COMPATIBLE RELAXATION BASED GEOMETRIC-ALGEBRAIC MULTIGRID

FEI CAO

Abstract. We develop compatible relaxation algorithms for smoothed aggregation-based multigrid coarsening. In the proposed method, we use the geometry of the given discrete problem on the finest level to coarsen the system together with compatible relaxation to form the sparsity structure of the interpolation operator and then apply energy minimization techniques to compute its entries. Of particular interest in this work is the idea to use a geometric coarsening algorithm based on a new more sharp variant of compatible relaxation. The proposed method is competitive with classical AMG in terms of its convergence properties for scalar anisotropic diffusion problems and allows explicit control of operator complexities. We present preliminary numerical experiments for a two-level scheme that demonstrate the potential of the proposed algorithm.

1. Introduction

We consider developing geometric-algebraic multigrid (GAMG) solvers for a given linear system of algebraic equations

$$A\mathbf{u} = \mathbf{f}, \tag{1.1}$$

coming from various discretizations of the scalar elliptic diffusion problem

$$\begin{cases} -\nabla \cdot (a(x)\nabla u) = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{1.2}$$

where $\Omega \in \mathbb{R}^d$. We consider the case $d = 2$ and $\Omega = [0,1] \times [0,1]$ and assume that the coefficient $a(x)$ has a jump discontinuity of several orders in magnitude. The discrete solution and right side satisfy $\mathbf{u}, \mathbf{f} \in \mathbb{R}^{\mathbf{n_c}}$ and $A \in \mathbb{R}^{n_c \times n_c}$ is a symmetric and positive definite matrix. Multigrid solvers for solving (1.1) involve a smoother, with error transfer operator given by $I - M^{-1}A$, and a coarse-level correction, with error transfer operator given by $I - \Pi_A = I - PA_c^{-1}P^TA$. The resulting two-level method, from which a multilevel method is defined recursively, reads

$$E_{TG} = (I - \Pi_A)(I - M^{-1}A). \tag{1.3}$$

Typically, the smoother M is fixed and then the prolongation operator P is constructed in an automated setup algorithm to complement the smoothing process. The setup algorithm constructs the interpolation matrix P and then computes the coarse level matrix using the Galerkin definition $A_c = P^TAP$. Hence, the main task in AMG is to construct a stable interpolation operator P such that a certain *approximation property* holds and both P and A_c are *sparse* matrices. Numerous algorithms have been developed for constructing matrix-dependent interpolation, going back to the original classical AMG algorithm [3, 4]. In this work, we construct the coarse grids using the geometry of the finest-grid discretization as in [7] and compute P and $A_c = P^TAP$ using compatible relaxation [2, 5] and energy minimizing algebraic multigrid [12, 15, 17, 6, 13].

In terms of developing multigrid methods for multicore systems there are two main issues that must additionally be considered. First, when the size of the problem decreases on coarse levels a large number of processors become idle. This issue is in general unavoidable since both geometric and algebraic multigrid methods achieve their optimality by treating global components of the error on increasingly coarser levels. Various techniques have been proposed to deal with this

issue, including using asynchronous multigrid methods [10] and multilevel domain decomposition methods [1] in which each processor owns the given problem on a local subdomain on the finest level and then owns an increasingly larger part of the fine domain on coarser levels.

The second issue concerns the communication required on coarse levels to compute residuals and coarse level corrections, which in turn is directly related to the fill in of the coarse level operators $A_c = P^T AP$ in an algebraic multigrid method. Our work focuses on treating this latter issue by using the geometry of the given problem and its discretization to coarsen the system together with compatible relaxation to form the sparsity structure of the interpolation operator and energy minimization to compute its entries.

2. Notation and preliminaries

The following identity which specifies the convergence rate of the two-level method is used in designing algorithms for constructing AMG interpolation

$$\|E_{TG}(P)\|_A^2 = 1 - \frac{1}{\sup_{\mathbf{v}} \kappa(\mathbf{v})}, \quad \kappa(\mathbf{v}) = \frac{\|(I-\Pi_*)\mathbf{v}\|_*^2}{\|\mathbf{v}\|_A^2}, \tag{2.1}$$

where Π_* is the $(\cdot,\cdot)_* = (\cdot,\cdot)_{\widetilde{M}}$ orthogonal projection on range(P), with $\widetilde{M} = M + M^T - A$ denoting the symmetrized smoother. The design of AMG methods is motivated by an approximation property of the coarse space, which given the prolongation operator P is defined as

$$\frac{\|(I-\Pi_X)\mathbf{v}\|_X^2}{\|\mathbf{v}\|_A^2} \leq \eta \ \forall \mathbf{v} \in \mathbb{R}^{n_c}, \tag{2.2}$$

where Π_X is the X-orthogonal projection on range(P), with X a suitably chosen symmetric and positive definite operator. We note that the left side in (2.2) will precisely determine the convergence rate if we choose $X = \widetilde{M}$. If X is not equal but spectrally equivalent to $\widetilde{M}$, namely,

$$\sigma_0^{-1}(X\mathbf{v},\mathbf{v}) \leq (\widetilde{M}\mathbf{v},\mathbf{v}) = \|\mathbf{v}\|_*^2 \leq \sigma_1(X\mathbf{v},\mathbf{v}),$$

then we then have that

$$\|(I-\Pi_X)\|_X \leq \|(I-\Pi_*)\|_X \leq \sigma_0\|(I-\Pi_X)\|_X$$
$$\|(I-\Pi_*)\|_* \leq \|(I-\Pi_X)\|_* \leq \sigma_1\|(I-\Pi_X)\|_X$$

