Chapter 4

Example of a two-grid algorithm

4.1 Introduction

Before we introduce multigrid algorithms, we first give a simple example of a two-grid algorithm. This is a pre-version of a multigrid algorithm. The example shows a simple two-point boundary value problem and a simple iterative solution method for which the behaviour of the iterative process can be analyzed exactly. In this example the main features of a multigrid algorithm are already visible. We consider the two-point boundary value problem

$$-u''(x) = f(x)\,, \quad x \in \Omega = (0,1), \tag{4.1}$$
$$u(0) = u(1) = 0\,.$$

Below, this problem will be discretised and its discretisation will be solved by means of the two-grid algorithm.

4.2 Discretisation

We provide $\bar{\Omega}$ with a uniform mesh $\bar{\Omega}_h = \{x_i | x_i = i/N;\ i = 0,1,...,N\}$, where h equals $1/N$, as is shown in Figure 4.1.

Figure 4.1: The one-dimensional mesh

The problem (4.1) is discretised by a finite difference method (FDM), using the mesh Ω_h. (The reader may check for himself that the use of the finite element method (FEM) leads to exactly the same discrete problem, when piecewise linear test and trial functions are used). Writing $u(x_i) =: u_i$, we have

$$u''(x_i) = \frac{1}{h^2}(u_{i-1} - 2u_i + u_{i+1}) + \mathcal{O}(h^2), \quad i = 1,2,...,N-1.$$

This can be verified with the aid of Taylor-expansions. Keeping this in mind the following discretisation of (4.1) is obvious:

$$\frac{1}{h^2}(\,-u_{i-1} + 2u_i - u_{i+1}) = f_i\,, \quad i = 1,2,...,N-1. \tag{4.2}$$

Implementation of the boundary conditions, $u_0 = u_N = 0$, yields the discrete linear system

$$L_h u_h = f_h \,, \tag{4.3}$$

with the $(N-1)\times(N-1)$ matrix

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

and the $(N-1)$ vectors

$$u_h = (u_1, u_2, ..., u_{N-1})^T.$$
$$f_h = (f_1, f_2, ..., f_{N-1})^T.$$

4.3 The smoothing process

We consider the damped Jacobi-relaxation method for the iterative solution of (4.3). One iteration step in this process reads

$$u_h^{(i+1)} = u_h^{(i)} - \omega D_h^{-1}(L_h u_h^{(i)} - f_h), \tag{4.4}$$

where D_h is the main diagonal of L_h and $\omega \in \mathbb{R}$ is the damping parameter. By $\hat{u}_h$ we denote the solution of (4.3); then the error in the i-th iterand is

$$e_h^{(i)} = u_h^{(i)} - \hat{u}_h.$$

and

$$e_h^{(i+1)} = M_h^{DJM} e_h^{(i)},$$

where M_h^{DJM} is the amplification-operator of the error for the damped Jacobi-method, which reads

$$M_h^{DJM} = I - \omega D_h^{-1} L_h, \tag{4.5}$$

as can be seen easily from (4.4). The eigensystem of the matrix

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

is $\{\lambda_m, \phi_m\}_{m=1,2...N-1}$, with eigenvalues

$$\lambda_m = \frac{2 - 2\,\cos(m\pi h)}{h^2} = \frac{4\,\sin^2(m\pi h/2)}{h^2}\,, \tag{4.6}$$

and eigenfunctions

$$\phi_m = (\sin(m\pi h),\, \sin(2m\pi h),\, ...,\, \sin((N-1)m\pi h))^T\,. \tag{4.7}$$

Noticing that ϕ_m must be considered as a discrete function on the mesh $\Omega_h = \{x_i | x_i = i/N;\, i = 1, 2, ..., N-1\}$, the following terminology is natural. An eigenfunction ϕ_m is said to have a low (high) frequency if m is small (large). From (4.5) and (4.6) it is clear that M_h^{DJM} has the same eigenfunctions ϕ_m as L_h and eigenvalues

$$\lambda(M_h^{DJM})_m = 1 - \frac{\omega}{2} 4 \sin^2(m\pi h/2) = \cos^2(m\pi h/2) + (1-2\omega)\sin^2(m\pi h/2). \tag{4.8}$$

The index m takes the values $1 \le m \le N-1$ and hence the arguments

$$0 < \frac{\pi h}{2} \le \frac{m\pi h}{2} \le \frac{(N-1)\pi h}{2} < \frac{\pi}{2}.$$

In order to get monotonous decreasing $\lambda(M_h^{DJM})_m$ for increasing m, we select $\omega = 1/2$; we then find

$$\lambda(M_h^{DJM})_m = \cos^2(m\pi h/2).$$

We see that eigenfunctions with a low frequency are damped slowly by M_h^{DJM}, whereas eigenfunctions having a high frequency are damped efficiently. After a few iterations with the damped Jacobi method ($\omega = 1/2$), the rapidly varying component in the error will have almost vanished. However, the slowly varying components will hardly be affected: the error has not become much smaller. The total error has become much smoother. Application of one step of the process (4.4) is therefore also called a *smoothing step.*

4.4 The coarse grid correction

We just saw how to approximate $\hat{u}_h$ with a smooth error. Now we will show how to decrease the size of the error when it is smooth. Let $\tilde{u}_h$ be the approximation with a smooth error of the discrete solution $\hat{u}_h$. Then the error $\tilde{e}_h$ given by $\tilde{e}_h = \tilde{u}_h - \hat{u}_h$ satisfies the equation

$$L_h \tilde{e}_h = L_h \tilde{u}_h - L_h \hat{u}_h = L_h \tilde{u}_h - f_h = d_h = -r_h, \tag{4.9}$$

r_h is the is the residual of $\tilde{u}_h$; d_h is the defect of $\tilde{u}_h$. If $\tilde{e}_h$ is a smooth function, we are well able to represent it well on a coarser grid. To find it we may solve the discrete problem on a coarser grid e.g.

$$L_H \tilde{e}_H = -r_H = d_H\ ,\ H\ =\ 2h, \tag{4.10}$$

where (4.10) is a discretisation of (4.9). We want that $\tilde{e}_H$ is a good coarse-grid approximation of $\tilde{e}_h$. For this we need

