Chapter 7

Selected Applications

Our model problem for highlighting basic multigrid concepts has been Poisson's equation in one and two dimensions. To see how multigrid can be applied to other problems, we now consider three variations on our model problem, each, we hope, of fairly general interest: Neumann boundary conditions, anisotropic equations, and variable-coefficient/mesh problems. In each case, we present a typical, but simplified, problem that we treat with a specially tailored multigrid technique. The method we propose is not necessarily the most robust, and we ignore complicating issues for the sake of clarity. The purpose here is to illustrate the application of multigrid to new problems and to gain an entry-level understanding of the broad scope of multigrid techniques.

Neumann Boundary Conditions

Dirichlet boundary conditions specify values of the unknown function on the boundary of the domain. In most cases, these so-called *essential boundary conditions* are easily treated in the discretization and in the multigrid solver by simply eliminating the corresponding unknowns in the equations. On the other hand, Neumann conditions involve *derivatives* of the unknown function. This relatively small change in the boundary conditions can be treated in a variety of ways; it also introduces a few important subtleties.

Consider the following two-point boundary value problem with homogeneous Neumann conditions at both end points (the inhomogeneous case is left for Exercise 8):

$$\begin{aligned} -u''(x) &= f(x), \quad 0 < x < 1\,, \\ u'(0) = u'(1) &= 0. \end{aligned} \tag{7.1}$$

With n an odd positive integer, denote the points of a uniform grid on the interval $[0,1]$ by $x_j = jh$, for $0 \le j \le n+1$, where $h = \frac{1}{n+1}$. A common way to discretize this problem involves the *ghost points* $x_{-1} = -\frac{1}{h}$ and $x_{n+2} = 1 + \frac{1}{h}$. These points are outside the problem domain and are used only temporarily to produce approximations at the boundary points.

With this extended grid, we can use central differences for the differential equation at the interior points *and* boundary points. We also use the ghost points to

form central difference approximations for the boundary conditions. These choices lead to the following set of discrete equations:

$$\begin{aligned} \frac{-u^h_{j-1} + 2u^h_j - u^h_{j+1}}{h^2} &= f^h_j, \qquad 0 \le j \le n+1\,, \\ \frac{u^h_1 - u^h_{-1}}{2h} &= 0\,, \\ \frac{u^h_{n+2} - u^h_n}{2h} &= 0\,. \end{aligned}$$

We could now choose to work with this entire system of equations, which requires careful treatment of the boundary equations in relaxation and coarsening. Instead, we eliminate the discrete boundary conditions. The two boundary condition equations imply that

$$u^h_{-1} = u^h_1 \quad \text{and} \quad u^h_{n+2} = u^h_n.$$

Eliminating the ghost points in the $j = 0$ and $j = n + 1$ equations yields the following $(n + 2) \times (n + 2)$ system of algebraic equations:

$$\begin{aligned} \frac{-u^h_{j-1} + 2u^h_j - u^h_{j+1}}{h^2} &= f^h_j, \qquad 1 \le j \le n\,, \\ \frac{2u^h_0 - 2u^h_1}{h^2} &= f^h_0\,, \\ \frac{-2u^h_n + 2u^h_{n+1}}{h^2} &= f^h_{n+1}\,. \end{aligned}$$

As before, we write this system in the matrix form

$$A^h \mathbf{u}^h = \mathbf{f}^h, \tag{7.2}$$

where now

$$A^h = \frac{1}{h^2} \begin{bmatrix} 2 & -2 & & & & & \\ -1 & 2 & -1 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & -1 & 2 & -1 & \\ & & & & -2 & 2 & \end{bmatrix}. \tag{7.3}$$

Note that system (7.2) involves the boundary unknowns u^h_0 and u^h_{n+1} and that A^h is an $(n + 2) \times (n + 2)$ nonsymmetric matrix.

The first observation to be made is that two-point boundary value problem (7.1) is not yet well-posed. First, if this problem has a solution u, it cannot be unique: because the system involves only derivatives of u, the function $u + c$ must also be a solution for any constant c (Exercise 1). Second and worse yet, we cannot even be sure that boundary value problem (7.1) has a solution. If a solution of (7.1) exists, then the source term must satisfy

$$\int_0^1 f(x)\,dx = 0 \tag{7.4}$$

(Exercise 2). This integral *compatibility condition* is necessary for a solution to exist: if f does not satisfy it, then there can be no solution. Fortunately, the compatibility

condition is also sufficient in general. Showing this fact is a little beyond the scope of this book. It amounts to proving that $-\frac{\partial^2}{\partial x^2}$ is a well-behaved operator on the appropriate space of functions u that have zero mean: $\int_0^1 u(x)\,dx = 0$.

This reasoning allows us to conclude that if f satisfies compatibility condition (7.4), then the following boundary value problem is well-posed:

$$\begin{aligned} -u''(x) &= f(x), \quad 0 < x < 1\,, \\ u'(0) = u'(1) &= 0, \\ \int_0^1 u(x)\,dx &= 0\,. \end{aligned} \tag{7.5}$$

The last condition says that of all the solutions $u + c$ that satisfy the differential equation and boundary conditions, we choose the one with zero mean.

Note that if u solves (7.5), then $u + c$ solves the same boundary value problem with $\int_0^1 u(x)\,dx = c$. Thus, the solution of (7.2) with mean value c is easily found from the zero-mean solution.

In a similar way, discrete system (7.2) is not well-posed. Let $\mathbf{1}^h$ be the $(n+2)$-vector whose entries are all 1, so the entries of $c\mathbf{1}^h$ are all c. The matrix A^h has a nontrivial null space consisting of the constant vectors $c\mathbf{1}^h$ (Exercise 3). Thus, if $\mathbf{u}^h$ is a specific solution of (7.2), then the general solution is $\mathbf{u}^h + c\mathbf{1}^h$ (Exercise 4).

By the Fundamental Theorem of Linear Algebra, (7.2) is solvable only if $\mathbf{f}^h$ is orthogonal to $\mathcal{N}(A^T)$. It is not difficult to show that the $(n+2)$-vector $(\frac{1}{2}, 1, \ldots, 1, \frac{1}{2})$ forms a basis for $\mathcal{N}(A^T)$. Thus, the system is solvable only if $\mathbf{f}$ is orthogonal to $\mathcal{N}((A^h)^T)$, which implies that

$$\frac{1}{2}f_0^h + \sum_{j=0}^{n} f_j^h + \frac{1}{2}f_{n+1}^h = 0.$$

As in the continuous problem, we have two issues to handle: solvability, which requires that $\mathbf{f}^h$ be orthogonal to the null space of $(A^h)^T$, and nonuniqueness, which means that two solutions differ by a vector in the null space of A^h. A useful observation is that if A is symmetric ($A = A^T$), then solvability and nonuniqueness can be dealt with together, because the null spaces of A and A^T are identical in that case.

Fortunately, there is an extremely simple way to symmetrize this problem: we just divide the first and last equations of the system by 2. This yields a symmetric system whose matrix is

$$\hat{A}^h = \frac{1}{h^2}\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}. \tag{7.6}$$

$\hat{A}^h$ *and* its transpose have as a null space multiples of the constant vector $\mathbf{1}^h$. The right-side vector becomes $\hat{\mathbf{f}}^h = [f_0/2, f_1, \ldots, f_n, f_{n+1}/2]^T$. Solvability is then guaranteed by ensuring that $\hat{\mathbf{f}}^h$ is orthogonal to the constant vector $\mathbf{1}^h$:

$$\langle \hat{\mathbf{f}}^h, \mathbf{1}^h \rangle = \sum_{j=0}^{n+1} \hat{f}_j^h = 0 \tag{7.7}$$

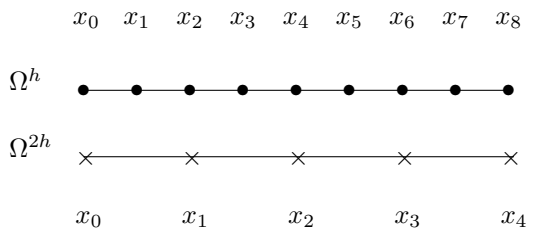

Figure 7.1: *For the Neumann boundary condition case, the solution is unknown at the end points x_0 and x_{n+1}, where n is an odd integer. The figure shows Ω^h and Ω^{2h} for the case $n = 7$.*

(Exercise 5). This is the discrete analogue of integral compatibility condition (7.4). Hence, the discrete analogue of the well-posed differential system (7.5) is

$$\begin{aligned} \frac{-u^h_{j-1} + 2u^h_j - u^h_{j+1}}{h^2} &= f^h_j, \qquad 1 \le j \le n\,, \\ \frac{u^h_0 - u^h_1}{h^2} &= \frac{f^h_0}{2}\,, \\ \frac{-u^h_n + u^h_{n+1}}{h^2} &= \frac{f^h_{n+1}}{2}\,, \\ \sum_{i=0}^{n+1} u^h_i &= 0\,, \end{aligned} \tag{7.8}$$

or simply

$$\hat{A}^h \mathbf{u}^h = \hat{\mathbf{f}}^h\,, \tag{7.9}$$

$$\langle \mathbf{u}^h, \mathbf{1}^h \rangle = 0\,. \tag{7.10}$$

As before, of all solutions to the problem, that last condition selects the solution with zero mean. Matrix system (7.9)–(7.10) is well-posed in the sense that it has a unique solution, provided $\hat{\mathbf{f}}^h$ satisfies (7.7).

