Node-nested multi-grid method with Delaunay coarsening

Hervé Guillard

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Rapports de Recherche

N°1898

Programme 6

Calcul scientifique, Modélisation et Logiciel numérique

NODE-NESTED MULTI-GRID WITH DELAUNAY COARSENING

Hervé GUILLARD

Mars 1993

Institut National
de Recherche
en Informatique
et en Automatique

2004 route des Lucioles
B.P. 93
06902 Sophia-Antipolis
France

Méthodes Multi-grilles par emboitement des nœuds du maillage utilisant une génération des niveaux grossiers par un algorithme de Delaunay

Hervé Guillard

Résumé

Sur des maillages non-structurés de type éléments finis, les algorithmes multi-grilles non-emboités nécéssitent de construire une séquence de maillages indépendants. On propose une méthode automatique de génération des niveaux grossiers à partir de la triangulation la plus fine. Cette méthode basée sur l'algorithme de Delaunay-Voronoï retriangule le nuage de points obtenus par élimination d'un certain nombre de nœuds de la triangulation de niveau supérieur. On présente l'algorithme et on montre qu'il possède de bonnes propriètés vis à vis des méthodes multigrilles. Quelques exemples d'applications à des maillages bi-dimensionnels sont présentés.

Node-Nested Multi-Grid method with Delaunay Coarsening

Hervé Guillard

abstract

For Finite-Element non-structured type meshes, the non-nested multigrid algorithms require to build a sequence of independent meshes. The paper proposes an automatic way to generate the coarse meshes given the finest one. The method first eliminates a set of points from the current mesh level and then uses the Delaunay-Voronoi algorithm to triangulate the remaining set of points. The algorithm is presented and it is shown that it owns good properties with respect to multigrid algorithms. Several examples of its application to bi-dimensional meshes are presented.

1. Introduction :

The discretization of partial differential equations coming from the modelisation of physical systems usually results in large systems of algebraic equations. Many different methods can be used to solve these linear or non-linear systems. However when the size of the system is large, methods that have a low order of complexity with respect to the number of degree of freedom of the system have to be preferred. This favours the use of Multigrid (MG) techniques whose computational cost is optimal with respect to the number of degree of freedom. In recent years, there has been intensive research into the design and analysis of these methods both from a practical and theoretical points of view. For elliptic systems of equations discretized by finite element techniques, most of this work has been done in the frame of the so-called variational multigrid technique where the finite element spaces are nested and the discrete bilinear forms on the coarse subspaces are inherited from the form on the finest level. From a practical point of view, this implies that the equations are solved on a sequence of grids that are obtained by successive subdivision. For instance in two dimensions, each triangle is subdivided in 4 by linking the midpoints of the edges thus generating a family of nested finite element spaces. Although this technique can be very powerful when combined with the use of adaptive refinement, it has several drawbacks. First this approach is actually a mesh generation technique for the construction of the finest level (that can be used independentely of any MG algorithm). When this mesh generation technique is applied to complex geometries, the control of the quality of the final mesh is not easy. Moreover, this option restricts the use of MG to the meshes generated by element subdivisions while one may want to benefit of the MG acceleration on given arbitrary meshes. The extension of the variational MG technique to three-dimensional problems is also not so straighforward. Actually, while in 2-D, the division of a given triangle gives rise to four triangles that conserve the ratio of the lenght of the edges thus ensuring that successive refinement will not degenerate, the corresponding property is lost in 3-D and successive subdivisions of a tetraedral mesh could potentially degenerate. These reasons motivate the use of alternate strategies where one can use as finest level a given arbitrary mesh, no matter how it is generated. This introduces the problem of the generation of the coarse levels. Let us consider for instance the mesh of figure 1.a., it is clear that there is no **natural** underlying triangulation that contains this mesh and therefore there is no natural way to generate a coarse mesh from this one. This is a situation that largely contrasts with the one encountered with structured meshes where the generation of the coarse levels is immediate. In previous works concerned with the implementation of MG techniques on non-structured meshes, apart from the Algebraic Multigrid technique (e.g. [14]), the technical difficulty of

the coarse mesh generation has been dealt with in essentially two ways :

In the volume agglomeration technique of [10], the underlying discretization method is a finite volume method. The coarse levels are then generated by agglomeration of control volumes. This results in a sequence of levels composed of embedded polygonal partition of the domain. From a technical point of view, this approach is closely related to the aggregation method used in economical sciences where no physical meshes can be identified [3]. However when this technique is applied to partial differential equations, only the finest level is composed of a finite-element mesh while the coarse levels are not defined by a set of nodes but only by a set of polygonal regions (or aggregated equations) . This creates difficulties in evaluating second-order derivatives (However, see the recent work of [9]) and consequently this approach has been primary successful for the approximation of first-order partial differential equations.

