AN ALGEBRAIC COMPOSITE FINITE ELEMENT MESH METHOD

S.S. Sarraf[*,a], E.J. López[a], V.E. Sonzogni[a], M.B. Bergallo[b]

[a] *Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), INTEC CONICET, Universidad Nacional del Litoral, Güemes 3450, Santa Fe, Argentina*
[b] *Departamento de Matemática, Facultad de Ingeniería Química, Universidad Nacional del Litoral, Santiago del Estero 2829, Santa Fe, Argentina*

Abstract

The composite finite element mesh method is useful for discretization error estimation and, in addition, for solution improvement with no increment in the computational cost. The technique consists in redefine over a given mesh the linear operator that arises from the discretization of a partial differential equation. This operator is modified according to an appropriate linear combination between the operators of the given mesh and of a coarse mesh, which must be a coarsening of the first one. In this work a novel algebraic composite mesh technique is proposed. The technique uses some tools from the Algebraic Multigrid method for the definition of the coarse mesh and the discrete space associated with it. Mesh coarsening is based on the fusion of elements in macroelements, with a new definition of the grid topology and basis functions. The agglomeration of elements is made in order to reduce the mesh anisotropy, which is of importance in the discretization of convection-diffusion-reaction problems. The discrete operator for the coarser mesh is obtained by the Galerkin Coarse Approximation, where inter-grid transfer operators are obtained using the graph of the coarse mesh. Several test problems with different boundary conditions are presented.

Key words: Algebraic Composite Mesh, Discretization Errors, Numerical solution improvement

[*]Corresponding author.
Email address: ssarraf@santafe-conicet.gov.ar (S.S. Sarraf)

1. Introduction

In this paper we present a new technique of Composite Mesh (CM) where each component of the mixture represents a mesh with different approximation error. Given two finite element meshes with nodes in common, an appropriate linear combination between the discrete operators that arise on each grid could give a better solution than the solutions obtained from each mesh individually and without increment in the computational cost [1].

The Algebraic Composite Mesh (ACM) technique presented in this work is the algebraic version of the Geometric Composite Mesh (GCM) strategy proposed by Bergallo *et al.* [1]. Starting from a finite element mesh, the coarse mesh that will take part in the linear combination comes from the element agglomeration of the first one. Such agglomeration is made by means of techniques taken from the Algebraic multigrid (AMG) method. Particularly in this paper, a modification of the algorithm for the construction of macroelements proposed by Okusanya [2] is employed. Then, the elements of a given mesh are properly agglomerated in order to obtain a new mesh where the elements do not necessarily preserve the geometric shape of the elements of the initial mesh.

It has been shown that it is feasible to use the GCM technique for unstructured meshes [3]. The resultant linear system obtained when the CM technique is applied can be solved by means of a direct method or an iterative method as, for example, the multigrid strategy [3]. In this paper, an algebraic composite finite element mesh technique for unstructured meshes is presented. The method is applied to some elliptic test problems on unstructured meshes, where discretization errors are analyzed. The resulting linear system of equations is solved by a direct method.

2. Algebraic Composite Mesh technique

We consider elliptic problems with the following form

$$\begin{aligned} -\nabla\cdot(\mu(x)\nabla u) &= f(x), \qquad \text{in } \Omega \\ u &= g_D, \qquad \text{on } \Gamma_D \\ \mu(x)\nabla u\cdot\mathbf{n} &= g_N, \qquad \text{on } \Gamma_N \end{aligned} \tag{1}$$

where the domain $\Omega \subset \mathbb{R}^{nde}$ is delimited by the boundary Γ, nde being the number of spatial dimensions, and $\mathbf{n}$ is the unit normal to Γ. The boundary domain is generally formed by parts with Dirichlet conditions (Γ_D) and

Neumann conditions ($\Gamma_N = \Gamma \setminus \Gamma_D$). The function $f(x)$ is the source term and $\mu(x) > 0 \, \forall x \in \Omega$ is the diffusivity coefficient.

The discretization of equation (1) leads to a system of linear equations as

$$\mathbf{A}_h \mathbf{u}_h = \mathbf{b}_h \tag{2}$$

where $\mathbf{A}_h$ is the discrete operator (stiffness matrix), $\mathbf{u}_h$ is the nodal unknowns vector and $\mathbf{b}_h$ is the constant terms vector. The subscript h refers to the element size of the mesh utilized in the discretization.

The goal of this work is to propose an Algebraic Composite Mesh (ACM) technique, as the algebraic version of the Composite Mesh (CM) method [1]. The composite finite element mesh applied to elliptic problems can be used to improve the numerical solution without an appreciable increment in the computational cost and also estimate the discretization error [1, 3, 4]. In the h version, the method consists in replacing the discrete operator for a given mesh (fine mesh), by a linear combination of the operators computed by using that mesh and a coarser mesh with nodes in common with the first one. In this case, the interpolation polynomials retain the same degree in both meshes. Then, assuming that the fine mesh is obtained from the homogeneous refinement of other grid, the connection between both meshes is forced with the shared nodes. The participation factor of each mesh in the compound model, *i.e.* the coefficient for the linear combination between the meshes, is introduced in such a way to minimize the discretization error.

