Chapter 9

Multilevel Adaptive Methods

Numerical problems often exhibit special features in small local regions that require resolution and accuracy well beyond what is required in the rest of the domain. In numerical weather modeling, isolated phenomena such as tornados or storm fronts may demand substantially enhanced accuracy. The numerical simulation of the flight of an aircraft may require especially accurate approximations around the fuselage or wings. In these and many other cases, it is wasteful to let local accuracy requirements dictate the global discretization and solution process. The goal of this chapter is to understand how to treat local demands in a multilevel context without overburdening the overall computation.

There is a wide variety of methods that effectively treat local demands. Here we consider the so-called *fast adaptive composite* grid method (FAC) [14]. Its distinctive features are that it always works with uniform grids and subgrids, and it is in tune with the variational theme that appears throughout this book. We begin with a one-dimensional example.

Consider the two-point boundary value problem

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

To keep matters simple, assume that a *local fine grid* with mesh size $h = \frac{1}{8}$ is needed on the interval $(\frac{1}{2}, 1)$ to resolve some special feature near $x = \frac{3}{4}$, but that a mesh size of $2h = \frac{1}{4}$ is deemed adequate for the rest of the domain. This need might arise, for example, in the presence of a source term f that is nonzero only near $x = \frac{3}{4}$ (Fig. 9.1).

Imagine first that we solve the problem on a *global* grid, Ω^h, with mesh size $h = \frac{1}{8}$, using a two-grid method based on linear interpolation, full weighting, and the variational properties as described in Chapter 5. We write the discrete problem as

$$A^h \mathbf{u}^h = \mathbf{f}^h\,. \tag{9.1}$$

In component form, we have

$$\begin{aligned} \frac{-u^h_{i-1} + 2u^h_i - u^h_{i+1}}{h^2} &= f^h_i, \qquad 1 \le i \le n\,, \\ u^h_0 = u^h_{n+1} &= 0, \end{aligned}$$

where $n = 7$ for this case.

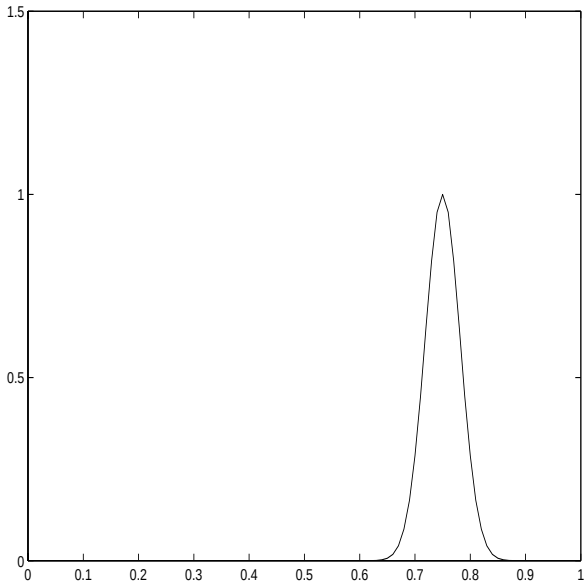

Figure 9.1: *Graph of a sharply peaked impulse function centered at* $x = \frac{3}{4}$.

This simple uniform grid involves unnecessary computation: the accuracy required *locally* at $x = \frac{3}{4}$ dictates the *global* mesh size $h = \frac{1}{8}$, forcing us to solve the problem with too many grid points. To reduce this waste, we can start by eliminating relaxation (the most expensive multigrid process) at fine-grid points where it is not needed, that is, in $[0, \frac{1}{2}]$. However, we can do better. As we will see, it is possible to reduce costs further by selectively eliminating interpolation, restriction, and the computation of residuals.

Consider Fig. 9.2, which shows the fine grid Ω^h with $n = 7$ interior points and the underlying coarse grid Ω^{2h} with $\frac{n-1}{2} = 3$ interior points. Regular multigrid relaxes on all fine-grid points, denoted by $*, \bullet$, and $\circ$, then performs coarse-grid correction based on the points denoted by $\times$.

With this notation, let us see how we might eliminate points of Ω^h in $[0, \frac{1}{2})$. Restricting relaxation to the local fine grid in $(\frac{1}{2}, 1)$ means that the Ω^h residuals change at $*$ and $\bullet$ points, because the approximation $\mathbf{v}^h$ changes at $*$ points. However, *fine-grid residuals do not change at* $\circ$ *points between successive coarse-grid solves.* Similarly, the coarse-grid residual does not change at $\times = \frac{1}{4}$ because it is obtained by restriction from $\circ$ points.

