AMGe BASED ON ELEMENT AGGLOMERATION

JIM E. JONES AND PANAYOT S. VASSILEVSKI

ABSTRACT. This paper contains the main ideas for an AMGe (algebraic multigrid for finite elements) method based on element agglomeration. In the method, coarse grid elements are formed by agglomerating fine grid elements. Compatible interpolation operators are constructed which yield coarse grid basis functions with a minimal energy property. Heuristics based on interpolation quality measures are used to guide the agglomeration procedure. The performance of the resulting method is demonstrated in two–level numerical experiments.

1. INTRODUCTION

Algebraic multigrid (AMG) [5], [6], [13], [14], was developed as a generalization of standard geometric multigrid to problems that either had no grid or were posed on unstructured grids where standard geometric multigrid methods are difficult to apply. The standard AMG method works well for many problems; however, its performance on some finite element problems is unsatisfactory. The heuristics used in standard AMG are based on properties of M-matrices, and finite element discretizations can produce non M-matrices. This deficiency in the standard AMG method led Brezina et al. [7] to develop AMGe (algebraic multigrid for finite elements). This previous paper showed how to use multigrid convergence theory and the local stiffness matrices for the individual finite elements to produce interpolation operators superior to those produced by standard AMG. This current paper uses AMGe ideas to produce not only interpolation operators, but coarse grids (and elements) as well. The coarse elements are based on agglomeration of fine elements. A key point is the construction of a local, compatible interpolation operator. The interpolation is local in the sense that degrees of freedom in an agglomerate interpolate only from other degrees of freedom in the same agglomerate. The interpolation is compatible in that the interpolation to degrees of freedom shared by two or more agglomerates is uniquely defined. In this way, the coarse element matrices are variationally related to the assembled matrices in a given agglomerated element, as well as (due to the compatibility) the global coarse matrix is variationally obtained from the global fine grid matrix.

Date: March 9, 1999–beginning; Revision: July 23, 2000; 2nd Revision: October 17; Today is October 17, 2000.

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

Key words and phrases. algebraic multigrid, element agglomeration, algebraic coarsening, implementation.

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.

In the remainder of this introductory section, we outline the proposed agglomeration AMGe method. The goal is to solve a system

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

where A is the positive definite matrix arising from a finite element discretization. In the agglomeration AMGe method, we assume that we have access to the individual element matrices. Our goal is to produce the components needed for a two–level solver: a coarse grid, grid transfer operators, and the coarse grid operator. In order to apply the method recursively (i.e. multigrid as opposed to two–level), individual element matrices on the coarse level must be produced. These goals are outlined below.

- GIVEN INFORMATION:
  1. A list $\mathcal{D}_f = \{d\}$ of the fine grid degrees of freedom (dof).
  2. A list $\mathcal{E}_f$ of fine grid elements $\{e\}$, where each element e, by definition, is a list of degrees of freedom, i.e., $e = \{d_1,\ d_2, \ldots,\ d_{n_e}\}$. Typically, $\mathcal{E}_f$ provides an overlapping partition of the set $\mathcal{D}_f$.
  3. The element matrices A_e, i.e., a list of $n_e \times n_e$ real numbers associated with the degrees of freedom of $e = \{d_1,\ d_2, \ldots,\ d_{n_e}\}$. Equivalently, one may say that a quadratic form $a_e(\mathbf{v},\ \mathbf{v}) = \mathbf{v}_e^T A_e \mathbf{v}_e$ is given where $\mathbf{v}$ is a vector (or discrete function) defined on $\mathcal{D}_f$ restricted to e; i.e., $\mathbf{v}_e = \mathbf{v}|_e = \begin{bmatrix} v(d_1) \\ v(d_2) \\ \vdots \\ v(d_{n_e}) \end{bmatrix}$.
     Note that this will be the notation consistently used throughout this paper, namely, for any subset $\Omega \subset \mathcal{D}$ and a vector $\mathbf{v}$ defined on $\mathcal{D}$ we will denote by $\mathbf{v}_\Omega = \mathbf{v}|_\Omega$ the restriction of $\mathbf{v}$ to Ω. When it simplifies the notation, we will sometimes use superscripts instead of subscripts with the same meaning (restriction to subset).
- OUTPUT COARSE INFORMATION:
  1. A coarse set of degrees of freedom, $\mathcal{D}_c \subset \mathcal{D}_f$.
  2. A set of coarse elements $\mathcal{E}_c = \{E_c\}$, i.e., an overlapping partition of $\mathcal{D}_c$.
  3. The coarse element matrices A_{E_c} for each $E_c \in \mathcal{E}_c$.
  4. An interpolation mapping $P:\ \mathcal{D}_c \mapsto \mathcal{D}_f$ such that $P = \begin{bmatrix} P_c^f \\ I \end{bmatrix} \begin{matrix} \}\mathcal{D}_f \setminus \mathcal{D}_c \\ \}\mathcal{D}_c \end{matrix}$.

To be specific, assume that our "algebraic" elements (i.e., a list of collections $\{e\}$ of degrees of freedom) come from a finite element triangulation of a three dimensional domain and respective conforming finite element spaces with nodal degrees of freedom. To create the coarse information we propose the following steps:

- Create a set of agglomerated elements $\mathcal{E} = \{E\}$ where each $E = e_1 \cup e_2 \cup \cdots \cup e_{n_E}$, $e_i \in \mathcal{E}_f$ and E is connected set. By connected, we mean that for any two elements, $e_i, e_j \in E$, there exists a connecting path of elements also in E beginning with e_i and ending with e_j such that consecutive elements in the path have non-empty intersection. This is a result of the "topological" algorithm used in the agglomeration procedure (Algorithm 4.1). Note each fine grid element e should belong to a unique agglomerated element.

- Define faces and vertices of the agglomerated elements as follows.
  - Consider all intersections $E_i \cap E_j$, for all pairs of different agglomerated elements E_i and E_j. An intersection of this type is called a face if it is a maximal one, i.e., if it is not contained in any other intersection. This defines the set of faces $\mathcal{F} = \{F\}$. We will also assume that a list of boundary faces $\partial\mathcal{D}$ will be given and we will append them to $\mathcal{E}_f$. A formal definition of a boundary face is then simply a maximal set of the type $E \cap \partial\mathcal{D}$, i.e., it is not a proper subset of any other intersection set (either of type $E_i \cap E_j$ or of type $E_i \cap \partial\mathcal{D}$);
  - Finally, consider all faces $F \in \mathcal{F}$ as lists of degrees of freedom. For each dof d compute the intersection $\cap\{F : \ d \in F\}$. The minimal (non–empty) intersections define the set of vertices $\mathcal{V} = \{V\}$.

For true finite element applications the last set of vertices will be disjoint sets; each vertex may contain more than one degree of freedom. This is the case if the underlying problem is a finite element discretization of a system of PDEs, like elasticity for example. For three-dimensional problems, one may refine the above algorithm to create edges of the agglomerated elements; edges are defined to be maximal intersections of faces. In order to keep the presentation simple we will focus mostly on two-dimensional problems.

At any rate, the above "topological" information (faces and vertices of elements) is readily provided by most of the finite element grid generators. So, one may assume that this information is given on the fine grid. If not, one can create it as explained above based on computing, for faces, the maximal intersection sets of the type $e_i \cap e_j$, $e_i \neq e_j$ or of the type $e_i \cap$ boundary surface.

In order to generate the same information on a coarse level, it can be advantageous to carry out the intersection sets algorithm by preserving the dimensionality (or topology) in the following sense. If E is an agglomerated element one has the option to represent E either in terms of the dofs of the original elements, or in terms of the faces of the original elements. If the agglomerated elements and the boundary surfaces $\partial\mathcal{D}$ are represented in terms of the faces of the original elements, then all non–empty intersections of the type $E_i \cap E_j$ or $E_i \cap \partial\mathcal{D}$ are maximal. This is the storage (agglomerated elements in terms of faces of elements) that we use in practice.

Definition 1.1 (Coarse degrees of freedom)**.** *Having computed the set of vertices we define our (minimal) coarse set of degrees of freedom to be those degrees of freedom which are contained in a vertex of an agglomerated element:*

$$\mathcal{D}_c = \{d \in \mathcal{D}_f : \ \exists V \in \mathcal{V} \text{ with } d \in V\}.$$

Note that in practice, one may have to enrich the minimal (vertex) set of coarse degrees of freedom for better performance.

Figure 1 shows the the coarse degrees of freedom for a 2–d scalar problem. Note that for a scalar problem, vertex and degree of freedom are synonymous.

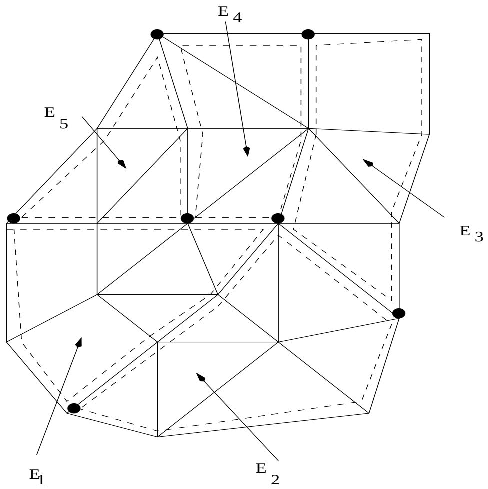

FIGURE 1. Triangulation of domain Ω into triangular and quadrilateral fine grid elements. Agglomerated elements $E_1, E_2, \ldots, E_5$ and coarse degrees of freedom.

Definition 1.2 (Coarse elements)**.** *For each agglomerated element E, we define a coarse element E_c consisting of degrees of freedom contained in a vertex of E, i.e.*

$$E_c = \mathcal{D}_c \cap E.$$

For each agglomerated element E (or equivalently for each coarse element E_c), we construct a local interpolation operator P_E. This operator maps a vector defined at coarse degrees of freedom in E_c to a vector defined at the fine degrees of freedom in E. We require the set of local interpolation operators be compatible in that: if $d \in E_1 \cap E_2$ then $P_{E_1}\mathbf{v}_{E_1^c}(d) = P_{E_2}\mathbf{v}_{E_2^c}(d)$ for all vectors $\mathbf{v}$. In words, compatibility means that at shared degrees of freedom, the interpolation rules for the agglomerates must agree. Compatibility implies the following restriction.

