Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Chapter 8

Algebraic Multigrid (AMG)

A natural question arises: Can we apply multigrid techniques when there is no *grid*? Suppose we have relationships among the unknowns that are similar to those in the model problem, but the physical locations of the unknowns are themselves unknown (or immaterial). Can we hope to apply the tools we have developed? A related question is: Can we apply multigrid in the case where grid locations are known but may be highly unstructured or irregular, making the selection of a coarse grid problematic? These are the problems that are addressed by a technique known as *algebraic multigrid*, or AMG [6]. We describe AMG using many of the concepts and principles developed in [18]. For a theoretical foundation of general algebraic methods, of which AMG is a part, see [5].

For any multigrid algorithm, the same fundamental components are required. There must be a sequence of grids, intergrid transfer operators, a relaxation (smoothing) operator, coarse-grid versions of the fine-grid operator, and a solver for the coarsest grid.

Let us begin by deciding what we mean by a *grid*. Throughout this chapter, we look to standard multigrid (which we refer to as the *geometric case*) to guide us in defining AMG components. In the geometric case, the unknown variables u_i are defined at known spatial locations (grid points) on a fine grid. We then select a subset of these locations as a coarse grid. As a consequence, a subset of the variables u_i is used to represent the solution on the coarse grid. For AMG, by analogy, we seek a subset of the variables u_i to serve as the coarse-grid unknowns. A useful point of view, then, is to identify the grid points with the indices of the unknown quantities. Hence, if the problem to be solved is $A\mathbf{u} = \mathbf{f}$ and

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix},$$

then the fine-grid points are just the indices $\{1, 2, \ldots, n\}$.

Having defined the grid points, the connections within the grid are determined by the *undirected adjacency graph* of the matrix A. Letting the entries of A be a_{ij}, we associate the vertices of the graph with the grid points and draw an edge between the ith and jth vertices if either $a_{ij} \neq 0$ or $a_{ji} \neq 0$. The connections in

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

$$A = \begin{bmatrix} X & X & & X & X & \\ X & X & X & X & & \\ & X & X & X & X & X \\ X & X & X & X & & \\ X & & X & & X & X \\ & & X & & X & X \end{bmatrix}$$

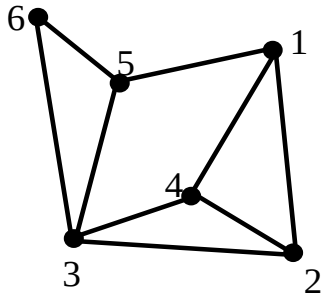

Figure 8.1: *The nonzero structure of A, where X indicates a nonzero entry, is shown on the left. The resulting undirected adjacency graph appears on the right.*

the grid are the edges in the graph; hence, the grid is entirely defined by the matrix A. A simple example of this relationship is given in Fig. 8.1.

Now that we can represent the fine grid, how do we select a coarse grid? With standard multigrid methods, smooth functions are *geometrically* or physically smooth; they have a low spatial frequency. In these cases, we assume that relaxation smooths the error and we select a coarse grid that represents smooth functions accurately. We then choose intergrid operators that accurately transfer smooth functions between grids.

With AMG, the approach is different. We first select a relaxation scheme that allows us to determine the nature of the smooth error. Because we do not have access to a physical grid, the sense of smoothness must be defined algebraically. The next step is to use this sense of smoothness to select coarse grids, which will be subsets of the unknowns. A related issue is the choice of intergrid transfer operators that allow for effective coarsening. Finally, we select the coarse-grid versions of the operator A, so that coarse-grid correction has the same effect that it has in geometric multigrid: it must eliminate the error components in the range of the interpolation operator.

Algebraic Smoothness

Having chosen a relaxation scheme, the crux of the problem is to determine what is meant by smooth error. If the problem provides no geometric information (for example, true grid point locations are unknown), then we cannot simply examine the Fourier modes of the error. Instead, we must proceed by analogy. In the geometric case, the most important property of smooth error is that it is not effectively reduced by relaxation. Thus, we now *define* smooth error loosely to be any error that is not reduced effectively by relaxation.

That was simple. Of course, we still need to figure out exactly what this definition means. To do this in the simplest case, we focus on weighted point Jacobi relaxation. We also assume that A is a symmetric *M-matrix*: it is symmetric ($A^T = A$) and positive-definite ($\mathbf{u}^T A\mathbf{u} > 0$ for all $\mathbf{u} \neq \mathbf{0}$) and has positive diagonal entries and nonpositive off-diagonal entries. These properties are shared by matrices arising from the discretization of many (not all) scalar elliptic differential equations. These assumptions are not necessary for AMG to work. However, the

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

original theory of AMG was developed for symmetric M-matrices, and if A is far from being an M-matrix, it is less likely that standard AMG will be effective in solving the problem.

Recall from Chapter 2 that the weighted point Jacobi method can be expressed as

$$\mathbf{v} \leftarrow \mathbf{v} + \omega D^{-1}(f - A\mathbf{v}),$$

where D is the diagonal of A. As in previous chapters, $\mathbf{v}$ is a computed approximation to the exact solution $\mathbf{u}$. Remember that the error propagation for this iteration can be written as

$$\mathbf{e} \leftarrow \left(I - \omega D^{-1}A\right)\mathbf{e}. \tag{8.1}$$

Weighted Jacobi relaxation, as we know, has the property that after making great progress toward convergence, it stalls, and little improvement is made with successive iterations. At this point, we define the error to be *algebraically smooth*. It is useful to examine the implications of algebraic smoothness. Because all of AMG is based on this concept, the effort is worthwhile.

By our definition, algebraic smoothness means that the size of $\mathbf{e}^{i+1}$ is not significantly less than that of $\mathbf{e}^i$. We need to be more specific about the concept of size. A natural choice is to measure the error in the A-norm, which is induced by the A-inner product. As defined in Chapter 5, we have

$$\|\mathbf{e}\|_A = (A\mathbf{e}, \mathbf{e})^{1/2}.$$

Using this norm and (8.1), we see that an algebraically smooth error is characterized by

$$\|\left(I - \omega D^{-1}A\right)\mathbf{e}\|_A \approx \|\mathbf{e}\|_A.$$

When we assume that $\omega = \alpha\|D^{-1/2}AD^{-1/2}\|^{-1}$ for some fixed $\alpha \in (0, 2)$ and that $\|D^{-1/2}AD^{-1/2}\|$ is $O(1)$ (for the model problem, it is bounded by 2), it can be shown (Exercise 1) that

$$(D^{-1}A\mathbf{e}, A\mathbf{e}) \ll (\mathbf{e}, A\mathbf{e}).$$

Writing this expression in components yields

$$\sum_{i=1}^{n} \frac{r_i^2}{a_{ii}} \ll \sum_{i=1}^{n} r_i e_i.$$

This implies that, *at least on average*, algebraically smooth error $\mathbf{e}$ satisfies

$$|r_i| \ll a_{ii}|e_i|.$$

We write this condition loosely as

$$A\mathbf{e} \approx \mathbf{0} \tag{8.2}$$

and read it as meaning that smooth error has relatively small residuals. We will appeal to this condition in the development of the AMG algorithm. While our analysis here is for weighted Jacobi, Gauss–Seidel relaxation is more commonly used for AMG. A similar, though slightly more complicated analysis can be performed for Gauss–Seidel relaxation and also leads to condition (8.2) (Exercise 11).

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

One immediate implication of (8.2) is that $r_i \approx 0$, so

$$a_{ii}e_i \approx -\sum_{j \neq i} a_{ij}e_j; \tag{8.3}$$

that is, if $\mathbf{e}$ is a smooth error, then e_i can be approximated well by a weighted average of its neighbors. This fact gives us an important foothold in determining an interpolation operator.

A short digression here should serve to clarify the difference between algebraic and geometric smoothness. We consider a simple example due to Stüben in his introduction to AMG [22]. Suppose the problem

$$-au_{xx} - cu_{yy} + bu_{xy} = f(x, y), \tag{8.4}$$

with homogeneous Dirichlet boundary conditions, is discretized on the unit square using a uniform grid and the finite-difference stencils

$$D^h_{xx} = \frac{1}{h^2}\begin{pmatrix} 1 & -2 & 1 \end{pmatrix}, \qquad D^h_{yy} = \frac{1}{h^2}\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix},$$

$$D^h_{xy} = \frac{1}{2h^2}\begin{pmatrix} -1 & 1 & \\ 1 & -2 & 1 \\ & 1 & -1 \end{pmatrix}. \tag{8.5}$$

Also suppose that the coefficients a, b, and c are locally constant, but have different values in different quadrants of the domain. Specifically, let the coefficients be defined in the square domain as shown below:

$a = 1$ $c = 1000$ $b = 0$	$a = 1$ $c = 1$ $b = 2$
$a = 1$ $c = 1$ $b = 0$	$a = 1000$ $c = 1$ $b = 0$

Note that this discretization does not produce an M-matrix.