Let Ω_H a discretization of the problem domain Ω and Ω_h the mesh obtained by a homogeneous refinement of Ω_H. Applying the Finite Element Method (FEM) for the discretization of problem (1), the systems of equations $\mathbf{A}_h \mathbf{u}_h = \mathbf{b}_h$ and $\mathbf{A}_H \mathbf{u}_H = \mathbf{b}_H$ are obtained with the meshes Ω_h and Ω_H, respectively. Now, we define the discrete operator $\mathbf{A}_{Hh}$ as the matrix with the coefficients of $\mathbf{A}_H$ and the same size as $\mathbf{A}_h$ where, in order to obtain the required size, we impose null coefficients for the nodes belonging to the fine mesh but not to the coarse mesh. In an analogous way, we define the vector $\mathbf{b}_{Hh}$. The approximate solution by the CM method $\mathbf{u}_{Hh}$ is obtained from the following system [4]

$$[\alpha \mathbf{A}_h + (1-\alpha)\mathbf{A}_{Hh}]\mathbf{u}_{Hh} = \alpha \mathbf{b}_h + (1-\alpha)\mathbf{b}_{Hh} \tag{3}$$

When $\alpha = 1$, the solution in the fine mesh is recovered and with $\alpha \to 0$ the solution tends to the corresponding solution in the coarse grid. The coefficient

α depends on the regularity of the exact solution of the problem [1]. The asymptotic error of the numerical approximation has the form

$$\|u - u_h\| = Ch^p + \mathcal{O}(h^q) \tag{4}$$

where C is a constant and $q > p$. Then, an extrapolation analysis of the error leads to the following estimation [1, 4, 5]

$$\alpha = \frac{(H/h)^p}{(H/h)^p - 1} \tag{5}$$

The improvement introduced by the CM method with respect to the FEM solution is verified in the nodal values of the solution and, thus, it must be evaluated using a discrete norm of the error. Let N_i, $i = 1, \ldots, M$ the nodes of the fine mesh Ω_h and V_h the discrete space associated with the mesh Ω_h. The interpolant $\pi_h u$ of u in the space V_h is defined as

$$\pi_h u(N_i) = u(N_i), \quad i = 1, \ldots, M \tag{6}$$

Then, the solution of the system (3) is a better approximation to $\pi_h u$ in V_h than the u_h solution given by FEM. This fact is verified in the tests presented in section §5.

As presented above, the GCM method requires a fine mesh coming from the homogeneous refinement of the coarse mesh. In addition, the discretization of equations must be done with both meshes in order to obtain the discrete operators $\mathbf{A}_h$ and $\mathbf{A}_H$. These two issues motivate us to propose a CM technique based on algebraic manipulation of the discrete operator computed with a given mesh. To do this we need to make a coarsening of that mesh, and define in a proper way the coarse space and the coarse discrete operator. Since the method involves two meshes with different discretization levels (element size), we can take some strategies from the AMG method to compute the mesh coarsening.

A unique coefficient in the linear combination for the whole set of mesh elements in equation (3) is used because the homogeneous refinement of the coarse mesh preserves the relationship H/h for all elements. In the proposed algebraic technique the coarse grid is obtained from the coarsening of a (generally unstructured) mesh and, thus, the element size ratio between the meshes will not be constant through the domain. Hence, in order to

generalize the formulation, we could apply a matrix $\boldsymbol{\alpha}$ in the linear combination as a variable participation factor. Then, the linear system for the ACM technique is written as

$$[\boldsymbol{\alpha}\mathbf{A}_h + (\mathbf{I} - \boldsymbol{\alpha})\mathbf{A}_{Hh}]\mathbf{u}_{Hh} = \boldsymbol{\alpha}\mathbf{b}_h + (\mathbf{I} - \boldsymbol{\alpha})\mathbf{b}_{Hh} \tag{7}$$

where $\mathbf{I}$ is the identity matrix.

3. Elements of the Algebraic Multigrid method

The multigrid method essentially considers a decomposition of the solution error into rough error components, which cannot be resolved on a coarser grid without aliasing, and the complementary smooth error component which can be resolved on the coarser grid. Given a hierarchical sequence of successively coarser grids, a recursive partitioning of the solution error may be made amongst these grids such that the associated error components on each coarse grid effectively form a basis for the smooth error component on the finest grid. The partitioning of the error is achieved through a set of interpolation (prolongation/restriction) operators for the transfer of error components between the spaces.

Consider two consecutive grids from the hierarchical sequence of meshes cited above. Let Ω_h the fine mesh of the pair and Ω_H the coarse mesh. The restriction operator projects vectors from the fine grid to the coarse one: $\mathbf{R}_h : \mathbb{R}^{n_h} \rightarrow \mathbb{R}^{n_H}$, where $n_h > n_H$ are the dimensions of the spaces of finite dimensions associated with the grids Ω_h and Ω_H respectively. The interpolation of vectors from the coarse mesh to the fine grid is done by means of the prolongation operator $\mathbf{P}_h : \mathbb{R}^{n_H} \rightarrow \mathbb{R}^{n_h}$ (see, for example, Mavriplis [6], Wesseling [7], Briggs *et al.* [8], etc.).

A multigrid algorithm requires an approximation to the operator of the fine grid within the space of the coarse mesh. The matrix for that space is called $\mathbf{A}_H$ and could be calculated by the following algebraic recursive relationship

$$\mathbf{A}_H = \mathbf{R}_h\mathbf{A}_h\mathbf{P}_h \tag{8}$$

where, although the operators $\mathbf{R}_h$ and $\mathbf{P}_h$ are independent, the choice $\mathbf{R}_h = \mathbf{P}_h^t$ results in the minimization of the error in the solution measured in the A-norm $\|\cdot\|_{\mathbf{A}_h}$ after coarse grid corrections, for symmetrical and positive definite stiffness matrices [7, 8]. This algebraic method for coarse space operator construction is called Galerkin Coarse Grid Approximation (GCA).