In the multi-mesh option, the sequence of coarse meshes are generated under the sole constraint that they approximately fit the same boundaries. In this approach, the different levels are totally unrelated and the finite-element spaces are non-nested. This technique has been used for instance for some aerodynamical problems in [11]. Although this approach is very general, its application leads to several difficulties. First to solve a problem in a given geometry, it is now required to construct not only a single mesh but a sequence of independent meshes filling approximately the same geometry. Practitionners of finite element methods know how tedious and time-comsuming can be the generation of a single mesh when complex geometries are in consideration. The need to generate many independent meshes multiplies the problem and is certainly not an argument in favour of the use of MG techniques. Moreover, multigrid efficiency requires that the ratio of the number of unknowns on the different levels be close to 2^d where d is the space dimension. This constraint is not easy to satisfy when independent meshes are created by automatic mesh generators. Finally, it should be mentionned that for these algorithms, the definition of the inter-grid transfer is complicated due to the lack of relation between the different levels.

In this work, we propose a general method to solve the coarse grid generation problem. The proposed algorithm combines the advantages of the volume agglomeration method with the ones of the multi-mesh options. Like in the volume agglomeration method, it generates the coarse level in a fully automatic way with no manual intervention requiring only the knowledge of the finest mesh. Moreover, the coarse grid construction ensures that the number of unknowns is approximately divided by

2^d on the successive coarse levels. But like in the multi-mesh methods, the discrete approximation on the different levels are consistent with the continuous operators and the method is not restricted to first-order partial differential equations. The organization of this paper is as follows : The next Section is devoted to the description of the proposed algorithm and give some remarks about its properties. Section 3 gives some examples of applications for bi-dimensional meshes. Finally, we end by some concluding remarks.

2. Delaunay Coarsening :

The idea underlying the proposed method is based on the use of the Delaunay-Voronoi algorithm to triangulate a set of points. We recall here the main features of this construction. Interested readers should consult the article by Aurenhammer [1] that reviews many questions related to Voronoi diagrams and gives an extensive bibliography on their use in a wide variety of fields. Given a finite set of points S in the space, the Voronoi diagram of S is the polygonal partition of the space constructed by associating to each data point the region of the space that is nearest to this point for the Euclidean distance. Closely related to the Voronoi diagram is the Delaunay triangulation of the convex hull of the data points S. It is defined as the unique triangulation of S such that no point i of S lies inside the circumsphere of any (triangle) tetraedron. Given a finite set of points S in the space, the Delaunay triangulation is unique provided there are no cyclic points. The use of the Delaunay triangulation is now becoming a rather common tool in automatic mesh generation and several methods are known to generate this triangulation in an optimal number of operations. In practice, the main difficulty associated to these method is to recover the boundary of the region of interest that does not necessary coincide with the convex hull of the data points. Again several algorithms do exist to constraint the triangulation to respect the boundaries, see [2], [4], [5]. For the two-dimensional examples displayed in Section 3, the triangulation have been generated using the algorithm described in [7, 8], the respect of the boundary been enforced by a simple swapping of the diagonal of the quadrilateral containing two boundary points [4], [6].

With the Delaunay-Voronoi as a tool to triangulate a set of points, the goal now is to generate from a given mesh a subset of nodes such that :
i) The geometry of the current mesh will approximately be respected by the triangulation of this subset.
ii) The number of nodes is approximately divided by a factor 2^d.

The following simple algorithm ensures the respect of these properties by splitting

the current set of nodes into two disjoint sets : the set of coarse nodes (that will belong to the coarse mesh) and the set of non-coarse nodes :

Coarsening Algorithm :

1. Re-order the list of nodes in such a way that the boundary nodes appear first in the list, then considering successively each node :
2. if the node is already a non-coarse node , go to the next node in the list.
3. else add this node to the list of coarse nodes and delete all its neighbors in the list of coarse nodes.
4. go to the next node in the list.

