Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Multi-Grid Methods for FEM and BEM Applications

by

Wolfgang Hackbusch

Preprint no.: 72 2003

Multi-Grid Methods for FEM and BEM Applications*

Wolfgang Hackbusch
Max-Planck-Institut *Mathematik in den Naturwissenschaften*
Inselstr. 22–26, D-04103 Leipzig, Germany
wh@mis.mpg.de

Abstract

After discretisation of the partial differential equations from mechanics one usually obtains large systems of (non)linear equations. Their efficient solution requires the use of fast iterative methods. Multi-grid iterations are able to solve linear and nonlinear systems with a rather fast rate of convergence, provided the problem is of elliptic type. The contribution describes the basic construction of multigrid methods, their ingredients, related methods, and their application to various problem classes.

Although most of the applications concern FEM discretisations of partial differential equations, there are also applications to integral equations as they occur in boundary element methods (BEM).

Keywords. Multi-grid method, two-grid method, nested iteration, systems of linear equations, systems of nonlinear equations, integral equations, eigenvalue problems, FEM, BEM

1 General Remarks on Multi-Grid Methods

1.1 Introduction

The solution of large systems of linear or even nonlinear equations is a basic problem when partial differential equations are discretised. Examples are the finite element equations in continuum or fluid dynamic problems. Since the dimension of these systems is often only limited by the available computer storage, the size of the arising systems is increasing due to the advances in computer technology. At the moment, systems with several millions of equations are of interest. The solution of such systems requires numerical methods which have a runtime proportional to the dimension of the system (so-called 'linear complexity' or 'optimal complexity'). Immediately when computers started to be available, one tried to find more efficient solvers. Multi-grid methods happened to be the first ones which reached the linear complexity for a rather large class of problems.

Due to its naming, multi-grid methods involve several 'grids'. The nature of this grid hierarchy is explained below in a general setting. A simple one-dimensional example can be found in §1.2.5.

1.1.1 The Standard Problem Structure

The situation we consider in the following is illustrated in diagram (1.1):

$$\begin{array}{rcl} \mathcal{P} & = & \mathcal{P}_{\text{continuous}} \\ & \downarrow & \text{discretisation process} \\ \mathcal{P}_{\text{discrete}} & = & \mathcal{P}_\ell \quad \leftrightarrows \quad \mathcal{P}_{\ell-1} \quad \leftrightarrows \ldots \leftrightarrows \quad \mathcal{P}_0. \end{array} \tag{1.1}$$

$\mathcal{P}_{\text{discrete}}$ is a given (discrete, i.e., finitely dimensional) algebraic problem. The most prominent example of such a problem is a *system of linear equations*. Therefore, the discussion of the corresponding linear multi-grid methods will fill the major part of this contribution. However, from the practical point of view, *nonlinear systems* may be a more important example of $\mathcal{P}_{\text{discrete}}$. Another example are *eigenvalue problems* $\mathcal{P}_{\text{discrete}}$. It is essential for the multi-grid approach to embed the discrete problem $\mathcal{P}_{\text{discrete}}$ into a *hierarchy* of discrete problems $\mathcal{P}_\ell, \mathcal{P}_{\ell-1}, \ldots, \mathcal{P}_0$, where the problem size (dimension) of $\mathcal{P}_k$ is increasing with increasing level number k. The largest dimension corresponds to $\mathcal{P}_{\text{discrete}} = \mathcal{P}_\ell$, while the lower dimensional problems $\mathcal{P}_{\ell-1}, \ldots, \mathcal{P}_0$ are used as auxiliary problems. The role of the continuous problem $\mathcal{P} = \mathcal{P}_{\text{continuous}}$ will be discussed below.

*To appear as Chapter 19 of Vol. I in Erwin Stein, René de Borst, and Thomas J.R. Hughes (eds.): *Encyclopedia of Computational Mechanics*. Wiley, Chichester

1.1.2 What are Multi-Grid Methods?

The multi-grid method can be considered as a *concept for the construction of fast iterative methods* based on a hierarchy $\mathcal{P}_\ell, \mathcal{P}_{\ell-1}, \ldots, \mathcal{P}_0$ for solving an algebraic problem $\mathcal{P}_{\text{discrete}} = \mathcal{P}_\ell$. To have an analogue, we may look at the finite element method. The finite element method is not a special discretisation of a particular problem but offers a whole class of discretisations (elements of different shape and order) to various problems (differential as well as integral equations). Similarly, the multi-grid technique is applicable to large classes of (discrete algebraic) problems and describes how to construct algorithms for solving these problems. Instead of 'multi-grid method' also other names are in use, e.g., 'multiple grid method'. The name states that several 'grids' are involved, which belong to different levels of a hierarchy, as indicated in (1.1). This also explains the alternative namings 'multi-level method' or 'multiple level method'. The word 'grid' in 'multi-grid method' originates from finite difference approximations by means of a regular grid. However, this does not mean that the multi-grid method is restricted to such discretisations. In fact, the major part of this contribution will refer to the finite element discretisation. Finite differences, finite elements or finite volumes are examples for the discretisation process in (1.1).

1.1.3 Which Problems can be Solved?

The continuous problem $\mathcal{P} = \mathcal{P}_{\text{continuous}}$ can be connected with a partial differential or integral equation of elliptic type.

Usually, the problems $\mathcal{P}_{\text{discrete}}$ to be solved by multi-grid are discrete analogues of a continuous problem $\mathcal{P} = \mathcal{P}_{\text{continuous}}$ derived by some discretisation indicated by $\downarrow$ in (1.1).

A given discretisation process offers an easy possibility to obtain not only one particular discrete problem $\mathcal{P}$ but a whole hierarchy of many discrete analogues $\mathcal{P}_k$ $(k = \ell, \ell-1, ..., 0)$ of the continuous problem corresponding to different dimensions. This hierarchy is needed in the multi-grid iteration.

The essence of the multi-grid method is the so-called 'coarse grid correction'. This requires the existence of a lower dimensional problem $\mathcal{P}_{\ell-1}$ which approximates the given problem $\mathcal{P}_\ell$ *in a reasonable way*. If the discretisation process is not given in advance, it must be created as a part of the multi-grid process.

On the other hand, it is often very hard to apply multi-grid, when the desired solution cannot be represented by a lower dimensional approximation. An example are boundary value problems with a geometrically complicated boundary (see end of §1.1.6)

1.1.4 Why is Multi-Grid Optimal?

If a multi-grid method is successful, the iterative process has the following characteristics:

(i) The convergence speed is uniform with respect to the problem size. Therefore, differently from simple iterative methods which become slower with increasing dimension, these algorithms can be used for large scale problems. The desired accuracy can be obtained by a fixed number of multi-grid steps. The resulting method is of linear complexity.

(ii) Due to the hopefully good approximation of problem $\mathcal{P}_\ell$ by $\mathcal{P}_{\ell-1}$, the convergence speed is expected to be much better (i.e., smaller) than 1. This implies that only very few iteration steps are necessary to obtain the desired accuracy.

The characteristics given above, describe the aim we want to obtain with a good multi-grid implementation. In fact, optimal convergence can be guaranteed in rather general cases. Here, 'general' means that no special algebraic properties of the matrix are required. In particular, the matrix may be neither symmetric nor positive definite.

Nevertheless, there is no easy way to get such results in *any* case. For singularly perturbed problems (e.g., problems with high Reynolds numbers) the convergence speed $\rho = \rho(\delta)$ may depend on a parameter δ. If $\rho(\delta)$ approaches 1 in the limit $\delta \to 0$ (or $\delta \to \infty$), the fast convergence promised in (ii) is lost. These difficulties give rise to the investigation of the so-called *robust multi-grid methods*, which by definition lead to uniform convergence rates $\rho(\delta)$.

1.1.5 Is Multi-Grid Easy to Implement?

As seen above, multi-grid methods require an environment of a hierarchy consisting of the problems $\mathcal{P}_k$ $(k = \ell, \ell-1, ..., 0)$ together with interacting mappings between neighbouring levels $k, k-1$ (these interactions are denoted by $\leftrightarrows$ in (1.1)). If this environment is present in the implementation anyway (e.g., as a part of the adaptive refinement process), the multi-grid method is rather easy to implement. If, however, only the problem description of $\mathcal{P}_{\text{discrete}}$ is given, the auxiliary problems $\mathcal{P}_k$ for $k < \ell$ together with the interactions $\leftrightarrows$

must be installed as a part of the multi-grid method, which of course makes the multi-grid implementation much more involved.

1.1.6 The Hierarchy of Problems

There are different possibilities to produce a hierarchy of discretisations. For discretisation methods based on *regular grids*, the hierarchy of grids induces the necessary hierarchy of discrete problems $\mathcal{P}_k$. In the case of finite element methods, the underlying triangulation replaces the grid. A hierarchy of nested triangulations yields a perfect problem hierarchy. Such a hierarchy may be the side-product of adaptive mesh refinement. In this case one proceeds from the coarsest to the finest level. The problem $\mathcal{P}_0$ corresponding to the coarsest mesh size should have a dimension small enough to be solved by standard methods. This can cause severe difficulties for complicated boundary value problems which need a high number of degrees of freedom (see, e.g., [16] and §3.4).

1.1.7 Notations

The norm $\|\cdot\|$ is the Euclidean norm when applied to vectors and the spectral norm for matrices (i.e., $\|\mathbf{A}\| := \max\{\|\mathbf{Ax}\| / \|\mathbf{x}\| : \mathbf{x} \neq \mathbf{0}\}$. $\langle\cdot,\cdot\rangle_k$ denotes the Euclidean scalar product of the vector space $\mathcal{U}_k$ (see below). The scalar product in $L^2(\Omega)$ is denoted by $(f,g)_{L^2(\Omega)} := \int_\Omega fgd\mathbf{x}$.

The Landau symbol $\mathcal{O}(f(x))$ means that the quantity is bounded by $C * f(x)$ for some constant C, when x tends to its characteristic limit (e.g., $x = h \to 0$ for the step size h or $x = n \to \infty$ for the dimension n). Further notations are explained below.

item	explanation	reference
$\mathbf{d}_k$	defect $\mathbf{L}_k\mathbf{u}_k - \mathbf{f}_k$	(2.2)
$\mathbf{f}_k$	vector of right-hand side (at level k)	(1.3)
h_k	grid size, mesh size	§1.2.1
$\mathcal{H}_k, \mathcal{H}^{\mathrm{FEM}} \subset \mathcal{H}$	finite element space $\subset$ energy space	§4, (4.4)
k, ℓ	level index	§1.2.1
$\mathbf{L}_k$	stiffness matrix of level k	(1.3), §4.1, §4.2
$\mathbf{M}_\ell, \mathbf{M}_\ell^{\mathrm{TGM}}, \ldots$	iteration matrix	§2.3
p	prolongation	(1.5)
P, P_k	bijection onto FE space (of level k)	§4.1, §4.2
r	restriction	(1.6)
$\mathbf{S}_k$	iteration matrix corresponding to $\mathcal{S}_k$	§2.1
$\mathcal{S}_k, \mathcal{S}_k^\nu$	smoothing iteration, ν-fold application of $\mathcal{S}_k$	§1.2.2, §2.2
$\mathcal{T}, \mathcal{T}_k$	triangulation (at level k)	§4.1, §4.2
$\mathbf{u}, \mathbf{u}_k$	solution vector (at level k)	(1.3), §4.2
u^k	finite element function at level k from $\mathcal{H}_k$	§4.3
$\mathcal{U}, \mathcal{U}_k$	linear space of vectors $\mathbf{u}, \mathbf{u}_k$	(1.5), §4.1
γ	$\gamma = 1$: V-cycle, $\gamma = 2$: W-cycle	(3.2)
$\delta_{\alpha\beta}$	$\delta_{\alpha\beta} = 0$ for $\alpha = \beta$, $\delta_{\alpha\beta} = 1$ otherwise	Kronecker symbol
ζ, ζ_k	contraction number	§2.3, (6.5)
$\rho(\cdot)$	spectral radius of a matrix	§2.3
Ω	domain of boundary value problem	§4
Ω_k	grid of level k, set of nodal values	§1.2.5

1.1.8 Literature

The first two-grid method is described by Brakhage [5] for the solution of an integral equation (see §9.1). The first two-grid method for the Poisson equation is introduced by Fedorenko [9], while [10] contains the first *multi*-grid method. The first more general convergence analysis is given by Bakhvalov [1].

Since there is a vast literature about multi-grid methods, we do not try to give a selection of papers. Instead we refer to the monographs [12], [25] (in particular devoted to problems of fluid dynamics), [6], [24] (and the literature cited therein) and to the proceedings of the *European Multigrid Conferences* [17] (containing an introduction to multi-grid), [18], [19], [21], [20].

1.2 Ingredients of Multi-Grid Iterations

1.2.1 Hierarchy of Linear Systems

We consider the solution of a linear system

$$\mathbf{L}_h \mathbf{u}_h = \mathbf{f}_h, \tag{1.2}$$

where h refers to the smallest mesh size $h = h_\ell$ of a hierarchy of mesh sizes $h_0 > h_1 > \ldots > h_{\ell-1} > h_\ell$. We write the linear system corresponding to the level k (mesh size h_k) as

$$\mathbf{L}_k \mathbf{u}_k = \mathbf{f}_k \qquad \text{for } k = \ell, \ell-1, ..., 0. \tag{1.3}$$

In particular, $\mathbf{L}_\ell \mathbf{u}_\ell = \mathbf{f}_\ell$ is an identical writing for (1.2).

1.2.2 Smoothing Iteration

Classical iterative methods like the Jacobi or the Gauss-Seidel iteration are needed for all levels $k > 0$. The purpose is not to reduce the iteration error but to produce smoother errors. This is the reason for the name 'smoothing iteration' or 'smoother'. The smoother is applied to the old iterate $\mathbf{u}_k^{\text{old}}$ and the right-hand side $\mathbf{f}_k$ and produces the new iterate $\mathbf{u}_k^{\text{new}} = \mathcal{S}_k(\mathbf{u}_k^{\text{old}}, \mathbf{f}_k)$.

