Electronic Transactions on Numerical Analysis.
Volume 10, 2000, pp. 1-20.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

GENERAL HIGHLY ACCURATE ALGEBRAIC COARSENING*

ACHI BRANDT†

Abstract. General purely algebraic approaches for repeated coarsening of deterministic or statistical field equations are presented, including a universal way to gauge and control the quality of the coarse-level set of variables, and generic procedures for deriving the coarse-level set of equations. They apply to the equations arising from variational as well as non-variational discretizations of general, elliptic as well as non-elliptic, partial differential systems, on structured or unstructured grids. They apply to many types of disordered systems, such as those arising in composite materials, inhomogeneous ground flows, "twisted geometry" discretizations and Dirac equations in disordered gauge fields, and also to non-PDE systems. The coarsening can be inexpensive with low accuracy, as needed for multigrid solvers, or more expensive and highly accurate, as needed for other applications (e.g., once-for-all derivation of macroscopic equations). Extensions to non-local and highly indefinite (wave) operators are briefly discussed. The paper re-examines various aspects of algebraic multigrid (AMG) solvers, suggesting some new approaches for relaxation, for interpolation, and for convergence acceleration by recombining iterants. An application to the highly-disordered Dirac equations is briefly reviewed.

Key words. multiscale algorithms, multigrid, algebraic multigrid, AMG, nonlinear AMG, unstructured grids, coarsening, distributive coarsening, homogenization, compatible relaxation, Dirac equations.

AMS subject classifications. 35A40, 65F10, 65K10, 65M55, 65N22, 65N55, 65Y05, 76M20.

1. Introduction.

Algebraic multigrid (AMG) algorithms are solvers of linear systems of equations which are based on multigrid principles but do not explicitly use the geometry of grids; see [20], [8], [32], [36]. The emphasis in AMG is on automatic procedures for *coarsening* the set of equations, relying exclusively on its algebraic relations. AMG is widely employed for solving discretized partial differential equations (PDEs) on unstructured grids, or even on structured grids when the coarse grid can no longer be structured, or when the PDE has highly disordered coefficients. AMG can also be used (as in [20]) for certain discrete systems not arising from differential equations.

The scope of AMG solvers has been rather limited, though. Its coarsening procedures have been inadequate for general non-scalar, or high-order, or non-elliptic and anisotropic PDE systems, and also for non-variational discretizations. The purpose of the present paper is to delineate general algebraic coarsening techniques that can be employed in all those cases. In fact, these techniques belong to a family of coarsening methods that turns out to be very successful in a much wider range of problems, including highly nonlinear systems, problems with discrete-state (e.g., Ising spin) or otherwise constrained variables, and with non-deterministic relations (as in statistical mechanics), and problems with moving particles (as in molecular dynamics): see some references in §2 below.

Two types of devices are developed here. The first is a general criterion for gauging, and a method to control, the quality of the *set* of coarse-level variables, prior to deriving the coarse-level equations: see §3. The second includes general approaches for deriving the coarse-level *equations* once the coarse variables are given.

Two such approaches are presented. In both of them one can control the level of coarsening accuracy, and the corresponding amount of computational work per coarse equation, by choosing the size of certain stencils: The error in approximating sufficiently "smooth" components (i.e., those slow to converge in relaxation) decreases *exponentially* with the size

* Received May 25, 1999. Accepted for publication February 1, 2000. Recommended by V. Henson. This research was supported by Grant No. 9680 from the Israel Ministry of Science, Grant No. NAG-1-1823 from NASA, and Grant No. 696/97 from the Israel Science Foundation; and by the Carl F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science.

†Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel.

of those stencils (see §6; the usefulness of high-accuracy coarsening is discussed in §9). The work per coarse equation increases proportionately to some *power* of that size (remaining of course *linear* in the overall number of variables).

The first approach for deriving the coarse equations is based on the premise that although in principle each coarse variable depends on *all* others, this dependence always decays exponentially with distance, due to our criterion for choosing the set of coarse variables. (For 2D discrete Poisson equations, for example, the dependence coefficient tends (after enough coarsening levels) exactly to $\exp(-\pi r^2/2)$, where r is the distance measured in meshsizes of the coarse level [33], [42].) Hence a highly accurate coarse equation can be constructed locally. This is done by solving a certain local optimization problem: see §4. We call this approach *direct coarsening*.

The second approach is based on the traditional *Galerkin coarsening*, where the interpolation and restriction operators are again derived by solving local optimization problems: see §10.

The direct approach surely yields stable and easily-relaxable coarse-level equations; with special care in the derivation of interpolation, comparable stability may also be achieved by the Galerkin procedure (see Appendix B).

A detailed comparison of the two approaches has not yet been done. The Galerkin approach seems to enjoy less expensive setup. In the cases where this is not the overriding issue, and in particular in highly repetitive systems (as most natural systems tend to be, provided they are not linearized or carelessly discretized), the direct approach is most widely applicable: it can be extended to nonlinear equations (see §8) and to non-deterministic problems in statistical mechanics and molecular mechanics.

Extensions of the coarsening methods to *dense* matrices (discretizing non-local physical laws) is briefly discussed in §11, as well as a generic way to treat highly-indefinite matrices (oscillatory operators).

Additional important AMG tips are given in §12. They include various types of relaxation schemes to suit a variety of situations, a general rule for the *local* adaptation of the amount of relaxation, various ways of accelerating convergence by recombining iterants, improved interpolation procedures, and "grey box" AMG. The use of a generalized type of coarsening, named "distributive coarsening" is explained in Appendix A; it is particularly handy for AMG solvers of discretized *non-scalar* PDE systems. An application of the various techniques of this article to the solution of the highly-disordered Dirac equations is reviewed in Appendix C.

2. The given equations. The description of coarsening schemes in this article will be in the framework of a linear system of equations

$$Au = b \,, \tag{2.1}$$

where A is an $N \times N$ matrix. It is important to note, however, that the described methods can be generalized to *nonlinear* (see §8) and *non-deterministic* problems. In fact, methods analogous to those described here have already been developed for coarsening nonlinear and non-deterministic systems, first in statistical physics (see [14, §13.2] and [22], [23], [24]), then also in molecular dynamics of macromolecules [14, §14.6] and of fluids [14, §14.7].

Conceptually, the general property implicitly assumed for the system (2.1) for the coarsening described below to be meaningful is *localizability*, i.e., it is implicitly assumed that each unknown u_i could be assigned a location in a low dimensional space, such that each equation in the system involves only neighboring unknowns. The dimension d of the embedding space is not really restricted to any particular value, however the cost of the coarsening scheme will be some CN, where C is bounded independently of N but rises exponentially with d.

We comment below (§11) that the localizability property, as stated here, is neither necessary nor sufficient, and that still more general coarsening schemes can be devised. However, the present description is enough for the coarsening of most discretized partial differential systems on structured or unstructured grids, including even non-elliptic, anisotropic and non-scalar systems with non-variational discretizations, even with disordered fields of coefficients, as well as indeed most *non*-PDE systems, provided they are localizable.

Examples of problems with *disordered coefficients* for which our automatic coarsening scheme can be used include: (1) Dirac equations in critical gauge fields, whose fast multigrid solution (used also for fast calculation and fast updates of main terms in the inverse matrix — see [12, §§8–9], [14, §§11–12] and [31]) is essential for progress in elementary-particle theory; (2) homogenization and fast solvers for composite materials or for flows in highly variable media (e.g., ground flows); (3) problems whose formally structured discretization grid represents highly skewed actual geometry (e.g., see [29], [30]); etc.

Non-PDE localizable problems for which the coarsening scheme can be, or has been, applied include: (1) The geodetic problem (for which the original AMG algorithm was developed [20]); (2) elastic structures made of discrete components (beams, trusses, etc.); (3) clustering problems (e.g., for image segmentation), where the clustering algorithm imitates the AMG coarsening process (see [35]); (4) molecular mechanics (where, however, most useful is the *non*-deterministic version of the coarsening scheme); etc.

In all these cases the coarsening scheme would yield AMG-type solvers costing $O(1)$ operations per unknown, with the option to adjust at will the coarsening accuracy. The choice of accuracy, and the use and cost of highly accurate schemes, are discussed below.

3. The coarse-level variables. The first task in coarsening a given system, such as the system of equations (2.1), is the choice of coarse-level variables. In different systems these variables can have very different character. For example, in Ising-spin systems [41] a coarse spin can be defined by the *majority rule* (i.e., it assumes the sign of the majority in a corresponding block of fine-level spins). In macromolecular systems, the coarse-level "atoms" can stand for the *average location* of several chemically-bonded fine-level atoms [14, §14.6]. In atomistic simulation of fluids, where the finest level is described by the location of its atoms, coarser levels employ grid functions, where each value of a function at a grid point stands for a quantitative property (e.g., the total mass, or the total electrostatic dipole moment) of the set of atoms in a corresponding grid cell [14, §14.7].

