Multigrid Methods

Volker John

Winter Semester 2013/14

Contents

Chapter 1

Literature

Remark 1.1 *Literature.* There are several text books about multigrid methods, e.g.,
- Briggs et al. (2000), easy to read introduction,
- Hackbusch (1985), the classical book, sometimes rather hard to read,
- Shaidurov (1995),
- Wesseling (1992), an introductionary book,
- Trottenberg et al. (2001).

□

Chapter 2

Model Problems

Remark 2.1 *Motivation.* The basic ideas and properties of multigrid methods will be explained in this course on two model problems. □

Example 2.2 *A two-point boundary value problem.* Consider the boundary value problem

$$
\begin{aligned}
-u'' &= f \quad \text{in } \Omega = (0,1), \\
u(0) = u(1) &= 0.
\end{aligned}
\tag{2.1}
$$

Often, this problem can be solved analytically.

Multigrid methods are solvers for linear system of equations that arise, e.g., in the discretization of partial differential equations. For this reason, discretizations of (2.1) will be considered: a finite difference method and a finite element method. These discretizations are described in detail in the lecture notes of Numerical Mathematics III.

Consider an equidistant triangulation of Ω with the nodes $0 = x_0 < x_1 < \ldots < x_N = 1$ with the distance $h = 1/N$ between two neighboring nodes.

The application of the second order finite difference scheme leads to a linear system of equations

$$
A\mathbf{u} = \mathbf{f}
\tag{2.2}
$$

with the tridiagonal matrix $A \in \mathbb{R}^{(N-1)\times(N-1)}$ with

$$
a_{ij} = \frac{1}{h^2}
\begin{cases}
2 & \text{if } i = j, i = 1, \ldots, N-1, \\
-1 & \text{if } i = j-1, i = 2, \ldots, N-1, \text{ or } i = j+1, i = 1, \ldots, N-2, \\
0 & \text{else},
\end{cases}
\tag{2.3}
$$

and the right-hand side

$$
(\mathbf{f})_i = f_i = f(x_i), \quad i = 1, \ldots, N-1.
$$

Using the P_1 finite element method leads to a linear system of equations (2.2) with the tridiagonal matrix

$$
a_{ij} = \frac{1}{h}
\begin{cases}
2 & \text{if } i = j, i = 1, \ldots, N-1, \\
-1 & \text{if } i = j-1, i = 2, \ldots, N-1, \text{ or } i = j+1, i = 1, \ldots, N-2, \\
0 & \text{else},
\end{cases}
\tag{2.4}
$$

and the right-hand side

$$
f_i = \int_{x_{i-1}}^{x_{i+1}} f(x)\varphi_i(x)\ dx, \quad i = 1, \ldots, N-1,
$$

where $\varphi_i(x)$ is the function from the local basis that does not vanish in the node x_i. Note that there is a different scaling in the matrices of the finite difference and the finite element method. □

Example 2.3 *Poisson equation in two dimension.* The Poisson equation in two dimensions with homogeneous boundary conditions has the form

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega = (0,1)^2, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{2.5}$$

Again, an equidistant grid is considered for the discretization of (2.5) with mesh width $h_x = h_y = h = 1/N$. The nodes are numbered lexicographically.

The application of the finite difference method with the five point stencil leads to a linear system of equations of dimension $(N-1) \times (N-1)$ with the matrix entries

$$a_{ij} = \frac{1}{h^2} \begin{cases} 4 & \text{if } i = j, \\ -1 & \text{if } i = j-1, i = j+1, i = j-(N+1), i = j+(N+1), \\ 0 & \text{else,} \end{cases}$$

with obvious modifications for the nodes near the boundary of the domain.

For applying the P_1 finite element method, the grid has to be decomposed into triangles. Using a decomposition where the edges are either parallel to the axes or parallel to the line $y = x$, one obtains the matrix

$$a_{ij} = \begin{cases} 4 & \text{if } i = j, \\ -1 & \text{if } i = j-1, i = j+1, i = j-(N+1), i = j+(N+1), \\ 0 & \text{else,} \end{cases}$$

again with obvious modifications for the degrees of freedom near the boundary. □

Remark 2.4 *Properties of the matrices.*
- The matrix A is sparse. In one dimension, there are not more than three non-zero entries per row and column, the matrix is even tridiagonal. In the two-dimensional case, there are not more than five non-zero entries per row and column.
- The matrix A is symmetric. It follows that all eigenvalues are real.
- The matrix A is positive definite, i.e.,

$$\mathbf{x}^T A \mathbf{x} > 0 \quad \forall \, \mathbf{x} \setminus \{\mathbf{0}\},$$

where the dimension of the vector $\mathbf{x}$ corresponds to the dimension of the matrix A. It follows that all eigenvalues are positive.
- The matrix A is diagonally dominant, i.e., it is

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

and there is at least one index for which the equal sign is not true. For the considered problems, the upper sign applies for all nodes or degrees of freedom which are close to the boundary.

It is well known from the course on iterative methods for sparse large linear systems of equations, Numerical Mathematics II, that these properties are favorable. In fact, also for multigrid methods, the state of the art is that most of the analysis is known for systems with symmetric positive definite matrices, or matrices which are only

slight perturbations of such matrices. However, in practice, multigrid methods often work very well also for the solution of systems with other matrices.

Even if the properties given above are favorable, the condition of the matrices might be large. A direct calculation reveals (this was an exercise problem in Numerical Mathematics II) that in one dimension, the eigenvalues of the finite element matrix A are

$$\lambda_k = \frac{4}{h} \sin^2\left(\frac{k\pi}{2N}\right), \quad k = 1, \ldots, N-1, \tag{2.6}$$

and the corresponding eigenvectors $\mathbf{v}_k = (v_{k,1}, \ldots, v_{k,N-1})^T$ with

$$v_{k,j} = \sin\left(\frac{jk\pi}{N}\right), \quad j, k = 1, \ldots, N-1. \tag{2.7}$$

Then, a direct calculation, using a theorem for trigonometric functions and a Taylor series expansion, shows for the spectral condition number

$$
\begin{aligned}
\kappa_2(A) &= \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \frac{\sin^2\left(\frac{(N-1)\pi}{2N}\right)}{\sin^2\left(\frac{\pi}{2N}\right)} = \frac{\sin^2\left((1-h)\frac{\pi}{2}\right)}{\sin^2\left(h\frac{\pi}{2}\right)} \\
&= \left(\frac{\sin\left(\frac{\pi}{2}\right)\cos\left(h\frac{\pi}{2}\right) - \cos\left(\frac{\pi}{2}\right)\sin\left(h\frac{\pi}{2}\right)}{\sin\left(h\frac{\pi}{2}\right)}\right)^2 = \left(\frac{\cos\left(h\frac{\pi}{2}\right)}{\sin\left(h\frac{\pi}{2}\right)}\right)^2 \\
&= \cot^2\left(h\frac{\pi}{2}\right) = \left(\frac{2\pi}{h} - \mathcal{O}(h)\right)^2 = \mathcal{O}\left(h^{-2}\right).
\end{aligned}
$$

Also in higher dimensions, the condition number is $\kappa_2(A) = \mathcal{O}\left(h^{-2}\right)$. $\qquad \square$

Example 2.5 *Behavior of iterative methods for the Poisson equation.* Consider the Poisson equation (2.5) with $f = 1$ for all $\mathbf{x} \in \Omega$ and the P_1 finite element discretization of this problem on meshes with different fineness.

Table 2.1 gives the number of iterations and the computing times for different solvers applied to the solution of this problem. The simulations were performed with the research code MooNMD, John and Matthies (2004). The SSOR method and the conjugate gradient method (CG) are already known from Numerical Mathematics II. For these methods, not the system $A\mathbf{u} = \mathbf{f}$ was solved, but the system

$$D^{-1}A\mathbf{u} = D^{-1}\mathbf{f},$$

where D is the diagonal of A. It is known from Numerical Mathematics II that the number of iterations for SSOR can be estimated to be proportional to the condition number of the matrix and the number of iterations for CG to be proportional to the square root of the condition number. If $\kappa_2(D^{-1}A) < \kappa_2(A)$, then the upper bound for the number of iterations becomes better. As comparison, the number of iterations with a multigrid method as solver and with a multigrid method as preconditioner within a flexible general minimized residual (GMRES) method are presented. Finally, the computing times for the application of the sparse direct solver UMFPACK, Davis (2004), are given. UMFPACK is the solver behind the backslash command in MATLAB.

Table 2.1: Example 2.5. Number of iterations and computing times (13/10/11 on a HP BL460c Gen8 2xXeon, Eight-Core 2700MHz). The number of degrees of freedom (d.o.f.) includes the Dirichlet values.

level	h	d.o.f.	SSOR ite	SSOR time	PCG ite	PCG time	MG ite	MG time	FGMRES+MG ite	FGMRES+MG time	UMFPACK ite	UMFPACK time
1	1/4	25	49	0	3	0	11	0	6	0	1	0
2	1/8	81	164	0	9	0	13	0	8	0	1	0
3	1/16	289	543	0	31	0	13	0	8	0	1	0
4	1/32	1089	2065	0.07	66	0.01	14	0.03	8	0.01	1	0.01
5	1/64	4225	7998	0.92	132	0.02	14	0.11	8	0.03	1	0.03
6	1/128	16641	31054	14.61	263	0.16	13	0.35	8	0.21	1	0.12
7	1/256	66049	> 100000		524	1.79	13	1.55	8	1.06	1	0.75
8	1/512	263169			1038	16.55	12	6.09	8	3.90	1	5.40
9	1/1024	1050625			1986	127.76	12	27.46	7	18.32	1	46.46
10	1/2048	4198401			3944	1041.68	12	111.03	7	68.38	1	
factor $\approx$		4	4	16	2	8	1	4	1	4	1	

The number of floating point operations per iteration is for all iterative methods proportional to the number of degrees of freedom. One gets for the complete number of operations the values from Table 2.2. One can observe that the estimate for the number of iterations is sharp for PCG. For the multigrid approaches, the total number of operations is proportional to the number of unknowns. Since in the solution of a linear system of equations, each unknown has to be considered at least once, the total number of operations is asymptotically optimal for multigrid methods.

Table 2.2: Example 2.5. Number of floating point operations, where n is the number of degrees of freedom.

method	op./iter.	no. of iterations	total no. of operations
SSOR	$\mathcal{O}\left(n\right)$	$\mathcal{O}\left(\kappa_2(A)\right) = \mathcal{O}\left(h^{-2}\right) = \mathcal{O}\left(n^2\right)$	$\mathcal{O}\left(n^3\right)$
PCG	$\mathcal{O}\left(n\right)$	$\mathcal{O}\left(\sqrt{\kappa_2(A)}\right) = \mathcal{O}\left(h^{-1}\right) = \mathcal{O}\left(n\right)$	$\mathcal{O}\left(n^2\right)$
MG	$\mathcal{O}\left(n\right)$	$\mathcal{O}\left(1\right)$	$\mathcal{O}\left(n\right)$

In addition, it can be seen that it is even more efficient to use the multigrid method as a preconditioner in a Krylov subspace method than as a solver. One has to use here the flexible GMRES method since the preconditioner is not a fixed matrix but a method. That means, the preconditioner might change slightly from iteration to iteration. The flexible GMRES method can cope with this difficulty.

The development of sparse direct solvers has shown remarkable progress in the last couple of years. One can observe that for the model problem, the direct solver is best for small and medium sized problems, up to about 100000 degrees of freedom. But for large problems, good iterative methods are still better. On the fine grid, UMFPACK is not able to solve the problem because there is an internal memory limitation in this program. □

Chapter 3

Detailed Investigation of Classical Iterative Schemes

3.1 General Aspects of Classical Iterative Schemes

Remark 3.1 *Motivation.* This chapter studies the reason for the inefficient behavior, observed in Example 2.5 for SSOR, of classical iterative schemes in detail. This study is performed for the one-dimensional problem (2.1) and the linear system of equations has the form

$$A\mathbf{u} = \mathbf{f}. \tag{3.1}$$

$\square$

Remark 3.2 *General approach.* Classical iterative schemes for the solution of (3.1) were introduced and studied in Numerical Mathematics II. Here, a short review is presented and notations are introduced.

Classical iterative schemes are based on a fixed point iteration for solving the linear system of equations. To this end, decompose the matrix

$$A = M - N, \tag{3.2}$$

where M is a non-singular matrix. Then, one can write system (3.1) in the fixed point form

$$M\mathbf{u} = N\mathbf{u} + \mathbf{f}$$

or

$$\mathbf{u} = M^{-1}N\mathbf{u} + M^{-1}\mathbf{f} =: S\mathbf{u} + M^{-1}\mathbf{f}.$$

Given an initial iterate $\mathbf{u}^{(0)}$, a fixed point iteration can be applied to this equation

$$\mathbf{u}^{(m+1)} = S\mathbf{u}^{(m)} + M^{-1}\mathbf{f}, \quad m = 0, 1, 2, \dots \tag{3.3}$$

This basic iterative approach might be also damped

$$\mathbf{u}^* = S\mathbf{u}^{(m)} + M^{-1}\mathbf{f}, \quad \mathbf{u}^{(m+1)} = \omega\mathbf{u}^* + (1-\omega)\mathbf{u}^{(m)}, \quad \omega \in \mathbb{R}^+,$$

such that

$$\mathbf{u}^{(m+1)} = (\omega S + (1-\omega)I)\,\mathbf{u}^{(m)} + \omega M^{-1}\mathbf{f}. \tag{3.4}$$

$\square$

Remark 3.3 *The residual equation.* Let **u** be the solution of (3.1) and $\mathbf{u}^{(m)}$ an approximation computed with (3.3) or (3.4). The error is denoted by

$$\mathbf{e}^{(m)} = \mathbf{u} - \mathbf{u}^{(m)}$$

and the residual by

$$\mathbf{r}^{(m)} = \mathbf{f} - A\mathbf{u}^{(m)}.$$

It is for the fixed point iteration (3.3)

$$
\begin{aligned}
S\mathbf{e}^{(m)} &= M^{-1}N\mathbf{u} - S\mathbf{u}^{(m)} = M^{-1}N\mathbf{u} - \mathbf{u}^{(m+1)} + M^{-1}\mathbf{f} \\
&= M^{-1}\left(N\mathbf{u} + \mathbf{f}\right) - \mathbf{u}^{(m+1)} = \mathbf{u} - \mathbf{u}^{(m+1)} = \mathbf{e}^{(m+1)}.
\end{aligned}
\tag{3.5}
$$

For both iterations (3.3) and (3.4), the so-called residual equation has the form

$$A\mathbf{e}^{(m)} = A\mathbf{u} - A\mathbf{u}^{(m)} = \mathbf{f} - A\mathbf{u}^{(m)} = \mathbf{r}^{(m)}. \tag{3.6}$$

$\square$

Remark 3.4 *To multigrid methods.* In multigrid methods, the residual equation (3.6) is used for updating the current iterate $\mathbf{u}^{(m)}$. An approximation $\tilde{\mathbf{e}}^{(m)}$ of $\mathbf{e}^{(m)}$ is computed from (3.6) and the new iterate is given by $\mathbf{u}^{(m+1)} = \mathbf{u}^{(m)} + \tilde{\mathbf{e}}^{(m)}$. An advantage of using the residual equation is that, at least close to the solution, $\mathbf{e}^{(m)}$ is small and the zero vector is a good initial guess for an iterative solution of (3.6).

$\square$

Remark 3.5 *To the convergence of classical iteration schemes.* From (3.5), it follows by induction that $\mathbf{e}^{(m)} = S^m \mathbf{e}^{(0)}$, such that

$$\left\|\mathbf{e}^{(m)}\right\| \le \|S^m\| \left\|\mathbf{e}^{(0)}\right\| \tag{3.7}$$

for each vector norm and its induced matrix norm. The iteration is called convergent if

$$\lim_{m \to \infty} \|S^m\| = 0$$

and $\|S^m\|$ is called contraction number of the fixed point iteration (3.3). It was shown in the course Numerical Mathematics II, Theorem 3.3 in the part on iterative solvers, that the fixed point iteration (3.3) converges for any initial iterate if and only if $\rho(S) < 1$, where $\rho(S) = \max_i |\lambda_i(S)|$ is the spectral radius of S. In connection with iterative schemes, the spectral radius is also called convergence factor. It is loosely speaking the worst factor for the reduction of the error in each step of the iteration.

For each eigenvalue $\lambda_i \in \mathbb{C}$ of a matrix $A \in \mathbb{R}^{n \times n}$ it is $|\lambda_i| \le \|A\|$, where $\|\cdot\|$ is any induced matrix norm. It follows that $\rho(S) \le \|S\|$.

Let $M \in \mathbb{N}$ be the smallest natural number for which

$$\frac{\left\|\mathbf{e}^{(M)}\right\|}{\left\|\mathbf{e}^{(0)}\right\|} \le 10^{-1},$$

i.e., the smallest number of iterations which are needed for reducing the error by the factor 10. This condition is satisfied approximately if

$$\frac{\left\|\mathbf{e}^{(M)}\right\|}{\left\|\mathbf{e}^{(0)}\right\|} \le \|S^M\| \approx \rho\left(S^M\right) = (\rho(S))^M \approx 10^{-1}.$$

It follows that

$$M \lesssim -\frac{1}{\log_{10} |\rho(S)|}.$$

The number $-\log_{10} |\rho(S)|$ is called rate of convergence. If it is close to zero, i.e., $\rho(S)$ is close to one, then M is large and the convergence is slow. The convergence becomes the faster the closer $\rho(S)$ is to zero.

$\square$

3.2 The Jacobi and Damped Jacobi Method

Remark 3.6 *The Jacobi and the damped Jacobi method.* The Jacobi method is given by $M = \text{diag}(A) = D$ in (3.2). A straightforward calculation shows, see also Numerical Mathematics II, that it has the form

$$\mathbf{u}^{(m+1)} = \mathbf{u}^{(m)} + D^{-1}\mathbf{r}^{(m)}.$$

Also the damped Jacobi method was introduced in Numerical Mathematics II

$$\mathbf{u}^{(m+1)} = \mathbf{u}^{(m)} + \omega D^{-1}\mathbf{r}^{(m)}, \quad \omega \in (0,1]. \tag{3.8}$$

A straightforward calculation shows that it can be written as a basic fixed point iteration (3.3) with $M = \omega^{-1}\text{diag}(A)$. The behavior of the (damped) Jacobi method shall be studied at the one-dimensional model problem (2.1). □

Remark 3.7 *Discrete Fourier modes.* To study the behavior of the (damped) Jacobi method for the one-dimensional model problem, it is sufficient to consider the homogeneous linear system of equations

$$A\mathbf{u} = \mathbf{0} \tag{3.9}$$

and an arbitrary initial iterate $\mathbf{u}^{(0)}$. The solution of the homogeneous system is $\mathbf{u} = \mathbf{0}$. Obviously, the matrix from the finite element discretization can be used without loss of generality.

Let b be a given integrable function in $[0,1]$ with $b(0) = b(1) = 0$. This function can be expanded in the form

$$b(x) = \sum_{k=1}^{\infty} b_k \sin(k\pi x),$$

where k is the wave number, b_k is the k-th Fourier coefficient, the functions $\sin(k\pi x)$ are called Fourier modes, and the frequency is $1/k$. Small wave numbers characterize long and smooth waves, whereas large wave numbers describe highly oscillating waves, see Figure 3.1.
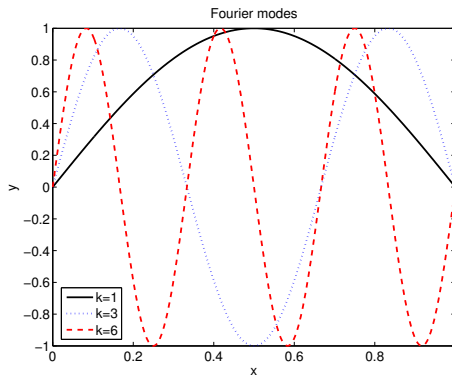

Figure 3.1: Fourier modes.

For the investigation of an iterative method applied to the solution of (3.9), it is of advantage to consider initial iterates (which are equal to the negative of the initial errors) which are discrete, i.e., at $x = j/N$, analogs of the Fourier modes:

$$\mathbf{u}^{(0)} = \left(u_1^{(0)}, \dots, u_{N-1}^{(0)}\right)^T \quad \text{with} \quad u_j^{(0)} = \sin\left(\frac{jk\pi}{N}\right), \quad j,k = 1, \dots, N-1. \tag{3.10}$$

Note that these discrete Fourier modes are also the eigenvectors of the matrix A, see (2.7).

The discrete Fourier modes in the lower part of the spectrum $1 \leq k < N/2$ are called low frequency or smooth modes. The modes in the upper part of the spectrum $N/2 \leq k \leq N-1$ are the so-called high frequency or oscillating modes. Note that the classification of the discrete modes depends on the number of nodes N. The discrete analogs of the Fourier modes have different properties on different grids. □

Example 3.8 *Application of the damped Jacobi method for the solution of the model problem.* The damped Jacobi method (3.8) with $\omega = 2/3$ is applied to the solution of the model in the following two situations:
- the number of intervals N is fixed and the wave number k is varied,
- the wave number k is fixed and the number of intervals N is varied.

For each simulation, 100 iterations were performed and the error is measured in the l^∞ vector norm $\|\cdot\|_\infty$. The obtained results are presented in Figures 3.2 and 3.3
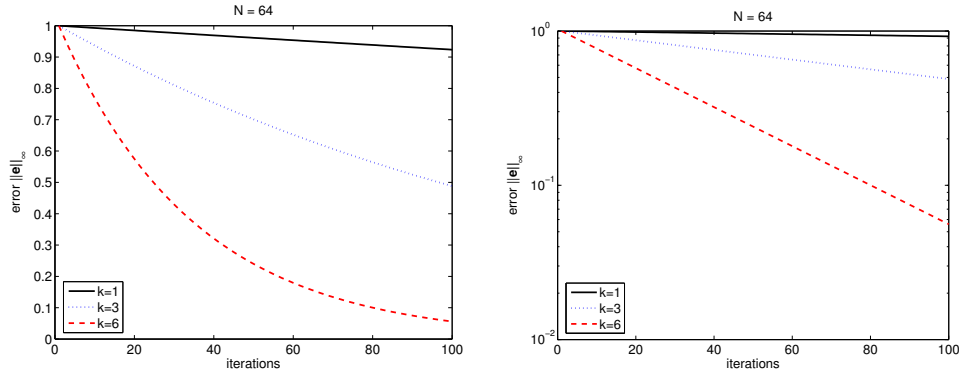

Figure 3.2: Convergence of the damped Jacobi method with $\omega = 2/3$ for initial iterates with different wave numbers on a fixed grid, left linear plot, right semilogarithmic plot.

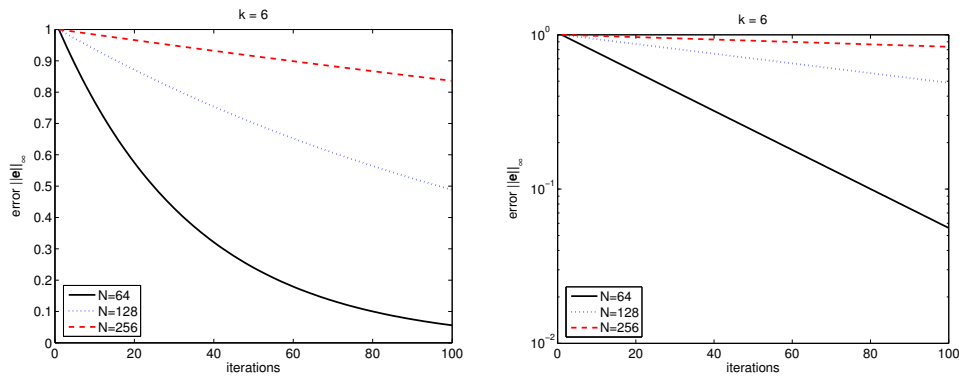Figure 3.3: Convergence of the damped Jacobi method with $\omega = 2/3$ on different grids for an initial iterate with a fixed wave number, left linear plot, right semilogarithmic plot.