To give an example we mention the simplest smoother, the *Richardson iteration*, which may be chosen in several cases,

$$\mathcal{S}_k : (\mathbf{u}_k, \mathbf{f}_k) \mapsto \mathbf{u}_k - \frac{1}{\|\mathbf{L}_k\|}(\mathbf{L}_k \mathbf{u}_k - \mathbf{f}_k). \tag{1.4}$$

1.2.3 Prolongations

We denote the space of vectors $\mathbf{u}_k$ and $\mathbf{f}_k$ by $\mathcal{U}_k$. The prolongation $p_{k,k-1}$ is a linear transfer from $\mathcal{U}_{k-1}$ to $\mathcal{U}_k$,

$$p_{k,k-1} : \mathcal{U}_{k-1} \to \mathcal{U}_k \quad \text{linear.} \tag{1.5}$$

In the following, we omit the indices $k, k-1$ and write $p : \mathcal{U}_{k-1} \to \mathcal{U}_k$.

1.2.4 Restrictions

The restriction $r_{k-1,k}$ is a linear transfer in the opposite direction,

$$r_{k-1,k} : \mathcal{U}_k \to \mathcal{U}_{k-1} \quad \text{linear.} \tag{1.6}$$

Again, we appreviate $r_{k-1,k}$ by r.

1.2.5 A One-Dimensional Example

Consider the 1D Dirichlet boundary value problem

$$-u''(x) = f(x) \quad \text{in } \Omega = (0,1)\,, \qquad u(x) = 0 \quad \text{at } x \in \Gamma = \{0,1\}\,. \tag{1.7}$$

Given a mesh size $h = 1/(n+1)$ with n being a power of 2, we define the grid Ω_h consisting of the points $x_\nu = \nu h$ $(\nu = 1, ..., n)$. The hierarchy of step sizes is

$$h_0 > h_1 > h_2 > ... > h_\ell = h \qquad \text{with } h_k = 2^{-1-k}. \tag{1.8}$$

Note that $h_0 = 1/2$ is the largest possible step size. The number of grid points in $\Omega_k = \{x_\nu = \nu h_k : 1 \le \nu \le n_k\}$ is $n_k = 2^{1+k} - 1$ (see Figure 1.1).

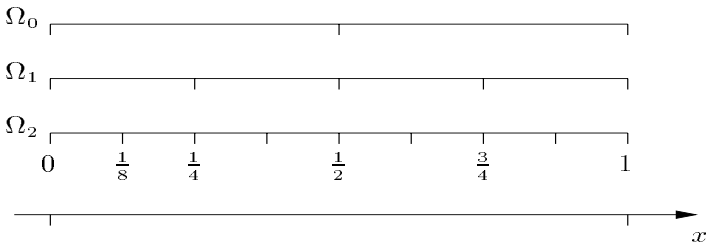

Figure 1.1: Grid hierarchy

The standard difference method yields the equations $h_k^{-2}\left[-u_{\nu-1}+2u_\nu-u_{\nu+1}\right]=f(\nu h_k)$ for $\nu=1,\ldots,n_k$ with boundary values $u_0=u_{n_k+1}=0$. The matrices $\mathbf{L}_k$ from (1.3) are

$$\mathbf{L}_k = h_k^{-2}\,\mathrm{tridiag}\,\{-1,2,-1\} \qquad (k=0,\ldots,\ell) \tag{1.9}$$

of size $n_k\times n_k$, while the vectors are $\mathbf{u}_k=(u_\nu)_{\nu=1}^{n_k}$, $\mathbf{f}_k=(f(\nu h_k))_{\nu=1}^{n_k}$.

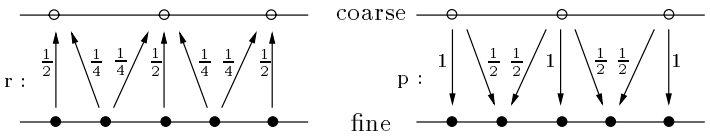

Figure 1.2: Prolongation and restriction in the 1D case

The prolongation $p=p_{k,k-1}$ from (1.5) can be realised by piecewise linear interpolation:

$$(p\,v_{k-1})(x) := \begin{cases} v_{k-1}(x) & \text{if } x\in\Omega_{k-1}\subset\Omega_k,\\ \dfrac{v_{k-1}(x-h_k)+v_{k-1}(x+h_k)}{2} & \text{if } x\in\Omega_k\backslash\Omega_{k-1}\end{cases} \tag{1.10}$$

(see Figure 1.2). p is described by the left rectangular matrix of size $n_k\times n_{k-1}$ in

$$p=\frac{1}{2}\begin{bmatrix} 1 & & & \\ 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & & \\ & & \ddots & \end{bmatrix}, \quad r=\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & & \\ & & 1 & 2 & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & 2 & 1 \end{bmatrix}. \tag{1.11}$$

The restriction $r=r_{k-1,k}$ from (1.6) is defined by the following weighted mean value

$$(r\mathbf{v}_{k-1})(x)=\tfrac{1}{4}\left[\mathbf{v}_k(x-h_k)+2\mathbf{v}_k(x)+\mathbf{v}_k(x+h_k)\right] \qquad \text{for } x\in\Omega_{k-1} \tag{1.12}$$

(see Figure 1.2). The corresponding rectangular matrix of size $n_{k-1}\times n_k$ is the second in (1.11).

A suitable smoother is Jacobi's iteration damped by the factor $\frac{1}{2}$: $\mathbf{u}_k^{j+1}=\mathcal{S}_k(\mathbf{u}_k^j,\mathbf{f}_k)$ with

$$\mathcal{S}_k(\mathbf{u}_k,\mathbf{f}_k) := \mathbf{u}_k-\frac{1}{2}\mathbf{D}_k^{-1}\left(\mathbf{L}_k\mathbf{u}_k-\mathbf{f}_k\right), \tag{1.13}$$

where $\mathbf{D}_k$ is the diagonal of $\mathbf{L}_k$, i.e. $\mathbf{D}_k=2h_k^{-2}\mathbf{I}$ (cf. (1.9)). Note that $\mathcal{S}_k$ from (1.13) is almost identical to (1.4), since $\frac{1}{\|\mathbf{L}_k\|}\approx\frac{1}{4}h_k^2$.

2 Two-Grid Iteration

The two-grid iteration is a preversion of the multi-grid iteration. It is not of practical interest, but it is an important building block which involves only the fine grid and one coarse grid.

In the following we explain the so-called *smoothing effect* of certain classical iterations. This gives rise to the smoothing iteration mentioned in §1.2.2.

2.1 Smoothing Effect

Usually, the purpose of an iteration $\mathbf{u}_k^0\mapsto\mathbf{u}_k^1\mapsto\ldots\mapsto\mathbf{u}_k^j$ is the fast convergence to the true solution, i.e., that the error $\mathbf{u}_k^j-\mathbf{u}_k$ decreases quickly. The purpose of an smoothing iteration is different. The error $\mathbf{u}_k^j-\mathbf{u}_k$ may stay as large as the starting error $\mathbf{u}_k^0-\mathbf{u}_k$, but it must become smoother (note that the error $\mathbf{u}_k^j-\mathbf{u}_k$ must become smooth; this does not concern the smoothness of the solution $\mathbf{u}_k$). The details are exemplified for the 1D example from §1.2.5.

We consider the *damped Jacobi iteration* (1.13) since its analysis is the most transparent one. Since $\mathbf{D}_k^{-1}=\frac{1}{2}h_k^2\mathbf{I}$, the damped Jacobi iteration equals

$$u_k^{j+1}=\mathbf{u}_k^j-\omega h_k^2\left(\mathbf{L}_k\,\mathbf{u}_k-\mathbf{f}_k\right)=\mathbf{S}_k\mathbf{u}_k^j+\omega h_k^2\mathbf{f}_k \qquad \text{with } \omega=\tfrac{1}{4}, \tag{2.1}$$

where $\mathbf{S}_k=\mathbf{I}-\omega h_k^2\mathbf{L}_k$ is called the *iteration matrix*. Note that the eigenvectors of $\mathbf{L}_k$ are

$$\mathbf{e}_k^\mu=\sqrt{2h_k}\left(\sin\left(\nu\mu\pi h_k\right)\right)_{\nu=1}^{n_k} \qquad \text{for } \mu=1,\ldots,n_k$$

(μ: frequency). The corresponding eigenvalues of $\mathbf{L}_k$ are $\lambda_{k,\mu} = 4h_k^{-2}\sin^2(\mu\pi h_k/2)$, i.e., $\mathbf{L}_k\,\mathbf{e}_k^\mu = \lambda_{k,\mu}\mathbf{e}_k^\mu$. Since $\mathbf{S}_k = \mathbf{I} - \omega h_k^2\mathbf{L}_k$ is the iteration matrix of the damped iteration (2.1), it has the same eigenvectors $\mathbf{e}_k^\mu$ as $\mathbf{L}_k$ and (for $\omega = \frac{1}{4}$) the eigenvalues

$$\lambda_\mu = 1 - \sin^2(\mu\pi h_k/2) = \cos^2(\mu\pi h_k/2) \qquad \left(1 \le \mu \le n_k = h_k^{-1} - 1\right)$$

shown in Figure 2.1. The choice $\omega = 1/2$ yields the standard Jacobi (dashed line in Figure 2.1).

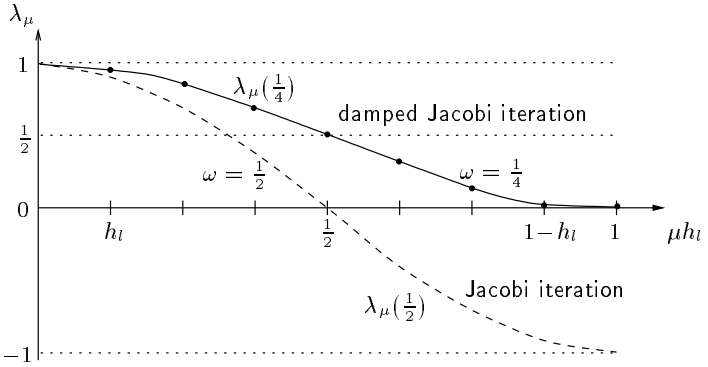

Figure 2.1: Eigenvalues of the iteration matrix as function of frequency μh_ℓ

The rate of convergence of the damped Jacobi iteration is $\lambda_1 = \cos^2(\pi h_k/2) = 1 - \frac{1}{4}\pi^2 h_k^2 + \mathcal{O}\left(h_k^4\right)$ proving the very slow convergence of the Jacobi iteration.

Even though iteration (2.1) converges very slowly, Figure 2.1 shows that components $\mathbf{e}_k^\mu$ with frequency $\mu \ge 1/(2h_k)$ are reduced at least by a factor $\frac{1}{2}$ per iteration. This means that the convergence rate of the damped Jacobi iteration restricted to the subspace $\operatorname{span}\{\mathbf{e}_k^\mu : 1/2 \le \mu h_k < 1\}$ of the *high frequencies* is 1/2. The iteration is rapidly convergent with respect to the high frequencies. The slow convergence is caused by the lower frequencies only. The construction of the multi-grid iteration is based on this observation.

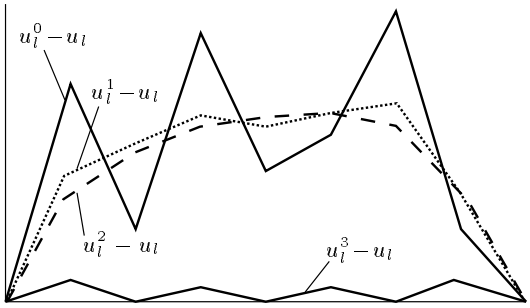

Figure 2.2: Smoothing effect of the damped Jacobi iteration. u_ℓ: exact discrete solution; $u_\ell^0, u_\ell^1, u_\ell^2$: smoothing iterates; u_ℓ^3: result after coarse-grid correction

The initial error $\mathbf{u}_k^0 - \mathbf{u}_k$ can be represented by the linear combination $\sum \alpha_\mu \mathbf{e}_k^\mu$. After ν steps of the damped Jacobi iteration, the error equals $\mathbf{u}_k^\nu - \mathbf{u}_k = \sum \beta_\mu \mathbf{e}_k^\mu$ with $\beta_\mu = \alpha_\mu \lambda_\mu^\nu$. The preceding consideration shows $\beta_\mu \approx \alpha_\mu$ for low frequencies but $|\beta_\mu| \ll |\alpha_\mu|$ for high frequencies. This fact can be expressed by saying that the error $\mathbf{u}_k^\nu - \mathbf{u}_k$ is *smoother* than $\mathbf{u}_k^0 - \mathbf{u}_k$. The first three graphs of Figure 2.2 illustrate the increasing smoothness of $\mathbf{u}_k^\nu - \mathbf{u}_k$ ($\nu = 0, 1, 2$). This is why we say that the iteration (2.1) serves as a *smoothing iteration.*

2.2 Structure of Two-Grid Iterations

The foregoing subsection showed that an appropriate smoothing iteration is quite an efficient method for reducing the high-frequency components. Convergence is only lacking with respect to low frequencies (smooth components). Therefore, one should combine this iteration with a second one having complementary properties. In particular, the second iteration should reduce the smooth error part very well. Such a complementary iteration can be constructed by means of the coarse grid with step size $h_{\ell-1} = 2h_\ell$ (see Figure 1.1).

