ELEMENT-FREE AMGe: GENERAL ALGORITHMS FOR COMPUTING INTERPOLATION WEIGHTS IN AMG

VAN EMDEN HENSON AND PANAYOT S. VASSILEVSKI

ABSTRACT. We propose a new general algorithm for constructing interpolation weights in algebraic multigrid (AMG). It exploits a proper extension mapping outside a neighborhood about a fine degree of freedom (dof) to be interpolated. The extension mapping provides boundary values (based on the coarse dofs used to perform the interpolation) at the boundary of the neighborhood. The interpolation value is then obtained by matrix dependent harmonic extension of the boundary values into the interior of the neighborhood.

We describe the method, present examples of useful extension operators, provide a two–grid analysis on model problems, and, by way of numerical experiments, demonstrate the successful application of the method to discretized elliptic problems.

1. INTRODUCTION

The classical algebraic multigrid (AMG) algorithm [2, 3, 9] was developed for operators represented by symmetric, positive-definite, M-matrices. While the algorithm works well for many real-world problems [10, 6, 11], there are situations in which it does not perform particularly well. One reason for this is that in some instances the classical definition of interpolation does not adequately interpolate the smooth modes of the error. More specifically, standard AMG interpolation makes certain assumptions about the nature of the smooth error which may not be valid for operators that are not M-matrices. A more sophisticated characterization of smooth error is required to develop an adequate interpolation formula.

To provide a better characterization of smooth error, a method known as AMGe, for element-based algebraic multigrid, was developed recently [4] for finite-element discretizations. AMGe provides an accurate interpolation formula by using the individual element stiffness matrices to construct a neighborhood matrix for each fine degree of freedom (dof). The sum of the individual stiffness matrices for all the elements containing the point at which the dof is defined, the neighborhood matrix acts as a local "Neumann"-type version of the original operator. According to AMGe theory, once the local matrix is developed and coarse-grid points are chosen, solving

Date: January 4, 2000–beginning; Today is January 16, 2001.

1991 *Mathematics Subject Classification.* 65F10, 65N20, 65N30.

Key words and phrases. algebraic multigrid, interpolation weights, sparse matrices, finite elements, unstructured meshes.

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 7000 East Avenue, Mail stop L-560, Livermore, CA 94550, email:{vhenson,panayot}@llnl.gov. This work was performed under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

a simple minimization problem yields the optimal interpolation operator for each dof. It is shown in [4] that the method indeed produces superior interpolation and leads to improved convergence rates on several types of problems, including both scalar problems and systems of PDEs, such as elasticity problems.

An obvious drawback to AMGe, naturally, is that it requires that the element stiffness matrices be available. While this is often the case, their storage can be expensive. Further, AMGe requires that coarse level elements be constructed and their individual stiffness matrices be available. Determining the coarse elements is a difficult and laborious task.

In this paper we examine the construction of the interpolation operator in both classical AMG and AMGe, and present them within a common framework. Our purpose is to extend and generalize the classical interpolation, which was originally motivated for M-matrices, to develop a rule applicable in more general settings. Accordingly, we propose a new method for determining the interpolation weights that attempts to capture the benefits of AMGe interpolation without requiring access to the individual element stiffness matrices. This method is applicable to finite difference, finite element, or finite volume discretizations, and we concentrate on the symmetric positive definite case. Essentially, it seeks to determine, for each fine dof, a neighborhood matrix that can be utilized in the same manner that the local assembled stiffness matrix is used in AMGe. We do this by defining a neighborhood for the fine dof and examining the rows of the original matrix that correspond to the points in that neighborhood. A set of exterior dofs is defined, and a mapping developed that extends functions on the neighborhood to the exterior dofs. This essentially imposes a set of boundary conditions on the neighborhood. Here we propose a unified way of building these boundary conditions. One may view them as an extension (extrapolation) of a vector defined on the neighborhood to its immediate exterior. This extension can be performed using constant vectors or any other vectors that may be of interest (such as the rigid body motions in elasticity problems). The extension can be built for each dof in the exterior based on the matrix sparsity pattern.

By incorporating the action of the extension operator into the local connections of the neighborhood, a modified local matrix is created. This matrix is then used in a manner similar to that employed in AMGe, that is, by solving a minimization problem, to create the interpolation operator. The construction of the extension operator and the respective minimization procedure to build the interpolation weights we consider as our main contribution. We give examples of several extension operators and show how they relate to both classical AMG and other, more recently proposed algorithms. A two–grid model analysis of the properties of the resulting interpolation mappings is provided as well. In particular, we prove that they exhibit approximately "harmonic" property as well as "partition of unity" property, desirable in standard two-grid analyses of the AMG methods.

Numerical results are presented demonstrating the method. We include both scalar problems and systems of PDEs in the form of elasticity problems. Finally, we draw some conclusions and comment on the direction that continued research will take.

It is important to note that while the choice of coarse-grid points, like the construction of the interpolation operator, is crucial to the success of the AMG method, we do not consider the coarse-grid selection here; rather, we leave that topic to future research while focusing on the interpolation problem here. Furthermore, we observe that neither AMGe nor the method proposed here are intended to replace or compete with classical AMG on problems characterized by simple M-matrices, such as the model Laplacian problem on a regular grid. Instead, they are intended for complicated problems, such as thin-body elasticity, posed typically on unstructured grids. Nonetheless, we apply the new method (and AMGe) to model problems because they illustrate, in simple fashion, the features of the methods. We therefore compare results of our method with AMGe on these problems, but do not include comparisons to classical AMG, which would be used in practice.

Some notational convention: to denote a vector we will use boldface, e.g., $\mathbf{v}, \mathbf{w}, \ldots$. The ith component of $\mathbf{v}$ will be denoted in different contexts as $\mathbf{v}(i)$, $v(i)$ or v_i. In the latter two cases v (i.e., not in boldface) will have a meaning of a "grid" function.

2. A framework for AMG interpolation

Assume that the problem $A\mathbf{x} = \mathbf{f}$ is to be solved, where A is a sparse, symmetric, positive-definite matrix. AMG is a multigrid method in which no geometric grid information is used (and often isn't available or doesn't even exist). Accordingly, all of the components of a multigrid algorithm, the hierarchy of grids, interpolation and restriction operators, and the coarse-grid versions of the original operator, must be constructed using only the information contained in the entries of A. For any multigrid algorithm, several basic components are required. In the case of AMG, they can be described as follows:

- A fine grid is required. For AMG, this is generally a set D comprising the degrees of freedom of the original problem.
- A coarse grid D_c is necessary. This set of dofs is typically a subset of D.
- An interpolation operator P is necessary to map vector functions defined on the coarse grid D_c to the fine grid D, $P: \ D_c \longrightarrow D$.
- A restriction operator $R : D \longrightarrow D_c$, mapping fine-grid functions to the coarse grid, is needed. For AMG the restriction is frequently defined by $R = P^T$, and we will use that definition here.
- A coarse-grid version of the original operator A is needed. For AMG the coarse operators are generally defined by the Galerkin relation $A_c = P^T AP$.
- A smoothing iteration, G, is used. It is typical to use a point-relaxation method such as Gauß-Seidel or Jacobi relaxation.

The basic two-grid algorithm can then be described as follows: Begin with an initial approximation $\mathbf{x}_0$ to the solution of $A\mathbf{x} = \mathbf{f}$.

1. Smooth the error by $\mathbf{x}_0 \leftarrow G(A, \mathbf{f}, \mathbf{x}_0)$.
2. Compute the residual $\mathbf{r} = \mathbf{f} - A\mathbf{x}_0$.
3. Restrict the residual to the coarse grid $\mathbf{f}_c = R\mathbf{r}$.
4. Solve the coarse-grid residual equation $\mathbf{e}_c = A_c^{-1}\mathbf{f}_c$.
5. Interpolate the coarse-grid error to the fine grid and correct the fine-grid approximation $\mathbf{x}_0 \leftarrow \mathbf{x}_0 + P\mathbf{e}_c$.

For a multigrid method, a hierarchy of coarse grids $D \equiv D_0 \supset D_1 \supset D_2 \supset \ldots \supset D_J$ is present, and the two-grid algorithm is applied recursively, i.e., the two-grid algorithm is repeated each time step 4 is encountered, except that a direct solve is employed at the coarsest-grid.

For the multigrid method an interpolation operator P_i is required mapping functions on each grid (level i) to the next finer grid (level $i-1$). Unlike many conventional (geometric) multigrid algorithms, in AMG the interpolation operators are rarely the same for different levels. Similarly, the Galerkin relation is employed to define versions of the original operator A on all levels, thusly: $A_{i+1} \equiv R_i A_i P_i$.

There are many ways in which to select the coarse-grid dofs in AMG [9, 11, 5]. Commonly, the coarse set D_c is a maximally independent subset of D, but this is not required. We will not discuss the question of coarse-grid selection further, except to note that each fine-grid dof i is connected to its nearest neighbors (e.g., j) by way of having a nonzero coefficient a_{ij}, and that the value of a interpolated function at i is typically a weighted average of the values of its nearest neighbors that are coarse-grid dofs. For the remainder of this paper, we shall simply assume that a coarse grid has been selected and that the coarse neighbors are known for any fine dof.