Using a zero right side and a random initial guess, the norm of the error essentially stops changing after eight sweeps of Gauss–Seidel. By our definition, this error is algebraically smooth. However, it does not appear to be smooth in the geometric sense (Fig. 8.2). In fact, in three of the four quadrants it is geometrically quite oscillatory! But because the iteration has stalled, this is precisely the error AMG must account for in coarse-grid correction. We return to this example later.

Influence and Dependence

Most of AMG rests on two fundamental concepts. We have just discussed the first concept, namely, smooth error. The second important concept is that of *strong dependence* or *strong influence*. Because of the dominance of the diagonal entry (A is an M-matrix), we associate the ith equation with the ith unknown; the job of the ith equation is to determine the value of u_i. Of course, it usually takes all of

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

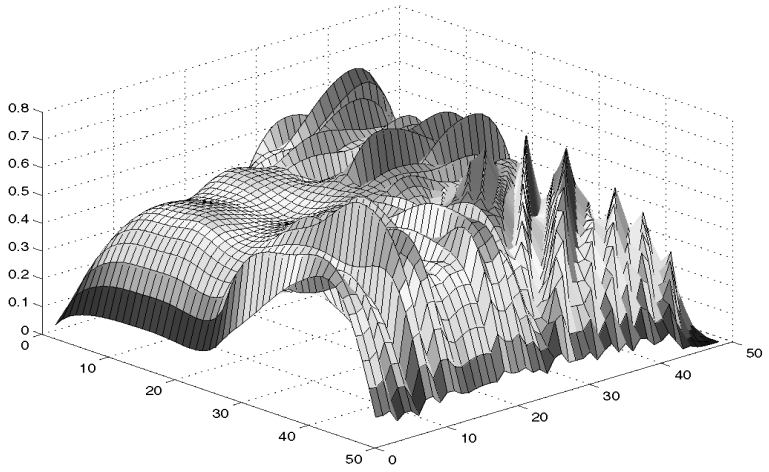

Figure 8.2: *Error that is algebraically smooth, but not geometrically smooth for* (8.4).

the equations to determine any given variable precisely. Nevertheless, our first task is to determine which *other* variables are most important in the ith equation; that is, which u_j are most important in the ith equation in determining u_i?

One answer to this question lies in the following observation: if the coefficient, a_{ij}, which multiplies u_j in the ith equation, is large relative to the other coefficients in the ith equation, then a small change in the value of u_j has more effect on the value of u_i than a small change in other variables in the ith equation. Intuitively, it seems logical that a variable whose value is instrumental in determining the value for u_i would be a good value to use in the interpolation of u_i. Hence, such a variable (point) should be a candidate for a coarse-grid point. This observation suggests the following definition.

> **Definition 1.** Given a threshold value $0 < \theta \leq 1$, the variable (point) u_i *strongly depends* on the variable (point) u_j if
>
> $$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\} . \tag{8.6}$$

This says that grid point i strongly depends on grid point j if the coefficient a_{ij} is comparable in magnitude to the largest off-diagonal coefficient in the ith equation. We can state this definition from another perspective.

> **Definition 2.** If the variable u_i strongly depends on the variable u_j, then the variable u_j *strongly influences* the variable u_i.

With the twin concepts of smooth error and strong influence/dependence in hand, we can return to the task of defining the multigrid components for AMG. As with any multigrid algorithm, we begin by defining a two-grid algorithm, then proceed to multigrid by recursion. Having defined the relaxation scheme, we have several tasks before us:

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

- select a coarse grid so that the smooth components can be represented accurately;
- define an interpolation operator so that the smooth components can be accurately transferred from the coarse grid to the fine grid; and
- define a restriction operator and a coarse-grid version of A using the variational properties.

Defining the Interpolation Operator

Assume for the moment that we have already designated the coarse-grid points. This means that we have a partitioning of the indices $\{1, 2, \ldots, n\} = C \cup F$, where the variables (points) corresponding to $i \in C$ are the coarse-grid variables. These coarse-grid variables are also fine-grid variables; the indices $i \in F$ represent those variables that are *only* fine-grid variables. Next, suppose that $e_i, i \in C$, is a set of values on the coarse grid representing a smooth error that must be interpolated to the fine grid, $C \cup F$. What do we know about e_i that allows us to build an interpolation operator that is accurate? With geometric multigrid, we use linear interpolation between the coarse grid points. With an unstructured, or perhaps nonexistent, grid, the answer is not so obvious.

If a C-point j strongly influences an F-point i, then the value e_j contributes heavily to the value of e_i in the ith (fine-grid) equation. It seems reasonable that the value e_j in the coarse-grid equation could therefore be used in an interpolation formula to approximate the fine-grid value e_i. This idea can be strengthened by noting that the following bound must hold for smooth error on average, that is, for most i (Exercise 2):

$$\sum_{j \neq i} \left(\frac{|a_{ij}|}{a_{ii}} \right) \left(\frac{e_i - e_j}{e_i} \right)^2 \ll 1, \quad 1 \leq i \leq n. \tag{8.7}$$

The left side of the inequality is a sum of products of nonnegative terms. These products must be very small, which means that one or both of the factors in each product must be small. But if e_i strongly depends on e_j, we know that $-a_{ij}$ could be comparable to a_{ii}. Therefore, for these strongly influencing e_j's, it must be true that $e_i - e_j$ is small; that is, $e_j \approx e_i$. We describe this by saying that *smooth error varies slowly in the direction of strong connection*. Thus, we have a justification for the idea that the fine-grid quantity u_i can be interpolated from the coarse-grid quantity u_j if i strongly depends on j.

For each fine-grid point i, we define N_i, the *neighborhood of* i, to be the set of all points $j \neq i$ such that $a_{ij} \neq 0$. These points can be divided into three categories:

- the neighboring coarse-grid points that strongly influence i; this is the *coarse interpolatory set* for i, denoted by C_i;
- the neighboring fine-grid points that strongly influence i, denoted by D_i^s; and
- the points that do not strongly influence i, denoted by D_i^w; this set may contain both coarse- and fine-grid points; it is called the set of *weakly connected neighbors*.

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

The goal is to define the *interpolation* operator I_{2h}^h (although physical grids may not be present, we continue to denote fine-grid quantities by h and coarse-grid quantities by $2h$). We require that the ith component of $I_{2h}^h \mathbf{e}$ be given by

$$\left(I_{2h}^h \mathbf{e}\right)_i = \begin{cases} e_i & \text{if } i \in C, \\ \displaystyle\sum_{j \in C_i} \omega_{ij} e_j & \text{if } i \in F, \end{cases} \tag{8.8}$$

where the interpolation weights, ω_{ij}, must now be determined.

Recall that the main characteristic of smooth error is that the residual is small: $\mathbf{r} \approx \mathbf{0}$. We can write the ith component of this condition as

$$a_{ii} e_i \approx - \sum_{j \in N_i} a_{ij} e_j.$$

Splitting the sum into its component sums over the coarse interpolatory set, C_i, the fine-grid points with strong influence, D_i^s, and the weakly connected neighbors, D_i^w, we have

$$a_{ii} e_i \approx - \sum_{j \in C_i} a_{ij} e_j - \sum_{j \in D_i^s} a_{ij} e_j - \sum_{j \in D_i^w} a_{ij} e_j. \tag{8.9}$$

To determine the ω_{ij}, we need to replace the e_j in the second and third sums on the right side of (8.9) with approximations in terms of e_i or e_j, where $j \in C_i$.

Consider the third sum over points that are weakly connected to point i. We distribute these terms to the diagonal coefficient; that is, we simply replace e_j in the rightmost sum by e_i, giving

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i \approx - \sum_{j \in C_i} a_{ij} e_j - \sum_{j \in D_i^s} a_{ij} e_j. \tag{8.10}$$

We can justify this distribution in the following way. Suppose we have underestimated the dependence, so that e_i does depend strongly on the value of the points in D_i^w. Then the fact that smooth error varies *slowly* in the direction of strong dependence means that $e_i \approx e_j$ and the distribution to the diagonal makes sense. Alternatively, suppose the value of e_i does not depend strongly on the points in D_i^w. Then the corresponding value if a_{ij} will be small and any error committed in making this assignment will be relatively insignificant.

