NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2000; **00**:1–18 *Prepared using* nlaauth.cls *[Version: 2002/09/18 v1.02]*

coarsening by compatible relaxation

O. E. Livne*

Computer Science Department, Gates 2B, Stanford University, Stanford, CA 94305-9025

SUMMARY

A proper coarse variable set for Algebraic Multigrid (AMG) solvers can be constructed using Compatible Relaxation (CR). For a given coarse set, CR is a modified relaxation scheme that keeps the values of the coarse variables intact. Its convergence factor is a predictor of the potential AMG efficiency, similarly to the smoothing factor in geometric multigrid. A practical CR-based coarse set construction algorithm is presented. Numerical results for for model problems are demonstrated. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: compatible relaxation; adaptive algebraic multigrid

1. ALGEBRAIC MULTIGRID

Algebraic Multigrid (AMG) solvers of large sparse linear systems of equations are based on multigrid principles but do not explicitly use the geometry of the grids; see for example [2, 6, 14, 15]. AMG has two phases: (a) Setup: several coarse levels are constructed, namely, *black-box coarsening* procedures produce a series of progressively smaller matrices, solely based on the entries of the original matrix; (b) Solve: the coarse levels are used in a standard multilevel cycling algorithm for solving the original matrix equations.

AMG has two main *advantages* over Geometric Multigrid (GMG): (a) it does not require specific tailoring for a new problem; rather, it aims at a black-box solver or preconditioner [15] for classes of matrices; (b) it can handle problems unapproachable by GMG, e.g., problems with no grids. Thus, AMG is widely for solving certain discretized Partial Differential Equations (PDEs) with smooth or disordered coefficients, on structured or unstructured grids, and even for problems that do not have a PDE origin.

The two main *drawbacks* of AMG are: (a) whenever GMG can be used, it is typically an order of magnitude faster than AMG, due to its optimized coarse stencils and AMG's additional setup phase; nevertheless, both are linear scaling ("$O(N)$") solvers, and AMG's overhead is quite justified by its extra robustness; (b) AMG's scope is rather limited; its coarsening procedures have been inadequate for general non-scalar, or high-order, or non-elliptic and anisotropic

*Correspondence to: Computer Science Department, Gates 2B, Stanford University, Stanford, CA 94305-9025. Phone: +1-650-725-4163. Fax: +1-650-723-3124. Email address: livne@stanford.edu

PDE systems, and also for non-variational discretizations. To address this second limitation, we need to systematically understand and improve AMG's ingredients: relaxation, coarse grids, and inter-level transfers. Improving AMG's coarse grid construction is the aim of this paper.

1.1. Adaptive AMG: Developer's Scheme

Recent ideas suggest that an improved AMG can be constructed by: (a) using physical information if available (e.g., stiffness matrices in finite element meshes [9]); (b) starting from an existing AMG solver and adapting it to its slow-to-converge errors [5, §17.2], [10]. The adaptation may be repeated in a close "self-correcting" loop until a good solver emerges.

In this spirit we propose the following "AMG Developer's Scheme" for approaching a new application with AMG (see Figure 1).

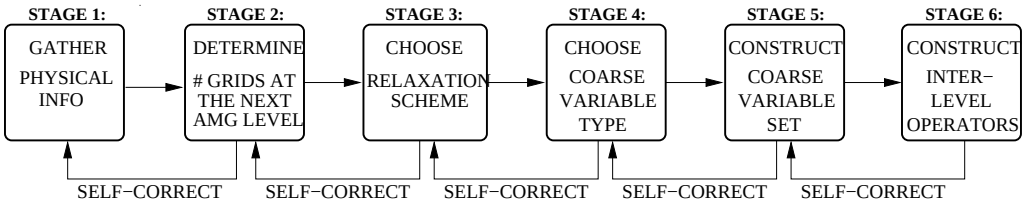

Figure 1. Adaptive AMG Developer's Six-Stage Scheme.

The scheme consists of six stages. At any given AMG coarsening level, simple physical information (such as the locations of variables in some low dimensional space) is first gathered; then, the number of coarse grids generated from the current (single) fine grid, is determined (usually one; a splitting to multiple coarse grids is required for example in indefinite problems [8, §6]; the word "grid" stands for a set of variables on a given level, even when they do not represent geometrically structured gridpoints). Next, we choose a suitable relaxation scheme that efficiently "smooths the error", thereby reducing its information content [4],[8, §2]. If possible, pointwise Gauss-Seidel (GS) is employed; more generally, one may use weighted distributed GS schemes [4, §11]. Stage 4 specifies the coarse variables type; they may range from subset of the original fine variables, to linear combinations of several fine ghost variables, where the fine variables themselves are also linear combinations of those ghost variables; see [4, App. A]. Stage 5 constructs the *set* of the next-coarse-level variables, by which algebraically smooth errors are to be approximated. Finally, the interpolation and restriction operators are computed, along with the corresponding Galerkin coarse grid operator [8, p. 15].

This scheme aims only at serving as a very general outline for AMG algorithms, in its current or a similar form; the ordering of stages may change, and self-correction may involve a simultaneous change in several stages (for instance, the relaxation scheme and the coarse variable set). Current adaptive AMG approaches assume that all first four stages have been performed once for all problems of a certain type. Stages 5 and 6 form the "on-line" AMG solver. However, all stages may require "real-time" decisions in future complex applications. Moreover, each stage may be revisited (in a closed "self-correcting" loop) based on the previous stages' result. For instance, if no good coarse set is feasible at stage 5, one should go back and change their type in stage 4. To realize the scheme, general criteria for gauging and controlling *each* stage are needed.

1.2. The Current Paper

We focus on stage 5: choosing the set of coarse variables. We thus assume that stages $1-4$ were already adequately executed (see §2).

Classical AMG selects the coarse variables from the fine (original) variables according so that each fine variable is "strongly coupled" to the coarse set [2, 6, 7, 14, 15]. However, algebraic connections can sometimes be misleading and inaccurate in determining the coarse variable set, as is often the case for non-M-matrices.

We propose an alternative approach that aims at more general linear systems, based on *Compatible Relaxation* (CR) [4, 8]. The CR convergence rate is a quantitative measure of the quantity of the coarse set, as explained in §3. In §4, we develop a CR-based practical algorithm for constructing the coarse variable set. Numerical experiments on several problems are described in §5. In §6, we include remarks on generalizations of the algorithm and suggestions for future research.

Our algorithms are mainly based on empirical observations; our theoretical contribution is local mode (Fourier) analysis of CR, illustrated in Appendix III.

