LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-JRNL-208036

Multiple Vector Preserving Interpolation Mappings in Algebraic Multigrid

P. S. Vassilevski, L. T. Zikatanov

November 16, 2004

SIAM Journal on Matrix Analysis with Applications

MULTIPLE VECTOR PRESERVING INTERPOLATION MAPPINGS IN ALGEBRAIC MULTIGRID

PANAYOT S. VASSILEVSKI, AND LUDMIL T. ZIKATANOV

Abstract. We propose algorithms for the construction of AMG (algebraic multigrid) interpolation mappings such that the resulting coarse space to span (locally and globally) any number of a priori given set of vectors. Specific constructions in the case of element agglomeration AMG methods are given. Some numerical illustration is also provided.

1. Introduction

We are given a s.p.d. sparse matrix $A: \mathbb{R}^n \mapsto \mathbb{R}^n$, which we assume corresponds to a finite element discretization of a 2nd order elliptic PDE. No element (or grid) hierarchy is assumed, so we can handle general unstructured meshes.

Our goal is to construct a two–grid, and by recursion, multigrid methods (or preconditioners) to effectively solve systems of equations with the matrix A. A typical two–grid method involves a smoother M (for example, Gauss–Seidel) and a coarse–grid correction. The latter gives rise to a projection matrix $\pi_A = P(P^TAP)^{-1}P^TA$, based on a interpolation matrix $P: \mathbb{R}^{n_c} \mapsto \mathbb{R}^n$, $n_c < n$. Thus, a two–grid error iteration matrix E_{TG}, takes the form,

$$E_{TG} = (I - M^{-T}A)(I - \pi_A)(I - M^{-1}).$$

Alternatively, one can define a two–grid preconditioner

$$B_{TG}^{-1} = [I,\ P]\, \widehat{B}_{TG}^{-1}\, [I,\ P]^T,$$

based on the block–factored matrix

$$\widehat{B}_{TG} = \begin{bmatrix} M & 0 \\ AP & I \end{bmatrix} \begin{bmatrix} \left(M + M^T - A\right)^{-1} & 0 \\ 0 & P^TAP \end{bmatrix} \begin{bmatrix} M^T & P^TA \\ 0 & I \end{bmatrix}.$$

By a straightforward computation, one can see that $E_{TG} = I - B_{TG}^{-1}A$.

In this paper we will focus on the construction of the interpolation mapping P. For matrices A coming from finite element discretization of 2nd order elliptic PDEs, a classical choice of P is that P has row-sum of ones, that is, P interpolates constants

Date: October 17, 2004–beginning; Today is November 3, 2004.

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

Key words and phrases. multiple vector preserving interpolation, algebraic multigrid, element agglomeration AMGe.

This work was performed under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

exactly. In matrix form, one has, for $\mathbf{1}_c = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{n_c}$ and $\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{n}$, that $P\mathbf{1}_c = \mathbf{1}$. One may generalize the above minimal requirement, such that for a given set of vectors $\mathbf{v}_k$, $k = 1, \ldots, m$, one has that $\text{Range}(P) \supset \text{Span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$.

We will adopt the finite element notion of "degrees of freedom" (or dofs), which is equivalent to row and column indices of A. The row indices of P are also referred to as "fine" (or fine–grid) dofs, whereas the column indices to as "coarse" (or coarse–grid) dofs. Finally, there will be natural embedding of the coarse dofs into the fine–grid dofs, in the sense that the coarse dofs $\mathcal{N}_c$ can be viewed as subset of the set of fine–grid dofs $\mathcal{N}$. Then, a common form of P is

$$P = \begin{bmatrix} W \\ I \end{bmatrix} \begin{matrix} \} \\ \} \end{matrix} \begin{matrix} \mathcal{N} \setminus \mathcal{N}_c \\ \mathcal{N}_c \end{matrix} . \tag{1.1}$$

The above setting of "fitting" exactly several vectors by an interpolation mapping P has some potential difficulties. For example, let $\{\mathbf{v}_k\}_{k=1}^m$ be such that when restricted to a local set of indices Ω be linearly dependent, that is

$$\sum_{k=1}^{m} \alpha_k \, \mathbf{v}_k|_\Omega = 0.$$

A main example, that we will focus on throughout this paper, will be the case of finite element matrices A. In particular we will assume that A is assembled from a set of local element matrices $\{A_e\}_{e \in \mathcal{T}}$. Consider now the case of Ω being the set of coarse dofs from a union of elements, $E = \cup e$. Then, one has

$$\sum_{k=1}^{m} \alpha_k \, \mathbf{v}_k|_{E \cap \mathcal{N}_c} = 0. \tag{1.2}$$

In the element agglomeration AMGe method ([JV01]) sets like E are used to define coarse elements. In order to apply recursion one has to define coarse element matrices appropriately; namely, one constructs P such that it interpolates dofs from E only from coarse dofs also in E. Let P restricted to E be P_E and let A_E be the local matrix corresponding to E, assembled from the element matrices A_e for all elements e that form E. Thus, based on the above property of P, one has that each vector restricted to E is interpolated by the columns of P restricted to E, that is, by the columns of P_E. Therefore, based on (1.2), and the fact that $\mathbf{v}_k|_E \in \text{Range}(P_E)$, or more precisely $\mathbf{v}_k|_E = P_E \left(\mathbf{v}_k|_{E \cap \mathcal{N}_c} \right)$, one gets that

$$\sum_{k=1}^{m} \alpha_k \, \mathbf{v}_k|_E = P_E \left(\sum_{k=1}^{m} \alpha_k \, \mathbf{v}_k|_{E \cap \mathcal{N}_c} \right) = 0.$$

The latter implies, that the set of vectors $\{\mathbf{v}_k\}$ restricted to E (not only to the coarse dofs in E) are linearly dependent. This need not necessarily be true for the given set of vectors $\{\mathbf{v}_k\}$. That is, the task of constructing P to meet the main requirement of the element agglomeration AMGe may not have a solution (in the above setting). One more point in practice is, that for relatively small sets E, the vectors $\{\mathbf{v}_k\}$ may be

nearly linear dependent locally. This would make the coarse element matrix $P_E^T A_E P_E$ have a near null–vector which is a spurious one, due only to our construction. This feature of spurious local (near) null–vectors may cause unnecessary difficulties in solving the coarse–grid problem. All difficulties above may only happen if we want to fit several vectors by one and the same P (of the form (1.1)); if only one vector is to be preserved such difficulties do not arise.