The following observations are of importance:
- On a fixed grid, there is a good damping of the high frequency errors whereas there is almost no damping of the low frequency errors (note that $\mathbf{u}^{(0)} = \mathbf{e}_j^{(0)}$).

- For a fixed wave number, the error is reduced on a coarser grid better than on a finer grid.
- The logarithm of the error decays linearly, i.e., the error itself decays geometrically. Thus, there is a constant $0 < C(k) < 1$ such that

$$\left\| \mathbf{e}^{(n)} \right\|_\infty \leq (C(k))^n \left\| \mathbf{e}^{(0)} \right\|_\infty.$$

In practice, one does not prescribe the number of iterations to be performed but the scheme is applied until the computed solution satisfies a certain criterion with respect to its accuracy. For $k = 6$ and the stopping criterion $\left\| \mathbf{e}^{(m)} \right\|_\infty = \left\| \mathbf{u}^{(m)} \right\|_\infty < 10^{-6}$, the number of iterations are given in Table 3.1. One can see that refining the mesh once, thus halving the mesh width and doubling the number of unknowns, then the number of iterations increases by the factor four.

Table 3.1: Convergence of the damped Jacobi method with $\omega = 2/3$ for the initial iterate with wave number $k = 6$.

N	no. of iterations
16	27
32	116
64	475
128	1908
256	7642
512	30576
1024	122314

$\square$

Remark 3.9 *Analytical considerations of the damped Jacobi method.* The iteration matrix of the damped Jacobi method (3.8) has the form

$$S_{\mathrm{jac},\omega} = I - \omega D^{-1} A = I - \frac{\omega h}{2} A, \tag{3.11}$$

where the diagonal of the finite element system matrix has been inserted. The convergence of the damped Jacobi method is determined by the eigenvalues of the iteration matrix. From the special form of this matrix, one can see that

$$\lambda_k(S_{\mathrm{jac},\omega}) = 1 - \frac{\omega h}{2}\lambda_k(A) = 1 - 2\omega \sin^2\left(\frac{k\pi h}{2}\right), \quad k = 1, \ldots, N-1, \tag{3.12}$$

where (2.6) and $h = 1/N$ have been used. The eigenvectors $\mathbf{v}_k$ of A, see (2.7), are the same as the eigenvectors of $S_{\mathrm{jac},\omega}$, *exercise.* $\square$

Lemma 3.10 Convergence of the damped Jacobi method. *The damped Jacobi method converges for the one-dimensional model problem for all initial iterates if $\omega \in (0, 1]$. The method converges fastest for $\omega = 1$.*

Proof: From Numerical Mathematics II it is known that the method converges for all initial iterates if and only if the spectral radius of the iteration matrix $\rho(S_{\mathrm{jac},\omega})$ is smaller than 1. Since it is

$$0 < \sin^2\left(\frac{k\pi h}{2}\right) = \sin^2\left(\frac{k\pi}{2N}\right) < 1, \quad \text{for } k = 1, \ldots, N-1,$$

it follows from (3.12) that $\lambda_k(S_{\mathrm{jac},\omega}) \in (-1, 1)$ for $k = 1, \ldots, N-1$, and $\omega \in (0, 1]$. Hence it is $\rho(S_{\mathrm{jac},\omega}) < 1$.

It is also known from Numerical Mathematics II that the method converges the faster the smaller $\rho(S_{\text{jac},\omega})$ is, i.e., one has to solve (*exercise*)

$$\min_{\omega \in (0,1]} \max_{k=1,\ldots,N-1} \left| 1 - 2\omega \sin^2\left(\frac{k\pi}{2N}\right) \right|.$$

∎

Remark 3.11 *General effect of the damped Jacobi method on the discrete Fourier modes.* For studying the effect of the damped Jacobi methods on the discrete Fourier modes, see (3.10), an arbitrary initial error $\mathbf{e}^{(0)}$ will be represented with respect to the basis $\{\mathbf{w}_1, \ldots, \mathbf{w}_{N-1}\}$, where $\mathbf{w}_1, \ldots, \mathbf{w}_{N-1}$ are the eigenvectors of $S_{\text{jac},\omega}$ and A,

$$\mathbf{e}^{(0)} = \sum_{k=1}^{N-1} c_k \mathbf{w}_k, \quad c_k \in \mathbb{R}.$$

Since the damped Jacobi method can be written in form (3.3), it follows from (3.5) that

$$\mathbf{e}^{(m)} = S_{\text{jac},\omega}^m \mathbf{e}^{(0)}.$$

Using the property of $\mathbf{w}_k$ being an eigenvector of $S_{\text{jac},\omega}$, one obtains

$$\begin{aligned}
\mathbf{e}^{(m)} &= \sum_{k=1}^{N-1} c_k S_{\text{jac},\omega}^m \mathbf{w}_k = \sum_{k=1}^{N-1} c_k S_{\text{jac},\omega}^{m-1}\left(S_{\text{jac},\omega}\mathbf{w}_k\right) = \sum_{k=1}^{N-1} c_k S_{\text{jac},\omega}^{m-1}\lambda_k\left(S_{\text{jac},\omega}\right)\mathbf{w}_k \\
&= \ldots = \sum_{k=1}^{N-1} c_k \lambda_k^m\left(S_{\text{jac},\omega}\right)\mathbf{w}_k.
\end{aligned}$$

This calculation shows that after m iterations, the initial error with respect to the k-th discrete Fourier mode is reduced by the factor $\lambda_k^m\left(S_{\text{jac},\omega}\right)$. If $|\lambda_k^m\left(S_{\text{jac},\omega}\right)|$ is close to 1, then the reduction will be small. A strong reduction will occur if $|\lambda_k^m\left(S_{\text{jac},\omega}\right)|$ is close to zero. □

Remark 3.12 *Effect on the smooth error modes.* Using (3.12), one finds that

$$\lambda_k\left(S_{\text{jac},\omega}\right) \approx 1 \quad \Longleftrightarrow \quad \sin^2\left(\frac{k\pi}{2N}\right) \approx 0 \quad \Longleftrightarrow \quad k \text{ small},$$

$$\lambda_k\left(S_{\text{jac},\omega}\right) \approx -1 \quad \Longleftrightarrow \quad \omega \sin^2\left(\frac{k\pi}{2N}\right) \approx 1 \quad \Longleftrightarrow \quad \omega \approx 1 \text{ and } k \text{ close to } N.$$

For $\lambda_1\left(S_{\text{jac},\omega}\right)$, one finds with a Taylor series expansion

$$\lambda_1\left(S_{\text{jac},\omega}\right) = 1 - 2\omega \sin^2\left(\frac{\pi h}{2}\right) \approx 1 - 2\omega\frac{h^2\pi^2}{4} = 1 - \omega\frac{h^2\pi^2}{2}.$$

This eigenvalue is close to 1 for all damping parameters $\omega \in (0,1]$. Hence, there is no choice of the damping parameter which results in an efficient damping of the smooth error modes connected to $\mathbf{w}_1$. In addition, $\lambda_1\left(S_{\text{jac},\omega}\right)$ is the closer to 1 the finer the grid is. It follows that refining the grid results in a worse convergence with respect of the smooth error modes. □

Remark 3.13 *Effect on the oscillating error modes.* The distribution of the eigenvalues of the iteration matrix for $\omega \in \{1, 2/3, 1/2\}$ and $N = 16$ is presented in Figure 3.4. As it was observed in the previous remark, none of the damping parameters gives a method that reduces the smooth error modes efficiently. In can be seen in Figure 3.4, using the damping parameter $\omega = 1$, the method does not damp
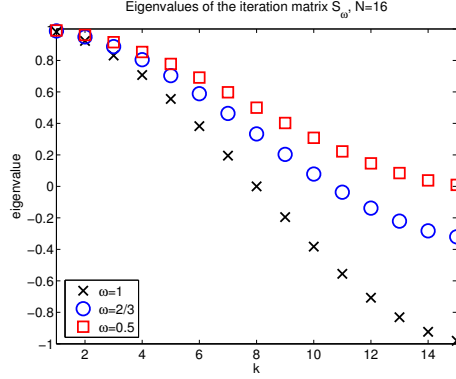
Figure 3.4: Eigenvalues of the iteration matrix $S_{\mathrm{jac},\omega}$ of the damped Jacobi method for different values of the damping parameter, $N = 16$.

efficiently the oscillating error modes neither, but it damps efficiently some intermediate error modes. The situation is much different for the damping parameter $\omega = 1/2$. For this parameter, it can be observed that the oscillating error modes are damped efficiently.

The situation as it occurs for $\omega = 1/2$ is of advantage, since it allows to distinguish clearly between the low and the high frequencies. With the damped Jacobi method and appropriate damping parameters, there is an iterative scheme that damps the high frequencies fast and the low frequencies slowly. Now, one needs another method with complementary properties to combine both methods. The construction of the complementary method is the goal of multigrid methods. □

Example 3.14 *Optimal damping parameter for the oscillating modes.* The damping parameter ω has to be determined such the one finds the smallest interval $[-\bar{\lambda}, \bar{\lambda}]$ with $\lambda_k\left(S_{\mathrm{jac},\omega}\right) \in [-\bar{\lambda}, \bar{\lambda}]$ for $k = N/2, \ldots, N-1$. This goal is achieved with $\omega = 2/3$. In this case it is, using the monotonicity of the sine function,

$$\frac{4}{3} \geq \frac{4}{3}\sin^2\left(\frac{k\pi}{2N}\right) \geq \frac{4}{3}\sin^2\left(\frac{N\pi}{4N}\right) = \frac{4}{3} \cdot \frac{1}{2} = \frac{2}{3}.$$

One gets

$$\max_{k \geq N/2}\left|\lambda_k\left(S_{2/3}\right)\right| = \max_{k \geq N/2}\left|1 - \frac{4}{3}\sin^2\left(\frac{k\pi}{2N}\right)\right| \leq \frac{1}{3},$$

see also Figure 3.4. It follows that the oscillating error modes are reduced in each iteration at least by the factor three. This damping rate for the oscillating error modes is called smoothing rate of the method. As one can see, the smoothing rate is for the damped Jacobi method (with fixed ω) independent of the fineness of the grid. □

Remark 3.15 *On the multigrid idea.* Consider a fixed Fourier mode $\sin(k\pi x)$ and its discrete representation $\sin(jk\pi/N)$, $j = 1, \ldots, N-1$. As already noted at the end of Remark 3.7, the classification of this mode depends on the fineness of the grid:

- If the grid is sufficiently fine, i.e., $k < N/2$, it is a smooth mode and it can be damped only slowly with the damped Jacobi method.
- If the grid is sufficiently coarse, i.e., $N/2 \leq k \leq N-1$, it is an oscillating mode and can be damped quickly with the damped Jacobi method.

From this observation, one can already derive the multigrid idea. On a fine grid, only the oscillating error modes on this grid are damped. The smooth modes on this grid are oscillating on coarser grids and they will be reduced on these grids. □

14

3.3 The Gauss–Seidel Method and the SOR Method

Remark 3.16 *The Gauss–Seidel Method and the SOR Method.* The Gauss–Seidel method and the SOR (successive over relaxation) method were also already introduced and studied in Numerical Mathematics II. Decompose the system matrix of (3.1) into

$$A = D + L + U,$$

where D is the diagonal, L is the strict lower part, and U is the strict upper part. The Gauss–Seidel method is obtained with $M = D + L$ and $N = -U$ in the fixed point method (3.3)

$$
\begin{aligned}
\mathbf{u}^{(m+1)} &= \underbrace{-\left(D+L\right)^{-1}U}_{S_{\mathrm{GS}}}\mathbf{u}^{(m)} + (D+L)^{-1}\mathbf{f} \\
&= \mathbf{u}^{(m)} + (D+L)^{-1}\left(\mathbf{f} - A\mathbf{u}^{(m)}\right) \quad m = 0,1,2,\dots.
\end{aligned}
$$

Using $M = \omega^{-1}D + L$ and $N = \omega^{-1}D - (D-U)$ gives the SOR method. This method can be written in the form

$$
\begin{aligned}
\mathbf{u}^{(m+1)} &= \mathbf{u}^{(m)} + \omega D^{-1}\left(\mathbf{f} - L\mathbf{u}^{(m+1)} - (D+U)\mathbf{u}^{(m)}\right) \\
&= \mathbf{u}^{(m)} + \left(\frac{D}{\omega} + L\right)^{-1}\left(\mathbf{f} - A\mathbf{u}^{(m)}\right), \quad m = 0,1,2,\dots.
\end{aligned}
$$

In the case $\omega = 1$, the Gauss–Seidel method is recovered. Writing the method component-wise

$$
u_i^{(m+1)} = u_i^{(m)} + \frac{\omega}{a_{ii}}\left(f_i - \sum_{j=1}^{i-1}a_{ij}u_j^{(m+1)} - \sum_{j=i}^{n}a_{ij}u_j^{(m)}\right),
$$

one can see that for the computation of $\mathbf{u}^{(m+1)}$ not only the old iterate $\mathbf{u}^{(m)}$ is used, as in the damped Jacobi method, but one uses the already computed components of $\mathbf{u}^{(m+1)}$.

By the last property, one can say that the SOR method is somewhat more advanced than the damped Jacobi method. However, it will turn out that the SOR method shows a similar behavior for the solution of the model problem as the damped Jacobi method. □

Remark 3.17 *Properties of the SOR method.* The properties of the SOR method were studied in Numerical Mathematics II. They will be summarized here.
- Lemma of Kahan[1]. If the SOR method converges for every initial iterates $\mathbf{u}^{(0)} \in \mathbb{R}^n$ then $\omega \in (0,2)$.
- If $A \in \mathbb{R}^{n\times n}$ a symmetric positive definite matrix. Then the SOR method converges for all initial iterates $\mathbf{u}^{(0)} \in \mathbb{R}^n$ if $\omega \in (0,2)$.

The rate of convergence depends on ω. It can be shown that for a certain class of matrices, to which also the matrix obtained in the discretization of the model problem belongs to, there is an optimal value $\omega_{\mathrm{opt}} \in (1,2)$. However, the determination of ω_{opt} is difficult in practice. For the model problem, one finds that ω_{opt} tends to 2 if the grids are refined, cf., Numerical Mathematics II, exercise problem 04/2.

The behavior of the SOR method depends on the numbering of the unknowns, which is in contrast to the damped Jacobi method. There are classes of problems where the efficiency of the SOR method depends essentially on the numbering of the unknowns. □

---

[1]William M. Kahan, born 1933

Example 3.18 *Application of the Gauss–Seidel method for the solution of the model problem.* The SOR method with $\omega = 1$ is studied in the same way as the damped Jacobi method in Example 3.8. The qualitatively behavior does not change for other values of the relaxation parameter. The numbering of the unknowns in the model problem is from left to right.

Figures 3.5 and 3.6 present the results. One can see that they are qualitatively the same as for the damped Jacobi method.
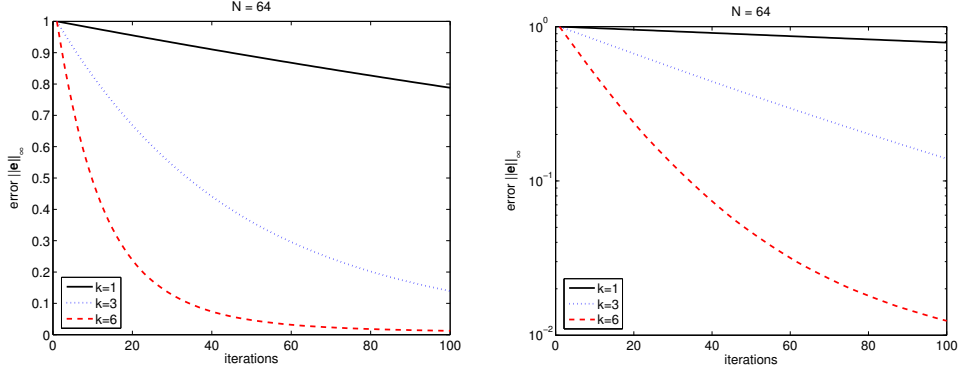

Figure 3.5: Convergence of the SOR method with $\omega = 1$ for initial iterates with different wave numbers on a fixed grid, left linear plot, right semilogarithmic plot.
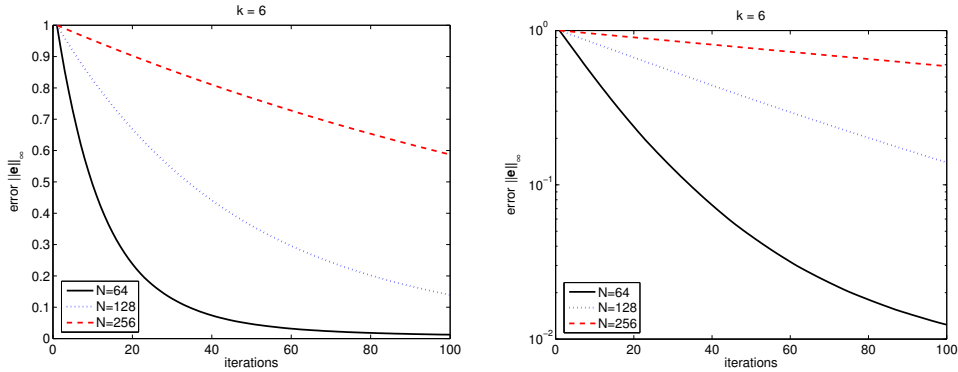
Figure 3.6: Convergence of the SOR method with $\omega = 1$ on different grids for an initial iterate with a fixed wave number, left linear plot, right semilogarithmic plot.

The number of iterations for $k = 6$ and the stopping criterion $\left\| \mathbf{e}^{(m)} \right\|_\infty = \left\| \mathbf{u}^{(m)} \right\|_\infty < 10^{-6}$ is presented in Table 3.2. Like for the Jacobi method, the number increases by the factor of four if the grid is refined once.

Altogether, one can draw for the SOR method the same conclusions as for the damped Jacobi method. □

Lemma 3.19 Some eigenvalues and eigenvectors of S_{GS}. *Let A be the matrix obtained by discretizing the model problem (2.1) with the finite element method. Then, some eigenvalues of the iteration matrix of the Gauss–Seidel method are given by*

$$\lambda_k \left(S_{\mathrm{GS}} \right) = \cos^2 \left(\frac{k\pi}{N} \right), \quad k = 1, \dots, N/2,$$

Table 3.2: Convergence of the SOR method with $\omega = 1$ for the initial iterate with wave number $k = 6$.

N	no. of iterations
16	274
32	1034
64	3859
128	14297
256	52595
512	191980
1024	–

and the corresponding eigenvectors are $\mathbf{w}_k = (w_{k,1}, \ldots, w_{k,N-1})^T$ *with*

$$w_{k,j} = (\lambda_k \, (S_{\mathrm{GS}}))^{j/2} \sin\left(\frac{jk\pi}{N}\right), \quad j = 1, \ldots, N-1.$$

Proof: One has to show that

$$S_{\mathrm{GS}} \mathbf{w}_k = \lambda_k \, (S_{\mathrm{GS}}) \, \mathbf{w}_k, \quad k = 1, \ldots, N-1.$$

Inserting the decomposition of S_{GS} gives

$$-(D+L)^{-1} U \mathbf{w}_k = \lambda_k \, (S_{\mathrm{GS}}) \, \mathbf{w}_k \quad \Longleftrightarrow \quad \lambda_k \, (S_{\mathrm{GS}}) \, (D+L) \, \mathbf{w}_k = -U \mathbf{w}_k.$$

Considering the j-th component and using the special form of the matrices D, L, and U, see Example 2.2, one obtains

$$\lambda_k \, (S_{\mathrm{GS}}) \left(\frac{2}{h} w_{k,j} - \frac{1}{h} w_{k,j-1}\right) = \frac{1}{h} w_{k,j+1}.$$

Scaling this equation by h and inserting the representation of the k-th eigenvector yields

$$\lambda_k \, (S_{\mathrm{GS}}) \left[2 \, (\lambda_k \, (S_{\mathrm{GS}}))^{j/2} \sin\left(\frac{jk\pi}{N}\right) - (\lambda_k \, (S_{\mathrm{GS}}))^{(j-1)/2} \sin\left(\frac{(j-1)k\pi}{N}\right) \right]$$
$$= (\lambda_k \, (S_{\mathrm{GS}}))^{(j+1)/2} \sin\left(\frac{(j+1)k\pi}{N}\right),$$

which is equivalent to

$$(\lambda_k \, (S_{\mathrm{GS}}))^{(j+1)/2} \left[2 \, (\lambda_k \, (S_{\mathrm{GS}}))^{1/2} \sin\left(\frac{jk\pi}{N}\right) - \sin\left(\frac{(j-1)k\pi}{N}\right) \right]$$
$$= (\lambda_k \, (S_{\mathrm{GS}}))^{(j+1)/2} \sin\left(\frac{(j+1)k\pi}{N}\right).$$

Applying the formula for the eigenvalues, noting that $\cos(k\pi/N) \geq 0$ for $k = 1, \ldots, N/2$, gives

$$(\lambda_k \, (S_{\mathrm{GS}}))^{(j+1)/2} \left[2 \cos\left(\frac{k\pi}{N}\right) \sin\left(\frac{jk\pi}{N}\right) - \sin\left(\frac{(j-1)k\pi}{N}\right) \right]$$
$$= (\lambda_k \, (S_{\mathrm{GS}}))^{(j+1)/2} \sin\left(\frac{(j+1)k\pi}{N}\right).$$

Using now the relation

$$2 \sin\left(\frac{\alpha + \beta}{2}\right) \cos\left(\frac{\alpha - \beta}{2}\right) = \sin\alpha + \sin\beta$$

with $\alpha = (j+1)k\pi/N$, $\beta = (j-1)k\pi/N$, then one finds that both sides are in fact identical.

17

For $j = 1$ and $j = N - 1$, one can perform the same calculation with formally introducing

$$w_{k,0} = (\lambda_k (S_{\mathrm{GS}}))^{0/2} \sin\left(\frac{0k\pi}{N}\right) = 0, \quad w_{k,N} = (\lambda_k (S_{\mathrm{GS}}))^{N/2} \sin\left(\frac{Nk\pi}{N}\right) = 0.$$

∎

Remark 3.20 *Discussion of Lemma 3.19.* The eigenvalues of S_{GS} are close to one for small k. They are close to zero if k is close to $N/2$. This situation is similar to the Jacobi method without damping.

One can derive, analogosly to the damped Jacobi method in Remark 3.11, the error formula

$$\mathbf{e}^{(m)} = \sum_{k=1}^{N-1} (\lambda_k (S_{\mathrm{GS}}))^m \mathbf{w}_k (S_{\mathrm{GS}}).$$

Using the eigenvectors $\mathbf{w}_k(S_{\mathrm{GS}})$ of S_{GS} as initial iterates, then it follows from the eigenvalues of S_{GS} that a fast error reduction can be expected only for $k \approx N/2$, whereas for k close to 0, there is only a slow convergence, see Table 3.3. It turns out that the situation for k being close to $N - 1$ is simular as for k being close to 0.

