SPECTRAL AMGE (ρAMGe)

T. CHARTIER*, R. D. FALGOUT†, V. E. HENSON†, J. JONES†,
T. MANTEUFFEL‡, S. MCCORMICK‡, J. RUGE‡, AND P. S. VASSILEVSKI†

Abstract. We introduce spectral AMGe (ρAMGe), a new algebraic multigrid method for solving systems of algebraic equations that arise in Ritz-type finite element discretizations of partial differential equations. The method requires access to the element stiffness matrices, which enables accurate approximation of algebraically "smooth" vectors (i.e., error components that relaxation cannot effectively eliminate). Most other algebraic multigrid methods are based in some manner on predefined concepts of smoothness. Coarse-grid selection and prolongation, for example, are often defined assuming that smooth errors vary slowly in the direction of "strong" connections (relatively large coefficients in the operator matrix). One aim of ρAMGe is to broaden the range of problems to which the method can be successfully applied by avoiding any implicit premise about the nature of the smooth error. ρAMGe uses the spectral decomposition of small collections of element stiffness matrices to determine local representations of algebraically smooth error components. This provides a foundation for generating the coarse level and for defining effective interpolation. This paper presents a theoretical foundation for ρAMGe along with numerical experiments demonstrating its robustness.

1. Introduction. Computational investigation is a vital tool for today's scientists and engineers. Modern computer simulation methods demand increasingly greater speed and accuracy, and are being applied to extraordinarily large systems of equations, with tens or hundreds of millions of unknowns. To solve these systems with the accuracy and speed required, massively parallel computers, and algorithms that effectively exploit their power, are essential. As problems grow larger, algorithms whose performance scales linearly with the problem size are necessary. Unfortunately, many of today's computational simulations use algorithms that do not scale in this sense.

Multigrid methods are scalable for many regular-grid problems. However, they can be extremely difficult to devise for the large unstructured grids that many simulations require. Algebraic Multigrid (AMG) overcomes this difficulty by abstracting, in an algebraic sense, the properties that make geometric multigrid methods effective. Ideally, this results in a method that is automatic and robust. The classical formulation of AMG grew from the efforts of Brandt, McCormick, Ruge, and Stüben in the 1980's. (For details, see [2, 1, 17].) Interest in AMG methods is growing rapidly, both in academia and in industry, because these methods have great potential for solving the large-scale problems common to many modern applications.

AMG has proved to be effective on a large class of problems (see, e.g., [8, 17]), especially scalar elliptic partial differential equations posed on unstructured grids. However, important applications, such as thin-body elasticity posed on unstructured grids, are not well handled by AMG, or, for that matter, most other general iterative methods.

It was shown recently [3, 13, 14] that by using some "extra" information beyond just the operator matrix, AMG methods can be extended effectively to a broader class of problems than previously possible. In particular, information carried in the individual element stiffness matrices enables the so-called element-based algebraic multigrid (AMGe) method to be effective in some cases. However, for certain important applications, such as thin-body elasticity on unstructured grids, robust and efficient solution methods remain problematic.

In this paper we introduce spectral AMGe, denoted ρAMGe (see also [6]), with the premise that no assumptions are to be pre-imposed on the nature of "smooth" error components (i.e., components that cannot be efficiently reduced by relaxation). Many AMG methods are founded, implicitly or overtly, on some pre-defined concept of smoothness. For example, coarse-grid selection and prolongation may be

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

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551. *email*:{rfalgout,vhenson,jjones,panayot}@llnl.gov. This work was performed, in part, under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

‡Applied Math Department, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526. *email*:{tmanteuf, stevem,jruge}@colorado.edu.

defined assuming that smooth errors vary slowly in the direction of "strong" connections; alternatively, it is sometimes presumed that certain vectors in the null-space (or near null-space) of the operator are known and form a basis for the smooth error components. One aim of ρAMGe is to avoid any such implicit or explicit premise. Instead, ρAMGe assumes that the fundamental characteristics of smooth error, for whatever specific operator is at hand, is carried implicitly in the spectra of the individual finite element stiffness matrices. On coarse grids this implies that local stiffness matrices can be constructed for the agglomerated coarse elements. Hence, ρAMGe bases its intergrid transfer operators on the eigenvectors of these local matrices, which are "patched" together in an appropriate way to form the columns of the prolongation operator.

The use of spectral decompositions of local stiffness matrices in an AMG context is not new: the methods developed in [9, 11] form the columns of interpolation from eigenvectors of stiffness matrices corresponding to local aggregates. These methods, extended to indefinite systems in [10] and further expounded in [15], are similar in spirit to the work presented here. Their methods differ, however, in that the aggregates they use do not in general partition the domain, so smoothing of the prolongation operator is required for efficient general multilevel performance. This may be contrasted to the method we present here, in which we compute an adequate interpolation operator directly. Another approach based on ideas similar to those presented here is given in [4]. The primary difference between this latter method and ρAMGe lies in the way the coarse-grid basis functions are constructed and in the way we define the interpolation formulas for the different agglomerates (the "arbitration" we describe in Section 3). Eigenvectors of agglomerate stiffness matrices are also used to correct error between grids in a domain decomposition setting in [16].

As the numerical results of Section 4 show, a primary benefit to our approach is in its robustness. Our research has the goal of broadening the range of problems to which AMG or AMGe can be successfully applied. By basing the method on inherent characteristics of the operator, as determined through the spectra of the element stiffness matrices, we avoid relying on predetermined assumptions on the nature of smooth error. In the next section we begin with a brief discussion of AMGe to motivate and lay the foundation for the development of ρAMGe in Section 3. We conclude with numerical results in Section 4 and additional comments in Section 5.

2. AMGe and the need for ρAMGe.

AMG 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) global stiffness matrix. (Existing AMG algorithms are often applied successfully in more general cases, but we make the SPD assumption here because it simplifies discussion.) In general, AMG methods solve (2.1) using only the information contained in the coefficients of the operator matrix without requiring access to the spatial grid. AMGe, however, uses somewhat more information by requiring knowledge of 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 does not, however, assume that finite elements and associated stiffness matrices exist for coarse-grid levels. Instead, it requires that some algorithm for algebraically constructing coarse-grid elements and stiffness matrices be available (for example, see [14]).

Standard AMG methods generally do not include diagonal scaling of the matrix, because such scaling may reduce the accuracy of interpolation, confuse the definition of strong connections, and, in general, alter the nature of smooth errors. Since common AMG schemes rely for their construction on some premise of how smooth error behaves (e.g., that it is "locally" constant), diagonal scaling can cause significant degradation in performance. However, AMGe makes no predefined characterization of "local" smoothness and diagonal scaling may used without concern. Such scaling simplifies the discussion and methodology. For example, if the matrix is scaled so that its diagonal $D = I$, then Richardson relaxation becomes a good choice for the smoother. (Otherwise, the diagonal entries of A should be taken into account in relaxation, as in the Jacobi and Gauss-Seidel methods.) Hence, for the remainder of the theoretical discussion in this paper, we are free to assume for convenience that matrix A has been scaled so that its diagonal is the identity, and that the relaxation we choose is simple Richardson. For a general SPD matrix with diagonal $D \neq I$, such scaling can be assured by a diagonal scaling that replaces A by $D^{-1/2}AD^{-1/2}$. If diagonal scaling is not used, then in general it would be wise to use a relaxation scheme such as damped Jacobi and adjust the following discussion accordingly.

2.1. The measure of interpolation quality in AMGe. To avoid relying on any assumptions about the specific nature of smooth errors, consider the following basic AMGe heuristic:

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

To make heuristic **H** more precise, define the index set $\mathcal{P} := \{1, 2, \ldots, n\}$. 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.* Let $P : \Re^{n_c} \to \Re^n$ denote an *interpolation operator* (that has not yet been chosen) and let $R : \Re^n \to \Re^{n_c}$ be some *restriction operator.* Define $Q : \Re^n \to \Re^n$ to be any convenient linear projection operator onto Range(P):

$$Q = PR,$$

for which $RP = I_c$, the identity on $\Re^{n_c}$. A specific form for Q (and, hence, R) is given in Section 2.2. Note that for any vector $\mathbf{e} \in \text{Range}(P)$, we have $Q\mathbf{e} = \mathbf{e}$. Thus, $I - Q$ can be used to measure the defect of interpolation for any Q.

A measure that shows how well Heuristic **H** is satisfied and that can be used to define an effective interpolation process [3] is

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

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product on $\Re^n \times \Re^n$ and $\|\cdot\|$ denotes both the Euclidean vector norm and its 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.*

Measure M is appropriate when a Richardson-like relaxation process is used in the multigrid scheme. As we see in Section 3, this leads us to base the coarsening process in ρAMGe on local eigenvectors of A. In general, coarsening should be based on eigenvectors of the error propagation matrix of the relaxation. We opt here to employ Richardson relaxation in the theoretical development because doing so simplifies the discussion. Since measure M can be readily modified to suit most common relaxation methods, developing theory based on other schemes is straightforward.