Thus, an algorithm for the construction of P must be provided that accounts for such local (near) linear dependence of the given set of vectors $\{\mathbf{v}_k\}$. This is the main objective of the present paper. Our approach that overcomes the above outlined difficulty is to construct a set of P_k, for $k = 0, 1, ..., m$, one P_k at the time, such that the resulting P is simply the collection $P = [P_1, P_2, \ldots, P_m]$ and every new P_k is a hierarchical complement of the preceding ones $P_1, ..., P_{k-1}$. More precisely, we seek P of the form,

$$[P_1, P_2, ..., P_m] = \begin{bmatrix} W_{1,1} & W_{1,2} & \ldots & W_{1,m} \\ W_{2,1} & W_{2,2} & \ldots & I \\ \vdots & \vdots & \ldots & \vdots \\ W_{m,1} & I & \ldots & 0 \\ I & 0 & \ldots & 0 \end{bmatrix}. \tag{1.3}$$

By this choice of the structure of P one guarantees that P has full column rank and therefore the Galerkin coarse matrix $P^T AP$ is s.p.d. (since A is).

The actual construction of P is explained in Section 2 where the details on constructing the columns of P are presented. In Section 3, we describe the choice of the sparsity pattern of the columns of P (the W–blocks in (1.3)) in the setting of the element agglomeration AMGe from [JV01]. The final section contains some numerical results illustrating the performance of the two–grid and corresponding multilevel AMG methods based on interpolation mappings that interpolate several vectors $\{\mathbf{v}_k\}$. We used in the tests the coefficient vectors of liner functions 1, x, y (the coordinates of the nodes of a given finite element mesh). It appeared that for sufficiently aggressive coarsening, the use of several vectors to construct P is beneficial for the resulting AMG compared to the more classical approach when only one vector is used to construct P. If the coarsening is not as aggressive, as one can easily see, all geometrically smooth vectors, are pretty close to constant vectors restricted to small neighborhoods. The latter ("small neighborhoods") refers to the support of the coarse basis functions (or equivalently, to the non–zero entries of each column of P).

2. Construction of multiple vector preserving interpolation

The construction of P as in (1.3) consists of the following steps. We assume that a coarse grid set $\mathcal{N}_c$ of coarse dofs, which we view as a subset of the fine grid dofs $\mathcal{N} = \{1, 2, \ldots, n\}$, is chosen. Then we split $\mathcal{N}_c$ into m groups of non–overlapping

sets, $\mathcal{N}_m^c \cup \mathcal{N}_{m-1}^c \cup \cdots \cup \mathcal{N}_1^c = \mathcal{N}_c$. Each P_k admits the form

$$P_k = \begin{bmatrix} \star \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \} & \mathcal{N} \setminus \left(\cup_{i=1}^k \mathcal{N}_i^c\right) \\ \} & \mathcal{N}_k^c \\ \} & \mathcal{N}_{k-1}^c \\ & \vdots \\ \} & \mathcal{N}_1^c \end{matrix} .$$

Note that P_k has number of columns equal to $|\mathcal{N}_k^c|$ (cardinality of $\mathcal{N}_k^c$). In practice, one starts with a current set $\mathcal{N}_c = \mathcal{N}_1^c$ and at step $k > 1$ one augments it with a newly constructed set $\mathcal{N}_k^c \subset \mathcal{N} \setminus \left(\cup_{i=1}^{k-1} \mathcal{N}_i^c\right)$, i.e., $\mathcal{N}_c := \mathcal{N}_c \cup \mathcal{N}_k^c$.

To avoid the potential local (near) linear dependence, after a column P_k for a $k \geq 1$ of P is constructed, such that $P_k \mathbf{v}_k^c = \mathbf{v}_k$ one updates the remaining vectors by computing the residuals

$$\mathbf{v}_l := \mathbf{v}_l^{\text{new}} = \mathbf{v}_l - P_k \mathbf{v}_l^c, \ l = k+1, \ \ldots, \ m.$$

Here, $\mathbf{v}_l^c$ are the restrictions of $\mathbf{v}_l$ onto the kth coarse grid $\mathcal{N}_k^c$. The latter corresponds to the row indices of the identity block of P_k. A simple observation is, that at every step $k \geq 1$, since $\mathbf{v}_l \in \text{Span}\{\text{Range}(P_k), \ \mathbf{v}_l^{\text{new}}\}$, $l > k$, one has

$$\begin{aligned} \text{Span}\{\text{Range}(P_1), \ \ldots, \ \text{Range}(P_k), \ \mathbf{v}_{k+1}^{\text{new}}, \ \ldots, \ \mathbf{v}_m^{\text{new}}\} \supset \\ \text{Span}\{\mathbf{v}_1, \ \ldots, \ \mathbf{v}_k, \ \mathbf{v}_{k+1}, \ \ldots, \ \mathbf{v}_m\}. \end{aligned}$$

Also, with the above residual modification of the vectors we ensure that the new vectors $\mathbf{v}_l$, for $l > k$, vanish at all preceding coarse dofs, $\cup \mathcal{N}_i^c$, $i = 1, \ldots, \ k$. Recall that we identify the coarse dofs $\mathcal{N}_k^c$ with certain fine dofs, namely, defined by the the row indices of P_k where the identity block of P_k is placed. The fact that the residual vector $\mathbf{v}_{k+1}$ vanishes at $\cup \mathcal{N}_i^c$, $i = 1, 2, \ldots, \ k$, allows us to seek P_{k+1} having zero rows corresponding to the set $\cup \mathcal{N}_i^c$, $i = 1, 2, \ldots, \ k$. Thus, we reduce the problem of constructing a mapping P that interpolates exactly $m \geq 1$ vectors $\mathbf{v}_k$, to m consecutive tasks of constructing a mapping P_k that interpolates only one vector exactly, namely the residual vector $\mathbf{v}_k := \mathbf{v}_k^{new}$. The actual algorithm that one can utilize in practice depends on the goal one wants to achieve. In AMG, one needs to construct a P that has certain minimal norm. For example, in [WCS00], and more recently in [XZ04], one constructs a P (in our case, P_k) that minimizes the trace of the coarse matrix $P_k^T A P_k$. This is not the only possible choice (see e.g., [Ch03]). Finally, note that for $k > 1$, P_k is sought vanishing at a starting set of coarse dofs $\mathcal{N}_1^c$, therefore one may expect that $P_k^T A P_k$ be relatively well–conditioned. Thus, the norm in which one seeks P_k (for $k > 1$) with optimal properties should not be that important (at least theoretically).