□

Table 3.3: Number of iterations for damping the norm of the error by the factor 100, $N = 64$.

k	$\mathbf{w}_k(S_{\mathrm{GS}})$
1	1895
3	207
6	50
16	6
32	1
60	115
63	1895

3.4 Summary

Remark 3.21 *Summary.* The investigation of classical iterative schemes led to the following important observations:

- Classical iterative schemes might damp highly oscillating discrete error modes very quickly. There is only a slow damping of the smooth discrete error modes.
- A smooth error mode on a given grid is generally on a coarser grid less smooth.

□

Chapter 4

Grid Transfer

Remark 4.1 *Contents of this chapter.* Consider a grid with grid size h and the corresponding linear system of equations

$$A^h \mathbf{u}^h = \mathbf{f}^h.$$

The summary given in Section 3.4 leads to the idea that there might be an iterative method for solving this system efficiently, which uses also coarser grids. In order to construct such a method, one needs mechanisms that transfer the information in an appropriate way between the grids. □

4.1 Algorithms with Coarse Grid Systems, the Residual Equation

Remark 4.2 *Basic idea for obtaining a good initial iterate with a coarse grid solution.* One approach for improving the behavior of iterative methods, at least at the beginning of the iteration, consists in using a good initial iterate. For the model problem, one can try to find a good initial iterate, e.g., by solving the problem approximately on a coarse grid, using only a few iterations. The application of only a few iterations is called smoothing, and the iterative method itself smoother, since only the oscillating error modes (on the coarse grid) are damped. The solution from the coarse grid can be used as initial iterate on the fine grid. □

Remark 4.3 *Study of the discrete Fourier modes on different grids.* Given a grid Ω^{2h}. In practice, a uniform refinement step consists in dividing in halves all intervals of Ω^{2h}, leading to the grid Ω^h. Then, the nodes of Ω^{2h} are the nodes of Ω^h with even numbers, see Figure 4.1.
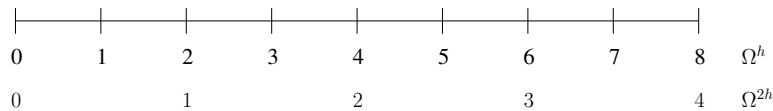

Figure 4.1: Coarse and fine grid.

Consider the k-th Fourier mode of the fine grid Ω^h. If $1 \le k \le N/2$, then it follows for the even nodes that

$$w_{k,2j}^h = \sin\left(\frac{2jk\pi}{N}\right) = \sin\left(\frac{jk\pi}{N/2}\right) = w_{k,j}^{2h}, \quad j = 1, \ldots, \frac{N}{2} - 1.$$

Hence, the k-th Fourier mode on Ω^h is the k-th Fourier mode on Ω^{2h}. From the definition of the smooth and oscillating modes, Remark 3.7, it follows that by going from the fine to the coarse grid, the k-th mode gets a higher frequency if $1 \le l \le N/2$. Note again that the notion of frequency depends on the grid size. The Fourier mode on Ω^h for $k = N/2$ is represented on Ω^{2h} by the zero vector.

For the transfer of the oscillating modes on Ω^h, i.e., for $N/2 < k < N$, one obtains a somewhat unexpected results. These modes are represented on Ω^{2h} as relatively smooth modes. The k-th mode on Ω^h becomes the negative of the $(N-k)$-th mode on Ω^{2h}:

$$
\begin{aligned}
w_{k,2j}^h &= \sin\left(\frac{2jk\pi}{N}\right) = \sin\left(\frac{jk\pi}{N/2}\right), \\
-w_{N-k,2j}^{2h} &= -\sin\left(\frac{j(N-k)\pi}{N/2}\right) = -\sin\left(\frac{2j(N-k)\pi}{N}\right) \\
&= -\sin\left(\frac{2jN\pi}{N} - \frac{2jk\pi}{N}\right) \\
&= -\underbrace{\sin(2j\pi)}_{=0}\cos\left(\frac{2jk\pi}{N}\right) + \underbrace{\cos(2j\pi)}_{=1}\sin\left(\frac{2jk\pi}{N}\right) \\
&= \sin\left(\frac{2jk\pi}{N}\right),
\end{aligned}
$$

i.e., $w_{k,2j}^h = -w_{N-k,2j}^{2h}$. This aspect shows that it is necessary to damp the oscillating error modes on Ω^h before a problem on Ω^{2h} is considered. Otherwise, one would get additional smooth error modes on the coarser grid. □

Remark 4.4 *The residual equation.* An iterative method for the solution of $A\mathbf{u} = \mathbf{f}$ can be applied either directly to this equation or to an equation for the error, the so-called residual equation. Let $\mathbf{u}^{(m)}$ be an approximation of $\mathbf{u}$, then the error $\mathbf{e}^{(m)} = \mathbf{u} - \mathbf{u}^{(m)}$ satisfies the equation

$$
A\mathbf{e}^{(m)} = \mathbf{f} - A\mathbf{u}^{(m)} =: \mathbf{r}^{(m)}. \tag{4.1}
$$

□

Remark 4.5 *Nested iteration.* This remark gives a first strategy for using coarse grid problems for the improvement of an iterative method for solving $A\mathbf{u}^h = \mathbf{f}^h$. This strategy is a generalization of the idea from Remark 4.2. It is called nested iteration:

- solve $A^{h_0}\mathbf{u}^{h_0} = \mathbf{f}^{h_0}$ on a very coarse grid approximately by applying a smoother,
- $\vdots$
- smooth $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$ on Ω^{2h},
- solve $A^h\mathbf{u}^h = \mathbf{f}^h$ on Ω^h by an iterative method with the initial iterate provided from the coarser grids.

However, there are some open questions with this strategy. How are the linear systems defined on the coarser grids? What can be done if there are still smooth error modes on the finest grid? In this case, the convergence of the last step will be slowly. □

Remark 4.6 *Coarse grid correction, two-level method.* A second strategy uses also the residual equation (4.1):

- Smooth $A^h\mathbf{u}^h = \mathbf{f}^h$ on Ω^h. This step gives an approximation $\mathbf{v}^h$ of the solution which still has to be updated appropriately. Compute the residual $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{v}^h$.

- Project (restrict) the residual to Ω^{2h}. The result is called $R(\mathbf{r}^h)$.
- Solve $A^{2h}\mathbf{e}^{2h} = R(\mathbf{r}^h)$ on Ω^{2h}. With this step, one obtains an approximation $\mathbf{e}^{2h}$ of the error.
- Project (prolongate) $\mathbf{e}^{2h}$ to Ω^h. The result is denoted by $P(\mathbf{e}^{2h})$.
- Update the approximation of the solution on Ω^h by $\mathbf{v}^h := \mathbf{v}^h + P(\mathbf{e}^{2h})$.

This approach is called coarse grid correction or two-level method. With this approach, one computes on Ω^{2h} an approximation of the error. However, also for this approach one has to answer some questions. How to define the system on the coarse grid? How to restrict the residual to the coarse grid and how to prolongate the correction to the fine grid? $\qquad\square$

4.2 Prolongation or Interpolation

Remark 4.7 *General remarks.* The transfer from the coarse to the fine grid is called prolongation or interpolation. In many situations, one can use the simplest approach, which is the linear interpolation. For this reason, this section will only consider this approach. $\qquad\square$

Example 4.8 *Linear interpolation for finite difference methods.* For finite difference methods, the prolongation operator is defined by a local averaging. Let Ω^{2h} be divided into $N/2$ intervals and Ω^h into N intervals. The node j on Ω^{2h} corresponds to the node $2j$ on Ω^h, $0 \le j \le N/2$, see Figure 4.1. Let $\mathbf{v}^{2h}$ be given on Ω^{2h}. Then, the linear interpolation

$$I_{2h}^h \; : \; \mathbb{R}^{N/2-1} \to \mathbb{R}^{N-1}, \quad \mathbf{v}^h = I_{2h}^h \mathbf{v}^{2h}$$

is given by

$$\begin{aligned}
v_{2j}^h &= v_j^{2h}, & j &= 1, \ldots, N/2 - 1, \\
v_{2j+1}^h &= \frac{1}{2}\left(v_j^{2h} + v_{j+1}^{2h}\right), & j &= 0, \ldots, N/2 - 1,
\end{aligned} \tag{4.2}$$

see Figure 4.2. For even nodes of Ω^h, one takes directly the value of the corresponding node of Ω^{2h}. For odd nodes of Ω^h, the arithmetic mean of the values of the neighbor nodes is computed.
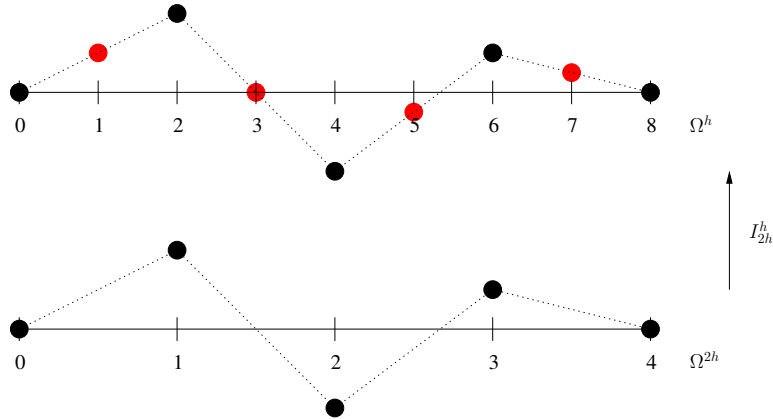

Figure 4.2: Linear interpolation for finite difference methods.

The linear prolongation is a linear operator, see below Lemma 4.10, between two finite-dimensional spaces. Hence, it can be represented as a matrix. Using the

standard basis of $\mathbb{R}^{N/2-1}$ and $\mathbb{R}^{N-1}$, then

$$
I_{2h}^h = \frac{1}{2}
\begin{pmatrix}
1 & & & & \\
2 & & & & \\
1 & 1 & & & \\
  & 2 & & & \\
  & 1 & 1 & & \\
  &   & 2 & \ddots & \\
  &   & 1 & \ddots & \\
  &   &   & & 1 \\
  &   &   & & 2 \\
  &   &   & & 1
\end{pmatrix}
\in \mathbb{R}^{(N-1)\times(N/2-1)}.
\tag{4.3}
$$

$\square$

Example 4.9 *Canonical prolongation for finite element methods.* Consider conforming finite element methods and denote the spaces on Ω^{2h} and Ω^h with V^{2h} and V^h, respectively. Because Ω^h is a uniform refinement of Ω^{2h}, it follows that $V^{2h} \subset V^h$. Hence, each finite element function defined on Ω^{2h} is contained in the space V^h. This aspect defines a canonical prolongation

$$
\mathcal{I}_{2h}^h \ : \ V^{2h} \to V^h, \quad v^{2h} \mapsto v^{2h}.
$$

The canonical prolongation will be discussed in detail for P_1 finite elements. Let $\{\varphi_i^{2h}\}_{i=1}^{N/2-1}$ be the local basis of V^{2h} and $\{\varphi_i^h\}_{i=1}^{N-1}$ be the local basis of V^h. Each function $v^{2h} \in V^{2h}$ has a representation of the form

$$
v^{2h}(x) = \sum_{i=1}^{N/2-1} v_i^{2h}\varphi_i^{2h}(x), \quad v_i^{2h} \in \mathbb{R}, \ i=1,\ldots,N/2-1.
$$

There is a bijection between V^{2h} and $\mathbb{R}^{N/2-1}$.

Let $j = 2i$ be the corresponding index on Ω^h to the index i on Ω^{2h}. From the property of the local basis, it follows that

$$
\varphi_i^{2h} = \frac{1}{2}\varphi_{j-1}^h + \varphi_j^h + \frac{1}{2}\varphi_{j+1}^h.
$$

Inserting this representation gives

$$
\begin{aligned}
v^{2h}(x) &= \sum_{i=1}^{N/2-1} v_i^{2h}\left(\frac{1}{2}\varphi_{2i-1}^h + \varphi_{2i}^h + \frac{1}{2}\varphi_{2i+1}^h\right) \\
&= v_1^{2h}\left(\frac{1}{2}\varphi_1^h + \varphi_2^h + \frac{1}{2}\varphi_3^h\right) \\
&\quad + v_2^{2h}\left(\frac{1}{2}\varphi_3^h + \varphi_4^h + \frac{1}{2}\varphi_5^h\right) \\
&\quad + v_3^{2h}\left(\frac{1}{2}\varphi_5^h + \varphi_6^h + \frac{1}{2}\varphi_7^h\right) \\
&\quad + \ldots.
\end{aligned}
$$

From this formula, one can see that the representation in the basis of V^h is of the following form. For basis functions that correspond to nodes which are already on Ω^{2h} (even indices on the fine grid), the coefficient is the same as for the basis function on the coarser grids. For basis functions that correspond to new nodes, the

coefficient is the arithmetic mean of the coefficients of the neighbor basis functions. Hence, if local bases are used, the coefficients for the prolongated finite element function can be computed by multiplying the coefficients of the coarse grid finite element function with the matrix (4.3). □

Lemma 4.10 Properties of the linear interpolation operator. *The operator* $I_{2h}^h : \mathbb{R}^{N/2-1} \to \mathbb{R}^{N-1}$ *defined in (4.2) is a linear operator. It has full rank and only the trivial kernel.*

Proof: *i)* I_{2h}^h *is a linear operator.* The operator is homogeneous, since for $\alpha \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^{N/2-1}$ it is

$$
\begin{aligned}
v_{2j}^h &= (\alpha \mathbf{v})_j = \alpha v_j, \\
v_{2j+1}^h &= \frac{1}{2}\left((\alpha \mathbf{v})_j + (\alpha \mathbf{v})_{j+1}\right) = \alpha \frac{1}{2}\left(v_j + v_{j+1}\right).
\end{aligned}
$$

The operator is additive. Let $\mathbf{v}, \mathbf{w} \in \mathbb{R}^{N/2-1}$, then

$$
\begin{aligned}
\left(I_{2h}^h(\mathbf{v}+\mathbf{w})\right)_{2j} &= (\mathbf{v}+\mathbf{w})_j = v_j + w_j = \left(I_{2h}^h(\mathbf{v})\right)_{2j} + \left(I_{2h}^h(\mathbf{w})\right)_{2j}, \\
\left(I_{2h}^h(\mathbf{v}+\mathbf{w})\right)_{2j+1} &= \frac{1}{2}((\mathbf{v}+\mathbf{w})_j + (\mathbf{v}+\mathbf{w})_{j+1}) = \frac{1}{2}(v_j + v_{j+1}) + \frac{1}{2}(w_j + w_{j+1}) \\
&= \left(I_{2h}^h(\mathbf{v})\right)_{2j+1} + \left(I_{2h}^h(\mathbf{w})\right)_{2j+1}.
\end{aligned}
$$

An homogeneous and additive operator is linear.

ii) I_{2h}^h *has full rank and trivial kernel.* Since $N/2 - 1 < N - 1$, both properties are equivalent. Let $\mathbf{0} = \mathbf{v}^h = I_{2h}^h(\mathbf{v}^{2h})$. From (4.2) it follows from the vanishing of the even indices of $\mathbf{v}^h$ immediately that $v_j^{2h} = 0$, $j = 1, \ldots, N/2 - 1$, i.e., $\mathbf{v}^{2h} = \mathbf{0}$. Hence, the only element in the kernel of I_{2h}^h is the zero vector. ∎

Remark 4.11 *Effect of the prolongation on different error modes.* Assume that the error, which is of course unknown is a smooth function on the fine grid Ω^h. In addition, the coarse grid approximation on Ω^{2h} is computed and it should be exact in the nodes of the coarse grid. The interpolation of this coarse grid approximation is a smooth function on the fine grid (there are no new oscillations). For this reason, one can expect a rather good approximation of the smooth error on the fine grid.

If the error on the fine grid is oscillating, then each interpolation of a coarse grid approximation to the fine grid is a smooth function and one cannot expect that the error on the fine grid is approximated well, see Figure 4.3.

Altogether, the prolongation gives the best results, if the error on the fine grid is smooth. Hence, the prolongation is an appropriate complement to the smoother, which works most efficiently if the error is oscillating. □

4.3   Restriction

Remark 4.12 *General remarks.* For the two-level method, one has to transfer the residual from Ω^h to Ω^{2h} before the coarse grid equation can be solved. This transfer is called restriction. □

Example 4.13 *Injection for finite difference schemes.* The simplest restriction is the injection. It is defined by

$$
I_h^{2h} : \mathbb{R}^{N-1} \to \mathbb{R}^{N/2-1}, \quad \mathbf{v}^{2h} = I_h^{2h}\mathbf{v}^h, \quad v_j^{2h} = v_{2j}^h, \quad j = 1, \ldots, \frac{N}{2} - 1,
$$

see Figure 4.4. For this restriction, one takes for each node on the coarse grid simply the value of the grid function at the corresponding node on the fine grid.
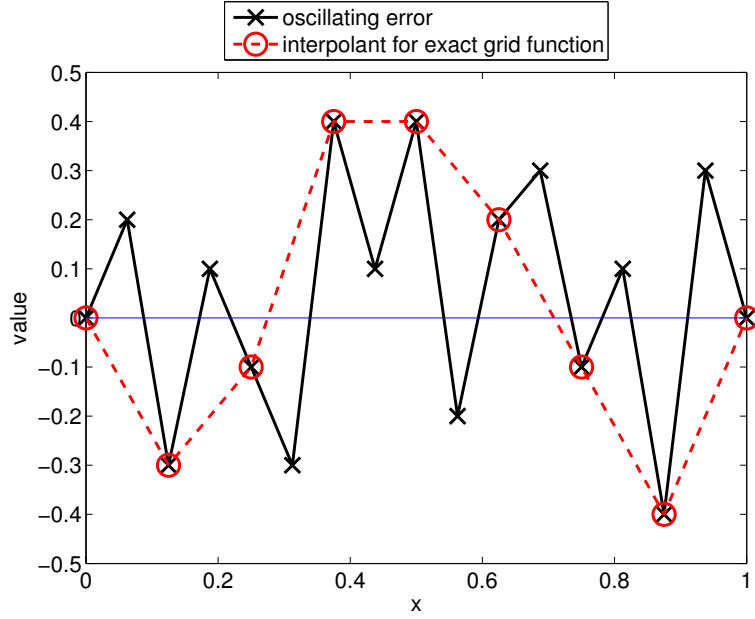
Figure 4.3: Oscillating error and interpolation.
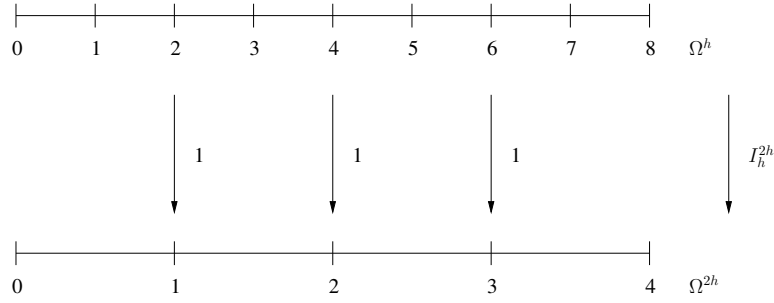
Figure 4.4: Injection.

It turns out that the injection does not lead to an efficient method. If one ignores every other node on Ω^h, then the values of the residual in these nodes, and with that also the error in these nodes, do not possess any impact on the system on the coarse grid. Consequently, these errors will generally not be corrected. □

Example 4.14 *Weighted restriction for finite difference schemes.* The weighted restriction uses all nodes on the fine grid. It is defined by an appropriate averaging

$$I_h^{2h} \quad : \quad \mathbb{R}^{N-1} \to \mathbb{R}^{N/2-1},$$
$$\mathbf{v}^{2h} \quad = \quad I_h^{2h}\mathbf{v}^h, \quad v_j^{2h} = \frac{1}{4}\left(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h\right), \quad j = 1, \dots, \frac{N}{2} - 1, \text{ (4.4)}$$

see Figure 4.5. For finite difference schemes, only the weighted restriction will be considered in the following.

If the spaces $\mathbb{R}^{N-1}$ and $\mathbb{R}^{N/2-1}$ are equipped with the standard bases, the matrix

Figure 4.5: Weighted restriction.

representation of the weighted restriction operator has the form

$$I_h^{2h} = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 & & & & \\ & 1 & 2 & 1 & & & \\ & & 1 & 2 & 1 & & \\ & & & & \ddots & & \\ & & & & 1 & 2 & 1 \end{pmatrix} \in \mathbb{R}^{(N/2-1)\times(N-1)}. \qquad (4.5)$$

With this representation, one can see an important connection between weighted restriction I_h^{2h} and interpolation I_{2h}^h:
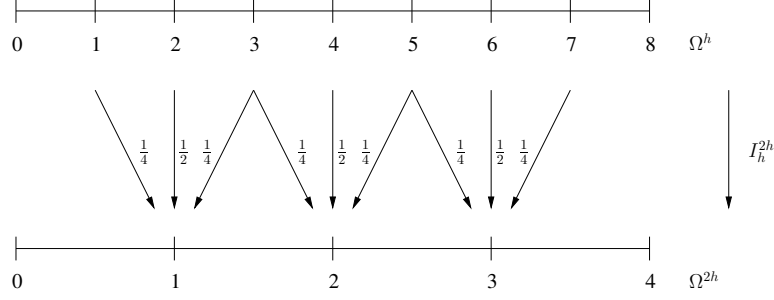
$$I_{2h}^h = 2\left(I_h^{2h}\right)^T.$$

$\square$

Lemma 4.15 Properties of the weighted restriction operator. *Let the restriction operator I_h^{2h} given by (4.4). This operator is linear. The rank of this operator is $N/2 - 1$ and the kernel has dimension $N/2$.*

    Proof: *i) Linearity. exercise.*

    ii) Rank and kernel. From linear algebra, it is known that the sum of the dimension of the kernel and the rank is $N - 1$. The rank of I_h^{2h} is equal to the dimension of its range (row rank). The range of I_h^{2h} is equal to $\mathbb{R}^{N/2-1}$, since every vector from $\mathbb{R}^{N/2-1}$ might be the image in the space of grid functions of Ω^{2h} of a vector corresponding to grid functions of Ω^h. Hence, the rank is $N/2 - 1$ and consequently, the dimension of the kernel is $N - 1 - (N/2 - 1) = N/2$. $\blacksquare$

Example 4.16 *Canonical restriction for finite element schemes.* Whereas for finite difference methods, one works only with vectors of real numbers, finite element methods are imbedded into the Hilbert space setting. In this setting, a finite element function is, e.g, from the space V^h, but the residual, which is the right-hand side minus the finite element operator applied to a finite element function (current iterate) is from the dual space $\left(V^h\right)^*$ of V^h. In this setting, it makes a difference if one restricts an element from V^h or from its dual space.

    For restricting a finite element function from V^h to V^{2h}, one can take the analogon of the weighted restriction. If local bases are used, then the coefficients of the finite element function from V^h are multiplied with the matrix (4.5) to get the coefficients of the finite element function in V^{2h}.