As a consequence, the two-level method is a uniform contraction in $\|\cdot\|_A$-norm *if and only if* η is uniformly bounded for some X which is spectrally equivalent to $\widetilde{M}$.

A more practical situation is obtained when assuming only σ_1 is known which then gives a sufficient condition for uniform convergence. A typical choice for X, which motivates the classical AMG approach, is $X = D$ (the diagonal of A). For many smoothers (for example Gauss-Seidel method for sparse matrices, Richardson method for finite difference equations), we have that $(\widetilde{M}\mathbf{v},\mathbf{v}) \simeq (D\mathbf{v},\mathbf{v})$. More generally, one assumes that $\widetilde{M} = M + M^T - A$ is an SPD matrix, which is a necessary and sufficient condition for the convergence of the smoother. The following measure considered in [8] then gives a more general approximation property involving the smoother

$$\mu(PZ_p,\mathbf{v}) := \frac{\|(I-PZ_p)\mathbf{v}\|_{\widetilde{M}}^2}{\|\mathbf{v}\|_A^2}. \tag{2.3}$$

Here, $Z_p : \mathbb{R}^n \to \mathbb{R}^{n_c}$ and we require that $Z_pP = I$ such that PZ_p is a projection. Note that for fixed P, this gives a bound on the convergence rate of the two-level method, since the measure immediately gives the bound $\kappa(\mathbf{v}) \leq \mu(PZ_p,\mathbf{v})$ all $\mathbf{v}$. That is, if we assume that

$$\mu(PR,\mathbf{v}) \leq K \ \forall \mathbf{v} \in \mathbb{R}^n \setminus \{0\} \tag{2.4}$$

is satisfied for some constant K. Then $K \geq 1$ and

$$\left\|(I - M^{-1}A)(I - \pi_A)\mathbf{v}\right\|_A \leq \left(1 - \frac{1}{K}\right)^{1/2} \|\mathbf{v}\|_A \,. \tag{2.5}$$

Hence, if P is constructed so that measure (2.3) is bounded by a constant for all $\mathbf{v} \neq 0$, then the resulting two-level method converges uniformly.

In such setting, sufficient conditions for two-level convergence are obtained by considering the optimal $\min - \max$ solution to (2.3)

$$\mu^\star = \min_P \max_{\mathbf{v}\neq 0} \mu(PZ_p, \mathbf{v}), \tag{2.6}$$

with fixed Z_p. Then, for a given set C, $\mu^\star$ gives an upper bound on K for all possible choices of P. Hence, $\mu^\star$ can be veiwed as a mesaure of the quality of C. Define $Z_s : \mathbb{R}^{n_f} \to \mathbb{R}^n$, where $n_f = n - n_c$, such that $Z_p Z_s = 0$. Note that then any error $\mathbf{v}$ can be written as

$$\mathbf{v} = Z_s \mathbf{v}_f + Z_p^T \mathbf{v}_c, \tag{2.7}$$

for some $\mathbf{v}_f$ and $\mathbf{v}_c$. The smoother needs to be effective on the space Range(Z_s).

Considering the more general measure

$$\mu_X(PZ_p, \mathbf{v}) := \frac{\|(I - PZ_p)\mathbf{v}\|_X}{\|\mathbf{v}\|_A^2}, \tag{2.8}$$

obtained by replacing $\widetilde{M}$ in $\mu(PZ_p, \mathbf{v})$ by any given SPD matrix X it follows that the arg min of μ_X satisfies $P_*^T A Z_s = 0$ and the minimum is

$$\mu_X^\star = \frac{1}{\lambda_{\min}((Z_s^T X Z_s)^{-1}(Z_s^T A Z_s))}. \tag{2.9}$$

Moreover, the arg min of μ_X is given by

$$P_\star = [Z_s \; Z_p^T] \begin{pmatrix} (Z_s^T A Z_s)^{-1} Z_s^T A Z_p^T \\ I \end{pmatrix} = (I - Z_s (Z_s^T A Z_s)^{-1} Z_s^T A) Z_p^T, \tag{2.10}$$

which gives a generalization of the so-called ideal interpolation in classical AMG.