With this description of the basic components of AMG, we can describe a simple framework for computing the entries of the interpolation operator. Let $i \in D$ be a fine-grid dof whose value is to be interpolated. We first define a subset $\Omega(i) \subset D$ to be the *neighborhood* of i. For now we place no particular restrictions on what dofs can be in $\Omega(i)$. For example, the set $\Omega(i)$ could consist of i and all of its nearest neighbors, or i and its nearest coarse neighbors, or i, its neighbors, and all of their neighbors. Indeed, within the framework we describe here, the exact character of the interpolation operator will depend largely on what sort of neighborhood is defined. Since the value at i will be interpolated from coarse points in the neighborhood, it is useful to denote the set of coarse dofs in the neighborhood to be $\Omega_c(i)$.

To construct the interpolation for i, we examine the entries of the operator A in the following way. We begin, without loss of generality, by permuting the rows and columns of A and partitioning it so the first set of rows and columns corresponds to i and the fine dofs in the neighborhood, that is, to $\Omega(i)\backslash\Omega_c(i)$. The next set of rows and columns corresponds to the coarse neighbors $\Omega_c(i)$, while the final set of rows and columns corresponds to the rest of the grid $D\backslash\Omega(i)$. Hence the partitioning of A, along with the identity of the rows corresponding to the partitions, appears as

$$A = \begin{pmatrix} A_{ff} & A_{fc} & * \\ * & * & * \\ * & * & * \end{pmatrix} \begin{array}{l} \}\ \Omega(i)\backslash\Omega_c(i) \\ \}\ \Omega_c(i) \\ \}\ D\backslash\Omega(i). \end{array}$$

For our purposes we are only concerned with two blocks of the partitioned matrix. The block A_{ff} gives the connections among i and the fine-grid neighbors while the block A_{fc} links i and the fine neighbors to the coarse neighbors.

Armed with these concepts of neighborhood partitioning of the operator, we can examine classical AMG, AMGe, and our proposed method in terms of choices of neighborhood and definition of the neighborhood matrices.

3. Interpolation in Classical AMG

For classical AMG [9], the interpolation is computed in the following fashion. The neighborhood $\Omega(i)$ is defined to be the dof i and all dofs connected to it (all j for which $a_{ij} \neq 0$). We then replace both A_{ff} and A_{fc} with modified versions, $\widehat{A}_{ff}$ and $\widehat{A}_{fc}$, respectively.

A_{ff} is modified in two ways. Let S_i denote the set of dofs that are *strongly connected* to the dof i, C_i denote the set of C-points in the neighborhood $\Omega(i)$, and let W_i denote the dofs that are *weakly connected* to the dof i. (By strongly (weakly) connected we mean that the magnitude of a_{ij} is greater (smaller) than some pre-defined threshold. A common choice is that if the magnitude of a_{ij} is less than θ times the largest magnitude of all off-diagonal entries in the ith row then j is considered to be weakly connected to i.) Then we modify the row of A_{ff} corresponding to the dof i (which we will hereafter refer to as the ith row, regardless of the actual numerical ordering) by

$$\widehat{a}_{ii} = a_{ii} + \sum_{j \in W_i} a_{ij} \tag{3.1}$$

$$\widehat{a}_{i,j} = \begin{cases} 0 & j \in W_i, \\ a_{i,j} & j \in S_i. \end{cases} \tag{3.2}$$

For $j \in W_i$ we replace the jth row of A_{ff} by a zero row, and then place a 1 in column j and -1 in column i. For all other rows of A_{ff}, i.e., for $j \in S_i$, we zero out the off-diagonal entries, and replace the diagonal entry a_{jj} with

$$\widehat{a}_{jj} = -\sum_{k \in C_i} a_{j,k}.$$

The block A_{fc} is modified to $\widehat{A}_{fc}$ by zeroing the jth row of for $j \in W_i$.

Once the modified blocks $\widehat{A}_{ff}$ and $\widehat{A}_{fc}$ are computed, the entries of the ith row of the interpolation matrix P are determined by taking the entries of the ith row of the matrix

$$-\left(\widehat{A}_{ff}^{-1}\widehat{A}_{fc}\right).$$

4. Interpolation in AMGe

For AMGe a similar description of the interpolation is easily given. In this setting, the neighborhood $\Omega(i)$ is defined naturally as the union of all finite elements having i as a vertex (Figure 1). In the figure, the set $\Omega(i)$ consists of all vertices in the shaded region, including i (the open circle in the center). The shaded region consists of the six triangular finite elements having i as a vertex. Members of $\Omega_c(i)$ are indicated by the square vertices. Since AMGe gives us access to the individual element stiffness matrices, we may create a neighborhood matrix $A_{\Omega(i)}$ simply by summing together all the individual element stiffness matrices of the elements in the neighborhood. In AMGe the interpolation operator for the dof i is determined by solving a constrained min-max problem, that is, by finding interpolation coefficients that minimize a certain measure from finite element theory. The solution to the min-max problem can be

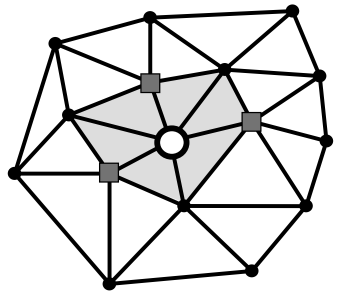

FIGURE 1. *The neighborhood of the fine dof i (large open circle).*

computed in several ways, one of which fits into the framework we are developing here. We partition the neighborhood matrix into the rows and columns associated with the fine dofs in the neighborhood and the rows and columns associated with the coarse dofs, as

$$A_{\Omega(i)} = \begin{pmatrix} A_{ff} & A_{fc} \\ * & * \end{pmatrix} \quad \begin{matrix} \} \ \Omega(i) \setminus \Omega_c(i) \\ \} \ \Omega_c(i). \end{matrix}$$

Again, our only interest is in the rows of the neighborhood matrix corresponding to the fine dofs, including i. With this partitioning, it turns out that one way to solve the min-max problem is to take, as the coefficients for the interpolation operator for i, the entries of the ith row of the matrix

$$-\left(A_{ff}^{-1} A_{fc}\right).$$

It is useful to note that, unlike the classical AMG case, there is no need to modify the matrix A_{ff} prior to computing the interpolation coefficients. Essentially, this is because the element stiffness matrices have built in to them local versions of the null space and near-null space of the operator; we do not need to make alterations to the local matrices to insure that these spaces are represented.

For many problems the AMGe method produces a superior interpolation, and results in good convergence rates [4]. In the remainder of this paper our goal is to accomplish a superior interpolation without the knowledge (and hence, expense) of the individual stiffness matrices.

5. Interpolation for Element-free AMGe

The process we propose for building the interpolation operator is very similar to the processes described for AMG and AMGe. Once again, we will proceed by defining a neighborhood of the fine dofs and an associated neighborhood matrix. Let ψ be a set of fine dofs whose values we wish to interpolate. We define $\Omega(\psi)$ to be the neighborhood of ψ, which includes the coarse dofs that will be used to interpolate the dofs in ψ. The set of coarse dofs in the neighborhood we denote $\Omega_c(\psi)$.

Now, however, we define a third set of dofs

$$\Omega_{\mathcal{X}}(\psi) = \{ j \notin \Omega(\psi) \mid a_{ij} \neq 0 \text{ for some } i \in \Omega(\psi) \setminus \Omega_c(\psi) \}.$$

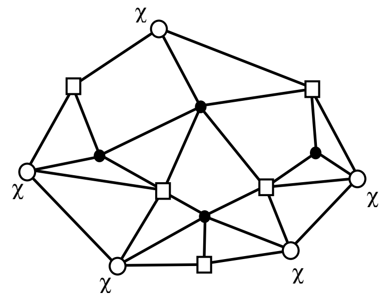

FIGURE 2. *The extended neighborhood $\overline{\Omega}(\psi)$, including the fine dofs to be interpolated (solid circles), the coarse interpolatory set $\Omega_c(\psi)$ (squares), and the extension dofs (open circles marked $\mathcal{X}$).*

That is, $\Omega(\psi)$ can be viewed as the interior of the set $\overline{\Omega}(\psi) \equiv \Omega(\psi) \cup \Omega_{\mathcal{X}}(\psi)$. Figure 2 gives an example of such a neighborhood.

We begin the construction of a neighborhood matrix by examining the rows of the matrix A that correspond to the fine dofs in $\Omega(\psi)$; that is, we will be concerned with the following partitioning of A:

$$A = \begin{pmatrix} A_{ff} & A_{fc} & A_{f\mathcal{X}} & 0 \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \quad \begin{array}{l} \} \ \Omega(\psi) \setminus \Omega_c(\psi) \\ \} \ \Omega_c(\psi) \\ \} \ \Omega_{\mathcal{X}}(\psi) \\ \} \ \text{everything else on grid.} \end{array}$$

5.1. **Local (neighborhood) quadratic form.** Our task next is to define a matrix associated with ψ that yields a local version of the operator A, performing the same function as does the neighborhood matrix in AMGe. To do this we first build an extension mapping (matrix) $E(\psi)$ that maps a vector defined on $\Omega(\psi)$ to $\overline{\Omega}(\psi)$

$$E(\psi) : \begin{pmatrix} \mathbf{v}_f \\ \mathbf{v}_c \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{v}_f \\ \mathbf{v}_c \\ \mathbf{v}_{\mathcal{X}} \end{pmatrix}$$

using the relation

$$\mathbf{v}_{\mathcal{X}} = E_{\mathcal{X}f}(\psi)\mathbf{v}_f + E_{\mathcal{X}c}(\psi)\mathbf{v}_c.$$

That is, the extension operator looks like

$$E = \begin{pmatrix} I & 0 \\ 0 & I \\ E_{\mathcal{X}f}(\psi) & E_{\mathcal{X}c}(\psi) \end{pmatrix}.$$

For now we will not be specific about the exact nature of the extension operator. Rather, we will describe how it may be used to develop an interpolation formula, after which we shall discuss desirable properties of the operator.