Let $\mathbf{u}_\ell^{\text{old}}$ be some given approximation to $\mathbf{u}_\ell = \mathbf{L}_{\ell-1}^{-1}\mathbf{f}_\ell$ which serves as starting value. Few steps of the smoothing iteration $\mathcal{S}_k$ (cf. (1.13)) will result in an intermediate value $\overline{u}_\ell$. From the previous subsection

we know that the error $\mathbf{v}_\ell = \overline{\mathbf{u}}_\ell - \mathbf{u}_\ell$ is smooth (more precisely, smoother than $\mathbf{u}_\ell^{\text{old}} - \mathbf{u}_\ell$). $\mathbf{v}_\ell$ can also be regarded as the exact correction since $\mathbf{u}_\ell = \overline{\mathbf{u}}_\ell - \mathbf{v}_\ell$ represents the exact solution. Inserting $\overline{\mathbf{u}}_\ell$ into the equation $\mathbf{L}_\ell\,\mathbf{u}_\ell - \mathbf{f}_\ell = 0$, we obtain the *defect*

$$\mathbf{d}_\ell = \mathbf{L}_\ell\,\overline{\mathbf{u}}_\ell - \mathbf{f}_\ell \tag{2.2}$$

of $\overline{\mathbf{u}}_\ell$, which vanishes if and only if $\overline{\mathbf{u}}_\ell$ is the exact solution $\mathbf{u}_\ell$. Because of $\mathbf{L}_\ell\,\mathbf{v}_\ell = \mathbf{L}_\ell\,\overline{\mathbf{u}}_\ell - \mathbf{L}_\ell\,\mathbf{u}_\ell = \mathbf{L}_\ell\,\overline{\mathbf{u}}_\ell - \mathbf{f}_\ell = \mathbf{d}_\ell$, the exact correction $\mathbf{v}_\ell$ is the solution of

$$\mathbf{L}_\ell\,\mathbf{v}_\ell = \mathbf{d}_\ell. \tag{2.3}$$

Equation (2.3) is of the same form as the original equation $\mathbf{L}_\ell\,\mathbf{u}_\ell = \mathbf{f}_\ell$. Solving $\mathbf{L}_\ell\,\mathbf{v}_\ell = \mathbf{d}_\ell$ *exactly* is as difficult as solving $\mathbf{L}_\ell\,\mathbf{u}_\ell = \mathbf{f}_\ell$. Nonetheless, $\mathbf{v}_\ell$ can be approximated better than $\mathbf{u}_\ell$, since $\mathbf{v}_\ell$ is smooth and smooth functions can be represented well by means of coarser grids.

To approximate the problem $\mathbf{L}_\ell\,\mathbf{v}_\ell = \mathbf{d}_\ell$ by a *coarse-grid equation*

$$\mathbf{L}_{\ell-1}\,\mathbf{v}_{\ell-1} = \mathbf{d}_{\ell-1}, \tag{2.4}$$

we have to choose $\mathbf{d}_{\ell-1}$ reasonably. Note that the matrix $\mathbf{L}_k$ is already defined by (1.9) for all levels $k \geq 0$, especially for $k = \ell - 1$. The right-hand side $\mathbf{d}_{\ell-1}$ should depend linearly on the original defect d_ℓ. This is where the *restriction* r from (1.6) is needed,

$$\mathbf{d}_{\ell-1} := r\,\mathbf{d}_\ell. \tag{2.5}$$

Having defined $\mathbf{d}_{\ell-1}$, we obtain $\mathbf{v}_{\ell-1} = \mathbf{L}_{\ell-1}\,\mathbf{d}_{\ell-1}$ as the exact solution of (2.4). We expect $\mathbf{v}_{\ell-1}$ to be an approximation of the exact correction $\mathbf{v}_\ell$. However, $\mathbf{v}_{\ell-1}$ is only defined on the coarse grid $\Omega_{\ell-1}$. We have to interpolate this coarse-grid function by

$$\overline{\mathbf{v}}_\ell = p\,\mathbf{v}_{\ell-1}, \tag{2.6}$$

where the *prolongation* p is announced in (1.5) (see Figure 1.2).

Since $\mathbf{u}_\ell = \overline{\mathbf{u}}_\ell - \mathbf{v}_\ell$ is the exact solution and $\overline{v}_\ell = p\,\mathbf{v}_{\ell-1}$ is supposed to approximate v_ℓ, one tries to improve the value $\overline{\mathbf{u}}_\ell$ by

$$u_\ell^{\text{new}} := \overline{\mathbf{u}}_\ell - \overline{\mathbf{v}}_\ell. \tag{2.7}$$

The step from $\mathbf{u}_\ell$ to $\mathbf{u}_\ell^{\text{new}}$ by (2.2)-(2.7) is called the *coarse-grid correction*. Combining the separate parts (2.2)-(2.7), we obtain the compact formula

$$\overline{\mathbf{u}}_\ell \longmapsto \overline{\mathbf{u}}_\ell - p\,\mathbf{L}_{\ell-1}^{-1}\,r\,(\mathbf{L}_\ell\,\overline{\mathbf{u}}_\ell - \mathbf{f}_\ell) \tag{2.8}$$

for the coarse-grid correction.

Figure 2.2 shows the errors $\mathbf{u}_\ell^i - \mathbf{u}_\ell$ after $i = 0, 1, 2$ damped Jacobi iterations. The coarse-grid correction applied to $\overline{\mathbf{u}}_\ell = \mathbf{u}_\ell^2$ yields $\mathbf{u}_\ell^3$. The graph of the error $\mathbf{u}_\ell^3 - \mathbf{u}_\ell$ in Figure 2.2 proves the success of the coarse-grid correction. Although the coarse-grid correction seems to be efficient, it cannot be used as an iteration by itself, since it does not converge. It is the combination of smoothing iteration and coarse-grid correction that is rapidly convergent, whereas both components by themselves converge slowly or not at all. The combination is called the *two-grid iteration* since two levels ℓ and $\ell - 1$ are involved.

We summarise the two-grid iteration below. In Step (a) we use the notation $\mathcal{S}_\ell^\nu(\mathbf{u}_\ell^j, \mathbf{f}_\ell)$ for the ν-fold application of the smoothing iteration $\mathcal{S}_\ell(\mathbf{u}_\ell^j, \mathbf{f}_\ell)$.

Two-grid iteration for solving $\mathbf{L}_\ell\,\mathbf{u}_\ell = \mathbf{f}_\ell$

Start: $\mathbf{u}_\ell^j$ given iterate		
Smoothing step:		
$\overline{\mathbf{u}}_\ell = \mathcal{S}_\ell^\nu(\mathbf{u}_\ell^j, \mathbf{f}_\ell)$	(ν smoothing steps)	(a)
Coarse-grid correction:		
$\mathbf{d}_\ell := \mathbf{L}_\ell\,\overline{\mathbf{u}}_\ell - \mathbf{f}_\ell$	(calculation of the defect)	(b_1)
$\mathbf{d}_{\ell-1} := r\,\mathbf{d}_\ell$	(restriction of the defect)	(b_2)
$\mathbf{v}_{\ell-1} := \mathbf{L}_{\ell-1}^{-1}\,\mathbf{d}_{\ell-1}$	(solution of coarse-grid eq.)	(b_3)
$\mathbf{u}_\ell^{j+1} := \overline{\mathbf{u}}_\ell - p\mathbf{v}_{\ell-1}$	(correction of $\overline{\mathbf{u}}_\ell$)	(b_4)

(2.9)

The number ν of smoothing iterations can be chosen independently of the grid size h_ℓ. Its influence on convergence will be described in Remark 2.2.

Below, the two-grid iteration (2.9) is written in a quasi-ALGOL style.

```
function TGM(k, u, f);                                   variant TGM^(ν)
begin if k = 0 then TGM := L_0^{-1} * f else
      begin u := S_k^ν(u, f);                                    (a')
            d := r * (L_k * u − f);                              (b'_{1,2})
            v := L_{k−1}^{-1} * d;                               (b'_3)
            TGM := u − p * v                                     (b'_4)
end   end;
```
(2.9')

The function TGM performs *one* iteration step at level k (first parameter). The third parameter $\mathbf{f}$ is the right-hand side $\mathbf{f}_k$ of the equation to be solved. The input value of the second parameter $\mathbf{u}$ is the given jth iterate $\mathbf{u}_k^j$ that is mapped into the output value $TGM = \mathbf{u}_k^{j+1}$. The second line "**if** $\ell = 0$ **then** ..." is added to have a well-defined algorithm for all levels $k \geq 0$. Note that $TGM = TGM^{(\nu)}$ depends on the choice of ν.

Iteration (2.9) can be regarded as the prototype of a two-grid method. However, there are many variants. Instead of applying first smoothing and thereafter the coarse-grid correction, we can interchange these parts. More generally, ν_1 smoothing iterations can be performed before and ν_2 iterations after the coarse-grid correction. The resulting algorithm is

```
function TGM(k, u, f);                                   variant TGM^(ν1,ν2)
begin if k = 0 then TGM := L_0^{-1} * f else
      begin u := S_k^{ν1}(u, f);                  (pre-smoothing)          (a)
            u := u − p * L_{ℓ−1}^{-1} * r * (L_ℓ * u − f);  (coarse-grid correction)  (b)
            TGM := S_ℓ^{ν2}(u, f)                 (post-smoothing)         (c)
end   end;
```
(2.10)

2.3 Two-Grid Convergence in the 1D Case

In the case of the one-dimensional model problem from §1.2.5, a discrete Fourier analysis can be applied. The explicit calculation can be found in [12] and [14, Sect. 10.3]. Here we give only the results.

The *iteration matrix* $\mathbf{M}_\ell = \mathbf{M}_\ell^{\mathrm{TGM}}(\nu)$ of the two-grid method (2.9) is defined by $\mathbf{u}_\ell^{j+1} = \mathbf{M}_\ell \mathbf{u}_\ell^j + \mathbf{N}_\ell \mathbf{f}_\ell$ and equals

$$\mathbf{M}_\ell^{\mathrm{TGM}}(\nu) = \left(\mathbf{I} - p\mathbf{L}_{\ell-1} r \mathbf{L}_\ell\right) \mathbf{S}_\ell^\nu,$$

where $\mathbf{S}_\ell = \mathbf{I} - \frac{1}{4}h_\ell^2 \mathbf{L}_\ell$ is the iteration matrix of the smoothing procedure from §2.1. The parameter ν of $\mathbf{M}_\ell^{\mathrm{TGM}}(\nu)$ shows the explicit dependence on the number of smoothing steps.

In the following we discuss the spectral radius and the spectral norm of $\mathbf{M}_\ell = \mathbf{M}_\ell^{\mathrm{TGM}}(\nu)$.

The *spectral radius* $\rho(\mathbf{M}_\ell)$ is defined as the maximal value $|\lambda|$ for all eigenvalues λ of $\mathbf{M}_\ell$. A linear iteration converges (i.e. $\mathbf{u}_\ell^j \to \mathbf{u}_\ell$ for all starting values $\mathbf{u}_\ell^0$) if and only if $\rho(\mathbf{M}_\ell) < 1$. The value of $\rho(\mathbf{M}_\ell)$ describes the *asymptotic* rate of convergence.

If we are interested in the uniform rate, i.e., $\|\mathbf{u}_\ell^{j+1} - \mathbf{u}_\ell\| \leq \zeta \|\mathbf{u}_\ell^j - \mathbf{u}_\ell\|$, the best *contraction number* ζ is given by the spectral norm $\zeta = \|\mathbf{M}_\ell\|$. While $\rho(\mathbf{M}_\ell)$ is the appropriate description for a larger number of iteration steps, the contraction number $\zeta = \|\mathbf{M}_\ell\|$ is needed if we perform only very few steps of an iteration.

Theorem 2.1 *Let the two-grid iteration of level ℓ be defined by (2.9) with $\nu \geq 1$. Then the spectral radius of the iteration matrix $\mathbf{M}_\ell = \mathbf{M}_\ell^{\mathrm{TGM}}(\nu)$ is bounded (uniformly for all $\ell \geq 0$) by*

$$\rho\left(\mathbf{M}_\ell^{\mathrm{TGM}}(\nu)\right) \leq \rho_\nu := \max\left\{\xi(1-\xi)^\nu + (1-\xi)\xi^\nu : \ 0 \leq \xi \leq \tfrac{1}{2}\right\} < 1; \tag{2.11}$$

hence, convergence follows. The spectral norm is bounded (uniformly in ℓ) by

$$\left\|\mathbf{M}_\ell^{\mathrm{TGM}}(\nu)\right\| \leq \zeta_\nu := \max\left\{\sqrt{\xi^2(1-\xi)^{2\nu} + (1-\xi)^2\xi^{2\nu}} : 0 \leq \xi \leq \tfrac{1}{2}\right\} < 1. \tag{2.12}$$

Special values and the asymptotic behaviour are given in

ν	1	2	3	4	5	10	$\nu \to \infty$
ρ_ν	1/2	1/4	1/8	0.0832	0.0671	0.0350	$\rho_\nu \approx c_1/(\nu+1)$ with $c_1 = 1/e$
ζ_ν	1/2	1/4	0.150	0.1159	0.0947	0.0496	$\zeta_\nu \approx c_2/(\nu+1)$ with $c_2 = \sqrt{2}/e$

The bounds ρ_ν and ζ_ν are uniform with respect to ℓ; hence, uniform with respect to the grid size h_ℓ. The spectral radii of classical iterative methods depend on h_ℓ and tend to 1 for $h_\ell \to 0$. Examples are Jacobi's iteration and the Gauss-Seidel iteration with $\rho(\mathbf{M}_\ell) = 1 - \mathcal{O}(h_\ell^2)$ or successive overrelaxation (SOR) with $\rho(\mathbf{M}_\ell) = 1 - \mathcal{O}(h_\ell)$ at best.

In sharp contrast to this behaviour, the two-grid iteration as well as the multi-grid iteration defined below have spectral radii and contraction numbers uniformly bounded by some number smaller than 1. As a consequence, an accuracy ε can be obtained by $j = \mathcal{O}(\log 1/\varepsilon)$ iterations, where j is independent of h_ℓ.

Remark 2.2 *(a) The rate ρ_ν and the contraction number ζ_ν improve with increasing ν. However, they do not decrease exponentially ($\approx C\eta^\nu$) but like C/ν. This behaviour can be shown to hold for much more general boundary value problems. For less regular problems C/ν has to be changed to C/ν^α, $0 < \alpha < 1$.*

(b) A consequence of $\zeta_\nu \approx C/\nu$ is the recommendation not to choose ν too large. Doubling of ν decreases ζ_ν to $\zeta_{2\nu} \approx \zeta_\nu/2$. But also the computational work is almost doubled (at least in the multi-grid version). Thus, the choice of ν is better than 2ν if $\zeta_\nu^2 \lessapprox \zeta_\nu/2$, i.e., if $\zeta_\nu \lessapprox 1/2$.