Requirement 1.1. *For $d \in \mathcal{D}_f$, let $N(d) = \cap\{$all agglomerated elements $E(d)$ that contain $d\}$. Then, the value $v(d)$ must be interpolated from the dofs at the vertices of $N(d)$. Note, we assume interpolation is the identity at vertices.*

Definition 1.3 (Interpolation mapping)**.** *Having constructed a compatible set of local interpolation mappings $\{P_E\}$, define a global mapping $P : \mathcal{D}_c \mapsto \mathcal{D}$ by $P\mathbf{v}_c|_E \equiv P_E\mathbf{v}_{E_c}$. Compatibility implies this uniquely defines P.*

Definition 1.4 (Coarse element matrices)**.** *Assume that a compatible set of interpolation operators $\{P_E\}$ has been computed. Let A_E be the assembled matrix corresponding to the agglomerated element $E = e_1 \cup e_2 \cup \cdots \cup e_{n_E}$ defined by*

$$\mathbf{v}_E^T A_E \mathbf{w}_E \equiv \sum_{i=1}^{n_E} \mathbf{v}_{e_i}^T A_{e_i} \mathbf{w}_{e_i} \quad \textit{for any } \mathbf{v}_E,\ \mathbf{w}_E. \tag{1.1}$$

Then, the coarse element matrix for the coarse element E_c, is defined by

$$A_E^c \equiv P_E^T A_E P_E. \tag{1.2}$$

Note that the global coarse (stiffness) matrix A^c defined as

$$A^c = P^T A P$$

can be assembled from the coarse element matrices, i.e., that

$$\mathbf{v}_c^T A^c \mathbf{w}_c = \sum_{i=1}^{n_c} \mathbf{v}_{E_i^c}^T A_{E_i}^c \mathbf{w}_{E_i^c}.$$

Indeed, for $E_i = \bigcup_{j=1}^{n_{E_i}} e_i^j$,

$$\begin{aligned}
\sum_{i=1}^{n_c} \mathbf{v}_{E_i^c}^T A_{E_i}^c \mathbf{w}_{E_i^c} &= \sum_{i=1}^{n_c} (P_{E_i} \mathbf{v}_{E_i^c})^T A_{E_i} (P_{E_i} \mathbf{w}_{E_i^c}) \\
&= \sum_{i=1}^{n_c} (P\mathbf{v}_c|_{E_i})^T A_{E_i} (P\mathbf{w}_c|_{E_i}) \\
&= \sum_{i=1}^{n_c} \sum_{j=1}^{n_{E_i}} (P\mathbf{v}_c|_{e_i^j})^T A_{e_i^j} (P\mathbf{w}_c|_{e_i^j}) \\
&= \sum_{i=1}^{n_f} (P\mathbf{v}_c|_{e_i})^T A_{e_i} (P\mathbf{w}_c|_{e_i}) \\
&= \mathbf{v}_c^T P^T A P \mathbf{w}_c.
\end{aligned}$$

We should mention at this point that there are other approaches of constructing AMG methods that target non–M–matrices. One example is the aggregation based AMG of Vanek, Mandel and Brezina [15]. In this method, one constructs aggregates (non–overlapping partition of the degrees of freedom) and forms a generally unstable (but simple) tentative prolongator. Finally, a smoothing step is applied in order to get a better quality interpolation. In Wan, Chan and Smith [17] a direct approach of constructing coarse bases is proposed. The bases are selected by minimizing a quadratic energy functional while enforcing locality and a partition of unity property. In Mandel, Brezina and Vanek [12] this approach was further developed by proposing fast algorithms for minimizing the quadratic functional. In Chan, Xu and Zikatanov [9] the construction of the agglomerated elements is used a posteriori in the sense that one first selects a coarse grid (as maximal independent set) and then agglomerated elements are constructed (based on the dual matrix graph). The agglomerates are subsequently divided into triangles and the procedure can be recursively applied. The interpolation weights are computed based on averaging. In that sense, the present paper substantially differs from [9]. Our agglomeration algorithm is different (the coarse dofs are selected after the agglomeration is performed), and we assume more information. Namely, similar to the original AMGe paper ([7]), we require access to the individual elements and respective element matrices on the fine grid. Note that this information is readily provided by most finite element grid generators. In contrast to [7] we are able to more systematically generate the input information (elements and their respective element matrices) on the coarse levels. This allows straightforward recursive use of the same two–level algorithm.

The remainder of the present paper is organized as follows. In Section 2 we consider the construction of the local interpolation mappings, based on a minimal energy principle. Section 3 deals with energy minimization property of the coarse basis. In Section 4, we specify an algorithm for agglomerating elements, which provides nicely matched agglomerated elements for structured triangular or quadrilateral meshes. We also discuss using measures of interpolation quality to guide the agglomeration procedure yielding semicoarsening for problems with anisotropy. In the final Section, the performance of the resulting method is demonstrated in two–level numerical experiments.

2. The local interpolation mappings

In this section we present an algorithm for generating the local interpolation mappings in a way that produces coarse grid basis functions with a quasi–minimal energy property. Most of the proofs in this section rely on basic properties of Schur complements of symmetric positive semi–definite matrices. A summary of these properties can be found, for example, in [1], §3.2. The problems that we target are second order scalar elliptic problems without the low order term as well as elasticity in 2-d and 3-d.

We begin by defining, for each fine grid dof d, the following sets:

- a neighborhood $\Omega(d) = \cup\{$all agglomerated elements $E(d)$ that contain $d\}$;
- a minimal set $N(d) = \cap\{$all agglomerated elements $E(d)$ that contain $d\}$.

Note that $N(d)$ can be a vertex, a face or even an agglomerated element. From the definition of vertices, each $N(d)$ contains at least one vertex. Note, also, that there might be multiple copies of $N(d)$, i.e., $N(d_i) = N(d_j)$ for a $d_i \neq d_j$. We next introduce the following definition for the boundary of the sets $N(d)$.

Definition 2.1. *For any set $N(d)$ different than a face or agglomerated element, define the boundary of $N(d)$, denoted $\partial N(d)$, to be the vertices contained in $N(d)$ (which is non–empty set). If $N(d)$ is a face of an agglomerated element, define $\partial N(d)$ as the dofs in $N(d)$ that belong to more than one face. Finally, if $N(d)$ is an agglomerated element E, define the boundary, ∂E, as the union of all faces of E.*

We now describe the construction of the local and compatible interpolation mappings. The set of interpolatory coarse dofs $d^c_1, \ldots, d^c_p$ that will be used to interpolate to d is constructed according to Requirement 1.1. That is, $d^c_1 = d$ if d belongs to a vertex; otherwise, the interpolatory coarse dofs are the vertices of the set $N(d)$.

To define the interpolation weights for a dof d we use the following recursive procedure. The interpolation is identity at vertices. Then, for the set $N(d)$ assume that the interpolation at dofs on $\partial N(d)$ has already been defined, i.e., $(P\mathbf{v}_c)|_{\partial N(d)}$ is well defined for $\mathbf{v}_c$ specified at the vertices of $N(d)$. Now, extend the definition of $P\mathbf{v}_c$ on $N(d) \setminus (\partial N(d))$ by considering the neighborhood $\Omega(d)$ of all agglomerated elements that contain d. Let $A_{\Omega(d)}$ be the assembled matrix corresponding to all elements contained in that neighborhood. Consider the following two–by–two block structure of $A_{\Omega(d)}$, corresponding to the partitioning $(\Omega(d) \setminus \partial N(d)) \cup \partial N(d)$,

$$A_{\Omega(d)} = \begin{array}{cl} \left[\begin{array}{cc} A_{ii} & A_{ib} \\ A_{bi} & A_{bb} \end{array} \right] & \begin{array}{l} \}\Omega(d) \setminus \partial N(d) \\ \}\partial N(d). \end{array} \end{array}$$

Here "i" stands for interior, and "b" for boundary dofs. Note that $\{d_1^c, \ldots, d_p^c\} \subset \partial N(d)$. The interpolation coefficients $w_{d,\, d_i^c}$, $i = 1, 2, \ldots, p$ are obtained by solving the following equation ($\mathbf{x}^c$ given),

$$A_{ii}\mathbf{x}^i + A_{ib}(P\mathbf{x}^c)_{\partial N(d)} = 0.$$

Then, the equation corresponding to a dof d_f in $N(d) \setminus \partial N(d)$, gives

$$(\mathbf{x}^i)_{d_f} = \left(-A_{ii}^{-1} A_{ib}(P\mathbf{x}^c)_{\partial N(d)}\right)\Big|_{d_f}.$$

That is, in particular for $d_f = d$, and $\mathbf{x}^c = \begin{array}{cl} \left[\begin{array}{c} 0 \\ 1 \end{array} \right] & \begin{array}{l} \}\text{vertices of } N(d) \setminus \{d_i^c\} \\ \}d_i^c \end{array} \end{array}$, one gets the interpolatory coefficient

$$w_{d,\, d_i^c} = \left(-\, A_{ii}^{-1} A_{ib} \left(P \begin{array}{cl} \left[\begin{array}{c} 0 \\ 1 \end{array} \right] & \begin{array}{l} \}\text{vertices of } N(d) \setminus \{d_i^c\} \\ \}d_i^c \end{array} \end{array} \right)_{\partial N(d)} \right)\Bigg|_d .$$

This approach assumes that A_{ii} is invertible. As the following lemma shows, this is always the case, for symmetric positive semidefinite matrices $A_{\Omega(d)}$, if the set of boundary dofs $\partial N(d)$ is sufficiently rich.

Lemma 2.1. *Given a set E, a union of fine elements, partition it into two groups: "f"–dofs denoted by $\mathcal{D}_{E,\, f}$ and "c"-dofs denoted $\mathcal{D}_{E,\, c}$. Let A_E be the assembled matrix corresponding to E partitioned as follows*

$$A_E = \left[\begin{array}{cc} A_{E,\, ff} & A_{E,\, fc} \\ A_{E,\, cf} & A_{E,\, cc} \end{array} \right].$$

If there exists a basis $\{\mathbf{d}_i\}$ for the null–space of the assembled, symmetric positive semidefinite matrix A_E, such that $\{\mathbf{d}_i\}$ restricted to $\mathcal{D}_{E,\, c}$ remain linearly independent, then $A_{E,\, ff}$ is invertible.

Proof. Assume that $A_{E,\, ff}\mathbf{x}^f = 0$. This implies that

$$\left[\begin{array}{c} \mathbf{x}^f \\ 0 \end{array} \right]^T A_E \left[\begin{array}{c} \mathbf{x}^f \\ 0 \end{array} \right] = 0,$$

and since A_E is positive semidefinite, this implies

$$A_E \begin{bmatrix} \mathbf{x}^f \\ 0 \end{bmatrix} = 0.$$

That is, $\begin{bmatrix} \mathbf{x}^f \\ 0 \end{bmatrix}$ is in the null–space of A_E. Therefore, we can expand it in terms of the basis of the null–space, i.e.,

$$\begin{bmatrix} \mathbf{x}^f \\ 0 \end{bmatrix} = \sum_i c_i \mathbf{d}_i.$$

The second block equation implies,

$$0 = \sum_i c_i \mathbf{d}_i^c.$$

The assumption that $\{\mathbf{d}_i\}$ remains linearly independent when restricted to $\mathcal{D}_{E,\,c}$ means that $\{\mathbf{d}_i^c\}$ are linearly independent. Thus all $c_i = 0$ and $\mathbf{x}^f = 0$. That is, $A_{E,\,ff}\mathbf{x}^f = 0$ implies $\mathbf{x}^f = 0$, hence $A_{E,\,ff}$ is invertible. $\square$

Remark 2.1. *For the model case of second order scalar elliptic equations,* $\mathcal{L}u \equiv -\operatorname{div}(a\nabla u) = F$, *a basis of the null–space of* A_E *is* $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ *and its restriction onto the set of coarse dofs is again the constant vector, hence it is linearly independent. The above lemma shows that the corresponding* $A_{E,\,ff}$ *will be invertible.*

Remark 2.2. *If* $\mathbf{x}$ *is in the null–space of* A_E*, i.e.*

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^f \\ \mathbf{x}^c \end{bmatrix} \text{ and } A_E\mathbf{x} = 0,$$

then

$$A_{E,\,ff}\mathbf{x}^f + A_{E,\,fc}\mathbf{x}^c = 0.$$

Thus the previously defined interpolation procedure is exact for vectors in the null–space of A_E*.*

In showing that the interpolation mappings produce coarse basis functions enjoying a certain energy minimization property, we rely on the following relationships between energy minimization and Schur complements.

