An Aggregation Multigrid Solver for Convection-diffusion Problems on Unstructured Meshes.

Hervé Guillard* and Petr Vaněk †

May 14, 1998

Abstract

We study an extension of the concept of smoothed aggregation to convection-diffusion equations. Since the approximation of the hyperbolic part of the equations yields unsymmetric matrices, the use of symmetric prolongator smoother is not efficient for the construction of the coarse convection matrices. This leads us to extend the concept of smoothed aggregation to the use of non-symmetric one-sided prolongator smoothers. In this paper, we also clarify the relationship that must exist between the degree of the prolongator polynomial smoother and the aggregation strategy. This work concludes by some numerical experiments illustrating the performance of the proposed method.

AMS-MOS Classification Primary: 65N55. Secondary: 65N220, 65T20.
Keywords : multigrid, cell centered discretization, finite volume, Aggregation methods

1 Introduction

A method that solves a N-unknowns problem in $\mathcal{O}(NlogN)$ operations is called optimal. For problems coming from the discretisation of partial differential equations on unstructured finite element type meshes, the multigrid (MG) method is the only known optimal method. Multigrid methods on unstructured meshes can be classified in two sets :

The first one gathers geometrical methods that use a hierarchy of explicitly built meshes : In Computational Fluid Dynamics, this approach has for instance being illustrated in [13]. It was observed that these methods can exhibit the

*INRIA, Sophia-Antipolis, BP 93, 06902, Sophia-Antipolis Cedex, France.
†Department of mathematics, University of Colorado at Denver, Denver CO 80217-3364 and University of west Bohemia, Plzeň, Czech Republic

same efficiency as their structured grid counterparts. However, these geometrical methods are plagued by the necessity to handle many different meshes of the same geometry. Recent progress in mesh generation techniques brought improvements in automating the mesh coarsening operation [8] [4] [7]. However, this way of coarsening a three-dimensional mesh remains an extremely difficult task and one may wonder if this problem is not in fact at least as complex as solving a PDE on the fine level grid.

The second group of methods tries to avoid all the geometrical complexities asssociated with the concurrent handling of a hierarchy of meshes and to base the MG procedure on purely algebraic concepts. A well-known example of this type of method is the Algebraic Multigrid Method (AMG) [19]. Algebraic concepts are also used in aggregation methods [5]. These methods are multilevel techniques that have been used in various areas, including economical sciences. Recently aggregation methods have been improved by the introduction of the concept of smoothed aggregation [22]. For problems arising from the discretization of elliptic partial differential equations, this last approach has proved to compete very successfully with geometrical MG methods.

Closely related to algebraic methods, the volume agglomeration MG method was originally introduced by Lallemand and Dervieux [17]. This method was designed for first-order differential equations and finite-volume discretisations where they have demonstrated efficiency comparable to those of the best geometrical methods [18][21][24]. However, attempts to extend the volume agglomeration MG method to elliptic and mixed hyperbolic-elliptic problems have met a mitigated success. Interpreting the volume agglomeration method as a Galerkin procedure, the origin of the difficulty is seen to lie in the poor stability properties of the coarse spaces associated with the transfer operators used in this method [11] [10]. Up to now, the fixes proposed to overcome this difficulty have not been entirely successful : Based on heuristic considerations, early works on aggregation methods have proposed to overcorrect the correction step by a scaling factor [1]. This recipe has also been rediscovered in [11] and [3]. Mavriplis and Venkatakrishnan [14] have reported numerical experiments using prolongations based on AMG concepts. Although good convergence rates were obtained, the procedure seems to be too costly to be practical. In [10] the use of a reconstruction (or smoothing) operator to improve the accuracy of the agglomeration prolongator was advocated but this approach was not preserving the algorithm complexity.

In this paper, we study the extension of the concept of smoothed aggregation to convection-diffusion equations. We show how the idea of smoothed aggregation can help to extend to elliptic-hyperbolic equations the volume agglomeration MG method by allowing the use of more efficient transfer operators. However due to the very different properties of the hyperbolic and elliptic matrices, we have to give up a pure Ritz-Galerkin approach and instead use different stategies to construct the prolongators for these matrices. More specifically,

while the elliptic part of the equations favors the use of a symmetric prolongator smoother and thus (see Section 4) of a mesh coarsening factor of 3, the hyperbolic part of the equations favors the use of the conventional coarsening factor of 2. To deal with this problem, we extend the idea of smoothed aggregation by allowing the use of non-symmetric prolongator smoothers.

The outline of this paper is as follows : First we introduce the continuous problem and its discretization, then in Section 3, we make precise the aggregation procedure we consider, describe the relationship of this technique to the volume agglomeration method and formalize some of its properties. Then in the next section, we motivate the concept of smoothed aggregation and make precise the relations that the aggregation procedure and prolongator smoother must satisfy to obtain an optimal algorithm. Section 5 describes the MG procedure for convection diffusion equations and finally, in Section 6 we present some numerical results.

2 Problem formulation and discretization

We consider solving the following partial differential equation :

$$-\nabla \cdot (\mu \nabla u) + div(\vec{v}u) = f \tag{1}$$

on a bounded domain $\Omega \subset I\!R^2$ with suitable boundary conditions. The data $f \in L^2$, $\vec{v}$ is a constant field on $\bar{\Omega}$ and μ a positive constant. Assuming that Ω is a polygonal domain, the discretization of (1) that we consider uses a triangulation $\mathcal{T}_h$ of Ω composed of simplicial elements. We assume that $\mathcal{T}_h$ is regular and quasi-uniform so that a unique mesh parameter h suffices to characterize the triangulation. In case of dominant convection ($\| \vec{v} \| h >> \mu$), it is well-known that the finite element approximation of (1) suffers from stability problems. In this paper, to stabilize the finite element approximation, we choose to use a finite-volume approach using an upwind first-order flux function to compute the convective terms. More specifically for each node i of the triangulation, we define the associated control volume Ω_i as the polygonal cell whose submits are the middle of the edge connecting i to its neighbors and the center of gravity of the triangles having i as a node (see Figure 1).

Then we integrate (1) on Ω_i to obtain :

$$-\int_{\partial\Omega_i} \mu \nabla u.\vec{n} dl + \int_{\partial\Omega_i} \vec{v}.\vec{n} u dl = \int_{\Omega_i} f dx \tag{2}$$

where $\vec{n}$ is the exterior unit normal vector to Ω_i. Since ∇u is constant on any triangle, the evaluation of the first term is straightforward. The computation of the second integral is performed by splitting the boundary $\partial\Omega_i$ of cell Ω_i into a series of interfaces $\partial\Omega_{ij}$ between Ω_i and its neighboring cells Ω_j. Then

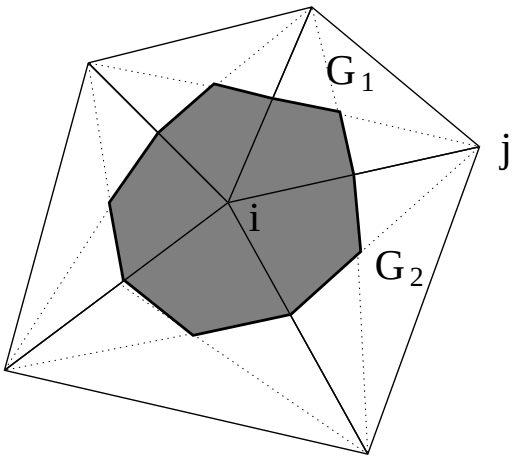

Figure 1: Control-volume used in the mixed finite element-finite volume method;

introducing a flux function $\Phi(u, v, \vec{n})$ the line integrals along each interface $\partial\Omega_{ij}$ are evaluated by :

$$\int_{\partial\Omega_{ij}} \vec{v}.\vec{n}u dl = \Phi(u_i, u_j, \vec{n}_{ij}) \tag{3}$$

where $\vec{n}_{ij} = \int_{\partial\Omega_{ij}} \vec{n} dl$. The flux function used in this work is :

$$\Phi(u, v, \vec{n}) = \frac{1}{2}[(\vec{v} \cdot \vec{n}u)_i + (\vec{v} \cdot \vec{n}u)_j - |\vec{v} \cdot \vec{n}|(u_j - u_i)]. \tag{4}$$

It can be shown that this finite volume method is totally equivalent to a P1 finite element method stabilized by a numerical dissipation term of the form :

$$-\sum_{j\in\kappa(i)} \frac{|\vec{v}.\vec{n}|}{2}(u_j - u_i)$$

where $\kappa(i)$ denote the set of neighbors of i (see e.g [9]). The main advantage of this approximation method is that the discretization of the convection terms corresponds to an M-matrix C^h. Actually, since $\sum_{j\in\kappa(i)} \vec{n}_{ij} = 0$, the discretization of the convective terms can be written as :

$$\frac{1}{2}\sum_{j\in\kappa(i)} (\vec{v} \cdot \vec{n}_{ij} - |\vec{v} \cdot \vec{n}_{ij}|)(u_j - u_i)$$

that obviously shows that $C^h_{ii} \geq 0$, $C^h_{ij} \leq 0$ and that :

$$\sum_{j\in\kappa(i)} |C^h_{ij}| \leq C^h_{ii}. \tag{5}$$

Thus C_h is a weakly diagonally dominant matrix with positive diagonal and negative off-diagonal. Since the strict inequality in (5) is reached on the boundary nodes corresponding to an inflow boundary ($\vec{v} \cdot \vec{n} < 0$), C^h is an M-matrix.