Assuming that the coarse variables have been constructed so that $\mu_X^\star$ is bounded for all $\mathbf{v} \neq 0$, then using a P that satisfies the following stability property also implies convergence of the resulting two-level method

$$\langle APZ_p\mathbf{v},\ PZ_p\mathbf{v}\rangle \leq \eta \langle A\mathbf{v},\ \mathbf{v}\rangle \quad \text{for all } \mathbf{v}, \tag{2.11}$$

where $\eta \geq 1$ is a constant. This more general result is interesting because it allows for various approaches of defining interpolation. Moreover, it separates the tasks of selecting the coarse variables and defining interpolation. We note that (2.11) has been used extensively in the literature [12, 15, 17, 6] to derive various techniques for constructing an energetically stable P. In addition, since $\widetilde{M}$ is SPD, we can apply the result in (2.10) and (2.9) for $X = \widetilde{M}$. Hence, if the generalized approximation property holds for some K and P is stable, then the convergence of the two-level method follows.

If we consider finding the operator P that minimizes $\mu_X^\star$ and let $(\cdot,\cdot)_* = (X\cdot,\cdot)$, then given $n_c = |C|$ and setting $P_* = Z_p^T = \left[\phi_1, \phi_2, \cdots, \phi_{n_c}\right]$, where ϕ_k solves the generalized eigenvalue problem

$$(A\phi_k, \mathbf{w}) = \mu_k(\phi_k, \mathbf{w})_* \quad \forall \mathbf{w} \in \mathbb{R}^n,$$

with $\mu_1 \leq \mu_2 \leq \mu_3 \leq \cdots \leq \mu_{n_c} \leq \cdots \leq \mu_n$ and $(\phi_k, \phi_l)_* = \delta_{kl}$, we have

$$\mu_X^\star = \frac{1}{\mu_{n_H+1}} \,.$$

To see this we set $Z_s = \left[\phi_{n_c+1}, \cdots, \phi_n\right]$. Then, $Z_p Z_s = 0$, $Z_p P = I$ and $Z_s^T X^{-1} A Z_s = \mathrm{diag}(\lambda_{n_c+1}, \cdots, \lambda_n)$. The result follows from the earlier result that

$$\mu_X^\star = \frac{1}{\lambda_{\min}((Z_s^T X Z_s)^{-1}(Z_s^T A Z_s))}.$$

Hence, the P_* that minimizes $\kappa(v)$ also minimizes the measure $\mu_X^\star$. Note that the result suggests that the coarse space must consist of small eigenvalues of $X^{-1}A$.

In the classical AMG formulation of interpolation

$$P = \begin{bmatrix} W \\ I \end{bmatrix} \begin{matrix} \} F \\ \} C \end{matrix},$$

where $C \subset \{1, ..., n\}$ and $F = \{1, ..., n\} \setminus C$, and $W \in \mathbb{R}^{|F|\times|C|}$ defines the interpolation weights. Similarly, consider the splitting such that

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}, \quad Z_p = \begin{bmatrix} 0 & I \end{bmatrix} \quad \text{and} \quad Z_s = \begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{2.12}$$

Then, $W_* = -A_{ff}^{-1} A_{fc}$, which gives the ideal interpolation operator often used in analyzing the classical AMG derivation of interpolation. Moreover, if we minimize $(A\mathbf{v}, \mathbf{v})$ for a given $\mathbf{v}_c$, this is the same as minimizing

$$\min_{\mathbf{v}_f}(A_{ff}\mathbf{v}_f, \mathbf{v}_f) + 2(A_{fc}\mathbf{v}_c, \mathbf{v}_f) + 2(A_{cc}\mathbf{v}_c, \mathbf{v}_c).$$

The last term can be dropped since $\mathbf{v}_c$ is fixed and the minimizer of $\min_{\mathbf{v}_f}(A_{ff}\mathbf{v}_f, \mathbf{v}_f) + 2(\mathbf{f}, \mathbf{v}_f)$, with $f := A_{fc}\mathbf{v}_c$ is $\mathbf{v}_f = A_{ff}^{-1}\mathbf{f}$, or $\mathbf{v}_f = A_{ff}^{-1}A_{fc}\mathbf{v}_c$ and, hence,

$$(S_A \mathbf{w}_c, \mathbf{w}_c) = \inf_{\mathbf{v}:\mathbf{v}_c=\mathbf{w}_c} (A\mathbf{v}, \mathbf{v}),$$

where $\mathbf{v} = \begin{bmatrix} * \\ \mathbf{w}_c \end{bmatrix}$, and $S_A = P_*^T A P = A_{cc} - A_{fc}^t A_{ff}^{-1} A_{cf}$ The latter result then shows that using the ideal interpolation operator also minimizes the constant η in the stability estimate given in (2.11). This observation motivates our choice of using energy minimization techniques to form P [12, 15].