Remark 2.3. *Consider a matrix* A *with any two–by–two blocking*

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}.$$

Assume A_{ff} *is invertible, and define the Schur complement of* A *on* c *as* $S_c \equiv A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$*. If* A *is symmetric positive semidefinite, then*

$$\mathbf{v}_c^T S_c \mathbf{v}_c = \inf_{\mathbf{v}|_c = \mathbf{v}_c} \mathbf{v}^T A \mathbf{v}. \tag{2.1}$$

In cases where A_{ff} is not invertible, (2.1) *can be used to define the Schur complement. Note that if A is symmetric positive semidefinite so is S_c. Finally, one has the identity*

$$A\mathbf{v} = \begin{bmatrix} 0 \\ S_c \mathbf{v}_c \end{bmatrix}, \tag{2.2}$$

for any minimizer $\mathbf{v}$, *i.e., for any vector* $\mathbf{v}$ *for which* $\mathbf{v}_c^T S_c \mathbf{v}_c = \mathbf{v}^T A \mathbf{v}$ *and* $\mathbf{v}\big|_c = \mathbf{v}_c$.

The following lemma is a straightforward consequence of Remark 2.3.

Lemma 2.2. *Using the notation of the previous remark, assume A_{ff} is invertible and let $\mathbf{v}_c$ be a null vector of S_c, then $\mathbf{v}_c$ can be uniquely extended to the null–space of A.*

We are now ready to show several energy minimization properties of the local interpolation mappings P_E formulated for simplicity for 2–d elements.

We first demonstrate an energy minimization property for dofs interior to an agglomerated element. Let d belong to a unique agglomerated element E. Thus the neighborhood $\Omega(d)$, used to define interpolation, consists of the fine–grid elements that are contained in E. Then, $P = P_E$ is constructed based on the following block–ordering of A_E,

$$A_E = \begin{bmatrix} A_{ii} & A_{ib} \\ A_{bi} & A_{bb} \end{bmatrix} \begin{array}{l} \} \;\; E \setminus \partial E \\ \} \;\; \partial E \end{array}$$

The coefficients of P_E are obtained by solving the equation ($\mathbf{x}^c$ given)

$$A_{E,\, ii}\mathbf{x}^i + A_{E,\, ib}(P_E \mathbf{x}^c)_{\partial E} = 0.$$

It is equivalent then to say that $\mathbf{x}^i = -A_{ii}^{-1} A_{ib}(P_E \mathbf{x}^c)$ solves the minimization problem,

$$\min_{\mathbf{x}:\ \mathbf{x}|_{\partial E} = (P_E \mathbf{x}^c)_{\partial E}} \mathbf{x}^T A_E \mathbf{x}, \tag{2.3}$$

By definition, $P_E \mathbf{x}_c\big|_d = -\, A_{ii}^{-1} A_{ib} (P_E \mathbf{x}^c)_{\partial E}\big|_d$ for all $d \in E$ that do not belong to a face of E.

We next show an energy minimization property for dofs on faces; this is used later to show a global energy minimization property of the coarse grid basis functions. For every face F, the neighborhood used to define interpolation is $E_F^+ \cup E_F^-$ where E_F^+ and E_F^- are the two neighboring agglomerated elements that form the face F (one of them can be $\emptyset$ if F is a boundary face).

Lemma 2.3. *For every face $F = E_F^+ \cap E_F^-$, the interpolation P minimizes the quadratic form $(\mathbf{w}_F)^T (S_{E_F^+,\, F} + S_{E_F^-,\, F}) \mathbf{w}_F$ for $\mathbf{w}_F$ fixed at the vertices of F, where $S_{E,\, F}$ denotes the Schur complement of A_E on F.*

Proof. Denote $E_1 = E_F^-$ and $E_2 = E_F^+$. Each dof on F which is not a vertex is interpolated from the vertices of F based on the assembled matrix $A_{E_1 \cup E_2}$ corresponding to the domain $E_1 \cup E_2$. To define P on F one looks at the matrix

$$A_{E_1 \cup E_2} = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix} \begin{array}{l} \}E_1 \cup E_2 \setminus (\text{ vertices of } F) \\ \}(\text{ vertices of } F). \end{array}$$

Then $(P\mathbf{v}^c)(d_f) = \left(-A_{ff}^{-1} A_{fc}\mathbf{v}^c\right)(d_f)$ for any $d_f \in F \setminus (\text{ vertices of } F)$. Equivalently, from the equations that define P on F,

$$A_{ff}\mathbf{w}^f + A_{fc}\mathbf{w}^c = 0,$$

one can eliminate the dofs that are on $E_1 \cup E_2 \setminus F$ thus ending up with the Schur complement problem,

$$S_{E_1 \cup E_2,\, F}\mathbf{w}^F\big|_{F \setminus (\text{ vertices of } F)} = 0, \quad \mathbf{w}^F = \begin{bmatrix} \mathbf{w}^f \\ \mathbf{w}^c \end{bmatrix} \begin{array}{l} \}F \setminus (\text{ vertices of } F) \\ \}\text{ vertices of } F. \end{array} \tag{2.4}$$

Since F is a separator for $E_1 \cup E_2$ one has that $S_{E_1 \cup E_2,\, F} = S_{E_1,\, F} + S_{E_2,\, F}$. Since $S_{E_1 \cup E_2,\, F}$ is symmetric semidefinite, problem (2.4) is equivalent to the following minimization problem,

$$\inf_{\mathbf{w}^F|_{\text{vertices of } F} = \mathbf{w}^c} (\mathbf{w}^F)^T S_{E_1 \cup E_2,\, F}\mathbf{w}^F.$$

By definition $\mathbf{w}^F = P\mathbf{w}^c$ solves problem (2.4), and thus has this equivalent minimization property. $\square$

Throughout the remainder of the paper we will assume the following relations between the null–spaces of the assembled matrices A_{E_-} and A_{E_+} for any two neighboring agglomerated elements E_- and E_+ that share a common face F.

ASSUMPTION 2.1. *For any $\mathbf{x}_{E_-}$ such that $A_{E_-}\mathbf{x}_{E_-} = 0$ there is an extension $\mathbf{x}$ of $\mathbf{x}_{E_-}$ defined on $E_- \cup E_+$ such that $A_{E_- \cup E_+}\mathbf{x} = 0$ and $\mathbf{x}|_{E_-} = \mathbf{x}_{E_-}$. Equivalently, $A_{E_+}\mathbf{x}_{E_+} = 0$ and $\mathbf{x}|_F = \mathbf{x}_{E_-}|_F$.*

As a corollary of the above assumption, the respective Schur complements $S_{E_-;\, F}$ and $S_{E_+;\, F}$ of A_{E_-} and A_{E_+}, on the face F are spectrally equivalent, or equivalently, have the same null–space.

Actually, the following local estimates hold:

Lemma 2.4. *Assume, in addition to Assumption 2.1, that every null–vector $\mathbf{v}$ of A_E restricted to a face F of E is uniquely determined from its vertex values $\mathbf{v}_c$ on F. Note that this is always the case if the set of coarse dofs on any F is sufficiently rich (see Lemma 2.1). If we have determined $\mathbf{x} = P_E\mathbf{x}_c$ first on ∂E, and then in the interior of E as specified above, the following local quadratic forms*

$$(P_E\mathbf{x}_c)^T A_E P_E\mathbf{x}_c, \quad \inf_{\mathbf{x}:\ \mathbf{x}|_{\mathcal{D}_c} = \mathbf{x}_c} \mathbf{x}^T A_E \mathbf{x},$$

are spectrally equivalent. That is, there exists a constant η_E such that

$$\inf_{\mathbf{x}:\ \mathbf{x}|_{\mathcal{D}_c} = \mathbf{x}_c} \mathbf{x}^T A_E \mathbf{x} \le (P_E\mathbf{x}_c)^T A_E P_E\mathbf{x}_c \le \eta_E \inf_{\mathbf{x}:\ \mathbf{x}|_{\mathcal{D}_c} = \mathbf{x}_c} \mathbf{x}^T A_E \mathbf{x}.$$

In other words, the coarse element matrix $A^c_{E_c}$ and the Schur complement S_c of A_E on $\mathcal{D}_c \cap E$, are spectrally equivalent.

Proof. To prove the result it is sufficient to show that both matrices have the same null–space. Assume now that $S_c \mathbf{x}_c = 0$. For any face F of E one can compute the Schur complement of S_c on F denoted by $S_{c,\,F}$. It is clear then (see (2.2)) that

$$S_{c,\,F} \mathbf{x}_{c,\,F} = 0. \tag{2.5}$$

Our goal is to show that $(P_E)^T A_E P_E \mathbf{x}_c = 0$, which is equivalent to $A_E(P_E \mathbf{x}_c) = 0$. By construction, one has $A_E(P_E \mathbf{x}_c) = 0$ in the interior of E. Also, from the definition of P_E for dofs on faces F (see (2.4)) one has

$$\left(S_{E,F} + S_{E_+,F}\right) \left(P_E \mathbf{x}_c\right)_F \big|_{F \setminus \text{vertices of } F} = 0.$$

Here, E_+ is the neighboring element to E which shares a common face F with E. From Assumption 2.1 it follows that $S_{E,F} + S_{E_+,F}$ and $S_{E,F}$ have the same null–space. Therefore their respective Schur complements on the vertices of F ($F \cap \mathcal{D}_c$), $\sigma_{c,\,F}$ and $S_{c,\,F}$ will have the same null–space. Then (2.5) implies that $\sigma_{c,\,F} \mathbf{x}_{c,\,F} = 0$. Applying identity (2.2) (based on Lemma 2.3) yields

$$\left(S_{E,F} + S_{E_+,F}\right) \left(P_E \mathbf{x}_c\right)_F = \begin{bmatrix} 0 \\ \sigma_{c,\,F} \mathbf{x}_{c,\,F} \end{bmatrix} \begin{array}{l} \} \;\; F \setminus \text{vertices of } F \\ \} \;\; \text{vertices of } F \end{array},$$

from which it follows that

$$\left(S_{E,F} + S_{E_+,F}\right) \left(P_E \mathbf{x}_c\right)_F = 0 \text{ on } F.$$

Again, the fact that $S_{E,F} + S_{E_+,F}$ and $S_{E,F}$ have the same null–space, implies that

$$S_{E,F} \left(P_E \mathbf{x}_c\right)_F = 0 \;\; \text{on } F.$$

This shows that $\left(P_E \mathbf{x}_c\right)_F$ is a restriction of a null–vector of A_E on F. Assumption 2.1 and the additional assumption we have made that every vector in the null-space of A_E restricted to a face is uniquely determined by its vertex values on that face, then imply that $\left(P_E \mathbf{x}_c\right)_{\partial E}$ is the restriction of a null–vector of A_E on ∂E. This together with the fact that $A_E \left(P_E \mathbf{x}_c\right) = 0$ in the interior of E, finally show that

$$A_E \left(P_E \mathbf{x}_c\right) = 0 \;\; \text{on } E.$$

This completes the proof that $P_E \mathbf{x}_c$ is in the null–space of A_E, i.e., that $\mathbf{x}_c$ is in the null–space of $A^c_{E_c}$. The converse is also true. Namely, $A^c_{E_c} \mathbf{x}_c = 0$ implies that $(P_E \mathbf{x}_c)^T A_E P_E \mathbf{x}_c = 0$ and since A_E is symmetric positive semi–definite, one gets that $A_E P_E \mathbf{x}_c = 0$, or that $P_E \mathbf{x}_c$ belongs to the null–space of A_E. Therefore, $\mathbf{x}_c = P_E \mathbf{x}_c|_{\text{vertices of } E}$ belongs to the null–space of the Schur complement S_c of A_E. ☐

We then have the following global estimate by summing up the local estimates over the individual agglomerated elements.

