Standard Multigrid Techniques for CFD

VKI's 28th CFD Lecture Series

P.W. Hemker

with the cooperation of B.Koren

CWI
Kruislaan 413
NL 1089 SJ Amsterdam
The Netherlands

Contents

Chapter 1

Defect Correct Processes

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 unknow quantity or function; a right-hand-side $y \in \hat{D}$ and a 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. Frequently we cannot or do not want to solve the above mentioned equation directly, because this would exceed our computational capabilities. On the other hand we may be able 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 DCP: Defect Correction Process [69] [5].

Defect correction processes 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. consider the approximate operator $\tilde{F}$, to obtain an appropriate *correction* quantity,
- apply this correction to the initial approximation to obtain a new (hopefully better) approximation.

Of course, the above process may now be repeated, using the newly obtained approximation as a new 'initial' approximation.
A few instances of the basic principle are well known. We mention a few examples:

Example 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 $z_1, z_2, ...$ of approximations for the solution is then generated by

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

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 a local linearisation of F, given by:

$$\tilde{F}_i(z) = F(z_i) + F'(z_i)\,(z - z_i)\,,$$

where $F'(z_i)$ is defined by

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

Example 1.2 *Iterative refinement*
A second example of a DCP is the iterative refinement of a given approximate solution for 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 respectively lower and upper triangular matrices), and then solving the two triangular systems $Lw = y$, $Uz = w$ directly. 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:

$$z_{i+1} = z_i - e_i, \qquad i = 0, 1, 2, ...,$$

with e_i the solution of the system

$$LUe_i = d_i, \qquad \text{where} \quad d_i = Az_i - y.$$

1.2 Elementary Defect Correction Processes

We consider the equation

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

where $F : D \subset E \to \hat{D} \subset \hat{E}$ is a bijective, continuous, generally nonlinear operator; E and $\hat{E}$ are Banach spaces. The domain D and the range $\hat{D}$ are closed subsets given with F; $\hat{D}$ contains an appropriate neighbourhood of y. Hence, 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 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:

- We assume that the *defect* indexdefect

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

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

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

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

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

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

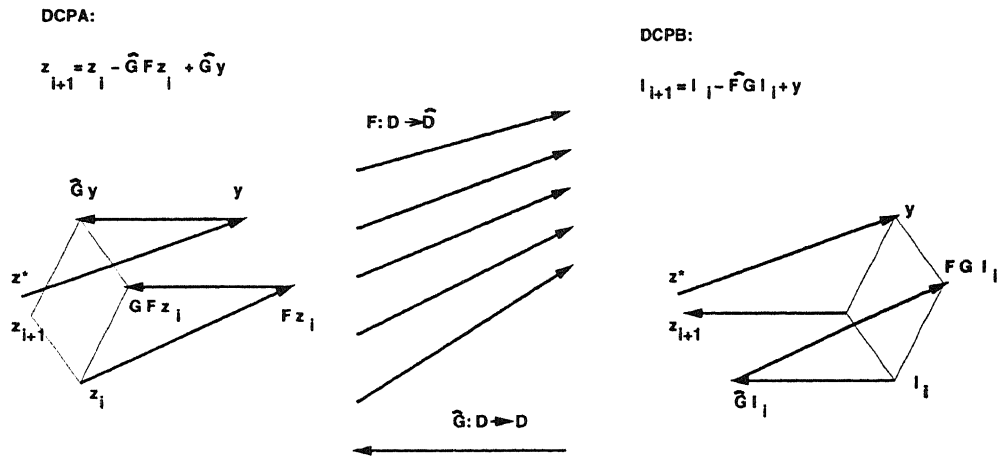


Figure 1.1: DCPA and DCPB

The basic defect correction processes are described by the the following iterative algorithms:

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

and

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

In the latter process we define approximations z_i by $z_i = \tilde{G}l_i$.

Remarks:

- In (DCPA) or (DCPB) we formulate both the iterative process and the (standard) initial approximant.
- 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).
- It is the existence of the approximate inverse $\tilde{G}$ which is essential, it is *not* the existence of the *approximate operator* $\tilde{F}$.

If and only if the mapping $\tilde{G} : \hat{D} \to D$ is injective, its left inverse $\tilde{F}$ exists. Similarly the right inverse $\tilde{F} : D \to \hat{D}$ of $\tilde{G}$ exists if and only if $\tilde{G}$ is surjective. In general our approximate inverse $\tilde{G}$ needs not 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

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

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 approach.

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

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

Definition 1.3 A mapping $f : X \to Y$ is called *affine* if there exists a constant element $c \in Y$ such that $f(\cdot) - c : X \to Y$ is a linear mapping.

This definition implies:

- $f(0) = c$,
- $f(x + y) = f(x) + f(y) - c$,
- $f(x - y + z) = f(x) - f(y) + f(z)$,

Theorem 1.4 If $\tilde{G}$ is an affine mapping, then the sequences $\{z_i\}$ in DCPA and $\{z_i\}$ in DCPB are identical.

Proof: Let $\{l_i\}_{i=0,1,2,\ldots}$, and $\{z_i\}_{i=0,1,2,\ldots}$, be defined as in DCPB, then:

$$\begin{aligned} z_0 &= \tilde{G}\,l_0 = \tilde{G}\,y, \\ z_{i+1} &= \tilde{G}\,l_{i+1} = \tilde{G}\,(l_i - F\tilde{G}\,l_i + y) \\ &= \tilde{G}\,l_i + \tilde{G}y - \tilde{G}\,(F\tilde{G}l_i) \\ &= z_i + \tilde{G}y - \tilde{G}Fz_i = (I - \tilde{G}F)z_i + \tilde{G}y. \end{aligned}$$

This means that the values from this sequence $\{z_i\}$ satisfy exactly the generation rules for the sequence $\{z_i\}$ from DCPA. Hence both sequences are identical. □

1.3 Convergence of the basic defect correction processes

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

Definition 1.5 F is called *bounded* if bounded subsets of D are mapped onto bounded subsets in $\hat{D}$, and F is called *Lipschitz* if

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

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 (mapping)* if $|||F||| < 1$; F is *non-expansive* if $|||F||| \leq 1$.

Apparently

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

and for a linear operator F

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

Definition 1.6 An iterative process $z^{(i+1)} = H(z^{(i)}, z^{(i-1)}, ...)$ has a *fixed point* $z^\star$ if $z^\star = H(z^\star, z^\star, ...)$.

Because D (or $\hat{D}$) is a closed subset of a Banach space E (or $\hat{E}$), the set D (or $\hat{D}$) is a complete metric space. For contraction mappings of a complete metric space into itself, we have the following important theorem.

Theorem 1.7 *(Banach's contraction principle).* Let M be a contraction mapping of a complete metric space D into itself. Then M has a unique fixed point u in D. Moreover, if x_0 is an arbitrary point in D, and $\{x_n\}$ is defined by

$$x_{n+1} = Mx_n, \quad (n = 0, 1, 2, ...),$$

then $\lim_{n\to\infty} x_n = u$ and

$$\|x_n - u\| \leq \frac{|||M|||^n}{1 - |||M|||} \|x_1 - x_0\| \tag{1.3}$$

Proof: Let $x_0\in D$ and let $x_{n+1} = Mx_n$, $(n = 0, 1, 2, ...)$, and set $k = |||M|||$. Then we have

$$\|x_{r+1} - x_{s+1}\| = \|Mx_r - Mx_s\| \leq k\|x_r - x_s\|$$

and hence

$$\|x_{r+1} - x_r\| \leq k^r\|x_1 - x_0\|.$$

For given p and q with $p > q$ we have

$$\begin{aligned} &\|x_p - x_q\| \leq k^q\|x_{p-q} - x_0\| \\ &\leq k^q\{\|x_{p-q} - x_{p-q-1}\| + ... + \|x_1 - x_0\|\} \\ &\leq k^q\{k^{p-q-1} + ... + k + 1\}\|x_1 - x_0\| \\ &\leq \tfrac{k^q}{1-k}\|x_1 - x_0\|. \end{aligned} \tag{1.4}$$

The right-hand side of eq.(1.4) tends to zero as $q \to \infty$. Hence the sequence $\{x_n\}$ is Cauchy, and since D is complete, $\{x_n\}$ converges to an element u of D. As $\|x_{n+1} - Mu\| = \|Mx_n - Mu\| \leq k\|x_n - u\|$ and $\|x_n - u\| \to 0$ for $n \to \infty$, we have $Mu = \lim_{n\to\infty} x_{n+1} = u$, i.e. u is a fixed point of M.

For uniqueness, suppose v is another fixed point of M, $v\in D$, $v \neq u$, and $v = Mv$, then $\|u - v\| = \|Mu - Mv\| \leq k\|u - v\|$. This gives $(1 - k)\|u - v\| \leq 0$. Since $1 - k > 0$ we have $\|u - v\| = 0$, i.e. $u = v$.

To obtain eq.(1.3) we have

$$\|u - x_n\| \leq \|u - x_p\| + \|x_p - x_n\| \leq \|u - x_p\| + \frac{k^n}{1-k}\|x_1 - x_0\|$$

for $n < p$ by (1.4). Letting $p \to \infty$, we obtain

$$\|u - x_n\| \leq \frac{k^n}{1-k}\|x_1 - x_0\|.$$

□

For many generalisations of the Banach contraction principle see [35]

1.3.1 Convergence of DCPA

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

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

Definition 1.8 We define the *amplification operator of the error* in DCPA to be

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

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:

$$\hat{z} = (I - \tilde{G}F)\hat{z} + \tilde{G}y,$$

and hence

$$\tilde{G}F\hat{z} = \tilde{G}y = \tilde{G}Fz^*.$$

As a direct consequence of this we find the following

Theorem 1.9 If DCPA has a fixed point $\hat{z} \in D$ 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).

The convergence of DCPA clearly depends on the *contractivity* of the amplification operator $M_A = I - \tilde{G}F$. We formulate the following

Theorem 1.10 Let $M_A : D \to D$ be a contraction and let $\tilde{G} : \hat{D} \to D$ be injective. Then DCPA converges to the solution z^* of (1.1).
Proof: The DCPA iteration operator $A : D \to D$ is given by:

$$Az = (I - \tilde{G}F)z + \tilde{G}y. \tag{1.5}$$

Clearly A is a contraction on (the complete metric space) D as well, hence a unique fixed point $\hat{z}$ for A exists and DCPA converges to this fixed point $\hat{z}$. By theorem 1.9 we know that $\hat{z}$ is a solution of (1.1). By assumption F is bijective, hence $\hat{z}$ is the unique solution of (1.1). □

Remark: Even, if $\tilde{G}$ is not injective, then 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 $\tilde{G}(\hat{D})$, 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 of DCPB

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

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

Definition 1.11 We define the *amplification operator of the residual* in DCPB to be

$$M_B = I - F\tilde{G}.$$

For any fixed point $\hat{l}$ of DCPB we have: $\hat{l} = (I - F\tilde{G})\,\hat{l} + y$, hence $F\tilde{G}\,\hat{l} = y = Fz^*$. As direct consequence of this, we find the following.

Theorem 1.12 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).

Remarks:

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

Theorem 1.13 Let $M_B : \hat{D} \to \hat{D}$ 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).

1.3.3 Convergence of DCPL

For DCPL (with $\tilde{G}$ 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 $\overline{M}_L = I - F\tilde{G}$ of the residual are both linear operators and for the convergence of the iteration we can consider $\|M_L\|$, $\|\overline{M}_L\|$, $\rho(M_L)$, $\rho(\overline{M}_L)$. (We notice that $\rho(M_L) = \rho(\overline{M}_L) = \lim_{n\to\infty}(\|M_L^n\|^{1/n})$.

Example 1.14 *Relaxation methods*
All *stationary, fully consistent iterative methods of degree one* 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 (cf. [76]). These iterative methods are defect correction processes of type A with approximate inverse $\tilde{G} = P^{-1}$. They are equivalent to the corresponding DCPBs, because P^{-1} is linear. Many of these methods (known as

relaxation methods) are often used for the solution of special sparse linear systems. We introduce the following notation:

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

L is a strictly lower triangular matrix $D = \text{diag}\,(A)$, a diagonal matrix, and U is a strictly upper triangular matrix. Using this notation, Table 1.1 summarises some possible choices for $\tilde{F} = P$, 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	Generalized Richardson	P non-singular diagonal matrix

Table 1.1: Relaxation methods

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

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

and

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

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 three examples: (i), (ii) and (iii).

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

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 non-linear equations, where $\tilde{G}(z) = (F'(z))^{-1}$, with F' the Fréchet derivative of the operator F in the problem (1.1). See e.g. example 1.1 .

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

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

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

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 and Atkinson's methods for the solution of Fredholm integral equations of the 2nd kind. (See [3] and [6]).

1.4.2 Combined DCPs

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

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

$$= (I - (\tilde{G} - \tilde{\tilde{G}}F\tilde{G} + \tilde{G})F)z_i + (\tilde{\tilde{G}} - \tilde{\tilde{G}}F\tilde{G} + \tilde{G})y.$$

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

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

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

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

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

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

$$\begin{pmatrix} z_{i+\sigma} \\ y \end{pmatrix} = \begin{pmatrix} I - \tilde{G}F & \tilde{G} \\ 0 & I \end{pmatrix}^{\sigma} \begin{pmatrix} z_i \\ y \end{pmatrix}$$

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

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

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

Then the relation

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

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

$$\hat{G} = \sum_{m=0}^{\sigma-1}(I-\tilde{G}F)^m\tilde{G} = [I-(I-\tilde{G}F)^\sigma]F^{-1}. \tag{1.6}$$

1.4.3 Iterative application of DCPs

We will now pay attention to another 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 $F^\star$. 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 and we will take better approximations that converge to $F^\star$ in some sense as the iteration proceeds.

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

$$(P_k) \qquad\qquad F_k z = y_k,$$

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

One possible application is to select $\{F_i\}$ which are discrete approximations of *higher and higher order* to a continuous operator F. The approximate inverse $\tilde{G} = F_0^{-1}$ may be kept 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* ([17]) or Full Multigrid (FMG) [9].

1.4.4 Recursive application of DCPs

Generally, 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 application of DCP's in which on the lowest level of recursion a very simple equation is to be solved. Multigrid iteration is an example of this procedure.

1.4.5 Generalisation for nonlinear problems: DCPN

A generalisation of DCPA, specially for nonlinear problems, can be introduced via DCPL. Let us first consider the process DCPA and suppose that $\tilde{G}$ is differentiable. Since $\tilde{G}'(\overline{y})$,

$\overline{y} \in \hat{D}$, 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 $\tilde{G}'(\overline{y})$:

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

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

$$\tilde{G}'(\overline{y})u = \mu \tilde{G}'(\overline{y})\,\frac{u}{\mu} \approx \mu \tilde{G}(\overline{y} + \frac{u}{\mu}) - \mu \tilde{G}(\overline{y}).$$

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

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

We then obtain the following DCP:

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

Remarks:

- In this new defect correction process one still has the freedom to choose the parameters μ and $\overline{y}$. For the choice $\overline{y} = y$ and $\mu = 1$, DCPN coincides with DCPA.
- For a large enough μ, we may guarantee that for any defect $Fz_i - y$, the operator $\tilde{G}$ is evaluated only in a sufficiently small neighbourhood of $\overline{y}$.
- In the general case (i.e. for arbitrary values of μ and $\overline{y}$), the solution z^* of (1.1) is a fixed point of DCPN. Sometimes the converse is also true:

Theorem 1.15 Let $\hat{z}$ be a fixed point of DCPN and let $\tilde{G} : \hat{D} \to D$ be injective. Then $\hat{z}$ is the solution of (1.1).
Proof: Because $\hat{z}$ is a fixed point of DCPN we immediately see that

$$\tilde{G}(\overline{y} + (F\hat{z} - y)/\mu) = \tilde{G}(\overline{y}).$$

By the assumption that $\tilde{G}$ is injective, we see that $F\hat{z} - y = 0$. □

Chapter 2

Defect correction and discretisation

2.1 Introduction

In the foregoing chapters we discussed defect correction and discretisation. It will now be shown how these techniques can be combined to approximate efficiently the solution of a continuous problem. The basic principle is that less accurate discretisations accelerate the solution of more accurate discretisations. More precisely

1. the solution of a lower order discretisation may accelerate the solution of a higher order discretisation, or
2. the solution of a coarser discretisation may accelerate the solution of a finer discretisation.

The first case will be considered in the next section, the latter will be treated in more detail in Section 2.3.

2.2 A fundamental theorem

Example 2.1 *An accurate and an inaccurate discretisation*
We consider a two-dimensional second-order linear elliptic boundary value problem, e.g. problem

$$\begin{cases} -\Delta u = f_\Omega & \text{on } \Omega, \\ u + \epsilon \frac{\partial u}{\partial n} = f_\Gamma & \text{on } \Gamma = \partial\Omega, \end{cases} \tag{2.1}$$

and two discretisations of it, both obtained by the finite element method. The first discretisation with piecewise linear functions on a triangularisation, is denoted by

$$F_h z_h = y_h, \quad F_h : E_h \to \hat{E}_h. \tag{2.2}$$

The other discretisation, with piecewise quadratics on the same triangularisation, is denoted by

$$F_h^* z_h = y_h^*, \quad F_h^* : E_h^* \to \hat{E}_h^*. \tag{2.3}$$

Because we use basicly the same triangularisation for both discretisations, we can identify E_h, E_h^*, $\hat{E}_h$, and $\hat{E}_h^*$ all with $\mathbb{R}^{N(h)}$, and the matrices F_h and F_h^* have the same dimension.

If the solution of the boundary value problem is smooth, the operator F_h^* will yield a more accurate approximation than F_h. From this point of view, it seems advantageous to solve (2.3) to obtain an approximation of the solution of the continuous problem.

Naturally, F_h^* has a more complex structure than F_h. Therefore we prefer to solve (2.2) and we shall use its solution as the initial value in a defect correction process for (2.3) as follows

$$\begin{cases} F_h z_h^{(1)} = y_h \\ F_h z_h^{(i+1)} = (F_h - F_h^*) z_h^{(i)} + y_h^* \quad i = 1, 2, \ldots . \end{cases} \tag{2.4}$$

A stationary point for the iteration (2.4) is a solution for (2.3). Later we shall see that in many cases one or a few of these iteration steps may be enough to obtain a sufficiently accurate approximation to the solution of the original problem. An approximation not less accurate than the one that will be obtained by direct solution of (2.3).
We will now place the above example in a more general context. We consider the continuous problem

$$Lu = f\,, \tag{2.5}$$

where $L : E^\alpha \to \hat{E}^\alpha$ 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. The solution of (2.5) is called $\hat{u}$. We consider a less accurate discretisation of (2.5):

$$L_h u_h = \bar{R}_h f \tag{2.6}$$

and a more accurate discretisation of (2.5):

$$L_h^* u_h = \bar{R}_h^* f \tag{2.7}$$

where $L_h, L_h^* : E_h^\alpha \to \hat{E}_h^\alpha$ are linear operators, and $\{E_h^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$, $\{\hat{E}_h^\alpha | \alpha_0 \leq \alpha \leq \alpha_1\}$ are scales of 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. As in the example, the defect correction process

$$\begin{cases} L_h u_h^{(1)} = \bar{R}_h f\,, \\ L_h u_h^{(i+1)} = (L_h - L_h^*) u_h^{(i)} + \bar{R}_h^* f\,, \quad i = 1, 2 \ldots , \end{cases} \tag{2.8}$$

is used to approximate the solution of (2.7). When the defect correction process (2.8) converges slowly (or does not converge) the use of (2.8) will be less efficient than the immediate solution of (2.7). The following theorem will show under what conditions the use of (2.8) might be preferable.

Theorem 2.2 Let the problem (2.5) be discretised by (2.6) and (2.7), and let

1. L_h be σ-stable for some $\sigma \geq 0$,
2. L_h be consistent of order $\tilde{p}$, for all $\tilde{p} \in [0, p]$,
3. L_h^* be consistent of order $\tilde{p}^*$, for all $\tilde{p}^* \in [0, p^*]$, $p^* > p$,
4. L_h and L_h^* be relatively consistent of order p.

Then the iterands in (2.8) satisfy the error estimate

$$\|u_h^{(i)} - R_h \hat{u}\|_{E_h^\omega} \leq C h^{\min(p^*, ip)} \|\hat{u}\|_{E^{\omega+\beta_i}}, \tag{2.9}$$

with $\beta_i = \min(\sigma + p^*, i(\sigma + p))$ and C independent of h and further ω, i such that $\omega, \omega + i(p + \sigma) \in [\alpha_0, \alpha_1]$.
Proof: We will first put the conditions 1) - 4) in a more explicit form:

1. $\exists C_1$ independent of h : $\|L_h^{-1}\|_{E_h^{\alpha-\sigma}\leftarrow\hat{E}_h^{\alpha}} \leq C_1$

2. p is a real number for which

$$\|\bar{R}_h L - L_h R_h\|_{\hat{E}_h^{\alpha}\leftarrow E^{\alpha+\tilde{p}}} \leq C_2 h^{\tilde{p}}$$

for all $0 \leq \tilde{p} \leq p$ and some C_2 independent of h. This implies that

$$\|\bar{R}_h L\hat{u} - L_h R_h \hat{u}\|_{\hat{E}_h^{\alpha}} = \|\bar{R}_h f - L_h R_h \hat{u}\|_{\hat{E}_h^{\alpha}}, \leq C_2 h^{\tilde{p}} \|\hat{u}\|_{E^{\alpha+\tilde{p}}} \tag{2.10}$$

for all $0 \leq \tilde{p} \leq p$.

3. p^* is a real number for which

$$\|\bar{R}_h^* L - L_h^* R_h\|_{\hat{E}_h^{\alpha}\leftarrow E^{\alpha+\tilde{p}}} \leq C_3 h^{\tilde{p}}$$

for all $0 \leq \tilde{p} \leq p$ and some C_3 independent of h. This implies that

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

for all $0 \leq \tilde{p} \leq p^*$

4.

$$\|L_h - L_h^*\|_{\hat{E}_h^{\alpha}\leftarrow E_h^{\alpha+\tilde{p}}} \leq C_4 h^{\tilde{p}}$$

for all $0 \leq \tilde{p} \leq p$ and some C_4 independent of h.

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

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

Hence

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

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

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

Because (2.7) is a discretisation of higher order than (2.6) we have $p^* \geq p$ and it follows that

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

and thus the theorem is proved for $i = 1$.

