An agglomeration multigrid method for unstructured grids*

Tony F. Chan † Jinchao Xu‡ Ludmil Zikatanov§

March 4, 1998

Abstract

A new agglomeration multigrid method is proposed in this paper for general unstructured grids. By a proper local agglomeration of finite elements, a *nested* sequence of finite dimensional subspaces are obtained by taking appropriate linear combinations of the basis functions from previous level of space. Our algorithm seems to be able to solve, for example, the Poisson equation discretized on any shape-regular finite element grids with nearly optimal complexity.

1 Introduction

In this paper, we discuss a multilevel method applied to problems on general unstructured grids. We will describe an approach for designing a multilevel method for the solution of large systems of linear algebraic equations, arising from finite element discretizations on unstructured grids. Our interest will be focused on the performance of an agglomeration multigrid method for unstructured grids.

One approach of constructing coarse spaces is based on generating node-nested coarse grids, which are created by selecting subsets of a vertex set, retriangulating the subset, and using piecewise linear interpolation between the grids (see [8, 5]). This still provides an automatic way of generating coarse grids and faster implementations of the interpolation in $O(n)$ time. The drawback is that in three dimensions, retetrahedralization can be problematic.

*To appear in the proceedings of *10-th International conference on Domain Decomposition methods*

†Department of Mathematics, UCLA, email: chan@math.ucla.edu. This author was supported in part by NSF ASC-9720257.

‡Center for Computational Mathematics and Applications, Department of Mathematics, Penn State University. email:xu@math.psu.edu, This author was supported in part by NSF DMS94-03915-1 and NSF DMS-9706949 through Penn State, by NSF ASC-92-01266 through UCLA and NSF ASC-9720257.

§Center for Computational Mathematics and Applications, Department of Mathematics, Penn State University, email: ltz@math.psu.edu. This author was supported in part by NSF DMS-9706949 through Penn State, NSF ASC-9720257 and also by Center for Computational Mathematics and Applications of Penn State University.

Another effective coarsening strategy has been proposed by Bank and Xu [1]. It uses the geometrical coordinates of the fine grid and the derefinement algorithm is based on the specified patterns of fine grid elements. The interpolation between grids is done by interpolating each fine grid node using only 2 coarse grid nodes. As a consequence of that the fill-in in the coarse grid matrices is reasonably small. The hierarchy of spaces is defined by interpolating the basis.

Recently a new approach, known as auxiliary space method, was proposed by Xu [16]. In this method only one non-nested (auxiliary) grid is created and then all consecutive grids are nested. This can be done by using as auxiliary grid a uniform one and interpolating the values from the original grid there. For a uniform grid then, there is a natural hierarchy of coarse grids and spaces. Such a procedure leads to optimal multigrid methods in some applications.

One promising new coarsening techniques is based on the agglomeration technique (see Koobus, Lallemand and Dervieux [11]). Instead of constructing a proper coarse grid, neighboring fine grid elements are agglomerated together to form macroelements. Since these agglomerated regions are not standard finite elements, appropriate basis functions and interpolation operators must be constructed on them. An algebraic construction of agglomerated coarse grid spaces has been investigated by Vaněk, Mandel, and Brezina [6] and Vaněk, Křížková [7]. Their approach uses a simple initial interpolation matrix, which might not be stable, and then this matrix is smoothed and stabilized by some basic relaxation schemes, e.g. Jacobi method.

The pure algebraic definition of the coarse spaces has the advantage that there is no need to use any geometrical information of the grid or of the shape of the grid and the kind of finite elements used. We refer to a paper by Ruge and Stuben [13] on algebraic multigrid. Recent developments in this direction have been made by Braess [2] and Reusken [12]. The main issue in using pure "black-box" algebraic derefinement is that the coarse grid operators usually become denser and it is not clear how to control their sparsity except in some special cases.

Another approach in the definition of coarse spaces, known as composite finite element method was recently investigated by Hackbusch and Sauter in [10]. This method gives coarse space constructions which result in only few degrees of freedom on the coarse grid, and yet can be applied to problems with complicated geometries.

In this paper, we will consider a new and rather simple technique for defining *nested* coarse spaces and the corresponding interpolation operators. Our earlier experience shows that the definition of the sparsity pattern of the transfer operators and the definition of these operators themselves is the most crucial point in designing multigrid algorithms for elliptic problems. In the present paper we propose a technique based on the graph-theoretical approach. Our goal is to construct a "coarse grid" using only the combinatorial (not the geometrical properties) of the graph of the underlying fine grid. This coarse grid is formed by groups of elements called agglomerated macroelements. Using this approach a macroelement grid can be constructed for any unstructured finite element triangulation. We can implement our algorithm with or without any use of the nodal coordinates. Based on this macroelement partition, we propose an interpolation technique which uses only arithmetic average based on clearly determined coarse grid nodes. This leads to savings in storage and CPU time, when such scheme is implemented. In fact, to store the interpolation matrix we only need to store integers. Although rather simple, such type of interpolation leads to a multigrid algorithm with nearly optimal performance. Moreover the algorithm naturally recovers the structure of the natural coarse grids if the fine grid is obtained by structured refinement. Although we present only 2D algorithms we believe that it can be extended for 3D problems as well.

The rest of the paper is organized as follows. In section 2 we state the differential problem and briefly comment on the finite element discretization. In section 3 we give the definition of the standard V-cycle preconditioner. In section 4 we describe in detail the two level coarsening algorithm. In section 4.3 the interpolation between grids is defined. The multilevel implementation of the algorithm is given in Section 4.4. The stability and approximation properties are investigated in Section 5 under rather mild assumptions on the geometry of the coarse grids. In Section 6 results of several numerical experiments are presented.