    In the two-level method, one has to restrict the residual, i.e., one needs a restriction from $\left(V^h\right)^*$ to $\left(V^{2h}\right)^*$. In this situation, a natural choice consists in using the dual prolongation operator, i.e.,

$$\mathcal{I}_h^{2h} \; : \; \left(V^h\right)^* \to \left(V^{2h}\right)^*, \quad \mathcal{I}_h^{2h} = \left(\mathcal{I}_{2h}^h\right)^*.$$

The dual operator is defined by

$$\left\langle \mathcal{I}_{2h}^h v^{2h}, r^h \right\rangle_{V^h,(V^h)^*} = \left\langle v^{2h}, \mathcal{I}_h^{2h} r^h \right\rangle_{V^{2h},(V^{2h})^*} \quad \forall\, v^{2h} \in V^{2h}, r^h \in \left(V^h\right)^*.$$

Thus, if local bases and the bijection between finite element spaces and the Euclidean spaces are used, then the restriction of the residual can be represented by the transposed of the matrix (4.3). This issue makes a difference of a factor of 2 compared with the matrix for the weighted restriction. □

Chapter 5

The Two-Level Method

Remark 5.1 *The two-level method.* In this chapter, the two-level method or coarse grid correction scheme will be analyzed. The two-level method, whose principle was already introduced in Remark 4.6, has the following form:

- Smooth $A^h \mathbf{u}^h = \mathbf{f}^h$ on Ω^h with some steps of a simple iterative scheme. This procedure gives an approximation $\mathbf{v}^h$. Compute the residual $\mathbf{r}^h = \mathbf{f}^h - A^h \mathbf{v}^h$.
- Restrict the residual to the coarse grid Ω^{2h} using the restriction operator I_h^{2h} (weighted restriction for finite difference methods, canonical restriction for finite element methods).
- Solve the coarse grid equation

$$A^{2h} \mathbf{e}^{2h} = I_h^{2h} \left(\mathbf{r}^h \right) \tag{5.1}$$

  on Ω^{2h}.
- Prolongate $\mathbf{e}^{2h}$ to Ω^h using the prolongation operator I_{2h}^h.
- Update $\mathbf{v}^h := \mathbf{v}^h + I_{2h}^h \left(\mathbf{e}^{2h} \right)$.

After the update, one can apply once more some iterations with the smoother. This step is called post smoothing, whereas the first step of the two-level method is called pre smoothing. $\qquad \square$

5.1 The Coarse Grid Problem

Remark 5.2 *The coarse grid system.* The two-level method still lacks a definition of the coarse grid matrix A^{2h}. This matrix should be a "Ω^{2h} version of the fine grid matrix A^h". Possible choices of A^{2h} will be discussed in this section. $\qquad \square$

Remark 5.3 *Definition of the coarse grid matrix by using a discrete scheme on* Ω^{2h}. A straightforward approach consists in defining A^{2h} by applying a finite difference or finite element method to the differential operator on Ω^{2h}. $\qquad \square$

Remark 5.4 *Definition of the coarse grid matrix by Galerkin projection.* Starting point for the derivation of an appropriate coarse grid matrix by the Galerkin projection is the residual equation

$$A^h \mathbf{e}^h = \mathbf{r}^h. \tag{5.2}$$

It will be assumed for the moment that $\mathbf{e}^h$ lies in the range of the prolongation operator I_{2h}^h. Then, there is a vector $\mathbf{e}^{2h}$ defined on the coarse grid such that

$$\mathbf{e}^h = I_{2h}^h \left(\mathbf{e}^{2h} \right).$$

Substituting this equation into (5.2) gives

$$A^h I_{2h}^h \left(\mathbf{e}^{2h} \right) = \mathbf{r}^h.$$

Applying now on both sides of this equation the restriction operator gives

$$I_h^{2h} A^h I_{2h}^h \left(\mathbf{e}^{2h} \right) = I_h^{2h} \mathbf{r}^h.$$

Comparing this definition with (5.1) leads to the definition

$$A^{2h} := I_h^{2h} A^h I_{2h}^h. \tag{5.3}$$

This definition of the coarse grid matrix is called Galerkin projection.

The derivation of (5.3) was based on the assumption that the error $\mathbf{e}^h$ is in the range of the prolongation. This property is in general not given. If it would be true, then an exact solution of the coarse grid equation would result in obtaining the solution of $A\mathbf{u}^h = \mathbf{f}^h$ with one step of the coarse grid correction scheme. Nevertheless, this derivation gives a motivation for defining A^{2h} in the form (5.3). □

Remark 5.5 *Matrix representation of the Galerkin projection.* For all operators on the right-hand side of (5.3), matrix representations are known, e.g., see (2.3), (4.3), and (4.5) for the case of the finite difference discretization. Using these

representations, one obtains

$$A^{2h}$$

$$= \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 & & & & \\ & 1 & 2 & 1 & & & \\ & & 1 & 2 & 1 & & \\ & & & & \ddots & & \\ & & & & 1 & 2 & 1 \end{pmatrix} \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \vdots & \vdots & \vdots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

$$\times \frac{1}{2} \begin{pmatrix} 1 & & & \\ 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & 1 & \\ & & 2 & \ddots \\ & & 1 & \ddots \\ & & & 1 \\ & & & 2 \\ & & & 1 \end{pmatrix}$$

$$= \frac{1}{8h^2} \begin{pmatrix} 1 & 2 & 1 & & & & \\ & 1 & 2 & 1 & & & \\ & & 1 & 2 & 1 & & \\ & & & & \ddots & & \\ & & & & 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & & & & & \\ 2 & -1 & 0 & & & & & \\ 0 & 0 & 0 & & & & & \\ -1 & 2 & -1 & & & & & \\ & & & \ddots & & & & \\ & & & & -1 & 2 & -1 \\ & & & & 0 & 0 & 0 \\ & & & & 0 & -1 & 2 \\ & & & & 0 & 0 & 0 \end{pmatrix}$$

$$= \frac{1}{8h^2} \begin{pmatrix} 4 & -2 & & & \\ -2 & 4 & -2 & & \\ & \vdots & \vdots & \vdots & \\ & & -2 & 4 & -2 \\ & & & -2 & 4 \end{pmatrix}$$

$$= \frac{1}{4h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \vdots & \vdots & \vdots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

This matrix has the form of the matrix (2.3) with h replaced by $2h$. Thus, in the case of the model problem, the matrix defined by the Galerkin projection (5.3) and the matrix (2.3) obtained by discretizing the differential operator on the coarse grid Ω^{2h} coincide.

In the finite element case, the matrices differ only by the factors in front of the parentheses: $1/2$, $1/h$, $1/2$, instead of $1/4$, $1/h^2$, $1/2$. Then, the final factor is $1/(2h)$ instead of $1/(4h^2)$. The factor $1/(2h)$ is exactly the factor of the finite element matrix on Ω^{2h}, see (2.4). That means, also in this case Galerkin projection and the discretization on Ω^{2h} coincide.

This connection of the Galerkin projection and of the discretized problem on Ω^{2h} does not hold in all cases (problems and discretizations), but it can be found often. $\qquad \square$

5.2   General Approach for Proving the Convergence of the Two-Level Method

Remark 5.6 *The iteration matrix of the two-level method.* For studying the convergence of the two-level method, one first has to find the iteration matrix $S_{2\mathrm{lev}}$ of this scheme. For simplicity, only the case of pre smoothing is considered, but no post smoothing.

Let S_{sm} be the iteration matrix of the smoother. The approximation of the solution before the pre smoothing step is denoted by $\mathbf{v}^{(n)}$ and the result after the update will be $\mathbf{v}^{(n+1)}$. Applying ν pre smoothing steps, then it is known from (3.7) that

$$\mathbf{e}^{(\nu)} = S_{\mathrm{sm}}^{\nu}\mathbf{e}^{(0)}, \quad \text{with } \mathbf{e}^{(0)} = \mathbf{u} - \mathbf{v}^{(n)}, \ \mathbf{e}^{(\nu)} = \mathbf{u} - \mathbf{v}_{\nu}^{(n)}.$$

It follows that

$$\mathbf{v}_{\nu}^{(n)} = \mathbf{u} - S_{\mathrm{sm}}^{\nu}\left(\mathbf{u} - \mathbf{v}^{(n)}\right),$$

where now $\mathbf{v}_{\nu}^{(n)}$ stands for $\mathbf{v}^h$ in the general description of the two-level method from Remark 5.1. It follows that

$$\mathbf{r} = \mathbf{f} - A^h\mathbf{v}_{\nu}^{(n)} = \mathbf{f} - A^h\mathbf{u} + A^h S_{\mathrm{sm}}^{\nu}\left(\mathbf{u} - \mathbf{v}^{(n)}\right) = A^h S_{\mathrm{sm}}^{\nu}\left(\mathbf{u} - \mathbf{v}^{(n)}\right).$$

Applying this formula in the two-level method from Remark 5.1, starting with the update step, one obtains

$$\begin{aligned}
\mathbf{v}^{(n+1)} &= \mathbf{v}_{\nu}^{(n)} + I_{2h}^h\left(\mathbf{e}^{2h}\right) \\
&= \mathbf{u} - S_{\mathrm{sm}}^{\nu}\left(\mathbf{u} - \mathbf{v}^{(n)}\right) + I_{2h}^h\left(A^{2h}\right)^{-1}I_h^{2h}\mathbf{r} \\
&= S_{\mathrm{sm}}^{\nu}\mathbf{v}^{(n)} + (I - S_{\mathrm{sm}}^{\nu})\left(A^h\right)^{-1}\mathbf{f} \\
&\quad + I_{2h}^h\left(A^{2h}\right)^{-1}I_h^{2h}A^h S_{\mathrm{sm}}^{\nu}\left(\left(A^h\right)^{-1}\mathbf{f} - \mathbf{v}^{(n)}\right) \\
&= \left(I - I_{2h}^h\left(A^{2h}\right)^{-1}I_h^{2h}A^h\right)S_{\mathrm{sm}}^{\nu}\mathbf{v}^{(n)} \qquad (5.4) \\
&\quad + \left((I - S_{\mathrm{sm}}^{\nu}) + I_{2h}^h\left(A^{2h}\right)^{-1}I_h^{2h}A^h S_{\mathrm{sm}}^{\nu}\right)\left(A^h\right)^{-1}\mathbf{f}.
\end{aligned}$$

Hence, the iteration matrix of the two-level method is given by

$$S_{2\mathrm{lev}} = \left(I - I_{2h}^h\left(A^{2h}\right)^{-1}I_h^{2h}A^h\right)S_{\mathrm{sm}}^{\nu}. \qquad (5.5)$$

Inserting $\mathbf{u} = \left(A^h\right)^{-1}\mathbf{f}$ into the two-level method (5.4) shows that $\mathbf{u}$ is a fixed point, *exercise*. It follows that in the case this fixed point is the only fixed point and that the two-level method converges, then it converges to $\mathbf{u}$. □

Remark 5.7 *Goal of the convergence analysis.* From the course Numerical Mathematics II, Theorem 3.3 in the part on iterative solvers, it is known that a sufficient and necessary condition for the convergence of the fixed point iteration is that $\rho\left(S_{2\mathrm{lev}}\right) < 1$. But the computation of $\rho\left(S_{2\mathrm{lev}}\right)$ is rather complicated, even in simple situations. However, from linear algebra it is known that $\rho\left(S_{2\mathrm{lev}}\right) \leq |||S_{2\mathrm{lev}}|||$ for induced matrix norms, e.g., the spectral norm. The goal of the convergence analysis will be to show that

$$|||S_{2\mathrm{lev}}||| \leq \rho < 1$$

independently of h. The analysis is based on a splitting of $S_{2\mathrm{lev}}$ in the form

$$S_{2\mathrm{lev}} = \left(\left(A^h\right)^{-1} - I_{2h}^h\left(A^{2h}\right)^{-1}I_h^{2h}\right)A^h S_{\mathrm{sm}}^{\nu}.$$

It follows that

$$|||S_{2\text{lev}}||| \leq ||| \left(A^h\right)^{-1} - I_{2h}^h \left(A^{2h}\right)^{-1} I_h^{2h}||| \; |||A^h S_{\text{sm}}^\nu|||. \qquad (5.6)$$

The first factor in (5.6) describes the effect of the coarse grid approximation. The second factor measures the efficiency of the smoothing step. The smaller the first factor is, the better is the coarse grid solution which approximates $\mathbf{e}^h$. Hence, the two essential components of the two-level method, the smoothing and the coarse grid correction, can be analyzed separately. $\qquad \square$

Definition 5.8 Smoothing property. The matrix S_{sm} is said to possess the smoothing property, if there exist functions $\eta(\nu)$ and $\overline{\nu}(t)$, whose definition is independent of h, and a number $\alpha > 0$ such that

$$|||A^h S_{\text{sm}}^\nu||| \leq \eta(\nu)h^{-\alpha} \quad \text{for all } 1 \leq \nu \leq \overline{\nu}(h), \qquad (5.7)$$

with $\eta(\nu) \to 0$ as $\nu \to \infty$ and $\overline{\nu}(h) = \infty$ or $\overline{\nu}(h) \to \infty$ as $h \to 0$. $\qquad \square$

Remark 5.9 *On the smoothing property.* The smoothing property does not necessarily mean that the smoothing iteration is a convergent iteration. It is only required that the error is smoothed in a certain way using up to $\overline{\nu}(h)$ smoothing steps. In fact, there are examples where divergent iterative schemes are good smoothers. But in this course, only the case $\overline{\nu}(h) = \infty$ will be considered, i.e., the case of a convergent smoothing iteration. $\qquad \square$

Definition 5.10 Approximation property. The approximation property holds if there is a constant C_a, which is independent of h, such that

$$||| \left(A^h\right)^{-1} - I_{2h}^h \left(A^{2h}\right)^{-1} I_h^{2h}||| \leq C_a h^\alpha \qquad (5.8)$$

with the same α as in the smoothing property. $\qquad \square$

Theorem 5.11 Convergence of the two-level method. *Suppose the smoothing property and the approximation property hold. Let $\rho > 0$ be a fixed number. If $\overline{\nu}(t) = \infty$ for all t, then there is a number $\underline{\nu}$ such that*

$$|||S_{2\text{lev}}||| \leq C_a \eta(\nu) \leq \rho, \qquad (5.9)$$

whenever $\nu \geq \underline{\nu}$.

    Proof: From (5.6) one obtains with the approximation property (5.8) and the smoothing property (5.7)

$$|||S_{2\text{lev}}||| \leq C_a h^\alpha \; \eta(\nu)h^{-\alpha} = C_a \eta(\nu).$$

Since $\eta(\nu) \to 0$ as $\nu \to \infty$, the right-hand side of this estimate is smaller than any given $\rho > 0$ if ν is sufficiently large, e.g., if $\nu \geq \underline{\nu}$. $\qquad \blacksquare$

Remark 5.12 *On the convergence theorem.* Note that the estimate $C_a \eta(\nu)$ is independent of h. The convergence theorems says that the two-level method converges with a rate that is independent of h if sufficiently many smoothing steps are applied. For many problems, one finds that only a few pre smoothing steps, i.e., 1 to 3, are sufficient for convergence. $\qquad \square$

5.3 The Smoothing Property of the Damped Jacobi Iteration

Remark 5.13 *Contents of this section.* In this section, the smoothing property of the damped Jacobi iteration for the model problem will be proved. Therefore, one has to estimate $|||A^h S^\nu_{\text{jac},\omega}|||$, where now the spectral matrix norm $\left\|A^h S^\nu_{\text{jac},\omega}\right\|_2$ is considered. In the proof, one has to estimate a term of the form $\left\|B(I-B)^\nu\right\|_2$ for some symmetric positive definite matrix with $0 < B \le I$, i.e., it is for all eigenvalues λ of B that $\lambda \in (0,1]$. □

Lemma 5.14 Estimate for a symmetric positive definite matrix. *Let* $0 < B = B^T \le I$, *then*

$$\left\|B(I-B)^\nu\right\|_2 \le \eta_0(\nu)$$

with

$$\eta_0(\nu) = \frac{\nu^\nu}{(\nu+1)^{\nu+1}}, \quad \nu \in \mathbb{N}. \tag{5.10}$$

Proof: The matrix $B(I-B)^\nu$ is symmetric, *exercise.*

First, the eigenvalues of $B(I-B)^\nu$ will be computed. Let $\lambda \in \mathbb{R}_+$ be an eigenvalue of B. It will be shown that $\lambda(1-\lambda)^\nu$ is an eigenvalue of $B(I-B)^\nu$. The proof is performed by induction Let $\nu = 1$. Then, one has

$$B(I-B)\mathbf{x} = B\mathbf{x} - BB\mathbf{x} = \lambda\mathbf{x} - B\lambda\mathbf{x} = \lambda\mathbf{x} - \lambda^2\mathbf{x} = \lambda(1-\lambda)\mathbf{x}.$$

Thus, the statement of the theorem is true for $\nu = 1$. The induction step has the form

$$
\begin{aligned}
B(I-B)^\nu\mathbf{x} &= B(I-B)(I-B)^{\nu-1}\mathbf{x} = B(I-B)^{\nu-1}\mathbf{x} - BB(I-B)^{\nu-1}\mathbf{x} \\
&= \lambda(1-\lambda)^{\nu-1}\mathbf{x} - B\lambda(1-\lambda)^{\nu-1}\mathbf{x} = \lambda(1-\lambda)^{\nu-1}\mathbf{x} - \lambda^2(1-\lambda)^{\nu-1}\mathbf{x} \\
&= \left(\lambda - \lambda^2\right)(1-\lambda)^{\nu-1}\mathbf{x} = \lambda(1-\lambda)^\nu\mathbf{x}.
\end{aligned}
$$

Since $0 \le B \le I$, one has $0 < \lambda \le 1$. Then, it is obvious that

$$0 \le \lambda(1-\lambda)^\nu \le 1,$$

since both factors are between 0 and 1. Hence $B(I-B)^\nu$ is positive semi-definite. One gets, using the definition of the spectral norm, the symmetry of the matrix, the eigenvalue of the square of a matrix, and the nonnegativity of the eigenvalues,

$$
\begin{aligned}
\left\|B(I-B)^\nu\right\|_2 &= \left(\lambda_{\max}\left((B(I-B)^\nu)^T B(I-B)^\nu\right)\right)^{1/2} \\
&= \left(\lambda_{\max}\left((B(I-B)^\nu)^2\right)\right)^{1/2} \\
&= \left((\lambda_{\max}(B(I-B)^\nu))^2\right)^{1/2} \\
&= \lambda_{\max}(B(I-B)^\nu) \\
&= \max_{\lambda \text{ is eigenvalue of } B} \lambda(1-\lambda)^\nu.
\end{aligned}
$$

Thus, one has to maximize $\lambda(1-\lambda)^\nu$ for $\lambda \in [0,1]$ to get an upper bound for $\left\|B(I-B)^\nu\right\|_2$. This expression takes the value zero at the boundary of the interval and it is positive in the interior. Thus, one can compute the maximum with standard calculus

$$\frac{d}{d\lambda}\lambda(1-\lambda)^\nu = (1-\lambda)^\nu - \nu\lambda(1-\lambda)^{\nu-1} = 0.$$

This necessary conditions becomes

$$1 - \lambda - \nu\lambda = 0 \quad \implies \quad \lambda = \frac{1}{1+\nu}.$$

It follows that

$$\|B(I-B)^\nu\|_2 \leq \frac{1}{1+\nu}\left(1-\frac{1}{1+\nu}\right)^\nu = \frac{\nu^\nu}{(1+\nu)^{1+\nu}}.$$

∎

Remark 5.15 *Damped Jacobi method.* Now, the smoothing property of the damped Jacobi method can be proved. The iteration matrix of the damped Jacobi method for the model problem is given by, see also (3.11),

$$S_{\text{jac},\omega} = I - \omega D^{-1}A^h, \quad \omega \in (0,1], \tag{5.11}$$

where $D^{-1}A^h$ is the same for the finite difference and the finite element method. □

Theorem 5.16 Smoothing property of the damped Jacobi method. *Let $S_{\text{jac},\omega}$ be the iteration matrix of the damped Jacobi method given in (5.11), let $\nu \geq 1$, $\nu \in \mathbb{N}$, and let $\omega \in (0,1/2]$. Then it is*

$$\left\|A^h S_{\text{jac},\omega}^\nu\right\|_2 \leq \frac{2}{\omega h}\eta_0(\nu),$$

where $\eta_0(\nu)$ was defined in (5.10).

Proof: The proof will be presented for the finite element method, it can be performed analogously for the finite difference method. For the finite element method, it is $D = 2I/h$. Hence, one gets

$$\begin{aligned}\left\|A^h S_{\text{jac},\omega}^\nu\right\|_2 &= \left\|A^h\left(I-\omega D^{-1}A^h\right)^\nu\right\|_2 = \left\|A^h\left(I-\frac{\omega h}{2}A^h\right)^\nu\right\|_2 \\ &= \frac{2}{\omega h}\left\|\frac{\omega h}{2}A^h\left(I-\frac{\omega h}{2}A^h\right)^\nu\right\|_2.\end{aligned}$$

The matrix $B = \frac{\omega h}{2}A^h$ is symmetric and positive definite and its eigenvalues are, see (2.6),

$$\lambda\left(\frac{\omega h}{2}A^h\right) = \frac{\omega h}{2}\lambda\left(A^h\right) = \frac{\omega h}{2}\frac{4}{h}\sin^2\left(\frac{k\pi}{2N}\right) \leq 2\omega \leq 1$$

with the assumptions of the theorem. Hence $B \leq I$ and Lemma 5.14 can be applied, which gives immediately the statement of the theorem. ∎

Remark 5.17 *To the smoothing property.*
- The smooting property does not hold for the non-damped Jacobi method or the SOR method with relaxation parameter $\omega \geq \omega_{\text{opt}}$, see (Hackbusch, 1994, p. 340).
- The bound $\eta_0(\nu)$ behaves like ν^{-1}, *exercise.* It follows that

$$\left\|A^h S_{\text{jac},\omega}^\nu\right\|_2 \leq \frac{2}{\omega h}\frac{1}{\nu}$$

and the smoothing rate is said to be linear, i.e., $\mathcal{O}\left(\nu^{-1}\right)$.

□

5.4 The Approximation Property

Remark 5.18 *Contents.* Proofs of the approximation property are not only of algebraic nature. They generally use properties of the underlying boundary value problem. Hence, results from the theory of partial differential equations, like error estimates, have to be applied. □

Remark 5.19 *Isomorphism between finite element spaces and Euclidean spaces.* There is a bijection between the functions in the finite element space V^h and the coefficients of the finite element functions in the space $\mathbb{R}^{n_h}$. This bijection is denoted by $P^h \ : \ \mathbb{R}^{n_h} \to V^h$, $\mathbf{v}^h \mapsto v^h$ with

$$v^h(x) = \sum_{i=1}^{n_h} v_i^h \varphi_i^h(x), \quad \mathbf{v}^h = \left(v_i^h \right).$$

If the Euclidean space $\mathbb{R}^{n_h}$ is equipped with the standard Euclidean norm, then the norm equivalence

$$C_0 h^{1/2} \left\| \mathbf{v}^h \right\|_2 \le \left\| P^h \mathbf{v}^h \right\|_{L^2((0,1))} \le C_1 h^{1/2} \left\| \mathbf{v}^h \right\|_2 \tag{5.12}$$

holds with constants that are independent of the mesh size, *exercise.*

There are commutation properties between the grid transfer operators and the bijektion. For instance, for a function given in V^{2h}, one gets the same result if one first applies the bisection to $\mathbb{R}^{n_{2h}}$ and then the interpolation to $\mathbb{R}^{n_h}$ or if one first applies the prolongation to V^h (imbedding) and then applies the bisection to $\mathbb{R}^{n_h}$, i.e.,

$$I_{2h}^h \left(P^{2h} \right)^{-1} v^{2h} = \left(P^h \right)^{-1} \mathcal{I}_{2h}^h v^{2h}, \tag{5.13}$$

where I_{2h}^h on the left-hand side is the matrix representation of the prolongation operator $\mathcal{I}_{2h}^h$ between the finite element spaces. Similarly, if the vector of coefficients is given on the fine grid, one can first apply the bijection and then the restriction or vice versa

$$\mathcal{I}_h^{2h} P^h \mathbf{v}^h = P^{2h} I_h^{2h} \mathbf{v}^h. \tag{5.14}$$

$\square$

Theorem 5.20 Approximation property for the finite element discretization. *Let A^h be defined in (2.4), A^{2h} be defined by the Galerkin projection (5.3), the prolongation $\mathcal{I}_{2h}^h$ be defined in Example 4.9, and the restriction in Example 4.16. Assume that the boundary value problem (2.1) is 2-regular, then the approximation property*

$$\left\| \left(A^h \right)^{-1} - I_{2h}^h \left(A^{2h} \right)^{-1} I_h^{2h} \right\|_2 \le Ch$$

holds.