Theorem 2.1. *The compatible local interpolation mapping $P = P_E$ is approximately harmonic in the sense, that its norm in energy inner product is bounded,* *i.e.,*

$$\begin{aligned}\mathbf{v}_c^T A_c \mathbf{v}_c &= (P\mathbf{v}_c)^T A (P\mathbf{v}_c)\\ &\le \sum_E \eta_E \inf_{\mathbf{v}_E|_{\mathcal{D}_c \cap E} = \mathbf{v}_c} \mathbf{v}_E^T A_E \mathbf{v}_E\\ &\le \eta \inf_{\mathbf{v}|_{\mathcal{D}_c} = \mathbf{v}_c} \mathbf{v}^T A \mathbf{v}.\end{aligned}$$

The exact harmonic mapping corresponds to the best constant $\eta = 1$. As shown in Lemma 2.4, $\eta = \max_{E\in\mathcal{E}} \eta_E$, thus the individual η_E can be estimated locally. With this result, a classical two–level Gauss–Seidel iteration (see, e.g., Bank and Dupont [3] *or Bank* [2]*) will have a convergence factor bounded by $\gamma^2 = 1 - \frac{1}{\eta}$.*

Remark 2.4. *Note that the proof of Theorem 2.1 does not require uniqueness of the minimizers (hence of P). Note however that we assumed uniqueness on the faces (see Lemma 2.4). Hence, it applies to element matrices coming from 2–d and 3–d elasticity. If one assumes a little more (see below Assumption 2.2) the uniqueness of P (or of the minimizers) is guaranteed.*

Namely, one may assume:

Assumption 2.2. *If d_c is a dof at a vertex and E is an agglomerated element containing that vertex, the only vector in the null–space of A_E and vanishing at d_c is the zero vector.*

For the model case of 2–d and 3–d second order scalar elliptic equations (of the form $\mathcal{L}u \equiv -\operatorname{div} a\nabla u = f$), this assumption holds. However, it may not hold for systems of PDEs (it is not true for elasticity problems, for example). If Assumption 2.2 holds, P_E is defined uniquely at the interior of $N(d)$ (edge, face or agglomerated element E) based on a Schur complement of $A_{\Omega(d)}$ (to $N(d)$) by harmonically extending the values from the boundary of $N(d)$ into its interior. In particular, one has (see (2.3)) that for each E the following identity holds, $\mathbf{w}_F = P_E \mathbf{w}_c|_F$, for any face (or edge) $F \subset E$,

$$\mathbf{w}_c^T A_E^c \mathbf{w}_c = \inf_{\mathbf{v}_E|_F = \mathbf{w}_F,\ \text{for all } F\subset E} \mathbf{v}_E^T A_E \mathbf{v}_E. \tag{2.6}$$

Remark 2.5. *The constants η_E in Lemma 2.4 are computable and can be used as local measures for interpolation quality in the sense that smaller η_E implies better interpolation. Theorem 2.1 shows that the local measures imply the approximate harmonic property of P. More details on how to compute measures of interpolation quality and its relation with other local constants are found in Section 4.*

3. Energy minimization properties of coarse basis functions

With the local interpolation operators defined, one can construct a coarse grid basis function v_d for each $d \in \mathcal{D}_c$ as follows. Define the coarse grid vector $\mathbf{v}_d^c$ that is one at d and zero elsewhere and define $\mathbf{v}_d$ as this vector interpolated to the fine grid (i.e., $\mathbf{v}_d = P\mathbf{v}_d^c$). It is clear then that it will be zero outside the neighborhood $\Omega(d) = \cup_{i=1}^p E_i$ of the given dof d. In this way, $\mathbf{v}_d$ can be viewed as a basis vector (function) of the interpolated coarse space. Using finite element terminology, one may also say that $\mathbf{v}_d$ is a fine grid vector representation of a coarse–grid basis function.

Lemma 3.1. *For the model problem of finite element matrices (before imposing Dirichlet boundary conditions) coming from second order scalar elliptic problems (2–d or 3–d), the $\{\mathbf{v}_d\}$ provide partition of unity, i.e.,*

$$\sum_{d\in\mathcal{D}_c} \mathbf{v}_d = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{3.1}$$

Proof. In the case of finite element matrices coming from 2–d (or 3–d) second order scalar elliptic problems, constant vectors are in the null–space of the element matrices. By Remark 2.2, if $\mathbf{v}_c = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathcal{R}^{n_c}$ then $\mathbf{v} = P\mathbf{v}_c = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathcal{R}^{n}$. This holds since $\mathbf{v}_E = P_E\mathbf{v}_{c,\,E_c}$ for each coarse element E_c (or agglomerated element E). This in particular implies that $\sum_{d\in\mathcal{D}_c} \mathbf{v}_d = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathcal{R}^{n}$. □

Corollary 3.1. *Consider the model case of finite element matrices (before imposing Dirichlet boundary conditions) coming from second order scalar elliptic problems (2–d or 3–d) on quasiuniform triangulation. Let $\{\mathbf{v}_d\}$ be the set of basis functions generated by the local interpolation operators. Let $\{\mathbf{w}_d\}$ be any other potential set of local basis functions, i.e. a basis function exists for each $d \in \mathcal{D}_c$ with $\mathbf{w}_d(d) = 1$ and $\mathbf{w}_d = 0$ outside of the neighborhood $\Omega(d)$. Then the following energy minimization property of $\{\mathbf{v}_d\}$ holds*

$$\sum_{d\in\mathcal{D}_c} \mathbf{v}_d^T A\mathbf{v}_d \leq C \sum_{d\in\mathcal{D}_c} \inf_{\mathbf{w}_d} \mathbf{w}_d^T A\mathbf{w}_d. \tag{3.2}$$

Proof. Applying the approximate harmonic property of P_E for each agglomerated element E (Lemma 2.4), one ends up with the estimate

$$(\mathbf{v}_d|_E)^T A_E(\mathbf{v}_d|_E) \leq \eta_E \inf_{\mathbf{w}_E:\ \mathbf{w}|_{\text{vertices of } E} = \mathbf{v}_d|_{\text{vertices of } E}} \mathbf{w}_E^T A_E\mathbf{w}_E.$$

Summing up over the agglomerated elements $E:\ E \subset \Omega(d)$, where $\Omega(d)$ is the union of all agglomerated elements that contain the vertex d (note that $\mathbf{v}_d$ is zero outside $\Omega(d)$), one ends up with the global estimate

$$\mathbf{v}_d^T A\mathbf{v}_d \leq \eta \inf_{\mathbf{w}_d:\ \mathbf{w}_d|_{\mathcal{D}_c} = \mathbf{v}_d|_{\mathcal{D}_c}} \mathbf{w}_d^T A_{\Omega(d)}\mathbf{w}_d, \quad \eta = \max_E \eta_E.$$

Note that $\mathbf{w}_d = 1$ at the vertex d and is zero at the remaining vertices, and it is also zero outside $\Omega(d)$, i.e., it is locally supported.

Finally, summing over all $d \in \mathcal{D}_c$ one ends up with the desired estimate

$$\sum_{d\in\mathcal{D}_c} \mathbf{v}_d^T A \mathbf{v}_d \le C \sum_{d\in\mathcal{D}_c} \inf_{\mathbf{w}_d:\ \mathbf{w}_d|_{\mathcal{D}_c}=\mathbf{v}_d|_{\mathcal{D}_c}} \mathbf{w}_d^T A_{\Omega(d)} \mathbf{w}_d = C \sum_{d\in\mathcal{D}_c} \inf_{\mathbf{w}_d:\ \mathbf{w}_d|_{\mathcal{D}_c}=\mathbf{v}_d|_{\mathcal{D}_c}} \mathbf{w}_d^T A \mathbf{w}_d.$$

□

Remark 3.1. *Theorem 3.1 shows, for the model case of finite element matrices coming from second order scalar elliptic equations as well as in the elasticity, that the coarse basis functions corresponding to the coefficient vectors $\mathbf{v}_d$ solve the energy minimization functional as defined in Wan, Chan and Smith* [17] *up to a multiplicative constant. Fast algorithms to solve the problem of energy minimization functional are proposed and analyzed in Mandel, Brezina and Vanek* [12].

Remark 3.2. *For finite element matrices coming from 2–d and 3–d second order scalar elliptic problems on quasiuniform triangulation, the coarse space produced by the above algorithm also admits a weak approximation property (or equivalently, provide partition of unity, see Lemma 3.1 and also estimate* (4.2)*) since the element matrices contain the constants in their null–space. Therefore the constant is exactly interpolated from the vertices of the agglomerated elements as the same constant on the rest of the agglomerated element. That is, with the above minimization property, the AMGe method can actually become an optimal (or almost optimal) order MG method if one can control the local constants η_E from Lemma 2.4 which depend on the way we agglomerate the elements at every coarsening step. If η gets large, a potential remedy might be the AMLI stabilization procedure (cf. Vassilevski* [16]*) which is like W–cycle or even more cycles. Approaches to rigorously study the convergence of the underlined AMG method can draw on the existing analytical tools for geometric MG convergence theory for finite element problems (see, e.g., the book by Bramble* [4]*). In the present paper we do not deal with multilevel convergence results.*

Remark 3.3. *One can actually apply the same interpolation procedure on agglomerated elements using it recursively to fine–grid element matrices coming from non–symmetric elliptic operator like convection–diffusion, e.g., $\mathcal{L}u \equiv -\operatorname{div}(\epsilon\nabla u) + \underline{b}\cdot\nabla u$. In Fig. 3 and 2 a coarse basis function is shown (face and rotated) using four levels of coarsening procedure for constant convection field $b_1 = 1$, $b_2 = -0.5$ and $\epsilon = 0.1$. Note also that in this case of convection–diffusion operator the basis functions computed on the coarse levels by the proposed AMGe method will provide partition of unity (as in the symmetric operator case), and hence the coarse spaces will admit a certain weak approximation property. The same applies for the so–called streamline diffusion operator $\mathcal{L}_\delta u \equiv -\operatorname{div}((\epsilon + \delta\underline{b}\ \underline{b}^T)\nabla u) + \underline{b}\cdot\nabla u$ where δ is a mesh–dependent parameter.*