2 A model problem and discretization

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain with boundary $\Gamma = \Gamma_D \cup \Gamma_N$, where Γ_D is a closed subset of Γ with positive measure. We consider the following variational formulation of elliptic PDE: Find $U \in H^1_D(\Omega)$ such that

$$a(U, v) = F(v) \text{ for all } v \in H^1_D(\Omega), \tag{1}$$

where

$$a(U, v) = \int_\Omega \alpha(x) \nabla U \cdot \nabla v dx, \quad F(v) = \int_\Omega F(x) v dx. \tag{2}$$

Here $H^1_D(\Omega)$ as usual denotes the Sobolev space which contains functions which vanish on Γ_D with square integrable first derivatives. It is well-known that (1) is uniquely solvable if $\alpha(x)$ is a strictly positive scalar function and F is square integrable.

We consider a finite element space of continuous piecewise linear functions $M_h \subset H^1_D(\Omega)$ defined on a triangulation T_h of Ω. Then the corresponding finite element discretization of (2) is: Find $u_h \in M_h$ such that

$$a(u_h, v_h) = F(v_h) \text{ for all } v_h \in M_h. \tag{3}$$

The discretization results in a linear system of equations:

$$Au = f, \tag{4}$$

where A is a symmetric and positive definite matrix, f is the right hand side and the nodal values of the discrete solution u_h will be obtained in the vector u after solving the system (4).

3 Multigrid method

In this section, we introduce the notation related to the multigrid method, and we define the (1-1) V-cycle preconditioner.

Let us consider the following simple iteration scheme:

$$u^{\ell+1} = u^{\ell} + B_J(f - Au^{\ell}) \quad \ell = 1, 2, \ldots, \tag{5}$$

where B_J is the V-cycle preconditioner B_J to be defined. We assume that we have given a nested sequence of subspaces $M_0 \subset \ldots \subset M_{J-1} \subset M_J \equiv M_h$, with $\dim(M_k) = n_k$. We assume that the matrices A_k, $k = 0, \ldots J$, are stiffness matrices associated with M_k. We also assume that the interpolation operators I_{k-1}^k and the smoothing operators S_k are given.

In our case B_J will correspond to $(1-1)$ V-cycle preconditioner. For given $g \in M_k$ we define $B_k g$ as follows:

Algorithm 3.1 *[(1-1) V-cycle]*

0. *If* $k = 0$ *then* $B_0 g = A_0^{-1} g$
1. Pre-smoothing*:* $x^1 = S_k^T g$
2. Coarse grid correction*:*
  1. $q^0 = (I_{k-1}^k)^T (g - A_k x^1)$*;*
  2. $q^1 = B_{k-1} q^0$*;*
  3. $x^2 = x^1 + I_{k-1}^k q^1$*;*
3. Post-smoothing*:* $B_k g = x^2 + S_k(g - A_k x^2)$.

The practical definition of such a preconditioner in the case of unstructured grids will be our main goal in the next sections. We will define proper interpolation (prolongation) operators I_{k-1}^k, for $k = 1, \ldots, J$ and the subspace M_{k-1} by interpolating the nodal basis in M_k. In order to have convergence of the iteration (5) independent of the mesh parameters, the subspaces have to satisfy certain stability and approximation properties, namely, that there exists a operator $\Pi_k : H^1(\Omega) \to M_k$ such that:

$$\|\Pi_k v\|_{1,\Omega} \le C\|v\|_{1,\Omega}, \tag{6}$$

$$\|v - \Pi_k v\|_{0,\Omega} \le Ch|v|_{1,\Omega}, \quad \forall v \in H^1(\Omega). \tag{7}$$

We will comment on these properties of the agglomerated spaces in § 5. Once the subspaces are defined, the V-cycle algorithm can be implemented in a straightforward fashion using as coarse grid matrices, given by $A_{k-1} = (I_{k-1}^k)^T A_k I_{k-1}^k$.

General discussions concerning the convergence of this type of method and its implementation can be found in the standard references, e.g. Bramble [3], Hackbusch [9], Xu [15].

4 Agglomerated macroelements

The main approach we will take in the construction of I_{k-1}^k will be first to define a coarse grid formed by macroelements (groups of triangles) and then interpolate locally within each macroelement. In this section, we will present an algorithm for the definition of the coarse grid consisting of macroelements. We first identify the *set of coarse grid nodes.* The interpolation from coarse grid to the fine grid will use the values at these nodes. As a next step, for a given node on the fine grid we have to define its ancestors on the coarse grid (i.e. the coarse grid nodes which will be used in the interpolation). These ancestors are determined by partitioning the fine grid into *agglomerated macroelements* (such macroelements can be seen on Fig. 1) which in some sense are analogue of the finite elements, because they have vertices which are precisely the coarse grid nodes, and their edges are formed by edges of the underlying fine grid.

