10 ALGEBRAIC MULTIGRID (AMG) FOR SPARSE MATRIX EQUATIONS

A. Brandt
Weizmann Institute of Science, Israel

S. McCormick
J. Ruge
Institute for Computational Studies, Colorado State University

This work was supported in part by the Air Force Office of Scientific research (Grant No. AFOSR82-0240) and the National Bureau of Standards.

ABSTRACT

Many matrix equations are either inherently discrete (e.g., in geodesy) or for certain practical purposes remote from their origin (e.g., a finite element discretization on a preselected irregular grid). AMG is an algorithm designed to solve such problems by using information contained only in the matrix while at the same time basing itself on multigrid principles. This paper introduces the basic AMG concepts, develops its foundations, and describes current AMG strategies.

1. INTRODUCTION

The class of problems to which multigrid methods have been applied successfully is constantly growing. So far, most of them have been derived from continuous problems, where knowledge of the underlying geometry is used as a guide to provide several uniform diescretizations of the domain. Each of these so-called grids or levels can be used as a uniform coarsening of the next finer one. The solution process, which involves relaxation sweeps on each grid, fine-grid-to-coarse-grid transfers of residuals, and coarse-to-fine interpolations of corrections, constitutes a very fast solver for the finest-grid equations. In fact, whether the equations are linear or not, a solution with algebraic error smaller than truncation error is typically obtained in four to ten *work units*, where a work unit is the amount of computer operations required to *express*

the equations (e.g., the work of a matrix multiply for linear equations).

Although this process seems to rely heavily on the geometry and continuous nature of the problem, the principles involved in solving the fine grid matrix system can be abstracted and applied to various classes of matrix problems. The aim of this paper is to develop such an algorithm, algebraic multigrid (AMG), that results from basing multigrid concepts solely on information contained in the matrix. The potential benefits of AMG include:

1. 'Black box' multigrid software. The virtues here are obvious, especially in light of the present state of multigrid software and the substantial human design effort needed for many multigrid applications.
2. Applications to unorganized grids. When the given grid is not topologically piecewise uniform, as in the case of finite element discretizations with arbitrary, irregular triangulations, conventional multigrid design may be faced with difficulties. Determination of coarsening (i.e., the coarse grids and their associated operators) may not be practical. A further difficulty in such problems is to choose both relaxation and coarsening when the problem exhibits certain directional properties. This includes anisotropic operators, which require either some form of block relaxation or semi-coarsening. Especially severe cases arise when the computational mesh is highly stretched as a result of coordinate transformations or Lagrangian

discretizations. These difficulties disappear in the AMG approach.

3. Pathological coefficient applications. Even for uniform discretizations, conventional coarsening may not be able to account for certain pathologies in the equation coefficients. Such is the case, for example, with five-point discretizations of two- dimensional diffusion problems where the diffusion coefficients are distributed in some particular patterns (see [1; Sec. 8])]. Again, AMG would have no difficulty in such cases since it would select much better coarse grids than the 'natural' ones.
4. Algebraic problems. AMG may be applied to large sparse linear and nonlinear systems which are not derived from continuous problems, including the geodetic application treated in [3; Sect. 6] and [7]. Many of these problems are such that each unknown is associated with a point in a low dimensional space (e.g., dimension 2 or 3) and that most of the points exhibit pairwise couplings (i.e., matrix entries) that either diminish or become smooth as the distances between the points increase. It is this feature of an algebraic problem that should allow for efficient solution by AMG.

Multilevel processing for the solution of problems not derived from continuous, nor even geometrically-based, systems has for a long time been used in economics (cf., [4] and the survey in [6]). In a sense, nobody 'invented' it; multilevel organization is simply

the way society itself found efficient. This concept led to iterative aggregation algorithms (cf., [6] and [10]) in which the coarsening (aggregation) procedures are supposedly given by the nature of the problem just as they are in usual multigrid algorithms and other geometrically-based multilevel processing. The main point of AMG, by contrast, is the complete automation of the coarsening procedures, including the selection of the coarser levels and the inter-level transfers, basing them on the given matrix only. The efficiency obtained by AMG is similar to the typical multigrid efficiency, which is not approached by the less sophisticated aggregation methods.

The AMG structure provides more than a fast linear solver. It can, for example, be used to solve with similar speed nonlinear problems, to update for negligible work the entire solution as the problem changes locally, to solve large problems in small storage and to give inexpensive representation and computation of the inverse of large sparse matrices. (See [3; Secs. 6.7 and 7].)

This paper is intended first to provide a *brief* description of AMG including its basic concepts and foundation. We further treat *in brief* several important issues and AMG strategies that have arisen since our first report [3]. However, even as we write this account, new attempts are being made at fundamental improvements in these strategies.

