Chapter 12

Multigrid and Domain Decomposition Methods

Chapters 10 and 11 dealt with preconditioners designed for general classes of matrices. The origin of the problem was not a factor in defining the preconditioner, although analysis of the preconditioner was sometimes limited to problems arising from certain types of partial differential equations. In this chapter we deal with preconditioners designed specifically for problems arising from partial differential equations. The methods are intended for use with broad classes of problems and are not restricted to one particular equation. Attempts have been made to extend some of these ideas to general linear systems, as in algebraic multigrid methods, but the extensions are not immediate.

Because the methods are designed for partial differential equation problems, their analysis may require detailed knowledge of the properties of finite difference and finite element approximations, while we are assuming just a basic familiarity with these ideas. For this reason, we restrict our analysis to the model problem, where properties shared by more general finite element and finite difference approximations can be verified directly, and we indicate how the analysis can be extended.

12.1. Multigrid Methods.

Multigrid methods were not originally described as a combination of an iteration scheme and a preconditioner, but it is clear that they can be viewed in this way. The first multigrid methods used simple iteration, so we will start with that approach. It will be apparent that the same preconditioners can be used with any of the other Krylov subspace methods described in Part I of this book.

Recall the simple iteration scheme

$$x_k = x_{k-1} + M^{-1}(b - Ax_{k-1}). \tag{12.1}$$

The error $e_k \equiv A^{-1}b - x_k$ is given by

$$e_k = (I - M^{-1}A)e_{k-1}, \tag{12.2}$$

so the norm of the error satisfies

$$\|e_k\| \leq \|I - M^{-1}A\| \cdot \|e_{k-1}\|.$$

The error is reduced quickly if $\|I - M^{-1}A\| << 1$.

Most multigrid methods can be written in the general form (12.1), where the iterates x_k represent quantities generated after a *coarse grid correction cycle* and a given number of *relaxation sweeps.* That is, given an approximation x_{k-1} to the solution, the multigrid algorithm generates a new approximation $x_{k-1,0}$ via a formula of the form

$$x_{k-1,0} = x_{k-1} + C(b - Ax_{k-1}), \tag{12.3}$$

where the matrix C represents a *coarse grid* approximation to A^{-1}. The method then generates a certain number, say ℓ, of new approximations, $x_{k-1,j}$, $j = 1, \ldots, \ell$ by performing *relaxation sweeps* of the form

$$x_{k-1,j} = x_{k-1,j-1} + G(b - Ax_{k-1,j-1}), \quad j = 1, \ldots, \ell, \tag{12.4}$$

where the matrix G also represents an approximation to A^{-1}. If we denote by x_k the quantity $x_{k-1,\ell}$, then we find

$$x_k = x_{k-1} + [I - (I - GA)^\ell (I - CA)A^{-1}](b - Ax_{k-1}),$$

and the error e_k satisfies

$$e_k = (I - GA)^\ell (I - CA)e_{k-1}. \tag{12.5}$$

Thus, for multigrid methods, the matrix $I - M^{-1}A$ in (12.2) is of the special form $(I - GA)^\ell (I - CA)$ for certain matrices C and G.

12.1.1. Aggregation Methods.

We first analyze iterations of the form (12.3–12.4) in a general setting, where the matrix C involves the inverse of a smaller matrix. Such methods are sometimes called *aggregation methods.*

While multigrid and aggregation methods can be applied to non-Hermitian and indefinite problems, the analysis here will be restricted to Hermitian positive definite problems. We will estimate the rate at which the A-norm of the error, $\|e_k\|_A \equiv \langle e_k, Ae_k\rangle^{1/2}$, is reduced. Taking norms on each side in equation (12.5), we find that

$$\|e_k\|_A \leq \|(I - GA)^\ell (I - CA)\|_A \cdot \|e_{k-1}\|_A.$$

The quantity $\|(I - GA)^\ell (I - CA)\|_A$ is called the *contraction number* of the method and will be denoted by σ. In terms of the 2-norm, σ is given by

$$\sigma = \|(I - A^{1/2}GA^{1/2})^\ell (I - A^{1/2}CA^{1/2})\|. \tag{12.6}$$

A simple bound for σ is

$$\sigma \leq \|I - A^{1/2}GA^{1/2}\|^\ell \cdot \|I - A^{1/2}CA^{1/2}\|. \tag{12.7}$$

The methods to be considered use matrices C and G, for which $\|I - A^{1/2}CA^{1/2}\| = 1$ and $\|I - A^{1/2}GA^{1/2}\| < 1$. Hence the methods are *convergent* whenever A is Hermitian and positive definite, and, moreover, they reduce the A-norm of the error at each step.

Inequality (12.7), however, is too crude an estimate to provide much useful information about the rate of convergence. In fact, the methods to be considered use matrices C and G, which are designed to complement each other in such a way that the norm of the matrix product in (12.6) is much less than the product of the norms in (12.7). Instead of inequality (12.7), we use the definition of the matrix norm to estimate σ by

$$\sigma \leq \max_{\substack{\|y\| = \|I - A^{1/2}CA^{1/2}\| \\ y \in \text{range}(I - A^{1/2}CA^{1/2})}} \|(I - A^{1/2}GA^{1/2})^{\ell} y\|. \tag{12.8}$$

If the range of $I - A^{1/2}CA^{1/2}$ is a restricted set of vectors on which $I - A^{1/2}GA^{1/2}$ is highly contractive, then the bound in (12.8) may be much smaller than that in (12.7).

We now define the form of the matrix C in iteration (12.3). Suppose A is an n-by-n matrix and $\hat{n} < n$. Let $I_{\hat{n}}^{n}$ be an arbitrary n-by-$\hat{n}$ matrix of rank $\hat{n}$, and define an $\hat{n}$-by-n matrix $I_{n}^{\hat{n}}$ by

$$I_{n}^{\hat{n}} = (I_{\hat{n}}^{n})^{H}. \tag{12.9}$$

Define an $\hat{n}$-by-$\hat{n}$ matrix $\hat{A}$ by

$$\hat{A} = I_{n}^{\hat{n}} A I_{\hat{n}}^{n}, \tag{12.10}$$

and take C to be the matrix

$$C = I_{\hat{n}}^{n} \hat{A}^{-1} I_{n}^{\hat{n}}. \tag{12.11}$$

The following theorem shows that when C is defined in this way, the matrix $A^{1/2}CA^{1/2}$ is just the orthogonal projector from $\mathbf{C}^n$ to the $\hat{n}$-dimensional subspace $A^{1/2} \cdot \text{range}(I_{\hat{n}}^{n})$.

THEOREM 12.1.1. *If C is defined by* (12.9–12.11), *then*

$$A^{1/2} \cdot \mathcal{R}(I - A^{1/2}CA^{1/2}) \subseteq \mathcal{N}(I_{n}^{\hat{n}}), \tag{12.12}$$

where $\mathcal{R}(\cdot)$ denotes the range and $\mathcal{N}(\cdot)$ the null space of an operator. The matrix $A^{1/2}CA^{1/2}$ is an orthogonal projector from $\mathbf{C}^n$ to an $\hat{n}$-dimensional subspace and hence

$$\|I - A^{1/2}CA^{1/2}\| = 1. \tag{12.13}$$

Proof. Using definitions (12.10) and (12.11), we find

$$I_{n}^{\hat{n}} A^{1/2}(I - A^{1/2}CA^{1/2}) = I_{n}^{\hat{n}} A^{1/2} - (I_{n}^{\hat{n}} A I_{\hat{n}}^{n}) A_{\hat{n}}^{-1} I_{n}^{\hat{n}} A^{1/2} = 0.$$

This establishes (12.12). To establish (12.13), note that since $I - A^{1/2}CA^{1/2}$ is a Hermitian matrix, its norm is the absolute value of its largest eigenvalue. If z is an eigenvector of this matrix with eigenvalue λ, then, using (12.12), we can write

$$I_n^{\hat{n}}A^{1/2}(I - A^{1/2}CA^{1/2})z = 0 = \lambda I_n^{\hat{n}}A^{1/2}z.$$

It follows that either $\lambda = 0$ or $A^{1/2}z \in \mathcal{N}(I_n^{\hat{n}})$. In the latter case, $CA^{1/2}z$ is zero and hence $\lambda = 1$. Since the eigenvalues of $I - A^{1/2}CA^{1/2}$, and hence of $A^{1/2}CA^{1/2}$, are 0's and 1's, this establishes that $A^{1/2}CA^{1/2}$ is an orthogonal projector and that (12.13) holds. □

