Introduction to Algebraic Multigrid

Course Notes of an Algebraic Multigrid Course at the

University of Heidelberg in the Wintersemester 1998/99

Version 1.1

Christian Wagner

Interdisziplinäres Zentrum für wissenschaftliches Rechnen
Im Neuenheimer Feld 368
Universität Heidelberg
D-69120 Heidelberg, Germany.

christian.wagner@iwr.uni-heidelberg.de

Contents

Preface

These course notes are the result of an algebraic multigrid course for graduate and PhD students at the Universität Heidelberg, Germany, in the Wintersemester 98/99. The intention of the course was to give the students enough familiarity with algebraic multigrid methods so that they can understand and apply those methods and are maybe able to conduct their own research in this field.

The discussed algebraic multigrid approaches are classified into several groups. One or two representatives of these groups are analyzed in detail, while the other members of these classes are briefly summarized. Theoretical and numerical results are presented. Since I did not compute my own numerical experiments, I relied on the results presented in the corresponding papers.

The selection of the algorithms is not assumed to be complete and does not induce any rating of the performance of algebraic multigrid methods. Any rating of the methods discussed in the course notes is avoided. Moreover, I believe that for each method a problem can be found where it outperforms all other algebraic multigrid schemes. I hope that after studying this monograph, the reader is able to decide by himself which algorithm might be best suited for his particular task.

Special thanks are due to the students participating in the course. Their interest in algebraic multigrid was the main motivation for me to write down these notes.

Please let me know about any errors in the presentation, missing algebraic multigrid approaches or papers which should be included in these course notes or in a future algebraic multigrid course. Feel free to email your remarks, your opinion, any correction, the descriptions of missing algorithm, or missing references to `Christian.Wagner@IWR.Uni-Heidelberg.de`.

The course notes can be downloaded free of charge from my homepage `www.iwr.uni-heidelberg.de/iwr/techsim/chris`. They shall be used for non-commercial purposes only.

Heidelberg, January 1999 Christian Wagner

1 Introduction

1.1 Definitions and Notations

In these course notes, the Euclidean scalar product is denoted by

$$(u, v)_2 = \sum_{i=1}^{n} u_i \, v_i , \quad u, v \in \mathbb{R}^n .$$

DEFINITION 1.1.1 *A symmetric matrix $A = A^T \in \mathbb{R}^{n \times n}$ is positive definite (spd) if*

$$(u, A\, u)_2 > 0 \quad \forall\, u \neq 0.$$

A spd matrix A induces the scalar product

$$(u, v)_A = (u, A\, v)_2 .$$

The Euclidean norm of a vector $u \in \mathbb{R}^n$ is denoted by

$$\|u\|_2 = \sqrt{(u, u)_2}. \tag{1.1.1}$$

Other norms are given by

$$\begin{aligned} \|u\|_A &= \sqrt{(u, u)_A}, \quad A \text{ spd}, & (1.1.2)\\ \|u\|_1 &= \sum_{i=1}^{n} |u_i|, & (1.1.3)\\ \|u\|_\infty &= \max_{i \le n} |u_i|. & (1.1.4) \end{aligned}$$

If A is the result of a discretization of a second order differential equation, the scalar product $(\cdot, \cdot)_A$ and the corresponding norm $\| \cdot \|_A$ are called energy scalar product and energy norm.

DEFINITION 1.1.2 *The matrix norm $\|\cdot\|$ associated to the vector norm $\|\cdot\|$ is defined by*

$$\|A\| = \sup_{u \in \mathbb{R}^n, u \neq 0} \frac{\|A\, u\|}{\|u\|}, \quad A \in R^{n \times n}.$$

Definition 1.1.2 implies the inequality

$$\|A\,B\| \leq \|A\|\|B\|.$$

DEFINITION 1.1.3 *Let $\sigma(A)$ denote the spectrum of A*

$$\sigma(A) = \{\,\lambda \in \mathbb{C}\,|\,\lambda \text{ is eigenvalue of } A\,\}.$$

Then, the spectral radius $\rho(A)$ is defined by

$$\rho(A) = \max_{\lambda\in\sigma(A)} |\lambda|.$$

The matrix norm bounds the spectral radius

$$\rho(A) \leq \|A\|.$$

The matrix norms associated to the vector norms in (1.1.1)–(1.1.4) are

$$\begin{aligned}
\|A\|_2 &= \sqrt{\rho(A^T A)}, \\
\|A\|_B &= \|B^{1/2} A B^{-1/2}\|_2, \quad B \text{ spd}, \\
\|A\|_1 &= \max_{j\leq n} \sum_{i\leq n} |a_{i,j}|, \\
\|A\|_\infty &= \max_{i\leq n} \sum_{j\leq n} |a_{i,j}|.
\end{aligned}$$

The inequality $A < B$, $A, B \in R^{n\times n}$ is always used as elementwise inequality, which means

$$\begin{aligned}
A < B \quad &\Leftrightarrow \quad a_{i,j} < b_{i,j}\ \forall\, i,j, \\
A \leq B \quad &\Leftrightarrow \quad a_{i,j} \leq b_{i,j}\ \forall\, i,j.
\end{aligned}$$

DEFINITION 1.1.4 *The matrix A is called M-matrix if*

(a) $a_{i,i} > 0 \quad \forall\, i,$

(b) $a_{i,j} \leq 0 \quad \forall\, i \neq j,$

(c) $A^{-1} \geq 0.$

REMARK 1.1.1 *If an irreducible matrix A (see e.g. [Hackbusch 1993; Hackbusch 1994]) fulfills the conditions (a) and (b) in Definition 1.1.4 and*

$$a_{i,i} \geq \sum_{j\neq i} |a_{i,j}| \quad \forall\, i$$

and

$$a_{i,i} > \sum_{j\neq i} |a_{i,j}|$$

for at least one i, then A is an M-matrix.

A lot of other criterions for M-matrices are summarized in [Berman and Plemmons 1979].

1.2 Linear Systems and Graphs

We consider linear systems of equations resulting from a discretization of an elliptic second order partial differential equation

$$\begin{aligned} \nabla \cdot (D(x)\, \nabla u(x) + v(x)\, u(x)) - \sigma(x)\, u(x) &= f(x), \quad x \in \Omega, \\ u(x) &= g(x), \quad x \in \Gamma_D, \\ \nabla u(x) \cdot n &= b(x), \quad x \in \Gamma_N, \end{aligned} \tag{1.2.1}$$

where $\Omega \subset \mathbb{R}^d$, $\Gamma_D, \Gamma_N \subset \mathbb{R}^{d-1}$, $D(x) : \Omega \to \mathbb{R}^{d\times d}$, $v(x) : \Omega \to \mathbb{R}^d$, $u(x), \sigma(x), f(x) : \Omega \to \mathbb{R}$, $g(x) : \Gamma_D \to \mathbb{R}$, and $b(x) : \Gamma_N \to \mathbb{R}$.

Finite element (see e.g. [Braess 1997]), finite volume (see e.g. [Hackbusch 1989; Schneider and Raw 1987]) or finite difference discretizations (see e.g. [Hackbusch 1989]) of (1.2.1) lead to a sparse system of linear equations.

For example, a finite difference discretization of the one-dimensional model problem

$$\begin{aligned} -u''(x) &= f(x), \quad x \in \Omega = (0,1), \\ u(x) &= 0, \quad x \in \Gamma_D = \{0,1\}, \end{aligned}$$

on the uniform grid

$$\Omega_h = \{\, x_i \in \Omega \,|\, x_i = i\,h,\ i = 1, \ldots, n,\ h = \frac{1}{n+1}\,\} \tag{1.2.2}$$

yields the linear system

$$A\,u = f, \quad A \in \mathbb{R}^{n\times n},\ u, f \in \mathbb{R}^n,$$

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}, \quad f_i = h^2\, f(x_i).$$

The finite difference discretization of the two-dimensional model problem,

$$\begin{aligned} -\Delta\, u(x) &= f(x), \quad x \in \Omega = (0,1) \times (0,1), \\ u(x) &= 0, \quad x \in \Gamma_D = \partial\Omega, \end{aligned} \tag{1.2.3}$$

on the uniform grid

$$\Omega_h = \{\, x_{(i,j)} \in \Omega \,|\, x_{(i,j)} = (i\,h, j\,h),\ i,j = 1, \ldots, n,\ h = \frac{1}{n+1}\,\}$$

with a lexicographic numbering (see e.g. [Hackbusch 1993; Hackbusch 1994]) of the grid points generates the linear system

$$A\,u = f, \quad A \in \mathbb{R}^{n\cdot n\times n\cdot n}, \quad u, f \in \mathbb{R}^{n\cdot n}, \quad f_{(i,j)} = h^2\, f(x_{(i,j)})$$

with the system matrix

$$A = \begin{pmatrix} Z & -I_n & & & \\ -I_n & Z & -I_n & & \\ & \ddots & \ddots & \ddots & \\ & & -I_n & Z & -I_n \\ & & & -I_n & Z \end{pmatrix}, \quad Z = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \tag{1.2.4}$$

($Z \in \mathbb{R}^{n\times n}$). I_n denotes the $n \times n$ identity matrix. The matrix A in (1.2.4) can be written in stencil notation (see e.g. [Hackbusch 1993; Hackbusch 1994]) as

$$A = \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}. \tag{1.2.5}$$

A sparse matrix A is always connected with a graph $G_A(V, E)$. In the remainder of this section, some definitions from graph theory are summarized. The interested reader is referred to [George and Liu 1981] for a more complete introduction.

Definition 1.2.1 *Corresponding to a sparse $n\times n$ matrix A with symmetric sparsity pattern (i.e. $a_{i,j} \neq 0 \Leftrightarrow a_{j,i} \neq 0$), let $G_A(V, E)$ be the graph that consists of a set $V = \{v_1, \dots v_n\}$ of n ordered vertices (nodes, unknowns), and a set of edges E such that the edge $e_{i,j} \in E$ exists (connecting v_i and v_j) if and only if $a_{i,j} \neq 0$, $i \neq j$.*
For a vertex v_i, the set of neighbor vertices N_i is defined by

$$N_i = \{\, v_j \in V \mid e_{i,j} \in E \,\}.$$

$|N|$ denotes the number of elements in the set N. The degree of a vertex $deg(v_i) = |N_i|$ is just the size of N_i.

The restriction to matrices with symmetric sparsity pattern does not cause any loss of generality. It is only a condition for the sparse matrix data structure (in the computer code), which must guarantee that $a_{j,i}$ is stored if $a_{i,j}$ is stored even if $a_{j,i} = 0$.
As an example case, Figure 1.1 shows the graph $G_A(V, E)$ of the matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}. \tag{1.2.6}$$

1.3 The Linear Iteration Scheme

In these course notes, the construction of efficient approximate inverses M for the linear iteration scheme

$$u^{(i+1)} = u^{(i)} + M^{-1}(f - A\,u^{(i)}) \tag{1.3.1}$$

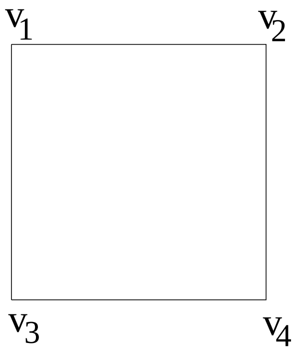

Figure 1.1: The graph of the matrix (1.2.6).

is discussed. These approximate inverses are in general good preconditioners for conjugate gradient type Krylov subspace methods as well (see e.g. [Golub and van Loan 1983; Hackbusch 1993; Hackbusch 1994; Barrett et al. 1994]).

The iteration matrix

$$T = I - M^{-1} A$$

describes the propagation of the error $e^{(i)} = u - u^{(i)}$

$$e^{(i+1)} = T\, e^{(i)}.$$

The sequence $u^{(i)}$ converges towards the solution u of the linear system $A\, u = f$ if and only if

$$\rho(T) < 1.$$

The spectral radius $\rho(T)$ of the iteration matrix describes the convergence rate of the linear iteration. For M and A spd, the error reduction of the conjugate gradient method is given by

$$\|e^{(i+1)}\| \leq \frac{\sqrt{\kappa(M^{-1}A)} - 1}{\sqrt{\kappa(M^{-1}A)} + 1} \|e^{(i)}\|.$$

$\kappa(B) = \|B\|\,\|B^{-1}\|$ is called condition of B.

Finally, two simple convergence results should be mentioned.

Lemma 1.3.1 *Let A be spd and M be given such that the error matrix $N = M - A$ is symmetric and positive semi-definite. Then, the energy norm of the iteration matrix $T = I - M^{-1} A$ is bounded by*

$$\|T\|_A < 1.$$

Proof. See [Hackbusch 1994], Remark 4.8.3. □

The following definition of a regular splitting was introduced in [Varga 1962].

Definition 1.3.1 *The matrix M describes a regular splitting $A = M - N$ of A, if M is regular,*

$$M^{-1} \geq 0, \quad \textit{and} \quad N \geq 0.$$

LEMMA 1.3.2 *Let A with $A^{-1} \geq 0$ and a regular splitting $A = M - N$ be given. Then, the spectral radius of the iteration matrix $T = I - M^{-1}A$ is given by*

$$\rho(T) = \frac{\rho(A^{-1}N)}{\rho(A^{-1}N) + 1} < 1.$$

PROOF. See [Varga 1962; Hackbusch 1993; Hackbusch 1994]. $\square$

2 Standard Multigrid Methods

The first multigrid method has been introduced by Fedorenko in 1961 [Fedorenko 1961]. Their actual efficiency was first realized by Brandt [Brandt 1973] and applied to a large class of problems. In 1976, independent of this development, Hackbusch introduced multigrid methods [Hackbusch 1976].

2.1 Smoothers

The development of multigrid methods started with a detailed analysis of classic iterative methods for the solution of a linear system of equations

$$A\,u = f$$

like the Jacobi, the Gauß-Seidel or the ILU scheme:

$$\begin{aligned} u^{(i+1)} &= u^{(i)} + M^{-1}(f - A\,u^{(i)}), \\ M &= D \quad \text{(Jacobi)}, \\ M &= D + L \quad \text{(Gauß-Seidel)}, \\ M &= (\tilde{D} + \tilde{L})\tilde{D}^{-1}(\tilde{D} + \tilde{U}) \quad \text{(ILU)} \end{aligned}$$

($A = D + L + U$, where L (U) is a strict lower (upper) triangular matrix).

It is easy to check that these methods reduce only the high frequency error components. Hence, instead of reducing the error $e^{(i)} = u - u^{(i)}$, they actually only smooth the error (see Figure 2.1). Therefore, these methods are called smoothers.

For a quantitative analysis of the damped Jacobi iteration

$$u^{(i+1)} = u^{(i)} + \omega\,D^{-1}(f - A\,u^{(i)}), \quad \omega \le 1,$$

note that the eigenvectors $e^{(\nu,\mu)}$ of A and $S = I - \omega\,D^{-1}A$ for the five point stencil (1.2.4) are

$$e^{(\nu,\mu)}_{(i,j)} = \sin(\pi\nu h i)\,\sin(\pi\mu h j), \quad \nu,\mu = 1,\ldots,n. \tag{2.1.1}$$

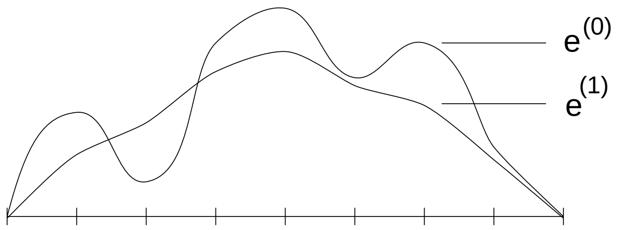

Figure 2.1: The smoothing effect.

The corresponding eigenvalues $\lambda^{(\nu,\mu)}(A)$ and $\lambda^{(\nu,\mu)}(S)$ of A and S are

$$\begin{aligned} \lambda^{(\nu,\mu)}(A) &= 4 - 2\cos(\pi\nu h) - 2\cos(\pi\mu h), \\ \lambda^{(\nu,\mu)}(S) &= 1 - \omega - 0.5\,\omega(\cos(\pi\nu h) + \cos(\pi\mu h)). \end{aligned} \tag{2.1.2}$$

Figure 2.2 shows $\lambda^{(\nu,\mu)}(S)$ for two values of ω. Hence, the low frequency error components are only slowly reduced by a damped Jacobi iteration. Similar results can be shown for the Gauß-Seidel and the ILU-Iteration (see e.g. [Briggs 1987; Hackbusch 1985]). A similar analysis and a detailed introduction to standard multigrid methods can be found in [Wesseling 1992].

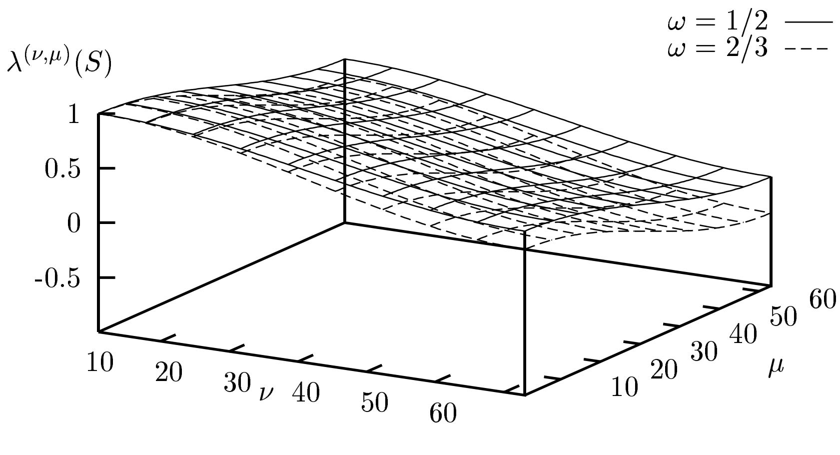

Figure 2.2: Spectrum of the damped Jacobi iteration ($h = 1/64$).

Therefore, after a couple of smoothing steps, a smooth correction must be added to the approximate solution. The idea of the multigrid methods is to compute the smooth correction v_H on a coarser grid and interpolate the correction on the fine grid

$$u^{(i+1)} = u^{(i)} + P\,v_H.$$

The coarse grid correction v_H is the solution of a linear system $A_H v_H = d_H$ on a coarse grid Ω_H.

2.2 Prolongation and Restriction

Let two uniform one-dimensional grids Ω_h and Ω_H (1.2.2) with $H = 2h$ be given. Then, a linear interpolation (prolongation) P from a coarse grid vector $u_H \in \mathbb{R}^{n_H}$ to the fine grid vector $u_h \in \mathbb{R}^{n_h}$, $n_h = 2n_H + 1$, is defined by

$$\begin{aligned}
(u_h)_1 &= (P\,u_H)_1 = 0.5\,(u_H)_1, \\
(u_h)_{n_h} &= (P\,u_H)_{n_h} = 0.5\,(u_H)_{n_H}, \\
(u_h)_i &= (P\,u_H)_i = \begin{cases} (u_H)_i & : \; i \text{ even } \wedge \; i = 2, \ldots, n-1, \\ 0.5\,((u_H)_{\frac{i-1}{2}} + (u_H)_{\frac{i+1}{2}}) & : \; i \text{ odd } \wedge \; i = 3, \ldots, n-2 \end{cases}
\end{aligned}$$

(see Figure 2.3).

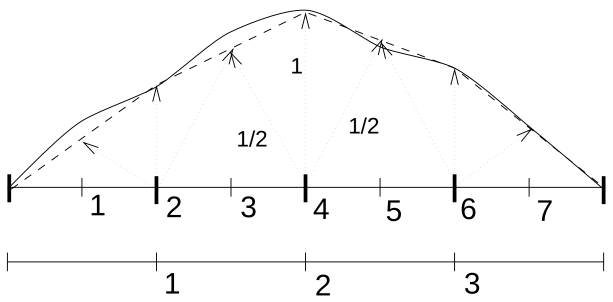

Figure 2.3: One-dimensional prolongation.

A linear interpolation for a two-dimensional grid is illustrated in Figure 2.4. In the Example shown in Figure 2.4, the components $(u_h)_{(2,2)}, (u_h)_{(3,3)}, (u_h)_{(3,4)}$ of the fine grid vector $u_h = P\,u_H$ are computed according to

$$\begin{aligned}
(u_h)_{(2,2)} &= 0.5\,[\,(u_H)_{(1,1)} + (u_H)_{(2,2)}\,], \\
(u_h)_{(3,3)} &= (u_H)_{(2,2)}, \\
(u_h)_{(3,4)} &= 0.5\,[\,(u_H)_{(2,2)} + (u_H)_{(2,3)}\,].
\end{aligned}$$

The linear interpolation can be written in stencil notation as

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 \\ 1/2 & 1/2 & 0 \end{bmatrix}.$$

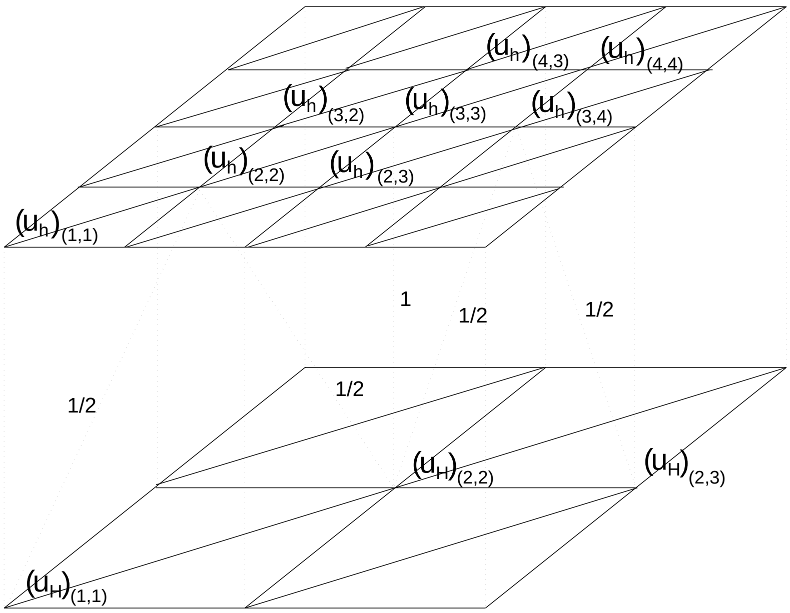

Figure 2.4: Two-dimensional prolongation.

A bilinear interpolation is given by

$$P = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}.$$

Interpolatory prolongations in three spatial dimensions are straightforward. Standard refinement techniques make sure, that each fine grid unknown can be linearly interpolated using two coarse grid nodes. Therefore, linear prolongation operators can be constructed on a hierarchy of regularly refined arbitrary finite element meshes.

Since a correction v_H has to be calculated, the right hand side d_H for the equation on the coarse grid is the result of a restriction of the fine grid residual $d_h = f_h - A_h\, u_h^{(i)}$ to the coarse grid. The most simple restriction R_{inj} is just to take the values of the residual at the corresponding fine grid nodes for the coarse grid right hand side. This means for the one-dimensional example in Subsection 2.2

$$(d_H)_i = (R_{\text{inj}}\, d_h)_i = (d_h)_{2\,i}.$$

The standard restriction is the adjoined operator of the prolongation

$$R = P^* = \left(\frac{h}{H}\right)^d P^T. \tag{2.2.1}$$

The factor $(\frac{h}{H})^d$ reflects the different dimensions of the fine and coarse grid spaces. This leads for the one-dimensional example in Section 2.2 to

$$(d_H)_i = (R\, d_h)_i = 0.5\,[0.5\,(d_h)_{2\,i-1} + (d_h)_{2\,i} + 0.5\,(d_h)_{2\,i+1}],$$

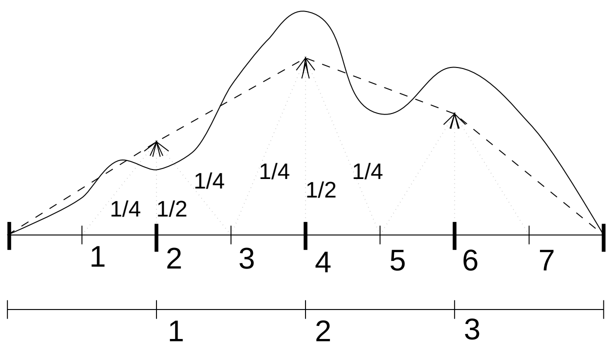

Figure 2.5: One-dimensional restriction.

(see Figure 2.5).

For the two-dimensional example in Figure 2.4, we get for $(d_H)_{(2,2)}$

$$\begin{aligned}(d_H)_{(2,2)} = (R\,d_h)_{(2,2)} &= 1/4\,[0.5\,(d_h)_{(2,2)} + 0.5\,(d_h)_{(2,3)} + 0.5\,(d_h)_{(3,2)} \\ &\quad +0.5\,(d_h)_{(3,4)} + 0.5\,(d_h)_{(4,3)} + 0.5\,(d_h)_{(4,4)} + (d_h)_{(3,3)}].\end{aligned}$$

2.3 The Multigrid Algorithm

In general, there are two possibilities for the choice of the matrix A_H on the coarse grid. One option is to discretize the partial differential equation on the coarse grid with the same method which has been applied on the fine grid.

The second possibility

$$A_H = R\,A_h\,P$$

is called Galerkin approximation. For some discretizations, these two options are equivalent.

The combination of a smoothing procedure and a coarse grid correction leads to the two-grid method. The smoothing procedure $S^\nu(u_h, f_h)$ returns an improved solution for the right hand side f_h starting with u_h and computing ν steps.

ALGORITHM 2.3.1 *Let two grids $\Omega_h \supset \Omega_H$, prolongation and restriction operators P, R between these grids, matrices $A_h = A$, A_H, and a smoothing iteration S be given. Then, the algorithm TGM(u_h, f_h) defines the approximate inverse M_{TGM}^{-1} for the two grid method.*

$\texttt{TGM}(u_h, f_h)$
{
$\quad u_h = S^{\nu_1}(u_h, f_h)$;
$\quad d_H = R\,(f_h - A_h\,u_h)$;
$\quad v_H = A_H^{-1}\,d_H$;
$\quad u_h = u_h + P\,v_H$;

```
        u_h = S^{ν2}(u_h, f_h);
    }
```

Since the exact solution of the coarse grid system $A_H\, v_h = d_H$ in Algorithm 2.3.1 is usually still very time consuming, it is recursively replaced by γ two-grid iteration steps. This yields the multigrid algorithm.

ALGORITHM 2.3.2 *Let a hierarchy of grids $\Omega_0 \subset \Omega_1 \subset \cdots \subset \Omega_{l_{\max}}$, prolongation and restriction operators $P_{l,l-1}, R_{l-1,l}$ between these grids, matrices A_l and smoothing iterations S_l on these grids be given. Then, the algorithm* MGM$(u_{l_{\max}}, f_{l_{\max}}, l_{\max})$ *defines the approximate inverse M^{-1}_{MGM} on the finest grid $\Omega_{l_{\max}}$.*

```
MGM(u_l, f_l, l)
{
    if(l = 0)  u_l = A_l^{-1} f_l;
    else
    {
        u_l = S_l^{ν1}(u_l, f_l);
        d_{l-1} = R_{l-1,l} (f_l - A_l u_l);
        v_{l-1} = 0;
        for(j = 0;  j < γ;  j = j + 1)  MGM(v_{l-1}, d_{l-1}, l - 1);
        u_l = u_l + P_{l,l-1} v_{l-1};
        u_l = S_l^{ν2}(u_l, f_l);
    }
}
```

For $\gamma = 1$ or $\gamma = 2$, the method is called V(ν_1, ν_2)-cycle or W(ν_1, ν_2)-cycle respectively. For a 4-level method, Figure 2.6 shows the order in which the grids are visited for $\gamma = 1$ and $\gamma = 2$. A dot represents a smoothing operation. The grid transfer operators are symbolized by lines.

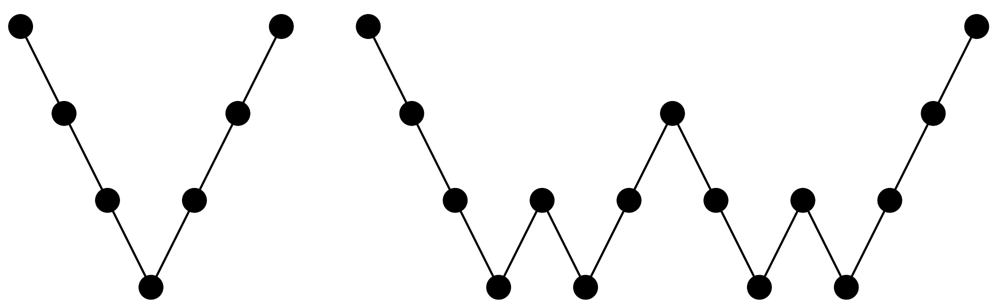

Figure 2.6: V- and W-cycle.

2.4 Numerical Complexity and Storage Requirement

We can assume, that the number of unknowns n_l on each level Ω_l and the storage requirement $\bar{s}_l$ on each level are bounded by

$$n_l \leq \vartheta^{l_{\max}-l}\, n, \quad n_{l_{\max}} = n, \quad \vartheta < 1,$$

and

$$\bar{s}_l \leq C_s\, n_l.$$

For uniform grids, ϑ is approximately given by $\vartheta \approx (1/2)^d$, where d denotes the spatial dimension. The storage requirement $s_{l_{\max}}$ for the multigrid method can therefore be estimated by

$$s_{l_{\max}} = \sum_{l=0}^{l_{\max}} \bar{s}_l \leq C_s\, n \sum_{l=0}^{l_{\max}} \vartheta^{l_{\max}-l} = C_s\, n\, \frac{1-\vartheta^{l_{\max}+1}}{1-\vartheta}.$$

Let $\bar{w}_l \leq C_w\, n_l$ be the computational work on level l aside from the work for the coarse grid correction. The numerical complexity w_l of a multigrid method starting on level l is then bounded by

$$w_l \leq \bar{w}_l + \gamma\, w_{l-1}.$$

This yields

$$w_{l_{\max}} \leq \sum_{l=0}^{l_{\max}} \bar{w}_l\, \gamma^{l_{\max}-l} \leq C_w\, n \sum_{l=0}^{l_{\max}} (\vartheta\, \gamma)^{l_{\max}-l} = C_w\, n \frac{1-(\vartheta\, \gamma)^{l_{\max}+1}}{1-(\vartheta\, \gamma)}.$$

Hence, for an efficient method, $\vartheta \cdot \gamma < 1$ is required.

Due to the computation of an exact solution on the coarsest grid, the constant C_w is determined by the coarsest grid. Hence, a better estimate can be obtained by using a different constant C_w^0 for the coarsest grid which, then, allows a smaller constant C_w for all other grids.

2.5 Convergence Theory

The classic multigrid convergence theory by Hackbusch [Hackbusch 1985] is based on a splitting of the iteration matrix

$$T_{TL,l} = S_l^{\nu_2} (I - P_{l,l-1}\, A_{l-1}^{-1}\, R_{l-1,l}\, A_l)\, S_l^{\nu_1}$$

of the two-level method (Algorithm 2.3.2) . Assuming $\nu_2 = 0$, $\nu_1 = \nu$, the norm of $T_{TL,l}$ is bounded by

$$\|T_{TL,l}\| \leq \|A_l^{-1} - P_{l,l-1}\, A_{l-1}^{-1}\, R_{l-1,l}\| \, \|A_l\, S_l^{\nu}\|.$$

The conditions

$$\begin{aligned} \|A_l\, S_l^\nu\| &\leq C_S\, \eta(\nu)\, h^{-\alpha}, \quad \eta(\nu) \to 0 \text{ as } \nu \to \infty, \\ \|A_l^{-1} - P_{l,l-1}\, A_{l-1}^{-1}\, R_{l-1,l}\| &\leq C_A\, h^\alpha \end{aligned}$$

leading to

$$\|T_{TL,l}\| \leq C_A\, C_S\, \eta(\nu)$$

are called smoothing property and approximation property. The smoothing property measures the performance of the smoother S_l and can be analyzed with linear algebra techniques only. The approximation property describes the quality of the approximation of the differential operator by the coarse grid matrix A_{l-1}. The investigation of the approximation property is e.g. based on results from the finite element theory.

The proof of the approximation property requires that the transfer operators P and R satisfy certain conditions. Let m_P and m_R denote the highest degree plus one of the polynomials that are exactly interpolated by P and R^* (see (2.2.1)). Hackbusch [Hackbusch 1985] deduces the following simple condition for the orders m_P and m_R of the transfer operators

$$m_P + m_R > 2\, m, \tag{2.5.1}$$

where $2\,m$ is the order of the differential operator. The necessity of (2.5.1) is shown in [Hemker 1990]. For example, linear interpolation P and $R = P^*$ leads to $2+2 > 2$ for a second order differential equation.

For a more detailed convergence theory, we refer to the multigrid literature (e.g. [Bramble, Pasciak, Wang, and Xu 1991; Hackbusch 1985; Xu 1992; Yserentant 1993]).

The theoretical results and the practical experience show that the standard multigrid method (Algorithm 2.3.2 with piecewise (bi-)linear prolongation $P_{l,l-1}$) converges very fast and independent of the mesh size h for problem (1.2.1) with smooth coefficients $D(x)$. But the performance is not robust for more complex problems like problems with jumping coefficients $D(x)$ or anisotropic problems. Only for some special problems, robust smoothers (Section 2.6) can improve the convergence. Many "real-life" problems require complicated domains which lead to very fine "coarsest" grids Ω_0. Since the equation $A_0\, u_0 = f_0$ on the coarsest grid has to be solved almost exactly, the efficiency of the multigrid method deteriorates for these problems. At this point, the most promising approaches for an efficient solution of these more complex problems are algebraic multigrid methods.

2.6 Robust Smoothers

To improve the performance of the standard multigrid method either a better smoother has to be applied or the coarse grid correction must be modified. This section focuses on so-called robust smoothers while the remainder of these course notes investigates the improvement of the coarse grid correction. A detailed and more complete coverage of smoothing techniques can be found e.g. in [Hackbusch 1985; Hackbusch 1994; Hackbusch 1994; Kettler 1981; Wesseling 1992; Wittum 1989b].

Ordering Techniques

Discretizations of the convection-diffusion equation

$$\nabla \cdot (\epsilon \nabla u(x) + v(x)\, u(x)) = f, \quad x \in \Omega \subset \mathbb{R}^d, \quad u : \Omega \to \mathbb{R},\ v : \Omega \to \mathbb{R}^d \qquad (2.6.1)$$

or the Navier-Stokes equation

$$\begin{aligned} -\Delta u + R\,(u \cdot \nabla)\, u + \nabla p &= f, \\ \nabla \cdot u &= 0, \end{aligned}$$

$u : \Omega \to \mathbb{R}^d$, $p : \Omega \to \mathbb{R}$, generate unsymmetric system matrices.

In some cases, a special ordering of the unknowns for a Gauß-Seidel smoother leads to an efficient method. For instance, an upwind finite volume discretization of (2.6.1) with $\epsilon \ll 1$ and $v = (1,0)^T$ yields with the downwind ordering shown in Figure 2.7 a system matrix $A = L + D + U$ with a strict lower triangular matrix L, a diagonal matrix D, and a strict upper triangular matrix U satisfying

$$\|U\| \ll \|L + D\|.$$

$M = L + D$ is therefore supposed to be a good approximation for A.

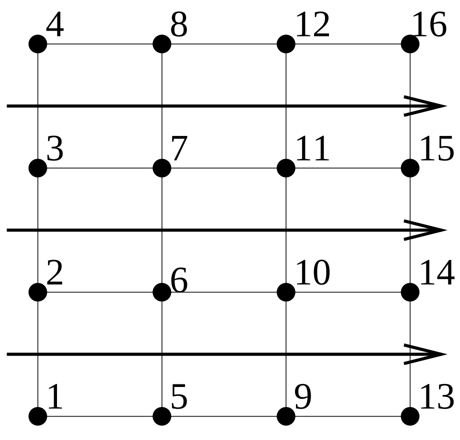

Figure 2.7: Downwind ordering.

Block Factorizations

Block factorizations are known [Kettler 1981; Wesseling 1992] to be efficient and robust smoothers for some two dimensional problems on uniform grids such as anisotropic problems (see Section 4.1). The unknowns are ordered such that the system matrix can be written in block tridiagonal form

$$A = \begin{pmatrix} D_1 & U_{1,2} & 0 & \dots & 0 \\ L_{2,1} & D_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & D_{n-1} & U_{n-1,n} \\ 0 & \dots & 0 & L_{n,n-1} & D_n \end{pmatrix}.$$

Block factorizations are often called line factorizations (ILLU) because the blocks often correspond to grid lines. An exact block factorization is given by

$$\begin{aligned} A \;=\; & \begin{pmatrix} B_1 & 0 & \dots & \dots & 0 \\ L_{2,1} & B_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & B_{n-1} & 0 \\ 0 & \dots & 0 & L_{n,n-1} & B_n \end{pmatrix} \\ & \begin{pmatrix} B_1^{-1} & 0 & \dots & \dots & 0 \\ 0 & B_2^{-1} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & B_{n-1}^{-1} & 0 \\ 0 & \dots & \dots & 0 & B_n^{-1} \end{pmatrix} \begin{pmatrix} B_1 & L_{1,2} & 0 & \dots & 0 \\ 0 & B_2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & B_{n-1} & L_{n-1,n} \\ 0 & \dots & \dots & 0 & B_n \end{pmatrix} \end{aligned}$$

with

$$B_i = D_i - L_{i,i-1}\, B_{i-1}^{-1}\, U_{i-1,i}.$$

Since the matrices B_i are dense, incomplete variants have been proposed. It is, for example, possible to compute the diagonal and the first off-diagonals of $L_{i,i-1}\, B_{i-1}^{-1}\, U_{i-1,i}$ relatively cheap. Let the operator $(Y)^{(p)}$ be defined by

$$(Y)^{(p)} = \begin{cases} y_{i,j} & : \ |i-j| \le \frac{p-1}{2}, \\ 0 & : \ \text{otherwise.} \end{cases}$$

Then, an incomplete block factorization is given by

$$B_i = D_i - (L_{i,i-1}\, B_{i-1}^{-1}\, U_{i-1,i})^{(3)}.$$

For more information, we refer to [Axelsson and Eijkhout 1987; Axelsson and Polman 1989; Hemker 1983; Kettler 1981].