Treating the second sum over D_i^s is a bit more complicated because we must be more careful with these strong connections. We might simply distribute these terms to the diagonal, and, indeed, this would work nicely for many problems. However, experience has shown that it is better to distribute the terms in D_i^s to C_i. Essentially, we want to approximate the e_j's in this sum with weighted sums of the e_k for $k \in C_i$. That is, we want to replace each e_j, where j is a fine-grid point that strongly influences i, with a linear combination of values of e_k from the coarse interpolatory set of the point i. We do this, for each fixed $j \in D_i^s$, by making the approximation

$$e_j \approx \frac{\displaystyle\sum_{k \in C_i} a_{jk} e_k}{\displaystyle\sum_{k \in C_i} a_{jk}}. \tag{8.11}$$

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

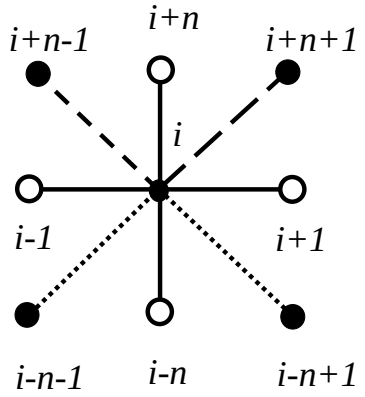

Figure 8.3: *Strong and weak influences on the point i. Coarse-grid points are shown as open circles. C-points with a strong influence on i are indicated with solid lines, F-points that strongly influence i with dashed lines, and F-points that weakly influence i with dotted lines.*

The numerator is appropriate because the e_j are strongly influenced by the e_k in proportion to the matrix entries a_{jk}. The denominator is chosen to ensure that the approximation interpolates constants exactly. Notice that this approximation *requires* that if i and j are any two strongly connected fine-grid points, then they must have at least one point common to their coarse interpolatory sets C_i and C_j.

If we now substitute (8.11) into (8.10) and engage in a spate of algebra (Exercise 3), we find that the interpolation weights are given by

$$\omega_{ij} = -\frac{a_{ij} + \displaystyle\sum_{m \in D_i^s} \left(\frac{a_{im} a_{mj}}{\displaystyle\sum_{k \in C_i} a_{mk}} \right)}{a_{ii} + \displaystyle\sum_{n \in D_i^w} a_{in}}. \tag{8.12}$$

The calculation of the interpolation weights can be illustrated with a simple example. Consider the operator A, defined on a uniform $n \times n$ grid, by the stencil

$$\begin{bmatrix} -\frac{1}{2} & -2 & -\frac{1}{2} \\ -1 & \frac{29}{4} & -1 \\ -\frac{1}{8} & -2 & -\frac{1}{8} \end{bmatrix}.$$

Assume that the partition of the grid into the C- and F-points corresponds to *red-black* coarsening. For a typical interior fine-grid point i, the four points directly north, south, east, and west are the coarse interpolatory set C_i. Using $\theta = 0.2$ as the dependence threshold, the strong and weak influences on i are shown in Fig. 8.3.

The points northwest and northeast of i form the set D_i^s (fine-grid points with strong influence), while the points to the southeast and southwest form the set D_i^w

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

(points with weak influence). For this example, (8.9) becomes

$$\frac{29}{4}e_i = \underbrace{2e_{i+n} + 2e_{i-n} + e_{i-1} + e_{i+1}}_{C_i} + \underbrace{\frac{1}{2}u_{i+n-1} + \frac{1}{2}e_{i+n+1}}_{D_i^s} + \underbrace{\frac{1}{8}e_{i-n-1} + \frac{1}{8}e_{i-n+1}}_{D_i^w}.$$

Substituting e_i for the D_i^w points, e_{i-n+1} and e_{i-n-1}, in the rightmost sum yields

$$\left(\frac{29}{4} - \frac{1}{8} - \frac{1}{8}\right) e_i = \underbrace{2e_{i+n} + 2e_{i-n} + e_{i-1} + e_{i+1}}_{C_i \text{ points}} + \underbrace{\frac{1}{2}u_{i+n-1} + \frac{1}{2}e_{i+n+1}}_{D_i^s}. \quad (8.13)$$

Point i depends strongly on $i+n-1$, which is itself strongly dependent on points $i-1$ and $i+n$ in C_i. Similarly, i depends strongly on $i+n+1$, which in turn depends strongly on $i+1$ and $i+n$ from the coarse interpolatory set C_i. Using (8.11), we obtain the approximations

$$e_{i+n-1} \approx \frac{-2e_{i-1} - e_{i+n}}{-(2+1)}, \qquad e_{i+n+1} \approx \frac{-2e_{i+1} - e_{i+n}}{-(2+1)},$$

which we can substitute into the sum over the strong F-points in (8.13). After all the algebra is done, (8.13) reduces to the interpolation formula

$$e_i = \frac{7}{21}e_{i+n} + \frac{6}{21}e_{i-n} + \frac{4}{21}e_{i+1} + \frac{4}{21}e_{i-1}.$$

It is worth noting that, as in many good interpolation formulas, the coefficients are nonnegative and sum to one. A worthwhile exercise is to show that under appropriate conditions, this must be the case (Exercise 4).

Selecting the Coarse Grid

The preceding discussion of the interpolation operator assumed that we had already designated points of the coarse grid. We must now turn our attention to this critical task. We use the twin concepts of strong influence/dependence and smooth error, just as we did in defining interpolation. As in the geometric problem, we rely on the fundamental premise that the coarse grid must be one

- on which smooth error can be approximated accurately,
- from which smooth functions can be interpolated accurately, and
- that has substantially fewer points than the fine grid, so that the residual problem may be solved with relatively little expense.

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

The basic idea is straightforward. By examining the suitability of each grid point to be a point of one of the C_i sets, we make an initial partitioning of the grid points into C- and F-points. Then, as the interpolation operator is constructed, we make adjustments to this partitioning, changing points initially chosen as F-points to be C-points in order to ensure that the partitioning conforms to certain heuristic rules.

Before we can describe the coarsening process in detail, we need to make two more definitions and to introduce these heuristics. Denote by S_i the set of points that strongly influence i; that is, the points on which the point i strongly depends. Also denote by S_i^T the set of points that strongly depend *on* the point i. Armed with these definitions, we describe two heuristic criteria that guide the initial selection of the C-points:

H-1: For each F-point i, every point $j \in S_i$ that strongly influences i either should be in the coarse interpolatory set C_i or should strongly depend on at least one point in C_i.

H-2: The set of coarse points C should be a maximal subset of all points with the property that no C-point strongly depends on another C-point.

To motivate heuristic **H-1**, we examine the approximation (8.11) that was made in developing the interpolation formula. Recall that this approximation applies to points $j \in D_i^s$ that consist of F-points strongly influencing the F-point i. Because e_i depends on these points, their values must be represented in the interpolation formula in order to achieve accurate interpolation. But because they have not been chosen as C-points, they are represented in the interpolation formula only by distributing their values to points in C_i using (8.11). It seems evident that (8.11) will be more accurate if j is strongly dependent on several points in C_i. However, for the approximation to be made at all, j must be strongly dependent on at least one point in C_i. Heuristic **H-1** simply ensures that this occurs.

Heuristic **H-2** is designed to strike a balance on the size of the coarse grid. Multigrid efficiency is generally controlled by two properties: convergence factor and number of WUs per cycle. If the coarse grid is a large fraction of the total points, then the interpolation of smooth errors is likely to be very accurate, which, in turn, generally produces better convergence factors. However, relatively large coarse grids generally mean a prohibitively large amount of work in doing V-cycles. By requiring that no C-point strongly depends on another, **H-2** controls the size of the coarse grid because C-points tend to be farther apart. By requiring C to be a maximal subset (that is, no other point can be added to C without violating the ban on mutual strong dependence), **H-2** ensures that C is big enough to produce good convergence factors.

It is not always possible to enforce both **H-1** and **H-2** (see Exercise 5). Because the interpolation formula depends on **H-1** being satisfied, we choose to enforce **H-1** rigorously, while using **H-2** as a guide. While this choice may lead to larger coarse grids than necessary, experience shows that this trade-off between accuracy and expense is generally worthwhile.

The basic coarse-point selection algorithm proceeds in two passes. We first make an initial coloring of the grid points by choosing a preliminary partition into C- and F-points. The goal in the first pass is to create a set of C-points that have good approximation properties and also tend to satisfy **H-2**. Once the initial

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

assignments have been made, we make a second pass, changing initial F-points to C-points as necessary to enforce **H-1**.

The Coloring Scheme

The first pass begins by assigning to each point i a measure of its potential quality as a C-point. There are several ways we can make this assessment, but the simplest is to count the number of other points strongly influenced by i. Because those points are the members of S_i^T, this count, λ_i, is the cardinality of S_i^T. Once the measures λ_i have been determined, we select a point with maximum λ_i value as the first point in C.

The point we just selected strongly influences several of the other points and should appear in the interpolation formula for each of them. This implies that the points that depend strongly on i should become F-points. We therefore assign all points in S_i^T to F, which is permissible because we already have a C-point, i, that strongly influences them. It is logical to look at *other* points that strongly influence these new F-points as potential C-points, because their values could be useful for accurate interpolations. Therefore, for each new F-point j in S_i^T, we increment the measure, λ_k, of each unassigned point k that strongly influences j; this would be each unassigned member of $k \in S_j$.