2. FRAMEWORK

For simplicity, we describe the construction of a coarse set for an $N \times N$ linear system of equations

$$Au = f \tag{1}$$

under the following assumptions:

- *Stage 1:* A is real symmetric (or Hermitian) sparse matrix; each of the vector entries, u_i, represents an unknown at a gridpoint in $\mathbb{R}^d$. Typically, $1 \leq d \leq 3$. We assume that the ith row of A has the form

$$(Au)_i = \sum_{j \in \mathcal{A}_i} a_{ij} u_j, \qquad i = 1, \ldots, N, \tag{2}$$

  where the stencil $\mathcal{A}_i$ includes i and some of the neighboring gridpoints j. Many discretized PDEs satisfy this requirement.
- *Stage 2:* We construct a single coarse variables set, discarding indefiniteness and "multiple coarsening" ideas (see [8]).
- *Stage 3:* We assume that point Gauss-Seidel (GS) relaxation is a good algebraic smoother [8, p. 5], [15].
- *Stage 4:* The coarse variable set C is a *subset* of the fine (original) variables $\{1, \ldots, N\}$. Thus, for each fine level vector $\tilde{u}$ (and each error vector, $e = \tilde{u} - u$, where u is the exact solution of (1)), there corresponds a unique coarse level vector, $\tilde{u}^c$ (respectively, e^c).

Our research questions are as follows:

1. How should we measure the quality of a given coarse variables set C ?
2. How should we construct a C of good quality?

3. COMPATIBLE RELAXATION IN THEORY

The set of coarse variables should be chosen so that their values determine, up to fast local processing, the values of the fine set [8, p. 14]. That is, each of the remaining fine variables (an "F-variable", where $F := \{1, \ldots, n\} \backslash C$ [15]) is a linear combination of a local set of neighboring C-variables, with a very weak remnant dependence on coarse variables outside the neighborhood that decays exponentially with the distance from the F-variable [8, p. 14]. For such a C, each coarse variable is defined locally by few fine variables, so it also mostly depends on its coarse neighbors; hence it is feasible to construct a *local* coarse level *equation* for each C-variable, whose accuracy exponentially increases with the size of its stencil's radius.

In classical AMG, this is achieved by choosing C so that each fine variable is "strongly coupled" to C through one or several equations [2, 6, 7, 14, 15]. A more general criterion for gauging, and a practical method to control, the quality of this set can be based on sweeps of *compatible relaxation*.

Definition 1. *Compatible Relaxation (CR) is a fine level relaxation scheme that keeps the coarse-level variables invariant (i.e., it keeps the fine-level configuration compatible with the same coarse-level configuration: u changes, but u^c does not).*

To monitor the error reduction in CR sweeps, we choose $f = 0$ in (1) so that the solution is $u = 0$. Note that $e = \tilde{u} - u = \tilde{u}$. In this case, CR for a given coarse set C consists of relaxing on

$$Ae = 0, \tag{3}$$

while keeping the constraints

$$e_i = 0, \qquad \forall i \in C. \tag{4}$$

Note that CR does not depend on f, and refers to the *error* e rather than the *solution* u. There are at least two possible types of CR.

- *Concurrent* CR: each relaxation step on (3) simultaneously keeps (4). Thus, concurrent compatible GS relaxation is equivalent to an "F-relaxation" [15] , i.e. it simply avoids relaxing those variables of e that belong to e^c. For more general types of coarse variables we may need to simultaneously update one or several fine variables in a "box relaxation" manner [1, §5] at each relaxation step.
- *Habituated* CR: a full relaxation sweep is performed on (3), followed by a sweep of imposing the constraints on e^c. Here, compatible GS is a full GS sweep, followed by setting all the coarse variables to 0. More generally, the constraints may be relaxed by one or a few sweeps of *Kaczmarz* relaxation.

Habituated CR is typically easier to implement than concurrent CR, because no change is required in the original relaxation for (3).

We base our C-construction on the following guidelines:

- The set C is guaranteed to be good if and only if compatible relaxation exhibits uniformly fast convergence rates [4, p. 2].
- Where these rates are too slow, a diluted "independent subset" of the slow-to-converge variables should be added to C. Following this modification of C, the convergence factor of CR should be checked again.

- The CR convergence factor is a predictor to the potential AMG efficiency, similarly to the *smoothing factor* in geometric multigrid [1]; however, it cannot be expected to be a precise measure of the AMG convergence factor, if only because it is possible to perform CR in different manners (e.g., concurrent or habituated) and obtain different numbers. Still, numerical results show that the two types of CR give the same *qualitative indication* of C (see §6).

Rigorous justification of these guidelines is under ongoing research. An recent fundamental paper by Falgout and Vassilievski rigorously explains the role of CR in the AMG framework [12]. In this paper we develop *local mode analysis* of CR for certain geometric grids, that may help understand its convergence rates through simple examples (see Appendix III).

4. PRACTICAL CONSTRUCTION ALGORITHM

We developed a practical construction algorithm for selecting the coarse level set that is based on Compatible Relaxation (CR). The CLC (Coarse Level Construction) algorithm was implemented in MATLAB and tested on simple model problems (see §5). A pseudo-code is given in Appendix II. The main flow of the algorithm is described in Fig. 2.

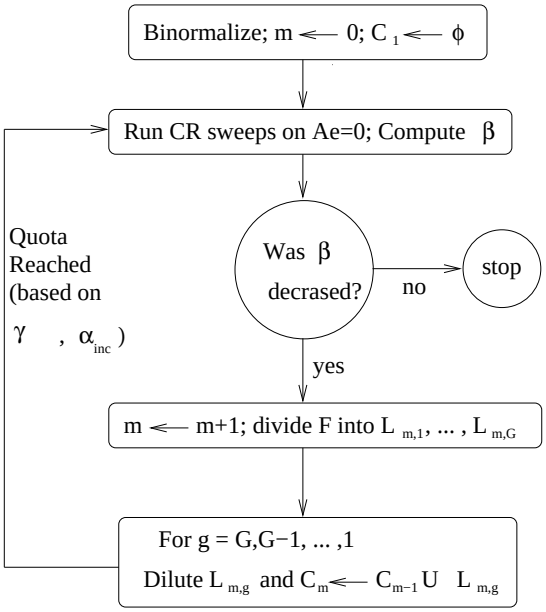

Figure 2. The algorithmic flow of CLC.

First, A is binormalized, that is, pre- and post-multiplied by a diagonal matrix chosen to make all of A's rows near unit l_2 norm. We start with an empty coarse set C, and add variables to C in a series of $\nu \leq \nu_{\max}$ steps. This yields a sequence of sets

$$\emptyset = C_1 \subseteq C_2 \subseteq \ldots \subseteq C_\nu.$$

At each step $m = 1, \ldots, \nu$, we compute the CR convergence rate and decide whether C_m is an adequate coarse set or not. If it is not, we assign each F_m-variable a measure of its slowness in CR sweeps, where $F_m := \{1, \ldots, N\} \backslash C_m$. Based on this measure, we form a list of candidate variables, sorted by their "slowness measure". This list is shrunk into a diluted independent set L_m of F-variables that are appended to C_m to form C_{m+1}. The CR convergence rate is then re-computed, and a new candidate list is formed, etc., until a good enough coarse set emerges.

