ALGEBRAIC MULTIGRID (AMG) FOR AUTOMATIC MULTIGRID SOLUTION WITH APPLICATION TO GEODETIC COMPUTATIONS*

by

A. Brandt [†]
S. McCormick[††]
J. Ruge[††]

Submitted to:
SIAM Journal on Scientific and Statistical Computing

Send all correspondence to:

Steve McCormick
Institute for Computational Studies
PO Box 1852
Fort Collins, CO 80522
(303) 491-6900

* This work was supported in part by the National Geodetic Survey (Contract No. NA82AAA02936), the NationalScience Foundation(Grant No.MCS78-03847), the Air Force Office of Scientific Research (Grant No. AFOSR82-0240) and the National Bureau of Standards

[†] Institute for Computational Studies, on leave from the Weizmann Institute of Science, Israel

[††] Institute for Computational Studies

[illegible]

ABSTRACT

Many matrix equations are either inherently discrete (e.g., geodesy) or for 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 and develops AMG and reports on numerical experiments with various matrix problems from differential equations and geodetic computations.

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 discretizations 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-to-coarse grid transfers of residuals, and coarse-to-fine grid 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.

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 from its origins and applied to a number of matrix problems. The aim of this paper is to develop an algorithm of this type, 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. Unorganized grid applications. When the given grid is not topologically piecewise uniform, such as for 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 for efficient solution, especially when the problem exhibits certain directional properties (e.g., anisotropic operators, which require either some form of block relaxation or semi-coarsening). 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 will 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 section 6. Many of these problems are such that each unknown is associated with a point of 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.

The need for basing inter-grid transfers on the strengths of the algebraic couplings has already been discovered in the application of multigrid to diffusion problems with strongly discontinuous coefficients [1]. That work led to a geometric multigrid algorithm [6] where the grid transfers are based solely on the algebraic equation. The present work, referenced briefly first in [3; Sec. 13.1], is a much further step in that direction since no explicit geometric information is required.

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., [5] and the survey in [7]). In a sense, nobody 'invented' it; multilevel organization is simply the way society itself found efficient. This concept led to iterative aggregation algorithms (cf., [7], [11]). In these algorithms, 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 to the inverse of large sparse matrices. Some of these prospects are discussed in Sections 6.7 and 7.

2. Multigrid Principles

A two-level multigrid scheme is briefly examined here in order to isolate the working principles of a successful multigrid application. Assume for focus that the problem is a continuous one of the form

$$LU = f \tag{2.1}$$

where L is a symmetric, positive definite, linear operator defined on a suitable class of functions with domain G in R^d, d a positive integer. Let G^h and G^{2h} be two uniform discretizations of G so that $G^h = \{x \text{ in } G: x = h(i_1, i_2, \ldots, i_d),$ $i_1, i_2 \ldots, i_d$ integers$\}$, and $G^{2h} = \{x \text{ in } G^h : i_1, i_2, \ldots, i_d \text{ even}\}$. Let grid transfer operators be denoted by

$$I_{2h}^{h} : G^{2h} \leftarrow G^{h} \qquad \text{(d-linear interpolation, for example)}$$

and

$$I_{h}^{2h} : G^{h} \leftarrow G^{2h} \qquad \text{(some weighted average or injection).}$$

Then let L^h and L^{2h} be defined as discrete approximations to L on G^h and G^{2h}, respectively. Let u^h denote the current approximation to the exact solution U^h of the fine grid equation

$$L^h U^h = f^h. \tag{2.2}$$

Then a multigrid cycle consists of applying some relaxation scheme, such as Gauss-Seidel or Jacobi iteration, followed by a residual equation transfer to G^{2h} via

$$L^{2h} V^{2h} = I_h^{2h} (f^h - L^h u^h). \tag{2.3}$$

This problem is then 'solved' and used to correct the fine grid solution approximation according to

$$(2.4) \qquad u^h \leftarrow u^h + I^h_{2h}V^{2h}.$$

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 (2.4) 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 L^{2h} in the right way. Note that (2.3) and (2.4) imply that the coarse grid correction is given by

$$(2.5) \qquad u^h \leftarrow u^h + I^h_{2h}(L^{2h})^{-1}I_h(f^h - L^h u^h).$$

Letting $e^h = U^h - u^h$ denote the error, then (2.5) can be rewritten as

$$(2.6) \qquad e^h \leftarrow (I - I^h_{2h}(L^{2h})^{-1}I^{2h}_h L^h)e^h.$$

Hence, the goal of the correction is roughly to eliminate error in $R(I^h_{2h})$, the range of the interpolation operator. Thus, if $e^h = I^h_{2h}\, v^{2h}$ for some v^{2h}, it is desirable that

$$(I - I^h_{2h}(L^{2h})^{-1}I^{2h}_h L^h)I^h_{2h}v^{2h} = 0$$

or, since I^h_{2h} should be full rank, that

$$I = (L^{2h})^{-1}I^{2h}_h L^h I^h_{2h}.$$

Hence,

$$(2.7) \qquad L^{2h} = I_h^{2h} L^h I_{2h}^h$$

can serve as a general prescription for the coarse-grid operator.

The two-level algorithm just described becomes a multigrid algorithm by applying it recursively; that is, a sequence of increasingly coarser grids $G^0, G^1, \ldots, G^M$ is defined so that $G^k = \{x \text{ in } G\colon\ x = 2^k h^0(i_1, i_2, \ldots, i_d),\ i_1, i_2, \ldots, i_d \text{ integers}\}$. Thus, grid k has mesh width $h^k = 2^k\, h^0$, where h^0 is the fine grid mesh width. Grid operators L^0, L^1, ..., L^M (for simplicity, L^k will sometimes be used in place of L^{h^k}, and similarly for other operators) are defined on all grids, and the number ν of relaxation sweeps per grid is specified. Given an approximation, u^h, to the solution of (2.2), then one multigrid cycle on grid $h = h^i$ is denoted by $u^h \leftarrow MG(h,\ u^h,\ f^h)$ and defined recursively by:

```
If $h = 2^M h^0$,    set $u^h = (L^h)^{-1} f^h$.
Otherwise do the following:
    Perform $\nu$ relaxation sweeps on (2.2);
    set $u^{2h} = 0$ and $f^{2h} = I_h^{2h}(f^h - L^h u^h)$;
    set $u^{2h} \leftarrow MG(2h, u^{2h}, f^{2h})$;and
    set $u^h \leftarrow u^h + I_{2h}^h u^{2h}$.
```

There are many possible variations of this algorithm, but this serves to illustrate the recursive nature of multigrid.

Since the solution process is essentially the same on all grids, in case L^h is a symmetric, positive definite operator, it is best if L^{2h} is also. This is easiest to ensure by defining L^h by (2.7) with

$$(2.8) \qquad I_h^{2h} = (I_{2h}^h)^T.$$

Note that properties (2.7) and (2.8) completely define the coarse grid problem in terms only of interpolation and the fine grid operator.

