}(\vec{n}_2 - \vec{n}_1), \quad (2.51)$$

$$\vec{n}_{23} = \frac{1}{6}(\vec{n}_3 - \vec{n}_2), \quad (2.52)$$

$$\vec{n}_{31} = \frac{1}{6}(\vec{n}_1 - \vec{n}_3), \quad (2.53)$$

where the scaled normals \vec{n}_1 , \vec{n}_2 and \vec{n}_3 are defined in (2.4), (2.5) and (2.6) respectively. The corresponding upwind parameters k_{12} , k_{23} and k_{31} are given by:

$$k_{lm} = u_i n_{lm_i}. \quad (2.54)$$

The cell residual can be rewritten as:

$$R_T = k_{12}(\phi_2 - \phi_1) + k_{23}(\phi_3 - \phi_2) + k_{31}(\phi_1 - \phi_3). \quad (2.55)$$

The distribution to the three nodes is given by:

$$R_1^{\text{FV}} = k_{12}^-(\phi_2 - \phi_1) + k_{31}^+(\phi_1 - \phi_3), \quad (2.56)$$

$$R_2^{\text{FV}} = k_{23}^-(\phi_3 - \phi_2) + k_{12}^+(\phi_2 - \phi_1), \quad (2.57)$$

$$R_3^{\text{FV}} = k_{31}^-(\phi_1 - \phi_3) + k_{23}^+(\phi_3 - \phi_2). \quad (2.58)$$

As before the superscripts $+$ and $-$ mean $\max(0, \cdot)$ and $\min(0, \cdot)$ respectively. Positivity and continuity are clearly seen, as well as the fact the scheme is not linearity preserving and not multi-dimensional upwind.

A linearity preserving version of the scheme can be obtained by applying the limiter function described in section 2.3.2 to the first-order scheme.

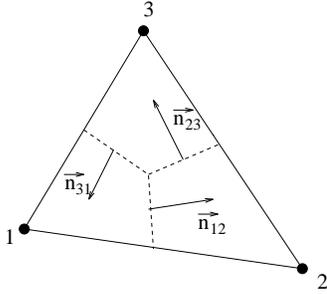


Figure 2.10: Cell T and scaled normals of the dual grid.

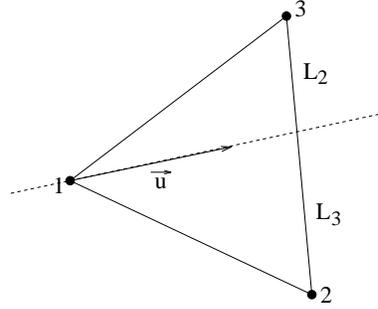


Figure 2.11: LDA scheme in case of two positive inflow parameters.

2.3.4 The LDA-scheme $(\mathcal{M}\mathcal{U}, \mathcal{L}\mathcal{P}, \mathcal{C})$

The distribution coefficients for the LDA (Low Diffusion A) scheme [119] are given by:

$$\beta_l^{\text{LDA}} = \frac{k_l^+}{\sum_{m=1}^3 k_m^+}, \quad (2.59)$$

where k_l^+ has the same meaning as in section 2.3.1. The definition (2.59) shows that β_l^{LDA} is bounded between 0 and 1 and is independent of the solution ϕ_m . Consequently the LDA-scheme is a linear multi-dimensional upwind scheme, which satisfies the linearity preservation property. As a result of Godunov's theorem the scheme therefore cannot be positive.

Note that if only one inflow parameter k_l is positive, the LDA, the PSI and the N-scheme are identical and the contributions of such triangles satisfy both the linearity preservation and the positivity property. However, if two inflow parameters are positive, the schemes differ. The geometrical interpretation of the LDA-scheme for this case is depicted in figure 2.11, where it is assumed that k_1 is negative and k_2 and k_3 are positive. β_2^{LDA} and β_3^{LDA} are defined by:

$$\beta_2^{\text{LDA}} = \frac{L_2}{L_2 + L_3}, \quad \beta_3^{\text{LDA}} = \frac{L_3}{L_2 + L_3}, \quad (2.60)$$

in which L_2 and L_3 are the lengths of the line segments shown in figure 2.11. For the continuity property the same argument holds as for the N-scheme.

2.3.5 The Galerkin-scheme $(\mathcal{L}\mathcal{P}, \mathcal{C})$

The Galerkin finite-element scheme is obtained by choosing the weight function w_l in equation (2.9) identical to the nodal basis function N_l . For linear advection on a linear triangle it has been shown in section 2.1 that any finite-element

scheme can be written in a residual distribution form. For the Galerkin method the distribution coefficients are:

$$\beta_l^{\text{GAL}} = \frac{1}{3}. \quad (2.61)$$

In section 1.2 it has been shown that a central discretization of the advection terms, which is identical to the Galerkin method, is unstable for pure advection, and artificial dissipation terms must be added, as is done in the Lax-Wendroff and the SUPG schemes discussed below, to stabilize it.

2.3.6 The Lax-Wendroff-scheme ($\mathcal{LP}, \mathcal{C}$)

Roe [100] showed that the Lax-Wendroff scheme could be interpreted as a residual distribution scheme with distribution coefficients:

$$\beta_l^{\text{LW}} = \frac{1}{3} + \frac{\Delta t}{2S_T} k_l, \quad (2.62)$$

where S_T is the area of the triangle, k_l the upwind parameter and Δt a cell time step, which introduces some arbitrariness into the scheme. The cell time step should not be confused with the nodal time steps (2.28). The former defines the distribution to the 3 nodes, while the latter is used to progress in time. In this work the cell time step is taken according to Rudgyard's definition [103]:

$$\Delta t = \nu_c \frac{2S_T}{3 \sum_{m=1}^3 |k_m|}, \quad (2.63)$$

which leads to:

$$\beta_l^{\text{LW}} = \frac{1}{3} + \nu_c \frac{k_l}{3 \sum_{m=1}^3 |k_m|}. \quad (2.64)$$

In equations (2.63) and (2.64) ν_c is the cell CFL number, usually chosen between $\frac{1}{2}$ and 1. The choice $\nu_c = \frac{2}{3}$ makes the scheme (2.64) identical to the Upwind Control Volume scheme of Giles *et al.*[42].

2.3.7 The SUPG-scheme ($\mathcal{LP}, \mathcal{C}$)

Two different types of methods can be identified to stabilize the Galerkin method for advection problems. The first type is the Discontinuous Galerkin (DG) method in which a Riemann flux function along element edges is used for stabilization. Consider the scalar conservation law

$$\frac{\partial \phi}{\partial t} + \frac{\partial f_i}{\partial x_i} = 0, \quad (2.65)$$

on a computational domain subdivided into arbitrary elements Ω_j . In the spatial DG method the solution inside each element is approximated by the expansion:

$$\phi_{\Omega_j} = \sum_{k=0}^N w_k \phi_{\Omega_j,k}(t). \quad (2.66)$$

A usual, but not necessary, choice for the degrees of freedom $\phi_{\Omega_j,k}$ is such that equation (2.66) is a Taylor series expansion around the centroid of Ω_j . The idea of using the gradient of the solution as an additional unknown has also been used in finite difference schemes, see [130]. The classical Galerkin formulation of equation (2.65) becomes:

$$\sum_j \left\{ \iint_{\Omega_j} w_l \left(\sum_k w_k \frac{\partial \phi_{\Omega_j,k}}{\partial t} + \frac{\partial f_i}{\partial x_i} \right) d\Omega \right\} = 0. \quad (2.67)$$

In contrast to the standard method, DG uses integration by parts on the divergence term. In combination with the fact that the basis functions w_k are only nonzero in element Ω_j , equation (2.67) becomes:

$$\iint_{\Omega_j} w_l \sum_k w_k \frac{\partial \phi_{\Omega_j,k}}{\partial t} d\Omega - \iint_{\Omega_j} \frac{\partial w_l}{\partial x_i} f_i d\Omega + \oint_{\partial\Omega_j} w_l f_i^R ds_i = 0. \quad (2.68)$$

Here ds_i are the components of the outward pointing surface normal and f_i^R is an (approximate) Riemann flux function, which enters the formulation, because the expansions (2.66) are in general discontinuous over the element boundaries. This Riemann flux function ensures stability due to the upwind biasing. Note that for $N = 0$ in the expansion (2.66), the spatial DG method is identical to the first-order, cell-centered finite-volume method. The DG method is not used in this work and the interested reader is referred to [24, 5] and references therein or to [69, 70] for a space-time DG method.

A different approach for stabilizing the Galerkin method is to introduce upwind biasing in the Petrov-Galerkin weight function w_l , see equation (2.8). Here only the Streamline Upwind Petrov Galerkin (SUPG) method is discussed. An overview of other stabilized Galerkin methods can be found in [80, 81]. In the SUPG method [19, 57] the weight function w_l is defined as:

$$w_l = N_l + \tau u_i \frac{\partial N_l}{\partial x_i}, \quad (2.69)$$

where τ is the SUPG intrinsic time scale, which for pure advection is given by:

$$\tau = \frac{h}{2 \|\mathbf{u}\|}. \quad (2.70)$$

For advection-diffusion problems this expression is multiplied by a function [80], such that it returns to the Galerkin discretization in the case that the cell Peclet

number is less than one. This is equivalent to the hybrid approach, described in section 1.3.2.

For linear triangles, use can be made of the relation (2.13), which results in the following distribution coefficients:

$$\beta_l^{\text{SUPG}} = \frac{1}{3} + \frac{hk_l}{2\|\mathbf{u}\|S_T}. \quad (2.71)$$

Both in this equation and in equation (2.70) h is a typical length scale of the cell. The usual definition is the maximum side length, although different definitions can be used in practice [80]. Comparing (2.71) with the distribution coefficient of the Lax-Wendroff scheme, equation (2.62), shows that both schemes are identical if the following time step definition for the Lax-Wendroff scheme is used:

$$\Delta t = \frac{h}{\|\mathbf{u}\|} \quad (2.72)$$

From equation (2.71) it is clear that the SUPG-scheme is linear and the β_l^{SUPG} is bounded. Consequently the scheme is not positive and an artificial dissipation term can be added to overcome this problem [60]. When certain simplifications are made, the scheme can still be cast into the residual distribution form:

$$\beta_l^{\text{SUPG+AV}} = \frac{1}{3} + \frac{hk_l}{2\|\mathbf{u}\|S_T} + \frac{\hat{\kappa}}{R_T} \frac{\frac{1}{2}\vec{\nabla}\phi \cdot \vec{n}_l}{S_T}, \quad (2.73)$$

where the artificial viscosity $\hat{\kappa}$ is defined as:

$$\frac{\hat{\kappa}}{R_T} = C_2 h \frac{\text{sgn}(R_T)}{\|\nabla\phi\| + h}. \quad (2.74)$$

For reasons of clarity the vector notation has been used in equations (2.73) and (2.74) instead of the tensor notation with the summation convention. The sgn function in (2.74) is given by:

$$\text{sgn}(r) = \begin{cases} -1 & \text{if } r < 0 \\ 0 & \text{if } r = 0 \\ 1 & \text{if } r > 0 \end{cases} \quad (2.75)$$

The constant C_2 in the artificial viscosity $\hat{\kappa}$ is usually set to 0.5. A numerical accuracy study of a smooth advection problem [84] shows that the addition of the artificial dissipation term degrades the order of accuracy from 2.2 to 1.6, the consequence for demanding monotonicity. The same study showed slightly less accuracy and stability for the SUPG-scheme with artificial dissipation compared to the PSI-scheme.

2.4 Treatment of non-constant advection speeds

From the definition of the schemes in the previous section it is clear they use a constant advection speed per cell and especially the positivity property heavily relies on this fact. However, most problems have non-constant advection speeds and a way must be found to overcome this contradiction. This section introduces the conservative linearization technique, which locally linearizes the advection speed in such a way that the schemes are still conservative.

Consider the scalar conservation law:

$$\frac{\partial \phi}{\partial t} + \frac{\partial f_i}{\partial x_i} = 0, \quad (2.76)$$

where $f_i = f_i(\phi)$ are the flux functions. The cell residual R_T is given by:

$$R_T = \iint_T \frac{\partial f_i}{\partial x_i} dT = \oint_{\Gamma} f_i n_i^{\text{ext}} d\Gamma, \quad (2.77)$$

where Γ is the edge of T and n_i^{ext} the component in the x_i -direction of its outward normal. The expression for R_T of equation (2.1) in case of a constant advection speed is:

$$R_T = u_i \iint_T \frac{\partial \phi}{\partial x_i} dT = u_i \oint_{\Gamma} \phi n_i^{\text{ext}} d\Gamma. \quad (2.78)$$

As ϕ is assumed to vary linearly over the triangle, only P1 elements being considered in this work, the integral in equation (2.78) is fixed and equals the expression given in (2.15). The idea now is to find a cell-average advection speed \bar{u}_i such that

$$\bar{u}_i \oint_{\Gamma} \phi n_i^{\text{ext}} d\Gamma \equiv \oint_{\Gamma} f_i n_i^{\text{ext}} d\Gamma, \quad (2.79)$$

i.e. equations (2.77) and (2.78) are required to be equal under the assumption of a constant advection speed \bar{u}_i per cell. Note that the conservation constraint (2.79) does not completely determine \bar{u}_i in multiple space dimensions and usually equation (2.79) is applied dimension by dimension. The degree of freedom left is the quadrature rule for the flux integral in equation (2.79). If f_i is a nonlinear function of ϕ usually a different quadrature rule than the trapezium rule must be used to avoid instabilities. In general it is quite cumbersome to find a cell-average state \bar{u}_i which obeys (2.79). However, if f_i is a quadratic function of ϕ a very simple analytical expression can be obtained. This is a severe restriction, but as will be shown in chapter 5, the perfect gas Euler system belongs to the class of equations having quadratic flux-functions.

In this case the cell residual R_T , equation (2.77), can be rewritten as:

$$\begin{aligned} \iint_T \frac{\partial f_i}{\partial x_i} dT &= \iint_T u_i \frac{\partial \phi}{\partial x_i} dT = \\ \left(\iint_T u_i dT \right) \frac{\partial \phi}{\partial x_i} &= S_T \left(\frac{u_{i,1} + u_{i,2} + u_{i,3}}{3} \right) \frac{\partial \phi}{\partial x_i}. \end{aligned} \quad (2.80)$$

Here use is made of (1): ϕ varies linearly with x_i , (2): $u_i = \frac{\partial f_i}{\partial \phi}$ is a linear function of ϕ and (3): the integral of a linear function over a triangle is the arithmetic average of that function times the area.

As an example, consider the two-dimensional Burgers equation in conservation form:

$$\frac{\partial \phi}{\partial t} + \frac{\partial \frac{1}{2} \phi^2}{\partial x} + \frac{\partial \phi}{\partial y} = 0 \quad (2.81)$$

and its quasi-linear version:

$$\frac{\partial \phi}{\partial t} + \phi \frac{\partial \phi}{\partial x} + \frac{\partial \phi}{\partial y} = 0. \quad (2.82)$$

It is obvious that only the x -component needs to be considered. Assuming a linear variation of ϕ over the triangle depicted in figure 2.1 results in:

$$\bar{u}_x \oint_{\Gamma} \phi n_x^{\text{ext}} d\Gamma = \frac{1}{2} \bar{u}_x (\phi_1 n_{1x} + \phi_2 n_{2x} + \phi_3 n_{3x}) \quad (2.83)$$

and:

$$\begin{aligned} \oint_{\Gamma} \frac{1}{2} \phi^2 n_x^{\text{ext}} d\Gamma &= -\frac{n_{3x}}{12} \left[\phi_1^2 + 4 \left(\frac{\phi_1 + \phi_2}{2} \right)^2 + \phi_2^2 \right] - \frac{n_{1x}}{12} \left[\phi_2^2 + 4 \left(\frac{\phi_2 + \phi_3}{2} \right)^2 + \phi_3^2 \right] \\ &\quad - \frac{n_{2x}}{12} \left[\phi_3^2 + 4 \left(\frac{\phi_3 + \phi_1}{2} \right)^2 + \phi_1^2 \right] \\ &= \frac{n_{1x}}{6} (\phi_1^2 - \phi_2 \phi_3) + \frac{n_{2x}}{6} (\phi_2^2 - \phi_3 \phi_1) + \frac{n_{3x}}{6} (\phi_3^2 - \phi_1 \phi_2) \\ &= \frac{1}{6} (\phi_1 + \phi_2 + \phi_3) (\phi_1 n_{1x} + \phi_2 n_{2x} + \phi_3 n_{3x}), \end{aligned} \quad (2.84)$$

where relation (2.7) and the Simpson quadrature rule, which is exact for this case, have been used. Comparing equations (2.83) and (2.84) it is clear that \bar{u}_x is given by:

$$\bar{u}_x = \frac{1}{3} (\phi_1 + \phi_2 + \phi_3), \quad (2.85)$$

i.e. the arithmetic average of the 3 nodal values.

2.5 The advection-diffusion equation

The Petrov-Galerkin interpretation of the residual distribution schemes allows a straightforward extension of the discretization technique to the advection-diffusion equation:

$$\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\nu \frac{\partial \phi}{\partial x_i} \right) = 0. \quad (2.86)$$

Here u_i is the advection vector and ν the positive diffusion coefficient. Multiplying equation (2.86) by the weight function w_l and integrating over the computational domain Ω results in:

$$\iint_{\Omega} w_l \frac{\partial \phi}{\partial t} d\Omega + \iint_{\Omega} w_l u_i \frac{\partial \phi}{\partial x_i} d\Omega - \iint_{\Omega} w_l \frac{\partial}{\partial x_i} \left(\nu \frac{\partial \phi}{\partial x_i} \right) d\Omega = 0. \quad (2.87)$$

Defining the neighborhood Ω_l as the part of the domain where w_l is not zero, see figure 2.3, making use of the relation:

$$w_l \frac{\partial}{\partial x_i} \left(\nu \frac{\partial \phi}{\partial x_i} \right) = \frac{\partial}{\partial x_i} \left(w_l \nu \frac{\partial \phi}{\partial x_i} \right) - \frac{\partial w_l}{\partial x_i} \nu \frac{\partial \phi}{\partial x_i} \quad (2.88)$$

and using the Gauss theorem, equation (2.87) becomes:

$$\begin{aligned} & \iint_{\Omega_l} w_l \frac{\partial \phi}{\partial t} d\Omega + \sum_{T \in \Omega_l} \iint_T w_l u_i \frac{\partial \phi}{\partial x_i} d\Omega + \\ & \sum_{T \in \Omega_l} \iint_T \frac{\partial w_l}{\partial x_i} \nu \frac{\partial \phi}{\partial x_i} d\Omega - \oint_{\partial \Omega} w_l \nu \frac{\partial \phi}{\partial n} = 0. \end{aligned} \quad (2.89)$$

The first two integrals in equation (2.89) are known, for their discretization has already been discussed in previous sections and will not be repeated here. In this work either Dirichlet or homogeneous Neumann boundary conditions are assumed and consequently the contour integral, which enters the formulation through the integration by parts, vanishes.

The residual distribution analogy of the Petrov-Galerkin discretization, section 2.1, provides a relation between the weight function w_l^T and the distribution coefficient β_l^T , see equation (2.17). Here the superscript T has been used to indicate that this relation is only valid for triangle T . The integral equation (2.17) does not determine w_l^T uniquely and an assumption must be made about its shape. Following the SUPG scheme, where for P1 elements the weight function is equal to the Galerkin weight function plus a constant, see section 2.3.7, w_l becomes:

$$w_l(x, y) = \sum_{T \in \Omega_l} w_l^T(x, y) = N_l(x, y) + \sum_{T \in \Omega_l} \left(\beta_l^T - \frac{1}{3} \right) \alpha^T(x, y). \quad (2.90)$$

Here $N_l(x, y)$ is the nodal basis function (= Galerkin weight function) and $\alpha^T(x, y) = 1$ inside triangle T and 0 outside.

As can be seen in equation (2.89), only the derivatives of w_l appear in the discretization of the diffusion term. As a result of the form chosen in equation (2.90) for w_l , this reduces to the gradient of the Galerkin weight function. This leads, together with the discrete advection term, to the complete semi-discrete form of equation (2.87) at point l :

$$S_l \frac{d\phi_l}{dt} + \sum_{T \in \Omega_l} (R_{l,inv}^T + R_{l,vis}^T) = 0, \quad (2.91)$$

in which

$$R_{l,inv}^T = \beta_l^T R_T \quad (2.92)$$

and

$$R_{l,vis}^T = \frac{1}{2} \vec{n}_l \cdot \nu \sum_{m=1}^3 \phi_m \frac{\vec{n}_m}{2S_T}. \quad (2.93)$$

In (2.93) use has been made of equation (2.13).

To determine the condition which ensures the local positivity for pure diffusion, $R_{l,vis}^T$ is rewritten as:

$$R_{l,vis}^T = -\frac{\nu}{4S_T} \vec{n}_l \cdot [(\phi_l - \phi_m) \vec{n}_m + (\phi_l - \phi_n) \vec{n}_n], \quad (2.94)$$

wherein l, m, n are a cyclic permutation of 1, 2, 3. Comparison with the positivity definition (2.24) shows the Galerkin discretization for pure diffusion is locally positive if the scalar products between the inward scaled normals are negative or zero, a condition which is met by triangles whose maximum angle does not exceed 90° . This is a severe restriction for the grid generator, especially in three space dimensions. In practice this condition is relaxed quite a lot for two reasons: (1) global positivity is less restrictive than its local form, see [11] where it has been demonstrated that the Galerkin discretization for diffusion is globally positive for a Delaunay triangulation, and (2) the viscous applications in this work are advection-dominated flows and so the overall coefficient c_{lm} in equation (2.24) is mainly determined by the advection part, which, for a positive scheme, is positive for any kind of triangle.

The diffusion term also has an influence on the time-step restriction, which becomes:

$$\frac{\text{CFL} S_l}{\Delta t_l} = \sum_{T \in \Omega_l} \frac{1}{\text{restr}_{\text{inv}}^l + \text{restr}_{\text{vis}}^l}, \quad (2.95)$$

where $\text{restr}_{\text{inv}}^l$ is the inviscid time step restriction per triangle, obtained by applying equation (2.27) for each cell, and $\text{restr}_{\text{vis}}^l$ is the diffusive time-step restriction, which is given by:

$$\text{restr}_{\text{vis}}^l = \frac{\nu \|\vec{n}_l\|^2}{4S_T}. \quad (2.96)$$

This is the result of applying the alternative positivity definition (2.26) in combination with the forward Euler time integrator to $R_{l,vis}$, equation (2.93).

To maintain the second-order accuracy for low cell Peclet numbers, a blending between an upwind (biased) and the central discretization of the advection term should be applied, as described in section 1.3.2 or reference[80]. However for the high Reynolds number applications of this work, chapter 6, the solutions did not show any improvement due to this blending and therefore all results are computed without it.

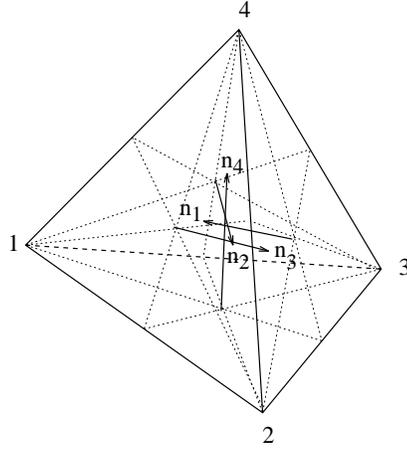


Figure 2.12: A generic tetrahedron with inward scaled normals.

2.6 Advection schemes in 3D

This section extends the schemes presented in section 2.3 to three space dimensions on linear tetrahedra, see figure 2.12. The inward scaled normals for this node numbering are given by:

$$\vec{n}_1 = \frac{1}{2} (\vec{x}_4 - \vec{x}_2) \times (\vec{x}_3 - \vec{x}_2), \quad (2.97)$$

$$\vec{n}_2 = \frac{1}{2} (\vec{x}_3 - \vec{x}_1) \times (\vec{x}_4 - \vec{x}_1), \quad (2.98)$$

$$\vec{n}_3 = \frac{1}{2} (\vec{x}_4 - \vec{x}_1) \times (\vec{x}_2 - \vec{x}_1), \quad (2.99)$$

$$\vec{n}_4 = \frac{1}{2} (\vec{x}_2 - \vec{x}_1) \times (\vec{x}_3 - \vec{x}_1). \quad (2.100)$$

Here \vec{x}_l is the position vector of node l .

As stated earlier, the distribution formulae to the nodes are identical in form to their 2D counterparts. However, the geometrical interpretation, especially for the N- and LDA-schemes, become more complicated and configurations occur which do not have a 2D counterpart. Earlier publications about these 3D advection schemes are [17, 30, 31, 16, 87].

The upwind parameters in three space dimensions take the form:

$$k_p = \frac{1}{3} u_i n_{p_i}. \quad (2.101)$$

Compared to the 2D case, equation (2.16), the factor $\frac{1}{2}$ is replaced by $\frac{1}{3}$, a consequence of the integration over the faces of the tetrahedron. The expression for the 3D cell residual, for constant advection speed u_i , and assuming a linear variation of ϕ , becomes:

$$R_T = u_i \int \int_T \frac{\partial \phi}{\partial x_i} dT = \sum_{m=1}^4 k_m \phi_k. \quad (2.102)$$

2.6.1 The N-scheme ($\mathcal{M}\mathcal{U}, \mathcal{P}, \mathcal{C}$)

The general formula for the N-scheme distribution to the nodes is

$$R_i^N = k_l^+ (\phi_l - \phi_{in}), \quad (2.103)$$

where

$$\phi_{in} = \frac{1}{4} \sum_{m=1}^4 k_m^- \phi_m. \quad (2.104)$$

Depending on the situation, one, two or three nodes receive a contribution. The geometrical interpretation for the three and two target cases are shown in figure 2.13. Starting from the inflow point, the advection vector is split into the necessary number of components and the residual for the target node is obtained by replacing the true advection vector by one of these components in equation (2.102). In the case of two target nodes, and consequently also two inflow nodes, the inflow point is located on the line segment connecting the inflow nodes, see figure 2.13(b). The coordinates of the inflow point, \vec{x}_{in} , are determined by:

$$\vec{x}_{in} = \frac{\sum_{m=1}^4 k_m^- \vec{x}_m}{\sum_{m=1}^4 k_m^-}, \quad (2.105)$$

where \vec{x}_m is the position vector of node m .

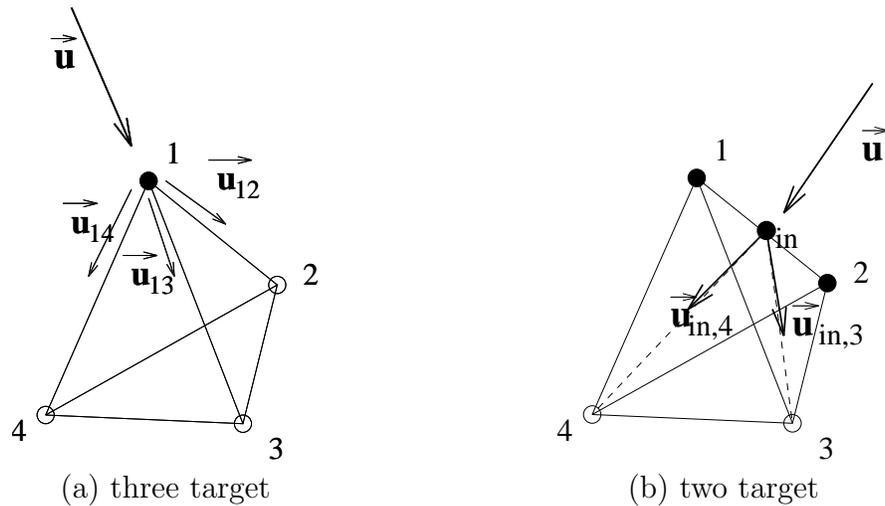


Figure 2.13: N-scheme configuration for (a) three target and (b) two target nodes.

2.6.2 The PSI or limited N-scheme ($\mathcal{M}\mathcal{U}, \mathcal{L}\mathcal{P}, \mathcal{P}, \mathcal{C}$)

For the nonlinear schemes the limiting approach is followed, see section 2.3.2, in which a positive, first-order scheme is converted into a nonlinear, positive, linearity-preserving scheme. As the N-scheme is the most accurate of these first-order schemes, attention will be restricted to nonlinear schemes based on the N-scheme, although the following limiting technique can be applied to any positive, first-order scheme.

The situation in three dimensions is more complicated than in two, because the N-scheme can have three targets, see previous section. The analysis in section 2.3.2 is only valid for two targets and consequently the theory must be extended. This was done by Sidilkover [110] for three targets. Below the extension to an arbitrary number of targets is given.

Suppose the N-scheme has n targets and the corresponding sub-residuals are given by R_l^N , $1 \leq l \leq n$. Define the quantity Q_l as:

$$Q_l = \frac{\sum_{\substack{k=1 \\ k \neq l}}^n [R_k]_l^-}{R_l^N + \sum_{\substack{k=1 \\ k \neq l}}^n [R_k]_l^+} \quad (2.106)$$

where

$$[R_k]_l^+ = \begin{cases} R_k^N, & \text{if } \text{sgn}(R_k^N) = \text{sgn}(R_l^N) \\ 0, & \text{otherwise} \end{cases} \quad (2.107)$$

and

$$[R_k]_l^- = \begin{cases} 0, & \text{if } \text{sgn}(R_k^N) = \text{sgn}(R_l^N) \\ -R_k^N, & \text{otherwise} \end{cases} \quad (2.108)$$

or

$$[R_k]_l^- = [R_k]_l^+ - R_k^N. \quad (2.109)$$

The sgn function is defined in equation (2.75). Then the limited subresiduals are given by:

$$R_l^{\text{LN}} = R_l^N + \sum_{\substack{k=1 \\ k \neq l}}^n \Psi(Q_k) R_k^N, \quad (2.110)$$

where $\Psi(r)$ is a limiter, for which identical conditions apply as in 2D. See [110] for the details about the proofs of positivity and linearity preservation for a general limiter.

If the minmod limiter (2.47) is used, equation (2.110) can be simplified. Without loss of generality suppose that the residual of target nodes 1 to p have the same sign and opposite to the sign of the residuals of the targets $p+1$ to n . Furthermore, assume that the following relation holds:

$$\left| \sum_{k=1}^p R_k^N \right| > \left| \sum_{k=p+1}^n R_k^N \right|. \quad (2.111)$$

It is easily verified that in this case the quantities Q_l , equation (2.106), simplify to:

$$Q_l = -\frac{\sum_{k=p+1}^n R_k^N}{\sum_{k=1}^p R_k^N} = \alpha, \quad l = 1..p \quad (2.112)$$

$$Q_l = -\frac{\sum_{k=1}^p R_k^N}{\sum_{k=p+1}^n R_k^N} = \frac{1}{\alpha}, \quad l = p+1..n \quad (2.113)$$

Due to the assumption (2.111), α is bounded between 0 and 1 and consequently:

$$\Psi^{minmod}(Q_l) = \alpha, \quad l = 1..p \quad (2.114)$$

$$\Psi^{minmod}(Q_l) = 1, \quad l = p+1..n. \quad (2.115)$$

The limited subresiduals, equation (2.110) then become:

$$R_l^{LN} = R_l^N + \sum_{\substack{k=1 \\ k \neq l}}^p \alpha R_k^N + \sum_{k=p+1}^n R_k^N = R_l^N (1 - \alpha) = \frac{\beta_l^N}{\sum_{k=1}^p \beta_k^N} R_T, \quad l = 1..p \quad (2.116)$$

$$R_l^{LN} = R_l^N + \sum_{k=1}^p \alpha R_k^N + \sum_{\substack{k=p+1 \\ k \neq l}}^n R_k^N = 0, \quad l = p+1..n \quad (2.117)$$

The cell-residual R_T is given by:

$$R_T = \sum_{k=1}^n R_k^N. \quad (2.118)$$

Equations (2.116) and (2.117) can be combined, resulting in:

$$R_l^{\text{PSI}} = \frac{\max(0, \beta_l^{\text{N}})}{\sum_{k=1}^n \max(0, \beta_k^{\text{N}})} R_T, \quad (2.119)$$

identical to the original PSI-scheme formulation (2.41). Clearly the distribution coefficients are bounded between 0 and 1 and the scheme is linearity preserving. Also the positivity proof is straightforward. Equation (2.119) is rewritten as:

$$R_l^{\text{PSI}} = \beta_l^{\text{PSI}} R_T = \underbrace{\frac{\beta_l^{\text{PSI}}}{\beta_l^{\text{N}}}}_{0 \leq \dots \leq 1} R_l^{\text{N}}. \quad (2.120)$$

Thus the distribution of the PSI-scheme is obtained by multiplying the positive N-scheme distribution by a nonlinear factor, which lies between 0 and 1.

2.6.3 The upwind finite-volume scheme $(\mathcal{P}, \mathcal{C})$

As in 2D, the normals of the dual grid can be constructed from the inward scaled normals, given in (2.97) to (2.100):

$$\vec{n}_{lm} = \frac{1}{12} (\vec{n}_m - \vec{n}_l). \quad (2.121)$$

The cell residual R_T in terms of these dual grid normals reads:

$$R_T = \sum_{l=1}^4 \sum_{m=l+1}^4 k_{lm} (\phi_m - \phi_l), \quad (2.122)$$

where

$$k_{lm} = u_i n_{lm_i}. \quad (2.123)$$

The distribution to the nodes for the upwind finite-volume scheme is then

$$R_l^{\text{FV}} = \sum_{\substack{m=1 \\ m \neq l}}^4 k_{lm}^- (\phi_m - \phi_l), \quad (2.124)$$

where use has been made of the fact that $k_{lm} = -k_{ml}$.

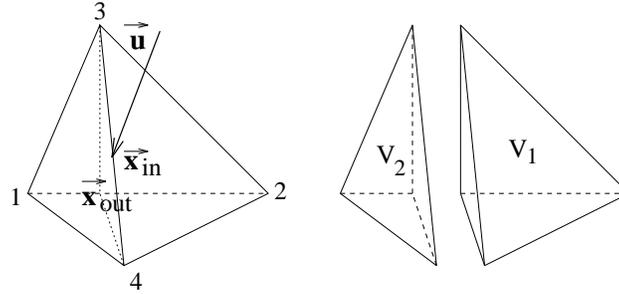


Figure 2.14: LDA-scheme configuration for two target nodes

2.6.4 The LDA-scheme $(\mathcal{M}\mathcal{U}, \mathcal{L}\mathcal{P}, \mathcal{C})$

The distribution coefficients for the LDA-scheme in three space dimensions are defined as:

$$\beta_l^{\text{LDA}} = \frac{k_l^+}{\sum_{m=1}^4 k_m^+}, \quad (2.125)$$

From this definition it is seen that β_l^{LDA} is bounded between 0 and 1. As for the 3D N-scheme the possible number of targets is 1, 2 and 3. The geometrical interpretation for the last two cases is shown in figure 2.14 and 2.15 respectively. In these figures \vec{x}_{in} is the inflow point, equation (2.105), and \vec{x}_{out} the outflow point, for which a similar formula exists:

$$\vec{x}_{\text{out}} = \frac{\sum_{m=1}^4 k_m^+ \vec{x}_m}{\sum_{m=1}^4 k_m^+}. \quad (2.126)$$

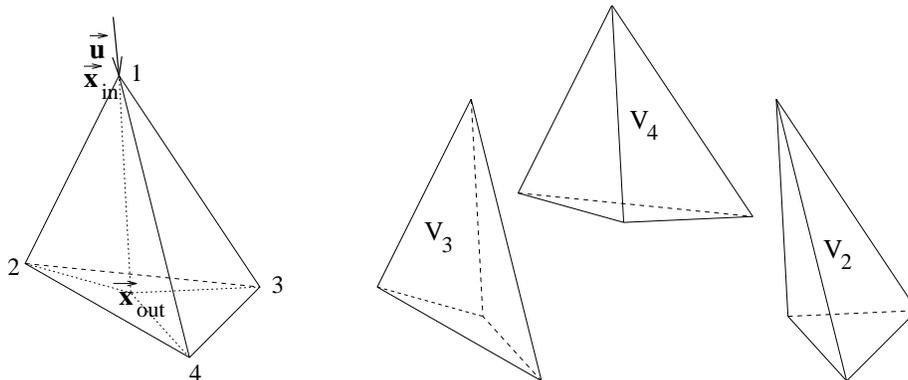


Figure 2.15: LDA-scheme configuration for three target nodes

The distribution coefficient β_l^{LDA} is equal to the ratio of subvolume V_l , see figures 2.14 and 2.15, and the volume of the entire tetrahedron.

2.6.5 The Lax-Wendroff scheme ($\mathcal{LP}, \mathcal{C}$)

The distribution coefficients of the 3D Lax-Wendroff scheme are:

$$\beta_l^{\text{LW}} = \frac{1}{4} + \frac{\Delta t}{3V_T} k_l, \quad (2.127)$$

where Δt is the cell time step and V_T the volume of the tetrahedron. As in two space dimensions, Rudgyard's definition [103] of the time step is used, which in 3D becomes:

$$\Delta t = \nu_c \frac{3V_T}{4 \sum_{m=1}^3 |k_m|}. \quad (2.128)$$

This results in:

$$\beta_l^{\text{LW}} = \frac{1}{4} + \nu_c \frac{k_l}{4 \sum_{m=1}^3 |k_m|}. \quad (2.129)$$

Again ν_c is the cell CFL number, which must be chosen between $\frac{1}{2}$ and 1.

2.6.6 The SUPG-scheme ($\mathcal{LP}, \mathcal{C}$)

Exactly the same modification to the Galerkin weight function is used as in 2D, see section 2.3.7. The result in the residual distribution form is:

$$\beta_l^{\text{SUPG}} = \frac{1}{4} + \frac{hk_l}{2 \|\mathbf{u}\| V_T}. \quad (2.130)$$

The addition of the nonlinear artificial dissipation term leads to:

$$\beta_l^{\text{SUPG+AV}} = \frac{1}{4} + \frac{hk_l}{2 \|\mathbf{u}\| S_T} + \frac{\hat{\kappa}}{R_T} \frac{\frac{1}{3} \vec{\nabla} \phi \cdot \vec{n}_l}{V_T}, \quad (2.131)$$

where the definition of the artificial viscosity $\hat{\kappa}$ is identical to the definition in two dimensions, equation (2.74).

2.7 Summary of the advection schemes

Table 2.1 summarizes the properties of the scalar schemes discussed in this chapter. It is clear that, of the schemes considered, only the PSI-scheme obeys all design criteria of section 2.2.

	Multi-dimensional upwind	Positive	Linearity preserving	Continuous
N-scheme	yes	yes	no	yes
PSI-scheme	yes	yes	yes	yes
Finite Volume scheme	no	yes	no	yes
LDA-scheme	yes	no	yes	yes
Lax-Wendroff scheme	no	no	yes	yes
SUPG-scheme	no	no	yes	yes

Table 2.1: Properties of the scalar schemes.

Chapter 3

System schemes

The goal of this work is the development of a robust multi-dimensional upwind discretization technique for non-commuting hyperbolic systems in general and the compressible Euler and high-Reynolds-number Navier-Stokes equations in particular. The former set of equations is hyperbolic in time, while the latter becomes hyperbolic in time in the limit $Re \rightarrow \infty$. Consequently the discretization technique for scalar advection, given in the previous chapter, must be extended to systems.

The early attempts of Roe, Deconinck and co-workers [26, 117, 120, 29, 85] were based on characteristic decompositions, *i.e.* finding characteristic forms of the governing equations which minimize the coupling terms, see equations (1.61). Next, the scalar schemes were applied to this "characteristic set" of advection equations where the coupling terms were treated as source terms. As it is impossible to discretize these terms, which are only zero for a commuting system, in a positive manner, instabilities inevitably occur at places of steep gradients. This leads to failure of the method and therefore the characteristic decompositions do not provide a very robust discretization technique.

A different approach has been followed here. Based on the observation that system generalizations of the scalar Lax-Wendroff and SUPG schemes exist since a long time [103, 49], an attempt has been made to extend the multi-dimensional upwind scalar schemes to systems. This is the main contribution of this thesis towards the development of a robust multi-dimensional upwind method. In this respect also the work of Sidilkover [109, 108, 110] must be mentioned, although his schemes are not multi-dimensional upwind in the sense of the definition given in section 2.2.1.

3.1 General concepts

Consider the hyperbolic system in time:

$$\frac{\partial U}{\partial t} + A_i \frac{\partial U}{\partial x_i} = 0. \tag{3.1}$$

Here U is a vector with N elements and the $N \times N$ Jacobian matrices A_i do not commute. As system (3.1) is hyperbolic in time, the generalized upwind parameters K_l ,

$$K_l = \frac{1}{d} A_i n_{li}, \quad (3.2)$$

where d is the number of spatial dimensions, have a complete set of real eigenvalues and eigenvectors. Consequently K_l can be written as:

$$K_l = R_l \Lambda_l L_l, \quad (3.3)$$

where the columns of R_l contain the right eigenvectors, Λ_l is a diagonal matrix of the eigenvalues and $L_l = R_l^{-1}$. The matrices K_l^+ and K_l^- , generalizing the scalar coefficients k_i^+ and k_i^- , are defined as:

$$K_l^+ = R_l \Lambda_l^+ L_l, \quad K_l^- = R_l \Lambda_l^- L_l. \quad (3.4)$$

Here Λ_l^+ contains the positive and Λ_l^- the negative eigenvalues: $\Lambda_l^\pm = \frac{1}{2}(\Lambda_l \pm |\Lambda_l|)$.

As for the scalar case, the cell residual R_T , which is a vector now, is obtained by integrating the spatial part of equation (3.1) over the control volume T , a triangle or a tetrahedron, resulting in:

$$\begin{aligned} R_T &= \iint_T A_i \frac{\partial U}{\partial x_i} dT \\ &= \oint_{\partial T} A_i n_i^{\text{ext}} U d\partial T. \end{aligned} \quad (3.5)$$

Assuming A_i constant per cell, and a linear variation of the elements of U (or equivalently integrating the contour integral of equation (3.5) with the trapezium rule) gives:

$$R_T = \sum_{l=1}^{d+1} K_l U_l. \quad (3.6)$$

Exactly as for the scalar case, fractions of R_T are distributed to the nodes and the nodal updates are obtained by assembling the contributions of all cells, leading to the semi-discretization:

$$\frac{dU_l}{dt} = -\frac{1}{S_l} \sum_T \beta_l^T R_T. \quad (3.7)$$

Also for the system case the mass matrix approximation (2.12) has been used. The coefficients β_l^T are now matrices and for consistency $\sum_m \beta_m^T = I$ (although this is not entirely true as will be shown in section 3.3.2), where the summation extends over the $d + 1$ vertices of the cell.

3.2 Properties of the system schemes

This section generalizes the scalar properties of Positivity and Linearity Preservation to systems and introduces the new property of Invariance for similarity transformations.

- **Positivity**

The generalization of the scalar property, described in section 2.2.2, is straightforward. Rewrite the semi-discrete form (3.7) of equation (3.1) as:

$$\frac{dU_l}{dt} + \sum_m C_{lm} (U_l - U_m) = 0. \quad (3.8)$$

Then the scheme is positive if all the matrices C_{lm} are non-negative, *i.e.* all their eigenvalues are positive or zero.

- **Linearity Preservation**

This is the ability to reproduce linear steady-state solutions exactly, also called k -exactness for linear polynomials. As in the scalar case this is satisfied if no updates are sent to the nodes when the cell residual R_T is zero. This implies that the eigenvalues of the distribution matrices β_l^T are bounded.

- **Invariance for similarity transformations**

An important design property of the system distribution schemes is that the residual sent to the nodes in conservative variables U , R_l^U , is independent of the choice of variables for which the actual distribution is performed. Consider a set of characteristic variables W , defined by the similarity transformation:

$$\partial W = \frac{\partial W}{\partial U} \partial U. \quad (3.9)$$

The hyperbolic system (3.1) transforms into:

$$\frac{\partial W}{\partial t} + B_i \frac{\partial W}{\partial x_i} = 0, \quad (3.10)$$

where

$$B_i = \frac{\partial W}{\partial U} A_i \frac{\partial U}{\partial W}. \quad (3.11)$$

The invariance property requires that the following relation between the residual in conservative variables, $R_l^U = \beta_l^U R_T^U$, and in characteristic variables, $R_l^W = \beta_l^W R_T^W$, is satisfied:

$$R_l^U = \frac{\partial U}{\partial W} R_l^W. \quad (3.12)$$

From equations (3.1), (3.6), (3.10) and (3.11) it is clear that the following must hold:

$$R_T^U = \frac{\partial U}{\partial W} R_T^W. \quad (3.13)$$

Combining this with equation (3.12) results in the following requirement for distribution matrices for the different set of variables:

$$\beta_l^W = \frac{\partial W}{\partial U} \beta_l^U \frac{\partial U}{\partial W}, \quad \beta_l^U = \frac{\partial U}{\partial W} \beta_l^W \frac{\partial W}{\partial U}. \quad (3.14)$$

For a commuting system, this property has as result that the discretization of the system (3.1) by system schemes equals the discretization of the characteristic system (3.10), which is a set of scalar advection equations, by scalar schemes and transforming the result back the the conservative variables U .

For the non-commuting case, the consequence of this invariance property is that characteristic decompositions, the early attempts to discretize the Euler equations, based on similarity transformations, such as the pseudo-Mach angle model [85] or the Deconinck-Hirsch decomposition [26] all lead to the same result. Furthermore, if in a certain set of characteristic variables an equation is fully decoupled, for example the entropy equation in the Euler equations, this is automatically detected by the system scheme and there is no need to transform the governing equations to this particular form, although in practice this is usually done for reasons of efficiency.

3.3 System schemes

This section presents the expressions for the system versions of the scalar schemes presented in the previous chapter. This generalization turns out to be straightforward for linear schemes, but it poses some severe problems for the nonlinear limited schemes.

3.3.1 The system N-scheme

This scheme, see [126, 128, 127], has been developed to overcome the stability problems of the characteristic decompositions near strong discontinuities and on high aspect ratio cells. For the system (3.1) the distribution to node l of the scheme is defined as:

$$R_l^N = K_l^+ (U_l - U_{in}), \quad (3.15)$$

where U_{in} ,

$$U_{in} = \left(\sum_{m=1}^{d+1} K_m^- \right)^{-1} \sum_{m=1}^{d+1} K_m^- U_m, \quad (3.16)$$

is the generalized inflow state, see equation (2.39) for the scalar case. The matrices K_l^+ and K_l^- are defined in (3.4). An alternative formulation is obtained when equations (3.15) and (3.16) are combined:

$$R_l^N = \sum_{m=1}^{d+1} C_{lm} (U_l - U_m), \quad C_{lm} = K_l^+ \left(\sum_{k=1}^{d+1} K_k^- \right)^{-1} K_m^- \quad (3.17)$$

Well-posedness of the scheme, that is the existence of $\left(\sum_k K_k^- \right)^{-1}$, is easy to prove for any hyperbolic system, see appendix A. Positivity though, that is that all eigenvalues of C_{lm} are positive or zero, is not so evident for a non-commuting system, as the argument that the product of three positive matrices is also positive does not necessarily hold. However Barth [14] showed that the N-scheme is energy stable, which implies positivity. Liu and Lax [68] proved that the energy functional is the only functional which can possibly be bounded for discretizations of multi-dimensional hyperbolic systems. This proof, given in appendix B for completeness, is only valid for a linear, symmetrizable system. Consequently, the system N-scheme can be used for any hyperbolic system, but it may not be positive if the system is not symmetrizable. Most systems, which model a physical conservation law, possess a convex entropy function, and can be symmetrized by their entropy variables [45]. The Euler equations belong to this class of symmetrizable systems, see e.g. [45] and appendix C.

