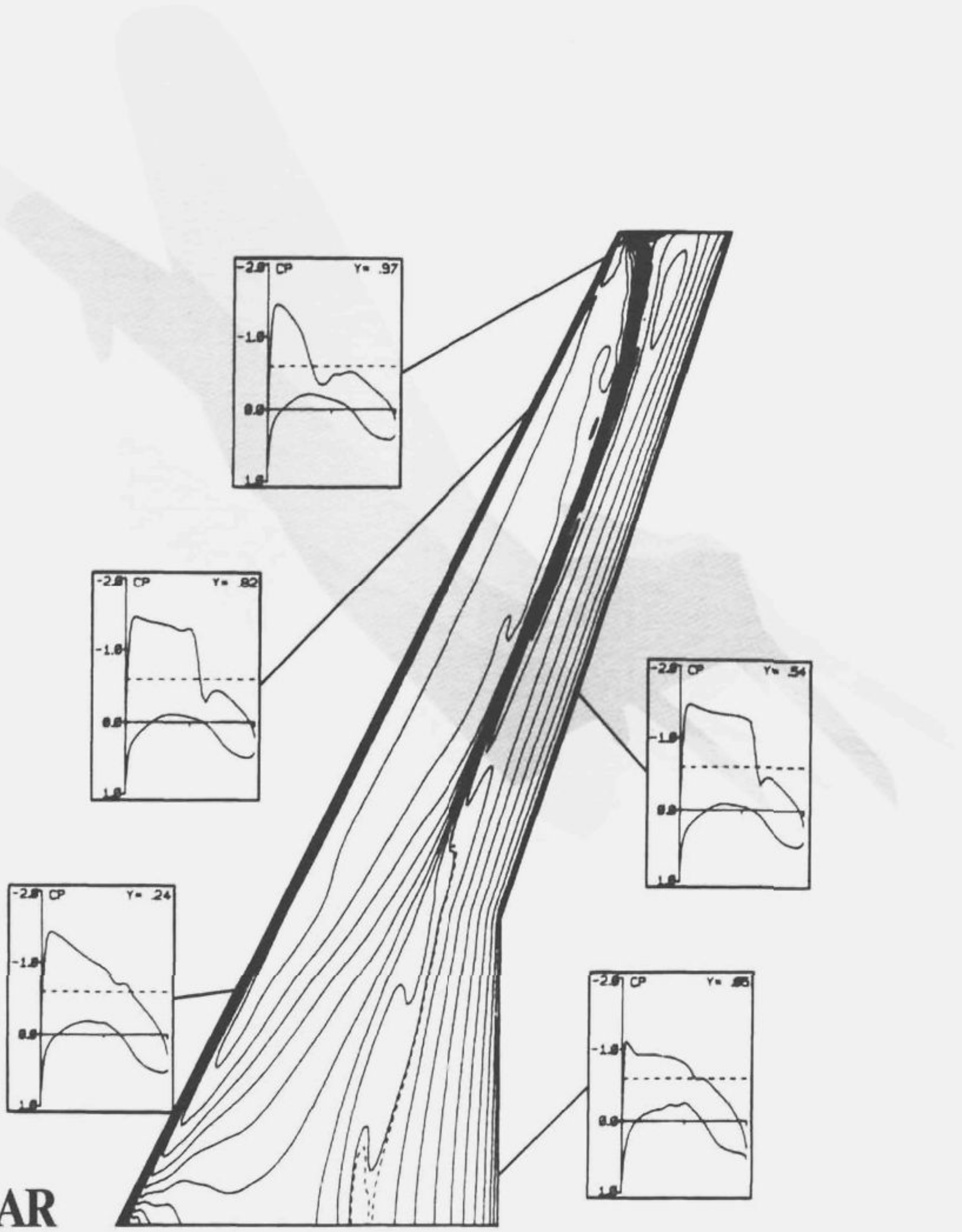


A NONLINEAR MULTIGRID METHOD FOR THREE-DIMENSIONAL TRANSONIC POTENTIAL FLOW

TR diss
1677

A.J. van der Wees

6-89079
37 0897
TR diss 1677

A NONLINEAR MULTIGRID METHOD FOR THREE-DIMENSIONAL TRANSONIC POTENTIAL FLOW

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus,
Prof. Drs. P. A. Schenck,
in het openbaar te verdedigen
ten overstaan van een commissie
door het College van Dekanen
daartoe aangewezen,
op dinsdag 1 november 1988
te 16.00 uur

door

ADRIAAN JAN VAN DER WEES,

geboren te Delft,
Wiskundig ingenieur

BIBLIOTHEEK TECHNISCHE UNIVERSITEIT
Prometheusplein 1
DELFT

Nationaal Lucht- en Ruimtevaartlaboratorium
Amsterdam
1988

TR diss
1677

Dit proefschrift is goedgekeurd door de promotor:
Prof. Dr. Ir. P. Wesseling

STELLINGEN

bij het proefschrift

A NONLINEAR MULTIGRID METHOD FOR THREE-DIMENSIONAL TRANSONIC POTENTIAL FLOW

door

A. J. van der Wees

1 november 1988

1. De introduktie van de Japanse supercomputers (Fujitsu, Hitachi, NEC) heeft bij alle supercomputer-leveranciers geleid tot een belangrijke verbetering van vectorisatie faciliteiten voor Fortran programma's. De auto-vectorisatie faciliteiten van een goede Fortran compiler zijn thans zodanig, dat supercomputers niet meer als "moeilijk te programmeren" mogen worden bestempeld.

2. Ondanks goede auto-vectorisatie faciliteiten van Fortran compilers op supercomputers zal voor produktie-programma's het verkrijgen van een hoge vectorisatiegraad middels een goede algoritme-keuze en geschikte programma-constructies noodzakelijk blijven, aangezien voor korte doorvoertijden van programma's nu eenmaal een zeer hoge vectorisatie-graad vereist is.

3. De bezetting van computers voor technisch-wetenschappelijk rekenwerk (ook supercomputers) wordt met name bepaald door zeer grote programma's, die zo complex zijn, dat een efficient gebruik van de computer-architectuur niet eenvoudig te realiseren is. Derhalve dient bij aanschaf van een computer aan de doorvoertijd van dergelijke programma's een zwaar gewicht te worden toegekend.

4. Doorberekening van kosten van het gebruik van (super)computers dient gebaseerd te zijn op de mate waarin een programma het draaien van andere programma's hindert. Derhalve dient op supercomputers het rekenen in scalaire mode, en het zodanig gebruik van de computer dat processoren nodeloos stil komen te staan, zwaar te worden belast. Anderzijds dient het rekenen buiten de produktie-uren vrijwel gratis te zijn.

5. De grote waarde van de multi-rooster methode voor stromingsberekeningen ligt veelal niet zozeer in de relatief hoge convergentie-snelheid die met deze methode kan worden bereikt, als wel in de relatief lage convergentie-nivo's die bij deze methode voor het verkrijgen van een oplossing met acceptabele nauwkeurigheid kunnen worden gespecificeerd.

6. Een "snelle" oplosmethode voor stelsels van vergelijkingen ontstaan uit discretisatie-methoden vereist in het algemeen het kiezen van een "snel" rekenrooster, dat wil zeggen een rekenrooster waaraan sterke beperkingen worden opgelegd (bijvoorbeeld een rechthoekige struktuur) of een rekenrooster waaraan zodanige eisen worden gesteld (bijvoorbeeld met betrekking tot maasverhoudingen) dat het snel oplossen van het desbetreffende stelsel vergelijkingen mogelijk wordt gemaakt.

7. Het op elkaar afstellen van rekenrooster-generator en stromingsoplosser, zoals in dit proefschrift beschreven, vindt binnen de numerieke aerodynamica veelvuldig plaats; de wijze waarop dit gebeurt komt echter in publicaties nauwelijks ter sprake.

8. Bij profielstromingen leidt het gebruik van asymptotische benaderingen voor verre-veld randvoorwaarden met name tot nauwkeuriger numerieke resultaten. Het verkleinen van het rekengebied is daarbij van secundair belang.
(ref: F.W. Wubs, J.W. Boerstoel en A.J. van der Wees,
Grid-size reduction in flow calculations on infinite domains by higher-order far-field asymptotics in numerical boundary conditions, J. Eng. Math. 18 (1984), p. 157-177)

9. Kleine programmeer-fouten kunnen in grote rekenprogramma's gedurende zeer lange tijd verborgen blijven doordat voor de ermee gepaard gaande anomalieën vanuit het geïmplementeerde model een goede verklaring kan worden gevonden.

10. Artikel 11, lid 2b van het competitie-reglement van de Koninklijke Nederlandse Schaakbond:

> "Spelers die voor de competitie voor een bepaald team zijn opgegeven, dienen voor dat team in deze competitie minimaal tweemaal te spelen. Overtreding van dit artikel wordt door de competitieleider bestraft met het in mindering brengen van twee matchpunten per speler en een door de Algemene Vergadering vast te stellen boete. (...)"

is in principe ter zake, maar druist in tegen in de Nederlandse schaakwereld volledig geaccepteerde gebruiken. Dit als gelegenheids-wetgeving ontstane artikel is derhalve nauwelijks uitvoerbaar.

11. Niet-professionele schakers die al te grote waarde hechten aan hun ELO rating lopen het gevaar de lust tot spelen te verliezen.

SUMMARY

In this thesis research is presented on the application of the nonlinear multigrid method to three-dimensional transonic potential flow. The flow is described by the full potential equation, which is discretized using a finite volume method. The smoothing algorithm in the multigrid method is a combination of Incomplete Lower Upper decomposition (ILU) and Strongly Implicit Procedure (SIP). In general this algorithm is a faster smoothing algorithm than the often used successive line relaxation, while it is also more robust, because it is uniformly stable in the supersonic regions of the flow.

The influence of computational grid properties, such as grid aspect ratio and grid skewness, on the multigrid convergence speed is investigated both theoretically and experimentally. The usefulness of the method for practical applications is demonstrated for the transonic flow about the DFVLR-F4 and ONERA-M6 wing.

ACKNOWLEDGEMENTS

The author is indebted to the management of the Nationaal Lucht- en Ruimtevaartlaboratorium (National Aerospace Laboratory NLR) for giving him the possibility to obtain a doctor's degree on part of his research work carried out at NLR.

This thesis is partly based on work carried out under contract with the Netherlands Agency for Aerospace Programs (NIVR), which kindly gave permission to use the work for the present thesis.

The author wishes to express great gratitude to his colleagues at NLR who, in many cases indirectly, contributed to this thesis. He especially owes thanks to J. van der Vooren and J.H. Meelker, for many years his direct colleagues in the "MATRICS" project.

CONTENTS

Page

1 INTRODUCTION 1

1.1 Purpose and scope 1

1.2 Brief description of transonic wing flow 4

2 FLOW MODEL 7

2.1 Flow configuration 7

2.2 Flow analysis 8

2.2.1 Equations of steady inviscid gasdynamics 8

2.2.2 Boundary conditions 13

2.3 Flow equation in general coordinates 15

2.3.1 General coordinates 15

2.3.2 Flow equation and boundary conditions in general coordinates 17

2.4 Linearization of the full potential equation 18

3 GRIDGENERATION 20

3.1 Gridgeneration process 20

3.2 Boundary condition on branch cut 26

4 FINITE VOLUME DISCRETIZATION 27

4.1 General remarks 27

4.2 Notation 28

4.3 Description of computational grid 29

4.4 Discretization of flow equation 32

4.4.1 General remarks 32

4.4.2 Finite volume discretization 33

4.4.3 Coupling operator 36

4.4.4 Directional bias in hyperbolic regions of the flow 38

4.4.5 Freestream consistent scheme 40

4.5 Boundary condition discretization 42

CONTENTS (Continued)

Page

5 SOLUTION METHOD 47
5.1 General remarks 47
5.2 Multigrid method 48
5.2.1 *Description of multigrid method* 48
5.2.2 Restriction of Neumann type boundary conditions 53
5.2.3 Convergence and efficiency of the multigrid method 54
5.3 Linearization and construction of system matrix 57
5.4 Smoothing algorithm 58
5.4.1 Description of ILU/SIP algorithm 58
5.4.2 Properties and applications of ILU/SIP 62

6 STABILITY AND SMOOTHING ANALYSIS OF ILU/SIP ALGORITHM 65
6.1 Smoothing analysis for elliptic problems 65
6.1.1 General description 65
6.1.2 Influence of grid aspect ratios on smoothing properties 68
6.1.3 Influence of grid skewness on smoothing properties 72
6.2 Stability analysis of ILU/SIP for hyperbolic problems 80

7 IMPLEMENTATION ASPECTS 85
7.1 General remarks 85
7.2 Storage and computational speed requirements 86

8 NUMERICAL EXPERIMENTS 89
8.1 General description and basic choices 89
8.2 Numerical experiments for confirmation of smoothing analysis 90
8.2.1 Incompressible flow in windtunnel with a bump on the bottom wall 90
8.2.2 Simplified wing in transonic flow 103
8.3 Convergence of solution method for realistic wings in transonic flow 107
8.3.1 General remarks 108
8.3.2 Convergence acceleration by local smoothings 112
8.3.3 Convergence of highly transonic flow calculations 112
8.3.4 Concluding remarks on convergence of MG-ILU/SIP method 112

CONTENTS (Continued)

Page

8.4 Computational results 115
8.4.1 Results of finite volume method 115
8.4.2 Transonic flow about DFVLR-F4 wing and ONERA-M6 wing 118
8.4.3 Results of flow computation at engineering accuracy 122

9 CONCLUSIONS AND FINAL REMARKS 127

REFERENCES 130

APPENDIX A: ABBREVIATIONS AND SYMBOLS 136

APPENDIX B: ERROR MATRICES FOR THE SLR AND ILU/SIP ALGORITHMS 141

CURRICULUM VITAE OF THE AUTHOR 148

SUMMARY IN DUTCH (Samenvatting in het Nederlands) 149

1 INTRODUCTION

1.1 Purpose and scope

The aerodynamic design process of an aircraft requires reliable tools to predict its aerodynamic behaviour in order to avoid design modifications at a later stage. Next to the windtunnel, a valuable tool in this respect is the numerical simulation of the flow. The flow simulation method developed in the present thesis is such a tool, developed in particular for the design of transport aircraft under cruise conditions.

Flow simulation methods for aircraft can be based on various levels of physical flow modelling. The choice depends on the purpose of the simulation and the related physical relevance required, as well as on the combination of numerical techniques and computing power at disposal. To date, the highest attainable level of physical flow modelling feasible for full aircraft configurations are the Reynolds-averaged Navier-Stokes equations. These equations describe almost all relevant flow phenomena, but require an adequate semi-empirical turbulence model. For full aircraft configurations, simulation methods based on the Reynolds-averaged Navier-Stokes equations require computation times in the order of days on even the fastest supercomputers existing today.

Simplifications of the flow model, maintaining physical relevance, are possible only if the flow is split up in viscosity dominated regions and nominally inviscid regions. Viscosity dominated regions are e.g. boundary layers and viscous wakes, and are always located at or generated from the airplane surface (inner flow). Inviscid regions are always located away from the airplane surface (outer flow). Although viscous forces are also not negligible in shocks, these can often be modelled adequately without taking viscosity into account.
In airplane aerodynamics, it is often possible to model the flow in viscous regions using a simplified form of the Navier-Stokes equations, the so-called thin layer equations, or the even simpler boundary layer equations.

In inviscid flows, all viscous terms in the Reynolds-averaged Navier-Stokes equations are dropped and the Euler equations result as the flow model. These equations retain the property to describe the transport of vorticity (rotational flow). Important sources of vorticity are boundary layers and viscous wakes. Also propellers and jet exhausts (propulsion systems) generate vorticity. Further sources of vorticity are shock waves (only in transonic flow). For full aircraft configurations, simulation methods based on the Euler equations require computation times in the order of hours on the fastest supercomputers existing today.

When shock waves are weak, the flow is approximately isentropic and the vorticity generated by the shocks is almost negligible. Then, if no other sources of vorticity need be modelled, the flow can be considered isentropic and irrotational, and potential theory can be used. In particular, potential theory is the exact model for the inviscid outer flow in the subsonic case. For full aircraft configurations, simulation methods based on potential theory require computation times in the order of minutes on the fastest supercomputers existing today and in the order of hours on classical mainframes.

An important design goal for transport aircraft is low drag, implying low fuel consumption. For transonic transport aircraft this has led to the so-called supercritical wings, which have local supercritical flow regions terminated by weak shock waves under cruise conditions, and consequently low wave drag. Under these circumstances potential theory is an adequate model for the major part of the inviscid flow.

At present, the situation with respect to flow simulation methods for the aerodynamic design of transport aircraft under cruise conditions is as follows. Reynolds-averaged Navier-Stokes methods and Euler methods are not yet developed far enough to be used routinely and repeatedly in aerodynamic design loops. In this respect potential methods are feasible at present, but they require improvements with respect to accuracy, robustness and computational speed. The potential method developed in this thesis is a contribution to the fulfilment of these requirements. Physical relevance requires that this potential method (covering only the outer flow) is coupled with a fast and reliable boundary layer computation method, but this aspect is outside the scope of this thesis.

The continuous growth of available computing power and the further improvement of flow solution algorithms will make the Reynolds-averaged Navier-Stokes and Euler equations better applicable in aerodynamic design loops in the future. Potential methods will then still be valuable for quite a long time, however, because in that situation they can be used as a comparatively fast computing tool for use in early stages of the aircraft design process.

In this thesis a relatively standard finite volume formulation will be used to discretize the transonic potential flow equation. Many of today's routinely used computer codes for the solution of the resulting system of equations still rely on the Successive Line Over Relaxation (SLOR) algorithm as developed by e.g. Murman and Cole [1] and Jameson [1]. Though this algorithm has proved to be fairly reliable in many cases of practical interest, it does not satisfy the practical requirements of accuracy, robustness and computational speed outlined above. An improvement has been the Approximate Factorization (AF) algorithm, which generally has better convergence properties (Holst [1]). However, the stable relaxation of the required boundary conditions for the auxiliary variables introduced by this method requires special attention (South and Hafez [1]). Another improvement has been the Conjugate Gradients (CG) method, which is a widely applicable convergence acceleration method that also has proved to be useful for transonic potential flow solution methods (Brédif [1]). A real breakthrough, however, seems to have been the MultiGrid (MG) method, introduced by Fedorenko [1], Hackbusch [1] and Brandt [1], and particularly popularized by the latter author. Today, the multigrid method is the only solution method whose efficiency is not affected when computational grids are further refined and higher convergence levels are required. This property has been confirmed in many applications and is also the reason for adopting multigrid in the finite volume method developed in this thesis. However, the robustness of the multigrid method depends heavily on the damping characteristics and the robustness of the relaxation algorithm used to smooth the short wavelength error components on each grid level.

The choice of the "best" smoothing algorithm in a multigrid method for a specific class of applications depends on the balance between its damping characteristics for short wavelength errors, its computational complexity and its robustness. In this respect it is important to note that grids for practical aircraft configurations often exhibit strongly different grid aspect ratios and variations in grid skewness and in grid stretching.

The smoothing algorithm most frequently used is Successive Line Relaxation (SLR); this algorithm has been shown to work in both two and three dimensions (Jameson [2], Boerstoel and Kassies [1], Shmilovich and Caughey [1]). However, considerable deterioration of the rate of convergence can be observed for multigrid methods with SLR smoothing on grids with high grid aspect ratio. Moreover, SLR is not stable for all local flow directions in the supersonic (hyperbolic) regions of the flow and is therefore not robust.

In a search for more efficient and robust smoothing algorithms several alternatives have been considered. In two-dimensional applications, the Alternate Direction Implicit (ADI) and the Approximate Factorization algorithms have been found to be successful multigrid smoothing algorithms (Schmidt and Jameson [1]). So far, no success has been reported for multigrid methods with AF or ADI smoothing in three-dimensional applications. Investigations by the present author along this line remained without success. Other possible smoothing algorithms are the Incomplete Lower Upper decomposition (ILU, Meijerink and van der Vorst [1]) and the Strongly Implicit Procedure (SIP, Stone [1]). Both algorithms have been investigated for two-dimensional transonic flow (Brédif [2], Sankar [1]). The work presented in this thesis combines these two algorithms into one algorithm: ILU/SIP. This algorithm is implicit in all three coordinate directions and uniformly stable in supersonic (hyperbolic) flow regions. The algorithm has also turned out to be very robust and is at least twice as fast as SLR when used as a multigrid smoothing algorithm (Van der Wees, Van der Vooren and Meelker [1]). No tuning of parameters is required, and its computer-resource requirements (processor time and memory) are moderate. In this thesis, its performance as a smoothing algorithm is a nonlinear multigrid method will be analyzed. The multigrid method will be applied to the computation of the subsonic and transonic potential flow about a wing.

1.2 Brief description of transonic wing flow

For readers not familiar with the aerodynamics of transport aircraft, a brief description of the transonic flow about a transport wing is given below. An extensive introduction to numerical flow simulation about aircraft can be found in Loeve and Van der Vooren [1] (in Dutch).

At high subsonic cruise conditions, the flow about a transport wing is often transonic. This means that the flow has embedded supersonic zones, mostly on the wing upper surface. Such zones occur whenever the flow accelerates locally beyond the speed of sound. Each zone is usually terminated on the downstream side by a shock wave, which causes the flow to decelerate again to subsonic speed almost discontinuously (Figs. 1.2.1, 1.2.2).

At the downstream (trailing) edge of the wing, the flow coming off

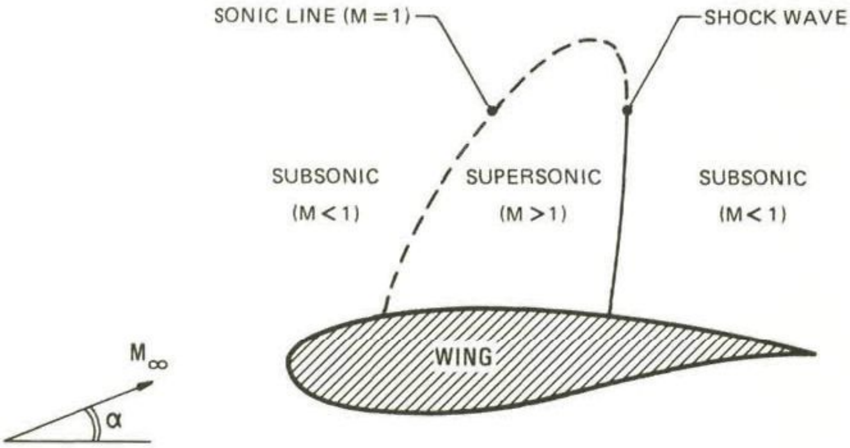


Fig. 1.2.1 Transonic flow about a wing

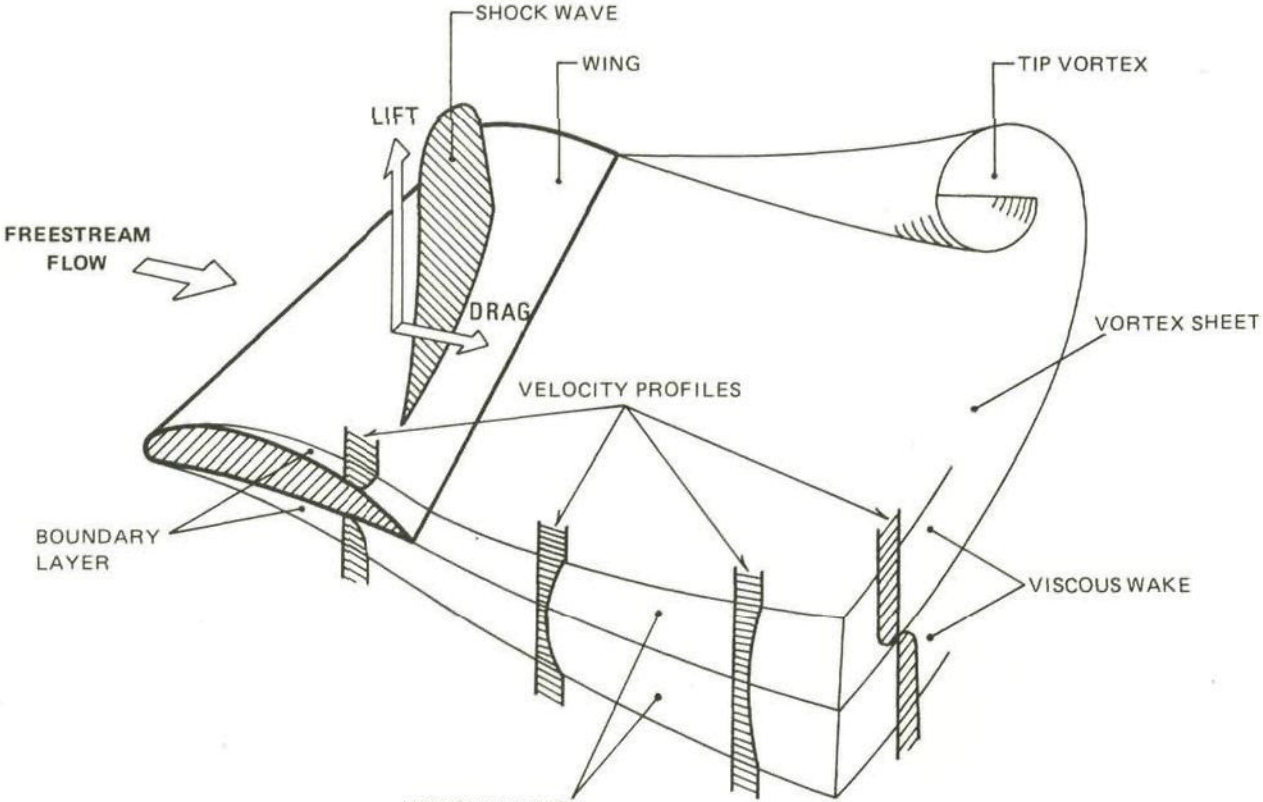


Fig. 1.2.2 Boundary layer and vortex sheet at a wing

the upper and the lower surface generally has different velocity and direction, mainly because the wing produces lift (Fig. 1.2.2). This causes a shear layer behind the wing, the so-called vortex sheet. This vortex sheet rolls up under the action of its self-induced velocities and produces the phenomenon of the tip vortex (Fig. 1.2.2).

The flow phenomena described above are of purely inviscid character. However, close to the wing and the vortex sheet, the flow is in reality dominated by viscous effects. This leads to the boundary layer and the viscous wake (Fig. 1.2.2).

The drag force acting on the wing is composed mainly of vortex induced drag (related to the phenomenon of lift), wave drag (caused by the shock waves) and friction drag (caused by the boundary layer).

2 FLOW MODEL

2.1 Flow configuration

In this thesis, the inviscid, steady, transonic potential flow about a simple aircraft configuration, viz. a symmetric wing, will be considered. The undisturbed freestream flow is uniform and parallel to the symmetry plane and therefore it suffices to consider semi-configurations, cf. figure 2.1.1.

The coordinate system x^i, i = 1,2,3, also called (x,y,z), in physical space is Cartesian, right-turning and attached to the wing, with its origin in the symmetry plane. The coordinate system x^i is defined as shown in figure 2.1.1.

Although this is not strictly necessary in Cartesian coordinates, in our notation we will distinguish between covariant and contravariant tensor components, in order to facilitate easy transformation to general coordinates at a later stage.

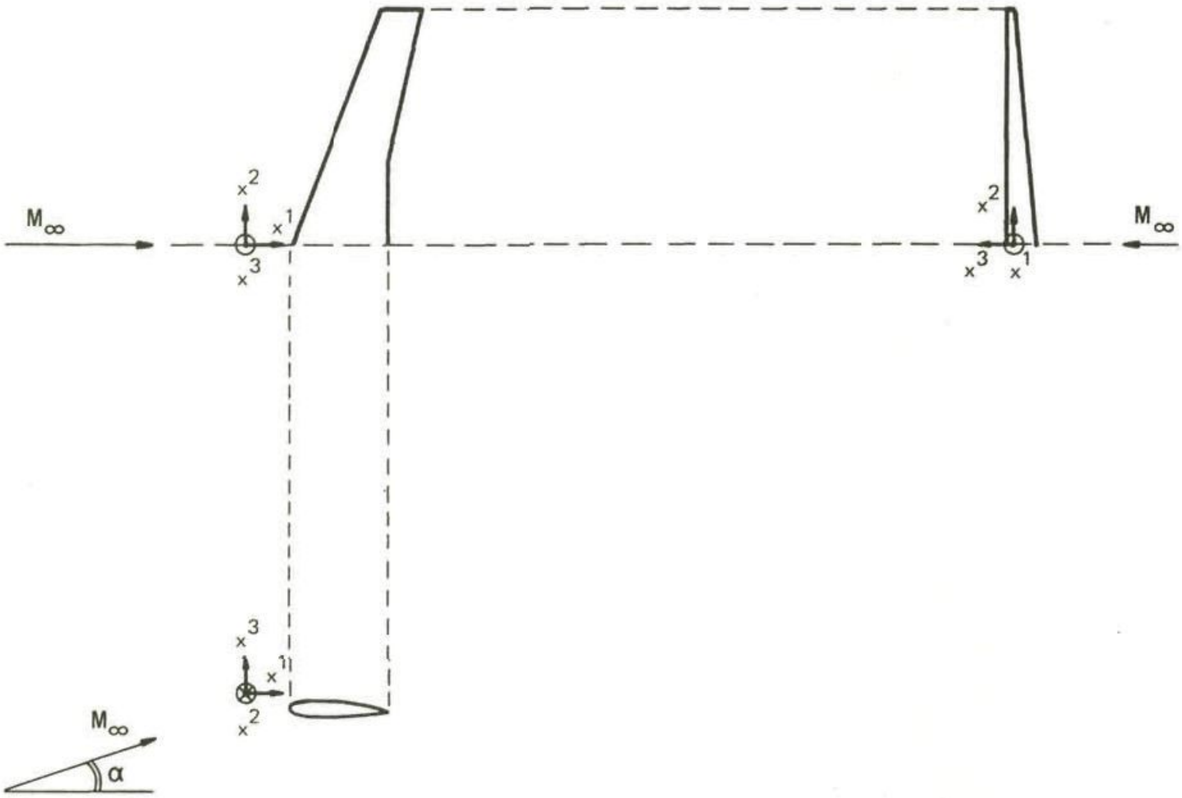


Fig. 2.1.1 Coordinate system in physical space attached to the wing

2.2 Flow analysis

2.2.1 Equations of steady inviscid gasdynamics

The aerodynamic equations of wing flow are well established and can be found in standard textbooks for aerodynamics, such as Liepmann and Roshko [1], Batchelor [1] or Shapiro [1].
Two state-of-the-art survey articles on computational aerodynamics are Holst, Slooff, Yoshihara and Ballhaus [1] and Holst [2].

The most complete model for continuum gasdynamics is provided by the Navier-Stokes equations, which model viscosity and describe turbulence phenomena. To date, the highest attainable level of physical modelling that is feasible for the flow computation about a wing are the Reynolds-averaged Navier-Stokes equations. These equations describe turbulence using a semi-empirical turbulence model.

In regions of the physical domain where viscous effects are negligible, e.g. outside boundary layers and viscous wakes, the Reynolds-averaged Navier-Stokes equations reduce to the Euler equations.
In this thesis we assume that the flow is independent of time (steady flow). Then the Euler equations are:

conservation of mass: $$\frac{\partial}{\partial x^i}(\rho u^i) = 0 , \tag{2.2.1a}$$

conservation of momentum: $$\frac{\partial}{\partial x^i}(\rho u^i u^j) = -\delta^{jk}\frac{\partial p}{\partial x^k} , \quad j = 1,2,3, \tag{2.2.1b}$$

conservation of energy: $$\frac{\partial}{\partial x^i}(\rho u^i h_0) = 0. \tag{2.2.1c}$$

Here the specific total enthalpy h_0 per unit volume is given by

$$h_0 = e + \frac{p}{\rho} + \frac{q^2}{2} , \quad q^2 = u^i u_i \tag{2.2.2}$$

and the specific internal energy e per unit volume is given by the caloric equation of state:

$$e = c_v T. \tag{2.2.3}$$

The system of equations is completed by the thermal equation of state for an ideal gas:

$$p = \rho RT \quad , \; R = c_p - c_v \, . \tag{2.2.4}$$

The entropy of an ideal gas is defined relative to its freestream value by

$$\Delta S = S - S_\infty = c_v \ln (p/\rho^\gamma) - c_v \ln (p_\infty/\rho_\infty^\gamma), \tag{2.2.5a}$$

$$\gamma = c_p/c_v. \tag{2.2.5b}$$

In equations (2.2.1) until (2.2.5) the velocity q, the pressure p and the density ρ are scaled such that

$$q_\infty = 1, \; \rho_\infty = 1, \; p_\infty = \frac{1}{\gamma M_\infty^2}. \tag{2.2.6}$$

It is important to note that in the Euler model viscosity has been set equal to zero. As a consequence the model allows both for compression shocks and for (non-physical) expansion shocks. The latter have to be removed from the model by imposing the additional condition that the second law of thermodynamics must be satisfied, requiring that the entropy may not decrease along a streamline.

The Euler equations (2.2.1) require the solution of a complex system of five equations, giving the unknown dependent variables ρ, u^1, u^2, u^3, p. The potential flow model, to be described subsequently, requires the solution of only one equation, giving the unknown variable Φ. We will now present the additional assumptions that allow simplification of the Euler equations to the so-called full potential equation.

Define the vorticity of the fluid as

$$\vec{\omega} = \text{rot } \vec{u} = \vec{\nabla} \times \vec{u}. \tag{2.2.7}$$

Define further a vortex line as a line which is everywhere tangent to the vorticity vector. A vortex sheet is defined as a two-dimensional manifold consisting of vortex lines. In this thesis we will assume that vortex sheets only leave from sharp edges on the downstream side of the wing (the wing trailing edge), thus excluding vortex sheets coming off e.g. the wing leading edge.

Using Stokes' theorem in vector analysis, the following equation follows from equation (2.2.7):

$$\int_A \vec{\omega}.\vec{n}dA = \int_A (\vec{\nabla} \times \vec{u}).\vec{n}\ dA = \oint_C \vec{u}\ d\vec{x}\ ,\ n\ A, \tag{2.2.8}$$

where C is an arbitrary closed curve and A *any surface with C as boundary*. The right term of equation (2.2.8) is called the circulation along curve C. Let be C a closed contour that moves with the fluid, then it can be shown that in frictionless barotropic flow (i.e. $\rho = \rho(p)$, thus excluding shocks) the circulation along streamlines is constant (Kelvin's theorem). Then for all streamlines that originate in a region without circulation, the left side of equation (2.2.8) must equal zero. Since C and A are arbitrary, it follows that the vorticity $\vec{\omega}$ must be zero, meaning that the flow is irrotational. Consequently, the only streamlines that can carry a nonzero vorticity are those that have passed through shocks.

Another useful relation is the Crocco-Vazsonyi relation, that can be derived from equations (2.2.1) to (2.2.7):

$$\vec{\omega}\times\vec{u} = \text{grad } h_0 - T \text{ grad } \Delta S,\ \Delta S = S-S_\infty. \tag{2.2.9}$$

This equation shows that where grad $h_0 \neq 0$ and/or grad $\Delta S \neq 0$, $\vec{\omega} \neq 0$, or (less likely) $\vec{\omega}$ is parallel to $\vec{u}$. It can be shown that S is constant along streamlines outside shocks. Across shocks S jumps and increases an amount depending on the shock strength. For weak shocks (upstream normal Mach number $M_1 \approx 1$) it can be derived that

$$S_2 - S_1 \approx \frac{2\gamma}{(\gamma+1)^2}\ \frac{(M_1-1)^3}{3}\ , \tag{2.2.10}$$

where subscripts 1 and 2 refer to conditions directly upstream and downstream of the shock. Hence, across shocks grad $S \neq 0$, and consequently $\vec{\omega} \neq 0$ according to equation (2.2.9). In transonic potential flow models it is assumed that shocks are weak ($M_1 < 1.3$, say). The fact that according to equation (2.2.10) the jump in entropy is only of third order in shock strength justifies the assumption of isentropic flow, i.e. $S_1 = S_2 = S_\infty$ throughout the flow. It can be shown that as a result of this assumption either mass or momentum in the direction normal to the shock will not be preserved exactly.

Under the assumption of isentropic flow the following formula for the pressure can be derived from equations (2.2.5) and (2.2.6),

$$p = \rho^{\gamma}/\ (\gamma M_\infty^2). \tag{2.2.11}$$

Next, we introduce Bernoulli's equation, which follows from the energy equation (2.2.1c), while using equations (2.2.1a,1b) and (2.2.4), and states that the total enthalpy is constant along streamlines and consequently throughout the flow field:

$$h_0 = \frac{q^2}{2} + \frac{\gamma-1}{\gamma}\frac{p}{\rho} = \frac{q_\infty^2}{2} + \frac{\gamma}{\gamma-1}\frac{p_\infty}{\rho_\infty} = \tfrac{1}{2} + \frac{1}{(\gamma-1)M_\infty^2}\ . \tag{2.2.12}$$

The so-called isentropic formula for the density follows from equations (2.2.11) and (2.2.12):

$$\rho = \{1 + \frac{\gamma-1}{2} M_\infty^2\ (1-q^2)\}^{\frac{1}{\gamma-1}}\ ,\ q^2 = u^i u_i = \delta^{ij} u_i u_j. \tag{2.2.13}$$

Note that this analytic formula thus replaces the energy equation in isentropic flow.

Under the assumption of irrotational flow it is possible to introduce a velocity potential Φ by

$$u_i = \frac{\partial \Phi}{\partial x^i},\ i = 1,2,3, \tag{2.2.14}$$

since then $\vec{\omega} = \text{rot}\ \vec{u} \equiv 0$.

In this thesis the velocity u_i, $i = 1,2,3$ is split in the (given) freestream component $u_{\infty i}$ and the perturbation velocity component $\partial\varphi/\partial x^i$, whence

$$u_i = u_{\infty i} + \frac{\partial \varphi}{\partial x^i},\ i = 1,2,3. \tag{2.2.15}$$

An important property of irrotational flow is that in regions with continuous flow (outside shocks and vortex sheets) the momentum equations are automatically satisfied if the mass conservation equation and the energy equation are satisfied. Therefore, the mass conservation equation (2.1.1a) can be used to describe the flow, which leads to the so-called full potential equation in conservation form:

$$\frac{\partial}{\partial x^j} (\rho\delta^{ij} u_i) \equiv \frac{\partial}{\partial x^j} (\rho\delta^{ij}(u_{\infty i} + \frac{\partial\varphi}{\partial x^i})) = 0 \qquad (2.2.16)$$

with the density ρ given by equation (2.2.13).

At shocks we will choose to conserve mass. Consequently, momentum in the direction normal to the shock will not be preserved.

For later reference, the following quantities are finally introduced.
The pressure coefficient C_p is given by

$$C_p = \frac{p-p_\infty}{\frac{1}{2}\rho_\infty q_\infty^2} = \frac{2}{\gamma M_\infty^2} [\{1 + \frac{\gamma-1}{2} M_\infty^2 (1-q^2)\}^{\frac{\gamma}{\gamma-1}} - 1] . \qquad (2.2.17)$$

The asymptotic approximation of equation (2.2.17) for $M_\infty = 0$ is

$$C_p = 1-q^2 . \qquad (2.2.18)$$

The local Mach number M is defined by

$$M^2 = \frac{q^2}{a^2} \qquad (2.2.19)$$

where the local speed of sound a is given by

$$a^2 = (\frac{\partial p}{\partial \rho})_{S = \text{constant}} = \gamma\frac{p}{\rho} = \frac{1}{M_\infty^2} + \frac{\gamma-1}{2} (1-q^2) . \qquad (2.2.20)$$

The sonic velocity q^* and the sonic density ρ^* follow from equations (2.2.13), (2.2.19), (2.2.20) by requiring $M^2 = 1$ and are given by

$$q^* = \{\frac{2}{(\gamma+1)M_\infty^2} + \frac{\gamma-1}{\gamma+1}\}^{\frac{1}{2}} , \qquad (2.2.21)$$

$$\rho^* = \{\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1} M_\infty^2\}^{\frac{1}{\gamma-1}} . \qquad (2.2.22)$$

2.2.2 Boundary conditions

On the wing surface and in the symmetry plane zero normal velocity is prescribed,

$$u^i n_i = 0. \qquad (2.2.23)$$

Equation (2.2.23) is not sufficient to fix the lift of the wing. This requires additionally that the flow at the sharp trailing edge of the wing satisfies the Kutta condition. Because air cannot flow around sharp trailing edges (this would require infinite centripetal acceleration) the Kutta condition requires that the flow speed remains finite there. The condition implies that the flow leaves the wing trailing edge in a direction between the tangents to the upper and lower wing surfaces.

The streamlines that leave the wing trailing edge constitute a vortex sheet (shear layer) S_w, which is modelled as a contact discontinuity and therefore has zero normal velocity and continuous pressure, i.e.

$$u^i n_i \;\; = 0 \text{ on upper and lower side of } S_w, \qquad (2.2.24a)$$

$$[\![\, p \,]\!] \;\; = 0 \text{ on } S_w. \qquad (2.2.24b)$$

The shape of the vortex sheet is to be determined as part of the solution. Equations (2.2.24) imply that there may be a jump in tangential velocity across S_w in the sense that the magnitude of the velocity is the same on both sides of S_w, but its direction is different. Defining the mean streamline direction $\vec{s}$ as the direction of the average of the tangential velocities on S_w, the streamline direction is $\vec{s} + \vec{t}$ on one side of S_w and $\vec{s} - \vec{t}$ on the other side of S_w, where $\vec{t}$ is normal to $\vec{s}$ and $\vec{n}$, see also figure 1.2.2. We can then replace the nonlinear boundary condition (2.2.24b) by the (equivalent) boundary condition

$$[\![\, u^i s_i \,]\!] \;\; = 0 \text{ along } \vec{s} \text{ in } S_w, \qquad (2.2.25)$$

where S_w is still to be determined as part of the solution.

Equations (2.2.24a) and (2.2.25) prescribe three conditions, while only two can be satisfied by the potential flow model. Therefore the conditions across the vortex sheet are simplified further by assuming a (fixed) shape of the vortex sheet and weakening the conditions across the vortex sheet. In this thesis we replace the two conditions of equation (2.2.24a) by a zero jump in normal mass flux. Then the conditions across the vortex sheet become

$$\llbracket \rho u^i n_i \rrbracket = 0 \text{ on } S_w, \tag{2.2.26a}$$

$$\llbracket u^i s_i \rrbracket = 0 \text{ along prescribed directions } \vec{s} \text{ in } S_w, \tag{2.2.26b}$$

with the position of the directions $\vec{s}$ prescribed in advance. Here $\vec{s}$ is chosen to coincide with the computational coordinate lines (to be introduced in section 2.3) that leave the wing at the trailing edge, let us say lines (ξ^2, ξ^3) = constant, and S_w is specified as the surface that is swept out by these lines. Using (cf. equation (2.2.14), (2.2.15))

$$\Phi = u_{\infty i} x^i + \varphi \tag{2.2.27}$$

it follows that equation (2.2.26b) is satisfied if we impose

$$\llbracket \varphi \rrbracket \Big|_{\xi^1,\xi^2,\xi^3} = \Gamma \Big|_{\xi^2,\xi^3}, \quad \Gamma \Big|_{\xi^2,\xi^3} = \llbracket \varphi \rrbracket_{\substack{\text{trailing}\\ \text{edge}}} \Big|_{\xi^2,\xi^3} \tag{2.2.28}$$

In many computational codes a zero jump in normal velocity instead of a zero jump in normal mass flux is prescribed across the vortex sheet. This means that equation (2.2.26a) is replaced by

$$\llbracket u^i n_i \rrbracket = 0. \tag{2.2.29}$$

As a result, in these codes mass will not be preserved across the vortex sheet. Because we will have exact mass conservation in the rest of the flow, we choose to maintain mass conservation also across vortex sheets. As a result, the normal velocity will be discontinuous across the vortex sheet.

In practice, the solution for the flow field (e.g. shock position, lift) is influenced to some extent by the direction in which the vortex sheet leaves the wing trailing edge. Because the flow solution is nearly singular at the wing trailing edge, an accurate numerical solution method will require a high grid resolution there.

Finally, the far field boundary conditions are formulated as follows. In the far field, except downstream, the velocity perturbation potential φ is set to zero,

$$\varphi = 0. \tag{2.2.30}$$

Downstream the flow is required to be undisturbed in a prescribed direction $\vec{s}_\infty$, e.g. the freestream direction:

$$s_\infty^i \frac{\partial \varphi}{\partial x^i} = 0. \tag{2.2.31}$$

This condition implies that all prescribed directions $\vec{s}$ in equation (2.2.26b) must lie in the direction $\vec{s}_\infty$ at downstream infinity.

Equations (2.2.23, 2.2.26a, 2.2.31) can be regarded as Neumann type boundary conditions. Equation (2.2.30) is a Dirichlet type boundary condition. Equation (2.2.28) is a periodic boundary condition.

2.3. Flow equation in general coordinates

2.3.1 General coordinates

The flow equation (2.2.16) will be solved in general coordinates ξ^α using the mapping

$$x^i = x^i(\xi^\alpha), \; i = 1,2,3 \; , \; \alpha = 1,2,3. \tag{2.3.1}$$

Define the Jacobian matrix H as

$$H = [\frac{\partial x^i}{\partial \xi^\alpha}] = \begin{bmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} & \frac{\partial x^1}{\partial \xi^3} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} & \frac{\partial x^2}{\partial \xi^3} \\ \frac{\partial x^3}{\partial \xi^1} & \frac{\partial x^3}{\partial \xi^2} & \frac{\partial x^3}{\partial \xi^3} \end{bmatrix} \qquad (2.3.2)$$

and the Jacobian as the determinant of H,

$$h = \det [H]. \qquad (2.3.3)$$

Then the contravariant metric tensor G is defined as

$$[G] = [H^T H]^{-1} = [g^{ij}] = [\delta^{\alpha\beta} \frac{\partial \xi^i}{\partial x^\alpha} \frac{\partial \xi^j}{\partial x^\beta}] = \begin{bmatrix} g^{11} & g^{12} & g^{13} \\ g^{21} & g^{22} & g^{23} \\ g^{31} & g^{32} & g^{33} \end{bmatrix}. \qquad (2.3.4)$$

Note that G is symmetric.

Covariant differentiation of scalars is defined as

$$\varphi_{,i} = \frac{\partial \varphi}{\partial \xi^i} . \qquad (2.3.5)$$

In this thesis we will not explicitly need the covariant differentiation of vectors.

The contravariant and covariant velocity tensors U^α and U_α are defined by

$$U^\alpha = \frac{\partial \xi^\alpha}{\partial x^i} u^i, \; U_\alpha = \frac{\partial x^i}{\partial \xi^\alpha} u_i. \qquad (2.3.6)$$

Note that capital letter variables always denote tensors defined on the ξ^α-coordinate system, while the corresponding small letter variables denote vectors defined on the x^i-coordinate system.

The following relation can be derived from equations (2.3.4) and (2.3.6),

$$U^i = g^{ij} U_j. \qquad (2.3.7)$$

2.3.2 Flow equation and boundary conditions in general coordinates

According to the principles of tensor calculus, the flow equation in general coordinates is obtained by writing it in the invariant scalar form, which reduces to the Cartesian form in Cartesian coordinates. Along this line the mass conservation equation (2.2.16) formulated in general coordinates becomes

$$\frac{1}{h}\frac{\partial}{\partial \xi^i}(\rho h U^i) = 0. \tag{2.3.8}$$

The covariant freestream velocity tensor $U_{\infty i}$ is given by

$$U_{\infty i} = \frac{\partial x^\alpha}{\partial \xi^i} u_{\infty\alpha} . \tag{2.3.9}$$

The covariant form of (2.2.15) is

$$U_i = U_{\infty i} + \varphi_{,i} \tag{2.3.10}$$

and hence

$$U^i = g^{ij}\,(U_{\infty j} + \varphi_{,j}). \tag{2.3.11}$$

The velocity q is obtained from

$$q^2 = U^i U_i = g^{ij} U_i U_j . \tag{2.3.12}$$

The boundary conditions (2.2.23), (2.2.26a) can be written in general coordinates using the identity

$$u^i n_i = U^\alpha N_\alpha . \tag{2.3.13}$$

The boundary condition (2.2.31) can be written in general coordinates using equation (2.3.6):

$$s^i_\infty \; \frac{\partial \xi^\alpha}{\partial x^i} \frac{\partial \varphi}{\partial \xi^\alpha} = 0. \tag{2.2.14}$$

2.4 Linearization of the full potential equation

For later use, we will derive the linearization of the flow equation (2.3.8) by considering a perturbation around a given potential $\bar{\varphi}$. Hence, we put

$$\varphi = \bar{\varphi} + \Psi,\ U^i = \bar{U}^i + \delta U^i,\ \rho = \bar{\rho} + \delta\rho, \tag{2.4.1}$$

with

$$\bar{U}^i = g^{ij}\,\bar{U}_j = g^{ij}\left(\frac{\partial x^\alpha}{\partial \xi^j}\,u_{\infty\alpha} + \bar{\varphi}_{,j}\right), \tag{2.4.2}$$

$$\bar{\rho} = \left\{1 + \frac{\gamma-1}{2}M_\infty^2\,(1-\bar{q}^2)\right\}^{\frac{1}{\gamma-1}},\ \bar{q}^2 = \bar{U}^i\bar{U}_i. \tag{2.4.3}$$

For example, $\bar{\varphi} = 0$ for linearization around the freestream flow. Linearization means that higher order terms are neglected. Taylor expansion gives:

$$\delta U_i = \Psi_{,i}\ ,\ \delta U^i = g^{ij}\Psi_{,i} \tag{2.4.4}$$

$$\begin{aligned}\delta q^2 &= \delta(U^iU_i) = \bar{U}^i\delta U_i + \bar{U}_i\delta U^i = \bar{U}^i\Psi_{,i} + \bar{U}_i g^{ij}\Psi_{,j}\\ &= 2\,\bar{U}^i\,\Psi_{,i}\ ,\end{aligned} \tag{2.4.5}$$

$$\begin{aligned}\delta\rho &= \delta\left(\left\{1 + \frac{\gamma-1}{2}M_\infty^2\,(1-q^2)\right\}^{\frac{1}{\gamma-1}}\right)\\ &= \tfrac{1}{2}M_\infty^2\left\{1 + \frac{\gamma-1}{2}M_\infty^2\,(1-\bar{q}^2)\right\}^{\frac{1}{\gamma-1}-1}\delta(-q^2)\\ &= \tfrac{1}{2}M_\infty^2\,(\bar{\rho})^{2-\gamma}\,(-2\bar{U}^i\,\Psi_{,i}) = -\bar{\rho}\,\frac{\bar{U}^i}{\bar{a}^2}\,\Psi_{,i},\end{aligned} \tag{2.4.6}$$

where the relation $(\bar{\rho})^{\gamma-1} = M_\infty^2\bar{a}^2$ following from equations (2.2.13, 2.2.20), has been used.

Perturbation of the flow equation (2.3.8) around its solution $\bar{\varphi}$ gives

$$\delta \left(\frac{1}{h}\frac{\partial}{\partial\xi^i}(\rho h U^i)\right) = \frac{1}{h}\frac{\partial}{\partial\xi^i}(h(\bar{\rho}\delta U^i + \bar{U}^i\delta\rho)) \tag{2.4.7}$$

$$= \frac{1}{h}\frac{\partial}{\partial\xi^i}\left(\bar{\rho}h\left(g^{ij} - \frac{\bar{U}^i\bar{U}^j}{\bar{a}^2}\right)\Psi_{,j}\right).$$

This is the linearized full potential equation in conservation form. In order to determine the character of equation (2.4.7), it is written in a quasi-linear form (which is not in conservation form):

$$\left(g^{ij} - \frac{\bar{U}^i\bar{U}^j}{\bar{a}^2}\right)\Psi_{,ij} = f(\bar{\varphi}), \tag{2.4.8}$$

where $f(\bar{\varphi})$ is not determinant for the character of the equation, Equation (2.4.8) is written in flow-aligned orthonormal coordinates by choosing $\bar{U}^1 \neq 0$, $\bar{U}^2 = 0$, $\bar{U}^3 = 0$ and $g^{ij} = 0$ if $i \neq j$. Then equation (2.4.8) can be written as

$$\left(g^{11} - \frac{\bar{U}^1\bar{U}^1}{\bar{a}^2}\right)\Psi_{,11} + g^{22}\Psi_{,22} + g^{33}\Psi_{,33} = \tag{2.4.9}$$

$$= g^{11}\left(1 - \frac{\bar{q}^2}{\bar{a}^2}\right)\Psi_{,11} + g^{22}\Psi_{,22} + g^{33}\Psi_{,33} =$$

$$= g^{11}(1 - \bar{M}^2)\Psi_{,11} + g^{22}\Psi_{,22} + g^{33}\Psi_{,33} = f(\bar{\varphi}).$$

Equation (2.4.9) is elliptic in the subsonic part of the flow ($\bar{M} < 1$), hyperbolic in the supersonic part of the flow ($\bar{M} > 1$) and locally parabolic at sonic surfaces ($\bar{M} = 1$).