To motivate (2.7) and (2.8) further, note for symmetric, positive definite L^h that Gauss-Seidel, as well as many other relaxation methods, can be viewed as a minimization algorithm for the 'energy' functional

$$(2.9) \qquad F^h(u^h)=u^{hT} L^h u^h-2u^{hT} f^h.$$

(Note that $F^h(u^h) - F^h(U^h) = e^{hT} L^h e^h$, the energy norm of the error.) Assume that G^{2h} and I^h_{2h} are given. Then, with an approximation u^h to U^h, it is optimal in terms of F^h to define the coarse grid problem (2.3) as that of finding a V^{2h} in G^{2h} that minimizes over G^{2h} the functional

$$(2.10) \qquad F^h(u^h+I^h_{2h}v^{2h}) = F^h(u^h)+v^{2h^T} (I^{h^T}_{2h}L^h I^h_{2h})v^{2h} - 2v^{2h^T} (I^{h^T}_{2h}(f^h-L^h u^h)).$$

(2.7) and (2.8) are then direct results of this discrete variational formulation. The main advantage of this variational formulation is that (2.10) completely eliminates the error in the range of interpolation. It is therefore sufficient to ensure that this range properly represents the error components not reduced by relaxation (cf., [9]).

3. AMG for Symmetric Positive-Type Matrices

The discrete variational formulation can, therefore, be used in the symmetric case as a basis for constructing the multigrid processes once choices are made for the coarse levels G^ℓ and their interpolation operators $I^{\ell-1}_\ell$, $1 \leq \ell \leq M$. To understand how these choices can be made based solely on the entries of the fine grid matrix L^o, it is first important to understand the concepts of strong dependence and smoothing as algebraic properties. This is done by way of example with the following class of matrices.

Definition. A symmetric matrix is positive-type if all of its off-diagonal entries are nonpositive and each row sum is nonnegative. If each row sum is zero, it is called a zero row sum matrix.

Unless otherwise stated, it will henceforth be assumed that L^o is positive-type and zero row sum. The latter assumption is for convenience in describing efficient coarsening and relaxation processes, but as noted in Section 3.1 is otherwise inessential. For a discussion of nonpositive-type matrices, see Section 4.

An important algebraic concept is that of a neighborhood, which is based on the following definition.

Definition. Variable i is strongly dependent on variable k with strength $\alpha > 0$ if

$$-L^o_{ik} \geq \alpha \max_j (-L^o_{ij}) \tag{3.1}$$

The quantifier 'with strength α' will be omitted when it is clear from context or when quantification is unnecessary. (Note that the maximum must be attained over $j \neq i$ since $L^o_{ii} \geq 0$, and that this maximum is positive unless the ith row of L^o is null. Moreover, because (3.1) is a relative measure, the strength of dependence of i on k may be arbitrarily different from that of k on i.) In what follows, local and neighborhood will be used as algebraic concepts based on strong dependence; e.g., k and i are direct neighbors if i depends strongly on k or k depends strongly on i.

The uniqueness of AMG lies in its attempt to automatically design the process of relaxation and coarsening. (Coarsening here refers to coarse level selection and interpolation prescription.) This design is a systematic problem since these processes should not be determined independently of each other (cf., [3; Section 3.3]). A basic decision made for the current AMG is to use point relaxation (e.g., Gauss-Seidel with C-F ordering described in Section 3.4) because its smoothing properties facilitate coarsening and because block relaxation (e.g., line relaxation) is inconvenient to automate (e.g., since lines are unspecified). In fact, the matrix equations which would arise in block relaxation, one matrix for each block, may best be solved by AMG based on point relaxation, thereby obviating the need for such blocks as anything but a conceptual tool.

It is not difficult to prove that properly chosen point relaxation quickly reduces errors that show large residuals compared with their own size, that is, errors that satisfy

$$(3.2) \qquad ||L^o e^o|| \sim ||L^o||.||e^o||.$$

More precisely, proper point relaxation will always converge efficiently until a stage is reached in which the errors produce relatively small residuals:

$$(3.3) \qquad ||L^o e^o|| \ll ||L^o||.||e^o||$$

Then and only then must point relaxation slow to an unacceptable rate, but it is just for such errors that smoothness occurs and may be used as a basis for coarsening strategies. Specifically, errors that satisfy (3.3) approximately lie in an invariant subspace of L^o corresponding to its small eigenvalues. For positive-type (and other) operators L^o, any vector e^o in such a subspace exhibits smoothness in the sense that it is approximately a local constant, that is,

$$(3.4) \qquad e_k^o \sim e_i^o \qquad \text{if } i \text{ strongly depends on } k.$$

Another way to see that error smoothness is produced directly by point

Gauss-Seidel is to interpret it as a local averaging process. Indeed, in terms of the error $e^o = U^o - u^o$, this relaxation is given by

(3.5) $$e_i^o \leftarrow \sum_{k \neq i} - \frac{L_{ik}^o}{L_{ii}^o} \quad e_k^o$$

where

(3.6) $$\sum_{k \neq i} - \frac{L_{ik}^o}{L_{ii}^o} = 1$$

The coarsening strategy is described in Section 3.2. It is important first to discuss the assumption of zero row sum.

3.1 Zero row sum/slack variable

Any positive-type matrix that has positive row sums can be altered to one with only zero row sums by augmenting it to include a slack variable. This is done by providing for each equation the slack variable with a coefficient equal to the negative of the original coefficient sum for that equation. The augmented L^o will maintain symmetry if the equation associated with the slack variable is defined as the negative sum of all the other equations. It is for convenience of concept and practice that such a variable is used in AMG when necessary to obtain zero row sum.

This process leads to a matrix that is, of course, singular: the vector θ, consisting of all 1's, constitutes a basis for its null space. There is, however, no cause for concern since any solution, u^o, of the augmented problem for L^o gives the unique solution of the original equation by the correction

(3.7) $$u^o - u^o - u_o^o \, \theta \quad ,$$

where u_o^o is the value of the slack variable. Note that the residual

$$r^o = f^o - L^o u^o$$

is unchanged by (3.7). Note also that (3.7) is a perfectly smooth change to the error so that it need only be performed on the coarsest level.

The use of a slack variable for positive-type problems is not only convenient but leads naturally to effective relaxation and coarsening procedures. AMG can easily be developed without it, but this is recommended only after acquiring an understanding of the role of the slack variable in each process.

3.2 <u>Coarsening (coarse level selection and interpolation prescription)</u>

There are two general principles guiding the coarsening process of AMG:

<u>Principle A.</u> The <u>total relative complexity</u> (that is, the ratio of the total over ℓ of the number of nonzero L^{ℓ}_{ij} to the number of nonzero L^{o}_{ij}) must be suitably bounded.

<u>Principle B.</u> Interpolation to any fine level variable must include coarse level variables on which it strongly depends.

Principle A is obvious and is concerned with the cost of a multigrid cycle. Principle B, which aims at obtaining good convergence rates, is derived from the fact that point relaxation smooths only along strong dependencies.

Although these loosely stated principles lead to many different coarsening techniques, the following one emerged as the most effective of those thus far analyzed.

G^{ℓ} will denote the set of level ℓ variables or unknowns. (Alternately, one may think in terms of gridpoints.) Principle B above assumes G^{ℓ} to be a subset of $G^{\ell-1}$. The interpolation from G^{ℓ} to $G^{\ell-1}$ is denoted by $I^{\ell-1}_{\ell}$ so that the level ℓ problem will be given by $L^{\ell}U^{\ell}=f^{\ell}$,where $L^{\ell} = (I^{\ell-1}_{\ell})^T L^{\ell-1} I^{\ell-1}_{\ell}$ and $f^{\ell} = (I^{\ell-1}_{\ell})^T(f^{\ell-1} - L^{\ell-1} u^{\ell-1})$, $1 \leq \ell \leq M$, and $L^o U^o = f^o$ is the given (finest level) problem.