This coarsening algorithm is then followed by a Delaunay-Voronoi triangulation of the set of coarse nodes. It is then applied recursively taking as initial mesh the newly generated triangulation until the number of remaining nodes is to small to describe the geometry.
It is worth noticing that this algorithm has exactly the same structure that the coarsening algorithm of [10]. Actually, the first application of the present algorithm and of the algorithm of [10] will generate the same type of structure, a coarse node in the present algorithm being the center of an agglomerated cell in the algorithm of [10]. However, this property will be lost after the generation of the second coarse level, the connectivity being different in the two cases after the first application of the algorithm.
We now proceed to show that the previous algorithm fulfils the requirements i and ii. First because the boundary nodes are the firsts to appear in the list and because the neighbors of coarse nodes are automatically non-coarse nodes, it is clear that the first layer of mesh points from the boundary will be deleted from the list of coarse nodes. Then only the nodes belonging to the second layer of mesh points from the boundary will be able to suppress a node from the list of coarse nodes. But these suppressed nodes cannot be boundary nodes because there is no direct connectivity between a boundary node and a second-layer node. Therefore, it is clear that the only point that can suppress a boundary node is itself a boundary node. In order to respect the geometry of the problem, we have then only to deal with boundary nodes. It is easy to re-order the numbering of the boundary nodes in such a way that the firsts to appear in the list are the ones corresponding to peculiar features of the geometry. For instance, in the bi-dimensional examples shown in Section 3, we

re-number the boundary nodes according to the absolute value of the angle formed with their two neighbors. Therefore the sharpest summits (in absolute value) of the polygonal region forming the geometry are the lasts to disappear and the geometry is approximately conserved through the different levels. At this point, it is worth mentioning that when very coarse meshes are created, a further deletion of boundary points can result in a gross change in the geometry, no matter how is done this deletion (for instance the deletion of any boundary node in figure 3.d will result in a dramatic change in the geometry). In general, this means that the number of boundary points is critical to describe the geometry and the coarsening algorithm has to be stopped. A convenient way to decide when to stop the coarsening algorithm is to measure the volume of the polygonal region covered by the new coarse mesh : if this volume is too different from the previous one, this implies that the number of boundary nodes of the new mesh is too small to describe the geometry and the coarsening algorithm has to be stopped. Note also, that is is possible to force certain boundary points to be always coarse nodes through the different levels at the price of certain manual intervention, this is for instance the case of the four nodes defining the square of figures 3.c and 3.d.

That the generated mesh is twice coarser (in each space direction) that the initial one (requirement ii) follows from the same arguments that was used in the previous discussion. Let us assume that the mesh is locally regular, $i.e$ that in a given region of the space, the size s of the elements (s is any convenient measure of the size of the elements, for instance their diameters) is included between h_m and h_M. Note that h_m and h_M does not have to be constant for the whole mesh but only to be slowly varying functions. Then, consider the first and second "shells" of nodes composed of the neighbors of node i and of the neighbors of its neighbors. Due to the assumption of local regularity, the "radius" of the first shell is comprised between h_m and h_M while the radius of the second one is comprised between $2h_m$ and $2h_M$. If i is a coarse node, then all the nodes of the first shell will be deleted and therefore the neighbors in the coarse mesh of i will be nodes belonging to the second shell. The average distance between i and its neighbors in the coarse mesh is then between $2h_m$ and $2h_M$. We note that the previous argument is not totally rigorous as the random numbering of nodes in a non-structured mesh may result in the deletion of nodes of the second shell and in that case the distance between i and the nearest point of the third shell scales with $3h_m$ and $3h_M$. A possible strategy to avoid this problem is to re-number the nodes by successive shells starting from the boundary (Note that if a frontal algorithm is used to generate the mesh, this is implicitely done). However it is very unlikely that this problem happens for **all** the nodes of the second shell and in practice we have found unnecessary to modify the numbering

of the nodes. Actually the following results show that the ratio of the number of nodes between two successive levels is extremely close to 2^d and further refinements of the algorithm does not seem profitable.

3. Examples of Applications :

NACA0012 Airfoil

Our first example concerns the mesh around a NACA0012 airfoil, the original mesh is composed of 800 nodes and 1514 elements. Figure 1 displays a part of the original mesh together with the first coarse mesh. It can be seen that the coarse nodes belongs to the fine triangulation, however, the elements of the coarse mesh are not deduced from the ones of the fine mesh by subdivision and the finite-element spaces are indeed not nested.