Brief has special significance here. During the course of our work on AMG an overwhelming collection of observations, concepts, and algorithmic variations suggested themselves. By necessity, we used experience, analysis, and some numerical experiments in our pursuit of efficient and robust AMG techniques. Many alternatives

to the AMG processes described below are easy to conceive, but their abundance prevents us from discussing them here.

This paper is organized as follows: Section 2 describes the algebraic properties of the iteration error that characterize slow convergence of relaxation methods; Section 3 motivates the basic principles of AMG; Section 4 describes the basic AMG processes; Section 5 includes a note on numerical performance; Section 6 describes a few major areas of further research; and the Appendix contains a somewhat detailed description of the current AMG coarsening strategy.

2. PROPERTIES OF SLOWLY CONVERGING ERRORS

2.1 Algebraic property: small residuals

It is first important to understand, in general algebraic terms, what type of errors become dominant when basic iterative schemes are slow to converge. A common feature of those iterative schemes is that at each step some correction to the approximate solution is calculated based on the current magnitute of certain residuals. Hence, the convergence must be slow if the residuals are in some sense small compared with the errors. The converse is also true: if convergence of a *proper* relaxation scheme is slow, the residuals must in some sense be small compared with the errors.

To see this, consider for example Kaczmarz relaxation applied to a general real or complex matrix equation

(2.1) $AX=b$.

We suppose the current approximation x is converging to a solution X. (To treat this most general case, where A need not even be square, we consider Kaczmarz method which converges whenever (2.1) is solvable [9].) Denoting by $e = x-X$ the current error vector, by $r = Ae = Ax-b$ the current residual vector, and by a_i the i-th row of A, then the Kaczmarz step corresponding to that row is to replace x by $x - (r_i/a_i a_i^t)a_i^t$, thereby forcing r_i to zero. A full Kaczmarz sweep is the employment of such a step for each row of A, in

the natural ordering. Let

(2.2) $\tilde{E} = \Sigma |e_i|^2$ and $R = \sum_i (|r_i|^2 / \sum_j |a_{ij}|^2)$

be square norms for errors and residuals, respectively, scaled as is evident so that they are comparable. One can then prove the following result [2; Thm. 3.4]:

Theorem 2.1. Let $\hat{R}^0$ and $\hat{R}^1$ denote the values of $\tilde{R}$ before and after one sweep of Kaczmarz relaxation, respectively, and let

$$\tilde{\gamma}_0 = [(1+\tilde{\gamma}_-)(1+\tilde{\gamma}_+)]^{-1},$$

$$\tilde{\gamma}_1 = [\tilde{\gamma}_-\tilde{\gamma}_+]^{-1},$$

$$\tilde{\gamma}_- = \max_i \left(\sum_{j<i} \left| \sum_\lambda a_{i\lambda}\bar{a}_{j\lambda} \right| / \sum_\lambda |a_{i\lambda}|^2 \right), \text{ and}$$

$$\tilde{\gamma}_+ = \max_i \left(\sum_{j<i} \left| \sum_\lambda a_{i\lambda}\bar{a}_{j\lambda} \right| / \sum_\lambda |a_{i\lambda}|^2 \right)$$

Then the decrease in $\tilde{E}$ over that sweep is at least $\tilde{\gamma}_0\hat{R}^0$; it is also at least $\tilde{\gamma}_1\hat{R}^1$.

This theorem says in essence that *slow convergence can occur when $\tilde{R}$ is small compared with $\tilde{E}$.* (Observe than in case A arises from the discretization of differential equations, or is similarly structured, $\tilde{\gamma}_i$ are completely local quantities; they do not depend on the size of A.)

For special classes of matrices A, simpler (and more effective) relaxation schemes can achieve the same goal. For example, for symmetric positive-definite matrices, an effective method is *Gauss-Seidel relaxation*, where now the step corresponding to the i-th equation is to replace x_i by $x_i - r_i/a_{ii}$, leaving the rest of x unchanged, again forcing r_i to zero. The error and residual square norms in this case are

(2.3) $E = e^t r$ and $R = \Sigma r_i^2 / |a_{ii}|$,

for which we have the following result [2; Thms 3.1 and 3.2]:

Theorem 2.2. Suppose A is symmetric with positive diagonal. Let R^0 and R^1 denote the values of R before and after one Gauss-Seidel sweep and let

$$\gamma_0 = [(1+\gamma_-)(1+\gamma_+)]^{-1},$$

$$\gamma_1 = [\gamma_-\gamma_+]^{-1},$$

$$\gamma_- = \max_i \sum_{j<i} |a_{ij}| / a_{ii}, \text{ and}$$

$$\gamma_+ = \max_i \sum_{j>i} |a_{ij}| / a_{ii}$$

Then the decrease in E over that sweep is at least $\gamma_0 R^0$; it is also at least $\gamma_1 R^1$.

