stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 107/81 JUNI

W.J.A. MOL

# ON THE CHOICE OF SUITABLE OPERATORS AND PARAMETERS IN MULTIGRID METHODS

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 65N20, 65N99, 65F10

# On the choice of suitable operators and parameters in multigrid methods

by

W.J.A. Mol

ABSTRACT

In this report we consider multigrid methods for the solution of elliptic boundary value problems. These methods are described by a simple Algol-like program. By special choices of some operators and parameters almost every multigrid strategy that has been proposed in the literature for linear problems can be recovered. Several possibilities for the restriction, prolongation, coarse grid and smoothing operators are considered. Furthermore, we consider the number of smoothing steps and the number of coarse grid corrections. Some comparitive experiments are described with the Poisson, anisotropic diffusion and the convection-diffusion equations.

KEY WORDS & PHRASES: *multigrid methods, prolongation, restriction, coarse grid operator, smoothing operator, approximate inverse, incomplete LU-decomposition, Galerkin approximation.*

CONTENTS

1. INTRODUCTION

2. MULTIGRID METHODS

2.1 Defect correction processes

2.2 The two-grid method

2.3 The multigrid method

2.4 The multigrid method of Frederickson

3. CHOICES OF PROLONGATION, RESTRICTION, COARSE GRID AND SMOOTHING OPERATORS

3.1 Prolongation and restriction operators

3.2 Coarse grid operators

3.3 Smoothing operators

4. COMPUTATIONAL COMPLEXITY OF MULTIGRID METHODS

4.1 Computational complexity of one multigrid iteration

4.2 Preliminary work

4.3 Efficiency of multigrid methods for the Poisson, anisotropic diffusion and convection-diffusion equation

4.4 Comparison with other authors

4.5 Final remarks

REFERENCES

# 1. INTRODUCTION

The numerical treatment of elliptic boundary value problems gives rise to the problem of how to solve large sparse systems of equations. Numerical methods for the solution of these systems can be divided into direct and iterative methods.

The most general direct method, Gauss-elimination, is not suitable to solve large sparse systems because its use of long computing time and large storage requirement. There are some variants which take advantage of the sparsity of the coefficient matrix of the problem. For instance, this can be done by rearranging rows and columns in the matrix (dissection method) or by making use of the Laplacian character of the problem (cyclic reduction and fast Fourier methods). The class of problems to which these methods can be applied is small. There are restrictions concerning the type of differential equations, boundary conditions and the shape of the region.

Many iterative methods are known: Jacobi, Gauss-Seidel, S.O.R. with several variants (by points, by lines, symmetric, etc.) and the A.D.I.-method. These methods are generally faster than direct methods especially if optimal parameters are known. They use little storage and the programming is simple. A still faster advanced iterative method developed by MEYERINK and v.d. VORST [13] is the ICCG-method, which is applicable if the system is symmetric. A non symmetric variant, described by WESSELING and SONNEVELD [19] (the PIDR-method) is equally fast.

It seems to be that iterative methods which are both fast and generally applicable are the multigrid methods. The inventor of these methods is FEDORENKO [6,7]. He described a multigrid method for the Poisson equation in a square, and he proved that the number of operations is O(N), with N the number of grid points. BAKHVALOV [2] gave a convergence proof of a second order boundary value problem with variable coefficients in a rectangle.

BRANDT [3,4,5] described a multigrid method similar to that of Fedorenko and Bakhvalov, and demonstrated its practical usefulness. Furthermore, he proposed ideas for adaptive discretization in certain parts of the region e.g. in the neighbourhood of singularities. ASTRACHANCEV [1] and NICOLAIDES [16,17] applied a multigrid method on finite element problems

and gave convergence proofs. Other convergence proofs and experiments are given by FREDERICHSON [8], HACKBUSCH [9,10], WESSELING [18,19,20] and MOL [14,15]. A survey of multigrid literature can be found in HEMKER [12].

From these proofs and many numerical experiments it can be concluded that multigrid methods need a number of operations of $0(N)$ for the solution of a very large class of linear and nonlinear elliptic boundary value problems whereas for other numerical methods this number is $0(N^{\alpha})$ with $\alpha > 1$, for a fixed accuracy.

A survey of various numerical methods is given in table 1.1. The last column gives the number of operations when the method is applied on a 2nd order elliptic boundary value problem. The estimate of the operations count of the PIDR method has a note of interrogation, because a rigorous theoretical estimate for this method is not available at the moment.

| Method | Year of appearance | c: coefficients are constant | s.a: systems self adjoint | number of operations |
|---|---|---|---|---|
| Gauss-elimination | < 1850 | - | - | $0(N^2)$ |
| S.O.R | 1954 | c | s.a | $0(N^{3/2})$ |
| A.D.I. | 1955 | c | s.a | $0(N \log N)$ |
| Reduktion/Fourier methods | 1965 | c | s.a | $0(N \log N)$ |
| ICCG | 1977 | c | s.a | $0(N^{5/4})$ |
| PIDR | 1910 | - | - | $0(N^{5/4})$? |
| Multigrid | 1962 | - | - | $0(N)$ |

Table 1.1 Number of operations for various methods.

A difficulty with multigrid methods is that there are many ways in which the basic ideas underlying these methods can be implemented. In this report a multigrid method is presented which is

- more efficient than other multigrid methods for which numerical experiments have been reported in sufficient detail.
- robust. That is, the method can be applied to a large variety of problems including singularly perturbed problems without adaptation of the multigrid method to the problem at hand.

In chapter 2 we give a description of a framework into which we can fit a large class of multigrid methods.

In chapter 3 we describe various possibilities for restriction, prolongation, coarse grid and smoothing operators.

In chapter 4 a comparison will be made between different multigrid methods applied to the Poisson, anisotropic diffusion and convection-diffusion equations. Furthermore, we compare the efficiency of our method with the efficiency reported in other publications.

## 2. MULTIGRID METHODS

### 2.1 Defect correction processes

Defect correction processes are general iterative processes for the solution of operator equations. Many well-known iterative processes can be classified into this category and among these are the multigrid methods (see HEMKER [11]). Here we consider only linear systems of equations which originate from the discretization of a 2nd order elliptic boundary value problem.

Consider a system of equations denoted by:

$$Au = f, \tag{2.1.1}$$

with A an $N \times N$ non-singular matrix and u and f N-vectors. The following defect correction process (DCP) will be considered:

$$\begin{aligned} u^{(1)} &= B^{(0)} f, \\ u^{(\nu+1)} &= G^{(\nu)} u^{(\nu)} + B^{(\nu)} f. \qquad \nu = 1,2,\ldots \end{aligned} \tag{2.1.2}$$

$B^{(\nu)}$ is called the approximate inverse of A and $G^{(\nu)}$ the amplification matrix. $G^{(\nu)}$ is defined by

(2.1.3) $\quad G^{(\nu)} = I - B^{(\nu)}A.$

If $B^{(\nu)} = B$ for all $\nu$ the process is called stationary.

EXAMPLE. Let A be decomposed as $A = \tilde{L} + \tilde{D} + \tilde{U}$ where $\tilde{U}$ is a strict upper and $\tilde{L}$ a strict lower triangular and $\tilde{D}$ a diagonal matrix. The Gauss-Seidel process reads as follows:

(2.1.4) $\quad (\tilde{L} + \tilde{D})u^{(\nu+1)} = -\tilde{U}u^{(\nu)} + f.$

This is a stationary DCP with $B^{(\nu)} = B = (\tilde{L} + \tilde{D})^{-1}$.

## 2.2 The two-grid method.

The two-grid method is a non-stationary DCP in which two different approximate inverses are used:

- Some relaxation method (e.g. Jacobi, Gauss-Seidel) on the fine grid damping short wavelength fluctuations in the residual $r^{(\nu)} = f - Au^{(\nu)}$.
- A coarse grid correction damping the long wavelength fluctuations in the residual $r^{(\nu)}$.

Suppose (2.1.1) is a system of equations belonging to a boundary value problem which is discretized on a uniform grid $\Omega^{\ell}$. For convenience we assume:

(2.2.1) $\quad \Omega^{\ell} = \{(x_1,x_2) \mid x_i = m_i.2^{-\ell},\ m_i = 0(1)2^{\ell},\ i = 1,2\}.$

A corresponding set of grid functions is defined by

(2.2.2) $\quad U^{\ell} = \{u^{\ell}: \Omega^{\ell} \to \mathbb{R}\}.$

System (2.1.1) is denoted now by

(2.2.3) $\quad A^{\ell}u^{\ell} = f^{\ell},$

with $A^{\ell}:U^{\ell} \to U^{\ell}$.

A two-grid method uses an analogue of (2.2.3) on a coarse grid $\Omega^{\ell-1}$ with mesh size $2^{-(\ell-1)}$:

$$A^{\ell-1}u^{\ell-1} = f^{\ell-1}. \tag{2.2.4}$$

Let be given a restriction operator $R^{\ell}$ and a prolongation operator $P^{\ell}$:

$$R^{\ell}:U^{\ell} \to U^{\ell-1} \qquad P^{\ell}:U^{\ell-1} \to U^{\ell}. \tag{2.2.5}$$

A coarse grid correction step in the two-grid method is defined by:

$$u^{\ell} := u^{\ell} + P^{\ell}(A^{\ell-1})^{-1}R^{\ell}(f^{\ell}-A^{\ell}u^{\ell}). \tag{2.2.6}$$

We omit the iteration index $\nu$ if no confusion is possible. One step in the two-grid method consists of $\rho$ sweeps with the relaxation method, a coarse grid correction step and $\tau$ sweeps with the relaxation method. This can be described in quasi-Algol as follows:

```
procedure  two-grid method (A^l, u^l, f^l, ρ, τ, P^l, R^l);
real array  A^l, u^l, f^l; integer ρ, τ; integer procedure P^l, R^l;
begin      integer n;
           for n:= 1(1)ρ  do  u^l := G^l u^l + B^l f^l;
           f^(l-1) := R^l (f^l - A^l u^l);
           u^(l-1) := (A^(l-1))^-1 f^(l-1);
           u^l := u^l + P^l u^(l-1);
           for n:= 1(1)τ do u^l := G^l u^l + B^l f^l;
end two-grid method;
```

Note that $f^{\ell-1}$ is a coarse grid approximation not to $f^{\ell}$ but to the residual $f^{\ell}-A^{\ell}u^{\ell}$.

The amplification matrix $G_2^{\ell}$ of one step of the two-grid method is

$$G_2^{\ell} = (G^{\ell})^{\tau}((A^{\ell})^{-1} - P^{\ell}(A^{\ell-1})^{-1}\ R^{\ell})(\hat{G}^{\ell})^{\rho}\ A^{\ell}, \tag{2.2.7}$$

with $G^{\ell} = I^{\ell} - B^{\ell}\ A^{\ell}$ and $\hat{G}^{\ell} = I^{\ell} - A^{\ell}\ B^{\ell}$ the amplification matrices of the

relaxation process. $(A^{\ell})^{-1} - P^{\ell}(A^{\ell-1})^{-1}R^{\ell}$ is called the relative convergence matrix.

Several authors, e.g. HACKBUSCH [10] and WESSELING [18], has shown that under certain assumptions $\|G_2^{\ell}\| \leq c < 1$ with $\|\ \|$ a suitable norm and c independent of mesh size $2^{-\ell}$. The two-grid method is completely determined by the discretizations $A^{\ell}$ and $A^{\ell-1}$, the restriction $R^{\ell}$, prolongation $P^{\ell}$, a relaxation method corresponding with an approximate inverse $B^{\ell}$ and the number of relaxation steps $\rho$ and $\tau$.

## 2.3. The multigrid method

The multigrid method makes use of a hierarchy of computational grids $\Omega^k$ and corresponding sets of grid functions $U^k$, $k = \ell-1(-1)1$ defined by (2.2.1) and (2.2.2) with $\ell$ replaced by k. The mesh size of $\Omega^k$ is $2^{-k}$, hence the grids $\Omega^k$ gets coarser as k gets smaller.

In the two-grid method we have to solve problem (2.2.4) on the coarse grid. The multigrid method approximates the solution $u^{\ell-1}$ of this problem by application of $\sigma$ iteration steps of the same two-grid method on the coarse level, and so on. On the coarsest grid the problem is solved exactly or approximately by some iterative method. For simplicity we will assume that the coarse problem is solved exactly. In quasi-Algol the multigrid method thus obtained is described by:

```
procedure multigrid method (k,A^k,u^k,f^k,ρ,σ,τ,P^k,R^k);
        real array A^k,u^k,f^k;
        integer procedure P^k,R^k; value k; integer k,ρ,σ,τ;
        begin integer n;
```

```
if k=1 then u^l := (A^l)^-1 f^l
else
begin for n:= 1(1)ρ do u^k := G^k u^k + B^k f^k;
      f^(k-1) := R^k (f^k - A^k u^k);
      u^(k-1) := 0;
      for n:= 1(1)σ do multigrid method (k-1, A^(k-1), u^(k-1), f^(k-1), ρ, σ, τ, P^(k-1), R^(k-1));
      u^k := u^k + P^k u^(k-1);
      for n:= 1(1)τ do u:= G^k u^k + B^k f^k;
end;
end;
```

This is the linear variant of the multigrid method. With some modification we get the non-linear multigrid method (see HEMKER [11]). BRANDT [4] calls the linear variant the CS-algorithm and the non-linear variant the FAS-algorithm. It is also possible to apply the multigrid method on increasingly finer grids, with the solution on a coarser grid as initial estimate of the solution on a finer grid. This is called the full multigrid algorithm.

Furthermore, it is possible to add conditional statements in the interation-loops taking the number of iterations dependent on the rate of convergence or other conditions that can be checked during the computation. Multigrid methods that make use of this possibility are called adaptive methods. Although BRANDT [3,4,5] shows that this strategy enhances the efficiency, we will use the algorithm with fixed $\rho,\sigma$, and $\tau$ because this strategy is more accessible for theoretical analysis and because we have found that with a fixed strategy the efficiency is already very good for a large variety of problems.

The amplification matrix of the multigrid algorithm $G_m^\ell$ described above is given by:

$$G_m^\ell = G_2^\ell + (G^\ell)^\tau P^\ell (G_m^{\ell-1})^\sigma (A^{\ell-1})^{-1} R^\ell (\hat{G}^\ell)^\rho A^\ell. \qquad (2.3.1)$$

If the same assumptions of the two-grid method and some additional conditions for $P^k$ and $R^k$ hold then it can be derived that

(2.3.2) $$\|G_m^{\ell}\| \leq \|G_2^{\ell}\| + C\|G_m^{\ell-1}\|^{\sigma}.$$

With $\|G_2^{\ell}\| \leq c < 1$ on each level it is possible to find a $\sigma$ such that $\|G_m^{\ell}\| \leq c < 1$. Often a small value of $\sigma$ (e.g. $\sigma=2$) can be shown to be sufficient to obtain $\|G_m^{\ell}\| \leq c < 1$ on all levels. Thus an upper bound for the norm of the amplification matrix of the multigrid method can be found which is strictly separated from 1 for all mesh sizes $h = 2^{-\ell}$ of the fine grid. Other iterative methods have $\|G\| \to 1$ for $h \to 0$.