4.1. Binormalization

Prior to starting the CLC sweeps, we *binormalize* A. That is, A is diagonally scaled by a diagonal matrix D, so that $\tilde{A} := DAD$ satisfies

$$\left| \sum_{j=1}^{N} |\tilde{a}_{ij}|^2 - 1 \right| \leq \xi, \qquad i = 1, \ldots, N, \tag{5}$$

for some small $\xi > 0$. Note that (5) can be easily achieved with $\xi = 0$ by diagonally left-scaling A to $\tilde{D}A$, where

$$\tilde{D}_{ii} := \left(\sum_{j=1}^{N} |a_{ij}|^2 \right)^{-\frac{1}{2}}, \qquad i = 1, \ldots, N,$$

however $\tilde{D}A$ is no longer symmetric. We developed a fast iterative algorithm that computes D to any desirably small ξ. The algorithm costs $O(N \log(1/\xi))$ operations, and is described and analyzed in [13].

We hereafter denote the resulting binormalized matrix by A and the corresponding scaled right side by f. Let $\tilde{u}$ be an approximate solution of (1), $e := u - \tilde{u}$ the error vector, and $r := Ae$ the residual vector. In [8, pp. 4–5], the A-normalized l_2 norm is used to compare residuals with errors. Let

$$\|e\| := \left(\sum_{i=1}^{N} |e_i|^2 \right)^{\frac{1}{2}}, \qquad \|r\|_{/A} := \left(\sum_{i=1}^{N} \frac{|r_i|^2}{\sum_{j=1}^{N} |a_{ij}|^2} \right)^{\frac{1}{2}}. \tag{6}$$

Note that $\|r\|_{/A} \approx \|r\|$ because of the binormalization. For all errors, $\|Ae\| \leq \|e\|$; for random errors, $\|Ae\|$ and $\|e\|$ are comparable in size; errors that are slow to convergence in relaxation are characterized by $\|Ae\| \ll \|e\|$ and are called "algebraically smooth" [8, p. 5],[15]. We assume that GS relaxation is a good "smoother", that is, starting from a random error it requires only a few sweeps to arrive at an algebraically smooth error. Therefore, the binormalization makes it possible to compare r and e in the same l_2 norm. Moreover, individual error values $|e_i|$ may be "compared" (even if not necessarily "comparable in size"): at any rate, if A would not be binormalized, the values $\{|e_i|\}_i$ could have assumed an arbitrary diagonal scaling and would not be comparable to each other.

4.2. CR Rate and Slowness Measure

Let m be the current CLC step. We start the step by running η CR tests. In each test k, $k = 1, \ldots, \eta$, we start with the initial values

$$e_i^{(k,0)} := \begin{cases} \text{random}[\frac{1}{2}, 1], & i \in F_m; \\ 0, & i \in C_m, \end{cases} \tag{7}$$

and perform λ CR sweeps on $Ae^{(k,0)} = 0$, yielding the result $e^{(k,\lambda)}$, where $e^{(k,l)}$ is the error after l CR sweeps. Note that $e^{(k,0)}$ is bounded away from 0 at the F-points, to avoid illusory small values at certain F-variables; our experiments show that this initial guess is to be preferred over random$[-1, 1]$. After the CR tests we compute the *slowness measure*

$$E_i := \max_{1 \le k \le \eta} |e_i^{(k,\lambda)}|, \forall i \notin C_m. \tag{8}$$

Variables with large E_i must be considered candidates to be added to C_m, because there is at least one random configuration e for which e_i was not effectively reduced by CR. The multitude of tests ensure that variables $i \in F_m$ cannot be waived from the candidate list only because $|e_i|$ appears small after several CR sweeps, for a specific random initial error.

In addition, we compute the estimated CR convergence rate by

$$\mu_{\text{cr}} := \frac{1}{\eta} \sum_{k=1}^{\eta} \frac{\|e^{(k,\lambda)}\|}{\|e^{(k,\lambda-1)}\|}. \tag{9}$$

4.3. Stopping Criterion

If C is empty, CR is equivalent to relaxation and μ_{cr} is typically close to 1; if $C = \{1, \ldots, N\}$, $\mu_{\text{cr}} = 0$ but no coarsening of the original system was actually achieved. This suggests that the quality of C should be measured by the estimated *cost-effectiveness* of AMG cycles that are based on C.

Let $\alpha := \frac{|C|}{N}$ be the coarsening ratio; assuming the coarse level problem is recursively solved by similar coarser levels, the total cost per fine variable of an AMG cycle is about

$$W = 1 + \gamma\alpha + \gamma^2\alpha^2 + \ldots = \frac{1}{1 - \gamma\alpha}. \tag{10}$$

Here $1 \le \gamma < \infty$ is the cycle index and is a prescribed parameter for CLC, usually depending on the dimension d of the problem. In most problem, however, the Galerkin coarse level operator has a larger stencil than the original fine stencil of A, especially when the original problem is defined on a structured grid. For instance, the 5-point Laplacian has $n_z = 5$ nonzero entries per row, and its Galerkin coarse operator (based on linear interpolation) has $sn_z = 9$ nonzeros per row, where s is the ratio of stencil sizes at the coarse and fine levels; in the example, $s = 1.8$. As a result, we replace (10) by

$$W = 1 + s\gamma\alpha + s\gamma^2\alpha^2 + \ldots = \frac{1 + (s-1)\gamma\alpha}{1 - \gamma\alpha}. \tag{11}$$

This formula assumes that the fill-in does not continue beyond the first coarse level. A robust default value for s is 1, unless we expect a fill-in of a "narrow" fine level stencil. Typically, $1 \le s \le 5$.

In geometric multigrid, the estimated cycle convergence factor (per fine level relaxation sweep) is $\overline{\mu}$, where $\overline{\mu}$ is the smoothing factor of relaxation and the inter-level transfers are assumed to be accurate enough in approximating low frequency errors. Here we use the similar estimate $\mu_{\rm cr}$ for the AMG convergence factor per relaxation sweep, as the error reduction per unit work is $\mu_{\rm cr}^{\frac{1}{W}}$. Because $\mu_{\rm cr}$ below a certain level cannot be attained by AMG cycles, similarly to small smoothing factors in geometric multigrid [1, §3], we use the criterion

$$\beta := \max\{.1, \mu_{\rm cr}\}^{\frac{1}{W}} \tag{12}$$

to assess the quality of C. The idea goal is to minimize β over all sets C. There is no need for an exact optimization, though: many coarse sets C may yield sub-optimal β values that are acceptable for AMG design. A typical "slice" of β versus α is unimodal as shown in Fig. 3.

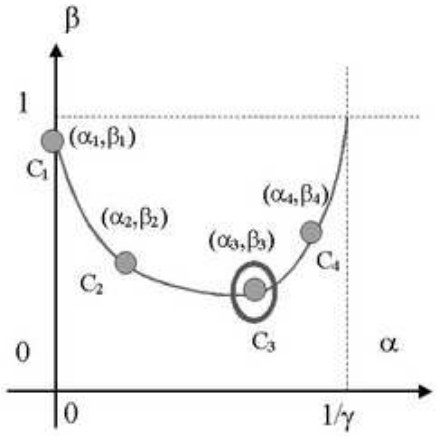

Figure 3. A plot of β versus α in the CLC algorithm. A series of progressively larger sets $C_1 \subseteq C_2 \subseteq \ldots$ is constructed, with their corresponding coarsening ratios $\alpha_1, \alpha_2, \ldots$, CR rates $\mu_1, \mu_2, \ldots$ and cost effectiveness measures $\beta_1, \beta_2, \ldots$.