We have not specified the choice of the sparsity pattern of the columns P_k of P, i.e., the sparsity of the W-blocks in (1.3). This is done in the following section for one specific AMG method, namely, the element agglomeration AMGe ([JV01]).

3. Selection of sparsity pattern of P in element agglomeration AMGe

In summary, the element agglomeration AMGe ([JV01]) assumes that in addition to the fine grid matrix A one has access to the fine–grid element matrices $\{A_e\}_{e\in\mathcal{T}}$ and the set of elements $e \subset \mathcal{T}$, the latter meaning that one knows relations describing the topology of the fine–grid elements. For example, one needs the incidence matrices element–dof, element–face, face–dof, etc. Based on an agglomeration algorithm (see [JV01]), one constructs coarse elements E, which are union on (connected) fine elements, and as described in detail in [V02] one can construct the topology of the coarse elements. In the "element agglomeration AMGe" method one imposes the restriction that the interpolation mapping P has the property that fine dofs that belong to an agglomerated element E are interpolated only from coarse dofs in E. This in particular implies that shared dofs by two or more agglomerated elements should be interpolated only from coarse dofs that are shared by the same agglomerated elements (that share that fine dof). In some instances this implies that such shared dofs must become coarse dofs. This is the case for all vertex dofs. That is, the set of vertices provides a natural minimal coarse grid. This was the choice made in the original paper [JV01]. The sparsity pattern of P is then clear. For every vertex dof "i", there is a column $\boldsymbol{\psi}_i$ in P and the non–zero entries of $\boldsymbol{\psi}_i$ correspond to a subset of all dofs in the union of agglomerated elements E that share that vertex. A natural subset would be the dofs in the union Ω_i of the agglomerated elements that share vertex "i" which are not on the boundary of Ω_i. The "boundary" set is defined as dofs in Ω_i that belong to other agglomerated elements (not in Ω_i). Also, one excludes from Ω_i all other coarse dofs (different from i). The resulting local set of fine dofs defines the support of the "i"th column of P. By this choice one guarantees that dofs belonging to an agglomerate E are interpolated only from coarse dofs in that agglomerated element. This is the main requirement that allows one to define a coarse element matrix in a natural way; namely, let A_E be the assembled local matrix corresponding to the agglomerated element E. Then, the coarse element matrix, corresponding to the coarse element E, reads $A_E^c = P_E^T A_E P_E$. Here P_E is the restriction of P to the set E, i.e., the rows of P_E correspond to the rows of P restricted to E and the columns of P_E correspond to the columns of P restricted to the coarse dofs from E.

Recall that in the setting of the preceding section, we have $P = [P_1,\ P_2,\ \ldots, P_m]$. Then the sets $\mathcal{N}_k^c$ are spread over the agglomerated elements and the respective column "i" of P_k will have support subjected to the agglomerated elements that share coarse dof "i" , the boundary dofs excluded, as well as all coarse dofs (different from "i") from the current and preceding sets $\mathcal{N}_j^c$, i.e., for $j \le k$, excluded.

4. Algorithms and numerical illustration

Algorithms for augmenting current coarse grid. It may be a typical situation in practice that an initial coarse grid $\mathcal{N}_c := \mathcal{N}_1^c$ is used to construct a $P := P_1$ that fits the first vector $\mathbf{v}_1$. Then after one has computed the residual vectors $\mathbf{v}_k := \mathbf{v}_k - P_1\mathbf{v}_k^c$, $k = 2,\ \ldots,\ m$ with $\mathbf{v}_k^c = \mathbf{v}_k|_{\mathcal{N}_1^c}$, one selects the additional coarse set $\mathcal{N}_2^c \subset \mathcal{N}\backslash\mathcal{N}_c$, sets $\mathcal{N}_c := \mathcal{N}_c \cup \mathcal{N}_2^c$, and proceeds further. To account for (near) linear local dependence,

one may want to incorporate a dropping strategy, that is, if some entries of the (residual) vectors are very small one drops them, by zeroing them. Then, in the case of element agglomeration AMGe, one may adopt the following strategy. Based on a "minimal intersection set" algorithm (see below), one can partition all AEs (agglomerated elements) into non–overlapping sets (groups) of dofs in $\mathcal{N} \setminus \mathcal{N}_c$. Here, at step k, $\mathcal{N}_c := \mathcal{N}_1^c \cup \cdots \cup \mathcal{N}_{k-1}^c$. These minimal intersection sets, in the case of true geometrical finite elements, are the vertices, edge interior, face interior and AE–interior dofs. Then, if $\mathbf{v}_k$ is non–zero at a given minimal intersection set, one selects a new coarse dof from that set, for example by looking at the local maximal absolute value of $\mathbf{v}_k$ restricted to that set. If $\mathbf{v}_k$ is zero (after potential dropping of small entries), one does not have to select any additional coarse dofs from that particular intersection set. All newly selected coarse dofs form $\mathcal{N}_k^c$.

The algorithm for forming the *minimal intersection sets* uses the relation "AE_dof" represented by the incidence matrix $\mathcal{E}$, defined as

$$\mathcal{E}_{ij} = \begin{cases} 1, & \text{if dof } j \text{ is in the agglomerated element } i, \\ 0, & \text{otherwise.} \end{cases}$$

Note that $\mathcal{E} \in \mathbb{R}^{n_E \times n}$, where n_E is the number of agglomerated elements and n is the number of degrees of freedom. Consider now $\mathcal{Z} = \mathcal{E}^T\mathcal{E} \in \mathbb{R}^{n\times n}$ and observe that

$$\mathcal{Z}_{ij} = \#\{E \mid i \in E \text{ and } j \in E\},$$

where # stands for cardinality. In other words, the right hand side of the above equation equals to the number of agglomerated elements E, such that both dofs i and $j \in E$. We then split the set of degrees of freedom $\{1, \ldots, n\}$ in non-overlapping sets $\{\mathcal{I}_k\}_{k=1}^{\ell}$ (called *minimal intersection sets*) with the property that i and j are in one and the same set $\mathcal{I}_k$ if and only if $\mathcal{Z}_{ij} = \mathcal{Z}_{ii} = \mathcal{Z}_{jj}$.