## 2.4. The multigrid method of Frederickson

When $\rho=0$ and $\sigma=\tau=1$ the multigrid method can be written in a simple way as follows:

$r^{\ell} := f^{\ell} - A^{\ell}u^{\ell}$;
for $k := \ell(-1)2$ do $r^{k-1} := R^k r^k$;
$e^1 := (A^1)^{-1} r^1$;
for $k := 2(1)\ell$ do
begin $e^k := P^k e^{k-1}$;
$e^k := G^k e^k + B^k r^k$;
end
$u^{\ell} := u^{\ell} + e^{\ell}$;

FREDERICKSON [8] used this method for the Poisson equation. It has been generalized to general elliptic equations including the Navier-Stokes equations by WESSELING [19] and MOL [14,15].

The amplification matrix of this algorithm is rather simple. For instance for a 4-grid method it is given by:

$$\begin{aligned} G_m^4 = {} & (I^4 - B^4A^4)\{(A^4)^{-1} - P^4(A^3)^{-1}R^4\}A^4 + \\ & + (I^4 - B^4A^4)P^4(I^3 - B^3A^3)\{(A^3)^{-1} - P^3(A^2)^{-1}R^3\}R^4A^4 + \\ & + (I^4 - B^4A^4)P^4(I^3 - B^3A^3)P^3(I^2 - B^2A^2)\{(A^2)^{-1} - P^2(A^1)^{-1}R^2\}R^3R^4A^4. \end{aligned} \tag{2.4.1}$$

It can be shown that every amplification matrix and relative convergence matrix reduces a particular range of wavelengths in the residual. $I^4-B^4A^4$ reduces the smallest and $(A^2)^{-1} - P^2(A^1)^{-1}R^2$ the largest wavelengths (see HEMKER [11]).

## 3. CHOICES OF PROLONGATION, RESTRICTION, COARSE GRID AND SMOOTHING OPERATORS

In this chapter various alternatives for the aforementioned operators are described, together with a suitable data structure. This data structure has been used earlier by FREDERICKSON [8].

### 3.1. Prolongation and restriction operators.

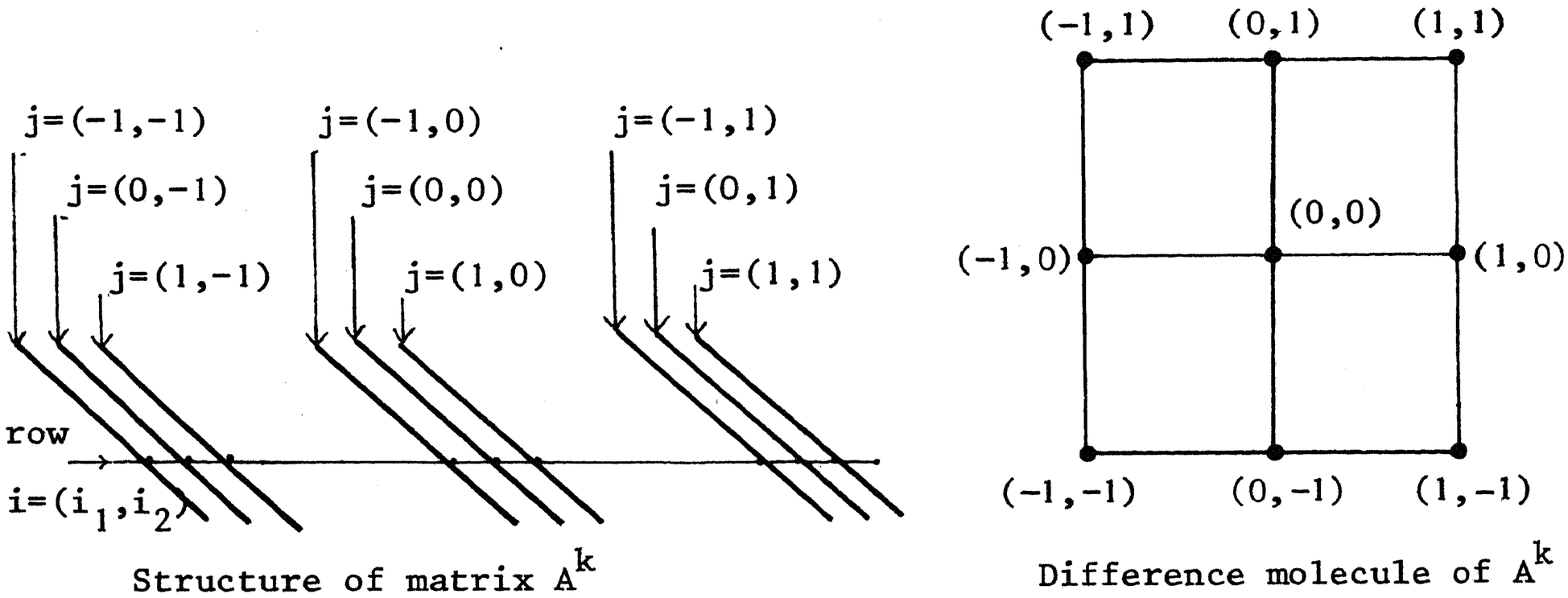
To enumerate the grid-points of $\Omega^k$ we define the following set of ordered pairs $i = (i_1,i_2)$:

$$(3.1.1) \qquad N^k = \{ i = (i_1,i_2) \in Z \mid i_{1,2} = 0(1)2^k\}, \qquad Z = \mathbb{Z} \times \mathbb{Z}.$$

A sparse matrix $A^k$ can be represented by difference molecules as follows: every molecule corresponds with a row in matrix $A^k$ (which corresponds with a point $i = (i_1,i_2) \in N^k$) and every point of the molecule corresponds with an element in this row.

When $A^k$ is a 9-point operator (i.e. $A^k$ has 9 diagonals) then the points of the difference molecule are enumerated by

$$(3.1.2) \qquad J = \{ j = (j_1,j_2) \in Z \mid j_{1,2} = 0, \pm 1\}.$$




Structure of matrix $A^k$ Difference molecule of $A^k$

Figure 3.1.1

The elements of the matrix $A^k$ are denoted by $A^k_{ij}$ with $i \in N^k$ and $j \in J$. With this notation matrix-vector multiplication is defined by

$$(A^k u^k)_i = \sum_{j \in J} A^k_{ij}\, u^k_{i+j}, \qquad i \in N^k. \tag{3.1.3}$$

Introduce the weighting-operator $t: Z \to \mathbb{R}$. Prolongation $P^k: U^{k-1} \to U^k$ and restriction $R^k: U^k \to U^{k-1}$ are defined as follows:

$$\begin{aligned} (P^k u^{k-1})_i &= 4 \sum_{j \in Z} t_{i-2j}\, u^{k-1}_j, \qquad i \in N^k, \\ (R^k u^k)_i &= \sum_{j \in Z} t_j\, u^k_{2i+j}, \qquad i \in N^{k-1}. \end{aligned} \tag{3.1.4}$$

Some possibilities for $t_j$ are given in the following table:

| $j$ | (0,0) | (1,0) | (0,1) | (-1,0) | (0,-1) | (-1,1) | (1,-1) | (1,1) | (-1,-1) | $\notin J$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 point $t_j$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 point $t_j$ | 1/2 | 1/8 | 1/8 | 1/8 | 1/8 | 0 | 0 | 0 | 0 | 0 |
| 7 point $t_j$ | 1/4 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 0 | 0 | 0 |
| 9 point $t_j$ | 1/4 | 1/8 | 1/8 | 1/8 | 1/8 | 1/16 | 1/16 | 1/16 | 1/16 | 0 |

Table 3.1.1 Choices of the weighting operator $t_j$.

The 1 and 5 point prolongations are excluded, because these are zero order interpolations.

We only consider the following possibilities for the prolongation

*7-point prolongation*

$$\begin{aligned} &(P^k u^{k-1})_{2i_1,2i_2} = u^{k-1}_{i_1,i_2}; \qquad (P^k u^{k-1})_{2i_1+1,2i_2} = \tfrac{1}{2}(u^{k-1}_{i_1,i_2} + u^{k-1}_{i_1+1,i_2}), \\ &(P^k u^{k-1})_{2i_1,2i_2+1} = \tfrac{1}{2}(u^{k-1}_{i_1,i_2} + u^{k-1}_{i_1,i_2+1}), \\ &(P^k u^{k-1})_{2i_1+1,2i_2+1} = \tfrac{1}{2}(u^{k-1}_{i_1+1,i_2} + u^{k-1}_{i_1,i_2+1}). \end{aligned} \tag{3.1.5}$$

*9-point prolongation*

as (3.1.5) except

$$(3.1.6)\qquad (P^k u^{k-1})_{2i_1+1,2i_2+1} = \frac{1}{4}(u^{k-1}_{i_1,i_2} + u^{k-1}_{i_1+1,i_2} + u^{k-1}_{i_1,i_2+1} + u^{k-1}_{i_1+1,i_2+1}).$$

For the restriction $R^k$ we consider

*1-point restriction (or injection)*

$$(3.1.7)\qquad (R^k u^k)_{i_1,i_2} = u^k_{2i_1,2i_2}.$$

*5-point restriction*

$$(3.1.8)\qquad (R^k u^k)_{i_1,i_2} =$$

$$= \frac{1}{2}u^k_{2i_1,2i_2} + \frac{1}{8}(u^k_{2i_1+1,2i_2} + u^k_{2i_1-1,2i_2+1} + u^k_{2i_1,2i_2+1} + u^k_{2i_1,2i_2-1}).$$

*7-point restriction*

$$(3.1.9)\qquad (R^k u^k)_{i_1,i_2} = \frac{1}{4}u^k_{2i_1,2i_2} + \frac{1}{8}(u^k_{2i_1+1,2i_2} + u^k_{2i_1,2i_2+1} + u^k_{2i_1-1,2i_2} +$$

$$+ u^k_{2i_1,2i_2-1} + u^k_{2i_1-1,2i_2+1} + u^k_{2i_1+1,2i_2-1}).$$

*9-point restriction*

$$(3.1.10)\qquad (R^k u^k)_{i_1,i_2} = \frac{1}{4}u^k_{2i_1,2i_2} + \frac{1}{8}(u^k_{2i_1+1,2i_2} + u^k_{2i_1,2i_2+1} + u^k_{2i_1-1,2i_2} +$$

$$+ u^k_{2i_1,2i_2-1}) +$$

$$+ \frac{1}{16}(u^k_{2i_1+1,2i_2+1} + u^k_{2i_1-1,2i_2+1} + u^k_{2i_1+1,2i_2-1} +$$

$$+ u^k_{2i_1-1,2i_2-1}).$$

On $U^k$ the following inner product is defined

$$(3.1.11) \qquad (u^k, v^k)_k = 4^{-k} \sum_i u_i^k v_i^k$$

Between 7-point and 9-point prolongation and restriction we have the special relation

$$(3.1.12) \qquad (R^k u^k, v^{k-1})_{k-1} = (u^k, P^k v^{k-1})_k, \quad \forall\, u^k \in U^k, \ \forall\, v^{k-1} \in U^{k-1}.$$

Thus in those cases the restriction is the adjoint of the prolongation

$$(3.1.13) \qquad R^k = (P^k)^T.$$

## 3.2. Coarse grid operators.

For the coarse grid operators $A^k$, $k < \ell$ two possibilities will be studied

(3.2.1) $A^k$ is a finite difference (FD) approximation on $\Omega^k$,

for instance, $A^k$ is the approximation (2.2.3) with $\ell$ replaced by k and

$$(3.2.2) \qquad A^{k-1} = R^k A^k P^k.$$

If $R^k = (P^k)^T$ we call this Galerkin-approximation (see WESSELING [20]). The elements of $A^{k-1}$ can be computed in the following way. Suppose the weighting operator $t_j$ is associated with the prolongation $P^k$ and $t_j^*$ with the restriction $R^k$.

With (3.1.3), (3.1.4) and (3.2.2) we have:

$$(3.2.3)\qquad (A^{k-1}u^{k-1})_i = (R^k A^k P^k u^{k-1})_i = 4\,R^k A^k \sum_j t_{i-2j} u_j^{k-1} =$$

$$= 4R^k \sum_m A^k_{im} \sum_j t_{i+m-2j}\; u_j^{k-1} = 4 \sum_n t_n^* \sum_m A^k_{2i+n,m} \sum_j t_{2i+n+m-2j} u_j^{k-1}$$

$$= 4 \sum_{j=i+p} \sum_p u^{k-1}_{i+p} \sum_n \sum_m t_n^*\, A^k_{2i+n,m}\; t_{n+m-2p} =$$

$$= 4 \sum_{m=v+2p-n} \sum_p u^{k-1}_{i+p} \sum_n \sum_v t_n^*\, A^k_{2i+n,v+2p-n}\; t_v =$$

$$= 4 \sum_{\substack{p\to j\\ n\to u}} \sum_j u^{k-1}_{i+j} \sum_u \sum_v t_u^*\, A^k_{2i+u,v+2j-u}\; t_v \,.$$

On the other hand $(A^{k-1}\; u^{k-1})_i$ is equal to

$$(3.2.4)\qquad (A^{k-1}\; u^{k-1})_i = \sum_j A^{k-1}_{ij}\; u^{k-1}_{i+j} \,.$$

From the last two relations we deduce that

$$(3.2.5)\qquad A^{k-1}_{ij} = 4 \sum_{\substack{u\,v\\ u,v\in Z}} \sum t_u^*\, A^k_{2i+u,v+2j-u}\; t_v, \qquad \begin{matrix} i\in N^{k-1}\\ j\in J \end{matrix} \,.$$

EXAMPLE 1. Let $A^k$ be a 7-point Toeplitz-matrix with coefficients

$$A^k_{ij} = \sigma_j,\ j\in J \setminus \{(1,1),(-1,-1)\}.$$

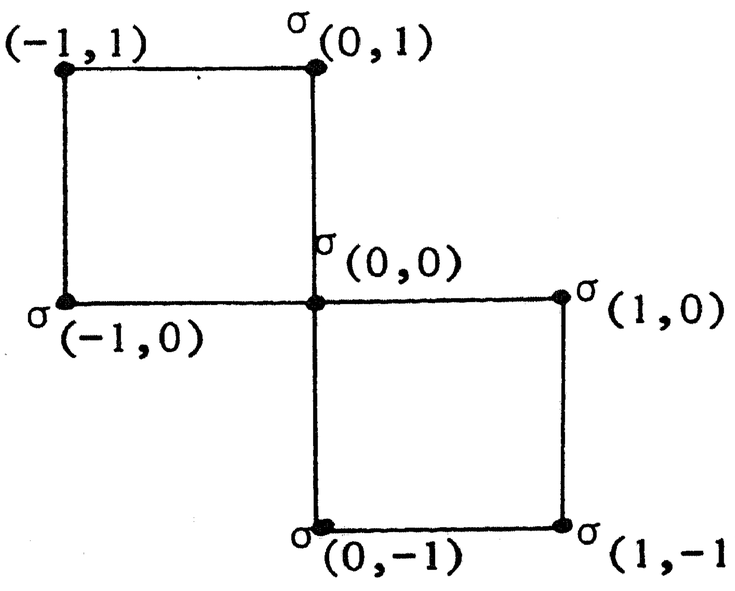



Figure 3.2.1. Difference molecule of $A^k$.