For the moment, suppose that we continue to store $\mathbf{v}^h$ at $\circ$ points, just for convenience. We must be sure that the algorithm starts with the correct right sides, f_i^{2h}, on the coarse grid, particularly at the point $\times = \frac{1}{4}$. We begin with a zero initial guess, $\mathbf{v}^h = \mathbf{0}$, so the residual at the coarse-grid point $\times = \frac{1}{4}$ is just the restriction of the fine-grid source term f_i^h. With this observation, we can give our first scheme:

- Initialize $\mathbf{v}^h = \mathbf{0}$ and $f_1^{2h} \equiv \left(I_h^{2h}\mathbf{f}^h\right)_1 = \frac{f_1^h + 2f_2^h + f_3^h}{4}$.
- Relax on $\mathbf{v}^h$ on the local fine grid $\{x_5^h, x_6^h, x_7^h\} = \{\frac{5}{8}, \frac{3}{4}, \frac{7}{8}\}$.

Ω^h	x_0^h	x_1^h	x_2^h	x_3^h	x_4^h	x_5^h	x_6^h	x_7^h	x_8^h
Ω^h	$\odot$	$\circ$	$\circ$	$\circ$	$\bullet$	$*$	$*$	$*$	$\odot$
$x =$	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{3}{4}$	$\frac{7}{8}$	1
Ω^{2h}	$\odot$		$\times$		$\times$		$\times$		$\odot$
Ω^{2h}	x_0^{2h}		x_1^{2h}		x_2^{2h}		x_3^{2h}		x_4^{2h}

Figure 9.2: *Local fine-grid points on Ω^h are denoted by $*$ and the interface point is denoted by $(\bullet)$. Computation at the remaining fine-grid points $(\circ)$ is eliminated. The coarse-grid points $(\times)$ comprise Ω^{2h}, while $\odot$ represents boundary points.*

- Compute fine-grid residual $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{v}^h$ and transfer it to the coarse grid:
$$f_2^{2h} \leftarrow \frac{r_3^h + 2r_4^h + r_5^h}{4}\,,\; f_3^{2h} \leftarrow \frac{r_5^h + 2r_6^h + r_7^h}{4}\,.$$
- Compute an approximation $\mathbf{v}^{2h}$ to the coarse-grid residual equation $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$.
- Update the residual at $\times = \frac{1}{4}$ for use in later cycles: $f_1^{2h} \leftarrow f_1^{2h} - \frac{-v_0^{2h}+2v_1^{2h}-v_2^{2h}}{(2h)^2}$.
- Interpolate the correction and update the approximation: $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h\mathbf{v}^{2h}$.

At this point, reducing the number of computations further becomes a little tricky. It might seem at first glance that the fine-grid approximation, $\mathbf{v}^h$, is never used outside of $[\frac{1}{2}, 1]$, but this is not quite true: the residual transfer in the third step involves the residuals $\mathbf{r}^h$ at $\bullet = \frac{1}{2}$ and $\circ = \frac{3}{8}$, which in turn involve $\mathbf{v}^h$ at $\circ = \frac{1}{4}$ and $\circ = \frac{3}{8}$. Thus, we need to store $\mathbf{v}^h$ at these two *border points* just outside the local fine grid. On the other hand, we do not need to store v_1^h at $\circ = \frac{1}{8}$. This observation suggests introducing a new variable w_1^{2h} that accumulates the coarse-grid solution at $\times = \frac{1}{4}$, which underlies the fine-grid point $\circ = \frac{1}{4}$. We can identify $w_1^{2h} = v_2^h$, allowing us to keep track of the solution at this border point. The solution at the other border point, $\circ = \frac{3}{8}$, will be found by interpolation. The improved algorithm now appears as follows:

- Initialize $\mathbf{v}^h = \mathbf{0}, w_1^{2h} = 0$, and $f_1^{2h} \equiv \left(I_h^{2h}\mathbf{f}^h\right)_1 = \frac{f_1^h + 2f_2^h + f_3^h}{4}$.
- Relax on $\mathbf{v}^h$ on the local fine grid $\{x_5^h, x_6^h, x_7^h\} = \{\frac{5}{8}, \frac{3}{4}, \frac{7}{8}\}$.
- Compute fine-grid residual $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{v}^h$ on the local fine grid, its interface point $x_4^h = \frac{1}{2}$, and the border point $x_3^h = \frac{3}{8}$, and transfer it to the coarse grid:
$$f_2^{2h} \leftarrow \frac{r_3^h + 2r_4^h + r_5^h}{4}\,,\; f_3^{2h} \leftarrow \frac{r_5^h + 2r_6^h + r_7^h}{4}\,.$$

- Compute an approximation $\mathbf{v}^{2h}$ to the coarse-grid residual equation $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$.
- Update the residual at $\times = \frac{1}{4}$ for use in later cycles: $f_1^{2h} \leftarrow f_1^{2h} - \frac{-v_0^{2h}+2v_1^{2h}-v_2^{2h}}{(2h)^2}$.
- Update the coarse-grid approximation at $\times = \frac{1}{4}$: $w_1^{2h} \leftarrow w_1^{2h} + v_1^{2h}$.
- Interpolate the correction and update the approximation: $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{v}^{2h}$ everywhere except at $\circ = \frac{1}{8}, \frac{1}{4}$.

