



**Centrum voor Wiskunde en Informatica**  
Centre for Mathematics and Computer Science

---

P.W. Hemker, B. Koren

Defect correction and nonlinear multigrid  
for the steady Euler equations

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

# Defect Correction and Nonlinear Multigrid for the Steady Euler Equations

P.W. Hemker, B. Koren

*Centre for Mathematics and Computer Science  
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands*

This report presents a set of lecture notes on multigrid methods for the efficient solution of the steady Euler and Navier-Stokes equations.

In these notes we first introduce defect correction as a general framework to solve linear or nonlinear equations. From this starting point different multigrid methods are introduced. Next a concise survey is given of multigrid methods in use for the solution of the steady Euler equations. Finally, based on recent research, efficient solvers for the steady Euler and Navier-Stokes equations are treated. The discretisation of the convection terms is based on a finite volume technique with Osher's approximate Riemann solver. A central finite volume technique is used for the discretisation of the diffusion terms.

In particular, for the Euler equations the use of defect correction in obtaining higher order accurate solutions is demonstrated.

*1980 Mathematics Subject Classification:* 65N30.

*Key Words and Phrases:* defect correction, multigrid methods, steady Euler equations, steady Navier-Stokes equations.

*Note:*

These lecture notes have been presented at the 1988 Lecture Series on Computational Fluid Dynamics at the Von Karman Institute, Rhode Saint Genese, Belgium.

Note NM-N8801

Centre for Mathematics and Computer Science  
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

## Contents

1 The Defect Correction Principle	1
1 Introduction	1
2 Two elementary correction processes	2
3 Convergence of the basic defect correction processes	4
1 Convergence for the DCPA	4
2 Convergence for the DCPB	5
3 Convergence for the DCPL	5
4 More elaborate versions of the principle	6
1 Non-stationary DCPs	6
2 Combination of DCPs	6
3 Iterative application of DCPs	7
4 Recursive application of DCPs	8
5 Generalisation for nonlinear problems	8
5 Relaxation processes	9
6 Theorem on defect correction for discretisations	9
2 Multigrid algorithms	11
1 Introduction	11
2 The two-level algorithm	12
3 The multi-level algorithm	13
4 The full multigrid method (FMG)	15
3 Multigrid approaches for compressible flow	16
1 The equations of compressible flow	16
1 The Navier-Stokes equations	17
2 The Euler equations	17
2 The discretisations	18
3 The multiple grid methods	20
1 Methods based on Lax-Wendroff type time stepping	20
2 Methods based on semidiscretisation and timestepping	22
3 Fully implicit methods	23
4 Multigrid for the first-order discretisation of the Euler equations	24
1 The first-order finite-volume discretisation	24
2 Osher's approximate Riemann solver	25
3 The numerical flux at the boundary	28
4 The linearisation of Osher's scheme	29
5 Multigrid iteration	30
1 A nested sequence of Galerkin discretisations	31
2 Multigrid strategy	32
3 Relaxation	32
4 Initial estimates	33
6 Conclusion	33
5 Defect correction for higher-order Euler computations	34
1 Second-order discretisation	34
2 The solution of the second-order discrete system	35
3 The complete multigrid algorithm	37
6 Solution of the Navier Stokes equations	38
1 Introduction	38
2 The discretisation method	38
1 Evaluation of convective fluxes	38
2 Evaluation of diffusive fluxes	39
3 Solution method	39
4 Numerical results	40
Acknowledgement	41
References	41
Figures	45



## 1. THE DEFECT CORRECTION PRINCIPLE

### 1.1 INTRODUCTION

Many problems in numerical mathematics can be cast into the form of an equation

$$Fz = y.$$

Here  $z \in D$  is an unknown quantity or function; a right-hand side  $y \in \hat{D}$  and a bijective mapping  $F: D \subset E \rightarrow \hat{D} \subset \hat{E}$  are given;  $E$  and  $\hat{E}$  are linear spaces. The element  $z \in D$  has to be found such that the equation  $Fz = y$  is satisfied. We may not be able to solve the above mentioned equation directly, because this exceeds our computational capabilities, whereas on the other hand we may be able to solve simpler equations that are all similar to the previous equation:

$$\tilde{F} \tilde{z} = \tilde{y},$$

for some approximation  $\tilde{F}: D \rightarrow \hat{D}$  of the operator  $F$  and for arbitrary  $\tilde{y} \in \hat{D}$ . Sometimes this yields the possibility to solve the original equation by means of an iterative process known as a Defect Correction Process (DCP).

DCPs are based on the following idea:

- let an initial approximation  $z_0$  for the solution to the original equation be given,
- consider the *defect*  $d_0 := F(z_0) - y$  of this initial approximation as a quantity which indicates to what extent the problem has (not) been solved,
- use this information in a simplified version of the problem, i.e. take the approximate operator  $\tilde{F}$ , to obtain an appropriate *correction* quantity,
- apply this correction to the initial approximation to obtain a new and hopefully better approximation.

The above process may be repeated, using the newly obtained approximation as a new ‘initial’ approximation.

A few instances of the basic principle is well known. We mention two examples:

**EXAMPLE 1.1.1 (NEWTON’S METHOD)** We are interested in computing a zero of the nonlinear function  $F$ . The equation to be solved is then given by

$$F(z) = 0.$$

Let an initial approximation  $z_0$  for the solution be given. A (hopefully convergent) sequence  $z_1, z_2, \dots$  of approximations for the solution is then generated by

$$z_{i+1} = z_i - (F'(z_i))^{-1} d_i, \quad i = 0, 1, 2, \dots,$$

where the defect  $d_i$  is defined by

$$d_i = F(z_i) - 0 = F(z_i).$$

Note that in each step of this iterative process a different approximation  $\tilde{F}_i$  for the function  $F$  is used. In the  $i$ -th step, this approximating function  $\tilde{F}_i$  is the local linearisation  $F'(z_i)$ , defined by:

$$F(z) = F(z_i) + F'(z_i)(z - z_i) + \mathcal{O}(\|z - z_i\|^2).$$

**EXAMPLE 1.1.2.** A second example of a DCP is the iterative refinement of a linear system. Usually one obtains an approximate solution  $z_0$  for the linear system  $Az = y$  by computing a decomposition  $A = LU$  ( $L$  and  $U$  lower resp. upper triangular matrices), and then solving the two triangular systems  $Lw = y, Uz = w$  by substitution. The approximation  $z_0$  will be contaminated by rounding errors that affect the matrix-decomposition and the solution of the triangular systems;  $z_0$  can be improved by the

following DCP ( *iterative refinement* ):

$$z_{i+1} = z_i - e_i, \quad i = 0, 1, 2, \dots,$$

with  $e_i$  the solution of the system

$$LUe_i = d_i; \quad d_i = Az_i - y.$$

## 1.2 TWO ELEMENTARY CORRECTION PROCESSES

We consider the equation

$$Fz = y, \tag{1.1}$$

where  $F: D \subset E \rightarrow \hat{D} \subset \hat{E}$  is a bijective, continuous and generally nonlinear operator;  $E$  and  $\hat{E}$  are Banach spaces. The domain  $D$  and the range  $\hat{D}$  are closed subsets depending on  $F$ ;  $\hat{D}$  contains an appropriate neighbourhood of  $y$  such that for every  $\tilde{y} \in \hat{D}$  there exists, in  $D$ , exactly one solution of  $Fz = \tilde{y}$ . The solution of the given equation (1.1) is denoted by  $z^*$ .

We call the problem of finding this  $z$  such that  $Fz = \tilde{y}$  (for a given  $\tilde{y} \in \hat{D}$ ) a *neighbouring problem*. In order to introduce the Defect Correction Processes to solve the equation (1.1), we make the following assumptions:

A1. We assume that the *defect*

$$d(\tilde{z}) := F\tilde{z} - \tilde{y}$$

can be evaluated for all approximate solutions  $\tilde{z} \in D$  of neighbouring problems  $F\tilde{z} = \tilde{y}$ .

A2. Furthermore, we assume that we can readily solve the *approximate problem*

$$\tilde{F}z = \tilde{y}, \tag{1.2}$$

for all  $\tilde{y} \in \hat{D}$ , i.e. we assume that we can evaluate the solution operator  $\tilde{G}$  of (1.2). In other words, we assume the existence of

$\tilde{G}: \hat{D} \rightarrow D$ , an *approximate inverse* of  $F$  such that (in some appropriate sense)

$$\tilde{G}F\tilde{z} \approx \tilde{z} \quad \text{for } \tilde{z} \in D$$

and

$$F\tilde{G}\tilde{y} \approx \tilde{y} \quad \text{for } \tilde{y} \in \hat{D}.$$

Let us assume that we know some approximation  $\tilde{z} \in D$  for  $z^*$  and that we have computed its defect

$$d(\tilde{z}) := F\tilde{z} - y = F\tilde{z} - Fz^*.$$

In the general (nonlinear) case, there are two ways to use this information for the computation of a (hopefully better) approximation  $\tilde{z}$  by means of solving problems of type (1.2), see Fig.1.1:

(A) The error  $\Delta z$  satisfies the relation:

$$\Delta z := \tilde{z} - z^* = F^{-1}(y + d(\tilde{z})) - F^{-1}y.$$

A new approximation  $\tilde{\tilde{z}}$  can be found by:

$$\begin{aligned} \tilde{\tilde{z}} &:= \tilde{z} - \Delta z = \tilde{z} - [\tilde{G}(y + d(\tilde{z})) - \tilde{G}y] = \\ &= \tilde{z} - \tilde{G}F\tilde{z} + \tilde{G}y = (I - \tilde{G}F)\tilde{z} + \tilde{G}y. \end{aligned}$$

We notice that  $\tilde{\tilde{z}}$  is a sum (in  $E$ ) of terms in  $D$ .

(B) The second approach is dual to (A) in the sense that (B) uses the addition in  $\hat{D}$ , whereas (A) uses the addition in  $D$ . Suppose we have  $l \in \hat{D}$ . We then may consider  $\tilde{z} = F^{-1}l$  as an approximation for  $z^*$ . The defect for  $\tilde{z}$  is:

$$d(\tilde{z}) = F\tilde{z} - y = l - y.$$

Introducing  $\tilde{l} = l - d(\tilde{z})$ , we obtain the exact solution  $z^*$  by  $z^* = F^{-1}\tilde{l}$ . Because we cannot invert  $F$  directly, we use the approximate inverse  $\tilde{G}$  instead of  $F^{-1}$ . We then obtain:

$$\begin{aligned}\tilde{z} &= \tilde{G}\tilde{l}, & d(\tilde{z}) &= F\tilde{G}\tilde{l} - y, \\ \tilde{l} &= \tilde{l} - d(\tilde{z}) = (I - F\tilde{G})\tilde{l} + y.\end{aligned}$$

The iterative use of these basic defect correction processes leads to the following algorithms:

$$(\text{DCPA}) \quad \begin{cases} z_0 &= \tilde{G}y, \\ z_{i+1} &= (I - \tilde{G}F)z_i + \tilde{G}y, \end{cases}$$

and

$$(\text{DCPB}) \quad \begin{cases} l_0 &= y; \\ l_{i+1} &= (I - F\tilde{G})l_i + y. \end{cases}$$

With the latter iteration we define approximations  $z_i$  by

$$z_i = \tilde{G}l_i.$$

REMARKS:

- (i) In (DCPA) or (DCPB) we formulate both the iterative process and the (standard) initial approximant.
- (ii) It is essential that  $\tilde{G}$  is relatively simple, i.e. it is much easier to find a solution for (1.2) than for (1.1).
- (iii) It is the existence of the approximate inverse  $\tilde{G}$  which is essential, *not* the existence of the approximate operator  $\tilde{F}$ .

The mapping  $\tilde{G}: \hat{D} \rightarrow D$  is injective if and only if its left inverse  $\tilde{F}$  exists; similarly the right inverse  $\tilde{F}: D \rightarrow \hat{D}$  of  $\tilde{G}$  exists if and only if  $\tilde{G}$  is surjective. In general the mapping  $\tilde{G}$  does not need to be linear and is neither necessarily injective nor surjective.

If  $\tilde{G}$  is injective, its left inverse  $\tilde{F}$  exists and we can write DCPB as

$$(\text{DCPB}^*) \quad \begin{cases} \tilde{F}z_0 &= y, \\ \tilde{F}z_{i+1} &= (\tilde{F} - F)z_i + y, \end{cases}$$

or, equivalently,

$$\begin{cases} z_0 &= \tilde{G}y, \\ z_{i+1} &= \tilde{G}[(\tilde{F} - F)z_i + y]. \end{cases}$$

In some applications, the operator  $\tilde{F} - F$  is much simpler to evaluate than  $F$ , so that there is an

advantage in using this formulation.

In case of a linear operator  $\tilde{G}$  we can write both DCPA and DCPB as:

$$(DCPL) \begin{cases} z_0 &= \tilde{G}y \\ z_{i+1} &= z_i - \tilde{G}(Fz_i - y) = z_i - \tilde{G}d(z_i). \end{cases}$$

DEFINITION 1.2.1. A mapping  $f: E \rightarrow \hat{E}$  is called *affine* if there exists a constant element  $c \in \hat{E}$  such that  $f(\cdot) - c: X \rightarrow Y$  is a linear mapping.  $\square$

THEOREM 1.2.1. If  $\tilde{G}$  is an affine mapping, then the sequences  $\{z_i\}$  in DCPA and  $\{z_i\}$  in DCPB are identical.  $\square$

The proof of this theorem is left as an exercise. In the proof we use the property of an affine mapping  $f$  that  $f(x - y + z) = f(x) - f(y) + f(z)$ .

### 1.3 CONVERGENCE OF THE BASIC DEFECT CORRECTION PROCESSES

In the following  $F: D \subset E \rightarrow \hat{D} \subset \hat{E}$  is a general nonlinear operator.

DEFINITION 1.3.1.  $F$  is called *bounded* if bounded subsets of  $D$  are mapped onto bounded subsets in  $\hat{D}$ .

$F$  is called *Lipschitz* if

$$\exists k > 0 \forall x, y \in D \quad \|Fx - Fy\|_{\hat{E}} \leq k \|x - y\|_E.$$

The *Lipschitz constant*  $\|F\|$  is defined by

$$\|F\| = \sup_{x, y \in D, x \neq y} \frac{\|Fx - Fy\|_{\hat{E}}}{\|x - y\|_E},$$

$F$  is called a *strict contraction* if  $\|F\| < 1$ .

Note: Apparently

$$\|Fx - Fy\|_{\hat{E}} \leq \|F\|_{D \subset E \rightarrow \hat{D} \subset \hat{E}} \|x - y\|_E$$

and for a linear operator  $F$

$$\|F\| = \|F\|.$$

DEFINITION 1.3.2. An iterative process  $z^{(i+1)} = H(z^{(i)}, z^{(i-1)}, \dots)$  has a *fixed point*  $z^*$  if  $z^* = H(z^*, z^*, \dots)$ .

#### 1.3.1 Convergence for the DCPA.

For DCPA we have  $z_{i+1} - z^* = (I - \tilde{G}F)z_i + \tilde{G}Fz^* - z^*$ , hence

$$z_{i+1} - z^* = (I - \tilde{G}F)z_i - (I - \tilde{G}F)z^*.$$

We define the *amplification operator of the error* in DCPA by

$$M_A = I - \tilde{G}F. \tag{1.3}$$

The exact solution  $z^*$  of (1.1) is a fixed point of the iteration DCPA, i.e.  $z^* = (I - \tilde{G}F)z^* + \tilde{G}y$ . Moreover, for any fixed point  $\hat{z}$  of DCPA we have:

$$\begin{aligned} \hat{z} &= (I - \tilde{G}F)\hat{z} + \tilde{G}y, \quad \text{hence} \\ \tilde{G}F\hat{z} &= \tilde{G}y = \tilde{G}Fz^* \end{aligned}$$

As a direct consequence of this we find the following

THEOREM 1.3.1. If DCPA has a fixed point  $\hat{z} \in \hat{D}$  with  $F\hat{z} \in \hat{D}$  and if  $\tilde{G}$  is injective, then  $F\hat{z} = y$ , i.e. then  $\hat{z}$  is a solution of (1.1).  $\square$

The convergence of DCPA clearly depends on the *contractivity* of the amplification operator  $M_A = I - GF$ . We formulate the following

THEOREM 1.3.2. Let  $M_A: D \rightarrow D$  be a contraction and let  $\tilde{G}: \hat{D} \rightarrow D$  be injective. Then DCPA converges to the solution  $z^*$  of (1.1).  $\square$

REMARK If  $\tilde{G}$  is not injective, the solution  $z^*$  of (1.1) and the fixed point  $\hat{z}$  of DCPA are mapped by  $\tilde{G}F$  onto the same element of  $G(D)$ , although we have not necessarily  $F\hat{z} = y = Fz^*$ . In other words: if  $\tilde{G}$  is not injective,  $\tilde{G}$  defines equivalence classes in  $\hat{D}$ , viz. the classes of points that are all mapped to the same point of  $D$ . Now  $F\hat{z}$  and  $Fz^*$  are elements of the same equivalence class.

### 1.3.2 Convergence for the DCPB.

For DCPB we have, with  $\tilde{G}l^* = z^*$ ,

$$l_{i+1} - l^* = (I - F\tilde{G}) l_i - (I - F\tilde{G}) l^*.$$

We define the *amplification operator of the residual* in DCPB to be

$$M_B = I - F\tilde{G}. \quad (1.4)$$

For any fixed point  $\hat{l}$  of DCPB we have:

$$\begin{aligned} \hat{l} &= (I - F\tilde{G}) \hat{l} + y, \quad \text{hence} \\ F\tilde{G} \hat{l} &= y = Fz^*. \end{aligned}$$

As direct consequence of this, we find:

THEOREM 1.3.3. If DCPB has a fixed point  $\hat{l} \in \hat{D}$  then  $\tilde{G}\hat{l}$  is a solution of (1.1) in  $\tilde{G}(\hat{D})$ . Because we have assumed  $F$  to be injective, we also know that  $\tilde{G}\hat{l}$  is the unique solution of (1.1).  $\square$

REMARK If  $\tilde{G}: \hat{D} \rightarrow D$  is not surjective, it may occur that solutions  $z^*$  of (1.1) have the property that  $z^* \notin \tilde{G}(\hat{D})$  and hence no  $\hat{l} \in \hat{D}$  exists such that  $\tilde{G}\hat{l} = z^*$ . In that case no fixed point  $\hat{l} \in \hat{D}$  can exist.

The convergence of DCPB depends on the contractivity of the amplification operator  $M_B = I - F\tilde{G}$ .

THEOREM 1.3.4. Let  $M_B: \hat{D} \rightarrow \hat{D}$  be a contraction on the Banach space  $\hat{D}$ . Then DCPB converges to an element  $l^* \in \hat{D}$  such that  $z^* = \tilde{G}l^*$  is the solution of (1.1).  $\square$

### 1.3.3 Convergence for the DCPL.

If  $G$  is linear we have:

$$\begin{aligned} z_{i+1} - z^* &= z_i - \tilde{G}Fz_i + \tilde{G}Fz^* - z^* = \\ &= z_i - z^* - \tilde{G}(Fz_i - Fz^*) = \\ &= (I - \tilde{G}F)z_i - (I - \tilde{G}F)z^*. \end{aligned}$$

If  $F$  is linear as well, the amplification operator  $M_L = I - \tilde{G}F$  of the error and the amplification operator  $\bar{M}_L = I - FG$  of the residual are both linear operators and for the convergence of the iteration we can consider  $\|M_L\|$ ,  $\|\bar{M}_L\|$ ,  $\rho(M_L) = \lim_{n \rightarrow \infty} (\|M_L^n\|)^{1/n}$  and  $\rho(\bar{M}_L)$ . Notice

that  $\rho(M_L) = \rho(\overline{M}_L)$ .

#### 1.4 MORE ELABORATE VERSIONS OF THE PRINCIPLE

In this section we extend the idea of the defect correction process in several ways. First, we allow different approximate inverses to serve in one iteration process, and we consider the process obtained when a fixed combination of approximate inverses is used repeatedly in a defect correction process. Secondly, it is possible to substitute different operators  $F_i$  for  $F$  during iteration. Further, we describe the iterative and the recursive application of the defect correction principle.

##### 1.4.1. Non-stationary DCPs.

We can use different approximate inverses in each iteration step and thus obtain non-stationary DCPs. Then the iteration steps of DCPA and DCPB read respectively

$$z_{i+1} = (I - \tilde{G}_i F)z_i + \tilde{G}_i y, \quad (1.5)$$

and

$$l_{i+1} = (I - F\tilde{G}_i)l_i + y. \quad (1.6)$$

In this way we are able to adapt the approximate inverse during the iteration and we can try to find proper sequences  $\{\tilde{G}_i\}$  to accelerate the convergence of the iteration. We mention the following three examples: (i), (ii) and (iii).

$$(i) \quad \tilde{G}_i = \tilde{G}(z_i).$$

Here the approximate inverse depends on the last iterand computed. This is the case e.g. in Newton's method for the solution of the nonlinear equations, where  $\tilde{G}(z) = (F'(z))^{-1}$ , with  $F'$  the Fréchet derivative of the operator  $F$  in the problem (1.1). See example 1.1.1 .

$$(ii) \quad \tilde{G}_i = \tilde{G}(\omega_i).$$

The approximate inverse depends on a single real parameter  $\omega_i$ . This is the case e.g. in non-stationary relaxation processes for the solution of linear systems. The value  $\omega_i$  can be taken from a fixed sequence of values or it can be computed adaptively during the iteration process.

$$(iii) \quad \tilde{G}_i \in \{\tilde{G}_1, \tilde{G}_2\}.$$

Here, in each iteration step the approximate inverse is chosen from a set of two (or possibly more) fixed approximate inverses. This is the case e.g. in Brakhage's[5] and Atkinson's[3] methods for the solution of Fredholm integral equations of the second kind.

##### 1.4.2 Combination of DCPs.

We now assume that  $F$  is linear and we consider a fixed combination of two *linear* approximate inverse operators  $\tilde{G}$  and  $\tilde{\tilde{G}}$ . Then we combine two iteration steps in the non-stationary DCPA in which, in turn, one or the other of two approximate inverses is used. These two iteration steps

$$z_{i+1/2} = (I - \tilde{G}F)z_i + \tilde{G}y$$

and

$$z_{i+1} = (I - \tilde{\tilde{G}}F)z_{i+1/2} + \tilde{\tilde{G}}y$$

combine into a single iteration step of the form

$$\begin{aligned} z_{i+1} &= (I - \tilde{\tilde{G}}F)(I - \tilde{G}F)z_i + (\tilde{\tilde{G}} - \tilde{\tilde{G}}F\tilde{G} + \tilde{G})y = \\ &= (I - (\tilde{\tilde{G}} - \tilde{\tilde{G}}F\tilde{G} + \tilde{G})F)z_i + (\tilde{\tilde{G}} - \tilde{\tilde{G}}F\tilde{G} + \tilde{G})y. \end{aligned}$$

This is again an iteration step of type (DCPA) with the approximate inverse

$$\hat{G} = \tilde{G} - \tilde{G}F\tilde{G} + \tilde{G}.$$

The amplification operator of this new process is the product of the amplification operators of the elementary processes:

$$M = I - \hat{G}F = (I - \tilde{G}F)(I - \tilde{G}F).$$

Thus, the combination of two different DCP-steps (with linear  $F$  and  $\tilde{G}$ ) can be seen as one 'big' DCP-step.

Still assuming that  $F$  and  $\tilde{G}$  are linear operators, we now consider  $\sigma$  applications of the same approximate inverse. Using DCPA, this can be described in matrix notation as follows:

$$\begin{aligned} \begin{bmatrix} z_{i+1} \\ y \end{bmatrix} &= \begin{bmatrix} I - \tilde{G}F & \tilde{G} \\ 0 & I \end{bmatrix}^{\sigma} \begin{bmatrix} z_i \\ y \end{bmatrix} = \\ &= \begin{bmatrix} (I - \tilde{G}F)^{\sigma} & \sum_{m=0}^{\sigma-1} (I - \tilde{G}F)^m \tilde{G} \\ 0 & I \end{bmatrix} \begin{bmatrix} z_i \\ y \end{bmatrix}. \end{aligned}$$

Thus, we see that these  $\sigma$  applications of the same DCPA-step lead to the following process:

$$z_{i+1} = (I - \tilde{G}F)^{\sigma} z_i + \sum_{m=0}^{\sigma-1} (I - \tilde{G}F)^m \tilde{G}y.$$

The relation

$$\sum_{m=0}^{\sigma-1} (I - \tilde{G}F)^m \tilde{G} = [I - (I - \tilde{G}F)^{\sigma}]F^{-1},$$

allows us to consider the above process as a new DCPA with amplification operator of the error

$$M = (I - \tilde{G}F)^{\sigma}$$

and approximate inverse

$$\hat{G} = \sum_{m=0}^{\sigma-1} (I - \tilde{G}F)^m \tilde{G} = [I - (I - \tilde{G}F)^{\sigma}]F^{-1}.$$

#### 1.4.3 Iterative application of DCPs.

We will now pay attention to a possibility mentioned before, viz. the substitution of different operators  $F_i$  for  $F$  during iteration. This is important if we study discretised continuous problems. We consider all (discrete)  $F_i$  as approximations to one (continuous) 'target' operator  $F^*$ . As long as the approximate solution is not a very good approximation, i.e. in the beginning of the iteration, we take operators  $\{F_i\}$  that are simple to evaluate. For increasing  $i$  they become better approximations and converge to  $F^*$  in some sense.

If we apply this technique, we approximately solve a sequence of problems  $\{P_k\}_{k=1,2,\dots}$  of the form

$$P_k : \quad F_k z = y_k,$$

where the approximate solution of  $(P_{k-1})$  is used as a starting value for the iteration of  $(P_k)$ .

One possible application is to select operators  $\{F_i\}$  that are discrete approximations of *increasingly*

higher order to an analytic operator  $F$ . The approximate inverse  $\tilde{G} = F_0^{-1}$  may be constant during the process.

Another example is the *Mesh Continuation Method* in which  $\{F_i\}$  are discretisations on *finer and finer meshes* of an analytic operator  $F$ . When combined with a multigrid technique for the solution of the discrete problems, this is called Nested Iteration[16] or Full Multigrid (FMG)[7].

#### 1.4.4 Recursive application of DCPs.

The evaluation of the approximate inverse operator  $\tilde{G}_i$  implies the solution of an equation which is (essentially) of a simpler type than the original equation. However, also this simpler equation may be of a kind that we want to solve by means of a DCP. For this we need an even simpler equation to solve, etc.. Thus, the execution of a single iteration step may imply the activation of a new (simpler to solve) DCP. In this way we can construct a recursive sequence of DCPs in which very simple equations are to be solved on the lowest level of recursion. Multigrid iteration is an example of this principle.