The invariance for similarity transformations is easily shown. As the system N-scheme is a first-order scheme, it is more natural to show that equation (3.12) holds rather than (3.14). For the generalized upwind parameters of equation (3.10), K_l^W , a similar relation holds as for its Jacobian matrices B_i , see equation (3.11). Consequently

$$K_l^{W\pm} = \frac{\partial W}{\partial U} K_l^{U\pm} \frac{\partial U}{\partial W}. \quad (3.18)$$

Hence, the update in conservative variables U , when the distribution is calculated in characteristic variables W , is given by:

$$\begin{aligned} R_l^N &= \frac{\partial U}{\partial W} K_l^{W+} (W_l - W_{in}) \\ &= \frac{\partial U}{\partial W} K_l^{W+} \frac{\partial W}{\partial U} \frac{\partial U}{\partial W} (W_l - W_{in}) \\ &= K_l^{U+} (U_l - U_{in}). \end{aligned} \quad (3.19)$$

This is identical to equation (3.15), which is obtained by applying the scheme directly to the variables U .

3.3.2 The system PSI-scheme

As explained in section 3.2, the consequence of the similarity transformation invariance property is that for commuting systems the result of the system schemes equals the transformed result of the scalar schemes applied to the characteristic form (3.10). Consequently the following relation must hold for β_l^{PSI} :

$$\beta_l^{\text{PSI}} = \left(\sum_{m=1}^{d+1} \beta_m^{\text{N}^+} \right)^{-1} \beta_l^{\text{N}^+}, \quad (3.20)$$

where

$$\beta_l^{\text{N}} = \frac{\partial U}{\partial W} D_l \frac{\partial W}{\partial U}, \quad (3.21)$$

D_l is a diagonal matrix with the k^{th} element d_l^k given by:

$$d_l^k = \frac{l^k \cdot R_l^{\text{N}}}{l^k \cdot R_T}, \quad (3.22)$$

l^k are the rows of $\frac{\partial W}{\partial U}$ and $\beta_l^{\text{N}^+}$ is the matrix based on the positive eigenvalues of the distribution matrix β_l^{N} . Note that the elements of D_l are the scalar distribution coefficients of the the characteristic system (3.10).

For a non-commuting system the limiting procedure (3.20) can be applied as well. However the β_l^{N} are now implicitly defined by:

$$R_l^{\text{N}} = \beta_l^{\text{N}} R_T, \quad (3.23)$$

where the expressions for R_l^{N} and R_T are given in equations (3.15) and (3.6) respectively. As β_l^{N} is a matrix and R_l^{N} and R_T are vectors, equation (3.23) does not provide enough equations to determine β_l^{N} uniquely and a certain form must be chosen. A logical extension of equation (3.21) is:

$$\beta_l^{\text{N}} = R_l D_l L_l, \quad (3.24)$$

where R_l and L_l contain respectively the right and left eigenvectors of the generalized upwind parameter K_l . The k^{th} element d_l^k of the diagonal matrix D_l is given by:

$$d_l^k = \frac{l_l^k \cdot R_l^{\text{N}}}{l_l^k \cdot R_T}, \quad (3.25)$$

where l_l^k is the k^{th} row of L_l . In general, for this definition $\sum_m \beta_m^{\text{N}}$ will not sum up to I . In principle this is not necessary, because to be consistent the only requirement is:

$$\left(\sum_m \beta_m^{\text{N}} \right) R_T = R_T, \quad (3.26)$$

expressing that $\sum_m \beta_m^N$ must have at least one eigenvalue equal to one, with R_T the corresponding eigenvector. However, for the limiting procedure (3.20) the matrix $\left(\sum_{m=1}^{d+1} \beta_m^{N+}\right)$ must not be singular and for the formulation (3.24) this is not the case. Therefore a correction term is added to equation (3.24),

$$\beta_l^N = R_l D_l L_l + \beta_l^{\text{corr}}, \quad (3.27)$$

such that $\sum_m \beta_m^N = I$. Consequently β_l^{corr} must satisfy the following requirement:

$$\sum_{m=1}^{d+1} \beta_m^{\text{corr}} = I - \sum_{m=1}^{d+1} R_m D_m L_m. \quad (3.28)$$

Again this is not sufficient to determine β_l^{corr} completely, but it can be used as a constraint in a minimization problem, which leads to the following form for β_l^N :

$$\beta_l^N = R_l D_l L_l + \frac{1}{d+1} \left(I - \sum_{m=1}^{d+1} R_m D_m L_m \right). \quad (3.29)$$

It can be proved that for symmetrizable systems formulation (3.29) has real eigenvalues, because the formulation (3.29) is symmetric for symmetric systems. Furthermore it is invariant for similarity transformations (and therefore it obeys the one-dimensional limit), it gives the correct expression if a system is diagonalizable and it is continuous. However its convergence behavior is not satisfactory - it usually stalls after a reduction in the residual of only two orders of magnitude.

As the limiting procedure (3.20) is expensive (numerical determination of the eigenvectors and eigenvalues of β_l^N), an alternative is to apply scalar limiting in characteristic residuals, where the characteristic variables must be chosen by the user. Of course this approach is not invariant for similarity transformations, but the results are (at the moment) comparable with the matrix limiting approach. The scalar limiting also suffers from convergence problems.

3.3.3 The upwind finite-volume scheme

The system version of the upwind finite-volume scheme is easily obtained from its scalar counterpart, see sections 2.3.3 and 2.6.3. The distribution to the nodes is given by:

$$R_l = \sum_{\substack{m=1 \\ m \neq l}}^{d+1} K_{lm}^- (U_m - U_l), \quad (3.30)$$

where K_{lm} are the generalized upwind parameters based on the normals of the dual grid \vec{n}_{lm} . These normals are defined in equations (2.51) to (2.53) for the two-dimensional case and in equation (2.121) for three space dimensions.

3.3.4 The system LDA-scheme

The system LDA-scheme is a straightforward extension of its scalar equivalent, section 2.3.4:

$$R_i^{\text{LDA}} = \beta_i^{\text{LDA}} R_T = K_i^+ \left(\sum_{k=1}^{d+1} K_k^+ \right)^{-1} R_T. \quad (3.31)$$

Existence of $\left(\sum_k K_k^+ \right)^{-1}$ is proved in appendix A for any hyperbolic system. It is easy to verify that for a similarity transformation of the type (3.9) the distribution matrices $\beta_i^{\text{LDA},U}$ and $\beta_i^{\text{LDA},W}$ obey the condition of invariance for similarity transformations, equation (3.14).

3.3.5 The system Lax-Wendroff scheme

The generalization of the Lax-Wendroff scheme with the time-step definition of Rudgyard, equation (2.64), reads:

$$R_i^{\text{LW}} = \beta_i^{\text{LW}} R_T = \left[\frac{1}{d+1} I + \nu_c K_i \left(\sum_{m=1}^{d+1} |K_m| \right)^{-1} \right] R_T, \quad (3.32)$$

in which ν_c is the cell CFL number, chosen between $\frac{1}{2}$ and 1. Note that the time step definition (2.63) effectively introduces a matrix time step for a system, which ensures that the scheme is invariant for similarity transformations.

3.3.6 The system SUPG-scheme

The main difficulty of the extension of the SUPG finite-element method to non-commuting hyperbolic systems is the generalization of the scalar intrinsic time scale τ , equation (2.70). This generalized stabilization matrix Θ is easily constructed for commuting systems by discretizing the characteristic form with the scalar SUPG scheme and transforming the result back to the conservative formulation. A consistent approach for non-commuting systems, which for a commuting system reduces to the previously defined form, was first given in [49]:

$$\Theta = \frac{h}{2} (A_i A_i)^{-\frac{1}{2}}. \quad (3.33)$$

Harten [45] proved that the matrix $A_i A_i$ is a symmetric positive-definite matrix and consequently the inverse square root, needed for Θ , exists.

For linear triangles and tetrahedra the scheme, without nonlinear artificial dissipation, can be put in distribution form:

$$\beta_i^{\text{SUPG}} = \frac{1}{d+1} I + \frac{K_i \Theta}{S_T}. \quad (3.34)$$

Here S_T is either the volume of the tetrahedron (3D) or the surface of the triangle (2D). The addition of the nonlinear artificial dissipation is usually done in a scalar way, as explained in section 2.3.7. Note that this approach violates the invariance for similarity transformations.

Chapter 4

The time integration method

The spatial discretization for either the scalar equation, chapter 2, or the coupled system, chapter 3, leads to a system of ordinary differential equations in time:

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

where $U = (U_1, U_2, \dots, U_l, \dots)^T$ is the vector of nodal states and $R(U)$ the discretized spatial part. Note that for systems, the nodal states U_l are themselves vectors, with each N elements. In this work, only steady-state problems are considered and the time dependent equations are just used in an iterative framework to reach this state. Consequently the accuracy of the time discretization is not important, see also section 2.1, and time integration methods can be chosen for their robustness and steady-state convergence behavior properties, rather than accuracy.

A distinction is made between explicit and implicit methods: the former are straightforward to implement and are quite robust, however the time step is limited, see equation (2.28), and therefore the convergence to the steady-state can be slow, especially for high-Reynolds-number turbulent flows. The family of Runge-Kutta explicit time integration schemes are discussed in section 4.1. On the other hand, certain implicit methods do not have the time step limitation, at least not for linear model problems, and the steady-state solution can be reached much faster. The disadvantage of implicit methods is that at every time step a linear system must be constructed and solved, which seriously complicates the implementation. Only one implicit method is used, backward Euler, which is explained in section 4.2.

For large 3D computations the resources of a single processor are often not enough to handle the problem and it must be solved in parallel to obtain a solution within a reasonable amount of time. The parallelization technique, *i.e.* the method to distribute the load over the processors, is based on domain decomposition and is discussed in section 4.3.

4.1 Runge-Kutta methods

The m -stage Runge-Kutta time integrator for equation (4.1) is defined by the following algorithm:

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

$$U^{(k)} = U^{(0)} + \alpha^k \Delta t R(U^{(k-1)}), \quad k = 1, \dots, m \quad (4.2)$$

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

Here n is the counter of consecutive time levels and α^k are constants between 0 and 1, characteristic of the method used. Δt is the time step and to speed up the convergence to the steady-state, local time stepping is used, *i.e.* for every node equation (2.28) determines the time step. Consequently the time integration process is not time accurate. Local time stepping is especially effective in cases where the differences in cell sizes are large.

Furthermore, the coefficients α^k can be chosen to optimize the smoothing properties rather than accuracy. The values given in table 4.1 provide optimal damping for the scalar N-scheme in combination with multigrid, see [22], and are used in this work for single grid computations.

Despite the local time stepping technique and the optimal coefficients α^k , explicit time stepping is still much too time consuming, especially for high-Reynolds-number turbulent flows. Therefore, in general, implicit methods are preferred. There is an exception, namely inviscid supersonic flow with shocks, for which explicit time integrators become competitive with implicit methods, see section 5.4.2.

$m = 1$	$\alpha^1 = 1.0$					
$m = 2$	$\alpha^1 = 0.7$	$\alpha^2 = 1.0$				
$m = 3$	$\alpha^1 = 0.28$	$\alpha^2 = 0.61$	$\alpha^3 = 1.0$			
$m = 4$	$\alpha^1 = 0.16$	$\alpha^2 = 0.32$	$\alpha^3 = 0.57$	$\alpha^4 = 1.0$		
$m = 5$	$\alpha^1 = 0.10$	$\alpha^2 = 0.21$	$\alpha^3 = 0.34$	$\alpha^4 = 0.55$	$\alpha^5 = 1.0$	
$m = 6$	$\alpha^1 = 0.07$	$\alpha^2 = 0.14$	$\alpha^3 = 0.23$	$\alpha^4 = 0.34$	$\alpha^5 = 0.53$	$\alpha^6 = 1.0$

Table 4.1: Coefficients for the multi-stage Runge-Kutta time integrators.

4.2 The backward Euler method

The main cause of the slow convergence of the Runge-Kutta methods is the time step restriction due to stability reasons. If unconditionally stable time integration methods, *i.e.* methods which are stable independent of the time step, are used, a tremendous decrease in computation time can be expected. Unfortunately unconditional stability can only be achieved with certain implicit methods, see [25],

and then only for linear equations discretized with linear schemes. For nonlinear equations, such as the Euler and Navier-Stokes equations, or linear equations discretized with nonlinear numerical schemes, the information obtained from the stability analysis is limited and in practice the time step must still be restricted, albeit to a much higher value than for explicit methods.

A whole class of unconditionally stable time-integration methods can be derived, see [25, 51], which are at most second-order accurate in time. As the time integration is used as a relaxation method to obtain the steady-state solution, stability is preferred over accuracy and only the most stable method, backward Euler, is considered here. Applying this method to the semi-discrete equation (4.1) results in:

$$\frac{U^{n+1} - U^n}{\Delta t} = R(U^{n+1}). \quad (4.3)$$

Again local time stepping is used. If the governing equations and/or the numerical schemes are nonlinear, equation (4.3) gives a set of nonlinear algebraic equations for the nodal state vectors U^{n+1} , which must be solved in an iterative manner. If Newton's method is chosen, U^{n+1} is obtained by the algorithm below:

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

$$\left[\frac{I}{\Delta t} - \frac{\partial R}{\partial U}(U^{(k-1)}) \right] (U^{(k)} - U^{(k-1)}) = R(U^{(k-1)}) - \frac{I}{\Delta t} (U^{(k-1)} - U^{(0)}), \quad k = 1, \dots, m \quad (4.4)$$

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

Here I is the identity matrix and m is the number of Newton steps to solve equation (4.3) to a desired accuracy. In every step of the method (4.4) a linear system of equations must be solved. This can be done by either direct solvers, based on Gaussian elimination, or by iterative solvers. For the problems considered in this work, direct solvers are too expensive in both memory and computation time and iterative solvers must be used, see section 4.2.2. The result is that three nested loops can be distinguished to solve the steady-state problem $R(U) = 0$, see figure 4.1. The first loop is the time-integration process, the second is the Newton loop to solve the nonlinear system (4.3) for every time step and the third is the loop to solve the linear system (4.4) for every Newton step. It must be

do m = 1, #time-steps	Time loop
do k = 1, #Newton steps	Non-linear iterative loop in backward Euler
do l = 1, #linear steps	Linear iterative loop to solve linear system

Figure 4.1: The three nested loops in the iteration process.

mentioned that normally the second and the third loop are solved approximately, because only the outer loop should reach full convergence. To be more specific, in the second loop usually only one Newton step is taken and the so-called linear backward Euler scheme is obtained:

$$\left[\frac{I}{\Delta t} - \frac{\partial R}{\partial U}(U^n) \right] (U^{n+1} - U^n) = R(U^n). \quad (4.5)$$

Equation (4.5) is a linear system of equations for U^{n+1} , which must be (approximately) solved.

4.2.1 Jacobian computation

Implicit methods such as backward Euler¹ require the use of a so-called Jacobian matrix $\frac{\partial R}{\partial U}(U)$, which contains the derivatives of the nodal residuals with respect to the nodal state vectors. For reasons of clarity, the following notation is introduced: $R_l(U_l, U_m)$ denotes the nodal residual at node l , which is a function of the state vector at node l , U_l , and of U_m , all the state vectors at the nodes m , being the nearest neighbors of l . Due to the compactness of the stencil of the spatial discretization, see chapters 2 and 3, only the nearest neighbors need to be considered. The backward Euler method in node l then reads:

$$\left[\frac{I}{\Delta t_l} - \frac{\partial R_l(U_l, U_m)}{\partial U_l} \right] \Delta U_l - \sum_m \frac{\partial R_l(U_l, U_m)}{\partial U_m} \Delta U_m = R_l(U_l, U_m), \quad (4.6)$$

where $\Delta U_m = U_m^{n+1} - U_m^n$. Note that, because U_m is a vector, $\frac{\partial R_l}{\partial U_m}$ is a dense matrix whose number of rows and columns equals the number of elements of U_l and U_m respectively. Consequently the global Jacobian matrix $\frac{\partial R}{\partial U}$ is a sparse block matrix.

The most convenient way to construct the entries $\frac{\partial R_l}{\partial U_m}$ in the global Jacobian is to construct $\frac{\partial R_l}{\partial U_m}$ locally at the cell level first, simultaneously with the distribution of the cell residual R_T to the nodes, and next to put them in the correct place in the global Jacobian. In this work, these local Jacobians, or element stiffness matrices in finite-element terminology, are determined either analytically or numerically.

Analytical Jacobians

The exact derivatives of the spatial discretization with respect to the nodal states can be computed analytically. However, given the complexity of both the governing equations and the numerical schemes, the exact expressions become very complicated and errors are easily introduced in both the analytical derivation and the computer coding. Therefore only approximate analytical Jacobians are considered

¹If in the rest of this work it is mentioned that the backward Euler method is used, it is meant the linear backward Euler method.

here. In this approximation a distinction is made between first and second-order schemes.

For the hyperbolic system (3.1) the first order N-scheme distribution is given by equation (3.17). For a nonlinear system, the generalized upwind parameters K_l (3.2) are a function of the solution variables because of the conservative linearization, see sections 2.4 and 5.1. Consequently the K_l^+ and K_l^- matrices, occurring in the N-scheme distribution (3.17), should be differentiated to obtain the correct expressions for the Jacobians. However, if it is assumed that the differences between nodal states (in space) are small, these quantities are multiplied with a small vector and can therefore be neglected. Although this assumption is violated near discontinuities, it turns out to be that the resulting approximate Jacobian is more stable than the exact Jacobian, especially for moving discontinuities. The approximate Jacobian, also called the Picard Jacobian, is identical in form to the Jacobian of a linear system:

$$\frac{\partial R_l}{\partial U_m} = -K_l^+ \left(\sum_{k=1}^{d+1} K_k^- \right)^{-1} K_m^-, \quad m \neq l \quad (4.7)$$

$$\frac{\partial R_l}{\partial U_l} = K_l^+ \left[I - \left(\sum_{k=1}^{d+1} K_k^- \right)^{-1} K_l^- \right]. \quad (4.8)$$

For the Euler equations, the conservative linearization, section 5.1, slightly complicates the situation, although the principle of assuming small differences between nodal states remains. The N-scheme distribution in combination with the conservative linearization is given in equation (5.15). The Picard Jacobians with respect to the Z -variables, $\frac{\partial R}{\partial Z}$, are easily obtained and must be multiplied with the transformation matrix $\frac{\partial Z_l}{\partial U_l}$ to get the desired expressions $\frac{\partial R}{\partial U}$. If the algebra is properly carried out, equations (4.7) and (4.8) are multiplied by the term $\frac{\partial U}{\partial Z}(\bar{Z}) \frac{\partial Z_l}{\partial U_l}$. Although not done here, this term could be approximated by I , the identity matrix, which would be consistent with the assumption of small differences between nodal states.

For the finite-volume scheme, section 3.3.3, a similar analysis can be done, which results in:

$$\frac{\partial R_l}{\partial U_m} = -K_{lm}^-, \quad m \neq l \quad (4.9)$$

$$\frac{\partial R_l}{\partial U_l} = - \sum_{\substack{m=1 \\ m \neq l}}^{d+1} K_{lm}^- \quad (4.10)$$

For second-order schemes the distribution to the nodes is given by:

$$R_l = \beta_l R_T, \quad (4.11)$$

where the distribution matrices β_l are bounded. Assuming that the product of the derivatives of β_l and the cell residual R_T is small and the upwind parameters K_l are constant, the Jacobians become:

$$\frac{\partial R_l}{\partial U_m} = \beta_l K_m. \quad (4.12)$$

Here use is made of equation (3.6) for the cell residual R_T . For all linear second-order schemes the approximation (4.12) works quite satisfactorily, but for the nonlinear PSI-scheme, the N-scheme Jacobians (4.7) and (4.8) must be used for stability reasons. Numerical experience has resulted in the observation that the Picard Jacobians work most efficiently for CFL numbers around 100.

Numerical Jacobians

As a result of the approximations of the analytical Jacobians discussed in the previous paragraph, Newton convergence, even if possible, can never be obtained. On the other hand, the expressions of the exact Jacobians become very complicated and an alternative is to compute them numerically. An extensive study by Issman [51] has shown that a first-order numerical approximation is accurate enough to keep Newton convergence.

Consider the local Jacobian $\frac{\partial R_l}{\partial U_m}$. Its k^{th} column, *i.e.* the derivatives of R_l with respect to the k^{th} element of U_m , can be approximated by truncating the Taylor series after the first-order term:

$$\left(\frac{\partial R_l}{\partial U_m} \right)_k \approx \frac{R_l(U_m + \varepsilon \vec{1}_k) - R_l(U_m)}{\varepsilon}. \quad (4.13)$$

Here $\left(\frac{\partial R_l}{\partial U_m} \right)_k$ indicates the k^{th} column of $\frac{\partial R_l}{\partial U_m}$ and $U_m + \varepsilon \vec{1}_k$ is the state vector U_m , whose k^{th} element has been perturbed by a small quantity ε . The value of ε is taken as:

$$\varepsilon = 10^{-7} \text{sign}(U_{m,k}) \max(|U_{m,k}|, 0.01), \quad (4.14)$$

where $U_{m,k}$ is the k^{th} element of U_m and the sign function is given by:

$$\text{sign}(r) = \begin{cases} -1 & \text{if } r < 0 \\ 1 & \text{if } r \geq 0 \end{cases}. \quad (4.15)$$

Equation (4.13) shows that one column of $\frac{\partial R_l}{\partial U_m}$ can be calculated with one extra residual evaluation. Consequently, N additional residual computations are required to determine $\frac{\partial R_l}{\partial U_m}$ completely, where N is the number of elements of U_m . Due to the compactness of the stencil of the spatial discretization, the Jacobians can be determined simultaneously with the distribution of the cell residual to the nodes at the price of N extra residual evaluations per nodal state vector. As the control volumes, triangles and tetrahedra, have $d+1$ nodes, where d is the number

of spatial dimensions, the complete local Jacobians can be computed numerically with $(d + 1)N$ additional residual computations. For the 2D Euler equations this number equals 12, but for the 3D turbulent Navier-Stokes equations with a two-equation turbulence model it is already 28. These numbers would be acceptable if indeed Newton convergence could be achieved. However, in practice this is not the case when the PSI-scheme is used and also not for turbulent problems, which require a special treatment to remain stable, see section 6.2.1. Consequently the numerical evaluation of the Jacobians is too expensive for these kinds of problems and the approximate forms are used, except for axisymmetric problems, see section 6.3.1.

4.2.2 Linear iterative solvers

At every time step a linear system of the type $Ax = b$ must be solved, see equation (4.5). The available memory of current computers excludes the use of direct solvers for the problems considered and iterative solution methods must be used. As the main interest of this thesis is the development and application of the multi-dimensional upwind discretization technique and not iterative solution methods for linear systems, attention will be restricted to general information and the reader is referred to the references [106, 105, 124, 112, 51] for the details. Moreover, the linear systems arising in the backward Euler method are solved by the AZTEC library [50] of Sandia National Laboratories and this library is used more or less as a black box in the computer code.

The Jacobian matrix of the spatial discretization is not symmetric definite and therefore algorithms for the solution of general matrices must be used. For this purpose AZTEC has been equipped with the so-called Krylov subspace methods such as GMRES [106], Bi-CGstab [124], CGS [112] and TFQMR [40]. Usually GMRES is the most stable of all methods, but it is also the most expensive in terms of memory. Bi-CGstab and TFQMR are compromises between stability and required memory, while the convergence behavior of CGS is normally too erratic to be a stable method, unless very low CFL numbers (< 10) are used. In this work GMRES is used in all the implicit computations.

For the efficient solution of realistic linear systems, it is essential to combine the Krylov method with a preconditioning technique, which transforms the original system $Ax = b$ into the better-conditioned system:

$$MAx = Mb, \tag{4.16}$$

where M is a non-singular preconditioning matrix. The $M = A^{-1}$ preconditioner leads to the solution of the problem in one iteration, but this is not feasible, because the computation of A^{-1} is more expensive than the solution of the linear system itself. Therefore M should approximate A^{-1} as closely as possible, while still being reasonably cheap to compute. At the present time, a standard preconditioner giving quite good results is the ILU(0) preconditioner, which is also available

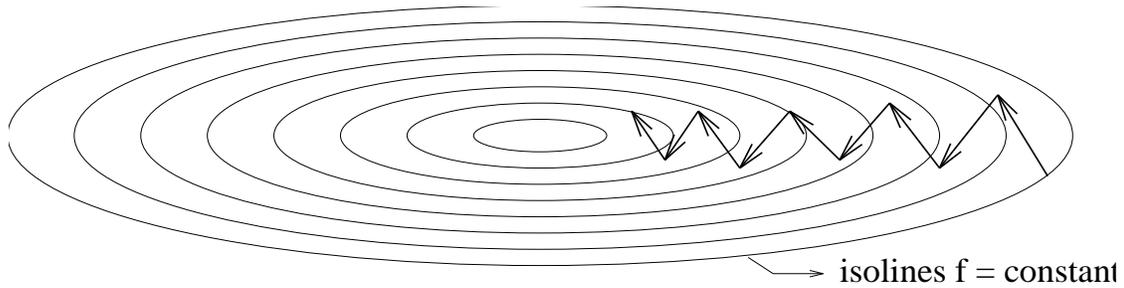


Figure 4.2: Method of steepest descent for the function $f(x, y) = \frac{x^2}{a^2} + \frac{y^2}{b^2}$, $a \gg b$. The arrows indicate one iteration step.

in AZTEC. It is constructed from the exact factorization formula of the matrix A , see e.g. [51], under the condition that the sparsity pattern of the original matrix is maintained. This immediately explains the name, as ILU stands for Incomplete Lower Upper and the 0 means that no additional entries, compared with the original matrix, are allowed. For systems, for example the Euler and Navier-Stokes equations, the Jacobian matrix is a block matrix and to be efficient also the preconditioner should be a block matrix. To make a distinction with the scalar case, the preconditioner is now called BILU(0), where the B indicates Block.

The BILU(0) method requires the storage of an additional matrix of the size of the original matrix, which can be too memory intensive for large 3D problems. Therefore a modified version of BILU(0), BMILU(0), has been implemented in AZTEC, which only requires the storage of an additional block diagonal, while the performance only slightly decreases.

To illustrate the effect of preconditioning, the following example is given. Consider the function:

$$f(x, y) = \frac{x^2}{a^2} + \frac{y^2}{b^2}, \quad a \gg b, \quad (4.17)$$

for which a minimum must be determined. One way to do this numerically is the method of steepest descent, shown graphically in figure 4.2. A starting point for this method is chosen randomly. A step starts off in the direction of the local gradient, *i.e.* normal to the isolines, and terminates if a local minimum of $f(x, y)$ on the line in that direction is reached. This point is the starting point of the next step and so on, until the gradient vanishes. If $a \gg b$, the isolines of $f(x, y)$ are ellipses with high eccentricity, see figure 4.2, and many iterations must be taken to reach the minimum. However if one considers the variables:

$$x^* = \frac{x}{a}, \quad y^* = \frac{y}{b}, \quad (4.18)$$

the function $f(x^*, y^*)$ reads:

$$f(x^*, y^*) = (x^*)^2 + (y^*)^2. \quad (4.19)$$

The isolines in this transformed coordinate system are concentric circles, see figure 4.3 and consequently the method of steepest descent converges in one iteration,

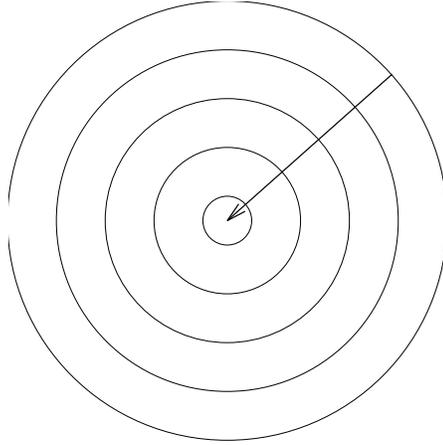


Figure 4.3: Method of steepest descent for the modified function $f(x^*, y^*) = (x^*)^2 + (y^*)^2$. The algorithm converges in one iteration, indicated by the arrow.

independent of the starting point. So by simply scaling the problem, the numerical algorithm converges much faster and this is exactly the intention of preconditioning. Note that in this particular example the coordinate transformation (4.18) corresponds with the preconditioner $M = A^{-1}$, because the preconditioned problem converges in one iteration, independent of the starting point.

To improve the performance of the preconditioner for the system $Ax = b$, a node renumbering scheme is used to approximately minimize the bandwidth of the Jacobian matrix. A simple method with fairly good results is the reverse Cuthill-McKee algorithm [67]. An example is seen in figure 4.4, which shows the Jacobian entries of the original numbering, obtained from the grid generator, and the reordered numbering respectively. The given matrices correspond to a grid consisting of 4626 nodes for a subsonic channel with a circular bump. The convergence histories, L2 norm of the density residuals as function of CPU time on a DEC alpha workstation (EV5.6 processor, 500 MHz), for the system N-scheme

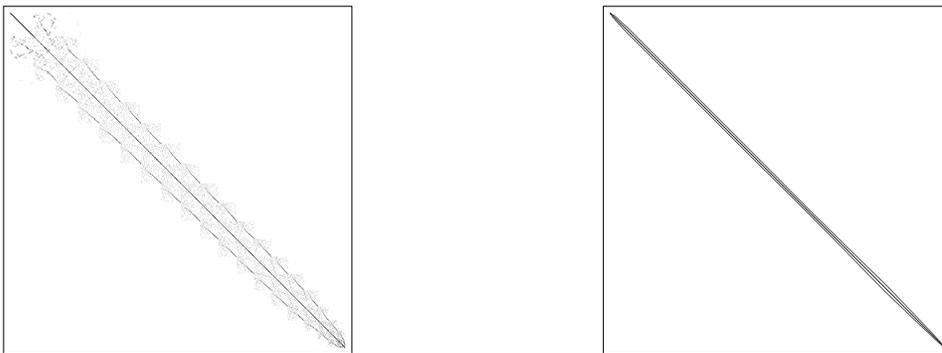


Figure 4.4: Jacobian entries for the original (left) and reordered (right) node numbering.

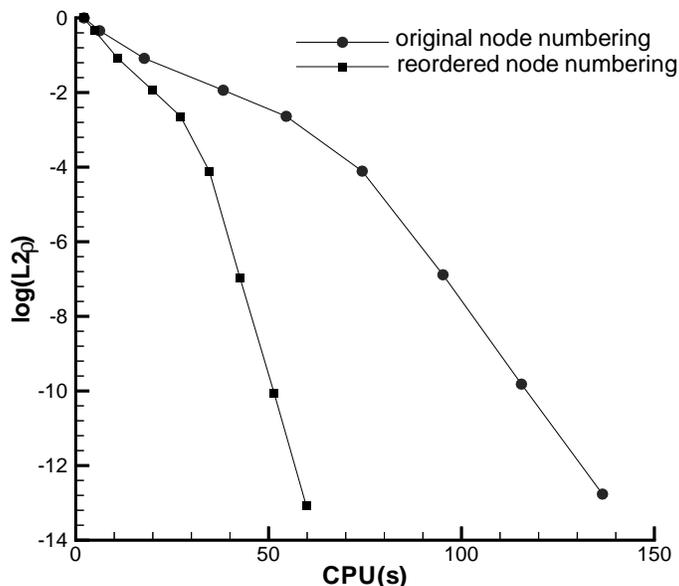


Figure 4.5: Convergence histories for the channel flow for the original and reordered node numbering.

directly applied to the Euler equations are shown in figure 4.5. The linear systems of the backward Euler time integrator are solved with BMILU(0) preconditioned GMRES. As can be seen in figure 4.5 the CPU-time is more than halved if the bandwidth is minimized, so it is definitely worthwhile. This gain in performance is even more impressive if one realizes that the time to build the Jacobian matrix and the right-hand side are identical for both problems and the saving only originates from the linear solver. The heuristic explanation of this observation is simple. The ILU(0) and MILU(0) preconditioners are obtained by ignoring the fill-in in the standard LU decomposition. As this fill-in only occurs within the bandwidth, it can be expected that ILU(0) and MILU(0) are better approximations of LU if the bandwidth is minimized. Consequently better preconditioners are obtained, which results in fewer GMRES iterations to solve to linear system.

Two remarks must be made. First, the saving in CPU-time in 3D is less spectacular than in 2D, because even the minimized bandwidth is still rather large, although it is always worthwhile. Indeed the Cuthill-McKee algorithm uses about 2 minutes of CPU-time on the same alpha workstation for a one million node grid. Second, if the performance of the preconditioner is independent of the bandwidth, e.g. Jacobi, minimization does not make sense, because the Krylov solver also performs independently of the bandwidth.

4.2.3 Memory requirements

From the previous sections it is clear that the fully-implicit backward Euler method requires the storage of the full Jacobian matrix and, at best, an additional block diagonal for the preconditioning. Furthermore the Krylov solver itself needs to store some vectors. Consequently the method can be very memory consuming and one might question if not too much so.

First consider the 2D case. For a regular grid, be it triangulated structured, figures 2.6 and 2.7, or unstructured, figure 2.8, the number of nearest neighbors is 6 and therefore the number of entries in the Jacobian matrix is 7 per node. Add the additional storage of a block diagonal for the BMILU(0) preconditioner, some memory for the linear solver itself and miscellaneous things (e.g. nodal state vectors), one gets roughly $8N^2 + 35N$ floating-point values that must be stored per node. In double precision a floating-point variable is 8 bytes long and therefore $64N^2 + 280N$ bytes of RAM are needed per node. For the Euler or laminar Navier-Stokes equations, $N = 4$ in 2D, this means 2.1 kbytes per node. Nowadays a workstation has easily 256 Mbytes of RAM and suppose that 200 can be used for the actual computation. Consequently on this type of workstations grids up till about 100,000 nodes can be treated, usually enough for practical computations. The conclusion of this exercise is that in 2D the memory requirement for the fully implicit backward Euler method is not an issue to be concerned about. If the turbulence equations are solved in a decoupled manner, see section 6.2.1, the additional memory overhead for turbulent computations is in the order of 10% and almost the same analysis applies.

In 3D the situation is different. The average number of nearest neighbors is 15 and about $17N^2 + 35N$ floating-point values must be stored per node, which requires $136N^2 + 280N$ bytes. For $N = 5$, Euler or Navier-Stokes equations, this results in 4.7 kbytes per node, and consequently roughly 225,000 nodes can be used per Gbyte of available RAM memory. Jameson [53] concludes that 10^7 nodes are needed to obtain a grid-converged turbulent solution for a transonic wing and thus about 45 Gbytes of RAM would be needed if the fully implicit backward Euler method is used for such problems. Issman [51] mentions that it is sufficient to store the Jacobians in single precision (4 bytes), which leads to something like 25 Gbytes for 10^7 nodes. This amount of memory is only available at a few parallel supercomputers, which makes the method (at the moment) not practical for so many nodes. In this work a maximum number of 1 million nodes is used for 3D computations, and for those problems it is nowadays possible to use a fully implicit method, albeit in parallel, see next section.

4.3 The parallelization technique

The latest generation of supercomputers, Cray T3, IBM SP2, Intel Paragon etc. are parallel machines with distributed memory. In order to be able to use them, the

domain decomposition technique is used. This means that the grid is decomposed into subdomains and every processor is responsible for at least one subdomain. Communication is used to exchange information between processors. The AZTEC library, which has been designed to solve linear systems on parallel computers, completely hides the details of the communication for the user. The only restriction is that every processor has one subdomain, *i.e.* the grid must be decomposed into the same number of subdomains as processors. This is easily achievable on unstructured grids as used in this work. The actual decomposition is done with the graph partitioning software METIS [61], which is public domain software.

AZTEC requires that the domain decomposition is done with the so-called Vertex Oriented Decomposition (VOD) technique, see figure 4.6. As a result, the separation between neighboring partitions occurs at the edge-level and nodes are

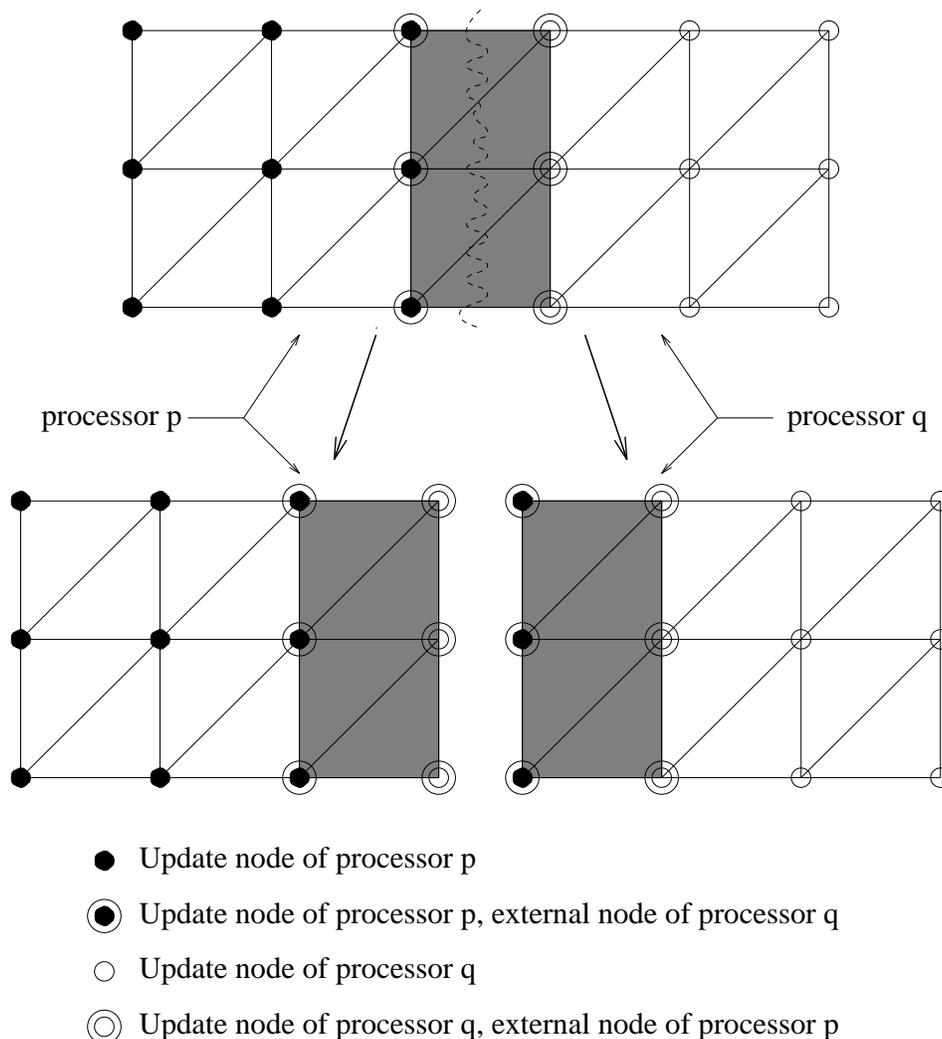


Figure 4.6: VOD, Close-up of the overlapping stripe and respective mapping onto processors p and q, update and external nodes.

assigned to a particular processor, which form its update set. The cut cell is stored on all partitions/processors, which have at least one node of the cell in their update set. So in figure 4.6 the shaded cells are stored on both processor p and processor q . The nodes belonging to cut cells, but not to the update set of a certain processor are called external nodes, see figure 4.6, and are also stored. These nodes of course belong to update sets of other processors and communication is needed to obtain their new values after every time step. As long as these external nodes have the correct value, all operations to build the residuals and Jacobians for the update nodes can be carried out in the same way as in the sequential algorithm. As soon as these operations have been completed, AZTEC solves the linear system in parallel, in which computation and communication overlap as much as possible, resulting in a smaller parallel overhead.

The only sequential bottleneck in the parallel Krylov algorithm is the inversion of the BILU(0) and BMILU(0) preconditioners. For this reason the parallel version of those preconditioners is modified by ignoring the interdomain links, the symbols \mathbf{x} in figure 4.7. The modified preconditioner can now be inverted domain by domain and the whole algorithm is fully parallel again. As a result of ignoring the interdomain links, the preconditioner will perform worse than its sequential counterpart and the linear solver is expected to need more Krylov iterations to reach the same level of convergence.

To determine the speed-ups on a different number of processors, Détery's test

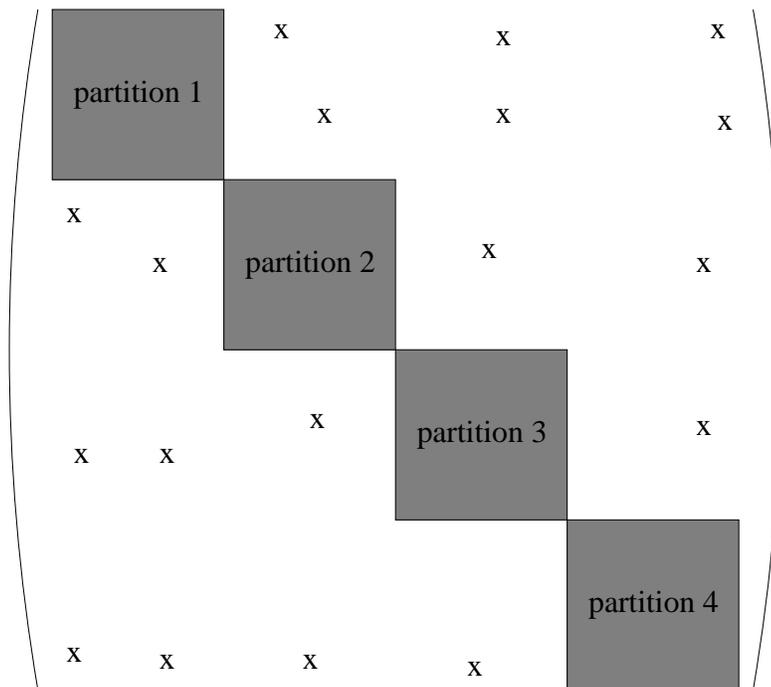


Figure 4.7: Ignoring the interdomain links, indicated by \mathbf{x} , for the parallel BILU(0) and BMILU(0) preconditioners.

# procs	CPU-time (s)	Speed up w.r.t. 1 procs	Speed up w.r.t. N/2 procs
1	$2.3910 \cdot 10^4$	-	-
2	$1.3414 \cdot 10^4$	1.78	1.78
4	$7.2687 \cdot 10^3$	3.29	1.85
8	$3.8505 \cdot 10^3$	6.21	1.89
16	$2.1943 \cdot 10^3$	10.90	1.75
32	$1.2863 \cdot 10^3$	18.44	1.71

Table 4.2: Speed-ups for the Détery test case C.

case C, a turbulent shock wave boundary layer interaction, see section 6.3.2, has been computed on 1, 2, 4, 8, 16 and 32 processors of the Cray T3E of the Delft University of Technology. For comparison, 1500 fully-implicit backward Euler time steps are taken for the first-order N-scheme, starting from a uniform flow field. To keep the CPU-times within limits, especially for one processor, a relatively coarse grid of 10,420 nodes has been used. The results are given in table 4.2. In every time step the Krylov solver has been terminated as soon as it has converged 2 orders of magnitude.

The following observations can be made. The speed-up from 1 to 2 processors is 1.78, quite far below the linear speed-up value of 2. This is almost entirely caused by the decreasing performance of the parallel preconditioner compared to its sequential counterpart, as on two processors $\pm 25\%$ more GMRES iterations were needed than one processor to reach the same level of accuracy for the linear solver. This effect is also present when more processors are used, although the relative decrease in performance of the preconditioner becomes less. For example, the speed-up from 4 to 8 processors is 1.89 and only $\pm 10\%$ more GMRES iterations were needed.

Furthermore, the grid is too coarse for execution on 16 or 32 processors. In the latter case only 325 nodes belong to the update set of each subdomain and therefore the relative amount of external nodes, see figure 4.6, is high. Consequently, the parallel overhead becomes rather large and the speed-ups decrease again - 1.71 when going from 16 to 32 processors.

Taking both the factors mentioned above into account, the values given in table 4.2 are quite satisfactory.

Chapter 5

The Euler equations

In this chapter the spatial discretization technique, described in detail in chapter 2 and 3, is applied to the Euler equations, written in conservation form as:

$$\frac{\partial U}{\partial t} + \frac{\partial F_j}{\partial x_j} = 0, \quad (5.1)$$

where the conservative variables U and the flux-vectors F_j are defined as:

$$U = \begin{pmatrix} \rho \\ \rho u_i \\ \rho E \end{pmatrix}, \quad (5.2)$$

$$F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ \rho H u_j \end{pmatrix}. \quad (5.3)$$

Here δ_{ij} is the Kronecker delta function and all other symbols have their usual meaning. Furthermore the thermally and calorically perfect gas assumption is made, for which the following relations hold:

$$H = E + \frac{p}{\rho}, \quad (5.4)$$

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \rho u_i u_i \right) = \frac{\gamma - 1}{\gamma} \left(\rho H - \frac{1}{2} \rho u_i u_i \right), \quad (5.5)$$

where the specific heat ratio γ is assumed to be constant and equal to 1.4. The corresponding quasi-linear form of (5.1) is:

$$\frac{\partial U}{\partial t} + A_j \frac{\partial U}{\partial x_j} = 0, \quad A_j = \frac{\partial F_j}{\partial U}. \quad (5.6)$$

The outline of this chapter is as follows: First the conservative linearization, which links the conservative form (5.1) with the quasi-linear form (5.6), is given, followed by the treatment of the boundary conditions. Following this, the Euler equations are written in the most optimal decoupled form by means of preconditioning, which improves the accuracy of the spatial discretization, as will be shown in the result section concluding this chapter.

5.1 Conservative linearization

The discretization technique described in this work, see chapter 2 and 3, needs the quasi-linear form of the equations. To retain conservation and hence capturing of discontinuities with their correct jump relations, a particular quasi-linear form is used, which is simultaneously conservative. The constraint it must obey is obtained in the same way as for the scalar equation, see section 2.4: integrate the spatial part of the conservative form (5.1) over a control volume T and perform the same operation for the quasi-linear form (5.6) with the assumption of constant Jacobian matrices per cell and require both forms to be equal. If Gauss' theorem is used, this results in:

$$\oint_{\partial T} F_j n_j^{\text{ext}} d\partial T \equiv \bar{A}_j \oint_{\partial T} U n_j^{\text{ext}} d\partial T, \quad (5.7)$$

where \bar{A}_j is a cell average of A_j . To be sure that the cell average Jacobians \bar{A}_j have the same properties as A_j , a sufficient condition is:

$$\bar{A}_j = A_j(\bar{U}), \quad (5.8)$$

i.e. A_j is evaluated at a cell average state \bar{U} . An additional degree of freedom can be introduced by noting that in the right-hand side of equation (5.7) it is not necessary to use the conservative variables U . Any set of independent variables, from now on called Z , can be used. This results in the following constraint for the cell-average state \bar{Z} :

$$\oint_{\partial T} F_j n_j^{\text{ext}} d\partial T \equiv \frac{\partial F_j}{\partial Z}(\bar{Z}) \oint_{\partial T} Z n_j^{\text{ext}} d\partial T. \quad (5.9)$$

The distribution schemes require a linear variation of the components of Z along the edges of the control volume, which for triangles and tetrahedra is equivalent to a linear variation over the entire control volume. This completely fixes the value of the contour integral on the right hand side of equation (5.9) and leaves the choice of variables Z and the integration rule for the fluxes as degrees of freedom.