Applying the theorem, inequality (12.8) becomes

$$\sigma \leq \max_{\substack{\|y\| = 1 \\ A^{1/2}y \in \mathcal{N}(I_n^{\hat{n}})}} \|(I - A^{1/2}GA^{1/2})^{\ell}y\|,$$

and since the null space of $I_n^{\hat{n}}$ is the orthogonal complement of the range of $I_{\hat{n}}^n$, this can be written as

$$\sigma \leq \max_{\substack{\|y\| = 1 \\ y \perp A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^n)}} \|(I - A^{1/2}GA^{1/2})^{\ell}y\|. \tag{12.14}$$

Suppose the matrix G is given. Let $d_1^2 \geq \cdots \geq d_n^2$ denote the eigenvalues of $(I - A^{1/2}GA^{1/2})^{\ell^H}(I - A^{1/2}GA^{1/2})^{\ell}$, and let $v_1, \ldots, v_n$ denote the corresponding orthonormal eigenvectors. For any vector y we can write $y = \sum_{i=1}^n \langle y, v_i\rangle v_i$ and

$$\|(I - A^{1/2}GA^{1/2})^{\ell}y\|^2 = \sum_{i=1}^{n} \langle y, v_i\rangle^2 d_i^2. \tag{12.15}$$

Now, in general, we have

$$\max_{\|y\|=1} \|(I - A^{1/2}GA^{1/2})^{\ell}y\| = |d_1|, \tag{12.16}$$

but with the additional constraint $y \perp A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^n)$, a smaller bound may be attained. If $I_{\hat{n}}^n$ can be chosen so that $v_1, \ldots, v_{\hat{n}}$—the eigenvectors corresponding to the $\hat{n}$ largest eigenvalues—lie in the space $A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^n)$, then y will have no components in the direction of these eigenvectors and expression (12.15) can be replaced by

$$\|(I - A^{1/2}GA^{1/2})^{\ell}y\|^2 = \sum_{i=\hat{n}+1}^{n} \langle y, v_i\rangle^2 d_i^2.$$

Under these *ideal* conditions—$v_1, \ldots, v_{\hat{n}} \in A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^n)$—the bound (12.16) is replaced by

$$\max_{\substack{\|y\| = 1 \\ y \perp A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^n)}} \|(I - A^{1/2}GA^{1/2})^{\ell}y\| = |d_{\hat{n}+1}|. \tag{12.17}$$

As an example, suppose G is taken to be of the form

$$G = \gamma I, \tag{12.18}$$

where the constant γ is chosen in an optimal or near optimal way. Then the eigenvectors $v_1, \ldots, v_n$ of $(I - A^{1/2}GA^{1/2})^{\ell H}(I - A^{1/2}GA^{1/2})^{\ell}$ are just the eigenvectors of A, and the eigenvalues $d_1^2, \ldots, d_n^2$ of this matrix are each of the form $(1 - \gamma\lambda_i)^{2\ell}$ for some i, where $\lambda_1 \le \cdots \le \lambda_n$ are the eigenvalues of A. In this case, the bound (12.16) becomes

$$\max_{\|y\|=1} \|(I - \gamma A)^{\ell} y\| = \max\{|1 - \gamma\lambda_1|^{\ell}, |1 - \gamma\lambda_n|^{\ell}\}.$$

To minimize this bound, take $\gamma = 2/(\lambda_n + \lambda_1)$ and then

$$\max_{\|y\|=1} \|(I - \gamma A)^{\ell} y\| = \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^{\ell} = \left(\frac{\kappa - 1}{\kappa + 1}\right)^{\ell}, \quad \kappa = \frac{\lambda_n}{\lambda_1}.$$

This is the usual bound on the convergence rate for the method of steepest descent.

On the other hand, suppose some of the eigenvectors of A, say, those corresponding to the $\hat{n}$ smallest eigenvalues of A, lie in the desired space $A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^{n})$. Then an improved bound like (12.17) holds, and this bound becomes

$$\max_{\substack{\|y\| = 1 \\ y \perp A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^{n})}} \|(I - \gamma A)^{\ell} y\| = \max\{|1 - \gamma\lambda_{\hat{n}+1}|^{\ell}, |1 - \gamma\lambda_n|^{\ell}\}. \tag{12.19}$$

To minimize this bound, take $\gamma = 2/(\lambda_n + \lambda_{\hat{n}+1})$, and (12.19) becomes

$$\max_{\substack{\|y\| = 1 \\ y \perp A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^{n})}} \|(I - \gamma A)^{\ell} y\| = \left(\frac{\lambda_n - \lambda_{\hat{n}+1}}{\lambda_n + \lambda_{\hat{n}+1}}\right)^{\ell} = \left(\frac{\hat{\kappa} - 1}{\hat{\kappa} + 1}\right)^{\ell}, \tag{12.20}$$

$$\hat{\kappa} = \frac{\lambda_n}{\lambda_{\hat{n}+1}}.$$

Thus, the effective condition number of A is reduced from $\kappa = \lambda_n/\lambda_1$ to $\hat{\kappa} = \lambda_n/\lambda_{\hat{n}+1}$. If the latter ratio is much smaller, as is typically the case when the matrix A approximates a self-adjoint elliptic differential operator, then much faster convergence is achieved by using a partial step of the form (12.3) than by iterating only with steps of the form (12.4).

12.1.2. Analysis of a Two-Grid Method for the Model Problem.

Recall that for the model problem $-\Delta u = f$ in the unit square with Dirichlet boundary conditions, the matrix A arising from a 5-point finite difference approximation on a grid of m-by-m interior points with spacing $h = 1/(m+1)$

and the (p, q)-components of the corresponding eigenvectors are

$$v_{p,q}^{(i,j)} = 2h \sin(ihp\pi) \sin(jhq\pi), \quad p, q = 1, \ldots, m, \tag{12.22}$$

as shown in Theorem 9.1.2. The eigenvalues are all positive and the smallest and largest eigenvalues are given in Corollary 9.1.2:

$$\lambda_{min} = 8h^{-2} \sin^2\left(\frac{\pi h}{2}\right) = 2\pi^2 + O(h^2), \tag{12.23}$$

$$\lambda_{max} = 8h^{-2} \sin^2\left(\frac{m\pi h}{2}\right) = 8h^{-2} + O(1). \tag{12.24}$$

For ih or jh of size $O(1)$, say, $i > (m+1)/4$ or $j > (m+1)/4$, we have $\lambda_{i,j} = O(h^{-2})$, which is the same order as λ_{max}. Hence if the $((m+1)/4)^2$ smallest eigencomponents in the error could be annihilated by solving a smaller problem, using a partial step of the form (12.3), then the ratio of the largest to the smallest remaining eigenvalue would be $O(1)$, independent of h. The bound (12.20) on the convergence rate of iteration (12.3–12.4) with $G = \gamma I$ would be a constant less than one and independent of h!

Note also that the eigenvectors corresponding to the smaller values of i and j are "low frequency." That is, the sine functions do not go through many periods as p and q range from 1 to m. Thus these eigenvectors could be represented on a coarser grid. We now show how the annihilation of the small eigencomponents can be accomplished, approximately, by solving the problem on a coarser grid.

Assume that $m+1$ is even and let $\hat{m} = (m-1)/2$ be the number of interior points in each direction of a coarser grid with spacing $\hat{h} = 2h$. Let $\hat{n} = \hat{m}^2$. Define the coarse-to-fine prolongation matrix $I_{\hat{n}}^n$ to be *linear* interpolation along horizontal, vertical, and diagonal (southwest to northeast) lines. That is, if w is a vector defined at the nodes $(1,1)$ – $(\hat{m}, \hat{m})$ of the coarse grid, define

$$(I_{\hat{n}}^n w)_{p,q} = \begin{cases} w_{p/2,q/2} & \text{if } p, q \text{ even}, \\ \frac{1}{2}(w_{(p+1)/2,q/2} + w_{(p-1)/2,q/2}) & \text{if } p \text{ odd, } q \text{ even}, \\ \frac{1}{2}(w_{p/2,(q+1)/2} + w_{p/2,(q-1)/2}) & \text{if } p \text{ even, } q \text{ odd}, \\ \frac{1}{2}(w_{(p+1)/2,(q+1)/2} + w_{(p-1)/2,(q-1)/2}) & \text{if } p, q \text{ odd}, \end{cases}$$

$$p, q = 1, \ldots, m. \tag{12.25}$$

THEOREM 12.1.2. *Let A be the* 5*-point Laplacian matrix so that $\lambda_{i,j}$ and $v^{(i,j)}$ satisfy* (12.21–12.22), *and let $I_{\hat{n}}^n$ be defined by* (12.25). *Let $v^{(1)}, \ldots, v^{(s)}$ denote the eigenvectors corresponding to the s smallest eigenvalues, $\lambda_1 \leq \cdots \leq \lambda_s$. If v is any vector in span$[v^{(1)}, \ldots, v^{(s)}]$, with $\|v\| = 1$, then v can be written in the form*

$$v = A^{1/2} I_{\hat{n}}^n w + \delta, \quad \textit{where } \|\delta\| \leq ch\lambda_s^{1/2} \tag{12.26}$$

for some $\hat{n}$-vector w where $c = 2 + \sqrt{6}$.