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

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

It now follows that

$$\|u_h^{(i+1)} - R_h\hat{u}\|_{E_h^{\omega}} \leq \|L_h^{-1}[L_h - L_h^*]\,[u_h^{(i)} - R_h\hat{u}]\|_{E_h^{\omega}} + \|L_h^{-1}[L_h^* R_h - \bar{R}_h^* L]\hat{u}\|_{E_h^{\omega}}$$

$$\leq \|L_h^{-1}\|_{E_h^\omega \leftarrow \hat{E}_h^{\omega+\sigma}} \|L_h - L_h^*\|_{\hat{E}_h^{\omega+\sigma} \leftarrow E_h^{\omega+\sigma+p}} \|u_h^{(i)} - R_h\hat{u}\|_{E_h^{\omega+\sigma+p}} +$$

$$\|L_h^{-1}\|_{E_h^\omega \leftarrow \hat{E}_h^{\omega+\sigma}} \|L_h^* R_h - \bar{R}_h^* L\|_{\hat{E}_h^{\omega+\sigma} \leftarrow E^{\omega+\sigma+p^*}} \|\hat{u}\|_{E^{\omega+\sigma+p^*}} .$$

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

$$\|u_h^{(i+1)} - R_h\hat{u}\|_{E_h^\omega} \leq C_1 C_4 h^p \|u_h^{(i)} - R_h\hat{u}\|_{E_h^{\omega+\sigma+p}} + C_1 C_3 h^{p^*} \|\hat{u}\|_{E^{\omega+\sigma+p^*}}.$$

Application of the induction hypothesis leads to

$$\|u_h^{(i+1)} - R_h\hat{u}\|_{E_h^\omega} \leq C_1 C_4 h^p C_5 h^{\min(p^*, ip)} \|\hat{u}\|_{E^{\omega+\sigma+p+\beta_i}} + C_1 C_3 h^{p^*} \|\hat{u}\|_{E^{\omega+\sigma+p^*}} \leq$$

$$\leq C_1 C_4 C_5 h^{\min(p^*,(i+1)p)} \|\hat{u}\|_{E^{\omega+\beta_{i+1}}} + C_1 C_3 h^{\min(p^*,(i+1)p)} \|\hat{u}\|_{E^{\omega+\beta_{i+1}}}$$

$$\leq C h^{\min(p^*,(i+1)p)} \|\hat{u}\|_{E^{\omega+\beta_{i+1}}} .$$

In the one but last inequality we used the fact that

$$\|\cdot\|_\alpha \leq \|\cdot\|_{\alpha+\eta} \text{ for } \eta \geq 0 .$$

□

Remarks:

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

$$\|L_h - L_h^*\|_{\hat{E}_h^\alpha \leftarrow E_h^{\alpha+\tilde{p}}} = \|(L_h - L_h^*) R_h \hat{P}_h\|_{\hat{E}_h^\alpha \leftarrow E_h^{\alpha+\tilde{p}}}$$

$$\leq \|L_h R_h - L_h^* R_h\|_{\hat{E}_h^\alpha \leftarrow E^{\alpha+\tilde{p}}} \|\hat{P}_h\|_{E^{\alpha+\tilde{p}} \leftarrow E_h^{\alpha+\tilde{p}}}$$

$$\leq C_6 \|L_h R_h - \bar{R}_h L + \bar{R}_h L - L_h^* R_h\|_{\hat{E}_h^\alpha \leftarrow E^{\alpha+\tilde{p}}}$$

$$\leq C_6 \{\|L_h R_h - \bar{R}_h L\|_{\hat{E}_h^\alpha \leftarrow E^{\alpha+\tilde{p}}} + \|\bar{R}_h L - L_h^* R_h\|_{\hat{E}_h^\alpha \leftarrow E^{\alpha+\tilde{p}}}\}$$

$$\leq C_6 \{C_2 h^{\tilde{p}} + C_3 h^{\tilde{p}}\} \leq C h^{\tilde{p}} , \text{ for all } 0 \leq \tilde{p} \leq p .$$

2.3 DCP with an approximate inverse of deficient rank

Example 2.3 *A fine and a coarser discretisation*
Here we show how a coarser discretisation can be used to accelerate the solution process for a finer discretisation. Again we take as a model problem the two-dimensional second-order linear elliptic boundary value problem. We are interested in a coarse and a fine finite-element discretisation of this problem with piecewise linear functions on a triangularisation. Again the finer discretisation is denoted by

$$F_h z_h = y_h , \tag{2.11}$$

and the coarser by

$$F_H z_H = y_H \,. \tag{2.12}$$

It is assumed that the two discretisations are nested (e.g. $H = 2h$), i.e. that there exist a prolongation P_{hH} and a restriction $\bar{R}_{Hh}$ with property

$$P_H = P_h P_{hH} \,,\; \bar{R}_H = \bar{R}_{Hh} \bar{R}_h \,.$$

It is clear that the solution of (2.11) is much more expensive to compute than the solution of (2.12), because the size of matrix for F_h is larger than that for F_H. For this reason we are inclined to use the solution of (2.12) as an auxiliary problem in a defect-correction process for the solution of (2.11). It seems natural to use $P_{hH} F_H^{-1} \bar{R}_{Hh}$ as an approximate inverse for F_h. This results in the *coarse grid correction process*

$$\begin{cases} z_h^{(1)} & = P_{hH} z_H \\ z_h^{(i+1)} & = z_h^{(i)} - P_{hH} F_H^{-1} \bar{R}_{Hh} (F_h z_h^{(i)} - y_h) \end{cases} \tag{2.13}$$

Notice that the operator $P_{hH} F_H^{-1} \bar{R}_{Hh}$ is not full rank: $\mathrm{Rank}(P_{hH} F_H^{-1} \bar{R}_{Hh}) = N(H) < N(h)$.
The following figure may illustrate this

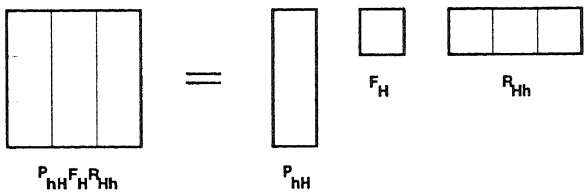


Figure 2.1: The incomplete rank matrix

From now on the above example will be generalised. The more general form of the iteration step in (2.13) is

$$z^{(i+1)} = z^{(i)} - \tilde{G}(F z^{(i)} - y) \,, \tag{2.14}$$

where

$\tilde{G} = PS\bar{R} : \hat{E}_h \to E_h$ is not full rank,
$S : \hat{E}_H \to E_H$ is full rank,
$\bar{R} : \hat{E}_h \to \hat{E}_H$ is a restriction and,
$P : E_H \to E_h$ is a prolongation.,
$F : E_h \to \hat{E}_h$ is full rank.

Definition 2.4 Let $S, \bar{R}, P$ and F be as above. Then we call the operator $E : E_H \to \hat{E}_H$, defined by

$$E = S^{-1} - \bar{R} F P \,,$$

the *deviation from the Galerkin approximation* corresponding to (2.14).

Notice that with the use of Galerkin approximation $S = (RFP)^{-1}$, we have $E = 0$.

Definition 2.5 Let $\hat{R}$ be a restriction related to P, i.e. its left inverse, $\hat{R} = (P^T P)^{-1} P^T$. Let e be the error in the approximation $z^{(i)}$, i.e. $e = z^{(i)} - F^{-1} y$. Then we define e_s, the *smooth part of the error,* and e_u, the *unsmooth* or *rapidly varying part of the error* by

$$e_s = P\hat{R}e; \quad e_u = (I - P\hat{R})e \,.$$

Remark: The operator $P\hat{R}$ represents the projection of E_h on Range$(P) \subset E_h$. The smooth part of the error is the part of the error lying in Range (P), the remaining part, e_u, lying in its complement Range$(P)^T$.

We now want to examine the effect of the amplification operator of the error, $M = I - \tilde{G}F$, on an error $e = e_s + e_u$. We examine e_s and e_u separately. Since $e_s \in$ Range(P), we have $e_s = P\hat{R}e_s$, and

$$\begin{aligned} Me_s = \; & MP\hat{R}e_s = (I - PS\bar{R}F)P\hat{R}e_s = P(I - S\bar{R}FP)\hat{R}e_s \\ & = P(I - S(S^{-1} - E))\hat{R}e_s = PSE\hat{R}e_s \in \text{Range}(P), \text{ and} \\ Me_u = \; & (I - PS\bar{R}EF)e_u = e_u - PS\bar{R}Fe_u. \end{aligned}$$

So we have the situation as shown in Figure 2.2. We can follow the same procedure for the residual.

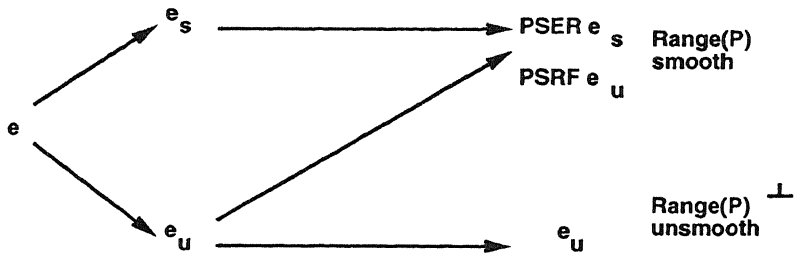


Figure 2.2: The effect of the coarse grid correction on the error

Definition 2.6 Let $\hat{\bar{P}}$ be a prolongation related to $\bar{R}$, i.e. a right inverse, e.g. $\hat{\bar{P}} = \bar{R}^T(R\bar{R}^T)^{-1}$. Let r be the residual for some approximation $z^{(i)}$, i.e. $r = Fz^{(i)} - y$. Then we define r_s, *the smooth part of the residual,* and r_u, the *unsmooth* or *rapidly varying part of the residual* by

$$r_s = \hat{\bar{P}}\bar{R}r; \quad r_u = (I - \hat{\bar{P}}\bar{R})r.$$

Remark: The operator $\hat{\bar{P}}\bar{R}$ represents the projection of $\hat{E}_h$ on the complement of Kernel$(\bar{R})$ and $r_u \in$ Kernel$(\bar{R})$.

The effect of the amplification operator of the residual, $\overline{M} = I - F\tilde{G}$, on $r = r_s + r_u$, can now be derived from its effect on r_s and r_u:

$$\begin{aligned} \overline{M}r_s &= (I - FPS\bar{R})r_s \\ &= (I - \hat{\bar{P}}\bar{R}FPS\bar{R} + \hat{\bar{P}}\bar{R}FPS\bar{R} - FPS\bar{R})r_s \\ &= (I - \hat{\bar{P}}(S^{-1} - E)S\bar{R} + (\hat{\bar{P}}\bar{R} - I)FPS\bar{R})r_s \\ &= (I + \hat{\bar{P}}ES\bar{R} - \hat{\bar{P}}\bar{R} - (I - \hat{\bar{P}}\bar{R})FPS\bar{R})r_s \\ &= \hat{\bar{P}}ES\bar{R}r_s - (I - \hat{\bar{P}}\bar{R})F\tilde{G}r_s, \end{aligned}$$

where $(I - \hat{\bar{P}}\bar{R})F\tilde{G}r_s \in$ Kernel$(\bar{R})$, and $\hat{\bar{P}}ES\bar{R}r_s \in$ Kernel$(\bar{R})^T$, the complement of Kernel(R), because $\hat{\bar{P}}\bar{R}(\hat{\bar{P}}ES\bar{R}r_s) = \hat{\bar{P}}ES\bar{R}r_s$.
Furthermore, because $r_u \in$ Kernel$(\bar{R})$, we have

$$\overline{M}r_u = (I - FPS\bar{R})r_u = r_u,$$

We summarise this in Figure 2.3.

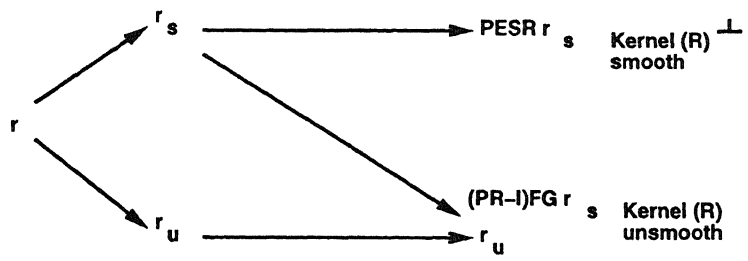


Figure 2.3: The effect of the coarse grid correction on the residual

Chapter 3

Multigrid algorithms

3.1 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 this problem on grids with meshwidth h and H:

$$F_h z_h = y_h \quad \text{and} \quad F_H z_H = y_H,\ H > h. \tag{3.1}$$

The operators $F_h : D_h \subset E_h \to \hat{D}_h \subset \hat{E}_h$ and $F_H : D_H \subset E_H \to \hat{D}_H \subset \hat{E}_H$ are mappings between discrete spaces E_h, $\hat{E}_h$, E_H and $\hat{E}_H$. Further a linear injection $P_{hH} : E_H \to E_h$ (*prolongation* or *interpolation*) and a linear surjection $R_{Hh} : \hat{E}_h \to \hat{E}_H$ (*restriction*) are given. A multigrid algorithm for the approximate solution of a discretised problem $F_h z_h = y_h$ is an iterative process in which one iteration cycle consists of

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

The relaxation steps are defect correction steps as e.g. damped Jacobi (JOR), Gauss-Seidel (GS), symmetric Gauss-Seidel (SGS), incomplete LU-decomposition iteration (ILU) etc. Their main purpose is to reduce the unsmooth part of the error (see Section 2.3). The remaining, smooth error 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

$$\tilde{G}_h = P_{hH} F_H^{-1} \bar{R}_{Hh} \tag{3.2}$$

(see Figure 3.1). The use of the approximate inverse (3.2) implies that we solve the defect equation on a coarse grid, with the help of a coarse discretisation of our problem. This is only meaningful if (part of) the error before the defect correction step is representable on the coarse grid.

Thus far we have only described an algorithm using two grids. If we do not solve the defect equation on a coarser grid directly, 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 a discretised problem directly 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*.

3.2 Two-level algorithms

Again we consider the two discretisations (3.1). Their relation via restrictions and prolongations is shown in Figure 3.1. We present the algorithms in the form of ALGOL-like

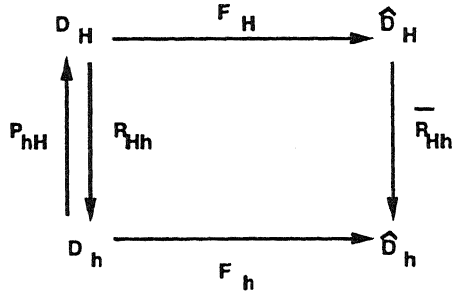


Figure 3.1: Relations between operators and spaces in a two-grid algorithm

programmes. First we describe two auxiliary procedures:

- **proc** solve = (**operator** F_h, **ref vector** z_h, **vector** y_h) **void:**
  This procedure uses the operator F_h and the right-hand-side y_h to solve (approximately) 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, **ref vector** z_h, **vector** y_h) **void:**
  This procedure performs one 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

$$F_h(z_h - e_h) = y_h = r_h + F_h z_h . \tag{3.3}$$

For a linear operator F_h this reduces to

$$F_h e_h = -r_h . \tag{3.4}$$

Instead of solving equation (3.3) directly, we compute the solution of a similar equation on a coarse grid

$$F_H w_H = \bar{R}_{Hh} r_h + F_H R_{Hh} z_h \tag{3.5}$$

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

$$F_H e_H = -\bar{R}_{Hh} r_h ,$$

which gives the correction quantity $P_{hH} e_H$. Now we describe one iteration step of the nonlinear two-level algorithm:

proc TGM = (**int** p, q, **ref vector** z_h, y_h) **void**:
begin
  to p **do** relax (F_h, z_h, y_h) ;
  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);
end;

For a linear operator F_h we can use the following simplified version:

proc TGML = (**int** p, q, **ref vector** z_h, y_h) **void**:
begin
  to p **do** relax (F_h, z_h, y_h);
  vector $r_H := \bar{R}_{Hh}(y_h - F_h z_h)$;
  vector $w_H := 0$;
  solve (F_H, w_H, r_H);
  $z_h := z_h + P_{hH} w_H$;
  to q **do** relax (F_h, z_h, y_h);
end;

Remark: The coarse-grid correction in the two-level algorithm can easily be seen as the instantiation of a DCPN as treated in Section 1.4.5. Therefore we take in (DCPN): $F := F_h$; $y := u_h$; $\tilde{G} := P_{hH} F_H^{-1} \overline{R}_{Hh}$ and $\overline{y} := \overline{P}_{hH} F_H R_{Hh} u_h^{(i)}$ with $\overline{P}_{hH}$ such that $\overline{R}_{Hh} \overline{P}_{hH} = I_H$. Then we find $\tilde{G}y = P_{hH} R_{Hh} u_h^{(i)}$ and with $u_H^{\text{old}} = R_{Hh} u_h^{(i)}$ and u_H^{new} the solution of

$$F_H u_H^{\text{new}} = F_H u_H^{\text{old}} - \overline{R}_{Hh}(F_h u_h^{(i)} - y_h)/\mu$$

the (DCPN) reads as

$$u_h^{(i+1)} = u_h^{(i)} - \mu P_{hH}(u_H^{\text{new}} - u_H^{\text{old}}) \ .$$

3.3 Multi-level algorithms

Consider a sequence of grids with meshwidths h_i, $h_{i-1} > h_i$, $i = 0, 1, 2, \ldots$. Often one uses $h_{i-1} = 2h_i$. We now describe one cycle of a multi-level algorithm to solve the problem $F_{h_i} z_{h_i} = r_{h_i}$. The algorithm uses a sequence of approximate solutions $z = [z_{h_i}, z_{h_{i-1}}, \ldots, z_{h_0}]$ and a sequence of right-hand-sides $r = [r_{h_i}, r_{h_{i-1}}, \ldots, r_{h_0}]$. At entrance these data are given only for the finest grid.

proc MGM = (**int** i, p, q, σ, **ref vector** z, r) **void**:
begin
  operator $F_h = F_{h_i}, F_H = F_{h_{i-1}}$;
  vector $z_h = z_{h_i}$, $y_h = r_{h_i}$;
  to p **do** relax (F_h, z_h, y_h);
  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 $(i-1, p, q, \sigma, z, r)$
fi;
$z_h := z_h + P_{hH}(w_H - R_{Hh} z_h)$;
to q **do** relax (F_h, z_h, y_h); **end**;

For a linear operator F_h we have the following simplified version:

proc MGML = (**int** i, p, q, σ,**ref** [ ] **vector** z, r) **void**:
begin
operator $F_h = F_{h_i}, F_H = F_{h_{i-1}}$;
vector $z_h = z_{h_i}, y_h = r_{h_i}$;
to p **do** relax (F_h, z_h, y_h);
vector $r_H = r_{h_{i-1}} := \bar{R}_{Hh}(y_h - F_h z_h)$;
vector $w_H = z_{h_{i-1}} := 0$;
if $i = 0$
then solve (F_H, w_H, r_H)
else to σ **do** MGML$(i-1, p, q, \sigma, z, r)$
fi;
$z_h := z_h + P_{hH} w_H$;
to q **do** relax (F_h, z_h, y_h);
end;

Multigrid methods based upon MGM, respectively MGML, are also know by the names FAS resp. CS. These names stand for *Full Approximation Storage scheme* and *Correction Storage scheme* respectively. As the names already indicate, with CS we only compute corrections on the coarser grid, whereas with FAS we compute approximate solutions on all levels. When the MGML-algorithm has converged we have $F_h z_h = y_h$; when the MGM-algorithm has converged we have in addition $z_{h_{i-1}} = R_{h_{i-1}h_i} z_{h_i}$, i.e. on the coarse grids we find the restriction of the solution on the fine grid.

Theorem 3.1 Consider an application of MGM where $h_i/h_{i+1} = H/h$ is constant for all $i = 0, 1, 2, \ldots$. 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, restriction and prolongation on 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

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

The above series converges if and only if $\sigma < (H/h)^d$. This implies that W_{tot} is proportional to W if σ is sufficiently small. □

In the above multigrid algorithms (MGM, MGML) the fixed numbers p, q and σ determine the *strategy* of the algorithm. Other multigrid algorithms may terminate iterations sooner or later, 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 σ have a *fixed strategy.* MG-cycles with $\sigma = 1$ are called V-cycles, those with $\sigma = 2$ are called W-cycles. V-cycles that have either $p = 0$ or $q = 0$ are called *sawtooth cycles.*

In the following figures 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 $\overline{M}_h^{REL}$ denote the amplification operators of the

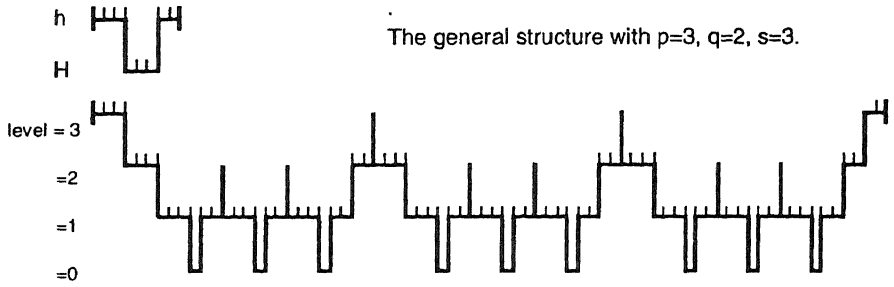


Figure 3.2: A general multigrid cycle with $p = 3, q = 2, \sigma = 3$.

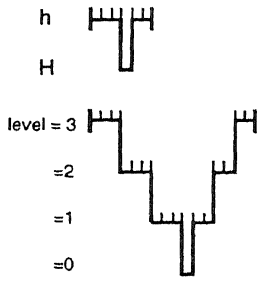


Figure 3.3: A V-cycle with $p = 3, q = 2$.

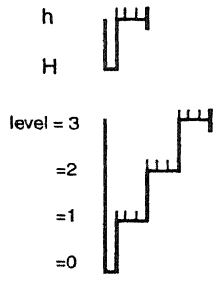


Figure 3.4: A saw-tooth cycle with $p = 0, q = 3, \sigma = 1$.

relaxation for error and residual respectively. For TGML the amplification operator for the error is given by

$$M_h^{TGML,p,q} = (M_h^{REL})^q (I_h - P_{hH} F_H^{-1} \tilde{R}_{Hh} F_h)(M_h^{REL})^p$$

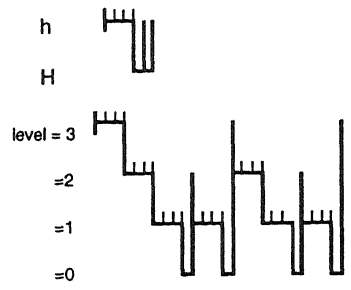


Figure 3.5: A W-cycle with $p = 3, q = 0, \sigma = 2$

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

We denote the amplification operator of a multi-level iteration step (MGML) on the h-level of discretisation by $M_h^{MGML,p,q,\sigma}$, or M_h^{MGML} for short. The same amplification operator on the next coarser level we denote by $M_H^{MGML,p,q,\sigma}$ or M_H^{MGML}. 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 σ steps of a DCP. The amplification operator of this DCP is given by M_H^{MGML}. Hence the approximate inverse of the σ iteration steps together is given by

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

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

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

and we have

$$\begin{aligned} M_h^{MGML,p,q,\sigma} &= (M_h^{REL})^q (I_h - P_{hH}(I_H - (M_H^{MGML})^\sigma) F_H^{-1} \bar{R}_{Hh} F_h)(M_h^{REL})^p \\ &= M_h^{TGML} + (M_h^{REL})^q P_{hH} (M_H^{MGML})^\sigma F_H^{-1} \bar{R}_{Hh} F_h (M_h^{REL})^p . \end{aligned} \tag{3.6}$$

3.4 Full multigrid method (FMG)

The multigrid cycles we described in the previous section 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 previously been 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,[ ] **int** i, **ref** [ ] **vector** z, y) **void**:
begin
  solve $(F_0, z[0], y[0])$; { – sufficiently accurate – }
  for k **to** l
  begin
    $z[k] := P^*_{k,k-1} z[k-1]$;

```
        for m to i[k] iteration (k, z[k], y[k]);
    end
end
```

Remarks:

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

The procedure "iteration" represents one step of a suitable iterative solution process. In the multigrid context we will use MGM (FAS) or MGML (CS) 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:

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

A typical FMG algorithm with one V-cycle ($p = \nu_1, q = \nu_2, \sigma = 1$) per coarse grid, is shown in figure 5.4.1. (i.e. we have $i[k] = 1$ for all k).

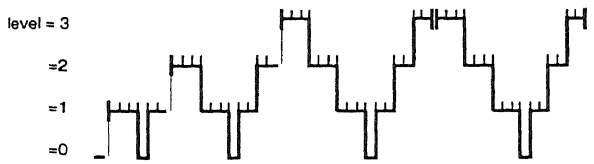


Figure 3.6: A FMG starting phase and an additional V-cycle

The strength of the FMG method is the combination of the sufficiently accurate initial estimate obtained by interpolation from the coarse grids, together with the convergence rate of the MG cycling procedure that is independent of the meshwidth. This combination makes that finally a strategy is found by which the solution of the discrete system on the finest grid is obtained with an accuracy that is of the same order of magnitude as the truncation error for this finest grid, by an amount of work that is only directly proportional to the number of degrees of freedom in the finest grid. Thus, the solution of the continuous problem is approximated up to truncation error accuracy by an amount of work that is proportional to the number of points in the finest grid. This is shown in the following theorem.