3 Multi-Grid Method

3.1 Definition of the Multi-Grid Iteration

In the previous section, the two-grid method proved to be a very fast iteration. However, in the more interesting case of multi-dimensional boundary value problems, the two-grid iteration is impractical because of the exact solution of $\mathbf{L}_{\ell-1}\mathbf{v}_{\ell-1} = \mathbf{d}_{\ell-1}$ required in (2.9b$_3$). The solution $\mathbf{v}_{\ell-1}$ is needed to compute the correction $\bar{\mathbf{v}}_\ell = p\,\mathbf{v}_{\ell-1}$. Since $\bar{\mathbf{v}}_\ell$ is only an approximation to $\mathbf{v}_\ell = \mathbf{L}_\ell^{-1}\mathbf{d}_\ell$, there is no need for the exact computation of $\bar{\mathbf{v}}_\ell$ and then of $\mathbf{v}_{\ell-1}$. It suffices to calculate an approximate solution $\tilde{\mathbf{v}}_{\ell-1}$ of

$$\mathbf{L}_{\ell-1}\,\mathbf{v}_{\ell-1} = \mathbf{d}_{\ell-1}. \tag{3.1}$$

For example, the approximation $\tilde{\mathbf{v}}_{\ell-1}$ can be obtained by some iterative process:

$$\mathbf{v}_{\ell-1}^0 := 0 \mapsto \mathbf{v}_{\ell-1}^1 \mapsto \ldots \mapsto \mathbf{v}_{\ell-1}^\gamma =: \tilde{\mathbf{v}}_{\ell-1} \quad (\gamma : \text{ number of iterations}).$$

Note that the coarse-grid system (3.1) is of the same form as the original problem $\mathbf{L}_\ell\,\mathbf{u}_\ell = \mathbf{f}_\ell$ (only ℓ is replaced by $\ell-1$). Hence, the two-grid iteration of §2 (with levels $\{\ell-1, \ell-2\}$ instead of $\{\ell,\ \ell-1\}$ can be used as an iterative solver for system (3.1) provided that $\ell - 1 > 0$. This combination yields a three-grid method involving the levels ℓ, $\ell-1$, $\ell-2$. The exact solution of (3.1) is replaced by γ steps of the two-grid iteration at the levels $\ell-1$, $\ell-2$ involving the solution of new auxiliary equations $\mathbf{L}_{\ell-2}\,\mathbf{v}_{\ell-2} = \mathbf{d}_{\ell-2}$ for γ different right-hand sides $\mathbf{d}_{\ell-2}$.

If $\ell - 2 > 0$, the exact solution of $\mathbf{L}_{\ell-2}\,\mathbf{v}_{\ell-2} = \mathbf{d}_{\ell-2}$ can again be approximated by a two-grid iteration at the levels $\ell-2$ and $\ell-3$. The resulting algorithm would be a four-grid method. This process can be repeated until all $\ell+1$ levels $\ell,\ \ell-1, \ell-2, \ldots,\ 1,\ 0$ are involved. Equations at level 0 must be solved exactly or approximated by some other iteration. Level 0 corresponds to the coarsest grid and therefore to the smallest system of equations. In our model example (1.9), there is only one ($n_0 = 2^{0+1} - 1 = 1$) unknown $\mathbf{u}_0(1/2)$ at level 0. The resulting $(\ell+1)$-level iteration, which is now called a *multi-grid iteration*, can easily be described by the following recursive program.

Multi-grid iteration $MGM^{(\nu)}$ for solving $\mathbf{L}_k\,\mathbf{u}_k = \mathbf{f}_k$

```
function MGM(k, u, f);
begin if k = 0 then MGM := L_0^{-1} * f else                    (a)
      begin u := S_k^ν(u, f);                                   (b)
            d := r * (L_k * u - f);                             (c1)
            v := 0;                                             (c2)
            for j := 1 to γ do v := MGM(k - 1, v, d);           (c3)
            MGM := u - p * v                                    (c4)
end   end;
```

(3.2)

The meaning of the parameters k, $\mathbf{u}, \mathbf{f}$ is the same as for the procedure TGM from (2.9'). One call of MGM performs one iteration of the multi-grid method (at level k). Comparison of procedure TGM with MGM shows that the only modification is the replacement of (2.9b$'_3$) by (3.2c$_2$-c$_3$). Instead of solving the coarse-grid equation (3.1) exactly, one applies γ iterations of the multi-grid iteration at level $\ell-1$ with right-hand side $\mathbf{d}$. One call of MGM at level ℓ gives rise to γ calls of MGM at level $\ell-1$ involving a further γ^2

calls at level $\ell - 2$, etc. The recursive process ends at level 0, where the auxiliary problems $\mathbf{L}_0\mathbf{v}_0 = \mathbf{d}_0$ are solved exactly (cf. (3.2a)).

We shall see that $\gamma = 1$ or $\gamma = 2$ are appropriate values of the iteration number γ. In the special case of $\gamma = 1$ (the so-called *V-cycle*), the multi-grid algorithm MGM can be written equivalently as

$$
\begin{array}{l}
\textbf{function } VCycle(\ell, \mathbf{u}_\ell, f_\ell); \\
\textbf{begin for } k := \ell \textbf{ step } -1 \textbf{ until } 1 \textbf{ do} \\
\qquad \textbf{begin } \mathbf{u}_k := \mathcal{S}_k^\nu(\mathbf{u}_k, \mathbf{f}_k);\ \mathbf{f}_{k-1} := r * (\mathbf{L}_k * \mathbf{u}_k - \mathbf{f}_k);\ \mathbf{u}_{k-1} := \mathbf{0} \textbf{ end;} \\
\qquad \mathbf{u}_0 := \mathbf{L}_0^{-1} * \mathbf{f}_0; \\
\qquad \textbf{for } k := 1 \textbf{ step } 1 \textbf{ until } \ell \textbf{ do } \mathbf{u}_k := \mathbf{u}_k - p * \mathbf{u}_{k-1}; \\
\qquad VCycle := \mathbf{u}_\ell \\
\textbf{end};
\end{array}
\tag{3.3}
$$

The sequence of operations during one step of the multi-grid iteration (3.3) (i.e., for $\gamma = 1$) is depicted in the left parts of Figures 3.1 and 3.2. The stave symbolises the scale of levels. In the case when $\gamma = 2$, the multi-grid iteration cannot be represented in such a simple form as (3.3). The right part of Figure 3.2 show the details of one multi-grid step at level 4 involving 2 (4, 8) iteration steps at level 3 (2,1, resp.). Due to the form of Figure 3.1, the iteration with $\gamma = 1$ is also called a *V-cycle*, while the name *W-cycle* is used for $\gamma = 2$.

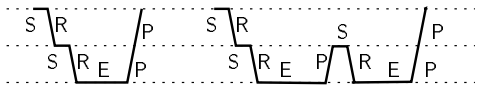

Figure 3.1: One V-cycle ($\gamma = 1$) and one W-cycle for $\ell = 2$. S: smoothing step; R: restriction of the defect; E: exact solving at level 0; P: correction $u_k \mapsto u_k - pu_{k-1}$

Figure 3.2: One V-cycle ($\gamma = 1$; left) and one W-cycle ($\gamma = 2$; right) for $\ell = 4$

It turns out that the properties of the multi-grid iteration are almost the same as those of the two-grid iteration. The contraction numbers of both methods are closely related. The bounds of the multi-grid contraction numbers (and of the convergence rates) are not only independent of the step sizes $h_\ell,\ h_{\ell-1}, \ldots, h_0$ but also independent of the total number of levels involved in the iteration.

Remark 3.1 *Analogously to the two-grid method $TGM^{(\nu_1,\nu_2)}$ from (2.10) involving pre- and post-smoothing, a multi-grid method $MGM^{(\nu_1,\nu_2)}$ can be constructed.*

3.2 Example

A simple test example is the Poisson equation $-\Delta u := -u_{xx} - u_{yy} = f$ in $\Omega = (0,1) \times (0,1)$ discretised by the five-point difference scheme

$$-\Delta_h u := h^{-2}\left[4u(x,y) - u(x-h,y) - u(x+h,y) - u(x,y-h) - u(x,y+h)\right]$$

with Dirichlet values $u(x,y) = x^2 + y^2$. The coarsest possible mesh size is $h_0 := 1/2$, the further mesh sizes are $h_k = 2^{-1-k}$. The table shows the iteration errors $e_m := \|\mathbf{u}_\ell^m - \mathbf{u}_\ell^{\text{exact}}\|_\infty$ after m steps of the multi-grid iteration ($\mathbf{u}_\ell^0 = \mathbf{0}$, W-cycle, 2 Gauss-Seidel pre-smoothing steps) at level $\ell = 7$ (corresponding to $h_\ell = 1/256$ and 65025 degrees of freedom).

m	e_m	ratio	m	e_m	ratio	m	e_m	ratio
0	1.984_{10}-0	—						
1	3.038_{10}-1	0.1531	4	5.219_{10}-5	0.0579	7	1.166_{10}-08	0.0619
2	1.605_{10}-2	0.0528	5	3.102_{10}-06	0.0594	8	7.713_{10}-10	0.0662
3	9.017_{10}-4	0.0562	6	1.884_{10}-07	0.0607	9	5.218_{10}-11	0.0677

The column 'ratio' shows the error reduction corresponding to a convergence rate of 0.067. Similar rates are observed for other step sizes. The starting value $\mathbf{u}_\ell^0 = \mathbf{0}$ is chosen here only for demonstration. Instead it is recommended to use the nested iteration from §6.

The results from above correspond to the W-cycle. The V-cycle shows a somewhat slower convergence rate of about 1.8_{10}-1 (the error for $m = 9$ is $e_9 = 4.98_{10}$-07). On the other hand, the V-cycle requires less computational work (see Remark 3.2). The choice $\gamma = 3$ is quite impractical. Although the computational work increases a lot, $\gamma = 3$ yields almost the same results as the W-cycle, e.g., $e_9 = 4.793_{10}$-11.

3.3 Computational Work

Let $n_k = \dim \mathcal{U}_k$ be the number of degrees of freedom at level k. Assuming $h_{k-1} \approx 2h_k$ and $\Omega \subset \mathbb{R}^d$, we have $n_k \approx 2^d n_{k-1}$. Due to recursivity, one call of $MGM\,(\ell, \cdot, \cdot)$ involves $\gamma^{\ell-k}$ calls of $MGM\,(k, \cdot, \cdot)$. As long as $\gamma 2^{-d} < 1$, the number of arithmetical operations spent at level k decreases exponentially with decreasing level. Indeed, even $\gamma 2^{-d} \le 1/2$ holds because of $d \ge 2$ ($d = 1$ is uninteresting) and $\gamma \le 2$ (V/W-cycle). Therefore, linear complexity holds.

Remark 3.2 *Let $n_k \approx 2^d n_{k-1}$. The cost of one call of $MGM\,(\ell, \cdot, \cdot)$ is bounded by $C_\ell n_\ell$ arithmetical operations, where for $d = 2$*

$$C_\ell \le \begin{cases} \frac{4}{3}\,[\nu C_S + C_D + C_C] + O(4^{-\ell}) & \text{for } \gamma = 1 \text{ (V-cycle)}, \\ 2\,[\nu C_S + C_D + C_C] + O(2^{-\ell}) & \text{for } \gamma = 2 \text{ (W-cycle)}, \end{cases} \tag{3.4}$$

where $C_S\, n_\ell$, $C_D\, n_\ell$, $C_C\, n_\ell$ and $C_0\, n_0$ are the costs for $u_\ell \mapsto \mathcal{S}_\ell(u_\ell, f_\ell)$, $(u_\ell, f_\ell) \mapsto r(L_\ell\, u_\ell - f_\ell)$, $(u_\ell, v_{\ell-1}) \mapsto u_\ell - p v_{\ell-1}$ and $f_0 \mapsto L_0^{-1} f_0$. For $d = 3$, the factors $\frac{4}{3}$ and 2 in (3.4) reduce to $\frac{4}{3}$ and $\frac{8}{7}$, respectively.

3.4 Algebraic Multi-Grid Methods

So far we have assumed that there is hierarchy of discretisations (cf. §1.1.6). If only the final discretisation is given or by some other reason the user is not providing coarser systems, this construction can be performed as part of the "algebraic multi-grid method" (AMG). There are different approaches how to select a coarse grid and how to define the prolongations and restrictions. The name "algebraic multi-grid method" indicates that only the algebraic system of equations is used as input (not necessarily geometric data and descriptions about the nature of the pde). As a consequence the resulting algorithm is more 'black-box'-like.

We refer the interested reader to [23], [3], [22], [11] and the literature given therein.

4 Application to Finite Element Equations

While the introductory example corresponds to a difference scheme, we now discuss the multi-grid method in the case of a finite element discretisation. The multi-grid ingredients from §1.2 are defined in a canonical way, provided that the finite element spaces are nested as explained below. Otherwise, hints are given in §4.5.

We assume that the boundary value problem is formulated in the weak variational form: Find $u \in \mathcal{H}$ such that

$$a(u, v) = f(v) \qquad \text{for all } v \in \mathcal{H}, \tag{4.1}$$

where the 'energy space' $\mathcal{H}$ may include the required homogeneous Dirichlet conditions (e.g., $\mathcal{H} = H_0^1(\Omega)$). In the case of scalar functions u the bilinear form $a(u, v)$ may be

$$a(u, v) := \int_\Omega \left(\sum_{\alpha,\beta=1}^{d} c_{\alpha,\beta} \frac{\partial u}{\partial x_\alpha} \frac{\partial v}{\partial x_\beta} + \sum_{\alpha=1}^{d} c_{\alpha,0} \frac{\partial u}{\partial x_\alpha} v + \sum_{\beta=1}^{d} c_{0,\beta} u \frac{\partial v}{\partial x_\beta} + c_{0,0} uv \right) dx.$$

The functional f in (4.1) is $f(v) = \int_\Omega f v dx$. In the case of inhomogeneous Neumann conditions, it contains a further term $\int_\Gamma \varphi v d\Gamma$ ($\Gamma = \partial\Omega$ denotes the boundary of Ω).

4.1 Finite Element Problem

The FE space $\mathcal{H}^{\mathrm{FEM}}$ is a finite-dimensional subspace of $\mathcal{H}$ and the FE problems reads:

$$\text{Find } u^{\mathrm{FEM}} \in \mathcal{H}^{\mathrm{FEM}} \text{ with} \qquad a(u^{\mathrm{FEM}}, v) = f(v) \quad \text{for all } v \in \mathcal{H}^{\mathrm{FEM}}. \tag{4.2}$$

The usual approach in the 2D case is to define $\mathcal{H}^{\mathrm{FEM}}$ by means of piecewise linear (quadratic, ...) elements corresponding to a triangulation $\mathcal{T}$ which is a set of triangles such that $\bigcup_{\tau\in\mathcal{T}} \tau = \overline{\Omega}$. In 3D, the 'triangulation' $\mathcal{T}$ contains tetrahedra etc.