A more formal algorithm to determine these sets is given below. As input the algorithm takes $\mathcal{Z}$ (or equivalently $\mathcal{E}$), and as output, we have the number of minimal intersection sets ℓ and also a vector $\chi = (\chi(i))_{i=1}^n$ such that $\chi(i) = k$ if and only if $i \in \mathcal{I}_k$.

Algorithm 4.1 (Minimal intersection sets)**.**

Set $\ell = 0$; $\chi(i) = 0$, $i = 1, \ldots, n$.
For $i = 1$ ***to*** n ***do***
  If $\chi(i) = 0$ ***then***
    $\ell = \ell + 1$
    For j *such that* $\chi(j) = 0$ ***and*** $\mathcal{Z}_{ij} = \mathcal{Z}_{ii} = \mathcal{Z}_{jj}$ ***do***
      Set $\chi(j) = \ell$
    end For
  end If
end For

Clearly, one can easily derive a version of this algorithm without explicitly forming $\mathcal{Z}$. In both implementations, the number of operations would be proportional to the number of non-zeros in $\mathcal{E}$.

Next, we present an algorithm on how to select additional coarse grid based on a given vector $\mathbf{v} := \mathbf{v}_k$.

TABLE 1. Two–grid (TG) results with linear vector preserving interpolation; coarsening factor = 128.

# elements	2048	8192	32768	131072
# dofs	1089	4225	16641	66049
# PCG iter	8	9	10	10
# TG iter	20	18	19	19
ϱ_{TG}	0.40	0.357	0.39	0.388

TABLE 2. Multilevel AMG results with linear vector preserving interpolation; first level coarsening factor = 128.

# elements	2048	8192	32768	131072
# dofs	1089	4225	16641	66049
# PCG iter	8	9	9	9
# AMG iter	20	18	20	23
ϱ_{AMG}	0.40	0.359	0.386	0.41
operator complexity	1.39	1.62	1.79	1.89
# levels	4	6	8	10

Algorithm 4.2 (Selecting additional coarse grid dofs in AMGe)**.**
Let $\mathcal{N} := \mathcal{N} \setminus \mathcal{N}_c$ be a grid complementary to an initial coarse grid $\mathcal{N}_c$. Let $\{E\}$ be a overlapping partition of the dofs in $\mathcal{N}$ into sets E which we refer to as AEs (or agglomerated elements).

Let $\mathbf{v}$ be a vector with a non–zero maximum norm; i.e., $\|\mathbf{v}\|_{\max} = \max_i \ |\mathbf{v}(i)| \neq 0$. We will further denote by $\mathbf{v}(i)$ the ith entry of $\mathbf{v}$. Consider a given drop tolerance $\delta \in [0, 1)$. One constructs an additional coarse grid $\mathcal{N}_{\mathbf{v}}^c$, depending on the vector $\mathbf{v}$, as a subset of the current $\mathcal{N}$ in the following steps.

- *drop small entries, i.e., if $|\mathbf{v}(i)| \leq \delta \ \|\mathbf{v}\|_{\max}$ set $\mathbf{v}(i) = 0$. Also, set $\mathcal{N}_{\mathbf{v}}^c = \emptyset$.*
- *construct the minimal intersection sets $\mathcal{I}$ based on two given relations "AE_dof" and "dof_AE" represented by the incidence matrices $\mathcal{E}$ and $\mathcal{E}^T$, or as described in Algorithm 4.1, based only on the product matrix $\mathcal{E}^T\mathcal{E}$.*
- *compute $\eta(\mathcal{I}) = \max \{|\mathbf{v}(i)|, \ i \in \mathcal{I}\}$ for every intersection set $\mathcal{I}$ and let $i_{\max}(\mathcal{I}) \in \mathcal{I}$ be such that $|\mathbf{v}(i_{\max})| = \eta(\mathcal{I})$.*
- *if $\eta(\mathcal{I}) > 0$, augment $\mathcal{N}_{\mathbf{v}}^c$ with $i_{\max}(\mathcal{I})$ for all intersection sets $\mathcal{I}$.*

We would like to point out that more efficient (w.r.t. storage) implementation of Algorithm 4.2 is feasible by combining it with the minimal intersection set algorithm. In such a case one does not need to explicitly form the minimal intersection sets, since $\eta(\mathcal{I})$, and $i_{\max}(\mathcal{I})$ can be found during the execution of the inner *"For"* loop in Algorithm 4.1.

Numerical tests. The tests problems were the Poisson equation discretized on a sequence of triangular meshes on the unit square domain with homogeneous Dirichlet boundary conditions using standard Lagrangian piecewise linear elements. We used

TABLE 3. Multilevel AMG results with constant vector preserving interpolation; first level coarsening factor = 128.

# elements	2048	8192	32768	131072
# dofs	1089	4225	16641	66049
# PCG iter	10	16	18	18
# AMG iter	57	74	80	81
ϱ_{AMG}	0.69	0.75	0.76	0.76
# levels	4	6	8	10

TABLE 4. Multilevel and two–level AMG results with linear vector preserving interpolation for anisotropic diffusion problem; first level coarsening factor = 100.

# elements	400	1600	6400	25600
# dofs	231	861	3321	13041
# PCG iter	7	15	23	38
# AMG iter	18	50	99*	99*
ϱ_{AMG}	0.374	0.71	0.84	0.94
operator complexity	1.15	1.51	1.86	1.96
# levels	2	4	6	8
two level # PCG iter	7	15	24	40
# TG iter	18	52	99*	99*
ϱ_{TG}	0.374	0.71	0.84	0.94

element agglomeration with large coarsening factor, that is, to get number of agglomerated elements equal to the number of fine-grid elements divided by the coarsening factor. In our tests the element coarsening factor was either 128 or as close as possible to 128. The initial coarse grid $\mathcal{N}_c = \mathcal{N}_1^c$ was simply the set of vertices of the resulting agglomerated elements (i.e., as in the original element agglomeration AMGe method from [JV01]). The sparsity pattern of the interpolation matrices was chosen as described in Section 3, i.e., based on the AEs and the current fine grid.

