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 of model problems, and, by way of numerical experiments, demonstrate the successful application of the method to discretized elliptic problems.

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

AMS subject classifications. 65F10, 65N20, 65N30

PII. S1064827500372997

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 neighborhood matrix—the sum of the individual stiffness matrices for all the elements containing the point at which the dof is defined—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 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

---

*Received by the editors June 6, 2000; accepted for publication (in revised form) January 10, 2001; published electronically July 10, 2001. This work was performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.


†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 7000 East Avenue, Mail stop L-560, Livermore, CA 94550 (vhenson@llnl.gov, panayot@llnl.gov).
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. We consider the construction of the extension operator and the respective minimization procedure to build the interpolation weights 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” properties as well as “partition of unity” properties, desirable in standard two-grid analyses of the AMG methods.

Numerical results are presented to demonstrate 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 conventions follow: to denote a vector we will use boldface, e.g., \( \mathbf{v}, \mathbf{w}, \ldots \). The \( i \)th component of \( \mathbf{v} \) will be denoted in different contexts as \( v(i) \), \( v^{(i)} \), 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 \( \mathbf{A}\mathbf{x} = \mathbf{f} \) is to be solved, where \( \mathbf{A} \) 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 \( \mathbf{A} \). 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 \( \mathbf{P} \) is necessary to map vector functions defined on the coarse grid \( D_c \) to the fine grid \( D, \mathbf{P} : D_c \rightarrow D \).
- A restriction operator \( \mathbf{R} : D \rightarrow D_c \), mapping fine-grid functions to the coarse grid, is needed. For AMG the restriction is frequently defined by \( \mathbf{R} = \mathbf{P}^T \), and we will use that definition here.
- A coarse-grid version of the original operator \( \mathbf{A} \) is needed. For AMG the coarse operators are generally defined by the Galerkin relation \( \mathbf{A}_c = \mathbf{P}^T \mathbf{A} \mathbf{P} \).
- A smoothing iteration, \( \mathbf{G} \), 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 \( \mathbf{A}\mathbf{x} = \mathbf{f} \).

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

For a multigrid method, a hierarchy of coarse grids \( D \equiv D_0 \supset D_1 \supset D_2 \supset \cdots \supset D_J \) 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 \( \mathbf{A} \) on all levels, thusly: \( A_{i+1} \equiv R_i^T A_i P_i \).

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 \( \Omega_{c}(i) \) 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 \( \Omega(i) \setminus \Omega_{c}(i) \). 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 \( D \setminus \Omega(i) \). Hence the partitioning of \( A \), along with the identity of the rows corresponding to the partitions, appears as

\[
A = \begin{pmatrix}
A_{ff} & A_{fc} & * \\
* & * & * \\
* & * & * \\
\end{pmatrix}
\]

For our purposes we are concerned only 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, \( \hat{A}_{ff} \) and \( \hat{A}_{fc} \), respectively.

\( A_{ff} \) is modified in two ways. Let \( S_{i} \) denote the set of dofs that are strongly connected to the dof \( i \), let \( C_{i} \) 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 \( \theta \) times the largest magnitude of all off-diagonal entries in the \( i \)th 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 \( i \)th row, regardless of the actual numerical ordering) by

\[
\hat{a}_{ii} = a_{ii} + \sum_{j \in W_{i}} a_{ij},
\]

\[
\hat{a}_{i,j} = \begin{cases} 
0, & j \in W_{i}, \\
 a_{i,j}, & j \in S_{i}.
\end{cases}
\]
For $j \in W_i$ we replace the $j$th 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

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

The block $A_{fc}$ is modified to $\hat{A}_{fc}$ by zeroing the $j$th row for $j \in W_i$. Once the modified blocks $\hat{A}_{ff}$ and $\hat{A}_{fc}$ are computed, the entries of the $i$th row of the interpolation matrix $P$ are determined by taking the entries of the $i$th row of the matrix

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

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 4.1). 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 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

$$A_{\Omega(i)} = \begin{pmatrix}
A_{ff} & A_{fc} \\
* & *
\end{pmatrix}_{\Omega(i)},$$

$$\Omega_c(i).$$

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 $i$th row of the matrix

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

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 into 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 ensure 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 \( \psi \) be a set of fine dofs whose values we wish to interpolate. We define \( \Omega(\psi) \) to be the neighborhood of \( \psi \), which includes the coarse dofs that will be used to interpolate the dofs in \( \psi \). The set of coarse dofs in the neighborhood we denote \( \Omega_c(\psi) \).