Note that we do not interpolate to $\circ = \frac{1}{4}$, because the solution at this point is held in w_1^{2h}, nor to $\circ = \frac{1}{8}$, because a fine-grid solution is not needed at this point.

We have devoted an absurd amount of effort here to avoiding computation at the single point $\circ = \frac{1}{8}$. However, it is important to imagine situations in which the local fine grid comprises a very small fraction of the domain. In such situations, points like $\circ = \frac{1}{8}$ predominate and the savings can be significant. As we see shortly, the savings can be even greater for two-dimensional problems.

We can take one more step and effectively remove the border points from the computation. Consider the residual f_2^{2h} at the interface point $\bullet = \frac{1}{2}$; it is obtained by full weighting the fine-grid residuals r_3^h, r_4^h, and r_5^h. Because w_1^{2h} accumulates the solution at $\circ = \frac{1}{4}$, we have $v_2^h = w_1^{2h}$; and because the solution at $\circ = \frac{3}{8}$ is determined by interpolation, we have $v_3^h = (w_1^{2h} + v_4^h)/2$. Thus, the fine-grid residuals that contribute to f_2^{2h} are computed as follows:

$$\begin{aligned} r_3^h &= f_3^h - \frac{-v_2^h + 2v_3^h - v_4^h}{h^2} \\ &= f_3^h - \frac{-w_1^{2h} + w_1^{2h} + v_4^h - v_4^h}{h^2} \\ &= f_3^h, \qquad (9.2) \\ r_4^h &= f_4^h - \frac{-(w_1^{2h} + v_4^h)/2 + 2v_4^h - v_5^h}{h^2} \\ &= f_4^h - \frac{-\frac{1}{2}w_1^{2h} + \frac{3}{2}v_4^h - v_5^h}{h^2}, \qquad (9.3) \\ r_5^h &= f_5^h - \frac{-v_4^h + 2v_5^h - v_6^h}{h^2}\,. \qquad (9.4) \end{aligned}$$

Writing $g_2^{2h} \equiv \frac{1}{4}\left(f_3^h + 2f_4^h + f_5^h\right)$, a bit of algebra yields the residual f_2^{2h} at the interface point:

$$\begin{aligned} f_2^{2h} &\equiv \frac{1}{4}\left(r_3^h + 2r_4^h + r_5^h\right) \qquad (9.5) \\ &= g_2^{2h} - \frac{-w_1^{2h} + 2v_4^h - v_6^h}{(2h)^2}\,. \end{aligned}$$

Note that the stencil on the right side of (9.5) corresponds to the usual Ω^{2h} stencil for the residual. This is a result of the variational properties at work. Yet, even in more general cases, it is possible to compute the *result* of transferring the fine-grid residual to the interface point(s). In turn, this allows us to eliminate the border points. These simplifications lead to our final algorithm.

Fast Adaptive Composite Grid Method (FAC)

- Initialize $\mathbf{v}^h = \mathbf{0}, w_1^{2h} = 0$, and $f_1^{2h} \equiv \left(I_h^{2h}\mathbf{f}^h\right)_1 = \frac{f_1^h + 2f_2^h + f_3^h}{4}$.
- Relax on $\mathbf{v}^h$ on the local fine grid $\{x_5^h, x_6^h, x_7^h\} = \{\frac{5}{8}, \frac{3}{4}, \frac{7}{8}\}$.
- Compute the right sides for the local coarse grid: $f_2^{2h} \leftarrow g_2^{2h} - \frac{-w_1^{2h} + 2v_4^h - v_6^h}{(2h)^2}$, $f_3^{2h} \leftarrow \frac{r_5^h + 2r_6^h + r_7^h}{4}$.
- Compute an approximation $\mathbf{v}^{2h}$ to the coarse-grid residual equation $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$.
- Update the residual at $\times = \frac{1}{4}$ for use in later cycles: $f_1^{2h} \leftarrow f_1^{2h} - \frac{-v_0^{2h} + 2v_1^{2h} - v_2^{2h}}{(2h)^2}$.
- Update the coarse-grid approximation at $\times = \frac{1}{4}$: $w_1^{2h} \leftarrow w_1^{2h} + v_1^{2h}$.
- Interpolate the correction and update the approximation: $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{v}^{2h}$ at the local fine grid and the interface points $\{\frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}\}$.

Summary of FAC Terms

- The *local fine grid* is the finest grid used for computation; it covers only that part of the domain where additional resolution is needed. FAC avoids computation on a global fine grid.
- The *global coarse grid*, Ω^{2h}, is the finest grid used for computation that covers the entire domain. The notation suggests a mesh refinement factor of two, but larger factors are permitted.
- *Interface points* are the boundary points of the local fine grid.
- *Border points* are fine-grid points that lie outside the local fine grid. They are used temporarily to develop special interface stencils.
- *Boundary points* are the usual points of the domain where boundary conditions apply.
- *Slave points*, which appear only in two and higher dimensions, are interface points that do not correspond to coarse-grid points.
- The *composite grid* is the combination of the fine- and coarse-grid points on which the discrete solution is ultimately determined.

