1. Erläutern Sie die algorithmische Struktur eines 'additiven Multigrid-Zyklus' AMGM (siehe Anhang, Abschnitt 4.2.1 aus [18]), insbesondere was seine Parallelisierbarkeit angeht. Verifizieren Sie auch die Identität (4.29).

2. Präsentieren Sie den Beweis von Theorem 14.2.

3. In the attachment you find a one-dimensional version of an assertion closely related to Theorem 14.3 (Lemma 6.5.36 by J. Xu, with some typos corrected), with proof. This refers to the 1D Poisson example, with $\mathcal{V} = H_0^1(\Omega)$, $\Omega = [0,1]$ (think of continuous, piecewise differentiable functions with zero boundary values). Here,
$$A(u,v) = a(u,v) = \int_0^1 u'(x)\, v'(x)\, dx$$
Instead of the H^1 norm $\|u\|_1$ you may take the H^1 seminorm
$$|u|_1 = \sqrt{a(u,u)}$$
which is an equivalent norm on $H_0^1(\Omega)$, and analogously on the subintervals Ω_i. In the last line of Xu's proof of Lemma 6.5.36, '$\cdot \lesssim \cdot$' means '$\cdot \le C \cdot$' with some constant C.

   Study the proof of Lemma 6.5.36, make a sketch, verify all steps and write down your own version of the proof.

   Hint: In a first step, show the (evident) estimate
$$a(I_0\, v, I_0\, v)_{[\xi,\eta]} \le a(v,v)_{[\xi,\eta]}$$
for any subinterval $[\xi,\eta] \subseteq [0,1]$, where $a(u,v)_{[\xi,\eta]} = \int_\xi^\eta u'(x)\, v'(x)\, dx$, and $I_0\, v$ is the linear interpolant of v over $[\xi,\eta]$ with $(I_0\, v)(\xi) = v(\xi)$ and $(I_0\, v)(\eta) = v(\eta)$.

   Remark: In the formulation of Xu the underlying space $\mathcal{V}$ and the $\mathcal{V}_i$, $i = 1 \ldots J$ have infinite dimension, while $\mathcal{V}_0$ is the (finite-dimensional) space of piecewise linears over the mesh (uniform partition) with mesh size $h_0 = 1/(J+1)$. In practice, $\mathcal{V} = \mathcal{V}_h$ and the $\mathcal{V}_i$ will be a finite-dimensional spaces, e.g., containing piecewise linears with respect to a finer mesh with mesh size $h < H$.

4. Exercise 14.2:

   Realize the overlapping AS preconditioner for the 1D Poisson example on $\Omega = [0,1]$ and FEM discretization with piecewise linears (mesh size h) and two overlapping subdomains (subintervals) $\Omega_1 = [0,\xi]$, $\Omega_2 = [\eta,1]$ with $\xi > \eta$, $\xi - \eta = \delta = H$. Choose, e.g., $\xi = 0.3$, $H = 0.1$ (fixed), $h = 0.01$, and reduce the size of h, keeping H fixed. Observe the performance of `pcg` preconditioned by such an AS step.

5. Eine Überlegung zum sogenannten 'Schur-Komplement':

   Wir schreiben eine SPD-Matrix $A \in \mathbb{R}^{n\times n}$ in der Block-Gestalt
$$A = \left(\begin{array}{c|c} B & C \\ \hline C^T & D \end{array}\right)$$
   mit quadratischen Diagonalblöcken B, D (diese sind ebenfalls SPD).

   a) Zeigen Sie: Die Matrix A lässt sich zerlegen gemäß
$$A = \left(\begin{array}{c|c} I & 0 \\ \hline C^T B^{-1} & I \end{array}\right) \cdot \left(\begin{array}{c|c} B & 0 \\ \hline 0 & S \end{array}\right) \cdot \left(\begin{array}{c|c} I & B^{-1}C \\ \hline 0 & I \end{array}\right) \tag{1}$$
   mit dem *Schur-Komplement* $S = D - C^T B^{-1} C$ von B. Zeigen Sie auch, dass S ebenfalls SPD ist.