The constants γ_i appearing in this theorem are sharp in many cases, while in others they are not far off and optimal constants can be derived by local mode analyses. Similar results hold for all other familiar relaxation schemes, such as Jacobi relaxation with a suitable under-relaxation parameter, block relaxation schemes (with a suitable change in the form of R), etc. (See [2;Sec. 3].)

2.2 Geometric and algebraic smoothness

When (2.1) represents a discretization of a differential equation LU = f, then small residuals r = Ae imply that e is locally an approximate solution to Le = 0. When L is elliptic, this implies that e is a smooth function. (A specialized sense of smoothness is implied when L is not elliptic.) This smoothness is the basis for the usual 'geometric' multigrid algorithms in which the smooth error functions produced by relaxation are subsequently approximated on coarser grids. To develop such a foundation for AMG, we must free the concept of smoothness from its geometric context.

Since E and R (or $\tilde{E}$ and $\tilde{R}$) were scaled to be comparable, it is only for special error components that the slow-convergence condition

$$R \ll E \qquad (\text{or } \tilde{R} \ll \tilde{E}) \tag{2.4}$$

can be satisfied. The deeper (2.4) is satisfied, the more special must be the error, and hence the fewer is the number of parameters needed to approximate it. Thus, broadly speaking, relaxation reduces the information content of the error and makes it approximateable by lower dimensional spaces ('coarser grids').

To see more specifically how these lower dimensional spaces should be constructed, consider first the special class of symmetric *positive-type* matrices, which form the prototype for our initial development of AMG. We call the matrix $A = (a_{ij})$ positive-type if $a_{ij} \le 0$ for all $i \ne j$, whereas, $\sum_j a_{ij} \ge 0$ for all i. (This is equivalent to the assumption that A is a diagonally dominant M-matrix.) In this case,

$$E = 1/2 \sum_{i,j} (-a_{ij})(e_i-e_j)^2 + \sum_i \left(\sum_j a_{ij}\right) e_i^2$$
$$1/2 \sum_{i,j} (-a_{ij})(e_i-e_j)^2.$$

By the Cauchy-Schwarz inequality, $E^2 \le R \Sigma a_{ii} e_i^2$,

so the slow-convergence condition (2.4) implies

$$\sum_{i,j} (-a_{ij})(e_i-e_j)^2 \ll \sum_i a_{ii} e_i^2. \tag{2.5}$$

Since $a_{ij} \le 0$ for $i \ne j$, this means that $e_i - e_j$ is on the average small compared with $\|e\|$ whenever a_{ij} is comparable with either $\max_{k \ne i} |a_{ik}|$ or $\max_{k \ne i} |a_{jk}|$, i.e., whenever i and j are *strongly coupled*. Thus, we conclude that Gauss-Seidel

relaxation (and similarly other schemes) *smoothes the error along strongest couplings* (and couplings comparable to the strongest).

Block relaxation can give stronger smoothing. By including the strongest couplings in the blocks, the error will also be smoothed along next-strongest couplings [2;Sec.3.5]. In our AMG development, however, block relaxation is not usually needed; smoothing along strongest couplings is enough because the coarser grid can be selected so that interpolation is always done along such couplings. (Similarly, in geometric multigrid, full smoothing, e.g., by line relaxation in case of anisotropic operators, is not needed if semi-coarsening is used; see [11;Sec. 4.2.1].)

The concept of smoothing along strong couplings can be generalized to *sparsely positive-definite* matrices (see [2]), and perhaps to most other matrix types of interest. In practice, the sense in which a given relaxation scheme smoothes for a given matrix A can algorithmically be detected by techniques described in Section 6.1 below. How that smoothing is exploited by AMG will be described in subsequent sections.

3. MULTIGRID PRINCIPLES

3.1 Two-level schemes

As we noted above, smooth errors can be well represented by fewer variables. To see how this can be done, we first introduce the notion of a two-level multigrid scheme. To this end, let G be a given ('fine') grid, that is, a finite set of points in R^d, where d is some positive integer. Suppose that a ('coarse') grid G^c is given that represents some 'coarsening' of G (e.g., G^c may be a subset of G) and let I_c: $G^c \to G$ ('interpolation') and I^c:$G \to G^c$ ('restriction') denote the intergrid transfers. Let A: $G \to G$ and A^c : $G^c \to G^c$ be linear operators. (We identify each grid with its associated space of grid functions.) Suppose b in G is given and let X be a solution of the fine grid equation

$$AX = b. \tag{3.1}$$

With x the current approximation to X, then one multigrid cycle consists of applying, say, several Gauss-Seidel sweeps followed by a residual equation transfer to G^c via

$$A^c v^c = I^c(b - Ax). \tag{3.2}$$

This problem is then solved (with present two-level case) and used to correct the fine grid approximation according to

$$x \leftarrow x + I_c v^c. \tag{3.3}$$

The reason for the efficiency of such a multigrid scheme is that, with proper choice of the grid transfer and coarse grid