Proof: Using the definition of an operator norm, the left-hand side of the approximation property (5.8) can be rewritten in the form

$$\sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} \frac{\left\| \left(\left(A^h \right)^{-1} - I_{2h}^h \left(A^{2h} \right)^{-1} I_h^{2h} \right) \mathbf{w}^h \right\|_2}{\| \mathbf{w}^h \|_2} \tag{5.15}$$

Let $A^h \mathbf{z}^h = \mathbf{w}^h$, $A^{2h} \mathbf{z}^{2h} = I_h^{2h} \mathbf{w}^h$, then the numerator can be written as

$$\left\| \mathbf{z}^h - I_{2h}^h \mathbf{z}^{2h} \right\|_2. \tag{5.16}$$

By construction, $\mathbf{z}^h$ is the solution of a finite element problem on the fine grid and $\mathbf{z}^{2h}$ is the solution of almost the same problem on the coarse grid. The right-hand side of the coarse grid problem is the restriction of the right-hand side of the fine grid problem. Therefore, it is a straightforward idea to apply results that are known from finite element error analysis. Consider the finite element problems

$$\left(\left(u^h \right)', \left(\varphi^h \right)' \right) = \left(P^h \mathbf{w}^h, \varphi^h \right) = \left(w^h, \varphi^h \right), \quad \forall \, \varphi^h \in V^h,$$

$$(7.5) \left(\left(u^{2h} \right)', \left(\varphi^{2h} \right)' \right) = \left(w^h, \varphi^{2h} \right), \quad \forall \, \varphi^{2h} \in V^{2h}.$$

Approximating the right-hand side of the first problem by the composite trapezoidal rule and using $\varphi_i^h(x_{i-1}) = \varphi_i^h(x_{i+1}) = 0$, $\varphi_i^h(x_i) = 1$, one gets

$$
\int_{x_{i-1}}^{x_{i+1}} w^h(x)\varphi_i^h(x) \; dx
$$
$$
\approx \quad h\frac{w^h(x_{i-1})\varphi_i^h(x_{i-1}) + w^h(x_i)\varphi_i^h(x_i)}{2} + h\frac{w^h(x_i)\varphi_i^h(x_i) + w^h(x_{i+1})\varphi_i^h(x_{i+1})}{2}
$$
$$
= \quad hw^h(x_i) = hw_i.
$$

This formula, which is exact for constant vectors $\mathbf{w}^h$, is the algebraic form of the right-hand side of the first problem $A^h\mathbf{u}^h = h\mathbf{w}^h$. With the definition of $\mathbf{z}^h$, one obtains

$$
\mathbf{z}^h = \left(A^h\right)^{-1}\mathbf{w}^h = h^{-1}\mathbf{u}^h = h^{-1}\left(P^h\right)^{-1}u^h.
$$

Using the commutation $P^{2h}I_h^{2h}\mathbf{w}^h = \mathcal{I}_h^{2h}P^h\mathbf{w}^h = \mathcal{I}_h^{2h}w^h$, see (5.14), the finite element function $z^{2h} = P^{2h}\mathbf{z}^{2h}$ is the solution of the coarse grid problem

$$
\left(\left(z^{2h}\right)', \left(\varphi^{2h}\right)'\right) = \left(\mathcal{I}_h^{2h}w^h, \varphi^{2h}\right) = \left(w^h, \mathcal{I}_{2h}^h\varphi^{2h}\right), \quad \forall\,\varphi^{2h} \in V^{2h},
$$

where the duality of prolongation and restriction was used, see Example 4.16. The canonical prolongation of φ^{2h} is the embedding, see Example 4.9, hence $\mathcal{I}_{2h}^h\varphi^{2h} = \varphi^{2h}$ and one obtains

$$
\left(\left(z^{2h}\right)', \left(\varphi^{2h}\right)'\right) = \left(w^h, \varphi^{2h}\right), \quad \forall\,\varphi^{2h} \in V^{2h}.
$$

With the same quadrature rule as on the fine grid, it follows that

$$
z^{2h} \quad = \quad \left(P^{2h}\right)\mathbf{z}^{2h} = (2h)^{-1}u^{2h} \quad \Longrightarrow
$$
$$
I_{2h}^h\mathbf{z}^{2h} \quad = \quad (2h)^{-1}I_{2h}^h\left(P^{2h}\right)^{-1}u^{2h} = (2h)^{-1}\left(P^h\right)^{-1}\mathcal{I}_{2h}^h u^{2h},
$$

where (5.13) was used. Since $\mathcal{I}_{2h}^h$ is the identity, one gets that (5.16) can be written in the form

$$
\left\|\mathbf{z}^h - I_{2h}^h\mathbf{z}^{2h}\right\|_2 = h^{-1}\left\|\left(P^h\right)^{-1}\left(u^h - u^{2h}\right)\right\|_2. \tag{5.17}
$$

Since the norm equivalence (5.12) should be applied, the error $\left\|u^h - u^{2h}\right\|_{L^2((0,1))}$ will be estimated. Let $u \in H_0^1((0,1))$ be the solution of the variational problem

$$
\left(u', \varphi'\right) = \left(w^h, \varphi\right) \quad \forall\,\varphi \in H_0^1((0,1)).
$$

This problem is by assumption 2-regular, i.e., it is $u \in H^2((0,1))$ and it holds $\|u\|_{H^2((0,1))} \leq c\left\|w^h\right\|_{L^2((0,1))}$. Then, it is known from Numerical Mathematics 3 that the error estimates

$$
\left\|u - u^h\right\|_{L^2((0,1))} \leq Ch^2\left\|w^h\right\|_{L^2((0,1))}, \quad \left\|u - u^{2h}\right\|_{L^2((0,1))} \leq C\left(2h\right)^2\left\|w^h\right\|_{L^2((0,1))}
$$

hold. Thus, one obtains with the triangle inequality

$$
\left\|u^h - u^{2h}\right\|_{L^2((0,1))} \leq \left\|u - u^h\right\|_{L^2((0,1))} + \left\|u - u^{2h}\right\|_{L^2((0,1))} \leq Ch^2\left\|w^h\right\|_{L^2((0,1))}. \tag{5.18}
$$

Finally, inserting (5.16), (5.17), (5.18) into (5.15) and using the norm equivalence

(5.12) gives

$$
\begin{aligned}
\sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} & \frac{\left\| \left(\left(A^h \right)^{-1} - I_{2h}^h \left(A^{2h} \right)^{-1} I_h^{2h} \right) \mathbf{w}^h \right\|_2}{\|\mathbf{w}^h\|_2} \\
= \quad & \sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} \frac{\left\| \mathbf{z}^h - I_{2h}^h \mathbf{z}^{2h} \right\|_2}{\|\mathbf{w}^h\|_2} \\
= \quad & Ch^{-1} \sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} \frac{\left\| \left(P^h \right)^{-1} \left(u^h - u^{2h} \right) \right\|_2}{\|\mathbf{w}^h\|_2} \\
\leq \quad & Ch^{-3/2} \sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} \frac{\left\| P^h \left(P^h \right)^{-1} \left(u^h - u^{2h} \right) \right\|_{L^2((0,1))}}{\|\mathbf{w}^h\|_2} \\
= \quad & Ch^{-3/2} \sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} \frac{\left\| u^h - u^{2h} \right\|_{L^2((0,1))}}{\|\mathbf{w}^h\|_2} \\
\leq \quad & Ch^{1/2} \sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} \frac{\left\| w^h \right\|_{L^2((0,1))}}{\|\mathbf{w}^h\|_2} \\
\leq \quad & Ch \sup_{\mathbf{w}^h \in \mathbb{R}^{n_h}} \frac{\left\| \mathbf{w}^h \right\|_2}{\|\mathbf{w}^h\|_2} \\
= \quad & Ch.
\end{aligned}
$$

∎

Remark 5.21 *On the approximation property.*
- In the one-dimensional model problem, the assumption on the regularity are satisfied if the right-hand side $f(x)$ is sufficiently smooth. In multiple dimensions, one needs in addition conditions on the domain.
- The proof is literally the same in higher dimensions.

□

5.5   Summary

Remark 5.22 *Summary.* This chapter considered the convergence of the two-level method or coarse grid correction scheme. First, an appropriate coarse grid operator was defined. It was shown that the spectral radius of the iteration matrix of the two-level method can be bounded with a constant lower than 1, independently of the mesh width h, if

- the smoothing property holds and sufficiently many smoothing steps are performed,
- and if the approximation property holds.

Considering the model problem (2.1), the smoothing property for the damped Jacobi problem with $\omega \in (0, 1/2]$ was proved as well as the approximation property. □

Chapter 6

The Multigrid Method

Remark 6.1 *Motivation.* The two-level method leaves an open question: How to solve the coarse grid equation

$$A^{2h}\mathbf{e}^{2h} = I_h^{2h}\left(\mathbf{r}^h\right) =: \mathbf{r}^{2h} \tag{6.1}$$

efficiently? The answer might be apparent: by a two-level method. The form (6.1) is not much different from the original problem. Thus, if one applies the two-level method to the original equation, its application to (6.1) should be easy. A recursive application of this idea, of using the two-level method for solving the coarse grid equation, leads to the multigrid method. □

6.1 Multigrid Cycles

Remark 6.2 *Notations.* To simplify the notations, the right-hand side vector of the residual equation will be denoted by $\mathbf{f}^{2h}$ instead of $\mathbf{r}^{2h}$ since it is just another right-hand side vector. The solution vector on the finest grid will be denoted by $\mathbf{u}^h$ and the current iterate by $\mathbf{v}^h$. Instead of denoting the solution vector on the coarse grid by $\mathbf{e}^{2h}$, it will be denoted by $\mathbf{v}^{2h}$. These notations can be used in an implementation of the method. □

Example 6.3 *A multigrid method.* Now, the two-level method will be imbedded into itself. It will be assumed that there are $l+1$ grids, $l \geq 0$, where the finest grid has the grid spacing h and the grid spacing increase by the factor 2 for each coarser grid. Let $L = 2^l$.
- Apply the smoother ν_1 times to $A^h\mathbf{u}^h = \mathbf{f}^h$ with the initial guess $\mathbf{v}^h$. The results is denoted by $\mathbf{v}^h$.
- Compute $\mathbf{f}^{2h} = I_h^{2h}\mathbf{r}^h = I_h^{2h}\left(\mathbf{f}^h - A^h\mathbf{v}^h\right)$.
  - Apply the smoother ν_1 times to $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$ with the initial guess $\mathbf{v}^{2h} = \mathbf{0}$. Denote the result by $\mathbf{v}^{2h}$.
  - Compute $\mathbf{f}^{4h} = I_{2h}^{4h}\mathbf{r}^{2h} = I_{2h}^{4h}\left(\mathbf{f}^{2h} - A^{2h}\mathbf{v}^{2h}\right)$.

    $\vdots$

    - Solve $A^{Lh}\mathbf{u}^{Lh} = \mathbf{f}^{Lh}$.

    $\vdots$

  - Correct $\mathbf{v}^{2h} := \mathbf{v}^{2h} + I_{4h}^{2h}\mathbf{v}^{4h}$.
  - Apply smoother ν_2 times to $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$ with the initial guess $\mathbf{v}^{2h}$.
- Correct $\mathbf{v}^h := \mathbf{v}^h + I_{2h}^h\mathbf{v}^{2h}$.
- Apply the smoother ν_2 times to $A^h\mathbf{u}^h = \mathbf{f}^h$ with the initial guess $\mathbf{v}^h$.

$\square$

Example 6.4 *Multigrid method with γ-cycle.* The multigrid scheme from Example 6.3 is just one possibility to perform a multigrid method. It belongs to a family of multigrid methods, the so-called multigrid methods with γ-cycle that have the following compact recursive definition:

$\mathbf{v}^h \leftarrow M_\gamma^h(\mathbf{v}^h, \mathbf{f}^h)$

1. Pre smoothing: Apply the smoother ν_1 times to $A^h \mathbf{u}^h = \mathbf{f}^h$ with the initial guess $\mathbf{v}^h$.
2. If Ω^h is the coarsest grid
   − solve the problem.
   else
   − Restrict to the next coarser grid: $\mathbf{f}^{2h} \leftarrow I_h^{2h} \left(\mathbf{f}^h - A^h \mathbf{v}^h \right)$.
   − Set initial iterate on the next coarser grid: $\mathbf{v}^{2h} = \mathbf{0}$.
   − If Ω^h is the finest grid, set $\gamma = 1$.
   − Call the γ-cycle scheme γ times for the next coarser grid:

$$\mathbf{v}^{2h} \leftarrow M_\gamma^{2h} \left(\mathbf{v}^{2h}, \mathbf{f}^{2h} \right).$$

3. Correct with the prolongated update: $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{v}^{2h}$.
4. Post smoothing: Apply the smoother ν_2 times to $A^h \mathbf{u}^h = \mathbf{f}^h$ with the initial guess $\mathbf{v}^h$.

In practice, only $\gamma = 1$ (V-cycle) and $\gamma = 2$ (W-cycle) are used. The names become clear if one has a look on how they move through the hierarchy of grids, see Figures 6.1 and 6.2. $\square$

Figure 6.1: Multigrid V-cycle ($\gamma = 1$), s – smoothing, r – restriction, p – prolongated, e – exact solver.

Example 6.5 *Multigrid F-cycle.* In between the V-cycle and the W-cycle is the F-cycle, see Figure 6.3. The F-cycle starts with the restriction to the coarsest grid. In the prolongation process, after having reached each level the first time, again a restriction to the coarsest grid is performed. $\square$

Remark 6.6 *To the multigrid cycles.*
- The system on the coarsest grid is often small or even very small. Then, it can be solved efficiently with a direct method (Gaussian elimination, Cholesky factorization). Otherwise, one can apply a few steps of an iterative scheme to computed a sufficiently good approximate solution.
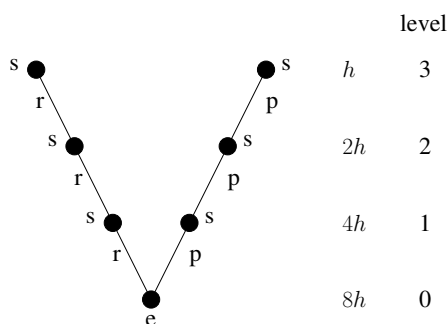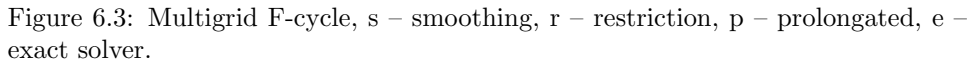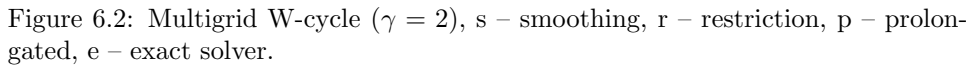
38

Figure 6.2: Multigrid W-cycle ($\gamma = 2$), s – smoothing, r – restriction, p – prolongated, e – exact solver.

Figure 6.3: Multigrid F-cycle, s – smoothing, r – restriction, p – prolongated, e – exact solver.

- In our experience, it is sometimes (depending on the problem) helpful to damp the correction after having prolongated the update. Let $\beta \in (0, 1]$ be given, then instead of Step 3 of the multigrid γ-cycle, the update has the form

$$\mathbf{v}^h \leftarrow \mathbf{v}^h + \beta I_{2h}^h \mathbf{v}^{2h}.$$

- The initial guess for the first pre smoothing step on the finest grid can be obtained by a nested iteration, see Remark 4.5. In the nested iteration, the system is first solved (or smoothed) on a very coarse gird, then one goes to the next finer grid and smoothes the system on this grid and so on, until the finest grid is reached. This approach is called full multigrid. If one uses on each grid, which is not the finest grid, one multigrid V-cycle for smoothing, the so-called full multigrid V-cycle is performed, see Figure 6.4. The full multigrid V-cycle looks like a F-cycle without restriction and pre smoothing.

  In practice, one solves the systems on the coarser grids up to a certain accuracy before one enters the next finer grid.

  $\square$

6.2   Convergence of the W-cycle

Remark 6.7 *Contents.* It will be proved that the sufficient conditions for the convergence of the two-level method, Theorem 5.11, almost imply the convergence of the multigrid W-cycle. The rate of convergence will be bounded by a number $\rho(\nu) < 1$ which depends on the number of pre smoothing steps and which is independent of the finest step size h and of the number of levels involved in the multigrid scheme.
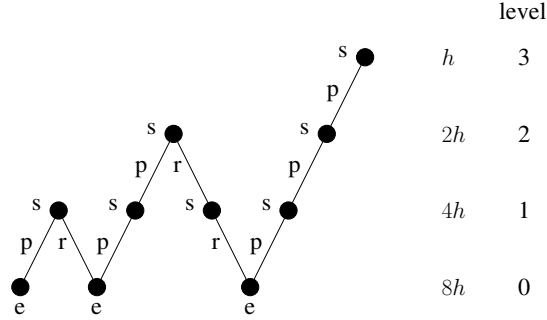
Figure 6.4: Full multigrid V-cycle, s – smoothing, r – restriction, p – prolongated, e – exact solver.

This technique cannot be applied to the multigrid V-cycle. The convergence theory for the V-cycle is more complicated and beyond the scope of this course. □

Remark 6.8 *Preliminarities.* As usual, one has to study the iteration matrix for the investigation of the convergence of an iterative solver. The levels of the multigrid hierarchy are numbered by $0, \ldots, l$, where level 0 is the coarsest grid. The iteration matrix of the two-level method on level l, where the corresponding mesh width should be h, is denoted by S_l and it has the form, see (5.5)

$$S_l(\nu) = \left(I - I_{l-1}^l \left(A_{l-1} \right)^{-1} I_l^{l-1} A_l \right) S_{\text{sm},l}^\nu. \tag{6.2}$$

This iteration matrix is the matrix without post smoothing.

The solution of $A_l \mathbf{u}_l = \mathbf{f}_l$ is a fixed point of the multigrid γ-cycle. This statement follows from the fact that it is a fixed point of the two-level method, see Remark 5.6.

□

Lemma 6.9 Iteration matrix of the multigrid γ-cycle. *The iteration matrix of the multigrid γ-cycle scheme is given by*

$$
\begin{aligned}
S_{\text{mg},l}(\nu) &= S_l(\nu) \quad \textit{if } l = 1, \\
S_{\text{mg},l}(\nu) &= S_l(\nu) + I_{l-1}^l \left(S_{\text{mg},l-1}(\nu) \right)^\gamma A_{l-1}^{-1} I_l^{l-1} A_l S_{\text{sm},l}^\nu \quad \textit{for } l \geq 2. \tag{6.3}
\end{aligned}
$$

Proof: For $l = 1$, the two-level method and the multigrid γ-cycle scheme are identical and the statement of the lemma follows immediately.

The proof for $l \geq 2$ will be performed by induction. Assume that (6.3) holds for $l - 1$. The iteration matrix $S_{\text{mg},l}(\nu)$ can be written in the form

$$S_{\text{mg},l}(\nu) = C_{\text{mg},l} S_{\text{sm},l}^\nu,$$

where $C_{\text{mg},l}$ represents the iteration matrix of the complete coarse grid correction, i.e., everything which was done on the levels $0, \ldots, l - 1$. This matrix has to be determined. To this end, consider the multigrid method with $\mathbf{f}_l = \mathbf{0}$ and let $\mathbf{u}_l$ being arbitrary. For the restricted residual, it holds

$$\mathbf{f}_{l-1} = I_l^{l-1} \left(\mathbf{f}_l - A_l \mathbf{u}_l \right) = -I_l^{l-1} A_l \mathbf{u}_l.$$

Then, in the multigrid γ-cycle, γ iterates $\mathbf{v}_{l-1}^{(1)}, \ldots, \mathbf{v}_{l-1}^{(\gamma)}$ are computed, starting with the initial iterate $\mathbf{v}_{l-1}^{(0)} = \mathbf{0}$. The multigrid γ-cycle on level $l - 1$, which is applied to

$$A_{l-1} \mathbf{u}_{l-1} = \mathbf{f}_{l-1}, \tag{6.4}$$

can be described with the basic form of a fixed point iteration given in (3.3)

$$\mathbf{v}_{l-1}^{(j+1)} = S_{\text{mg},l-1}(\nu) \mathbf{v}_{l-1}^{(j)} + N_{l-1} \mathbf{f}_{l-1}. \tag{6.5}$$

40

From Remark 6.8 it follows that the solution of (6.4) is the fixed point of (6.5). One obtains

$$
\begin{aligned}
\mathbf{v}_{l-1}^{(1)} &= S_{\mathrm{mg},l-1}(\nu)\mathbf{v}_{l-1}^{(0)} + N_{l-1}\mathbf{f}_{l-1} = N_{l-1}\mathbf{f}_{l-1} \\
\mathbf{v}_{l-1}^{(2)} &= S_{\mathrm{mg},l-1}(\nu)N_{l-1}\mathbf{f}_{l-1} + N_{l-1}\mathbf{f}_{l-1} \\
\mathbf{v}_{l-1}^{(3)} &= S_{\mathrm{mg},l-1}(\nu)\left(S_{\mathrm{mg},l-1}(\nu)N_{l-1}\mathbf{f}_{l-1} + N_{l-1}\mathbf{f}_{l-1}\right) + N_{l-1}\mathbf{f}_{l-1} \\
&= \left(S_{\mathrm{mg},l-1}(\nu)\right)^2 N_{l-1}\mathbf{f}_{l-1} + S_{\mathrm{mg},l-1}(\nu)N_{l-1}\mathbf{f}_{l-1} + N_{l-1}\mathbf{f}_{l-1} \\
&\ \vdots \\
\mathbf{v}_{l-1}^{(\gamma)} &= \sum_{k=0}^{\gamma-1} \left(S_{\mathrm{mg},l-1}(\nu)\right)^k N_{l-1}\mathbf{f}_{l-1} \\
&= \sum_{k=0}^{\gamma-1} \left(S_{\mathrm{mg},l-1}(\nu)\right)^k N_{l-1}\left(-I_l^{l-1}\left(A_l\mathbf{u}_l\right)\right). \tag{6.6}
\end{aligned}
$$

Let $\mathbf{u}_{l-1}$ be the fixed point of (6.5) and the solution of (6.4), then it is

$$
\begin{aligned}
\mathbf{u}_{l-1} &= S_{\mathrm{mg},l-1}(\nu)\mathbf{u}_{l-1} + N_{l-1}\mathbf{f}_{l-1} = S_{\mathrm{mg},l-1}(\nu)\mathbf{u}_{l-1} + N_{l-1}A_{l-1}\mathbf{u}_{l-1} \\
&= \left(S_{\mathrm{mg},l-1}(\nu) + N_{l-1}A_{l-1}\right)\mathbf{u}_{l-1}.
\end{aligned}
$$

It follows that

$$
I = S_{\mathrm{mg},l-1}(\nu) + N_{l-1}A_{l-1}
$$

and

$$
N_{l-1} = \left(I - S_{\mathrm{mg},l-1}(\nu)\right)A_{l-1}^{-1}. \tag{6.7}
$$

Using (telescopic sum)

$$
\sum_{k=0}^{\gamma-1} x^k(1-x) = \sum_{k=0}^{\gamma-1} x^k - \sum_{k=0}^{\gamma-1} x^{k+1} = 1 - x^\gamma,
$$

one obtains from (6.6) and (6.7)

$$
\begin{aligned}
\mathbf{v}_{l-1}^{(\gamma)} &= \left(\sum_{k=0}^{\gamma-1} \left(S_{\mathrm{mg},l-1}(\nu)\right)^k \left(I - S_{\mathrm{mg},l-1}(\nu)\right)\right) A_{l-1}^{-1}\left(-I_l^{l-1}\left(A_l\mathbf{u}_l\right)\right) \\
&= \left(I - \left(S_{\mathrm{mg},l-1}(\nu)\right)^\gamma\right) A_{l-1}^{-1}\left(-I_l^{l-1}\left(A_l\mathbf{u}_l\right)\right). \tag{6.8}
\end{aligned}
$$

From the coarse grid correction, step 3 auf the multigrid γ-cycle scheme, see Example 6.4, it follows for the result of the multigrid γ-cycle that

$$
\mathbf{u}_l^{\mathrm{new}} := C_{\mathrm{mg},l}\mathbf{u}_l = \mathbf{u}_l + I_{l-1}^l \mathbf{v}_{l-1}^{(\gamma)}.
$$

Inserting (6.8), one obtains for the iteration matrix of the coarse grid correction

$$
\begin{aligned}
C_{\mathrm{mg},l} &= I + I_{l-1}^l \left(I - \left(S_{\mathrm{mg},l-1}(\nu)\right)^\gamma\right) A_{l-1}^{-1}\left(-I_l^{l-1}A_l\right) \\
&= I - I_{l-1}^l A_{l-1}^{-1} I_l^{l-1} A_l + I_{l-1}^l \left(S_{\mathrm{mg},l-1}(\nu)\right)^\gamma A_{l-1}^{-1} I_l^{l-1} A_l.
\end{aligned}
$$

Hence, the iteration matrix of the multigrid γ-cycle scheme is given by

$$
\begin{aligned}
S_{\mathrm{mg},l}(\nu) &= C_{\mathrm{mg},l}S_{\mathrm{sm},l}^\nu \\
&= \left(I - I_{l-1}^l A_{l-1}^{-1} I_l^{l-1} A_l\right) S_{\mathrm{sm},l}^\nu + I_{l-1}^l \left(S_{\mathrm{mg},l-1}(\nu)\right)^\gamma A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu.
\end{aligned}
$$

The first term is equal to $S_l(\nu)$, see (6.2). Thus, (6.3) is proved for level l under the assumption that it holds for level $l-1$.