We construct a neighborhood matrix from the first block of rows of the partitioned matrix

$$\left(\widehat{A}_{ff}, \widehat{A}_{fc}\right) = (A_{ff}, A_{fc}, A_{f\mathcal{X}}) \begin{pmatrix} I & 0 \\ 0 & I \\ E_{\mathcal{X}f}(\psi) & E_{\mathcal{X}c}(\psi) \end{pmatrix}$$

so that

$$\widehat{A}_{ff} = A_{ff} + A_{f\mathcal{X}}E_{\mathcal{X}f}(\psi) \qquad \text{and} \qquad \widehat{A}_{fc} = A_{fc} + A_{f\mathcal{X}}E_{\mathcal{X}c}(\psi).$$

For any vector $\begin{bmatrix} \mathbf{v}_f \\ \mathbf{v}_c \end{bmatrix}$, consider its extension $\mathbf{v} = \begin{bmatrix} \mathbf{v}_f \\ \mathbf{v}_c \\ \mathbf{v}_{\mathcal{X}} \end{bmatrix}$, where $\mathbf{v}_{\mathcal{X}}$ is given by $\mathbf{v}_{\mathcal{X}} = E_{\mathcal{X}f}(\psi)\mathbf{v}_f + E_{\mathcal{X}c}(\psi)\mathbf{v}_c$. Let

$$\widehat{\mathbf{v}} = \begin{bmatrix} -A_{ff}^{-1}\left(A_{fc}\mathbf{v}_c + A_{f\mathcal{X}}\mathbf{v}_{\mathcal{X}}\right) \\ \mathbf{v}_c \\ \mathbf{v}_{\mathcal{X}} \end{bmatrix}$$

be the so–called harmonic extension of $\mathbf{v}|_{\Omega_c(\psi)\cup\Omega_{\mathcal{X}}(\psi)}$ into $\Omega(\psi) \setminus \Omega_c(\psi)$. That is, one extends $\mathbf{v}$, restricted to the "boundary" $\Omega_c(\psi) \cup \Omega_{\mathcal{X}}(\psi)$, into the "interior" $\Omega(\psi) \setminus \Omega_c(\psi)$.

We use the $\mathbf{v}_f$ that minimizes the difference $\mathbf{v} - \widehat{\mathbf{v}}$ in energy norm in the interpolation procedure. Since

$$\mathbf{v} - \widehat{\mathbf{v}} = \begin{bmatrix} \mathbf{v}_f - (\widehat{\mathbf{v}})_f \\ 0 \\ 0 \end{bmatrix},$$

its energy norm is computable and equals

$$\begin{aligned} ||\mathbf{v} - \widehat{\mathbf{v}}||_A^2 &= (\mathbf{v}_f - (\widehat{\mathbf{v}})_f)^T A_{ff}(\mathbf{v}_f - (\widehat{\mathbf{v}})_f) \\ &= (\mathbf{v}_f + A_{ff}^{-1}\left(A_{fc}\mathbf{v}_c + A_{f\mathcal{X}}\mathbf{v}_{\mathcal{X}}\right))^T A_{ff}(\mathbf{v}_f + A_{ff}^{-1}\left(A_{fc}\mathbf{v}_c + A_{f\mathcal{X}}\mathbf{v}_{\mathcal{X}}\right)). \end{aligned}$$

Since A_{ff} is positive definite, this implies that if we solve the equation

$$\begin{aligned} 0 &= A_{ff}\mathbf{v}_f + (A_{fc}\mathbf{v}_c + A_{f\mathcal{X}}\mathbf{v}_{\mathcal{X}}) \\ &= (A_{ff} + A_{f\mathcal{X}}E_{\mathcal{X}f})\mathbf{v}_f + (A_{fc} + A_{f\mathcal{X}}E_{\mathcal{X}c})\mathbf{v}_c \\ &= \widehat{A}_{ff}\mathbf{v}_f + \widehat{A}_{fc}\mathbf{v}_c, \end{aligned}$$

the minimization of $||\mathbf{v}_f - (\widehat{\mathbf{v}})_f||_A$ is attained with zero minimum by $\mathbf{v}_f = -\widehat{A}_{ff}^{-1}\widehat{A}_{fc}\mathbf{v}_c$.

We can actually show (see Remark 7.1 and Lemma 7.1) that in the model finite element case considered in Section 7 the minimization procedure is equivalent to a quadratic functional minimization involving Neumann assembled matrices, as in the AMGe method (cf., [4]).

It is natural to ask whether $\widehat{A}_{ff}$ is invertible. If $E_{\mathcal{X}f} = 0$, there is no difficulty, since then $\widehat{A}_{ff} = A_{ff}$. In general, if $E_{\mathcal{X}f}$ is sufficiently small in norm $A_{ff} + A_{f\mathcal{X}}E_{\mathcal{X}f}$ will be invertible.

6. Examples of extension operators

We describe here four extension operators E that can be used to construct the interpolation operator in the element-free approach. These are by no means all the useful extensions that we could concoct; they form, however, a simple set of examples that will allow us to demonstrate the efficacy of the method and its underlying philosophy.

The first we call the L_2-*extension* because it is a simple averaging method. Given $\mathbf{v}$ defined on $\Omega(i)$, we wish to extend it to $\mathbf{v}_{\mathcal{X}}$, defined on $\Omega_{\mathcal{X}}(\psi)$. Suppose that $i_{\mathcal{X}}$ is an exterior dof, that is, a point from $\Omega_{\mathcal{X}}(\psi)$ whose value we wish to determine from the values of the dofs in $\Omega(i)$. Let $S = \{j \in \Omega(i) : a_{i_{\mathcal{X}},j} \neq 0\}$; that is, S comprises those dofs in $\Omega(i)$ to which the point $i_{\mathcal{X}}$ is connected. It seems natural to consider using a simple average over these dofs as the extension at $i_{\mathcal{X}}$. Thus, the extension formula, for the dof $i_{\mathcal{X}}$, is given by

$$\mathbf{v}_{\mathcal{X}}(i_{\mathcal{X}}) = \frac{1}{\displaystyle\sum_{j \in S} 1} \sum_{j \in S} \mathbf{v}(j).$$

A somewhat more sophisticated extension we call the A-*extension* because it is a simple operator-induced method. The A-extension operator for the dof $i_{\mathcal{X}}$ is given by the formula

$$\mathbf{v}_{\mathcal{X}}(i_{\mathcal{X}}) = \frac{1}{\displaystyle\sum_{j \in S} |a_{i_{\mathcal{X}},j}|} \sum_{j \in S} \left(|a_{i_{\mathcal{X}},j}| \mathbf{v}(j)\right).$$

It may be seen that in this case the extension to the exterior is a simple weighted average of the values of the neighborhood dofs to which the exterior point is connected. The weights in the average are given by the absolute values of the matrix coefficients.

The two methods just described share the property that they are computed point-by-point. That is, the extension formulas for the dofs in $\Omega_{\mathcal{X}}(\psi)$ are determined independently. A second feature shared by the methods is that if the neighborhood vector $\mathbf{v}$ is constant, then the extended values are also constant, and have the same value as the entries of the neighborhood vector. This feature is clearly desirable for many elliptic PDEs, where the constant vector is in the null space or near-null space of the operator A.

The third example we describe is based on the minimization of a quadratic functional. Again, let $\mathbf{v}$ be a vector defined on $\Omega(i)$ that we wish to extend to $\Omega_{\mathcal{X}}(\psi)$. We construct the extension to be that operator which produces $\mathbf{v}_{\mathcal{X}}$ that minimizes the functional $Q(\mathbf{v}_{\mathcal{X}})$, where

$$Q(\mathbf{v}_{\mathcal{X}}) = \sum_{\substack{i_{\mathcal{X}} \in \Omega_{\mathcal{X}}(\psi) \\ j \in \Omega(i)}} |a_{i_{\mathcal{X}},j}| \left(v_{i_{\mathcal{X}}} - v_j\right)^2. \tag{6.1}$$

It is evident that, like the previous extension operators, if $\mathbf{v}$ is constant on $\Omega(i)$ then the dofs in $\Omega_{\mathcal{X}}(\psi)$ will also have the same constant value. Unlike the previous extension operators, which are determined one dof at a time, this is a "simultaneous" extension, computing formulas for extending to all of the exterior dofs together. As such, it is necessarily more expensive to compute. We also note that this extension, and the interpolation it generates, is equivalent to the method recently proposed in [1].

A final example is given by minimizing the following "cut–off" quadratic functional:

$$(\theta \mathbf{v})^T A_{\overline{\Omega}(\psi)}(\theta \mathbf{v}) \mapsto \min$$

where $\overline{\Omega}(\psi) \equiv \Omega(\psi) \cup \Omega_{\mathcal{X}}(\psi)$, subject to $\mathbf{v}_f,\ \mathbf{v}_c$ fixed. Here

$$\theta = \begin{bmatrix} I & 0 \\ 0 & \theta_{\mathcal{X}} \end{bmatrix} \begin{array}{l} \} \ \Omega(\psi) \\ \} \ \Omega_{\mathcal{X}}(\psi) \end{array}$$

is a diagonal matrix. A good choice is, a diagonal matrix $\theta_{\mathcal{X}}$ formed from the vector

$$\underline{\theta}_{\mathcal{X}} = -(A_{\mathcal{X}\mathcal{X}})^{-1} \left[A_{\mathcal{X}f},\ A_{\mathcal{X}c} \right] \begin{bmatrix} (1)_f \\ (1)_c \end{bmatrix}.$$

Here we used the blocks of A corresponding to its $\Omega_{\mathcal{X}}(\psi)$ rows.