In general it can be quite difficult to determine \bar{Z} . As for the scalar case, section 2.4, the situation drastically simplifies if a set of variables Z can be found for which the fluxes F_j are quadratic expressions in the elements of Z , the so-called Roe parameter vector [99]. For the Euler equations, this parameter vector exists and is given by:

$$Z = \sqrt{\rho} \begin{pmatrix} 1 \\ u_i \\ H \end{pmatrix}. \quad (5.10)$$

As explained for the Burgers equation, see equations (2.83) to (2.85), it can be verified [28] that the linearization (5.9) based on the arithmetic average of the Z variables,

$$\bar{Z} = \frac{1}{d+1} \sum_{k=1}^{d+1} Z_k, \quad (5.11)$$

is conservative, where d is the number of spatial dimensions and Z_k the value of Z in vertex k . Equation (5.9) might suggest that the generalized upwind parameters K_l , equation (3.2), have to be based on the Jacobians $\frac{\partial F_j}{\partial Z}$. This is not necessary, as the right hand side of equation (5.9) can be rewritten as:

$$\frac{\partial F_j}{\partial Z}(\bar{Z}) \oint_{\partial T} Z n_j^{\text{ext}} d\partial T = A_j(\bar{Z}) \frac{\partial U}{\partial Z}(\bar{Z}) \oint_{\partial T} Z n_j^{\text{ext}} d\partial T, \quad (5.12)$$

where the conservative flux Jacobians A_j are defined in equation (5.6). The matrices K_l are based on these Jacobians while consistent nodal values of the conservative variables $U_l, U_l^{\text{consistent}}$, are introduced:

$$U_l^{\text{consistent}} = \frac{\partial U}{\partial Z}(\bar{Z}) Z_l, \quad (5.13)$$

satisfying

$$\frac{\partial U^{\text{consistent}}}{\partial x_i} = \frac{\partial U}{\partial Z}(\bar{Z}) \frac{\partial Z}{\partial x_i} \quad (5.14)$$

at the cell level. To illustrate this procedure, the nodal distribution of the system N-scheme (section 3.3.1) for the Euler equations is given by:

$$\begin{aligned} R_l^{\text{N}} &= \sum_{m=1}^{d+1} C_{lm}(\bar{Z}) (U_l^{\text{consistent}} - U_m^{\text{consistent}}) \\ &= \sum_{m=1}^{d+1} C_{lm}(\bar{Z}) \frac{\partial U}{\partial Z}(\bar{Z}) (Z_l - Z_m), \end{aligned} \quad (5.15)$$

$$C_{lm} = K_l^+ \left(\sum_{k=1}^{d+1} K_k^- \right)^{-1} K_m^-. \quad (5.16)$$

5.2 Boundary conditions

The conservative linearization and the system schemes of chapter 3 completely determine the discretization of an interior node, *i.e.* a node which does not belong to the boundary of the computational domain. However, for nodes that do belong to the boundary, the contribution from the boundary conditions must be taken into account. For well-posedness of a hyperbolic problem, the number of imposed boundary conditions should be equal to the number of characteristics which enter the computational domain, see [136, 46]. This number is identical to the number of positive eigenvalues of the upwind parameter K_l based on the normal pointing into the domain. For the Euler equations these eigenvalues are:

$$\begin{aligned} \lambda_1 &= \vec{u} \cdot \vec{N} + a, & 1 \times & \text{acoustic wave} \\ \lambda_2 &= \vec{u} \cdot \vec{N} - a, & 1 \times & \text{acoustic wave} \\ \lambda_k &= \vec{u} \cdot \vec{N}, \quad 3 \leq k \leq d+2 & d \times & \text{entropy and shear waves} \end{aligned} \quad (5.17)$$

where a is the speed of sound,

$$a = \sqrt{\frac{\gamma p}{\rho}}, \quad (5.18)$$

\vec{N} the unit normal pointing into the domain and d again the number of spatial dimensions. The following boundary conditions may be identified for the Euler equations:

- Supersonic inlet
- Supersonic outlet
- Subsonic inlet
- Subsonic outlet
- Far field
- Impermeable wall
- Symmetry plane, with the axisymmetric centerline as a special case

These will be discussed in the following sections.

5.2.1 Supersonic inlet

For a supersonic inlet all characteristics (5.17) enter the domain and consequently the complete state vector must be prescribed. In practice this means that nodes belonging to a supersonic inlet boundary are not updated in the time integration loop.

5.2.2 Supersonic outlet

All characteristics leave the domain and therefore no boundary condition must be prescribed. This makes the treatment of supersonic outlet nodes identical to interior nodes.

5.2.3 Subsonic inlet

A subsonic inlet boundary is characterized by a velocity less than the speed of sound and consequently the eigenvalue λ_2 , see equation (5.17), is negative. Therefore imposing the entire state vector is not permitted. The usual choice is to prescribe the total pressure p_t , total density ρ_t and the flow direction. In two space dimensions this flow direction is completely determined by the angle of incidence α ; in three space dimensions both the angle of incidence α and the side slip angle β must be prescribed, see figure 5.1. To complete the state vector, one

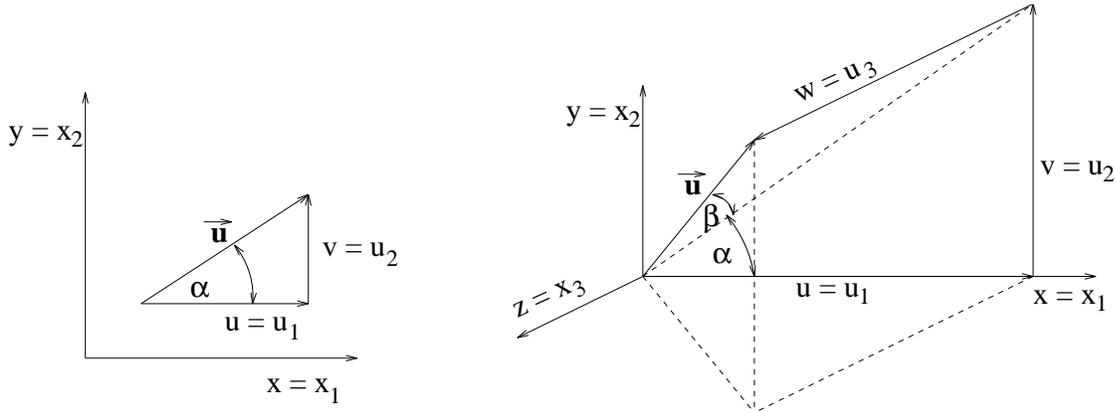


Figure 5.1: Definition of the angle of incidence α in 2D and the angle of incidence α and side slip angle β in 3D.

variable must be extrapolated from the computational domain. Theoretically this should be the characteristic variable corresponding to λ_2 . However this variable depends on the unit interior normal \vec{N} and only exists in differential form, which gives some practical problems. Therefore in this work simply the Mach number, $M = \sqrt{u_i u_i / a^2}$, is extrapolated.

In general there are two possibilities to impose the above boundary conditions: (1) strongly - every node on the boundary exactly obeys the imposed conditions, or (2) weakly - the effect of the boundary conditions is taken into account by means of an additional flux through the boundary faces. The strongly-enforced, nodal, boundary condition is solved by a one-dimensional flow problem in the direction of the nodal normal, see [84]. However, nodal normals are not well-defined if the boundary is not smooth, for example at sharp trailing edges or corners. Consequently imposing the boundary conditions becomes somewhat arbitrary. This is especially true in three space dimensions. Therefore only weakly-enforced boundary conditions, which use the well-defined face normals, are used in this work, except for supersonic inlets and outlets (sections 5.2.1 and 5.2.2), symmetry planes (section 5.2.7), and walls with no-slip boundary conditions, which are treated in the next chapter.

Consider the boundary face Γ_l with nodes 1 and 2 and the scaled inward normal \vec{n} , see figure 5.2. The ghost nodes 1* and 2* have the same geometrical position as 1 and 2 respectively, but their state vector is determined by the imposed boundary conditions completed with the extrapolated variables from the domain. In case of the subsonic inlet p_t , ρ_t and the flow direction are prescribed and the Mach number is taken equal to values in nodes 1 and 2 for 1* and 2*. This procedure leads to a flux difference between the ghost nodes and the ordinary nodes, whose integral along Γ_l is given by:

$$\int_{\Gamma_l} [F_j(U^*) N_j - F_j(U) N_j] d\Gamma_l, \quad (5.19)$$

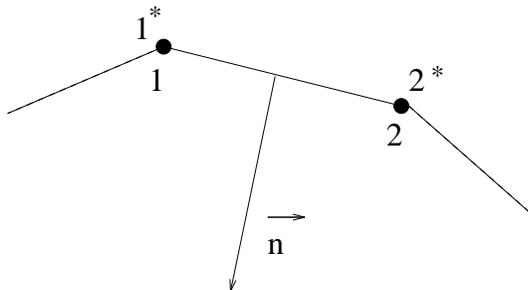


Figure 5.2: Boundary face Γ_l with nodes 1 and 2, ghost nodes 1^* and 2^* , and scaled inward normal \vec{n} .

where F_j are the flux functions of the Euler equations, see equation (5.3), and N_j is the component of \vec{N} in the x_j direction. \vec{N} is the corresponding unit normal of \vec{n} . A discrete approximation of equation (5.19) using the trapezium rule results into:

$$\frac{1}{2} [\{F_j(U_{1^*})n_j - F_j(U_1)n_j\} + \{F_j(U_{2^*})n_j - F_j(U_2)n_j\}]. \quad (5.20)$$

As for the Euler equations both the fluxes F_j and the conservative variables U are quadratic expressions in the elements of the Roe parameter vector Z , the following relation is valid [99]:

$$F_j(U_m)n_j - F_j(U_l)n_j = 2K_{ml}(U_m - U_l), \quad (5.21)$$

where K_{ml} is the generalized upwind parameter based on \vec{n} , see equation (3.2), and evaluated at the Roe averaged state $\bar{Z} = \frac{1}{2}(Z_m + Z_l)$. Consequently equation (5.20) can be written as:

$$K_{1^*1}(U_{1^*} - U_1) + K_{2^*2}(U_{2^*} - U_2). \quad (5.22)$$

To take into account that only information corresponding to incoming characteristics is used, the matrices K in equation (5.22) are replaced by K^+ . Therefore the following term must be added to the discretization of the cell to which Γ_l belongs:

$$K_{1^*1}^+(U_{1^*} - U_1) + K_{2^*2}^+(U_{2^*} - U_2) \quad (5.23)$$

and distributed to the nodes 1 and 2. This distribution is done as follows:

$$R_1^{\text{BC}} = \alpha K_{1^*1}^+(U_{1^*} - U_1) + (1 - \alpha) K_{2^*2}^+(U_{2^*} - U_2) \quad (5.24)$$

$$R_2^{\text{BC}} = \alpha K_{2^*2}^+(U_{2^*} - U_2) + (1 - \alpha) K_{1^*1}^+(U_{1^*} - U_1) \quad (5.25)$$

where α is a coefficient between $\frac{1}{2}$ and 1. It was found that for subsonic inlets the results were virtually identical for different choices of α and therefore $\alpha = 1$ is used in all computations, because this results in the most stable algorithm.

In three space dimensions the treatment of a boundary face, which is a triangle, is completely analogous with the two-dimensional case. The distribution to the three nodes is given by:

$$R_1^{\text{BC}} = \alpha K_{1^*1}^+ (U_{1^*} - U_1) + \frac{1}{2} (1 - \alpha) [K_{2^*2}^+ (U_{2^*} - U_2) + K_{3^*3}^+ (U_{3^*} - U_3)] \quad (5.26)$$

$$R_2^{\text{BC}} = \alpha K_{2^*2}^+ (U_{2^*} - U_2) + \frac{1}{2} (1 - \alpha) [K_{1^*1}^+ (U_{1^*} - U_1) + K_{3^*3}^+ (U_{3^*} - U_3)] \quad (5.27)$$

$$R_3^{\text{BC}} = \alpha K_{3^*3}^+ (U_{3^*} - U_3) + \frac{1}{2} (1 - \alpha) [K_{1^*1}^+ (U_{1^*} - U_1) + K_{2^*2}^+ (U_{2^*} - U_2)] \quad (5.28)$$

Again for subsonic inlets $\alpha = 1$ is chosen.

5.2.4 Subsonic outlet

For a subsonic outlet the eigenvalue λ_1 , equation (5.17), is the only positive eigenvalue and thus one boundary condition must be imposed. The usual choice is to prescribe the static pressure and to extrapolate the density ρ and the velocity components u_i from the domain. This completely determines the states in the ghost nodes and exactly the same procedure is applied as described in the previous section.

5.2.5 Far field

The states U^* in the ghost nodes are taken equal to the far field conditions. This looks like an overspecification of the number of boundary conditions, because no distinction is made between inflow and outflow. However the replacement of K by K^+ , see equation (5.23), which can be interpreted as a multiplication with the characteristic projector $K^+ K^{-1}$, see [9], leads to ignorance of the superfluous conditions.

5.2.6 Impermeable wall

The impermeable wall boundary condition is treated in a slightly different way than what is described in section 5.2.3. The impermeability condition, $u_n = u_j N_j = 0$, should be imposed for this kind of boundary face, leading to the desired flux function:

$$(F_j(U) N_j)_{surf} = \begin{pmatrix} 0 \\ p N_j \\ 0 \end{pmatrix}, \quad (5.29)$$

where again \vec{N} is the corresponding unit normal of the inward scaled normal \vec{n} , see figure 5.2. Usually the actual flux function, $F_j(U) N_j$, does not obey the impermeability condition and therefore a discrete approximation of

$$\int_{\Gamma_l} F_j^{corr}(U) N_j d\Gamma_l, \quad (5.30)$$

is added to the discretization, where

$$F_j^{corr}(U) N_j = (F_j(U) N_j)_{surf} - F_j(U) N_j = - \begin{pmatrix} \rho u_n \\ \rho u_i u_n \\ \rho H u_n \end{pmatrix}. \quad (5.31)$$

Again the trapezium rule is used to approximate (5.30) and the distribution to nodes in two space dimensions is:

$$R_1^{BC} = \frac{1}{2} \{ \alpha F_j^{corr}(U_1) n_j + (1 - \alpha) F_j^{corr}(U_2) n_j \} \quad (5.32)$$

$$R_2^{BC} = \frac{1}{2} \{ \alpha F_j^{corr}(U_2) n_j + (1 - \alpha) F_j^{corr}(U_1) n_j \} \quad (5.33)$$

Together with equations (5.26) to (5.28) the distribution to the nodes for a three-dimensional problem is obvious. The coefficient α is taken equal to 0.75, which is a compromise between accuracy and stability.

5.2.7 Symmetry plane

Usually a symmetry plane (symmetry line in two dimensions) corresponds to a flat plane and the nodal normals are well-defined, a fact which will be used later on. The boundary conditions for such a plane are:

$$u_n = \frac{\partial}{\partial n} (\vec{u} - u_n \vec{N}) = \frac{\partial \rho}{\partial n} = \frac{\partial p}{\partial n} = 0. \quad (5.34)$$

An elegant way to impose these conditions is the introduction of a layer of finite ghost cells, see figure 5.3. The states in the ghost nodes k^* can be determined by

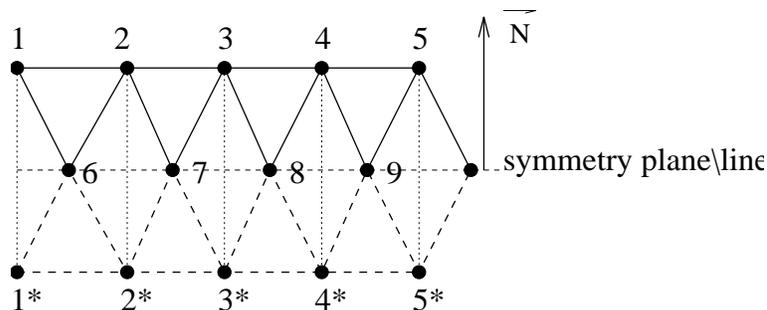


Figure 5.3: Layer of ghost cells for the symmetry plane boundary conditions.

a second-order approximation to the boundary conditions (5.34),

$$u_{n_{k^*}} = -u_{n_k}, \quad \left(\vec{u} - u_n \vec{N} \right)_{k^*} = \left(\vec{u} - u_n \vec{N} \right)_k, \quad \rho_{k^*} = \rho_k, \quad p_{k^*} = p_k, \quad (5.35)$$

and the ghost cells can be treated as if they were ordinary cells.

The implementation of the symmetry boundary conditions described above works well, but it is quite cumbersome due to the introduction of the layer of ghost cells. The situation can be drastically simplified by the following observation: suppose that the normal velocities u_n at the nodes on the symmetry plane are identically zero. Then, for reasons of symmetry, the contribution of the ghost cells to the density, tangential momentum and energy residuals of these nodes is equal to their corresponding cells on the other side of the symmetry plane and the contribution to the normal momentum residual is reversed and thus their sum is zero. By the introduction of the ghost cells, the area of the median duals, see figure 2.2, of the nodes on the symmetry plane has been doubled. Consequently in the semi-discretized equations (3.7) both the median duals S_i and the nodal residuals $\sum_T \beta_i^T R_T$ of the density, tangential momentum and energy equation are

multiplied by a factor two and effectively the same semi-discrete system is obtained as if the contributions from the ghost cells would not have been taken into account. This of course is only valid if the normal velocities on the symmetry plane are zero. Reversing this reasoning results in the conclusion that the boundary conditions (5.34) are automatically satisfied if the normal velocities u_n at the nodes on the symmetry plane are strongly enforced to be zero, which is much simpler than the procedure with the layer of ghost cells.

A special case of this type of boundary condition is the centerline of an axisymmetric problem. So the only thing necessary to do in this case is simply to require the radial velocities to be zero at the centerline nodes.

5.2.8 Implicit boundary conditions

The boundary conditions add terms to the discretization. To avoid stability problems, these terms must also be treated in an implicit manner for the backward Euler time integrator. A distinction is made between strongly and weakly enforced boundary conditions.

In the former case, (some of) the nodal values at the new time level are known and this information must be incorporated in the Jacobian matrix, see equation (4.5). Consider for example the symmetry plane. As explained in section 5.2.7, this boundary condition is satisfied if the normal velocities, $u_n = u_j N_j$, at the nodes on the symmetry plane are zero, where N_j is the component in x_j direction of the unit normal vector of the symmetry plane, see figure 5.3. Consequently the following relation holds:

$$\rho u_j^{sym} N_j = 0, \quad \forall t. \quad (5.36)$$

Assume that at the current time level equation (5.36) is satisfied. To assure that it also holds at the next time level, the updates of ρu_j are related:

$$\Delta \rho u_j^{sym} N_j = 0. \quad (5.37)$$

The entry of one of the momentum equations in the Jacobian matrix must be replaced by equation (5.37). To avoid problems, the entry corresponding to the largest component (in absolute size) of \vec{N} is selected.

A similar treatment is applied to the supersonic inlet nodes, section 5.2.1, and nodes belonging to a no-slip wall, section 6.1.

For weakly enforced boundary conditions, the extra added terms, see e.g. equations (5.24) and (5.25), must be treated implicitly. For the impermeable wall boundary condition, section 5.2.6, these terms have a relatively simple form and the determination of the derivatives is straightforward. For the subsonic inlet, subsonic outlet and far field boundary conditions, it is assumed that the differences between the states in the ghost nodes and the corresponding ordinary nodes are small, such that their product with the derivatives of $K_{l^*l}^+$, see equations (5.24) and (5.25), can be neglected. In that case, a simple expression is obtained as well.

5.3 Preconditioning

Roe *et al.*[102, 86, 77] observed that the preconditioning technique of van Leer *et al.*[131], originally developed for speeding up convergence, also results in the most decoupled form of the Euler equations and that space discretizations based on this form are superior to discretizations of the original equations. First the preconditioning in two space dimensions is explained, which leads to a conservative form of the method of characteristics for supersonic flow, followed by the extension to three dimensions.

5.3.1 The two-dimensional case

Consider the set of variables:

$$\partial Q = \begin{pmatrix} \frac{\partial p}{\rho a} \\ \partial u \\ \partial v \\ \frac{\partial p}{p} - \frac{\gamma \partial \rho}{\rho} \end{pmatrix}, \quad (5.38)$$

where the tensor notation has been dropped for reasons of clarity. The last component of ∂Q is the differential form of the unscaled entropy $s = \ln \frac{p}{\rho^\gamma}$. For this set of variables the Euler equations take a relatively simple form:

$$\frac{\partial Q}{\partial t} + \begin{pmatrix} u & a & 0 & 0 \\ a & u & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix} \frac{\partial Q}{\partial x} + \begin{pmatrix} v & 0 & a & 0 \\ 0 & v & 0 & 0 \\ a & 0 & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix} \frac{\partial Q}{\partial y} = 0. \quad (5.39)$$

Note that the last equation is fully decoupled from the others, showing that entropy is advected along streamlines. An even simpler form is obtained in the streamline coordinate system (ξ, η) , see figure 1.12:

$$\frac{\partial \tilde{Q}}{\partial t} + a \underbrace{\begin{pmatrix} M & 1 & 0 & 0 \\ 1 & M & 0 & 0 \\ 0 & 0 & M & 0 \\ 0 & 0 & 0 & M \end{pmatrix}}_{A_{\tilde{Q}}} \frac{\partial \tilde{Q}}{\partial \xi} + a \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{B_{\tilde{Q}}} \frac{\partial \tilde{Q}}{\partial \eta} = 0, \quad (5.40)$$

where $\partial \tilde{Q}$,

$$\partial \tilde{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \partial Q = \begin{pmatrix} \frac{\partial p}{\rho a} \\ \partial \tilde{u} \\ \partial \tilde{v} \\ \frac{\partial p}{p} - \frac{\gamma \partial \rho}{\rho} \end{pmatrix}, \quad (5.41)$$

is the set of ∂Q variables in the (ξ, η) system and θ is the flow direction with respect to the Cartesian coordinate system.

The idea of preconditioning is to change the transient behavior of (5.40) while keeping the same steady-state equation. To accomplish this, the spatial part of (5.40) is multiplied by a matrix P , which results in:

$$\frac{\partial \tilde{Q}}{\partial t} + P \left(A_{\tilde{Q}} \frac{\partial \tilde{Q}}{\partial \xi} + B_{\tilde{Q}} \frac{\partial \tilde{Q}}{\partial \eta} \right) = 0. \quad (5.42)$$

To assure that the character of the original system is conserved, the sign of the eigenvalues of the preconditioned Euler equations must be identical to the sign of the eigenvalues of the original system. Therefore a necessary condition is that the preconditioning matrix is positive-definite. Van Leer *et al.*[131] give the following formulation with the goal of clustering the eigenvalues of the preconditioned system:

$$P = \begin{pmatrix} \frac{\chi}{\beta^2} M^2 & \frac{-\chi}{\beta^2} M & 0 & 0 \\ \frac{-\chi}{\beta^2} M & \frac{\chi}{\beta^2} + 1 & 0 & 0 \\ 0 & 0 & \chi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.43)$$

The parameters β and χ are defined as:

$$\beta = \sqrt{\max(\epsilon^2, |M^2 - 1|)} \quad (5.44)$$

$$\chi = \frac{\beta}{\max(M, 1)}, \quad (5.45)$$

where ϵ is a small value, typically 0.05, to avoid the sonic point singularity. Introducing the "steady-state" characteristic variables $\partial\tilde{W}$,

$$\partial\tilde{W} = \begin{pmatrix} \beta & 0 & M & 0 \\ \beta & 0 & -M & 0 \\ 1 & M & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \partial\tilde{Q} = \begin{pmatrix} \beta \frac{\partial p}{\rho a} + M \partial\tilde{v} \\ \beta \frac{\partial p}{\rho a} - M \partial\tilde{v} \\ \frac{\partial p}{\rho a} + M \partial\tilde{u} \\ \frac{\partial p}{p} - \frac{\gamma \partial \rho}{\rho} \end{pmatrix}, \quad (5.46)$$

transforms the system (5.42) into:

$$\frac{\partial\tilde{W}}{\partial t} + \underbrace{\tilde{u} \begin{pmatrix} \chi\nu^+ & \chi\nu^- & 0 & 0 \\ \chi\nu^- & \chi\nu^+ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{A_{\tilde{W}}} \frac{\partial\tilde{W}}{\partial \xi} + \underbrace{\tilde{u} \begin{pmatrix} \frac{\chi}{\beta} & 0 & 0 & 0 \\ 0 & \frac{-\chi}{\beta} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{B_{\tilde{W}}} \frac{\partial\tilde{W}}{\partial \eta} = 0, \quad (5.47)$$

where ν^+ and ν^- are functions of the Mach number,

$$\nu^+ = \frac{M^2 - 1 + \beta^2}{2\beta^2}, \quad \nu^- = \frac{M^2 - 1 - \beta^2}{2\beta^2}, \quad (5.48)$$

which are bounded between $[0, 1]$ and $[-1, 0]$ respectively. Clearly the third and fourth equations are decoupled and their quantities advected along the streamline. These correspond to entropy, the fourth variable, and total enthalpy. Indeed it is straightforward to show that the following relation holds:

$$\partial H = a \partial\tilde{W}_3 + \frac{a^2}{\gamma(\gamma - 1)} \partial\tilde{W}_4, \quad (5.49)$$

where $\partial\tilde{W}_3$ and $\partial\tilde{W}_4$ are the third and fourth components of $\partial\tilde{W}$. Furthermore for supersonic flow, $M > 1$, $\nu^+ = 1$ and $\nu^- = 0$, also the first and second equations of the system (5.47) are decoupled, and the Euler equations reduce to a set of scalar advection equations, also used in the method of characteristics [66]. The advection

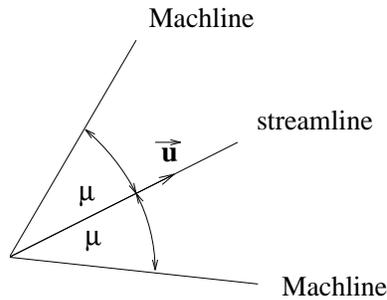


Figure 5.4: Streamline and Mach lines for a two-dimensional supersonic flow.

direction of $\partial\tilde{W}_1$ and $\partial\tilde{W}_2$ are the Mach lines, see figure 5.4, where the Mach angle μ is the angle between the streamline and the Mach lines:

$$\mu = \arctan\left(\frac{1}{\beta}\right). \quad (5.50)$$

For an isentropic (potential) flow the so-called acoustic variables, $\partial\tilde{W}_1$ and $\partial\tilde{W}_2$, can be integrated to give the well-known Prandtl-Meyer functions [66].

For a subsonic flow, $M < 1$, $\nu^+ = 0$ and $\nu^- = -1$, the acoustic variables form a Cauchy-Riemann type subsystem, which explains the partially elliptic nature of the steady, subsonic Euler equations.

As at the characteristic level the entropy and enthalpy equations are always decoupled, it is possible to use a combination of different schemes, for example the PSI-scheme for the advection of entropy and total enthalpy along the streamline and the system Lax-Wendroff scheme for the acoustic subsystem. It will be shown in section 5.4.1 that this particular combination gives the best results in terms of spurious entropy production for subcritical flows.

Once again it is stressed that, despite the fact that the characteristic form (5.47) is used to define the upwind parameters, the method is conservative and the conservative variables are updated at every time step. As explained earlier in equation (5.13), this is accomplished by the introduction of consistent nodal values of the characteristic variables,

$$\tilde{W}^{consistent} = \frac{\partial\tilde{W}}{\partial Z}(\bar{Z}) Z, \quad (5.51)$$

for the discretization of the system (5.47). The update of the conservative variables is then obtained by inverting the steps (5.39), (5.40), (5.42) and (5.47) in reversed order. These steps correspond to the transformation of the conservative variables U to the Q variables, to the transformation to the streamline frame, to the multiplication with the preconditioning matrix P and to the transformation of the \tilde{Q} variables into the characteristic variables \tilde{W} . Mathematically this means that the quasi-linear form (5.6) is written in a very special way:

$$\frac{\partial U}{\partial t} + \underbrace{\frac{\partial U}{\partial Q} \frac{\partial Q}{\partial \tilde{Q}} P^{-1} \frac{\partial \tilde{Q}}{\partial \tilde{W}}}_{\text{evaluated at } \bar{Z}} \left(\underbrace{A_{\tilde{W}} \frac{\partial \tilde{W}}{\partial \xi} + B_{\tilde{W}} \frac{\partial \tilde{W}}{\partial \eta}}_{\text{Used to define upwind parameters}} \right) = 0. \quad (5.52)$$

The transformation matrix from the ∂Q variables in the Cartesian frame to the streamline frame, $\frac{\partial Q}{\partial \tilde{Q}}$, is given by:

$$\frac{\partial Q}{\partial \tilde{Q}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (5.53)$$

which is the inverse of the matrix given in equations (5.41).

The introduction of consistent nodal values of the characteristic variables leads to an additional error compared to a direct discretization of the quasi-linear system (5.39) or even (5.47). This is a consequence of requiring conservation and is only needed if discontinuities are present. For subcritical flow, it is possible to discretize the non-conservative form, see [39].

Note that in the formulation (5.52) P^{-1} appears. The inverse of the preconditioning matrix given in equation (5.43) has a quadratic singularity at $M = 0$, and therefore stability problems can be expected when solving the system (5.52) for low Mach number flows or if stagnation points are present. This is indeed the case if an explicit time integration method is used, but the fully implicit backward Euler method avoids these problems for the Euler equations. However, for the Navier-Stokes equations this stability problem could not be overcome and the preconditioner (5.43) could not be used for viscous flows. Some viscous preconditioners exist, see [122, 37], but these are more heuristic.

5.3.2 The three-dimensional case

In three dimensions the same procedure is followed and the preconditioning matrix of van Leer *et al.*[131] in the streamline coordinate system (ξ, η, ζ) , see figure 5.5, for the set of variables $\partial\tilde{Q} = (\frac{\partial p}{\rho a}, \partial\tilde{u}, \partial\tilde{v}, \partial\tilde{w}, \partial s)^T$ is:

$$P = \begin{pmatrix} \frac{\chi}{\beta^2} M^2 & \frac{-\chi}{\beta^2} M & 0 & 0 & 0 \\ \frac{-\chi}{\beta^2} M & \frac{\chi}{\beta^2} + 1 & 0 & 0 & 0 \\ 0 & 0 & \chi & 0 & 0 \\ 0 & 0 & 0 & \chi & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.54)$$

The definitions of β and χ are identical to those in 2D, see equations (5.44) and (5.45).

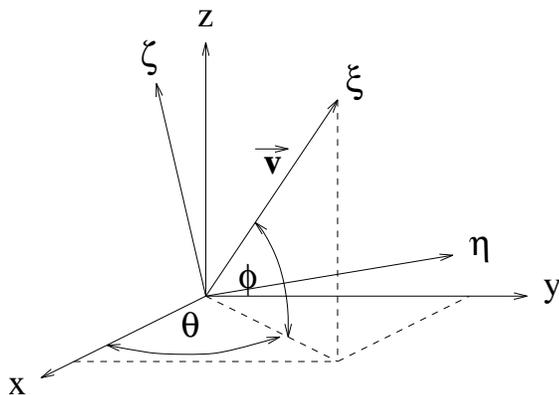


Figure 5.5: The 3D streamline coordinate system.

Introduction of the characteristic variables $\partial\tilde{W}$, see [16],

$$\partial\tilde{W} = \begin{pmatrix} \beta \frac{\partial p}{\rho a} + M \cos \psi \partial\tilde{v} + M \sin \psi \partial\tilde{w} \\ \beta \frac{\partial p}{\rho a} - M \cos \psi \partial\tilde{v} - M \sin \psi \partial\tilde{w} \\ M \sin \psi \partial\tilde{v} - M \cos \psi \partial\tilde{w} \\ \frac{\partial p}{\rho a} + M \partial\tilde{u} \\ \frac{\partial p}{p} - \frac{\gamma \partial \rho}{\rho} \end{pmatrix}, \quad (5.55)$$

where ψ is an arbitrary angle in the crossflow plane (η, ζ) , leads to the characteristic system:

$$\frac{\partial\tilde{W}}{\partial t} + A_{\tilde{W}} \frac{\partial\tilde{W}}{\partial \xi} + B_{\tilde{W}} \frac{\partial\tilde{W}}{\partial \eta} + C_{\tilde{W}} \frac{\partial\tilde{W}}{\partial \zeta} = 0, \quad (5.56)$$

$$A_{\tilde{W}} = \tilde{u} \begin{pmatrix} \chi\nu^+ & \chi\nu^- & 0 & 0 & 0 \\ \chi\nu^- & \chi\nu^+ & 0 & 0 & 0 \\ 0 & 0 & \chi & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad B_{\tilde{W}} = \tilde{u} \begin{pmatrix} \frac{\chi}{\beta} \cos \psi & 0 & \frac{\chi}{\beta} \sin \psi & 0 & 0 \\ 0 & \frac{-\chi}{\beta} \cos \psi & \frac{\chi}{\beta} \sin \psi & 0 & 0 \\ \frac{\chi}{2\beta} \sin \psi & \frac{\chi}{2\beta} \sin \psi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (5.57)$$

$$C_{\tilde{W}} = \tilde{u} \begin{pmatrix} \frac{\chi}{\beta} \sin \psi & 0 & \frac{-\chi}{\beta} \cos \psi & 0 & 0 \\ 0 & \frac{-\chi}{\beta} \sin \psi & \frac{-\chi}{\beta} \cos \psi & 0 & 0 \\ \frac{-\chi}{2\beta} \cos \psi & \frac{-\chi}{2\beta} \cos \psi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where ν^+ and ν^- are defined in equation (5.48). As in two space dimensions, total enthalpy and entropy (the fourth and fifth equations of system (5.56)) always decouple and are advected along the streamline. In contrast to the two-dimensional case, the acoustic equations never decouple and form a 3×3 subsystem, even for supersonic flow. Through the property of invariance for similarity transformations, section 3.2, the nodal residuals of the conservative variables are independent of the choice of the angle ψ . Exceptions are the scalar limited system PSI-scheme, section 3.3.2, and the system SUPG-scheme with artificial dissipation, section 3.3.6, because these schemes do not possess this property.

The 3D counterpart of the special quasi-linear form (5.52) is:

$$\frac{\partial U}{\partial t} + \underbrace{\frac{\partial U}{\partial Q} \frac{\partial Q}{\partial \tilde{Q}} P^{-1} \frac{\partial \tilde{Q}}{\partial \tilde{W}}}_{\text{evaluated at } \bar{Z}} \left(\underbrace{A_{\tilde{W}} \frac{\partial\tilde{W}}{\partial \xi} + B_{\tilde{W}} \frac{\partial\tilde{W}}{\partial \eta} + C_{\tilde{W}} \frac{\partial\tilde{W}}{\partial \zeta}}_{\text{Used to define upwind parameters}} \right) = 0, \quad (5.58)$$

where the transformation matrix $\frac{\partial Q}{\partial \tilde{Q}}$ is given by:

$$\frac{\partial Q}{\partial \tilde{Q}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos \theta \cos \phi & -\sin \theta & -\cos \theta \sin \phi & 0 \\ 0 & \sin \theta \cos \phi & \cos \theta & -\sin \theta \sin \phi & 0 \\ 0 & \sin \phi & 0 & \cos \phi & 0 \end{pmatrix}, \quad (5.59)$$

in combination with the following definitions of the angles θ and ϕ , see also figure 5.5:

$$\begin{aligned} \cos \theta &= \frac{u}{\sqrt{u^2 + v^2}} & \cos \phi &= \frac{\sqrt{u^2 + v^2}}{\sqrt{u^2 + v^2 + w^2}} \\ \sin \theta &= \frac{v}{\sqrt{u^2 + v^2}} & \sin \phi &= \frac{w}{\sqrt{u^2 + v^2 + w^2}} \end{aligned} \quad (5.60)$$

Here u , v and w are the velocity components in the Cartesian frame, where for clarity the tensor notation has not been used. Note that by this definition ϕ is bounded between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$.

5.4 Results

This section presents results for three different test cases: (1) subsonic flow over a NACA-0012 airfoil, (2) supersonic flow in a channel with a smoothly varying cross-section and (3) transonic flow over an M6-wing. The purpose of the first test case is to found out what (combination of) scheme(s) gives the most accurate result for flows without discontinuities. The channel flow has been used to demonstrate the very good shock-capturing capabilities of the discretization technique and the M6-wing shows that the extension to three space dimensions does not give any problems.

5.4.1 Subsonic flow over a NACA-0012 airfoil

This subcritical flow, $M_\infty = 0.63$, $\alpha = 2^\circ$, has been used for a detailed comparison of the results of several schemes. The grid shown in figure 5.6 has been generated with a frontal Delaunay method [82] and consists of 2355 nodes, with 140 nodes on the airfoil and 33 nodes on the far field boundary, which is located 20 chords away from the airfoil. Figure 5.7 shows a convergence history for this test case, where the preconditioned Euler equations are discretized with the system Lax-Wendroff scheme for the coupled acoustic subsystem and the scalar PSI-scheme for the entropy and total enthalpy equations. An initial solution is obtained by applying the system N-scheme directly to the Euler equations. This first-order solution itself is obtained in 9 Newton iterations ($\text{CFL}_{\max} = 10^6$), as shown in the first part of figure 5.7. This solution is compared with the first-order upwind finite-volume scheme on the median dual grid in figure 5.8, showing a clear improvement in accuracy for the N-scheme.

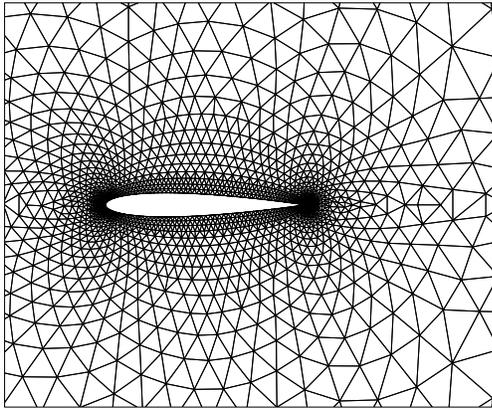


Figure 5.6: Part of the grid for the subsonic NACA-0012 airfoil, $M_\infty = 0.63$, $\alpha = 2^\circ$

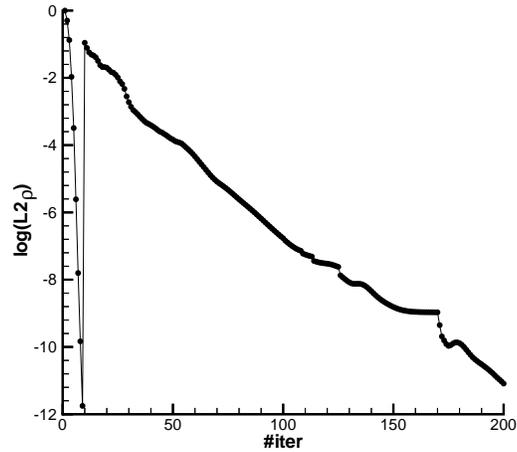


Figure 5.7: Convergence history for the nonlinear discretized preconditioned Euler equations.

To enhance stability for the second-order computation, the PSI-scheme has been replaced by the N-scheme in the implicit part. Due to this inconsistency, Newton convergence cannot be obtained and the CFL number has been restricted to a maximum value of 100. As the second-order discretization uses multiple schemes, it is difficult to compute the (approximate) analytical Jacobian and therefore the numerical version has been used. The convergence history shown in figure 5.7 is typical for nonlinear schemes. For linear schemes, both first and second-order, Newton convergence was always obtained for this test case.

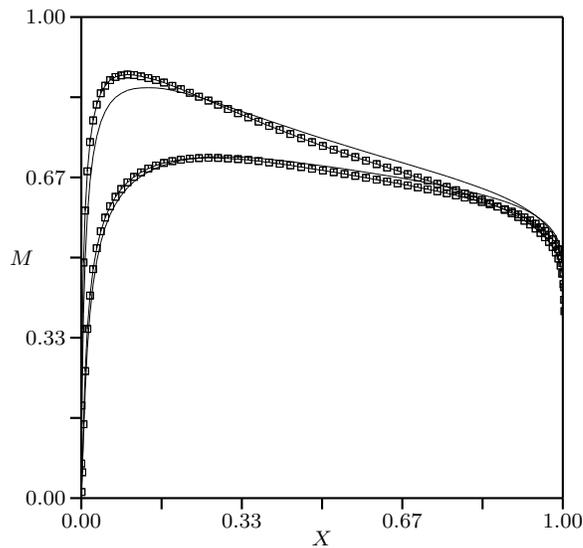


Figure 5.8: Mach number distribution on the airfoil, N-scheme (symbols indicating mesh points) and system finite-volume scheme (straight line).

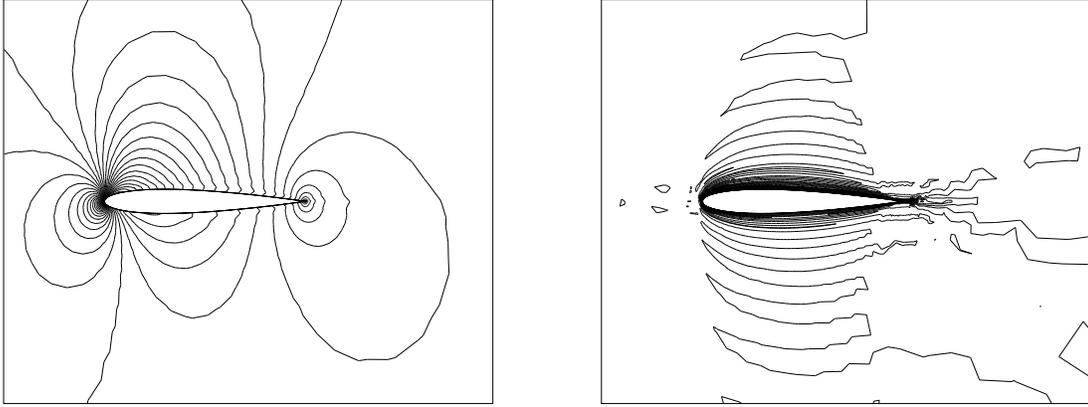


Figure 5.9: Mach number (left) and entropy (right) isolines for the Lax-Wendroff scheme directly applied to the Euler equations

The solutions of several (combinations of) schemes are compared. They have been applied either directly to the Euler equations or to the preconditioned Euler equations. Mach number and entropy (defined as $\ln(\frac{p}{\rho^\gamma}) - \ln(\frac{p_\infty}{\rho_\infty^\gamma})$) isolines for some solutions are depicted in the figures 5.9 to 5.13. In all these figures the step between the Mach number isolines is 0.02 and between the entropy isolines 0.002, so that a fair comparison is possible. The results are summarized in table 5.1, giving the minimum and maximum Mach number, the minimum and maximum entropy level and the lift and drag coefficients. Both the entropy level and the drag coefficient are zero for the exact solution. For the reference solution a relatively fine grid of 8975 nodes has been used, in combination with the Lax-Wendroff scheme for the acoustic subsystem and the scalar PSI-scheme for the advection of entropy and total enthalpy along the streamline, *i.e.* the preconditioned Euler equations have

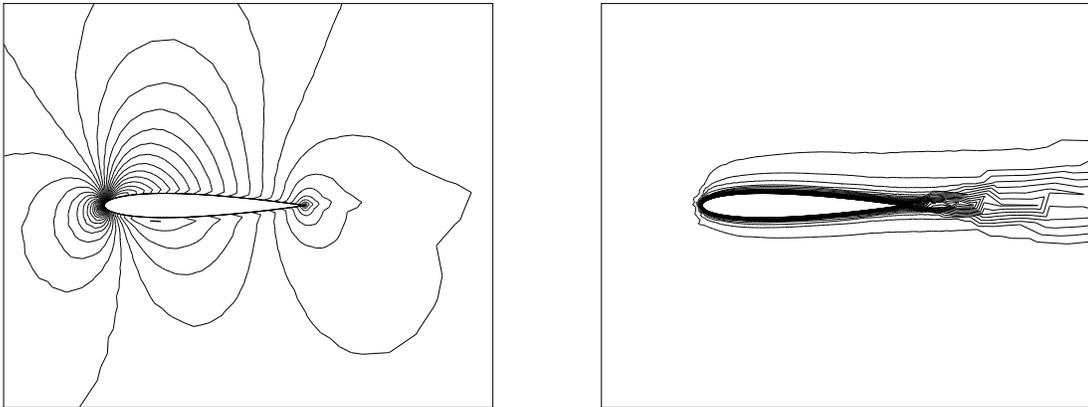


Figure 5.10: Mach number (left) and entropy (right) isolines for the PSI-scheme directly applied to the Euler equations

	M_{\min}	M_{\max}	s_{\min}	s_{\max}	C_l	C_d
Euler (N)	0.01	0.87	0.0000	0.0213	0.2756	0.0178
Euler (LDA)	0.05	0.97	-0.0076	0.0145	0.3245	0.0005
Euler (LW)	0.05	0.97	-0.0031	0.0087	0.3223	0.0008
Euler (PSI)	0.05	0.94	0.0000	0.0200	0.3085	0.0042
Prec. Euler (N)	0.02	0.88	0.0000	0.0152	0.2785	0.0182
Prec. Euler (LDA)	0.04	0.97	-0.0026	0.0055	0.3201	0.0012
Prec. Euler (LW)	0.03	0.98	-0.0014	0.0019	0.3215	0.0010
Prec. Euler (PSI)	0.03	0.95	0.0000	0.0115	0.3098	0.0042
Prec. Euler (LDA+PSI)	0.04	0.97	-0.0006	0.0039	0.3210	0.0012
Prec. Euler (LW+PSI)	0.05	0.98	-0.0006	0.0008	0.3219	0.0011
Ref. Sol. (20 chord lengths)	0.01	0.98	-0.0009	0.0001	0.3221	0.0004
Ref. Sol. (120 chord lengths)	0.01	0.99	-0.0009	0.0001	0.3314	0.0001

Table 5.1: Results for the subsonic NACA-0012 airfoil ($M_\infty = 0.63$, $\alpha = 2^\circ$).

been discretized. The second reference solution has the same number of nodes near the airfoil, but the far field location is now 120 chord lengths away from the body, instead of 20. The following conclusions can be drawn:

- the fully nonlinear PSI-scheme performs worse than the linear second-order schemes, especially Lax-Wendroff. Partially this is explained by the fact that the monotonicity requirement leads to more diffusive schemes. Another reason is that the system PSI-scheme used in this work is not entirely satisfactory yet from the theoretical point of view, see the discussion in section 7.1.1, and improvements may be expected if a better formulation is found. However it is clear that monotonic (system) schemes should only be used if they are required, and not for smooth solutions like this test case.
- for the same scheme, the discretization based on the preconditioned Euler equations is consistently more accurate than that without preconditioning. This is clear from the table, and also from the isolines plots, see for example figures 5.9 and 5.11 for the Lax-Wendroff scheme.
- for the same set of equations (preconditioned or ordinary Euler), the Lax-Wendroff scheme performs better than the LDA-scheme, see table 5.1. This can be explained by the smaller dissipation of the Lax-Wendroff scheme at the stagnation point, where the error is created. For the standard Euler equations it can be proved that at the stagnation point the system LDA-, PSI- and N-schemes are identical.
- the hybrid scheme Lax-Wendroff/PSI has improved accuracy over the pure linear Lax-Wendroff scheme and is in fact remarkably accurate ($s_{\min} = -0.0006$, $s_{\max} = 0.0008$), figure 5.13. Clearly, using a positive scheme for entropy (and enthalpy) is beneficial to remove spurious entropy oscillations.