Now, however, we define a third set of dofs:

\[
\Omega_X(\psi) = \{ j \notin \Omega(\psi) \mid a_{ij} \neq 0 \text{ for some } i \in \Omega(\psi) \setminus \Omega_c(\psi) \},
\]

That is, \( \Omega(\psi) \) can be viewed as the interior of the set \( \Omega(\psi) \equiv \Omega(\psi) \cup \Omega_X(\psi) \). Figure 5.1 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 \( A \):

\[
A = \begin{pmatrix}
A_{ff} & A_{fc} & A_{fx} & 0 \\
* & * & * & * \\
* & * & * & * \\
* & * & * & *
\end{pmatrix}
\]

\[
\begin{aligned}
\Omega(\psi) \setminus \Omega_c(\psi), \\
\Omega_c(\psi), \\
\Omega_X(\psi), \\
\text{everything else on grid.}
\end{aligned}
\]

5.1. Local (neighborhood) quadratic form. Our next task is to define a matrix associated with \( \psi \) 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 \( \Omega(\psi) \),

\[
E(\psi) : \begin{pmatrix} v_f \\ v_c \end{pmatrix} \longrightarrow \begin{pmatrix} v_f \\ v_c \\ v_X \end{pmatrix},
\]

using the relation

\[
v_X = E_{Xf}(\psi)v_f + E_{Xc}(\psi)v_c.
\]

That is, the extension operator looks like

\[
E = \begin{pmatrix}
I & 0 \\
0 & I \\
E_{Xf}(\psi) & E_{Xc}(\psi)
\end{pmatrix}.
\]
Fig. 5.1. The extended neighborhood $\Omega(\psi)$, including the fine dofs to be interpolated (solid circles), the coarse interpolatory set $\Omega_c(\psi)$ (squares), and the extension dofs (open circles marked $X$).

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

$$\left( \hat{A}_{ff}, \hat{A}_{fc} \right) = (A_{ff}, A_{fc}, A_{fX}) \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ E_{Xf}(\psi) & E_{Xc}(\psi) \end{pmatrix}$$

so that

$$\hat{A}_{ff} = A_{ff} + A_{fX} E_{Xf}(\psi) \quad \text{and} \quad \hat{A}_{fc} = A_{fc} + A_{fX} E_{Xc}(\psi).$$

For any vector $[\begin{smallmatrix} v_f \\ v_c \\ v_X \end{smallmatrix}]$, consider its extension

$$v = \begin{bmatrix} v_f \\ v_c \\ v_X \end{bmatrix},$$

where $v_X$ is given by $v_X = E_{Xf}(\psi)v_f + E_{Xc}(\psi)v_c$. Let

$$\tilde{v} = \begin{bmatrix} -A_{ff}^{-1} (A_{fc} v_c + A_{fX} v_X) \\ v_c \\ v_X \end{bmatrix}$$

be the so-called harmonic extension of $v|_{\Omega_c(\psi) \cup \Omega_X(\psi)}$ into $\Omega(\psi) \setminus \Omega_c(\psi)$. That is, one extends $v$, restricted to the “boundary” $\Omega_c(\psi) \cup \Omega_X(\psi)$, into the “interior” $\Omega(\psi) \setminus \Omega_c(\psi)$.

We use the $v_f$ that minimizes the difference $v - \tilde{v}$ in energy norm in the interpolation procedure. Since

$$v - \tilde{v} = \begin{bmatrix} v_f - (\tilde{v})_f \\ 0 \\ 0 \end{bmatrix},$$

its energy norm is computable and equals

$$\|v - \tilde{v}\|_A^2 = \begin{pmatrix} v_f - (\tilde{v})_f \end{pmatrix}^T A_{ff} (v_f - (\tilde{v})_f) = \begin{pmatrix} v_f + A_{ff}^{-1} (A_{fc} v_c + A_{fX} v_X) \end{pmatrix}^T A_{ff} (v_f + A_{ff}^{-1} (A_{fc} v_c + A_{fX} v_X)).$$
Since $A_{ff}$ is positive definite, this implies that if we solve the equation
\[
0 = A_{ff} v_f + (A_{fc} v_c + A_{fX} v_X) \\
= (\hat{A}_{ff} + \hat{A}_{fX} E_{Xf}) v_f + (\hat{A}_{fc} + \hat{A}_{fX} E_{Xc}) v_c \\
= \hat{A}_{ff} v_f + \hat{A}_{fc} v_c,
\]
the minimization of $||v_f - (\hat{v})_f||_A$ is attained with zero minimum by $v_f = -\hat{A}_{ff}^{-1} \hat{A}_{fc} v_c$.