It is easily seen that the extension mapping is actually defined as

$$\begin{aligned} \mathbf{v}_{\mathcal{X}} &= E_{\mathcal{X}c}\mathbf{v}_c + E_{\mathcal{X}f}\mathbf{v}_f \\ &= -\theta_{\mathcal{X}}^{-1}(A_{\mathcal{X}\mathcal{X}})^{-1} \left[A_{\mathcal{X}f},\ A_{\mathcal{X}c} \right] \begin{bmatrix} \mathbf{v}_f \\ \mathbf{v}_c \end{bmatrix}. \end{aligned}$$

Note that this extension mapping is also a simultaneous extension operator and an averaging one; i.e., if $\mathbf{v}_c = (1)_c$ and $\mathbf{v}_f = (1)_f$, then $\mathbf{v}_{\mathcal{X}} = (1)_{\mathcal{X}}$.

6.1. **Classical AMG as an extension method.** The interpolation method of the classical AMG algorithm popularized by Ruge and Stüben [9] may be viewed as an extension method. Here the neighborhood is just the dof to be interpolated together with the dofs that will be used to compute the interpolated value. That is, $\Omega(i) = \{i\} \cup \Omega_c(i)$. The extended neighborhood then includes all fine dofs that are connected to i,

$$\Omega_{\mathcal{X}}(\psi) = \{ j \notin \Omega(i) \ :\ a_{ij} \neq 0 \}.$$

An A-extension is defined in the following manner. For each $i_{\mathcal{X}} \in \Omega_{\mathcal{X}}(\psi)$, set $v_{i_{\mathcal{X}}} = v_i$ if $i_{\mathcal{X}}$ is weakly connected to i (Recall that in classical AMG, as developed for M-matrices, the dof i is said to be strongly connected to the dof j if

$$-a_{ij} > \theta \max_{k \neq i}(-a_{ik})$$

where θ is a user-specified parameter, and weakly-connected otherwise). If $i_{\mathcal{X}}$ is strongly connected to i the extension is defined by

$$v_{i_{\mathcal{X}}} = \frac{1}{\displaystyle\sum_{j \in \Omega_c(\psi)} a_{i_{\mathcal{X}},j}} \sum_{j \in \Omega_c(\psi)} a_{i_{\mathcal{X}},j} v_j.$$

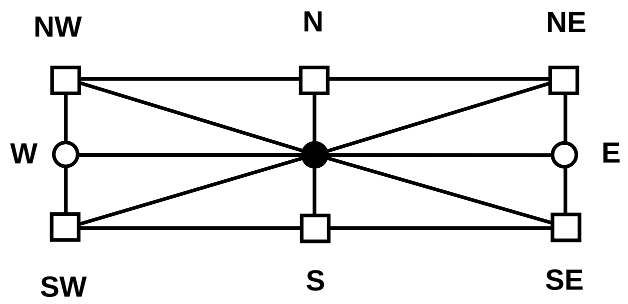

FIGURE 3. *The neighborhood of the fine dof i (large solid circle) for the stretched quadrilateral element problem. The problem is semicoarsened; squares denote the coarse neighbors* $\Omega_c(i)$ *while the open circles are the exterior points* $\Omega_{\mathcal{X}}(\psi)$.

6.2. **An Example of the Extensions.** A simple example should suffice to illustrate these extension methods. Suppose the problem $-U_{xx} - U_{yy} = f(x, y)$ is discretized using a regular Cartesian grid of points making up the vertices of quadrilateral elements. Suppose further that the elements had dimension $h_x \times h_y$ where $h_x \gg h_y$. As $h_y/h_x \to 0$ the operator stencil tends toward

$$\begin{bmatrix} -1 & -4 & -1 \\ 2 & 8 & 2 \\ -1 & -4 & -1 \end{bmatrix}.$$

Since there is effectively no coupling between a given point and its neighbors to the east or west, the appropriate choice is to semicoarsen, selecting every other line of points with constant y-coordinate to be coarse points. Using the same logic, the natural interpolation is to have each fine dof interpolated only using the values to the north and south of it, each with equal weighting of 1/2. Consider the interpolation of one point, i, shown in the center of its neighborhood in Figure 3. For either the L_2- or A-extensions, we might select $\Omega(i) = \{i\} \cup \Omega_c(i)$ where, in this instance, $\Omega_c(i) = \{N, S, SW, NW, SE, NE\}$. Then $\Omega_{\mathcal{X}}(\psi) = \{W, E\}$. We see then that $A_{ff} = [8]$, $A_{fc} = [\ -4\ \ -4\ \ -1\ \ -1\ \ -1\ \ -1\ ]$, and $A_{f\mathcal{X}} = [\ 2\ \ 2\ ]$. For the A-extension it is easy to compute the extension operators

$$E_{\mathcal{X}c} = \frac{1}{12}\begin{pmatrix} 1 & 1 & 4 & 4 & & \\ 1 & 1 & & & 4 & 4 \end{pmatrix} \qquad \text{and} \qquad E_{\mathcal{X}f} = \frac{1}{12}\begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

from which

$$\widehat{A}_{ff} = \left(\frac{104}{12}\right) \qquad \text{and} \qquad \widehat{A}_{fc} = \frac{1}{3}\begin{pmatrix} -11 & -11 & -1 & -1 & -1 & -1 \end{pmatrix}$$

which yields an interpolation operator

$$P_A = \frac{1}{26}\begin{pmatrix} 11 & 11 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

We see that the values to the north and south are used in the interpolation with weights $11/26 \approx 0.423$ and that the four points diagonally adjacent to i all are weighted $1/26 \approx 0.038$. The ideal weights, of course, are 0.5 and 0, respectively, so the interpolation weights computed by the A-extension method, while quite good, are not perfect.

A similar calculation for the weights using the L_2-extension yields the interpolation operator

$$P_{L_2} = \frac{1}{44}\left(16 \quad 16 \quad 3 \quad 3 \quad 3 \quad 3 \right).$$

Here the dofs to the north and south are weighted $16/44 \approx 0.364$ while the diagonally adjacent dofs are weighted by $3/44 \approx 0.068$. For this problem, then, the A-extension is significantly better than the L_2-extension.

By contrast, it is a straightforward calculation to show that classical AMG produces the interpolation operator

$$P_{AMG} = \frac{1}{12}\left(4 \quad 4 \quad 1 \quad 1 \quad 1 \quad 1 \right),$$

where the north and south dofs are weighted by $4/12 \approx 0.333$ and the diagonally adjacent dofs are weighted by $1/12 \approx 0.083$; these weights are farther from the ideal than the weights produced by either the A- or L_2-extension.

Finally consider the extension operator based on minimizing the "cut–off" quadratic functional. The additional matrix blocks involved read:

$$\begin{aligned} A_{\mathcal{X}\mathcal{X}} &= \begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}, \\ A_{\mathcal{X}f} &= \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \\ A_{\mathcal{X}c} &= \begin{bmatrix} -1 & -1 & -4 & -4 & 0 & 0 \\ -1 & -1 & 0 & 0 & -4 & -4 \end{bmatrix}. \end{aligned}$$

The vector $\underline{\theta}_{\mathcal{X}} = -A_{\mathcal{X}\mathcal{X}}^{-1}[A_{\mathcal{X}f}\ A_{\mathcal{X}c}] \begin{bmatrix} (1)_f \\ (1)_c \end{bmatrix} = (1)_{\mathcal{X}}$. This is seen as follows

$$A_{\mathcal{X}f}(1)_f = 2(1)_{\mathcal{X}},\ A_{\mathcal{X}c}(1)_c = -10(1)_{\mathcal{X}},$$

and hence

$$A_{\mathcal{X}f}(1)_f + A_{\mathcal{X}c}(1)_c = -8(1)_{\mathcal{X}},$$

which implies

$$\underline{\theta}_{\mathcal{X}} = -A_{\mathcal{X}\mathcal{X}}^{-1}(A_{\mathcal{X}f}(1)_f + A_{\mathcal{X}c}(1)_c) = -\frac{1}{8}[-8(1)_{\mathcal{X}}] = (1)_{\mathcal{X}}.$$

That is, the diagonal matrix θ is the identity and hence the extension matrices then read:

$$\begin{aligned} E_{\mathcal{X}f} &= -A_{\mathcal{X}\mathcal{X}}^{-1}A_{\mathcal{X}f} &&= -\tfrac{1}{4}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \\ E_{\mathcal{X}c} &= -A_{\mathcal{X}\mathcal{X}}^{-1}A_{\mathcal{X}c} &&= \tfrac{1}{8}\begin{bmatrix} 1 & 1 & 4 & 4 & 0 & 0 \\ 1 & 1 & 0 & 0 & 4 & 4 \end{bmatrix}. \end{aligned}$$

The modified matrices $\widehat{A}_{ff}$ and $\widehat{A}_{fc}$ take the form:

$$
\begin{aligned}
\widehat{A}_{ff} &= A_{ff} + A_{f\mathcal{X}} E_{\mathcal{X}f} &&= 8 - [2,\ 2]\tfrac{1}{4}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
& &&= 7, \\
\widehat{A}_{fc} &= A_{fc} + A_{f\mathcal{X}} E_{\mathcal{X}c} &&= [-4, -4, -1, -1, -1, -1] + [2,\ 2]\tfrac{1}{8}\begin{bmatrix} 1 & 1 & 4 & 4 & 0 & 0 \\ 1 & 1 & 0 & 0 & 4 & 4 \end{bmatrix} \\
& &&= [-4, -4, -1, -1, -1, -1] + [\tfrac{1}{2}, \tfrac{1}{2}, 1, 1, 1, 1] \\
& &&= [-\tfrac{7}{2}, -\tfrac{7}{2}, 0, 0, 0, 0].
\end{aligned}
$$

That is, the interpolation coefficients are the "perfect" ones (corresponding to semi-coarsening):

$$\Big((-\widehat{A}_{ff})^{-1}\widehat{A}_{fc}\Big)_i = [\frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0].$$

7. Two–grid analysis for a model finite element problem

Before providing numerical results, we present an analysis of the quality of the "element-free AMGe" interpolation. That is, we prove an "approximate" harmonic property of the interpolation mapping and show that it provides a partition of unity. Specifically, we assume that the problem is a standard finite element discretization of a second order elliptic problem

$$a(u,\ v) \equiv \int a(x)\nabla u \cdot \nabla v \; dx = (f,\ v) \quad v \in V,$$

where V is a finite element space of piecewise linear functions over quasiuniform triangular elements that cover a given 2–d polygonal domain. For simplicity, we assume that homogeneous Neumann boundary conditions are imposed and that $(f,\ 1) = 0$ (to insure solvability).