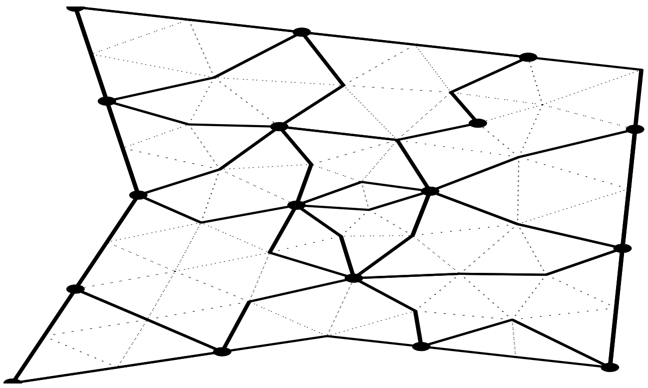

Figure 1: An example of macroelements

4.1 Some basic graph theory

In this subsection we introduce some basic notation and definitions. Given a finite element triangulation T_h of Ω, we consider the corresponding graph, denoted by $G = (V, E)$, where V is the set of vertices (grid nodes) and E is the set of edges (boundaries of the triangles). In this definition, the concept of vertex and edge happen to be the same for the underlying triangulation and for the graph corresponding to the stiffness matrix. Associated with the graph G, we will form our coarse grid on the so called maximal independent set (MIS, for short) which is a set of vertices having the following two properties: any two vertices in this set are *independent* in the sense that they are not connected by an edge, and the set is *maximal* in the sense that an addition of any vertex to the set will invalidate the aforementioned independent property. The *graph distance* between two vertices $v, w \in V$ is defined to be the length of the shortest path between these two vertices. A *matching* in G is any collection of edges such that no two edges in this collection share a vertex.

The construction of the macroelements will be based on the dual mesh (graph) of G defined as follows. Given a triangulation T_h and associated graph G, the dual graph $G' = (V', E')$ of G is:

- Each element $T \in T_h$ is a vertex in G'.
- Two vertices in G' are connected by an edge if and only if they share an edge in G, i.e. $(T_1, T_2) \in E'$ if and only if $T_1 \cap T_2 \in E$ (see Fig. 2).

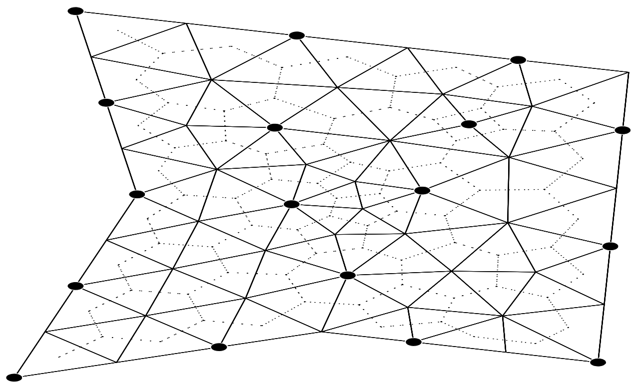

Figure 2: A triangulation and its dual mesh.

4.2 Two level coarsening algorithm

In this section we describe in detail the heuristic algorithm for forming a coarse grid macroelements from a given finite element triangulation.

As a first step we define the set of coarse nodes to be a MIS in G. An MIS is obtained by a simple "greedy" (locally optimal) algorithm given as follows.

Algorithm 4.1 (MIS)

1. *Pick an initial set of vertices V_0 (for example all boundary vertices).*
2. *Repeat:*
   (a) *Apply a "greedy" algorithm to find MIS in V_0.*
   (b) *Mark all nodes at distance 1 from V_0 (here distance is the graph distance).*
   (c) *Take as V_0 all vertices which are at distance 2 from V_0 and have not been explored (marked).*
3. *until $V_0 = \emptyset$.*
4. *Complete MIS by applying one step of "greedy" algorithm on V.*

To define the macroelements, we use the fact that separating two triangles on the fine grid and putting them in different groups is equivalent to removing an edge in the dual graph G'.

We now describe how to form the initial partition of G into groups of elements.

For any coarse grid node k (i.e. $k \in MIS$) we pick the edges in G having this node as an end. To this set of edges $E_k \subset E$ corresponds a set $E'_k \subset E'$, namely E'_k contains exactly all edges between all $T \in T_h$ which have this particular coarse node as a vertex. As a first step we remove E'_k from E'. Applying this procedure for all coarse grid nodes results in a subgraph of G', $G^* = (V^*, E^*)$ where $V^* = V'$ and $E^* = E' \setminus \cup_k E'_k$. The connected components in G^* will form the initial partition of Ω into groups of elements (see Fig. 3).

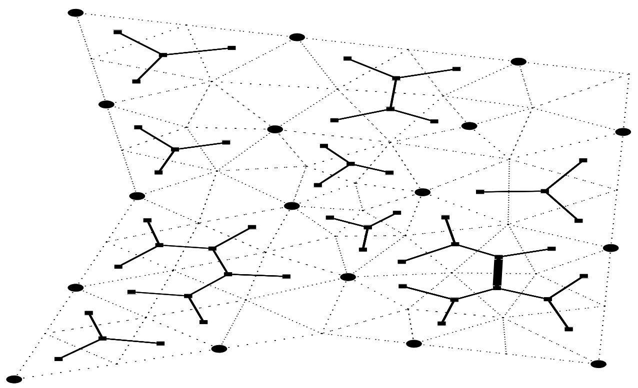

Figure 3: *Connected components in G^*.*

We note that there might be some isolated vertices in G^* and also some of the connected components might be considerably large. We first deal with the large groups (such a group can be seen on Fig. 3 in the right bottom corner of the domain) and we break them into smaller pieces. We consider a group of elements $M \subset T_h$ that corresponds to one connected component in G^* and denote the set of edges in M by E_M. We intend to break this group in pieces if there is an "interior" edge $e \in E_M$ such that $e \cap \partial M = \emptyset$. This breakup is done as follows (our considerations here are restricted only on $M \subset T_h$):

- From the subgraph formed by all edges $e \subset E_M$ such that $e \cap \partial M = \emptyset$, we form a matching. On the model grid (see Fig. 3) there is only one such edge in the whole domain.