The ingredients of our multigrid algorithm for solving (7.9)–(7.10) are basically the same as before, except we now must account for the extra equations associated with the unknowns at the boundary points and the zero-mean condition. Figure 7.1 shows a fine grid, Ω^h, with $n + 2 = 9$ points and the corresponding coarse grid Ω^{2h} with $\frac{n+3}{2} = 5$ points.

The approximation $\mathbf{v}^h$ must include the entries v^h_0 and v^h_{n+1}, and the coarse-grid correction $\mathbf{v}^{2h}$ must include v^{2h}_0 and $v^{2h}_{(n+1)/2}$. Relaxation is now performed at *all* the fine-grid points, including x_0 and x_{n+1}. Relaxation at the end points is based on the second and third equations of (7.8):

$$\begin{aligned} v^h_0 &\leftarrow v^h_1 + h^2 \hat{f}^h_0, \\ v^h_{n+1} &\leftarrow v^h_n + h^2 \hat{f}^h_{n+1}. \end{aligned}$$

To account for zero-mean condition (7.10), we now just add the step

$$\mathbf{v}^h \leftarrow \mathbf{v}^h - \frac{\langle \mathbf{v}^h, \mathbf{1}^h \rangle}{\langle \mathbf{1}^h, \mathbf{1}^h \rangle} \mathbf{1}^h, \tag{7.11}$$

which is just the Gram–Schmidt method applied to orthogonality condition (7.10). (Note that $\langle \mathbf{1}^h, \mathbf{1}^h \rangle = n + 2$.) This global step can be applied before or after relaxation on each level. In principle, this step can wait until the very end of the multigrid algorithm since its primary role is to produce an approximation with the correct average. From a mathematical point of view, it does not really matter if the intermediate approximation $\mathbf{v}^h$ has a nonzero mean. However, from a numerical point of view, it is probably best to apply (7.11) often enough to prevent a large constant from creeping in and swamping accuracy.

The coarsening process is similar to that for the Dirichlet case, but now we must remember to interpolate corrections to the end point approximations. These end points are present on both fine and coarse grids, so the correction process is straightforward. For example, recalling that the terms v_j^{2h} are errors computed on the coarse grid, the correction steps are given by

$$\begin{aligned} v_0^h &\leftarrow v_0^h + v_0^{2h}\,, \\ v_1^h &\leftarrow v_1^h + \frac{v_0^{2h} + v_1^{2h}}{2}\,. \end{aligned} \tag{7.12}$$

What about restriction of residuals to the coarse grid? In the Dirichlet case, we chose I_h^{2h} to be the transpose of I_{2h}^h scaled by $\frac{1}{2}$:

$$I_h^{2h} = \frac{1}{2}\left(I_{2h}^h\right)^T\,.$$

We will use the same restriction operator here, but we must determine its form at the boundary points. Correction step (7.12) shows that interpolation relates v_0^{2h} to v_0^h using the weight 1 and relates v_0^{2h} to v_1^h using the weight $\frac{1}{2}$. Thus, reversing these relationships (which is what the transpose does) and scaling them by $\frac{1}{2}$, we arrive at the restriction step (Exercise 6)

$$\hat{f}_0^{2h} \leftarrow \frac{1}{2}\hat{f}_0^h + \frac{1}{4}\hat{f}_1^h\,.$$

The coarse-grid matrix we get from the Galerkin condition, $\hat{A}^{2h} = I_h^{2h}\hat{A}^h I_{2h}^h$, is just the coarse-grid version of $\hat{A}^h$ defined in (7.3): this follows from Table 5.1 for the interior points, and from Table 7.1 for the boundary point $x = 0$ ($x = 1$ is analogous). One consequence of this relationship is that the multigrid scheme preserves the variational properties we used in the Dirichlet case.

Another subtlety that we need to address is the discrete compatibility condition. If we are given a general source vector $\hat{\mathbf{f}}^h$, we can be sure that the fine-grid problem has a solution simply by testing the zero-mean condition (7.7). But what do we do if this test fails? We could simply stop because this problem has no solution. However, we might find it advantageous to solve the nearby problem created by making the replacement

$$\hat{\mathbf{f}}^h \leftarrow \hat{\mathbf{f}}^h - \frac{\langle \hat{\mathbf{f}}^h, \mathbf{1}^h \rangle}{\langle \mathbf{1}^h, \mathbf{1}^h \rangle}\mathbf{1}^h\,. \tag{7.13}$$

But what about the coarse-grid problem? The answer is that the coarse-grid equation will be solvable, at least in theory, whenever the fine-grid equation is solvable (Exercise 7). However, to be sure that numerical round-off does not perturb the

	0		1
$\hat{\mathbf{e}}_0^{2h}$	1		0
$I_{2h}^h \hat{\mathbf{e}}_0^{2h}$	1	$\frac{1}{2}$	0
$\hat{A}^h I_{2h}^h \hat{\mathbf{e}}_0^{2h}$	$\frac{1}{2h^2}$	0	$-\frac{1}{2h^2}$
$I_h^{2h} \hat{A}^h I_{2h}^h \hat{\mathbf{e}}_0^{2h}$	$\frac{1}{4h^2}$		$-\frac{1}{4h^2}$

Table 7.1: *Calculation of the first row of* $\hat{A}^{2h} = I_h^{2h} \hat{A}^h I_{2h}^h$ *at the boundary* $x = 0$.

solvability too much, it is probably best to incorporate a Gram–Schmidt step analogous to (7.13) on the coarse grids:

$$\hat{\mathbf{f}}^{2h} \leftarrow \hat{\mathbf{f}}^{2h} - \frac{\langle \hat{\mathbf{f}}^{2h}, \mathbf{1}^{2h} \rangle}{\langle \mathbf{1}^{2h}, \mathbf{1}^{2h} \rangle} \mathbf{1}^{2h} .$$

Note that $\langle \mathbf{1}^{2h}, \mathbf{1}^{2h} \rangle = \frac{n+3}{2}$.

Numerical example. Consider the two-point boundary value problem

$$\begin{aligned} -u''(x) &= 2x - 1, \qquad 0 < x < 1, \\ u'(0) = u'(1) &= 0. \end{aligned}$$

By integrating the differential equation twice and applying the boundary conditions, we find that the function $u(x) = \frac{x^2}{2} - \frac{x^3}{3} + c$ solves the problem for any constant c. The zero-mean solution corresponds to $c = -\frac{1}{12}$, and we use this function as our exact solution in the numerical experiments. Solutions are approximated on a succession of grids to illustrate the effectiveness of the algorithm as n increases.

Symmetrization of the problem produces the source vector $\hat{\mathbf{f}}$, where $\hat{f}_i = f_i$ for $1 \le i \le n$,

$$\hat{f}_0 = \frac{f_0}{2} = -\frac{1}{2}, \quad \text{and} \quad \hat{f}_{n+1} = \frac{f_{n+1}}{2} = \frac{1}{2}.$$

It can be checked that $\hat{\mathbf{f}}$ satisfies discrete compatibility condition (7.7). The restriction, interpolation, and coarse-grid operators described above are used with V(2,1)-cycles based on Gauss–Seidel relaxation. For this problem, which is computed in double precision, we observed that it was enough to apply the Gram–Schmidt process once at the end of each V-cycle.

Table 7.2 shows the results of this experiment. Each test is stopped when the discrete L^2 norm of the residual drops below 10^{-10}. Listed are the grid sizes, the final residual norm, the average convergence factor, the final error norm (the difference between the computed solution and the sampled version of the continuous solution), and the number of V-cycles required to reach the desired tolerance.

Grid size n	$\|r^h\|_h$	Average conv. factor	$\|e\|_h$	Number of cycles
31	6.3e−11	0.079	9.7e−05	9
63	1.9e−11	0.089	2.4e−05	10
127	2.6e−11	0.093	5.9e−06	10
255	3.7e−11	0.096	1.5e−06	10
511	5.7e−11	0.100	3.7e−07	10
1027	8.6e−11	0.104	9.2e−08	10
2047	2.1e−11	0.112	2.3e−08	10
4095	5.2e−11	0.122	5.7e−09	11

Table 7.2: *Numerical results for* $-u''(x) = f(x), u'(0) = u'(1) = 0$. *Shown are discrete* L^2 *norms of the final residual and the final error, average convergence factors, and number of* V(2,1)*-cycles required to reduce the residual norm to less than* 10^{-10}.

It is evident from the table that the method is very robust on this problem in that the speed of convergence is essentially independent of problem size. Furthermore, with the strict tolerance on the residual norm used, the discrete problems are solved well below the level of discretization error. The final errors represent the discretization errors themselves: they decrease by almost exactly one-fourth with each doubling of n, which is evidence of $O(h^2)$ behavior.