For **H** to be satisfied, M should have a small bound for all $\mathbf{e} \in \Re^n \setminus \{0\}$. A relation between a bound on M and convergence was established in [3]: if there exists $K \geq 1$ such that $M \leq K$ for all $\mathbf{e}$, then the two-level convergence factor κ is bounded by

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

Hence, the smaller the bound is on M, the better the bound is on the convergence factor. In AMGe measure M essentially *defines* algebraically smooth error, and is used in the construction of the grids and operators, but no *additional* assumptions on the character of smooth errors are made. This should make AMGe more general than many other methods, because it does not restrict itself to cases

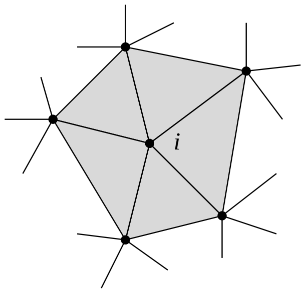

FIG. 2.1. *Shaded elements are members of the element neighborhood $\mathcal{T}_i$; their stiffness matrices are summed to form the neighborhood stiffness matrix B_i. Solid black circles denote nodes belonging to the point neighborhood $\mathcal{V}_i$.*

where algebraically smooth errors are slowly varying along "strong" connections (as classical AMG assumes) or that the smooth errors are represented well by local partitions of supplied near-null-space components (as smooth aggregation methods assume).

Measure M is not explicitly used in ρAMGe in practice, but it provides the essential foundation for the method, as we shall show in Section 3.

2.2. Localizing the measure of interpolation quality. We now indicate how measure M allows us to algebraically define interpolation weights for accurate coarse-grid correction. As a practical matter, we wish to keep the value of M as small as possible. The main difficulty with this aim is that M is a global measure and evaluation of M requires knowledge of Q, which in turn requires knowledge of the coarse grid and interpolation weights. The problem is that we want to use this measure to *construct* components locally for coarse-grid correction, so a global measure is impractical. Straightforward localization of M using information only in A does not provide the robustness we seek.

AMGe instead uses the element stiffness matrices to localize measure M and to determine local representations of algebraically smooth error components. These representations are then used to define the coarse grid and determine a local interpolation operator, which is in turn incorporated into global operator P. Before showing how this is done, we need some notation.

Let $\mathcal{M}_\alpha$ be the set of points belonging to an element α:

$$\mathcal{M}_\alpha := \left\{ j \in \mathcal{P} \ : \ \boldsymbol{\varepsilon}_j^T A_\alpha \boldsymbol{\varepsilon}_j \neq 0 \right\}, \tag{2.4}$$

where $\boldsymbol{\varepsilon}_j$ is the canonical basis vector associated with point $j \in \mathcal{P}$. Next, define the *element neighborhood* of a point $i \in \mathcal{P}$ to be the set of elements

$$\mathcal{T}_i := \left\{ \alpha \in \mathcal{E} \ : \ \boldsymbol{\varepsilon}_i^T A_\alpha \boldsymbol{\varepsilon}_i \neq 0 \right\}, \tag{2.5}$$

and define the *point neighborhood* of a point i to be the associated point set

$$\mathcal{V}_i := \bigcup_{\alpha \in \mathcal{T}_i} \mathcal{M}_\alpha.$$

The *neighborhood stiffness matrix* in the element neighborhood of i is defined by

$$B_i := \sum_{\alpha \in \mathcal{T}_i} A_\alpha. \tag{2.6}$$

Figure 2.1 shows a typical case, where a point i is shown along with its element neighborhood and point neighborhood.

With the notation in place, measure M can now be "localized." The definition of M in (2.2) suggests that appropriate local versions of Q and A should be used in this localization. Neighborhood stiffness matrices B_i defined in (2.6) suffice for replacing A in (2.2). To localize Q, suppose $\mathcal{P}$ is partitioned into coarse-grid points, C, and fine-grid points, F: $\mathcal{P} = C \cup F$ and $C \cap F = \emptyset$. The row of Q corresponding to point $i \in F$ is given by $\mathbf{q}_i^T := \boldsymbol{\varepsilon}_i^T Q$. Assume that interpolation to $i \in F$ uses only values from the *coarse-point neighborhood of* i defined by $C_i := \mathcal{V}_i \cap C$. Let $\mathcal{Z}_i := \{\mathbf{v} \in \Re^n \,:\, v_j = 0, \;\; \text{for } j \notin C_i\}$ the set of vectors having nonzero values only at points in the coarse-point neighborhood of i. Hence, Q is constructed so that $\mathbf{q}_i \in \mathcal{Z}_i$.

We localize M over a neighborhood of a point $i \in F$ by defining

$$M_i(\mathbf{q}_i, \mathbf{e}) := \frac{\left\langle (\boldsymbol{\varepsilon}_i - \mathbf{q}_i)^T \mathbf{e}, (\boldsymbol{\varepsilon}_i - \mathbf{q}_i)^T \mathbf{e} \right\rangle \|B_i\|}{\langle B_i \mathbf{e}, \mathbf{e} \rangle}, \tag{2.7}$$

where $\mathbf{q}_i \in \mathcal{Z}_i$ and $\mathbf{e} \notin \text{Null}(B_i)$, the null space of matrix B_i.

Letting $\mathbf{e}$ in (2.7) be an eigenvector of neighborhood stiffness matrix B_i (called the *local eigenvectors*), the localized measure of interpolation quality can be used to develop a local sense of algebraic smoothness. Hence, it is practical to apply heuristic **H** to (2.7) and to attempt to bound M_i in the process of determining a suitable coarse grid and interpolation scheme. These local measures should be small, so "ideal interpolation" would be defined such that $\mathbf{q}_i$ is the arg min of the following constrained min-max problem:

$$K_i := \min_{\mathbf{q}_i \in \mathcal{Z}_i} \max_{\mathbf{e} \perp \text{Null}(B_i)} M_i(\mathbf{q}_i, \mathbf{e}), \quad \text{subject to} \;\; (\boldsymbol{\varepsilon}_i - \mathbf{q}_i)^T \mathbf{e} = 0, \;\; \forall\, \mathbf{e} \in \text{Null}(B_i). \tag{2.8}$$

Heuristic **H** is best satisfied locally by solving (2.8). Selecting the rows of the interpolation operator in this fashion is called *element interpolation.* For a given coarse grid, a global bound may be obtained by solving (2.8) for all $i \in F$. From (2.3), we recall that bounding M bounds the global convergence factor. A rigorous relation between the local and global measures of interpolation quality is given in [3]: if $M_i \le K_i, \; \forall\, i \in F$, then

$$M \le K = \max_{\alpha \in \mathcal{E}} \sum_{i \in F \cap \mathcal{M}_\alpha} K_i.$$

That is, bounding all local measures of interpolation quality M_i yields a bound on global measure M and a bound on convergence factor κ is obtained.

In [3], solving (2.8) was found to provide better convergence results than traditional AMG on several important problems such as 2D linear elasticity, by virtue of the element interpolation process. For those experiments, coarse grids were chosen geometrically. However, the ability to select coarse grids algebraically is a fundamental component of any AMG algorithm. Indeed, the fundamental requirement of AMGe, as stated above, is that the index set $\mathcal{P}$ be partitioned into C and F "points," and thus implies that the grids be conforming and that the degrees of freedom be associated with physical locations. Although the latter is frequently true, the former need not be, and is, in fact, a serious limitation. Instead, we wish to devise a method in which the coarse degrees of freedom are associated with particular agglomerates of elements, but not with any specific location in the domain.

Moreover, AMGe requires that coarse grids permit effective element interpolation from still coarser grids. Many coarse-grid selection strategies can be devised that utilize measures M_i; for example, M_i can be used to compare the quality of one potential coarse grid versus another. Unfortunately, because of the large number of possibilities, no practical coarse-grid selection algorithm can compare the measures of all potential coarse grids, even when such decisions are localized to small collections of neighboring elements. Addressing this difficulty has led to a new concept that bypasses the usual coarse-grid selection process entirely, and instead defines interpolation that satisfies basic AMGe heuristic **H** in a much more direct way. This new approach, called *spectral AMGe*, is described in the next section, and its advantages over more standard "grid-based" coarsening algorithms are explored.