The process is then repeated. A new unassigned point i is found with maximum λ_i and it is assigned to C. The unassigned points $j \in S_i^T$ are then assigned to F and the measures of the unassigned points in S_j are incremented by 1. This process continues until all points have been assigned to C or F.

It is useful to observe that the coarsening determined by this method depends on several factors. Among the most influential is the order in which the grid points are scanned when seeking the next point with maximal λ. Because many, if not most, of the grid points will have the maximal value at the start, any of them could be selected as the first coarse point. Once the first point is selected, the rest proceeds as outlined. Again, any time there is more than one point having the maximal value, there are many possible coarsenings. The heuristics ensure that whatever specific coarse grid is obtained, it will have the desired properties: it provides a good representation of smooth error components, while keeping the size of the coarse grid reasonably small.

This coloring algorithm is best illustrated by an example. The upper left drawing in Fig. 8.4 shows the graph of the matrix for a nine-point stencil representing the Laplacian operator on a uniform grid. The operator stencil is

$$\frac{1}{h^2}\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}. \tag{8.14}$$

For this example, the dependence threshold is immaterial; for any θ, every connection is one of strong dependence. Hence, each point strongly influences, and depends strongly upon, each of its neighbors. Initially, all interior points have a measure of $\lambda = 8$, all points along the side of the grid have a measure of $\lambda = 5$, and the four corner points of the grid have a measure of $\lambda = 3$. We also assume the points are stored in lexicographic order.

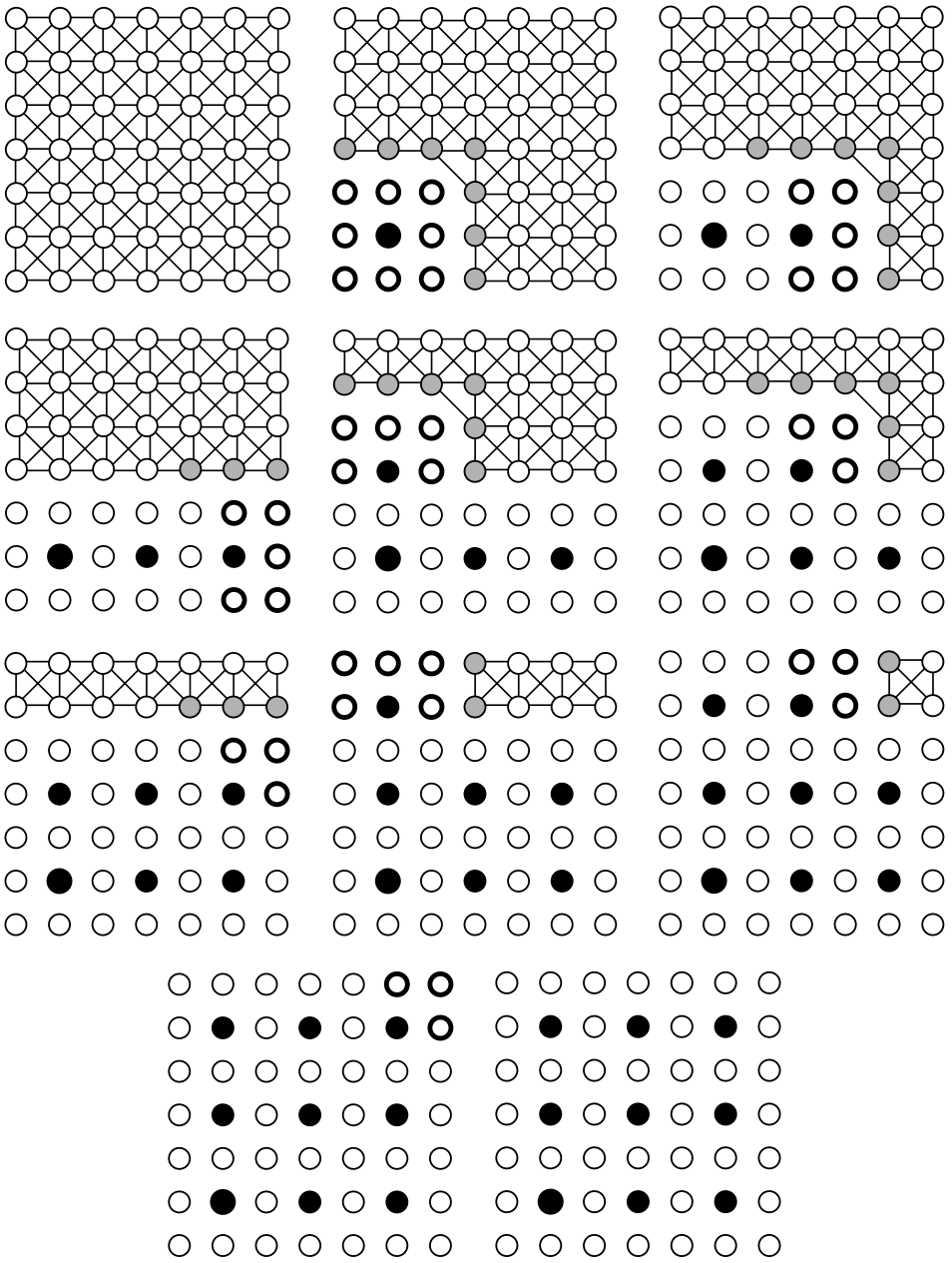

Figure 8.4: *Sequence of coloring steps for the nine-point Laplacian on a uniform grid. The upper left diagram is the original grid, the lower right the final coloring.*

The remaining drawings in Fig. 8.4 show the coloring of the grid as it evolves. At each step, newly designated C-points are shown in black; newly designated F-points are shown in white with a heavy border; and "undecided" neighbors of the new F-points whose λ values are updated are shaded in gray. Edges of the graph are removed as the algorithm accounts for the dependencies of the new C- and F-points. The lower right drawing of the figure displays the completed coloring.

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

It is useful to observe that for this example, coarse-grid selection is complete after this pass. A careful examination of the lower right drawing in the figure reveals that, because of the high connectivity of the graph, both **H-1** and **H-2** are satisfied by the C/F coloring produced. In addition, the coarse grid produced is exactly the standard full coarsening that one would use for a geometric multigrid method for this problem.

It is also instructive to examine the application of this coarsening to the five-point Laplacian stencil on a uniform grid. Once again, the first pass of the process not only produces a coloring satisfying both heuristics, but one that is very common in geometric solutions to the problem as well (Exercise 6).

It is not difficult to concoct an example that does not work so well. One example is illustrated in Fig. 8.5. Again, the nine-point Laplacian stencil (8.14) is used on an $n \times n$ uniform grid, but now we add the periodic boundary conditions $u_{i,j} = u_{i\pm n, j\pm n}$ with $n = 7$. Note that the initial measure is $\lambda = 8$ for all grid points. The thin lines indicate the extent of the grid with the first periodic replication shown outside the lines. At each step, newly designated C-points are shown in black; newly designated F-points are shown in white with a heavy border; and "undecided" neighbors of the new F-points whose λ values are updated are shaded in gray. Edges of the graph are removed as the algorithm accounts for the dependencies of the new C- and F-points. The final first-pass coloring (lower right of figure) violates **H-1**, with a large number of F-F dependencies between points not sharing a C-point.

This is an example of a problem for which it is impossible to satisfy both **H-1** and **H-2**. Because the coloring algorithm satisfies **H-2** and we have determined that we must satisfy **H-1**, a second pass for the coarsening algorithm must be done. In this pass, we examine each of the F-points in turn and determine if there are F-F dependencies with points not depending strongly on a common C-point. If this is the case, then one of the two F-points is tentatively changed into a C-point, and the examination moves on to the next F-point. When further F-F dependencies are encountered, we first attempt to satisfy **H-1** using points on the list of tentative points. The idea is to satisfy **H-1** by converting a minimal number of F-points into C-points.

Figure 8.6 displays the coarsening produced for the periodic nine-point Laplacian after the second pass of the coloring algorithm. The extra C-points are shaded black with a rim of dark gray. It can be seen that while the coloring now satisfies **H-1**, it no longer satisfies **H-2**. The coarse grid has a few more points, but the interpolation formula can be built for each F-point. The details of the second-pass algorithm are left as an exercise (Exercise 7).