- Remove the edges in the dual corresponding to the edges in the matching. In Fig. 3 this edge in the dual is drawn with thick line (near the right bottom corner of the domain). The pieces obtained by removing this edge are clearly seen on Fig. 1).

The situation with the isolated vertices in G^* is simpler. We propose two different ways of dealing with them as follows:

1. Since each isolated triangle (vertex in G^*) has as one vertex being a coarse grid point, the edge opposite to this vertex does not have a coarse grid node as an end, because our set of coarse grid nodes is a MIS. We group together two neighbors sharing this edge to form a macroelement. If such edge happens to be a boundary edge, we leave a single triangle to be a macroelement.

2. We group together all isolated neighbors. If such a group does not have more than 4 coarse grid vertices then we consider it as a new agglomerated macroelement. If it has more than 4 coarse grid vertices we proceed as in the previous step coupling triangles in this group two by two. In partitioning our model grid we have used precisely this way of grouping isolated triangles (see Fig 3, Fig. 1).

It is obvious that all triangles from the triangulation are either in a connected component in G^* or are isolated vertices in G^*. Thus we have explored all the triangles and every $T \in T_h$ is in some macroelement (see Fig. 1).

To summarize we give the following short description of the algorithm for agglomerating elements into macroelements:

Algorithm 4.2 (Coarse grid macroelements)

1. *Identify coarse grid nodes by finding an MIS.*

2. *For any coarse node, remove all dual edges surrounding it.*

3. *Find connected components in the remaining dual graph. These connected components form most of the agglomerated regions.*

4. *Breakup "large" macroelements into smaller pieces.*

5. *Group the remaining triangles into contiguous groups as additional macroelements.*

6. *The remaining connected components in the dual are called "agglomerated elements".*

Remark 4.3 *Note that this algorithm will give a unique partition in agglomerated macroelements up to the choice of MIS and the edges in the matchings (if we need further breakup of large connected components in G^*).*

We would like to elaborate a little more on the input data needed for the algorithm to work. The input we used was:

1. The grid (i.e. list of elements and correspondence "vertex–element"). From this correspondence we can easily define G in the usual way: two vertices are connected by an edge if and only if they share element.

2. The auxiliary graph G' whose vertices are the elements and the correspondence between edges in G and edges in G'.

Note that the algorithm we have described do not need the correspondence between edges in G and G' to be $(1-1)$ (as it is between the dual and primal graph). The only fact we used was: for a given edge in G the set of edges in G' which have to be removed is uniquely determined. This observation is important and will be used in the multilevel implementation of the algorithm.

4.3 The definition of coarse subspaces

In the present section we will describe a simple interpolation technique using the agglomerated macroelements. We also give a description how a multilevel variant of our derefinement algorithm can be implemented. With a grid agglomeration obtained as above, we need to define a coarse finite element space associated with the macroelements. This is equivalent to defining the interpolation between M_J and M_{J-1}. The interpolation is defined in the following way:

- Coarse nodes:
  - For the coarse nodes we simply define the interpolation to be the identity.
- Interior nodes:
  - For the nodes interior to the macroelements we use the arithmetic average of the values at coarse grid nodes defining the macroelement. This situation can be seen in Fig. 4.
- Edge nodes:
  - If the fine grid node lies on a macroelement edge, then its value is defined to be the average of the 2 coarse grid nodes defining the macro-edge (in Fig. 4 such a node is j_1).
  - If the fine grid node lies on more than one macro-edge, then its value is defined to be the simple arithmetic average of all the values corresponding to the different edges (in Fig. 4 such a node is j_2).

As an example we give the interpolated values at fine grid nodes for the grids in Fig. 4):

$$
\begin{aligned}
I_{J-1}^{J} v_h(x_{j_1}) &= \tfrac{v_h(x_{k_1})+v_h(x_{k_2})}{2}, \\
I_{J-1}^{J} v_h(x_{j_2}) &= \tfrac{v_h(x_{k_1})+v_h(x_{k_2})+v_h(x_{k_3})}{3}, \\
I_{J-1}^{J} v_h(x_{j}) &= \tfrac{v_h(x_{k_1})+v_h(x_{k_2})+v_h(x_{k_3})+v_h(x_{k_4})+v_h(x_{k_5})}{5}.
\end{aligned}
$$

This simple interpolation has the advantage that the matrix corresponding to it can be stored in the computer memory using only integers. The matrix vector multiplication is easier to perform and this basis preserves the constant function.

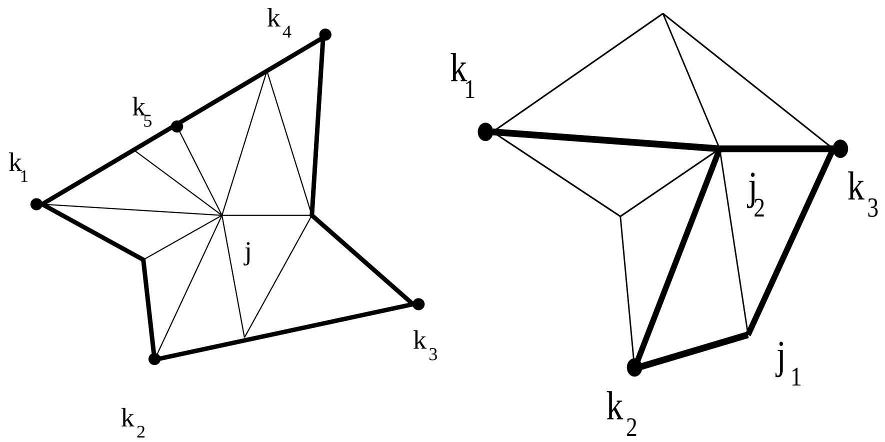

Figure 4: Example of interpolation. Thick lines mark the macroelement boundaries.