Proof. First suppose that $v = v^{(i,j)}$ is an eigenvector. Then for any $\hat{n}$-vector w, we have

$$\|v^{(i,j)} - A^{1/2} I_{\hat{n}}^{n} w\| = \|A^{1/2}(\lambda_{i,j}^{-1/2} v^{(i,j)} - I_{\hat{n}}^{n} w)\| \le \|A^{1/2}\| \ \lambda_{i,j}^{-1/2} \ \|v^{(i,j)} - I_{\hat{n}}^{n} \tilde{w}\|,$$

where $\tilde{w} = \lambda_{i,j}^{1/2} w$. Since, from (12.24), the norm of $A^{1/2}$ is bounded by $2\sqrt{2}\ h^{-1}$, we have

$$\|v^{(i,j)} - A^{1/2} I_{\hat{n}}^{n} w\| \le 2\sqrt{2}\ h^{-1}\ \lambda_{i,j}^{-1/2}\ \|v^{(i,j)} - I_{\hat{n}}^{n} \tilde{w}\|. \tag{12.27}$$

Let $\tilde{w}^{(i,j)}$ match $v^{(i,j)}$ at the nodes of the coarse grid so that

$$\left(I_{\hat{n}}^{n} \tilde{w}^{(i,j)}\right)_{p,q} = \begin{cases} v_{p,q}^{(i,j)} & \text{if } p,q \text{ even}, \\ \frac{1}{2}\left(v_{p-1,q}^{(i,j)} + v_{p+1,q}^{(i,j)}\right) & \text{if } p \text{ odd, } q \text{ even}, \\ \frac{1}{2}\left(v_{p,q-1}^{(i,j)} + v_{p,q+1}^{(i,j)}\right) & \text{if } p \text{ even, } q \text{ odd}, \\ \frac{1}{2}\left(v_{p-1,q-1}^{(i,j)} + v_{p+1,q+1}^{(i,j)}\right) & \text{if } p,q \text{ odd}, \end{cases}$$

for $p,q = 1,\ldots,m$. Then from formula (12.22) for $v^{(i,j)}$ it follows that

$$\left(v^{(i,j)} - I_{\hat{n}}^{n} \tilde{w}^{(i,j)}\right)_{p,q} = \begin{cases} 0 & \text{if } p,q \text{ even}, \\ v_{p,q}^{(i,j)}(1-\cos(ih\pi)) & \text{if } p \text{ odd, } q \text{ even}, \\ v_{p,q}^{(i,j)}(1-\cos(jh\pi)) & \text{if } p \text{ even, } q \text{ odd}, \\ v_{p,q}^{(i,j)}(1-\cos(ih\pi)\cos(jh\pi)) - & . \\ \quad z_{p,q}^{(i,j)} \sin(ih\pi)\sin(jh\pi) & \text{if } p,q \text{ odd}, \end{cases}$$

$$z_{p,q}^{(i,j)} \equiv 2h\cos(ihp\pi)\cos(jhq\pi) \quad \text{for } p,q \text{ odd}. \tag{12.28}$$

Note that if $z_{p,q}^{(i,j)}$ is defined by (12.28) for all points p and q, then the vectors $z^{(i,j)}$ are orthonormal (Exercise 12.1), as are the vectors $v^{(i,j)}$ (Exercise 9.3). If $z^{(i,j)}$ is defined to be 0 at the other grid points, then it can be checked (Exercise 12.1) that

$$\|z^{(i,j)}\|^2 \le 1/4 \quad \text{and} \quad \langle z^{(i,j)}, v^{(i,j)}\rangle = 0.$$

Summing over p and q and using the formula $1 - \cos x = 2\sin^2(x/2)$, we have

$$\begin{aligned} \|v^{(i,j)} - I_{\hat{n}}^{n} \tilde{w}^{(i,j)}\|^2 \ = \ & 4\sin^4(ih\pi/2) \sum_{p \text{ odd, } q \text{ even}} (v_{p,q}^{(i,j)})^2 \\ & +4\sin^4(jh\pi/2) \sum_{p \text{ even, } q \text{ odd}} (v_{p,q}^{(i,j)})^2 \\ & +(1-\cos(ih\pi)\cos(jh\pi))^2 \sum_{p,q \text{ odd}} (v_{p,q}^{(i,j)})^2 \\ & \sin^2(ih\pi)\sin^2(jh\pi) \sum_{p,q \text{ odd}} +(z_{p,q}^{(i,j)})^2. \end{aligned}$$

From (12.21) it follows that

$$\max\{4\sin^4(ih\pi/2), 4\sin^4(jh\pi/2)\} \le \frac{1}{4}h^4\lambda_{i,j}^2, \tag{12.29}$$

$$\begin{aligned}
(1-\cos(ih\pi)\cos(jh\pi))^2 &= \left[1-\left(1-2\sin^2(ih\pi/2)\right)\left(1-2\sin^2(jh\pi/2)\right)\right]^2 \\
&\le 4\left(\sin^2(ih\pi/2)+\sin^2(jh\pi/2)\right)^2 \\
&\le \frac{1}{4}h^4\lambda_{i,j}^2, \quad \text{and} \qquad (12.30)
\end{aligned}$$

$$\begin{aligned}
\sin^2(ih\pi)\sin^2(jh\pi) &= [2\sin(ih\pi/2)\cos(ih\pi/2)]^2[2\sin(jh\pi/2)\cos(jh\pi/2)]^2 \\
&\le 16\sin^2(ih\pi/2)\sin^2(jh\pi/2) \\
&\le \frac{1}{2}h^4\lambda_{i,j}^2. \qquad (12.31)
\end{aligned}$$

Making these substitutions and using the fact that $\|v^{(i,j)}\|^2 = 1$ and $\|z^{(i,j)}\|^2 \le 1/4$, we can write

$$\|v^{(i,j)} - I_{\tilde{n}}^n \tilde{w}^{(i,j)}\|^2 \le \frac{1}{4}h^4\lambda_{i,j}^2\|v^{(i,j)}\|^2 + \frac{1}{2}h^4\lambda_{i,j}^2\|z^{(i,j)}\|^2 \le \frac{3}{8}h^4\lambda_{i,j}^2$$

or

$$\|v^{(i,j)} - I_{\tilde{n}}^n \tilde{w}^{(i,j)}\| \le \sqrt{\frac{3}{8}}h^2\lambda_{i,j}.$$

Combining this with (12.27) gives

$$\|v^{(i,j)} - A^{1/2} I_{\tilde{n}}^n w^{(i,j)}\| \le \sqrt{3}\ h\ \lambda_{i,j}^{1/2}. \tag{12.32}$$