Remark 3.4. *We finally remark, that the presented AMGe method can be used in the so–called "homogenization" procedures to generate averaged coarse problems from problems on computationally unfeasible highly refined meshes and possibly with oscillatory coefficients (cf., e.g.,* [11] *and references therein, see also* [10]*). The*

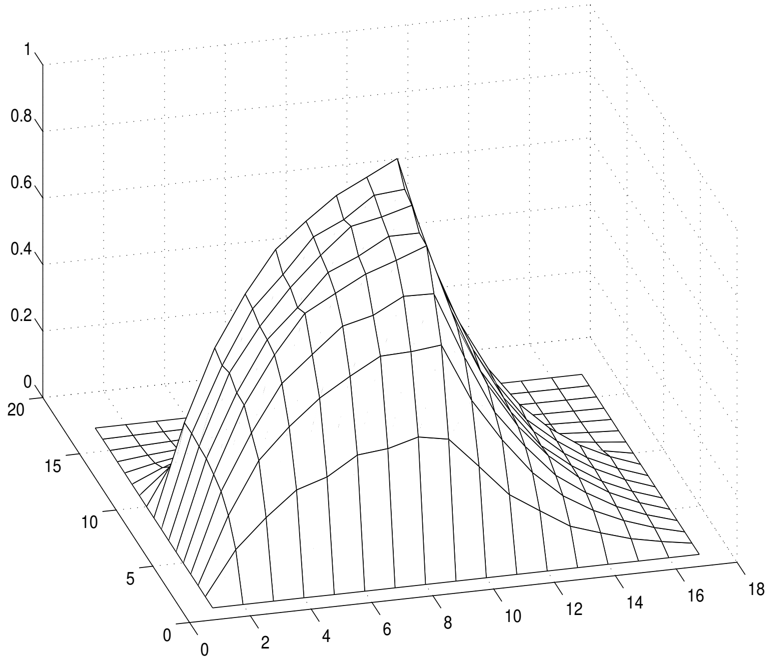

FIGURE 2. AMGe constructed "minimum energy" coarse basis function for convection–diffusion operator.

difference that we see here is that our coarsening procedure is local. We require the solution of small local problems (involving a few elements) rather than large subdomain solves in order to compute the effective coarse grid basis functions (or coarse–grid element matrices).

4. Algorithms for element agglomeration

This section introduces the algorithm we have used in selecting the coarse grid agglomerates. The algorithm relies of the faces and edges of the original elements $\{e\}$; to simplify the discussion, we will focus mainly on 2–d elements (i.e., having faces and vertices only). The method is based on the face–face graph of the fine grid elements (i.e. face f_1 and f_2 are neighbors if they share a common vertex) and uses an integer weight $w(f)$ for each face f. The eliminated faces f will have a weight $w(f) = -1$.

Algorithm 4.1 (Element agglomeration based on face–face graph)**.**

- **initiate**: *Set* $w(f) = 0$ *for all faces* f;
- **global search**: *Find a face* f *with maximal* $w(f)$; *set* $E = \emptyset$;
  1. *Set* $E = E \cup e_1 \cup e_2$, *where* $e_1 \cap e_2 = f$, *and set* $w_{\max} = w(f)$, $w(f) = -1$;
  2. *Increment* $w(f_1) = w(f_1) + 1$ *for all faces* f_1 *such that* $w(f_1) \neq -1$ *and* f_1 *is a neighbor of* f;

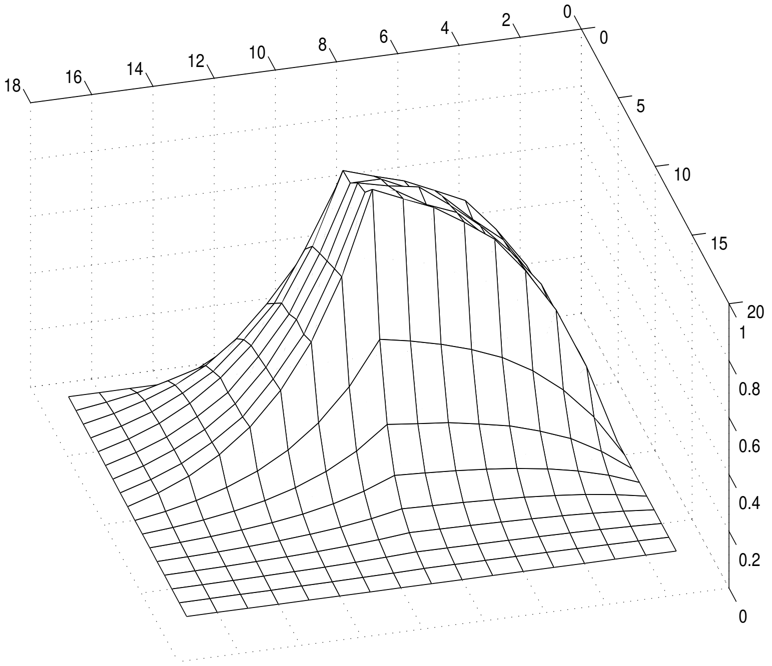

FIGURE 3. AMGe constructed "minimum energy" coarse basis function for convection–diffusion operator, rotated.

3. *Increment* $w(f_2) = w(f_2) + 1$ *for all faces* f_2 *such that* $w(f_2) \neq -1$*,* f_2 *is a neighbor of* f*, and* f_2 *and* f *are faces of a common element;*
4. *From the neighbors of* f*, choose a face* g *with a maximal* $w(g)$*; if* $w(g) \geq w_{\max}$*, set* $f = g$*, and go to step (1);*
5. *If all neighbors of* f *have smaller weight than* $w_{\max}$*, the agglomerated element* E *is complete; set* $w(g) = -1$ *for all faces of the elements* e *contained in* E*; go to step* **global search***;*

This algorithm tends to produce nicely matched agglomerated elements; and produces standard multigrid coarsening (up to boundary effects) for structured grid problems using linear or bilinear elements. See Figures 4 and 5 for the results of this procedure applied to a uniform triangular mesh after one and two agglomeration steps, respectively. The setup cost of the algorithm is linear, i.e., proportional to the total number of dofs. The algorithm is easily implemented using, for example, double linked lists.

Figures 6 and 7 show the results of the algorithm for several unstructured problems. Figures 8, 10, 12, and 14 show fine unstructured grids using triangular elements, the agglomerated elements are shown in Figures 9, 11, 13, and 15, respectively. The latter are the actual grids on which the first set of numerical test was performed.

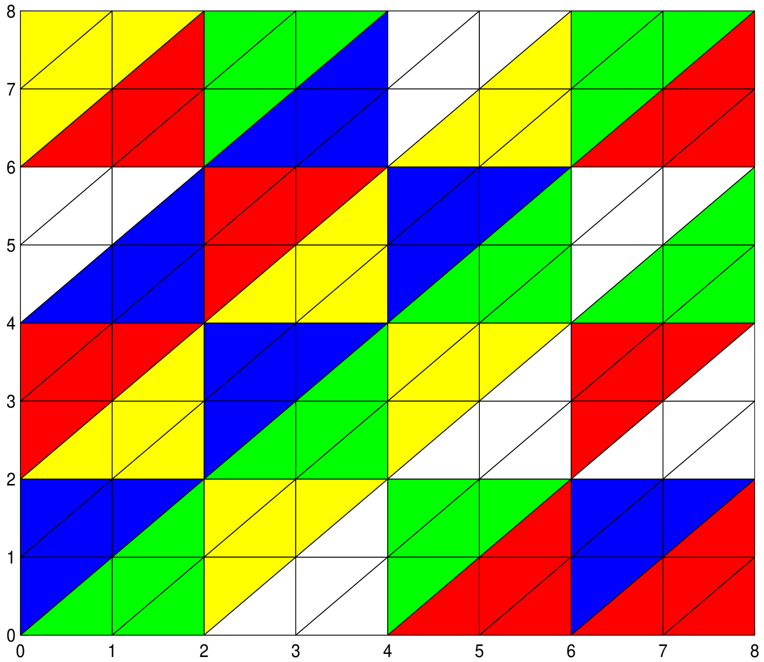

FIGURE 4. Agglomerated elements for structured triangular mesh: one step of agglomeration.

In 3–d one has the opportunity to introduce edges. Then one may construct more refined agglomeration algorithms that exploit this additional topological information; namely, the edge–edge and edge–face graphs. This information, however, has not been utilized in the present paper.

It is important to note that the above algorithm does not take into account any matrix entries while agglomerating the elements. For structured grid problems with anisotropy, it will produce full-coarsening. To produce semi-coarsening for such problems, one can introduce barriers. This can be implemented by assigning to each face another (binary) weight $a(f) = \begin{cases} 0, & \text{acceptable,} \\ 1, & \text{unacceptable.} \end{cases}$ To prevent agglomeration through a face f, one can simply set $a(f) = 1$ and then in step 4. of Algorithm 4.1. one searches for a face g a neighbor to f which is with a maximal weight $w(g)$ and if $a(g) = 1$ (i.e., unacceptable) one looks for an acceptable face g_a (neighbor to f) such that $w(g_a) = w(g)$. If such a face does not exist the agglomeration step is terminated and the agglomerated element E is ready.

The way we have put barriers on the faces is based on the element matrices; namely, given a face $f = e_1 \cap e_2$ assemble $A_{e_1 \cup e_2}$ and ask if the dofs on f can be well interpolated from the rest of the dofs in $e_1 \cup e_2$. If the resulting measure of interpolation quality is reasonable, we say that the face f is acceptable; otherwise we label f as unacceptable by initializing $a(f) = 1$ to prevent agglomeration of e_1

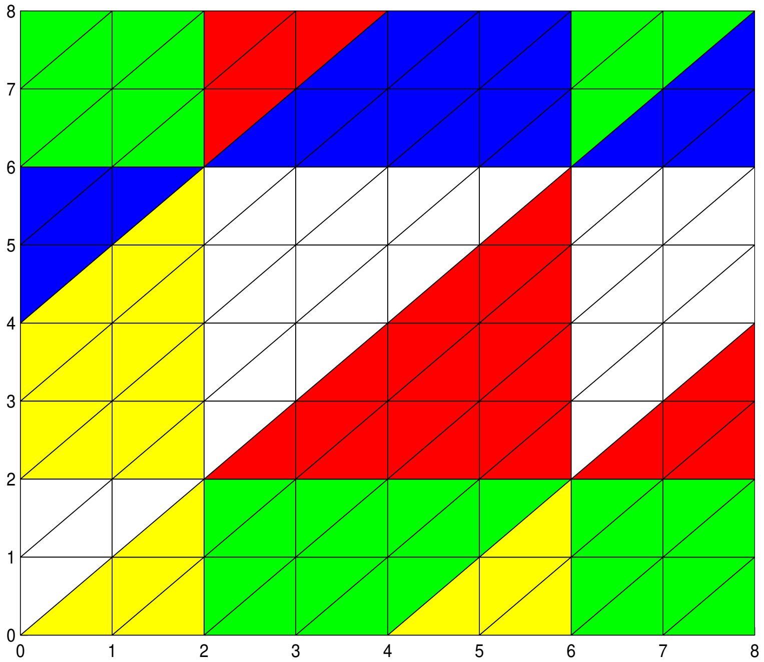

FIGURE 5. Agglomerated elements for structured triangular mesh: two steps of agglomeration.