The functions $u \in \mathcal{H}^{\mathrm{FEM}}$ are represented by means of the *nodal basis*, i.e., there are a set $\mathcal{I}$ of indices α associated with 'nodal points' $\mathbf{x}_\alpha \in \mathbb{R}^d$ and a basis

$$\mathcal{B} = \{\phi_\alpha : \alpha \in \mathcal{I}\} \subset \mathcal{H}^{\mathrm{FEM}}$$

with the interpolation property $\phi_\alpha(\mathbf{x}_\beta) = \delta_{\alpha\beta}$ for all $\alpha, \beta \in \mathcal{I}$. Setting $u_\alpha := u(\mathbf{x}_\alpha)$, we obtain

$$u = \sum\nolimits_{\alpha\in\mathcal{I}} u_\alpha \phi_\alpha \qquad \text{for any } u \in \mathcal{H}^{\mathrm{FEM}}. \tag{4.3}$$

The coefficients u_α form the *coefficient vector* $\mathbf{u} = (u_\alpha)_{\alpha\in\mathcal{I}} \in \mathcal{U}$. Relation (4.3) gives rise to the interpolation $\mathbf{u} \mapsto u := P\mathbf{u} \in \mathcal{H}^{\mathrm{FEM}}$ by means of (4.3). P is a bijection from $\mathcal{U}$ onto $\mathcal{H}^{\mathrm{FEM}}$.

The stiffness matrix $\mathbf{L}$ is given by the entries $\mathbf{L}_{\alpha\beta} = a(\phi_\beta, \phi_\alpha)$, while the right-hand side vector is $\mathbf{f}$ with $f_\alpha = f(\phi_\alpha)$. Altogether, we obtain the finite element equation $\mathbf{Lu} = \mathbf{f}$.

4.2 Nested Finite Element Spaces

In order to get a family of finite element problems $\mathbf{L}_k\mathbf{u}_k = \mathbf{f}_k$, we may assume a sequence of *nested subspaces*, i.e., there are finite element spaces $\mathcal{H}_k$ $(0 \le k \le \ell)$ replacing $\mathcal{H}^{\mathrm{FEM}}$ in (4.2) such that

$$\mathcal{H}_0 \subset \mathcal{H}_1 \subset \ldots \subset \mathcal{H}_\ell \subset \mathcal{H}. \tag{4.4}$$

This setting corresponds to conforming finite element methods. The non-conforming case will be considered in §4.6. But even in the conforming case (i.e., $\mathcal{H}_\ell \subset \mathcal{H}$ for all levels ℓ), one may use subspaces with $\mathcal{H}_{\ell-1} \not\subset \mathcal{H}_\ell$ (see §4.5).

The easiest way to construct these nested subspaces is by repeated refinement of the triangulation. Let $\mathcal{T}_0$ be the coarsest triangulation. The refined triangulations $\mathcal{T}_k$ must satisfy

$$\text{each triangle } \tau \in \mathcal{T}_k \text{ is the union of triangles } \tau'_1, \tau'_2, \ldots \text{ from } \mathcal{T}_{k+1} \qquad (k = 0, \ldots, \ell-1).$$

Then, elements being piecewise linear (quadratic) on $\tau \in \mathcal{T}_k$ are piecewise linear (quadratic) on $\tau' \in \mathcal{T}_{k+1}$; hence, $u \in \mathcal{H}_k$ belongs also to $\mathcal{H}_{k+1}$, i.e., $\mathcal{H}_k \subset \mathcal{H}_{k+1}$ as required in (4.4).

We introduce the following notation: $\mathbf{u}_k, \mathbf{f}_k \in \mathcal{U}_k$ for vectors at level k, while $\mathbf{L}_k$ is the stiffness matrix at level k. The nodal basis vectors are $\phi_{k,\alpha}$ $(\alpha \in \mathcal{I}_k,\ k \in \{0, \ldots, \ell\})$, where $\mathcal{I}_k$ is the index set of level k. The interpolation (4.3) is now denoted by $P_k : \mathcal{U}_k \to \mathcal{H}_k$.

4.3 Canonical Prolongations and Restrictions for FE Equations

The multi-grid prolongation $p : \mathcal{U}_{k-1} \to \mathcal{U}_k$ (see (1.5)) is uniquely defined as follows. Let $\mathbf{u}_{k-1} \in \mathcal{U}_{k-1}$ a given coefficient vector and consider the corresponding finite element function $u^{k-1} = P_{k-1}\mathbf{u}_{k-1} \in \mathcal{H}_{k-1}$. Since $\mathcal{H}_{k-1} \subset \mathcal{H}_k$, u^{k-1} allows a basis representation $u^{k-1} = P_k\mathbf{u}_k$ (cf. (4.3)) by means of a unique coefficient vector $\mathbf{u}_k \in \mathcal{H}_k$. Define $p\mathbf{u}_{k-1}$ by $\mathbf{u}_k$. Hence, the formal definition of p is $P_k^{-1}P_{k-1}$:

$$\boxed{\begin{array}{ccc} \mathcal{U}_{k-1} & \xrightarrow[p]{} & \mathcal{U}_k \\ \downarrow P_{k-1} & & \downarrow P_k \\ \mathcal{H}_{k-1} & \xrightarrow{\text{inclusion}} & \mathcal{H}_k \end{array} \quad , \qquad p = P_k^{-1}P_{k-1}.} \tag{4.5}$$

The *canonical restriction* is $r = p^*$, where p is the canonical prolongation from above, i.e.,

$$\langle r\mathbf{f}_k, \mathbf{v}_{k-1}\rangle_{k-1} = \langle \mathbf{f}_k, p\mathbf{v}_{k-1}\rangle_k \qquad \text{for all } \mathbf{v}_{k-1} \in \mathcal{U}_{k-1}. \tag{4.6}$$

Here, $\langle\cdot,\cdot\rangle_k$ denotes the Euclidean scalar product of $\mathcal{U}_k$.

4.4 Coarse-Grid Matrix and Coarse-Grid Correction

With a view to the next section we state the characteristic relation between p, r, and the stiffness matrices $\mathbf{L}_k$, $\mathbf{L}_{k-1}$.

Remark 4.1 *Let $\mathbf{L}_k$ and $\mathbf{L}_{k-1}$ be the finite element stiffness matrices of §4.1 and let p and r be canonical. Then $\mathbf{L}_{k-1}$ coincides with the Galerkin product*

$$\mathbf{L}_{k-1} = r\,\mathbf{L}_k\,p. \tag{4.7}$$

The coarse grid correction $\mathcal{C}_k(\mathbf{u}_k, \mathbf{f}_k) := \mathbf{u}_k - p\,\mathbf{L}_{k-1}^{-1}\, r\,(\mathbf{L}_k\,\mathbf{u}_k - \mathbf{f}_k)$ in (2.8) can be reformulated in terms of the finite element function $u^k = P_k\,\mathbf{u}_k \in \mathcal{H}_k$ as follows.

Proposition 4.2 *Assume $\mathcal{H}_{k-1} \subset \mathcal{H}_k$ and let p, r be the canonical mappings from §4.3. Let $a(\cdot,\cdot)$ be the bilinear form of the underlying variational formulation. Then the coarse grid correction $\mathbf{u}_k \mapsto \mathbf{u}_k - p\mathbf{v}_{k-1}$ from (3.2c$_4$) is equivalent to the mapping*

$$u^k \quad \mapsto \quad u^k - v^{k-1}, \tag{4.8a}$$

$$a\left(v^{k-1}, w^{k-1}\right) = a\left(u^k, w^{k-1}\right) - f(w^{k-1}) \qquad \text{for all } w^{k-1} \in \mathcal{H}_{k-1}. \tag{4.8b}$$

where $v^{k-1} \in \mathcal{H}_{k-1}$ is the finite element solution of (4.8b). The solvability of (4.8b) is equivalent to the regularity of $\mathbf{L}_{k-1}$.

The corresponding representation for nonconforming finite elements is given in (4.14).

4.5 Non-Nested Finite Element Spaces

So far, we have assumed the finite element spaces to be nested: $\mathcal{H}_0 \subset \mathcal{H}_1 \subset \ldots \subset \mathcal{H}_\ell$. If $\mathcal{H}_k$ are standard finite element spaces over a triangulation $\mathcal{T}_k$, the inclusion $\mathcal{H}_{k-1} \subset \mathcal{H}_k$ requires that all coarse triangles $t \in \mathcal{T}_{k-1}$ must be the union of some fine triangles $t_1, \ldots, t_k \in \mathcal{T}_k$. This condition may be violated close to a curved boundary or near a curved interface. It may even happen that $\mathcal{H}_{k-1}$ and $\mathcal{H}_k$ are completely unrelated except that $\dim \mathcal{H}_{k-1} < \dim \mathcal{H}_k$ expresses that $\mathcal{H}_{k-1}$ gives rise to a coarser discretisation. In these cases, the definition (4.5) of the canonical prolongation cannot be applied.

The cheapest remedy is the definition of p by means of the *finite element interpolation*. Let $\mathbf{u}_{k-1} \in \mathcal{U}_{k-1}$ be the coefficient vector corresponding to the nodal basis of $\mathcal{H}_{k-1}$, i.e.,$(P_{k-1}\,\mathbf{u}_{k-1})(\mathbf{x}_\alpha) = u_{k-1,\alpha}$ for $\mathbf{x}_\alpha \in \Omega_{k-1}$. Here, Ω_k is the set of nodal points of the triangulation $\mathcal{T}_k$. The desired vector $p\,\mathbf{u}_{k-1} \in \mathcal{U}_k$ has to represent the values $P_k\,p\,\mathbf{u}_{k-1}$ on Ω_k. Even when $P_k\,p\,\mathbf{u}_{k-1}$ cannot coincide with $P_{k-1}\mathbf{u}_{k-1}$ in $\mathcal{H}$, it can interpolate $P_{k-1}\mathbf{u}_{k-1}$:

$$(p\,\mathbf{u}_{k-1})(\mathbf{x}_\alpha) := (P_{k-1}\mathbf{u}_{k-1})(\mathbf{x}_\alpha) \qquad \text{for all } \mathbf{x}_\alpha \in \Omega_k. \tag{4.9}$$

The next remark ensures that definition (4.9) is a true generalisation of the canonical choice.

Remark 4.3 *If $\mathcal{H}_{k-1} \subset \mathcal{H}_k$, definition (4.9) yields the canonical prolongation p.*

As soon as p is defined, we can define the restriction by (4.6), that means by $r = p^*$.

4.6 Nonconforming Finite Element Discretisations

In the following, we give only the construction of the discretisation and multi-grid ingredients. For more details we refer to [4].

4.6.1 Nonconforming Discretisation

Let $\mathcal{H}_k \subset L^2(\Omega)$ be a family of (nonconforming) finite element spaces, i.e., we do not assume $\mathcal{H}_k \subset \mathcal{H}$. Moreover, the spaces $\mathcal{H}_k$ are not supposed to be nested $\mathcal{H}_{k-1} \not\subset \mathcal{H}_k$. Instead of the bilinear form $a(\cdot,\cdot)$ a mesh-dependent bilinear form $a_k(\cdot,\cdot)$ on $\mathcal{H}_k \times \mathcal{H}_k$ is used. For $f \in \mathcal{H}^0$, the variational problem is discretised by

$$u^k \in \mathcal{H}_k \quad \text{with} \quad a_k(u^k, v^k) = f(v^k) \qquad \text{for all } v^k \in \mathcal{H}_k. \tag{4.10}$$

Again the isomorphism between $\mathcal{U}_k$ and $\mathcal{H}_k$ is denoted by P_k (cf. §4.3).

4.6.2 Multi-Grid Prolongation

The *canonical* prolongation $p = P_k^{-1} \circ P_{k-1}$ given by (4.5) is based on the inclusion $\mathcal{H}_{k-1} \subset \mathcal{H}_k$. Since $\mathcal{H}_{k-1} \not\subset \mathcal{H}_k$, the prolongation p must be defined differently. The inclusion $\mathcal{H}_{k-1} \subset \mathcal{H}_k$ is to be replaced by a suitable (linear) mapping

$$\iota : \mathcal{H}_{k-1} \to \mathcal{H}_k. \tag{4.11}$$

Once ι is constructed, we are able to define the canonical prolongation and restriction by

$$p := P_k^{-1} \circ \iota \circ P_{k-1} \quad \text{and} \quad r := p^* = P_{k-1}^* \circ \iota^* \circ (P_k^*)^{-1}. \tag{4.12}$$

Although the algorithm needs only the above mapping $\iota : \mathcal{H}_{k-1} \to \mathcal{H}_k$, it is easier to define ι on a larger space $\Sigma \supset \mathcal{H}_{k-1} + \mathcal{H}_k$ such that ι restricted to $\mathcal{H}_k$ is the identity. Since we only require $\Sigma \subset L^2(\Omega)$, no global smoothness is necessary.