An interesting subtlety of this problem is that we could have chosen to use *one-sided differences* for approximating the boundary derivatives:

$$u'(0) \approx \frac{u_1 - u_0}{h} \quad \text{and} \quad u'(1) \approx \frac{u_{n+2} - u_{n+1}}{h}.$$

This choice would have resulted in exactly the same operator matrix, $\hat{A}^h$, that we obtained by symmetrizing the discretization based on central differences. However, the source vector would be the original vector, $\mathbf{f}$, rather than the vector $\hat{\mathbf{f}}$. Using this system, and leaving the remainder of the algorithm unchanged, we found experimentally that the speed of convergence and residual errors were approximately as we see in Table 7.2, as expected. However, the final errors (that is, the discretization errors) were much larger and did not decrease nearly as fast with increasing n, because one-sided differences are not as accurate as central difference ($(O(h)$ compared to $O(h^2)$). ◇◇

Anisotropic Problems

The problems treated thus far have been limited to matrices with constant nonzero off-diagonal entries, namely, $-\frac{1}{h^2}$. Such matrices naturally arise in treating Poisson's equation on uniform grids. Our first departure from this situation is to consider two-dimensional problems in which the matrices have two different constants appearing in the off-diagonal terms. Such matrices arise when

- the differential equation has constant, but different, coefficients for the derivatives in the coordinate directions, or

- when the discretization has constant but different mesh sizes in each coordinate direction.

The model problems for these two cases are, respectively, the differential equation

$$-u_{xx} - \epsilon u_{yy} = f \tag{7.14}$$

discretized on a uniform grid of mesh size h and Poisson's equation ($\epsilon = 1$) discretized on a grid with constant mesh size $h_x = h$ in the x-direction and constant mesh size $h_y = \frac{h}{\sqrt{\epsilon}}$ in the y-direction. To be very distinct from the isotropic case $\epsilon = 1$, we assume that $0 < \epsilon << 1$.

It is interesting that these two *anisotropic* model problems lead to the same five-point stencil (Exercise 9):

$$A^h = \frac{1}{h^2} \begin{pmatrix} & -\epsilon & \\ -1 & 2+2\epsilon & -1 \\ & -\epsilon & \end{pmatrix}. \tag{7.15}$$

This relationship between problems with variable coefficients and problems with variable mesh sizes is important; it means that we can think of either example as we develop a method to treat anisotropic problems. Note that the *weak connection* in the vertical direction in these examples arises, respectively, from a small coefficient of the y-derivative term or a large grid spacing in the y-direction.

This departure from the model (isotropic) problem unfortunately leads to trouble with the standard multigrid approach: multigrid convergence factors degrade as ϵ tends to zero, and even for $\epsilon \approx 0.1$, poor performance can be expected.

To understand why multigrid deteriorates, consider (7.14) in the limiting case $\epsilon = 0$. The matrix then becomes

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

which means that the discrete problem becomes a collection of one-dimensional Poisson equations in the x-direction, with no connections in the y-direction. Gauss–Seidel or damped Jacobi point relaxation will smooth in the x-direction because it does so in one dimension. However, the lack of connections in the y-direction means that errors on one x-line have nothing to do with errors on any other: errors in the y-direction will generally have a random pattern, far from the smoothness needed for coarsening to work well.

We can gain further insight into the problems that arise with anisotropy if we look at the eigenvalues of the iteration matrix. Recalling Exercise 12 of Chapter 4, if the weighted Jacobi method with parameter ω is applied to the model Poisson equation in two dimensions on an $n \times n$ grid, the eigenvalues of the iteration matrix are given by

$$\lambda_{k,\ell} = 1 - \omega \left(\sin^2 \left(\frac{k\pi}{2n} \right) + \sin^2 \left(\frac{\ell\pi}{2n} \right) \right), \quad 1 \le k, \ell \le n.$$

The wavenumbers (frequencies) k and ℓ correspond to the x- and y-directions, respectively. The same analysis (Exercise 10) applied to the problem (7.14) reveals

that the eigenvalues of the Jacobi iteration matrix are

$$\lambda_{k,\ell} = 1 - \frac{2\omega}{1+\epsilon}\left(\sin^2\left(\frac{k\pi}{2n}\right) + \epsilon \sin^2\left(\frac{\ell\pi}{2n}\right)\right), \quad 1 \le k, \ell \le n.$$

Notice that for small ϵ, the contributions from the wavenumbers in the y-direction (the direction of weak coupling) are insignificant. Thus, there is little variation in the eigenvalues with respect to the y-wavenumbers. The variation in eigenvalues with respect to the x-wavenumbers is what we expect in the one-dimensional problem.

The eigenvalue picture is best given in the two plots of Fig. 7.2, with $\epsilon = 0.05, n = 16$, and $\omega = \frac{2}{3}$. The upper plot shows the variation in eigenvalues with respect to ℓ (the y-wavenumber) on lines of constant k; notice that there is little variation in the eigenvalues along a single line in this direction. The lower plot shows the variation in eigenvalues with respect to k (the x-wavenumber) on lines of constant ℓ. This time we see individual curves that look much like the eigenvalue curves for the one-dimensional problem. In addition, the curves are tightly bunched, meaning that the convergence is much the same along any horizontal line of the grid. It should be mentioned that local mode analysis, as discussed in Chapter 4, could have been used to reach many of these same conclusions.

These observations suggest two possible strategies for dealing with the anisotropy:

- Because we expect typical multigrid convergence for the one-dimensional problems along x-lines, we should "do multigrid" and coarsen the grid in the x-direction, but not in the y-direction.
- Because the equations are strongly coupled in the x-direction, it will prove advantageous to solve for entire lines of unknowns in the x-direction all at once; this is the goal of *line* or *block relaxation*.

We now investigate these two strategies in more detail.

Semicoarsening/point relaxation

Because relaxation smooths in the x-direction, it makes sense to coarsen horizontally by eliminating every other vertical (y-) line. However, because point relaxation does not smooth in the y-direction, we should *not* coarsen vertically; all horizontal lines should be retained on the coarse grid. This semicoarsening process is depicted in Fig. 7.3. This means that when we write Ω^{2h}, we really mean the coarse grid that has the original grid spacing in the y-direction and twice the original grid spacing in the x-direction.

Interpolation can be done in a one-dimensional way along each horizontal line, giving coarse-grid correction equations of the form

$$v^h_{2i,j} \leftarrow v^h_{2i,j} + v^{2h}_{i,j}, \quad v^h_{2i+1,j} \leftarrow v^h_{2i+1,j} + \frac{v^{2h}_{i,j} + v^{2h}_{i+1,j}}{2}.$$

Semi-coarsening is not as "fast" as *full* coarsening: going from the fine to the coarse grid, the number of points is reduced by a factor of about two with semicoarsening, as opposed to the usual factor of about four. This means that W-cycles lose $O(n)$ complexity; however, V-cycle complexity remains $O(n)$ (Exercise 12).