$R^k$ and $P^k$ are 7-point restriction and prolongation. Then $A^{k-1}$ is also a 7-point Toeplitz-matrix with elements $A^{k-1}_{ij} = \bar{\sigma}_j$, $j \in J \setminus \{(1,1),(-1,-1)\}$.

$$\bar{\sigma}_{(0,0)} = \frac{5}{8}\ \sigma_{(0,0)} + \frac{3}{8}\ (\sigma_{(1,0)} + \sigma_{(0,1)} + \sigma_{(-1,0)} + \sigma_{(0,-1)} + \sigma_{(1,-1)} + \sigma_{(-1,1)})$$

$$\bar{\sigma}_{(1,0)} = \frac{1}{16}\ \sigma_{(0,0)} + \frac{3}{8}\ \sigma_{(1,0)} + \frac{1}{8}\ (\sigma_{(0,1)} + \sigma_{(1,-1)})$$

$$\bar{\sigma}_{(0,1)} = \frac{1}{16}\ \sigma_{(0,0)} + \frac{3}{8}\ \sigma_{(0,1)} + \frac{1}{8}\ (\sigma_{(1,0)} + \sigma_{(-1,1)})$$

(3.2.6)
$$\bar{\sigma}_{(-1,0)} = \frac{1}{16}\ \sigma_{(0,0)} + \frac{3}{8}\ \sigma_{(-1,0)} + \frac{1}{8}\ (\sigma_{(0,-1)} + \sigma_{(-1,1)})$$

$$\bar{\sigma}_{(0,-1)} = \frac{1}{16}\ \sigma_{(0,0)} + \frac{3}{8}\ \sigma_{(0,-1)} + \frac{1}{8}\ (\sigma_{(-1,0)} + \sigma_{(1,-1)})$$

$$\bar{\sigma}_{(1,-1)} = \frac{1}{16}\ \sigma_{(0,0)} + \frac{3}{8}\ \sigma_{(1,-1)} + \frac{1}{8}\ (\sigma_{0,-1)} + \sigma_{(1,0)})$$

$$\bar{\sigma}_{(-1,1)} = \frac{1}{16}\ \sigma_{(0,0)} + \frac{3}{8}\ \sigma_{(-1,1)} + \frac{1}{8}\ (\sigma_{(0,1)} + \sigma_{(-1,0)})\ .$$

Note that when $A^k$ is a 5-point operator, $A^{k-1}$ is a 7-point operator in general.

An exceptional case is the Poisson equation. If $\sigma_{(0,0)} = 4$ and $\sigma_{(1,0)} = \sigma_{(0,1)} = \sigma_{(-1,0)} = \sigma_{(0,-1)} = -1$, $\sigma_{(1,-1)} = \sigma_{(-1,1)} = 0$ then $\bar{\sigma}_{(0,0)} = 1, \bar{\sigma}_{(1,0)} = \bar{\sigma}_{(0,1)} = \bar{\sigma}_{(1,0)} = \bar{\sigma}_{(0,-1)} = -\frac{1}{4}$, $\bar{\sigma}_{(1,-1)} = \bar{\sigma}_{(-1,1)} = 0$. Thus $A^{k-1}$ is equal to the finite difference approximation on $\Omega^{k-1}$.

EXAMPLE 2. Let $A^k$ be a 9-point Toeplitz-matrix with coefficients $A_{ij} = \sigma_j$, $j \in J$ and $R^k$ and $P^k$ are 9-point restriction and prolongation respectively. Then $A^{k-1}$ is also a 9-point Toeplitz matrix with elements $A^{k-1}_{ij} = \bar{\sigma}_j$, $j \in J$

$$\bar{\sigma}_{(0,0)} = \frac{9}{16}\ \sigma_{(0,0)} + \frac{3}{8}\ (\sigma_{(1,0)} + \sigma_{(0,1)} + \sigma_{(-1,0)} + \sigma_{(0,-1)}) +$$
$$+ \frac{1}{4}\ (\sigma_{(-1,-1)} + \sigma_{1,-1)} + \sigma_{(-1,1)} + \sigma_{(1,1)})$$

$$\bar{\sigma}_{(1,0)} = \frac{3}{32}\ \sigma_{(0,0)} + \frac{3}{8}\ \sigma_{(1,0)} + \frac{1}{16}\ (\sigma_{(0,1)} + \sigma_{(0,-1)}) +$$
$$+ \frac{1}{4}\ (\sigma_{(1,-1)} + \sigma_{(1,1)})$$

$$\bar{\sigma}_{(0,1)} = \frac{3}{32}\sigma_{(0,0)} + \frac{3}{8}\sigma_{(0,1)} + \frac{1}{16}(\sigma_{(1,0)} + \sigma_{(-1,0)}) +$$

$$+ \frac{1}{4}(\sigma_{(-1,1)} + \sigma_{(1,1)})$$

$$\bar{\sigma}_{(-1,0)} = \frac{3}{32}\sigma_{(0,0)} + \frac{3}{8}\sigma_{(-1,0)} + \frac{1}{16}(\sigma_{(0,1)} + \sigma_{(0,-1)}) +$$

$$+ \frac{1}{4}(\sigma_{(-1,-1)} + \sigma_{(-1,1)})$$

(3.2.7)
$$\bar{\sigma}_{(0,-1)} = \frac{3}{32}\sigma_{(0,0)} + \frac{3}{8}\sigma_{(0,-1)} + \frac{1}{16}(\sigma_{(1,0)} + \sigma_{(-1,0)}) +$$

$$+ \frac{1}{4}(\sigma_{(-1,-1)} + \sigma_{(1,-1)})$$

$$\bar{\sigma}_{(-1,-1)} = \frac{1}{64}\sigma_{(0,0)} + \frac{1}{4}\sigma_{(-1,-1)} + \frac{1}{16}(\sigma_{(-1,0)} + \sigma_{(0,-1)})$$

$$\bar{\sigma}_{(1,-1)} = \frac{1}{64}\sigma_{(0,0)} + \frac{1}{4}\sigma_{(1,-1)} + \frac{1}{16}(\sigma_{(1,0)} + \sigma_{(0,-1)})$$

$$\bar{\sigma}_{(-1,1)} = \frac{1}{64}\sigma_{(0,0)} + \frac{1}{4}\sigma_{(-1,1)} + \frac{1}{16}(\sigma_{(0,1)} + \sigma_{(-1,0)})$$

$$\bar{\sigma}_{(1,1)} = \frac{1}{64}\sigma_{(0,0)} + \frac{1}{4}\sigma_{(1,1)} + \frac{1}{16}(\sigma_{(1,0)} + \sigma_{(0,1)}).$$

In general, when $A^k$ is a 5- or 7-point operator, $A^{k-1}$ is a 9-point operator.

In the Poisson case we get

$$\bar{\sigma}_{(0,0)} = \frac{3}{4}; \quad \bar{\sigma}_{(1,0)} = \bar{\sigma}_{(0,1)} = \bar{\sigma}_{(-1,0)} = \bar{\sigma}_{(0,-1)} = -\frac{1}{8},$$

$$\bar{\sigma}_{(-1,-1)} = \bar{\sigma}_{(1,-1)} = \bar{\sigma}_{(-1,1)} = \bar{\sigma}_{(1,1)} = -\frac{1}{16}.$$

It is easy to prove that for $k \to \infty$ the limiting operator $A^1$ is equal to the Raleigh-Ritz-Galerkin discretization over bilinear splines of the Laplacian.

## 3.3. Smoothing operators.

We introduce two families of smoothing processes:

- DCP with $B^k$ the approximate inverse of $A^k$ as defined by FREDERICKSON [8]
- DCP with $B^k = (L^k U^k)^{-1}$ with $L^k, U^k$ the incomplete LU - decomposition of $A^k$.

### 3.3.1. The approximate inverse APINV.

FREDERICKSON [8] introduced a certain smoothing process for his multigrid method. The process is a stationary DCP. Consider the system of equations $A^k u^k = f^k$. We can write this as

$$(3.3.1.1) \qquad \sum_{m \in J} A^k_{im}\, u^k_{i+m} = f^k_i \,, \quad i \in N^k.$$

The inverse $H^k$ of $A^k$ is defined as

$$(3.3.1.2) \qquad \sum_{j \in Z} H^k_{ij}\, f^k_{i+j} = u^k_i, \quad i \in N^k.$$

Combining these two expressions we obtain

$$(3.3.1.3) \qquad \sum_j H^k_{ij} \sum_{m \in J} A^k_{i+j,m}\, u^k_{i+j+m} \underset{m=s-j}{=} \sum_j H^k_{ij} \sum_s A^k_{i+j,s-j}\, u^k_{i+s}$$
$$= \sum_s u^k_{i+s} \sum_j H^k_{ij}\, A^k_{i+j,s-j} = u^k_i.$$

From this we can compute the elements of the inverse $H^k$ from the following system of equations:

$$(3.3.1.4) \qquad \sum_{\substack{j \\ j \in Z}} H^k_{ij}\, A^k_{i+j,s-j} = \delta^k_{s,0}, \quad i \in N^k,\ s \in Z,$$

where $\delta$ denotes the Kronecker delta. These are $4^k$ systems with $4^k$ equations. Instead of an exact inverse we can compute an approximated inverse $B^k$:

$$(3.3.1.5) \qquad \sum_{\substack{j \\ j \in J}} B^k_{ij}\, A^k_{i+j,s-j} = \delta^k_{s,0}\,, \ i \in N^k\,, \ s \in J.$$

$$B^k_{ij} = 0\,, \ i \in N^k\,, \ j \notin J.$$

Now we have to solve $4^k$ systems of only 9 equations. We call this $B^k$ the 9-point APINV approximate inverse. In the same way, 1-point, 5-point or 7-point APINV approximate inverses can be constructed with $B^k_{ij} \neq 0$ for the following values of j

| approximate inverse | j for which $B^k_{ij} \neq 0$ |
|---|---|
| 1-point APINV | (0,0) |
| 5-point APINV | (0,0),(1,0),(-1,0),(0,1),(0,-1) |
| 7-point APINV | (0,0),(1,0),(-1,0),(0,1),(0,-1),(-1,1),(1,-1) |

Note that application of 1-point APINV gives rise to the Jacobi relaxation process.

EXAMPLE. If $A^k$ is a 9-point Toeplitz-matrix with coefficients $A^k_{ij} = \sigma_j, j \in J$ and $B^k$ is the 9-point APINV approximate inverse of $A^k$ with coefficients $B^k_{ij} = \tau_j$, $j \in J$, then the product $B^k A^k$ is a 25-point Toeplitz matrix with coefficients $\pi_j$, $j \in \{(j_1,j_2) | j_{1,2} = 0,\pm1,\pm2\}$. The coefficients $\pi_j$, $j \in J$, are defined as follows:

$$\pi_{(0,0)} = \sum_{\substack{j \\ j\in J}} \tau_j \ \sigma_{-j}$$

$$\pi_{(1,0)} = \tau_{(0,0)}\sigma_{(1,0)} + \tau_{(1,0)}\sigma_{(0,0)} + \tau_{(0,1)}\sigma_{(1,-1)} + \tau_{(0,-1)}\sigma_{(1,1)} + \tau_{(1,-1)}\sigma_{(0,1)} + \tau_{(1,1)}\sigma_{(0,-1)}$$

$$\pi_{(0,1)} = \tau_{(0,0)}\sigma_{(0,1)} + \tau_{(1,0)}\sigma_{(-1,1)} + \tau_{(0,1)}\sigma_{(0,0)} + \tau_{(-1,0)}\sigma_{(1,1)} + \tau_{(-1,1)}\sigma_{(1,0)} + \tau_{(1,1)}\sigma_{(-1,0)}$$

(3.3.1.6)

$$\pi_{(-1,0)} = \tau_{(0,0)}\sigma_{(-1,0)} + \tau_{(0,1)}\sigma_{(-1,1)} + \tau_{(-1,0)}\sigma_{(0,0)} + \tau_{(0,-1)}\sigma_{(-1,1)} + \tau_{(-1,-1)}\sigma_{(0,1)} + \tau_{(-1,1)}\sigma_{(0,-1)}$$

$$\pi_{(0,-1)} = \tau_{(0,0)}\sigma_{(0,-1)} + \tau_{(1,0)}\sigma_{(-1,-1)} + \tau_{(-1,0)}\sigma_{(1,-1)} + \tau_{(0,-1)}\sigma_{(0,0)} + \tau_{(-1,-1)}\sigma_{(1,0)} + \tau_{(1,-1)}\sigma_{(-1,0)}$$

$$\pi_{(1,-1)} = \tau_{(0,0)}\sigma_{(1,-1)} + \tau_{(1,0)}\sigma_{(0,-1)} + \tau_{(0,-1)}\sigma_{(1,0)} + \tau_{(1,-1)}\sigma_{(0,0)}$$

$$\pi_{(-1,1)} = \tau_{(0,0)}\sigma_{(-1,1)} + \tau_{(0,1)}\sigma_{(-1,0)} + \tau_{(-1,0)}\sigma_{(0,1)} + \tau_{(-1,1)}\sigma_{(0,0)}$$

$$\pi_{(-1,-1)} = \tau_{(0,0)}\sigma_{(-1,-1)} + \tau_{(-1,0)}\sigma_{(0,-1)} + \tau_{(0,-1)}\sigma_{(-1,0)} + \tau_{(-1,-1)}\sigma_{(0,0)}$$

$$\pi_{(1,1)} = \tau_{(0,0)}\sigma_{(1,1)} + \tau_{(1,0)}\sigma_{(0,1)} + \tau_{(0,1)}\sigma_{(1,0)} + \tau_{(1,1)}\sigma_{(0,0)}$$

The coefficients $B^{k}_{ij}$, $j \in J$ can be computed from the relations

(3.3.1.7) $\pi_{(0,0)} = 1; \quad \pi_j = 0, \quad j \in J \setminus (0,0)$.