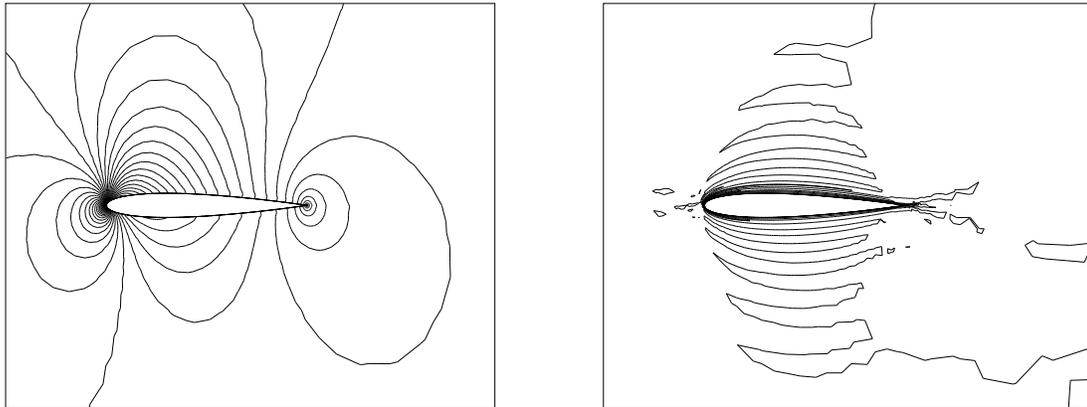


Figure 5.11: Mach number (left) and entropy (right) isolines for the Lax-Wendroff scheme applied to the preconditioned Euler equations.

- there is no clear relation between the spurious entropy production and the drag, as can be seen in table 5.1. One of the worst schemes (LDA directly on Euler, not shown in the isolines) actually has the lowest drag coefficient. This observation was also made in [35]. The explanation is the cancellation of errors in integrating the pressure over the profile.
- all solutions on the grid of figure 5.6 underpredict the lift coefficient compared to the fine grid solution with the far field at 120 chord lengths away. As the reference solution, with the far field location at 20 chord lengths, also underpredicts the lift, see table 5.1, it can be concluded that the far field is located too close to the airfoil. A possible solution would be to apply a vortex correction, but this has not been done.

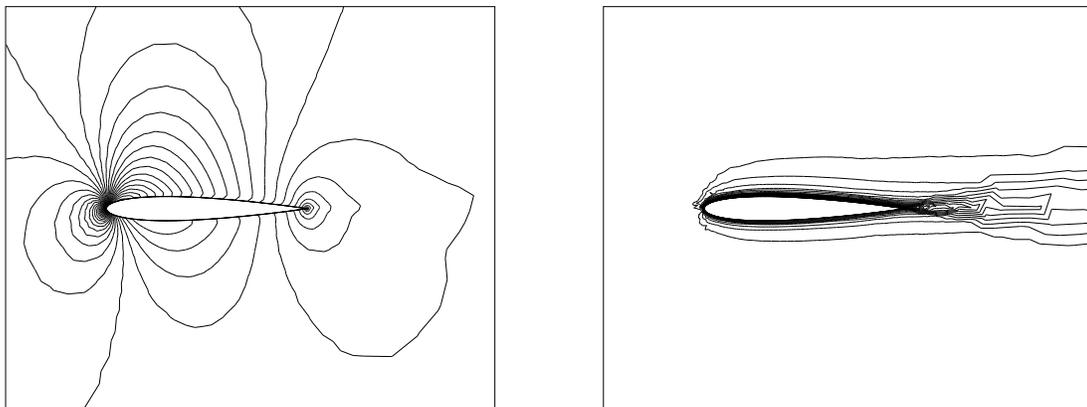


Figure 5.12: Mach number (left) and entropy (right) isolines for the PSI-scheme applied to the preconditioned Euler equations.

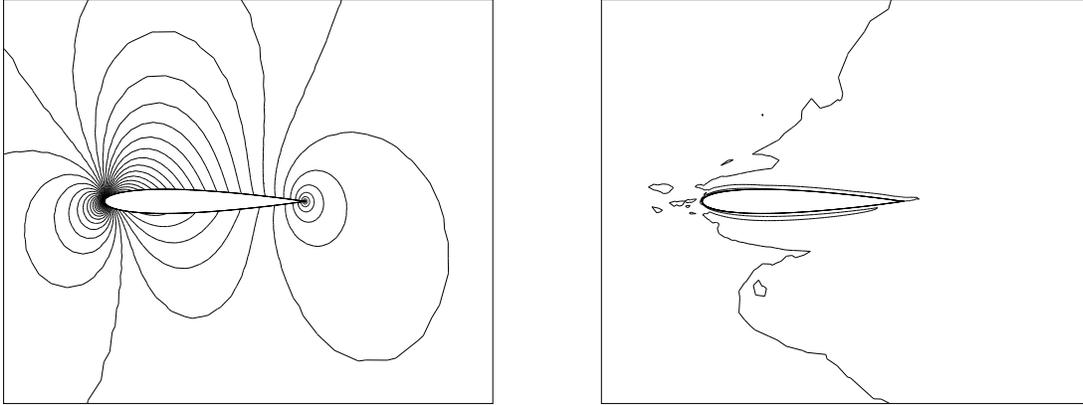


Figure 5.13: Mach number (left) and entropy (right) isolines for the mixed Lax-Wendroff/PSI-scheme applied to the preconditioned Euler equations: Lax-Wendroff on acoustic subsystem and PSI on entropy and enthalpy.

- the fully nonlinear PSI-scheme, which is the only second-order positive scheme in this set, is too dissipative in the stagnation point, figures 5.10 and 5.12. However, it is the only scheme, apart from the first-order N-scheme, that maintains positive entropy. It keeps monotone the entropy error originating at the stagnation point, and confined in a narrow layer around the profile.

5.4.2 Supersonic channel flow

The geometry of this channel with a smoothly varying cross section is defined in figure 5.14. The lower wall is formed by two quadratic polynomials, such that the wall is continuous and differentiable at the matching point $(x, y) = (1.0, 0.1)$. The inflow Mach number is 2.0 and consequently the flow is fully supersonic. The numerical solutions obtained with the multi-dimensional upwind method are compared with a method of characteristics (with constant entropy and total enthalpy) in combination with shock-fitting [123].

Three grids, denoted as coarse, medium and fine, have been created with a frontal Delaunay method [82]. They contain 6636, 26,263 and 114,990 nodes respectively. The coarse version is shown in figure 5.15. First a comparison is made

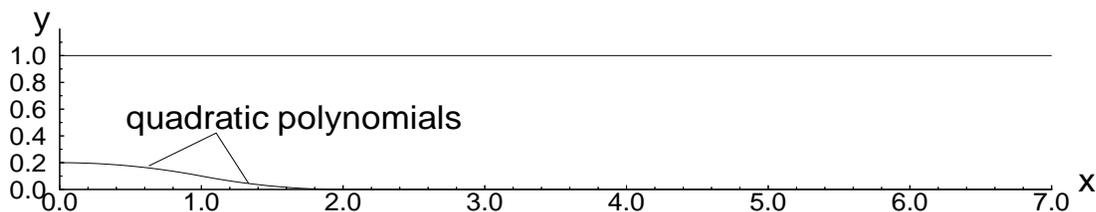


Figure 5.14: Geometry definition of the channel.

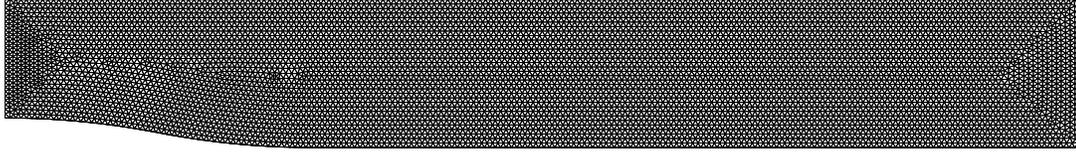
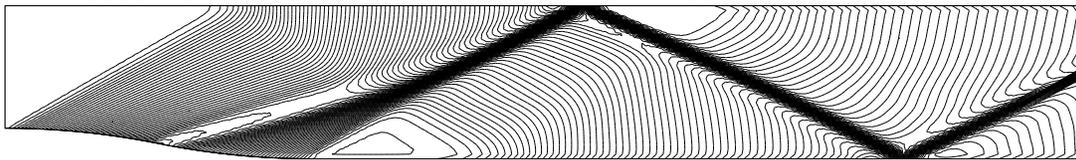


Figure 5.15: Coarse grid (6636 nodes) for the supersonic channel flow.

between the preconditioned and non-preconditioned solutions, both with the PSI-scheme on the coarse grid. The Mach number isolines are shown in figure 5.16. It is clear that the solution based on the preconditioned equations looks much smoother than the solution obtained by applying the discretization technique directly to the Euler equations. The latter even shows some wiggles. The difference between the two discretizations has two causes: (1) the preconditioning technique uses the most optimal form of the steady equations and (2) for 2D supersonic flow the characteristic form of the preconditioned equations become four decoupled scalar advection equations, while for the standard Euler equations only the entropy is decoupled and the system scheme is applied to the 3×3 subsystem. As already explained in section 3.3.2, the current version of the system PSI-scheme is certainly not optimal and has convergence problems. This is clearly visible in figure 5.17, which shows the convergence histories for both computations. Due to stability reasons, the second order scheme could not be used directly and therefore an initial solution has been computed first. For the preconditioned Euler equations this initial solution is obtained with first the N-scheme directly on the Euler equations followed by the N-scheme on the preconditioned equations. In both cases Newton convergence was obtained using numerical Jacobians. The starting CFL

Preconditioned Euler



Euler

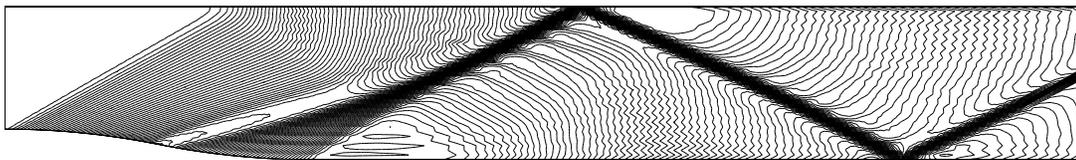


Figure 5.16: Mach number isolines on the coarse grid for the preconditioned (top) and standard (bottom) Euler equations, PSI-scheme.

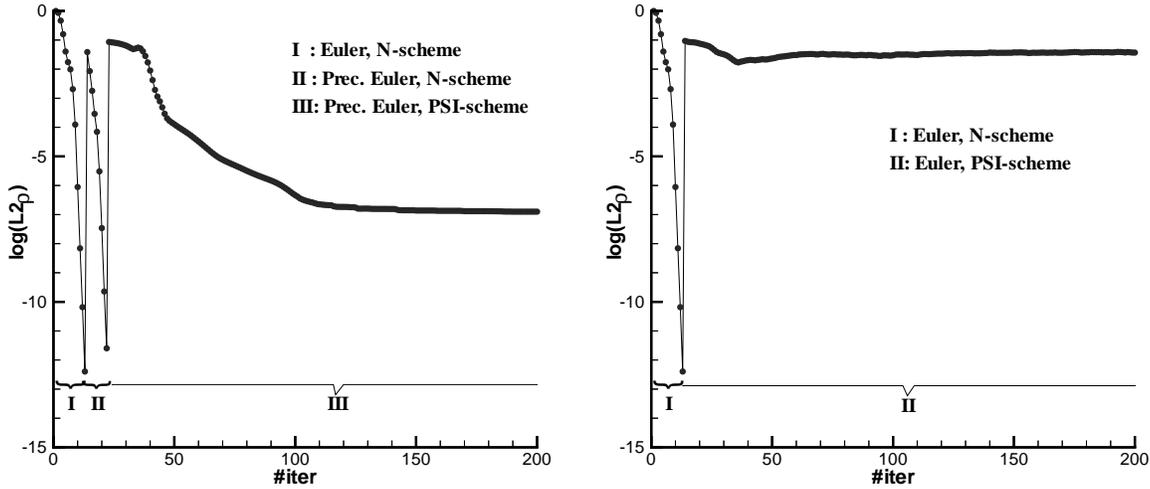


Figure 5.17: Convergence histories for the supersonic channel on the coarse grid for the preconditioned (left) and standard (right) Euler equations.

number was 1 and multiplied every time step by 10 until the maximum of 10^6 was reached. The second-order solution was computed with numerical N-scheme Jacobians with a CFL number of 100. As can be seen in the lefthand picture in figure 5.17, the nonlinear scheme converges approximately 6 orders of magnitude, relative to the restart, before it stalls.

Also for the discretization of the standard Euler equations the first-order N-scheme solution is used as initial solution for the second-order computation. The righthand picture in figure 5.17 clearly shows the convergence problems of the system PSI-scheme (with approximate N-scheme Jacobians and $CFL = 40$).

On the medium and fine grids only the solution of the preconditioned Euler equations has been computed, because on the coarse grid it has been shown to be superior to the solution of the standard Euler equations. The convergence histories (not shown) in terms of number of iterations are similar to the coarse grid, see figure 5.17, and is thus independent of the mesh size for this test case. Mach number isolines are depicted in figure 5.18, where the solutions are compared with a method of characteristics combined with shock fitting (roughly 15,000 data points) [123]. It shows that the medium and fine grid solutions are nearly identical, which can also be seen in the cross section plots, figure 5.19. Both predict a slip line, which originates from the place where the compression waves start to form the shock. The method of characteristics does not predict this slip line, because it assumes constant entropy (and also, correctly, total enthalpy). Also the coarse grid solution, figure 5.16, does not show this slip line, which indicates that this grid is not fine enough for the capturing of this phenomenon.

Mach number distributions at $y = 0.3$ and at $y = 0.7$ are shown in figure 5.19. The coarse grid solution slightly deviates in the compression fan and near the shocks, but in general it agrees well with the medium and fine grid solutions and the method of characteristics. Solution adaptivity, see e.g. [20], is probably the

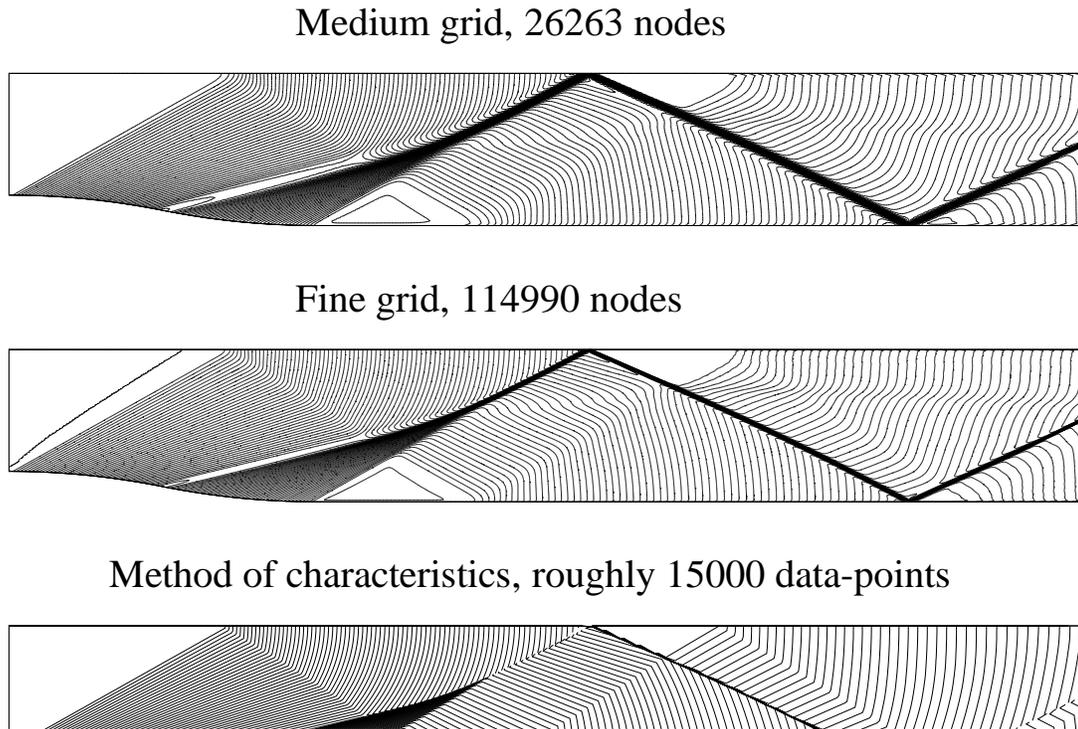


Figure 5.18: Mach number isolines on the medium (top) and fine grid (middle) for the preconditioned Euler equations and the method of characteristics (bottom).

only way to capture the previously mentioned slip line without using extremely fine grids.

Also entropy distributions at the same cross-sections have been made. These are shown in figure 5.20. The method of characteristics has been omitted in these plots, because it assumes constant entropy. It is clear that the entropy is not captured entirely monotonically, typically an overshoot occurs in one point. These cuts however should be interpreted with care, because the data points do not correspond with grid points, but are linearly interpolated between the three closest nodes. Apparently the correct prediction of the entropy is a difficult task, as, especially for the distribution at $y = 0.7$, the three solutions differ quite a bit.

The cut at $y = 0.3$ can be used to estimate the accuracy of the discretization technique. In the first part, $x < 5$, the entropy difference should be zero, because the flow experiences isentropic expansions and compressions. The entropy levels for the coarse, medium and fine grid at $x = 2.5$ (after the isentropic compression) are $14.2 \cdot 10^{-4}$, $4.0 \cdot 10^{-4}$ and $0.92 \cdot 10^{-4}$ respectively. As the spacing is approximately reduced by a factor 2 from the coarse to medium and from the medium to fine grid, it can be concluded that the discretization is second order accurate.

In order to illustrate that, for fully supersonic flow, explicit time integration can be competitive with the fully implicit method, figure 5.21 shows the conver-

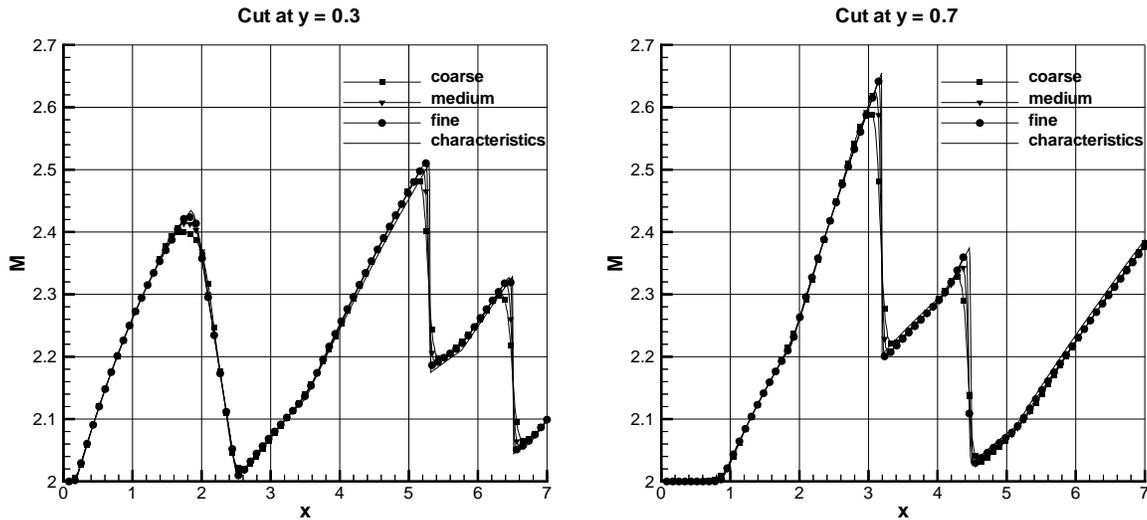


Figure 5.19: Mach number distributions at $y = 0.3$ and $y = 0.7$ for the coarse, medium and fine grid solution of the preconditioned Euler equations compared with the method of characteristics.

gence histories for the coarse grid as a function of the CPU-time on a DEC alpha workstation, equipped with a 500 MHz EV5.6 processor. For the implicit computation the CFL strategy and the strategy to obtain an initial solution has been described earlier in this section. The 4-stage Runge-Kutta time integrator, $m = 4$ in section 4.1, with $CFL = 1.5$ has been used for the explicit computation. For this method the PSI-scheme could be used immediately. It is clear from the righthand picture in figure 5.21 that also the explicit time integrator does not prevent the system PSI-scheme from stalling; on the contrary, the implicit method even con-

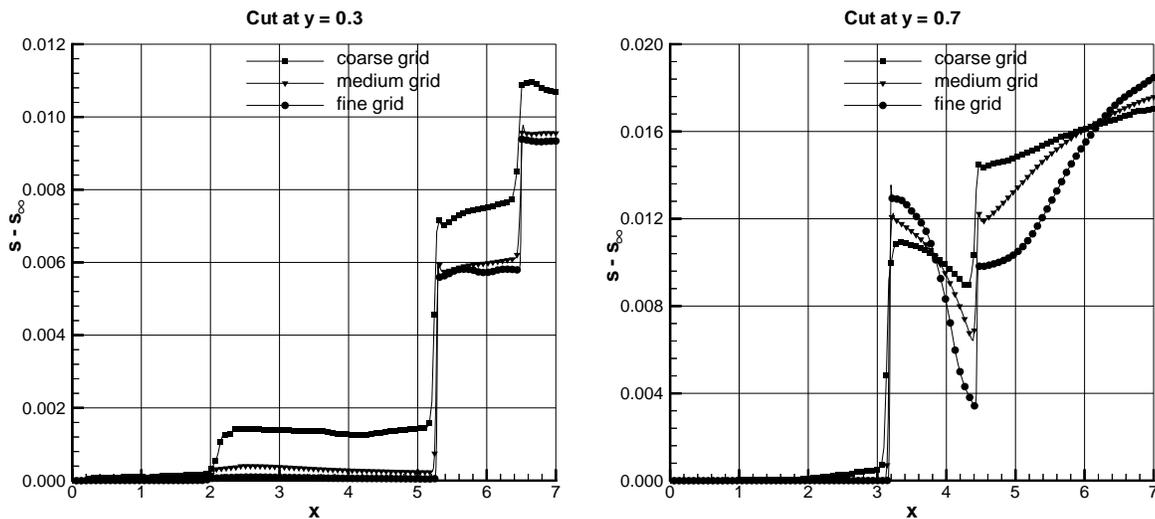


Figure 5.20: Entropy distributions at $y = 0.3$ and $y = 0.7$ for the coarse, medium and fine grid solution of the preconditioned Euler equations.

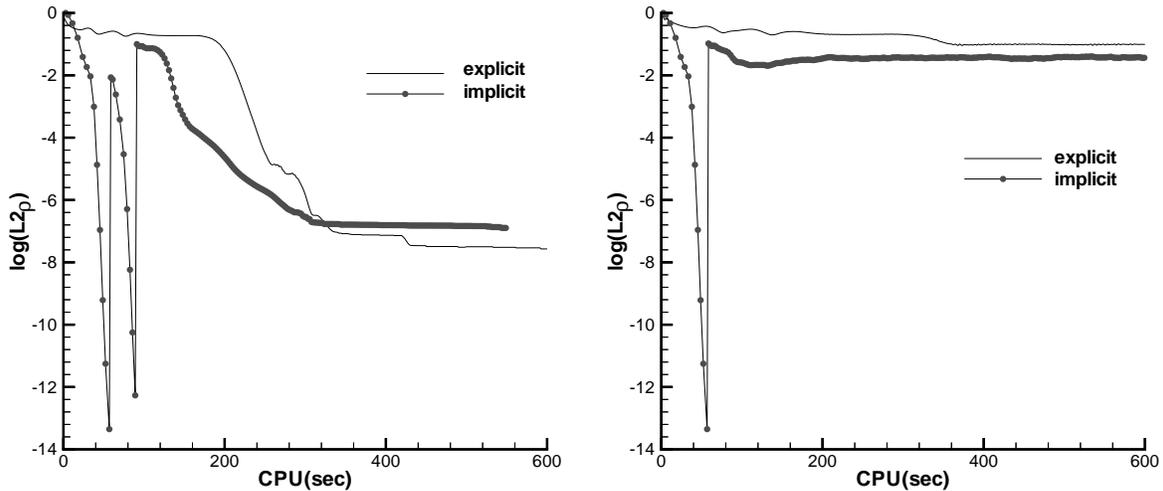


Figure 5.21: Convergence histories for the supersonic channel on the coarse grid for the preconditioned (left) and standard (right) Euler equations. Comparison between the explicit and implicit method.

verges a little better. For the preconditioned case the convergence history is quite good, but the the system PSI-scheme is not needed here, because the Euler equations are written as four scalar advection equations, see equation (5.47). As shown in the left picture of figure 5.21, both the explicit and implicit methods converge about 7 orders of magnitude in 300 seconds before they stall. The reasons why, for this test case, the explicit time integrators are competitive, are (1): explicit time integrators converge much faster for supersonic flow than for subsonic, and (2): the inability of the implicit method to obtain Newton convergence in combination with nonlinear schemes.

5.4.3 M6 wing

The ONERA M6 wing is a well documented test case for three-dimensional flows from low to transonic speeds [1]. The selected conditions, $M_\infty = 0.84$, $\alpha = 3.06^\circ$, correspond to the transonic speed regime. The grid, consisting of 273,653 nodes and 1,664,160 tetrahedra, has been created with the grid generator of the University of Swansea [90]. The surface mesh of the wing and part of the symmetry plane is shown in figure 5.22. The far field boundary is half a sphere with a radius of 12.5 root-chord lengths. Its origin is located half way the root chord.

The solution has been computed on 32 processors of the Cray T3E of the University of Delft. As can be seen in figure 5.23, Newton convergence has been obtained for the first-order N-scheme directly applied to the Euler equations, after an initial solution had been computed with 20 iterations in combination with analytical Jacobians and a CFL number of 100. These initial iterations are needed to place the shocks in a more or less correct position, so that they do not need to move any more when the numerical Jacobians are selected; moving shock waves

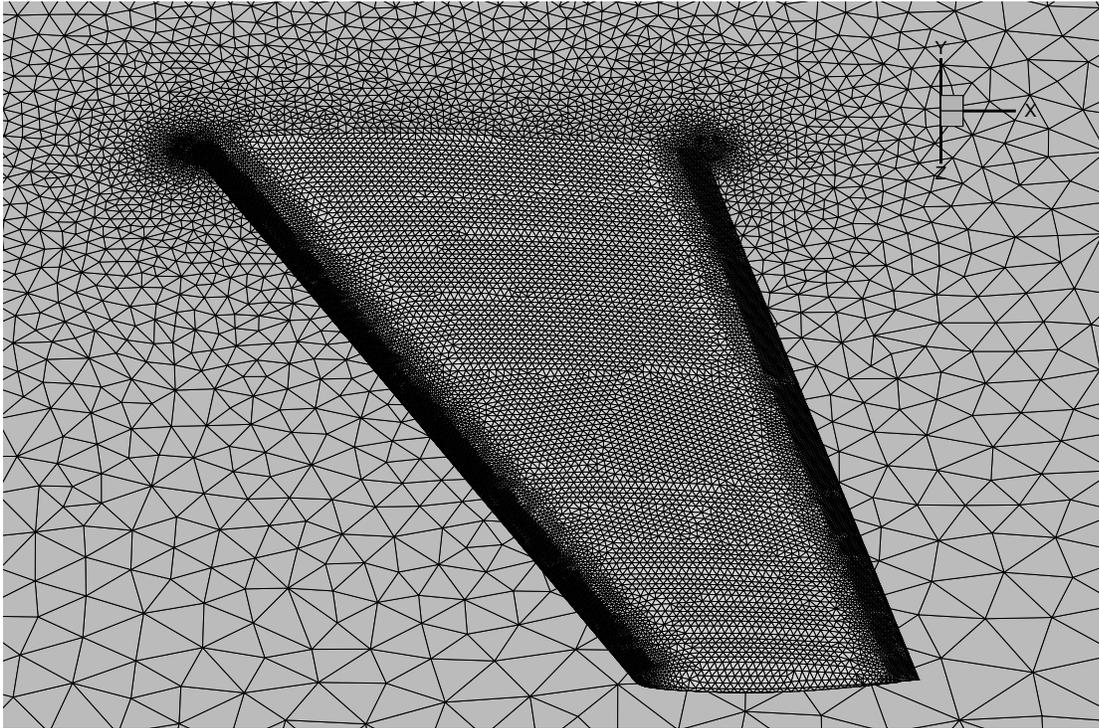


Figure 5.22: Surface grid of the M6 wing; 25371 nodes on the wing and 6101 nodes on the symmetry plane.

cause failure of the exact Newton method, unless very low CFL numbers (5-10) are used. As soon as the shocks are in position, the CFL number can be triggered to a maximum value of 10^6 in combination with numerical Jacobians.

The second-order computation, PSI-scheme applied directly to the Euler equa-

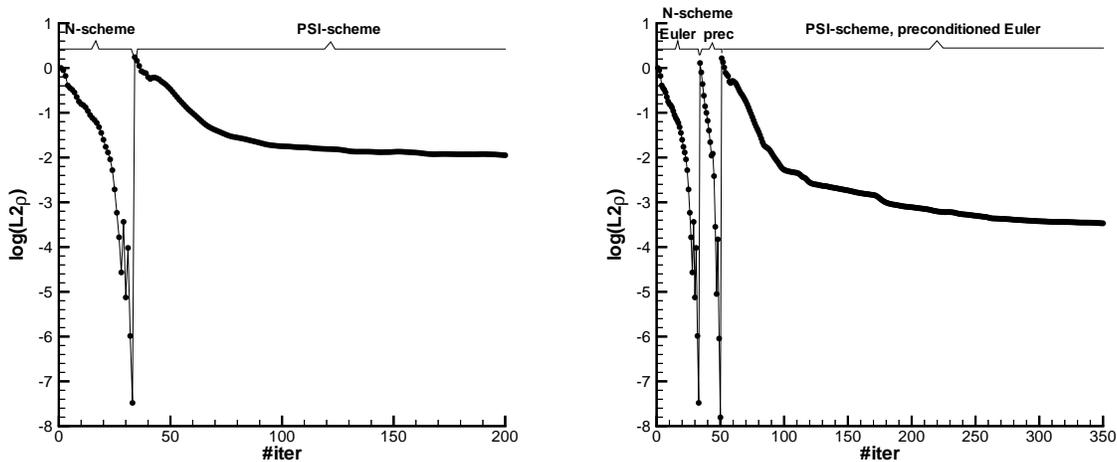


Figure 5.23: Convergence history for the M6 wing; left: Euler equations, right: preconditioned Euler equations.

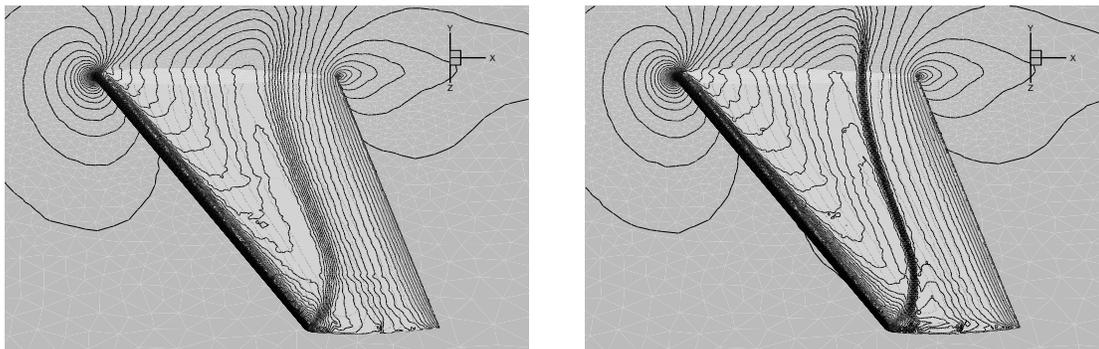


Figure 5.24: Mach number isolines for the two first-order N-scheme solutions, left: Euler equations, min = 0.17, max = 1.41, step = 0.02, right: preconditioned Euler equations, min = 0.13, max = 1.48, step = 0.02.

tions, analytical N-scheme Jacobians, CFL = 100, starts from the fully converged first-order solution and, as before, the convergence stalls after two orders of magnitude, see figure 5.23. The total CPU-time for this computation is approximately $2\frac{1}{2}$ hours.

The first-order N-scheme solution has also been used as starting solution for the preconditioned Euler equations, first in combination with the N-scheme, followed by the PSI-scheme, see the right picture of figure 5.23. Newton convergence is obtained for the preconditioned first-order solution, while again the second-order scheme stalls, but in this case only after $3\frac{1}{2}$ orders of magnitude. The total CPU-time, including both first-order computations, is roughly 8 hours.

To illustrate the beneficial effect of preconditioning, Mach number isolines of both first-order solutions are shown in figure 5.24. The shock is clearly captured much more sharply for the preconditioned Euler equations and the expansion near the tip reaches a higher value, indicating that the solution is less diffusive.

Mach number isolines on the wing upper surface and in the symmetry plane for the second-order solution of the preconditioned Euler equations are depicted in figure 5.25. A shock originates from the root leading edge and interacts with the "standard" transonic shock at approximately 85% of the span. This "leading edge" shock is not predicted by two-dimensional theory and is caused by the three-dimensional effects introduced by the 30° sweep angle of the M6 wing. Near the tip a lambda-shock structure is observed.

Experimental surface pressure distributions at several spanwise cross-sections are available [1] and figures 5.26 to 5.32 show the comparison between the experiments and the computation of the preconditioned Euler equations with the PSI-scheme. Due to the absence of shocks on the pressure side, the pressure distribution on this part of the wing is mainly determined by inviscid phenomena. Consequently, the results of the computation and experiments should be almost identical. This is indeed the case, except at the tip, where the grid, see figure 5.22, is too coarse to capture the vortex correctly.

The prediction of the location of the first shock on the suction side is rather

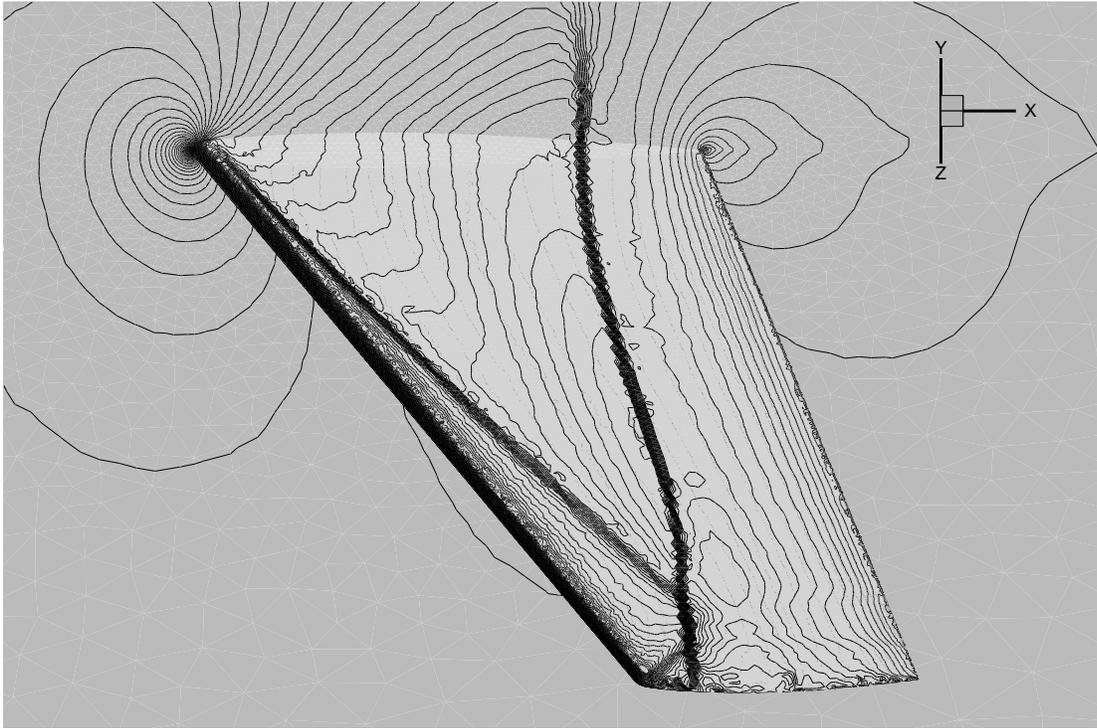


Figure 5.25: Mach number isolines on the wing upper surface and in the symmetry plane, preconditioned Euler equations, PSI-scheme, min = 0.10, max = 1.64, step = 0.02.

good, see figures 5.26, 5.27 and 5.28. Only near the position where it interacts with the second shock, the first shock is predicted too far downstream, figure 5.29.

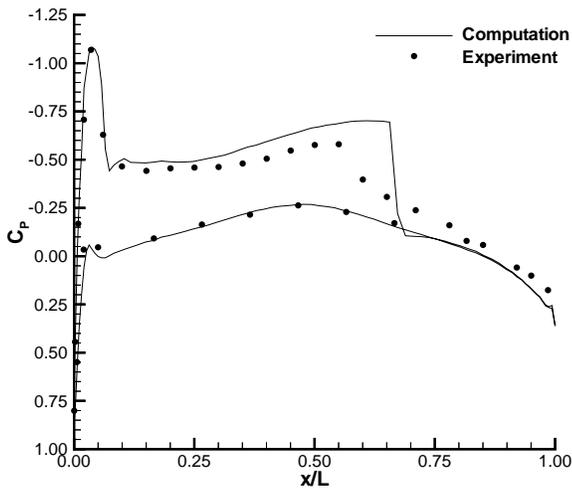


Figure 5.26: Computed and experimental pressure distribution, $z/b = 0.20$

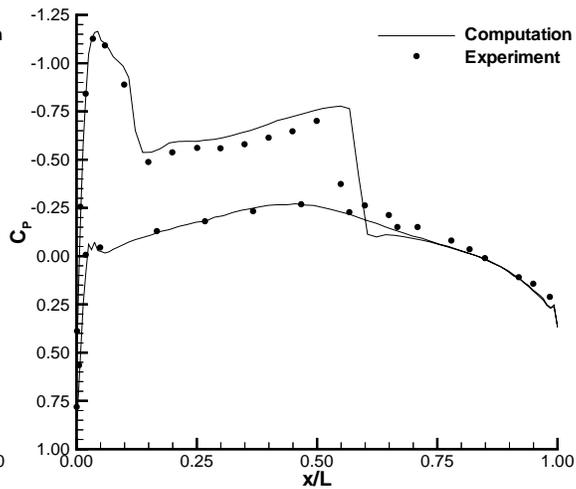


Figure 5.27: Computed and experimental pressure distribution, $z/b = 0.44$

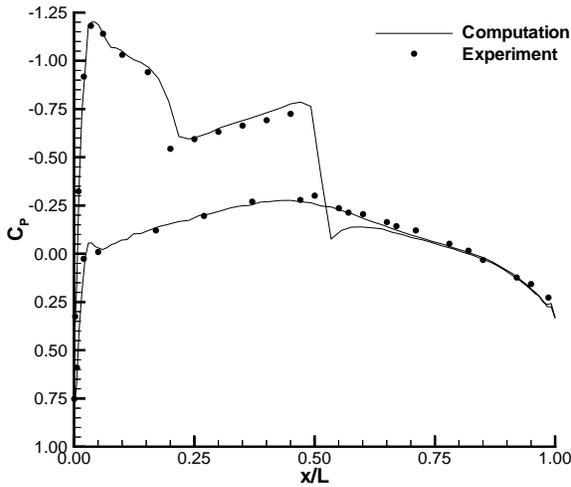


Figure 5.28: Computed and experimental pressure distribution, $z/b = 0.65$

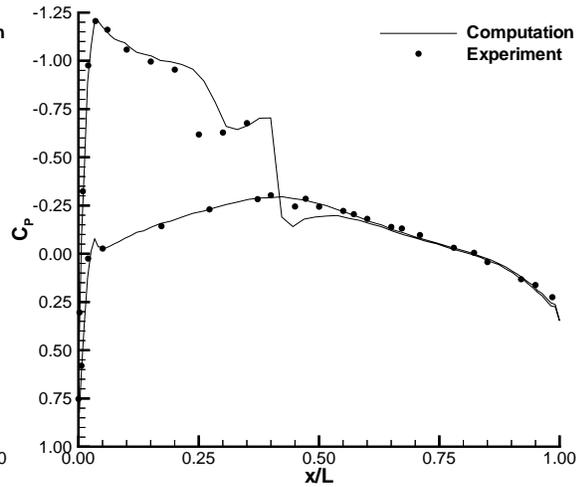


Figure 5.29: Computed and experimental pressure distribution, $z/b = 0.80$

The consequence of neglecting the viscous terms (the Reynolds number based on the mean aerodynamic chord in the experiments is $11.72 \cdot 10^6$) is that the second shock is computed too far downstream and therefore it is too strong. Consequently the pressure downstream of the shock is too high in the computation. Figures 5.26, 5.27 and 5.28 show this phenomenon particularly clearly.

Taking into account that the computation is inviscid, the agreement between computation and experiment is quite good, except at the tip, but the reason for this has already been mentioned.

To give an indication how the results of the unstructured multi-dimensional upwind discretization compare with a standard structured grid method, this test case

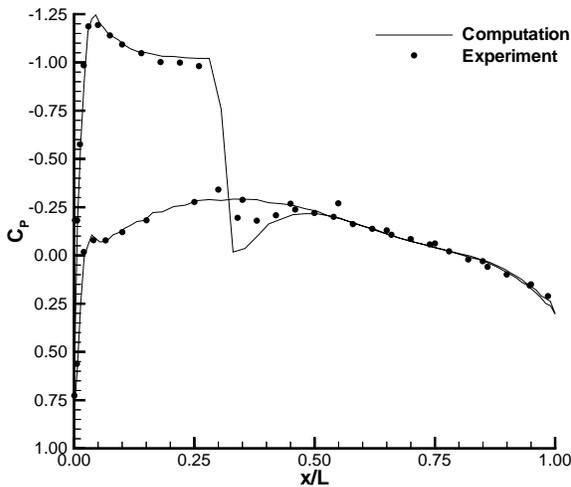


Figure 5.30: Computed and experimental pressure distribution, $z/b = 0.90$

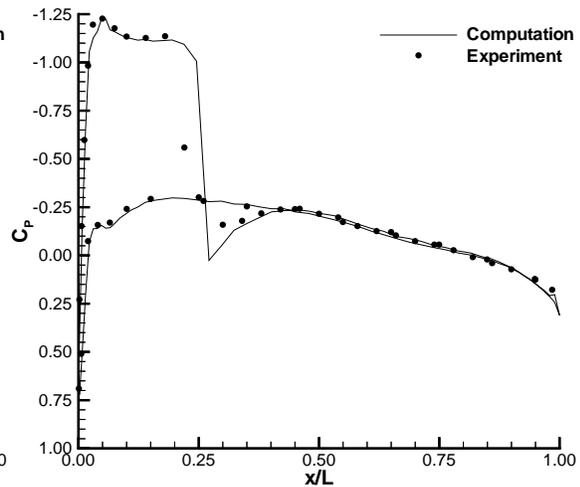


Figure 5.31: Computed and experimental pressure distribution, $z/b = 0.95$

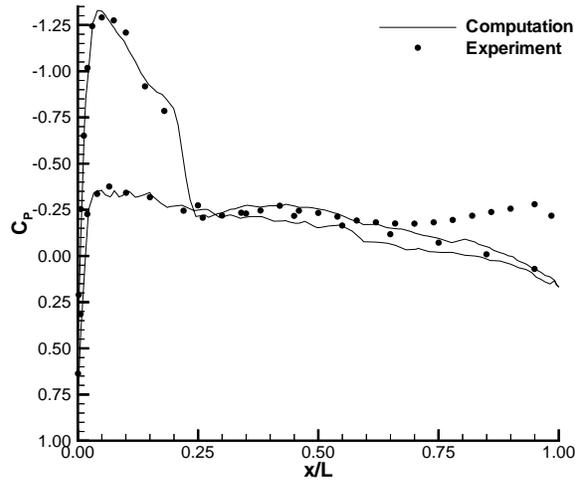


Figure 5.32: Computed and experimental pressure distribution, $z/b = 0.99$

has been computed with the DASA EUFLEX solver on a H-type grid with in total 267,960 nodes. The discretization technique used is a cell-centered finite-volume method in combination with MUSCL interpolation and a linearized Riemann solver [36].

Plots for the relative total pressure losses, $(1 - p_t/p_{t\infty})$, at four spanwise cross-sections are shown in figures 5.33 to 5.36. For the finite-volume solver the points in these figures correspond to grid points, while for the unstructured grid method these values have been linearly interpolated. This also explains the slightly wiggly behavior of the corresponding curves, compared to the finite-volume solution. As in isentropic flow the total pressure is constant, the relative loss of this quantity is a measure of the quality of the numerical solution. The negative losses predicted by the EUFLEX code near the leading edge of the wing are caused by the fact

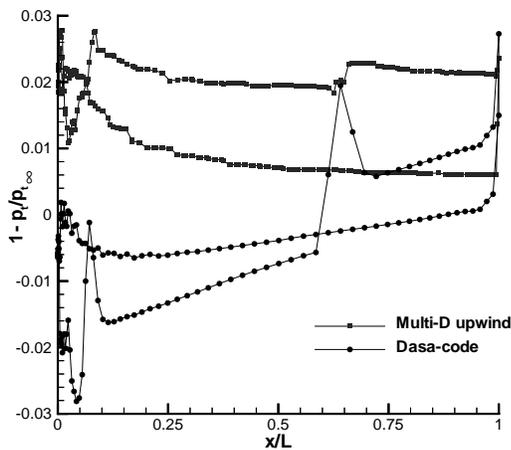


Figure 5.33: Computed relative total pressure losses, $z/b = 0.27$

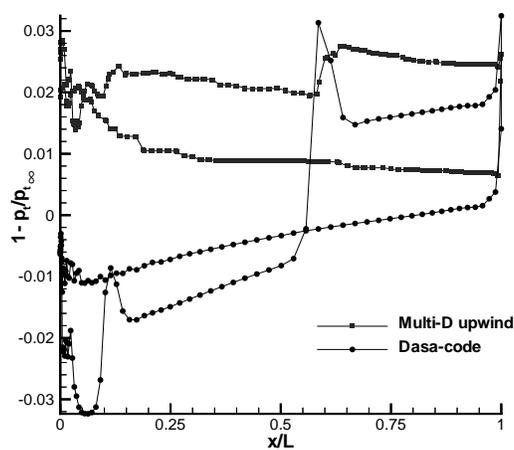


Figure 5.34: Computed relative total pressure losses, $z/b = 0.42$

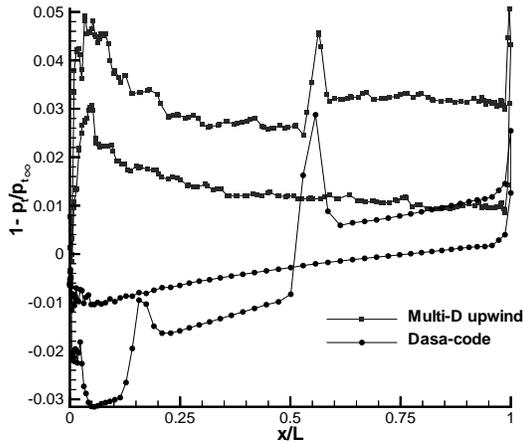


Figure 5.35: Computed relative total pressure losses, $z/b = 0.56$

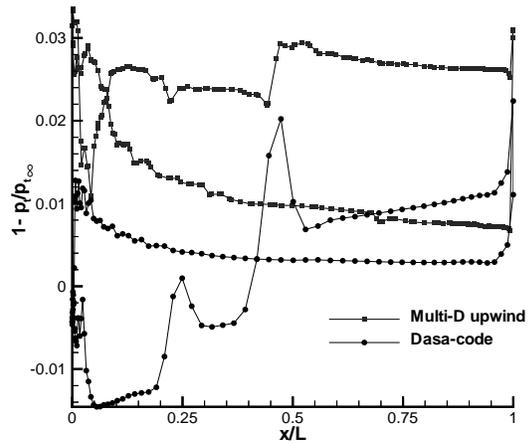


Figure 5.36: Computed relative total pressure losses, $z/b = 0.73$

that the limiter in the MUSCL reconstruction phase has been switched off in that part of the flow. The losses for the multi-dimensional upwind discretization are positive, due to the discretization with the monotonic system PSI-scheme.