A promising technique for the construction of these interpolation operators which satisfy the rules outlined above is based on the agglomeration technique [9, 10] which operates by fusing neighboring fine mesh entities to form coarse mesh macroentities. This provides a natural and automatic way for coarse space construction. The agglomeration technique defined on the Finite Element space can be vertex based (nodal) [11] or element based [12, 13]. The choice of elemental agglomeration is motivated by the need to address higher order of accuracy for the interpolation operators [14].

4. Coarse space agglomeration

The coarse space agglomeration applied in this work was proposed by Okusanya [2] and is based on the fusion of elements into macroelements with a proper definition of the coarse grid topology and basis functions. The driving force behind the agglomeration is the reduction of mesh anisotropy, which becomes important for convection-diffusion problems [2]. The main difference between this method and the one described by Chan *et al.* [14] is that the coarse mesh elements are not converted in standard elements by retriangulation, but are general polygons formed by agglomeration of fine mesh elements.

The topology of the coarse mesh is built partitioning the elements in groups of macroelements. A macroedge is defined to be the ordered collection of fine grid edges which are shared by two neighboring macroelements. To complete the definition of the coarse grid graph, the coarse nodes are chosen to be the fine grid nodes where three or more macroedges meet. Macrolements with exactly two coarse nodes are modified by the addition of extra supporting coarse nodes using fine mesh nodes which lie on the macroedge connecting these two coarse nodes, as shown in figure 1.

4.1. Elemental agglomeration algorithm

The coarsening algorithm consists in a modification of the elementary agglomeration algorithm proposed by Okusanya [2]. It is based on the removal of mesh anisotropy and makes use of edge lengths such that the geometry for the coarse spaces is defined entirely in terms of the fine grid. That is, the macroedge lengths are simply the sum of the edge lengths of the constituting fine mesh edges. If the fine grid geometry is not specified, then this technique becomes a purely topological one where the elements are assumed to

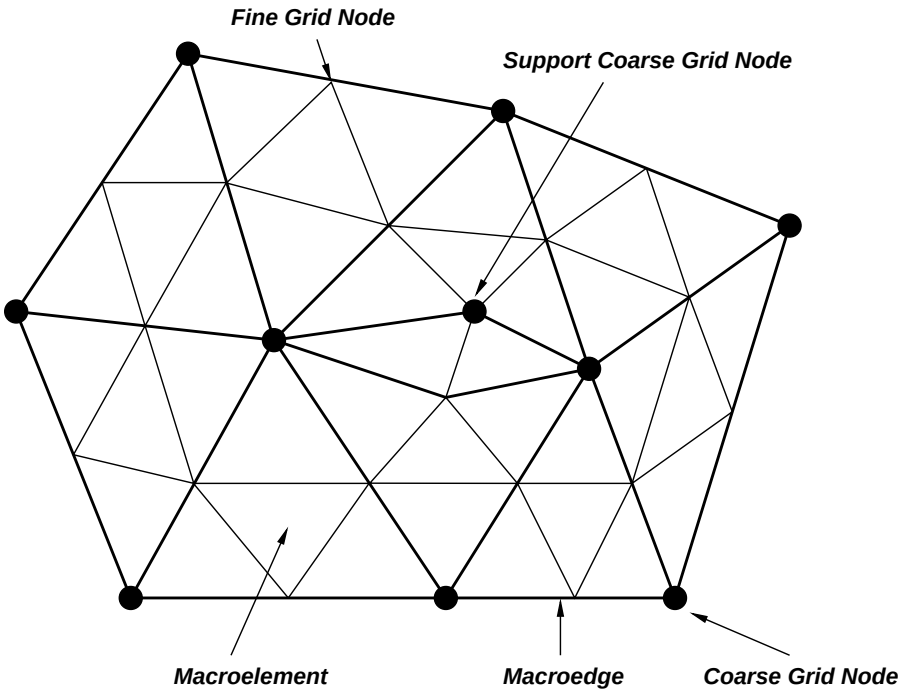

Figure 1: Coarse space topology with exceptional macroelement bearing two coarse nodes and extra support node.

be isotropic. The decision to agglomerate two neighboring elements is determined by a geometry based connectivity concept which is term *macroelement skew* [2].

Definition 1: *For a macroelement defined by a general polygon, the macroelement skew is a measure of anisotropy and is defined as the area of the n-gon divided by the area of an isotropic n-gon with the same perimeter.*

In extreme cases, this measure is zero for co-linear polygon vertices and unity for an isotropic n-gon. The concept of macroelement skew can be extended to 3D through a suitable redefinition such as ratio of macroelement volume to macroelement circumsphere volume similar to the control volume skew described by Venkatakrishnan *et al.* [15]. The macroelemental areas for the coarse spaces are also easy to compute as they are simply sums of the agglomerated element areas. In order to complete the operators required for this algorithm, an edge based connectivity concept called *edge skew* is

defined.