The main drawback of this approximation is that it is at best first-order accurate. The use of more accurate flux function can be advocated, however this would result in the loss of the M-matrix property of C^h. Thus, in practice, the use of more accurate discretization methods (usually of MUSCL type) for the approximation of the convective terms are combined with a Defect-Correction approach involving solving several linear systems defined with the first-order matrix C^h.

3 The aggregation/agglomeration technique

Given a linear algebraic problem $Ax = f$ where A is an $n \times n$ square matrix, a multigrid-type algorithm can be set-up in the following way : First, we set $l = 1, n_1 = n$ and $\mathcal{S}_1 = \{1, \ldots, n\}$ and we define a partition $\{\mathcal{A}_p^l; p = 1, \ldots, n_{l+1}\}$ of $\mathcal{S}_l$. The process is then repeated by setting $\mathcal{S}_{l+1} = \{1, \ldots, n_{l+1}\}$ and defining a new partition of this set until n_{l+1} is sufficiently small. The construction of the aggregates $\mathcal{A}_p^l$ is a crucial step of the algorithm. It can be performed purely algebraically by inspection of the coefficients of the matrix A. Such an algorithm is for instance described in [22]. However, for problems whose discretizations rely on a nearest neighbor stencil, the graph of the matrix is equivalent to the graph of the grid, therefore the definition of the partitions can be performed with any mesh partitionning algorithm. This is the approach that we have followed in this work where the recursive inertia algorithm has been used to define the partitions. The recursive inertia algorithm is a well-known technique used in domain decomposition methods, a short description of this algorithm is given in Appendix 1.
Once the partitions of the set $\{1, \ldots, n\}$ have been obtained, the coarse grid problems are created by a simple equation summing technique : First all the equations related to a degree of freedom i belonging to the same aggregate $\mathcal{A}_p^l$ are summed, then we assume that all the degrees of freedom that belong to the same aggregate have the same value. Starting from a $n_l \times n_l$ problem of the type :

$$A^l x^l = f^l \quad \text{where} \quad x_l, f_l \in I\!R^{n_l},$$

this process creates a $n_{l+1} \times n_{l+1}$ problem :

$$A^{l+1} x^{l+1} = f^{l+1} \;\; ; \;\; x_{l+1}, f_{l+1} \in I\!R^{n_{l+1}}$$

where the entries of the matrix A^{l+1} are defined by :

$$A_{p,q}^{l+1} = \sum_{i \in \mathcal{A}_p^l} \sum_{j \in \mathcal{A}_q^l} A_{i,j}^l$$

and where the right-hand side is given by :

$$f_p^{l+1} = \sum_{i \in \mathcal{A}_p^l} f_i^l$$

It is easy to see that this procedure is actually a Galerkin technique where A^{l+1} is defined by :

$$A^{l+1} = (\bar{P}^l)^t A^l \bar{P}^l$$

with a prolongation matrix $\bar{P}^l$ defined by :

$$\bar{P}_{ip}^l = \begin{cases} 1 & \text{if } i \in \mathcal{A}_p \\ 0 & otherwise \end{cases} \tag{6}$$

Now, let us define a subset V_1 of H^1 by a finite-element basis $\{\phi_1^1, \ldots, \phi_n^1\}$. If on the fine level, any vector $x \in I\!R^n$ is associated with a function of $V_1 \subset H^1(\Omega)$ by the finite element interpolator :

$$\pi_h : x \to u_h = \sum_{i=1}^{n} x_i \varphi_i^1, \tag{7}$$

then the coarse spaces $\{V_l; l = 2, \ldots, L\}$ generated by the aggregation procedure are spanned by the coarse grid basis function $\varphi_p^l; p = 1, \ldots, n_l$ recursively defined by :

$$\varphi_p^l = \sum_{j \in \mathcal{A}_p^{l-1}} \varphi_j^{l-1} = \sum_{j=1}^{n_{l-1}} (\bar{P}^l)_{pj}^t \varphi_j^{l-1} \tag{8}$$

Among the properties that the simple aggregation prolongation possesses, an extremely important one is that the Galerkin construction of the coarse grid matrix approximately preserves the number of non-zero entries per lines of the original matrix $A_1 := A$.

To see that, let us introduce the undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ associated to a given $n \times n$ square matrix A where $\mathcal{V} = \{1, \ldots, n\}$ is the set of nodes of the graph and $\mathcal{E}$ the set of edges defined by :

$$(i, j) \in \mathcal{E} \quad \text{iff} \quad a_{ij} \neq 0$$

We can define the A-distance : dist_A between two nodes (i, j) of the graph $\mathcal{G}$ as the minimal number of edges in a path connecting i to j. This definition can be trivially extended to the case of two sets $\mathcal{A}, \mathcal{B} \subset \mathcal{V}$ by :

$$\text{dist}_A(\mathcal{A}, \mathcal{B}) = \min_{i \in \mathcal{A}, j \in \mathcal{B}} \text{dist}_A(i, j).$$

Now let $\{\mathcal{A}_p\}_{p=1}^P$ be a partition of $\{1, \ldots, n\}$ and let B be a $P \times P$ matrix with $P < n$ we introduce the notion of **closest neighbor stencil** by :

Definition 3.1 *B has a closest neighbor stencil with respect to the matrix A if* $\forall (p,q) \in \{1,\ldots,P\}^2$*:*

$$dist_A(\mathcal{A}_p, \mathcal{A}_q) > 1 \Longrightarrow (B)_{pq} = 0$$

We then say that p is a direct neighbor of q if $\mathrm{dist}_A(\mathcal{A}_p, \mathcal{A}_q) = 1$.

It is easy to see that the Galerkin construction of the coarse grid matrices $\{A_l\}_{l=2}^L$ with the aggregation prolongation defined by (6) results in matrices that have this property :

Lemma 3.1 *Let $A^{l+1} = (\bar{P})^t A^l \bar{P}$ then A^{l+1} has a closest neighbor stencil with respect to A^l.*

Proof: This follows from $(A^{l+1})_{pq} = \sum_{j\in\mathcal{A}_p} \sum_{k\in\mathcal{A}_q} A^l_{jk}$ and the fact that $\mathrm{dist}_{A_l}(\mathcal{A}_p, \mathcal{A}_q) > 1 \Rightarrow \forall (j,k) \in \mathcal{A}_p \times \mathcal{A}_q \quad \mathrm{dist}_{A_l}(j,k) > 1$. As by definition $\mathrm{dist}_{A_l}(j,k) > 1 \Rightarrow A^l_{jk} = 0$ we get the result. *Q.E.D.*

An obvious consequence of the previous lemma is that the maximum number of non-zero entries per line of any matrix A_l in the coarse grid hierarchy $\{A_1,\ldots,A_L\}$ is stricly equal to the maximum number of direct neighbors of any node belonging to $\mathcal{V}_l$. Assuming that the partition algorithm keeps this number approximately constant, we can deduce that the number of non-zero entries per line is constant through the whole coarse matrix hierarchy.