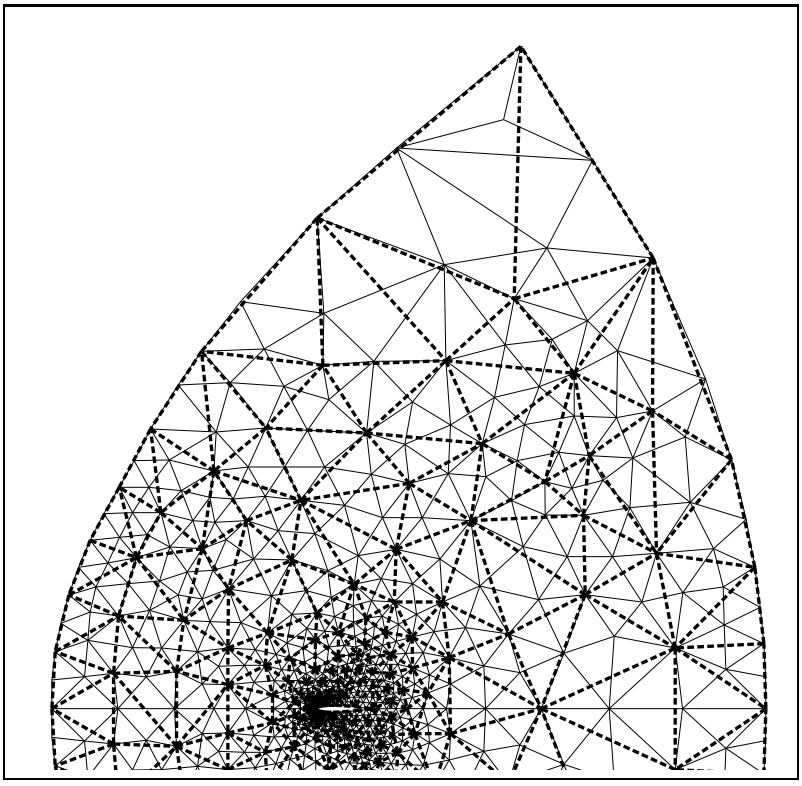

Figure 1. Partial view of the initial and first coarsened meshes

Figure 2 shows the successively generated meshes. Two facts worths to be noticed : First for this relatively simple shape, an extremely coarse mesh is finally generated, it has only 6 elements and the airfoil is reduced to a simple line. Second even if the initial mesh is symmetric, the coarse ones are not. The reason being that the process of elimination of the nodes of the fine mesh proceeds according to the nodes numbering and the node numbering is not symmetric even if the mesh is. It has been noted [12] that this non-symmetry of the coarse levels may slow down the convergence of a Full Multigrid process when the solution is itself symmetric. For instance in this problem when the angle of attack is non-zero and the solution is not symmetric, FMG is rather efficient for computing the flow around the obstacle but for a null angle of attack where a symmetric solution is expected, a decrease in efficiency is indeed noticed [12]. For these cases, it is then advisable to always consider symmetric meshes, this can be simply done by working with an half geometry.