operators, the error that cannot be eliminated by the coarse grid correction (3.4) is effectively reduced by relaxation. To ensure that the coarse grid correction actually does what it is intended to do, it is necessary to define A^c in the right way. Note that (3.2) and (3.3) imply that the coarse grid correction is given by

$$x \leftarrow x + I_c A^{c^{-1}} I^c (b - Ax). \tag{3.4}$$

Letting $e = x - X$ denote the error, then (3.4) can be rewritten as

$$e \leftarrow (I - I_c A^{c^{-1}} I^c A)e. \tag{3.5}$$

Hence, the goal of the correction is roughly to eliminate error in $R(I_c)$, the range of the interpolation operator. Thus, if $e = I_c v^c$ for some v^c, it is desirable that

$$(I - I_c A^{c^{-1}} I^c A)\, I_c v^c = 0$$

or, since I_c should be full rank, that

$$A^c = I^c A I_c. \tag{3.6a}$$

Note that, with this choice if A^c, the coarse grid correction becomes a projection.

For the special case that A is a real symmetric positive definite matrix, we can derive this same form for A^c together with a prescription for I^c by way of a variational condition for the energy norm $\|e\|_E = e^t A e$. Specifically, by seeking a vector v^c in G^c that minimizes $\| e - I_c v^c \|_E$, we are led to (3.6) with

$$I^c = I_c{}^t. \tag{3.6b}$$

These 'variational relations' (3.6) are in fact used in AMG for such matrices to form I^c and A^c once I_c has been determined.

In many applications for geometric multigrid algorithms, the coarse grid approximates smooth fine grid vectors in a fairly strong sense. Specifically, suppose E and R represent the error measures

in either (2.2) or (2.3), which we use interchangeable with $\| e \|_E^2$ and $\| e \|_R^2$, respectively, when we need to depict the error e on which they depend. Then we are often assured that, for any e in G, there exists an e^c in G^c so that

$$\| e - I_c e^c \|_E \le \gamma \| e \|_R. \tag{3.7}$$

(Here, as in Section 2, we use γ generically to denote quantities that depend only locally on A.) Let e_1, e_2, and e_3, respectively, be the errors in such a multigrid application just after the previous coarse grid correction, after relaxation has slowed, and after the subsequent coarse grid correction. Then the e_2^c for e_2 guaranteed by this approximation condition, together with the variational property of the coarse grid correction, the slow-convergence relation (2.4), and the fact that E decreases in relaxation, all imply that

$$\|e_3\|_E \le \|e_2 - I_c e_2^c\|_E \le \gamma \|e_2\|_R \ll \|e_2\|_E \le \|e_1\|_E.$$

Thus, E must be substantially reduced in each full cycle.

Efficient two-level algorithms can in fact be obtained under much weaker conditions. Thus, in terms of the norm $\| e \|_W = \Sigma a_{ii} e_i^2$, it is enough that, for any e in G, there exists an e^c in G^c satisfying

$$\| e - I_c e^c \|_W \le \gamma \| e \|_E. \tag{3.8}$$

(See [2; Thm. 4.1].) Such a condition, which is weaker than (3.7), can be guaranteed by interpolation along strong connections. For example, in our prototype symmetric positive-type case, interpolation can be (and is) taken to be regular averaging, i.e.,

$$I_c = (w_{ik}), \quad w_{ik} \ge 0, \quad \sum_k w_{ik} = 1, \tag{3.9}$$

in which case (3.8) is satisfied if, for every e in G, there exists an e^c in G^c so that

$$(3.10)\qquad \sum_{i,k} a_{ii} w_{ik} (e_i - e_k^c)^2 \le \frac{\gamma}{2} \sum_{i,j} (-a_{ij})(e_i - e_j)^2 .$$

A simple way to satisfy (3.10) is to regard G^c as a subset of G and let the interpolation coefficient w_{ik} be proportional to the strength of connection a_{ik}. More precisely, let C denote the indices that correspond to G^c in G and let F denote the other indices. Then $v = I_c v^c$ is defined so that

$$(3.11)\qquad v_i = \begin{cases} v_i^c & i \text{ in } C \\ \dfrac{\sum_{j \in C} (-a_{ij}) v_j^c}{\sum_{j \in F} (-a_{ij})} & i \text{ in } F \end{cases}$$

Then it is easy to see [2; Sec. 4.3] that (3.10) is satisfied by

$$(3.12)\qquad \gamma = 1/2 \min_{i \in C} \sum_{j \in F} (-a_{ij}) / a_{ii} .$$

The important conclusion here is that, if there is sufficient total *direct* strength to C points for every F point, then by (2.3) we are guaranteed that a full coarse-grid correction followed by one relaxation sweep must reduce the error measure E by a substantial factor independent of the matrix size. Actually, for satisfying (3.10), we do not need to be so restrictive as to guarantee that the strength to the C points is *direct*, nor do we need to interpolate from all available C points; but we defer these issues to Section 4.