It is important to see FAC from a different perspective. We developed the scheme by eliminating the $\circ$ points from the fine grid and producing the top two uniform grids (Ω^h and Ω^{2h}) shown in Fig. 9.3. The question we need to ask now is: At what points are we actually approximating the solution? When we use FAC, the solution is approximated at the *composite grid* (Ω^c in Fig. 9.3) consisting of

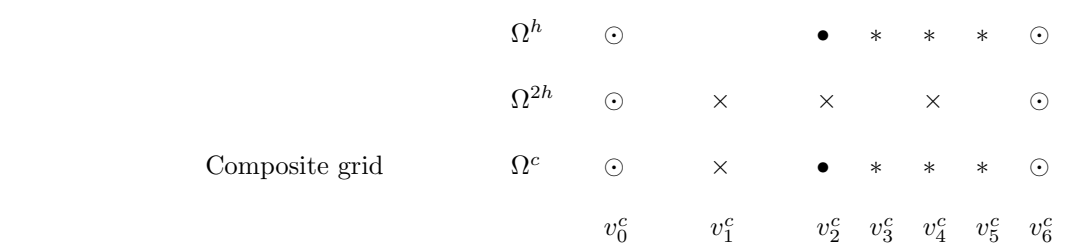

Figure 9.3: *FAC is done entirely on uniform grids, such as the top two grids shown above. However, the solution is in effect determined on the nonuniform composite grid shown at the bottom.*

- local fine-grid points ($*$),
- interface points ($\bullet$), and
- coarse-grid points that do not lie under the local fine-grid or interface points ($\times$ points in $(0, \frac{1}{2})$).

FAC operates only on uniform grids, but it effectively solves problems on compositive grids. It is fully consistent with multigrid principles because, in the variational case, FAC is equivalent to solving the problem by multigrid using global grids, but local relaxation. This property distinguishes FAC from other adaptive methods.

While we never need to construct the composite grid, it serves as an important conceptual tool for understanding the refinement algorithm that we just developed. For example, because FAC is a method for solving the problem on the composite grid, we should be able to derive the associated composite-grid equations. This will allow us to see what algebraic problem FAC actually solves.

To derive these composite-grid equations, first consider the composite-grid approximation $\mathbf{v}^c$ that FAC produces (we include the boundary points v_0^c and v_6^c for completeness):

$$v_0^c = 0\,,\ v_1^c = w_1^{2h}\,,\ v_2^c = v_4^h\,,\ v_3^c = v_5^h\,,\ v_4^c = v_6^h\,,\ v_5^c = v_7^h\,,\ v_6^c = 0\,.$$

We also need to define the source vector $\mathbf{f}^c$. Leaving the definition of f_2^c aside for the moment, we have

$$f_1^c = \frac{f_1^h + 2f_2^h + f_3^h}{4}\,,\ f_3^c = f_5^h\,,\ f_4^c = f_6^h\,,\ f_5^c = f_7^h\,.$$

Now imagine that the FAC scheme has converged, by which we mean that $\mathbf{v}^c$ does not change from one cycle to the next. Inspecting the algorithm shows that the stencils in the uniform regions ($\times$ and $\circ$ points) are the usual ones with the appropriate mesh sizes:

$$\begin{aligned} \frac{-v_0^c + 2v_1^c - v_2^c}{(2h)^2} &= f_1^c\,, \\ \frac{-v_{i-1}^c + 2v_i^c - v_{i+1}^c}{h^2} &= f_i^c\,, \quad 3 \le i \le 5\,. \end{aligned} \tag{9.6}$$

The difficult stencil is the one at the interface point $\bullet = \frac{1}{2}$. Consider (9.5) and remember we are assuming that FAC has converged, so relaxation does not change v_5^h. This means that r_5^h must be zero. Using this observation and substituting (9.2) and (9.3) into (9.5) gives us

$$\begin{aligned} f_2^{2h} &= \frac{1}{4}\left(r_3^h + 2r_4^h\right) \\ &= \frac{1}{4}\left(f_3^h + 2f_4^h\right) - \left(-w_1^{2h} + 3v_4^h - 2v_5^h\right)/(2h)^2 . \end{aligned}$$

We must have $f_2^{2h} = 0$, or else the FAC coarse-grid step would compute a nonzero correction to $\mathbf{v}^c$ and we could not have converged. We thus conclude that the stencil at $\bullet = \frac{1}{2}$ is

$$\frac{-w_1^{2h} + 3v_4^h - 2v_5^h}{(2h)^2} = \frac{1}{4}\left(f_3^h + 2f_4^h\right) .$$