For the linear equation system (2.1), we can generally assume that each coarse level variable u_k^c stands for a linear combination of fine-level variables

$$u_k^c = \sum \mu_{ki} u_i \ , \tag{3.1}$$

with fixed weights μ_{ki}, vanishing outside a small local neighborhood. (For a generalization of (3.1), particularly useful for PDE systems with distribution relaxation schemes, see Appendix A.) In classical AMG procedures (see [20], [8], [32], [36]), for example, the coarse variables are identified with a *subset* of the fine variables, i.e., each coarse variable k is identified with one fine-level variable $i = i(k)$, hence

$$\mu_{ki} = \delta_{i,i(k)} \ . \tag{3.2}$$

In all these cases there exists one general and practical *criterion* for determining the quality of the coarse variable *set*, independently and prior to the construction of the coarse level *equations*. The criterion of course must depend on the fine-level relaxation scheme (or Monte-Carlo scheme, in statistical field problems). For example, *full* coarsening schemes

ETNA
Kent State University
etna@mcs.kent.edu

are known to be adequate for some anisotropic problems relaxed by line (or plane) relaxation schemes, whereas *semi* coarsening must be employed for such problems if pointwise relaxation schemes are used [2], [6, §3.3].

To introduce this general criterion, for a given relaxation scheme and a given set of coarse variables, we define the concept of *compatible relaxation*. This is a modified relaxation scheme which keeps the coarse-level variables invariant. For example, for the classical AMG coarsening (3.2) with the classical-AMG Gauss-Seidel relaxation, a compatible relaxation sweep simply means a relaxation sweep that avoids relaxing any fine-level variable i which is identified with a coarse-level one ($i = i(k)$); this has been called *F-relaxation* in AMG literature [20] (see more on that in §12.4). In the case of the more general coarsening (3.1), a compatible relaxation scheme can be formed by relaxing simultaneously two (or more) variables at a time, in a way that keeps $\sum_i \mu_{ki} u_i$ unchanged for all k. Generally, each step of the compatible scheme will consist of several ($O(1)$) simultaneous steps of the original scheme, in a way that preserves the values of the coarse variables. In addition, we require that each compatible-relaxation *sweep* will be *complete*, in the sense that its moves span the entire space spanned by the original relaxation sweep, modulo the coarse variable invariance. (The analogous concept of *compatible Monte Carlo*, for statistical-field problems, was introduced in [22], [23], [24].)

A general measure for the quality of the set of coarse variables is the convergence rate of the compatible relaxation.

A necessary condition for a coarsening scheme to yield an efficient multigrid algorithm (e.g., with convergence per cycle independent of the number N of unknowns) is that the compatible-relaxation convergence factor is bounded away from 1 (independently of N). In most cases of interest this is also a *sufficient* condition, provided of course that the coarse-variable set is provided with sufficiently accurate coarse-grid *equations* (e.g., using the methods described below).

The compatible relaxation can be used as a *practical* tool in the process of selecting the coarse variables. Starting with an initial set C_0 of coarse variables, whose values are held fixed at zero, a few sweeps of compatible relaxation (with vanishing right-hand side so that the solution should converge to zero, but starting of course from a suitable non-zero first approximation) will easily detect convergence slowness if the coarse-variable set is inadequate. Moreover, a set of variables C_1 that should be added to the coarse set is thereby exposed: C_1 must be a subset of the set S_1 of variables which are slow to converge (to zero). As usual in AMG, C_1 should be chosen so that each variable in S_1 "strongly depends" on C_1. In systems where this notion (strong dependence) is not sufficiently well defined, one can assume any non-vanishing dependence to be strong; this may result in a too sparse set C_1, but then this compatible-relaxation selection tool can be applied once more, now with the set $C_0 \cup C_1$ instead of C_0. Very few such iterations will suffice. The initial set C_0 can be empty. In the case of structured grids, C_0 can be chosen as the standard set of full coarsening.

Note that such a coarse-variable selection scheme is suitable for highly parallel processing, assuming of course that the relaxation scheme is suitable for such processing. Furthermore, in parallel processing based on domain decomposition, the compatible relaxation sweeps can be confined to disjoint subdomains, setting to zero each difference (in the relaxed equations) between variables belonging to different subdomains. Such a procedure will tend to select more coarse variables near the subdomain boundaries, but this is exactly desirable from the point of view of creating coarse-grid equations which do not "penetrate" deep into neighboring subdomains.

It should finally be mentioned that instead of increasing the set of course variables one can increase the degree of simultaneity in the relaxation: see §12.1.

4. Derivation of coarse-level equations. Given the system of equations (2.1) (which may be the residual equations of an original system, i.e., b may be the vector of residuals and u the sought correction to a given approximate solution), and the coarse-level variables (3.1), we want to find a system of *coarse equations* approximately satisfied by these *coarse variables*. We will construct one coarse equation for each coarse variable; this will promote stability of the equations and efficiency of the simple Gauss-Seidel relaxation at the coarse level.

The fast convergence of the compatible relaxation implies that the fine-level solution to equation (2.1) at each point can be found *locally*, given just its *coarse neighborhood*, with very weak remnant dependence on values outside that neighborhood: the remnant dependence must decay *exponentially* (or even faster) as a function of the neighborhood radius. Since the coarse values u^c themselves depend, via (3.1), only on the fine values around them, it follows that each coarse value u^c_k depends solely on the other coarse values in a neighborhood around it, except for a remnant, exponentially (or faster) decaying dependence on values outside that neighborhood.

Hence, to derive a coarse equation for a particular coarse unknown, u^c_0 say, we first select a small neighborhood Ω^c_0 of u^c_0. This neighborhood is the set of coarse points k whose unknowns u^c_k will participate in the derived equation; in particular $0 \in \Omega^c_0$. The larger the set Ω^c_0, the more accurate (but also more expensive to derive and operate) is the derived equation (see §5 below). Denoting $\Omega^{c-}_0 = \Omega_0 - \{0\}$, our aim is to find (an approximation to) the value of u^c_0 given the set of values $\{u^c_k\colon\ k \in \Omega^{c-}_0\}$.

We also select a corresponding fine-level neighborhood Ω_0, such that Ω_0 includes all the fine points i needed to define Ω^c_0 (i.e., if $k \in \Omega^c_0$ and $\mu_{ki} \neq 0$ then $i \in \Omega_0$), as well as sufficiently many neighboring points, to ensure (or make it likely) that Ω_0 has no "holes".

Denoting by a_{ij} the terms of the matrix A $(i, j = 1, \ldots, N)$, we define $\Omega^{^I}_0$ to be the *interior* of Ω_0, i.e.,

$$\Omega^{^I}_0 = \{i \mid i \in \Omega_0 \text{ and if } a_{ij} \neq 0 \text{ then } j \in \Omega_0\}\ .$$

In deriving the coarse equations, we first require the following two obvious constraints:

$$\sum_j a_{ij} u_j = b_i \quad \text{for all} \quad i \in \Omega^{^I}_0\ , \tag{4.1}$$

$$\sum_i \mu_{ki} u_i = u^c_k \quad \text{for all} \quad k \in \Omega^{c-}_0\ . \tag{4.2}$$

Under these two constraints u^c_0 still has that remnant dependence on the unconstrained values outside $\Omega^{^I}_0$; allowing those values to be unreasonably large can still affect u^c_0 to any arbitrary extent. We thus should seek to satisfy (4.1) and (4.2) while keeping the relevant values outside $\Omega^{^I}_0$ as small as possible, in some norm. The exact choice of that norm is not of crucial importance, since it affects only that remnant, exponentially small dependence. We have chosen to minimize the following quadratic form

$$Q_0(u) = \sum_{i \in \Omega^{^I}_0,\ j \in \Omega_0 - \Omega^{^I}_0} |a_{ij}|(u_i - u_j)^2\ . \tag{4.3}$$

This form has two advantages, which may be significant when the neighborhood size is small (to save work). First, Q_0 is based on *differences*. As a result, in the case that the local rows of A have zero sums (as would often be the case for discretized PDEs), if $u^c_k = \sum \mu_{ki}$

for all $k \in \Omega_0^{c-}$ then the minimization of (4.3) under the constraints (4.1) and (4.2) will yield $u_i = 1$ for all $i \in \Omega_0$, hence $u_0^c = \sum \mu_{0i}$. This means that the resulting coarse-level equation will be satisfied by a *constant* (or, more precisely, by the coarse values corresponding to a constant fine-level solution). More generally, the coarse equations will tend to be nearly satisfied by *smooth* solutions when this tendency exists at the fine level. Another advantage of the form (4.3) is that it tends to attach a larger weight to boundary values u_j which are likely to affect more the interior solution in general, and u_0^c in particular. (In fact, for non-elliptic problems with strong directional dependence, a more precise, direction-depending weighting may be beneficial: see Appendix B.)