Distributive Iterations

Instead of solving $A\,u = f$,

$$A\,B\,\bar{u} = f, \quad u = B\,\bar{u}$$

might be solved. B is chosen such that a good approximate inverse M for $A\,B$ can be found which leads to a convergent iteration

$$\|I - M^{-1}\,(A\,B)\| < 1.$$

This induces the iterative scheme

$$u^{(i+1)} = u^{(i)} + B\,M^{-1}\,(f - A\,u^{(i)}). \tag{2.6.2}$$

The scheme (2.6.2) is called distributive iteration, because the matrix B distributes the correction $M^{-1}\,(f - A\,u^{(i)})$ over the elements of u. A general discussion of this approach is given by Wittum [Wittum 1986; Wittum 1989a], who shows that a number of well known iterative methods for the Stokes and Navier-Stokes equations can be interpreted as distributed iterations. Taking $B = A^T$ and choosing M to be the Gauß-Seidel method for $A\,A^T$ leads to the Kaczmarz [Kaczmarz 1937] method which converges for every regular matrix A. Convergence is, however, usually very slow.

2.7 The Hierarchical Basis Multigrid Method

The basic idea of the hierarchical basis multigrid method (HBMG) is a transformation of the system matrix A from the nodal basis to the hierarchical basis (see Figure 2.8)

$$\bar{A} = J^T\,A\,J.$$

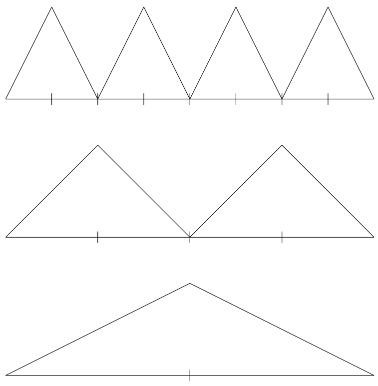

Figure 2.8: The hierarchical basis in 1D.

Let two nested finite element triangulations $\Omega_l \supset \Omega_{l-1}$ and corresponding finite element spaces $V_l \supset V_{l-1}$ be given. The standard basis for V_l is the nodal basis

$$\{\varphi_i^{(l)}\}_{i=1}^{n_l}, \quad \varphi_i^{(l)}(x_j) = \delta_{i,j}.$$

An alternative basis is the two-level hierarchical basis

$$\{\varphi_i^{(l)},\ x_i \in \Omega_l \setminus \Omega_{l-1} \text{ and } \varphi_i^{(l-1)},\ x_i \in \Omega_l \cap \Omega_{l-1}\},$$

where the basis functions $\varphi_i^{(l)}$ corresponding to nodes $v_i \in \Omega_l \cap \Omega_{l-1}$, which occur on both grids Ω_l and Ω_{l-1}, are replaced by the corresponding coarse grid nodal basis function $\varphi_i^{(l-1)}$. The splitting of the set of nodes/vertices $\Omega_l = V$ into

$$V = F \oplus C, \quad F = \Omega_l \setminus \Omega_{l-1}, \quad C = \Omega_l \cap \Omega_{l-1}$$

induces a partition of the unknowns u, the right hand side f and the system matrix A

$$\begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} u_F \\ u_C \end{pmatrix} = \begin{pmatrix} f_F \\ f_C \end{pmatrix}.$$

Any function $g(x) \in V_l$ can be expanded by using either the nodal basis

$$g(x) = \sum_{i=1}^{n_l} g_i \, \varphi_i^{(l)}(x)$$

or the two-level hierarchical basis

$$g(x) = \sum_{v_i \in F} \bar{g}_i \, \varphi_i^{(l)}(x) + \sum_{v_i \in C} \bar{g}_i \, \varphi_i^{(l-1)}(x).$$

This defines a mapping J of the form

$$J = \begin{pmatrix} I_{n_F} & J_{FC} \\ 0 & I_{n_C} \end{pmatrix}. \tag{2.7.1}$$

which transforms any coefficient vector $\bar{g}$ of the representation of the function $g(x)$ in the two-level hierarchical basis to the coefficient vector g of the representation with respect to the nodal basis, i.e.

$$g = J \, \bar{g}.$$

The system matrix with respect to the two-level hierarchical basis $\bar{A}_l$ is then given by

$$\bar{A} = J^T A \, J$$

with

$$\begin{aligned} \bar{A}_{FC} &= A_{FC} + A_{FF} \, J_{FC}, & \bar{A}_{CF} &= A_{CF} + J_{FC}^T \, A_{FF}, \\ \bar{A}_{FF} &= A_{FF}, & \bar{A}_{CC} &= A_{CC} + J_{FC}^T \, A_{FF} \, J_{FC} = A_H. \end{aligned}$$

The multilevel hierarchical basis is obtained by recursively replacing the basis functions $\varphi_i^{(l)}$ corresponding to nodes v_i, which exist on coarser levels too, with the corresponding coarse grid nodal basis function $\varphi_i^{(l-1)}$. An example for a 3-level hierarchical basis is shown in Figure 2.8.

EXAMPLE 2.7.1

$$A = \left(\begin{array}{cccc|ccc} 2 & & & & -1 & & \\ & 2 & & & -1 & -1 & \\ & & 2 & & & -1 & -1 \\ & & & 2 & & & -1 \\ \hline -1 & -1 & & & 2 & & \\ & -1 & -1 & & & 2 & \\ & & -1 & -1 & & & 2 \end{array}\right)$$

represents the discretization of $h^2 \frac{\partial^2}{\partial x^2}$ *on the finest grid in Figure 2.8. The transformation with*

$$J = \left(\begin{array}{cccc|ccc} 1 & & & & 1/2 & & \\ & 1 & & & 1/2 & 1/2 & \\ & & 1 & & & 1/2 & 1/2 \\ & & & 1 & & & 1/2 \\ \hline & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{array}\right)$$

to the two-level hierarchical basis yields

$$\bar{A} = J^T A J = \left(\begin{array}{cccc|ccc} 2 & & & & & & \\ & 2 & & & & & \\ & & 2 & & & & \\ & & & 2 & & & \\ \hline & & & & 1 & -1/2 & \\ & & & & -1/2 & 1 & -1/2 \\ & & & & & -1/2 & 1 \end{array}\right)$$

This induces that the transformation of a matrix

$$A = \left[\begin{array}{ccc} -1 & 2 & -1 \end{array}\right]$$

to the multilevel hierarchical basis leads to a diagonal matrix $\bar{A}$.

Since $\bar{D}^{-1}\bar{A}$, where $\bar{D} = \mathrm{diag}(\bar{A})$, is well conditioned, the inverse of $\bar{A}$ can be approximated by a Jacobi or a Gauß-Seidel iteration. This leads to the hierarchical basis multigrid method.

The HBMG can be written and implemented as standard multigrid method, with the modification that the smoother operates only on those unknowns, which do not exist on coarser levels. Hence, each unknown is modified only on one multigrid level. Moreover, in this sense, each unknown is assigned to exactly one level in the HBMG, while in the standard multigrid method some unknowns occur on several levels.

The convergence theory is typically based on the strengthened Cauchy-Schwarz inequality

$$(v_F, \bar{A}_{FC}\, v_C)_2^2 \le \xi^2\, (v_F, \bar{A}_{FF}\, v_F)_2\, (v_C, \bar{A}_{CC}\, v_C)_2 \quad \forall\, v_F \in \mathbb{R}^{n_F},\ v_C \in \mathbb{R}^{n_C} \tag{2.7.2}$$

with $\xi < 1$. While the convergence is almost independent of the mesh width h for 2D problems, the HBMG is less efficient for 3D problems.

As explained in [Griebel 1994], multigrid methods can be viewed as Jacobi- or Gauß-Seidel iteration of a transformed system as well. For a detailed discussion of the HBMG, we refer to [Bank 1997; Bank, Dupont, and Yserentant 1988; Yserentant 1986].

3 Matrix-Dependent Transfer Operators

Multigrid methods with matrix-dependent transfer operators are based on the multigrid algorithm (2.3.2). Only the linear prolongation and restriction operators are replaced by new operators. For the construction of these transfer operators, information from the system matrix is used. The restriction $R_{l-1,l}$ and the coarse grid matrices A_l are defined by

$$R_{l-1,l} = P^T_{l,l-1}, \quad A_{l-1} = R_{l-1,l}\, A_l\, P_{l,l-1}$$

for all algorithms discussed in this chapter.

3.1 A One-Dimensional Model Problem

The discretization of the one-dimensional differential equation

$$\begin{aligned} -\frac{\partial}{\partial x}\left(D(x) \frac{\partial\, u(x)}{\partial\, x}\right) &= f(x), \quad x \in \Omega = (0,1) && (3.1.1)\\ u(0) = u(1) &= 0, \\ D(x) &= \left\{ \begin{array}{ccl} \epsilon & : & x < i_0\, h, \\ 1 & : & i_0\, h \le x \le i_1\, h, \\ \epsilon & : & x > i_1\, h, \end{array} \right. \end{aligned}$$

on the equidistant grid $\Omega_h = \{\, x \in \Omega \,|\, x = i\,h,\ i = 1,\ldots,n,\ h = \frac{1}{n+1} \,\}$ leads to the system matrix

$$\begin{pmatrix} 2\,\epsilon & -\epsilon & & & & & & & & & \\ -\epsilon & 2\,\epsilon & -\epsilon & & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & & & \\ & & -\epsilon & 1+\epsilon & -1 & & & & & & \\ & & & -1 & 2 & -1 & & & & & \\ & & & & \ddots & \ddots & \ddots & & & & \\ & & & & & -1 & 1+\epsilon & -\epsilon & & & \\ & & & & & & -\epsilon & 2\,\epsilon & -\epsilon & & \\ & & & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & & -\epsilon & 2\,\epsilon & \end{pmatrix}.$$

For $\epsilon \to 0$, the eigenvector corresponding to the largest eigenvalue of the Jacobi iteration $S = I - \omega\, D^{-1}\, A$,

$$D^{-1}\,A = \begin{pmatrix} 1 & -1/2 & & & & & & & & & \\ -1/2 & 1 & -1/2 & & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & & & \\ & & \frac{-\epsilon}{1+\epsilon} & 1 & \frac{-1}{1+\epsilon} & & & & & & \\ & & & -1/2 & 1 & -1/2 & & & & & \\ & & & & \ddots & \ddots & \ddots & & & & \\ & & & & & \frac{-1}{1+\epsilon} & 1 & \frac{-\epsilon}{1+\epsilon} & & & \\ & & & & & & -1/2 & 1 & -1/2 & & \\ & & & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & & -1/2 & 1 & \end{pmatrix},$$

converges towards the vector $e^{(0)}$

$$(e^{(0)})_i = \begin{cases} i\,h & : \ x < i_0\,h, \\ i_0\,h & : \ i_0\,h \le x \le i_1\,h, \\ \frac{i_0}{i_1-n-1} i\,h - \frac{i_0\cdot(n+1)}{i_1-n-1} h & : \ x > i_1\,h \end{cases}$$

(see Figure 3.1).

While the eigenvalue $\lambda^{(0)}$, $S\,e^{(0)} = \lambda^{(0)}\,e^{(0)}$, converges towards 1 for $\epsilon \to 0$, the linear interpolation at x_{i_0} and x_{i_1} is only of first order.

$$\begin{aligned} \Big((1 - P\,R_{\text{inj}})\,e^{(0)}\Big)_{i_0} &= i_0\,h - 0.5\,[(i_0-1)\,h + i_0\,h] = 0.5\,h, \\ \Big((1 - P\,R_{\text{inj}})\,e^{(0)}\Big)_{i_1} &= i_0\,h - 0.5\,[i_0\,h + i_0\,h + \frac{i_0}{i_1-n-1} h] = -\frac{1}{2}\frac{i_0}{i_1-n-1} h. \end{aligned}$$

Therefore, the multigrid convergence deteriorates for $\epsilon \to 0$. For interface problems like (3.1.1), better prolongations and restrictions can be constructed using matrix

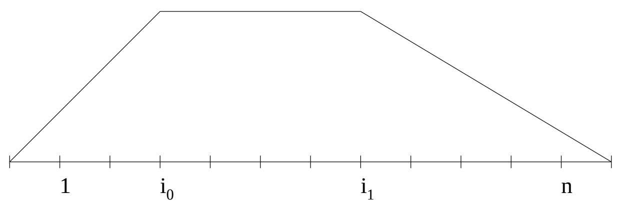

Figure 3.1: Eigenvector $e^{(0)}$.

information. The description of matrix-dependent transfer operators in this chapter follows the paper of de Zeeuw [de Zeeuw 1990]. Similar construction schemes for matrix-dependent transfer operators have been proposed e.g. in [Alcouffe, Brandt, Dendy, and Painter 1981; Dendy 1982; Fuhrmann 1994; Fuhrmann 1995; Hackbusch 1985; Kettler 1981; Reusken 1994; Reusken 1995; Reusken 1996; Wagner, Kinzelbach, and Wittum 1997].

3.2 Prolongation Stencil

The two-dimensional equidistant grid

$$\Omega_l = \{ (x_1, x_2) \in \Omega \,|\, x_1 = i\, h_l,\ x_2 = j\, h_l,\ i, j \in \mathbb{Z} \}$$

is split into four disjunct subsets ($h_{l-1} = 2\, h_l$)

$$\begin{array}{rcl}
\Omega_{l,(0,0)} & = & \Omega_{l-1}, \\
\Omega_{l,(1,0)} & = & \{ (x_1 + h_l, x_2) \in \Omega_l \,|\, (x_1, x_2) \in \Omega_{l-1} \}, \\
\Omega_{l,(0,1)} & = & \{ (x_1, x_2 + h_l) \in \Omega_l \,|\, (x_1, x_2) \in \Omega_{l-1} \}, \\
\Omega_{l,(1,1)} & = & \{ (x_1 + h_l, x_2 + h_l) \in \Omega_l \,|\, (x_1, x_2) \in \Omega_{l-1} \}.
\end{array}$$

The following prolongation $P_{l,l-1}$ is introduced

$$(P_{l,l-1}\, u_{l-1})(x) = \begin{cases}
u_{l-1}(x) & : \ x \in \Omega_{l,(0,0)}, \\
b_l(x)\, u_{l-1}(x + h_l\,(-1,0)) & \\
\quad + a_l(x)\, u_{l-1}(x + h_l\,(1,0)) & : \ x \in \Omega_{l,(1,0)}, \\
b_l(x)\, u_{l-1}(x + h_l\,(0,1)) & \\
\quad + a_l(x)\, u_{l-1}(x + h_l\,(0,-1)) & : \ x \in \Omega_{l,(0,1)}, \\
b_l(x)\, u_{l-1}(x + h_l\,(-1,1)) & \\
\quad + c_l(x)\, u_{l-1}(x + h_l\,(1,1)) & \\
\quad + d_l(x)\, u_{l-1}(x + h_l\,(-1,-1)) & \\
\quad + a_l(x)\, u_{l-1}(x + h_l\,(1,-1)) & : \ x \in \Omega_{l,(1,1)}.
\end{cases}$$

$P_{l,l-1}$ can be written in stencil notation as

$$P_{l,l-1}(x) = \begin{bmatrix} a_l(x + h_l\,(-1,1)) & a_l(x + h_l\,(0,1)) & d_l(x + h_l\,(1,1)) \\ a_l(x + h_l\,(-1,0)) & 1 & b_l(x + h_l\,(1,0)) \\ c_l(x + h_l\,(-1,-1)) & b_l(x + h_l\,(0,-1)) & b_l(x + h_l\,(1,-1)) \end{bmatrix}. \quad (3.2.1)$$

3.3 Properties of the Coarse Grid Equation

LEMMA 3.3.1 *$P_{l,l-1}$ defined by (3.2.1) satisfies*

$$P_{l,l-1}\,1_{l-1} = 1_l \;\Leftrightarrow\; \begin{cases} a_l(x) + b_l(x) = 1 & : \quad x \in \Omega_{l,(1,0)} \cup \Omega_{l,(0,1)}, \\ a_l(x) + b_l(x) + c_l(x) + d_l(x) = 1 & : \quad x \in \Omega_{l,(1,1)}, \end{cases}$$

where $1_l = (1, \ldots, 1)^T \in \mathbb{R}^{n_l}$.

PROOF. Straightforward computation. □

LEMMA 3.3.2 *Assume $P_{l,l-1}\,1_{l-1} = 1_l$, $R_{l-1,l} = P^T_{l,l-1}$, and $f_{l-1} = R_{l-1,l}\,f_l$. Then, f_{l-1} and $A_{l-1} = R_{l-1,l}\,A_l\,P_{l,l-1}$ have the following properties:*

(i) $\sum_{i=1}^{n_{l-1}} (f_{l-1})_i = \sum_{i=1}^{n_l} (f_l)_i,$

(ii) $\sum_{i,j=1}^{n_{l-1}} (A_{l-1})_{i,j} = \sum_{i,j=1}^{n_l} (A_l)_{i,j},$

(iii) $\sum_{j=1}^{n_l} (A_l)_{i,j} = 0 \;\; \forall i \quad \Rightarrow \quad \sum_{j=1}^{n_{l-1}} (A_{l-1})_{i,j} = 0 \;\forall i,$

(iv) $\sum_{i=1}^{n_l} (A_l)_{i,j} = 0 \;\; \forall j \quad \Rightarrow \quad \sum_{i=1}^{n_{l-1}} (A_{l-1})_{i,j} = 0 \;\; \forall j,$

(v) $A_l = A_l^T \quad \Rightarrow \quad A_{l-1} = A^T_{l-1}.$

PROOF.

(i) $1^T_{l-1}\,f_{l-1} = 1^T_{l-1}\,R_{l-1,l}\,f_l = (P_{l,l-1}\,1_{l-1})^T\,f_l = 1_l^T\,f_l.$

(ii) $1^T_{l-1}\,A_{l-1}\,1_{l-1} = 1^T_{l-1}\,R_{l-1,l}\,A_l\,P_{l,l-1}\,1_{l-1} = (P_{l,l-1}\,1_{l-1})^T\,A_l\,1_l = 1_l^T\,A_l\,1_l.$

(iii) $A_{l-1}\,1_{l-1} = R_{l-1,l}\,A_l\,P_{l,l-1}\,1_{l-1} = R_{l-1,l}\,A_l\,1_l = R_{l-1,l}0_l = 0_{l-1}.$

(iv) $A^T_{l-1}\,1_{l-1} = (R_{l-1,l}\,A_l\,P_{l,l-1})^T\,1_{l-1} = R_{l-1,l}\,A_l^T\,1_l = R_{l-1,l}0_l = 0_{l-1}.$

(v) $$\begin{aligned} A^T_{l-1} &= (R_{l-1,l}\,A_l\,P_{l,l-1})^T = P^T_{l,l-1}\,A_l^T\,R^T_{l-1,l} = R_{l-1,l}\,A_l^T\,P_{l,l-1} \\ &= R_{l-1,l}\,A_l\,P_{l,l-1} = A_{l-1}. \end{aligned}$$

□

For a generalization of part (iii) and (iv) in Lemma 3.3.2, we refer to [de Zeeuw 1990].

3.4 Matrix-Dependent Prolongation

The system matrix A_l is split into a symmetric matrix S_l and an asymmetric matrix Z_l.

$$\begin{aligned} S_l &= \frac{1}{2}(A_l + A_l^T), \\ Z_l &= \frac{1}{2}(A_l - A_l^T). \end{aligned}$$

This corresponds to splitting the stencil of A_l at $x \in \Omega_l$

$$A_l = \begin{bmatrix} A_l(x)(-1,1) & A_l(x)(0,1) & A_l(x)(1,1) \\ A_l(x)(-1,0) & A_l(x)(0,0) & A_l(x)(1,0) \\ A_l(x)(-1,-1) & A_l(x)(0,-1) & A_l(x)(1,-1) \end{bmatrix} \tag{3.4.1}$$

as follows

$$\begin{aligned} A_l(x)(i,j) &= S_l(x)(i,j) + Z_l(x)(i,j), \qquad (3.4.2) \\ S_l(x)(i,j) &= \frac{1}{2}[A_l(x)(i,j) + A_l(x + h_l(i,j))(-i,-j)], \\ Z_l(x)(i,j) &= \frac{1}{2}[A_l(x)(i,j) - A_l(x + h_l(i,j))(-i,-j)]. \end{aligned}$$

(3.4.2) is rewritten as

$$\begin{bmatrix} a_7 & a_8 & a_9 \\ a_4 & a_5 & a_6 \\ a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} s_7 & s_8 & s_9 \\ s_4 & s_5 & s_6 \\ s_1 & s_2 & s_3 \end{bmatrix} + \begin{bmatrix} z_7 & z_8 & z_9 \\ z_4 & z_5 & z_6 \\ z_1 & z_2 & z_3 \end{bmatrix}.$$

The symmetric part is decomposed by

$$\begin{aligned} \begin{bmatrix} s_7 & s_8 & s_9 \\ s_4 & s_5 & s_6 \\ s_1 & s_2 & s_3 \end{bmatrix} &= -s_{147}\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - s_{369}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \qquad (3.4.3) \\ &\quad -s_{123}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} - s_{789}\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &\quad -s_1\begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ -1 & 1 & 0 \end{bmatrix} + s_7\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &\quad +s_3\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} - s_9\begin{bmatrix} 0 & 1 & -1 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \Sigma\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

where $s_{qrs} = s_q + s_r + s_s$ and $\Sigma = \sum_{i=1}^{9} s_i$. The elementary stencils in (3.4.3) can be identified as discretizations of the differential operators $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, $\frac{\partial^2}{\partial x\, \partial y}$. Thus, schematically, the diffusion coefficients $D(x)$ in the differential operator $\nabla \cdot (D(x)\nabla)$ are found to be as in Figure 3.2. Similarly, the coefficients of the convection term $\nabla \cdot (v\, u)$ (see (1.2.1)) are approximated by

$$\begin{aligned} c_1 &= (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7), \\ c_2 &= (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3). \end{aligned}$$

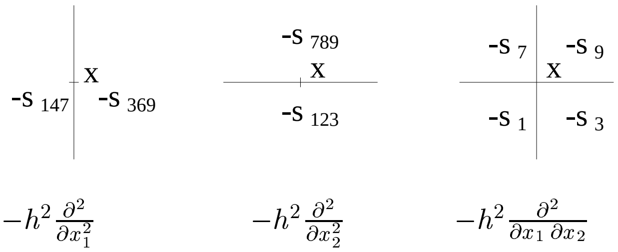

Figure 3.2: Diffusion coefficients.

Assume $a_l(x)$ and $b_l(x)$ at $\Omega_{l,(0,1)}$ and $\Omega_{l,(1,0)}$ have already been constructed. In order to get smooth vectors after the interpolation, the prolongation is supposed to satisfy

$$\sum_{i,j=-1}^{1} A_l(x)(i,j)\,(P_{l,l-1}\,u_{l-1})(x + h_l(i,j)) = 0 \quad \forall x \in \Omega_{l,(1,1)}.$$

Substituting the weights $a_l(x)$ and $b_l(x)$ at $\Omega_{l,(0,1)}$ and $\Omega_{l,(1,0)}$ as defined in (3.2.1) leads for $x \in \Omega_{l,(1,1)}$ to

$$\begin{aligned} a_l(x) &= -[A_l(x)(1,-1) + A_l(x)(0,-1) \cdot a_l(x + h_l(0,-1)) \\ &\quad + A_l(x)(1,0) \cdot a_l(x + h_l(1,0))][A_l(x)(0,0)]^{-1}, & (3.4.4) \\ b_l(x) &= -[A_l(x)(-1,1) + A_l(x)(-1,0) \cdot b_l(x + h_l(-1,0)) \\ &\quad + A_l(x)(0,1) \cdot b_l(x + h_l(0,1))][A_l(x)(0,0)]^{-1}, & (3.4.5) \\ c_l(x) &= -[A_l(x)(1,1) + A_l(x)(1,0) \cdot b_l(x + h_l(1,0)) \\ &\quad + A_l(x)(0,1) \cdot a_l(x + h_l(0,1))][A_l(x)(0,0)]^{-1}, & (3.4.6) \\ d_l(x) &= -[A_l(x)(-1,-1) + A_l(x)(0,-1) \cdot b_l(x + h_l(0,-1)) \\ &\quad + A_l(x)(-1,0) \cdot a_l(x + h_l(-1,0))][A_l(x)(0,0)]^{-1}. & (3.4.7) \end{aligned}$$

REMARK 3.4.1 *If $b_l(x+h_l\,z)+a_l(x+h_l\,z) = 1$ for $z \in \{(-1,0),(1,0),(0,-1),(0,1)\}$ and $A_l(x)(0,0) \neq 0$, then*

$$a_l(x) + b_l(x) + c_l(x) + d_l(x) = 1 - \frac{\sum_{i,j=-1}^{1} A_l(x)(i,j)}{A_l(x)(0,0)}$$

holds for $x \in \Omega_{l,(1,1)}$. In addition, if $A_l(x)(i,j) \leq 0$ for $(i,j) \neq (0,0)$, $A_l(x)(0,0) > 0$, and both $b_l(x + h_l\,z), a_l(x + h_l\,z) \geq 0$, then $a_l(x), b_l(x), c_l(x), d_l(x) \geq 0$.

PROOF. The remark follows immediately from (3.4.4) – (3.4.7). □

The same procedure is applied in the horizontal and the vertical direction. Therefore, we restrict the description to the horizontal direction. In order to simplify the notation, we write for some $x \in \Omega_{l,(1,0)}$

$$(P_{l,l-1}\, u_{l-1})(x) = w_W\, u_W + w_E\, u_E,$$

with $u_W = u_{l-1}(x + h_l(-1,0))$ and $u_E = u_{l-1}(x + h_l(1,0))$.

To determine the weights w_W and w_E, we formulate some guidelines for their construction.

(i)
$$w_W + w_E = 1 - \frac{\sum_{i,j=-1}^{1} A_l(x)(i,j)}{A_l(x)(0,0)} \quad \text{and} \quad w_W, w_E \geq 0.$$

In Section 3.3, we have seen that we should satisfy $w_W + w_E = 1$ for any matrix with $\sum_{i,j=-1}^{1} A_l(x)(i,j) = 0$.

(ii) In the simple case

$$A_l(x)(i,j) \begin{cases} = 0 & : \quad (i,j) \neq (0,0), \\ \neq 0 & : \quad (i,j) = (0,0), \end{cases}$$

$w_W + w_E = 0$ is required. This is an optimal choice, because (local) smoothing solves the equation at once and any nonzero coarse grid correction would be harmful.

(iii) In the one-dimensional case,

$$\begin{gathered} s_4 = -D_W, \quad s_6 = -D_E, \quad s_5 = D_W + D_E + \Sigma, \\ z_4 = -\tfrac{1}{2}c_1, \quad z_6 = +\tfrac{1}{2}c_1, \\ s_i = z_i = 0 \text{ for } i < 4 \text{ or } i > 6, \end{gathered}$$

the prolongation should reduce to interpolation by means of the differential operator. This is achieved by

$$w_W = \frac{D_W + \frac{1}{2}c_1}{D_W + D_E + \Sigma}, \quad w_E = \frac{D_E - \frac{1}{2}c_1}{D_W + D_E + \Sigma}.$$

With these guidelines in mind, the following formulas for w_W and w_E are proposed

$$\begin{aligned} D_W &= \max(|s_{147}|, |s_1|, |s_7|), \quad D_E = \max(|s_{369}|, |s_3|, |s_9|), \\ D_N &= \max(|s_{789}|, |s_7|, |s_9|), \quad D_S = \max(|s_{123}|, |s_1|, |s_3|), \\ \sigma &= \min\left(1, |1 - \frac{\Sigma}{a_5}|\right), \\ w'_W &= \sigma\left(\frac{D_W}{D_W + D_E} + \frac{1}{2}\frac{c_1}{D_W + D_E + D_N + D_S}\right), \end{aligned}$$

$$\begin{aligned}
w'_E &= \sigma\left(\frac{D_E}{D_W+D_E} - \frac{1}{2}\frac{c_1}{D_W+D_E+D_N+D_S}\right), \\
w_W &= \min(\sigma, \max(0, w'_W)), && (3.4.8)\\
w_E &= \min(\sigma, \max(0, w'_E)). && (3.4.9)
\end{aligned}$$

Remark 3.4.2

$$0 \le w_W \le \sigma \le 1, \quad 0 \le w_E \le \sigma \le 1.$$

Example 3.4.1 *Let a differential operator* $\mathcal{A}$

$$\begin{aligned}
\mathcal{A}\, u &= \nabla\cdot(D(x)\,\nabla u(x)), \quad x\in\Omega=(0,1)\times(0,1)\\
D(x) &= \begin{cases} D_L & : \ x_1<\xi, \\ D_R & : \ x_1\ge\xi, \end{cases} \quad \xi<1,
\end{aligned}$$

be given. Then,

$$(P_{l,l-1}\, u_{l-1})(\xi) = w_W\, u_W + w_E\, u_E,$$

with

$$w_W = \frac{D_L}{D_L+D_R}, \quad w_E = \frac{D_R}{D_L+D_R}.$$

3.5 The Coarse Grid Stencil

In order to apply the same procedure recursively on all levels, the coarse grid matrix $A_H = R\,A\,P$ needs to have a 9-point stencil as the matrix A.

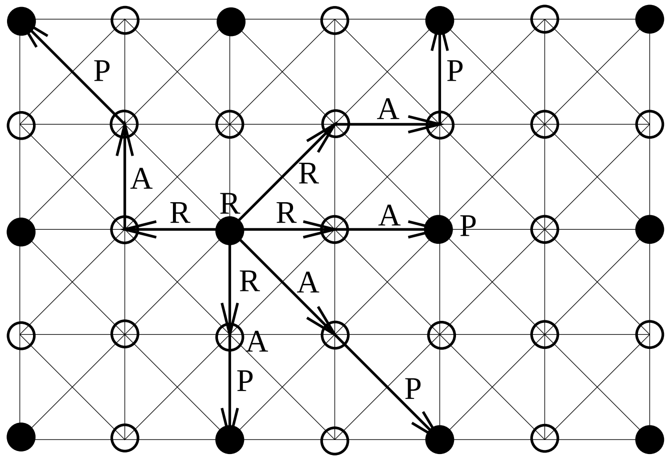

Figure 3.3: Construction of the coarse grid stencil.

Figure 3.3 illustrates how the coarse grid stencil is obtained. Starting at a node which exists on the coarser grid too (C-node, black dot), we move according to the restriction to an F-node (circle) which only exists on the fine grid or we stay at the coarse grid node. Thus, at most, we can move to a neighbor node.

The next step following the non-zero entries in the system matrix again leads us either to a neighbor node or we stay at the same node (diagonal entry). If we have arrived at an F-node, due to the prolongation, we move to an C-node neighbor. If we have arrived at an C-node, we do not move.

Hence, all together, starting at a C-node we can only move 3 steps following the non-zero matrix entries. Only the direct neighbors on the coarse grid of a C-node are in this range. Therefore, we will get a 9-point stencil on the coarser grid.

3.6 Similar Methods

Alcouffe, Brandt, Dendy and Painter propose in [Alcouffe, Brandt, Dendy, and Painter 1981] a similar approach for the construction of matrix-dependent transfer operators. For the matrix stencil (3.4.1), the weights

$$w_W = \frac{-A_l(x)(-1,1) - A_l(x)(-1,0) - A_l(x)(-1,1)}{A_l(x)(0,0) + A_l(x)(0,-1) + A_l(x)(0,1)}, \tag{3.6.1}$$

$$w_E = \frac{-A_l(x)(1,1) - A_l(x)(1,0) - A_l(x)(1,1)}{A_l(x)(0,0) + A_l(x)(0,-1) + A_l(x)(0,1)} \tag{3.6.2}$$

are suggested. Note that in the symmetric case $A_l = A_l^T$ if

$$A_l(x)(i,j) \le 0 \text{ for } (i,j) \ne (0,0), \quad A_l(x)(0,0) > 0, \quad \text{and} \quad \sum_{i,j=-1}^{1} A_l(x)(i,j) = 0$$

(3.4.8), (3.4.9) and (3.6.1), (3.6.2) are equivalent.

The weights for $x \in \Omega_{l,(1,1)}$ in [Alcouffe, Brandt, Dendy, and Painter 1981] and the weights (3.4.4)–(3.4.7) in [de Zeeuw 1990] are identical.

3.7 Numerical Experiments

Experiment 3.7.1

$$\begin{aligned} -\nabla \cdot (D(x)\nabla u(x)) &= 1, \quad x \in \Omega = (1,25) \times (1,25), \\ D\frac{\partial u}{\partial n} &= -\frac{1}{2}u, \quad x \in \partial\Omega \end{aligned}$$

is discretized on an equidistant 25×25 *grid. Four grid levels are used.* $D(x) = D_1 = 1/3$ *outside the shaded region and* $D(x) = D_2 = 10^4\, D_1$ *inside the shaded region (see Figure 3.4).*

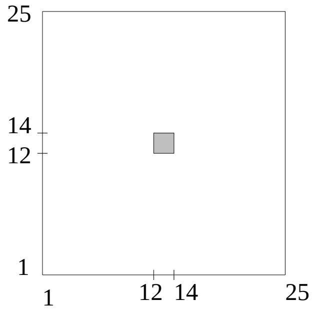

Figure 3.4: Diffusion coefficient.

De Zeeuw [de Zeeuw 1990] compares the numbers of iteration steps for his method (MGDeZeeuw), the standard multigrid method (MGM), and an algorithm proposed by Kettler [Kettler 1981] (MGKettler) for a 10^{-9} reduction of the residual ($\|d^{(m)}\|_2 < 10^{-9}\,\|d^{(0)}\|_2$). The results are shown in Table 3.1. In all algorithms, 1 post-smoothing step with ILLU (block-ILU factorization, see [Kettler 1981]) is computed.

MGM	MGDeZeeuw	MGKettler
52	8	19

Table 3.1: Iteration steps for Experiment 3.7.1.

Alcouffe et al. report convergence rates of their algorithm (MGABDP) proposed in [Alcouffe, Brandt, Dendy, and Painter 1981] and of the SOR method for the same problem (see Table 3.2). In MGABDP, 1 pre- and 1 post-smoothing step with Gauß-Seidel is applied.

D_2/D_1	MGABDP	SOR
10^{-4}	0.486	0.731
10^{-2}	0.486	0.731
10^{0}	0.490	0.744
10^{2}	0.532	0.875
10^{4}	0.537	0.990

Table 3.2: Convergence rates for Experiment 3.7.1.

Experiment 3.7.2 *The staircase problem*

$$-\nabla\cdot(D(x)\nabla u(x)) \;=\; f, \quad x\in\Omega=(1,17)\times(1,17),$$

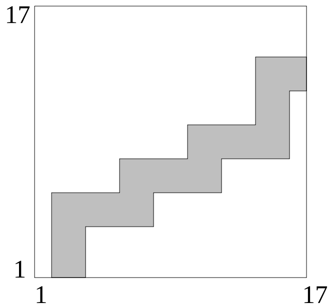

Figure 3.5: Diffusion coefficient for the staircase problem.

$$D\frac{\partial u}{\partial n} = -\frac{1}{2}u, \quad x \in \partial\Omega$$

is discretized on an equidistant 17×17 *grid.* $D(x) = D_1 = 10^3$ *and* $f(x) = f_1 = 1$ *inside the shaded region;* $D(x) = D_2 = 1$ *and* $f(x) = f_2 = 0$ *outside the shaded region (see Figure 3.5). Three grid levels are used.*

MGM	MGDeZeeuw	MGKettler
130	9	10

Table 3.3: Iteration steps for Experiment 3.7.2.

De Zeeuw's results are summarized in Table 3.3. The average convergence rate of MGABDP is 0.59.

4 Interpolation Methods

For some problems, the grid hierarchy applied in standard multigrid methods may not be optimal. Therefore, an important feature of algebraic multigrid methods is the possibility to choose coarse grids which are more appropriate for the given problem.

The algebraic multigrid methods discussed in this chapter combine the multigrid algorithm (Algorithm 2.3.2), matrix-dependent transfer operators, and strategies for the selection of the coarse grids.

Since it is a priori not known how many coarse grids will be constructed, the coarse levels are now numbered in the opposite direction. The finest grid and the corresponding system matrix are called Ω_0 and $A_0 = A$ respectively. Ω_{l+1} denotes the grid which arises from the coarsening of Ω_l. Therefore, the AMG cycle reads as follows.