Definition 2: *For an element which borders a macroelement/element on a given edge, edge skew is defined as the macroelement skew of the macroelement which would be created if the element is merged with the macroelement/element across that edge.*

Now, we present the algorithm in detail:

Procedure 1: *Macroelemental Construction*

- **Step 1:** Consider the graph of the mesh $G = (V, E)$ and calculate the edge length for the edges E.
- **Step 2:** Obtain a seed element: If there is no seed element in the queue, choose any suitable element which does not belong to any macroelement.
- **Step 3:** Perform accretion around the seed element. Fuse unassigned neighboring elements with edge skew larger than some specified fraction (typically 0.75) of the average edge skew.
- **Step 4:** Enqueue seed elements. New seed elements are placed in the queue to continue the algorithm. These are chosen to be elements which share a vertex but no edges with the last macroelement created. In 3D, this would extended to elements which share a vertex and/or an edge but no faces with the macroelement.
- **Step 5:** Repeat **Step 2** until either all elements belong to a macroelement or there are no more seed elements.

After the algorithm terminates, post-processing is necessary to deal with 'sliver' elements. These are fine mesh elements which were not originally selected by the algorithm to be merged into a macroelement. A determination of which macroelement to merge these elements with is made *a priori* based on skew edge.

In the step 2 of the macroelemental construction algorithm it is given the instruction to choose an appropriate seed element, which is somewhat vague. In fact, the method to select the proper seed element leads to totally different results. When the algorithm is applied to a structured mesh, it

would be desirable retrieve to a 4:1 isotropic agglomeration in 2D. Precisely, the election of the seed element in step 2 will determine the success of the implementation of the algorithm. In this work, we make the selection of a seed element based on labels of the nodes, being able to obtain the ratio 4:1 in structured meshes.

4.2. Interpolation and restriction operators

The construction of the interpolation operators may be facilitated by the definition of nodal basis functions on the coarse space and serves as a natural extension of the Finite Element algorithm on these coarse spaces. For the coarse space basis functions construction we follow the method proposed by Okusanya [2]. In this method, the construction of the basis functions is made by using the topology and geometry if provided. If the geometry is not given, then the elements are assumed to be isotropic, which leads to a purely topological interpolant. The basis functions are defined using graph distance interpolation on both the boundary and interior, which is geometric weighted to form a more accurate interpolant. The algorithm leads to a quasi-linear interpolant as shown in Figure 2.

Figure 2 shows the basis function Φ defined over the agglomerated macroelement for the coarse mesh node at $\mathbf{x}_0$. This basis function is constructed from a graph distance interpolation over the macroelement and weighted with edge length.

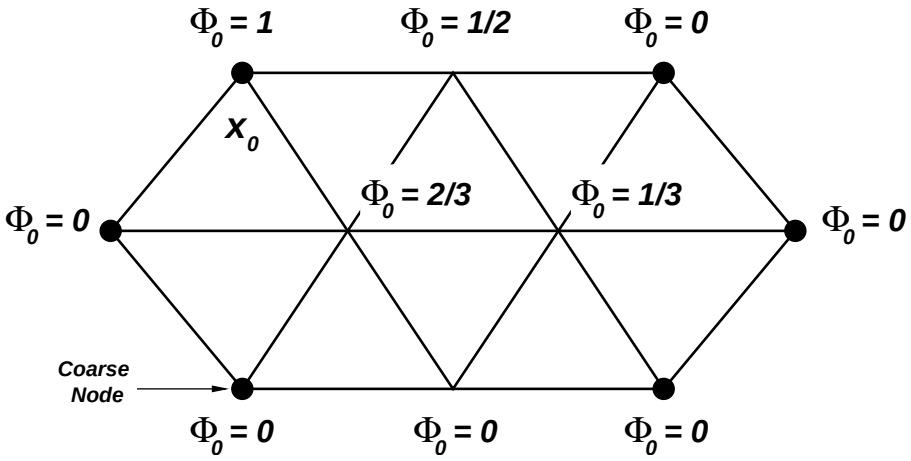

Figure 2: Coarse space basis function based on graph distance.

An important component of the algorithm for the coarse space basis functions construction is the Breadth First Search (BFS) algorithm, which is essentially a Greedy algorithm for graph traversal. The detailed BFS algorithm is given in appendix A. The algorithm for the construction of the coarse space interpolant is given below.

Procedure 2: *Basis Function Construction*

- **Step 1:** For each macroelement, create a local subgraph. In the process, create an ordering of the boundary edges such that the boundary can be traversed.
- **Step 2:** Extract the list of interior vertices. Extract the ordered list of coarse grid vertices by traversing the boundary edges.
- **Step 3:** For all fine mesh edge vertices which lie between consecutive coarse mesh nodes, construct length weighted interpolation data. The macroedge length is also computed simultaneously.
- **Step 4:** Interior vertex interpolation. For each coarse grid node in the macroelement, a BFS iteration on the local subgraph is done with the coarse grid node as seed. Both the level set as the graph distance from the coarse node is recorded for all interior (fine) nodes in the subgraph during the process. The graph distance of each fine grid node from the coarse grid nodes is then computed. For each fine grid node, these distances are then weighted to sum to unity.