Using Lagrange multipliers, the minimization of (4.3) under the constraints (4.1)–(4.2) yields a linear system of equations (Euler equations) of the form $M_0\widetilde{u} = \widetilde{b}$, where $\widetilde{u}$ is the vector of the local unknowns, including $\{u_i : i \in \Omega_0\}$ and the Lagrange multipliers, and $\widetilde{b}$ includes the data $\{b_i\colon\ i \in \Omega_0^I\}$ and $\{u_k^c : k \in \Omega_0^{c-}\}$. What we actually need to know is not the solution $\widetilde{u} = M_0^{-1}\widetilde{b}$ for any particular data set $\widetilde{b}$; what we need is the *dependence* of u_0^c on $\widetilde{b}$, or more precisely, the coefficients $\{a_{0k}^c\}$ and $\{w_{0i}\}$ in the relation

$$u_0^c = \sum_{k\in\Omega_0^{c-}} a_{0k}^c u_k^c + \sum_{i\in\Omega_0^I} w_{0i} b_i \ . \tag{4.4}$$

These coefficients can of course straightforwardly be calculated from M_0^{-1}. Assuming for example (3.2), or $u_0^c = u_{i(0)}$, the coefficients in (4.4) depend only on *one row* of M_0^{-1} (the row corresponding to the position of $u_{i(0)}$ in $\widetilde{u}$), hence only on the corresponding *column* of the transposed matrix $(M_0^T)^{-1}$. That column can be calculated by just solving the equation

$$M_0^T x = e_0 \ , \tag{4.5}$$

where e_0 is the unit vector corresponding to the position of $u_{i(0)}$ in $\widetilde{u}$.

The derived equation (4.4) *is the desired coarse-level equation;* w_{0k} *are effectively the weights of the fine-to-coarse residual transfer*. The equation can of course be divided through by any number. Dividing it through by $\sum_i w_{0i}$ would make it comparable to the given fine-grid equation; i.e., the fine and the coarse linear operators will give nearly the same value when operating on a smooth function.

For a good coarse-level approximation, exact residual transfer is usually not needed: the weights w_{0i} can be changed very much, provided the total contribution of each residual (b_i) remains the same, i.e., $\sum_0 w_{0i}$ should not change. (This is not true at special parts of the domains, e.g., near boundaries or singularities; but the effect of such parts on the overall convergence rate can be made small by using adaptive relaxation sweeps (see §12.2). Also, the freedom to change w_{0i} does not hold of course for the residual transfers used in Galerkin coarsening (cf. §10).)

With minor obvious changes, the above coarsening scheme can be used also for the *vectorial* case, i.e., when each u_i stands for a vector of several variables, and each a_{ij} is correspondingly a matrix. This yields one way to coarsen discretized *non-scalar PDE systems*. Another way is of course to use the scheme above in its non-vectorial form, disregarding the grouping of the variables into vectors. For many non-scalar PDE systems, however, the best coarsening is *distributive* (see Appendix A).

4.1. Yavneh scheme. Various other local optimization schemes for deriving the coarse equations can be designed. The following one, proposed by Irad Yavneh, is particularly attractive.

In this scheme one optimizes in terms of the fine-to-coarse transfer operator, i.e., the weights $\{w_{0i}\}$. They are chosen so that recombining the fine-level equations (4.1) with these weights will yield as closely as possible an equation involving only the desired stencil. That is, having chosen (as before) the coarse stencil Ω_0^c on which the equation for u_0^c should be constructed, and defining the coefficients resulting from a recombination of (4.1) by

$$a_{0j}^w = \sum_{i \in \Omega_0^I} w_{0i} a_{ij} \ , \tag{4.6}$$

one wants to minimize those coefficients a_{0j}^w that do not participate in Ω_0^c. In particular, in the common case of coarsening of the type (3.2), one wants to choose $\{w_{0i}\}_{i \in \Omega_0^I}$ so as to

$$\text{minimize} \sum_{j \notin \overline{\Omega}_0^c} g_j |a_{0j}^w|^2 \tag{4.7}$$

under a linear normalization

$$\sum_{i \in \Omega_0^I} c_i w_{0i} = 1 \tag{4.8}$$

and under the zero-sum constraint

$$\sum_{j \notin \overline{\Omega}_0^c} a_{0j}^w = 0 \ , \tag{4.9}$$

where $\overline{\Omega}_0^c = \{ i = i(k) \mid k \in \Omega_0^c \}$, and $g_j > 0$ are weights which decrease (e.g., exponentially) with the "distance" of j from 0. (This distance can be loosely defined algebraically, e.g., as the inverse of the sum of the inverses of algebraic couplings along a chain of points leading from 0 to j.) The exact values of these weights $\{g_j\}$ are not really important (see example in §6). The zero-sum constraint ensures correct treatment of constant solutions; other similar constraints may be added.

Once the minimizing weights $\{w_{0i}\}$ have been calculated, the corresponding values of $\{a_{0j}^w\}_{j \in \overline{\Omega}_0^c}$ form the coefficients of the desired coarse-level equation at the point 0.

In the more general case (3.1), the objective should be to minimize the component of the vector $\{a_{0j}^w\}$ which is orthogonal to that subspace of the space $\{u_j\}$ which is spanned by the vectors $\{\sum \mu_{ki} u_i\}_k$. This means to choose $\{w_{0i}\}_{i \in \Omega_0^I}$ and $\{w_{0k}^c\}_{k \in \Omega_0^c}$ so as to

$$\text{minimize} \sum_j g_j \left(a_{0j}^w - \sum_k w_{0k}^c \mu_{kj} \right)^2 \tag{4.10}$$

under the normalization as above and the zero-sum constraint

$$\sum_j \left(a_{0j}^w - \sum_k w_{0k}^c \mu_{kj} \right) = 0 \ . \tag{4.11}$$

This can be simplified to an optimization only in terms of $\{w_{0i}\}$ in the case of *disjoint* coarsening, i.e., if for each i there exists at most one k for which $\mu_{ki} \neq 0$.

The Yavneh scheme is attractive since it is simpler than the previous one and requires to solve a smaller system of equations. Like the former scheme, it can be generalized to the vectorial case and also to distributive coarsening (see Appendix A). Moreover, in this scheme one can choose to further improve the coarse equation on a given stencil Ω_0^c by increasing the set Ω_0^I (see example in §6).

5. Cost and cost reduction. Taking for example the simple case (3.2), and assuming that for each $k \in \Omega_0^{c-}$ the Lagrange multiplier associated with (4.2) has already been eliminated from the equations along with the Euler equation associated with $u_{i(k)}$, the dimension of M_0 in (4.5) is $\nu_0 \times \nu_0$, where $\nu_0 = n_0 + n_0^I$ while n_0 and n_0^I are the numbers of points in Ω_0 and Ω_0^I, respectively. In the Yavneh scheme the corresponding dimension is only $n_0^I \times n_0^I$.

Solving a system of ν_0 or even just n_0^I equations per each coarse equation may still be very expensive. But this cost can be drastically reduced by the following devices.

(i) *Lowering the number of equations* can obviously be achieved by lowering the coarsening accuracy (see §6). The accuracy of (4.4) depends mainly on the size of Ω_0^c, so for a given accuracy one should choose the smallest Ω_0 containing the variables that define Ω_0^c. A possible practice, in particular, is to include many of the latter in the boundary $(\Omega_0 - \Omega_0^I)$.

(ii) *Elimination ordering*: Many of the unknowns in (4.5) can readily be eliminated. Clever elimination ordering can quickly reduce the number of unknowns by a substantial factor.

(iii) *Low coarsening ratio*: Instead of reducing the number of unknowns by a large factor (e.g., 4) in one coarsening step, this can be done in two steps. For high-accuracy coarsening (aiming at coarse grid equations each of which couples a certain substantial number of unknowns), lowering the coarsening ratio will proportionately lower ν_0 and n_0^I and hence more than pay for the increased number of coarsening steps.

(iv) *Total reduction step*: The first one or two coarsening steps, especially in the case of low coarsening ratio, can sometimes be executed fast and with full accuracy by a straightforward elimination of part (typically around one half) of the variables, as in the "total reduction" algorithm [33]. This results in larger stencils at the coarse level, so the complexity of the system (measured in terms of the number of non-zero terms in the matrix) is not really reduced. Such steps can therefore be repeated only until stencils of some target size (depending on the desired accuracy: see §6) are reached.

(v) *Exploiting repetitions*: Saving work is of course especially crucial at the finest levels. On those levels one often has (or can create, by more careful discretization) a highly repetitive system, where the same set of equations (4.5) can be formed at many different points 0. The equations (4.5) need be solved only once for all these points. More generally, the equation may depend on a parameter which does *not* repeat itself (as in nonlinear or linearized equations; cf. §8). A possible approach then is to solve (4.5) for several representative values of this parameter and keep the solution in a table, from which the solution to other values of the parameter can be *interpolated*.

6. Exponential accuracy. While the work of calculating a coarse grid equation rises *proportionately* to some power of n_0, the accuracy of that equation increases (i.e., its error for smooth components decreases) *exponentially* in some power of n_0.

Indeed, as explained above, from the fast convergence of the compatible relaxation it follows that once all other coarse values are given, u_0^c is uniquely determined, and its dependence on any other coarse value u_k^c falls off at least exponentially with the "distance" between them. (This distance can be defined *algebraically*, as mentioned above.)