Let us denote, for any element e,

$$\varrho(e) = \sup_{x\in e} \max_{\underline{\xi}\in\mathbb{R}^2} \frac{\underline{\xi}^T a(x)\underline{\xi}}{\underline{\xi}^T\underline{\xi}} \cdot \tag{7.1}$$

In the following, we assume that the differential operator coefficients are essentially constant in each element, so that $\varrho(e)$ gives rise to the local ellipticity constant.

Further, we assume (only for simplicity) that the neighborhood $\overline{\Omega}(i) \equiv \Omega(i) \cup \Omega_{\mathcal{X}}(i)$ for any fine dof i is formed by union of triangles that share dof i as a common vertex. Thus we will use i instead of ψ denoting the neighborhoods ($\Omega(i)$, $\Omega_{\mathcal{X}}(i)$, and $\Omega_c(i)$) and the extension mappings. In particular, we denote $E_i = [E_{\mathcal{X}f},\ E_{\mathcal{X}c}]$ where for brevity $E_{\mathcal{X}f} = E_{\mathcal{X}f}(i)$ and $E_{\mathcal{X}c} = E_{\mathcal{X}c}(i)$. A closer look at the analysis to follow, however, shows that it applies as well to more general (i.e., larger) neighborhoods.

In what follows, for any subdomain (union of triangles) G, we let $a_G(.,.)$ denote the bilinear form a restricted to G. The corresponding subdomain matrix (assembled from the individual element matrices A_e) will be denoted by A_G^N. We omit the superscript N when there is no confusion between A_G^N and A_G, the submatrix of the original matrix A (corresponding to G). Note that in the latter case A_G corresponds to a matrix with homogeneous Dirichlet boundary conditions imposed on $\partial(G \cup \{\text{elements neighboring } G\})$.

For this discussion we assume that E_i, the local extension mapping used to build the interpolation coefficients, is based on averaging, although no specific rule is assumed. We do, however, assume that $E = E(i)$ has the particular form

$$\begin{bmatrix} I & 0 \\ 0 & I \\ 0 & E_{\mathcal{X}c} \end{bmatrix} \begin{array}{l} \}\Omega(i) \\ \}\Omega_c(i) \\ \}\Omega_{\mathcal{X}}(i) \end{array}$$

That is, $E_{\mathcal{X}f} = 0$ and $E_i = [0,\ E_{\mathcal{X}c}]$.

Remark 7.1. *The general case of $E_i = [E_{\mathcal{X}f},\ E_{\mathcal{X}c}]$ can be reduced to the particular case above by using the modified extension mapping $\widehat{E}_i = \left[0,\ \widehat{E}_{\mathcal{X}c}\right]$ where*

$$\widehat{E}_{\mathcal{X}c} = E_{\mathcal{X}f}\left(-\widehat{A}_{ff}^{-1}\widehat{A}_{fc}\right) + E_{\mathcal{X}c}.$$

To see this, recall that $\widehat{A}_{ff} = A_{ff} + A_{f\mathcal{X}}E_{\mathcal{X}f}$ and $\widehat{A}_{fc} = A_{fc} + A_{f\mathcal{X}}E_{\mathcal{X}c}$, and note that the modified extension mapping extends a constant vector defined on $\Omega_c(i)$ to be the same constant on $\Omega_{\mathcal{X}}(i)$, that is,

$$\begin{aligned} \widehat{E}_{\mathcal{X}c}(1)_c &= -E_{\mathcal{X}f}\widehat{A}_{ff}^{-1}\widehat{A}_{fc}(1)_c + E_{\mathcal{X}c}(1)_c \\ &= E_{\mathcal{X}f}(1)_f + E_{\mathcal{X}c}(1)_c \\ &= (1)_{\mathcal{X}}. \end{aligned}$$

Here we have used the fact that since (for the model second order elliptic problem) $A_{ff}(1)_f + A_{f\mathcal{X}}(1)_{\mathcal{X}} + A_{fc}(1)_c = 0$ then $A_{ff}(1)_f + A_{f\mathcal{X}}\left(E_{\mathcal{X}f}(1)_f + E_{\mathcal{X}c}(1)_c\right) + A_{fc}(1)_c = 0$. That is, $\widehat{A}_{ff}(1)_f + \widehat{A}_{fc}(1)_c = 0$, implying that $(1)_f = -\widehat{A}_{ff}^{-1}\widehat{A}_{fc}(1)_c$.

We still must show that the modified extension mapping $\widehat{E}_i$ leads to the same interpolation as does E_i, i.e., that

$$-A_{ff}^{-1}\left(A_{f\mathcal{X}}\widehat{E}_{\mathcal{X}c} + A_{fc}\right) = -\widehat{A}_{ff}^{-1}\widehat{A}_{fc}.$$

For this we observe that

$$\begin{aligned} -A_{ff}^{-1}(A_{f\mathcal{X}}\widehat{E}_{\mathcal{X}c} + A_{fc}) &= -A_{ff}^{-1}\left[A_{f\mathcal{X}}\left(E_{\mathcal{X}f}(-\widehat{A}_{ff}^{-1}\widehat{A}_{fc}) + E_{\mathcal{X}c}\right) + A_{fc}\right] \\ &= -A_{ff}^{-1}\left[A_{f\mathcal{X}}E_{\mathcal{X}c} + A_{fc} - A_{f\mathcal{X}}E_{\mathcal{X}f}\widehat{A}_{ff}^{-1}\widehat{A}_{fc}\right] \\ &= -A_{ff}^{-1}\left[\widehat{A}_{fc} - A_{f\mathcal{X}}E_{\mathcal{X}f}\widehat{A}_{ff}^{-1}\widehat{A}_{fc}\right] \\ &= -A_{ff}^{-1}\left[\widehat{A}_{ff} - A_{f\mathcal{X}}E_{\mathcal{X}f}\right]\widehat{A}_{ff}^{-1}\widehat{A}_{fc} \\ &= -A_{ff}^{-1}\left(A_{ff}\right)\widehat{A}_{ff}^{-1}\widehat{A}_{fc} \\ &= -\widehat{A}_{ff}^{-1}\widehat{A}_{fc}. \end{aligned}$$

◇

Consider the minimization problem

$$\text{(7.2)} \qquad \text{find } v_f \text{ such that } \begin{bmatrix} v_f \\ v_c \\ E_i v \end{bmatrix}^T A^N_{\overline{\Omega}(i)} \begin{bmatrix} v_f \\ v_c \\ E_i v \end{bmatrix} = \inf_{\substack{w:\, w_c = v_c \\ w_{\mathcal{X}} = E_i w}} a_{\overline{\Omega}(i)}(w,\ w).$$

Thus we seek v_f, the value of w on $\Omega(i) \setminus \Omega_c(i)$, which minimizes the quadratic form $a_{\overline{\Omega}(i)}(w,\ w)$ when the values of w are fixed at the coarse points and are "slave" at the exterior points ($\Omega_{\mathcal{X}}(i)$); that is, they are extrapolated from the interior $\Omega(i)$ and the coarse points $\Omega_c(i)$ by $E_i w$.