The objectives of the coarsening process described below is to determine M, G^1, G^2, ..., G^M and I_1^0, I_2^1, ..., I_M^{M-1}. This is done recursively: the construction of G^{ℓ} and $I_{\ell}^{\ell-1}$ is based on the previously constructed matrix $L^{\ell-1}$, which for simplicity is assumed to be zero row sum and positive-type. (See, however, Section 3.4)

It is convenient to first describe the interpolation matrix $I_{\ell}^{\ell-1}$. To this end, let $C^{\ell-1} = G^{\ell}$ and $F^{\ell-1} = G^{\ell-1} - G^{\ell}$. (The superscript ℓ-1 will be omitted whenever it is clear from context.) The interpolation presently used in AMG depends on a prescribed integer $p \geq 1$ and is given by the following definition.

__Definition.__ For a given point i of $F^{\ell-1}$, let $k_1, \ldots, k_{p_i}$ denote p_i distinct points of $C^{\ell-1}$ that give the largest values of $-L_{ik}$, where p_i is the minimum of p and the number of nonzeros L_{ik}. Then __p-point interpolation__ $I_{\ell}^{\ell-1}$ is defined by $u^{\ell-1} = I_{\ell}^{\ell-1} u^{\ell}$, where $u_i^{\ell-1} = u_i^{\ell}$ when i is a point of $C^{\ell-1}$ and

$$(3.8) \qquad u_i^{\ell-1} = \frac{\sum_{j=1}^{p_i} I_{ik_j}^{\ell-1} u_{k_j}^{\ell}}{\sum_{j=1}^{p_i} L_{ik_j}^{\ell-1}}$$

when i is a point of $F^{\ell-1}$.

Describing how AMG selects the coarse levels $C = G^{\ell}$ is not so simple. Since this rather complex process is described in some detail in the appendix, the discussion here will be limited to a brief account of its motives. It is first convenient to list the basic objectives of the coarse level selection process, which depend on the following parameters: positive integers p,t and a positive real α.

__Objective 1.__ For each i in F, let p_i be as above and let β be the strength of the p_i^{th} strongest point on which it depends. Then there must exist p_i points $k_1, k_2, \ldots, k_{p_i}$ in C on which i depends with strength at least $\alpha\beta$.

<u>Objective 2.</u> If there are at least t nonzeros in row i, then i must be in C.

<u>Objective 3.</u> F should have as many points as is practically possible without violating objectives 1 and 2.

<u>Objective 4.</u> The coarsest level, G^M, should have only a small number of points.

Objective 1 is an attempt to supply each F point with strong dependence on p C points. Using point relaxation as AMG presently does, it is clear from earlier work with anisotropic operators (cf. [3; Sec.3.3]) that strong dependence on to coarse grid points is necessary for the coarse grid to properly correct the smooth error. There are compelling reasons why proper p-point interpolation is sufficient for many problems. To see this, suppose that each variable depends strongly on many other variables, with all these many dependencies being comparably strong. (The case of many strong dependencies for just some exceptional variables can be excluded, due to Objective 2). Relaxation would then efficiently smooth along <u>all</u> these dependencies, so a small number of them would be enough to use in interpolation. This can more easily be viewed in <u>geometric</u> framework, where each variable is associated with a point in a low-dimensional space (the 'underlying space'), and couplings $L^o_{ij} \neq 0$ are between variables i and j associated with neighboring points. It is easy to see then that the radius of smoothing (i.e., the wavelength of error components efficiently reduced by relaxation) is the same as the radius of strong couplings. Hence, even if the number of strong couplings per point is large, just a few of them at a time are enough to interpolate a reasonable approximation to the (relaxed) error. Actually, this consideration also suggests a certain connection between the dimension d of the underlying space and the number of points p that should be used in each interpolation. It seems that $p \geq d + 1$ should be used if one wants to avoid large local jumps in values of the interpolant.

This reasoning stems from early attempts to devise an AMG algorithm where in place of Objective 1 was the condition that, for each F point, no less than a fixed fraction δ (e.g., $\delta=0.5$) of the total strength of its dependence must be on C points (i.e., $-\sum_{k\in C} L_{ik} \geq \delta L_{ii}$ for each i in F). Convergence rates were very attractive, but the complexity was quite unstable. The point is that the strong dependence on C points should be based not on their total value but on how many there are. This will maintain convergence yet stabilize complexity.

To see heuristically how complexity is controlled, assume again a geometric background where $j_h h$ is the 'radius' of the operator on level h in that geometry and $k_h h$ is the radius of interpolation. Then on the next coarser level (say of 'mesh size' H), it follows from (2.7)-(2.8) that

$$j_H \leq \frac{2h}{H} k_h + \frac{h}{H} j_h \qquad \text{(approximately)}. \tag{3.9}$$

Hence, complexity is eventually (i.e., for much coarser grids) determined by k_h, not j_h. If only p of the 'nearest' neighbors are used in interpolation, then k_h is roughly fixed. If the ratio h/H is also approximately fixed, then (3.9) implies that the radius of L^H will eventually be bounded according to

$$j_H \leq \frac{2h}{H-h} k_h. \tag{3.10}$$

(In two dimensions, for example, keeping $k_h \sim 1$ by choosing p=3 yields $H \sim \sqrt{2}\,h$ so that $j_h < 5$. Though this bound is independent of h, numerical experiments indicate that it is much too pessimistic. The reason is that locally the coarsening is roughly in one direction, and in that direction $H \sim 2h$, so (3.10) yields $j_H \lesssim 2$.)

Objective 2 is important for controlling complexity since coarse grid dependencies are created predominantly by F-to-F fine grid connections. It is especially important that points like the slack variable that are more or less global are kept for the coarse level.

The aims of Objectives 3 and 4 in terms of controlling complexity are obvious. Objective 3 is presently achieved by examining the points of G in turn and making them F points whenever the other objectives would not thereby be violated. Objective 4 is achieved simply by prescribing when the coarsening process is to be stopped.

A description of the coarsening process based on these objectives is given in the appendix. It is presently based on sweeping over the points of G to select C points that should contribute most to satisfying Objectives 1 and 2 for as many F points as possible. There are, however, several variations of this process currently under study.

Another coarsening process is given in terms of a sequence of nonnegative reals $(\varepsilon^o, \varepsilon^1, \dots, \varepsilon^M)$.

Operator truncation: If $L^\ell_{ij} \le \varepsilon^\ell |L_{ii}|$ for some $j \neq i$, then the value of L_{ij} should be set to zero and the diagonal adjusted to maintain zero row sum.

The purpose of operator truncation is again to control complexity. However, the use of operator truncation can be dangerous. Unless ε^ℓ is tied to the size of G^ℓ, then slow convergence rates may result. For example, if L^ℓ is a discretization of a second-order d dimensional partial differential operator, then the truncation sequence should satisfy $\varepsilon^\ell << O(N_\ell^{-1/d})$, where N_ℓ is the number of points in G^ℓ. Otherwise, the modified L^ℓ will be an arbitrarily bad approximation to the original L^ℓ in the smooth error components. If d is unknown, then this criterion can be used safely with d=1.

3.3 C-F Relaxation.

Another motivation for AMG is to view it as a sort of approximate reduction-type algorithm. Briefly, suppose C and F could be arranged so that all of the connections from any point in F are necessarily in C. Then simply relaxing on the F points alone guarantees that the residual error at each point is zero, that is, the actual error is in the range of interpolation. This in turn implies that a coarse grid correction based on the variational principle (2.10) produces an exact solution. Of course, the difficulty with this reduction approach is that Objective 3 is not achieved. In fact, the complexity principle is generally violated.