3.2 Multi-level scheme

The two-level algorithm described in the previous section in effect reduces the problem of solving an equation in A to that in A^c. To solve the latter effectively, we must continue to process to yet coarser levels. Thus, we define a multi-level (multigrid) algorithm by applying the two-level algorithm recursively.

In particular, the *set-up* process for a multi-level algorithm starts with $G^0 = G$ and $A^0 = A$ and determines G^c, A^c, I_c, and I^c from G and A according to the principles of Section 3.1. Let $G^1 = G^c$, $A^1 = A^c$, $I_1 = I_c$, and $I^1 = I^c$. Then this coarsening process continues by determining G^i, A^i, I_i, and I^i from G^{i-1} and A^{i-1} for each i = 2, 3, ..., M in turn according to the same procedures. Thus, A^i is the grid G^i operator and $I_i : G^i \to G^{i-1}$ and $I^i : G^{i-1} \to G^i$ are the intergrid transfers.

To describe the multi-level *solution* process, let the number ν of relaxation sweeps per grid be specified. Given f^i on grid G^i and an approximation, x^i, on G^i to the solution of

$$(3.13)\qquad A^i x^i = f^i ,$$

then one multi-level V-cycle on grid G^i is denoted by $x^i \leftarrow MGV(i, x^i, f^i)$ and is defined recursively by:

If i = M, set $x^i \leftarrow A^{i-1} f^i$.

Otherwise, do the following:

perform ν Gauss-Seidel sweeps on (3.13);

$x^{i+1} \leftarrow 0$ and $f^{i+1} \leftarrow I^{i+1}(f^i - A^i x^i)$;

$x^{i+1} \leftarrow MGV(i+1, x^{i+1}, f^{i+1})$; and

$x^i \leftarrow x^i + I_{i+1} x^{i+1}$.

4. AMG FOR SYMMETRIC POSITIVE-TYPE MATRICES

The two primary tasks in developing an AMG solver for a given symmetric matrix A are the determination of C, the coarse grid variables, and the computation of I_c, the interpolation formula. The fine-grid-to-coarse-grid transfer I^c and the coarse grid operator A^c are then determined by the variational relations (3.6). Other AMG processes, such as the relaxation ordering (C-F) and cycling schemes, are usually straightforward (cf.[3]). In this section, we discuss issues that relate to these two primary tasks for symmetric positive-type matrices.

4.1 Prototype AMG and its limitations

For each i in F, let $S^I(i)$ denote the set of points in C that are used in the interpolation formula for i, that is, $S^I(i) = \{k \text{ in } C : w_{ik} \neq 0\}$. As we noted at the end of Section 3.1 for our prototype matrices (i.e., symmetric positive-type), to develop an AMG with good two-grid convergence rates, it is enough to ensure that every point i in F has sufficient total *direct* strength to $S^I(i)$. In our early efforts, this premise was used to develop a prototype AMG which was based on ensuring that each i in F had at least a certain fraction of its total direct strength to $S^I(i)$.

A major difficulty with this prototype AMG is its tendency to exhibit fairly high *total relative complexity* , where by this we

mean the total number of non-zero entries of all grid matrices divided by the number of nonzeros in A. This is because this complexity depends very sensitively on the number of connections used in interpolation - and this is large when AMG is forced to achieve sufficient direct strength.

A second difficulty with our prototype is that it can in fact exhibit slow V-cycle convergence. That is, even though it achieves the approximation conditions in (3.8) that guarantee acceptable two-grid convergence rates, the stronger conditions (3.7) needed for V-cycle convergence may be violated. Simple geometric examples in [2; Sec. 4.7] show in fact that this prototype AMG can produce low-order interpolation that results in slow V-cycle rates.

4.2 Indirect strength

Condition (3.10) is easily satisfied by taking sufficiently strong direct connections into account in interpolation, but it can also be satisfied through indirect connections. For example, certainly

$$w_{ik}\,(e_i - e_k^c)^2 \le 2\,w_{ik}\,(e_i - e_j)^2 + 2\,w_{ik}\,(e_j - e_k^c)^2.$$

Suppose point i strongly depends on point j which in turn strongly depends on point k; we say that point i is strongly but indirectly coupled to point k. Such indirect strength can thus be used to satisfy (3.10). This means that, even when large neighborhoods of strong couplings exist, suitable approximation can still be achieved provided the neighbors of the points in $S^I(i)$ largely overlap the neighbors of i. (See [2; Sec.4.4].)

To take indirect strength into account, we first modified our prototype algorithm, using what we call [3] 'p-point interpolation'.