Now let V_s be the matrix whose columns are the eigenvectors $v^{(i,j)}$ corresponding to the s smallest eigenvalues, and let $v = V_s\xi$ be an arbitrary vector in the span of the first s eigenvectors, with $\|v\| = \|\xi\| = 1$. Consider approximating v by the vector $A^{1/2}I_{\tilde{n}}^n(\tilde{W}_s\Lambda_s^{-1/2}\xi)$, where $\tilde{W}_s$ has columns $\tilde{w}^{(i,j)}$ corresponding to the s smallest eigenvalues and Λ_s is the diagonal matrix of these eigenvalues. The difference $\delta \equiv v - A^{1/2}I_{\tilde{n}}^n(\tilde{W}_s\Lambda_s^{-1/2}\xi)$ is given by

$$\delta = A^{1/2}\,\Delta_s\,\Lambda_s^{-1/2}\xi, \tag{12.33}$$

where the columns of Δ_s are the vectors $\delta^{(i,j)} \equiv v^{(i,j)} - I_{\tilde{n}}^n\tilde{w}^{(i,j)}$. From (12.28) the vector $\Delta_s\Lambda_s^{-1/2}\xi$ can be written in the form $d+e$, where

$$d = \begin{cases} 0 & \text{at even–even grid points,} \\ V_s^{oe}\hat{\xi} & \text{at odd–even grid points,} \\ V_s^{eo}\tilde{\xi} & \text{at even–odd grid points,} \\ V_s^{oo}\check{\xi} & \text{at odd–odd grid points,} \end{cases}$$

$$e = \begin{cases} 0 & \text{at even–even, odd–even, and even–odd grid points,} \\ Z_s^{oo}\dot{\xi} & \text{at odd–odd grid points,} \end{cases}$$

where V_s^{oe}, V_s^{eo}, V_s^{oo}, and Z_s^{oo} consist of the rows of V_s or Z_s (the matrix whose columns are the vectors $z^{(i,j)}$ defined in (12.28) for indices (i, j) corresponding to the s smallest eigenvalues) corresponding to the odd–even, even–odd, and odd–odd grid points and

$$\hat{\xi}_{i,j} = \lambda_{i,j}^{-1/2}\xi_{i,j}(1-\cos(ih\pi)), \quad \tilde{\xi}_{i,j} = \lambda_{i,j}^{-1/2}\xi_{i,j}(1-\cos(jh\pi)),$$

$$\check{\xi}_{i,j} = \lambda_{i,j}^{-1/2}\xi_{i,j}(1-\cos(ih\pi)\cos(jh\pi)), \quad \dot{\xi}_{i,j} = \lambda_{i,j}^{-1/2}\xi_{i,j}\sin(ih\pi)\sin(jh\pi).$$

Each of the matrices V_s^{oe}, V_s^{eo}, V_s^{oo}, and Z_s^{oo} has norm less than or equal to 1, because it is part of an orthogonal matrix. Since $\|\xi\| = 1$, it follows that

$$\begin{aligned} \|d\|^2 &\le \max_{i,j} \lambda_{i,j}^{-1}(1-\cos(ih\pi))^2 + \max_{i,j} \lambda_{i,j}^{-1}(1-\cos(jh\pi))^2 \\ &\quad + \max_{i,j} \lambda_{i,j}^{-1}(1-\cos(ih\pi)\cos(jh\pi))^2, \\ \|e\|^2 &\le \max_{i,j} \lambda_{i,j}^{-1}\sin^2(ih\pi)\sin^2(jh\pi). \end{aligned}$$

Using (12.29–12.31), we have

$$\|d\|^2 \le \frac{3}{4}h^4\lambda_s, \quad \|e\|^2 \le \frac{1}{2}h^4\lambda_s,$$

and from (12.33) and (12.24), it follows that

$$\|\delta\| \le \|A^{1/2}\| \left(\|d\| + \|e\|\right) \le (2+\sqrt{6})h\lambda_s^{1/2}. \quad \square$$

The constant $2 + \sqrt{6}$ in (12.26) is not the best possible estimate because the piecewise linear interpolant of $v^{(i,j)}$ used in the theorem is not the best possible coarse grid approximation to $v^{(i,j)}$.

COROLLARY 12.1.1. *If $y \perp A^{1/2} \cdot \mathcal{R}(I_{\hat{n}}^n)$ and $\|y\| = 1$, then*

$$\sum_{i=1}^{s} \langle y, v^{(i)} \rangle^2 \le c^2h^2\lambda_s, \quad c = 2+\sqrt{6}. \tag{12.34}$$

Proof. The left-hand side of inequality (12.34) is the square of the norm of the vector $v = \sum_{i=1}^{s} \langle y, v^{(i)} \rangle v^{(i)}$, and according to Theorem 12.1.2 this vector satisfies

$$\frac{v}{\|v\|} = A^{1/2}I_{\hat{n}}^n w + \delta, \quad \|\delta\| \le (2+\sqrt{6})h\lambda_s^{1/2}$$

for some $\hat{n}$-vector w. The condition $y \perp A^{1/2}\mathcal{R}(I_{\hat{n}}^n)$ and $\|y\| = 1$ implies

$$\left| \left\langle y, \frac{v}{\|v\|} \right\rangle \right| = |\langle y, \delta \rangle| \le \|\delta\|.$$

Since $\langle y, v \rangle = \langle v, v \rangle = \|v\|^2$, the desired result (12.34) follows. $\square$

We now use Theorem 12.1.2 and Corollary 12.1.1 to bound the quantity on the right-hand side of (12.14), again assuming that $G = \gamma I$. In this case, inequality (12.14) can be written in the form

$$\sigma^2 \leq \max_{\substack{\|y\|=1 \\ y \perp A^{1/2}\cdot\mathcal{R}(I_{\tilde{n}}^n)}} \left(\sum_{i=1}^{s} \langle y, v^{(i)}\rangle^2 (1-\gamma\lambda_i)^{2\ell} + \sum_{i=s+1}^{n} \langle y, v^{(i)}\rangle^2 (1-\gamma\lambda_i)^{2\ell} \right)$$

Taking $\gamma = 2/(\lambda_n + \lambda_{s+1})$, we can write

$$\sigma^2 \leq \max_{\substack{\|y\|=1 \\ y \perp A^{1/2}\cdot\mathcal{R}(I_{\tilde{n}}^n)}} \left(\sum_{i=1}^{s} \langle y, v^{(i)}\rangle^2 + \left(\frac{\kappa'-1}{\kappa'+1}\right)^{2\ell} \left(1 - \sum_{i=1}^{s} \langle y, v^{(i)}\rangle^2 \right)\right),$$

where $\kappa' = \lambda_n/\lambda_{s+1}$. Applying Corollary 12.1.1 (and using the fact that a function of the form $x + ((\kappa'-1)/(\kappa'+1))^{2\ell}(1-x)$ is an increasing function of x for $0 \leq x \leq 1$), this becomes

$$\sigma^2 \leq c^2h^2\lambda_s + \left(\frac{\kappa'-1}{\kappa'+1}\right)^{2\ell} (1 - c^2h^2\lambda_s), \tag{12.35}$$

provided that $c^2h^2\lambda_s \leq 1$.

Now, from (12.24) we know that λ_n is bounded by αh^{-2}, where $\alpha = 8$. (We are using the symbolic constants c and α instead of their actual values because similar results hold for more general finite element and finite difference equations for some constants c and α that are independent of h but are not necessarily the same ones as for the model problem.) Let $\beta > 0$ be any number less than or equal to α and such that

$$c^2\beta < 1.$$

Choose s so that λ_s is the largest eigenvalue of A less than or equal to βh^{-2}:

$$\lambda_s \leq \beta h^{-2}, \quad \lambda_{s+1} > \beta h^{-2}.$$

Then expression (12.35) becomes

$$\sigma^2 \leq c^2\beta + \left(\frac{\kappa'-1}{\kappa'+1}\right)^{2\ell} (1 - c^2\beta), \quad \kappa' \leq \frac{\alpha}{\beta}.$$

We thus obtain a bound on σ^2 that is *strictly less than one* and *independent of* h. For example, choosing $\beta = 1/(2c^2)$ gives

$$\sigma^2 \leq \frac{1}{2} + \frac{1}{2}\left(\frac{\kappa'-1}{\kappa'+1}\right)^{2\ell}, \quad \kappa' \leq 2\alpha c^2.$$

For the model problem, this establishes $\kappa' \leq 16(2+\sqrt{6})^2$ and, for $\ell = 1$, $\sigma \leq .997$. This is a *large overestimate* of the actual contraction number for the two-grid method, but it does establish convergence at a rate that is independent of h. To obtain a better estimate of σ, it would be necessary to derive a sharper bound on the constant c in Theorem 12.1.2.

12.1.3. Extension to More General Finite Element Equations. The key to the analysis of section 12.1.2 was Theorem 12.1.2, showing that vectors in the span of eigenvectors associated with small eigenvalues of A can be well approximated on a coarser grid. This is true in general for standard finite element matrices and often for finite difference matrices. It is a consequence of the fact that the (functions represented by the) eigenvectors corresponding to smaller eigenvalues on both the fine and coarse grids provide good approximations to eigenfunctions of the elliptic differential operator, and hence they also approximate each other. For the analogues of Theorem 12.1.2 and Corollary 12.1.1 in a more general setting, see [64]. Similar results can be found in [104].