Nevertheless, this viewpoint motivates the concept of performing F point relaxation before the coarse grid correction step in AMG. But because the error is no longer expected to lie in the range of interpolation, it is important to smooth this error beforehand. This suggests the use of C-F relaxation which resembles the concept of red-black ordering (cf., [8]): Gauss-Seidel relaxation is first performed (via some ordering) on the C points with a subsequent sweep (via some ordering) over the F points. This constitutes one C-F sweep, which is the basis for relaxation in the current AMG.

Analogously to red-blck schemes, the C-F ordering is expected to be suitable for parallel or vector processing, where simultaneous relaxation is performed first on the C points and then on the F points.

3.4 Maintenance of positive-type

In the above construction of G^ℓ and $I_\ell^{\ell-1}$, with L^ℓ then given by (2.7), it was assumed that $L^{\ell-1}$ is symmetric, positive-type, and zero row sum. These properties are assumed for L^o, but use of this construction recursively requires that they hold for L^1, L^2,..., L^{M-1} as well. So assuming they hold for $L^{\ell-1}$, they will now be proved for L^ℓ.

Theorem. Using p-point interpolation (3.8), then the operator L^ℓ constructed according to (2.7) and (2.8) is symmetric, nonnegative-definite, and zero row sum. For p=1, it is positive-type.

Proof. Let θ^ℓ be the vector associated with G^ℓ, all of whose entries are unity, $0 \leq \ell \leq M$. By (3.8) it follows that $I_\ell^{\ell-1}\, \theta^\ell = \theta^{\ell-1}$. Hence, by (2.7) and because $L^{\ell-1}\, \theta^{\ell-1} = 0$, then $L^\ell \theta^\ell = 0$. Thus, zero row sums are maintained. The symmetry and semi-definiteness of L^ℓ follow trivially from (2.7) and (2.8).

To prove the assertion for p=1, let e_j^ℓ denote the j th column of the identity matrix id^ℓ for G^ℓ and let $E_j^{\ell-1} = I_\ell^{\ell-1}\, e_j^\ell$. Note that

$$L_{ik}^\ell = e_i^{\ell^T} (I_\ell^{\ell-1^T} L^{\ell-1}\, I_\ell^{\ell-1})\, e_k^\ell = E_i^{\ell-1^T}\, L^{\ell-1}\, E_k^{\ell-1}$$

But with p=1 it follows that the vectors $E_i^{\ell-1}$ and $E_k^{\ell-1}$ have no nonzero entries in common. Thus, L_{ik}^ℓ is a sum of off-diagonal entries of $L^{\ell-1}$, all of which are nonnegative, and the theorem is proved.

It is generally possible to maintain positive-type only for p=1, so some positive off-diagonal entries of $L^1, \ldots, L^M$ are to be expected in general. However, this is not a real difficulty since these entries can never be very large. Indeed, the above processes do not really require any of the matrices, including L^o, to be positive-type. All that is needed is that the coefficients $L_{ij}^{\ell-1}$ used in the p-point interpolation (3.8) do not change sign. Thus, it is sufficient that the dominating dependencies are of the same sign. Operators that satisfy this condition may be called essentially positive-type. For any reasonable value of p, if L^o is essentially positive-type, then so also will be L^ℓ. In fact, any discretization of a second-order elliptic differential operator produces such an L^o. (Such operators can also be turned into strictly positive-type by the addition of certain variables; cf., [2].) The results of numerical experiments with AMG applied to such operators show the same performance as for operators that are strictly positive-type.

4. Remarks on the General Case

4.1 Relaxation

For symmetric semi-definite matrices, point Gauss-Seidel relaxation (preferably with $C-F$ ordering) is probably the best choice without additional information. Several operators L^h should, however, require a 'distributive' scheme [3; Sec. 3.4]. It may be advisable to use some form of 'collective' relaxation [3; Sec. 3.4] if it is known that L^h is derived from a system, that is, if there is an underlying geometric set of points, each of which is associated with a fixed number of variables. Collective relaxation is still point oriented, but it involves simultaneously solving all the equations associated with each point. For this purpose, it is necessary, of course, to specify in advance what unknowns and what equations are associated with each point.

4.2 Global Constraints

Many problems involve 'global' constraints, such as integral normalization conditions where some norm of U^h is prescribed to uniquely determine the solution. Even conditions which look like pointwise ones (e.g., prescribing the origin in the geodetic problem) are actually global in the sense that the entire solution is sensitive to it. The requirement that the slack variable vanish is such a global condition. In any case, these conditions should not be incorporated into relaxation since its objectives are merely local. In fact, enforcing them may have adverse effects on local smoothing. They should be enforced rather on the coarsest levels after making sure that their residuals are properly transferred from the finer levels.

For this purpose, it is desirable that the AMG user specify which equations represent global conditions.

4.3 Coarsening

For non-symmetric problems, prescription (2.8) is not necessarily the best. Thus, the coarsening process should now determine both I_{2h}^{h} and I_{h}^{2h}. The main objective of the coarsening process should generally be to calculate good approximations to errors that satisfy (3.3). Without specific information, for general L^h it is possible to define strong dependence and base interpolation on the rows of L^h as before, or, instead, the rows of L^h+L^{hT}. Instead of the adjoint of interpolation, residual weighting should be based on the rows of L^{hT}. Techniques of this type are currently being studied in connection with AMG.

Other possible strategies for treating general L^h include appeal to a variational formulation if one exists or use of the matrix transpose by forming (implicitly or explicitly) L^hL^{hT} or $L^{hT}L^h$. These are also currently under study.

4.4 Coarsening for a system

For systems that include q>1 variables per grid point (arising, e.g., from discretization of a system of q differential equations), the matrix will generally not even be essentially positive-type. Most often in such (and other) situations, one can partition the vector of unknowns U^o into q subvectors (each with exactly one entry per grid point, i.e., each corresponding to the discretization of one of the unknown functions), so that the principal submatrix of L^o corresponding to each subvector is essentially positive-type. In this case, when relaxation slows down, so that (3.3) is satisfied, smoothing may occur within each subvector. Interpolation can therefore be designed accordingly, provided the partition of subvectors has been given. This partition should be known at the origin of the problem since the variables of different subvectors normally have different physical meaning.

In some more difficult cases, the given q unknowns at each geometrical point must be transformed into a new set of q unknowns before smoothing can occur within each subvector (see examples in Sec. 6.3). Such transformations can be determined and applied automatically by AMG, although they are not in the current version.

Another possible approach for a system is to coarsen in terms of the given geometric points; namely, to choose which grid points will appear, each one with all its associated unknowns, on the coarse level. Objective 1 should then be replaced by a more complex one which requires not only that any variable have strong dependence on the coarse grid variables, but also that each linear combination of the q variables at any point have such strong dependency. Coarsening algorithm which incorporate such requirements are now under development.

5. Numerical Results for PDE's

This section reports on the results of several representative experiments with AMG on matrices that arise in solving two-dimensional partial differential boundary value problems. For each example, the region is the unit square $\Omega = [0,1]x[0,1]$, the boundary conditions are Dirichlet with zero data, and the discretization is conventional central finite differences.