Now, let us comment on the relationship of this approach with a finite volume re-discretization on a coarse mesh. This approach is intensively used in Computational Fluid Dynamics where it is known as agglomeration multi-grid methods [17, 18, 24] on non-strutured meshes while the term cell-centered multigrid method is used on structured meshes [20]. The idea of agglomeration algorithms is to fuse, or agglomerate, neighboring fine grid control volumes to form a reduced set of larger polygonal control volumes, and to re-discretize the equations on these control volumes. To see the relationship of this approach to the aggregation based algebraic MG method, let us consider the finite volume discretization of the equation :

$$div\vec{F} = f \tag{9}$$

with suitable boundary conditions. In (9) $\vec{F}$ is a smooth (possibly non-linear) function of the unknown u. Now, given a tiling of the computational domain by non-overlapping control volumes $\{\Omega_i; i = 1,\ldots,n\}$, the discretization of (9) is performed by integration of (9) on the computational cell Ω_i resulting in the equation :

$$\sum_{j\in\kappa(i)} \int_{\partial\Omega_{ij}} \vec{F}\cdot\vec{n}dl = \int_{\Omega_i} f dx, \tag{10}$$

where $\kappa(i)$ denote the set of all control volumes that have a common boundary with cell Ω_i while $\partial\Omega_{ij}$ stands for the common interface to cells i and j and $\vec{n}$ is the exterior unit normal vector on $\partial\Omega_{ij}$. In (10) the approximation of the line integral is done by the introduction of a flux function $\Phi(u, v, \vec{n})$ such that :

$$\int_{\partial\Omega_{ij}} \vec{F} \cdot \vec{n} dl \sim \Phi(u_i, u_j, \vec{N}_{ij}),$$

where $\vec{N}_{ij} = \int_{\partial\Omega_{ij}} \vec{n} dl$. Depending on the problem to be solved many different flux function can be used, however they all share the important conservation property :

$$\Phi(u, v, -\vec{n}) = -\Phi(v, u, \vec{n}) \tag{11}$$

that simply means that what is lost/gained by one cell is gained/lost by the other. The coarse grid problem defined by the aggregation procedure is given by :

$$\sum_{i \in \mathcal{A}_p} \sum_{j \in \kappa(i)} \Phi(u_i, u_j, \vec{N}_{ij}) = \sum_{i \in \mathcal{A}_p} \int_{\Omega_i} f dx. \tag{12}$$

We can write the sum $\sum_{i \in \mathcal{A}_p} \sum_{j \in \kappa(i)} \Phi(u_i, u_j, \vec{N}_{ij})$ as :

$$\sum_{i \in \mathcal{A}_p} \sum_{j \in \kappa(i) \cap \mathcal{A}_p} \Phi(u_i, u_j, \vec{N}_{ij}) + \sum_{i \in \mathcal{A}_p} \sum_{j \in \kappa(i) j \cap \bar{\mathcal{A}}_p} \Phi(u_i, u_j, \vec{N}_{ij})$$

where $\bar{\mathcal{A}}_p = \{1, \ldots, n\}/\mathcal{A}_p$ and the first sum cancels out due to the conservation property (11). Defining now $\kappa(p) = \{q; \exists j \in \mathcal{A}_q \text{ such that } j \in \kappa(i) \text{ for } i \in \mathcal{A}_p\}$, equation (12) can be rewritten as :

$$\sum_{q \in \kappa(p)} \sum_{i \in \mathcal{A}_p} \sum_{j \in \mathcal{A}_q \cap \kappa(i)} \Phi(u_p, u_q, \vec{N}_{ij}) = \sum_{i \in \mathcal{A}_p} \int_{\Omega_i} f dx. \tag{13}$$

Assume now that the flux function is linear with respect to its third argument, then (13) takes the form :

$$\sum_{q \in \kappa(p)} \Phi(u_p, u_q, \sum_{i \in \mathcal{A}_p} \sum_{j \in \mathcal{A}_q \cap \kappa(i)} \vec{N}_{ij}) = \sum_{i \in \mathcal{A}_p} \int_{\Omega_i} f dx. \tag{14}$$

Now, given the system of aggregates $\{\mathcal{A}_p; p = 1, \ldots, P\}$, we can associate to each aggregate $\mathcal{A}_p$, the geometrical *agglomerate*

$$\Omega_p = \cup_{i \in \mathcal{A}_p} \Omega_i$$

and it is readily seen that $\vec{N}_{pq} = \sum_{i \in \mathcal{A}_p} \sum_{j \in \mathcal{A}_q \cap \kappa(i)} \vec{N}_{ij}$ is precisely the line integral along the interface $\Omega_p \cap \Omega_q$ of the unit normal vector. Hence (14) can be written as :

$$\sum_{q \in \kappa(p)} \Phi(u_p, u_q, \vec{N}_{pq}) = \int_{\Omega_p} f dx \tag{15}$$

and thus appears as a finite volume discretization of (9) using the set of geometrical agglomerates Ω_p as computational cells.

The previous discussion shows that the identity between the volume agglomeration method and the algebraic aggregation method relies on the fact that the flux function is linear. This will be the case, for instance if a centered flux :

$$\Phi(u, v, \vec{n}) = \frac{1}{2}(\vec{v} \cdot \vec{n}(u + v))$$

is used, however, many numerical flux function do not satisfy this property. For instance, the flux function used in Section 2 is given by :

$$\Phi(u, v, \vec{n}) = \frac{1}{2}(\vec{v} \cdot \vec{n}(u + v) - |\vec{v} \cdot \vec{n}|(v - u))$$

and is not linear due to the appeareance of the absolute value of the normal velocity in the second term.

We also note that the simple aggregation prolongation possesses the following weak approximation property : If the fine level basis functions $\{\phi_1^1, \ldots, \phi_n^1\}$ are P1 or Q1 finite element basis functions, then the coarse grid spaces V_l defined by (8) satisfy :

Lemma 3.2 *There exists a sequence of mappings $Q_l : V_1 \to V_l$ with $Q_1 = Id$ and a constant C_1 independent of l such that :*

$$\forall u \in V_1, \| u - Q_l u \|_{L^2} \leq C_1 h_l |u|_{H^1} \tag{16}$$

where h_l is defined as the maximun diameter of the geometrical agglomerates Ω_i^l on level l defined recursively by : $\Omega_i^l = \cup_{k \in \mathcal{A}_i^l} \Omega_k^{l-1}$ where Ω_k^1 are the control cells defined in section 2 (figure 1).

Proof: This will follow from the fact that the prolongation $\bar{P}$ reproduces locally the constants. For a detailed proof see [12]. *Q.E.D.*

4 The concept of smoothed aggregation

In this section, we consider the purely diffusive case ($\vec{v} \equiv 0$) with the stiffness matrix $A^1 = \{A_{ij}^1 \equiv a(\phi_i^1, \phi_j^1)\}$, where $a(\cdot, \cdot)$ is an coercive and bounded bilinear form on H^1. Our aim here is to substantiate the concept of smoothed agglomeration introduced in [22]. First we recall that the efficiency of multigrid methods stems from two properties :
i) In contrast to other iterative methods, MG algorithms have a convergence rate that does not depend on the number of grid points.
ii) The cost of a multigrid cycle is proportional to the number of grid points.

With respect to the first point, we observe that assuming a standard "smoothing property" (satisfied by the usual smoothers like Jacobi or Gauss-Seidel Relaxations), the weak approximation property (16) is sufficient to ensure that the convergence rate is almost independent of the number of unknows (the dependence is only via a logarithmic factor) *provided the mappings Q_l satisfy the following stability property* :

Stability property : There exists a constant C_2 such that for any level l and any $u \in V_1$:

$$|Q_l u|_{H^1} \leq C_2 |u|_{H^1} \tag{17}$$

see [2] and [25].

Unfortunately, the property (17) is **not** satisfied by the aggregation procedure. A simple example is sufficient to understand why : Let $\{\Omega_i^1\}$ be a regular tiling of the plane by square cells of lengthside h and let us associate to this tiling the set V^1 spanned by Q^1 basis functions defined at the vertices of this tiling. Now consider the aggregation process defined by recursively fusing together 4 cells resulting in tilings $\{\Omega_i^l\}$ defined by squares of lengthside $2^{l-1}h$ as indicated in Figure 2.