Next, we need an auxiliary space S connected with Σ and $\mathcal{H}_k$ via the mappings σ and π, as shown in the following *commutative* diagram:

$$\begin{array}{rcl} \Sigma & \overset{\sigma}{\longrightarrow} & S \\ \text{inclusion } \uparrow & \nwarrow & \downarrow \pi \\ \mathcal{H}_{k-1} & \overset{\iota}{\longrightarrow} & \mathcal{H}_k \\ P_{k-1} \uparrow & & \uparrow P_k \\ \mathcal{U}_{k-1} & \overset{p}{\longrightarrow} & \mathcal{U}_k \end{array} .$$

$\pi : S \to \mathcal{H}_k$ is required to be *injective.* The desired mapping ι (more precisely, its extension to Σ) is the product

$$\iota = \pi \circ \sigma : \ \Sigma \to \mathcal{H}_k. \tag{4.13}$$

For the simplest nonconforming finite elements, the *Crouzeix-Raviart element*, we specify the spaces and mappings from above in

Example 4.4 *Let $\mathcal{T}_{k-1}$ be the coarse triangulation of the domain Ω, while $\mathcal{T}_k$ is obtained by regular halving of all triangle sides. $\mathcal{H}_k$ is the space of all piecewise linear functions which are continuous at the midpoints of edges in $\mathcal{T}_k$. Define the nodal point set $\Omega_k = \{\mathbf{x}_\alpha : \alpha \in \mathcal{I}_k\}$ by all midpoints of edges in $\mathcal{T}_k$ (except boundary points in the case of Dirichlet conditions). For all $\mathbf{x}_\alpha \in \Omega_k$, basis functions $b^k_\alpha \in \mathcal{H}_k$ are defined by $b^k_\alpha(\mathbf{x}_\beta) = \delta_{\alpha\beta}$ $(\alpha, \beta \in \mathcal{I}_k)$. Then, $\mathcal{U}_k$ is the coefficient space which is mapped by $P_k : \mathbf{u}_k = (u_{k,\alpha})_{\alpha\in\mathcal{I}_k} \mapsto u^k = \sum_{\alpha\in\mathcal{I}_k} u_{k,\alpha} b^k_\alpha$ onto $\mathcal{H}_k$. Similarly, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$, and the isomorphism P_{k-1} are defined.*

An appropriate space Σ is the space of piecewise linear elements with respect to the fine triangulation $\mathcal{T}_k$ that may be discontinuous. Obviously, $\mathcal{H}_{k-1} + \mathcal{H}_k \subset \Sigma$.

We set $S := \mathcal{U}_k$, $\pi := P_k$, and define σ as follows: Every nodal point $\mathbf{x}_\alpha \in \Omega_k$ is the midpoint of the common side of adjacent triangles $t, t' \in \mathcal{T}_k$. We define the image σv by

$$(\sigma v)_\alpha := \tfrac{1}{2}\left[v|_t(\mathbf{x}_\alpha) + v|_{t'}(\mathbf{x}_\alpha)\right] \qquad \text{for all } \mathbf{x}_\alpha \in \Omega_k \text{ and } v \in \Sigma.$$

Here, the linear function $v|_t$ is understood to be extended to the closure of t.

The multi-grid prolongation is $p = \sigma \circ P_{k-1}$.

4.6.3 Coarse-Grid Correction

Given p from (4.12) and $r := p^*$, the coarse-grid correction takes the standard form (2.8). Its FE interpretation is as follows.

Let an FE approximation $\tilde{u}^k \in \mathcal{H}_k$ be given. Its defect $d^k \in \mathcal{H}_k$ is defined by

$$(d^k, w^k)_{L^2(\Omega)} = a_k(\tilde{u}^k, w^k) - f(w^k) \qquad \text{for all } w^k \in \mathcal{H}_k.$$

Using (4.10) and the error $e^k := \tilde{u}^k - u^k$, one obtains a characterisation of d^k by

$$(d^k, w^k)_{L^2(\Omega)} = a_k(e^k, w^k) \qquad \text{for all } w^k \in \mathcal{H}_k.$$

Then the correction $e^{k-1} \in \mathcal{H}_{k-1}$ is determined as the solution of the *FE coarse-grid equation*

$$a_{k-1}(e^{k-1}, w^{k-1}) = a_k(e^k, \iota w^{k-1}) \qquad \text{for all } w^{k-1} \in \mathcal{H}_{k-1}. \tag{4.14}$$

Here ι is the mapping specified in (4.11). It is required for converting the function w^{k-1} from $\mathcal{H}_{k-1}$ into a function in $\mathcal{H}_k$. The correction yields the new approximation $u^{k,\text{new}} := \tilde{u}^k - \iota e^{k-1}$.

5 Additive Variant

5.1 The Additive Multi-Grid Algorithm

If one denotes the coarse-grid correction (2.8) by $\mathcal{C}_k(\mathbf{u}_k, \mathbf{f}_k) := \mathbf{u}_k - p\,\mathbf{L}_{k-1}^{-1}\, r\,(\mathbf{L}_k\,\mathbf{u}_k - \mathbf{f}_k)$, the two-grid iteration is the *product* $\mathcal{C}_k \circ \mathcal{S}_k^\nu$, i.e., $TGM(k, \mathbf{u}_k, \mathbf{f}_k) = \mathcal{C}_k(\mathcal{S}_k^\nu(\mathbf{u}_k, \mathbf{f}_k), \mathbf{f}_k)$. Instead of the product one can form the sum of the correction $\delta_{k,k} := \mathbf{u}_k - \mathcal{S}_k^\nu(\mathbf{u}_k, \mathbf{f}_k)$ of the smoothing procedure and the coarse-grid correction $\delta_{k,k-1} := p\mathbf{L}_{k-1}^{-1} r\,(\mathbf{L}_k\,\mathbf{u}_k - \mathbf{f}_k)$. Damping the sum of these terms by a factor ϑ, one obtains the iteration $\mathbf{u}_k^j \mapsto \mathbf{u}_k^{j+1} := \mathbf{u}_k^j - \vartheta\,(\delta_{k,k} + \delta_{k,k-1})$.

In the multi-grid case, one can try to separate the computations at all levels $0, \ldots, \ell$. We give a description of this algorithm which looks rather similar to the multiplicative algorithm (3.2) with $\gamma = 1$. The following additive multi-grid iteration $AMGM^{(\nu)}$ for solving $L_k\, u_k = f_k$ uses a damping factor ϑ which is irrelevant if the iteration is embedded into a cg-like acceleration method.

$$
\begin{array}{lll}
\textbf{function } AMGM(k, \mathbf{u}, \mathbf{f}); & & \\
\textbf{begin if } k = 0 \textbf{ then } AMGM := \vartheta * \mathbf{L}_0^{-1} * \mathbf{f} \textbf{ else} & & (5.1a) \\
\quad \textbf{begin } \mathbf{d} := r\,(\mathbf{f} - \mathbf{L}_k * \mathbf{u})\,; & & (5.1b) \\
\quad\quad \mathbf{u} := \mathbf{u} + \vartheta * (\mathcal{S}_k^{\nu}(\mathbf{u}, \mathbf{f}) - \mathbf{u})\,; & & (5.1c) \\
\quad\quad \mathbf{v} := 0\,;\ \mathbf{v} := AMGM(k-1, \mathbf{v}, \mathbf{d})\,; & & (5.1d) \\
\quad\quad AMGM := \mathbf{u} + p\,\mathbf{v} & & (5.1e) \\
\textbf{end} \quad \textbf{end}; & &
\end{array}
$$

Since (5.1b) does not influence the smoothing part (5.1c), both parts can be performed in parallel. The results of both parts are joined in (5.1e).

For the standard (multiplicative) two-grid method we know from §2.3 that the convergence improves in a particular way when the number ν of smoothing steps increases. The same holds for general multi-grid methods. However, this behaviour is not true for the additive variant as shown in [2].

5.2 Interpretation as Subspace Iteration

Similar to algorithm (3.2), one can resolve the recursivity in (5.1e). For this purpose, we write explicitly $p = p_{k,k-1}$ and $r = r_{k-1,k}$ (as in (1.5), (1.6)) and compose these mappings to obtain

$$p_{\ell,k} := p_{\ell,\ell-1} \circ p_{\ell-1,\ell-2} \circ \ldots \circ p_{k+1,k}, \quad r_{k,\ell} := r_{k,k+1} \circ \ldots \circ r_{\ell-1,\ell} \qquad \text{for } k \le \ell,$$

where $p_{\ell,\ell} = r_{\ell,\ell} = I$ in the case of the empty product. From the given right-hand side $\mathbf{f}_\ell$ one defines $\mathbf{f}_k := r_{k,\ell}\mathbf{f}_\ell$ for $k = 0, \ldots, \ell$. Then the additive multi-grid iteration $AMGM^{(\nu)}(\mathbf{u}_\ell, \mathbf{f}_\ell)$ described in (5.1) is given by

$$AMGM^{(\nu)}(\mathbf{u}_\ell, \mathbf{f}_\ell) = \mathbf{u}_\ell + \vartheta \sum\nolimits_{k=1}^{\ell} p_{\ell,k}\, \delta\mathbf{v}_k, \tag{5.2}$$

where the corrections $\delta\mathbf{v}_k$ at level k are defined by

$$
\begin{array}{ll}
\delta\mathbf{v}_\ell := \mathcal{S}_\ell^{\nu}(\mathbf{u}_\ell, \mathbf{f}_\ell) - \mathbf{u}_\ell & \text{for } k = \ell, \\
\delta\mathbf{v}_k := \mathcal{S}_k^{\nu}(\mathbf{0}, \mathbf{f}_k) & \text{for } k = 1, \ldots, \ell - 1, \\
\delta\mathbf{v}_0 := \mathbf{L}_0^{-1} * \mathbf{f}_0 & \text{for } k = 0.
\end{array}
$$

The case $k = \ell$ can easily be seen from (5.1c). Since the lower levels use the starting value $\mathbf{v} = \mathbf{0}$, $\mathcal{S}_k^{\nu}(\mathbf{u}_k, \mathbf{f}_k) - \mathbf{u}_k$ simplifies to $\mathcal{S}_k^{\nu}(\mathbf{0}, \mathbf{f}_k)$.

The interest in subspace iterations has two different reasons. **(i)** The computation of the corrections $\delta\mathbf{v}_k$ can be performed *in parallel*. **(ii)** The interpretation as subspace iteration allows quite different convergence proofs (in particular for the V-cycle) than the standard multiplicative version. Although the resulting statements are weaker, they require also weaker assumptions (see [6]).

6 Nested Iteration

6.1 Algorithm

6.1.1 Starting and Terminating an Iteration

The natural approach is to start with a more or less accurate initial value u_ℓ^0 and to perform several iteration steps:

$$\tilde{\mathbf{u}}_\ell := \mathbf{u}_\ell^0;\ \textbf{for } j := 1 \textbf{ to } i \textbf{ do } \tilde{\mathbf{u}}_\ell := MGM(\ell, \tilde{\mathbf{u}}_\ell, \mathbf{f}_\ell); \tag{6.1}$$

The error of $\tilde{\mathbf{u}}_\ell$ satisfies

$$\|\tilde{\mathbf{u}}_\ell - \mathbf{u}_\ell\| \le \zeta^i \|\mathbf{u}_\ell^0 - \mathbf{u}_\ell\|, \tag{6.2}$$

where ζ is the contraction number of the iteration (cf. §2.3) and $\mathbf{u}_\ell$ the solution of $\mathbf{L}_\ell\, \mathbf{u}_\ell = \mathbf{f}_\ell$. In particular, the simplest choice $\mathbf{u}_\ell^0 = 0$ yields an estimate of the relative error

$$\|\tilde{\mathbf{u}}_\ell - \mathbf{u}_\ell\| \,/\, \|\mathbf{u}_\ell\| \le \zeta^i \qquad (\text{if } \mathbf{u}_\ell^0 = \mathbf{0}). \tag{6.3}$$

In order to obtain a fixed (relative) error ε, one needs $i \geq \log(\varepsilon)/\log(\zeta) = \mathcal{O}(|\log(\varepsilon)|)$ iterations, where we exploit the fact that the contraction number ζ of the multi-grid iteration is ℓ-independent. Usually, ε is not explicitly given and one has to judge a suitable value. Except special cases, it is useless to take ε smaller than the discretisation error $\varepsilon_{\text{disc}}$ (i.e., the difference between $\mathbf{u}_\ell$ and the continuous solution u). Often, the quantitative size of $\varepsilon_{\text{disc}}$ is not known a priori, but only its asymptotic behaviour $\mathcal{O}(h_\ell^{\varkappa})$ ($\varkappa$: consistency order). From $\varepsilon := \varepsilon_{\text{disc}} = \mathcal{O}(h_\ell^{\varkappa})$ one concludes that $i = \mathcal{O}(|\log(h_\ell)|)$ iterations are required to obtain an iterate u_ℓ with an error of the size of the discretisation error. The corresponding number of arithmetical operations is $\mathcal{O}(n_\ell|\log(h_\ell)|) = \mathcal{O}(h_\ell^{-d}|\log(h_\ell)|)$.

6.1.2 Basic Algorithm

The nested iteration described below (also called 'full multi-grid method') has several advantages:

- Although no a priori knowledge of the discretisation error is needed, the nested iteration produces approximations $\tilde{\mathbf{u}}_\ell$ with error $\mathcal{O}(\varepsilon_{\text{disc}})$.
- The nested iteration is cheaper than the simple approach (6.1). An approximation with error $\mathcal{O}(\varepsilon_{\text{disc}})$ is calculated by $\mathcal{O}(n_\ell)$ operations.
- Besides $\tilde{\mathbf{u}}_\ell$, the solutions $\tilde{\mathbf{u}}_{\ell-1},\ \tilde{\mathbf{u}}_{\ell-2}, \ldots$ corresponding to the coarser grids are also approximated and are at one's disposal.

In principle, the nested iteration can be combined with any iterative process. The idea is to provide a good *starting guess* $\mathbf{u}_\ell^0$ by means of iterating on a coarser grid.

In this section, the index ℓ characterises the level of the finest grid, i.e., ℓ is the maximal level number, while the index $k \in \{0, 1, \ldots, \ell\}$ is used for the intermediate levels. We assume that the discrete equations $\mathbf{L}_k\mathbf{u}_k = \mathbf{f}_k$ are given for all levels.

A program-like formulation of the nested iteration reads as follows:

Nested Iteration

$\tilde{\mathbf{u}}_0 := \mathbf{u}_0 = \mathbf{L}_0^{-1}\mathbf{f}_0$; (6.4a)
for $k := 1$ **to** ℓ **do**
begin $\tilde{\mathbf{u}}_k := \tilde{p}\,\tilde{\mathbf{u}}_{k-1}$; (6.4b)
for $j := 1$ **to** i **do** $\tilde{\mathbf{u}}_k := MGM(k, \tilde{\mathbf{u}}_k, \mathbf{f}_k)$ (6.4c)
end;

6.1.3 Implementational Details

Starting Value at Level 0 The exact solution of $\mathbf{L}_0\,\mathbf{u}_0 = \mathbf{f}_0$ is not necessary. One may replace (6.4a) by $\tilde{\mathbf{u}}_0 \approx \mathbf{L}_0^{-1}\mathbf{f}_0$, provided the $\|\tilde{\mathbf{u}}_0 - \mathbf{u}_0\|$ is small enough.