4.4 Multilevel implementation

A straightforward multilevel implementation of the coarsening algorithm, can be done by simply retriangulating the set of coarse grid points and apply the derefinement algorithm to the obtained triangulation. In this section we will propose another version, which has the advantage that it operates only on the graph and does not use nodal coordinates and real numbers arithmetic.

To apply the algorithm recursively, we need to define the same input data, but using the coarse grid. We first define the elements (triangles, or triples of vertices) on the coarse grid in the following way:

- Consider every macroelement as a polygon with m vertices $(k_1, k_2, \dots, k_m)$ in counter-clockwise ordering (m is the number of coarse grid vertices forming the macroelement). We triangulate it with $m-2$ triangles in the following way:
  1. If $m \leq 3$ stop.
  2. Form the triangles (k_1, k_2, k_3) and (k_1, k_3, k_m).
  3. Remove k_1 and k_2 from the polygon, set $k_1 \leftarrow k_m$ and $k_{i-1} \leftarrow k_i$ for $i = 3, \dots m-1$, $m \leftarrow m-2$. Go to 1.
- If a fine grid node lies on more than one macro-edge we form a m-gon with vertices the coarse grid points surrounding it (see Fig. 4, such a node is j_2). We triangulate this m-gon in the same way as we did in the previous step. Such a m-gon is shown in Fig. 5 on the right. This triangle corresponds to node j_2 in Fig. 4.

The coarse grid configurations corresponding to Fig. 4 are given in Fig. 5. Thus we define the graph $G_c = (V_c, E_c)$ corresponding to the coarse grid to be the following:

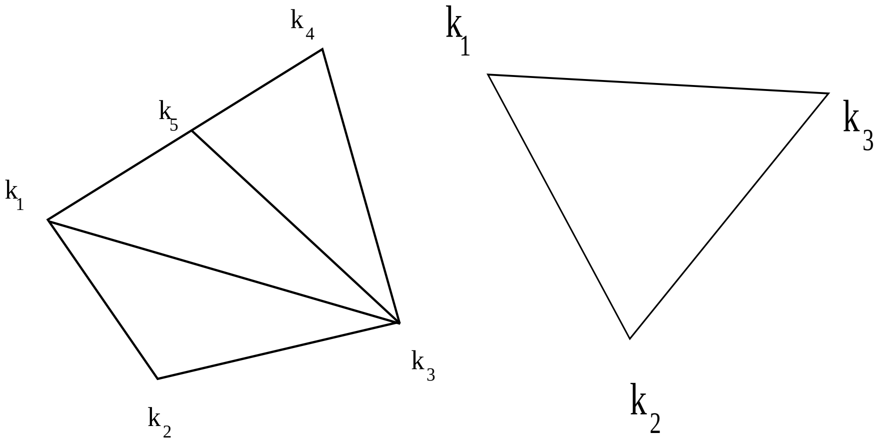

Figure 5: The coarse grid elements

- V_c: Vertices are the coarse grid nodes
- E_c: Two vertices are connected by an edge if and only if they are in one and the same triangle.
- V_c': Vertices are the triangles we have formed.
- E_c': Two triangles are connected by an edge if they share an edge in G_c.

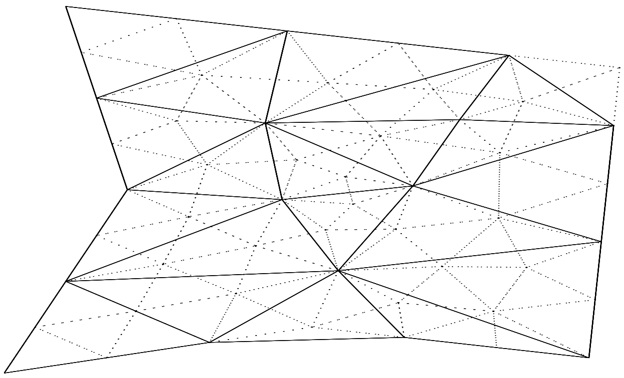

Figure 6: Coarse grid graph

The issue we have to address here is that in this way we might not get a valid finite element triangulation. It might happened that for some edge in G_c, there are more than 2 triangles adjacent to it. But as we pointed out before we only need an auxiliary graph G_c' and a correspondence between E_c and E_c'

which we have defined. In Fig. 6 the graph G_c for the model finite element grid is plotted. As it can be seen, we obtained a valid triangulation and in practice this is often the case.

A simple application of the algorithm yields:

> If the fine grid is obtained by a successive halving refinement and if the MIS on each level coincide the original coarse grid set, then the macro-elements will coincide exactly with the underlined (nested) coarse grids.