3. ρAMGe. We seek the solution, $\mathbf{u} \in \Re^n$, of the linear system in (2.1). Like its predecessor AMGe, ρAMGe does not rely on the usual AMG idea that strength of connection is determined by the relative size of matrix entries, nor does it incorporate the implicit assumption that the constant function is smooth. Therefore, we are free to assume for convenience that matrix A is diagonally scaled so that its diagonal part is the identity. We continue to assume that A is SPD and that the element stiffness matrices, A_α, are available, are symmetric nonnegative definite, and satisfy $A = \sum_{\alpha \in \mathcal{E}} A_\alpha$. (Note that diagonal scaling of A implies a scaling for each A_α, but the diagonal of the scaled A_α is generally not the identity.)

The concept of an *agglomerate* is central to this work. An agglomerate is defined simply as a set of elements. We observe that a point on the grid may lie in the interior of an agglomerate in the sense that it does not appear in any other agglomerate, or it may lie on the boundary of the agglomerate in the sense that it is shared by two or more agglomerates. Now, 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 belonging to elements in τ. We need the local version of this matrix, so we define a "localization" operator, $Loc(\cdot)$, that takes a matrix as its operand and discards the zero rows and columns corresponding to points not in the agglomerate. For convenience, we denote $Loc(A_\tau)$ by A^τ; the superscript τ is used to indicate that the matrix is a *local agglomerate stiffness matrix*.

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. It would be prohibitively expensive to do this globally, however. Instead, in ρAMGe we attempt to achieve the objective by choosing local eigenvectors, in analogy to what is done in the spectral element method for discretizing partial differential equations.

The basic idea is to partition the set of elements into agglomerates, then to assemble their corresponding stiffness matrices. Eigenvectors belonging to the small eigenvalues of the local agglomerate matrices are used directly as columns of the interpolation matrix (with some modifications noted below). Conceptually, this is a very attractive idea, and has several advantages over the "grid-based" methods of standard AMG and AMGe. Such grid-based methods, in which the coarse grid is taken as a subset of the finer grid, are very natural, but place rather severe requirements on the actual grids and sets of interpolation points chosen. However, the choice of agglomerates appears to be far less critical. It is tempting to try to mimic standard AMG methods by agglomerating elements in the direction of strong connection, although it is not clear, in an element setting, how to define such a direction. Fortunately, results seem to indicate that this is unnecessary, at least for the cases we have thus far tested. It is the choice of the eigenvectors of the local agglomerate matrices, rather than the choice of the agglomerates themselves, that must permit representation of functions that are smooth in any particular direction. A more subtle restriction imposed in grid-based coarsening is on the type of smooth functions that can actually be represented in terms of their values at a few grid points. A good example of this is the Helmholtz problem, $\Delta u + k^2 u = f$. For suitable wave number k, the global null space comprises waves with a fixed wavelength but arbitrary direction. This subspace can be very difficult to represent by standard grid-based coarsening methods, but can conceivably be accurately approximated directly in terms of general local functions (although we leave that particular problem for future study).

3.1. A two-level algorithm. As is often done when defining a multilevel algorithm, we proceed by first defining and analyzing a two-level method, then extend it to the multiple-level case recursively. The two-level method is standard:

1. Perform relaxation on the fine level problem $A^h u^h = f^h$.
2. Compute the fine-level residual r^h and restrict it to be the right side of the coarse-level problem: $f^H = Rr^h$.
3. Solve the coarse-level problem $e^H = (A^H)^{-1} f^H$.
4. Interpolate the coarse-level error to the fine level and use it to correct the fine level approximation, $u^h \leftarrow u^h + Pe^H$.

Where ρAMGe differs from other approaches is in the manner of constructing its basic coarsening components, namely, coarse-level unknowns, coarse-level "elements," and prolongation P. For restriction, we use $R = P^T$, and, for the coarse-level operator, we use $A^H = P^T AP$. We thus focus our attention on construction of the basic coarsening components of the two-grid method.

Let $\mathcal{G}$ be a partition of element space $\mathcal{E}$ into agglomerates of the elements used to generate matrix A (e.g., continuous piecewise bilinear functions on square elements). Let $\tau \in \mathcal{G}$ be arbitrary. We assume, for now, that $\mathcal{G}$ has been determined on the fine level.

To determine coarse-level sets of agglomerates in an algebraic, recursive manner, we need a few important concepts. In particular, we need to define *algebraically* the notions of "interior" and "boundary" of an agglomerate. To this end, consider a point, $p \in \mathcal{P}$, that is in agglomerate $\tau \in \mathcal{G}$. 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).

In what follows, we 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. That is, 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. We could instead pose our methods and theory in the finite element *function* space setting using Sobolev inner products and norms, and associated weak forms. This would change scale considerations and local-to-global connections, but the basic principles would be the same. We choose here to take the more algebraic perspective.

The following two-level algorithm determines interpolation operator P as a mapping from the coarse degrees of freedom within each agglomerate to the fine degrees of freedom corresponding to points 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. Then $n_c = \sum_{\tau \in \mathcal{G}} m_\tau$.

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.
   - Select the eigenvectors corresponding to the m_τ eigenvalues of smallest magnitude. Extend the eigenvectors with zeros outside of τ. These vectors 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

$$w_\tau^p = a_\tau^p / \sum_{\sigma \in \mathcal{G}} a_\sigma^p,$$

   where, for $p \in P$, 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 $p \in \tau^0$. (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, $P^T AP$.

3.1.1. Analysis of the two-level algorithm. The simplicity of ρAMGe lends itself to a simple theory. We show first that the coarse subspace preserves the null space of A. For this to make sense, we must temporarily allow A to be singular. In practice, what we really wish to conclude from this theorem is that the coarse level adequately represents vectors that are in the null space of the operator A, up to the effects of boundary conditions and reaction or Helmholtz terms (e.g., those arising from discretization of $-\Delta u + bu$ for $b > 0$). 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 on the coarse level.

THEOREM 3.1. *Suppose the coarsening is constructed such that, for every agglomerate τ, the intra-agglomerate interpolation operator P_τ contains a basis for the null space of local agglomerate stiffness matrix A^τ; that is, $\forall\tau,\quad m_\tau \geq \dim(\mathrm{Null}(A^\tau))$. 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 of eigenvectors is used to ameliorate conflicts between interpolation formulas at points on 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

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

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

$$\kappa \leq \sqrt{1 - \frac{1}{K}}. \tag{3.3}$$

Recall that A^τ denotes $Loc(A_\tau)$, which is restricted matrix A_τ with all rows and columns corresponding to points not in τ eliminated. Let $\lambda_1(A^\tau) \leq \lambda_2(A^\tau) \leq \ldots$ denote the eigenvalues of A^τ. 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

$$\|A\| \cdot \|\mathbf{e} - \boldsymbol{\epsilon}\|^2 \leq \frac{\delta}{\eta} < A\mathbf{e}, \mathbf{e} > . \tag{3.4}$$

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

$$\|A_\tau\| \cdot \|\mathbf{e}_\tau - \boldsymbol{\epsilon}_\tau\|_\tau^2 \leq \delta\langle A_\tau \mathbf{e}_\tau, \mathbf{e}_\tau\rangle_\tau, \tag{3.5}$$

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 (3.4) 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 (3.4), we now use properties of W_τ and the Cauchy–Schwarz inequality to conclude that

$$\begin{aligned}
\|\mathbf{e}-\boldsymbol{\epsilon}\|^2 &= (\mathbf{e}-\boldsymbol{\epsilon})^T \left(\sum_{\tau\in\mathcal{G}} W_\tau(\mathbf{e}-\boldsymbol{\epsilon}) \right) \\
&= (\mathbf{e}-\boldsymbol{\epsilon})^T \left(\sum_{\tau\in\mathcal{G}} W_\tau(\mathbf{e}_\tau-\boldsymbol{\epsilon}_\tau) \right) \\
&= \sum_{\tau\in\mathcal{G}} \left(W_\tau^{\frac{1}{2}}(\mathbf{e}-\boldsymbol{\epsilon}) \right)^T \left(W_\tau^{\frac{1}{2}}(\mathbf{e}_\tau-\boldsymbol{\epsilon}_\tau) \right) \\
&\le \left[\sum_{\tau\in\mathcal{G}} (\mathbf{e}-\boldsymbol{\epsilon})^T W_\tau (\mathbf{e}-\boldsymbol{\epsilon}) \right]^{\frac{1}{2}} \left[\sum_{\tau\in\mathcal{G}} \|W_\tau^{\frac{1}{2}}(\mathbf{e}_\tau-\boldsymbol{\epsilon}_\tau)\|_\tau^2 \right]^{\frac{1}{2}} \\
&= \|\mathbf{e}-\boldsymbol{\epsilon}\| \left[\sum_{\tau\in\mathcal{G}} \|W_\tau^{\frac{1}{2}}(\mathbf{e}_\tau-\boldsymbol{\epsilon}_\tau)\|_\tau^2 \right]^{\frac{1}{2}}.
\end{aligned}$$