3 GRIDGENERATION

3.1 Gridgeneration process

Curvilinear, boundary conforming (boundary fitted) grids will be used to discretize the physical space around the wing. The computational grid will be of C-H topology (figure 3.1.1), i.e. will consist of a number of grid planes normal to the spanwise direction of the wing; in each grid plane the chordwise (wrap-around) gridlines resemble a "C". The coordinates of the grid points will be generated using shearing transformations, conformal mappings, scaling transformations and transfinite interpolation, employing various stretching functions. The computational domain, which will generally be called computational space in this thesis, will be a rectangular box of 2NI*NJ*NK computational cells. Of these cells, NI*NJ*NK cells are situated in the upper as well as in the lower part of the physical space (i.e. above and below the wing).

The wing is mapped from xyz-space onto the computational space as follows. First a line $(x_s(y), z_s(y))$, called singular line, is chosen inside the wing and extended beyond the wing tip, cf. figure 3.1.2.

Next, the following shearing transformation is carried out to remove the wing-sweep:

$$\begin{aligned} \bar{X} &= x - x_s(y) \\ \bar{Y} &= y \\ \bar{Z} &= z - z_s(y) \end{aligned} \tag{3.1.1}$$

which gives the type of configuration depicted in figure 3.1.3.

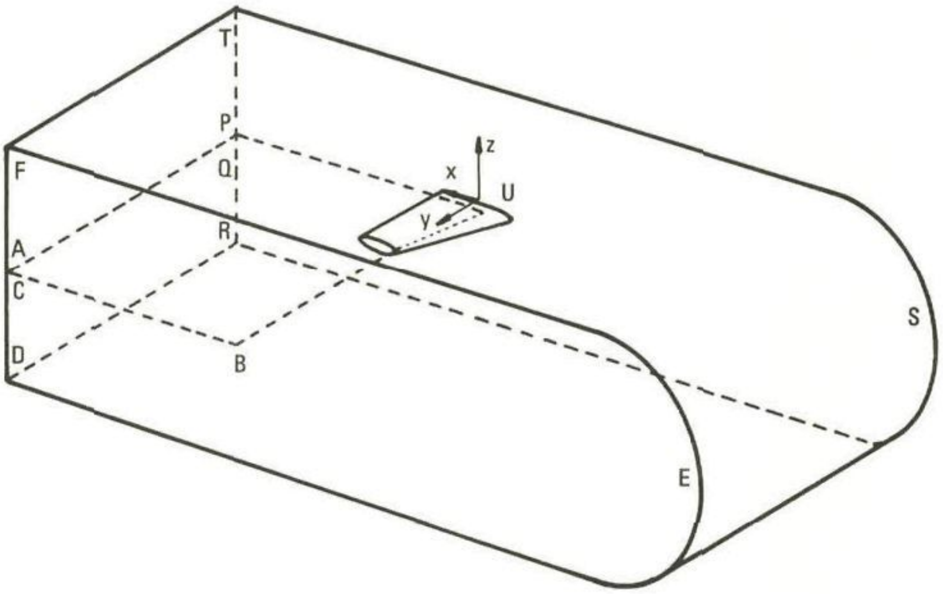
Subsequently, the wing is unfolded by applying in each $\bar{X}\bar{Z}$-plane the so-called square-root mapping, which is a conformal transformation:

$$(\tilde{X} + i\tilde{Z})^2 = 2(\bar{X} + i\bar{Z}) \tag{3.1.2}$$

$$\tilde{Y} = \bar{Y}$$

which results in the situation sketched in figure 3.1.4.

PHYSICAL SPACE



COMPUTATIONAL SPACE

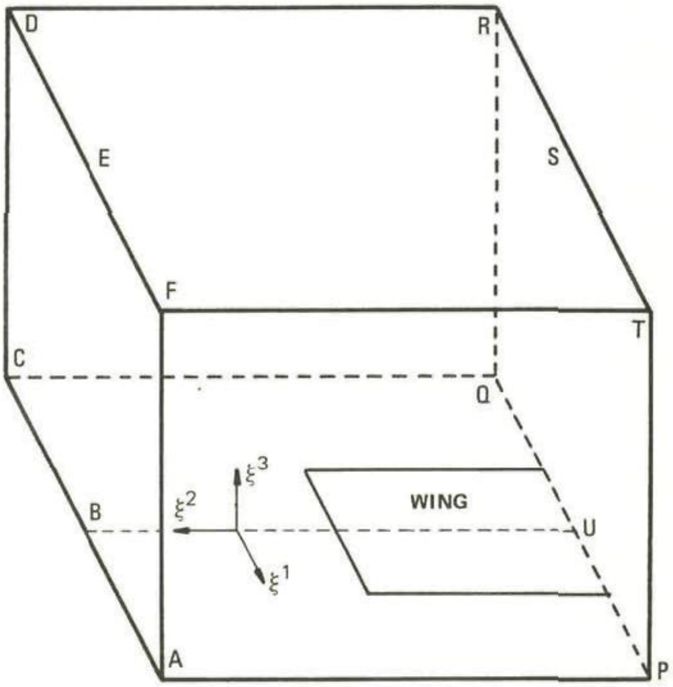


Fig. 3.1.1 Curvilinear boundary conforming grid of C-H topology

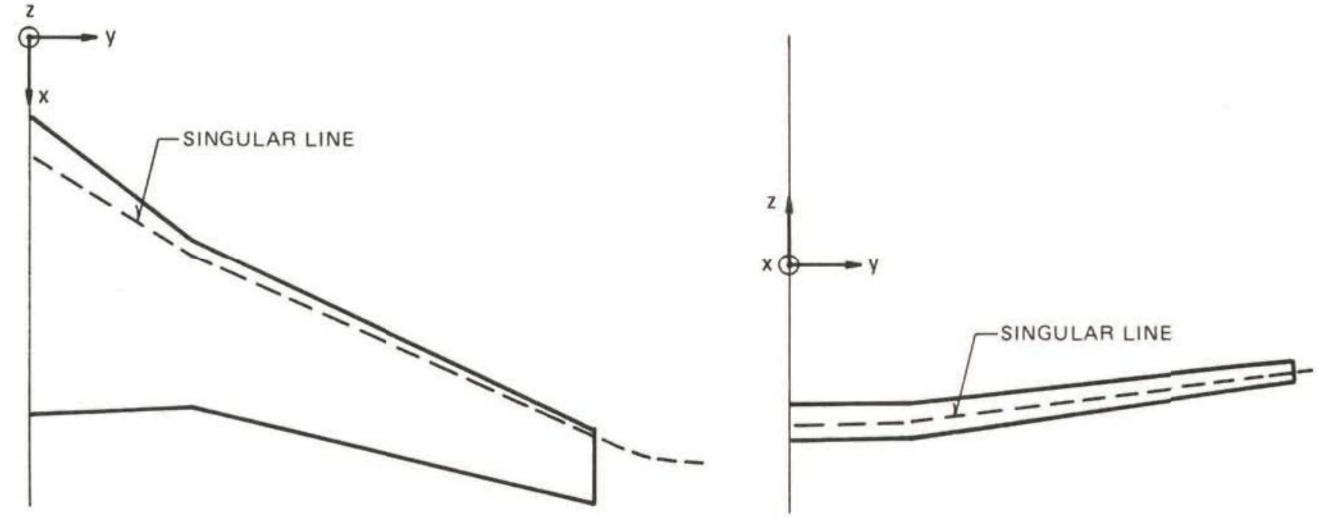


Fig. 3.1.2 Choice of singular line ($x_s(y)$, $z_s(y)$)

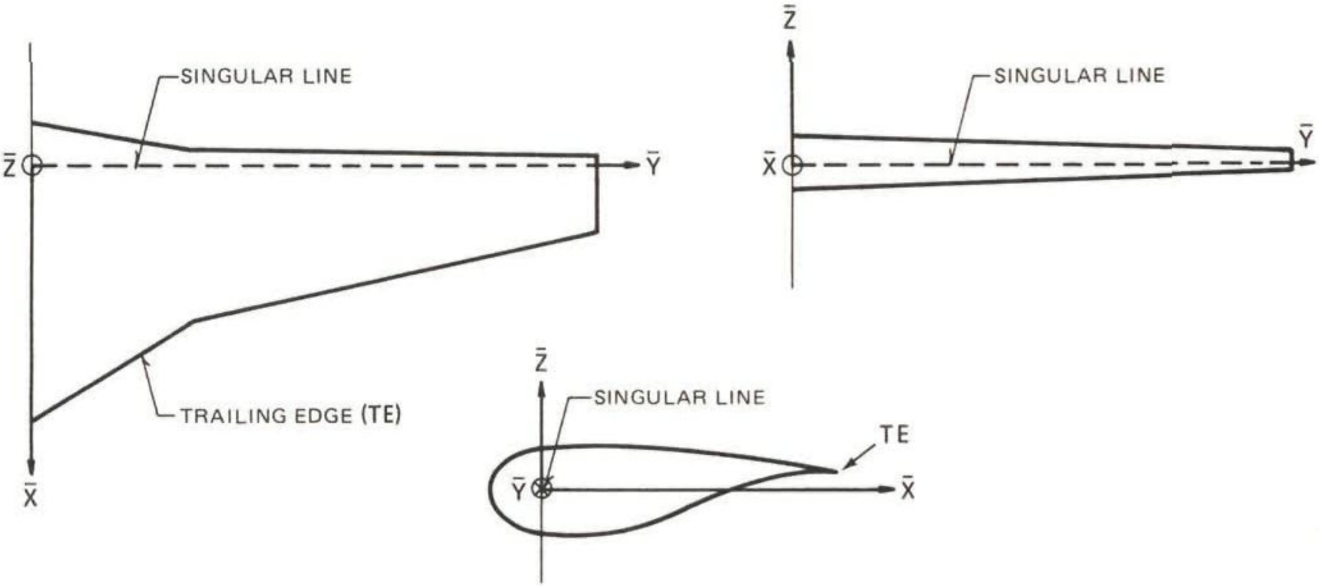


Fig. 3.1.3 Wing after shearing transformation

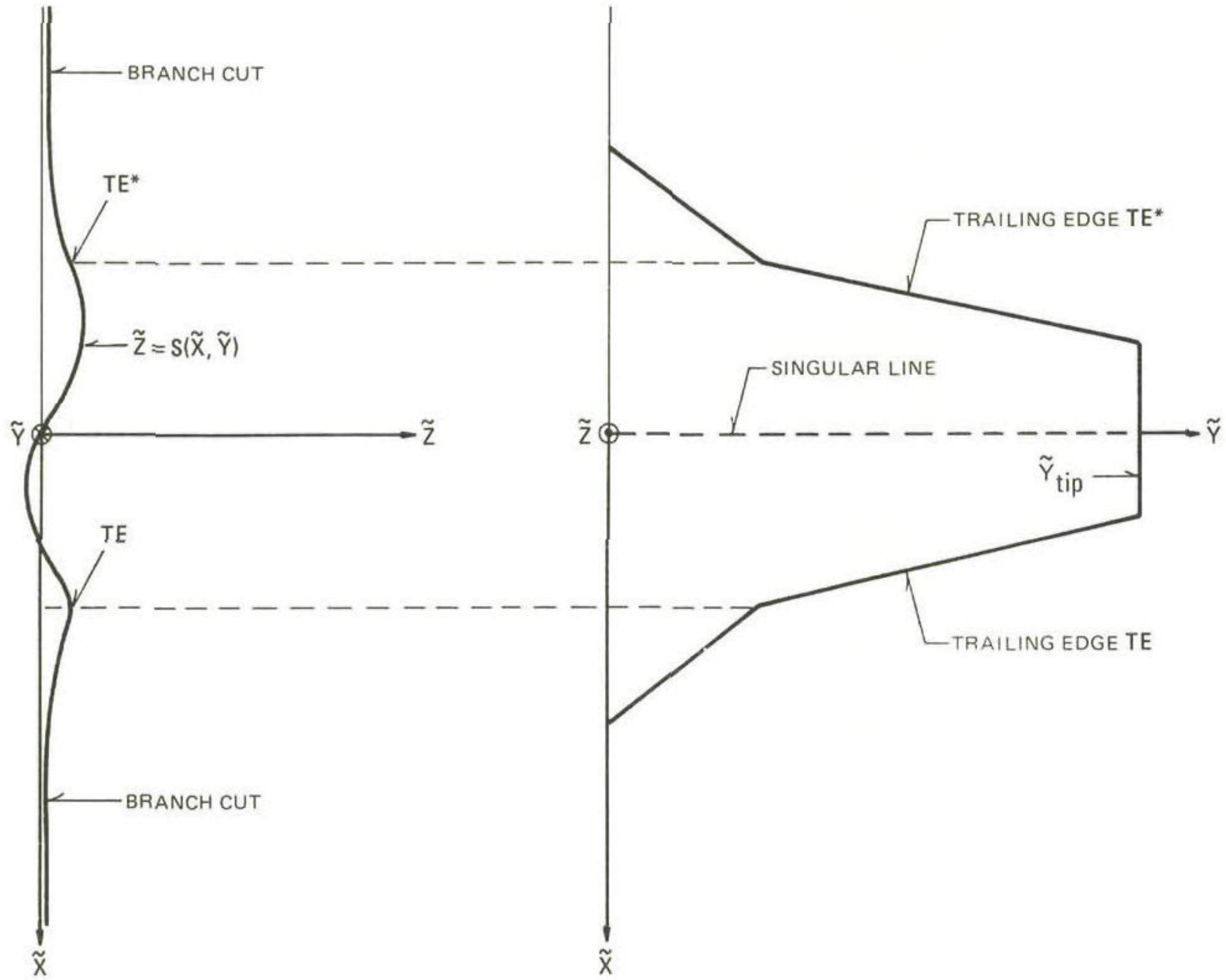


Fig. 3.1.4 Computational space after square-root mapping

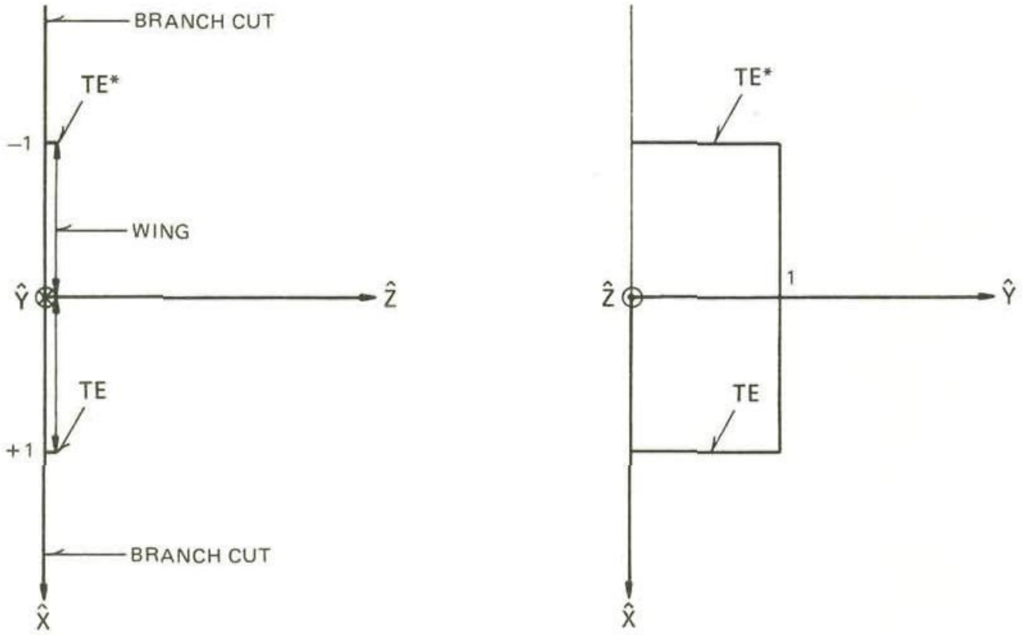


Fig. 3.1.5 Computational space after the mapping described by equation (3.1.3)

The square-root transformation requires the specification of a branch cut. This branch cut starts at the singular line, leaves the wing trailing edge in a direction between the tangents to the upper and lower wing surfaces and must lie in the direction $\vec{s}_\infty$ at downstream infinity, cf. equation (2.2.31). Across the branch cut (artificial) continuity type boundary conditions will be specified in the next section.

Next, a scaling and shearing transformation is applied, which maps the wing profile and the branch cut, i.e. $S(\tilde{X},\tilde{Y})$, into the plane $\tilde{Z} = 0$, and makes the wing planform rectangular:

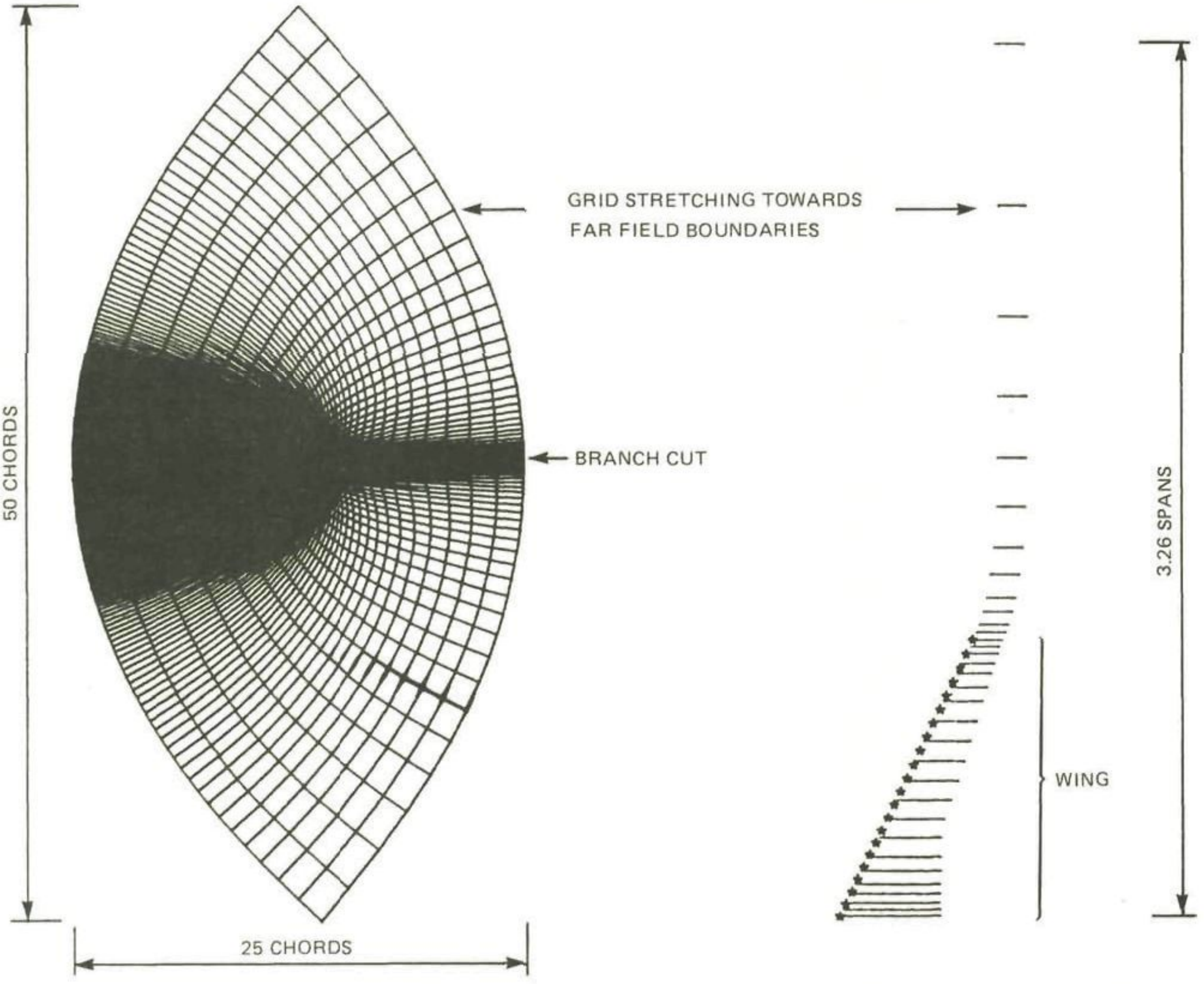
$$\begin{aligned}
&\hat{X} = \frac{\tilde{X}}{C(\tilde{Y})}\,,\ C(\tilde{Y}) = \tilde{X}_{TE} \text{ if } \tilde{X} > 0,\ C(\tilde{Y}) = \tilde{X}_{TE^*} \text{ if } \tilde{X} < 0,\\
&\hat{Y} = \tilde{Y}/\tilde{Y}_{tip}, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.1.3)\\
&\hat{Z} = \tilde{Z} - S(\tilde{X},\tilde{Y}).
\end{aligned}$$

This results in the situation presented in figure 3.1.5.

After the specification of outer dimensions for the $\hat{X}\hat{Y}\hat{Z}$-region and some more scaling and stretching transformations the computional space depicted in figure 3.1.1 is obtained.

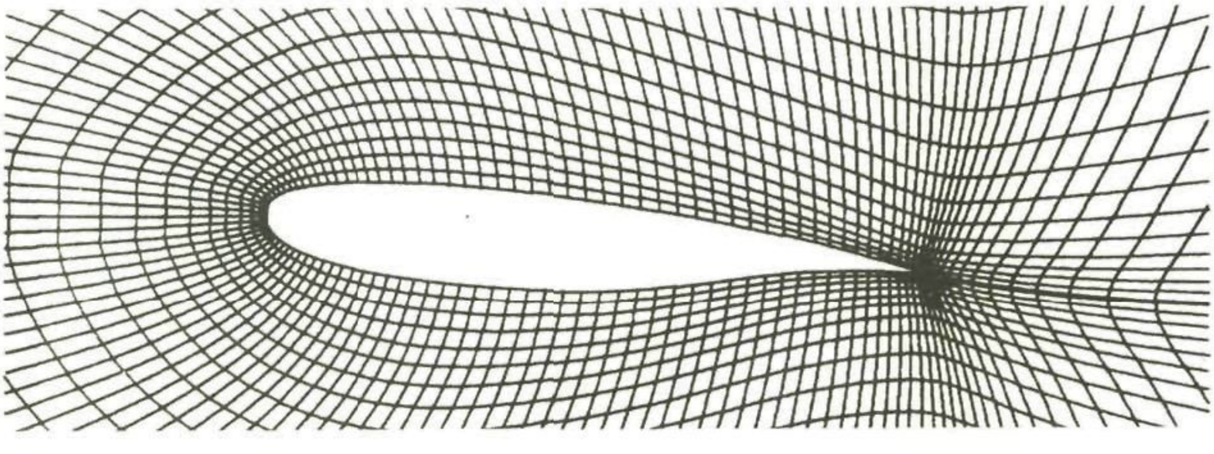
Next, the computational grid is generated as follows. First, the computational space is divided into a number of subspaces. Secondly, grid point distributions are specified on the edges of all computational subspaces. Thirdly, the faces of all computational subspaces are covered with grid points using the grid points on the edges of the subspaces and applying transfinite interpolation. Finally, the interiors of the subspaces are filled with grid points using the grid points at the faces and applying transfinite interpolation. In all steps, a variety of grid stretching facilities is available, while care is taken that geometric derivatives are continuous over computational subbox interfaces and over the branch cut. Finally, the inverse transformation from the discrete $\hat{X}\hat{Y}\hat{Z}$-space to the physical xyz-space is carried out, i.e. the (x,y,z)-coordinates of the grid points are computed.

A typical grid resulting from the gridgeneration process described above is shown in figure 3.1.6. The grid is boundary conforming to the



a) Grid plane perpendicular to the wing ($\xi^1 \xi^3$–plane)

b) Grid plane distribution in spanwise direction (ξ^2–direction)



c) Detail of grid plane near the wing ($\xi^1 \xi^3$–plane)

Fig. 3.1.6 Computational grid around DFVLR-F4 wing

wing and reasonable orthogonal in planes perpendicular to the wing surface. Most skewness occurs in the spanwise direction due to the sweep of the wing (figure 3.1.6b). On the whole, grid skewness is moderate. The grids have very fine, almost square, meshes near the wing surface (figure 3.1.6c) and are highly stretched towards the far field boundaries (figures 3.1.6a, 3.1.6b).

In the gridgeneration process care has been taken to make the grid boundary conforming to the wing. However, no efforts have been made to make the grid boundary conforming to the wing tip, i.e. the wing tip contours are not exactly represented by grid lines. The grid at the wing tip can be made boundary conforming by using a grid which has C-O instead of C-H topology. The generation of such a grid is, however, more complicated.

3.2 Boundary condition on branch cut

The mapping (3.1.2) requires a branch cut to make it one-to-one. Across the branch cut continuity of potential and normal mass flux are prescribed:

$$\begin{aligned} [\![\varphi]\!] &= 0, \\ [\![\rho u^i n_i]\!] &= 0. \end{aligned} \qquad (3.2.1)$$

In most, if not all, computational codes the vortex sheet is chosen to coincide with the branch cut. In that case equation (3.2.1) is replaced by equations (2.2.26a) and (2.2.28).

4 FINITE VOLUME DISCRETIZATION

4.1 General remarks

In this chapter a description will be given of the discretization of the full potential equation (2.3.8) on a curvilinear grid. A so-called finite volume (FV) scheme will be used. Alternative discretizations could have been a finite difference (FD) scheme or a finite element method (FEM) scheme.

When using a FD scheme, derivatives in the partial differential equation (pde) are directly approximated by divided differences on the given grid. When using FV or FEM schemes, the grid is considered to consist of a finite set of cells which cover the computational space. With FV, the scheme follows from an integration of the pde over a control volume; with FEM the discretization follows from a variational formulation of the pde over a control volume. FEM schemes are often applied using unstructured grids, while FV schemes are usually applied using structured grids, i.e. each interior cell has the same ordering with respect to its neighbours. This restriction has the advantage that the generation of the system matrix is cheaper and that faster solution methods are available. Compared with the FD scheme, the FV scheme can best be considered as a well chosen FD scheme, which been formulated such, that it facilitates an efficient implementation, viz. requiring only one (expensive) velocity/density evaluation per cell, while a FD scheme will generally require three such evaluations. Compared with both the FD and the FEM scheme, the FV scheme has the advantage that a conservative scheme, admitting weak solutions of the pde (shocks), is very easy to obtain. The latter is essential to obtain the correct shock strength (see Holst, Slooff, Yoshihara and Ballhaus [1]).

The FV scheme used in this thesis is a slightly modified version of the scheme introduced by Jameson and Caughey [1]. Well known codes in computational aerodynamics applications, such as FLO27 and FLO30, are based on this scheme. The scheme avoids explicit use of the smoothness of the grid and/or of the solution by performing a proper averaging of geometrical and solution dependent quantities. The scheme is a combination of a basic scheme, which follows from the Bateman variational principle in a systematic way, and a correction term (called coupling term), to avoid decoupling of even and odd grid points. In this thesis a coupling term is presented which, compared with the coupling term by Jameson and Caughey [1], retains the favourable properties of the original scheme with respect to avoiding explicit use of smoothness properties of the grid and the flow solution.

4.2 Notation

In this chapter the following notation will be used:

- Indices referring to grid points will be abbreviated as follows, e.g.:

$$(\ldots\ldots)_{i+p,j+q,k+r} \quad \text{is abbreviated to } (\ldots\ldots)_{pqr}\,, \qquad (4.2.1)$$
$$(\ldots\ldots)_{i+p+1,j+q,k+r} \quad \text{is abbreviated to } (\ldots\ldots)_{p+1,qr}.$$

- $(ab)_{pqr}$ is the abbreviation of $a_{pqr}\,b_{pqr}$.
- δ_i and $\overleftarrow{\delta}_i$ are the central and backward difference operators which, for the ξ^1-direction, are defined as follows:
  - central difference operator: $\delta_1\varphi_{000} = \varphi_{\frac{1}{2}00} - \varphi_{-\frac{1}{2}00}$, (4.2.2)
  - backward difference operator: $\overleftarrow{\delta}_1\varphi_{000} = \varphi_{000} - \varphi_{-100}$, (4.2.3)

  The backward difference operator will also be referred to as upwind difference operator.
  It is convenient to introduce also the central difference operator $\tilde{\delta}_i$ over two points which, for the ξ^1-direction, is defined by

$$\tilde{\delta}_1\varphi_{000} = \tfrac{1}{2}(\varphi_{100} - \varphi_{-100}). \qquad (4.2.4)$$

- μ_i is the averaging operator in the i-direction which, for the ξ^1-direction, is defined by

$$\mu_1\varphi_{000} = \tfrac{1}{2}(\varphi_{\frac{1}{2}00} + \varphi_{-\frac{1}{2}00}). \qquad (4.2.5)$$

- Products of averaging operators will be denoted by

$$\mu_{12} \equiv \mu_1\mu_2 \equiv \mu_2\mu_1. \qquad (4.2.6)$$

Averaging and differencing operators commute:

$$\mu_i \delta_j \equiv \delta_j \mu_i. \qquad (4.2.7)$$

- In case an operand is composed of more than one variable, it will be *placed between parentheses*, e.g. $\mu_1(a+b)$ *or* $\mu_1(ab)$.

- $\bar{\delta}_i$ is a composite differencing/averaging operator:

$$\bar{\delta}_i \equiv \delta_i \mu_j \mu_k, \ i \neq j \neq k \neq i. \qquad (4.2.8)$$

4.3 Description of computational grid

The flow equation, equation (2.3.8), is solved in curvilinear coordinates ξ^α, $\alpha = 1,2,3$. The grid is given by a set of computational cells, called finite volumes, in x^i-space, which are the images of unit cubes in ξ^α-space (figure 4.3.1). These unit cubes are called primary cells.

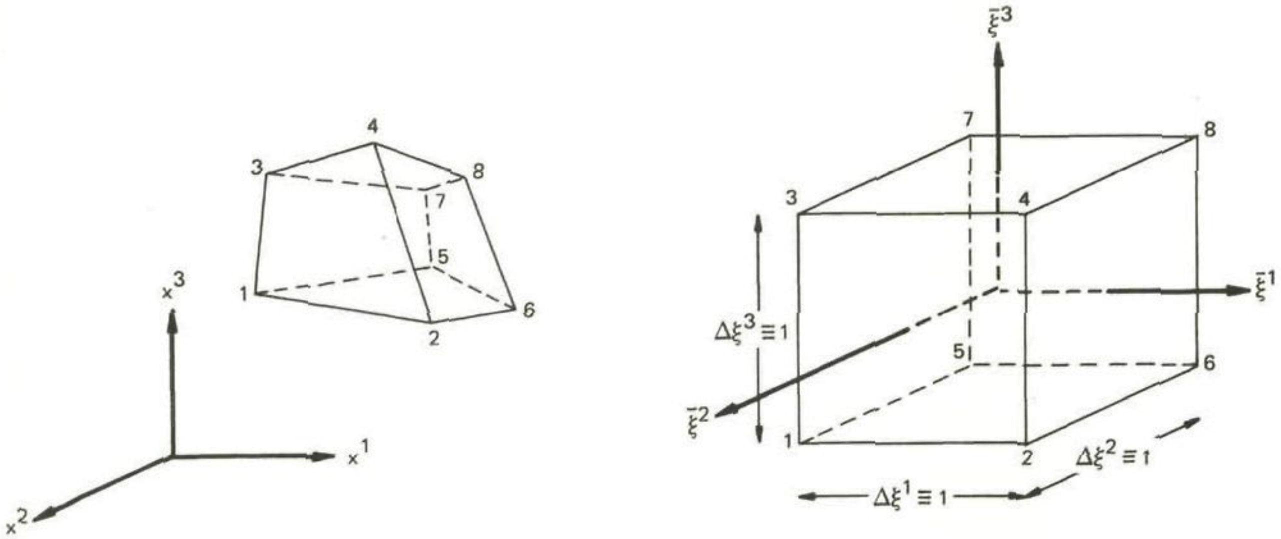


Fig. 4.3.1 Finite volumes in x^i-space and ξ^i-space

The boundaries of the computational domain are chosen to coincide with primary cell boundaries.

For each primary cell, a local Cartesian coordinate system $(\bar{\xi}^1,\bar{\xi}^2,\bar{\xi}^3)$ is defined with its origin at the cell centre. The eight corners have local coordinates $(\bar{\xi}^1,\bar{\xi}^2,\bar{\xi}^3) = (\pm\frac{1}{2},\pm\frac{1}{2},\pm\frac{1}{2})$.

The velocity disturbance potential φ and the coordinates x^i are given in the corners of the cell. In each computational cell trilinear interpolation is used to compute derivatives of variables:

$$x^\alpha = 8 \sum_{i=1}^{8} x_i^\alpha \left(\frac{1}{4} + \bar{\xi}_i^1\bar{\xi}^1\right)\left(\frac{1}{4} + \bar{\xi}_i^2\bar{\xi}^2\right)\left(\frac{1}{4} + \bar{\xi}_i^3\bar{\xi}^3\right), \tag{4.3.1a}$$

$$\varphi = 8 \sum_{i=1}^{8} \varphi_i \left(\frac{1}{4} + \bar{\xi}_i^1\bar{\xi}^1\right)\left(\frac{1}{4} + \bar{\xi}_i^2\bar{\xi}^2\right)\left(\frac{1}{4} + \bar{\xi}_i^3\bar{\xi}^3\right), \tag{4.3.1b}$$

where the subscripts i = 1, ..., 8 refers to the cornerpoints of the cell.

Geometrical quantities will be computed as follows. In primary cell centers the computation of the elements of the Jacobi matrix, equation (2.3.2), follows directly from equation (4.3.1a) and is given by

$$\left(\frac{\partial x^i}{\partial \xi^\alpha}\right)_{000} = \bar{\delta}_\alpha x^i_{000}. \tag{4.3.2}$$

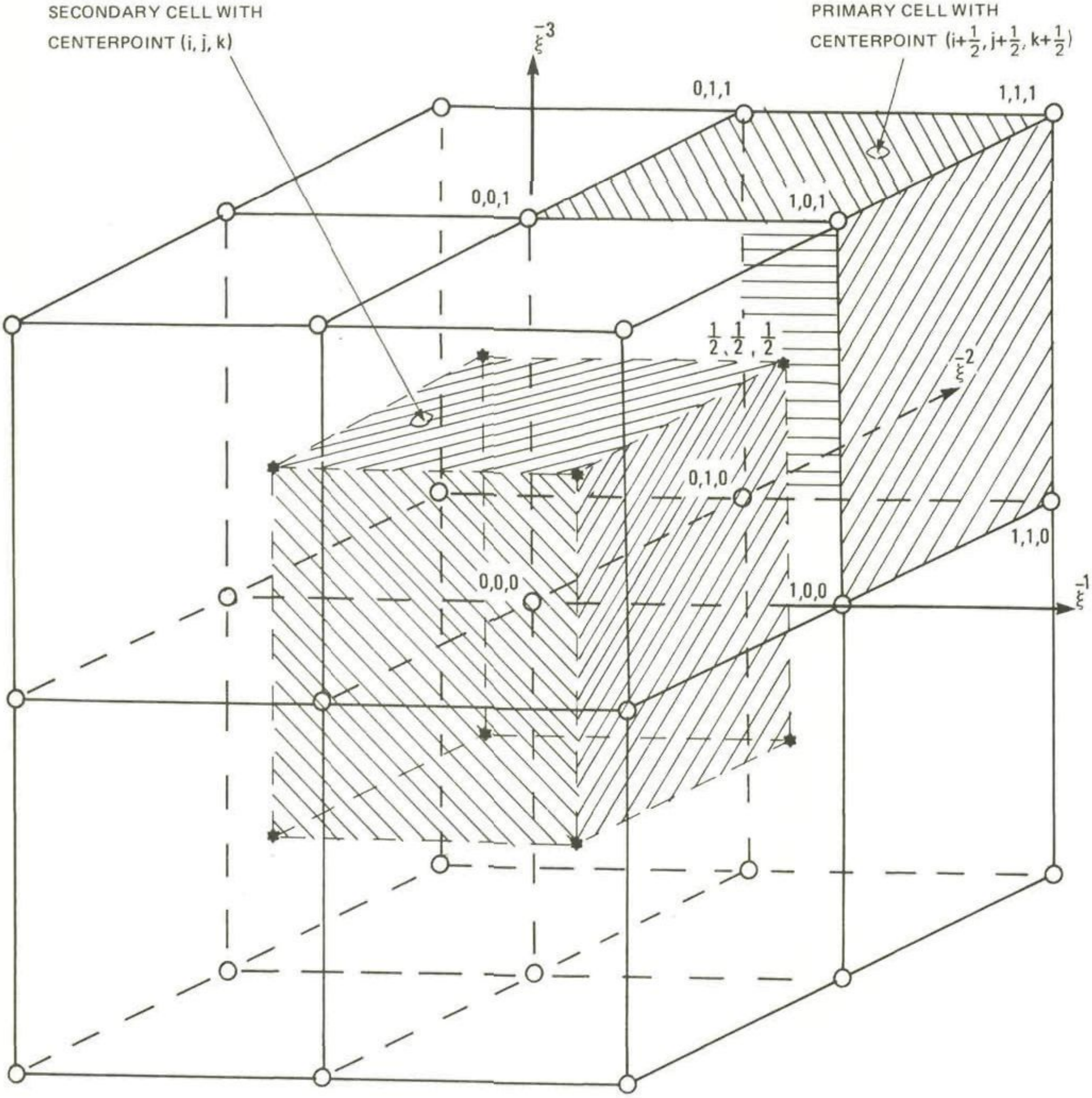
In boundary grid points the geometrical quantities are computed in a standard way using central differences whenever possible. For example at a $\xi^3 = 0$ boundary these quantities are given by

$$\left(\frac{\partial x^i}{\partial \xi^\alpha}\right)_{000} = \tilde{\delta}_\alpha x^i_{000}, \quad \alpha \neq 3, \tag{4.3.3a}$$

$$\left(\frac{\partial x^i}{\partial \xi^3}\right)_{000} = x^i_{001} - x^i_{000}. \tag{4.3.3b}$$

The Jacobian h, representing the volume of the primary cell in x^i-space, and the contravariant metric tensor g^{ij} can be computed subsequently using equations (2.3.3) and (2.3.4) respectively.

The computation of potential derivatives in primary cell centers follows directly from equation (4.3.1b) and is given by


SECONDARY CELL WITH CENTERPOINT (i, j, k)
PRIMARY CELL WITH CENTERPOINT $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$
$\bar{\xi}^3$
0,1,1
1,1,1
0,0,1
1,0,1
$\frac{1}{2}, \frac{1}{2}, \frac{1}{2}$
$\bar{\xi}^2$
0,1,0
1,1,0
0,0,0
1,0,0
$\bar{\xi}^1$

GIVEN IN FLUX POINT ✱ $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$	ρ	DENSITY	φ	PERTURB. POTENTIAL	GIVEN IN GRID POINT ○ (i, j, k)
	h	JACOBIAN	x^i	COORDINATES	
	U^i	CONTR. VELOCITIES	$L^h\varphi^h$	FLOWEQUATION RESIDUAL	
	g^{ij}	CONTR. METRIC TENSOR	$u_{\infty i}$	FREESTREAM VALUES	

Fig. 4.3.2 Three-dimensional arrangement of primary and secondary cells in ξ^{α}-space

$$(\frac{\partial\varphi}{\partial\xi^{\alpha}})_{000} = \bar{\delta}_{\alpha}\varphi_{000}. \qquad (4.3.4)$$

In each boundary grid point, e.g. a $\xi^3 = 0$ boundary, potential derivatives are computed similarly to geometrical quantities, equation (4.3.3), from

$$(\frac{\partial\varphi}{\partial\xi^{\alpha}})_{000} = \tilde{\delta}_{\alpha}\varphi^{i}_{000},\ \alpha \neq 3, \qquad (4.3.5a)$$

$$(\frac{\partial\varphi}{\partial\xi^{3}})_{000} = \varphi_{001} - \varphi_{000}. \qquad (4.3.5b)$$

Freestream velocity components $u_{\infty\alpha}$ can be specified in each grid point (and will generally be constant). Using trilinear interpolation, their value in primary cell centers is computed by averaging:

$$(u_{\infty\alpha})_{000} = \mu_{123}(u_{\infty\alpha})_{000}. \qquad (4.3.6)$$

For the purpose of discretizing the flow equation, secondary cells are introduced. The secondary cell centers are the cornerpoints of the primary cells; the cornerpoints of the secondary cells are the primary cell centers, see figure 4.3.2.

4.4 Discretization of flow equation

4.4.1 General remarks

The flow equation (2.3.8) is represented symbolically as

$$L\varphi = 0. \qquad (4.4.1)$$

The discretization of this equation is denoted as

$$L^{h}\varphi^{h} = 0. \qquad (4.4.2)$$

In subsection 4.4.2 the description of the $O(h^2)$-accurate operator L^h will be given. In the subsections 4.4.3, 4.4.4 and 4.4.5 the discretization scheme will be extended as follows,

$$(L^h + L^h_{coupl} + L^h_{bias})\ \varphi^h = f^h_\infty, \tag{4.4.3}$$

where L^h_{coupl} is an $O(h^3)$-accurate operator to avoid decoupling of even and odd grid points (subsection 4.4.3), L^h_{bias} is an $O(h)$-accurate operator used in the supersonic region of the flow to suppress expansion shocks (subsection 4.4.4) and f^h_∞ is an $O(h^2)$-accurate source-term which compensates the freestream discretization error in the far field (subsection 4.4.5).

4.4.2 Finite volume discretization

The basic finite volume (FV) scheme follows directly from the Bateman variational principle, formulated here as follows:

"A stationary point of the functional

$$J[\varphi] = \iiint_\Omega p h d\xi^1 d\xi^2 d\xi^3 - \iint_{\partial\Omega_F} \varphi F_n h d\partial\Omega \tag{4.4.4}$$

with p defined by

$$p = \frac{1}{\gamma M^2_\infty}\ \rho^\gamma \tag{4.4.5}$$

and

$$\rho = \{1 + \frac{\gamma-1}{2} M^2_\infty\ (1 - U^i U_i)\}^{\frac{1}{\gamma-1}}, \tag{4.4.6}$$

$$U^i = g^{ij}U_j;\ U_i = U_{\infty i} + \frac{\partial\varphi}{\partial\xi^i}. \tag{4.4.7}$$

is a solution of the full potential equation."

In order to satisfy the boundary conditions, in equation (4.4.4) F_n is the prescribed mass flux on $\partial\Omega_F$, and φ is chosen to belong to the class of continuous and almost everywhere differentiable functions satisfying $\varphi = \varphi_o$ on $\partial\Omega\backslash\partial\Omega_F$. The boundary $\partial\Omega$ is assumed to be smooth.

The Bateman principle is justified as follows. Variation of equation (4.4.4) with respect to φ gives, using equation (2.4.6),

$$\delta J[\varphi] = J[\varphi + \delta\varphi] - J[\varphi] =$$

$$= - \iiint_{\Omega} \rho h U^i \frac{\partial \delta\varphi}{\partial \xi^i}\, d\xi^1 d\xi^2 d\xi^3 - \iint_{\partial\Omega_F} \delta\varphi F_n h\, d\partial\Omega = 0 \qquad (4.4.8)$$

The variation of J is taken over φ sufficiently smooth, whence partial integration is allowed in (4.4.8). Application of the divergence theorem gives

$$\iiint_{\Omega} \delta\varphi \frac{\partial}{\partial \xi^i}(\rho h U^i)\, d\xi^1 d\xi^2 d\xi^3 + \iint_{\partial\Omega + \partial\Omega_s} \delta\varphi \rho h u^i n_i\, d\partial\Omega - \iint_{\partial\Omega_F} \delta\varphi F_n h\, d\partial\Omega = 0, \qquad (4.4.9)$$

where n_i is the outward unit normal on the boundary $\partial\Omega + \partial\Omega_s$ and $\partial\Omega_s$ are surfaces across which $\rho h U^i$ is not differentiable, such as shocks. Equation (4.4.9) must be satisfied for all smooth enough $\delta\varphi$, and hence

$$\frac{\partial}{\partial \xi^i}(\rho h U^i) = 0 \qquad \text{in the interior of } \Omega$$

$$\rho u^i n_i = F_n \qquad \text{on } \partial\Omega_F \qquad (4.4.10)$$

$$\rho u^i n_i \text{ must be continuous on } \partial\Omega_s$$

$$\varphi = \varphi_o \ (\text{i.e. } \delta\varphi \equiv 0) \qquad \text{on } \partial\Omega \backslash \partial\Omega_F$$

Consequently, the flow indeed satisfies the boundary conditions and the full potential equation (2.3.8) outside shocks, while mass is conserved across shocks.

The Bateman variational principle will now be used to derive a discretization of the full potential equation. The potential φ will be chosen as follows:

$$\varphi = \varphi_{i,j,k}\, \lambda_{i,j,k} \qquad (4.4.$$

where the pyramid function $\lambda_{i,j,k} = 1$ in grid point (i,j,k), zero in all other grid points and of trilinear form. The values of $\varphi_{i,j,k}$ are such that $\varphi = \varphi_0$ in the grid points of $\partial\Omega\backslash\partial\Omega_F$. The requirement $\partial J[\varphi]/\partial\varphi_{\bar{i},\bar{j},\bar{k}} = 0$ then gives (switching to Greek indices to denote differentiation)

$$- \iiint_{\Omega} \rho h U^{\alpha} \frac{\partial\lambda_{i,j,k}}{\partial\xi^{\alpha}} d\xi^1 d\xi^2 d\xi^3 - \iint_{\partial\Omega_F} \lambda_{\bar{i},\bar{j},\bar{k}} F_n h d\partial\Omega$$

$$= - \iiint_{\Omega} \rho h g^{\alpha\beta} \left(U_{\infty\beta} + \varphi_{i,j,k} \frac{\partial\lambda_{i,j,k}}{\partial\xi^{\beta}}\right) \frac{\partial\lambda_{\bar{i},\bar{j},\bar{k}}}{\partial\xi^{\alpha}} d\xi^1 d\xi^2 d\xi^3 +$$

$$- \iint_{\partial\Omega_F} \lambda_{\bar{i},\bar{j},\bar{k}} F_n h d\partial\Omega = 0. \qquad (4.4.12)$$

The FV scheme follows directly from equation (4.4.12) by considering the contributions from the eight primary cells of which $(\bar{i},\bar{j},\bar{k})$ is a cornerpoint (figure 4.3.2) to the left integral of (4.4.12). It follows from equations (4.4.9) and (4.4.12) that the FV scheme represents an integration of the full potential equation over a control volume, viz. *the eight primary cells of which* $(\bar{i},\bar{j},\bar{k})$ *is a cornerpoint.*

In the interior of Ω the fully equivalent finite difference formulation of equation (4.4.12) is

$$\frac{1}{\mu_{123} h_{000}} \tilde{\delta}_i (\rho h U^i) = 0, \qquad (4.4.13)$$

with ρ *given by equation (4.4.6) and* U^i *given by equation (4.4.7).* The derivation of geometrical and solution dependent quantities, necessary to compute ρ, U^i and h, has been given in section 4.3.

Equation (4.4.13) represents a discrete mass balance over a secondary cell. Consequently, mass is conserved over a secondary cell : the scheme is conservative. The contravariant (mass) flux tensor $F^i = \rho h U^i$ is obtained on each face of the secondary cell by averaging four contravariant *mass flux tensors computed in the corners of the cell face.*

Figure 4.4.1 gives a graphical interpretation of the scheme (4.4.13) at a boundary with prescribed Neumann boundary conditions.

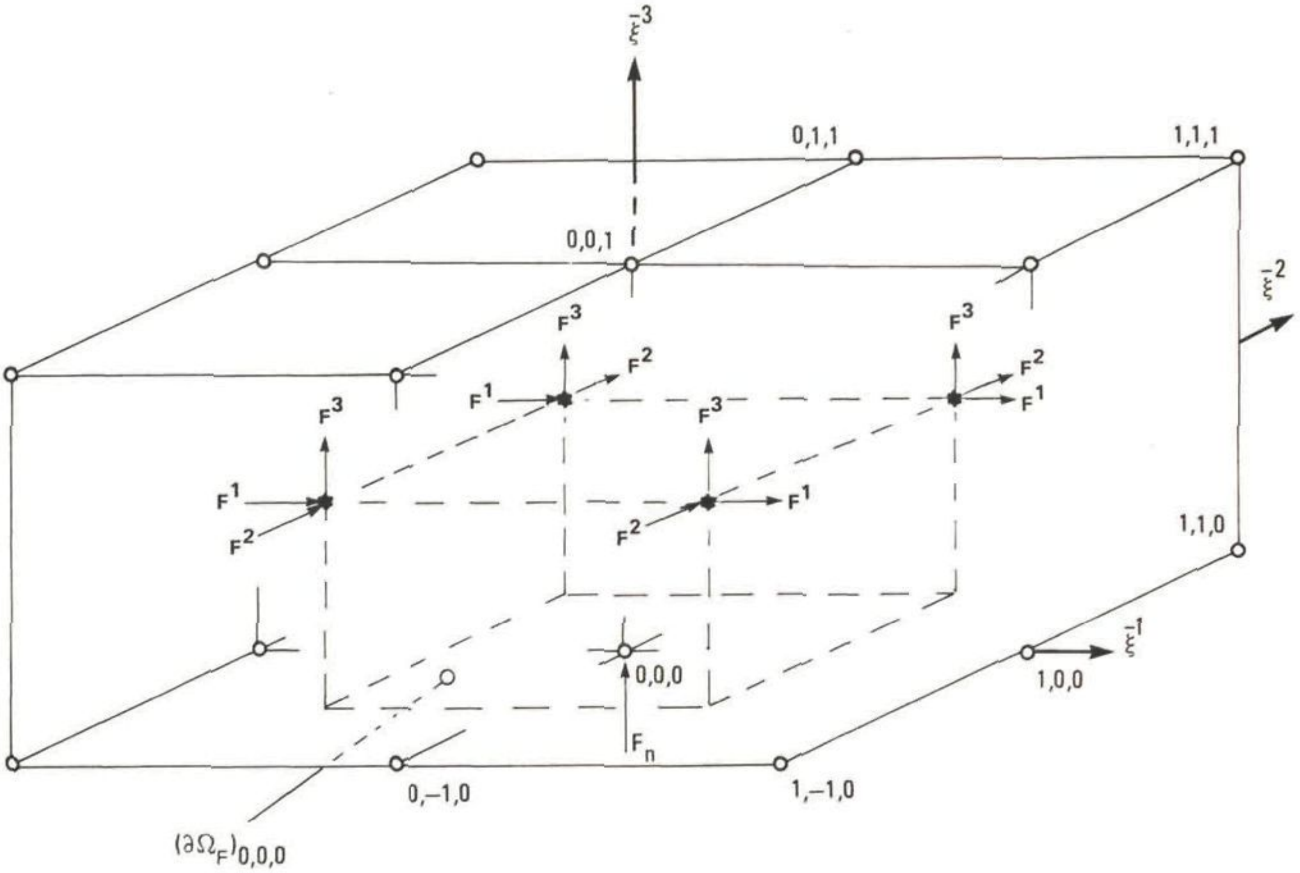


Fig. 4.4.1 Three-dimensional representation of the FV-scheme at a boundary with prescribed Neumann boundary conditions

4.4.3 Coupling operator

The scheme just presented has the undesirable property that odd and even grid points are decoupled when the scheme is applied to incompressible flow ($M_\infty = 0$) *on a uniform grid with* square meshes. In the two-dimensional case we then obtain the so-called rotated Laplace scheme

$$\tfrac{1}{2}\,(\varphi_{-1,-1} + \varphi_{1,-1} - 4\varphi_{00} + \varphi_{-1,1} + \varphi_{1,1}) = 0, \qquad (4.4.14)$$

which is known to produce oscillatory ("sawtooth") solutions, because odd and even grid points are decoupled.