- a restriction $\bar{R}_{Hh}$,
- a coarse-grid operator L_H,
- a prolongation P_{hH},

as is shown in Figure 4.2. For L_H we take an operator similar to L_h, only with a mesh $H = 2h$ instead of h. Thus, L_H is a $(N/2-1) \times (N/2-1)$ matrix. (We assume that N is an even integer.) A simple prolongation, P_{hH}, is found by linear interpolation. This operator P_{hH} is defined by $u_h = P_{hH} u_H$ such that

$$\left\{ \begin{array}{lcll} u_h(x_i) & = & u_H(x_i) & \text{if } x_i \in \Omega_H, \\ u_h(x_i) & = & (u_H(x_i+h) + u_H(x_i-h))/2 & \text{if } x_i \notin \Omega_H, \end{array} \right.$$

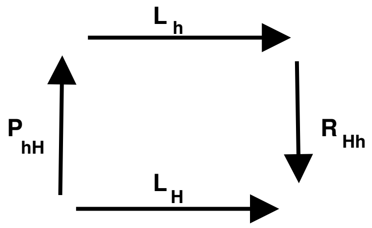

Figure 4.2: fig.4.2.2

for all $x_i = \Omega_h$. A simple restriction, $\bar{R}_{Hh}$, is found by *injection.* This restriction is defined by $f_H = \bar{R}_{Hh} f_h$, with $f_h \in \hat{E}_h$ and $f_H \in \hat{E}_H$ such that

$$f_H(x_i) = f_h(x_i), \text{ for all } x_i \in \Omega_H \,.$$

Another possible *restriction* is the *weighed restriction.* defined by $f_H = \bar{R}^*_{Hh} f_h$, with $f_h \in \hat{E}_h$ and $f_H \in \hat{E}_H$ such that

$$f_H(x_i) = (f_h(x_i + h) + 2 f_h(x_i) + f_h(x_i - h))/4,$$

for all $x_i \in \Omega_H$. The operators $L_H, P_{hH}, \bar{R}_{Hh}$ and $\bar{R}^*_{Hh}$ can be described explicitly by their matrices

$$L_H = \frac{1}{H^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}, \text{ an } (N/2-1) \times (N/2-1) \text{ matrix,} \tag{4.11}$$

$$P_{hH} = \begin{pmatrix} 1/2 & & & & \\ 1 & & & & \\ 1/2 & 1/2 & & & \\ & 1 & & & \\ & 1/2 & 1/2 & & \\ & & 1 & & \\ & & 1/2 & 1/2 & \\ & & & \vdots & \\ & & & 1/2 & 1/2 \\ & & & & 1 \\ & & & & 1/2 \end{pmatrix}, \text{ an } (N-1) \times (N/2-1) \text{ matrix,.} \tag{4.12}$$

$$\bar{R}_{Hh} = \begin{pmatrix} 0 & 1 & 0 & & & & & & \\ & & 0 & 1 & 0 & & & & \\ & & & & 0 & 1 & 0 & & \\ & & & & & & \ddots & & \\ & & & & & & 0 & 1 & 0 \end{pmatrix}, \text{an } (N/2-1) \times (N-1) \text{ matrix, and} \tag{4.13}$$

$$\bar{R}^*_{Hh} = \begin{pmatrix} 1/4 & 1/2 & 1/4 & & & & \\ & & 1/4 & 1/2 & 1/4 & & \\ & & & & \ddots & & \\ & & & & 1/4 & 1/2 & 1/4 \end{pmatrix}, \text{ also an } (N/2-1) \times (N-1) \text{ matrix .} \tag{4.14}$$

The vectors u_h and u_H have the form

$$u_h = (u_h(h), u_h(2h), ..., u_h((N-1)h))^T,$$

and

$$u_H = (u_H(H), u_H(2H), ..., u_H((N/2-1)H))^T.$$

The coarse discretisation (4.10) is used to improve the approximate solution $\tilde{u}_h$ as follows:

- first we compute $\tilde{e}_H$ from either $L_H\tilde{e}_H = d_H = \bar{R}_{Hh}d_h$ or $L_H\tilde{e}_H = d_H = \bar{R}^*_{Hh}d_h$,
- then we subtract the interpolated error from the approximate solution:

$$\hat{\tilde{u}}_h := \tilde{u}_h - P_{hH}\tilde{e}_H.$$

This correction of the approximate solution is the *coarse grid correction step :*

$$\hat{\tilde{u}}_h := \tilde{u}_h - P_{hH}L_H^{-1}\bar{R}_{Hh}(L_h\tilde{u}_h - f_h). \tag{4.15}$$

The amplification operator of the error in a coarse grid correction step is clearly either

$$M_h^{CGC} = I - P_{hH}L_H^{-1}\bar{R}_{Hh}L_h$$

or

$$M_h^{CGC*} = I - P_{hH}L_H^{-1}\bar{R}^*_{Hh}L_h\,.$$

4.5 The two level algorithm

The Two Level Algorithm (TLA), which is a preversion of the Multi Level Algorithm, is an iterative process for the solution of (4.3), in which each step consists of

- a number of p smoothing steps,
- a coarse grid correction step,
- a number of q smoothing steps.