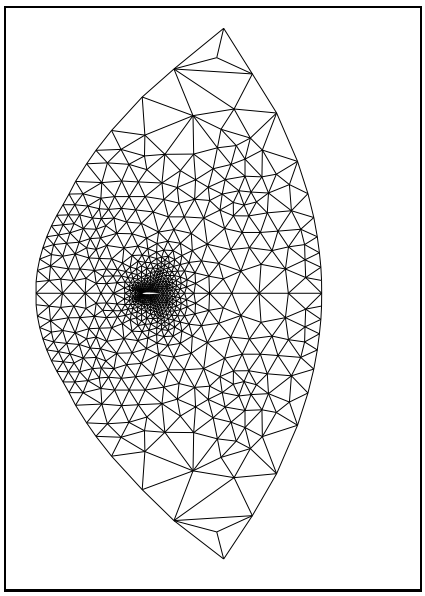

Figure 2.a : initial level

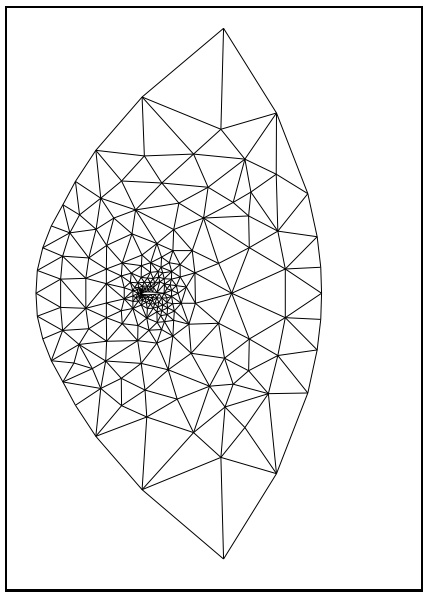

Figure 2.b : after one coarsening

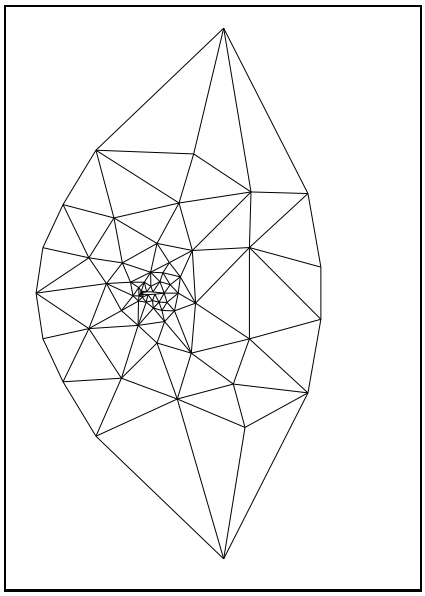

Figure 2.c : second coarsening

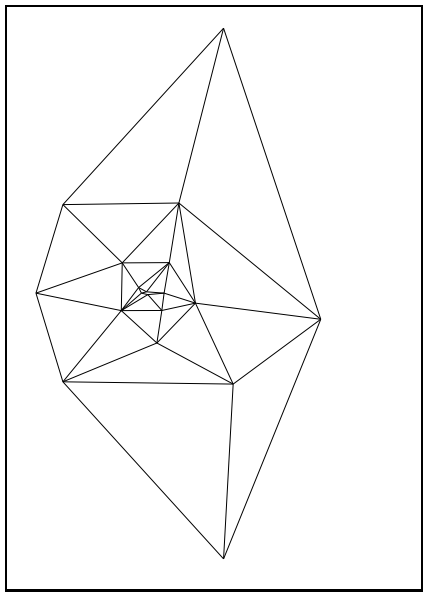

Figure 2.d : third coarsening

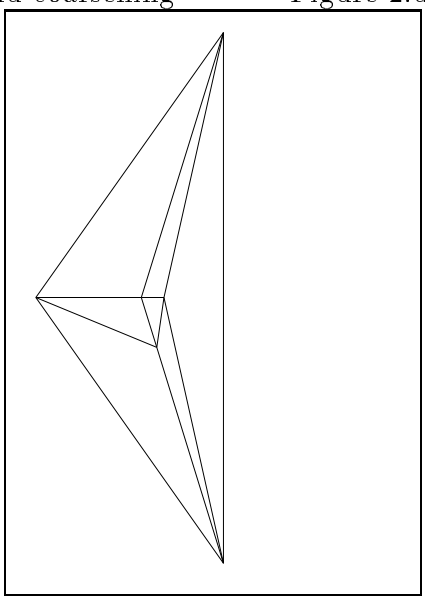

Figure 2.e : coarser mesh

Table I shows the number of nodes and elements of the successively generated meshes. The coarsening process has been stopped after four successive applications of the algorithm. Figure 2.e indeed shows that there is no further possibility of coarsening.

Table I.

	nb nodes	ratio	nb elements
Level 0	800		1514
Level 1	223	3.59	404
Level 2	67	3.33	113
Level 3	19	3.53	28
Level 4	6	3.17	6

It can be checked that the ratio of the number of nodes on two successive levels is close to 4. Moreover, examination of figures 2.a and 2.b for instance, shows that the doubling of the space steps is essentially local and that the average relative density of grid points is conserved from one level to the other one.

Mesh around a Spark plug

Our second example is composed of a mesh around a bi-dimensional model of a spark plug. The initial mesh has been generated using a Delaunay algorithm. The geometry of this example is more complex that the preceeding one and it has not been possible to generate a very coarse mesh without changing the geometry of the problem. Only three applications of Delaunay coarsening have been possible. Actually the third level has been generated by imposing manually that the four nodes defining the square will not be deleted (They are neighbors of coarse nodes in the second coarse level, see Figure 3.c). Table II shows again that the ratio of the number of nodes on two successive levels is close to 4.