The necessary matrix coefficients for the prolongation operator $\mathbf{P}_h$ may now be extracted from the basis functions. For the 2D case, the algorithm reads as follows

Procedure 3: *Prolongation Operator Construction*

- **Step 1:** For every fine mesh node i in a macroelement which corresponds to a coarse mesh node j, set the prolongation operator coefficient
$$\mathbf{P}_h(i,j) = 1 \tag{9}$$
- **Step 2:** For every other fine grid node i in macroelement which does not correspond to a coarse grid node, given the length weighted graph

distance $dist(i,j)$ from every coarse node j, set the prolongation operator coefficient

$$\mathbf{P}_h(i,j) = \frac{\frac{1}{dist(i,j)}}{\sum_j \frac{1}{dist(i,j)}} \tag{10}$$

The restriction operator $\mathbf{R}_h$ is simply defined using the GCA formulation. The success of the ACM method depends strongly on how good the approximation of the coarse space matrix $\mathbf{A}_H$ is to $\mathbf{A}_h$. Let us choose the restriction operator to be

$$\mathbf{R}_h = \boldsymbol{\sigma}\mathbf{P}_h^t \tag{11}$$

where $\boldsymbol{\sigma}$ is a suitable scaling factor. If $\mathbf{R}_h$ is to be used to transfer residuals to the coarse grid (as we need in the ACM technique), then the correct value of the scaling depends on the scaling of the fine grid and coarse grid problems. This implies that the coarse grid discretization should be consistent with the governing differential equation in the same way as the fine grid discretization. Let A_j represents a characteristic area (*e.g.* control volume area) on the fine grid associated with the fine node j and let $\bar{A}_i$ represents a corresponding characteristic area on the coarse grid associated with coarse node i. Finite Element schemes in 2D lead to a scaling rule which states that

$$\sum_j \mathbf{R}_h(i,j)A_j = \bar{A}_i \tag{12}$$

This rule is derived by considering the integral terms for the interior fluxes. However, the boundary flux integral terms are line integrals which need a modification of the restriction operator. Let L_j represents a characteristic length on the fine grid associated with boundary fine node j and let $\bar{L}_i$ represents a corresponding characteristic length on the coarse grid associated with boundary coarse node i. The corresponding scaling rule for the restriction operator as applied to the boundary integral term is

$$\sum_j \mathbf{R}_h(i,j)L_j = \bar{L}_i \tag{13}$$

In order to deal with the dual scaling issues, Okusanya [2] proposes to introduce the splitting:

$$\mathbf{A}_h = \mathbf{A}_h^{gal} + \mathbf{A}_h^{bc} \tag{14}$$

where $\mathbf{A}_h^{gal}$ consists of the Galerkin terms which scale with area and $\mathbf{A}_h^{bc}$ consists of the boundary condition terms which scale with length. Based on

the splitting (14), the restriction operator is split into $\mathbf{R}_h^{gal}$ and $\mathbf{R}_h^{bc}$ such that GCA definition for the coarse space matrix becomes

$$\mathbf{A}_H = \mathbf{R}_h^{gal}\mathbf{A}_h^{gal}\mathbf{P}_h^{gal} + \mathbf{R}_h^{bc}\mathbf{A}_h^{bc}\mathbf{P}_h^{bc} \tag{15}$$

where $\mathbf{R}_h^{gal} = \boldsymbol{\sigma}^{gal}\mathbf{P}_h^{gal^t}$ and $\mathbf{R}_h^{bc} = \boldsymbol{\sigma}^{bc}\mathbf{P}_h^{bc^t}$. The construction for $\mathbf{P}_h^{gal}$ exactly follows the algorithm presented in section §4.2. However, $\mathbf{P}_h^{bc}$ is constructed by deleting the row entries for all the interior fine grid nodes in $\mathbf{P}_h^{gal}$.

Finally, the following splitting for the residual

$$\mathbf{r}_h = \mathbf{r}_h^{gal} + \mathbf{r}_h^{bc} \tag{16}$$

is introduced, such that a restriction for the residual can be written as

$$\mathbf{r}_H = \mathbf{R}_h^{gal}\mathbf{r}_h^{gal} + \mathbf{R}_h^{bc}\mathbf{r}_h^{bc} \tag{17}$$

5. Test problems

In order to show the effectiveness of the proposed ACM strategy, we will consider three different 2D cases with analytical solution. With these tests our goal is to evaluate the algebraic construction of the discrete operators independently of the mesh coarsening. Then, we use fine grids arising from the homogeneous refinement of a given mesh (the 'coarse mesh' in the corresponding problem). This particularization of the technique allows to use the formulation (3) instead of the most general represented by equation (7).

5.1. Poisson problem in a rectangular domain

Consider the problem of Poisson in a rectangular domain with a Neumann boundary condition given in (18).

$$\begin{aligned} -\Delta u(x,y) &= 2\pi^2 \cos\pi x \sin\pi y, \quad \text{in } \Omega = (1,2)\times(0,3) \\ u(x,0) &= 0, \quad \text{on } 1 \le x \le 2 \\ u(x,3) &= 0, \quad \text{on } 1 \le x \le 2 \\ u(1,y) &= -\sin\pi y, \quad \text{on } 0 \le y \le 3 \\ u_x(2,y) &= 0, \quad \text{on } 0 \le y \le 3 \end{aligned} \tag{18}$$

The exact solution is given by the expression

$$u(x,y) = \cos\pi x \sin\pi y \tag{19}$$

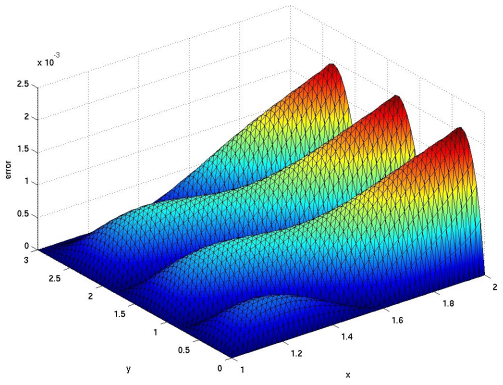

Figure 3: Nodal FEM errors between the numerical and exact solution for the Poisson problem in a rectangular domain.