The vectors that we tried to fit were 1, x, and y, i.e. $\mathbf{v}_1 = \mathbf{1} = (1)_{i\in\mathcal{N}}$, $\mathbf{v}_2 = (x_i)_{i\in\mathcal{N}}$, and $\mathbf{v}_3 = (y_i)_{i\in\mathcal{N}}$. Here, for a dof i, $(x_i,\ y_i)$ stands for its geometrical coordinates.

In the construction of all P_k, $k = 1, 2, 3$ we used the construction of P_k with the prescribed sparsity pattern so that the resulting coarse matrix has minimal trace. The algorithm implemented was based on the additive Schwarz method as described in [XZ04]. However, the inversion of the actual Schwarz blocks was done only approximately, by few Gauss–Seidel iterations. The 1st vector (the constant one) led to a matrix P_1 which was computed by a little more accurate inversion of the Schwarz blocks, whereas for the remaining vectors the Schwarz blocks were approximated only by one symmetric Gauss–Seidel iteration. To be more specific, let Ω_i be the support set of the ith column of a given P_k. Let I_i be the extension by zero of a vector defined

on Ω_i. Then, the ith Schwarz block is defined as $T_i = I_i \left(I_i^T A I_i\right)^{-1} I_i^T$. The Gauss–Seidel iterations are used to evaluate (approximately) the actions of $\left(I_i^T A I_i\right)^{-1}$. Then, if one wants to build a $P := P_k$ such that $P\mathbf{v}^c = \mathbf{v}$, where $\mathbf{v}^c = \mathbf{v}|_{\mathcal{N}_c}$ one proceeds as follows. Let $T = \sum_{i \in \mathcal{N}_c} g_i^2 T_i$. Here, $g_i = \mathbf{v}_k(i)$. Then, $\boldsymbol{\psi}_i = g_i T_i T^{-1} \mathbf{v}$ defines the ith column of P. The vector $\mathbf{x} = T^{-1}\mathbf{v}$ is computed by the preconditioned CG method applied to $T\mathbf{x} = \mathbf{v}$ with a simple diagonal matrix $D = \sum_{i \in \mathcal{N}_c} g_i^2 I_i D_i^{-1} I_i^T$, where $D_i = \operatorname{diag} I_i^T A I_i$, as preconditioner. Obviously, $P\mathbf{v}_c = \sum_{i \in \mathcal{N}_c} g_i \boldsymbol{\psi}_i = \mathbf{v}$. The fact that the resulting P leads to $P^T A P$ with minimal trace is proved in [XZ04]. The motivation to choose the ith coarse dof as in Algorithm 4.2 is that then T gets better conditioned. Note that too small a value of $g_i = \mathbf{v}(i)$ will make T almost singular, or very ill–conditioned.

We show, for illustration, a complete set of coarse basis "functions" (columns of the interpolation mapping P) in Figs. 3, 4, and 5, corresponding to one agglomerated element from the mesh shown in Fig. 2. In Fig. 6, we show that these basis functions span locally, the constant one, and the coordinate vectors (x_i) and (y_i).

In the presented tables, one finds two–level and multilevel convergence results for a $V(1,1)$–cycle AMG, used both as a preconditioner in the CG method, as well as in a stationary iterative process. The stopping tolerance was 10^{-9} relative reduction of the initial residual norm. The smoother was the overlapping Schwarz method. The Schwarz domains were the elements (at the given level) viewed as sets of dofs. At the initial mesh, these corresponded to a three–by–three matrices. In the case of stationary iterations we also show the asymptotic convergence factor ϱ. One can compare the methods, see Table 2 and Table 3, using the same agglomerated elements, but different interpolation matrices P. Namely, Table 2 corresponds to P that preserved the linear vectors, whereas in Table 3 only the constant vector was exactly interpolated.

Also for comparison, in Table 1 the two–level results are shown to demonstrate how close the multilevel results (from Table 2) match the two–level ones.

Finally, in order to demonstrate that just a simple aggressive element agglomeration may not always work, we show in Table 4, results of the same nature as above, now for a difficult anisotropic diffusion problem. The anisotropy is generally not grid aligned. One of the meshes, after one level of agglomeration, is shown in Fig. 1. It is well–known that such problems would require special coarsening, based for example on a "compatible relaxation" algorithm, which in short, selects coarse dofs, tight to the convergence of the smoother restricted to a grid complementary to the coarse one (a strong theoretical justification for this argument is given in [FVZ04]). The notion of "compatible relaxation" is introduced in [B00]. For some more details in this direction we refer to [L04] and [FV04]. Alternative efficient coarsening can be achieved, in the element agglomeration AMGe framework, based on a variant of the spectral AMGe method from [Ch03]. The topic of compatible relaxation, however is not the focus of the present paper.

Conclusions. In conclusion, utilizing several vectors in the construction of AMG interpolation mappings can be useful if relatively aggressive coarsening is exploited

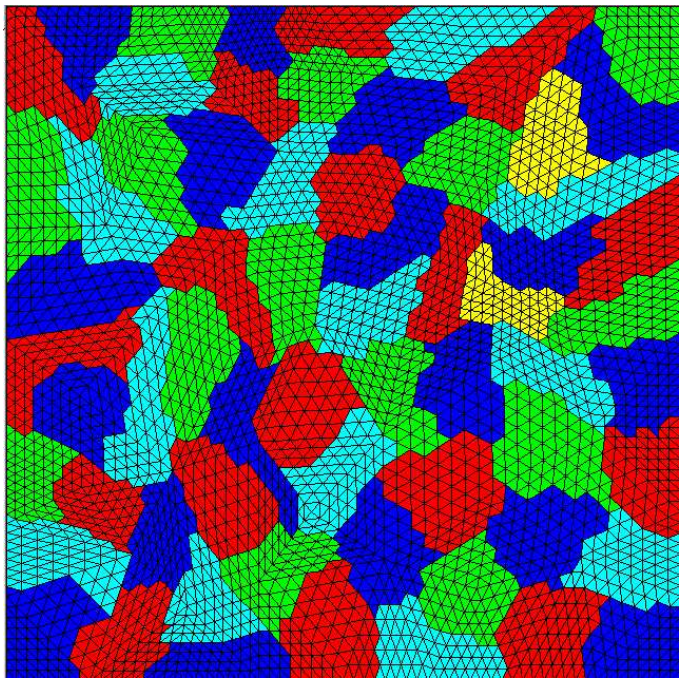

FIGURE 1.
Agglomerated elements from unstructured fine mesh; coarsening factor = 100. *Initial mesh has 6400 elements; # agglomerated elements* = 64.

in the algorithm. Potential applications of such interpolation mappings (or resulting coarse spaces) can be expected in non–linear AMG methods or in homogenization methods, where one needs more accurate (from approximation point of view) coarse spaces. These topics are not considered in the present paper.