and e_2. To implement this approach, one must be able to access the quality of the interpolation for the dofs on f. A measure of interpolation quality was proposed in [7]. In our setting, it can be reformulated as follows. Given the interpolation mapping P defined by interpolating dofs on f from the rest of the dofs in $e_1 \cup e_2$, define the quadratic form (or matrix) W_{ff} for vectors on f by,

$$\mathbf{v}_f^T W_{ff} \mathbf{v}_f = \inf_{\mathbf{v}_c} (\mathbf{v} + P\mathbf{v}_c)^T A_{e_1 \cup e_2} (\mathbf{v} + P\mathbf{v}_c); \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_f \\ 0 \end{bmatrix} \begin{array}{l} \}\text{dofs on } f, \\ \}e_1 \cup e_2 \setminus f. \end{array}$$

Then the measure of interpolation quality (denoted by M_1 in [7]) is,

$$m_P = \frac{1}{\lambda_{\min}[D_{ff}^{-1} W_{ff}]}, \tag{4.1}$$

where, D_{ff} is, for example, the diagonal of $A_{e_1 \cup e_2}$ restricted to f. Small m_P indicates good quality interpolation; interpolation well approximates functions with low energy. In finite element notation, small m_P means that the functions v_c from the coarse space can approximate well the fine–grid functions v in a weighted L^2–norm $\|.\|_0$. To show this, let m be a bound such that;

$$\inf_{v_c} \|v - v_c\|^2_{0,\, e_1 \cup e_2} \leq m \; a_{e_1 \cup e_2}(v, v) \quad \text{for all } v: \;\; v|_{\mathcal{D}_c} = v_c|_{\mathcal{D}_c}\,. \tag{4.2}$$

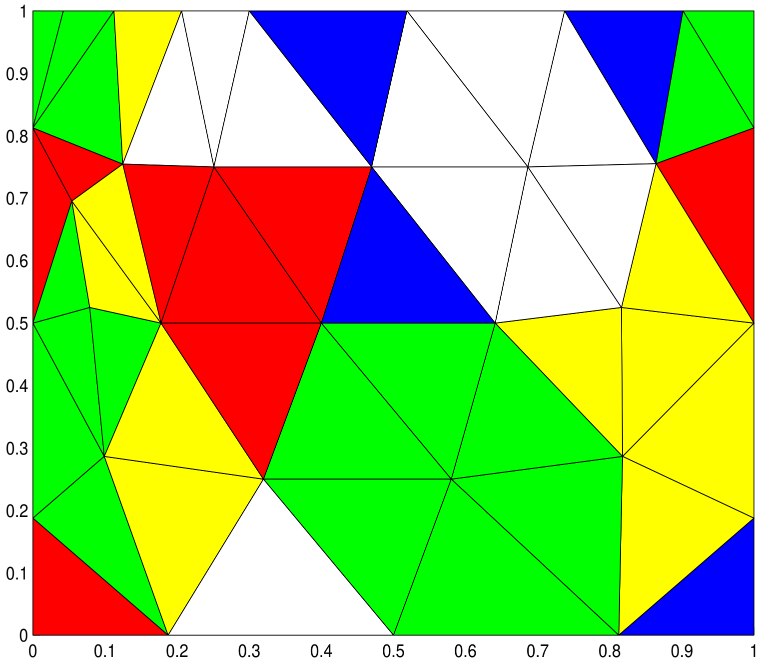

FIGURE 6. Agglomerated elements: rectangular domain with unstructured triangular elements.

This is equivalent (letting $v = v_f + v_c$ above), to

$$\|v_f\|^2_{0,\, e_1\cup e_2} \leq m \inf_{v_c} a_{e_1\cup e_2}(v_f + v_c,\ v_f + v_c) \quad \text{for all } v_f :\ v_f|_{\mathcal{D}_c} = 0.$$

In vector notation, this becomes,

$$\begin{aligned}(\mathbf{v}_f)^T D_{ff}\mathbf{v}_f \ &\leq m \inf_{\mathbf{v}_c} (\mathbf{v} + P\mathbf{v}_c)^T A_{e_1\cup e_2}(\mathbf{v} + P\mathbf{v}_c), \text{ for all } \mathbf{v} = \begin{bmatrix} \mathbf{v}_f \\ 0 \end{bmatrix} \\ &= m\ (\mathbf{v}_f)^T W_{ff}\mathbf{v}_f, \text{ for all } \mathbf{v}_f.\end{aligned}$$

This with the best choice of m, leads to the definition (4.1) of the measure m_P. It is clear, from (4.2), that smaller m_P corresponds to better interpolation quality.

Remark 4.1. *One can actually compute the minimum,*

$$\mathbf{v}_f^T W_{ff}\mathbf{v}_f = \min_{\mathbf{v}_c} (\mathbf{v} + P\mathbf{v}_c)^T A_{e_1\cup e_2}(\mathbf{v} + P\mathbf{v}_c), \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_f \\ 0 \end{bmatrix} \begin{matrix} \} & \textit{dofs on } f \\ \} & (e_1 \cup e_2)\setminus f \end{matrix}.$$

One has, with $A := A_{e_1\cup e_2}$ *and* $\mathbf{v}_c := t\mathbf{v}_c$ *for any* $t \in \mathcal{R}$*,*

$$(\mathbf{v} + tP\mathbf{v}_c)^T A(\mathbf{v} + tP\mathbf{v}_c) = \mathbf{v}^T A\mathbf{v} + 2t\mathbf{v}^T AP\mathbf{v}_c + t^2(P\mathbf{v}_c)^T AP\mathbf{v}_c.$$

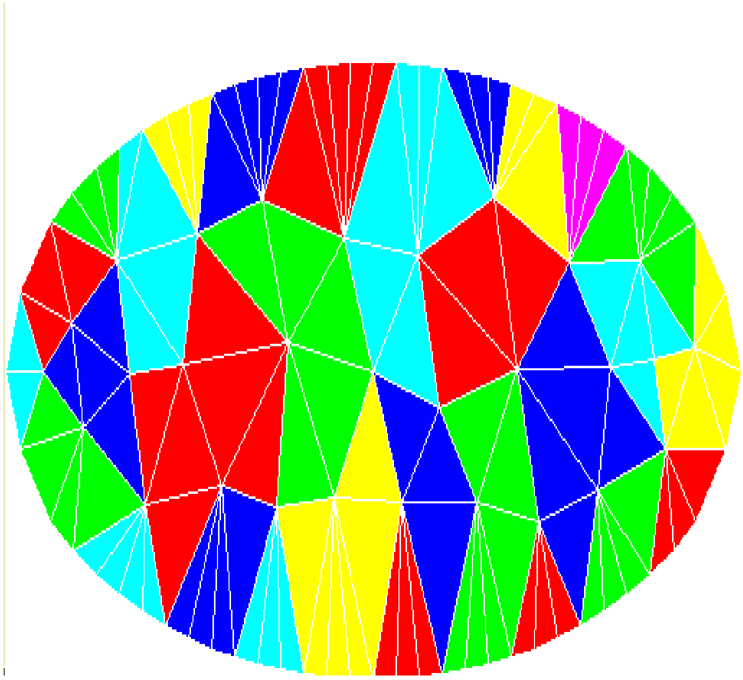

FIGURE 7. Agglomerated elements: elliptical domain with triangular elements.

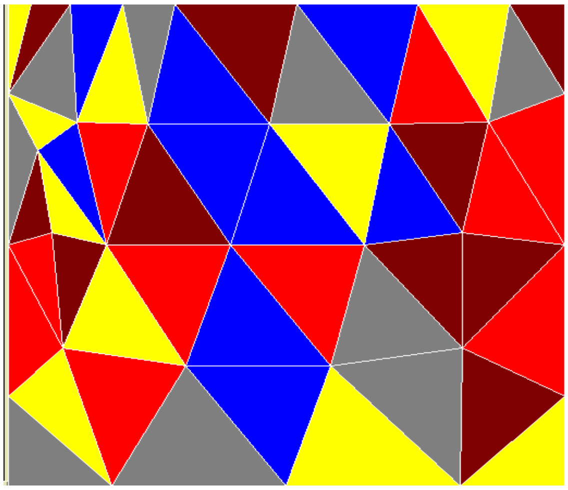

FIGURE 8. Fine elements: rectangular domain with 48 unstructured triangular elements.

The minimum with respect to t is achieved for $t = -\frac{\mathbf{v}^T AP\mathbf{v}_c}{(P\mathbf{v}_c)^T AP\mathbf{v}_c}$ and equals,

$$\mathbf{v}^T A\mathbf{v} - \frac{(\mathbf{v}^T AP\mathbf{v}_c)^2}{(P\mathbf{v}_c)^T AP\mathbf{v}_c}.$$

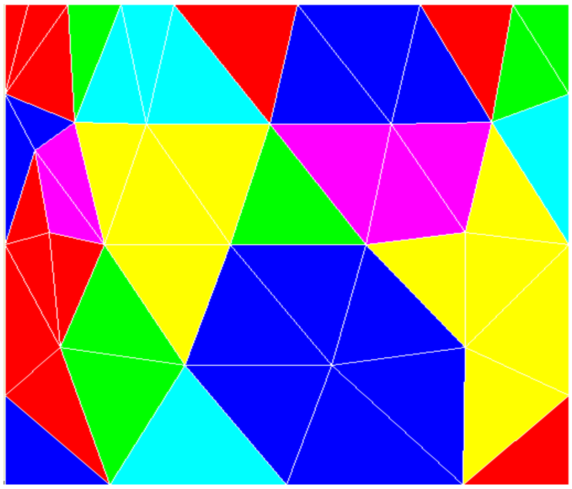

FIGURE 9. Agglomerated elements: rectangular domain with unstructured triangular elements.

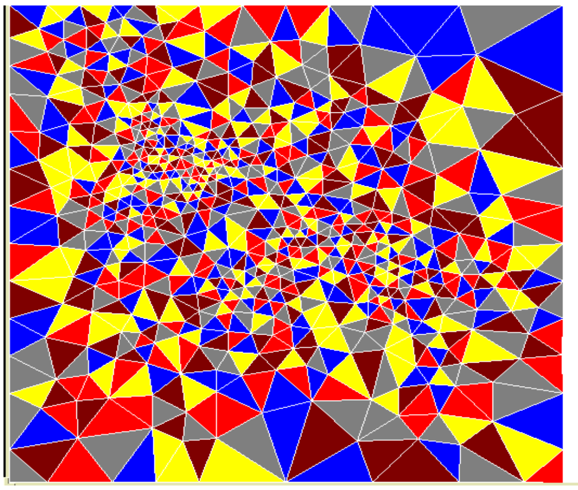

FIGURE 10. Fine elements: rectangular domain with 1001 unstructured triangular elements.