Two examples highlight some important features of the coarsening algorithm. First, consider the problem

$$-u_{xx} - u_{yy} = f(x, y), \tag{8.15}$$

with homogeneous Dirichlet boundary conditions. Suppose (8.15) is discretized by finite elements using regular quadrilaterals of dimension $h_x \times h_y$. If $h_y/h_x \to 0$, then the finite element stencil approaches

$$\begin{pmatrix} -1 & -4 & -1 \\ 2 & 8 & 2 \\ -1 & -4 & -1 \end{pmatrix}.$$

While this stencil does not correspond to an M-matrix, the AMG coarsening algo-

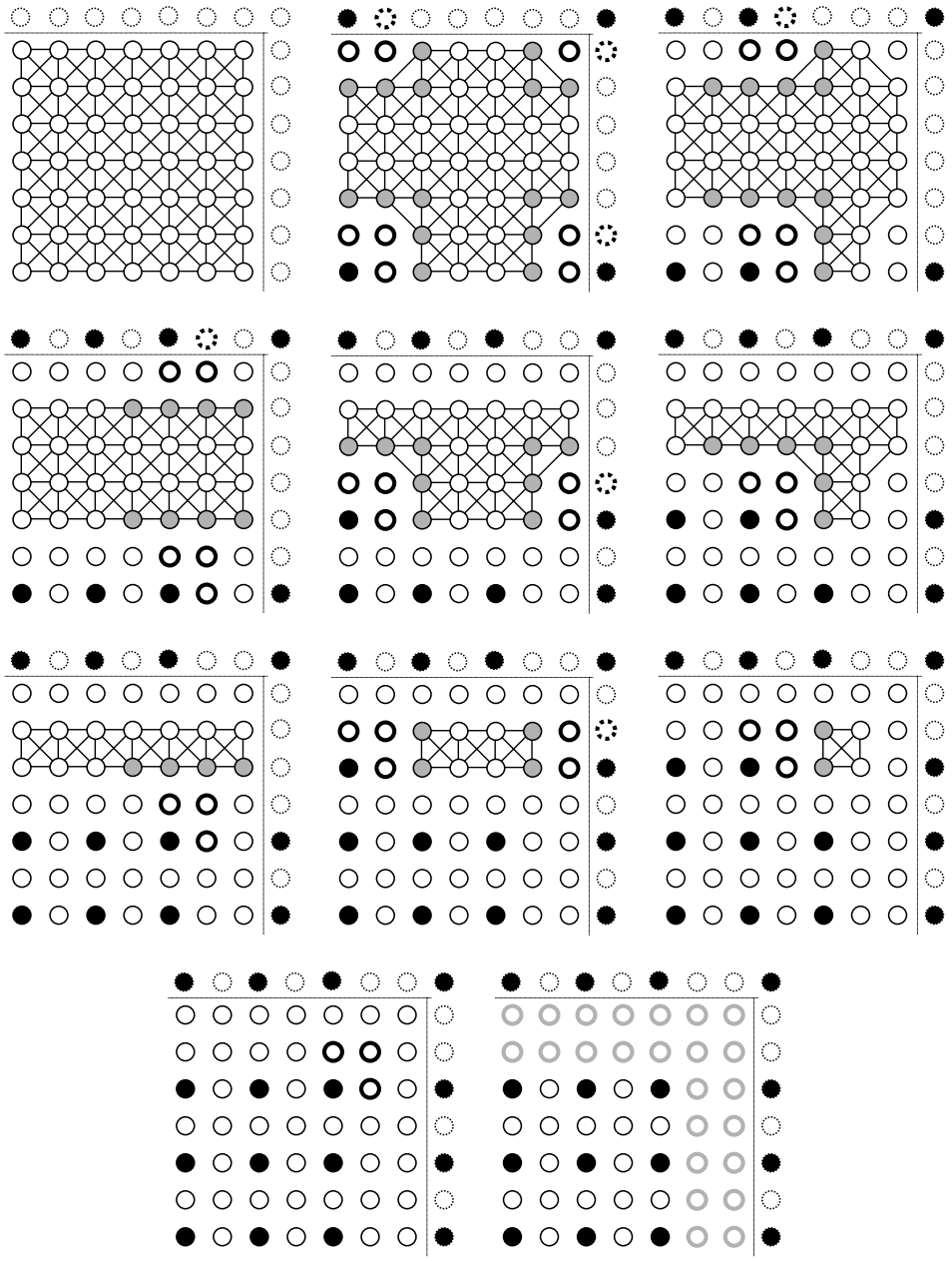

Figure 8.5: *Sequence of coloring steps for the nine-point Laplacian on a uniform grid with periodic boundary conditions. The upper left diagram is the original grid, the lower center the final coloring (first coloring pass). The gray circles in the lower right diagram are points that violate heuristic* **H-1**.

rithm performs quite well on this problem and is very instructive. Note that the problem is essentially equivalent to the discretization of

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

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

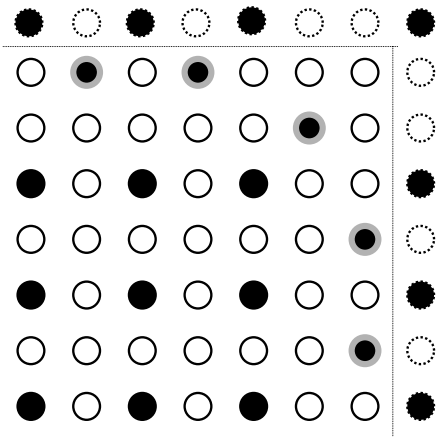

Figure 8.6: *Second coloring pass for the nine-point Laplacian problem with periodic boundary conditions. The added C-points are shown in the final coloring as black dots with thick gray outlines.*

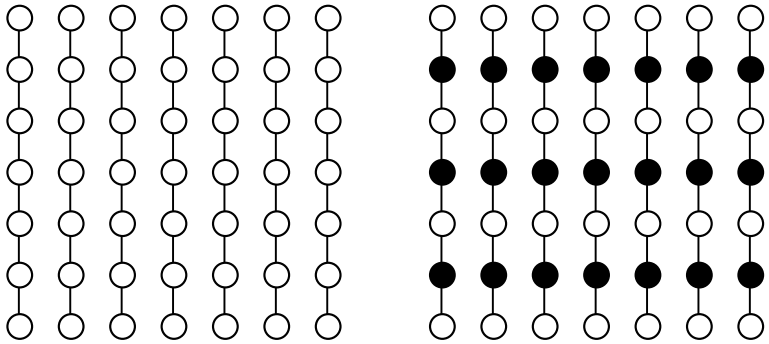

Figure 8.7: *Semi-coarsening produced for a stretched mesh (or anisotropic) problem. Shown on the left is the grid with only the strong dependencies indicated. On the right, the final coarsening is displayed.*

where ϵ is very small. The important feature is that the problem shows strong dependence in the y-direction, and little or no strong dependence in the x-direction. There are several subtle nuances associated with this discretization; our interest here, however, is in the coarse-grid selection.

The coloring produced for this problem is shown in Fig. 8.7. The algorithm generates a semicoarsened grid that is coarsened only in the y-direction. Again, this is precisely the coarse grid that would be used by a geometric semicoarsening method for this problem. Perhaps the most important observation is that the grid has been coarsened *only in the direction of strong dependence*. This makes sense: because smooth error varies slowly in the direction of strong dependence, interpolation can be performed accurately in that direction. AMG coarsening must exhibit this critical property.

Consider the nine-point Laplacian (Fig. 8.4) and the five-point Laplacian (Exercise 6). The coarsening for the five-point operator is called the red-black coarsening.

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

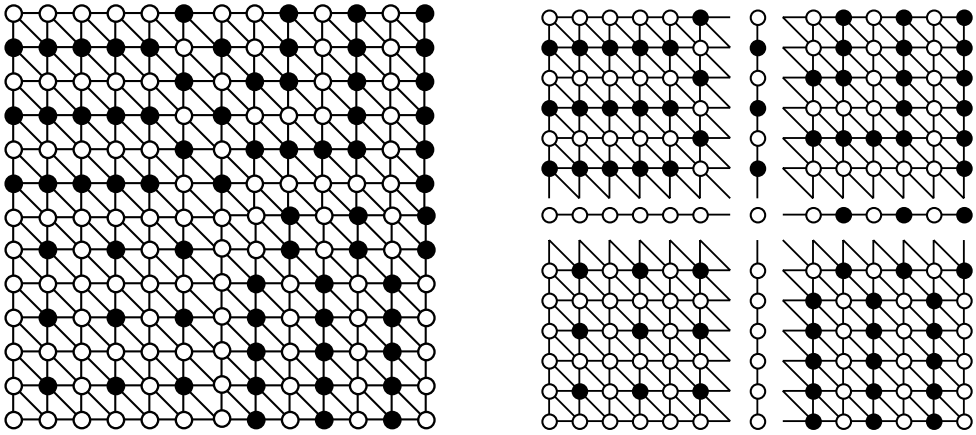

Figure 8.8: *Coarsening of problem* (8.4) *is shown on the left. Observe how the coarsening is in the direction of strong dependence in each of the four quadrants of the grid (right).*

The operator exhibits dependence in both the x- and y-directions, and the red-black coarsening is the only pattern that coarsens each grid line in both the x- and y-directions. The nine-point operator exhibits dependence in all directions: x, y, and diagonal directions. The resulting coarsening, known as *full coarsening*, coarsens in each of these directions. It is not possible to coarsen along x, y, and the diagonal grid line simultaneously; the algorithm instead coarsens every other line in each direction, with the intervening lines not represented on the coarse grid. The strong connectivity of this operator ensures that there are sufficient points available to interpolate the intervening lines well.