ALGORITHM 4.0.1 *Let a hierarchy of grids $\Omega_0 \supset \Omega_1 \supset \cdots \supset \Omega_{l_{\max}}$, prolongation and restriction operators $P_{l,l+1}, R_{l+1,l}$ between these grids, and matrices A_l be constructed and smoothing iterations S_l on these grids be given. Then, the function* `AMG`$(u_0, f_0, 0)$ *defines the approximate inverse M_{AMG}^{-1} on the finest grid Ω_0.*

```
AMG(u_l, f_l, l)
{
    if(l = l_max)  u_l = A_l^{-1} f_l;
    else
    {
        u_l = S_l^{ν1}(u_l, f_l);
        d_{l+1} = R_{l,l+1} (f_l − A_l u_l);
        v_{l+1} = 0;
        for(j = 0;  j < γ;  j = j + 1)  AMG(v_{l+1}, d_{l+1}, l + 1);
        u_l = u_l + P_{l,l+1} v_{l+1};
        u_l = S_l^{ν2}(u_l, f_l);
    }
}
```

4.1 An Anisotropic Model Problem

A discretization of the anisotropic differential equation

$$\begin{aligned} -\nabla \cdot (D(x)\nabla u(x)) &= f(x), \quad x \in \Omega = (0,1) \times (0,1) \\ u(x) &= g(x), \quad x \in \partial\Omega, \\ D(x) &= \begin{pmatrix} \epsilon & 0 \\ 0 & 1 \end{pmatrix} \end{aligned} \tag{4.1.1}$$

on the equidistant grid

$$\Omega_h = \{\, x_{(i,j)} \in \Omega \,|\, x_{(i,j)} = (i\,h, j\,h),\ i,j = 1,\ldots,n,\ h = \frac{1}{n+1}\,\}$$

produces the system matrix A

$$A = \begin{bmatrix} 0 & -1 & 0 \\ -\epsilon & 2+2\epsilon & -\epsilon \\ 0 & -1 & 0 \end{bmatrix}. \tag{4.1.2}$$

The eigenvectors $e^{(\nu,\mu)}$ and eigenvalues $\lambda^{(\nu,\mu)}$of the damped Jacobi iteration S for A are given by

$$\begin{aligned} e^{(\nu,\mu)}_{(i,j)} &= \sin(\pi\nu hi)\,\sin(\pi\mu hj), \quad \nu,\mu = 1,\ldots,n, \\ \lambda^{(\nu,\mu)}(A) &= 2+2\epsilon - 2\epsilon\cos(\pi\nu h) - 2\cos(\pi\mu h), \\ \lambda^{(\nu,\mu)}(S) &= 1-\omega-\omega\frac{2}{2+2\epsilon}(\epsilon\cos(\pi\nu h)+\cos(\pi\mu h)). \end{aligned}$$

Figure 4.1 shows $\lambda^{(\nu,\mu)}(S)$ for $\omega = 0.5$, $\epsilon = 1$ and, $\epsilon = 0.001$.

Obviously, for ($\epsilon = 0.001$), the high frequency modes in the vertical direction (j, μ) are efficiently damped. But, the damping in the horizontal direction is almost independent of the frequency (ν). Thus, error modes with high frequency components in the horizontal direction and low frequency components in the vertical direction are not reduced. Hence, after a couple of smoothing steps, the error is smooth in the direction of the strong coupling of the unknowns (vertical direction). In the direction of the weak coupling (horizontal direction), the error may still be rough. Therefore, (linear) interpolation in the direction of the weak coupling of the unknowns will not be successful.

Robust smoothers (see Section 2.6) like ILU_β [Wittum 1989b] or line smoothers [Kettler 1981] can be combined with standard multigrid in order to get an robust method for anisotropic problems. However, ILU_β and especially line smoothers are only efficient for structured grids and anisotropy in one direction only. In [Kettler and Wesseling 1986], it is shown that line smoother are not robust for problems in three spatial dimensions.

If standard point smoothers are used, interpolation in the direction of the weak coupling must be avoided. Thus, algebraic multigrid methods typically do not coarsen the grid in the direction of the weak coupling. For structured grids and anisotropy in one direction only, this technique is called semi-coarsening (see Figure 4.2). But in contrast to line smoothers, this technique is not restricted to these cases.

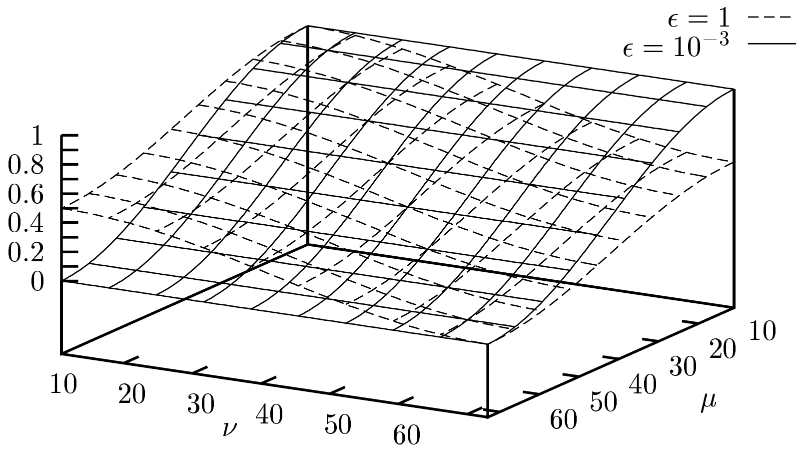

Figure 4.1: Damped Jacobi iteration for an anisotropic problem ($h = 1/64$).

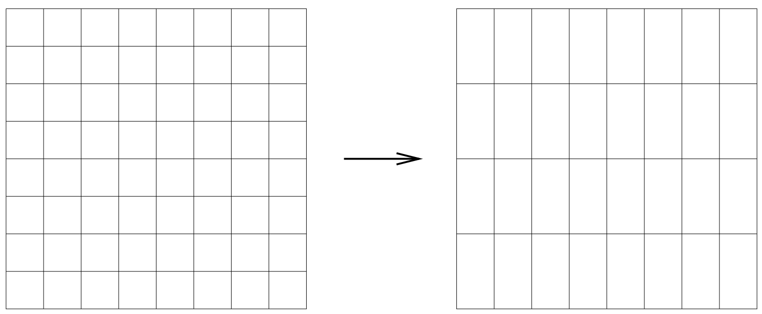

Figure 4.2: Semi-coarsening.

4.2 Algebraic Smoothness

Section 4.2 – 4.6 follow the classic paper of Ruge and Stüben [Ruge and Stüben 1987]. For the theoretical discussion, the system matrix A is always assumed to be symmetric and positive definite. The prolongation operators P must have full rank and the restriction R and the coarse grid matrices A_H are defined by

$$R = P^T, \quad A_H = R\,A\,P.$$

Along with their corresponding norms $\|u\|_{\mathcal{H}^i} = \sqrt{(u,u)_{\mathcal{H}^i}}$, $i \in \{0,1,2\}$, the scalar products

$$(u,v)_{\mathcal{H}^0} = (D\,u,v)_2, \quad (u,v)_{\mathcal{H}^1} = (A\,u,v)_2, \quad (u,v)_{\mathcal{H}^2} = (D^{-1}A\,u,A\,v)_2$$

are needed. $D = \operatorname{diag}(A)$ denotes the diagonal of A. In some sense, these norms are discrete counterparts of the H^k-Sobolev (semi-)norms (see e.g. [Braess 1997; Hackbusch 1986]).

In algebraic multigrid, an error e is called smooth if it is slow to converge with respect to a smoother S. For example

$$\|S\,e\|_{\mathcal{H}^1} \approx \|e\|_{\mathcal{H}^1}.$$

Depending on A an algebraically smooth error may well be highly oscillation geometrically.

For typical relaxation schemes like Gauß-Seidel, the inequality

$$\|S\,e\|^2_{\mathcal{H}^1} \leq \|e\|^2_{\mathcal{H}^1} - \alpha\|e\|^2_{\mathcal{H}^2} \tag{4.2.1}$$

holds with $\alpha > 0$ (e.g. $\alpha = 1/4$) [Ruge and Stüben 1987]. Therefore, an smooth error has to satisfy $\|e\|_{\mathcal{H}^2} \ll \|e\|_{\mathcal{H}^1}$. Applying the Cauchy-Schwarz inequality to $(Ae,e)_2 = (D^{-1/2}Ae, D^{1/2}e)_2$ shows

$$\begin{aligned} \|e\|^2_{\mathcal{H}^1} &= (D^{-1/2}\,A\,e, D^{1/2}\,e)_2 \\ &\leq \|D^{-1/2}\,A\,e\|_2 \|D^{1/2}\,e\|_2 = \|e\|_{\mathcal{H}^2}\,\|e\|_{\mathcal{H}^0}. \end{aligned}$$

Therefore, $\|e\|_{\mathcal{H}^2} \ll \|e\|_{\mathcal{H}^1}$ implies $\|e\|_{\mathcal{H}^1} \ll \|e\|_{\mathcal{H}^0}$, or more explicitly

$$\begin{aligned} (A\,e,e)_2 &= \frac{1}{2}\sum_{i,j} -a_{i,j}(e_i - e_j)^2 + \frac{1}{2}\sum_{i,j} a_{i,j}e_i^2 + \frac{1}{2}\sum_{i,j} a_{i,j}e_j^2 \\ &= \frac{1}{2}\sum_{i,j} -a_{i,j}(e_i - e_j)^2 + \sum_i \left(\sum_j a_{i,j}\right) e_i^2 \ll \sum_i a_{i,i}e_i^2. \end{aligned} \tag{4.2.2}$$

For the important case $\sum_{i \neq j} |a_{i,j}| \approx a_{i,i}$, this means that, on the average for each i,

$$\begin{aligned} \frac{1}{2}\sum_{j \neq i} -a_{i,j}(e_i - e_j)^2 &\ll a_{i,i}e_i^2, \\ \sum_{j \neq i} \frac{|a_{i,j}|}{a_{i,i}} \frac{(e_i - e_j)^2}{e_i^2} &\ll 2. \end{aligned}$$

In other words, a smooth error generally varies slowly in the direction of strong connections, i.e. from e_i to e_j if $\frac{|a_{i,j}|}{a_{i,i}}$ is relatively large.

4.3 Two-Level Convergence

Without essential loss of generality, we restrict our discussion of the two-level convergence to the case with post-smoothing only. Assuming the smoothing property (4.2.1), we will give a general condition on the prolongation P that allows an estimation of the two-level convergence factor $\|S\,T\|_{\mathcal{H}^1}$.

THEOREM 4.3.1 *Let A be spd, $T = I - P\,(R\,A\,P)^{-1}\,R\,A$, $R = P^T$ and let S satisfy the smoothing property (4.2.1). Suppose that the interpolation P has full rank and that for each e*

$$\min_{e_H} \|e - P\,e_H\|^2_{\mathcal{H}^0} \leq \beta\,\|e\|^2_{\mathcal{H}^1} \tag{4.3.1}$$

with $\beta > 0$ independent of e. Then, $\alpha \leq \beta$ and the two-level convergence factor satisfies

$$\|S\,T\|_{\mathcal{H}^1} \leq \sqrt{1 - \frac{\alpha}{\beta}}.$$

PROOF. Since range(T) is orthogonal to range(P) with respect to $(\cdot,\cdot)_{\mathcal{H}^1}$,

$$\|e\|^2_{\mathcal{H}^1} = (Ae, e - P\,e_H)_2 \quad \forall e \in \text{range}(T).$$

Using the Cauchy-Schwarz inequality, we get

$$\begin{aligned} \|e\|^2_{\mathcal{H}^1} &= ((D^{-1/2}\,A\,e, D^{1/2}\,(e - P\,e_H))_2 \\ &\leq \|D^{-1/2}\,A\,e\|_2 \|D^{1/2}\,(e - P\,e_H)\|_2 \\ &= \|e\|_{\mathcal{H}^2}\,\|e - P\,e_H\|_{\mathcal{H}^0}. \end{aligned}$$

Because of (4.3.1), this proofs $\|e\|^2_{\mathcal{H}^1} \leq \beta\,\|e\|^2_{\mathcal{H}^2}$ for all $e \in \text{range}(T)$. Therefore,

$$\|T\,e\|^2_{\mathcal{H}^1} \leq \beta\,\|T\,e\|^2_{\mathcal{H}^2} \quad \forall e$$

holds. Finally,

$$\begin{aligned} 0 \leq \|S\,T\,e\|^2_{\mathcal{H}^1} &\leq \|T\,e\|^2_{\mathcal{H}^1} - \alpha\|T\,e\|^2_{\mathcal{H}^2} \\ &\leq \|T\,e\|^2_{\mathcal{H}^1} - \frac{\alpha}{\beta}\|T\,e\|^2_{\mathcal{H}^1} \\ &\leq \left(1 - \frac{\alpha}{\beta}\right)\|T\,e\|^2_{\mathcal{H}^1} \leq \left(1 - \frac{\alpha}{\beta}\right)\|e\|^2_{\mathcal{H}^1} \end{aligned}$$

proofs the theorem. □

Now, we consider the graph $G_A(V,E)$ (see Section 1.2) of the system matrix A. Let a splitting of the set of vertices (nodes/unknowns) $V = \{v_1, \ldots, v_n\}$ into a set C of C-vertices (C-nodes/C-unknowns) and a set F of F-vertices (F-nodes/F-unknowns)

$$V = C \cup F = C \oplus F$$

be given, Then, the C-unknowns build the coarse grid $\Omega_H = C$. An algorithm for the labeling of the nodes as C-node or F-node is presented in Section 4.6. To simplify the notation, we write

$$\sum_{v_j \in C} \text{ or } \sum_{v_i \in F} \quad \text{instead of} \quad \sum_{j \text{ with } v_j \in C} \text{ or } \sum_{i \text{ with } v_i \in F}$$

in the remainder of the course notes.

An interpolation (prolongation) P from the coarse grid to the fine grid is defined by

$$(P\, e_H)_i = \sum_{v_j \in C} w_{i,j}\, (e_H)_j, \quad v_i \in V \quad \text{and } w_{i,j} = \delta_{i,j} \quad \text{if } v_i \in C. \tag{4.3.2}$$

$\delta_{i,j}$ denotes the Kronecker symbol.

LEMMA 4.3.1 *Assume A is spd and P is of the form (4.3.2) with $w_{i,j} \geq 0$. Then, property (4.3.1) is satisfied with $\beta > 0$ for any given set C of C-unknowns if the two inequalities*

$$\sum_{v_i \in F} \sum_{v_j \in C} a_{i,i}\, w_{i,j}\, (e_i - e_j)^2 \;\leq\; \frac{\beta}{2} \sum_{i,j} -a_{i,j}\, (e_i - e_j)^2, \tag{4.3.3}$$

$$\sum_{v_i \in F} a_{i,i}\, (1 - s_i)\, e_i^2 \;\leq\; \beta \sum_i \left(\sum_j a_{i,j} \right) e_i^2 \tag{4.3.4}$$

hold with $s_i = \sum_{v_j \in C} w_{i,j} \leq 1$.

PROOF.

$$\begin{aligned}
\min_{e_H} \|e - P\, e_H\|_{\mathcal{H}^0}^2 &\leq \|e - P\, R_{\text{inj}}\, e\|_{\mathcal{H}^0}^2 = \sum_{v_i \in F} a_{i,i} \left(e_i - \sum_{v_j \in C} w_{i,j}\, e_j \right)^2 \\
&= \sum_{v_i \in F} a_{i,i} \left(\sum_{v_j \in C} w_{i,j}\, (e_i - e_j) + (1 - s_i)\, e_i \right)^2 \\
&\leq \sum_{v_i \in F} a_{i,i} \left(\sum_{v_j \in C} w_{i,j}^2\, (e_i - e_j)^2 + (1 - s_i)^2\, e_i^2 \right) \\
&\leq \sum_{v_i \in F} a_{i,i} \left(\sum_{v_j \in C} w_{i,j}\, (e_i - e_j)^2 + (1 - s_i)\, e_i^2 \right).
\end{aligned}$$

(4.2.2), (4.3.3) and, (4.3.4) show

$$\begin{aligned}
&\sum_{v_i \in F} a_{i,i} \left(\sum_{v_i \in C} w_{i,j} (e_i - e_j)^2 + (1 - s_i)\, e_i^2 \right) \\
&\qquad \leq \beta \left(\frac{1}{2} \sum_{i,j} -a_{i,j}\, (e_i - e_j)^2 \right) + \beta \sum_i \left(\sum_j a_{i,j} \right) e_i^2 = \beta\, \|e\|_{\mathcal{H}^1}^2.
\end{aligned}$$

Hence, (4.3.3) and (4.3.4) are sufficient for (4.3.1). □

4.4 Interpolation Operators

The tools developed in the previous section are now used to construct interpolation operators which guarantee uniform two-level convergence. Throughout this section, A is assumed to be a symmetric, weakly diagonally dominant ($a_{i,i} \geq \sum_{j\neq i} |a_{i,j}|$) M-matrix.

The neighborhood N_i of the node v_i is defined by (see Section 1.2)

$$N_i = \{ v_j \in \Omega_h \mid j \neq i,\ a_{i,j} \neq 0 \}. \tag{4.4.1}$$

As a first step, we consider interpolation nodes $v_j \in C_i$ with $C_i \subseteq N_i \cap C$ and corresponding weights of the form

$$w_{i,j} = \eta_i \, |a_{i,j}|, \quad 0 \leq \eta_i \leq \frac{1}{\sum\limits_{v_j \in C} |a_{i,j}|}, \tag{4.4.2}$$

which ensures $s_i \leq 1$. In order for property (4.3.1) to hold, it is, due to Lemma 4.3.1, obviously sufficient to require for every $v_i \in F$, $v_j \in C_i$

$$0 \leq a_{i,i} \, w_{i,j} \leq \beta \, |a_{i,j}|, \quad 0 \leq a_{i,i} \, (1 - s_i) \leq \beta \sum_j a_{i,j}. \tag{4.4.3}$$

From these simple inequalities, more practical condition can be derived, which can be used to develop an automatic coarsening algorithm with β as input parameter. An example is given in the following remark.

REMARK 4.4.1 *Let $\beta \geq 1$ be fixed and the interpolation weights be given by (4.4.2) with $\eta_i = 1/ \sum\limits_{v_j \notin C_i} a_{i,j}$. Assume that A is a symmetric, weakly diagonally dominant M-matrix and the C-points are picked such that for each $v_i \in F$ there is a nonempty set $C_i \subseteq N_i \cap C$ with*

$$\beta \sum_{v_j \notin C_i} a_{i,j} \geq a_{i,i}.$$

Then, (4.3.1) is satisfied.

PROOF.

$$a_{i,i} \, w_{i,j} = a_{i,i} \, \eta_i \, |a_{i,j}| = \frac{a_{i,i}}{\sum\limits_{v_j \notin C_i} a_{i,j}} |a_{i,j}| \leq \beta \, |a_{i,j}|,$$

$$a_{i,i} \, (1 - s_i) = a_{i,i} \left(1 - \frac{\sum\limits_{v_j \in C_i} |a_{i,j}|}{\sum\limits_{v_j \notin C_i} a_{i,j}} \right) = a_{i,i} \, \frac{\sum\limits_j a_{i,j}}{\sum\limits_{v_j \notin C_i} a_{i,j}} \leq \beta \sum_j a_{i,j}.$$

□

More general interpolation weights can be obtained by

$$w_{i,j} = \eta_i \left(|a_{i,j}| + \sum_{v_k \in D_i} \eta_{i,j}^{(k)} \, |a_{i,k}| \right), \quad v_i \in F, \quad v_j \in C_i, \tag{4.4.4}$$

where $D_i = N_i \setminus (C_i \cap N_i)$ denotes the set of neighbor nodes which are not in C_i. Here, a weighted distribution of all noniterpolatory connections $a_{i,k}$, $v_k \in D_i$ to the interpolation points $v_j \in C_i$ is performed. C_i is not required to be a subset of N_i. Thus, long range interpolation from points without a direct connection is allowed.

REMARK 4.4.2 *If A is a symmetric, weakly diagonally dominant M-matrix, C, $C_i \subseteq C$ are picked such that for each k and $v_j \in N_k \cap C$*

$$\zeta \sum_{v_m \in C_i} |a_{k,m}| \geq \sum_{v_t \in F_{k,j}} |a_{k,t}|, \quad v_i \in F_{k,j}, \quad F_{k,j} = \{\, v_i \in F \mid v_j \in C_i \wedge v_k \in D_i \,\},$$

for a $\zeta > 0$ and the weights $w_{i,j}$ are defined by (4.4.4) and

$$\eta_i = \frac{1}{a_{i,i}}, \quad \eta_{i,j}^{(k)} = \frac{|a_{k,j}|}{\sum\limits_{v_m \in C_i} |a_{k,m}|}, \tag{4.4.5}$$

then, property (4.3.1) is satisfied with some β which only depends on ζ but not on A.

PROOF. See the proof of Theorem 5.5 in [Ruge and Stüben 1987]. □

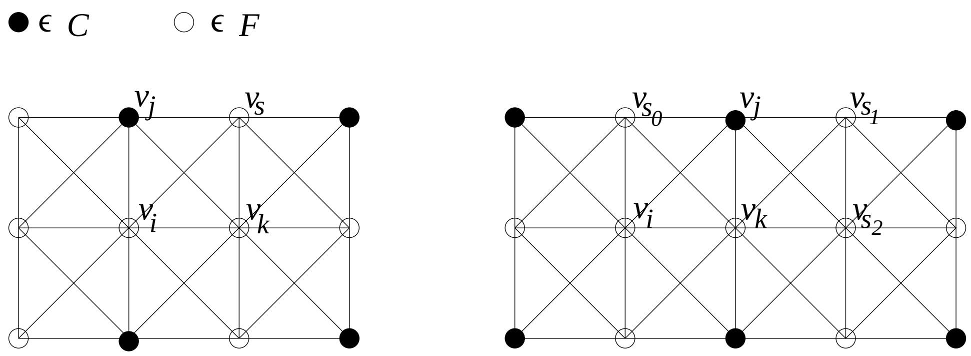

Figure 4.3: Example for the construction of interpolation weights.

EXAMPLE 4.4.1 *Let us consider a system matrix A with the constant 9-point stencil*

$$A = \begin{bmatrix} -\frac{1}{4} & -\frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{2} & 3 & -\frac{1}{2} \\ -\frac{1}{4} & -\frac{1}{2} & -\frac{1}{4} \end{bmatrix}. \tag{4.4.6}$$

Figure 4.3 shows two parts of the corresponding graph. For the situation on the left hand side, $F_{k,j} = \{v_i, v_s\}$ and the condition for ζ in Remark 4.4.2 reads

$$\zeta\,(1/4 + 1/4) \geq 1/2 + 1/2 \quad \Rightarrow \quad \zeta \geq 2.$$

The interpolation coefficient $w_{i,j}$ *is given by*

$$w_{i,j} = \frac{1}{3}\,[1/2 + (1/2 \cdot 1/2 + 1/2 \cdot 1/2 + 1/4 \cdot 1 + 1/4 \cdot 1)] = 1/2.$$

The situation on the right hand side yields $F_{k,j} = \{v_i, v_{s_0}, v_{s_1}, v_{s_2}\}$,

$$\zeta\,(1/2 + 1/2) \geq 1/2 + 1/2 + 1/4 + 1/4 \quad \Rightarrow \quad \zeta \geq 1.5,$$

and

$$w_{i,j} = \frac{1}{3}\,[1/4 + (1/2 \cdot 1/2 + 1/2 \cdot 1/2)] = 1/4.$$

The interpolation weights defined by (4.4.4) and (4.4.5) can be deduced by observing that

$$a_{i,i}\, e_i \approx - \sum_j a_{i,j}\, e_j.$$

for a smooth vector e. Replacing all e_j with $j \neq C_i$ by

$$e_j = \frac{\sum\limits_{v_k \in C_i} |a_{j,k}|\, e_k}{\sum\limits_{v_m \in C_i} |a_{j,m}|}$$

yields those weights. The weights which are used in the Ruge-Stüben AMG are slightly different. They are defined in Section 4.6. Similar techniques can be found e.g. in [Heppner 1999; Reusken 1994; Wagner, Kinzelbach, and Wittum 1997] (see Section 4.7).

4.5 Multigrid Convergence

In order to recursively set up a multigrid algorithm, the essential properties of the system matrix must carry over to the coarser grids.

REMARK 4.5.1 *Let A be a symmetric, weakly diagonally dominant M-matrix and let the interpolation weights (4.4.2) satisfy (4.4.3) with $\beta \leq 2$. Then, $A_H = P^T A\, P$ is a symmetric, weakly diagonally dominant M-matrix as well.*

PROOF. Because of the Galerkin formulation, A_H is spd. The entries $a^H_{i,j}$ of A_H are computed according to

$$\begin{aligned}
a^H_{i,j} &= \sum_{m,k} w_{m,i}\, a_{m,k}\, w_{k,j} \\
&= a_{i,j} + \sum_{v_m \in F} (w_{m,i}\, a_{m,j} + w_{m,j}\, a_{i,m}) + \sum_{v_m, v_k \in F} w_{m,i}\, a_{m,k}\, w_{k,j} \\
&= a_{i,j} + \sum_{v_m \in F} (w_{m,i} \left(a_{m,j} + \frac{1}{2} \sum_{v_k \in F} w_{k,j}\, a_{m,k} \right) \\
&\quad + \sum_{v_m \in F} (w_{m,j} \left(a_{i,m} + \frac{1}{2} \sum_{v_k \in F} w_{k,i}\, a_{k,m} \right),
\end{aligned}$$

where $w_{i,j} = \delta_{i,j}$ for $v_i, v_j \in C$ has been used. Due to the assumption $a_{m,m}\, w_{m,j} \leq 2\,|a_{m,j}|$ ($v_m \in F$, $v_j \in C$)

$$a_{m,j} + \frac{1}{2} \sum_{v_k \in F} w_{k,j}\, a_{m,k} \leq a_{m,j} + \frac{1}{2} w_{m,j}\, a_{m,m} \leq 0 \tag{4.5.1}$$

holds, which implies, using the symmetry of A, $a^H_{i,j} \leq 0$, $i \neq j$. Adding up $\sum_j a^H_{i,j}$ leads to

$$\begin{aligned}
\sum_j a^H_{i,j} &= \sum_{v_j \in C} a_{i,j} + \sum_{v_m \in F} (w_{m,i} \left(\sum_{v_j \in C} a_{m,j} + \frac{1}{2} \sum_{v_k \in F} s_j\, a_{m,k} \right) \\
&\quad + \sum_{v_m \in F} s_m \left(a_{i,m} + \frac{1}{2} \sum_{v_k \in F} w_{k,i}\, a_{k,m} \right) \\
&= \sum_j a_{i,j} - \sum_{v_j \in F} a_{i,j} + \sum_{v_m \in F} (w_{m,i} \left(\sum_j a_{m,j} + \frac{1}{2} \sum_{v_k \in F} s_j\, a_{m,k} \right) \\
&\quad - \sum_{v_m \in F} w_{m,i} \sum_{v_k \in F} a_{m,k} + \sum_{v_m \in F} s_m \left(a_{i,m} + \frac{1}{2} \sum_{v_k \in F} w_{k,i}\, a_{k,m} \right) \\
&= \sum_j a_{i,j} + \sum_{v_m \in F} (w_{m,i} \left(\sum_j a_{m,j} + \frac{1}{2} \sum_{v_k \in F} (s_j - 1)\, a_{m,k} \right) \\
&\quad + \sum_{v_m \in F} (s_m - 1) \left(a_{i,m} + \frac{1}{2} \sum_{v_k \in F} w_{k,i}\, a_{k,m} \right).
\end{aligned}$$

(4.4.3) yields

$$\sum_j a_{m,j} + \frac{1}{2} \sum_{v_k \in F} (s_k - 1)\, a_{m,k} \geq \sum_j a_{m,j} + \frac{1}{2} (s_m - 1)\, a_{m,m} \geq 0.$$

This implies together with (4.5.1) , $s_m \leq 1$, and $\sum_j a_{i,j} \geq 0$ the diagonal dominance of A_H. □

The construction of the interpolation operators in Section 4.4 is mainly based on the condition (4.3.1). Unfortunately, condition (4.3.1) is not strong enough for a successful V-cycle convergence theory. A more suitable condition for a V-cycle convergence theory would be

$$\min_{e_H} \|e - P\, e_H\|^2_{\mathcal{H}^1} \leq \beta\, \|e\|^2_{\mathcal{H}^2} \quad \text{or} \quad \min_{e_H} \|e - P\, e_H\|^2_{\mathcal{H}^0} \leq \beta\, \|e\|^2_{\mathcal{H}^2}. \tag{4.5.2}$$

For smooth errors e, (4.5.2) is much stronger than (4.3.1) because $\|e\|_{\mathcal{H}^2} \ll \|e\|_{\mathcal{H}^1}$. For instance, in the case of a second order scalar partial differential equation and smooth error frequencies on a regular grid of the mesh size h typically $\|e\|_{\mathcal{H}^2} = O(h)\, \|e\|_{\mathcal{H}^1}$.

However, the interpolation operators developed in Section 4.4 turned out to be good enough in practice if certain objectives (Criterion 4.6.1 and 4.6.2) are added to the interpolation requirements.

4.6 The Coarsening Strategy

An important idea of the coarsening strategy is the strong connection.

DEFINITION 4.6.1 *A node $v_i \in V$ is strongly connected to a node $v_j \in V$ with respect to A if*

$$-a_{i,j} \geq \theta \max_{m \neq i}(-a_{i,m}).$$

N_i^S denotes the set of all strongly connected neighbors of v_i

$$N_i^S = \{v_j \in N_i \mid -a_{i,j} \geq \theta \max_{m \neq i}(-a_{i,m})\}.$$

The set of nodes which are strongly connected to v_i is denoted by

$$(N_i^S)^T = \{v_j \in V \mid v_i \in N_j^S\}.$$

The interpolatory nodes C_i are the strong C-node neighbors

$$C_i = N_i^S \cap C.$$

The noninterpolatory nodes D_i are split into strong D_i^S and weak D_i^W noninterpolatory nodes

$$D_i = N_i \setminus C_i, \quad D_i^S = D_i \cap N_i^S, \quad D_i^W = D_i \setminus D_i^S.$$

Once again looking at

$$a_{i,i}\, e_i \approx -\sum_j a_{i,j}\, e_j$$

for a smooth vector e, we can expect that the interpolation of e_j, $v_j \in D_i$

$$e_j = \frac{\sum\limits_{v_k \in C_i} |a_{j,k}| e_k}{\sum\limits_{v_m \in C_i} |a_{j,m}|}$$

is better for nodes v_j which are strongly connected to the nodes in C_i. If we add nodes to C_i, very likely the quality of the interpolation increases. But on the other hand, the stencil sizes of the prolongation and hence the stencil sizes of the coarse grid matrices and the numerical cost grows.

Therefore, the following criteria are taken as guidelines in order to get good interpolations and a reasonable numerical complexity.

CRITERION 4.6.1 *For each node $v_i \in F$, each node $v_j \in N_i^S$ should either be in C, or should be strongly connected to at least one node in C_i.*

CRITERION 4.6.2 *C should be a maximal subset of all nodes with the property that no two C-nodes are strongly connected to each other.*

Criterion 4.6.1 shall ensure that the interpolation is good enough while Criterion 4.6.2 enforces the coarsening algorithm to generate coarse grids with significantly less nodes than the fine grid. Since it is not always possible to satisfy strictly both criteria, Criterion 4.6.2 is only taken as guideline while Criterion 4.6.1 must strictly be satisfied.

The coarsening algorithm is divided into two steps. In the first step, a relatively quick C-node choice is performed. C-nodes are distributed uniformly over the grid attempting to enforce Criterion 4.6.2. In the second part, the tentative F-nodes are tested to satisfy Criterion 4.6.1 and the interpolation weights are computed. Tentative F-nodes not satisfying Criterion 4.6.1 are labeled as C-nodes.

ALGORITHM 4.6.1 *Let a matrix A with corresponding graph $G_A(V,E)$ (see Section 1.2) be given. Then,* `RS_Coarsen1(`$G_A(V,E)$`)` *performs the first step of the Ruge-Stüben coarsening algorithm.*

```
RS_Coarsen1(G_A(V,E))
{
    C = {}; F = {}; U = V;
    for(i = 1; i ≤ |V|; i = i + 1) z_i = |(N_i^S)^T|;
    while(U ≠ {})
    {
        get v_i ∈ U with z_i maximal;
        C = C ∪ {v_i}; U = U \ {v_i};
        for(v_j ∈ (N_i^S)^T ∩ U)
        {
            F = F ∪ {v_j}; U = U \ {v_j};
            for(v_k ∈ N_j^S ∩ U) z_k = z_k + 1;
        }
        for(v_j ∈ N_i^S ∩ U) z_j = z_j − 1;
    }
}
```

z_i measures the value of the node v_i as C-node given the current status of C and F. Initially, nodes with many other nodes strongly connected to them become C-nodes. Later, the tendency is to pick as C-nodes those on which many F-node depend. This tends to produce grids with only very few strong C-C connections (Criterion 4.6.2). In addition, Algorithm 4.6.1 guarantees that each F-node produced has at least one strong connection to a C-node.

ALGORITHM 4.6.2 *Let a tentative splitting in C- and F-nodes generated by Algorithm 4.6.1 be given. Then,* `RS_Coarsen2(`$G_A(V,E)$`,`C`,`F`)` *computes the interpolation weights and determines the final F/C-labeling.*

```
RS_Coarsen2(G_A(V,E),C,F)
{
    T = {};
    while(F \ T ≠ {})
```

```
    {
        pick v_i ∈ F \ T;
        T = T ∪ {v_i}; done = 0;
        C_i = N_i^S ∩ C; D_i^S = N_i^S \ C_i; D_i^W = N_i \ N_i^S; C̄_i = {};
        while(!done)
        {
            d_i = a_{i,i} + Σ_{v_k ∈ D_i^W} a_{i,k}; d_j = a_{i,j} ∀v_j ∈ C_i;
            for(v_k ∈ D_i^S)
            {
                if(N_k^S ∩ C_i ≠ {}) d_j = d_j + a_{i,k} a_{k,j} / Σ_{v_m ∈ C_i} a_{k,m} ∀v_j ∈ C_i;
                else
                {
                    if(C̄_i ≠ {}) { C = C ∪ {v_i}; F = F \ {v_i}; break;}
                    else
                    {
                        C̄_i = {v_k}; C_i = C_i ∪ {v_k}; D_i^S = D_i^S \ {v_k};
                        done = 0; break;
                    }
                }
            }
        }
        if(v_i ∈ F) {C = C ∪ C̄_i; F = F \ C̄_i; w_{i,j} = −d_j/d_i ∀v_j ∈ C_i;}
    }
}
```

The main idea of Algorithm 4.6.2 is to sequentially test each tentative F-node v_i if Criterion 4.6.1 is satisfied. If a node $v_k \in D_i^S$ is found which is not strongly connected to a C-node in C_i, this node k is tentatively put into the sets C_i and $\bar{C}_i$, and the testing of Criterion 4.6.1 for v_i is restarted. If Criterion 4.6.1 is now fulfilled k is labeled as C-node and is permanently added to C_i. If Criterion 4.6.1 is still not valid, the node v_i is marked as C-node and the next tentative F-node is tested. The algorithm terminates if all tentative F-nodes are in the set of tested nodes T.

The computation of the interpolation coefficients $w_{i,j}$ is summarized by the formula

$$\begin{aligned} w_{i,j} &= -\frac{1}{\tilde{a}_{i,i}} \left(a_{i,j} + \sum_{v_k \in D_i^S} \frac{a_{i,k}\, a_{k,j}}{\sum\limits_{v_m \in C_i} a_{k,m}} \right), \qquad (4.6.1) \\ \tilde{a}_{i,i} &= a_{i,i} + \sum_{v_k \in D_i^W} a_{i,k}. \end{aligned}$$

EXAMPLE 4.6.1 *If $\theta < 1/2$, then $D_i^W = \{\}$ for A in (4.4.6). Hence, the coefficients $w_{i,j}$ in Example 4.4.1 satisfy (4.6.1). The F/C-labeling shown in Figure 4.3 is a typical result of Algorithm 4.6.1 and Algorithm 4.6.2 for A in (4.4.6).*

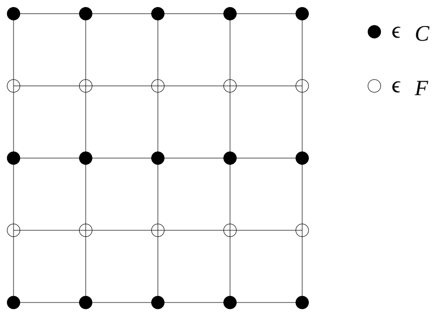

Figure 4.4: Possible F/C-labeling for an anisotropic problem.