In composite-grid terms, with $f_2^c \equiv \frac{1}{4}\left(f_3^h + 2f_4^h\right)$, we have

$$\frac{-v_1^c + 3v_2^c - 2v_3^c}{(2h)^2} = f_2^c. \tag{9.7}$$

One of the main concerns in the design of any numerical algorithm is how to measure the error. This is a complex issue that involves various error sources (discretization error, algebraic error, and floating-point errors), the choice of norms, and various ways to approximate the error. Most of these issue are beyond the scope of this book, but a few brief comments are in order:

- Estimating discretization error is of particular interest in adaptive methods because it can be used to decide where to refine. There are many conventional approaches that involve solving local problems or estimating higher derivatives of the emerging solution. However, the presence of several levels of discretization enables multilevel methods to use extrapolation: you can compare the approximations on two or more levels to predict the error on the finest level. For this purpose, it is convenient to have the full approximations available on each level, so it can be useful to rewrite the FAC *correction* scheme that we have presented as an FAS scheme (Exercise 1).

- The algebraic error can also be estimated in several ways, but a natural estimate is to apply a norm to the residual of the composite-grid equations. Remember, the composite grid consists of the uniform fine grid in the refinement region, so any point within this region has a standard fine-grid equation. Similarly, points outside the refinement region have standard coarse-grid equations. But the interface has special equations (9.7) that require special residuals. Putting this together for our example gives

$$\begin{aligned} r_1^c &= \frac{1}{4}\left(f_1^h + 2f_2^h + f_3^h\right) - \frac{2w_1^{2h} - v_4^h}{(2h)^2}, \quad \text{standard coarse-grid residual;} \\ r_2^c &= \frac{1}{4}\left(f_3^h + 2f_4^h\right) - \frac{-w_1^{2h} + 3v_4^h - 2v_5^h}{(2h)^2}, \quad \text{interface residual from (9.7);} \\ r_i^c &= f_i^h - \frac{-v_{i+1}^h + 2v_{i+2}^h - v_{i+3}^h}{h^2}, \ 3 \le i \le 5, \ \text{standard fine-grid residual.} \end{aligned}$$

- Scaling the error norms is particularly important in adaptive methods because of the presence of different levels of resolution. The proper scales depend on the discretization and other factors. For our example, you can think of the residual $\mathbf{r}^c$ as a source term that is constant in a neighborhood of each node. Thus, r_1^c represents the value in the interval $(\frac{1}{4}-h\,,\,\frac{1}{4}+h)$, r_2^c represents the value in the interval $(\frac{1}{2}-h\,,\,\frac{1}{2}+\frac{h}{2})$, and r_i^c represents the value in the interval $(\frac{i+2}{8}-\frac{h}{2}\,,\,\frac{i+2}{8}+\frac{h}{2})$ for $i=3,4,5$. (This does not cover the interval $[0,1]$, but we can simply assume that the source is zero in $[0,\,\frac{1}{16})$ and $(\frac{15}{16},1]$.) Using the premise that the residual represents a piecewise constant source term, we can obtain the norm of $\mathbf{r}^c$ by taking the integral of the square of the source term defined on $(0,1)$, which yields

$$\|\mathbf{r}^c\| = \sqrt{2h(r_1^c)^2 + \frac{3}{2}h(r_2^c)^2 + h\sum_3^5 (r_i^c)^2}\,.$$

Two-Dimensional Problems

With some modification, the above ideas can be extended to two-dimensional problems. Figure 9.4 shows a square domain in the plane with a local fine grid in the northeast corner. There are many points in this two-dimensional grid that are analogous to $\circ = \frac{1}{8}$, at which fine-grid computations can be neglected. For this reason,

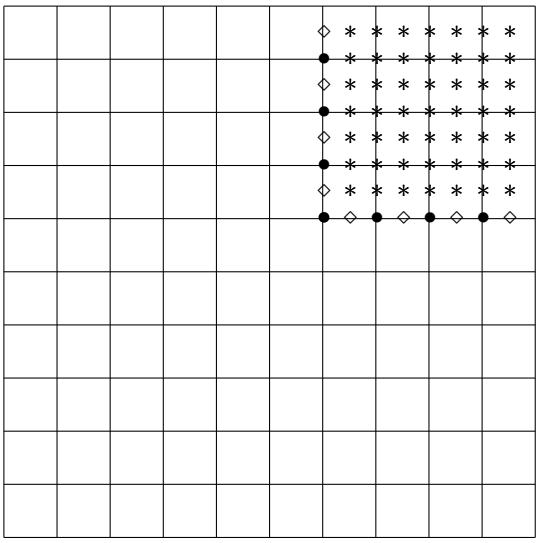

Figure 9.4: *Composite grid on a two-dimensional domain. The coarse-grid points are at the intersection of the solid horizontal and vertical lines. The local fine grid consists of $*$ points. Interface points coinciding with coarse-grid points are marked by $\bullet$. The new slave points ($\diamond$) are interface points that do not overlay a fine-grid point. Boundary points have been omitted.*

the FAC scheme shows substantial savings over standard multigrid on a global fine grid.