We have checked this exponential accuracy by explicitly computing a number of cases of PDEs discretized (and coarsened) on uniform grids, so that the accuracy of the produced coarse equations can easily be judged by comparisons with the PDEs, via Taylor expansions. Of particular interest are *non-elliptic* equations with characteristic directions that are not aligned with the grid, for which the derivation of coarse equations is problematic even on such well structured grids. Namely, in order to yield good (i.e., deeply penetrating) approximations to "smooth characteristic components" (solution components that are smooth in all directions, but are even smoother in the characteristic directions), the coarse equation must not only approximate the same differential equation, but also have the same cross-characteristic numerical viscosity (or, more generally, the same cross-characteristic first differential approximation (FDA); see [5], [27], [18], [17]). Thus, we have particularly examined what coarse-grid equations are produced by our scheme on uniform grids in the case of non-aligned non-elliptic equations.

As an example, consider a fine-level uniform grid on which the difference equation at each point is given by the five-point stencil

$$\frac{1}{h}\begin{bmatrix} & 1 & \\ 3 & -7 & 1 \\ & 2 & \end{bmatrix} . \tag{6.1}$$

Taylor expansions show that this is an approximation to the differential operator $-2\partial/\partial x - \partial/\partial y$. Keeping more terms in the Taylor expansions show that the first differential approximation (FDA) to (6.1) is

$$-2\frac{\partial}{\partial x} - \frac{\partial}{\partial y} + h\left(2\frac{\partial^2}{\partial x^2} + 1.5\frac{\partial^2}{\partial y^2}\right) . \tag{6.2}$$

Introducing the characteristic coordinates $\eta = (2x + y)/\sqrt{5}$ and $\xi = (x - 2y)/\sqrt{5}$, this operator can be written as

$$-2.236067978\frac{\partial}{\partial \eta} + h\left(1.6\frac{\partial^2}{\partial \xi^2} + .4\frac{\partial^2}{\partial \xi \partial \eta} + 1.9\frac{\partial^2}{\partial \eta^2}\right) . \tag{6.3}$$

The cross-characteristic numerical viscosity is the coefficient $(1.6h)$ of $\partial^2/(\partial\xi)^2$.

Using standard coarsening (i.e., the coarse grid lines are every other line of the fine grid, both horizontally and vertically, and the coarse grid values are thus a subset of the fine grid values), we examine several possible choices of Ω_0. One of them is depicted in Fig. 1. The weights a^c_{0k} of the coarse grid equation produced by the scheme of §4 are shown (to a limited precision) in Table 1(A). To appreciate the accuracy of the coarse grid operators, we apply to them Taylor expansions (dividing through by $\sum_k w_{0k}$, to make them comparable to the find-grid operator), obtaining, in terms of the characteristic coordinates, the FDA:

TABLE 1

The coarse-level stencil produced from the fine-level stencil (9): (A) *for the coarsening neighborhood shown in* Fig. 1; (B) *by total reduction.*

$$\text{(A)} \quad \begin{bmatrix} -.0247 & .1613 & .0078 \\ .3604 & -1 & .2321 \\ .0962 & .1950 & -.0281 \end{bmatrix}$$

$$\text{(B)} \quad \begin{bmatrix} & & -.0007 & & \\ & .0124 & .0703 & .0014 & \\ -.0558 & .5830 & -1 & .0648 & -.0007 \\ & .0496 & .2812 & .0055 & \\ & & -.0110 & & \end{bmatrix}$$

$$-1.911\frac{\partial}{\partial \eta} + .059\frac{\partial}{\partial \xi} + h\left(2.518\frac{\partial^2}{\partial \xi^2} + .006\frac{\partial^2}{\partial \xi \partial \eta} + .519\frac{\partial^2}{\partial \eta^2}\right) . \tag{6.4}$$

ETNA
Kent State University
etna@mcs.kent.edu

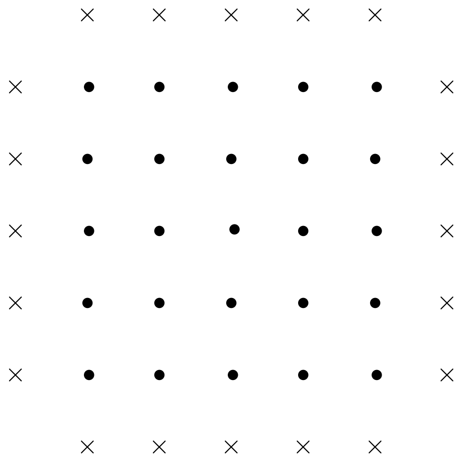

FIG. 1. *A coarsening neighborhood is shown. The points of* Ω_0^I *are shown by* $\bullet$*, those of* $\Omega_0 - \Omega_0^I$ *by* $\times$*, those of* Ω_0^{c-} *by a square, and the point* 0 *is circled.*

Comparing this FDA to that of the fine level we see a 15% error in approximating the differential operator, and a much larger 50% error in approximating the cross-characteristic numerical viscosity of the fine-grid operator (the $\partial_{\xi\xi}$ coefficient).

Trying next a 5×5 set Ω_0^c, one would obtain, by any of the above schemes, the coarse stencil shown in Table 1(B), which includes in fact only 13 non-zero terms. The FDA of this stencil is

$$
\begin{aligned}
&-2.2360679775\frac{\partial}{\partial\eta} + .0000000000\frac{\partial}{\partial\xi} \\
&+h\left(1.6000000000\frac{\partial^2}{\partial\xi^2} + .2333333333\frac{\partial^2}{\partial\xi\partial\eta} + 2.1321428571\frac{\partial^2}{\partial\eta^2}\right)
\end{aligned}
\tag{6.5}
$$

yielding a perfect agreement with the differential operator and with the cross-characteristic numerical viscosity. The reason behind this perfect accuracy is that *any* five-point operator with constant coefficients has such a 13-point coarse operator *exactly* reproduced by it, as realized by *total reduction* [33]. The interesting question however is what accuracy can be obtained at the coarse level without producing a stencil larger than that given at the fine level.

Results for such coarsenings are presented in Table 2, produced by Yavneh with his scheme. The first group of results is for 3×3 stencils; i.e., the fine-level stencil is the one given by (6.1), and the coarse stencil is calculated from it on a 3×3 set Ω_0^c. The second group of results is for 5×5 stencils, the fine one being given in Table 1(B), the coarse one is based on a 5×5 set Ω_0^c. In each group, results are shown for varying sizes of Ω_0^I (which is also the size of the set of equations one has to solve). In each case the results show the important FDA coefficients of the produced coarse operator, and also the value of $\|e\|^2 = \sum_{j \notin \overline{\Omega}_j^c} |a_{0j}^w|^2$, measuring how close is the recombination of the fine operator to the resulting coarse operator. In all these examples we chose g_j in (4.7) to be identically 1 in Ω_0^I, and 0.001 in $\Omega_0 - \Omega_0^I$, and in (4.8) $c_i = \delta_{i,i(0)}$.

The results demonstrate the exponential decrease of the error, both as a function of the stencils' size and as a function of the size of Ω_0^I. (However, directionality is not yet properly treated here; see Appendix B.)

TABLE 2

Stencils	Ω_0^I	∂_ξ	∂_η	$h\partial_{\xi\xi}$	$\|e\|^2$
3×3	5×5	-1.7650923184	-0.0075441496	1.9053747271	0.0079751054
	7×7	-2.1484328781	-0.0131764596	1.7306793477	0.0006201392
	9×9	-2.2166402537	0.0327414291	1.6300309414	0.0000931895
	11×11	-2.2160903796	0.0448098772	1.6016329307	0.0000056882
	13×13	-2.2232626388	0.0193871620	1.6113853806	0.0000017780
	15×15	-2.2325136498	0.0019497360	1.6091775322	0.0000000821
	17×17	-2.2358768405	-0.0006948566	1.6036607007	0.0000000977
	19×19	-2.2366847559	-0.0002560197	1.5995111307	0.0000000077
	21×21	-2.2365770265	-0.0000074721	1.5979139388	0.0000000073
	23×23	-2.2363868082	0.0000334611	1.5977685811	0.0000000013
	25×25	-2.2362696702	0.0000305181	1.5980485897	0.0000000008
	27×27	-2.2362054242	0.0000243788	1.5983834645	0.0000000002
	29×29	-2.2361686425	0.0000200605	1.5986779134	0.0000000001
5×5	7×7	-2.3559819123	-0.0762739902	1.5537216699	0.0000389217
	9×9	-2.2380092343	-0.0013657252	1.5994001829	0.0000000737
	11×11	-2.2360974287	-0.0000214143	1.5999799739	0.0000000002
	13×13	-2.2360704140	-0.0000007703	1.6000022831	0.0000000000
Fine level		-2.2360679775	0.0000000000	1.6000000000	

7. Gauging and controlling accuracy. Unlike the above example, in a general algebraic setting the accuracy of the coarse grid equation is not directly known. There is, however, a general way to estimate it (and increase it, if needed), by observing the values of the produced weights a_{0k}. These weights should generally decrease exponentially with the distance from 0 to k, and one can roughly estimate the size of the largest "missing weights", i.e., the weights that would be added upon increasing the coarsening neighborhood Ω_0. This size is a rough estimate for the accuracy. Another general, more straightforward estimate is provided by the above error norm $\|e\|$ (or its extension to the more general case (3.1)).