Theorem 3.2 If we consider a Nested Iteration process with a sequence of discretisations $F_k z_k = y_k$, $k = 0, 1, 2, \ldots$,

$$F_k = F_{h_k}, \quad z_k = z_{h_k}, \quad y_k = y_{h_k}, \qquad h_0 > h_1 > h_2 > \ldots , \tag{3.7}$$

such that (i) $h_{k-1}/h_k \leq C_1$ for all k, (ii) the discretisations $F_k z_k = y_k$ and $F_{k-1} z_{k-1} = y_{k-1}$ are relatively convergent of order p, i.e.

$$\|P^*_{k,k-1} z_{k-1} - z_k\| \leq C_0 h^p_{k-1} \,, \tag{3.8}$$

and (iii) the convergence factor of the iterative procedure "iteration" is independent of h, i.e.

$$\|z^{(j+1)}_k - z_k\| \leq C_2 \|z^{(j)}_k - z_k\| \,,$$

with C_2 independent of h. Then, *with* $i[k] = i$ *independent of* k, the result of the nested iteration procedure $\tilde{z}_k = z^{(i)}_k$ satisfies

$$\|\tilde{z}_k - z_k\|_{E_k} \leq \frac{C^p_1 C^i_2}{1 - C^p_1 C^i_2 \|P\|} C_0 h^p_k \,, \tag{3.9}$$

where $\|P\| = \sup_k \|P^*_{k,k-1}\|$ and it is assumed that $C^p_1 C^i_2 \|P\| \leq 1$.
Proof: For each $k \geq 0$ we know

$$\|z^{(i)}_k - z_k\| \leq C^i_2 \, \|z^{(0)}_k - z_k\| \,.$$

In the nested iteration algorithm it is required that z_0 is computed sufficiently accurate. We require

$$\|\tilde{z}_0 - z_0\| \leq C^p_1 \; C^i_2 \; C_0 h^p_0,$$

i.e. we require that truncation error accuracy has been attained. Then the statement of the theorem is satisfied for $k = 0$. For an arbitrary $k > 0$ we use induction. We assume that the theorem is satisfied for $k - 1$, then

$$\begin{aligned} \|z^{(0)}_k - z_k\| &= \|P^*_{k,k-1} z^{(i)}_{k-1} - z_k\| \\ &\leq \|P^*_{k,k-1}\| \; \|z^{(i)}_{k-1} - z_{k-1}\| + \|P^*_{k,k-1} z_{k-1} - z_k\| \\ &\leq \|P\| \; \|z^{(i)}_{k-1} - z_{k-1}\| + C_0 h^p_{k-1} \end{aligned}$$

so that

$$\begin{aligned} \|z^{(i)}_k - z_k\| &\leq C^i_2 \; \|P\| \; \|z^{(i)}_{k-1} - z_{k-1}\| + C^i_2 C^p_1 C_0 h^p_k \\ &\leq C^i_2 C^p_1 C_0 h^p_k + C^i_2 \; \|P\| \; \{C^i_2 C^p_1 C_0 h^p_{k-1} + C^i_2 \; \|P\| \; \{...\}\} \\ &= C_0 C^p_1 C^i_2 \, [h^p_k + C^i_2 \|P\| \{h^p_{k-1} + C^i_2 \|P\| \{...\}\}] \\ &= C_0 C^p_1 C^i_2 h^p_k \, [1 + C^i_2 \|P\| \{C^p_1 + C^i_2 \|P\| \{...\}\}] \\ &= C_0 C^p_1 C^i_2 h^p_k \, [1 + C^i_2 \|P\| C^p_1 + (C^i_2 \|P\| C^p_1)^2 + ...] \\ &= \tfrac{C^p_1 C^i_2}{1 - C^p_1 C^i_2 \|P\|} \, C_0 h^p_k \,. \end{aligned}$$

□

Remark: Notice that in the error estimate we recognise: $C_0 h^p_k$ the truncation error on level k, and C_2 the convergence factor of the iteration cycle. Usually, the mesh ratio $C_1 = 2$ and

$$\|P\| = \sup_H \|P_{hH}\| = \sup_H \sup_{x_H \in E_H} \frac{\|P_{hH} x_H\|_{E_H}}{\|x_H\|_{E_H}} = 1 \,.$$

Remark: We notice that the definition of relative convergence is in terms of the inverse operator

$$\|P_{hH} L^{-1}_H \bar{R}_{Hh} - L^{-1}_h\| \leq C H^p \,.$$

We recognise the equivalence with the assumption in the above theorem by

$$\begin{aligned} & \|(P_{hH}L_H^{-1}\bar{R}_{Hh} - L_h^{-1})y_h\| \leq CH^p\|y_h\| \\ \Longleftrightarrow\ & \|P_{hH}L_H^{-1}\bar{R}_{Hh}y_h - L_h^{-1}y_h\| \leq CH^p\|y_h\| \\ \Longleftrightarrow\ & \|P_{hH}z_H - z_h\| \leq\ C\ H^p\|y_h\| \\ \Longleftrightarrow\ & \|P_{hH}z_H - z_h\| \leq\ C\ H^p\frac{\|\bar{R}_H y\|}{\|y\|}\|y\| \end{aligned}$$

If $\bar{R}_H$ is bounded and stable then $c_1\|y\| \leq \|R_H y\| \leq c_2\|y\|$ and we recognise the equivalence immediately. We can derive the relative convergence of order p between two discretisations also from the convergence of both. This is shown in the following theorem.

Theorem 3.3 Consider the continuous equation $Fz = y$ and let the sequence of discretisations $F_h z_h = y_h$, $h \in \mathcal{H}$ have an order of convergence p. If there exists a nested sequence of stable prolongations

$$\{P_h \mid P_h : E_h \to E,\ h \in \mathcal{H}\}$$

such that $P_h P_{hH} = P_H$ for $h, H \in \mathcal{H}$, $H > h$, then the discretisations $F_h z_h = y_h$ and $F_H z_H = y_H$ are relatively convergent of order p.

Proof:

$$\begin{aligned} \|P_{hH}z_H - z_h\| &\leq \|\hat{\bar{R}}_h P_h(P_{hH}z_H - z_h)\| \\ &\leq \|\hat{\bar{R}}_h\|\ \|P_h P_{hH}z_H - P_h z_h\| \\ &\leq \|\hat{\bar{R}}_h\|\ (\|P_H z_H - z\| + \|z - P_h z_h\|) \\ &\leq C(CH^p + Ch^p) \quad \leq CH^p . \end{aligned}$$

□

Associated with a nested iteration is a particular type of multigrid iteration cycle. This iteration cycle consists of consecutive V- or W-cycles on a sequence of finer and finer grids. This is similar to the initial phase of an FMG process. This iteration cycle is called an *F-cycle*. It can be considered as the repeated execution of the FMG starting phase, first applied to the solution and later on the corrections to the solution.

Chapter 4

Local mode analysis

4.1 Fourier transforms of continuous functions

In this section we collect well-known results with respect to Fourier transforms of functions that are defined (almost everywhere) on domains in the real n-dimensional space. All results mentioned in this section can be found in general texts as e.g. [36], [47], [56], [60].

Let $u \in L^2(\mathbb{R}^n)$, then its Fourier transform $\hat{u}$ is defined by

$$\hat{u}(y) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} e^{-ixy}\, u(x)\, dx\,. \tag{4.1}$$

Furthermore, a back-transformation formula is available:

$$\tilde{u}(x) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} e^{+ixy}\, \hat{u}(y)\, dy\,, \tag{4.2}$$

such that $\tilde{u}(x) = u(x)$ almost everywhere on $\mathbb{R}^n$. Moreover

$$\hat{u} \in L^2(\mathbb{R}^n) \quad \text{and} \quad \|u\|_{L^2(\mathbb{R}^n)} = \|\hat{u}\|_{L^2(\mathbb{R}^n)}\,.$$

We say that the Fourier transformation is a norm-invariant bijection $L^2(\mathbb{R}^n) \to L^2(\mathbb{R}^n)$.

The above definition of a Fourier transformation can be generalised to more general functions than $L^2(\mathbb{R}^n)$-functions. The same definition applies to the set of "tempered distributions" (see e.g. [60]); in this case -again- the back transformation is available.

We see that the Fourier transform (FT) of a function defined on $\mathbb{R}^n$ is a function defined on $\mathbb{R}^n$ itself. A Fourier transformation can also be defined for a finite set of equally spaced data. In this case the FT of a set of N data (the Finite Fourier transform) is again a set of N coefficients (see e.g. [19]).

The FT of a periodic function (or, what is the same, the FT of a function defined on a torus) is a countable infinite set of coefficients. Analogously, in the following sections we shall introduce the Fourier transformation on an infinite set of equally spaced data. In this case the FT of such a "gridfunction" will be periodic function (a function defined on a torus).

Definition 4.1 Let $h = (h_1, ..., h_n) \in \mathbb{R}^n$ be given, then the *h-periodisation* of a function $u : \mathbb{R}^n \to \mathbb{C}$ is defined by

$$\tilde{u}(x) = \sum_{k \in \mathbb{Z}^n} u(x - kh),$$

where $kh = (k_1 h_1, ..., k_n h_n)$.

We notice that $\tilde{u}(x)$ is a periodic function on $\mathbb{R}^n$ with period h; it is completely defined by a mapping $[0,h) \to \mathbb{C}$, where $[0,h)$ is defined by

$$[0,h) = [0,h_1) \times [0,h_2) \times ... \times [0,h_n)\,.$$

The FT of a function $\tilde{u}(x)$ defined on the torus $[0,h)$ is (cf. [36]) a sequence $\{c_k\}_{k\in\mathbb{Z}^n}$ defined by

$$c_k = \frac{1}{h^n(2\pi)^n}\int_0^h e^{-2\pi ikx/h}\tilde{u}(x)dx, \tag{4.3}$$

from which it is clear that $c_k = \hat{u}(\frac{2\pi k}{h})/h^n$. Also the Fourier transform on $[0,h)$ has its back-transformation. From this we see that the knowledge of $\hat{u}(y)$ at only certain equally spaced points is enough to restore a periodisation of the original function u, whereas the complete definition of $\hat{u}(y)$ (almost everywhere on $\mathbb{R}^n$) is needed to find the function $u(x)$ itself.

4.2 Gridfunctions

For a fixed "mesh" $h = (h_1, ..., h_n)$ with $h_i > 0$, $i = 1, 2, ..., n$, the regular infinite n-dimensional grid $\mathbb{Z}_h^n$ is defined by

$$\mathbb{Z}_h^n = \{jh | j\in\mathbb{Z}^n\}\,.$$

Further we introduce the notation

$$j/h = (j_1/h_1, ..., j_n/h_n),$$

$$h^n = h_1 \cdot h_2 \cdot ... \cdot h_n,$$

$$T_h^n = (2\pi/h]^n = (-\pi/h_1, \pi/h_1] \times ... \times (-\pi/h_n, \pi/h_n].$$

We call T_h^n an n-dimensional torus.

Definition 4.2 A complex or a real *gridfunction* is a mapping $\mathbb{Z}_h^n \to \mathbb{C}^d$, respectively $\mathbb{Z}_h^n \to \mathbb{R}^d$, where d is the dimension of the range of the mapping.

Remark: We will restrict ourselves to the scalar real or complex gridfunction $\mathbb{Z}_h^n \to \mathbb{R}$ or $\mathbb{Z}_h^n \to \mathbb{C}$ and, unless stated otherwise, we shall use the word gridfunction for this kind of gridfunction exclusively. It is immediate that, with the usual addition and scalar multiplication, the set of all gridfunctions is a vector space. This vector space we denote by

$$l_h(\mathbb{Z}_h^n)$$

or, shortly, by l_h. For any $p \geq 1$ or $p = \infty$ the space l_h can be provided with a norm

$$\|u_h\|_p = (h^n \sum_{j\in\mathbb{Z}^n} |u_h(jh)|^p)^{1/p}, \quad 1 \leq p < \infty, \tag{4.4}$$

or

$$\|u_h\|_\infty = \sup_{j\in\mathbb{Z}^n} |u_h(jh)|\,. \tag{4.5}$$

For a fixed p, $1 \leq p \leq \infty$, all gridfunctions with a finite norm $\|\cdot\|_p$ form a subspace of $l_h(\mathbb{Z}_h^n)$, which is denoted by

$$l_h^p(\mathbb{Z}_h^n)\,.$$

It is obvious that for any p, $1 \leq p \leq \infty$, $l_h^p(\mathbb{Z}_h^n)$ is a Banach space (cf. [75] p.35). Moreover, for $p = 2$ we know that $l_h^2(\mathbb{Z}_h^n)$ is a Hilbert space with the inner product

$$< u_h, v_h >_h = h^n \sum_{j \in \mathbb{Z}^n} u_h(jh)\overline{v_h(jh)}. \tag{4.6}$$

4.3 The Fourier transform of a gridfunction

Let $u_h : \mathbb{Z}_h^n \to \mathbb{C}$ be a gridfunction. We give the following

Definition 4.3 The Fourier transform $\widehat{u_h} \in L^2(T_h^n)$ of $u_h \in l_h^2(\mathbb{Z}_h^n)$ is a function $T_h^n \to \mathbb{C}$, defined by

$$\widehat{u_h}(\omega) = (\frac{h}{\sqrt{2\pi}})^n \sum_{j \in \mathbb{Z}^n} e^{-ijh\omega} u_h(jh). \tag{4.7}$$

The inverse transformation is given by

$$u_h(jh) = (\frac{1}{\sqrt{2\pi}})^n \int_{\omega \in T_h^n} e^{+ijh\omega} \widehat{u_h}(\omega)\, d\omega. \tag{4.8}$$

Remarks:

- We denote the Fourier transform also by

$$\widehat{u_h} \;=\; \mathbf{FT}(u_h).$$

- $\widehat{u_h}$ can also be considered as a $[2\pi/h]^n$-periodic function $\widehat{u_h} : \mathbb{R}^n \to \mathbb{C}$.

- By the Parseval equality we have

$$\|u_h\|_2 = \|\widehat{u_h}\|_{L^2(T_h^n)}.$$

  Hence the Fourier transformation operator $\mathbf{FT}$: $l_h^2 \to L^2(T_h^n)$ is a unitary operator.

- With the identity $\int_{T_h^n} e^{ih\omega(j-k)} d\omega = (\frac{2\pi}{h})^n \delta_{jk}$ we easily verify the back-transformation formula.

In the back-transformation formula (4.8) we see that any gridfunction $u_h \in l_h^2$ can be considered as a linear combination of gridfunctions $e_{h,\omega}$ of the form

$$e_{h,\omega}(jh) = e^{ijh\omega}, \quad \omega \in T_h^n,\ j \in \mathbb{Z}^n, \tag{4.9}$$

i.e. a periodic gridfunction with period $[\frac{2\pi}{h\omega}]^n$; such $e_{h,\omega}$ is called a *simple mode* . The parameter ω is called the *frequency* of this mode and $\widehat{u_h}(\omega)$ is the *amplitude* of the component (mode) with frequency ω in u_h.

Because $e_{h,\omega} \equiv e_{h,\omega+2\pi k/h}$, for all $k \in \mathbb{Z}^n$, a frequency ω cannot be distinguished from a frequency $\omega + 2\pi k/h$. This phenomenon is called *aliasing* .

It is clear from definition 4.3 that the range of frequencies that can be represented on a fine grid (small h) is larger than the range of those which can be represented on a coarser grid (large h). This explains why smooth gridfunctions (i.e. gridfunctions with relatively small amplitudes for the high frequencies) can be much better represented on a coarser grid than less smooth gridfunctions.

Remarks:

- For functions u_h defined on a bounded domain Ω_h, $\Omega_h = [-Nh, Nh]^n \cap \mathbb{Z}_h^n$, with periodic boundary conditions, a similar definition can be given for the Fourier transform $\widehat{u_h}$. The only difference is that the set of frequencies, T_h^n, available for such gridfunction u_h is discrete:

$$T_h^n = \{(\pi k_1/Nh, ..., \pi k_n/Nh) \mid -N < k_j < N,\ j = 1, ..., n\} \tag{4.10}$$

- Functions on a bounded domain with homogeneous Dirichlet or Neumann boundary conditions can be considered as a special case of problems with periodic boundary conditions. In this case the elementary components in these functions are all of the form

$$e_{h,\omega} \pm e_{h,\omega} = e^{ijh\omega} \pm e^{-ijh\omega}.$$

and the Fourier analysis method gives an exact description of symmetric operators $B_h : l_h(\Omega_h) \to l_h(\Omega_h)$ because for symmetric operators we have $(B_h)_{i,k} = (B_h)_{k,i}$ and hence $\widehat{B_h}(\omega) = \widehat{B_h}(-\omega)$. (See Section 4.5 for a definition of $\widehat{B_h}$).

4.4 The relation between FTs of a function restricted to different grids

In this section we will present two theorems. One gives the relation between the FT of a continuous function defined on $\mathbb{R}^n$ and the FT of its restriction to the grid $\mathbb{Z}_h^n$. The other gives the relation between the FT of a gridfunction and the FT of its canonical restriction to a coarser grid.

Theorem 4.4 Let $u \in L^2(\mathbb{R}^n)$ be a continuous function with FT $\hat{u}$. Its restriction to the grid $\mathbb{Z}_h^n$ is defined by

$$u_h(jh) = (R_h^0 u)(jh) = u(jh),\ j \in \mathbb{Z}^n. \tag{4.11}$$

Then we have the following relation between $\hat{u}$ and $\widehat{u_h}$:

$$\widehat{u_h}(\omega) = \sum_{k \in \mathbb{Z}^n} \hat{u}(\omega + 2\pi k/h), \tag{4.12}$$

i.e. $\widehat{u_h}$ is the $[2\pi/h]^n$-periodisation of $\hat{u}$ (def. 4.1).
Proof:

$$\begin{aligned}
\widehat{u_h}(\omega) &= (\tfrac{h}{\sqrt{2\pi}})^n \textstyle\sum_j e^{-ijh\omega} u(jh) \\
&= (\tfrac{h}{2\pi})^n \textstyle\sum_j e^{-ijh\omega} \sum_k \int_{-\pi/h}^{\pi/h} e^{ijh(y+2\pi k/h)} \hat{u}(y + 2\pi k/h) dy \\
&= (\tfrac{h}{2\pi})^n \textstyle\sum_j e^{-ijh\omega} \int_{-\pi/h}^{\pi/h} e^{ijhy} \sum_k \hat{u}(y + 2\pi k/h) dy
\end{aligned}$$

Using (4.7) and (4.8) we see that this equals

$$\sum_{k \in \mathbb{Z}^n} \hat{u}(\omega + 2\pi k/h).$$

□

Definition 4.5 Let $u_h \in l_h(\mathbb{Z}_h^n)$ then its *canonical q-restriction* $R_q^0 u_h, (q \in \mathbb{Z}^n)$, is the gridfunction u_H defined on $\mathbb{Z}_H^n = \mathbb{Z}_{qh}^n$, defined by

$$(R_q^0 u_h)(jH) = u_H(jH) = u_h(jqh). \tag{4.13}$$

Sometimes we write R_{Hh}^0, with $H = qh$, instead of R_q^0.

Lemma 4.6 If $u_h \in l_h^p(\mathbb{Z}_h^n)$ then $R_q^0 u_h \in l_H^p(\mathbb{Z}_H^n)$ with $H = qh$.
Proof:

$$\begin{aligned} \|u_H\|^p_{l_H^p(\mathbb{Z}_H^n)} &= q^n h^n \textstyle\sum_{j\in\mathbb{Z}^n} |u_h(jqh)|^p \\ &\leq q^n h^n \textstyle\sum_{j\in\mathbb{Z}^n} |u_h(jh)|^p \\ &\leq q^n \|u_h\|^p_{l_h^p(\mathbb{Z}_h^n)} < \infty . \end{aligned}$$

□

Theorem 4.7

$$(\widehat{R_q u_h})(\omega) = \sum_{p\in[0,q)} \widehat{u_h}(\omega + 2\pi p/H), \text{ for all } \omega \in T_H^n, H = qh, q \in \mathbb{Z}^n, q > 0 . \qquad (4.14)$$

Proof: The proof is left as an exercise. □

Theorem 4.7 shows us that, using the restriction R_q^0 with $q \in \mathbb{Z}^n, q = (q_1, ..., q_n)$, we get aliasing of $Q = q_1....q_n$ frequencies onto one.

4.5 Toeplitz operators and their FTs

Let $A : l_h(\mathbb{Z}_h^n)$ be a linear operator. We are interested to know if for any given $\omega \in T_h^n$ the operator A has an eigenvalue λ_ω corresponding with an eigenfunction $e_{h,\omega}$ as defined by (4.9). Suppose it to be true. Then, denoting A by its matrix representation (a_{mj}), $m, j \in \mathbb{Z}^n$, the following holds

$$\sum_{j\in\mathbb{Z}^n} a_{mj} e_{h,\omega}(jh) = \lambda_\omega e_{h,\omega}(mh)$$

$$\sum_{j\in\mathbb{Z}^n} a_{mj} e^{ijh\omega} = \lambda_\omega e^{imh\omega},$$

hence

$$\lambda_\omega = \sum_{j\in\mathbb{Z}^n} a_{mj} e^{i(j-m)h\omega} = \sum_{k\in\mathbb{Z}^n} a_{m,m+k} e^{ikh\omega} .$$

Because λ_ω is independent of m, an a_{-k} should exist such that $a_{m,m+k} = a_{-k}$ for all $m \in \mathbb{Z}^n$. These considerations give rise to the following.

Definition 4.8 A linear operator $A : l_h(\mathbb{Z}_h^n) \to l_h(\mathbb{Z}_h^n)$ whose matrix elements a_{mj} satisfy the relation $a_{m,m+k} = a_{-k}$ for all $m \in \mathbb{Z}^n$ and certain a_{-k}, is called a *Toeplitz operator.*

Lemma 4.9 Let $A : l_h(\mathbb{Z}_h^n) \to l_h(\mathbb{Z}_h^n)$ be a Toeplitz operator with matrix representation (a_{mj}). Then for any $\omega \in T_h^n$

$$\lambda_\omega = \sum_{k\in\mathbb{Z}^n} a_{-k} e^{ikh\omega} \qquad (4.15)$$

is an eigenvalue of A, corresponding with the eigenfunction $e_{h,\omega}$.

Definition 4.10 Let $a_h, u_h \in l_h(\mathbb{Z}_h^n)$ be two gridfunctions. The *a_h-convolution* of u_h, denoted by $a_h \star u_h \in l_h(\mathbb{Z}_h^n)$, is defined by

$$(a_h \star u_h)(mh) = \sum_{j\in\mathbb{Z}^n} a_h((m-j)h)\, u_h(jh) \text{ for all } m \in \mathbb{Z}^n. \qquad (4.16)$$

We can identify the a_h-convolution with a Toeplitz operator A_h on $l_h(\mathbb{Z}_h^n)$ in the following sense:

$$A_h : l_h(\mathbb{Z}_h^n) \to l_h(\mathbb{Z}_h^n) \text{ such that } : A_h u_h = a_h \star u_h. \tag{4.17}$$

I.e. the Toeplitz operator A_h is uniquely identified by the gridfunction $a_h \in l_n(\mathbb{Z}_h^n)$ as an a_h-convolution. The gridfunction a_h is related to the matrix elements of A by the relation

$$a_h(-kh) = a_{-k} \text{ for all } k \in \mathbb{Z}^n. \tag{4.18}$$

By (4.18) follows

$$\begin{aligned} (Au_h)(mh) &= \textstyle\sum_{j\in\mathbb{Z}^n} a_{mj}\, u_h(jh) \\ &= \textstyle\sum_{j\in\mathbb{Z}^n} a_{-(j-m)}\, u_h(jh) \\ &= \textstyle\sum_{j\in\mathbb{Z}^n} a_h((m-j)h)\, u_h(jh) \\ &= (a_h * u_h)(mh), \quad \text{for all } m\in\mathbb{Z}^n,\ u_h \in l_h. \end{aligned} \tag{4.19}$$

We say that the gridfunction a_h *generates* the Toeplitz operator $A = a_h \star$.

Definition 4.11 The *Fourier transform* $\widehat{A_h} : T_h^n \to \mathbb{C}$ of a *Toeplitz operator* A_h is defined by

$$A_h e_{h,\omega} = \widehat{A_h}(\omega) e_{h,\omega}. \tag{4.20}$$

Using (4.15) we see that this means that

$$\widehat{A_h}(\omega) = \lambda_\omega = \sum_{k\in\mathbb{Z}^n} a_h(kh) e^{-ikh\omega}. \tag{4.21}$$

An immediate consequence of this is the following

Lemma 4.12

$$\widehat{A_h u_h}(\omega) = \widehat{A_h}(\omega)\,\widehat{u_h}(\omega). \tag{4.22}$$

Combining (4.21) and definition 4.3 we obtain the following

Lemma 4.13

$$\widehat{A_h}(\omega) = \left(\frac{\sqrt{2\pi}}{h}\right)^n \widehat{a_h}(\omega).$$

Examples of Toeplitz operators are linear difference operators with constant coefficients, defined on the space of gridfunctions, such that the same difference equation is applied at each gridpoint (i.e. the matrix elements of this operator satisfy $a_{m,m+k} = a_{-k}$ for all $m\in\mathbb{Z}^n$). Another example is the translation operator T_{kh}

$$(T_{kh} u_h)(jh) = u_h((j-k)h). \tag{4.23}$$

A typical property of, for example, the above mentioned linear difference operators, is that the difference equation applied to some gridpoint only relates a few neighbouring gridpoints. Examining the definition of a convolution product

$$(a_h \star u_h)(mh) = \sum_{k\in\mathbb{Z}^n} a_h(kh)\ u_h((m-k)h)$$

we see that this means that the generating gridfunction a_h has *finite support:* $a_h(kh) \neq 0$ only for some $k\in\mathbb{Z}^n$ in a neighbourhood of $0\in\mathbb{Z}^n$.