In the order to see what the effect is of a TLA step on the error in an approximate solution, we decompose the error into eigenfunction components

$$e_h = \sum_m \alpha_m\phi_m,$$

and we consider the effect on a single eigenfunction. The amplification operator of the error in one step of the two level algorithm will be denoted by M_h^{TLA}. For ϕ_m as given (4.7) we shall compute

$$M_h^{TLA}\phi_m = (M_h^{DJM})^q(M_h^{CGC})(M_h^{DJM})^p\phi_m \text{ and}$$
$$M_h^{TLA*}\phi_m = (M_h^{DJM})^q(M_h^{CGC*})(M_h^{DJM})^p\phi_m.$$

We will first consider the two level algorithm in which the simple restriction $\bar{R}_{Hh}$ is used in the coarse-grid correction step. To this end we first compute

$$M_h^{CGC}\phi_m = \phi_m - P_{hH}L_H^{-1}\bar{R}_{Hh}L_h\phi_m$$

$$= \phi_m - \frac{4}{h^2}\sin^2(m\pi h/2)P_{hH}L_H^{-1}\bar{R}_{Hh}\phi_m.$$

$$P_{hH}L_H^{-1}\bar{R}_{Hh}\begin{pmatrix}\sin(\pi mh)\\ \sin(2\pi mh)\\ \sin(3\pi mh)\\ \vdots\\ \sin((N-1)\pi mh)\end{pmatrix} = P_{hH}L_H^{-1}\begin{pmatrix}\sin(2\pi mh)\\ \sin(4\pi mh)\\ \sin(6\pi mh)\\ \vdots\\ \sin((N-2)\pi mh)\end{pmatrix}.$$

Since $2h = H$, and

$$(\sin(\pi mH), \sin(2\pi mH), \sin(3\pi mH), \ldots, \sin((N/2-1)\pi mH))^T$$

is an eigenvector of $H^2 L_H$, with eigenvalue $4\sin^2(m\pi H/2)$, the last expression equals

$$\frac{H^2}{4\sin^2(m\pi H/2)}P_{hH}\begin{pmatrix}\sin(\pi mH)\\ \sin(2\pi mH)\\ \sin(3\pi mH)\\ \vdots\\ \sin((N/2-1)\pi mH)\end{pmatrix} =$$

$$= \frac{H^2}{4\sin^2(m\pi H/2)}\cdot\frac{1}{2}\begin{pmatrix}\sin(0)+\sin(\pi mH)\\ 2\ \sin(\pi mH)\\ \sin(\pi mH)+\sin(2\pi mH)\\ 2\ \sin(2\pi mH)\\ \vdots\\ 2\ \sin((N/2-1)\pi mH)\\ \sin((N/2-1)\pi mH)+\sin((N/2)\pi mH)\end{pmatrix} =$$

$$= \frac{h^2}{\sin^2(m\pi h)}\cdot\frac{1}{2}\begin{pmatrix}2.\sin(\pi mh)\cos(\pi mh)\\ 2.\sin(2\pi mh)\\ 2.\sin(3\pi mh)\cos(\pi mh)\\ 2.\sin(4\pi mh)\\ 2.\sin(5\pi mh)\cos(\pi mh)\\ \vdots\\ 2.\sin((N-2)\pi mh)\\ 2.\sin((N-1)\pi mh)\cos(\pi mh)\end{pmatrix}$$

the last equality being valid because

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

If we write $C_m := \cos(\pi mh/2)$ and $S_m := \sin(\pi mh/2)$ and take $a = C_m^2$, $b = -S_m^2$, we obtain

$$\begin{aligned} a - b &= 1\,,\\ a + b &= \cos(m\pi h)\,. \end{aligned}$$

If we furthermore notice that

$$\sin(k\pi(N-m)h) = \sin(k\pi N_h - k\pi mh) = \sin(k\pi - k\pi mh) =$$

$$(-1)^k\sin(-k\pi mh) = (-1)^{k+1}\sin(k\pi mh)$$

then it is clear that

$$\frac{h^2}{\sin^2(m\pi h)} \begin{pmatrix} \sin(\pi mh)\cos(\pi mh) \\ \sin(2\pi mh) \\ \sin(3\pi mh)\cos(\pi mh) \\ \sin(4\pi mh) \\ \vdots \\ \sin((N-2)\pi mh) \\ \sin((N-1)\pi mh)\cos(\pi mh) \end{pmatrix}$$

$$\frac{h^2}{\sin^2(m\pi h)} \left[a \begin{pmatrix} \sin(\pi mh) \\ \sin(2\pi mh) \\ \sin(3\pi mh) \\ \sin(4\pi mh) \\ \vdots \\ \sin((N-1)\pi mh) \end{pmatrix} + b \begin{pmatrix} +\sin(\pi mh) \\ -\sin(2\pi mh) \\ +\sin(3\pi mh) \\ -\sin(4\pi mh) \\ \vdots \\ +\sin((N-1)\pi mh) \end{pmatrix} \right] =$$

$$= \frac{h^2(a\phi_m + b\phi_{N-m})}{\sin^2(\pi mh)},$$

so we have because $\sin^2(\pi mh) = 4S_m^2 C_m^2$

$$P_{hH} L_H^{-1} \bar{R}_{Hh} \phi_m = \frac{h^2(C_m^2 \phi_m - S_m^2 \phi_{N-m})}{4S_m^2 C_m^2} \tag{4.16}$$

and

$$M_h^{CGC} \phi_m = \phi_m - \frac{1}{C_m^2}(C_m^2 \phi_m - S_m^2 \phi_{N-m}) = \frac{S_m^2}{C_m^2} \phi_{N-m}\,.$$

Using (4.8) with $\omega = 1/2$ this results in

$$\begin{aligned} M_h^{TLA} \phi_m &= (M_h^{DJM})^q (M_h^{CGC})(M_h^{DJM})^p \phi_m \\ &= C_m^{2p} (M_h^{DJM})^q (M_h^{CGC}) \phi_m \\ &= C_m^{2p-2} S_m^2 (M_h^{DJM})^q \phi_{N-m} \\ &= C_{N-m}^{2q} S_m^2 C_m^{2p-2} \phi_{N-m}\,. \end{aligned}$$

Because $C_{N-m} = S_m$ and $S_{N-m} = C_m$ we have

$$M_h^{TLA} \phi_m = S_m^{2q+2} C_m^{2p-2} \phi_{N-m}\,,$$

and

$$M_h^{TLA} \phi_{N-m} = C_m^{2q+2} S_m^{2p-2} \phi_m\,.$$

Thus we see that

$$\begin{aligned} M_h^{TLA}(\alpha\,\phi_m + \beta\,\phi_{N-m}) &= \beta\, C_m^{2q+2} S_m^{2p-2} \phi_m + \alpha S_m^{2q+2} C_m^{2p-2} \phi_{N-m} \\ &= \alpha'\,\phi_m + \beta'\,\phi_{N-m}, \end{aligned}$$

which can be denoted in matrix notation by

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \begin{pmatrix} 0 & C_m^{2q+2}\, S_m^{2p-2} \\ S_m^{2q+2}\, C_m^{2p-2} & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix},$$

and we write therefore

$${}_m M_h^{TLA} = \begin{pmatrix} 0 & C_m^{2q+2} S_m^{2p-2} \\ S_m^{2q+2} C_m^{2p-2} & 0 \end{pmatrix}.$$