#### 1.4.5 Generalisation for nonlinear problems.

A generalisation of DCPA, specially for nonlinear problems, may be introduced via DCPL. Let us first consider the process DCPA and suppose that  $\tilde{G}$  is differentiable. Since  $\tilde{G}'(\tilde{y})$ ,  $\tilde{y} \in \hat{D}$ , is a linear operator we can use it to obtain a linear approximation (linearisation) of the operator  $\tilde{G}$ . We then simply use DCPL with approximate inverse  $\tilde{G}'(\tilde{y})$ :

$$z_{i+1} = z_i - \tilde{G}'(\tilde{y})(Fz_i - y).$$

For  $\mu \in \mathbb{R}$ ,  $\mu \neq 0$  and  $u \in \hat{E}$  we have:

$$\tilde{G}'(\tilde{y})u = \mu \tilde{G}'(\tilde{y}) \frac{u}{\mu} \approx \mu \tilde{G}(\tilde{y} + u/\mu) - \mu \tilde{G}(\tilde{y}). \quad (1.7)$$

Because, in general, the Fréchet derivative  $\tilde{G}'(\tilde{y})$  is not available for computation, we replace (using the above with  $u = Fz_i - y$ ) the form  $\tilde{G}'(\tilde{y})(Fz_i - y)$  by

$$\mu \tilde{G}(\tilde{y} + (Fz_i - y)/\mu) - \mu \tilde{G}(\tilde{y}).$$

We then obtain the following DCP:

$$(DCPN) \quad \begin{cases} z_0 &= \tilde{G}y \\ z_{i+1} &= z_i - \mu \tilde{G}(\tilde{y} + (Fz_i - y)/\mu) + \mu \tilde{G}(\tilde{y}). \end{cases}$$

#### REMARKS

- (i) In this new defect correction process there are two free parameters  $\mu$  and  $\tilde{y}$  to choose. For the choice  $\tilde{y} = y$  and  $\mu = 1$ , DCPN coincides with DCPA.
- (ii) For a large enough  $\mu$ , we may guarantee that for any finite defect  $Fz_i - y$ , the operator  $\tilde{G}$  is evaluated in a sufficiently small neighbourhood of  $\tilde{y}$ .
- (iii) In the general case (i.e. for arbitrary values of  $\mu$  and  $\tilde{y}$ ), the solution  $z^*$  of (1.1) is a fixed point of DCPN. The converse may also be true:

**THEOREM 1.4.1.** Let  $\hat{z}$  be a fixed point of DCPN and let  $\tilde{G}: \hat{D} \rightarrow D$  be injective. Then  $\hat{z}$  is the solution of (1.1).  $\square$



### 1.5 RELAXATION PROCESSES

All stationary fully consistent iterative methods of degree one (cf. [71]) for the solution of linear systems  $Ax = b$  can be written as

$$x_{i+1} = x_i - P(Ax_i - b),$$

where  $P$  is a non-singular matrix. These iterative methods are defect correction processes of type  $A$  with approximate inverse  $G = P$ . Because  $P$  is linear, they are equivalent to the corresponding DCPBs. We introduce the following notation:

$$A = L + D + U;$$

$L$  is a strictly lower triangular matrix,  $D$  a diagonal matrix,  $D = \text{diag}(A)$ , and  $U$  is a strictly upper triangular matrix. Using this notation, the following table summarises some possible choice for  $\tilde{F} = P^{-1}$ , together with the name of the corresponding iterative method.

$\tilde{F} = P^{-1}$	Name of the method	Remarks
$D$	J (Jacobi)	
$\omega^{-1}D$	JOR	$\omega > 0$
$D + L$	GS (Gauss Seidel)	
$\omega^{-1}D + L$	SOR	$\omega > 0$
$P^{-1}I$	RF (Stationary Richardson)	$P$ scalar
$P^{-1}$	GRF (Generalised RF)	$P$ non-singular diagonal matrix

Other relaxation processes such as ILU-relaxation and ILLU-relaxation[19,21], where approximate LU-decompositions are used as approximate operators  $\tilde{F}$ , are also of the type DCPL.

### 1.6 THEOREM ON DEFECT CORRECTION FOR DISCRETISATIONS

We consider the continuous problem

$$Lu = f, \quad (1.8)$$

where  $L: E^\alpha \rightarrow \hat{E}^\alpha$  is a linear operator, and  $\{E^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$ ,  $\{\hat{E}^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$  are scales of Banach spaces. (i.e. nested sequences of Banach spaces with stronger norms  $\|\cdot\|_\alpha$  for larger values of  $\alpha$ ). The solution of (1.8) is called  $\hat{u}$ . We consider a lower order accurate discretisation of (1.8):

$$L_h u_h = \bar{R}_h f \quad (1.9)$$

and a higher order accurate discretisation:

$$L_h^* u_h = \bar{R}_h^* f \quad (1.10)$$

where  $L_h, L_h^*: E_h^\alpha \rightarrow \hat{E}_h^\alpha$  are linear operators, and both  $\{E_h^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$  and  $\{\hat{E}_h^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$  are scales of (discrete) Banach spaces corresponding with  $\{E^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$  and  $\{\hat{E}^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$  respectively.

The defect correction process

$$\begin{cases} L_h u_h^{(1)} = \bar{R}_h f, \\ L_h u_h^{(i+1)} = (L_h - L_h^*) u_h^{(i)} + \bar{R}_h^* f, \quad i = 1, 2, \dots, \end{cases} \quad (1.11)$$

is used to approximate the solution of (1.10). The following theorem shows the order of accuracy of the intermediate results.

**THEOREM 1.6.1** Let the problem (1.8) be discretised by (1.9) and (1.10), and let

- 1)  $L_h$  be stable,
  - 2)  $L_h$  be consistent of order  $p$ ,
  - 3)  $L_h^*$  be consistent of order  $p^*$ ,
  - 4)  $L_h$  and  $L_h^*$  be relatively consistent of order  $p$ ,
- Then the iterands in (1.11) satisfy the error estimate

$$\|u_h^{(i)} - R_h \hat{u}\|_{E_h^*} \leq Ch^{\min(p^*, ip)} \|\hat{u}\|_{E^{\omega+\beta_i}},$$

with  $\beta_i = \min(p^*, ip)$  and  $C$  independent of  $h$  and  $\omega, i$  such that  $\omega, \omega + ip \in [\alpha_0, \alpha_1]$ .

PROOF (sketch)

We first put the conditions 1) - 4) in a more explicit form:

- 1)  $\exists C_1$  independent of  $h$  :  $\|L_h^{-1}\|_{E_h^* \leftarrow E_h^*} \leq C_1$
- 2)  $p$  is a real number for which

$$\|\bar{R}_h L - L_h R_h\|_{E_h^* \leftarrow E^{\omega+p}} \leq C_2 h^p,$$

$C_2$  independent of  $h$ . This implies that

$$\|\bar{R}_h L \hat{u} - L_h R_h \hat{u}\|_{E_h^*} = \|\bar{R}_h f - L_h R_h \hat{u}\|_{E_h^*} \leq C_2 h^p \|\hat{u}\|_{E^{\omega+p}}.$$

- 3)  $p^*$  is a real number for which

$$\|\bar{R}_h^* L - L_h^* R_h\|_{E_h^* \leftarrow E^{\omega+p^*}} \leq C_3 h^{p^*},$$

$C_3$  independent of  $h$ . This implies that

$$\|\bar{R}_h^* L \hat{u} - L_h^* R_h \hat{u}\|_{E_h^*} = \|\bar{R}_h^* f - L_h^* R_h \hat{u}\|_{E_h^*} \leq C_3 h^{p^*} \|\hat{u}\|_{E^{\omega+p^*}}.$$

- 4)

$$\|L_h - L_h^*\|_{E_h^* \leftarrow E^{\omega+p^*}} \leq C_4 h^p,$$

$C_4$  independent of  $h$ .

Now we are in the position to prove the theorem by induction:

$$\begin{aligned} u_h^{(1)} - R_h \hat{u} &= L_h^{-1} \bar{R}_h f - R_h \hat{u} \\ &= L_h^{-1} [\bar{R}_h L - L_h R_h] \hat{u}. \end{aligned}$$

Hence

$$\begin{aligned} \|u_h^{(1)} - R_h \hat{u}\|_{E_h^*} &\leq \\ \|L_h^{-1}\|_{E_h^* \leftarrow E_h^*} \|\bar{R}_h L - L_h R_h\|_{E_h^* \leftarrow E^{\omega+p}} \|\hat{u}\|_{E^{\omega+p}}. \end{aligned}$$

With the use of 1) ( $\alpha = \omega$ ) and 2) ( $\alpha = \omega$ ) we find

$$\|u_h^{(1)} - R_h \hat{u}\|_{E_h^*} \leq C_1 C_2 h^p \|\hat{u}\|_{E^{\omega+p}}.$$

Because (1.10) is a discretisation of higher order than (1.9), we have  $p^* \geq p$  and it follows that

$$\|u_h^{(1)} - R_h \hat{u}\|_{E_h^*} \leq Ch^{\min(p^*, p)} \|\hat{u}\|_{E^{\omega+\min(p^*, p)}}.$$

Thus, the theorem is proved for  $i = 1$ .

Now, suppose that the theorem holds for some  $i \geq 1$ . We will prove it for  $i + 1$ :

$$\begin{aligned} u_h^{(i+1)} - R_h \hat{u} &= L_h^{-1} [(L_h - L_h^*) u_h^{(i)} + \bar{R}_h^* f] - R_h \hat{u} \\ &= u_h^{(i)} - R_h \hat{u} - L_h^{-1} [L_h^* u_h^{(i)} - \bar{R}_h^* L \hat{u}] \\ &= [I - L_h^{-1} L_h^*] [u_h^{(i)} - R_h \hat{u}] - L_h^{-1} [L_h^* R_h - \bar{R}_h^* L] \hat{u} \end{aligned}$$

$$= L_h^{-1} [L_h - L_h^*] [u_h^{(i)} - R_h \hat{u}] - L_h^{-1} [L_h^* R_h - \bar{R}_h^* L] \hat{u}.$$

It now follows that

$$\begin{aligned} & \|u_h^{(i+1)} - R_h \hat{u}\|_{E_h^*} \\ & \leq \|L_h^{-1} [L_h - L_h^*] [u_h^{(i)} - R_h \hat{u}]\|_{E_h^*} + \|L_h^{-1} [L_h^* R_h - \bar{R}_h^* L] \hat{u}\|_{E_h^*} \\ & \leq \|L_h^{-1}\|_{E_h^* \leftarrow \hat{E}_h^*} \|L_h - L_h^*\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} \|u_h^{(i)} - R_h \hat{u}\|_{E_h^{*,p}} + \\ & \quad \|L_h^{-1}\|_{E_h^* \leftarrow \hat{E}_h^*} \|L_h^* R_h - \bar{R}_h^* L\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} \|\hat{u}\|_{E_h^{*,p}}. \end{aligned}$$

Using 1) ( $\alpha = \omega$ ), 3) ( $\alpha = \omega$ ) and 4) ( $\alpha = \omega$ ) we find

$$\begin{aligned} & \|u_h^{(i+1)} - R_h \hat{u}\|_{E_h^*} \\ & \leq C_1 C_4 h^p \|u_h^{(i)} - R_h \hat{u}\|_{E_h^{*,p}} + C_1 C_3 h^{p^*} \|\hat{u}\|_{E_h^{*,p}}. \end{aligned}$$

Application of the induction hypothesis leads to

$$\begin{aligned} & \|u_h^{(i+1)} - R_h \hat{u}\|_{E_h^*} \\ & \leq C_1 C_4 h^p C h^{\min(p^*, ip)} \|\hat{u}\|_{E_h^{*,p}, \beta_i} + C_1 C_3 h^{p^*} \|\hat{u}\|_{E_h^{*,p}} \\ & \leq C_1 C_4 C h^{\min(p^*, (i+1)p)} \|\hat{u}\|_{E_h^{*,\beta_{i+1}}} + C_1 C_3 h^{\min(p^*, (i+1)p)} \|\hat{u}\|_{E_h^{*,\beta_{i+1}}} \\ & \leq C h^{\min(p^*, (i+1)p)} \|\hat{u}\|_{E_h^{*,\beta_{i+1}}}. \end{aligned}$$

□

#### REMARKS

1. The theorem requires no stability of  $L_h^*$ .
2. If  $\bar{R}_h = \bar{R}_h^*$  and the set of restrictions  $\{R_h\}$  is stable, the requirement 4) in the theorem follows from the requirements 2) and 3) in the theorem:

$$\begin{aligned} & \|L_h - L_h^*\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} \\ & = \|(L_h - L_h^*) R_h \hat{P}_h\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} \\ & \leq \|\hat{P}_h\|_{E_h^{*,p} \leftarrow E_h^{*,p}} \|L_h R_h - L_h^* R_h\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} \\ & \leq C_5 \|L_h R_h - \bar{R}_h L + \bar{R}_h L - L_h^* R_h\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} \\ & \leq C_5 \|L_h R_h - \bar{R}_h L\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} + C_5 \|\bar{R}_h L + L_h^* R_h\|_{\hat{E}_h^* \leftarrow E_h^{*,p}} \\ & \leq C_5 [C_2 h^p + C_3 h^{p^*}] \leq C h^p. \end{aligned}$$

□

## 2 MULTIGRID ALGORITHMS

### 2.1 INTRODUCTION

In the framework of defect correction processes, multigrid algorithms are easy to explain. For this purpose we consider a continuous problem (1.1) and two discretisations of (1.1) on grids with mesh size  $h$  and  $H$ ,  $H > h$ ,

$$F_h z_h = y_h \quad \text{and} \quad F_H z_H = y_H. \quad (2.1)$$

The operators  $F_h : D_h \subset E_h \rightarrow \hat{D}_h \subset \hat{E}_h$  and  $F_H : D_H \subset E_H \rightarrow \hat{D}_H \subset \hat{E}_H$  are mappings between the discrete spaces  $E_h$ ,  $\hat{E}_h$ ,  $E_H$  and  $\hat{E}_H$ . Further a linear injection  $P_{hH} : E_H \rightarrow E_h$  ( *interpolation* or *prolongation* ) and a linear surjection  $\bar{R}_{Hh} : E_h \rightarrow E_H$  ( *restriction* ) are given.

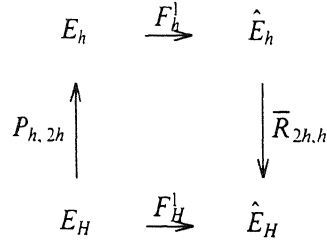


Fig. 2.1 Two-grid mappings and grid transfers.

A multigrid algorithm for the approximate solution of the discretised problem  $F_h z_h = y_h$  is an iterative process in which one iteration cycle consists of

- 1)  $p$  (pre-) relaxation steps ( $p \in \mathbb{N}$ );
- 2) a coarse grid correction step;
- 3)  $q$  (post-) relaxation steps ( $q \in \mathbb{N}$ ).

The relaxation steps are simple defect correction steps as e.g. damped Jacobi relaxation (JOR), Gauss-Seidel (GS), symmetric Gauss-Seidel (SGS), incomplete LU-decomposition iteration (ILU). Their main purpose is to reduce the non-smooth part of the defect. The remaining, smooth defect can be represented well on a coarser grid by means of some restriction.

The coarse grid correction step is a defect correction step of type DCPA where the approximate inverse  $\tilde{G}_h$  is given by

$$\tilde{G}_h = P_{hH} F_H^{-1} \bar{R}_{Hh}. \quad (2.2)$$

The use of the approximate inverse (2.2) implies that we solve the defect equation on the coarse grid, with the help of a coarse discretisation of our problem. Thus we have obtained a two grid method.

If we do not solve the defect equation on the coarser grid exactly, but if we approximate its solution by application of a few iteration steps of the same algorithm on the coarser level, we obtain a recursive process where we have to solve directly a discretised problem only on the very coarsest grid. The resulting process is a true multigrid-algorithm.

One complete iteration step in a multigrid process is called a multigrid cycle.

## 2.2 THE TWO-LEVEL ALGORITHM

Again we consider the two discretisations (2.1) and their relation via restrictions and prolongations as shown in Fig.2.1. We present the algorithms in the form of ALGOL-like programmes. First we describe two auxiliary procedures:

**proc solve = ( operator  $F_h$ , vector  $z_h, y_h$  ).**

This procedure uses the operator  $F_h$  and the right-hand side  $y_h$  to solve (approximately), on a single grid, the equation

$$F_h z_h = y_h.$$

On entry,  $z_h$  should contain an appropriate initial value for the (possibly) iterative solution process. On exit  $z_h$  contains a (better approximate) solution of the problem  $F_h z_h = y_h$ .

**proc relax = ( operator  $F_h$ , vector  $z_h, y_h$  ).**

This procedure performs one (essentially single grid) iteration step of a suitable relaxation method for the equation  $F_h z_h = y_h$ .

Now we explain the essential coarse-grid-correction step. Given an approximation  $z_h$  to the true solution  $z_h^*$  of our discrete problem, we consider the residual

$$r_h = y_h - F_h z_h.$$

With the error  $e_h = z_h - z_h^*$ , we have

$$F_h(z_h - e_h) = y_h = r_h + F_h z_h. \quad (2.3)$$

For a linear operator  $F_h$  this reduces to

$$F_h e_h = -r_h. \quad (2.4)$$

Instead of solving equation (2.3) directly, we compute the solution of a similar equation on a coarse grid

$$F_H w_H = \bar{R}_{Hh} r_h + F_H R_{Hh} z_h \quad (2.5)$$

and then use  $P_{hH}(w_H - R_{Hh} z_h)$  as an approximation for the error  $e_h$  and hence as a correction quantity in the DCP. For a linear operator  $F_h$  the above reduces to

$$F_H e_H = -\bar{R}_{Hh} r_h, \quad (2.6)$$

which gives the correction quantity  $P_{hH} e_H$ . We prefer (approximate) solution of (2.6) above that of (2.4) because the smaller number of unknowns makes it a cheaper process. The algorithm for one cycle of the two-level algorithm reads

```

proc TGM = ( int  $p, q$ , vector  $z_h, y_h$  ):
begin
  to  $p$  do relax ( $F_h, z_h, y_h$ ) enddo ;
  vector  $r_H := \bar{R}_{Hh}(y_h - F_h z_h) + F_H R_{Hh} z_h$ ;
  vector  $w_H := R_{Hh} z_h$ ;
  solve ( $F_H, w_H, r_H$ ):
   $z_h := z_h + P_{hH}(w_H - R_{Hh} z_h)$ ;
  to  $q$  do relax ( $F_h, z_h, y_h$ ) enddo;
end;

```

### 2.3 THE MULTI-LEVEL ALGORITHM

Consider a sequence of grids with mesh widths  $h_i$ ,  $h_{i-1} > h_i$ ,  $i = 0, 1, 2, \dots$ . Often we use  $h_{i-1} = 2h_i$ . We can now describe one cycle of a multi-level algorithm to solve the problem  $F_h z_h = r_h$ . The algorithm uses a sequence of approximate solutions  $z = \{z_h, z_{h_{i-1}}, \dots\}$  and a sequence of right-hand sides  $r = \{r_h, r_{h_{i-1}}, \dots\}$ . At entrance these data are given only for the finest grid.

```

proc MGM = (int i,p,q,σ, sequence of vectors  $z_h, r_h$ ):
begin
  operator  $F_h = F_h, F_H = F_{h_{i-1}}$ ;
  vector  $z_h = z_h, y_h = r_h$ ;
  to p do relax ( $F_h, z_h, y_h$ ) enddo;
  vector  $r_H = r_{h_{i-1}} := \bar{R}_{Hh}(y_h - F_h z_h) + F_H R_{Hh} z_h$ ;
  vector  $w_H = z_{h_{i-1}} := R_{Hh} z_h$ ;
  if i = 0
  then solve ( $F_H, w_H, r_H$ )
  else to σ do MGM (i-1,p,q,σ,  $W_H, r_H$ ) enddo
  endif;
   $z_h := z_h + P_{hH}(w_H - R_{Hh} z_h)$ ;
  to q do relax ( $F_h, z_h, y_h$ ) enddo
end;

```

Multigrid cycles based on the above (recursive) nonlinear MGM algorithm are also known by the name FAS-cycles ( FAS = *Full Approximation Scheme* ).

**THEOREM 2.1** Consider an application of MGM where  $h_i/h_{i+1} = H/h$  is constant for all  $i = 0, 1, 2, \dots$ . Let  $d$  be the dimension of the grid, i.e. the number of space dimensions. If  $\sigma < (H/h)^d$  then the total amount of work in a multigrid cycle is proportional to the amount of the work on the fine grid.

**PROOF** Let  $W$  be the amount of computational work needed to perform relaxations, operator evaluations, restrictions and prolongations on the finest grid. This number is proportional to the number of grid points in the finest grid. On every next coarser grid the number of nodal points is reduced by a factor  $(h/H)^d$ . Hence the amount of work on the coarser grid is reduced by the same factor. If we consider an infinite number of grids, the total amount of work is given by

$$W_{\text{tot}} = \sum_{n=0}^{\infty} [\sigma(h/H)^d]^n W = \frac{W}{1 - \sigma(h/H)^d}. \quad (2.7)$$

The above series converges for  $\sigma < (H/h)^d$ ; hence  $W_{\text{tot}}$  is proportional to  $W$ .  $\square$

In the above multigrid algorithm the fixed numbers  $p, q$  and  $\sigma$  determine the *strategy* of the algorithm. Other multigrid algorithms may terminate iterations sooner, depending on the convergence or other conditions that can be checked during the computation. Multigrid algorithms that make use of this possibility have an *adaptive strategy*; algorithms where the iterations are controlled only by the fixed numbers  $p, q$  and  $\sigma$  have a *fixed strategy*. MG-cycles with  $\sigma = 1$  are called *V-cycles*, those with  $\sigma = 2$  are called *W-cycles*. A V-cycle with  $p + q = 1$  is called a *sawtooth cycle*.

In Fig.2.2 we show for some fixed strategies how is switched between the different levels of discretisation. We see that - essentially - most relaxation sweeps are performed on the coarser levels. In all diagrams the number of the levels is 4, the coarsest level is denoted by 0. Segments between tick-marks on a level  $> 0$  denote the execution of a relaxation step on this level; a segment on level 0 denotes the direct solution on the coarsest level.

Let  $M_h^{REL}$  and  $\bar{M}_h^{REL}$  denote the (linear) amplification operators of the relaxation for error and residual respectively. For TGM the amplification operator for the error is given by

$$M_h^{TGM,p,q} = (M_h^{REL})^q (I_h - P_{hH} F_H^{-1} \bar{R}_{Hh} F_h) (M_h^{REL})^p \quad (2.8)$$

$$= (M_h^{REL})^q (F_h^{-1} - P_{hH} F_H^{-1} \bar{R}_{Hh}) (\bar{M}_h^{REL})^p F_h.$$

Multigrid theory[16] shows that, under suitable conditions,

$$\| M_h^{TGM,p,q} \| \leq C < 1,$$

with  $C$  independent of  $h$ . We denote the amplification operator of a multi-level iteration step (MGM) on the  $h$ -level of discretisation by  $M_h^{MGM,p,q,\sigma}$ , or  $M_h^{MGM}$  for short. The same amplification operator on the next coarser level we denote by  $M_H^{MGM,p,q,\sigma}$  or  $M_H^{MGM}$ . In the multigrid cycle the approximate inverse is not given by

$$P_{hH} F_H^{-1} \bar{R}_{Hh},$$

because  $F_H^{-1}$  is approximated by application of  $\sigma$  steps of a DCP. The amplification operator of this DCP is given by  $M_H^{MGM}$ . Hence the approximate inverse of the  $\sigma$  iteration steps together is given by

$$(I_H - (M_H^{MGM})^\sigma) F_H^{-1}$$

(see section 1.4.2). Consequently, the amplification operator of the coarse grid correction in MGM is

$$I_h - P_{hH} (I_H - (M_H^{MGM})^\sigma) F_H^{-1} \bar{R}_{Hh} F_h$$

and we have

$$\begin{aligned} M_h^{MGM,p,q,\sigma} &= (M_h^{REL})^q (I_h - P_{hH} (I_H - (M_H^{MGM})^\sigma) F_H^{-1} \bar{R}_{Hh} F_h) (M_h^{REL})^p \\ &= M_h^{TGM} + (M_h^{REL})^q P_{hH} (M_H^{MGM})^\sigma F_H^{-1} \bar{R}_{Hh} F_h (M_h^{REL})^p. \end{aligned} \quad (2.9)$$

This shows that for a contractive coarse grid correction and for sufficiently large  $\sigma$ , the convergence of MGM is essentially as good as that of TGM. Often  $\sigma = 1$  or  $\sigma = 2$  is sufficiently large.

## 2.4 THE FULL MULTIGRID METHOD (FMG)

The multigrid cycles we described in the previous subsection yield iterative improvement of a solution on a fine grid and, therefore, they need some initial estimate of the solution on this finest grid to start with. One possible algorithm is to obtain the initial estimate by interpolation from a solution on the next coarser grid, which has been previously calculated by a similar algorithm. Using this algorithm we start solving the problem on the coarsest grid. An algorithm of this type is called a *Nested Iteration* process:

**proc** nested iteration = ( int  $l$ , sequence of int  $it$ , sequence of vectors  $z, y$  ):

**begin**

    solve ( $F_0, z[0], y[0]$ ); # sufficiently accurate #

**for**  $i$  to  $l$  **do**

$z[i] := P_{i,i-1}^* z[i-1]$ ;

**for**  $j$  to  $it[i]$

            do iteration ( $i, z[i], y[i]$ ) **enddo**;

**enddo**

**end**

### REMARK

- $l+1$  is the number of levels available; the coarsest level is again denoted by 0.
- $y[i]$  is the right-hand side for the equation to be solved on level  $i$ .
- $it[i]$  denotes the number of iteration steps needed on level  $i$ .
- $P_{i,i-1}^*$  is an interpolation from level  $i-1$  to level  $i$ . This operator is not necessarily the same as the prolongation operator used in the the multigrid cycles; usually it is more accurate.

The procedure “iteration” represents one step of a suitable iterative solution process. In the multigrid context we will use MGM to replace “iteration”. In this case we call the resulting method a *full multigrid method* (FMG).

In an FMG-algorithm the coarse grids have a double function:

- i) providing the iterative process with an initial estimate
- ii) speeding up the process on a finer grid.

A typical FMG algorithm with one V-cycle per coarse grid, is shown in Fig.2.3 (where we have  $it[i] = 1$  for all  $i$ ).

**THEOREM 2.2** Consider a sequence of discretisations  $F_h, z_h = r_h$ , with  $h_{i-1}/h_i > C_1, i = 0, 1, 2, \dots$ . If the discrete equations are relatively convergent of order  $p$ , i.e. if

$$\|P_{i,i-1}z_{i-1} - z_i\| \leq C_0 h_{i-1}^p,$$

and the convergence of the iteration cycle is independent of  $h$ , i.e. for the iterands  $z_i^{(j)}$

$$\|z_i^{(j+1)} - z_i\| \leq C_2 \|z_i^{(j)} - z_i\|,$$

then, with  $m$  cycles on each level, the result  $\tilde{z}_h = z_h^{(m)}$  of the Nested Iteration process satisfies

$$\|\tilde{z}_h - z_h\| < \frac{C_1^m C_2^m}{1 - C_1^m C_2^m \|P\|} \cdot C_0 h_i^p$$

assumed that  $C_1^m C_2^m \|P\| < 1$ , where  $\|P\| = \sup_i \|P_{i,i-1}\|$ .