That AMG automatically coarsens in the directions of dependence is made apparent by the coarsening produced for the problem given in (8.4). The coarsening is shown in Fig. 8.8. Observe that in the upper left quadrant, where $a = 1$, $c = 1000$, and $b = 0$, the strong dependence is in the y-direction, and, in that region, the grid is coarsened primarily in the y-direction. Similarly, in the lower right quadrant where the coefficients are $a = 1000$, $c = 1$, and $b = 0$, the strong dependence is in the x-direction, and the grid is coarsened in the x-direction. In the lower left quadrant, the problem is just the normal Laplacian, and the coarsening is the standard full coarsening we observed for the nine-point Laplacian operator. Finally, in the upper right quadrant, where $a = 1$, $c = 1$, and $b = 2$, the coarsening is in the northwest/southeast direction, which is the direction of strong dependence for the stencil given in (8.5).

Coarse-Grid Operators

Recall that although physical grids may not be present, we continue to denote fine-grid quantities by h and coarse-grid quantities by $2h$. Once the coarse grid is chosen and the interpolation operator I_{2h}^{h} is constructed, the restriction operator

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

I_h^{2h} is defined using the usual variational property

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

The coarse-grid operator is constructed using the Galerkin condition

$$A^{2h} = I_h^{2h} A^h I_{2h}^h. \tag{8.16}$$

The reason for defining interpolation and the coarse operator by these variational principle is that the resulting coarse-grid correction is optimal in the A^h-norm (Exercise 8).

Cycling Algorithms

We have now defined all the components necessary to create a two-grid correction algorithm for AMG: a relaxation scheme, a set of coarse-grid points C, a coarse-grid operator A^{2h}, and intergrid transfer operators I_h^{2h} and I_{2h}^h. Although we have discussed weighted Jacobi, Gauss–Seidel relaxation is often preferred. The two-grid correction algorithm appears exactly as it did for geometric multigrid, as shown below.

AMG Two-Grid Correction Cycle

$$\mathbf{v}^h \leftarrow \text{ AMG}(\mathbf{v}^h, \mathbf{f}^h).$$

- Relax ν_1 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ with initial guess $\mathbf{v}^h$.
- Compute the fine-grid residual $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{v}^h$ and restrict it to the coarse grid by $\mathbf{r}^{2h} = I_h^{2h}\mathbf{r}^h$.
- Solve $A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}$ on Ω^{2h}.
- Interpolate the coarse-grid error to the fine grid by $\mathbf{e}^h = I_{2h}^h\mathbf{e}^{2h}$ and correct the fine-grid approximation by $\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{e}^h$.
- Relax ν_2 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ with initial guess $\mathbf{v}^h$.

Having defined the two-grid correction algorithm, we can define other multigrid cycling schemes for AMG guided by geometric multigrid. For example, to create a V-cycle algorithm, we simply replace the direct solution of the coarse-grid problem with a recursive call to AMG on all grids except the coarsest grid, where we use a direct solver. W-cycles, μ-cycles, and FMG-cycles can also be created by strict analogy to the geometric multigrid case.

Costs of AMG

How expensive is the AMG algorithm? The geometric case has regular and predictable costs, both in terms of the storage and floating-point operation counts. By contrast, there are no *predictive* cost analyses available for AMG. This is because we do not know, in advance, the ratio of coarse- to fine-grid points. Furthermore, this ratio is unlikely to remain constant as the grid hierarchy is established. However, we

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

have two simple tools that provide *a posteriori* cost estimates. Such estimates are useful in analyzing the performance of AMG and in building confidence that AMG is an effective tool for many problems. Additionally, they are extremely useful in debugging, code tuning, and algorithm development.

> **Definitions.** *Grid complexity* is the total number of grid points, on all grids, divided by the number of grid points on the finest grid. *Operator complexity* is the total number of nonzero entries, in all matrices A^{kh}, divided by the number of nonzero entries in the fine-grid operator A^h.

Thus, in the geometric case on the model problem, we have grid complexities of about 2, $\frac{4}{3}$, and $\frac{8}{7}$ for the one-, two-, and three-dimensional problems, respectively. Grid complexity gives an accurate measure of the storage required for the right sides and approximation vectors and can be directly compared to geometric multigrid.

Operator complexity indicates the total storage space required by the operators A^{kh} over all grids, which is generally not necessary in the geometric case. But it also has another use. Just as in the geometric case, the work in the solve phase of AMG is dominated by relaxation and residual computations, which are directly proportional to the number of nonzero entries in the operator. Hence, the work of a V-cycle turns out to be essentially proportional to the operator complexity. This proportionality is not perfect, but operator complexity is generally considered a good indication of the expense of the AMG V-cycle.

Performance

By its very nature, AMG is designed to apply to a broad range of problems, many far removed from the partial differential equations for which geometric multigrid was developed. The performance of AMG will naturally depend in large part on the peculiarities of the problem to which it is applied. Hence, we cannot hope to give a complete discussion of AMG performance. Accordingly, we close this chapter by discussing two numerical PDE examples.

Numerical example. Because it is helpful to illustrate the similarities and differences between the geometric and algebraic approaches, the first example is that of Chapter 4:

$$\begin{array}{rcll} -u_{xx} - u_{yy} & = & 2[(1-6x^2)y^2(1-y^2) + (1-6y^2)x^2(1-x^2)] & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \end{array} \tag{8.17}$$

where Ω is the unit square. Finite differences are applied on a uniform grid, with $n = 16, 32$, and 64 grid lines in each coordinate direction. We use a V(1,1)-cycle with Gauss–Seidel relaxation, sweeping first over the C-points, then over the F-points. This $C-F$ relaxation is the AMG analogue of a red-black Gauss–Seidel sweep. The results of the three experiments are shown in Table 8.1. Note that we use the standard (unscaled) Euclidean norm because we do not presume any geometric structure. In other words, we do not necessarily have the concept of mesh size. Also, we measure the residual only because the error is not ordinarily available.

In reality, AMG is unnecessary for this problem because it is precisely the case for which the geometric case was designed; nonetheless, the analysis of the performance of AMG on this problem is illuminating. We observe first that AMG exhibits

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

V-cycle	N	$\|\|\mathbf{r}\|\|_2$	Ratio of $\|\|\mathbf{r}\|\|_2$	N	$\|\|\mathbf{r}\|\|_2$	Ratio of $\|\|\mathbf{r}\|\|_2$	N	$\|\|\mathbf{r}\|\|_2$	Ratio of $\|\|\mathbf{r}\|\|_2$
0	16	1.73e + 01	–	32	3.49e + 01	–	64	7.01e+01	–
1	16	6.29e − 01	0.03	32	2.52e + 00	0.07	64	6.88e + 00	0.10
2	16	1.44e − 02	0.02	32	9.63e − 02	0.03	64	2.92e − 01	0.04
3	16	3.56e − 04	0.02	32	3.84e − 03	0.03	64	1.28e − 02	0.04
4	16	9.52e − 06	0.03	32	1.54e − 04	0.04	64	5.68e − 04	0.04
5	16	2.55e − 07	0.03	32	6.24e − 06	0.04	64	2.55e − 05	0.04
6	16	6.87e − 09	0.03	32	2.52e − 07	0.04	64	1.16e − 06	0.05
7	16	1.85e − 10	0.03	32	1.02e − 08	0.04	64	5.38e − 08	0.05
8	16	4.95e − 12	0.03	32	4.12e − 10	0.04	64	2.52e − 09	0.05
9	16	1.37e − 13	0.03	32	1.66e − 11	0.04	64	1.19e − 10	0.05
10	16	5.44e − 14	0.46	32	7.29e − 13	0.04	64	6.34e − 12	0.05
11	16	4.36e − 14	0.80	32	3.37e − 13	0.46	64	2.58e − 12	0.41
12				32	3.17e − 13	0.93	64	2.69e − 12	1.04

Table 8.1: *The table shows the results of AMG* V*-cycles applied to boundary value problem* (8.17)*. The* 2*-norm (Euclidean norm) of the residual after each* V*-cycle and the ratio of the residual norms on successive* V*-cycles are tabulated for* $n = 16, 32,$ *and* 64*.*

A^{ph}	Number of rows	Number of nonzeros	Density (% full)	Average entries per row
A^h	4096	20224	0.001	4.9
A^{2h}	2048	17922	0.004	8.8
A^{4h}	542	4798	0.016	8.9
A^{8h}	145	1241	0.059	8.6
A^{16h}	38	316	0.219	8.3
A^{32h}	12	90	0.625	7.5
A^{64h}	5	23	0.920	4.6

Table 8.2: *Properties of* A^{ph}*,* $p = 2^k, k = 0, 1, \ldots, 6$*, for AMG applied to a two-dimensional problem.*

the same type of convergence that was observed in Chapter 4 with geometric multigrid. The residual norm decreases by a relatively constant factor with each V-cycle. This continues until it levels off after about 12 V-cycles near 10^{-13}, where round-off error is on the order of the residual norm itself. Although we do not show the error, we would find that, as in the geometric case, the level of discretization error is reached after about six V-cycles. The errors would also exhibit the same reduction by a factor of four with each doubling of the resolution, as in the geometric case.