From these figures it is clear that both discretization techniques have difficulties predicting the leading edge correctly. As there are no discontinuities present in this part of the flow, the total pressure loss should be zero. Grid refinement near the leading edge is the only solution to obtain better results.

After the leading edge the multi-dimensional upwind method gives slightly better results than the finite-volume method. This can especially be seen in figures 5.34 and 5.35, where the total pressure loss after the leading edge region is more constant for the multi-dimensional upwind discretization. Also the transonic shock is captured monotonically for this method, except in figure 5.35, while the finite-volume method shows an overshoot in all cross-sections.

The conclusion of this comparison is that the multi-dimensional upwind discretization shows the same deficiencies as a standard structured finite-volume method in the leading edge region of the wing, while after this region it shows improved accuracy. However, it must be said that, although they have a comparable number of nodes, the grids for the two discretization techniques are not identical and this must be kept in mind.

Chapter 6

The Navier-Stokes equations

In this chapter numerical results for the compressible Navier-Stokes equations are given. In conservation form these equations read:

$$\frac{\partial U}{\partial t} + \frac{\partial F_j}{\partial x_j} = \frac{\partial F_j^v}{\partial x_j}. \quad (6.1)$$

The conservative variables U and the inviscid flux vectors F_j are given in equation (5.2) and (5.3), while the viscous flux vectors F_j^v are defined as:

$$F_j^v = \begin{pmatrix} 0 \\ \tau_{ij} \\ u_k \tau_{jk} - q_j \end{pmatrix}. \quad (6.2)$$

Here τ is the viscous stress tensor and q the heat flux vector. The set of Navier-Stokes equations (6.1) is closed if these quantities can be written as a function of the elements of U . In this work air is considered, which is assumed to be a Newtonian fluid for which the Stokes hypothesis is valid. Then τ is a linear function of the velocity gradients:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (6.3)$$

where μ is the molecular viscosity coefficient. According to kinetic theory [23, 47], μ is only a function of temperature for monatomic gases. Although air is a mixture of mainly diatomic gases, this result is also valid for air at moderate temperatures and pressures. In all computations, the semi-empirical formula of Sutherland [134] is used:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T_0 + S}{T + S}. \quad (6.4)$$

The constants for air are:

$$\mu_0 = 1.716 \cdot 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}, \quad (6.5)$$

$$T_0=273.15K, \quad (6.6)$$

$$S=110.55K. \quad (6.7)$$

The heat flux is modeled according to Fourier's law:

$$q_j = -\kappa \frac{\partial T}{\partial x_j}, \quad (6.8)$$

where the thermal conductivity coefficient κ is related to μ and the specific heat capacity at constant pressure, c_p , by the non-dimensional Prandtl number:

$$Pr = \frac{\mu c_p}{\kappa}. \quad (6.9)$$

For air the Prandtl number is approximately constant for temperatures between 200 and 600 K and equal to 0.72. This approximation is used in all the computations presented in this chapter.

To be sure that all variables are approximately $O(1)$ and thus reducing the errors due to the finite precision of computers (and also decreasing the condition number of the linear systems in the implicit time integration method), the governing equations (6.1) are solved in their non-dimensional form. For this purpose the following reference quantities are introduced:

- reference density: ρ_∞
- reference velocity: U_∞
- reference length: L_{ref}
- reference temperature: T_∞
- reference viscosity: μ_∞

The subscript ∞ indicates the freestream value. These reference conditions directly define the non-dimensional density $\bar{\rho}$, velocities \bar{u}_i , coordinates \bar{x}_i , temperature \bar{T} and viscosity $\bar{\mu}$. Other desired non-dimensional quantities are:

$$\bar{t} = \frac{U_\infty t}{L_{ref}}, \quad \bar{p} = \frac{p}{\rho U_\infty^2}, \quad \bar{E} = \frac{E}{U_\infty^2}, \quad \bar{H} = \frac{H}{U_\infty^2}, \quad \bar{a} = \frac{a}{U_\infty}. \quad (6.10)$$

If these expressions are substituted into the Navier-Stokes equations (6.1), its non-dimensional equivalent is obtained. The state vector and inviscid fluxes are form-identical to their dimensional counterparts, but the viscous terms have been slightly changed:

$$\bar{\tau}_{ij} = \frac{1}{Re_\infty} \bar{\mu} \left(\frac{\partial \bar{u}_i}{\partial \bar{x}_j} + \frac{\partial \bar{u}_j}{\partial \bar{x}_i} - \frac{2}{3} \frac{\partial \bar{u}_k}{\partial \bar{x}_k} \delta_{ij} \right), \quad (6.11)$$

$$\bar{q}_j = -\frac{1}{Re_\infty} \frac{\bar{\mu}}{(\gamma - 1)Pr} \frac{\partial \bar{a}^2}{\partial \bar{x}_j}, \quad (6.12)$$

where Re_∞ is the freestream Reynolds number:

$$Re_\infty = \frac{\rho_\infty U_\infty L_{ref}}{\mu_\infty}. \quad (6.13)$$

In the following the bars to indicate non-dimensional quantities are omitted and all variables are assumed to be dimensionless.

In section 2.5 it was shown that for the advection-diffusion equation a Galerkin discretization of the viscous terms is consistent with the multi-dimensional upwind schemes for the advective part. A similar analysis holds for the viscous fluxes of the Navier-Stokes equations and therefore it will not be repeated here. The only issue left to be discussed for the laminar Navier-Stokes equations is the treatment of the walls with no-slip boundary conditions, which is given in section 6.1. For the other types of boundary conditions it is assumed that the viscous terms can be neglected and the Euler boundary conditions, section 5.2, are applied.

However most aeronautical flows are turbulent, which complicates the problem significantly. In principle the turbulent flow field is a solution of the Navier-Stokes equations (6.1), but the number of grid points needed to resolve properly all turbulent length scales for high Reynolds number flows is beyond current and foreseeable computer resources. Again, consider the airliner of section 1.2 in cruise condition with $Re_\infty = 10^7$. The ratio between the large turbulent scale and the Kolmogorov micro scale is $Re^{\frac{3}{4}}$ [121]. Consequently $O(Re^{\frac{3}{4}})$ points are needed per dimension to resolve all length scales. As turbulence is essential three-dimensional, at least $O(Re^{\frac{9}{4}})$ grid points are needed. For the Reynolds number of the airliner this gives $O(10^{16})$ grid points, far beyond the resources of current and near-future computers.

The approach described above is called Direct Numerical Simulation (DNS) of the turbulence and gives all the details of the flow field. Engineers do not need all this information and one can try to model the turbulence. A possibility is to divide the turbulence length scales in two categories: (1) small scales which cannot be resolved and must be modeled and, (2): large scales which can be resolved. This approach is Large Eddy Simulation (LES) and for the airliner of this example it needs $O(10^9)$ grid points. Again this number is too large for current computers, but it might come within the range of near-future (10-20 years) supercomputers [53].

Nowadays the only practical possibility for computing such a high Reynolds number turbulent flow is to model the influence of the turbulence on the mean flow. This is Reynolds averaging (or better, for compressible flows, Favre averaging [38]) and the resulting set of equations are called the Reynolds Average Navier-Stokes (RANS) equations. These equations are identical to (6.1) but the laminar viscous stress tensor τ in equation (6.2) is replaced by the sum of τ and the turbulent or Reynolds stress tensor σ and similarly for the heat flux. The Reynolds stresses must be modeled by a so-called turbulence model. One can distinguish zero-equation or algebraic models, e.g. Baldwin-Lomax [7], one-equation models, which solve a transport equation for the eddy-viscosity (to be defined later), e.g.

Baldwin-Barth [6] or Spalart-Allmaras [113], two-equation models, which model the eddy-viscosity by two additional transport equations and second-order closure models, which solve a transport equation for every element of the Reynolds stress tensor σ . The zero-equation models are non-local in nature, typically information along a line crossing the boundary layer is needed, and therefore they are difficult to implement on unstructured grids. For this reason and the fact that their results are not as good as the results of one- and two-equation models they are not used in this work. The second-order closure models are very much an area of ongoing research [132] and will therefore not be considered either. Section 6.2 describes the four models used for the computations.

The last section of this chapter contains results of both laminar and turbulent test cases.

6.1 Walls with no-slip boundary conditions

For these walls the no-slip boundary condition $u_i = 0$ must be prescribed in combination with a boundary condition for the temperature, either an isothermal ($T = T_{wall}$) or an adiabatic wall ($\frac{\partial T}{\partial n} = 0$). The no-slip and isothermal wall conditions are enforced strongly, *i.e.* every node belonging to a wall exactly obeys the imposed conditions. When explicit time integration is used, this can be done after every time step. However, for an implicit method this is not the correct approach, because this corresponds with an explicit treatment of the boundary conditions, see section 5.2.8. Therefore in the Jacobian matrix $\frac{\partial R}{\partial U}$ the rows corresponding with the momentum updates of no-slip walls are replaced by the relation:

$$\Delta \rho u_i^{\text{no-slip wall}} = 0. \quad (6.14)$$

For the isothermal wall condition the implicit treatment is more complicated. T_{wall} and thus E_{wall} is prescribed, but not zero and therefore the conservative variable ρE changes. Consequently the rows of $\frac{\partial R}{\partial U}$ corresponding with the energy equation of no-slip wall nodes are replaced by:

$$\Delta (\rho E)_i^{\text{no-slip wall}} = E_{i,wall} \Delta \rho_i^{\text{no-slip wall}}, \quad (6.15)$$

which directly links the update of ρE to the update of ρ .

The adiabatic wall boundary condition $\frac{\partial T}{\partial n} = 0$ is a homogeneous Neumann boundary condition, which is automatically satisfied within the finite-element framework by omitting the boundary integral in the partial integration of the viscous terms. Therefore nothing additional needs to be done for this boundary condition.

6.2 Turbulence models

The turbulence models used in this work are either one-equation or two-equation models, which refers to number of additional transport equations added to the

Navier-Stokes equations to model the turbulent flow field. The generic form of these additional equations is:

$$\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\nu_\phi \frac{\partial \phi}{\partial x_i} \right) + \Sigma^+ + \Sigma^-. \quad (6.16)$$

Here ϕ stands for any of the turbulent quantities, ν_ϕ is a diffusion coefficient corresponding to ϕ , Σ^+ the production source term and Σ^- the destruction source term. Note that the turbulent variables are always advected along the streamlines. First it is explained how this type of equation is discretized, followed by the models used in the computations.

6.2.1 Discretization of the turbulent transport equations

Equation (6.16) is an advection-diffusion equation with source terms. The discretization of the advection-diffusion part has already been treated in chapter 2 and will therefore not be repeated here. As it is essential that the turbulent quantity ϕ remains positive, the advective part must be discretized with a positive scheme. In practice this means that either the N- or PSI-scheme is used.

In principle the Petrov-Galerkin interpretation of the schemes, see section 2.1, leads to a straightforward discretization of the source terms Σ^+ and Σ^- . If these terms are assumed to be constant per control volume, the distribution to node l of cell T is:

$$R_{l,source} = \beta_l S_T (\Sigma^+ + \Sigma^-), \quad (6.17)$$

where S_T is the area/volume of the triangle/tetrahedron and β_l the distribution coefficient of the advection scheme. The distribution (6.17) shows that the source term must be upwinded (in practice upwind schemes are used) to have a consistent discretization. Numerical experiments however showed that this approach was not very stable and a point wise discretization of the source terms has been used. This leads to the following semi-discrete version of equation (6.16) at node l :

$$S_l \frac{d\phi_l}{dt} + \sum_{T \in \Omega_l} (R_{l,inv}^T + R_{l,vis}^T) = S_l (\Sigma_l^+ + \Sigma_l^-). \quad (6.18)$$

Here S_l is the area of the median dual cell of node l , see figure 2.2, Ω_l indicates all triangles which belong to neighborhood of node l , see figure 2.3, and $R_{l,inv}^T$ and $R_{l,vis}^T$ are the result of the discretization of the advection and the diffusion term over each cell respectively. Although formally first-order accurate, a study on a model equation showed that the reduction in accuracy due to this inconsistent source term discretization is negligible. If Σ_l^+ or Σ_l^- contain gradients, these are reconstructed with the Galerkin scheme:

$$\nabla \vartheta_l = \frac{1}{S_l} \sum_{T \in \Omega_l} \frac{S_T}{d+1} \nabla \vartheta^T, \quad (6.19)$$

where $\nabla\vartheta_l$ is the gradient of a certain quantity ϑ in node l , $\nabla\vartheta^T$ the constant gradient in cell T and d the number of spatial dimensions.

In general the source terms Σ^+ and Σ^- are stiff, which makes explicit time integration impossible and some kind of implicit algorithm must be used. To increase stability, Patankar [89] proposed to treat only the destruction source term Σ^- in an implicit manner. Usually Σ^- does not contain derivatives and is only a function of the nodal state U_l . Consequently an implicit treatment of Σ_l^- in combination with the explicit forward Euler time integrator, $m = 1$ in section 4.1, for the other terms leads to the following point-implicit method:

$$\left(\frac{1}{\Delta t_l} - \frac{\partial \Sigma_l^-}{\partial \phi_l} \right) (\phi_l^{n+1} - \phi_l^n) = R_l. \quad (6.20)$$

As $\frac{\partial \Sigma_l^-}{\partial \phi_l}$ is negative, the update is under-relaxed compared to the fully explicit algorithm. This makes the use of semi-explicit time integrators even less attractive for turbulent flows than for laminar ones and consequently implicit methods are considered.

For the fully implicit backward Euler method, section 4.2, the Jacobian matrices of the entire residual R_l must be computed. As the turbulent quantities are advected along the streamline, there is a coupling with the Navier-Stokes equations. This coupling however is neglected for two reasons, (1): due to the explicit treatment of the production source term Σ_l^+ , Newton convergence cannot be expected anyway and therefore the relatively weak coupling with the Navier-Stokes equations does not have much influence, and (2): it was found, see also e.g. [74], that neglecting these terms was beneficial for stability. For two-equation models even the coupling between the two turbulent quantities is neglected. An extra advantage of this approach is a saving of a factor two in memory (for two-equation models), which especially in 3D is not negligible.

6.2.2 The Spalart-Allmaras model

The Spalart-Allmaras model [113] is an empirical one-equation model, especially developed for aerodynamic flows. It employs a transport equation for $\tilde{\nu}$, which is a variable related to the kinematic eddy-viscosity ν_t . This eddy-viscosity is used to model the turbulent stress tensor σ :

$$\sigma_{ij} = \frac{1}{Re_\infty} \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (6.21)$$

with $\mu_t = \rho \nu_t$. Note that the non-dimensional form of σ is used, so μ_t (and also ν_t) is the ratio of the eddy-viscosity and freestream viscosity. The turbulent heat flux is modeled as:

$$q_{tj} = - \frac{1}{Re_\infty} \frac{\mu_t}{(\gamma - 1) Pr_t} \frac{\partial a^2}{\partial x_j}, \quad (6.22)$$

where the turbulent Prandtl number Pr_t is taken constant and equal to 0.90. The non-dimensional form of the Spalart-Allmaras model reads:

$$\underbrace{\frac{\partial \tilde{\nu}}{\partial t}}_I + \underbrace{u_i \frac{\partial \tilde{\nu}}{\partial x_i}}_{II} = \underbrace{\frac{1}{Re_\infty} \frac{1}{\sigma} \left[\frac{\partial}{\partial x_i} \left((\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_i} \right) + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_i} \frac{\partial \tilde{\nu}}{\partial x_i} \right]}_{III} + \underbrace{c_{b1} (1 - f_{t2}) \tilde{S} \tilde{\nu}}_{IV} - \underbrace{\frac{1}{Re_\infty} \left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2} \right) \left(\frac{\tilde{\nu}}{D} \right)^2}_V + \underbrace{Re_\infty f_{t1} \Delta U^2}_{VI}. \quad (6.23)$$

Term I is the time-derivative, which forms together with the advective part II the Lagrangian derivative. The diffusion term III is partially auto-diffusive, *i.e.* the diffusion coefficient appearing in term III is $\tilde{\nu}$ itself. The production term is given by IV , where

$$\tilde{S} = |\Omega| + \frac{1}{Re_\infty} \frac{\tilde{\nu}}{\kappa^2 D^2} f_{v2}. \quad (6.24)$$

$|\Omega|$ is the magnitude of the vorticity and D the distance to the nearest wall. Term V is the destruction term and VI the so-called trip term, which triggers the transition. ΔU is the velocity difference with respect to the nearest trip point, see later. The kinematic eddy-viscosity ν_t is given by:

$$\nu_t = \tilde{\nu} f_{v1}. \quad (6.25)$$

The definition of the model is completed by the functions:

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{\frac{1}{6}}, \quad (6.26)$$

$$g = r + c_{w2} (r^6 - r), \quad (6.27)$$

$$r = \min \left(\frac{\tilde{\nu}}{Re_\infty \tilde{S} \kappa^2 D^2}, 10 \right), \quad (6.28)$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}, \quad (6.29)$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad (6.30)$$

$$f_{t1} = c_{t1} g_t \exp \left(-c_{t2} \frac{\Omega_t^2}{\Delta U^2} [D^2 + g_t^2 D_t^2] \right), \quad (6.31)$$

$$g_t = \min \left(0.1, \frac{\Delta U}{|\Omega_t| \Delta x_t} \right), \quad (6.32)$$

$$f_{t2} = c_{t3} \exp \left(-c_{t4} \chi^2 \right). \quad (6.33)$$

The function f_{t2} is designed to keep a laminar solution laminar such that the model does not trigger transition to turbulence itself. Consequently, the trip points (or lines in 3D), *i.e.* the points where the flow becomes turbulent, must be specified by the user and the function f_{t1} , equation (6.32), starts the transition. In this function $|\Omega_t|$ is the magnitude of the vorticity at the nearest trip point and D_t the distance to this point. The tangential grid spacing at the trip point Δx_t , equation (6.32), is used for numerical reasons, see [113]. The Spalart-Allmaras model is closed by the constants:

$$\begin{aligned} c_{b1} &= 0.1355, & c_{b2} &= 0.622, & \sigma &= \frac{2}{3} & \kappa &= 0.41, \\ c_{w1} &= \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, & c_{w2} &= 0.3, & c_{w3} &= 2.0, & c_{v1} &= 7.1, \\ c_{t1} &= 1.0, & c_{t2} &= 2.0, & c_{t3} &= 1.2, & c_{t4} &= 0.5. \end{aligned} \quad (6.34)$$

It is possible that for some cases the constants of the c_t -series must be changed to enforce transition. Spalart and Allmaras [113] recommend a maximum freestream value of $\tilde{\nu}$ of 0.1. The boundary condition on a no-slip wall is simply $\tilde{\nu} = 0$.

The full model requires the knowledge of the transition points, either by an educated guess or by knowing the experimental trip points. However it is possible to use the model in "full turbulent mode" by simply setting $f_{t1} = f_{t2} = 0$. In this case the transition takes place due to numerical reasons and it is hoped that this is close enough to, for example, the leading edge of the airfoil. If this does not happen, the prediction of the transition point should not be trusted and a separate prediction model must be used, see e.g. [115].

6.2.3 The $k - \omega$ model

Experience indicates that, unlike many other two-equation models, the $k - \omega$ model [135] does not require damping functions in the viscous sublayer. Therefore, it is numerically easier to handle than for example the low-Reynolds version of the $k - \varepsilon$ model [64]. The two additional transport equations for k , the turbulent kinetic energy, and ω , its specific dissipation rate, written in non-dimensional (k is made dimensionless by U_∞^2 and ω by $\frac{L}{U_\infty}$) conservation form are given by:

$$\underbrace{\frac{\partial \rho k}{\partial t}}_I + \underbrace{\frac{\partial \rho k u_i}{\partial x_i}}_{II} = \underbrace{\frac{1}{Re_\infty} \frac{\partial}{\partial x_i} \left[(\mu + \sigma_{k1} \mu_t) \frac{\partial k}{\partial x_i} \right]}_{III} + \underbrace{\sigma_{ij} \frac{\partial u_i}{\partial x_j}}_{IV} - \underbrace{\beta^* \rho \omega k}_V \quad (6.35)$$

$$\underbrace{\frac{\partial \rho \omega}{\partial t}}_I + \underbrace{\frac{\partial \rho \omega u_i}{\partial x_i}}_{II} = \underbrace{\frac{1}{Re_\infty} \frac{\partial}{\partial x_i} \left[(\mu + \sigma_{\omega 1} \mu_t) \frac{\partial \omega}{\partial x_i} \right]}_{III} + \underbrace{Re_\infty \frac{\gamma_1 \rho}{\mu_t} \sigma_{ij} \frac{\partial u_i}{\partial x_j}}_{IV} - \underbrace{\beta_1 \rho \omega^2}_V \quad (6.36)$$

Here the terms I indicate the time-derivatives, II the advection parts, III the diffusion terms, IV the production terms and V the destruction terms. Compared

with the Reynolds stress tensor for the Spalart-Allmaras model, equation (6.21), the definition of σ_{ij} for two-equation models has now slightly changed:

$$\sigma_{ij} = \frac{1}{Re_\infty} \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij}. \quad (6.37)$$

The last term is present due to the fact that the turbulent normal stresses must sum up to $-2\rho k$. In one-equation models this effect is ignored, since there is no separate equation for k . The modeling of the turbulent heat flux, equation (6.22), remains unchanged. The non-dimensional eddy viscosity is defined as:

$$\mu_t = Re_\infty \frac{\rho k}{\omega}. \quad (6.38)$$

The modeling constants are:

$$\begin{aligned} \sigma_{k1} = 0.5, \quad \sigma_{\omega 1} = 0.5, \quad \beta_1 = 0.075 \\ \beta^* = 0.09, \quad \kappa = 0.41, \quad \gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1} \kappa^2}{\sqrt{\beta^*}}. \end{aligned} \quad (6.39)$$

Menter [75] recommends the following boundary conditions:

$$\begin{aligned} \text{Free stream: } \omega_\infty = (1 \rightarrow 10), \quad k_\infty = \frac{10^{-(2 \rightarrow 5)} \omega_\infty}{Re_\infty} \\ \text{No-slip wall: } \omega = \frac{10}{Re_\infty} \frac{6\mu}{\rho \beta_1 (\Delta y_1)^2}, \quad k = 0 \end{aligned} \quad (6.40)$$

where Δy_1 is the normal grid spacing at the wall.

Due to the introduction of the turbulent kinetic energy k , the definition of the total energy becomes:

$$\rho E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho u_i u_i + \rho k, \quad (6.41)$$

i.e. the turbulence is taken into account in the conservation of energy, although this effect is of secondary importance.

Of more concern is the presence of the term $-\frac{2}{3}\rho k$ in the turbulent normal stresses, see equation (6.37). This term, when discretized with the Galerkin method, could lead to instabilities, because it appears in the Navier-Stokes equations as a first derivative. Therefore it is incorporated in the pressure by the introduction of an effective pressure p' ,

$$p' = p + \frac{2}{3} \rho k, \quad (6.42)$$

and discretized in an upwind manner. This and the contribution of k to the total energy changes the acoustic eigenvalues of the inviscid Jacobians A_i , equation (5.6). Instead of $u_i \pm a$ these are now given by $u_i \pm a'$, where

$$a' = \sqrt{(\gamma - 1) \left(H - \frac{1}{2} u_i u_i \right) + \left(1 - \frac{\gamma}{3} \right) k}. \quad (6.43)$$

Also this effect has been taken into account when the $k - \omega$ or any other two-equation turbulence model was used.

6.2.4 The BSL model

The main disadvantage of the $k - \omega$ model is its sensitivity to the rather arbitrary freestream values of k and ω , equation (6.40), in the outer part of the boundary layer. Menter's idea [75] was to combine the good properties of the $k - \omega$ model, *i.e.* no need for wall functions and accurate prediction in the near wall region of the boundary layer, with the good properties of the $k - \varepsilon$ model, mainly its freestream independence, and to avoid the bad properties of both models. The result is the Baseline (BSL) model, which is a blending between the $k - \varepsilon$ and $k - \omega$ models, expressed in $k - \omega$ formulation:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial \rho k u_i}{\partial x_i} = \frac{1}{Re_\infty} \frac{\partial}{\partial x_i} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i} \right] + \sigma_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho \omega k \quad (6.44)$$

$$\begin{aligned} \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho \omega u_i}{\partial x_i} = & \frac{1}{Re_\infty} \frac{\partial}{\partial x_i} \left[(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_i} \right] + 2\rho(1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \\ & + Re_\infty \frac{\gamma \rho}{\mu_t} \sigma_{ij} \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2, \end{aligned} \quad (6.45)$$

where the blending function F_1 has been defined as:

$$F_1 = \tanh(\arg_1^4) \quad (6.46)$$

$$\arg_1 = \min \left[\max \left(\frac{\sqrt{k}}{0.09\omega D}, \frac{1}{Re_\infty} \frac{500\mu}{\rho D^2 \omega} \right), \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega} D^2} \right] \quad (6.47)$$

$$CD_{k\omega} = \max \left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-20} \right) \quad (6.48)$$

In equation (6.48) D is the distance to the nearest wall.

The constants appearing in the BSL model are actually not constants any more, but a blending between those appearing in the $k - \omega$ model, equation (6.39), and those appearing in the $k - \varepsilon$ model:

$$\begin{aligned} \sigma_{k2} = 1.0, \quad \sigma_{\omega 2} = 0.856, \quad \beta_2 = 0.0828 \\ \beta^* = 0.09, \quad \kappa = 0.41, \quad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2} \kappa^2}{\sqrt{\beta^*}}. \end{aligned} \quad (6.49)$$

If any of the $k - \omega$ constants are indicated by ϕ_1 and any of the $k - \varepsilon$ constants by ϕ_2 , then the "constants" of the BSL model, indicated by ϕ , are given by:

$$\phi = F_1 \phi_1 + (1 - F_1) \phi_2. \quad (6.50)$$

Here F_1 is defined in equation (6.46). The freestream and wall boundary conditions are identical to the $k - \omega$ model, see equation (6.40).

6.2.5 The SST model

The Shear-Stress Transport (SST) model [75] uses the same transport equations for k and ω and the same blending function F_1 as the BSL model, equations (6.44), (6.45) and (6.46), but the $k - \omega$ constants have been slightly changed:

$$\begin{aligned} \sigma_{k1} &= 0.85, \quad \sigma_{\omega1} = 0.5, \quad \beta_1 = 0.075 \\ \beta^* &= 0.09, \quad \kappa = 0.41, \quad \gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega1}\kappa^2}{\sqrt{\beta^*}}. \end{aligned} \quad (6.51)$$

A second modification is the definition of the eddy-viscosity:

$$\mu_t = Re_\infty \frac{a_1 \rho k}{\max(a_1 \omega, |\Omega| F_2)} \quad (6.52)$$

$$F_2 = \tanh(\text{arg}_2^2) \quad (6.53)$$

$$\text{arg}_2 = \max\left(2 \frac{\sqrt{k}}{0.09 \omega D}, \frac{1}{Re_\infty} \frac{500 \mu}{\rho D^2 \omega}\right) \quad (6.54)$$

where $|\Omega|$ is again the magnitude of the vorticity vector. The constant a_1 is taken equal to 0.31. Comparison with the definition (6.38) of the BSL model shows that the eddy-viscosity of the SST-model is at most equal to, but usually less than, that of the BSL model. This reduces the overprediction of σ in adverse pressure gradient flows and improves the model's performance for such flows [75, 76, 104].

6.3 Results

In this section results are given for four test cases: (1) laminar, hypersonic flow over a hyperboloid flare, (2) transonic, turbulent shock-wave/boundary-layer interaction in a channel, (3) interaction of the base flow and an exhaust plume of a rocket model in a supersonic flow and (4) laminar and turbulent flow over an ogive cylinder at angle of attack. Two of these test cases, (1) and (3), are axisymmetric problems and consequently the axisymmetric source terms Σ^{ax} ,

$$\Sigma^{\text{ax}} = -\frac{\rho v}{y} \begin{pmatrix} 1 \\ u \\ v \\ H \end{pmatrix}, \quad (6.55)$$

must be discretized. Due to the singularity at the centerline, the very stable point-wise treatment of these source terms, see section 6.2.1, is not possible and at the moment the LDA-scheme is the best of all possible options to discretize these terms for axisymmetric problems.

6.3.1 Hyperboloid flare

This axisymmetric problem is the low Reynolds number case for the hyperboloid flare of the MSTP Workshop 1996 [107], for which a laminar flow can be assumed. The flow conditions are given in table 6.1. The geometry, shown in figure 6.1,

M_∞	6.73
T_∞	59.65 K
T_w	310 K
Re_∞/m	$3.50 \cdot 10^6/m$
Re_L	$2.073 \cdot 10^5$
L	0.05924 m

Table 6.1: Flow conditions for the hyperboloid flare test case

where the coordinates are dimensionless with the body length $L = 0.05924$ m, is an approximation of the windward side of the HERMES 1.0 contour at 30° angle of attack and 20° flap deflection. The hyperboloid part of the body is given by a combination of two fourth-order polynomials, see [107]. The grid is a 400×100 triangulated structured grid, taken from [18]; a coarsened version is shown in figure 6.2. Mach number and C_p isolines for the system PSI-scheme are depicted in figures 6.3 and 6.4 respectively. The Mach number isolines are smooth and clearly show the separated region between the separation and the reattachment shocks. Also the slip lines, which originate at the interaction points between those shocks and the bow shock can be seen in figure 6.3. In contrast to the Mach number, the pressure is wiggly, especially in the nose region. Albeit to a lesser extent, these wiggles are even present in the first-order N-scheme solution (not shown). Therefore the cause of these wiggles is probably a combination of the

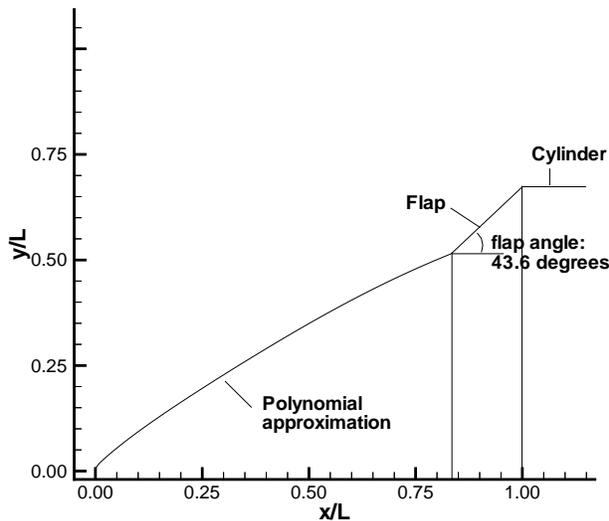


Figure 6.1: Geometry definition of the hyperboloid flare.

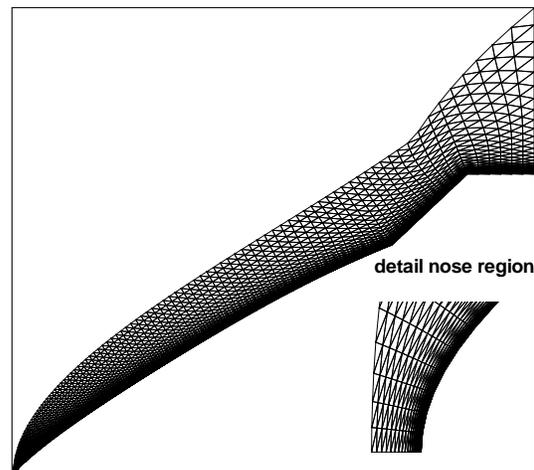


Figure 6.2: Triangulated 100×25 grid (Real grid is 400×100).

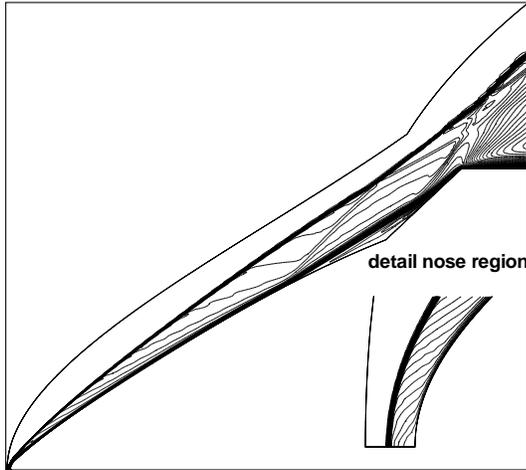


Figure 6.3: Mach number isolines for the hyperboloid flare, system PSI-scheme: min = 0.0, max = 6.73, step = 0.1.

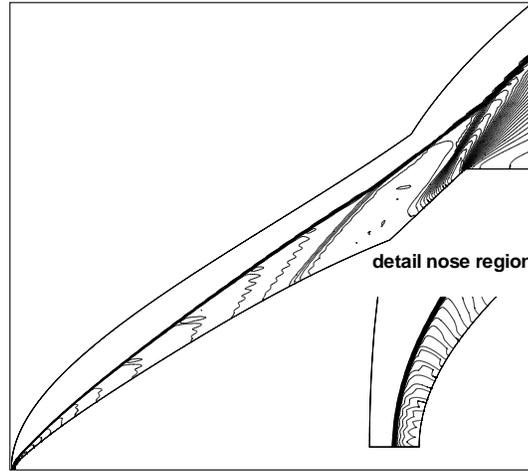


Figure 6.4: C_p isolines for the hyperboloid flare, system PSI-scheme: min = 0.0, max = 1.8, step = 0.04.

distribution of the axisymmetric source terms with the system LDA-scheme and the non-optimal performance of the current system PSI-scheme.

The convergence history is shown in figure 6.5. Due to the discretization of the axisymmetric source terms, only numerical Jacobians could be used and consequently the CFL-strategy was very heuristic. An initial solution (N-scheme) has been computed on the triangulated 200×50 grid, the first part of figure 6.5. As can be seen, Newton convergence has been obtained after a number of initial iterations. The maximum CFL number was 10^6 . The solution on this grid has been transferred to the 400×100 grid and used as initial solution for the first-order computation. Also on this grid the residuals could be driven to "machine zero". Finally, the second-order computation was started from this solution, in combination with N-scheme Jacobians and CFL = 100. As can be seen in figure 6.5, the second-order scheme converges approximately 2 orders relative to the first-order solution and then it stalls. The total amount of CPU time is 13 hours on a DEC alpha workstation (EV5.6 processor operating at 500 MHz).

For this test case surface pressure and heat transfer have been measured in the RWG tunnel of the DLR in Göttingen [63, 62]. Figures 6.7 and 6.8 show comparisons between the experimental data and numerical solutions obtained with the multi-dimensional upwind discretization technique of this thesis and a standard finite-volume solver on structured grids [73]. Although not measured, the numerical skin friction coefficients are computed and shown in figure 6.6. The pressure distribution of the current discretization technique agrees slightly better with the experiments than the standard finite-volume solver, see figure 6.7.

The two experimental runs for the Stanton number, figure 6.8, differ quite a bit, especially in the separation region, which shows that the experimental uncertainty

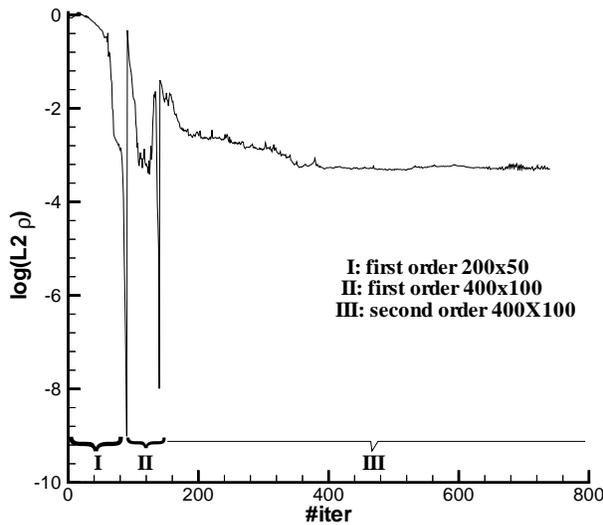


Figure 6.5: Convergence history for the hyperboloid flare.

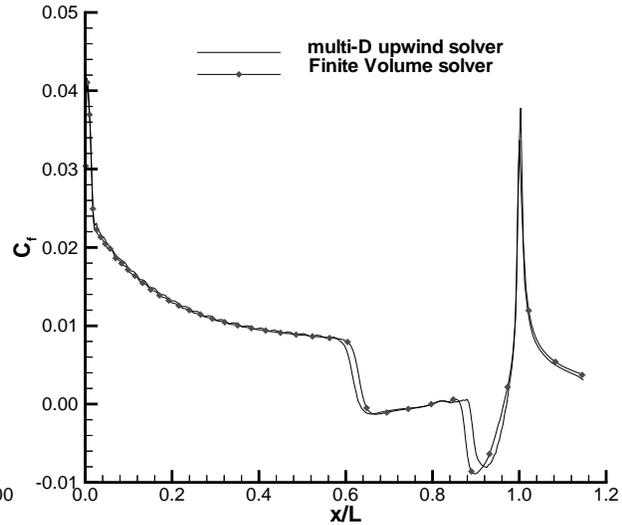


Figure 6.6: Computed skin friction coefficients for the hyperboloid flare.

is rather high. The values of run 15 fit quite well both computations, although the agreement with the multi-dimensional discretization is again slightly better. However, due to the uncertainties in the measurements, no decisive conclusions can be drawn.

Finally, the location of the first separation point has been determined experimentally using pressure data, oil flow visualization and liquid crystals. Numerically this location has been determined by means of the pressure distribution, figure 6.7, the skin friction coefficient, figure 6.6, and the Stanton number, figure 6.8. The results are summarized in table 6.2, where $L_{sep} = x_{hinge} - x_{sep}$, $x_{hinge} = 0.0493667$ m.

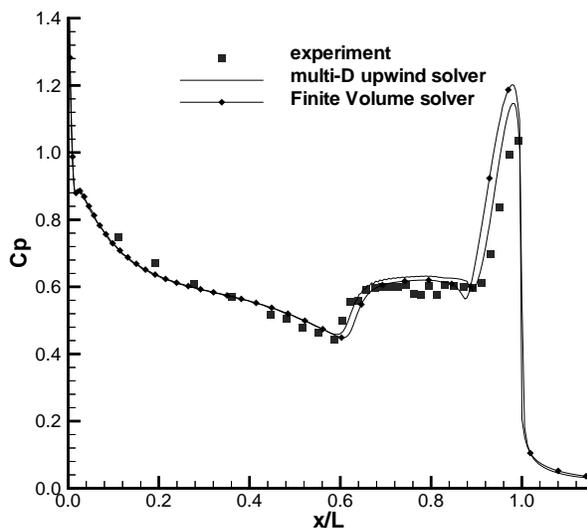


Figure 6.7: Computed and experimental pressure coefficients for the hyperboloid flare.

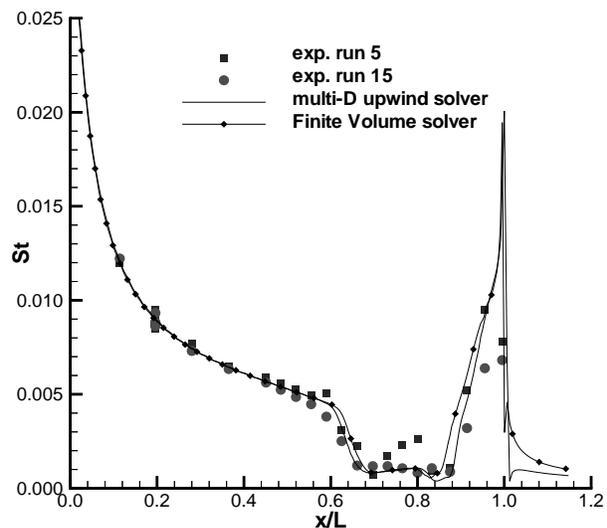


Figure 6.8: Computed and experimental Stanton numbers for the hyperboloid flare.

M_∞ :	0.615
T_t :	300 K
P_t :	96 kPa
Wall:	adiabatic
Re_∞/m	$10.96 \cdot 10^6/m$

Table 6.3: Flow conditions for the Détery bump (case C)

Model	Back pressure
SA	62.5 kPa
$k - \omega$	63.0 kPa
BSL	63.0 kPa
SST	61.5 kPa

Table 6.4: Back pressures for the different turbulence models

this shock-wave, the boundary-layer separates and a lambda shock structure is formed, see figure 6.10. At the (computational) inlet the incoming boundary layer is assumed to have zero thickness. As the interaction region is located far away from the inlet, this assumption does not influence the results in this region. In the experiments [34] an adjustable second throat was used to position the shock at the desired location. This is numerically a very difficult task and the usual approach is to adapt the back pressure at the outlet such that the location of the pressure jump at the lower wall coincides with the experiments. This depends strongly on the length of the computational domain and the turbulence model used. The outlet has been chosen at $x = 0.4261$ m and the correct back pressures, in combination with this computational domain, for the different turbulent models are given in table 6.4.

Two grids were created to compute the solution: a relatively coarse mesh, 10,420 nodes, which has been used to determine the speed-up's, see section 4.3, and a finer grid consisting of 41,231 nodes. Both are shown in figure 6.11 and have been created with the hybrid structured/unstructured approach of Carette [20]. For both grids the normal spacing on the walls is variable: $1.0 \cdot 10^{-6}$ m at the inlet and $5.5 \cdot 10^{-6}$ m at the outlet. The stretching factor in the viscous region is 1.2.

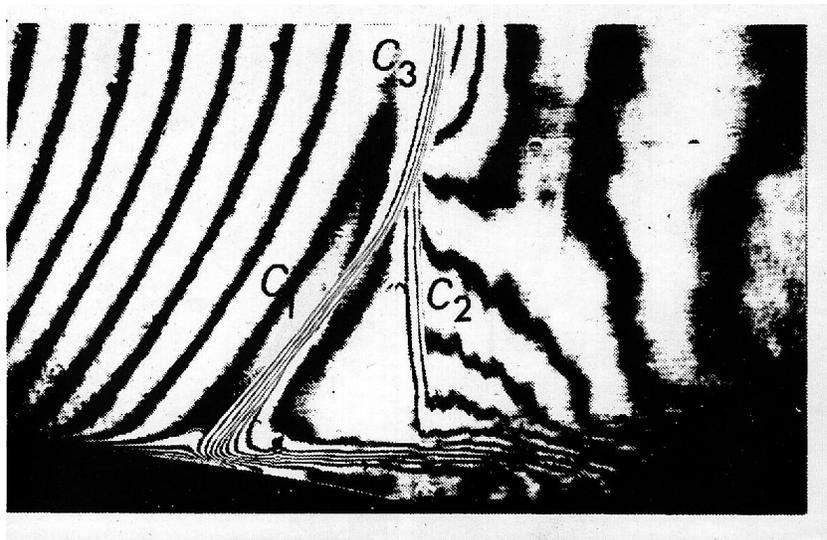


Figure 6.10: Interferogram in the region of interest

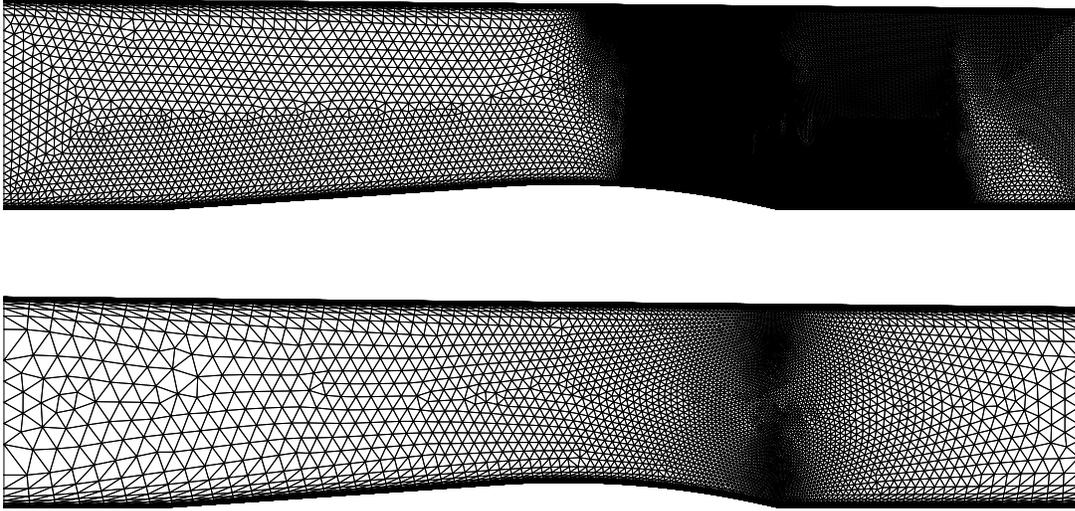


Figure 6.11: Fine grid (upper), 41,231 nodes, and coarse grid (lower), 10,420 nodes, for the Déleré bump.

To improve the resolution in the interaction region, the grids are locally refined, which is clearly visible in figure 6.11.

The convergence histories on the fine grid for the four turbulence models considered are given in figure 6.12. For all cases approximate N-scheme Jacobians are used in combination with a CFL number of 100, both for the first-order N-scheme and the second-order PSI-scheme. For the two-equation turbulence models, the CPU time on 8 processors of the Cray T3E was approximately 12 hours. The

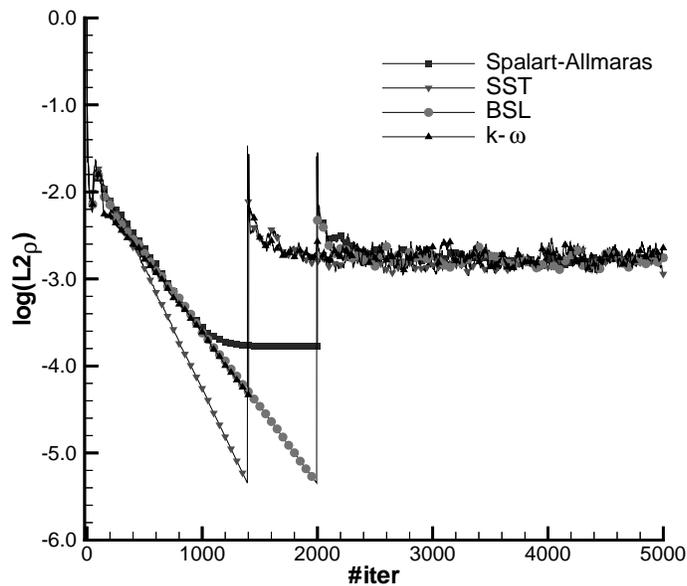


Figure 6.12: Convergence histories for the Déleré bump on the fine grid.

Spalart-Allmaras solution could be obtained in roughly 60% of this time, *i.e.* $7\frac{1}{2}$ hours. For the first-order scheme, all two-equation models converge quite well, especially the SST model, while the Spalart-Allmaras stalls, see figure 6.12. For the second-order scheme all models stall after one order of magnitude relative to the restart. However the maximum value of the eddy-viscosity did not change more than 0.2% between iteration 4000 and 5000 and the solutions have been assumed to be converged.