Suppose now that the Toeplitz operator A_h is generated by a gridfunction a_h with finite support, i.e.

$$\begin{cases} a_h(-kh) = s_k & k \in V, \\ a_h(-kh) = 0 & k \notin V, \end{cases}$$

for some *finite* set $V \subset \mathbb{Z}^n$, containing $0 \in \mathbb{Z}^n$. Then we often associate A_h with a *stencil*. For $n = 2$ such a stencil is given by

$$A_h = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & s_{-1,1} & s_{0,1} & s_{1,1} & \cdots \\ \cdots & s_{-1,0} & s_{0,0} & s_{1,0} & \cdots \\ \cdots & s_{-1,-1} & s_{0,-1} & s_{1,-1} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}_h ,$$

where only non-zero s_k are given. This notation (the *stencil notation*) is very convenient if V is small.

We will now give a few detailed examples of Toeplitz operators and their FT.

Example 4.14
Consider a central difference discretisation of the Laplace operator Δ in one dimension on a regular infinite grid $\mathbb{Z}_h^1$. The stencil representation is:

$$L_h = \frac{1}{h^2} [1, -2, 1]_h .$$

We can now, very easily, determine $\widehat{L_h}$, using (4.21):

$$\widehat{L_h}(\omega) = \frac{1}{h^2}(e^{ih\omega} + e^{-ih\omega} - 2) = \frac{1}{h^2}(2\cos(h\omega) - 2) = -\frac{4}{h^2}\sin^2(h\omega/2) . \qquad (4.24)$$

Example 4.15
The translation operator T_{kh} (see (4.23)) can be written as a convolution:

$$(T_{kh}u_h)(mh) = u_h((m-k)h) = \sum_{j \in \mathbb{Z}^n} a_h(jh)\, u_h((m-j)h),$$

where $a_h(kh) = 1$, and $a_h(jh) = 0$ if $k \neq j$. Hence we can identify T_{kh} with the following stencil ($n = 1$):

$$T_{kh} = [1, 0, ..., 0, \underline{0}, 0, ..., 0].$$

with k terms on either side of $\underline{0}$.
Its FT is given by

$$\widehat{T_{kh}}(\omega) = e^{-ikh\omega} = \cos(kh\omega) - i\,\sin(kh\omega) .$$

Because $|\widehat{T_{kh}}(\omega)| = 1$ for all $\omega \in T_h^1$ we see that application of a translation operators to u_h does not affect the absolute value of $\widehat{u_h}(\omega)$.

4.6 Consistency of a discrete operator

For linear differential operators with constant coefficients we can use Fourier analysis to determine the order of consistency of a discretisation of this operator. First we recall

the following, well known property. A linear partial differential operator L of order N with constant coefficients, given by

$$L = \sum_{|\alpha| \leq N} c_\alpha \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}}, \quad (|\alpha| = \alpha_1 + \ldots + \alpha_n),$$

has the property that for any $\xi \in \mathbb{C}^n$ the function $\mathbb{R}^n \to \mathbb{C}: x \to e^{ix\cdot\xi}$ is mapped by L onto a multiple of itself:

$$Le^{ix\cdot\xi} = p(\xi)e^{ix\cdot\xi},$$

with $p(\xi) = \sum_{|\alpha| \leq N} c_\alpha (i\xi)^\alpha$, the characteristic polynomial. Here $x \cdot \xi = \sum_{k=1}^n x_k \xi_k$ and $\eta^\alpha = \eta_1^{\alpha_1} \ldots \eta_n^{\alpha_n}$. The characteristic polynomial p is called the *symbol* of L. In analogy to definition 4.11 we can write

$$\hat{L}(\omega) = p(\omega).$$

Consider a linear partial differential operator L with constant coefficients and its discretisation L_h on some grid $\mathbb{Z}_h^n$. We are interested in the truncation error

$$\tau_h = L_h R_h - \bar{R}_h L, \quad \tau_h : E \to \hat{E}_h$$

R_h and $\bar{R}_h$ are the canonical restriction. For this purpose we consider some arbitrary $\omega \in T_h^n$ and the truncation error of L_h for the function

$$e_\omega(x) = e^{i\omega\cdot x}, \quad e_\omega \in E.$$

We find

$$\begin{aligned} \tau_h(e_\omega)(x) &= (L_h R_h e_\omega - \bar{R}_h L e_\omega)(x) \\ &= L_h e^{i\omega jh} - R_h \hat{L}(\omega) e^{i\omega\cdot x} \\ &= (\widehat{L_h}(\omega) - \hat{L}(\omega)) e^{ijh\omega}, \end{aligned}$$

hence

$$\|\tau_h(e_\omega)\| = |\widehat{L_h}(\omega) - \hat{L}(\omega)|. \tag{4.25}$$

For given L and L_h this expression can be developed in powers of h and the order of consistency of L_h can be determined.

Example 4.16 *Consistency order*
We take $L = \Delta$ (in two dimensions),

$$L_h = \frac{1}{h^2} \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}_h.$$

We find

$$\begin{aligned} \hat{L}(\omega) &= -\omega_1^2 - \omega_2^2, \\ \widehat{L_h}(\omega) &= \tfrac{1}{h^2}[2\cos(h\omega_1) + 2\cos(h\omega_2) - 4] \\ &= -\tfrac{1}{h^2}[4\sin^2(\omega_1 h/2) + 4\sin^2(\omega_2 h/2)]. \end{aligned}$$

Hence

$$\begin{aligned} |\widehat{L_h}(\omega) - \hat{L}(\omega)| &= |\frac{4}{h^2}\sin^2(\omega_1 h/2) + \frac{4}{h^2}\sin^2(\omega_2 h/2) - \omega_1^2 - \omega_2^2| \\ &= \frac{1}{12}h^2 |\omega_1^4 + \omega_2^4| + \mathcal{O}(h^4) \quad \text{for} \quad h \to 0. \end{aligned}$$

We conclude that the order of consistency of L_h is 2.

4.7 The smoothing factor for relaxation methods

Consider a multigrid algorithm where for each coarser mesh the meshwidth is related to finer mesh by $H = 2h$. In Section 4.3 we saw that the range of frequencies that can be represented on the coarser grid is smaller than the range available on the finer grid. For $H = 2h$ the frequencies that cannot be correctly transferred (i.e. without aliasing) to the next coarser grid are all $\omega \in T_h^n \setminus T_{2h}^n$.

If we want the coarse-grid-correction in the multigrid algorithm to be successful, the relaxation steps should damp the amplitudes of the error modes with frequencies $\omega \in T_h^n \setminus T_{2h}^n$. This smoothing property of a given relaxation method with amplification operator M_h^{REL} for the error, is expressed in terms of the so-called *smoothing factor* μ:

$$\mu = \sup_{\omega \in T_h^n \setminus T_{2h}^n} |\widehat{M_h^{REL}}(\omega)| . \tag{4.26}$$

This smoothing factor μ describes how well the high-frequency components in the error are reduced by the relaxation method.

If L_h is a Toeplitz operator and B_h, an approximate inverse of L_h, is a Toeplitz operator as well, then

$$M_h^{REL} = I_h - B_h L_h$$

is a Toeplitz operator, and

$$\widehat{M_h^{REL}}(\omega) = I - \widehat{B_h}(\omega)\widehat{L_h}(\omega) . \tag{4.27}$$

Example 4.17 *computation smoothing factor*
We return to Example 4.14 where we considered

$$L_h = \frac{1}{h^2}[1, -2, 1]_h, \quad \widehat{L_h}(\omega) = \frac{1}{h^2}(2\cos(h\omega) - 2).$$

As a relaxation method we choose damped Jacobi relaxation with parameter α, i.e. we use

$$B_h = \alpha(\mathrm{diag}(L_h))^{-1} = h^2\,[0, \frac{-\alpha}{2}, 0] .$$

We find $\widehat{B_h}(\omega) = \frac{-h^2\alpha}{2}$, hence

$$\widehat{M_h^{REL}}(\omega) = 1 - \widehat{B_h}(\omega)\,\widehat{L_h}(\omega) = 1 - \alpha + \alpha\,\cos(h\omega) .$$

One can show that $\alpha = 2/3$ yields optimal smoothing of this relaxation method for the given problem. For $\alpha = 2/3$ the smoothing factor is $\mu = 1/3$.

4.8 Restrictions and prolongations

In section 4.4 we introduced the canonical restriction R_h^0 and the canonical q-restrictions $R_q^0 : l_h(\mathbb{Z}_h^n) \to l_{qh}(\mathbb{Z}_{qh}^n)$. Now we show that an arbitrary restriction, where for each gridpoint the same stencil is used) can be described in terms of a Toeplitz operator.

Definition 4.18 A homogeneous (*weighted*) *q-restriction* of a gridfunction u_h defined on $\mathbb{Z}_h^n$ to a gridfunction u_H, $H = qh$, defined on $\mathbb{Z}_H^n$, denoted by $R_q(a_h)u_h$, is defined by

$$(R_q(a_h)u_h)(jH) = u_H(jH) = \sum_{k \in \mathbb{Z}^n} a_h(kh)\; u_h((qj - k)h) . \tag{4.28}$$

The gridfunction a_h generates a Toeplitz operator. If we denote this operator by A_h (i.e. $A_h = a_h\star$) the expression (4.28) can be written in the following form:

$$(R_q(a_h)u_h)(jH) = (R_q^0 A_h u_h)(jqh)\,. \tag{4.29}$$

From this we may conclude, using theorem 4.7 and lemma 4.12, that

$$\mathbf{FT}(R_q(a_h)u_h)(\omega) = \sum_{p\in[0,q)} \widehat{A_h}(\omega + 2\pi p/qh)\ \widehat{u_h}(\omega + 2\pi p/qh) \tag{4.30}$$

The frequencies $\omega + 2\pi p/qh$, $p \neq 0$, we call the *higher harmonics* of the frequency ω. The total number of harmonics (ω itself included) is

$$Q = q_1 \cdot\cdot\cdot\cdot\cdot q_n.$$

Let us first consider the case $n = 1, q = 2$. Then we have

$$\mathbf{FT}(R_q(a_h)u_h)(\omega) = \sum_{p=0}^{1} \widehat{A_h}(\omega + \pi p/h)\ \widehat{u_h}(\omega + \pi p/h).$$

Another notation of this is

$$\widehat{u_H}(\omega) = \mathbf{FT}(R_q(a_h)u_h)(\omega) = (\widehat{A_h}(\omega),\ \widehat{A_h}(\omega + \pi/h)) \begin{pmatrix} \widehat{u_h}(\omega) \\ \widehat{u_h}(\omega + \pi/h) \end{pmatrix}$$

$$=:\ \vec{\mathbf{FT}}(R_q(a_h))(\omega) \cdot\ \vec{\mathbf{FT}}(u_h)(\omega), \quad \omega\in T^1_{qh}.$$

For general $n = 1, q > 1, \omega\in T^n_{qh}$, we write also

$$\mathbf{FT}(R_q(a_h)u_h)(\omega) = \vec{\mathbf{FT}}(R_q(a_h))(\omega)\ \ \vec{\mathbf{FT}}(u_h)(\omega), \tag{4.31}$$

where $\vec{\mathbf{FT}}(R_q(a_h))(\omega)$ is a $1 \times Q$-matrix with components $\widehat{A_h}(\omega + 2\pi p/qh)$, $p\in[0,q)$, and $\vec{\mathbf{FT}}(u_h)(\omega)$ a vector with Q components $\widehat{u_h}(\omega + 2\pi p/qh)$, $p\in[0,q)$.

Example 4.19 $(n = 1, q = 2)$

$$A_h = a_h\star = [1/4, 1/2, 1/4]_h,$$
$$\widehat{A_h}(\omega) = \tfrac{1}{2} + \tfrac{1}{2}\cos(\omega h) = \cos^2(\omega h/2),$$
$$\widehat{A_h}(\omega + \pi/h) = \sin^2(\omega h/2),$$
$$\mathbf{FT}(R_q(a_h))(\omega) = (\cos^2(\omega h/2)\,,\, \sin^2(\omega h/2))\cdot$$

Example 4.20 $(n = 2, q = 2)$

$$A_h = a_h\star = \begin{bmatrix} & 1/8 & \\ 1/8 & 1/2 & 1/8 \\ & 1/8 & \end{bmatrix}_h,$$

Here $\widehat{A_h}(\omega) = \frac{1}{2}\cos^2(\omega_1 h/2) + \frac{1}{2}\cos^2(\omega_2 h/2)$, and analogously we find $\widehat{A_h}(\omega_1, \omega_2 + \pi/h)$, $\widehat{A_h}(\omega_1 + \pi/h, \omega_2)$ and $\widehat{A_h}(\omega_1 + \pi/h, \omega_2 + \pi/h)$. Thus, we find

$$\mathbf{FT}(R_q(a_h))(\omega) = \frac{1}{2}\begin{pmatrix} \cos^2(\omega_1 h/2) + \cos^2(\omega_2 h/2) \\ \cos^2(\omega_1 h/2) + \sin^2(\omega_2 h/2) \\ \sin^2(\omega_1 h/2) + \cos^2(\omega_2 h/2) \\ \sin^2(\omega_1 h/2) + \sin^2(\omega_2 h/2) \end{pmatrix}.$$

Definition 4.21 Let $u_H \in l_H(\mathbb{Z}_H^n)$ be a gridfunction defined on $\mathbb{Z}_H^n$, then its *flat q-prolongation* $P_q^0 u_H$, $q \in \mathbb{Z}^n$, is the gridfunction u_h, $h = H/q$, $u_h \in l_h(\mathbb{Z}_h^n)$, defined by

$$u_h(jh) = (P_q^0 u_H)(jh) = \begin{cases} u_H(kqh), & \text{if } j = kq, \\ 0 & \text{otherwise.} \end{cases} \tag{4.32}$$

One can easily verify the following

Theorem 4.22

$$\mathbf{FT}(P_q^0 u_H)(\omega) = \widehat{u_h}(\omega) = q^{-n}\widehat{u_H}(\omega),\ \omega \in T_h^n. \tag{4.33}$$

Proof: The proof is left as en exercise. □

The interpretation of this theorem is that $\widehat{u_h}(\omega)$ is the periodic continuation of $\widehat{u_H}(\omega)$, except for the factor

$$q^{-n} = Q^{-1}.$$

Analogous to definition 4.18 any homogeneous q-prolongation can be described in terms of a Toeplitz operator $A_h = a_h \star$:

$$P_q(a_h) = A_h P_q^0. \tag{4.34}$$

Example 4.23 *Linear interpolation in one dimension ($n = 1, q = 2$)*

$$A_h = a_h \star = [\tfrac{1}{2}, 1, \tfrac{1}{2}]_h,$$

$$\widehat{u_h}(\omega) = \mathbf{FT}(A_h P_q^0 u_H)(\omega) = \widehat{A_h}(\omega)\mathbf{FT}(P_q^0 u_H)(\omega)$$

$$= q^{-1}\,\widehat{A_h}(\omega)\widehat{u_H}(\omega) = \tfrac{1}{2}(1 + \cos(\omega h))\widehat{u_H}(\omega) = cos^2(\omega h/2)\widehat{u_H}(\omega), \quad \omega \in T_h^1\,.$$

(Notice that for $\widehat{u_H}$, originally defined on T_H^1, now its periodic continuation on T_h^1 is used.) We can write this also as

$$\vec{\mathbf{FT}}(u_h) = \begin{pmatrix} \cos^2(\omega h/2) \\ \sin^2(\omega h/2) \end{pmatrix} \widehat{u_H}\,, \quad \omega \in T_H^1.$$

Analogous to (4.31) we introduce the notation

$$\vec{\mathbf{FT}}(u_h)(\omega) = \ \mathbf{FT}(P_q(a_h))(\omega)\ \ \mathbf{FT}(u_H)(\omega)\,. \tag{4.35}$$

Here $\vec{\mathbf{FT}}(P_q(a_h))$ is a $Q \times 1$- matrix, given by

$$\vec{\mathbf{FT}}(u_h)(\omega) = \ \mathbf{FT}(P_q(a_h)(\omega) = \frac{1}{Q}\begin{pmatrix} \widehat{A_h}(\omega) \\ \vdots \\ \widehat{A_h}(\omega + 2\pi/q_h) \\ \vdots \end{pmatrix}, \quad \omega \in T_H^n,$$

and $\vec{\mathbf{FT}}(u_h)(\omega)$ a Q-vector.

4.9 The order of a restriction or a prolongation

In the previous section we saw that a restriction or a prolongation could be associated with a gridfunction a_h, an operator A_h or its FT $\widehat{A_h}(\omega)$. Such a FT is a trigonometric polynomial in $\theta = \omega h$. If there is no possibility of confusion in this section we write $A(\theta)$ instead of $\widehat{A_h}(\omega)$ (for restrictions) or $\frac{1}{Q}\widehat{A_h}(\omega)$ for prolongations.

Definition 4.24 The *(primary) order* or *low-frequency order* of a prolongation $P_q(a_h)$ or of a restriction $R_q(a_h)$, is the largest number $m \geq 0$ for which

$$A(\theta) = 1 + \mathcal{O}(|\theta|^m), \quad \text{for } |\theta| \to 0\,.$$

Definition 4.25 For a given grid-coarsening factor $q \in \mathbb{Z}^n$, $H = qh$, the *secondary order* or *high-frequency order* of a prolongation $P_q(a_h)$ or of a restriction $R_q(a_h)$, is the largest number $m \geq 0$ for which

$$A(\theta + 2\pi p/q) = \mathcal{O}(|\theta|^m)\ (|\theta| \to 0),$$

for all $p \in [0, q)^n \subset \mathbb{Z}^n, p \neq 0^n$.

The use of these two definitions will become clear in section 4.11 where primary and secondary orders play a role in the convergence of multigrid methods. We will now give a few examples of the computation of primary and secondary orders of some prolongations and restrictions. In all examples we assume $q = 2$.

Example 4.26 *Canonical q-restriction* $R_q^0, n = 1$

$$A_h = a_h\star = [0, 1, 0]_h,$$

$A(\theta) = 1$, hence the primary order is ∞ and the secondary order is 0.

Example 4.27 *Linear interpolation,* $n = 1$

$$A_h = a_h\star = [1/2, 1, 1/2]_h,$$

$$\frac{1}{Q}\widehat{A_h}(\omega) = \cos^2(\omega h/2) \quad \text{(see example 4.23)},$$

$$A(\theta) = \cos^2(\theta/2) = 1 - \tfrac{1}{4}\theta^2 + \mathcal{O}(\theta^4) \quad \text{for} \quad \theta \to 0\,,$$

hence the primary order is also 2.

$$A(\theta + \pi) = \cos^2(\frac{\theta + \pi}{2}) = \sin^2(\theta/2) = \tfrac{1}{4}\theta^2 + \mathcal{O}(\theta^4) \quad \text{for} \quad \theta \to 0,$$

hence the secondary order is also 2.

Example 4.28 *Linear interpolation,* $n = 2$

$$A_h = a_h\star = \begin{bmatrix} 1/2 & 1/2 & \\ 1/2 & 1 & 1/2 \\ & 1/2 & 1/2 \end{bmatrix}_h.$$

We write $h\omega = (\phi, \theta)$ and find

$$\begin{aligned} 4A(\phi,\theta) = \; & 1 + \tfrac{1}{2}e^{i\phi} + \tfrac{1}{2}e^{-i\phi} + \tfrac{1}{2}e^{i\theta} + \tfrac{1}{2}e^{-i\theta} + + \tfrac{1}{2}e^{i(\phi-\theta)} + \tfrac{1}{2}e^{-i(\phi-\theta)} \\ & = 4\cos(\theta/2)\cos(\phi/2)\cos((\phi-\theta)/2). \end{aligned}$$

Expanding $A(\phi, \theta)$ in a Taylor series yields

$$\begin{aligned} A(\phi,\theta) \; & = (1 - \tfrac{1}{2}(\tfrac{\theta}{2})^2 + \mathcal{O}(|\theta|^4)) \cdot (1 - \tfrac{1}{2}(\tfrac{\phi}{2})^2 + \mathcal{O}(|\phi|^4)) \cdot (1 - \tfrac{1}{2}(\tfrac{\phi-\theta}{2})^2 + \mathcal{O}(|\phi-\theta|^4)) \\ & = 1 - \tfrac{1}{4}(\theta^2 + \phi^2 - \phi\theta) + \mathcal{O}((\phi,\theta)^4), \quad \text{for} \quad (\phi,\theta) \to 0\,. \end{aligned}$$

We conclude that the primary order of linear interpolation in two dimensions is 2. For the higher harmonics we find

$$A(\pi - \phi, \theta) = \tfrac{1}{4}(\theta\phi - \phi^2) + \mathcal{O}((\phi,\theta)^4), \quad \text{for} \quad (\phi,\theta) \to 0,$$

and similar expressions for $A(\phi, \pi - \theta)$ and $A(\pi - \phi, \pi - \theta)$. We conclude that the secondary order is 2.

Remark: The same holds for the linear weighted restriction (7-point restriction).

Example 4.29 *Bilinear interpolation, $n = 2$*

$$A_h = a_h\star = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}_h .$$

$$\begin{aligned} 4A(\phi,\theta) \; & = 1 + \tfrac{1}{2}e^{i\phi} + \tfrac{1}{2}e^{-i\phi} + \tfrac{1}{2}e^{i\theta} + \tfrac{1}{2}e^{-i\theta} + \\ & + \quad \tfrac{1}{4}e^{i(\phi-\theta)} + \tfrac{1}{4}e^{-i(\phi-\theta)} + \tfrac{1}{4}e^{i(\phi+\theta)} + \tfrac{1}{4}e^{-i(\phi+\theta)} \\ & = 4\cos^2(\phi/2)\cos^2(\theta/2)\,. \end{aligned}$$

Expanding $A(\phi, \theta)$ in a Taylor series we find

$$\begin{aligned} A(\phi,\theta) & = (1 - (\frac{\phi}{2})^2 + \mathcal{O}(|\phi|^4))(1 - (\frac{\theta}{2})^2 + \mathcal{O}(|\theta|^4)) \\ & = 1 - \tfrac{1}{4}(\phi^2 + \theta^2) + \mathcal{O}((\phi,\theta)^4) \quad \text{for} \quad (\phi,\theta) \to 0\,. \end{aligned}$$

The primary order of bilinear interpolation is 2. Similarly we find that the secondary order is also 2.

Remark: The same holds for the full-weighting (FW) restriction (9-point restriction).

Example 4.30 *Cubic interpolation, $n = 1$*

$$A_h = a_h\star = \left[-\frac{1}{16}, 0, \frac{9}{16}, 1, \frac{9}{16}, 0, -\frac{1}{16}\right]_h .$$

$$\begin{aligned} A(\phi) \; & = \tfrac{1}{Q}\widehat{A_h}(\phi/h) = \tfrac{1}{2}\widehat{A_h}(\phi/h) \\ & = \tfrac{1}{16}[8 + 9\cos\phi - \cos 3\phi] \\ & = (1 - \tfrac{1}{2}(\phi/2)^2 + ...)^2[1 + (1 + ...)((\phi/2)^2 + ...)] \\ & = 1 + \mathcal{O}(|\phi|^4) \quad \text{for} \quad \phi \to 0\,. \end{aligned}$$

The primary order of cubic interpolation in 1-D is 4. Similar considerations for $A(\pi - \phi)$ yield secondary order 4.

Remark: One can show that also for cubic interpolation in two dimensions, both primary and secondary order are 4.