More specifically, we adopted the new objective of requiring essentially that each point i in F have its p strongest connections to $S^I(i)$, p a small fixed integer. This limited the connections to $S^I(i)$ so that complexity was better controlled (cf.[3] and [8]), but it did not go far enough to ensure total strength (direct and indirect) to $S^I(i)$. Consider the geometric example of a fully uniform five-point discretization of the LaPlacian in two dimensions with p = 2. There is no means for AMG as such to decide which of the four neighbors of a given i to place in $S^I(i)$, yet only by interpolating from points on two opposing sides of i can linear interpolation be obtained. Such 'two-sided' interpolation is the only way here to achieve the stronger approximation conditions (3.7).

Our *current* version of AMG places no artificial bound on the number of points in $S^I(i)$, but rather attempts to make $S^I(i)$ as small as possible while ensuring that the total direct and indirect strength is sufficient. Specifically, the objectives of the point choice strategy for our current algorithm are:

1) For every point i in F, each point j on which it strongly depends most either be in $S^I(i)$ or must itself depend strongly on a point in $S^I(i)$.
2) Total relative complexity should be minimized, that is, there should be as few C points as possible and $S^I(i)$ should be as small as possible.
3) The set-up processes should be fast. Its speed is limited by the time, t, it takes to compute A^c once G^c and Ic have been defined, so the goal here is to keep the set-up time at about 2t. (t is determined by how well Objective 2 is achieved and the speed of the solution process depends on this and the nature of the approximation by the coarse grid.)

Note that Objective 1 overcomes some of the difficulty with the previous algorithms of achieving the stronger approximation conditions (3.7). For instance, for the five-point example of the previous paragraph, our current strategy produces a red-black coarsening so that all neighbors of each point are in $S^I(i)$.

In order to achieve speed, a quick first approximation to the coarse grid is obtained with Algorithm A, a brief description of which is included in the Appendix. This distributes the C points over the grid in such a way that few C-point-to-C-point strong connections exist and the F points have connections to at least one (though usually more) C points. For geometric operators with fairly uniform strong connections, this tends to give a uniform distribution of C points suitable for interpolation. The C point density is controlled by a parameter ε_1.

Of course, this algorithm does not guarantee that Objective 1 is achieved. However, this may be tested and corrected at the same time the interpolation coefficients are being computed. This is done in Algorithm B, which is also described in the Appendix. Here, an F point is interpolated from a subset of the strong connections, as defined by another parameter, ε_2 (usually $\varepsilon_2 < \varepsilon_1$), and points not used in interpolation must be strongly connected to $S^I(i)$. There are several ways that this strength can be measured, but in any case, a third parameter, ε_3, is used to control it. If, while computing the coefficients for a point i in F, a point j also in F is found not to be strongly

connected to $S^I(i)$, j is itself made a 'conditional' C point. If, after making j such a point, there is yet another strong connection of i which is not a C point and is not strongly connected to the new set $S^I(i)$ (including j), then i itself is made a C point and j regains its status as an F point. (It is not worth the added complexity to keep i an F point if it requires more than one additional C point to do it.)

4.3 Improved interpolation formulae

This revised coarse grid point choice strategy described in the previous section lends itself to an improvement in the interpolation formula. To be specific, suppose i is in F. Then the motivation for the operator interpolation depicted in (3.11) starts with the assumption for relaxed errors that $r_i = 0$ (approximately). Thus, for such errors,

$$(4.1)\quad e_i = \sum_{k \neq i} (-a_{ik})\, e_k \quad \text{(approximately)}.$$

The points not in $S^I(i)$ in this summation are then eliminated by using the smoothness condition which, for our prototype matrices, is written as

$$(4.2)\quad \sum_{j \in F} a_{ij}\, e_i = \sum_{j \in F} a_{ij} e_j \quad \text{(approximately)}.$$

This leads directly to (3.11). However, the revised C point choice strategy provides more information at points not used in interpolation. Thus, for each point j not in $S^I(i) \cup \{i\}$ it can be assumed that

$$(4.3)\quad e_j = \sum a_{jk} e_k \,/\, \sum a_{jk},$$

where the summation here is take over all k in $S^I(i) \cup \{i\}$. That is, to determine w_{ik}, any a_{ij} in (4.1) can be replaced by changing the values of a_{ik} according to the stencil for point j. The values w_{ik} that result from this need not be represented explicitly here,

5. NOTES ON NUMERICAL PERFORMANCE

The purpose of this paper is to describe the current status of AMG, not to report extensively on its performance characteristics. Numerial results for earlier AMG algorithms have been reported elsewhere (cf. [3] and [8]) so that its potential is well established. However, the current algorithm is an improvement in several important ways that bear on its performance:

Complexity. The present strategy exhibits total relative operator complexities that are typically about 2.5, well below earlier AMG values.

Convergence rates. Asymptotic convergence factors are more consistent and usually better than those reported in [3] and [8].