In the case of discretized PDEs, another way of estimating the accuracy of the coarse operator is of course to compare its action on smooth components to the action of the fine-level operator.

If the estimated accuracy is not sufficient (e.g., for the purposes discussed in §9), one can of course proceed to boost it. This can be done either by increasing the neighborhood Ω_0 or by adding more points to the coarse level. The former is preferable if and only if the convergence of the compatible relaxation is already fast (exhibiting a convergence factor below 0.7, say).

8. Nonlinear equations. A general way of coarsening (or indeed solving) nonlinear equations is by (iterative) linearization, which is often done already at the stage of *discretizing* the nonlinear PDE (as common in finite-element schemes). In many cases this would be the most effective approach; but not always.

Most nonlinear PDE systems, e.g., in fluid dynamics, are *autonomous*. That is, the same nonlinear system governs at all points, and (away from boundaries) it is independent of any external information. This property makes it possible to create highly *repetitive* nonlinear discretization, having, e.g., the same nonlinear equation (or the same vector of equations) at each grid point (or at least at each grid point of a given "kind", with only a small number of kinds). The coarsening in such a case can be very inexpensive, since the coarse grid equation can be derived once for all points (or once per each *kind* of coarse grid points, with relatively few such kinds). However, the autonomy of the equations, or the repetitive nature of their discretization, is of course lost upon linearization. (It is also lost if the grid is arbitrarily unstructured. To retain a high degree of repetitiveness, semi-structured grids are advisable, like those described in [2], [6] and [1].) Thus, direct coarsening of the nonlinear equations is called for, so as to produce *autonomous coarse equations*.

Moreover, nonlinear coarsening permits the use of nonlinear versions of the multigrid solver, such as FAS [2], thereby avoiding linearization iterations and making it possible to see and manipulate the full solution on coarse levels [6]. More specifically and more important, this may allow once-for-all coarsening, i.e., derivation of macroscopic equations, or homogenization.

The least expensive nonlinear algebraic coarsening is the *term-by-term coarsening*. It requires the user of this algebraic system to supply more information about his problem (cf. §12.5): He should specify the physical species to which each discrete variable and each discrete equation belongs, and describe each discrete equation as a sum of functions (e.g., products) of terms, each term being a discrete linear operator applied to one physical species. To each of these operators one can then apply the coarsening scheme of §4 above, normalizing the resulting linear coarse-level operator by dividing it through by the corresponding value of $\sum_k w_{0k}$. This will constitute one term of the coarse equation. The overall coarse equation is the sum of functions (e.g., products) of these terms — sum and functions corresponding to those that form the fine level equation.

This term-by-term coarsening is suitable for nonlinear *elliptic* systems. For a *non-elliptic* system, however, as mentioned before, it is important to approximate not only the differential operator, but also some FDA terms (such as the cross-characteristic numerical viscosity), hence each coarsened term should include the full non-elliptic *factor of the principal determinant* (cf. [15]).

For example, in 2D fluid dynamics, a usual factor of the principal determinant is the advection operator $u\partial_x + v\partial_y$, whose coefficients (u, v) are part of the unknown functions (hence the nonlinearity). If one coarsens separately each of the six involved linear terms $(u, v, w, \partial_x, \partial_y, \partial_z)$, the combined coarse operator will have a numerical viscosity which is very different from the fine-level one. To change this, the full advection operator should be coarsened as one term. To avoid linearization, one can coarsen it m times, say, for m different values of u/v (e.g., for m equidistant values of the angle $\operatorname{arc\,tg}(u/v)$), yielding a table of m coarse operators. For the actual value of u_0/v_0 found at a coarse point one then uses an operator resulting from interpolating to u_0/v_0 from that table (cf. Item (v) in §5).

9. Usefulness of higher accuracy. For the purpose of multi-level (multigrid) *cycles*, a moderate coarsening accuracy would usually suffice. For example, a coarse grid equation with at most 10% error for all "smooth" components (i.e., those slow to converge in relaxation) can yield a multilevel cycle with a convergence factor close to 0.1. By performing successively any number of such cycles, any desired accuracy can rapidly be obtained. This will usually be more cost effective than deriving higher accuracy coarsening.

On unstructured grids and for other disordered algebraic systems (see lists of examples in §2), and even on structured grids in the case of non-elliptic systems, even that moderate accuracy may be difficult to obtain by standard approaches, so the use of the above new coarsening techniques (or those in §10) can be very useful.

Moreover, in many other cases, the still higher degrees of coarsening accuracy obtainable through those techniques (by investing more work) are really needed.

Examples:

(i) Problems with a large number of AZMs (see §12.3).

(ii) Once-for-all coarsening, for the purpose, e.g., of deriving the macroscopic equations of the given system.

(iii) Cases in which one needs to solve many problems like (2.1), where large parts of b or A do not change from problem to problem, so re-computing at fine levels of those parts can be avoided by having accurately coarsened them before. One important such case is the calculation of many desired terms (e.g., the main diagonal) of A^{-1}. Moreover, in important cases (e.g., quantum field calculations), those terms of A^{-1} need be repeatedly updated upon each change in A occuring during certain Monte-Carlo simulations (see [12, §§8–9]).

(iv) A computing environment which makes it preferable to use as few multigrid cycles as possible, such as massive parallel processing with poor inter-processor communications, or a computer with a very high-speed cache memory.

10. Galerkin coarsening. An efficient (and popular) way to obtain a coarse-level system $A^c u^c = b^c$ is through the Galerkin form

$$A^c = RAP,\ b^c = Rb, \tag{10.1}$$

where P is a coarse-to-fine interpolation operator (prolongation) and R is a fine-to-coarse restriction (or residual-transfer) operator. For simple cases (e.g., discretized *scalar* PDE) and to a certain accuracy, P and R can be derived by the traditional AMG method (described in §12.4). More generally, P and R can be derived by a procedure analogous to that of §4. The main advantage (over the direct derivation of A^c as in §4) is that for a given accuracy these operators require many less coarse points (i.e., much smaller coarsening neighborhood Ω_0^c) than the full coarse operator A^c, hence a much smaller set of equations need to be solved at each step. Also, the set of equations can be simplified, as pointed out below. On the other hand, the stability of the coarse-level equation produced by the Galerkin procedure is questionable: see Appendix B.

To find the interpolation P to a *fine* variable u_0, we select a small neighborhood Ω_0^c of coarse points *from* which the interpolation to u_0 should be made, and also a fine-level neighborhood Ω_0 "around" Ω_0^c (in the same way as in §4), its interior being again denoted by $\Omega_0^{'}$. Given values u_k^c at the coarse points, i.e., given

$$\sum_i \mu_{ki} u_i = u_k^c \quad \text{for all} \quad k \in \Omega_0^c, \tag{10.2}$$

(note the difference from (4.2), where Ω_0^{c-} appeared instead of Ω_0^c), we again minimize (4.3), now under the constraints (4.1) and (10.2). (An important modification of (4.3) for our purpose here is discussed in Appendix B.) This yields a set of equations which determines, among others, the value of u_0 and its linear dependence on the data, written (upon omitting the dependence on $\{b_i\}$) as

$$u_0 = \sum_{k \in \Omega_0^c} p_{0k} u_k^c\ , \tag{10.3}$$

which is the desired interpolation. As in §4 above, the coefficients p_{0k} can be obtained by solving just *once* a transposed system of equations.

A variant of *Yavneh scheme* (§4.1) can also be adopted for finding P. The restriction R can be derived by the same procedures, applied to A^T instead of A.

A simplified (though less accurate) procedure can be applied in the special (but very common) case (3.2), for which (10.2) is reduced to

$$u_{i(k)} = u_k^c\ ,\ k \in \Omega_0^c. \tag{10.4}$$

We can then choose to replace the constraints (4.1) and the minimization of (4.3) by the subset of constraints

$$\sum_j a_{ij} u_j^i = b_i \ , \ i \in \Omega_0^{^I} - \Omega_0^c \cap \Omega_0^{^I} \ , \tag{10.5}$$

where $u_j^i = u_j$ if $j \in \Omega_0^{^I} \cup \Omega_0^c$ and $u_j^i = u_i$ otherwise. The set of equations (10.4)–(10.5) now determines, among others, the value of u_0 and its linear dependence on data (10.3), and the interpolation coefficients p_{0k} can again be found by solving just once a transposed set.

In the *variational* case (where A is symmetric and definite, or nearly definite) this simplified procedure coincides with the *minimization interpolation*, introduced independently by several authors: see [38], [39], [13, §9], [12, §10] and [12, §11.4]; actually, John Ruge has used such interpolation schemes in his AMG codes several years earlier.