Specifically, given a set of aggregates, $\{\Omega_i\}_{i=1}^{n_c}$ such that $\cup_i \Omega_i = \Omega$, consider

$$\chi \;=\; \{Q \;: Q_{ji} = 0 \;\text{ for all }\; j \notin \Omega_i \text{ and } \; Q\mathbf{1}_c = \mathbf{e}\}. \tag{2.13}$$

Here, $\mathbf{e}$ is an arbitrary nonzero element of $\mathbb{R}^n$. Then, the interpolation that we use in our algorithm is defined as the unique solution of the following constrained minimization problem:

$$P = \arg\min J(Q) := \mathrm{trace}(Q^T A Q), \quad Q \in \chi. \tag{2.14}$$

It is well known that the i-th column of the solution to (2.14) is given by

$$[P]_i = I_i A_i^{-1} I_i^T M_a \mathbf{e}, \quad M_a^{-1} = \sum_{i=1}^{n_c} I_i A_i^{-1} I_i^T, \tag{2.15}$$

where $I_i \in \mathbb{R}^{n\times n_i}$ and $(I_i)_{kl} = \delta_{kl}$ if both k and l are in Ω_i and zero otherwise, and $A_i = I_i^T A I_i$. Typically, the constraint vector $\mathbf{e}$ is chosen as a near kernel component in an attempt to satisfy the above mentioned weak approximation property (2.2).

2.1. **Compatible relaxation.** Compatible relaxation, as defined by Brandt [2], is *a modified relaxation scheme that keeps the coarse-level variables invariant.* From our assumption that $Z_p Z_s = 0$ it follows that we can define the compatible relaxation iteration as

$$\mathbf{v}_{k+1} = (I - (Z_s^T M Z_s)^{-1}(Z_s^T A Z_s))\mathbf{v}_k, \tag{2.16}$$

where the convergence rate of this iteration is related to the measure $\mu^\star$ in (2.6) as follows

$$\mu^\star \leq \frac{\Delta^2}{2-\omega} \cdot \frac{1}{1-\rho_s}. \tag{2.17}$$

Here $\Delta \geq 1$ measures the deviation of M from its symmetric part [8] and

$$\rho_s = \left\| (I - M_s^{-1} A_s) \right\|_{A_s}, \tag{2.18}$$

with $M_s = (Z_s^T M Z_s)$ and $A_s = (Z_s^T A Z_s)$. Note that, although we use ρ to represent the spectral radius of a matrix, the quantity ρ_s is in general only an upper bound for the spectral radius of compatible relaxation; it is equal to the spectral radius when M is symmetric.

If iteration (2.16) is fast to converge, then $\mu^\star$ is bounded, that is, fast convergence of CR implies a coarse variable set of good quality and the existence of a P such that the resulting two-level method is uniformly convergent. One can then estimate the value of ρ_s in (2.18) in practice by running the compatible relaxation iteration in (2.16) and monitoring its convergence. In classical AMG, we have

$$Z_p = \begin{bmatrix} 0 & I \end{bmatrix} \quad \text{and} \quad Z_s = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

and the iteration in (2.16) is just simple F-relaxation that is striaghtforward to compute. We note that though this variant of CR is user-friendly, it has been observed in practice that its spectral radius does not provide an accurate prediction of the convergence rate of the two-level method with ideal interpolation in general [6].

However, when $P = \begin{bmatrix} W \\ I \end{bmatrix}, W = -A_{ff}^{-1} A_{fc}$ and $Z_s = \begin{bmatrix} 0 \\ I \end{bmatrix}$ then it is easy to show that

$$I - P(P^T A P)^{-1} P^T A = Z_S (Z_S^T A Z_S)^{-1} Z_S^T A = \begin{pmatrix} I_f & -W \\ 0 & 0 \end{pmatrix}, \tag{2.19}$$

implying that $\mu^\star$, the optimal $\min - \max$ solution to (2.3), can be accurately estimated in practice if an estimate of A_{ff}^{-1} is available. Moreover, the fast convergence of the F-relaxation form of CR implies that A_{ff} is well conditioned and that A_{ff}^{-1} can be efficiently estimated using a polynomial approximation [6]; in our tests we use the Conjugate Gradient iteration to form an approximation.

As discussed in [9], to use CR in the smoothed aggregation (SA) setting we use the fact that spectral radius of E_{TG} remains unchanged if we replace P by PZ for any nonsingular matrix Z. Thus, CR can be used to estimate $\|E_{TG}\|_A$ by using a post-scaled version of a given interpolation operator, e.g., in the SA setting we consider one can reorder the prolongation operator so that it has the form

$$P = \begin{bmatrix} P_f \\ P_c \end{bmatrix} \tag{2.20}$$

for some invertible matrix P_c. Coarse variables are then represented by columns of P_c^{-1}. But, from the observation above, the interpolation operator

$$\bar{P} = P P_c^{-1} = \begin{bmatrix} W \\ I \end{bmatrix}, \tag{2.21}$$

can be used. Here, the appropriate form for the coarse variables is given as in the classical AMG setting where coarsening based on the CR can be implemented as in [5, 6].