That is,

$$\|\mathbf{e}-\boldsymbol{\epsilon}\|^2 \le \sum_{\tau\in\mathcal{G}} \|W_\tau^{\frac{1}{2}}(\mathbf{e}_\tau-\boldsymbol{\epsilon}_\tau)\|_\tau^2.$$

Therefore,

$$\begin{aligned}
\|\mathbf{e}-\boldsymbol{\epsilon}\|^2 &\le \sum_{\tau\in\mathcal{G}} \|W_\tau^{\frac{1}{2}}(\mathbf{e}_\tau-\boldsymbol{\epsilon}_\tau)\|_\tau^2 \\
&\le \sum_{\tau\in\mathcal{G}} \|\mathbf{e}_\tau-\boldsymbol{\epsilon}_\tau\|_\tau^2 \\
&\le \delta \sum_{\tau\in\mathcal{G}} \frac{\langle A_\tau \mathbf{e}_\tau, \mathbf{e}_\tau\rangle_\tau}{\|A_\tau\|} \\
&\le \frac{\delta}{\eta\|A\|} \sum_{\tau\in\mathcal{G}} \langle A_\tau \mathbf{e}_\tau, \mathbf{e}_\tau\rangle \\
&= \frac{\delta}{\eta\|A\|} \langle A\mathbf{e}, \mathbf{e}\rangle.
\end{aligned}$$

□

We have as yet said nothing about how we select a value for m_τ, the number of coarse degrees of freedom in each agglomerate $\tau \in \mathcal{G}$. However, determining the number of coarse degrees of freedom to use should involve a balancing of both the effectiveness of the coarse-grid correction and the cost accrued in the grid and operator complexities. These complexities may be more fully discussed in a multilevel, rather than a two-level, framework. Accordingly, we postpone discussion of the choice of m_τ until we have described the multilevel algorithm and dispensed with certain complications that arise in the multilevel approach.

3.2. A multilevel algorithm. In a multilevel algorithm, a coarse grid is treated by relaxation, followed by correction from a still coarser grid. The effectiveness of the components used in a multilevel algorithm is determined largely by the character of the coarse grids. Here, we turn our attention to altering the two-level ρAMGe algorithm in ways that aim to produce coarse-level problems that are themselves amenable to further coarsening by the ρAMGe approach.

Two-level ρAMGe, as proposed in Section 3.1, produces promising numerical results (see Section 4.) Therefore, recursion of the two-level algorithm is a natural step in the development of a multilevel algorithm. Local stiffness matrices play a fundamental role in the construction of multigrid components in ρAMGe. 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 the coarse level to enable a local spectral decomposition there. (We could appeal to the finest grid stiffness matrices, but this is infeasible for large problems with many coarse levels.) An efficient multilevel scheme thus requires some coarse-grid version of these local stiffness matrices. Such matrices could be created in a manner analogous to the coarse-grid operator by taking $P^T A_\alpha P$. However, since the *number* of matrices is not reduced in this method, this approach would also be very costly.

An obvious solution is simply to combine the fine-level stiffness matrices in some sensible way. A natural choice is to form the coarse-grid element matrix corresponding to each agglomerate τ by computing $P^T A_\tau P$. However, as we show below, problems can arise with this approach, so we describe several potentially more useful alternatives for defining coarse-grid elements.

3.2.1. Coarse-grid elements and creep. Construction of the coarse-grid elements begins with some agglomerate of fine-grid elements. We refer to these as *e-agglomerates* to distinguish them from the agglomerates used to construct interpolation matrix P, which we refer to as *p-agglomerates*. For now, it is best to think of the e-agglomerates and the p-agglomerates as being the same. Later, we choose them to be different as a means of avoiding the difficulties we describe in this section.

While we refer to the coarse-grid matrices as *element* matrices, they are in reality purely algebraic entities, and the familiar geometric correspondence between the element (seen as a subset of the domain) and its nodes can be lost. We can even lose the property that the coarse-grid matrix can be written as a sum of its element matrices. What is important, however, is that these element matrices preserve a local sense of smoothness and, in particular, can be used to exactly represent the global null space.

Before proceeding, it is instructive to introduce the idea of *creep*. This difficulty we first encountered with coarse-level element matrices in earlier work on AMGe, where coarse-level elements are simply taken as unions of finer-level elements, the coarse grids chosen as some appropriate subset of the finer grid, and interpolation is then defined. We observed that the coarse-level matrices in AMGe could occasionally exhibit spurious null vectors that do not correspond in any way to the global null space. A necessary (but not sufficient) condition for this to occur is that the coarse-grid nodes involved in the coarse-grid element matrix extend beyond the union of corresponding fine-grid elements. The term "creep," as originally coined, describes this particular geometric difficulty, but has come more generally to mean the appearance of spurious null vectors of coarse-grid element (or agglomerate) matrices; these are local null vectors that do not correspond to global null vectors.

Creep is a 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 those of the global matrix (and, in fact, are *exact* for the null space) and that relaxation cannot effectively reduce such components, accurate interpolation of all such local components is critical to the success of the method. The presence of creep indicates an unnecessarily large null space in the coarse-grid agglomerate. Since we must choose *all* local eigenvectors belonging to zero eigenvalues if we want (3.5) to hold, then we must have $m_\tau \geq \dim(\mathrm{Null}(A^\tau))$. Thus, uncontrolled creep can lead to unacceptable complexity. Given the importance of this concept, we now examine it from both an algebraic and a geometric perspective in the ρAMGe setting.

An algebraic perspective on creep. 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 let $\mathcal{M}_\gamma^c$ be the set of coarse-grid degrees of freedom used to interpolate values 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 $Loc(\sum_{\alpha\in\gamma} A_\alpha)$. Choosing a large enough m_τ therefore induces creep, which in turn results in even larger complexity on coarser levels since $|\mathcal{M}^c_\gamma|$ tends to increase with increasing m_τ.

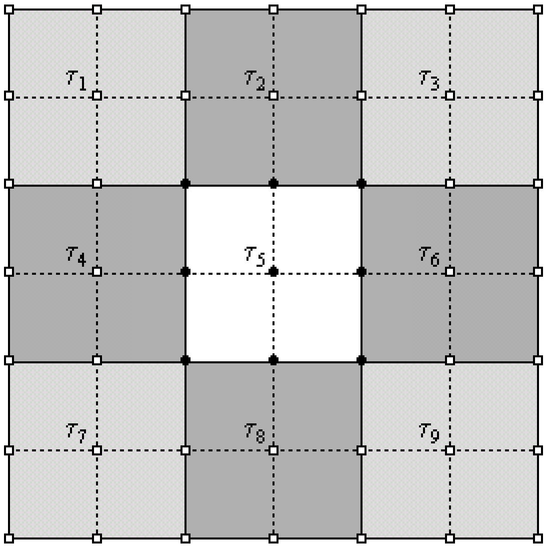

FIG. 3.1. *Partition of a domain into 36 square elements. The shaded regions represent p-agglomerates, $\tau_1, \tau_2, \ldots, \tau_9$. The solid dots (•) indicate set $\mathcal{M}_{\tau_5}$, while all other degrees of freedom are indicated by open squares (□). $X_{\tau_5} = \{\tau_i : i = 1, 2, \ldots, 9\}$.*

3.2.2. Fuzzy elements and creep. Consider the uniform grid of square elements depicted in Figure 3.1. The shaded regions represent p-agglomerates and the dotted lines delineate elements. Note that this 6×6 array of elements contains nine 2×2 p-agglomerates, $\tau_1, \tau_2, \ldots, \tau_9$. Recall that we are taking e-agglomerates and p-agglomerates to be the same. Now, the arbitration of interpolation formulas for nodes on the boundary of τ_5 results in values on τ_5 being interpolated from the coarse-grid degrees of freedom corresponding to $\tau_1, \tau_2, \ldots, \tau_9$. Were the boundary nodes of τ_5 to be interpolated only from coarse-grid degrees of freedom corresponding to τ_5, no creep would result in the coarse element associated with τ_5. This could be accomplished with an arbitration that assigned each fine-grid degree of freedom to exactly one p-agglomerate (i.e., if p is a fine-grid degree of freedom shared by more than one p-agglomerate, set $w^p_\tau = 1$ for one specific τ and $w^p_\tau = 0$ for all others.) Unfortunately, this could easily interfere with the ability to approximate smooth functions not explicitly included in the local interpolation. For instance, for a Poisson operator, taking this approach with $m_\tau = 1$ gives piecewise constant interpolation, which would exactly interpolate the global near null space, but yields unsatisfactory multilevel convergence. An alternative is to enlarge the e-agglomerate to all of X_{τ_5}, so that the finer grid agglomerate matrix is able to "see" all interpolated functions.