Table II.

	nb nodes	ratio	nb elements
Level 0	2801		4091
Level 1	549	5.10	967
Level 2	141	3.89	222
Level 3	43	3.27	55

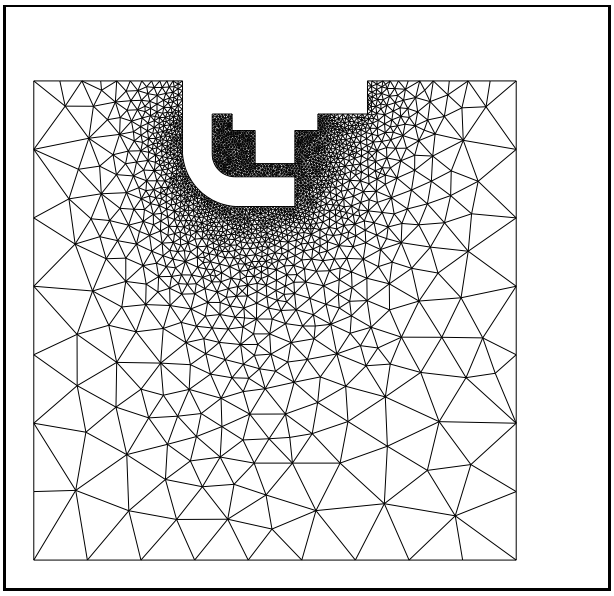

Figure 3.a : initial level

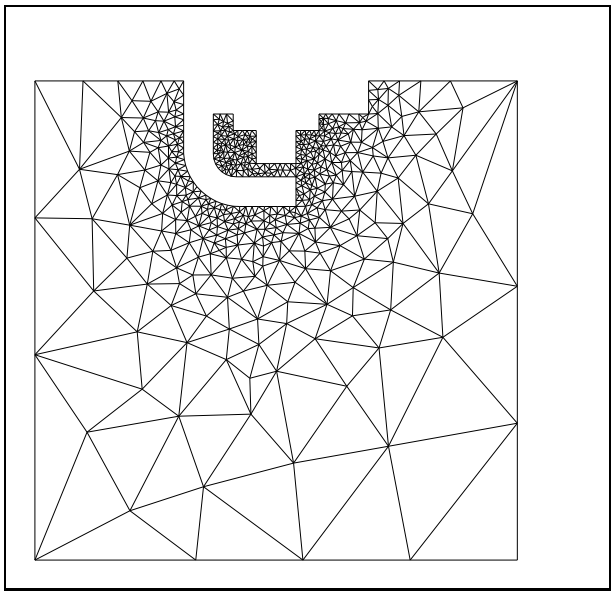

Figure 3.b : after one coarsening

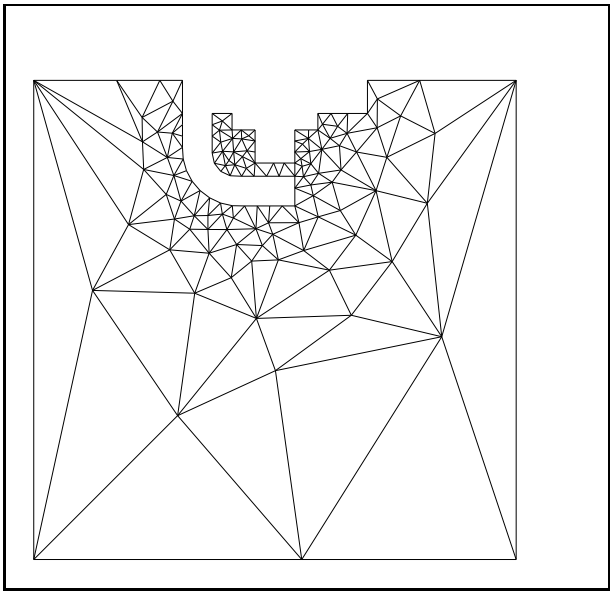

Figure 3.c : second coarsening

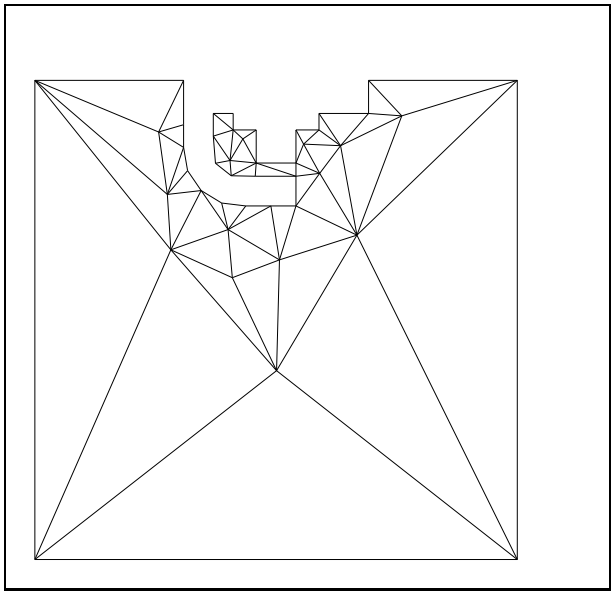

Figure 3.d : coarser mesh

Supersonic Air Intake

Our final example is a mesh generated by a Delaunay algorithm around a bi-dimensional model of an air intake. The number of nodes of this mesh is much larger than for the previous two examples because the extremely sharp edge of the upper part of the intake has to be adequately represented. Consequently the size of the elements varies strongly between the different parts of the domain. However, Figure 4. shows that the relative density of grid points translates from one level to the next coarser ones. The level 3 mesh is the coarser that can be generated without changing the geometry of the domain, small elements remain around the nose of the air intake but the intake is itself represented only by one element and further coarsening will change the nature of the problem. For this relatively large mesh, it can be checked from Table III that the ratio of the number of nodes on two successive levels is extremely close to 4.

Table III.

	nb nodes	ratio	nb elements
Level 0	6574		12695
Level 1	1600	4.11	2975
Level 2	399	4.01	688
Level 3	105	3.8	156