Mach number isolines for the four turbulence models can be found in figure 6.13. The $k - \omega$ and BSL solutions look very similar and fail both to predict the lambda shock observed in the experiment, see figure 6.10. The SST and, albeit to a much smaller extent, the Spalart-Allmaras model do show the lambda shock structure and therefore it can be expected that these two solutions will agree better with the experimental results. From figure 6.13 it is clear that the SST model also predicts a small lambda shock on the upper wall, which has not been observed in the experiments. This might be an indication that for this test case the limiting of the eddy-viscosity, equation (6.52), is too strong. Note that especially the solution of the SST-model shows quite a few wiggles downstream of the normal shock. As these wiggles are also present in the first-order solution, although less pronounced, figure 6.14, they are probably caused by the shock-capturing of the method. The wiggly behavior after shocks is also observed for the standard Roe scheme [91], which in one space dimension is equivalent to the upwind residual distribution scheme. A zoom of the grid superimposed on the Mach isolines in the region of interest for the SST and Spalart-Allmaras solution is shown in figure 6.15. The shocks are captured with either one or zero internal nodes and especially for the normal shock in the SST-solution it alternates between these two values. Clearly the wiggles originate from here. It must be said that on coarser grids this effect is less evident, because of more numerical cross diffusion.

Figures 6.16 and 6.17 show the computed and experimental pressure distributions on the lower and upper walls respectively. The coarse and fine grid solutions are nearly identical and therefore it is assumed that especially the fine grid results are grid converged. As the pressure distribution in front of the shock-wave is determined by inviscid phenomena, all models give identical results in these parts of the flow field. It is clear that the SST-model gives the best agreement with the experimental data, although it does not fit exactly. Especially the pressure plateau between the two legs of the lambda-shock on the lower wall is not predicted. The Spalart-Allmaras model predicts something that looks like such a plateau, but at too high a level, probably caused by the higher back pressure (Spalart-Allmaras: 62.5 kPa, SST: 61.5 kPa). The BSL and the $k - \omega$ models are nearly identical and give the worst agreement with the experimental pressure distribution, mainly due to the inability to generate the lambda shock. Also at the upper wall the SST-model agrees best with the experimental data, again followed by the Spalart-Allmaras model.

Apart from the deficiencies in the turbulence models, there is another possible

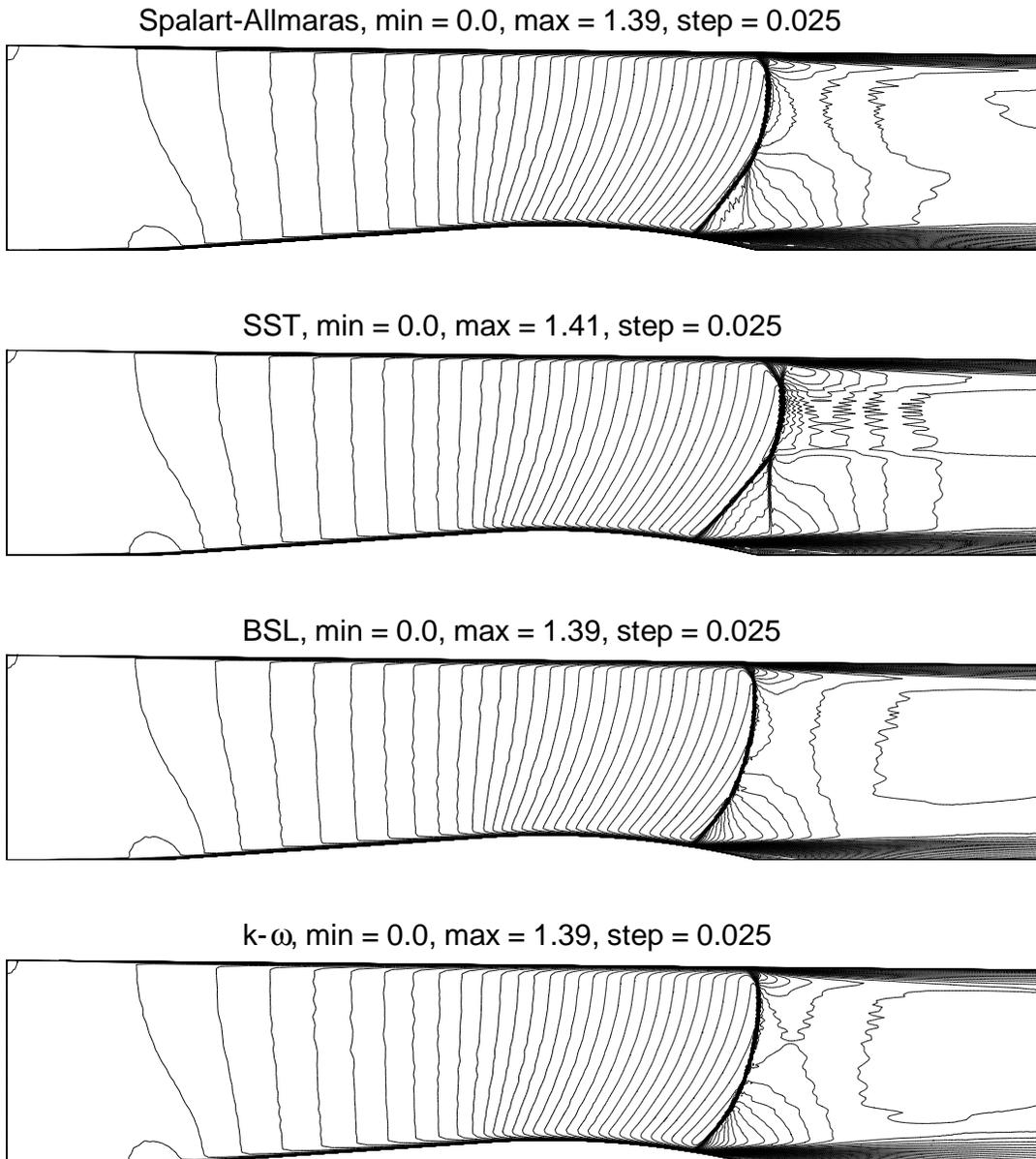


Figure 6.13: Mach number isolines for the solutions of the four turbulence models on the fine grid for the Délyery bump, PSI-scheme.

cause for the differences between computation and experiment, namely the treatment of the outlet boundary. As described earlier, the constant back pressure at the outlet has been adjusted such that the shock on the lower wall is located at the correct position. In the experiments an adjustable second throat is used to choke the flow, which allows a pressure variation over the computational outlet. As can be seen in figures 6.16 and 6.17, the experimental pressure at the location of the computational outlet is indeed different for the lower and upper walls. However the result of an SST computation with an outlet located much further downstream,

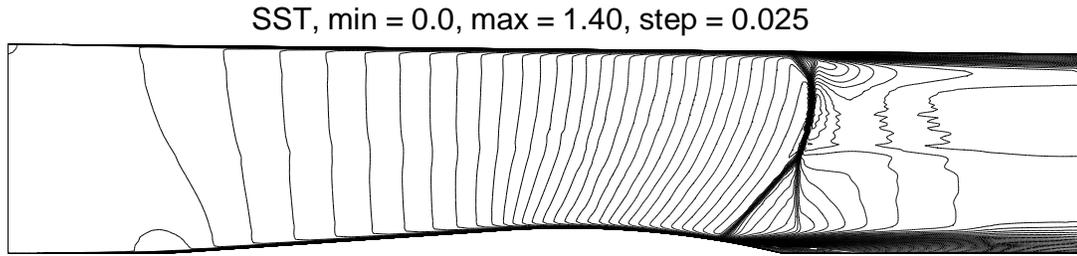


Figure 6.14: Mach number isolines for the SST solution on the fine grid for the Déleury bump, N-scheme.

at $x = 0.7$ m, did not show a significant change in the wall pressure distributions and therefore it has been assumed that the differences between computation and experiments are not caused by the outlet boundary treatment.

Apart from the wall pressure distributions, also the velocity profiles in the flow at several x -locations have been measured [34]. The results are shown in figures 6.18 to 6.21. Figures 6.18 and 6.19 clearly show the separated region after the first shock, which approximately ends at $x = 0.33$ for the Spalart-Allmaras model, the SST-model and the experiments, see figure 6.20. This region is shorter for both the BSL and the $k - \omega$ model, as can be seen in figure 6.20. Although the

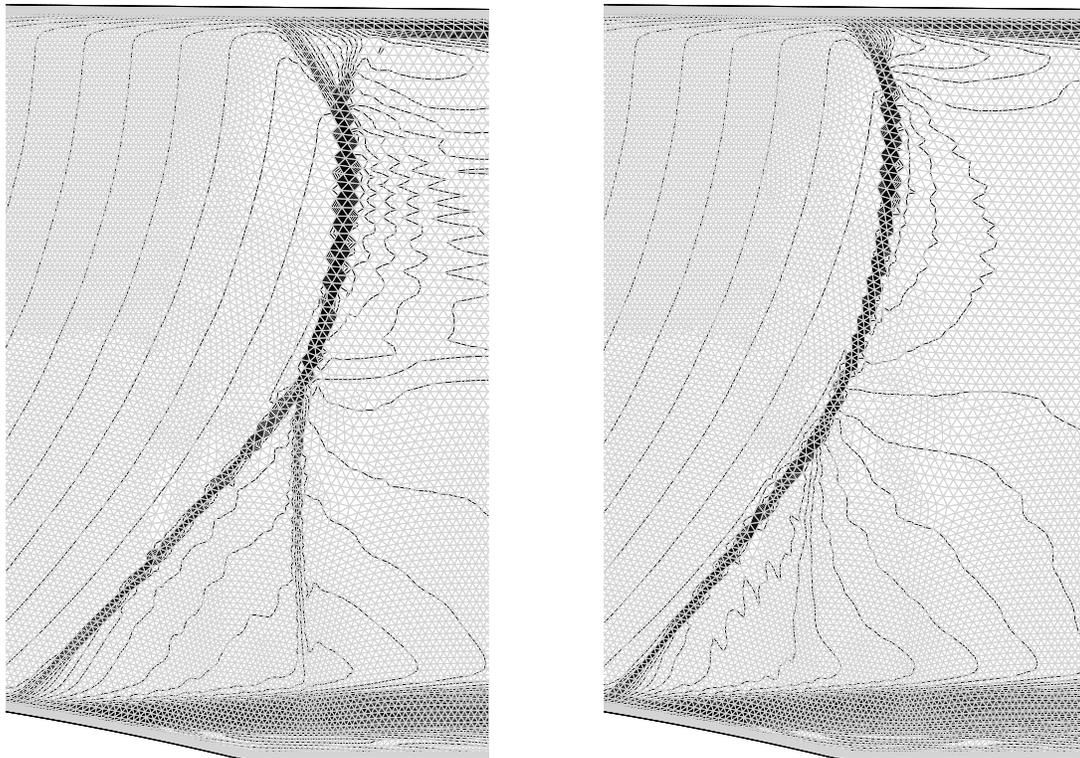


Figure 6.15: Zoom of figure 6.13 in the region of interest for the SST model (left) and the Spalart-Allmaras model (right).

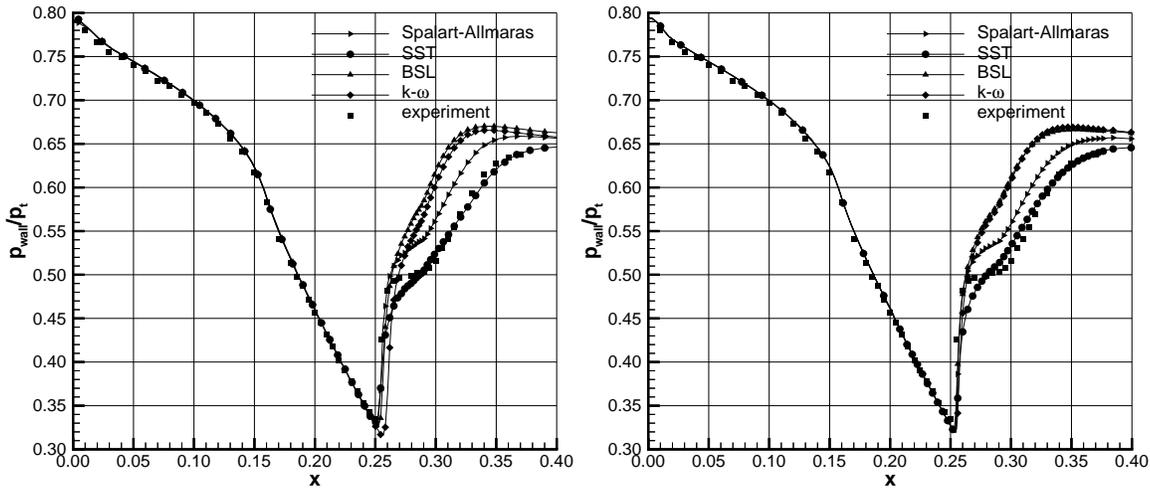


Figure 6.16: Computed and experimental pressure distribution on the lower wall, left: coarse grid, right: fine grid.

recirculation region is predicted by all turbulence models considered, the quantitative numerical results are far from the experimental values, especially in the cross-section $x = 0.29$, see figure 6.19. Note that the SST-model does not give a better velocity profile than the other models, although its wall pressure distribution is closer to the experiment, especially on the lower wall. At $x = 0.38$, figure 6.21, the flow is fully reattached for all turbulence models.

From the numerical point of view, the conclusion for this test case is that the discretization technique developed here, in combination with the implicit integration method results in a very stable algorithm: starting from a uniform flow did not pose any problem for the solver.

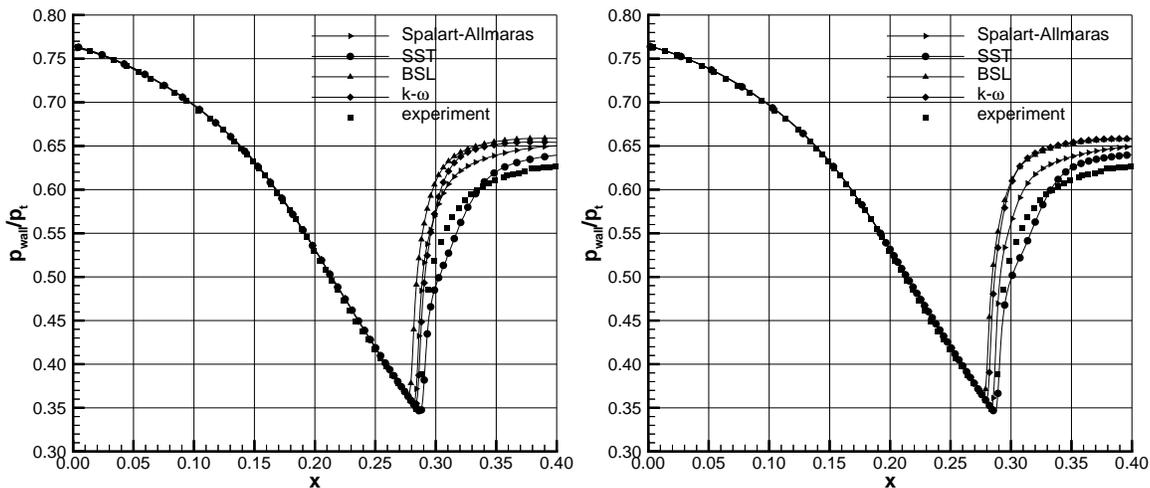


Figure 6.17: Computed and experimental pressure distribution on the upper wall, left: coarse grid, right: fine grid.

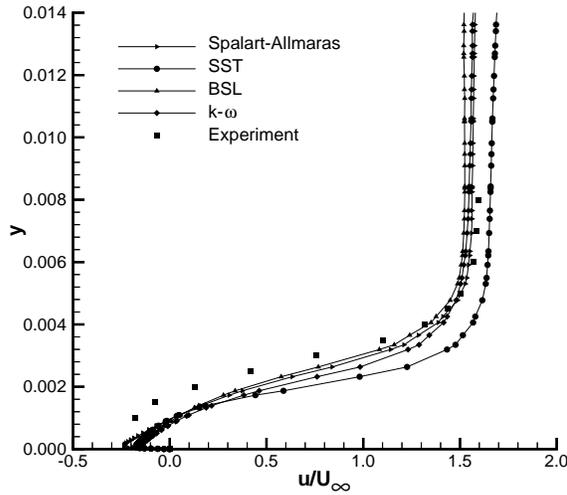


Figure 6.18: Experimental and computed velocity profiles for the fine grid at $x = 0.27$.

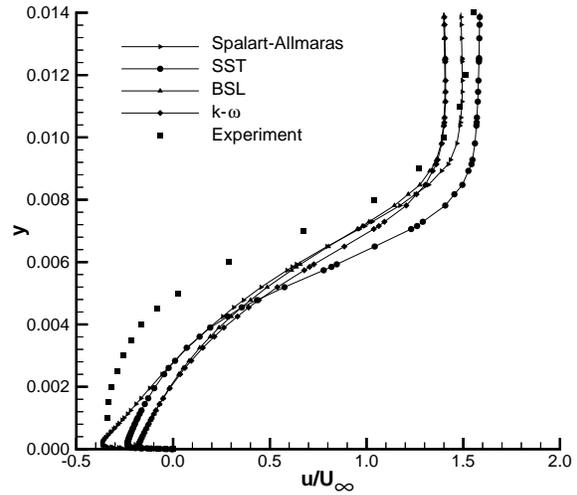


Figure 6.19: Experimental and computed velocity profiles for the fine grid at $x = 0.29$.

As can be expected for such a complicated flow, none of the considered turbulence models is able to predict the flow field very accurately. The wall pressure distributions of especially the SST-model agree reasonably well with the experimental data, but for more sensitive quantities like velocity profiles the difference between computation and experiment can be as large as a factor 2, especially in and after separation regions.

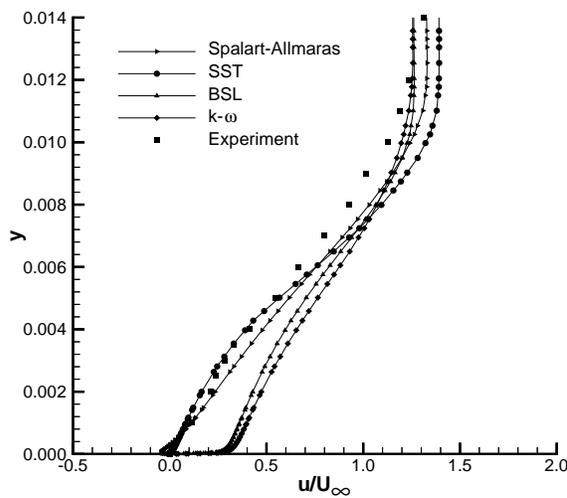


Figure 6.20: Experimental and computed velocity profiles for the fine grid at $x = 0.33$.

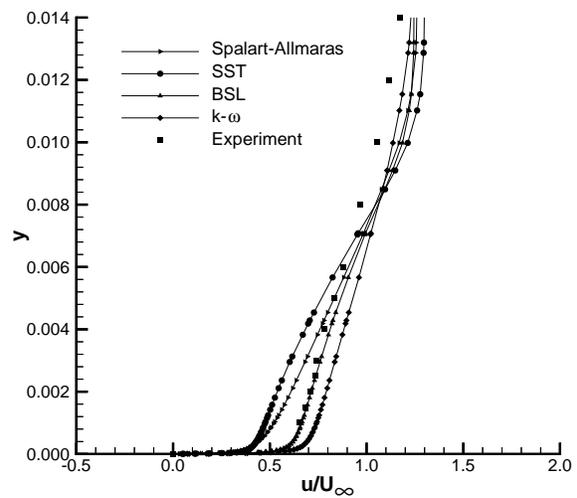


Figure 6.21: Experimental and computed velocity profiles for the fine grid at $x = 0.38$.

6.3.3 Rocket model with exhaust nozzle

This axisymmetric blunted cone-cylinder configuration with exhaust nozzle, whose geometry is defined in figure 6.22, has been investigated experimentally in the Delft University transonic/supersonic TST 27 windtunnel [8]. The main interest of this experiment is the influence of the exhaust plume on the base flow.

Experimental data is available for a whole set of conditions [8], but only one case has been computed here, see table 6.5. Here P_{t_j} and T_{t_j} indicate the total

M_∞	T_t (K)	P_t (MPa)	Re_∞/m	P_{t_j} (MPa)	T_{t_j} (K)	T_{wall} (K)
2.98	285	0.5751	$46.5714 \cdot 10^6/m$	3.13	285	263

Table 6.5: Flow conditions for the rocket model.

pressure and the total temperature of the jet respectively. Note that the value of T_{t_j} corresponds with a cold plume. The conical nozzle inside the model has been designed such that the jet exit Mach number is approximately 4. Together with the conditions of table 6.5, this corresponds to an underexpanded jet. The interior nozzle has been computed assuming a zero radial velocity at the subsonic inlet. In combination with the total conditions given in table 6.5, this completes the information needed to treat the subsonic inlet boundary, see section 5.2.3. In the numerical simulations the shape of the nozzle upstream of the throat has been slightly changed, such that the inlet Mach number is approximately 0.2. It is assumed that the influence of this approximation on the solution in the base region is negligible. In the experiments the exterior boundary layer has been tripped near the nose, see figure 6.22, and a fully turbulent flow has been assumed in the computations.

The hybrid structured/unstructured approach [20] has been used to generate two grids. The coarse version, shown in figure 6.23, has 15,505 nodes, of which approximately 12,000 are located in the structured part near the walls. Due to the

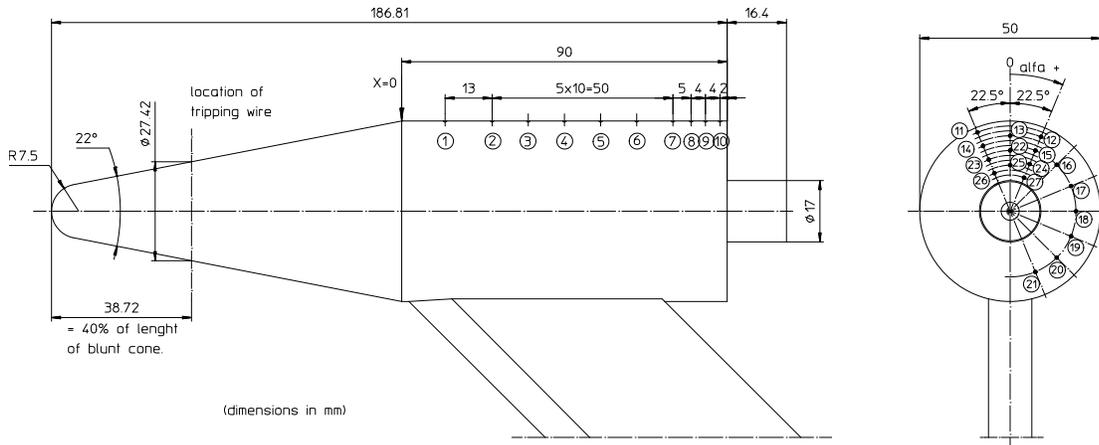


Figure 6.22: Geometry definition of the rocket model and location of the pressure taps.

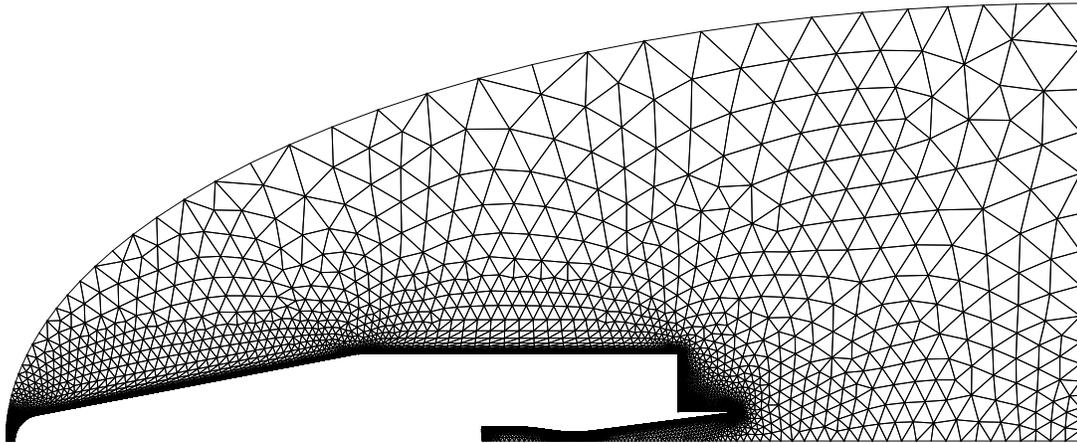


Figure 6.23: Coarse grid (15,505 nodes) for the rocket model.

high Reynolds number the initial normal spacing is 10^{-6} m, leading to aspect ratios of more than 4000. As can be seen in figure 6.23, the unstructured part of the grid is quite coarse and in fact much too coarse to predict the flow field downstream of the exhaust nozzle accurately. Therefore a very fine grid consisting of 119,519 nodes has been created. The grid in the base region is shown in figure 6.24. To investigate the solution in this base region even in more detail, it has been computed separately on a 925×481 triangulated structured grid, which is on an average 5 times finer in each coordinate direction than the grid shown in figure 6.24. The computational domain is a rectangular box, where the x -coordinate varies between 0.18681 m and 0.23 m and the y -coordinate between 0.0085 m and 0.029 m. On the inflow parts of

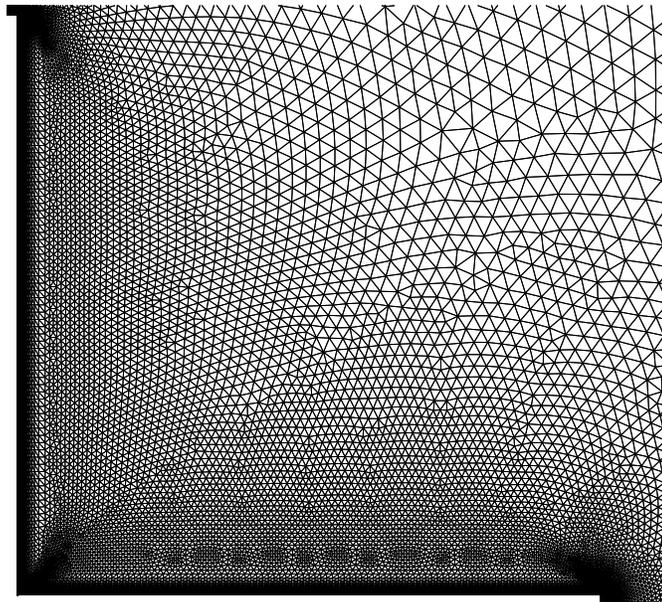


Figure 6.24: Base region of the fine grid (119,519 nodes) of the rocket model.

the boundaries $x = 0.18681$ m, $y = 0.0085$ m and $y = 0.029$ m a supersonic inlet has been assumed - the flow variables have been linearly interpolated from the second-order solution of the fine grid. The outflow boundary $x = 0.23$ m has been chosen such that the flow is supersonic and consequently no boundary conditions need to be described there.

An initial solution on the fine grid has been obtained by interpolating the first-order coarse grid solution to this fine grid. The convergence history, starting from this solution, for the SST model, computed on 16 processors of the Cray T3E, is given in figure 6.25, which, again, shows the stalling of the second-order scheme. Due to stability reasons, the CFL number had to be restricted to 20, for both the first- and second-order computations in combination with numerical N-scheme Jacobians. At first glance this seems very low, but the explicit Runge-Kutta time integrators were unstable, regardless of the CFL number. These explicit schemes even failed for the first-order, laminar case, when restarting from the fully converged (implicit) solution. The total CPU-time for the two-equation models is approximately 19 hours. As for the Délerly bump, the solution in combination with the Spalart-Allmaras model could be obtained in 60% of this time.

Mach number isolines for the fine grid SST solution are depicted in figure 6.26. As the main characteristics of the flow are determined by inviscid phenomena, except in the base region, the solutions of the other turbulence models look very similar and are therefore not shown. The plume and barrel shock and the slip line in between are clearly visible, as well as the fact that the nozzle is underexpanded.

Computational and experimental pressure distributions for the coarse and fine grids on the cylindrical part of the model are shown in figure 6.27. It can be seen that the boundary layer thickness has some influence near the corners (compare

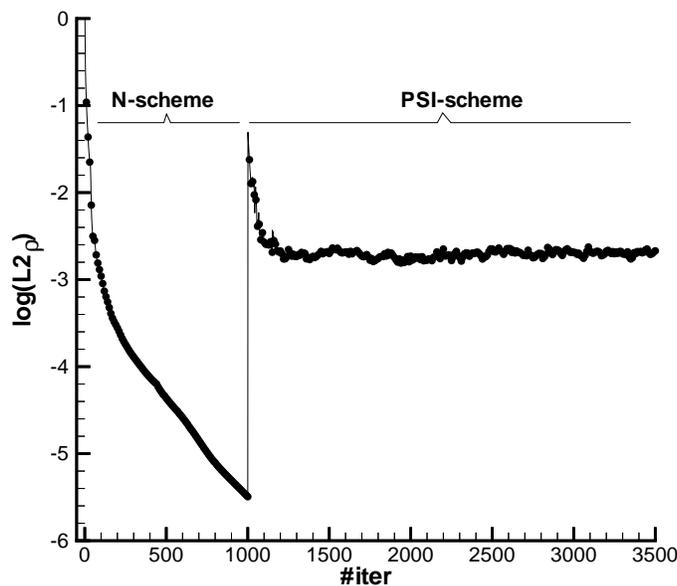


Figure 6.25: Convergence history for the rocket model on the fine grid, SST model.

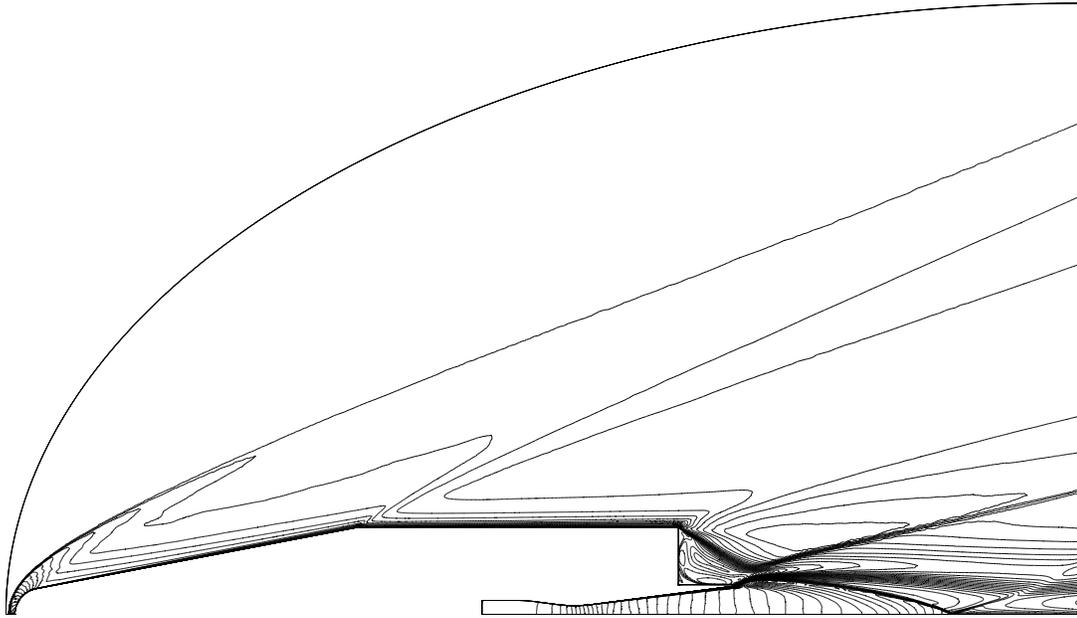


Figure 6.26: Mach number isolines for the rocket model, fine grid, SST turbulence model, PSI-scheme; min = 0.0, max = 6.3, step = 0.15.

the turbulent and laminar solutions), but the pressure in this part of the flow field is mainly determined by the inviscid equations. All results are nearly identical and fit the experimental data quite well. On the cylindrical part the solutions are grid converged, as both grids give the same results for all turbulence models.

As explained earlier, the main interest of this experiment is the flow in the base region. An indication of the flow field can be seen in figure 6.28, which shows the

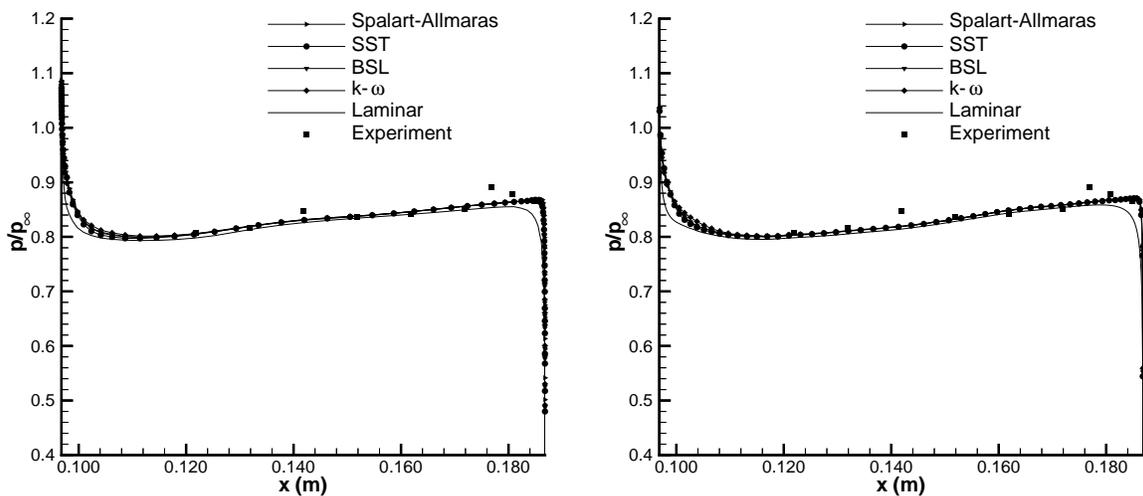


Figure 6.27: Computed and experimental pressure distribution on the cylindrical part for the rocket model, left: coarse grid, right: fine grid.

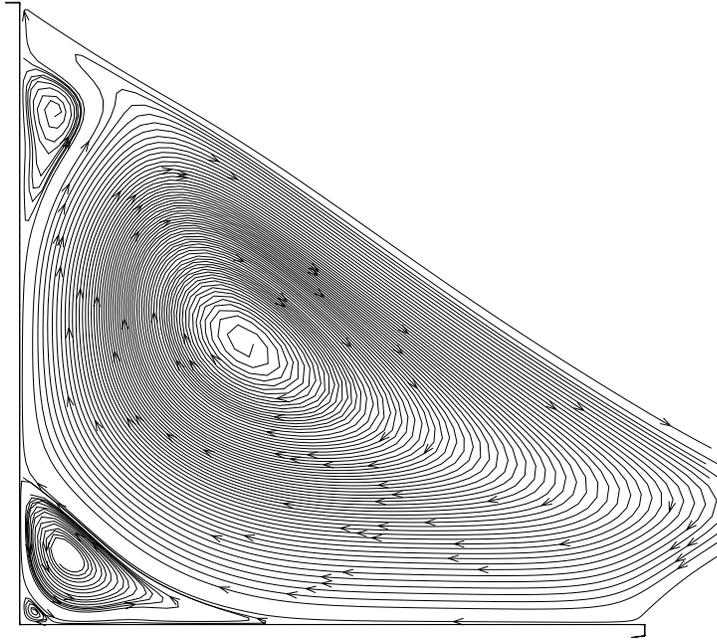


Figure 6.28: Computed streamlines in the base region of the rocket model for the SST turbulence model on the fine grid, PSI-scheme.

computed streamlines based on the numerical solution of the SST model. Theoretically these streamlines must be closed because of mass conservation reasons, but due to integration errors this is numerically not possible. It is observed that four counter-rotating vortices are present in the base region. Consequently the flow is extremely complex and together with the fact that the Mach number is rather low in that area, this makes the problem very difficult to compute for a compressible flow solver.

Despite this, results have been obtained and the calculated pressure distributions in this region are compared with the experimental results in figure 6.29 and 6.30, see figure 6.22 for the location of the pressure taps.

The following conclusions can be drawn.

- especially from figure 6.30 it is clear that the laminar solution is completely wrong. On the triangulated structured grid it is highly unsteady and consequently the pressure distribution shown only serves as illustration - it can be quite different some time steps later. Numerically Von Karman vortex sheets are observed. On the coarse and fine grid, see figure 6.29, the solution is steady, due to numerical diffusion. Note that the computed pressure distributions on these grids are a factor 3 too low compared with the experimental data.
- all solutions on the coarse grid are not grid converged in the base region, as the computed pressures are lower than on the fine grid. The pressure distribution in the base region of the solution obtained with the Spalart-Allmaras model on the fine grid is identical to the distribution computed on

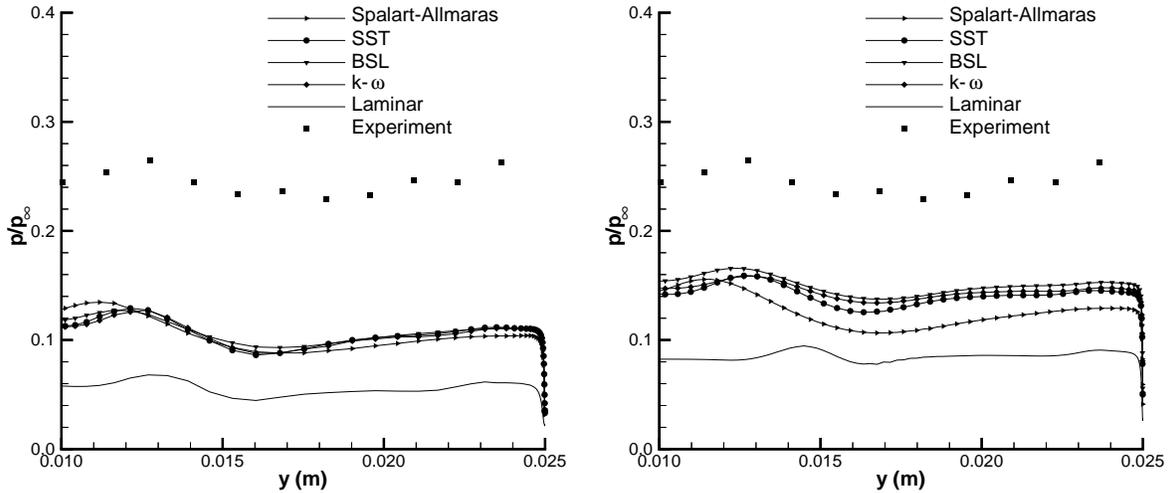


Figure 6.29: Computed and experimental pressure distribution in the base region for the rocket model, left: coarse grid, right: fine grid.

the triangulated structured grid in the base region, see figures 6.29 and 6.30. Consequently this solution is grid converged. However, it is clear that the solutions of the two-equation models on the fine grid are not grid converged, as the base pressure distribution on the triangulated structured grid is 25 to 50% higher.

- all turbulence models predict the base pressure too low compared to the experimental values. Taking into account the conclusion mentioned above, this difference is caused by the deficiency of the turbulence model in case of the Spalart-Allmaras model. For the two-equation models the situation is not

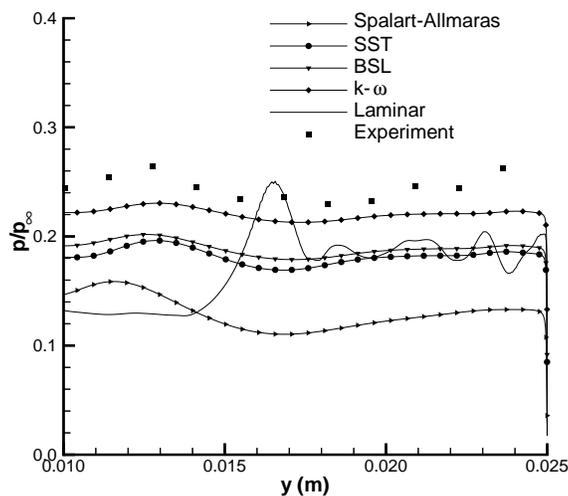


Figure 6.30: Computed and experimental pressure distribution in the base region for the rocket model, triangulated 925×481 structured grid in base region.

so evident. As the base pressure distributions on the grids used differ quite a lot, the solutions cannot be assumed grid converged and it might be possible that on even finer grids the base pressure increases further. Furthermore, it is not clear what is the effect of imposing the entire solution (interpolated from the fine grid solution) on the inflow boundaries of the triangulated structured grid in the base region. There are indications that, especially in the interaction region between the exhaust plume and the base region, the solution changes quite a bit and thus a wrong solution is imposed there.

- on the triangulated structured grid, figure 6.30, the $k - \omega$ model predicts a significantly higher pressure in the base region than the BSL and SST-models, whose formulations are almost identical to the $k - \varepsilon$ model in this part of the flow field, see sections 6.2.4 and 6.2.5. Due to the limiting of the eddy-viscosity, equation (6.52), the pressure distribution of the SST-model is slightly lower than the distribution obtained with the BSL-model. Furthermore, all two-equation models give a much higher pressure than the Spalart-Allmaras model on this grid.

The main conclusion is that the correct prediction of the base flow characteristics is a difficult problem, which requires extremely fine grids, especially for the two-equation models. The result obtained with the $k - \omega$ model on the triangulated 925×481 structured grid in the base region is surprisingly good, as it is believed that the base flow characteristic cannot be predicted accurately by eddy-viscosity turbulence models. Apparently this is not the case here, as long as sufficiently fine grids are used. However, it might be a coincidence and more cases should be computed to draw a decisive conclusion.

The computational effort to obtain a converged solution on the triangulated structured grid is enormous - ± 100 CPU hours on 16 processors of the Cray T3E for the two-equation models. The one-equation Spalart-Allmaras model does not need such fine grids, but the computed base pressure is almost a factor two too low.

It should be noted that only pressure distributions have been compared. An even more important, and probably also more difficult problem is the correct prediction of the heat flux in the base region, especially for hot plumes.

6.3.4 Ogive cylinder

In this section the results of the laminar and turbulent computation over an ogive cylinder are discussed. The geometry definition is shown in figure 6.31. It has been investigated experimentally by ONERA [2] for a whole set of incidence angles, but only $\alpha = 10^\circ$ has been computed here. The flow conditions are given in table 6.6. In the experiments two cases are distinguished: the case where the boundary layer undergoes natural transition and the case where the boundary layer has been tripped near the apex. For the numerical computation fully laminar and fully turbulent flow have been assumed respectively.

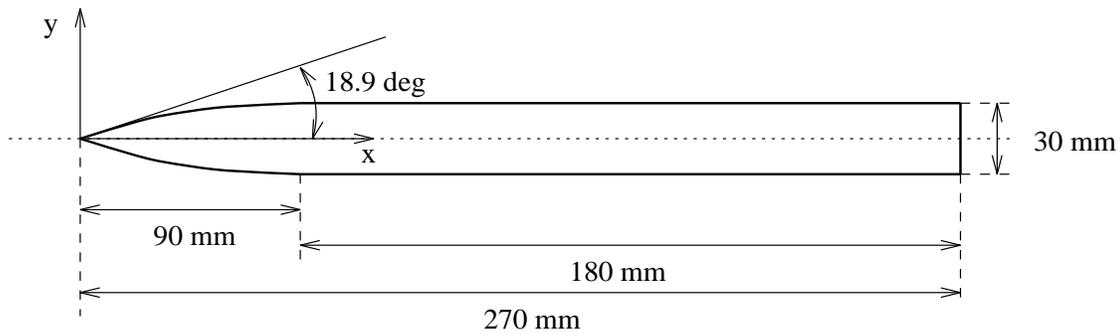


Figure 6.31: Geometry definition of the ogive cylinder.

M_∞	2.0
α	10°
T_t	330 K
Wall	adiabatic
Re_∞/m	$5.33 \cdot 10^6/m$
P_t	50 kPa

Table 6.6: Flow conditions for the ogive cylinder test case.

The laminar case

The mesh has been created with the grid generator of the University of Swansea [90] and consists of 390,965 nodes and 2,302,869 tetrahedra. Due to limitations

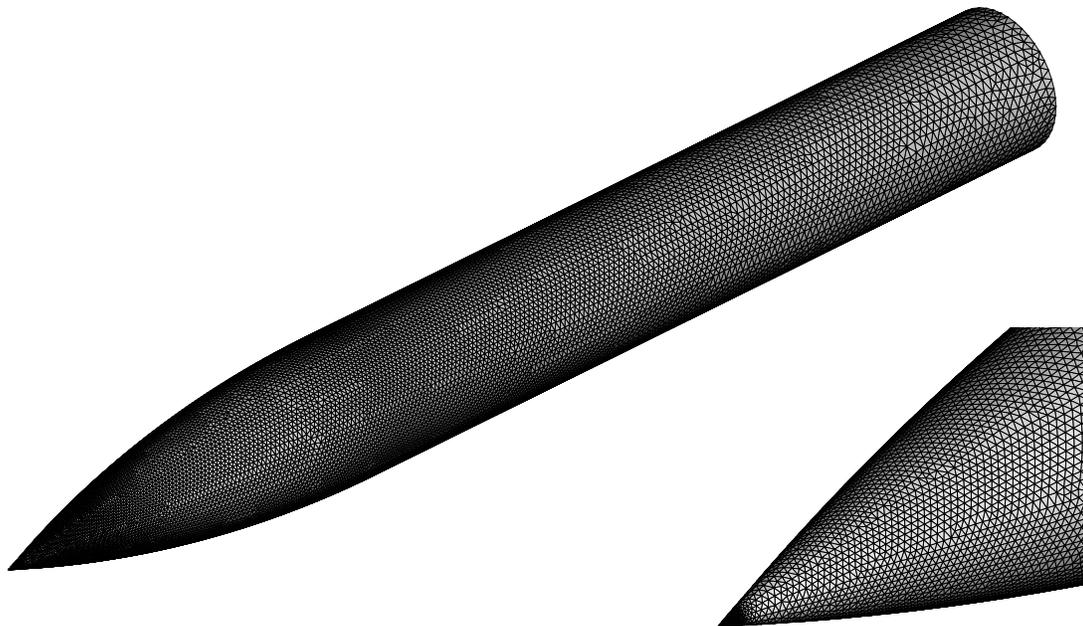


Figure 6.32: Surface grid of the ogive cylinder (17,893 nodes), entire and zoom near the apex.

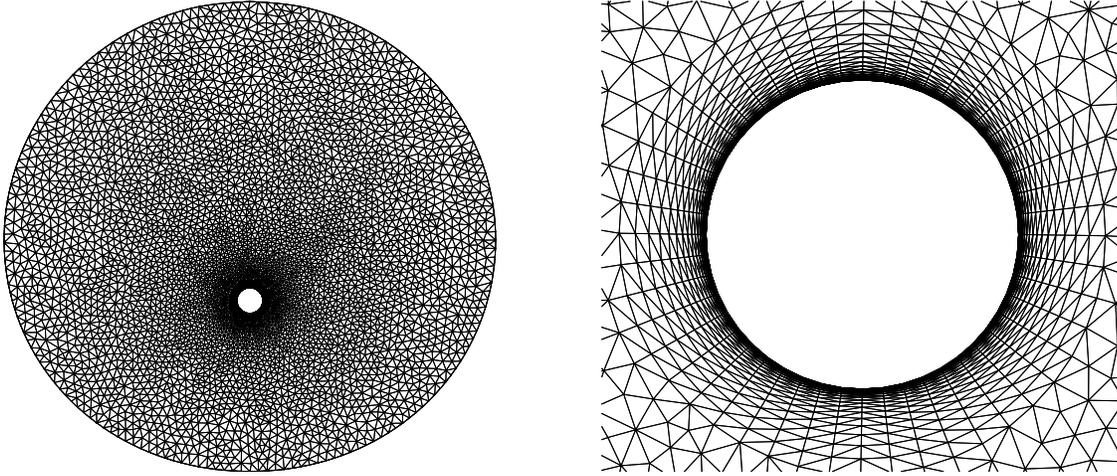


Figure 6.33: Surface grid of the outflow plane, entire (left) and zoom (right) (5480 nodes).

of the grid-generation software, a symmetry plane could not be used and the full problem had to be computed. The surface mesh of the ogive-cylinder and of the outflow plane are shown in figure 6.32 and 6.33 respectively. In the boundary layer 26 viscous layers are used with an initial normal spacing of 10^{-5} m and a stretching factor of 1.2. This guarantees a smooth transition to the isotropic part of the grid, as can be seen in figure 6.33.