One can write the iteration matrix for $l = 1$ also in form (6.3), using the definition $S_{\mathrm{mg},0}(\nu) := 0$. Then, (6.3) holds for $l = 1$ and hence it holds for all $l \geq 1$ by induction. $\blacksquare$

Remark 6.10 *Estimate of the spectral norm of the iteration matrix.* The iteration matrix $S_{\mathrm{mg},l}(\nu)$ of the multigrid γ-cycle scheme is the sum of the iteration matrix of the two-level method and a perturbation. It will be shown that this perturbation is, under certain assumptions, small.

The spectral norm of $S_{\mathrm{mg},l}(\nu)$ will be estimated in a first step by the triangle inequality and the rule for estimating the norm of products of matrices

$$
\begin{aligned}
\left\| S_{\mathrm{mg},l}(\nu) \right\|_2 &\leq \left\| S_l(\nu) \right\|_2 + \left\| I_{l-1}^l \left(S_{\mathrm{mg},l-1}(\nu) \right)^\gamma A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \right\|_2 \\
&\leq \left\| S_l(\nu) \right\|_2 + \left\| I_{l-1}^l \right\|_2 \left\| S_{\mathrm{mg},l-1}(\nu) \right\|_2^\gamma \left\| A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \right\|_2 . \quad (6.9)
\end{aligned}
$$

Now, bounds for all factors on the right-hand side of (6.9) are needed. $\qquad\Box$

Remark 6.11 *Assumptions on the prolongation operator.* It will be assumed that the prolongation is a bounded linear operator with a bound independent of l, i.e., there is a constant $\bar{c}_p$ such that

$$
\left\| I_{l-1}^l \right\|_2 \leq \bar{c}_p \quad \forall \, l \geq 1. \tag{6.10}
$$

In addition, a bound of $\left\| I_{l-1}^l \right\|_2$ from below will be needed. Thus, it will be assumed that there is a constant $\underline{c}_p > 0$ such that for all $\mathbf{u}_{l-1}$ defined on level $l-1$ it is

$$
\underline{c}_p^{-1} \left\| \mathbf{u}_{l-1} \right\|_2 \leq \left\| I_{l-1}^l \mathbf{u}_{l-1} \right\|_2 \quad \forall \, l \geq 1. \tag{6.11}
$$

The assumptions (6.10) and (6.11) are satisfied for the prolongation operator defined in Section 4.2. These properties can be deduced, e.g., by using the definition of the operator norm, *exercise.* $\qquad\Box$

Remark 6.12 *Assumptions on the smoother.* It will be assumed that there is a constant c_s such that

$$
\left\| S_{\mathrm{sm},l}^\nu \right\|_2 \leq c_s \quad \forall \, l \geq 1, 0 < \nu < \infty. \tag{6.12}
$$

This assumption is satisfied, e.g., for the damped Jacobi iteration, $S_{\mathrm{sm},l} = S_{\mathrm{jac},\omega}$ applied to the model problem, with $c_s = 1$. It was shown in the proof of Lemma 3.10 that $\rho(S_{\mathrm{jac},\omega}) < 1$. Since $S_{\mathrm{jac},\omega}$ is a symmetric matrix, it is $\left\| S_{\mathrm{jac},\omega} \right\|_2 = \rho(S_{\mathrm{jac},\omega})$. It follows that

$$
\left\| S_{\mathrm{sm},l}^\nu \right\|_2 = \left\| S_{\mathrm{sm},l} \right\|_2^\nu = \rho(S_{\mathrm{jac},\omega})^\nu < 1.
$$

$\qquad\Box$

Lemma 6.13 Estimate of last term in (6.9) with the iteration matrix of the two-level method. *Suppose* (6.11) *and* (6.12), *then*

$$
\left\| A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \right\|_2 \leq \underline{c}_p \left(c_s + \left\| S_l(\nu) \right\|_2 \right). \tag{6.13}
$$

Proof: One gets with (6.11)

$$
\left\| A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \mathbf{u}_l \right\|_2 \leq \underline{c}_p \left\| I_{l-1}^l A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \mathbf{u}_l \right\|_2
$$

for all $\mathbf{u}_l$, where it is noted that $A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \mathbf{u}_l$ is a vector on level $l-1$. Using the definition of an operator norm gives

$$
\left\| A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \right\|_2 \leq \underline{c}_p \left\| I_{l-1}^l A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \right\|_2 . \tag{6.14}
$$

The right-hand side of this estimate can be rewritten as follows

$$
\begin{aligned}
I_{l-1}^l A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu &= S_{\mathrm{sm},l}^\nu - A_l^{-1} A_l S_{\mathrm{sm},l}^\nu + I_{l-1}^l A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^\nu \\
&= S_{\mathrm{sm},l}^\nu - \left(A_l^{-1} - I_{l-1}^l A_{l-1}^{-1} I_l^{l-1} \right) A_l S_{\mathrm{sm},l}^\nu \\
&= S_{\mathrm{sm},l}^\nu - S_l(\nu).
\end{aligned}
$$

Using this identity in (6.14), applying the triangle inequality, and assumption (6.12) gives

$$
\begin{aligned}
\left\| A_{l-1}^{-1} I_l^{l-1} A_l S_{\mathrm{sm},l}^{\nu} \right\|_2 & \leq \underline{c}_p \left(\left\| S_{\mathrm{sm},l}^{\nu} \right\|_2 + \left\| S_l(\nu) \right\|_2 \right) \\
& \leq \underline{c}_p \left(c_s + \left\| S_l(\nu) \right\|_2 \right).
\end{aligned}
$$

∎

Remark 6.14 *Impact on estimate* (6.9). Only the case will be considered that the number ν of smoothing steps is sufficiently large such that the two-level method converges, i.e., it is

$$
\| S_l(\nu) \|_2 < 1.
$$

Inserting (6.13) into (6.9) and using the assumption on the number of smoothing steps yields, together with (6.10),

$$
\begin{aligned}
\left\| S_{\mathrm{mg},l}(\nu) \right\|_2 & \leq \left\| S_l(\nu) \right\|_2 + \underline{c}_p \left\| I_{l-1}^l \right\|_2 \left\| S_{\mathrm{mg},l-1}(\nu) \right\|_2^{\gamma} \left(c_s + \left\| S_l(\nu) \right\|_2 \right) \\
& \leq \left\| S_l(\nu) \right\|_2 + \underline{c}_p \bar{c}_p \left(c_s + 1 \right) \left\| S_{\mathrm{mg},l-1}(\nu) \right\|_2^{\gamma} \\
& = \left\| S_l(\nu) \right\|_2 + c^* \left\| S_{\mathrm{mg},l-1}(\nu) \right\|_2^{\gamma}.
\end{aligned} \tag{6.15}
$$

This inequality is of the recursive form

$$
x_1 = x, \quad x_l \leq x + c^* x_{l-1}^{\gamma}, \quad l \geq 2, \tag{6.16}
$$

with $x = \| S_l(\nu) \|_2 < 1$ and for $l = 1$ the multigrid and the two-level method coincide. □

Lemma 6.15 Bound for the iterates of inequality (6.16). *Assume that $c^* \gamma > 1$. If $\gamma \geq 2$ and*

$$
x \leq x_{\max} := \frac{\gamma - 1}{\gamma} \left(c^* \gamma \right)^{-\frac{1}{\gamma - 1}},
$$

then every iterate of (6.16) *is bounded by*

$$
x_l \leq \frac{\gamma}{\gamma - 1} x < 1.
$$

Proof: The proof of the bound is performed by induction. For $l = 2$, one has

$$
\begin{aligned}
x_2 & \leq x + c^* x_1^{\gamma} \leq x + c^* x^{\gamma} = x \left(1 + c^* x^{\gamma - 1} \right) \\
& \leq x \left(1 + c^* x_{\max}^{\gamma - 1} \right) = x \left(1 + c^* \left(\frac{\gamma - 1}{\gamma} \right)^{\gamma - 1} \frac{1}{c^* \gamma} \right) \\
& = x \left(1 + \left(\frac{(\gamma - 1)^{\gamma - 1}}{\gamma^{\gamma}} \right) \right) = x \left(1 + \frac{1}{\gamma - 1} \left(1 - \frac{1}{\gamma} \right)^{\gamma} \right) \\
& \leq x \frac{\gamma - 1 + 1}{\gamma - 1} = x \frac{\gamma}{\gamma - 1},
\end{aligned}
$$

since $\left(1 - \frac{1}{\gamma} \right)^{\gamma} < 1$ (positive power of a real number in $(0,1)$).

Let the statement be already proved for $l - 1$, then one obtains with the assumption of the induction

$$
\begin{aligned}
x_l & \leq x + c^* x_{l-1}^{\gamma} \leq x + c^* \left(\frac{\gamma}{\gamma - 1} \right)^{\gamma} x^{\gamma} \\
& = x \left(1 + c^* \left(\frac{\gamma}{\gamma - 1} \right)^{\gamma} x^{\gamma - 1} \right) \leq x \left(1 + c^* \left(\frac{\gamma}{\gamma - 1} \right)^{\gamma} x_{\max}^{\gamma - 1} \right) \\
& = x \left(1 + \left(\frac{\gamma}{\gamma - 1} \right)^{\gamma} \left(\frac{\gamma - 1}{\gamma} \right)^{\gamma - 1} \frac{1}{\gamma} \right) = x \left(1 + \frac{1}{\gamma - 1} \right) = x \frac{\gamma}{\gamma - 1}.
\end{aligned}
$$

Using now the assumption on x and the assumption $c^*\gamma > 1$, one gets

$$\frac{\gamma}{\gamma - 1} x \le \frac{\gamma}{\gamma - 1} x_{\max} = (c^*\gamma)^{-\frac{1}{\gamma-1}} < 1.$$

∎

Remark 6.16 *To Lemma 6.15.* The condition $\gamma \ge 2$ is used in the definition of $x_{\max}$. Note that $x_{\max} < 1$ since both factors are lower than 1.

In the case of the W-cylce, i.e., $\gamma = 2$, the statement of the Lemma 6.15 implies

$$x \le x_{\max} = \frac{1}{4c^*}, \quad x_l \le 2x = 2\,\|S_l(\nu)\|_2\,.$$

□

Theorem 6.17 Convergence of the multigrid γ-cycle for $\gamma \ge 2$. *Suppose $\gamma \ge 2$, (6.10), (6.11), (6.12) with $\nu(h) = \infty$, and the assumptions of Theorem 5.11. Let $\rho \in (0,1)$ be a fixed number. Then there is a number $\underline{\nu}$ such that*

$$\|S_{\mathrm{mg},l}(\nu)\|_2 \le \rho < 1, \quad \|S_{\mathrm{mg},l}(\nu)\|_2 \le \frac{\gamma}{\gamma - 1} C_a \eta(\nu), \qquad (6.17)$$

whenever the number of smoothing iterations ν is larger or equal than $\underline{\nu}$. The estimates (6.17) are independent of the level l and the number of levels. The function $\eta(\nu)$ is defined in the smoothing property (5.7) and the constant C_a is defined in the approximation property.

Proof: Starting point of the proof is inequality (6.15). Lemma 6.15 will be applied with
$$x = \|S_l(\nu)\|_2\,, \quad x_l = \|S_{\mathrm{mg},l}(\nu)\|_2\,.$$
Without loss of generality, one can choose

$$c^* > \frac{1}{\gamma} \quad \Longleftrightarrow \quad c^*\gamma > 1.$$

In particular, c^* can be chosen so large that

$$x \le \frac{\gamma - 1}{\gamma} (c^*\gamma)^{-\frac{1}{\gamma-1}} \le \frac{\gamma - 1}{\gamma} \rho < 1.$$

Note that large values of c^* imply small values of x, which can be always obtained by applying sufficiently many smoothing steps. Thus, the assumptions of Lemma 6.15 are satisfied and one obtains

$$\|S_{\mathrm{mg},l}(\nu)\|_2 \le \frac{\gamma}{\gamma - 1} \|S_l(\nu)\|_2 = \frac{\gamma}{\gamma - 1} x \le \rho.$$

The second estimate is obtained recursively. Using formally the same computations as in the proof of Lemma 6.15, one gets

$$\|S_{\mathrm{mg},2}(\nu)\|_2 \le \frac{\gamma}{\gamma - 1} \|S_2(\nu)\|_2 \le \frac{\gamma}{\gamma - 1} C_a \eta(\nu),$$

and by induction
$$\|S_{\mathrm{mg},l}(\nu)\|_2 \le \frac{\gamma}{\gamma - 1} \|S_l(\nu)\|_2 \le \frac{\gamma}{\gamma - 1} C_a \eta(\nu),$$
The details of this proof are an *exercise*. ∎

Remark 6.18 *To Theorem 6.17.*
- The theorem states the convergence of the multigrid γ-cycle with a rate of convergence that is independent of the level. The estimate of this rate, i.e., $\frac{\gamma}{\gamma-1} C_a \eta(\nu)$, is in general somewhat pessimistic.

- A similar result can be proved if only post smoothing and no pre smoothing is applied, as well as in the case that both pre and post smoothing are used.
- The convergence proof for the V-cycle, i.e., $\gamma = 1$, does not rely on the convergence of the two-level method. In this proof, the multigrid iteration matrix is analyzed directly, e.g., see (Hackbusch, 1985, pp. 164).
- For problems without symmetric positive definite system matrix, multigrid works often quite well. But only very little is proved on the convergence of multigrid methods for such problems. Results on the multigrid convergence for problems without symmetric positive definite matrix are in general for problems which are only a slight perturbation of a s.p.d. problem. But many interesting problems are not small perturbations of a s.p.d. problem, like convection-dominated convection-diffusion equations or the Navier–Stokes equations. In these fields, many questions concerning the theory of multigrid methods are open. Some results for convection-diffusion problems can be found in Reusken (2002); Olshanskii and Reusken (2004).

$\square$

6.3 Computational Work of the Multigrid γ-Cycle

Remark 6.19 *Goal.* So far it is proved that the rate of convergence for the multigrid γ-cycle is bounded by a number $\rho < 1$ independently of the level. That means, the number of iterations for solving the equation up to a certain accuracy is bounded from above by a constant which is independent of the level, i.e., one needs on each grid level essentially the same number of iterations to solve the equation. This behavior is in contrast to the classical iteration schemes or the PCG method, where the number of iterations increases by the factor of 4 or 2, respectively, if the grid is refined once, cf. Table 2.1.

Let N_l the number of degrees of freedom on level l, $1 \leq l \leq L$. To obtain an optimal algorithm, one needs to show that the number of operations (flops) per multigrid cycle behaves like $\mathcal{O}(N_l)$. Since the number of multigrid cycles for the solution of the linear system is bounded uniformly, i.e., independently of l, it follows that then also the solution of the linear system requires $\mathcal{O}(N_l)$ operations. $\square$

Remark 6.20 *Assumptions on the computational costs of the components of the multigrid method.* The following bounds for the number of operations are assumed for the basic components of the multigrid method:
- one smoothing step $\mathbf{u}_l := S_l(\mathbf{u}_l)$

$$\text{flops} \leq c_s N_l, \quad l \geq 1,$$

- restriction $\mathbf{f}_{l-1} = I_l^{l-1} (\mathbf{f}_l - A_l \mathbf{v}_l)$

$$\text{flops} \leq c_r N_l, \quad l \geq 1,$$

- prolongation and correction $\mathbf{u}_l := \mathbf{u}_l + I_{l-1}^l \mathbf{v}_{l-1}$

$$\text{flops} \leq c_p N_l, \quad l \geq 1,$$

- coarsest grid problem $\mathbf{u}_0 = A_0^{-1} \mathbf{f}_0$

$$\text{flops} \leq c_0.$$

For sparse matrices and the prolongation and restriction which were introduced in Chapter 4, the bounds are true. The system on the coarsest grid can be solved, e.g.,

by Gaussian elimination. Then, c_0 depends on the number of degrees of freedom on the coarsest grid, but not on N_l.

Let
$$c_h = \sup_{l \geq 1} \frac{N_{l-1}}{N_l}.$$

For uniformly refined grids, i.e., $h_{l-1} = 2h_l$, this constant has the value $c_h = 2^{-d}$, where d is the dimension of the domain. $\qquad\square$

Theorem 6.21 Number of operations for the multigrid γ-cycle. *Set $\theta = c_h\gamma$, let $\theta < 1$ and let the assumptions from Remark 6.20 be satisfied. Then, one cycle of the multigrid γ-cycle with ν smoothing steps on each level requires $c_l N_l$ operations, where*
$$c_l < \frac{\nu c_s + c_r + c_p}{1 - \theta} + \theta^{l-1} \frac{c_0}{N_1}. \tag{6.18}$$

Proof: One iteration at level l involves γ^{l-k} iterations at level k, $1 \leq k \leq l$, since there are

- γ iterations on level $l - 1$,
- at each of these iterations, γ iterations on level $l - 2$, i.e., γ^2 iterations on level $l - 2$,
- and so on.

On level 0, γ^{l-1} coarsest grid systems have to be solved, since in each of the γ^{l-1} situations where one is on level 1, level 0 is called. Using the assumptions on the costs of the basic components of the multigrid method, on obtains the following costs

$$
\begin{aligned}
&(\nu c_s + c_r + c_p)\, N_l + \gamma\, (\nu c_s + c_r + c_p)\, N_{l-1} + \gamma^2\, (\nu c_s + c_r + c_p)\, N_{l-2} \\
&\quad + \ldots + \gamma^{l-1}\, (\nu c_s + c_r + c_p)\, N_1 + \gamma^{l-1} c_0 \\
&= (\nu c_s + c_r + c_p)\left(N_l + \gamma N_{l-1} + \ldots + \gamma^{l-1} N_1 \right) + \gamma^{l-1} c_0 \\
&= (\nu c_s + c_r + c_p)\, N_l \left(1 + \gamma \frac{N_{l-1}}{N_l} + \ldots + \gamma^{l-1} \frac{N_1}{N_l} \right) + \gamma^{l-1} c_0 \\
&= (\nu c_s + c_r + c_p)\, N_l \left(1 + \gamma \frac{N_{l-1}}{N_l} + \gamma^2 \frac{N_{l-2}}{N_{l-1}} \frac{N_{l-1}}{N_l} + \ldots \right) + \gamma^{l-1} c_0 \\
&\leq (\nu c_s + c_r + c_p)\, N_l \left(1 + \gamma c_h + \gamma^2 c_h^2 + \ldots + \gamma^{l-1} c_h^{l-1} \right) + \gamma^{l-1} c_0 \\
&= (\nu c_s + c_r + c_p)\, N_l \left(1 + \theta + \theta^2 + \ldots + \theta^{l-1} \right) + \gamma^{l-1} c_0 \tag{6.19} \\
&\leq (\nu c_s + c_r + c_p)\, \frac{N_l}{1 - \theta} + \theta^{l-1} \frac{c_0}{c_h^{l-1}} \frac{N_l}{N_l} \\
&\leq \left(\frac{\nu c_s + c_r + c_p}{1 - \theta} + \theta^{l-1} \frac{c_0}{N_1} \right) N_l,
\end{aligned}
$$

since $c_h^{l-1} \geq N_1/N_l$ for $l \geq 1$. $\qquad\blacksquare$

Remark 6.22 *On the bound* (6.18). The bound (6.18) depends formally on l. One can remove this dependence by using that $\theta^{l-1} < \theta$. However, in the form (6.18) it becomes clearer that the importance of the flops of the coarsest grid solver decreases with increasing level. $\qquad\square$

Example 6.23 *Computational costs for different cycles.* Consider a standard uniform refinement, i.e., it is $c_h = 2^{-d}$, where d is the dimension of the domain.

For one dimension, the theory applies for the V-cycle because $\gamma c_h = 1/2$, but not for the W-cycle since $\gamma c_h = 1$.

In two dimensions, one has for the V-cycle $\gamma c_h = 1/4$ and for the W-cycle $\gamma c_h = 1/2$. Then, one obtains from (6.18) the following estimates for the computational costs:

- V-cycle

$$c_l < \frac{4}{3} \left(\nu c_s + c_r + c_p \right) + \left(\frac{1}{4} \right)^{l-1} \frac{c_0}{N_1},$$

- W-cycle

$$c_l < 2 \left(\nu c_s + c_r + c_p \right) + \left(\frac{1}{2} \right)^{l-1} \frac{c_0}{N_1}.$$

Neglecting the flops for the coarsest grid solver, a W-cycle for a two-dimensional problem requires roughly 1.5 times the number of flops of a V-cycle.

In three dimensions, one finds for the V-cycle that $\gamma c_h = 1/8$ and for the W-cycle that $\gamma c_h = 1/4$. Then, the number of flops per cycle is bounded by

- V-cycle

$$c_l < \frac{8}{7} \left(\nu c_s + c_r + c_p \right) + \left(\frac{1}{8} \right)^{l-1} \frac{c_0}{N_1},$$

- W-cycle

$$c_l < \frac{4}{3} \left(\nu c_s + c_r + c_p \right) + \left(\frac{1}{4} \right)^{l-1} \frac{c_0}{N_1}.$$

Hence, the W-cycle is only 1.167 times as expensive as the V-cycle.