Prolongation $\tilde{p}$ The starting value in (6.4b) is obtained from the coarse-grid approximation $\tilde{\mathbf{u}}_{\ell-1}$ by means of some interpolation $\tilde{p}$. From the programming point of view, the simplest choice is $\tilde{p} = p$ from (1.5). However, interpolations $\tilde{p}$ of higher order than p may be taken into considerations, too (see Remark 6.1).

If an asymptotic expansion is valid, Richardson's extrapolation can be applied to compute an fairly accurate value $\tilde{\mathbf{u}}_k$ from $\tilde{\mathbf{u}}_{k-1}$ and $\tilde{\mathbf{u}}_{k-2}$ (details in §5.4 of [12]).

Iterations per Level At each level, i iterations are performed. An appropriate choice of i is discussed in §6.2. The same value i can be chosen for all levels, since the contraction numbers of the multi-grid iteration are bounded independently of the level $0 \leq k \leq \ell$. Since most of the computational work is spent at level ℓ, it can be advantageous to choose $i_\ell \leq i_{\ell-1} = i_{\ell-2} = \ldots = i_1$ (cf. Remark 6.4).

Adaptive Choice of the Finest Level The nested iteration can be interpreted in two different ways.

From level ℓ down to 0. The finest level ℓ together with the stiffness matrix $\mathbf{L}_\ell$ and the right-hand side $\mathbf{f}_\ell$ is given. Then all levels $k < \ell$ are only introduced to support the multi-grid process. The right-hand side $\mathbf{f}_k$ should be computed by $\mathbf{f}_k := r\mathbf{f}_{k+1}$ for $k = \ell - 1, \ldots, 0$.

From level 0 to ℓ. The nested iteration becomes a part of the discretisation process, if we choose the finer levels adaptively. Then the newest informations (a posteriori error estimates, comparison of $\tilde{\mathbf{u}}_\ell$ and $\tilde{\mathbf{u}}_{\ell-1}$, etc.) can be used to decide whether a further refinement is needed.

In the second case, the loop $k = 1, \ldots, \ell$ in (6.4) should be written as while-statement: '$k := k + 1$, *while the discretisation is not fine enough*'.

6.2 Analysis of the Nested Iteration

The nested iteration (6.4) requires the specification of an iteration number i. The following analysis suggests how to choose i.

Let ζ_k be the contraction number of the multi-grid iteration employed at level k:

$$\|u_k^{j+1} - u_k\| \le \zeta_k \|u_k^j - u_k\|. \tag{6.5}$$

As pointed out before, the numbers ζ_k are uniformly bounded by some $\zeta < 1$. Set

$$\zeta := \max\{\zeta_k : 1 \le k \le \ell\}, \tag{6.6}$$

where ℓ is the maximum level from (6.4). The discretisation error is the (suitably defined) difference between the discrete solution $\mathbf{u}_k = \mathbf{L}_k^{-1}\mathbf{f}_k$ and the continuous solution u. The difference between $\mathbf{u}_k$ and $\mathbf{u}_{k-1} = \mathbf{L}_{k-1}^{-1}\mathbf{f}_{k-1}$ may be called the *relative discretisation error* (error of $\mathbf{u}_{k-1}$ relative to $\mathbf{u}_k$). We suppose an a priori estimate with a consistency order $\varkappa$,

$$\|\tilde{p}\,\mathbf{u}_{k-1} - u_k\| \le C_1 h_k^{\varkappa} \qquad \text{for } 1 \le k \le \ell. \tag{6.7}$$

Note that the exponent $\varkappa$ in (6.7) depends on the consistency order of the discretisation *and* on the interpolation order of $\tilde{p}$. Therefore, we are led to

Remark 6.1 *The interpolation order of $\tilde{p}$ should at least be equal to the consistency order of the discretisation.*

Consider the standard case of a second order discretisation ($\varkappa = 2$) of a second order differential equation ($2m = 2$). By Note 6.1, $\tilde{p}$ should be at least piecewise linear. Hence, $\tilde{p}$ may be the standard prolongation p.

To indicate the levels involved, we write $\tilde{p} = \tilde{p}_{k \leftarrow k-1}$. We define the constants

$$C_{20} := \max\{\|\tilde{p}_{k\leftarrow k-1}\| : 1 \le k \le \ell\}, \tag{6.8a}$$
$$C_{21} := \max\{(h_{k-1}/h_k)^{\varkappa} : 1 \le k \le \ell\}, \tag{6.8b}$$
$$C_2 := C_{20}C_{21}. \tag{6.8c}$$

One can show $C_{20} = 1$ for the most frequent choices of $\tilde{p}$. Moreover, $C_{21} = 2^{\varkappa}$ holds for the usual sequence $h_k = h_0/2^k$. Hence, the value of C_2 is *explicitly known*: $C_2 = 2^{\varkappa}$.

Theorem 6.2 (Error analysis) *Assume (6.7) and $C_2\zeta^i < 1$ with C_2 from (6.8a-c), ζ from (6.6), and i from (6.4c). Set $C_3(\zeta, i) := \zeta^i/(1 - C_2\zeta^i)$. Then the nested iteration (6.4) with i steps of the multi-grid iteration per level results in $\tilde{\mathbf{u}}_k$ $(0 \le k \le \ell)$ satisfying the error estimate*

$$\|\tilde{\mathbf{u}}_k - \mathbf{u}_k\| \le C_3(\zeta, i)\, C_1\, h_k^{\varkappa} \qquad \textit{for all } k = 0, \ldots, \ell \textit{ with } \mathbf{u}_k = \mathbf{L}_k^{-1}\mathbf{f}_k\,. \tag{6.9}$$

Theorem 6.2 ensures that the errors at all levels $k = 0, 1, \ldots, \ell$ differ from the (bound of the) relative discretisation error $C_1 h_k^{\varkappa}$ only by a factor $C_3(\zeta, i)$ which is explicitly known. The only condition on i is $C_2\zeta^i < 1$. The nested iteration is as cheap as possible if $i = 1$ satisfies $C_2\zeta^i < 1$. With $C_2 = 2^{\varkappa}$ from above, this condition becomes $2^{\varkappa}\zeta < 1$. Assuming in addition the standard case of $\varkappa = 2$, we obtain

Remark 6.3 *Assume $C_2 = 4$ and $\zeta < 1/4$. Then Theorem 6.2 holds for $i = 1$. Note that the rates observed in §3.2 are far below the critical bound $1/4$.*

Remark 6.4 (computational work) *Let W_k be work for one multi-grid iteration at level k. Assuming $h_{k-1} \approx 2h_k$ and $\Omega \subset \mathbb{R}^d$, the dimensions n_k of the system at level k should satisfy $n_k \approx 2^d n_{k-1}$. Since the work for $\tilde{p}$ is less dominant, the cost of the nested iteration (6.4) is about $i * W_\ell/\left(1 - 2^{-d}\right)$. In the 2D case, the value becomes $\frac{4}{3}iW_\ell$. Obviously, the complete nested iteration is only insignificantly more expensive than the work spent on level ℓ alone.*

6.3 Numerical Example

We apply the nested iteration (6.4) to the five-point scheme discretisation of

$$\begin{array}{rcll} -\Delta u & = & f := -\Delta(\exp(x + y^2)) & \text{in } \Omega = (0,1) \times (0,1), \\ u & = & \varphi := \exp(x + y^2) & \text{on } \Gamma = \partial\Omega. \end{array} \tag{6.10}$$

The multi-grid iteration MGM in (6.4c) uses $\nu = \gamma = 2$. The results of the nested iteration are shown, where $\tilde{p}$ is the *cubic* interpolation in the interior and the quadratic interpolation near the boundary. Let $\mathbf{u}_k^{\text{exact}}$

be the exact solution $\exp(x+y^2)$ restricted to the grid, while $\mathbf{u}_k$ is the exact discrete solution. We have to distinguish between the *iteration errors* $\|\tilde{\mathbf{u}}_k^i - \mathbf{u}_k\|$, the *total error* $\|\tilde{\mathbf{u}}_k^i - \mathbf{u}_k^{\text{exact}}\|$ with i from (6.4c), and the *discretisation error* $\|\mathbf{u}_k - \mathbf{u}_k^{\text{exact}}\|$. The table shows the maximum norm of the total errors $\|\tilde{\mathbf{u}}_k^i - \mathbf{u}_k^{\text{exact}}\|$ for $i = 1$ and $i = 2$.

ℓ	h_ℓ	$i=1$	$i=2$	$i=\infty$
0	1/2	7.9944658_{10}-2	7.9944658_{10}-2	7.9944658_{10}-2
1	1/4	3.9908756_{10}-2	2.9215605_{10}-2	2.8969488_{10}-2
2	1/8	1.5788721_{10}-2	8.1023136_{10}-3	8.0307789_{10}-3
3	1/16	3.2919346_{10}-3	2.0768391_{10}-3	2.0729855_{10}-3
4	1/32	5.7591549_{10}-4	5.2253758_{10}-4	5.2247399_{10}-4
5	1/64	1.3291689_{10}-4	1.3093946_{10}-4	1.3093956_{10}-4

The last column ($\tilde{\mathbf{u}}_k^\infty = \mathbf{u}_k$!) contains the discretisation error which should be in balance with the iteration error. Obviously, the choice $i = 1$ is sufficient. $i = 2$ needs the double work and cannot improve the total error substantially.

6.4 Use of Acceleration by Conjugate Gradient Methods

Given an iteration, it is often recommended to improve the convergence speed by the method of conjugate gradients (in the positive definite case) or by variants which apply to more general cases (see [14, Sect. 9]). Here, two remarks are of interest.

If the matrix $\mathbf{L}_k$ in $\mathbf{L}_k\mathbf{u}_k = \mathbf{f}_k$ is symmetric and positive definite, one should use a *symmetric* multi-grid variant, i.e., $MGM^{(\nu_1,\nu_2)}$ from Remark 3.1 with $\nu_1 = \nu_2$ and a symmetric smoothing iteration (it is even sufficient that pre-smoothing is adjoint to post-smoothing). Then the standard conjugate gradient (cg) method can be applied. However, the use of cg is recommended only if the rate of the multi-grid convergence is not sufficiently fast; otherwise, the overhead for the cg-method does not pay.

7 Nonlinear Equations

In the following, we consider the nonlinear elliptic problem $\mathcal{L}(u) = 0$, where $\mathcal{L}$ is a nonlinear operator (e.g., $\mathcal{L}(u) = \operatorname{div}\rho(u)\operatorname{grad} u - f$ or $\mathcal{L}(u) = \Delta u + uu_x - f$). In the following, we assume that $\mathcal{L}(u) = 0$ is discretised with respect to a hierarchy of grids:

$$\mathcal{L}_k(\mathbf{u}_k) = \mathbf{f}_k \qquad \text{for } k = 0, \ldots, \ell. \tag{7.1}$$

Even if one is only interested in the solution of $\mathcal{L}_k(\mathbf{u}_k) = \mathbf{0}$, the multi-grid approach in §7.2 leads to problems (7.1) with *small* right-hand sides $\mathbf{f}_k$. The smallness of $\mathbf{f}_k$ ensures that (7.1) has a unique *local* solution (we do *not* require global uniqueness of the solutions).

7.1 Newton's Method and Linear Multi-Grid

Newton's method requires the derivative (Jacobi matrix) of $\mathcal{L}_k$ which we denote by $\mathbf{L}_k(\mathbf{u}_k) = \frac{\partial}{\partial \mathbf{u}_k}\mathcal{L}_k(\mathbf{u}_k)$. Then the Newton iteration $\mathbf{u}_k^{m+1} = \mathbf{u}_k^m - [\mathbf{L}_k(\mathbf{u}_k^m)]^{-1}\left(\mathcal{L}_k(\mathbf{u}_k^m) - \mathbf{f}_k\right)$ requires the solution of the linear system $\mathbf{L}_k(\mathbf{u}_k)\delta_k = \mathbf{d}_k$ for the defect $\mathbf{d}_k := \mathbf{L}_k(\mathbf{u}_k) - \mathbf{f}_k$. The latter task can be performed by the multi-grid iteration from above.

7.2 Nonlinear Multi-Grid Iteration

The approach from §7.1 requires the computation of the Jacobi matrix $\mathbf{L}_k(\mathbf{u}_k)$. This can be avoided by applying the nonlinear multi-grid iteration $NMGM$ from (7.3) below. Since $NMGM$ uses approximations $\tilde{\mathbf{u}}_k$ of $\mathcal{L}_k(\mathbf{u}_k) = \mathbf{0}$ for $k \le \ell - 1$, we start with the *nonlinear nested iteration*, which produces $\tilde{\mathbf{u}}_k$ as well as their defects $\tilde{\mathbf{f}}_k := \mathcal{L}_k(\tilde{\mathbf{u}}_k)$.

```
solve L_0(ũ_0) = f_0 approximately;          (e.g., by Newton's method)
for k := 1 to ℓ do
begin  f̃_{k-1} := L_{k-1}(ũ_{k-1});          (defect of ũ_{k-1})
       ũ_k := p̃ ũ_{k-1};                     (start at level k as in (6.4b))
       for i := 1 to i do ũ_k := NMGM(k, ũ_k, f_k)   (as in (6.4c))
end;
```
(7.2)

Now we define the iteration $NMGM$ which uses $\tilde{\mathbf{u}}_{k-1}, \tilde{\mathbf{f}}_{k-1}$ as reference point (since in the linear case $\tilde{\mathbf{u}}_{k-1} = \mathbf{0}$ is the reference point, we do not see $\tilde{\mathbf{u}}_{k-1}$ in the linear multi-grid method).

```
function NMGM(k, u, f);
begin if k = 0 then NMGM := ũ_0 else          ( ũ_0 approximation to L_0(ũ_0) = f )
    begin for i := 1 to ν do u := S_k(u, f);   (pre-smoothing)
        d := r(L_k(u) − f);                    (restriction of defect)
        ε := ε(d);                             (small positive factor)
        δ := f̃_{k−1} − ε ∗ d;                  (right-hand side at level k − 1)      (7.3)
        v := ũ_{k−1};                          (starting value for correction)
        for i := 1 to γ do v := NMGM(k − 1, v, δ);
        NMGM: = u + p(v − ũ_{k−1})/ε           (coarse-grid correction)
end end;
```

Here, $\mathcal{S}_k(\mathbf{u}_k, \mathbf{f}_k)$ is a nonlinear smoothing iteration for $\mathcal{L}_k(\mathbf{u}_k) = \mathbf{f}_k$. For instance, the analogue of the Richardson iteration (1.4) is $\mathcal{S}_k(\mathbf{u}_k, \mathbf{f}_k) = \mathbf{u}_k - \omega_k(\mathcal{L}_k(\mathbf{u}_k) - f_k)$, where $\omega_k \approx 1/\left\|\mathbf{L}_k(\mathbf{u}_k)\right\|$. The factor $\varepsilon(\mathbf{d})$ may, e.g., be chosen as $\sigma/\left\|\mathbf{d}\right\|$ with a small number σ. The smallness of ε guarantees that $\mathcal{L}_{k-1}(\mathbf{v}_{k-1}) = \tilde{\mathbf{f}}_{k-1} - \varepsilon * \mathbf{d}$ has a unique local solution close to $\tilde{\mathbf{u}}_{k-1}$. Note that $\tilde{\mathbf{f}}_{k-1}, \tilde{\mathbf{u}}_{k-1}$ are produced by (7.2).