12.1.4. Multigrid Methods. The two-grid algorithm described in sections 12.1.1–12.1.2 is not practical in most cases because it requires solving a linear system on a grid of spacing $2h$. Usually this is still too large a problem to solve directly. The algorithm could be applied recursively, and the solution to the problem on the coarser grid could be obtained by projecting the right-hand side onto a still coarser grid, solving a linear system there, interpolating the solution back to the finer grid, performing relaxation steps there, and repeating this cycle until convergence. If the grid-$2h$ problem is solved very accurately, however, this method would also be time consuming. A number of cycles might be required to solve the problems on coarser grids before ever returning to the fine grid where the solution is actually needed.

Instead, coarser grid problems can be "solved" very inaccurately by performing just one relaxation sweep until the coarsest level is reached, at which point the problem is solved directly. Let grid levels $0, 1, \ldots, J$ be defined with maximum mesh spacings $h_0 \leq h_1 \leq \cdots \leq h_J$, and let $A^{(j)}$ denote the coefficient matrix for the problem at level j. The linear system on the finest grid is $Au = f$, where $A \equiv A^{(0)}$. The multigrid *V-cycle* consists of the following steps.

Given an initial guess u_0, compute $r_0 \equiv r_0^{(0)} = f - Au_0$. For $k = 1, 2, \ldots$,

For $j = 1, \ldots, J-1$,
Project $r_{k-1}^{(j-1)}$ onto grid level j; that is, set

$$f^{(j)} = I_{j-1}^{j} r_{k-1}^{(j-1)},$$

where I_{j-1}^{j} is the restriction matrix from grid level $j-1$ to grid level j.

Perform a relaxation sweep (with zero initial guess) on grid level j; that is, solve

$$G\delta_{k-1}^{(j)} = f^{(j)}$$

and compute

$$r_{k-1}^{(j)} = f^{(j)} - A^{(j)}\delta_{k-1}^{(j)}.$$

endfor

Project $r_{k-1}^{(J-1)}$ onto grid level J by setting $f^{(J)} = I_{J-1}^{J} r_{k-1}^{(J-1)}$ and solve on the coarsest grid $A^{(J)} d_{k-1}^{(J)} = f^{(J)}$.

For $j = J-1, \ldots, 1$,
Interpolate $d_{k-1}^{(j+1)}$ to grid level j and add to $\delta_{k-1}^{(j)}$; that is, replace

$$\delta_{k-1}^{(j)} \leftarrow \delta_{k-1}^{(j)} + I_{j+1}^{j} d_{k-1}^{(j+1)},$$

where I_{j+1}^{j} is the prolongation matrix from grid level $j+1$ to grid level j.
Perform a relaxation sweep with initial guess $\delta_{k-1}^{(j)}$ on grid level j; that is, set

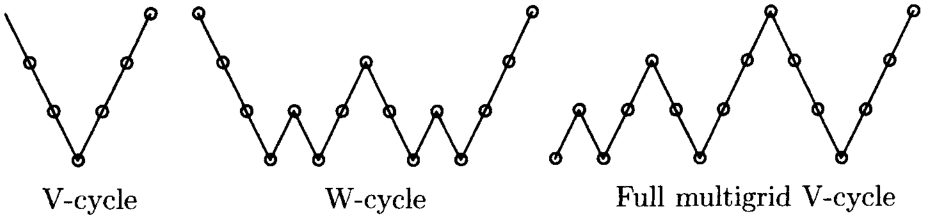
$$d_{k-1}^{(j)} = \delta_{k-1}^{(j)} + G^{-1}(f^{(j)} - A^{(j)}\delta_{k-1}^{(j)}).$$

endfor

Interpolate $d_{k-1}^{(1)}$ to grid level 0 and replace $u_{k-1} \leftarrow u_{k-1} + I_1^0 d_{k-1}^{(1)}$.

Perform a relaxation sweep with initial guess u_{k-1} on grid level 0; that is, set $u_k = u_{k-1} + G^{-1}(f - Au_{k-1})$. Compute the new residual $r_k \equiv r_k^{(0)} = f - Au_k$.

This iteration is called a *V-cycle* because it consists of going down through the grids from fine to coarse, performing a relaxation sweep on each grid, then coming back up from coarse to fine, and again performing a relaxation sweep at each level, as pictured below. (Sometimes an initial relaxation sweep on the fine grid is performed before projecting the residual onto the next coarser grid.) Other patterns of visiting the grids are also possible. In the *W-cycle*, for instance, one uses two V-cycles at each of the coarser levels, resulting in a pattern like that shown below for four grid levels. In the *full multigrid V-cycle*, one starts on the coarsest grid, goes up one level and then back down, up two levels and then back down, etc., until the finest level is reached. This provides the initial guess for the standard V-cycle, which is then performed.



V-cycle W-cycle Full multigrid V-cycle

The restriction and prolongation matrices I_j^{j+1} and I_{j+1}^j, as well as the relaxation scheme with matrix G, can be tuned to the particular problem. For the model problem, the linear interpolation matrix I_{j+1}^j is appropriate, although it is not the only choice, and it is reasonable to define the restriction matrix I_j^{j+1} to be $I_{j+1}^{j^T}$, as in section 12.1.1. The damped Jacobi relaxation scheme described in section 12.1.1 is convenient for analysis, but other relaxation schemes may perform better in practice. The red–black Gauss–Seidel relaxation method is often used. (That is, if nodes are ordered so that the matrix A has the form (9.9), then G is taken to be the lower triangle of A.)

Figure 12.1 shows the convergence of the multigrid V-cycle with red–black Gauss–Seidel relaxation for the model problem for grid sizes $h = 1/64$ and $h = 1/128$. The coarsest grid, on which the problem was solved directly, was of size $h = 1/4$. Also shown in Figure 12.1 is the convergence curve for MICCG(0). The work per iteration (or per cycle for multigrid) for these two algorithms is similar. During a multigrid V-cycle, a (red–black) Gauss–Seidel relaxation step is performed once on the fine grid and twice on each of the coarser grids. Since the number of points on each coarser level grid is about 1/4 that of the finer grid, this is the equivalent of about 1 and 2/3 Gauss–Seidel sweeps on the finest grid. After the fine grid relaxation is complete, a new residual must be computed, requiring an additional matrix–vector multiplication on the fine grid. In the MICCG(0) algorithm, backsolving with the L and L^T factors of the MIC decomposition is twice the work of backsolving with a single lower triangular matrix in the Gauss–Seidel method, but only one matrix–vector multiplication is performed at each step. The CG algorithm also requires some inner products that are not present in the multigrid algorithm, but the multigrid method requires prolongation and restriction operations that roughly balance with the work for the inner products. The exact operation count is implementation dependent, but for the implementation used here (which was designed for a general 5-point matrix, not just the Laplacian), the operation count per cycle/iteration was about $41n$ for multigrid and about $30n$ for MICCG(0).

It is clear from Figure 12.1 that for the model problem, the multigrid method is by far the most efficient of the iterative methods we have discussed. Moreover, the multigrid method demonstrated here is *not* the best. The number of cycles can be reduced even further (from 9 down to about 5 to achieve an error of size 10^{-6}) by using the W-cycle or the full multigrid V-cycle, with more accurate restriction and prolongation operators. The work per cycle is somewhat greater, but the reduction in number of cycles more than makes up for the slight increase in cycle time. (See Exercise 12.2.)