Although the decoupling problem formally cannot occur for compressible flow, where odd and even grid points are coupled implicitly in a nonlinear manner by the density relation, the phenomenon is suppressed rigorously by adding a so-called explicit coupling term to the scheme.

It can easily be verified that on a uniform grid with square meshes the rotated Laplace scheme, equation (4.4.14), and the standard Laplace scheme

$$\varphi_{-1,0} + \varphi_{0,-1} - 4\varphi_{00} + \varphi_{0,1} + \varphi_{1,0} = 0 \qquad (4.4.15)$$

are related as follows:

$$\underbrace{\bar{\delta}_i(\delta^{ij}\bar{\delta}_j\varphi)}_{\substack{\text{rotated}\\\text{Laplace}\\\text{scheme}}} - \underbrace{\varepsilon\,\delta^{kl}\delta_k\delta_i(\delta^{ij}\delta_l\delta_j\varphi)}_{\text{coupling term}} = \underbrace{\delta_i(\delta^{ij}\delta_j\varphi)}_{\substack{\text{standard}\\\text{Laplace}\\\text{scheme}}},\ i \neq k \qquad (4.4.16)$$

with $\varepsilon = \frac{1}{2}$. Note that $\varepsilon = 1/3$ yields the nine point fourth order accurate Laplace scheme.

There are two ways to implement the above coupling term:

1. Shift in every secondary cell the calculation of the mass flux tensor F^i on a ξ^i-face (face perpendicular to the ξ^i-direction) from the cornerpoints to the centerpoint of that ξ^i-face. For example, on a ξ^1-face with centerpoint $(\frac{1}{2},0)$ the flux tensor $\tilde{F}^1_{\frac12\frac12} = \delta^{1j}\bar{\delta}_j\varphi_{\frac12\frac12}$ is modified by adding a term

$$\tilde{F}^1_{\frac12 0} - \tilde{F}^1_{\frac12\frac12} = \tilde{U}^1_{\frac12 0} - \tilde{U}^1_{\frac12\frac12} = \delta^{1j}\delta_j\varphi_{\frac12 0} - \delta^{1j}\bar{\delta}_j\varphi_{\frac12\frac12} \qquad (4.4.17)$$

   This approach is followed by Jameson and Caughey [1]. (In this reference also the three-dimensional formulation can be found.)

2. Shift in every secondary cell the calculation of the divided ξ^i-difference of the potential, i.e. $\bar{\delta}_i\varphi$, cf. equation (4.3.4), from the cornerpoints of a ξ^i-face to the centerpoint of that ξ^i-face. For example, on a ξ^1-face with centerpoint $(\frac{1}{2},0)$ the divided potential difference $\bar{\delta}_i\varphi_{\frac12\frac12}$, i=1, is modified by adding a term

$$\delta_i\varphi_{\frac12 0} - \bar{\delta}_i\varphi_{\frac12\frac12}\ ,\ i = 1. \qquad (4.4.18)$$

   This approach is followed in this thesis.

In case of in incompressible flow (M_∞=0) on a uniform grid with square meshes, the above two implementations are identical. In case of compressible flow on a curvilinear grid, however, the first implementation is costly, because the calculation of the mass flux tensor in the centerpoint of a ξ^i-face ($\tilde{F}^1_{\frac12 0}$ in equation (4.4.17)) requires extra evaluations of the density. Therefore, the coupling term is added by Jameson and Caughey [1] in a linearized form (see also section 2.4). For example, equation (4.4.17) is written as

$$
\begin{aligned}
F^{1}_{\frac{1}{2}0} - F^{1}_{\frac{1}{2}\frac{1}{2}} &= (\rho U^{1})_{\frac{1}{2}0} - (\rho U^{1})_{\frac{1}{2}\frac{1}{2}} \\
&\approx -\frac{1}{2}\left(\frac{\partial}{\partial \xi^{j}}(\rho U^{i})_{\frac{1}{2}\frac{1}{2}}\right) \qquad , i = 1,\ j = 2, \\
&\approx -\frac{1}{2}\left(\bar{\rho}\left(g^{ik} - \frac{\bar{U}^{i}\bar{U}^{k}}{\bar{a}^{2}}\right) \frac{\partial}{\partial \xi^{j}} \frac{\partial}{\partial \xi^{k}} \varphi\right)_{\frac{1}{2}\frac{1}{2}} , i = 1,\ j = 2, \\
&\approx -\frac{1}{2}\, \bar{\rho}_{\frac{1}{2}\frac{1}{2}} \left(g^{ik} - \frac{\bar{U}^{i}\bar{U}^{k}}{\bar{a}^{2}}\right)_{\frac{1}{2}\frac{1}{2}} \bar{\delta}_{j}\bar{\delta}_{k}\varphi_{\frac{1}{2}\frac{1}{2}} , i = 1,\ j = 2. \qquad (4.4.19)
\end{aligned}
$$

It is usual to add only the k=i-part of this coupling term to the finite volume scheme. Likewise, only the i-part of the coupling term (4.4.18) is added to the finite volume scheme in the general form of its formulation.

The coupling term presented here has the advantage that it does not use the smoothness of the flow quantities, as is required for the linearization applied in equation (4.4.19). In any case, however, the coupling term should be added multiplied by a small factor κ ($0 \leq \kappa \leq 1$) only, so that its "smearing" effect on the solution is minimized. In the experiments (chapter 8) $\kappa = .1$ has been chosen.

If the grid is sufficiently smooth, the coupling term does not affect the order of accuracy of the FV-scheme, i.e. the scheme remains second order accurate.

4.4.4 Directional bias in hyperbolic regions of the flow

In supersonic regions of the flow, a bias has to be added to the discretization to suppress non-physical expansion shocks. The scheme described here has been adopted from Osher,Hafez,Whitlow [1] and closely resembles the scheme used by Boerstoel and Kassies [1]. In the upwind scheme which follows, the mass flux ρq is retarded against the flow direction if a computational cell is supersonic. At sonic surfaces and shocks special transition operators are applied to suppress expansion shocks and to avoid shock smearing. The contravariant flux tensor

$$F^i = (\rho q)\ h\ \frac{U^i}{q} \qquad (4.4.20)$$

is evaluated in primary cell centers as follows:

. Both actual cell (i,j,k) and upwind cell subsonic:

$$(F^i)_{000} = (\rho q)_{000}\ h_{000}\ (\frac{U^i}{q})_{000}, \qquad (4.4.21)$$

. Actual cell supersonic, upwind cell subsonic (sonic surface):

$$(F^i)_{000} = (\rho^* q^*)_{000}\ h_{000}\ (\frac{U^i}{q})_{000}, \qquad (4.4.22)$$

. Actual cell supersonic, upwind cell supersonic:

$$(F^i)_{000} = (\rho q)_{upwind}\ h_{000}\ (\frac{U^i}{q})_{000}, \qquad (4.4.23)$$

where upwind means retarded over a distance h against the U^i-direction, for example, if $U^i > 0$, $(\rho q)_{upwind} = (\rho q)_{-1,00}$.

. Actual cell subsonic, upwind cell supersonic (shock surface); we consider two possibilities:

- *Enquist-Osher shock approximation:*

$$(F^i)_{000} = ((\rho q)_{000} + (\rho q)_{upwind} - (\rho^* q^*)_{000}) h_{000}\ (\frac{U^i}{q})_{000}, \quad (4.4.24)$$

- Godunov - shock approximation:

$$\sigma = ((\rho q)_{000} - (\rho q)_{upwind})\ /\ (\rho_{000} - \rho_{upwind}), \qquad (4.4.25)$$

<u>if</u> ($\sigma \geqq 0$) <u>then</u> use equation (4.4.23)

<u>else</u> use equation (4.4.21) <u>endif</u>,

where σ is the shock-speed in the face between point (i,j,k) and its upwind point. The use of σ as a supersonic/subsonic delimiter at the shock follows from the relation

$$\frac{\partial(\rho q)}{\partial\rho} = \left(\frac{M^2-1}{M^2}\right) q. \tag{4.4.26}$$

The flux ρq is continuous at the shock and it follows from equation (4.4.26) that it has its maximum at sonic conditions.

It appears that in transonic potential flow calculations the Enquist-Osher shock approximation and the Godunov shock approximation give nearly the same numerical results. In this thesis the Godunov shock approximation will be used, because of its a somewhat better physical background.

Using results from section 2.4 it can be derived that the mass flux biasing scheme described above adds a term

$$\frac{1}{h}\frac{\partial}{\partial\xi^l}\left\{\Delta\xi^k\,\delta^{jl}_{ik}\,\frac{\overleftarrow{\partial}}{\overleftarrow{\partial}\xi^j}(\rho q)\;h\,\frac{U^i}{q}\right\} \approx$$

$$\rho\,(1-M^2)\,\Delta\xi\,(\phi_{ss\xi^1} + \phi_{ss\xi^2} + \phi_{ss\xi^3}) \tag{4.4.27}$$

to the discretized differential equation in the supersonic parts of the flow. This term is usually called an artifical viscosity term. As a result, the discretization becomes first order accurate in supersonic parts of the flow. At shocks the flow is approximated by a weak solution of the partial differential equation.

4.4.5 Freestream consistent scheme

A deficiency of the finite volume scheme described before is that it is not identically satisfied for the freestream flow values, i.e. the substitution of the freestream velocity field leads to a nonzero residual in the flow discretization, equation (4.4.13). There are basically two ways to avoid this problem. One approach is to devise a scheme that is identically satisfied by the freestream condition. For two-dimensional flow such a scheme has been used successfully by Flores, Holst, Kwak and Batiste [1]. However, it is also shown in this reference that generalization to three dimensions will be cumbersome.

Another possible approach is the addition of a source term to the discrete flow equation which cancels the leading term of the freestream flow discretization error. It can be shown that this source term does not to affect the formal accuracy of the scheme.

Here, a hybrid approach will be followed. The density is evaluated such that the correct value $\rho = 1$ is obtained for the freestream flow; the flow equation will be discretized following the second approach by adding a source term.

The freestream velocity distribution is described by

$$u^i = u^i_\infty \, , \; \varphi = 0. \tag{4.4.28}$$

Then, according to equation (2.3.6)

$$u_i = \frac{\bar{\partial}\xi^\alpha}{\bar{\partial}x^i} U_\alpha = \frac{\bar{\partial}\xi^\alpha}{\bar{\partial}x^i} \frac{\bar{\partial}x^\beta}{\bar{\partial}\xi^\alpha} u_{\infty\beta} = u_{\infty i}, \tag{4.4.29}$$

where $\bar{\partial}$ denotes divided differences of geometrical quantities. Consequently

$$q^2 = u^i u_i = u^i_\infty u_{\infty i} = 1 \tag{4.4.30}$$

so that ρ is 1. Hence, the scheme is freestream consistent with respect to the calculation of the density ρ. This is due to the splitting of the velocity in a freestream part $u_{\infty i}$ and a perturbation part $\partial\varphi/\partial x^i$.

A freestream consistent discretization of the flow equation (4.4.13) will be obtained by adding the following source term f_∞ to the right side of the finite volume scheme:

$$f_\infty = \frac{1}{h} \tilde{\delta}_i(\rho_\infty h U^i_\infty). \tag{4.4.31}$$

With the freestream value $\rho_\infty = 1$ and U^i_∞ according to equation (2.3.6) we obtain

$$f_\infty = \frac{1}{h}\,\tilde{\delta}_i\left(h\,\delta^{\alpha\beta}\,\frac{\bar{\partial}\xi^i}{\bar{\partial}x^\beta}\,u_{\infty\alpha}\right) \tag{4.4.32}$$

Consequently, the source term depends only on the type of discretization of the flow equation (2.3.8) and the geometrical quantities (4.4.29), but not on the flow solution. For incompressible flow ($\rho = 1$), the source term f_∞ completely cancels the freestream flow discretization error, while for compressible flow only the leading term of the freestream flow discretization error is cancelled.

It is shown by Flores, Holst, Kwak and Batiste [1] and also by Melnik and Mead [1] that this approach can lead to a substantial improvement in the numerical accuracy of the discretization. The best results are obtained, however, if special schemes are devised which are satisfied identically for freestream conditions, as has been shown by Flores, Holst, Kwak and Batiste [1].

It can be demonstrated that the source term f_∞ is particularly effective in the far field region of the flow, where the grid is coarse and hence the geometric differencing is inaccurate. Near the wing, where the grid is very fine, the influence of the source term f_∞ is negligible.

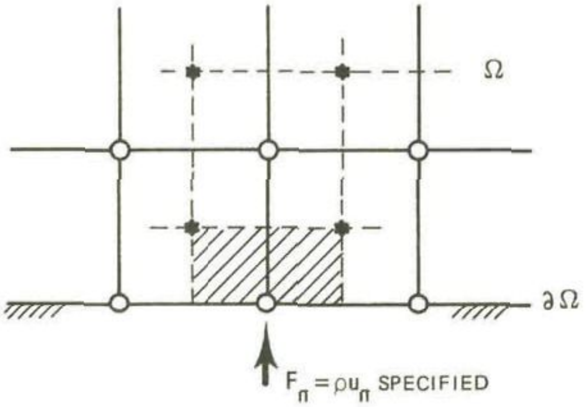
The freestream correction source term f_∞ is applied in interior grid points only, as its implementation in boundary grid points with prescribed Neumann boundary conditions is complex.

4.5 Boundary condition discretization

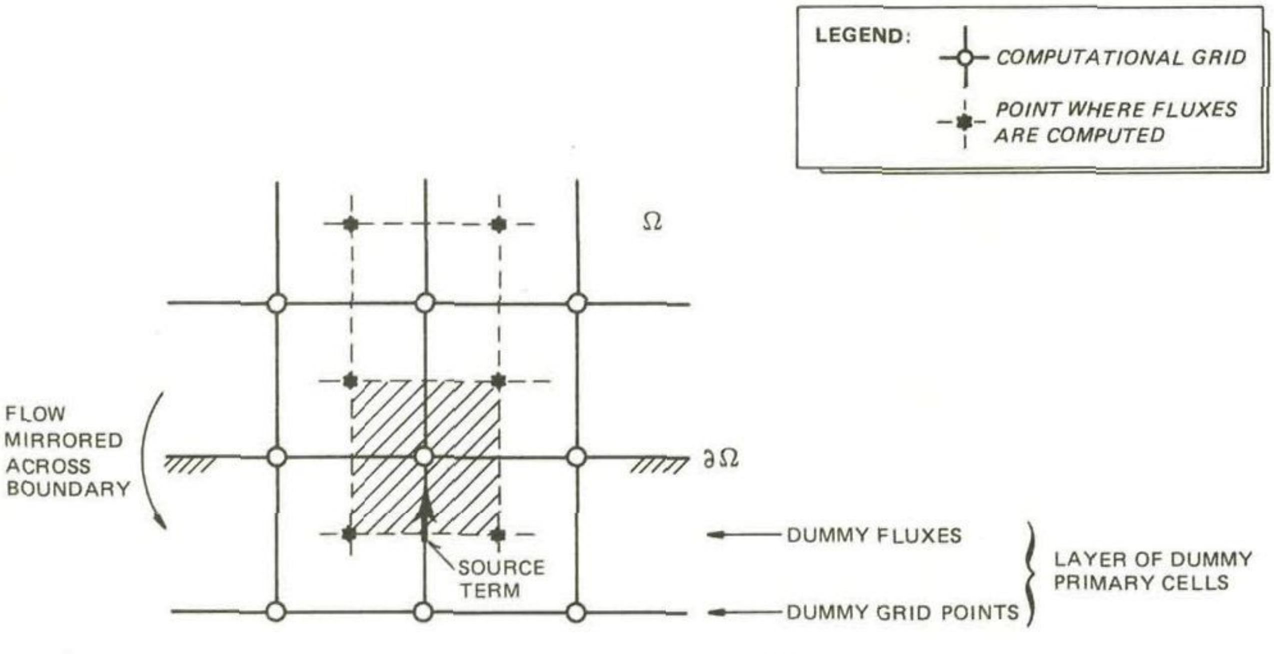
Boundary conditions other than Dirichlet conditions are implemented using a so-called dummy grid point approach. Adjacent to each face of the computational box a layer of dummy primary cells is defined (figure 4.5.1). The flow in the primary cell layer at the boundary is mirrored in the dummy primary cell layer as follows (consider a boundary $\xi^3 = 0$):

$$(g^{ij})_{00,-\frac{1}{2}} := g^{ij}_{00,\frac{1}{2}} \ \underline{\text{if}}\ i = j\ \underline{\text{or}}\ (i \neq 3\ \underline{\text{and}}\ j \neq 3), \tag{4.5.1a}$$

$$(g^{3j})_{00,-\frac{1}{2}} := -g^{3j}_{00,\frac{1}{2}} \ \underline{\text{if}}\ j \neq 3, \tag{4.5.1b}$$



a) Flow discretization at a boundary


LEGEND:
COMPUTATIONAL GRID
POINT WHERE FLUXES ARE COMPUTED
Ω
FLOW MIRRORED ACROSS BOUNDARY
∂Ω
DUMMY FLUXES
SOURCE TERM
DUMMY GRID POINTS
LAYER OF DUMMY PRIMARY CELLS

b) Flow discretization at a boundary using dummy grid points

Fig. 4.5.1 Flow discretization at boundary with prescribed Neumann boundary condition $\rho u_n = F_n$ prescribed (two-dimensional view)

$$(h)_{00,-\frac{1}{2}} := (h)_{00,\frac{1}{2}}, \tag{4.5.1c}$$

$$(U_{\infty i})_{00,-\frac{1}{2}} := (U_{\infty i})_{00,\frac{1}{2}}, \quad i \neq 3 \tag{4.5.1d}$$

$$(U_{\infty 3})_{00,-\frac{1}{2}} := -(U_{\infty 3})_{00,\frac{1}{2}}, \tag{4.5.1e}$$

$$\varphi_{00,-1} := \varphi_{00,1}. \tag{4.5.1f}$$

As a consequence, a homogeneous boundary condition $F_n = \rho u_n = 0$ is automatically satisfied. Therefore in this case the flow discretization over the boundary secondary cell, figure 4.5.1a, given by equation (4.4.12), can be replaced by the flow discretization over a complete boundary secondary cell, figure 4.5.1b, given by equation (4.4.13). It follows that the discretization at the boundary is the same as in the interior, which is a great advantage in code development.

The inhomogeneous boundary condition $\rho u_n = F_n \neq 0$ is implemented by adding a source term to the flow discretization over the complete boundary secondary cell, given by equation (4.4.13):

$$\frac{1}{\mu_{123}h_{000}} \tilde{\delta}_i (\rho h U^i)_{000} = G_n, \tag{4.5.2}$$

where the source term G_n is nonzero only at the boundary. The term G_n is obtained by formulating equation (4.4.12) both for the upper and lower half of the secondary cell sketched in figure 4.5.1:

$$\left[\frac{1}{\mu_{123}h_{000}} \tilde{\delta}_i \; (\rho h U^i)_{000}\right]_{\substack{\text{upper}\\ \text{half}\\ \text{of cell}}} - \frac{1}{\mu_{123}h_{000}} \left(\iint_{\partial\Omega_F} F_n h d\partial\Omega\right) = 0, \tag{4.5.3a}$$

$$\left[\frac{1}{\mu_{123}h_{000}} \tilde{\delta}_i \; (\rho h U^i)_{000}\right]_{\substack{\text{lower}\\ \text{half}\\ \text{of cell}}} + \frac{1}{\mu_{123}h_{000}} \left(\iint_{\partial\Omega_F} (-F_n) h d\partial\Omega\right) = 0, \tag{4.5.3b}$$

where the minus sign in front of F_n in equation (4.5.3b) follows from the mirroring of the flow (and consequently also of the specified inflow) across the boundary. By addition of equations (4.5.3a) and

(4.5.3b) it follows that in equation (4.5.2) G_n is given by

$$G_n = \frac{2}{\mu_{123}h_{000}} (\iint_{\partial\Omega_F} F_n h d\partial\Omega). \tag{4.5.4}$$

Across the branch cut the vortex sheet boundary condition (equation (2.2.26a))

$$[\![\rho u_n]\!] = 0 \tag{4.5.5}$$

is implemented as follows. At the branch cut a transpiration flux $F_n = \rho u_n$ is specified which serves as an in/outflow on the upper/lower side of the cut. The value of F_n is computed from available data near the cut. Subsequently the system is made determinate by prescribing the correct potential jump across the cut (equation (2.2.28)):

$$[\![\varphi]\!] = \Gamma_{t.e.} \tag{4.5.6}$$

In the downstream Trefftz-plane equation (2.2.31) is implemented as follows. We consider a boundary $\xi^1 = 0$ and define instead of equations (4.5.1)

$$(g^{ij})_{-\frac{1}{2},00} := (g^{ij})_{\frac{1}{2},00} , \tag{4.5.7a}$$

$$(h)_{-\frac{1}{2},00} := (h)_{\frac{1}{2},00} , \tag{4.5.7b}$$

$$(U_{\infty i})_{-\frac{1}{2},00} := (U_{\infty i})_{\frac{1}{2},00} ; \tag{4.5.7c}$$

the dummy grid point potential value is obtained from straightforward discretization of equation (2.2.31):

$$(s^i_\infty \frac{\partial \xi^j}{\partial x^i} \frac{\partial \varphi}{\partial \xi^j})_{000} \approx (s^i_\infty (\delta_j x^i)^{-1} \tilde{\delta}_j \varphi)_{000} = 0, \tag{4.5.8}$$

with $(\tilde{\delta}_1 \varphi)_{000}$ defined here as $(\tilde{\delta}_1 \varphi)_{000} = \varphi_{000} - \varphi_{-1,00}$.

Experience has shown that, due to the coarseness of the grid in ξ^1-direction near the Trefftz plane, it is better to use a one sided, first order accurate, formula for $\tilde{\delta}_1$, like the one above, than a central, second order accurate, formula.

5 SOLUTION METHOD

5.1 General remarks

The basic notion in multigrid methods is that a system of equations having N unknowns should be solved in an amount of work proportional to N, i.e. should have complexity N. History has shown that this is not easy to achieve. In table 5.1.1 the complexity of a variety of methods is given for the solution of the Poisson equation using finite difference discretization (except for FFT) on a rectangular grid (see also Sonneveld, Wesseling and de Zeeuw [1]):

TABLE 5.1.1

Complexity of various methods for the solution of the Poisson equation on a rectangle

Method	Complexity (d = problem dimension)	Complexity for $d = 2$, $N = 10^4$ as factor of N	Complexity for $d = 3$, $N = 10^6$ as factor of N
Gauss	$N^{3-2/d}$	10^4 N	10^8 N
SOR	$N^{1+1/d}$	10^2 N	10^2 N
FFT, ADI	NlnN	9.2 N	13.8 N
PCG	$N^{1+1/2d}$	10 N	10 N
MG	N	N	N

In the above table the complexity has been evaluated for two representative problem sizes. Multiplicative constants have been deleted.

The Gauss matrix decomposition method is a well-known direct method. For large N it is however only of academic interest for the type of application under consideration. A reasonable order of complexity is provided by iterative methods such as SOR (Successive Over Relaxation), ADI (Alternating Direction Implicit algorithm), PCG (Preconditioned Conjugate Gradient method) and MG (MultiGrid method) and by the direct method FFT (Fast Fourier Transform). Table 5.1.1 shows that SOR, which was developed about 30 years ago, is reasonably fast, but for large N it cannot compete with ADI, FFT or PCG, which were developed about 20 years ago, or with MG, the development of which started about 10 years ago. The table also shows that MG has the lowest complexity and should be the fastest converging algorithm when N goes to infinity. In practice MG is not significantly faster than ADI, FFT or PCG in many applications for medium-sized problems, as the proportionality factor (multiplicative constant) in front of N is relatively large for MG when compared with the factors of ADI, FFT and PCG. On the other hand, FFT is not applicable when the equation contains variable coefficients, whereas ADI contains parameters whose optimal values are not known in general.

The basis for multigrid methods has already been supplied 20 years ago through exploratory research of the Russian researchers Fedorenko [1] and Bakhvalov [1]. The true potential of the method for practical applications was brought to light ten years ago due to research by Brandt [1] and Hackbush [1], whereas the publications by Nicolaides [1] and Wesseling [1] should also be mentioned in this context. Two-dimensional transonic potential flow has been one of the first realistic applications of multigrid (South and Brandt [1], Fuchs [1]).

In the last ten years the multigrid method has matured and has proved to be a comparatively generally applicable acceleration method for iterative matrix solution algorithms. Applications are still mostly restricted to the area of computational fluid dynamics, but this situation has recently started to change.

An important drawback of multigrid is its relatively complex coding. For time-independent two-dimensional second order elliptic equations general purpose software is nowadays coming available, e.g. MGD1 and MGD5 (Hemker, Wesseling and de Zeeuw [1]), recently incorporated in the NAG library of mathematical subroutines, MG00 (Foerster and Witsch [1]) and BOXMG (Dendy [1]). Also for time-dependent two-dimensional Navier-Stokes-like equations progress has been made (Pau and Lewis [1]). Future use of multigrid as a "black-box" solution method therefore comes within reach.

5.2 Multigrid method

5.2.1 Description of multigrid method

An excellent introduction to the multigrid method is the survey article by Stüben and Trottenberg [1], while the pioneering article by Brandt [1] has already become classical.

The concept underlying the multigrid method is to eliminate efficiently each Fourier component of the error spectrum on the coarsest possible grid. This concept relies on the use of relaxation algorithms that are very efficient in damping those components of the error whose wavelength, in at least one of the coordinate-directions, is comparable to the mesh size.

For nonlinear equations, the nonlinear multigrid method (NMG or FAS: Full Approximation Storage (Brandt [1])) has been used by many investigators, e.g. South and Brandt [1], Jameson [2], McCarthy and Reyhner [1], Raj [1], Caughey [1], and is now widely accepted. A brief outline of the NMG method follows below.

Consider the discretized nonlinear boundary value problem

$$L^N \varphi^N = F^N \tag{5.2.1}$$

on the finest grid G^N of a sequence of grids G^K, $K = N, N-1, \ldots, 2, 1$, of increasing mesh size in the computational domain. Here, G^{K-1} is constructed from G^K by leaving out every other grid point in each coordinate direction, hence the mesh size in any coordinate-direction satisfies $h^{K-1} = 2h^K$.

Since NMG is a recursive process, it is sufficient to explain the relationship between the problems that must be solved on the grids G^K and G^{K-1}. Suppose that the G^K-problem is (with $\hat{F}^N = F^N$)

$$L^K \varphi^K = \hat{F}^K \tag{5.2.2}$$

where φ^K is an as yet unknown approximation of φ^N on G^K and L^K is a discretization similar to L^N of the differential equation to be solved on grid G^K. Note that φ^K can only contain Fourier components for which the wavelength $\lambda^K \geqq 2h^K$ in each coordinate direction. Further, let ϕ^K be a given approximation of φ^K. Then the desired correction $\psi^K = \varphi^K - \phi^K$ can be solved from the correction equation

$$L^K(\phi^K + \psi^K) - L^K \phi^K = R^K, \tag{5.2.3}$$

where the residual R^K is defined by

$$R^K = \hat{F}^K - L^K \phi^K. \tag{5.2.4}$$

As it is more efficient to approximate the smooth part of ψ^K (Fourier component wavelength $\lambda^K \geq 2h^{K-1}$ in all coordinate-directions), as much as possible on the coarser grid G^{K-1} rather than G^K, it is worthwhile to approximate equation (5.2.3) on the grid G^{K-1}. In NMG this approximation is

$$L^{K-1}(I_K^{K-1}\phi^K + \psi^{K-1}) - L^{K-1}(I_K^{K-1}\phi^K) = W_K^{K-1}R^K. \tag{5.2.5}$$

Here I_K^{K-1} and W_K^{K-1} are restriction operators (not necessarily the same) that map gridfunctions on G^K into gridfunctions on G^{K-1}. Equation (5.2.5) is rewritten as

$$L^{K-1}\,\varphi^{K-1} = \hat{F}^{K-1}, \tag{5.2.6}$$

where

$$\varphi^{K-1} = I_K^{K-1}\phi^K + \psi^{K-1} \tag{5.2.7}$$

and

$$\hat{F}^{K-1} = W_K^{K-1}R^K + L^{K-1}\,(I_K^{K-1}\phi^K). \tag{5.2.8}$$

Equation (5.2.6) has the same structure as equation (5.2.2), which allows a recursive description of multigrid methods.
Equation (5.2.6) is approximately solved, with result ϕ^{K-1}. The resulting obvious approximation of ψ^{K-1} is

$$\Psi_{\text{appr.}}^{K-1} = \phi^{K-1} - I_K^{K-1}\phi^K, \tag{5.2.9}$$

see equation (5.2.7).
It follows that an improved approximation ϕ^K of φ^K is obtained by

$$\phi^K := \phi^K + I_{K-1}^K\,\Psi_{\text{appr.}}^{K-1} = \phi^K + I_{K-1}^K(\phi^{K-1} - I_K^{K-1}\phi^K), \tag{5.2.10}$$

where I^{K}_{K-1} is a prolongation operator that maps gridfunctions on G^{K-1} into gridfunctions on G^{K}.

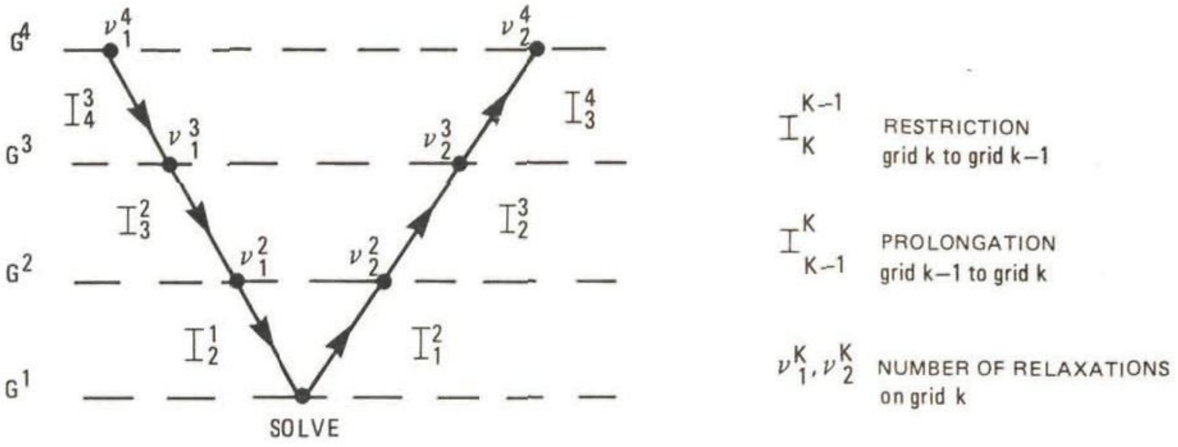
This way, mainly the non-smooth part of φ^{K} (Fourier component wavelengths between $2h^{K-1}$ and $2h^{K}$ in at least one coordinate direction) need to be determined from the G^{K}-problem, equation (5.2.2) or (5.2.3). As the relaxation algorithm was required to be very efficient in damping error components in precisely that part of the spectrum, the non-smooth part of φ^{K} can be determined by doing only a few relaxation sweeps.

The two-grid NMG method that has just been explained is described concisely by the following algorithm in a quasi-programming language (cf. Hackbusch [1], p. 187).

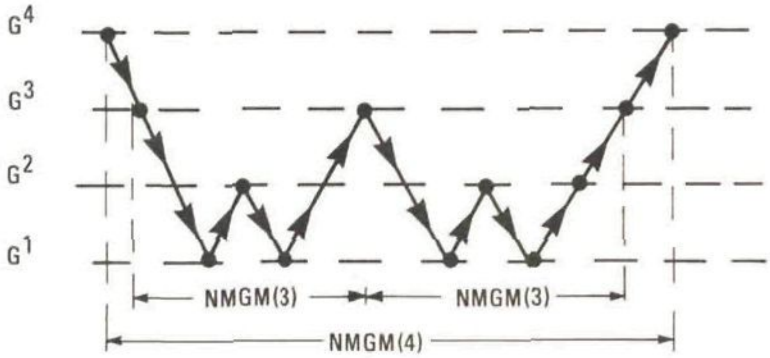
```
procedure NMGM $(K,\phi^{K},F^{K})$
integer K
array $\phi^{K},F^{K}$
begin array $\phi^{K-1}$, $\hat{F}^{K-1}$
    if K = 1 then
       solve $L^{K}\phi^{K} = F^{K}$
    else
         for i := 1(1)$\nu_1$ do $\phi^{K} := S_1(K,\phi^{K},F^{K})$ od

         $\phi^{K-1} := I^{K-1}_{K}\phi^{K}$

         $\hat{F}^{K-1} := W^{K-1}_{K}(F^{K} - L^{K}\phi^{K}) + L^{K-1}\phi^{K-1}$

         for i := 1(1)$\gamma$ do NMGM$(K-1,\phi^{K-1},\hat{F}^{K-1})$ od

         $\phi^{K} := \phi^{K} + I^{K}_{K-1}(\phi^{K-1} - I^{K-1}_{K}\phi^{K})$

         for i := 1(1)$\nu_2$ do $\phi^{K} := S_2(K,\phi^{K},F^{K})$ od
    endif
end
```

Algorithm 5.2.1: The NMG method description

Here S_1 and S_2 stand for relaxation processes. The order in which the grids G^{K},K=1,..,N, are visited depends on the value of γ. With $\gamma = 1$ one obtains the so-called V-cycle, while $\gamma = 2$ results in W-cycles, see figure 5.2.1.


G4
G3
G2
G1
ν41
ν31
ν21
ν42
ν32
ν22
I34
I23
I12
I43
I32
I21
SOLVE
IK-1K RESTRICTION grid k to grid k−1
IKK-1 PROLONGATION grid k−1 to grid k
νK1, νK2 NUMBER OF RELAXATIONS on grid k

a) Example of a V–cycle on 4 grids



b) Example of a W–cycle on 4 grids

Fig. 5.2.1 Examples of V- and W-cycles on 4 grids

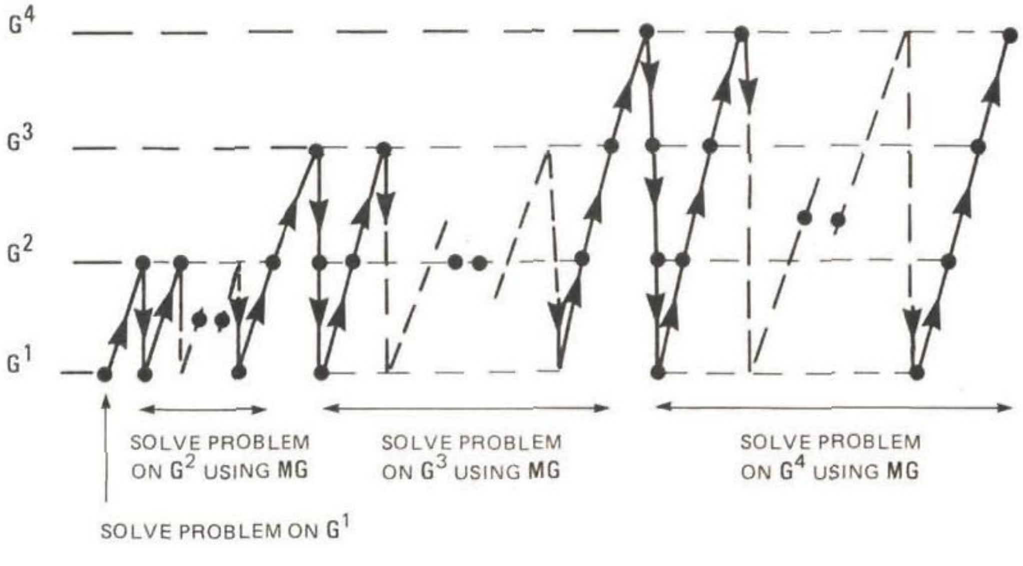

G4
G3
G2
G1
SOLVE PROBLEM ON G2 USING MG
SOLVE PROBLEM ON G3 USING MG
SOLVE PROBLEM ON G4 USING MG
SOLVE PROBLEM ON G1

Fig. 5.2.2 Schematic of nested iteration

Nested iteration (or FMG, Full MultiGrid, Brandt [1]) is applied to obtain a well converged starting solution on each consecutive level of the multigrid process, see figure 5.2.2.

The prolongation and restriction operator are chosen as follows. Trilinear as well as tricubic interpolation is used for the prolongation operator. Weighted restriction is chosen for the residuals, using the weighted operator shown in figure 5.2.3, which assigns a combination of fine grid point values to the coarse grid point in the middle of the box. Weighted restriction or injection, i.e. $\varphi^{K-1} = \varphi^{K}$ for grid points that are both present on G^K and G^{K-1}, is used for the potential restriction. The best choice of operators is problem-dependent and we therefore refer to chapter 8 for the choices actually made.

5.2.2 Restriction of Neumann type boundary conditions

In the multigrid method the restriction of Neumann type boundary conditions requires careful attention. As a consequence of the dummy grid point approach used, the Neumann boundary condition will not be satisfied exactly before restriction. This residual in the boundary condition is interpreted as a normal velocity through the boundary and will be treated as follows. On a ξ^i = constant surface on grid G^K the normal velocity is defined as

$$\hat{g}_{i,j,k} = (U^j N_j)_{i,j,k} = (U^j N_j)^{+}_{i,j,k} + (U^j N_j)^{-}_{i,j,k}, \qquad (5.2.11)$$

where the plus- and minus-sign refer to the upper and lower half of the ξ^i-boundary (figure 4.5.1).

On a ξ^i = constant surface the covariant tensor $N^{(i)}_j$, j = 1,2,3, is obtained as follows. In Cartesian coordinates the unit normal to a surface $\xi^i(x^1,x^2,x^3)$ = constant is given by the normalized gradient on that surface:

$$n^{(i)}_{\alpha} = \frac{\partial \xi^i}{\partial x^\alpha} \Big/ \sqrt{\frac{\partial \xi^i}{\partial x^\beta}\,\frac{\partial \xi^i}{\partial x^\gamma}\,\delta^{\beta\gamma}} = \frac{\partial \xi^i}{\partial x^\alpha}/\sqrt{g^{ii}} \qquad (5.2.12)$$

Using equation (2.3.6) we find

$$N^{(i)}_{j} = \frac{\partial x^\alpha}{\partial \xi^j}\, n^{(i)}_{\alpha} = \delta^i_j/\sqrt{g^{ii}}. \qquad (5.2.13)$$

Applying the mirroring rules from section (4.5) the following expression can be derived from equation (5.2.11),

$$\hat{g}^{K}_{i,j,k} = \left(\sqrt{g^{ii}}\,\frac{\partial\varphi}{\partial\xi^{i}}\right)_{i,j,k} = \left(\sqrt{g^{ii}}\right)_{i,j,k}\left(\tilde{\delta}_{i}\varphi_{i,j,k}\right) \text{ on grid } G^{K} \qquad (5.2.14)$$

This residual is restricted to coarse grids using a restriction operator $\bar{W}^{K-1}_{K}$, that is a two-dimensional version of W^{K-1}_{K}, see figure 5.2.3. This leads to a normal velocity

$$\hat{g}^{K-1} = \bar{W}^{K-1}_{K}\hat{g}^{K} \qquad (5.2.15)$$

on a coarse grid G^{K-1} which is used to prescribe the dummy value on surface ξ^{i} = constant on grid G^{K-1} as follows, (compare equation (5.2.14))

$$\left(\sqrt{g^{ii}}\right)_{i,j,k}\left(\tilde{\delta}_{i}\varphi_{i,j,k}\right) = \hat{g}^{K-1}_{i,j,k} \text{ on grid } G^{K-1}. \qquad (5.2.16)$$

This way, dummy grid point potential values have been defined on grid G^{K-1} by using the given dummy grid point potential values on grid G^{K}, see also figure 5.2.4.

After carrying out the above procedure the term $L^{K-1}I^{K-1}_{K}\phi^{K}$ can be computed to obtain $\hat{F}^{K-1}$ in equation (5.2.8). Subsequently relaxations can be performed on grid G^{K-1}, defining dummy grid point potential values as described in section 4.5.

5.2.3 Convergence and efficiency of the multigrid method

Recently Hackbusch [2] has given a useful unifying description of multigrid convergence theory as known at present. Multigrid convergence theory currently mainly applies to linear (or weakly nonlinear) fully elliptic problems. The basic assumptions in the convergence proofs are:

- a "smoothing property" for the smoothing algorithm, i.e. the algorithm must "smooth" the error on each grid level efficiently,

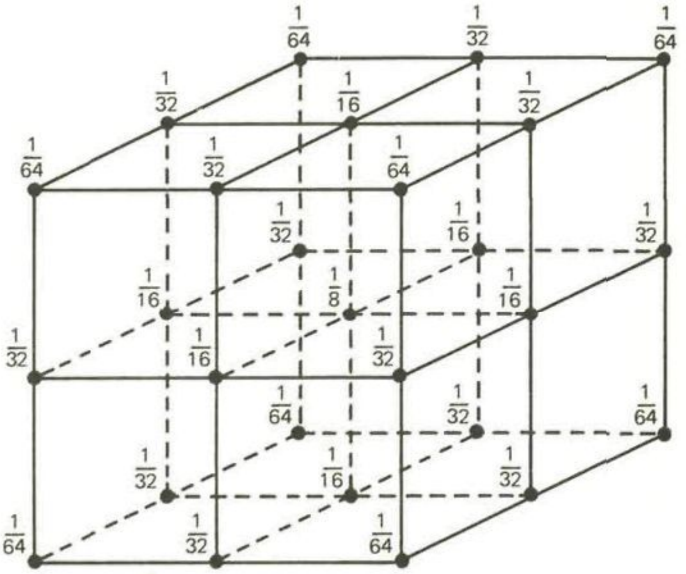


Fig. 5.2.3 Schematic of weighted restriction operator

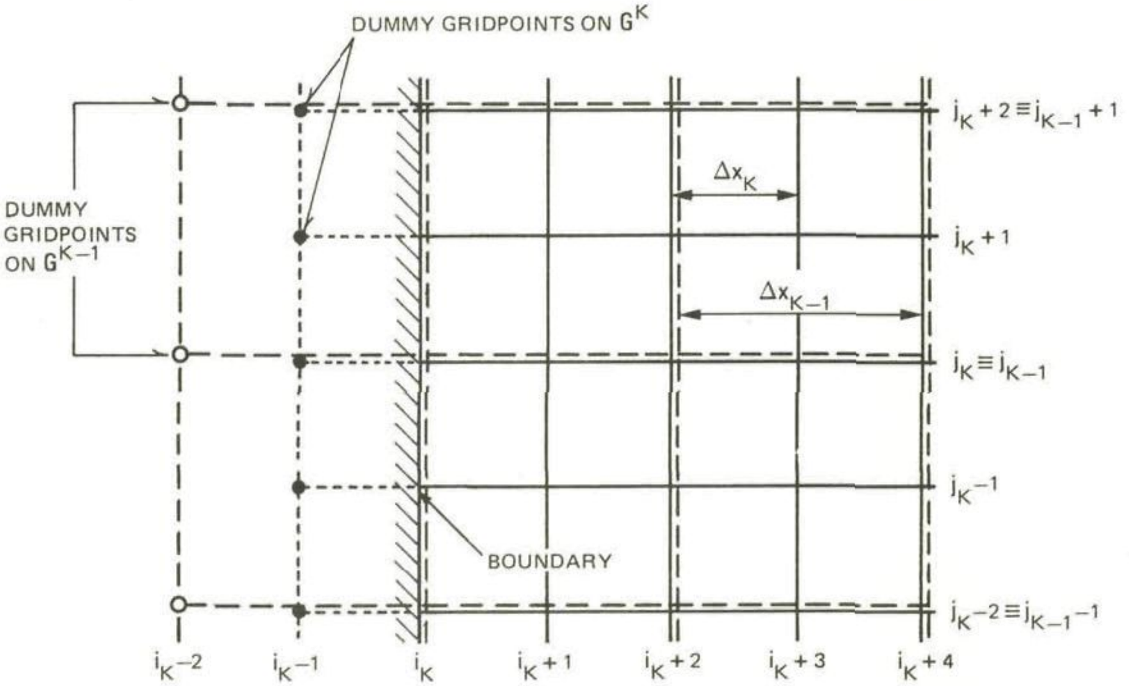


Fig. 5.2.4 Example of dummy gridpoints on G^K and G^{K-1} for a two-dimensional situation

- an "approximation property" for the coarse grid operator, i.e. the coarse grid operator L^{K-1} must "approximate" the fine grid operator L^K well enough.

Under the above assumptions a two-level multigrid convergence proof can be given. Then the step to the general multi-level multigrid convergence proof is small for the W-cycle ($\gamma \geqq 2$ in algorithm 5.2.1), as it only requires the additional assumptions of boundedness of the smoothing operator and Lipschitz-continuity of the prolongation operator. The general multi-level multigrid convergence proof for the V-cycle ($\gamma = 1$) has as yet only been given for symmetric matrices.

At this moment multigrid convergence for nonlinear problems can be proven if the problem is sufficiently linear in the neighbourhood of its solution.

The amount of work required for multigrid convergence (= efficiency) can be estimated as follows. In most flow solver codes the flow residual calculation forms the major part of the computational work and therefore only the work involved in ν flow relaxations (νC_s) and in the residual restriction (C_s) is counted here. The total work involved in one multigrid iteration is therefore approximately

$$W = C_s\ (1 + \nu)\ (n^{(N)} + \gamma n^{(N-1)} + \gamma^2 n^{(N-2)} + \ldots\ldots + \gamma^{N-1} n^{(1)});$$

$$n^{(K)} = n^{(N)}/8^{N-K} \qquad (5.2.17)$$

where $n^{(K)}$ is the number of grid points on grid G^K.
Hence,

$$W \leq C_s\ (1 + \nu)\ n^{(N)}(1 + \frac{\gamma}{8} + \frac{\gamma^2}{64} + \ldots\ldots\ldots)$$

$$= C_s\ (1 + \nu)\ n^{(N)}.\ \frac{1}{1 - \gamma/8} \text{ if } \gamma < 8. \qquad (5.2.18)$$

Therefore the multigrid method requires in an order $n^{(N)}$ amount of work if the contraction number per multigrid iteration ζ_s is bounded away from 1, uniformly in $n^{(N)}$. This requirement is usually satisfied due to the smoothing property of the smoothing relaxation algorithm and a proper choice of multigrid restriction, coarse grid and prolongation operators.

The efficiency of a multigrid method (which is proportional to the total amount of work required for convergence) to a specified accuracy is now defined as

$$E = -W \; / \; {}^{10}\log \zeta_s \tag{5.2.19}$$

5.3 Linearization and construction of system matrix

In preparation for the description of the smoothing method to be employed in the multigrid method we here describe the linearization of the operators $L = L^K$ and the construction of the system matrix.

The nonlinear equation $L\varphi = f$, described in section 2.3, is linearized on each grid in a straightforward manner by putting

$$\varphi = \phi + \Delta\varphi \quad , \qquad \phi \text{ given,} \tag{5.3.1}$$

and subsequently deleting all terms of order $(\Delta\varphi)^2$ and smaller. This leads to the equation

$$\tilde{L}[\phi]\Delta\varphi = f - L\phi. \tag{5.3.2}$$

The righthandside is the residual for $\varphi = \phi$ of the flow equation (2.3.8). The lefthandside is the Fréchet-derivative for $\varphi = \phi$, which has already been derived in section 2.4. The expression for $\tilde{L}[\phi]$ follows from equation (2.4.8),

$$\tilde{L}[\phi] \; \Delta\varphi = (g^{ij} - \frac{\bar{U}^i\bar{U}^j}{\bar{a}^2}) \; \frac{\partial}{\partial\xi^i} \, \frac{\partial}{\partial\xi^j} \, \Delta\varphi. \tag{5.3.3}$$

The system matrix $\tilde{L}$ and the relaxation schemes that apply to $\tilde{L}$ have a complex nonzero pattern involving the grid points (i+p,j+q,k+r) where p, q and r range from -1 to 1 in the elliptic part of the flow and can be -2 or 2 in the hyperbolic part of the flow. Hence, in general there are 125 points in the stencil of the FV discretization, 64 of which may be nonzero.

A sparser scheme can be obtained by subtracting terms of order $\Delta\xi^i\Delta\xi^j$ from the lefthand side of equation (5.3.3) by using the approximation

$$(\Delta\varphi)_{i+p,j+q,k+r} \approx (\Delta\varphi)_{i+p,j,k} + (\Delta\varphi)_{i,j+q,k} + (\Delta\varphi)_{i,j,k+r} - 2(\Delta\varphi)_{i,j,k}. \qquad (5.3.4)$$

In this way an approximation to $\tilde{L}$ is obtained which involves terms of the type $(\Delta\varphi)_{i+p,j+q,k+r}$, where p, q and r still range from -2 to 2, but this time with the restriction that only one of them can be nonzero at the same time. Schemes with such a sparsity pattern are easier amenable to application of the ILU/SIP relaxation scheme. The price paid is the deletion of crossderivative terms in $\Delta\varphi$ from the iteration scheme. These terms are always present due to the linearization of the density and in many cases also due to the skewness of the grid. Experience has shown, however, that the convergence properties of the iteration scheme are not affected much by this deletion as long as the skewness of the grid is moderate (see also chapter 6). A still sparser approximation to $\tilde{L}$ is obtained by setting (e.g. in the ξ^1-direction)

$$\Delta\varphi_{i+2,j,k} - 3\,\Delta\varphi_{i+1,j,k} + 3\,\Delta\varphi_{i,j,k} - \Delta\varphi_{i-1,j,k} \approx$$

$$\Delta\varphi_{i+1,j,k} - 3\,\Delta\varphi_{i,j,k} + 3\,\Delta\varphi_{i-1,j,k} - \Delta\varphi_{i-2,j,k}, \qquad (5.3.5)$$

which can be used to eliminate the entry $\Delta\varphi_{i+2,j,k}$ or $\Delta\varphi_{i-2,j,k}$ (nonzero in non-subsonic parts of the flow) from the iteration scheme. It is simple to show that equation (5.3.5) corresponds with reversing the upwind bias, and consequently also reversing the sign of the numerical viscosity, in the iteration scheme for non-subsonic parts of the flow, compare subsection 4.4.4. As a result of applying equation (5.3.5), the storage required for the ILU/SIP relaxation scheme will be diminished significantly, while practice has shown that the convergence properties of the iteration scheme are not affected.

5.4 Smoothing algorithm

5.4.1 Description of ILU/SIP algorithm

For reasons of robustness and efficiency an algorithm named ILU/SIP has been used as the smoothing algorithm in the multigrid method. Its properties will be compared against the properties of the more usual SLR

(Successive Line Relaxation) algorithm in the next chapter.

An extensive description of ILU and SIP can be found in Meijerink and Van der Vorst [1] and in Stone [1] respectively. Here, only a brief description will be given.

The equation to be solved follows from the previous section and is written as

$$\tilde{L}[\phi]\Delta\varphi = f - L\phi. \tag{5.4.1}$$

In the grid point (i,j,k) the expression $\tilde{L}[\phi]\Delta\varphi$ can be written as

$$\{\tilde{L}[\phi]\Delta\varphi\}_{i,j,k} = \sum_{p,q,r} a^{pqr} (\Delta\varphi)_{i+p,j+q,k+r}, \tag{5.4.2}$$

$$p,q,r \in \{-2, -1, 0, 1\}, \tag{5.4.3}$$

with only one index p,q or r nonzero at the same time.

An iteration scheme to solve equation (5.4.1) can be described as

$$L^*[\phi^n]\Delta\varphi^n = f - L\phi^n, \tag{5.4.4}$$

$$\phi^{n+1} = \phi^n + \Delta\varphi^n, \tag{5.4.5}$$

where the iteration matrix L^* is chosen such that it is easily invertible and preferably also is a good approximation of the system matrix $\tilde{L}$. The error matrix B is defined by

$$L^* = \tilde{L} + B. \tag{5.4.6}$$

This results in the modified equation

$$\begin{aligned} B[\phi^n]\Delta t\, \overleftarrow{\partial}_t \phi^{n+1} &= f - \tilde{L}[\phi^n]\phi^{n+1} + \tilde{L}[\phi^n]\phi^n - L\phi^n \\ &= g[\phi^n] - \tilde{L}[\phi^n]\phi^{n+1} \end{aligned} \tag{5.4.7}$$

with

$$\Delta t \; \overleftarrow{\partial}_t \phi^{n+1} = \Delta\varphi^n = \phi^{n+1} - \phi^n. \tag{5.4.8}$$

With both ILU and SIP, an incomplete lower/upper (Gauss) decomposition of the systemmatrix $\tilde{L}$ is carried out. For each algorithm, this decomposition is performed using a prespecified nonzero pattern (sparsity pattern), here coinciding with the pattern of $\tilde{L}$. In carrying out the Gauss matrix decomposition process for the lower triangular part of $\tilde{L}$, nonzero entries will be generated outside this pattern. The treatment of these nonzero entries determines the form of the error matrix B that will be obtained. This treatment differs for ILU and SIP.