The examples include:

1. The diffusion problem

$$-(d_1 u_x)_x - (d_2 u_y)_y = f$$

where the functions d_1 and d_2 were chosen as follows:

a. (Strongly discontinuous)

$$d_1(x,y) = d_2(x,y) = \begin{cases} 10^3 & 0.25 \leq x,y \leq 0.75 \\ 1 & \text{otherwise} \end{cases}$$

b. (Widely varying in direction and magnitude)

$$d_1(x,y) = 10^{3(x-y)^2}$$

$$d_2(x,y)=1+10^3 \sin (\pi\ xy)$$

c. (Singularity)

$$d_1(x,y) = d_2(x,y) = x^2 + y^2$$

d. (Anisotropic)

$$d_1(x,y) \equiv \varepsilon \ , \qquad d_2(x,y) \equiv 1 \ .$$

In all cases the usual 5-point (divergence-form) discretization was used.

2. The Laplacian plus a strong cross-derivative term

$$-\Delta u + \varepsilon\, u_{xy} = f,$$

where

$$-2 \leq \varepsilon \leq 2.$$

The experiments were with the seven-point stencil given by

$$L_\varepsilon^h = \frac{1}{h^2}\begin{pmatrix} 0 & -(1+\frac{\varepsilon}{2}) & \frac{\varepsilon}{2} \\ -(1+\frac{\varepsilon}{2}) & 4+\varepsilon & -(1+\frac{\varepsilon}{2}) \\ \frac{\varepsilon}{2} & -(1+\frac{\varepsilon}{2}) & 0 \end{pmatrix}.$$

Thus,

$$L_2^h = \frac{1}{h^2} \begin{pmatrix} 0 & -2 & 1 \\ -2 & 6 & -2 \\ 1 & -2 & 0 \end{pmatrix}$$

and

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

The examples were run on a CYBER 720 with h = 1/32 using a 3-point interpolation version of AMG incorporating V-cycling with two relaxation sweeps per level (one each before and after coarse level correction). The convergence factors, shown in Table 1, are asymptotic in the sense that the runs were carried out until the per cycle rates stabilized. Thus, the rates are much better for the first few cycles, which is more cycles than is usually needed in practice.

There are several things to note about these results. First, the asymptotic convergence factors are generally at most 0.25. In fact, the less well-behaved the problem (i.e., the farther from a Poisson problem), the better is this rate. Note that Problem 2 withε=-2.0 converges in one cycle. This is because AMG recognizes it as a one dimensional problem for which multigrid is a direct method.

The only exception seems to be Problem 2 with $\varepsilon > 1$. The bad rates are due not so much to the fact that this problem is far from positive type, but to the fact that discretization here is rather bad: it shows <u>no</u> dependence in the direction in which the differential operator has its main dependence (its <u>only</u> dependence, in case $\varepsilon = 2$). In fact, the slow convergence is for components for which the discretization error is large and hence not much convergence is desired anyway.

Finally, each of the experiments involved setup and per-cycle costs (in floating point operations, or FOPS) that were both typically about 15 work units (i.e., matrix multiplies). These results are from an earlier research version of AMG and both factors should be substantially reduced (probably by a factor of 2) by developing a more efficient AMG implementation. Nevertheless, this and work with other larger problems verify the O(N) complexity of AMG. (The fact that the cost and rate results in Table 1 are essentially independent of N was verified by several experiments with h ranging from 1/16 to 1/128.) The relative total complexity (defined in Sec. 3.2) will improve in more complex problems (cf. Sec. 6.6).

6. Geodetic Equations

6.1 General nonlinear weighted least squares

Geodetic adjustment involves nonlinear least squares which is generally a problem of minimizing a functional

$$(6.1) \qquad P(u) = \Sigma w_k g_k^2 (u)$$

over a vector space of unknowns $u=(u_1 ,u_2 ,\ldots)$, where the w_k are prescribed weights and each $g_k=0$ represents an observation equation (which is assumed to be approximately zero at the minimum of P). Assuming that u approximately minimizes P, then each $g_k(u)$ is small and yields the following simplification:

$$(6.2) \qquad \frac{\partial^2 P}{\partial u_i \partial u_j} = 2 \sum_k w_k \left(g_k \frac{\partial^2 g_k}{\partial u_i \partial u_j} + \frac{\partial g_k}{\partial u_i} \frac{\partial g_k}{\partial u_j} \right)$$

$$\sim 2 \sum_k w_k \frac{\partial g_k}{\partial u_i} \frac{\partial g_k}{\partial u_j}$$

An approximate Newton correction, E, to u is then given by the solution of

$$(6.3) \qquad L^o E = f^o$$

where

$$L^o_{ij} = \sum_k w_k \frac{\partial g_k}{\partial u_i} \frac{\partial g_k}{\partial u_j}.$$

and

$$f^o_i = - \sum_k w_k g_k \frac{\partial g_k}{\partial u_i}.$$

Thus, L^o is symmetric and positive semi-definite near the minimum. Note that a good stopping criteria for AMG is that the change in P per cycle is small compared to P, because further changes will be much smaller yet.

6.2 Geodetic least squares

For a comprehensive description of geodetic problems, the reader is referred to [10]. In the present paper, it is assumed that the geodetic stations lie on a bounded plane. (This is no real loss of generality since the solution process for the nonplanar geoid is almost identical. In any case, this is assumed only to simplify the discussion since the numerical results are actually for the nonplanar model.)

The basic unknowns are the latitude, x_i, and longitude, y_i, of each geodetic station i. In addition, there is the artificial unknown, z_l, called orientation, which is the reference direction used in the angle measurements. Note that the vector of unknowns is $u = (x,y,z)^T$. There are two types of observations g_k involving stations i=i(k) and j=j(k), namely, directional

$$(6.4) \qquad g_k(u) = \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) - z_l - \alpha_k ,$$

where α_k is the angle measured between station i and j from direction $z_l=z_{l(k)}$, and distance

$$(6.5) \qquad g_k(u) = r_{ij} - \delta_k,$$

where δ_k is the distance measured between stations i and j and

$$r_{ij} = ((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}.$$

The entries of L^o in (6.3) corresponding to directional observations are given in terms of the quantities

$$(6.6)\qquad \frac{\partial g_k}{\partial x_i} = -\frac{\partial g_k}{\partial x_j} = \frac{y_i - y_j}{r_{ij}^2}$$

$$\frac{\partial g_k}{\partial y_i} = -\frac{\partial g_k}{\partial y_j} = \frac{x_j - x_i}{r_{ij}^2}$$

$$\frac{\partial g_k}{\partial z_l} = -1$$

Notice that the x variables are strongly dependent on stations with relatively similar latitudes (vertical dependence) while the y are on those with similar longitudes (horizontal dependence). x has only weak horizontal and y has only weak vertical dependencies. Moreover, there is no (direct) dependence between distinct z variables.

The entries of L^o for distance observations involve

$$\frac{\partial g_k}{\partial x_i} = -\frac{\partial g_k}{\partial x_j} = \frac{x_i - x_j}{r_{ij}}$$

$$(6.7)\qquad \frac{\partial g_k}{\partial y_i} = -\frac{\partial g_k}{\partial y_j} = \frac{y_i - y_j}{r_{ij}}$$

$$\frac{\partial g_k}{\partial z_l} = 0$$

Notice that the geometric reltionships are now reversed: the vertical dependencies for x and the horizontal dependencies for y are weak.

The linearized equation (6.3) can be written in block form as

$$(6.8)\qquad \begin{pmatrix} XX^T & XY^T & XZ^T \\ YX^T & YY^T & YZ^T \\ ZX^T & ZY^T & ZZ^T \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} X^TF \\ Y^TF \\ Z^TF \end{pmatrix}$$

where entries of X, Y and Z are given in (6.6) and (6.7). Thus,

$$(6.9)\qquad L^o = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}^T .$$

By virtue of (6.6) and (6.7), XX^T and YY^T are positive-type and zero row sum, ZZ^T contains only zero and ones, and Z^TZ is a diagonal matrix whose diagonal entries are just the total number of observations in which each orientation appears.