We begin 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. In trying to examine this method on coarser levels, it is often useful to associate each coarse-grid element with its core. Referring to Figure 3.1, note that a fuzzy agglomerate with core $\gamma = \tau_5$ would have $\tau_1 \cup \ldots \cup \tau_4 \cup \tau_6 \cup \ldots \cup \tau_9$ as its fuzz.

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: the matrix corresponding to the fuzzy agglomerate with core γ is

$$\hat{A}_\gamma = \alpha_1 A_\gamma + \alpha_2 A_{\bar{\gamma}\setminus\gamma}. \tag{3.6}$$

Empirically it seems reasonable to take $\alpha_1 = 1.0$ and $\alpha_2 = 0.5$.

One difficulty is that if we simply 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 use only the agglomerate matrix corresponding to γ. Furthermore, there is no guarantee that using $P^T\hat{A}_\gamma P$ eliminates creep. To avoid this difficulty, we instead use a local interpolation operator in place of the global operator P for constructing the element matrix. Specifically, we construct local interpolation operator 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 formulas 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 $(P_\gamma^{fuzz})^T\hat{A}_\gamma P_\gamma^{fuzz}$ and referred to as a *fuzzy element*, maintains the appropriate null space while tending to eradicate creep. As long as $Loc(P_\gamma^{fuzz})$ is full rank, then the corresponding coarse-grid element matrix has no spurious null vectors. An interesting consequence is that the resulting coarse grid is nonconforming in the sense that the global coarse-grid operator is no longer the sum of the coarse-grid element matrices.

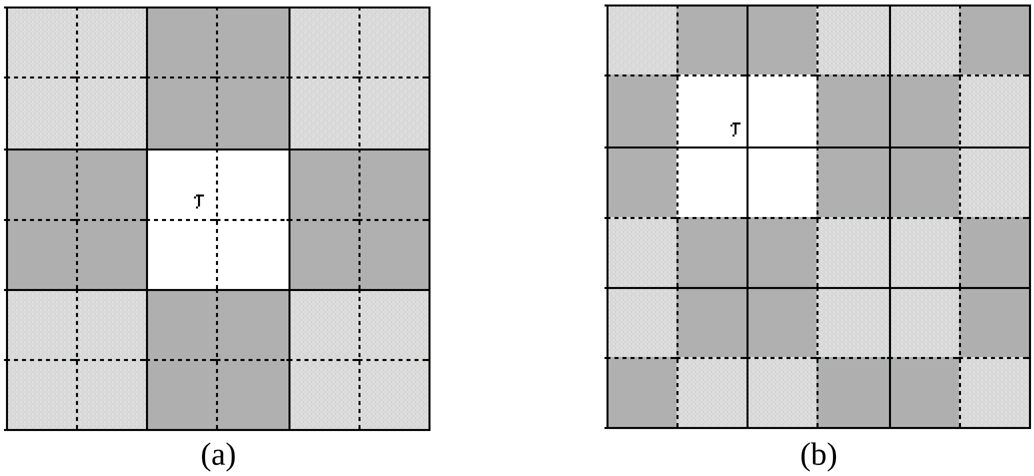

FIG. 3.2. *Partition of a grid into square elements, delineated by solid and dotted lines. The solid lines delineate element cores (e-agglomerates) used to form the coarse-grid elements. The single-color regions (shaded or white) in each figure represent the p-agglomerates used to determine interpolation. (a) The p-agglomerates coincide with the coarse-grid elements. (b) The p-agglomerates are staggered with respect to the coarse-grid elements.*

3.3. Choosing m_τ: balancing accuracy and complexity. We must still define a specific process for selecting the value of m_τ for the two-level ρAMGe algorithm and, recursively, for the multilevel algorithm. This decision affects the construction of fundamental components of the multigrid process. Increasing m_τ produces a richer interpolation and resulting coarse grid in the sense that it improves effectiveness of the coarse-grid correction, but it also increases operator and coarse-grid complexity. A careful balance must be made to optimize accuracy versus cost. To accomplish this, we first define $\kappa^{1/w}$ to be the *effective convergence factor* of one ρAMGe cycle, where κ is a bound on its convergence factor and w is the number of work units it requires. One work unit is defined as the cost of one fine-grid relaxation step. Our goal is to minimize $\kappa^{1/w}$.

Suppose we wished to optimize accuracy without regard to complexity. Then, for all $\tau \in \mathcal{G}$, we would select $m_\tau = n_\tau$, the dimension of local agglomerate stiffness matrix A^τ. In general, $\sum_{\tau\in\mathcal{G}} n_\tau > n$; hence, choosing $m_\tau = n_\tau$ yields the impractical result that the coarse-grid problem would systematically contain more degrees of freedom than the fine-grid problem. While this example is clearly extreme, it underscores the need to balance accuracy and complexity in selecting m_τ.

To achieve this balance, we develop an *accuracy/cost measure* that we use to guide the process of selecting a coarse grid. The example just given dictates that we must consider both components, accuracy and complexity, in order to define an effective tool for coarsening. In addition, any useful measure must use local qualities, assuming them to effectively represent the global characteristics of the algorithm. By contrast, any global approach to building a useful accuracy/cost measure would be prohibitively expensive.

Accuracy. First consider the component of the measure that addresses accuracy. We take the convenient premise here that coarse-grid correction completely eliminates all error in the directions of the first m_τ eigenvectors, and that the remaining eigenvectors are reduced by one Richardson relaxation, with optimal parameter ω. The error propagation matrix in this case is given in terms of the local agglomerate stiffness matrix by $I - \omega A^\tau$. We desire that this iteration effectively reduce the $m_\tau+1, m_\tau+2, \ldots, n_\tau$ eigenvectors of A^τ. Observe that if $\lambda_{m_\tau+1}, \lambda_{m_\tau+2}, \ldots, \lambda_{n_\tau}$ are the corresponding eigenvalues of A^τ, then the eigenvalues of Richardson's error propagation matrix are $1-\omega\lambda_{m_\tau+1}, 1-\omega\lambda_{m_\tau+2}, \ldots, 1-\omega\lambda_{n_\tau}$. The best overall reduction occurs when this range is centered at 0, that is, when $-(1-\omega\lambda_{n_\tau}) = 1-\omega\lambda_{m_\tau+1}$. This relation determines the optimal value of $\omega = 2/(\lambda_{n_\tau}+\lambda_{m_\tau+1})$, which achieves the optimal error reduction factor of

$$\begin{aligned} 1-\omega\lambda_{m_\tau+1} &= 1 - \frac{2\lambda_{m_\tau+1}}{\lambda_{n_\tau}+\lambda_{m_\tau+1}} \\ &= \frac{\lambda_{n_\tau}-\lambda_{m_\tau+1}}{\lambda_{n_\tau}+\lambda_{m_\tau+1}}. \end{aligned} \tag{3.7}$$

This reduction factor is used as a main component of the accuracy/cost measure we develop here.

Cost of the algorithm using a grid complexity estimate. An estimate of cost of the algorithm constitutes the second component of the accuracy/cost measure. In AMG, algorithmic cost is measured in terms of either *grid complexity*, defined as the total number of degrees of freedom on all grids divided by the number of degrees of freedom on the fine grid, or *operator complexity*, defined as the total number of nonzero entries in all operator matrices on all levels divided by the number of nonzero entries in the fine-grid operator matrix. These quantities are used because the number of arithmetic operations per cycle, as well as the the amount of memory required in an algorithm, can be expressed in terms proportional to them. As we did with the accuracy component of the measure, we base our definitions of the cost component on local information; hence, a measure must be designed to estimate global cost based on local information. The underlying premise, of course, is that the local pattern is replicated throughout the domain.