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 $\hat{A}_{ff}$ is invertible. If $E_{Xf} = 0$, there is no difficulty, since then $\hat{A}_{ff} = A_{ff}$. In general, if $E_{Xf}$ is sufficiently small in norm, $A_{ff} + A_{fX} E_{Xf}$ 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 $v$ defined on $\Omega(i)$, we wish to extend it to $v_X$, defined on $\Omega_X(\psi)$. Suppose that $i_X$ is an exterior dof, that is, a point from $\Omega_X(\psi)$ whose value we wish to determine from the values of the dofs in $\Omega(i)$. Let $S = \{j \in \Omega(i) : a_{i_X,j} \neq 0\}$; that is, $S$ comprises those dofs in $\Omega(i)$ to which the point $i_X$ is connected. It seems natural to consider using a simple average over these dofs as the extension at $i_X$. Thus, the extension formula, for the dof $i_X$, is given by
\[
v_X(i_X) = 1 \sum_{j \in S} v(j).
\]

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 $i_X$ is given by the formula
\[
v_X(i_X) = \sum_{j \in S} \frac{|a_{i_X,j}|}{\sum_{j \in S} |a_{i_X,j}|} v(j).
\]

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 $\Omega_X(\psi)$ are determined independently. A second feature shared by the methods is that if the neighborhood vector $v$ 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 $v$ be a vector defined on $\Omega(i)$ that we wish to extend to $\Omega_X(\psi)$. We construct the extension to be that operator which produces $v_X$ that minimizes the functional $Q(v_X)$, where

$$Q(v_X) = \sum_{i \in \Omega_X(\psi)} \sum_{j \in \Omega(i)} |a_{i,j}| (v_i - v_j)^2.$$  

(6.1)

It is evident that, like the previous extension operators, if $v$ is constant on $\Omega(i)$ then the dofs in $\Omega_X(\psi)$ 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:

$$\langle \theta v \rangle^T A_{\Omega(\psi)} \langle \theta v \rangle \rightarrow \min,$$

where $\Omega(\psi) \equiv \Omega(\psi) \cup \Omega_X(\psi)$ subject to $v_f, v_c$ fixed. Here

$$\theta = \begin{bmatrix} I & 0 \\ 0 & \theta_X \end{bmatrix} \begin{bmatrix} \Omega(\psi), \\ \Omega_X(\psi) \end{bmatrix}$$

is a diagonal matrix. A good choice is a diagonal matrix $\theta_X$ formed from the vector

$$\theta_X = -(A_{\Omega_X})^{-1} [A_{\Omega_f}, A_{\Omega_c}] \begin{bmatrix} (1)_f \\ (1)_c \end{bmatrix}.$$  

Here we used the blocks of $A$ corresponding to its $\Omega_X(\psi)$ rows.

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

$$v_X = E_{\Omega_c} v_c + E_{\Omega_f} v_f$$

$$= -\theta_X^{-1} (A_{\Omega_X})^{-1} [A_{\Omega_f}, A_{\Omega_c}] \begin{bmatrix} v_f \\ v_c \end{bmatrix}.$$  

Note that this extension mapping is also a simultaneous extension operator and an averaging one; i.e., if $v_c = (1)_c$ and $v_f = (1)_f$, then $v_X = (1)_X$.

### 6.1. Classical AMG as an extension method.

The interpolation method of the classical AMG algorithm popularized by Ruge and Stuben [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 $i$:

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

An $A$-extension is defined in the following manner. For each $i_X \in \Omega_X(\psi)$, set $v_{i_X} = v_i$ if $i_X$ 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 $\theta$ is a user-specified parameter, and weakly connected otherwise.) If $i_X$ is strongly connected to $i$, the extension is defined by

$$v_{i_X} = \frac{1}{\sum_{j \in \Omega(i)} a_{i_X,j}} \sum_{j \in \Omega(i)} a_{i_X,j} v_j.$$ 

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 using only 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 6.1. 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 $\Omega_X(\psi) = \{W,E\}$. We see then that $A_{ff} = [8]$, $A_{fc} = [\begin{array}{cccc} -4 & -4 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{array}]$, and $A_{fX} = [\begin{array}{c} 2 \\ 2 \end{array}]$. For the $A$-extension it is easy to compute the extension operators

$$E_{Xc} = \frac{1}{12} \begin{bmatrix} 1 & 1 & 4 & 4 \\ 1 & 1 & 4 & 4 \end{bmatrix} \quad \text{and} \quad E_{Xf} = \frac{1}{12} \begin{bmatrix} 2 \\ 2 \end{bmatrix},$$

from which

$$\hat{A}_{ff} = \frac{(104)}{12} \quad \text{and} \quad \hat{A}_{fc} = \frac{1}{3} \begin{bmatrix} -11 & -11 & -1 & -1 & -1 \end{bmatrix},$$

which yields an interpolation operator

$$P_A = \frac{1}{26} \begin{bmatrix} 11 & 11 & 1 & 1 & 1 \end{bmatrix}. $$
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 $i$ are all 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

$$P_{L_2} = \frac{1}{44} \begin{pmatrix} 16 & 16 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ \end{pmatrix}.$$  