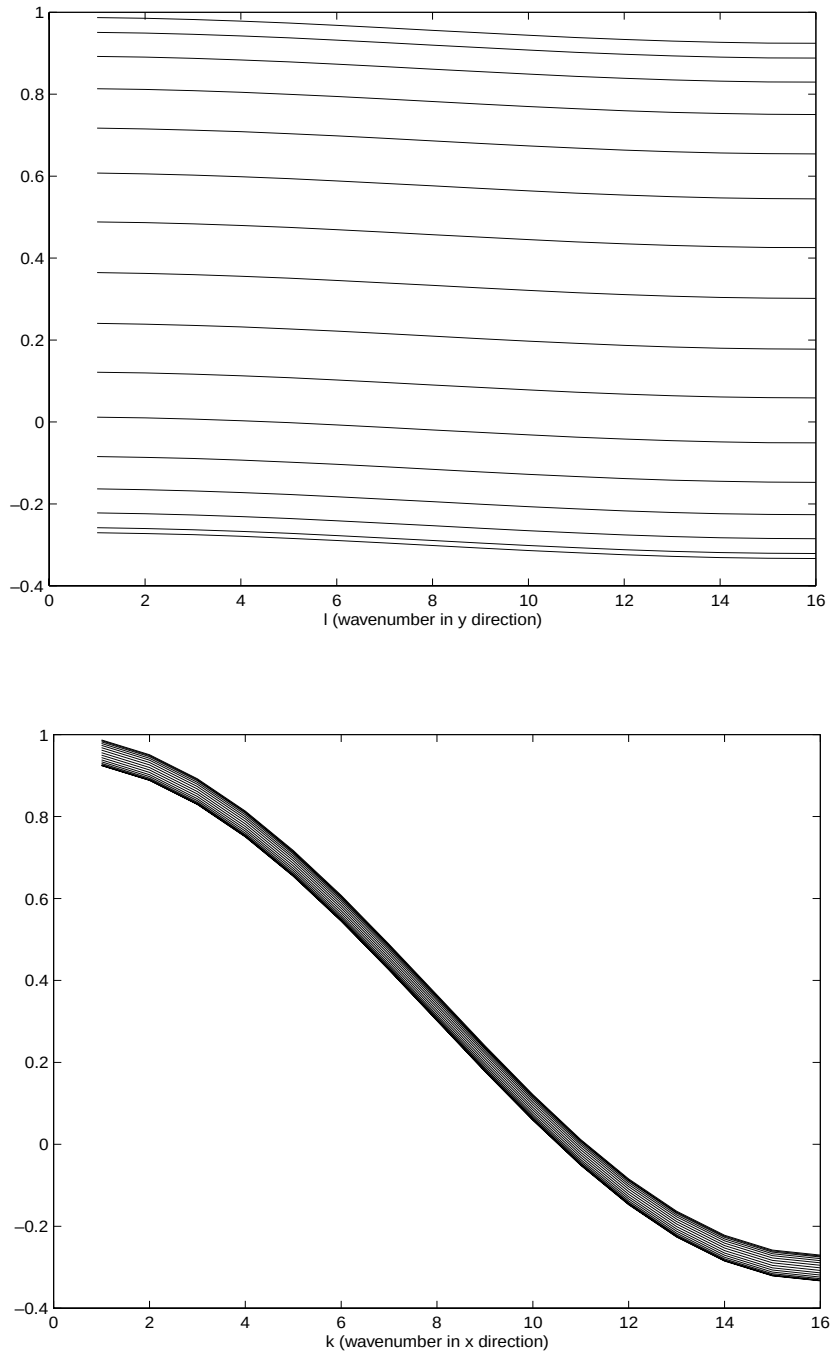

Figure 7.2: *The upper plot shows the eigenvalues of the Jacobi iteration matrix, with* $\omega = \frac{2}{3}$*, along lines of constant* k *(wavenumber in the* x*-direction). The upper curve corresponds to* $k = 1$ *and the lower curve to* $k = n = 16$*. The lower plot shows the same eigenvalues along lines of constant* ℓ *(wavenumber in the* y*-direction). The small parameter in* (7.14) *is* $\epsilon = 0.05$*.*

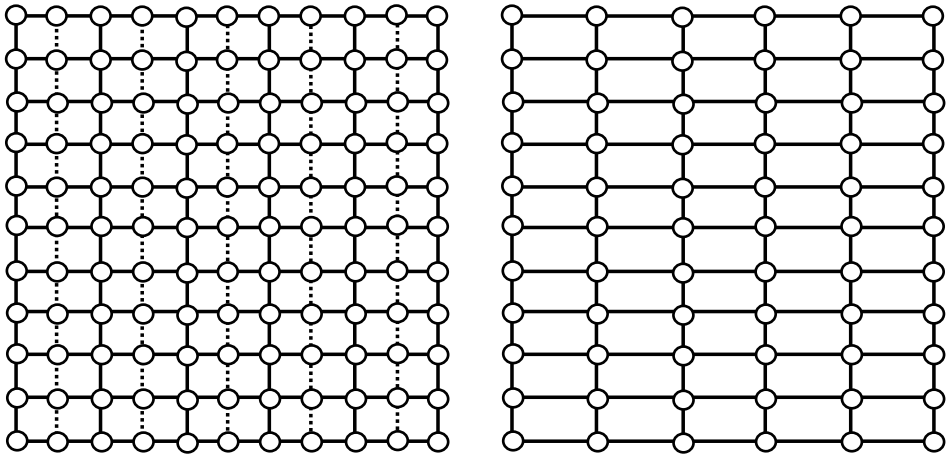

Figure 7.3: *The original fine grid (left, solid and dashed lines) is semicoarsened in the x-direction by deleting the dashed lines (right).*

Line relaxation/full coarsening

Line (or block) relaxation can be developed by writing a system of linear equations in block form. In the case of (7.14), if we order the unknowns along lines of constant y (because strong coupling is in the x-direction), the matrix A^h can be written in block form as

$$A^h = \begin{bmatrix} D & -cI & & & \\ -cI & D & -cI & & \\ \cdot & -cI & D & -cI & \\ & \cdot & \cdot & \cdot & -cI \\ & & & -cI & D \end{bmatrix}, \tag{7.16}$$

where $c = \frac{\epsilon}{h^2}$ and I is the identity matrix. For this model problem, the diagonal blocks D are tridiagonal and identical. Each block is associated with an individual horizontal grid line and has the stencil $\frac{1}{h^2}\begin{pmatrix} -1 & 2+2\epsilon & -1 \end{pmatrix}$.

One sweep of the *block Jacobi method* consists of solving a tridiagonal system for each line of constant y. The jth system of the sweep has the form

$$D\mathbf{v}_j^h = \mathbf{g}_j^h \,,$$

where $\mathbf{v}_j^h$ is the jth subvector of $\mathbf{v}^h$ with entries $\left(\mathbf{v}_j^h\right)_i = v_{i,j}^h$, and $\mathbf{g}_j^h$ is the jth right-side vector with entries

$$\left(\mathbf{g}_j^h\right)_i = f_{i,j}^h + \frac{\epsilon}{h^2}\left(v_{i,j-1}^h + v_{i,j+1}^h\right).$$

Because D is tridiagonal, these systems can be solved efficiently with some form of Gaussian elimination. The operation count for relaxation remains $O(n)$, and so does the cost of either V- *or* W-cycle solvers. A weighted block Jacobi method results from averaging the current approximation and a full Jacobi update using a damping parameter ω.

To see exactly why line relaxation in the direction of strong coupling is effective, we need to look at the convergence properties of the iteration matrix. A brisk

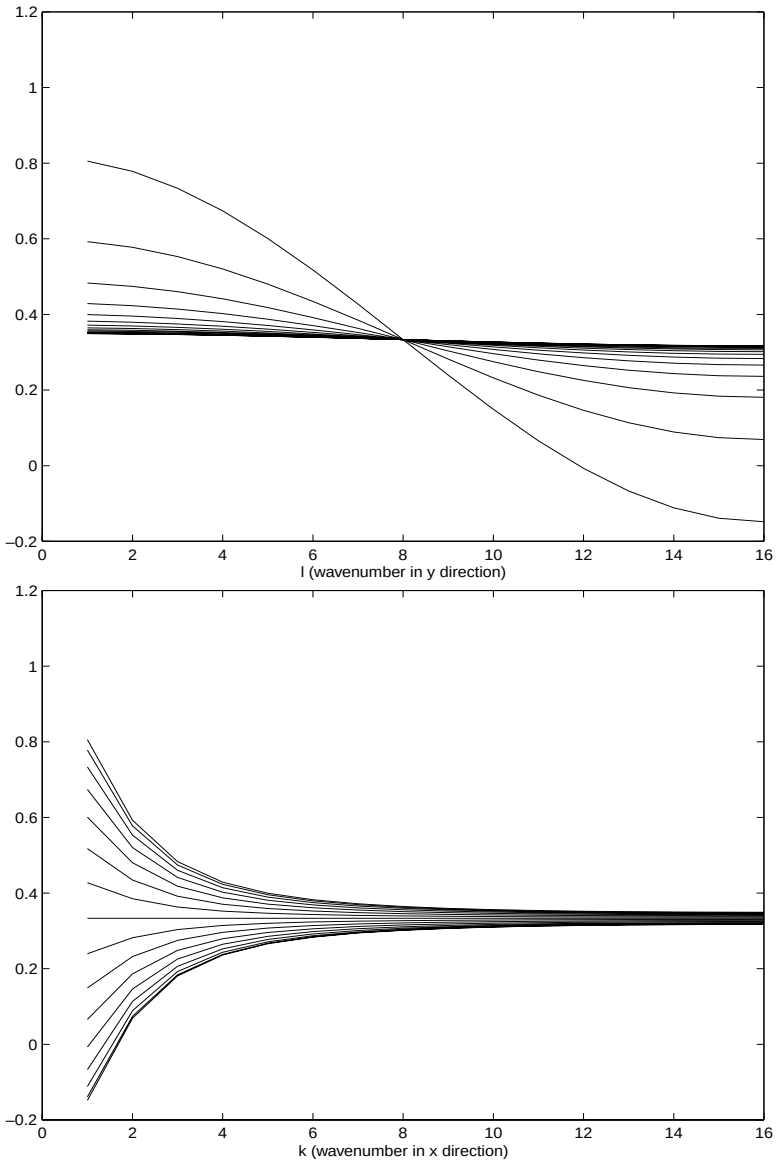

Figure 7.4: *The upper plot shows the eigenvalues of the block Jacobi iteration matrix, with* $\omega = \frac{2}{3}$*, along lines of constant* k *(wavenumber in the* x*-direction). The upper curve corresponds to* $k = 1$ *and the flattest curve to* $k = n = 16$*. The lower plot shows the same eigenvalues along lines of constant* ℓ *(wavenumber in the* y*-direction). The upper curve corresponds to* $\ell = 1$ *and the lower curve corresponds to* $\ell = n = 16$*. The small parameter in* (7.14) *is* $\epsilon = 0.05$*.*

calculation (Exercise 11) confirms that the eigenvalues of the *block* Jacobi iteration matrix (with damping parameter ω) are given by

$$\lambda_{k,\ell} = 1 - \frac{2\omega}{2\sin^2(\frac{k\pi}{2n}) + \epsilon}\left(\sin^2\left(\frac{k\pi}{2n}\right) + \epsilon\sin^2\left(\frac{\ell\pi}{2n}\right)\right), \quad 1 \le k, \ell \le n.$$

As before, k and ℓ are the wavenumbers in the x- and y-directions, respectively. The eigenvalues for the block Jacobi method (with $\omega = \frac{2}{3}$ and $\epsilon = 0.05$) are displayed in Fig. 7.4, which is analogous to Fig. 7.2 for the point Jacobi method. The top figure shows the eigenvalues along lines of constant k; the lower figure shows the same eigenvalues along lines of constant ℓ. The most striking difference between the eigenvalue plots for the two Jacobi methods is the reduction in the size of the eigenvalues: the maximum magnitude of the eigenvalues of the point method is 0.98, while the maximum magnitude of the eigenvalues of the block method is 0.81. This reduction in the size of the eigenvalues results in improved convergence. This improvement is typical of block methods that collect additional coefficients in the diagonal blocks. However, for anisotropic problems, the improvement occurs only if the coefficients in the direction of strong coupling appear in the diagonal blocks.