EXAMPLE 4.6.2 *Algorithm 4.6.1 and Algorithm 4.6.2 may generate an F/C-labeling as shown in Figure 4.4 for the anisotropic problem in (4.1.2). In this case, the interpolation weights are given by*

$$\tilde{a}_{i,i} = 2 + 2\,\epsilon - 2\,\epsilon, \quad w_{i,j} = -\frac{1}{2}(-1+0) = \frac{1}{2}.$$

4.7 Similar Methods

Reusken proposed in [Reusken 1994] a similar method. The system matrix A is partitioned into four blocks according to the F/C-labeling of the unknowns. For the labeling, a given finite element grid hierarchy is used in [Reusken 1994]. After that, a modified system matrix $\tilde{A}$ is constructed by a lumping procedure.

$$A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \to \begin{pmatrix} \tilde{A}_{FF} & \tilde{A}_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} = \tilde{A}.$$

The blocks $\tilde{A}_{FF}$ and $\tilde{A}_{FC}$ are given by

$$\tilde{A}_{FF} = \operatorname{diag}(A_{FF}), \tag{4.7.1}$$

$$\tilde{a}_{i,j} = a_{i,j} + \sum_{v_k \in D_i} \frac{a_{i,k}}{|C_i \cap N_k|}. \tag{4.7.2}$$

For the two-grid method, $\tilde{A}$ is inverted exactly by block-elimination

$$\begin{aligned} \tilde{A}^{-1} &= \begin{pmatrix} I_{n_F} & -\tilde{A}_{FF}^{-1}\,\tilde{A}_{FC} \\ 0 & I_{n_C} \end{pmatrix} \begin{pmatrix} \tilde{A}_{FF}^{-1} & 0 \\ 0 & A_S^{-1} \end{pmatrix} \begin{pmatrix} I_{n_F} & 0 \\ -A_{CF}\,\tilde{A}_{FF}^{-1} & I_{n_C} \end{pmatrix}, \\ A_S &= A_{CC} - A_{CF}\,\tilde{A}_{FF}^{-1}\,\tilde{A}_{FC}. \end{aligned}$$

The definitions

$$P = \begin{pmatrix} -\tilde{A}_{FF}^{-1}\,\tilde{A}_{FC} \\ I_{n_C} \end{pmatrix}, \quad R = \begin{pmatrix} -A_{CF}\,\tilde{A}_{FF}^{-1} & I_{n_C} \end{pmatrix}, \tag{4.7.3}$$

$$A_H = R\,\tilde{A}\,P = A_{CC} - A_{CF}\,\tilde{A}_{FF}^{-1}\,\tilde{A}_{FC}, \tag{4.7.4}$$

of the prolongation P, the restriction R, and the coarse grid matrix A_H yield

$$\tilde{A}^{-1} = \begin{pmatrix} \tilde{A}_{FF}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + P\, A_H^{-1}\, R.$$

In the same way, the matrices A_l on the coarse levels are recursively constructed.

ALGORITHM 4.7.1 *Let a system matrix $A = A_0$ and an F/C-labeling strategy be given and $\tilde{A}$ be defined by (4.7.1) and (4.7.2). Then,* `Reusken_SCGM`*(A_0) computes the coarse grid matrices A_l.*

```
Reusken_SCGM(A_0)
{
    for(l = 0;  l < l_max;  l = l + 1)
    {
        A_l → Ã_l;
        P_{l,l+1} = ( -Ã_{l,FF}^{-1} Ã_{l,FC} ; I_{n_{l,C}} ),   R_{l+1,l} = ( -A_{l,CF} Ã_{l,FF}^{-1}   I_{n_{l,C}} );
        A_{l+1} = R_{l+1,l} Ã_l P_{l,l+1};
    }
}
```

Note that for symmetric A, the coarse grid matrices are not necessarily symmetric. The corresponding multigrid method can be written as algebraic multigrid cycle (Algorithm 4.0.1) with the additional step

$$u_{l,F} = u_{l,F} + \tilde{A}_{l,FF}^{-1}\, d_{l,F}.$$

ALGORITHM 4.7.2 *Let transfer operators $P_{l,l+1}, R_{l+1,l}$ and coarse grid matrices A_l be constructed by Algorithm 4.7.1. Then, Reusken proposes in [Reusken 1994] the following semi-algebraic multigrid algorithm.*

```
Reusken_SAMG(u_l, f_l, l)
{
    if(l = l_max)  u_l = A_l^{-1} f_l;
    else
    {
        u_l = S_l^{ν1}(u_l, f_l);
        d_l = f_l - A_l u_l;
        u_{l,F} = u_{l,F} + Ã_{l,FF}^{-1} d_{l,F};
        d_{l+1} = R_{l+1,l} d_l;
        v_{l+1} = 0;
        for(j = 0;  j < γ;  j = j + 1) Reusken_SAMG(v_{l+1}, d_{l+1}, l + 1);
        u_l = u_l + P_{l,l+1} v_{l+1};
        u_l = S_l^{ν2}(u_l, f_l);
    }
}
```

A modified version of this technique has been introduced in [Wagner, Kinzelbach, and Wittum 1997]. Looking at

$$\sum_j a_{i,j}\, x_j = f_i, \tag{4.7.5}$$

the unknowns $x_k \in D_i$ are approximated by

$$x_k \to \sum_{v_j \in C_i} \frac{a_{k,j}\, x_j}{\sum_{v_m \in C_i} a_{k,m}} + \frac{f_k}{a_{k,k}}, \quad v_k \in D_i. \tag{4.7.6}$$

Substituting $x_j \in D_i$ in (4.7.5) with (4.7.6) leads to

$$x_i \to \frac{1}{a_{i,i}} \left(\sum_{v_j \in C_i} a_{i,j} x_j + \sum_{v_k \in D_i} \frac{\sum_{v_j \in C_i} a_{i,k}\, a_{k,j} x_j}{\sum_{v_m \in C_i} a_{k,m}} + \sum_{v_k \in D_i} \frac{a_{i,k}\, f_k}{a_{k,k}} + f_{i,i} \right).$$

In matrix form, this can be written as

$$A\, x = f \quad \to \quad \tilde{A}\, x = F\, f,$$

where

$$\tilde{A} = \begin{pmatrix} \tilde{A}_{FF} & \tilde{A}_{FC} \\ A_{CF} & A_{CC} \end{pmatrix}, \tag{4.7.7}$$

$$\tilde{A}_{FF} = \operatorname{diag}(A_{FF}), \tag{4.7.8}$$

$$\tilde{a}_{i,j} = a_{i,j} + \sum_{v_k \in D_i} \frac{a_{i,k}\, a_{k,j}}{\sum_{v_m \in C_i} a_{j,m}}, \tag{4.7.9}$$

$$F = \begin{pmatrix} 2\, I_{n_F} - A_{FF} \tilde{A}_{FF}^{-1} & \\ & I_{n_C} \end{pmatrix}. \tag{4.7.10}$$

Then, the modified linear system $\tilde{A}\, x = F\, f$ is solved by simple block-elimination as described above.

The prolongatiosn $P_{l,l+1}$, the restrictions $R_{l+1,l}$ and the coarse grid matrices A_{l+1} are constructed as in Algorithm 4.7.1 but with the modified definition of $\tilde{A}_l$.

ALGORITHM 4.7.3 *Let a system matrix $A = A_0$ and an F/C-labeling strategy be given and $\tilde{A}$ be defined by (4.7.7)–(4.7.9). Then,* `SchurMG_CGM`*(A_0) computes the transfer operators $P_{l,l+1}$, $R_{l+1,l}$ and the coarse grid matrices A_l.*

`SchurMG_CGM`(A_0)
{
  `for`($l = 0$; $l < l_{\max}$; $l = l + 1$)
  {
    $A_l \to \tilde{A}_l$;
    $P_{l,l+1} = \begin{pmatrix} -\tilde{A}_{l,FF}^{-1}\, \tilde{A}_{l,FC} \\ I_{n_{l,C}} \end{pmatrix}, \quad R_{l+1,l} = \begin{pmatrix} -A_{l,CF}\, \tilde{A}_{l,FF}^{-1} & I_{n_{l,C}} \end{pmatrix}$;

```
        A_{l+1} = R_{l+1,l} Ã_l P_{l,l+1};
    }
}
```

$$A_{l+1} = R_{l+1,l}\,\tilde{A}_l\,P_{l,l+1}\,;$$

This leads to a multigrid method which can be written as standard algebraic multigrid algorithm (Algorithm 4.0.1) with the additional steps

$$\begin{aligned} d_l &= F_l\, d_l, \\ u_{l,F} &= u_{l,F} + \tilde{A}_{l,FF}^{-1}\, d_{l,F}. \end{aligned}$$

ALGORITHM 4.7.4 *Let transfer operators $P_{l,l+1}, R_{l+1,l}$ and coarse grid matrices A_l be constructed by Algorithm 4.7.3 and F_l be defined by (4.7.10). Then, the Schur-complement multigrid method introduced in [Wagner, Kinzelbach, and Wittum 1997] reads*

```
SchurMG(u_l, f_l, l)
{
    if(l = l_max)  u_l = A_l^{-1} f_l;
    else
    {
        u_l = S_l^{ν1}(u_l, f_l);
        d_l = F_l (f_l − A_l u_l);
        u_{l,F} = u_{l,F} + Ã_{l,FF}^{-1} d_{l,F};
        d_{l+1} = R_{l+1,l} d_l;
        v_{l+1} = 0;
        for(j = 0; j < γ; j = j + 1) SchurMG(v_{l+1}, d_{l+1}, l + 1);
        u_l = u_l + P_{l,l+1} v_{l+1};
        u_l = S_l^{ν2}(u_l, f_l);
    }
}
```

The methods by Reusken and Wagner discussed in this section have been proposed and tested in [Reusken 1994; Wagner, Kinzelbach, and Wittum 1997] only on a given standard multigrid hierarchy. Heppner combined both methods with several F/C-labeling strategies. For the results, we refer to [Heppner 1999].

4.8 Numerical Experiments

EXPERIMENT 4.8.1 *The anisotropic differential equation*

$$\begin{aligned} -\nabla\cdot(D(x)\nabla u(x)) &= f(x), \quad x\in\Omega=(0,1)\times(0,1) \\ u(x) &= g(x), \quad x\in\partial\Omega, \\ D(x) &= \begin{pmatrix} \epsilon & 0 \\ 0 & 1 \end{pmatrix}, \end{aligned}$$

is discretized on the equidistant grid ($h = 1/64$)

$$\Omega_h = \{\, x_{(i,j)} \in \Omega \,|\, x_{(i,j)} = (i\,h, j\,h),\ i,j = 1,\dots,n,\ h = \frac{1}{n+1}\,\} \tag{4.8.1}$$

using the 5-point stencil (4.1.2).

EXPERIMENT 4.8.2 *The convection-diffusion problem*

$$\begin{aligned} -\nabla \cdot (\epsilon \nabla u(x) + v(x)\,u(x)) &= f(x), \quad x \in \Omega = (0,1) \times (0,1) \\ u(x) &= g(x), \quad x \in \partial\Omega, \\ v_1(x) &= -4\,x_1(x_1 - 1)(1 - 2\,x_2), \\ v_2(x) &= 4\,x_2(x_2 - 1)(1 - 2\,x_1), \end{aligned}$$

is discretized on the equidistant grid (4.8.1) ($h = 1/64$) with a 5-point finite difference upwind scheme. The rotating convection $v(x)$ is illustrated in Figure 4.5.

EXPERIMENT 4.8.3 *The interface problem*

$$\begin{aligned} -\nabla \cdot (\mathcal{D}(x)\nabla u(x)) &= f(x), \quad x \in \Omega = (0,1) \times (0,1) \\ u(x) &= g(x), \quad x \in \partial\Omega, \\ \mathcal{D}(x) &= \begin{cases} 1 & : \ 0 \le x \le 0.5 \ \wedge \ 0 \le y \le 0.5, \\ 10 & : \ 0 \le x \le 0.5 \ \wedge \ 0.5 \le y \le 1.0, \\ 100 & : \ 0.5 \le x \le 1.0 \ \wedge \ 0 \le y \le 0.5, \\ 1000 & : \ 0.5 \le x \le 1.0 \ \wedge \ 0.5 \le y \le 1.0, \end{cases} \end{aligned}$$

is discretized on the equidistant grid (4.8.1) ($h = 1/64$) using a 5-point stencil.

Ruge and Stüben [Ruge and Stüben 1987] report asymptotic convergence rates for their algorithm (Algorithm 4.0.1 + Algorithm 4.6.1 + Algorithm 4.6.2) with V(1,1)-cycle, Gauß-Seidel relaxation and $\theta = 0.25$. The results for Experiment 4.8.1 and Experiment 4.8.2 can be found in Table 4.1 and Table 4.2 respectively. The asymptotic convergence rate for Experiment 4.8.3 is 0.082. Further results are presented in Section 6.6.1.

ϵ	0.001	0.01	0.1	1	10	100	1000
convergence rate	0.082	0.094	0.063	0.054	0.079	0.095	0.083

Table 4.1: Asymptotic convergence rates for Experiment 4.8.1.

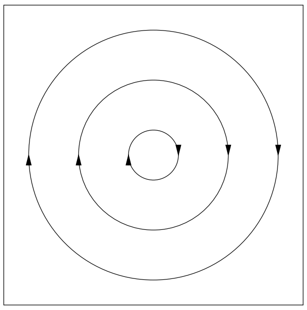

Figure 4.5: Rotating convection.

ϵ	0.1	0.001	0.00001
convergence rate	0.056	0.160	0.173

Table 4.2: Asymptotic convergence rate for Experiment 4.8.2.

5 Aggregation Methods

An iteration step of the aggregation algebraic multigrid methods is described by Algorithm 4.0.1. The main difference to the interpolation methods is the construction of the transfer operators and the coarse grids. In interpolation methods, typically, each coarse grid degree of freedom has an directly associated degree of freedom on the fine grid. Since aggregation methods cluster the fine grid unknowns to aggregates representing the unknowns on the coarse grid, aggregation methods do not allow such a simple identification of degrees of freedom on the coarse and the fine grid.

Cell-centered finite volume discretizations lead to aggregation methods in a very natural way. The same idea can be applied to standard vertex-centered discretizations.

5.1 Cell-Centered Multigrid Methods

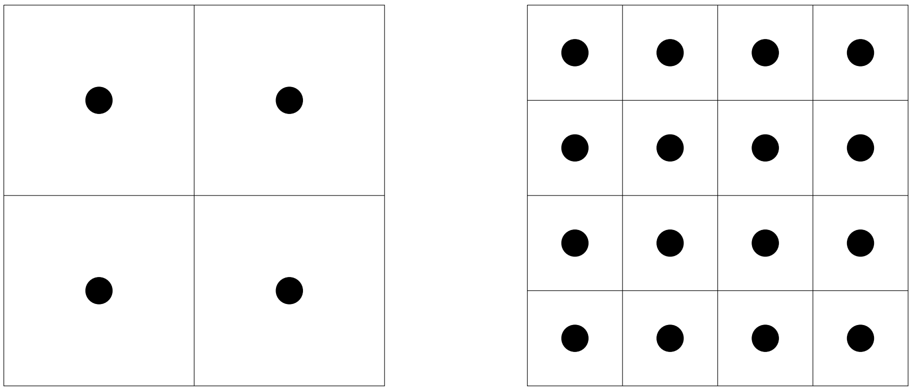

Figure 5.1: Cell-centered discretization.

Cell-centered finite volume discretizations (see e.g. [Wesseling 1992]) generate linear systems where the unknowns are located in the center of the elements of the triangulation (see Figure 5.1). Hence, depending on the partial differential equation, the unknowns describe for example the pressure, the electrostatic potential or the concentration in the corresponding grid cell.

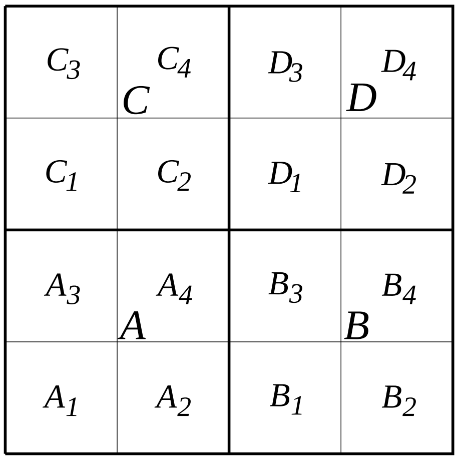

Figure 5.2: Cell-centered multigrid.

For the cell center arrangement in Figure 5.2, first order prolongations and restrictions can be easily derived by

$$\begin{aligned} (P^{(0)}\, u_H)_{A,i} &= u_{H,A}, \quad i = 1, 2, 3, 4, \\ (R^{(0)}\, d_h)_A &= \frac{1}{4} \sum_{i=1}^{4} d_{h,A_i}. \end{aligned}$$

Bilinear interpolation gives

$$(P^{(1)}\, u_H)_{A,4} = \frac{1}{16} (9\, u_{H,A} + 3\, u_{H,B} + 3\, u_{H,C} + u_{H,D}). \tag{5.1.1}$$

The other values of $P^{(1)}\, u_H$ follow by symmetry in the same fashion. The corresponding restriction is obtained by $R^{(1)} = 1/4\,(P^{(1)})^T$. A generalization to one and three spatial dimensions is straightforward [Wesseling 1992].

It has been shown in [Khalil and Wesseling 1992; Wesseling 1988] that cell-centered multigrid methods can handle interface problems (see Section 3.1, Experiment 3.7.1, and Experiment 3.7.2) with simple transfer operators. A suitable choice is the zeroth-order interpolation

$$P^{(0)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

in combination of the bilinear restriction $R^{(1)}$ and the Galerkin coarse grid matrix $R^{(1)}\, A\, P^{(0)}$. This gives $m_P = 1$, $m_R = 2$ such that (2.5.1) is satisfied. Since C^1 continuity cannot be assumed at the interfaces, the zeroth-order interpolation is a reasonable choice.

5.2 Smoothed Basis Functions

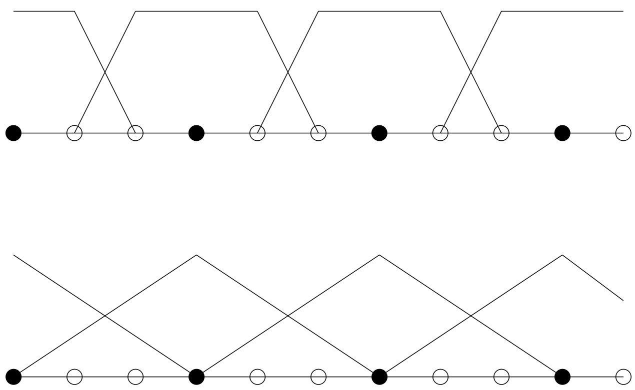

Figure 5.3: Smoothed basis functions.

The standard one-dimensional model problem

$$A\,u = f, \quad A = [-1 \;\; 2 \;\; -1], \quad A \in \mathbb{R}^{n_h \times n_h} \tag{5.2.1}$$

is considered. The application of the smoother S

$$S = I - \frac{2}{3} D^{-1} A, \quad D = \mathrm{diag}(A),$$

to a vector $\phi = (0, \ldots, 0, 1, 1, 1, 0, \ldots, 0)^T$ yields the smoothed vector

$$S\,\phi = \psi = (0, \ldots, 0, \frac{1}{3}, \frac{2}{3}, 1, \frac{2}{3}, \frac{1}{3}, 0, \ldots, 0)^T.$$

Vectors of the form $\psi \in \mathbb{R}^{n_h}$ and $\phi \in \mathbb{R}^{n_h}$ (see Figure 5.3) can be used as coarse grid basis vectors for a multigrid algorithm. In that sense, the solution u is approximated by a linear combination of the coarse grid basis vectors

$$\begin{aligned} u &\approx \sum_{i \le n_H} \phi_i \cdot (u_H)_i \quad \Leftrightarrow \quad u \approx P_\phi\, u_H, \quad P_\phi \in \mathbb{R}^{n_h \times n_H}, \\ u &\approx \sum_{i \le n_H} \psi_i \cdot (u_H)_i \quad \Leftrightarrow \quad u \approx P_\psi\, u_H, \quad P_\psi \in \mathbb{R}^{n_h \times n_H}, \end{aligned}$$

where

$$P_\phi = (\phi_1, \ldots, \phi_{n_H}), \quad P_\psi = (\psi_1, \ldots, \psi_{n_H}).$$

Substituting u in (5.2.1) and multiplying both sides with $R_\phi \in \mathbb{R}^{n_H \times n_h}$ or $R_\psi \in \mathbb{R}^{n_H \times n_h}$ leads to the coarse grid systems

$$R_\phi\, A\, P_\phi\, u_H = R_\phi\, f, \quad R_\psi\, A\, P_\psi\, u_H = R_\psi\, f.$$

Note that, the number of degrees of freedom on the coarse grid n_H generated by R_ϕ, P_ϕ or R_ψ, P_ψ is approximately 1/3 of the number of fine grid unknowns

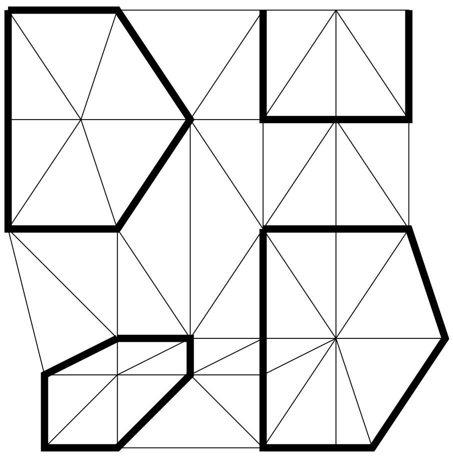

Figure 5.4: Typical 2D aggregates.

($n_H \approx 1/3\, n_h$). The standard (1D) multigrid prolongation induced by the basis functions $(0,\dots,0,1/2,1,1/2,0,\dots,0)^T$ leads to a coarsening factor of about 1/2 ($n_H \approx 1/2\, n_h$). Nevertheless, the prolongation P_ψ is still a linear interpolation scheme.

In two (or three) spatial dimensions, the basis functions ϕ are constructed by clustering some unknowns to so-called aggregates (see Figure 5.4). Then, the basis function ϕ_i corresponding to the aggregate $\mathcal{A}_i$ is defined by

$$(\phi_i)_j = \begin{cases} 1 & : \quad j \in \mathcal{A}_i, \\ 0 & : \quad j \notin \mathcal{A}_i. \end{cases}$$

In the remainder of this chapter, the algebraic multigrid method by Vanek, Brezina, and Mandel which is based on smooth aggregation is discussed [Vanek, Mandel, and Brezina 1994; Vanek, Mandel, and Brezina 1996; Vanek, Brezina, and Mandel 1998]. The aggregation method by Braess [Braess 1995] is briefly summarized.

5.3 The Construction of the Aggregates

Following [Vanek, Mandel, and Brezina 1996], this section describes the construction of a system of aggregates $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$ based on the graph $G_{A_l}(V_l, E_l)$ (see Section 1.2) of the matrix A_l.

For a given parameter θ, the strongly coupled neighborhood of the node v_i is defined as

$$N_i^l(\theta) = \{v_j \in V_l \,|\, |a_{i,j}| \geq \theta \sqrt{a_{i,i}\, a_{j,j}}\,\}.$$

ALGORITHM 5.3.1 *Let an* $n_l \times n_l$ *matrix* A_l *with the corresponding graph* $G_{A_l}(V_l, E_l)$ *and* $\theta \in (0,1]$ *be given. Then* `Aggregate(`$G_{A_l}(V_l, E_l)$`)` *generates a disjoint covering* $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$ *of the set* $V_l = \{v_1, \dots, v_{n_l}\}$.

```
Aggregate(G_{A_l}(V_l, E_l))
{
    /* Initialization */
    U = {v_i ∈ V_l | N_i^l(0) ≠ {v_i} };
    j = 0;
    /* Step 1 */
    for(v_i ∈ U)
    {
        if(N_i^l(θ) ⊂ U) { j = j + 1; A_j^l = N_i^l(θ); U = U \ A_j^l; }
    }
    /* Step 2 */
    for(z ≤ j) Ā_z^l = A_z^l;
    for(v_i ∈ U)
    {
        for(z ≤ j)
        {
            if(N_i^l(θ) ∩ Ā_z^l ≠ {}) {A_z^l = A_z^l ∪ {v_i}; U = U \ {v_i}; break;}
        }
    }
    /* Step 3 */
    for(v_i ∈ U)
    {
        j = j + 1; A_j^l = N_i^l(θ) ∩ U; U = U \ A_j^l;
    }
}
```

In the initialization phase of Algorithm 5.3.1, the set U of not yet clustered nodes is filled. Isolated nodes are not aggregated. After that, in step 1, disjoint strongly coupled neighborhoods are selected as the initial approximation of the covering. Step 2 adds remaining nodes $v_i \in U$ to one of the sets $\mathcal{A}_z^l$ to which the node v_i is strongly connected if any such set exists. Finally, in step 3, the still remaining nodes $v_i \in U$ are clustered into aggregates that consist of subsets of strongly coupled neighborhoods.

5.4 The Construction of the Prolongators

In [Vanek, Brezina, and Mandel 1998], a construction scheme for prolongators for systems of differential equation is introduced. For an easier description, the discussion in the section is limited to scalar equations.

Let a system matrix $A = A_0 \in \mathbb{R}^{n_0 \times n_0}$ be given. The goal of this section is to create a hierarchy of tentative prolongators (prolongation operators) Y_{l+1}^l such that for a given vector $t_0 \in \mathbb{R}^{n_0}$,

$$t_0 \in \text{range}(Y_l^0), \quad Y_l^0 = Y_1^0\, Y_2^1 \cdots Y_l^{l-1}, \quad Y_0^0 = I, \quad l = 0, \ldots, l_{\max}. \tag{5.4.1}$$

A typical example for t_0 is $t_0 = (1, \ldots, 1)^T$ or $t_0 = \text{kernel}(A)$.

Similar so-called test vectors have been introduced in [Wittum 1992; Wagner 1997a; Wagner and Wittum 1997]. For systems of partial differential equations, these test vectors t_l become matrices B^l in [Vanek, Brezina, and Mandel 1998].

To enforce (5.4.1), we simultaneously create the prolongator Y_{l+1}^l and a vector $t_{l+1} \in \mathbb{R}^{n_{l+1}}$ satisfying

$$Y_{l+1}^l \, t_{l+1} = t_l, \tag{5.4.2}$$

where t_l has been constructed during the setup of Y_l^{l-1} or is given for $l = 0$. Note that (5.4.2) induces

$$Y_l^0 \, t_l = t_0. \tag{5.4.3}$$

The tentative prolongator Y_{l+1}^l is constructed from a given disjoint system of aggregates $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$ as discussed in Section 5.3. The column of Y_{l+1}^l associated with the aggregate $\mathcal{A}_i^l$ is formed by restriction of the columns of t_l onto the aggregate $\mathcal{A}_i^l$. Each aggregate gives rise to one degree of freedom on the coarse grid.

The detailed algorithm follows. For ease of presentation, we assume that the fine level unknowns are numbered by consecutive numbers within each aggregate. This assumption can be easily avoided by renumbering.

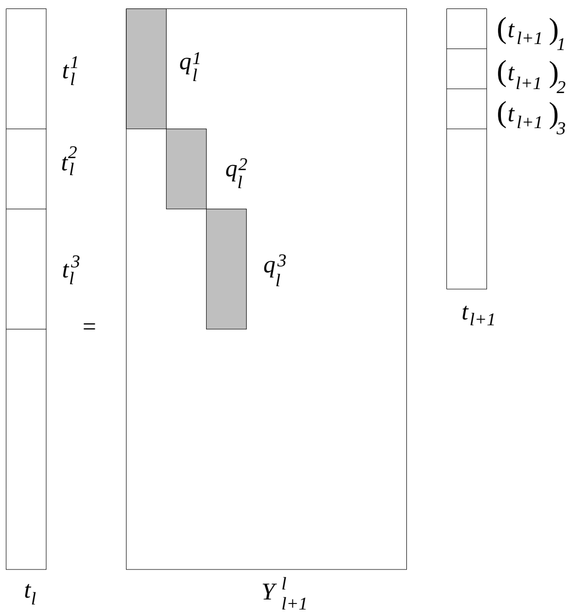

Figure 5.5: The tentative prolongator Y_{l+1}^l.

ALGORITHM 5.4.1 *For a given system of aggregates $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$ and a vector $t_l \in \mathbb{R}^{n_l}$ satisfying (5.4.3), a tentative prolongator Y_{l+1}^l, a vector t_{l+1} satisfying (5.4.2), and degrees of freedom on level $l+1$ are created as follows:*

1. *Partition the vector* $t_l \in \mathbb{R}^{n_l}$ *into* n_{l+1} *blocks* $t_l^i \in \mathbb{R}^{d_i}$, $i = 1, \dots, n_{l+1}$, *each corresponding to the set of degrees of freedom on an aggregate* $\mathcal{A}_i^l$ *(see Fig.5.5).* d_i *denotes the number of degrees of freedom associated with the aggregate* $\mathcal{A}_i^l$.
2. *Write* $t_l^i = q_l^i \, (t_{l+1})_i$, *where* $\|q_l^i\| = 1$.
3. *Set (see Fig.5.5)*
$$Y_{l+1}^l = \begin{pmatrix} q_l^1 & & \\ & \ddots & \\ & & q_l^{n_{l+1}} \end{pmatrix}, \quad t_{l+1} = \begin{pmatrix} (t_{l+1})_1 \\ \vdots \\ (t_{l+1})_{n_{l+1}} \end{pmatrix}.$$
4. *The coarsening gives rise to one degree of freedom on the coarse level (the i-th column of* Y_{l+1}^l*) for each aggregate* $\mathcal{A}_i^l$.

The smoothed composite prolongator $P_{0,l}$ is defined by

$$P_{0,l} = Z_0 \, Y_1^0 \cdots Z_{l-1} \, Y_l^{l-1}, \quad P_{0,0} = I,$$

where Z_l denotes a prolongator smoother. The prolongation $P_{l,l+1}$, the restriction $R_{l+1,l}$ and the coarse grid matrices A_{l+1} are defined by

$$P_{l,l+1} = Z_l \, Y_{l+1}^l, \quad R_{l+1,l} = (P_{l,l+1})^T, \quad A_{l+1} = R_{l+1,l} \, A_l \, P_{l,l+1}, \quad l = 0, \dots, l_{\max} - 1, \tag{5.4.4}$$

For the convergence analysis, the prolongator smoother

$$\begin{aligned} Z_l &= I - \frac{4}{3 \, \bar{\lambda}_l^W} W_l^{-1} \, A_l, \\ W_l &= (Y_l^0)^T \, (Y_l^0), \quad \bar{\lambda}_l^W \geq \rho(W_l^{-1} \, A_l), \end{aligned} \tag{5.4.5}$$

turned out to be useful. Later, we will show for system matrices A produced by a finite element discretization of an elliptic partial equation, that

$$\bar{\lambda}_l^W = 9^{-l} \, \bar{\lambda} \tag{5.4.6}$$

is a possible choice for $\bar{\lambda}_l^W$, where $\bar{\lambda} \geq \rho(A)$ is an upper bound for the spectral radius of A. A more heuristicly motivated prolongator smoother

$$\begin{aligned} Z &= I - \omega \, D^{-1} \, \hat{A}, \\ \hat{a}_{i,j} &= \begin{cases} a_{i,j} & : \; j \in N_i(\epsilon), \\ 0 & : \; \text{otherwise}, \end{cases} \quad i \neq j, \\ \hat{a}_{i,i} &= a_{i,i} - \sum_{j \neq i} (a_{i,j} - \hat{a}_{i,j}), \\ \omega &= \frac{2}{3}, \quad \epsilon = 0.08 \left(\frac{1}{2}\right)^l, \end{aligned} \tag{5.4.7}$$

is proposed in [Vanek, Mandel, and Brezina 1996].

5.5 Convergence Theory

This section summarizes the convergence theory in [Vanek, Brezina, and Mandel 1998]. The convergence analysis is restricted to the symmetric case $A = A^T$. The following additional assumptions are required.

Assumption 5.5.1 *The system matrix A is spd. The prolongation $P_{l,l+1}$, the restriction $R_{l+1,l}$ and the coarse grid matrices A_{l+1} are defined by (5.4.4) with Z_l in (5.4.5) and Y_{l+1}^{l} is constructed by Algorithm 5.4.1. The smoothers*

$$S_l = I - (M_l^S)^{-1}\, A_l, \quad M_l^S \text{ spd},$$

satisfy

$$\lambda_{\min}(S_l) \geq 0, \quad \lambda_{\max}(M_l^S) \leq C_S^2\, \rho(A_l), \tag{5.5.1}$$

with a constant $C_S > 0$ independent of the level.

Definition 5.5.1 *The hierarchy of coarse grid spaces $\mathcal{V}_{l_{\max}} \subset \mathcal{V}_{l_{\max}-1} \subset \cdots \subset \mathcal{V}_0$ is induced by the smoothed composite prolongators $P_{0,l}$*

$$\mathcal{V}_l = \operatorname{range}(P_{0,l}).$$

The spectral radius of $\rho(A_l)$ of the coarse grid matrices is defined by

$$\rho(A_l) = \max_{u\in\mathcal{V}_l} \left(\frac{\|u\|_A}{\|u\|_{\mathcal{V}_l}}\right)^2$$

where

$$\|u\|_{\mathcal{V}_l} = \min\{\|x\| \, : \, u = P_{0,l}\, x,\ x \in \mathbb{R}^{n_l}\}.$$

The approximate inverse corresponding to one algebraic multigrid iteration step (Algorithm 4.0.1) is denoted by M and the iteration matrix by $T = I - M^{-1}\, A$.

The convergence theory is based on an abstract convergence result in [Bramble, Pasciak, Wang, and Xu 1991]. It can be written in our notation as follows.

Theorem 5.5.1 *([Bramble, Pasciak, Wang, and Xu 1991], Theorem 1). Assume there are linear mappings $Q_l : \mathcal{V}_0 \to \mathcal{V}_l$, $Q_0 = I$ and constants $C_1, C_2 > 0$ such that*

$$\begin{aligned} \|Q_l\, u\|_A &\leq C_1\, \|u\|_A \quad \forall\, u \in \mathcal{V}_0, \quad l = 0, \dots l_{\max} - 1, && (5.5.2)\\ \|(Q_l - Q_{l+1})\, u\|_{\mathcal{V}_l} &\leq \frac{C_2}{\sqrt{\rho(A_l)}}\, \|u\|_A \quad \forall\, u \in \mathcal{V}_0, \quad l = 0, \dots l_{\max} - 1. && (5.5.3) \end{aligned}$$

Then, if Assumption 5.5.1 is fulfilled,

$$\|T\, e\|_A \leq \left(1 - \frac{1}{C_0}\right)\, \|e\|_A \quad \forall\, e \in \mathcal{V}_0$$

holds with $C_0 = (1 + C_1 + C_2\, C_S)\, l_{\max}$.

The assumption (5.5.2) addresses the (algebraic) smoothness of the coarse grid basis functions. The smoothing of the tentative prolongators Y_{l+1}^{l} is necessary to satisfy (5.5.2). The condition (5.5.3) requires that constant functions are exactly interpolated. Therefore, the filter condition (5.4.1) must hold for a constant test vector t_0. Note that the connection to the considered multigrid method is made by the coarse grid spaces $\mathcal{V}_l$.

Before the main convergence result is formulated, the composite aggregate and the associated norm must be introduced. The composite aggregate $\tilde{\mathcal{A}}_i^l$ is the aggregate $\mathcal{A}_i^l$ understood as the corresponding set of unknowns on the finest level. Formally, $\tilde{\mathcal{A}}_i^l$ is defined by

$$\tilde{\mathcal{A}}_i^l = \mathcal{A}_i^{l,0}, \quad \text{where} \quad \mathcal{A}_i^{l,l} = \mathcal{A}_i^l, \quad \mathcal{A}_i^{l,j-1} = \bigcup_{z \in \mathcal{A}_i^{l,j}} \mathcal{A}_z^{j-1}.$$

The corresponding discrete l^2-(semi)norm of the vector $x \in \mathbb{R}^{n_0}$ is given by

$$\|x\|_{l^2(\tilde{\mathcal{A}}_i^l)} = \left(\sum_{\text{dof } j \text{ of } \tilde{\mathcal{A}}_i^l} x_j^2 \right)^{1/2}.$$

THEOREM 5.5.2 *Let Assumption 5.5.1 hold with $\bar{\lambda}_l^W$ in (5.4.6) and the tentative prologators Y_{l+1}^l created using the test vector $t_0 \in \mathbb{R}^{n_0}$ and the aggregates $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$, $l = 0, \dots, l_{\max} - 1$. Assume there is a constant $C_A > 0$ such that*

$$\sum_{i=1}^{n_{l+1}} \min_{w \in \mathbb{R}} \|u - t_0\, w\|^2_{l^2(\tilde{\mathcal{A}}_i^l)} \le C_A \frac{9^l}{\bar{\lambda}} \|u\|_A^2 \tag{5.5.4}$$

for every $u \in \mathbb{R}^{n_0}$ and every $l = 0, \dots, l_{\max} - 1$. Then,

$$\|T\,e\|_A \le \left(1 - \frac{1}{C_0}\right) \|e\|_A \quad \forall\, e \in \mathcal{V}_0$$

holds with $C_0 = [2 + C_A\, C_S + \frac{4}{3} C_S + \frac{1}{3} C_A\,(1 + \frac{4}{3} C_S) l_{\max}]^2\, l_{\max}$.

PROOF. For a detailed proof, we refer to [Vanek, Brezina, and Mandel 1998]. A sketch of the proof can be found in the rest of this section. □

The following lemma prepares the proof of Theorem 5.5.2. Readers not interested in the technical details of the proof may skip this part and continue with the verification of the assumptions of Theorem 5.5.2 in the next section.

The idea of the proof is to verify the assumptions of Theorem 5.5.1 (5.5.2) and (5.5.3) form the properties of Z_l and Y_l^0 rather than $P_{0,l}$. It is shown in [Vanek, Brezina, and Mandel 1998], that for the prolongator smoother Z_l in (5.4.1) and $\bar{\lambda}_l^W$ in (5.4.6) the assumptions (5.5.2) and (5.5.3) are equivalent to (5.5.5) and (5.5.6) in the following lemma.