Suppose $m_\tau := m$ eigenvectors are chosen in every agglomerate. Then the number of degrees of freedom on the coarse grid is $m|\mathcal{G}|$, where $|\mathcal{G}|$ is the number of agglomerates on the fine grid. To obtain a local estimate of the number of degrees of freedom on the fine grid, we could make the convenient assumption that every agglomerate on the fine grid contains precisely m^f degrees of freedom. In this case, however, $m^f|\mathcal{G}|$ overestimates the number of degrees of freedom on the fine grid because while the degrees of freedom in the interiors of the agglomerates are counted correctly, the degrees of freedom on the agglomerate boundaries are shared by agglomerates and, hence, are counted more than once. Instead, we devise a measure that counts a degree of freedom with a weight proportional to the number of agglomerates to which it belongs: the count for $p \in \mathcal{P}$ is $1/|\mathcal{N}(p)|$, where $\mathcal{N}(p) = \{\tau \in \mathcal{G} : p \in \tau\}$ is the agglomerate neighborhood of p. Note that the count for a point in $\tau^0 \cap \mathcal{P}$ is 1. Thus the weighted number of fine-grid degrees of freedom for agglomerate $\tau \in \mathcal{G}$ is defined by

$$N_\tau := \sum_{p\in\tau} \frac{1}{|\mathcal{N}(p)|}.$$

If $N_\tau := \tilde{N}$ is the same for all $\tau \in \mathcal{G}$, then the number of fine-grid points is $\tilde{N}|\mathcal{G}|$. Thus, an estimated coarsening ratio can be given by $c_g := m/\tilde{N}$, and the cost of one ρAMGe cycle is roughly proportional to

$$1 + c_g + c_g^2 + \ldots = \frac{1}{1 - c_g} \quad = \quad \frac{\tilde{N}}{\tilde{N} - m}. \tag{3.8}$$

An accuracy/grid-cost measure. To estimate the accuracy (error reduction) versus cost of the algorithm, we may compute an estimate of convergence per work unit (in this case, one fine-grid Richardson relaxation sweep) using (3.7) and (3.8):

$$\mu_g := accuracy^{1/cost} = \left(\frac{\lambda_{n_\tau} - \lambda_{m_\tau + 1}}{\lambda_{n_\tau} + \lambda_{m_\tau + 1}} \right)^{\frac{\tilde{N} - m}{\tilde{N}}}. \tag{3.9}$$

By construction, (3.9) reflects the accuracy as it relates to grid complexity (accounting for the notation μ_g).

Cost of the algorithm using an operator complexity estimate. Another perspective on algorithmic cost is obtained by considering operator complexity, which may in fact be better than grid complexity as a measure of work performed in a cycle. 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 $m_{\tau_f} := \tilde{N}/n_e$. To elaborate on the difference between a measure of grid complexity versus operator complexity, note that the size of m affects not only the number of rows in the coarse-grid operator, but also the number of nonzeros per row. The coarsening ratio is estimated to be $c_g = m/\tilde{N}$, so, assuming that operator complexity is proportional to the square of the number of degrees of freedom per element, then the ratio of the number of nonzeros in the coarse-grid operator to those in the fine-grid operator is $c_{op} := (m/m_{\tau_f})^2/n_e$. A two-level operator complexity growth measure can be defined by

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

Considering multiple levels argues against using higher values of m_τ. Specifically, a p-level operator complexity growth measure may be defined by

$$\text{“p-level” cost} := 1 + c_{op} + c_{op}^2 + \ldots + c_{op}^{(p-1)}. \tag{3.10}$$

In analogy to grid complexity growth measure (3.8), operator complexity growth measure (3.10) could be defined as $1/(1 - c_{op})$. However, practical experiences suggest that, for some test cases, c_{op} can be greater than one when moving from the fine level to the first coarse level, resulting in an infinite (or negative) complexity measure. Consider Poisson's equation, discretized using bilinear functions on rectangular elements. The use of 2×2 agglomerates on the finest level necessitates $m_\tau = 1$ in order to maintain $c_{op} < 1$, but this is far too restrictive for rectangular elements with large aspect ratios. In such cases, the assumption, implicit in the estimate, that c_{op} grows at the same rate on all coarser grids is not generally warranted. Unfortunately, more realistic growth rate estimates for c_{op} are difficult to derive. To avoid having separate complexity measures on different levels, we employ a compromise in which the number of levels in the estimate is fixed. For small p, measure (3.10) does not restrict m_τ excessively; hence, we use small p in practice. In fact, empirical results indicate that a 3-level cost measure ($p = 3$) produces the most satisfactory results. To reflect the true cost of the algorithm without resorting to an arbitrary choice of p, we intend to refine the operator complexity estimate in future research. We also observe that, if we really wish to restrict $c_{op} < 1$ on all levels, larger agglomerates could likely be used, allowing more freedom in the choice of m_τ.

An accuracy/operator-cost measure. Estimating the accuracy (error reduction) versus cost of the algorithm using operator complexity allows us to produce the alternate accuracy/cost measure

$$\mu_{op} := accuracy^{1/cost} = \left(\frac{\lambda_{n_\tau} - \lambda_{m_\tau + 1}}{\lambda_{n_\tau} + \lambda_{m_\tau + 1}} \right)^{1/(1 + c_{op} + c_{op}^2 + \ldots + (c_{op})^{p-1})}. \tag{3.11}$$

We have two accuracy/cost measures, μ_g and μ_{op}. Limited numerical tests indicate that use of the 3-level operator complexity measure to determine m_τ produces the most robust approach. A numerical comparison between the uses of μ_{op} and μ_g appears in Section 4.1.

A few comments on cost/accuracy measures. We developed the geometric and algebraic complexity measures, and from them the two accuracy/cost measures μ_g and μ_{op}, with the explicit assumption that the discretization is roughly the same everywhere (thus allowing the substitutions $\mathbf{m}_\tau := m$ and $N_\tau := \tilde{N}$). This works well enough on regular grids (such as the ones used in the numerical tests in Section 4), recognizing that minor modification occurs because of agglomerates involving boundary elements. For truly unstructured grids, of course, the discretization can vary systematically over different regions of the domain, so that in fact μ_g and μ_{op} should depend on τ, or at least on regional averages. Indeed, one could imagine that the use of the measures to guide the coarsening could be applied regionally, with the goal of implementing careful local actions to control the global complexities resulting from the coarsening process. However, we have not yet implemented an algorithm to do this, preferring instead for now to apply the measures as a result of a fixed "average" over the grid. Continuing research is being pursued into effective methods for nonuniform grids.

In a similar vein, we observe that the optimal value of the Richardson relaxation weighting parameter $\omega = 2/(\lambda_{n_\tau} + \lambda_{m_\tau+1})$ is also dependent on the agglomerate τ. In practice, of course, it is impractical to compute a weighting factor for each agglomerate. The accuracy/cost measure can be used to guide development of the multigrid operators for regular-grid problems by applying a global average for the weighting parameter. More frequently in practice, however, it is simpler to use Gauss-Seidel or some other relaxation in which the cost is fixed.

Finally, we note that one assumption that we have made is that the reduction in the number of degrees of freedom between the first two grids is repeated on the coarser levels. While this assumption is not unreasonable (although not always strictly accurate) for scalar problems, there are many systems problems for which this is unlikely to occur. In certain elasticity problems, for example, especially in 3D, the first coarse operator often exhibits a noticeably smaller reduction in the number of degrees of freedom than is seen on the succeeding coarse levels. For situations such as these, it is possible that an effective strategy could be devised in which more realistic estimates of costs could be developed by computing different values of c_g or c_{op} for each coarser level as the coarsening proceeds.

3.4. 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 $m|X_\gamma|$. 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 p-agglomerates as 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. This is illustrated in Figures 3.2 (a) and 3.2 (b). In both figures, the central 2×2 agglomerate is an element core, γ, while the white regions marked with τ show the orientations of the p-agglomerates with respect to γ. In Figure 3.2 (a), X_γ consists of the central p-agglomerate and all 8 surrounding p-agglomerates, so that $|X_\gamma| = 9$. In Figure 3.2 (b), where the single-color p-agglomerates (shaded or white) are staggered with respect to the central e-agglomerate, $|X_\gamma| = 4$. Since m would be the same in either case, storage for the coarse-grid elements is proportional to 81 in the first case and 16 in the second, so that staggered elements give better than a five-fold reduction in coarse-grid element matrix storage requirements. In a similar $3D$ problem, savings of the staggered case would be 64/729 over the unstaggered case, better than a factor of 11. Further, assuming that we again construct 2×2 p-agglomerates on the coarser grid, the sizes of the coarse-grid agglomerate matrices would be $16m \times 16m$ in the unstaggered case and $9m \times 9m$ in the staggered case. Assuming an $O(N^3)$ cost for computing eigenvectors and eigenvalues, staggering reduces work here by a factor of about 5.6 in $2D$ and about 13.3 in $3D$. ($O(N^3)$ is the cost of computing a complete machine-accurate eigenvector decomposition. More efficient methods should be used in practice, but the matrix size would still strongly affect the cost of this computation.) The smaller size also reduces the work involved in computing the element matrices on the next-coarser grid. An added benefit of these staggered agglomerates is that smaller $|X_\gamma|$ tends to limit creep even without the addition of fuzz to the e-agglomerates, although we generally use the fuzz in practice.

3.4.1. An algorithm for choosing staggered coarse-grid agglomerates. A current area of research is to study some of the many ways to choose such staggered elements and agglomerates. Here we describe one approach that seems very promising. Assume that we are on level k and are given level k elements and associated matrices. We begin by partitioning the set of level k elements into a set of level $k+1$ e-agglomerates (element cores), denoted by $\mathcal{C}$. (While an automatic agglomeration algorithm such as that developed by Jones and Vassilevski [14] could be used, the numerical tests reported here generally involve the use of standard 2×2 e-agglomerates.)