Both of these strategies are effective for the anisotropic models. Either one can be used when there is a weak connection in a coordinate direction that is known beforehand. Choosing between the two often depends on the nature of the application and the computer architecture involved. However, problems in which the weak direction is unknown or changing within the domain may not be efficiently handled by either method. For more robustness, we could use point relaxation and alternate semicoarsening between the x- and y-directions. However, the increased number of coarse grids that this approach requires is too awkward for most applications. We could also choose to apply full coarsening and alternating line relaxation that switches between x- and y-line relaxation. This approach can also be a bit awkward for some applications, particularly as it extends to three dimensions. Another choice is to apply *both* semi-coarsening and line relaxation.

Semicoarsening/line relaxation

Suppose we want to develop one method that can handle either of the following stencils:

$$A_1^h = \frac{1}{h^2}\begin{pmatrix} & -\epsilon & \\ -1 & 2+2\epsilon & -1 \\ & -\epsilon & \end{pmatrix} \text{ or } \quad A_2^h = \frac{1}{h^2}\begin{pmatrix} & -1 & \\ -\epsilon & 2+2\epsilon & -\epsilon \\ & -1 & \end{pmatrix}.$$

Semi-coarsening in the x-direction could be used to handle A_1^h and y-line relaxation to handle A_2^h. If we do both, the problem may be viewed as a stack of pencils in the y-direction as shown in Fig. 7.5. Line relaxation in the y-direction is used to solve the problem associated with each pencil, and coarsening is done simply by deleting every other pencil. There is no assumption here about the direction of weak connections; in fact, this approach applies well to the isotropic case. Again, we lose the $O(n)$ complexity of W-cycles, but V-cycle complexity remains $O(n)$ (Exercise 13).

Numerical example. Many of the above ideas can be illustrated with a fairly simple example. Consider equation (7.14),

$$-u_{xx} - \epsilon u_{yy} = f,$$

with homogeneous Dirichlet boundary conditions. The discretized problem has the stencil given by (7.15). We apply three different schemes to this problem for

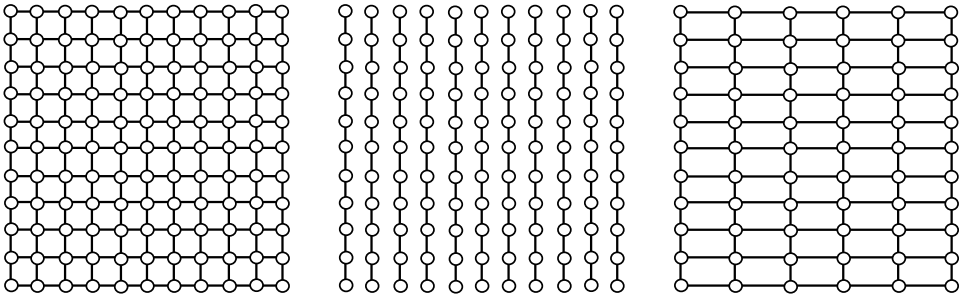

Figure 7.5: *It is possible to combine line relaxation and semicoarsening. The original grid (left) is viewed as a collection of pencils for line relaxation in the y-direction (center) and semi-coarsened in the x-direction (right).*

selected values of ϵ, ranging from large values, so that coupling is dominant in the y-direction, to very small values, so that coupling is dominant in the x-direction. Our example uses the right-side function

$$f(x, y) = 2(y - y^2) + 2\epsilon(x - x^2),$$

which produces the exact solution

$$u(x, y) = (x - x^2)(y - y^2).$$

The first approach consists of a standard V(2,1)-cycle, based on Gauss–Seidel relaxation, full coarsening, full weighting, and linear interpolation. We do not expect this method to be effective, except when $\epsilon \approx 1$. A simple demonstration pinpoints the difficulty we expect to encounter in this experiment.

Letting $\epsilon = 0.001$ and using a 16×16 grid with $n = 16$ points, we begin the iteration with a random initial guess. Figure 7.6 shows a surface plot of the error after 50 sweeps of Gauss–Seidel. We also show the error along a line of constant y (middle figure) and a line of constant x (bottom figure). Relaxation apparently smooths the error nicely in the x-direction, the direction of strong coupling, but leaves highly oscillatory error in the y-direction. While it is easy to envision approximating the smooth error curve on a coarser grid, we could not hope to do so with the oscillatory error. Therefore, we must keep all the points in the y-direction if we wish to represent the error accurately in that direction.

These observations lead to the second approach, which is semicoarsening. Because smoothing in the y-direction with point Gauss–Seidel is ineffective when ϵ is small, we coarsen *only* in the x-direction. This means that the coarse and fine grids have the same number of points in the y-direction. Full weighting and linear interpolation are used, but in a one-dimensional way that involves neighbors in the x-direction only. Semicoarsening means that we do not use neighbors in the y-direction for either restriction or interpolation.

It is important to notice that once the semicoarsening has been performed, the discrete operator must be altered to fit the new geometry. That is, because the grid spacing is no longer the same in the two directions, the residual and relaxation

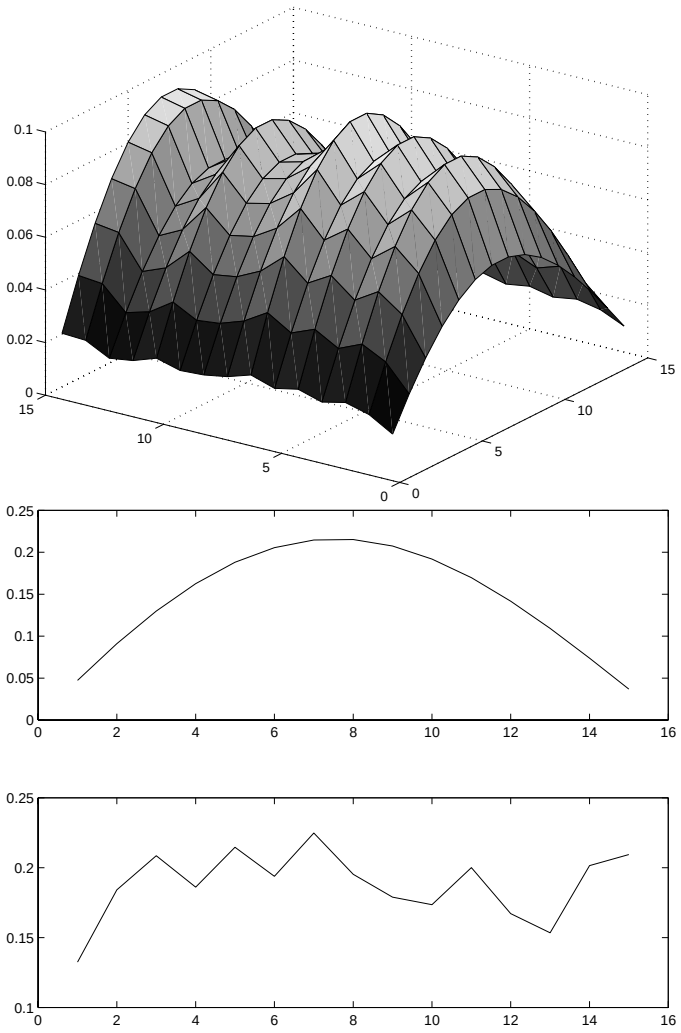

Figure 7.6: *Top: Error after* 50 *sweeps of pointwise Gauss–Seidel on the equation* $-u_{xx} - \epsilon u_{yy} = 0$*, beginning with a random initial guess. Middle: Error in the strongly coupled* x*-direction along a line of constant* y*. Bottom: Error along a line of constant* x*, in the weakly coupled* y*-direction.*

operators must use a stencil like

$$A = \begin{pmatrix} & -\frac{\epsilon}{h_y^2} & \\ -\frac{1}{h_x^2} & \left(\frac{2}{h_x^2} + \frac{2\epsilon}{h_y^2}\right) & -\frac{1}{h_x^2} \\ & -\frac{\epsilon}{h_y^2} & \end{pmatrix},$$

	ϵ								
Scheme	1000	100	10	1	0.1	0.01	0.001	0.0001	0
V(2,1)-cycles	0.95	0.94	0.58	0.13	0.58	0.90	0.95	0.95	0.95
Semi-C	0.99	0.99	0.98	0.93	0.71	0.28	0.07	0.07	0.07
Semi-C/line relax	0.04	0.08	0.08	0.08	0.07	0.07	0.07	0.08	0.08

Table 7.3: *Convergence factors for the three different schemes for various values of ϵ. The* V(2,1) *scheme is just standard multigrid, with Gauss–Seidel relaxation, linear interpolation, and full weighting. The semi-C scheme uses semicoarsening in the strongly coupled x-direction. The semi-C/line relax scheme combines semicoarsening in the x-direction with line relaxation in the y-direction.*

where h_x and h_y are the grid spacings in the x- and y-directions, respectively. Our earlier discussion suggests that this scheme will work well when the coupling is dominant in the x-direction (small ϵ), but that it may work poorly otherwise.