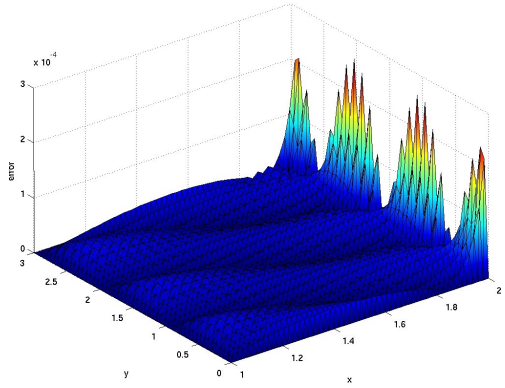

Figure 4: Nodal ACM errors between the numerical and exact solution for the Poisson problem in a rectangular domain.

In this case, the domain is discretized by an unstructured grid with 5000 elements and 2601 nodes. The resulting grid is then the fine mesh of the problem and the coarse one is obtained by means of the elemental agglomeration algorithm presented in section §4.1.

The participation factor for the mesh composition is obtained taking into account the regularity of the exact solution $\alpha = 4/3$.

Nodal errors between the numerical solution and exact solution are shown in figures 3 and 4 for FEM and ACM, respectively. The error results in standard L_∞ ($\|\mathbf{e}\|_\infty$) and euclidean norm ($\|\mathbf{e}\|_2$) are presented in table 1. As can be observed in table 1 the euclidean norm presents the best result due to the composite mesh gives an improvement of solution in a nodal sense.

Poisson problem in a rectangular domain		
	$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$
FEM	0.0393	2.1000×10^{-3}
ACM	0.0015	2.6424×10^{-4}

Table 1: Euclidean norm and L_∞ norm of errors for the Poisson problem in a rectangular domain.

5.2. Poisson problem in a domain with a hole

Consider the Poisson problem in a rectangular domain with an elliptic-shaped hole Ω as shown in figure 5 where the test equation and its boundary

Γ are given in equation 20.

$$\begin{aligned} \Delta u &= 10, \quad \text{in } \Omega \\ u(x,y) &= 4, \quad \text{on } \Gamma \\ u(x,-2) &= 16+x^2, \quad \text{on } -3 \leq x \leq 3 \\ u(x,2) &= 16+x^2, \quad \text{on } -3 \leq x \leq 3 \\ u(-3,y) &= 9+4y^2, \quad \text{on } -2 \leq y \leq 2 \\ u(3,y) &= 9+4y^2, \quad \text{on } -2 \leq y \leq 2 \end{aligned} \tag{20}$$

The exact solution is given by

$$u(x,y) = x^2 + 4y^2 \tag{21}$$

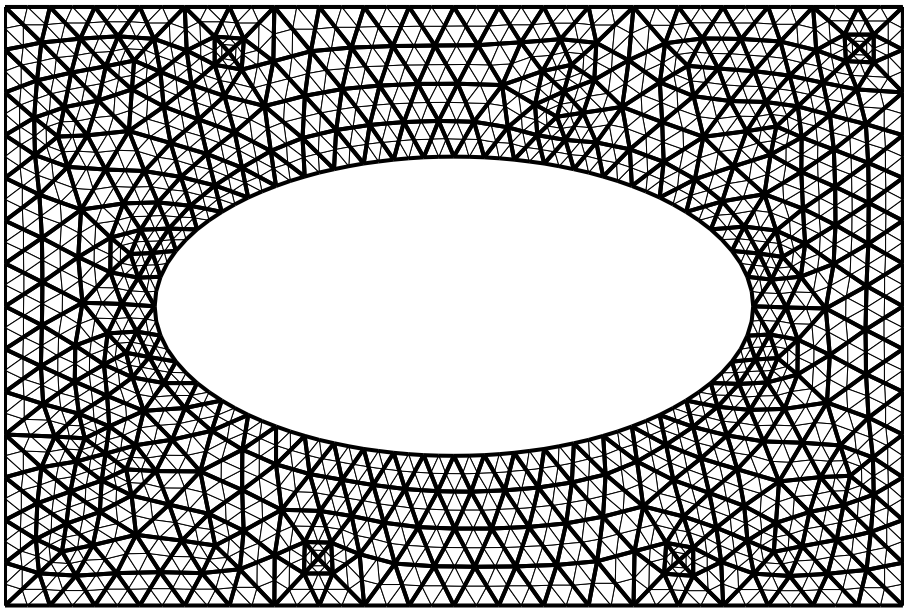

Figure 5: Fine and coarse meshes for the Poisson problem in a rectangular domain with an elliptic-shaped hole.

The domain is discretized by an unstructured grid with 2720 elements and 1476 nodes. The resulting grid is the fine mesh of the problem and the coarse one is obtained by means of the elemental agglomeration algorithm §4.1.

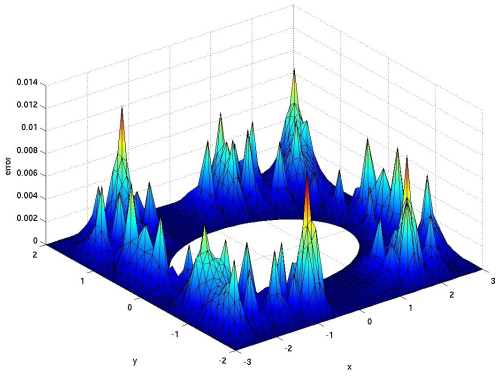

Figure 6: Nodal FEM errors between the numerical and exact solution for the Poisson problem with a domain with a hole.