**PROOF (sketch)**

For all  $i \geq 0$  we have  $\|z_i^{(m)} - z_i\| \leq C_2^m \|z_i^{(0)} - z_i\|$ . On the coarsest level  $z_0$  should be approximated sufficiently well (truncation error accuracy):  $\|\tilde{z}_0 - z_0\| \leq C_2^m C_1^m C_0 h_0^p$ . Then the theorem holds for  $i = 0$ .

For  $i > 0$

$$\begin{aligned} \|z_i^{(0)} - z_i\| &= \|P_{i,i-1}z_{i-1}^{(m)} - P_{i,i-1}z_{i-1} + P_{i,i-1}z_{i-1} - z_i\| \\ &\leq \|P_{i,i-1}\| \|z_{i-1}^{(m)} - z_{i-1}\| + \|P_{i,i-1}z_{i-1} - z_i\| \\ &\leq \|P\| \|z_{i-1}^{(m)} - z_{i-1}\| + C_0 h_{i-1}^p. \end{aligned}$$

By induction we have

$$\begin{aligned} \|z_i^{(m)} - z_i\| &\leq C_2^m \|P\| \|z_{i-1}^{(m)} - z_{i-1}\| + C_2^m C_0 C_1^m h_i^p \\ &\leq C_2^m C_0 C_1^m h_i^p + C_2^m \|P\| \{C_2^m C_0 C_1^m h_{i-1}^p + C_2^m \|P\| \{\dots\}\} \\ &= \frac{C_2^m C_1^m}{1 - C_2^m C_1^m \|P\|} \cdot C_0 h_i^p \end{aligned}$$

□

Thus, the theorem shows that, for suitable constants  $C_1$  and  $C_2$ , only a fixed (small) number of iterations,  $it[i]=m$ , is sufficient on all levels to reduce the iteration error to the same size as the truncation error.

### 3. MULTIGRID APPROACHES FOR COMPRESSIBLE FLOW

#### 3.1 THE EQUATIONS OF COMPRESSIBLE FLOW

The efficient solution of flow problems is one of the earliest aims for multigrid methods [6]. Most progress in the development of MG has, however, been made in the field of elliptic partial differential equations. For the more complex equations that describe flow problems, the development of MG was hanging back. Early work was done by Brandt [8,9,65] for the Stokes equations and both the incompressible and compressible Navier-Stokes equations.



Still, many attempts were made to apply MG ideas to improve the efficiency of flow computations. Assuming irrotational flow, it is described by the potential equation, which -in the interesting case of transonic flow- is of mixed hyperbolic and elliptic type. By the use of MG, substantial improvements were made in the solution procedures for these equations[4, 10, 27, 50, 54, 65].

In section 3.3 we give a small survey of the several multiple grid approaches used for the solution of the Euler- and Navier-Stokes equations of compressible flow. Most of the work has been done for Euler problems in 2 space dimensions. Only recently attempts are made to apply MG methods to problems in 3 space dimensions and for the full compressible Navier-Stokes equations. First we make some brief remarks on the equations in 2-D and their discretisations.

### 3.1.1 The Navier-Stokes equations.

On a two-dimensional domain  $\Omega^* \subset \mathbb{R}^2$ , the 2-D *Navier-Stokes equations*, describing the physical laws of conservation of mass, momentum and energy, can be written as

$$\frac{\partial}{\partial t} q + \frac{\partial}{\partial x} F(q) + \frac{\partial}{\partial y} G(q) = 0, \quad (3.1a)$$

where

$$F(q) = f(q) - \text{Re}^{-1} r(q), \quad G(q) = g(q) - \text{Re}^{-1} s(q), \quad (3.1b)$$

and

$$q = (\rho, \rho u, \rho v, \rho e)^T,$$

$$f = (\rho u, \rho u^2 + p, \rho uv, \rho uH)^T,$$

$$g = (\rho v, \rho vu, \rho v^2 + p, \rho vH)^T,$$

$$r = (0, \tau_{xx}, \tau_{xy}, \kappa \text{Pr}^{-1}(\gamma-1)^{-1}(c^2)_x + u\tau_{xx} + v\tau_{xy})^T,$$

$$s = (0, \tau_{xy}, \tau_{yy}, \kappa \text{Pr}^{-1}(\gamma-1)^{-1}(c^2)_y + u\tau_{xy} + v\tau_{yy})^T.$$

Here  $\rho$ ,  $u$ ,  $v$ ,  $e$  and  $p$  respectively represent density, velocity in x- and y- direction, specific energy and pressure;  $H = e + p/\rho$  is the specific enthalpy. For a perfect gas

$$p = (\gamma-1)\rho(e - \frac{1}{2}(u^2 + v^2));$$

$\gamma$  is the ratio of specific heats. The unknown vector  $q(t, x, y)$  describes the state of the gas as a function of time and space and  $f$  and  $g$  are the convective fluxes in the x- and y- direction respectively.  $\text{Re}$  and  $\text{Pr}$  denote the Reynolds and Prandtl number; thermal conductivity is given by  $\kappa$ ;  $c = \sqrt{\gamma p / \rho}$  is the local speed of sound; and

$$\tau_{xx} = (\lambda + 2\mu)u_x + \lambda v_y,$$

$$\tau_{xy} = \mu(u_x + v_x),$$

$$\tau_{yy} = (\lambda + 2\mu)v_y + \lambda u_x,$$

where  $\lambda$  and  $\mu$  are viscosity coefficients. Often Stokes' assumption of zero bulk viscosity is used:  $3\lambda + 2\mu = 0$ .

### 3.1.2 The Euler equations.

The *Euler equations* are obtained from (3.1a,b) by neglecting viscous and heat conduction effects in (3.1b); then

$$F(q) = f(q), \quad G(q) = g(q). \quad (3.1c)$$

The time dependent Euler equations form a hyperbolic system: written in the quasi-linear form

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial x} + \frac{\partial g}{\partial q} \cdot \frac{\partial q}{\partial y} = 0,$$

the matrix

$$k_1 A + k_2 B = k_1 \frac{\partial f}{\partial q} + k_2 \frac{\partial g}{\partial q} \quad (3.2)$$

has real eigenvalues for all directions  $(k_1, k_2)$ .

These eigenvalues are  $(k_1 u + k_2 v) \pm c$  and  $(k_1 u + k_2 v)$  (a double eigenvalue). The sign of the eigenvalues determines the direction in which the information about the solution is carried along the line with direction  $(k_1, k_2)$  as time develops.

Because of the nonlinearity, solutions of the Euler equations may develop discontinuities, even if the initial flow ( $t=t_0$ ) is smooth. To allow discontinuous solutions, (3.1) is rewritten in its integral form

$$\frac{\partial}{\partial t} \iint_{\Omega} q \, dx \, dy + \int_{\partial\Omega} (F n_x + G n_y) \, ds = 0, \quad \text{for all } \Omega \subset \Omega^*; \quad (3.3)$$

$\partial\Omega$  is the boundary of  $\Omega$  and  $(n_x, n_y)$  is the outward normal vector at the wall  $\partial\Omega$ .

The form (3.3) of equation (3.1) shows clearly the character of the system of conservation laws: the increase of  $q$  in  $\Omega$  can be caused only by the inflow of  $q$  over  $\partial\Omega$ . In symbolic form (3.3) is written as

$$q_t + N(q) = 0. \quad (3.4)$$

The solution of the weak form (3.3) of (3.1a,c) is known to be non-unique and a physically realistic solution (which is the limit of a flow with vanishing viscosity) is known to satisfy the entropy condition (cf. [44, 45]).

Interested mainly in the *steady state* equations, obtained by the assumption  $\partial q / \partial t = 0$ , we can concentrate on the solution methods for the steady Euler equations:

$$N(q) = 0. \quad (3.5)$$

Notice that  $N$  can be seen as a nonlinear mapping between two Banach spaces,  $N: X \rightarrow Y$ .

### 3.2 THE DISCRETISATIONS

For the discretisation of (3.1) or (3.3), two different approaches can be taken. First, the time and space discretisations can be made at once. This leads, for example, to discretisation schemes of Lax-Wendroff type. An initial state of the fluid,  $q_h^{(n)}$ , defined on a discrete grid, is advanced over one time-step. Using a second-order approximation in time, this yields

$$q_h^{(n+1)} = q_h^{(n)} + \Delta t (q_h)_t + \frac{1}{2} (\Delta t)^2 (q_h)_{tt}. \quad (3.6)$$

With the equation (3.1a,c), we arrive at

$$q_{ij}^{(n+1)} = q_{ij}^{(n)} - \Delta t (f_x + g_y)_{ij} + \frac{1}{2} (\Delta t)^2 \left\{ [A(f_x + g_y)]_x + [B(f_x + g_y)]_y \right\}_{ij},$$

where  $A$  and  $B$  are defined by (3.2). Using various difference approximations of the bracketed terms in the right-hand side, different Lax-Wendroff type discretisations may be obtained.

Typically this type of discretisation is made on a rectangular grid. If the domain  $\Omega^*$  is not rectangular, a 1-1-mapping  $(x, y) \longleftrightarrow (\xi, \eta)$  between the physical domain and a rectangular computational domain can be constructed. Then the differential equation and the boundary conditions are reformulated on this computational domain.

A property of most of these Lax-Wendroff discretisations is that, when by time-stepping a steady state is obtained, such that  $q_h^{(n+1)} = q_h^{(n)}$ , the discrete steady state still depends on  $\Delta t$ . This is caused by the fact that the discrete term with  $(\Delta t)^2$  in (3.6) does not vanish in general.

A second approach is to distinguish clearly between the time and the space discretisation by the method of lines. First, a space discretisation is made for the partial differential equation (3.4), by which it is reduced to the large system of ordinary differential equations (ODEs)

$$\frac{\partial}{\partial t} q_h = N_h(q_h). \quad (3.7)$$

Now, to find an approximation of the time-dependent solution of (3.4), any method can be used for the integration of this system of ODEs. The solution of the steady state can be computed by solving (3.7) until all transients have died out. Alternatively, we can avoid the ODEs (3.7) and solve the non-linear system

$$N_h(q_h) = 0 \quad (3.8)$$

by other (more direct) means. In both cases (3.7) and (3.8), we find a steady approximate solution  $q_h$  which is independent of the choice of a time step.

For the construction of the semidiscrete system (3.7) or (3.8) on a non-rectangular domain  $\Omega^*$ , again a mapping  $(x,y) \leftrightarrow (\xi,\eta)$  can be introduced and finite difference approximations (of an arbitrarily high order) can be used to construct a space discretisation of the transformed steady equation

$$[y_\eta F(q) - x_\eta G(q)]_\xi + [-y_\xi F(q) + x_\xi G(q)]_\eta = 0.$$

Another way to construct the system (3.7) on a non-rectangular grid is by a *finite volume* technique. Here, the starting point for the discretisation is (3.3). Without an *a-priori* transformation, the domain  $\Omega^*$  is divided into a set of disjoint (quadrilateral) cells  $\Omega_{ij}$ . The discrete representation  $q_h$  of  $q$  is given by the values  $q_{ij}$ , the (mean) values of  $q$  in the cell  $\Omega_{ij}$ . Using different approximations for the computation of fluxes between the cells  $\Omega_{ij}$ , different finite volume discretisations are obtained. A conservative scheme is easily obtained by computing a unique approximation for each flux over the boundary between two neighbouring cells.

In order to define a proper sequence of discretisations as  $h \rightarrow 0$  for a non-rectangular grid, a formal relation between the vertices of cells  $\Omega_{ij}$  and a regular grid can be given, again by a mapping  $(x,y) \leftrightarrow (\xi,\eta)$ . If this mapping is smooth enough (Fig.3.1), it can be proved that for refinements  $h \rightarrow 0$  which correspond with regular refinements in  $(\xi,\eta)$ , space discretisations up to second order can be obtained by finite volumes. An advantage of the finite volume technique is that the untransformed equations can be used, even for a complex region. Boundary condition information is also usually simpler for finite volume methods.

With the finite volume technique, both central difference and upwind type finite volume schemes are used. They differ by the computation of the flux between neighbouring cells  $\Omega_{ij}$ .

(1) For a central difference type, the flux over a cell wall  $\Gamma_{LR}$  between two cells with states  $q_L$  and  $q_R$  is computed as  $f^*(\frac{1}{2}(q_L + q_R))$ , where  $f^* = k_1 f + k_2 g$  is the flux normal to  $\Gamma_{LR}$ . On a Cartesian grid this scheme reduces to the usual central difference scheme. In order to stabilise this scheme, and to prevent the uncoupling of odd and even cells in the grid, it is necessary to supplement it with some kind of artificial dissipation (artificial viscosity).

(2) For upwind difference type discretisations, *numerical flux functions*  $f^*(q_L, q_R)$  are introduced to compute the flux over  $\Gamma_{LR}$ . Several functions  $f^*$  are possible. They solve approximately the Riemann problem of gas-dynamics: they approximate the flux between two (initially) uniform states  $q_L$  and  $q_R$ . Approximate Riemann solvers have been proposed by Steger and Warming[68] van Leer[46], Roe[61], Osher[55,58] and others. In section 4.2 we give a description of Osher's scheme, for further descriptions we refer to the literature mentioned. For a consistent scheme,  $f^*(q, q) = f^*(q)$ , i.e. the numerical flux function with equal arguments conforms with the genuine flux function in (3.1c). All these upwind flux-functions have in common that they are purely one-sided if all characteristics point into the same direction, i.e.  $f^*(q_L, q_R) = f^*(q_L)$  if the flow of all information is from left to right. More details are given in section 4.

### 3.3 THE MULTIPLE GRID METHODS

When a multiple grid technique is used to solve the system of nonlinear (differential) equations (3.7) or (3.8), we assume the existence of a nested set of grids. Usually this nesting is such that a set of  $2 \times 2$  cells in a fine mesh form a single cell in the next coarser one. (No staggered grids!) The coarser grids are used to effect the acceleration of a basic iterative (time marching or relaxation) procedure on the finest grid.

Slightly generalising the equations (3.7) or (3.8) to

$$\frac{\partial}{\partial t} q_h = N_h(q_h) - r_h \quad (3.9)$$

or

$$N_h(q_h) = r_h, \quad (3.10)$$

where  $r_h$  denotes a possible correction or source term, we can write the basic iterative procedure as

$$q_h^{(n+1)} \leftarrow \mathcal{G}^h(q_h^{(n)}, r_h). \quad (3.11)$$

The usual coarse grid acceleration algorithm is as follows: starting with an approximation  $q_h^{(k)}$  on the finest mesh, and some approximation  $q_{2h}^{(0)}$  on the next coarser (e.g.  $q_{2h}^{(0)} = R_{2h,h} q_h^{(k)}$ ), first an approximate solution is found for the coarse grid problem

$$N_{2h}(q_{2h}) = N_{2h}(q_{2h}^{(0)}) - \bar{R}_{2h,h}(N_h(q_h^{(n)}) - r_h), \quad (3.12)$$

(cf. eq. (2.5)) and then the value  $q_h^{(k)}$  is updated by

$$q_h^{(k+1)} = q_h^{(k)} + P_{h,2h}(q_{2h} - q_{2h}^{(0)}). \quad (3.13)$$

The combination of (3.12) and (3.13) is the *coarse grid correction* (CGC) step. The solution  $q_{2h}$  of (3.12) can be approximated e.g. by an (accelerated) iteration process on the 2h-grid again.

We shall see later in this section that, besides this usual coarse grid acceleration procedure, the coarser grids sometimes play a different role in the acceleration process [35, 53].

As we saw in chapter 2, a multigrid FAS cycle for the solution of (3.10) now consists of the following steps:

- (0) start with an approximate solution  $q_h$ .
- (1) improve  $q_h$  by application of  $p$  nonlinear (pre-) relaxation iterations to  $N_h(q_h) = r_h$ .
- (2) if the present grid is not the coarsest, improve  $q_h$  by application of one coarse-grid-correction step, where the approximation of (3.12) is effected by  $\sigma$  FAS-cycles to this coarser grid problem; if the present grid is the coarsest, simply skip to (3).
- (3) improve  $q_h$  by application of  $q$  nonlinear (post-) relaxation iterations to  $N_h(q_h) = r_h$ .

#### 3.3.1 Methods based on Lax-Wendroff type time stepping.

A paper by Ni[53] was among the first to apply a MG acceleration to the (isenthalpic) Euler equations. He uses the following time-stepping procedure as a basic iteration. Starting with an initial state  $q_h^{(n)}$ , where the values  $q_{ij}^{(n)}$  are given at the grid points, he first computes the following quantities, by means of a control volume centered integration method with fluxes interpolated from corner values:

$$\begin{aligned} \Delta q_{i+\frac{1}{2},j+\frac{1}{2}} = & -\frac{1}{2} \frac{\Delta t}{\Delta x} [(F_{i+1,j} - F_{i,j}) + (F_{i+1,j+1} - F_{i,j+1})] \\ & -\frac{1}{2} \frac{\Delta t}{\Delta y} [(G_{i,j+1} - G_{i,j}) + (G_{i+1,j} - G_{i+1,j+1})], \\ F_{i,j} = & F(q_{i,j}^{(n)}) \quad \text{etc.} \end{aligned} \quad (3.14)$$

These increments then are distributed over the mesh points, using direction-weighted means (cell-increments are distributed over mesh-point values):

$$\Delta q_{ij} = \frac{1}{4} \sum_{l=\pm 1} \sum_{k=\pm 1} \left[ I - k \frac{\Delta t}{\Delta x} A_{i+\frac{k}{2},j+\frac{l}{2}} - l \frac{\Delta t}{\Delta y} B_{i+\frac{k}{2},j+\frac{l}{2}} \right] \Delta q_{i+\frac{k}{2},j+\frac{l}{2}}, \quad (3.15)$$

$$q_{ij}^{(n+1)} = q_{ij}^{(n)} + \Delta q_{ij}.$$

By the use of the Jacobian matrices  $A$  and  $B$ , this distribution formula has a kind of upwind effect, but for transonic or supersonic cases an artificial damping is still necessary.

Symbolically, this time stepping process (3.14)-(3.15) is described as:

$$\text{compute } \Delta q_h^{\text{cell}}, \quad (3.16a)$$

$$\text{with cell values } \Delta q_{i+1/2,j+1/2} \approx - \Delta t \int_{\partial \Omega_{i+1/2,j+1/2}} (f \cdot n_x + g \cdot n_y) ds / (\Delta x \cdot \Delta y);$$

$$q_h^{(n+1)} := q_h^{(n)} + D_h \Delta q_h^{\text{cell}}. \quad (3.16b)$$

The operator  $D_h$  is the distribution operator that transfers the cell centered corrections to the grid points by means of (3.15).

The coarse grid acceleration as introduced in [53] by Ni deviates from the usual coarse grid scheme (3.12) (3.13). In [53] the coarse grid correction is obtained by first computing corrections at coarser cells,  $\Delta q_{2h}^{\text{cell}}$ . This can be done by restriction of  $\Delta q_h$  to the 2h-grid. Then the corrections  $\Delta q_{2h}^{\text{cell}}$  are distributed to the coarser mesh points similar to (3.15), and the coarse grid correction is interpolated to the fine grid.

Thus, here the coarse grid correction reads

$$\Delta q_{2h}^{\text{cell}} := R_{2h,h} \Delta q_h^{\text{cell}}, \quad (3.17a)$$

$$q_h^{(n+1)} := q_h^{(n)} + P_{h,2h} D_{2h} \Delta q_{2h}^{\text{cell}}. \quad (3.17b)$$

where  $P_{h,2h}$  is a (bi-)linear interpolation operator.

Since the coarse grid corrections are based on fine grid residuals, it is obvious that the possible convergence to a steady state yields a solution of the system (3.8).

In the same way the correction procedure can be repeated on progressively coarser grids. Therefore, in (3.17),  $2h$  should be replaced by  $2^m h$ . We notice that, in contrast with the usual MG-method as described in section 2, here the corrections on the different levels can be computed independent of each other. This yields the possibility to compute all coarse grid corrections,  $m = 1, \dots, L$ , in parallel and to form the correction

$$q_h^{(n+1)} = q_h^{(n)} + \sum_{m=1}^L P_{h,2^m h} D_{2^m h} \Delta q_{2^m h}^{\text{cell}}.$$

at once [70]. When optimal use of modern multi-processor computers is to be made, it is also possible to perform both computations (3.16a) and (3.17) in parallel [37, 70].

We see that there are still possibilities to form different variants in the Ni-type multigrid Euler solver. First, any other Lax-Wendroff-type time-marching procedure can be used for (3.16a). In [11, 34, 36] Johnson applies the popular MacCormack scheme. Further, in (3.17a) various restrictions,  $R_{2h,h}$ , can be used. It transfers the values of the fine grid corrections to a single value for each control volume in the coarser grid. Injection of the correction in the main point of the corresponding cell is often used [35], but also weighted averages are an obvious choice.

Heuristically, the elucidation for the accelerating effect of the corrections (2.4) is, that these coarse grid corrections may move disturbances of the steady state over the distance of many mesh cells in one time step, whereas the accuracy of the final solution is only determined by the finest grid. Apparently, it is also necessary that the Lax-Wendroff schemes used in combination with this coarse grid correction are (by the choice of a suitable  $\Delta t$  or otherwise) sufficiently dissipative to reduce the high frequency disturbances that are present in the initial approximation and those introduced during the process by the interpolation in (3.13). Up to now, no complete mathematical theory has been

developed to explain and to quantify the amount of acceleration, which is clearly found in the many computations that use the described method.

### 3.3.2 Methods based on semidiscretisation and time stepping.

When only the solution of the steady state is to be computed, the time-accurate integration of the system of ODEs is wasteful. The convergence of (3.4) to steady state is slow. However, the desire to have a procedure that solves transient as well as steady state problems, coding convenience, or the restrictions imposed by the optimal use of vector computers may be a reason to prefer time-stepping methods. When no time accuracy is desired, many devices are known to accelerate the integration process (cf. [63]). For the solution of the Euler equations, these devices include: (i) local time-stepping, which means that the step size in the integration process may differ over different parts of the domain  $\Omega^*$ ; (ii) enthalpy damping, where a-priori knowledge about the behaviour of the enthalpy over  $\Omega^*$  is used (e.g.  $H$  constant over  $\Omega^*$ ); (iii) residual smoothing, (iv) implicit residual averaging, and (v) implicit corrected viscosity acceleration[15]. In residual smoothing and implicit residual averaging the fact is used that instability effects appear first for high frequencies, so that larger time steps are possible when the residual is smooth.

For all explicit integration methods, stability requirements set a limit to the size of the possible time steps (CFL limits). Implicit integration procedures can be unconditionally stable, but they require the solution of a (nonlinear) system in each individual time step.

An important code, based on a time-stepping method has been developed by Jameson, Schmidt and Turkel[30]. They use an explicit time-stepping method of Runge-Kutta type. This *multistage time-stepping procedure* is a specially adapted Runge-Kutta method, where the hyperbolic (=convective) and the parabolic (=dissipative) parts of  $N_h(q_h)$  are treated separately. The Runge-Kutta coefficients in the k-stage Runge-Kutta schemes ( $k=3,4$ ), are selected not only for their large stability bounds, but also with the aim to improve the damping of the high frequency modes. In the k stages of the Runge-Kutta process, the updating of the dissipative part is frozen at the first stage. This saves a substantial part of the computational effort.

The multigrid scheme used by Jameson [28] is a FAS sawtooth cycle with  $q=1$ . The restriction  $R_{2h,h}$  ( $\bar{R}_{2h,h}$ ) is defined by volume-weighted averaging of the states (respectively summation of changes of states). The prolongation  $P_{h,2h}$  is defined by bilinear interpolation. The basic smoothing procedure is the "multistage time-stepping scheme". On the coarser grids the stability bounds for the time step, which are  $\mathcal{O}(h)$ , allow larger time steps. On each grid the time step is varied locally to yield a fixed Courant number, and the same Courant number is used on all grids, so that progressively larger time steps are used after each transfer to a coarser grid. As for Ni's method, the reasoning is that disturbances from the steady state will be more rapidly expelled from the domain  $\Omega^*$  by the larger time steps. The interpolation of corrections back to the fine grid introduces high frequency errors, which cannot be rapidly expelled. These errors should be locally damped. Hence, to obtain a fast rate of convergence, the time-stepping process should rapidly damp the high frequency errors.

In [33] Jespersen announces an interesting theorem on the use of the MG process in combination with a time-stepping procedure. This theorem asserts the following. Let  $N_h(q_h) = 0$  be a space discretisation of  $N(q) = 0$ , which is consistent, i.e.

$$N_h(R_h(q)) - \bar{R}_h N(q) = \mathcal{O}(h),$$

and let the time-stepping procedure be consistent in time

$$q_h^{(n+1)} = q_h^{(n)} + \Delta t_{(n)} [N_h(q_h^{(n)}) - r_h] + \mathcal{O}((\Delta t_{(n)})^2).$$

If we consider the sawtooth algorithm, with  $q=1$ ,  $p=0$ ,  $\sigma=1$ , and if  $P_h$  and  $R_h$  satisfy an approximation property (i.e. for a smooth function  $q$  the prolongation and restriction in the state space are such that  $P_h R_h q - q = \mathcal{O}(h)$ ), then the MG algorithm on  $L$  grids is a consistent, first-order in time, discretisation of (3.4) with time step  $\Delta t_{tot} = \sum_{j=1, \dots, L} \Delta t_j$ .

This theorem formalises in a sense the heuristic reasoning that on coarser grids the deviations from steady state can be expelled faster by the use of larger time steps. This may suggest that more, say  $k > 1$ , steps on the coarser grids would improve the convergence even more. However, the theorem regards consistency; stability is not considered. In the same paper [33] Jespersen shows by an example that convergence is lost when a large number of relaxations is made on the coarse grid. In fact a strong stability condition of the form  $\Delta t / \Delta x \leq \mathcal{O}(k^{-1})$  seems to appear.

### 3.3.3 Fully implicit methods.

Most methods so far developed are based on the concept of integrating the equations (3.4) in time until a steady state is reached. If we are only interested in a possible solution of the steady state equation (3.5) and assume that this solution is unique, we may disregard the time-dependence completely. Further, assuming that a suitable space discretisation takes into account the proper characteristic directions, we can restrict ourselves simply to the solution of the nonlinear system (3.8) or

$$N_h(q_h) = r_h. \quad (3.18)$$

Also, if the time-dependent system (3.9) is solved by means of an implicit time-stepping method -in order to circumvent the stability bounds on  $\Delta t$ -, we have to solve systems (3.18) at each step time step. As soon as we mix time-dependent solution with these implicit solution methods and give up time accuracy for (3.18), there is little or no difference between these time stepping procedures and (nonlinear) relaxation methods.

Starting with the nonlinear system (3.18), two direct MG approaches are open. We can either apply the nonlinear multiple grid algorithm (FAS) directly to the system (3.18) or we may apply linearisation (Newton's method) and use the linear version of multiple grid for the solution of the resulting linear systems. Jespersen [32] gives an extensive recital of the (dis)advantages of both approaches. Both have been used with success for the Euler equations.

Linearisation has been used by Jespersen [31] and Mulder [51]; the nonlinear FAS procedure is used by Steger [67], Jespersen [31] and Hemker-Spekrijse [25, 26].