These results to think about using different strategies for different dimensions. The V-cycle is always more efficient whereas the W-cycle is generally more stable. Since the efficiency gain of the V-cycle in three dimensions is only small, one should apply there the W-cycle. In two dimensions, one should first try if the V-cycle works. As alternative, one can use in both cases the F-cycle. The computation of the numerical costs of the F-cycle is an *exercise*. □

Corollary 6.24 Number of flops for $\theta = 1$. *Let the notations be as in Theorem 6.21 and let $\theta = 1$. Then, the number of operations on level l is bounded by*

$$\left(\left(\nu c_s + c_r + c_p \right) l + \frac{c_0}{N_1} \right) N_l.$$

Proof: The proof starts like the proof of Theorem 6.21 until (6.19). Then, one sets $\theta = 1$ in (6.19) to obtain the statement of the corollary. ∎

Example 6.25 *W-cycle in one dimension.* The corollary states that the number of flops for the W-cycle in one dimension is not proportional to N_l. Hence, the W-cycle is not optimal in one dimension. □

Remark 6.26 *Memory requirements of the multigrid method.* The sparse matrix on level l requires the storage of $c_m N_l$ numbers, where c_m is independent of l. In addition, one has to store the arrays $\mathbf{v}_l$ and $\mathbf{f}_l$, which are $2N_l$ numbers. It follows that the total storage requirements are

$$
\begin{aligned}
(2 + c_m) \sum_{k=0}^{l} N_k &= (2 + c_m) \left(N_l + N_l \frac{N_{l-1}}{N_l} + N_l \frac{N_{l-1}}{N_l} \frac{N_{l-2}}{N_{l-1}} + \dots \right) \\
&\leq (2 + c_m) N_l \sum_{k=0}^{l-1} c_h^k \\
&\leq \frac{(2 + c_m) N_l}{1 - c_h},
\end{aligned}
$$

if $c_h < 1$. A method that works only on the finest grid requires at least the storage of $(2 + N_l)$ numbers. Thus, for uniform standard refinement, i.e., $c_h = 2^{-d}$, one has for

- $d = 1$: that the multigrid method needs 100 %,
- $d = 2$: that the multigrid method needs 33.3 %,
- $d = 3$: that the multigrid method needs 14.3 %,

more memory than a single grid algorithm on the finest grid. □

Chapter 7

Algebraic Multigrid Methods

Remark 7.1 *Motivation.* The (geometric) multigrid methods described so far need a hierarchy of (geometric) grids, from the coarsest one ($l = 0$) to the finest one. On all levels but the coarsest one, the smoother will be applied and on the coarsest level, the system is usually solved exactly. However, the following question arises:

- What should be done if the available coarsest grid possesses already that many degrees of freedom that the use of a direct solver takes too much time ?

This situation will happen frequently if the problem is given in a complicated domain in $\mathbb{R}^d$, $d \in \{2, 3\}$, see Figure 7.1 for an (academic) example. Complicated domains are very likely to be given in applications. Then, the application of a grid generator will often lead to (coarsest) grids that are so fine that a refinement would lead to so many degrees of freedom that an efficient simulation of the problem is not possible. Altogether, there is just one grid.

To handle the situation of a coarsest grid with many degrees of freedom, there are at least two possibilities.

- *One level iterative scheme.* In the case that there is a geometric grid hierarchy but the coarse grid is already fine, one can use a simple iterative method, e.g., the smoother, to solve the system on the coarsest grid approximately. Then, the smooth error modes on this grid are not damped. However, experience shows that this approach works in practice sometimes quite well.
  If there is just one grid available, a Krylov subspace method can be used for solving the arising linear systems of equations.
- *Iterative scheme with multilevel ideas.* Construct a more complicated iterative method which uses a kind of multigrid idea for the solution of the system on the coarsest geometric grid. The realization of this multigrid idea should be based only on information which can be obtained from the matrix on the coarsest grid. This type of solver is called Algebraic Multigrid Method (AMG).

□

7.1 Components of an AMG and Definitions

Remark 7.2 *Components.* An AMG possesses the same components as a geometric multigrid method:
- a hierarchy of levels,
- a smoother,
- a prolongation,
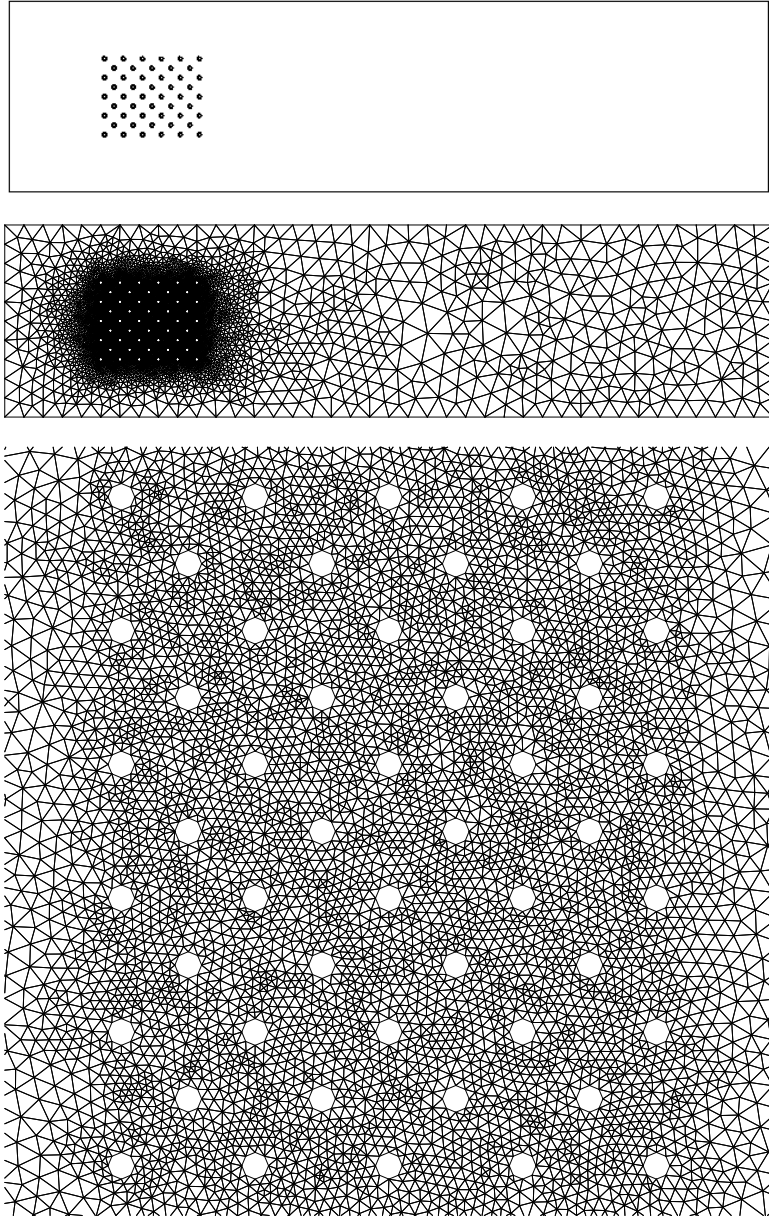- a restriction,
- coarse grid operators.

Figure 7.1: Top: domain with many holes (like the stars in the flag of the United Stars); middle: triangular grid from a grid generator; bottom: zoom into the region with the holes.

A level or a grid is a set of unknowns or degrees of freedom. In contrast to geometric multigrid methods, the hierarchy of levels is obtained by starting from a finest level and reducing the number of unknowns to get the coarser levels.

AMGs restrict themselves on using only simple smoothers, e.g., the damped Jacobi method. This approach is in contrast to geometric multigrid methods, whose efficiency can be enhanced by using appropriate smoothers.

In this course, only the case of symmetric positive definite matrices will be considered. Then, the restriction is always defined as the transpose of the prolongation, i.e.,

$$I_f^c = \left(I_c^f\right)^T,$$

where "f" refers to the fine grid and "c" to the next coarser grid.

The coarse grid operator is defined by the Galerkin projection

$$A^c = I_f^c A^f I_c^f.$$

<div align="right">□</div>

Remark 7.3 *Main tasks in the construction of AMGs.* There remain two main tasks in the construction of an AMG:

- An appropriate hierarchy of levels has to be constructed fully automatically, using only information from the matrix on the current grid to construct the next coarser grid.
- One has to define an appropriate prolongation operator.

These two components will determine the efficiency of the AMG.

In contrast to geometric multigrid methods, an AMG constructs from a given grid a coarser grid. Since the final number of coarser grids is not known a priori, it makes sense to denote the starting grid by level 0, the next coarser grid by level 1 and so on.

The coarsening process of an AMG should work automatically, based only on information from the matrix on the current level. To describe this process, some notation is needed. AMGs are set up in an algebraic environment. However, it is often convenient to use a grid terminology by introducing fictitious grids with grid points being the nodes of a graph which is associated with the given matrix $A = (a_{ij})$.

<div align="right">□</div>

Definition 7.4 Graph of a matrix, set of neighbor vertices, coupled vertices. Let A be a sparse $n \times n$ matrix with a symmetric sparsity pattern, i.e., a_{ij} is allowed to be non-zero if and only if a_{ji} is allowed to be non-zero. Let $\Omega = G_A(V, E)$ be the graph of the matrix consisting of a set

$$V = \{v_1, \ldots, v_n\}$$

of n ordered vertices (nodes, unknowns, degrees of freedom) and a set of edges E such that the edge e_{ij}, which connects v_i and v_j for $i \neq j$, belongs to E if and only if a_{ij} is allowed to be non-zero.

For a vertex v_i, the set of its neighbor vertices N_i is defined by

$$N_i = \{v_j \in V \ : \ e_{ij} \in E\}.$$

The number of elements in N_i is denoted by $|N_i|$.

If $e_{ij} \in E$, then the vertices v_i and v_j are called coupled or connected. □

Example 7.5 *Graph of a matrix.* Consider the matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}. \tag{7.1}$$

Let the vertex v_i correspond to the i-th unknown, i.e., to the degree of freedom that corresponds to the i-th row of the matrix. Then the graph of A has the form as given in Figure 7.2. It is

$$E = \{e_{12}, e_{21}, e_{13}, e_{31}, e_{24}, e_{42}, e_{34}, e_{43}\}.$$

<div align="right">□</div>

Figure 7.2: Graph $\Omega = G_A(V, E)$ of the matrix (7.1).

7.2 Algebraic Smoothness

Remark 7.6 *Notations.* In geometric multigrid methods, an error is called smooth if it can be approximated well on some pre-defined coarser level. In AMGs there are no pre-defined grids. Let S be the smoother on Ω, then an error is said to be algebraically smooth if the convergence of the fixed point iteration with the matrix S is slow, i.e., $S\mathbf{e} \approx \mathbf{e}$.

To define the property of algebraic smoothness precisely, some inner products and norms of vectors have to be introduced. Let D be the diagonal matrix corresponding to $A \in \mathbb{R}^{n \times n}$ and let $(\cdot, \cdot)$ be the Euclidean inner product of two vectors

$$(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} u_i v_i.$$

Then, the following inner products and norms are defined

$$
\begin{aligned}
(\mathbf{u}, \mathbf{v})_0 &= (D\mathbf{u}, \mathbf{v}), & \|\mathbf{u}\|_0 &= (\mathbf{u}, \mathbf{u})_0^{1/2}, \\
(\mathbf{u}, \mathbf{v})_1 &= (A\mathbf{u}, \mathbf{v}), & \|\mathbf{u}\|_1 &= (\mathbf{u}, \mathbf{u})_1^{1/2}, \\
(\mathbf{u}, \mathbf{v})_2 &= \left(D^{-1}A\mathbf{u}, A\mathbf{v}\right), & \|\mathbf{u}\|_2 &= (\mathbf{u}, \mathbf{u})_2^{1/2}.
\end{aligned}
$$

The norm $\|\mathbf{u}\|_1$ is sometimes called energy norm.

In this course, only classes of matrices will be considered where $\rho\left(D^{-1}A\right)$ is uniformly bounded, i.e., the spectral radius is bounded independently of the grid. This property holds for many classes of matrices which occur in applications. □

Lemma 7.7 Properties of the norms. *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix. Then the following inequalities hold for all $\mathbf{v} \in \mathbb{R}^n$:*

$$
\begin{aligned}
\|\mathbf{v}\|_1^2 &\leq \|\mathbf{v}\|_0 \|\mathbf{v}\|_2, & (7.2)
\end{aligned}
$$

$$
\begin{aligned}
\|\mathbf{v}\|_2^2 &\leq \rho\left(D^{-1}A\right)\|\mathbf{v}\|_1^2, & (7.3)
\end{aligned}
$$

$$
\begin{aligned}
\|\mathbf{v}\|_1^2 &\leq \rho\left(D^{-1}A\right)\|\mathbf{v}\|_0^2. & (7.4)
\end{aligned}
$$

Proof: (7.2). This estimate follows from the Cauchy–Schwarz inequality and the

symmetry of D

$$
\begin{aligned}
\|\mathbf{v}\|_1^2 &= (A\mathbf{v}, \mathbf{v}) = \mathbf{v}^T A \mathbf{v} = \mathbf{v}^T A D^{-1/2} D^{1/2} \mathbf{v} = \left(D^{-1/2} A \mathbf{v}, D^{1/2} \mathbf{v} \right) \\
&\leq \left\| D^{-1/2} A \mathbf{v} \right\| \left\| D^{1/2} \mathbf{v} \right\| \\
&= \left(D^{-1/2} A \mathbf{v}, D^{-1/2} A \mathbf{v} \right)^{1/2} \left(D^{1/2} \mathbf{v}, D^{1/2} \mathbf{v} \right)^{1/2} \\
&= \left(A \mathbf{v}, D^{-1/2} D^{-1/2} A \mathbf{v} \right)^{1/2} \left(\mathbf{v}, D^{1/2} D^{1/2} \mathbf{v} \right)^{1/2} \\
&= \left(A\mathbf{v}, D^{-1} A \mathbf{v} \right)^{1/2} \left(\mathbf{v}, D\mathbf{v} \right)^{1/2} \\
&= \|\mathbf{v}\|_2 \|\mathbf{v}\|_0,
\end{aligned}
$$

where $\|\cdot\|$ is here the Euclidean vector norm.

(7.3). The matrix $D^{-1}A$ is in general not a symmetric matrix. However, it has the same eigenvalues as the symmetric matrix $A^{1/2}D^{-1}A^{1/2}$, since from

$$
D^{-1} A \mathbf{x} = \lambda \mathbf{x}
$$

one obtains with $\mathbf{x} = A^{-1/2}\mathbf{y}$

$$
D^{-1} A A^{-1/2} \mathbf{y} = \lambda A^{-1/2} \mathbf{y} \quad \Longleftrightarrow \quad A^{1/2} D^{-1} A^{1/2} \mathbf{y} = \lambda \mathbf{y}.
$$

In particular, the spectral radii of both matrices are the same. Using the definition of the positive definiteness, one sees that $A^{1/2}D^{-1}A^{1/2}$ is positive definite since the diagonal of a positive definite matrix is a positive definite matrix. Hence, one gets, using a well known property of the spectral radius for symmetric positive definite matrices (Rayleigh quotient)

$$
\rho \left(D^{-1} A \right) = \rho \left(A^{1/2} D^{-1} A^{1/2} \right) = \lambda_{\max} \left(A^{1/2} D^{-1} A^{1/2} \right) = \sup_{\mathbf{x} \in \mathbb{R}^n} \frac{\left(A^{1/2} D^{-1} A^{1/2} \mathbf{x}, \mathbf{x} \right)}{(\mathbf{x}, \mathbf{x})}.
$$

Setting now $\mathbf{x} = A^{1/2}\mathbf{v}$ gives an estimate of the spectral radius from below

$$
\rho \left(D^{-1} A \right) \geq \frac{\left(A^{1/2} D^{-1} A^{1/2} A^{1/2} \mathbf{v}, A^{1/2} \mathbf{v} \right)}{(A^{1/2} \mathbf{v}, A^{1/2} \mathbf{v})} = \frac{(D^{-1} A \mathbf{v}, A \mathbf{v})}{(A\mathbf{v}, \mathbf{v})} = \frac{\|\mathbf{v}\|_2^2}{\|\mathbf{v}\|_1^2},
$$

where the symmetry of A was also used.

(7.4). The matrix $D^{-1}A$ has also the same eigenvalues as the matrix $D^{-1/2}AD^{-1/2}$, since from

$$
D^{-1} A \mathbf{x} = \lambda \mathbf{x}
$$

it follows with $\mathbf{x} = D^{-1/2}\mathbf{y}$ that

$$
D^{-1} A D^{-1/2} \mathbf{y} = \lambda D^{-1/2} \mathbf{y} \quad \Longleftrightarrow \quad D^{-1/2} A D^{-1/2} \mathbf{y} = \lambda \mathbf{y}.
$$

Hence, $\rho \left(D^{-1} A \right) = \rho \left(D^{-1/2} A D^{-1/2} \right)$. The matrix $D^{-1/2} A D^{-1/2}$ is symmetric and positive definite, which follows by the definition of the positive definiteness and the assumed positive definiteness of A. Using the Rayleigh quotient yields

$$
\rho \left(D^{-1} A \right) = \rho \left(D^{-1/2} A D^{-1/2} \right) = \lambda_{\max} \left(D^{-1/2} A D^{-1/2} \right) = \sup_{\mathbf{x} \in \mathbb{R}^n} \frac{\left(D^{-1/2} A D^{-1/2} \mathbf{x}, \mathbf{x} \right)}{(\mathbf{x}, \mathbf{x})}.
$$

Setting $\mathbf{x} = D^{1/2}\mathbf{v}$, it follows that

$$
\rho \left(D^{-1} A \right) \geq \frac{\left(D^{-1/2} A D^{-1/2} D^{1/2} \mathbf{v}, D^{1/2} \mathbf{v} \right)}{(D^{1/2} \mathbf{v}, D^{1/2} \mathbf{v})} = \frac{(A\mathbf{v}, \mathbf{v})}{(D\mathbf{v}, \mathbf{v})} = \frac{\|\mathbf{v}\|_1^2}{\|\mathbf{v}\|_0^2}.
$$

■

Lemma 7.8 On the eigenvectors of $D^{-1}A$. *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix and ϕ be an eigenvector of $D^{-1}A$ with the eigenvalue λ, i.e.,*

$$D^{-1}A\phi = \lambda\phi.$$

Then it is

$$\|\phi\|_2^2 = \lambda \|\phi\|_1^2, \quad \|\phi\|_1^2 = \lambda \|\phi\|_0^2.$$

Proof: The first statement is obtained by multiplying the eigenvalue problem from the left with $\phi^T A$ giving

$$\left(A\phi, D^{-1}A\phi\right) = \lambda\left(A\phi, \phi\right).$$

The second equality follows from multiplying the eigenvalue problem from left with $\phi^T D$

$$\left(\phi, DD^{-1}A\phi\right) = \lambda\left(\phi, D\phi\right).$$

■

Definition 7.9 Smoothing property of an operator. A smoothing operator S is said to satisfy the smoothing property with respect to a symmetric positive definite matrix A if

$$\|S\mathbf{v}\|_1^2 \le \|\mathbf{v}\|_1^2 - \sigma \|\mathbf{v}\|_2^2 \tag{7.5}$$

with $\sigma > 0$ independent of $\mathbf{v}$.

Let $\mathcal{A}$ be a class of matrices. If the smoothing property (7.5) is satisfied for all $A \in \mathcal{A}$ for a smoothing operator S with the same σ, then S is said to satisfy the smoothing property uniformly with respect to $\mathcal{A}$. □

Remark 7.10 *On the smoothing property.* The definition of the smoothing property implies that S reduces the error efficiently as long as $\|\mathbf{v}\|_2$ is relatively large compared with $\|\mathbf{v}\|_1$. However, the smoothing operator will become very inefficient if $\|\mathbf{v}\|_2 \ll \|\mathbf{v}\|_1$. □

Definition 7.11 Algebraically smooth error. An error $\mathbf{v}$ is called algebraically smooth if $\|\mathbf{v}\|_2 \ll \|\mathbf{v}\|_1$. □

Remark 7.12 *Algebraically smooth error.* An algebraically smooth error is a vector for which an iteration with S converges slowly. The term "smooth" for this property is used for historical reasons.

It will be shown now that the damped Jacobi iteration satisfies the smoothing property (7.5) uniformly for symmetric positive definite matrices. □

Lemma 7.13 Equivalent formulation of the smoothing property. *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix and let the smoothing operator be of the form*

$$S = I - Q^{-1}A$$

with some non-singular matrix Q. Then the smoothing property (7.5) is equivalent to

$$\sigma\left(Q^T D^{-1}Q\mathbf{v}, \mathbf{v}\right) \le \left(\left(Q + Q^T - A\right)\mathbf{v}, \mathbf{v}\right) \quad \forall\, \mathbf{v} \in \mathbb{R}^n. \tag{7.6}$$

Proof: It is

$$
\begin{aligned}
\|S\mathbf{v}\|_1^2 &= (AS\mathbf{v}, S\mathbf{v}) = \left(A\left(I - Q^{-1}A\right)\mathbf{v}, \left(I - Q^{-1}A\right)\mathbf{v}\right) \\
&= (A\mathbf{v}, \mathbf{v}) - \left(AQ^{-1}A\mathbf{v}, \mathbf{v}\right) - \left(A\mathbf{v}, Q^{-1}A\mathbf{v}\right) + \left(AQ^{-1}A\mathbf{v}, Q^{-1}A\mathbf{v}\right) \\
&= \|\mathbf{v}\|_1^2 - \left(Q^T Q^{-1}A\mathbf{v}, Q^{-1}A\mathbf{v}\right) - \left(QQ^{-1}A\mathbf{v}, Q^{-1}A\mathbf{v}\right) + \left(AQ^{-1}A\mathbf{v}, Q^{-1}A\mathbf{v}\right) \\
&= \|\mathbf{v}\|_1^2 - \left(\left(Q^T + Q - A\right)Q^{-1}A\mathbf{v}, Q^{-1}A\mathbf{v}\right).
\end{aligned}
$$

Hence, the algebraic smoothing property (7.5) is equivalent to the condition that for all $\mathbf{v} \in \mathbb{R}^n$:

$$
\begin{aligned}
\sigma \|\mathbf{v}\|_2^2 &\leq \left(\left(Q^T + Q - A \right) Q^{-1} A \mathbf{v}, Q^{-1} A \mathbf{v} \right) &\Longleftrightarrow \\
\sigma \left(D^{-1} A \mathbf{v}, A \mathbf{v} \right) &\leq \left(\left(Q^T + Q - A \right) Q^{-1} A \mathbf{v}, Q^{-1} A \mathbf{v} \right) &\Longleftrightarrow \\
\sigma \left(D^{-1} Q \mathbf{y}, Q \mathbf{y} \right) &\leq \left(\left(Q^T + Q - A \right) \mathbf{y}, \mathbf{y} \right),
\end{aligned}
$$

with $\mathbf{y} = Q^{-1} A \mathbf{v}$. Since the matrices A and Q are non-singular, $\mathbf{y}$ is an arbitrary vector from $\mathbb{R}^n$. Hence, the statement of the lemma is proved. ∎

Theorem 7.14 Algebraic smoothing property of the damped Jacobi method. *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric and positive definite matrix and let $\eta > \rho \left(D^{-1} A \right)$. Then, the damped Jacobi iteration with the damping parameter $\omega \in (0, 2/\eta)$ satisfies the algebraic smoothing property (7.5) uniformly with $\sigma = \omega(2 - \omega\eta)$.*

Proof: The damped Jacobi iteration satisfies the assumptions of Lemma 7.13 with $Q = \omega^{-1} D$. Hence, the algebraic smoothing property (7.5) is eqivalent to (7.6), which gives

$$
\sigma \left(\frac{D}{\omega^2} \mathbf{v}, \mathbf{v} \right) \leq \left(\frac{2D}{\omega} \mathbf{v}, \mathbf{v} \right) - (A\mathbf{v}, \mathbf{v}) \quad \Longleftrightarrow \quad (A\mathbf{v}, \mathbf{v}) \leq \left(\left(\frac{2}{\omega} - \frac{\sigma}{\omega^2} \right) D \mathbf{v}, \mathbf{v} \right)
$$

$$
\Longleftrightarrow \quad \|\mathbf{v}\|_1^2 \leq \left(\frac{2}{\omega} - \frac{\sigma}{\omega^2} \right) \|\mathbf{v}\|_0^2 . \tag{7.7}
$$

From inequality (7.4) and the assumption on η it follows for all $\mathbf{v} \in \mathbb{R}^n$ that

$$
\|\mathbf{v}\|_1^2 \leq \rho \left(D^{-1} A \right) \|\mathbf{v}\|_0^2 < \eta \|\mathbf{v}\|_0^2 .
$$

Thus, if

$$
\eta \leq \left(\frac{2}{\omega} - \frac{\sigma}{\omega^2} \right), \tag{7.8}
$$

then (7.7) is satisfied (sufficient condition) and the damped Jacobi iteration fulfills the algebraic smoothing property. One obtains from (7.8)

$$
\sigma \leq 2\omega - \eta\omega^2 = \omega \left(2 - \omega\eta \right) .
$$

Obviously it is $\sigma > 0$ if $\omega \in (0, 2/\eta)$. ∎

Remark 7.15 *On the algebraic smoothing property.*
- The optimal value of ω, which gives the largest σ is $\omega^* = 1/\eta$, such that $\sigma = 1/\eta$. This statement can be proved by standard calculus, *exercise*.
- The algebraic smoothing property can be proved also for the Gauss–Seidel iteration.