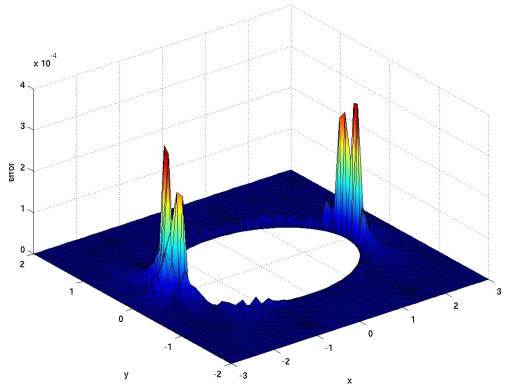

Figure 7: Nodal ACM errors between the numerical and exact solution for Poisson problem with a domain with a hole.

The participation factor for the mesh composition is obtained taking into account the regularity of the exact solution, then $\alpha = 4/3$ [3].

Nodal errors between the numerical solution and exact solution are shown in figures 6 and 7 for FEM and ACM, respectively. Table 2 presents the error L_∞ and euclidean norms. As can be observed in this table, the error measured in euclidean norm diminishes approximately a magnitude order for the ACM method with respect to FEM, which implies that the solution improvement is verified in the whole domain.

Poisson problem with a domain with a hole		
	$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$
FEM	0.0845	1.3000×10^{-2}
ACM	0.0010	3.5087×10^{-4}

Table 2: Euclidean norm and L_∞ norm of errors for the Poisson problem with a domain with a hole.

5.3. Laplace Problem

Consider the problem of Laplace $\Delta u = 0$ in a L-shaped domain as shown in figure 8 and where the exact solution is given in polar coordinates by

$$u(r, \phi) = r^{2/3} \sin \frac{2}{3}\phi \tag{22}$$

The domain is discretized by an unstructured grid with 4008 elements and 2085 nodes. The resulting grid is the fine mesh of the problem and the coarse one is obtained by means of the elemental agglomeration algorithm §4.1.

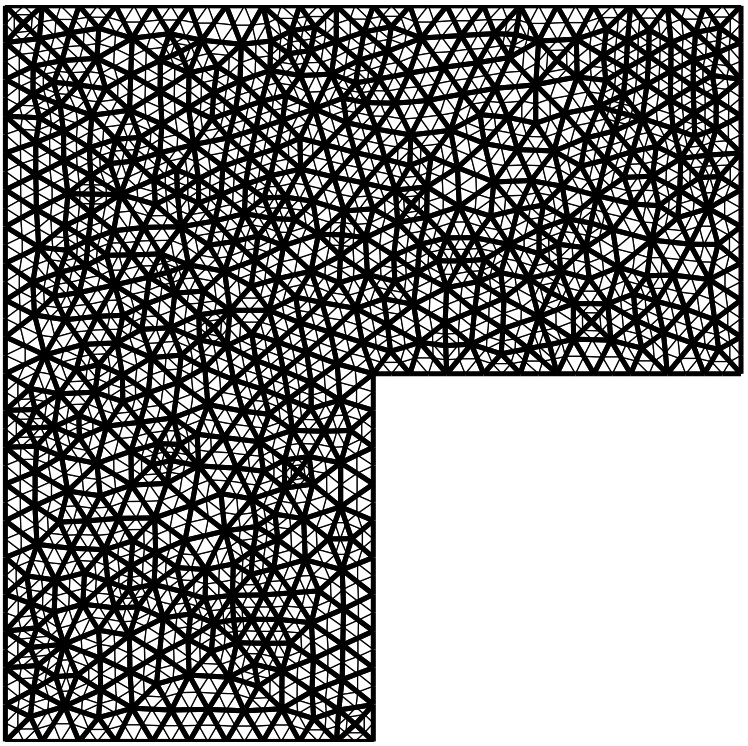

Figure 8: Fine and coarse meshes for the Laplace problem in a L-shaped domain.

The participation factor for the mesh composition is obtained taking into account the regularity of the exact solution [3]:

$$\alpha = \frac{2^{5/3}}{2^{5/3} - 1}$$

Nodal errors between the analytical solution and numerical solutions for FEM and ACM are shown in figures 9 and 10, respectively. The standard L_∞ and euclidean norms of the error are presented in table 3. Again, a reduction of the error norm can be reached when the ACM technique is applied. In this case the relative reduction between FEM and ACM errors is smaller than in previous examples due to the lower regularity in the exact solution.

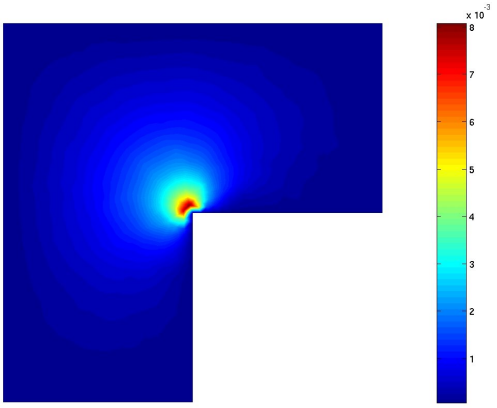

Figure 9: Nodal FEM errors between the numerical and exact solution for the Laplace problem.