A new type of point arises in two-dimensional problems. Denoted by $\diamond$ in Fig. 9.4, the *slave points* are interface points that do not correspond to a coarse-grid point. These points are needed to provide a complete set of boundary points for the local fine grid. They are determined simply by averaging the two neighboring interface point values. For example, in Fig. 9.4, a typical slave point on a horizontal interface would be determined by

$$v^h_{i+1,j} = \frac{v^h_{i,j} + v^h_{i+2,j}}{2}.$$

The one-dimensional FAC method can be generalized in all respects to two dimensions. Special restriction formulas at the interface points corresponding to (9.5) can be used to eliminate border points. Similarly, with some additional work, it is possible to find the coarse-grid and interface equations as was done in one dimension (Exercise 4). However, our goal is not to expose these details (which can be found in [14]). Rather, we will only outline the flow of the two-dimensional FAC method in fairly qualitative terms.

Fast Adaptive Composite Grid Method (FAC) in Two Dimensions

- Initialize $\mathbf{v}^h = \mathbf{0}$. Initialize $\mathbf{w}^{2h} = 0$ and $\mathbf{f}^{2h} = I^{2h}_h \mathbf{f}^h$ at coarse-grid points outside the local fine grid.
- Relax on $\mathbf{v}^h$ on the local fine grid.
- Compute the right sides for the local coarse grid using special stencils at the interface points and the usual residual transfer $\mathbf{f}^{2h} = I^{2h}_h \mathbf{f}^h$ at points underlying the local fine grid.
- Compute an approximation $\mathbf{v}^{2h}$ to the coarse-grid residual equation $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$.
- Update the coarse-grid residual for use in later cycles: $\mathbf{f}^{2h} \leftarrow \mathbf{f}^{2h} - A^{2h}\mathbf{v}^{2h}$ at points outside the local fine grid.
- Update the coarse-grid approximation: $\mathbf{w}^{2h} \leftarrow \mathbf{w}^{2h} + \mathbf{v}^{2h}$ at coarse-grid points outside the local fine grid.
- Interpolate the correction and update the approximation: $\mathbf{v}^h \leftarrow \mathbf{v}^h + I^h_{2h}\mathbf{v}^{2h}$ at the local fine grid and interface points. Interpolate to the slave points from their interface neighbors.

We have presented FAC as a two-grid scheme. Like other multigrid methods, it is done recursively in practice: in the fourth step of the above algorithm, the solution of the global coarse-grid problem is done using further applications of the basic two-grid scheme. This procedure leads to V-cycles defined on a sequence of grids, possibly involving several global coarse grids together with several telescoping finer grids.

An FMG–FAC scheme is more subtle. One could apply a straightforward FMG cycling scheme to the uniform grids that are used in the V-cycle process. However, this would be too costly when there are many levels with few points per level (Exercise 3). A more efficient approach is to use a sequence of increasingly coarse composite grids as the basis for FMG, a strategy that takes us beyond the scope of this book.

The fundamental basis for FAC is the composite grid. In fact, probably the most effective way to apply FAC to a given problem is to *start* by developing accurate composite-grid equations. The FAC algorithm is then a natural extension of multigrid, where the finest level is the composite grid, relaxation is simply restricted to the local fine grid, and coarsening is based on restricting the composite grid residual to the global coarse grid.

A popular alternative to FAC is the *multilevel adaptive technique*, or *MLAT* [3]. Although these two approaches can lead to similar algorithms in the end, the basis for their development is very different. While FAC is based on the composite grid, MLAT comes from generalizing the full approximation scheme (FAS). A little inspection of FAS, as developed in Chapter 6, shows that the finest grid need not be global: one can relax only on a local fine grid and correct the coarse-grid equations by FAS only at points that lie under it. MLAT can thus be developed from a global-grid FAS code, and it often proves effective for handling adaptive refinement.

Numerical example. To illustrate FAC performance in a simple setting, FAC is applied to Poisson's equation with homogeneous Dirichlet boundary conditions on the unit square in the plane. The exact two-spike solution, as shown in Fig. 9.5, has values of magnitude 1 but opposite sign at $(\frac{1}{4}, \frac{1}{4})$ and $(\frac{3}{4}, \frac{3}{4})$, falls off steeply in the neighborhood of these points, and is zero at the boundary. In the tests, we used composite grids constructed from a global grid of mesh size h and refinement