LEMMA 5.5.1 *Let the prolongator smoother be given by (5.4.5) with $\bar{\lambda}_l^W$ in (5.4.6). Assume that C_1 and C_W are such that there are linear mappings*

$$\tilde{Q}_l : \mathbb{R}^{n_0} \to \mathbb{R}^{n_l}, \quad l = 0, \dots, l_{\max}, \quad \tilde{Q}_0 = I,$$

with

$$\begin{aligned} \|Y_l^0 \tilde{Q}_l u - Y_{l+1}^0 \tilde{Q}_{l+1} u\|_2^2 &\le C_1^2 \frac{9^l}{\bar{\lambda}} \|u\|_A^2 \quad \forall u \in \mathbb{R}^{n_0}, \quad l = 0, \dots, l_{\max} - 1, && (5.5.5)\\ \operatorname{cond}(W_l) &\le C_W^2, \quad l = 0, \dots, l_{\max}. && (5.5.6) \end{aligned}$$

Then, if Assumption 5.5.1 is satisfied,

$$\|T e\|_A \le \left(1 - \frac{1}{C_0}\right) \|e\|_A \quad \forall e \in \mathcal{V}_0$$

holds with

$$C_0 = [2 + C_1 C_W C_S + \frac{4}{3} C_W C_S + \frac{1}{3} C_1 (1 + \frac{4}{3} C_W C_S) l_{\max}]^2 \, l_{\max}.$$

PROOF. The proof uses several algebraic properties of the prolongator smoother Z_l and shows that $Q_l = Y_l^0 \tilde{Q}_l$ fulfills (5.5.2) and (5.5.3). The detailed proof can be found in [Vanek, Brezina, and Mandel 1998]. □

The key assumption (5.5.5) of Lemma 5.5.1 is a weak approximation property for disaggregated functions. With the choice $\tilde{Q}_l = W_l^{-1} (Y_l^0)^T$, the mappings $Y_l^0 \tilde{Q}_l$ are orthogonal projections onto range(Y_l^0). Since range$(Y_{l+1}^0) \subset$ range(Y_l^0), v exists with

$$\begin{aligned} (u - Y_l^0 \tilde{Q}_l u, Y_l^0 \tilde{Q}_l u - Y_{l+1}^0 \tilde{Q}_{l+1} u)_2 &= (u - Y_l^0 \tilde{Q}_l u, Y_l^0 \tilde{Q}_l v)_2 \\ &= ((Y_l^0)^T (I - Y_l^0 \tilde{Q}_l) u, \tilde{Q}_l v)_2 = 0, \end{aligned}$$

and we obtain

$$\begin{aligned} \|u - Y_{l+1}^0 \tilde{Q}_{l+1} u\|_2^2 &= \|Y_l^0 \tilde{Q}_l u - Y_{l+1}^0 \tilde{Q}_{l+1} u\|_2^2 + \|u - Y_l^0 \tilde{Q}_l u\|_2^2 \\ &\ge \|Y_l^0 \tilde{Q}_l u - Y_{l+1}^0 \tilde{Q}_{l+1} u\|_2^2. \end{aligned}$$

Hence, from the minimization property of the orthogonal projection follows

$$\|Y_l^0 \tilde{Q}_l u - Y_{l+1}^0 \tilde{Q}_{l+1} u\|_2^2 \le \|u - Y_{l+1}^0 \tilde{Q}_{l+1} u\|_2^2 = \min_{u_{l+1} \in \mathbb{R}^{n_{l+1}}} \|u - Y_{l+1}^0 u_{l+1}\|_2^2.$$

Thus, the weak approximation property in the more usual form

$$\min_{u_{l+1} \in \mathbb{R}^{n_{l+1}}} \|u - Y_{l+1}^0 u_{l+1}\|_2^2 \le \bar{C}_A^2 \|u\|_A^2$$

induces (5.5.5) with $\bar{C}_A^2 = C_1^2 \frac{9^l}{\bar{\lambda}}$.

The tentative prolongators Y_{l+1}^l constructed in Algorithm 5.4.1 are block-diagonal matrices with normalized blocks. Therefore, $(Y_{l+1}^l)^T Y_{l+1}^l = I_{l+1}$ and

$$W_l = (Y_l^0)^T Y_l^0 = (Y_1^0 Y_2^1 \cdots Y_l^{l-1})^T Y_1^0 Y_2^1 \cdots Y_l^{l-1} = I_l,$$

showing $C_W = 1$ in (5.5.6).

Finally,

$$\begin{aligned}
\min_{u_{l+1}\in\mathbb{R}^{n_{l+1}}} \|u - Y^0_{l+1}\, u_{l+1}\|_2^2 &\le \sum_{i=1}^{n_{l+1}} \min_{u_{l+1}\in\mathbb{R}^{n_{l+1}}} \|u - Y^0_{l+1}\, u_{l+1}\|^2_{l^2(\tilde{\mathcal{A}}^l_i)} \\
&\le \sum_{i=1}^{n_{l+1}} \min_{w\in\mathbb{R}} \|u - t_0\, w\|^2_{l^2(\tilde{\mathcal{A}}^l_i)}
\end{aligned}$$

completes the proof of Theorem 5.5.2.

5.6 The Convergence Results Applied to a Model Problem

The goal of this section is to demonstrate the verification of the key assumption (5.5.4) for a simple model problem. For the verification of the smoothing assumption (5.5.1), we refer to [Bramble, Pasciak, Wang, and Xu 1991; Hackbusch 1985; Hackbusch 1993; Hackbusch 1994; Reusken 1991].

We consider the discretization $A\,u = f$ of the second order scalar elliptic problem

$$a(\mathbf{u}, \mathbf{v}) = f(\mathbf{v}), \quad \forall\, \mathbf{v} \in \mathcal{V}_h, \quad \mathbf{u} \in \mathcal{V}_h$$

on a quasi-uniform finite element mesh τ_h on $\Omega \subset \mathbb{R}^d$ associated with a finite element space $\mathcal{V}_h$ of piecewise linear or bilinear basis functions φ_i, $\|\varphi_i\|_{L^\infty} = 1$. $a(\cdot,\cdot)$ is supposed to be H^1-equivalent ($c\,|\mathbf{u}|^2_{H^1(\Omega)} \le a(\mathbf{u},\mathbf{u}) \le C\,|\mathbf{u}|^2_{H^1(\Omega)}$).

On level 0, each degree of freedom is associated with one finite element node not lying on a Dirichlet boundary.

ASSUMPTION 5.6.1 *On every level $l < l_{\max}$, for each aggregate $\mathcal{A}^l_i$ there is a ball $\mathcal{B}^l_i \in \mathbb{R}^d$ such that*

1. *all finite element vertices of the corresponding composite aggregate $\tilde{\mathcal{A}}^l_i$ are located within $\mathcal{B}^l_i$;*
2. $\mathrm{diam}(\mathcal{B}^l_i) \le C_G\, 3^l\, h$, *where h is the characteristic mesh-size of τ_h and $C_G > 0$ is a positive constant independent of the level;*
3. *there is an integer constant N independent of the level such that every point $x \in \Omega$ belongs to at most N balls $\mathcal{B}^l_i$. (Overlaps of the balls are bounded.)*

According to [Vanek, Brezina, and Mandel 1998], Algorithm 5.3.1 generates aggregates satisfying Assumption 5.6.1.

Let $t_0 = (1, \ldots, 1)^T$ and $\Omega' \subset \Omega$ denote the domain consisting of all elements of the mesh τ_h, that are not adjacent to a finite element node with prescribed Dirichlet boundary conditions. Then,

$$\sum_i \varphi_i(x) = 1, \quad x \in \Omega',$$

and, as all active degrees of freedom are located in $\bar{\Omega}'$, the equivalence of discrete and continuous L^2-norms gives

$$h^d \, \|u - t_0 \, b\|^2_{l^2(\tilde{\mathcal{A}}_i^l)} \leq C \, \|\mathbf{u} - b\|^2_{L^2(\mathcal{B}_i^l \cap \Omega')} \leq C \, \|E \, \mathbf{u} - b\|^2_{L^2(\mathcal{B}_i^l)},$$

where C is a generic constant and $E \, : \, H^1(\Omega) \to \{ \mathbf{v} \, | \, |\mathbf{v}|_{H^1(\mathbb{R}^d)} < \infty \}$ is the extension operator satisfying

$$E \, \mathbf{u} = \mathbf{u} \quad x \in \Omega, \quad |E \, \mathbf{u}|_{H^1(\mathbb{R}^d)} \leq C \, |\mathbf{u}|_{H^1(\Omega)}.$$

Due to the scaled Poincare inequality, there is a number b_i^l for each ball $\mathcal{B}_i^l$ such that

$$\|E \, \mathbf{u} - b_i^l\|_{L^2(\mathcal{B}_i^l)} \leq C \, \mathrm{diam}(\mathcal{B}_i^l) \, |E \, \mathbf{u}|_{H^1(\mathcal{B}_i^l)}.$$

Here, C is a Poincare constant on the unit ball. Hence, for all balls $\mathcal{B}_i^l$

$$\min_{b_i^l \in \mathbb{R}} \|u - t_0 \, b_i^l\|^2_{l^2(\tilde{A}_i^l)} \leq C \, h^{-d} \, \mathrm{diam}(\mathcal{B}_i^l)^2 \, |E \, \mathbf{u}|^2_{H^1(\mathcal{B}_i^l)}$$

holds. From the assumption $\mathrm{diam}(\mathcal{B}_i^l) \leq C \, 3^l \, h$, the bounded overlaps of the balls $\mathcal{B}_i^l$, the property $|E \, \mathbf{u}|_{H^1(\mathbb{R}^d)} \leq C \, |\mathbf{u}|_{H^1(\Omega)}$, the well known estimate $\rho(A) \leq C \, h^{-2}$, and the H^1-equivalence of $a(\cdot, \cdot)$, we get

$$\sum_{i=1}^{n_{l+1}} \min_{w \in \mathbb{R}^1} \|u - t_0 \, w\|^2_{l^2(\tilde{\mathcal{A}}_i^l)} \leq \frac{9^l}{h^{d-2}} \, |E \, \mathbf{u}|^2_{H^1(\mathbb{R}^d)} \leq C'_A \frac{9^l}{\rho(A) \, h^d} \, a(\mathbf{u}, \mathbf{u}) \leq C_A \frac{9^l}{\rho(A)} \|u\|^2_A$$

completing the verification of (5.5.4).

5.7 Similar Methods

A simple aggregation method has been proposed by Braess in [Braess 1995]. Braess clusters the unknowns into groups of between one and four unknowns (see Figure 5.6).

Those groups are constructed in two steps. In the first step, strongly coupled unknowns are glued to pairs. After that, the pairs are combined to groups of two pairs.

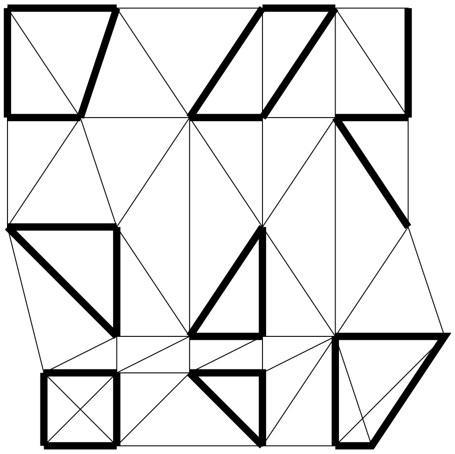

Figure 5.6: Typical groups of unknowns.

ALGORITHM 5.7.1 *Let an $n_l \times n_l$ matrix A_l with an corresponding graph $G_{A_l}(V_l, E_l)$ be given. Then,* `Group(`$G_{A_l}(V_l, E_l)$`)` *generates a disjoint covering $\{\mathcal{G}_i\}_{i=1}^{n_{l+1}}$ of the set $V_l = \{v_1, \ldots, v_{n_l}\}$.*

Group($G_{A_l}(V_l, E_l)$)
{
    /* construction of pairs */
    $U = V_l$; $q = 0$;
    while($U \neq \{\}$)
    {
        choose $v_i \in U$;
        if($N_i \cap U$ = $\{\}$) $\mathcal{P}_q = \{v_i\}$;
        else
        {
            determine $v_j \in U$ with $\frac{a_{i,j}^2}{a_{i,i}\, a_{j,j}}$ maximal;
            $\mathcal{P}_q = \{v_i, v_j\}$;
        }
        $U = U \setminus \mathcal{P}_q$; $q = q + 1$;
    }
    /* construction of groups */
    $U = \bigcup_{i<q} \mathcal{P}_i$; $q = 0$;
    while($U \neq \{\}$)
    {
        choose $\mathcal{P}_i \in U$;
        $z_{i,j}$ = number of $\{a_{s,t} \,|\, s \in \mathcal{P}_i,\ t \in \mathcal{P}_j\}$;
        if($z_{i,j} = 0\ \forall j$) $\mathcal{G}_q = \mathcal{P}_i$;
        else
        {
            determine j^* with z_{i,j^*} is maximal;

$\mathcal{G}_q = \mathcal{P}_i \cup \mathcal{P}_{j^*}\,;$
}
$U = U \setminus \mathcal{G}_q\,;\;\; q = q+1\,;$
}
}

N_i (see Section 1.2) denotes the neighborhood of the node i.

The prolongation $P_{l,l+1} = (p_{i,j})$, the restriction $R_{l+1,l}$ and the coarse grid matrices A_{l+1} are defined by

$$p_{i,j} = \begin{cases} 1 & : \; i \in \mathcal{G}_j, \\ 0 & : \; i \notin \mathcal{G}_j, \end{cases} \qquad R_{l+1,l} = P^T_{l,l+1}, \quad A_{l+1} = \frac{1}{\alpha}\, R_{l+1,l}\, A_l\, P_{l,l+1},$$

where α is a constant (e.g. $\alpha = 1.8$).

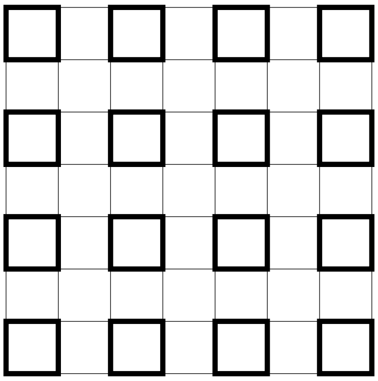

Figure 5.7: Regular clustering.

In order to motivate the factor α, we observe that, using the regular clustering in Figure 5.7, the coarse grid matrix A_{l+1} for

$$A_l = \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}$$

is given by

$$A_{l+1} = \frac{1}{\alpha} \begin{bmatrix} & -2 & \\ -2 & 8 & -2 \\ & -2 & \end{bmatrix}.$$

The coarse grid correction $c_l = P_{l,l+1}\, A^{-1}_{l+1}\, R_{l+1,l}\, A_l\, e_l$ for a smooth error e_l should be pretty close to the error e_l. In the situation in Figure 5.7, a smooth residual d_l is approximately multiplied with a factor 4 after the restriction

$$(d_{l+1})_i = \sum_{j \in \mathcal{G}_i} (d_l)_j \approx 4 \cdot (d_l)_j, \quad j \in \mathcal{G}_i.$$

If e_l and d_{l+1} are close to the eigenvectors corresponding to the smallest eigenvalues of A_l and A_{l+1} respectively, c_l is approximately given by

$$c_l = 4\,\frac{\lambda_{\min}(A_l)}{\lambda_{\min}(A_{l+1})}\,e_l\,.$$

Since (see 2.1.2)

$$\begin{aligned}\lambda_{\min}(A_l) &\approx 4(\pi\,h_l)^2,\\ \lambda_{\min}(A_{l+1}) &\approx 2\,\frac{1}{\alpha}\cdot 4(\pi\,h_{l+1})^2 = \frac{1}{\alpha}\cdot 8(\pi\,2\,h_l)^2 = \frac{1}{\alpha}\cdot 32(\pi\,h_l)^2,\end{aligned}$$

we get

$$c_l = 4\,\frac{\alpha}{8}\,e_l,$$

thus, $\alpha \approx 2$ should be a good choice.

In [Braess 1995], the method discussed above is only applied as a 4-level method ($l_{\max} = 3$), independent of the number of unknowns.

5.8 Numerical Experiments

In [Wesseling 1988], average convergence rates for a cell-centered multigrid method with prolongation $P^{(0)}$ and restriction $R^{(1)}$ (see Section 5.1) are described.

Experiment 5.8.1

$$\begin{aligned}-\nabla\cdot(D(x)\nabla u(x)) &= x_1\,x_2, \quad x \in \Omega = (0,1)\times(0,1),\\ u(x) &= x_1^2 + x_2^2, \quad x \in \partial\Omega\end{aligned}$$

is discretized on an equidistant grid with mesh size h. $D(x) = D_1 = 2$ outside the shaded regions and $D(x) = D_2 = 1/3\cdot 10^5$ inside the shaded regions (see Figure 5.8).

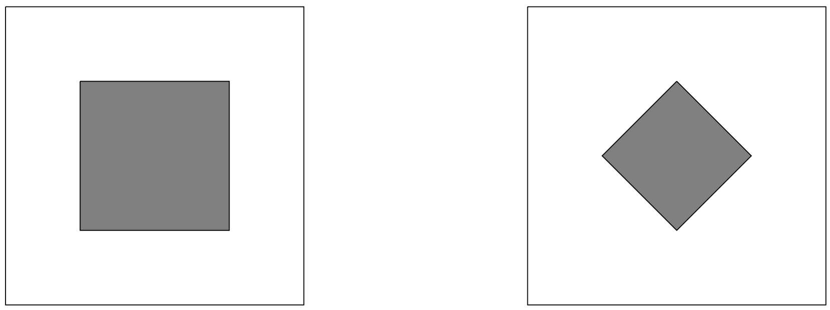

Figure 5.8: Diffusion coefficient for Experiment 5.8.1.

The average convergence rates for Experiment 5.8.1 are presented in Table 5.1 for these sizes of the shaded regions which yield the worst convergence rates.

h	convergence, Figure 5.8 left	convergence, Figure 5.8 right
1/16	0.362	0.300
1/32	0.304	0.273
1/64	0.290	0.237

Table 5.1: Average convergence rates for Experiment 5.8.1.

The convergence of the method in [Vanek, Mandel, and Brezina 1996; Vanek, Brezina, and Mandel 1998] has been tested for the prolongator smoother in (5.4.7) and $\nu_1 = \nu_2 = \gamma = 1$. The pre-smoother is an SOR step with relaxation parameter $\omega = 1.0$ in the forward sweep and $\omega = 1.85$ in the backward sweep. Post-smoothing is symmetric consisting of a backward sweep $\omega = 1.85$ and a forward sweep $\omega = 1.0$. The average convergences rates for a only 10^{-5} reduction of the residual can be found in Table 5.2.

Experiment 5.8.2 *The anisotropic problem with jumps in the coefficients*

$$\begin{aligned} -\nabla \cdot \left(\begin{pmatrix} \epsilon & 0 \\ 0 & \epsilon^{-1} \end{pmatrix} \nabla u(x) \right) &= f(x), \quad x \in \Omega = (0,1) \times (0,1), \\ u(x) &= 0, \quad x \in \partial\Omega, \\ \epsilon &= \begin{cases} 0.01 & : \ x \in (0,0.5) \times (0,0.5), \\ 1 & : \ x \in (0,0.5) \times (0.5,1), \\ 100 & : \ x \in (0.5,1) \times (0,1), \end{cases} \end{aligned}$$

is discretized with piecewise linear basis functions on a regular grid with 10^6 *unknowns.*

Experiment 5.8.3 *The 3D problem*

$$\begin{aligned} -\nabla \cdot (D(x)\, \nabla u(x)) &= f(x), \quad x \in \Omega = (0,1) \times (0,1) \times (0,1), \\ u(x) &= 0, \quad x \in \partial\Omega, \\ D(x) &= \begin{pmatrix} \exp(r_1(x)) & 0 & 0 \\ 0 & \exp(r_2(x)) & 0 \\ 0 & 0 & \exp(r_3(x)) \end{pmatrix} \end{aligned}$$

($r_i(x) \in [\ln(10^{-2}), \ln(10^{+2})]$) with random coefficients $r_i(x)$ is discretized with piecewise linear basis functions on a regular grid with 68921 *unknowns.*

In [Braess 1995], the performance of the algebraic multigrid method proposed in [Braess 1995] as preconditioner in a conjugate gradient algorithm is compared with line-SSOR. The number of iteration steps for an only 10^{-5} reduction of the residual are reported for three different groundwater simulation problems. The smoother for the algebraic multigrid method is line-SSOR and three coarse grids ($l_{\max} = 3$) are generated. Note that the coarsening in [Vanek, Brezina, and Mandel 1998] is much

Experiment	convergence	$l_{\max}$
5.8.2	0.1	4
5.8.3	0.21	3

Table 5.2: Average convergence rates.

preconditioner	realization	n_0	iter. steps	time
line-SSOR	1	9246	30	4.6 s
Braess_AMG	1	9246	7	5.4 s
line-SSOR	2	16464	278	83 s
Braess_AMG	2	16464	128	105 s
line-SSOR	3	45666	150	60 s
Braess_AMG	3	45666	30	81 s

Table 5.3: Iteration steps of the preconditioned CG-methods.

faster than in [Braess 1995]. The numbers of iteration steps and the computing times are shown in Table 5.3.

For the five-point stencil (1.2.4), the numbers of conjugate gradient steps with ICC (incomplete Cholesky) as preconditioner or the algebraic multigrid method by Braess ($l_{\max} = 3$ and SSOR ($\omega = 1$) smoothing) as preconditioner are compared in Table 5.4.

n_0	preconditioner	iter. steps	time
400	Braess_AMG	5	0.13 s
400	ICC	15	0.08 s
1600	Braess_AMG	6	0.38 s
1600	ICC	26	0.46 s
6400	Braess_AMG	7	1.62 s
6400	ICC	49	3.00 s
14400	Braess_AMG	7	3.76 s
14400	ICC	72	9.84 s

Table 5.4: Iteration steps of the preconditioned CG-methods.

6 ILU-type Methods

The main goal of this section is to show the close relationship of algebraic multigrid methods (especially hierarchical basis multigrid schemes, Section 2.7) and ILU schemes (see e.g. [Hackbusch and Wittum 1993; Saad 1996b]). An ILU method with some characteristic multigrid features (NGILU) has been introduced in [van der Ploeg, Botta, and Wubs 1996]. In [Bank and Xu 1994; Bank and Smith 1998] an (algebraic) hierarchical basis multigrid method is constructed using an ILU technique. A similar technique is applied for the construction of an algebraic multigrid method in [Bank and Wagner 1999].

6.1 Graph Theoretical Aspects

As already discussed in Section 1.2, a sparse matrix A is always connected with a graph $G_A(V,E)$, consisting of a set of vertices (nodes, unknowns) $V = \{v_1, \dots v_n\}$ and a set of edges E. For a vertex v_i, the set of neighbor vertices N_i is defined by

$$N_i = \{\, v_j \in V \mid e_{i,j} \in E \,\}.$$

Definition 6.1.1 *A clique $V_C \subseteq V$ is a set of vertices which are all pairwise connected; that is*

$$v_i, v_j \in V_C, i \neq j \;\; \Rightarrow \;\; e_{i,j} \in E.$$

With a proper ordering of the vertices a clique corresponds to a dense submatrix of the matrix A.

In graph theoretical terms, a single Gaussian elimination step transforms the graph $G_A(V,E)$ to a new graph $G'_{A'}(V',E')$ as follows.

1. Eliminate the vertex v_i and all its incident edges from G_A. Denote the resulting set of edges $E_1 \subseteq E$. Set $V' = V - \{v_i\}$.

2. For each distinct pair $v_j, v_r \in N_i$ in V', add the edge $e_{j,r}$ to the set of E_F of fill-in edges if not already present in E_1. Set $E' = E_1 \cup E_F$.

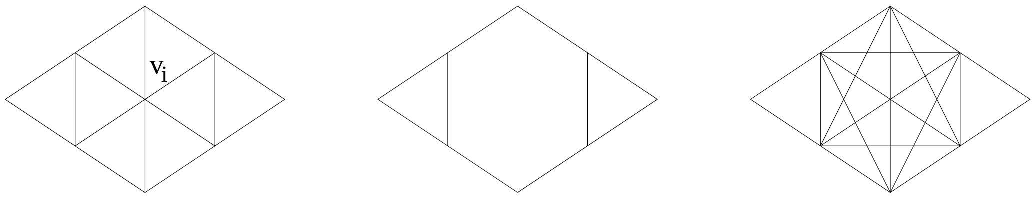

Figure 6.1: Illustration of a single Gaussian elimination step. The original mesh is shown on the left. v_i and its incident edges are removed (middle) and fill-in edges are added (right).

The elimination process is illustrated in Figure 6.1.

Note that the set N_i in G_A becomes a clique in $G'_{A'}$. Within this framework, the classical ILU-decomposition does not allow any fill-in $E_F = \{\}$. This forces the new matrix A' corresponding to $G'_{A'}$ to have the same sparsity structure as the corresponding submatrix of A. The graph $G'_{A'}$ would then correspond to the center of Figure 6.1.

To interpret hierarchical basis multigrid (HBMG, Section 2.7) as a generalized ILU scheme, we consider the case of two nested meshes, where the fine mesh is an uniform refinement of the coarse mesh, generated by pairwise connecting the midpoints of the coarse grid edges in the usual way [Bank, Dupont, and Yserentant 1988; Bank 1994; Hackbusch 1985]. Here the direct sum decomposition $V = C \oplus F$, where C is the set of coarse grid vertices and F is the of fine grid vertices (those not in C) can be made. For each vertex v_i there is an unique pair of vertices $v_j, v_k \in C$ such that v_i is the midpoint of the edge connecting v_j and v_k on the coarse grid ($v_i = (v_j + v_k)/2$). The pair v_j, v_k is called vertex parents.

We now view HBMG as an ILU algorithm in which only selected fill-in edges are allowed. In this algorithm, we sequentially eliminate the vertices in the set F as follows.

1. Eliminate the vertex v_i and all its incident edges from G_A. Denote the resulting set of edges $E_1 \subseteq E$. Set $V' = V - \{v_i\}$.

2. Let $v_j, v_k \in N_i$ denote the parents of v_i. Create a set E_F of fill-in edges of the form
$$e_{j,m} \text{ or } e_{k,m} \text{ with } v_m \in N_i,\ m \neq k, j$$
for edges not already present in E_1. Set $E' = E_1 \cup E_F$.

In other words, the classical HBMG algorithm connects the vertex parents $v_j, v_k \in C$ of an vertex $v_i \in F$ with all neighbors $v_m \in N_i$ of v_i.

An even more simple elimination strategy is to use just one vertex parent. Although this does not correspond to the HBMG, it has been studied in a different context as a partitioning scheme for general graphs [Karypis and Kumar 1995a; Karypis and Kumar 1995b]. The elimination algorithm is similar to the case of two parents

1. Eliminate the vertex v_i and all its incident edges from G_A. Denote the resulting set of edges $E_1 \subseteq E$. Set $V' = V - \{v_i\}$.

2. Let $v_j \in N_i$ denote the parent of v_i. Create a set E_F of fill-in edges of the form

$$e_{j,m} \text{ with } v_m \in N_i, \ m \neq j$$

for edges not already present in E_1. Set $E' = E_1 \cup E_F$.

The one-parents scheme produces fewer fill-in edges. When the initial graph is a finite element triangulation (or a tetrahedral mesh in three space dimensions), the graph $G'_{A'}$ remains a finite element triangulation. This property can be maintained at all steps of the elimination process through a careful selection of the parents. In general, this is not possible for the two-parents scheme. Both elimination techniques are illustrated in Figure 6.2.

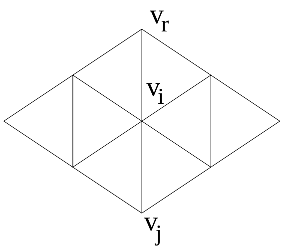

Figure 6.2: Illustration of the one- and two-parents elimination schemes. The original mesh is shown on the left. The fill-in patterns for two parents (v_j and v_k) and one parent v_j are shown in the middle and on the right respectively.

The extension to more than two parents nodes is straightforward. Allowing all vertices in N_i to be parents results in classical Gaussian elimination.

In the case of a sequence of regularly refined meshes, if the triangulation represents the stiffness matrix on the fine grid, after all fine-grid nodes are eliminated using either the one- or the two-parents scheme, the resulting graph is just the coarse grid triangulation. This is illustrated in Figure 6.3 and Figure 6.4.

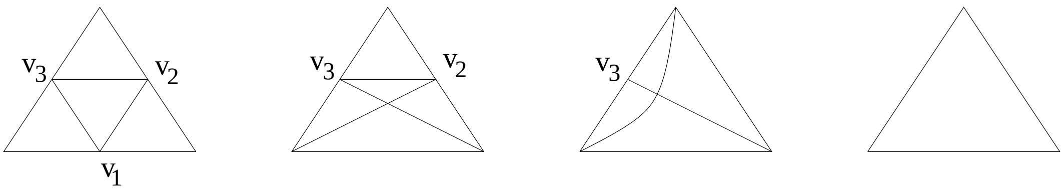

Figure 6.3: Two-parents elimination schemes for a regularly refined sequence of grids.

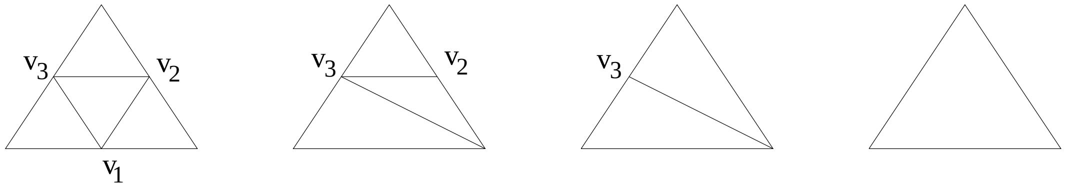

Figure 6.4: One-parents elimination schemes for a regularly refined sequence of grids.

For unstructured meshes, the main problem is to determine reasonable vertex parents for each vertex to be eliminated. Once this is done, the elimination (coarsening) is done exactly as in the case of nested meshes. This may lead to graphs which are not finite element triangulations.

6.2 Schur-Complement Approximation

The following simple example illustrates the connection between ILU schemes and the hierarchical basis multigrid algorithm. Let $A \in \mathbb{R}^{n \times n}$ be a sparse matrix arising from the discretization of a partial differential equation. A can be partitioned as

$$A = \begin{pmatrix} d & r^T \\ l & \hat{A} \end{pmatrix}, \quad d \neq 0, \quad r, l \in \mathbb{R}^{n-1}, \quad \hat{A} \in \mathbb{R}^{(n-1)\times(n-1)}.$$

The hierarchical basis multigrid method is based on a change of basis, from the nodal basis to the hierarchical basis (see Figure 2.8). The first step of the transformation involves forming the matrix

$$\begin{aligned} L_1 A U_1 &= \begin{pmatrix} 1 & 0 \\ \tilde{l} & I \end{pmatrix} \begin{pmatrix} d & r^T \\ l & \hat{A} \end{pmatrix} \begin{pmatrix} 1 & \tilde{r}^T \\ 0 & I \end{pmatrix} \\ &= \begin{pmatrix} d & r^T + d\,\tilde{r}^T \\ l + d\,\tilde{l} & \hat{A} + \tilde{l}\,r^T + l\,\tilde{r}^T + \tilde{l}\,d\tilde{r}^T \end{pmatrix}. \end{aligned}$$

If the first vertex v_1 is associated with the refinement of the edge between v_i and v_j, for classical HBMG, $\tilde{l}$ and $\tilde{r}$ are given by

$$\tilde{l} = \tilde{r} = (e_i + e_j)/2,$$

where e_i is the i-th column vector of the identity matrix I_{n-1}.

On the other hand, the two-parents elimination scheme chooses the vectors $\tilde{l}$ and $\tilde{r}$ according to

$$\tilde{l} = \tilde{l}_i\, e_i + \tilde{l}_j\, e_j, \quad \tilde{r} = \tilde{r}_i\, e_i + \tilde{r}_j\, e_j,$$

assuming that v_i and v_j are the parents of the first vertex v_1. Here $\tilde{l}_i, \tilde{l}_j, \tilde{r}_i, \tilde{r}_j$ are simply the multipliers in an incomplete LU factorization of the matrix A.

The matrix $\tilde{l}\, r^T + l\, \tilde{r}^T + \tilde{l}\, d\tilde{r}^T$ typically creates some fill-in in the rows and columns corresponding to the vertex parents. These are precisely the fill-in edges shown in Figure 6.2. Note that the elimination step can be viewed as adding a rank one perturbation to the exact Schur-complement $A_S = \hat{A} - l\, d^{-1}\, r^T$

$$\hat{A} + \tilde{l}\, r^T + l\, \tilde{r}^T + \tilde{l}\, d\tilde{r}^T = A_S + (\tilde{l} + d^{-1}\, l)\, d\, (\tilde{r} + d^{-1}\, r)^T.$$

In the standard formulation of HBMG as well as for the incomplete LU decomposition, the next step is a transformation of the same form applied to the matrix $\hat{A} + \tilde{l}\, r^T + l\, \tilde{r}^T + \tilde{l}\, d\tilde{r}^T$.

Two popular choices exist for the selection of the multipliers in $\tilde{l}$ and $\tilde{r}$. For the ILU decomposition, the multipliers are chosen to zero certain components in $\tilde{l} + d^{-1}\, l$ and $\tilde{r} + d^{-1}\, r$. In the MILU approach (see [Gustafsson 1978; Hackbusch and Wittum 1993; Wittum 1989b]), the multipliers are chosen that

$$A\, t_r = M_{\text{MILU}}\, t_r \quad \text{and} \quad t_l^T\, A = t_l^T\, M_{\text{MILU}}$$

holds for certain test vectors t_r, t_l. The usual choice is $t_r, t_l = (1, \ldots, 1)^T$.

6.3 The Incomplete Factorization Multigraph Algorithm

The ILU-type algebraic multigrid methods (e.g. [Bank and Smith 1998; Bank and Wagner 1999; van der Ploeg, Botta, and Wubs 1996]) differ in the approximation of the Schur-complements, i.e. the choice of the multipliers in $\tilde{l}$ and $\tilde{r}$, and in the ordering of the unknowns. In this section, the method proposed in [Bank and Smith 1998] is discussed.

6.3.1 Ordering

Ordering means to find a permutation matrix $\mathcal{P}$ such that the reordered matrix $\mathcal{P}\, A\, \mathcal{P}^T$ has some desired properties. For classic sparse Gaussian elimination, the permutation matrix $\mathcal{P}$ is constructed based solely on the graph of the matrix (e.g. a minimum degree ordering [George and Liu 1981]) and not on the values of the matrix elements. In the multigraph algorithm, both the graph and the numerical values of the matrix are used to construct the ordering and the vertex parents. To simplify the description, only the ordering and the elimination of one vertex v_i is explained. The remaining vertices are ordered inductively by the same algorithm.

Let $f^{(i,j)}$ describe the number of fill-in edges which must be added if $v_j \in N_i$ is chosen as the only vertex parents of v_i. Then, the quality $r_\gamma^{(i,j)}$ is given by

$$r_\gamma^{(i,j)} = \frac{|a_{i,j}| + |a_{j,i}|}{\rho_i (f^{(i,j)} + \gamma)}, \quad \gamma > 0,$$

where

$$\rho_i = |a_{i,i}| + \sum_{j \neq i} (|a_{i,j}| + |a_{j,i}|).$$

Then, for the case of one vertex parent, the quality function $q_1(v_i)$

$$q_1(v_i) = \max_{v_j \in N_i} r_\gamma^{(i,j)} \tag{6.3.1}$$

represents a compromise between choosing a parent v_j which is as strongly connected to v_i as possible, and choosing v_j to cause as little fill-in as possible. The parameter γ indirectly controls the number of fill-in edges resulting from the ordering. Experimentally, $\gamma = 10$ turned out to be a good choice for the one parent algorithm.

The quality function $q_2(v_i)$ for the two parent scheme is developed in a similar fashion. Suppose that $v_j, v_k \in N_i$ $(k \neq j)$. Then, the qualities are given by

$$r_\gamma^{(i,j,k)} = \frac{|a_{i,j}| + |a_{j,i}| + |a_{i,k}| + |a_{k,i}|}{\rho_i(f^{(i,j,k)} + \gamma)},$$

where $f^{(i,j,k)}$ is the number of fill-in edges required if the pair $\{v_j, v_k\}$ is chosen as parents. $q_2(v_i)$ is then defined by

$$q_2(v_i) = \max\{\hat{q}_0(v_i), \hat{q}_1(v_i), \hat{q}_2(v_i)\} \tag{6.3.2}$$

with

$$\hat{q}_0(v_i) = \frac{|a_{i,i}|}{\rho_i\, \gamma}, \quad \hat{q}_1(v_i) = \max_{v_j \in N_i} r_\gamma^{(i,j)}, \quad \hat{q}_2(v_i) = \max_{v_j, v_k \in N_i} r_\gamma^{(i,j,k)}.$$

$\gamma = 50$ was determined experimentally for the two parent algorithm. Actually, the two parent algorithm offers the possibility to each vertex of having zero, one or two parents. Along with the quality function, tentative parents are assigned to each vertex. In case where no parents can be assigned, $q_p(v_i) = 0$, $(p = 1, 2)$. We are now in the position to formulate the ordering algorithm.