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

$$P_{AMG} = \frac{1}{12} \begin{pmatrix} 4 & 4 & 1 & 1 & 1 \\ \end{pmatrix},$$

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

$$A_{XX} = \begin{bmatrix} 8 & 0 \\ 0 & 8 \\ \end{bmatrix},$$

$$A_{xf} = \begin{bmatrix} 2 \\ \end{bmatrix},$$

$$A_{xc} = \begin{bmatrix} -1 & -1 & -4 & -4 & 0 & 0 \\ -1 & -1 & 0 & 0 & -4 & -4 \\ \end{bmatrix}.$$  

The vector $\theta_X = -A^{-1}_{XX}[A_{xf} A_{xc}][^{(1)}_{(c)}] = (1)_X$. This is seen as follows:

$$A_{xf}(1)_f = 2(1)_X, \quad A_{xc}(1)_c = -10(1)_X,$$

and hence

$$A_{xf}(1)_f + A_{xc}(1)_c = -8(1)_X,$$

which implies

$$\theta_X = -A^{-1}_{XX}(A_{xf}(1)_f + A_{xc}(1)_c) = -\frac{1}{8} [-8(1)_X] = (1)_X.$$  

That is, the diagonal matrix $\theta$ is the identity and hence the extension matrices then read

$$E_{xf} = -A^{-1}_{XX} A_{xf} = -\frac{1}{4} \begin{bmatrix} 1 \\ 1 \\ \end{bmatrix},$$

$$E_{xc} = -A^{-1}_{XX} A_{xc} = \frac{1}{8} \begin{bmatrix} 1 & 1 & 4 & 4 & 0 & 0 \\ 1 & 1 & 0 & 0 & 4 & 4 \\ \end{bmatrix}.$$
The modified matrices $\hat{A}_{ff}$ and $\hat{A}_{fc}$ take the form

$$\hat{A}_{ff} = A_{ff} + A_{fx}E_{xf} = 8 - [2, 2]\frac{1}{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= 7,$$

$$\hat{A}_{fc} = A_{fc} + A_{fx}E_{xc} = [-4, -4, -1, -1, -1, -1] + [2, 2]\frac{1}{8} \begin{bmatrix} 1 & 1 & 4 & 4 & 0 & 0 \\ 1 & 1 & 0 & 0 & 4 & 4 \end{bmatrix}$$

$$= [-\frac{7}{2}, -\frac{7}{2}, 0, 0, 0, 0].$$

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

$$\left((\hat{A}_{ff})^{-1}\hat{A}_{fc}\right)_i = \begin{bmatrix} 1 & 1 \\ -\frac{1}{2} & 0 \end{bmatrix} 0, 0, 0 \right].$$

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

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

where $V$ is a finite element space of piecewise linear functions over quasi-uniform triangular elements that cover a given two-dimensional polygonal domain. For simplicity, we assume that homogeneous Neumann boundary conditions are imposed and that $(f, 1) = 0$ (to ensure solvability).

Let us denote, for any element $e$,

$$(7.1) \quad g(e) = \sup_{x \in e} \max_{\xi \in \mathbb{R}^2} \frac{\xi^T a(x)\xi}{\xi^T \xi}. $$

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

Further, we assume (only for simplicity) that the neighborhood $\Omega(i) = \Omega(X(i))$ for any fine dof $i$ is formed by a union of triangles that share dof $i$ as a common vertex. Thus we will use $i$ instead of $\psi$ to denote the neighborhoods ($\Omega(i)$, $\Omega_X(i)$, and $\Omega_c(i)$) and the extension mappings. In particular, we denote $E_i = [E_{xf}, E_{xc}]$ where for brevity $E_{xf} = E_{xf}(i)$ and $E_{xc} = E_{xc}(i)$. 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 $a_G(\ldots)$ 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

\[
\begin{bmatrix}
I & 0 \\
0 & I \\
0 & E_{Xc}
\end{bmatrix}
\begin{bmatrix}
\Omega(i) \\
\Omega_c(i) \\
\Omega_X(i)
\end{bmatrix}
\]}

That is, \( E_Xf = 0 \) and \( E_i = [0, E_{Xc}] \).