We perform CLC steps as long as β can be (substantially) decreased. Specifically, if β increases over two consecutive CLC steps (or if almost no variables are added to C_m after forming the candidate list L_m; see §4.4), we stop and output the set C_m that yielded the minimal β over all steps $m = 1, \ldots, \nu$ (e.g., in Fig. 3, C_3 is chosen).

4.4. Forming A Candidate List

If the quality of C_m is not yet acceptable by the β-criterion, we form a candidate list of variables L_m to be appended to C_m. We initialize $L_m = F_m$ and divide it into G equivalence classes $L_{m,1}, \ldots, L_{m,G}$ (e.g., $G = 10-30$; we used $G = 30$). Groups are sorted by descending E-value; there is no need for a costly sort *within* each group. The groups are scanned by descending E-value order; within each group, if i is still in L_m, we add it to C_m and remove i and all its algebraic neighbors of depth at most t [15] from L_m (or neighbors a with connection above a certain threshold δ only, if "connections" can be safely used; for simplicity and generality we use $t = 1$ and $\delta = 0$). This "dilution" process is continued until there are no L_m variables

in the current group. Then we proceed to the next group; and so on. To better control the amount of variables appended to C_m at each CLC step, we add a limited quota of variables per step. Let $\alpha_m := |C_m|/N$, $m = 1, \ldots, \nu$; then α_{m+1} should not exceed

$$(1 - \alpha_{\text{inc}})\alpha_m + \alpha_{\text{inc}}\alpha_{\max}, \qquad \alpha_{\max} := \frac{1}{\gamma}. \tag{13}$$

Variables that are not added to C_m but remain slow to converge will be detected and appended during the next steps.

4.5. Control Parameters

The CLC algorithm can be controlled by the various parameters involved. Its main computational cost is in the CR sweeps, i.e. $O(\nu\eta\lambda N)$ operations. We used $\nu = 3, \eta = 4$ and $\lambda = 20$, thus at most $3 \times 4 \times 20 = 240$ CR sweeps are performed. This number may be reduced for practical applications.

The ratio s is set to 1 unless geometric information is known. The cycle index γ was set to 1.5. Our experiments show that s and γ do not significantly affect the results.

Finally, α_{inc} controls the number of actual CLC steps; to make sufficiently fine steps along the curve $\beta(\alpha)$, α_{inc} should not be excessively high. We used $\alpha_{\text{inc}} = .25$.

5. NUMERICAL EXPERIMENTS

5.1. An Example

we first consider the 2D Laplace equation

$$LU := -U_{xx} - U_{yy} = F, \qquad (x, y) \in [0, 1]^2, \tag{14}$$

with Dirichlet boundary conditions. Eq. (14) is discretized on a uniform 16×16 finite element mesh with meshsize $h := \frac{1}{16}$ in x and y. The resulting stencil is

$$L^h = \frac{1}{h^2} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \tag{15}$$

Eliminating the boundary variables and lexicographically ordering the internal variables U^h_{kl}, F^h_{kl} at the gridpoints $\{(kh, lh) : 1, k, l, \leq n - 1\}$. Thus we obtain a matrix system $Au = f$ that satisfies the requirements of §2. Here $N := (16 - 1)^2 = 225$. The corresponding binormalized stencil (for a sufficiently small h and away from the boundaries) reads

$$\begin{bmatrix} -.112 & -.112 & -.112 \\ -.112 & .943 & -.112 \\ -.112 & -.112 & -.112 \end{bmatrix}. \tag{16}$$

We start with $C_1 = \emptyset$ and obtain $\alpha_1 = 0, \mu_1 = .94, \beta_1 = .94$. Next, $L_1 = \{1, \ldots, N\}$ is divided into 30 equivalence classes by the slowness measure E (eq. (8)). The first equivalence class $L_{1,1}$ yields the variables marked by gray circles in Fig. 4a. We then choose a diluted set over $L_{1,1}$ and add it to C_1 (Fig. 4b). This process is repeated for $L_{1,2}$, and gives the

coarse set shown in Fig. 4c; then, $L_{1,3}, \ldots$ are processed until the quota (13) is reached. The resulting set is C_2 (Fig. 4d), for which $\alpha_2 = .23, \mu_2 = .50, \beta_2 = .63$. Similarly, we obtain C_3 (Fig. 4e) by defining L_2 and choosing an independent set over it that is appended to C_2. Here $\alpha_3 = .34, \mu_3 = .41, \beta_3 = .64$. Note that $\mu_{\rm cr}$ improved but β did not; thus, we could stop and set $C := C_3$ (Fig. 4f) or try to generate C_4 to see whether β may further reduced.