The third approach we consider is appropriate when the size of ϵ is unknown, so that the anisotropy could be in either coordinate direction. Here we use semicoarsening in the x-direction and line relaxation in the y-direction. Again, we use full weighting and linear interpolation with neighbors in the x-direction only. Semicoarsening should take care of strong coupling in the x-direction, while line relaxation should handle strong coupling in the y-direction.

The asymptotic convergence factors for the three approaches and for various choices of ϵ are displayed in Table 7.3. The standard V(2,1)-cycle approach works well only when ϵ is very near 1. It converges poorly for values of ϵ that differ from 1 by an order of magnitude or more. The semicoarsening approach performs extremely well when ϵ is very small, but its convergence deteriorates even when $\epsilon = 0.01$ and 0.1. There is a subtle lesson here. Semicoarsening is effective provided $\frac{1}{h_x^2}$ remains larger than $\frac{\epsilon}{h_y^2}$ as h_x is doubled. This means that the coarse grids see much the same anisotropy as the fine grid. However, if $\frac{1}{h_x^2}$ and $\frac{\epsilon}{h_y^2}$ become comparable in size during the x-coarsening process, then convergence rates will decrease. This effect can be observed in Table 7.3. With $n = 16$ and $\epsilon = 0.1$, coarsening in the x-direction quickly produces coupling in the y-direction. Rather than continue to semicoarsen in the x-direction, it is best to return to standard coarsening in this situation.

Finally, the combination of semicoarsening with line relaxation is extremely robust, giving excellent convergence results regardless of the degree of anisotropy.

◇◇

Variable-Mesh Problems

Nonuniform grids are commonly used in practice to accommodate irregularities in the problem domain or multiple scales in the emerging solution. To see how multigrid might handle such problems, consider the one-dimensional Poisson equation

$$\begin{aligned} -u''(x) &= f(x), \quad 0 < x < 1, \\ u(0) = u(1) &= 0. \end{aligned} \tag{7.17}$$

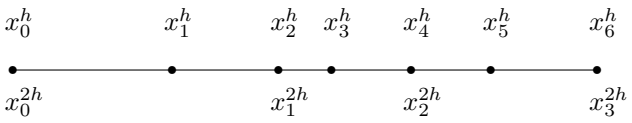

Figure 7.7: *With a variable mesh, the coarse grid is defined as the even-numbered grid points of the fine grid.*

With n a positive even integer, define a nonuniform grid on $[0\,,1]$ by the points x_j^h, $0 \le j \le n$, and mesh sizes $h_{j+\frac{1}{2}} \equiv x_{j+1}^h - x_j^h$, $0 \le j \le n-1$. Using second-order finite differences, we can discretize (7.17) on this grid. The result is (Exercise 14)

$$-\alpha_j^h u_{j-1}^h + \left(\alpha_j^h + \beta_j^h\right) u_j^h - \beta_j^h u_{j+1}^h = f_j^h\,, \quad 1 \le j \le n-1\,, \tag{7.18}$$

where the coefficients are given by

$$\alpha_j^h = \frac{2}{h_{j-\frac{1}{2}}\left(h_{j-\frac{1}{2}} + h_{j+\frac{1}{2}}\right)}\,, \quad \beta_j^h = \frac{2}{h_{j+\frac{1}{2}}\left(h_{j-\frac{1}{2}} + h_{j+\frac{1}{2}}\right)}\,, \quad 1 \le j \le n-1\,. \tag{7.19}$$

Note that the operator in (7.18) reduces to the standard stencil $\frac{1}{h^2}\,(-1 \quad 2 \quad -1)$ in the uniform grid case, $h_{i+\frac{1}{2}} = h$.

We can solve this discrete system by modifying the standard multigrid scheme to accommodate variable-mesh sizes. The operator I_{2h}^h is again based on linear interpolation, but it must now account for the nonuniformity of the grid. Choosing the coarse grid (Fig. 7.7) to consist of every other fine-grid point, $x_j^{2h} \equiv x_{2j}^h$, $0 \le j \le \frac{n}{2}$, the interpolation process, $\mathbf{v}^h = I_{2h}^h \mathbf{v}^{2h}$, is defined by

$$\begin{aligned} v_{2j}^h &= v_j^{2h}\,, \\ v_{2j+1}^h &= \frac{h_{2j+\frac{3}{2}} v_j^{2h} + h_{2j+\frac{1}{2}} v_{j+1}^{2h}}{h_{2j+\frac{1}{2}} + h_{2j+\frac{3}{2}}}, \quad 1 \le j \le \frac{n}{2} - 1. \end{aligned} \tag{7.20}$$

This formula comes from constructing a linear function in the interval $[x_j^{2h}\,, x_{j+1}^{2h}]$ that interpolates $\mathbf{v}^{2h}$ at the end points (Exercise 15).

Another way to derive (7.20) is to apply *operator interpolation*. The basic idea is to assume that relaxation does not significantly change the error, so that the residual must be almost zero. Using (7.18) at x_{2j+1}^h, we have

$$-\alpha_{2j+1}^h e_{2j}^h + \left(\alpha_{2j+1}^h + \beta_{2j+1}^h\right) e_{2j+1}^h - \beta_{2j+1}^h e_{2j+2}^h \approx 0. \tag{7.21}$$

Solving for e_{2j+1}^h yields

$$e_{2j+1}^h \approx \frac{\alpha_{2j+1}^h e_{2j}^h + \beta_{2j+1}^h e_{2j+2}^h}{\alpha_{2j+1}^h + \beta_{2j+1}^h}.$$

This suggests that we define interpolation by the formula

$$\begin{aligned} v_{2j}^h &= v_j^{2h}\,, \\ v_{2j+1}^h &= \frac{\alpha_{2j+1}^h v_j^{2h} + \beta_{2j+1}^h v_{j+1}^{2h}}{\alpha_{2j+1}^h + \beta_{2j+1}^h}, \quad 1 \le j \le \frac{n}{2} - 1\,. \end{aligned} \tag{7.22}$$

It is important that linear interpolation (7.20) and operator interpolation (7.22) be identical for our one-dimensional model problem (Exercise 16). Note that both (7.20) and (7.22) reduce to a simple average for the uniform grid case.

Full weighting can now be defined by the variational property

$$I_h^{2h} \equiv \frac{1}{2}\left(I_{2h}^h\right)^T$$

and the coarse-grid operator can be constructed from the Galerkin principle

$$A^{2h} = I_h^{2h} A^h I_{2h}^h .$$

This definition of A^{2h} yields a stencil that is similar, but not identical, to the stencil obtained by using (7.18)–(7.19) on the coarse grid. However, the Galerkin principle does hold if this system is rescaled by $(h_{j-\frac{1}{2}} + h_{j+\frac{1}{2}})/2$ (Exercise 17).

Variable-mesh problems in two dimensions can be handled in a similar way, but the first issue in this case is the discretization itself. Forming an accurate discrete approximation can be tricky on nonuniform grids in two and higher dimensions. Moreover, if the grid is *logically rectangular* (it can be indexed simply by i, j), then coarsening is generally straightforward; but unstructured grids can make coarsening seem almost impossible. Instead of treating such cases here, we defer this issue to our discussion of the algebraic multigrid algorithm (Chapter 8), which was designed just for this purpose.

Variable-Coefficient Problems

Another common practical problem is the solution of variable coefficient differential equations. Again we consider a simple case to see how multigrid might handle such problems. Our model problem is the scalar elliptic equation

$$\begin{aligned} -(a(x)u')'(x) &= f(x), \quad 0 < x < 1, \\ u(0) = u(1) &= 0, \end{aligned} \tag{7.23}$$

where $a(x)$ is a positive function on the interval $(0\,,1)$. A *conservative* or *self-adjoint* method for discretizing (7.23) is developed as follows.