In terms of solver performance, AMG appears to be equivalent to geometric multigrid for this problem. However, there are other factors to be considered in examining AMG. These include the time required to do the setup and the storage required by the method.

We first examine the storage requirements. For the above experiment with $n = 64$, the operators A^{ph}, where $p = 2^k$ on levels $k = 0, 1, \ldots, 6$, have the properties shown in Table 8.2. Several observations are in order here. First, the initial coarse grid, Ω^{2h}, has 2048 points, exactly half the number on the finest grid. This occurs

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

because the five-point Laplacian operator yields the red-black coarsening described earlier (and in Exercise 6). However, each succeeding coarse grid has approximately one-fourth the number of points as the next finer grid. This may be understood by observing that the average number of nonzeros per row, which is 4.9 on the fine grid with the five-point operator (boundary points account for the average being below 5), increases to 8.8 on Ω^{2h} and remains above 8 for the next few grids. Evidently, the Galerkin coarse grid operators have become, effectively, nine-point Laplacian operators! The nine-point Laplacian operator yields full coarsening, in which each succeeding grid has one-fourth the number of points as the next finer grid. It is illuminating (Exercise 9) to examine the interpolation stencils and the formation of the coarse-grid operators to discover why the five-point operator becomes a nine-point operator on the coarse grid.

Summing the number of rows of all operators and dividing by the number of rows in the fine grid matrix (4096) shows that the grid complexity for this problem is 1.68. Thus, we know that storage of the vector of unknowns and the right sides require 1.68 times the space required for the fine-grid quantities. By contrast, the geometric approach has a grid complexity of about $\frac{4}{3}$. The difference can be explained by the fact that the first coarse grid produced by AMG is the red-black grid, while the geometric approach does full coarsening on all grids.

Summing the number of nonzeros in all operators and dividing by the number of nonzeros in the fine-grid operator shows that the operator complexity is 2.205. Thus, the matrices on all levels require just over twice the storage of the original operator A^h. In many cases, operators do not need to be stored in the geometric approach, so there is no explicit operator complexity to be considered. On the other hand, operator complexity also reflects the cost of one relaxation sweep on all grids, so a V(1,1)-cycle of AMG on this problem costs about 4.4 WUs (2.2 WUs on the descent and the ascent). This figure should be compared to $\frac{8}{3}$ WUs in the geometric case (neglecting the cost of intergrid transfers, as before).

Turning to the cost of performing the setup phase, we noted earlier that it is difficult to predict or even account for the amount of arithmetic involved. We can, however, make an observation about the "wall clock time" taken by the setup phase. In the $n = 64$ example, the setup phase required 0.27 seconds, while the solution phase required 0.038 seconds per V-cycle. The setup phase, then, required approximately the same amount of time as seven V-cycles.

These experiments show that AMG performs quite well on a model problem, but that it is more expensive in terms of storage, computation, and time than geometric multigrid. However, AMG is not intended for use on nicely structured problems; it is intended for problems on which geometric multigrid cannot be applied. We close our discussion with such an example. ◇◇

Numerical example. Consider the problem

$$-\nabla \cdot (a(x,y)\nabla u) = f(x,y) \tag{8.18}$$

in a domain Ω with $u = 0$ on $\partial\Omega$. Specifically, let $\Omega = ([0,10] \times [0,5.5])/B$, where B is the disk of radius $\frac{1}{2}$ centered on $(1.25, 2.75)$ shown in Fig. 8.9. The problem is discretized on the unstructured triangulation shown in Fig. 8.10. The discontinuous

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

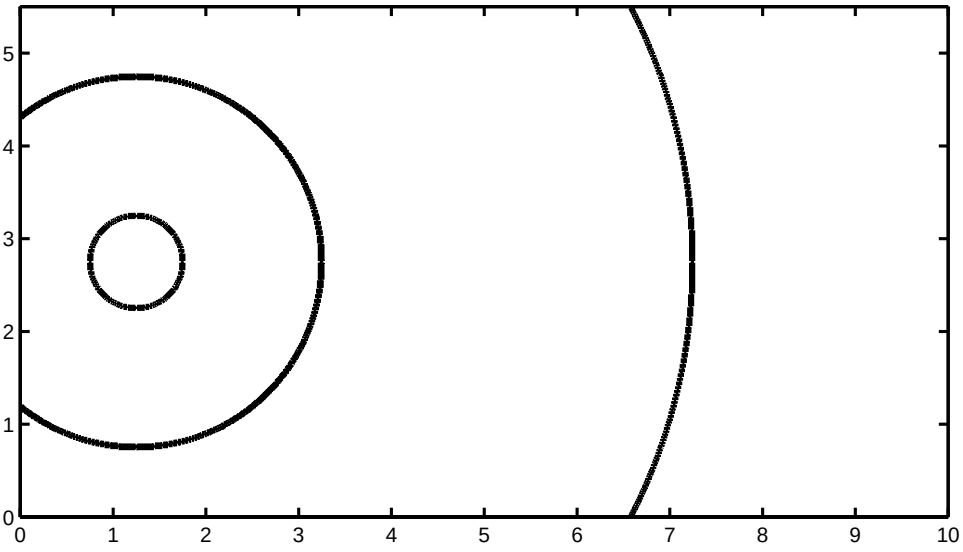

Figure 8.9: *The computational domain for* (8.18).

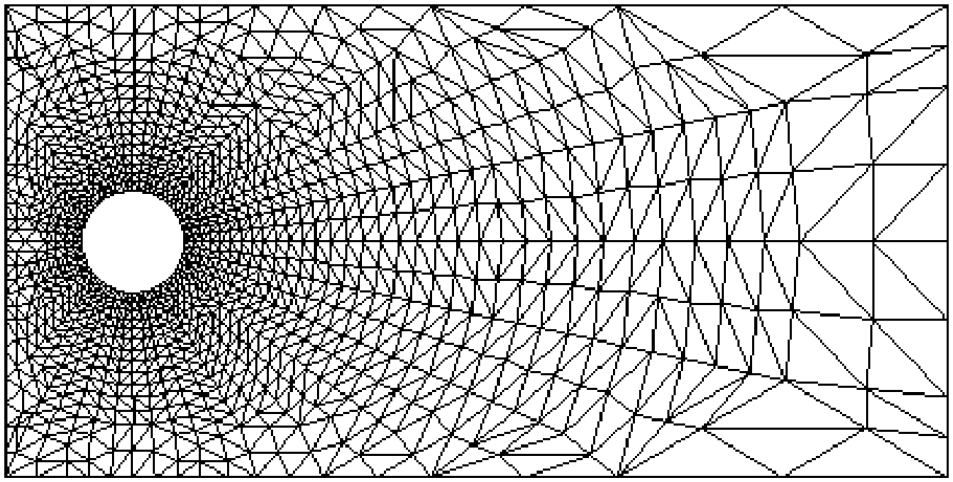

Figure 8.10: *The discretization mesh for* (8.18).

diffusion coefficient is given by

$$a(x,y) = \begin{cases} 0 & \text{if } \sqrt{(x-1.25)^2+(y-2.75)^2} < \frac{1}{2} \quad \text{(interior of hole)} \\ 1 & \text{if } \frac{1}{2} \le \sqrt{(x-1.25)^2+(y-2.75)^2} < 2 \\ 1000 & \text{if } 2 \le \sqrt{(x-1.25)^2+(y-2.75)^2} < 6 \\ 1 & \text{else.} \end{cases}$$

If (x, y) is between the circles, centered about the point $(1.25, 2.75)$, of radius 2

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Level	Number of rows	Number of nonzeros	Density (% full)	Average entries per row
0	4192	28832	0.002	6.9
1	2237	29617	0.006	13.2
2	867	14953	0.020	17.2
3	369	7345	0.054	19.9
4	152	3144	0.136	20.7
5	69	1129	0.237	16.4
6	30	322	0.358	10.7
7	20	156	0.390	7.8
8	18	125	0.383	6.9
9	3	9	1.000	3

Table 8.3: *Statistics on coarsening for AMG applied to* (8.18).