Note that (6.8) contains no artificial constraints such as the specifications of a particular station as the origin. Thus, U in this model problem is determined at most up to origin shifts and rotations. This singularity is no difficulty, however, since such constraints can be satisfied after the approximation has been computed. Inclusion of these constraints in L^o would, in fact, cause AMG to introduce a slack variable and impose the constraints only at convergence. This would be done implicitly by performing (3.7).

Since L^o itself is not positive-type, AMG must be modified to apply here. It is first useful to consider the following examples.

6.3 Simple examples

In the following examples, it is assumed that the stations (approximately) coincide with the nodes of a uniform grid in the plane and that $w_k = 1$. Each example is characterized by the specific type of observation performed over all of its stations.

a. Distance to 8 nearest neighbors: The equations (6.8) turn out to be the same as those arising in the simplest central finite difference discretization of the differential system

$$\begin{pmatrix} -3\partial_{11} -2\partial_{22} & -4\partial_1\partial_2 \\ -4\partial_2\partial_1 & -2\partial_{11} -3\partial_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f^1 \\ f^2 \end{pmatrix}$$

where ∂_1 denotes differentiation in the horizontal direction, ∂_2 vertical, $\partial_{jj} = \partial_j\partial_j$ and f^j are some functions. It is easy to see that this system is elliptic and that lexicographically ordered Gauss-Seidel has a multigrid smoothing rate of 0.53. Thus, one would expect AMG in this case to converge at a rate similar to that for conventional multigrid.

b. **Distances to 4 nearest neighbors:** The equations (6.8) are now equivalent to the discretization of the differential operator

$$\begin{pmatrix} -\partial_{11} & 0 \\ 0 & -\partial_{22} \end{pmatrix}$$

which is semi-elliptic. Coarsening can, therefore, be done within each variable x and y individually, with x being coarsened horizontally and y vertically. The lexicographic Gauss-Seidel smoothing rate in this semicoarsening process is then 0.45 (see [3; Sec. 3.3]).

c. **Distance to 4 diagonal neighbors:** The equations (6.8) are as those that arise in discretizing

$$\begin{pmatrix} -2\partial_{11} \quad -2\partial_{22} & -4\partial_1\partial_2 \\ -4\partial_1\partial_2 & -2\partial_{11} \quad -2\partial_{22} \end{pmatrix}$$

This is again semi-elliptic, but semicoarsening individually in x and y as in the previous example would now fail, since it is in terms of other variables (x+y and x-y) that we now have connections in only one preferred direction. Although such degeneracies are perhaps unlikely to happen in a real geodetic problem, a robust algorithm must take them into account, since a local <u>tendency</u> toward such degeneracy may occasionally appear.

6.4 <u>Numerical results: a scalar case</u>

The tests reported in the following sections were based on a data set supplied by Dr. Alan Pope of the National Geodetic Survey. It represents a latitude-longitude adjustment problem for 250 stations and includes 1941 observations on 250 latitude, 250 longitude, and 324 orientation unknowns. As before, all runs were made on a Cyber 720 using two relaxations per level and

3-point interpolation.

These experiments are designed to seperate the two main difficulties associated with multigrid solution of the geodetic problem: the very unorganized grid structure and the fact that it is a system of three unknowns (x, y, and z). The latter difficulty is not the essential one since various non-scalar systems have been previously solved by multigrid processes. On the other hand, the unorganized nature of the geodetic grids is one of the main reasons AMG was conceived. The first test isolates this feature by focusing on a related scalar problem. (Such problems actually arise in geodetic contexts, such as elevation adjustment, and are very common in finite element discretizations.) To expedite the effort, this test was developed as a single variable version of (6.8), namely,

$$(6.9) \qquad XX^T x = X^T F,$$

which can be thought of as a latitude adjustment problem, where the longitudes y and orientations z are given. For this problem, AMG achieved an asymptotic per-cycle convergence factor of 0.22 and a total relative complexity (Section 3.2) of 3.5. This is similar to the results reported in Section 5.

6.5 Numerical results: two unknown functions

The next test included the two unknown functions x and y, with orientation omitted, which is represented by the system

$$(6.10) \qquad \begin{pmatrix} XX^T & XY^T \\ YX^T & YY^T \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X^T F \\ Y^T F \end{pmatrix} .$$

There are several versions of AMG that can be adopted for this system (cf., Section 4.4), but acceptable results have already been achieved for a simplified process based on viewing (6.10) as a point-oriented block system. For this process, coarsening is based on taking the x-y pairs associated with each geometric point (or station) and making decisions based on the dependence between these pairs. In the version tested, point i was judged to be strongly dependent on point j if some linear combination of the associated x-y pairs exhibited strong dependence. Note that this can be easily decided by computing, say, the maximum row sum norm of $A_{ii}^{-1} A_{ij}$, where

$$A_{ii} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + A_{ij} \begin{pmatrix} x_j \\ y_j \end{pmatrix}$$

are the terms in the equations for point i involving points i and j. This process also incorporates simultaneous relaxation where each x-y pair is in turn changed in order that both equations at the associated point are satisfied. Because the convergence rates were somewhat degraded over the scalar case, an F-cycle (cf., [4]) version of AMG was used, yielding an asymptotic per-cyle convergence factor of 0.36 and a total relative complexity of 3.7.

These results can be much improved with the introduction of a fully reliable coarsening process. More specifically, coarsening should ensure for each F point i that every possible rotation on the pair (x_i, y_i) will have two or three strong dependencies on C points. (The 3-point interpolation process used thus far does not generally supply that many strong dependencies because many of the A_{ij} matrices are actually singular. See example b and c in Section 6.3.) To test this claim, the same run was made except with 5-point interpolation. Again, this will not guarantee that the objective here is reached, but for the small model problem it should be close--and it requires no additional programming. The asymptotic per-cycle convergence rate was 0.10 with a total relative complexity of 4.6. Thus, though the waste in this expiditious test is evident in the increased complexity, the rate appears to support the potential of the new approach.

6.6 Comments on complexity

Total relative complexity is the total complexity (i.e., number of nonzero coefficients) of the matrices on all levels relative to that of L^{o}. This gives a measure of the cost of a V-cycle but not an F-cycle, so the results of the previous section must be considered with this in mind. However, these results are only preliminary with much room for improvement, so they should not in any case be taken too seriously.

By geometrical arguments (cf. Section 3.2), the complexity per station of the coarse levels for large problems is essentially only dependent on the complexity per station of interpolation, not on L^{o}. Hence, the total relative complexity would be smaller accordingly as the complexity per station of L^{o} is larger. Since the complexity per station of the North American geodetic problem is roughly four times that of the present model, the total relative complexity for this problem should be substantially smaller than the present results.