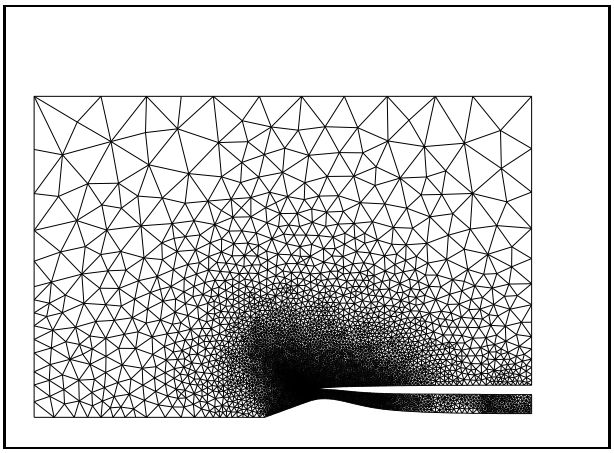

Figure 4.a : initial level

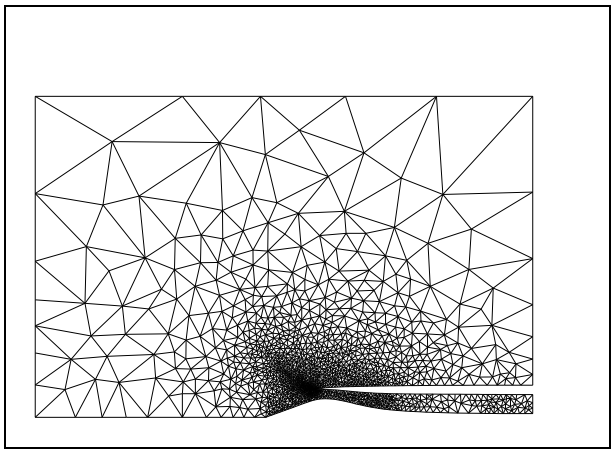

Figure 4.b : after one coarsening

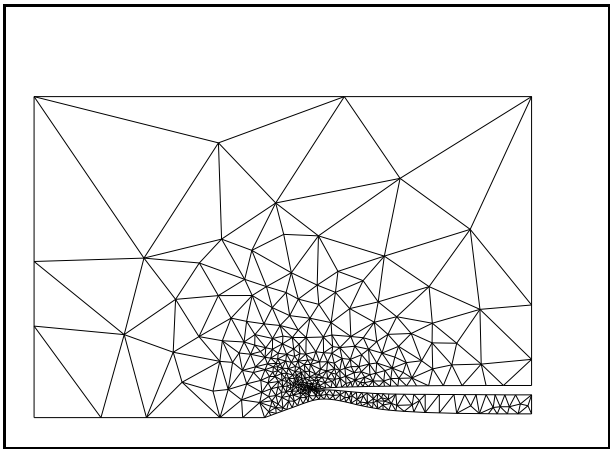

Figure 4.c : second coarsening

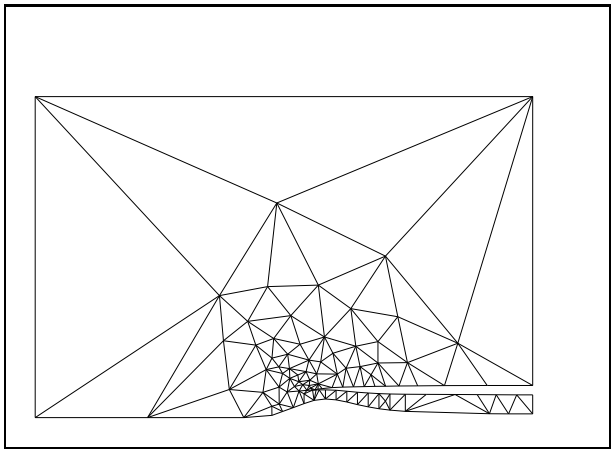

Figure 4.d : coarser mesh

4. Concluding Remarks

This paper has presented a method to generate a sequence of coarse meshes from a given arbitrary fine one. The method is fully automatic and does not require any manual intervention. Moreover the way the coarse meshes are generated ensures that the space step is approximately doubled in each space direction at each application of the algorithm. This algorithm thus largely simplifies the use of MG methods in the non-nested finite element framework, first by avoiding the need to manipulate and generate different independent grids and second by simplifying the coarse to fine inter-grid transfers that may be trivial injections. Up to now, the use of node-nested meshes has been applied with success to the MG solution of the 2-D Euler Equations [13]. Future works will consider the extension of the algorithm to three dimensional geometries and the study of the convergence properties of elliptic MG in the node-nested framework.

Acknowledgements

This study was supported in part by DRET under contract 90-185. Thanks to Alain Dervieux for its obstinacy to force me to write down this report and to Eric Morano who has tested the use of node-nested meshes in the multigrid method he has developped for the Euler and Navier-Stokes equations.

References

[1] F. Aurenhammer, Voronoi diagrams-A survey of a fundamental data structure, ACM Computing surveys, **23**,No 3,(1991), pp 345-405.

[2] T. J. Baker, Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation, Engineering with Computers, **5**, (1989), pp 161-175.