The coefficients $\pi_j$, $j \in \{(j_1,j_2) | j_{1,2} = 0, \pm 1, \pm 2\} \setminus J$ are given by

$$\pi_{(2,0)} = \tau_{(1,0)}\sigma_{(1,0)} + \tau_{(1,-1)}\sigma_{(1,1)} + \tau_{(1,1)}\sigma_{(1,-1)}$$

$$\pi_{(0,2)} = \tau_{(0,1)}\sigma_{(0,1)} + \tau_{(1,1)}\sigma_{(-1,1)} + \tau_{(-1,1)}\sigma_{(1,1)}$$

$$\pi_{(-2,0)} = \tau_{(-1,0)}\sigma_{(-1,0)} + \tau_{(-1,-1)}\sigma_{(-1,1)} + \tau_{(-1,1)}\sigma_{(-1,-1)}$$

$$\pi_{(0,-2)} = \tau_{(0,-1)}\sigma_{(0,-1)} + \tau_{(1,-1)}\sigma_{(-1,-1)} + \tau_{(-1,-1)}\sigma_{(1,-1)}$$

(3.3.1.8) $\pi_{(2,1)} = \tau_{(1,0)}\sigma_{(1,1)} + \tau_{(1,1)}\sigma_{(1,0)}$

$$\pi_{(-1,2)} = \tau_{(0,1)}\sigma_{(-1,1)} + \tau_{(-1,1)}\sigma_{(0,1)}$$

$$\pi_{(1,2)} = \tau_{(0,1)}\sigma_{(1,1)} + \tau_{(1,1)}\sigma_{(1,0)}$$

$$\pi_{(-2,1)} = \tau_{(-1,0)}\sigma_{(-1,1)} + \tau_{(-1,1)}\sigma_{(-1,0)}$$

$$\pi_{(-2,-1)} = \tau_{(-1,0)}\sigma_{(-1,-1)} + \tau_{(-1,-1)}\sigma_{(-1,0)}$$

$$\pi_{(1,-2)} = \tau_{(0,-1)}\sigma_{(1,-1)} + \tau_{(1,-1)}\sigma_{(0,-1)}$$

$$\pi_{(-1,-2)} = \tau_{(0,-1)}\sigma_{(-1,-1)} + \tau_{(-1,-1)}\sigma_{(0,-1)}$$

$$\pi_{(2,-1)} = \tau_{(1,0)}\sigma_{(1,-1)} + \tau_{(1,-1)}\sigma_{(1,0)}$$

$$\pi_{(2,2)} = \tau_{(1,1)}\sigma_{(1,1)}; \quad \pi_{(-2,2)} = \tau_{(-1,1)}\sigma_{(-1,1)}$$

$$\pi_{(-2,-2)} = \tau_{(-1,-1)}\sigma_{(-1,-1)}; \quad \pi_{(2,-2)} = \tau_{(1,-1)}\sigma_{(1,-1)}.$$

End of example.

Assume

(3.3.1.9) $\quad B^k A^k = I + C^k$ ,

with $I^k$ the identity matrix and $C^k$ a rest matrix. In the given example $C^k$ is a 16-point Toeplitz matrix with the elements $\pi_j$ given in (3.3.1.8). The amplification matrix of the process corresponding with APINV approximate inverse is

(3.3.1.10) $\quad G^k = I^k - B^k A^k = - C^k.$

The smoothing process is defined by

(3.3.1.11) $\quad u^k = u^k + B^k(f^k - A^k u^k)$

or by

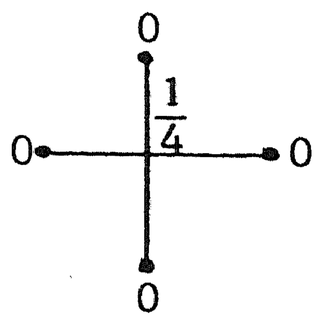
(3.3.1.12) $\quad u^k = G^k u^k + B^k f^k = - C^k u^k + B^k f^k.$

EXAMPLE: Poisson case: $\sigma_{(0,0)} = 4$, $\sigma_{(1,0)} = \sigma_{(-1,0)} = \sigma_{(0,1)} = \sigma_{(0,-1)} = 1$,

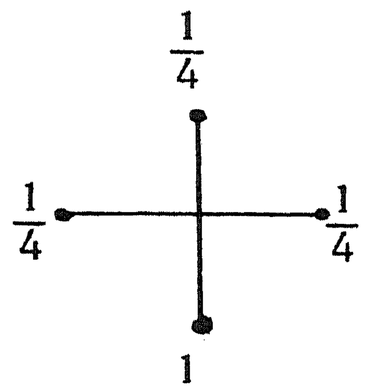
$\sigma_{(-1,-1)} = \sigma_{(1,-1)} = \sigma_{(-1,1)} = \sigma_{(1,1)} = 0.$

1-point APINV (Jacobi)

$B^k$:

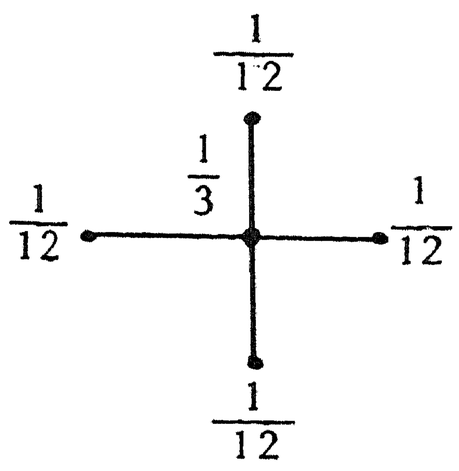



$G^k$:

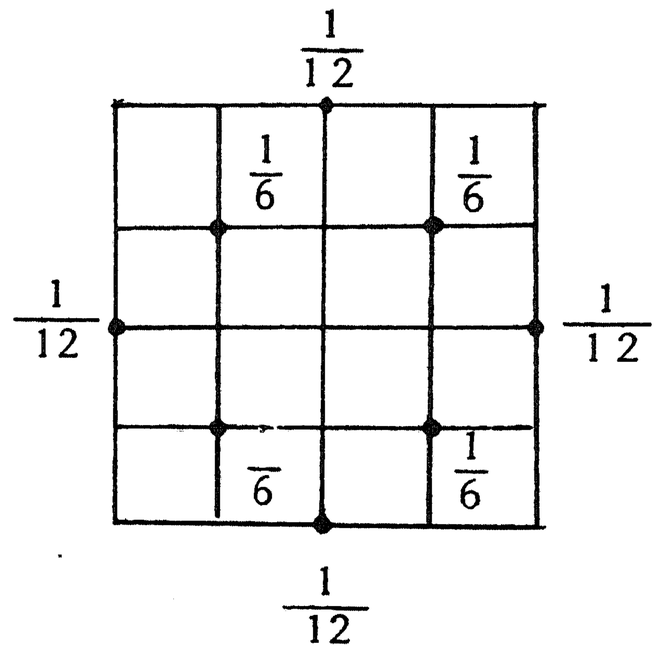



5-point APINV

$B^k$:

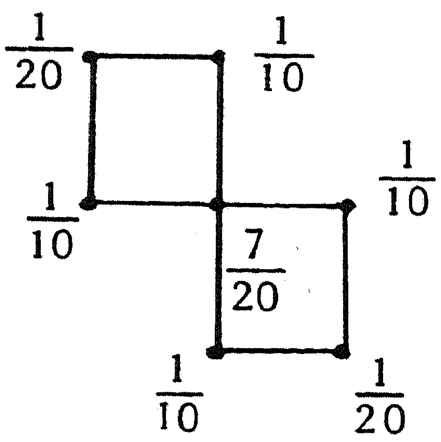



$G^k$:

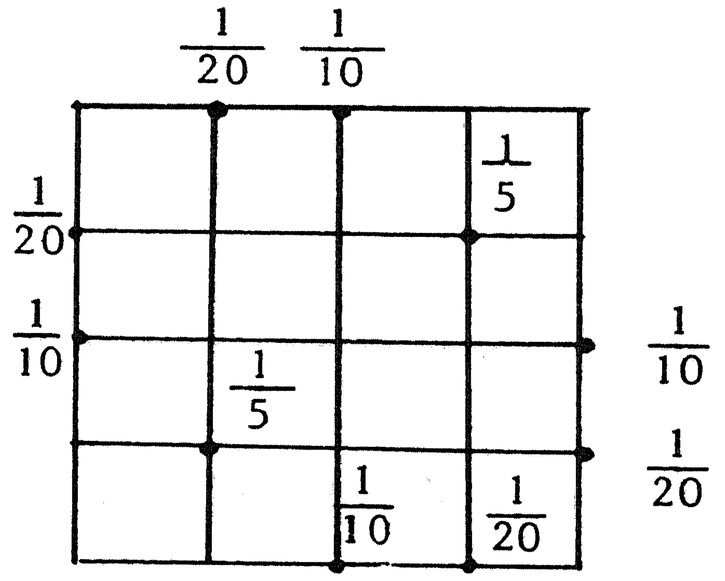



7-point APINV

$B^k$:

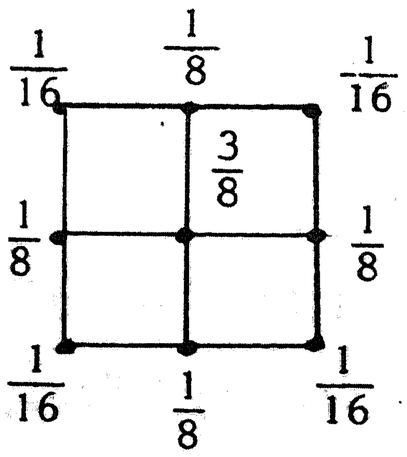



$G^k$:

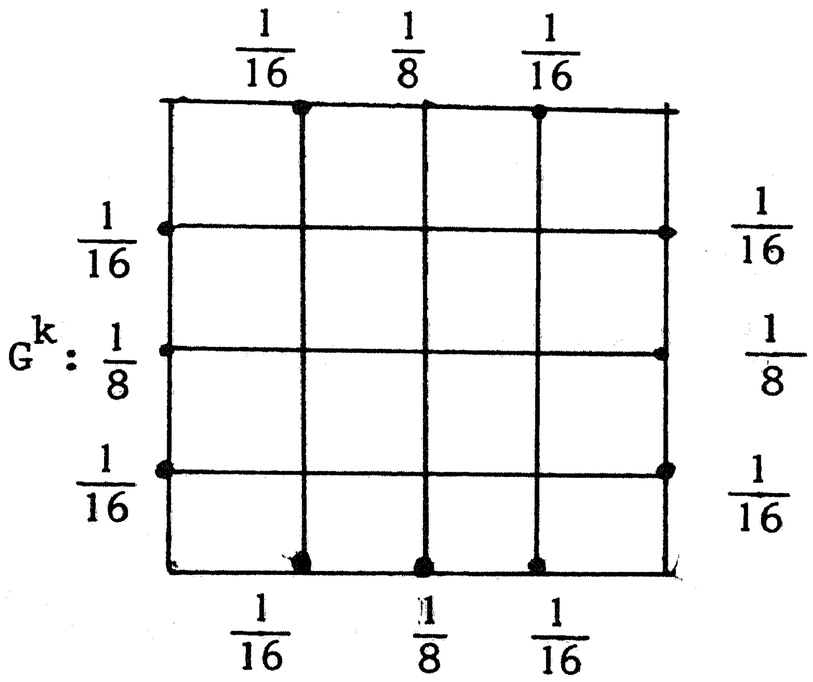



9-point APINV

$B^k$:




$G^k$:




End of example

Analogous to the Jacobi-relaxation process it is possible to introduce damping. The damped APINV-process is defined by

$$(3.3.1.13)\quad u^k = \omega(-C^k u^k + B^k f^k) + (1-\omega)u^k \quad , \quad 0 < \omega < 1.$$

3.3.2. Incomplete LU-decomposition.

MEYERINK and v.d. VORST [13] used the incomplete LU-decomposition (ILU) as preconditioning for a conjugate gradient process. WESSELING and SONNEVELD [19] introduced ILU as smoothing operator for multigrid methods.

Suppose $L^k$ is a lower triangular matrix and $U^k$ is an upper triangular matrix. $A^k$ is written as

$$(3.3.2.1)\quad A^k = L^k U^k - R^k,$$

with $R^k$ a rest matrix. Then we can define a DCP with approximate inverse

$$(3.3.2.2)\quad B^k = (L^k U^k)^{-1}.$$

The amplification matrix is

$$(3.3.2.3)\quad G^k = I^k - B^k A^k = (L^k U^k)^{-1} R^k.$$

The ILU-smoothing process is defined by

$$(3.3.2.4)\quad u^k = u^k + (L^k U^k)^{-1} (f^k - A^k u^k)$$

or by

$$(3.3.2.5)\quad u^k = (L^k U^k)^{-1} (R^k u^k + f^k).$$

The ILU-decomposition is defined as

$$(3.3.2.6)\quad \sum_{\substack{j\in J^- \\ \ell-j\in J^+}} L^k_{ij}\, U^k_{i+j,\ell-j} = A^k_{i,\ell}, \qquad \ell \in J,$$

with for $J^-$ and $J^+$ the following possibilities:

| | $J^-$ | $J^+$ |
|---|---|---|
| 5 point ILU | {(0,0),(0,-1),(-1,0)} | {(0,0),(1,0),(0,1)} |
| 7 point ILU | {(0,0),(0,-1),(1,-1),(0,-1)} | {(0,0),(1,0),(-1,1),(0,1)} |
| 9 point ILU | {(0,0),(0,-1),(1,-1),(0,-1),(-1,-1)} | {(0,0),(1,0),(-1,1),(0,1),(1,1)} |

EXAMPLE. If $L^k$ and $U^k$ are Toeplitz matrix corresponding to a 9-point ILU with coefficients $L^k_{ij} = \lambda_j$, and $U^k_{ij} = \mu_j$,