Note that (7.3) is a true generalisation of the linear iteration MGM: If $NMGM$ is applied to a linear problem (i.e., $\mathcal{L}_k(\mathbf{u}_k) := \mathbf{L}_k\mathbf{u}_k - \mathbf{f}_k$), it produces the same iterates as MGM independently of the choice of the reference values $\tilde{\mathbf{u}}_{k-1}$.

If $\mathbf{L}_k(\mathbf{u}_k)$ is Lipschitz continuous and if further technical conditions are fulfilled, one can show that the asymptotic convergence rate of $NMGM$ coincides with the rate of the linear iteration MGM applied to the linearised problem with the matrices $\mathbf{L}_k := \mathbf{L}_k(\mathbf{u}_k^{\text{exakt}})$. Also other specifications of $\tilde{\mathbf{u}}_{k-1}, \tilde{\mathbf{f}}_{k-1}, \varepsilon$ are possible (see §9 of [12]).

If, by some reason, the nested iteration is not used, the value $\tilde{\mathbf{u}}_{k-1}$ can be replaced by $r\mathbf{u}_k$, while $\tilde{\mathbf{f}}_{k-1} := \mathcal{L}_{k-1}(\tilde{\mathbf{u}}_{k-1})$ (FAS: "full approximation storage method" from [8]).

8 Eigenvalue Problems

The (continuous) eigenvalue problem reads $Lu = \lambda u$, where u satisfies homogeneous boundary values. The hierarchy of discrete eigenvalue problems is

$$\mathbf{L}_k\mathbf{u}_k = \lambda\mathbf{u}_k \quad \text{for } k = 0, \ldots, \ell$$

(possibly with $\lambda\mathbf{I}$ replaced by the mass matrix $\lambda\mathbf{M}_k$). Again the two-grid method consists of smoothing and coarse-grid correction based on the defect $\mathbf{d}_k = \mathbf{L}_k\mathbf{u}_k - \lambda\mathbf{u}_k$. However, the computation of a correction $\delta\mathbf{u}_k$ from $(\mathbf{L}_k - \lambda\mathbf{I})\delta\mathbf{u}_k = \mathbf{d}_k$ is problematic, since $\mathbf{L}_k - \lambda\mathbf{I}$ becomes singular for the eigenvalue λ. Nevertheless, $(\mathbf{L}_k - \lambda\mathbf{I})\delta\mathbf{u}_k = \mathbf{d}_k$ is solvable, since the right-hand side $\mathbf{d}_k$ belongs to the image space. Furthermore, the non-uniqueness of $\delta\mathbf{u}_k$ is harmless, since the kernel lies in the eigenspace. These statements are only approximately true for the restricted coarse-grid equation $(\mathbf{L}_{k-1} - \lambda\mathbf{I})\mathbf{v}_{k-1} = r\mathbf{d}_{k-1}$. Therefore, certain projections are necessary. The complete multi-grid algorithms can be found in §12 of [12].

If $\mathbf{L}_k$ is not symmetric, the right- and left-eigenvectors can be computed simultaneously. It is advantageous to compute a group of eigenvectors by combining the multi-grid approach with the Ritz method.

9 Applications to the Boundary Element Method (BEM)

There are two quite different groups of BEM problems which can be solved by means of multi-grid methods. Integral equations of the 2nd kind are treated in §9.1, while integral equations with hypersingular kernel are mentioned in §9.2. In both cases, the arising matrices $\mathbf{K}_k$ are fully populated. Except the coarsest grid, the only operation needed is the matrix-vector multiplication $\mathbf{u}_k \mapsto \mathbf{K}_k\mathbf{u}_k$. Its naive implementation requires $\mathcal{O}(n^2)$ arithmetical operations. Therefore, one should use the fast multiplication described in the author's contribution *Panel Clustering Techniques and Hierarchical Matrices for BEM and FEM* in this encyclopaedia.

9.1 Application to Integral Equations of the Second Kind

Fredholm integral equations of second kind have the form $\lambda u = Ku + f$ ($\lambda \neq 0$, f given) with

$$(Ku)(x) := \int_D s(x,y)u(y)dy \qquad \text{for } x \in D,$$

where the kernel s of the integral operator K is given.

The *Picard iteration* $u \mapsto \frac{1}{\lambda}(Ku - f)$ converges only if $|\lambda| > \rho(K)$, but in many important applications the Picard iteration has a smoothing effect: non-smooth functions e are mapped into a smooth function $\frac{1}{\lambda}Ke$ by only one step. This enables the following *multi-grid iteration of the second kind*, which makes use of a hierarchy $\lambda\mathbf{u}_k = \mathbf{K}_k\mathbf{u}_k + \mathbf{f}_k$ of discrete equations.

```
function MGM(k, u, f);                                  MGM solving λu_k = K_k u_k + f_k
begin if k = 0 then MGM := (λI − K_0)^{-1} f else
      begin u := (1/λ)(K_k ∗ u + f);                    (Picard iteration)
            d := r(λu_k − K_k u − f_k);                 (restriction of defect)        (9.1)
            v := 0;                                     (start value for correction)
            for i := 1 to 2 do v := MGM(k − 1, v, d);   (W-cycle)
            MGM := u − pv                               (coarse-grid correction)
end   end;
```

Because of the strong smoothing effect, MGM has convergence rates $\mathcal{O}(h_k^\alpha)$ with $\alpha > 0$. Hence, with increasing dimension (decreasing step size h_k) the iteration becomes faster! The value α depends on the smoothness of the image of K, on the order of the interpolation p, etc.

Note that we need only the multiplication by the matrices $\mathbf{K}_k$. It is not required that K is an integral operator with explicitly known kernel s. Iteration (9.1) applied to a fixed point equation $\lambda u = Ku + f$ has the same properties, provided that K shows the smoothing effect. For further details we refer to §12 in [12] and §5 in [15].

The nonlinear fixed point equation $\lambda u = \mathcal{K}(u)$ can be solved by an analogous version of the nonlinear multi-grid method from §7.

9.2 Application to Integral Equations with Hypersingular Kernel

Boundary value problems $Lu = 0$ with inhomogeneous Neumann conditions can be formulated by means of hypersingular integral operators. For $L = \Delta$ and $\partial u/\partial n = \phi$ on $\Gamma = \partial\Omega \subset \mathbb{R}^3$, the solution u is given by the double-layer potential

$$u(\mathbf{x}) = \int_\Gamma f(\mathbf{y}) \frac{\partial}{\partial n_y} s(\mathbf{x},\mathbf{y}) d\Gamma_y (\mathbf{x} \in \Gamma) \qquad \text{with } s(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi \|\mathbf{x}-\mathbf{y}\|},$$

provided that f satisfies

$$\int_\Gamma f(\mathbf{y}) \frac{\partial}{\partial n_x} \frac{\partial}{\partial n_y} s(\mathbf{x},\mathbf{y}) d\Gamma_y = \phi(\mathbf{x}) \qquad \text{for } \mathbf{x} \in \Gamma.$$

Since $\partial^2 s/\partial n_x \partial n_y$ has a non-integrable singularity, the integral must be understood in the sense of Hadamard. The variational formulation uses the energy space $\mathcal{H} = H^{1/2}(\Gamma)$. Using piecewise linear elements on Γ, we arrive at the setting (4.2) with the symmetric bilinear form

$$a(u,v) = \frac{1}{2} \int_\Gamma \int_\Gamma [u(\mathbf{x}) - u(\mathbf{y})] [v(\mathbf{x}) - v(\mathbf{y})] \frac{\partial}{\partial n_x} \frac{\partial}{\partial n_y} s(\mathbf{x},\mathbf{y}) d\Gamma_x d\Gamma_y$$

(cf. §8.3 of [15]). Since $a(\cdot,\cdot)$ is *elliptic* with respect to the energy space specified above, the discrete problems $\mathbf{A}_k\mathbf{f}_k = \phi_k$ can be solved in the same way as FEM systems from §4. In particular, p and r should be the canonical transfer mappings from §4.3. The only difference is that $\mathbf{A}_k$ is fully populated, whereas FEM matrices are sparse. This is the reason, why the fast panels clustering techniques should be implemented.

9.3 Application to First Kind Integral Equations with Weakly Singular Kernel

In the case of the single layer potential equation $Ku = f$ with $Ku = \int_\Gamma s(\mathbf{x},\mathbf{y})u(\mathbf{y})d\Gamma_x$ (an example of an integral equation of the first kind with weakly singular kernel) the standard smoothing procedure is not applicable. The reason is that the integral operator K has the negative order -1 (while the hypersingular integral operator from §9.2 has the positive order $+1$). As a consequence, the low eigenvalues of K are associated with the oscillatory eigenfunctions, while the high eigenvalues belong to the smooth eigenfunctions. Therefore, a Richardson-like iteration reducing the high frequencies is not a smoothing procedure.

As a remedy (cf. [7]), the smoothing iteration must use a preconditioning of the equation by an operator D of order > 1: $DKu = Df$ or $KDv = f$ or $D_1KD_2v = D_1f$.

References

[1] Bakhvalov NS. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Comput. Math. Math. Phys.* 1966, **6**(5):101–135.

[2] Bastian P, Hackbusch W and Wittum G. Additive and multiplicative multi-grid - a comparison. *Computing* 1998, **60**:345–364.

[3] Braess D. Towards algebraic multigrid for elliptic problems of second order. *Computing* 1995; **55**:379–393.

[4] Braess D, Dryja M and Hackbusch W. Grid transfer for nonconforming FE-discretisations with application to non-matching grids. *Computing* 1999; **63**:1–25.

[5] Brakhage H. Über die numerische Behandlung von Integralgleichungen nach der Quadraturformelmethode. *Numer. Math.* 1960; **2**:183–196.

[6] Bramble JH and Zhang X. The analysis of multigrid methods. In Volume VII of *Handbook of Numerical Analysis*, Ciarlet PG and Lions JL (eds). Elsevier: Amsterdam, 1993; 173-415

[7] Bramble JH, Leyk Z and Pasciak JE. The analysis of multigrid methods for pseudo-differential operators of order minus one. *Math. Comp.* 1994; **63**:461-478.

[8] Brandt A. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.* 1977; **31**:333–390.

[9] Fedorenko RP. A relaxation method for solving elliptic difference equations. *USSR Comput. Math. Math. Phys.* 1961; **1**:1092–1096.

[10] Fedorenko RP. The speed of convergence of one iterative process. *USSR Comput. Math. Math. Phys.* 1964; **4**:227–235.

[11] Haase G, Langer U, Reitzinger R and Schöberl J. Algebraic multigrid methods based on element preconditioning. *Int. J. Comput. Math.* 2001; **78**:575-598

[12] Hackbusch W. *Multi-Grid Methods and Applications.* SCM 4. Springer: Berlin, 1985.

[13] Hackbusch W. *Elliptic Differential Equations. Theory and Numerical Treatment.* Volume 18 of *SCM.* Springer: Berlin, 1992 (2nd printing, Springer: Berlin, 1992) — 2nd German edition: *Theorie und Numerik elliptischer Differentialgleichungen.* Teubner: Stuttgart, 1996.

[14] Hackbusch W. *Iterative Solution of Large Sparse Systems.* Springer: New York, 1994 — 2nd German edition: *Iterative Lösung großer schwachbesetzter Geichungssysteme.* Teubner: Stuttgart, 1993.

[15] Hackbusch W. *Integral Equations. Theory and Numerical Treatment.* ISNM 128. Birkhäuser: Basel, 1995 — 2nd German edition: *Integralgleichungen. Theorie und Numerik.* Teubner: Stuttgart, 1997.

[16] Hackbusch W and Sauter SA. Composite finite elements for the approximation of PDEs on domains with complicated micro-structures. *Numer. Math.* 1997; **75**:447–472.

[17] Hackbusch W and Trottenberg U. *Multigrid Methods.* LNM 960. Springer: Berlin, 1982.

[18] Hackbusch W and Trottenberg U. *Multigrid Methods II.* LNM 1228. Springer: Berlin, 1986.

[19] Hackbusch W and Trottenberg U. *Multigrid Methods III.* ISNM 98. Birkhäuser: Basel, 1991.

[20] Hackbusch W and Wittum G. *Multigrid Methods V.* LNCSE 3. Springer: Berlin, 1998.

[21] Hemker PW and Wesseling P. *Multigrid Methods IV.* ISNM 116. Birkhäuser: Basel, 1994.

[22] Mandel J. Brezina M and Vaněk P. Energy optimization of algebraic multigrid bases. *Computing* 1999; **62**:205–228.

[23] Stüben K. Algebraic multigrid (AMG): experiences and comparisons. *Appl. Math. Comput.* 1983; **13**:419–451.

[24] Trottenberg U, Oosterlee C and Schüller A. *Multigrid.* Academic Press: San Diego, 2001

[25] Wesseling P. *An Introduction to Multigrid Methods.* Wiley: Chichester, 1991.