The multigrid method described here works well for a variety of problems, including nonsymmetric differential equations, such as $-\triangle u + cu_x = f$, as well as for the model problem. It should be noted, however, that while the performance of ICCG and MICCG is not greatly changed if the model problem

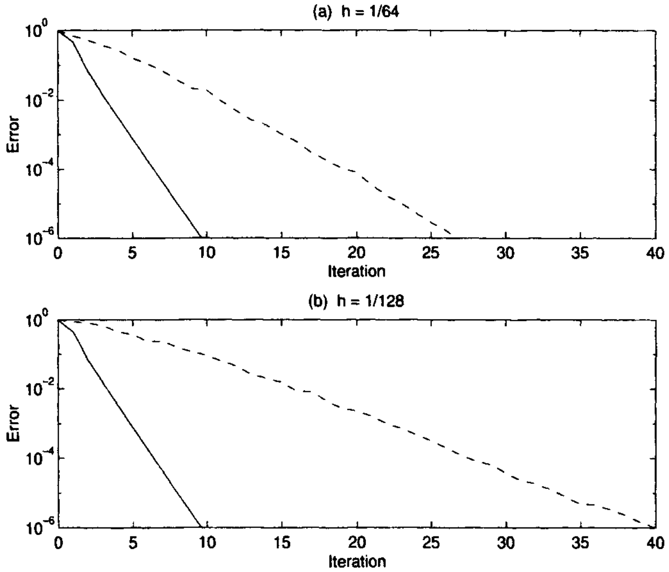


FIG. 12.1. *Convergence of the multigrid V-cycle (solid) and MICCG*(0) *(dashed) for the model problem.*

is replaced by the diffusion equation (9.1) with a highly varying diffusion coefficient, this is not the case for the multigrid method. The multigrid algorithm used for the model problem will still converge at a rate that is independent of h if applied to the diffusion equation, but the convergence rate will be *greatly* affected by the variation in the diffusion coefficient. For problems with discontinuous diffusion coefficients, linear interpolation, as used here, is not really appropriate. It should be replaced by a form of interpolation that takes account of the discontinuities [1].

For this reason, instead of thinking of *the* multigrid method, one should view the multigrid approach as a framework for developing iterative methods (that is, preconditioners to be used with simple iteration or other Krylov subspace methods). Sometimes, based on known properties of the differential equation, one can identify appropriate prolongation, restriction, and relaxation matrices that will result in a multigrid method whose convergence rate is not only independent of h but is much better than that of other methods for realistic mesh sizes. One should look for both a relaxation method that damps high frequencies very rapidly and restriction and prolongation matrices having the property that the low frequency components of the error are greatly reduced when the residual is projected onto a coarser grid, a problem is solved on that grid, and the solution is interpolated to the finer grid and added to the previous approximation. Such multigrid methods have been developed for a wide variety of physical problems. This is not always possible, however. For problems that are barely resolved on the grid of interest, it may be unclear how the problem should even be defined on coarser level grids, and one cannot expect to gain much information from a "solution" on a such a grid.

12.1.5. Multigrid as a Preconditioner for Krylov Subspace Methods.

Some multigrid aficionados will argue that if one has used the proper restriction, prolongation, and relaxation operators, then the multigrid algorithm will require so few cycles (one or two full multigrid V-cycles to reach the level of truncation error) that it is almost pointless to try to accelerate it with CG-like methods. This may be true, but unfortunately such restriction, prolongation, and relaxation schemes are not always known. In such cases, CG, GMRES, QMR, or BiCGSTAB acceleration may help.

Equivalently, one can consider multigrid as a preconditioner for one of these Krylov subspace methods. To solve an equation $Mz = r$ with the multigrid V-cycle preconditioner M as coefficient matrix, one simply performs one multigrid V-cycle with right-hand side r and initial guess zero.

For some interesting examples using multigrid as a preconditioner for GMRES and BiCGSTAB, see [108]. The use of multigrid (with damped Jacobi relaxation) as a preconditioner for the CG algorithm for solving diffusion-like equations is described in [4].

12.2. Basic Ideas of Domain Decomposition Methods.

Simulation problems often involve complicated structures such as airplanes and automobiles. Limitations on computer time and storage may prevent the modeling of the entire structure at once, so instead a piece of the problem is studied, e.g., an airplane wing. If different parts of the problem could be solved independently and then the results somehow pieced together to give the solution to the entire problem, then a loosely coupled array of parallel processors could be used for the task. This is one of the motivations for domain decomposition methods. Even if the domain of the problem is not so complicated, one might be able to break the domain into pieces on which the problem is more easily solved, e.g., rectangles on which a fast Poisson solver could be used or subdomains more suitable for multigrid methods. If the solutions of the subproblems can be combined in a clever way to solve the overall problem, then this may provide a faster and more parallel solution method than applying a standard iterative method directly to the large problem. We will see that this solution approach is equivalent to using a preconditioner that involves solving on subdomains. The clever way of combining the solutions from subdomains is usually a CG-like iterative method.

Domain decomposition methods fall roughly into two classes—those using *overlapping domains*, such as the additive and multiplicative Schwarz methods, and those using *nonoverlapping domains*, which are sometimes called *substructuring* methods. If one takes a more general view of the term "subdomain," then the subdomains need not be contiguous parts of the physical domain at all but may be parts of the solution space, such as components that can be represented on a coarser grid and those that cannot. With this interpretation, multigrid methods fall under the heading of domain decomposition methods.

In this chapter, we describe some basic domain decomposition methods but give little of the convergence theory. For further discussion, see [123] or [77].

12.2.1. Alternating Schwarz Method. Let $\mathcal{L}$ be a differential operator defined on a domain Ω, and suppose we wish to solve the boundary value problem

$$\begin{aligned} \mathcal{L}u = f \quad &\text{in } \Omega, \\ u = g \quad &\text{on } \partial\Omega. \end{aligned} \tag{12.36}$$

The domain Ω is an open set in the plane or in 3-space, and $\partial\Omega$ denotes the boundary of Ω. We denote the closure of Ω by $\bar{\Omega} \equiv \Omega \cup \partial\Omega$. We have chosen Dirichlet boundary conditions ($u = g$ on $\partial\Omega$), but Neumann or Robin boundary conditions could be specified as well.

The domain Ω might be divided into two overlapping pieces, Ω_1 and Ω_2, such that $\Omega = \Omega_1 \cup \Omega_2$, as pictured in Figure 12.2. Let Γ_1 and Γ_2 denote the parts of the boundaries of Ω_1 and Ω_2, respectively, that are not part of the boundary of Ω. To solve this problem, one might guess the solution on Γ_1 and solve the problem

$$\begin{aligned} \mathcal{L}u_1 = f \quad &\text{in } \Omega_1, \\ u_1 = g \quad \text{on } \partial\Omega \cap \partial\Omega_1, \quad &u_1 = g_1 \quad \text{on } \Gamma_1, \end{aligned} \tag{12.37}$$

where g_1 is the initial guess for the solution on Γ_1. Letting g_2 be the value of u_1 on Γ_2, one then solves

$$\begin{aligned} \mathcal{L}u_2 = f \quad &\text{in } \Omega_2, \\ u_2 = g \quad \text{on } \partial\Omega \cap \partial\Omega_2, \quad &u_2 = g_2 \quad \text{on } \Gamma_2. \end{aligned} \tag{12.38}$$

If the computed solutions u_1 and u_2 are the same in the region where they overlap then the solution to problem (12.36) is

$$u = \begin{cases} u_1 & \text{in } \Omega_1 \backslash \Omega_2, \\ u_2 & \text{in } \Omega_2 \backslash \Omega_1, \\ u_1 \equiv u_2 & \text{in } \Omega_1 \cap \Omega_2. \end{cases}$$

If the values of u_1 and u_2 differ in the overlap region, then the process can be repeated, replacing g_1 by the value of u_2 on Γ_1, and re-solving problem (12.37), etc. This idea was introduced by Schwarz in 1870 [120], not as a computational technique, but to establish the existence of solutions to elliptic problems on regions where analytic solutions were not known. When used as a computational technique it is called the *alternating Schwarz method.*

A slight variation of the alternating Schwarz method, known as the *multiplicative Schwarz method*, is more often used in computations. Let the problem (12.36) be discretized using a standard finite difference or finite element method, and assume that the overlap region is sufficiently wide so

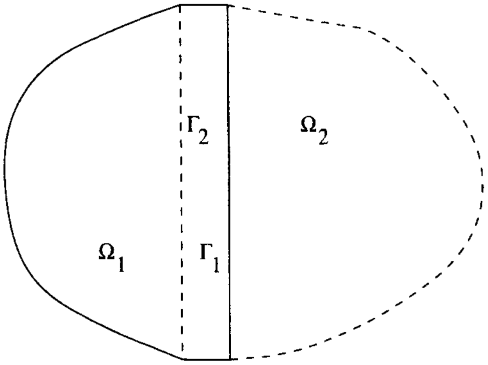


FIG. 12.2. *Decomposition of domain into two overlapping pieces.*

that nodes in $\Omega_1 \backslash \Omega_2$ do not couple to nodes in $\Omega_2 \backslash \Omega_1$, and vice versa. Assume also that the boundaries Γ_1 and Γ_2 are grid lines. If nodes in $\Omega_1 \backslash \Omega_2$ are numbered first, followed by nodes in $\Omega_1 \cap \Omega_2$, and then followed by nodes in $\Omega_2 \backslash \Omega_1$, then the discretized problem can be written in the form

$$\begin{pmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} u_{\Omega_1 \backslash \Omega_2} \\ u_{\Omega_1 \cap \Omega_2} \\ u_{\Omega_2 \backslash \Omega_1} \end{pmatrix} = \begin{pmatrix} f_{\Omega_1 \backslash \Omega_2} \\ f_{\Omega_1 \cap \Omega_2} \\ f_{\Omega_2 \backslash \Omega_1} \end{pmatrix}, \tag{12.39}$$

where the right-hand side vector f includes contributions from the boundary term $u = g$ on $\partial\Omega$.