Algorithm 6.3.1 *Let an $n \times n$ matrix A be given and quality functions q_p be defined by (6.3.1) or (6.3.2). Then, the ordering algorithm according to [Bank and Smith 1998] reads*

```
Order(p)
{
    /* Initialization */
    for(i ≤ n)
    {
        compute q_p(v_i);
        U = U ∪ {v_i};
    }
    /* Elimination */
    get v_i ∈ U with q_p(v_i) maximal;
    while (v_i exists && q_p(v_i) > 0)
```

```
    {
        U = U \ {v_i};
        order the vertex v_i;
        the tentative parents of v_i become the actual parents;
        eliminate v_i;
        update the graph and the partially factored matrix;
        for(v_j ∈ N_i)) update q_p(v_j);
        get v_i ∈ U with q_p(v_i) maximal;
    }
}
```

There is an interesting modification of Algorithm 6.3.1. We might set (artificially) $q_p(v_j) = 0$ for all vertices which have been chosen as parent vertices. Hence these vertices v_j will not be eliminated. When U contains only vertices v_j with $q_p(v_j) = 0$, the remaining nodes are called coarse grid vertices and those eliminated are fine grid vertices. This provides a two level blocking analogous to two level multigrid methods. Reinitializing $q_p(v_j)$ for the remaining vertices and restarting the elimination process, leads to a natural multilevel blocking.

6.3.2 Factorization

For the factorization, a diagonal matrix Σ is defined by

$$\begin{aligned} \sigma_i &= \max\{0\,,\,-\sum_{j=1}^{n}(a_{i,j}+a_{j,i})\},\\ \Sigma &= \operatorname{diag}(\sigma_1,\ldots,\sigma_n). \end{aligned}$$

Actually, an incomplete factorization of $A+\Sigma$ is computed. Such an a priori shift is a simple way to insure the existence and the stability of the factorization. The first elimination step leads to

$$A+\Sigma = \begin{pmatrix} d & r^T \\ l & \hat{A} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l\,d^{-1} & I \end{pmatrix} \begin{pmatrix} d & 0 \\ 0 & \hat{A}-l\,d^{-1}\,r^T \end{pmatrix} \begin{pmatrix} 1 & d^{-1}\,r^T \\ 0 & I \end{pmatrix}.$$

The sparsity pattern of the matrix $\hat{A} - l\,d^{-1}\,r^T$ will generally not coincide with the allowed pattern. Thus, $l\,d^{-1}\,r^T$ is decomposed into

$$-l\,d^{-1}\,r^T = S_1 + N_1,$$

where S_1 has the required sparsity pattern. N_1 is the error matrix for the first step. With the same technique, the first column/row of $\hat{A}+S_1$ is eliminated. If the factorization is inductively continued as $\hat{A}+S_1 = \bar{A} = \bar{L}\,\bar{D}\,\bar{U}+\bar{N}$, we get

$$\begin{aligned} A+\Sigma &= \begin{pmatrix} 1 & 0 \\ l\,d^{-1} & I \end{pmatrix} \begin{pmatrix} d & 0 \\ 0 & \bar{A}+N_1 \end{pmatrix} \begin{pmatrix} 1 & d^{-1}\,r^T \\ 0 & I \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ l\,d^{-1} & \bar{L} \end{pmatrix} \begin{pmatrix} d & 0 \\ 0 & \bar{D} \end{pmatrix} \begin{pmatrix} 1 & d^{-1}\,r^T \\ 0 & \bar{U} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \bar{N}+N_1 \end{pmatrix} \\ &= L\,D\,U + N'. \end{aligned} \tag{6.3.3}$$

Thus,

$$A = L\,D\,U + N' - \Sigma = L\,D\,U + N.$$

The matrix $-l\,d^{-1}\,r^T$ is decomposed as $S_1 + N_1$ by a procedure similar to classical MILU [Gustafsson 1978; Wittum 1989b]. Suppose that $i > j$ and the entry $l_i\,d^{-1}\,r_j$ is not allowed in the fill-in pattern. Consider the matrix $F^{(i,j)}$ which is zero except for the four elements

$$F^{(i,j)}_{i,j} = -l_i\,d^{-1}\,r_j, \quad F^{(i,j)}_{j,i} = -l_j\,d^{-1}\,r_i, \quad F^{(i,j)}_{i,i} = F^{(i,j)}_{j,j} = (l_i\,d^{-1}\,r_j + l_j\,d^{-1}\,r_i)/2.$$

Then, N_1 and S_1 are defined by

$$N_1 = \sum_{(i,j)'} F^{(i,j)}, \quad S_1 = -l\,d^{-1}\,r^T - N_1,$$

where the sum is taken over all pairs $(i,j)'$ falling outside the allowed fill-in pattern. This is a typical MILU approach, aside from the unusual averaging of the diagonal entries appearing in $F^{(i,j)}$.

ALGORITHM 6.3.2 *The incomplete factorization multigraph preconditioner M is defined by*

$$M = L\,D\,U$$

with L, D, U in (6.3.3).

6.4 The Multilevel ILU Decomposition

The multilevel ILU decomposition (MLILU) [Bank and Wagner 1999] is an ILU-type algebraic multigrid method including more classic multigrid features than the incomplete factorization multigraph algorithm. In particular, distinct levels are defined which allow the computation of smoothing steps on the coarse grids.

6.4.1 The Labeling Scheme

Since the labeling algorithm proposed in [Bank and Wagner 1999] is rather complex, the description of the scheme has to be restricted to some basic features.

Similar to the incomplete factorization multigraph algorithm, parent nodes are assigned to each node which is eliminated. In general, two parent nodes must be assigned. Only in some special cases, one or more than two parent nodes are allowed.

The basic idea of the labeling strategy is to mark all nodes on each level either as F-node or as C-node. While the F-nodes are eliminated, the C-nodes represent the unknowns on the next coarser level. Only C-nodes are allowed to be parent nodes. A node is only marked as F-node, if an appropriate set of parent nodes can be assigned.

Aside from a couple of exceptions, an appropriate set of parent nodes for the node (vertex) v_i needs to contain two strongly coupled neighbors v_j and v_k, i.e.

$$a^l_{i,j} \geq \sigma \max_q \{a^l_{i,q}\}, \quad a^l_{i,k} \geq \sigma \max_q \{a^l_{i,q}\}$$

with a parameter σ, $0 < \sigma < 1$. Since $A_l = (a^l_{i,j})$, "strongly coupled" refers to the matrix A_l on the currently labeled level and therefore not to the partially factored matrix as in Section 6.3.

In order to locally minimize the number of fill-in edges and the number of C-nodes, each pair of neighbor nodes v_j and v_k strongly coupled to v_i gets a weight

$$w_i^{(j,k)} = \alpha_0 - \alpha_1\, nc_i^{(j,k)} - \alpha_2\, ne_i^{(j,k)} - \alpha_3\, \xi^{(j,k)}, \quad \alpha_0, \alpha_1, \alpha_2, \alpha_3 \geq 0 \tag{6.4.1}$$

depending on

- the number of new C-nodes $nc_i^{(j,k)}$,
- the number of fill-in edges $ne_i^{(j,k)}$,
- the coupling $\xi^{(j,k)}$ between v_j and v_k.

$\xi^{(j,k)}$ is small for weakly coupled v_j and v_k.

The weight w_i for the node v_i is then defined by

$$w_i = \max_{v_j,\, v_k \in N_i} \{w_i^{(j,k)}\} \tag{6.4.2}$$

and the corresponding pair of parent nodes is called optimal pair. We can now formulate the labeling algorithm for one level l. The same algorithm is applied to all levels. The construction of the coarse grid matrices is discussed in the following section.

Algorithm 6.4.1 *Let an $n_l \times n_l$ matrix A_l be given and weights w_i be defined by (6.4.1) and (6.4.2). Then, the labeling of the unknowns on level l is performed by* `Label`(A_l).

```
Label(A_l)
{
    /* Initialization */
    for(i ≤ n)
    {
        if (|N_i| = {}) label v_i as F-node; /* no parent nodes needed */
        else
        {
            compute w_i;
            U = U ∪ {v_i};
        }
    }
```

```
    /* Labeling */
    while (nodes with an appropriate set of parent nodes exist)
    {
        get v_i ∈ U with w_i maximal;
        U = U \ {v_i};
        label the node v_i as F-node;
        the optimal parents v_j, v_k of v_i become the actual parents;
        label the parent nodes v_j, v_k as C-nodes;
        for(v_q ∈ N_i ∪ N_j ∪ N_k) update w_q;
    }
    /* Remaining nodes */
    while (U ≠ {})
    {
        get v_i ∈ U ;
        label the node v_i as C-node;
        U = U \ {v_i};
    }
}
```

6.4.2 Decomposition

As a first step, we describe the elimination of the first column of the matrix $A^{(i)} \in \mathbb{R}^{n^{(i)} \times n^{(i)}}$

$$A^{(i)} = \begin{pmatrix} d_i & r_i^T \\ l_i & \bar{A}^{(i)} \end{pmatrix},$$

$d_i \in \mathbb{R}$, $d_i \neq 0$, $r_i, l_i \in \mathbb{R}^{n^{(i+1)}}$, $\bar{A}^{(i)} \in \mathbb{R}^{n^{(i+1)} \times n^{(i+1)}}$, $n^{(i+1)} = n^{(i)} - 1$.

Exact Gaussian elimination leads to the factorization

$$A^{(i)} = \begin{pmatrix} 1 & 0 \\ l_i d_i^{-1} & I_{n^{(i+1)}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & A_{\mathrm{S}}^{(i)} \end{pmatrix} \begin{pmatrix} d_i & r_i^T \\ 0 & I_{n^{(i+1)}} \end{pmatrix}$$

with the $n^{(i+1)} \times n^{(i+1)}$ identity matrix $I_{n^{(i+1)}}$ and the Schur-complement

$$A_{\mathrm{S}}^{(i)} = \bar{A}^{(i)} - l_i d_i^{-1} r_i^T.$$

In order to limit the fill-in, the Schur-complement is approximated by

$$A^{(i+1)} = \bar{A}^{(i)} - l_i d_i^{-1} \bar{r}_i^T - \bar{l}_i d_i^{-1} r_i^T + \bar{l}_i d_i^{-1} \bar{r}_i^T, \quad \bar{r}_i, \bar{l}_i \in \mathbb{R}^{n^{(i+1)}}. \tag{6.4.3}$$

The values of the non-zero components in $\bar{r}_i$ and $\bar{l}_i$ are controlled by a test vector t. Since the size $n^{(i)} \times n^{(i)}$ of the matrices $A^{(i)}$ decreases for increasing i, only the last $n^{(i)}$ components of the test vector are required for the i-th elimination step. We therefore define the operator $[v]_m$ which returns the last m components of the vector v. Thus, the test vector $t^{(i)}$ for the i-th elimination step is given by $t^{(i)} = [t]_{n^{(i)}}$.

The standard choice for t is $t = (1, \dots, 1)^T$. In particular, the non-zero components of $\bar{r}_i$ and $\bar{l}_i$ are given by

$$(\bar{r}_i)_j = \begin{cases} \dfrac{|(r_i)_j| \sum\limits_{z \le n^{(i+1)}} (r_i)_z t_{z+1}^{(i)}}{\sum\limits_{z \in P_i} |(r_i)_z|\, t_{z+1}^{(i)}} & : \; j \in P_i \;\wedge \sum\limits_{z \in P_i} |(r_i)_z|\, t_{z+1}^{(i)} \ge 0, \\ 0 & : \; \text{otherwise}, \end{cases}$$

$$(\bar{l}_i)_j = \begin{cases} \dfrac{|(l_i)_j| \sum\limits_{z \le n^{(i+1)}} (l_i)_z t_{z+1}^{(i)}}{\sum\limits_{z \in P_i} |(l_i)_z|\, t_{z+1}^{(i)}} & : \; j \in P_i \;\wedge \sum\limits_{z \in P_i} |(l_i)_z|\, t_{z+1}^{(i)} \ge 0, \\ 0 & : \; \text{otherwise}. \end{cases}$$

If $\sum\limits_{z \in P_i} |(r_i)_z|\, t_{z+1}^{(i)} \ge 0$ and $\sum\limits_{z \in P_i} |(l_i)_z|\, t_{z+1}^{(i)} \ge 0$, the construction scheme satisfies the filter conditions

$$(0, \bar{r}_i^T - r_i^T)\, t^{(i)} = 0, \quad (t^{(i)})^T \begin{pmatrix} 0 \\ \bar{l}_i - l_i \end{pmatrix} = 0.$$

Since,

$$A^{(i+1)} - A_S^{(i)} = (\bar{l}_i - l_i) d_i^{-1} (\bar{r}_i - r_i)^T,$$

the error matrix $N^{(i)}$

$$\begin{aligned} N^{(i)} &= \begin{pmatrix} 1 & 0 \\ l_i d_i^{-1} & I_{n^{(i+1)}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & A^{(i+1)} \end{pmatrix} \begin{pmatrix} d_i & r_i^T \\ 0 & I_{n^{(i+1)}} \end{pmatrix} - A^{(i)} \\ &= \begin{pmatrix} 1 & 0 \\ l_i d_i^{-1} & I_{n^{(i+1)}} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & A^{(i+1)} - A_S^{(i)} \end{pmatrix} \begin{pmatrix} d_i & r_i^T \\ 0 & I_{n^{(i+1)}} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ 0 & (\bar{l}_i - l_i)\, d_i^{-1}\, (\bar{r}_i - r_i)^T \end{pmatrix} \end{aligned}$$

is symmetric and positive semi-definite if $A^{(i)}$ is symmetric and positive definite.

Using the same technique, all F-nodes are eliminated iteratively. This yields the two-level decomposition

$$M_{l,\mathrm{TL}} = L_l \begin{pmatrix} I_{n_{l,F}} & 0 \\ 0 & A_{l+1} \end{pmatrix} U_l$$

with a lower triangular matrix L_l and an upper triangular matrix U_l of the form

$$L_l = \begin{pmatrix} * & 0 \\ * & I_{n_{l,C}} \end{pmatrix}, \quad U_l = \begin{pmatrix} * & * \\ 0 & I_{n_{l,C}} \end{pmatrix}.$$

For A_l spd, it can be shown that A_{l+1} is spd and the two-level error matrix $N_{l,\mathrm{TL}} = M_{l,\mathrm{TL}} - A_l$ is symmetric and positive semi-definite. The filter conditions

$$N_{l,\mathrm{TL}}\, t = 0, \quad t^T N_{l,\mathrm{TL}} = 0$$

still hold.

A_{l+1} represents the system matrix on the coarser level $l+1$. Next, on level $l+1$, the unknowns are labeled. Then, the F-nodes on level $l+1$ are eliminated which yields a new matrix A_{l+2}. The same procedure is applied to level $l+2$ ... .

We are now in the position to formulate the MLILU algorithm.

ALGORITHM 6.4.2 *Assume the matrices U_i, L_i and A_i have been computed by an $l_{\max}$-level MLILU decomposition and smoothers S_i are defined. The operator $[v]_m$ returns the last m components of the vector v. Then, the function* $\texttt{MLILU}(0, u, f)$ *calculates one iteration step of the MLILU method.*

$\texttt{MLILU}(i, u_i, f_i)$
$\{$
  $\texttt{if}(i == l_{\max})\ \ u_i = A_i^{-1} f_i\,;$
  $\texttt{else}$
  $\{$
    $u_i = S_i^{\nu_1}(u_i, f_i)\,;$
    $d_i = f_i - A_i\, u_i\,;$
    $d_i = L_i^{-1} d_i\,;$
    $d_{i+1} = [d_i]_{n_{i+1}}\,;$
    $v_{i+1} = 0;$
    $\texttt{for}(j = 0; j < \gamma; j = j+1)\quad \texttt{MLILU}(i+1, v_{i+1}, d_{i+1})\,;$
    $[d_i]_{n_{i+1}} = v_{i+1}\,;$
    $u_i = u_i + U_i^{-1}\, d_i\,;$
    $u_i = S_i^{\nu_2}(u_i, f_i)\,;$
  $\}$
$\}$

$u_i, d_i, f_i, v_i \in \mathbb{R}^{n_i}$.

In Algorithm 6.4.2, the matrices L_i and U_i play the role of restriction and prolongation respectively. Without smoothing steps, the MLILU decomposition becomes an ILU decomposition with a special ordering and a special approximation strategy (6.4.3). In this case, distinct levels are not necessary.

THEOREM 6.4.1 *Let A be spd. Then, the energy norm of the iteration matrix $I_n - M^{-1}A$, where M is the MLILU decomposition of A ($\nu_1 = \nu_2 = 0, \gamma = 1$), is smaller than one*

$$\|I_n - M^{-1}A\|_A < 1$$

independent of the of $\bar{r}_i$ and $\bar{l}_i$ in (6.4.3). The filter conditions

$$M\, t = A\, t, \quad t^T M = t^T A$$

hold.

PROOF. The proof is based on the positive semi-definiteness of the error matrix. A detailed proof can be found in [Bank and Wagner 1999]. □

6.5 The Nested Grids ILU Decomposition

For the Nested grids ILU decomposition (NGILU) introduced in [van der Ploeg, Botta, and Wubs 1996], the unknowns are numbered according to a nested multigrid partitioning. After that, an LU-factorization based on a drop tolerance is constructed.

6.5.1 Numbering

Let a sequence of nested grids $\Omega_0 \supset \Omega_1 \supset \cdots \supset \Omega_{l_{\max}}$ be given. Similar to the hierarchical basis multigrid, the set of unknowns $\tilde{\Omega}_l$ on level l is then defined by

$$\tilde{\Omega}_l = \Omega_l \setminus \Omega_{l+1}, \quad \text{where} \quad \Omega_{l_{\max}+1} = \{\}.$$

Hence, each unknown is assigned to exactly one level. The unknowns are numbered such that if

$$u_i \in \tilde{\Omega}_l \text{ and } u_j \in \tilde{\Omega}_m \text{ with } l < m \quad \Rightarrow \quad i < j.$$

For instance, on an uniform sequence of grids, this may lead to the numbering

1	2	3	4	5	6	7
8	*41*	9	*42*	10	*43*	11
12	13	14	15	16	17	18
19	*44*	20	**49**	21	*45*	22
23	24	25	26	27	28	29
30	*46*	31	*47*	32	*48*	33
34	35	36	37	38	39	40

where

$$u_i \in \tilde{\Omega}_0, \; i \le 40, \quad u_i \in \tilde{\Omega}_1, \; 41 \le i \le 48, \quad \tilde{\Omega}_2 = u_{49}.$$

6.5.2 Factorization

In [van der Ploeg, Botta, and Wubs 1996], a splitting $A = LU + N$ is constructed for matrices A with $\operatorname{diag}(A) = I$. Of course, all matrices A with $a_{i,i} \neq 0$ can be transformed into this form by simple diagonal scaling.

The construction scheme guarantees, that all entries $n_{i,j}$ of the error matrix N are bounded by a threshold parameter $\varepsilon_{i,j}$, $|n_{i,j}| \le \varepsilon_{i,j}$ $(i \neq j)$.

Given the first $i-1$ rows of L and U, the row i of L and U is constructed from $N = A - LU$ as

$$n_{i,j} = a_{i,j} - \sum_{k=1}^{\min\{i,j\}} l_{i,k}\, u_{k,j}. \tag{6.5.1}$$

Let $j < i$. Hence, $u_{k,j}$ has already been calculated for $k \leq j$. Using (6.5.1), $l_{i,j}$ is inductively constructed from

$$n_{i,j} + l_{i,j}\, u_{j,j} = a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}\, u_{k,j}. \tag{6.5.2}$$

starting at $j = 1$. Note that $l_{i,k}$ for $k < j$ has been computed from (6.5.2) with $j = k$. Then,

$$l_{i,j} = \begin{cases} \frac{a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}\, u_{k,j}}{u_{j,j}} & : \quad |a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}\, u_{k,j}| \geq \varepsilon_{i,j} \vee \text{edge } e_{i,j} \text{ exists} \\ 0 & : \quad \text{otherwise.} \end{cases}$$

For $j > i$, (6.5.1) yields with $l_{i,i} = 1$

$$n_{i,j} + u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} l_{i,k}\, u_{k,i}. \tag{6.5.3}$$

Because $u_{k,i}$ is already computed for $k < i$, we can set

$$u_{i,j} = \begin{cases} a_{i,j} - \sum_{k=1}^{i-1} l_{i,k}\, u_{k,j} & : \quad |a_{i,j} - \sum_{k=1}^{i-1} l_{i,k}\, u_{k,j}| \geq \varepsilon_{i,j} \vee \text{edge } e_{i,j} \text{ exists} \\ 0 & : \quad \text{otherwise.} \end{cases}$$

The neglected fill-in $n_{i,j}$ in the case $l_{i,j} = 0$ or $u_{i,j} = 0$ is added to the main diagonal. Hence, with $l_{i,i} = 1$, $u_{i,i}$ is computed from

$$n_{i,i} + u_{i,i} = a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}\, u_{k,i},$$

where

$$n_{i,i} = -\sum_{k \neq i} n_{i,k}$$

and $n_{i,k}$ is calculated according to (6.5.2) and (6.5.3). Hence,

$$\sum_k n_{i,k} = 0,$$

and therefore

$$N\, t = 0, \quad t = (1, \ldots, 1)^T.$$

For a symmetric matrix A, an incomplete Cholesky decomposition with $L^T = U$ is constructed in a similar fashion.

On uniform grids, the drop tolerance is $\varepsilon_{i,j}$ is chosen as

$$u_i \in \tilde{\Omega}_l,\; u_j \in \tilde{\Omega}_k \quad \Rightarrow \quad \varepsilon_{i,j} = \alpha^{\max\{l,k\}}\, \varepsilon^{(0)}$$

with $0 < \alpha < 1$. For 2D-problems $\varepsilon^{(0)} = \alpha = 0.2$ turned out to be a good choice. The optimal value for α for 3D problems is smaller.

ALGORITHM 6.5.1 *The nested grids ILU decomposition (NGILU) M is defined by*

$$M = L\, U,$$

where L and U are constructed as described above.

6.6 Numerical Experiments

6.6.1 Incomplete Factorization Multigraph Algorithm

In [Bank and Smith 1998], the number of iteration steps for a 10^{-6} reduction of the residual are compared for the incomplete factorization multigraph algorithm and the algebraic multigrid method in [Ruge and Stüben 1987] (see Section 4). The incomplete factorization multigraph algorithm is applied as preconditioner in an composite step biconjugate gradient procedure. The following problems are considered.

Experiment 6.6.1

$$-\Delta u = 1 \quad \textit{in } \mathcal{D},$$

with Dirichlet boundary conditions where $\mathcal{D}$ is a domain in the shape of Lake Superior. The results are shown in Table 6.1.

unknowns	AMG steps	AMG time/s	one parent steps	one parent time/s	two parents steps	two parents time/s
$5 \cdot 10^3$	8	0.17	11	0.11	9	0.10
$20 \cdot 10^3$	10	1.33	16	0.79	12	0.70
$80 \cdot 10^3$	12	8.07	27	6.14	15	4.10

Table 6.1: Results for Experiment 6.6.1.

For the 2D Poisson equation on an uniform grid, the number of iteration steps is proportional to the logarithm of number of unknowns.

Experiment 6.6.2 *This experiment features discontinuous and anisotropic coefficients. A simple 2D-domain is divided into three regions $\Omega_1, \Omega_2, \Omega_3$ where the differential equations*

$$\begin{aligned} -\epsilon\,\Delta u &= 0 \quad \textit{in } \Omega_1, \\ -\Delta u &= 1 \quad \textit{in } \Omega_2, \\ -\frac{\partial^2 u}{\partial x^2} - \epsilon\,\frac{\partial^2 u}{\partial y^2} &= 1 \quad \textit{in } \Omega_3 \end{aligned}$$

are discretized with Dirichlet and Neumann boundary conditions. See Table 6.2 for the results.

Experiment 6.6.3

$$-\nabla\cdot(\nabla u + \beta\, u) = 0 \quad \textit{and} \quad -\nabla\cdot(\nabla u - \beta\, u) = 0$$

with Dirichlet and Neumann boundary conditions. $|\beta| \approx 10^4$ inside a narrow curved band and $\beta = 0$ in the rest of the domain. The problem is taken from semiconductor device modeling. The results for both equations are shown in Table 6.3.

unknowns	AMG steps	AMG time/s	one parent steps	one parent time/s	two parents steps	two parents time/s
$5 \cdot 10^3$	49	1.24	35	0.38	14	0.17
$20 \cdot 10^3$	48	6.76	53	2.74	19	1.15
$80 \cdot 10^3$	32	22.1	128	28.9	26	7.29

Table 6.2: Results for Experiment 6.6.2.

unknowns	AMG steps	AMG time/s	one parent steps	one parent time/s	two parents steps	two parents time/s
$5 \cdot 10^3$	div		93	1.69	70	1.58
$20 \cdot 10^3$	49	7.45	96	9.28	64	6.99
$80 \cdot 10^3$	11	8.11	119	51.10	68	34.00
$5 \cdot 10^3$	div		10	0.2	7	0.18
$20 \cdot 10^3$	div		10	1.01	7	0.88
$80 \cdot 10^3$	13	9.62	12	5.54	8	4.48

Table 6.3: Results for Experiment 6.6.3.

6.6.2 Multilevel ILU Decomposition

In this section, the performance of the MLILU algorithm is analyzed (see [Bank and Wagner 1999]). For all experiments, V-cycle with one post-smoothing Gauß-Seidel step is applied ($\gamma = 1$, $\nu_1 = 0$, $\nu_2 = 1$). As test vector, $t = (1, 1, \ldots, 1)^T$ is used.

The average convergence rates of the first s steps

$$k_s = \left(\frac{\|f - A\,u^{(s)}\|_2}{\|f - A\,u^{(0)}\|_2} \right)^{1/s}, \quad k_s^s < 10^{-10},$$

necessary for a ten orders of magnitude reduction of the residual are reported.

Experiment 6.6.4

$$\Delta\,\phi = 0 \quad in\ \mathcal{D},$$

where $\mathcal{D}$ is a complex domain (drift chamber). The boundary conditions are of Neumann and Dirichlet type. A detailed description of this problem can be found in [Bastian 1996] and [Wagner 1995]. The coarsest triangulation consists of 112 triangles, some with very small angles. Table 6.4 shows convergence rates for several grids, which are obtained from the regular refinement of the coarsest triangulation ($\sigma = 1/2$).

unknowns	convergence
890	$k_{15} = 0.210$
3578	$k_{21} = 0.322$
14330	$k_{23} = 0.360$
14330 ($\gamma = 2$)	$k_{11} = 0.114$

Table 6.4: Results for Experiment 6.6.4.

A similar, small dependence of the convergence rates on the number of unknowns for the fine grids can be observed for the standard multigrid method, although the convergence rates are much worse for standard multigrid. The convergence rates for the Poisson equation on an uniform grid are independent of the number of unknowns.

Experiment 6.6.5 *The results for the anisotropic differential equation*

$$\begin{aligned} \epsilon_x \frac{\partial^2 u}{\partial x^2} + \epsilon_y \frac{\partial^2 u}{\partial y^2} = 0 \quad & \textit{in } \Omega = (0,1) \times (0,1), \\ u(x,y) = \frac{x+y}{2} \quad & \textit{on } \partial\Omega, \end{aligned}$$

where

$$\begin{array}{lll} \epsilon_x = \epsilon, & \epsilon_y = 1, & x < 1/2,\ y < 1/2, \\ \epsilon_x = 1, & \epsilon_y = \epsilon, & x \geq 1/2,\ y < 1/2, \\ \epsilon_x = 1, & \epsilon_y = \epsilon, & x < 1/2,\ y \geq 1/2, \\ \epsilon_x = \epsilon, & \epsilon_y = 1, & x \geq 1/2,\ y \geq 1/2, \end{array}$$

which is discretized with linear basis functions on an uniform grid ($h = 1/128$, $\sigma = 1/2$) are presented in Table 6.5. Figure 6.5 shows a typical graph of a coarse grid matrix for small values of ϵ indicating a kind of semi-coarsening..

ϵ	convergence
1	$k_{11} = 0.109$
10^{-2}	$k_{15} = 0.207$
10^{-4}	$k_{19} = 0.288$
10^{-6}	$k_{19} = 0.288$

Table 6.5: Results for Experiment 6.6.5.

Experiment 6.6.6 *A typical interface model problem is given by*

$$\nabla \cdot (D\ \nabla u) \quad = \quad 0 \quad \textit{in } \Omega = (0,1) \times (0,1),$$

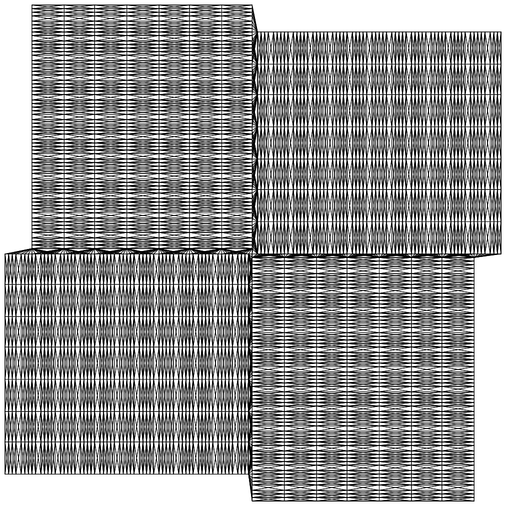

Figure 6.5: Coarse grid for Experiment 6.6.5.

$$
\begin{aligned}
D &= \begin{cases} 1 & : \quad \delta < y < 1-\delta \ \wedge \ \delta < x < 1-\delta, \\ \epsilon & : \quad \textit{otherwise}, \end{cases} \\
u(x,y) &= \frac{x+y}{2} \quad \textit{on } \partial\Omega.
\end{aligned}
$$

The convergence results for several values of ϵ and δ can be found in Table 6.6. The differential equation was discretized with bilinear basis functions on an uniform grid ($h = 1/128$, $\sigma = 1/4$). The graph of a coarse grid matrix for $\epsilon = 10^{-6}$ and $\delta = 33/128$ is presented in Figure 6.6.

ϵ	conv. rate, $\delta = 1/4$	conv. rate, $\delta = 33/128$
10^6	$k_9 = 0.072$	$k_{16} = 0.24$
10^4	$k_9 = 0.072$	$k_{16} = 0.24$
10^2	$k_9 = 0.072$	$k_{16} = 0.24$
10^0	$k_8 = 0.051$	$k_8 = 0.051$
10^{-2}	$k_{10} = 0.095$	$k_{18} = 0.27$
10^{-4}	$k_{10} = 0.097$	$k_{18} = 0.27$
10^{-6}	$k_{10} = 0.099$	$k_{18} = 0.27$

Table 6.6: Results for Experiment 6.6.6.

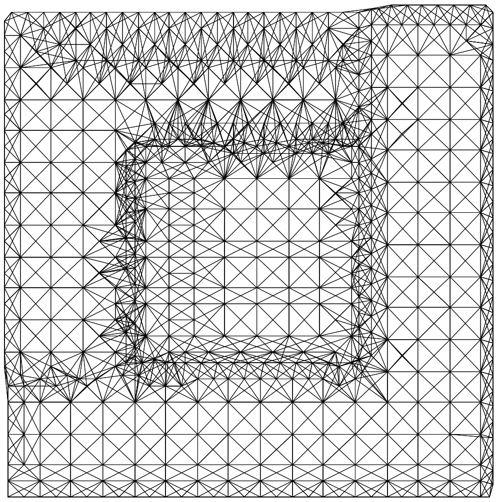

Figure 6.6: Coarse grid for Experiment 6.6.6.

EXPERIMENT 6.6.7 *The convection term in the differential equation*

$$\epsilon \Delta u - \sin(\pi x)\cos(\pi y)\tfrac{\partial u}{\partial x} + \sin(\pi y)\cos(\pi x)\tfrac{\partial u}{\partial y} = 0 \quad in\ \Omega = (0,1)\times(0,1),$$
$$u(x,y) = \sin(\pi x) + \sin(13\pi x) + \sin(\pi y) + \sin(13\pi y) \quad on\ \partial\Omega$$

simulates a rotating flow (see Figure 4.5). Convergence rates and a typical graph of a coarse grid matrix are shown in Table 6.7 and Figure 6.7 respectively. (uniform grid, bilinear basis functions, $h = 1/128$, $\sigma = 1/2$).

ϵ	convergence
1	$k_9 = 0.066$
10^{-2}	$k_{18} = 0.275$
10^{-4}	$k_{22} = 0.342$
10^{-6}	$k_{21} = 0.333$

Table 6.7: Results for Experiment 6.6.7.

6.6.3 Nested Grids ILU Decomposition

The performance of the NGILU decomposition applied as preconditioner in a conjugate gradient algorithm is considered in this section. In [van der Ploeg, Botta, and Wubs 1996], the number of steps s until the stopping criterion

$$\|L^{-1}(f - A\,u^{(s)})\|_2 \le 10^{-6}\|L^{-1}(f - A\,u^{(0)})\|_2$$

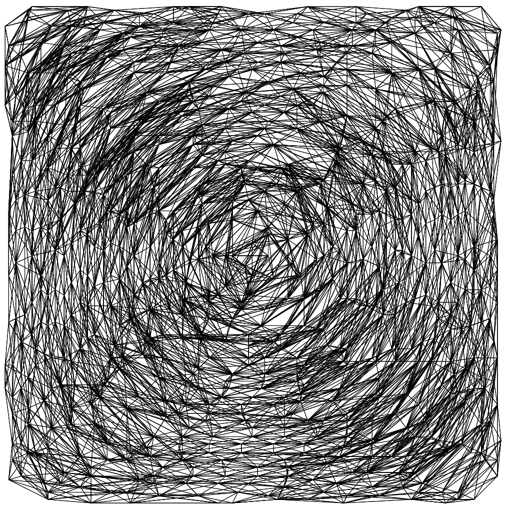

Figure 6.7: Coarse grid for Experiment 6.6.7.

is reached are reported.

EXPERIMENT 6.6.8 *The Poisson equation*

$$\Delta u = 1 \quad \textit{in } (0,1) \times (0,1),$$

with Dirichlet and Neumann boundary conditions was discretized on an uniform and on stretched $m \times m$ grids. The results for the uniform mesh can be found in Table 6.8, the results for the stretched grids are shown in Table 6.9.

m	steps	non-zero entries/m^2	flops/m^2
64	7	3.6	204
128	7	4.0	217
256	7	4.3	226
512	7	4.5	233
1024	7	4.6	236

Table 6.8: Results for Experiment 6.6.8.

$\frac{h_{\max}}{h_{\min}}$	$m = 64$ steps	$m = 64$ entries/m^2	$m = 128$ steps	$m = 128$ entries/m^2	$m = 256$ steps	$m = 256$ entries/m^2
10^2	5	11.5	7	11.2	9	11.2
10^4	4	15.6	5	22.0	9	25.1
10^6	3	15.0	5	22.4	6	31.0

Table 6.9: Results for Experiment 6.6.8 on stretched grids.

7 Block-Elimination Methods

The basic idea of the block-elimination algebraic multigrid methods, is to construct a factorization of the system matrix A of the form

$$\begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} = \begin{pmatrix} I_{n_F} & 0 \\ A_{CF}\,A_{FF}^{-1} & I_{n_C} \end{pmatrix} \begin{pmatrix} A_{FF} & 0 \\ 0 & A_S \end{pmatrix} \begin{pmatrix} I_{n_F} & A_{FF}^{-1}\,A_{FC} \\ 0 & I_{n_C} \end{pmatrix}$$

In general, the Schur-complement $A_S = A_{CC} - A_{CF}\,A_{FF}^{-1}\,A_{FC}$ is a dense matrix. Thus, it must be approximated.

Nevertheless, a tridiagonal matrix A can be reordered, such that the blocks A_{FF} and A_{CC} are diagonal and the Schur-complement A_S is again tridiagonal. Hence, in the same way, A_S can be decomposed. This method is called cyclic reduction and can be used, for example, for solving a linear system with a tridiagonal matrix or with a special block-tridiagonal matrix.

Several methods based on this block-elimination idea have been introduced e.g. in [Axelsson and Vassilevski 1989; Axelsson and Vassilevski 1990; Fuhrmann 1994; Fuhrmann 1995; Notay 1997; Notay 1998; Reusken 1997; Reusken 1998].

Classic line smoothers or line preconditioners like ILLU (see e.g. [Kettler 1981]) or the frequency filtering decomposition [Wittum 1992; Wagner 1997a; Wagner 1997b] are based on a similar partitioning of the system matrix into more blocks corresponding to grid lines.

7.1 A Semi-Algebraic Multigrid Preconditioner

The preconditioners introduced and analyzed in [Axelsson and Vassilevski 1989; Axelsson and Vassilevski 1990] are based on block-elimination, although a given finite element grid hierarchy is used and the coarse grid matrices are computed by standard finite element discretization on these grids.