Letting n be a positive even integer, consider the *uniform* grid defined by the points $x_j^h = \frac{j}{n}, \quad 0 \le j \le n$. As usual, suppose that the vector $\mathbf{v}^h$ approximates $\mathbf{u}$ at these grid points, but that values of the coefficient a are taken at the cell centers $x_{j+\frac{1}{2}}^h = \frac{j+\frac{1}{2}}{n}$; thus, we have $a_{j+\frac{1}{2}}^h = a(x_{j+\frac{1}{2}}^h)$ for $0 \le j \le n-1$. The discrete system that results (Exercise 18) is

$$\begin{aligned} \frac{1}{h^2}\left(-a_{j-\frac{1}{2}}^h u_{j-1}^h + (a_{j-\frac{1}{2}}^h + a_{j+\frac{1}{2}}^h)u_j^h - a_{j+\frac{1}{2}}^h u_{j+1}^h\right) &= f_j^h, \quad 1 \le j \le n-1, \\ u_0^h = u_n^h &= 0. \end{aligned} \tag{7.24}$$

Note that (7.24) becomes the usual discretization in the case $a(x) = 1$.

For slowly varying $a(x)$, we can use a standard multigrid scheme much like the algorithm used for the model problem. One reasonable approach in this case is to use linear interpolation for I_{2h}^h, full weighting for I_h^{2h}, and the following stencil for A^{2h}:

$$A^{2h} = \frac{1}{(2h)^2}\left(-a_{j-\frac{1}{2}}^{2h} \qquad a_{j-\frac{1}{2}}^{2h} + a_{j+\frac{1}{2}}^{2h} \qquad -a_{j+\frac{1}{2}}^{2h}\right), \qquad 1 \le j \le \frac{n}{2} - 1, \tag{7.25}$$

where the coefficients are determined by the simple averages

$$a^{2h}_{j+\frac{1}{2}} = \frac{a^h_{2j+\frac{1}{2}} + a^h_{2j+\frac{3}{2}}}{2}\,, \quad 0 \le j \le \frac{n}{2} - 1\,. \tag{7.26}$$

This same stencil can be obtained (Exercise 19) from the Galerkin principle

$$A^{2h} = I^{2h}_h A^h I^h_{2h}.$$

The performance of such a *standard* multigrid scheme will degrade as $a(x)$ begins to vary significantly from cell to cell. One way to understand how this degradation happens is to compare (7.24), the uniform mesh discretization of the variable-coefficient problem, to (7.18)–(7.19), the variable-mesh discretization of the constant coefficient Poisson problem. From the perspective of the fine-grid point x^h_{2j+1}, the variable coefficient problem can be viewed as a Poisson problem with variable mesh sizes defined by

$$h_{2j+\frac{1}{2}} = \frac{\gamma^h_{2j+1}}{a^h_{2j+\frac{1}{2}}}\,, \quad h_{2j+\frac{3}{2}} = \frac{\gamma^h_{2j+1}}{a^h_{2j+\frac{3}{2}}}\,, \tag{7.27}$$

where

$$\gamma^h_{2j+1} = h\sqrt{\frac{2a^h_{2j+\frac{1}{2}}a^h_{2j+\frac{3}{2}}}{a^h_{2j+\frac{1}{2}} + a^h_{2j+\frac{3}{2}}}}\,, \tag{7.28}$$

(Exercise 20). The point that we should get from these rather cumbersome formulas is that the variable-coefficient problem on a uniform mesh can be viewed locally as a constant-coefficient equation on a nonuniform mesh. This relationship is important because it says that standard multigrid for the variable-coefficient problem corresponds to using interpolation based on simple averaging for the nonuniform mesh problem. If the point x^h_{2j+1} of the variable mesh is close to x^h_{2j}, but far from x^h_{2j+2}, then simple averaging cannot accurately represent smooth components (Exercise 21).

A simple remedy is to define interpolation for the variable-coefficient problem guided by the variable-mesh case. The easiest way is to use operator interpolation, which gives (Exercise 22)

$$\begin{aligned} v^h_{2j} &= v^{2h}_j\,, \\ v^h_{2j+1} &= \frac{a^h_{2j+\frac{1}{2}}v^{2h}_j + a^h_{2j+\frac{3}{2}}v^{2h}_{j+1}}{a^h_{2j+\frac{1}{2}} + a^h_{2j+\frac{3}{2}}}\,, \quad 1 \le j \le \frac{n}{2} - 1. \end{aligned} \tag{7.29}$$

Restriction and the coarse-grid operator can then be defined by the variational relations

$$I^{2h}_h = \frac{1}{2}\left(I^h_{2h}\right)^T \qquad \text{and} \qquad A^{2h} = I^{2h}_h A^h I^h_{2h}\,.$$

Variable-coefficient problems can be handled in two dimensions by an analogous, but more complicated, operator interpolation scheme. The principle difficulty in extending the one-dimensional approach is that the two-dimensional version of (7.21)

$a(x) = 1 + \rho \sin(k\pi x)$							$a(x) = 1 + \rho\,\mathrm{rand}(x)$
ρ	$k = 3$	$k = 25$	$k = 50$	$k = 100$	$k = 200$	$k = 400$	
0	0.085	0.085	0.085	0.085	0.085	0.085	0.085
0.25	.084	.098	.098	.094	.093	.083	0.083
0.50	.093	.185	.194	.196	.195	.187	0.173
0.75	.119	.374	.387	.391	.390	.388	0.394
0.85	.142	.497	.510	.514	.514	.526	0.472
0.95	.191	.681	.690	.694	.699	.745	0.672

Table 7.4: *Average convergence factors for* 20 V*-cycles of the variable-coefficient method. With* $a(x) = 1 + \rho \sin(k\pi x)$*, there is strong dependence on* ρ*, but relative insensitivity to all but the smallest value of* k*. With* $a(x) = 1 + \rho\ rand(x)$*, convergence factors depend strongly on the amplitude* ρ*.*

generally cannot be solved for the error corresponding to the fine-grid point in terms of neighboring coarse-grid points alone. For example, for the nine-point stencil

$$A_{i,j}^h = \frac{1}{3h^2} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \tag{7.30}$$

with a fully coarsened grid (i.e., every other fine-grid line is deleted), the equation at any fine-grid point involves at least four points that do not correspond to coarse-grid points. This difficulty can be reduced by considering semicoarsening algorithms, but the equations at fine-grid points still involve other fine-grid points that do not belong to the coarse grid. Nevertheless, there are effective ways to deal with this difficulty that involve collapsing the stencil to eliminate these troublesome fine-grid couplings, which in turn allow approximate operator interpolation. We leave this issue to the study of algebraic multigrid in the next chapter.

Numerical example. We illustrate the performance of a basic multigrid method applied to two variable-coefficient problems. We solve the one-dimensional problem (7.23) using the discretization in (7.24) and coarse-grid operator (7.25). The variable coefficient is

$$a(x) = 1 + \rho \sin(k\pi x), \tag{7.31}$$

for various choices of $\rho > 0$ and positive integers k, and

$$a(x) = 1 + \rho\,\mathrm{rand}(x), \tag{7.32}$$

where rand(x) returns a random number between –1 and 1. Noting that $a(x) = 1$ corresponds to the Poisson equation, these functions correspond to perturbations of the model problem. By increasing the size of the perturbation and the amount of local change in the perturbation, we are able to explore the behavior of the method for a wide variety of situations.

The grid size in all tests is $n = 1024$, which we found to be representative of the tests done over a wide range of n . We use V(2,1)-cycles based on Gauss–Seidel relaxation, full weighting, and linear interpolation. The results are displayed in Table 7.4.

For coefficients (7.31), the method depends strongly on the value of ρ, but very little on the value of the wavenumber k. For $\rho = 0.25$, the method works very well,

with results akin to the model Poisson problem. However, with increasing ρ, performance degrades rapidly until convergence becomes quite poor. This dependence on ρ is to be expected: for ρ near 1, $a(x)$ has significant variation, from values near 0 to values near 1. One might expect that performance would also depend strongly on the wavenumber k. However, apart from the noticeable jump in convergence factors between $k = 3$ and $k = 25$, this does not appear to be the case, suggesting that standard multigrid methods are more sensitive to the amplitude of coefficient variations than to their frequency.

For coefficient (7.32), we drop the regular oscillatory variation in $a(x)$ in favor of random jumps. The various amplitudes, ρ, of the coefficient match those in the first case. Here, we again see a strong correlation between the amplitude and the convergence factor. For small amplitudes, the method performs well, but the performance degrades as ρ nears 1. It is interesting that the method seems relatively unaffected by the random variations in the coefficient. It performs at least as well with the random coefficients as it does with all but the lowest wavenumber case ($k = 3$) of the smoothly varying coefficients. ◇◇

These examples serve to highlight the point that a basic multigrid scheme may be used quite effectively for the variable-coefficient problem, provided the variation in the coefficient function is not too drastic. On the other hand, widely varying coefficients require more sophisticated multigrid methods. One approach to such problems is to use an algebraic multigrid method, which is the subject of the next chapter.

Exercises

Neumann Boundary Conditions

1. **Differential nonuniqueness.** Show that Neumann problem (7.1) does not have a *unique* solution by showing that if u solves (7.1), then $u + c$ is also a solution for any constant c. This means that if *any* particular solution exists, then another can be constructed simply by adding a constant. Now show that adding a general constant produces *all* solutions of (7.1). Hint: If u and v solve (7.1), then $(u - v)'' = 0$, so $u - v = ax + c$ for some constants a and c; now apply the boundary conditions.