Starting with an initial guess $u^{(0)}$ (which actually need only be defined on Γ_1 for a standard 5-point discretization or, more generally, on points in $\Omega_2 \backslash \Omega_1$ that couple to points in $\Omega_1 \cap \Omega_2$), the multiplicative Schwarz method for the discretized system generates approximations $u^{(k)}$, $k = 1, 2, \ldots$, satisfying

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} u_{\Omega_1 \backslash \Omega_2} \\ u_{\Omega_1 \cap \Omega_2} \end{pmatrix}^{(k)} = \begin{pmatrix} f_{\Omega_1 \backslash \Omega_2} \\ f_{\Omega_1 \cap \Omega_2} - A_{23} u^{(k-1)}_{\Omega_2 \backslash \Omega_1} \end{pmatrix},$$

$$\begin{pmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} u_{\Omega_1 \cap \Omega_2} \\ u_{\Omega_2 \backslash \Omega_1} \end{pmatrix}^{(k)} = \begin{pmatrix} f_{\Omega_1 \cap \Omega_2} - A_{21} u^{(k)}_{\Omega_1 \backslash \Omega_2} \\ f_{\Omega_2 \backslash \Omega_1} \end{pmatrix}. \tag{12.40}$$

The first equation corresponds to solving the problem on Ω_1, using boundary data obtained from $u^{(k-1)}_{\Omega_2 \backslash \Omega_1}$. The second equation corresponds to solving the problem on Ω_2, using boundary data obtained from $u^{(k)}_{\Omega_1 \backslash \Omega_2}$.

Note that this is somewhat like a *block Gauss–Seidel method* for (12.39), since one solves the first block equation using old data on the right-hand side and the second block equation using updated data on the right-hand side, but in this case the blocks overlap:

$$\left(\begin{array}{ccccc} A_{11} & & A_{12} & | & 0 \\ & - & - & - & - \\ A_{21} & | & A_{22} & | & A_{23} \\ - & - & - & - & \\ 0 & | & A_{32} & & A_{33} \end{array}\right).$$

Let E_i^T, $i = 1, 2$, be the rectangular matrix that takes a vector defined on all of Ω and *restricts* it to Ω_i:

$$E_1^T \begin{pmatrix} v_{\Omega_1 \backslash \Omega_2} \\ v_{\Omega_1 \cap \Omega_2} \\ v_{\Omega_2 \backslash \Omega_1} \end{pmatrix} = \begin{pmatrix} v_{\Omega_1 \backslash \Omega_2} \\ v_{\Omega_1 \cap \Omega_2} \end{pmatrix}, \quad E_2^T \begin{pmatrix} v_{\Omega_1 \backslash \Omega_2} \\ v_{\Omega_1 \cap \Omega_2} \\ v_{\Omega_2 \backslash \Omega_1} \end{pmatrix} = \begin{pmatrix} v_{\Omega_1 \cap \Omega_2} \\ v_{\Omega_2 \backslash \Omega_1} \end{pmatrix}.$$

The matrix E_i takes a vector defined on Ω_i and *extends* it with zeros to the rest of Ω. The matrices on the left in (12.40) are of the form $E_i^T A E_i$, where A is the coefficient matrix in (12.39). Using this notation, iteration (12.40) can be written in the equivalent form

$$E_1^T u^{(k)} \leftarrow E_1^T u^{(k-1)} + (E_1^T A E_1)^{-1} E_1^T (f - Au^{(k-1)}), \quad u^{(k)}_{\Omega_2 \backslash \Omega_1} \leftarrow u^{(k-1)}_{\Omega_2 \backslash \Omega_1},$$

$$E_2^T u^{(k)} \leftarrow E_2^T u^{(k)} + (E_2^T A E_2)^{-1} E_2 (f - Au^{(k)}).$$

Writing this as two half-steps and extending the equations to the entire domain, the iteration becomes

$$u^{(k-1/2)} = u^{(k-1)} + E_1 (E_1^T A E_1)^{-1} E_1^T (f - Au^{(k-1)}),$$

$$u^{(k)} = u^{(k-1/2)} + E_2 (E_2^T A E_2)^{-1} E_2^T (f - Au^{(k-1/2)}).$$

Defining $B_i \equiv E_i (E_i^T A E_i)^{-1} E_i^T$, these two half-steps can be combined to give

$$u^{(k)} = u^{(k-1)} + (B_1 + B_2 - B_2 A B_1)(f - Au^{(k-1)}). \tag{12.41}$$

This is the simple iteration method described in section 2.1 with preconditioner $M^{-1} = B_1 + B_2 - B_2 A B_1$.

One could also consider solving (12.39) using an *overlapping block Jacobi-type method*; that is, using data from the previous iterate in the right-hand sides of both equations in (12.40). This leads to the set of equations

$$u^{(k)}_{\Omega_1} = u^{(k-1)}_{\Omega_1} + (E_1^T A E_1)^{-1} E_1^T (f - Au^{(k-1)}),$$

$$u^{(k)}_{\Omega_2} = u^{(k-1)}_{\Omega_2} + (E_2^T A E_2)^{-1} E_2^T (f - Au^{(k-1)}),$$

where $u_{\Omega_i}^{(k)} \equiv E_i^T u^{(k)}$. The value of $u^{(k)}$ in the overlap region has been set in two different ways by these equations. For the multiplicative Schwarz method, we used the second equation to define the value of $u^{(k)}$ in the overlap region; for this variant it is customary to take $u_{\Omega_1 \cap \Omega_2}^{(k)}$ to be the *sum* of the two values defined by these equations. This leads to the *additive Schwarz method*:

$$(12.42) \qquad u^{(k)} = u^{(k-1)} + (B_1 + B_2)(f - Au^{(k-1)}).$$

In this case, the preconditioner $M^{-1} = B_1 + B_2$ is Hermitian if A is Hermitian, so the simple iteration (12.42) can be replaced by the CG or MINRES algorithm. In fact, some form of acceleration or damping factor must be used with iteration (12.42) to ensure convergence. To solve the preconditioning equation $Mz = r$, one simply solves a problem on each of the two subdomains independently, using boundary data from the previous iterate, and adds the results.

We will not prove any convergence results for the additive and multiplicative Schwarz preconditioners, but note that for the model problem and many other elliptic differential equations using say, GMRES acceleration, these preconditioners have the following properties (see [123]):

- The number of iterations (to reduce the initial residual norm by a fixed factor) is independent of the mesh size, provided that the overlap region is kept fixed. (Remember, however, that with only two subdomains the solution time on each subdomain grows as the mesh size is decreased!)
- The overlap region can be quite small without greatly affecting the convergence rate.
- The number of iterations for the multiplicative variant is about half that for the additive algorithm. (This is similar to the relation between ordinary Gauss–Seidel and Jacobi iterations (10.20, 10.22).)

12.2.2. Many Subdomains and the Use of Coarse Grids. The multiplicative and additive Schwarz methods are easily extended to multiple subdomains. We will concentrate on the additive variant because it provides greater potential for parallelism. If the region Ω is divided into J overlapping subregions $\Omega_1, \ldots, \Omega_J$, then the additive Schwarz preconditioner is

$$(12.43) \qquad M^{-1} \equiv \sum_{i=1}^{J} B_i,$$

where $B_i = E_i(E_i^T A E_i)^{-1} E_i^T$, as in the previous section. To apply M^{-1} to a vector r, one solves a problem on each subdomain with right-hand side $E_i^T r$ and adds the results. These subdomain solves can be carried out in parallel, so it is desirable to have a large number of subdomains.