b) Zeigen Sie für $x = (x_1, x_2)^T$:

$$\|(x_1, 0)^T\|_A = \|x_1\|_B, \qquad \|x_2\|_S = \inf_{x_1} \|(x_1, x_2)^T\|_A = \|(-B^{-1}C\, x_2, x_2)^T\|_A.$$

c) Zeigen Sie:

$$\lambda_{min}(A) \leq \lambda_{min}(S) \leq \lambda_{max}(S) \leq \lambda_{max}(A)$$

d) Die Gestalt von A legt die Wahl von Vorkonditionierern der folgenden Bauart nahe:

$$\tilde{A} = \left(\begin{array}{c|c} I & 0 \\ \hline K^T & I \end{array}\right) \cdot \left(\begin{array}{c|c} \tilde{B} & 0 \\ \hline 0 & \tilde{S} \end{array}\right) \cdot \left(\begin{array}{c|c} I & K \\ \hline 0 & I \end{array}\right) \approx A$$

wobei $\tilde{B} \approx B$ und $\tilde{S} \approx S$ SPD und 'einfach zu invertieren' sein sollen, und $K \approx B^{-1}C$.

(i) Zeigen Sie: $\tilde{A}$ ist SPD für beliebige K.

(ii) Erläutern Sie, wie man basierend auf Cholesky-Zerlegungen für $\tilde{B}$ und $\tilde{S}$ den Vorkonditionierer $r \mapsto \tilde{A}^{-1}\, r$ realisiert.

(iii) Analog zu (ii), speziell für $K = \tilde{B}^{-1}C$.

4.2 ▪ Grid decomposition methods

For linear algebraic systems resulting from partial differential equations discretizations, a large group of vector space decompositions for the purpose of preconditioning is based on hierarchical grid decompositions. This links the approach with classical multigrid techniques described in the previous chapter. To make this connection even more clear, let us introduce another variant of multigrid iteration known as the additive multigrid method.

4.2.1 ▪ Additive multigrid method

The multigrid preconditioners considered earlier in the text are sometimes called *multiplicative* multigrid methods. Consequently, computations in a multiplicative method are performed on different grid levels and cannot be run in parallel. For computers with parallel architecture, it is often more efficient to use an *additive* variant of a multigrid method. Moreover, additive multigrid methods turn out to be more convenient (also from the implementation point of view) for problems discretized on locally refined grids. Building preconditioners for locally refined grids is discussed at the end of this chapter.

Thus, assume we are looking for a solution to $Az = f$, where A is a finite element stiffness matrix with discrete space $\mathbb{U}_h$. Repeating assumptions from Section 3.3, let a hierarchy of finite element spaces be given as

$$\mathbb{U}_0 \subset \mathbb{U}_1 \subset \cdots \subset \mathbb{U}_k \subset \cdots \subset \mathbb{U}_l = \mathbb{U}_h. \tag{4.27}$$

Subspace $\mathbb{U}_0$ corresponds to the coarsest grid. Canonical isomorphisms between $\mathbb{U}_k$ and Euclidean vector spaces of coefficients $\mathbb{R}_k$ lead to the chain of vector spaces,

$$\mathbb{R}_0 \underset{r_1}{\overset{p_1}{\rightleftarrows}} \mathbb{R}_1 \underset{r_2}{\overset{p_2}{\rightleftarrows}} \cdots \underset{r_k}{\overset{p_k}{\rightleftarrows}} \mathbb{R}_k \underset{r_{k+1}}{\overset{p_{k+1}}{\rightleftarrows}} \cdots \underset{r_l}{\overset{p_l}{\rightleftarrows}} \mathbb{R}_l = \mathbb{R}_h, \tag{4.28}$$

with prolongation and restriction operators p_k and r_k. Assume that each of the grid level matrices $A_l = A$ and A_k, $k = 0, \ldots, l-1$, is given or computed as was explained in subsection 3.2.3.

We define one iteration of an additive method for solving the system $Az = f$ as the one execution of the procedure $z^{new} = AMGM(l, z^{old}, f)$, where z^{old} is a given approximation to z:

$\bar{z} = AMGM(k, z, f)$
{
If $k = 0$, then

0. $\bar{z} = A_0^{-1} f$ (exact solve on the coarsest grid), exit the procedure;

Else

1. $d = r_k(A_k z - f)$ (restriction of the residual to a coarser grid);
2. $z^0 = z$, $z^{i+1} = z^i - W_k^{-1}(A_k z^i - f)$, $i = 0, \ldots, \nu - 1$ (ν basic smoothing iterations);
3. $e = 0$ (the initial guess to the error on the coarser grid);
4. call $\bar{e} = AMGM(k-1, e, d)$ (recursive call of the multigrid on the coarser grid);
5. $\bar{z} = z^\nu - \theta p_k \bar{e}$ (coarse grid correction of z^ν, $\theta \in \mathbb{R}$ is a parameter),