To meet the memory requirement for the implicit solver, the code base been run on 32 processors of the Cray T3E. Newton convergence has been obtained for the first-order scheme in 15 iterations, in combination with numerical Jacobians, see figure 6.34. The starting CFL number was 1.0 and was multiplied every time step by 5.0, until a maximum of 10^6 . The second-order scheme, with approximate

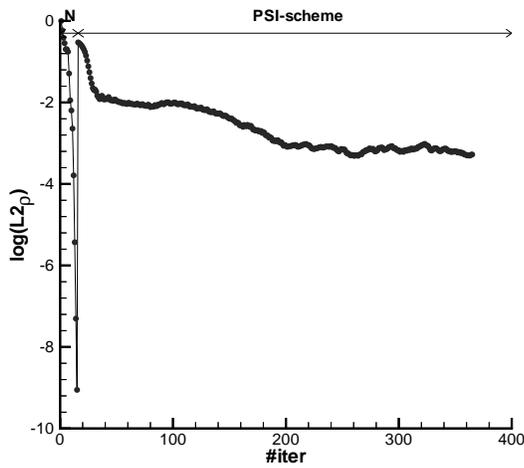


Figure 6.34: Convergence history for the laminar ogive cylinder, coarse grid.

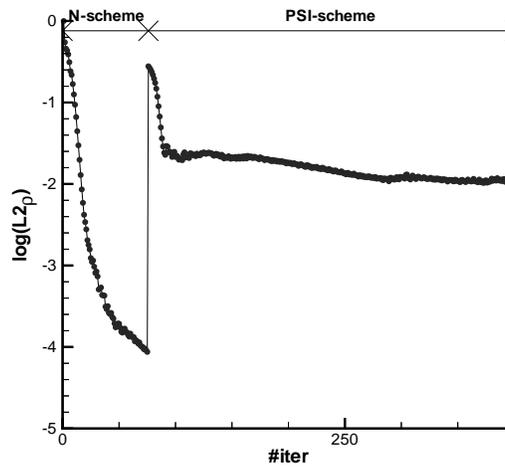


Figure 6.35: Convergence history for the laminar ogive cylinder, fine grid.

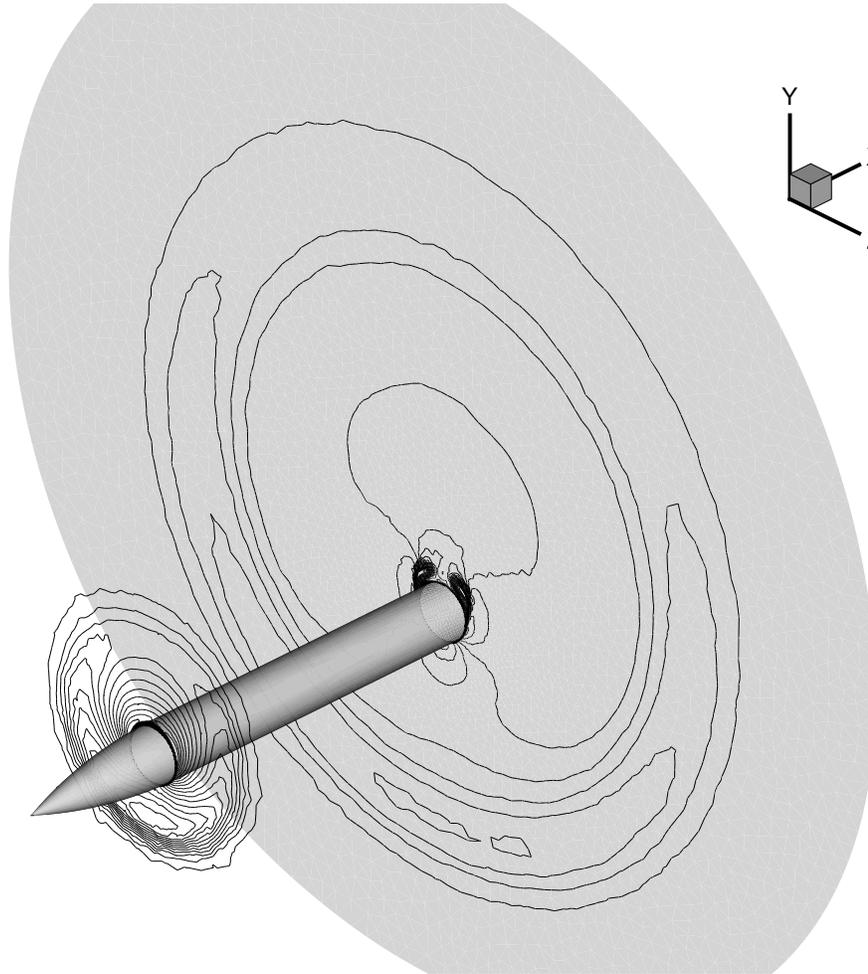


Figure 6.36: Mach number isolines for the laminar ogive cylinder, system PSI-scheme, min = 0.0, max = 2.1, step = 0.02.

N-scheme Jacobians and CFL = 100, stalls after $2\frac{1}{2}$ orders relative from the restart. The total CPU time is roughly 5 hours, of which 90% is taken by the second-order scheme.

Mach number isolines in the outflow plane ($x = 0.27$ m) and in the plane $x = 0.08$ m are shown in figure 6.36. In the outflow plane the primary vortices are clearly visible and also the bow shock can be identified. The plane $x = 0.08$ m does not correspond with a set of cell faces, and therefore the tetrahedra are cut and the solution must be interpolated. This is the cause that the isolines in this plane do not look as smooth as in the outflow plane.

The skin friction lines are depicted in figure 6.37. Two separation lines can be distinguished and consequently two counter-rotating vortices are present (on each side of the body). The primary separation line shows a wavy behavior, an indication that this solution is not stable and most likely unsteady.

The laminar solution has also been computed on the grid used for the turbulent

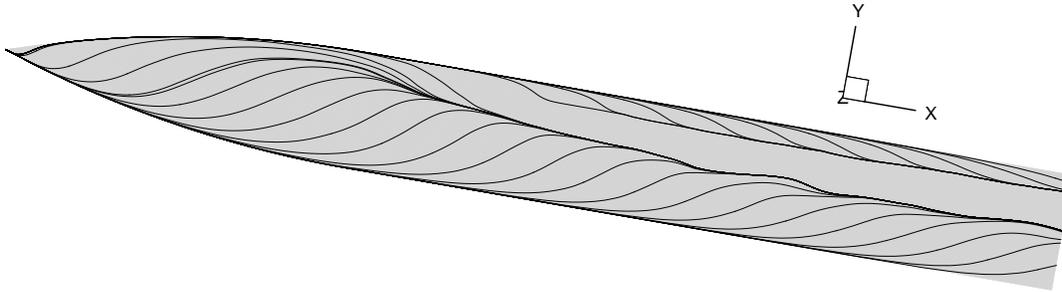


Figure 6.37: Computed skin friction lines for the laminar ogive cylinder, system PSI-scheme.

case. The convergence history is shown in figure 6.35. The initial normal spacing of this grid is 10 times smaller than the coarse grid and thus the aspect ratios of the cells near the body are much larger (maximum value ± 1500). Consequently, the nonlinear problem is more difficult to solve and no Newton convergence could be obtained for the first-order scheme. Therefore both the first and the second-order solutions are computed in combination with analytical N-scheme Jacobians and $CFL = 100$. On this fine grid, the second-order solution only converges one order of magnitude relative from the restart, figure 6.35. Also this indicates that the laminar solution is probably unsteady. The total CPU-time on 64 processors of the Cray T3E is approximately 7 hours.

Figures 6.39 to 6.43 compare the computed surface pressure distributions on the two grids with the experimental values in several crossflow planes. The angle θ is defined in figure 6.38. In the first two cross-sections, figures 6.39 and 6.40, the agreement is quite good, but further downstream it becomes worse. In the plane $x = 0.135$ m, figure 6.41 the numerical solutions start to deviate from the

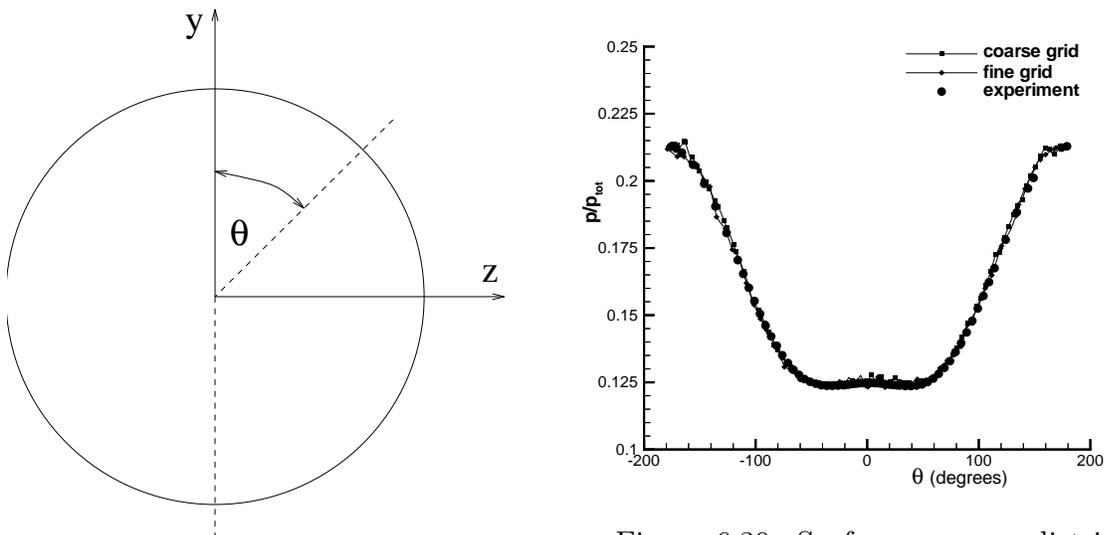


Figure 6.38: Definition of angle θ .

Figure 6.39: Surface pressure distribution, $x = 0.045$ m, laminar case.

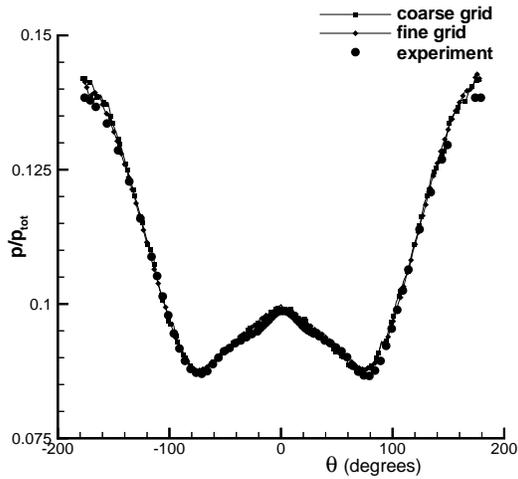


Figure 6.40: Surface pressure distribution, $x = 0.09$ m, laminar case.

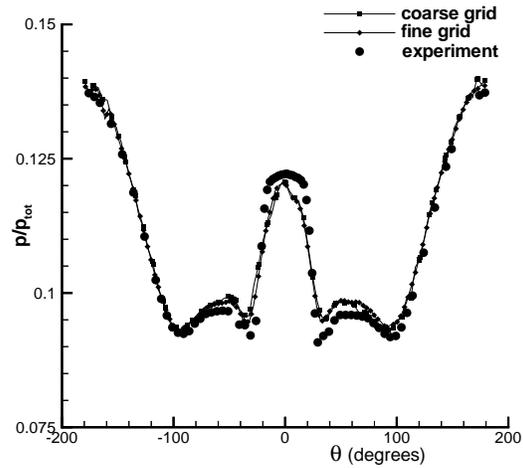


Figure 6.41: Surface pressure distribution, $x = 0.135$ m, laminar case.

experimental values, but on both grids the pressure distribution is still symmetric. This is not the case anymore in figures 6.42 and 6.43. Especially on the fine grid, the solution is asymmetric. The probable explanation for this is that the flow in the experiment is transitional, while in the computation laminar conditions have been assumed. Apparently the transition to turbulence stabilizes the flow field, such that the experimental flow field is steady.

Note that the asymmetry is found, only because the full problem is computed.

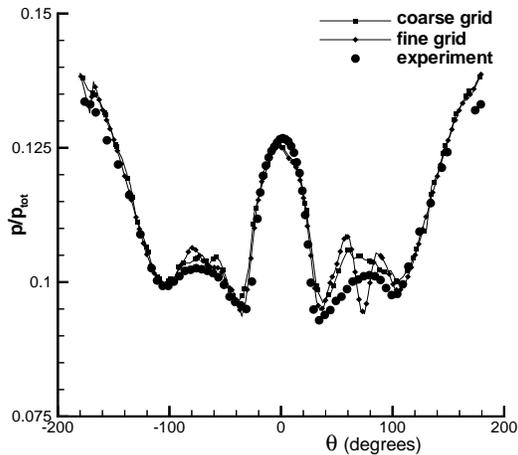


Figure 6.42: Surface pressure distribution, $x = 0.18$ m, laminar case.

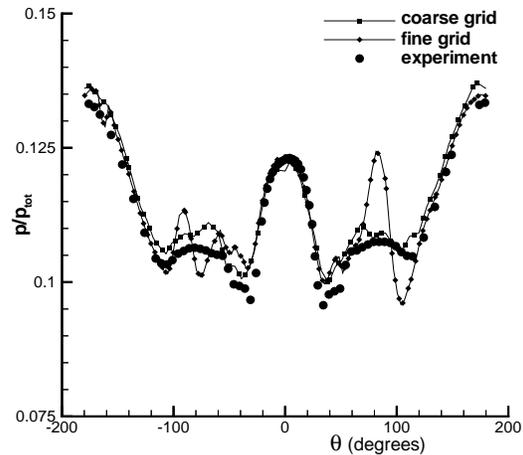


Figure 6.43: Surface pressure distribution, $x = 0.225$ m, laminar case.

The turbulent case

Also this mesh has been created with the grid generator from the University of Swansea [90] and has also been used as the fine grid for the laminar case. It consists of 850,196 nodes (5,048,481 tetrahedra), the initial normal spacing is 10^{-6} m and 38 viscous layers with a stretching factor of 1.2 have been used to guarantee a smooth transition from the viscous layers to the isotropic part of the grid. The surface grid is slightly finer than the laminar grid, 23,187 nodes on the ogive cylinder and 12,758 in the outflow plane.

The convergence history, analytical N-scheme Jacobians and $CFL = 100$ for both the N- and PSI-scheme, for the SST-model is shown in figure 6.44. To meet the memory requirements for the implicit solver, the computation must be done on 64 processors of the Cray T3E and it took approximately 36 CPU-hours. The first-order N-scheme is converging quite well and probably more orders of magnitude can be reached than the approximately 6 orders where it has been stopped. The PSI-scheme converges about 3 orders relative from the restart, which is much more than the laminar solution on this grid, see figure 6.35. As the maximum value of the eddy-viscosity did not change more than 0.1% between iteration 700 and 750, it has been assumed that the solution was converged. Consequently, the turbulent case is steady.

Mach number isolines in the outflow plane for the laminar and turbulent solution are shown in figure 6.45 and 6.46 respectively. Clearly, for the turbulent solution the vortices are located more above the body than for the laminar computation. This is caused by the fact that a turbulent boundary layer can sustain adverse pressure gradients better than a laminar one. consequently, the primary separation line has been shifted in leeward direction of the model - compare fig-

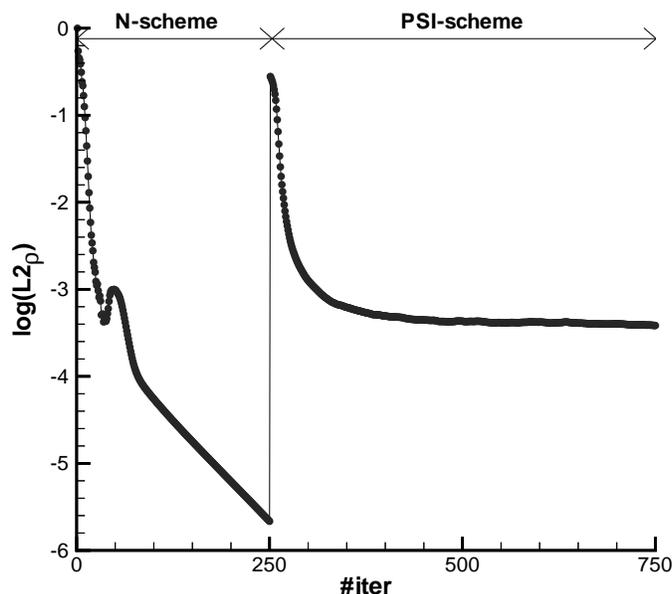


Figure 6.44: Convergence history for the turbulent ogive cylinder, SST-model.

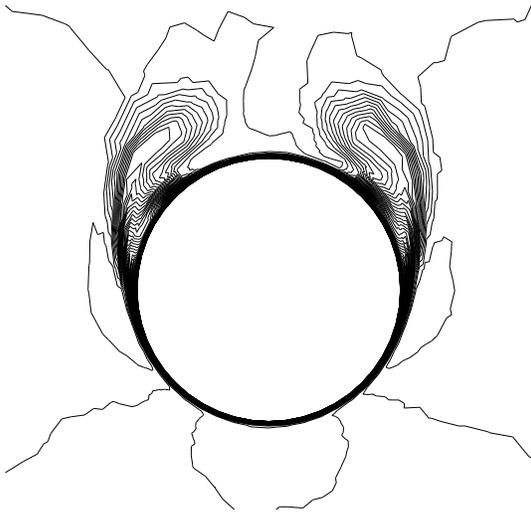


Figure 6.45: Mach number isolines in the outflow plane for the laminar solution, PSI-scheme, min = 0.0, max = 2.08, step = 0.04



Figure 6.46: Mach number isolines in the outflow plane for the turbulent SST solution, PSI-scheme, min = 0.0, max = 2.09, step = 0.04

ures 6.37 and 6.47. The primary separation line is much straighter for the turbulent case, indicating that the solution is more stable than the laminar solution.

Figures 6.48 to 6.53 show the computed and measured surface pressure distributions in several x cross-sections for the PSI-scheme solution in combination with the SST turbulence model. The agreement with the experimental data is good, although there are some slight deviations, especially when the vortex is present, see figures 6.51 and 6.52. This might be a grid effect and in principle computations on a finer grid and with other turbulence models should be performed to verify grid convergence and the effect of the turbulence model. However this has been omitted, because of the large computational effort (36 hours on 64 processors) to obtain a solution for this problem.

Also experimental field data is available in certain x cross-sections [2]. To

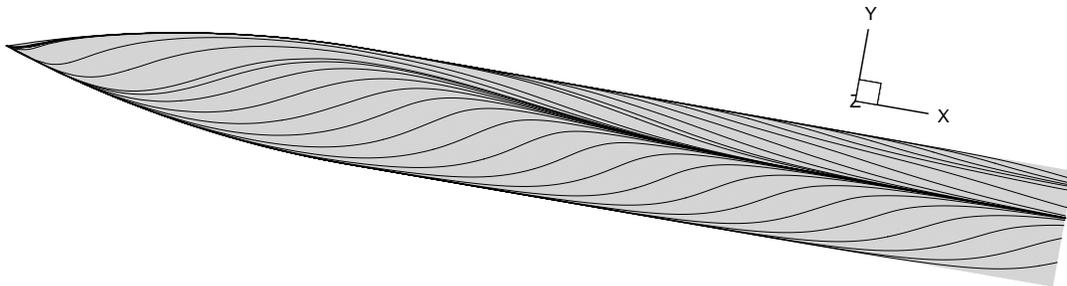


Figure 6.47: Computed skin friction lines for the turbulent ogive cylinder, system PSI-scheme, SST-model.

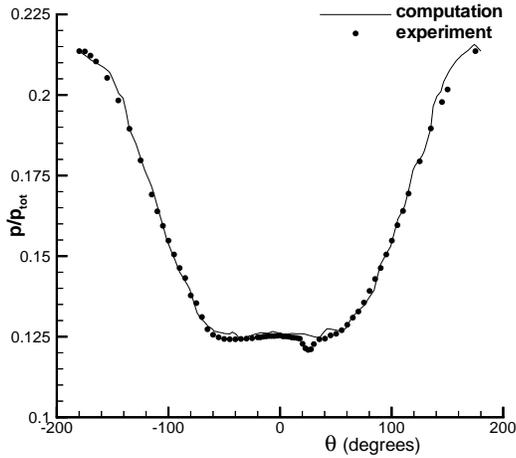


Figure 6.48: Surface pressure distribution, $x = 0.045$ m, turbulent case, SST-model.

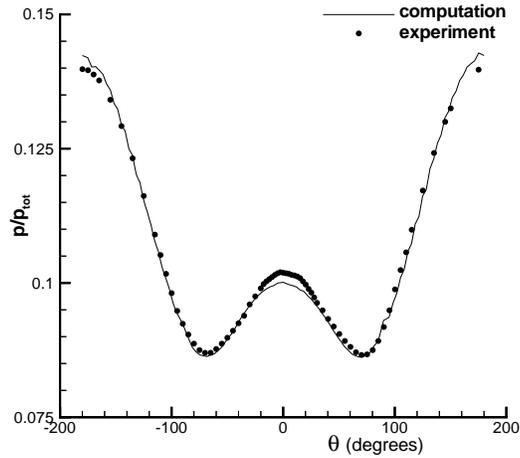


Figure 6.49: Surface pressure distribution, $x = 0.09$ m, turbulent case, SST-model.

compare the position of the primary vortex, p_t isolines are shown in figures 6.54 and 6.55 for both the computation and the experiment. No measurements have been performed in the boundary layer and therefore this layer is not visible in the experimental isolines. From figures 6.54 and 6.55 it is clear that the computation predicts a larger and weaker vortex than the experiments. Also the core is much less pronounced in the computation. The cause for these differences will be a combination of the deficiencies of the turbulence model and the fact that the solution is not grid converged, especially for field data. Indeed the grid outside the boundary layer is rather coarse. As the isotropic part of the turbulent grid is

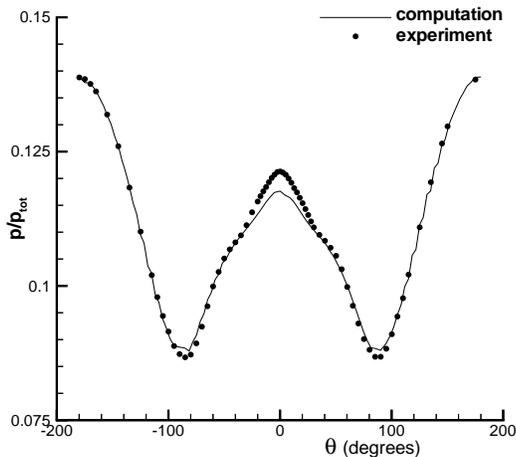


Figure 6.50: Surface pressure distribution, $x = 0.135$ m, turbulent case, SST-model.

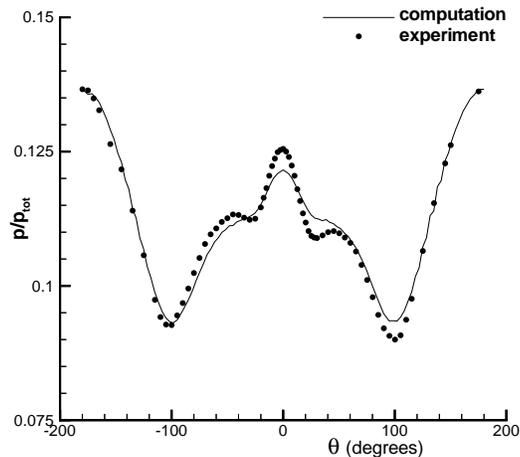


Figure 6.51: Surface pressure distribution, $x = 0.18$ m, turbulent case, SST-model.

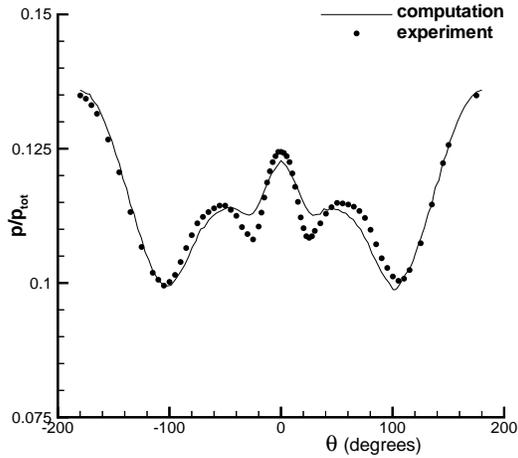


Figure 6.52: Surface pressure distribution, $x = 0.225$ m, turbulent case, SST-model.

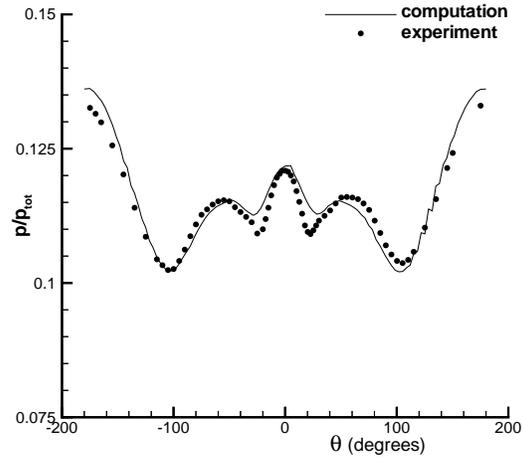


Figure 6.53: Surface pressure distribution, $x = 0.255$ m, turbulent case, SST-model.

almost identical to that of the laminar grid, an indication of the coarseness of the inviscid part of the mesh can be seen in figure 6.33. Consequently, the computed vortex is smeared due to numerical diffusion and becomes weaker.

This test case shows that the multi-dimensional upwind discretization technique is able to solve 3D viscous flows. In combination with the backward Euler time integration method, the algorithm is very stable. Starting from a uniform flow field, also for the turbulent quantities, did not pose any problems for the flow solver. Arguments have been given for the unsteadiness of the laminar case. The

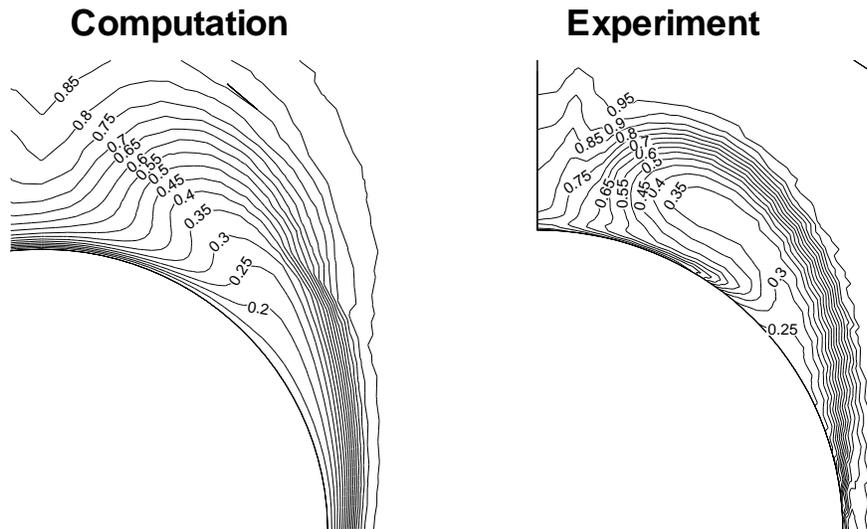


Figure 6.54: Computed and experimental p_t isolines (dimensionless by $p_{t\infty}$) in the cross-section $x = 0.21$ m, SST-model.

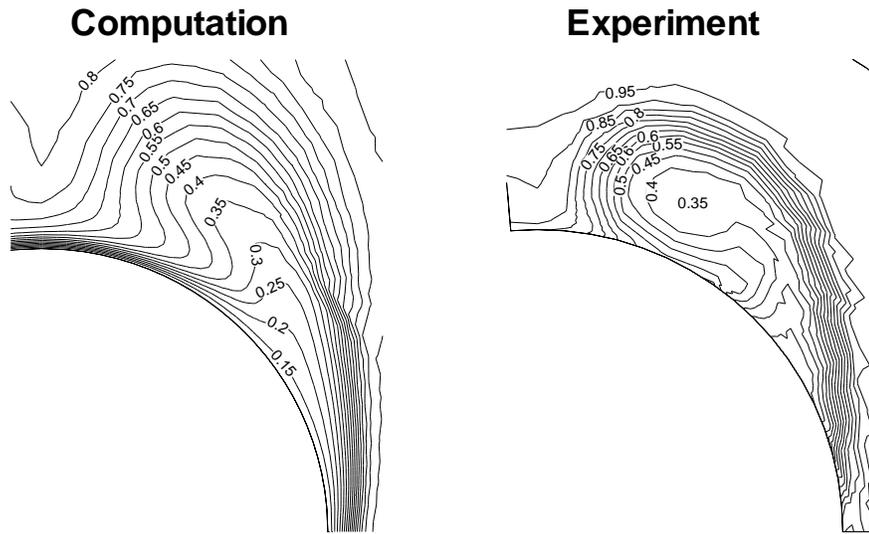


Figure 6.55: Computed and experimental p_t isolines (dimensionless by $p_{t\infty}$) in the cross-section $x = 0.24$ m, SST-model.

solution of the turbulent problem shows good agreement with the experimental surface pressures, although from the turbulence modeling point of view, this flow is simpler than the shock-wave/boundary-layer interaction, section 6.3.2 or the base flow problem, section 6.3.3. The agreement with the measured field data, p_t , is less good, which is caused by a combination of the deficiencies of the turbulence model and the coarseness of the grid outside the boundary layer.

Chapter 7

Conclusions and future prospects

7.1 Conclusions

The general conclusion is that in this work probably an important step has been taken in the development of a robust multi-dimensional upwind discretization technique for non-commuting hyperbolic systems on unstructured grids. It is also clear that the method is not working perfectly yet. The following sections summarize the conclusions of this work.

7.1.1 The spatial discretization

An attempt has been made to extend the previously developed multi-dimensional upwind scalar schemes [118, 119, 116, 84, 111] to non-commuting hyperbolic systems in general and to those representing the Euler equations in particular. It has been shown that the generalization of linear schemes, e.g. N-scheme, is straightforward and that the discretization based on them is very robust even for high aspect-ratio triangles, traditionally the problem area for "standard" upwind algorithms on unstructured grids. Positivity of the system N-scheme can only be proved for linear hyperbolic systems which can be symmetrized, see appendix B. Fortunately most systems which model a physical conservation law possess the symmetrizing property, although they are not linear.

The generalization of the nonlinear schemes, which combine second-order accuracy with monotonicity, is more difficult. The present form of the system PSI-scheme, see section 3.3.2, gives acceptable results, but its convergence behavior is far from optimal. Also from the theoretical point of view the presented system PSI-scheme is not entirely correct. Consider for example a supersonic flow for which all eigenvalues of the generalized upwind parameter K_l are negative for the most upwind node. Consequently the matrix K_l^+ is identical to zero and it is to be expected that the corresponding distribution matrix of the N-scheme, β_l^N , which is needed for the limiting procedure (3.20), is zero as well. However in the current formulation this is not the case, see equation (3.29), and consequently the number of zero eigenvalues of β_l^N and of K_l^+ differ. Ideally one would like to have

these numbers equal, but for all the attempted formulations with this property, the matrix $\left(\sum_m \beta_m^{N^+}\right)$, see equation (3.20), could become singular and the system PSI-scheme was not defined. Note that, although in the current formulation β_l^{PSI} is not zero if K_l^+ is zero, the limited scheme is still consistent, *i.e.* the product of β_l^{PSI} with R_T is zero. This is caused by the sequence of the matrix multiplications in equation (3.20).

7.1.2 Time integration and parallelization

Most results presented in the chapters 5 and 6 have been computed with the fully implicit backward Euler time integrator. Except for fully supersonic, inviscid flow, see section 5.4.2, this method reduces the CPU-time by at least an order of magnitude compared to explicit time integration schemes. This is especially true for turbulent flows. However the large amount of memory required for 3D computations, section 4.2.3, forms a serious bottleneck and therefore it is worthwhile to consider other iterative methods to obtain the steady-state solution, see section 7.2.2.

The memory requirement is partially fulfilled by using multiple processors in the computation. As the stencil of the spatial discretization only involves the nearest neighbors, the parallelization algorithm is rather easy to implement. The use of the AZTEC library [50], which has been especially designed for the parallel solution of linear systems arising in finite-element type problems, simplifies it even further, because the communication pattern and the local node renumbering is done by this library and the "only" task left for the user is the proper definition of the Jacobian matrix. The speed-up's up to 32 processors on a quite coarse grid, 10,420 nodes, are satisfactory.

The linear systems arising in the backward Euler method are solved with Krylov subspace methods, like GMRES, BiCGstab etc., all available in AZTEC. Although these algorithms perform independently of the bandwidth of the matrix, it was found beneficial to minimize this bandwidth, because the efficiency of the BILU and BMILU preconditioners depends very much on this bandwidth. In this way, the CPU-time could be reduced with a factor two for certain two-dimensional problems.

7.1.3 Euler results

The solutions presented in chapter 5 clearly show the gain in accuracy if the spatial discretization is applied to the preconditioned Euler equations instead of the standard Euler equations. The reason is that by means of preconditioning the Euler equations are written in their optimal decoupled form, which makes it possible to use the optimal scalar PSI-scheme instead of the "unfinished" system PSI-scheme.

For a subcritical flow, section 5.4.1, the best results, in terms of spurious entropy production, were obtained if the nonlinear, monotonic PSI-scheme was used for the decoupled scalar equations, *i.e.* entropy and total enthalpy, and the Lax-Wendroff system scheme for the acoustic subsystem. This is quite surprising, because it is well-known that nonlinear second-order schemes are more diffusive than linear ones. Apparently the extra diffusion of the scalar PSI-scheme is beneficial for the entire solution.

For the supersonic channel flow, section 5.4.2, fine grids are needed to capture all the physical phenomena involved in a complex shock system. This example really showed the need for solution adaptive grid techniques as described in [20] to obtain a grid-converged solution with an acceptable number of nodes. Although the coarse grid solution cannot be considered as grid converged (the weak slip line is not predicted), the Mach number distributions in the cross-sections agree remarkably well with the finer grids and the method of characteristics. The entropy distributions for the coarse, medium and fine grid showed that the correct prediction of this variable is much more difficult than the Mach number. Furthermore, slight overshoots have been observed, although these results should be interpreted with care, because interpolated solutions have been shown and not the actual values in the nodes.

The comparison of the computed inviscid pressure distribution for the M6 wing, section 5.4.3, shows a fairly good agreement with the experiments. The location of the shock, which originates from the root leading edge, is predicted accurately. Due to the omission of the viscous terms, the second shock is computed too far downstream, the usual error of a conservative Euler solution. The largest deviation between computation and experiment occurs at the wing tip, because the grid in this part of the flow field is too coarse to capture the tip vortex correctly. A comparison with a standard structured grid finite-volume code shows that the accuracy of the two algorithms is comparable, which is a distinct improvement, as standard upwind algorithms on unstructured grids are known to be more diffusive than their structured grid counterparts. Both the solution of the unstructured multi-dimensional upwind algorithm and the solution of the structured finite-volume solver show the difficulties that both algorithms have in order to solve the leading edge correctly. Grid refinement is one way to obtain better results in this region. Another possibility would be to use a combination of Lax-Wendroff/PSI in the stagnation region, as this has been shown to reduce the spurious entropy production in the 2D case. The amount of CPU-time needed to compute the solution on a 275,000 node grid, $2\frac{1}{2}$ hours for the standard Euler equations and 8 hours for the preconditioned equations both on 32 processors of the Cray T3E, indicates that the fully implicit method is reasonably efficient for the nonlinear system schemes, but improvements can certainly be made.

7.1.4 Navier-Stokes results

It has been shown that the Petrov-Galerkin finite-element interpretation of the residual distribution schemes allows a straightforward extension of the discretization technique to the Navier-Stokes equations. For linear triangles and tetrahedra this effectively results in a Galerkin discretization for the viscous terms.

The laminar solution for the hyperboloid flare, section 6.3.1, agrees well with the experiments and probably the experimental uncertainty is larger than the error in the numerical results. Also this test case was used to compare the accuracy of the present unstructured grid algorithm with a standard finite-volume method on structured grids. The multi-dimensional upwind method seems slightly more accurate, but certainly comparable with the structured grid algorithm, just as for the M6-wing.

The laminar results for the ogive cylinder, section 6.3.4, are somewhat disappointing. Although the coarse grid solution showed some agreement with the experiments, the finer grid predicted an "unsteady", asymmetric solution. As the experimental results give a symmetric solution it is to be expected that the flow in the experiments is of the transitional type. This expectation is confirmed first by the observation that the computed and measured pressure distributions at the first two cross-sections agree quite well, but further downstream the deviation between computation and experiment becomes larger, secondly that the turbulent solution on the same grid converges two orders of magnitude more than the laminar solution, and that the primary separation line shows a wavy behavior, which is an indication that the solution is unstable.

The modeling of turbulent flows has been done with the Reynolds-Averaged Navier-Stokes equations, which must be closed by a suitable turbulence model. The one- and two-equation models considered in this thesis are discretized in a consistent manner with the mean flow equations. Only the turbulent source terms are treated in a special way, see section 6.2.1, such that a stable numerical method is obtained. The consequence of this treatment, which can be interpreted as an automatic under-relaxation, is that it is impossible to obtain Newton convergence for turbulent problems and clearly stability has been preferred over efficiency.

For the turbulent shock-wave/boundary-layer interaction, section 6.3.2, the SST model clearly outperforms the $k - \omega$, the BSL and the Spalart-Allmaras model for the wall pressure distributions, although also the SST result does not exactly fit the experiments. However, the computed velocity profiles in and after the recirculation region are far off from the measured ones and the SST-model does certainly not perform better than the other models for this quantity.

The solution on the fine grid showed a few wiggles in the Mach number behind the normal shock. These wiggles, which are also present in the first order N-scheme solution, are probably caused by the very sharp shock-capturing of the method, typically zero or one internal node. On coarser grids the wiggles are less pronounced, because of more numerical cross-wind diffusion. This phenomena is also present in the Roe scheme [91], which is identical to the multi-dimensional

upwind scheme in one space dimension.

The most difficult test case discussed in this thesis, both from a numerical and a fluid dynamical point of view, is definitely the prediction of the base flow characteristics of the axisymmetric rocket model with exhaust plume, section 6.3.3. The complicated flow in this part, four counter-rotating vortices, combined with the low Mach number, makes the problem very difficult to solve and extremely fine meshes are needed to obtain a grid converged solution - even the presented results of the two-equation models on the triangulated 925×481 structured grid in the base region might not be grid converged.

All turbulence models consistently underpredict the values of the base pressure - the best numerical result ($k - \omega$ model) gives a pressure, which is approximately 90 % of the experimental value. The result of the $k - \omega$ model is significantly better than the results obtained with the BSL and SST-models. The Spalart-Allmaras model predicts an even lower base pressure than these two-equation models. However, almost a factor two is saved in terms of CPU-time for this one-equation model, and it has been shown to be grid-converged - something that could not be shown for the two-equation models. The conclusion for the turbulence models considered is that the Spalart-Allmaras model is not able to predict the physical phenomena in the base region correctly, while especially the $k - \omega$ model gives a surprisingly good result. However extremely fine grids are needed to obtain this result, which makes it practically not applicable. Furthermore, other cases should be computed in order to find out whether this result is a coincidence or not.

Numerically, the supersonic expansion from the cylindrical part of the model to the base region turned out to be a problem for the discretization technique in combination with the explicit Runge-Kutta time integrators, which could not be overcome. Even when a fully converged, laminar, first-order solution was used as starting solution, the code blew up in a few iterations. The actual number depended on the selected CFL number, but it always failed - even in the case $CFL = 10^{-5}$ it is not stable. On the other hand, the fully implicit approach is able to give results, although the maximum CFL number must be restricted to 20 for stability reasons. These stability problems are not only caused by the base region, but also by the very high aspect ratio cells (± 4000), which are needed because of the high Reynolds number of the flow. The total amount of CPU-time for the two-equation models, 19 hours on 16 processors of the Cray T3E for the fine grid and 100 hours on the same number of processors for the triangulated 925×481 structured grid in the base region, is very high for an axisymmetric problem.

The turbulent solution (SST model) of the ogive cylinder, section 6.3.4, shows that the discretization technique developed in this thesis can be applied to 3D turbulent flows without any problem. Comparison with experimental surface pressure distributions in several crossflow planes shows good agreement - only some slight deviations were found, especially in the regions where the main vortices are present. These deviations are caused by a combination of a grid resolution problem (the

grid may be too coarse in those particular regions) and the turbulence model. Due to the large memory requirement, a finer grid could not be used and a grid convergence study could not be performed. The comparison with the experimental field data (total pressure) shows that the grid outside the boundary layer is indeed too coarse for an accurate prediction of the vortices. Consequently, the computed total pressure distributions in several cross-sections differ considerably from the measured values. As far as the turbulence model is considered, this flow does not contain adverse pressure gradients and it is to be expected that the performances of the $k - \omega$, BSL and SST models are very similar.

The CPU-time of 36 hours on 64 processors of the Cray T3E is more or less standard for a 3D turbulent flow field computation. If the results of the 2D computations can be extrapolated to three space dimensions, the Spalart-Allmaras solution can be obtained in approximately 20 hours on the same number of processors.

7.2 Prospects

7.2.1 The spatial discretization

The main obstacle at the moment for the use of the multi-dimensional upwind method presented in this work is the performance of the nonlinear system PSI-scheme. A more stable version of this scheme has to be developed in order to make the method really competitive with existing finite-volume and finite-element methods. It is possible that the limiting approach presented in section 3.3.2 is not the correct way to obtain second-order accuracy while keeping the monotonicity property. A different approach for system schemes has been proposed by Sidilkover [109, 108, 110], although his method is neither continuous nor invariant for similarity transformations. Furthermore, if the basic first-order scheme is the N-scheme, his method is not straightforward to apply. However the convergence behavior is much better than the theoretically more consistent method presented in section 3.3.2 of this work.

Another future extension of the spatial discretization can be the use of hybrid grids to combine the accuracy of structured grids with the geometrical flexibility of unstructured grids. Indeed it has been shown in the introduction that the multi-dimensional upwind discretization on quadrilaterals is even more accurate than on triangles.

7.2.2 The nonlinear iterative solver

The fully backward Euler method with full storage of the Jacobian matrix is very memory consuming. A reduction in the required memory is given by the Jacobian free method, where the Jacobian matrix $\frac{\partial R}{\partial U}$ is not stored [51]. This can be done, because in the Krylov subspace algorithms [106, 105, 124, 112] $\frac{\partial R}{\partial U}$ always occurs

in a matrix-vector product. This matrix-vector product can be interpreted as a Fréchet derivative, which can be approximated by a finite-difference formulation [51]. However this approach, which must be repeated in every Krylov step, is about four times as expensive as an ordinary matrix-vector product and therefore only feasible if just a few Krylov iterations are used. Of course the Jacobian matrix is not built, which saves CPU-time compared to full storage. The memory requirements can be reduced by a factor 10 or more (depending on the number of governing equations) in three space dimensions.

At first glance the Jacobian free method looks very attractive, but there are two disadvantages. First, as a finite-difference method is used to approximate the matrix-vector product, the corresponding matrix is the exact Newton matrix, which has been shown to be less stable than the Picard matrix. This can particularly become an issue in the first few iterations for problems with moving shocks. It is probably possible to approximate the Picard matrix in the Jacobian free method, but this will certainly require some tricky programming. The second disadvantage is of a more fundamental nature. In section 4.2.2 it has been explained that the Krylov method must be used in combination with a good preconditioner to obtain an efficient algorithm. As the Jacobian is not stored in the matrix free method, it is impossible to use the BILU and BMILU preconditioners. The Block Jacobi preconditioner, based for example on the Picard matrix, would be a possibility, because it only requires the additional storage of a block diagonal, which is acceptable. However, the performance of this preconditioner is only acceptable for diagonal dominant matrices and consequently only low CFL numbers can be used (1-5).

A better option would be a multi-level preconditioner [51], where information on coarser grid levels is used to approximate the fine grid solution. Currently at Sandia both the Jacobian free method and multi-level preconditioners are being implemented in the AZTEC library.

A completely different approach to speed up convergence would be full multi-grid. This has already been done for unstructured grids [51, 72, 71] and results are encouraging. Probably semi-coarsening and wall functions on the coarser grid levels must be used in order to make the method suitable for high Reynolds number, turbulent problems, but this should be possible. Work in this direction has already started in the framework of a Brite-Euram program, starting from the software developed in this thesis.

7.2.3 Euler/Navier-Stokes solutions

Solution adaptive grid algorithms will be essential to obtain grid converged solutions in a reasonable amount of time, especially for transonic and supersonic flow fields with complex shock-wave/boundary-layer and shock-shock interactions. In two space dimensions a lot of work has been done by Carette [20], who showed that for inviscid computations solution adaptation of the grid is feasible, but for viscous computations adaptive regriding is to be preferred.

To improve the turbulence modeling, it may be worthwhile to implement second-order closure models. Although numerically much more difficult to handle than the one- and two-equation models used in this thesis, the method described in section 6.2.1 should be able to deal with this. There are even plans [32] to start LES with the current discretization technique. Due to limitations in computer resources LES will be restricted to two space dimensions and assuming a periodic behavior in the third, but it will be very interesting to see if the current method is accurate enough.

If this is not, higher-order methods should be considered. In the finite-element framework the Galerkin Least Squares (GLS) method [80, 81] offers this possibility. This GLS method is identical to the SUPG method on linear triangles and tetrahedra, but is slightly different on higher-order elements. The design of a higher-order, monotonic method in the multi-dimensional upwind framework is extremely difficult, even for scalar advection.

Another issue is viscous preconditioning. As shown in chapter 5, preconditioning improves the quality of the spatial discretization. However the Euler preconditioner used in that chapter does not work for the Navier-Stokes equations and also in this area a lot of work remains to be done.

Appendix A

Existence proof of $\left(\sum_m K_m^-\right)^{-1}$

The system versions of the, N-, PSI- and LDA-scheme require the existence of the matrices $\left(\sum_m K_m^-\right)^{-1}$ and $\left(\sum_m K_m^+\right)^{-1}$, see chapter 3. This proof is quite straightforward.

Equations (2.7) and (3.2) give:

$$\sum_m K_m = 0, \tag{A.1}$$

Together with the splitting of K_l in K_l^+ and K_l^- ,

$$K_l = K_l^+ + K_l^-, \tag{A.2}$$

this results in:

$$\sum_m K_m^- = -\sum_m K_m^+ = \frac{1}{2} \sum_m |K_m|, \tag{A.3}$$

where $|K_l|$ is the absolute value of K_l . As $|K_l|$ is at worst non-negative, $\sum_m |K_m|$

is only singular if all $|K_l|$'s have at least one eigenvalue zero. However this would mean that all matrices A_i 's in equation (3.1) have an eigenvalue zero and that the corresponding eigenvector is identical for all A_i 's. In practice these conditions never occur and even if they would occur, this equation can be decoupled from the system and the system scheme could be used on the remaining part. Consequently the

matrices $\left(\sum_m K_m^-\right)^{-1}$, $\left(\sum_m K_m^+\right)^{-1}$ and $\left(\sum_m |K_m|\right)^{-1}$ exist for any hyperbolic system.