In the case of ILU, the nonzero entries mentioned before are simply deleted (so that they need not even be computed). It is found that $\{B\Delta\varphi\}_{i,j,k}$ has the form

$$\{B\Delta\varphi\}_{i,j,k} = \sum_{p,q,r} b^{pqr}(\Delta\varphi)_{i+p,j+q,k+r},$$

$$p,q,r \in \{-2,-1,0,1\},\; p = 0 \text{ or } q = 0 \text{ or } r = 0, \tag{5.4.9}$$

with two indices p, q or r nonzero at the same time. The modified equation has the form

$$\left(c + d\Delta\xi^i \frac{\partial}{\partial\xi^i} + e\Delta\xi^i\Delta\xi^j \frac{\partial}{\partial\xi^i}\frac{\partial}{\partial\xi^j} + \ldots\right) \Delta t \; \overleftarrow{\partial}_t \phi^{n+1} =$$

$$g[\phi^n] - \tilde{L}[\phi^n]\phi^{n+1}. \tag{5.4.10}$$

In case of ILU with lumping the nonzero entries mentioned before are computed explicitly and added to the main diagonal of the system matrix, It can be shown, however, that this does not improve the smoothing algorithm.

In case of SIP, all $c\Delta t\overleftarrow{\partial}_t\phi^{n+1}$- and $d\;\Delta\xi^i \frac{\partial}{\partial\xi^i}\,\Delta t\overleftarrow{\partial}_t\phi^{n+1}$- terms are annihilated in the modified equation (5.4.10) by making use of the approximation (compare equation (5.3.4))

$$(\Delta\varphi)_{i+p,j+q,k+r} \approx (\Delta\varphi)_{i+p,j,k} + (\Delta\varphi)_{i,j+q,k} + (\Delta\varphi)_{i,j,k+r} - 2(\Delta\varphi)_{i,j,k}. \qquad (5.4.11)$$

as follows. Each of the nonzero entries mentioned before is added to the off-diagonal entries a^{p00}, a^{0q0}, a^{00r} and twice subtracted from the main diagonal entry a^{000} of $\tilde{L}$. All these entries are present in the sparsity pattern of $\tilde{L}$. The term $B\Delta\varphi^n$ now takes the form

$$\{B\Delta\varphi^n\}_{i,j,k} = \sum_{p,q,r} b^{pqr}\,\{(\Delta\varphi^n)_{i+p,j+q,k+r} - (\Delta\varphi^n)_{i+p,j,k} - (\Delta\varphi^n)_{i,j+q,k} - (\Delta\varphi^n)_{i,j,k+r} + 2(\Delta\varphi^n)_{i,j,k}\}, \qquad (5.4.12)$$

$p,q,r \in \{-2,-1,0,1\}$, $p = 0$ or $q = 0$ or $r = 0$.

with two indices p, q or r nonzero at the same time.
Consequently the modified equation of SIP has the form (compare equation (5.4.10)):

$$\left(e\,\Delta\xi^i\Delta\xi^j \frac{\partial}{\partial\xi^i}\frac{\partial}{\partial\xi^j} + \ldots\right) \Delta t\,\overset{\leftarrow}{\partial}_t \phi^{n+1} = g[\phi^n] - \tilde{L}[\phi^n]\,\phi^{n+1}. \qquad (5.4.13)$$

Usually, only a fraction α, $0 \leqq \alpha \leqq 1$ of $(\Delta\varphi^n)_{i+p,j+q,k+r}$ is approximated using equation (5.4.11). This will be denoted as SIP(α); hence, SIP(0) is identical to ILU. Therefore, whereas SIP is described here as a modification of classical ILU, ILU can also be looked upon as a special version of SIP. The family of algorithms described here will consequently be referred to as ILU/SIP(α). Usually, SIP is not described within the framework of ILU, but by a set of recursive formulas (Sankar [1], Stone [1], Zedan and Schneider [1]). The explicit derivation of these formulas is cumbersome and is here formulated implicitly by implementing SIP as a modified form of classical ILU.

ILU/SIP algorithm

The ILU/SIP is described in detail in appendix B. We therefore restrict ourselves to a global description here.

For simplicity, the principle of the construction of ILU decompositions will be explained for the standard 3D 7-point discretization of the Laplacian. The computational points will be ordered lexicographically, i.e. a point (i,j,k) in a computational box [-NI : NI, 0 : NJ, 0 : NK] is given the sequence number N = i + NI+1 + (NI+NI+1) * (j + k*(NJ+1)). The resulting system matrix $\tilde{L}$ for the 3D Laplace operator is sketched in figure 5.4.1. The figure shows that the sparsity pattern of the system matrix consists of j-lines (j,k constant) which together consitute k-planes (k constant).

In ILU decompositions two steps can be distinguished:

1. Decomposition of the system matrix into a lower and an upper triangular matrix. This is done by "sweeping" the diagonals of the lower triangular part of the system matrix. Here "sweeping" means making matrix elements zero in the standard way as in a standard Gauss decomposition while operating on fill-in as described above (equation (5.4.11)). The matrix is now upper triangular.
2. Solving for $\Delta\varphi$ by backsubstitution, as in standard Gauss decomposition.

The sparsity pattern of the matrices $\tilde{L}$ and B in case of the ILU/SIP algorithm is sketched in figure 5.4.2.

5.4.2 Properties and applications of ILU/SIP

An important property of the ILU/SIP algorithm is the absence of a preferred sweep-direction. With successive line relaxation (SLR) the sweep-direction strongly influences the smoothing properties of the algorithm, and in many cases the nature of the problem is such that it is is necessary to perform line relaxations in all three coordinate directions, in order to obtain a good convergence. The absence of a preferred sweep-direction makes ILU/SIP a suitable smoothing algorithm for use within the multigrid method, especially in those cases where the flow-direction varies strongly (e.g. in air-intakes) and where mesh aspect ratios vary widely (as in the problem considered in this thesis). The price paid is a slightly higher computation time per relaxation, compared to SLR, and the need to store the entire (sparse) upper triangular matrix U. (The algorithm is coded as a plane-by-plane algorithm, that is only a 2D plane of the system matrix (and possibly the lower matrix) is stored.) The storage required is acceptable, however, and comparable to the storage required for the finite volume scheme (see chapter 7).

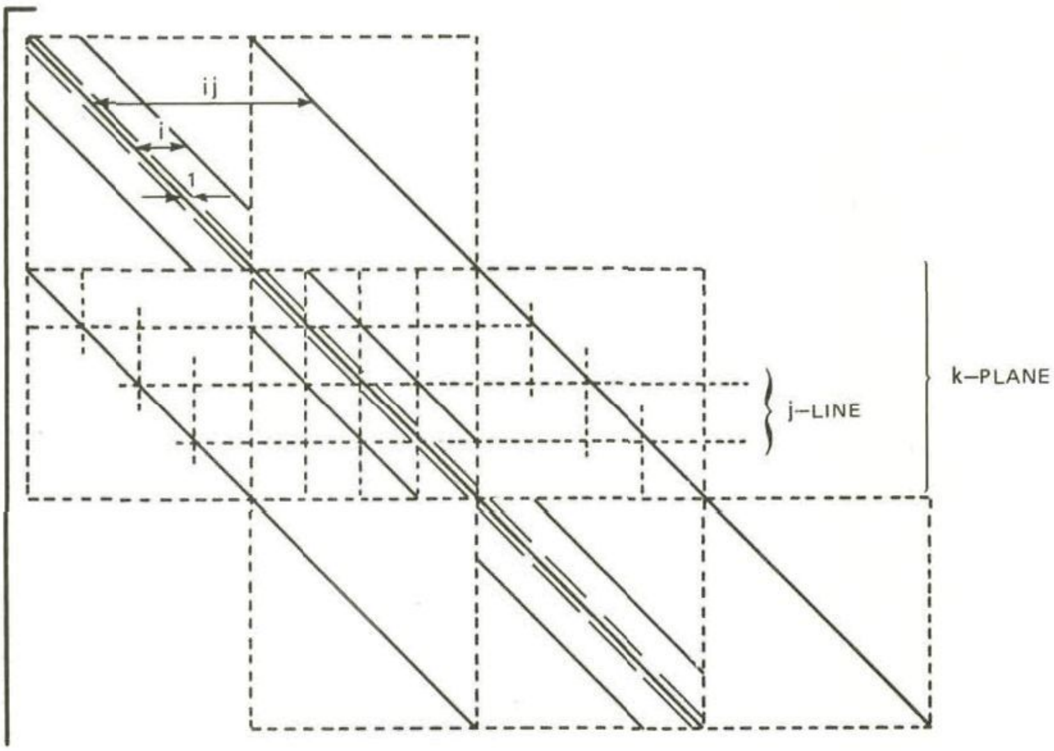


Fig. 5.4.1 Upper left corner of the sparsity pattern of the system matrix $\tilde{L}$ for the 7-point discretization of the Laplace-operator on an i·j·k-grid

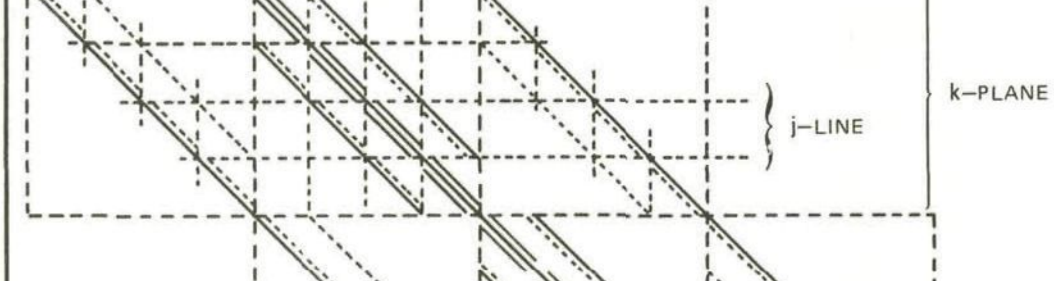


Fig. 5.4.2 Upper left corner of the sparsity patterns of the system matrix $\tilde{L}$ (drawn lines) and the error matrix B (drawn and dotted lines) for an ILU/SIP-decomposition of the 7-point discretization of the Laplace-operator on an i·j·k-grid

The alternative is, of course, to use incomplete LU decomposition per plane, requiring storage of one 2D plane of ILU factors only. Such an algorithm is in fact a three-dimensional generalization of the two-dimensional SLR algorithm; this alternative has been analyzed in Van der Wees, Van der Vooren and Meelker [1], Van der Wees [1].

A further useful property of ILU/SIP is its flexibility with respect to the choice of the sparsity pattern for the lower and upper triangular matrices L and U. For instance, less sparse difference-molecules than the 7-point Laplace-discretization can be fully accommodated within the ILU/SIP algorithm by a simple extension of the sparsity pattern. Extension of the sparsity pattern of the L and U matrices can also be applied to improve the smoothing properties and robustness of the ILU/SIP algorithm in the multigrid method. The price to be paid is, of course, that a less sparse upper matrix U has to be stored. In Van der Wees, Van der Vooren and Meelker [1], Van der Wees [1] an experiment on a relatively simple configuration shows that extension of the sparsity pattern used in ILU/SIP indeed improves its performance. In realistic flow cases, however, the total computation then becomes I/O-bound on the NLR Cyber 180-855. The extra I/O time required for the transfer of the denser upper triangular matrix U makes the use of extended matrix patterns in ILU/SIP unattractive on that machine (see also chapter 7).

Multiple applications (Sonneveld, Wesseling and de Zeeuw [1], Hemker [1], Wesseling [2], Kettler [1,2], Wesseling and Sonneveld [1]) have shown that use of ILU smoothing in the multigrid method has led to a very fast and robust tool for the solution of a wide variety of 2D elliptic problems. An application of using ILU within the multigrid method for the solution of 2D transonic flow is given by Brédif [2], Nowak and Wesseling [1] and using SIP with multigrid by Sanker [1]. The applicability of MG-ILU/SIP to 3D subsonic and transonic potential flow has been shown in Van der Wees, Van der Vooren and Meelker [1], Van der Wees [1] for a simple non-lifting wing embedded in a rectangular grid. The capabilities of MG-ILU/SIP for transonic potential flows around realistic wings embedded in curvilinear grids have been demonstrated in Van der Wees [2].

6 STABILITY AND SMOOTHING ANALYSIS OF ILU/SIP ALGORITHM

6.1 Smoothing analysis for elliptic problems

6.1.1 General description

An effective tool to investigate locally the damping of the various (high-frequency) components of the error is the local mode (Fourier) analysis, introduced by Brandt [1]. The local mode analysis will be applied here to elliptic problems, while the relaxation of hyperbolic problems, basically a time-marching process will be analyzed in the next section using a modified equation technique. In both cases the assumption is made that the coefficients of the system matrix are the same for all grid points. It is also assumed that boundary conditions are periodic or that the number of grid points is infinite in all three coordinate directions. Consequently, both analysis methods are only locally valid in smooth parts of the flow outside shocks and sonic surfaces. Comparison with practical computations may show discrepancies, because boundary conditions may influence the smoothing behaviour.
The local mode analysis consists of two steps. First the equations

$$\left.\begin{aligned} (L\varphi)_{i,j,k} &= f_{i,j,k} \\ e^{n}_{i,j,k} &= \phi^{n}_{i,j,k} - \varphi_{i,j,k} \\ \Delta\varphi^{n}_{i,j,k} &= \phi^{n+1}_{i,j,k} - \phi^{n}_{i,j,k} = e^{n+1}_{i,j,k} - e^{n}_{i,j,k} \end{aligned}\right\} \qquad (6.1.1)$$

for grid point (i,j,k) are combined with equations (5.4.4) and (5.4.6) to give

$$(L+B)(e^{n+1}-e^{n})_{i,j,k} = -Le^{n}_{i,j,k}. \qquad (6.1.2)$$

Next, the Fourier component $e^{n}_{ijk} = (G[\mu,\theta,\omega])^{n} e^{\hat{\imath}(i\mu+j\theta+k\omega)}$ is substituted into this expression, leading to the reduction-factor

$$\rho[\mu,\theta,\omega] = \left|G[\mu,\theta,\omega]\right|, \qquad (6.1.3)$$

which can be analyzed as a function of the "frequencies" $-\pi \leq \mu,\theta,\omega \leq \pi$. For the use in the multigrid method, in particular the high-frequency modes of the error (at least one "frequency" is high, e.g. $\frac{\pi}{2} < |\mu| \leq \pi$) should be damped efficiently. A good measure for this is the smoothing number $\bar{\rho}$: the maximum value of $\rho(\mu,\theta,\omega)$ over the high-frequency part of the (μ,θ,ω)-domain.
In case $\bar{\rho} < 1$, but $\rho > 1$ for some low-frequency mode of the error, the smoothing algorithm will be considered "divergent". In practice, the multigrid method may work for such a case, because according to multigrid convergence theory the smoothing algorithm may be slowly diverging for low frequency modes if the smoothing algorithm is sufficiently efficient for high-frequency modes of the error.

In this section smoothing numbers $\bar{\rho}$ will frequently be compared using the smoothing efficiency

$$N_{it} = -1 \; / \; {}^{10}\log \bar{\rho}, \qquad (6.1.4)$$

i.e. the number of iterations necessary to reduce the error in the high-frequency modes by a factor 10.

The local mode analysis will be used to analyze the ILU/SIP smoothing properties for the Poisson equation

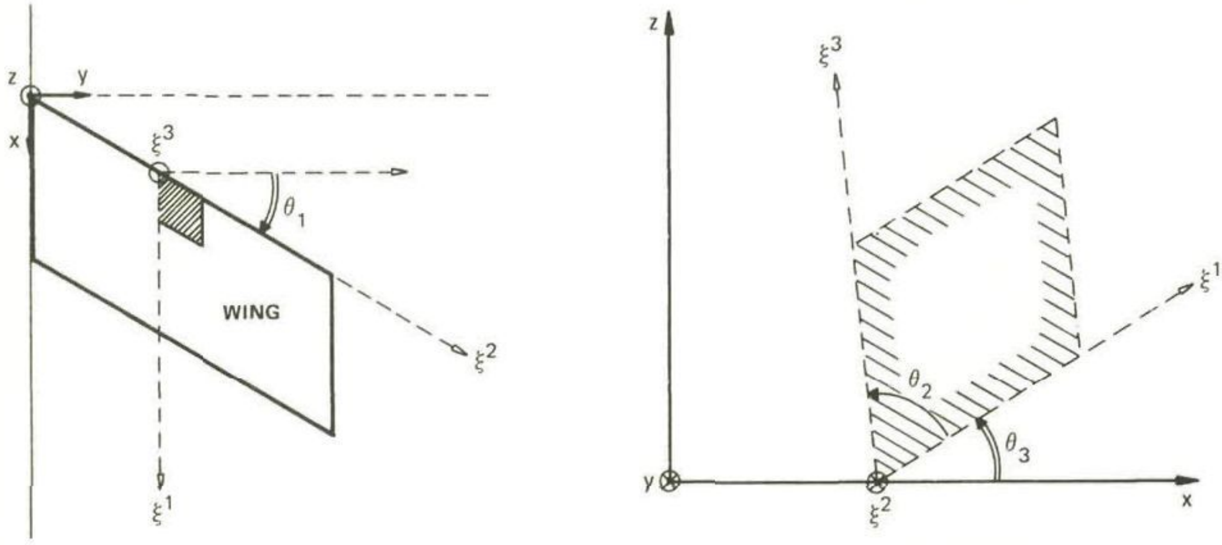
$$\varphi_{xx} + \varphi_{yy} + \varphi_{zz} = f \qquad (6.1.5)$$

discretized using central differences on a uniform skew grid with straight grid lines. Consequently, the following equation is analyzed

$$g^{ij} \frac{\partial}{\partial \xi^i} \frac{\partial}{\partial \xi^j} \varphi = f. \qquad (6.1.6)$$

Equation (6.1.5) will be used to analyze the influence of the following two grid properties on the error smoothing of the ILU/SIP algorithm:

1. *The influence of the two cell aspect ratios* $\Delta\xi^i/\Delta\xi^j$, $i \neq j$, *considering a rectangular grid.* Both cell aspect ratios can vary from very small to very large in a realistic wing grid, see figure 3.1.6.



a) Wing sweep angle θ_1

b) Grid skewness angle θ_2 and grid rotation angle θ_3

Fig. 6.1.1 Definition of the angles θ_1, θ_2 and θ_3

2. The influence of grid skewness. The following parameters will be considered in the analysis:
   - Wing sweep, i.e. the angle θ_1 between the y- and the ξ^2-axis, see figure 6.1.1a.
   - Grid skewness in a wing-normal gridplane. For that purpose, the angle θ_2 between the ξ^1- and ξ^3-axis is introduced, see figure 6.1.1b.
   - Grid rotation, i.e. the angle θ_3 between the ξ^1- and the x^1-axis, see figure 6.1.1b.

The relation between the physical coordinate system (x^1,x^2,x^3) and the computational coordinate system (ξ^1,ξ^2,ξ^3) is described by the Jacobian H (see equation (2.3.2)):

$$H = [\frac{\partial x^i}{\partial \xi^\alpha}] = \begin{bmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} & \frac{\partial x^1}{\partial \xi^3} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} & \frac{\partial x^2}{\partial \xi^3} \\ \frac{\partial x^3}{\partial \xi^1} & \frac{\partial x^3}{\partial \xi^2} & \frac{\partial x^3}{\partial \xi^3} \end{bmatrix} = \begin{bmatrix} \cos\theta_3 & \sin\theta_1 & \cos(\theta_2+\theta_3) \\ 0 & \cos\theta_1 & 0 \\ \sin\theta_3 & 0 & \sin(\theta_2+\theta_3) \end{bmatrix} \qquad (6.1.7)$$

The contravariant metric tensor $G = [H^T H]^{-1}$ is easily derived from this equation.

Discretization of equation (6.1.6) using central difference operators gives a relatively dense system matrix due to crossderivatives in the partial differential equation. If this system matrix were to be used as a basis for the ILU/SIP algorithm, the crossderivative terms would severely affect the cost-effectiveness of the method. Therefore these terms will not be incorporated in the relaxation process, leading to the following relaxation equation for the solution of the desired correction $\Delta\varphi$ for a given ϕ:

$$\left(g^{11} \frac{\partial^2}{\partial\xi^1 \partial\xi^1} + g^{22} \frac{\partial^2}{\partial\xi^2 \partial\xi^2} + g^{33} \frac{\partial^2}{\partial\xi^3 \partial\xi^3}\right) \Delta\varphi^n_{i,j,k} =$$

$$= \omega \left(f_{i,j,k} - g^{ij} \frac{\partial}{\partial\xi^i} \frac{\partial}{\partial\xi^j} \phi^n_{i,j,k}\right) \qquad (6.1.8)$$

where ω is an over/under-relaxation factor which has been introduced for later use. The ILU/SIP algorithm given in chapter 5 will be compared with SLR-X (line relaxation along ξ^1-lines, i.e. lines of constant ξ^2 and ξ^3), SLR-Y and SLR-Z. The use of SLR-X and SLR-Z are relatively common in transonic potential flow calculations. In the SLR algorithm there is no need to drop crossderivative terms from the system equation and consequently this algorithm will be applied directly to equation (6.1.6). The SLR algorithm will be used lexicographically for i,k,j, i.e. for SLR-X with increasing ξ^3 and for each ξ^3 with increasing ξ^2. The same ordering of variables will be used for ILU/SIP.

In appendix B expressions are derived for the computation of the smoothing number $\bar{\rho}$ for the various algorithms considered. Here, only results will be given.

All data-processing and -presentation for the smoothing analysis results presented in this section has been sone using the EDIPAS system (Heerema and Kreijkamp [1]).

6.1.2 Influence of grid aspect ratios on smoothing properties

In figure 6.1.2 the smoothing efficiency N_{it} is plotted for ILU/SIP, considering a rectangular grid with mesh sizes $\Delta\xi^1$, $\Delta\xi^2$ and $\Delta\xi^3$. The values taken for $\Delta\xi^i$ correspond to $\Delta\xi$-values used later in table 6.1.1. Figure 6.1.2 shows N_{it} as a function of the relaxation

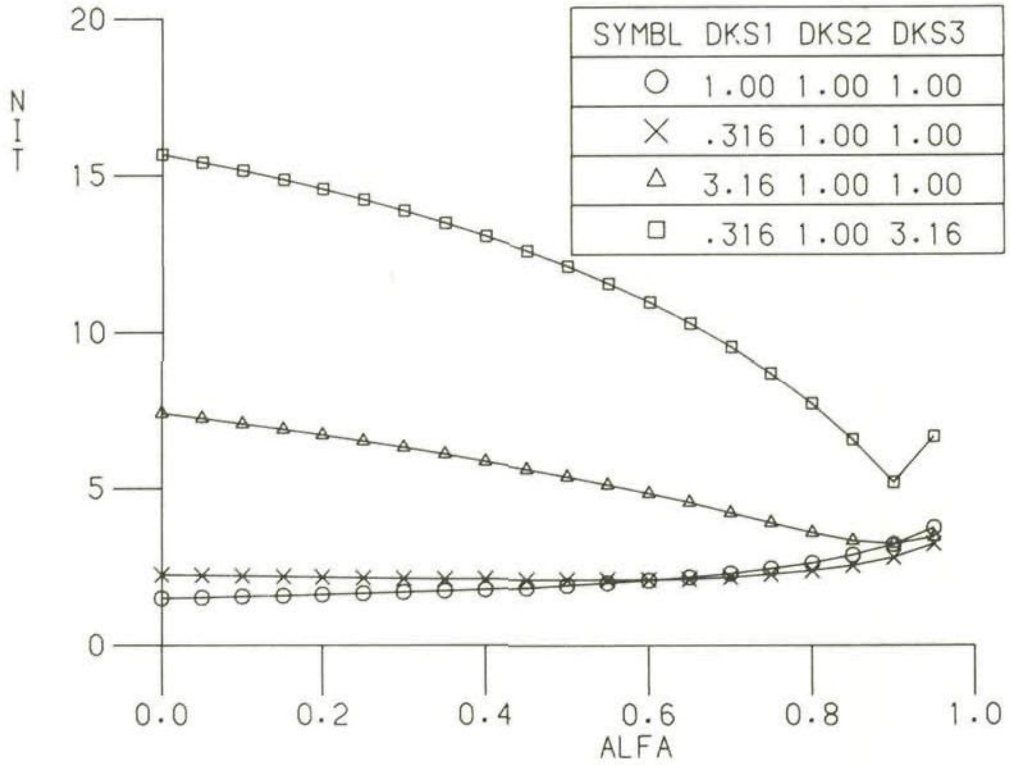


Fig. 6.1.2 Smoothing efficiency of ILU/SIP(α) as a function of the relaxation parameter α for several values of $(\Delta\xi^1,\Delta\xi^2,\Delta\xi^3)$ on a rectangular grid

parameter α in ILU/SIP. The numbers presented have been obtained for the frequency values $\pm \frac{2\pi}{K}$, K = 2, 3, 4, 8, 16, 32, 64, 128, 256 and infinity. These values have been found to be representative for the entire frequency domain.

Figure 6.1.2 shows that the smoothing efficiency of ILU/SIP improves with increasing α in case one mesh size is much larger than the other two mesh sizes, but that the smoothing efficiency of ILU/SIP deteriorates with increasing α in other cases. Because wing grids usually have a large mesh size in the spanwise direction, see figure 3.1.6, α will be taken close to 1. Because in practice the value 1.0 turns out not to be optimal, as will be shown in section 6.2, the ILU/SIP parameter will be set to .70. This parameter will no longer be mentioned explicitly in this chapter. The value α = .70 was also found to be reasonably optimal in the SIP-analysis by Zedan and Schneider [1].

In table 6.1.1 the smoothing number $\bar{\rho}$ is given for a large set of variations of $\Delta\xi^1$, $\Delta\xi^2$, $\Delta\xi^3$ on a rectangular grid. In this table, SLR-X, SLR-Y and SLR-Z indicate successive line relaxation along ξ^1-, ξ^2- and ξ^3-lines respectively.

TABLE 6.1.1

Smoothing numbers $\bar{\rho}$ for various algorithms in case of the elliptic testproblem, equation (6.1.6), on a rectangular grid

ΔX	ΔZ	ΔY	g^{11}	g^{22}	g^{33}	ILU/SIP	SLR-X	SLR-Z	SLR-Y
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	.3681	.5000	.5000	.5000
.3162	1.0000	1.0000	10.0001	1.0000	1.0000	.3493	.4982	.8462	.8462
1.0000	.3162	1.0000	1.0000	10.0001	1.0000	.3493	.8462	.4982	.8462
1.0000	1.0000	.3162	1.0000	1.0000	10.0001	.3493	.8462	.8462	.4982
3.1623	1.0000	1.0000	.1000	1.0000	1.0000	.5815	.9091	.8346	.8346
1.0000	3.1623	1.0000	1.0000	.1000	1.0000	.5815	.8346	.9091	.8346
1.0000	1.0000	3.1623	1.0000	1.0000	.1000	.5815	.8346	.8346	.9091
.5623	1.0000	1.7783	3.1624	1.0000	.3162	.4449	.6397	.8346	.8681
.5623	1.7783	1.0000	3.1624	.3162	1.0000	.4449	.6397	.8681	.8346
1.0000	.5623	1.7783	1.0000	3.1624	.3162	.4449	.8346	.6397	.8681
1.0000	1.7783	.5623	1.0000	.3162	3.1624	.4449	.8346	.8681	.6397
1.7783	.5623	1.0000	.3162	3.1624	1.0000	.4449	.8681	.6397	.8346
1.7783	1.0000	.5623	.3162	1.0000	3.1624	.4449	.8681	.8346	.6397
.1000	1.0000	1.0000	100.0000	1.0000	1.0000	.3870	.4982	.9806	.9806
1.0000	.1000	1.0000	1.0000	100.0000	1.0000	.3870	.9806	.4982	.9806
1.0000	1.0000	.1000	1.0000	1.0000	100.0000	.3870	.9806	.9806	.4982
10.0000	1.0000	1.0000	.0100	1.0000	1.0000	.9349	.9901	.9804	.9804
1.0000	10.0000	1.0000	1.0000	.0100	1.0000	.9349	9804	.9901	.9804
1.0000	1.0000	10.0000	1.0000	1.0000	.0100	.9349	.9804	.9804	.9901
.3162	1.0000	3.1623	10.0001	1.0000	.1000	.7858	.8346	.9804	.9821
.3162	3.1623	1.000	10.0001	.1000	1.0000	.7858	.8346	.9821	.9804
1.0000	.3162	3.1623	1.0000	10.0001	.1000	.7858	.9804	.8346	.9821
1.0000	3.1623	.3162	1.0000	.1000	10.0001	.7858	.9804	.9821	.8346
3.1623	.3162	1.0000	.1000	10.0001	1.0000	.7858	.9821	.8346	.9804
3.1623	1.0000	.3162	.1000	1.0000	10.0001	.7858	.9821	.9804	.8346

It can be seen from figure 3.1.6 that all ratios of $\Delta\xi^1$, $\Delta\xi^2$ and $\Delta\xi^3$ presented in table 6.1.1 are generally present in a qualitative sense in a realistic wing grid. The relative mesh sizes considered in table 6.1.1, i.e. $(\sqrt{10})^k$ and $(10)^k$, $k = -1, 0, 1$, are however moderate when compared to realistic cases. For example near the leading edge of the wing, $\Delta\xi^2$ can easily be over twenty times larger than $\Delta\xi^1$, $\Delta\xi^3$.

It can be concluded from table 6.1.1 that SLR is very sensitive to mesh aspect ratios: even in the moderate cases considered, an almost unacceptable value $\bar{\rho} = .98$ is obtained. ILU/SIP is far less sensitive than SLR. In the case two mesh sizes are given and the third dimension is to be chosen, table 6.1.1 is a useful guideline for making this choice such that a good smoothing efficiency is obtained. Clearly, three different mesh sizes is unfavourable for the smoothing algorithm and should, if possible, be avoided. With the grids used in actual wing calculations, this is possible only by requiring that grids have the property $\Delta\xi^1 \approx \Delta\xi^3$. Smoothing numbers $\bar{\rho}$ for that case are shown in figure 6.1.3. The curves containing large circles refer to multigrid algorithms with selective coarsening, viz. coarsening in $\xi^1\xi^3$-planes only. A close examination of the damping of the high frequency modes reveals that the bad smoothing rate $\bar{\rho}$ for $\Delta\xi^1 << \Delta\xi^2$ is caused by a mode which has a very low frequency in ξ^1- and ξ^3-direction, but a very high frequency in ξ^2-direction. When there is no coarsening applied in the ξ^2-direction, this mode can equally well be smoothed on all coarser grids, and hence a "smaller spectrum" need be smoothed on the fine grid. Thus, the smoothing factor $\bar{\rho}$ improves at the cost of a less efficient coarse grid correction process, as the coarser grids will now contain more grid points.

The following conclusions can be drawn from Figure 6.1.3:

- In general, the convergence deterioration is more severe for $\Delta\xi^1 << \Delta\xi^2$ (occurring near the wing, figure 3.1.6) than for $\Delta\xi^1 >> \Delta\xi^2$ (occurring in the far field, figure 3.1.6).
- In general, ILU/SIP is less sensitive to the aspect ratio $\Delta\xi^1/\Delta\xi^2$, $\Delta\xi^1 = \Delta\xi^3$ than SLR, except when $\Delta\xi^1 >> \Delta\xi^2$, where SLR-Y does a perfect job.
- In case $\Delta\xi^1 << \Delta\xi^2$, mesh halving, i.e. doubling the number of grid points in the ξ^2-direction, can pay off in terms of total computing cost.
- Selective coarsening can lead to a convergence rate $\bar{\rho}$ which is independent of $\Delta\xi^1/\Delta\xi^2 < 1$ if $\Delta\xi^1 = \Delta\xi^3$.

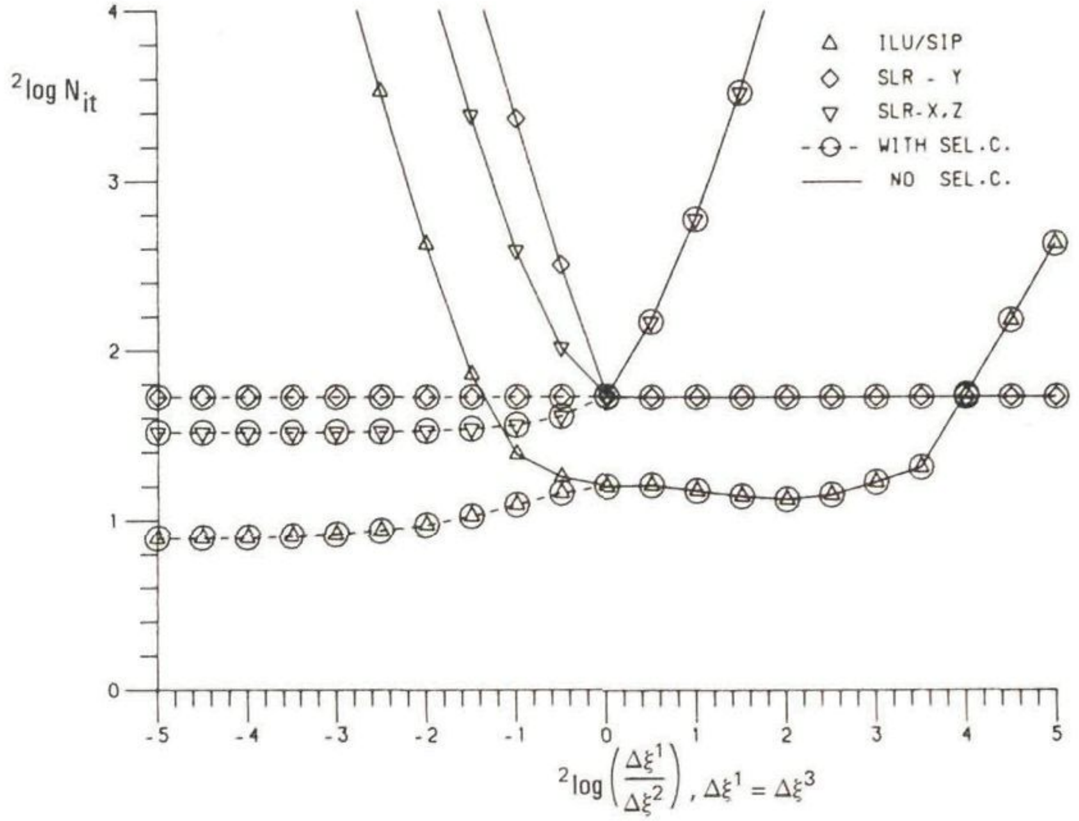


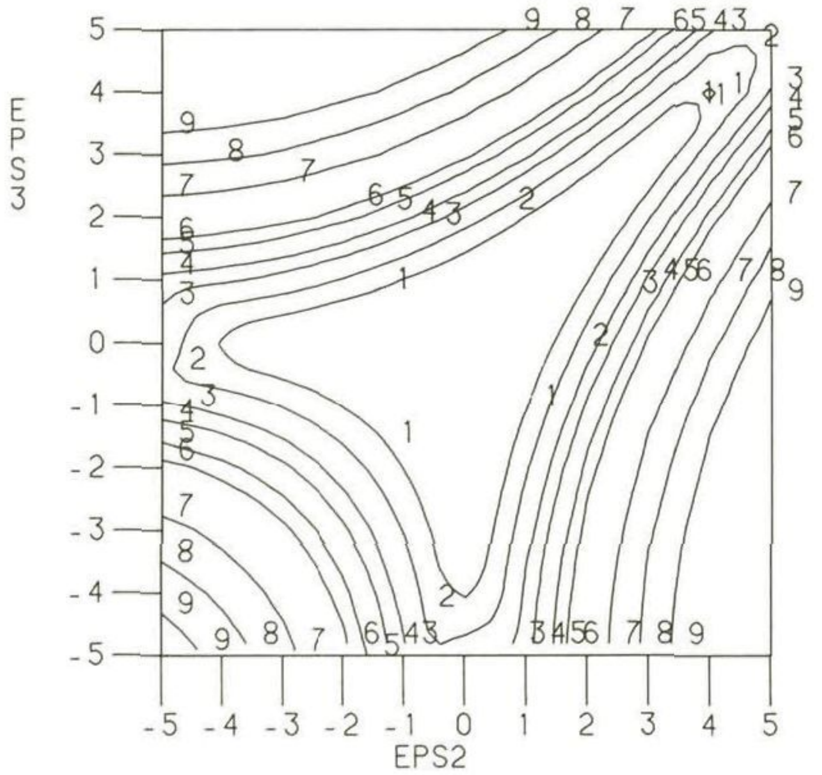
Fig. 6.1.3 Smoothing efficiency N_{it} for various algorithms for the special case $\Delta\xi^1 = \Delta\xi^3$ on a rectangular grid

It can be shown that in the general case coarsening should be applied only in the direction where $\Delta\xi^i$, i = 1,2,3, is shortest. In realistic cases, where the whole variety of mesh sizes considered in table 6.1.1 is present, this obviously leads to a very complex multigrid method.

Finally, in figure 6.1.4 the iso-$\bar{\rho}$ plots are shown for ILU/SIP with and without selective coasening. In these plots $\Delta\xi^1$ has been set to unity, while $\Delta\xi^2$ and $\Delta\xi^3$ vary. The iso-$\bar{\rho}$ values in figure 6.1.4 of lines identified by 1, 2,....,10 correspond to N_{it} (= $-1/{}^{10}\log\bar{\rho}$) values of 2, 3.5, 5, 7.5, 10, 15, 20, 50, 100 and 200 respectively. The plots show in a quantitatively more detailed fashion what already has been deduced qualitatively from table 6.1.1 and figure 6.1.3.

6.1.3 Influence of grid skewness on smoothing properties

The influence of grid skewness (non-orthogonality) on the smoothing properties of ILU/SIP can best be analyzed on a grid with unit aspect ratios, because on a grid having aspect ratios unequal to the unity the

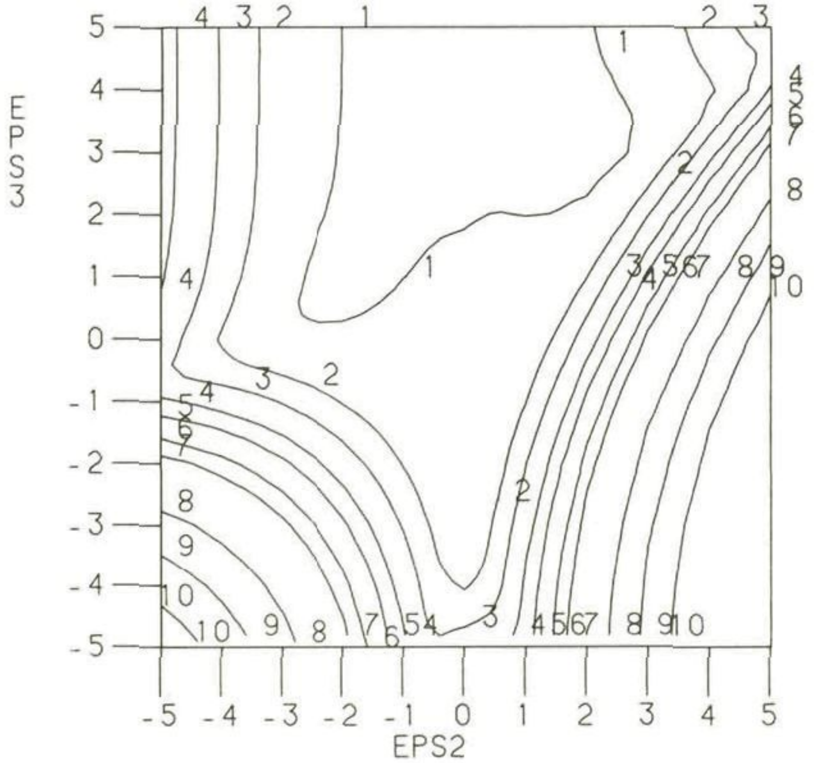


CURVE	$\bar{\rho}$
1	.518
2	.631
3	.736
4	.794
5	.858
6	.891
7	.955
8	.977
9	.989

EPS2 = $^2\log \Delta\xi^2$

EPS3 = $^2\log \Delta\xi^3$

a) ILU/SIP without selective coarsening



CURVE	$\bar{\rho}$
1	.316
2	.518
3	.631
4	.736
5	.794
6	.858
7	.891
8	.955
9	.977
10	.989

EPS2 = $^2\log \Delta\xi^2$

EPS3 = $^2\log \Delta\xi^3$

b) ILU/SIP with selective coarsening in ξ^2–direction

Fig. 6.1.4 Iso-$\bar{\rho}$ plots for ILU/SIP with and without selective coarsening as a function of $\Delta\xi^2, \Delta\xi^3$ on a rectangular grid; $\Delta\xi^1 = 1$

influences of aspect ratios easily tend to dominate the influences of grid skewness. Therefore, in this subsection $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$ is taken. The results obtained are found to be also valid for $\Delta\xi^i \neq 1$, at least in a qualitative sense.

Figure 6.1.5a shows the influence of the wing sweep angle θ_1 and the grid skewness angle θ_2 on the smoothing properties of ILU/SIP. In this figure the grid rotation angle θ_3 is taken equal to zero, i.e. the ξ^1-axis coincides with the x^1-axis. The figure shows primarily that ILU/SIP can become unstable for high wing sweep angles (> 60°) or extreme grid skewness angles (< 20°). Underrelaxation (see equation (6.1.8)) can however remove this deficiency (figure 6.1.5b).

The instability phenomenon can be explained as follows. Considering only two dimensions, the following equation is analyzed (equation 6.1.6):

$$g^{11}\,\varphi_{\xi^1\xi^1} + 2\,g^{12}\,\varphi_{\xi^1\xi^2} + g^{22}\,\varphi_{\xi^2\xi^2} = 0. \tag{6.1.9}$$

Deletion of crossderivative terms, see subsection 6.1.1, results in the following relaxation equation for the solution of $\Delta\varphi^n$ for a given ϕ^n (equation 6.1.8):

$$\begin{aligned}(g^{11}\,\partial^2_{\xi^1\xi^1} + g^{22}\,\partial_{\xi^2\xi^2})\Delta\varphi^n = \\ -\,\omega\,(g^{11}\,\partial^2_{\xi^1\xi^1} + 2g^{12}\,\partial_{\xi^1}\partial_{\xi^2} + g^{22}\,\partial^2_{\xi^2\xi^2})\,\phi^n\end{aligned} \tag{6.1.10}$$

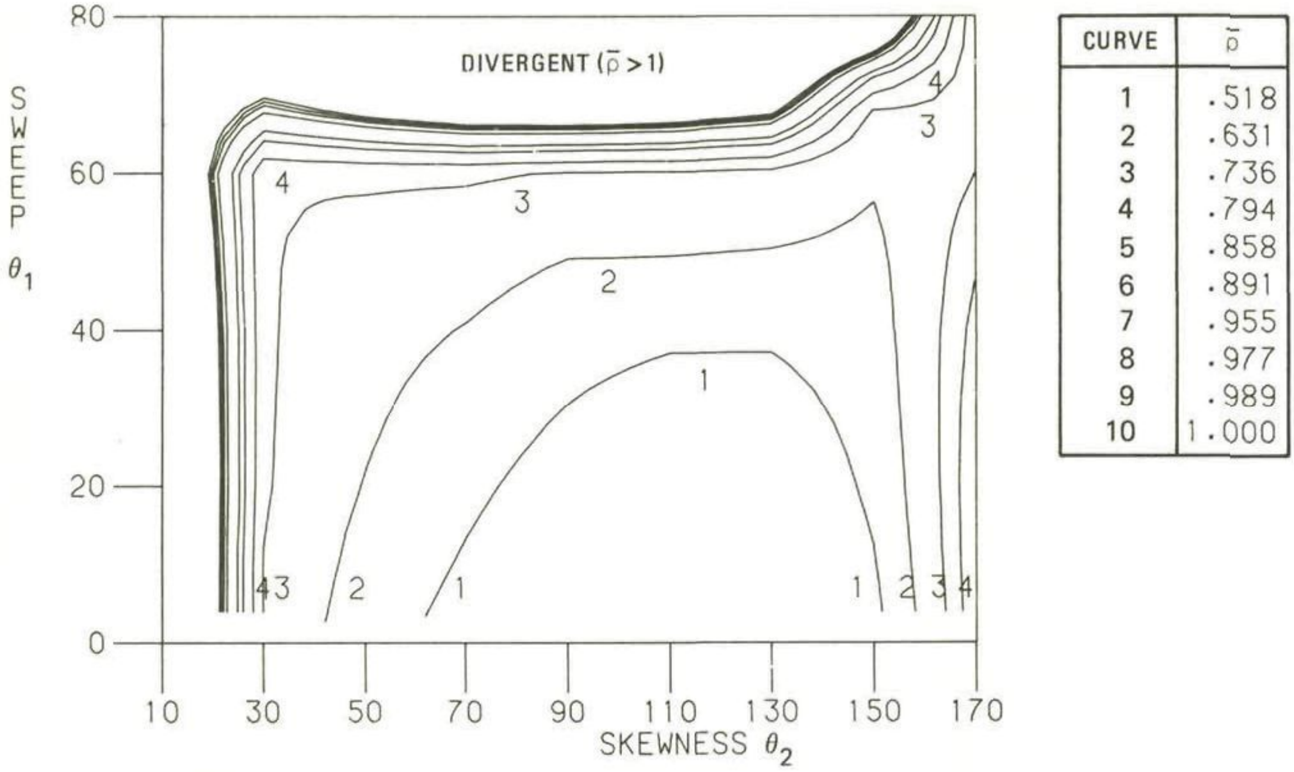
where ω is the underrelaxation factor mentioned previously in this section.

For $\theta_1 \approx 90°$, the ξ^1-axis becomes nearly aligned with the ξ^2-axis and consequently

$$g^{12} \approx g^{11} \approx g^{22}. \tag{6.1.11}$$

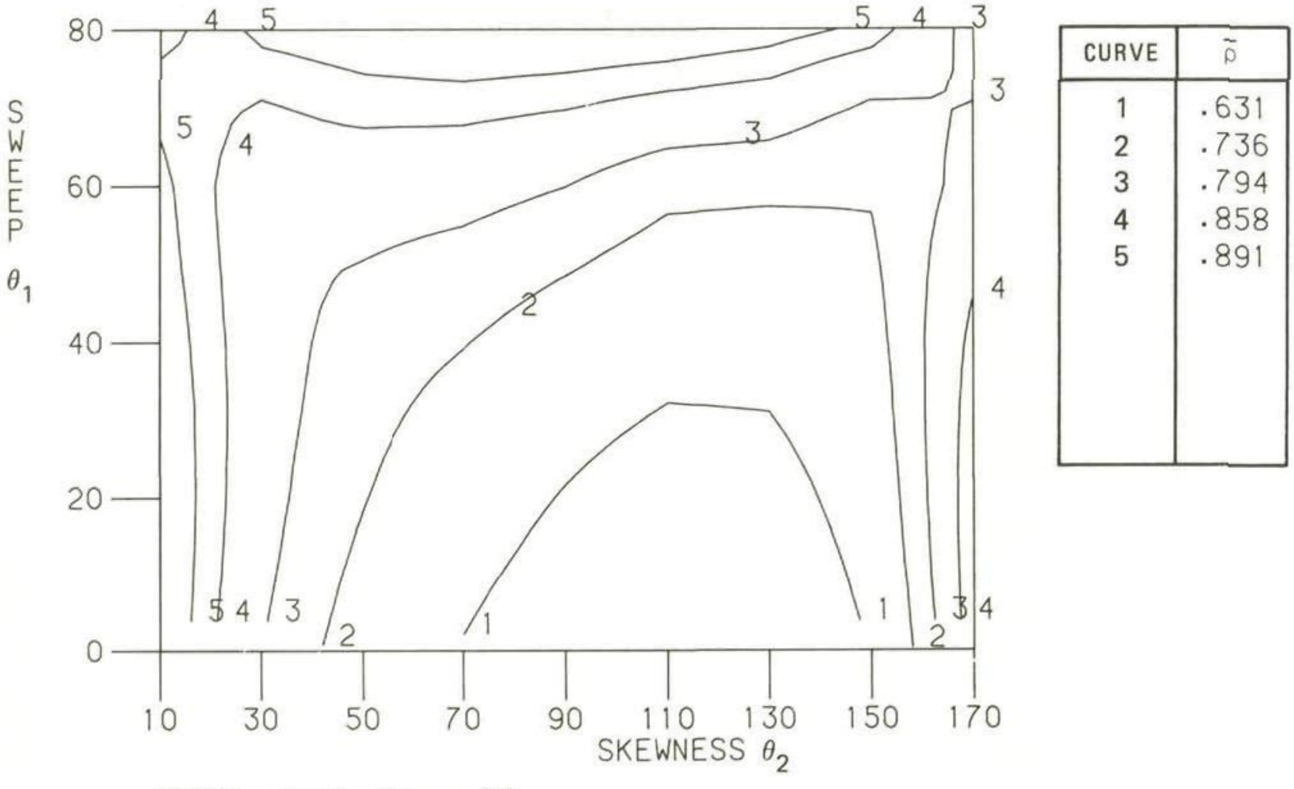
Then one effectively solves

$$(\Delta\varphi^n)_{\xi^1\xi^1} = -\,2\omega\,\phi^n_{\xi^1\xi^1}\,. \tag{6.1.12}$$



CURVE	$\bar{\rho}$
1	.518
2	.631
3	.736
4	.794
5	.858
6	.891
7	.955
8	.977
9	.989
10	1.000

a) Without underrelaxation



CURVE	$\bar{\rho}$
1	.631
2	.736
3	.794
4	.858
5	.891

b) With underrelaxation, $\omega = 0.7$

Fig. 6.1.5 Iso-$\bar{\rho}$ plots for ILU/SIP with and without underrelaxation as a function of wing sweep and grid skewness; $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$, $\theta_3 = 0^o$

Using equations (6.1.1) to (6.1.3) the reduction factor ρ follows from

$$(G-1)\ (L+B)\ e^n = -2\omega L e^n\ ,\ L = \partial^2_{\xi^1\xi^1}. \tag{6.1.13}$$

A close examination of the numerics shows that there exist Fourier components e^n for which $Be^n \approx -\varepsilon L e^n$, ε small and positive, and therefore

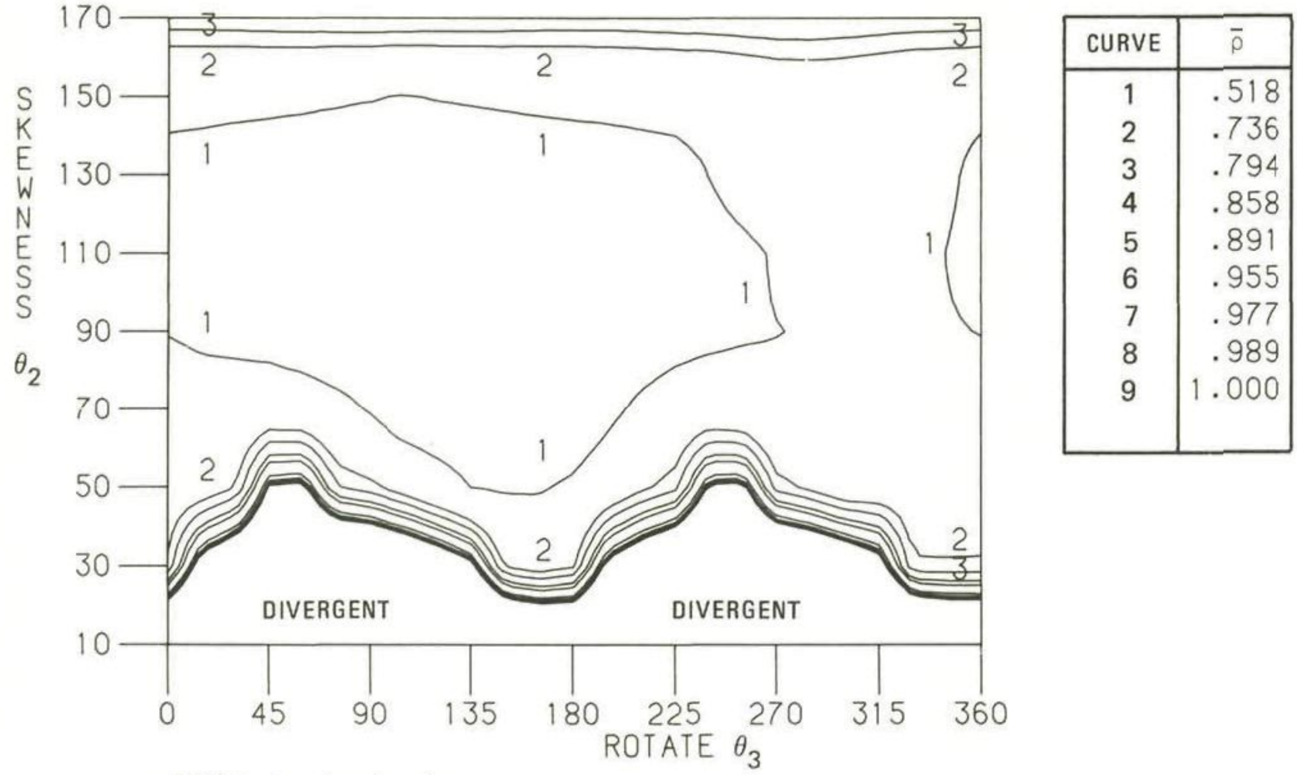
$$\rho = |G| = \left|1 - \frac{2\omega}{1-\varepsilon}\right| = \tag{6.1.14}$$

Consequently the algorithm can be divergent for $\omega = 1$ on very skew grids, but taking $\omega < 1$ can remove this difficulty. The value $\omega = .70$ appears to be effective for suppressing divergence phenomena in all possible cases. Of course, it is also possible to extend the sparsity pattern used in the ILU/SIP algorithm such that all crossderivative terms can be accomodated within this pattern, but, as already mentioned in subsection 5.4.2, this will severely affect the cost-effectiveness of the method in less skew cases.