**Remark 7.1.** The general case of \( E_i = [E_Xf, E_{Xc}] \) can be reduced to the particular case above by using the modified extension mapping \( \hat{E}_i = [0, \hat{E}_{Xc}] \), where

\[
\hat{E}_{Xc} = E_Xf \left( -\hat{A}_{ff}^{-1} \hat{A}_{fe} \right) + E_{Xc}.
\]

To see this, recall that \( \hat{A}_{ff} = A_{ff} + A_{fx}E_Xf \) and \( \hat{A}_{fe} = A_{fe} + A_{fx}E_{Xc} \), and note that the modified extension mapping extends a constant vector defined on \( \Omega_c(i) \) to be the same constant on \( \Omega_X(i) \), that is,

\[
\hat{E}_{Xc}(1)_c = -E_Xf \hat{A}_{ff}^{-1} \hat{A}_{fe}(1)_c + E_{Xc}(1)_c
\]
\[
= E_Xf(1)_f + E_{Xc}(1)_c
\]
\[
= (1)_X.
\]

Here we have used the fact that since (for the model second-order elliptic problem)

\[
A_{ff}(1)_f + A_{fx}(1)_X + A_{fe}(1)_c = 0, \text{ then } A_{ff}(1)_f + A_{fx}(E_Xf(1)_f + E_{Xc}(1)_c) + A_{fe}(1)_c = 0.
\]

That is, \( \hat{A}_{ff}(1)_f + \hat{A}_{fe}(1)_c = 0 \), implying that \( (1)_f = -\hat{A}_{ff}^{-1} \hat{A}_{fe}(1)_c \).

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

\[
-A_{ff}^{-1} \left( A_{fx} \hat{E}_{Xc} + A_{fe} \right) = -\hat{A}_{ff}^{-1} \hat{A}_{fe}.
\]

For this we observe that

\[
-A_{ff}^{-1} (A_{fx} \hat{E}_{Xc} + A_{fe}) = -A_{ff}^{-1} \left[ A_{fx} \left( E_Xf(-\hat{A}_{ff}^{-1} \hat{A}_{fe}) + E_{Xc} \right) + A_{fe} \right]
\]
\[
= -A_{ff}^{-1} \left[ A_{fx}E_{Xc} + A_{fe} - A_{fx}E_Xf \hat{A}_{ff}^{-1} \hat{A}_{fe} \right]
\]
\[
= -A_{ff}^{-1} \left[ \hat{A}_{fe} - A_{fx}E_Xf \hat{A}_{ff}^{-1} \hat{A}_{fe} \right]
\]
\[
= -A_{ff}^{-1} \left[ \hat{A}_{ff} - A_{fx}E_Xf \right] \hat{A}_{ff}^{-1} \hat{A}_{fe}
\]
\[
= -A_{ff}^{-1} (A_{ff}) \hat{A}_{ff}^{-1} \hat{A}_{fe}
\]
\[
= -\hat{A}_{ff}^{-1} \hat{A}_{fe}.
\]

Consider the minimization problem

\[
(7.2) \text{ find } v_f \text{ such that } \begin{bmatrix} v_f \\ v_c \end{bmatrix}^T A_{\Pi(i)}^N \begin{bmatrix} v_f \\ v_c \end{bmatrix} = \inf_{w: w_c = v_c \atop w_X = E_i w} a_{\Pi(i)}(w, w).
\]

Thus we seek \( v_f \), the value of \( w \) on \( \Omega(i) \setminus \Omega_c(i) \), which minimizes the quadratic form \( a_{\Pi(i)}(w, w) \) when the values of \( w \) are fixed at the coarse points and are "slave" at the
exterior points ($\Omega_X(i)$); 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 $-A^{-1}_{ff}(A_{fx}E_{xc} + A_{fc})$. That is, the minimizer is given by $w_f = v_f \equiv -A^{-1}_{ff}(A_{fx}E_{xc} + A_{fc})v_c$.

**Proof.** Consider the Neumann matrix

$$A_N^{\Pi(i)} = \begin{bmatrix} A_{ff} & A_{fc} & A_{fx} \\ A_{cf} & A_{cc}^{N} & A_{cx}^{N} \\ A_{xf} & A_{xc}^{N} & A_{xx}^{N} \end{bmatrix}.$$  