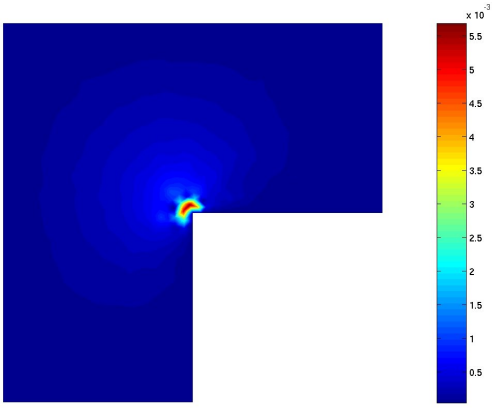

Figure 10: Nodal ACM errors between the numerical and exact solution for the Laplace problem.

L-shaped domain test problem		
	$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$
FEM	3.5390×10^{-2}	8.1450×10^{-3}
ACM	1.2665×10^{-2}	5.7305×10^{-3}

Table 3: Euclidean norm and L_∞ norm of errors for the Laplace problem.

6. Conclusion

In this work we present a novel Composite Mesh strategy based on algebraic definition of the coarse mesh topology and the discrete spaces and operators associated with it. The mesh coarsening tools were taken from an Algebraic Multigrid method that uses elemental agglomeration. The tests presented were solved using a fine mesh obtained from the homogeneous refinement of another grid. These test problems shown that the algebraic construction of the coarse operators performs correctly, allowing to reach with the ACM method the main features of its geometric counterpart. Using the optimum participation factor in the grid 'mixture' for the examples solved we obtain numerical approximations with smaller nodal errors than the produced by a FEM solution with the same discretization level.

Acknowledgments

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, grants PIP-02552/00,

PIP 5271/05), Universidad Nacional del Litoral (Argentina, grant CAI+D 2005-10-64) and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT 12-14573/2003 (Lambda), PME 209/2003 (Cluster)).

A. Breadth First Search algorithm

The definition for the BFS algorithm is [16]

Definition: *The Breadth First Search (BFS) is a search algorithm which considers neighbors of a vertex, that is, outgoing edges of the vertex's predecessor in the search, before any outgoing edges of the vertex such that extremes are searched last.*

The detailed description of the BFS algorithm is given in the following pseudocode

```
Unmark all vertices
Choose some starting vertex x
Mark x
Set list L = x
Set tree T = x
Set level set (LS) of x = 0
while L nonempty
  Choose some vertex v from front of list
  Visit v
  for each unmarked neighbor w
    Mark w
    Set LS(w) = LS(v)+1
    Add it to end of list
    Add edge (v,w) to T
```

References

[1] M.B. Bergallo, C.E. Neuman, V.E. Sonzogni, Composite mesh concept based FEM error estimation and solution improvement, Computer Methods in Applied Mechanics and Engineering, volume 188, pages 755-774, 2000.

[2] T. Okusanya, Algebraic Multigrid for Stabilized Finite Element Discretizations of the Navier Stokes Equations, Massachusetts Institute of Technology, 2002.

[3] S.S. Sarraf, M.B. Bergallo, V.E. Sonzogni, Problemas elípticos resueltos mediante mallas compuestas aplicando métodos Multigrilla, in Mecánica Computacional Vol. XXVI (ENIEF 07), pages 696-710, 2007.

[4] V.E. Sonzogni, M.B. Bergallo, C.E. Neuman, Uso de una malla compuesta para estimar errores de discretización y mejorar la solución en elementos finitos, in Mecánica Computacional Vol. XVI (MECOM 96), pages 260-263, 1996.

[5] S. Toro, V.E. Sonzogni, C.E. Neuman, Elementos finitos de diferentes órdenes para problemas de elasticidad plana y mezcla de sus mallas, in Mecánica Computacional Vol. XXIV, pages 3171-3185, 2005.

[6] D.J. Mavriplis, Multigrid techniques for unstructured meshes, Institute for Computer Applications in Science and Engineering Report, volume 95-27, 1995.

[7] P. Wesseling, An Introduction to Multigrid Methods, John Wiley & Sons, 1992.

[8] W.L. Briggs, V.E. Hemson, S.F. McCormick, A Multigrid Tutorial, SIAM, 2000.

[9] M.H. Lallemand, H. Steve, A. Dervieux, Unstructured multigridding by volume agglomeration: current status, Computational Fluids, volume 21, pages 397-433, 1992.

[10] B. Koobus, M.H. Lallemand, A. Dervieux, Unstructured volume agglomeration MG: solution of the Poisson equation, International Journal of Numerical Methods in Fluids, volume 18, pages 27-42, 1994.

[11] B.R. Hutchinson, G.D. Raithby, A Multigrid Method Based on the Additive Correction Strategy, Numerical Heat Transfer, volume 9, pages 511-537, 1986.

[12] M. Brezina, A.J. Cleary, R:D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, J.W. Ruge, Algebraic Multigrid

Based on Element Interpolation (AMGe), Society for Industrial and Applied Mathematics, volume 22, pages 1570-1592, 2000.

[13] J.E. Jones, P.S. Vassilevski, AMGe Based on Element Agglomeration, CASC Lawrence Livermore National Laboratory Report, 1999.

[14] T.F. Chan, S. Go, L. Zikatanov, Lecture Notes on Multilevel Methods for Elliptic Problems on Unstructured Grids, VKI 28th Computational Fluid Dynamics, pages 1-76, 1997.

[15] V. Venkatakrishnan, D. Mavriplis, Agglomeration Multigrid for the 3D Euler equations, AIAA, 94-0069, 1994.

[16] National institute of standards and definitions. http://hissa.nist.gov/dads/terms.html.