Hence,

$$\mathbf{v}_f^T W_{ff} \mathbf{v}_f = \min_{\mathbf{v}_c} \left(\mathbf{v}_f^T A_{ff} \mathbf{v}_f - \frac{(\mathbf{v}^T A P \mathbf{v}_c)^2}{(P\mathbf{v}_c)^T A P \mathbf{v}_c} \right), \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_f \\ 0 \end{bmatrix}.$$

I.e., $\mathbf{v}_f^T W_{ff} \mathbf{v}_f \leq \mathbf{v}_f^T A_{ff} \mathbf{v}_f$. *Here,* A_{ff} *represents the* f-f *block of* $A := A_{e_1 \cup e_2}$ *(see equation* (4.3) *below). Note that if there is a* $\mathbf{v}_c$ *such that* $(AP\mathbf{v}_c)|_f = 0$, *then* $\mathbf{v}_f^T W_{ff} \mathbf{v}_f = \mathbf{v}_f^T A_{ff} \mathbf{v}_f$. *The latter is true also for the so–called "optimal"* P, *i.e.,*

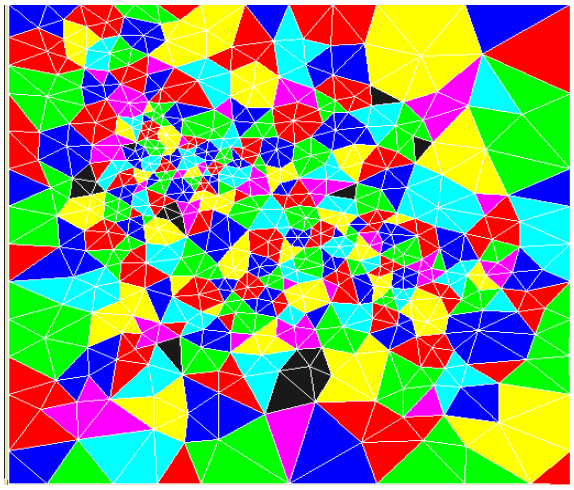

FIGURE 11. Agglomerated elements: rectangular domain with unstructured triangular elements.

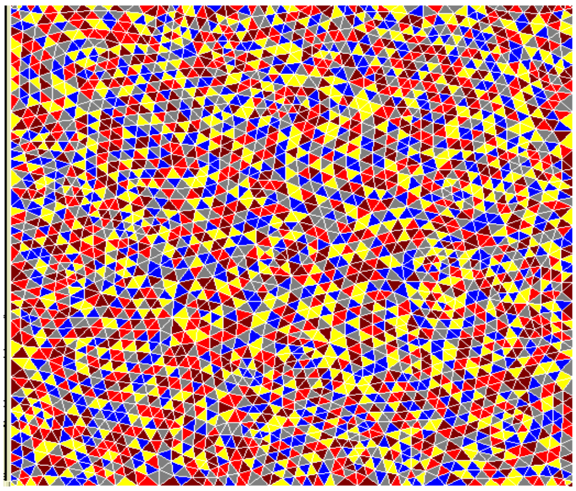

FIGURE 12. Fine elements: rectangular domain with 4 016 unstructured triangular elements.

such that $P = -A_{ff}^{-1} A_{fc}$, *where* $A_{e_1 \cup e_2}$ *is partitioned as follows:*

$$A_{e_1 \cup e_2} = \left[\begin{array}{cc} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{array} \right] \begin{array}{l} \} \quad \textit{dofs on } f \\ \} \quad (e_1 \cup e_2) \setminus f \end{array} . \tag{4.3}$$

In that case, $m_P = \frac{1}{\lambda_{min}\left[D_{ff}^{-1} A_{ff}\right]}$.

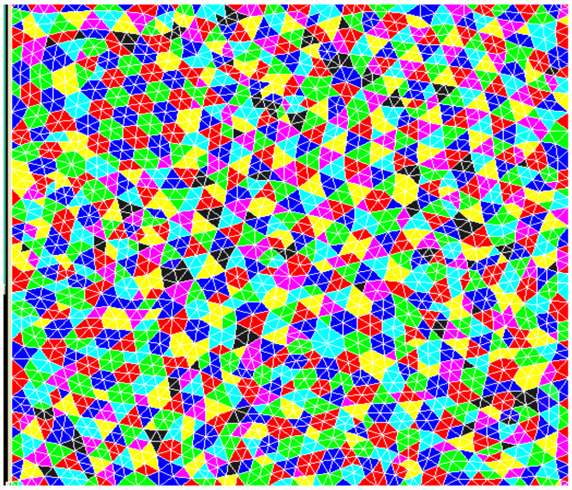

FIGURE 13. Agglomerated elements: rectangular domain with unstructured triangular elements.

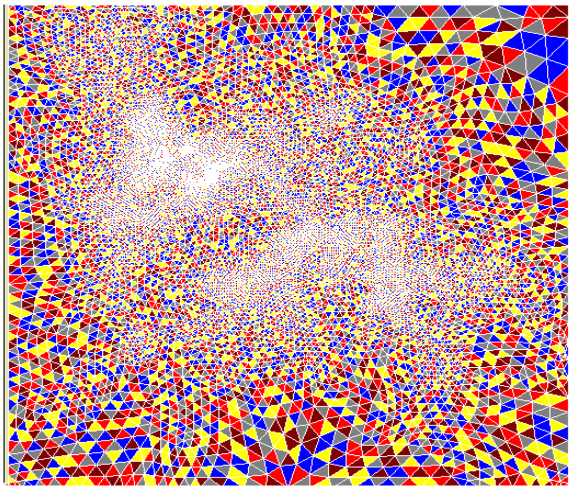

FIGURE 14. Fine elements: rectangular domain with 16 000 unstructured triangular elements.

Remark 4.2. *Note that if instead of D_{ff} one uses in* (4.1) *the principal submatrix A_{ff} of A corresponding to the fine dofs that are not coarse, then $m_P = \frac{1}{1-\gamma^2}$ where $\gamma \in [0, 1)$ stands for the cosine of the abstract angle between the coarse space $V_c = \{\mathbf{v}^c = P\mathbf{v}_c\}$ and its hierarchical complement $V_f = \left\{\mathbf{v}^f = \begin{bmatrix} \mathbf{v}_f \\ 0 \end{bmatrix}\right\}$. The angle is*

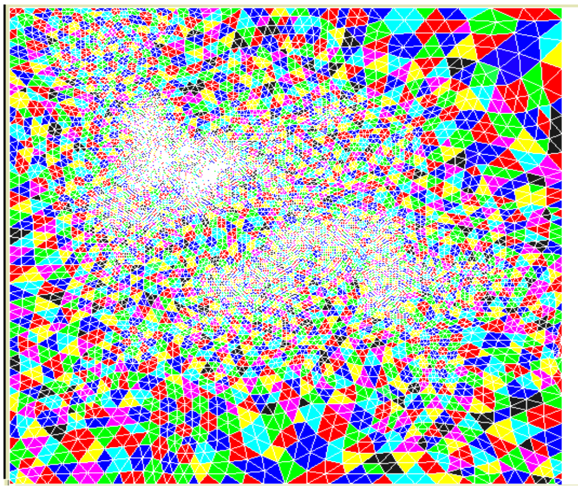

FIGURE 15. Agglomerated elements: rectangular domain with unstructured triangular elements.

measured in energy inner product, i.e.,

$$(\mathbf{v}^f)^T A_{e_1 \cup e_2} \mathbf{v}^c \leq \gamma \ \sqrt{(\mathbf{v}^f)^T A_{e_1 \cup e_2} \mathbf{v}^f} \sqrt{(\mathbf{v}^c)^T A_{e_1 \cup e_2} \mathbf{v}^c}, \textit{ for all } \mathbf{v}^f \in V_f, \ \mathbf{v}^c \in V_c.$$

For a proof of the relation $m_P = \frac{1}{1-\gamma^2}$ see, e.g., Vassilevski [16].

Instead of m_P one can use γ as a measure of the interpolation quality. Then small γ will correspond to small m_P and hence to good quality interpolation, whereas γ close to one will imply large m_P and hence poor quality interpolation.

In following example, we will use γ to define a measure for strength on connections between neighboring elements and thus label faces as acceptable or unacceptable. Consider two fine elements e_1 and e_2 sharing a face f as shown in Figure 16. Let i_{F_2} be an interpolation rule for dof x_3 from x_1 and x_5, and i_{F_1} be an interpolation rule for dof x_4 from x_2 and x_6, these could be constructed as proposed in the previous section. For 2–d scalar elliptic problems with constant coefficients, these are linear interpolants along the faces F_1 and F_2 treating x_1, x_2, x_5 and x_6 as coarse–grid nodes and x_3 and x_4 as complementary to the coarse–grid, fine–grid nodes. Then, given a coarse function v_c defined at the nodes x_1, x_2, x_5 and x_6, the mapping $P_c^f v_c = \begin{cases} i_{F_1} v_c, & x = x_4, \\ i_{F_2} v_c, & x = x_3, \end{cases}$ defines a coarse–to–fine prolongation operator.

Let $E = e_1 \cup e_2$ and let A_E be the assembled matrix corresponding to E. Given a coarse grid vector $\mathbf{v}_c$, let $\widehat{v}_c = P_c^f v_c$ be its representation on the fine–grid. Then the local fine–grid space is decomposed as $P_c^f v_c \oplus v_f^0$, where v_f^0 are the fine–grid functions which vanish on the coarse–grid. As mentioned, the cosine $\gamma \in [0, 1)$ of the angle between these components can be used to measure a strength of connection between e_1 and e_2 with respect to the given matrix A_E (or pair of element matrices A_{e_1} and

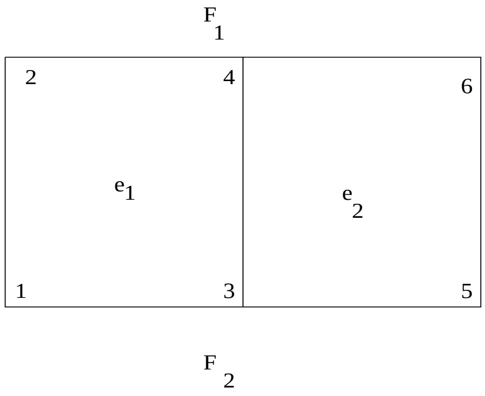

FIGURE 16. Neighboring elements e_1 and e_2 with a common face $f = \{x_3, x_4\}$; the nodes x_1, x_2, x_5 and x_6 are viewed as coarse–grid nodes.