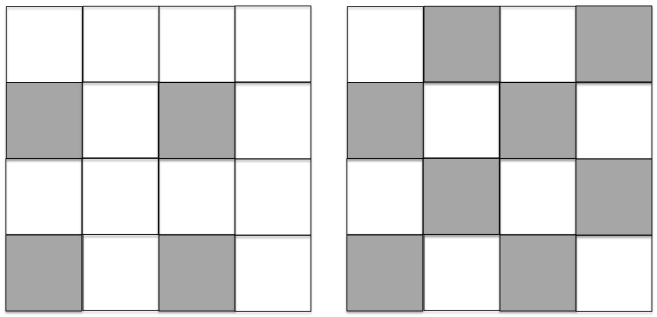

FIGURE 4.1. Distribution of the jump coefficient $a(x)$. Left: Distribution of P1 and P3; Right: Distribution of P2 and P4.

3. Geometric-Algebraic Multigrid Algorithm

The GAMG algorithm we propose uses (mesh dependent) CR based coarsening to determine the number of basis functions required for each aggregate in order to determine the sparsity structure of interpolation together with energy minimization to compute its nonzero entries. The overall algorithm is outlined as follows.

(a) Use the given fine grid to choose a quad tree (or oct-tree in 3d) to define the sets of aggregates on all levels using the approach proposed in [7] and developed for GPUs in [16].
(b) For each level of the resulting multilevel hierarchy initialize $\mathcal{C} = \{\text{seed points of the aggregates}\}$.
(c) While CR convergence is too slow
$$\mathcal{C} = \mathcal{C} \cup \{i : \text{node } i \quad \text{has the largest point-wise error in the aggregate } \Omega_i\}.$$
(d) Multiply the resulting tentative P by the system matrix A to obtain the sparsity pattern, as in smoothed aggregation [14].
(e) Compute the nonzero entries of P using energy minimization [17].

In the above algorithm, the number of basis functions to use in a given aggregate Ω_i is given by the cardinality of $\mathcal{C} \cap \Omega_i$. We note that the standard aggregation algorithm normally uses a *seed point* about which the aggregates are formed. These points make natural first choices for $\mathcal{C}$ in step (b) of the above algorithm.

4. Numerical Results

In our tests, we consider four different test problems defined on structured grids, corresponding to different distributions of the diffusion coefficient $a(x)$ in (1.2). The first two problems, P1 and P3, are ones where the interfaces of the jumps do not intersect, namely,

$$a(x) = \begin{cases} 1 & x \in \Omega_1 \ , \\ 10^{-k_{ij}} & x \in \Omega \setminus \Omega_1 \ , \end{cases} \tag{4.1}$$

where the domain Ω_1 corresponds to the one given by the white regions in the plot on the left in Figure 4.1. For problem P1, $k_{ij} = k \in \mathbb{Z}^+$ for all i, j, and for problem P3 $k_{ij} \in \{1, 2, \ldots, 8\}$, where the values are selected randomly with uniform distribution (using build-in Matlab function `randi`). In the next two problems, P2 and P4, we consider a checkerboard pattern for the distribution of the jumps, where now Ω_1 corresponds to the white regions in the plot on the right in Figure 4.1. For problem P3, the distribution in $\Omega \setminus \Omega_1$ is again uniform with $k_{ij} = k \in \mathbb{Z}^+$ for all i, j and for problem P4 we select the value k_{ij} randomly as in problem P2.

We use a cell-centered finite volume method for discretizing P1-P4 and choose a structured grid $0 = x_0 < x_1 < \cdots < x_{N+1}$, $x_i = \frac{i}{N+1}$, and $0 = y_0 < y_1 < \cdots < y_{N+1}$, $y_j = \frac{j}{N+1}$. Note that

$h = h_x = h_y = \frac{1}{N+1}$. Here, each cell $[x_{i-1}, y_{j-1}] \times [x_i, y_j]$ is used as the control volume and the unknowns are located at each cell center $(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}) = (x_{i-1} + \frac{h}{2}, y_{j-1} + \frac{h}{2})$.

To define $a(x)$ on the interfaces of neighboring subdomains we use a harmonic average, e.g., for an interface S_* we have $a^- \neq a^+$ in general, where $a^- = a(x)$ in the volume on one side of the interface S_* and $a^+ = a(x)$ in the volume on the other side of S_*. We then write the discretized system as

$$a_e u_{i+1,j} + a_w u_{i-1,j} + a_n u_{i,j+1} + a_s u_{i,j-1} - (a_e + a_w + a_n + a_s) u_{i,j} = f_{i,j}, \tag{4.2}$$

where $f_{i,j} = \int_V f \mathrm{d}V$ and V is the control volume $[x_{i-1}, y_{j-1}] \times [x_i, y_j]$ and a_* are harmonic averages of $a(x)$ on the two neighboring cells as in [11]. We assume Dirichlet boundary condition and if an edge S_* is on the boundary of Ω, we set $u_{i+\frac{1}{2},j} = 0$.