The simplified procedure requires the solution of a set of $\overset{I}{n}_0$ equations while the full procedure requires solving $n_0 + \overset{I}{n}_0$ equations (cf. §5). The latter is however expected to be more accurate, since the former does not use the given equations ($\sum_j a_{ij} u_j = b_i$) at the coarse interior points ($j \in \Omega_0^c \cap \Omega_0^{^I}$). Detailed comparisons of the two procedures, and of the entire Galerkin approach vs. that of §4, have not yet been conducted.

11. Extension to non-local and oscillatory operators. Most (perhaps all) non-local operators in physical problems have certain smoothness properties in terms of the coordinates $x = (x^1, x^2, \ldots, x^d)$ of the underlying d-dimensional physical space (or space + times; thus usually $1 \le d \le 4$). If carefully discretized (see, e.g., [26] and [25]), the discrete equations can inherit these properties. Thus, if $G(x, y)$ is the weight of a discrete unknown located at $y = (y^1, y^2, \ldots, y^d)$ in the discrete equation located at x, the *kernel* $G(x, y)$, as function of the continuous variables x and y, will normally be either "smooth" or "asymptotically smooth" or suitably "asymptotically smoothly oscillatory" (for the exact meaning of these concepts, see [10]).

A smooth-kernel part of the fine-level equations can directly (and without linearizations) be represented on the coarse level by "anterpolation" (see §3 of [10]). An asymptotically smooth kernel can be written as a sum of a smooth kernel and a local kernel (see §4 of [10], and also [26], [25]); its smooth-kernel part is coarsened by anterpolation, while its local part (together with other local terms) is coarsened by the methods of Secs. 4 or 10.

Those methods are inadequate, however, for oscillatory kernels or even for local equations with oscillatory solutions, such as the highly indefinite Helmholtz equation $\Delta u + k^2 u = f$, and other wave equations. For such problems, approaches should be developed similar to those in [19]. This is far from trivial in the case of unstructured grids or disordered equations.

In [19] the coarse level is described as $\sum_m A_m(x, y) \exp(ik_1^m x + ik_2^m y)$, where $A_m(x, y)$ are smooth functions (thus representable on the coarse level) and $\left\{(k_1^m, k_2^m)\right\}_m$ are points on the "principal circle" $k_1^2 + k_2^2 = k^2$, so each of these exponential functions solves the *homogeneous* Helmholtz equation. The coarser the level, the larger the number of terms that should be used in the sum (i.e., the better the needed resolution of the principal circle by the points $\left\{(k_1^m, k_2^m)\right\}_m$ — this since on a coarser grid each $A_m(x, y)$ is a smoother function, hence it can represent only a smaller neighborhood on the principal circle).

In the purely algebraic or disordered case, the exponential functions should be replaced by numerical functions obtained by relaxation of the homogeneous equation, separated from each other by local orthogonalizations. Each term in the sum should then be similarly separated into more terms on the next coarser level. And so on.

12. Additional important AMG tips.

12.1. Relaxation schemes. Classical AMG uses always the Gauss-Seidel (GS) relaxation scheme. For general systems, such as those arising in discretizing non-scalar PDE systems, GS is not always adequate. A relaxation scheme that can always be used is the Kacmarz scheme [37]. In cases that GS *is* applicable, it is usually much more effective than Kacmarz. (The work of applying a Kacmarz sweep is roughly twice that of a GS, and for the smoothing obtained by one GS sweep three Kacmarz sweeps are in some cases needed.)

In many cases where GS *cannot* be applied, one can still employ a *Box GS* (BGS) scheme, with effectiveness substantially above that of Kacmarz. The BGS is defined by small local boxes, described geometrically or algebraically. Each step of BGS consists of satisfying all the equations in a box by changing all the variables in that box. Neighboring boxes can overlap. If the boxes are large, their overlap should be comparable to their size.

For discretized PDE systems, *distributive* relaxation schemes can often be designed at the differential level (see Appendix A).

A useful tool in multigrid is *block relaxation*, i.e., relaxing simultaneously several discrete equations. This may serve as an alternative for increasing the number of coarse variables, and may indeed be guided by the same device: compatible relaxation. When the latter detects slowness, small strongly-coupled subsets of S_1 (see §3) can each be marked for simultaneous relaxation of its associated equations. That will change the relaxation scheme and will increase the convergence rate of the correspondingly changed *compatible* relaxation. This approach may be competitive for well-defined small subsets of S_1, but will be less efficient than the alternative (adding variables to the coarse level) when the size of the subset is required to increase (upon detecting slowness even with the *changed* compatible relaxation).

12.2. Adaptive relaxation. A general and important rule for multigrid efficiency, not always adequately respected, is the following.

Adaptive Relaxation Rule: Keep adding relaxation passes at any local neighborhood where the *normalized* residuals are much larger in magnitude than their larger-scale average magnitude. (The normalized residuals are the residuals of the normalized discrete equations. The system (2.1) is normalized when the ℓ_2 norm of each column and each row of A equals, at least approximately, to 1.)

Relaxation adaptation (whose importance for multigrid was already emphasized in Appendix A.9 of [2]) is useful in eliminating multigrid slowness caused by various local singularities, such as boundaries (especially with re-entrant corners; cf. [1]), strong shocks, source singularities, etc. Due to the limited regions where residuals are affected by such singularities, one can afford adding there *many* relaxation passes (if needed). The theoretical usefulness of these extra passes is shown in [11], and discussed in more details in [9].

The adaptive relaxation concept is fully suitable for a purely algebraic setting. A good general practice is to check whether it is needed, by occasionally comparing the global maximum and the global average of the magnitude of the normalized residuals. If the maximum is much (e.g., 10 times) larger than the average, call adaptive relaxation.

12.3. Recombination of iterants. The lower the coarse-grid equation accuracy, the larger the number of solution components that are so ill approximated that they diverge or converge too slowly in the multi-level solver. this is particularly true for almost-zero modes (AZMs), i.e., eigenvectors of A with unusually close to zero eigenvalues. Such modes often reflect some ill defined global moves, such as rigid-body motions of the entire system in problems of elasticity, or a gliding motion of two rigid bodies along their contact surface. Such AZMs also plague various disordered problems, such as Dirac equations on critical gauge fields.

For problems with *many* AZMs, a general cure, as mentioned above, is to increase the coarsening accuracy. A small number m of AZMs (such as those associated with global rigid body motions) may, however, persist in their ill convergence even at higher accuracies. A general way then to eliminate them is by recombining $m+1$ iterants (each being, for example, the approximate solution obtained after another multi-level cycle) so as to minimize the ℓ_2 residual norm; see, e.g., [21].

An efficient (and popular) way to organize such iterant recombinations is to regard the multigrid cycle as a preconditioner in a conjugate-gradient or GMRES procedure. However, note that in addition such recombinations may be very useful at *coarse-level sub*-cycles. Such coarse-level recombinations should converge those AZMs that are well approximated on the first coarse level, but not on still coarser levels. Also, using the FAS scheme, coarse-level recombinations can replace fine-level recombinations, saving dramatically on the amount of storage needed to store previous iterants. See examples in [40].

For special types of problems the number of ill-converging AZMs will remain large even at higher coarsening accuracy. In particular this will occur for discretized PDEs which have many AZMs already in the *differential* systems, such as wave equations with wavelengths small compared with the domain size, or Dirac equations in a quenched hot Gauge field with correlation lengths small compared with, again, the domain size (cf. Appendix C). In such cases it is not enough to recombine iterants *globally* (i.e., with each iterant being multiplied by a *constant*); it is necessary to recombine also on other, *intermediate* scales of variation (i.e., with each iterant being multiplied by a *function* defined on an intermediate coarse level). Ultimately, such procedures will meet the one described at the end of §11.

12.4. Interpolation. A general efficient way to interpolate is to use *compatible relaxation* (see §3; the use of F-relaxation in interpolation was studied in [28] and [36]). One can initialize it by any other interpolation scheme, if one exists, but even with a "trivial" start, the work of this interpolation would be modest, due to the fast convergence of the compatible relaxation. A good trivial start can be obtained just by introducing as many zero differences between (strongly) coupled fine grid points as permitted by the compatibility constraints (3.1), in a simple greedy manner. (In the case of FAS multigrid scheme, where the coarse level stands not for corrections but for the full solution, instead of "zero differences" read "previous differences", i.e., the differences existing on the fine grid prior to the coarse grid correction.)