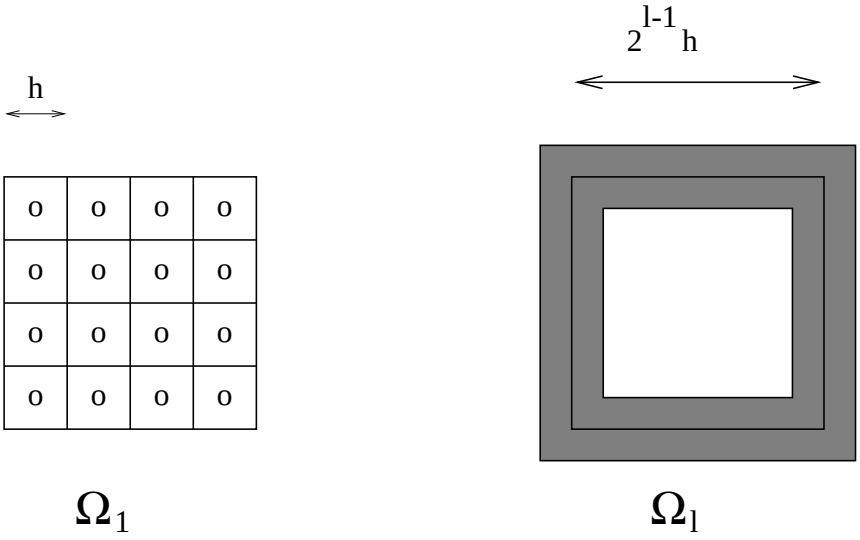

Figure 2: Coarsening process of a regular square mesh;

Since the basis functions ϕ_j^l spanning the coarse space V^l are simply a sum of the basis functions defined on the geometrical agglomerate $\{\Omega_j^l\}$, on any level l these basis functions are nonconstant only on a strip of elements as shown at

Figure 2. A crude estimate of the energy of these basis functions thus give :

$$|\phi_j^l|^2_{H^1} = \int_\Omega \nabla\phi_j^l.\nabla\phi_j^l \quad \sim (\tfrac{1}{h})^2 \times \text{grey area}$$
$$= (\tfrac{1}{h})^2 \times [((2^{l-1}+1)h)^2 - (2^{l-1}-1)h)^2] = 2^{l+1}$$

and there is no chance to find mappings Q_l that can satisfy (17) with a constant C_2 independent of l.

The aim of the smoothed aggregation procedure [22] is precisely to change the aggregation operator $\bar{P}$ into an improved one P that will now satisfy the stability property (17). However this improvment must be carefully done in order *to preserve the closest neighbor property* of the coarse grid matrices since this property is necessary to fulfill the requirement *ii)* and to give an algorithm of optimal complexity.

This will be done in the following way : Since the coarse grid basis functions $\varphi_p^{l+1} = \sum_{j\in\mathcal{A}_p} \varphi_j^l$ that result from the plain aggregation have too large an energy, it is natural to look for improved basis functions by *smoothing* the elements of the coarse space V_{l+1}. To this end, let us introduce $n_l \times n_l$ "smoothing matrices" S_l and define now the improved prolongation P^l by :

$$P^l = S_l \bar{P}^l \tag{18}$$

We want that the matrix S_l owns the following properties :

i) P^l still reproduce locally the constants (i.e the new basis function ϕ_i^l still decompose the unity : $\sum_{i=1}^{n_l} \phi_i^l = 1$).

ii) The energy of the basis functions for any level l is bounded independently of l

iii) The coarse grid matrices $A^{l+1} = (\bar{P}^l)^t S_l A^l S_l \bar{P}^l$ still possess a closest neighbor stencil.

Consider first condition *i)*. Let π_h be the finite element interpolator (7). On any level $l \geq 1$, the i–th basis function is defined by :

$$\phi_i^l = \pi_h S_1 \bar{P}^1 \ldots S_{l-1}\bar{P}^{l-1} e_l^i \tag{19}$$

where e_l^i is the i–th canonical basis vector of $\mathbb{R}^{n_l}$ and the decomposition of unity : $\sum_{i=1}^{n_l} \phi_i^l = 1$ will follow from (19) if we have :

$$S_{l-1}\bar{P}^{l-1}\mathbb{1}_l = \mathbb{1}_{l-1} \quad \forall l$$

where $\mathbb{1}_l$ is the n_l–vector $(1, 1, \ldots, 1)$. Since the aggregation prolongator $\bar{P}^{l-1}$ obviously satisfies :

$$\bar{P}^{l-1}\mathbb{1}_l = \mathbb{1}_{l-1}$$

condition *i)* is thus equivalent to the requirement that :

$$S_{l-1}\mathbb{1}_{l-1} = \mathbb{1}_{l-1} \quad \forall l \tag{20}$$

Assume that (20) holds for levels $1, \ldots, l-1$, then denoting by $< ., . >_l$ the Euclidian inner product in $\mathbb{R}^{n_l}$, we have :

$$\begin{aligned} < A^l \mathbb{1}_l, \mathbb{1}_l >_l &=< A^{l-1} S_{l-1} \bar{P}^{l-1} \mathbb{1}_l, S_{l-1} \bar{P}^{l-1} \mathbb{1}_l >_{l-1} \\ &=< A^{l-1} \mathbb{1}_{l-1}, \mathbb{1}_{l-1} >_{l-1} = \ldots \\ &=< A^1 \mathbb{1}_1, \mathbb{1}_1 >_1 = |\sum_{i=1}^{n_1} \phi_i^1|^2_{H^1} = 0 \end{aligned}$$

Therefore $\mathbb{1}_l \in \text{Ker}(A^l)$ and to respect (20) S_l can be constructed as a polynomial p in the matrix A^l such that $p(0) = 1$.

Let us focuss on the property *iii)*. In view of the discussion in Section 5, we will consider a slight extension of the definition of the coarse grid matrices : *i.e.* we will allow the presence of two different smoothing polynomials p_L and p_R not necessarily equal and will define the coarse grid matrices as :

$$A^{l+1} = (\bar{P}^l)^t p_L(A^l) A^l p_R(A^l) \bar{P}^l$$

We then have the result :

Proposition 4.1 *Let $k = deg(p_L) + deg(p_R)$, if the system of agglomerates $\mathcal{A}_p$ satisfies :*

$$dist_{A^l}(\mathcal{A}_p, \mathcal{A}_q) > 1 \Longrightarrow dist_{A^l}(\mathcal{A}_p, \mathcal{A}_q) \geq k+2 \tag{21}$$

then $A^{l+1} = (\bar{P}^l)^t p_L(A^l) A^l p_R(A^l) \bar{P}^l$ has a closest neighbor stencil with respect to A^l.

Proof : Let $(p,q) \in \{1, \ldots, n_{l+1}\}^2$ such that $\text{dist}_{A_l}(\mathcal{A}_p, \mathcal{A}_q) > 1$ and consider the entry $A^{l+1}_{p,q}$. We have :

$$(A^{l+1})_{pq} = \sum_{i \in \mathcal{A}_p} \sum_{j \in \mathcal{A}_q} [A^l p_L(A^l) p_R(A^l)]_{ij}$$

We will prove that all the terms in this sum are zero. For this, observe that there exist some real coefficients $\{\alpha_r\}_{r=1}^{k+1}$ such that :

$$A^l p_L(A^l) p_R(A^l) = \sum_{r=1}^{k+1} \alpha_r (A^l)^r$$

and thus $[A^l p_L(A^l) p_R(A^l)]_{ij} = \sum_{r=1}^{k+1} \alpha_r [(A^l)^r]_{ij}$. We now remark that any term in the above sum can be written as :

$$[(A^l)^r]_{ij} = \sum_{s_1=1}^{n_l} (A^l)_{is_1} \sum_{s_2=1}^{n_l} (A^l)_{s_1 s_2} \sum_{s_3=1}^{n_l} \cdots \sum_{s_{r-1}=1}^{n_l} (A^l)_{s_{r-2} s_{r-1}} (A^l)_{s_{r-1} j}$$

but all the products $(A^l)_{is_1} (A^l)_{s_1 s_2} \ldots (A^l)_{s_{r-2} s_{r-1}} (A^l)_{s_{r-1} j}$ are zero, for the existence of $r-1$ indices $s_1, s_2, \ldots, s_{r-1}$ such that

$$(A^l)_{is_1} (A^l)_{s_1 s_2} \ldots (A^l)_{s_{r-2} s_{r-1}} (A^l)_{s_{r-1} j} \neq 0$$

would imply that the A^l distance between i and j is less or equal to r.
But this is impossible since $r \leq k+1$ while by assumption (21) $\mathrm{dist}_{A^l}(\mathcal{A}_p, \mathcal{A}_q) \geq k+2$.