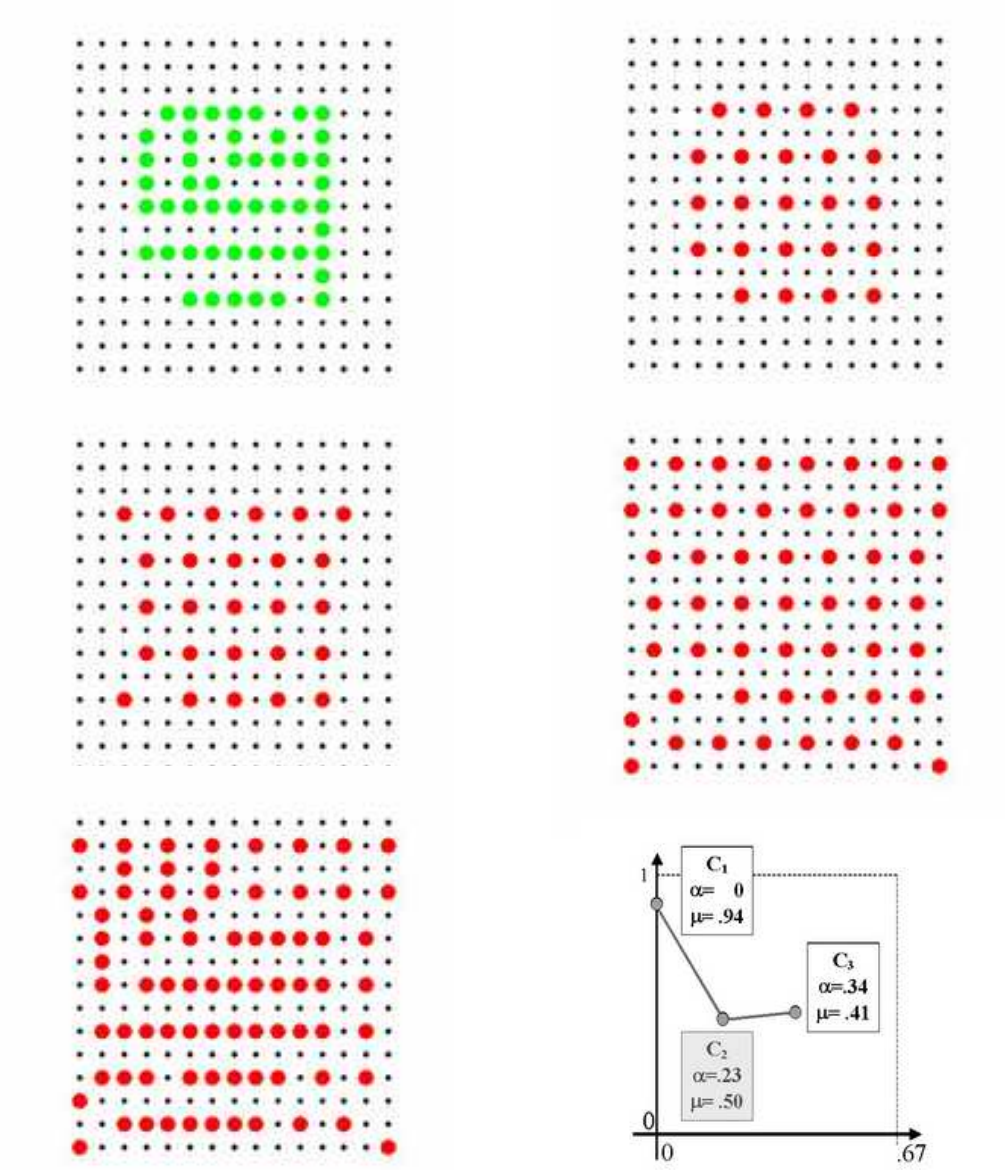

Figure 4. (a) the set $L_{1,1}$; (b) the diluted $L_{1,1}$; (c) the coarse set after adding a diluted set over $L_{1,2}$; (d) the set C_2; (e) the set C_3; (f) the curve $\beta(\alpha)$.

5.2. Several Model Problems

We illustrate the CLC algorithm on several discretized PDEs. Our model problems are as follows.

- *Problem A:* isotropic 5-point Laplacian,
$$U_{xx} + U_{yy} = f, \quad (x, y) \in \Omega := [0, 1]^2, \quad U = 0 \text{ on } \partial\Omega.$$
The normalized discretization stencil (for an $n \times n$ grid with a sufficiently small meshsize $h := \frac{1}{n+1}$ and far from boundaries) is
$$\begin{bmatrix} & -.224 & \\ -.224 & .894 & -.224 \\ & -.224 & \end{bmatrix}.$$
- *Problem B:* anisotropic 5-point Laplacian,
$$U_{xx} + \varepsilon U_{yy} = f, \quad (x, y) \in \Omega,$$
$$\begin{bmatrix} & -.0004 & \\ -.408 & .817 & -.408 \\ & -.0004 & \end{bmatrix};$$
boundary conditions as in Problem A; $\varepsilon = 10^{-3}$.
- *Problem C:* anisotropic rotated Laplacian,
$$(c^2 + \varepsilon s^2)U_{xx} + 2(1 - \varepsilon)csU_{xy} + (\varepsilon c^2 + s^2)U_{yy} = f, \ (x, y) \in [0, 1]^2,$$
$$\begin{bmatrix} .131 & -.048 & -.131 \\ -.360 & .817 & -.360 \\ -.131 & -.048 & .131 \end{bmatrix}$$
boundary conditions as in Problem A; here $c := \cos\theta, s := \sin\theta, \varepsilon = 10^{-3}$, and $\theta = 20^0$.
- *Problem D:* the biharmonic operator,
$$\Delta^2 U = U_{xxxx} + 2U_{xxyy} + U_{yyyy} = f, \qquad (x, y) \in [0, 1]^2,$$
$$\begin{bmatrix} & & .038 & & \\ & .077 & -.308 & .077 & \\ .038 & -.308 & .769 & -.308 & .038 \\ & .077 & -.308 & .077 & \\ & & .038 & & \end{bmatrix}$$
with $U = 0$ and $\partial U/\partial n = 0$ at the boundary.

Classical AMG performs well for problems A,B,C [15]; problem D is considered much harder for it, because the stencil does not form an M-matrix. Figures 5 and 6 shows the results of CLC for all model problems and $h = \frac{1}{32}$. For each problem, we show the fine and coarse variables locations in space, and the curve $\beta(\alpha)$.

The final β obtained is similar for all model problems, ranging from .68 to .77. The coarse grids obtained are not geometrically ordered, yet yield quite a small β. For instance, standard uniform coarsening of Problem A gives $\alpha = .25, \mu_{\text{CR}} = .547, and \beta = .686$ (see Appendix III), versus $\beta = .691$ found by CLC. Moreover, while CLC addresses Problems B and C with the same efficiency, in Problem B standard geometric coarsening would fail, and "semi-coarsening" is required [1]; for Problem C, even "semi-coarsening" requires a delicate construction of inter-level transfers [11].