ALGORITHM 7.1.1 *Let a hierarchy of nested grids* $\Omega_0 \supset \Omega_1 \supset \cdots \supset \Omega_{l_{\max}}$ *and system matrices* A_l *on these levels be given. Then, the proconditioner* $M = M^{(0)}$ *in [Axelsson and Vassilevski 1990] is recursively defined by*

$$M_l = \begin{pmatrix} B_{l,FF} & 0 \\ \tilde{A}_{l,CF} & A_{l,C} \end{pmatrix} \begin{pmatrix} I_{n_{l,F}} & -B_{l,FF}^{-1} \tilde{A}_{l,FC} \\ 0 & I_{n_{l,C}} \end{pmatrix}, \tag{7.1.1}$$

where

$$\tilde{A}_{l,FC} = A_{l,FC} + (A_{l,FF} - B_{l,FF})\, J_{l,FC}, \quad \tilde{A}_{l,CF} = A_{l,CF} + J_{l,FC}^T (A_{l,FF} - B_{l,FF}),$$

$J_{l,FC}$ *is the FC-block from the hierarchical basis transformation (2.7.1) in Section 2.7,* $B_{l,FF}$ *is an approximation for* $A_{l,FF}$*, and* $A_{l,C}$ *is implicitly defined by*

$$\begin{aligned} A_{l_{\max}-1,C} &= M_{l_{\max}} = A_{l_{\max}} \\ (A_{l,C})^{-1} &= [I - P_\gamma(M_{l+1}^{-1} A_{l+1})]\, A_{l+1}^{-1}, \quad l < l_{\max} - 1. \end{aligned}$$

P_γ *is a polynomial of degree* $\gamma \geq 1$ *and satisfies*

$$0 \leq P_\gamma(t) < 1 \quad \forall\, t \in (0,1], \quad P_\gamma(0) = 1.$$

Possible choices for the matrix $B_{l,FF}$ are either $B_{l,FF} = A_{l,FF}$ which leads to the method analyzed in [Axelsson and Vassilevski 1989] or a maybe scaled incomplete (block-)factorization of $A_{l,FF}$. For the theoretical analysis, $B_{l,FF}$ is supposed to fulfill

$$1 \leq \frac{(v, B_{l,FF}\, v)_2}{(v, A_{l,FF}\, v)_2} \leq 1 + b \tag{7.1.2}$$

with a constant b.

Examples for the polynomials $P_\gamma(t)$ are

$$P_\gamma(t) = (1-t)^\gamma \tag{7.1.3}$$

or

$$P_\gamma(t) = \frac{T_\gamma\left(\dfrac{1+\alpha-2t}{1-\alpha}\right) + 1}{T_\gamma\left(\dfrac{1+\alpha}{1-\alpha}\right) + 1}, \quad 0 \leq \alpha \leq 1, \tag{7.1.4}$$

where $T_\gamma(x)$ is the Chebyshev polynomial

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_s(x) = 2\,x\,T_{s-1}(x) - T_{s-2}(x), \quad s > 1.$$

For $\alpha \to 1$, the polynomials based on the Chebyshev polynomials converge to the polynomials in (7.1.3). For an optimal value of α, good estimates of b in (7.1.2) and the constant ξ in the strengthened Cauchy-Schwarz inequality (2.7.2) are required.

(7.1.1) can be rewritten as

$$M_l^{-1} =$$
$$\begin{pmatrix} I_{n_{l,F}} & -B_{l,FF}^{-1}\tilde{A}_{l,FC} \\ 0 & I_{n_{l,C}} \end{pmatrix} \begin{pmatrix} B_{l,FF}^{-1} & 0 \\ 0 & A_{l,C}^{-1} \end{pmatrix} \begin{pmatrix} I_{n_{l,F}} & 0 \\ -\tilde{A}_{l,CF}B_{l,FF}^{-1} & I_{n_{l,C}} \end{pmatrix}$$
$$= \begin{pmatrix} B_{l,FF}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -B_{l,FF}^{-1}\tilde{A}_{l,FC} \\ I_{n_{l,C}} \end{pmatrix} A_{l,C}^{-1} \begin{pmatrix} -\tilde{A}_{l,CF}B_{l,FF}^{-1} & I_{n_{l,C}} \end{pmatrix}$$

Hence, the structure of Algorithm 7.1.1 resembles the structure of the classical multigrid method, where γ coarse grid correction steps are computed on every grid level. Note that in Algorithm 7.1.1, $P_1(t) = (1-t)$ and $P_2(t) = (1-t)^2$ yield a V-cycle and W-cycle respectively. Furthermore, the multiplication with $-\tilde{A}_{l,CF}B_{l,FF}^{-1}$, which occurs in the forward substitution, can be seen as a restriction. The multiplication with $-B_{l,FF}^{-1}\tilde{A}_{l,FC}$ during the backward substitution can be seen as a prolongation and $B_{l,FF}^{-1}$ is a smoothing step for the fine grid unknowns only.

The difference with the classical multigrid method is that only the fine grid nodes are modified by the smoother. In this respect, the method is similar to the hierarchical basis multigrid method. It is well known, that the hierarchical basis multigrid method does not converge independent of the mesh size especially in three spatial dimensions. Therefore, we can not expect mesh size independent convergence for $P_1(t) = (1-t)$. However, as stated in the following theorem, mesh size independent convergence can be obtained for γ large enough.

THEOREM 7.1.1 *Let A_l be symmetric and positive definite and assume that the strengthened Cauchy-Schwarz inequality (2.7.2) holds with $\xi < 1$. The preconditioner M_0 defined in Algorithm 7.1.1 with P_γ in (7.1.4) and B_{FF} satisfying (7.1.2) is spectrally equivalent to A_0 if $\gamma > (1-\xi^2)^{-1/2}$.*

PROOF. The proof is rather technical. We therefore refer to [Axelsson and Vassilevski 1990]. □

7.2 The Approximated Cyclic Reduction Preconditioner

The approximated cyclic reduction preconditioner introduced in [Reusken 1997; Reusken 1998] is based on the cyclic reduction/block-elimination idea. The coarse grids are constructed using a maximal independent set labeling technique. The coarse grid matrices are determined by an approximation of the Schur-complements resulting from the block-elimination and can be computed by a typical Galerkin approximation.

7.2.1 F/C-Partitioning

In this section, the F/C-labeling strategy proposed in [Reusken 1997; Reusken 1998] is discussed. As we have seen in Section 4, if simple (point) smoother are used, to enhance robustness, coarsening should only occur in the direction of strong connections. Let $G_A(V, E)$ denote the graph of the matrix A where V is the set of vertices (unknowns) and E is the set of edges (non-zero matrix entries). Every diagonal connection $e_{i,i} \in E$ is labeled strong. For every non-isolated vertex $v_i \in V$, an edge $e_{i,j} \in E$ is labeled strong if

$$|a_{i,j}| \geq \beta \max_{j \neq i} |a_{i,j}|$$

with a given parameter $0 \leq \beta < 1$. An edge is labeled weak if it is not strong. This yields a splitting $E = E_S \cup E_W$ of the edges into strong E_S and weak E_W edges. The directed graph consisting of the vertices V and the set of strong edges E_S is called the reduced graph and is denoted by $G_A(V, E_S)$.

The next step is the construction of a maximal independent set F of the reduced graph $G_A(V, E_S)$. Several different techniques for the construction of a maximal independent set are known [Saad 1996a]. Two vertices v_i and v_j are said to be independent with respect to E_S if $e_{i,j} \notin E_S$ and $e_{j,i} \notin E_S$. A subset V_M of V is called independent set if every two vertices in V_M are independent. V_M is a maximal independent set if no proper superset of V_M is independent.

The algorithm proposed in [Reusken 1997; Reusken 1998] consists of a graph traversal and a labeling method. For the graph traversal, the breadth first search (BFS) [Horowitz and Sahni 1984] is applied. The BFS algorithm starts with a vertex $v_0 \in V$ and marks it as visited. Unvisited vertices adjacent from v_0 are visited next and so on. This approach is applied to every connected component of $G_A(V, E_S)$. The labeling scheme is summarized in the following algorithm.

ALGORITHM 7.2.1 *Let a reduced graph $G_A(V, E_S)$ be given and N_i^S be defined by*

$$N_i^S = \{\, v_j \in V \,|\, |a_{i,j}| \geq \beta \max_{j \neq i} |a_{i,j}| \text{ and } i \neq j \,\}.$$

Then, the set F of vertices labeled F by `BFS_Label()` *is a maximal independent set with respect to $G_A(V, E_S)$. If $G_A(V, E_S)$ can be split into several unconnected sub-graphs,* `BFS_Label()` *is applied to all these sub-graphs.*

```
BFS_Label()
{
    FIFO_in(v_0);
    F = {}; C = {};
    while (FIFO_not_empty())
    {
        v_i = FIFO_out();
        if (v_i is not labeled)
        {
            if (v_j ∈ N_i^S ∩ F exists)
```

```
            {
                label(v_i) = C;
                C = C ∪ v_i ;
            }
            else
            {
                label(v_i) = F;
                F = F ∪ v_i ;
                for (v_j ∈ N_i^S) { label(v_j) = C; C = C ∪ v_j ; }
            }
        }
        for (v_j ∈ N_i^S && v_j not visited ) FIFO_in(v_j);
    }
}
```

Note that for many related coarsening techniques with maximal independent sets (e.g. [Reusken 1996; Ruge and Stüben 1987]) the coarse grid nodes V_C build an independent set.

The F/C-partitioning induces a

$$A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix}$$

partitioning of the system matrix. Let $\hat{\beta}$ be defined by

$$\hat{\beta} = \begin{cases} 0 & : \quad E_W = \{\}, \\ \sup\limits_{e_{i,j} \in E_W} \frac{|a_{i,j}|}{\max\limits_j |a_{i,j}|} & : \quad \text{otherwise.} \end{cases}$$

$deg(v_i) = |N_i|$ is the number of neighbor vertices $N_i = \{\, v_j \in V \mid e_{i,j} \in E \text{ and } i \neq j \,\}$. The maximum degree of the graph G_A is denoted by $\Delta(G_A) = \max\limits_{v_i \in V} deg(v_i)$.

LEMMA 7.2.1 *Consider the F/C-partitioning as described above and assume that the system matrix A is a weakly diagonally dominant M-matrix. Then, the following holds:*

$$\| I_{n_F} - D_{FF}^{-1} A_{FF} \|_\infty \leq \frac{(\Delta(G_A) - 1)\, \hat{\beta}}{(\Delta(G_A) - 1)\, \hat{\beta} + 1},$$

where $D_{FF} = \operatorname{diag}(A_{FF})$.

PROOF. Since A is an M-matrix,

$$\| I_{n_F} - D_{FF}^{-1} A_{FF} \|_\infty = \| (I_{n_F} - D_{FF}^{-1} A_{FF})\, 1_{n_F} \|_\infty,$$

where $1_{n_F} \in \mathbb{R}^{n_F}$ is given by $1_{n_F} = (1, \ldots, 1)^T$. Then, with $W_k = N_k \cap F$ for $v_k \in F$ and $deg(v_k) > 0$

$$\sum_{v_j \in W_k} \frac{|a_{k,j}|}{a_{k,k}} \le \frac{\sum_{v_j \in W_k} |a_{k,j}|}{\sum_{v_j \in N_K} |a_{k,j}|} = \frac{\sum_{v_j \in W_k} |a_{k,j}|}{\sum_{v_j \in W_k} |a_{k,j}| + \sum_{v_j \in N_k \cap C} |a_{k,j}|}. \tag{7.2.1}$$

Because $deg(v_k) > 0$, there is at least one strong edge $e_{k,i}$, $v_i \in C$, and thus $|W_k| \le \Delta(G_A) - 1$. Using the notation $m_k = \max_{j \neq k} |a_{k,j}|$, we get

$$\sum_{v_j \in W_k} |a_{k,j}| \le (\Delta(G_A) - 1)\, \hat{\beta}\, m_k.$$

Combining this with (7.2.1) yields

$$\sum_{v_j \in W_k} \frac{|a_{k,j}|}{a_{k,k}} \le \frac{(\Delta(G_A) - 1)\, \hat{\beta}\, m_k}{(\Delta(G_A) - 1)\, \hat{\beta}\, m_k + m_k}. \tag{7.2.2}$$

Since

$$\sum_{v_j \in W_k} \frac{|a_{k,j}|}{a_{k,k}} = 0 \quad \text{for } deg(v_k) = 0,$$

(7.2.2) proofs the lemma. □

7.2.2 Schur-Complement Approximation

The Schur-complement $A_S = A_{CC} - A_{CF}\, A_{FF}^{-1}\, A_{FC}$ is the result of a block Gaussian elimination. Since A_S is in general a dense matrix, A_S has to be approximated. Note that any left transformation of the form

$$\begin{pmatrix} I_{n_F} & 0 \\ -B_{CF} & I_{n_C} \end{pmatrix} \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} - B_{CF}\, A_{FF} & A_{CC} - B_{CF}\, A_{FC} \end{pmatrix}$$

does not change the Schur-complement. We consider a sequence of such left transformations with matrices $B_{CF} \in \mathbb{R}^{n_C \times n_F}$ of the form $B_{CF} = A_{CF}\, D_{FF}^{-1}$ ($D_{FF} = \operatorname{diag}(A_{FF})$ or $B_{CF} = A_{CF}\, \tilde{D}_{FF}^{-1}$. The matrix $\tilde{D}_{FF}$ is the diagonal matrix which satisfies

$$A_{FF}\, 1_{n_F} = \tilde{D}_{FF}\, 1_{n_F}. \tag{7.2.3}$$

In particular, for $k \in \mathbb{N}$ the sequences $A^{(k)}$ and $\tilde{A}^{(k)}$ are recursively defined by

$$A^{(0)} = A, \tag{7.2.4}$$

$$A^{(k)} = \begin{pmatrix} I_{n_F} & 0 \\ -A_{CF}\, D_{FF}^{-1} & I_{n_C} \end{pmatrix} A^{(k-1)}, \quad k > 0, \tag{7.2.5}$$

$$\tilde{A}^{(k)} = \begin{pmatrix} I_{n_F} & 0 \\ -A_{CF}\, \tilde{D}_{FF}^{-1} & I_{n_C} \end{pmatrix} A^{(k-1)}, \quad k > 0. \tag{7.2.6}$$

The Schur-complements of $A^{(k)}$ and $\tilde{A}^{(k)}$ are equal for all k. The definitions in (7.2.5) and (7.2.6) yield

$$
\begin{aligned}
A_{CF}^{(k)} &= A_{CF}\,(I_{n_F} - D_{FF}^{-1}\,A_{FF})^k, && (7.2.7)\\
\tilde{A}_{CF}^{(k)} &= A_{CF}\,(I_{n_F} - D_{FF}^{-1}\,A_{FF})^{k-1}(I_{n_F} - \tilde{D}_{FF}^{-1}\,A_{FF}). && (7.2.8)
\end{aligned}
$$

This implies with Lemma 7.2.1 that the blocks $A_{CF}^{(k)}$ and $\tilde{A}_{CF}^{(k)}$ are "small" for k sufficiently large and that, for k large enough, the diagonal blocks $A_{CC}^{(k)}$ and $\tilde{A}_{CC}^{(k)}$ might be reasonable approximations of A_S. We will use $\tilde{A}_{CC}^{(2)}$ for our preconditioner. For the analysis in the remainder of the section, we consider arbitrary $k > 0$.

Lemma 7.2.2 *Let A be a weakly diagonally dominant M-matrix with a given F/C-partitioning as described in Section 7.2.1. Then,*

$A_{CC}^{(k)}$ *is a weakly diagonally dominant M-matrix for all k,*

$A_S = A_{CC}^{(k)} - N^{(k)}$ *is a regular splitting for all k (see Definition 1.3.1),*

$\lim\limits_{k\to\infty} A_{CC}^{(k)} = A_S.$

Proof. A_S is a weakly diagonally dominant M-matrix and obviously $A_{CF} \le 0$ and $A_{FC} \le 0$ (componentwise inequalities). Due to Lemma 7.2.1,

$$\rho(\Delta_{FF}) < 1 \quad \text{for} \quad \Delta_{FF} = I_{n_F} - D_{FF}^{-1}\,A_{FF} \ge 0.$$

Hence, we have the representation

$$A_S = A_{CC} - A_{CF}\left(\sum_{j=0}^{\infty} \Delta_{FF}^j\, D_{FF}^{-1}\right) A_{FC}.$$

Using (7.2.7), we obtain by induction

$$A_{CC}^{(k)} = A_{CC} - A_{CF}\left(\sum_{j=0}^{k-1} \Delta_{FF}^j\, D_{FF}^{-1}\right) A_{FC}.$$

Thus,

$$N^{(k)} = A_{CC}^{(k)} - A_S = A_{CF}\left(\sum_{j=k}^{\infty} \Delta_{FF}^j\, D_{FF}^{-1}\right) A_{FC} \ge 0. \qquad (7.2.9)$$

All off-diagonal entries of $A_{CC}^{(k)}$ are non-positive because all off-diagonal entries of A_{CC} are non-positive and $\Delta_{FF}^j\, D_{FF}^{-1} \ge 0$ for all j. We conclude that $A_{CC}^{(k)}$ is an M-matrix and the splitting $A_S = A_{CC}^{(k)} - N^{(k)}$ is regular. $A_{CC}^{(k)} \ge A_S$ and $A_S\, 1_{n_C} \ge 0$ show that $A_{CC}^{(k)}$ is weakly diagonally dominant. Finally, $\lim\limits_{k\to\infty} N^{(k)} = 0$ leads to $\lim\limits_{k\to\infty} A_{CC}^{(k)} = A_S$. $\square$

According to Lemma 7.2.2 and Lemma 1.3.2 $A_{CC}^{(k)}$ is a stable approximation of A_S and

$$\rho(I_{n_C} - (A_{CC}^{(k)})^{-1} A_S) < 1$$

(see e.g. [Varga 1962]). However, the approximation of A_S is very poor for smooth vectors. Therefore, the resulting preconditioner for reasonable small k behaves like a smoother. To improve the approximation quality for smooth vectors, the block-elimination is constructed with the matrix $\tilde{D}_{FF}$ which satisfies the filter condition (7.2.3).

LEMMA 7.2.3 *With the same assumptions as in Lemma 7.2.2, the following holds for* $\tilde{A}_{CC}^{(k)}$ *in (7.2.6).*

(a) $\tilde{A}_{CC}^{(k)}$ *has only non-positive off-diagonal entries for all* k,

(b) $\tilde{A}_{CC}^{(k)}$ *is weakly diagonally dominant for all* k,

(c) if $\tilde{A}_{CC}^{(k)}$ *is regular then it is an M-matrix,*

(d) $\tilde{A}_{CC}^{(k)}$ *is an M-matrix for* k *sufficiently large,*

(e) $\lim\limits_{k\to\infty} \tilde{A}_{CC}^{(k)} = A_S$,

(f) if $A_{FC}\, w = -A_{FF}\, 1_{N_F}$ *then* $\tilde{A}_{CC}^{(k)}\, w = A_S\, w$ *for all* k,

(g) $I_{n_C} - \tilde{A}_{CC}^{(k)} A_S^{-1} = A_{CF}(I_{n_F} - D_{FF}^{-1} A_{FF})^{k-1} \tilde{D}_{FF}^{-1} (\tilde{D}_{FF} \quad A_{FC}) A^{-1} \begin{pmatrix} 0 \\ I_{n_C} \end{pmatrix}$.

PROOF. By definition

$$\tilde{A}_{CC}^{(k)} = A_{CC}^{(k-1)} - A_{CF}^{(k-1)}\, \tilde{D}_{FF}^{-1}\, A_{FC}. \tag{7.2.10}$$

Because $A_{CC}^{(k-1)}$ is an M-matrix (Lemma 7.2.2), $A_{CF}^{(k-1)} \leq 0$ (see (7.2.7)), $\tilde{D}_{FF}^{-1} > 0$, and $A_{FC} \leq 0$, the result (a) holds. (7.2.5) leads to

$$A_{CC}^{(k)}\, 1_{n_C} + A_{CF}^{(k)}\, 1_{n_F} = A_{CC}^{(k-1)}\, 1_{n_C} + A_{CF}^{(k-1)}\, 1_{n_F} - A_{CF}^{(k-1)} D_{FF}^{-1} (A_{FF}\, 1_{n_F} + A_{FC}\, 1_{n_C}) \geq 0.$$

The inequality follows by induction from

$$A_{CC}\, 1_{n_C} + A_{CF}\, 1_{n_F} \geq 0, \quad A_{FF}\, 1_{n_F} + A_{FC}\, 1_{n_C} \geq 0.$$

The filter condition (7.2.3) shows

$$\tilde{D}_{FF}\, 1_{n_F} + A_{FC}\, 1_{n_C} = A_{FF}\, 1_{n_F} + A_{FC}\, 1_{n_C} \geq 0 \quad \text{and} \quad \tilde{D}_{FF}^{-1} A_{FC}\, 1_{n_C} \geq -1_{n_F}.$$

This yields with (7.2.10) the result in (b)

$$\tilde{A}_{CC}^{(k)}\, 1_{n_C} = A_{CC}^{(k-1)}\, 1_{n_C} - A_{CF}^{(k-1)}\, \tilde{D}_{FF}^{-1}\, A_{FC}\, 1_{n_C} \geq A_{CC}^{(k-1)}\, 1_{n_C} + A_{CF}^{(k-1)}\, 1_{n_F} \geq 0.$$

Since $\tilde{A}_{CC}^{(k)}$ is weakly diagonally dominant and the off-diagonal entries are non-positive, we conclude from Gerschgorin's circle theorem that every real eigenvalue of $\tilde{A}_{CC}^{(k)}$ is non-negative. Therefore, if $\tilde{A}_{CC}^{(k)}$ is regular, $\tilde{A}_{CC}^{(k)}$ is an M-matrix [Berman and Plemmons 1979].

Lemma 7.2.2 and (g) induce $\rho(I_{n_C} - \tilde{A}_{CC}^{(k)} A_S^{-1}) < 1$ for k sufficiently large and, hence, $\tilde{A}_{CC}^{(k)}$ is regular for k sufficiently large.

We obtain from (7.2.7), (7.2.9), and (7.2.10)

$$
\begin{aligned}
\tilde{A}_{CC}^{(k)} - A_S &= A_{CC}^{(k-1)} - A_S - A_{CF}^{(k-1)} \tilde{D}_{FF}^{-1} A_{FC} \\
&= A_{CF} \left(\sum_{j=k-1}^{\infty} \Delta_{FF}^{j} D_{FF}^{-1} \right) A_{FC} - A_{CF}^{(k-1)} \tilde{D}_{FF}^{-1} A_{FC} \\
&= A_{CF}(I_{n_F} - D_{FF}^{-1} A_{FF})^{k-1} A_{FF}^{-1} A_{FC} - A_{CF}^{(k-1)} \tilde{D}_{FF}^{-1} A_{FC} \\
&= A_{CF}(I_{n_F} - D_{FF}^{-1} A_{FF})^{k-1} (I_{n_F} - \tilde{D}_{FF}^{-1} A_{FF}) A_{FF}^{-1} A_{FC} \qquad (7.2.11)
\end{aligned}
$$

which proofs (e). For w in (f)

$$
(I_{n_F} - \tilde{D}_{FF}^{-1} A_{FF}) A_{FF}^{-1} A_{FC}\, w = (I_{n_F} - \tilde{D}_{FF}^{-1} A_{FF})\, 1_{n_F} = 0
$$

holds. Thus, (7.2.11) shows (f).

From the identity

$$
A^{-1} = \begin{pmatrix} I_{n_F} & -A_{FF}^{-1} A_{FC} \\ 0 & I_{n_C} \end{pmatrix} \begin{pmatrix} A_{FF}^{-1} & 0 \\ 0 & A_S^{-1} \end{pmatrix} \begin{pmatrix} I_{n_F} & 0 \\ -A_{CF} A_{FF}^{-1} & I_{n_C} \end{pmatrix},
$$

we get

$$
(I_{n_F} \;\; 0)\; A^{-1} \begin{pmatrix} 0 \\ I_{n_C} \end{pmatrix} = -A_{FF}^{-1} A_{FC} A_S^{-1}
$$

and

$$
\begin{aligned}
-(\tilde{D}_{FF} - A_{FF}) A_{FF}^{-1} A_{FC} A_S^{-1} &= (\tilde{D}_{FF} - A_{FF})\; (I_{n_F} \;\; 0)\; A^{-1} \begin{pmatrix} 0 \\ I_{n_C} \end{pmatrix} \\
&= \begin{pmatrix} \tilde{D}_{FF} & A_{FC} \end{pmatrix} A^{-1} \begin{pmatrix} 0 \\ I_{n_C} \end{pmatrix} \\
&\qquad - (I_{n_F} \;\; 0)\; A\, A^{-1} \begin{pmatrix} 0 \\ I_{n_C} \end{pmatrix} \\
&= \begin{pmatrix} \tilde{D}_{FF} & A_{FC} \end{pmatrix} A^{-1} \begin{pmatrix} 0 \\ I_{n_C} \end{pmatrix}.
\end{aligned}
$$

Finally, the multiplication of (7.2.11) with $-A_S^{-1}$

$$
I_{n_C} - \tilde{A}_{CC}^{(k)} A_S^{-1} = -A_{CF}(I_{n_F} - D_{FF}^{-1} A_{FF})^{k-1} (I_{n_F} - \tilde{D}_{FF}^{-1} A_{FF}) A_{FF}^{-1} A_{FC} A_S^{-1}
$$

proofs (g). $\square$

REMARK 7.2.1 *$\tilde{A}^{(2)}_{CC}$ can be computed as*

$$\tilde{A}^{(2)}_{CC} = \left(-A_{CF}\, D^{-1}_{FF} \quad I_{n_C}\right)\, A \begin{pmatrix} -\tilde{D}^{-1}_{FF}\, A_{FC} \\ I_{n_C} \end{pmatrix}.$$

Introducing the matrix-dependent prolongation and restriction

$$P = \begin{pmatrix} -\tilde{D}^{-1}_{FF}\, A_{FC} \\ I_{n_C} \end{pmatrix}, \quad R = \left(-A_{CF}\, D^{-1}_{FF} \quad I_{n_C}\right)$$

$\tilde{A}^{(2)}_{CC}$ can be constructed using the Galerkin approach $\tilde{A}^{(2)}_{CC} = R\,A\,P$.

Due to the different approximations of A^{-1}_{FF} in the prolongation P and the restriction R, the coarse grid matrix $\tilde{A}^{(2)}_{CC}$ is not necessarily symmetric if A is symmetric. If D^{-1}_{FF} is used in both P and R, the approximation of A_S is stable but very poor for smooth vectors. On the other hand, if $\tilde{D}^{-1}_{FF}$ is used for P and R then due to instabilities the approximation turned out to be poor for certain problems.

The multigrid convergence theory of Hackbusch [Hackbusch 1985] is based on the approximation and the smoothing property (see Section 2). In [Hackbusch 1985], it is shown that the approximation property is closely related to a regularity property which holds for a certain class of elliptic partial differential equations. Below, in Lemma 7.2.4 a sort of an algebraic regularity term is introduced. The norms in Lemma 7.2.4 for vectors $w_C \in \mathbb{R}^{n_C}$ and $w_F \in \mathbb{R}^{n_F}$ need to be understood in the sense

$$\|w_C\| = \|\begin{pmatrix} 0 \\ w_C \end{pmatrix}\|, \quad \|w_F\| = \|\begin{pmatrix} w_F \\ 0 \end{pmatrix}\|.$$

LEMMA 7.2.4 *With the assumptions of Lemma 7.2.1, the following holds for $\tilde{A}^{(2)}_{CC}$ in (7.2.6):*

$$\|I_{n_C} - \tilde{A}^{(k)}_{CC}\, A^{-1}_S\| \le \|A_{CF}(I_{n_F} - D^{-1}_{FF}A_{FF})^{k-1}\tilde{D}^{-1}_{FF}\|\, \|(\tilde{D}_{FF} \quad A_{FC})\, A^{-1}\|.$$

PROOF. Use (g) in Lemma 7.2.3 and $\|\begin{pmatrix} 0 \\ I_{n_C} \end{pmatrix}\| = 1$. □

The first term on the right hand side in Lemma 7.2.4 can be controlled with linear algebra arguments only. Since $(\tilde{D}_{FF} \quad A_{FC})$ is a difference operator, bounds for A^{-1} as an operator between spaces with different smoothness properties are required in order to bound the second term on the right hand side in Lemma 7.2.4. Such results can be considered as discrete counterparts of results in the regularity theory for continuous elliptic boundary value problems. For a detailed discussion, we refer to [Hackbusch 1985] Section 6.3.2.

Numerical experiments showed that the Schur-complement approximations $\tilde{A}^{(2)}_{CC}$ are in general less sparse than the system matrix A. To avoid too dense coarse grid matrices, the smallest off-diagonal entries in absolute values are added to the diagonal entry (lumping) such that the number of entries per row is bounded by a given constant MSIZE (e.g. MSIZE $= 3\,\Delta(G_A)$).

7.2.3 Model Problems

In this section, we consider the norms

$$\|A_{CF}(I_{n_F} - D_{FF}^{-1}A_{FF})^{k-1}\tilde{D}_{FF}^{-1}\|_2, \quad \|(\tilde{D}_{FF} \;\; A_{FC})\, A^{-1}\|_2$$

for a couple of model problems. These model problems represent discretizations of partial differential equations on the unit square $\Omega = (0,1) \times (0,1)$ with periodic boundary conditions on an uniform grid with mesh size h.

We consider the 9-point stencil

$$A = \begin{bmatrix} -\frac{1}{4} & -\frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{2} & 3 & -\frac{1}{2} \\ -\frac{1}{4} & -\frac{1}{2} & -\frac{1}{4} \end{bmatrix}.$$

For $\beta > 0.5$ (see Section 7.2.1), standard red-black coloring yields a maximal independent set which results in the blocks

$$\begin{aligned} A_{FF} &= \begin{bmatrix} -\frac{1}{4} & & -\frac{1}{4} \\ & 3 & \\ -\frac{1}{4} & & -\frac{1}{4} \end{bmatrix}, \quad A_{FC} = A_{CF} = \begin{bmatrix} & -\frac{1}{2} & \\ -\frac{1}{2} & & -\frac{1}{2} \\ & -\frac{1}{2} & \end{bmatrix}, \\ D_{FF} &= 3\, I_{n_F}, \quad \tilde{D}_{FF} = 2\, I_{n_F}. \end{aligned}$$

Using

$$\|A_{CF}\|_2^2 \le \|A_{CF}\|_1\, \|A_{CF}\|_\infty \le 4, \quad \|\tilde{D}_{FF}\|_2 = 2,$$

we obtain

$$\|A_{CF}(I_{n_F} - D_{FF}^{-1}A_{FF})^{k-1}\tilde{D}_{FF}^{-1}\|_2 \le \|I_{n_F} - D_{FF}^{-1}A_{FF}\|_2^{k-1} \le \left(\frac{1}{3}\right)^{k-1}.$$

Note that

$$(\tilde{D}_{FF} \;\; A_{FC})\, A^{-1} = (I_{n_F} \;\; 0)\, \hat{A}\, A^{-1},$$

where

$$\hat{A} = \frac{1}{2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}.$$

Straightforward Fourier analysis proves $\|\hat{A}\, A^{-1}\|_2 \le 1$ and therefore,

$$\|I_{n_C} - \tilde{A}_{CC}^{(k)}\, A_S^{-1}\|_2 \le \left(\frac{1}{3}\right)^{k-1}.$$

For the five-point stencil discretization of the anisotropic diffusion equation

$$A = \begin{bmatrix} & -\epsilon & \\ -1 & 2 + 2\,\epsilon & -1 \\ & -\epsilon & \end{bmatrix}$$

with $0 < \epsilon \ll 1$ and β large enough, standard semi-coarsening (coarsening only in x-direction) produces a maximal independent set. This yields the blocks

$$\begin{aligned} A_{FF} &= \begin{bmatrix} & -\epsilon & \\ & 2+2\,\epsilon & \\ & -\epsilon & \end{bmatrix}, \quad A_{FC} = A_{CF} = \begin{bmatrix} -1 & 0 & -1 \end{bmatrix}, \\ D_{FF} &= (2+2\,\epsilon)\, I_{n_F}, \quad \tilde{D}_{FF} = 2\, I_{n_F}. \end{aligned}$$

Similar to the first example,

$$\|A_{CF}(I_{n_F} - D_{FF}^{-1}A_{FF})^{k-1}\tilde{D}_{FF}^{-1}\|_2 \le \|I_{n_F} - D_{FF}^{-1}A_{FF}\|_2^{k-1} \le \left(\frac{\epsilon}{\epsilon+1}\right)^{k-1}.$$

follows and

$$(\tilde{D}_{FF} \quad A_{FC})\, A^{-1} = (I_{n_F} \quad 0)\, \hat{A}\, A^{-1},$$

holds for

$$\hat{A} = \begin{bmatrix} & 0 & \\ -1 & 2 & -1 \\ & 0 & \end{bmatrix}.$$

Note that the algebraic regularity is only measured using the difference operator $\hat{A}$ which contains differences only in the direction of the strong connections. In the classic multigrid convergence theory, there is a severe deterioration of regularity for $\epsilon \to 0$. Here, however, due to the problem dependent measure of the regularity, a Fourier analysis yields

$$\|(\tilde{D}_{FF} \quad A_{FC})\, A^{-1}\|_2 \le 1 \quad \Rightarrow \quad \|I_{n_C} - \tilde{A}_{CC}^{(k)}\, A_S^{-1}\|_2 \le \left(\frac{\epsilon}{\epsilon+1}\right)^{k-1}.$$

Let us consider the five-point stencil discretization of a convection-diffusion equation with the stencil

$$A = \begin{bmatrix} & -\epsilon & \\ -1-\epsilon & 1+4\,\epsilon & -\epsilon \\ & -\epsilon & \end{bmatrix}, \quad 0 < \epsilon \ll 1.$$

As in the previous example, semi-coarsening generates a maximal independent set with respect to G_A. The stencils of the blocks are given by

$$\begin{aligned} A_{FF} &= \begin{bmatrix} & -\epsilon & \\ & 1+4\,\epsilon & \\ & -\epsilon & \end{bmatrix}, \quad A_{FC} = A_{CF} = \begin{bmatrix} -1-\epsilon & 0 & -\epsilon \end{bmatrix}, \\ D_{FF} &= (1+4\,\epsilon)\, I_{n_F}, \quad \tilde{D}_{FF} = (1+2\,\epsilon)\, I_{n_F}. \end{aligned}$$

With the same technique as in the previous examples, we obtain

$$\|A_{CF}(I_{n_F} - D_{FF}^{-1}A_{FF})^{k-1}\tilde{D}_{FF}^{-1}\|_2 \le \|I_{n_F} - D_{FF}^{-1}A_{FF}\|_2^{k-1} \le \left(\frac{2\,\epsilon}{1+4\,\epsilon}\right)^{k-1}.$$

and

$$(\tilde{D}_{FF} \quad A_{FC})\, A^{-1} = (I_{n_F} \quad 0)\, \hat{A}\, A^{-1},$$

for

$$\hat{A} = \left[\begin{array}{ccc} & 0 & \\ -1\epsilon & 1+2\,\epsilon & -\epsilon \\ & 0 & \end{array} \right].$$

Fourier eigenvalue analysis shows

$$\|\hat{A}\, A^{-1}\|_2 \le 1 \quad \Rightarrow \quad \|I_{n_C} - \tilde{A}^{(k)}_{CC}\, A_S^{-1}\|_2 \le \left(\frac{2\,\epsilon}{1+4\,\epsilon}\right)^{k-1}.$$

7.2.4 The Preconditioner

The coarse grid matrices for the algebraic multigrid preconditioner in [Reusken 1997; Reusken 1998] are recursively constructed as explained in Section 7.2.2.

ALGORITHM 7.2.2 `Reusken_CGM`*(A) constructs the coarse grid matrices* A_l, $l = 1, \ldots, l_{\max}$ *for a given matrix* $A = A_0$.

```
Reusken_CGM(A)
{
    A_0 = A;
    for(l = 1; j < l_max; l = l + 1)
    {
        A_l = Ã^(2)_{l-1,CC};
        reduce the stencil size of A_l by lumping if necessary;
    }
}
```

To formulate one iteration step of the preconditioner, we need to introduce a special smoother for the F-vertices. $\mathcal{S}^{\nu}_{l,FF}(x_{l,F}, f_{l,F})$ computes an approximate solution $x_{l,F}$ of the linear system $A_{l,FF}\, x^*_{l,F} = f_{l,F}$ using ν Gauß-Seidel iterations starting with the initial guess $x^{(0)}_{l,F} = D^{-1}_{l,FF}\, f_{l,F}$.

ALGORITHM 7.2.3 *Assume the coarse grid matrices* A_i *have been constructed as in Algorithm 7.2.2 and partitioned according to Algorithm 7.2.1. Let a fine grid smoother* $\mathcal{S}^{\nu}_l(x_l, f_l)$ *be given as explained above. The function* `Reusken_AMG`$(0, u, f)$ *calculates one iteration step of the preconditioner in [Reusken 1997; Reusken 1998].*

```
Reusken_AMG(l, u_l, f_l)
{
    if(l == l_max)  u_l = A_l^{-1} f_l;
    else
    {
        S^ν_{l,FF}(z_F, f_{l,F});
```

```
        f_{l,C} = f_{l,C} - A_{l,CF} z_F ;
        f_{l+1} = f_{l,C} ;
        v_{l+1} = 0;
        for(j = 0; j < γ; j = j + 1)   Reusken_AMG(l + 1, v_{l+1}, f_{l+1});
        u_{l,C} = v_{l+1} ;
        f_{l,F} = f_{l,F} - A_{l,FC} u_{l,C} ;
        S^ν_{l,FF}(u_{l,F}, f_{l,F});
    }
}
```

7.3 Approximated Inverses in Algebraic Multigrid Methods

In [Notay 1998], the approximation of the block A_{FF} of the symmetric and positive definite system matrix

$$A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \tag{7.3.1}$$

by a matrix B_{FF} in the two-level block-elimination preconditioner

$$M = \begin{pmatrix} B_{FF} & 0 \\ A_{CF} & A_C \end{pmatrix} \begin{pmatrix} I_{n_F} & B_{FF}^{-1} A_{FC} \\ 0 & I_{n_C} \end{pmatrix} \tag{7.3.2}$$

is analyzed. $A_C = A_C^T$ is supposed to be a spectrally equivalent approximation of the Schur-complement $A_S = A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}$. Since A_{FF} is usually a well conditioned matrix, only little attention is paid to the approximation of A_{FF} in many papers dealing with block-elimination methods. Although A_{FF} is well conditioned, inverting A_{FF} exactly might be quite expensive and the approximation of A_{FF}^{-1} might lead to very inefficient preconditioners as shown in [Notay 1998] and summarized below.