Analogously we find for the amplification operator of the residual of theTLA-algorithm

$$ {}_m\overline{M}_h^{TLA} = \begin{pmatrix} 0 & S_m^{2p}C_m^{2q} \\ C_m^{2p}S_m^{2q} & 0 \end{pmatrix} . $$

The matrices ${}_mM_h^{TLA}$ and ${}_m\overline{M}_h^{TLA}$ have eigenvalues $\lambda_{1,2} = \pm(C_mS_m)^{p+q}$, thus

$$ \rho(M_h^{TLA}) = \rho(\overline{M}_h^{TLA}) \ \leq \ 2^{-(p+q)} . $$

We now will determine the spectral norm of M_h^{TLA} in order to know by what factor the norm of the error decreases by a single TLA iteration step. To this end it is convenient to have the following lemma available.

Lemma 4.1

$$ \max_{x\in[0,1]} x^a(1-x)^b = \frac{a^ab^b}{(a+b)^{a+b}} =: f(a,b) $$

Proof: The proofs follows by straight forward computation ∎

We will use the following equality

$$ \|M_h^{TLA}\| = \max_{m=1,2,\ldots,N/2} \sqrt{\max_{1,2} |\lambda_{1,2}(({}_mM_h^{TLA})^T({}_mM_h^{TLA}))|} . $$

We easily compute

$$ ({}_mM_h^{TLA})^T({}_mM_h^{TLA}) = \begin{pmatrix} S_m^{4q+4}C_m^{4p-4} & 0 \\ 0 & C_m^{4q+4}S_m^{4p-4} \end{pmatrix} ; $$

Hence,

$$ \begin{aligned} \|M_h^{TLA}\| &= \max_m \{S_m^{2q+2}C_m^{2p-2}, C_m^{2q+2}S_m^{2p-2}\} \\ &\leq \max_{t\in[0,1]} t^{q+1}(1-t)^{p-1} \\ &= f(q+1,p-1) = \frac{(q+1)^{q+1}(p-1)^{p-1}}{(p+q)^{p+q}} . \end{aligned} $$

And we conclude

$$ \begin{array}{lll} \text{if } p=0 & \text{then} & \lim_{h\to 0} \|M_h^{TLA}\| \to \infty \ , \\ \text{if } p=1 & \text{then} & \lim_{h\to 0} \|M_h^{TLA}\| = 1 \ , \\ \text{if } p>1 & \text{then} & \lim_{h\to 0} \|M_h^{TLA}\| \leq f(q+1,p-1) \ < \ 1 \ . \end{array} $$

Now the same can be done for $\overline{M}_h^{TLA}$:

$$ \|\overline{M}_h^{TLA}\| = \max_{m=1,2,\ldots,N/2} \sqrt{\max_{1,2} | \lambda_{1,2}(({}_m\overline{M}_h^{TLA})^T({}_m\overline{M}_h^{TLA})) |} . $$

Since

$$ ({}_m\overline{M}_h^{TLA})^T({}_m\overline{M}_h^{TLA}) = \begin{pmatrix} S_m^{4q}C_m^{4p} & 0 \\ 0 & C_m^{4q}S_m^{4p} \end{pmatrix} $$

it follows that

$$ \begin{aligned} \|\overline{M}_h^{TLA}\| &= \max_m \{S_m^{2q}C_m^{2p}, C_m^{2q}S_m^{2p}\} \\ &\leq \max_{t\in[0,1]} t^q(1-t)^p \\ &= f(q,p) = \frac{q^q\, p^p}{(p+q)^{p+q}} . \end{aligned} $$

And we conclude

$$\begin{array}{lll} \text{if } p=0 & \text{then} & \lim_{h\to 0}\|\bar{M}_h^{TLA}\| = 1 \ , \\ \text{if } q=0 & \text{then} & \lim_{h\to 0}\|\bar{M}_h^{TLA}\| = 1 \ , \\ \text{if } p,q\neq 0 & \text{then} & \lim_{h\to 0}\|\bar{M}_h^{TLA}\| \leq f(q,p) \ < \ 1 \ . \end{array}$$

In summary

$$\rho(M_h^{TLA}) = \rho(\overline{M}_h^{TLA}) \ \leq \ 2^{-(p+q)}$$

$$\begin{array}{lll} \|M_h^{TLA}\| \to \infty & \text{for } h\to 0 & \text{if } p=0 \\ \|M_h^{TLA}\| \to 1 & & \text{if } p=1 \\ \|M_h^{TLA}\| \leq f(q+1,p-1) < 1 & & \text{if } p>1 \end{array}$$

$$\begin{array}{lll} \|\bar{M}_h^{TLA}\| \to 1 & \text{for } h\to 0 & \text{if } p=0 \\ \|\bar{M}_h^{TLA}\| \to 1 & & \text{if } q=0 \\ \|\bar{M}_h^{TLA}\| \leq f(q,p) < 1 & & \text{if } p,q\neq 0 \end{array}$$

If we apply the TLA with $\bar{R}^*_{Hh}$ instead of $\bar{R}_{Hh}$ we get

$$P_{hH}L_H^{-1}\bar{R}^*_{Hh}\phi_m =$$

$$= P_{hH}L_H^{-1}\tfrac{1}{4}\cdot\begin{pmatrix} \sin(m\pi h)+2\sin(2m\pi h)+\sin(3m\pi h) \\ \sin(3m\pi h)+2\sin(4m\pi h)+\sin(5m\pi h) \\ \vdots \\ \sin((2k-1)m\pi h)+2\sin(2km\pi h)+\sin((2k+1)m\pi h) \\ \vdots \\ \sin((N-3)m\pi h)+2\sin((N-2)m\pi h)+\sin((N-1)m\pi h) \end{pmatrix}$$