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

$$a_{\Pi(i)}(w, w) = \begin{bmatrix} w_f \\ v_c \end{bmatrix}^T \begin{bmatrix} A_{ff} & A_{fc} & A_{fx} \\ A_{cf} & A_{cc}^{N} & A_{cx}^{N} \\ A_{xf} & A_{xc}^{N} & A_{xx}^{N} \end{bmatrix} \begin{bmatrix} w_f \\ v_c \end{bmatrix} = \begin{bmatrix} w_f \\ v_c \end{bmatrix}^T \begin{bmatrix} A_{ff} & A_{fc} + A_{fx}E_{xc} \\ A_{cf} & A_{cc}^{N} + A_{cx}^{N}E_{xc} + E_{xc}^T(A_{xc}^{N} + A_{xx}^{N}E_{xc}) \end{bmatrix} \begin{bmatrix} w_f \\ v_c \end{bmatrix} \times \begin{bmatrix} w_f \\ v_c \end{bmatrix}.$$  

Minimizing this symmetric positive semidefinite quadratic form with respect to $w_f$ is equivalent to solving the equation

$$A_{ff}w_f + (A_{fc} + A_{fx}E_{xc})v_c = 0,$$

which is the same equation that specifies $v_f$ in the element-free AMGe interpolation procedure. \(\square\)

In the next lemma we will remove the constraint on $v$ being fixed at the $\Omega_X(i)$ points.

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

$$q_1(v_c, v_c) \equiv \inf_{v: \Omega_X(i) = v_c} a_{\Pi(i)}(v, v), \quad \text{and} \quad q_2(v_c, v_c) \equiv \inf_{v: \Omega_X = E_i v, v_c \text{ fixed}} a_{\Pi(i)}(v, v).$$

That is, there exists a positive constant $\eta$ such that

$$q_1(v_c, v_c) \leq q_2(v_c, v_c) \leq \eta q_1(v_c, v_c) \quad \text{for all } v_c.$$  

**Proof.** We show the proof for two-dimensional domains. For other domains, we must scale $\varrho$ 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 $a_{\Pi(i)}(v, v)$.
with \( v : v_c = \text{const} \) and \( E_i v = 0 \) on \( \Omega_X(i) \). Note that \( a_{\Pi(i)}(v, v) = 0 \) 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 \( \eta \) is bounded independently of \( E_i \), one first easily sees that
\[
a_{\Pi(i)}(v, v) \leq C \sum_{c \in \Pi(i)} \varrho(c) \sum_{l, k \in e} (v(l) - v(k))^2.
\]

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 \( i_X \in \Omega_X(i) \)
\[
v(i_X) \equiv (E_i v)(i_X) = \sum_{k \in \Omega_c(i) \cup \{i\}} a_{i_X, k} v(k),
\]
where
\[
\sum_{k \in \Omega_c(i) \cup \{i\}} a_{i_X, k} = 1, \quad \text{and} \quad a_{i_X, k} \geq 0.
\]

Then, for any \( j \in \Omega_c(i) \cup \{i\} \),
\[
(v(i_X) - v(j)) = \sum_{k \in \Omega_c(i) \cup \{i\}} a_{i_X, k} (v(k) - v(j)),
\]
and hence
\[
(v(i_X) - v(j))^2 \leq \sum_{k \in \Omega_c(i) \cup \{i\}} a_{i_X, k}^2 \sum_{k \in \Omega_c(i) \cup \{i\}} (v(k) - v(j))^2 \leq \sum_{k \in \Omega_c(i) \cup \{i\}} (v(k) - v(j))^2.
\]

As a result we see that, for \( v_X = E_i v \),
\[
q_2(v_c, v_c) \leq a_{\Pi(i)}(v, v)
\]
\[
\leq C \sum_{c \in \Pi(i)} \varrho(c) \sum_{l, k \in e} (v(l) - v(k))^2 \leq C \max_{e \in \Pi(i)} \varrho(e) \sum_{k, j \in \Omega_c(i) \cup \{i\}} (v(k) - v(j))^2.
\]

Finally, since \( v_f \) is arbitrary on the right-hand side of this inequality,
\[
q_2(v_c, v_c) \leq C \max_{e \in \Pi(i)} \varrho(e) \inf_{v_f} \left( \sum_{e \in \Omega_c(i)} \varrho(e) \sum_{k, j \in \Omega_c(i) \cup \{i\}} (v(k) - v(j))^2 \right) .
\]
It is also true that

\[ q_1(v_c, v_c) \simeq \inf_{v: \ v_c \text{ fixed}} \left( \sum_{e \in \Omega(i)} g(e) \sum_{l, k \in e} (v(l) - v(k))^2 \right). \]

This shows that \( \eta \) can be chosen bounded independently of the actual averaging extension mapping \( E_i \). □

The above estimates involve the factor

\[ \frac{\max_{e \in \Omega(i)} \varrho(e)}{\min_{e \in \Omega(i)} \varrho(e)}. \]

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 \( P v_c \), is proved in the same way as Lemma 7.2.