Robustness. The current algorithm is more robust in that its performance does not degrade as it did for certain pathological cases.

However, even the current strategy is not without room for improvement and is currently undergoing basic modification. (See the next Section.) We therefore defer reporting on numerical results to future efforts.

6. FURTHER DEVELOPMENTS

There are many aspects that are currently under study for AMG, but the main objective for the basic algorithm that will have general impact on AMG is to *ensure* that error smoothness is properly exploited. We discuss below two of the main approaches that we are now considering in this direction. They will be discussed briefly in the context of positive-type problems, although their generalization to other cases (especially systems) may provide an important vehicle for extending AMG to broader classes. Both approaches can be viewed as attempts to obtain information on coupling strength beyond direct connections. The first attempts to do this implicitly, while the second takes a more explicit approach.

6.1 Pre-relaxation.

For simplicity, imagine first that A is a symmetric M-matrix but is scaled so that some of its row sums are negative. Though a diagonal scaling exists that can convert A to positive-type, thus making AMG application straightforward, it can generally be too expensive to precisely determine. However, A need only be *essentially* positive-type [3] so that such scaling need only be approximate. The point here is that we would like to automatically determine an approximate sense of smoothness, that is, a sense of the nature of errors that produce small residuals for A.

The concept of pre-relaxation is to attempt this by performing relaxation on the homogeneous equation

(6.1) $AE = 0$

and examining the resulting approximate solutions, which of course are now also the actual errors. For the simplest case, pre-relaxation starts with *one* initial guess e that, without further information, is probably best chosen as the inverse square roots of the diagonal entries of A. Several Gauss-Seidel relaxation sweeps are then performed on e for (6.1) with the result (possibly normalized) defining entries of the diagonal matrix D_e. A is then replaced by $D_e A D_e$. The first level of coarsening is determined, and the pre-relaxation process is continued until all levels are chosen.

Initial experiments with a modified version of the above algorithm have proved very successful. In fact, just a few pre-relaxation sweeps starting with a constant vector seem to be enough to restore AMG efficiency for positive-type matrices that have been mis-scaled by random powers of 10 between 1 and 20.

More sophisticated vector pre-relaxation techniques would use a simultaneous set of several vectors that are forced to maintain local independence, are subjected to several relaxation sweeps for (6.1), and are subsequently examined to determine interpolation formulae that optimally represent the local subspaces that these vectors generate.

6.2. Total strength of connection

Simply stated, it is important to ensure for each F point that the strength to points $S^I(i)$ used to interpolate to it is a reasonable fraction of its total dependence. However, this strength

to the points in $S^I(i)$ should account not only for the *direct* couplings, a_{ij}, but for indirect ones as well. Thus, if i is connected to j which is not in $S^I(i)$, then we should assess the contribution to the strength of point j via its connections to $S^I(i)$. The objective here is to measure the degree of 'overlap' (described in [2; Sec. 4.4]) to ensure a suitably large γ in (3.8). Precisely how this should be done is a subject of our current efforts.

Appendix. Current AMG Coarsening Strategy

The matrix $A = (a_{ij})_{nxn}$ is assumed to be symmetric and of positive-type. We use the following notation:

$G = \{1, \ldots, n\}$ (the 'grid' points for A)

$N(i) = \{j \neq i: \ a_{ij} \neq 0\}$ (the neighborhood of a point i)

$S(i, \varepsilon) = \{j \neq i: -a_{ij} \geq \varepsilon \max_{k \neq i} -a_{ik}\}$ (the set of points on which i strongly depends relative to the parameter ε)

$S^T(i, \varepsilon) = \{j \neq i: i \text{ in } S(j, \varepsilon)\}$ (the set of points which strongly depend on i)

In addition, it is useful to have a measure of dependence of a point i on a set of points P. This can be defined in several different ways, the simplest of which is the following:

$$d(i, P) = \max_{j \in P} (-a_{ij}) / \max_{k \neq i} (-a_{ik}).$$

(An alternative is $\sum_{j \in P} (-a_{ij}) / \max_{k \neq i} (-a_{ij})$.) Using this function, the set of strong connections of a set of points, relative to ε, can be defined as follows: $S^T(P, \varepsilon) = \{j \notin P: d(j,P) \geq \varepsilon\}$. Finally, the set of points which are used to interpolate to a point i in is denoted by $S^I(i)$. In this algorithm, $S^I(i) \subset S(i, \varepsilon_2)$.

ALGORITHM A

(Initial C point choice)

Parameter ε_1 defines strong connections.

1. Set $F \leftarrow \phi$, $C \leftarrow \phi$, U (undecided) $\leftarrow G$. Set η_i to the number of elements of $S^T(i, \varepsilon_1)$ for all i.
2. Pick i in U with maximal η_i. If $\eta_i = 0$ or $U = \phi$, stop.
3. Put i in C.