Using

$$\sin((2k-1)\alpha)+\sin(2k\alpha)+\sin(2k\alpha)+\sin((2k+1)\alpha)$$
$$= \ 2\ \sin(\frac{1}{2}(4k-1)\alpha)\cos(\alpha/2)+2\sin((4k+1)\alpha/2)\cos(\alpha/2)$$
$$= \ 4\ \sin(2k\alpha)\cos^2(\alpha/2)\,,$$

it is not difficult to see that the above expression equals

$$\begin{aligned} & P_{hH}L_H^{-1}\tfrac{1}{4}\begin{pmatrix} 4\cos^2(m\pi h/2)\sin(2m\pi h) \\ 4\cos^2(m\pi h/2)\sin(4m\pi h) \\ \vdots \\ 4\cos^2(m\pi h/2)\sin(2(N/2-1)m\pi h) \end{pmatrix} = \\ & = \ C_m^2 P_{hH}L_H^{-1}\begin{pmatrix} \sin(m\pi H) \\ \sin(2m\pi H) \\ \vdots \\ \sin((N/2-1)m\pi H) \end{pmatrix} = \\ & = \ \frac{h^2(C_m^2\phi_m - S_m^2\phi_{N-m})}{4S_m^2} \end{aligned} \tag{4.17}$$

(cf. the computation which led to (4.16)). Thus we find

$$M_h^{CGC*}\phi_m = \phi_m - \frac{h^2(C_m^2\phi_m - S_m^2\phi_{N-m})}{4S_m^2}\cdot\frac{4S_m^2}{h^2} =$$

$$= S_m^2\phi_m + S_m^2\phi_{N-m} ,$$

and

$$M_h^{CGC*}\phi_{N-m} = C_m^2\phi_m + C_m^2\phi_{N-m} .$$

This can be written in matrix form again as ${}_mM_h^{CGC*} = \begin{pmatrix} S_m^2 & C_m^2 \\ S_m^2 & C_m^2 \end{pmatrix}$ because

$$M_h^{CGC*}(\alpha\phi_m + \beta\phi_{N-m}) = (\alpha S_m^2 + \beta C_m^2)\phi_m + (\alpha S_m^2 + \beta C_m^2)\phi_{N-m}$$

$$= \alpha'\phi_m + \beta'\phi_{N-m} ,$$

with

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = {}_mM_h^{CGC*} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} .$$

Using the fact that ${}_mM_h^{DJM} = \begin{pmatrix} C_m^2 & 0 \\ 0 & S_m^2 \end{pmatrix}$ we find

$${}_mM_h^{TLA*} = ({}_mM_h^{DJM})^q({}_mM_h^{CGC*})({}_mM_h^{DJM})^p$$

$$= \begin{pmatrix} C_m^{2p+2q}S_m^2 & C_m^{2q+2}S_m^{2p} \\ C_m^{2p}S_m^{2q+2} & C_m^2S_m^{2p+2q} \end{pmatrix}$$

Similarly we find for the amplification operator of the residual

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = {}_m\overline{M}_h^{CGC*} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} S_m^2 & S_m^2 \\ C_m^2 & C_m^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} .$$

The effect of the completeTLA-algorithm on low- and high-frequency components of the residual is now described by

$${}_m\overline{M}_h^{TLA*} = \begin{pmatrix} C_m^2 & 0 \\ 0 & S_m^2 \end{pmatrix}^q \begin{pmatrix} S_m^2 & S_m^2 \\ C_m^2 & C_m^2 \end{pmatrix} \begin{pmatrix} C_m^2 & 0 \\ 0 & S_m^2 \end{pmatrix}^p$$

$$= \begin{pmatrix} S_m^2C_m^{2p+2q} & C_m^{2q}S_m^{2p+2} \\ S_m^{2q}C_m^{2p+2} & C_m^2S_m^{2p+2q} \end{pmatrix} .$$

The eigenvalues of both ${}_mM_h^{TLA*}$ and ${}_m\overline{M}_h^{TLA*}$ are 0 and $S_m^2C_m^{2p+2q} + C_m^2S_m^{2p+2q}$. Therefore we have for the spectral radius of M_h^{TLA*} and $\overline{M}_h^{TLA*}$,

$$\rho(M_h^{TLA*}) = \rho(\overline{M}_h^{TLA*}) = \max_m |S_m^2C_m^{2p+2q} + C_m^2S_m^{2p+2q}|$$

$$\leq \max_{t\in[0,1]} (t(1-t)^{p+q} + (1-t)t^{p+q}).$$

Using Lemma 4.1 we find that

$$\rho(M_h^{TLA*}) = \rho(\overline{M}_h^{TLA*}) \leq 2\,f(1, p+q)$$

and thus

$$\begin{aligned} \rho(M_h^{TLA*}) = \rho(\overline{M}_h^{TLA*}) \quad & \leq 1 && \text{if } p+q = 0(\text{ without the lemma}), \\ & \leq 1/2 && \text{if } p+q = 1, \\ & \leq \tfrac{8}{27} && \text{if } p+q = 2. \end{aligned}$$

To compute $\rho(M_h^{TLA*}) = \rho(\overline{M}_h^{TLA*})$ for large $p+q$ we first notice that for large n

$$\max_{t\in[0,1]}((1-t)t^n) = f(1,n) = \frac{n^n}{(n+1)^{n+1}} = \frac{1}{n}(1-\frac{1}{n+1})^{n+1} \approx \frac{1}{ne},$$

which shows that

$$\rho(M_h^{TLA*}) = \rho(\overline{M}_h^{TLA*}) \approx \frac{1}{(p+q)e} \text{ for } (p+q)\to\infty\,.$$

This describes the convergence rate of the two level algorithm for large $p+q$.