Remark 4.4 *Note that although the multilevel sequence of grids is non-nested the corresponding finite dimensional spaces are nested, because the basis in* M_{k-1} *is always defined as a linear combination of the fine grid basis via the interpolation. It is also clear from the definition that the arithmetic average interpolation preserves the constant function in each macroelement on all levels.*

5 On the convergence

In this section, we briefly discuss the convergence of the aforementioned multigrid method. We shall prove a result concerning the stability and approximation properties of the agglomerated coarse spaces. As a result we can conclude that our multigrid algorithm converges uniformly if the number of levels is fixed. We are yet to extend our result to truly multilevel case.

Given a triangulation $\mathcal{T}_h$ and the corresponding linear finite element space $M_h \subset H^1(\Omega)$, let $M_H \subset M_h$ be obtained by the agglomeration algorithm described in the previous section. Let $Q_H : H^1(\Omega) \to M_H$ be the L^2-projection. The assumption we make is for every macroelement G_H there exists an auxiliary big simplex K_H of diameter H, containing G_H together with all its neighboring elements from the fine grid. We also assume that $H/h \leq c$, for some constant c.

We claim that for every $v \in H^1(\Omega)$ the following stability and approximation properties hold:

$$
\begin{aligned}
\|Q_H v\|_{1,\Omega} &\leq C\|v\|_{1,\Omega}, &(8)\\
\|v - Q_H v\|_{0,\Omega} &\leq CH|v|_{1,\Omega}. &(9)
\end{aligned}
$$

We shall give detailed proof of our claim. Our proof is based on an averaged nodal value interpolation similar to the one described in Scott and Zhang [14]. Given any "coarse node" x_k, let F_k be an $n-1$ dimensional face from T_h that contains x_k. Let $\psi_k(x)$ be the linear function on F_k such that

$$< v, \psi_k >_{0,F_k} = v(x_k) \quad \forall v \in \mathcal{P}_1(F_k).$$

Now define $\Pi_H : H^1(\Omega) \to M_H$ by

$$(\Pi_H v)(x_k) = < v, \psi_k >_{0,F_k}$$

for each coarse node x_k, and the value of $\Pi_H v$ on all other fine grid nodes are determined by the prolongation operator. We claim that for any $v \in H^1(\Omega)$

$$\begin{aligned} \|\Pi_H v\|_{1,\Omega} &\leq C\|v\|_{1,\Omega}, & (10)\\ \|v - \Pi_H v\|_{0,\Omega} &\leq CH|v|_{1,\Omega}. & (11) \end{aligned}$$

We shall first prove (11). By the extension theorem, we may assume that $v \in H^1(\mathbb{R}^n)$ satisfying

$$|v|_{1,\mathbb{R}^n} \leq C|v|_{1,\Omega}.$$

Let now G_H be a macroelement. By construction we can find an auxiliary big simplex K_H (with diameter bounded by cH) that contains G_H together with all its neighboring elements from the fine grid. Now let us introduce the affine mapping $K_H \to \hat{K}$, where $\hat{K}$ is the standard reference element. Correspondingly we will have $G_H \to \hat{G}$, $v \to \hat{v}$, and $\Pi_H v \to \hat{\Pi}\hat{v}$.

We now consider $\hat{\Pi}$. It is easy to see that by trace theorem we have

$$\|\hat{v} - \hat{\Pi}\hat{v}\|_{0,\hat{G}} \leq C\|\hat{v}\|_{1,\hat{K}}, \quad \forall \hat{v} \in H^1(\hat{K})$$

and by construction $\hat{\Pi}$ is invariant on constant functions, namely $\hat{\Pi}\hat{c} = \hat{c}$, for any $\hat{c} \in \mathbb{R}^1$. Therefore

$$\begin{aligned} \|\hat{v} - \hat{\Pi}\hat{v}\|_{0,\hat{G}} &= \inf_{\hat{c} \in \mathbb{R}^1} \|\hat{v} + \hat{c} - \hat{\Pi}(\hat{v} + \hat{c})\|_{0,\hat{G}} \\ &\leq C \inf_{\hat{c} \in \mathbb{R}^1} \|\hat{v} + \hat{c}\|_{1,\hat{K}} \leq C|\hat{v}|_{1,\hat{K}}. \end{aligned}$$

By scaling back to K_H we get

$$\|v - \Pi_H v\|_{0,G_H} \leq CH|v|_{1,K_H}.$$

Summing over all macroelements we have

$$\begin{aligned} \|v - \Pi_H v\|_{0,\Omega}^2 &\leq \sum_{G_H \subset \Omega} \|v - \Pi_H v\|_{0,G_H}^2 \\ &\leq CH^2 \sum_{K_H \supset G_H} |v|_{1,K_H}^2 \\ &\leq CH^2|v|_{1,\mathbb{R}^n}^2 \leq CH^2|v|_{1,\Omega}^2. \end{aligned}$$

This proves (11)

We shall now prove (10). The proof uses the standard scaling argument and invariance of $\hat{\Pi}$ on $\mathcal{P}_0(\hat{K})$. We have

$$\begin{aligned} |\Pi_H v|_{1,G_H} \leq CH^{\frac{n}{2}-1}|\hat{\Pi}\hat{v}|_{1,\hat{G}} &= CH^{\frac{n}{2}-1} \inf_{\hat{c} \in \mathbb{R}^1} |\hat{\Pi}(\hat{v} + \hat{c})|_{1,\hat{G}} \\ &\leq CH^{\frac{n}{2}-1} \inf_{\hat{c} \in \mathbb{R}^1} \|\hat{v} + \hat{c}\|_{1,\hat{K}} \leq CH^{\frac{n}{2}-1}|\hat{v}|_{1,\hat{K}}. \end{aligned}$$

By scaling back to K_H we get the desired estimate (10).