Furthermore, the matrices $\sum_m K_m^+$ and $\sum_m |K_m|$ have strictly positive and $\sum_m K_m^-$ strictly negative eigenvalues.

Appendix B

Energy-stability of the N-scheme

This appendix contains the energy-stability proof of the system N-scheme. Barth [14] proved this property and it is included in this thesis for completeness. The proof is given for two space dimensions, but the extension to 3D is trivial.

Consider the hyperbolic system

$$\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial x} + B \frac{\partial U}{\partial y} = 0. \quad (\text{B.1})$$

In this appendix it is assumed that the Jacobians A and B are symmetric, possibly through the introduction of entropy variables, see appendix C. Friedrichs [41] showed that in this case the L^2 norm of a solution, also called the energy functional, is of bounded growth, *i.e.*

$$\frac{\partial}{\partial t} \left(\iint_{\Omega} U^T U d\Omega \right) \leq ct, \quad (\text{B.2})$$

where ct is a constant. It is desirable that the numerical solution possesses a discrete analogue of this property.

The discretization of equation (B.1) in operator form is:

$$D \frac{dU}{dt} + \mathcal{L}(U) U = 0, \quad (\text{B.3})$$

where D is the approximate mass-matrix, see equation (2.12) and $\mathcal{L}(U)$ the spatial operator of the scheme considered. Using the relation

$$\frac{d(U^T D U)}{dt} = \frac{dU^T}{dt} D U + U^T D \frac{dU}{dt} \quad (\text{B.4})$$

and the fact that the transpose of equation (B.3) is given by

$$\frac{dU^T}{dt} D + U^T \mathcal{L}^T(U) = 0, \quad (\text{B.5})$$

results in the following equation for the discrete energy functional:

$$\frac{1}{2} \frac{d(U^T D U)}{dt} + U^T L U = 0. \quad (\text{B.6})$$

Here L is the spatial energy operator, defined as

$$L = \frac{1}{2} (\mathcal{L}(U) + \mathcal{L}^T(U)). \quad (\text{B.7})$$

A numerical scheme is called energy stable if the operator L is positive semidefinite, *i.e.* $U^T L U \geq 0$ and thus $\frac{d(U^T D U)}{dt} \leq 0$. As D is a diagonal matrix with positive coefficients, the addition of the mass-matrix in the formulation does not matter.

The operators \mathcal{L} and L are grid dependent and therefore it is more logical to look at the local operators at the element level, indicated by \mathcal{L}^e and L^e . As the global energy operator is the sum of the local ones, a sufficient condition for positive semidefiniteness of L is that L^e is positive semidefinite. Note that this condition is more restrictive than positive semidefiniteness of L itself.

For the N-scheme distribution, equation (3.17), \mathcal{L}^e is given by:

$$\mathcal{L}^e(U) = \begin{bmatrix} K_1^+ + K_1^+ N^+ K_1^- & K_1^+ N^+ K_2^- & K_1^+ N^+ K_3^- \\ K_2^+ N^+ K_1^- & K_2^+ + K_2^+ N^+ K_2^- & K_2^+ N^+ K_3^- \\ K_3^+ N^+ K_1^- & K_3^+ N^+ K_2^- & K_3^+ + K_3^+ N^+ K_3^- \end{bmatrix}, \quad (\text{B.8})$$

where

$$N^+ = \left(\sum_p K_p^+ \right)^{-1} = - \left(\sum_p K_p^- \right)^{-1}. \quad (\text{B.9})$$

In appendix A it has been proved that N^+ is positive definite. The local energy operator becomes:

$$L^e = \frac{1}{2} \begin{bmatrix} 2K_1^+ + K_1^+ N^+ K_1^- + K_1^- N^+ K_1^+ & K_1^+ N^+ K_2^- + K_1^- N^+ K_2^+ & K_1^+ N^+ K_3^- + K_1^- N^+ K_3^+ \\ K_2^+ N^+ K_1^- + K_2^- N^+ K_1^+ & 2K_2^+ + K_2^+ N^+ K_2^- + K_2^- N^+ K_2^+ & K_2^+ N^+ K_3^- + K_2^- N^+ K_3^+ \\ K_3^+ N^+ K_1^- + K_3^- N^+ K_1^+ & K_3^+ N^+ K_2^- + K_3^- N^+ K_2^+ & 2K_3^+ + K_3^+ N^+ K_3^- + K_3^- N^+ K_3^+ \end{bmatrix} \quad (\text{B.10})$$

where use has been made of the relation:

$$(K_l^+ N^+ K_m^-)^T = K_m^- N^+ K_l^+, \quad (\text{B.11})$$

which is valid because the system (B.1) is symmetric.

First consider the scalar case. According to Gerschgorin, L^e is positive semidefinite if:

- 1 Diagonal terms ≥ 0 .
- 2 Off-diagonal terms ≤ 0 .
- 3 Sum of row/column elements ≥ 0 .

L^e clearly obeys the first and second criterion, but it fails the third, as the sum of the elements of row l is $\frac{1}{2}K_l$, which can be both positive and negative. Consequently L^e does not seem to be positive semidefinite. However, the spatial operator \mathcal{L}^e can be modified as follows:

$$\mathcal{L}^{e'} = \mathcal{L}^e - \frac{1}{2} \begin{bmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{bmatrix}. \quad (\text{B.12})$$

For constant Jacobians A and B in equation (B.1), the added term is identical to zero when accumulated over all triangles belonging to the neighborhood Ω_l , see figure 2.3. For non-constant Jacobians this is not the case and for such problems the analysis is not valid from here on. However, it might still give a good indication.

The modified energy operator based on $\mathcal{L}^{e'}$ becomes:

$$L^{e'} = \frac{1}{2} \begin{bmatrix} |K_1| + K_1^+ N^+ K_1^- + K_1^- N^+ K_1^+ & K_1^+ N^+ K_2^- + K_1^- N^+ K_2^+ & K_1^+ N^+ K_3^- + K_1^- N^+ K_3^+ \\ K_2^+ N^+ K_1^- + K_2^- N^+ K_1^+ & |K_2| + K_2^+ N^+ K_2^- + K_2^- N^+ K_2^+ & K_2^+ N^+ K_3^- + K_2^- N^+ K_3^+ \\ K_3^+ N^+ K_1^- + K_3^- N^+ K_1^+ & K_3^+ N^+ K_2^- + K_3^- N^+ K_2^+ & |K_3| + K_3^+ N^+ K_3^- + K_3^- N^+ K_3^+ \end{bmatrix} \quad (\text{B.13})$$

Now the row/column sum is identically zero and $L^{e'}$ is positive semidefinite for the scalar case.

For the system case the proof is more complicated. After some manipulations $L^{e'}$ can be rewritten as:

$$L^{e'} = \frac{1}{2} \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} [N^+] \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix}^T + L^+ + L^-, \quad (\text{B.14})$$

where

$$L^+ = \frac{1}{2} \begin{bmatrix} K_1^+ & 0 & 0 \\ 0 & K_2^+ & 0 \\ 0 & 0 & K_3^+ \end{bmatrix} - \frac{1}{2} \begin{bmatrix} K_1^+ \\ K_2^+ \\ K_3^+ \end{bmatrix} [N^+] \begin{bmatrix} K_1^+ \\ K_2^+ \\ K_3^+ \end{bmatrix}^T \quad (\text{B.15})$$

and

$$L^- = \frac{1}{2} \begin{bmatrix} -K_1^- & 0 & 0 \\ 0 & -K_2^- & 0 \\ 0 & 0 & -K_3^- \end{bmatrix} - \frac{1}{2} \begin{bmatrix} -K_1^- \\ -K_2^- \\ -K_3^- \end{bmatrix} [N^+] \begin{bmatrix} -K_1^- \\ -K_2^- \\ -K_3^- \end{bmatrix}^T. \quad (\text{B.16})$$

$L^{e'}$ is positive semidefinite if each of the three suboperators in equation (B.13) is positive semidefinite. For the first suboperator the proof is trivial, as it is obvious that

$$\frac{1}{2} V^T \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} [N^+] \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix}^T V \geq 0, \quad \forall V. \quad (\text{B.17})$$

The proof for L^+ and L^- is less straightforward and the so-called Golub lemma is used.

Lemma: The matrix

$$L = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{bmatrix} - \begin{bmatrix} A \\ B \\ C \end{bmatrix} N \begin{bmatrix} A \\ B \\ C \end{bmatrix}^T, \quad N = (A + B + C)^{-1} \quad (\text{B.18})$$

is positive semidefinite for all $A, B, C \in \mathcal{R}^{n \times n}$ symmetric positive definite.

Proof: Define

$$Z = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{bmatrix} \quad (\text{B.19})$$

and the transformation

$$Z^{-\frac{1}{2}} L Z^{-\frac{1}{2}} = \begin{bmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & I_n \end{bmatrix} - \begin{bmatrix} A^{\frac{1}{2}} \\ B^{\frac{1}{2}} \\ C^{\frac{1}{2}} \end{bmatrix} N \begin{bmatrix} A^{\frac{1}{2}} \\ B^{\frac{1}{2}} \\ C^{\frac{1}{2}} \end{bmatrix}^T = I_{3n} - M. \quad (\text{B.20})$$

Here the abbreviation M has been introduced for convenience. The eigenvalues of M can be determined by using a standard result for Kronecker products: let $u, v \in \mathcal{R}^{n \times m}$, $m \leq n$, then the m nonzero eigenvalues of the Kronecker product $u \otimes v^T$ equal the eigenvalues of $v^T u$. For the matrix M this implies that

$$\begin{aligned} \text{Eigenvalues } M &= \text{Eigenvalues} \left(N^{\frac{1}{2}} (A + B + C) N^{\frac{1}{2}} \right) + 2n \text{ zeros} \\ &= \text{Eigenvalues} (I_n) + 2n \text{ zeros.} \end{aligned} \quad (\text{B.21})$$

Consequently the eigenvalues of $I_{3n} - M$, equation (B.20), are zeros and ones and thus $I_{3n} - M$ is positive semidefinite. This implies that

$$V^T (I_{3n} - M) V \geq 0 \quad \forall V. \quad (\text{B.22})$$

Rewriting this equation to

$$V^T Z^{\frac{1}{2}} Z^{-\frac{1}{2}} (I_{3n} - M) Z^{-\frac{1}{2}} Z^{\frac{1}{2}} V = V^T \left(Z^{\frac{1}{2}} \right)^T L Z^{\frac{1}{2}} V \geq 0 \quad \forall V, \quad (\text{B.23})$$

where use has been made of the fact that $Z^{\frac{1}{2}}$ is symmetric, and introducing $\tilde{V} = Z^{\frac{1}{2}} V$, gives:

$$\tilde{V}^T L \tilde{V} \geq 0 \quad \forall \tilde{V}. \quad (\text{B.24})$$

As $Z^{\frac{1}{2}}$ is invertible, \tilde{V} is arbitrary. Equation (B.24) clearly shows that L is positive semidefinite.

End proof.

In case A , B and C , equation (B.18), are positive semidefinite, the lemma is still valid by considering the perturbed matrices $A_\varepsilon = A + \varepsilon I$, $B_\varepsilon = B + \varepsilon I$ and $C_\varepsilon = C + \varepsilon I$ and letting $\varepsilon \rightarrow 0$.

Back to the problem of the positive semidefiniteness of L^e . It is clear that L^+ , equation (B.15), and L^- , equation (B.16), are matrices of the form given in equation (B.18), and thus all three suboperators in equation (B.14) are positive semidefinite. Consequently L^e is positive semidefinite as well.

For the PSI-scheme energy stability cannot be shown, even not for the scalar case [14].

Appendix C

Symmetrizability of the Euler equations

From appendix B it is clear that the energy stability, and thus the positivity, of the N-scheme heavily relies on the fact that the system B.1 is symmetric. In this appendix it is shown that a sufficient condition is that the system can be symmetrized. The Euler equations are given as an example.

The conclusions from Harten [45] are summarized. Consider the hyperbolic system:

$$\frac{\partial U}{\partial t} + A_i \frac{\partial U}{\partial x_i} = 0. \quad (\text{C.1})$$

Assume that for this system a convex entropy function $\mathcal{S}(U)$ exists, such that

$$U^T = \frac{\partial \mathcal{S}}{\partial V}, \quad (\text{C.2})$$

where V is a set of variables with the same number of elements as U . The system (C.1) is called symmetrizable if the function \mathcal{S} exists.

It can be proved that then in the transformed system

$$B_0 \frac{\partial V}{\partial t} + B_i \frac{\partial V}{\partial x_i} = 0, \quad (\text{C.3})$$

the matrix $B_0 = \frac{\partial U}{\partial V}$ is symmetric positive definite and the matrices $B_i = A_i B_0$ are symmetric. The set of variables V , also called entropy variables, is given by:

$$V^T = \frac{\partial \mathcal{S}}{\partial U}, \quad (\text{C.4})$$

Furthermore, the similarity transformation $\partial W = B_0^{-\frac{1}{2}} \partial U$, which transforms equation (C.1) into

$$\frac{\partial W}{\partial t} + C_i \frac{\partial W}{\partial x_i} = 0, \quad C_i = B_0^{-\frac{1}{2}} A_i B_0^{\frac{1}{2}}, \quad (\text{C.5})$$

also results in a symmetric system.

As the eigenvalues of the energy operator of the discretization of the symmetric system (C.5) are identical to the eigenvalues of the same operator applied to equation (C.1), a sufficient condition for energy stability of the N-scheme is that the system (C.1) can be symmetrized.

For the Euler equations an infinite number of entropy functions \mathcal{S} exists, see [45, 9]. Hughes *et al.*[49] showed that if it is required that \mathcal{S} also symmetrizes the Navier-Stokes equations, this arbitrariness is reduced to the requirement that \mathcal{S} should be a linear function of the thermodynamic entropy $s = \ln\left(\frac{p}{\rho^\gamma}\right)$. The choice

$$\mathcal{S} = -\frac{\rho s}{\gamma - 1} \tag{C.6}$$

results in the entropy variables

$$V = \begin{pmatrix} -\frac{s}{\gamma-1} + \frac{\gamma+1}{\gamma-1} - \frac{\rho E}{p} \\ \frac{\rho u_j}{p} \\ -\frac{\rho}{p} \end{pmatrix}. \tag{C.7}$$

Bibliography

- [1] AGARD Advisory Report No 138. *Experimental Data Base for Computer Program Assesment*. AGARD, 1979.
- [2] AGARD Advisory Report No 303. *A Selection of Experimental Test Cases for the Validation of CFD codes*. AGARD, 1994.
- [3] R. Abgrall. Approximation of the Multidimensional Riemann Problem in Compressible Fluid Mechanics by a Roe Type Method. *SIAM Journal of Numerical Analysis*, 1994. Submitted for publication.
- [4] G. Ashford and K. Powell. An Unstructured Grid Generation and Adaptive Solution Technique for High-Reynolds-number Compressible Flows. In *VKI LS 1996-06, Computational Fluid Dynamics*, 1996.
- [5] H.L. Atkins and C.-W. Shu. Quadrature-free implementation of discontinuous Galerkin methods for hyperbolic equations. Technical Report 96-51, ICASE, 1996.
- [6] B.S. Baldwin and T.J. Barth. A one-equation turbulence transport model for high Reynolds numer wall-bounded flows. 1991. AIAA paper 91-0610.
- [7] B.S. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flows. 1978. AIAA paper 78-0275.
- [8] W.J. Bannink, P.G. Bakker, and E.M. Houtman. FESTIP AEROTHERMODYNAMICS: Experimental Investigation of Base Flow and Exhaust Plume Interaction. Technical Report M-775, Faculty of Aerospace Engineering, Delft, 1997.
- [9] T. Barth. Freiburg Lectures on Theory and Numerics for Conservation Laws. 1997.
- [10] T. Barth and D. Jespersen. The design and application of upwind schemes on unstructured meshes. 1989. AIAA paper 89-0366.
- [11] T.J. Barth. On unstructured grids and solvers. In *VKI LS 1990-04, Computational Fluid Dynamics*, 1990.

- [12] T.J. Barth. Recent Developments in High Order k-Exact Reconstruction on Unstructured Meshes. 1993. AIAA paper 93-0668.
- [13] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.
- [14] T.J. Barth. Some Working Notes on Energy Analysis of the N-Scheme. 1996. NASA internal report.
- [15] T.J. Barth and P.O. Frederickson. Higher Order Solution of the Euler Equations on Unstructured Grids using Quadratic Reconstruction. 1990. AIAA paper 90-0013.
- [16] A. Bonfiglioli. *Studio di Algoritmi per l'Approssimazione di Sistemi Differenziali Iperbolici; Applicazioni al Calcolo di Flussi Comprimibili e Non Viscosi in Configurazioni Tridimensionali*. PhD thesis, Politecnico di Bari, Italy, 1996.
- [17] G. Bourgois, H. Deconinck, and P.L. Roe. Multidimensional upwind schemes for scalar advection on tetrahedral meshes. In *Proceedings of the 1st European CFD Conference*, Brussels, Belgium, September 1992. Elsevier.
- [18] R. Broglia, M. Manna, G. Degrez, and H. Deconinck. Axisymmetric Navier-Stokes Computation of Two Hyperboloid Flares in Reentry Configuration. in *Proceedings 4th European High Velocity Database Workshop, ESTEC, Noordwijk*, 1994.
- [19] A. Brooks and T. Hughes. Streamline Upwind/Petrov Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.
- [20] J.-C. Carette. *Adaptive Unstructured Mesh Algorithms and SUPG Finite Element Method for High Reynolds Number Compressible Flows*. PhD thesis, Université Libre de Bruxelles, Belgium, 1997.
- [21] J.-C. Carette, H. Deconinck, H. Paillère, and P.L. Roe. Multidimensional upwinding: its relation to finite elements. *Numerical Methods in Fluids*, 20:935–955, 1995.
- [22] L.A. Catalano, M. Napolitano, and H. Deconinck. Optimal Multi-stage Schemes for Multigrid Smoothing of two-dimensional Advection Operators. *Communications in Applied Numerical Methods*, 8, 1992.
- [23] S. Chapman and T.G. Cowling. *The Mathematical Theory of non-uniform Gases, 2nd edition*. Cambridge University Press, 1952.

- [24] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional Systems. Technical Report 97-43, ICASE, 1997.
- [25] G. Dahlquist and A. Bjork. *Numerical Methods*. Englewood Cliffs, NJ Prentice-Hall, 1974.
- [26] H. Deconinck, Ch. Hirsch, and J. Peuteman. Characteristic decomposition methods for the multidimensional Euler equations. In *Lecture Notes in Physics*, volume 264. Springer-Verlag, 1986.
- [27] H. Deconinck and B. Koren, editors. *Euler and Navier-Stokes Solvers Using Multi-dimensional Upwind Schemes and Multigrid Acceleration*, volume 57 of *Notes on Numerical Fluid Mechanics*. Vieweg, 1997. Results of the BRITE/EURAM Projects AERO-CT89-0003 and AER2-CT92-00040, 1989-1995.
- [28] H. Deconinck, P.L. Roe, and R. Struijs. A Multi-dimensional Generalization of Roe's Flux Difference Splitter for the Euler Equations. *Journal of Computers and Fluids*, 22:215–222, 1993.
- [29] H. Deconinck, R. Struijs, G. Bourgois, H. Paillère, and P.L. Roe. Multi-dimensional upwind methods for unstructured grids. In *Unstructured Grid Methods for Advection Dominated Flows*, May 1992. AGARD-R-787.
- [30] H. Deconinck, R. Struijs, G. Bourgois, and P.L. Roe. Compact advection schemes on unstructured grids. In *VKI LS 1993-04, Computational Fluid Dynamics*, 1993.
- [31] H. Deconinck, R. Struijs, G. Bourgois, and P.L. Roe. High resolution shock capturing cell-vertex advection schemes for unstructured grids. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.
- [32] G. Degrez. Private communication. 1998.
- [33] M. Delanaye. *Polynomial reconstruction finite volume schemes for the compressible Euler and Navier-Stokes equations on unstructured adaptive grids*. PhD thesis, Université de Liège, Belgium, 1996.
- [34] J.M Déleroy. Experimental Investigations of Turbulence Properties in Transonic Shock/Boundary Layer Interactions. *AIAA-Journal*, 21(2):180–185, 1983.
- [35] A. Dervieux, B. van Leer, J. Periaux, and A. Rizzi, editors. *Numerical Simulation of Compressible Euler Flows*, volume 26 of *Notes on Numerical Fluid Mechanics*. Vieweg, 1989. Proceedings of the GAMM workshop on the Numerical Simulation of Compressible Euler flows, held at INRIA, Rocquencourt (France), June 1986.

- [36] A. Eberle, A. Rizzi, and E.H. Hirschel. *Numerical Solutions of the Euler Equations for Steady Flow Problems*, volume 34 of *Notes on Numerical Fluid Mechanics*. Vieweg, 1992.
- [37] L.-E. Eriksson. A preconditioned Navier-Stokes Solver for Low Mach Number Flows. In *Proceedings of the 3rd European CFD Conference*, pages 199–205, Paris, France, 1996. Wiley.
- [38] A. Favre. Equation des gaz turbulents compressibles. *Journal de Mécanique*, 4(3), 1965.
- [39] J.P. Frederiksen. A Residual Distribution Method for 3D Low Mach Number Flows. In *VKI Project Report 1996-10*, 1996.
- [40] R.W. Freund. A transpose free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM, Journal on Scientific and Statistical Computing.*, 14, 1993.
- [41] K.O. Friedrichs. Symmetric Hyperbolic Linear Differential Equations. *Communications on Pure and Applied Mathematics*, 7:345–392, 1954.
- [42] M. Giles, W.K. Anderson, and T.W. Roberts. Upwind Control Volumes: a new upwind approach. 1990. AIAA paper 90-0104.
- [43] H. Gilquin, J. Laurens, and C. Rosier. Le problème de Riemann pour la linéarisation des équations de la dynamique des gaz en dimension 2 et 3 d'espace. January 1994. Sixième Séminaire sur les Ecoulements de Fluides Compressibles, CEA Saclay.
- [44] S. Godunov. A difference method for the numerical calculation of discontinuous solutions of the hydrodynamic equations. *Mat. Sbornik.*, 47:271–306, 1959. Translated as JPRS 7225 by U.S. Dept. of Commerce, 1960.
- [45] A. Harten. On the Symmetric Form of Systems of Conservations Laws with Entropy. *Journal of Computational Physics*, 49:151–164, 1983.
- [46] Ch. Hirsch. *Numerical Computation of Internal and External Flows, Vol. I and II*. Wiley, 1990.
- [47] J.O. Hirschfelder, C.F. Curtiss, and R.B. Bird. *Molecular Theory of Gases and Liquids*. Wiley, 1954.
- [48] T.J.R. Hughes and A. Brooks. A multi-dimensional upwind scheme with no crosswind diffusion. *Finite Element Methods for Convection Dominated Flows, ASME*, 34:19–35, 1979.

- [49] T.J.R. Hughes and M. Mallet. A New Finite Element Formulation for Computational Fluid Dynamics: III. The Generalized Streamline Operator For Advective-Diffusive Systems. *Computer Methods in Applied Mechanics and Engineering*, 58:305–328, 1986.
- [50] S.A. Hutchinson, J.N. Shadid, and R.S. Tumiaro. Aztec User's Guide: Version 1.1. Technical Report SAND95-1559, Sandia National Laboratories, 1995.
- [51] E. Issman. *Implicit Solution Strategies for Compressible Flow Equations on Unstructured Grids*. PhD thesis, Université Libre de Bruxelles, Belgium, 1997.
- [52] A. Jameson. Artificial Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows. 1993. AIAA paper 93-3359.
- [53] A. Jameson. Essential Elements of Computational Algorithms for Aerodynamic Analysis and Design. Technical Report 97-68, ICASE, 1997.
- [54] A. Jameson, T.J. Baker, and N.P. Weatherill. Calculation of Inviscid Transonic Flow over a Complete Aircraft. 1986. AIAA paper 86-0103.
- [55] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. 1981. AIAA paper 81-1259.
- [56] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, 1987.
- [57] C. Johnson. Finite element methods for flow problems. In *Proceedings of the AGARD-VKI Lecture Series on Unstructured Grid Methods for Advection Dominated Flows*, 1992. AGARD Report 787.
- [58] C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 45:285–312, 1984.
- [59] C. Johnson, A.H. Schatz, and L.B. Wahlbin. Crosswind smear and pointwise errors in streamline diffusion finite element methods. *Math. Comput.*, 47:25–38, 1987.
- [60] C. Johnson, A. Szepessy, and P. Hansbo. On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws. *Mathematics of Computation*, 54:107–129, 1990.

- [61] G. Karypis and V. Kumar. Metis, Unstructured Graph Partitioning and Sparse Matrix Ordering System, v.2.0. Technical report, University of Minnesota, Department of Computer Science, 1995.
- [62] P. Krogmann. Hyperboloid-Flare experiments at Mach 6.8 in RWG, Technical Note HT-TN-0-2074-DLR. Technical report, DLR, 1994.
- [63] P. Krogmann. Calibration and Validation Measurements in the Ludwig Tube Tunnel B (RWG), ESA/CNES HT-TR-E-1-301-DGLR. Technical report, DLR, 1996.
- [64] B.E. Launder and D.B. Spalding. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3:269–289, 1974.
- [65] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Verlag, 1990.
- [66] H.W. Liepmann and A. Roshko. *Elements of Gasdynamics*. John Wiley & Sons, Inc., 1957.
- [67] J. W-H Liu and A. George. *Computer Solution of Large Definite Systems*. Prentice-Hall series in computational mathematics Englewood Cliffs, 1981.
- [68] X. Liu and P.D. Lax. Positive Schemes for Solving Multi-dimensional Hyperbolic Systems of Conservation Laws. *CFD Journal*, 5(2):133–156, 1996.
- [69] R.B. Lowrie. *Compact Higher-Order Numerical Methods for Hyperbolic Conservation Laws*. PhD thesis, University of Michigan, USA, 1996.
- [70] R.B. Lowrie, P.L. Roe, and B. van Leer. A space-time discontinuous galerkin method for the time-accurate numerical solution of hyperbolic conservation laws. 1995. 12th AIAA CFD Conference, San Diego, Paper 95-1658.
- [71] D.J. Mavriplis. Directional Agglomeration Multigrid Techniques for High-Reynolds Number Viscous Flows. Technical Report 98-7, ICASE, 1998.
- [72] D.J. Mavriplis. Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes. Technical Report 98-6, ICASE, 1998.
- [73] C. Mensink. *A 2-D Parallel MultiBlock Method for Viscous and Inviscid Compressible Flow*. PhD thesis, VKI/TU Twente, The Netherlands, 1992.
- [74] F. Menter, H. Grotjans, and F. Unger. Numerical Aspects of Turbulence Modelling for the Reynolds Averaged Navier-Stokes Equations. In *VKI LS 1997-02, Computational Fluid Dynamics*, 1997.
- [75] F.R. Menter. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *AIAA-Journal*, 32(8):1598–1605, 1994.

- [76] F.R. Menter and C.L. Rumsey. Assessment of Two-Equation Turbulence Models for Transonic Flows. 1994. AIAA paper 94-2343.
- [77] L.M. Mesaros and P.L. Roe. Multidimensional fluctuation splitting schemes based on decomposition methods. 1995. 12th AIAA CFD Conference, San Diego, Paper 95-1699.
- [78] A. Mizukami. An Implementation of the Streamline-Upwind/Petrov Galerkin method for linear triangular elements. *Computer Methods in Applied Mechanics and Engineering*, 49:357–364, 1985.
- [79] A. Mizukami and T.J.R. Hughes. A Petrov-Galerkin finite element method for convection-dominated flows: an accurate upwinding technique for satisfying the maximum principle. *Computer Methods in Applied Mechanics and Engineering*, 50:181–193, 1985.
- [80] T. De Mulder. *Stabilized Finite Element Methods for Turbulent Incompressible Single-Phase and Dispersed Two-Phase Flows*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 1997.
- [81] T. De Mulder. Stabilized Finite Element methods (SUPG, GLS, ...) for incompressible flows. In *VKI LS 1997-02, Computational Fluid Dynamics*, 1997.
- [82] J.-D. Müller, P.L. Roe, and H. Deconinck. A Frontal Approach for Internal Node Generation in Delaunay Triangulations. *Numerical Methods in Fluids*, 17:241–256, 1993.
- [83] U. Nävert. *A Finite Element Method for Convection-Diffusion Problems*. PhD thesis, Department of Computer Science, Chalmers University of Technology, Göteborg, Sweden, 1982.
- [84] H. Paillère. *Multidimensional Upwind Residual Distribution Schemes for the Euler and Navier-Stokes Equations on Unstructured Grids*. PhD thesis, Université Libre de Bruxelles, Belgium, 1995.
- [85] H. Paillère, J.-C. Carette, and H. Deconinck. Multidimensional upwind and SUPG methods for the solution of the compressible flow equations on unstructured grids. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.
- [86] H. Paillère, H. Deconinck, and P.L. Roe. Conservative upwind residual-distribution schemes based on steady characteristics of the Euler equations. 1995. 12th AIAA CFD Conference, San Diego, Paper 95-1700.
- [87] H. Paillère, H. Deconinck, and E. van der Weide. Upwind Residual Distribution methods for compressible flow: An alternative to Finite Volume and Finite Element methods; Part I: Scalar Schemes. In *VKI LS 1997-02, Computational Fluid Dynamics*, 1997.

- [88] H. Paillère, E. van der Weide, and H. Deconinck. Multidimensional upwind methods for inviscid and viscous compressible flows. In *VKI LS 1995-02, Computational Fluid Dynamics*, 1995.
- [89] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, 1980.
- [90] O. Hassan; K. Morgan; E.J. Probert; J. Peraire. Unstructured tetrahedral mesh generation for three-dimensional viscous flows. *International journal for numerical methods in engineering*, 39:549–567, 1996.
- [91] J.J. Quirk. A Contribution to the Great Riemann Solver Debate. *International journal for numerical methods in fluids*, 18:555–574, 1994.
- [92] G. Raithby. A critical evaluation of upstream differencing applied to problems involving fluid flow. *Computer Methods in Applied Mechanics and Engineering*, 9:75–103, 1976.
- [93] G. Raithby. Skew upstream differencing schemes for problems involving fluid flow. *Computer Methods in Applied Mechanics and Engineering*, 9:153–164, 1976.
- [94] J. Rice and R. Schnipke. A monotone streamline upwind method for convection-dominated problems. *Computer Methods in Applied Mechanics and Engineering*, 48:313–327, 1985.
- [95] P. Roe. Fluctuations and Signals - A Framework for Numerical Evolution Problems. In *Numerical Methods for Fluid Dynamics*. Academic Press, 1982.
- [96] P. Roe. Finite volume methods for the compressible Navier-Stokes equations. In *Numerical Methods in Laminar and Turbulent Flow, Vol. 5, Part 2*, 1987. Proceedings of the Fifth International Conference, Montreal, Canada.
- [97] P. Roe. Optimum Positive Linear Schemes for Advection in Two and Three Dimensions. *SIAM Journal of Numerical Analysis*, 6:1542–1568, 1992.
- [98] P.L. Roe. Characteristic-based schemes for the Euler equations. *International Journal for Numerical Methods in Engineering.*, 4:551, 1972.
- [99] P.L. Roe. Approximate Riemann Solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43(2), 1981.
- [100] P.L. Roe. Linear advection schemes on triangular meshes. Technical report, Cranfield Institute of Technology, November 1987. CoA 8720.
- [101] P.L. Roe. "Optimum" upwind advection on a triangular mesh. Technical Report 90-75, ICASE, 1990.

- [102] P.L. Roe. Multidimensional upwinding: motivation and concepts. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.
- [103] M. Rudgyard. Cell-vertex methods for steady inviscid flow. In *VKI LS 1993-04, Computational Fluid Dynamics*, 1993.
- [104] C. Rumsey, T. Gatski, S. Ying, and A. Bertelrud. Prediction of high-lift flows using turbulence closure models. 1997. AIAA paper 97-2260.
- [105] Y. Saad. Krylov Subspace Techniques, Conjugate Gradients, Preconditioning and Sparse Matrix Solvers. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.
- [106] Y. Saad and M.H. Schultz. A general minimal residual algorithm for solving nonsymmetric linear systems. *SIAM, Journal on Scientific and Statistical Computing.*, 7:865–869, 1986.
- [107] R. Schwane. Description of the Testcases: MSTP Workshop 1996. Reentry Aerothermodynamics and Ground-to-Flight Extrapolation. *ESA ESTEC Doc. YPA/1889/RS, Noordwijk, NL*, 1996.
- [108] D. Sidilkover. A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible Euler equations. Technical Report 94-84, ICASE, 1994.
- [109] D. Sidilkover. A genuinely multidimensional upwind scheme for the compressible Euler equations. In J. Glimm, M.J. Graham, J.W. Grove, and B.J. Plohr, editors, *Proceedings of the Fifth International Conference on Hyperbolic Problems: Theory, Numerics, Applications*. World Scientific, June 1994.
- [110] D. Sidilkover. Multidimensional upwinding and multigrid. 1995. 12th AIAA CFD Conference, San Diego, Paper 95-1759.
- [111] D. Sidilkover and P.L. Roe. Unification of Some Advection Schemes in Two Dimensions . Technical Report 95-10, ICASE, 1995.
- [112] P. Sonneveld. CGS: a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM, Journal on Scientific and Statistical Computing.*, 10:36–52, 1989.
- [113] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamical flows. *La Recherche Aérospatiale*, (1):5–21, 1994.
- [114] D.B. Spalding. A Novel Finite-Difference Formulation for Differential Expressions Involving Both First and Second Derivatives. *Annual Review Fluid Mechanics*, 18:337–365, 1986.

- [115] J. Steelant and E. Dick. Conditioned Navier-Stokes equations for the calculation of transitional flows. In *Proceedings of the 2nd European CFD Conference*, pages 495–502, Stuttgart, Germany, 1994.
- [116] R. Struijs. *A Multi-Dimensional Upwind Discretization Method for the Euler Equations on Unstructured Grids*. PhD thesis, The University of Delft, The Netherlands, 1994.
- [117] R. Struijs, H. Deconinck, P. De Palma, P.L. Roe, and K.G. Powell. Progress on Multidimensional Upwind Euler Solvers for Unstructured Grids. 1991. AIAA paper 91-1550.
- [118] R. Struijs, H. Deconinck, and P. Roe. Fluctuation Splitting Schemes for multidimensional convection problems: an alternative to finite volume and finite element methods. In *VKI LS 1990-04, Computational Fluid Dynamics*, 1990.
- [119] R. Struijs, H. Deconinck, and P. Roe. Fluctuation Splitting Schemes for the 2D Euler Equations. In *VKI LS 1991-01, Computational Fluid Dynamics*, 1991.
- [120] R. Struijs, H. Deconinck, and P.L. Roe. Fluctuation Splitting Schemes for the 2D Euler Equations. In *VKI LS 1991-01, Computational Fluid Dynamics*, 1991.
- [121] H. Tennekes and J.L. Lumley. *A first course in turbulence*. MIT Press, 1972.
- [122] E. Turkel, V.N. Vatsa, and R. Radespiel. Preconditioning Methods for Low-Speed Flows. Technical Report 96-57, ICASE, 1996.
- [123] J. Vader. A Local Characteristics Method Encompassing Shock-shock Interaction. Master's thesis, Norwegian Institute of Technology, Norway, 1994.
- [124] H.A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. *SIAM, Journal on Scientific and Statistical Computing.*, 13:631–644, 1992.
- [125] E. van der Weide and H. Deconinck. Fluctuation Splitting Schemes for the Euler equations on Quadrilateral Grids. In *ICFD'95, Conference on Numerical Methods for Fluid Dynamics*, Oxford, April 1995.
- [126] E. van der Weide and H. Deconinck. Positive Matrix Distribution Schemes for Hyperbolic Systems, with application to the Euler Equations. pages 747–753, Paris, France, 1996. Wiley.
- [127] E. van der Weide and H. Deconinck. Upwind Residual Distribution methods for compressible flow: An alternative to Finite Volume and Finite Element

- methods; Part II: System Schemes and Applications. In *VKI LS 1997-02, Computational Fluid Dynamics*, 1997.
- [128] E. van der Weide, E. Issman, H. Deconinck, and G. Degrez. A Parallel Implicit Multidimensional Upwind Cell Vertex Navier-Stokes Solver for Hypersonic Applications. In *Aerothermodynamics and Propulsion Integration for Hypersonic Vehicles*, October 1996. AGARD-R-813.
- [129] B. van Leer. Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection. *Journal of Computational Physics*, 23:276–299, 1977.
- [130] B. van Leer. Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov’s Method. *Journal of Computational Physics*, 32:101–136, 1979.
- [131] B. van Leer, W.-T. Lee, and P. Roe. Characteristic Time-Stepping or Local Preconditioning of the Euler equations. In *AIAA 10th Computational Fluid Dynamics Conference*, 1991. AIAA-91-1552-CP.
- [132] D. Vandromme. Overview of Turbulence Models for CFD in Industry. In *VKI LS 1995-03, Industrial Computational Fluid Dynamics*, 1995.
- [133] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. 1993. AIAA paper 93-0880.
- [134] W.G. Vincenti and Jr. C.H. Kruger. *Introduction to Physical Gas Dynamics*. John Wiley & Sons, 1965.
- [135] D.C. Wilcox. Reassessment of the Scale-Determining Equation for Advanced Turbulence Models. *AIAA-Journal*, 26(11):1299–1310, 1988.
- [136] H.C. Yee. Numerical Approximation of Boundary Conditions with Application to Inviscid Equations of Gas Dynamics. Technical report, NASA, 1981. TM 81265.

Summary

In this thesis an attempt has been made to extend previously developed multi-dimensional upwind schemes for scalar advection to non-commuting hyperbolic systems in general and the compressible Euler and Navier-Stokes equations in particular.

In chapter 1 the motivation for the use of multi-dimensional upwind schemes is given by comparing the numerical diffusion coefficients of the standard first-order upwind scheme and the multi-dimensional scheme. It is shown that the latter is less diffusive, especially when the advection direction is not aligned with the grid. Furthermore it is explained that unstructured grids are used for their ability to reduce the grid generation time for complex geometries.

Chapter 2 gives a review of the scalar schemes, also including more classical schemes like Lax-Wendroff and the upwind finite-volume scheme. A local linearization is explained, which links the conservative formulation with the quasi-linear version of the governing equations. Especially the positivity property heavily relies on the existence of such a "conservative" quasi-linear form. Due to the Petrov-Galerkin finite-element analogy of these schemes, the extension of the discretization method to advection-diffusion problems is straightforward. For linear triangles and tetrahedra, the only elements considered in this work, this effectively means that the viscous term is discretized with the Galerkin method.

The extension of the scalar schemes to non-commuting systems is described in chapter 3. This is straightforward for linear schemes, because the scalar parameters in these schemes generalize to matrices. However for the nonlinear schemes, which combine second-order accuracy and monotonicity, some parameters generalize to vectors and consequently the extension to systems poses some severe problems. A formulation for the nonlinear system PSI-scheme is presented, which gives quite good results, but its convergence behavior is far from optimal.

The iterative framework to solve the set of nonlinear algebraic equations, which is the result of the spatial discretization, is the contents of chapter 4. Both the explicit Runge-Kutta time integrators and the implicit backward Euler method are discussed. For the latter method the full Jacobian matrix is stored and the linear systems are solved by the parallel, iterative solver library AZTEC of Sandia. The parallelization of the computer code has been done for two reasons. First, it is necessary to compute the solution in a reasonable amount of time and second, it is needed to obtain the required memory for the fully implicit algorithm, especially

in three space dimensions.

Chapter 5 shows some applications for the Euler equations. First a conservative linearization for these equations is given, followed by a detailed discussion about preconditioning, both in two and three space dimensions. This technique splits off the entropy and total enthalpy equation and for 2D supersonic flows the Euler equations are even written as four scalar advection equations, like the method of characteristics. The results presented in this chapter show the improved quality if the multi-dimensional upwind discretization technique is applied to the preconditioned Euler equations rather than to the standard equations.

An additional feature for the Navier-Stokes equations, chapter 6, is the discretization of the turbulence models, which must be added to these equations to model the influence of the turbulence on the mean flow. A stable method is obtained if the advection terms are discretized with one of the positive schemes and the source terms are treated in a pointwise manner. Furthermore, to enhance stability, only the destruction source terms are treated implicitly, which can be interpreted as a kind of under-relaxation of the turbulence equations. To keep the method at an acceptable efficiency level, only the fully implicit backward Euler time integrator is used for turbulent problems, albeit in combination with approximate Picard Jacobians for the advective parts. Results of this chapter show that the method can be applied without any problem to high Reynolds number turbulent problems.

Samenvatting

Dit proefschrift beschrijft een poging om de eerder ontworpen multi-dimensionale upwind schema's voor scalaire advection uit te breiden naar niet-commuterende hyperbolische systemen in het algemeen en de compressibele Euler- en Navier-Stokes-vergelijkingen in het bijzonder.

De motivatie voor het gebruik van multi-dimensionale upwind schema's wordt in hoofdstuk 1 gegeven door de numerieke diffusie-coëfficiënten van het standaard eerste orde upwind schema met het multi-dimensionale schema te vergelijken. Er wordt aangetoond dat de laatste minder diffuus is, met name wanneer de advection-richting niet samenvalt met de richting van de roosterlijnen. Verder wordt er uitgelegd dat niet-gestructureerde roosters gebruikt worden, omdat deze in staat zijn de benodigde tijd voor de roostergeneratie van gecompliceerde geometrieën te reduceren.

Hoofdstuk 2 geeft overzicht van de scalaire schema's. Dit overzicht bevat tevens meer klassieke schema's zoals Lax-Wendroff en het "upwind" eindige volume schema. Een lokale linearisatie wordt uitgelegd, die de conservatieve formulering verbindt met de quasi-lineaire versie van de beschrijvende vergelijkingen. Met name de positiviteits-eigenschap hangt sterk af van het bestaan van zo'n "conservatieve" quasi-lineaire vorm. Dankzij de Petrov-Galerkin eindige elementen analogie van deze schema's is de uitbreiding van de discretisatie methode naar advection-diffusie problemen eenvoudig. Voor lineaire driehoeken en tetraëders, de enige elementen die worden beschouwd in dit werk, betekent dit effectief dat de viskeuze term gediscrètiseerd wordt met de Galerkin methode.

De uitbreiding van de scalaire schema's naar niet-commuterende systemen wordt beschreven in hoofdstuk 3. Dit is eenvoudig voor lineaire schema's, omdat de scalaire parameters in deze schema's generaliseren tot matrices. Echter voor niet-lineaire schema's, die tweede orde nauwkeurigheid combineren met monotoniteit, generaliseren sommige parameters tot vektoren en dientengevolge levert de uitbreiding naar systemen een paar problemen op. Een formulering voor het system PSI-schema, dat vrij goede resultaten geeft, wordt gepresenteerd, maar helaas is het convergentiegedrag verre van optimaal.

Het iteratieve raamwerk voor het oplossen van het stelsel van niet-lineaire algebraïsche vergelijkingen, dat het resultaat is van de ruimtelijke discretisatie, is het onderwerp van hoofdstuk 4. Zowel de expliciete Runge-Kutta tijdsintegratiemethoden als de impliciete "backward" Euler methode worden besproken. Voor

de laatstgenoemde wordt de volledige Jacobiaan matrix opgeslagen en de lineaire stelsels worden opgelost door middel van de software bibliotheek AZTEC van Sandia. De parallellisatie van het computer programma is gedaan om twee redenen. Ten eerste is het nodig om de oplossing binnen een redelijke tijdsduur te berekenen en ten tweede is het noodzakelijk om het benodigde geheugen voor het volledig impliciete algoritme te verkrijgen, vooral in drie ruimtelijke dimensies.

Hoofdstuk 5 laat een paar toepassingen voor de Euler vergelijkingen zien. Ten eerste wordt de conservatieve linearisatie voor deze vergelijkingen gegeven, gevolgd door een gedetailleerde beschrijving van preconditionering, zowel in twee als drie ruimtelijke dimensies. Deze techniek splitst de entropie- en totale enthalpie-vergelijking af en voor 2D supersone stromingen worden de Euler vergelijkingen zelfs geschreven als vier scalaire advection vergelijkingen, zoals dit ook het geval is in de karakteristieken-methode. De getoonde resultaten in dit hoofdstuk laten de verbeterde kwaliteit zien als de multi-dimensionale upwind discretisatie toegepast wordt op de gepreconditioneerde Euler vergelijkingen i.p.v. de gewone vergelijkingen.

Een bijkomend verschijnsel voor de Navier-Stokes vergelijkingen, hoofdstuk 6, is de discretisatie van de turbulentie-modellen, die toegevoegd moeten worden aan deze vergelijkingen om de invloed van de turbulentie op de gemiddelde stroming te modelleren. Een stabiele methode wordt verkregen indien de advection-termen worden gediscetiseerd met een van de positieve schema's en de brontermen per roosterpunt worden behandeld. Verder wordt om de stabiliteit te vergroten alleen de destructie bronterm impliciet behandeld, wat geïnterpreteerd kan worden als een soort onderrelaxatie van de turbulentie vergelijkingen. Om de methode op een acceptabel efficiëntie-niveau te houden wordt alleen de volledig impliciete "backward" Euler tijdsintegratie-methode gebruikt voor turbulente stromingsproblemen, maar in combinatie met benaderende Picard Jacobianen voor de advection termen. Resultaten van dit hoofdstuk laten zien dat de methode zonder problemen kan worden toegepast voor turbulente stromingen met een hoog Reynolds-getal.

Acknowledgments

It is not feasible to mention everybody who have supported me, or whom I have worked with over the last $4\frac{1}{2}$ years by name. Therefore I restrict myself to a few people, without doing an injustice to the others.

First of all, I am grateful to both my promotors, prof.dr.ir. Herman Deconinck and prof.dr.ir. P.G. Bakker. Herman, thank you that you gave me the opportunity to do this research, that I have had the freedom to do what I wanted to do during this period of time and for reading and correcting the manuscript. Prof. Bakker, I enjoyed my visits to Delft, when, during our conversations, you made me clear that not everything was as obvious as I thought it was. Furthermore, the final result would have contained some errors, if you had not read it so carefully.

I thank all my colleagues at the von Karman Institute for providing a pleasant working environment. Especially Emmanuel and Jean-Christophe for including me in the IcARus project, which gave me the opportunity to develop my C-programming skills and understanding of the spatial and temporal discretization in such a way that I was able to develop the parallel, 3D code later. Nick, for checking my English grammar - in particular all the adverbs, for which I kept on forgetting the addition -ly. In return, I will read your one-page Dutch summary, which is probably about one percent of the effort you had when reading my thesis.

Furthermore, Bert Houtman is acknowledged for his assistance to get the code running on the Cray T3E.

Last, I would like to mention my family and friends back home in Oudewater. You made me feel at home every time I was there and I have always liked playing football with you (if I could play, for being injured was more the rule than the exception in my case).

Edwin van der Weide
Sint-Genesius-Rode, June 1998

Curriculum Vitae

Edwin Theodorus Antonius van der Weide

geboren op 17 augustus 1970 te Oudewater

juni 1988	diploma VWO Sint-Antonius College, Gouda
januari 1992 - maart 1992	stage von Karman Instituut, Sint-Genesius-Rode (B) Onderwerp: Preconditioning van de Euler vergelijkingen
december 1993	diploma Luchtvaart- en Ruimtevaarttechniek, cum laude Technische Universiteit Delft Afstudeeronderwerp: Numerieke simulatie van quasi-1D thermo-chemische niet-evenwichtstromingen
januari 1994 - juni 1998	promotie-onderzoek von Karman Instituut, Sint-Genesius-Rode (B) Technische Universiteit Delft