In all these papers upwind discretisations have been used. In [31, 67] the Steger-Warming scheme is used; [51] uses the differentiable van Leer flux-splitting method; [25, 26] use Osher's flux difference splitting. In [14] Dick also considers Roe's flux difference splitting for the 1-D Euler equations.

When Newton's method is applied for linearisation, it may be difficult to start in the domain of contraction of the iteration. Therefore, Mulder [51] introduces the so called Switched Evolution Relaxation (SER) scheme, which is a chimera of a forward Euler time-stepping and a Newton method:

$$\left[ \frac{1}{\Delta t} I - \frac{\partial}{\partial q} N_h(q_h^{(n+1)}) \right] (q_h^{(n+1)} - q_h^{(n)}) = N_h(q_h^{(n)}). \quad (3.19)$$

For  $\Delta t \rightarrow 0$ , this gives the simple time stepping procedure; for  $\Delta t \rightarrow \infty$ , (3.19) is equivalent to Newton's method. In the actual computation  $\Delta t$  varies, depending on the size of the residual, such that (3.19) is initially a time stepping procedure and becomes Newton's method in the final stages of the solution process.

In a FAS procedure, a natural way to obtain an initial estimate is -of course- the use of Full Multi-Grid (FMG) [7]. The initial estimate is obtained by interpolation from the approximate solution on the coarser grid(s). For many problems this process gives very good results, even if one starts with rough approximations on a really coarse grid. In the sections 4,5 and 6 we will give a more detailed description of a fully implicit MG method.

#### 4. MULTIGRID FOR THE FIRST-ORDER DISCRETISATION OF THE EULER EQUATIONS

##### 4.1 THE FIRST-ORDER FINITE VOLUME DISCRETISATION

To discretise (3.4), the domain  $\Omega$  is subdivided into disjunct quadrilateral cells  $\Omega_{i,j}$ , in a regular fashion such that

$$\Omega = \bigcup_{i,j} \Omega_{i,j}.$$

We restrict ourselves to subdivisions that are topologically equivalent with simple square meshes, such that  $\Omega_{i,j}$  and  $\Omega_{i,j\pm 1}$  or  $\Omega_{i\pm 1,j}$  are neighbouring cells. Further we denote the neighbours of  $\Omega_{i,j}$  by  $\Omega_{ijk}$ , ( $k=N,S,E,W$ ) and a common wall by  $\Gamma_{ijk} = \bar{\Omega}_{ij} \cap \bar{\Omega}_{ijk}$ . The boundary of  $\Omega_{ij}$  is given by  $\partial\Omega_{ij} = \bigcup_{k=N,S,E,W} \Gamma_{ijk}$ . The restriction to this kind of regular geometry is not necessary for the discretisation method but leads to simple data structures when the method is implemented.

By integration of (3.4) over  $\Omega_{i,j}$  we obtain

$$\frac{\partial}{\partial t} \iint_{\Omega_{i,j}} q \, dx \, dy + \int_{\partial\Omega_{i,j}} (f \cdot n_x + g \cdot n_y) \, ds = 0 \quad (4.1a)$$

or

$$V_{ij} \frac{\partial}{\partial t} q_{ij} + \sum_k \int_{\Gamma_{ijk}} (f \cdot n_x + g \cdot n_y) \, ds = 0, \quad (4.1b)$$

where  $V_{ij}$  is the volume of cell  $\Omega_{i,j}$  and  $q_{ij}$  is the mean value of  $q$  over  $\Omega_{i,j}$ . Further we introduce the notation

$$\int_{\Gamma_{ijk}} (f \cdot n_x + g \cdot n_y) \, ds = f_{ijk} \cdot s_{ijk}, \quad (4.2)$$

where  $s_{ijk}$  is the length of  $\Gamma_{ijk}$  and  $f_{ijk}$  is the mean flux outward  $\Omega_{i,j}$  over the side  $\Gamma_{ijk}$ . It is easy to see that, if  $\Omega_{i,j}$  and  $\Omega_{i',j'}$  are neighbours with a common side

$$\Gamma_{ijk} = \Gamma_{i'j'k'},$$

then  $f_{ijk} = -f_{i'j'k'}$ . The space discretisation of (3.4) is done according to the Godunov principle: the state  $q(t, x, y)$  is approximated by  $q_{ij}(t)$  for all  $\Omega_{i,j}$  and the mean fluxes  $f_{ijk}$  are approximated from the states in the adjacent cells. For this purpose, a computed flux  $f_{ijk}(q_{ij}^k, q_{ijk}^k)$  is introduced to replace  $f_{ijk}$ . Here,  $q_{ij}^k$  and  $q_{ijk}^k$  are approximations of  $q$  at both sides of  $\Gamma_{ijk}$ . Thus we obtain the semi-discretisation of (3.4):

$$V_{ij} \frac{\partial}{\partial t} q_{ij} = - \sum_k s_{ijk} f_{ijk}(q_{ij}^k, q_{ijk}^k), \quad (4.3)$$

and for the steady equations we obtain the discrete system of equations

$$N_h(q_h) = 0, \quad (4.4)$$

which is short for

$$(N_h(q_h))_{ij} := \sum_k s_{ijk} f_{ijk}(q_{ij}^k, q_{ijk}^k) = 0 \quad \forall i, j.$$

Notice that  $N_h$  can be seen as a mapping between two discrete Banach spaces:  $N_h : X_h \rightarrow Y_h$ .

If the cell  $\Omega_{ij}$  is adjacent to the boundary of  $\Omega$ , i.e.  $\Gamma_{ijk} \subset \partial\Omega$ , then a state  $q_{ijk}$  is possibly not available. In that case  $f_{ijk}$  is computed from  $q_{ij}$  and the boundary conditions at  $\Gamma_{ijk}$ .

The main difficulty in the discretisation of (4.2) is the construction of a proper approximation  $f_{ijk}$  for a given  $q_{ij}$  and  $q_{ijk}$ . A possible approach is to consider the state  $q(t, x, y)$  at  $t = t_0$  as piecewise constant over the cells  $\Omega_{ij}$  and to compute (approximately) the fluxes over the walls as a quasi one-



dimensional problem during a small time  $(t_0, t_0 + \Delta t)$ , by solving the Riemann-problem for gasdynamics [18, 64]. These fluxes are used as  $f_{ijk}(q_{ij}, q_{ijk})$ . Approximate Riemann-solvers have been proposed by Steger-Warming [68], Van Leer [18, 46, 47], Roe [60, 61], Osher [56, 58] and others. An overview of upwind schemes has been given in [13].

The possible irregularity of the mesh is easily dealt with by making use of the invariance of the Euler equations under rotation of the coordinate system. Let the normal of a skew wall  $\Gamma_{ijk}$ , directed from  $\Omega_{ij}$  to  $\Omega_{ijk}$ , be given by  $(n_1, n_2) = (\cos\phi_{ijk}, \sin\phi_{ijk})$ , then the simple local rotation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} n_1 & n_2 \\ -n_2 & n_1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

reduces the computation of  $f_{ijk}$  to the approximate solution of the one-dimensional Riemann problem in the  $x$ -direction, i.e.

$$f_{ijk} := f_{ijk}(q_{ij}^k, q_{ijk}^k) = T_{ijk}^{-1} f(T_{ijk} q_{ij}, T_{ijk} q_{ijk}), \quad (4.5)$$

where

$$T_{ijk} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & n_1 & n_2 & 0 \\ 0 & -n_2 & n_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The function  $f(\dots)$  is called the *numerical flux function*. We see that the quantities  $s_{ijk}$  and  $\phi_{ijk}$  are the only geometrical data about the mesh, needed to set up equation (4.4). Handling an irregular mesh by this finite volume approach, there is no need to introduce a transformation for the equations. They remain simply in their form (3.1). Further it is immediately clear that - in this way - the discrete system is fully conservative, also for the non-uniform mesh.

An additional advantage of this finite volume approach is that we can easily set up the residual  $N_h(q_h)$  and its linearisation  $dN_h(q_h)/dq_h$  by assembling the contributions that are computed for each cell wall separately. This assembling procedure is completely analogous to the finite element technique, where the construction of the load vector and the stiffness matrix is done by assembling the element stiffness matrices.

#### 4.2 OSHER'S APPROXIMATE RIEMANN SOLVER

In these lectures Osher's approximate Riemann-solver is used for the numerical flux  $f(q_0, q_1)$  in (4.5). In the remainder of this section we give a short description of this function. In fact, we may distinguish two strongly related variants of it: the O-(original) variant and the P-(physical) [26] variant. Here we restrict ourselves to the P-variant. The advantages of the Osher discretisation procedure can be found e.g. in [56, 58]. It is our experience that it yields very reliable discretisations. Its main disadvantage seems its supposed complexity when compared with other approximate Riemann solvers. An objective of our exposition is to show that the scheme can be implemented in a simple and straightforward way. Further, we need this description for reference and to show (in section 4.4) how its linearisation is obtained.

According to Osher, the numerical flux function in (4.5) is defined by

$$f(q_0, q_1) = \frac{1}{2} \{f(q_0) + f(q_1) - \int_{q_0}^{q_1} |f_q(w)| dw\}, \quad (4.6)$$

where  $|f_q(w)|$  is the absolute value of the matrix  $f_q(w)$ , as defined by

$$|f_q(w)| := R |\Lambda| R^{-1}.$$

Here  $|\Lambda|$  is the diagonal matrix of the absolute values of the eigenvalues  $\lambda$  of  $f_q(w)$ . These

eigenvalues form the diagonal matrix  $\Lambda$  in the eigenvalue- eigenvector decomposition

$$f_q(w) = R \Lambda R^{-1}.$$

In (4.6) the integration path is still to be defined, but we know that the matrix has a complete set of eigenvalues  $\lambda_k$  viz.  $\lambda_1 = u - c$ ,  $\lambda_2 = \lambda_3 = u$ ,  $\lambda_4 = u + c$ , (where  $c = \sqrt{\gamma p / \rho}$  is the speed of sound) and a set of 3 corresponding eigenspaces  $R_1, R_{2,3}$  and  $R_4$ .

The integral  $\int_{q_0}^{q_1} |f_q(w)| dw$  is computed along a path  $q = q(s)$ ,  $0 \leq s \leq 1$ ,  $q(0) = q_0$ ,  $q(1) = q_1$ . This path is divided into subpaths  $\Gamma_k$ ,  $k = 1, 2, 3$ , connecting the states  $q_{(k-1)/3}$  and  $q_{k/3}$ . These subpaths  $\Gamma_k$  are constructed such that on  $\Gamma_k$  the direction of the path  $\frac{\partial q(s)}{\partial s}$  is tangential to  $R_{m(k)}$ , an eigenvector. Feasible choices for  $R_{m(k)}$  are  $k = 1$ :  $R_{m(k)} = R_1$ ;  $k = 2$ :  $R_{m(k)} = R_{2,3}$ ;  $k = 3$ :  $R_{m(k)} = R_4$ . (These are the choices made in the P-variant, other choices are made for the O-variant.)

The states  $q_{1/3}$  and  $q_{2/3}$  are determined by means of the Riemann invariants  $\psi_l^{m(k)}(q(s))$ ,  $l \neq m$ ,  $l = 1, 2, 3, 4$ , which are invariant quantities along  $\Gamma_k$ . These  $\psi_l^m(q)$ ,  $m = 1, 2, 3, 4$  are

$$\begin{aligned} \psi_3^4 &= \psi_3^1 = v, \\ \psi_1^4 &= \psi_4^1 = z, \\ \psi_2^1 &= u + \frac{2}{\gamma-1} c, \\ \psi_2^4 &= u - \frac{2}{\gamma-1} c, \\ \psi_1^2 &= \psi_1^3 = u, \\ \psi_4^2 &= \psi_4^3 = p, \end{aligned} \tag{4.7}$$

where  $z = \ln(p\rho^{-\gamma})$ . Thus,  $q_{1/3}$  and  $q_{2/3}$  are determined from  $q_0$  and  $q_1$  by the equations

$$\psi_l^{m(k)}(q_{(k-1)/3}) = \psi_l^{m(k)}(q_{k/3}), \quad k = 1, 2, 3, \quad l \neq m(k).$$

These are 8 equations for the 8 unknowns in  $q_{1/3}$  and  $q_{2/3}$ .

Expressing the state  $q$  in the dependent variables  $u, v, c$  and  $z$ , we obtain directly  $z_{1/3} = z_0$ ,  $z_{2/3} = z_1$ ,  $v_{1/3} = v_0$ ,  $v_{2/3} = v_1$ .

Introducing  $\alpha = \exp((z_1 - z_0)/(2\gamma))$ ,  $p_{1/3} = p_{2/3}$  leads to

$$\frac{c_{2/3}}{c_{1/3}} = \exp\left[\frac{z_{2/3} - z_{1/3}}{2\gamma}\right] = \alpha, \tag{4.8}$$

and we arrive at the linear system

$$\begin{aligned} u_{1/3} + \frac{2}{\gamma-1} c_{1/3} &= u_0 + \frac{2}{\gamma-1} c_0 =: \Psi_0, \\ u_{2/3} - \frac{2}{\gamma-1} c_{2/3} &= u_1 - \frac{2}{\gamma-1} c_1 =: \Psi_1, \\ c_{2/3} &= \alpha c_{1/3}, \\ u_{2/3} &= u_{1/3}. \end{aligned} \tag{4.9}$$

A meaningful solution exists as long as no cavitation occurs ( $\Psi_0 > \Psi_1$ ).

This system is easily solved as

$$\begin{aligned} c_{1/3} &= \frac{\gamma-1}{2} \frac{\Psi_0 - \Psi_1}{1 + \alpha}, \\ c_{2/3} &= \alpha c_{1/3}, \end{aligned} \tag{4.10}$$

$$u_{1/2} := u_{1/3} = u_{2/3} = \frac{\Psi_1 + \alpha \Psi_0}{1 + \alpha}.$$

The relevant eigenvalues at the points  $q_{k/3}$ ,  $k = 1, 2, 3$ , are

$$\begin{aligned}\bar{\lambda}_0 &:= \lambda_{m(1)}(q_0) = u_0 - c_0, \\ \bar{\lambda}_{1/3} &:= \lambda_{m(1)}(q_{1/3}) = u_{1/3} - c_{1/3}, \\ \bar{\lambda}_{1/2} &:= \lambda_{m(2)}(q_{1/3}) = \lambda_{m(2)}(q_{2/3}) = u_{1/3} = u_{2/3}, \\ \bar{\lambda}_{2/3} &:= \lambda_{m(3)}(q_{2/3}) = u_{2/3} + c_{2/3}, \\ \bar{\lambda}_1 &:= \lambda_{m(3)}(q_1) = u_1 + c_1.\end{aligned}\tag{4.11}$$

Because  $\lambda_{1,4}$  are genuinely nonlinear eigenvalues,  $\lambda_{m(k)}$  is monotonous along  $\Gamma_k$ ,  $k = 1, 3$  and  $\lambda_{m(k)}(q(s))$  changes sign at most once along these  $\Gamma_k$ . E.g. a sonic point  $q_{s1}$  with  $\lambda_{m(1)}(q(s_1))$  exists on  $\Gamma_1$  if  $\bar{\lambda}_0 \cdot \bar{\lambda}_{1/3} \leq 0$ . This sonic point is computed from the linear system

$$\begin{aligned}v_s &= v_0, & u_s - c_s &= 0, \\ z_s &= z_0, & u_s + \frac{2}{\gamma - 1} c_s &= \Psi_0.\end{aligned}\tag{4.12}$$

Similarly, a sonic point  $q_{s2}$  is found on  $\Gamma_3$  if  $\bar{\lambda}_{2/3} \cdot \bar{\lambda}_1 \leq 0$ .

Along the path  $q(s)$ ,  $0 \leq s \leq 1$ ,  $\lambda_{m(k)}(q(s))$  may change sign only at the points  $q_{1/3}$  or  $q_{2/3}$  and eventually at a sonic point  $q_{s1}$  or  $q_{s2}$ .

Thus from (4.6) we obtain

$$\begin{aligned}f(q_0, q_1) &= f(q_0) (\text{sign}(\bar{\lambda}_0) + 1) / 2 \\ &+ f(q_{s1}) (\text{sign}(\bar{\lambda}_{1/3}) - \text{sign}(\bar{\lambda}_0)) / 2 \\ &+ f(q_{1/3}) (\text{sign}(\bar{\lambda}_{1/2}) - \text{sign}(\bar{\lambda}_{1/3})) / 2 \\ &+ f(q_{2/3}) (\text{sign}(\bar{\lambda}_{2/3}) - \text{sign}(\bar{\lambda}_{1/2})) / 2 \\ &+ f(q_{s2}) (\text{sign}(\bar{\lambda}_1) - \text{sign}(\bar{\lambda}_{2/3})) / 2 \\ &+ f(q_1) (1 - \text{sign}(\bar{\lambda}_1)) / 2.\end{aligned}\tag{4.13}$$

In most cases, many eigenvalues  $\bar{\lambda}$  will have equal signs and  $f(q_0, q_1)$  is computed as the sum of only a few  $f(q)$ . Further we notice that  $f(q_0, q_1)$  is a continuous function in all  $\bar{\lambda}$ 's and we see  $\bar{\lambda}_{1/3} < \bar{\lambda}_{1/2} < \bar{\lambda}_{2/3}$ . Because of this continuity we may neglect the case of a zero eigenvalue  $\bar{\lambda}$  and we compute the numerical flux as

$$\begin{aligned}f(q_0, q_1) &= \text{if } \bar{\lambda}_0 > 0 \text{ then } f(q_0) \\ &+ \text{if } \bar{\lambda}_0 \cdot \bar{\lambda}_{1/3} < 0 \text{ then } \text{sign}(\bar{\lambda}_{1/3}) \cdot f(q_{s1}) \\ &+ \text{if } \bar{\lambda}_{1/3} \cdot \bar{\lambda}_{1/2} < 0 \text{ then } f(q_{1/3}) \\ &+ \text{if } \bar{\lambda}_{1/2} \cdot \bar{\lambda}_{2/3} < 0 \text{ then } f(q_{2/3}) \\ &+ \text{if } \bar{\lambda}_{2/3} \cdot \bar{\lambda}_1 < 0 \text{ then } \text{sign}(\bar{\lambda}_1) \cdot f(q_{s2}) \\ &+ \text{if } \bar{\lambda}_1 < 0 \text{ then } f(q_1).\end{aligned}\tag{4.14}$$

This expression seems rather complex. However, if the ordered sequence  $\bar{\lambda}_0, \bar{\lambda}_{1/3}, \bar{\lambda}_{1/2}, \bar{\lambda}_{2/3}, \bar{\lambda}_1$  can be split in two parts (possibly empty), the first of which contains only negative and the second only positive signs, then a  $\hat{q}$  exists such that simply  $f(q_0, q_1) = f(\hat{q})$ . We identify this state  $\hat{q}$  as the state of the gas at the cell wall. This situation occurs for the supersonic cases, on a sonic line and for subsonic flow. If we exclude the unlikely cases that  $u_{1/2} < 0$  and  $u_0 - c_0 > 0$ , or  $u_{1/2} > 0$  and  $u_1 + c_1 < 0$ ,

the numerical fluxes near a shock are the only ones for which  $f(q_0, q_1)$  is found to be a sum of more (viz. 3) terms  $f(q)$ . For more details we refer to [26, 66].

#### 4.3 THE NUMERICAL FLUX AT THE BOUNDARY

The flux of the conservative variables  $f_{ijk}$ , at the boundary of the domain  $\Omega$  is partially determined by  $q_{ij}$ , the state of the flow near the boundary and partially by the boundary conditions. To compute the value of these  $f_{ijk}$  we determine first the state  $q_B = q_{ijk}$  at the boundary  $\delta\Omega$ , depending on  $q_{ij}$  and on the boundary conditions. Then  $f(q_{ij}, q_B)$ , as described in section 4.2, is used to compute the boundary flux.

In order to see what boundary conditions are required at the boundary for a properly posed problem, we first consider a time-dependent one-dimensional problem on a half-line

$$\frac{\partial}{\partial t} + \frac{\partial}{\partial x} f(q) = 0, \quad t \geq 0, \quad x \geq 0. \quad (4.15)$$

In quasi-linear form we write (4.15) as

$$q_t + A(q) q_x = 0, \quad (4.16)$$

where  $A(q) = df/dq$ .

For the hyperbolic system (4.16), a complete set of real eigenvalues  $\Lambda(q)$  and linearly independent eigenspaces  $R(q)$  exists and we obtain

$$q_t + R(q) \Lambda(q) R^{-1}(q) q_x = 0. \quad (4.17)$$

Assuming the existence of a  $w(q)$  such that

$$\frac{dw}{dq} = R^{-1}(q), \quad (4.18)$$

we find the uncoupled system

$$w_t + \Lambda(w) w_x = 0. \quad (4.19)$$

Clearly, for any component  $w_i$  for which  $\lambda_i \leq 0$ , the value  $w_i(t, 0)$ ,  $t \geq 0$ , is determined by  $w_i(0, x)$ ,  $x \geq 0$ . For these components the characteristics leave the domain  $x > 0$ . However, for components for which  $\lambda_i > 0$ , characteristics enter the domain and boundary conditions are to be given; i.e. for each  $\lambda_i > 0$  a boundary condition  $B_i(w, t) = 0$  is required and the complete set of conditions should yield a non-singular  $dB_i/dw_j$  for all variables  $w_j$  for which  $\lambda_j > 0$ . Returning to the original dependent variables  $q$ , this means that a set boundary conditions  $B_i(q, t) = 0$  is required such that

$$\frac{dB_i}{dq} \cdot \frac{dq}{dw_j} = \frac{dB}{dq} R^+(q) \quad (4.20)$$

is non-singular.

$R(q) = dq/dw$  is the set of right eigenvectors of  $A(q)$  and  $\{dq/dw_j \mid \lambda_j > 0\} = R^+(q)$  is the rectangular matrix of eigenvectors corresponding to the positive eigenvalues.

We notice that, for the discretisation of the 2-D problem (3.4) near the boundary, the boundary conditions are considered as locally one-dimensional. This is completely consistent with the discretisation over internal cell walls as treated in section 4.2.

To satisfy the boundary conditions in the discrete equations (4.4) we determine  $q_B$ , the state at the boundary, such that it satisfies the boundary conditions, i.e.  $B(q_B) = 0$ , and the equality

$$f_{ijk} = f(q_B) = f(q_B, q_{ij}). \quad (4.21)$$

In view of (4.6) the equality (4.21) implies

$$\int_{q_B}^{q_{ij}} f_q(w) dw = \int_{q_B}^{q_{ij}} |f_q(w)| dw, \quad (4.22)$$

i.e.  $q_B$  should satisfy the boundary conditions and should be connected with  $q_{ij}$  by a path  $q(s)$  such that

$$\lambda_{m(k)}(q(s)) \geq 0. \quad (4.23)$$

Such a path can be constructed again as a sum of subpaths along eigenvectors, as described in section 4.2 for  $q_{ij}$  and  $q_{ijk}$ . Now only the eigenvectors corresponding to the positive eigenvalues can be used and the number of subpaths depends on the type of the boundary conditions (i.e. depends on the number of ingoing characteristics). The endpoints of  $\Gamma_k$  are computed by means of the Riemann invariants (as in section 4.2) and the boundary data.

#### 4.4 THE LINEARISATION OF OSHER'S SCHEME

Both in the case of a complete linearisation of the discrete system (4.4) as well as in the case where only local linearisation is applied in a nonlinear relaxation method, we need convenient expressions for  $dN_h(q_h)/dq_h$ . From (4.4) we obtain

$$\begin{aligned} \frac{\partial(N_h(q_h))_{ij}}{\partial q_{lm}} &= \frac{\partial}{\partial q_{lm}} \sum_k f_{ijk}(q_{ij}, q_{ijk}) s_{ijk} \\ &= \sum_k s_{ijk} \frac{\partial}{\partial q_{lm}} f_{ijk}(q_{ij}, q_{ijk}) = \\ &= \sum_k s_{ijk} \frac{\partial}{\partial q_{ij}} f_{ijk}(q_{ij}, q_{ijk}) \quad \text{if } \Omega_{lm} = \Omega_{ij}, \end{aligned} \quad (4.24a)$$

$$= s_{ijk} \frac{\partial}{\partial q_{ijk}} f_{ijk}(q_{ij}, q_{ijk}) \quad \text{if } \Omega_{lm} = \Omega_{ijk}, \quad (4.24b)$$

$$= 0 \quad \text{otherwise.} \quad (4.24c)$$

Now, in view of (4.5), the computation of  $dN_h(q_h)/dq_h$  reduces to evaluations of

$$f'_{(0)}(q_0, q_1) = \frac{\partial}{\partial q_0} f(q_0, q_1) \quad \text{and} \quad f'_{(1)}(q_0, q_1) = \frac{\partial}{\partial q_1} f(q_0, q_1).$$

A matrix  $dN_h(q_h)/dq_h$  can be assembled per cell wall as explained for  $N_h(q_h)$  in section 4.2.

If in (4.24a)  $q_{ijk} = q_B$  is a boundary state, then a relation  $q_{ijk} = q_B(q_{ij})$  exists and the corresponding term in (4.24a) is to be read as

$$\begin{aligned} s_{ijk} \frac{d}{dq_{ij}} f_{ijk}(q_{ij}, q_{ijk}) &= s_{ijk} \frac{d}{dq_{ij}} f_{ijk}(q_{ij}, q_B(q_{ij})) \\ &= s_{ijk} \frac{d}{dq_{ij}} \left\{ T^{-1} f(Tq_{ij}, Tq_B(q_{ij})) \right\} \\ &= s_{ijk} T^{-1} f'_{(0)}(Tq_{ij}, Tq_B) T + s_{ijk} T^{-1} f'_{(1)}(Tq_{ij}, Tq_B) T \frac{dq_B}{dq_{ij}}. \end{aligned} \quad (4.25)$$

Here  $T$  denotes  $T_{ijk}$  as in eq. (4.5). The derivatives  $dq_B/dq_{ij}$  depend on the type of boundary condition and are derived in each case from the relations  $q_B(q_{ij})$  as described in section 4.3.