Consequently

$$\|v - Q_H v\|_{0,\Omega} \leq \|v - \Pi_H v\|_{0,\Omega} \leq CH|v|_{1,\Omega}.$$

and

$$\begin{aligned} |Q_H v|_{1,\Omega} &\leq |Q_H v - \Pi_H v|_{1,\Omega} + |\Pi_H v|_{1,\Omega} \\ &\leq C(h^{-1}\|Q_H v - \Pi_H v\|_{0,\Omega} + \|v\|_{1,\Omega}) \leq C|v|_{1,\Omega}. \end{aligned}$$

By the convergence theory in Bramble, Pasciak, Wang, Xu [4] we use the estimates (8) and (9) to conclude that: *the agglomeration multigrid algorithm converges uniformly with respect to h if the number of levels is fixed.*

6 Numerical examples

We consider the Laplace equation:

$$\begin{cases} -\Delta u = 1, \quad (x,y) \in \Omega \subset \mathrm{I\!R}^2, \\ u(x,y) = 0, \quad (x,y) \in \partial\Omega. \end{cases} \tag{12}$$

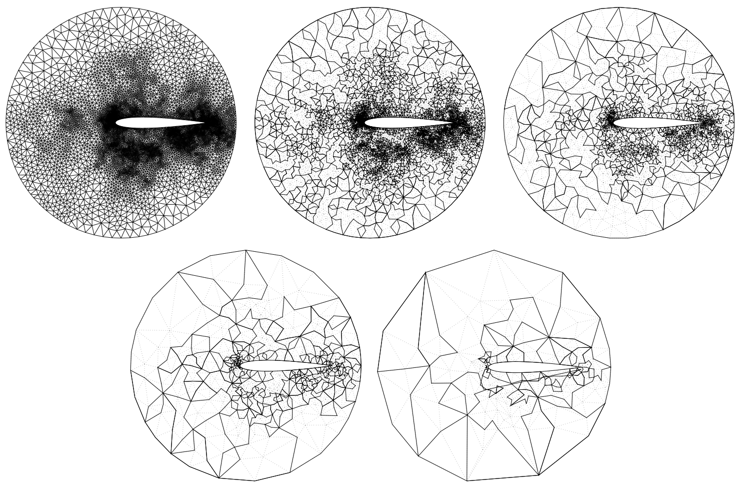

Figure 7: Macroelements for one element airfoil: level= 5 $N_h = 12665$; level= 4 $N_H^1 = 3404$; level= 3 $N_H^2 = 928$; level= 2 $N_H^3 = 257$; level= 1 $N_H^4 = 74$; level= 0 $N_H^5 = 24$.

In these examples we use the standard V-cycle preconditioner and the outer acceleration is done by the conjugate gradient method. In the V-cycle we use 1-pre and 1-post smoothing steps. The smoothing operator is forward Gauß-Seidel. The PCG iterations are terminated when the relative residual is less than 10^{-6}. We also present the examples using the variable V-cycle , doubling the smoothing steps on each level. We are interested in checking numerically the convergence of PCG preconditioned with V-cycle based on the simple interpolation we derived.

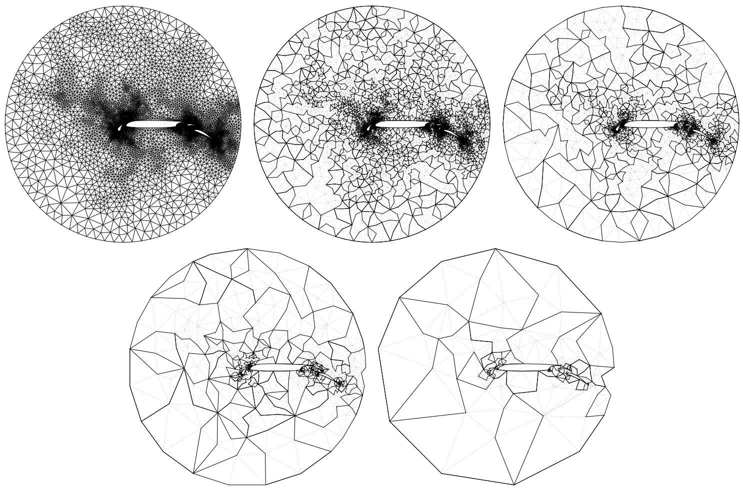

Figure 8: Macroelements for four element airfoil: level= 5 $N_h = 12850$; level= 4 $N_H^1 = 3444$; level= 3 $N_H^2 = 949$; level= 2 $N_H^3 = 270$; level= 1 $N_H^4 = 80$; level= 0 $N_H^5 = 26$.

In Figures 7–8, we plot the macroelements for different unstructured grids and different number of levels to illustrate the coarsening algorithm. These fine grids are obtained by Delaunay triangulation of randomly placed point sets. They are not obtained by any refinement procedure. Figure 9 shows the convergence histories for a varying number of unknowns on two types of grids. One of these (one-element airfoil) has one internal boundary, the other one has four internal boundaries.

As interpolation, we use the one described in § 4.3. The numerical experiments suggest that for isotropic problems (such as Laplace equation), the convergence of the variable V-cycle seems to be uniform with respect to the mesh size h.

References

[1] R. Bank and J. Xu, *An algorithm for coarsening unstructured meshes*, Numer. Math. **73** (1996), no. 1, 1–36.