Figure 6.1.6 shows the smoothing number as a function of the skewness angle θ_2 and the grid rotation angle θ_3 at a given wing sweep $\theta_1 = 30°$, which is a typical value for civil aircraft configurations. Once more, ILU/SIP shows unstable behaviour, which can be removed by underrelaxation.

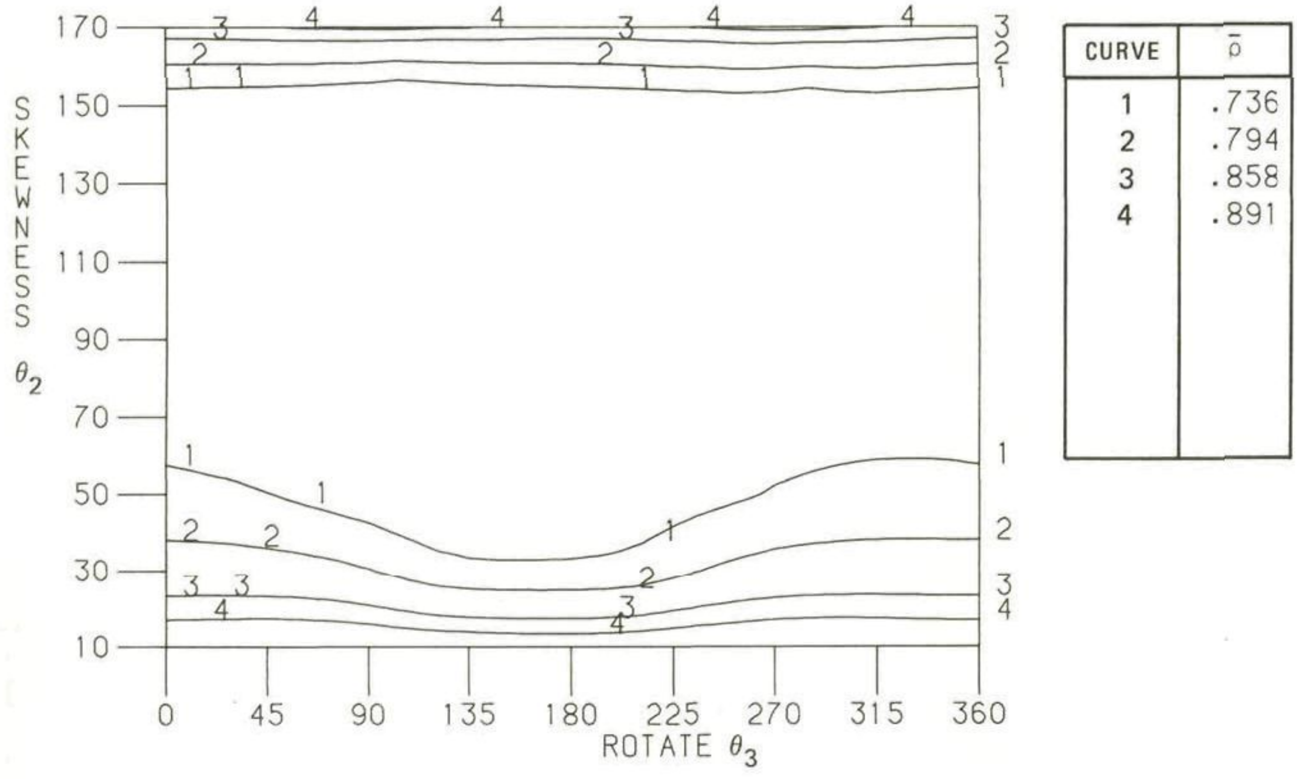
In Figure 6.1.7 the smoothing number is given as a function of the wing sweep θ_1 and the grid rotation angle θ_3 at a given grid skewness angle, which has been set to 45 degrees here. The figure shows that ILU/SIP becomes unstable for nonzero grid rotation angles at high wing sweep angles. Once more, underrelaxation by a factor $\omega = .70$ stabilizes the algorithm.

The conclusion can be drawn that ILU/SIP can show an unstable behaviour due to high wing sweep angles and/or large skewness angles in the grid. The unstability is caused by the deletion of crossderivative terms in the relaxation process and can be removed by underrelaxation with a factor $\omega = .70$. The price paid for applying this underrelaxation is however considerable. Figure 6.1.8 compares ILU/SIP smoothing efficiencies on a rectangular grid with and without underrelaxation. Because



CURVE	$\bar{\rho}$
1	.518
2	.736
3	.794
4	.858
5	.891
6	.955
7	.977
8	.989
9	1.000

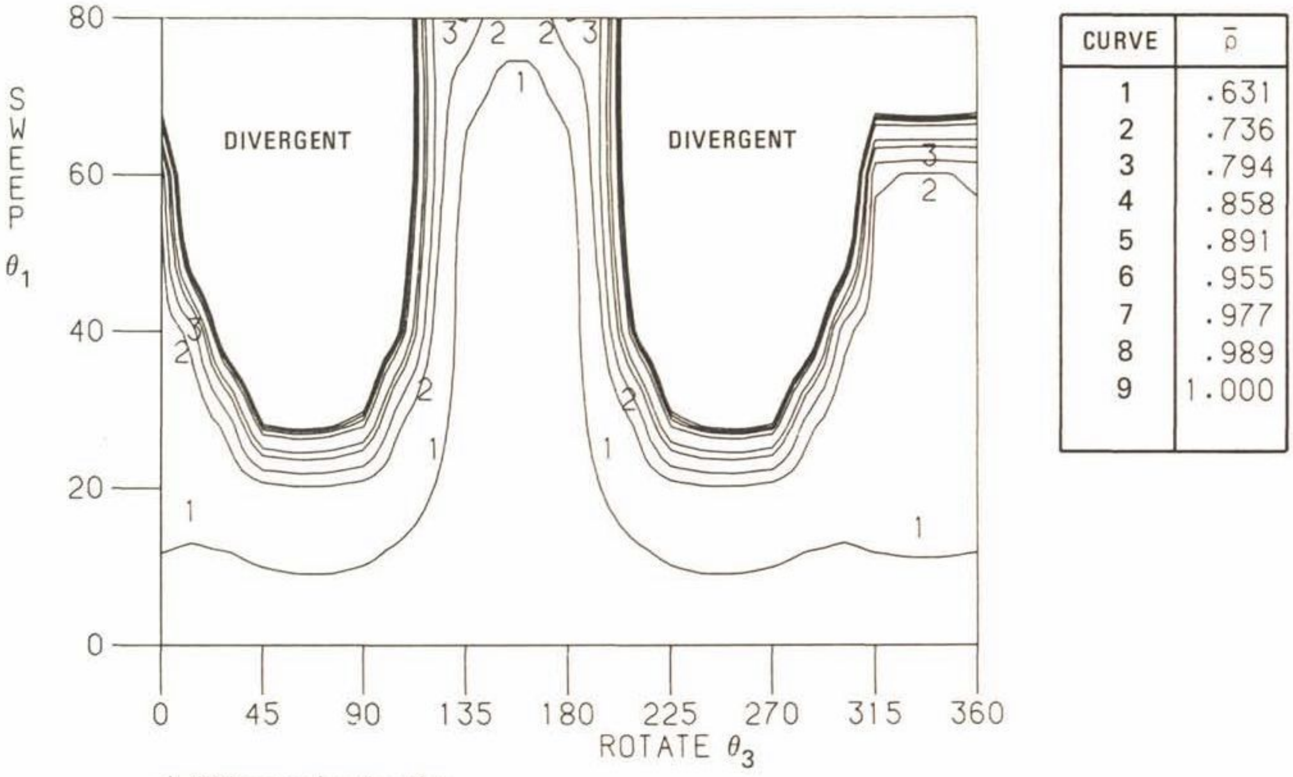
a) Without underrelaxation



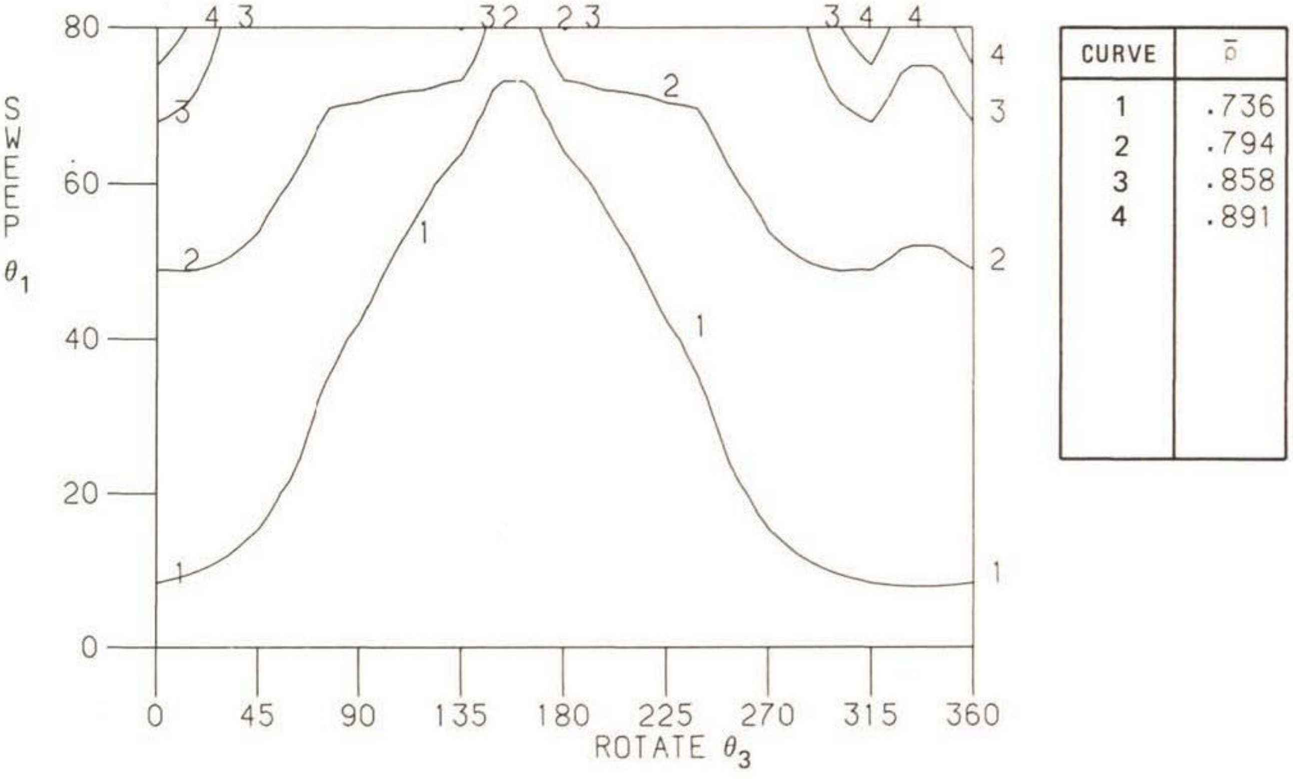
CURVE	$\bar{\rho}$
1	.736
2	.794
3	.858
4	.891

b) With underrelaxation, $\omega = 0.7$

Fig. 6.1.6 Iso-$\bar{\rho}$ plots for ILU/SIP with and without underrelaxation as a function of grid skewness and grid rotation; $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$, wing sweep $\theta_1 = 30^\circ$

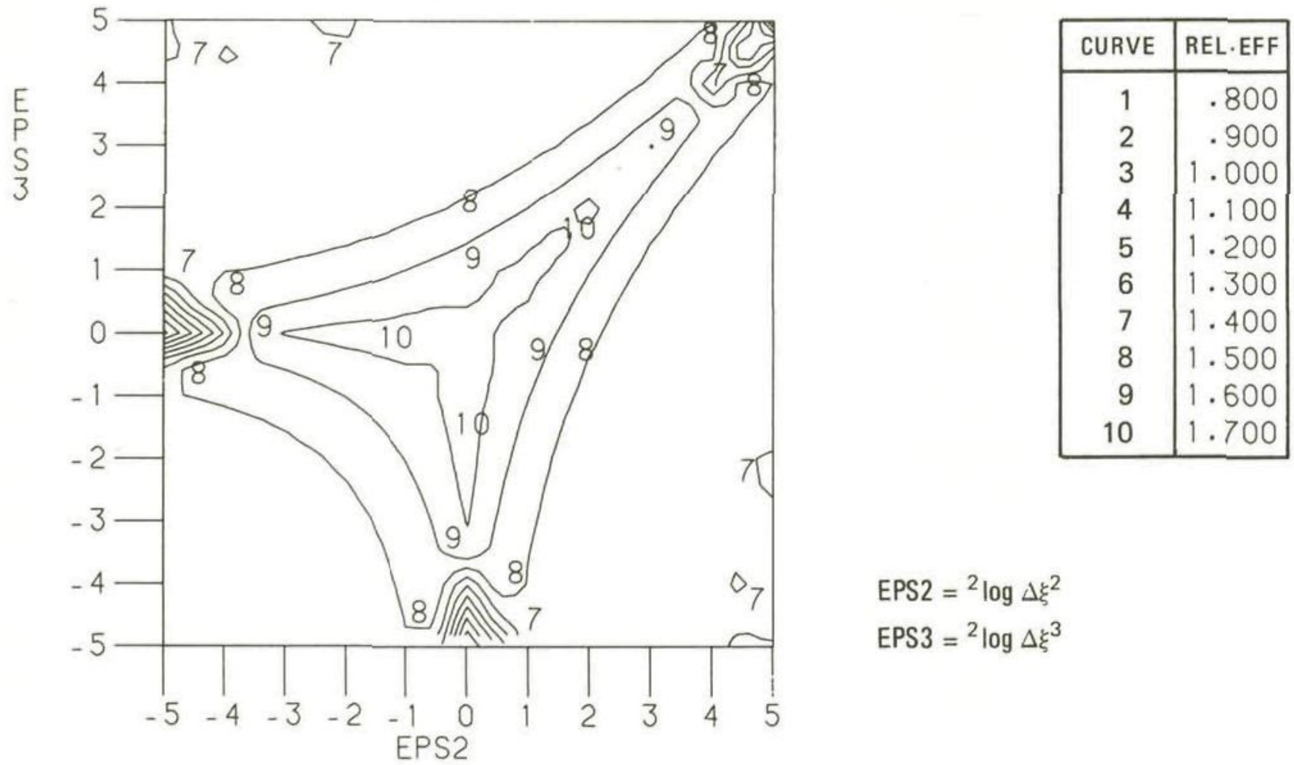


a) Without underrelaxation



b) With underrelaxation, $\omega = 0.7$

Fig. 6.1.7 Iso-$\bar{\rho}$ plots for ILU/SIP with and without underrelaxation as a function of wing sweep and grid rotation; $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3$; grid skewness $\theta_2 = 45^\circ$

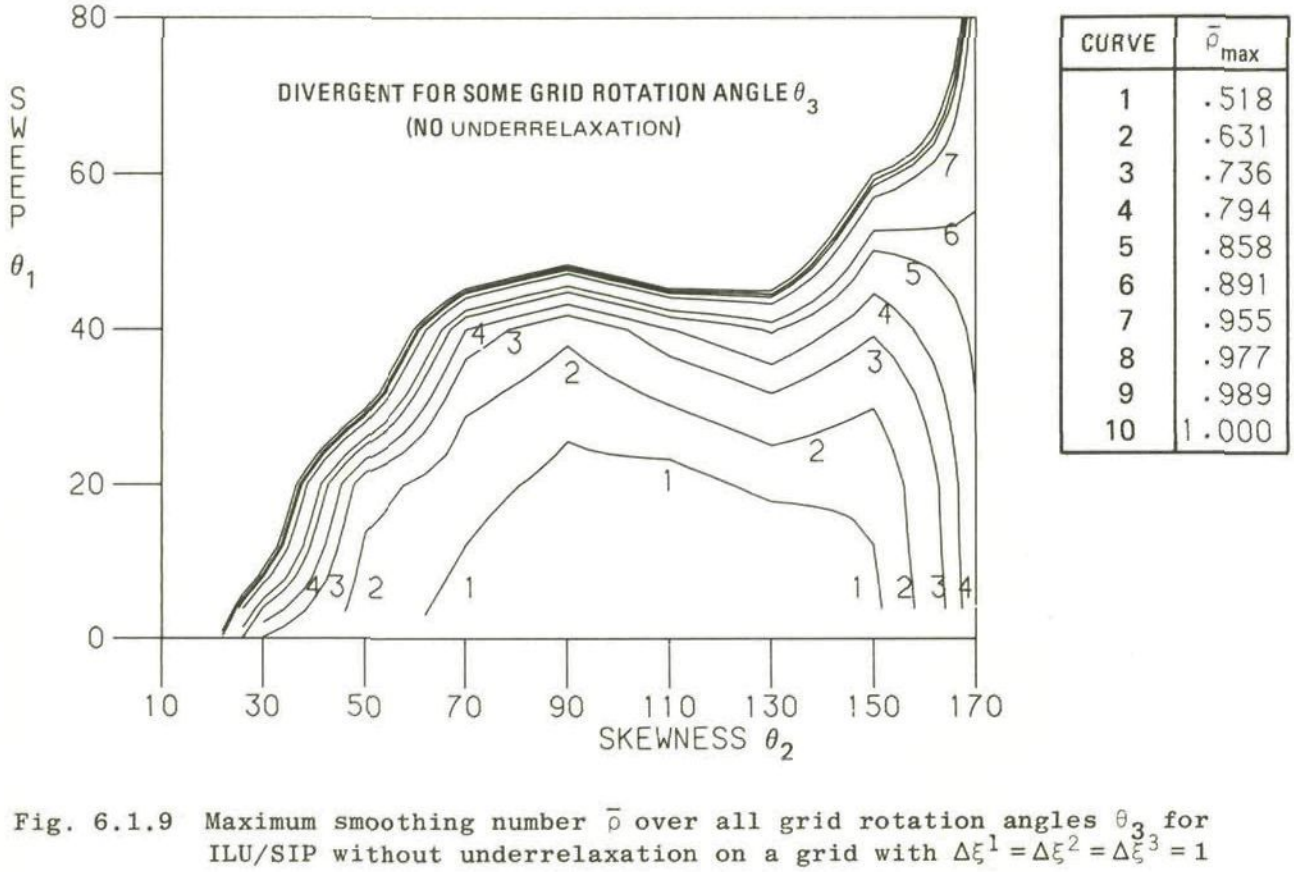


CURVE	REL·EFF
1	.800
2	.900
3	1.000
4	1.100
5	1.200
6	1.300
7	1.400
8	1.500
9	1.600
10	1.700

EPS2 = $^2\log \Delta\xi^2$

EPS3 = $^2\log \Delta\xi^3$

Fig. 6.1.8 Iso-lines of relative smoothing efficiency of ILU/SIP with versus without underrelaxation, i.e. $N_{it}(\omega = .7)/N_{it}(\omega = 1)$, on a rectangular grid; $\Delta\xi^1 = 1$



CURVE	$\bar{\rho}_{max}$
1	.518
2	.631
3	.736
4	.794
5	.858
6	.891
7	.955
8	.977
9	.989
10	1.000

Fig. 6.1.9 Maximum smoothing number $\bar{\rho}$ over all grid rotation angles θ_3 for ILU/SIP without underrelaxation on a grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$

the grid is rectangular, underrelaxation is not necessary in this case. The figure shows that underrelaxation generally causes a convergence slow-down of 40 - 50%, which is quite substantial. Therefore, it seems appropriate to try to avoid underrelaxation by conditioning the gridgenerator such that grid skewness is not great. Figure 6.1.9 shows to what extent sweep and skewness are allowable on a grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3$. The figure shows that in general a 45 degrees deviation from rectangularity will form no problem for the flow solver; however, at more severe deviations from rectangularity instability problems can, but need not necessarily occur, depending on the mesh sizes of the computational cell considered.

6.2 Stability analysis of ILU/SIP for hyperbolic problems

The application of ILU/SIP to hyperbolic problems will be analyzed using the modified equation method. Using this heuristic technique, an iterative process is associated with a time-integration process in the following way:

$$\phi^{n+1} - \phi^n = \Delta t \, \overleftarrow{\partial}_t \phi^{n+1} \qquad (6.2.1)$$

Consequently, the iterative process is associated with the time-integration of an initial value problem for a differential equation, the so-called modified equation. Let this initial value problem be well posed and let furthermore the integration process for the initial value problem be stable. Then it may safely be assumed that the iterative process is stable.
A more extensive discussion of the modified equation method can be found in Jameson [1].

Firstly we will show that the initial value problem for the modified equation is well posed. This will be established by adding an explicit so-called temporal damping term to the modified equation as derived in subsection 5.4.1.
It has been shown in subsection 5.4.1 that the modified equation of SIP has the following form (equation (5.4.13))

$$(e\Delta\xi^i\Delta\xi^i \frac{\partial}{\partial\xi^i}\frac{\partial}{\partial\xi^j} + \ldots\ldots) \, \Delta t \, \overleftarrow{\partial}_t \phi^{n+1} = g[\phi^n] - \tilde{L}[\phi^n] \, \phi^{n+1} \qquad (6.2.2)$$

where $\tilde{L}[\phi^n]\phi^{n+1}$ is given by equation (5.3.3):

$$\tilde{L}[\phi^n]\phi^{n+1} = \Delta\xi^i \, \Delta\xi^j \, (g^{ij} - \frac{\bar{U}^i\bar{U}^j}{\bar{a}^2}) \frac{\partial}{\partial\xi^i} \frac{\partial}{\partial\xi^j} \phi^{n+1} \tag{6.2.3}$$

In equations (6.2.2), (6.2.3) the quantities with bars are evaluated for $\varphi = \phi^n$; the unit mesh sizes $\Delta\xi^i$, $i = 1,2,3$, have been reintroduced in these equations for later use.

In streamline coordinates equation (6.2.3) can be written as (cf. equation (2.4.9)).

$$\tilde{L}[\phi^n]\phi^{n+1} = (1-\bar{M}^2) \, (\Delta s)^2 \, \phi^{n+1}_{ss} + (\Delta m)^2 \, \phi^{n+1}_{mm} + (\Delta n)^2 \, \phi^{n+1}_{nn} \tag{6.2.4}$$

Because $g[\phi^n]$ does not influence the character of the modified equation, see equation (5.4.7), we will investigate the following form of the modified equation (6.2.2):

$$(e\Delta\xi^i\Delta\xi^j \, \frac{\partial}{\partial\xi^i} \frac{\partial}{\partial\xi^j} + \ldots\ldots\ldots) \, \Delta t \, \overleftarrow{\partial}_t\phi^{n+1} +$$
$$+ (1-\bar{M}^2) \, (\Delta s)^2 \, \phi^{n+1}_{ss} + (\Delta m)^2 \, \phi^{n+1}_{mm} + (\Delta n)^2 \, \phi^{n+1}_{nn} = 0 \tag{6.2.5}$$

with $\bar{M}>1$ in supersonic (hyperbolic) regions of the flow.

Not much is known about the third order equation (6.2.5). We therefore explicitly add a well-chosen temporal damping term to this equation:

$$- 2\alpha\Delta t\overleftarrow{\partial}_t\Delta s \, \phi^{n+1}_s + (e\Delta\xi^i\Delta\xi^j \, \frac{\partial}{\partial\xi^i} \frac{\partial}{\partial\xi^j} + \ldots.) \, \Delta t \, \overleftarrow{\partial}_t\phi^{n+1} +$$
$$+ (1-\bar{M}^2) \, (\Delta s)^2 \, \phi^{n+1}_{ss} + (\Delta m)^2 \, \phi^{n+1}_{mm} + (\Delta n)^2 \, \phi^{n+1}_{nn} = 0, \; \alpha>0. \tag{6.2.6}$$

In Jameson [1] it is explained why the added term in equation (6.2.6) is the proper temporal damping term for a hyperbolic equation.

It will be shown that the modified equation (6.2.6) is a well posed initial value problem. In the limit $\Delta\xi^i$, $\Delta\xi^j \to 0$ the second term in equation (6.2.6) can be neglected in comparison with the first term of that equation. Applying a proper scaling we may put $\Delta t = \Delta s = \Delta m = \Delta n$ and consequently equation (6.2.6) can be written as

$$- 2\alpha\overleftarrow{\partial}_t \, \phi^{n+1}_s + (1-\bar{M}^2) \, \phi^{n+1}_{ss} + \phi^{n+1}_{mm} + \phi^{n+1}_{nn} = 0, \; \alpha>0. \tag{6.2.7}$$

We now introduce the following variable:

$$T = t - \frac{\alpha s}{\bar{M}^2-1} \, , \; \alpha>0. \tag{6.2.8}$$

Then equation (6.2.7) can be written as

$$\frac{\alpha^2}{\bar{M}^2-1}\,\phi_{TT}^{n+1} + (1-\bar{M}^2)\,\phi_{ss}^{n+1} + \phi_{mm}^{n+1} + \phi_{nn}^{n+1} = 0,\ \alpha>0. \tag{6.2.9}$$

This equation is hyperbolic for $\bar{M}>1$, where s is the time-like coordinate and T, m and n are spatial coordinates. Consequently, t (the iteration direction) is a mixed time-like/spatial coordinate. Equation (6.2.9) is known to be well-posed.

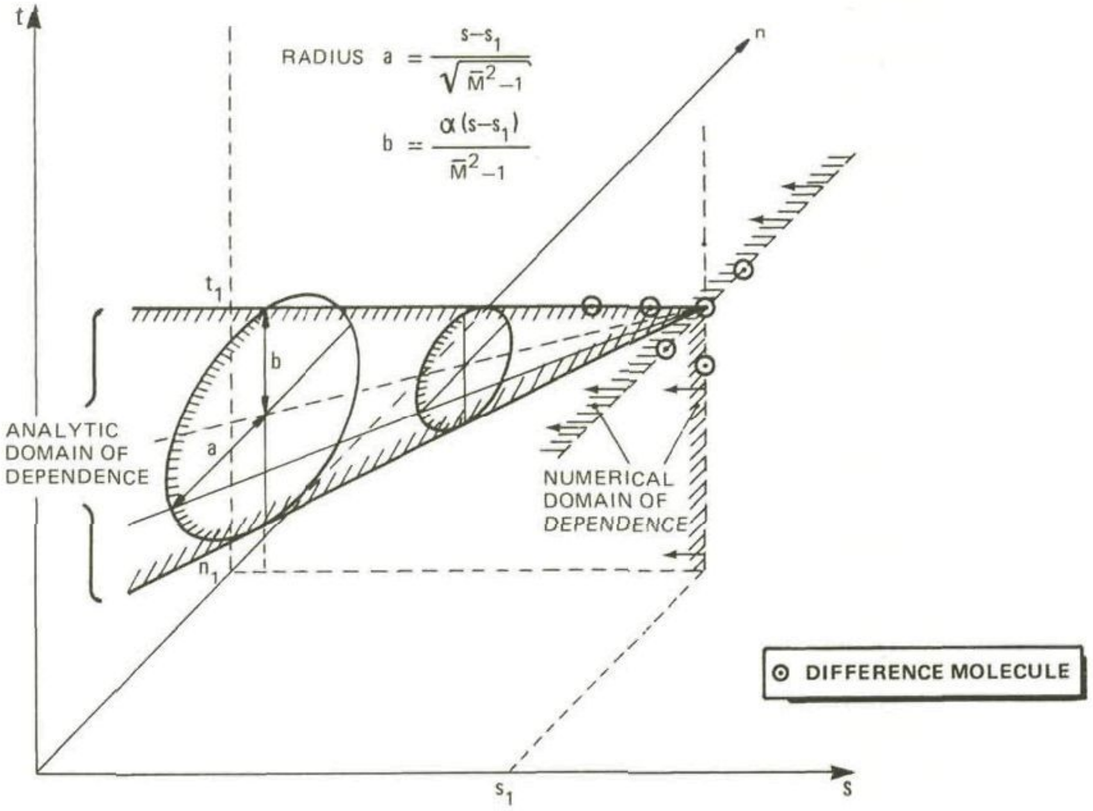
Subsequently we will show that the integration process for the initial value problem (6.2.7) is stable. We therefore investigate the characteristics of equation (6.2.9) in point (s_1,T_1,m_1,n_1):

$$\left(\frac{s-s_1}{\sqrt{\bar{M}^2-1}}\right)^2 = \left(\frac{\sqrt{\bar{M}^2-1}}{\alpha}\,(T-T_1)\right)^2 + (m-m_1)^2 + (n-n_1)^2, \tag{6.2.10}$$

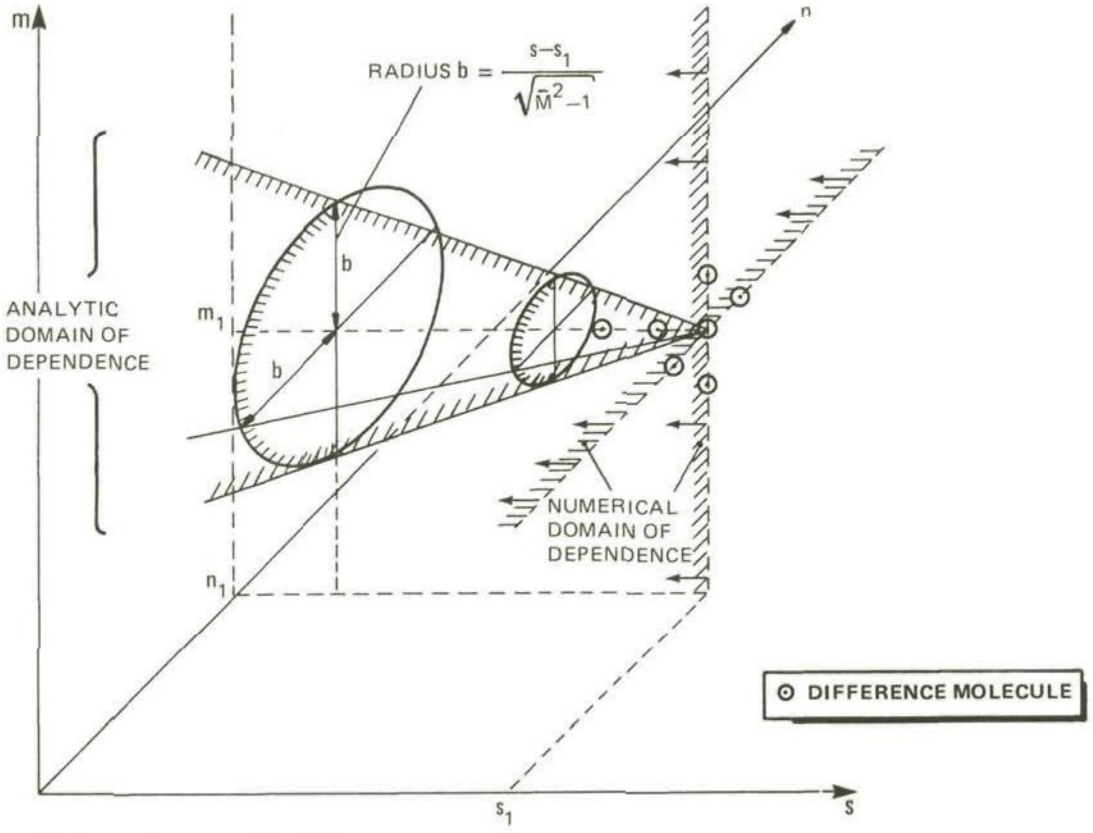
where $T = T(t,s)$ is given by equation (6.2.8).
For the purpose of this analysis, the mass flux upwind scheme presented in subsection (4.4.4) can be regarded as a retarded differencing scheme in the time-like s-direction and as a central differencing scheme in the m- and n-direction. This is exactly true if (compare equation (4.4.27)) the stream direction s is along one of the computational coordinate axes; otherwise it is true by approximation.
In the iteration direction t, a retarded differencing scheme is applied, see equation (6.2.1).

The characteristics given in equation (6.2.10) and the difference molecule described above are assembled in figure 6.2.1. This figure shows that the analytical domain of dependence is embedded in the numerical domain of dependence and consequently the integration algorithm for the initial value problem (6.2.7) is stable. The figure shows that the integration algorithm is in fact a time-stepping scheme, both in the s (time-like) and in the t (iteration) direction, while solving implicitly per m,n-plane of constant s and t.

It thus has been shown heuristically that the integration process for the initial value problem (6.2.7) is stable, which is due to the addition of the temporal damping term in equation (6.2.6). Consequently the ILU/SIP iteration algorithm with the temporal damping term is stable in hyperbolic parts of the flow, and can be associated with an integration algorithm in the s-direction while advancing the solution one step in the iteration direction t.


t
n
RADIUS $a = \frac{s-s_1}{\sqrt{\bar{M}^2-1}}$
$b = \frac{\alpha(s-s_1)}{\bar{M}^2-1}$
t_1
b
a
n_1
ANALYTIC DOMAIN OF DEPENDENCE
NUMERICAL DOMAIN OF DEPENDENCE
⊙ DIFFERENCE MOLECULE
s_1
s

a) (t, n, s)–space $(m = m_1)$


m
n
RADIUS $b = \frac{s-s_1}{\sqrt{\bar{M}^2-1}}$
b
b
m_1
n_1
ANALYTIC DOMAIN OF DEPENDENCE
NUMERICAL DOMAIN OF DEPENDENCE
⊙ DIFFERENCE MOLECULE
s_1
s

b) (m, n, s)–space $(T = T_1)$

Fig. 6.2.1 Characteristics and domains of dependence for equation (6.2.9) in the (t,n,s)-space $(m = m_1)$ and in the (m,n,s)-space $(T = T_1)$

In the ILU/SIP algorithm the temporal damping term is implemented as follows. Instead of applying the ILU/SIP incomplete decomposition to $\tilde{L}$ in equation (5.4.1), where $\tilde{L}$ is given by equation (5.3.3), the ILU/SIP algorithm is applied (in the supersonic (hyperbolic) region of the flow) to

$$\begin{aligned} &\tilde{\tilde{L}}[\phi^n]\Delta\varphi^n = f - L\phi^n, \\ &\tilde{\tilde{L}}\,[\phi^n]\Delta\varphi^n = \tilde{L}[\phi^n]\Delta\varphi^n - \alpha\Delta s\,\frac{\bar{U}^i}{\bar{q}}\,\frac{\partial}{\partial\xi^i}\,\Delta\varphi^n \end{aligned} \tag{6.2.11}$$

where the backward difference operator $\overleftarrow{\delta}_i$ is used to discretize $\frac{\partial}{\partial\xi^i}$ and α is chosen of the form

$$\alpha = \bar{\rho}\bar{q}\ (C_5(\bar{M}^2-1) + (1-M_c^2))\ ,\ C_5 = 1\ ,\ M_c = .7. \tag{6.2.12}$$

Experience indicates that C_5 and $1-M_c^2$ must be chosen large enough, so that the added explicit temporal damping term is dominant over the second term in equation (6.2.6). On the other hand, α shall be kept as small as possible, so that the opening angle of the characteristic cone in the s,t-plane is kept as small as possible, compare figure 6.2.1, thus minimizing "history"-effects in the time-integration (i.e. iteration) process.

7 IMPLEMENTATION ASPECTS

7.1 General remarks on ILU/SIP

On many large computers the size of the main memory is not more than about 2-4 million numbers. This implies that the computational data cannot be loaded entirely in memory.

An important requirement with respect to the choice of a relaxation algorithm in the multigrid method has therefore been that it should *be possible to implement it on a computer of large, but limited, main* memory (.5-4 million numbers). Therefore the algorithm has been designed to have a plane-by-plane structure, so that a code can be set up requiring only a limited number of planes in main memory at the same moment. Vectorizability is desirable but not an absolute requirement.

The evaluation of residuals is expensive in finite volume codes. Because residuals have to be evaluated in each relaxation, even a rather expensive algorithm can be cost-effective, provided its convergence is fast. For this reason the reduction-factors in section 6.1 have not been weighted by the amount of work required for each algorithm, because if the evaluation of residuals is expensive, the work for each algorithm will be about the same per iteration.

Because it is fully implicit, the ILU/SIP algorithm requires that the entire computational domain is updated simultaneously. Hence the whole upper triangular matrix U must be stored, requiring storage-capacity for four large vectors (of length equal to the solution-vector). As in general this upper matrix U will not be stored in main memory, IO-transfers have to be made in each relaxation. These transfers can be carried out plane-by-plane. The absence of a preferred sweep-direction (see section 5.4.2) can be used to advantage by choosing the sweep-direction and the number of planes to be transferred simultaneously in such *a way that an optimal balance between main memory size and IO-time* is obtained.

The vectorizibility of the ILU/SIP algorithm has not yet been thoroughly investigated. The only non-vectorizible part in the algorithm is in its present form the inversion of the inner (pentadiagonal) bandmatrix. In the context of SLR-algorithms, three colour pattern relaxation has proved to be an effective way to vectorize such inversion algorithms and it is believed that this can also be applied to ILU/SIP.

7.2 Storage and computational speed requirements

The total memory requirement of the ILU/SIP algorithm, the multigrid method and the finite volume discretization is given in table 7.2.1: the code requires background memory for about 23 numbers per grid point.

Experience has shown that a sufficiently accurate transonic potential flow calculation about a realistic transport wing can require in the order of 300.000 grid points on a CH-topology grid. This indicates that sufficient accuracy in the case of a realistic aircraft configuration (wing, body, nacelles etc.) can easily raise the number of grid points required to 1.000.000. Since the flow solver requires about 23 numbers per grid point, it follows that the total memory requirement can easily become 23 million (64 bit) words. Because of these considerations the flow solver has no upper limit of the size of computational grid(s) employed. The flow solver has been developed such that it does not arrange for I/O data handling on the program level. Consequently, the flow solver can be made operational on computers that have a sufficiently large main memory (say ≧ 25 million numbers) or a virtual operating system supported by a sufficiently large main memory (say ≧ 1.5 million numbers). The Cray 2 and the ETA-10 are in the first category while the NLR Cyber 180-855 and the Cyber 205 are in the second category. Also the Cray X-MP and NEC SX-2, having a comparably fast accessible memory on extended memory (solid state devices), are in the first category. An important side effects of the decision not to arrange for I/O data handling on the program level is that the software is simpler and consequently easier to develop and maintain.

The flow solver has been developed on a Cyber 180-855 sequential computer under the NOS/VE operating system supporting a virtual environment. This computer has at present a central memory of 3 million (64 bit) words and has a computation speed of 2.1 million floating point operations per second. The computation time required by the flow solver is at the moment about 12 seconds per 1000 grid points (.45 seconds per iteration per 1000 grid points) for a solution of engineering accuracy. This amounts to a computation time of about one hour for a realistic transport wing using the CH-topology grid of 300.000 grid points mentioned before. The corresponding turn-around time is about 3-4 times the computation time under stand-alone conditions.

TABLE 7.2.1.

Memory requirements on three sample grids in case of plane-by-plane ordering of the ILU/SIP algorithm in the multigrid method (K = kilo = 1000)

Grid point variable	Background memory per global grid point	Number of grid planes to be stored simultaneously in main memory	Main memory per grid plane grid point
1 disturbance potential	1	5	5
1 status variable	1	5	5
3 coordinates	3	0	0
6 metrics + 1 Jacobian	7	4	28
3 ILU/SIP variables + 1 residual	4	3	12
7 ILU/SIP work variables	0	1	7
3 freestream flow variables	3	4	12
sub-total	19		69
multigrid overhead (20%)	3.8		13.8
total	22.8		82.8

NX*NY*NZ grid / Storage		176*32*32 (= 180 K cells)	220*40*48 (= 442 K cells)	320*48*64 (= 983 K cells)
background memory		4109 K	9631 K	22413 K
total main memory in case of plane-by-plane ordering	NX*NY	466 K	729 K	1271 K
	NX*NZ	466 K	874 K	1696 K
	NY*NZ	85 K	159 K	254 K

About 2/3 of the computation time and the turn-around time is spent on computing the flow velocities, the flow density and the flow equation residual. About 1/4 of the computation and turn-around time is spent on the ILU/SIP smoothing algorithm.

It is clear that operation of the flow solver in a practical design environment (when turn-around times shorter than one quarter of an hour are required) requires a supercomputer like the aforementioned Cray 2, ETA-10, Cray X-MP or NEC SX-2.

8 NUMERICAL EXPERIMENTS

8.1 General description and basic choices

In this chapter numerical experiments are presented that show the computational efficiency and robustness of the MG-ILU/SIP method. Two types of experiments are described:

- in section 8.2 numerical experiments for simple cases are described, for confirmation of the smoothing analysis given in chapter 6;
- *in section 8.3 numerical experiments for practical cases are presented* to illustrate the usefulness of the method for practical transonic potential flow calculations.

Finally, in section 8.4 results of the finite volume discretization are given and transonic potential flow solutions are shown for the DFVLR-F4 and the ONERA-M6 wing.

Transonic flows are highly nonlinear in the vicinity of shocks. Shocks are captured by the finite volume scheme as narrow zones of steep pressure gradients. The positions of shocks are not known a priori and have to be found in the course of the solution process. The experiments must demonstrate the applicability of the algorithm to transonic flows with shocks. It will appear, however, that the asymptotic rate of convergence of the multigrid process is dominated by certain subsonic regions in the flow. These are:

- the vicinity of the leading and trailing edge of the wing, where one mesh size is considerably larger than the other two ($\Delta\xi^2 >> \Delta\xi^1, \Delta\xi^3$), especially in the far field region outboard the wing-tip (figure 3.1.6);
- regions where the three mesh sizes differ widely from each other; this can easily occur in the far field region in grid planes perpendicular to the wing.

Unless specified otherwise, the following basic choices have been made in the experiments:

- The ILU/SIP parameter α is set to unity in hyperbolic regions of the flow, while $0 \leq \alpha \leq 1$ in elliptic regions. It will be shown that α=.70 is a good value in the elliptic region, and in such cases this parameter will not be mentioned explicitly.
- Either a three level or a four level multigrid method is used, employing weighted restriction of residuals (figure 5.2.3), injection

of potentials and tricubic prolongation. In Van der Wees, van der Vooren and Meelker [1] some experiments are described which indicate that this is a good choice. A slightly faster convergence can be obtained by taking weighted restriction instead of injection of potentials. It has been found, however, that this choice is not robust, see subsection 8.3.1.

- A V- or W-cycle (figure 5.2.1) will be used, denoted by $V_N(x,y)$ or $W_N(x,y)$, in which x smoothings are performed on the coarsest grid and y smoothings on the finer grids after each prolongation, while N indicates the number of grid levels in the multigrid process. The number of smoothings on a grid will always be chosen such, that the high-frequency part of the error is sufficiently smoothed on that grid. A guideline for this choice is given by the smoothing analysis for elliptic problems, section 6.1.
- The computation starts with a full multigrid cycle (nested iteration, figure 5.2.2).
- The multigrid convergence rate is measured by monitoring the maximum norm of the residual on the finest grid versus the number of work units (WU). One work unit is equivalent to the work required for one fine grid smoothing. Work involved in prolongation and restriction is not counted.
- Discontinuities in the geometry, such as the wing trailing edge, are chosen to coincide with grid lines that are present on several grid levels. The wing-tip requires special treatment, as will be explained in subsection 8.3.1.
- The freestream flow is used as the starting solution.

8.2 Numerical experiments for confirmation of smoothing analysis

8.2.1 Incompressible flow in windtunnel with a bump on the bottom wall

The smoothing analysis presented in section 6.1 will be confirmed by computing the incompressible flow ($M_\infty = 0.$, hence we are solving the Laplace equation) in a windtunnel with a bump on the bottom wall. A sample configuration is given in figure 8.2.1.

The flow domain and the grid will be varied such that the influence of grid aspect ratio and grid skewness on the rate of convergence of the multigrid method can be compared with results of the smoothing analysis presented in section 6.1.

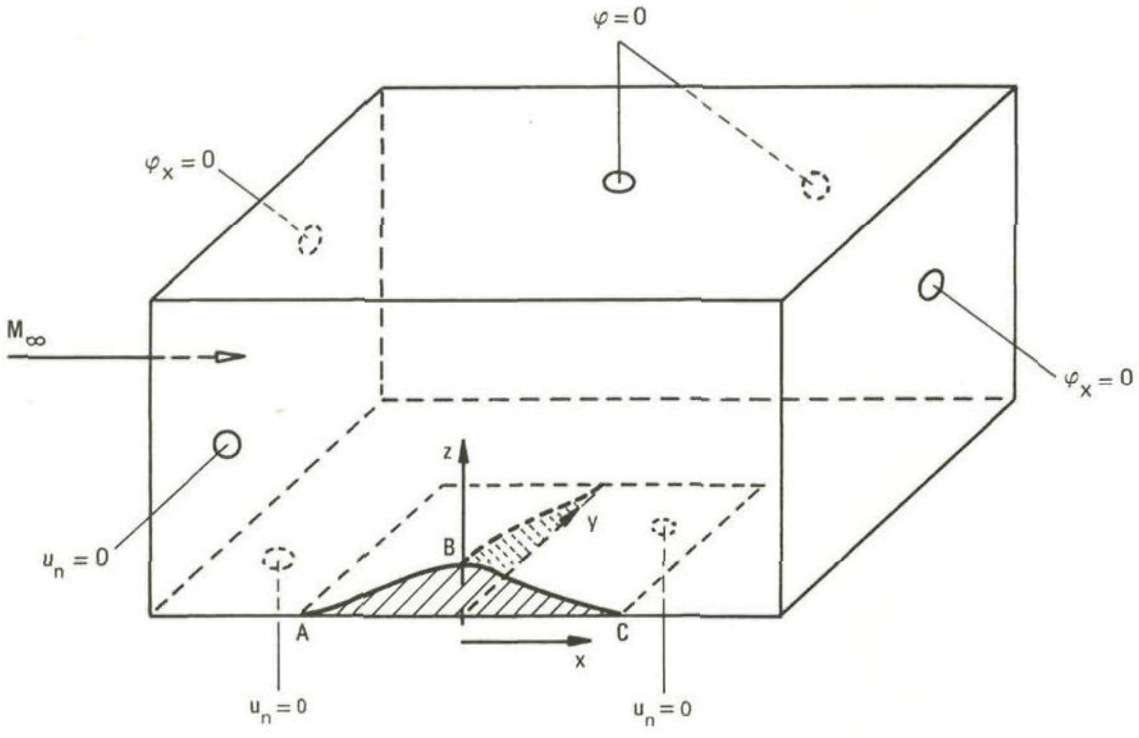


Fig. 8.2.1 Sketch of a windtunnel with a bump on the bottom wall

Flow configuration

The following flow domain is used (compare equation (6.1.7)):

$$\left.\begin{aligned}
&(\xi^1,\xi^2,\xi^3) \quad [-1,1] * [0,2] * [0,2] \\
&\bar{\xi}^1 = \Delta\xi^1 \,.\, \xi^1 \\
&\bar{\xi}^2 = \Delta\xi^2 \,.\, \xi^2 \\
&\bar{\xi}^3 = \Delta\xi^3 \,.(\xi^3 + \xi^3_{\text{bump}}) \\
&x = \bar{\xi}^1 \cos\theta_3 + \bar{\xi}^2 \sin\theta_1 + \bar{\xi}^3 \cos(\theta_2+\theta_3) \\
&y = \bar{\xi}^2 \cos\theta_1 \\
&z = \bar{\xi}^1 \sin\theta_3 + \bar{\xi}^3 \sin(\theta_2+\theta_3)
\end{aligned}\right\} \qquad (8.2.1)$$

where the bump is given by a third degree polynomial which has continuous derivatives at the edges of the bump:

$$\xi^3_{\text{bump}} = \varepsilon \ (2|\xi^1|-1)^2 \ (4|\xi^1|+1) \ (\xi^2-1)^2 \quad \text{for}$$

$$(\xi^1,\xi^2) \in [-\tfrac{1}{2},\tfrac{1}{2}] * [0,1], \tag{8.2.2}$$

$$= 0 \text{ elsewhere.}$$

In equation (8.2.1) $\Delta\xi^i$, $i = 1,2,3$, are parameters to stretch the three coordinate directions. The parameters θ_1, θ_2 and θ_3, are respectively the (given) sweep, skewness and grid rotation angle of the domain, compare figure 6.1.1. As in section 6.1, the above parameters will be varied to investigate their influence on the rate of convergence of the multigrid method.

The parameter ε in equation (8.2.2) controls the thickness of the bump. Thus the root section $\xi^2 = 0$ corresponds to a $2\varepsilon\Delta\xi^3/\Delta\xi^1$ thick cusped non-lifting airfoil. In the experiments $\varepsilon = .0005$.

The grid contains 32*32*32 cells and is equidistant in ξ^i-space. Consequently the mesh size in $\bar{\xi}^i$-direction is $2\Delta\xi^i/32$.
Due to the thinness of the bump the boundary fitted grid will hardly be distorted by the bump.

The coupling factor κ (subsection 4.4.3) is set to unity, so that effectively the Laplace equation is solved using central differences.

In this section the full multigrid cycle (nested iteration) is not used: the multigrid process is started on the finest grid level.

Optimal value of ILU/SIP-parameter α

Figure 8.2.2 presents the convergence history of the MG-ILU/SIP method on a grid with $\Delta\xi^1 = 1$, $\Delta\xi^2 = 10$, $\Delta\xi^3 = 1$, $\theta_1 = 0°$, $\theta_2 = 90°$, $\theta_3 = 0°$ for various values of the parameter α, employing a $V_4(60,6)$-cycle. The figure shows that the convergence improves with increasing α, as predicted in section 6.1, figure 6.1.2. The convergence for $\alpha = 1.$ was poor, as predicted in section 6.2.
As was also found in section 6.1 the value $\alpha = .7$ is reasonably optimal. This value for α will therefore be adopted in all subsequent numerical experiments of this subsection and will no longer be mentioned explicitly.