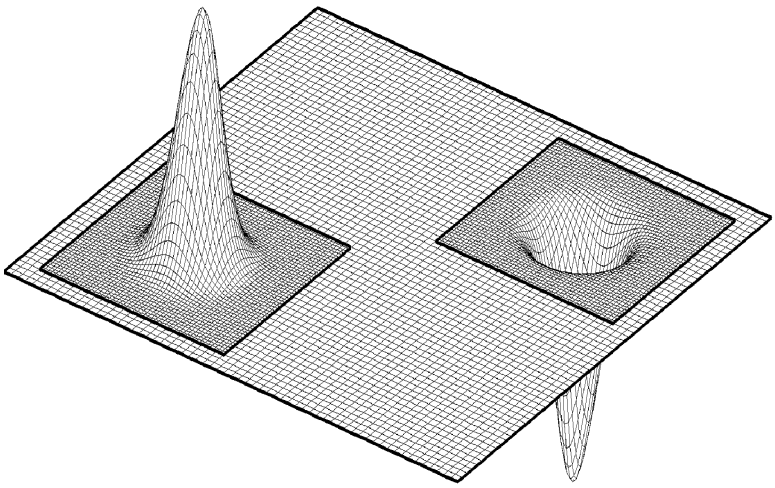

Figure 9.5: *Two-spike solution of the composite grid problem based on a* 63×63 *interior global grid, with mesh size* $h = \frac{1}{64}$*, and* 51×51 *patches of mesh size* $\frac{h}{2} = \frac{1}{128}$ *centered at* $(\frac{1}{4}, \frac{1}{4})$ *and* $(\frac{3}{4}, \frac{3}{4})$.

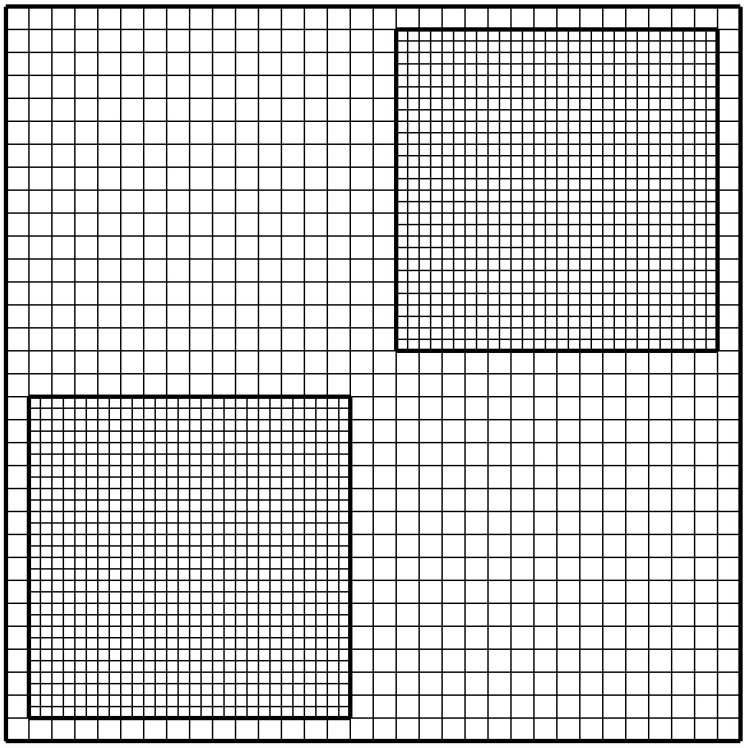

Figure 9.6: *Aerial view of a composite grid based on a* 31×31 *interior global grid of mesh size* $h = \frac{1}{32}$ *and* 27×27 *patches of mesh size* $\frac{h}{2} = \frac{1}{64}$ *centered at* $(\frac{1}{4}, \frac{1}{4})$ *and* $(\frac{3}{4}, \frac{3}{4})$.

patches of size $\frac{h}{2}$ in the subregions $[\frac{1}{32}, \frac{15}{32}] \times [\frac{1}{32}, \frac{15}{32}]$ and $[\frac{17}{32}, \frac{31}{32}] \times [\frac{17}{32}, \frac{31}{32}]$. Tests were done for $h = \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$, with the case $h = \frac{1}{64}$ shown in Fig. 9.6.

Table 9.1 contains the results of applying a V(1,0)-cycle to each of the three composite grids. The convergence factor was obtained by applying many cycles to the homogeneous problem (with zero right side), starting with a random initial guess, and observing the worst-case composite-grid residual reduction factor. Note

Global h	Convergence factor	Discrete L^2 norm of discretization error
1/32	0.362	2.34e − 2
1/64	0.367	5.742 − 3
1/128	0.365	1.43e − 3

Table 9.1: *Estimates for asymptotic* V(1,0)*-cycle convergence factors and discrete* L^2 *norm of the discretization errors. FAC was applied using composite grids with three different global grids of mesh size* h*, each with two local patches of mesh size* $\frac{h}{2}$.