Example 4.31 *Half-weighting, $n = 2$*

$$A_h = a_h\star = \begin{bmatrix} & 1/8 & \\ 1/8 & 1/2 & 1/8 \\ & 1/8 & \end{bmatrix}_h .$$

$$\begin{aligned} \widehat{A_h}(\phi/h, \theta/h) &= \tfrac{1}{2} + \tfrac{1}{8}e^{i\phi} + \tfrac{1}{8}e^{-i\phi} + \tfrac{1}{8}e^{i\theta} + \tfrac{1}{8}e^{-i\theta} \\ &= \tfrac{1}{2} + \tfrac{1}{4}\cos\phi + \tfrac{1}{4}\cos\theta \\ A(\phi,\theta) &= \tfrac{1}{Q}\widehat{A_h}(\phi/h, \theta/h) \\ &= \tfrac{1}{4}\widehat{A_h}(\phi/h, \theta/h) \\ &= \tfrac{1}{4} - \tfrac{1}{64}(\phi^2 + \theta^2) + \mathcal{O}((\phi,\theta)^4) \quad \text{for} \quad (\phi,\theta) \to 0, \end{aligned}$$

hence the primary order is 2.

$$\begin{aligned} A(\pi - \phi, \theta) &= \tfrac{1}{8}(\sin^2(\phi/2) + \cos^2(\theta/2)) \\ &= \tfrac{1}{8}(\tfrac{1}{2}(\phi/2)^2 + 1 - \tfrac{1}{2})^2 + \mathcal{O}((\phi,\theta)^4) \\ &= \tfrac{1}{8} + \tfrac{1}{64}(\phi^2 - \theta^2) + \mathcal{O}((\phi,\theta)^4) \quad \text{for} \quad (\phi,\theta) \to 0, \end{aligned}$$

hence the secondary order is 0.

4.10 Fourier analysis in the case of mutual influencing frequencies

In Section 4.8 we saw that $\widehat{u_h}(\omega)$, $\omega \in T_h^n$ could also be denoted by a Q-vector $\vec{\mathbf{FT}}(u_h)(\omega)$, $\omega \in T_H^n$ by taking the harmonic frequencies together

$$\vec{\mathbf{FT}}(u_h)(\omega) = \begin{pmatrix} \widehat{u_h}(\omega) \\ \vdots \\ \widehat{u_h}(\omega + 2\pi p/qh) \\ \vdots \end{pmatrix}, \quad \omega \in T_H^n = T_{qh}^n,\ p \in [0, q). \tag{4.36}$$

Consistent with this notation we also introduce the notation

$$\vec{\mathbf{FT}}(A_h)(\omega) = \text{ diag } (\widehat{A_h}(\omega + 2\pi p/qh)), \quad \omega \in T_{qh}^n,\ p \in [0, q) \tag{4.37}$$

to replace $\widehat{A_h}(\omega)$, $\omega \in T_h^n$.

Remark: $\widehat{A_h}(\omega)\,\widehat{u_h}(\omega)$, $\omega \in T_h^n$, is now denoted as $\vec{\mathbf{FT}}(A_h)(\omega)\ \vec{\mathbf{FT}}(u_h)(\omega)$, $\omega \in T_H^n$.

The above notation enables us to apply the usual Fourier analysis in the case of mutual influencing of harmonic frequencies. For example, let A_H be a Galerkin-approximation of A_h, i.e.

$$A_H = R_{Hh} A_h P_{hH}, \tag{4.38}$$

where R_{Hh} and P_{hH} are homogeneous restrictions and prolongations, and A_h a Toeplitz operator. We find

$$\mathbf{FT}(A_H)(\omega) = \vec{\mathbf{FT}}(R_{Hh})(\omega)\ \vec{\mathbf{FT}}(A_h)(\omega)\ \vec{\mathbf{FT}}(P_{hH})(\omega), \tag{4.39}$$

where $\vec{\mathbf{FT}}(R_{Hh})(\omega)$ is the $(1 \times Q)$-matrix and $\vec{\mathbf{FT}}(P_{hH})(\omega)$ the $(Q \times 1)$- matrix (cf. (4.31) and (4.35)).

Similarly we find for the coarse-grid-correction (CGC) operator

$$M_h^{CGC} = I_h - P_{hH} L_H^{-1} R_{Hh} L_h \tag{4.40}$$

the following FT:

$$\vec{\mathbf{FT}}(M_h^{CGC})(\omega) = I - \vec{\mathbf{FT}}(P_{hH})(\omega)\ \mathbf{FT}(L_H^{-1})(\omega)\ \vec{\mathbf{FT}}(R_{Hh})(\omega)\ \vec{\mathbf{FT}}(L_h)(\omega),\ \omega \in T_H^n\,. \tag{4.41}$$

For $n = 2, q = 2$ (hence $Q = 4$) (4.41) is a (4×4)- matrix:

Because $\|\widehat{u_h}\|_{L^2(T_h^n)} = \|u_h\|_{l_h^2(\mathbb{Z}_h^n)}$, we have the following relation for the norms of the amplification operators:

$$\begin{aligned}
\|M_h\|_{2\leftarrow 2} &= \sup_{u_h \in l_h^2(\mathbb{Z}_h^n)} \frac{\|M_h u_h\|_2}{\|u_h\|_2} = \\
&= \sup_{u_h} \frac{\|\ \vec{\mathbf{FT}}(M_h u_h)\|_{L^2}}{\|\ \vec{\mathbf{FT}}(u_h)\|_{L^2}} = \sup_{\widehat{u_h}} \frac{\|\ \vec{\mathbf{FT}}(M_h)(\omega)\ \widehat{u_h}(\omega)\|_{L^2(T_h^n)}}{\|\widehat{u_h}(\omega)\|_{L^2(T_h^n)}} \\
&\leq \sup_{\omega \in T_H^n} \|\ \vec{\mathbf{FT}}(M_h)(\omega)\|_2\,.
\end{aligned}$$

Since there is no coupling between non-harmonic-related frequencies we also have

$$\rho(M_h) = \sup_{\omega \in T_H^n} |\lambda_{\max}(\ \vec{\mathbf{FT}}(M_h)(\omega))|\,. \tag{4.42}$$

4.11 Requirements for transfer operators

Let L be an ordinary differential operator of order M:

$$L = \sum_{m=0}^{M} c_m \frac{d^m}{dx^m}\cdot \tag{4.43}$$

Its FT or symbol is then given by

$$\widehat{L}(\omega) = \sum_{m=0}^{M} c_m (i\omega)^m, \tag{4.44}$$

(Section 4.6). Let L_h and L_H be two discretisations of L of order $\tilde{p}$ on $\mathbb{Z}_h^n$ and $\mathbb{Z}_H^n$ respectively. Then $\widehat{L_h}(\omega)$ is a trigonometric polynomial of degree M in ω:

$$\widehat{L_h}(\omega) = \sum_{m=0}^{M} \frac{c_m}{h^m} S_m(h\omega), \tag{4.45}$$

where S_m is the FT of the m^{th} order part of the discrete operator L_h. Furthermore, we have

$$|\widehat{L_h}(\omega) - \widehat{L}(\omega)| = \mathcal{O}(h^{\tilde{p}}) \text{ for } h \to 0\,, \tag{4.46}$$

(see (4.25)) and

$$\begin{aligned}
S_m(\omega h) &\to (\omega h)^m, \quad \text{for} \quad \omega h \to 0, \\
S_m(\omega h) &= \mathcal{O}(1), \quad (-\pi \leq \omega h \leq \pi).
\end{aligned}$$

We consider a fixed ω and $h \to 0$ and we define $\theta = \omega h$.

Using the above we find:

$$\left\| \frac{\widehat{L_h}(\omega)}{\widehat{L_H}(\omega)} \right\| = \left\| \frac{\hat{L}(\omega) + \mathcal{O}(h^{\tilde{p}})}{\hat{L}(\omega) + \mathcal{O}(H^{\tilde{p}})} \right\| = 1 + \mathcal{O}(h^{\tilde{p}})(h \to 0),$$

and

$$\left\| \frac{\widehat{L_h}((\omega + 2\pi p/qh)}{\sum_{m=0}^{M} \frac{c_m}{(qh)^m}(\omega qh)^m} \right\| = \frac{\mathcal{O}(h^{-M})}{\mathcal{O}(h^{-M}(\omega h)^M)} = \mathcal{O}(h^{-M}) \text{ for } h \to 0) .$$

Now we are interested in the behaviour of the CGC-amplification factors for harmonic frequencies in a neighbourhood of the origin. For simplicity we restrict ourselves to the case $n = 1, q = 2$. The amplification operator for the error in one CGC-step is given by (4.40), its FT, $\widehat{M_h^{CGC}}(\omega)$, is given in (4.41) and can now be denoted as

$$\widehat{M}(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} \hat{P}(\omega) \\ \hat{P}(\omega + \pi/h) \end{pmatrix} \widehat{L_H^{-1}}(\omega)(\hat{R}(\omega), \hat{R}(\omega+\pi/h)) \begin{pmatrix} \widehat{L_h}(\omega) & 0 \\ 0 & \widehat{L_h}(\omega + \pi/h) \end{pmatrix}.$$

If we assume that the primary and secondary orders of the prolongation (restriction) are given by m_1 and m_2 (n_1 and n_2), then we find for $\theta \to 0$:

$$\begin{aligned} \widehat{M}(\theta) &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 + \mathcal{O}(\theta^{m_1}) \\ \mathcal{O}(\theta^{m_2}) \end{pmatrix} (1 + \mathcal{O}(\theta^{n_1}), \mathcal{O}(\theta^{n_2})) \begin{pmatrix} 1 + \mathcal{O}(\theta^{\hat{p}}) & 0 \\ 0 & \mathcal{O}(\theta^{-M}) \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{O}(\theta^{m_1}) + \mathcal{O}(\theta^{n_1}) + \mathcal{O}(\theta^{\hat{p}}) & \mathcal{O}(\theta^{n_2-M}) \\ \mathcal{O}(\theta^{m_2}) & 1 + \mathcal{O}(\theta^{n_2+m_2-M}) \end{pmatrix}. \end{aligned}$$

It is clear that convergence requires $m_1 > 0, n_1 > 0$ and $\tilde{p} > 0$. Further, for $\theta \to 0$ the eigenvalues of $M(\theta)$ are

$$\lambda_1 = 1 + \ldots, \text{ and } \lambda_2 = \mathcal{O}(\theta^{n_2+m_2-M}).$$

For convergence of the CGC-process $n_2 + m_2 \geq M$ is required. Further, $n_2 \geq M$, $m_2 \geq 0$ is required for $\|M(\theta)\|$ to be bounded. Similarly we derive the requirements $m_2 \geq M$, $n_2 \geq 0$ for $\|\overline{M}(\theta)\|$ to be bounded.

Chapter 5

Multigrid approaches for compressible flow (with B. Koren)

5.1 The equations of compressible flow

The efficient solution of flow problems is one of the earliest aims for multigrid methods [7]. Most progress in the development of multigrid has, however, been made in the field of elliptic partial differential equations. For the more complex equations that describe flow problems, the development of multigrid was hanging back. Early work was done by Brandt [64, 9, 8] for the Stokes equations and both the incompressible and compressible Navier-Stokes equations.

Still many more attempts were made to apply multigrid ideas to improve the efficiency of flow computations. Assuming the absence of rotation, flows are described by the potential equation, which -in the interesting case of transonic flow- is of mixed hyperbolic and elliptic type. By the use of multigrid, substantial improvements were made in the solution procedures for these equations [64, 4, 26, 48, 10, 51].

In Section 5.3 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 problems in two space dimensions. Only recently attempts are made to apply multigrid methods to problems in three space dimensions. First we make some brief remarks on the equations in two dimensions and their discretisations.

5.1.1 The Navier-Stokes equations

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

$$\frac{\partial}{\partial t}\, q + \frac{\partial}{\partial x}\, F(q) + \frac{\partial}{\partial y}\, G(q) = 0, \tag{5.1}$$

where

$$F(q) = f(q) - \mathrm{Re}^{-1}\, r(q), \qquad G(q) = g(q) - \mathrm{Re}^{-1}\, s(q), \tag{5.2}$$

and

$$q = (\rho, \rho u, \rho v, \rho e)^T,$$
$$f = (\rho u, \rho u^2 + p, \rho uv, \rho uh)^T,$$

$$g = (\rho v, \rho v u, \rho v^2 + p, \rho v h)^T,$$

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

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

Here ρ , 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

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

γ 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. Re and Pr denote the Reynolds and Prandtl number; thermal conductivity is given by κ; $c = \sqrt{\gamma p/\rho}$ is the local speed of sound; and

$$\tau_{xx} = (\lambda + 2\mu)\,\partial u/\partial x + \lambda\,\partial v/\partial y,$$

$$\tau_{xy} = \mu\,(\partial u/\partial x + \partial v/\partial x),$$

$$\tau_{yy} = (\lambda + 2\mu)\,\partial v/\partial y + \lambda\,\partial u/\partial x,$$

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

5.1.2 The Euler equations

The *Euler equations* are obtained from (5.1),(5.2) by neglecting viscous and heat conduction effects in (5.2); then

$$F(q) = f(q), \qquad G(q) = g(q). \tag{5.3}$$

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

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

the matrix

$$k_1 A + k_2 B = k_1 \frac{\partial f}{\partial q} + k_2\,\frac{\partial g}{\partial q} \tag{5.4}$$

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, (5.1) is rewritten in its integral form

$$\frac{\partial}{\partial t}\int\int_{\Omega} q\,dx\,dy + \int_{\partial\Omega}(F\,n_x + G\,n_y)\,ds \;=\; 0, \qquad \text{for all } \Omega \subset \Omega^\star, \tag{5.5}$$

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

The form (5.5) of equation (5.1) shows clearly the character of the system of conservation laws: the increase of q in Ω can be caused only by the inflow of q over $\partial\Omega$. In symbolic form (5.5) is written as

$$q_t + N(q) = 0. \tag{5.6}$$

The solution of the weak form (5.5) of (5.1), (5.3) 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. [45, 46]).

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:

$$N(q) = 0. \tag{5.7}$$

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

5.2 The discretisations

For the discretisation of (5.1) or (5.5), 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

$$q_h^{(n+1)} = q_h^{(n)} + \Delta t\ (q_h)_t + \frac{1}{2}(\Delta t)^2\ (q_h)_{tt}\ . \tag{5.8}$$

With the equations (5.1), (5.3) we arrive at

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

where A and B are defined by (5.4). 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 Ω^* is not rectangular, a 1-1-mapping $(x, y) \leftrightarrow (\xi, \eta)$ 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 $q_{ij}^{(n+1)} = q_{ij}^{(n)}$, the discrete steady state still depends on Δt. This is caused by the fact that the discrete term with $(\Delta t)^2$ in (5.8) 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 (5.6), by which it is reduced to the large system of ordinary differential equations (ODEs),

$$\frac{\partial}{\partial t}\ q_h = N_h(q_h)\ . \tag{5.9}$$

Now, to find an approximation of the time-dependent solution of (5.6), any method can be used for the integration of this system of ODEs. The solution of the steady state can

be computed by solving (5.9) until all transients have died out. Alternatively, we can avoid the ODEs (5.9) and solve the nonlinear system

$$N_h(q_h) = 0 \tag{5.10}$$

by other (more direct) means. In both cases (5.9) and (5.10), 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 (5.9) or (5.10) on a non-rectangular domain $\Omega^\star$, 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 (5.9) on a non-rectangular grid is by a *finite volume* technique. Here, the starting point for the discretisation is (5.5). Without an *a-priori* transformation, the domain $\Omega^\star$ is divided into a set of disjoint (quadrilateral) cells Ω_{ij}. The discrete representation q_h of q is given by the values q_{ij}, the (mean) values of q in the cell Ω_{ij}. Using different approximations for the computation of fluxes between the cells Ω_{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 $h \to 0$ for a non-rectangular grid, a formal relation between the vertices of cells Ω_{ij} and a regular grid can be given, again by a mapping $(x, y) \leftrightarrow (\xi, \eta)$. If this mapping is smooth enough, it can be proved that for refinements $h \to 0$ which correspond with regular refinements in (ξ, η), space discretisations up to second order can be obtained by finite volumes. An advantage of the finite volume technique is that the un-transformed 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 $\Omega_{i,j}$.

(1) For a central difference type, the flux over a cell wall Γ_{LR} between two cells with states q_L and q_R is computed as $f^\star(\frac{1}{2}(q_L + q_R))$, where $f^\star = k_1 f + k_2 g$ is the flux normal to Γ_{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* $f^\star(q_L, q_R)$ are introduced to compute the flux over Γ_{LR}. Several functions $f^\star$ 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 [72], Roe [59], Osher [52, 55] and others. In Section 6.2 we give a description of Osher's scheme, for further descriptions we refer to the literature mentioned. For a consistent scheme, $f^\star(q, q) = f^\star(q)$, i.e. the numerical flux function with equal arguments conforms with the genuine flux function in (5.3). All these upwind flux-functions have in common that they are purely one-sided if all characteristics point into the same direction, i.e. $f^\star(q_L, q_R) = f^\star(q_L)$ if the flow of all information is from left to right. More details are given in Section 6.

5.3 The multiple grid methods

When a multiple grid technique is used to solve the system of nonlinear (differential) equations (5.9) or (5.10), we assume the existence of a nested set of grids. Usually this nesting is such that a set of 2×2 cells in a fine mesh forms 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 (5.9) or (5.10) to

$$\frac{\partial}{\partial t} q_h = N_h(q_h) - r_h \tag{5.11}$$

or

$$N_h(q_h) = r_h, \tag{5.12}$$

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

$$q_h^{(n+1)} \leftarrow \mathcal{G}^h\,(q_h^{(n)}, r_h). \tag{5.13}$$

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

$$N_{2h}(q_{2h}) = N_{2h}(q_{2h}^{(0)}) - R(N_h(q_h^{(n)} - r_h), \tag{5.14}$$

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

$$q_h^{(k+1)} = q_h^{(k)} + P_{h,2h}(q_{2h} - q_{2h}^{(0)})\ . \tag{5.15}$$

The combination of (5.14) and (5.15) is the *coarse grid correction* (CGC) step. The solution q_{2h} of (5.14) 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 [50, 34].

As we saw in Chapter 2, a multigrid FAS cycle for the solution of (5.12) 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 (5.14) is effected by σ 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 $N_h(q_h) = r_h$.

5.3.1 Methods based on Lax-Wendroff type time stepping

A paper by Ni [50] was among the first to apply a multigrid 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:

$$\begin{aligned} \Delta q_{i+\frac{1}{2},j+1/2} &= -\frac{1}{2}\frac{\Delta t}{\Delta x}[(F_{i+1,j}-F_{i,j})+(F_{i+1,j+1}-F_{i,j+1})] \\ &\quad -\frac{1}{2}\frac{\Delta t}{\Delta y}[(G_{i,j+1}-G_{i,j})+(G_{i+1,j}-G_{i+1,j+1})], \\ F_{i,j} &= F(q_{i,j}^{(n)}) \quad \text{etc..} \end{aligned} \tag{5.16}$$

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

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

$$q_{ij}^{(n+1)} = q_{ij}^{(n)} + \Delta q_{ij}\ . \tag{5.17}$$

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 (5.16)-5.17) is described as:

$$\text{compute} \quad \Delta q_h^{cell}, \tag{5.18}$$

with cell values

$$\Delta q_i + 1/2, j+1/2 \approx -\Delta t \int_{\partial\Omega_i+1/2,j+1/2}(f\,n_x + g\,n_y)\,ds\,/(\Delta x\,\Delta y);$$

$$q_h^{(n+1)} := q_h^{(n)} + D_h\,\Delta q_h^{cell}\ . \tag{5.19}$$

The operator D_h is the distribution operator that transfers the cell centered corrections to the grid points by means of (5.17).

The coarse grid acceleration as introduced in [50] by Ni deviates from the usual coarse grid scheme (5.14)-(5.15). In [50] 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 Δq_h to the $2h$-grid. Then the corrections $\Delta q_{2h}^{\text{cell}}$ are distributed to the coarser mesh points similar to (5.17), and the coarse grid correction is interpolated to the fine grid. Thus, here the coarse grid correction reads

$$\Delta q_{2h}^{cell} := R_{2h,h}\,\Delta q_h^{cell}, \tag{5.20}$$

$$q_h^{(n+1)} := q_h^{(n)} + P_{h,2h}D_{2h}\,\Delta q_{2h}^{cell}\ . \tag{5.21}$$

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 (5.10).

In the same way the correction procedure can be repeated on progressively coarser grids. Therefore, in (5.20), $2h$ should be replaced by $2^m h$. We notice that, in contrast with the usual multigrid 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, $m = 1, ..., L$, in parallel and to form the correction

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

at once [71]. When optimal use of modern multi-processor computers is to be made, it is also possible to perform both computations (5.18) and (5.20) in parallel

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 (5.18). In [11, 33]. Johnson applies the popular MacCormack scheme. Further, in (5.20) 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 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 Δt 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 (5.15). 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.

5.3.2 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 (5.6) 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. [62]). 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 $\Omega^\star$; (ii) enthalpy damping, where a-priori knowledge about the behaviour of the enthalpy over $\Omega^\star$ is used (e.g. h constant over $\Omega^\star$); (iii) residual smoothing, (iv) implicit residual averaging, and (v) implicit corrected viscosity acceleration [16] 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 [29]. 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 $R_{2h,h}$ ($\overline{R}_{2h,h}$) 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 Ω^* 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 [32] Jespersen announces an interesting theorem on the use of the multigrid 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.

$$N_h(R_h(q)) - RN(q) = \mathcal{O}(h),$$

and let the time-stepping procedure be consistent in time

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

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 multigrid algorithm on L grids is a consistent, first-order in time, discretisation of (5.6) with time step $\Delta t_{\text{tot}} = \sum_{j=1,\ldots,L} \Delta t_j$.

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 [32] 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.

5.3.3 Fully implicit methods

Most methods so far developed are based on the concept of integrating the equations (5.6) in time until a steady state is reached. If we are only interested in a possible solution of the steady state equation (5.7) 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 (5.10) or

$$N_h(q_h) = r_h \ . \tag{5.22}$$

Also, if the time-dependent system (5.11) is solved by means of an implicit time-stepping method - in order to circumvent the stability bounds on Δt - we have to solve systems (5.22) at each step time step. As soon as we mix time-dependent solution with these

implicit solution methods and give up time accuracy for (5.22), there is little or no difference between these time stepping procedures and (nonlinear) relaxation methods.

Starting with the nonlinear system (5.22), two direct multigrid approaches are open. We can either apply the nonlinear multiple grid algorithm (FAS) directly to the system (5.22) 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 [31] 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 [30] and Mulder [49]; the nonlinear FAS procedure is used by Steger [67], Jespersen [30] and Hemker-Spekreijse [65, 25] and Dick [15].

In all these papers upwind discretisations have been used. In [30, 67] the Steger-Warming scheme is used; [49] uses the differentiable van Leer flux-splitting method; [25, 25] use Osher's flux difference splitting, and [15] uses Lombard's flux difference splitting. (In [14] Dick also considers Roe's flux difference splitting for the one-dimensional 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 [49] introduces the so called Switched Evolution Relaxation scheme, which is a chimera of a forward Euler time-stepping and a Newton method:

$$\left[\frac{1}{\Delta t} I - \frac{\partial}{\partial q} N_h(q_h^{(n+1)}) \right] (q_h^{(n+1)} - q_h^{(n)}) = N_h(q_h^{(n)}) . \tag{5.23}$$

For $\Delta t \to 0$, this gives the simple time stepping procedure; for $\Delta t \to \infty$, (5.23) is equivalent to Newton's method. In the actual computation Δt varies, depending on the size of the residual, such that (5.23) 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) [8]. 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 6, 7 and 8 we will give a more detailed description of a fully implicit multigrid method.