[2] D. Braess, *Towards algebraic multigrid for elliptic problems of second order*, Computing **55** (1995), 379–393.

1-element airfoil

nodes	Reduction factor
$N_h = 72139$	0.27049
$N_h = 18152$	0.22695
$N_h = 4683$	0.19406
$N_h = 1315$	0.15017

4-element airfoil

nodes	Reduction factor
$N_h = 72233$	0.28608
$N_h = 18328$	0.21359
$N_h = 4870$	0.18889
$N_h = 1502$	0.14799

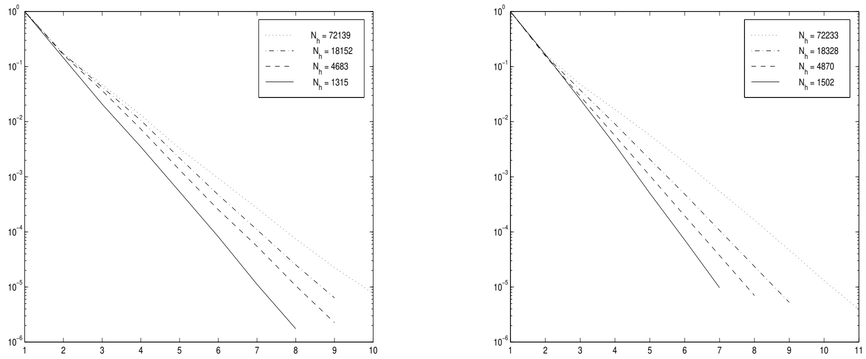

Figure 9: Convergence history and average reduction per iteration for varying number of unknowns, V-cycle

[3] J. Bramble, *Multigrid methods*, Pitman, Notes on Mathematics, 1994.

[4] James H. Bramble, Joseph E. Pasciak, Junping Wang, and Jinchao Xu, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp. **57** (1991), no. 195, 23–45.

[5] T. F. Chan and Barry Smith, *Domain decomposition and multigrid methods for elliptic problems on unstructured meshes*, Domain Decomposition Methods in Science and Engineering, Proceedings of the Seventh International Conference on Domain Decomposition, October 27-30, 1993, The Pennsylvania State University (David Keyes and Jinchao Xu, eds.), American Mathematical Society, Providence, 1994, also in Electronic Transactions on Numerical Analysis, v.2, (1994), pp. 171–182.

[6] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multi-grid by smoothed aggregation for second and forth order elliptic problems*, Computing **56** (1996), 179–196.

1-element airfoil

nodes	Reduction factor
$N_h = 72139$	0.18255
$N_h = 18152$	0.16627
$N_h = 4683$	0.14028
$N_h = 1315$	0.12149

4-element airfoil

nodes	Reduction factor
$N_h = 72233$	0.17827
$N_h = 18328$	0.14621
$N_h = 4870$	0.13057
$N_h = 1502$	0.13269

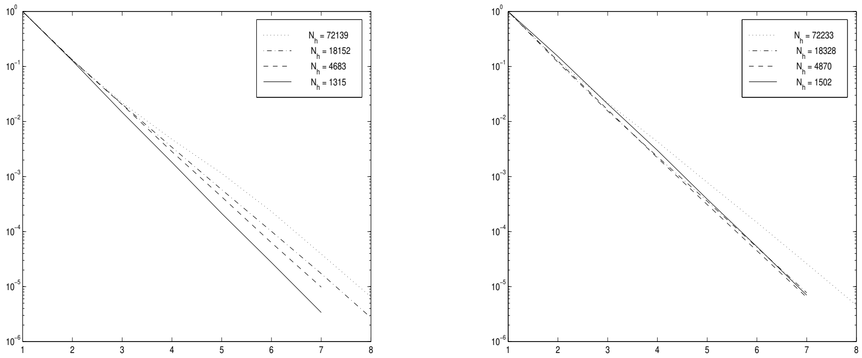

Figure 10: Convergence history and average reduction per iteration for varying number of unknowns, variable V-cycle

[7] P. Vaněk and J. Křižkovà, *Two-level preconditioner with small coarse grid appropriate for unstructured meshes*, Numer. Linear Algebra Appl. **3** (1996), no. 4, 255–274.

[8] H. Guillard, *Node-nested multi-grid method with Delaunay coarsening*, Tech. Report RR-1898, INRIA, Sophia Antipolis, France, March 1993.

[9] W. Hackbusch, *Multi-grid methods and applications*, Springer Verlag, New York, 1985.

[10] W. Hackbusch and S. A. Sauter, *Composite finite elements for the approximation of PDEs on domains with complicated micro-structures*, Numerische Mathematik **75** (1995), 447–472.

[11] B. Koobus, M. H. Lallemand, and A. Dervieux, *Unstructured volume-agglomeration MG: solution of the Poisson equation*, International Journal for Numerical Methods in Fluids **18** (1994), no. 1, 27–42.

[12] A. A. Reusken, *A multigrid method based on incomplete Gaussian elimination*, Numer. Linear Algebra Appl. **3** (1996), no. 5, 369–390.

[13] J. W. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid methods (Philadelphia, Pennsylvania) (S. F. McCormick, ed.), Frontiers in applied mathematics, SIAM, 1987, pp. 73–130.

[14] L. R. Scott and S. Zhang, *Finite element interpolation of nonsmooth functions satisfying boundary conditions*, Math. Comp. **54** (1990), 483–493.

[15] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review **34** (1992), 581–613.

[16] J. Xu, *The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids*, Computing **56** (1996), 215–235.