REFERENCES

[B00] A. BRANDT, "*General highly accurate algebraic coarsening*", Electronic Transactions on Numerical Analysis **10**(2000), pp. 1–20.

[Ch03] T. CHARTIER, R. FALGOUT, V. E. HENSON, J. JONES, T. MANTEUFFEL, S. MCCORMICK, J. RUGE, AND P. S. VASSILEVSKI, "*Spectral AMGe (ρAMGe)*", SIAM Journal on Scientific Computing, **25**(2003), pp. 1–26.

[FV04] R. D. FALGOUT AND P. S. VASSILEVSKI, "*On Generalizing the AMG Framework*", SIAM Journal on Numerical Analysis, to appear.

[FVZ04] R. D. FALGOUT, P. S. VASSILEVSKI AND L. T. ZIKATANOV, "*On the two–grid convergence estimates*", Numerical Linear Algebra with Applications (submitted).

[JV01] J. E. JONES AND P. S. VASSILEVSKI, "*AMGe based on element agglomerations*", SIAM Journal on Scientific Computing **23**(2001), pp. 109-133.

[L04] O. LIVNE, "*Coarsening by compatible relaxation*", Numerical Linear Algebra with Applications, 11(2004), pp. 205–227.

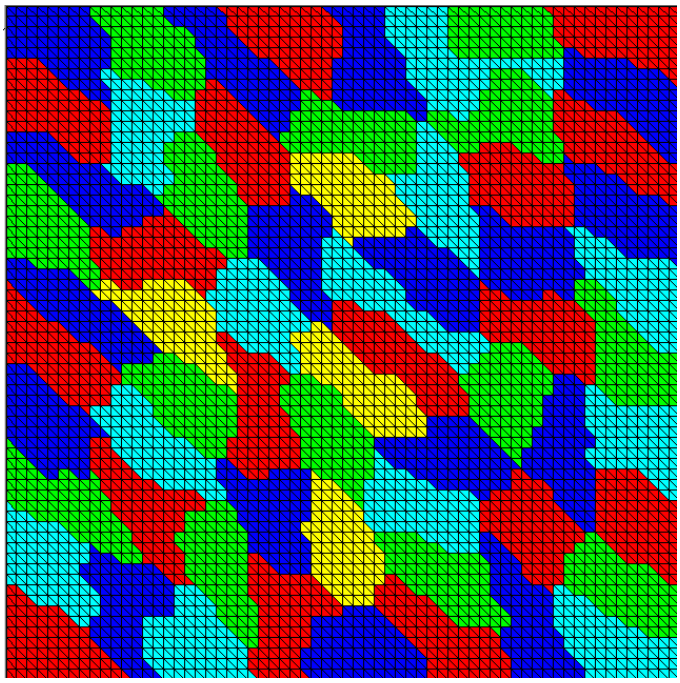

FIGURE 2.
Agglomerated elements; coarsening factor = 128. *Initial mesh has 8192 elements; # agglomerated elements* = 64.

[V02] P. S. VASSILEVSKI, "*Sparse matrix element topology with application to AMG and preconditioning*", Numerical Linear Algebra with Applications, **9**(2002), pp. 429–444.

[WCS00] W. L. WAN, T. F. CHAN AND B. SMITH, *An energy-minimizing interpolation for robust multigrid methods*. SIAM Journal on Scientific Computing, **21**(4), (2000), pp. 1632–1649.

[XZ04] J. XU AND L. T. ZIKATANOV, *On an energy minimizing basis in algebraic multigrid methods*. Computing and Visualization in Science, (2004) (in press).

CENTER FOR APPLIED SCIENTIFIC COMPUTING, LAWRENCE LIVERMORE NATIONAL LABORATORY, P.O. BOX 808, L-560, LIVERMORE, CA 94551, U.S.A.
E-mail address: panayot@llnl.gov

CENTER FOR COMPUTATIONAL MATHEMATICS AND APPLICATIONS, DEPARTMENT OF MATHEMATICS, THE PENNSYLVANIA STATE UNIVERSITY, UNIVERSITY PARK, PA 16802
E-mail address: ludmil@psu.edu