In order to see what is the effect of a *single* TLA iteration step on the residual we have to study $\|\overline{M}_h^{TLA*}\|$, the spectral norm of the amplification operator of the residual.

$$\|\overline{M}_h^{TLA*}\| = \max_{m=1,2,\ldots,N/2} \sqrt{\max_{1,2} |\,\lambda_{1,2}((\,{}_m\overline{M}_h^{TLA*})^T(\,{}_m\overline{M}_h^{TLA*}))\,|}$$

$$= \max_m \sqrt{S_m^4 C_m^{4p+4q} + S_m^{4q} C_m^{4p+4} + C_m^{4q} S_m^{4p+4} + C_m^4 S_m^{4p+4q}}$$

$$= \max_m \sqrt{(S_m^4 C_m^{4q} + C_m^4 S_m^{4q})(C_m^{4p} + S_m^{4p})}.$$

Hence,

$$\limsup_{h\to 0} \|\overline{M}_h^{TLA*}\| \approx$$

$$\max_{0\le t\le 1} \sqrt{t^2(1-t)^{2q} + (1-t)^2 t^{2q}} \cdot \sqrt{(1-t)^{2p} + t^{2p}}.$$

If $p=q=0$ we find

$$\lim_{h\to 0} \|\overline{M}_h^{TLA*}\| \approx \max_{0\le t\le 1} \sqrt{t^2 + (1-t)^2}\,\sqrt{2} = \sqrt{2}.$$

If $q=0, p\neq 0$

$$\lim_{h\to 0} \|\overline{M}_h^{TLA*}\| =$$

$$\max_{0\le t\le 1} \sqrt{t^2 + (1-t)^2}\,\sqrt{(1-t)^{2p} + t^{2p}} = 1, \quad (t=0).$$

If $p=0, q\neq 0$

$$\lim_{h\to 0} \|\overline{M}_h^{TLA*}\| = \sqrt{2} \max_{0\le t\le 1} \sqrt{t^2(1-t)^{2q} + (1-t)^2 t^{2q}}\,,$$

and by lemma 4.1 this yields $1/2$ for $q=1$, and for $q\to\infty$ this yields

$$\sqrt{2} \max_{0\le t\le 1} \sqrt{t^2(1-t)^{2q}} = \sqrt{2} \max_{t\in[0,1]} (t(1-t)^q) =$$

$$= \sqrt{2}\,\frac{q^q}{(q+1)^{q+1}} \to \frac{\sqrt{2}}{qe}\,.$$

If $p,q>0$

$$\|\overline{M}_h^{TLA*}\| \approx \max_m \sqrt{S_m^4 C_m^{4q} C_m^{4p}} = \max_m (S_m^2 C_m^{2p+2q})\,,$$

$$\lim_{h\to 0} \|\overline{M}_h^{TLA*}\| = \max_{0\le t\le 1} t(1-t)^{p+q} = \frac{1}{p+q}\left(\frac{p+q}{p+q+1}\right)^{p+q+1} \to \frac{1}{e(p+q)}.$$

for $p+q\to\infty$.

Summarizing we see

$$
\begin{array}{lll}
\rho(M_h^{TLA*}) = \rho(\overline{M}_h^{TLA*}) & \to 1 \ \text{ for } h \to 0, & \text{if } p+q=0, \\
& \to \frac{1}{2} & \text{if } p+q=1, \\
& \to \frac{8}{27} & \text{if } p+q=2, \\
& \to \frac{1}{e(p+q)} & \text{for } (p+q) \to \infty \cdot
\end{array}
$$

$$
\begin{array}{lll}
\|\overline{M}_h^{TLA*}\| & \to \sqrt{2} \text{ for } h \to 0, & \text{if } p=0, q=0 \\
& \to 1 & \text{if } p \neq 0, q=0, \\
& \to \frac{1}{2} & \text{if } p=0, q=1, \\
& \to \frac{\sqrt{2}}{qe} & \text{if } p=0, q \to \infty, \\
& \to \frac{1}{e(p+q)} & \text{if } p \neq 0 \, (p+q) \to \infty.
\end{array}
$$

$\|M_h^{TLA*}\|$ yields the same values, with p and q interchanged.

Remark: In [9] we can see that the difference between the above values for finite values of h and for $h \to 0$ is very small ($\mathcal{O}(h^2)$).

We conclude that relaxations *after* coarse grid corrections are of use for a small norm of the amplification operator of the *residual,* whereas relaxations *before* yield a small norm of the operator of the *error.* We noticed this already in a more theoretical context in chapter 3.

Chapter 5

Multigrid algorithms

5.1 Introduction

In the framework of defect correction processes multigrid algorithms are easy to explain. For this purpose we consider a continuous problem 2.1 and two discretisations of this problem on grids with meshwidth h and H:

$$F_h z_h = y_h \quad \text{and} \quad F_H z_H = y_H,\, H > h. \tag{5.1}$$

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

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

The relaxation steps are defect correction steps as e.g. damped Jacobi (JOR), Gauss-Seidel (GS), symmetric Gauss-Seidel (SGS), incomplete LU-decomposition iteration (ILU) etc. Their main purpose is to reduce the unsmooth part of the error (see Section 3.3). The remaining, smooth error can be represented well on a coarser grid by means of some restriction.

The coarse grid correction step is a defect correction step of type DCPA where the approximate inverse $\tilde{G}_h$ is given by

$$\tilde{G}_h = P_{hH} F_H^{-1} \bar{R}_{Hh} \tag{5.2}$$

(see Figure 5.1). The use of the approximate inverse (5.2) implies that we solve the defect equation on a coarse grid, with the help of a coarse discretisation of our problem. This is only meaningful if (part of) the error before the defect correction step is representable on the coarse grid.

Thus far we have only described an algorithm using two grids. If we do not solve the defect equation on a coarser grid directly, but if we approximate its solution by application of a few iteration steps of the same algorithm on the coarser level, we obtain a recursive process where we have to solve a discretised problem directly only on the very coarsest grid. The resulting process is a true multigrid-algorithm. One complete iteration step in a multigrid process is called a multigrid *cycle.*

5.2 Two-level algorithms

Again we consider the two discretisations (5.1). Their relation via restrictions and prolongations is shown in Figure 5.1. We present the algorithms in the form of ALGOL-like programmes.