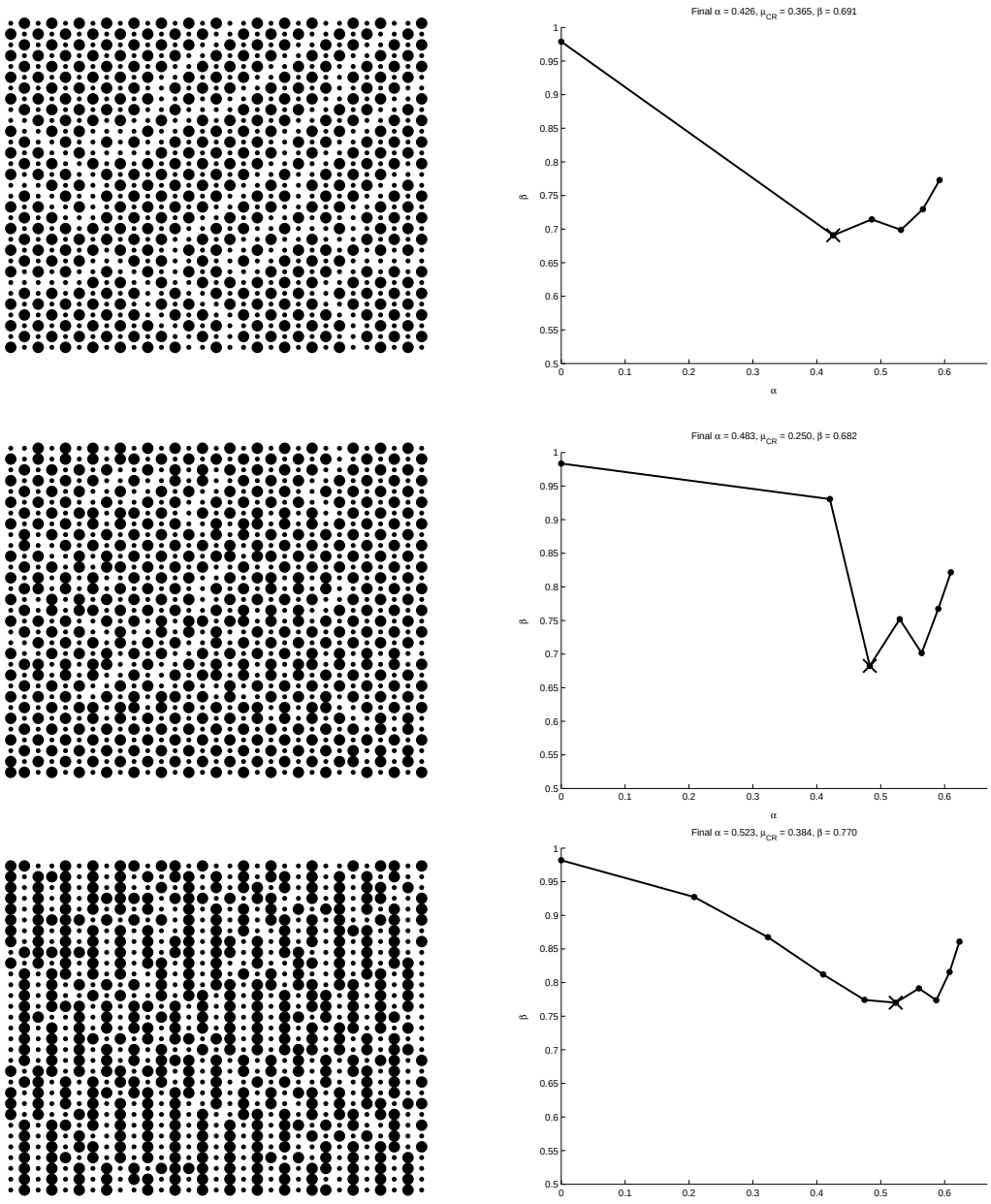

Figure 5. Numerical results for problem A (first row) through C (third row). Left figures: The coarse grid (large circles) and the original fine grid (small circles). Right figures: β as a function of α during the CLC steps. For each figure, the final step is marked with an "x".

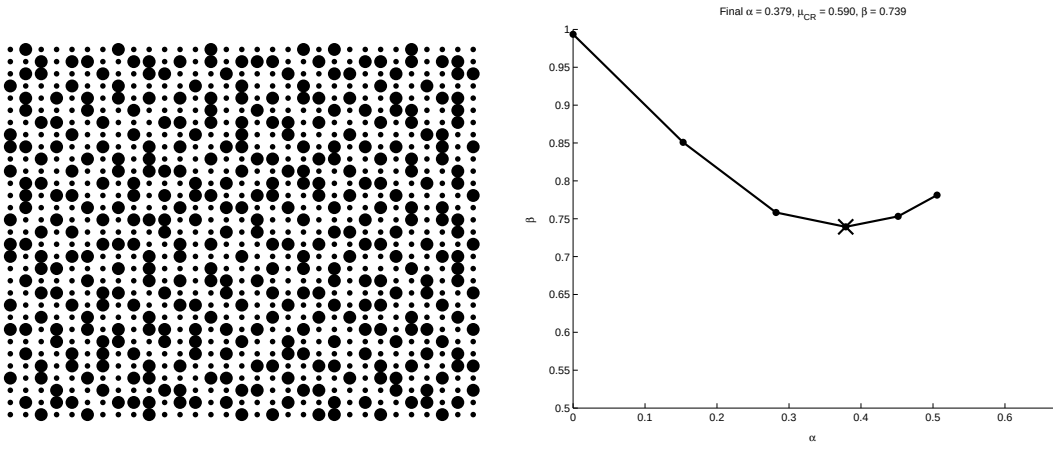

Figure 6. Numerical results for problem D. Notation are as in Fig. 5.

6. CONCLUDING REMARKS

Our main contribution is a practical coarse variable set construction algorithm for AMG, based on the quantitative measure of compatible relaxation convergence rate. The algorithm resulted in "adequate" coarse variables sets with respect to this measure, for all presented model problems, regardless of the meshsize length. In addition, we developed local mode analysis for CR to supplement the algorithm with a theoretical analysis tool.

It should be emphasized that our CLC algorithm can be regarded only as an initial step toward improved AMG coarse level construction and realization of the "Developer's Scheme". Directions for possible future research are outlined as follows.

- The CLC algorithm was illustrated here on simple model problems; however, it makes a "stand-alone" module that can be immediately incorporated into the existing AMG software. It would then be possible to assess its effect in practical applications. The cost of CLC is $O(N)$, but the constant can be rather large (up to several hundred fine grid work units). This cost can be reduced, for instance, by better minimum-search methods along the curve $\beta(\alpha)$ to reduce the number of stages. At any rate, it is important to minimize β only *approximately*.
- The algorithm selects its candidates for the coarse level set by the absolute value of the error at each variable, after several CR sweeps. This criterion may be adequate for many scalar PDEs, but needs to be generalized for PDE *systems*, indefinite systems, problems with small diagonal entries, etc. The extension to such cases, as well as to non-symmetric matrices, also involves a change in the relaxation scheme. GS relaxation may be replaced by Kaczmarz relaxation, or other appropriate choices [4, §17.2].
- Much better results may be obtained by choosing a different *type* of coarse variables (stage 4 in the scheme). Preliminary experiments show that employing *local averages* as coarse variables can significantly improve β. For instance, in Problem C (§5), aligning the coarse grid with the flow might be much easier if the coarse variables are certain

weighted local averages of the fine variables, instead of individual fine variables. More generally, if no small enough β *can* found for a certain problem, it is a clear indication to go back and "self-correct" stage 4 (or other stages, e.g., the relaxation scheme). The current CLC algorithm is designed for choosing subsets of the fine variables and needs to be generalized to such cases. Alternatively to adding coarse variables to C, subsets of the slow to converge variables can be relaxed *simultaneously*; that is, the relaxation scheme is to be modified as well as the coarse set, during CLC steps.

- CR can be performed in different manners, e.g., habituated CR. Experiments on the model problems of §5 (that are not given in this paper) show that although habituated CR is slower than concurrent CR, the qualitative unimodal behavior of β versus α in CLC is similar; see also Appendix III.2. Habituated CR may in practice be more easily implemented than concurrent CR and needs to be further investigated.

Study of other parts of the scheme is in progress. In particular, the evolving field of adaptive AMG [4, §17.2],[8],[10] relates to deriving interpolation operators in stage 6 that would realize the CR predictions. An interesting conjecture is that *concurrent CR rates can always be attained by the actual AMG cycles, provided that the inter-level transfers are correctly constructed* (see Appendix III.1). Otherwise, CR rates should be viewed only as a rough guideline for the potential AMG efficiency. As a first check, AMG may be tested in a two-level form on model problems A-D of §5, with the grids constructed by CR, to observe whether the predicted efficiency can be attained.

Systematic development of all stages, with the self-correction strategy within each stage, may well lead to substantial improvement of existing AMG methods.