We noticed already that the integration paths are easily expressed in the dependent variables  $u, v, c$  and  $z$ . The numerical flux and its partial derivatives are also conveniently expressed in these variables. The flux vector  $f = (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T$  is found as a function of  $q = (c, u, v, z)^T$  by noting that

$$\begin{aligned} \rho &= \left( e^{-z} c^2 / \gamma \right)^{\frac{1}{\gamma-1}}, \\ p &= \rho c^2 / \gamma, \end{aligned}$$

$$E = \rho(u^2 + v^2)/2 + \frac{\rho c^2}{\gamma(\gamma - 1)}.$$

In these variables the Jacobian matrix of the flux

$$\frac{df}{dq} = \frac{\partial(\rho u, \rho u^2 + p, \rho uv, u(E + p))}{\partial(c, u, v, z)}$$

reads

$$f'(q) = \frac{df}{dq} = \begin{pmatrix} \beta \rho u / c & \rho & 0 & -\beta \rho u / 2 \\ \beta \rho (u^2 + c^2) / c & 2 \rho u & 0 & -\beta (\rho u^2 + p) / 2 \\ \beta \rho uv / c & \rho v & \rho u & -\beta \rho uv / 2 \\ \beta u (E + p + \rho c^2) / c & \rho u^2 + E + p & \rho uv & -\beta u (E + p) / 2 \end{pmatrix}, \quad (4.26)$$

where  $\beta = 2/(\gamma - 1)$ . In terms of this matrix, from (4.14) follows

$$\begin{aligned} \frac{\partial}{\partial q_0} f(q_0, q_1) = & \text{if } \bar{\lambda}_0 > 0 \quad \text{then } f'(q_0) \\ & + \text{if } \bar{\lambda}_0 \cdot \bar{\lambda}_{1/3} < 0 \text{ then } \text{sign}(\bar{\lambda}_{1/3}) f'(q_{s1}) \frac{\partial q_{s1}}{\partial q_0} \\ & + \text{if } \bar{\lambda}_{1/3} \cdot \bar{\lambda}_{1/2} < 0 \text{ then } \pm f'(q_{1/3}) \frac{\partial q_{1/3}}{\partial q_0} \\ & + \text{if } \bar{\lambda}_{1/2} \cdot \bar{\lambda}_{2/3} < 0 \text{ then } \pm f'(q_{2/3}) \frac{\partial q_{2/3}}{\partial q_0}. \end{aligned} \quad (4.27)$$

The derivatives  $\partial q / \partial q_0$ ,  $q = q_{s1}, q_{1/3}, q_{2/3}$ , are derived from the differentiable relations (4.8)-(4.12). The explicit expressions are found in [26] and [66].

In this way the matrices  $f'_{(0)}(q_0, q_1)$  and  $f'_{(1)}(q_0, q_1)$  are readily computed. It appears that both Jacobians are continuous functions of  $q_0$  and  $q_1$  as long as  $\bar{\lambda}_{1/2} = u_{1/3} = u_{2/3} \neq 0$ . An efficient implementation is possible; for this it is profitable that the fluid state is (remains) expressed in the state variables  $c, u, v$  and  $z$ .

#### 4.5. MULTIGRID ITERATION

In order to solve (4.4), we first generalise the problem slightly to

$$N_h(q_h) = r_h. \quad (4.28)$$

We use iteration with the full approximation scheme (FAS). For this we need a sequence of discretisations

$$N_h(q_h), \quad \text{with } h_0 > h_1 > \dots > h_l = h.$$

For the mesh width  $h_{i-1}$  we take  $h_{i-1} = 2h_i$ . For an irregular mesh we delete each second line of mesh points to obtain the coarser grid.

As explained in section 2, one FAS cycle for the solution of (4.28) consists of the steps:

- (0) start with an approximate solution  $q_h$ ;
- (1) improve  $q_h$  by application of  $p$  nonlinear (pre-) relaxation iterations to  $N_h(q_h) = r_h$ ;
- (2) compute the residual  $N_h(q_h)$ ;
- (3) find an approximation of  $q_h$  on the next coarser grid, say  $q_{2h}$ . ( Either use a restriction  $q_{2h} = R_{2h,h} q_h$ , or use another previously obtained approximation  $q_{2h}$  );
- (4) compute

$$r_{2h} = N_{2h}(q_{2h}) + \bar{R}_{2h,h}(r_h - N_h(q_h)) ;$$

(5) approximate the solution of

$$N_{2h}(q_{2h}) = r_{2h} \quad (4.29)$$

by application of  $\sigma$  FAS cycles. The result is  $\tilde{q}_{2h}$ ;

(6) correct the current solution by

$$q_h := q_h + P_{h,2h}(\tilde{q}_{2h} - q_{2h});$$

(7) improve  $q_h$  by application of  $q$  nonlinear (post-) relaxation iterations to  $N_h(q_h) = r_h$ .

The steps (2)-(6) in this process are the *coarse grid correction*. These steps are skipped on the coarsest grid  $h_0$ . For the solution of the nonlinear system (4.4), FAS iteration is simply applied with  $r_h = 0$  on the finest grid. During the FAS iteration, on the coarser grids, non-zero right-hand sides appear in (4.29).

In order to complete the description of the FAS-cycle we need to be explicit about:

- (1) the choice of the operators  $N_{2h}$ ,  $P_{h,2h}$ ,  $\bar{R}_{2h,h}$  and eventually  $R_{2h,h}$ ;
- (2) the FAS strategy, i.e. the numbers  $p$ ,  $q$ ,  $\sigma$ ;
- (3) the nonlinear relaxation method.

#### 4.5.1 A nested sequence of Galerkin discretisations.

For the operators  $P_{h,2h}$  and  $\bar{R}_{2h,h}$  we make a choice that is consistent with the concept of our finite volume discretisation. This discretisation is essentially a weighted residual method, where the solution is approximated by a piecewise constant function (on cells  $\Omega_{ij}$ ) and where the residual is weighted by characteristic functions on all  $\Omega_{ij}$ . From this point of view, it is natural to use a piecewise constant interpolation for  $P_{h,2h}$  and to use addition over subcells for  $\bar{R}_{2h,h}$ . Notice that  $\bar{R}_{2h,h} = P_{h,2h}^T$ . With these choices it is clear that

$$N_{2h}(q_{2h}) = \bar{R}_{2h,h} N_h(P_{h,2h} q_{2h}), \quad (4.30)$$

i.e. the coarse grid finite volume discretisation is a formal Galerkin approximation of the fine grid finite volume discretisation. Using (4.30) on all different levels we obtain a nested sequence of discretisations, i.e. the following scheme of operators and spaces is commutative.

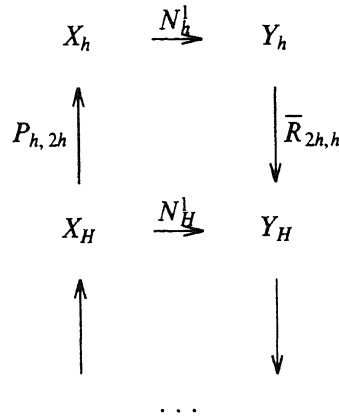


Fig. 4.1 Nested sequence of discretisations

The effect of the Galerkin approximation  $N_{2h} = \bar{R}_{2h,h} N_h P_{h,2h}$  on the approximate solution  $\tilde{q}_h$  obtained after a coarse grid correction is the following. If we take  $q_{2h} = R_{2h,h} q_h$  in step (3) of the algorithm, with  $R_{2h,h}$  such that  $R_{2h,h} P_{h,2h} = I_{2h}$  is the identity operator on  $X_{2h}$ , and if (4.29) is solved exactly, then

$$\begin{aligned} & \bar{R}_{2h,h} \left[ r_h - N_h P_{h,2h} R_{2h,h} \tilde{q}_h \right] \\ &= \bar{R}_{2h,h} \left[ N_h q_h - N_h P_{h,2h} R_{2h,h} q_h \right], \end{aligned}$$

or, for the restriction of the residual

$$\begin{aligned} & \bar{R}_{2h,h} \left[ r_h - N_h(\tilde{q}_h) \right] \\ &= \bar{R}_{2h,h} \left[ \left( N_h q_h - N_h P_{h,2h} R_{2h,h} q_h \right) - \left( N_h \tilde{q}_h - N_h P_{h,2h} R_{2h,h} \tilde{q}_h \right) \right]. \end{aligned} \tag{4.31}$$

In the neighbourhood of a solution, the difference  $q_h - \tilde{q}_h$  will be small and  $N_h$  will approximately behave as a linear function: the restriction of its residual will be very small, viz.  $\mathcal{O}(\|q_h - \tilde{q}_h\|^2)$ . For a smooth operator  $N_h$ , this implies

$$\bar{R}_{2h,h} [r_h - N_h(\tilde{q}_h)] = \mathcal{O}(\|q_h - \tilde{q}_h\|^2).$$

Because  $\bar{R}_{2h,h}$  is an addition over 4 neighbouring cells, this means that the restriction of the residual mainly contains high frequency components. A small restriction of the residual means that possible large residuals over neighbouring cells cancel: the residual is rapidly varying. Local relaxation methods should be able to eliminate such residuals efficiently.

#### 4.5.2 Multigrid strategy.

Experience with multigrid algorithms in another context makes it plausible that  $p = q = \sigma = 1$  is a good choice for a strategy. This is the choice mainly used in our experiments. Other choices with small values for  $p$ ,  $q$  and  $\sigma$  can be made. What is best depends much on the relaxation used, and research can be made seeking the most efficient combination. Up to now, it appears that different  $(p, q, \sigma)$ -strategies are not much different in efficiency. Usually a smaller convergence factor is compensated by a corresponding amount of additional work.

#### 4.5.3 Relaxation.

Clearly, whether a sequence of Galerkin approximations is used or not, the important feature for a relaxation method in a multiple grid context (both linear and nonlinear) is its capability to damp the high frequency components in the error (or in the residual). Therefore the difference scheme should be sufficiently dissipative. The first-order upwind schemes usually are. An advantage of these schemes over central differences is that this numerical dissipation is well defined and independent of an artificial parameter for the added dissipation, which is necessary for the central difference schemes.

For the relaxation method we may consider several alternatives. For nonlinear MG methods they are all of the collective Gauss-Seidel (GS) type, where for each cell the 4 variables  $(u, v, c, z)$  are recomputed simultaneously. For the solution of these nonlinear  $4 \times 4$  systems, one or more steps of a Newton-iteration are used until the local residual is reduced below a specified amount. In almost all cases it appeared most efficient to take this tolerance so crude that no more than one iteration step per point (= volume) is performed.

Possible relaxations are: (1) LEX: GS-relaxation with lexicographical ordering; (2) SGS1: symmetric Gauss-Seidel from north-west to south-east and vice versa; (3) SGS2: the same from north-east to south-west; (4) RB: using a checkerboard ordering of the points.

In almost all cases the same relaxation can be used in both step (1) and (7) of the algorithm. Another good choice is SGS3: to use SGS1 for the pre- and SGS2 for the post-relaxation. In[25] we compared some of these relaxations in combination with a uniform grid. There also the effect of



other strategies  $(p, q, \sigma)$  was considered. For a standard model problem and a non-uniform grid, results of such an MG-procedure are shown in Fig.4.2-4.4.

The smoothing behaviour of these relaxations can be analyzed by Local Mode Analysis. Here we should notice that the *smoothing factor* as used for common elliptic problems, has no significant meaning for the Euler equations because we have to take into account characteristic (unstable) modes. A local mode analysis should follow more the lines used for elliptic singular perturbation problems, cf. e.g. [38]. Jespersen [31] has published some results. He shows that for a subsonic and a supersonic case SGS has a reasonably good smoothing behaviour, when applied to a first-order scheme. Of course, the non-symmetric GS relaxation is only effective if the direction of the relaxation sufficiently conforms with the direction of the characteristics. If we study plots of reduction factors obtained by Local Mode Analysis (spectral radii, or norms for the error/ residual amplification operator), e.g. when SGS is applied to the Euler equations, we see that two SGS sweeps are usually sufficient for a significant reduction of the high frequencies [Hemker, unpublished results]. For second-order schemes the smoothing rates are not satisfactory.

#### 4.5.4 Initial estimates.

For the nonlinear multigrid as described above, it is important to start with reasonably good initial estimates. Since we do not want to provide sophisticated a priori estimates, we can use the FMG technique to compute the estimates.

In many cases, in the FMG-method a very crude initial estimate on the coarsest grid is used - e.g. a uniform flow satisfying the inlet and outlet boundary conditions -. To obtain a first estimate on each finer level, first the solution on the coarser grid is improved by a single FAS cycle and then the approximation is interpolated to the finer grid. These steps are repeated on the finer levels until the finest level has been reached (cf. section 2.3).

The interpolation used to obtain the first guess on each level should be of high enough order to comply with the accuracy of the discretisation. In our case, where the discretisation is of first order, the first-order prolongation  $P_{h, 2h}$  as used in the Galerkin approximation is not accurate enough, and a second-order bilinear interpolation is necessary.

#### 4.6. CONCLUSION

For transonic computations [25, 26, 39] we have seen that real multigrid efficiency can be obtained for the steady Euler equations, i.e. the rate of convergence for FAS iteration is large ( $\approx 0.3/\text{FAScycle}$ ) and almost independent of the number of cells in the mesh. A good sequence of discretisations is obtained by the consistent use of the finite volume technique. It yields a conservative discretisation and it prescribes both the prolongations and the restrictions for the MG algorithm. The result is a nested sequence of Galerkin discretisations.

Probably the most important ingredient in the finite volume discretisation is the choice of a good numerical flux function. A slight variant of Osher's approximate Riemann-solver appears to be a very reliable choice. The reason for its excellent performance might be the fact that it allows a completely consistent treatment of the interior and the boundaries of the domain. Both at the domain boundaries and in the interior, the appropriate Riemann invariants are used to transfer information over cell boundaries. Further, the numerical flux has smooth derivatives, which avoids problems when Newton's method is used in the relaxation.

By the use of the FMG technique, sufficiently accurate initial estimates could be obtained (for about the work of  $1/3$  FAS-cycle). For some interesting problems [26, 39], only a single FAS iteration (with  $p = q = \sigma = 1$ , SGS3-relaxation) appears to be sufficient for obtaining truncation error accuracy. This means that these (non-isenthalpic and non-isentropic) steady Euler problems can be solved by an amount of work that is equivalent with about  $(4/3) \times 2$  nonlinear symmetric Gauss-Seidel relaxations sweeps.

## 5. DEFECT CORRECTION FOR HIGHER ORDER EULER COMPUTATIONS

### 5.1 SECOND ORDER DISCRETISATION

The first-order discretisation discussed in section 4.1 has a number of advantages: it is conservative, monotonous and it gives a sharp representation of discontinuities (shocks and contact discontinuities), as long as these are aligned with the mesh. Further it allows an efficient solution of the discrete equations by a multigrid method. Disadvantages are: the low order of accuracy (many points are required to find an accurate representation of a smooth solution) and the fact that it is highly diffusive for oblique discontinuities (the discontinuities are smeared out over a large number of cells). For a first-order (upwind) scheme these are well-known facts and it leads to the search for higher-order methods.

A key property of the discretisation, that we want to preserve in a second-order scheme, is the conservation of  $q$ , because it allows discontinuities to be captured as weak solutions of (3.1) and avoids the necessity of a shock fitting technique. Therefore, we consider only schemes that are still based on (4.4), and we select  $f_{ijk}(q_{ij}^k, q_{ijk}^k)$  that yield a better approximation to (4.2) than (4.5).

The higher-order schemes can be obtained in two different ways. Higher order interpolation is used either for the states (i.e. in  $X_h$ ) or for the fluxes (i.e. in  $Y_h$ ). The first approach, also called the MUSCL approach, is used e.g. in [2, 12, 48] the second in [57, 67]. In the first case, in (4.4)  $q_{ij}^k$  and  $q_{ijk}^k$  are obtained by some interpolation from  $q_h = \{q_{ij}\}$ . In the latter,  $f_{ijk}(q_{ij}^k, q_{ijk}^k)$  is obtained from  $\{f_{ijk}(q_{ij}^k, q_{ijk}^k)\} \cup \{f_{ijk}(q_{ij}^k, q_{ij}^k)\}$ . In the following we restrict ourselves to the MUSCL approach.

From the point of view of finite volume discretisation, a straightforward way to form a more accurate approximation is to replace the first-order approximation (4.5) by a second-order one. Instead of the piecewise constant approximation  $\tilde{q}(x, y)$  over cells, we may consider a piecewise bilinear function  $\tilde{q}(x, y)$  on a set of  $2 \times 2$  cells (a *superbox*). Such a superbox on the  $h$ -level corresponds with a single cell at the  $2h$ -level. Over the boundaries of the superbox  $\tilde{q}(x, y)$  can be discontinuous; in the superbox  $\tilde{q}(x, y)$  is determined by  $q_{2i, 2j}$ ,  $q_{2i+1, 2j}$ ,  $q_{2i, 2j+1}$  and  $q_{2i+1, 2j+1}$ . Using such a bilinear function, we see that the central difference approximation is used for flux computations inside the superboxes; at superbox boundaries interpolation is made from the left and the right and the approximate Riemann solver is used to compute the flux at the boundary. We denote the corresponding discrete operator by  $N_h^S$ . It is easily shown that this *superbox scheme* is second-order accurate in the sense that

$$\bar{R}_{2h,h}(N_h^S(R_h q) - \bar{R}_h N(q)) = \mathcal{O}(h^2).$$

Instead of the finite volume superbox scheme, we can also adopt a finite difference approach. Interpolation from the left (right) can be used to obtain a value  $q_{ijk}^l$  ( $q_{ijk}^r$ ) at the left (right) side of all walls  $\Gamma_{ijk}$ . The simplest second-order scheme is the central differencing scheme. Here the interpolation is done irrespective of a particular characteristic direction. Central differencing yields  $f(q_0, q_1) = f((q_0 + q_1)/2)$  for the numerical flux function. (So, it makes no distinction between left and right side.) In contrast with the first-order schemes, the central difference scheme is under-diffusive, which may lead to instabilities. When a central scheme is used alone, an artificial additional diffusion (dissipation) term is added to stabilise the solution [29, 63].

To improve the stability behaviour, it is better to take into account the domain of dependence of the solution (the direction of the characteristics) and to distinguish between interpolated values from the left and from the right at a cell wall. For simplicity of notation we shall exemplify this only for the 1-D case. Generalisation to 2-D is straightforward.

In 1-D, eq.(4.4) reduces to

$$N_h(q_h)_i = f_{i+1/2} - f_{i-1/2}, \quad (5.1)$$

where  $f_{i+\frac{1}{2}} = f(q_{i+\frac{1}{2}}^l, q_{i+\frac{1}{2}}^r)$ .

We define  $\Delta q_{i+1/2} = q_{i+1} - q_i$  and find the second-order upwind interpolated values from the left and from the right respectively

$$\begin{aligned} q'_{i+\frac{1}{2}} &= q_i + \frac{1}{2} \Delta q_{i-\frac{1}{2}} , \\ q'_{i-\frac{1}{2}} &= q_{i+1} - \frac{1}{2} \Delta q_{i+\frac{1}{2}} . \end{aligned} \quad (5.2)$$

Notice that on a non-equidistant grid, with these simple expressions, second-order accuracy is guaranteed only if the grid is sufficiently smooth.

Although other instability problems may arise [39], stability properties of these one-sided approximations are better than for central approximations, but still monotonicity is not preserved. The usual way to force the monotonicity is to introduce a limiting function  $\psi$  [66, 69], and to interpolate by

$$\begin{aligned} q'_{i+\frac{1}{2}} &= q_i + \frac{1}{2} \psi'_{i+1/2} \Delta q_{i-\frac{1}{2}} \\ q'_{i-\frac{1}{2}} &= q_i - \frac{1}{2} \psi'_{i-1/2} \Delta q_{i+\frac{1}{2}} , \end{aligned} \quad (5.3)$$

where  $\psi' = \psi(R)$  and  $\psi' = \psi(1/R)$  are chosen, depending on  $R = \Delta q_{i+1/2} / \Delta q_{i-1/2}$ , such that  $q'_{i-1/2}$  lies between  $q_{i-1}$  and  $q_i$ , and  $q'_{i+1/2}$  between  $q_i$  and  $q_{i+1}$ , (cf. [66, 69]). One possible choice is the Van Albada limiter [1, 66].

$$\psi(R) = \frac{R^2 + R}{R^2 + 1} .$$

Van Leer [48] proposes a linear combination of the one-sided and central interpolation. Parametrised by  $\kappa$  we obtain

$$\begin{aligned} q'_{i+\frac{1}{2}} &= q_i + \frac{1}{4} \left[ (1-\kappa) \Delta q_{i-\frac{1}{2}} + (1+\kappa) \Delta q_{i+\frac{1}{2}} \right] , \\ q'_{i-\frac{1}{2}} &= q_i - \frac{1}{4} \left[ (1-\kappa) \Delta q_{i+\frac{1}{2}} + (1+\kappa) \Delta q_{i-\frac{1}{2}} \right] . \end{aligned} \quad (5.4)$$

This general formula contains for instance: ( $\kappa = -1$ ) the one-sided second-order scheme (5.2); ( $\kappa = 0$ ) Fromm's scheme; ( $\kappa = 1/3$ ) a "third-order" upwind biased scheme; and ( $\kappa = 1$ ) the central difference scheme. (Notice that the "third-order" scheme is third-order consistent in a 1-D situation; in 2-D the scheme is second-order accurate. In 1-D, the superbox scheme,  $N_h^\delta$ , corresponds to the use of  $\kappa = +1$  for odd  $i$ , and  $\kappa = -1$  for even  $i$ .)

The interpolation (5.4) is well defined in the interior cells of the domain. In the cells near the boundary  $\partial\Omega^*$ , one of the values  $\Delta q_{i\pm 1/2}$  is not defined, by the absence of a value  $q_i$  corresponding to a point outside  $\Omega^*$ . Here a different approximation should be used. In our computations we set  $\Delta q_{i+1/2} = \Delta q_{i-1/2}$  at the cell  $\Omega_i$  near the boundary. This corresponds with the "superbox" approximation for these cells. For the superbox scheme and for the scheme (5.4), with different values of  $\kappa$ , results are shown in [39]. Some of them are also shown in Fig.5.1-5.3. The second-order surface pressure distributions in Fig.5.2b are preceded by first-order distributions (Fig.5.2a). (Notice the very fast convergence for the latter.)

Thus, with the MUSCL approach we have constructed a second-order accurate semi-discretisation of (3.4)

$$(q_h)_t + N_h^2(q_h) = 0 . \quad (5.5)$$

## 5.2 THE SOLUTION OF THE SECOND-ORDER DISCRETE SYSTEM

One possibility to find the solution of the steady state equations

$$N_h^2(q_h) = 0 , \quad (5.6)$$

is to take an initial guess and to solve the semi-discretised equation (5.5) for  $t \rightarrow \infty$ , i.e. to compute the time dependent solution  $q_h(t)$  until initial disturbances have sufficiently died out. Just as for the first-order discretised equations, we take the other (fully implicit) approach and try to solve the

system

$$N_h^2(q_h) = r_h, \quad (5.7)$$

directly.

However, if we try to solve the second-order discretisation (5.6) in the same manner as we do the first-order equations, we may expect difficulties because the nonlinear equations (5.6) are less stable. The second-order discretisations are less diffusive, and (as already mentioned) in the case of central differences clearly "anti-diffusive". This may lead not only to non-monotonous solutions, but it can also cause a Gauss Seidel relaxation not to reduce the rapidly varying error components.

A local mode analysis of smoothing properties of GS relaxation for first- and second-order upwind Euler discretisations can be found in [31]. There, the flux splitting upwind scheme of Steger and Warming[68] is analyzed, whereas we apply Osher's scheme. Numerical evidence that convergence for the relaxation process of a second-order upwind procedure is slower than for a first-order scheme, is also found in [49, 52]. Here van Leer's flux splitting scheme[46] was used.

To obtain second-order accurate solutions, we do not try to solve the system  $N_h^2(q_h) = 0$  as such. We use the first-order operator  $N_h^1$  to find the higher-order accurate approximation in a defect correction iteration

$$N_h^1(q_h^{(1)}) = 0, \quad (5.8a)$$

$$N_h^1(q_h^{(i+1)}) = N_h^1(q_h^{(i)}) - N_h^2(q_h^{(i)}). \quad (5.8b)$$

If the problem is smooth enough, the accuracy of  $q_h^{(i)}$  is of order 2 for  $i \geq 2$  (Theorem 1.6.1). If the solution is not smooth (higher-order derivatives are dominating), there is no reason to expect the solution of (5.6) to be more accurate than the solution of (5.8a). Nevertheless, in [20, 22, 39] evidence is given that a few defect correction steps may improve the solution considerably. This is also shown in Fig.5.6.

In fact we may use  $q_h^{(i+1)} - q_h^{(i)}$  as an error indicator. In the smooth parts of the solution  $q_h^{(1)} - q_h^{(1+i)} = \mathcal{O}(h)$ ,  $q_h^{(2)} - q_h^{(2+i)} = \mathcal{O}(h^2)$ ; where these differences are larger, e.g.  $\mathcal{O}(1)$ , the solution is not smooth (relative to the grid used). Then grid adaptation is to be considered rather than the choice of a higher-order method, if a more accurate solution is wanted. Equation (5.8b) describes an iterative process, in which a first-order system has to be solved (iteratively) in each step. In practice the inner iteration is restricted to a single cycle. In Fig.5.2, it is shown that this is an efficient procedure.

In a multigrid environment, where solutions on more grids are available, it is natural also to consider other approaches to compute higher-order solutions, such as (1) Richardson extrapolation; (2)  $\tau$ -extrapolation; or (3) Brandt's double discretisation.

The two extrapolation methods can be well used to find a more accurate solution if the solution is smooth indeed[20]. A drawback is that these methods rely on the existence of an asymptotic expansion of the (truncation) error for  $h \rightarrow 0$ , and -globally- no a-priori information about the validity of such assumption is available. Another disadvantage is that the accurate solution (for Richardson extrapolation) or the estimate for the truncation error ( $\tau$ -extrapolation) is obtained at the one-but-finest level and no high resolution of local phenomena is obtained. Whereas we want not only a high order of accuracy, but also an accurate representation of possible discontinuities, it is advised to use Richardson extrapolation (only) as a cheap means to find a higher-order initial estimate for the iteration process (5.8b).

Since the evaluation of  $N_h^2(q_h)$  is hardly more expensive than the evaluation of  $N_h^1(q_h)$ , the costs to compute the defect in (5.8b) is of the same order as the evaluation of the relative truncation error  $\tau_{2h,h}(q_h) = N_{2h}^1(R_{2h,h}q_h) - \bar{R}_{2h,h}N_h^1(q_h)$ . This makes us to prefer (5.8b) to  $\tau$ -extrapolation.

Having both a first- and a second-order discrete operator at our disposal, Brandt's double discretisation[7] seems another efficient way to find a second-order accurate solution. However, we have bad experience in applying it to the Euler equations. In particular when solving (contact) discontinuities.

Using the Collective SGS relaxation and a second-order scheme based on (5.4), we experienced serious problems in the computation of the numerical fluxes, caused by virtual cavitation of the flow. Our explanation is the following. In Brandt's double discretisation each iteration cycle consists of a smoothing step towards the solution of  $N_h^1(q_h) = r_h^1$ , and a coarse grid correction step towards the solution of  $N_h^2(q_h) = r_h^2$ . At a discontinuity, the differences between the results after the first and the second half-step may be considerable. In our case these differences resulted in such large differences in values for  $q_{ij}^k$  and  $q_{ijk}^k$ , that the numerical flux  $f_{ijk}(q_{ij}^k, q_{ijk}^k)$  could not be properly evaluated. (The solution of the Riemann problem with the two states  $q_{ij}^k$  and  $q_{ijk}^k$  shows cavitation.)