Spectral radius of E_{TG} with ideal P

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4
16^2	.259	.255	.297	0.442	.250	.250	.290	.651	.250	.250	.286	.753	.250	.250	.285	.828
32^2	.261	.256	.302	0.491	.251	.251	.303	.703	.250	.250	.293	.890	.250	.250	.290	.943
64^2	.261	.256	.304	0.492	.251	.251	.305	.706	.250	.250	.296	.952	.250	.250	.292	.991
128^2	.261	.256	.305	0.495	.252	.251	.306	.723	.250	.250	.301	.961	.250	.251	.296	.994

Compatible relaxation iteration (2.19)

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4
16^2	.240	.235	.249	.209	.233	.231	.244	.210	.232	.231	.239	.217	.232	.231	.231	.225
32^2	.245	.243	.253	.198	.241	.241	.250	.204	.240	.241	.247	.205	.240	.241	.239	.220
64^2	.244	.242	.252	.200	.239	.239	.250	.205	.239	.239	.247	.216	.239	.239	.237	.225
128^2	.234	.237	.220	.202	.240	.240	.231	.206	.240	.240	.236	.214	.240	.240	.238	.223

TABLE 4.1. Spectral radius of two-grid methods with ideal interpolation (top) and results for compatible relaxation (bottom) applied to Problems P1-P4.

4.1. **Compatible relaxation and aggregation.** We begin our numerical experiments by applying the sharp variant of compatible relaxation from (2.19) to problems P1-P4. We choose regular 2×2 aggregates to coarsen the problem and use standard full-coarsening ($h \to 2h$) to define the coarse variable set $C \subset \Omega$ that are viewed as the seed points for the aggregates. The results of the tests for varying mesh spacing h and different choices of the problem size $N \times N$ and jump discontinuity defined by parameter k are given in the tables in Table 4.1. The table at the top contains approximations of the spectral radius of E_{TG} for the chosen full coarsening together with ideal interpolation. And the bottom table contains estimates of the two-grid convergence rate obtained by running 5 steps of the iteration (2.19) starting with a random initial guess, where the action of A_{ff}^{-1} is approximated by $L = 2$ diagonally preconditioned Conjugate Gradient iterations.

Here we see that the CR converges quickly for all problems and choices of the parameters except for Problem P4. Moreover, it is clear that one can obtain an accurate estimate of $\mu^\star$ using the sharp CR iteration with few inner PCG iterations, again in all cases except for Problem P4. This poor performance for Problem P4 is of course expected since in this case A_{ff} is not well conditioned and the convergence of F-relaxation in this case is $\rho = .983$. Overall, these results suggest that full coarsening is not a good choice for problem P4, that is, it suggests that a single basis function for each 2×2 aggregate is not sufficient to obtain a good solver for Problem P4 in general. We note that if we instead consider red-black coarsening for problem P4, then F-relaxation becomes an exact solve, i.e., the spectral radius of this iteration is zero and A_{ff} is well conditioned. With the same red-black coarsening the sharp CR iteration with 2 inner PCG iterations gives an estimate of the spectral radius of $\rho = .248$ for the case $k = 8$ with $h = 1/2^5$, where the true spectral radius

is .250. We note that the approach gives similar results for other problem sizes and values of k for this problem. Hence, in practice one can run F-relaxation until the iteration converges quickly and then use the sharp variant to obtain a more accurate estimate of $\mu^\star$.

Problem P1

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)
16^2	0.177	5	128	1,026	0.185	5	128	1,026	0.186	5	128	1,026	0.186	5	128	1,026
32^2	0.180	5	512	4,354	0.187	5	512	4,354	0.191	5	512	4,354	0.186	5	512	4,354
64^2	0.186	5	2,048	17,922	0.188	5	2,048	17,922	0.188	5	2,048	17,922	0.188	5	2,048	17,922
128^2	0.186	5	8,192	72,706	0.188	5	8,192	72,706	0.188	5	8,192	72,706	0.189	5	8,192	72,706

Problem P2

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)
16^2	0.182	5	128	1,026	0.183	5	128	998	0.191	2	128	998	0.193	2	128	998
32^2	0.191	5	512	4,354	0.192	4	512	4,354	0.192	2	512	4,294	0.189	2	512	4,294
64^2	0.196	5	2,048	17,922	0.194	4	2,048	17,798	0.193	2	2,048	17,798	0.188	2	2,048	17,798
128^2	0.196	5	8,192	72,706	0.194	4	8,192	72,454	0.194	5	8,192	72,454	0.189	2	8,192	72,454