□

Remark 7.16 *The algebraic smooth error for M-matrices.* The meaning of "$\mathbf{v}$ being an algebraic smooth error" will be studied in some more detail for symmetric positive definite M-matrices. This class of matrices was introduced in the course on numerical methods for convection-dominated problems.

An algebraic smooth error satisfies $\|\mathbf{v}\|_2 \ll \|\mathbf{v}\|_1$. By (7.2), this property implies

$$
\|\mathbf{v}\|_1 \ll \|\mathbf{v}\|_0 . \tag{7.9}
$$

For a symmetric matrix $A \in \mathbb{R}^{n \times n}$, it is, *exercise*,

$$
\|\mathbf{v}\|_1 = \frac{1}{2} \sum_{i,j=1}^n \left(-a_{ij} \right) \left(v_i - v_j \right)^2 + \sum_{i=1}^n s_i v_i^2, \quad \text{with } s_i = \sum_{j=1}^n a_{ij}
$$

being the i-th row sum of A. It follows from (7.9) that

$$\frac{1}{2} \sum_{i,j=1}^{n} \left(-a_{ij}\right) \left(v_i - v_j\right)^2 + \sum_{i=1}^{n} s_i v_i^2 \ll \sum_{i=1}^{n} a_{ii} v_i^2. \qquad (7.10)$$

Let A be an M-matrix. Then $a_{ij} \leq 0$, i.e., $|a_{ij}| = -a_{ij}$ for $i \neq j$. In many applications, it is $s_i = 0$. Then, from (7.10) it follows on the average for each i (consider just a fixed i)

$$\sum_{j=1}^{n} \frac{|a_{ij}|}{a_{ii}} \frac{\left(v_i - v_j\right)^2}{v_i^2} \ll 1.$$

In the sum, there are only nonnegative terms. Thus, if $|a_{ij}|/a_{ii}$ is large, then $\left(v_i - v_j\right)^2 / v_i^2$ has to be small such that the sum becomes small. One says, schemes which satisfy the smoothing property (7.5) smooth the error along the so-called strong connections, i.e., where $|a_{ij}|/a_{ii}$ is large, since for these connections a good smoothing can be expected on the given grid. This property implies that the corresponding nodes i and j do not need to be both on the coarse grid. □

7.3 Coarsening

Remark 7.17 *Goal.* Based on the matrix information only, one has to choose in the graph of the matrix nodes which become coarse nodes and nodes which stay on the fine grid. There are several strategies for coarsening. In this course, a standard way will be described. It will be restricted to the case that $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite M-matrix. □

Definition 7.18 Strong coupling. A variable (node) i is said to be strongly coupled to another variable j if

$$-a_{ij} \geq \varepsilon_{\text{str}} \max_{a_{ik} < 0} |a_{ik}|$$

with fixed $\varepsilon_{\text{str}} \in (0, 1)$. The set of all strong couplings of i is denoted by

$$S_i = \{j \in N_i \ : \ i \text{ is strongly coupled to } j\}.$$

The set S_i^T of strong transposed couplings of i consists of all variables j which are strongly coupled to i

$$S_i^T = \{j \in N_i \ : \ i \in S_j\}.$$

□

Remark 7.19 *On strong couplings.*
- Even for symmetric matrices, the relation of being strongly coupled is in general not symmetric. Consider, e.g.,

$$A = \begin{pmatrix} 5 & -1 & -0.1 \\ -1 & 3 & -0.1 \\ -0.1 & -0.1 & 3 \end{pmatrix}, \quad \varepsilon_{\text{str}} = 0.25.$$

  Then, one gets $S_1 = \{2\}, S_2 = \{1\}, S_3 = \{1, 2\}$, such that $S_1 = \{2, 3\}, S_2 = \{1, 3\}, S_3 = \emptyset$.
- The actual choice of ε_{str} is in practical computations not very critical. Values of around 0.25 are often used.

□

Remark 7.20 *Aspects of the coarsening process.* In the coarsening process, one has to pay attention to several aspects.

- The number of coarse nodes (C-nodes) should not be too large, such that the dimension of the coarse system is considerably smaller than the dimension of the fine system.
- Nodes i and j, which are strongly coupled, have a small relative error

$$(e_i - e_j)^2 / e_i^2$$

  such that a coarse grid correction of this error is not necessary. That means, it will be inefficient to define both nodes as coarse nodes.
- All fine nodes (F-nodes) should have a substantial coupling to neighboring C-nodes. In this way, the F-nodes obtain sufficient information from the C-nodes.
- The distribution of the C-nodes and F-nodes in the graph should be reasonably uniform.

$\square$

Remark 7.21 *A standard coarsening procedure.* A standard coarsening procedure starts by defining some first variable i to become a C-node. Then, all variables j that are strongly coupled with i, i.e., all $j \in S_i^T$, become F-nodes. Next, from the remaining undecided variables, another one is defined to become a C-node and all variables which are strongly coupled to it and are still undecided become F-nodes, and so on. This process stops if all variables are either C-nodes or F-nodes.

To obtain a uniform distribution of the C-nodes and F-nodes, the process of choosing C-nodes has to be done in a certain order. The idea consists in starting with some variable and to continue from this variable until all variables are covered. Therefore, an empirical "measure of importance" λ_i for any undecided variable to become a C-node is introduced

$$\lambda_i = \left| S_i^T \cap U \right| + 2 \left| S_i^T \cap F \right|, \quad i \in U, \tag{7.11}$$

where U is the current set of undecided variables, F the current set of F-nodes and $|\cdot|$ is the number of elements in a set. One of the undecided variables with the largest value of λ_i will become the next C-node. After this choice, all variables which are strongly coupled to the new C-node become F-nodes and for the remaining undecided variables, one has to update their measure of importance.

With the measure of importance (7.11), there is initially the tendency to pick variables which are strongly coupled with many other variables to become C-nodes, because $|U|$ is large and $|F|$ is small, such that the first term dominates. Later, the tendency is to pick as C-nodes those variables on which many F-nodes depend strongly, since $|F|$ is large and $|U|$ is small such that the second term in λ_i becomes dominant. Thus, the third point of Remark 7.20 is taken into account. $\square$

Example 7.22 *A standard coarsening procedure.* Consider a finite difference discretization of the Laplacian in the unit square with the five point stencil. Assuming that the values at the boundary are known, the finite difference scheme gives for the interior nodes i the following matrix entries, apart of a constant factor,

$$a_{ij} = \begin{cases} 4 & \text{if } i = j, \\ -1 & \text{if } j \text{ is left, right, upper, or lower neighbor of } i, \\ 0 & \text{else.} \end{cases}$$

Taking an arbitrary ε_{str}, then each node i is strongly coupled to its left, right, upper, and lower neighbor. Consider a 5×5 patch and choose some node as C-node. In

the first step, one obtains

$$
\begin{array}{ccccc}
U & U & U & U & U \\
U & U & U & U & U \\
U & U & F & U & U, \\
U & F & C & F & U \\
U & U & F & U & U
\end{array}
$$

where for U it is $\lambda_i = 2 + 2 \cdot 2 = 6$ and for U it is either $\lambda_i = 4 + 2 \cdot 0 = 4$ or $\lambda_i = 3 + 2 \cdot 1 = 5$. The next step gives, e.g.,

$$
\begin{array}{ccccc}
U & U & U & U & U \\
U & U & U & F & U \\
U & U & F & C & F, \\
U & F & C & F & U \\
U & U & F & U & U
\end{array}
$$

with $\lambda_i = 2 + 2 \cdot 2 = 6$ for U and $\lambda_i \leq 5$ else. Continuing this process leads to

$$
\begin{array}{ccccc}
U & U & F & U & U \\
U & F & C & F & U \\
F & C & F & C & F, \\
U & F & C & F & U \\
U & U & F & U & U
\end{array}
$$

and so on.

In this particular example, one obtains a similar coarse grid as given by a geometric multigrid method. However, in general, especially with non-symmetric matrices, the coarse grid of the AMG looks considerably different than the coarse grid of a geometric multigrid method. □

Remark 7.23 *On coarsening strategies.*
- In the standard coarsening scheme, none of the C-nodes is strongly coupled to any of the C-nodes created prior itself. However, since the relation of being strongly coupled might be non-symmetric, in particular for non-symmetric matrices, this property may not be true the other way around. Numerical experience shows that in any case the resulting set of C-nodes is close to a maximal set of variables which are not strongly coupled among each other.
- Other ways of coarsening can be found, e.g., in K. Stüben "Algebraic Multigrid (AMG): An introduction with applications", which is part of Trottenberg et al. (2001).

□

7.4 Prolongation

Remark 7.24 *Prolongation.* The last component of an AMG, which has to be defined, is the prolongation. It will be matrix-depend, in contrast to geometric multigrid methods. □

Remark 7.25 *Construction of an prolongation operator.* To motivate the construction of an prolongation operator, once more the meaning of an error to be algebraically smooth will be discussed. From the geometric multigrid methods, it is known that the prolongation has to work well for smooth functions, see Remark 4.11. By definition, an algebraic smooth error is characterized by

$$
S\mathbf{e} \approx \mathbf{e}
$$

or

$$\|\mathbf{e}\|_2 \ll \|\mathbf{e}\|_1.$$

In terms of the residual

$$\mathbf{r} = \mathbf{f} - A\mathbf{v} = A\mathbf{u} - A\mathbf{v} = A(\mathbf{u} - \mathbf{v}) = A\mathbf{e},$$

this inequality means that

$$\left(D^{-1}A\mathbf{e}, A\mathbf{e}\right) \ll (A\mathbf{e}, \mathbf{e}) \quad \Longleftrightarrow \quad \left(D^{-1}\mathbf{r}, \mathbf{r}\right) \ll (\mathbf{r}, \mathbf{e}).$$

One term in both inner products is the same. One can interprete this inequality in the way that on the average, algebraic smooth errors are characterized by a scaled residual (first argument on the left-hand side) to be much smaller than the error (second argument on the right-hand side). On the average, it follows that

$$a_{ii}^{-1}r_i^2 \ll |r_i e_i| \quad \Longleftrightarrow \quad |r_i| \ll a_{ii}|e_i|.$$

Thus, $|r_i|$ is close to zero and one can use the approximation

$$0 \approx r_i = a_{ii}e_i + \sum_{j \in N_i} a_{ij}e_j. \tag{7.12}$$

Let i be a F-node and $P_i \subset C_{\mathrm{nod}}$ a subset of the C-nodes, where the set of C-nodes is denoted by C_{nod}, the so-called interpolatory points. The goal of the prolongation consists in obtaining a good approximation of e_i using information from the coarse grid, i.e., from the C-nodes contained in P_i. Therefore, one likes to compute prolongation weights ω_{ij} such that

$$e_i = \sum_{j \in P_i} \omega_{ij}e_j \tag{7.13}$$

and e_i is a good approximation for any algebraic smooth error which satisfies (7.12). □

Remark 7.26 *Direct prolongation.* Here, only the so-called direct prolongation in the case of A being an M-matrix will be considered. Direct prolongation means that $P_i \subset C_{\mathrm{nod}} \cap N_i$, i.e., the interpolatory nodes are a subset of all coarse nodes which are coupled to i. Inserting the ansatz (7.13) into (7.12) gives

$$e_i = \sum_{j \in P_i} \omega_{ij}e_j = -\frac{1}{a_{ii}} \sum_{j \in N_i} a_{ij}e_j. \tag{7.14}$$

If $P_i = N_i$, then the choice $\omega_{ij} = -a_{ij}/a_{ii}$ will satisfy this equation. But in general, $P_i \subsetneq N_i$. If there are sufficiently many nodes which are strongly connected to i contained in P_i, then for the averages it holds

$$\frac{1}{\sum_{j \in P_i} a_{ij}} \sum_{j \in P_i} a_{ij}e_j \approx \frac{1}{\sum_{j \in N_i} a_{ij}} \sum_{j \in N_i} a_{ij}e_j.$$

Inserting this relation into (7.14) leads to the proposal for using matrix-dependent prolongation weights

$$\omega_{ij} = -\left(\frac{\sum_{k \in N_i} a_{ik}}{\sum_{k \in P_i} a_{ik}}\right)\frac{a_{ij}}{a_{ii}} > 0, \quad i \in F, j \in P_i.$$

Summation of the weights gives

$$\sum_{j \in P_i} \omega_{ij} = -\left(\frac{\sum_{k \in N_i} a_{ik}}{\sum_{k \in P_i} a_{ik}}\right)\frac{\sum_{j \in P_i} a_{ij}}{a_{ii}} = \frac{a_{ii} - s_i}{a_{ii}} = 1 - \frac{s_i}{a_{ii}},$$

where s_i is the sum of the i-th row of A. Thus, if $s_i = 0$, then $\sum_{j \in P_i} \omega_{ij} = 1$ such that constants are prolongated exactly. □

7.5 Concluding Remarks

Example 7.27 *Behavior of an AMG for the Poisson equation.* The same situation as in Example 2.5 will be considered. In the code MooNMD, a simple AMG is implemented. The number of iterations and computing times for applying this method as solver or as preconditioner in a flexible GMRES method are presented in Table 7.1.

Table 7.1: Example 7.27. Number of iterations and computing times (14/01/24 on a HP BL460c Gen8 2xXeon, Eight-Core 2700MHz). The number of degrees of freedom (d.o.f.) includes the Dirichlet values. The time for the setup of the AMG is included into the total solution time.

level	h	d.o.f.	AMG		FGMRES+AMG		setup time
			ite	time	ite	time	(FGMRES+AMG)
1	1/4	25	1	0	1	0	0
2	1/8	81	1	0	1	0	0
3	1/16	289	34	0.01	18	0.01	0
4	1/32	1089	41	0.02	19	0.01	0.01
5	1/64	4225	45	0.13	21	0.08	0.03
6	1/128	16641	47	0.69	22	0.43	0.15
7	1/256	66049	51	3.81	23	2.66	1.32
8	1/512	263169	49	25.08	24	14.82	7.28
9	1/1024	1050625	50	157.14	24	119.96	84.95
10	1/2048	4198401	50	1500.75	24	1333.09	1103.40

It can be seen, that using AMG as preconditioner is more efficient than using it as solver. The number of iterations for both applications of AMG is constant independently of the level. However, the solution time does not scale with the number of degrees of freedom. The reason is that in the used implementation, the time for constructing the AMG does not scale in this way but much worse. Comparing the results with Table 2.2, one can see that AMG is not competitive with a geometric multigrid method, if the geometric multigrid method works well.

□

Remark 7.28 *Concluding remarks.*
- A number of algebraic results for AMGs is available, see the survey paper of K. Stüben. But there are still many open questions, even more than for the geometric multigrid method.
- As seen in Example 7.27, in problems for which a geometric multigrid method can be applied efficiently, the geometric multigrid method will in general outperform AMG. But there are classes of problems for which AMG is as efficient or even more efficient than a geometric multigrid method. One of the most important fields of application for AMG are problems for which a geometric multigrid method cannot be performed.

□

Chapter 8

Outlook

Remark 8.1 *More general problems.*
- There are only few contributions to the analysis of multigrid methods for problems which are not symmetric positive definite or a slight perturbation of such problems. One example where it is nothing proved are linear convection-diffusion equations which are convection-dominated. However, the practical experience is that multigrid solvers, with appropriate preconditioners, work reasonably well for such problems.
- The key for the efficiency of the multigrid method is generally the smoother. There is a lot of experience for scalar problems, e.g., for convection-diffusion problems often SSOR or ILU work reasonably well, see Example 8.2. For coupled problems, sometimes the construction of smoothers is already complicated. For instance, many discretizations for the Navier–Stokes equations lead to matrices where a number of diagonal entries are zero. In this case, one cannot apply classical iterative schemes since these schemes require the division by the diagonal entries.
- Algebraic multigrid methods are usually applied to scalar problems. There are only few proposals of algebraic multigrid methods for coupled problems.
- The extension of the multigrid idea to nonconforming finite element discretizations is possible.

$\square$

Example 8.2 *Convection-diffusion problem in two dimensions.* A standard convection-diffusion test problem in two dimensions has the form

$$\begin{aligned} -\varepsilon \Delta u + (1,0)^T \cdot \nabla u &= 1 \quad \text{in } \Omega = (0,1)^2, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

see the lecture notes of the course on numerical methods for convection-dominated problems. Considering $\varepsilon = 10^{-8}$ with the Q_1 finite element method and the SUPG stabilization, then one obtains the iterations and computing times as shown in Table 8.1. In these simulations, the multigrid methods were applied with the F-(ν, ν)-cycle, where ν is the number of pre and post smoothing steps. In the geometric multigrid method, a SSOR smoother was used and in the algebraic multigrid method, a ILU smoother.

Table 8.1: Example 8.2. Number of iterations and computing times (14/01/23 on a HP BL460c Gen8 2xXeon, Eight-Core 2700MHz). The number of degrees of freedom (d.o.f.) includes the Dirichlet values.

level	h	d.o.f.	FGMRES+MG, F(3,3)		FGMRES+MG F(10,10)		FGMRES+AMG F(3,3)		FGMRES+AMG F(5,5)		UMFPACK	
			ite	time	ite	time	ite	time	ite	time	ite	time
0	1/16	289	1	0	1	0	6	0	4	0	1	0
1	1/32	1089	6	0.03	2	0.01	8	0.05	5	0.05	1	0.01
2	1/64	4225	9	0.10	3	0.05	11	0.42	7	0.41	1	0.02
3	1/128	16641	15	0.44	5	0.25	17	3.19	12	4.64	1	0.16
4	1/256	66049	26	2.68	9	1.73	30	34.32	23	42.20	1	1.35
5	1/512	263169	47	20.09	16	8.59	no conv.		149	866.58	1	10.27
6	1/1024	1050625	145	252.80	29	66.69			no conv.		1	75.17
7	1/2048	4198401	302	2057.39	76	838.18						

One can see in Table 8.2 that none of the solvers behaves optimal, i.e,. for none of the solvers, the computing time scales with the number of degrees of freedom. The most efficient solvers in this example are the direct solver (note that this is a two-dimensional problem) and the geometric multigrid as preconditioner with sufficiently many smoothing steps. On the finest grid, only the geometric multigrid approaches work since the direct solver terminates because of internal memory limitations. In the multigrid methods, one can well observe the effect of increasing the number of smoothing steps.

Altogether, the linear systems obtained for convection-dominated problem are usually hard to solve and so far an optimal solver is not known. □

Remark 8.3 *Multigrid methods with different finite element spaces.* One can apply the multigrid idea also with different (finite element) spaces. For instance, consider just one grid. As coarse grid space, one can use P_1 finite elements and as fine grid space P_2 finite elements. With these two finite element spaces, one can perform a two-level method.

This idea has been used in the construction of finite element spaces for higher order finite elements. It is known from numerical studies that multigrid methods with the same finite element space on all levels might become inefficient for higher order elements because it is hard to construct good smoothers. On the other hand, multigrid methods are usually more efficient for lower order elements. The idea consists in using on the fine grid the higher order finite element space as the finest level of the multigrid hierarchy and using as next coarser level of this hierarchy a first order finite element space on the same geometric grid. On the coarser geometric grids, one uses also low order finite elements. In this way, one has a multigrid method for the higher order discretization which uses low order discretizations on the coarser grids. Some good experience with this approach is reported in the literature. □

Remark 8.4 *Simulations in practice.* The great difficulty of the application of multigrid methods for problems from practice comes from the situation that in practice the domains are often complicated. A good initial triangulation of a complicated domain leads already to a fine mesh. Often, the computational resources can just afford this mesh such that there is no mesh hierarchy available. Also, generally (in industrial codes) there is only one type of discrete space, e.g., P_1 finite elements, available. Altogether, in this situation one has to use a different solver. □

Bibliography

Briggs, W., E. Henson, and S. McCormick, 2000: *A multigrid tutorial.* 2d ed., SIAM, Philadelphia, PA.

Davis, T. A., 2004: Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, **30 (2)**, 196–199, doi:10.1145/992200.992206, URL `http://dx.doi.org/10.1145/992200.992206`.

Hackbusch, W., 1985: *Multi-Grid Methods and Applications.* Springer Series in Computational Mathematics, 4, Springer-Verlag, Berlin-Heidelberg-New York.

Hackbusch, W., 1994: *Iterative solution of large sparse systems of equations*, Applied Mathematical Sciences, Vol. 95. Springer-Verlag, New York, xxii+429 pp., doi:10.1007/978-1-4612-4288-8, URL `http://dx.doi.org/10.1007/978-1-4612-4288-8`, translated and revised from the 1991 German original.

John, V. and G. Matthies, 2004: MooNMD—a program package based on mapped finite element methods. *Comput. Vis. Sci.*, **6 (2-3)**, 163–169, doi:10.1007/s00791-003-0120-1, URL `http://dx.doi.org/10.1007/s00791-003-0120-1`.

Olshanskii, M. A. and A. Reusken, 2004: Convergence analysis of a multigrid method for a convection-dominated model problem. *SIAM J. Numer. Anal.*, **42 (3)**, 1261–1291, doi:10.1137/S0036142902418679, URL `http://dx.doi.org/10.1137/S0036142902418679`.

Reusken, A., 2002: Convergence analysis of a multigrid method for convection-diffusion equations. *Numer. Math.*, **91 (2)**, 323–349, doi:10.1007/s002110100312, URL `http://dx.doi.org/10.1007/s002110100312`.

Shaidurov, V., 1995: *Multigrid Methods for Finite Elements*, Mathematics and Its Application, Vol. 318. Kluwer Academic Publisher.

Trottenberg, U., C. W. Oosterlee, and A. Schüller, 2001: *Multigrid.* Academic Press Inc., San Diego, CA, xvi+631 pp., with contributions by A. Brandt, P. Oswald and K. Stüben.

Wesseling, P., 1992: *An Introduction to Multigrid Methods.* John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, corrected reprint by R.T. Edwards, Inc. 2004.

Index