[3] F. Chatelin, W. L. Miranker, "Acceleration by Aggregation of successive Approximation methods", Linear Algebra and its Applications, **43**, pp 17-47, (1982)

[4] P. L. George, Génération de maillage par une méthode de type Voronoï, INRIA Research Report 1398, (1991).

[5] P. L. George, Génération de maillage par une méthode de type Voronoï, Partie 2 : Le cas tridimensionnel, INRIA Research Report 1664, (1992).

[6] P. L. George, F. Hetch, E. Saltel, Automatic mesh generator with specified boundary, Comp. Meth. in Appl. Mech and Engineering, **92**, (1991),pp 269-288.

[7] F. Hermeline, Une méthode automatique de maillage en dimension n, Thèse de 3e cycle, Paris 6, 1980.

[8] F. Hermeline, Triangulation automatique d'un polyèdre en dimension n, RAIRO, Analyse Numérique, **16**, No 3, (1982), pp 211-242.

[9] B. Koobus, M.H. Lallemand, A. Dervieux, Unstructured Volume Agglomeration MG, Solution of the Poisson equation, submitted to Int. J. for Numerical Meth. in Fluid.

[10] M.H. Lallemand, H. Steve, A. Dervieux, Computers & Fluids, **21**, (1992), pp397-433.

[11] D. Mavriplis and A. Jameson, Multigrid solution of the two-dimensional Euler equations on unstructured meshes, AIAA paper 87-0350, (1987).

[12] E. Morano, Résolution des équations d'Euler par une méthode multigrille stationnaire, Ph D Thesis, University of Nice, 1992.

[13] E. Morano, H. Guillard, A. Dervieux, M.P. Leclerc, B. Stouffet, Faster relaxations for non-structured MG with Voronoi Coarsening, *Computational Fluid Dynamics '92, Proceedings of the first European Computational Fluid Dynamics Conference, Brussels, 7-11 september 1992, Ch. Hirsch, J. Periaux, W. Kordulla (eds)*, Vol I, pp 69-74, Elsevier, 1992.

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