### 5.3 THE COMPLETE MULTIGRID ALGORITHM

We aim at the efficient computation of the approximate solution  $q_h$  of the Euler equations for a given mesh and we assume that also  $L$  coarser meshes exist. We denote the level of refinement by  $i$  and the approximate solution at level  $i$  by  $q_{(i)} = q_{2^{L-i}h}$ . The coarser grids,  $i < L$ , are not only used for the realisation of FAS-iteration steps as described in section 4.5, but also for the construction of the initial estimate for the iteration process. The algorithm used to obtain the initial estimate and further iterands in the defect correction process is as follows:

- (0) start with an approximation for  $q_{(0)}$ ;
- (1) for  $i$  from 0 to  $L-1$  do
  - (1a) for  $j$  from 1 to  $k_i$  do FASCYCLE ( $N_{(i)}^1 q_{(i)} = 0$ ) enddo;
  - (1b)  $q_{(i+1)} := P_{i+1,i}^2 q_{(i)}$ ;
- (1) enddo;
- (2) for  $j$  from 1 to  $k_L$  do FASCYCLE ( $N_{(L)}^1 q_{(L)} = 0$ ) enddo;
- (3)  $q_{(L)} := q_{(L)} + P_{L,L-1}^S (R_{L-1,L}^1 q_{(L)} - q_{(L-1)})$ ;
- (4) for  $d$  from 1 to  $dcps$  do
  - (4a)  $r_{(L)} := N_{(L)}^1(q_{(L)}) - N_{(L)}^2(q_{(L)})$ ;
  - (4b) for  $j$  from 1 to  $k_d$  do FASCYCLE ( $N_{(L)}^1 q_{(L)} = r_{(L)}$ ) enddo;
- (4) enddo;

Stage (1) is an FMG process to obtain a first-order accurate initial estimate at level  $L$ . The prolongation  $P_{i+1,i}^2$  is a bilinear interpolation procedure and, hence, accurate enough to retain the first-order accuracy on the finer mesh. Asymptotically, the discretisation error for  $q_{(i)}$  is bounded by  $Ch_{(i)} = \mathcal{O}(2^{L-i}h)$  for  $h_{(L)} = h \rightarrow 0$ . Now theorem 2.2 shows that, for a fixed  $k_i = k$  at all levels, the iteration error at level  $i$  is  $\approx Ch_{(i)}\mu^k / (1 - 2\mu^k)$ , where  $\mu$  is an upper bound for the FAS-convergence factor. Therefore, to obtain a first-order accurate solution, for iteration (1a) it is not necessary to reduce the iteration error in  $q_{(i)}$  by a factor much smaller than  $\mu^k \approx 1/3$ . This means that a single FAS step as described in section 4 may be sufficient (i.e.  $k = 1$ ). Not being sure about the validity of the asymptotic assumption, we set  $k_i = 2$ ,  $i = 1, 2, \dots, L$ . Stage (2) is the FAS-iteration to obtain the solution to  $N_h^1(q_h) = 0$  up to truncation error accuracy.

Stage (3) is a Richardson extrapolation step to find a second-order initial estimate for  $q_h$ . The prolongation  $P_{L,L-1}^S$  and the restriction  $R_{L-1,L}^1$  are piecewise bilinear interpolation over superboxes and averaging over cells, respectively, so that  $R_{L-1,L}^1 P_{L,L-1}^S = I_{L-1}$  is the identity, and  $P_{L,L-1}^S R_{L-1,L}^1$  is a projection operator. With the asymptotic expansion for the error in  $q_h$  as

$$q_h = R_h \hat{q} + h^p R_h e + \mathcal{O}(h^{p+1}), \quad (5.9)$$

where  $\hat{q}$  is the exact solution, we obtain for  $p=1$  the second-order extrapolation

$$R_{2h} \hat{q} = 2 R_{2h,h} q_h - q_{2h} + \mathcal{O}(h^2). \quad (5.10)$$

We find the extrapolated value of  $q_h$  in (3) as the sum of (5.10) and  $(I_i - P_{i,i-1}^S R_{i-1,i}^0) q_h \in \text{Ker}(R_{2h})$ . We notice that formally the approximation of  $q_{(L)}$  after stage (3) is still  $\mathcal{O}(h)$ , unless  $q_{(L-1)}$  is an  $\mathcal{O}(h^2)$  approximation, and stage (2) can reduce the (smooth) error

component  $R_h e$  by a factor  $\mathcal{O}(h)$ . Nevertheless, we see in practice that already for small values of  $k_i, i = 1, 2, \dots, L$ , the Richardson extrapolation can reduce the error significantly[20].

Stage (4) is the defect correction iteration (5.8b). If the defect correction iteration starts with a first-order initial approximation, for second-order accuracy it is sufficient to take  $\text{dcps} = 1$ . This necessitates an improvement of the error by a factor  $\mathcal{O}(h)$  in the iteration (4b), i.e. we need  $\text{kd} = \mathcal{O}(\log(h))$ . However, since the FAS iteration is the expensive part of the computation in (4), for most purposes we take  $\text{kd} = 1$  and a sufficiently large number for  $\text{dcps}$ . Results for the algorithm can be found in [20, 22, 24, 39, 42, 43].

## 6. SOLUTION OF THE NAVIER STOKES EQUATIONS

### 6.1 INTRODUCTION

Mainly based on the Euler method described in the sections 3 to 5, a Navier-Stokes method has been developed recently [23, 40, 41, 62]. Our first objective was the efficient and accurate computation of laminar, steady, 2-D, compressible flows at practically relevant (i.e. high) Reynolds numbers, but (still) at subsonic or low-supersonic Mach numbers. The non-isenthalpic Euler code developed earlier appeared to be a good starting point for this purpose.

### 6.2 THE DISCRETISATION METHOD

The discretisation is based on a strict splitting of the Navier-Stokes fluxes in a convective and a diffusive part, according to (3.1b). This splitting is retained throughout the discretisation, both for the discrete approximation of the internal fluxes, and for the boundary fluxes (boundary conditions). To keep the possibility of Euler flow discontinuities to be captured, the equations are again discretised in their integral form (3.3). In fact, as  $1/\text{Re} = 0$ , the Navier Stokes discretisation reduces to exactly the Euler discretisations described in the sections 4 and 5.

A straightforward and simple discretisation of the integral form is obtained by subdividing the integration region  $\Omega$  into quadrilateral finite volumes  $\Omega_{i,j}$ , and by requiring that the conservation laws hold for each finite volume separately:

$$\int_{\partial\Omega_{i,j}} (f(q)n_x + g(q)n_y)ds - \frac{1}{\text{Re}} \int_{\partial\Omega_{i,j}} (r(q)n_x + s(q)n_y)ds = 0, \quad \forall i,j. \quad (6.1)$$

For the evaluation of the convective flux vectors we make use again of the rotational invariance of the flow equations. We do not do so for the diffusive flux vectors. Given our simple central discretisation of diffusive terms, use of rotational invariance for the latter is hardly advantageous. Thus, the discretised equations become

$$\int_{\partial\Omega_{i,j}} T^{-1}(n_x, n_y) f(T(n_x, n_y)q)ds - \frac{1}{\text{Re}} \int_{\partial\Omega_{i,j}} (r(q)n_x + s(q)n_y)ds = 0, \quad \forall i,j, \quad (6.2)$$

with  $T(n_x, n_y)$  the rotation matrix in (4.5).

#### 6.2.1 Evaluation of convective fluxes.

The computation of the first- and second-order discrete approximation of the convective fluxes is made in the same way as for the Euler equations. Considering for instance the numerical flux function  $(f(q))_{i+\frac{1}{2},j} = f(q_{i+\frac{1}{2},j}^l, q_{i+\frac{1}{2},j}^r)$ , where the superscripts  $l$  and  $r$  refer to the left and right side of volume wall  $\partial\Omega_{i+\frac{1}{2},j}$ , first-order accurate convection is obtained again by taking  $q_{i+\frac{1}{2},j}^l = q_{i,j}$  and  $q_{i+\frac{1}{2},j}^r = q_{i+1,j}$ . Higher-order accurate convection is obtained again with the  $\kappa$ -schemes as introduced in (5.4), with  $\kappa \in \mathbb{R}$  ranging from  $\kappa = -1$  (fully one-sided upwind) to  $\kappa = 1$  (central). The value  $\kappa = 1/3$  has appeared to be optimal. To avoid spurious non-monotonicity, a new limiter has been constructed by Koren [23] for the  $\kappa = 1/3$  approximation:

$$\psi(R) = \frac{R + 2R^2}{2 - R + 2R^2}. \quad (6.3)$$

### 6.2.2 Evaluation of diffusive fluxes.

For the evaluation of the diffusive fluxes at a volume wall, it is necessary to compute  $\text{grad}(u)$ ,  $\text{grad}(v)$  and  $\text{grad}(c^2)$  at that wall. For this we use a standard technique [59]. To compute for instance  $(\text{grad}(u))_{i+\frac{1}{2},j}$ , we use Gauss' theorem:

$$(\nabla u)_{i+\frac{1}{2},j} = \frac{1}{A_{i+\frac{1}{2},j}} \int_{\partial\Omega_{i+\frac{1}{2},j}} u \mathbf{n} ds, \quad (6.4)$$

with  $\mathbf{n} = (n_x, n_y)^T$ , and  $\partial\Omega_{i+\frac{1}{2},j}$  the boundary and  $A_{i+\frac{1}{2},j}$  the area of a quadrilateral dummy volume  $\Omega_{i+\frac{1}{2},j}$  of which the vertices  $z = (x, y)$  are defined by:

$$z_{i,j \pm \frac{1}{2}} = \frac{1}{2}(z_{i-\frac{1}{2},j \pm \frac{1}{2}} + z_{i+\frac{1}{2},j \pm \frac{1}{2}}). \quad (6.5)$$

A similar expression exists for  $z_{i \pm \frac{1}{2},j}$ .

The line integrals  $\int_{\partial\Omega_{i+\frac{1}{2},j}} u n_x ds$  and  $\int_{\partial\Omega_{i+\frac{1}{2},j}} u n_y ds$  are approximated by

$$\begin{aligned} \int_{\partial\Omega_{i+\frac{1}{2},j}} u n_x ds = & u_{i+1,j} (y_{i+1,j+\frac{1}{2}} - y_{i+1,j-\frac{1}{2}}) \\ & + u_{i+\frac{1}{2},j+\frac{1}{2}} (y_{i,j+\frac{1}{2}} - y_{i+1,j+\frac{1}{2}}) \\ & + u_{i,j} (y_{i,j-\frac{1}{2}} - y_{i+1,j-\frac{1}{2}}) \\ & + u_{i+\frac{1}{2},j-\frac{1}{2}} (y_{i+1,j-\frac{1}{2}} - y_{i,j-\frac{1}{2}}), \end{aligned} \quad (6.6a)$$

and

$$\begin{aligned} \int_{\partial\Omega_{i+\frac{1}{2},j}} u n_y ds = & u_{i+1,j} (x_{i+1,j-\frac{1}{2}} - x_{i+1,j+\frac{1}{2}}) \\ & + u_{i+\frac{1}{2},j+\frac{1}{2}} (x_{i+1,j+\frac{1}{2}} - x_{i,j+\frac{1}{2}}) \\ & + u_{i,j} (x_{i,j+\frac{1}{2}} - x_{i,j-\frac{1}{2}}) \\ & + u_{i+\frac{1}{2},j-\frac{1}{2}} (x_{i,j-\frac{1}{2}} - x_{i+1,j-\frac{1}{2}}), \end{aligned} \quad (6.6b)$$

with for  $u_{i+\frac{1}{2},j \pm \frac{1}{2}}$  the central expression

$$u_{i+\frac{1}{2},j \pm \frac{1}{2}} = \frac{1}{4}(u_{i,j} + u_{i,j \pm 1} + u_{i+1,j} + u_{i+1,j \pm 1}). \quad (6.7)$$

Similar expressions are used for the other gradients, and other walls. For sufficiently smooth grids this central diffusive flux computation is second-order accurate. For details about the discretisation and the treatment of the diffusive flux at the boundary, we refer to [40, 41, 62].

## 6.3 SOLUTION METHOD

To efficiently solve the system of discretised Navier-Stokes equations, again symmetric point Gauss-Seidel relaxation, accelerated by nonlinear multigrid (FAS), is applied. With a scalar convection diffusion equation as model, local mode analysis shows that 'symmetric point Gauss-Seidel with multigrid' converges fast for the first-order discretised equation, for any value of the mesh Reynolds number  $h/\epsilon$  [40]. However, it appears to converge very slowly for the higher-order ( $\kappa=1/3$ ) discretised equation, for small and moderately large values of  $h/\epsilon$ . It even appears to diverge for large values of  $h/\epsilon$  [41]. Clearly the origin of this is the higher-order discretisation of the convection operator. As with the Euler equations, the difficulty in inverting the higher-order operator is bypassed by introducing defect correction as an outer iteration for the nonlinear multigrid cycling. Let  $F_h(q_h)$  denote the full, second-order accurate discrete operator, and  $\tilde{F}_h(q_h)$  the less accurate operator

that can be easily inverted. Then iterative defect correction can be written as

$$\begin{aligned}\tilde{F}_h(q_h^1) &= 0, \\ \tilde{F}_h(q_h^{n+1}) &= \tilde{F}_h(q_h^n) - \omega F_h(q_h^n), \quad n = 1, 2, \dots, N,\end{aligned}\tag{6.8}$$

where  $n$  denotes the  $n$ th iterand, and  $\omega$  a damping factor. The standard value for  $\omega$  is:  $\omega = 1$ . Special attention has been paid to the choice of the approximate operator  $\tilde{F}_h(q_h)$  for the Navier-Stokes equations. The operator necessarily has *first-order accurate* convection, but the amount of diffusion can be chosen freely. This freedom has been exploited. Three approximate operators have been considered: (i) an operator without diffusion, (ii) an operator with partial diffusion, and (iii) an operator with full, second-order accurate diffusion. The *first approximate operator*, which neglects diffusion, was already known from the Euler work. Given its successful application there, it may be expected to be suitable for very large values of the mesh Reynolds number. The *second approximate operator* neglects the cross derivatives in the diffusive terms, but it has full second-order diffusion stemming from the remaining derivatives. The special feature of this operator is that the same five-point data structure can be used, for the evaluation of the convective and diffusive fluxes in the Navier-Stokes equations. The operator combines elegance and simplicity with a rather good resemblance to the higher-order operator. The *third approximate operator* resembles the higher-order operator most closely, and therefore has the best convergence properties. In the case of this third approximate operator, for sufficiently smooth problems and a second-order accurate  $F_h$ , theorem 1.6.1 predicts the solution to be second-order accurate after a single Defect Correction cycle. Because the discrete approximations of the diffusive flux is only zeroth order for the cases (i) and (ii), theory does not give such guarantee for these approximate operators. Local mode analysis applied to a model equation, and experiments with the Navier-Stokes equations showed the third approximate operator to have the best convergence properties indeed. Its relative complexity has been taken for granted. The results presented in the next section have all been obtained with this operator.

Though the mesh Reynolds numbers in the computations performed were large, we obeyed the multigrid requirement  $m_r + m_p > 2$ , [7, 16], where  $m_r$  and  $m_p$  denote the order of accuracy of the defect restriction and the correction prolongation respectively, and where 2 is the order of the differential equations. This was achieved by using a piecewise constant restriction ( $m_r = 1$ ) and a piecewise bilinear prolongation ( $m_p = 2$ ). For further details about the multigrid method applied we refer to [40].

#### 6.4 NUMERICAL RESULTS

To evaluate the method described, we considered as reference test case the experiment from [17], performed at  $Re = 2.96 \cdot 10^5$ . First we tried to make a satisfactory grid. Since the present code has the possibility to compute Euler flows, it is easy to optimise the grid for convection only. For the present test case this led via the rectangular  $80 \times 32$ -grid shown in Fig.6.1a to the oblique  $80 \times 32$ -grid in the same figure. The latter grid has been fitted to the incoming shock.

The corresponding *inviscid* surface pressure distributions as obtained by Osher's scheme, and with the first-order, the non-limited  $\kappa = 1/3$  and the limited  $\kappa = 1/3$  approximation are given in Fig.6.1b. The poor solution quality on the rectangular grid is clear.

Together with the measured data, the computed *viscous* surface pressure distributions are given in Fig.6.1c. First we consider the results obtained on the rectangular grid. Given the bad inviscid solutions, obtained on the regular grid, it should be noticed that the good resemblance of the experimental and the second-order accurate viscous surface pressure distribution is absolutely fake. Since for this standard test case most authors use rectangular grids, and since most codes smear out discontinuities which are not aligned with the grid, a lot of good resemblance ever found for this test case might in fact be deceptive. Considering the results obtained on the oblique grid and comparing at first the computed surface pressure distributions, we see that diffusion has done its job in qualitatively different ways. In downstream direction, the second-order pressure distribution in the interaction region shows successively: a compression, a plateau and another compression. The computed second-order accurate surface pressure distribution is characteristic for a shock wave - boundary layer



interaction with separation bubble, i.e. with separation and re-attachment, whereas the first-order distribution typically is the distribution belonging to a non-separating flow. Given the occurrence of a separation bubble in the experimental results indeed, the first-order solution (on this  $80 \times 32$ -grid) has to be rejected. Comparing the second-order and measured surface pressure distribution, it appears that the latter is more strongly diffused. An explanation for this quantitative difference is lacking. Due to all kinds of uncertainties a detailed quantitative comparison is probably impossible.

In Fig.6.2 some measured and computed velocity profiles are given. Once more, the figures clearly show the good quality of the second-order results. Remarkable for both the first- and second-order velocity profiles is the good agreement with the experimental data in the upper part of the boundary layer at  $x = 1.22$ . Both solutions seem to give a correct prediction of the growth of the boundary layer thickness through the interaction region. For a detailed account of convergence rates and computing times we refer to [41].

#### ACKNOWLEDGEMENT

These notes were compiled from previous notes and reports, some of which were written in cooperation with other people. In particular we want to acknowledge Stefan Spekreijse and Jurgen Rusch for their cooperation in CFD research.

#### REFERENCES

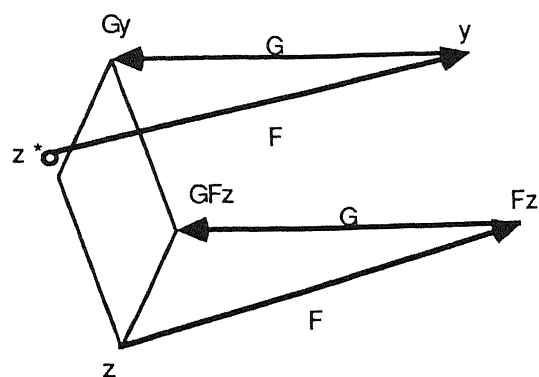
1. G.D. VAN ALBADA, B. VAN LEER, and W.W. ROBERTS (1982). A comparative study of computational methods in cosmic gasdynamics, *Astron. Astrophys.*, 108, 76-84.
2. W.K. ANDERSON, J.L. THOMAS, and B.VAN LEER (1985). *A comparison of finite volume flux vector splittings for the Euler equations*, AIAA-85-0122.
3. K.E. ATKINSON (1973). Iterative variants of the Nystrom method for the numerical solution of integral equations, *Numer. Math.*, 22, 17-33.
4. J.W. BOERSTOEL (1982). A multigrid algorithm for steady transonic potential flows around aerofoils using Newton iteration, *J. Comp. Phys.*, 48, 314-343.
5. H. BRAKHAGE (1960). Ueber die numerische Behandlung von Integralgleichungen nach der Quadraturformelmethode, *Numer. Math.*, 2, 183-196.
6. A. BRANDT (1972). Multi-level adaptive technique (MLAT) for fast numerical solutions to boundary value problems, in *Procs 3rd Int. Conf. Numerical Methods in Fluid Mechanics*, 82-89, Springer Verlag, Berlin.
7. A. BRANDT (1982). Guide to multigrid development, in *Multigrid Methods*, 220-312, ed. W. Hackbusch and U. Trottenberg, Springer Verlag.
8. A. BRANDT (1982.). Multigrid solutions to steady state compressible Navier Stokes equations I, in *Computing Methods in Applied Science and Engineering*, 407-422, ed. R. Glowinski and J.L. Lions, North Holland Publ. Comp..
9. A. BRANDT and N. DINAR (1979). Multigrid solutions to elliptic flow problems, in *Numerical Methods for Partial Differential Equations*, ed. S.V. Parter, Academic Press, New York, etc..
10. D.A. CAUGHEY (1983). *Multigrid calculation of three dimensional transonic potential flows*, AIAA Paper 83-0374.
11. R.V. CHIMA and G.M. JOHNSON. *Efficient solution of the Euler and Navier Stokes equations with a vectorized Multi Grid algorithm*, AIAA-83-1893, (Danvers 1983).
12. P. COLELLA and P.R. WOODWARD (1984). The Piecewise Parabolic Method (PPM) for Gas Dynamical Simulations, *J. Comp. Phys*, 52, 174-201.
13. H. DECONINCK (1987). *A survey of upwind principles for the multidimensional Euler equations*, Preprint 1987-10/CFD, Von Karman Institute for Fluid Dynamics, Rhode St Genese.
14. E. DICK (1984). A multigrid method for the Cauchy-Riemann and steady state Euler equations based on flux difference splitting, in *Efficient Solution of Elliptic Systems*, ed. W. Hackbusch, Vieweg Verlag.

15. J.A. ESSERS (1982). Explicit and implicit corrected viscosity schemes for the computation of steady transonic flows, in *Proceedings of the 4th GAMM-Conference on Numerical Methods in Fluid Mechanics*, ed. H. Viviand, Vieweg Verlag.
16. W. HACKBUSCH (1985). *Multigrid Methods and Applications*, 4, Springer Verlag, Berlin, Heidelberg.
17. R.J. HAKKINEN, I. GREBER, L. TRILLING, and S.S. ABARBANEL (1958). *The Interaction of an Oblique Shock Wave with a Laminar Boundary Layer*, NASA-memorandum 2-18-59 W.
18. A. HARTEN, P.D. LAX, and B.VAN LEER (1983). On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, *SIAM Review*, 25, 35-61.
19. P.W. HEMKER (1984). Multigrid methods for problems with a small parameter in the highest derivative, in *Numerical Analysis, Procs. Dundee Conference 1983*, 106-121, ed. D.F. Griffiths, Springer Verlag.
20. P.W. HEMKER (1986). Defect correction and higher order schemes for the multigrid solution of the steady Euler equations, in *Multi-Grid Methods II*, 150-165, ed. W. Hackbusch and U. Trottenberg, Springer Verlag, Proceedings of the 2nd European Conference on Multigrid Methods, held at Cologne, October 1-4, 1985.
21. P.W. HEMKER, R. KETTLER, P. WESSELING, and P.M. DE ZEEUW (1983). Multigrid Methods: Development of fast solvers, *Appl. Math. & Computat.*, 13, 311-326.
22. P.W. HEMKER and B. KOREN (1986). *A nonlinear multi-grid method for the steady Euler equations*, To appear in the procs of GAMM workshop on the Numerical solution of the Euler equations (INRIA, Rocquencourt, France, June 1986).
23. P.W. HEMKER and B. KOREN (1988). *Multigrid, defect correction and upwind schemes for the steady compressible Navier-Stokes equations*, CWI Report NM-R88xx.
24. P.W. HEMKER, B. KOREN, and S.P. SPEKREIJSE (1986). *A nonlinear multi-grid method for the efficient solution of the steady Euler equations*, 308-313, Springer Verlag, Tenth International Conference on Numerical Methods in Fluid Dynamics. Proceedings, Beijing, 1986..
25. P.W. HEMKER and S.P. SPEKREIJSE (1985). Multigrid solution of the steady Euler equations, in *Advances in Multi-Grid Methods*, 33-44, ed. D. Braess, W. Hackbusch and U. Trottenberg, Vieweg Publ. Comp., Proceedings Multigrid Conference, Oberwolfach, dec. 1984..
26. P.W. HEMKER and S.P. SPEKREIJSE (1986). Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations, *Appl. Num. Math.*, 2, 475- 493.
27. A. JAMESON. Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple grid method, in *Procs AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, 1979*, 122-146.
28. A. JAMESON (1983). Solution of the Euler equations for two dimensional transonic flow by a multigrid method, *Appl. Math. and Computat.*, 13, 327-355.
29. A. JAMESON (1983). Numerical Solution of the Euler Equations for Compressible Inviscid Fluids, in *Procs 6th International Conference on Computational Methods in Applied Science and Engineering*, Versailles, France.
30. A. JAMESON, W. SCHMIDT, and E. TURKEL (1981). *Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes*, AIAA-81-1259.
31. D.C. JESPERSEN (1983). Design and implementation of a multigrid code for the Euler equations, *Appl. Math. and Computat.*, 13, 357-374.
32. D.C. JESPERSEN (1984). *Recent developments in multigrid methods for the steady Euler equations*, von Karman Inst., Rhode-St.Genese, Belgium.
33. D.C. JESPERSEN (1985). *A time-accurate multiple-grid algorithm.*, AIAA-85-1493-CP..
34. G.M. JOHNSON (1982). *Multiple grid acceleration of Lax Wendrof algorithms*, NASA Technical Memorandum 82843.
35. G.M. JOHNSON (1983). Multiple grid convergence acceleration of viscous and inviscid flow computations, *Appl. Math. and Computat.*, 13, 375-398.
36. G.M. JOHNSON (1983). Accelerated solution of the steady state Euler equations., in *Recent*

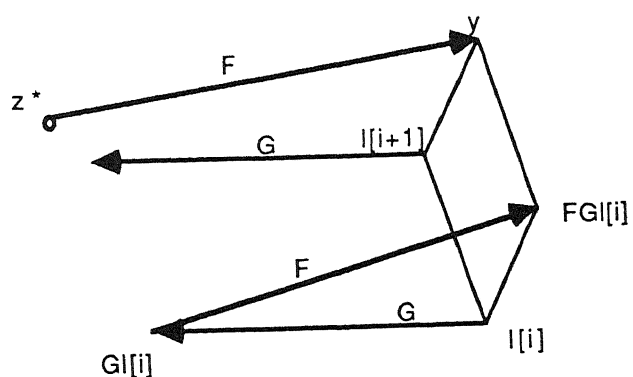
*Advances in Numerical methods in Fluids. Vol.4.*, ed. W.G. Habashi, Pineridge Press.