Lemma 7.1. *The solution to the minimization problem* (7.2) *produces the same interpolation coefficients as does element-free AMGe, namely, those given by* $-A_{ff}^{-1}(A_{f\mathcal{X}}E_{\mathcal{X}c} + A_{fc})$*. That is, the minimizer is given by* $w_f = v_f \equiv -A_{ff}^{-1}(A_{f\mathcal{X}}E_{\mathcal{X}c} + A_{fc})v_c$*.*

Proof. Consider the Neumann matrix

$$A^N_{\overline{\Omega}(i)} = \begin{bmatrix} A_{ff} & A_{fc} & A_{f\mathcal{X}} \\ A_{cf} & A^N_{cc} & A^N_{c\mathcal{X}} \\ A_{\mathcal{X}f} & A^N_{\mathcal{X}c} & A^N_{\mathcal{X}\mathcal{X}} \end{bmatrix}.$$

We use the superscript "N" for the blocks which differ from the corresponding blocks of $A_{\overline{\Omega}(i)}$, the principal submatrix of the original matrix A corresponding to the subdomain $\overline{\Omega}(i)$. Note that the "N" blocks are not accessible (available) and not used in our algorithm. We have $E_i v|_{\Omega_{\mathcal{X}}(i)} = E_{\mathcal{X}c}v_c$. Hence, $a_{\overline{\Omega}(i)}(w,\ w)$ for $w_c = v_c$ and $w_{\mathcal{X}} = E_i w|_{\Omega_{\mathcal{X}}(i)}$ leads to the following matrix expression:

$$\begin{aligned} a_{\overline{\Omega}(i)}(w,\ w) &= \begin{bmatrix} w_f \\ v_c \\ E_{\mathcal{X}c}v_c \end{bmatrix}^T \begin{bmatrix} A_{ff} & A_{fc} & A_{f\mathcal{X}} \\ A_{cf} & A^N_{cc} & A^N_{c\mathcal{X}} \\ A_{\mathcal{X}f} & A^N_{\mathcal{X}c} & A^N_{\mathcal{X}\mathcal{X}} \end{bmatrix} \begin{bmatrix} w_f \\ v_c \\ E_{\mathcal{X}c}v_c \end{bmatrix} \\ &= \begin{bmatrix} w_f \\ v_c \end{bmatrix}^T \begin{bmatrix} A_{ff} & A_{fc} + A_{f\mathcal{X}}E_{\mathcal{X}c} \\ A_{cf} + E^T_{\mathcal{X}c}A_{\mathcal{X}f} & A^N_{cc} + A^N_{c\mathcal{X}}E_{\mathcal{X}c} + E^T_{\mathcal{X}c}(A^N_{\mathcal{X}c} + A^N_{\mathcal{X}\mathcal{X}}E_{\mathcal{X}c}) \end{bmatrix} \\ &\qquad \times \begin{bmatrix} w_f \\ v_c \end{bmatrix}. \end{aligned}$$

Minimizing this symmetric positive semi–definite quadratic form with respect to w_f is equivalent to solving the equation

$$A_{ff}w_f + (A_{fc} + A_{f\mathcal{X}}E_{\mathcal{X}c})v_c = 0,$$

which is the same equation that specifies v_f in the element–free AMGe interpolation procedure. □

In the next lemma we will remove the constraint on v being fixed at the $\Omega_{\mathcal{X}}(i)$ points.

Lemma 7.2. *The following quadratic forms are spectrally equivalent,*

$$q_1(v_c,\ v_c) \equiv \inf_{v:\ v|_{\Omega_c(i)} = v_c} a_{\overline{\Omega}(i)}(v,\ v), \quad \textit{and} \quad q_2(v_c, v_c) \equiv \inf_{\substack{v:\ v_c \textit{ fixed} \\ v_{\mathcal{X}} = E_i v}} a_{\overline{\Omega}(i)}(v,\ v).$$

That is, there exists a positive constant η such that

$$q_1(v_c, v_c) \le q_2(v_c, v_c) \le \eta\, q_1(v_c, v_c) \quad \textit{for all } v_c.$$

Proof. We show the proof for two-dimensional domains. For other domains, we must scale ϱ by the local mesh size appropriately; otherwise, the proof remains the same. It suffices to show that the two quadratic forms have the same null–space. The null–space of q_1 is $v_c = \text{const}$ and the null-space of q_2 is the same as that of $a_{\overline{\Omega}(i)}(v,\ v)$ with $v:\ v_c = \text{const}$ and $E_i v = \text{const}$ on $\Omega_{\mathcal{X}}(i)$. Note that $a_{\overline{\Omega}(i)}(v,\ v) = 0$ implies v_f is the same constant as v_c. Then $E_i v$ is also the same constant, since it is an averaging operator based on the values of v_c and v_f. Hence the forms q_1 and q_2 both vanish only for constant v_c. In order to show that the constant η is bounded independently of E_i, one first easily sees that

$$a_{\overline{\Omega}(i)}(v, v) \le C \sum_{e \subset \overline{\Omega}(i)} \varrho(e) \sum_{l,\, k \in e} (v(l) - v(k))^2.$$

The constant C depends only on the number of points used in the averaging procedure (E_i), i.e., it is bounded by the total number of coarse points $\Omega_c(i)$ (plus the interior point i). The dofs l and k in the summation are either coarse dofs or i, and $\varrho(e)$ is defined in (7.1) to be the maximal value of the local ellipticity bound associated with the original elliptic operator coefficient $a(x)$. More specifically, for each $i_{\mathcal{X}} \in \Omega_{\mathcal{X}}(i)$

$$v(i_{\mathcal{X}}) \equiv (E_i v)(i_{\mathcal{X}}) = \sum_{k \in \Omega_c(i) \cup \{i\}} \alpha_{i_{\mathcal{X}},\, k} v(k),$$

where

$$\sum_{k \in \Omega_c(i) \cup \{i\}} \alpha_{i_{\mathcal{X}},\, k} = 1, \quad \text{and} \quad \alpha_{i_{\mathcal{X}},\, k} \ge 0.$$

Then, for any $j \in \Omega_c(i) \cup \{i\}$,

$$(v(i_{\mathcal{X}}) - v(j)) = \sum_{k \in \Omega_c(i) \cup \{i\}} \alpha_{i_{\mathcal{X}},\, k} (v(k) - v(j)),$$

and hence

$$\begin{aligned} (v(i_{\mathcal{X}}) - v(j))^2 \; &\le \sum_{k \in \Omega_c(i) \cup \{i\}} \alpha^2_{i_{\mathcal{X}},\, k} \sum_{k \in \Omega_c(i) \cup \{i\}} (v(k) - v(j))^2 \\ &\le \sum_{k \in \Omega_c(i) \cup \{i\}} (v(k) - v(j))^2. \end{aligned}$$

As a result we see that, for $v_{\mathcal{X}} = E_i v$,

$$\begin{aligned} q_2(v_c,\ v_c) &\le a_{\overline{\Omega}(i)}(v,\ v) \\ &\le C \sum_{e\subset\overline{\Omega}(i)} \varrho(e) \sum_{l,k\in e} (v(l)-v(k))^2 \\ &\le C\, \frac{\max\limits_{e\in\overline{\Omega}(i)} \varrho(e)}{\min\limits_{e\in\overline{\Omega}(i)} \varrho(e)} \sum_{e\subset\overline{\Omega}(i)} \varrho(e) \sum_{k,j\in e\cap(\Omega_c(i)\cup\{i\})} (v(k)-v(j))^2. \end{aligned}$$

Finally, since v_f is arbitrary on the right–hand side of this inequality,

$$q_2(v_c,\ v_c) \le C\, \frac{\max\limits_{e\in\overline{\Omega}(i)} \varrho(e)}{\min\limits_{e\in\overline{\Omega}(i)} \varrho(e)} \inf_{v_f} \left(\sum_{e\subset\Omega_c(i)} \varrho(e) \sum_{k,j\in e\cap(\Omega_c(i)\cup\{i\})} (v(k)-v(j))^2 \right).$$

It is also true that

$$q_1(v_c, v_c) \simeq \inf_{v:\, v_c \text{ fixed}} \left(\sum_{e\subset\overline{\Omega}(i)} \varrho(e) \sum_{l,\, k\in e} (v(l)-v(k))^2 \right).$$

This shows that η can be chosen bounded independently of the actual averaging extension mapping E_i. □

The above estimates involve the factor

$$\frac{\max\limits_{e\in\overline{\Omega}(i)} \varrho(e)}{\min\limits_{e\in\overline{\Omega}(i)} \varrho(e)}.$$

Whether it is large or small depends on the selection of the coarse grid (the coarse grid reflects the form of the neighborhood $\Omega(i)$), which we do not consider in the present paper.

Then the following corollary, involving the element-free AMGe interpolated vector Pv_c, is proved in the same way as Lemma 7.2.

Corollary 7.1. *Consider the extended neighborhood of i, $\widehat{\Omega}(i) = \cup\{e,\ e\subset\overline{\Omega}(i)$ or $e\subset\overline{\Omega}(j)$, for all $j\in\Omega_{\mathcal{X}}(i)\}$. There is a constant $\kappa = \kappa_{\hat{\Omega}(i)} > 0$, locally estimated, such that the following bound holds:*

$$a_{\overline{\Omega}(i)}(Pv_c,\ Pv_c) \le \kappa \inf_{w:\, w_c=v_c} a_{\hat{\Omega}(i)}(w,\ w).$$

Proof. Let v be defined on $\widehat{\Omega}(i)$ as follows:

$$v(k) = \begin{cases} (Pv_c)(k), & k\in\overline{\Omega}(i), \\ v_c(k), & k \text{ is a coarse dof outside } \overline{\Omega}(i), \\ (E_j v)(k), & k\in\Omega_{\mathcal{X}}(j), \text{ for some } j\in\Omega_{\mathcal{X}}(i). \end{cases}$$

We see that v at every fine dof k in $\widehat{\Omega}(i)$ is an average value of some neighboring coarse dofs from $\widehat{\Omega}(i)$. Hence, in the same way as in the proof of Lemma 7.2, we establish the inequality

$$a_{\hat{\Omega}(i)}(v, v) \leq \kappa \inf_{w:\, w_c = v_c} a_{\hat{\Omega}(i)}(w,\ w).$$

Since $a_{\overline{\Omega}(i)}(Pv_c,\ Pv_c) \leq a_{\hat{\Omega}(i)}(v, v)$, the desired result follows. □

For each fine dof i, define $\mathcal{Z}(i)$ to be the number of overlapping domains on $\hat{\Omega}(i)$, that is, the number of domains $\hat{\Omega}(j)$ such that $\hat{\Omega}(j) \cap \hat{\Omega}(i) \neq \emptyset$. Then we may state the following theorem.