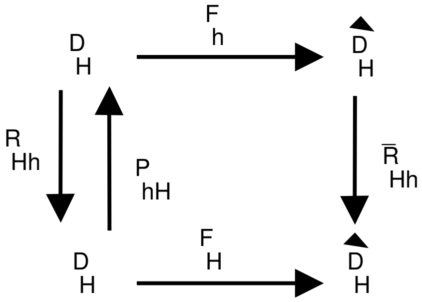

Figure 5.1: Relations between operators and spaces in a two-grid algorithm

First we describe two auxiliary procedures:

- **proc** solve = (**operator** F_h, **ref vector** z_h, **vector** y_h) **void:**
  This procedure uses the operator F_h and the right-hand-side y_h to solve (approximately) the equation
  $$F_h z_h = y_h\ .$$
  On entry, z_h should contain an appropriate initial value for the (possibly) iterative solution process. On exit z_h contains a (better approximate) solution of the problem $F_h z_h = y_h$.

- **proc** relax = (**operator** F_h, **ref vector** z_h, **vector** y_h) **void:**
  This procedure performs one iteration step of a suitable relaxation method for the equation $F_h z_h = y_h$.

Now we explain the essential coarse grid correction step. Given an approximation z_h to the true solution z_h^* of our discrete problem, we consider the residual

$$r_h = y_h - F_h z_h.$$

With the error $e_h = z_h - z_h^*$, we have

$$F_h(z_h - e_h) = y_h = r_h + F_h z_h. \tag{5.3}$$

For a linear operator F_h this reduces to

$$F_h e_h = -r_h. \tag{5.4}$$

Instead of solving equation (5.3) directly, we compute the solution of a similar equation on a coarse grid

$$F_H w_H = \bar{R}_{Hh} r_h + F_H R_{Hh} z_h \tag{5.5}$$

and then use $P_{hH}(w_H - R_{Hh} z_h)$ as an approximation for the error $-e_h$, and hence as a correction quantity in the DCP. For a linear operator F_h the above reduces to

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

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

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

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

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

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

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

the (DCPN) reads as

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

5.3 Multi-level algorithms

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

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

```
        else to σ do MGM (i − 1, p, q, σ, z, r)
    fi;
    z_h := z_h + P_hH (w_H − R_Hh z_h);
    to q do relax (F_h, z_h, y_h); end;
```

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

```
proc MGML = (int i, p, q, σ, ref [ ] vector z, r ) void:
begin
    operator F_h = F_{h_i}, F_H = F_{h_{i-1}} ;
    vector z_h = z_{h_i}, y_h = r_{h_i} ;
    to p do relax (F_h, z_h, y_h);
    vector r_H = r_{h_{i-1}} := R̄_Hh (y_h − F_h z_h);
    vector w_H = z_{h_{i-1}} := 0;
    if i = 0
        then solve (F_H, w_H, r_H)
        else to σ do MGML(i − 1, p, q, σ, z, r)
    fi;
    z_h := z_h + P_hH w_H;
    to q do relax (F_h, z_h, y_h);
end;
```

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

Theorem 5.1 Consider an application of MGM where $h_i/h_{i+1} = H/h$ is constant for all $i = 0, 1, 2, \ldots$. Let d be the dimension of the grid, i.e. the number of space dimensions. If $\sigma < (H/h)^d$ then the total amount of work in a multigrid cycle is proportional to the amount of the work on the fine grid.

Proof: Let W be the amount of computational work needed to perform relaxations, operator evaluations, restriction and prolongation on the finest grid. On every next coarser grid the number of nodal points is reduced by a factor $(h/H)^d$. Hence the amount of work on the coarser grid is reduced by the same factor. If we consider an infinite number of grids, the total amount of work is given by

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

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

In the above multigrid algorithms (MGM, MGML) the fixed numbers p, q and σ determine the *strategy* of the algorithm. Other multigrid algorithms may terminate iterations sooner or later, depending on the convergence or other conditions that can be checked during the computation. Multigrid algorithms that make use of this possibility have an *adaptive strategy;* algorithms where the iterations are controlled only by the fixed numbers p, q and σ have a *fixed strategy.* MG-cycles with $\sigma = 1$ are called V-cycles, those with $\sigma = 2$ are called W-cycles. V-cycles that have either $p = 0$ or $q = 0$ are called *sawtooth cycles.*

In the following figures we show for some fixed strategies how is switched between the different levels of discretisation. We see that -essentially- most relaxation sweeps are performed on the coarser levels. In all diagrams the number of the levels is 4, the coarsest level is denoted by 0. Segments between tick-marks on a level > 0 denote the execution of a relaxation step on this level; a segment on level 0 denotes the direct solution on the coarsest level. Let

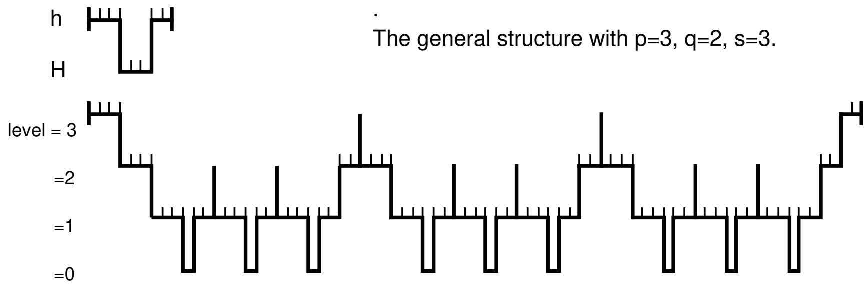

Figure 5.2: A general multigrid cycle with $p = 3, q = 2, \sigma = 3$

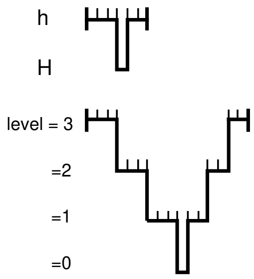

Figure 5.3: A V-cycle with $p = 3, q = 2$.

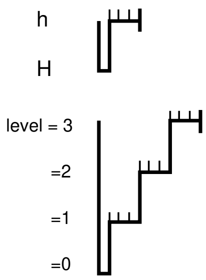

Figure 5.4: A saw-tooth cycle with $p = 0, q = 3, \sigma = 1$.