37. G.M. JOHNSON, J.M. SWISSHELM, and S.P. KUMAR. *Concurrent processing adaptations of a multiple-grid algorithm*, AIAA Paper 85-1508-CP.
38. R. KETTLER (1982). Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods, in *Multigrid Methods*, 502 - 534, ed. W. Hackbusch and U. Trottenberg, Springer Verlag.
39. B. KOREN (1986). *Evaluation of second order schemes and defect correction for the multigrid computation of airfoil flows with the steady Euler equations*, CWI Report NM-R8616, Accepted for publication in *J.Comp.Phys.*.
40. B. KOREN (1988). *First-order Upwind Schemes and Multigrid for the Steady Navier-Stokes equations*, CWI Report NM-R88xx.
41. B. KOREN (1988). *Higher-order Upwind Schemes and Defect Correction for the Steady Navier-Stokes equations*, CWI Report NM-R88xx.
42. B. KOREN and S.P. SPEKREIJSE (1987). *Solution of the steady Euler equations by a multigrid method*, To appear in the Proc. 3rd Copper Mountain Conference on Multigrid Methods
43. B. KOREN and S.P. SPEKREIJSE (1987). Multigrid and defect correction for the efficient solution of the steady Euler equations, in *Research in Numerical Fluid Dynamics*, 87-100, ed. P. Wesseling, Vieweg Verlag, proceedings of the 25th Meeting of the Dutch Association for Numerical Fluid Dynamics.
44. P.D. LAX (1971.). Shock waves and entropy, in *Contributions to Nonlinear Functional Analysis*, ed. E.H. Zarantonello, Acad. Press, New York.
45. P.D. LAX (1973). *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*, 11, SIAM Publication, Philadelphia.
46. B.VAN LEER. Flux-vector splitting for the Euler equations, in *Procs. 8th Intern. Conf. on numerical methods in fluid dynamics, Aachen, June, 1982*, Springer Verlag.
47. B.VAN LEER (1984). On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe, *SIAM J.N.A.*, 5, 1.
48. B.VAN LEER (1984). Upwind difference methods for aerodynamic problems governed by the Euler equations, in *Large-Scale Computations in Fluid Mechanics*, 327-336 (vol2), ed. B.E. Engquist, S. Osher and R.C.J. Somerville, American Mathematical Society, Providence, Rhode Island.
49. B.VAN LEER and W.A. MULDER (1985). Relaxation methods for hyperbolic conservation laws, in *Numerical Methods for the Euler Equations of Fluid Dynamics*, 312 - 333, ed. F. Angrand, A. Der-vieux, J.A. Desideri and R. Glowinski, SIAM, Philadelphia.
50. D.R. MCCARTHY and T.A. REYHNER (1982). Multigrid code for three dimensional transonic potential flow about inlets, *AIAA Journal*, 20, 45-50.
51. W.A. MULDER (1985). Multigrid relaxation for the Euler equations, *J. Comput. Phys.*, 60, 235 - 252.
52. W.A. MULDER and B.VAN LEER (1985). Experiments with implicit upwind methods for the Euler equations, *J. Comput. Phys.*, 59, 232 - 246.
53. RON-HO NI (1982). A multiple grid scheme for solving the Euler equations, *AIAA Journal*, 20, 1565-1571.
54. Z. NOWAK and P. WESSELING (1985). Multigrid acceleration of an iterative method with applications to transonic potential flow, in *Computing Methods in Applied Sciences and Engineering*, 199-217, ed. R. Glowinski and J.L. Lions, North-Holland Publ. Comp..
55. S. OSHER (1981). Numerical solution of singular perturbation problems and hyperbolic systems of conservation laws, in *Analytical and Numerical Approaches to Asymptotic problems in Analysis*, ed. O. Axelsson, L.S. Frank and A. van der Sluis, North Holland Publ. Comp..
56. S. OSHER and S. CHAKRAVARTHY (1983). Upwind schemes and boundary conditions with applications to Euler equations in general geometries, *J. Comp. Phys.*, 50, 447-481.
57. S. OSHER and S. CHAKRAVARTHY (1984). High resolution schemes and the entropy condition, *SIAM J. Numer. Anal.*, 21, 955-984.

58. S. OSHER and F. SOLOMON (1982). Upwind difference schemes for hyperbolic systems of conservation laws, *Math. Comp.*, 38, 339-374.
59. R. PEYRET and T.D. TAYLOR (1983). *Computational Methods for Fluid Flow*, Springer verlag.
60. P.L. ROE (1981). The use of the Riemann problem in finite difference schemes, in *Procs. 7th Int. Conf. Num. Meth. Fl. Dyn.*, 354-359, ed. Reynolds & McCormack, Springer Verlag.
61. P.L. ROE (1981). Approximate Riemann solvers, parameter vectors and difference schemes, *J. Comp. Phys.*, 43, 357-372.
62. J.J. RUSCH (1986). *The use of defect correction for the solution of the 2-D compressible Navier-Stokes equations with large Reynolds number*, CWI Report NM-R8623.
63. W. SCHMIDT and A. JAMESON. Euler solvers as an analysis tool for aircraft aerodynamics, in *Advances in Computational Transonics*, ed. W.G.Habashi, Pineridge Press, Swansea.
64. J. SMOLLER (1983). *Shock waves and reaction diffusion equations*, 258, Springer Verlag.
65. J.C. SOUTH and A. BRANDT (1977). Application of a multi-level grid method to transonic flow calculations, in *Proc. of Workshop on Transonic Flow problems in Turbomachinery, Monterey, 1976*, 180-206, ed. T.C. Adamson and M.F. Platzer, Hemisphere.
66. S.P. SPEKREIJSE (1987). *Multigrid Solution of the Steady Euler Equations*, Doctoral thesis, Delft University of Technology.
67. J.L. STEGER (1981). *A preliminary study of relaxation methods for the inviscid conservative gas-dynamics equations using flux splitting*, Nasa Contractor Report 3415.
68. J.L. STEGER and R.F. WARMING (1981). Flux vector splitting of the inviscid gasdynamics equations with application to finite difference methods, *J. Comp. Phys.*, 40, 263-293.
69. P.K. SWEBY (1984). High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J.Numer.Anal.*, 21, 995-1011.
70. J.M. SWISSELM and G.M. JOHNSON (1985). Numerical simulation of three-dimensional flowfields using the Cyber 205, in *Supercomputer Applications*, ed. R.W. Numrich, Plenum Publ. Comp..
71. D. YOUNG (1971). *Iterative Solution of Large Linear Systems*, Academic Press, New York, London.



a. DCPA.



b. DCPB.

Figure 1.1. Defect correction approaches.

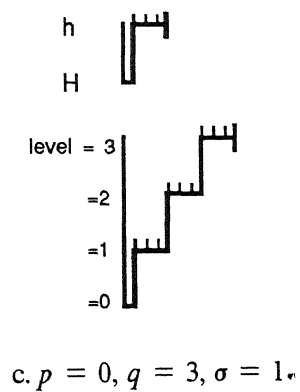
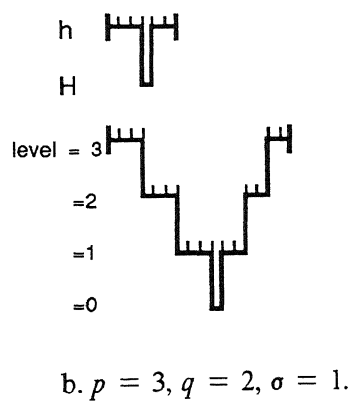
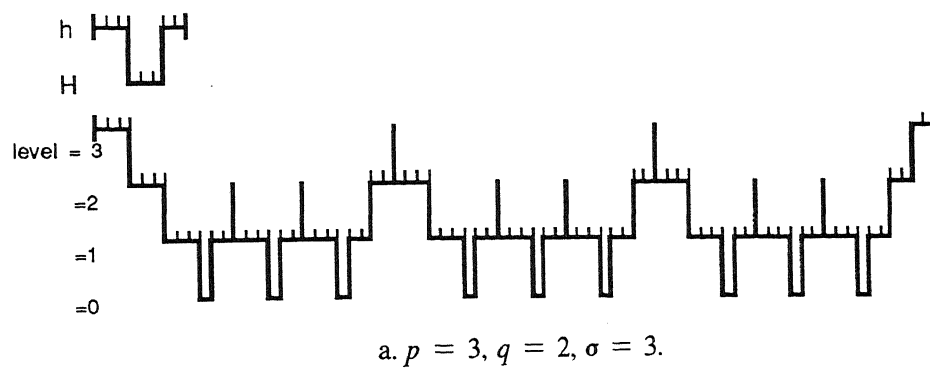
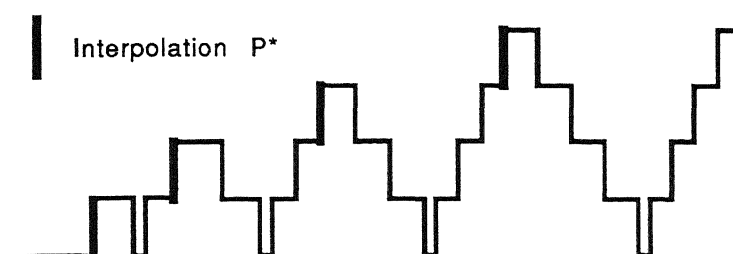
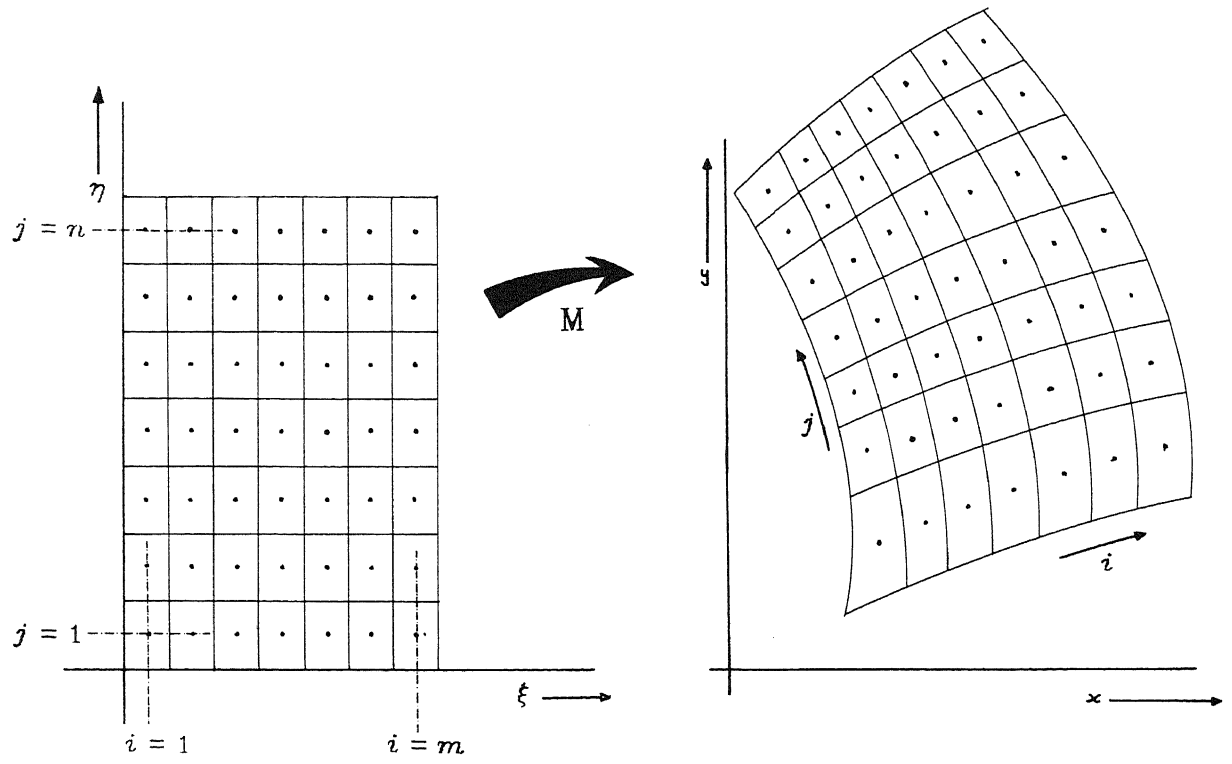
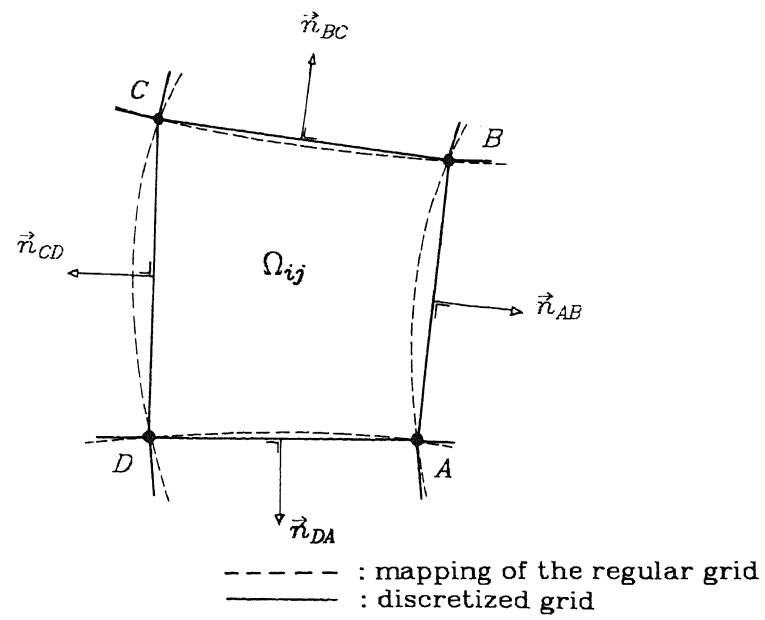


Figure 2.2. Multigrid strategies.





a. Example of a mapping.



b. Finite volume in an irregular grid.

Figure 3.1. Finite volume grid.

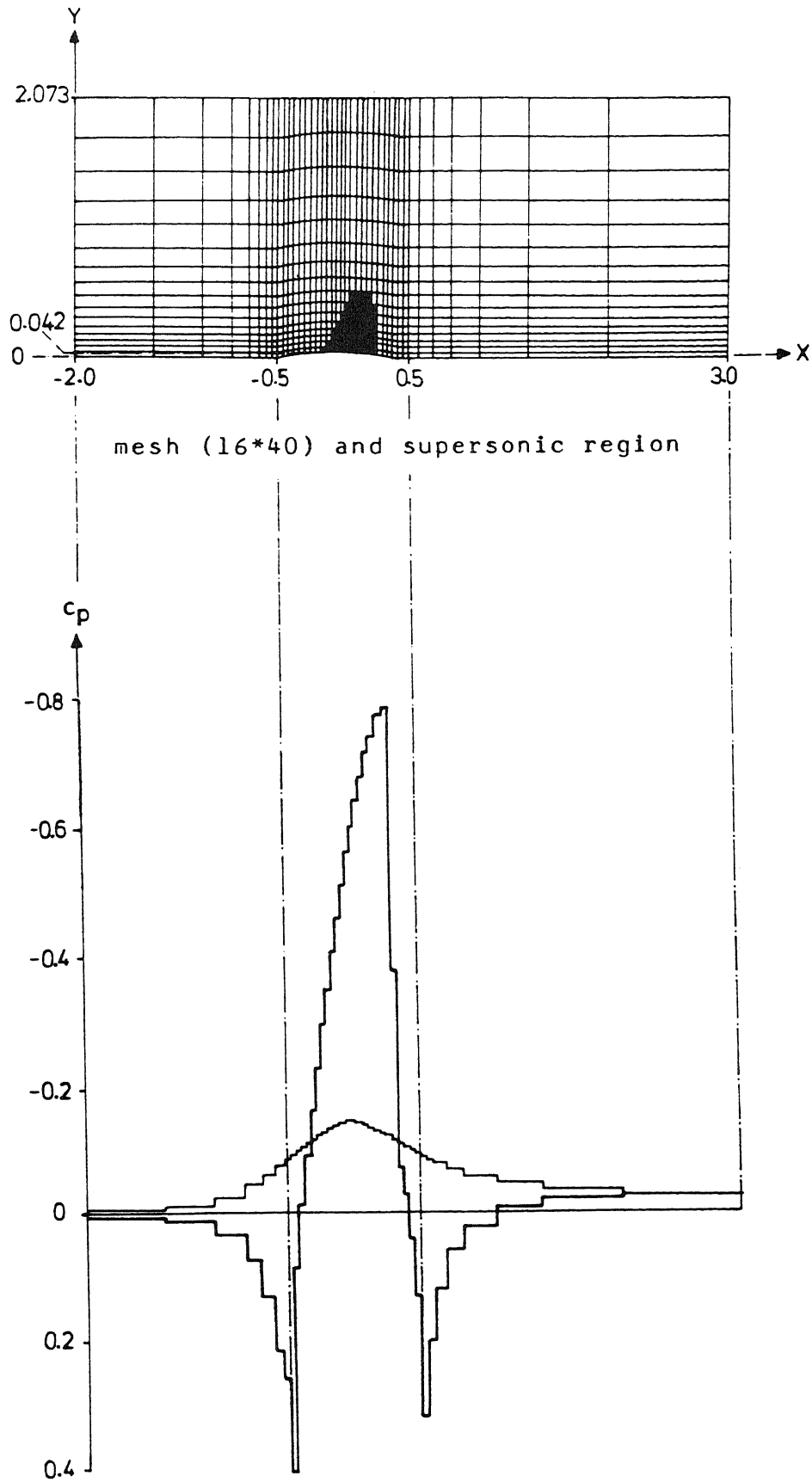


Figure 4.2. Grid at level 4, supersonic region and pressure distribution along lower and upper surface.

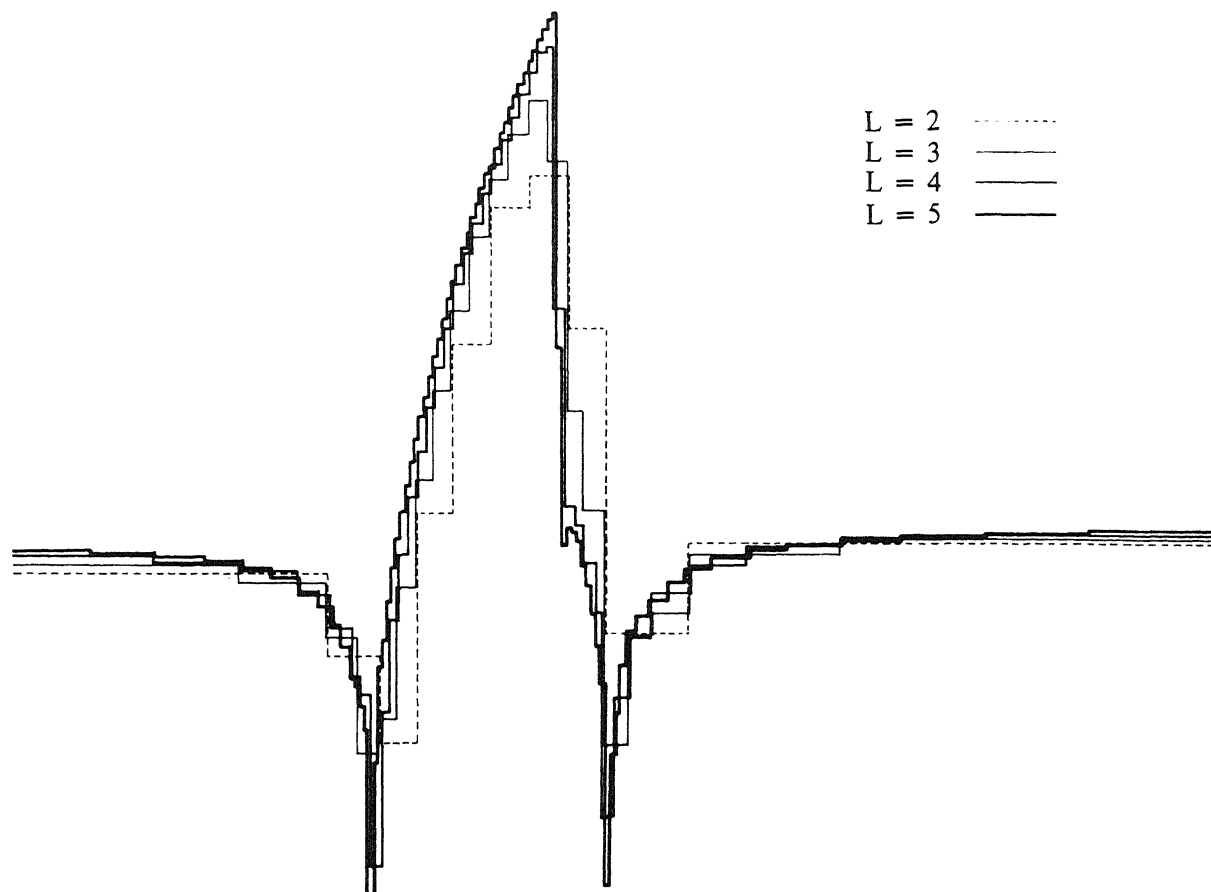


Figure 4.3. Pressure distribution ( $c_p$ ) along lower surface, for  $L=2, 3, 4, 5$ .

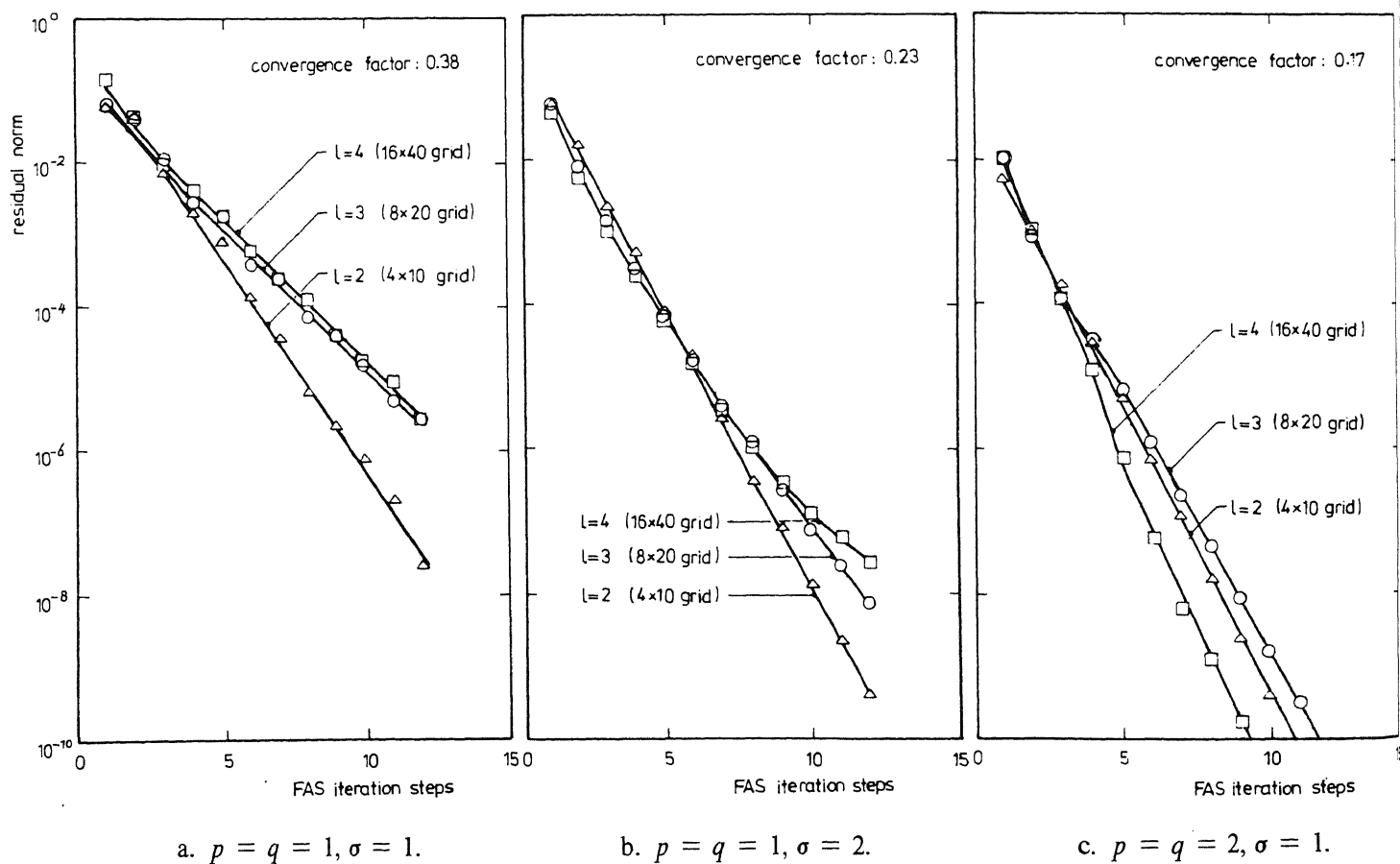
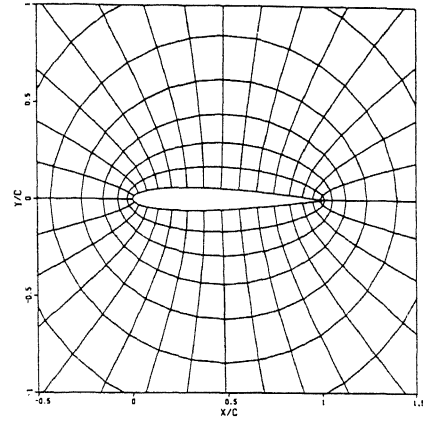
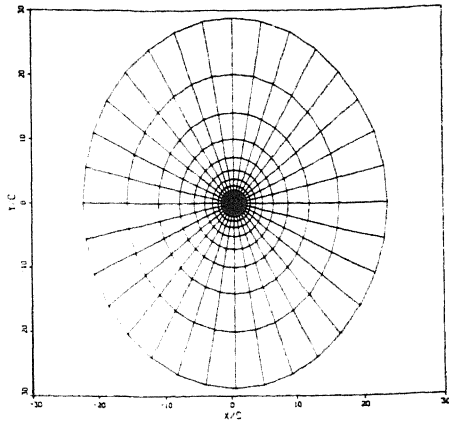
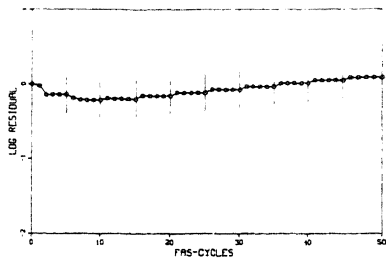
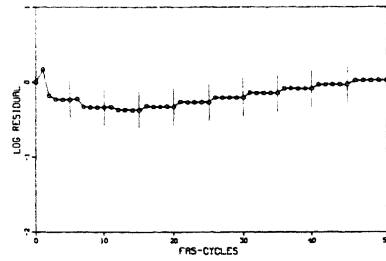
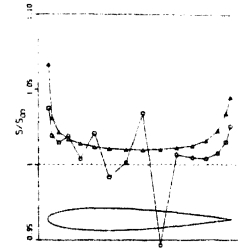


Figure 4.4. Convergence histories transonic channel flow ( $M_{inlet} = 0.85$ ).

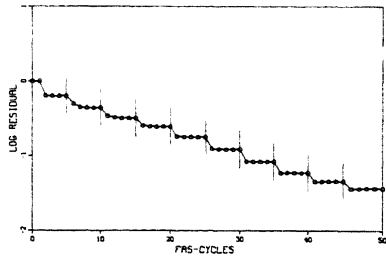
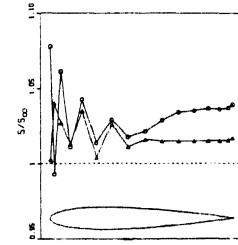


a.  $32 \times 16$ -grid.

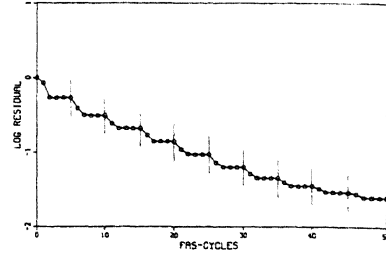
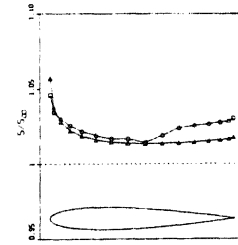
Central.