**Corollary 7.3.** Consider the extended neighborhood of \( i, \tilde{\Omega}(i) = \cup \{ e, \ e \subset \Omega(i) \text{ or } e \subset \Omega(j) \text{ for all } j \in \Omega_X(i) \} \). There is a constant \( \kappa = \kappa_{\tilde{\Omega}(i)} > 0 \), locally estimated, such that the following bound holds:

\[ a_{\tilde{\Omega}(i)}(P v_c, P v_c) \leq \kappa \inf_{w: \ w_c = v_c} a_{\tilde{\Omega}(i)}(w, w). \]

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

\[ v(k) = \begin{cases} (P v_c)(k), & k \in \Omega(i), \\ v_c(k), & k \text{ is a coarse dof outside } \Omega(i), \\ (E_j v)(k), & k \in \Omega_X(j), \text{ for some } j \in \Omega_X(i). \end{cases} \]

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

\[ a_{\tilde{\Omega}(i)}(v, v) \leq \kappa \inf_{w: \ w_c = v_c} a_{\tilde{\Omega}(i)}(w, w). \]

Since \( a_{\tilde{\Omega}(i)}(P v_c, P v_c) \leq a_{\tilde{\Omega}(i)}(v, v) \), the desired result follows. □

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

**Theorem 7.4.** The element-free interpolation mapping \( P \) exhibits the following approximate harmonic property:

\[ a(P v_c, P v_c) \leq \kappa \inf_{w: \ w_c = v_c} a(w, w), \]

where the constant \( \kappa = \max_{i=\text{fine dof}} \kappa_{\tilde{\Omega}(i)} Z(i), \) and the \( \kappa_{\tilde{\Omega}(i)} \) are the local constants from Corollary 7.3.

**Proof.** The proof simply follows from the fact that

\[ a(P v_c, P v_c) \leq \sum_{i=\text{fine dof}} a_{\tilde{\Omega}(i)}(P v_c, P v_c) \]

and by summation of the local estimates from Corollary 7.3. □
Another important property of the element-free interpolation mapping $P$ is that it partitions unity, as we show in the following theorem.

**Theorem 7.5.** $P$ provides a partition of unity. Specifically, the row sums of $P$ are 1.

**Proof.** Let $v_f = P v_c$ 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 $\Omega X(i)$ 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 $P$. $\Box$

**Remark 7.2.** Theorems 7.4 and 7.5 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 system problems the unknowns are split into physical variables. That is, for scalar problems the rule is as described in section 6, while for two-dimensional 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

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

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

where $G$ is the unit square. The matrix of diffusion coefficients includes functions with relatively benign characteristics—there are 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 8.1 displays the coarsening sequence for a typical
Fig. 8.1. Sequence increasingly coarse elements, formed by element agglomeration.

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 8.1 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 dofs tends to decrease by 50–60% with each successive level.

For each of the four interpolation rules, the problem is solved using a preconditioned CG 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 8.2. For each problem size, we display, for each interpolation rule, the number of preconditioned CG iterations required to achieve the desired residual size and $\varrho$, the average convergence factor over the iterations.
Table 8.1  
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.

<table>
<thead>
<tr>
<th>Level</th>
<th>No. of elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25600</td>
</tr>
<tr>
<td>0</td>
<td>nz</td>
</tr>
<tr>
<td>0</td>
<td>90321</td>
</tr>
<tr>
<td>1</td>
<td>32580</td>
</tr>
<tr>
<td>2</td>
<td>14305</td>
</tr>
<tr>
<td>3</td>
<td>7193</td>
</tr>
<tr>
<td>4</td>
<td>3458</td>
</tr>
<tr>
<td>5</td>
<td>1580</td>
</tr>
<tr>
<td>6</td>
<td>714</td>
</tr>
<tr>
<td>7</td>
<td>274</td>
</tr>
<tr>
<td>8</td>
<td>120</td>
</tr>
<tr>
<td>9</td>
<td>42</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 8.2  
CG convergence results; unstructured triangular fine grid; second-order elliptic problem; 
$V(1,1)$-cycle MG, Gauss–Seidel smoother used as preconditioner.