One can of course prepare the interpolation weights once for all in advance, especially in the cases where this is needed anyway for deriving the coarse grid equations (as in §10). In many cases, instead of the accurate interpolation described in §10, one can of course use the traditional AMG *interpolation by iterated projection of coefficients* [20], [32], [36], defined as follows. Assuming the case (3.2) above, the interpolation has the general form

$$u_i = \sum_k w_{ik} u_k^c \ , \ \sum w_{ik} = 1 \ , \ w_{i(\ell),k} = \delta_{\ell k} \ , \qquad (12.1)$$

and it is designed to be the approximate solution to the homogeneous system $Au = 0$, whose i-th equation can be written in the form

$$u_i = \sum_j \widehat{a}_{ij} u_j \ \text{where} \ \ \widehat{a}_{ij} = -a_{ij}/a_{ii}. \qquad (12.2)$$

(For indefinite systems, using the i-th equation for writing u_i may be inappropriate; instead one can interpolate in terms of the *ghost variables* used in the relaxation process, hence employ in (12.2) $\{\widetilde{a}_{ij}\}$ instead of $\{a_{ij}\}$ — see Appendix A). In view of (12.2), given *any* interpolation $\{w_{ik}\}$ an *improved interpolation* $\{w'_{ik}\}$ can be defined by

$$w'_{ik} = \sum_j \widehat{a}_{ij} w_{jk} \ , \ (i \in F) \qquad (12.3)$$

where F is the set of fine-level variables which are *not* identified with a coarse-level variable ($i \neq i(\ell)$ for all ℓ). Starting with the specific interpolation

$$w_{ik} = a_{ik}^{(m)} / \sum_{\ell \notin F} a_{i\ell}^{(m)} \ , \ (i \in F, \ k \notin F) \ , \tag{12.4}$$

where $a_{ij}^{(m)} = \sum_\ell a_{i\ell}^{(m-1)} a_{\ell j}$ for $m > 1$ and $a_{ij}^{(1)} = a_{ij}$ (usually $1 \leq m \leq 3$), the traditional AMG interpolation *improves* it several times, each time using (12.3), then *truncates* the result (replaces every small w_{ik} by zero) and *renormalizes* (replaces each remaining w_{ik} by w_{ik}/w_i, so that the new $\{w_{ik}\}$ satisfies $\sum_k w_{ik} = 1$). (For indefinite systems one can also use $\widetilde{a}_{ij} = \sum_\ell a_{\ell i} a_{\ell j}$ instead of $a_{ij}^{(m)}$ in (12.4).)

Even if the interpolation (of whatever sort) is already good enough in the sense that it yields fast *asymptotic* multigrid convergence, it may be beneficial to add a number of compatible relaxation sweeps after the *first* interpolation to any particular level, and perhaps even a (smaller) number of such sweeps after the *second*. This is similar to the advised higher-order FMG interpolation (see, e.g., [6, §7.1]).

12.5. Gray AMG. Instead of the purely black-box AMG, which requires from its users no information other than the system of equations (the matrix A and the right-hand side b), a simpler and more advanced "gray-box" AMG can be constructed by requiring from its users additional information which is easy to supply, such as the geometrical location and the physical species of each unknown and each equation (cf. [8]). These can be very useful for cheaper and more effective execution of various AMG tasks, such as: decomposition of the domain for parallel processing; first crude selection of the coarse-level variables (next improved by the method of §3 above); defining the coarsening neighborhoods Ω_0 (see §4); organizing the boxes for BGS relaxation (§12.1); initializing the compatible-relaxation interpolation (see §12.4); etc. The gray box is of course essential for the nonlinear algebraic coarsening (as described in §8).

Sometimes it is even easy for the user to supply several entire coarse *levels* — their variables and equations — in addition to the target (finest) level. Indeed, in his discretization routine the target meshsize is often simply a parameter, and the routine can be applied several times with larger values of that parameter to create several coarser levels. This of course can drastically reduce the AMG setup cost, and permit the direct use of a nonlinear (FAS) solver.

Appendix A. Distributive coarsening. Very successful multigrid smoothers have been based on *distributive* relaxation schemes, where each relaxation step consists of *distributing* changes to several variables, according to some pre-assigned pattern. (For early simple examples, see [3], [16].) A general way to describe and design such schemes is in terms of *ghost variables* [4, §4.1], [6, §3.4]. In particular, the distributive Gauss Seidel (DGS) scheme, described in such terms, yields a generic method to design relaxation for general discretized PDE systems [7], [15]: The design of the distribution operator is usually done at the *differential* level, then translated to the discrete system.

For the discrete system (2.1), the ghost variables constitute a vector v such that $u = Mv$, where the (sparse) *distribution matrix* M has been designed (e.g., by discretizing the differential distribution operator) so that the resulting discrete system $AMv = b$ is easier to relax. For example, $\widetilde{A} = AM$ is often a diagonally dominant, or at least a definite, matrix, so that the system $\widetilde{A}v = b$ can be relaxed by the simple Gauss-Seidel scheme. (More generally, *weighted* relaxation schemes can be suitable for $\widetilde{A}$; see [6], [7], [15]). In practice, the variables v do not explicitly appear in the calculations (hence their ghostly name): each change δv in v implies a pattern of changes δu *distributed* to the genuine variables u through the relation $\delta u = M\delta v$; this is exactly the pattern one needs to efficiently relax the original system.

A special distribution matrix that can always safely be used is $M = A^T$, yielding the distributive relaxation of Kacmarz (see [37]). It can be used when nothing better is known, even though it is often substantially slower than other (distributive) schemes.

For all the cases where distributive relaxation is used, the definition (3.1) of the coarse-level variables can assume the more general form

$$u_k^c = \sum_i \widetilde{\mu}_{ki} v_i \ ,$$

where v_i are the ghost variables. We call this *distributive coarsening*. As a special but the most usual case, analogously to (3.2), the coarse variables can be identified with a *subset* of the ghost variables.

A *compatible* distributive relaxation, a version of the distributive relaxation that keeps the (distributive) coarse variables invariant, can easily be designed, and used to control the selection of these variables (cf. §3).

Notice that the distributive coarse-level variables, and the equations derived for them (e.g., by the method of §4) are different in nature from those of the fine level. In particular, in their turn they will generally *not* need distributive relaxation or distributive coarsening.

We have tested such a distributive coarsening for the highly disordered Dirac equations on critical or hot gauge fields (see Appendix C).

Appendix B. Directionality and coarse-level stability. The bare local discrete approximations to advection equations lack directionality. A simple example is the approximation $\big[u(x+h) - u(x)\big]/h$ to du/dx: whether the information travels from left to right or from right to left depends on whether we interpret this discrete equation as an equation for $u(x)$ or for $u(x+h)$. More generally, in any dimension, the sign of the stream-wise numerical viscosity associated with a given discrete advection equation, which is the sign that determines the direction of the information flow, depends on the point around which we Taylor-expand that discrete equation. In numerical schemes this is often determined by the placement of the boundary (or the boundary conditions), which in turn implies an exact geometrical position that can be associated with each discrete equation.

Since the directionality is missing in the local rows of A, it can be introduced into the coarse-level A^c, in the Galerkin form (10.1), only through the intergrid transform operators P and R. If these operators lack direction preference (as do the most common interpolation operators), then A^c may turn out unstable, or more generally, upon further coarsening steps A^c may grow increasingly unstable (in the sense that $\|A^c u^c\|$ may become smaller and smaller for some vectors u^c which are each highly oscillatory on the scale of its grid, unnaturally oscillating *in the advection direction*).

To avoid such instabilities, P and R should have the suitable directionality. This is quite naturally obtained with the traditional AMG interpolation (see §12.4), since the form (12.2) of the i-th discrete equation assigns its location to u_i, and this assignment would usually imply the intended directionality. With the more accurate and general derivation of interpolation described in §10, such directionality can be introduced via the definition of the objective functional Q_0: instead of $|a_{ij}|$ in (4.3), use a weighting q_{ij} that is much higher for points j upstream from 0 (the point *to* which an interpolation value is being constructed), and much lower for j downstream. For example, q_{ij} can be the value of u_0 upon solving equation (4.1) with the boundary conditions $u_\ell = \delta_{\ell j}$, ($\ell \in \Omega_0 - \Omega_0^I$); but there is no need or value to such a precise (and expensive) choice of the weights.

The problem of instability does not arise in the coarsening described in §4 (since a definite location is assigned to each coarse-level equation), but it does arise in that of §4.1. Here it can be eliminated by introducing direction dependence to the size of the weights $\{g_j\}$.

Appendix C. Application to Dirac equations. In collaboration with M. Rozantsev, we have applied algebraic coarsening methods to the $U(1)$ Dirac equations in 2D. This linear, first-order PDE system, discretized on a uniform grid with periodic boundary conditions (using the Euclidean staggered formulation described in [34]), is particularly challenging because it has highly disordered coefficients: They represent a quenched gauge field produced stochastically, its div is random while its curl is small at low temperatures and random at high temperatures; above a certain critical temperature the solution values have finite correlation lengths, which tend to zero as the temperature increases.

The AMG solver that we have developed employs Kacmarz relaxation and distributive coarsening with the Kacmarz distributions (see Appendix A), which is usually a very good distribution for first-order PDE systems such as this. The coarse-level set of variables is first selected *geometrically* (taking every fourth fine-level ghost variable, in a certain fixed 2D pattern). Then this set is enhanced using the compatible (distributive) relaxation tool (see §3), thereby adding another 10%–20% of the ghost variables to the coarse level. The coarse-level equations have been derived using the method of §4, for two choices of Ω_0^c: a 3×3 and a 5×5 coarse-grid stencil, each including also all those extra coarse variables added (following the compatible relaxation test) at the corresponding Ω_0^I region. The coarse-to-fine interpolation of corrections has been done by several passes of compatible relaxation (see §12.4). Recombination of iterants (§12.3) has also been employed.