M_h^{REL} and $\overline{M}_h^{REL}$ denote the amplification operators of the relaxation for error and residual respectively. For TGML the amplification operator for the error is given by

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

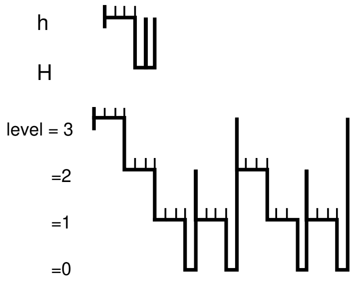

Figure 5.5: A W-cycle with $p = 3, q = 0, \sigma = 2$

We denote the amplification operator of a multi-level iteration step (MGML) on the h-level of discretisation by $M_h^{MGML,p,q,\sigma}$, or M_h^{MGML} for short. The same amplification operator on the next coarser level we denote by $M_H^{MGML,p,q,\sigma}$ or M_H^{MGML}. In the multigrid cycle the approximate inverse is not given by

$$P_{hH} F_H^{-1} \bar{R}_{Hh},$$

because F_H^{-1} is approximated by application of σ steps of a DCP. The amplification operator of this DCP is given by M_H^{MGML}. Hence the approximate inverse of the σ iteration steps together is given by

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

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

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

and we have

$$\begin{aligned} M_h^{MGML,p,q,\sigma} &= (M_h^{REL})^q (I_h - P_{hH}(I_H - (M_H^{MGML})^\sigma) F_H^{-1} \bar{R}_{Hh} F_h)(M_h^{REL})^p \\ &= M_h^{TGML} + (M_h^{REL})^q P_{hH} (M_H^{MGML})^\sigma F_H^{-1} \bar{R}_{Hh} F_h (M_h^{REL})^p . \end{aligned} \tag{5.6}$$

5.4 Full multigrid method (FMG)

The multigrid cycles we described in the previous section yield iterative improvement of a solution on a fine grid and therefore they need some initial estimate of the solution on this finest grid to start with. One possible algorithm is to obtain the initial estimate by interpolation from a solution on the next coarser grid, which has previously been calculated by a similar algorithm. Using this algorithm we start solving the problem on the coarsest grid. An algorithm of this type is called a *Nested Iteration* process:

proc nested iteration = (**int** l,[ ] **int** i, **ref** [ ] **vector** z, y) **void**:
begin
    solve $(F_0, z[0], y[0])$; { – sufficiently accurate – }
    for k **to** l
    begin
        $z[k] := P^*_{k,k-1} z[k-1]$;
        for m **to** $i[k]$ iteration $(k, z[k], y[k])$;
    end
end

Remarks:

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

The procedure "iteration" represents one step of a suitable iterative solution process. In the multigrid context we will use MGM (FAS) or MGML (CS) to replace "iteration". In this case we call the resulting method a *full multigrid method* (FMG). In an FMG-algorithm the coarse grids have a double function:

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

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

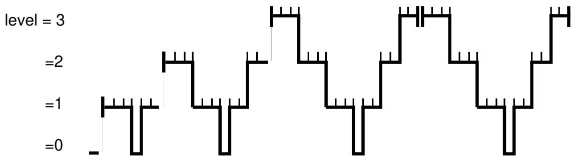

Figure 5.6: A FMG starting phase and an additional V-cycle

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

Theorem 5.2 If we consider a Nested Iteration process with a sequence of discretisations $F_k z_k = y_k$, $k=0,1,2,\dots$,

$$F_k = F_{h_k}, \quad z_k = z_{h_k}, \quad y_k = y_{h_k}, \qquad h_0 > h_1 > h_2 > \dots , \tag{5.7}$$

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

$$\|P^*_{k,k-1} z_{k-1} - z_k\| \le C_0 h^p_{k-1} , \tag{5.8}$$

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

$$\|z_k^{(j+1)} - z_k\| \le C_2 \|z_k^{(j)} - z_k\| ,$$

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

$$\|\tilde{z}_k - z_k\|_{E_k} \leq \frac{C_1^p C_2^i}{1 - C_1^p C_2^i \|P\|} C_0 h_k^p \,, \tag{5.9}$$

where $\|P\| = \sup_k \|P^*_{k,k-1}\|$ and it is assumed that $C_1^p C_2^i \|P\| \leq 1$.
Proof: For each $k \geq 0$ we know

$$\|z_k^{(i)} - z_k\| \leq C_2^i \, \|z_k^{(0)} - z_k\| \,.$$

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

$$\|\tilde{z}_0 - z_0\| \leq C_1^p \; C_2^i \; C_0 h_0^p,$$

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

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

so that

$$\begin{aligned}
\|z_k^{(i)} - z_k\| &\leq C_2^i \; \|P\| \; \|z_{k-1}^{(i)} - z_{k-1}\| + C_2^i C_1^p C_0 h_k^p \\
&\leq C_2^i C_1^p C_0 h_k^p + C_2^i \; \|P\| \; \{C_2^i C_1^p C_0 h_{k-1}^p + C_2^i \; \|P\| \; \{...\}\} \\
&= C_0 C_1^p C_2^i \, [h_k^p + C_2^i \|P\| \{h_{k-1}^p + C_2^i \|P\| \{...\}\}] \\
&= C_0 C_1^p C_2^i h_k^p \, [1 + C_2^i \|P\| \{C_1^p + C_2^i \|P\| \{...\}\}] \\
&= C_0 C_1^p C_2^i h_k^p \, [1 + C_2^i \|P\| C_1^p + (C_2^i \|P\| C_1^p)^2 + ...] \\
&= \frac{C_1^p C_2^i}{1 - C_1^p C_2^i \|P\|} \, C_0 h_k^p \,.
\end{aligned}$$

■

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

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

Remark: We notice that our definition of relative convergence (Definition 2.29) was in terms of

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

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

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

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

Theorem 5.3 Consider the continuous equation $Fz = y$ and let the sequence of discretisations $F_h z_h = y_h,\ h \in \mathcal{H}$ have an order of convergence p. If there exists a nested sequence of stable prolongations

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

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

Proof:

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

■

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