Problem P3

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)
16^2	0.203	5	128	1,026	0.195	5	128	1,026	0.203	5	128	1,026	0.204	5	128	1,026
32^2	0.200	5	512	4,354	0.205	5	512	4,354	0.206	5	512	4,354	0.204	5	512	4,354
64^2	0.200	5	2,048	17,922	0.203	5	2,048	17,922	0.203	5	2,048	17,922	0.202	5	2,048	17,922
128^2	0.203	5	8,192	72,706	0.205	5	8,192	72,706	0.205	5	8,192	72,706	0.203	5	8,192	72,706

Problem P4

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)	ρ	#It	n_c	nnz(A_c)
16^2	0.201	5	128	1,026	0.226	5	128	1,026	0.216	5	128	1,026	0.218	5	128	1,026
32^2	0.210	5	512	4,354	0.229	5	512	4,354	0.236	5	512	4,354	0.234	5	512	4,354
64^2	0.215	5	2,048	17,922	0.230	5	2,048	17,922	0.250	5	2,048	17,922	0.240	5	2,048	17,922
128^2	0.217	5	8,192	72,706	0.229	5	8,192	72,706	0.252	5	8,192	72,706	0.251	5	8,192	72,706

TABLE 4.2. Two level results for classical AMG applied to Problems P1-P4

4.2. **GAMG results.** In Table 4.2, we present two-level results of classical AMG applied to problems P1-P4 for sake of comparison. The classical AMG algorithm is applied as a black box method with strength of connection parameter $\theta = .25$. We report the problem size on the fine-mesh, the spectral radius of the two-level method, ρ, the number of two-grid PCG iterations needed to reduce the residual by eight orders of magnitue, the size of the coarse level problem, n_c, and the number of nonzero entries in the coarse-level matrix, $nnz(A_c)$ for $k = 1, 2, 4, 8$. We note that classical AMG performs well for all problems, including P4. The good performance of the method for P4 suggests that full coarsening is not a good choice for this problem, since the two-grid method with full coarsening and ideal interpolation performs poorly in this case as reported in Table 4.1.

Next, in Table 4.3 we report results for the algorithm outlined in Subsection 3, where we use regular 2×2 aggregates to coarsen the problem and a single basis function for each aggregate to define a tentative interpolation operator and then apply one smoothing step to form the sparsity structure of P. The entries in P are then computed using the energy minimization (2.15), where the constraint vector is computed by applying 15 Gauss Seidel sweeps to $A\mathbf{e} = 0$ starting with an initial guess as the constant vector.

The approach performs well for all tests except for Problem P4, where the performance is just slightly better than the results obtained using classical AMG with full coarsening and ideal interpolation. It is clear that this improved performance of the GAMG method over classical AMG with

ideal interpolation is related to the more general form of interpolation used in smoothed aggregation, but we are still investigating this behavior in order to obtain a complete understanding of this issue. We note that the method outperforms classical AMG (with better convergence rates and lower grid and operator complexities) in all cases except for Problem P4. We note in addition that the GAMG solver results in 12 nonzero entries per row while classical AMG has at most 9 nonzero entries per row. Though the GAMG algorithm results in additional fill-in in the coarse-level matrix, the sparsity is set explicitly and, hence, known a priori.

Again, the poor performance of the GAMG method for P4 suggests that strength of connection may be needed to adjust the support of the basis functions given on each aggregate and/or add additional basis functions may be required in order to obtain effective interpolation. The latter observation is further supported by the CR results from the previous section, which indicate that two basis functions are needed for this problem when using regular 2×2 aggregates.

Problem P1

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)
16^2	0.119	5	64	762	0.111	5	64	762	0.122	5	64	762	0.121	5	64	762
32^2	0.122	5	256	3,012	0.119	5	256	3,012	0.138	5	512	3,012	0.140	4	256	3,012
64^2	0.149	5	1,024	12,676	0.152	5	1,024	12,676	0.151	5	1,024	12,676	0.155	5	1,024	12,676
128^2	0.151	5	4,096	40,626	0.155	5	4,096	40,626	0.161	5	4,096	40,626	0.161	5	4,096	40,626

Problem P2

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)
16^2	0.108	5	64	762	0.107	5	64	762	0.117	5	64	762	0.119	5	64	762
32^2	0.113	5	256	3,012	0.117	5	256	3,012	0.121	5	512	3,012	0.130	4	256	3,012
64^2	0.119	5	1,024	12,676	0.122	5	1,024	12,676	0.127	5	1,024	12,676	0.138	5	1,024	12,676
128^2	0.120	5	4,096	40,626	0.121	5	4,096	40,626	0.140	5	4,096	40,626	0.140	5	4,096	40,626

Problem P3

	$k=1$				$k=2$				$k=4$				$k=8$			
Size	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)
16^2	0.081	3	64	762	0.090	3	64	762	0.103	4	64	762	0.105	4	64	762
32^2	0.111	4	256	3,012	0.105	4	256	3,012	0.111	4	512	3,012	0.112	4	256	3,012
64^2	0.118	4	1,024	12,676	0.117	4	1,024	12,676	0.118	4	1,024	12,676	0.123	4	1,024	12,676
128^2	0.118	4	4,096	40,626	0.119	4	4,096	40,626	0.120	4	4,096	40,626	0.124	4	4,096	40,626