<table>
<thead>
<tr>
<th>Interp. rule</th>
<th>400 elts</th>
<th>1600 elts</th>
<th>6400 elts</th>
<th>25600 elts</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMGe</td>
<td>iterations</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>0.115</td>
<td>0.172</td>
<td>0.252</td>
</tr>
<tr>
<td></td>
<td>0.115</td>
<td>0.172</td>
<td>0.252</td>
<td>0.289</td>
</tr>
<tr>
<td>$A$-extension</td>
<td>iterations</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>0.118</td>
<td>0.158</td>
<td>0.218</td>
</tr>
<tr>
<td></td>
<td>0.118</td>
<td>0.158</td>
<td>0.218</td>
<td>0.247</td>
</tr>
<tr>
<td>$L_2$-extension</td>
<td>iterations</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>0.119</td>
<td>0.161</td>
<td>0.227</td>
</tr>
<tr>
<td></td>
<td>0.119</td>
<td>0.161</td>
<td>0.227</td>
<td>0.249</td>
</tr>
<tr>
<td>Quadratic funct. min.</td>
<td>iterations</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
<td>$\varrho$</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>0.105</td>
<td>0.152</td>
<td>0.222</td>
</tr>
<tr>
<td></td>
<td>0.105</td>
<td>0.152</td>
<td>0.222</td>
<td>0.231</td>
</tr>
</tbody>
</table>
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 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 to 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 (see Figure 8.2). The domain of the problem is $G = (0,1) \times (0,d)$ with $d \leq 1$. For $d \ll 1$ this is the thin beam problem. The problem is

$$u_{xx} + \frac{1}{2} u_{yy} + \frac{1 + \nu}{2} v_{xy} = f_1,$$

$$\frac{1 + \nu}{2} u_{xy} + \frac{1 - \nu}{2} v_{xx} + v_{yy} = f_2,$$

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 agglomerationalgorithm 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 $h = 0.025, 0.0125$, and $0.00625$ for the thin beam. The coarsening histories of the agglomeration algorithm are shown in Table 8.3. Table 8.4 shows the results of the experiments for the beam problem. As in section 8.1, preconditioned CG 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.
Table 8.3
Coarsening history; structured rectangular fine grid; two-dimensional elasticity, $d = 1$.

<table>
<thead>
<tr>
<th>Level</th>
<th>Thick beam $d = 1.0$</th>
<th>Thin beam $d = 0.05$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$h = 0.050$</td>
<td>$h = 0.025$</td>
</tr>
<tr>
<td>0</td>
<td>nz 14884</td>
<td>3684</td>
</tr>
<tr>
<td></td>
<td>dols 882</td>
<td>3362</td>
</tr>
<tr>
<td>1</td>
<td>nz 10440</td>
<td>40880</td>
</tr>
<tr>
<td></td>
<td>dols 264</td>
<td>924</td>
</tr>
<tr>
<td>2</td>
<td>nz 4128</td>
<td>17248</td>
</tr>
<tr>
<td></td>
<td>dols 84</td>
<td>264</td>
</tr>
<tr>
<td>3</td>
<td>nz 1000</td>
<td>4956</td>
</tr>
<tr>
<td></td>
<td>dols 32</td>
<td>94</td>
</tr>
<tr>
<td>4</td>
<td>nz 256</td>
<td>1404</td>
</tr>
<tr>
<td></td>
<td>dols 16</td>
<td>38</td>
</tr>
<tr>
<td>5</td>
<td>nz 64</td>
<td>324</td>
</tr>
<tr>
<td></td>
<td>dols 8</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>nz 144</td>
<td>576</td>
</tr>
<tr>
<td></td>
<td>dols 12</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>nz 64</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>dols 8</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>nz 64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dols 8</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.4
CG convergence results; structured rectangular fine grid; two-dimensional elasticity, $d = 1$, $V(1,1)$-cycle MG, Gauss–Seidel smoother used as preconditioner.

<table>
<thead>
<tr>
<th>Interp. rule</th>
<th>Thick beam, $d = 1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$h = 0.050$</td>
</tr>
<tr>
<td>AMGe</td>
<td>iterations 16</td>
</tr>
<tr>
<td></td>
<td>$\rho$ 0.172</td>
</tr>
<tr>
<td>A-extension</td>
<td>iterations 12</td>
</tr>
<tr>
<td></td>
<td>$\rho$ 0.099</td>
</tr>
<tr>
<td>$L_2$-extension</td>
<td>iterations 13</td>
</tr>
<tr>
<td></td>
<td>$\rho$ 0.101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interp. rule</th>
<th>Thin beam, $d = 0.05$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$h = 0.025$</td>
</tr>
<tr>
<td>AMGe</td>
<td>iterations 17</td>
</tr>
<tr>
<td></td>
<td>$\rho$ 0.180</td>
</tr>
<tr>
<td>A-extension</td>
<td>iterations 20</td>
</tr>
<tr>
<td></td>
<td>$\rho$ 0.227</td>
</tr>
<tr>
<td>$L_2$-extension</td>
<td>iterations 18</td>
</tr>
<tr>
<td></td>
<td>$\rho$ 0.203</td>
</tr>
</tbody>
</table>

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 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 entails 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