The different tests we ran, on a 32×32 grid, proved that *all and each one* of the above devices is necessary for fast convergence in the more difficult cases. Very good asymptotic convergence rates have been obtained (e.g., a convergence factor of 0.2 to 0.3 per two-level cycle) with the 5×5 coarse stencil even for *hot* (practically random) gauge fields, provided some 15% extra points were added to the coarse level and upto 8 iterants were recombined. For *critical* gauge fields only a couple of iterants needed to be recombined. The detailed results are reported in [31].

REFERENCES

[1] D. BAI AND A. BRANDT, *Local mesh refinement multilevel techniques*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 109–134.

[2] A. BRANDT, *Multi-level adaptive solutions to boundary value problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] ———, *Multi-level adaptive techniques* (*MLAT*) *for singular perturbation problems*, in Numerical Analysis of Singular Perturbation Problems, P. W. Hemker and J. J. H. Miller, eds., Academic Press, New York, 1979, pp. 53–142.

[4] ———, *Numerical stability and fast solutions to boundary value problems*, in Boundary and Interior Layers — Computational and Asymptotic Methods, J. J. H. Miller, ed., Boole Press, Dublin, 1980, pp. 29–49.

[5] ———, *Multi-grid solvers for non-elliptic and singular-perturbation steady-state problems*, Weizmann Institute of Science, Rehovot, December 1981.

[6] ———, *Guide to multigrid development*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, 1982, pp. 220–312.

[7] ———, *Multigrid Techniques*: *1984 Guide, with Applications to Fluid Dynamics*, 1984, 191 pages, ISBN-3-88457-081-1; GMD-Studien Nr. 85; Available from GMD-AIW, Postfach 1316, D-53731, St. Augustin 1, Germany, 1984.

[8] ———, *Algebraic multigrid theory*: *The symmetric case*, in Preliminary Proc. Int. Multigrid Conf., Copper Mountain, Colorado, April 6–8, 1983; Appl. Math. Comput., 19 (1986), pp. 23–56.

[9] ———, *Rigorous local mode analysis of multigrid*, in Preliminary Proc. 4^{th} Copper Mountain Conf. on Multigrid Methods, Copper Mountain, Colorado, April 1989.

[10] ———, *Multilevel computations of integral transforms and particle interactions with oscillatory kernels*, Comput. Phys. Comm., 65 (1991), pp. 24–38.

[11] ———, *Rigorous quantitative analysis of multigrid*: I. *Constant coefficients two level cycle with L_2 norm*, SIAM J. Numer. Anal., 31 (1994), pp. 1695–1730.

[12] ———, *The Gauss Center research in multiscale scientific computation*, Electron. Trans. Numer. Anal., 6 (1997) pp. 1–34.

[13] ———, *The Gauss Center Research in multiscale scientific computation*, Gauss Center Report WI/GC–7, The Weizmann Institute of Science, Rehovot, March 1997. Appeared also in the MGNET Preliminary Proc. 8^{th} Copper Mountain Conf. on Multigrid Methods, April 1997.

[14] ———, *Multiscale Scientific Computation*: *Six year Research Summary*, in the web site http://www.wisdom.weizmann.ac.il/∼achi

[15] ———, *Barriers to Achieving Textbook Multigrid Efficiency in CFD*, ICASE Interim Report No. 32, NASA/CR-1998-207647. Gauss-Center Report WI/GC–10, June, 1998 (updated December, 1999).

[16] A. BRANDT AND N. DINAR, *Multi-grid solutions to elliptic flow problems*, in Numerical Methods for Partial Differential Equations, S. Parter, ed., Academic Press, New York, 1979, pp. 53–147.

[17] A. BRANDT AND B. DISKIN, *Multigrid solvers for non-aligned sonic flows*, SIAM J. Sci. Comput., in press.

[18] ———, *Multigrid solvers for the non-aligned sonic flow*: *the constant coefficient case*, Gauss Center Report WI/GC–8, Weizmann Institute of Science, October, 1997. Comput. & Fluids, 28 (1999), pp. 511–549.

[19] A. BRANDT AND I. LIVSHITZ, *Wave-ray multigrid method for standing wave equations*, Electron. Trans. Numer. Anal., 6 (1997), pp. 162–181.

[20] A. BRANDT, S. MCCORMICK AND J. RUGE, *Algebraic multigrid* (AMG) *for automatic multigrid solution with application to geodetic computations*, Institute for Computational Studies, POB 1852, Fort Collins, Colorado, 1982.

[21] A. BRANDT AND V. MIKULINSKY, *Recombining iterants in multigrid algorithms and problems with small islands*, SIAM J. Sci. Comput., 16 (1995), pp. 20–28.

[22] A. BRANDT AND D. RON, *Statistically optimal renormalization group flow and coarse-to-fine Monte-Carlo acceleration*, Gauss Center Report WI/GC–11, The Weizmann Institute of Science, Rehovot, May 1999.

[23] A. BRANDT AND D. RON, *Truncation-error free renormalization group flow and coarse-to-fine acceleration*, Phys. Rev. Lett., submitted.

[24] A. BRANDT AND D. RON, *Renormalization multigrid* (*RMG*): *Statistically optimal renormalization group flow and coarse-to-fine Monte Carlo acceleration*, J. Statist. Phys., submitted.

[25] A. BRANDT AND C. H. VENNER, *Multilevel evaluation of integral transforms on adaptive grids*, Gauss Center Report WI/GC–5, The Weizmann Institute of Science, Rehovot, April 1996. Multigrid Methods V, Lecture Notes in Computational Science and Engineering, Vol. 3, W. Hackbusch and G. Wittum, eds., Springer-Verlag, Berlin, pp. 21–24.

[26] A. BRANDT AND C. H. VENNER, *Multilevel evaluation of integral transforms with asymptotically smooth kernels*, Gauss Center Report WI/GC–2, The Weizmann Institute of Science, Rehovot, April 1995. SIAM J. Sci. Comput., 19 (1998), pp. 468–492.

[27] A. BRANDT AND I. YAVNEH, *On multigrid solution of high-Reynolds incompressible entering flows*, J. Comput. Phys., 101 (1992), pp. 151–164.

[28] A. KRECHEL AND K. STÜBEN, *Operator dependent interpolation in algebraic multigrid*, in Proc. Fifth European Multigrid Conf., Stuttgart, October 1–4, 1996; Lecture Notes in Computational Science and Engineering 3, Springer, 1998.

[29] J. E. MOREL, J. E. DENDY, JR., M. HALL AND S. WHITE, *A cell-centered Lagrangian-mesh diffusion differencing scheme*, JCP, 42 (1981), p. 20.

[30] J. E. MOREL, M. R. ROBERTS AND M. J. SHASHKOV, *A local support-operators diffusion scheme for quadrilateral r–z meshes*, JCP, 144 (1998), pp. 17–51.

[31] M. ROZANTSEV, *Multigrid Computation of the Fermionic Interactions in QCD Simulations*, Ph.D. Thesis, Weizmann Institute of Science, 2000.

[32] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.

[33] J. SCHRÖDER, U. TROTTENBERG AND H. REUTERSBERG, *Reuktionsverfahren fuer differenzengleichungen bei randwertaufgaben II*, Numer. Math., 26 (1976), pp. 429–459.

[34] H. SHARATCHANDRA, H. THUN AND P. WEISZ, *Sussking fermions on a Euclidean lattice*, Nuclear. Phys. B, 192 (1981), p. 205.

[35] E. SHARON, A. BRANDT AND R. BASRI, *Fast multiscale image segmentation*, Submitted to IEEE International Conference in Computer Vision and Pattern Recognition, South Carolina, June 2000.

[36] K. STÜBEN, *Algebraic multigrid* (*AMG*): *An introduction with applications*, GMD Report 53, March 1999.

[37] K. TANABE, *Projection methods for solving a singular system of linear equations and its applications*, Numer. Math., 17 (1971), pp. 203–214.

[38] W. L. WAN, *An energy-minimizing interpolation for multigrid methods*, CAM Report 97–18, Department of Mathematics, UCLA, April 1997.

[39] W. L. WAN, F. CHAN AND B. SMITH, *An energy-minimizing interpolation for robust multigrid methods*, CAM Report 98–6, Department of Mathematics, UCLA, February 1998. SIAM J. Sci. comp., to appear.

[40] T. WASHIO AND C. W. OOSTERLEE, *Krylov subspace acceleration for nonlinear multigrid schemes*, Electron. Trans. Numer. Anal. (1998).

[41] K. G. WILSON, *Problems in physics with many scales of length*, Scientific American, August 1979, p. 158.

[42] H. ZIMARE, *Beweis der sternenkonvergenz und untersuchung der stabilitaet beim verfahren der totalen reduktion*, Ph.D. Thesis, University of Cologne, 1980.