Theorem 7.1. *The element–free interpolation mapping P exhibits the following approximate harmonic property:*

$$a(Pv_c,\ Pv_c) \leq \kappa \inf_{w:\, w_c = v_c} a(w,\ w),$$

where the constant $\kappa = \max_{i=\text{fine dof}} \kappa_{\hat{\Omega}(i)} \mathcal{Z}(i)$, and the $\kappa_{\hat{\Omega}(i)}$ are the local constants from Corollary 7.1.

Proof. The proof simply follows from the fact that

$$a(Pv_c,\ Pv_c) \leq \sum_{i=\text{fine dof}} a_{\overline{\Omega}(i)}(Pv_c,\ Pv_c)$$

and by summation of the local estimates from Corollary 7.1. □

Another important property of the element–free interpolation mapping P is that it partitions unity, as we show in the following theorem.

Theorem 7.2. *P provides a partition of unity. Specifically, the row sums of P are 1.*

Proof. Let $v_f = Pv_c$ be given by $v_f = \sum_{i_c \in \Omega_c(i)} \alpha_{i,\, i_c} v_c(i_c)$. Assume that $v_c(i_c) = 1$ on $\Omega_c(i)$. Now, P uses the formula that minimizes (7.2) and the minimum (zero) is achieved for $E_i v(j) = 1$ at $\Omega_{\mathcal{X}}(i)$ and $v_f = 1$. That is, we find that

$$1 = \sum_{i_c \in \Omega_c(i)} \alpha_{i,\, i_c},$$

which is the desired unity row–sum property of P. □

Remark 7.2. *Theorems 7.1 and 7.2 are the main goals of many two–grid convergence analyses and they imply convergence of the respective two–grid AMG methods, cf., e.g.,* [14], [8], [13], *and* [7].

8. Numerical experiments

We describe here several sets of numerical experiments designed to test the efficacy of the element-free AMGe methods described above. For each of several problems we apply a set of interpolation rules within an AMG code. The problems are then solved using a CG solver, preconditioned with one V-cycle of AMG.

The interpolation rules are:

- the AMGe rule [7] for the finite element problems;

- three element–free AMGe rules from Section 6:
  1. L_2–extension;
  2. A–extension;
  3. (only for scalar PDE) the simultaneous extension based on minimizing the quadratic functional (6.1) described in Section 6.

For systems problems the unknowns are split into physical variables. That is, for scalar problems the rule is as described in Section 6, while for 2-d elasticity, with physical variables u and v (displacement in the x- and y-directions, respectively) we perform the extensions (and associated interpolation) of exterior dofs of type u using only neighborhood dofs of type u; similarly, the extension to exterior dofs of type v are carried out using neighborhood dofs of type v; this applies both to L_2 and A–extensions. The local neighborhood about a point is defined by the sparsity pattern of the matrix about that point and the averaging involves only dofs from the sparsity pattern set S (see Section 6).

8.1. **An elliptic problem on a triangular element mesh.** We apply the various interpolation rules to a second order elliptic PDE

$$-\nabla \cdot (A(x,y)\nabla u) = f(x,y) \quad \text{on } G \tag{8.1}$$

$$u(x,y) = g(x,y) \quad \text{on } \partial G \tag{8.2}$$

where G is the unit square. The matrix of diffusion coefficients includes functions with relatively benign characteristics– there is both spatial variability and jump discontinuity in the coefficients, but the jumps are of relatively small magnitude and the variation is mild. The discretization is by a finite-element method on an unstructured triangular mesh. The coarsening algorithm is one of element agglomeration. That is, the coarse grids are the vertices of coarse elements produced by an agglomeration algorithm proposed in [7]. Figure 4 displays the coarsening sequence for a typical problem. Here the fine grid comprises 1600 elements, the first coarse grid has 382 elements, and the remaining grids have 93, 33, 15, 7, 3, and 1 elements. Table 1 gives the coarsening details for four different versions of this problem. It may be seen that the number of elements decreases by about 75% at each coarsening for the first few coarsenings, after which it decreases by about 50% per level. The number of nonzero entries in the matrix decreases by approximately 50% per level, while the number of degrees of freedom tends to decrease by 50-60% with each successive level.

For each of the four interpolation rules, the problem is solved using a preconditioned conjugate gradient method, where the preconditioning consists of a single V(1,1)-cycle of AMG, with a Gauß-Seidel smoother. The iteration is run until the residual is less than 10^{-8} in norm. We report the results in Table 2. For each problem size we display, for each interpolation rule, the number of preconditioned CG iterations required to achieve the desired residual size and ϱ, the average convergence factor over the iterations.

Examination of the results reveals that all three of the extension methods, A-extension, L_2-extension, and quadratic functional minimization, perform at least as well on this problem as does AMGe. In some cases the performance of the extension methods is marginally better than AMGe. The amount of work entailed for

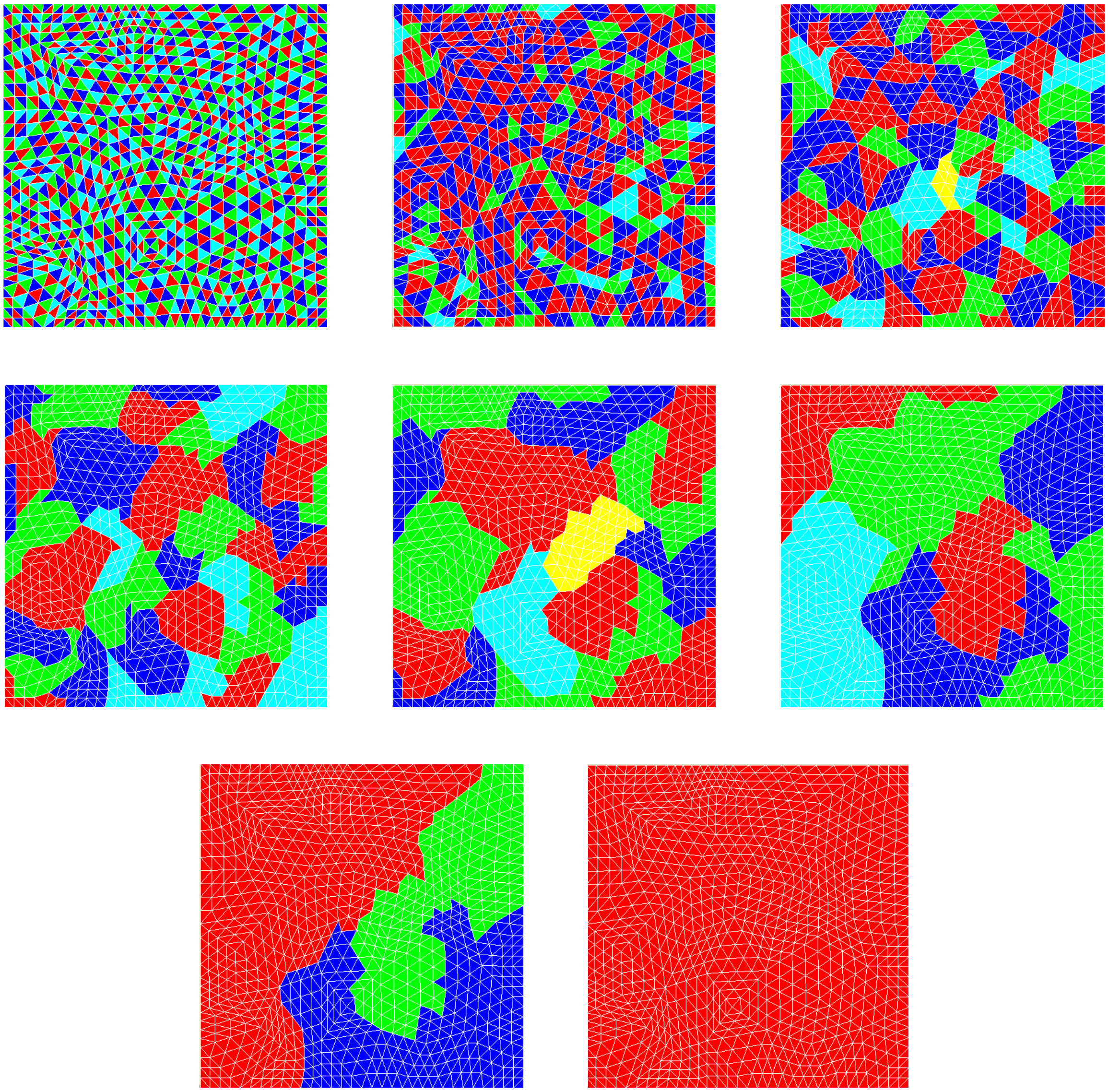

FIGURE 4. *Sequence increasingly coarse elements, formed by element agglomeration.*

the A-extension and the L_2-extension methods is comparable to that of AMGe, provided that the neighborhoods are selected to be of comparable size as the element neighborhoods (which is the case in these experiments). For the quadratic functional minimization the work is somewhat greater, but still comparable. The advantage of the element-free methods is, of course, that there is no requirement to have the actual individual stiffness matrices that are required in AMGe. For this experiment this represents a considerable savings in storage.