6.7 Further research for geodetic equations

Following is a list of topics for further research that are relevant to geodetic computations:

a. Inverses. For some applications, it is desired to have the full inverse of the matrix L^{o}. Since this inverse is dense, for large geodetic problems there is simply too much data to be represented, let alone computed, by conventional methods. There are, however, properties of this data that allow for multi-level representation and computation. Specifically, the entries of this inverse representing couplings between two stations become smooth functions as the distance between the stations increases, so they can be represented on coarser levels; the greater the distance is, the coarser the representation can be. Thus, only couplings between neighboring stations need be represented on the finest level. Moreover, such a representation for the inverse of L^o can be computed inexpensively by recursively using a similar representation for the inverse of L^1. The Full Approximation Scheme (FAS; cf., [4; Sec. 5] or [3; Sec. 8]) version of multi-level processing should be used for these purposes.

b. Partial inverses. A further very substantial savings in computer time and storage can be obtained if only a relatively few entries of the inverse of L^o are requested. This follows from observing that only coarse level couplings need be computed for regions that are sufficiently far from the regions that are coupled by these requested entries.

c. Geodetic updates. Occasionally, some changes in the data of a geodetic problem may arise from new measurements. Finding the influence of such changes on the computed solution would be very inexpensive if the multi-level structure were preserved since this could be accomplished by one multi-level cycle, in which relaxation on each level would be confined to a neighborhood of the changed data. (The geometric size of this neighborhood is larger for coarser levels, but the number of relaxed stations on each level remains fixed and rather small.)

d. The nonlinear geodetic problem. The above tests were made on a linearized version of the geodetic problem since that was the version supplied. In order to directly treat the nonlinear problem, the only needed programming change is in calculating the residuals f_i^o (see (6.3)): the values of g_k and its partial derivatives should be calculated directly from their nonlinear expressions (6.4)-(6.5), at least once per cycle. Since a good initial approximation is furnished for the geodetic problem, one may then be able to solve the nonlinear problem in basically the same amount of work as in solving one linearized problem. When the initial approximation is not sufficiently good, one can still achieve the same efficiency by incorporating, in conjunction with FAS, transfer of the nonlinear relations to the coarse levels.

e. Storage management. The North American Geodetic Survey includes some 6×10^6 observations and 2.5×10^5 stations. The problem is therefore too large to be stored, let alone solved, in core memory. The multi-level approach offers very efficient storage management possibilities. (See Section 7.g.)

f. Using geometric information. How much more efficiency the use of geometric information (see Section 7.b) can produce in solving geodetic problems is a question deserving further study.

7. Further research (general)

a. Algorithmic variations. Several variations on the basic AMG algorithm may lead to greater efficiency, including approaches for achieving the coarsening objectives of Section 3.2 that are more systematic. Their increase in expense must, of course, be measured against improvement in performance. Other variations concern the objectives, such as the modification of Objective 1 to allow 'twice removed' connection strengths, that is, points that depend strongly on points that in turn depend strongly on C points.

b. Use of geometric information. AMG presently makes no use of geometric information. Many applications can, however, provide simple but important geometric features of the problem, such as the relationship between strong dependence and geometric distance, or the relative geometric positions of the unknowns. The latter would be especially useful for achieving higher order interpolation accuracy.

c. General nonlinear problems. Multigrid solvers can usually be applied directly to a nonlinear problem via FAS, solving it as fast as solving the corresponding linearized problem, without requiring any (global) linearization. There are, however, certain questions to be investigated concerning the application of this scheme in the AMG-type setting, where no grids nor differential origins are given and where generality is the objective: A linear problem is specified by a matrix, but how does one generally specify the nonlinear problem?, How should FAS be combined with variational coarsening?, etc.

d. Sequence of problems. For time dependent problems or continuation processes, for example, conventional multigrid can often take advantage of the nature of the solution changes beyond that of other methods (cf., [3; Sec. 15]). This is true of AMG as well. In addition, the AMG coarsening can remain fixed for many problems in the sequence and should be updated only when slower convergence rates are sensed.

e. Design aid. With careful attention to input/output features, a version of AMG could be developed as a design tool. By providing an analysis of the strengths of dependence, AMG could act as a guide in the development of conventional multigrid applications.

f. Storage-efficient algorithms. For problems too large to be processed in core memory, the multi-level approach offers very efficient algorithms because its finer-level processes are completely local; its global processes are performed only on coarse grids, fitting easily into limited storage (cf., [3; Sec. 8.7]).

		Total Relative	Asymptotic Factor
Problem	ε	Complexity	Per Cycle
1a.	---	3.99	0.23
1b.	---	3.67	0.06
1c.	---	4.04	0.25
1d.	1.0	3.89	0.22
1d.	0.5	3.42	0.15
1d.	0.1	3.72	0.09
1d.	0.01	3.42	0.08
1d.	2.0	3.42	0.14
1d.	10.0	3.69	0.10
1d.	100.0	3.42	0.08
2.	0.5	3.48	0.25
2.	1.0	3.41	0.30
2.	1.5	3.43	0.46
2.	2.0	3.42	0.73
2.	-0.5	3.41	0.17
2.	-1.0	2.59	0.19
2.	-1.5	3.32	0.10
2.	-2.0	1.74	0.00

TABLE 1

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, 'Generalized local maximum principles for finite difference operators,' Math. Comp. 27 (1973), 685-718.

3. 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.

4. A. Brandt, 'Multilevel adaptive solutions to boundary-value problems,' Math. Comp. 31 (1977), 333-390.

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

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

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

8. H. Foerster, K. Stuben and U. Trottenberg, 'Nonstandard multigrid techniques using checkered relaxation and intermediate grids,' Elliptic Problem Solvers, (M. H. Schultz, ed.), Acad. Press, NY, 1981, 285-300.

9. S. F. McCormick and J. Ruge, 'Multigrid methods for variational problems,' SIAM J. Num. Anal., to appear.

10. A. Pope, 'Geodetic computational and sparsity,' Proc. Sparse Matrix Symp., Fairfield Glade, TN, Oct. 1982.

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

ACKNOWLEDGEMENT. The authors are indebted to Dr. Klaus Stuben, for the numerical results reported in Section 5 and to Dr. Alan Pope for providing the test case and much guidance.

APPENDIX

The application of AMG to the problem

$$L^oU^o = f^o, \qquad L^o\colon R^{No} \to R^{No}$$

can be divided into two separate processes: coarsening (determination of the coarse grid points and prescription of the grid transfer operators) and solution by multigrid cycling. An outline of the coarsening algorithm is as follows:

1. Set $\ell \leftarrow 0$ and $G^\ell \leftarrow \{1,2,\ldots,n_o\}$.
2. Split G^ℓ into two sets $\mathcal{C}^\ell$ and $\mathcal{F}^\ell$, where $\mathcal{C}^\ell$ is the set of coarse grid points, and set $G^{\ell+1} \leftarrow \mathcal{C}^\ell$.
3. Determine $I^\ell_{\ell+1}$ and set $I^{\ell+1}_\ell \leftarrow (I^\ell_{\ell+1})^T$.
4. Compute $L^{\ell+1} = I^{\ell+1}_\ell L^\ell I^\ell_{\ell+1}$.
5. If $G^{\ell+1}$ has few enough points to permit quick solution of the problem, set $M \leftarrow \ell+1$ and stop. Otherwise, set $\ell \leftarrow \ell+1$ and go to 2.

Steps 2 and 3 are closely related; a more detailed description of them is the subject of this appendix.

Four parameters are used in this algorithm: $n_1 (\geq 0)$, $n_2 (\geq \max\{n_1,1\})$, $\varepsilon \geq 0$, and $\alpha < 1$. The specific form that objective 1 of Section 3.2 takes here is: for each point i in $\mathcal{F}^\ell$ with at least n_2 neighbors (j is a neighbor if i in $L^\ell_{ij} \neq 0$) between n_1 and n_2 of these neighbors must be in $\mathcal{C}^\ell$, and the total strength of dependence of i on these neighbors must be at least $1 - \varepsilon L^o_{ii}$ times the total strength of dependence of i on its n_2 strongest neighbors. This set of neighbors is used to define interpolation to point i, and is used in choosing the sets $\mathcal{C}^\ell$ and $\mathcal{F}^\ell$. The specific form that objective 2 takes is in terms of α which limits the total number of connections that any point in $\mathcal{F}^\ell$ can have. Thus, if a point i has more than α times the average number of connections, it is forced to be a $\mathcal{C}^\ell$ point.