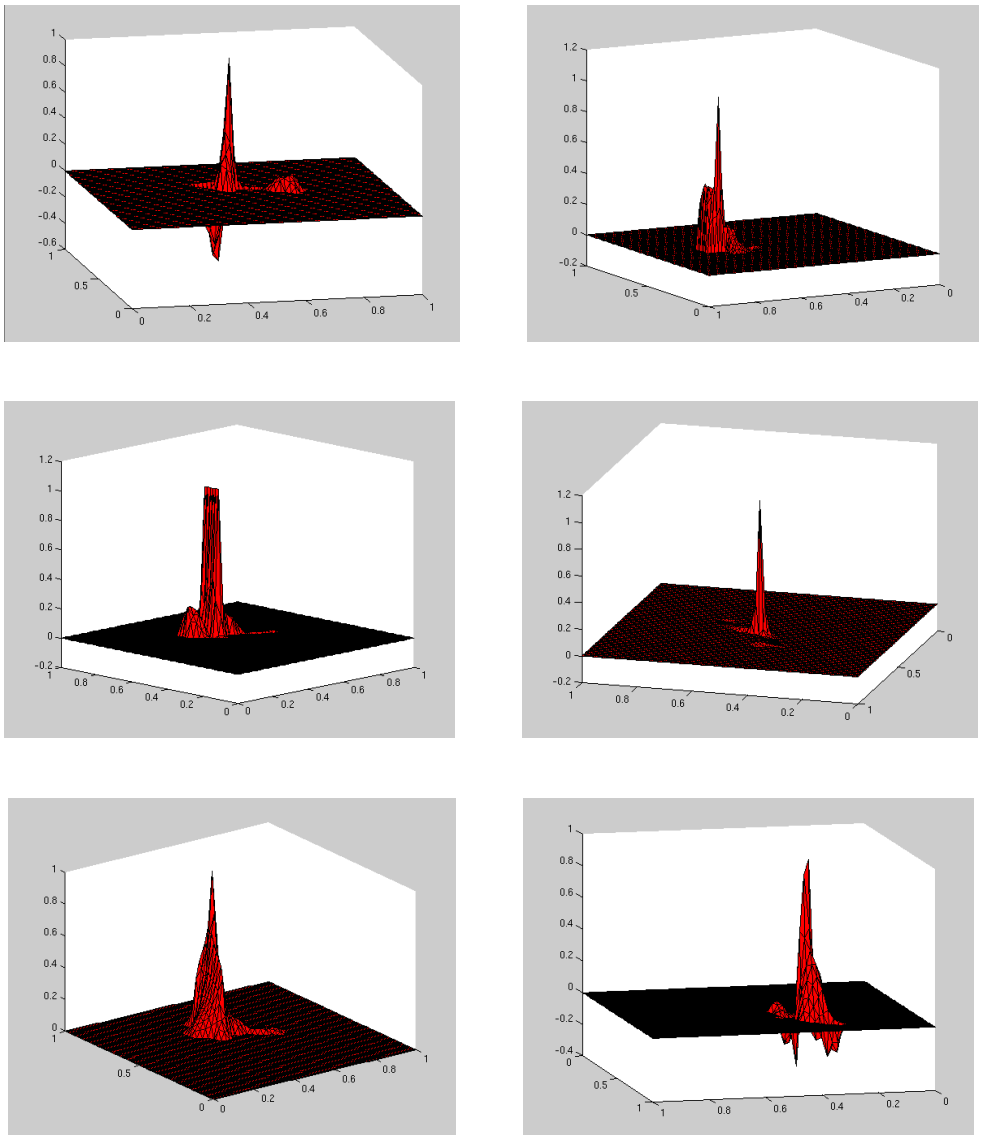

FIGURE 3.
Basis functions associated with one agglomerated element; first 6 ones.

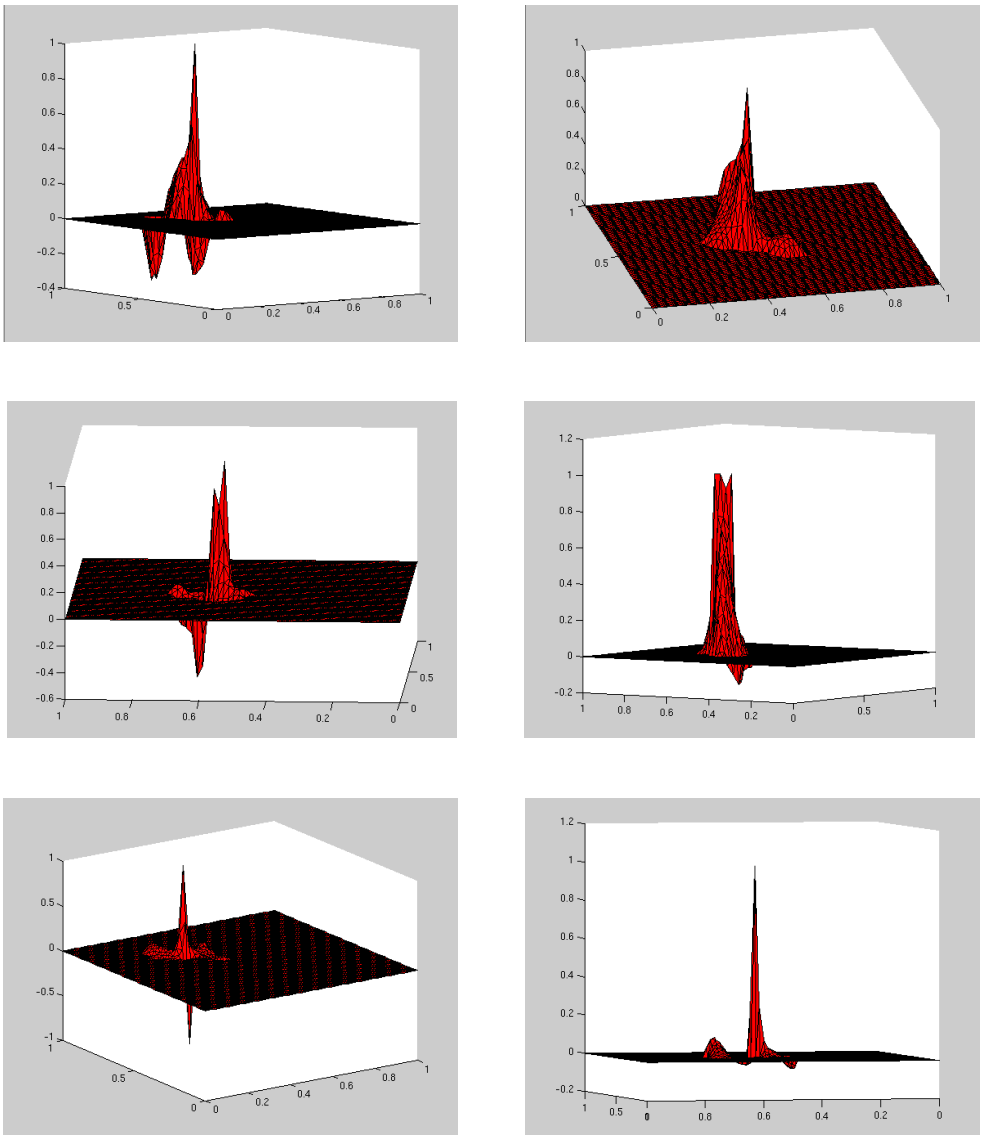

FIGURE 4.
Basis functions associated with one agglomerated element; sixth to 12th ones.