Remark 4.1 *Intuitively, the larger the degree of the polynomial is, the better smoothing effect can be expected. On the other hand, the previous result shows that to preserve the optimality of the algorithm, the graph distance between non-neighboring aggregates must increase with the degree of the polynomial. In other words, the ratio of the number of coarse and fine level degrees of freedom must decrease with the degree of the polynomials.*

Let k be the total degree of the prolongator smoothers. Consider the aggregate $\mathcal{A}_p$ and let $\mathcal{A}_q$ be the closest aggregate to $\mathcal{A}_p$ that does not belong to the direct neighbors of $\mathcal{A}_p$. Then there is a common neighbor $\mathcal{A}_s$ of the aggregates $\mathcal{A}_p$, $\mathcal{A}_q$ such that the shortest path $\mathcal{P}$ connecting $\mathcal{A}_p$ with $\mathcal{A}_q$ is of the form $i_1, \ldots, i_r$, where $i_1 \in \mathcal{A}_p, i_r \in \mathcal{A}_q$ and $i_j \in \mathcal{A}_s$ $\forall i = 2, \ldots, r-1$. To prevent the coarse-level entry A^{l+1}_{pq} from being nonzero, we need $r - 1 \equiv dist_{A^l}(\mathcal{A}_p, \mathcal{A}_q) \geq k+2$, which is possible only if the path $\mathcal{P}$ passes through at least $k+1$ nodes of the aggregate $\mathcal{A}_s$. Roughly speaking, to guarantee that A^{l+1} corresponds to a closest neighbor stencil when using prolongator smoothers of total degree k, we need the aggregates to be at least $k+1$ nodes "wide". When coarsening a regular mesh, this means that we have to aggregate at least $k+1$ nodes in each spatial direction, leading to aggregates consisting of at least $(k+1)^d$ nodes. For this reason, we say that $k+1$ is the the coarsening factor that must be associated to prolongator smoothers of total degree k and in the sequel, we will assume that the use of a polynomial smoother of total degree k is associated with a coarsening strategy that results in the construction of aggregates consisting of approximately $(k+1)^d$ nodes.

We now concentrate on requirement *ii)*. Let ϕ^l_i be a basis function of V^l. Since the fine grid matrix A^1 corresponds to H^1–equivalent form,

$$\begin{aligned} |\phi^l_i|^2_{H^1} \sim < A^l e^i_l, e^i_l >_l = & < S^2_{l-1} A^{l-1} \bar{P}^{l-1} e^i_l, \bar{P}^{l-1} e^i_l >_l \\ & \leq \rho(S^2_{l-1} A^{l-1}) \parallel \bar{P}^{l-1} e^i_l \parallel^2_l, \end{aligned}$$

where $\parallel \cdot \parallel_l = < \cdot, \cdot >^{1/2}_l$. Now observe that $\parallel \bar{P}^{l-1} e^i_l \parallel^2_l$ is the number of degrees of freedom in the i–th aggregate on level l. In view of Remark 4.1, if $\deg S^2_{l-1} = k$, the number of degrees of freedom per aggregate must be equal to $(k+1)^2$ in order to preserve the closest neighbor property of the coarse matrices. Summing up the above considerations, to assure the uniform boundedness

$$|\phi^l_i|_{H^1} \leq C,$$

we need polynomials smoothers $p(A^{l-1})^2$ of degree k satifying

$$\rho(S^2_{l-1} A^{l-1}) \leq \frac{\rho(A^{l-1})}{(k+1)^2}, \quad p(0) = 1$$

or equivalently, we need a polynomial $p(x)$ such that:

$$\max_{x\in[0,\bar{\rho}(A^{l-1})]} xp^2(x) \leq \frac{\bar{\rho}(A^{l-1})}{(k+1)^2}$$

where $\bar{\rho}(A^{l-1})$ is an available upper bound of $\rho(A^{l-1})$.

Once again, in view of the forthcoming discussion of Section 5, we will consider a slightly more general case and will allow two different smoothing polynomials p_L and p_R. Denoting $q(x)$ the product $p_L(x)p_R(x)$, we have the following result that is a slight generalization of a result due to Jan Mandel :

Proposition 4.2 *For any integer $k > 0$ there is a unique polynomial q of degree k satisfying $q(0) = 1$ such that*

$$\max\{xq(x); x \in [0,1]\} = \frac{1}{(k+1)^2} \tag{22}$$

Moreover this polynomial satisfies :

$$0 \leq q(x) \leq 1$$

and if $k = 2r$ is even, there exists a polynomial $p(x)$ of degree r satisfying $p^2(x) = q(x)$.

Proof. We look for $h(x) = xq(x)$ as the scaled and transformed Čebyšev polynomial :

$$h(x) = \frac{c}{2}[1 - T_{k+1}(1-2x)]$$

where $T_{k+1}(x) = \cos((k+1)\mathrm{Arccos}(x))$ is the $k+1$–th Čebyšev polynomial of degree $k+1$. The maxima of $h(x)$ are equal to c and are reached at the points $x_j = (1 - \cos(j\pi/k+1))/2$. Thus in order to satisfy (22) we set :

$$c = \frac{1}{(k+1)^2}$$

To prove that $h(x)$ is of the form $xq(x)$ with $q(0) = 1$, we observe first that obviously $h(0) = 0$. Then computing $h'(x)$ we get $h'(x) = cT'_{k+1}(1-2x)$.
A simple computation then shows that precisely $T'_{k+1}(1) = (k+1)^2$ proving that $h(x)/x = 1$ for $x = 0$.

If $k = 2r$, $h(x)$ has $2r+1$ roots

$$x_j = \frac{1}{2}(1 - \cos\frac{2j\pi}{2r+1}) \qquad j = 0, \ldots, r.$$

The value $j = 0$ gives the simple root $x_j = 0$ while the values $j = 1, \ldots, r$ correspond to double roots. Therefore

$$q(x) = \frac{h(x)}{x} = (1 - \frac{x}{x_1})^2 \ldots (1 - \frac{x}{x_r})^2$$

is indeed of the form $p(x)^2$.

If on the contrary k is odd $k = 2r + 1$, $h(x)$ has $2r + 2$ distinct simple roots given by :

$$x_j = \frac{1}{2}(1 - \cos\frac{j\pi}{r+1}) \quad j = 0, \ldots, 2r+1$$

The value $j = 0$ gives the simple root $x_0 = 0$ while $j = r + 1$ gives $x_{r+1} = 1$. The remaining roots are

$$x_j^{\pm} = \frac{1}{2}(1 \pm \cos\frac{j\pi}{r+1}) \quad j = 1, \ldots, r.$$

It remains to prove that $0 \leq q(x) \leq 1$. Using the expression

$$h(x) = \frac{1}{2T'_{k+1}(1)}[1 - T_{k+1}(1-2x)],$$

this is equivalent to prove that

$$0 \leq \frac{1 - T_{k+1}(1-2x)}{2T'_{k+1}(1)x} \leq 1 \quad \forall x \in [0,1].$$

Setting $y = 1 - 2x$ for $y \in [-1, 1]$, this becomes:

$$0 \leq \frac{1 - T_{k+1}(y)}{T'_{k+1}(1)(1-y)} \leq 1 \quad \forall y \in [-1,1].$$

Since $T_{k+1}(y) \in [-1, 1]$, the lower bound is obvious while the upper bound gives

$$\frac{1 - T_{k+1}(y)}{(1-y)} \leq T'_{k+1}(1)$$

that expresses the well-known fact that the graph of any Čebyšev polynomial lies above its tangent at $x = 1$. *Q.E.D.*

Remark 4.2 *For any coarsening strategy, there exists one and only one polynomial smoother that allows us to respect both the complexity of the algorithm (closest neighbor property) and the boundeness of the energy of the basis functions. For a coarsening factor of $k + 1$, the use of polynomial smoothers of degree $> k$ will result in the loss of the closest neighbor property of the coarse matrices while the use of a polynomial of degree $< k$ will be unable to control the growth of the energy of the basis functions. The fact that there exists* an unique *polynomial that can achieve these two objectives is quite remarkable.*

Remark 4.3 *For a true multigrid strategy, the only interesting cases correspond to a coarsening strategy by the usual factor 2 or by the factor 3 advocated in [22]. The use of larger coarsening factors will results in a extremely fast decrease of the number of degrees of freedom. Thus the use of large coarsening factors cannot be associated with the definition of a coarse grid hierarchy involving many grids. However, it is still possible to use large coarsening factors in a two-grid setting see [12].*

Remark 4.4 *The smallest coarsening factor is 2, according to the previous results, this coarsening factor has to be associated with a polynomial smoother of degree 1 given by :*

$$p(x) = 1 - x.$$

The smallest coarsening factor allowing the use of an even polynomial smoother such that $p_L = p_R$ is 3. In this case, the polynomial smoother must be of degree 2 and is given by :

$$p(x) = (1 - \frac{4x}{3})^2$$

5 The multigrid solver in the convection-diffusion case

We now return to the general case ($\vec{v} \not\equiv 0$) and consider a system of linear algebraic equations

$$Ax = (C + D)x = f, \tag{23}$$

where D is a symmetric positive definite matrix corresponding to the discretization of a second order H^1–elliptic operator and C a non-symmetric M–matrix obtained by discretization of a first-order L^2–antisymmetric operator. In the previous section, we have designed a prolongation operator suitable for the diffusive part of the algorithm.