A_{e_2} that correspond to the pair of elements e_1 and e_2). Recall the constant γ is defined as the best constant in the strengthened Cauchy inequality

$$a_E(\widehat{v}_c,\ v_f^0) \leq \gamma\ \sqrt{a_E(\widehat{v}_c,\ \widehat{v}_c)}\ \sqrt{a_E(v_f^0,\ v_f^0)}, \quad \text{for all } \widehat{v}_c,\ v_f^0. \tag{4.4}$$

To write this inequality in matrix–vector notation, let

$$P = \left[\left[\begin{array}{c} I \\ 0 \end{array} \right], P_c^f \right],$$

and $\widehat{A}_E = P^T A_E P$. Consider the following two–by–two blocking of $\widehat{A}_E$,

$$\widehat{A}_E = \left[\begin{array}{cc} A_{E;\, ff} & \widehat{A}_{E;\, fc} \\ \widehat{A}_{E;\, cf} & A_{E;\, cc} \end{array} \right] \begin{array}{l} \}\ \text{complementary fine–grid nodes; i.e., } x_3,\ x_4 \\ \}\ \text{coarse nodes; i.e., } x_1, x_2, x_5,\ x_6. \end{array}$$

Note that $A_{E;\, cc}$ is the resulting coarse matrix corresponding to E. Then the strengthened Cauchy inequality (4.4) reads:

$$\mathbf{v}_c^T \widehat{A}_{E;\, cf} \mathbf{v}_f^0 \leq \gamma\ \sqrt{\mathbf{v}_c^T A_{E;\, cc} \mathbf{v}_c}\ \sqrt{\mathbf{v}_f^{0\,T} A_{E;\, ff} \mathbf{v}_f^0}, \quad \text{for all } \mathbf{v}_c,\ \mathbf{v}_f^0.$$

A way to compute γ is to find the largest eigenvalue $m = \lambda_{\max} \geq 1$ of the generalized eigenvalue problem

$$A_{E,\, cc} \mathbf{q} = \lambda S_{E,\, f} \mathbf{q},$$

where $S_{E,\, f}$ is the Schur complement of $\widehat{A}_E$ on f, i.e., $S_{E,\, f} = A_{E,\, cc} - \widehat{A}_{E;\, cf} \left(A_{E;\, ff} \right)^{-1} \widehat{A}_{E;\, fc}$. Then $\gamma = \sqrt{1 - \frac{1}{m}}$.

Definition 4.1 (Strongly connected elements)**.** *We call e_1 and e_2 strongly connected if γ is close to zero, i.e., when the resulting local coarse space is almost orthogonal to its complementary (the so–called two–level hierarchical complementary) space.*

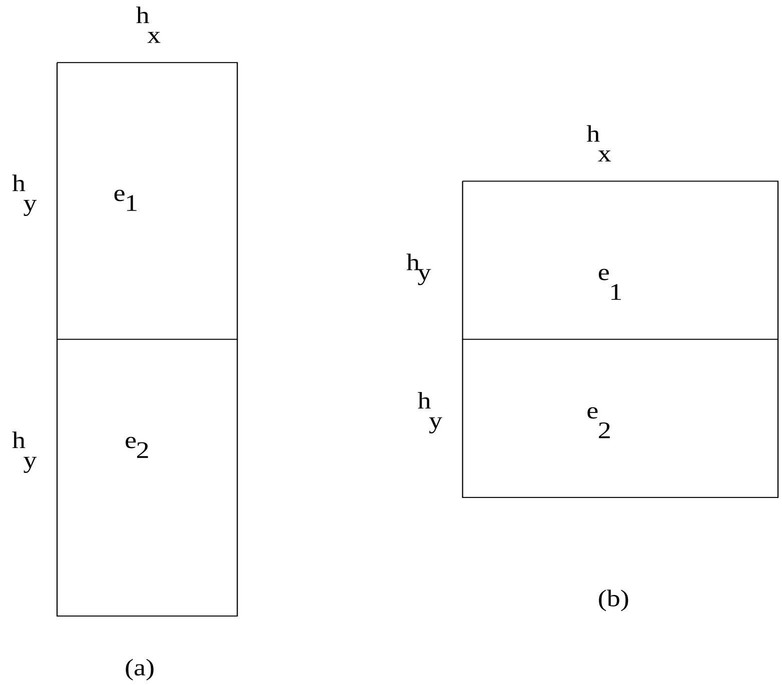

FIGURE 17. Neighboring elements e_1 and e_2; (a) $h_x < h_y$, (b) $h_x > h_y$.

Algorithm 4.1 can be modified to agglomerate only strongly connected elements. One would set a threshold α and label a face f unacceptable if $\gamma > \alpha$ by initializing $a(f) = 1$.

4.1. **Examples of γ.** We conclude this section with examples showing that this definition of strongly connected elements can lead to the correct semicoarsening for anisotropic problems. Consider the model second order elliptic bilinear form, which restricted to an element e reads,

$$a_e(\varphi, \psi) = \int_e \left(\frac{\partial \varphi}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial \varphi}{\partial y}\frac{\partial \psi}{\partial y} \right) \, dx \; dy. \tag{4.5}$$

Consider two vertically adjacent rectangular elements (see Fig. 17) and bilinear test functions. Consider the cases:

(a) anisotropic elements $h_x < h_y$; $h_x = 0.1h_y$, $\gamma = 0.8649$; $h_x = 0.01h_y$, $\gamma = 0.8660$; these values of γ indicate that the elements are weakly connected and one should not agglomerate them.
(b) anisotropic elements $h_x > h_y$; $h_x = 10h_y$, $\gamma = 0.1698$; $h_x = 100h_y$, $\gamma = 0.0173$; This example shows that since γ is close to zero, that the elements are strongly connected and hence one should agglomerate this pair of elements.
(c) for comparison, if $h_x = h_y$, $\gamma = 0.7746$ (or $\gamma^2 = \frac{3}{5}$).

TABLE 1. Two–grid convergence results; unstructured triangular grid; Laplace operator, Gauss–Seidel smoother.

grid #	1	2	3	4
# fine elements	48	1 001	4 016	16 016
# coarse elements	20	242	1 016	3 859
# fine dof	35	523	2 085	8 095
# coarse dof	27	281	1 083	3 515
# iterations	7	9	8	8
ϱ	0.159	0.320	0.256	0.260

Thus, this measure correctly leads to coarsening only in the direction of small mesh size.

5. Numerical experiments

In this section we present some preliminary numerical results that show the potential of the proposed element agglomeration AMGe method.

We have tested the two–grid method with the coarse–grid obtained using the agglomeration algorithm described in Section 4. After the coarse degrees of freedom were selected the interpolation mapping is constructed as described in Section 2. We used one forward Gauss–Seidel iteration as a pre–smoother and one backward Gauss–Seidel iteration for a post–smoothing. The stopping criterion was relative reduction of the residual ℓ^2–norm by a factor of 10^{-6}.

We tested two set of problems:

- Poisson equation discretized on square domain on four "unstructured" rectangular grids shown in Fig. 8, 10, 12, and 14, and and the respective grids with agglomerated elements are shown in Fig. 9, 11, 13, and 15. Dirichlet boundary conditions were imposed, and the results are collected in the Table 1.
- Elasticity equation coming from minimizing the quadratic functional discretized with square bilinear elements.

$$\int_{\Omega} \left[\frac{1+\nu}{2}(\partial_x u + \partial_y v)^2 + \frac{1-\nu}{2}(\partial_x u - \partial_y v)^2 + \frac{1-\nu}{2}(\partial_y u + \partial_x v)^2 \right] \, dx dy. \tag{5.1}$$

  Here $\nu = \frac{1}{3}$. Again, Dirichlet boundary conditions were imposed, and these results are in Table 2.

One notices the similar convergence factors ϱ and # iterations for Poisson and elasticity problems.

Acknowledgments. The present authors would like to acknowledge the stimulating discussions with the authors of [7] without which this research direction may not have been pursued. The unstructured triangular grids used in our numerical tests were provided by Dr. Ludmil Zikatanov. Finally, the careful reading of the paper by the referees is gratefully appreciated.

TABLE 2. Two–grid convergence results; structured rectangular grid; elasticity operator, Gauss–Seidel smoother.

grid #	1	2	3	4
# fine elements	400	900	1600	2500
# coarse elements	118	253	438	673
# fine dof	882	1922	3362	5202
# coarse dof	314	624	1034	1544
# iterations	9	9	9	9
ϱ	0.251	0.245	0.254	0.248

REFERENCES

[1] O. Axelsson, Iterative Solution Methods, Cambridge University Press, New York, 1994.

[2] R. E. Bank, Hierarchical bases and the finite element method, Acta Numerica (1996), vol. **5**, Cambridge University Press, pp. 1–43.

[3] R. E. Bank and T. Dupont, Analysis of a two–level scheme for solving finite element equations, Report CNA-159, Center for Numerical Analysis, The University of Texas at Austin, 1980.

[4] J. H. Bramble, Multigrid methods, *Pitman Research Notes in Mathematical Series*, # 234, Longman Scientific & Technical, UK, 1995 (second edition).

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

[6] A. Brandt, S. McCormick and J. W. Ruge, Algebraic multigrid (AMG) for sparse equations, in: (D. J. Evans, ed.) *Sparsity and its applications (Loughborough 1983)*, Cambridge Univ. Press, Cambridge, 1985, pp. 257-284.

[7] 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 J. Sci. Comput. (to appear).

[8] T. F. Chan and B. F. Smith, Domain decomposition and multigrid algorithm for elliptic problems on unstructured meshes, in: "DOMAIN DECOMPOSITION METHODS IN SCIENTIFIC AND ENGINEERING COMPUTING", Proceedings of the 7th International Conference on Domain Decomposition Methods, AMS, Providence, Rhode Island, Contemporary Mathematics 180(1994), pp. 175–189.

[9] T. F. Chan, J. Xu, and L. T. Zikatanov, An agglomeration multigrid method for unstructured grids, in: "DOMAIN DECOMPOSITION METHODS 10", Proceedings of the 10th International Conference on Domain Decomposition Methods, AMS, Providence, Rhode Island, Contemporary Mathematics 218(1998), pp. 67–81.

[10] R. P. Fedorenko, Finite superelements method, in: "Recent Advances in Numerical Methods and Applications, II", Proceedings of the Fourth International Conference, NMA'98, Sofia, Bulgaria, (Iliev, Kaschiev, Margenov, Sendov and Vassilevski eds.), World Scientific, Singapore, 1999, pp. 16–26.

[11] T. Y. Hou, X.-H. Wu, and Z. Cai, Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients, Math. Comp. **68**(1999), 913–943.

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

[13] J. W. Ruge and K. Stüben, Algebraic multigrid, in: (S. McCormick, ed.) *Multigrid Methods*, Philadelphia, PA, 1987, pp. 73–130.

[14] K. Stüben, Algebraic Multigrid (AMG): An Introduction with Applications, GMD Report **53**(1999), GMD - Forschungszentrum Informationstechnik GmbH, Schloss Birlinghoven, Sankt Augustin, Germany.
[15] P. Vanek, J. Mandel, and M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing 56(1996), 179-196.
[16] P. S. Vassilevski, On two ways of stabilizing the HB multilevel methods, SIAM Review 39(1997), 18–53.
[17] W. L. Wan, T. F. Chan and B. F. Smith, An energy minimizing interpolation for robust multigrid methods, SIAM J. Sci. Comput. 21(2000), 1632-1649.

Center for Applied Scientific Computing, UC Lawrence Livermore National Laboratory, Mail Stop L-560, 7000 East Avenue, Livermore, CA 94550, U.S.A.
E-mail address: jjones@llnl.gov, vassilevski1@llnl.gov