8.2. **Two dimensional elasticity, the thin beam.** We consider next the two dimensional plane-stress elasticity problem on a cantilevered beam, fixed at one end. The domain of the problem is $G = (0, 1) \times (0, \ d)$ with $d \leq 1$. For $d \ll 1$ this is the

TABLE 1. *Coarsening history for the problem $-\nabla\cdot A(x.y)\nabla u = f$ on an unstructured triangular fine grid. For each level of each problem size, "nz" is the number of nonzero entries in the operator matrix, "dofs" gives the number of degrees of freedom, and "elts" gives the number of finite elements in the agglomerated grid.*

		No. of elements			
level		25600	6400	1600	400
0	nz	90321	22761	5781	1491
	dofs	13041	3321	861	231
	elts	25600	6400	1600	400
1	nz	32898	9540	2602	1094
	dofs	4108	1152	330	114
	elts	6013	1427	382	76
2	nz	14305	4361	1397	470
	dofs	1507	451	143	50
	elts	1489	374	93	26
3	nz	7193	2098	634	199
	dofs	643	198	64	23
	elts	392	117	33	11
4	nz	3458	975	304	88
	dofs	302	91	32	12
	elts	158	47	15	5
5	nz	1580	453	126	36
	dofs	140	45	16	6
	elts	70	22	7	2
6	nz	714	188	46	16
	dofs	68	22	8	4
	elts	33	10	3	1
7	nz	274	84	16	
	dofs	30	12	4	
	elts	14	5	1	
8	nz	120	30		
	dofs	16	6		
	elts	7	2		
9	nz	42	16		
	dofs	8	4		
	elts	3	1		
10	nz	16			
	dofs	4			
	elts	1			

thin-beam problem. The problem is

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

TABLE 2. *CG convergence results; unstructured triangular fine grid; second order elliptic problem; $V(1,1)$-cycle MG, Gauß–Seidel smoother used as preconditioner.*

Interp. rule		400 elts	1600 elts	6400 elts	25600 elts
AMGe	iterations	14	16	21	23
	ϱ	0.115	0.172	0.252	0.289
A-extension	iterations	13	15	19	20
	ϱ	0.118	0.158	0.218	0.247
L_2-extension	iterations	13	16	19	21
	ϱ	0.119	0.161	0.227	0.249
quadratic funct.	iterations	13	15	19	19
min.	ϱ	0.105	0.152	0.222	0.231

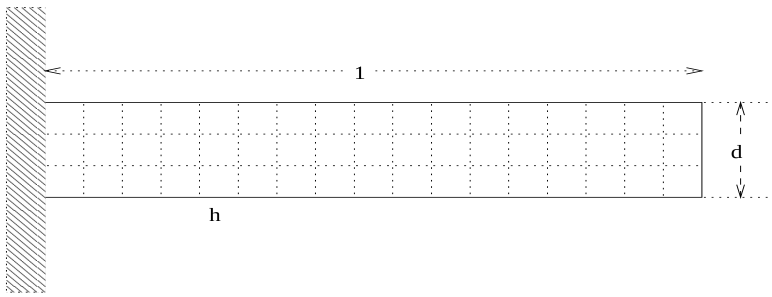

FIGURE 5. *The thin-beam elasticity problem domain. Homogeneous Dirichlet boundary conditions are applied at $x = 0$.*

where u and v are displacements in the x and y directions, respectively. This can be a difficult problem for standard multigrid methods, especially when the domain is long and thin [6]. The problem is discretized using uniform square finite elements of size h. Nodal coarsening is used, with the coarse nodes being the vertices of elements created by the agglomeration algorithm from [7]. After certain levels of coarsening the algorithm agglomerates only along the x–direction.

We present results in both the thick beam ($d = 1.0$) and thin beam ($d = 0.05$) cases. In both cases we use $\nu = 1/3$. For each case we present results for three sizes of the discretization parameter: $h = 0.05, 0.025$, and 0.0125 for the thick beam and and $h = 0.025, 0.0125$, and 0.00625 for the thin beam. The coarsening histories of the agglomeration algorithm are shown in Table 3. Table 4 shows the results of the experiments for the beam problem. As in section 8.1, preconditioned conjugate gradient is used as the solver, with a single V(1,1)-cycle of AMG as the preconditioner, with a Gauß-Seidel smoother. For this problem we show the number of iterations required to achieve a residual norm less than 10^{-8}, and also the convergence factor of the final iteration. For this problem we do not implement the quadratic minimization method described in section 6. That method is for scalar problems, while this problem is a system of PDEs. We use the AMGe method described in [4] and compare it with the A- and L_2-extension methods described above. Our expectation is that AMGe

TABLE 3. *Coarsening history; structured rectangular fine grid; 2–d elasticity, $d = 1$*

		Thick Beam $d = 1.0$			Thin Beam $d = 0.05$		
level		$h = 0.050$	$h = 0.025$	$h = 0.0125$	$h = 0.025$	$h = 0.0125$	$h = 0.00625$
0	nz	14884	58564	232324	3388	12532	48100
	dofs	882	3362	13122	246	810	2898
1	nz	10440	40880	161760	1664	7328	30656
	dofs	264	924	3444	88	252	820
2	nz	4128	17248	70488	784	3744	10152
	dofs	84	264	924	44	132	252
3	nz	1000	4956	19056	384	1152	3816
	dofs	32	94	284	24	48	132
4	nz	256	1404	6128	144	384	1152
	dofs	16	38	104	12	24	48
5	nz	64	324	1668	64	144	384
	dofs	8	18	42	8	12	24
6	nz		144	576		64	144
	dofs		12	24		8	12
7	nz		64	144			64
	dofs		8	12			8
8	nz			64			
	dofs			8			

should outperform the element-free methods, at least on the thin beam problem; this is the problem for which AMGe was originally developed. We observe, however, that for the thick beam problems the element-free methods both outperform AMGe. First, we note that it takes fewer iterations to reach the tolerance. It is also apparent that the element-free methods are more scalable, in that the number of iterations does not grow with the problem size. The AMGe method requires more iterations for larger problems.

For the thin beam problem, we observe the results we naturally expect. That is, AMGe outperforms the element-free methods, requiring fewer iterations. Further, AMGe appears to be more scalable on this problem than the extension methods. The L_2-extension method exhibits a distinct lack of scalability as the problem grows larger.

9. Conclusions

In this paper we propose a general rule for building interpolation weights in AMG, thus extending the applicability of AMG to more general settings than the traditional M–matrix case. The applications include elliptic problems on unstructured finite element grids, where both scalar problems and systems (like elasticity) are considered. The element-free AMGe method seems as competitive as the AMGe methods but

TABLE 4. CG convergence results; structured rectangular fine grid; 2-d elasticity, $d = 1$, $V(1,1)$-cycle MG, Gauß–Seidel smoother used as preconditioner.

Thick Beam, $d = 1.0$				
Interp. rule		$h = 0.050$	$h = 0.025$	$h = 0.0125$
AMGe	iterations	16	18	20
	ϱ	0.172	0.206	0.234
A-extension	iterations	12	12	12
	ϱ	0.099	0.098	0.097
L_2-extension	iterations	13	13	13
	ϱ	0.101	0.102	0.104
Thin Beam, $d = 0.05$				
Interp. rule		$h = 0.025$	$h = 0.0125$	$h = 0.00625$
AMGe	iterations	17	18	19
	ϱ	0.180	0.198	0.22
A-extension	iterations	20	23	22
	ϱ	0.227	0.286	0.280
L_2-extension	iterations	18	20	27
	ϱ	0.203	0.243	0.254

entail much less overhead. The element information and the element matrices, in particular, are essential for the AMGe methods but are not required for element-free AMGe. If we assume more information is available (such as the rigid body modes in the case of elasticity) it may be incorporated into the construction of the extension mappings. Thus element–free AMGe can be made to reproduce the extra modes in the interpolation from their coarse values. This property is important in the AMG methods for elasticity problems (cf. [12]), and incorporating it into element-free AMGe is a subject of ongoing research.

REFERENCES

[1] A. BRANDT, *Generally highly accurate algebraic coarsening*, Elec. Trans. Num. Anal., 10 (2000), pp. 1–20.

[2] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid (AMG) for automatic multigrid solutions with application to geodetic computations.* Report, Inst. for Computational Studies, Fort Collins, Colo., October 1982.

[3] ——, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, 1984.

[4] M. BREZINA, A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM Journal on Scientific Computing, 22 (2000), pp. 1570–1592.

[5] A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, AND J. E. JONES, *Coarse-grid selection for parallel algebraic multigrid*, in Proceedings of the Fifth International Symposium on Solving Irregularly Structured Problems in Parallel, vol. 1457 of Lecture Notes in Computer Science, Springer-Verlag, 1998, pp. 104–115.

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

[7] J. E. Jones and P. S. Vassilevski, *Amge based on element agglomeration*, SIAM Journal on Scientific Computing, (to appear).

[8] J. Mandel, M. Brezina, and P. Vaněk, *Energy optimization of algebraic multigrid bases*, Computing, (to appear).

[9] J. W. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.

[10] K. Stüben, *Algebraic multigrid (AMG): experiences and comparisons*, Appl. Math. Comput., 13 (1983), pp. 419–452.

[11] ———, *A review of algebraic multigrid.* GMD Report 69, November 1999.

[12] P. Vanek, M. Brezina, and R. Tezuar, *Two-grid method for linear elasticity on unstructured meshes*, SIAM J. Sci. Comput., (to appear).

[13] P. Vaněk, J. Mandel, and M. Brezina, *Convergence of algebraic multigrid based on smoothed aggregation.* UCD/CCM Report 126, Center for Computational Mathematics, University of Colorado at Denver, February 1998. http://www-math.cudenver.edu/ccmreports/rep126.ps.gz.

[14] W. L. Wan, T. F. Chan, and B. Smith, *An energy-minimizing interpolation for robust multigrid methods*, SIAM J. Sci. Comput., (to appear).