Unfortunately, the properties of the matrices D and C are very different and this prolongator is not suitable for the convection matrice C. The main reason why the smoothed prolongator $S_l \bar{P}^l$ is unefficient to construct the coarse convection matrices has to deal with the *centered* character of the smoothing matrices S_l. While the original convection matrix has an *upwind* character, the product $S_l C_l S_l$ do not have this property since S_l is centered. This leads to a progressive loss of the M-matrix property of the convection matrices resulting in inaccurate coarse grid correction ([6]).

On the other hand, the plain aggregation approach preserves the M–matrix character of the coarse matrices and the fact that the matrix has a closest neighbor stencil. Moreover, although we are not aware of any available convergence theory for MG solutions of hyperbolic problems on unstructured meshes, it is widely believed that the MG solutions of hyperbolic systems do not require an approximation property stronger than (16). We also observe that the use of the aggregation restriction is consistent with the finite-volume philosophy used in the discretization : the integral of a function on an agglomerate is equal to the sum of the integrals defined on each pieces that compose this agglomerate. Therefore, for all these reasons, we adopt the point of view of using the plain aggregation technique for the convective matrices while the diffusive ones will be obtained by the smoothed agglomeration method.

In this respect, the question arises of the order of the polynomial smoother that defines the "elliptic" prolongator. As we have seen, the choice of the degree of this polynomial and of the coarsening factor are not independent. We expect that the use of symmetric polynomial smoother is more efficient for the elliptic part of the equations and this favors the use of a polynomial of degree 2 and of a coarsening factor of 3. However, the convective part of the equations is totally independent of the order of the polynomial smoother and the only relevant factor for these operators is simply the *size* of the coarse spaces. This would favor the use of a coarsening factor of 2 and consequently the use of a polynomial smoother of degree 1. The examination of these questions is our main motivation to extend the symmetric smoothed aggregation procedure of [22] by allowing the use of non-symmetric smoothers p_L and p_R.

With this in mind, the coarsening procedure will proceed as follows :
First, we specify full rank prolongation matrices $\bar{P}^l$ using the aggregation technique described in Section 3. These *tentative* prolongators will be used for the coarsening of the "convective parts" C_l in a standard variational way, i.e. $C_{l+1} = (\bar{P}^l)^T C_l \bar{P}^l$. The coarsening of the "diffusive" parts D_l will be done using the *smoothed aggregation* technique. We define the left and right square smoothing matrices S_l^L and S_l^R of the same order as the order of the matrix A_l and define recursively the coarse "diffusive" matrices by $D_{l+1} = (\bar{P}^l)^t S_l^L D_l S_l^R \bar{P}^l$.

Summing up, for given matrices $C_1 := C$, $D_1 := D$, we generate the components of the multigrid algorithm by the following procedure :

ALGORITHM 5.1 : Set-up phase

1. *Initialize* $l = 1$.
2. *Repeat*
   - *(a) construct the aggregation based* $n_l \times n_{l+1}$ *prolongator* $\bar{P}^l$, *where* n_l *is the order of* A_l *and* $n_{l+1} < n_l$ *(*n_{l+1} *is the result of this construction).*
   - *(b) construct the left and right* $n_l \times n_l$ *smoothing matrices* S_l^L *and* S_l^R,
   - *(c) create the coarse-level matrices by*
   $$\begin{aligned} D_{l+1} &= (\bar{P}^l)^T S_l^L D_l^L S_l^R \bar{P}^l, \quad C_{l+1} = (\bar{P}^l)^T C_l^L \bar{P}^l, \\ A_{l+1} &= C_{l+1} + D_{l+1}, \end{aligned}$$
   - *(d) setup the smoothers* $\mathcal{S}_l^{pre}$, $\mathcal{S}_l^{post}$ *and increment the level by* $l \leftarrow l+1$
3. *until* A_{l+1} *is sufficiently small to be treated efficiently by a direct solver.*

We denote L the number of levels obtained after completion of algorithm 5.1. Once the components of the multigrid algorithm have been created, the iterative solution of (23) can proceed by

$$x \leftarrow MG(x, f),$$

where $MG(.,.) \equiv MG_1(.,.)$ is defined by Algorithm 5.2 with $x_1 = x$ and $f_1 = f$.

ALGORITHM 5.2 : Solving phase
For the coarsest level $l = L$, we define

$$MG_l(x_l, f_l) = A_l^{-1} f_l.$$

For $l = 1, \ldots, L-1$, we set $MG_l(x_l, f_l) = z_l$, where z_l is obtained by

1. $x_l \leftarrow \mathcal{S}_l^{pre}(x_l, f_l)$,
2. *set* $f_{l+1} = (\bar{P}^l)^T S_l^L (A_l x_l - f_l)$; $x_{l+1} = 0$,
3. *repeat* $x_{l+1} \leftarrow MG_{l+1}(x_{l+1}, f_{l+1})$ γ *times, (γ is a given cycle parameter)*
4. $x_l \leftarrow x_l - S_l^R \bar{P}^l x_{l+1}$
5. $x \leftarrow \mathcal{S}_l^{post}(x_l, f_l)$,
6. *set* $z_l = x_l$.

6 Numerical results

Example 1: Our aim in this first example is to study the influence of the degree of the polynomial smoother on the performance of the method. We consider solving the Laplace equation on the square $[0,1] \times [0,1]$. The discretization is done on a square mesh of space size $h = 1/n$ using $P1$ finite elements. In this experiment, we use a W-cycle with 2-forward and backward Gauss-Seidel pre and post-smoothing per cycle. We define the average residual reduction per cycle as :

$$Arr = \left(\frac{\| b - Ax^m \|_{l^2}}{\| b \|_{l^2}} \right)^{1/m}$$

where m is the number of iterations necessary to reduce the relative residual by a factor 10^{-8}. Figure 3 displays the average residual reduction per cycle versus the number of grid points in one direction. In this figure, the curves labeled 2×2 indicate the use of a one-sided polynomial smoother of degree one (i.e $p_L = Id$ and $p_R = Id - \omega A^l$) associated with a coarsening factor of 2 while the curves labeled 3×3 denote the use of a symmetric polynomial smoother of degree 2 ($p_L = p_R = Id - \omega A^l$) associated with a coarsening by a factor 3. Both cases clearly show that the convergence is uniform with respect to the mesh size. We note that there is no significant difference between the two coarsening strategies and that approximately the same convergence factor is obtained by the use of a polynomial smoother of degree one or 2. We also show in this figure, the results obtained by applying an over-correction parameter to the coarse grid correction *i.e* the results obtained by modifying the coarse grid correction step of algorithm **5.2** into :

$$u_l \leftarrow u_l + \alpha_l S_l^T \bar{P}^l u_{l+1}$$

with $\alpha_l > 1$. This simple technique produces a clear improvment in the convergence factor. Justification for the use of an overcorrection parameter can be

found in [23]. In the present case, the value of the optimal parameter α_l have been found based on trial and error strategy. Another possibility is to determine this parameter by minimizing the energy norm of the error as done in [23].

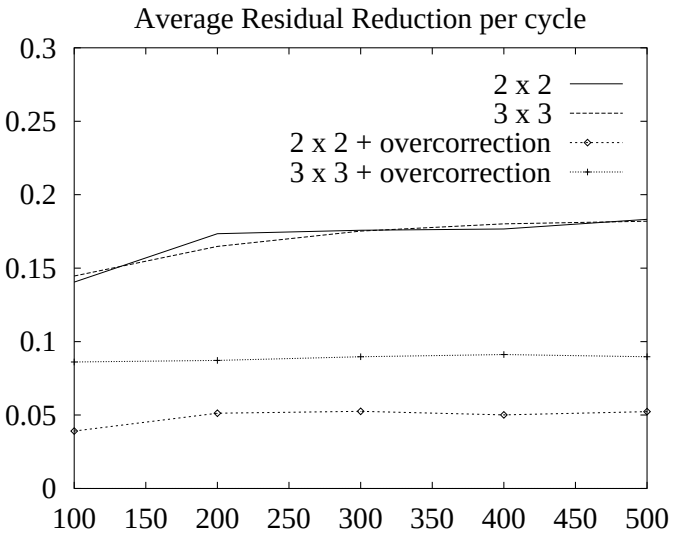

Figure 3: Average Residual Reduction versus space step;

Example 2: In the previous example, the agglomeration algorithm **5.2** produces perfect results and the size (and shape) of the agglomerates on all level is uniform within the meshes. On a truly unstructured mesh, these properties may be lost and it may be more difficult to control the shape regularity of the agglomerates as well as the sparsity of the coarse grid matrices. In the present example, we consider the approximation of the Laplace equation on the mesh depicted in Figure 4.