   For each j in $S^T(i, \ ) \cap U$, do the following:

Put j in F.

For each k in $S(j,\varepsilon_1) \cap U$, set $\eta_k \leftarrow \eta_k + 1$

4. Go to 2.

ALGORITHM B

(Final C point choice and

interpolation definition)

Parameters: ε_2 and ε_3

Let F and C be the fine and coarse points resulting from Algorithm A. Let T be the set of F points which have been tested. Initially, $T = \phi$. Any time a point is put in C, it is automatically removed from T and, of course, F.

1. If $F \subset T$, stop. Otherwise, pick some i in $F-T$. Set $S^T(i) = S(i, \varepsilon_2) \cap C$. Set $T \leftarrow T \cup \{i\}$ and $m = 0$.

2. For each k in $S^I(i)$, set $w_{ik} = a_{ik}$. Set $w_{ii} = a_{ii}$. Set $P = N(i) \cap F$.

   a. If $P = \phi$, go to 3. Otherwise pick j in P. Set $P \leftarrow P - \{j\}$. If j is not in $S(i, \varepsilon_2)$, go to e. Otherwise, proceed with b.

   b. If $d(j, S^I(i)) \geq \varepsilon_3$*, go to e. Otherwise, set $m \leftarrow m + 1$. If $m = 1$, go to c. Otherwise, go to d.

   c. Set $j_p = j$, $S^I(i) \leftarrow S^I(i) \cup \{j\}$, and go to 2. (j becomes a conditional C point.)

   d. Put i in C. Leave j_p in F and go to 1.

   e. (Distribute the weight for point j.)

   Let $b = a_{ij} + \sum_{k \in S^I(i)} a_{jk}$.

   If $b = 0$, set $w_{ii} \leftarrow w_{ii} + a_{ij}$. Otherwise,

   for all λ in $S^I(i) \cup \{i\}$, set $w_{i\lambda} \leftarrow w_{i\lambda} + a_{ij} \cdot a_{i\lambda} / b$.

   Go to a.

3. For all k in $S^I(i)$, set $w_{ik} \leftarrow -a_{ik}/w_{ii}$. If $m = 1$, set $C \leftarrow C \cup \{j_p\}$.

   Go to 1.

*An alternative criterion here is

$$d(j, S^I(i)) \geq \varepsilon_3 \cdot (-a_{ij}) / \max_{k \neq i} (-a_{ik}).$$

This reduces complexity somewhat since it says that the dependence of j on $S^I(i)$ need not be large if dependence of i on j is not large.

The results of this algorithm are the final C/F splitting and the interpolation formula, which for any i in F and e^c in $G^c \sim C$, is given by

$$e_i = \sum_{k \in S^I(i)} w_{ik} e_k^c .$$

REFERENCES

1. R. W. Alcouffe, A. Brandt, J. E. Dendy and J. W. Painter, 'The multigrid methods for the diffusion equation with strongly discontinuous coefficients,' SIAM J. Sci. Stat. Comput. 2 (1981), 430-454.

2. A. Brandt, 'Algebraic multigrid theory: the symmetric case (a preliminary version)', Prelim. Proc.Int'l MG Conf., Copper Mountain, April, 1983.

3. A. Brandt, S. McCormick and J. Ruge, 'Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations,' report, Inst. Comp. Studies, Colo. State Univ. 1982.

4. G. Dantzig and P. Wolfe, 'Decomposition principle for linear programs,' Operations Research 8 (1960), 101-110.

5. J. E. Dendy, 'Black box multigrid,' Proc. Symp. Num. Boundary Cond. Procedures and Multigrid Meths., NASA-Ames Res. Center, 1981 (H. Lomax, ed.), NASA Conf. Publ. 2202.

6. L. M. Dudkin ad E. B. Yershov, 'Interindustry input-output models and the material balances of separate products,' Planned Economy 5 (1965), 54-63.

7. J. Ruge, 'Algebraic multigrid (AMG) for geodetic survey problems,' Prelim. Proc. Int'l. MG Conf., Copper Mountain, April, 1983.

8. K. Stuben, 'Algebraic multigrid (AMG): experiences and comparisons,' Prelim. Proc. Int'l MG Conf., Copper Mountain, April, 1983.

9. K. Tanabe, 'Projection methods for solving a singular system of linear equations and its applications,' Num. Math. 17 (1971), 203-214.

10. I. Y. Vakhutinsky, L. M. Dudkin and A. A. Ryvkin, 'Interative aggregation--a new approach to the solution of large-scale problems,' Econometrica 47 (1979), 821-841.

11. A. Brandt, 'Guide to multigrid development,' Proc. Conf. at Koln-Porz, Nov., 1981 (W. Hackbusch, U. Trottenberg, eds.), Lecture Notes Math. 960, Springer-Verlag, Berlin, 1982.