APPENDIX

II. PSEUDO CODE OF CLC

Algorithm CLC (Coarse Level Construction)
Input: $\nu, \eta, \lambda, s, \gamma, \alpha_{\text{inc}}, G, d, \delta$
Output: C

1. Set $C \leftarrow \emptyset$, $\alpha_{\max} \leftarrow \frac{1}{\gamma}$, $\beta_{\min} \leftarrow -1; \mu_0 \leftarrow .1$.
2. For $m = 1$ to ν do steps 3–18.
3. If $(m == 1)$ goto step 11.
4. If $(m == 2)$ $\alpha_c \leftarrow 1$; else $\alpha_c \leftarrow (1 - \alpha_{\text{inc}})\alpha_{m-1} + \alpha_{\text{inc}}\alpha_{\max}$.
5. $L \leftarrow \{1, \ldots, n\}/C; E_{\max} \leftarrow \max_{1 \le i \le N} E_i; E_{\min} \leftarrow \min_{1 \le i \le N} E_i;$
   $\sigma \leftarrow \frac{E_{\max} - E_{\min}}{G}$.
6. Divide L into G equivalence classes $L_1, \ldots, L_G$, where
$$L_g := \{i \quad | \quad E_{\min} + (g-1)\sigma \le E_i \le E_{\min} + g\sigma\}.$$
7. For $g = G$ downto 1, do steps 8–10.
8. For $i = 1$ to $|L_g|$ do steps 9–10.
9. If $(i \in L_g), C \leftarrow C \cup \{i\}$;
   $L_g \leftarrow L_g \backslash (\{j \in \mathcal{A}_i \quad | \quad |a_{ij}|/|a_{ii}| > \delta \text{ or } j = i\} \cap L_g)$.

10. If $(\frac{|C|}{N} > \alpha_c)$ goto step 11.
11. Set $\mu_{\rm CR} \leftarrow 0$; For $i = 1$ to N $E_i \leftarrow 0$.
12. For k = 1 to η do steps 4–6.
13. Set $e_i^{(k)} = 0$ for all $i \in C$ and $e_i^{(k)} = \text{random}[0.5, 1]$ for $i \in \{1, \ldots, n\}/C$.
14. Perform λ CR sweeps on $Ae_k = 0$. Let $e_k^{(\lambda)}$ be the result after j sweeps.
15. Set $\mu \leftarrow \frac{\|e_k^{(\lambda)}\|_2}{\|e_k^{(\lambda-1)}\|_2}$, $\mu \leftarrow \mu + \frac{\mu}{\lambda}$; For $i = 1$ to N, $E_i \leftarrow \max\{E_i, |e_i^{(k)}|\}$.
16. $C_m \leftarrow C; \alpha_m \leftarrow |C_m|; W_m \leftarrow \frac{1+\gamma\alpha_m(s-1)}{1-\gamma\alpha_m}; \beta_m \leftarrow \max\{\mu_0, \mu\}^{\frac{1}{W_m}}$.
17. If $((m = 1)$ or $(\beta_m < \beta_{\min}))$, $\beta_{\min} \leftarrow \beta_m, \tilde{m} \leftarrow m$.
18. If $((m = \nu)$ or $(\alpha_m \geq \alpha_{\max})$ or $((m > 1)$ and $(|\alpha_m - \alpha_{m-1}| < 10^{-5}))$ or $((m \geq 3)$ and $(\beta_m > \beta_{m-1})$ and $(\beta_{m-1} > \beta_{m-2}))$, $C \leftarrow C_{\tilde{m}}$ and stop.

III. LOCAL MODE ANALYSIS OF CR

III.1. Concurrent CR

As an example of local mode analysis of compatible relaxation, we consider the 5-point Laplacian operator on a uniform infinite or periodic grid with meshsize h, and compute the asymptotic convergence rate of concurrent compatible Gauss-Seidel relaxation, for a uniform $2h$ coarse grid. For further details on local mode analysis in multigrid, see [16] and its references list.

Consider a general function on the h-grid that vanishes at the coarse set, with a scaled frequency $\theta := (\theta_1, \theta_2) \in [-\pi, \pi]^2$; namely,

$$e_{i_1,i_2} := e^{(i_1\theta_1 + i_2\theta_2)} \times \begin{cases} A, & i_1 \text{ even}, i_2 \text{ even}; \\ B, & i_1 \text{ odd}, i_2 \text{ even}; \\ C, & i_1 \text{ even}, i_2 \text{ odd}; \\ 0, & i_1 \text{ odd}, i_2 \text{ odd}, \end{cases} \qquad \forall i_1, i_2 \in \mathbb{Z}. \tag{17}$$

This function is invariant under compatible relaxation. The result of CR on e is a function $\overline{e}$ that has the form (17) with different coefficients $\overline{F} := (\overline{A}, \overline{B}, \overline{C})^T$ replacing $F := (A, B, C)^T$. The relation between F and $\overline{F}$ is

$$M_-(\theta)\overline{F} = M_+(\theta)F,$$

$$M_-(\theta) := \begin{pmatrix} 4 & -e^{-i\theta_1} & -e^{-i\theta_2} \\ -e^{-i\theta_2} & 4 & 0 \\ -e^{-i\theta_1} & 0 & 4 \end{pmatrix}, \qquad M_+(\theta) := \begin{pmatrix} 0 & e^{i\theta_1} & e^{i\theta_2} \\ e^{i\theta_2} & 0 & 0 \\ e^{i\theta_1} & 0 & 0 \end{pmatrix}.$$

Thus, the *Fourier symbol* $\hat{M}_C(\theta)$ of concurrent CR is

$$\hat{M}_c(\theta) := M_-^{-1}(\theta)M_+(\theta). \tag{18}$$

In fact, the traditional definition of $\hat{M}_c(\theta)$ would result in a similarity transformation on (18), because we need to transform the function in (17) to the Fourier exponents basis and back, to get the "response" of CR to invariant subspaces of Fourier components; however, the asymptotic convergence rate is the same when we work with (18). By numerical (or

analytical) maximization of $\rho(\hat{M}_c(\theta))$ over all θ, we obtain the asymptotic convergence factor for concurrent CR (see also Fig. 7). Specifically,

$$\mu_{\text{cr}} := \max_{\theta\in[-\pi,\pi]^2} \rho(\hat{M}_c(\theta)) = \rho(\hat{M}_C(\pm\pi,\pm\pi)) = \frac{1+2\sqrt{2}}{7} \approx .547. \tag{19}$$

Note that μ_{cr} is close to the *smoothing factor* of the corresponding relaxation scheme, that is $\overline{\mu} = .5$ [1, §1].