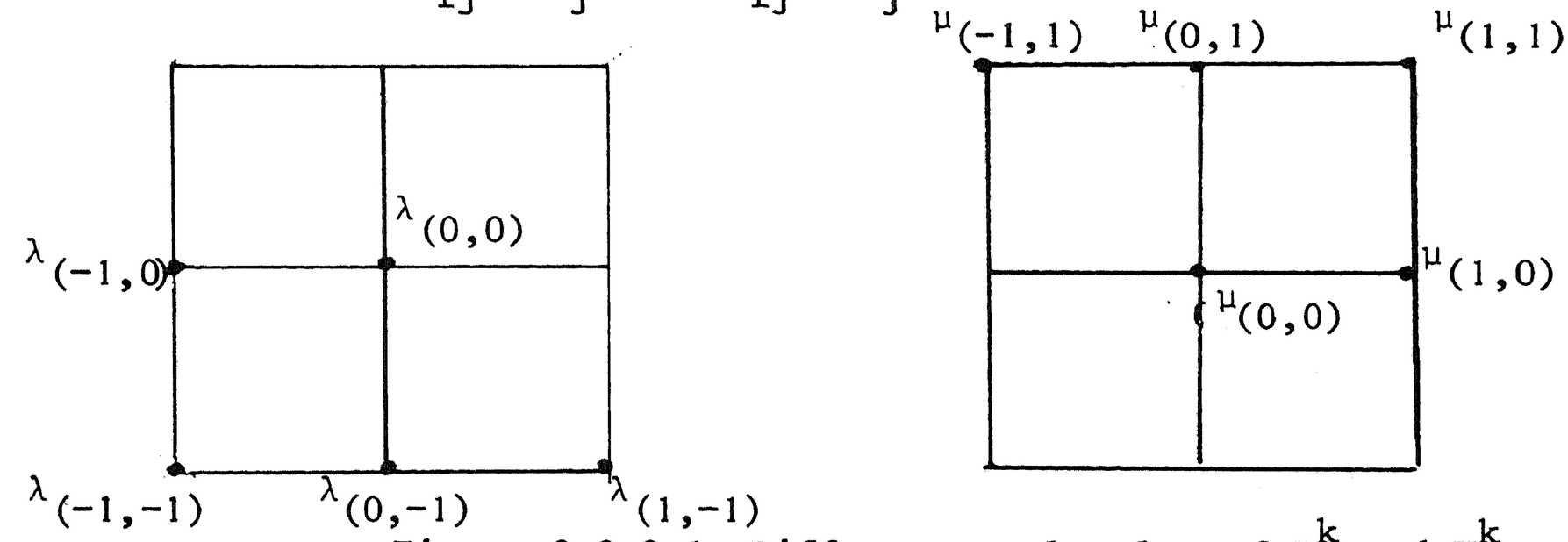



Figure 3.3.2.1. Difference molecules of $L^k$ and $U^k$

then the product $L^k\ U^k$ is a 13-point Toeplitz-matrix with coefficients $\rho_j$

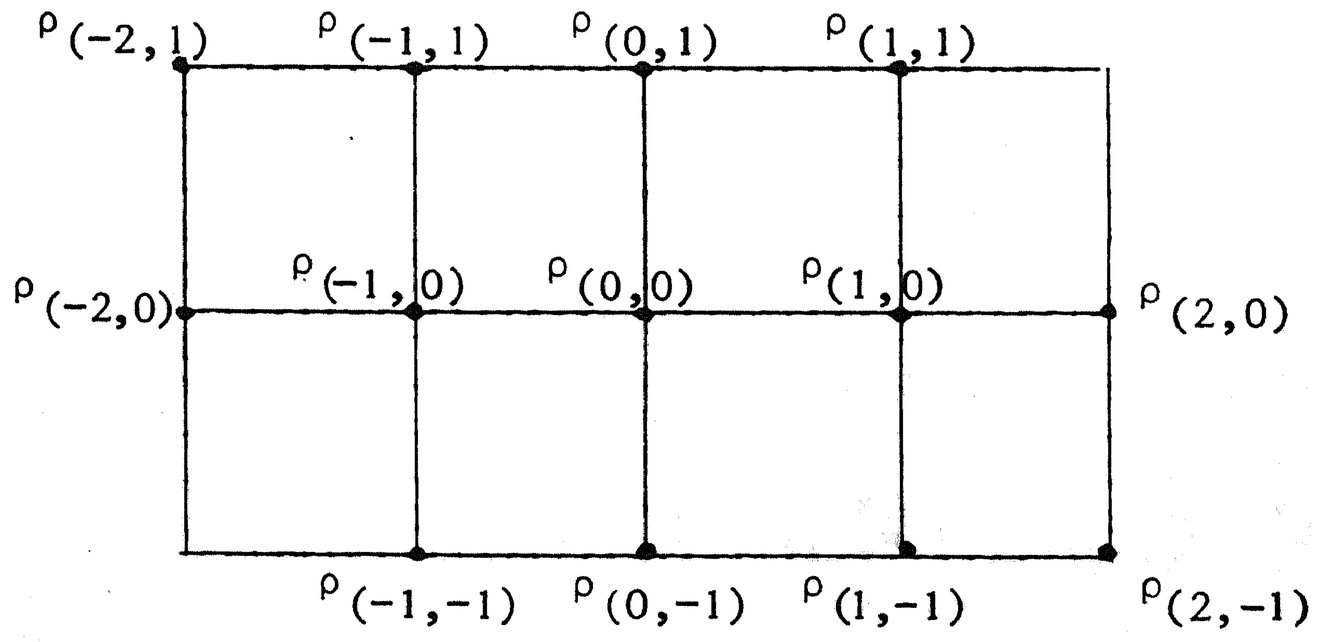



Figure 3.2.2.2. Difference molecule of $L^k * U^k$.

The coefficients $\rho_j$ are computed as follows:

$$\rho_{(0,0)} = \lambda_{(0,0)}\,\mu_{(0,0)} + \lambda_{(-1,0)}\,\mu_{(1,0)} + \lambda_{(1,-1)}\mu_{(-1,1)} + \lambda_{(0,-1)}\,\mu_{(0,1)} + \lambda_{(-1,-1)}\mu_{(1,1)}$$

$$\rho_{(1,0)} = \lambda_{(0,0)}\,\mu_{(1,0)} + \lambda_{(1,-1)}\,\mu_{(0,1)} + \lambda_{(0,-1)}\,\mu_{(1,1)}$$

(3.3.2.7)
$$\rho_{(0,1)} = \lambda_{(0,0)}\,\mu_{(0,1)} + \lambda_{(-1,0)}\,\mu_{(1,1)}$$

$$\rho_{(-1,0)} = \lambda_{(-1,0)}\,\mu_{(0,0)} + \lambda_{(0,-1)}\,\mu_{(-1,1)} + \lambda_{(-1,-1)}\,\mu_{(0,1)}$$

$$\rho_{(0,-1)} = \lambda_{(0,-1)}\,\mu_{(0,0)} + \lambda_{(-1,-1)}\,\mu_{(1,0)}$$

$$\rho_{(1,-1)} = \lambda_{(1,-1)}\,\mu_{(0,0)} + \lambda_{(0,-1)}\,\mu_{(1,0)}$$

$$\rho_{(-1,1)} = \lambda_{(0,0)}\,\mu_{(-1,1)} + \lambda_{(-1,0)}\,\mu_{(0,1)}$$

$$\rho_{(-1,-1)} = \lambda_{(-1,-1)}\,\mu_{(0,0)}; \qquad \rho_{(2,0)} = \lambda_{(1,-1)}\mu_{(1,1)}$$

$$\rho_{(2,-1)} = \lambda_{(1,-1)}\mu_{(1,0)}$$

$$\rho_{(1,1)} = \lambda_{(0,0)}\mu_{(1,1)}; \qquad \rho_{(-2,0)} = \lambda_{(-1,-1)}\mu_{(-1,1)}$$

$$\rho_{(-2,1)} = \lambda_{(-1,0)}\mu_{(-1,1)}$$

In order to make the ILU decomposition unique we can require

(3.3.2.8) $\lambda_{(0,0)} = 1.$

When we have a 9-point $A^k$, $\lambda_j$ and $\mu_j$ can be computed from

(3.3.2.9) $\rho_j = \sigma_j,\ j \in J.$

In that case the real matrix $R^k$ has 4 points: $\rho_{(2,0)}$, $\rho_{(2,-1)}$, $\rho_{(-2,0)}$ and $\rho_{(-2,1)}$.

In the following table we list the values of $(j_1,j_2)$ for which $\rho_j \neq 0$ in the rest matrix.

| | $5p\ A^k$ | $7\ p\ A^k$ | $9\ p\ A^k$ |
|---|---|---|---|
| 5 p ILU | (-1,1),(1,-1) | | |
| 7 p ILU | (-2,1),(2,-1) | (-2,1),(2,-1) | |
| 9 p ILU | (-2,1),(2,-1) | (-2,1),(2,-1) | (-2,1),(2,-1)<br>(-2,0),(2,0) |

Table 3.3.2.1. Values of j for which $\rho_j \neq 0$ in $R^k$.

For instance when $A^k$ is Poisson then the various ILU-decompositions are with $\lambda_{(0,0)} = 1$:

5-point ILU : $\mu_{(0,0)} = 2 + \sqrt{2} = 3.4142$ ; $\mu_{(1,0)} = \mu_{(0,1)} = -1$ ;

$$\lambda_{(-1,0)} = \lambda_{(0,-1)} = \frac{-1}{\mu_{(0,0)}} = -0.2929$$

$$\rho_{(-1,1)} = \rho_{(1,-1)} = 0.2929$$

7-point ILU = 9-point ILU : $\mu_{(0,0)} = 3.294168$

$$\mu_{(1,0)} = -1.101507 \qquad \lambda_{(-1,0)} = -0.334381$$

$$\mu_{(-1,1)} = -0.334381 \qquad \lambda_{(0,-1)} = -0.101507$$

$$\mu_{(0,1)} = -1 \qquad \lambda_{(0,-1)} = -0.303567$$

$$\rho_{(-2,1)} = \rho_{(2,-1)} = 0.11181.$$

End of example.

The ILU-decomposition can be computed by recursive formulas similar to the Crout method (for the complete LU-decomposition). Consider a 9-point matrix $A^k$ with elements

$$A^k_{i,(0,0)} = a_i,\ A^k_{i,(1,0)} = \beta_i,\ A_{i,(-1,1)} = \delta_i,\ A_{i,(0,1)} = \gamma_i,\ A_{i,(1,1)} = \varepsilon_i$$

$$A^k_{i,(-1,0)} = b_i,\ A_{i,(1,-1)} = d_i,\ A_{i,(0,-1)} = c_i,\ A_{i,(-1,-1)} = e_i$$

In each point $i \in N^k$ we compute,
(elements that are not defined are replaced by zeros):

$$b_{(i_1+1,i_2)} = \frac{b_{(i_1+1,i_2)}}{a_i} \qquad d_{(i_1-1,i_2+1)} = \frac{d_{(i_1-1,i_2+1)}}{a_i}$$

$$c_{(i_1,i_2+1)} = \frac{c_{(i_1,i_2+1)}}{a_i} \qquad e_{(i_1+1,i_2+1)} = \frac{e_{(i_1+1,i_2+1)}}{a_i}$$

(3.3.2.10)

$$a_{(i_1+1,i_2)} = a_{(i_1+1,i_2)} - \beta_i \, b_{(i_1+1,i_2)}$$

$$\delta_{(i_1+1,i_2)} = \delta_{(i_1+1,i_2)} - \gamma_i \, b_{(i_1+1,i_2)}$$

$$\gamma_{(i_1+1,i_2)} = \gamma_{(i_1+1,i_2)} - \varepsilon_i \, b_{(i_1+1,i_2)}$$

$$d_{(i_1,i_2+1)} = d_{(i_1,i_2+1)} - c_{(i_1,i_2+1)}\beta_i$$

$$c_{(i_1+1,i_2+1)} = c_{(i_1+1,i_2+1)} - e_{(i_1+1,i_2+1)}\beta_i$$

$$a_{(i_1-1,i_2+1)} = a_{(i_1-1,i_2+1)} - d_{(i_1-1,l_2+1)}\delta_i$$

$$a_{(i_1,i_2+1)} = a_{(i_1,i_2+1)} - c_{(i_1,i_2+1)}\gamma_i$$

$$a_{(i_1+1,i_2+1)} = a_{(i_1+1,i_2+1)} - e_{(i_1+1,i_2+1)}\varepsilon_i$$

$$b_{(i_1,i_2+1)} = b_{(i_1,i_2+1)} - c_{(i_1,+i_2+1)}\delta_i$$

$$b_{(i_1+1,i_2+1)} = b_{(i_1+1,i_2+1)} - e_{(i_1+1,i_2+1)}\gamma_i$$

$$\beta_{(i_1-1,i_2+1)} = \beta_{(i_1-1,i_2+1)} - d_{(i_1-1,i_2+1)}\gamma_i$$

$$\beta_{(i_1,i_2+1)} = \beta_{(i_1,i_2+1)} - c_{(i_1,i_2+1)}\varepsilon_i$$

The elements are computed recursively for $i_1 = 0(1)2^k$, $i_2 = 0(1)2^k$. In the following figure the dots denote the elements, which are changed with the formulas (3.3.2.10).