The coarse grid point selection algorithm itself can be broken down into three parts: determination of the interpolation point sets for each point, assignment of the initial interpolation values for each point, and the actual $\mathcal{C}^{\ell}$ point selection. The interpolation values are assigned to a point on the basis of its inclusion in the interpolation sets of other points, and points with high interpolation values are chosen to be coarse grid points.

In order to explain this algorithm, let the neighbors of point i on a grid be defined by

$$N(i) = \left\{ j:\ j \neq i \quad \text{and} \quad L^{\ell}_{ij} \neq 0 \right\}.$$

The number of elements in a set $S \subset G^{\ell}$ will be denoted by $|S|$. Then the determination of the interpolation set for a point i actually consists of finding a collection of subsets of N(i), denoted by B(i), so that if i is in $\mathcal{F}^{\ell}$, then one of these subsets must be in $\mathcal{C}^{\ell}$. B(i) can be characterized by two disjoint subsets of N(i), denoted by $S_1(i)$ and $S_2(i)$, and a number $p_3(i)$, so that

$$B(i) = \{ S_1(i) \cup S_3(i):\ S_3(i) \subset S_2(i) \text{ and } |S_3(i)| = p_3(i) \}.$$

This can be determined as follows:

1) If $|N(i)| < n_2$, then let $S_1(i) = N(i)$, $S_2(i) = \emptyset$, $p_2(i) = 0$, $p_3(i) = 0$, and $p(i) = p_1(i) = |N(i)|$.

2) If $|N(i)| \geq n_2$, then let

$$s_i = \max\{ \textstyle\sum_{j \in S} |L^{\ell}_{ij}|:\ S \subset N(i) \text{ and } |S| = n_2 \};$$

$$\bar{B}(i) = \{ S \subset N(i);\ S_1 - \textstyle\sum_{j \in S} |L^{\ell}_{ij}| < \varepsilon \cdot |L^{\ell}_{ii}|$$

$$p(i) = \min_{S\varepsilon\overline{B}(i)} |S|;$$

$$B(i) = \{S\varepsilon\overline{B}(i): |S| = p(i)\};$$

$$S_1(i) = \bigcap_{S\varepsilon B(i)} S, \; p_1(i) = |S_1(i)|;$$

$$S_2(i) = \left(\bigcup_{S\varepsilon B(i)} S\right) - S_1(i);$$

$$p_2(i) = |S_2(i)|; \text{ and}$$

$$p_3(i) = p(i) - p_1(i).$$

The second process is the assignment of interpolation values to all points i. This is based on the importance of i for possible use in the interpolation of all other points and is defined as follows:

$$v(i) = \sum_{j:i\varepsilon S_1(j)} \frac{1}{p(j)} + \sum_{j:i\varepsilon S_2(j)} \frac{p_3(j)}{p(j)p_2(j)}$$

In addition, if

$$|N(i)| \geq \alpha > \sum_{j\varepsilon G^{\ell}} |N(j)|/|G^{\ell}|,$$

set $v(i) \leftarrow v_{max}$

where v_{max} is set high enough so that $v_{max} \geq v(i)$ for all i.

The third process is $\mathcal{C}^{\ell}$ selection given by

1. Set $\mathcal{C}^{\ell} \leftarrow \emptyset$ and $\mathcal{F}^{\ell} \leftarrow G^{\ell}$.
2. If v(i)=0 for all i not in $\mathcal{C}^{\ell}$, stop. Otherwise, pick i so that v(i) is a maximum. (Note: In order to avoid searches, points are organized in doubly linked lists, each list containing

points with a small range of values. A pointer is kept pointing to one end of the list to which insertions are made as values are changed. The choice of the point with the highest value of v(i) is actually made by choosing the first element of the nonempty list corresponding to the highest range of values.)

```
3.  For each $j \varepsilon S_1(i)$, set $v(j) \leftarrow v(j) - \frac{1}{p(i)}$
4.  For each $j \varepsilon S_2(i)$, set $v(j) \leftarrow v(j) - \frac{p_3(i)}{p(i) \cdot p_2(i)}$

5.  For each $j$ such that $i \varepsilon S_1(j)$:

    set $S_1(j) \leftarrow S_1(j) - \{i\}$;

    if $p(j) > 1$, then:

        for all $j_2 \varepsilon S_1(j)$, set $v(j_2) \leftarrow v(j_2) + \frac{1}{p(j)-1} - \frac{1}{p(j)}$

        and for all $j_2 \varepsilon S_2(j)$, set $v(j_2) \leftarrow v(j_2) + (\frac{1}{p(j)-1} - \frac{1}{p(j)}) \cdot \frac{p_3(j)}{p_2(j)}$ ;

    set $p(j) \leftarrow p(j) - 1$.

6.  For each $j$ such that $i \varepsilon S_2(j)$:

    if $p_3(j) = 1$, set $S_2(j) = \emptyset$;

        otherwise, set $S_2(j) \leftarrow S_2(j) - \{i\}$;

    for all $j_2 \varepsilon S_1(j)$, set $v(j_2) \leftarrow v(j_2) + \frac{1}{p(j)-1} - \frac{1}{p(j)}$ ;

    for all $j_2 \varepsilon S_2(j)$, set $v(j_2) \leftarrow v(j_2) \frac{p_3(j)-1}{(p(j)-1)(p_2(j)-1)} - \frac{p_3(j)}{p(j) \cdot p_2(j)}$ ;

    set $p(j) \leftarrow p(j)-1$, $p_2(j) \leftarrow p_2(j)-1$, $p_3(j) \leftarrow p_3(j)-1$;

    if $p_3(j) = 0$, set $p_2(j) \leftarrow 0$.
```

7. Set $\mathcal{C}^{\ell} \leftarrow \mathcal{C}^{\ell} \cup \{i\}$, $\mathcal{F}^{\ell} \leftarrow \mathcal{F}^{\ell} - \{i\}$, and go to 2.

$\mathcal{C}^{\ell}$ and $\mathcal{F}^{\ell}$ are now defined. Thus, for every i in $\mathcal{F}^{\ell}$, there is a set S(i) in $\mathcal{C}^{\ell}$ which can be used to interpolate to point i. However, this set need not be unique. Thus, to specify interpolation, pick the subset S(i) of $\mathcal{C}^{\ell}$ in B(i) on which i has the greatest total strength of dependence; the interpolation matrix $I_{\ell+1}^{\ell}$ is defined with entries b_{ij}^{ℓ}, $i \varepsilon G^{\ell}$, $j \varepsilon G^{\ell+1}$, so that for $i \varepsilon \mathcal{C}^{\ell}$,

$$b_{ij}^{\ell} = \delta_{ij}$$

and for $i \varepsilon \mathcal{F}^{\ell}$,

$$b_{ij}^{\ell} = \begin{cases} -L_{ij}^{\ell} \Big/ \sum\limits_{k \notin S(i)} L_{ik}^{\ell} & \text{if } j \varepsilon S(i) \\ \\ 0 & \text{if } j \notin S(i) \end{cases}$$