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

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.

The local stiffness matrices over each p-agglomerate, $\tau \in \mathcal{G}$, can now be assembled, and interpolation can be defined (as described previously). We can then 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 $(P_\gamma^{fuzz})^T \hat{A}_\gamma P_\tau^{fuzz}$ (as in Section 3.2.2). 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. Also, it is now possible to discard the element matrices on the current level, since they will no longer be required.

A nice, but nonessential, feature of this approach comes from the fact that each p-agglomerate is associated with a particular fine-grid degree of freedom (i in the above algorithm). On the finest grid, it is common that a number of degrees of freedom are defined at physical nodes of the elements. This nodal association is "inherited" by the block of coarse-grid degrees of freedom defined over the p-agglomerate. This formally gives the coarse-grid elements the same structure as the fine grid, so that each element is associated with some set of nodes and each node is associated with a set of degrees of freedom. We may therefore rewrite the algorithm above in terms of "nodes" on each level, rather than the degrees of freedom, which improves efficiency. Furthermore, this effect also provides some advantages in terms of writing agglomeration algorithms based on generalized "faces" of elements, where a face can be defined as a maximal intersection of two elements.

As noted above, staggered agglomerates reduce complexity and limit creep. This can be seen, to some extent, in the results presented next, although exposing these effects fully requires more extensive tests than those we offer.

3.5. A note about relaxation. The ideas behind ρAMGe can be generalized by minimizing a Rayleigh quotient that takes relaxation into account. In the forgoing discussion, we did this using Richardson relaxation, because the error propagation matrix is $I - sA$ and we minimize $\langle Ax, x\rangle / \langle x, x\rangle$. In general, we could choose to minimize $\langle Bx, x\rangle / \langle x, x\rangle$ where the error propagation matrix is $I - B$ and B is symmetric. Other choices could be made when B is nonsymmetric. We chose Richardson here to simplify the theoretical discussion.

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

For detailed results using the original AMGe (as outlined in Section 2), see [3]. Comparing results of such an AMGe algorithm to ρAMGe poses a difficult (if not impossible) task: as noted in Section 2, robust coloring algorithms for the original AMGe scheme are problematic and currently do not even exist. Furthermore, ρAMGe develops coarse grids that *per se* are not subsets of the fine grids. Because of this, we lose a sense of C-points, F-points, and unknowns on coarser levels, making comparison still more difficult.

While the ρAMGe theory assumes Richardson (or 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

$$C_{\mathcal{P}} = \frac{\sum_{k=1}^{m} n_{\mathcal{P}}^k}{n_{\mathcal{P}}^1} \tag{4.1}$$

and operator complexity by

$$C_L = \frac{\sum_{k=1}^{m} n_L^k}{n_L^1}. \tag{4.2}$$

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: $\kappa = ||r^{20}||/||r^{19}||$. Matrix A is diagonally scaled, so that its diagonal part is the identity matrix.

The tables giving the experimental results display grid and operator complexities as well as estimated asymptotic convergence factors.

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. Here, we illustrate by numerical example the evolution of ρAMGe from a two-level to a multilevel algorithm. Along the way, we observe the numerical results that give rise to the mathematical stepping stones leading to a multilevel ρAMGe.

In particular, we consider the Poisson equation discretized with bilinear functions on a 32×32 square grid of square elements. Homogeneous Dirichlet boundary conditions are assumed, and they are eliminated from the matrix during discretization. Standard 2×2 agglomeration is forced on each level.

Creep. Table 4.1 illustrates the effect of creep. This effect is clearly evident as the level becomes coarser (larger k). Due to the Dirichlet boundary conditions, $\dim(\mathrm{Null}(A^\tau))$ varies somewhat over τ in the domain. Therefore, Table 4.1 measures $\max_{\tau \in \mathcal{G}^{k-1}}(\dim(\mathrm{Null}(A^\tau)))$ as an indication of creep in the k-level problem. Note that we consider $\tau \in \mathcal{G}^{k-1}$ since stiffness matrices are unnecessary in the direct solve on level k. The effect of creep on operator complexity alone is evidence enough that major enhancements in the two-level ρAMGe algorithm are required.

Staggering elements. Complexity is reduced by staggering coarse-grid elements relative to fine-grid agglomerates (see Section 3.4). The Poisson problem serves as an instructive example regarding the benefit of this algorithm enhancement. It is clearly seen in Table 4.1 that staggering the elements can significantly improve multilevel algorithm convergence and complexity (improvement is observed in both types of complexity, but it is especially dramatic for operator complexity).

Eliminating creep with fuzzy elements. The null cardinality column of Table 4.1 reveals that staggering elements relative to fine-grid agglomerates reduces, but does not entirely eliminate, creep. In fact, creep appears as early as level $k = 3$. For large-scale problems necessitating many more levels and, perhaps, unstructured grids, creep could wreak much more serious devastation on the multigrid components. The use of fuzzy elements in ρAMGe to eliminate creep was described in Section 3.2.2 and is illustrated in Table 4.1. The effectiveness of the technique is clearly illustrated in the table, where the null cardinality remains at unity for all levels in which the fuzzy elements are employed.

	k	κ	$C_{\mathcal{P}}$	C_L	dim(Null(A^τ))
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
Staggered elements	2	0.05	1.53	2.0	1
	3	0.05	1.76	2.6	2
	4	0.07	1.88	3.0	2
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 4.1

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, $C_{\mathcal{P}}$ and C_L are the respective grid and operator complexities, and dim(Null(A^τ)) *is the maximum dimension of the agglomerate null spaces; tests used μ_{op} with $p = 3$.*

	μ_{op}			μ_g			
k	κ	$C_{\mathcal{P}}$	C_L	κ	$C_{\mathcal{P}}$	C_L	dim(Null(A^τ))
2	0.05	1.53	2.00	0.05	1.53	2.00	1
3	0.05	1.67	2.23	0.05	1.77	2.79	1
4	0.05	1.71	2.29	0.05	1.86	3.18	1
5	0.05	1.72	2.30	0.05	1.91	3.37	1

Table 4.2

Multilevel ρAMGe for Poisson's equation discretized with bilinear functions on square elements on a 32×32 grid, using either operator (μ_{op}) or grid (μ_g) complexity.

The use of staggering with fuzzy elements appears to reduce operator complexity somewhat, as well as to eliminate creep. This reduction in operator complexity is observable (though relatively slight) in Table 4.1; however, we note without demonstration that it is somewhat more noticeable for more computationally difficult problems, such as linear elasticity. In addition, the staggered approach actually involves less work in the creation of (and storage for) the fuzzy elements, as described above (Section 3.4 and Figure 3.2).

Using the accuracy/cost measure and fuzzy weighting. Even for traditional AMG, a bewildering host of options exists for the algorithm implementation. This is also true of ρAMGe. To find our way through these options to robust algorithms, we begin by experimenting on the model problem choices moderated by the above discussion. We then test the most promising options on other problems to suggest their robustness.

We present in Table 4.2 results of choices regarding the complexity measure and the weight of the agglomerate fuzz for the Poisson problem. The data show that the complexity of coarse grids depends highly on the accuracy/cost measure, so that selecting one such measure to implement directly affects the multigrid process. Further, we see that the accuracy/cost estimate incorporating a three-level operator complexity growth measure (μ_{op} in (3.11) with $p = 3$) shows lower complexity (both grid and operator) than does the measure incorporating grid complexity (μ_g in (3.9)). The remaining numerical tests therefore use μ_{op} with a three-level operator complexity growth measure. A second option is that fuzzy elements contain a core and a fuzz that can be weighted in various ways in the construction of agglomerates (see Section 3.2.2). Table 4.2 reports on agglomerates that are formed using (3.6) with weights $\alpha_1 = 1.0$ and $\alpha_2 = 0.5$. Limited testing with these (and other) choices show them to be the

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 4.3
Linear elasticity discretized with bilinear functions on square elements ($h_x = h_y = 1/32$) on a 32×32 grid.

most robust of those considered; this preliminary conclusion requires further study, however, especially for a broader range of applications.

Accuracy estimate. The convergence bound developed in the theory of ρAMGe tends to be pessimistic, as is the case with its predecessor, AMGe. For example, the two-level accuracy estimate is 0.67 for Poisson's equation discretized with bilinear functions on square elements on a 32×32 grid. Despite this bound, in practice we often see much lower convergence factors, as is amply demonstrated in Table 4.2. Yet, the theoretical estimates serve in guiding the choices necessary in constructing multigrid components, and give insight into the way that algorithm development must proceed. Developing an accuracy estimate yielding a tighter convergence bound is an open, and important, question in ρAMGe research.