}

Note that the recursive call to the procedure in step 4 is independent of the result of smoothing iterations in step 2 and can be performed in parallel. This is in contrast to

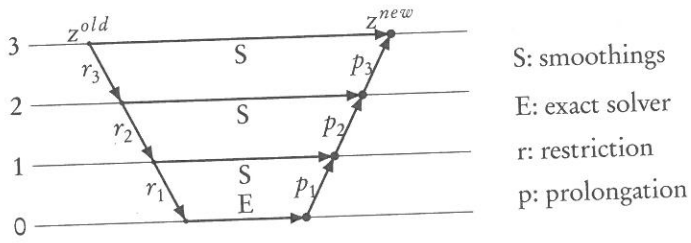

Figure 4.1. *One cycle of the additive method.*

V- and W-cycles studied previously. Therefore, the smoothing iterations on all grid levels can be performed in parallel. Schematically the additive method is illustrated in Figure 4.1.

The iteration matrix of the additive two-grid method is

$$M_A = I - (\theta\, p_h A_H^{-1} r_h + (I - S^\nu) A_h^{-1}) A_h, \tag{4.29}$$

where S is the iteration matrix of the smoothing iterations, and the coarse problem in step 4 is solved exactly. We leave checking (4.29) to the reader for an exercise.

From (4.29) we see that the additive multigrid method does not turn out to be the exact solver if $||S|| < 1$ (smoothing iterations are convergent) and $\nu \to \infty$. This is in contrast to the multiplicative one. The additive method would turn to an exact solver only if θ depends on ν, and $\theta(\nu) \to 0$ for $\nu \to \infty$. To ensure convergence of the additive method, one has to make a correct choice of the parameter θ. The optimal value of θ is not known a priori. For this reason, the additive method is almost always used as a preconditioner in a Krylov subspace method with $\theta = 1$ and rarely as an independent iterator. We study convergence of such iterations further in this chapter.

To see how the additive multigrid method fits into the framework of space decomposition, let us write out the preconditioner B_A defined by one iteration of AMGM,

$$B_A = \theta^l \mathbf{p}_0 A_0^{-1} \mathbf{r}_0 + \sum_{k=1}^{l} \theta^{l-k} \mathbf{p}_k (I - S_k^\nu) A_k^{-1} \mathbf{r}_k,$$

where $\mathbf{p}_l = I$ and $\mathbf{p}_k = p_l \cdots p_{k+1}$, $\mathbf{r}_k = \mathbf{p}_k^*$; cf. the diagram (4.28). For the particular case of $\theta = 1$ and $\nu = 1$, we see that the additive multigrid method defines the preconditioner of a form similar to (4.6):

$$B_A = \mathbf{p}_0 A_0^{-1} \mathbf{r}_0 + \sum_{k=1}^{l} \mathbf{p}_k W_k^{-1} \mathbf{r}_k. \tag{4.30}$$

A natural question is, Does (4.30) correspond to a space decomposition? If yes, then it can be studied within the developed framework.

4.2.2 ▪ BPX preconditioner

We take another viewpoint on the additive multigrid method above. Consider a finite element problem with the symmetric elliptic bilinear form $a(\cdot,\cdot)$ on a space $\mathbb{U}_h$. For example, $a(\psi,\phi) = (\nabla\psi, \nabla\phi)$ for the Poisson problem. This gives rise to the system of linear algebraic equations

$$Az = f \quad \text{and} \quad \langle A\cdot,\cdot\rangle = a(P_l\cdot, P_l\cdot). \tag{4.31}$$

where $\mathcal{I}_k \in \mathbb{R}^{n\times n_k}$ is the representation matrix of the nodal basis $\{\phi_i^k\}$ in $\mathcal{M}_k$ in terms of the nodal basis $\{\phi_i\}$ of $\mathcal{M}$, i.e.

$$\Phi^k = \Phi\mathcal{I}_k$$

with $\Phi^k = (\phi_1^k, \cdots, \phi_{n_k}^k)$ and $\Phi = (\phi_1, \cdots, \phi_n)$.