Chapter 6

Multigrid for the first-order discretisation of the Euler equations

6.1 The first-order finite volume discretisation

To discretise (5.6), the domain Ω is subdivided into disjunct quadrilateral cells $\Omega_{i,j}$, in a regular fashion such that

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

We restrict ourselves to subdivisions that are topologically equivalent with simple square meshes, such that $\Omega_{i,j}$ and $\Omega_{i,j\pm1}$ or $\Omega_{i\pm1,j}$ are neighbouring cells. Further we denote the neighbours of $\Omega_{i,j}$ by Ω_{ijk}, (k=N,S,E,W) and a common wall by $\Gamma_{ijk} = \bar{\Omega}_{ij} \cap \bar{\Omega}_{ijk}$. The boundary of Ω_{ij} is given by $\partial\Omega_{ij} = \cup_{k=N,S,E,W}\Gamma_{ijk}$. 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 (5.6) over $\Omega_{i,j}$ we obtain

$$\frac{\partial}{\partial t}\int\int_{\Omega_{i,j}} q\ dx\,dy + \int_{\delta\Omega_{i,j}} (f\,n_x + g\,n_y)\ ds = 0 \tag{6.1}$$

or

$$V_{ij}\ \frac{\partial}{\partial t}\ q_{ij} + \sum_k \int_{\Gamma_{ijk}} (f\,n_x + g\,n_y)\ ds = 0, \tag{6.2}$$

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

$$\int_{\Gamma_{ijk}} (f\,n_x + g\,n_y)\ ds = f_{ijk}\,s_{ijk}, \tag{6.3}$$

where s_{ijk} is the length of Γ_{ijk} and f_{ijk} is the mean flux outward $\Omega_{i,j}$ over the side Γ_{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 (5.6) 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 $f_{ijk}(q^k_{ij}, q^k_{ijk})$ is introduced to replace f_{ijk}. Here, q^k_{ij} and q^k_{ijk} are approximations of q at both sides of Γ_{ijk}. Thus we obtain the semi-discretisation of (5.6):

$$V_{ij}\ \frac{\partial}{\partial t}\ q_{ij} = -\sum_k\ s_{ijk}\ f_{ijk}(q^k_{ij}, q^k_{ijk})\,, \tag{6.4}$$

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

$$N_h(q_h) = 0, \tag{6.5}$$

which is short for

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

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

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

The main difficulty in the discretisation of (6.3) 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 Ω_{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 [20, 63]. These fluxes are used as $f_{ijk}(q_{ij}, q_{ijk})$. Approximate Riemann-solvers have been proposed by Steger-Warming [68], Van Leer [20, 72, 73], Roe [59, 58], Osher [53, 55] 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 Γ_{ijk}, directed from Ω_{ij} to Ω_{ijk}, be given by $(n_1, n_2) = (\cos\phi_{ijk}, \sin\phi_{ijk})$, then the simple local rotation

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

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

$$f_{ijk} := f_{ijk}(q_{ij}^k, q_{ijk}^k) = T_{ijk}^{-1}\, f(T_{ijk}\, q_{ij},\, T_{ijk}\, q_{ijk}), \tag{6.6}$$

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 $f(\,.\,,\,.\,)$ is called the *numerical flux function*. We see that the quantities s_{ijk} and ϕ_{ijk} are the only geometrical data about the mesh, needed to set up equation (6.5). 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 (5.1). 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.

6.2 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 (6.6). 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) [25] variant. Here we restrict ourselves to the P-variant. The advantages of the Osher discretisation procedure can be found e.g. in [53, 55]. 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 6.4) how its linearisation is obtained.

According to Osher, the numerical flux function in (6.6) is defined by

$$f(q_0, q_1) = \frac{1}{2}\ f(q_0) + f(q_1) - \int_{q_0}^{q_1} |f_q(w)|\, dw \tag{6.7}$$

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 λ of $f_q(w)$. These eigenvalues form the diagonal matrix Λ in the eigenvalue- eigenvector decomposition

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

In (6.7) the integration path is still to be defined, but we know that the matrix has a complete set of eigenvalues λ_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 $q = q(s)$, $0 \le s \le 1$, $q(0) = q_0$, $q(1) = q_1$. This path is divided into subpaths Γ_k, $k = 1, 2, 3$, connecting the states $q_{(k-1)/3}$ and $q_{k/3}$. These subpaths Γ_k are constructed such that on Γ_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 Γ_k. These $\psi_l^m(q)$, $m = 1, 2, 3, 4$ are

$$\begin{aligned}
\psi_3^4 &= \psi_3^1 = v, \\
\psi_1^4 &= \psi_4^1 = z, \\
\psi_2^1 &= u + \frac{2}{\gamma - 1}\, c, \\
\psi_2^4 &= u - \frac{2}{\gamma - 1}\, c, \\
\psi_1^2 &= \psi_1^3 = u, \\
\psi_4^2 &= \psi_4^3 = p,
\end{aligned} \tag{6.8}$$

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

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

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

$$\frac{c_{2/3}}{c_{1/3}} = \exp\left(\frac{z_{2/3} - z_{1/3}}{2\gamma}\right) = \alpha, \tag{6.9}$$

and we arrive at the linear system

$$\begin{aligned} u_{1/3} + \frac{2}{\gamma - 1} c_{1/3} = u_0 + \frac{2}{\gamma - 1} c_0 &=: \Psi_0, \\ u_{2/3} - \frac{2}{\gamma - 1} c_{2/3} = u_1 - \frac{2}{\gamma - 1} c_1 &=: \Psi_1, \\ c_{2/3} = \alpha\, c_{1/3}&, \\ u_{2/3} = u_{1/3}&. \end{aligned} \tag{6.10}$$

A meaningful solution exists as long as no cavitation occurs ($\Psi_0 > \Psi_1$).
This system is easily solved as

$$\begin{aligned} c_{1/3} &= \frac{\gamma - 1}{2} \frac{\Psi_0 - \Psi_1}{1 + \alpha}, \\ c_{2/3} &= \alpha\, c_{1/3}, \\ u_{1/2} := u_{1/3} = u_{2/3} &= \frac{\Psi_1 + \alpha \Psi_0}{1 + \alpha}\, u. \end{aligned} \tag{6.11}$$

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

$$\begin{aligned} \bar{\lambda}_0 &:= \lambda_{m(1)}(q_0) = u_0 - c_0, \\ \bar{\lambda}_{1/3} &:= \lambda_{m(1)}(q_{1/3}) = u_{1/3} - c_{1/3}, \\ \bar{\lambda}_{1/2} &:= \lambda_{m(2)}(q_{1/3}) = \lambda_{m(2)}(q_{2/3}) = u_{1/3} = u_{2/3}, \\ \bar{\lambda}_{2/3} &:= \lambda_{m(3)}(q_{2/3}) = u_{2/3} + c_{2/3}, \\ \bar{\lambda}_1 &:= \lambda_{m(3)}(q_1) = u_1 + c_1. \end{aligned} \tag{6.12}$$

Because $\lambda_{1,4}$ are genuinely nonlinear eigenvalues, $\lambda_{m(k)}$ is monotonous along Γ_k, $k = 1, 3$ and $\lambda_{m(k)}(q(s))$ changes sign at most once along these Γ_k. E.g. a sonic point q_{s1} with $\lambda_{m(1)}(q(s_1))$ exists on Γ_1 if $\bar{\lambda}_0 \bar{\lambda}_{1/3} \leq 0$. This sonic point is computed from the linear system

$$\begin{aligned} v_s = v_0\,, \qquad & u_s - c_s = 0, \\ z_s = z_0\,, \qquad & u_s + \frac{2}{\gamma - 1} c_s = \Psi_0\, u. \end{aligned} \tag{6.13}$$

Similarly, a sonic point q_{s2} is found on Γ_3 if $\bar{\lambda}_{2/3} \bar{\lambda}_1 \leq 0$.

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 (6.7) we obtain

$$f(q_0, q_1) = f(q_0)\ (\text{sign}\ (\bar{\lambda}_0) + 1)/2 \tag{6.14}$$

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

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

$$
\begin{aligned}
f(q_0, q_1) = \quad & \text{if } \overline{\lambda}_0 > 0 \text{ then} && f(q_0) \\
& + \text{ if } \overline{\lambda}_0\,\overline{\lambda}_{1/3} < 0 \text{ then} && \text{sign}\,(\overline{\lambda}_{1/3})\;\; f(q_{s1}) \\
& + \text{ if } \overline{\lambda}_{1/3}\,\overline{\lambda}_{1/2} < 0 \text{ then} && f(q_{1/3}) \\
& + \text{ if } \overline{\lambda}_{1/2}\,\overline{\lambda}_{2/3} < 0 \text{ then} && f(q_{2/3}) \\
& + \text{ if } \overline{\lambda}_{2/3}\,\overline{\lambda}_1 < 0 \text{ then} && \text{sign}\,(\overline{\lambda}_1)\;\; f(q_{s2}) \\
& + \text{ if } \overline{\lambda}_1 < 0 \text{ then} && f(q_1).
\end{aligned}
\tag{6.15}
$$

This expression seems rather complex. However, if the ordered sequence $\overline{\lambda}_0, \overline{\lambda}_{1/3}, \overline{\lambda}_{1/2}, \overline{\lambda}_{2/3}, \overline{\lambda}_1$ can be split in two parts (possible 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 [65, 25, 66].

6.3 The numerical flux at the boundary

The flux of the conservative variables f_{ijk}, at the boundary of the domain Ω 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 6.2, 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

$$
\frac{\partial}{\partial t} + \frac{\partial}{\partial x} f(q) = 0, \qquad t \geq 0, \;\; x \geq 0. \tag{6.16}
$$

In quasi-linear form we write (6.16) as

$$
q_t + A(q)\, q_x = 0\,, \tag{6.17}
$$

where $A(q) = df/dq$.
For the hyperbolic system (6.17), a complete set of real eigenvalues $\Lambda(q)$ and linearly independent eigenspaces $R(q)$ exists and we obtain

$$
q_t + R(q)\,\Lambda(q)\,R^{-1}(q)\; q_x = 0\,. \tag{6.18}
$$

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

$$\frac{dw}{dq} = R^{-1}(q)\,, \tag{6.19}$$

we find the uncoupled system

$$w_t + \Lambda(w)\; w_x = 0\,. \tag{6.20}$$

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 of boundary conditions $B_i(q,t) = 0$ is required such that

$$\frac{dB_i}{dq}\,\frac{dq}{dw_j} = \frac{dB}{dq}\,R^{+}(q) \tag{6.21}$$

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 two-dimensional problem (5.6) 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 6.2.

To satisfy the boundary conditions in the discrete equations (6.5) 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 .

$$f_{ijk} = f(q_B) = f(q_B, q_{ij})\,. \tag{6.22}$$

In view of (6.7) the equality (6.22) implies

$$\int_{q_B}^{q_{ij}} f_q(w)\,dw = \int_{q_B}^{q_{ij}} |\, f_q(w)\,|\,dw\,, \tag{6.23}$$

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

$$\lambda_{m(k)}\,(q(s)) \;\geq\; 0\,. \tag{6.24}$$

Such a path can be constructed again as a sum of subpaths along eigenvectors, as described in Section 6.2 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 Γ_k are computed by means of the Riemann invariants (as in Section 6.2) and the boundary data.

6.4 The linearisation of Osher's scheme

Both in the case of a complete linearisation of the discrete system (6.5) 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 (6.5) we obtain

$$\frac{\partial (N_h(q_h))_{ij}}{\partial q_{lm}} = \frac{\partial}{\partial q_{lm}} \sum_k f_{ijk}(q_{ij}, q_{ijk})\ s_{ijk}$$

$$= \sum_k s_{ijk} \frac{\partial}{\partial q_{lm}} f_{ijk}(q_{ij}, q_{ijk}) =$$

$$= \textstyle\sum_k\ s_{ijk} \frac{\partial}{\partial q_{ij}} f_{ijk}(q_{ij}, q_{ijk}) \quad \text{if } \Omega_{lm} = \Omega_{ij}, \tag{6.25}$$

$$= s_{ijk} \frac{\partial}{\partial q_{ijk}} f_{ijk}(q_{ij}, q_{ijk}) \quad \text{if } \Omega_{lm} = \Omega_{ijk}, \tag{6.26}$$

$$= 0 \quad \text{otherwise .} \tag{6.27}$$

Now, in view of (6.6), 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 6.2.

If in (6.25) $q_{ijk} = q_B$ is a boundary state, then a relation $q_{ijk} = q_B(q_{ij})$ exists and the corresponding term in (6.25) is to be read as

$$s_{ijk} \frac{d}{dq_{ij}} f_{ijk}(q_{ij}, q_{ijk}) = s_{ijk} \frac{d}{dq_{ij}} f_{ijk}(q_{ij}, q_B(q_{ij})) \tag{6.28}$$

$$= s_{ijk} \frac{d}{dq_{ij}} \left\{ T^{-1} f(Tq_{ij}, Tq_B(q_{ij})) \right\}$$

$$= s_{ijk}\ T^{-1} f'_{(0)}(Tq_{ij}, Tq_B)\ T + s_{ijk}\ T^{-1} f'_{(1)}(Tq_{ij}, Tq_B)\ T \frac{dq_B}{dq_{ij}}\,.$$

Here T denotes T_{ijk} as in eq. (6.6). 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 6.3.

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

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

$$p = \rho c^2/\gamma\,,$$

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

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

$$f'(q) = \frac{df}{dq} = \begin{pmatrix} \beta\rho u/c & \rho & 0 & -\beta\rho u/2 \\ \beta\rho(u^2+c^2)/c & 2\rho u & 0 & -\beta(\rho u^2+p)/2 \\ \beta\rho uv/c & \rho v & \rho u & -\beta\rho uv/2 \\ \beta u(E+p+\rho c^2)/c & \rho u^2+E+p & \rho uv & -\beta u(E+p)/2 \end{pmatrix}, \tag{6.29}$$

where $\beta = 2/(\gamma-1)$. In terms of this matrix, from (6.15) follows

$$\begin{aligned} \frac{\partial}{\partial q_0} f(q_0, q_1) = \quad & \text{if } \overline{\lambda}_0 > 0 \quad \text{then} \quad f'(q_0) \\ & + \text{if } \overline{\lambda}_0\,\overline{\lambda}_{1/3} < 0 \text{ then} \quad \text{sign}\,(\overline{\lambda}_{1/3})\, f'(q_{s1}) \frac{\partial q_{s1}}{\partial q_0} \\ & + \text{if } \overline{\lambda}_{1/3}\,\overline{\lambda}_{1/2} < 0 \text{ then} \quad \pm f'(q_{1/3}) \frac{\partial q_{1/3}}{\partial q_0} \\ & + \text{if } \overline{\lambda}_{1/2}\,\overline{\lambda}_{2/3} < 0 \text{ then} \quad \pm f'(q_{2/3}) \frac{\partial q_{2/3}}{\partial q_0} \,. \end{aligned} \tag{6.30}$$

The derivatives $\partial q/\partial q_0$, $q = q_{s1}$, $q_{1/3}$, $q_{2/3}$, are derived from the differentiable relations (6.9)-(6.13). The explicit expressions are found in [25] and [65].

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 $\overline{\lambda}_{1/2} = u_{1/3} = u_{2/3} \neq 0$. 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.

6.5 Multigrid iteration

In order to solve (6.5), we first generalise the problem slightly to

$$N_h(q_h) = r_h \,. \tag{6.31}$$

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

$$N_{h_i}(q_{h_i}), \qquad \text{with} \quad h_0 > h_1 > \ldots > h_l = h \,.$$

For the mesh width h_{i-1} we take $h_{i-1} = 2\,h_i$. 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 (6.31) consists of the steps: start with an approximate solution q_h; improve q_h by application of p nonlinear (pre-) relaxation iterations to $N_h(q_h) = r_h$; compute the residual $N_h(q_h)$; 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}); compute

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

approximate the solution of

$$N_{2h}(q_{2h}) = r_{2h} \tag{6.32}$$

by application of σ FAS cycles. The result is $\tilde{q}_{2h}$; correct the current solution by

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

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 (6.5), 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 (6.32).

In order to complete the description of the FAS-cycle we need to be explicit about:
(1) the choice of the operators N_{2h}, $P_{h,2h}$, $\overline{R}_{2h,h}$ and eventually $R_{2h,h}$;
(2) the FAS strategy, i.e. the numbers p, q, σ;
(3) the nonlinear relaxation method.

6.5.1 A nested sequence of Galerkin discretisations

For the operators $P_{h,2h}$ and $\overline{R}_{2h,h}$ 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 $\Omega_{i,j}$) and where the residual is weighted by characteristic functions on all $\Omega_{i,j}$. 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 $\overline{R}_{2h,h}$. Notice that $\overline{R}_{2h,h} = P_{h,2h}^T$. With these choices it is clear that

$$N_{2h}(q_{2h}) = \bar{R}_{2h,h}\, N_h\, (\, P_{h,2h} q_{2h}\,)\,, \tag{6.33}$$

i.e. the coarse grid finite volume discretisation is a formal Galerkin approximation of the fine grid finite volume discretisation. Using (6.33) on all different levels we obtain a nested sequence of discretisations, i.e. the following scheme of operators and spaces is commutative.

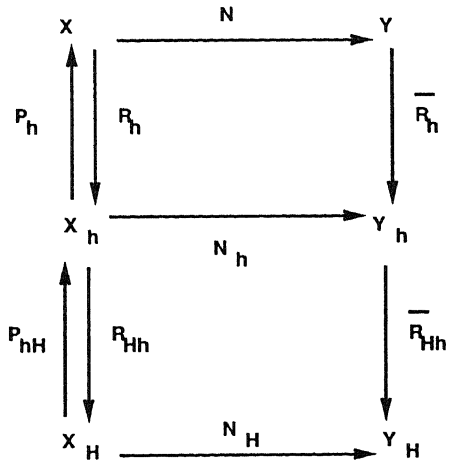


Figure 6.1: Nested sequence of discretisations

The effect of the Galerkin approximation $N_{2h} = \overline{R}_{2h,h}\, N_h\, P_{h,2h}$ 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 (6.32) is solved exactly, then

$$\bar{R}_{2h,h}\,[\, r_h - N_h\, P_{h,2h}\, R_{2h,h}\, \tilde{q}_h]$$

$$= \bar{R}_{2h,h}\,[\, N_h q_h - N_h\, P_{h,2h}\, R_{2h,h}\, q_h]\,,$$

or, for the restriction of the residual

$$\bar{R}_{2h,h}\,[\,r_h - N_h(\tilde{q}_h)] \tag{6.34}$$

$$= \bar{R}_{2h,h}\,[(N_h q_h - N_h\,P_{h,2h}\,R_{2h,h}\,q_h\,) - (N_h \tilde{q}_h - N_h\,P_{h,2h}\,R_{2h,h}\,\tilde{q}_h\,)]\ .$$

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

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

Because $\overline{R}_{2h,h}$ 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.

6.5.2 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 σ 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, σ)-strategies are not much different in efficiency. Usually a smaller convergence factor is compensated by a corresponding amount of additional work.

6.5.3 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 multigrid 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 4×4 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 [24] we compared some of these relaxations in combination with a uniform grid. There also the effect of other strategies (p, q, σ) was considered. For a standard model problem and a non-uniform grid, results of such a multigrid procedure are shown in Fig.8.1-8.3.

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. [37]. Jespersen [30] 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.

6.5.4 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.

6.6 Conclusion

For transonic computations [24, 25, 39, 42] 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 (≈ 0.3 per FAS-cycle) 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 multigrid 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 [25, 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.

Chapter 7

Defect correction for higher order Euler computations

7.1 Second order discretisation

The first-order discretisation discussed in Section 6.1 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 (5.1) and avoids the necessity of a shock fitting technique. Therefore, we consider only schemes that are still based on (6.5), and we select $f_{ijk}(q^k_{ij}, q^k_{ijk})$ that yield a better approximation to (6.3) than (6.6).

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, 73] the second in [54, 67]. In the first case, in (6.5) q^k_{ij} and q^k_{ijk} are obtained by some interpolation from $q_h = \{q_{ij}\}$. In the latter, $f_{ijk}(q^k_{ij}, q^k_{ijk})$ is obtained from $f_{ijk}(q^k_{ij}, q^k_{ijk}) \cup f_{ijk}(q^k_{ij}, q^k_{ij})$. 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 (6.6) 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 2×2 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^S_h. It is easily shown that this *superbox*

scheme is second-order accurate in the sense that

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

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^l_{ijk} (q^r_{ijk}) at the left (right) side of all walls G_{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 [62, 27].

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.(6.5) reduces to

$$N_h(q_h)_i = f_{i+1/2} - f_{i-1/2}\,, \tag{7.1}$$

where $f_{i+1/2} = f(q^l_{i+1/2}, q^r_{i+1/2})$.

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

$$q^l_{i+1/2} = q_i + \frac{1}{2}\ \Delta q_{i-1/2}, \tag{7.2}$$

$$q^r_{i+1/2} = q_{i+1} - \frac{1}{2}\ \Delta q_{i+\frac{3}{2}}\ .$$

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 ψ [70, 65], and to interpolate by

$$q^l_{i+1/2} = q_i + \frac{1}{2}\ \psi^l_{i+1/2}\ \ \Delta q_{i-1/2}, \tag{7.3}$$

$$q^r_{i-1/2} = q_i - \frac{1}{2}\ \psi^r_{i-1/2}\ \ \Delta q_{i+1/2},$$

where $\psi^l = \psi(\mathrm{R})$ and $\psi^r = \psi(1/\mathrm{R})$ are chosen, depending on $\mathrm{R} = \Delta q_{i+1/2}/\Delta q_{i-1/2}$, such that $q^l_{i-1/2}$ lies between q_{i-1} and q_i, and $q^r_{i+1/2}$ between q_i and q_{i+1}, (cf. [70, 65]). One possible choice is the Van Albada limiter [1, 65].

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

Van Leer [73] proposes a linear combination of the one-sided and central interpolation. Parametrised by κ we obtain

$$q^l_{i+1/2} = q_i + \frac{1}{4}\ \left[(1-\kappa)\Delta q_{i-1/2} + (1+\kappa)\Delta q_{i+1/2}\right], \tag{7.4}$$

$$q^r_{i-1/2} = q_i - \frac{1}{4}\left[(1-\kappa)\Delta q_{i+1/2} + (1+\kappa)\Delta q_{i-1/2}\right] .$$

This general formula contains for instance: ($\kappa = -1$) the one-sided second-order scheme (7.2); ($\kappa = 0$) 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^S, corresponds to the use of $\kappa = +1$ for odd i, and $\kappa = -1$ for even i.

The interpolation (7.4) is well defined in the interior cells of the domain. In the cells near the boundary $\partial\Omega^\star$, one of the values $\Delta q_{i\pm1/2}$ is not defined, by the absence of a value q_i corresponding to a point outside $\Omega^\star$. 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 Ω_i near the boundary. This corresponds with the "superbox" approximation for these cells. For the superbox scheme and for the scheme (7.4), with different values of κ, results are shown in [39]. Some of them are also shown in Fig.8.4-8.9. The second-order surface pressure distributions in Fig.8.9 are preceded by first-order distributions (Fig.8.8). (Notice the very fast convergence for the latter.)

Thus, with the MUSCL approach we have constructed a second-order accurate semi-discretisation of (5.6)

$$(q_h)_t + N_h^2(q_h) = 0 . \tag{7.5}$$

7.2 The solution of the second-order discrete system

One possibility to find the solution of the steady state equations

$$N_h^2(q_h) = 0, \tag{7.6}$$

is to take an initial guess and to solve the semi-discretised equation (7.5) for $t \to \infty$, 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

$$N_h^2(q_h) = r_h, \tag{7.7}$$

directly.

However, if we try to solve the second-order discretisation (7.6) in the same manner as we do the first-order equations, we may expect difficulties because the nonlinear equations (7.6) 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 [30]. 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, 74]. Here van Leer's flux splitting scheme [72] 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

$$N_h^1(q_h^{(1)}) = 0, \tag{7.8}$$

$$N_h^1(q_h^{(i+1)}) = N_h^1(q_h^{(i)}) - N_h^2(q_h^{(i)}). \tag{7.9}$$

If the problem is smooth enough, the accuracy of $q_h^{(i)}$ is of order 2 for $i \geq 2$ (Theorem 2.2). If the solution is not smooth (higher-order derivatives are dominating), there is no reason to expect the solution of (7.6) to be more accurate than the solution of (7.8). Nevertheless, in [21, 39, 38] evidence is given that a few defect correction steps may improve the solution considerably. This is also shown in Fig.8.8-8.9.

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 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 (7.9) 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.8.6, 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) τ-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 [21]. A drawback is that these methods rely on the existence of an asymptotic expansion of the (truncation) error for $h \to 0$, 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 (τ-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 (7.9).

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 (7.9) 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 (7.9) to τ-extrapolation. See [42] for some numerical results.

Having both a first- and a second-order discrete operator at our disposal, Brandt's double discretisation [8] 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 (7.4), 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^k_{ij} and q^k_{ijk} shows cavitation.)