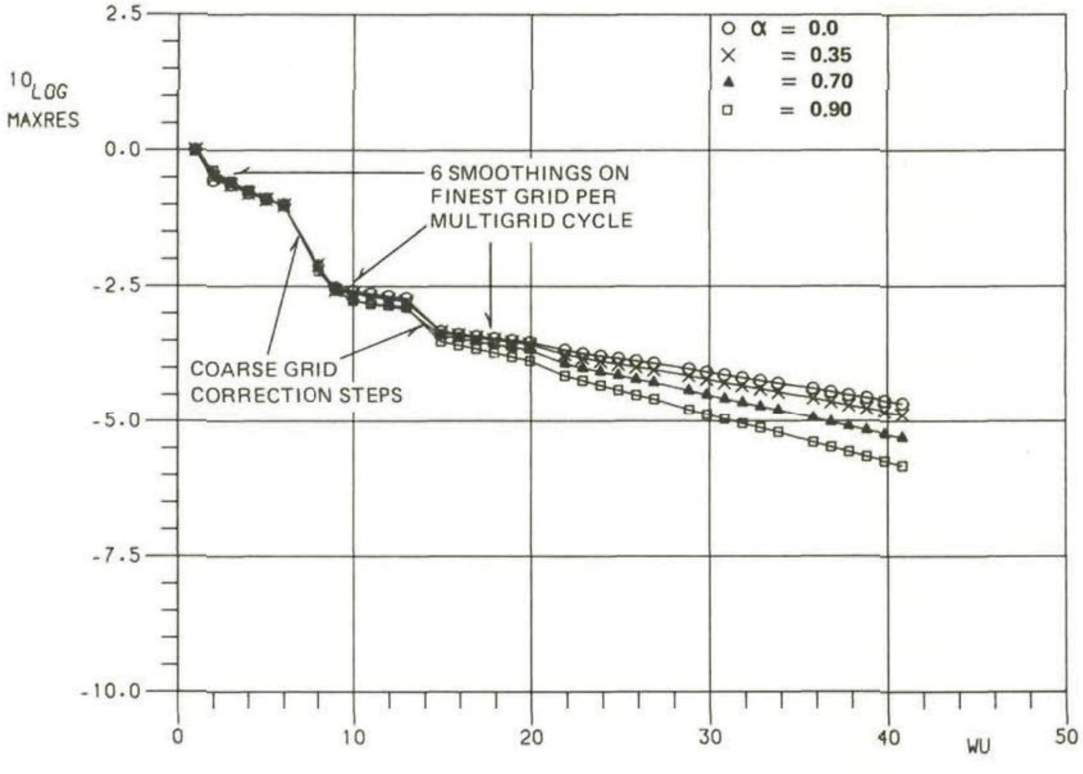


Fig. 8.2.2 MG-ILU/SIP(α) convergence history on a rectangular grid with $\Delta\xi^1 = \Delta\xi^3 = 1$, $\Delta\xi^2 = 10$, for various values of α

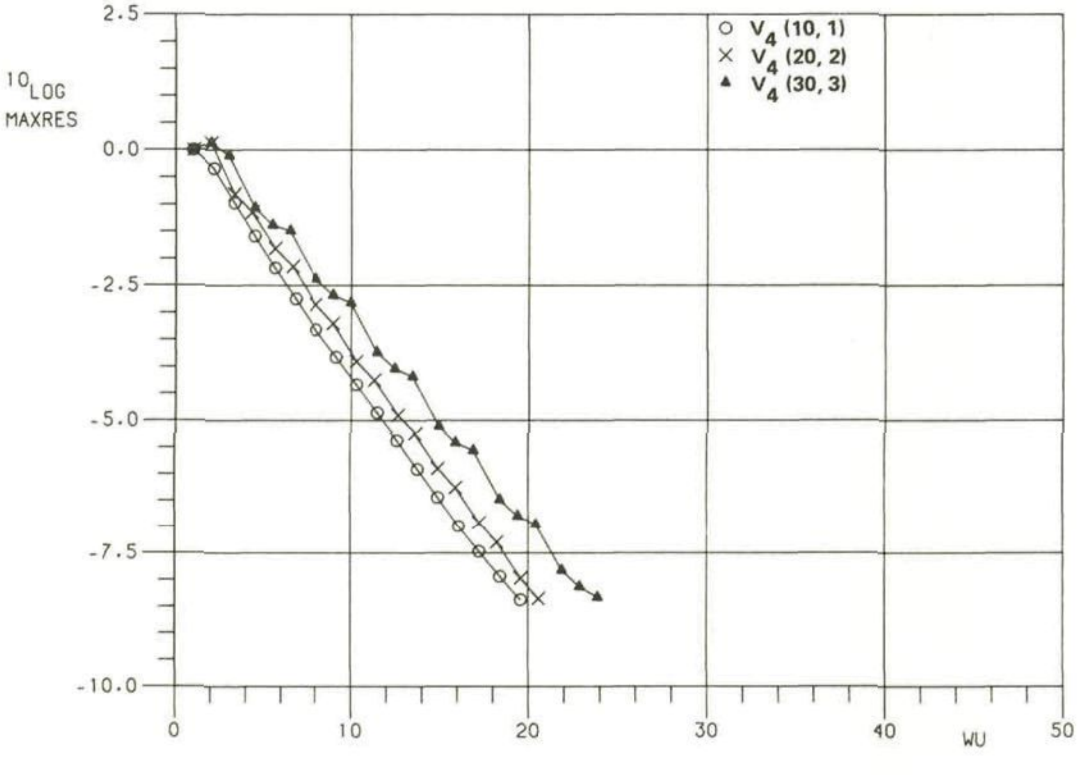


Fig. 8.2.3 MG-ILU/SIP convergence history on rectangular grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$ for various V_4-cycles

Optimal number of relaxations per grid

Figure 8.2.3 presents the MG-ILU/SIP convergence history of the MG-ILU/SIP method on a rectangular grid ($\theta_1 = 0°$, $\theta_2 = 90°$, $\theta_3 = 0°$) with $\Delta\xi^i = 1$, $i = 1,2,3$, depending on the number of relaxations in a V-cycle. As predicted by smoothing analysis (table 6.1.1), the convergence is faster than in the previous case. The optimal number of relaxations per grid is 2 here, but the differences are small.

Influence of grid aspect ratios on rate of convergence

Figure 8.2.4 shows the convergence history of the MG-ILU/SIP method on a rectangular grid ($\theta_1 = 0°$, $\theta_2 = 90°$, $\theta_3 = 0°$) for several values of the aspect ratio $\varepsilon = \Delta\xi^1/\Delta\xi^2$ with $\Delta\xi^1 = \Delta\xi^3$. This experiment is to be compared with the smoothing analysis results presented in figure 6.1.3. The figure confirms the conclusion following from smoothing analysis, that the rate of convergence is better for a large aspect ratio $\varepsilon = \Delta\xi^1/\Delta\xi^2$ than for a small aspect ratio. The figure also shows that grid refinement in order to enlarge $\Delta\xi^1/\Delta\xi^2$ can pay off in case of a small aspect ratio $\Delta\xi^1/\Delta\xi^2$: the efficiency $N_{it}(\lambda)$ for the cases $\varepsilon = 1., .316, .10, .032$ is 2.5, 3.3, 13.2, 44.9 respectively; defining the total amount of work as the number of iterations times the number of grid points to cover a (given) physical space, the amount of work is proportional to $N_{it}(\lambda)$ multiplied by ε in this experiment, which equals 2.5, 1.04, 1.32, 1.42 for the cases $\varepsilon = 1., .316, .10, .032$. This means that although the grid with $\varepsilon = .032$ contains 10 times fewer grid points than the grid with $\varepsilon = .316$, it is 37% less efficient when the total amount of work for convergence on this grid is considered. This also means that a 10 times higher resolution ($\varepsilon = .316$ instead of $\varepsilon = .032$) can be obtained at lower computational cost.

The multigrid cycle employed in figure 8.2.4 is $V_4(30,3)$, but for the cases $\varepsilon = .10$ and $\varepsilon = .032$ the $V_4(60,6)$ and the $V_4(120,12)$ cycle have been chosen, because smoothing analysis indicated that smoothing would be comparatively inefficient for these cases.

Influence of grid skewness on rate of convergence

Figure 8.2.5 shows the convergence history of the MG-ILU/SIP method on a grid with equal mesh sizes $\Delta\xi^i = 1$ at different sweep angles θ_1 with $\theta_2 = 90°$ and $\theta_3 = 0°$. This experiment is to be compared with the

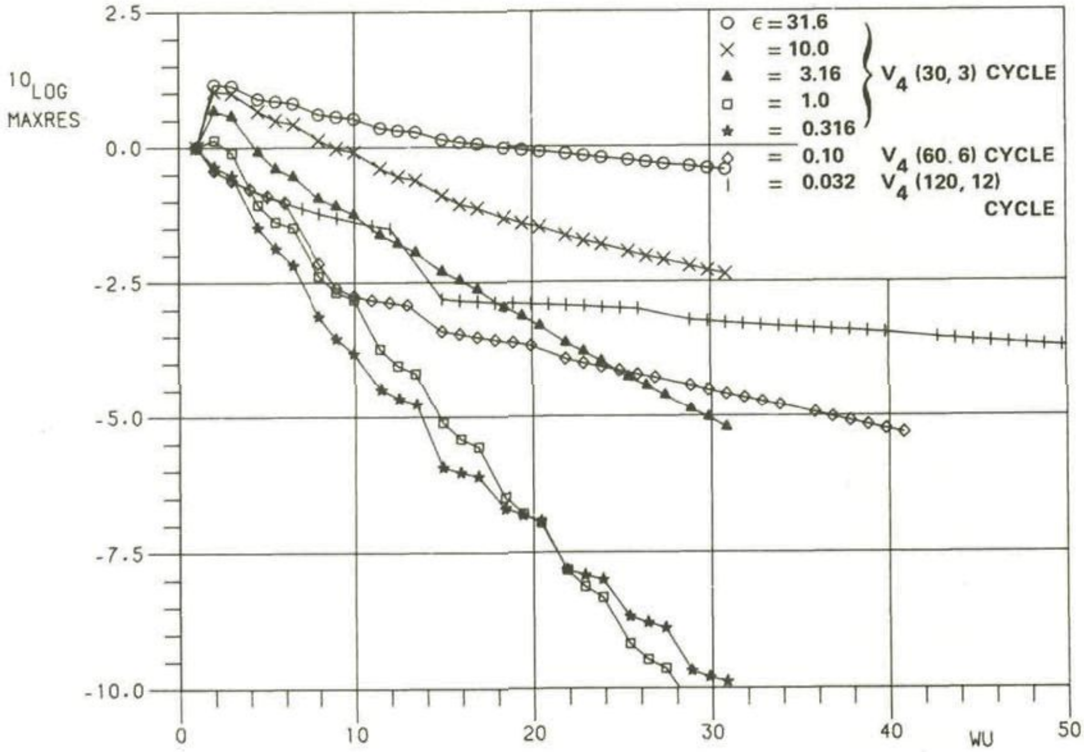


Fig. 8.2.4 MG-ILU/SIP convergence history for various values of $\varepsilon = \Delta\xi^1/\Delta\xi^2$ with $\Delta\xi^1 = \Delta\xi^3$ on a rectangular grid

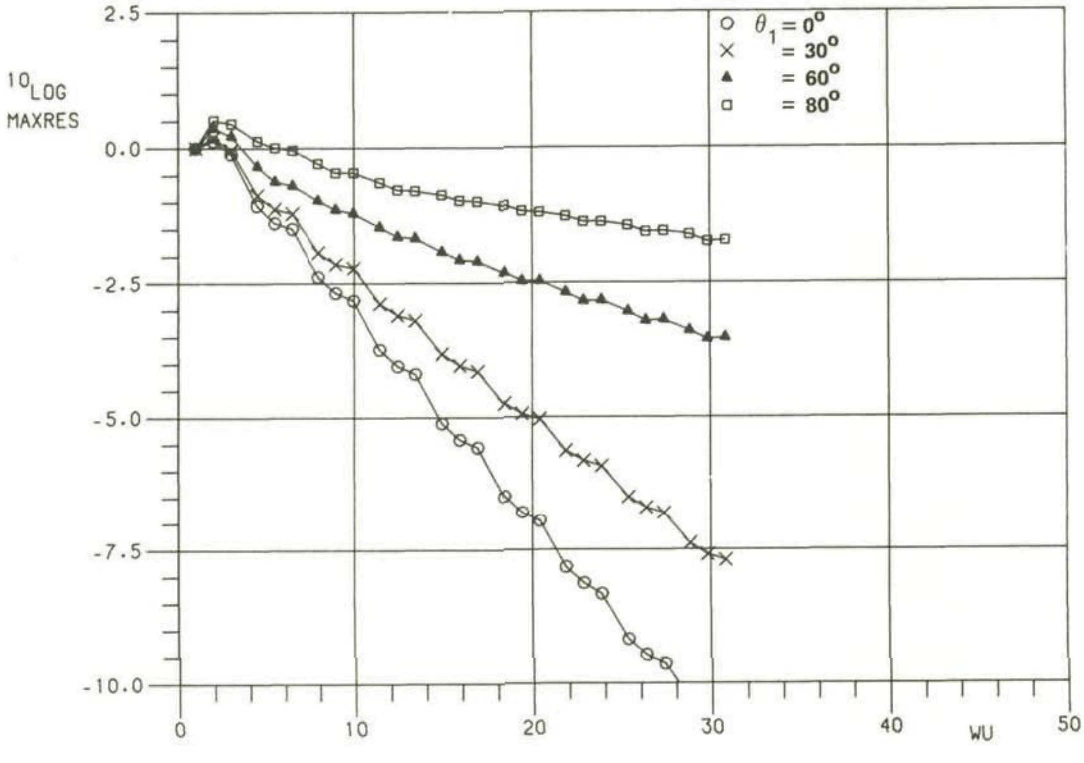


Fig. 8.2.5 MG-ILU/SIP convergence history on a grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$ at various sweep angles θ_1; $\theta_2 = 90^\circ$, $\theta_3 = 0^\circ$

smoothing analysis results presented in figure 6.1.5a. The multigrid cycle used is $V_4(30,3)$.

The figure confirms the conclusion following from figure 6.1.5c that convergence does not suffer too much from low degrees of grid skewness: the rate of convergence is comparable for $\theta_1 = 0°$ and $\theta_1 = 30°$. At larger sweep angles (say, $\theta_1 > 45°$) convergence slows down considerably, as predicted by smoothing analysis, cf. figure 6.1.5a.

Figure 8.2.6 shows the convergence history of the MG-ILU/SIP method on a grid with equal mesh sizes $\Delta\xi^i = 1$ at various skewness angles θ_2, with the sweep angle $\theta_1 = 30°$ and the grid rotation angle $\theta_3 = 0°$. This figure can be compared with the smoothing analysis results presented in figure 6.1.6a. The multigrid cycle used is $V_4(30,3)$.

As in the previous figure, figure 8.2.6 shows that convergence does not suffer too much from low degrees of grid skewness. At higher grid skewness (say at larger deviations from orthogonality than 45°) convergence slows down considerably, as predicted by smoothing analysis, cf. figure 6.1.6a.

Figure 8.2.7 shows the convergence history of the MG-ILU/SIP method on a grid with equal mesh sizes $\Delta\xi^i = 1$ at various grid rotation angles θ_3, with the wing sweep angle $\theta_1 = 30°$ and the skewness angle $\theta_2 = 45°$. This figure can be compared with the smoothing analysis results presented in figure 6.1.6a. The multigrid cycle used is $V_4(30,3)$.

Convergence is indeed slower for grid rotation angle $\theta_3 = 90°$ than for $\theta_3 = 0°$ and $\theta_3 = 45°$, as predicted by smoothing analysis, cf. figure 6.1.6a.

Figure 8.2.8 shows the convergence history of the MG-ILU/SIP method on a grid with equal mesh sizes $\Delta\xi^i = 1$ at various grid rotation angles θ_3 while the wing sweep angle $\theta_1 = 60°$ and the grid skewness angle $\theta_2 = 45°$. The multigrid cycle is $V_4(60,6)$. The smoothing analysis results presented in figure 6.1.7a lead one to expect divergence for this case for some rotation angles θ_3 unequal to zero. Indeed, bad convergence is observed for $\theta_3 = 90°$.

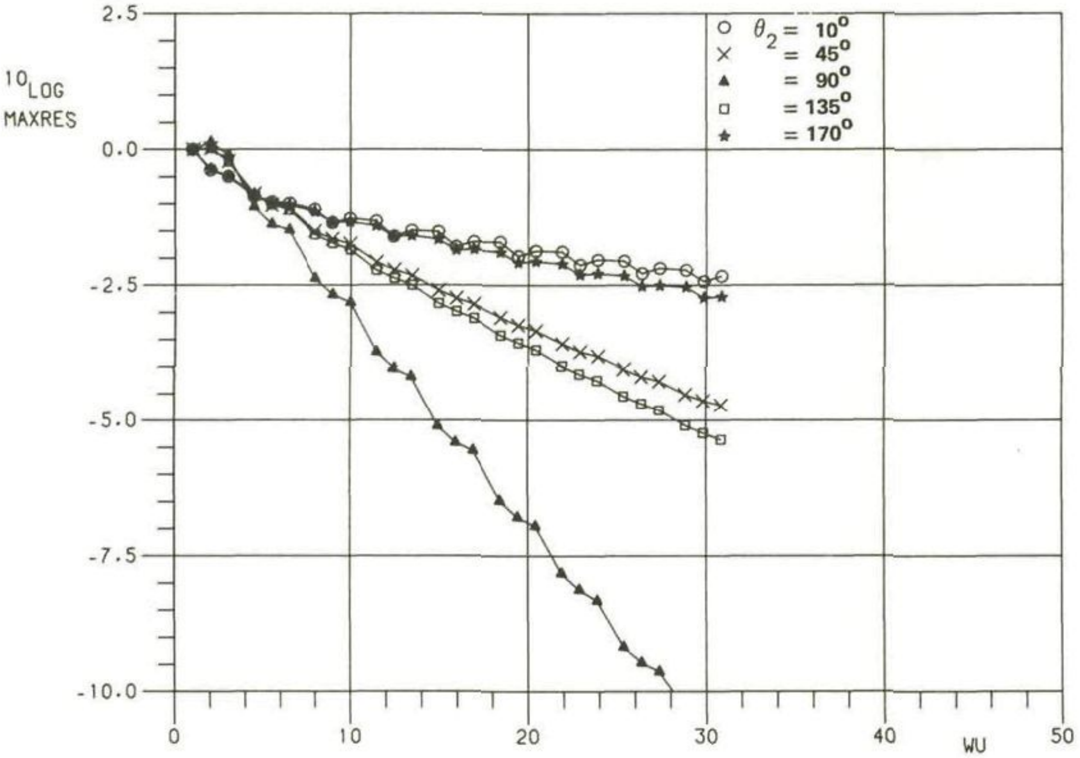


Fig. 8.2.6 MG-ILU/SIP convergence history on a grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$ and wing sweep $\theta_1 = 30^o$ at various grid skewness angles θ_2; $\theta_3 = 0^o$

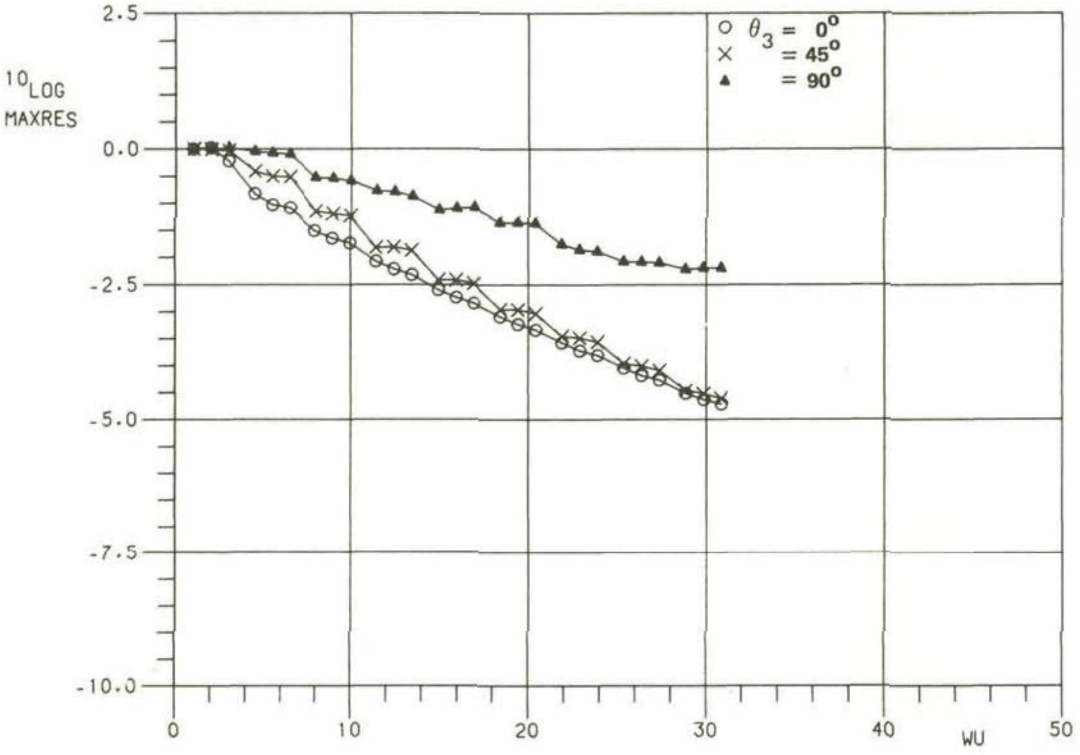


Fig. 8.2.7 MG-ILU/SIP convergence history on a grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$, wing sweep $\theta_1 = 30^o$ and grid skewness angle $\theta_2 = 45^o$ at various grid rotation angles θ_3

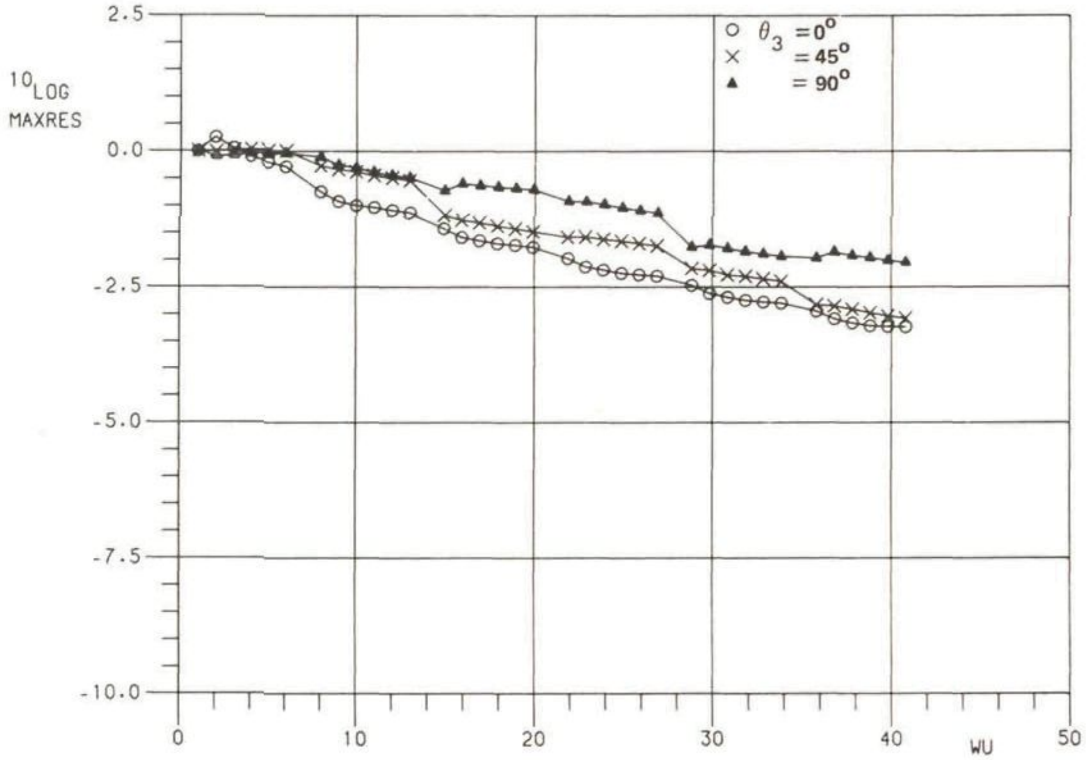


Fig. 8.2.8 MG-ILU/SIP convergence history on a grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$, wing sweep $\theta_1 = 60^o$ and grid skewness angle $\theta_2 = 45^o$ at various grid rotation angles θ_3

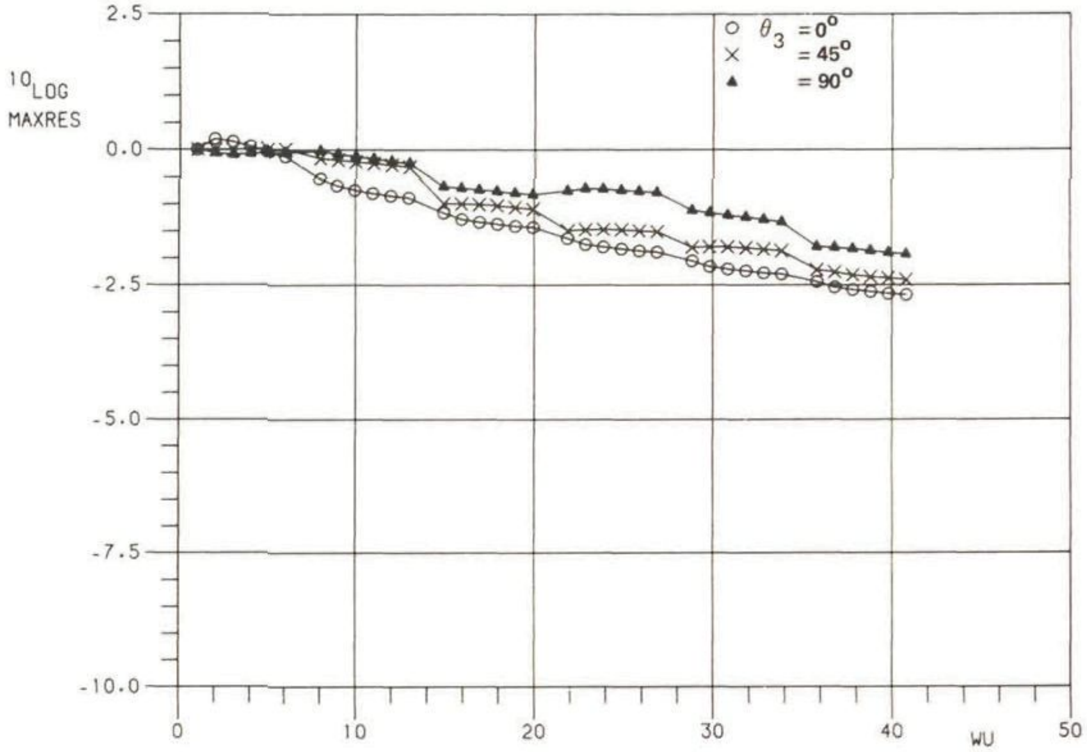


Fig. 8.2.9 MG-ILU/SIP convergence history applying underrelaxation with $\omega = .7$ on a grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$, wing sweep $\theta_1 = 60^o$ and grid skewness angle $\theta_2 = 45^o$ at various grid rotation angles θ_3

The smoothing analysis results in section 6.1, lead one to expect that better convergence may be obtained if underrelaxation with $\omega = .70$ is applied. Figure 8.2.9 confirms this: the convergence history for $\theta_3 = 90°$ has become more regular. The improvement is however small, mainly because the convergence history presented in figure 8.2.8 is not as bad as would be expected from the smoothing analysis. This is explained as follows. In the case $\theta_3 = 90°$, presented in figure 8.2.8 (no underrelaxation), ILU/SIP has the smoothing property (smoothing number $\bar{\rho}<1$) but is unstable for a low-frequency component of the error, namely an error which has a wavelength of approximately 8 meshes in all three coordinate directions. In the case $\theta_3 = 90°$ presented in figure 8.2.9 (with underrelaxation) ILU/SIP has been stabilized by applying underrelaxation, but has also become a less efficient smoothing algorithm because of that underrelaxation. Consequently, the convergence for $\theta_3 = 90°$ presented in figure 8.2.8 is to some degree a matter of luck (the multigrid cycle is apparently well chosen), while in figure 8.2.9 the convergence behaviour corresponds with conclusions obtained from the smoothing analysis.

Influence of grid stretching on convergence

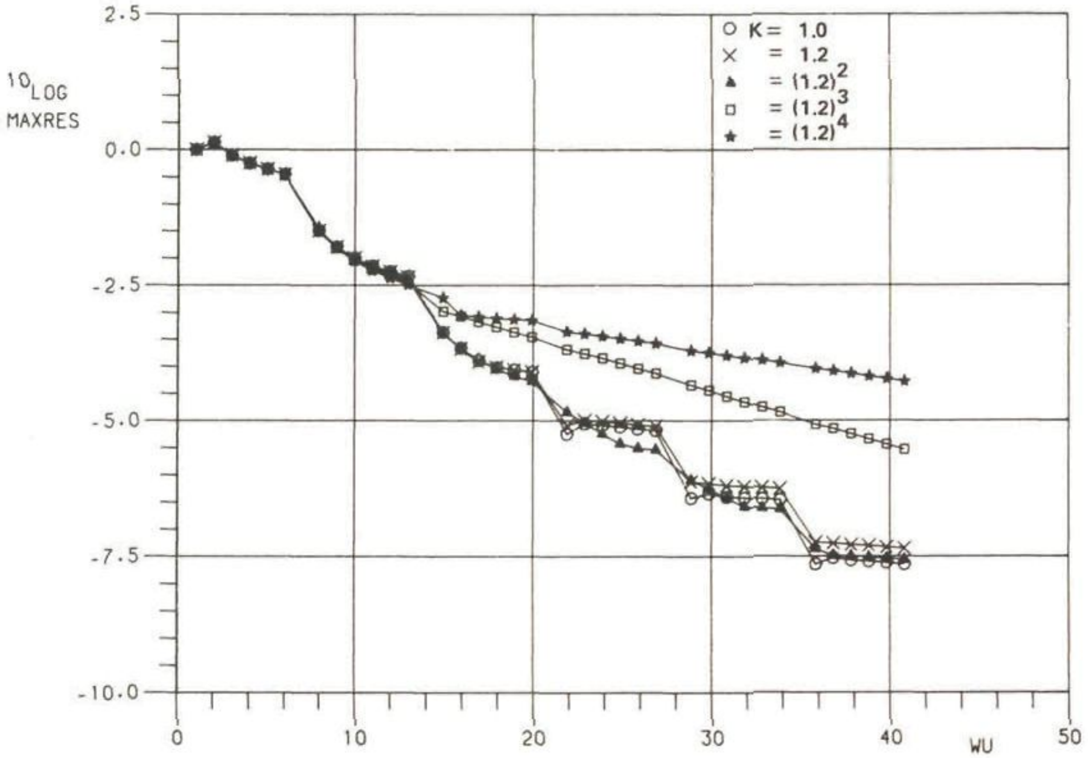
Grid stretching (successive meshes having unequal mesh sizes) cannot be incorporated in the local mode (smoothing) analysis presented in section 6.1, because the assumption of constant matrix coefficients is violated. Therefore the influence of grid stretching is investigated experimentally here.

Consider a rectangular grid with $\Delta\xi^i = 1$, $\theta_1 = 0°$, $\theta_2 = 90°$, $\theta_3 = 0°$ containing 32*32*32 cells in which the 8 central cells in the ξ^2-direction are stretched by a factor K as follows:

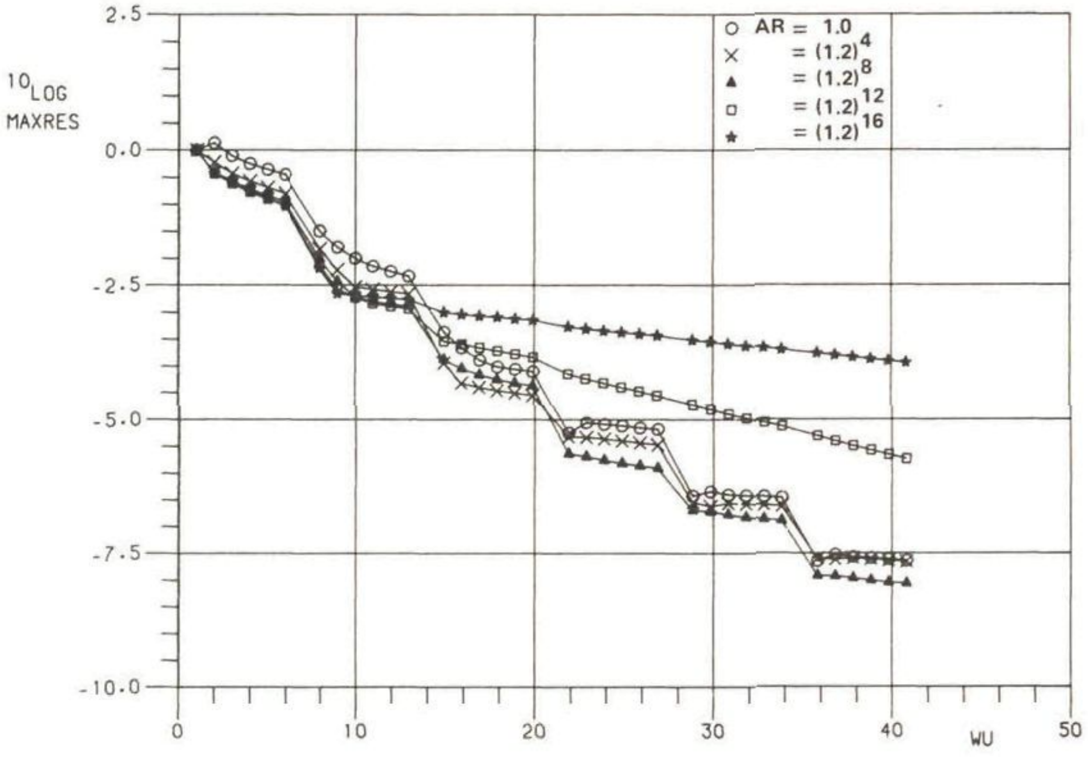
cell no.	1-12	13	14	15	16	17	18	19	20	21-32
size $\Delta\xi^2$	1.	K	K^2	K^3	K^4	K^4	K^3	K^2	K	1.

The grid then locally has an aspect ratio $\Delta\xi^2/\Delta\xi^1 = K^4$, but in the major part of the grid this aspect ratio is 1. The value of K will be $(1.2)^N$, $N = 0,1,2,3,4$.

The convergence history on this stretched grid will be compared with that on a grid having an aspect ratio $\Delta\xi^2/\Delta\xi^1 = (K^4)^N$ everywhere, $N = 0,1,2,3,4$. Figure 8.2.10 shows these convergence histories,



a) Locally stretched grid



b) Grid with constant aspect ratio

Fig. 8.2.10 MG-ILU/SIP convergence history on a locally stretched grid and a uniform grid with comparable aspect ratio

employing a $V_4(60,6)$-cycle. The figure clearly shows that on a grid having only locally an aspect ratio of $(K^4)^\alpha$ multigrid converges about equally fast as a grid having globally an aspect ratio of $(K^4)^\alpha$. Consequently, the largest local aspect ratio ("worst" aspect ratio) of the grid strongly influences the ultimate rate of convergence. This conclusion also holds in more general cases, as will be shown in the next sections.

Discussion of convergence results

The convergence results presented so far compare well in a qualitative sense with the smoothing analysis presented in section 6.1: all major findings of that section have been confirmed by the numerical experiments just described. However, close examination reveals differences between the rate of convergence as analytically predicted by smoothing analysis (section 6.1) and the actual rate of convergence observed in the numerical experiments of this section. These differences are to be attributed to violation of the assumptions that are made when using the smoothing analysis as a prediction for the actual rate of convergence of the multigrid method. These assumptions are:

1. The convergence speed does not deteriorate at boundaries, which are assumed to be absent in smoothing analysis.
2. The coarse grid correction step gives an adequate reduction of the low-frequency modes of the error and does not amplify the high-frequency modes of the error. Theoretically this condition can be satisfied by a suitable choice of the restriction, coarse grid and prolongation operator.
3. The numerical evaluation of the smoothing number $\bar{\rho}$ is sufficiently accurate.

Under the above assumptions, one can compare the smoothing number $\bar{\rho}$, obtained from smoothing analysis, with the error reduction per work unit λ, observed in the numerical experiment. In case the aforementioned assumptions are satisfied, these numbers must be about equal. In this comparison the number λ must be measured while not counting the work that is done on the coarse grids in the experiment, so that one compares the numbers $\bar{\rho}$ and λ on the basis of one smoothing step on the finest grid.

In all experiments presented so far, two phases can be distinguished in the convergence history, viz. an initial and an asymptotic convergence phase. Usually the convergence speed is slowest in the asymptotic convergence phase, because it is then dominated by the error mode for which the method is converging most slowly. In nearly all numerical experiments presented in this subsection, the asymptotic convergence phase is dominated by the smoothing of errors near boundaries of the computational domain (the experiment on the rectangular grid with $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$ (figure 8.2.3) being the only exception).

Figure 8.2.11 gives $N_{it}(\lambda)$ *(corrected for the work on the coarse* grids) versus $N_{it}(\bar{\rho})$ in the asymptotic convergence phase for the experiments given in figure 8.2.4, which appears to be reasonably characteristic for all experiments presented.

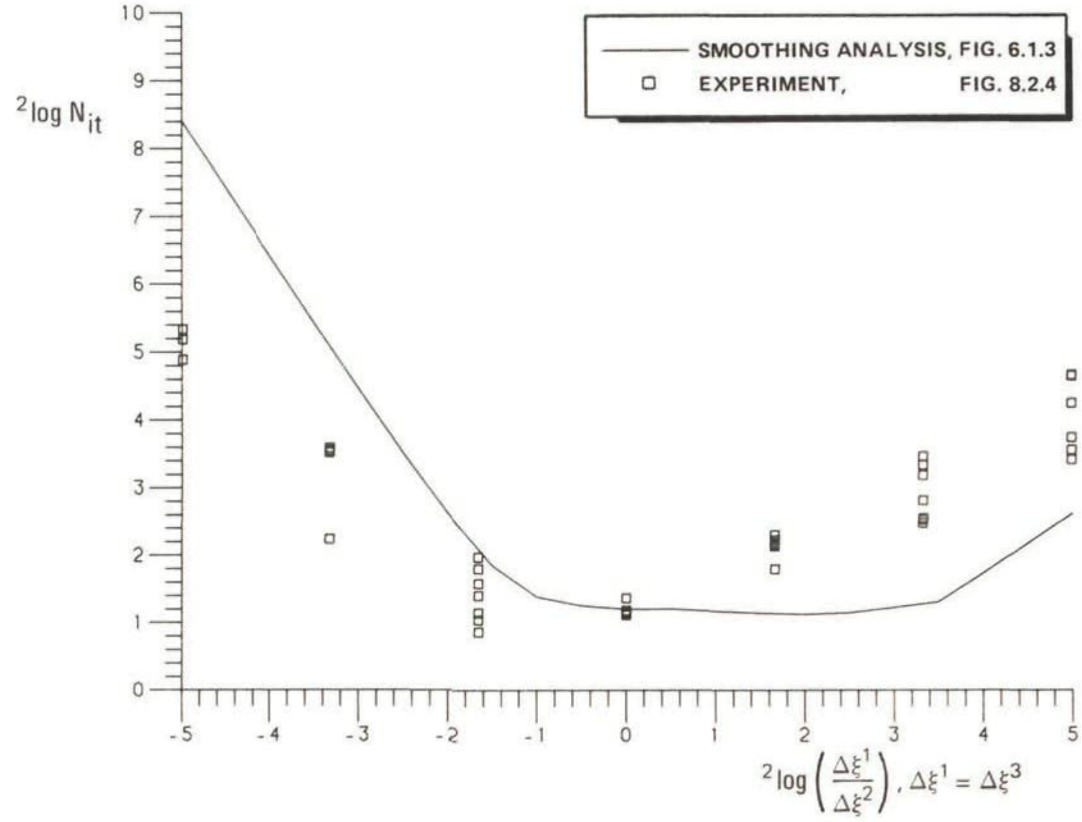


Fig. 8.2.11 Smoothing efficiency $N_{it}(\bar{\rho})$ compared with the asymptotic (corrected) error reduction efficiency per work unit $N_{it}(\lambda)$ for the experiment presented in figure 8.2.4

Figure 8.2.11 clearly shows that for $\varepsilon = \Delta\xi^1/\Delta\xi^2 > 1$ the convergence is slower than is to be expected on the basis of smoothing analysis. It was found that the method is slower near the boundaries with Neumann boundary conditions than in the interior. Therefore, the first assumption in this comparison (namely that the convergence speed does not deteriorate at boundaries) is not satisfied. Because in practice wing grids do not have the property $\varepsilon = \Delta\xi^1/\Delta\xi^2 > 1$ at most grid points with prescribed Neumann boundary conditions, we have not tried to improve the convergence for this case by using special boundary relaxation procedures.

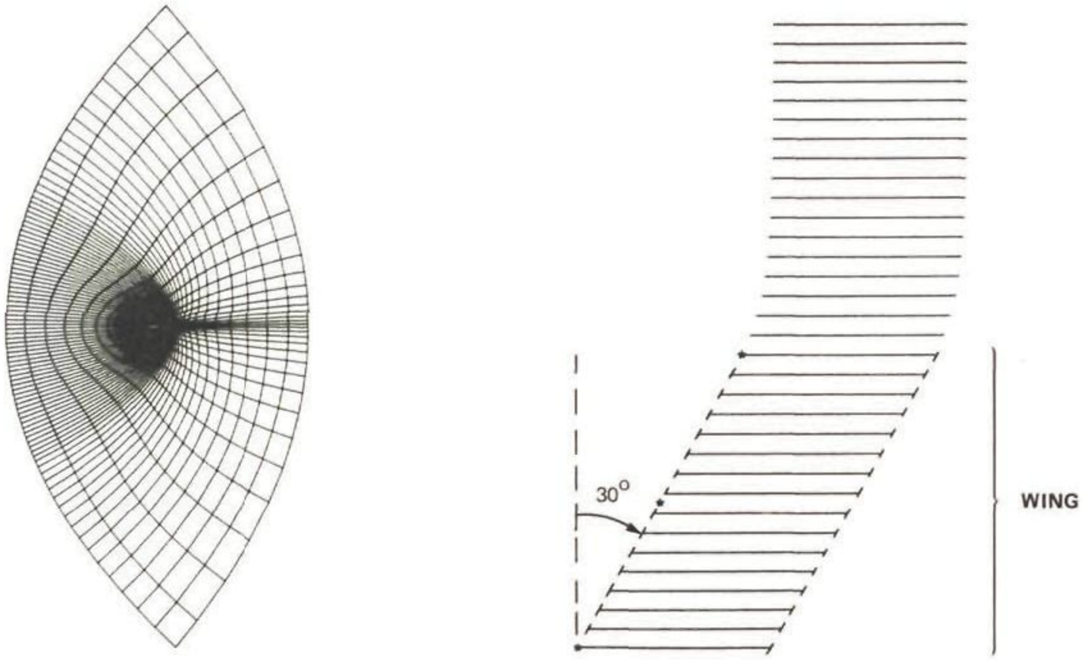
The case $\varepsilon = \Delta\xi^1/\Delta\xi^2 = 1$ converges approximately as predicted by smoothing analysis.

The case $\varepsilon = \Delta\xi^1/\Delta\xi^2 < 1$ converges faster than predicted by smoothing analysis. As the smoothing number $\bar{\rho}$ only gives a lower bound for the rate of convergence, a better rate of convergence, as encountered in this case, is possible of course.

The final conclusion of the exercise undertaken in this subsection is, that the smoothing analysis does not always provide an accurate prediction of the convergence rate measured in numerical experiments, because the rate of convergence rate is different at boundaries than in the interior of the computational domain.

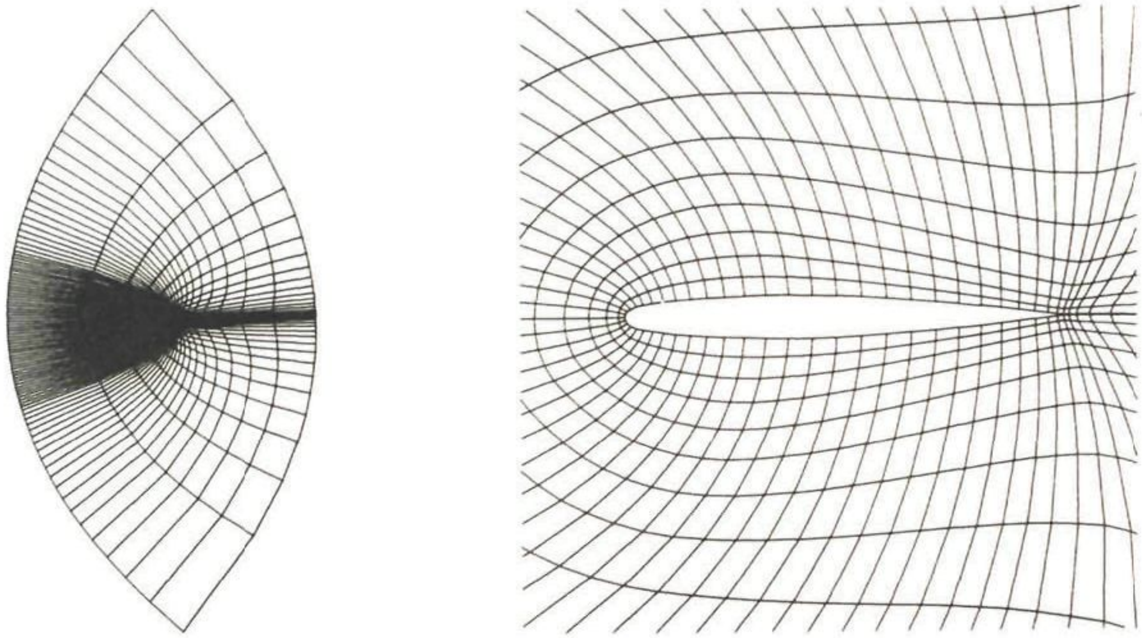
8.2.2 Simplified wing in transonic flow

Another set of experiments is presented that resemble realistic flows more closely. Again, comparison will be made with smoothing analysis presented in section 6.1. The transonic flow about a simple swept back wing of constant chord and profile (figure 8.2.12) at freestream Mach number $M_\infty = .84$ and angle of attack $\alpha = 3°$ is computed. The wing profile is an ONERA-D profile. The grid contains 88*16*16 cells in the circumferential, wing-normal and spanwise directions. There are 56*7 cells adjacent to the wing. In grid planes perpendicular to the wing, two different kinds of grids have been generated. Grid A (figure 12a) has the property $\Delta\xi^1 \cong \Delta\xi^3$ to a reasonable extent, while grid B (figure 12b) does not have this property at all and instead has computational cells with mesh sizes of three different orders of magnitude. Near the wing, both grids are approximately the same (figure 12d). In both cases, the grid spacing in the spanwise direction is uniform. Grid refinement in the spanwise direction will be investigated by doubling the number of cells in that direction. These refined grids will be referred to as AA and BB. The multigrid cycle used is $W_3(12,6)$. The full multigrid cycle is not used in this experiment.



a) Grid plane A perpendicular to wing ($\xi^1\xi^3$–plane)

c) Grid plane distribution in spanwise direction, refined grid (ξ^2–direction)



b) Grid plane B perpendicular to wing

d) Detail of grid plane near the wing

Fig. 8.2.12 Computational grid used for transonic flow about swept wing

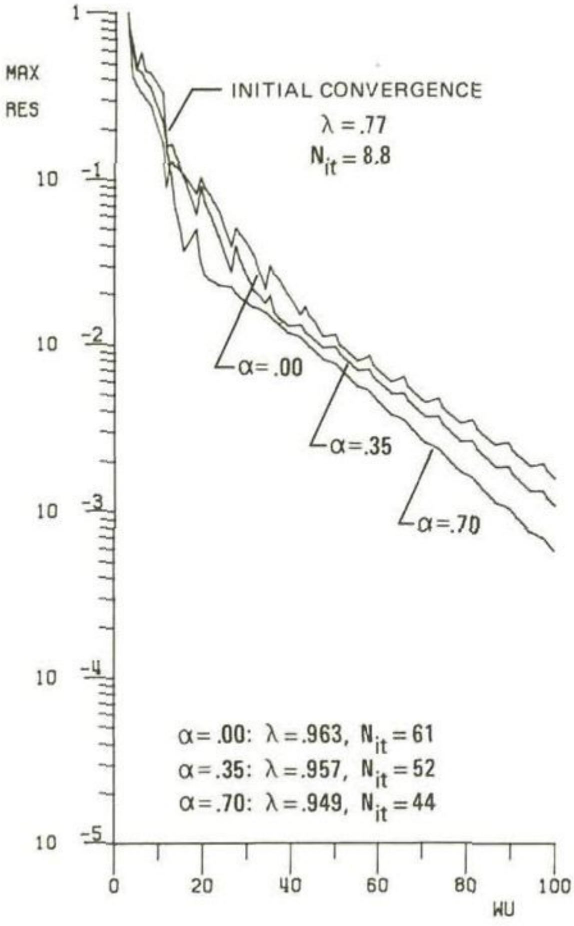


Fig. 8.2.13 Convergence history of MG-ILU/SIP(α) for several values of α on grid A

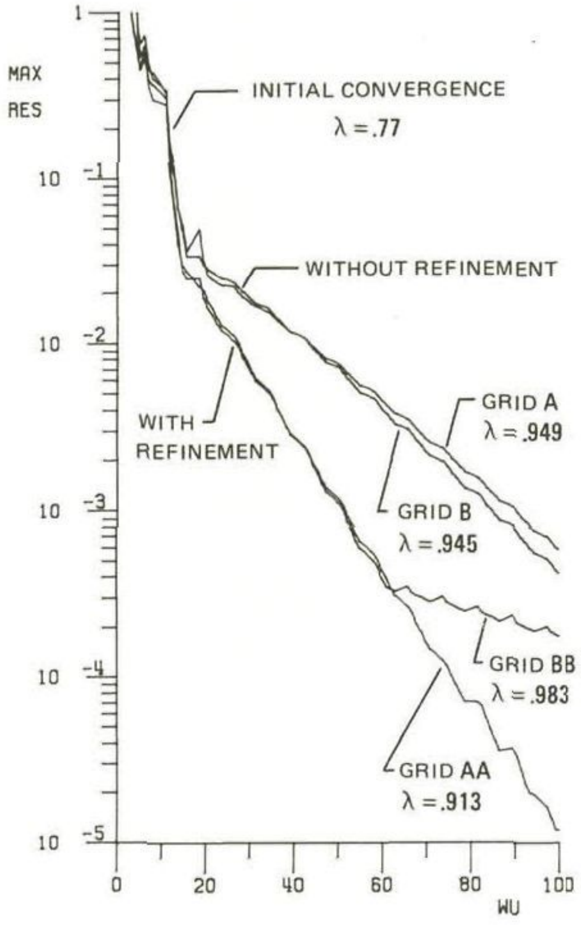


Fig. 8.2.14 Convergence history of MG-ILU/SIP(.7) on grids A,B and grids AA,BB (with spanwise refinement)

Figure 8.2.13 shows the convergence history of the MG-ILU/SIP(α) method on grid A for several values of α in the elliptic (subsonic) region. Two convergence phases can be distinguished, viz. initial convergence and asymptotic convergence. Initial convergence is usually fast and establishes the global characteristics of the flow. The asymptotic convergence, however, is in most cases much slower, because it is dominated by the cells which have the "worst" combination of mesh sizes in view of the smoothing analysis presented in section 6.1. The convergence level obtained after the initial phase, usually about 1 - 1.5 orders of magnitude reduction in the residual, is in many cases sufficient for engineering applications. Figure 8.2.13 shows, that asymptotic convergence is best for $\alpha = .70$; for $\alpha = .35$ and $\alpha = 0$ respectively the method is asymptotically 18% and 39% slower. This result also follows from the smoothing analysis presented in section 6.1. The value $\alpha = 1$ is generally not allowed in the elliptic region because of an unreliable convergence (see also section 6.2).

Figure 8.2.14 shows the effects of doubling the number of cells in the spanwise direction. The convergence history on grids A and B is about the same. In both cases, the maximum residual is located at the trailing edge of the tip section, where the grids are similar. The refined grids AA and BB are obtained from grids A and B respectively by doubling the number of cells in the spanwise direction. The convergence on grid AA is nearly twice as fast as on grid A. Such improved convergence was already predicted in section 6.1, figure 6.1.3. The result implies that doubling the resolution in the spanwise direction does not lead to an increase in computation time. The convergence on grid BB initially shows the same improvement as grid AA. However, at 60 work units there occurs a sudden slowdown of the asymptotic convergence rate. This happens when the maximum residual, which was originally at the wing trailing edge, jumps to the branch cut in the far field plane downstream. Here the cells have mesh sizes of three different orders of magnitude and consequently, the convergence slows down considerably, as already predicted in section 6.1.