This mesh contains originally 3114 nodes. To study the behavior of the solver with respect to the mesh size, the mesh is refined by a regular subdivision of triangles. The larger mesh used in this example contain 194480 nodes. The average residual reduction factor versus the square root of the number of unknows is displayed in Figure 5. We use in this example, a V-cycle preconditionner with two forward and backward Gauss-Seidel relaxations in the pre and post smoothing steps and the outer acceleration is performed by the GMRES algorithm.

As in the previous example, it can be observed that the convergence properties of the solver are roughly mesh independent. In contrast to the previous case, the use of prolongator smoothers of degree 2 gives slighly better results than

pb_2d

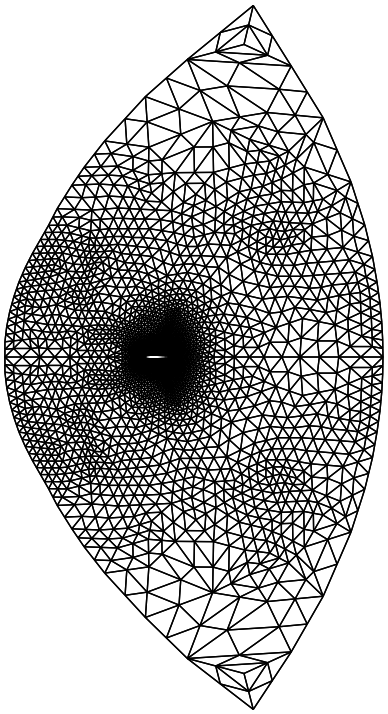

Figure 4: Unstructured mesh around an airfoil;

the use of a first degree polynomial. There may be two main reasons for this behavior : First that the efficiency of the smoother is better with a polynomial smoother of higher degree than with a simple linear function but also that the size and shape regularity of the agglomerates are better with a 3 x 3 coarsening than with a 2 x 2 coarsening. For instance, if one considers a regular tiling of the plane by a hexagonal mesh, it is easy to see that the coarsening by a factor 3 will result in a regular mesh while coarsening by a factor 2 will produce an irregular pattern. Although on unstructured meshes the number of neighbors of a given node is arbitrary, it can be observed that in general, the average number of neighbors per node is close to 6. Thus one can suspect that coarsening by a factor 3 will produce agglomerates of better shapes than with a coarsening factor equal to 2.

Example 3: We now turn to an analysis of the purely convective case obtained with $v = (1, 1)$ and $\kappa = 0$. Again a V-cycle preconditionner together with an outer acceleration performed by the GMRES method are used. The results are displayed in Figure 6.

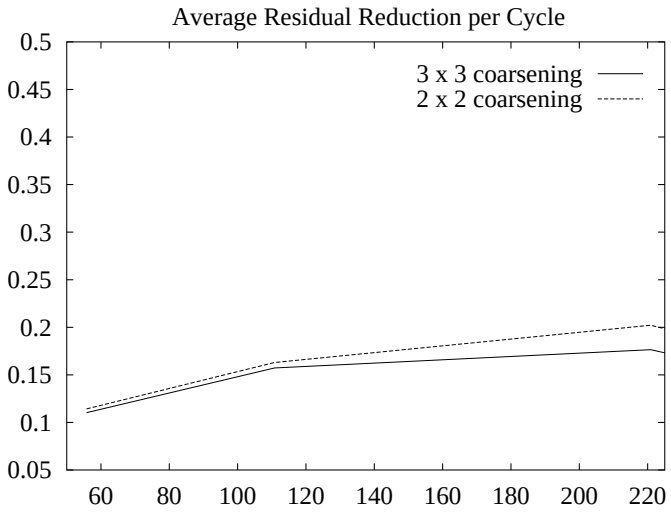

Figure 5: Average Residual Reduction versus space step;

Although, the slope of the curves are decreasing with the number of unknows, it is clear that a mesh independent regime has not yet been reached even with the largest mesh containing 194480 nodes. This result is somehow unexpected because there are recurrent assertions and experimental evidences in the CFD literature (e.g. [16, 17, 24]) to state that the volume agglomeration MG method produces mesh independent results. This seems to indicate that for purely hyperbolic problems, the variational framework is not an adequate one.
We also note that the results are significantely worse than in the purely diffusive case. In contrast with the purely diffusive case, coarsening by a factor 2 yields better convergence properties than a coarsening strategy by a factor 3. This can be explained very simply : Since for a purely convective equation, the coarse grid correction involve only the agglomeration operators, the most important factor for the efficiency of the coarse grid correction is simply the size of the coarse space. Roughly speaking, the larger the coarse space is, the better the coarse grid correction will be. Comparing the size of the coarse space for coarsening factor 2 and 3, we see that in 2-D the dimension of the coarse space produced by a coarsening factor of 2 is approximately twice the dimension of the coarse space obtained with a coarsening factor of 3. Thus, the fact that better convergence properties results from a coarsening factor of 2 is no surprise.

Example 4: The algebraic construction of the coarse grid matrices has been done both to preserve the "upwind" character of the convection matrix and to obtain acceptable approximation properties for the elliptic part. The

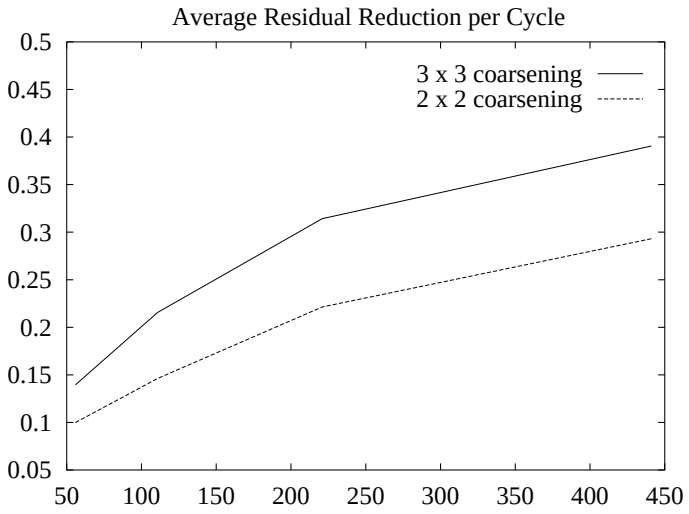

Figure 6: Average Residual Reduction versus space step;

previous numerical experiments have shown that the algebraic algorithm works reasonnably well in these two limiting cases. However, this does not guarantee that the algorithm will behave in an efficient manner away from these limiting cases. In this example, we study the behavior of algorithm **5.2** away from the purely convective or diffusive regime. The mesh and the algorithm are identical to the ones used in example 2. Figure 7 displays the behavior of the average residual reduction factor with respect to the Peclet number defined as :

$$Pe = \frac{\| v \| h}{\kappa}$$

where $h = \sqrt{\mathcal{A}/n}$ with $\mathcal{A}$ the area of the computational domain and n the total number of nodes.