7.3 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, iL, are not only used for the realisation of FAS-iteration steps as described in Section 6.5, 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 $(N^1_{(i)}\, q_{(i)} = 0)$
  enddo;
  (1b) $q_{(i+1)} := P^2_{i+1,i}\, q_{(i)}$;
(1) **enddo**;
(2) **for** j **from** 1 **to** k_L **do** FASCYCLE $(N^1_{(L)}\, q_{(L)} = 0)$
enddo;
(3) $q_{(L)} := q_{(L)} + P^S_{L,L-1}\,(R^1_{L-1,L}\, q_{(L)} - q_{(L-1)})$;
(4) **for** d **from** 1 **to** dcps **do**
  (4a) $r_{(L)} := N^1_{(L)}(q_{(L)}) - N^2_{(L)}(q_{(L)})$;
  (4b) **for** j 1 kd **do** FASCYCLE $(N^1_{(L)}\, q_{(L)} = r_{(L)})$
  enddo;
(4) **enddo**;

Stage (1) is an FMG process to obtain a first-order accurate initial estimate at level L. The prolongation $P^2_{i+1,i}$ 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 $C\, h_{(i)} = \mathcal{O}(2^{l-i}h)$ for $h_{(L)} = h \to 0$. Now theorem 2.2 shows that, for a fixed $k_i = k$ at all levels, the iteration error at level i is $\approx C h_{(i)}\, \mu^k/(1 - 2\mu^k)$, where μ 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,...,L. Stage (2) is the FAS-iteration to obtain the solution to $N^1_h(q_h) = 0$ up to truncation error accuracy. Stage (3) is a Richardson extrapolation step to find a second-order initial estimate for q_h. The prolongation $P^S_{L,L-1}$ and the restriction $R^1_{L-1,L}$ are piecewise bilinear interpolation over superboxes and averaging over cells, respectively, so that $R^1_{L-1,L}P^S_{L,L-1} = I_{L-1}$ is the identity, and $P^S_{L,L-1}R^1_{L-1,L}$ is a projection operator. With the asymptotic expansion for the error in q_h as

$$q_h = R_h\hat{q} + h^p R_h\, e + \mathcal{O}(h^{p+1}), \tag{7.10}$$

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

$$R_{2h}\,\hat{q} = 2\; R_{2h,h}q_h - q_{2h} + \mathcal{O}(h^2)\;. \tag{7.11}$$

We find the extrapolated value of q_h in (3) as the sum of (7.11) and $(I_i - P^S_{i,i-1} R^0_{i-1,i}) q_h \in \mathrm{Ker}(R_{2h})$. 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, $i = 1, 2, ..., L$, the Richardson extrapolation can reduce the error significantly [21]. Stage (4) is the defect correction iteration (7.9). If the defect correction iteration starts with a first-order initial approximation, for second-order accuracy it is sufficient to take dcps=1. This necessitates an improvement of the error by a factor $\mathcal{O}(h)$ in the iteration (4b), i.e. we need kd $= \mathcal{O}(\log(h))$. However, since the FAS iteration is the expensive part of the computation in (4), for most purposes we take kd=1 and a sufficiently large number for dcps. Results for the algorithm can be found in [21, 23, 25, 39, 43, 44, 42].

Chapter 8

Solution of the Navier-Stokes equations

8.1 Introduction

Mainly based on the Euler method described in the sections 3 to 5, a Navier-Stokes method has been developed recently [61, 22, 41, 40] 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.

8.2 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 (5.2). 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 (5.5). In fact, as $1/Re = 0$, the Navier-Stokes discretisation reduces to exactly the Euler discretisations described in the Sections 6 and 7.

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

$$\int_{\partial\Omega_{i,j}} (f(q)n_x + g(q)n_y)ds - \frac{1}{Re}\int_{\partial\Omega_{i,j}} (r(q)n_x + s(q)n_y)ds = 0\,, \quad \forall i,j. \tag{8.1}$$

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

$$\int_{\partial\Omega_{i,j}} T^{-1}(n_x,n_y)f(T(n_x,n_y)q)ds - \frac{1}{Re}\int_{\partial\Omega_{i,j}} (r(q)n_x + s(q)n_y)ds = 0\,, \quad \forall i,j, \tag{8.2}$$

with $T(n_x,n_y)$ the rotation matrix in (6.6).

8.2.1 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 $(f(q))_{i+1/2,j} = f(q^l_{i+1/2,j}, q^r_{i+1/2,j})$, where the superscripts l and r refer to the left and right side of volume wall $\partial\Omega_{i+1/2,j}$, first-order accurate convection is obtained again by taking $q^l_{i+1/2,j} = q_{i,j}$ and $q^r_{i+1/2,j} = q_{i+1,j}$. Higher-order accurate convection is obtained again with the κ-schemes as introduced in (7.4), 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 [22] for the $\kappa = 1/3$ approximation:

$$\psi(R) = \frac{R + 2R^2}{2 - R + 2R^2}\, u. \tag{8.3}$$

8.2.2 Evaluation of diffusive fluxes

For the evaluation of the diffusive fluxes at a volume wall, it is necessary to compute grad(u), grad(v) and grad(c^2) at that wall. For this we use a standard technique [57]. To compute for instance $(\text{grad}(u))_{i+1/2,j}$, we use Gauss' theorem:

$$(\nabla u)_{i+1/2,j} = \frac{1}{A_{i+1/2,j}} \int_{\partial\Omega_{i+1/2,j}} u\, \mathbf{n}\, ds, \tag{8.4}$$

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

$$z_{i,j\pm1/2} = \frac{1}{2}(z_{i-1/2,j\pm1/2} + z_{i+1/2,j\pm1/2})\,. \tag{8.5}$$

A similar expression exists for $z_{i\pm1/2,j}$.
The line integrals $\int_{\partial\Omega_{i+1/2,j}} u n_x ds$ and $\int_{\partial\Omega_{i+1/2,j}} u n_y ds$ are approximated by

$$\begin{aligned} \int_{\partial\Omega_{i+1/2,j}} u\, n_x\, ds \;=\; & u_{i+1,j}\;(y_{i+1,j+1/2} - y_{i+1,j-1/2}) \\ & + u_{i+1/2,j+1/2}\;(y_{i,j+1/2} - y_{i+1,j+1/2}) \\ & + u_{i,j}\;(y_{i,j-1/2} - y_{i,j+1/2}) \\ & + u_{i+1/2,j-1/2}\;(y_{i+1,j-1/2} - y_{i,j-1/2})\,, \end{aligned} \tag{8.6}$$

and

$$\begin{aligned} \int_{\partial\Omega_{i+1/2,j}} u\, n_y\, ds \;=\; & u_{i+1,j}\;(x_{i+1,j-1/2} - x_{i+1,j+1/2}) \\ & + u_{i+1/2,j+1/2}\;(x_{i+1,j+1/2} - x_{i,j+1/2}) \\ & + u_{i,j}\;(x_{i,j+1/2} - x_{i,j-1/2}) \\ & + u_{i+1/2,j-1/2}\;(x_{i,j-1/2} - x_{i+1,j-1/2})\,, \end{aligned} \tag{8.7}$$

with for $u_{i+1/2,j\pm1/2}$ the central expression

$$u_{i+1/2,j\pm1/2} = \frac{1}{4}(u_{i,j} + u_{i,j\pm1} + u_{i+1,j} + u_{i+1,j\pm1})\,. \tag{8.8}$$

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 [61, 41, 40].

8.3 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/ϵ [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/ϵ. It even appears to diverge for large values of h/ϵ [40]. 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

$$\tilde{F}_h(q_h^1) = 0, \tag{8.9}$$

$$\tilde{F}_h(q_h^{n+1}) = \tilde{F}_h(q_h^n) - \omega F_h(q_h^n), \qquad n = 1, 2, ..., N,$$

where n denotes the nth iterand, and ω a damping factor. The standard value for ω 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 2.2 predicts the solution to be second-order accurate after a single Defect Correction cycle. Because the discrete approximations of the diffusive flux are 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 $m_r + m_p > 2$, [17, 8], 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 [41].

8.4 Numerical results

To evaluate the method described, we considered as reference test case the experiment from [18], performed at $Re = 2.96\,10^5$. 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 80×32-grid shown in Fig.8.12 to the oblique 80×32-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 $k = 1/3$ and the limited $k = 1/3$ approximation are given in Fig.8.12. 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.8.12. 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 rectangular grids were often used, 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 was 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 80×32-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.8.13 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].

Bibliography

[1] G. D. Van Albada, B. Van Leer, and W. W. Roberts. A comparative study of computational methods in cosmic gasdynamics. *Astron. Astrophys.*, 108:76–84, 1982.

[2] W. K. Anderson, J. L. Thomas, and B. Van Leer. A comparison of finite volume flux vector splittings for the Euler equations, 1985.

[3] K. E. Atkinson. Iterative variants of the Nystrom method for the numerical solution of integral equations. *Numer. Math.*, 22:17–33, 1973.

[4] J. W. Boerstoel. A multigrid algorithm for steady transonic potential flows around aerofoils using newton iteration. *J. Comp. Phys.*, 48:314–343, 1982.

[5] K. Böhmer, P. Hemker, and H. J. Stetter. The defect correction approach. *Computing Suppl.*, 5:1–32, 1984.

[6] H. Brakhage. Ueber die numerische behandlung von integralgleichungen nach der quadraturformelmethode. *Numer. Math.*, 2:183–196, 1960.

[7] A. Brandt. Multi-level adaptive technique (MLAT) for fast numerical solutions to boundary value problems. In *Procs 3rd Int. Conf. Numerical Methods in Fluid Mechanics*, volume 18 of *Lect. Notes in Physics*, pages 82–89, Berlin, 1973. Springer Verlag.

[8] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods*, volume 960 of *Lect. Notes in Mathematics*, pages 220–312. Springer Verlag, 1982.

[9] A. Brandt and N. Dinar. Multigrid solutions to elliptic flow problems. In S.V. Parter, editor, *Numerical Methods for Partial Differential Equations*, pages 53–147, New York, etc., 1979. Academic Press.

[10] D. A. Caughey. Multigrid calculation of three dimensional transonic potential flows. *Appl.Math.Comp.*, 13:241–260, 1983.

[11] R. V. Chima and G. M. Johnson. Efficient solution of the Euler and Navier Stokes equations with a vectorized multi-grid algorithm. Technical Report 83-1893, AIAA, 1983.

[12] P. Colella and P. R. Woodward. The piecewise parabolic method (PPM) for gas dynamical simulations. *J. Comp. Phys*, 52:174–201, 1984.

[13] H. Deconinck. A survey of upwind principles for the multidimensional Euler equations. Technical report, Von Karman Institute for Fluid Dynamics, Rhode St Genese, 1987.

[14] E. Dick. A multigrid method for the Cauchy-Riemann and steady state Euler equations based on flux difference splitting. In W. Hackbusch, editor, *Efficient Solution of Elliptic Systems*, pages 20–37. Vieweg Verlag, 1984.

[15] E. Dick. A multigrid method for steady Euler equations, based on flux-difference splitting with respect to primitive variables. In W. Hackbusch, editor, *Robust Multi-Grid Methods*, pages 69–85. Vieweg Verlag, 1988.

[16] J. A. Essers. Explicit and implicit corrected viscosity schemes for the computation of steady transonic flows. In H. Viviand, editor, *Proceedings of the 4th GAMM-Conference on Numerical Methods in Fluid Mechanics*. Vieweg Verlag, 1982.

[17] W. Hackbusch. *Multigrid Methods and Applications*. Springer Verlag, 1985.

[18] R. J. Hakkinen, I. Greber, L. Trilling, and S. S. Abarbanel. The interaction of an oblique shock wave with a laminar boundary layer, 1958.

[19] R. W. Hamming. *Digital Filters*. Prentice Hall, Inc., Englewood Cliffs, 1977.

[20] A. Harten, P. D. Lax, and B. Van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25:35–61, 1983.

[21] P. W. Hemker. Defect correction and higher order schemes for the multigrid solution of the steady Euler equations. In W. Hackbusch and U. Trottenberg, editors, *Multi-Grid Methods II*, pages 150–165. Springer Verlag, 1986. Proceedings of the 2nd European Conference on Multigrid Methods, held at Cologne, October 1-4, 1985.

[22] P. W. Hemker and B. Koren. Multigrid, defect correction and upwind schemes for the steady Navier-Stokes equations. In K.W.Morton and M.J.Baines, editors, *Numerical methods for fluid dynamics III*, pages 153–170, Oxford, 1988. Clarendon Press.

[23] P. W. Hemker and B. Koren. A nonlinear multi-grid method for the steady Euler equations. In A. Dervieux, B. Van Leer, J. Periaux, and A. Rizzi, editors, *Numerical Simulation of Compressible Euler Flows*, pages 175–196, Braunschweig, Wiesbaden, 1989. Vieweg Verlag. Procs of GAMM workshop on the Numerical solution of the Euler equations, INRIA, Rocquencourt, France, June 1986.

[24] P. W. Hemker and S. P. Spekreijse. Multigrid solution of the steady Euler equations. In D. Braess, W. Hackbusch, and U. Trottenberg, editors, *Advances in Multi-Grid Methods*, pages 33–44, Braunschweig, 1985. Vieweg Publ. Comp. Proceedings of the Conference, Oberwolfach, dec. 1984.

[25] P. W. Hemker and S. P. Spekreijse. Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations. *Appl. Num. Math.*, 2:475–493, 1986.

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

[27] A. Jameson. 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, 1983.

[28] A. Jameson. Solution of the Euler equations for two dimensional transonic flow by a multigrid method. *Appl. Math. and Computat.*, 13:327–355, 1983.

[29] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using runge-kutta time-stepping schemes. Technical Report 81-1259, AIAA, 1981.

[30] D. C. Jespersen. Design and implementation of a multigrid code for the Euler equations. *Appl. Math. and Computat.*, 13:357–374, 1983.

[31] D. C. Jespersen. *Recent developments in multigrid methods for the steady Euler equations.* von Karman Inst., Rhode-St.Genese, Belgium, 1984. Lecture Notes of a course given at VKI, March 12-16, 1984.

[32] D. C. Jespersen. A time-accurate multiple-grid algorithm. Technical Report 85-1493-CP, AIAA, 1985.

[33] G. M. Johnson. Accelerated solution of the steady state Euler equations. In W.G. Habashi, editor, *Recent Advances in Numerical methods in Fluids. Vol.4.*, Recent advances in numerical methods in fluids. Pineridge Press, 1983.

[34] G. M. Johnson. Multiple grid convergence acceleration of viscous and inviscid flow computations. *Appl. Math. and Computat.*, 13:375–398, 1983.

[35] M. C. Joshi and R. K. Bose. *Some Topics in Nonlinear Functional Analysis.* Wiley Eastern Ltd., New Delhi etc., 1985.

[36] Y. Katznelson. *An Introduction to Harmonic Analysis.* Wiley, New York, etc., 1968.

[37] R. Kettler. Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods*, volume 960 of *Springer Lecture Notes in Mathematics*, pages 502 – 534. Springer Verlag, 1982.

[38] B. Koren. Defect correction and multigrid for an efficient and accurate computation of airfoil flows. *J. Comp. Phys.*, 77:183–200, 1988.

[39] B. Koren. Euler flow solutions for transonic shock wave - boundary layer interaction. *Int. J. Numer. Meth. Fluids*, 9:59–73, 1989.

[40] B. Koren. Multigrid and defect correction for the steady Navier-Stokes equations. *J. Comput. Phys.*, 87:25–46, 1990.

[41] B. Koren. Upwind discretization of the steady Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 11:99–117, 1990.

[42] B. Koren. *Multigrid and defect correction for the steady Navier-Stokes equations, application to aerodynamics*, volume 74 of *CWI Tracts.* CWI, 1991.

[43] B. Koren and S. P. Spekreijse. Multigrid and defect correction for the efficient solution of the steady Euler equations. In P. Wesseling, editor, *Research in Numerical Fluid Dynamics*, Notes on Numerical Fluid Mechanics, pages 87–100. Vieweg Verlag, 1987. proceedings of the 25th Meeting of the Dutch Association for Numerical Fluid Dynamics.

[44] B. Koren and S. P. Spekreijse. Solution of the steady Euler equations by a multigrid method. In S.F. McCormick, editor, *Lecture Notes in Pure and Applied Mathematics*, volume 110, pages 323–336, New York, 1988. Dekker.

[45] P. D. Lax. Shock waves and entropy. In E.H. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, New York, 1971. Acad. Press.

[46] P. D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, volume 11 of *Regional Conference Series in Applied Mathematics.* SIAM Publication, Philadelphia, 1973.

[47] J. L Lions and E. Magenes. *Problemes aux Limites Non Homogenes.* Dunod, Paris, 1968.

[48] D. R. McCarthy and T. A. Reyhner. Multigrid code for three dimensional transonic potential flow about inlets. *AIAA Journal*, 20:45–50, 1982.

[49] W. A. Mulder. Multigrid relaxation for the Euler equations. *J. Comput. Phys.*, 60:235 – 252, 1985.

[50] Ron-Ho Ni. A multiple grid scheme for solving the Euler equations. *AIAA Journal*, 20:1565–1571, 1982.

[51] Z. Nowak and P. Wesseling. Multigrid acceleration of an iterative method with applications to transonic potential flow. In R. Glowinski and J.L. Lions, editors, *Computing Methods in Applied Sciences and Engineering*, volume 6, pages 199–217. North-Holland Publ. Comp., 1985.

[52] S. Osher. Numerical solution of singular perturbation problems and hyperbolic systems of conservation laws. In L.S. Frank O. Axelsson and A. van der Sluis, editors, *Analytical and Numerical Approaches to Asymptotic problems in Analysis*, Mathematics Studies 47. North Holland Publ. Comp., 1981.

[53] S. Osher and S. Chakravarthy. Upwind schemes and boundary conditions with applications to Euler equations in general geometries. *J. Comp. Phys.*, 50:447–481, 1983.

[54] S. Osher and S. Chakravarthy. High resolution schemes and the entropy condition. *SIAM J. Numer. Anal.*, 21:955–984, 1984.

[55] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comp.*, 38:339–374, 1982.

[56] A. Papoulis. *The Fourier Integral and its Applications.* McGraw-Hill, New York, etc., 1962.

[57] R. Peyret and T. D. Taylor. *Computational Methods for Fluid Flow.* Springer Verlag, 1983.

[58] P. L. Roe. The use of the Riemann problem in finite difference schemes. In Reynolds and McCormack, editors, *Procs. 7th Int. Conf. Num. Meth. Fl. Dyn.*, volume 141 of *Lecture Notes in Physics*, pages 354–359. Springer Springer Verlag, 1980 1981.

[59] P. L. Roe. Approximate Riemann solvers, param
J. Comp. Phys., 43:357–372, 1981.

[60] W. Rudin. *Functional Analysis.* Tata McGraw-H

[61] J. J. Rusch. The use of defect correction for th
Navier-Stokes equations with large Reynolds nur

[62] W. Schmidt and A. Jameson. Euler solvers as a
namics. In W.G. Habashi, editor, *Advances in C*
Pineridge Press.

[63] J. Smoller. *Shock waves and reaction diffusion eq*
der mathematische Wissenschaften. Springer Ver

[64] J. C. South and A. Brandt. Application of a mu
flow calculations. In T.C. Adamson and M.F. P
on Transonic Flow problems in Turbomachinery
Hemisphere, 1977.

[65] S. P. Spekreijse. Multigrid solution of monoto
hyperbolic conservation laws. *Math. Comp.*, 49:1

[66] S.P. Spekreijse. *Multigrid Solution of the Steady E*
Tracts. CWI, 1988.

[67] J. L. Steger. A preliminary study of relaxation me
gasdynamics equations using flux splitting, 1981.

[68] J. L. Steger and R. F. Warming. Flux vector spl
equations with application to finite difference met
1981.

[69] H. J. Stetter. The defect correction principle an
Math., 29:425–443, 1978.

[70] P. K. Sweby. High resolution schemes using flux li
laws. *SIAM J.Numer.Anal.*, 21:995–1011, 1984.

[71] J. M. Swisshelm and G. M. Johnson. Numerica
flowfields using the Cyber 205. In R.W. Numric
tions. Plenum Publ. Comp., 1985.

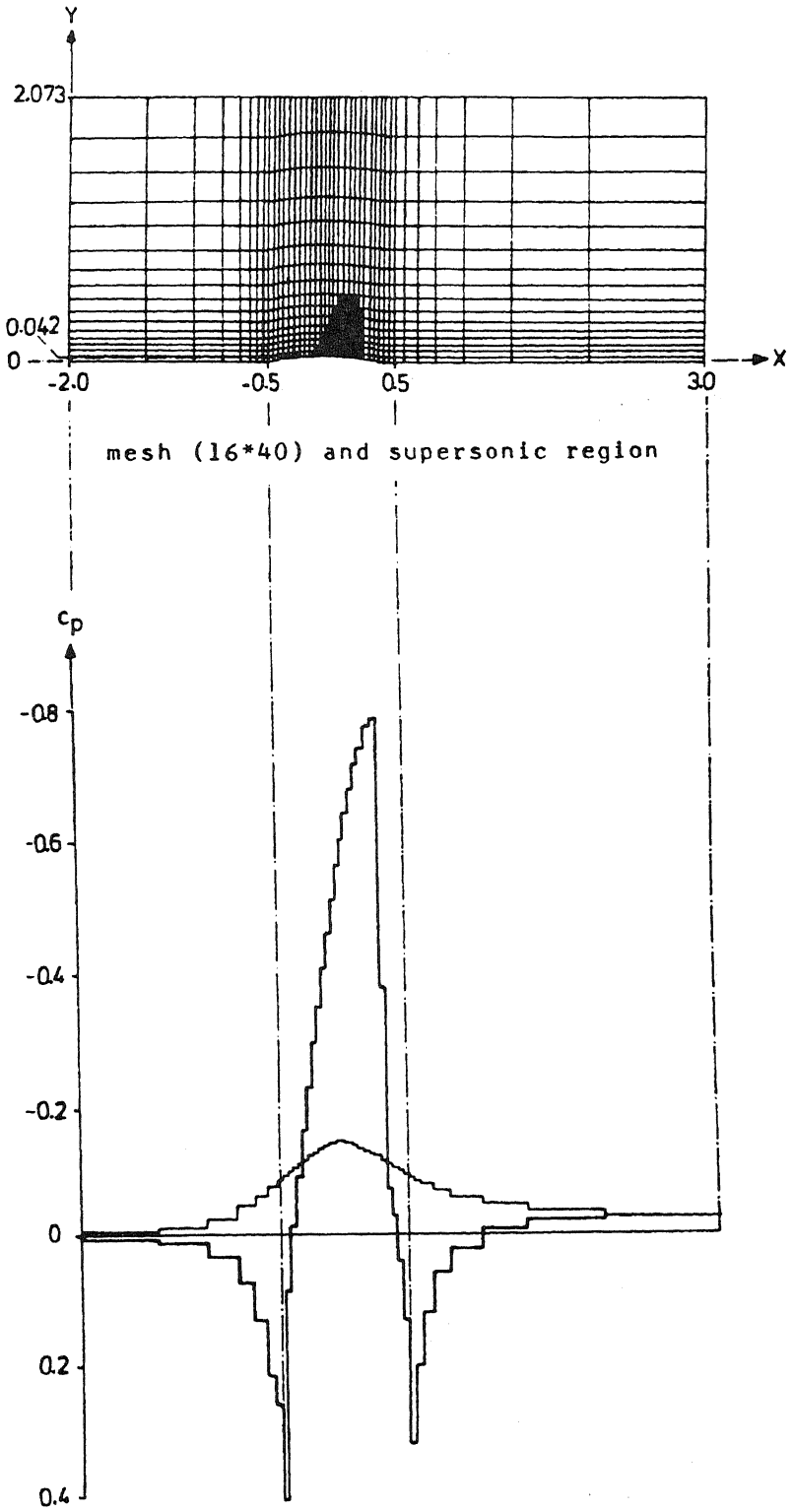
[72] B. Van Leer. Flux-vector splitting for the Euler
Conf. on numerical methods in fluid dynamics, A
in Physics 170. Springer Verlag, 1982.

[73] B. Van Leer. On the relation between the upwind
Engquist-Osher and Roe. *SIAM J.N.A.*, 5:1, 198

[74] B. Van Leer and W. A. Mulder. Relaxation me
laws. In J. A. Désidéri F. Angrand, A. Dervieux a
ical Methods for the Euler Equations of Fluid Dy
phia, 1985. SIAM.