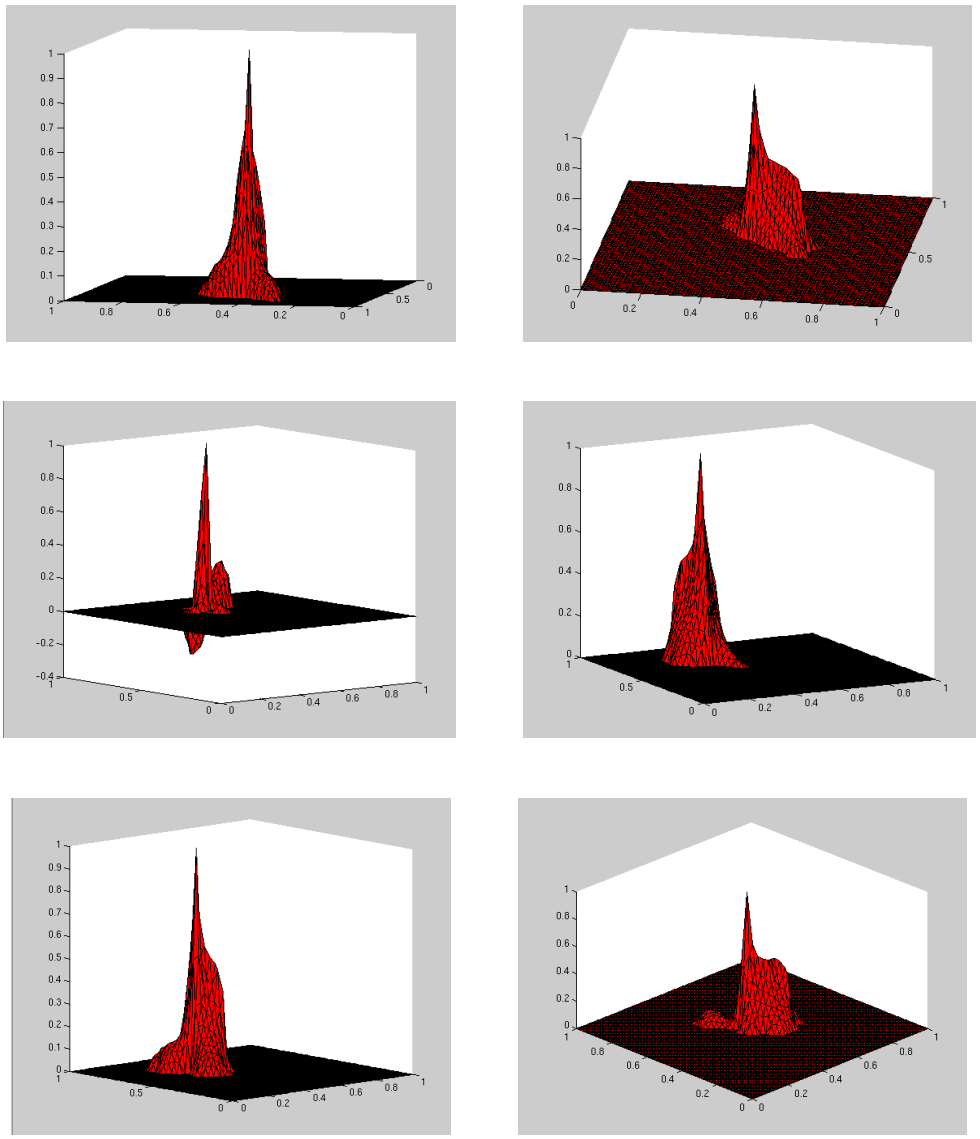

FIGURE 5.
Basis functions associated with one agglomerated element; 13th to 18th ones.

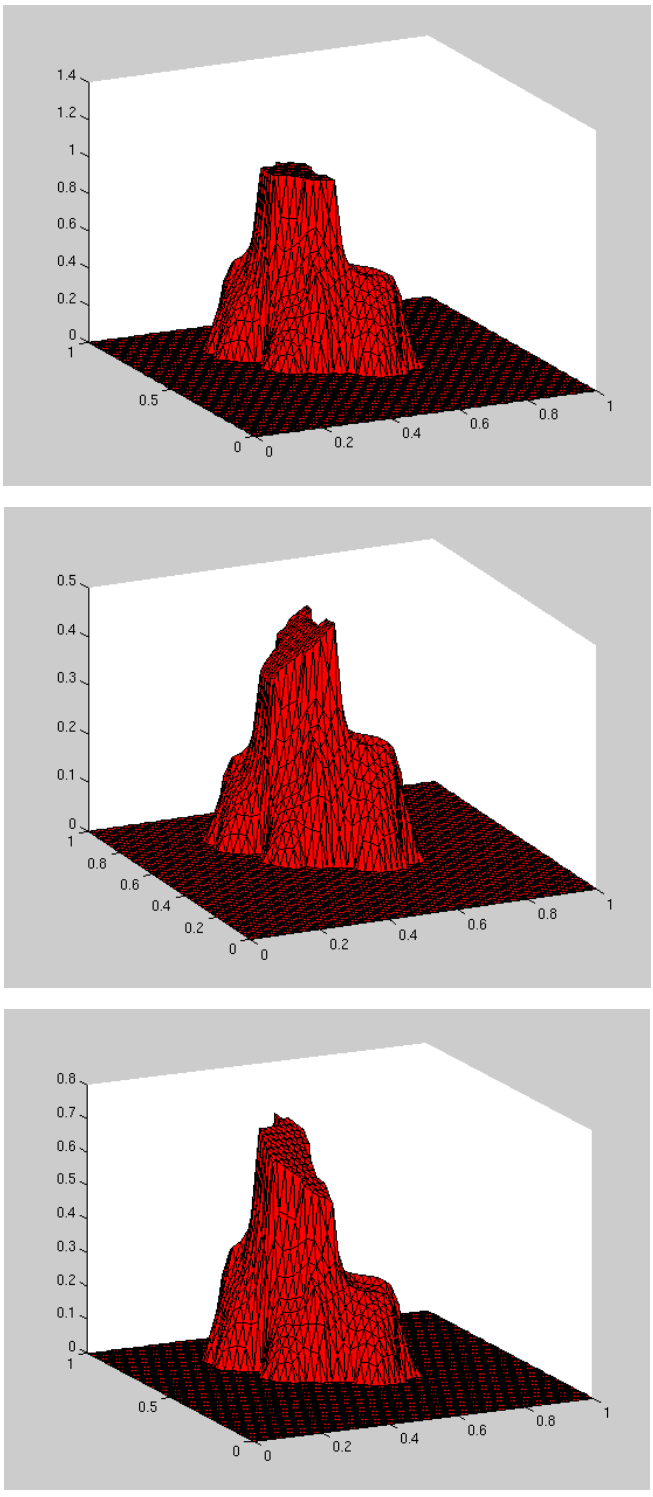

FIGURE 6.
Local representation of 1, x, *and* y *functions associated with one agglomerated element.*