In order to derive a lower bound for the condition number $\kappa(M^{-1} A)$ the Rayleigh quotients

$$\bar{r} = \frac{(\bar{v}, A\,\bar{v})_2}{(\bar{v}, M\,\bar{v})_2}, \quad \underline{r} = \frac{(\underline{v}, A\,\underline{v})_2}{(\underline{v}, M\,\underline{v})_2}$$

associated to vectors of the form

$$\bar{v} = \begin{pmatrix} -B_{FF}^{-1} A_{FC} v_C \\ v_C \end{pmatrix}, \quad \underline{v} = \begin{pmatrix} -A_{FF}^{-1} A_{FC} v_C \\ v_C \end{pmatrix},$$

are evaluated. A straightforward computation shows

$$\begin{aligned}
(\bar{v}, A\,\bar{v})_2 &= (v_C, A_S\, v_C)_2 + (v_C, A_{CF}\,[A_{FF}^{-1} - 2\,B_{FF}^{-1} + B_{FF}^{-1} A_{FF}\, B_{FF}^{-1}]\, A_{FC}\, v_C)_2, \\
(\bar{v}, M\,\bar{v})_2 &= (v_C, A_C\, v_C)_2, \\
(\underline{v}, A\,\underline{v})_2 &= (v_C, A_S\, v_C)_2, \\
(\underline{v}, M\,\underline{v})_2 &= (v_C, A_C\, v_C)_2 + (v_C, A_{CF}\,[B_{FF}^{-1} - 2\,A_{FF}^{-1} + A_{FF}^{-1} B_{FF}\, A_{FF}^{-1}]\, A_{FC}\, v_C)_2.
\end{aligned}$$

Let λ_M be defined by $\lambda_M = \lambda_{\max}(B_{FF}^{-1} A_{FF})$, then using

$$A_{FF}^{-1} - 2\, B_{FF}^{-1} + B_{FF}^{-1} A_{FF}\, B_{FF}^{-1} = (I_{n_F} - B_{FF}^{-1} A_{FF})^2\, A_{FF}^{-1},$$

we get for all w_F

$$\begin{aligned}(w_F, [B_{FF}^{-1} - 2\, A_{FF}^{-1} + A_{FF}^{-1}\, B_{FF}\, A_{FF}^{-1}]\, w_F)_2 = & \\ &= (w_F, [A_{FF}^{-1} B_{FF}][A_{FF}^{-1} - 2\, B_{FF}^{-1} + B_{FF}^{-1} A_{FF}\, B_{FF}^{-1}]\, w_F)_2 \\ &\leq \lambda_M^{-1} (w_F, [I_{n_F} - B_{FF}^{-1} A_{FF}]^2\, A_{FF}^{-1}\, w_F)_2.\end{aligned}$$

Hence, with

$$\begin{aligned} q_{v_C} &= \frac{(v_C, A_S\, v_C)_2}{(v_C, A_C\, v_C)_2}, \\ g_{v_C} &= \frac{(v_C, A_{CF}\, [I_{n_F} - B_{FF}^{-1} A_{FF}]^2\, A_{FF}^{-1}\, A_{FC}\, v_C)_2}{(v_C, A_S\, v_C)_2}, \end{aligned}$$

we obtain

$$\bar{r} = q_{v_C}(1 + g_{v_C}), \quad \underline{r} \leq \frac{1}{q_{v_C}^{-1} + \lambda_M^{-1}\, g_{v_C}}.$$

Since $\lambda_{\max}(M^{-1} A) \geq \bar{r}$ and $\lambda_{\min}(M^{-1} A) \leq \underline{r}$, this implies

$$\kappa(M^{-1} A) = \frac{\lambda_{\max}(M^{-1} A)}{\lambda_{\min}(M^{-1} A)} \geq \frac{\bar{r}}{\underline{r}} \geq (1 + g_{v_C}) \left(1 + \frac{q_{v_C}}{\lambda_M} g_{v_C}\right). \tag{7.3.3}$$

(7.3.3) shows that the two-level method can only be efficient if g_{v_C} is bounded above independently of the mesh size. If A is the result of a standard finite element discretization of a second order elliptic partial differential equation, $(v_C, A_S\, v_C)_2 = O(h^2)$ for smooth vectors v_C which further implies

$$(v_C, A_{CF}\, A_{FF}^{-1}\, A_{FC}\, v_C)_2 \approx (v_C, A_{CC}\, v_C)_2 = O(1).$$

Hence, g_{v_C} can only be bounded independently of the mesh size if B_{FF}^{-1} acts nearly as an exact inverse of A_{FF} for these smooth vectors.

To illustrate this, let A be the standard five-point stencil (1.2.5) discretization of the Laplacian on the unit square with a uniform mesh Ω_h of mesh size h

$$\Omega_h = \{\, (x, y) \in \Omega \,|\, (x, y) = (i\, h, j\, h),\ i, j = 1, \ldots, n,\ h = \frac{1}{n+1}\,\}.$$

The set of vertices $V = \Omega_h$ is split into $V = F \oplus C$ such that

$$F = \Omega_h \setminus \Omega_{2\,h}, \quad C = \Omega_h \cap \Omega_{2\,h}.$$

Assume that $B_{FF} = D_{FF} = \mathrm{diag}(A_{FF})$. The vectors $e^{(\nu,\mu)}$

$$e^{(\nu,\mu)} = \sin(\nu\, \pi\, x)\, \sin(\nu\, \pi\, y), \quad (x, y) \in \Omega_{2h}$$

are shown [Reusken 1995] to be the eigenvectors of A_S with corresponding eigenvalues

$$\lambda^{(\nu,\mu)} = 4\left(\frac{1}{4-2\,(c_\nu+c_\mu)}+\frac{1}{4-2\,(c_\nu-c_\mu)}+\frac{1}{4+2\,(c_\nu-c_\mu)}+\frac{1}{4+2\,(c_\nu+c_\mu)}\right)^{-1}$$

where $c_\nu = \cos(\nu\,\pi\,h)$, $c_\mu = \cos(\mu\,\pi\,h)$. For $\nu=\mu$, it takes the simpler expression

$$\lambda^{(\nu,\nu)} = \frac{8\,(1-c_\nu^2)}{2-c_\nu^2}.$$

It is easy to check that

$$(I_{n_F} - D_{FF}^{-1}\,A_{FF})^2 A_{FC}\,e^{(\nu,\nu)} = \frac{c_\nu^2}{2}\,A_{FC}\,e^{(\nu,\nu)}.$$

Thus, because $A_{CC} = 4\,I_{n_C}$ and $(I_{n_F} - D_{FF}^{-1}\,A_{FF}) = (I_{n_F} - D_{FF}^{-1}\,A_{FF})^T$

$$\begin{aligned} g_{e^{(\nu,\nu)}} &= \frac{c_\nu^2}{2}\,\frac{(e^{(\nu,\nu)}, [A_{CC}-A_S]\,e^{(\nu,\nu)})_2}{(e^{(\nu,\nu)}, A_S\,e^{(\nu,\nu)})_2} \\ &= \frac{c_\nu^2}{2}\,\frac{c_\nu^2}{2\,(1-c_\nu^2)}. \end{aligned}$$

Therefore, for the smoothest mode $\nu=1$,

$$g_{e^{(1,1)}} \approx \frac{h^{-2}}{4\,\pi^2}$$

showing with (7.3.3) that $\kappa(M^{-1}\,A) = O(h^{-4})$!

The analysis in [Notay 1998] shows that the following two conditions

$$\begin{aligned} (v_F, B_{FF}\,v_F)_2 &\le (v_F, A_{FF}\,v_F)_2 \quad \forall\, v_F \in \mathbb{R}^{n_F}, && (7.3.4)\\ (v_C, A_{CF}\,B_{FF}^{-1}\,A_{FC}\,v_C)_2 &\le (1-\xi)(v_C, A_{CC}\,v_C)_2 \\ &\quad +\xi(v_C, A_{CF}\,A_{FF}^{-1}\,A_{FC}\,v_C)_2 \quad \forall\, v_C \in \mathbb{R}^{n_C} && (7.3.5) \end{aligned}$$

with $\xi < 1$ are sufficient to get an appropriate approximation B_{FF} of A_{FF}. This is relatively weak since ξ might be negative, but nevertheless sufficient to entail an acceptable behavior, as it can be seen on an intuitive basis: whenever

$$(v_C, A_{CC}\,v_C)_2 = (1+O(h^2))(v_C, A_{CF}\,A_{FF}^{-1}\,A_{FC}\,v_C)_2,$$

(7.3.4) and (7.3.5) imply

$$\begin{aligned} (v_C, A_{CF}\,A_{FF}^{-1}\,A_{FC}\,v_C)_2 &\le (v_C, A_{CF}\,B_{FF}^{-1}\,A_{FC}\,v_C)_2 \\ &\le (1+(1-\xi)\,O(h^2))(v_C, A_{CF}\,A_{FF}^{-1}\,A_{FC}\,v_C)_2, \end{aligned}$$

i.e. B_{FF}^{-1} has to act nearly as an exact inverse for the corresponding modes as long as $1-\xi = O(1)$.

(7.3.4) and (7.3.5) are the main assumptions for the following theorem in [Notay 1998].

THEOREM 7.3.1 *Let A in (7.3.1) be symmetric and positive definite and let M be defined by (7.3.2) with A_{FF} and B_{FF} regular. Assume that (7.3.5) and*

$$\beta\,(v_F, A_{FF}\,v_F)_2 \le (v_F, B_{FF}\,v_F)_2 \le (v_F, A_{FF}\,v_F)_2 \quad \forall v_F \in \mathbb{R}^{n_F}$$

hold for some β with $0 < \beta < 1$ and $\xi < 1$. Further assume that $A_C = A_C^T$ is spectrally equivalent to the Schur-complement $A_S = A_{CC} - A_{CF}\,A_{FF}^{-1}\,A_{FC}$

$$\eta\,(v_C, A_S\,v_C)_2 \le (v_C, A_C\,v_C)_2 \le \zeta\,(v_C, A_S\,v_C)_2 \quad \forall v_C \in \mathbb{R}^{n_C}$$

with $0 \le \eta \le 1 \le \zeta$. Then,

$$\alpha\,(v, A\,v)_2 \le (v, M\,v)_2 \le \gamma\,(v, A\,v)_2 \quad \forall v \in \mathbb{R}^n,$$

where γ is the smallest root of

$$\gamma^2 - \gamma(\eta + 1 - \xi + \beta\,\xi) + \beta\,\eta = 0$$

and α is the largest root of

$$\alpha^2 - \alpha(\zeta + 1 - \xi + \beta\,\xi) + \beta\,\zeta = 0.$$

Moreover,

$$\begin{aligned} \gamma &\ge \frac{\eta\,\beta}{\eta + 1 - \xi + \xi\,\beta}, \\ \alpha &\le \zeta\left(1 + \frac{(1-\beta)(1-\xi)}{\zeta - \beta}\right). \end{aligned}$$

PROOF. Since the proof is rather technical, we refer to [Notay 1998]. □

Finally, an approximation B_{FF} of A_{FF} which meets the conditions (7.3.4) and (7.3.5) has to be found. As shown above, simple Jacobi smoothing $B_{FF} = \mathrm{diag}(A_{FF})$ is not sufficient. Therefore, an MILU [Axelsson 1994; Dupont, Kendall, and Rachford 1968; Gustafsson 1978; Wittum 1989b] factorization of A_{FF} is analyzed in [Notay 1998].

PROPOSITION 7.3.1 *[Berman and Plemmons 1979] A symmetric matrix C that has non-positive off-diagonal entries is an (possibly singular) M-matrix if and only if is positive semi-definite or, equivalently, if and only if $C\,x \ge 0$ for some $x > 0$.*

PROPOSITION 7.3.2 *[Notay 1998] Let B_{FF} be the result of a (possibly blockwise) MILU decomposition of a symmetric M-matrix A_{FF} satisfying the filter condition*

$$B_{FF}\,x_F = A_{FF}\,x_F \ge 0$$

for a vector $x_F > 0$. Then, $A_{FF} - B_{FF}$ is a symmetric M-matrix.

Proposition 7.3.2 shows (7.3.4) for an MILU factorization B_{FF} of A_{FF}. In the remainder of this section, (7.3.5) is discussed.

LEMMA 7.3.1 *Let A be a symmetric and positive definite M-matrix and let $x = (x_F, x_C)^T > 0$ be a vector with $A\,x \geq 0$. (Such a vector exists for all M-matrices.) $\tilde{D}_{FF}$ denotes the diagonal matrix which satisfies the filter condition*

$$\tilde{D}_{FF}\,x_F = A_{FF}\,x_F.$$

Then,

$$(v_F, B_{FF}\,v_F)_2 \geq (v_F, \tilde{D}_{FF}\,v_F)_2 \tag{7.3.6}$$

yields (7.3.5) with $\xi = 0$.

PROOF. Since $\tilde{A}\,x = A\,x$ for

$$\tilde{A} = \begin{pmatrix} \tilde{D}_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix},$$

$\tilde{A}$ is a symmetric M-matrix and, hence, positive definite. Note that $(1-\xi)\,A_{CC} - A_{CF}\,(B_{FF}^{-1} - \xi\,A_{FF}^{-1})\,A_{FC}$ is the Schur-complement of

$$\begin{pmatrix} B_{FF} & A_{FC} \\ A_{CF} & A_{CC} - \xi A_S \end{pmatrix}.$$

Thus, (7.3.5) holds if and only if the latter matrix is positive definite. Due to assumption (7.3.6), it is sufficient to check

$$v^T \begin{pmatrix} \tilde{D}_{FF} & A_{FC} \\ A_{CF} & A_{CC} - \xi A_S \end{pmatrix} v \geq 0 \quad \forall\, v,$$

which obviously holds for $\xi = 0$ because $\tilde{A}$ is positive definite. □

A better bound $\xi \geq 1/2$ is shown in [Notay 1998] for $x = (1, \ldots, 1)^T$ when A is a linear finite element matrix.

Having Lemma 7.3.1 in mind, only (7.3.6) is left to prove. This is done in the following lemma for a class of factorizations B_{FF}. For instance, an (possibly blockwise) MILU factorization of an M-matrix is a member of this class.

LEMMA 7.3.2 *Let $B_{FF} = (Q - F^T)\,Q^{-1}\,(Q - F)$ be such that Q is a symmetric M-matrix and $F \geq 0$ is strictly upper triangular. Assume that $B_{FF}\,y_F \geq 0$ for some $y_F > 0$. Then,*

$$(v_F, B_{FF}\,v_F)_2 \geq (v_F, \Delta_{FF}\,v_F)_2 \quad \forall\, v_F$$

where Δ_{FF} is the diagonal matrix with $B_{FF}\,y_F = \Delta_{FF}\,y_F$.

PROOF. Since Q is an M-matrix, $Q^{-1} \geq 0$. From

$$(I - Q^{-1}\,F)\,x_F = Q^{-1}[\Delta_{FF}\,x_F + F^T\,(I - Q^{-1}\,F)\,x_F]$$

it is easily seen by induction that $(I - Q^{-1}\,F)\,x_F \geq 0$. Let Z be the diagonal matrix such that $Z\,x_F = Q^{-1}\,F\,x_F$. This yields

$$B_{FF} - \Delta_{FF} = (Z - F^T\,Q^{-1})\,Q\,(Z - Q^{-1}\,F) + [Q - Z\,Q\,Z - (I-Z)\,F - F^T\,(I-Z) - \Delta].$$

The first term of the right hand side is positive semi-definite with $(Z - Q^{-1}F)\,x_F = 0$. Therefore, the term under brackets satisfies

$$[Q - Z\,Q\,Z - (I - Z)\,F - F^Z\,(I - Z) - \Delta]\,x_F = 0$$

and has non-positive off-diagonal entries because Z is diagonal with $I - Z \geq 0$. Therefore, this term is a (singular) M-Matrix and, hence, positive semi-definite. □

COROLLARY 7.3.1 *The choice $y_F = x_F$ and $\Delta_{FF} = \tilde{D}_{FF}$ with x_F, $\tilde{D}_{FF}$ in Lemma 7.3.1 and y_F and Δ_{FF} in Lemma 7.3.2 proofs (7.3.6). Thus, due to Lemma 7.3.1 and Proposition 7.3.2, the assumptions of Theorem 7.3.1 are met for an MILU decomposition B_{FF} of A_{FF}.*

7.4 Numerical Experiments

The performance of the semi-algebraic multigrid preconditioner in Section 7.1 is numerically tested in [Axelsson and Vassilevski 1990]. As matrix B_{FF} in Algorithm 7.1.1 a block-ILU decomposition of A_{FF} is applied and the polynomials (7.1.4) with an appropriate choice of α are used.

EXPERIMENT 7.4.1 *The standard model problem*

$$\Delta\,u(x) = f, \quad x \in \Omega_L$$

on the L-shaped domain in Figure 7.1 with mixed Dirichlet and Neumann boundary conditions is discretized on a regular grid.

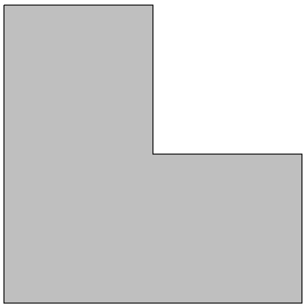

Figure 7.1: L-shaped domain Ω_L.

The number of conjugate gradient iteration steps, the average reduction factor $\bar{\rho}$ and the computing times (on an obviously not very powerful machine) with Algorithm 7.1.1 as preconditioner are shown for $\gamma = 1$ in Table 7.1, for $\gamma = 2$ in Table 7.2, and for $\gamma = 2$ in Table 7.3.

unknowns	steps	$\bar{\rho}$	time/s
736	15	0.2431	30.6
3008	18	0.3050	134
12160	21	0.3526	600
48896	24	0.4020	2719

Table 7.1: Results for Experiment 7.4.1 for $\gamma = 1$.

unknowns	steps	$\bar{\rho}$	time/s
736	7	0.0324	51
3008	7	0.0331	189
12160	7	0.0327	711
48896	7	0.0337	2788

Table 7.2: Results for Experiment 7.4.1 for $\gamma = 2$.

unknowns	steps	$\bar{\rho}$	time/s
736	4	0.0039	86.7
3008	4	0.0043	351
12160	4	0.0055	1368

Table 7.3: Results for Experiment 7.4.1 for $\gamma = 3$.

Reusken discusses in [Reusken 1997; Reusken 1998] the performance of the approximate cyclic reduction preconditioner as preconditioner in an GMRES(5) iteration (restart after 5 iteration steps). For all experiments, the parameters $\beta = 0.7$, $\nu = 2$, and MSIZE $= 14$ are used. The coarsening factor for the presented experiments is slidly larger than 1/2.

EXPERIMENT 7.4.2 *The discretized convection-diffusion equation*

$$\epsilon\, \Delta\, u(x,y) + a(x,y)\, \frac{\partial\, u(x,y)}{\partial\, x} + b(x,y)\, \frac{\partial\, u(x,y)}{\partial\, y} = f, \quad (x,y) \in \Omega = (0,1) \times (0,1),$$

where

$$\begin{aligned} a(x,y) &= \begin{cases} 0.1 & : \ (x,y) \in (0.5,0.8) \times (0.5,0.8), \\ 100 & : \ \textit{otherwise}, \end{cases} \\ b(x,y) &= \begin{cases} 0.2 & : \ (x,y) \in (0.5,0.8) \times (0.5,0.8), \\ 200 & : \ \textit{otherwise}, \end{cases} \end{aligned}$$

with Dirichlet boundary conditions and an uniform grid with mesh size $h = 1/96$ *(9025 unknowns) is considered.*

The numbers of GMRES(5) steps for a 10^{-10} reduction of the residual for several values of ϵ are documented in Table 7.4.

ϵ/h	steps
1000	7
1	5
0.001	3

Table 7.4: Results for Experiment 7.4.2.

The results for a rotated anisotropic equation (Experiment 7.4.3) can be found in Table 7.5.

EXPERIMENT 7.4.3 *The rotated anisotropic equation*

$$-(\epsilon\, c^2 + s^2)\, \frac{\partial^2\, u(x,y)}{\partial\, x^2} - 2\,(\epsilon - 1)\, \frac{\partial^2\, u(x,y)}{\partial\, x\, \partial\, y} - (\epsilon\, s^2 + c^2)\, \frac{\partial^2\, u(x,y)}{\partial\, y^2} = f,$$

$$(x,y) \in \Omega = (0,1) \times (0,1),$$

$$u(x,y) = 0, \quad (x,y) \in \partial\Omega,$$

with $0 < \epsilon < 1$, $c = \cos\varphi$, $s = \sin\varphi$ *and*

$$\varphi = \begin{cases} \frac{\pi}{4} & : \ x \leq \frac{1}{2}, \\ -\frac{\pi}{4} & : \ x > \frac{1}{2}, \end{cases}$$

is discretized on an uniform mesh ($h = 1/96$, 9025 unknowns), resulting in a discrete operator with stencils

$$A = \begin{bmatrix} \frac{1}{2}(\epsilon - 1) & -\epsilon & 0 \\ -\epsilon & 3\,\epsilon + 1 & -\epsilon \\ 0 & -\epsilon & \frac{1}{2}(\epsilon - 1) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & -\epsilon & \frac{1}{2}(\epsilon - 1) \\ -\epsilon & 3\,\epsilon + 1 & -\epsilon \\ \frac{1}{2}(\epsilon - 1) & -\epsilon & 0 \end{bmatrix}$$

on the left half ($x \leq 1/2$) and on the right half ($x > 1/2$).

ϵ/h	steps
0.5	4
10^{-2}	3
10^{-4}	2

Table 7.5: Results for Experiment 7.4.3.

Experiment 7.4.4 *SHERMAN3 with 5005 unknowns and 20033 non-zero entries is taken from the Harwell-Boeing collection. 6 GMRES(5) steps are required for a 10^{-10} reduction of the residual.*

Notay compares in the following experiment (see [Notay 1998]) the condition of $M^{-1}A$ (see (7.3.2)) for B_{FF} is either an ILU decomposition of A_{FF} or an MILU decomposition of A_{FF}. The results are shown in Table 7.6.

Experiment 7.4.5 *The model problem*

$$\begin{aligned} \Delta\, u(x,y) &= f, \quad (x,y) \in \Omega = (0,1) \times (0,1), \\ u(x,y) &= g, \quad (x,y) \in \partial\Omega \end{aligned}$$

is discretized on two uniform grids Ω_h and $\Omega_{2\,h}$. The coarse grid matrix A_C in (7.3.2) is just the system matrix on the coarse grid $\Omega_{2\,h}$.

h	$\kappa(M^{-1}A)$ ILU	$\kappa(M^{-1}A)$ MILU
16	2.78	2.45
32	6.00	2.54
64	28.3	2.58
128	258	2.58

Table 7.6: Results for Experiment 7.4.5.

The condition of $M^{-1}A$ with an MILU decomposition B_{FF} of A_{FF} is reported for Experiment 7.4.6 and Experiment 7.4.7 in Table 7.7.

EXPERIMENT 7.4.6 *The interface model problem*

$$\nabla \cdot (D \nabla u(x,y)) = f, \quad (x,y) \in \Omega = (0,1) \times (0,1),$$

with

$$D = \begin{cases} 1000 & : \quad (x,y) \in (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4}), \\ 1 & : \quad \textit{otherwise}, \end{cases}$$

is discretized on two uniform grids Ω_h *and* Ω_{2h}. *The coarse grid matrix* A_C *in (7.3.2) is just the system matrix on the coarse grid* Ω_{2h}.

EXPERIMENT 7.4.7 *The anisotropic problem*

$$\nabla \cdot (D \nabla u(x,y)) = f, \quad (x,y) \in \Omega = (0,1) \times (0,1),$$

with

$$D = \begin{pmatrix} d_x & 0 \\ 0 & d_y \end{pmatrix},$$

where

$$\begin{aligned} d_x &= \begin{cases} 1000 & : \quad (x,y) \in (\frac{1}{4}, \frac{1}{2}) \times (\frac{1}{4}, \frac{1}{2}), \\ 1 & : \quad \textit{otherwise}, \end{cases} \\ d_y &= \begin{cases} 1000 & : \quad (x,y) \in (\frac{1}{2}, \frac{3}{4}) \times (\frac{1}{2}, \frac{3}{4}), \\ 1 & : \quad \textit{otherwise}, \end{cases} \end{aligned}$$

is discretized on two uniform grids Ω_h *and* Ω_{2h}. *The coarse grid matrix* A_C *in (7.3.2) is just the system matrix on the coarse grid* Ω_{2h}.

h	$\kappa(M^{-1}A)$ Experiment 7.4.6	$\kappa(M^{-1}A)$ Experiment 7.4.7
32	2.85	2.93
64	2.87	3.04
128	2.87	3.10

Table 7.7: Results for Experiment 7.4.5.

Bibliography

Alcouffe, R. E., A. Brandt, J. E. Dendy, and J. W. Painter (1981). The multigrid-method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comput. 2*, 430–454.

Axelsson, O. (1994). *Iterative solution methods.* University Press, Cambridge.

Axelsson, O. and V. Eijkhout (1987). Robust vectorizable preconditioners for three-dimensional elliptic difference equations with anisotropie. In H. J. J. te Riele, T. J. Dekker, and H. A. van der Vorst (Eds.), *Algorithms and applications on vector and parallel computers.* North Holland.

Axelsson, O. and B. Polman (1989). A robust preconditioner based on algebraic substructuring and two-level grids. In W. Hackbusch (Ed.), *Robust multi-grid methods, NNFM 23*, Braunschweig. Vieweg-Verlag.

Axelsson, O. and P. Vassilevski (1989). Algebraic multilevel preconditioning methods; part I. *Numer. Math. 56*, 157–177.

Axelsson, O. and P. Vassilevski (1990). Algebraic multilevel preconditioning methods; part II. *SIAM J. Numer. Anal. 27*, 1569–1590.

Bank, R. (1994). *PLTMG Users Guide Version 7.0.* SIAM, Philadelphia.

Bank, R. E. (1997). Hierarchical bases and the finite element method. *Acta Numerica.*

Bank, R. E., T. F. Dupont, and H. Yserentant (1988). The hierarchical basis multigrid method. *Numer. Math. 52*, 427–458.

Bank, R. E. and R. K. Smith (1998). The incomplete factorization multigraph algorithm. *SIAM J. on Scientific Computing.* to appear.

Bank, R. E. and C. Wagner (1999). Multilevel ILU decomposition. *Numerische Mathematik.* to appear.

Bank, R. E. and J. Xu (1994). The hierarchical basis multigrid method and incomplete LU decomposition. In D. Keyes and J. Xu (Eds.), *Seventh International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pp. 163–173. AMS, Providence, Rhode Island.

Barrett, R., M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst (1994). *Templates for* *the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, Philadelphia.

Bastian, P. (1996). *Parallele adaptive Mehrgitterverfahren.* Teubner Skripten zur Numerik. Teubner-Verlag, Stuttgart.

Berman, A. and R. J. Plemmons (1979). *Nonnegative matrices in the mathematical science.* Academic Press, New York.

Braess, D. (1995). Towards algebraic multigrid methods for elliptic problems of second order. *Computing 55*, 379–393.

Braess, D. (1997). *Finite Elemente.* Springer, Berlin.

Bramble, J., J. Pasciak, J. Wang, and J. Xu (1991). Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp. 57*, 23–45.

Brandt, A. (1973). Multi-level adaptive technique for fast numerical solution to boundary value problems. In *3rd Internationel Conference on Numerical Methods in Fluid Mechanics, Paris, 1972, Lecture Notes in Physics 18*, pp. 83–89. Springer, Berlin.

Briggs, W. L. (1987). *A multigrid tutorial.* SIAM, Philadelphia.

de Zeeuw, P. M. (1990). Matrix-dependent prolongations and restrictions in a black box multigrid solver. *J. Comput. Appl. Math. 33*, 1–27.

Dendy, J. E. (1982). Black box multi-grid. *J. Comput. Physics 48*, 366–386.

Dupont, T., R. P. Kendall, and H. H. Rachford (1968). An approximate factorization procedure for solving self-adjoined elliptic difference equations. *SIAM J. Numer. Anal. 5*, 559–573.

Fedorenko, R. P. (1961). A relaxation scheme for the solution of elliptic differential equations (russian). *UdSSR Comput. Math. Phys 1,5*, 1092–1096.

Fuhrmann, J. (1994). *Zur Verwendung von Mehrgitterverfahren bei der numerischen Behandlung elliptischer partieller Differentialgleichungen zweiter Ordnung mit variablen Koeffizienten.* Ph. D. thesis, TU Chemnitz-Zwickau.

Fuhrmann, J. (1995). A modular algebraic multilevel method. Technical Report Preprint 203, WIAS, Berlin.

George, A. and J. Liu (1981). *Computer solutions of large sparse positive definite systems.* Prentice Hall, Engelwood Cliffs.

Golub, G. H. and C. F. van Loan (1983). *Matrix computations.* Johns Hopkins University Press.

Griebel, M. (1994). *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen.* Teubner Verlag, Stuttgart.

Gustafsson, I. (1978). A class of first order factorization methods. *BIT 18*, 142–156.

Hackbusch, W. (1976). Ein iteratives Verfahren zur schnellen Auflösung elliptischer Randwertprobleme. Technical Report 76-12, Universität Köln.

Hackbusch, W. (1985). *Multi-Grid Methods and Applications.* Springer, Berlin.

Hackbusch, W. (1986). *Theorie und Numerik elliptischer Differentialgleichungen.* Teubner, Stuttgart.

Hackbusch, W. (1989). On first and second order box schemes. *Computing 41*, 277–296.

Hackbusch, W. (1993). *Iterative Löser großer schwachbesetzer Gleichungssysteme.* Teubner, Stuttgart.

Hackbusch, W. (1994). *Iterative solution of large sparse systems.* Springer, New York.

Hackbusch, W. and G. Wittum (1993). *Incomplete Decompositions — Theory, Algorithms and Applications, NNFN 41.* Vieweg Verlag, Braunschweig.

Hemker, P. W. (1983). Multigrid methods for problems with a small parameter in the highest derivate. In D. Griffiths (Ed.), *Numerical analysis, Proceedings, Dundee 1983*, Berlin. Springer-Verlag.

Hemker, P. W. (1990). On the order of prolongations and restrictions in multigrid procedures. *J. Comp. Appl. Math. 32*, 423–429.

Heppner, I. (1999). *Algebraische Mehrgitterverfahren für Systeme partieller Differentialgleichungen.* Ph. D. thesis, Universität Heidelberg. in preparation.

Horowitz, E. and S. Sahni (1984). *Fundamentals of data structures in Pascal.* Pitman, London.

Kaczmarz, S. (1937). Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin de l'Academie Polonaise des Sciences et Lettres 35*, 355–357.

Karypis, G. and V. Kumar (1995a). Analysis of a multilevel graph partitioning. Technical Report 95-037, University of Minnesota, Minneapolis.

Karypis, G. and V. Kumar (1995b). A fast and high quality multilevel scheme for partitioning irregular graphs. Technical Report 95-035, University of Minnesota, Minneapolis.

Kettler, R. (1981). Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. In W. Hackbusch and U. Trottenberg (Eds.), *Multigrid Methods, Lecture Notes in Math. 960*, pp. 502–534. Springer, Berlin.

Kettler, R. and P. Wesseling (1986). Aspects of multigrid methods for problems in three dimensions. *Appl. Math. Comp. 19*, 159–168.

Khalil, M. and P. Wesseling (1992). Vertex-centered and cell-centered multigrid for interface problems. *J. Comp. Phys. 98*, 1–10.

Notay, Y. (1997). Optimal V cycle algebraic multilevel preconditioning. Technical Report GANMN 97-05, Université Libre de Bruxelles, Brussels, Belgium. *http://homepages.ulb.ac.be/˜ynotay.*

Notay, Y. (1998). Using approximate inverses in algebraic multilevel methods. *Numer. Math.*. to appear.

Reusken, A. (1991). A new lemma in multigrid convergence theory. Technical Report RANA report 91-07, Department of Mathematics and Computing Science, Eindhoven.

Reusken, A. (1994). Multigrid with matrix-dependent transfer operators for convection-diffusion problems. In P. W. Hemker and P. Wesseling (Eds.), *Seventh International Symposium on Domain Decomposition Methods for Partial Differential Equations*. Birkhäuser, Basel.

Reusken, A. (1995). Fourier analysis of a robust multigrid method for convection-diffusion equations. *Numerische Mathematik 71*, 365–398.

Reusken, A. (1996). A multigrid method based on incomplete gaussian elimination. *J. Num. Lin. Alg. with Appl. 3*, 369–390.

Reusken, A. (1997). On the approximate cyclic reduction preconditioner. Technical Report Bericht Nr. 144, Institut für Geometrie und Praktische Mathematik, RWTH Aachen.

Reusken, A. (1998). An approximate cyclic reduction scheme. In W. Hackbusch and G. Wittum (Eds.), *Multigrid Methods V. Lecture Notes in Computational Science and Engineering*. Springer, Heidelberg.

Ruge, J. W. and K. Stüben (1987). Algebraic multigrid. In S. F. McCormick (Ed.), *Multigrid methods*. SIAM, Philadelphia.

Saad, Y. (1996a). ILUM: A multi-elimination ILU preconditioner for general sparse matrices. *SIAM J. Sci. Comput. 17*, 830–847.

Saad, Y. (1996b). *Iterative methods for sparse linear systems*. PWS Publishing Company, Boston.

Schneider, G. E. and M. J. Raw (1987). Control volume finite element method for heat transfer and fluid flow using colocated variables. *Numer. Heat Transfer 11*, 363–390.

van der Ploeg, A., E. Botta, and F. Wubs (1996). Nested grids ILU-decomposition (NGILU). *J. Comput. Appl. Math. 66*, 515–526.

Vanek, P., M. Brezina, and J. Mandel (1998). Convergence of algebraic multigrid based on smoothed aggregation. Technical Report 126, Center for Computational Mathematics, University of Colorado at Denver.

Vanek, P., J. Mandel, and M. Brezina (1994). Algebraic multigrid on unstructured meshes. Technical report, Center for Computational Mathematics, University of Colorado at Denver.

Vanek, P., J. Mandel, and M. Brezina (1996). Algebraic multigrid by smoothed aggregation for second order and forth order elliptic problems. *Computing 56*, 179–196.

Varga, R. (1962). *Matrix iterative analysis*. Prentice-Hall, Englewood Cliffs.

Wagner, C. (1995). *Frequenzfilternde Zerlegungen für unsymmetrische Matrizen und Matrizen mit stark variierenden Koeffizienten*. Ph. D. thesis, Universität Stuttgart. ICA-Bericht 95/7, Stuttgart.

Wagner, C. (1997a). Tangential frequency filtering decompositions for symmetric matrices. *Numerische Mathematik 78*, 119–142.

Wagner, C. (1997b). Tangential frequency filtering decompositions for unsymmetric matrices. *Numerische Mathematik 78*, 143–163.

Wagner, C., W. Kinzelbach, and G. Wittum (1997). Schur-complement multigrid — a robust method for groundwater flow and transport problems. *Numerische Mathematik 75*, 523–545.

Wagner, C. and G. Wittum (1997). Adaptive filtering. *Numerische Mathematik 78*, 305–328.

Wesseling, P. (1988). Cell-centered multigrid for interface problems. *J. Comp. Phys. 79*, 85–91.

Wesseling, P. (1992). *An introduction to multigrid methods*. Wiley, Chichester.

Wittum, G. (1986). *Distributive Iterationen für indefinite Systeme als Glätter in Mehrgitterverfahren am Beispiel der Stokes- und Navier-Stokes-Gleichungen mit Schwerpunkt auf unvollständigen Zerlegungen*. Ph. D. thesis, Universität Heidelberg.

Wittum, G. (1989a). On the convergence of multigrid methods with transforming smoothers. Theory with application to the Navier-Stokes equations. *Numer. Math. 54*, 543–563.

Wittum, G. (1989b). On the robustness of ILU smoothing. *SIAM J. Sci. Statist. Comput. 10*, 699–717.

Wittum, G. (1992). *Filternde Zerlegungen – Schnelle Löser für Gleichungssysteme*. Teubner Skripten zur Numerik. Teubner-Verlag, Stuttgart.

Xu, J. (1992). Iterative methods by space decomposition and subspace correction. *SIAM Review 34*, 581–613.

Yserentant, H. (1986). On the multi-level splitting of finite element spaces. *Numer. Math. 49*, 379–412.

Yserentant, H. (1993). Old and new convergence proofs for multigrid methods. *Acta Numerica*.