Consider the Krylov space generated by f and $M^{-1}A$:

$$\text{span}\{f, (M^{-1}A)f, (M^{-1}A)^2 f, \ldots\}.$$

(For a Hermitian positive definite problem we could equally well consider the Krylov space generated by $L^{-H}f$ and $L^{-1}AL^{-H}$, where $M = LL^H$.) Suppose f has nonzero components in only one of the subdomains, say, Ω_1. Since a standard finite difference or finite element matrix A contains only local couplings, the vector Af will be nonzero only in subdomains that overlap with Ω_1 (or, perhaps, subdomains that are separated from Ω_1 by just a few mesh widths). It is only for these subdomains that the right-hand side $E_i^T f$ of the subdomain problem will be nonzero and hence that $M^{-1}Af$ will be nonzero. If this set of subdomains is denoted S_1, then it is only for subdomains that overlap with (or are separated by just a few mesh widths from) subdomains in S_1 that the next Krylov vector $(M^{-1}A)^2 f$ will be nonzero. And so on. The number of Krylov space vectors will have to reach the length of the shortest path from Ω_1 to the most distant subregion, say, Ω_J, before any of the Krylov vectors will have nonzero components in that subregion. Yet the solution $u(\mathbf{x})$ of the differential equation and the vector u satisfying the discretized problem $Au = f$ may well have nonzero (and not particularly small) components in all of the subdomains. Hence any Krylov space method for solving $Au = f$ with a zero initial guess and the additive Schwarz preconditioner will require at least this shortest path length number of iterations to converge (that is, to satisfy a reasonable error tolerance). As the number of subdomains increases, the shortest path between the most distant subregions also increases, so the number of iterations required by, say, the GMRES method with the additive Schwarz preconditioner will also increase. The reason is that there is no mechanism for global communication among the subdomains.

An interesting cure for this problem was proposed by Dryja and Widlund [36]. In addition to the subdomain solves, solve the problem on a coarse grid whose elements are the subregions of the original grid. If this problem is denoted $A_C u_C = f_C$ and if I_C^F denotes an appropriate type of interpolation (say, linear interpolation) from the coarse to the fine grid, then the preconditioner M^{-1} in (12.43) is replaced by

$$M^{-1} = \sum_{i=1}^{J} B_i + I_C^F A_C^{-1} (I_C^F)^T. \tag{12.44}$$

It turns out that this small amount of global communication is sufficient to eliminate the dependence of the number of iterations on the number of subdomains. For this *two-level method*, the number of iterations is independent of both the mesh width h and the subdomain size H, assuming that the size of the overlap region is $O(H)$.

A two-level method such as this, however, must still require more than $O(n)$ work if both the subdomain and coarse grid solvers require more than $O(p)$ work, where p is the number of points in the subdomain or coarse grid. With just a few large subdomains, the subdomain solves will be too expensive and with many small subdomains, the coarse grid solve will be the large subproblems recursively by another application of the two-level preconditioner. The multilevel domain decomposition methods bear much resemblance to (and are sometimes identical with) standard multigrid methods described in section 12.1.

12.2.3. Nonoverlapping Subdomains. Many finite element codes use a decomposition of the domain into nonoverlapping subregions to define an ordering of the unknowns for use with Gaussian elimination. If the domain Ω is divided into two pieces, Ω_1 and Ω_2, with interface Γ, and if points in the interior of Ω_1 (that do not couple to points in Ω_2) are numbered first, followed by points in the interior of Ω_2 (that do not couple to points in Ω_1) and then by points on the interface Γ, then the linear system takes the form

$$\begin{pmatrix} A_{11} & 0 & A_{1,\Gamma} \\ 0 & A_{22} & A_{2,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,2} & A_{\Gamma,\Gamma} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_\Gamma \end{pmatrix}. \tag{12.45}$$

If the matrices A_{11} and A_{22} can be inverted easily, then the variables u_1 and u_2 can be eliminated using Gaussian elimination and a much smaller *Schur complement* problem solved on the interface Γ:

$$\begin{aligned} &(A_{\Gamma,\Gamma} - A_{\Gamma,1}A_{11}^{-1}A_{1,\Gamma} - A_{\Gamma,2}A_{22}^{-1}A_{2,\Gamma})u_\Gamma = \\ &f_\Gamma - A_{\Gamma,1}A_{11}^{-1}f_1 - A_{\Gamma,2}A_{22}^{-1}f_2. \end{aligned} \tag{12.46}$$

Once u_Γ is known, it can be substituted into (12.45) to obtain u_1 and u_2.

The coefficient matrix in (12.46) is small but dense and very expensive to form. It can be applied to a vector, however, by performing a few sparse matrix vector multiplications and solving on the subdomains. Hence an iterative method might be applied to (12.46). This is the idea of *iterative substructuring* methods. It is an idea that we have already seen in the solution of the transport equation in section 9.2. The source iteration described there can be thought of as an angular domain decomposition method using iterative substructuring (although the source iteration method was actually developed before these terms came into widespread use).

As we saw in section 9.2, the simple iterative substructuring method for that problem was equivalent to a block Gauss–Seidel iteration for the original linear system. While the transport equation is not elliptic, if the same idea were applied to a linear system arising from an elliptic differential equation, the convergence rate would not be independent of the mesh size. To obtain a convergence rate that is independent of, or only weakly dependent on, the mesh size, a preconditioner is needed for the system (12.46). Since the actual matrix in (12.46) is never formed, the standard Jacobi, Gauss–Seidel, SOR, and incomplete Cholesky-type preconditioners cannot be used.

Many preconditioners for (12.46) have been proposed, and a survey of these preconditioners is beyond the scope of this book. For a discussion of several such *interface preconditioners*, see [123].

Comments and Additional References.

Multigrid methods were first introduced by Fedorenko [49], and an early analysis was given by Bakhvalov [9]. A seminal paper by Brandt demonstrated the effectiveness of these techniques for a variety of problems [19]. An excellent introduction to multigrid methods, without much formal analysis, is given in a short book by Briggs [21], and additional information can be found in [18, 77, 78, 84, 98, 141, 146].

Exercises.

12.1. Show that the vectors $z^{(i,j)}$, $i, j = 1, \ldots, m$ with components

$$z_{p,q}^{(i,j)} = 2h\cos(ihp\pi)\cos(jhq\pi), \quad p, q = 1, \ldots, m,$$

defined on an m-by-m grid with spacing $h = 1/(m+1)$, form an orthonormal set. Show that if $\hat{z}_{p,q}^{(i,j)}$ is equal to $z_{p,q}^{(i,j)}$ if p and q are both odd but 0 otherwise, then

$$\|\hat{z}^{(i,j)}\| \leq \frac{1}{2} \quad \text{and} \quad \langle \hat{z}^{(i,j)}, v^{(i,j)} \rangle = 0,$$

where $v^{(i,j)}$ is defined in (12.22).

12.2. A *work unit* is often defined as the number of operations needed to perform one Gauss–Seidel sweep on the finest grid in a multigrid method. For a two-dimensional problem, approximately how many work units are required by (a) a multigrid V-cycle with a presmoothing step (that is, with a relaxation sweep performed on the fine grid at the beginning of the cycle as well as at the end), (b) a multigrid W-cycle, and (c) a full multigrid V-cycle? (You can ignore the work for restrictions and prolongations.) How do your answers change for a three-dimensional problem, assuming that each coarser grid still has mesh spacing equal to twice that of the next finer grid?

12.3. What is the preconditioner M in the iteration (12.5) if $\ell = 1$? Compute the two-grid preconditioner M described in section 12.1.2 for a small model problem. Is it a regular splitting? Are the entries of M close to those of A? (It is unlikely that one would choose this matrix as a preconditioner for A if one looked only at the entries of A and did not consider the origin of the problem!)

12.4. In the two-level additive Schwarz method, suppose that each subdomain and coarse grid solve requires $O(p^{3/2})$ work, where p is the number of points in the subdomain or coarse grid. (This is the cost of backsolving with a banded triangular factor for a matrix of order p with bandwidth $p^{1/2}$.) Assuming that the number of points on the coarse grid is approximately equal to the number of subdomains, what is the optimal number of subdomains needed to minimize the total work in applying the preconditioner (12.44), and how much total work is required?