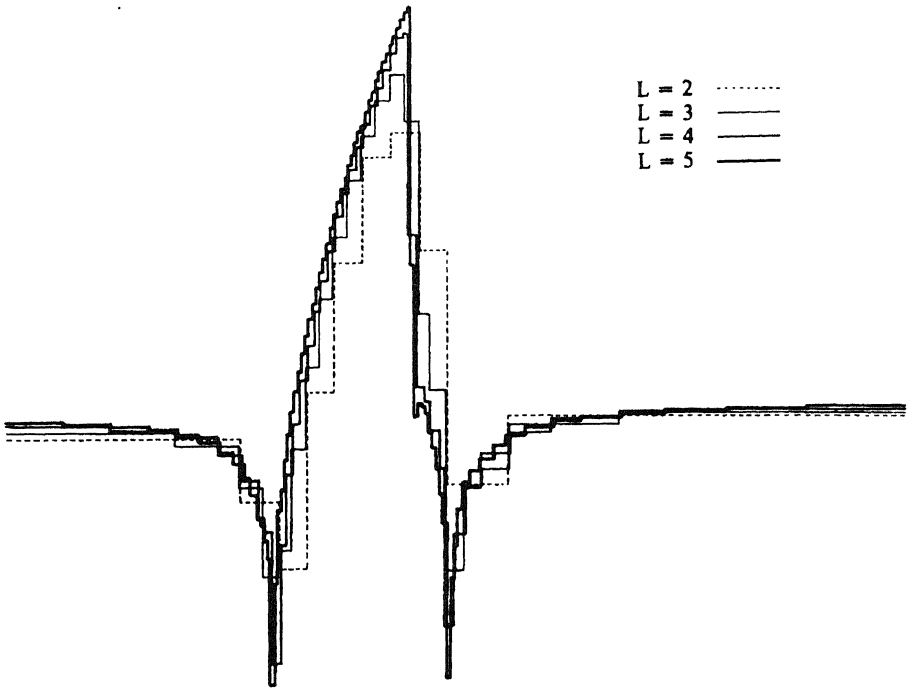
[75] K. Yosida. *Functional Analysis.* Springer Verlag, Berlin, etc., 1974.

[76] D. Young. *Iterative Solution of Large Linear Systems.* Academic Press, New York, London, 1971.



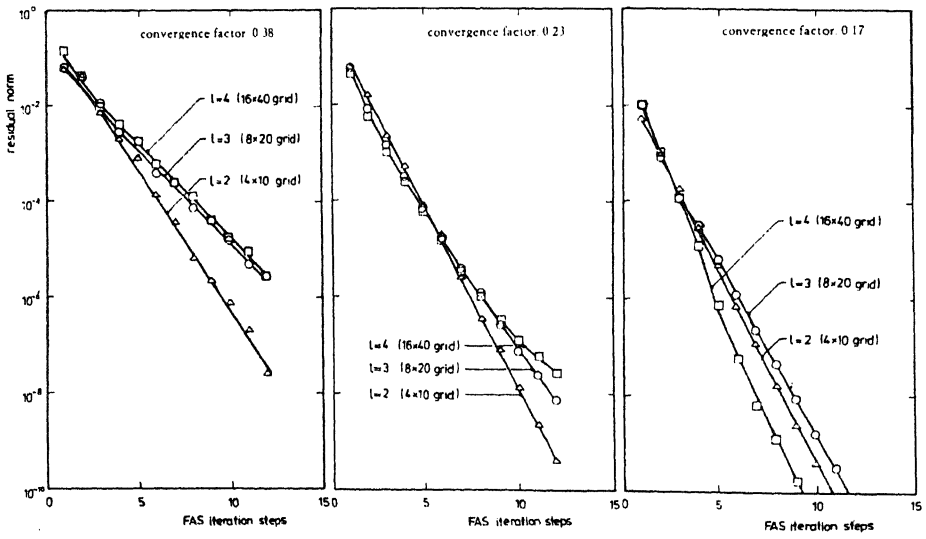
Grid ($L = 4$), supersonic region and pressure distribution along lower and upper surface

Figure 8.1: Transonic flow in a channel, $M_{\text{inlet}} = 0.85$.



Pressure distribution, c_p, along lower surface, for L=2,3,4,5.

Figure 8.2: Transonic flow in a channel, $M_{\text{inlet}} = 0.85$.



$p = q = 1, \sigma = 1$ $p = q = 1, \sigma = 2$ $p = q = 2, \sigma = 1$

Figure 8.3: Convergence histories for transonic channel flow, $M_{\text{inlet}} = 0.85$.

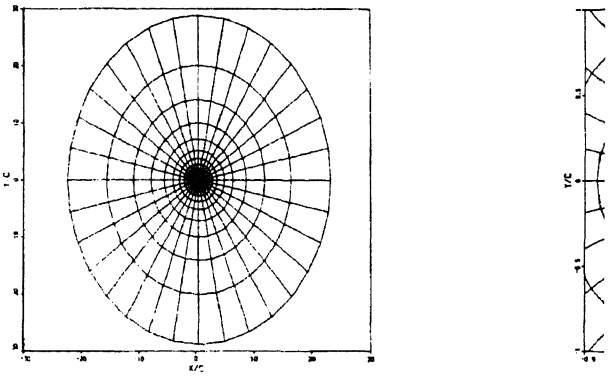
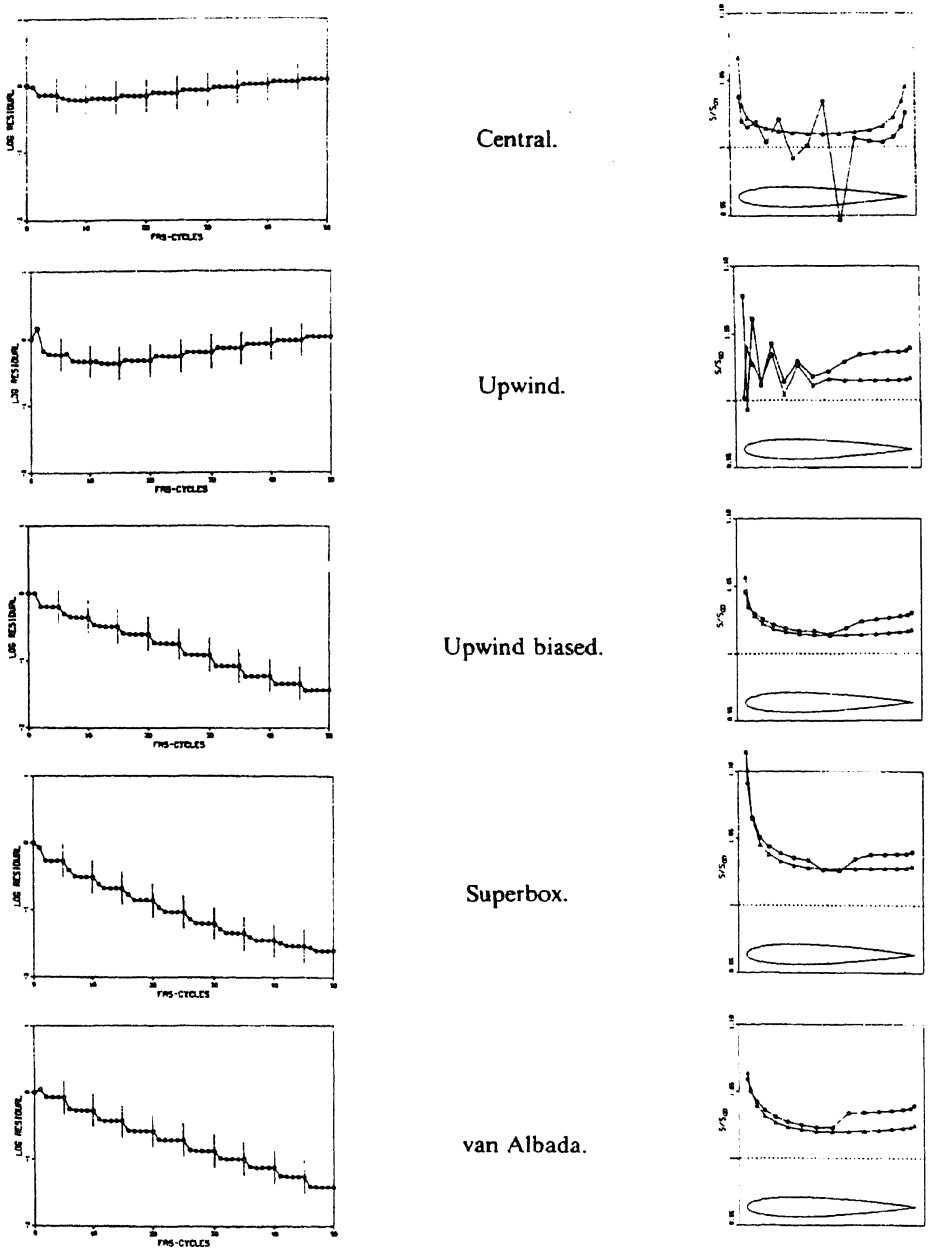
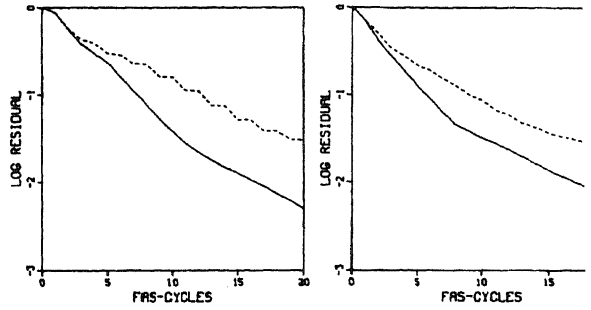


Figure 8.4: 32×16 grid for the NA

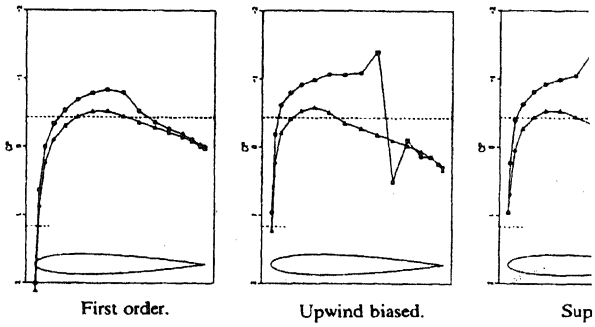


Convergence histories: FAS and DCP cycles. Entropy distributions, $s = p\rho^{-\gamma}$.

Figure 8.5: Convergence histories (left) and surface entropy distributions (right), on the 32×16 grid for the NACA0012 airfoil.



(a) Convergence histories: for 1 FAS-cycl
and for 2 FAS-cycles per DCP-



(b) Converged surface pressure

Figure 8.6: Results on the 32×16 grid for the NACA

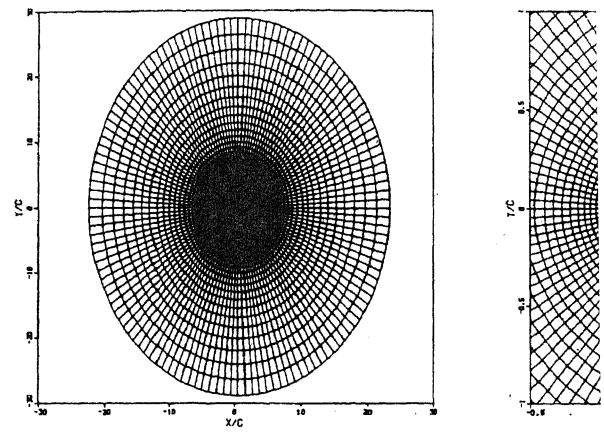
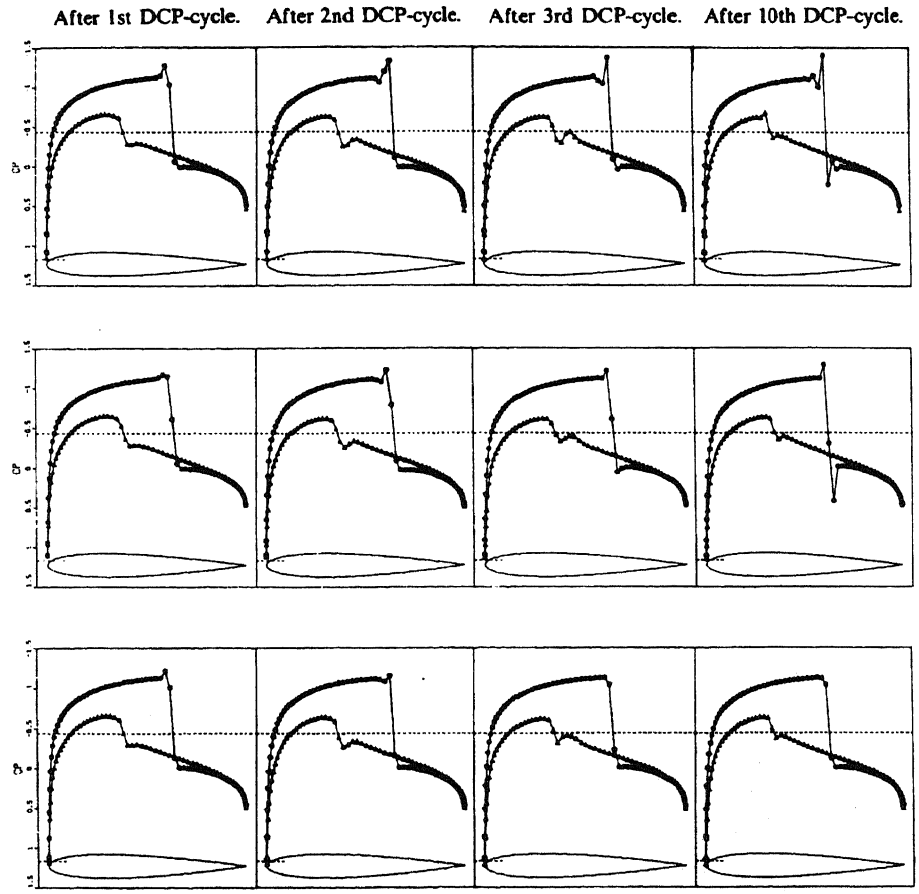


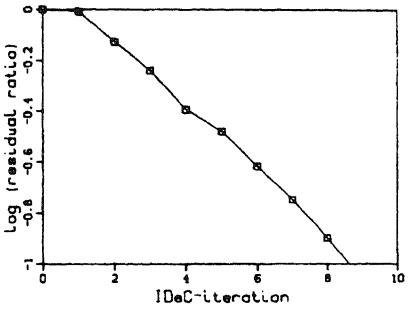
Figure 8.7: 128×64 grid for the N

Figure 8.8: First order results on the 128×64 grid, $M_\infty = 0.8$, $\alpha = 1.25^0$.

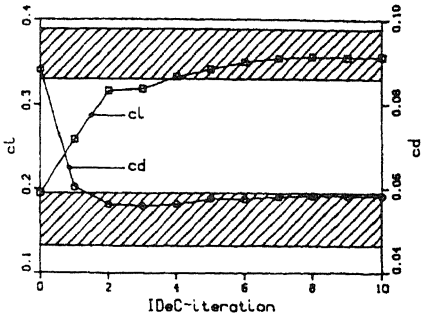


Upwind biased (top), superbox (middle) and Van Albada limiter (bottom).

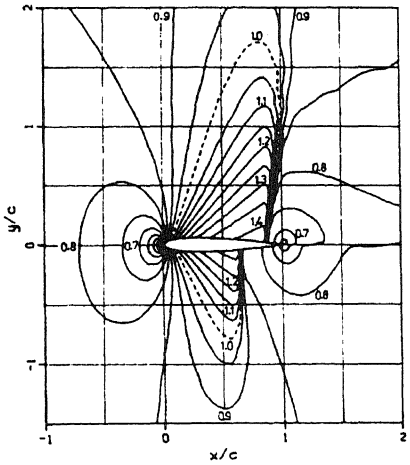
Figure 8.9: Second order results on the 128×64 grid, $M_\infty = 0.8$, $\alpha = 1.25^o$.

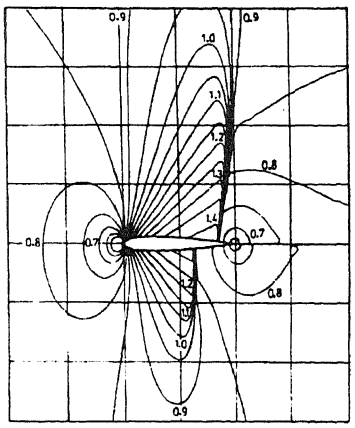


a. Convergence history residual ratio.

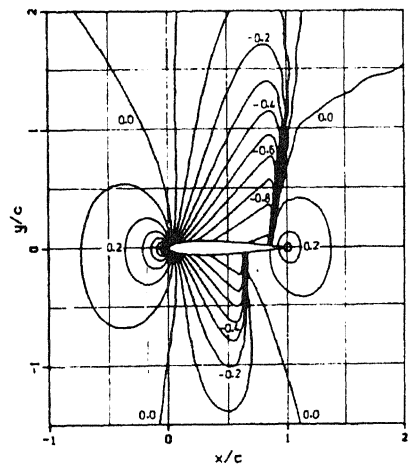


b. Convergence history lift and drag coefficient.

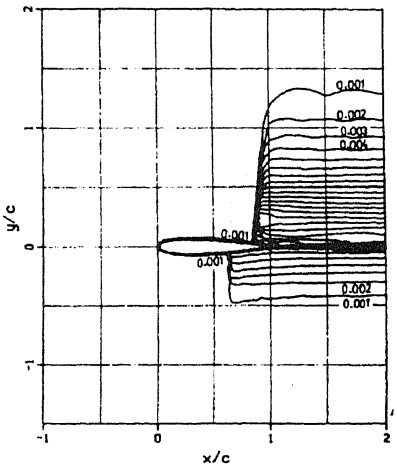




c. Mach number distributions; present result (left) and result Schmidt & Jameson (right).



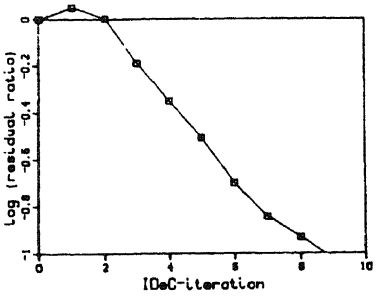
d. Present pressure distribution (c_p).



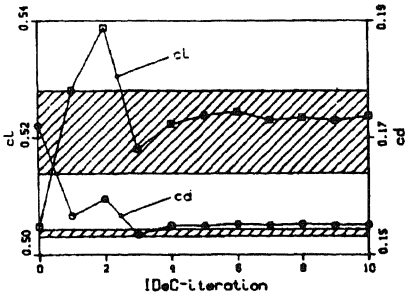
e. Present entropy distribution ($s/s_\infty - 1$).

Concergence history and solution components on the 128×32 grid for the NACA0012 airfoil

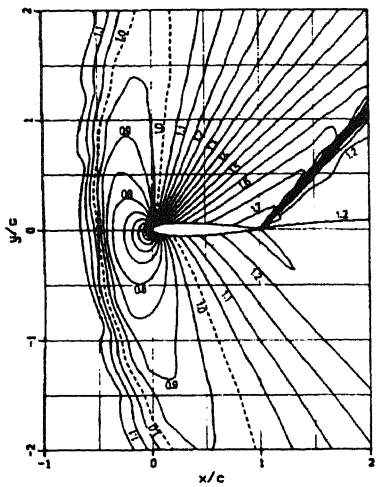
Figure 8.10: Results for transonic flow at $M_\infty = 0.85$, $\alpha = 1.0^0$.

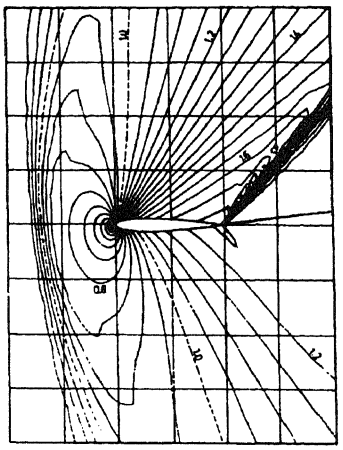


a. Convergence history residual ratio.

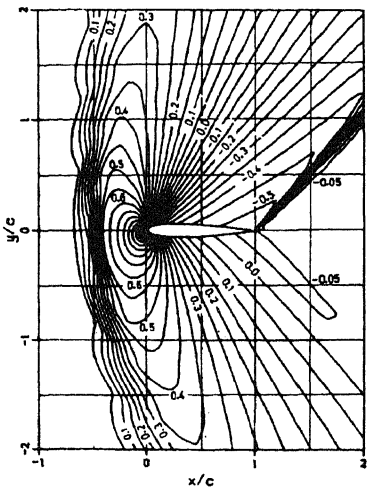


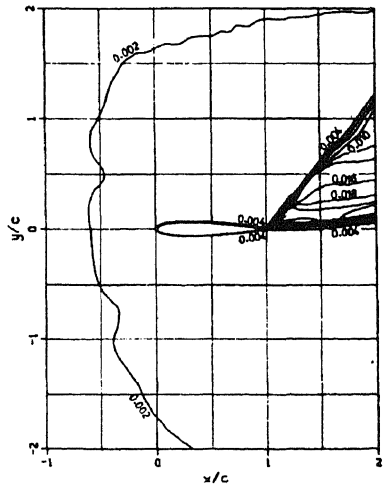
b. Convergence history lift and drag coefficient.





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



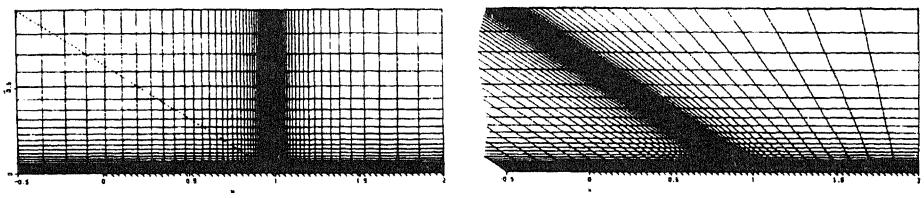


d. Present pressure distribution (c_p).

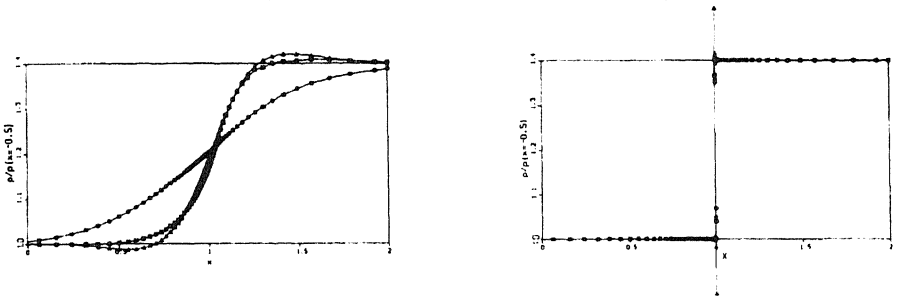
e. Present entropy distribution ($s/s_\infty - 1$).

Concergence history and solution components on the 128×32 grid for the NACA0012 airfoil

Figure 8.11: Results for supersonic flow at $M_\infty = 1.2$, $\alpha = 7^0$.

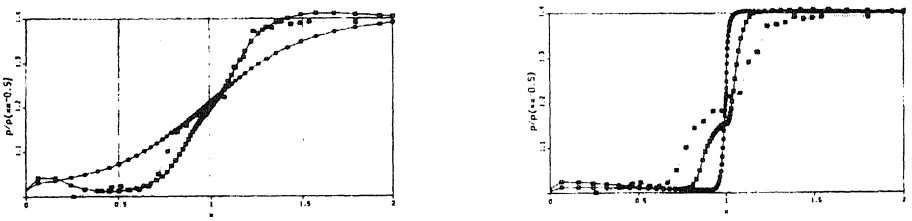


(a) Grids (rectangular and skew)



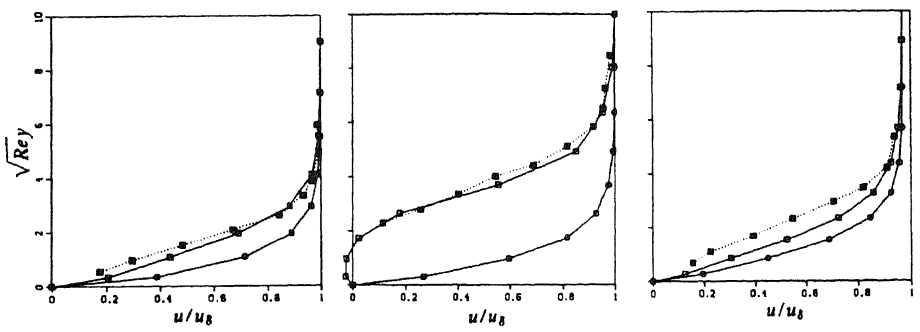
Inviscid distributions

(b) ◯: first order, △: non-limited second-order, □: limited second-order



(c) ◯: first order, □: limited second-order, ■ : measured

Figure 8.12: Grids and corresponding surface distributions



a. At x = 0.77 b. At x = 0.97 c. At x = 1.22

(c) (◯: first order, ■ (solid): limited second-order, ■ (dashed): measured)

Figure 8.13: Velocity profiles at $x = 0.5$, $x = 1.0$, and at $x = 1.5$.