The following lesson can be learned from figure 8.2.14. Due to choosing the freestream flow as a starting solution, the error will be highly non-uniform at the start: the initial error will be large at the wing, where the freestream flow is not a good guess for the final solution; the initial error will be very small in the far field, where the freestream flow is hardly disturbed by the presence of the wing. Because the initial error is so small in the far field, slow convergence in the far field grid shows up only in the asymptotic convergence phase. As a two to three orders of magnitude error reduction is more than sufficient for most practical applications, one can benefit from this observation as follows. The grid should be generated such, that the solution method *will be fast in regions where the freestream flow is not a good guess* for the final solution. In other regions one can afford to generate a grid for which the method will not give fast convergence. As a consequence, however, one accepts a slow asymptotic convergence phase and therefore very accurate (highly converged) results will be hard to obtain.

8.3 Convergence of solution method for realistic wings in transonic flow

8.3.1 General remarks

Numerical experiments on transonic flows around realistic wings indicated that the solution process is easily trapped in a nonlinear divergence in the wing tip region. A close examination of the problem revealed that this divergence was caused by the circulation update, equation (4.5.6), outside the tip. Whenever a nonzero circulation (lift) value was generated outside the wing tip on the wing extension, the relaxation of this circulation (setting it to the correct zero value) generated a small spurious region of highly supersonic flow outside the wing tip, which the solution process could not eliminate (probably because switching back from supersonic to subsonic flow is a highly nonlinear process). The nonlinear divergence phenomenon only showed up after a coarse grid correction, because then the flow corrections on a grid are the largest. This difficulty has been overcome by requiring that the first grid plane outside the wing tip should be present on several levels of the multigrid process. Also it is required to use a linear prolongation operator in the wing tip region. This way it is rigorously avoided that a section of the wing extension gets a nonzero circulation value after a coarse grid correction, which was already identified as the cause of the difficulty. For simplicity, linear *prolongation has been used for the whole computational domain.*

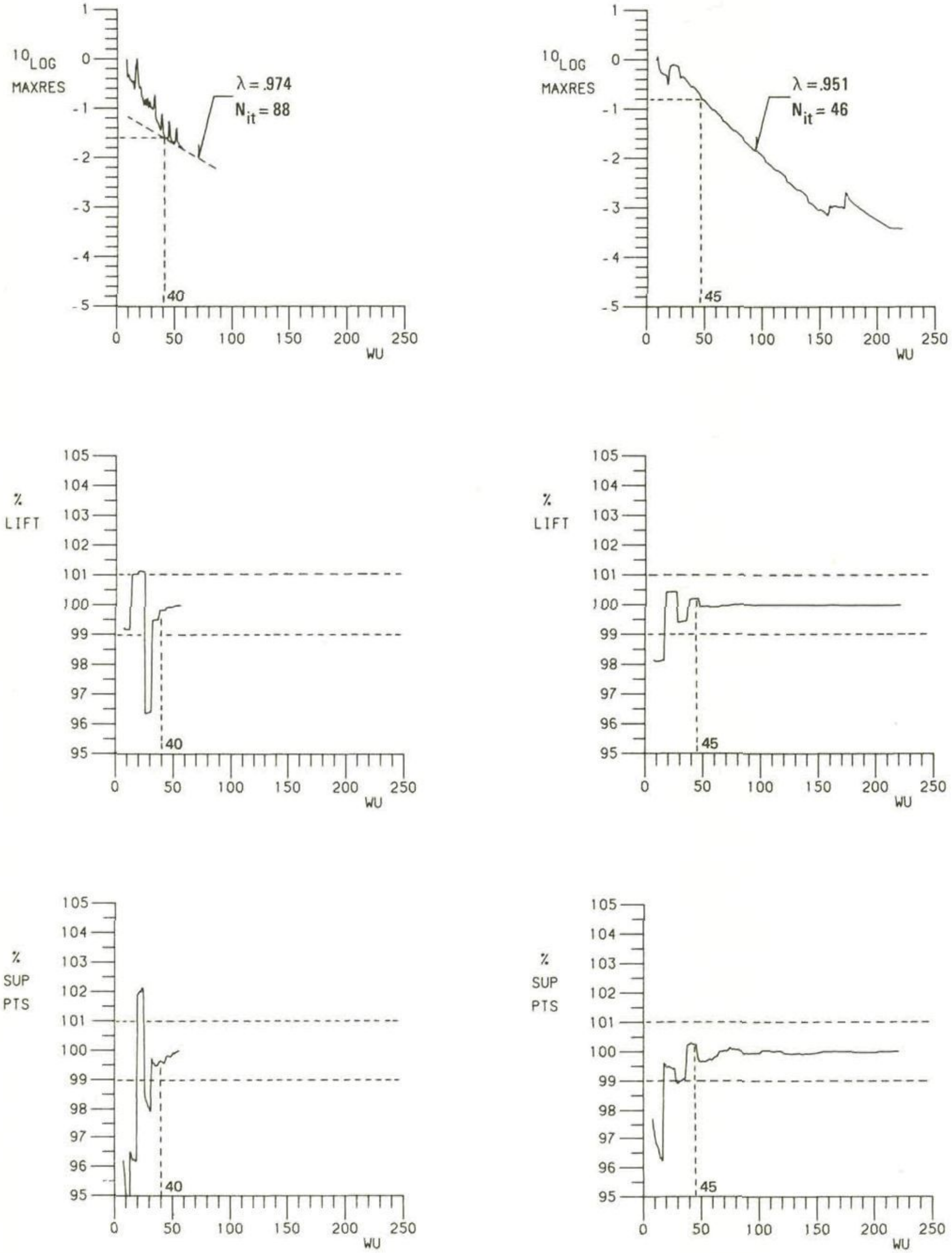
Numerical experiments also indicated that weighted restriction instead of injection of potentials may not be robust for wing flows. In case of weighted restriction, spanwise grid sections on the coarse grid have a different trailing edge potential jump value Γ on the coarse and on the fine grid during restriction. In the experiments this caused non-convergence phenomena for the potential jump Γ on the finest grid, which did not occur when potential injection was used.

8.3.2 Convergence acceleration by local smoothings

In the section 8.2 it has been shown that the asymptotic convergence phase of the multigrid process is dominated by the smoothing in cells that have the "worst" combination of mesh sizes. Applying extra local smoothings in these cells can improve the convergence significantly.

In the wing flow problems considered in this thesis some difficult cells are always located near the wing leading or trailing edge, where the meshes are very fine in wing-normal and chordwise direction and comparatively large in the spanwise direction. The worst aspect ratios can be found on the wing extension, where the grid is stretched towards the far field boundary (figure 3.1.6). It therefore suffices to apply local relaxations only in the first fourth part (i.e. $K \in [0, NK/4]$) of the cells normal to the wing, the wing extension and the branch cut. During these local relaxations the potential is kept fixed on the outer plane $K = NK/4$, while on all other boundaries the usual boundary conditions are applied. An advantage of this approach is that this local smoothing highly resembles the global smoothing employed.

The effect of the local smoothings on the convergence of the MG-ILU/SIP method will be shown by computing the transonic flow about the DFVLR-F4 wing at Mach number $M_\infty = .75$ and angle of attack $\alpha = .84$ degrees. This wing and flow condition are representative for the supercritical wings of today and this flow case has also been studied by a GARTEUR action group (Carr [1]). The grid, shown in figure 3.1.6, involves 176*32*32 (180224) cells and the multigrid method employs 4 grids. In figure 8.3.1a a $W_4(12,6)$-strategy without local smoothings is chosen, while in figure 8.3.1b a $V_4(12,6)$-strategy with 2 local smoothings per global smoothing is chosen. Consequently, in the latter case one smoothing relaxation corresponds to $1 + 2*\frac{1}{4} = 1.5$ work units. The flow solution computed in this experiment will be presented in subsection 8.4.2.



a) Without local smoothings

b) With local smoothings

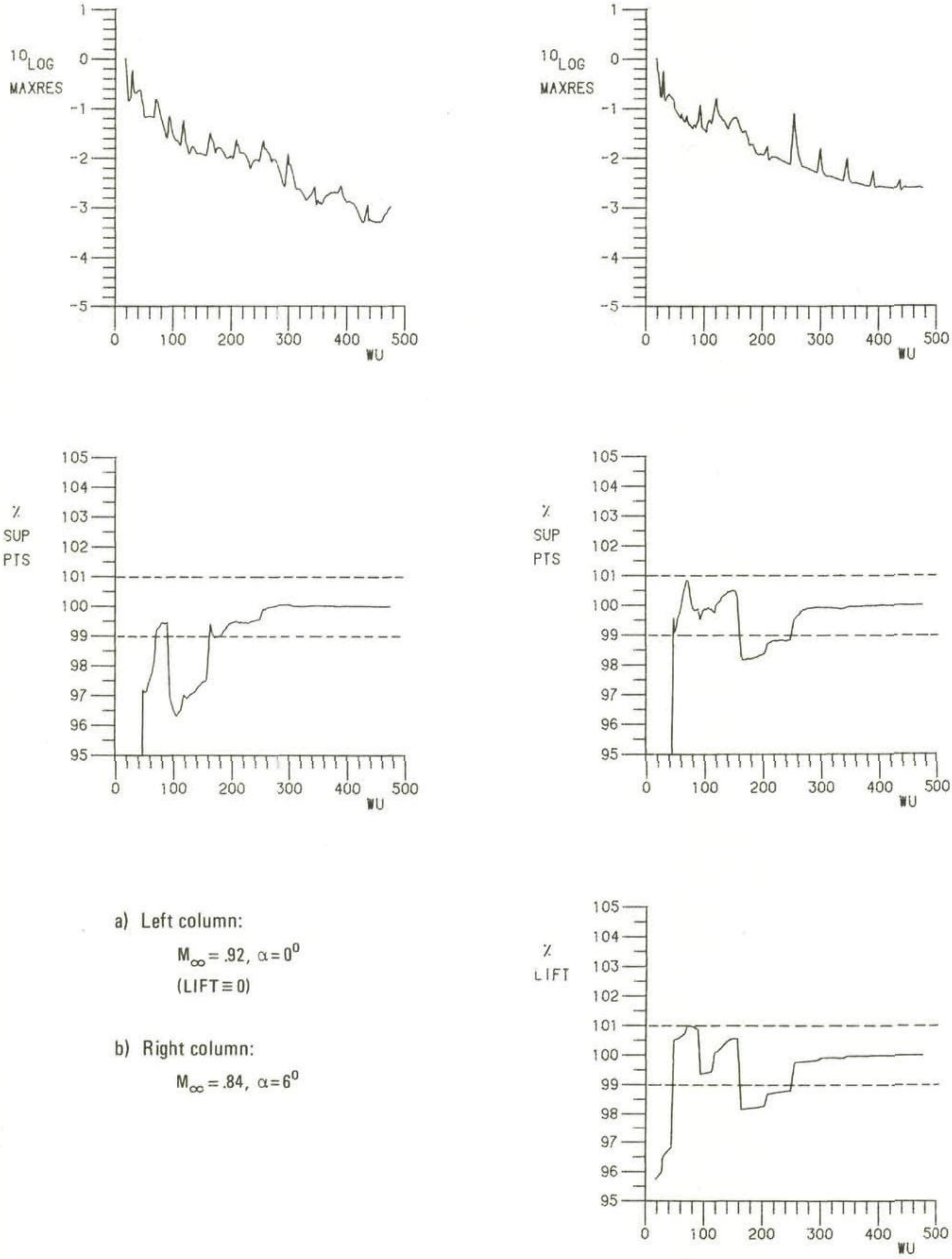
Fig. 8.3.1 Comparison of MG-ILU/SIP convergence with and without local smoothings for DFVLR-F4 wing at M_∞= .75, α = .84 on 176·32·32 grid

Figure 8.3.1 shows that the convergence history becomes much "smoother" when local smoothings are applied, and consequently convergence is more reliable. This smoother convergence is due to the extra smoothing of the shocks, as a major part of the supersonic zone and shocks is located in the local smoothing region. In the initial convergence phase, the method with local smoothings is about equally fast as the method without local smoothings, but in the asymptotic convergence phase the method with local smoothings is nearly two times faster than the method without local smoothings.

In figure 8.3.1b the sudden increase in the maximum residual at 160 WU is probably caused by a shock movement.

In this experiment, the initial convergence phase is dominated by the residual at the extension of the leading edge outside the wing tip. Here $\Delta\xi^2 \approx 17\Delta\xi^1$, $\Delta\xi^1 \approx \Delta\xi^3$, which is indeed a very extreme aspect ratio, considering figure 6.1.3. The final convergence phase is dominated by the cells near the leading edge of the wing extension near the far field plane, where the $\Delta\xi^2/\Delta\xi^1$- and $\Delta\xi^2/\Delta\xi^3$-aspect ratios are well over one hundred. In such cases many smoothings are required on each grid to smooth the error, cf. figure 6.1.3, and the only good remedy here is to take special measures, e.g. to apply extrapolation techniques (see e.g. Van der Wees, van der Vooren and Meelker [1]), to use plane-relaxation as introduced by Thole [1] or to use the multigrid method as a preconditioner in a conjugate gradients algorithm (see e.g. Sonneveld, Wesseling and de Zeeuw [1]).

Figure 8.3.1 also shows the convergence history of the lift (i.e. the sum of Γ for all spanwise grid sections) and the number of supersonic grid points (i.e. the size of the supersonic region). After about 40-45 work units with only .8-1.6 digits reduction in the maximum residual, the lift and the number of supersonic points have converged to well within 1% of their final value, which is sufficient for engineering applications. This good convergence can for a large extent be contributed to the usage of the full multigrid method : the starting solution on the fine grid, which is a prolongated converged coarse grid solution, approximates the final values of the lift and the number of supersonic points already within 5%.



a) Left column:

$M_\infty = .92$, $\alpha = 0^0$

(LIFT ≡ 0)

b) Right column:

$M_\infty = .84$, $\alpha = 6^0$

Fig. 8.3.2 Convergence history of the MG-ILU/SIP method for ONERA-M6 wing on 176·32·32 grid

8.3.3 Convergence of highly transonic flow calculations

In this subsection the MG-ILU/SIP convergence history is given for the calculation of the transonic flow about the ONERA-M6 wing, which is a well-known testcase for transonic flow codes. Each wing section is an ONERA-D profile, which is a symmetric profile. Consequently, at zero angle of attack the flow will be symmetric, with supersonic bubbles both on the upper and the lower side of the wing.

Figure 8.3.2 shows the convergence history of the MG-ILU/SIP method for the transonic flow about the ONERA-M6 wing at $M_\infty = .92$, $\alpha = 0°$ and at $M_\infty = .84$, $\alpha = 6°$. In the first flow case about 10 percent of the grid points in the 176*32*32 grid is supersonic. The flow solutions computed in these experiments will be presented in subsection 8.4.2. Due to the strong shocks that are present in the solution, the multigrid process only converged well if a $V_4(20,10)$-cycle was used, i.e. the number of ILU/SIP smoothings per grid level was increased from six (as used in the previous subsection) to ten. Local smoothings were not applied in this experiment. In the final phase of the solution process shown in figure 8.3.2 a $V_4(40,20)$- and a $V_3(80,40)$-cycle (thus using only three grid levels in the multigrid process) had to be used. Yet the convergence process remains very irregular due to shock movements after each coarse grid correction step.

8.3.4 Concluding remarks on convergence of MG-ILU/SIP method

Two figures are reproduced from Van der Wees [1], showing convergence results for the transonic flow in a windtunnel with a bump on the bottom wall at $M_\infty = .95$. In figure 8.3.3 it is shown that the convergence history of the number of supersonic grid points is better for MG-ILU/SIP than for MG-SLR. The single grid SLOR method is significantly worse than the multigrid methods. Figure 8.3.4 shows a crossplot of the mean residual norm (l_1-norm) versus the number of supersonic grid points.

At about 1.5 orders of magnitude reduction in the residual, MG-ILU/SIP has already nearly reached the final number of supersonic grid points, while MG-SLR is still about 1% away from this final value. The single grid SLOR method is, however, still far from the final value. If the number of supersonic grid points (development of the supersonic zone) is interpreted as a measure for the "quality of the solution", it is obvious from figure 8.3.4 that multigrid methods (MG-SLR and MG-ILU/SIP) provide (at a certain error level) solutions of far better

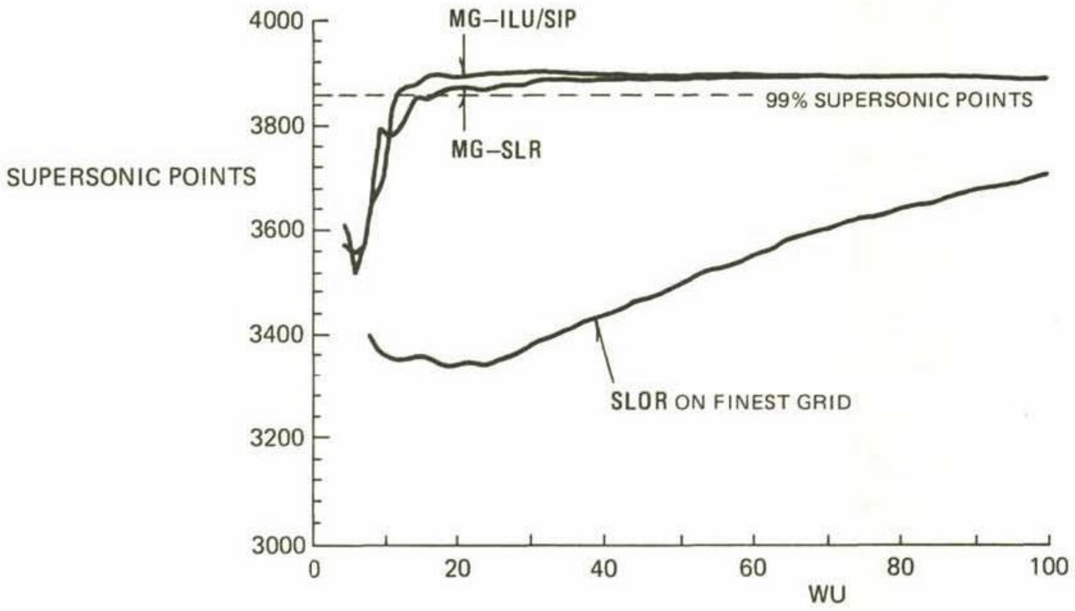


Fig. 8.3.3 Convergence histories of number of supersonic grid points of several algorithms for windtunnel with a bump on the bottom wall at $M_\infty = .95$ on $96 \cdot 24 \cdot 16$ grid; about 11% of the grid points is supersonic

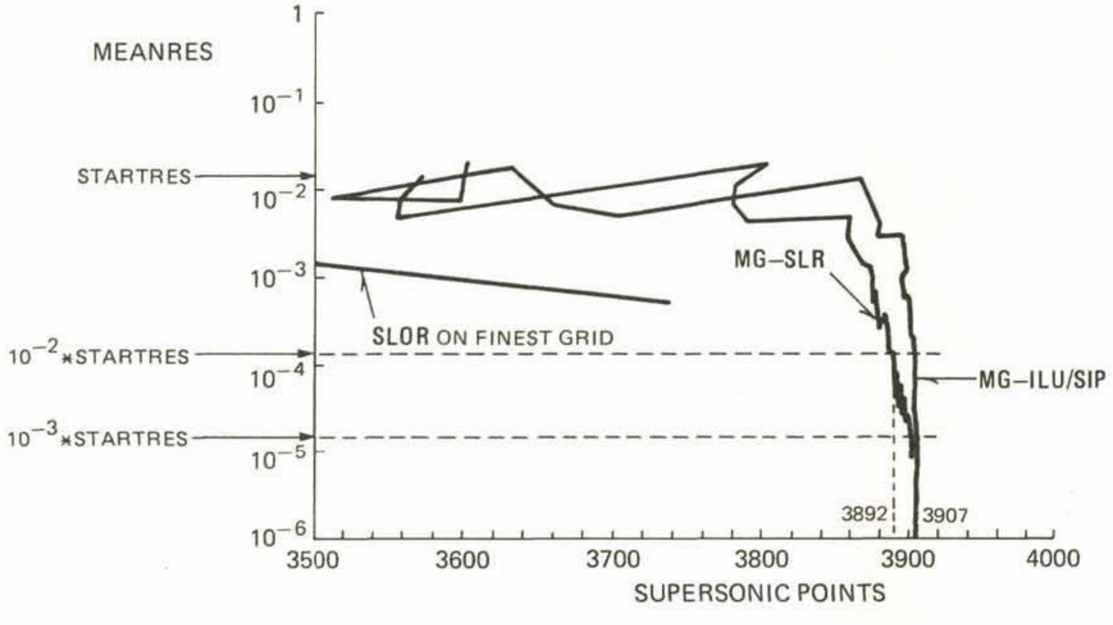


Fig. 8.3.4 Plot of mean residual norm versus the number of supersonic grid points for the experiment of figure 8.3.3; the figure shows the number of supersonic points at a certain error level for several algorithms

quality than the corresponding single grid methods. This is explained by a more efficient approximation of the low-frequency modes of the solution. Moreover, the MG-ILU/SIP method provides a solution which is of better quality than the one provided by the MG-SLR method. This can probably be used to advantage by specifying a lower convergence level.

In conclusion, a remark is made on the efficiency of the multigrid method. It has been found that in the transonic potential flow problems described in this section the multigrid efficiency is not independent of the mesh size h (i.e. the number of equations N). This is in contrast with the considerations given in section 5.1 and with experiences in many other applications (e.g. elliptic problems), where it has been found that the multigrid efficiency is independent of the mesh size h. Indeed, for elliptic problems (subsonic flow) the convergence of the present multigrid method is independent of h, but this property does not hold true for transonic flow. The explanation is probably that the subsonic/supersonic interface, i.e. the sonic surfaces and the shock surfaces, have to be found as part of the solution, that is, can be seen as a free surface between an elliptic and a hyperbolic problem. When the grid is refined ($h \to 0$, $N \to \infty$), this interface has to be computed with more detail and especially at the line where a sonic surface and a shock surface meet it has been observed that the amount of work required to converge this region of the flow increases when h goes to zero (in the iterative process the grid points located in this region tend to oscillate between the subsonic and supersonic states for some time). Once the flow regions around shocks and sonic surfaces have converged, we may perhaps expect that in this final convergence phase the multigrid efficiency is independent of h, but in practice this phase will almost never be reached, see for example the irregular convergence histories of figures 8.3.1 and 8.3.2.

8.4 Computational results

8.4.1 Results of finite volume method

Figure 8.4.1 shows the accuracy of the finite volume method for the incompressible flow ($M_\infty = 0$) about a thin (RAE) wing with 5% thick NACA four digit series airfoil. The angle of attack is $\alpha = 5°$. This wing has been used as a testcase to compare several existing panel methods (Sytsma, Hewitt and Rubbert [1]). The grid used involves 176*32*32 cells and resembles the grid used in section 8.3.

The results are compared with results of the higher order panel method of Roberts, which emerged from the comparisons in Sytsma, Hewitt and Rubbert [1] as being the most accurate. The agreement is excellent. Only minor deviations occur at the trailing edge and close to the tip. These deviations are believed to be caused by the (small) differences in the implementation of the Kutta condition, the different spanwise distributions of the wing surface grid, and the different treatment of the wing tip. Which of the two results of figure 8.4.1 is the most accurate is unknown.

An important feature of the finite volume (FV) discretization is its freestream consistency; this feature ensures that the freestream identically satisfies the finite volume scheme in the far field. Figure 8.4.2 illustrates the importance of this feature; in a transonic flow calculation for the DFVLR-F4 wing (see section 8.3.2) the chordwise position of the shock system on the wing upper surface is predicted 3 - 5% chord length too far upstream if the freestream consistency term in the finite volume scheme is switched off.

An illustration of the results of the mass flux-vector splitting scheme used is given in figure 8.4.3. This figure shows that the scheme resolves strong (supersonic to subsonic) shocks over, say, three mesh widths (there are about two points in the shock), also in case the shock is not aligned with the grid on the wing. Weak (supersonic to supersonic) shocks are however smeared over, say, six meshwidths, because there are no facilities in the scheme to detect and capture these weak shocks.

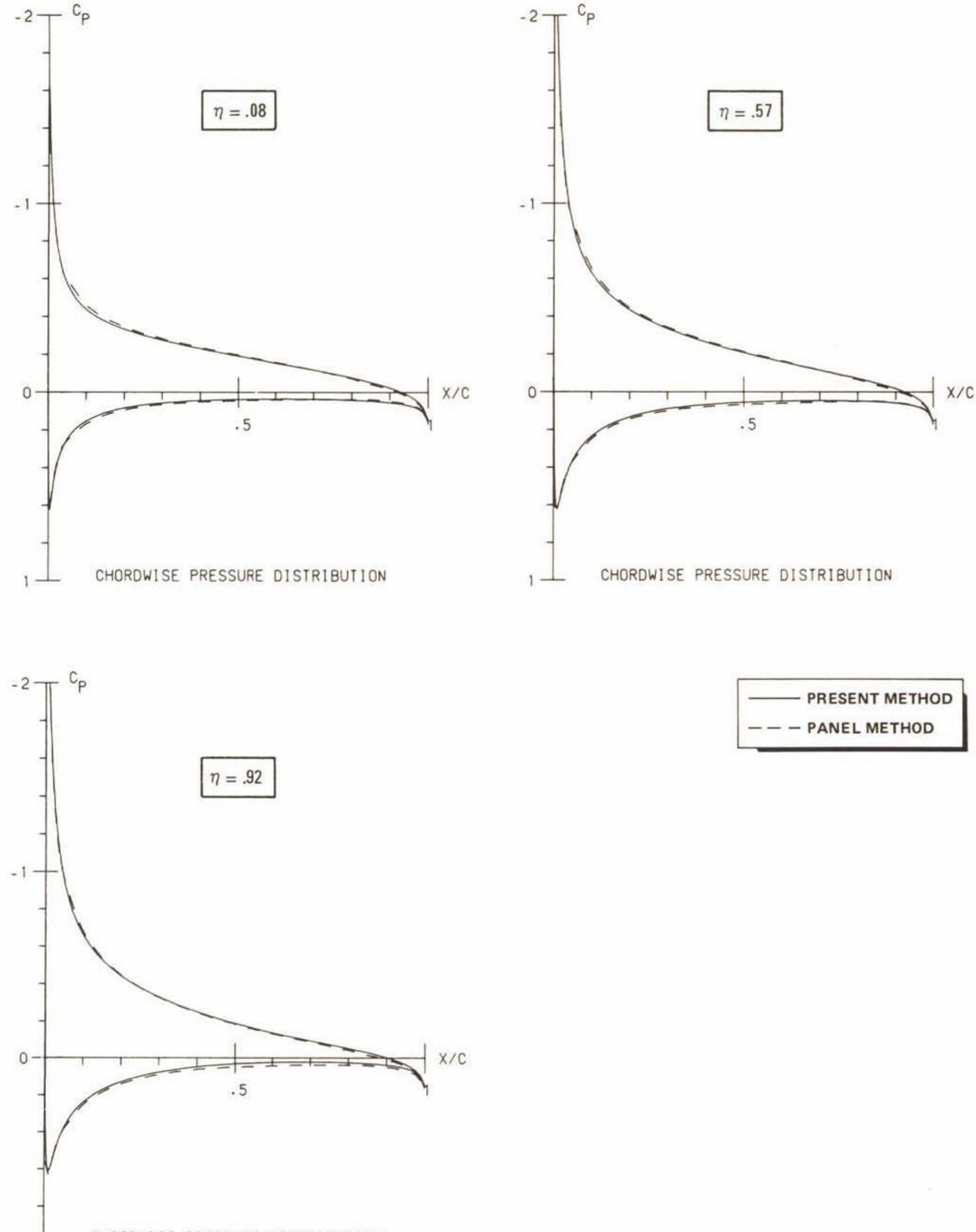


Fig. 8.4.1 Pressure distribution for 5% thick (RAE) wing at $M_\infty = 0$, $\alpha = 5^o$, compared with the higher order panel method of Roberts (Sytsma e.a. [1])

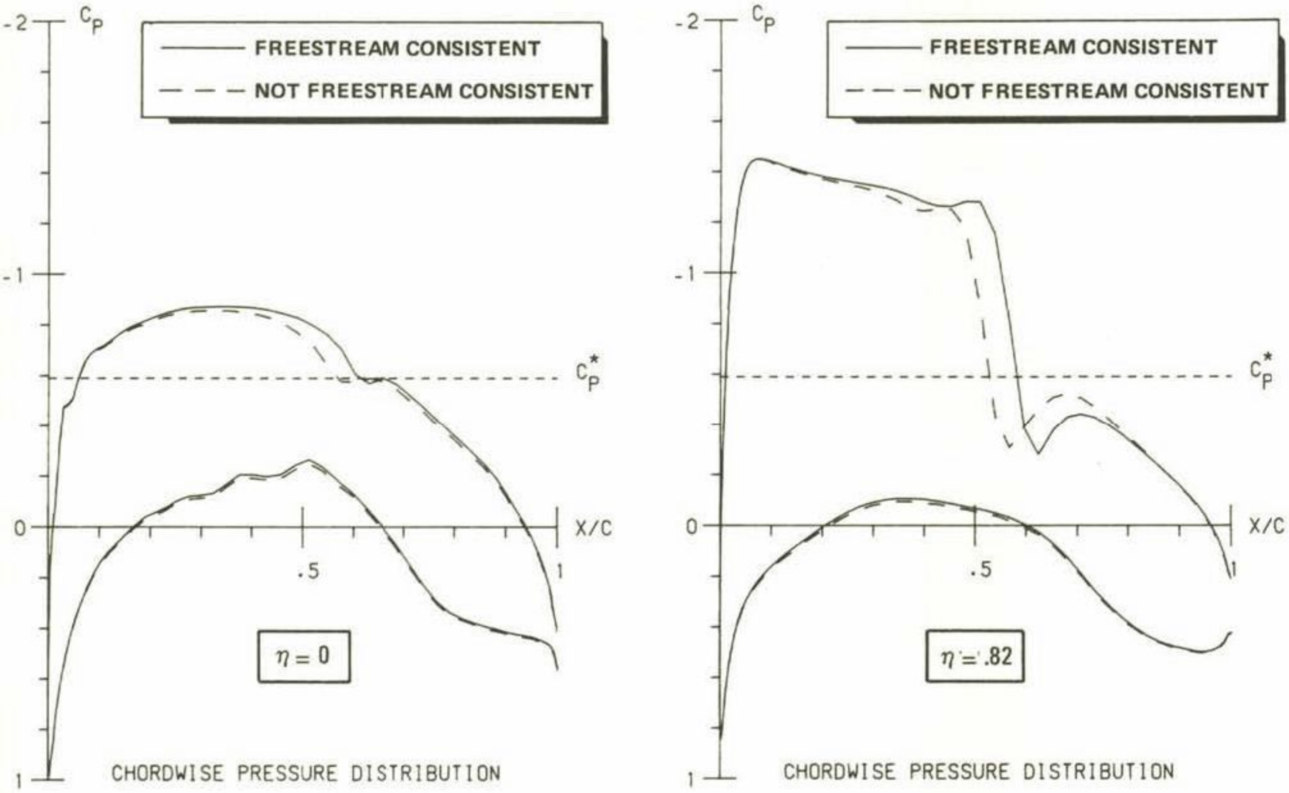
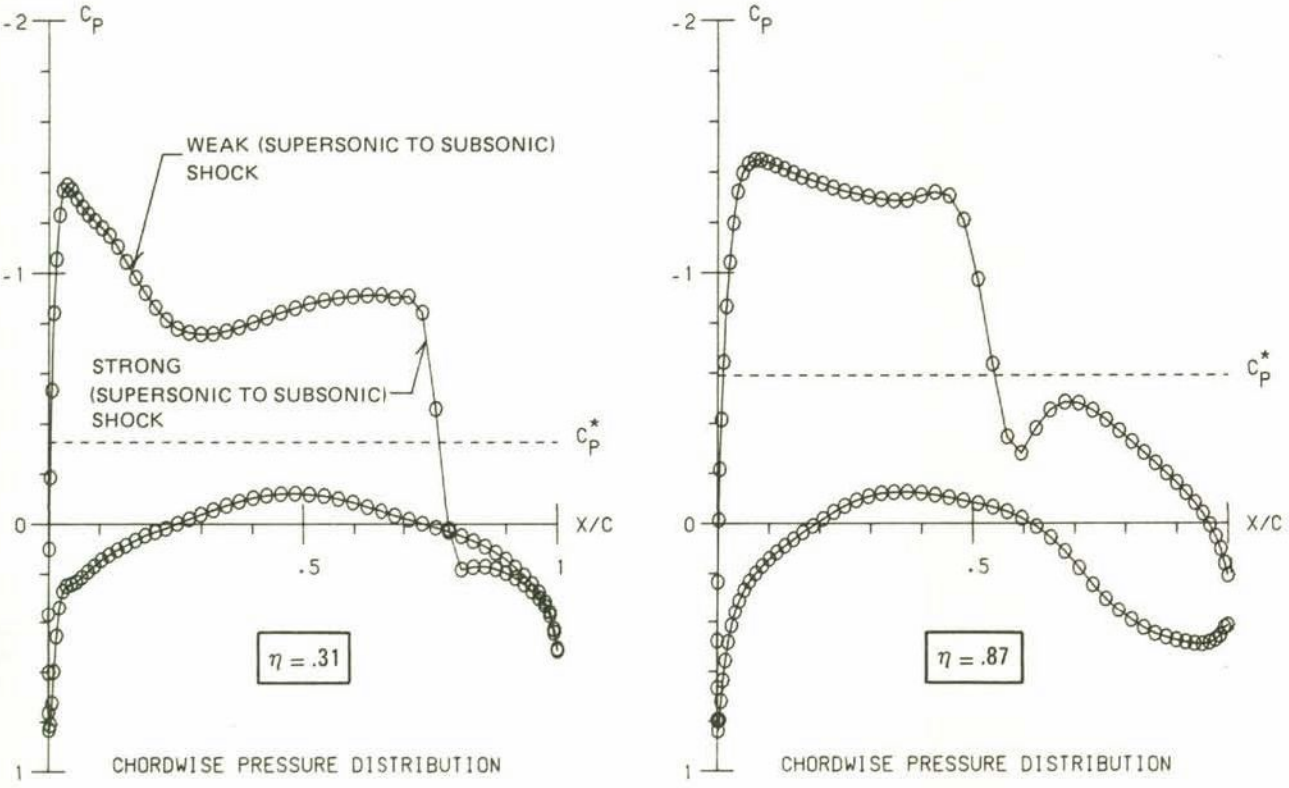


Fig. 8.4.2 Comparison of transonic potential flow solutions on DFVLR-F4 wing with and without a freestream consistent finite volume scheme



a) ONERA–M6 wing, $M_\infty = .84$, $\alpha = 6^0$, shock aligned with grid on wing

b) DFVLR–F4 wing, $M_\infty = .75$, $\alpha = .84^0$, shock not aligned with grid on wing

Fig. 8.4.3 Results of mass flux-vector splitting scheme

8.4.2 Transonic flow about DFVLR-F4 and ONERA-M6 wing

In this subsection computational results will be shown for the transonic flow about the DFVLR-F4 wing, which is representative for the supercritical (transport) wing of today, and for the transonic flow about the ONERA-M6 wing, which is a well-known testcase for transonic flow codes, because it is blown at relatively high freestream Mach-numbers and it is known to have a solution with a double shock system. The convergence histories for these flow cases have already been given in subsections 8.3.2 and 8.3.3.

Figure 8.4.4 shows the pressure distributions and the upper surface isobars for the transonic flow about the DFVLR-F4 wing at $M_\infty = .75$, $\alpha = .84°$. The results are obtained on the grid shown in figure 3.1.6. The grid involves 176*32*32 (= 180224) cells, of which 112*19 are adjacent to the wing.

Figure 8.4.5 shows the pressure distributions and the upper surface isobars for the transonic flow about the ONERA-M6 wing at $M_\infty = .92$, $\alpha = 0°$. The grid is similar to the grid used in the previous experiment.

Figure 8.4.6 shows the pressure distributions and the upper surface isobars for the transonic flow about the ONERA-M6 wing at $M_\infty = .84$, $\alpha = 6°$. The oblique (weak) supersonic to supersonic shock, which is known to be present at this flow condition, is smeared appreciably. This shock smearing has already been discussed in the previous subsection (figure 8.4.3a).

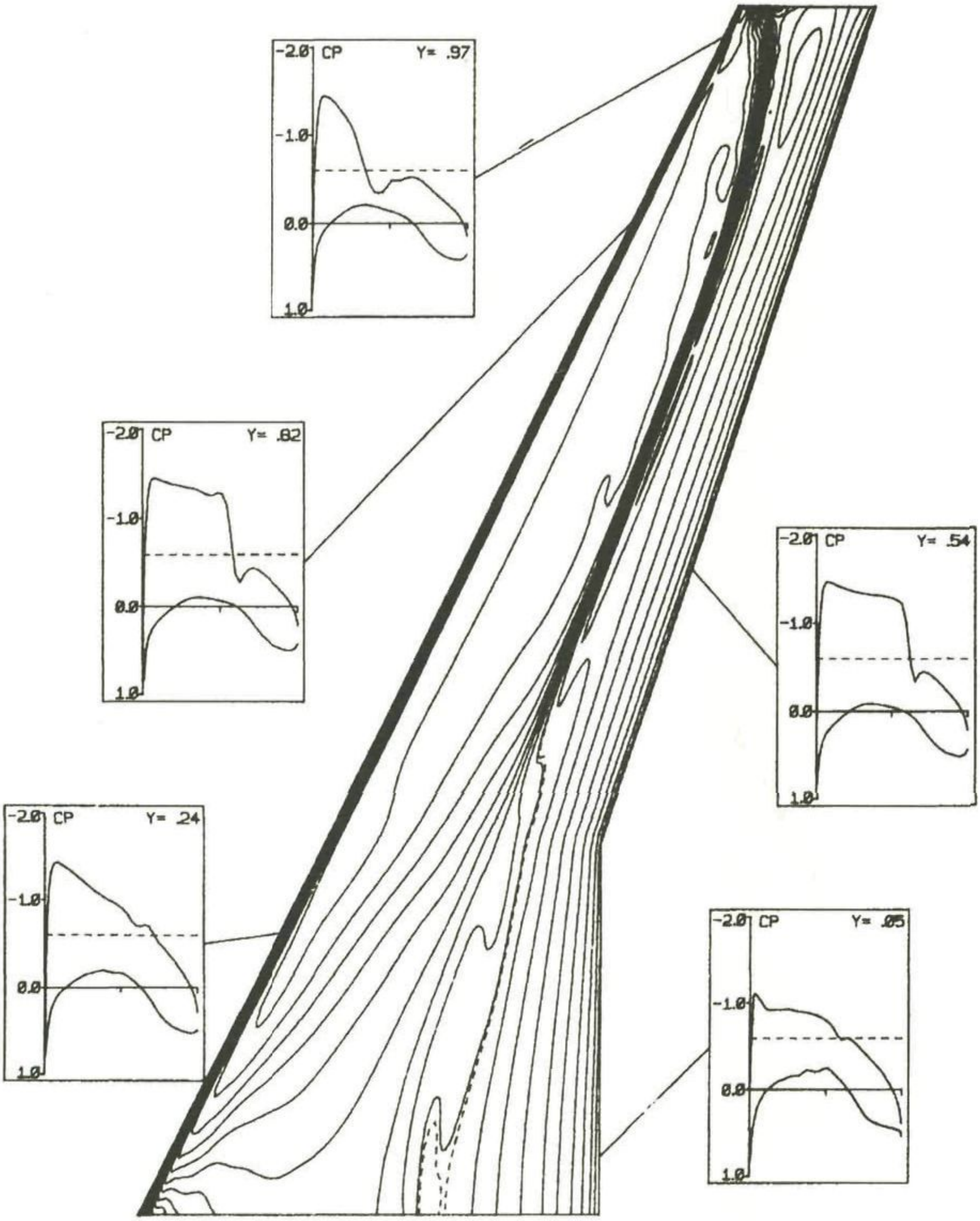


Fig. 8.4.4 Pressure distributions and upper surface isobars for computed transonic potential flow solution on DFVLR-F4 wing at $M_\infty = .75$, $\alpha = .84^\circ$ on 176·32·32 grid

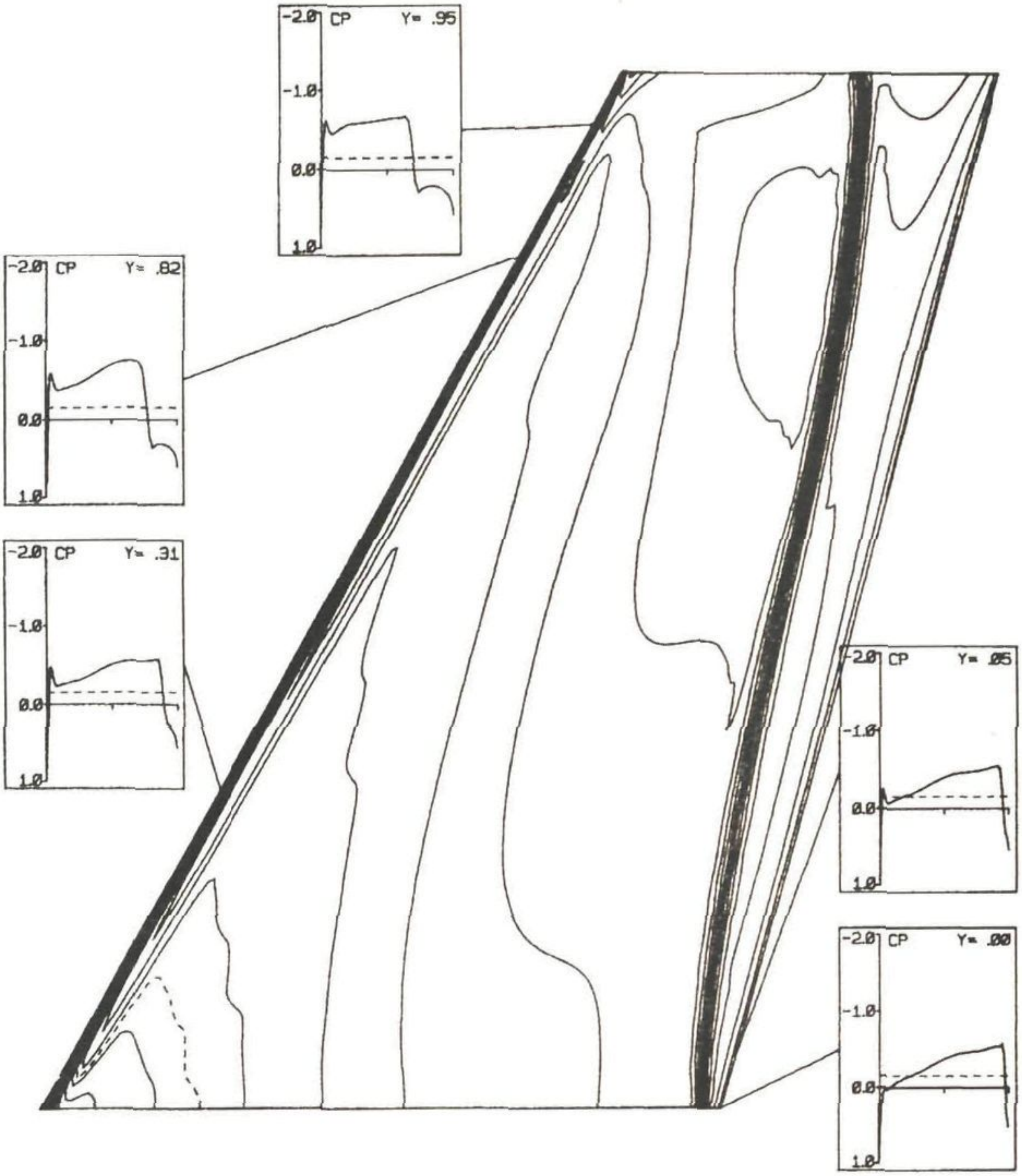


Fig. 8.4.5 Pressure distributions and upper surface isobars for computed transonic potential flow solution on ONERA-M6 wing at $M_\infty = .92$, $\alpha = 0^o$ on $176 \cdot 32 \cdot 32$ grid

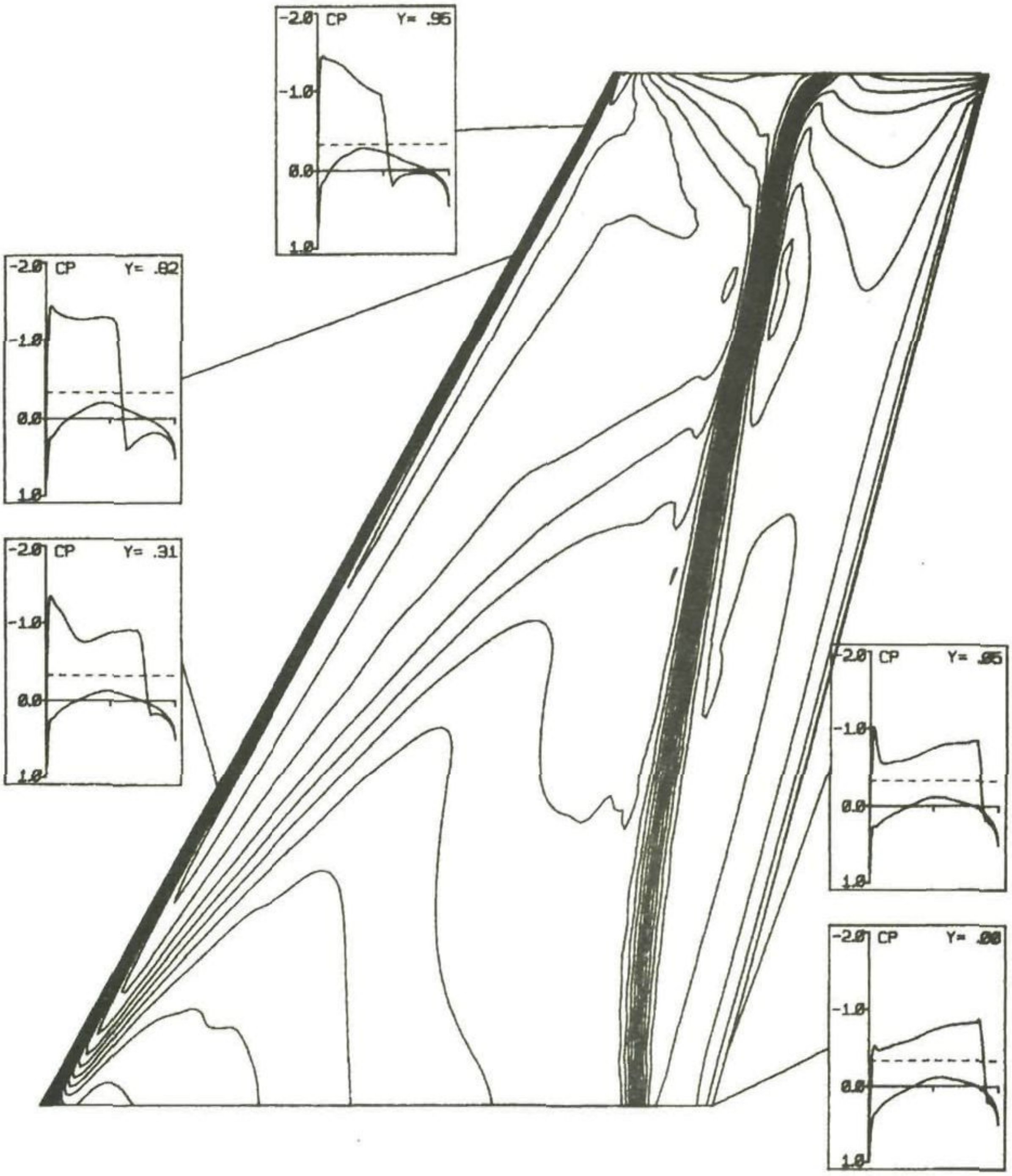


Fig. 8.4.6 Pressure distributions and upper surface isobars for computed transonic potential flow solution on ONERA-M6 wing at $M_\infty = .84$, $\alpha = 6^\circ$ on $176 \cdot 32 \cdot 32$ grid

8.4.3 Results of flow computation at engineering accuracy

For wing flows there are two practical convergence criteria:

1. Convergence of important global parameters to "engineering accuracy", for example 99% convergence of the lift to its final converged value and 99% convergence of the number of supersonic grid points (i.e. the supersonic zone) to its final value.
2. Convergence of the solution, for example the pressure distribution, to some accuracy. Here this will be done by inspecting pressure distributions and checking whether any visible differences can be found between the solution obtained and the final, converged solution ("approximate convergence").

Figure 8.4.7 shows that for the solution of the transonic flow about the DFVLR-F4 wing the first criterion is satisfied at approximately 55 workunits. The convergence point satisfying the second criterion is reached at a little more than 80 workunits. The solutions at the two convergence points are shown in figure 8.4.8 for five representative sections along the wing: the root section $\eta = 0$, a section at $\eta = .13$ having (probably) a forward and a rear shock, a section at the kink in the trailing edge ($\eta = .31$), a section mid-wing ($\eta = .46$) and a section near the wing tip ($\eta = .82$). The figures show that at engineering accuracy the shock location is converged, but not the shock profile: at $\eta = .31$ the shock system of the final solution has not yet been resolved, while also at $\eta = .82$, where the shock is oblique to the grid in the spanwise direction, the shock profile is poorly converged. At the "practically converged" point there is only a visible difference at the kink section in the wing ($\eta = .31$), but at all other stations the solution is fully converged, i.e. there are no significant, visible, differences.