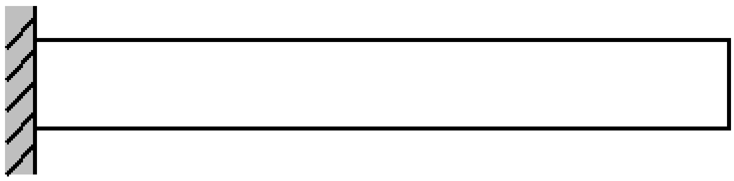

FIG. 4.1. *Plane-stress cantilever beam problem.*

4.2. Applying ρAMGe to linear elasticity. Much of the motivation for research into AMGe methods began in an effort to develop robust methods to solve matrix equations arising from systems of differential equations. At the focus of current ρAMGe research is the 2D linear elasticity system

$$\begin{aligned} u_{xx} + \frac{1-\beta}{2} u_{yy} + \frac{1+\beta}{2} v_{xy} &= f_1, \\ \frac{1+\beta}{2} u_{xy} + \frac{1-\beta}{2} v_{xx} + v_{yy} &= f_2, \end{aligned} \tag{4.3}$$

where u and v are displacements in the x and y directions, respectively. Throughout the numerical tests, we employ the value $\beta = 1/2$, which yields the Poisson ratio $\nu = \beta/(1+\beta) = 1/3$. The problem, depicted in Figure 4.1, has free boundaries, except on the left where $u = v = 0$. We discretize (4.3) 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 ($n_x h_x$ by $n_y h_y$) varies in size, depending on the values of n_x, n_y, h_x, and h_y.

4.2.1. Forcing agglomerates. To apply ρAMGe to (4.3) in this study, the agglomerates are predetermined on each level. We consider the effects of several choices. For all experiments, we use μ_{op} as the accuracy/cost measure and the parameter values $\alpha_1 = 1.0$ and $\alpha_2 = 0.5$ in (3.6) to form fuzzy agglomerates.

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 contains only one agglomerate, where the coarse-grid problem is solved directly. Table 4.3 reveals that reasonable grid and operator complexities are present throughout all 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 4.4

Linear elasticity discretized with stretched rectangular elements ($h_x = 1/32$ and $h_y = 1/320$) on a 32×32 grid.

	Square Elements			Stretched Elements		
k	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 4.5

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.

The experiments reported in Table 4.4 use a rectangular domain on which stretched (rectangular) elements with an aspect ratio of $10 : 1$ (i.e., $h_x = 1/32, h_y = 1/320$) are used to discretize to a 32×32 grid. Various patterns for agglomeration are forced on each level. Forcing 2×1, 1×2, or 2×2 agglomerates all yield acceptable two-level convergence factors, reflecting a certain flexibility toward agglomeration in ρAMGe (see Table 4.4).

This flexibility apparently disappears if 2×1 agglomerates are forced for the 3-level test. In fact, 2×1 fine-grid agglomerates produce coarse-grid elements with a relatively large number of associated eigenvalues close to 0. This effect on the spectrum of coarse-grid elements does not occur with 1×2 agglomerates, which results in more encouraging convergence factors, as seen in Table 4.4.

Finally, ρAMGe is applied to a thin 2-D plane-stress cantilever beam (the beam is a single-element thick). Specifically, linear elasticity equations (4.3) 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 first with square elements ($h_x = h_y = 1/64$) and then with stretched elements possessing a $10 : 1$ aspect ratio (i.e., $h_x = 1/64, h_y = 1/(640)$. In both cases, 2×1 agglomerates are forced on each level.

Table 4.5 reveals that ρAMGe performs well on the 2-D plane-stress cantilever beam. Acceptable results are attained when the domain is partitioned using either square or stretched elements. This robustness is an important feature of ρAMGe and bodes well for its potential to efficiently solve a larger family of problems.

4.2.2. Selecting agglomerates algebraically. The numerical results suggest that ρAMGe is often not very sensitive to the type of agglomeration chosen. Therefore, there is promise that an algebraic algorithm can be designed to select agglomerates. Yet, Table 4.4 suggests that a robust algorithm should still take the operator into consideration.

k	Square Elements			Stretched Elements		
	κ	$C_{\mathcal{P}}$	C_L	κ	$C_{\mathcal{P}}$	C_L
2	0.14	1.49	1.94	0.22	1.72	3.09
3	0.19	1.63	2.20	0.28	2.11	5.29
4	0.21	1.67	2.27	0.36	2.31	6.74
5	0.24	1.69	2.29	0.43	2.39	7.37

TABLE 4.6

Linear elasticity discretized with bilinear functions on rectangular elements on a 32×32 *grid, with the Jones-Vassilevski agglomeration algorithm used for aggregation. Again, stretched elements use a* $10:1$ *aspect ratio.*

As an initial step in this direction, we consider ρAMGe using the Jones-Vassilevski algebraic agglomeration algorithm introduced in [14]. This graph-based routine was designed for AMGe and is suitable for 2D and 3D domains. Its motivation is that it should restore (up to boundary effects) regular 2D triangulations obtained by uniform refinement using either triangles or rectangular elements. In general, the algorithm provides compact agglomerated elements and tends to provide quasi-uniform coarse elements when the fine-grid triangulation is quasi-uniform. This results in reasonable complexities for the AMGe algorithms. Even though these tests use a version of the algorithm in [14] that does not take the operator into account in making its agglomeration decision, the results are nevertheless encouraging.

First consider linear elasticity equations (4.3) on a 32×32 regular grid of rectangular bilinear elements. The problem is discretized both with square ($h_x = h_y = 1/32$) and stretched elements possessing a $10:1$ aspect ratio ($h_x = 1/32, h_y = 1/320$). The agglomeration routine chooses 2×2 agglomerates in the interior of the domain and tends to choose irregular agglomerates for elements on the boundaries. Table 4.6 shows results for a domain partitioned with square and stretched elements. The Jones-Vassilevski routine chooses the same agglomeration pattern for both problems because both domains contain 32×32 elements.

For the single-element thick 2-D plane-stress cantilever beam, the agglomeration routine chooses 2×1 agglomerates on each level, the same agglomerates we predetermined for the experiments of Table 4.5. Hence, that table serves double duty– here as a report of the numerical results for tests using the algebraic agglomeration routine. The occurrence of the same agglomeration pattern is a product of the limited choices for aggregation for a single-element thick domain.

These results are very promising; we envision the possibility of designing an automatic, robust agglomeration routine for ρAMGe. The creation of such an algorithm is an active area of research.

5. Future work. Our goal in devising ρAMGe, as reflected in its rather simple theory, is to create a general approach to AMG 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 AMG methods to fulfill the needs of modern computational simulation.

Designers of algebraic methods such as ρAMGe must choose from a huge number of options. Undoubtedly, research into ρAMGe methods will continue and result in enhancements to the algorithm. As we have indicated, numerous choices must be made regarding agglomeration strategies, selection of staggered/unstaggered agglomerates, fuzzy agglomerate generation, employment of an accuracy/cost measure to guide agglomeration choices, and several other questions. While we have touched on these questions, much work remains to be done before they can be fully addressed. Still, we believe that ρAMGe shows promise in fulfilling its underlying motivation of providing an AMG solver that broadens the base of applicability of AMG 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] M. Brezina, I. Heberton, J. Mandel, and P. Vaněk, *An iterative method with convergence rate chosen a priori*, UCD/CCM Report 140, University of Colorado at Denver, February 1999.

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

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

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

[8] 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.

[9] J. Fish and V. Belsky, *Generalized aggregation multilevel solver*, International Journal for Numerical Methods in Engineering, 40 (1997), pp. 4341–4361.

[10] J. Fish, Y. Qu, and A. Suvorov, *Towards robust two-level methods for indefinite systems*, International Journal for Numerical Methods in Engineering, 45 (1999), pp. 1433–1456.

[11] J. Fish, A. Suvorov, and V. Belsky, *Automated adaptive multilevel solver*, Comp. Meth. Appl. Mech. Eng., 149 (1997), pp. 267–287.

[12] T. E. Giddings and J. Fish, *An algebraic two-level preconditioner for asymmetric, positive-definite systems*, International Journal for Numerical Methods in Engineering, 52 (2001), pp. 1443–1463.

[13] 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.

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

[15] V. Korneev and J. Fish, *Two-level methods for spatial problems based on aggregation*, Izv. VUZ. Matematika, 11 (2001), pp. 42–61.

[16] J. Mandel, *Balancing domain decomposition*, Communications on Numerical Methods in Engineering, 9 (1993), pp. 233–241.

[17] 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.