This figure shows that for all the tested meshes and for the two different coarsening factors, the average residual reduction factor increases with the Peclet number until it reaches a maximum for a Peclet number between 1 and 2, then it decreases and for $Pe > 10$ it roughly stabilizes to the values obtained in the purely convective case. This clearly indicates that the difficulty of solving the system is maximum when the hyperbolic and diffusive parts balance each other. However, we observe that the growth of the convergence factor is quite moderate and that it does not destroy the good convergence properties of the solver. Figure 8 shows an enlargement of these results for small Peclet number. It is seen that even a moderate amount of convection (say $Pe \leq 0.25$) produces a deterioration of the results with respect to the purely diffusive case. However,

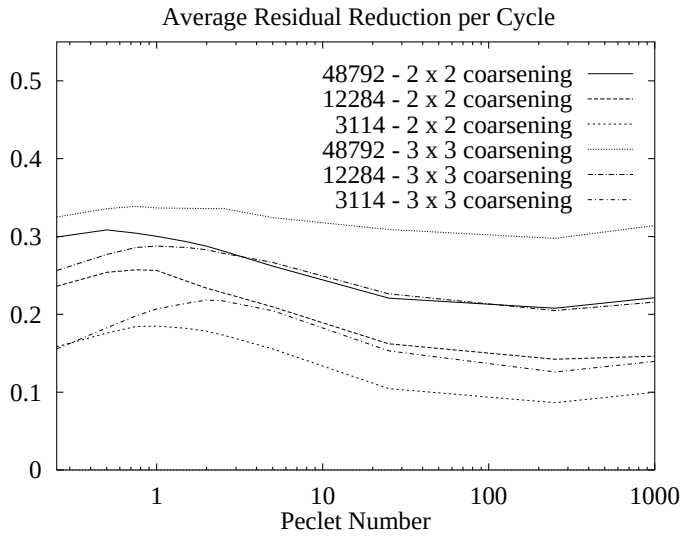

Figure 7: Average Residual Reduction versus Peclet number;

this deterioration is quite modest and subsequent increase of the Peclet number results in only a very slow increase of the convergence factor until a maximum is reached for a Peclet number between 1 and 2.

Appendix 1 : The recursive inertia algorithm

The recursive inertia algorithm is one of the simplest mesh partitionning algorithm. It is based on the simple idea of slicing the mesh into different pieces. Let $(x_1, y_1), \ldots, (x_n, y_n)$ be the vertices of a simplex mesh $\mathcal{T}_h$, we first compute the principal inertia axis of the mesh defined as the eigenvector corresponding to the leading eigenvalue of the inertia matrix defined by :

$$I = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

with :

$$\begin{aligned} I_{xx} &= \sum_{i=1}^{n} (y_i - y_{cg})^2 \\ I_{yx} &= I_{xy} = -\sum_{i=1}^{n} (x_i - x_{cg})(y_i - y_{cg}) \\ I_{yy} &= \sum_{i=1}^{n} (x_i - x_{cg})^2 \end{aligned}$$

where (x_{cg}, y_{cg}) denote the center of gravity of the set of vertices of the mesh. Then we project the vertices ω_i on this axis, sort them according to their coordinate on this axis and create several groups of vertices according to their projected coordinates on the inertia axis. This algorithm thus tends to create slices of mesh orthogonal to the principal inertia axis. The process is then repeated for each individual group of vertices until a sufficient number of groups have been created.

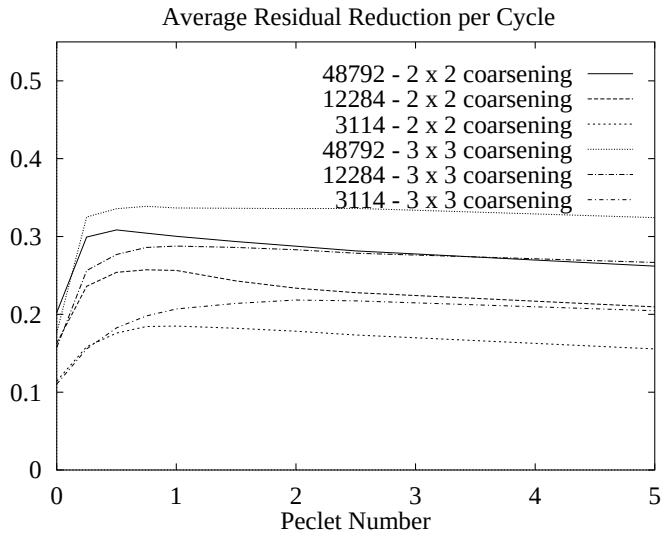

Figure 8: Average Residual Reduction versus Peclet number;

Acknowledgments

Part of this work have been performed while the first author was enjoying the hospitality of the Center for Aerospace structures of the University of Colorado in Boulder. We also thank Luc Fournier of Inria Sophia-Antipolis for allowing us to use the mesh partitionning algorithm that he has developped for his own studies in Volume Agglomeration MG methods.

References

[1] R. Blaheta, A mulilevel method with overcorrection by aggregation for solving discrete elliptic problems, J. Comput. Appl. Math., **24**, pp 227-239, 1988.

[2] J. H. Bramble, J.E. Pasciak, J. Wang and J. Xu, Convergence estimates for multigrid algorithms without regularity assumptions, Math. Comp., **57**, pp. 23-45, 1991.

[3] D. Braess, Towards algebraic multigrid for elliptic problems of second order, Computing, **55**, pp 379-393,1995.

[4] T. Chan, B. Smith, Domain decomposition and multigrid algorithms for elliptic problems unstructured meshes, Electronic Transactions in Numerical Analysis, **2**, pp 171-182, 1994.

[5] F. Chatelin and W.L. Miranker, Acceleration by aggregation of successive approximation methods, Linear Algebra with Applications, **43**, pp 17-47, 1982.

[6] P. M. de Zeeuw, "Matrix dependent prolongations and restrictions in a black-box multigrid solver, J. Comp. and Appl. Math., **33**, pp 1-27, 1990.

[7] R. Fournier, H. Guillard, "Node-nested multigrid in 3-D : Coarsening of 3-D surface meshes", in preparation.

[8] H. Guillard, "Node-nested multigrid method with Delaunay Coarsening", INRIA Rpt, No 1898, May 1993.

[9] H. Guillard, "Mixed Element Volume approach in Computational Fluid Dynamics", von Karman Institute Lecture Series 1995, March 13-17, 1995.

[10] H. Guillard and N. Marco, "Some aspects of multigrid methods on non-structured meshes", in *Proceedings of the 7th Copper Mountain Conference on Multigrid Methods*, NASA Conference Publication 3339, pp 347-362.

[11] B. Koobus, M.H. Lallemand, A. Dervieux, Unstructured Volume Agglomeration MG, Solution of the Poisson equation, Int. J. for Numerical Meth. in Fluid, **18**, pp 27-42, 1994.

[12] J. Křižkova and P. Vaněk, Two-level preconditionner with small coarse-grid appropriate for unstructured meshes, Numerical Linear Algebra with Applications, **3**, pp 255-274, 1996.

[13] D. Mavriplis, A. Jameson, L. Martinelli, Multi-grid solution of the Navier-Stokes equations on triangular meshes, ICASE Rpt No 89-11, 1989.

[14] D. Mavriplis and V. Venkatakrishnan, Agglomeration multigrid for viscous turbulent flows, AIAA paper 94-2332, June 94; to appear in Computers and Fluids.

[15] E. Morano, A. Dervieux, H. Guillard, M.P. Leclercq, B. Stoufflet "Faster Relaxations for non-structured MG with Voronoi coarsening" *in Proceedings of CFD'92,*,**vol I**,pp 69-74, (Hirsch et al. Eds), Elsevier 1992.

[16] E. Morano, A. Dervieux "Steady Relaxations Methods for Unstructured MG Euler and Navier-Stokes solutions" submitted

[17] M.H. Lallemand, A. Dervieux, A multigrid finite element method for solving the two-dimensional Euler equations, Proceedings of the third Copper Mountain conference on Multigrid methods, Lecture Notes in Pure and Applied mathematics, Ed S. F. Mc Cormick, Marcel Dekker Inc, April 87, pp 337-363.

[18] M.H. Lallemand, H. Steve, A. Dervieux, Unstructured multigridding by volume agglomeration : current status. Computers & Fluids, **21**, pp397-433, 1992.

[19] J. Ruge and K. Stuben, Algebraic Multigrid Methods in *Multigrid Methods* (S. Mc Cormick Ed), Frontiers in Applied Math, Vol 5, SIAM, Philadelphia.

[20] P. Wesseling, An introduction to Multigrid methods, Wiley, 1992.

[21] G. Carré, A. Dervieux, Unstructured multigrid for $k - \epsilon$ kernel, Workshop on efficient models for aeronautics, Manchester, November 11-14, 1994.

[22] P. Vaněk, J. Mandel and M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing, **56**, pp. 179-196, 1996.

[23] P. Vaněk, S. Míka, Analysis of the two-level algorithm with overcorrection, Appl. Math.,**37**, pp. 13–28, 1992.

[24] V. Venkatakrishnan, D.Mavriplis, Unstructured multigrid for the 3-D Euler equations, AIAA paper 94-0069, January 1994.

[25] J. Xu, Iterative methods by subspace decomposition and subspace correction, SIAM Review, **34**, pp. 581-613, 1992.