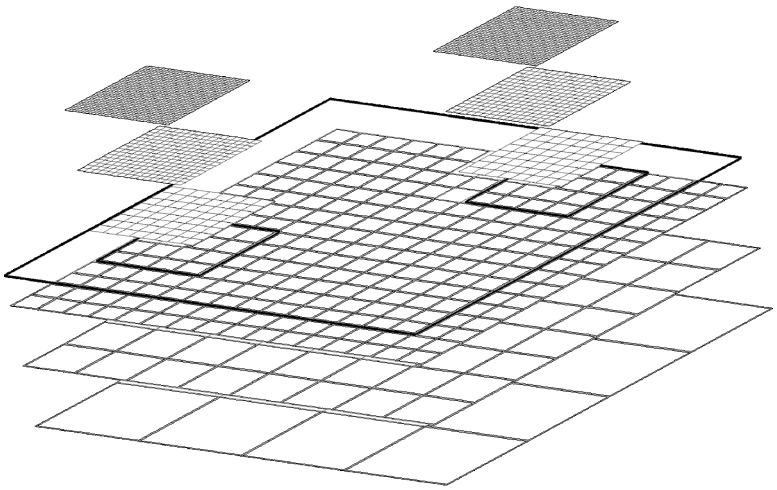

Figure 9.7: *Side view of a global* 15×15 *grid with two coarser supporting grids and three successively finer local grids.*

the apparent h-independence of these convergence factors. These factors are quite acceptable considering the fact that V(1,0)-cycles were used. Also shown are the discretization error estimates, obtained by applying several V(1,0)-cycles to the two-spike problem and measuring the difference, in the discrete L^2 norm, between the computed and the exact solutions, evaluated on the composite grid. Note the apparent $O(h^2)$ behavior of the discretization error.

Another illustration of the grids that comprise the composite grids is given in Fig. 9.7. Here we show a side view of a global grid of mesh size $h = \frac{1}{8}$ and local grids of mesh size $\frac{h}{8} = \frac{1}{64}$ in the subregions $[\frac{1}{8}, \frac{3}{8}] \times [\frac{1}{8}, \frac{3}{8}]$ and $[\frac{5}{8}, \frac{7}{8}] \times [\frac{5}{8}, \frac{7}{8}]$. Note that the effective refinement factor (ratio of mesh sizes of the finest patch and finest global grid) is eight. Large refinement factors are permissible with FAC because local grids with factors of two are included between the global and finest local grids. For similar reasons, the finest global grid is supported by a full sequence of coarser global grids, as shown. $\diamond\diamond$

Exercises

1. **FAS version of FAC.** The FAC algorithm we have described is a correction scheme in which the coarse-grid problem is the residual equation. Rewrite FAC for the one-dimensional model problem as an FAS scheme in which the full approximation equations are solved on the coarse grid.

2. **Higher mesh refinement factors.** We developed the FAC scheme assuming a *mesh refinement factor* of two: the global coarse-grid mesh size is $2h$, while the local fine-grid mesh size is h. Rewrite FAC for solving the one-dimensional model problem when the coarse-grid mesh size is $4h = \frac{1}{4}$ and the fine-grid mesh size is $h = \frac{1}{16}$. Assume that the local fine grid covers $(\frac{1}{2}, 1)$ as

before. (For full efficiency, FAC should include the intermediate grid of mesh size $2h$ that covers $(\frac{1}{2}, 1)$, but the two-grid scheme suffices here.)

3. **Cycling cost.** Show that V-cycles maintain $O(n)$ complexity for locally refined grids in the sense that the cost of an FAC V-cycle is proportional to the number, n, of *composite-grid* points. For concreteness, consider a sequence of q grids on the unit square in the plane, constructed by starting from a global $m \times m$ grid and refining successive grids by a factor of two in the northeast quadrant (so that each grid is $m \times m$). Now show that W-cycles do *not* maintain $O(n)$ complexity. Similarly, show that FMG based on these uniform grids also does not maintain $O(n)$ complexity. Finally, show that FMG based on coarsened *composite* grids does maintain $O(n)$ complexity. (Coarse composite grids are formed by coarsening all grids in the sequence, so the first coarse composite grid is formed from q grids of size $\frac{m}{2} \times \frac{m}{2}$.)

4. **Two-dimensional interface source terms.** Deriving the Ω^{2h} and Ω^c interface equations is somewhat complicated in two dimensions. Take the first step in this direction by forming the source term for Ω^{2h} and Ω^c at the interface corner point in Fig. 9.4.

5. **Symmetry.** Show that the composite-grid matrix A^c is nonsymmetric by showing that the coefficient in (9.6) that connects v_2^c to v_3^c is not equal to the corresponding coefficient in (9.7). Show, however, that symmetry can be retained by rescaling: the matrix $D^c A^c$ is symmetric, where D^c is the 5×5 diagonal matrix with entries $d_1^c = d_2^c = 2h$ and $d_3^c = d_4^c = d_5^c = h$.

6. **Neumann problem.** Rederive FAC applied to Neumann problem (7.1) of Chapter 7. Show that the principles of Neuman boundary conditions apply by showing that the null space and range of the rescaled composite-grid matrix, $D^c A^c$ of Exercise 5, are equal and consist of scalar multiples of the constant vector $\mathbf{1}$.