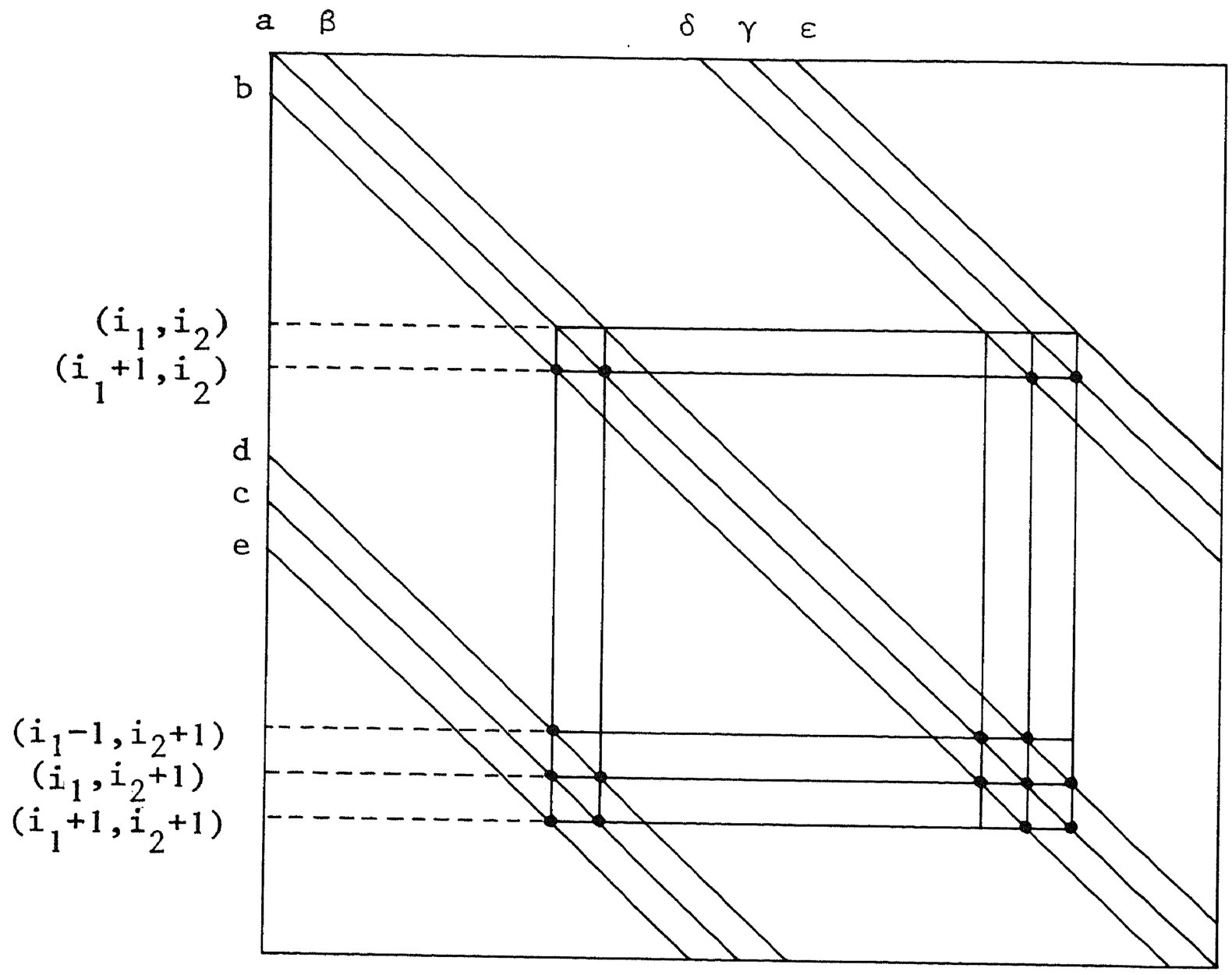



Figure 3.3.2.3. Computation of ILU-decomposition of $A^k$.

The elements of the rest matrix $R^k$ are given by

$$\text{5 p ILU: } \rho_{i,(-1,1)} = b_{(i_1+1,i_2)} \cdot \gamma_i;\ \rho_{i,(1,-1)} = c_{(i_1,i_2+1)} \cdot \beta_i$$

(3.3.2.11)
$$\text{7 p ILU: } \rho_{i,(-2,1)} = b_{(i_1+1,i_2)} \cdot \delta_i;\ \rho_{i,(2,-1)} = d_{(i_1-1,i_2+1)} \cdot \beta_i$$

$$\text{9 p ILU: } \rho_{i,(-2,1)} = b_{(i_1+1,i_2)} \cdot \delta_i;\ \rho_{i,(2,-1)} = d_{(i_1-1,i_2+1)} \cdot \beta_i$$

$$\rho_{i,(-2,0)} = e_{(i_1+1,i_2+1)} \cdot \delta_i;\ \rho_{i,(2,0)} = d_{(i_1-1,i_2+1)} \cdot \varepsilon_i$$

Note that an ILU-decomposition which is according to (3.3.2.10) but only with with divisions yields a method which is equivalent with symmetric Gauss-Seidel (SGS). Suppose $A^k$ can be decomposed as $A^k = \tilde{L}^k + \tilde{D}^k + \tilde{U}^k$ with $\tilde{L}^k$ a strict lower triangular, $\tilde{D}^k$ a diagonal and $\tilde{U}^k$ a strict upper triangular matrix.

Then $\tilde{L}^k$ and $\tilde{U}^k$ satisfy:

$$(3.3.2.12)\quad \tilde{L}^k = L^k \tilde{D}^k - \tilde{D}^k \qquad \tilde{U}^k = U^k - \tilde{D}^k.$$

The amplification matrix of Gauss-Seidel forward is given in (2.1.4). The amplification matrix of symmetric Gauss-Seidel is given by

$$(3.3.2.13)\quad G^k = (\tilde{U}^k + \tilde{D}^k)^{-1} \tilde{L}^k (\tilde{L}^k + \tilde{D}^k)^{-1} \tilde{U}^k.$$

With (3.3.2.12) and (3.3.2.1) we can derive that

$$(3.3.2.14)\quad \begin{aligned} G^k &= (U^k)^{-1}(L^k \tilde{D}^k - \tilde{D}^k)(L^k \tilde{D}^k)^{-1}(\tilde{U}^k - \tilde{D}^k) = \\ &= (L^k U^k)^{-1}(L^k U^k - L^k \tilde{D}^k - U^k + \tilde{D}^k) = \\ &= (L^k U^k)^{-1}(L^k U^k - A^k) = (L^k U^k)^{-1} R^k. \end{aligned}$$

Thus $G^k$ is identical with the amplification matrix of an ILU-decomposition.

In practice Gauss-Seidel forward is performed according to

$$(3.3.2.15)\quad u^k_i = \frac{1}{A^k_{i,(0,0)}} \{ f^k_{i_1,i_2} - \sum_{j \in J \setminus (0,0)} A^k_{i,j}\, u^k_{i+j} \}.$$

For $u^k_{i+j}$, $j \in \{(-1,-1),(0,-1),(1,-1),(-1,0)\}$, we take the most recent values.

It is also possible to define Gauss-Seidel by lines for instance in the $i_1$-direction.

$$(3.3.2.16)\quad A^k_{i,(-1,0)}\, u^k_{(i_1-1,i_2)} + A^k_{i,(0,0)} u^k_{(i_1,i_2)} + A^k_{i,(1,0)}\, u^k_{(i_1+1,i_2)} = f^k_{i_1,i_2} - \sum_{j \in J^*} A^k_{ij}\, u^k_{i+j}, \quad J^* = J \setminus \{(-1,0),(0,0),(1,0)\}$$

For each line a triadiagonal system with $2^k$ equations has to be solved.

# 4. COMPUTATIONAL COMPLEXITY OF MULTIGRID METHODS

The computational complexity T of an iterative process depends on the required accuracy $\varepsilon$, on the rate of convergence $R = \ell n\ S(G)$, with S(G) the spectral radius of G and the computational complexity of one application of the process W:

$$T = \frac{|\ell n \varepsilon|}{R}\ W. \tag{4.1}$$

Instead of the rate of convergence S(G) we use an experimental quantity, which is a measure for S(G) namely the average reduction factor defined by:

$$r_{av} = \left(\frac{\|f-Au^{(\nu_0)}\|}{\|f-Au^{(0)}\|}\right)^{1/\nu_0}, \quad \nu_0 \neq 0, \tag{4.2}$$

with $\nu_0$ the smallest integer such that

$$\|f-Au^{(\nu_0)}\| < \varepsilon. \tag{4.3}$$

$\|.\|$ is the Euclidian norm.

## 4.1. Computational complexity of one multigrid iteration.

In this section an estimate will be derived for the computational complexity W of one multigrid iteration as defined by the procedure *multigrid method* in section 2.3. An operation will be defined as an element from the set {+,-,*,/,sqrt}.

W consists of the computational work of the smoothing processes ($W_s$), the restrictions ($W_r$), the prolongations ($W_p$) and some additional work consisting of the computation of the residuals and additions in the prolongation step ($W_q$)

$$W = W_s + W_r + W_p + W_q. \tag{4.1.1}$$

The following quantities are defined:

| part of the procedure *multigrid method* in section 2.3 | number of operations per grid point |
|---|---|
| $u^k := G^k u^k + B^k f^k$ | $a_s$ |
| $f^{k-1} := R^k \tilde{f}^k$ | $a_r$ |
| $\tilde{u}^k := P^k u^{k-1}$ | $a_p$ |
| $\tilde{f}^k := f^k - A^k u^k$ <br> $u^k := u^k + \tilde{u}^k$ | $a_q$ |

For $\ell$ large, $\sigma < 4$ and with neglecting the computational work on the coarsest grid k=1 we have:

$$W_s = (\rho+\tau)\{a_s' + \frac{\sigma}{4-\sigma} a_s''\}4^\ell$$

$$W_r = \frac{\sigma}{4-\sigma} a_r 4^\ell$$

(4.1.2)

$$W_p = \frac{4}{4-\sigma} a_p 4^\ell$$

$$W_q = \{a_q' + \frac{\sigma}{4-\sigma} a_q''\}4^\ell$$

Because the coarse grid matrices $A^k$, $k < \ell$ may have more diagonals than $A^\ell$ we distinguish between the numbers $a_s'$ and $a_q'$ on the finest grid and $a_s''$ and $a_q''$ on the coarser grids.

Note that in the special case $\rho = 0$ and $\sigma = 1$ the multigrid method of section 2.4 can be used. Then $a_q'' = 0$.

The number of operations per grid point for one APINV-smoothing step according to (3.3.1.11) is given in table 4.1.1. (1) denotes the computation of $\tilde{r}^k := f^k - A^k u^k$ and (2) the computation of $u^k := u^k + B^k \tilde{r}^k$.

| Smoothing | | $5p - A^k$ | | $7p - A^k$ | | $9p - A^k$ | |
|---|---|---|---|---|---|---|---|
| | | * | + | | | | |
| 5p-APINV | (1) | 5 | 5 | | | | |
| | (2) | 5 | 5 | | | | |
| | $a_s$ | 20 | | | | | |
| | | * | + | * | + | | |
| 7p-APINV | (1) | 5 | 5 | 7 | 7 | | |
| | (2) | 7 | 7 | 7 | 7 | | |
| | $a_s$ | 24 | | 28 | | | |
| | | * | + | * | + | * | + |
| 9p-APINV | (1) | 5 | 5 | 7 | 7 | 9 | 9 |
| | (2) | 9 | 9 | 9 | 9 | 9 | 9 |
| | $a_s$ | 28 | | 32 | | 36 | |

Table 4.1.1. Operations per grid point for APINV-smoothing process

In table 4.1.2 we summarize $a_s$ of the ILU-smoothing process according to (3.3.2.5). (1) denotes the computation of $\tilde{r}^k := R^k u^k + f^k$ and (2) the computation of $u^k := (L^k U^k)^{-1} \tilde{r}^k$.

| Smoothing | | 5p - $A^k$ | | | 7p - $A^k$ | | | 9p - $A^k$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | * | * | + | | | | | | |
| 5p-ILU | (1) | | 2 | 2 | | | | | | |
| | (2) | 1 | 4 | 4 | | | | | | |
| | $a_s$ | 13 | | | | | | | | |
| | | / | * | + | / | * | + | | | |
| 7p-ILU | (1) | | 2 | 2 | | 2 | 2 | | | |
| | (2) | 1 | 6 | 6 | 1 | 6 | 6 | | | |
| | $a_s$ | 17 | | | 17 | | | | | |
| | | / | * | + | / | * | + | / | * | + |
| 9p-ILU | (1) | | 2 | 2 | | 2 | 2 | | 4 | 4 |
| | (2) | 1 | 6 | 6 | 1 | 6 | 6 | 1 | 8 | 8 |
| | $a_s$ | 17 | | | 17 | | | 25 | | |

Table 4.1.2. Operations per grid point for ILU-smoothing process

In table 4.1.3 we give the $a_s$ for Jacobi and Gauss-Seidel by points and by lines. It is assumed that a tridiagonal system is solved at a cost of 1 division, 6 multiplications and 3 additions per point.

| Smoothing | | 5p - $A^k$ | | | 7p - $A^k$ | | | 9p - $A^k$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | / | * | + | / | * | + | / | * | + |
| Point Jacobi and Gauss-Seidel | | 1 | 4 | 4 | 1 | 6 | 6 | 1 | 8 | 8 |
| | $a_s$ | 9 | | | 13 | | | 17 | | |
| | | / | * | + | / | * | + | / | * | + |
| Line Gauss-Seidel | | 1 | 8 | 5 | 1 | 10 | 7 | 1 | 12 | 9 |
| | $a_s$ | 14 | | | 18 | | | 22 | | |

Table 4.1.3. Operations per grid point for Jacobi and Gauss-Seidel

The numbers of operations per grid point for restriction and prolongation can be counted from the formulas (3.1.5)-(3.1.10).

| | | 1 point | | 5 point | | 7 point | | 9 point | |
|---|---|---|---|---|---|---|---|---|---|
| Restriction | | * | + | * | + | * | + | * | + |
| | | 0 | 0 | 2 | 4 | 2 | 6 | 3 | 8 |
| | $a_r$ | 0 | | 6 | | 8 | | 11 | |
| Prolongation | | | | | | * | + | * | + |
| | | | | | | 0.75 | 0.75 | 0.75 | 1.25 |
| | $a_p$ | | | | | 1.50 | | 2 | |

Table 4.1.4. Operations per grid point for restriction and prolongation

Finally, we consider $a_q$.

When ILU is used the residual can be computed as follows

$$(r^k)^{(\nu)} := f^k - A^k (u^k)^{(\nu)} := f^k - (L^k U^k - R^k)(u^k)^{(\nu)} := \tag{4.1.3}$$
$$:= R^k \{ (u^k)^{(\nu)} - (u^k)^{(\nu-1)} \}.$$

Since $R^k$ has 2 or 4 diagonals (see table 3.3.2.1), this is a very cheap way to compute the residual. The residual of Jacobi, APINV, and Gauss-Seidel (points and lines) is computed according to $r^k := f^k - A^k u^k$.

| | | $5p - A^k$ | | $7p - A^k$ | | $9p - A^k$ | |
|---|---|---|---|---|---|---|---|
| APINV, Jacobi, point/line Gauss-Seidel | | * | + | * | + | * | + |
| | | 5 | 6 | 7 | 8 | 9 | 10 |
| | $a_q$ | 11 | | 15 | | 19 | |
| 5p - ILU | | * | + | | | | |
| | | 2 | 3 | | | | |
| | $a_q$ | 5 | | | | | |
| 7p ILU | | * | + | * | + | * | + |
| | | 2 | 3 | 2 | 3 | | |
| | $a_q$ | 5 | | 5 | | | |
| 9p - ILU | | * | + | * | + | * | + |
| | | 2 | 3 | 2 | 3 | 4 | 5 |
| | $a_q$ | 5 | | 5 | | 9 | |

Table 4.1.5. Operations per grid point for the computation of the residual plus 1 addition of the prolongation.

## 4.2. Preliminary work.

Before starting the multigrid process we have to compute approximate inverses $B^k$, $k = \ell(-1)1$, when APINV and ILU- smoothing is used, and coarse grid operators $A^k$, $k = \ell-1(-1)1$, when the $R^k A^k P^k$ - coarse grid approximation is applied.