Problem P4

	$k=1$				$k=2$				$k=4,6$				$k=8$			
Size	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)	ρ	#It	n_H	nnz(A_H)
16^2	0.216	6	64	762	0.322	6	64	762	0.518	8	64	762	0.543	8	64	762
32^2	0.210	6	256	3,012	0.355	7	256	3,012	0.550	9	256	3,012	0.838	13	256	3,012
64^2	0.215	6	1,024	12,676	0.361	7	1,024	12,676	0.578	9	1,024	12,676	0.902	18	1,024	12,676
128^2	0.216	6	4,096	40,626	0.360	7	4,096	40,626	0.580	9	4,096	40,626	0.931	20	4,096	40,626

TABLE 4.3. Results of GAMG algorithm applied to Problems P1-P4.

5. Conclusion

We proposed a GAMG algorithm for constructing an energy minimization form of interpolation that is suitable for multilevel methods for solvering scalar diffusion problems with discontinuous coefficients. The method uses the fine mesh to construct the aggregates and a new sharp variant of compatible relaxation to determine the number of basis functions needed per aggregate to obtain effective interpolation. We demonstrated the components of the algorithm in a simplified two-grid setting on structured meshes. Our current research focuses on extending the method to handle general unstructured meshes and anisotropic problems which in turn requires deriving a

more general form of interpolation that uses in addition strength of connection and adaptive and bootstrap multigrid techniques.

ACKNOWLEDGEMENTS

We thank James Brannick, Rob Falgout and Xiaozhe Hu for their comments and suggestions on the proposed method. We would also like to acknowledge that the sharp compatible relaxation method was originally brought to our attention by Falgout.

References

[1] Brandt A. and Diskin B. Multigrid solvers on decomposed domains. *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, 157:135–155, 1994.
[2] A. Brandt. General highly accurate algebraic coarsening. *ETNA*, 10:1–20, 2000. Special issue on the Ninth Copper Mountain Conference on Multilevel Methods.
[3] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations. Technical report, Colorado State University, Fort Collins, Colorado, 1983.
[4] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and its applications (Loughborough, 1983)*, pages 257–284. Cambridge Univ. Press, Cambridge, 1985.
[5] J. Brannick and R. Falgout. Compatible relaxation and coarsening in algebraic multigrid. *SIAM J. Sci. Comp.*, 32(3):1393–1416, 2010.
[6] J. Brannick and L.Zikatanov. Algebraic multigrid methods based on compatible relaxation and energy minimization. In *Proceedings of the 16th International Conference on Domain Decomposition Methods*, volume 55 of *Lecture Notes in Computational Science and Engineering*, pages 15–26. Springer, 2007.
[7] Marian Brezina, Petr Vank, and Panayot S. Vassilevski. An improved convergence analysis of smoothed aggregation algebraic multigrid. *Numerical Linear Algebra with Applications*, 19(3):441–469, 2012.
[8] R. D. Falgout and P. S. Vassilevski. On generalizing the AMG framework. *SIAM J. Numer. Anal.*, 42(4):1669–1693, 2004. UCRL-JC-150807.
[9] Robert D. Falgout, Parallel Multigrid, Algebraic Multigrid, and Adaptive Amg. Smoothed aggregation amg theory and compatible relaxation amg for electromagnetic problems, 2010.
[10] Barry Lee, Stephen F. McCormick, Bobby Philip, and linebreak DanielJ. Quinlan. Asynchronous fast adaptive composite-grid methods: Numerical results. *SIAM J. Sci. Comput.*, 25(2):682–700, February 2003.
[11] C. Liu, Z. Liu, and S. McCormick. An efficient multigrid scheme for elliptic equations with discontinuous coefficients. *Communications in Applied Numerical Methods*, 8:621–631, 1992.
[12] Jan Mandel, Marian Brezina, and Petr Vanek. Energy optimization of algebraic multigrid bases. 1998. CU Denver Technical Report.
[13] J. B. Schroder. Smoothed aggregation solvers for anisotropic diffusion. *Numer. Linear Algebra Appl.*, 19:296–312, 2012.
[14] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.
[15] W. L. Wan, Tony F. Chan, and Barry Smith. An energy-minimizing interpolation for robust multigrid methods. *SIAM J. Sci. Comp.*, 21(4):16321649, 2000.
[16] Lu Wang, Xiaozhe Hu, Jonathan Cohen, and Jinchao Xu. A parallel auxiliary grid algebraic multigrid method for graphic processing units. *SIAM J. Sci. Comput.*, 35(3):263283, 2013.
[17] Jinchao Xu and Ludmil Zikatanov. On an energy minimizing basis for algebraic multigrid methods. *Comput. Vis. Sci.*, 7(3-4):121–127, 2004.