2. **Differential solvability.** Show that Neumann problem (7.1) is solvable only for the special source terms f that satisfy integral compatibility condition (7.4). Hint: Assume u solves (7.1), then integrate both sides of the equation and apply the boundary conditions.

3. **Discrete null space.** Show by inspecting (7.3) that the null space of A^h is the set $\{c\mathbf{1}^h\}$.

4. **Discrete nonuniqueness.** Show that discrete system (7.2)–(7.3) does not have a *unique* solution by proving that if $\mathbf{u}^h$ is a particular solution, then the general solution is $\mathbf{u}^h + c\mathbf{1}^h$. Hint: First use Exercise 3 to confirm that $\mathbf{u}^h + c\mathbf{1}^h$ is indeed a solution; then prove that any two solutions must differ by a vector in the null space of A^h.

5. **Discrete solvability.** Show that discrete system (7.9)–(7.10) is solvable only for the special source terms $\hat{\mathbf{f}}^h$ that satisfy discrete compatibility condition (7.7). Hint: Use Exercise 3, the fact that the range of a matrix is the orthogonal complement of the null space of its transpose, and the symmetry of $\hat{A}^h$.

6. **Restriction at the boundary points.** Using the interpolation operator at the left boundary given in (7.12) and the variational condition $I_h^{2h} = \frac{1}{2}\left(I_{2h}^h\right)^T$, derive the restriction formula at the left boundary point, $f_0^{2h} \leftarrow \frac{1}{2}f_0^h + \frac{1}{4}f_1^h$.

7. **Coarse-grid solvability.** Consider the coarse-grid equation

$$A^{2h}\mathbf{u}^{2h} = I_h^{2h}\left(\mathbf{f}^h - A^h\mathbf{u}^h\right).$$

   (a) Show that interpolation preserves constants; that is, $\mathbf{1}^h = I_{2h}^h\mathbf{1}^{2h}$.

   (b) Show that the variational property, $I_{2h}^h = c(I_h^{2h})^T$, guarantees that the discrete compatibility condition (7.7) is satisfied (that is, the coarse-grid problem is solvable).

   (c) The variational property is actually not necessary for the discrete compatibility condition to hold. Show that it is enough for restriction to satisfy the property that its column sums equal a given constant γ (this means that a fine-grid residual is distributed to coarse-grid points with weights that sum to γ).

8. **Inhomogeneous Neumann conditions.** Consider the two-point boundary value problem

$$\begin{aligned} -u''(x) &= f(x), \quad 0 < x < 1\,, \\ u'(0) &= g_0\,, \\ u'(1) &= g_1\,, \end{aligned}$$

   where g_0 and g_1 are given constants. Modify the essential concepts developed for (7.1) to accommodate this inhomogeneous case. In particular, show that

   (a) compatibility condition (7.4) now involves a nonzero right side;

   (b) the right side f^h of (7.2) and $\hat{f}_h$ of (7.9) change accordingly to incorporate g_0 and g_1;

   (c) the rest of the development (the uniqueness condition of (7.5) and (7.10), the discrete compatibility condition in (7.7), and the coarse-grid correction process) is essentially unchanged.

Anisotropic Problems

9. **Two sources of anisotropy.** Show that, after discretization of the two-dimensional Poisson equation (7.14), the small parameter $\epsilon > 0$ in front of the u_{yy} term is equivalent to using a mesh spacing of $h_x = h$ in the x-direction and mesh spacing $h_y = \frac{h}{\sqrt{\epsilon}}$ in the y-direction.

10. Eigenvalues for point Jacobi method. Suppose the weighted point Jacobi method is applied to the system of equations corresponding to (7.15). Recall that A^h can be expressed as $D - L - U$, where D represents the diagonal elements of A^h and U and L represent the respective upper and lower triangular parts of A^h. Then the weighted Jacobi iteration matrix is given by $P_J = I - \omega D^{-1} A^h$.

(a) Write out a typical equation of the eigenvalue system $P_J \mathbf{v} = \lambda \mathbf{v}$.

(b) Assume an $n \times n$ grid and an eigenvector solution of the form

$$v_{ij} = \sin\left(\frac{ik\pi}{n}\right) \sin\left(\frac{j\ell\pi}{n}\right), \quad 1 \le k, \ell \le n-1.$$

Using sine addition rules, simplify this eigenvalue equation, cancel common terms, and show that the eigenvalues are given by

$$\lambda_{k\ell} = 1 - \frac{2\omega}{1+\epsilon}\left(\sin^2\left(\frac{k\pi}{2n}\right) + \epsilon \sin^2\left(\frac{\ell\pi}{2n}\right)\right), \quad 1 \le k, \ell \le n.$$

11. Eigenvalues for block Jacobi method. Consider A^h given in (7.16) and write it in the form $A^h = \mathcal{D} - U - L$, where $\mathcal{D}$ is the block diagonal matrix consisting of the blocks D on the diagonal and U and L are the respective lower and upper triangular parts of A^h.

(a) Show that the weighted block Jacobi iteration matrix can be written in the form

$$P_J = \mathcal{D}^{-1}(U + L) = I - \omega \mathcal{D}^{-1} A^h.$$

(b) Write out a typical equation of the eigenvalue system $P_J \mathbf{v} = \lambda \mathbf{v}$.

(c) Assume an $n \times n$ grid and an eigenvector solution of the form

$$v_{ij} = \sin\left(\frac{ik\pi}{n}\right) \sin\left(\frac{j\ell\pi}{n}\right), \quad 1 \le k, \ell \le n-1.$$

Using sine addition rules, simplify this eigenvalue equation, cancel common terms, and show that the eigenvalues are given by

$$\lambda_{k,\ell} = 1 - \frac{2\omega}{2\sin^2\left(\frac{k\pi}{2n}\right) + \epsilon}\left(\sin^2\left(\frac{k\pi}{2n}\right) + \epsilon \sin^2\left(\frac{\ell\pi}{2n}\right)\right), \quad 1 \le k, \ell \le n.$$

(d) Compare the magnitudes of the eigenvalues of the block Jacobi method to those of the point Jacobi method (previous problem).

12. Semicoarsening/point relaxation complexity. Using the techniques of Chapter 4, for an $n \times n$ grid, show that with semicoarsening and point relaxation, the computational cost of a W-cycle is larger than $O(n^2)$, but that a V-cycle retains $O(n^2)$ complexity.

13. **Semicoarsening/line relaxation complexity.** Modifying the argument of the previous problem, show that with semicoarsening and line relaxation, the computational cost of a V-cycle is $O(n^2)$.

Variable-Mesh and Variable-Coefficient Problems

14. **Variable-mesh discretization.** Show that when model problem (7.17) is discretized using second-order finite difference approximations, difference equations (7.18) and (7.19) result.

15. **Linear interpolation for variable meshes.** Show that linear interpolation on a nonuniform grid leads to the formula expressed in (7.20). Hint: Construct a linear function in the interval $[x_j^{2h}, x_{j+1}^{2h}]$ that equals $\mathbf{v}^{2h}$ at the end points, then evaluate it at $x_{2j+1}^h = x_j^{2h} + h_{2j+\frac{1}{2}}$.

16. **Operator interpolation for variable meshes.** Show that interpolation formulas (7.20) and (7.22) are algebraically equivalent.

17. **Galerkin operator for variable meshes.** Show that the Galerkin principle applied to problem (7.18)–(7.19) leads to a similar, but generally different stencil. Show that the Galerkin principle does hold if (7.18) is multiplied on both sides by $(h_{j-\frac{1}{2}} + h_{j+\frac{1}{2}})/2$.

18. **Variable-coefficient discretization.** Verify that the discretization of the variable-coefficient problem (7.23) results in the difference equation (7.24).

19. **Galerkin operator for variable coefficients.** Show that the Galerkin coarse-grid operator for (7.24), using linear interpolation and full weighting, is the same as operator (7.25)–(7.26) obtained by simple averages of the fine-grid coefficients.

20. **Variable-coefficient versus variable-mesh.** Verify formula (7.27)–(7.28) that relates the variable-coefficient problem on a uniform mesh to Poisson's equation on a nonuniform mesh.

21. **Simple averaging for variable meshes.** To see that interpolation defined by simple averages can be ineffective for grids with widely varying mesh sizes, consider the three grid points $x_0^h = x_0^{2h} = 0, x_1^h = \epsilon h$, and $x_2^h = x_1^{2h} = 2h$, where h and ϵ are small positive parameters. Note that the smooth linear function $u(x) = \frac{x}{2h}$ is 0 at x_0^h and 1 at x_2^h. Show that it cannot be approximated well at x_1^h by the average of these end point values. Obtain an expression for the error at x_1^h in terms of h and ϵ and discuss the behavior as $\epsilon \to 0$.

22. **Interpolation for variable-coefficient problems.** Use the operator interpolation approach (assume the error satisfies the residual equation) to derive interpolation formulas (7.29).