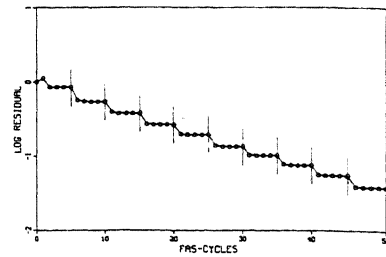
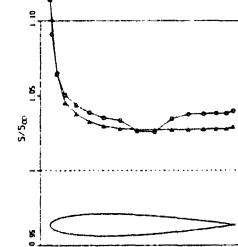
Upwind.



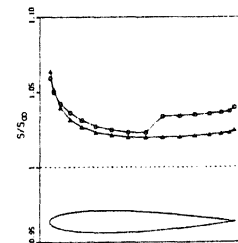
Upwind biased.



Superbox.

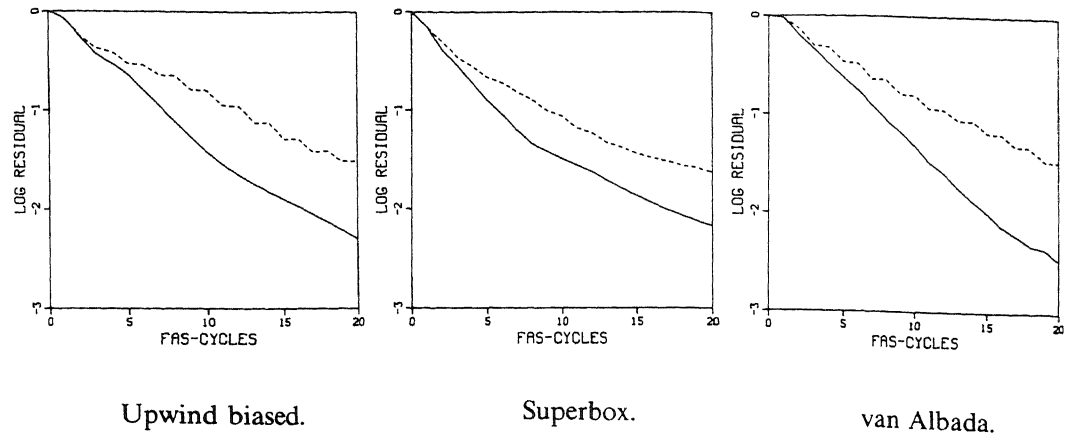


van Albada.

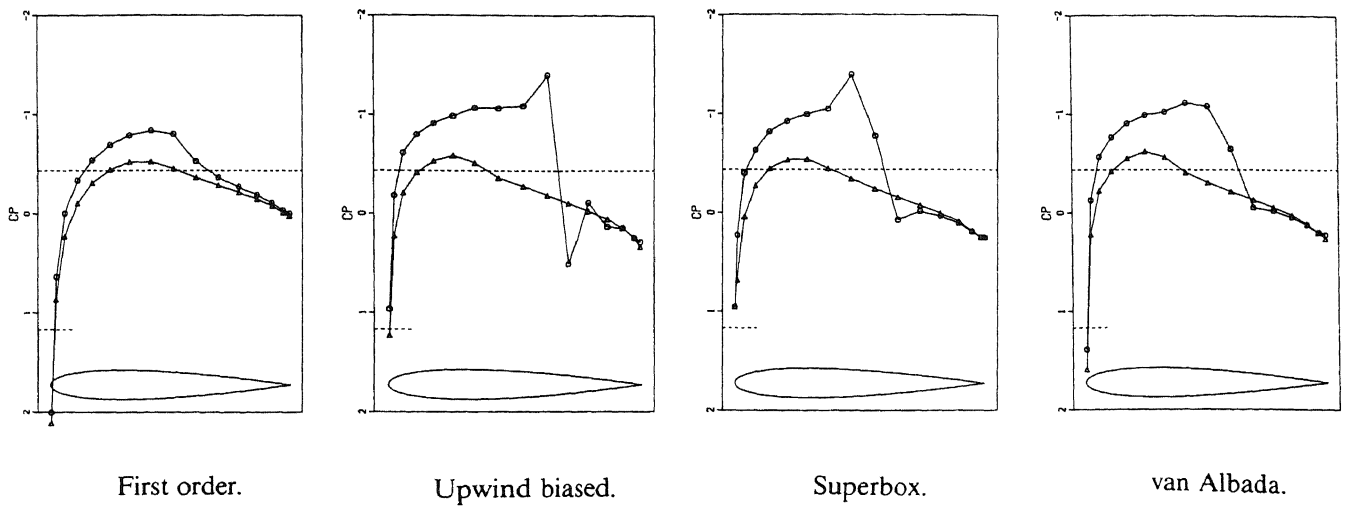


b. Convergence histories.

c. Surface entropy distributions ( $s = p\rho^{-\gamma}$ ).

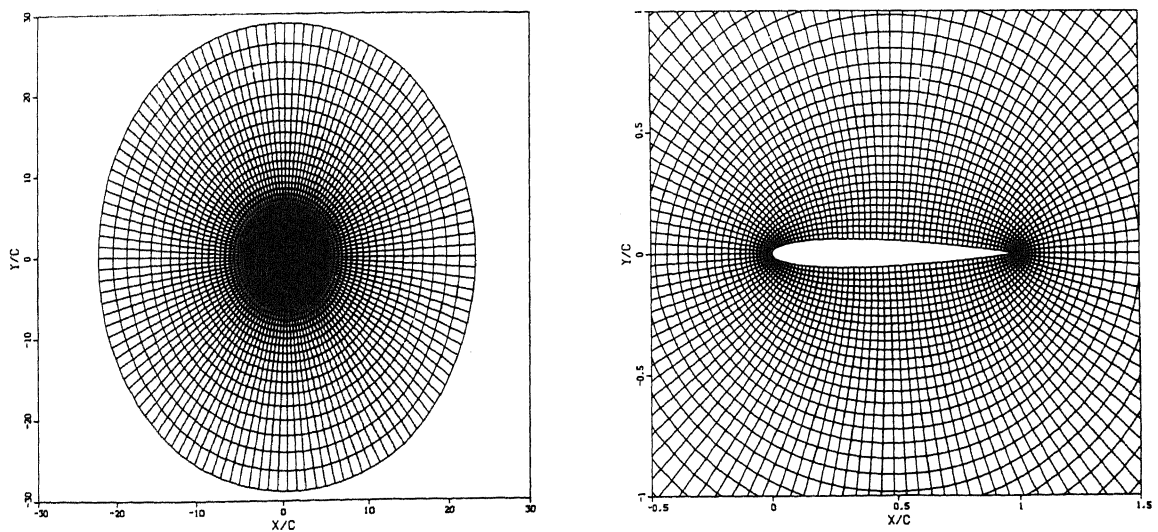


a. Convergence histories for 1 FAS-cycle per DCP-cycle (solid) and 2 FAS-cycles per DCP-cycle (dashed)



b. Converged surface pressure distributions.

Figure 5.2. Results obtained on  $32 \times 16$ -grid for NACA0012-airfoil at  $M_\infty = 0.8$ ,  $\alpha = 1.25^\circ$ , using 1 or 2 FAS-cycles per DCP-cycle.



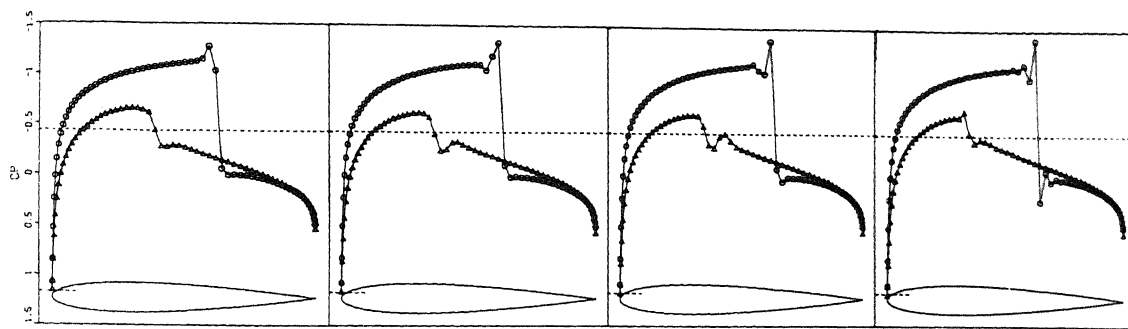
a.  $128 \times 64$ -grid.



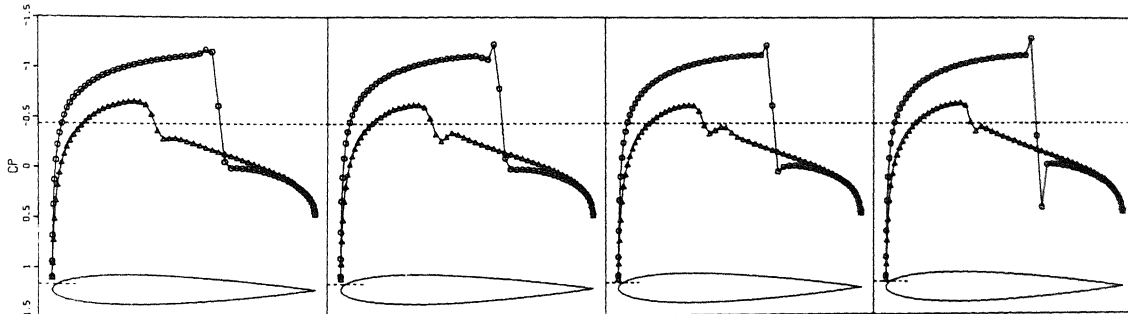
After 1 FAS-cycle.

After 10 FAS-cycles.

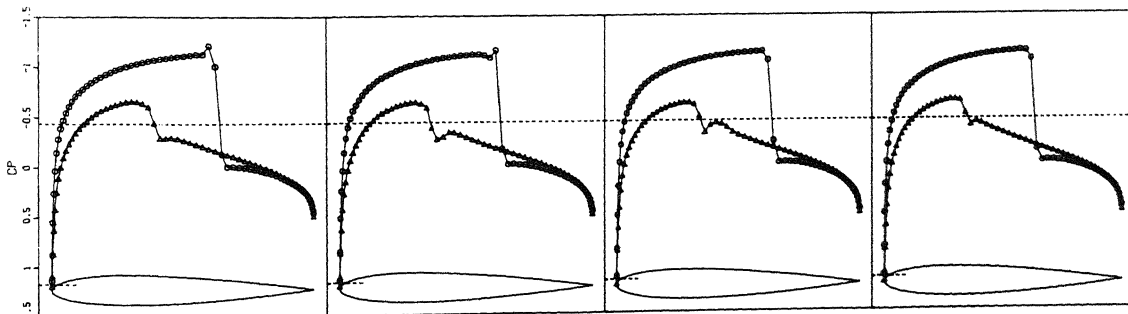
b. First-order.



Upwind biased



Superbox.

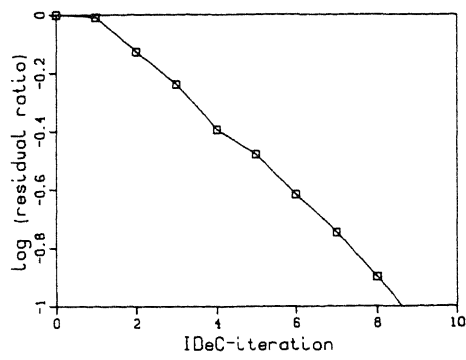


van Albada.

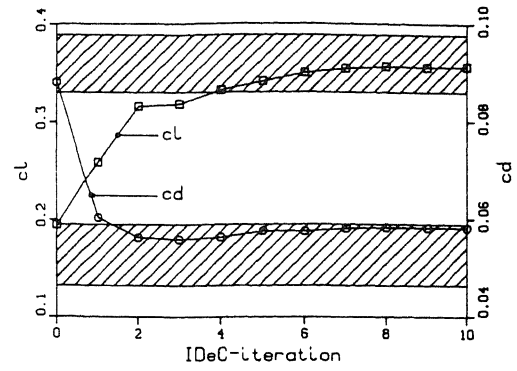
After 1st DCP-cycle. After 2nd DCP-cycle. After 3rd DCP-cycle. After 10th DCP-cycle.

c. Second-order.

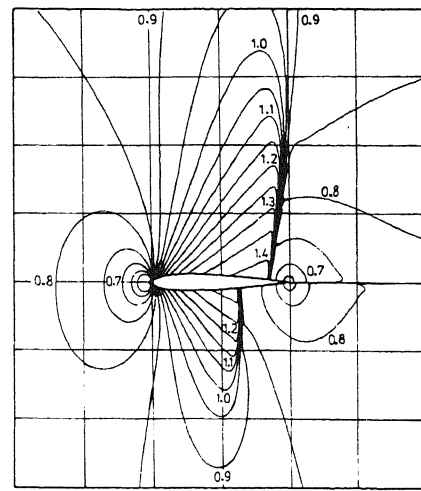
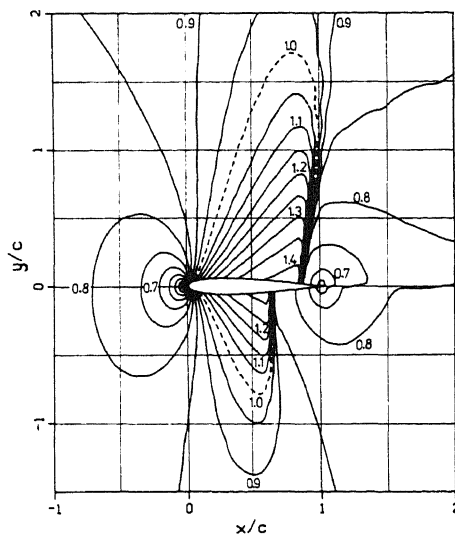
Figure 5.3. Surface pressure distributions obtained on  $128 \times 64$ -grid for NACA0012-airfoil at  $M_\infty = 0.8$ ,  $\alpha = 1.25^\circ$ .



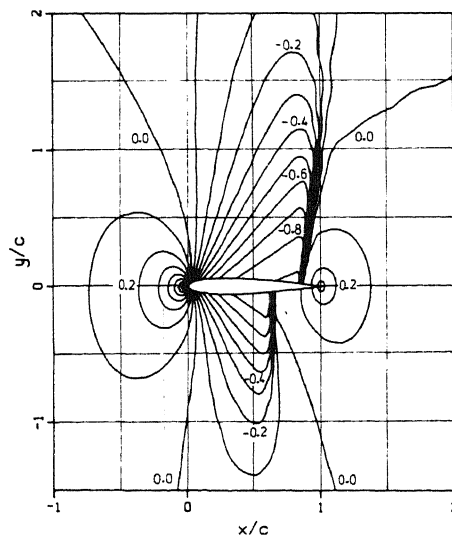
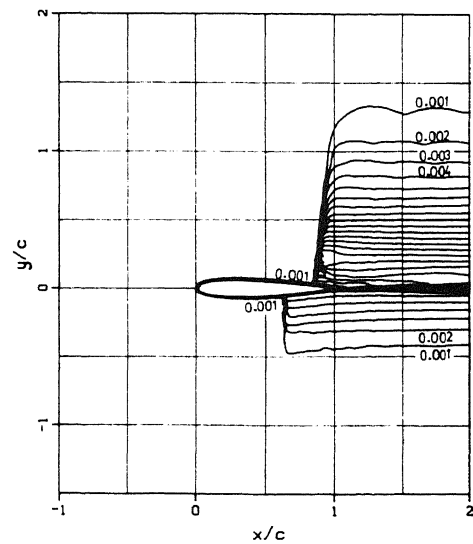
a. Convergence history residual ratio.

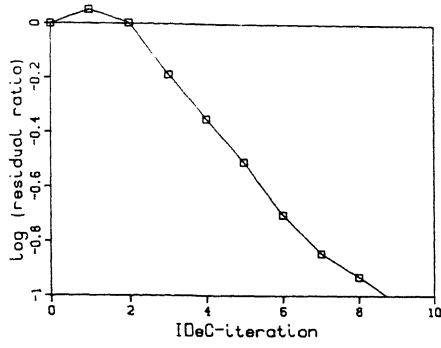


b. Convergence history lift and drag coefficient.

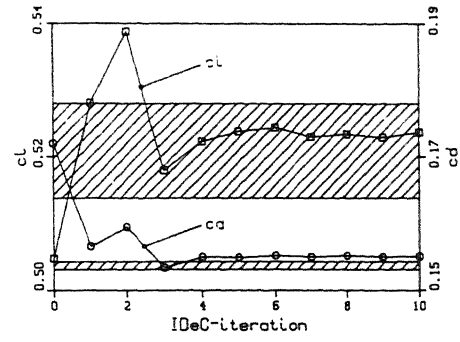


c. Mach number distributions; present result (left) and result Schmidt &amp; Jameson (right).

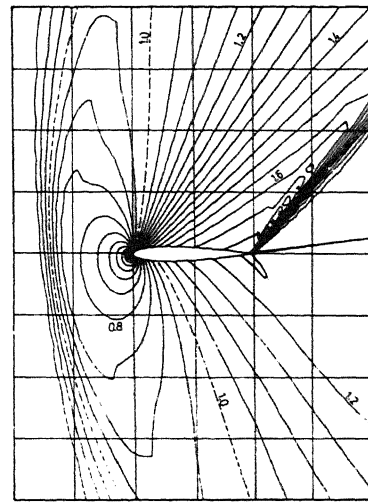
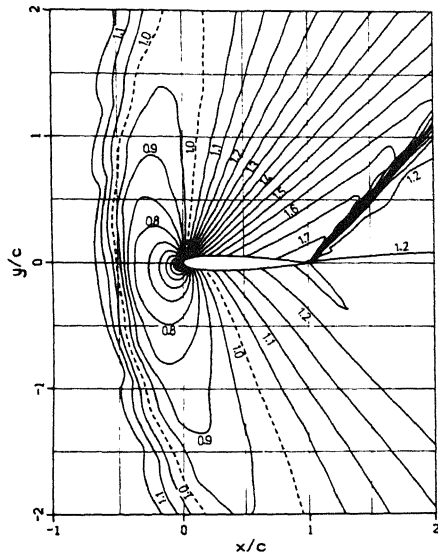
d. Present pressure distribution ( $c_p$ ).e. Present entropy distribution ( $s/s_\infty - 1$ ).Figure 5.4. Results obtained on  $128 \times 32$ -grid for NACA0012-airfoil at  $M_\infty = 0.85$ ,  $\alpha = 1^\circ$ .



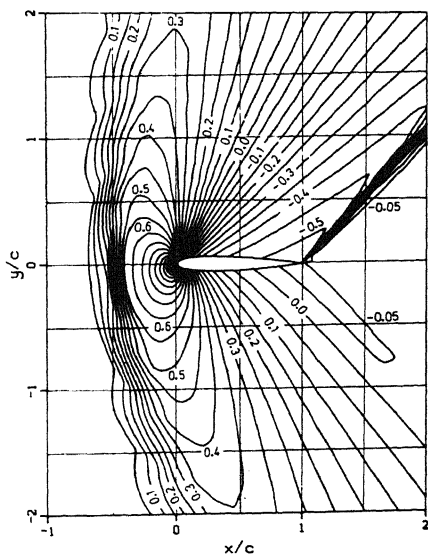
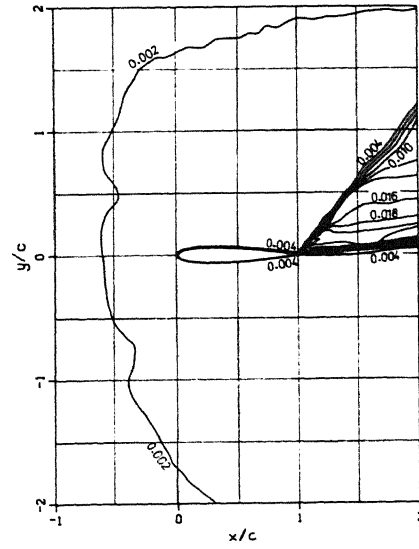
a. Convergence history residual ratio.

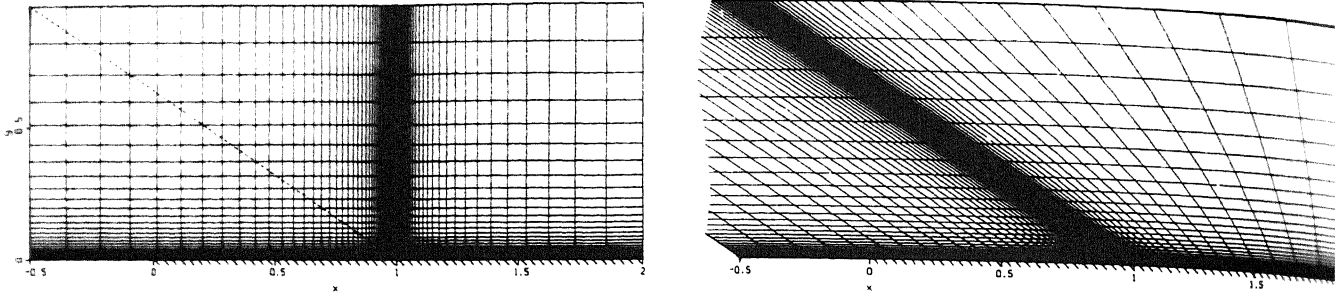


b. Convergence history lift and drag coefficient.

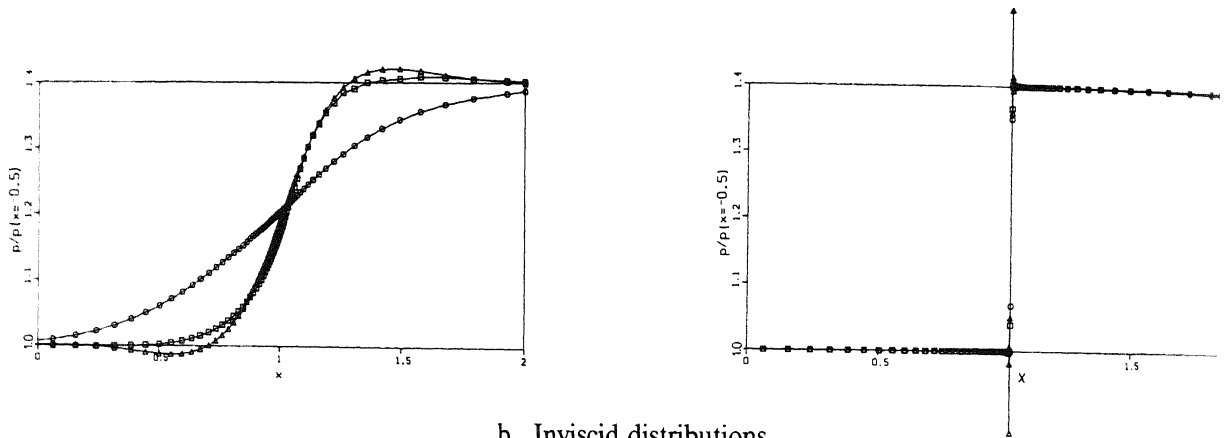


c. Mach number distributions; present result (left) and result Veuillot &amp; Vuillot (right).

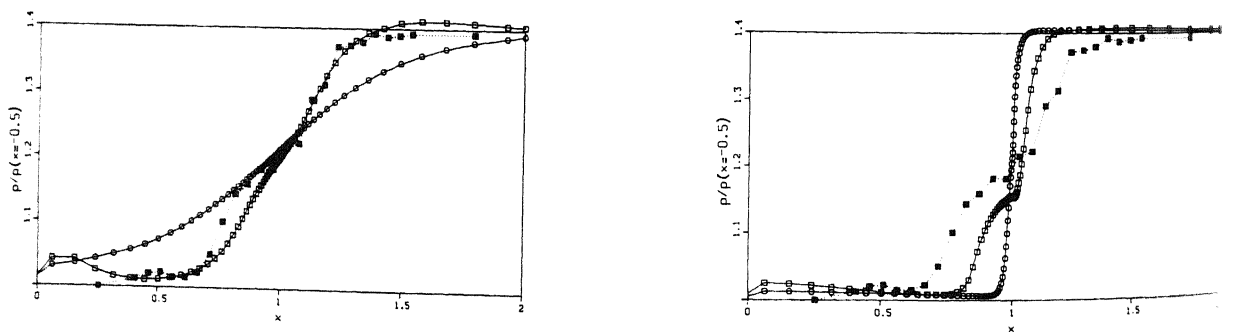
d. Present pressure distribution ( $c_p$ ).e. Present entropy distribution ( $s/s_\infty - 1$ ).Figure 5.5. Results obtained on  $128 \times 32$ -grid for NACA0012-airfoil at  $M_\infty = 1.2$ ,  $\alpha = 7^\circ$ .



a. Grids.



b. Inviscid distributions.  
 (○: first-order, △: non-limited higher-order, □: limited higher-order).



c. Viscous distributions.  
 (○: first-order, □: limited second-order, ■: measured).

Figure 6.1. Grids and corresponding surface pressure distributions.

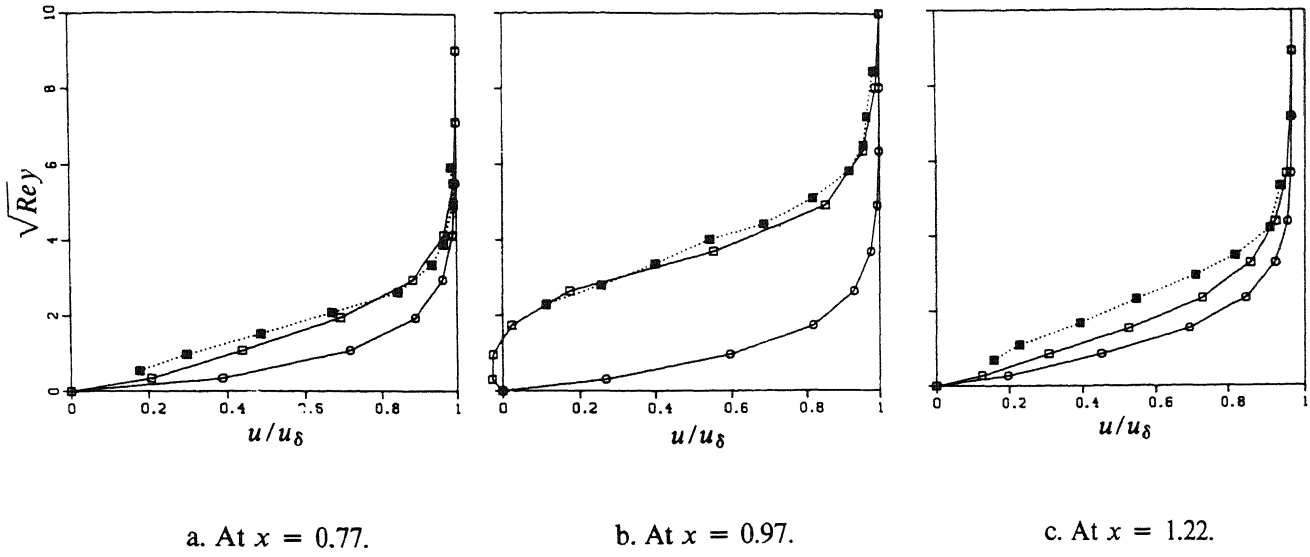


Figure 6.2. Velocity profiles ( $\circ$ : first-order,  $\square$ : limited second-order,  $\blacksquare$ : measured).