The following quantities are defined:

| | number of operations per gridpoint |
|---|---|
| $B^k$ | $a_b$ |
| $A^{k-1} := R^k A^k P^k$ | $a_c$ |

For $\ell$ large the preliminary work for the computation of $B^k$, $k \leq \ell$, denoted by $PW_b$, and for the computation of $R^k A^k P^k$, $k \leq \ell$, denoted by $PW_c$ is approximately:

$$PW_b = (a'_b + \frac{1}{3}a''_b)4^\ell$$

(4.2.1)

$$PW_c = (\frac{1}{4}a'_c + \frac{1}{12}a''_c)4^\ell.$$

We distinguish between $a'_b$ and $a'_c$ on the finest grid and $a''_b$ and $a''_c$ on the coarser grids.

In the APINV case the approximate inverses are computed according to (3.3.1.5). Per grid point a system of 5,7 or 9 equations has to be solved. Assuming that the solution of a system of n equations costs $\frac{1}{2}n^2 - \frac{1}{2}n$ divisions, $\frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n$ multiplications and additions the following table can be given.

| | | $5p - A^k$ | | | $7p - A^k$ | | | $9p - A^k$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 5p-APINV | | / | * | + | | | | | | |
| | | 10 | 50 | 50 | | | | | | |
| | $a_b$ | 110 | | | | | | | | |
| 7p-APINV | | / | * | + | / | * | + | | | |
| | | 21 | 133 | 133 | 21 | 133 | 133 | | | |
| | $a_b$ | 287 | | | 287 | | | | | |
| 9p-APINV | | / | * | + | / | * | + | / | * | + |
| | | 36 | 276 | 276 | 36 | 276 | 276 | 36 | 276 | 276 |
| | $a_b$ | 588 | | | 588 | | | 588 | | |

Table 4.2.1. Operations per grid point for the computation of the APINV approximate inverse

The preliminary work needed for the computation of the APINV approximate inverses is rather large.* Table 4.2.2 shows that the preliminary work for the ILU-smoothing process is much less. (1) denotes the computation of $L^k$ and $U^k$ according to the recursive formulas (3.3.2.10) and (2) denotes the computation of $R^k$ according to (3.3.2.11).

| | | 5p - $A^k$ | | | 7p - $A^k$ | | | 9p - $A^k$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | / | * | + | | | | | | |
| 5p-ILU | (1) | 2 | 2 | 2 | | | | | | |
| | (2) | | 2 | | | | | | | |
| | $a_b$ | | 8 | | | | | | | |
| | | / | * | + | / | * | | | | |
| 7p-ILU | (1) | 3 | 7 | 7 | 3 | 7 | 7 | | | |
| | (2) | | 2 | | | 2 | | | | |
| | $a_b$ | | 19 | | | 19 | | | | |
| | | / | * | + | / | * | + | / | * | + |
| 9p-ILU | (1) | 3 | 7 | 7 | 3 | 7 | 7 | 4 | 12 | 12 |
| | (2) | | 2 | | | 2 | | | 4 | |
| | $a_b$ | | 19 | | | 19 | | | 32 | |

Table 4.2.2. Operations per grid point for the computation of $L^k$, $U^k$ and $R^k$ for ILU-smoothing process.

Finally, we consider the computation of the coarse grid operators according to (3.2.5). All possible combinations of $R^k$ and $P^k$ are taken.

---

* We remark that FREDERICKSON [8] has proposed his method for the Poisson equation. The preliminary work is zero in that case, because the approximate inverse is known a priori.

| | | $5p - A^k$ | | | | | | | | $7p - A^k$ | | | | | | | | $9p - A^k$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $1p - R^k$ | | $5p - R^k$ | | $7p - R^k$ | | $9p - R^k$ | | $1p - R^k$ | | $5p - R^k$ | | $7p - R^k$ | | $9p - R^k$ | | $1p - R^k$ | | $5p - R^k$ | | $7p - R^k$ | | $9p - R^k$ | |
| | | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + |
| $7p-P^k$ | | 18 | 4 | 62 | 34 | 90 | 54 | 118 | 72 | 27 | 6 | 90 | 54 | 116 | 78 | 148 | 100 | | | | | | | 184 | 128 |
| | $a_c$ | 22 | | 96 | | 144 | | 190 | | 33 | | 144 | | 194 | | 248 | | | | | | | | 312 | |
| $9p-P^k$ | | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + | * | + |
| | | 18 | 4 | 86 | 48 | 118 | 72 | 150 | 96 | 34 | 10 | 118 | 72 | 148 | 100 | 184 | 128 | 50 | 16 | 150 | 96 | 184 | 128 | 218 | 160 |
| | $a_c$ | 22 | | 134 | | 190 | | 246 | | 44 | | 190 | | 248 | | 312 | | 66 | | 246 | | 312 | | 378 | |

Table 4.2.3. Operations per grid point for the computation of a coarse grid operator $A^{k-1} = R^k A^k P^k$

## 4.3 Efficiency of multigrid methods for the Poisson, anisotropic diffusion and convection-diffusion equations.

The procedure *multigrid method* described in section 2.3 is a general algorithm for the solution of a linear 2nd order elliptic boundary value problem. The question is how to choose the parameters $\rho,\sigma,\tau$ and the operators $P^k, R^k, B^k$ and $A^k$ in the procedure *multigrid method* for the method to be fast for a large variety of problems.

In table 4.3.1 we mention the variants of the multigrid method that we consider.

| variant | $\rho$ | $\sigma$ | $\tau$ | $R^k$ | $P^k$ | course grid operator | smoothing process |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 7 | 7 | $R^k A^k P^k$ | 7p - ILU |
| 2 | 0 | 1 | 1 | 7 | 7 | FD | 7p - ILU |
| 3 | 0 | 1 | 1 | 1 | 9 | FD | 7p - ILU |
| 4 | 0 | 1 | 1 | 9 | 9 | $R^k A^k P^k$ | 9p - ILU |
| 5 | 1 | 1 | 1 | 7 | 7 | $R^k A^k P^k$ | 7p - ILU |
| 6 | 1 | 1 | 1 | 7 | 7 | FD | 7p - ILU |
| 7 | 0 | 2 | 1 | 7 | 7 | $R^k A^k P^k$ | 7p - ILU |
| 8 | 1 | 1 | 0 | 7 | 7 | $R^k A^k P^k$ | 7p - ILU |
| 9 | 1 | 1 | 0 | 7 | 7 | FD | 7p - ILU |
| 10 | 0 | 1 | 1 | 7 | 7 | $R^k A^k P^k$ | 7p - APINV |
| 11 | 0 | 1 | 1 | 7 | 7 | $R^k A^k P^k$ | SGS |
| 12 | 0 | 1 | 1 | 7 | 7 | FD | SGS |

Table 4.3.1. Various variants of the multigrid method

Table 4.3.2. gives the operation counts of one multigrid iteration for the 12 variants of table 4.3.1. It has been compiled using the operation counts in the preceeding sections. The table refers to a general 5 and 7 point operator $A^\ell$ with variable coefficients. When there is a difference between 5 and 7 point operators $A^\ell$, the results for the 7 point operators are placed within brackets.

For easy comparison, the computational complexity is expressed in work

units. One work unit, as introduced by BRANDT [4] and denoted here as WUGS, is the number of operations necessary for one Gauss-Seidel sweep on the finest grid. Similarly, we define one WULU as the work of one 7p-ILU smoothing step on the finest grid i.e. 17N operations (see table 4.1.2). $N = O(4^{\ell})$ is the number of points on the finest grid.

| variant | $W_s$ | $W_r$ | $W_p$ | $W_q$ | W | $PW_b$ | $PW_c$ | PW |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.33 | 0.16 | 0.12 | 0.29 | 1.90 | 1.49 | 3.07(3.80) | 4.56(5.29) |
| 2 | 1.33 | 0.16 | 0.12 | 0.29 | 1.90 | 1.49 | - | 1.49 |
| 3 | 1.33 | 0 | 0.16 | 0.29 | 1.78 | 1.49 | - | 1.49 |
| 4 | 1.49 | 0.22 | 0.16 | 0.29 | 2.16 | 1.49 | 5.47(6.44) | 7.22(8.19) |
| 5 | 2.66 | 0.16 | 0.12 | 0.39 | 3.33 | 1.75 | 3.07(3.80) | 4.56(5.29) |
| 6 | 2.66 | 0.16 | 0.12 | 0.39 | 3.33 | 1.49 | - | 1.49 |
| 7 | 2.00 | 0.47 | 0.18 | 0.58 | 3.23 | 1.49 | 3.07(3.80) | 4.56(5.29) |
| 8 | 1.33 | 0.16 | 0.12 | 0.39 | 2.00 | 1.49 | 3.07(3.80) | 4.56(5.29) |
| 9 | 1.33 | 0.16 | 0.12 | 0.39 | 2.00 | 1.49 | - | 1.49 |
| 10 | 1.96(2.20) | 0.16 | 0.12 | 0.65(0.88) | 2.89(3.36) | 22.50 | 3.07(3.80) | 25.6(26.3) |
| 11 | 1.57(2.04) | 0.16 | 0.12 | 0.65(0.88) | 2.50(3.20) | - | 3.07(3.80) | 3.07(3.80) |
| 12 | 1.41(2.04) | 0.16 | 0.12 | 0.65(0.88) | 2.34(3.20) | - | - | - |

Table 4.3.2. Operations count per grid point expressed in WULU for the variants of the multigrid method; $W_s, W_r, W_p$ and $W_q$ are defined in (4.1.2), W in (4.1.1), $PW_b$ and $PW_c$ in (4.2.1); PW is the total preliminary work: $PW_b + PW_c$.

Some conclusions can be derived from table 4.3.2.

- For all variants with ILU the smoothing work $W_s$ and the additional work $W_q$ are independent of the fact whether $A^{\ell}$ is a 5 or a 7 point operator.
- The computational complexity of SGS is larger than the computational complexity of 7p-ILU, but the latter needs preliminary work.
- The contribution of the restriction and prolongation work $W_r$ and $W_p$ to the total work W is rather small (at most 0.20 W).
- The computational complexity of the construction of $L^k$, $U^k$ and $R^k$, $k = \ell(-1)1$ is 0.45 W (variant 5 and 6) up to 0.84 W (variant 3).

- The computational complexity of the construction of the Galerkin approximations $R^k A^k P^k$, $k = \ell(-1)2$ is 0.92 W (variant 5 with 5p-$A^\ell$) up to 2.98 W (variant 4 with 7p-$A^\ell$).
- The APINV-smoothing process in variant 10 requires much preliminary work.

To study the efficiency of the various variants, we compute the average reduction factor for 3 test problems. The test problems have constant coefficients, so the operation counts of table 4.3.2 can be improved for these special cases.

Problem 1. The Poisson equation:

$$u_{x_1x_1} + u_{x_2x_2} = 4, \tag{4.3.1}$$

defined on the unit square $\Omega = (0,1)x(0,1)$ with boundary conditions $u = x_1^2 + x_2^2$ on the boundary $\partial\Omega$. The exact solution is $u = x_1^2 + x_2^2$. The problem is discretized by central differences on the grid $\Omega^\ell$ (2.2.1) with $\ell = 4$. The boundary conditions are substituted in the difference equations. The multigrid iteration is started with the zero solution. The required accuracy is $\varepsilon = 10^{-6}$

| variant | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | 4 | 6 | 7 | 4 | 3 | 4 | 4 | 5 | 6 | 10 | 6 | 10 |
| $r_{av}$ | 0.020 | 0.054 | 0.085 | 0.020 | 0.0061 | 0.014 | 0.020 | 0.055 | 0.065 | 0.29 | 0.071 | 0.214 |
| $t_{10}$ | 1.12 | 1.50 | 1.66 | 1.27 | 1.50 | 1.80 | 1.90 | 1.59 | 1.68 | 5.38 | 2.18 | 3.49 |
| TC | 12.16 | 12.89 | 13.95 | 15.86 | 14.55 | 14.81 | 17.48 | 14.56 | 13.49 | 54.5 | 18.07 | 23.40 |

Table 4.3.3. Results of the multigrid method applied to the Poisson equation.

M: number of multigrid iterations

$r_{av}$: average reduction factor (4.2)

$t_{10} = \dfrac{W}{|\log r_{av}|}$ : number of operations per grid point for 0.1 reduction of the residual

TC = M.W + PW: total number of operations per grid point expressed in WULU with W and PW given in table 4.3.2.

Looking at the total complexity TC it can be concluded that variant 1 has the smallest complexity although there is only little difference with variant 2. In the Poisson case, the coarse grid operators of variant 1 and 2 are the same except for the boundary points. Application of $R^k A^k P^k$ makes variant 1 somewhat faster than variant 2 at the expense of preliminary work.

Comparing variants 2 and 3 it seems that 7-point restriction is better than injection.

Comparison of variant 1 and 4 shows that 9-point restriction and prolongation combined with 9p-ILU does not accelerate the method.

Considering variants 1,5 and 6 we see that additional smoothing, which almost doubles the work in one multigrid iteration step, does not reduce $r_{av}$ enough to make variant 5 and 6 competitive.

Variant 7 shows that increasing of the number of coarse grid corrections does not imporve the $r_{av}$ of variant 1, while W of variant 7 is about 1.7 W of variant 1.

Comparison of the variants 1, 8 and 9 shows that it is better to smooth after than before coarse grid correction. Comparison of the variants 1, 10, 11 and 12 shows that ILU is a better smoother than SGS or APINV.

Problem 2. The anisotropic diffusion equation:

$$u_{x_1x_1} + 0.01\, u_{x_2x_2} = 2.02, \tag{4.3.2}$$

and

$$0.01u_{x_1x_1} + u_{x_2x_2} = 2.02, \tag{4.3.3}$$

defined on the unit square and with the same boundary conditions, exact solution, discretization, starting values and accuracy requirements as in the Poisson case.

| variant | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | 4 | 5 | 5 | 4 | 3 | 3 | 4 | 5 | 5 | 33 | 27 | 27 |
| $r_{av}$ | 0.014 | 0.026 | 0.028 | 0.014 | 0.002 | 0.002 | 0.014 | 0.031 | 0.030 | 0.70 | 0.61 | 0.61 |
| $t_{10}$ | 1.02 | 1.20 | 1.15 | 1.17 | 1.23 | 1.23 | 1.74 | 1.33 | 1.31 | 18.66 | 11.65 | 10.90 |
| TC | 12.16 | 10.99 | 10.39 | 15.86 | 14.55 | 11.48 | 17.48 | 14.56 | 11.49 | 121.0 | 70.6 | 63.2 |

Table 4.3.4. The 12 variants applied to equation (4.3.2). For legenda see Table 4.3.3.

| variant | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | 2 | 3 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 33 | 27 | 27 |
| $r_{av}$ | 1E-4 | 12E-4 | 4E-4 | 1E-4 | 1E-7 | 1E-7 | 1E-4 | 13E-4 | 14E-4 | 0.70 | 0.61 | 0.61 |
| $t_{10}$ | 0.48 | 0.65 | 0.52 | 0.54 | 0.48 | 0.48 | 0.81 | 0.69 | 0.70 | 18.66 | 11.65 | 10.90 |
| TC | 8.36 | 7.19 | 5.05 | 11.54 | 11.22 | 8.15 | 11.02 | 10.56 | 7.49 | 121.0 | 70.6 | 63.2 |

Table 4.3.5. The 12 variants applied to equation (4.3.3). For legenda see Table 4.3.3.