III.2. Habituated CR

We work with the same framework of Appendix III.1. Habituated CR is a two-stage process: a relaxation sweep (denoted by R), followed by a correction sweep (denoted by S) of setting the coarse variables to zero. Thus, its Fourier symbol $\hat{M}_H(\theta)$ is $\hat{S}(\theta)\hat{R}(\theta)$, where $\hat{R}(\theta)$ is the relaxation symbol and $\hat{S}(\theta)$ is the correction symbol. First, we note that a general grid-h function has the form

$$e_{i_1,i_2} := e^{(i_1\theta_1+i_2\theta_2)} \times \begin{cases} A, & i_1 \text{ even}, i_2 \text{ even}; \\ B, & i_1 \text{ odd}, i_2 \text{ even}; \\ C, & i_1 \text{ even}, i_2 \text{ odd}; \\ 0, & i_1 \text{ odd}, i_2 \text{ odd}, \end{cases} \qquad \forall i_1, i_2 \in \mathbb{Z} \tag{20}$$

and can be expressed as a linear combination of four Fourier modes,

$$e_{i1,i2} = \hat{A}e^{(i_1\theta_1+i_2\theta_2)} + \hat{B}e^{(i_1\theta_1+i_2(\theta_2+\pi))} + \hat{C}e^{(i_1(\theta_1+\pi)+i_2\theta_2)} + \hat{D}e^{(i_1(\theta_1+\pi)+i_2(\theta_2+\pi))},$$

where

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = X \begin{pmatrix} \hat{A} \\ \hat{B} \\ \hat{C} \\ \hat{D} \end{pmatrix}, \qquad X := \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Note that $X^{-1} = \frac{1}{4}X$. In [3] it is shown that

$$\hat{R}(\theta) = \text{diag}\left(r(\theta_1,\theta_2), r(\theta_1,\theta_2+\pi), r(\theta_1+\pi,\theta_2), r(\theta_1+\pi,\theta_2+\pi)\right), \tag{21}$$

where

$$r(\theta_1,\theta_2) := \frac{e^{-i\theta_1}+e^{-i\theta_2}}{4-e^{i\theta_1}-e^{i\theta_2}}.$$

The process S transforms a function (20) with coefficients $(A,B,C,D)^T$ to the function (17) with $(A,B,C,0)^T$. Thus, its Fourier symbol is

$$\hat{S}(\theta) = X \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} X^{-1} = \frac{1}{4} \begin{pmatrix} 3 & 1 & 1 & -1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 3 & 1 \\ -1 & 1 & 1 & 3 \end{pmatrix}. \tag{22}$$

By numerical (or analytical) maximization of $\rho(\hat{M}_H(\theta))$ over all θ, we obtain the asymptotic convergence factor for habituated CR (see also Fig. 7). Specifically,

$$\mu_{\text{cr}} := \max_{\theta\in[-\pi,\pi]^2} \rho(\hat{M}_H(\theta)) = \rho(\hat{M}_H(0,0)) = \frac{3+\sqrt{33}}{12} \approx .729. \tag{23}$$

Note that habituated CR is slower than concurrent CR for this model problem. This behavior was observed in other problems; indeed, in habituated CR, we "ruin" the values at the coarse variables only to set them back at zero, while concurrent CR keeps them intact throughout the sweep. Still, the Fourier symbols behave similarly versus θ in both CR variants (see Fig. 7).

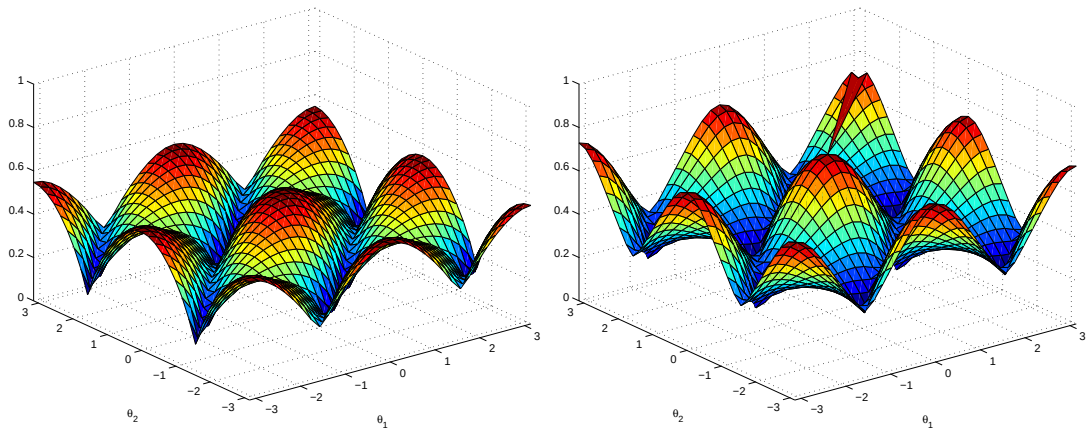

Figure 7. CR Fourier Symbol versus $\theta = (\theta_1, \theta_2)$. Left: concurrent CR. Right:habituated CR.

ACKNOWLEDGEMENTS

The author is grateful to Prof. Achi Brandt from the Weizmann Institute of Science for sharing his innovative ideas and suggestions on the concept and implementation of compatible relaxation, and to Prof. Steve McCormick from the University of Colorado at Boulder for his very fruitful suggestions and comments.

REFERENCES

1. A. Brandt. Multigrid Techniques: 1984 Guide, with Applications to Fluid Dynamics, 191 pages, 1984, ISBN-3-88457-081-1, GMD Studien Nr. 85. Available from GMD-AIW, Postfach 1316, D-53731, St. Augustin 1, Germany, 1984.
2. A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comput.*, 19:23–56, 1986.
3. A. Brandt. Rigorous quantitative analysis of multigrid: I. Constant coefficients two level cycle with L_2 norm. *SIAM J. Numer. Anal.*, 31:1695–1730, 1994.
4. A. Brandt. General highly accurate algebraic coarsening schemes. *Electr. Trans. Num. Anal.*, 10:1–20, 2000.
5. A. Brandt. Multiscale Scientific Computation: Review 2001. In, T. Barth, R. Haimes and T. Chan, eds.: *Multiscale and Multiresolution Methods*, Springer Verlag, 2001. (Proceeding of the Yosemite Educational Symposium, October 2000).
6. 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.
7. A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and its Applications* (Evans, D. J., ed.), Cambridge University Press, Cambridge, 1984, pp. 257–284.

8. A. Brandt and D. Ron. Multigrid solvers and multilevel optimization strategies. In *Multilevel Optimization and VLSICAD* (J. Cong and J. R. Shinnerl, eds.), Kluwer Academic Publishers, Boston, 2002, pp. 1–69 (in preparation). 9
9. M. Brezina, A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, and J.W. Ruge. Algebraic multigrid based on element interpolation (AMGe). LLNL Technical Report UCRL-JC-131752, submitted to *SIAM J. Sci. Comput.*
10. M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, Adaptive smoothed aggregation (aSA), *SIAM J. Sci. Comp.* (submitted).
11. B. Diskin, Multigrid algorithm with conditional coarsening for the non-aligned sonic flow, *Elect. Trans. Numer. Anal.*, 6:106–119, 1997.
12. R. Falgout and P. Vassilievski, On generalizing the framework of AMG (in preparation).
13. O. Livne and G. H. Golub. Scaling by binormalization (in preparation).
14. J. Ruge, and K. Stüben. Algebraic multigrid. In *Multigrid Methods* (McCormick, S. F., ed.), SIAM, Philadelphia, 1987, pp. 73–130.
15. K. Stüben. Algebraic multigrid (AMG): An introduction with applications. Guest appendix in [16].
16. U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, 2000.