Note that if we define $\mathcal{I}_k^{k+1} \in \mathbb{R}^{n_{k+1}\times n_k}$ such that

$$\Phi^k = \Phi^{k+1}\mathcal{I}_k^{k+1},$$

Then

$$\mathcal{I}_k = \mathcal{I}_{J-1}^J \cdots \mathcal{I}_{k+1}^{k+2}\mathcal{I}_k^{k+1}.$$

This identity is very useful for the efficient implementation of (6.4.34) on both serial and parallel fashions, cf. Xu and Qin [.xu qin.]. If $\mathcal{R}_k$ are given by the Richardson iteration, we have

$$\mathcal{B} = \sum_{k=1}^{J} h_k \mathcal{I}_k \mathcal{I}_k^t. \tag{6.4.35}$$

From (6.4.34) or (6.4.35), we see that the preconditioner depends entirely on the transformation between the nodal bases on multilevel spaces. For this reason, we shall call such kinds of preconditioners *multilevel nodal basis preconditioners.*

6.5 Domain decomposition with overlaps

Given an integer J, consider the following uniform partition of $(0, 1)$:

$$0 = y_0 < y_1 < \cdots < y_{J+1} = 1, \quad y_j = \frac{j}{J+1}(j = 0 : J+1).$$

We assume that $\{y_j\} \subset \mathcal{N}_h$.

Based on this partition, a domain decomposition may be obtained as follows:

$$\Omega = \sum_{i=1}^{J} \Omega_i \quad \text{with} \quad \Omega_i = (y_{i-1}, y_{i+1}).$$

Based on these subdomains, the subspaces $\mathcal{V}_i$ $(1 \le i \le J)$ are defined by

$$\mathcal{V}_i = \{v \in \mathcal{V} : v(x) = 0 \quad \forall\, x \in \Omega \setminus \Omega_i\}.$$

If the number of subdomains J is too large, the above subspaces are not sufficient to produce an optimal algorithms. In regard to this consideration, we introduce a coarse finite element subspace $\mathcal{V}_0$ defined from the aforementioned partition of size $h_0 = 1/(J+1)$.

Lemma 6.5.36 *For the subspaces $\mathcal{V}_i$ $(0 \le i \le J)$, we have*

$$\mathcal{V} = \sum_{i=0}^{J} \mathcal{V}_i. \tag{6.5.37}$$

Furthermore there is a constant C_0 that is independent of h, h_0 or J, such that for any $v \in \mathcal{V}$, there are $v_i \in \mathcal{V}_i$ that satisfy $v = \sum_{i=0}^{J} v_i$ and

$$\sum_{i=0}^{J} a(v_i, v_i) \le C_0 a(v, v). \tag{6.5.38}$$

Proof. Let $I_0 : \mathcal{V} \mapsto \mathcal{V}_0$ be the nodal value interpolant. Given $v \in \mathcal{V}$, define $v_0 = I_0 v$,

$$v_1(x) = \begin{cases} (v - I_0 v)(x) & y_0 \le x \le y_1 \\ \frac{1}{2}(v - I_0 v)(x) & y_1 \le x \le y_2 \\ 0 & y_2 \le x \le 1, \end{cases}$$

for $j = 2 : J-1$

$$v_j(x) = \begin{cases} \frac{1}{2}(v - I_0 v)(x) & y_{j-1} \le x \le y_{j+1} \\ 0 & \text{elsewhere} \end{cases}$$

and

$$v_J(x) = \begin{cases} 0 & 0 \le x \le y_{J-1} \\ \frac{1}{2}(v - I_0 v)(x) & y_{J-1} \le x \le y_J \\ (v - I_0 v)(x) & y_J \le x \le 1. \end{cases}$$

Obviously $v_i \in \mathcal{V}_i$ satisfying $v = \sum_{i=0}^{J} v_i$ and

$$\begin{aligned} \sum_{i=0}^{J} A(v_i, v_i) &\lesssim \|I_0 v\|_1^2 + \sum_{i=1}^{J} \|(v - I_0 v)\|_{1,\Omega_i}^2 \\ &\lesssim \|I_0 v\|_1^2 + \|(v - I_0 v)\|_1^2 \\ &\lesssim \|v\|_1^2. \end{aligned}$$

□

Lemma 6.5.39

$$K_0 \le C_0/\omega_0 \quad \text{and} \quad K_1 \le C.$$

Proof. The first estimate follows directly from (5.3.15) and (6.5.36). To prove the second estimate, we define

$$Z_i = \{1 \le j \le J : \Omega_i \cap \Omega_j \ne \emptyset\}.$$

By the construction of the domain decomposition, there exists a fixed integer n_0 such that

$$|Z_i| \le n_0 \quad \forall\, 1 \le i \le J.$$