V-cycle	$\|\|\mathbf{r}\|\|_2$	Ratio of $\|\|\mathbf{r}\|\|_2$
0	1.00e + 00	–
1	4.84e − 02	0.05
2	7.59e − 03	0.16
3	1.72e − 03	0.23
4	4.60e − 04	0.27
5	1.32e − 04	0.29
6	3.95e − 05	0.30
7	1.19e − 05	0.30
8	3.66e − 06	0.31
9	1.12e − 06	0.31
10	3.43e − 07	0.31
11	1.05e − 07	0.31
12	3.22e − 08	0.31

Table 8.4: *Convergence of AMG* V*-cycles applied to* (8.18) *in terms of the* 2*-norm (Euclidean norm) of the residual.*

and 6, then the diffusion coefficient is three orders of magnitude larger than it is elsewhere.

This problem has features that would challenge a geometric multigrid algorithm. First, and most important, it is discretized on an unstructured grid. It is not at all obvious how to generate a sequence of coarse grids for this problem. In addition, the discontinuous coefficient causes real difficulty. Since the jump follows circular patterns and is not grid aligned, it is not possible to overcome this problem either by semicoarsening or by line relaxation.

To understand how AMG performs on this problem, we first examine the coarsening statistics in Table 8.3. One interesting phenomenon is that the relative density of the operators (percentage of nonzeros) increases as the grids become coarser. This increasing density can be seen in the last column, where the average number of entries per row increases through several coarse levels. Fortunately, the operator complexity is not adversely affected: a short calculation shows that the operator complexity is 2.97, while the grid complexity is 1.89. It is also interesting to note that the first coarse-grid operator has more actual nonzero coefficients than does the fine-grid operator. This is relatively common for AMG on unstructured grids.

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

The convergence of the method is summarized in Table 8.4. Using the standard 2-norm, we see that after 11 V-cycles, the residual norm reached 10^{-7} and the process attained an asymptotic convergence factor of 0.31 per V-cycle. Clearly, AMG does not converge as rapidly for this problem as for the previous model problem, where we saw a convergence factor of about 0.1. Nevertheless, in light of the unstructured grid and the strong, non-aligned jump discontinuities in the coefficients, the convergence properties are quite acceptable. ◇◇

Exercises

1. **Smooth error.** Under the assumptions that $\omega = \|D^{-1/2}AD^{-1/2}\|^{-1}$ and $\|D^{-1/2}AD^{-1/2}\|$ is $O(1)$, show that an implication of the smooth error condition, $\| \left(I - \omega D^{-1}A\right) \mathbf{e}\|_A \approx \|\mathbf{e}\|_A$, is that

$$(D^{-1}A\mathbf{e}, A\mathbf{e}) \ll (\mathbf{e}, A\mathbf{e}).$$

   Show that this property can be expressed as

$$||\mathbf{r}||_{D^{-1}} \ll ||\mathbf{e}||_A.$$

2. **Smooth error.** Show that for a symmetric M-matrix A, smooth error varies slowly in the direction of strong influence as follows. Assume $\|r\|_{D^{-1}} \ll \|e\|_A$.

   (a) Show that $||\mathbf{e}||_A << ||\mathbf{e}||_D$. Hint: $||\mathbf{e}||_A^2 = (A\mathbf{e}, \mathbf{e}) = (D^{-1/2}A\mathbf{e}, D^{1/2}\mathbf{e}) \leq ||\mathbf{r}||_{D^{-1}}||\mathbf{e}||_D$.

   (b) Show that if (as is often true) $\sum_{j\neq i} |a_{ij}| \approx a_{ii}$, then

$$\sum_i \sum_{j\neq i} |a_{ij}|(e_i - e_j)^2 << \sum_i a_{ii}e_i^2.$$

   Hint: Show that $(A\mathbf{e}, \mathbf{e}) = \frac{1}{2}\sum_{ij}(-a_{ij})(e_i - e_j)^2 + \sum_{ij} a_{ij}e_i^2$.

   (c) Conclude that (8.7) must hold on average, that is, for most i.

3. **Deriving interpolation weights.** Carry out the calculation that leads to expression (8.12) for the interpolation weights.

4. **Properties of interpolation weights.** Let A be an M-matrix whose rows sum to zero for interior points, and let the ith row represent an interior point. Show that substituting $e_j = e_i$ in the sum over D_i^s in (8.9) and in the sum over D_i^w in (8.11) produces interpolation coefficients ω_{ij} that are nonnegative and sum to unity.

5. **Enforcing the heuristics.** Construct a simple one-dimensional example in which the two heuristics **H-1** and **H-2** cannot be simultaneously enforced. Hint: Use periodic boundary conditions and odd n.

6. **Coarse-grid selection.** Show that the first pass of the coarsening algorithm (the initial coloring phase) applied to the five-point Laplacian stencil

$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}$$

   produces standard red-black coarsening (the C-points are the red squares on a checkerboard; the F-points are the black squares).

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

7. **Second coloring pass.** The central idea behind the second-pass coloring algorithm is to test each F-point, i, in turn, to ensure that each point in D_i^s depends strongly on at least one point in C_i. When an F-point, i, is found to depend strongly on another F-point, j, that does *not* depend strongly on a point in C_i, then j is made (tentatively) into a C-point; testing of the points in D_i^s then begins again. If all those points now depend strongly on points in C_i, then j is put permanently in C. However, if some other point in D_i^s is found that does not depend strongly on a point in C_i, then i itself is placed in C and j is removed from the tentative C-point list. The process is repeated for the next F-point and continues until all F-points have been treated.

   Apply this second-pass algorithm to the nine-point Laplacian operator with periodic boundaries (Figure 8.5) and determine the final coarsening. Observe that the final coarsening depends on the order in which the F-points are examined.

8. **Using variational properties.** Show that using the variational properties to define interpolation and the coarse-grid operator is optimal in the following way. The two-grid correction scheme, given in the text, corrects the fine-grid approximation $\mathbf{v}^h$ by a coarse-grid interpolant $I_{2h}^h \mathbf{v}^{2h}$ that gives the least error in the sense that

$$\|\mathbf{e}^h - I_{2h}^h \mathbf{v}^{2h}\|_{A^h} = \min_{\mathbf{w}^{2h}} \|\mathbf{e}^h - I_{2h}^h \mathbf{w}^{2h}\|_{A^h},$$

   where $\mathbf{e}^h = \mathbf{u}^h - \mathbf{v}^h$ is the error in $\mathbf{v}^h$.

9. **Coarsening the five-point operator.** Examine the five-point Laplacian operator on a uniform two-dimensional grid, and determine why the Galerkin process alters it to the nine-point operator on the coarse grid. Hint: This can be done symbolically by examining which entries of the stencil will be used to interpolate the various points and then by symbolically carrying out the Galerkin multiplication $A^{2h} = I_h^{2h} A^h I_{2h}^h$.

10. **Smooth error implications.** Show that if $A = D - L - U$ is a symmetric M-matrix and $Q = D + L$ (Gauss–Seidel) or $Q = D$ (Jacobi), then the quantities $(Q^{-1}A\mathbf{e}, \mathbf{r})$ and $(\mathbf{e}, \mathbf{r})$ are nonnegative.

11. **Smooth error.** Show that if $A = D - L - U$ is a symmetric M-matrix and Gauss–Seidel produces errors that satisfy

$$\left\| \left(I - (D+L)^{-1}A\right) \mathbf{e} \right\|_A \approx \|\mathbf{e}\|_A,$$

    then

$$||\mathbf{r}||_{D^{-1}} \ll ||\mathbf{e}||_A.$$

    Hint: Show first that $\|(D+L)^{-1}Ae\|_A \ll \|e\|_A$. Then show and use the fact that $(D + L^T)^{-1}A(D+L)^{-1} \leq D^{-1}$.

12. **V-cycle costs.** Let the grid complexity be denoted σ^Ω and the operator complexity be denoted σ^A. In addition, define the following quantities: κ^A, the average number of nonzero entries per row over all levels; κ^I, the average number of interpolation points per F-point; n_m^A, the number of nonzero entries in A^m; and n_m^C and n_m^F, the number of C- and F-points, respectively, on grid Ω^m.

(a) Show that the number of floating-point operations on level m for one relaxation sweep, residual transfer, and interpolation are $2n_m^A$, $2n_m^A + 2\kappa^I n_m^F$, and $n_m^C + 2\kappa^I n_m^F$, respectively.

(b) Noting that

$$\sum_m n_m^F \approx n,$$

show that the total flop count for a V(ν_1, ν_2)-cycle is given approximately by

$$n(2(\nu+1)\kappa^A\sigma^\Omega + 4\kappa^I + \sigma^\Omega - 1),$$

where $\nu = \nu_1 + \nu_2$.

Downloaded 03/08/21 to 132.178.207.22. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms