SPECTRAL AMGE (ρAMGe)

TIMOTHY P. CHARTIER*

Abstract. Spectral AMGe (ρAMGe), is a new algebraic multigrid method for solving discretizations that arise in Ritz-type finite element methods for partial differential equations. The method assumes access to the element stiffness matrices in order to lessen certain presumptions that can limit other algebraic methods. ρAMGe uses the spectral decomposition of small collections of element stiffness matrices to determine local representations of algebraically "smooth" error components. This decomposition provides the basis for generating a coarse grid and for defining effective interpolation. This paper presents a theoretical foundation for ρAMGe along with numerical results that demonstrate the efficiency and robustness of the method.

1. Introduction. Computational simulation drives many branches of today's scientific research. Modern applications require large-scale computing which in turn demand efficient iterative algorithms. Current interest in algebraic multigrid methods grows in the academic and industrial sectors by virtue of its potential for solving such large-scale problems.

Spectral AMGe (ρAMGe) is a new algebraic multigrid algorithm that presumes very little about the nature of smooth error components. All current algebraic methods rely somewhere in their construct on a premise of smoothness. Typically, the setup phase of an algebraic multigrid process presumes either that such smooth errors vary slowly along "strong" connections or that some (near) null space components are given. The aim of ρAMGe is to avoid any such premise in an attempt to broaden its base of applicability. This paper presents a theoretical foundation for ρAMGe as well as numerical results that demonstrate the robustness of the method.

2. AMGe methods and the AMGe measure. Algebraic multigrid methods seek the solution, $\mathbf{u} \in \Re^n$, to the linear system

$$A\mathbf{u} = \mathbf{f}, \tag{2.1}$$

where A is a $n \times n$ symmetric positive definite (SPD) matrix. The SPD requirement is not very strict in that minor modifications to the algorithm should allow it to apply to problems that depart at least somewhat from this case. Yet, the SPD assumption simplifies discussion.

AMGe methods also assume access to the set of finite element stiffness matrices (and therefore the set of associated finite elements, $\mathcal{E}$) that arise in Ritz-type finite element methods for partial differential equations. Specifically, AMGe assumes access to a decomposition $A = \sum_{\alpha \in \mathcal{E}} A_\alpha$, where for each element α, A_α is symmetric nonnegative definite. AMGe methods do not, however, assume the ability to create new finite element stiffness matrices that might be used for coarser levels, for example.

The foundation of AMGe methods is built upon the following heuristic:

H: *Interpolation must be able to approximate an eigenvector with error proportional to the size of the associated eigenvalue.*

Most algebraic multigrid solvers rely on the usual AMG ideas of strength of connection determined by relative sizes of matrix entries, or an implicit assumption that the constant function is smooth. This heuristic correlates smooth error only to the eigenvectors of A. The goal of AMGe methods is to use this more generalized sense of smoothness to build effective components for coarse-grid correction.

To this end, AMGe uses the element stiffness matrices and a measure derived from multigrid theory to determine local representations of algebraically "smooth" error components. Exploiting the information available from this measure results in effective interpolation as described in [3].

To make heuristic **H** more precise, define the index set $\mathcal{P} := \{1, 2, \dots, n\}$. We refer to the elements, p, of $\mathcal{P}$ as *grid points* so that the coordinates in $\Re^n$ represent the *fine-grid variables*. Similarly, if $n_c < n$ is the number of *coarse-grid points*, then the coordinates in $\Re^{n_c}$ represent the *coarse-grid variables*.

*Department of Mathematics, University of Washington, P.O. Box 354350 Seattle, Washington 98195. *email*:chartier@math.washington.edu

Now let $P : \Re^{n_c} \to \Re^n$ denote the *interpolation operator* (that we have not yet chosen). Define $Q : \Re^n \to \Re^n$ to be any convenient linear projection operator onto Range(P):

$$Q = PR,$$

for some restriction operator $R : \Re^n \to \Re^{n_c}$ for which $RP = I_c$, the identity on $\Re^{n_c}$. Note that for any vector $\mathbf{e} \in$ Range(P), we have $Q\mathbf{e} = \mathbf{e}$. Thus, $I - Q$ can be used to measure the defect of interpolation for any Q designed for our multigrid cycle. We now define a measure of how well heuristic **H** is satisfied:

$$M(Q, \mathbf{e}) := \frac{\langle (I-Q)\mathbf{e}, (I-Q)\mathbf{e}\rangle \, \|A\|}{\langle A\mathbf{e}, \mathbf{e}\rangle}, \tag{2.2}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product on $\Re^n \times \Re^n$ and $\| \cdot \|$ denotes the Euclidean vector or vector-induced matrix-norm. (Any other practical inner product could be used here provided A remains self-adjoint with respect to it.) We call (2.2) the *measure of interpolation quality*.

M should have a small bound for all $\mathbf{e} \in \Re^n \setminus \{0\}$ and for any associated discretization parameters. A relation between a bound on M and convergence was proved in [3]: if $M \le K$ for all $\mathbf{e}$, then the two-level convergence factor is bounded by

$$\kappa \le \sqrt{1 - \frac{1}{K}}. \tag{2.3}$$

Therefore, the smaller the bound on M, the better the convergence bound.

Note that AMGe's generalized sense of smoothness results in the freedom to assume for convenience that matrix A is diagonally scaled, so that its diagonal part is the identity matrix. We will make this assumption for the remainder of the paper.

3. ρAMGe. The motivation for ρAMGe comes from an attempt to choose, in some sense, the best subspace to serve as the coarse-grid space. Consider measure $M(Q, \mathbf{e})$ in (2.2), where Q is the best projection (i.e., the Euclidean projection) onto Range$(P) \subset \Re^n$. Ideally, we would choose Range$(P) \subset \Re^n$ of dimension n_c so that $\sup_{e \neq 0} M(Q, \mathbf{e})$ is minimized. It is easy to see then that the best choice for Range(P) is the subspace of eigenvectors of A belonging to its n_c smallest eigenvalues. Of course, it would be prohibitively expensive to do this globally. ρAMGe is an attempt to achieve this objective by choosing local eigenvectors, in analogy to the spectral element method for discretizing partial differential equations.

The notion of an *agglomerate*, defined simply as a set of elements, is central to ρAMGe. If τ is an agglomerate, then we extend our notation so that A_τ is the sum of the associated element stiffness matrices. Note that A_τ is of dimension $n \times n$, the same as A, so that it is filled with zeros except in the block of rows and columns corresponding to points $p \in \tau$. We also need the local version of this matrix, so we define a "localization" operator, $Loc(\cdot)$, that takes its matrix operand and discards all of its zero rows and columns that correspond to points not in the agglomerate. Then, for convenience, we denote $Loc(A_\tau)$ by A^τ (with τ raised to a superscript) and refer to it as a *local* agglomerate stiffness matrix for emphasis.

The basic idea of ρAMGe is to partition the set of elements into agglomerates, then to assemble the local stiffness matrices corresponding to these agglomerates. Eigenvectors of these local agglomerate matrices corresponding to the small eigenvalues are then used directly (with some modifications noted below) as columns of the interpolation matrix.

We should note that the ideas behind ρAMGe can be generalized by minimizing a Rayleigh quotient that takes relaxation into account. In the following, we do this for Richardson because the error propagation matrix is $I - sA$ and we minimize $\langle Ax, x\rangle / \langle x, x\rangle$. In general, we could minimize $\langle Bx, x\rangle / \langle x, x\rangle$ when the error propagation matrix is $I - B$ and B is symmetric, or other choices when B is nonsymmetric.

3.1. A two-level algorithm. Let $\mathcal{G}$ be a partition of element space $\mathcal{E}$ into agglomerates τ of the elements that created A (e.g., continuous piecewise bilinear functions on squares elements). For the moment, assume that $\mathcal{G}$ has been decided on the fine level.

To implement this algorithm recursively, we need to view this structure algebraically. To this end, consider a point, $p \in \mathcal{P}$, that is in agglomerate $\tau \in \mathcal{G}$. Then we say that $p \in \tau^0$ (the interior of τ) if p is in no other agglomerate; otherwise, $p \in \partial\tau$ (the boundary of τ).

The following two-level algorithm determines interpolation operator P as a mapping from the coarse degrees of freedom determined within each agglomerate to the fine-degrees of freedom corresponding to $p \in \mathcal{P}$. For this construction, we assume some ordering of the agglomerates in $\mathcal{G}$. For the moment, suppose that the number of coarse degrees of freedom, m_τ, in each agglomerate $\tau \in \mathcal{G}$ has been predetermined. Denote $n_c = \sum_{\tau\in\mathcal{G}} m_\tau$ for later reference.

1. For each $\tau \in \mathcal{G}$, form an intra-agglomerate interpolation operator, P_τ, as follows:
   - Compute the eigenvector decomposition of local agglomerate stiffness matrix A^τ.
   - From the first m_τ eigenvectors extended with zeros outside of τ, form the columns of P_τ corresponding to the coarse degrees of freedom in τ; fill the other columns with zeros.
2. To form the global interpolation operator, P, define the following "partition of unity": for each agglomerate $\tau \in \mathcal{G}$, let $W_\tau = \mathrm{diag}(w_\tau^1, w_\tau^2, ..., w_\tau^n)$ be the $n \times n$ diagonal matrix with entries $w_\tau^p = a_\tau^p / \left(\sum_{\sigma\in\mathcal{G}} a_\sigma^p\right)$, where a_τ^p denotes the diagonal entry of A_τ associated with p. Note that $w_\tau^p = 0$ if $p \notin \tau$ and $w_\tau^p = 1$ if p is in the interior of τ. (While $\sum_{\sigma\in\mathcal{G}} a_\sigma^p = 1$ because of diagonal scaling of A on the finest grid, this more general form is useful later in the multilevel case.) The W_τ have the properties that $\sum_\tau W_\tau = I$ (identity matrix) and $\|W_\tau\| = \|W_\tau\|_\tau \le 1$. Now define the global interpolation operator by
$$P = \sum_\tau W_\tau P_\tau. \tag{3.1}$$
3. Construct the usual Galerkin coarse-grid operator $P^T A P$.

In what follows, we continue to focus on *matrices*, *vectors*, and *Euclidean space*. Thus, $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ represent the Euclidean inner product and norm, respectively, on the global vector space $\Re^n$, and $\langle\cdot,\cdot\rangle_\tau$ and $\|\cdot\|_\tau$ represent the local Euclidean inner product and norm, respectively, on the local vectors of node values for agglomerate τ. (When applied to a vector in $\Re^n$, we take $\langle\cdot,\cdot\rangle_\tau$ and $\|\cdot\|_\tau$ to mean that applied to the restriction of the vector to the agglomerate.)

3.1.1. Analysis of the two-level algorithm. The simplicity of ρAMGe lends itself to a simple theory. Our first theorem shows that the coarse subspace preserves the null space of A. For this theorem to make sense, we temporarily allow A to be singular. What we really want to conclude from this theorem in practice is that the coarse level adequately represents vectors that are in the null space of the operator up to boundary conditions. This property is important for coarsening because such near null space error components usually have very small energy and must therefore be approximated very well by the coarse level.

THEOREM 3.1. *Suppose that coarsening includes a basis for the null space of local agglomerate stiffness matrix A^τ: $m_\tau \ge |\mathrm{Null}(A^\tau)|$ for every agglomerate τ, where $|\cdot|$ denotes cardinality. Then the null space of A is a subset of the range of interpolation: if $A\mathbf{u} = 0$, then there exists a $\mathbf{v} \in \Re^{n_c}$ such that $\mathbf{u} = P\mathbf{v}$.*

Proof. Assume that $A\mathbf{u} = 0$. Then, for every $\tau \in \mathcal{G}$, we must have that $A_\tau \mathbf{u}_\tau = 0$, where $\mathbf{u}_\tau$ is formed from $\mathbf{u}$ by zeroing out its entries outside of τ. Hence, by our assumption on m_τ, there exists a coarse-grid vector, $\mathbf{v}_\tau$, that is also zero outside τ and such that $\mathbf{u}_\tau = P_\tau \mathbf{v}_\tau$. Let $\mathbf{v}$ be the coarse-grid vector that agrees with $\mathbf{v}_\tau$ on τ, for each $\tau \in \mathcal{G}$. Then we have
$$P\mathbf{v} = \sum_\tau W_\tau P_\tau \mathbf{v}_\tau = \sum_\tau W_\tau \mathbf{u}_\tau = \sum_\tau W_\tau \mathbf{u} = I\mathbf{u} = \mathbf{u}.$$
□

Averaging eigenvectors is used to ameliorate the conflict between interpolation formulas at points in the intersection of agglomerates. Theorem 3.1 shows that prolongation nevertheless exactly interpolates components of the null space. We now show that the coarse level approximates all errors in proportion to their energy: $M(Q, \mathbf{e})$ is uniformly bounded when Q is the Euclidean projection. Thus, our objective is now to find a small bound, K, for which

$$\|A\| \inf_{\boldsymbol{\epsilon} \in \text{Range}(P)} \|\mathbf{e} - \boldsymbol{\epsilon}\|^2 \leq K \langle A\mathbf{e}, \mathbf{e} \rangle, \tag{3.2}$$

for any $\mathbf{e} \in \Re^n$. Note, by (2.2)-(2.3), that the two-level convergence factor is then bounded by $\kappa \leq \sqrt{1 - \frac{1}{K}}$.

Remember that A^τ denotes $Loc(A_\tau)$. Let $\lambda_1(A^\tau) \leq \lambda_2(A^\tau) \leq \ldots$ denote its eigenvalues. Finally, let the maximal overlap in $\mathcal{G}$ mean the maximum over $p \in \mathcal{P}$ of the size of agglomerate neighborhood $\mathcal{N}(p) := \{\tau \in \mathcal{G} : p \in \tau\}$.

THEOREM 3.2. *Assume that the discretization is quasi-uniform in the sense that the local agglomerate stiffness matrices satisfy $\|A^\tau\| \geq \eta \|A\|$, for all $\tau \in \mathcal{G}$. Assume also that we have chosen m_τ so well that the local measure is uniformly bounded, meaning that there exists a $\delta > 0$ such that $\|A^\tau\| \leq \delta \lambda_{m_\tau+1}(A^\tau)$, for all $\tau \in \mathcal{G}$, where $\lambda_{m_\tau+1}(A^\tau)$ denotes the $(m_\tau + 1)$th smallest eigenvalue of A^τ. Then, for any vector $\mathbf{e} \in \Re^n$, there exists a global interpolant, $\boldsymbol{\epsilon}$, in the range of P such that*

$$\|A\| \cdot \|\mathbf{e} - \boldsymbol{\epsilon}\|^2 \leq \frac{c\delta}{\eta} < A\mathbf{e}, \mathbf{e} >, \tag{3.3}$$

where, c is a constant that depends only on the maximal overlap in $\mathcal{G}$.

Proof. Let $\mathbf{e} \in \Re^n$ be given. Note that our assumption on m_τ is equivalent to the assumption that, for any agglomerate $\tau \in \mathcal{G}$, there exists an $\boldsymbol{\epsilon}_\tau$ in the range of P_τ such that

$$\|A_\tau\| \cdot \|\mathbf{e}_\tau - \boldsymbol{\epsilon}_\tau\|_\tau^2 \leq \delta \langle A_\tau \mathbf{e}_\tau, \mathbf{e}_\tau \rangle_\tau, \tag{3.4}$$

where $\mathbf{e}_\tau$ is formed from $\mathbf{e}$ by zeroing out its entries outside of τ. We now construct an $\boldsymbol{\epsilon}$ in the range of P to satisfy (3.3) in a way that corresponds to the construction of P itself: $\boldsymbol{\epsilon} = \sum_\tau W_\tau \boldsymbol{\epsilon}_\tau$. To show that $\mathbf{e}$ satisfies (3.3), we now use properties of W_τ to conclude that

$$\begin{aligned}
\|\mathbf{e} - \boldsymbol{\epsilon}\|^2 &= \| \sum_{\tau \in \mathcal{G}} W_\tau (\mathbf{e} - \boldsymbol{\epsilon})\|^2 = \| \sum_{\tau \in \mathcal{G}} W_\tau (\mathbf{e} - \boldsymbol{\epsilon}_\tau)\|^2 \leq c \sum_{\tau \in \mathcal{G}} \|W_\tau (\mathbf{e} - \boldsymbol{\epsilon})\|^2 = c \sum_{\tau \in \mathcal{G}} \|W_\tau (\mathbf{e}_\tau - \boldsymbol{\epsilon}_\tau)\|_\tau^2 \\
&\leq c \sum_{\tau \in \mathcal{G}} \|\mathbf{e}_\tau - \boldsymbol{\epsilon}_\tau\|_\tau^2 \leq c\delta \sum_{\tau \in \mathcal{G}} \frac{\langle A_\tau \mathbf{e}_\tau, \mathbf{e}_\tau \rangle_\tau}{\|A_\tau\|} \leq \frac{c\delta}{\eta \|A\|} \sum_{\tau \in \mathcal{G}} \langle A_\tau \mathbf{e}_\tau, \mathbf{e}_\tau \rangle \\
&= \frac{c\delta}{\eta \|A\|} \langle A\mathbf{e}, \mathbf{e} \rangle.
\end{aligned}$$

□

3.1.2. Choosing m_τ. Our two-level ρAMGe algorithm will use an *accuracy/cost measure* to select the value of m_τ. The measure considers local qualities that are assumed to represent the global characteristics of the algorithm.

Accuracy. The first component of the measure addresses accuracy. We take the convenient premise here that coarse-grid correction completely eliminates error in the direction of the first m_τ eigenvectors, and that the remaining eigenvectors are reduced by one Richardson iteration with optimal parameter ω. The error propagation matrix in this case is given in terms of the local agglomerate stiffness matrix by $I - \omega A^\tau$. This corresponds to the optimal reduction factor of

$$1 - \omega \lambda_{m_\tau+1} = 1 - \frac{2\lambda_{m_\tau+1}}{\lambda_{n_\tau} + \lambda_{m_\tau+1}} = \frac{\lambda_{n_\tau} - \lambda_{m_\tau+1}}{\lambda_{n_\tau} + \lambda_{m_\tau+1}}. \tag{3.5}$$

Cost using operator complexity estimate. An estimate of cost, specifically operator complexity, constitutes the second component of the accuracy/cost measure. Again, such decisions are considered locally, so a measure must be designed to estimate global cost based on local information. The premise is that the local pattern is replicated throughout the domain.

Let n_e denote the number of elements per agglomerate. Considering a specific agglomerate, τ, an estimate of the number of points belonging to fine-grid element $\tau_f \subset \tau$ is given by $m_{\tau_f} := N_\tau / n_e$. Therefore, a two-level operator complexity growth measure can be defined by

$$two - level\ cost := 1 + c_{op}.$$

Taking multiple levels into account biases further against higher m_τ. A p-level operator complexity growth measure can thus be defined by

$$p - level\ cost := 1 + c_{op} + c_{op}^2 + \ldots + c_{op}^{(p-1)}. \tag{3.6}$$

Refining this estimate to better reflect the true cost without the artificial choice of p is an area of active research.

Accuracy/Growth-Cost. To optimize accuracy (error reduction) versus cost, we can compute an estimate of convergence per work unit (in this case, one fine-grid Richardson iteration) using (3.5) and (3.6):

$$\mu_{op} := accuracy^{1/cost} = \left(\frac{\lambda_{n_\tau} - \lambda_{m_\tau+1}}{\lambda_{n_\tau} + \lambda_{m_\tau+1}} \right)^{1/(1+c_{op}+c_{op}^2+\ldots+(c_{op})^{p-1})}. \tag{3.7}$$

Limited numerical tests showed that using the 3-level operator complexity measure to determine m_τ produced the most robust approach. This two-level ρAMGe algorithm produces promising numerical results (see Section 3.4.)

3.2. A multilevel algorithm. Given the numerical results of the two-level method, recursion is a natural step in the development of a multilevel algorithm. We start by describing several temporary roadblocks that occur after straightforward implementation of such recursion and then discuss needed alterations that form the full ρAMGe method.

3.2.1. Coarse-grid elements and creep. In ρAMGe, local stiffness matrices play a fundamental role in the construction of multigrid components. In the two-level algorithm, only the fine-grid element stiffness matrices are required. However, when a coarse level itself must be coarsened, we need element stiffness matrices for that level to enable a local spectral decomposition there. An efficient multilevel scheme thus requires some coarse-grid version of these local stiffness matrices. A natural first choice is to combine these matrices in some sensible way into agglomerates and then form the coarse-grid element matrix corresponding to each agglomerate τ by computing $P^T A_\tau P$. However, as shown below, problems can arise with this approach, so we describe several potentially more useful alternatives for defining coarse-grid elements. In all cases, construction of the coarse-grid elements starts with some agglomerate of fine-grid elements. We refer to these as *e-agglomerates* to distinguish them from those used to construct the interpolation matrix P, which we refer to as *p-agglomerates*.

Before proceeding, it is instructive to introduce the idea of "creep." This effect appears if, for instance, coarse-level elements are taken as unions of finer-level elements. Then, coarse-level matrices can exhibit spurious null vectors that do not correspond in any way to the global null space. The term "creep" has come to describe such a phenomenon.

Creep is an initial pitfall in extending our two-level ρAMGe algorithm to multiple levels. Since we assume that the null and near null spaces of agglomerate matrices provide a good local approximation to that of the global matrix (and, in fact, are exact for the null space) and that relaxation cannot effectively reduce such components, incorporation of all such local components in interpolation is critical to the success of the method. The presence of creep then corresponds to an unnecessarily large null space in the coarse-grid agglomerate. Since we must choose all local eigenvectors belonging to zero eigenvalues if we want (3.4) to hold, then we must have $m_\tau \geq |\mathrm{Null}(A^\tau)|$. Thus, uncontrolled creep can lead to unacceptable complexity.

The existence of creep is easiest to see from an algebraic argument. Suppose that we have already determined the set, $\mathcal{G}$, of p-agglomerates and defined interpolation P as described in Section 3.1. For any e-agglomerate γ, let $\mathcal{M}_\gamma$ be the set of fine-grid degrees of freedom associated with it and $\mathcal{M}_\gamma^c$ be the set of coarse-grid degrees of freedom that interpolate to $\mathcal{M}_\gamma$. Now, if $|\mathcal{M}_\gamma^c| > |\mathcal{M}_\gamma|$, then there must exist some nonzero coarse-grid function defined over $\mathcal{M}_\gamma^c$ that interpolates to zero over $\mathcal{M}_\gamma$. (It should be noted that this is a sufficient but not a necessary condition for the existence of creep.) This means that if we construct the coarse-grid element matrix corresponding to γ in a straightforward way as $Loc(\sum_{\alpha\in\gamma} P^T A_\alpha P)$, then this matrix will have a larger null space than the local fine-grid matrix $Loc(\sum_{\alpha\in\gamma} A_\alpha)$. A large enough m_τ therefore induces creep, which results in even larger complexity on coarser levels since $|\mathcal{M}_\gamma^c|$ tends to increase with increasing m_τ.

3.2.2. Fuzzy elements and creep. To eliminate the spurious null vectors that can arise in the coarse-grid element associated with the e-agglomerate γ, define the extended e-agglomerate $\bar{\gamma}$ of a given e-agglomerate γ by $\bar{\gamma} := \bigcup_{\tau\in X_\gamma} \tau$ where $X_\gamma := \{\tau \in \mathcal{G} : \mathcal{M}_\gamma \cap \mathcal{M}_\tau \neq \emptyset\}$. Since $\bar{\gamma}$ is itself an agglomerate, $\mathcal{M}_{\bar{\gamma}}$ is the set of fine-grid degrees of freedom corresponding to $\bar{\gamma}$.

Our goal is to start with a set of e-agglomerates that form a partition of the fine-grid elements, and then extend each e-agglomerate, γ, to cover $\bar{\gamma}$. This new *fuzzy agglomerate* contains γ as its *core* and $\bar{\gamma} \setminus \gamma$ as its *fuzz*. For simplicity, where it causes no confusion, we still refer to this fuzzy agglomerate as γ.

We can now form a matrix corresponding to this fuzzy e-agglomerate, where, for generality, we separately weight contributions from its core and associated fuzz. For example, we can form the matrix corresponding to the fuzzy agglomerate with core γ by

$$\hat{A}_\gamma = \alpha_1 A_\gamma + \alpha_2 A_{\bar{\gamma}\setminus\gamma}. \tag{3.8}$$

In practice, we found it useful to take $\alpha_1 = 1.0$ and $\alpha_2 = 0.5$.

One problem here is that, if we just take $P^T \hat{A}_\gamma P$ as the coarse-grid element matrix, it can include many more coarse-grid degrees of freedom than if we had used only the agglomerate matrix corresponding to γ, and there is no guarantee that we have eliminated creep. To avoid this difficulty, we use a local interpolation operator in place of the global operator P for constructing the element matrix. In particular, we construct the local interpolation formula, P_γ^{fuzz}, exactly as described in Section 3.1, but with $\hat{A}_\gamma$ replacing A and X_γ replacing $\mathcal{G}$. Thus, arbitration of interpolation for shared degrees of freedom is performed only over the p-agglomerates in X_γ. Theorem 3.1 guarantees that the null space of $\hat{A}_\gamma$ is interpolated exactly by P_γ^{fuzz}. Thus, the corresponding coarse-grid element matrix, defined by $(P_\gamma^{fuzz})^T \cdot \hat{A}_\gamma \cdot P_\gamma^{fuzz}$, referred to as a *fuzzy element*, maintains the appropriate null space while tending to eradicate creep.

3.3. Reducing the dimension of coarse-grid element matrices. Let γ be an e-agglomerate. We have seen that the coarse-grid matrix corresponding to the resulting element (fuzzy or not) has nonzero rows and columns corresponding to all coarse-grid degrees of freedom associated with each p-agglomerate $\tau \in X_\gamma$. Thus, if there are m coarse-grid degrees of freedom introduced for each such τ, then the size (row dimension) of the coarse-grid element matrix is $m \cdot |X_\gamma|$. Storage of the elements is proportional to the square of this value, while eigenvector/eigenvalue computation of the coarser-grid p-agglomerate matrices can increase roughly with its cube. Thus, reducing this size can have a large payoff in terms of both storage and computation time.

While it seems natural to use the p-agglomerates as the cores of the coarse-grid elements, this almost guarantees that X_γ is as large as possible. An alternative that can dramatically reduce the coarse-grid element size is to stagger the e-agglomerates with respect to the p-agglomerates.

3.3.1. An algorithm for choosing staggered coarse-grid agglomerates. There are many ways to choose staggered coarser-grid elements and agglomerates. Here, we describe the approach we took that seemed the most compatible with the existing ρAMGe code. Assume that we are on level

k and are given level k elements and associated matrices. We start by partitioning the set of level k elements into the level $k+1$ e-agglomerates (element cores). Denote this set by $\mathcal{C}$.

The following algorithm determines the set, $\mathcal{G}$, of p-agglomerates staggered with respect to the element cores, $\gamma \in \mathcal{C}$. In the following, we define $E_i := \{\gamma \in \mathcal{C} : i \in \mathcal{M}_\gamma\}$ and $\mathcal{T}_i := \{\alpha \in \mathcal{E} : i \in \mathcal{M}_\alpha\}$.

1. For each degree of freedom i, initialize the weight $w_i = |E_i|$.
2. If $w_i = 0$, for all i, then exit. Otherwise, pick i with maximal weight to serve as a seed for the staggered p-agglomerate, τ_i.
3. Start the agglomerate by setting $\tau_i = \mathcal{T}_i$ (the element neighborhood of i). Extend τ_i by adding all elements in the interior of E_i not already in another p-agglomerate.
4. Set $w_j = 0, \ \forall\, j \in \mathcal{M}_{\tau_i}$.
5. Let $\mathcal{G} \leftarrow \mathcal{G} \cup \{\tau_i\}$.
6. Go to step 2.

We can now assemble the local stiffness matrices over each p-agglomerate, $\tau \in \mathcal{G}$, and define interpolation as described previously. Then, we can build the fuzzy elements for each e-agglomerate $\gamma \in \mathcal{C}$, and construct the corresponding coarse-grid element matrices corresponding to the elements by determining $(P_\gamma^{fuzz})^T \cdot \hat{A}_\gamma \cdot P_\tau^{fuzz}$ (as in Section 3.2.2). Note that, at this point, the agglomerate information is no longer needed and can be discarded, since it is only used in defining interpolation and determining the element fuzz. It is also possible at this point to discard the element matrices on the current level, since they are no longer required.

As noted in the previous section, staggered agglomerates reduce complexity and limit creep. This can be seen to some extent in the results presented in the following section, although exposing these fully would require more extensive tests.

3.4. Numerical results. We apply the ρAMGe algorithm numerically to two illustrative examples: a Poisson equation and a plane-stress cantilever beam, both discretized with bilinear functions on rectangular elements. The Poisson equation is included for instructive purposes: while many successful AMG approaches exist for solving Poisson problems, important facets of ρAMGe are more readily illustrated within this context.

While the theory assumes Jacobi smoothing, simple Gauss-Seidel with any convenient ordering is also a natural choice for relaxation. The numerical tests throughout this section use a V(1,1) cycle with Gauss-Seidel relaxation.

Let $n_\mathcal{P}^k$ denote the number of degrees of freedom on level k and n_L^k the number of non-zero entries in the level k operator, A^k. Grid complexity is thus given by $C_\mathcal{P} = \left(\sum_{k=1}^m n_\mathcal{P}^k\right)/n_\mathcal{P}^1$ and operator complexity by $C_L = \left(\sum_{k=1}^m n_L^k\right)/n_L^1$.

The asymptotic convergence factor, κ, is estimated by applying 20 V(1,1) cycles to $Au = 0$ with random initial guess u^0 and computing the ratio of the last two residual norms. We diagonally scaled A so that its diagonal part is the identity matrix.

The experiments also use a three-level operator complexity growth measure (μ_{op} in (3.7) with $p = 3$). Additionally, agglomerates were formed using (3.8) with $\alpha_1 = 1.0$ and $\alpha_2 = 0.5$.

The tables display grid and operator complexities as well as estimated asymptotic convergence factors.

3.4.1. Numerical illustration of multilevel algorithm development. While producing impressive two-level results, the ρAMGe algorithm as detailed in Section 3.2 does not extend easily to the multilevel case. This subsection numerically illustrates the evolution of ρAMGe from a two-level to a multilevel algorithm. We consider the numerical revelations that propelled mathematical insight in the research. In particular, we consider the Poisson equation on a square grid discretized with 32×32 bilinear functions on square elements. The boundary conditions are Dirichlet and are eliminated from the matrix during discretization. Standard 2×2 agglomeration is forced on each level.

Creep. As seen in Table 3.1, the effects of creep are clearly evident as the level, k, becomes coarser. Due to the Dirichlet boundary conditions, $|\mathrm{Null}(A^\tau)|$ varies somewhat over τ in the domain. Therefore, Table 3.1 measures $\max_{\tau \in \mathcal{G}^{k-1}}(|\mathrm{Null}(A^\tau)|)$ as an indication of creep in the k-level problem. Note that we are considering $\tau \in \mathcal{G}^{k-1}$ since stiffness matrices are unnecessary in the direct solve on level k. The

	k	κ	$C_{\mathcal{P}}$	C_L	$\lvert\mathrm{Null}(A^\tau)\rvert$
Unstaggered elements	2	0.04	1.47	1.9	1
	3	0.18	1.96	11.1	10
	4	0.19	2.85	90.6	98
Unstaggered fuzzy elements	2	0.04	1.47	1.9	1
	3	0.05	1.60	2.4	1
	4	0.05	1.63	2.5	1
Staggered fuzzy elements	2	0.05	1.53	2.0	1
	3	0.05	1.67	2.2	1
	4	0.05	1.71	2.3	1

Table 3.1

The effects of creep and benefits of staggering elements in ρAMGe with and without the use of fuzzy elements; for Poisson's equation discretized with bilinear functions on square elements on a 32×32 grid; k is the number of levels, κ is the observed convergence factor bound, $C_{\mathcal{P}}$ and C_L are the respective grid and operator complexities, and $\lvert\mathrm{Null}(A^\tau)\rvert$ is the maximum dimension of the agglomerate null spaces.

effect of creep on operator complexity alone is evidence enough for the need of enhancements in the two-level ρAMGe algorithm.

Fuzzy and staggered elements. The tendency of fuzzy elements to eliminate creep is clearly illustrated in Table 3.1. The use of staggered with fuzzy elements is then apparently used effectively to reduce complexity. Such a difference in operator complexity is reflected in Table 3.1 and is even more noticeable for a more computationally difficult problem such as linear elasticity. The staggered approach also involves less work in the creation of the fuzzy elements.

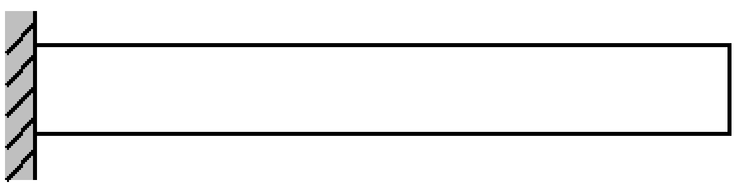

Fig. 3.1. *Plane-stress cantilever beam problem.*

3.4.2. Applying ρAMGe to linear elasticity. Developing more robust methods to solve matrix equations that result from systems of differential equations motivated research into AMGe methods. At the focus of current ρAMGe research is the 2D linear elasticity system

$$\begin{aligned} u_{xx} + \frac{1-\beta}{2}u_{yy} + \frac{1+\beta}{2}v_{xy} &= f_1, \\ \frac{1+\beta}{2}u_{xy} + \frac{1-\beta}{2}v_{xx} + v_{yy} &= f_2, \end{aligned} \tag{3.9}$$

where u and v are displacements in the x and y directions, respectively. Throughout our tests, we use the value $\beta = 1/2$, which yields the Poisson ratio $\nu = \beta/(1+\beta) = 1/3$. The problem, depicted in Figure 3.1, has free boundaries, except on the left where $u = v = 0$. We discretized (3.9) with bilinear finite elements on a uniform $n_x \times n_y$ rectangular array of cells with spacing $h_x \times h_y$. The actual domain will vary in scale ($n_x h_x$ by $n_y h_y$).

3.4.3. Forcing agglomerates. To apply ρAMGe to (3.9), the agglomerates were predetermined for each level. We consider the effects of several such choices.

First, consider a square domain with $n_x = n_y = 32$ partitioned with bilinear functions on square elements (i.e., $h_x = h_y = 1/32$). The algorithm forces 2×2 agglomerates on each level so that level 5

k	κ	$C_{\mathcal{P}}$	C_L
2	0.13	1.52	2.00
3	0.17	1.69	2.32
4	0.17	1.74	2.42
5	0.17	1.77	2.47

TABLE 3.2
Linear elasticity discretized with bilinear functions on square elements ($h_x = h_y = 1/32$) on a 32×32 grid.

contains only one agglomerate, where the coarse-grid problem is directly solved. Table 3.2 shows that promising grid and operator complexities are present throughout the levels.

Size of forced agglomerates	k	κ	$C_{\mathcal{P}}$	C_L
1×2	2	0.36	1.74	3.15
	3	0.41	2.06	3.81
	4	0.42	2.22	4.13
	5	0.43	2.32	4.31
2×2	2	0.42	1.75	3.17
	3	0.41	2.20	5.71
	4	0.42	2.44	7.64
	5	0.42	2.59	9.23
2×1	2	0.22	1.71	3.09
	3	0.86	2.04	3.86
	4	0.89	2.24	4.36
	5	0.94	2.36	4.65

TABLE 3.3
Linear elasticity discretized with stretched rectangular elements ($h_x = 1/32$ and $h_y = 1/(10 * 32)$) on a 32×32 grid.

Table 3.3 uses a rectangular domain where stretched (rectangular) elements with an aspect ratio of $10 : 1$ (i.e., $h_x = 1/32, h_y = 1/(10 * 32)$) discretized the 32×32 grid. Different patterns for agglomeration are forced on each level. Forcing 2×1, 1×2, or 2×2 agglomerates yields acceptable two-level convergence factors, which reflect a certain flexibility toward agglomeration in ρAMGe.

Such flexibility disappears when 2×1 agglomerates are forced for the 3-level test since the 2×1 fine-grid agglomerates produce coarse-grid elements with a relatively large number of associated eigenvalues close to 0. Such an effect on the spectrum of coarse-grid elements does not occur with 1×2 agglomerates, which results in more encouraging convergence factors. Note that the tests confirm standard intuition in that 1×2 agglomeration produces the best results of the three agglomeration patterns in terms of convergence rates and operator complexity.

Next, ρAMGe is applied to a single-element thick 2-D plane-stress cantilever beam. Specifically, linear elasticity equations (3.9) are discretized with bilinear functions on rectangular elements on a $n_x \times 1$ grid, with $n_x = 64$ for the tests. The problem is discretized with square elements ($h_x = h_y = 1/64$) and alternatively with stretched elements possessing a $10 : 1$ aspect ratio (i.e., $h_x = 1/64, h_y = 1/(10 * 64)$. In either case, 2×1 agglomerates are forced on each level.

ρAMGe performs well on the 2-D plane-stress cantilever beam as reflected in Table 3.4. Acceptable results are attained when the domain is partitioned using either square elements or (the more difficult to solve) stretched elements. Such robustness is a notable feature of ρAMGe and bodes well for its potential to efficiently solve a larger family of problems.

3.4.4. Selecting agglomerates algebraically. As suggested by these results, ρAMGe is not very sensitive to the type of agglomeration chosen. Therefore, there is promise that an algebraic

k	Square Elements			Stretched Elements		
	κ	$C_{\mathcal{P}}$	C_L	κ	$C_{\mathcal{P}}$	C_L
2	0.24	1.51	1.51	0.28	1.75	2.12
3	0.25	1.82	1.88	0.29	2.12	2.65
4	0.27	1.99	2.08	0.32	2.37	3.09
5	0.39	2.09	2.19	0.35	2.52	3.37
6	0.43	2.15	2.26	0.39	2.61	3.54

TABLE 3.4

Single-element thick 2-D plane-stress cantilever beam discretized with rectangular elements on a 64×1 grid. Stretched elements use a $10:1$ aspect ratio.

algorithm can be designed to select agglomerates. Yet, Table 3.3 suggests that a robust algorithm should still take the operator into consideration.

Initial tests with such algorithms (including the Jones-Vassilevski algebraic agglomeration algorithm introduced in [9]) matched or were comparable to the results of the previous experiments. This shows promise toward the potential of designing a robust agglomeration routine for ρAMGe. The creation of such an algorithm is an area of active research.

4. Future Work. ρAMGe, as its simple theory suggests, represents a general approach to algebraic multigrid unhindered by too many heuristics. We have explored only a few of the problems that might be effectively solved by ρAMGe, which, with its general sense of smoothness, increases the promise of algebraic multigrid methods to fulfill the needs of modern computational simulation.

Designers of algebraic methods such as ρAMGe must choose from a plethora of options. Undoubtedly, research into ρAMGe methods will continue and result in enhancements to the algorithm. Still, the contents of this paper reflect considerable progress in the development of scalable iterative methods. Specifically, ρAMGe shows promise in fulfilling its underlying motivation of providing an algebraic multigrid solver that broadens the base of applicability of algebraic multigrid methods.

REFERENCES

[1] A. Brandt, *Algebraic multigrid theory: The symmetric case*, in Preliminary Proceedings for the International Multigrid Conference, Copper Mountain, Colorado, April 1983.

[2] A. Brandt, S. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for automatic multigrid solutions with application to geodetic computations*, tech. report, Inst. for Computational Studies, Fort Collins, Colorado, October 1982.

[3] M. Brezina, J. Cleary, R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick, and J. Ruge, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM Journal on Scientific Computing, 22 (2000), pp. 1570–1592.

[4] W. Briggs, V. Henson, and S. McCormick, *A Multigrid tutorial 2nd ed.*, SIAM, Philadelphia, 2000.

[5] T. Chartier, , R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, , and P. Vassilevski, *Spectral agglomeration AMGe*, in preparation, (2001).

[6] T. Chartier, *Element-Based Algebraic Multigrid (AMGe) and Spectral AMGe*, PhD thesis, University of Colorado, 2001.

[7] J. Cleary, R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick, G. Miranda, and J. Ruge, *Robustness and scalability of algebraic multigrid*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1886–1908.

[8] V. E. Henson and P. S. Vassilevski, *Element-free amge: General algorithms for computing interpolation weights in amg*, SIAM Journal on Scientific Computing, 23 (2001), pp. 629–650.

[9] J. Jones and P. Vassilevski, *AMGe based on agglomeration*, SIAM Journal on Scientific Computing, 23 (2001), pp. 109–133.

[10] J. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1987, ch. 4, pp. 73–130.