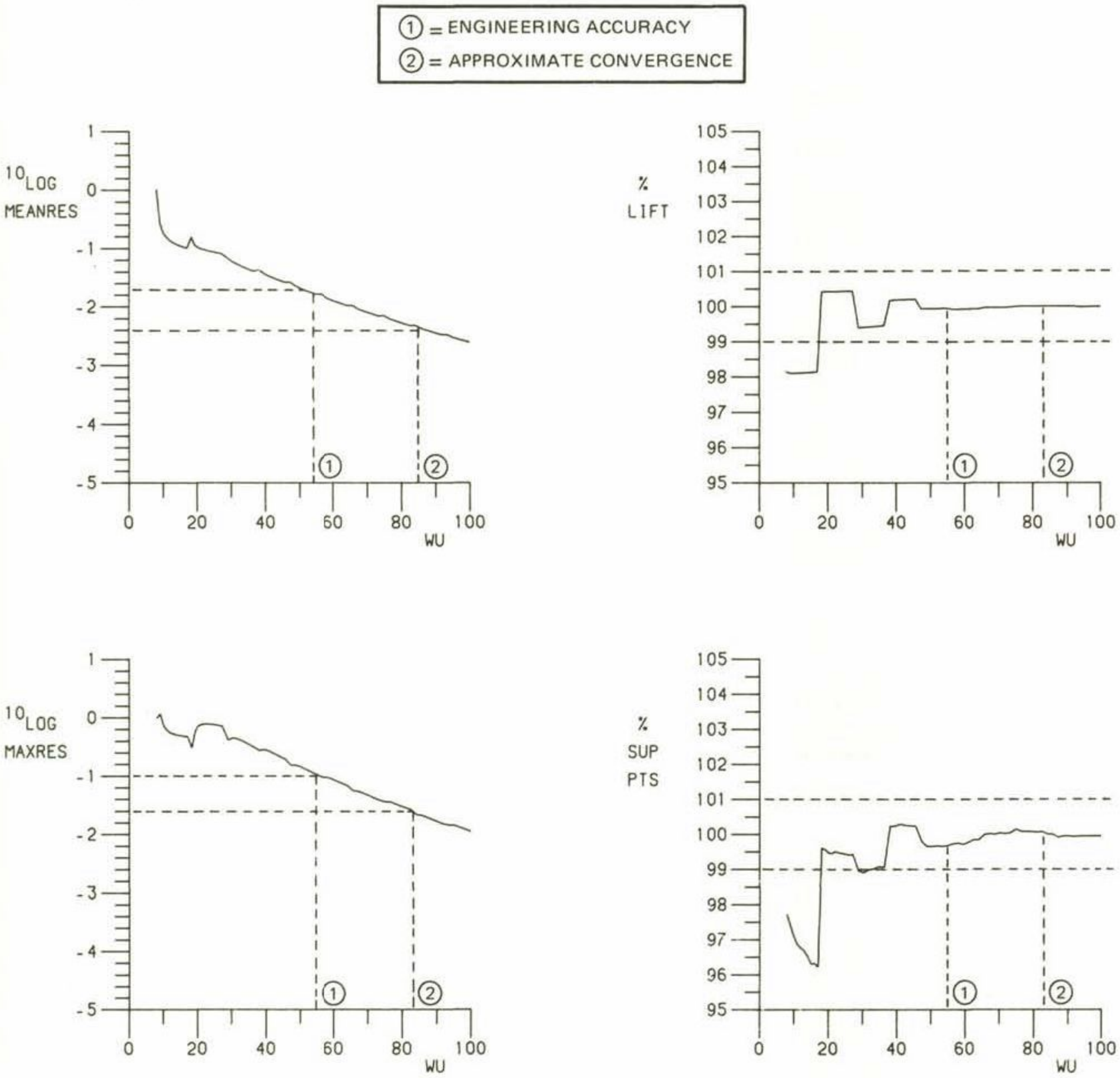
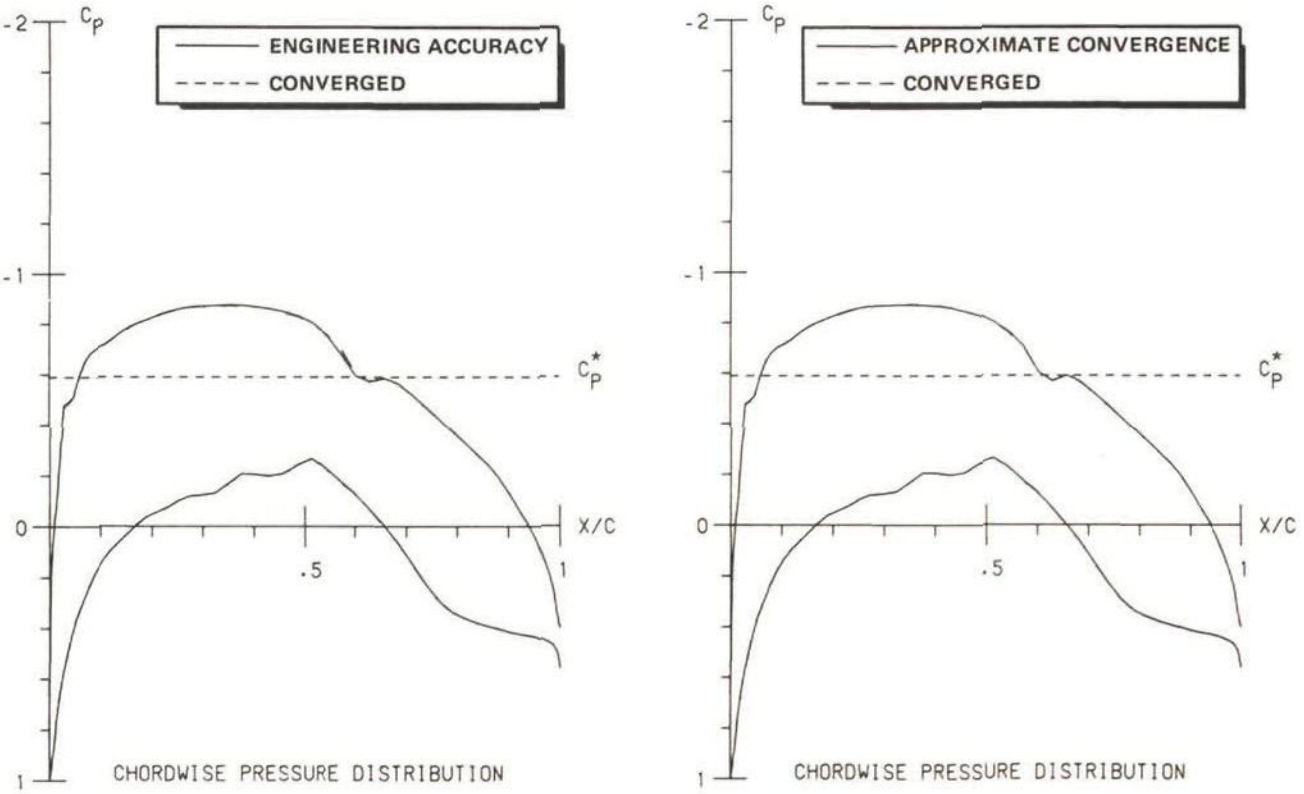
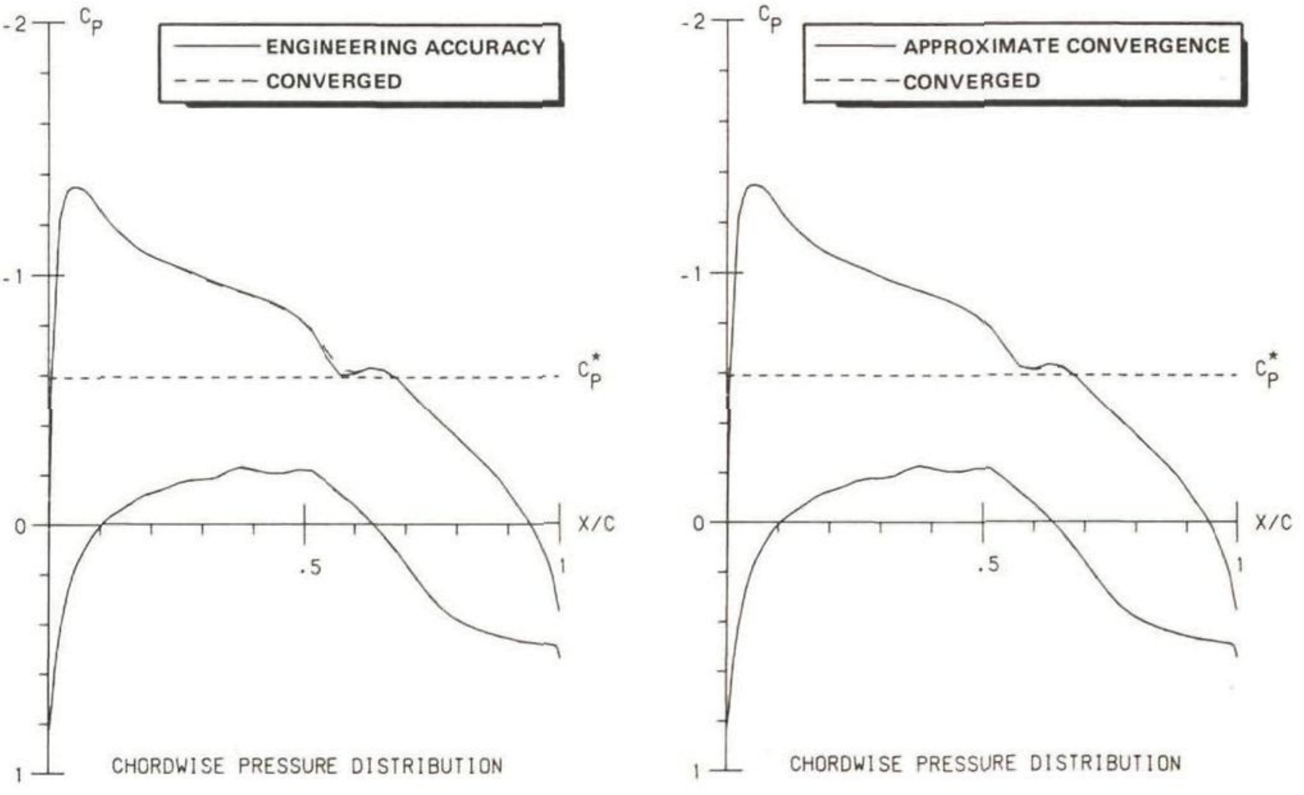


Fig. 8.4.7 "Engineering accuracy" and "approximate convergence" convergence levels for flow solution history of DFVLR-F4 wing at $M_{\infty}= .75$, $\alpha = .84^{o}$ (detail of figure 8.3.1b)

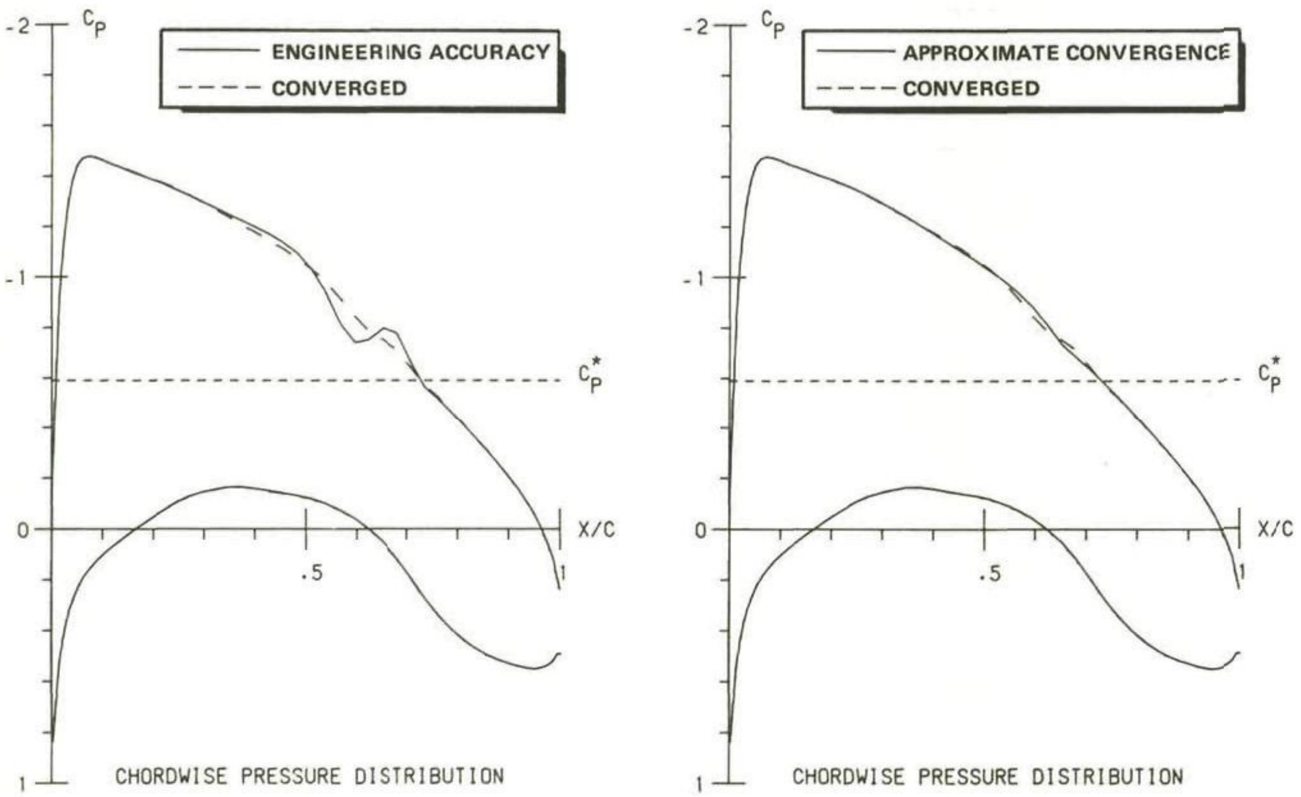


a) Spanwise station $\eta = 0$

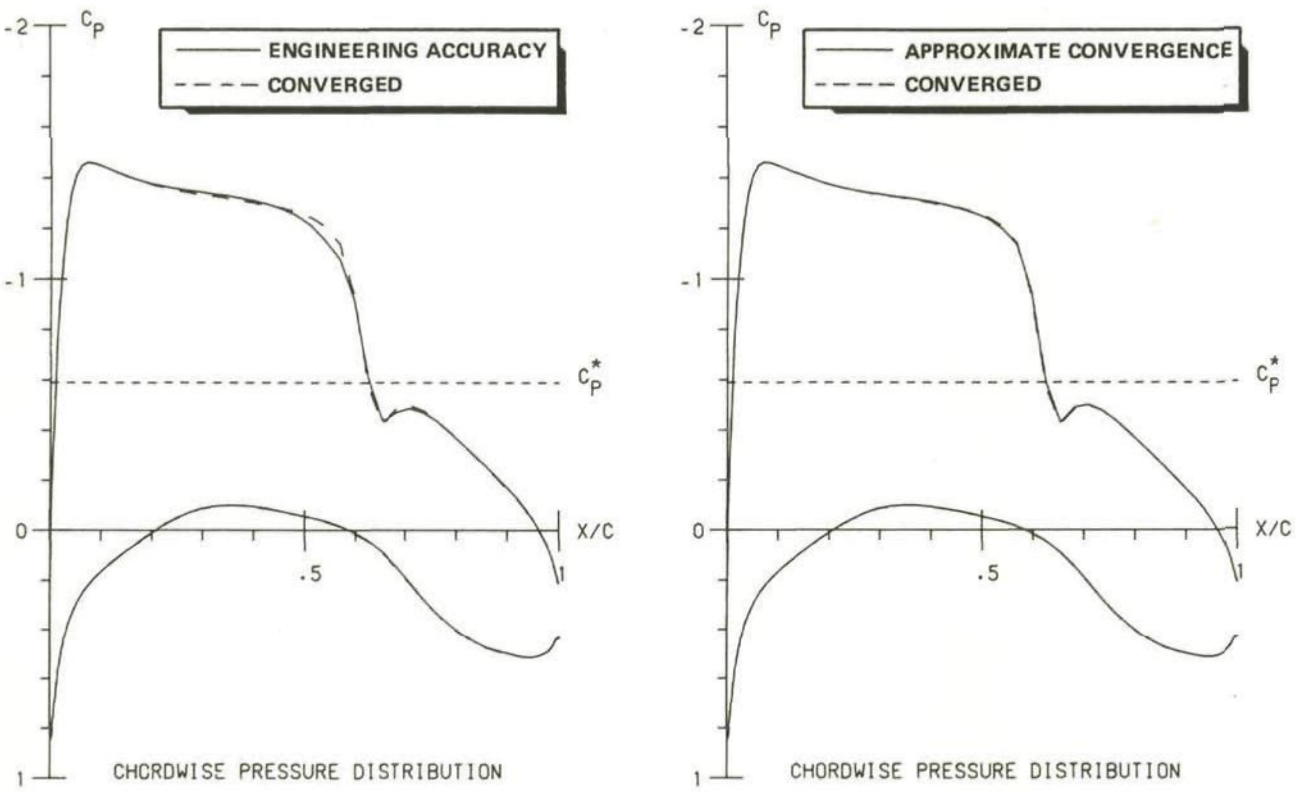


b) Spanwise station $\eta = .13$

Fig. 8.4.8 Comparison of computed flow solutions for DFVLR-F4 wing at $M_\infty = .75$, $\alpha = .84$ on $176 \cdot 32 \cdot 32$ grid at "engineering accuracy" and "approximate convergence" convergence levels

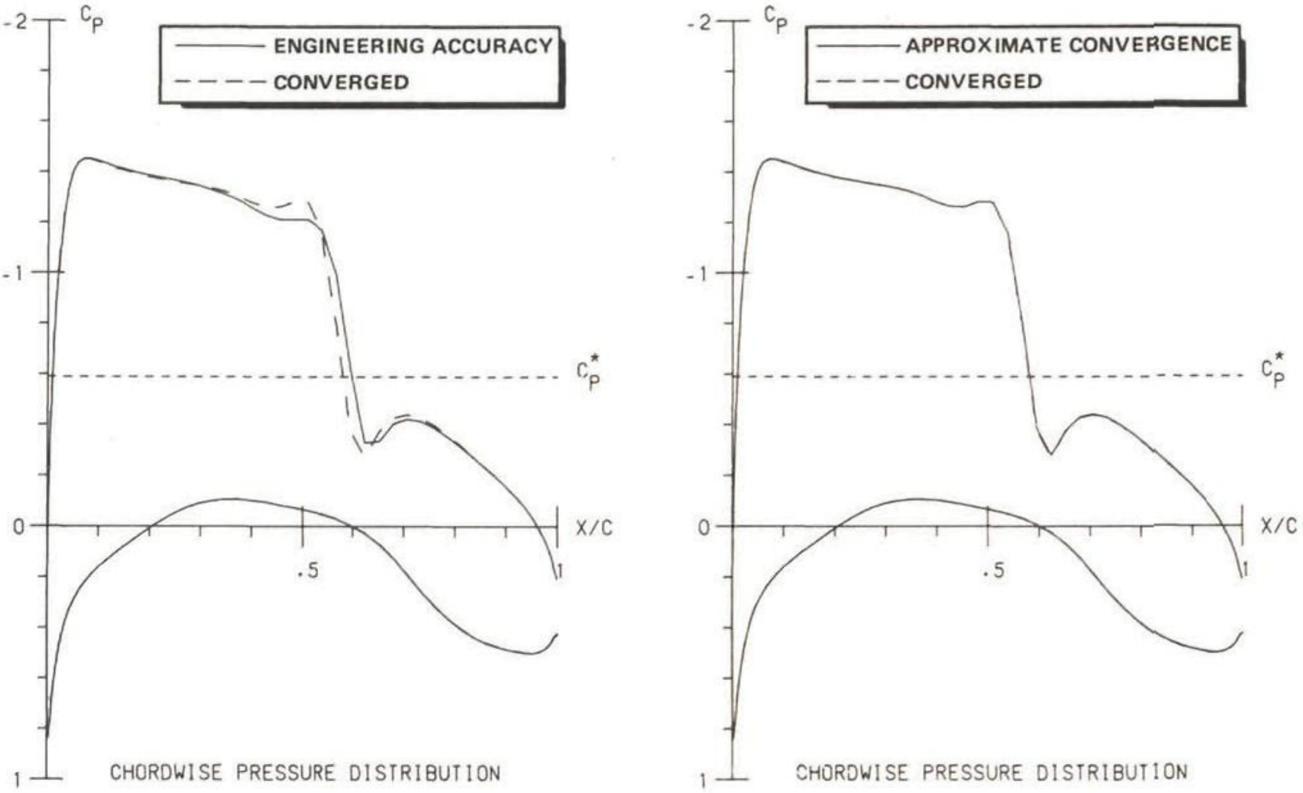


c) Spanwise station $\eta = .31$



d) Spanwise section $\eta = .46$

Fig. 8.4.8 Continued



e) Spanwise section $\eta = .82$

Fig. 8.4.8 Concluded

9 CONCLUSIONS AND FINAL REMARKS

The Incomplete Lower Upper decomposition/Strongly Implicit Procedure (ILU/SIP) has been investigated for use as a smoother in a nonlinear multigrid (MG) method for the solution of the transonic full potential equation in three dimensions. The method has been applied to transonic flows around wings.

The main conclusions of the research presented can be summarized as follows:

- *In subsonic (elliptic) regions of the flow ILU/SIP has been shown to* be quite robust with respect to a wide set of variations of mesh sizes (subsection 6.1.2). The rate of convergence generally improves with increasing ILU/SIP parameter α, as long as α is not chosen too close to one. A good choice is $\alpha = .70$ (sections 6.1 and 8.2).

- In supersonic (hyperbolic) regions of the flow the ILU/SIP parameter α should be set to unity ($\alpha = 1.$). An explicit temporal damping term is required to obtain unconditional stability (section 6.2).

- ILU/SIP is reasonably robust with respect to grid skewness, i.e. it shows a moderate convergence degradation as long as the angle between grid coordinate lines deviates less than 45 degrees from orthogonal. *In case of larger skewness angles an underrelaxation of the residual* may be necessary to keep the algorithm stable (subsection 6.1.3). Such large deviations from orthogonality are however also undesirable from an approximation point of view, because it affects the accuracy of the discretization.

- The rate of convergence of MG-ILU/SIP can be improved by performing a priori grid optimization in those regions of the grid where the flow is subsonic. It has been shown that computational cells having mesh sizes of three different orders of magnitude should be avoided. This requirement can be satisfied by using a grid which is reasonably square in grid planes perpendicular to the wing (subsections 6.1.2, 8.2.2).

- The grid should be generated such that the solution method is fast in regions where the freestream flow (the starting solution) is not a good guess for the final solution. In other regions one can afford to generate a grid for which the solution method will be relatively slow (subsection 8.2.2).

- The calculation of the transonic flow about a realistic wing indicates that MG-ILU/SIP needs only .8-1.4 order of magnitude reduction of the maximum residual to provide results for the lift and the size of the supersonic zone of engineering accuracy. Convergence to this engineering level of accuracy is relatively fast compared with the asymptotic convergence rate (subsection 8.3.2).

- In flow calculations about wings the asymptotic convergence rate is dominated by the computational cells that have the "worst" combination of aspect ratios. In practical wing grids the asymptotic convergence is generally rather slow, and very accurate (highly converged) results will be hard to attain (subsection 8.3.2). A way to improve on this situation (if required) is to apply extrapolation techniques (see e.g. Van der Wees, van der Vooren and Meelker [1]), to use plane-relaxation as introduced by Thole [1] or to use the multigrid method as a preconditioner in a conjugate gradients algorithm.

- At a certain (specified) reduction of the error level, MG-SLR as well as MG-ILU/SIP provide solutions of better quality than the corresponding single grid algorithm. This property is of great value in practice, because it can be used to advantage by specifying a lower convergence level (subsection 8.3.4), thus reducing computation time.

- The unconditional stability as well as the robustness with respect to aspect ratios (conclusions 2 and 3), but also the absence of a preferred sweep-direction in the coding, are especially of value if complicated configurations involving strongly varying local flow directions and highly stretched grids are considered (e.g. complex aircraft configurations, air intakes).

If one wants to avoid that a solution method for three-dimensional problems such as considered here becomes overly complicated, one has to accept a certain degree of degradation of rate of convergence under certain circumstances. However, one may design the grid such that the convergence is sufficiently fast for practical applications; in general this will involve adding extra grid points. To this end, a grid analysis tool has been developed (Doctor [1]) which gives an indication whether the grid is sufficiently suited to allow for fast solution by MG-ILU/SIP (e.g. maximum skewness angles, maximum aspect ratios and estimated rates of convergence are listed). This makes it possible to adapt a grid to the requirements of a flow solver.

REFERENCES

Bakhvalov, N.S., [1], On the convergence of a relaxation method with natural constraints on the elliptic operator, USSR Comput. Math. and Math. Phys. 6, 5, 1966.

Batchelor, G.K., [1], An introduction to Fluid Dynamics, Cambridge University Press, 1967.

Boerstoel, J.W. and Kassies, A. [1], Integrating multi-grid relaxation into a robust fast-solver for transonic potential flow around lifting aerofoils, AIAA Paper 83-1885, 1983.

Brandt, A. [1], Multi-level adaptive solutions to boundary value problems, Math. of Comp., Vol. 31, Nr. 138, 1977.

Brédif, M. [1], Finite element calculation of potential flow around wings,ONERA-TP-1984-068, 1984.

Brédif, M. [2], A fast finite element method for transonic potential flow calculations, AIAA 83-0507, 1983.

Carr, M.P. [1], Accuracy study of transonic flow computations for three dimensional wings, Paper No. 19 presented at AGARD Symp. on validation of Comp. Fl. Dyn., Lisbon, Portugal, 2-5 May, 1988.

Caughey, D.A. [1], Multi-grid calculation of three-dimensional transonic potential flows, AIAA 83-0374, 1983.

Dendy Jr., J.E., Black Box Multigrid, J. Comp. Phys., 48, pp. 366-386, 1982.

Doctor, F. [1], User's guide of ANAGRID, a program to analyse the quality of a grid, NLR memorandum IN-87-007 L, 1987.

Fedorenko, R.P. [1], A relaxation method for solving elliptic difference equations, USSR Comput. Math. and Math. Phys. 1, 5, 1961.

Flores, J., Holst, T.L., Kwak, D. and Batiste, D. [1], A New Consistent Spatial Differencing Scheme for the Transonic Full Potential Equation, AIAA 83-0373, 1983.

Foerster, H., Witsch, K. [1], Multigrid software for the solution of elliptic systems on rectangular domains: MG00 (release 1). In: Hackbusch-Trottenberg.

Fuchs, L.J. [1], Finite difference methods for plane, steady inviscid transonic flows, TRITA-GAD-2 rep., Stockholm, 1977.

Hackbusch, W. [1], A fast iterative method solving Poisson's equation in general region. In: Bulisch, R., Griegorieff, R.D., Schröder, J. (eds.), Numerical treatment of differential equations. Proc. Oberwolfach, July 1976, Lecture Notes in Math. 631, Springer, Berlin, 1978.

Hackbusch, W. [2], Multi-Grid Methods and Applications, Springer Series in Computional Mathematics 4, Springer, Berlin, 1985.

Hackbusch, W. and Trottenberg, U. (eds.) [1], Multi-grid methods, Proceedings, Köln-Porz, Nov. 1981, Lecture Notes in Mathematics 960, Springer, Berlin 1982.

Heerema, F.J. and Kreijkamp, H.A. [1], A multi-discipline and integrated facility for management and evaluation of engineering data, presented at 6th Int. Conf. and Exh. on Computers in Design Engineering, CAD84, Brighton UK, April 3-5, 1984.

Hemker, P.W. [1], The incomplete LU-decomposition as a relaxation method in multi-grid algorithms. In: J.J.H. Miller (ed.), Boundary and interior layers - computational and asymptotic methods, pp. 306-311, Boole Press, Dublin, 1980.

Hemker, P.W., Wesseling,P. and de Zeeuw, P.M. [1], A portable vector-code for autonomous multigrid modules, Centre for Mathematics and Informatics Report NW 154/83, 1983.

Holst, T.L. [1], Numerical solution of transonic wing flow fields, AIAA Paper 82-0105, 1982.

Holst, T.L. [2], Numerical Computation of transonic flow governed by the full-potential equation, VKI Lecture Series 1983-04, paper 4, 1983.

Holst, T.L., Slooff, J.W., Yoshihara, H., and Ballhaus, W.F. [1], Applied computational transonic aerodynamics, AGARD Agardograph 266, 1982.

Jameson, A. [1], Numerical solution of the three-dimensional transonic flow over a yawed wing, AIAA Paper presented at the 1st AIAA Comp. Fluid, Dyn. Conf., Palm Springs, Cal., July 19-20, pp. 18-26, 1973.

Jameson, A. [2], Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple grid method, AIAA 79-1458 CP, 1979.

Jameson, A. and Caughey, D.A. [1], A finite volume method for transonic potential flow calculations, AIAA Paper 77-635-CP, 1977.

Kettler, R. [1], Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. In: Hackbusch-Trottenberg.

Kettler, R. [2], Linear multigrid methods for numerical reservoir simulation. Ph.D. Thesis, University of Technology Delft, The Netherlands, 1987.

Liepmann, H.W., Roshko, A. [1], Elements of gasdynamics, Wiley and Sons, New York, 1957.

Loeve, W. and Van der Vooren, J. [1], Stromingsberekeningen voor vliegtuigen. In: poly-automatiseringszakboekje, ISBN 90-6228-086-2, 1987, (in Dutch).

Meijerink, J.A. and Van der Vorst, H.A. [1], An iterative solution method for linear problems of which the coefficient matrix is a symmetric M-matrix, Math. of Comp., Vol. 31, Nr. 137, 1977.

McCarthy, D.R., Reyhner, R.A. [1], A multi-grid code for three-dimensional transonic potential flow about axisymmetric inlets at angle of attack, AIAA 80-1365, 1980.

Melnik, R.E., Mead, H.R. [1], A Multi-Grid Method for the Computation of Viscid/Inviscid Interaction on Airfoils, AIAA 83-0234, 1983.

Murman, E.M. and Cole, J.D. [1], Calculation of plane steady transonic flows, AIAA Journal, Vol. 9, 1971, pp. 114-121.

Nicolaides, R.A. [1], On multiple grid and related techniques for solving discrete elliptic systems, J. Comp. Phys. 19, 1975.

Nowak, Z. and Wesseling, P. [1], Multigrid acceleration of an iterative method with applications to transonic potential flow. In: R. Glowinski and J.L. Lions (eds.), Computing methods in applied sciences ans engineering VI. Proceedings, Versailles, 1983, pp. 199-217, North-Holland, Amsterdam, 1984.

Osher, S., Hafez, M.M. and Whitlow jr., W. [1], Entropy condition satisfying approximations for the full potential equation of transonic flow, Math. of Comp., Vol. 44, Nr. 169, 1985.

Pau, V., Lewis, E. [1], Application of the multigrid technique to the pressure-correction equation for the SIMPLE algorithm, In: Trottenberg, U., Hackbush, W. (eds.), Multigrid Methods: Special Topics and Applications, papers presented at 2nd Eur. Conf. on Multigrid Methods, Cologne 1985, GMD Studien Nr. 110, 1986.

Raj, P. [1], A multigrid method for transonic wing analysis and design, AIAA 83-0262, 1983.

Sankar, N.L. [1], A multi-grid strongly implicit procedure for two-dimensional transonic potential flow problems, AIAA Paper 82-0391, 1982.

Schmidt, W., and Jameson, A. [1], Applications of multi-grid methods for transonic flow calculations. In: Hackbusch-Trottenberg.

Shapiro, A. [1], The dynamics and thermodynamics of compressible fluid flow, Ronald Press Company, New York, 1953.

Shmilovich, A. and Caughey, D.A. [1], Application of the multi-grid method to calculations of transonic potential flow about wing-fuselage combinations, J. Comp. Phys. 48, pp. 462-484, 1982.

Sonneveld, P., Wesseling, P. and De Zeeuw, P.M. [1], Multigrid and Conjugate Gradient methods as convergence acceleration techniques. In: Paddon, P.J. and Holstein, H. (eds.), Multigrid methods for integral and differential equations, Clarendon Press, Oxford, 1985.

South jr, J.C. and Hafez, M.M. [1] Stability analysis of intermediate boundary conditions in approximate factorization schemes, AIAA Paper 83-1898-CP, 1983.

South, J.C., Brandt, A. [1], Applications of a multi-level grid method to transonic flow calculations, ICASE Report 76-8, March 1976.

Stone, H.L. [1], Iterative solution of implicit approximations of multi-dimensional partial difference equations, SIAM J. Numer. Anal., Vol. 5, Nr. 3, 1968.

Stüben, K. and Trottenberg, U. [1], Multigrid methods: fundamental algorithms, model problem analysis and applications. In: Hackbusch-Trottenberg.

Sytsma, H.S., Hewitt, B.L. and Rubbert, P.E. [1], A comparison of panel methods for subsonic flow computation, AGARDograph No. 241, February 1979.

Thole, C.A., Basic smoothing procedures for the multigrid treatment of ellipic 3D-operators, Proc. Conf. Oberwolfach, Dec. 8-13, 1984. In: Braess, D. Hackbusch, W. Trottenberg, U. (eds.), Advances in Multi-Grid Methods, Viewig & Sohn, Braunschweig/Wiesbaden, 1986.

Van der Wees, A.J., Van der Vooren, J. and Meelker, J.H. [1], Robust calculation of 3D transonic potential flow based on the non-linear FAS multi-grid method and incomplete LU decomposition, AIAA Paper 83-1950-CP, 1983.

Van der Wees, A.J. [1], Robust calculation of 3D transonic potential flow based on the non-linear FAS multi-grid method and a mixed ILU/SIP algorithm. In: J.G. Verwer (ed.), Colloquium Topics in Numerical Analysis, CWI Syllabus 5, 1985.

Van der Wees, A.J. [2], FAS multigrid employing ILU/SIP smoothing: a robust fast solver for 3D transonic potential flow. In: Hackbusch, W. and Trottenberg, U. (eds.), Multigrid Methods II, Proc. 2nd Eur. Conf. on Multigrid Methods, Cologne 1985, Lecture Notes in Mathematics 1228, Springer Berlin, 1986.

Wesseling, P. [1], Numerical solution of the stationary Navier-Stokes equations by means of a multiple grid method and Newton iteration, Report NA-18, Delft University of Technology, 1977.

Wesseling, P. [2], A robust and efficient multigrid method, In: Hackbusch-Trottenberg.

Wesseling, P., Sonneveld, P. [1], Numerical experiments with a multiple grid and a preconditioned Lanczos method. In: R. Rautman (ed.), Approximation Methods for Navier-Stokes Problems, Proceedings, *Paderborn 1979, Lecture Notes in Math. 771, pp. 543-562, Springer,* Berlin, 1980.

Zedan, M. and Schneider, G.E. [1], 3-D Modified strongly implicit procedure for finite difference heat conduction modelling, AIAA Paper 81-1136, 1981.

APPENDIX A

Abbreviations and symbols

The list of abbreviations and symbols given in this appendix serves as a reference table for abbreviations and symbols that are used throughout one or more chapters. Abbreviations and symbols that are used only locally in the thesis are therefore not mentioned.

Abbreviations

FAS	Full Approximation Scheme
FV	Finite Volume (method)
ILU	Incomplete Lower Upper decomposition (algorithm)
MG	MultiGrid (method)
NLR	National Aerospace Laboratory NLR
SIP	Strongly Implicit Procedure (algorithm)
SLOR, SLR	Successive Line (Over) Relaxation
t.e., TE	trailing edge
WU	Work Unit

General list of symbols

C_p	pressure coefficient
G	contravariant metric sensor
H	Jacobian matrix
L	general nonlinear operator
M	Mach number
N	normal tensor
NI	half of number of cells in chordwise direction
NJ	number of cells in spanwise direction

NK	number of cells in wing-normal direction
U	velocity tensor
a	speed of sound
g^{ij}, $g^{\alpha\beta}$	contravariant metric tensor component
g_{ij}, $g_{\alpha\beta}$	covariant metric tensor component
h	Jacobian
m,n	mutually orthogonal directions normal to stream direction s
n	normal vector
q	velocity
s	stream direction
u	velocity vector
x	Cartesian coordinate system
Γ	potential jump at trailing edge of wing
α	ILU/SIP parameter
α	angle of attack (when used in combination with M)
γ	specific heat ratio $\equiv$ 1.4 for diatomic gas
$\delta^{ij}, \delta^{i}_{j}$	Kronecker symbol
ξ	general coordinate system
ρ	density
φ	perturbation potential
ϕ	correction perturbation potential; approximation of φ
$(\ )^{i}, (\ )^{\alpha}$	contravariant component in i- (or α-) direction
$(\ )^{i}, (\ )^{\alpha}$	covariant component in i- (or α-) direction
$(\ )_{,i}$	differentiation in i-direction
$(\ )_{i,j,k}$	variable value in point (i,j,k)
$(\ )_{n}$	component in normal direction
$(\ )_{pqr}$	variable value in point (i+p, j+q, k+r)
$(\ )_{\infty}$	freestream value
$(\ )^{*}$	sonic value

List of symbols specific for chapter 2

R	specific gas constant
S	entropy
S_w	vortex sheet
T	temperature
e	specific internal energy
c_p	specific heat at constant pressure
c_v	specific heat at constant volume
h_0	specific total enthalpy
p	pressure
Φ	potential
ω	vorticity
$\overline{(\ )}$	"frozen" value for $\varphi = \bar{\varphi}$
$[\![\ ]\!]$	jump operator

List of symbols specific for chapter 4

F	contravariant flux tensor
$\partial\Omega$	boundary of Ω
$\partial\Omega_F$	part of $\partial\Omega$ with prescribed influx
Ω	computational domain
δ_i	central difference operator in i-direction
$\overleftarrow{\delta}_i$	backward difference operator in i-direction
$\tilde{\delta}_i$	central difference operator in i-direction using grid points $i+1$, $i-1$
$\bar{\delta}_i$	composite differencing/averaging operator in i-direction
μ_i	averaging operator in i-direction

$\bar{\xi}_i$	local general coordinate system
$(\ )^h$	discrete form of variable/operator

List of symbols specific for chapter 5

B	error matrix
F, f, g	general right hand side terms
G^K	grid on grid level K
I_K^{K-1}	potential restriction operator from grid K to grid K-1
$\tilde{L}$	Fréchet derivative of L, also system matrix
L^*	iteration matrix
N	finest grid level
R	residual
W_K^{K-1}	residual restriction operator from grid K to grid K-1
a^{pqr}	matrix entry of L corresponding to grid point (i+p, j+q, k+r) in the difference molecule
b^{pqr}	matrix entry of B corresponding to grid point (i+p, j+q, k+r) in a difference molecule
$\hat{g}$	normal velocity at boundary
h	mesh size
γ	number of coarse grid corrections per multigrid cycle
$\tilde{\delta}_i$	central difference operator in i-direction using grid points i+1, i-1
λ	wavelength
ν	number of smoothings
$\Delta\xi$	mesh size
$\Delta\varphi$	potential correction
ψ	correction
$(\ )^K$	variable/operator on grid K, K = N, ...,1
$(\ )^n$	variable at iteration number n

List of symbols specific for chapters 6 and 8

N_{it}	smoothing efficiency
$V_N(x,y)$	V-cycle on N grid levels performing x smoothings on grid 1 and y smoothings after each prolongation
f	right hand side term
η	scaled spanwise coordinate
θ_1	wing sweep angle
θ_2	grid skewness angle
θ_3	grid rotation angle
λ	error reduction per work unit
$\Delta\xi^i$	mesh size in i-direction
$\bar{\rho}$	smoothing number (maximum short wave reduction factor)
ω	over-/underrelaxation factor

APPENDIX B

Error matrices for the SLR and ILU/SIP algorithms

In this appendix error matrices will be derived for SLR and ILU/SIP when solving the partial differential equation

$$g^{\alpha\beta}\varphi_{,\alpha\beta} = f. \tag{B.1}$$

These error matrices are required for the calculation of the smoothing numbers presented in section 6.1 (smoothing analysis for elliptic problems).

For the application of Fourier smoothing analysis it is necessary to assume that the diagonals of the system matrix A and the lower and upper triangular matrices L and U are constant. This is the case if we solve (B.1) with constant coefficients $g^{\alpha\beta}$ on an infinitely large domain, i.e. for the grid point indices (i,j,k) we have

$$(i,j,k) \in (-\infty,\infty)^3. \tag{B.2}$$

The following central finite difference discretization will be used (using the notation of section 4.2),

$$\varphi_{,\alpha\beta} \approx \begin{cases} \delta_\alpha \delta_\beta \varphi & \text{if } \alpha = \beta, & \text{(B.3a)} \\ \tilde{\delta}_\alpha \tilde{\delta}_\beta \varphi & \text{if } \alpha \neq \beta. & \text{(B.3b)} \end{cases}$$

This leads to the algebraic equation

$$(A\varphi)_{i,j,k} = f_{i,j,k}. \tag{B.4}$$

A relaxation process to solve (B.4) can be described as

$$(A^{*}\Delta\varphi)_{i,j,k} = ((A+B)\Delta\varphi)_{i,j,k} = f_{i,j,k} - (A\varphi)_{i,j,k}. \tag{B.5}$$

The operators A and B can conveniently be represented in stencil operator notation, i.e. (redefining i,j) :

$$(A\varphi)_i = \sum_j A(i,j)\varphi_{i+j}, \quad i = (i_1,\dots,i_d), \quad j = (j_1,\dots,j_d), \quad d=3 \qquad (B.6)$$

In equation (B.6) A(i,j) is the coefficient referring to point j in the stencil in point i. For example, in case of the Laplace operator on a rectangular domain we have

$$\begin{aligned} A(i,(-1,0,0)) &= A(i,(1,0,0)) = -g^{11} \quad, \\ A(i,(0,-1,0)) &= A(i,(0,1,0)) = -g^{22} \quad, \\ A(i,(0,0,-1)) &= A(i,(0,0,1)) = -g^{33} \quad, \\ A(i,(0,0,0)) &= 2(g^{11}+g^{22}+g^{33}) \quad, \\ A(i,j) &= 0 \text{ elsewhere} \quad, \end{aligned} \qquad (B.7)$$

so that equation (B.6) becomes

$$\begin{aligned} (A\varphi)_{i,j,k} = &- g^{11}\varphi_{i-1,j,k} - g^{22}\varphi_{i,j-1,k} - g^{33}\varphi_{i,j,k-1} \\ &+ 2\,(g^{11} + g^{22} + g^{33})\,\varphi_{i,j,k} \\ &- g^{11}\varphi_{i+1,j,k} - g^{22}\varphi_{i,j+1,k} - g^{33}\varphi_{i,j,k+1}. \end{aligned} \qquad (B.8)$$

The stencil of the product AC of the two operators A and C, where C also has a stencil of type (B.6), is given by

$$\begin{aligned} (AC\varphi)_i &= \sum_j A(i,j)\,(C\varphi)_{i+j} \\ &= \sum_j A(i,j) \sum_k C(i+j,k)\,\varphi_{i+j+k} \\ &= \sum_{j,k} A(i,j)\,C(i+j,k)\,\varphi_{i+j+k} \end{aligned} \qquad (B.9)$$

and consequently

$$\begin{aligned} (AC\varphi)_i &= \sum_l D(i,l)\,\varphi_{i+l}\,, \\ D(i,l) &= \sum_{j+k=l} A(i,j)\,C(i+j,k). \end{aligned} \qquad (B.10)$$

Of course, a stencil operator A, equation (B.6), can be associated with a system matrix A: A(i,j) is the entry for row (i.e. equation number) i and column (i.e. stencil entry) number i+j. Equation (B.10) then represents multiplication of the matrices A and C. Note that the ordering of the rows and the columns (numbering of the unknowns) has not yet been specified for the system matrix A. We will assume lexicographical ordering, i.e. the columns in row i of the system matrix are ordered from left to right as follows:

$$\begin{aligned}&A(i,(0,0,-1)),\ A(i,(0,-1,0)),\ A(i,(-1,0,0)),\\ &A(i,(0,0,0)),\\ &A(i,(1,0,0)),\ A(i,(0,1,0)),\ A(i,(0,0,1)),\end{aligned} \tag{B.11}$$

compare figure (5.4.1). Hence, the first three entries of equation (B.11) are located in the lower triangular part of A, and the last three entries are located in the upper triangular part of A. Due to the assumptions made in Fourier smoothing analysis, the entries A(i,(p,q,r)) do not depend on the location i in the matrix.

We finally introduce a notation for the sparsity (nonzero) pattern of a matrix, namely

P_L^A : sparsity pattern of lower trangular part of A,

P_U^A : sparsity pattern of upper triangular part of A.

Note that the matrix main diagonal A(i,0) = A(i,(0,0,0)) does not belong to the above patterns.
In the general case where in equation (B.1) $g^{\alpha\beta} \neq 0$ for all α and β, the stencil of the operator A has a 27-point structure, viz.

$$\begin{aligned}&P_L^A = \{(p,q,-1),\ (p,-1,0),(-1,0,0)\},\\ &P_U^A = \{(1,0,0),\ (p,1,0),\ (p,q,1)\},\\ &p,q, \in \{-1,0,1\}.\end{aligned} \tag{B.12}$$

The error matrix B of the lexicographical SLR algorithm along ξ^1-lines is by definition given as

$$(B\Delta\varphi)_i = \sum_j - A(i,j)\Delta\varphi_{i+j},\ j \in P_U^A,\ j \neq (1,0,0), \qquad (B.13)$$

so that $A^* = A+B$ consists of the lower triangular part of A, the main diagonal (0,0,0) and one upper diagonal (1,0,0). As a consequence, this matrix A^* can cheaply be inverted, namely per ξ^1-line.

The ILU/SIP algorithm (described in section 5.4.1) has an error matrix that consists of two parts. The first part B_1 represents the error due to the deletion of the crossderivative terms from the iteration scheme,

$$(B_1\Delta\varphi)_i = - \sum_j A(i,j)\Delta\varphi_{i+j},\ j \in P_L^A \cup P_U^A,\ j \notin P^{\tilde{A}}, \qquad (B.14)$$

so that the stencil of the operator $\tilde{A} = A+B_1$ has a 7-point structure, like (B.7):

$$P^{\tilde{A}} = P_L^{\tilde{A}} \cup (0,0,0) \cup P_U^{\tilde{A}} \qquad ,$$

$$P_L^{\tilde{A}} = \{(0,0,-1),\ (0,-1,0),(-1,0,0)\},$$

$$P_U^{\tilde{A}} = \{(1,0,0),(0,1,0),(0,0,1)\} \qquad . \qquad (B.15)$$

The second part B_2 of the error matrix B results from the incomplete decomposition of $\tilde{A} = A+B_1$, which corresponds to an exact decomposition of $\tilde{A}+B_2$. The matrix B_2 follows (by definition) from

$$LU = \tilde{A} + B_2, \qquad (B.16)$$

with the sparse lower and upper triangular matrices L and U computed by the ILU/SIP incomplete decomposition algorithm.

In the ILU/SIP algorithm described hereafter, the main diagonal of the lower triangular matrix is chosen as $L(i,0) = 1$. The elimination of the lower triangular part of the matrix $\tilde{A}$ is done row by row, while per row the elimination is done in a lexicographical order (see equation (B.11)).

After execution of the ILU/SIP algorithm, the equation $\tilde{A}\Delta\varphi = f$ has been transformed to $U\Delta\varphi = g$, which can be solved by backsubstitution.

```
for lexicographically increasing i do
    for lexicographically increasing $j \in P_L^{\tilde{A}}$ do
        $S \quad := \{k | k \in P_U^{\tilde{A}},\ j+k \notin P^{\tilde{A}}\}$
        $L(i,j) := \tilde{A}(i,j) \,/\, (U(i+j,0) + \alpha \sum_{k \in S} U(i+j,k))$
        $f(i) \quad := f(i) - L(i,j)\, g(i+j)$
        for $k \in P_U^{\tilde{A}}$ do
            if $j+k \in P^{\tilde{A}}$ then
                $\tilde{A}(i,j+k) := \tilde{A}(i,j+k) - L(i,j)\, U(i+j,k)$
            else
                $\tilde{A}(i,0) \quad := \tilde{A}(i,0) + \alpha\, L(i,j)\, U(i+j,k)$
                $\tilde{A}(i,k) \quad := \tilde{A}(i,k) - \alpha\, L(i,j)\, U(i+j,k)$
            endif
        od
    od
    for $j \in P_U^{\tilde{A}} \cup \{(0,0,0)\}$ do $U(i,j) = \tilde{A}(i,j)$ od
od
```

Algorithm B.1 : ILU/SIP(α) - algorithm for incomplete decomposition of $\tilde{A}\Delta\varphi = f$

With the sparsity patterns given by equation (B.15) this ILU/SIP algorithm generates lower and upper triangular matrices L and U that satisfy

$$L(i,j) = \tilde{A}(i,j) \ / \ (U(i+j,0) + \alpha \sum_k U(i+j,k)), \ j \in P_L^{\tilde{A}},$$

$$\text{where } k \in P_U^{\tilde{A}}, \ j+k \notin P^{\tilde{A}};$$

$$U(i,0) = \tilde{A}(i,0) - \sum_j L(i,j) \ U(i+j,-j)$$

$$+ \alpha \sum_j \sum_k L(i,j) \ U(i+j,k),$$

$$\text{where } j \in P_L^{\tilde{A}} \text{ and } k \in P_U^{\tilde{A}}, \ k \neq -j;$$

$$U(i,k) = \tilde{A}(i,k) - \alpha \sum_j L(i,j) \ U(i+j,k) \qquad , \ k \in P_U^{\tilde{A}},$$

$$\text{where } j \in P_L^{\tilde{A}}, \ j \neq -k. \tag{B.17}$$

Under the assumption of constant matrix coefficients (i.e. $L(i,j)$, $U(i,j)$ and $\tilde{A}(i,j)$ do not depend on i) we can solve the above system of equations numerically by iteration, starting from $U(i,j) = \tilde{A}(i,j)$, $j \in P_U^{\tilde{A}} \cup \{(0,0,0)\}$. In case of $\alpha = 0$ the system of equations can also be solved analytically, giving

$$U(i,0) = \tfrac{1}{2} \left(A(i,0) \pm \sqrt{A^2(i,0) - 4 \sum_{j \in P_L^{\tilde{A}}} \tilde{A}(i,j) \ \tilde{A}(i,-j)} \ \right), \tag{B.18}$$

where the plus sign applies in practice. The expressions for $L(i,j)$ and $U(i,k)$ now follow directly from equations (B.17,18).
Finally, the elements of the error matrix B_2 are obtained from $B_2 = LU-\tilde{A}$,

$$B_2(i,j+k) = L(i,j) \ U(i+j,k) \ , \ j \in P_L^{\tilde{A}} \ , \ k \in P_U^{\tilde{A}}, \ j+k \neq 0;$$

$$B_2(i,j) \quad = - \alpha \sum_k L(i,j) \ U(i+j,k) \ , \ j \in P_L^{\tilde{A}},$$

$$\text{where } k \in P_U^{\tilde{A}}, \ k \neq -j;$$

$$B_2(i,k) = -\alpha \sum_j L(i,j)\ U(i+j,k),\ k \in P_U^{\tilde{A}},$$

$$\text{where } j \in P_L^{\tilde{A}},\ j \neq -k;$$

$$B_2(i,0) = \alpha \sum_j \sum_k L(i,j)\ U(i+j,k).$$

$$\text{where } j \in P_L^{\tilde{A}} \text{ and } k \in P_U^{\tilde{A}},\ k \neq -j. \qquad \text{(B.19)}$$

It can be verified that for the given patterns $P_L^{\tilde{A}}$ and $P_U^{\tilde{A}}$ the matrix B is symmetric, i.e. $B(i,j) = B(i,-j)$ if the matrix $\tilde{A}$ is symmetric.

Curriculum vitae of the author

The author was born in Delft on March 16, 1957. After finishing his secondary education at the Christelijk Lyceum Delft in 1975, he studied technical mathematics at the University of Technology Delft. He graduated in December 1980 on the subject "Application of incomplete LU decomposition and Galerkin approximation in multigrid methods for the Cauchy-Riemann equations".

In January 1981 the author joined the National Aerospace Laboratory NLR, where he worked mainly on the subject of calculating the transonic potential flow about an aircraft. The present thesis is an outcome of this work. In 1986 the author was engaged in extending the NLR computer network with a modern supercomputer.

The author is married and has a two years old son.

Summary in Dutch (Samenvatting in het Nederlands)

Een niet-lineaire multi-rooster methode voor drie-dimensionale transsone potentiaalstromingen

Het aerodynamisch ontwerpproces van transport-vliegtuigen vereist betrouwbare rekenmethoden voor het voorspellen van het aerodynamisch gedrag. Dit proefschrift presenteert een rekenmethode voor het numeriek simuleren van de stroming rond een transport-vliegtuig onder kruiscondities. Onder deze condities is de stroming transsoon, dat wil zeggen overwegend subsoon, met lokaal supersone gebieden nabij de vleugel. Deze supersone gebieden worden meestal aan de stroomafwaartse zijde begrensd door een schokgolf, waar doorheen de stroming vrijwel discontinu tot subsone snelheid wordt afgeremd. Als stromingsmodel wordt in dit proefschrift de *volledige potentiaalvergelijking* gebruikt. Deze tweede orde partiële differentiaalvergelijking is van gemengd elliptisch/hyperbolisch karakter en bovendien sterk niet-lineair nabij schokgolven.

De genoemde volledige potentiaalvergelijking wordt op een ruimtelijk rekenrooster gediscretiseerd met een eindige volume methode, waarbij in supersone (hyperbolische) stromingsgebieden een asymmetrie wordt aangebracht door het stroomopwaarts retarderen van massafluxen. Het resulterende stelsel vergelijkingen wordt opgelost met een niet-lineaire multi-rooster methode, waarbij voor het fouten gladstrijkend algoritme ILU/SIP is ontwikkeld. In dit proefschrift wordt dit algoritme in detail beschreven en nader onderzocht. ILU/SIP (Incomplete Lower Upper decomposition/Strongly Implicit Procedure) blijkt in het subsone gedeelte van de stroming sneller te zijn dan de meer gebruikte lijnrelaxatie algoritmen, en is bovendien *onvoorwaardelijk stabiel* in het supersone gedeelte van de stroming. Het algoritme is in hoge mate robuust met betrekking tot extreme maasverhoudingen in het rekenrooster. Het ILU/SIP algoritme wordt in dit proefschrift geanalyseerd voor zowel een elliptisch als een hyperbolisch modelprobleem middels respectievelijk een gladstrijk-analyse en een stabiliteitsanalyse. De waarde van de gladstrijk-analyse wordt in dit proefschrift geëvalueerd aan de hand van een incompressibele stroming in een windtunnel met een gladde hobbel op het bodemvlak. De praktische bruikbaarheid van de methode wordt gedemonstreerd aan de hand van transsone potentiaalstromingen rond twee vleugelmodellen, namelijk de DFVLR-F4 en de ONERA-M6 vleugel. Aangetoond wordt dat a priori optimalisatie van het rekenrooster tot op zekere hoogte noodzakelijk is voor het verkrijgen van een voldoende snel rekenproces. Tevens wordt aangetoond dat de grote waarde van de multi-rooster methode vooral ligt in de relatief lage convergentie niveau's die bij deze methode voor het verkrijgen van een oplossing met acceptabele nauwkeurigheid kunnen worden gespecificeerd.