First, we remark that APINV and (point) SGS are bad smoothers in both cases. It is known that line relaxation is better in anisotropic cases.

Multigrid methods with ILU seems to be very robust in the sense that the smoothing operator does not need to be adapted to these anisotropic problems. In the first case the variants with ILU are somewhat slower than in the second, but they are at least as fast as in the Poisson case.

Considering the total complexity TC it can be concluded that variant 3 is superior to the other variants in both anisotropic cases.

Problem 3. The convection-diffusion equation:

$$0.001(u_{x_1x_1} + u_{x_2x_2}) - v_1u_{x_1} - v_2u_{x_2} = 1, \tag{4.3.4}$$

with a) $v_1 = 1$, $v_2 = 0$ c) $v_1 = 1$, $v_2 = 1$

b) $v_1 = 0$, $v_2 = 1$ d) $v_1 = 1$, $v_2 = -1$

defined on the unit square with boundary conditions $u = 0$ on $\partial\Omega$. The 2nd derivatives are discretized by central differences, the first derivatives with Il'ins method:

$$u_{x_1}\Big| \cong \frac{(1+\alpha_{i_1 i_2})(u_{i_1+1,i_2} - u_{i_1 i_2}) + (1-\alpha_{i_1 i_2})(u_{i_1 i_2} - u_{i_1-1,i_2})}{2h_\ell},$$

(4.3.5)

$$u_{x_2}\Big| \cong \frac{(1+\beta_{i_1 i_2})(u_{i_1,i_2+1} - u_{i_1 i_2}) + (1-\beta_{i_1 i_2})(u_{i,i_2} - u_{i_1,i_2-1})}{2h_\ell},$$

with $h_\ell = 2^{-\ell}$.

$\alpha_{i_1 i_2}$ and $\beta_{i_1 i_2}$ are the Il'in coefficients:

$$\alpha_{i_1 i_2} = -\coth\left(\frac{v_1 h}{0.002}\right) + \frac{0.002}{v_1 h_\ell}.$$

(4.3.6)

$$\beta_{i_1 i_2} = -\coth\left(\frac{v_2 h}{0.002}\right) + \frac{0.002}{v_2 h_\ell}.$$

The same starting iterand and required accuracy is taken as in the Poisson case.

| variant | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| case a | M | 3 | 3 | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 16 | 3 | 3 |
| | $r_{av}$ | 0.0030 | 0.0063 | 0.0072 | 0.0018 | 7E-5 | 9E-5 | 0.0024 | 0.0079 | 0.0088 | 0.47 | 0.0056 | 0.0057 |
| case b | M | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 16 | 3 | 3 |
| | $r_{av}$ | 7E-5 | 7E-5 | 0.0001 | 6E-5 | 2E-8 | 3E-8 | 6E-5 | 0.0002 | 0.0002 | 0.47 | 0.0043 | 0.0045 |
| case c | M | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 16 | 1 | 1 |
| | $r_{av}$ | 3E-9 | 5E-9 | 6E-9 | 3E-9 | 4E-9 | 4E-9 | 2E-8 | 2E-8 | 2E-8 | 0.47 | 4E-9 | 4E-9 |
| case d | M | 4 | 6 | 6 | 4 | 3 | 3 | 6 | 6 | 6 | 16 | 10 | 10 |
| | $r_{av}$ | 0.040 | 0.093 | 0.073 | 0.045 | 0.0092 | 0.0095 | 0.0039 | 0.090 | 0.090 | 0.47 | 0.25 | 0.27 |

Table 4.3.6. The 12 variants applied to the cases a,b,c and d of the convection-diffusion equation (4.3.4). For legenda see table 4.3.3.

We see that the multigrid methods with ILU or with SGS are slow in case d, but the $r_{av}$ is comparible with the $r_{av}$ in the Poisson case

| variant | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $t_{10}$ | 1.36 | 1.84 | 1.57 | 1.60 | 1.64 | 1.65 | 2.29 | 1.91 | 1.91 | 8.81 | 4.15 | 4.12 |
| TC | 12.16 | 12.89 | 12.17 | 15.86 | 14.55 | 11.48 | 17.48 | 16.56 | 13.49 | 71.84 | 28.07 | 23.4 |

Table 4.3.7. The 12 variants applied to case d of (4.3.4).
For legenda lee table 4.3.3.

Comparing TC for all variants it can be concluded that variant 6 is best for case d of the convection-diffusion equation.

In the operation counts for the preliminary work, the work to compute the Il'in discretizations (4.3.5) is neglected. When Galerkin is used as coarse grid approximation, this work is done only on the finest grid, but when FD is used, some additional work has to be done on the coarse grids. When we take this work into account, variant 1 is best for case d of the convection-diffusion equation.

From the results of the three problems the following conclusions can be drawn:

- 7p-ILU is a better smoother than APINV or SGS. ILU makes the multigrid method efficient and robust, in the sense that the smoothing operator does not need to be adapted to the singularly perturbed problems 2 and 3.
- Comparison of variant 1 with 2, 5 and 6, 8 and 9 and 11 with 12 shows that a multigrid method with Galerkin approximation $R^k A^k P^k$ is at least as fast as a multigrid method with FD as coarse grid operator. It depends on the amount of preliminary work which coarse grid operator makes the multigrid method more efficient.
- Smoothing after rather than before coarse grid correction is preferable.
- Increasing of the number of coarse grid corrections beyond 1 is not worthwhile.
- Application of the expensive 9-point prolongation and restriction is not

better than the use of the 7-point prolongation and restriction. In the anisotropic cases, injection and 9-point prolongation appears to be better than 7-point prolongation and restriction.

## 4.4. Comparison with other authors

Several authors have reported experiments with multigrid methods applied to elliptic boundary value problems. A survey of their methods is given in the following table.

| Name of the author | Coarse grid operator | Smoothing operator | $P^k$ | $R^k$ |
|---|---|---|---|---|
| FEDORENKO [6] | FD | point GS | 9 | 1 or 9 |
| BRANDT [3,4] | FD | point/line GS | polynomial interpolation or 9 | 1 or 9 |
| FREDERICKSON [8] | $R^k A^k P^k$ | 9p-APINV | 9 | 9 |
| NICOLAIDES [17] | $R^k A^k P^k$ | point/line GS | 7 or 9 | 7 or 9 |
| HACKBUSCH [9] | $R^k A^k P^k$ | point/line chess board GS | 9 | 9 |
| WESSELING [20] | $R^k A^k P^k$ | 7p-ILU | 7 | 7 |

Table 4.4.1. Survey of various multigrid methods on the literature.

Comparisons can be made between the methods when the Poisson equation is solved in a square with Dirichlet boundary conditions. N is the number of grid points on the finest grid.

With variant 1 of the multigrid method we find in table 4.3.3 that the number of operations per grid point for 0.1 reduction of the residual is $t_{10} = 19N$ operations. This number is even smaller, when we exploit the fact that the coefficients in the Poisson case are constant.

FEDORENKO [6] reports for 0.001 reduction of the residual about 185 operations-per-point, thus $t_{10} \cong 62N$ operations (exploiting the fact that the coefficients are constant).

BRANDT mentions in [4] for the solution of the Poisson problem 28N operations (with savings of operations because of the constant coefficients).

FREDERICKSON [8] finds $r_{av} = 0.45$ with W = 54 operations-per-point (with savings for the constant coefficients), thus $t_{10} \cong 156N$ operations.

NICOLAIDES [17] reports results for combinations of a multigrid method with finite element methods. For the case with linear elements he reports $t_{10} = 3.9$ WUGS $\cong 35N$ operations.

HACKBUSCH [9] uses Gauss Seidel with chess-board ordering of the grid-points. The computational complexity of one smoothing step in his method is equal to the computational complexity of two point Gauss Seidel steps. He uses $R^k A^k P^k$ with $R^k$ and $P^k$ the 9-point restriction and prolongation, so that on the coarse grids he has 9-point operators $A^k$. According to table 4.1.3 the smoothing work on the finest grid is 18 operations-per-point and on each coarse grid 34 operations-per-point. The total complexity W of one multigrid iteration step with $\rho = 0$, $\sigma = \tau = 1$ is then 2.76 WULU. He finds $r_{av} = 0.048$. Thus $t_{10} \cong 35N$ operations.

WESSELING [20] also did experiments with variant 1 of our multigrid method. The only difference is that he does not eliminate the boundary conditions from his equations. He finds $r_{av} = 0.023$. Thus $t_{10} \cong 20N$ operations.

The anisotropic diffusion equation (4.3.2) has also been treated by HACKBUSCH [9] and WESSELING [20].

We find with variant 1 of the multigrid method $t_{10} \cong 17N$ operations.

WESSELING reports $r_{av} = 0.053$ with the same multigrid method (but without eliminating the boundary conditions). Thus $t_{10} \cong 25N$ operations.

HACKBUSCH uses chess-board line Gauss Seidel with 28 operations-per-point on the finest grid and 44 operations-per-point on the coarse grids. The computational complexity of one multigrid iteration is W = 3.54 WULU.

With $r_{av} = 0.063$ he has $t_{10} = 50N$ operations.

## 4.5. Final remarks

It has been shown that multigrid methods with 7p-ILU smoothing, $\rho = 0$, $\sigma = \tau = 1$, with Galerkin or FD coarse grid approximation and with 7-point restriction and prolongation or with injection and 9-point prolongation are fast for the problems here considered.

It is not clear which coarse grid approximation is better. Galerkin approximation delivers a fast method at the expense of preliminary work.

In the cases of the Poisson and the convection-diffusion equations the combination: 7-point restriction and prolongation results in a fast multigrid method. In the anisotropic cases the combination: injection and 9-point prolongation appears to be better.

Other model problems, such as the biharmonic problem, a problem with variable coefficients and a nonlinear problem are treated with variant 10 of the multigrid method in MOL [14]. The multigrid method turns out to be as fast as for the Poisson equation in all these cases.

In WESSELING and SONNEVELD [19] and MOL [15] results are reported for the Navier-Stokes equations. The nonlinear equations are Newton-linearized and the system in each Newton iteration is solved by variant 1 of the multigrid method. It appears that the average reduction factor $r_{av}$ in each Newton step is comparable with $r_{av}$ of the Poisson equation.

ACKNOWLEDGEMENT

The author wishes to thank Prof. P. Wesseling for his guidance and dr. P.W. Hemker for his valuable comments on an earlier version of the paper.

REFERENCES

[1] ASTRACHANCEV, G.P., *An iterative method of solving elliptic net problems*, USSR Comp. Math. Math. Phys. 11, no. 2 (1971), 171-182.

[2] BAKHVALOV, N.S., *On the convergence of a relaxation method with natural constraints on the elliptic operator*, USSR Comp. Math. Math. Phys. 6, no. 5 (1966), 101-135.

[3] BRANDT, A., *Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems*, 3rd Int. Conf. on Num. Meth. in Fluid Mechanics (Paris 1972), in: Lecture Notes in Physics 18, Springer, Berlin & New York (1973), 82-89.

[4] BRANDT, A., *Multi-level adaptive solutions to boundary value problems*, Math. Comp. 31 (1977), 333-390.

[5] BRANDT, A., *Multi-level adaptive techniques (MLAT) for singular perturbation problems*, in: Numerical Analysis of Singular Perturbation Problems, P.W. Hemker and J.J.H. Miller eds., Academic Press, London (1979), 53-142.

[6] FEDORENKO, R.P., *A relaxation method for solving elliptic difference equations*, USSR Comp. Math. Math. Phys. 1 (1962), 1092-1096.

[7] FEDORENKO, R.P., *The speed of convergence of one iterative process*, USSR Comp. Math. Math. Phys. 4 no. 3 (1964), 227-235.

[8] FREDERICKSON, P.O., *Fast approximate inversion of large sparse linear systems*, Mathematics report 7-75, Lakehead University, Ontario, Canada (1975).

[9] HACKBUSCH, W., *On the multigrid method applied to difference equations*, Computing 20 (1978), 291-306.

[10] HACKBUSCH, W., *Convergence of multi-grid iterations applied to difference equations*, Math. Comp. 34 (1980), 425-440.

[11] HEMKER, P.W., *Introduction to multi-grid methods*, Nieuw Archief voor Wiskunde (3), XXIX (1981), 71-101.

[12] HEMKER, P.W., *Multigrid bibliography*, in: Colloquium Numerical Integration of Partial Differential Equations, J. Verwer ed., Mathematical Centre, Amsterdam (1980), 88-97.

[13] MEYERINK, J.A. & v.d. VORST, H.A., *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. 31 (1977), 148-162.

[14] MOL, W.J.A., *A multigrid method applied to some simple problems*, Memorandum nr. 287, Appl. Math. Dept., Twente University of Technology (1979).

[15] MOL, W.J.A., *Numerical solution of the Navier-Stokes equations by means of a multigrid method and Newton-iteration*, 7th Int. Conf. on Num. Meth. in Fluid Mechanics (Standord 1980), in: Lecture Notes in Physics 141, Springer, Berlin & New York (1981), 285-291.

[16] NICOLAIDES, R.A., *On the $\ell^2$ convergence of an algorithm for solving finite element equations*, Math. Comp. 31 (1977), 892-906.

[17] NICOLAIDES, R.A., *On some theoretical and practical aspects of multigrid methods*, Math. Comp. 33 (1979), 933-952.

[18] WESSELING, P., *The rate of convergence of a multiple grid method*, Conference on Numerical Analysis (Dundee 1979), in: Lecture Notes in Math. 773, G.A. Watson ed., Springer, Berlin & New York (1980), 164-184.

[19] WESSELING, P. & SONNEVELD, P., *Numerical experiments with a multiple grid and a preconditioned Lanczos method*, Conference on Approximation Methods for Navier-Stokes problems (Paderborn 1979), in: Lecture Notes in Math. 771, R. Rautmann ed., Springer, Berlin (1980), 543-562.

[20] WESSELING, P., *Theoretical and practical aspects of a multigrid method*, Report NA-37, Delft University of Technology (1980).