AUTOMATIC 3D MESH GENERATION WITH PRESCRIBED MESHED BOUNDARIES

P.L. George, F. Hecht, E. Saltel
INRIA Domaine de Voluceau- Rocquencourt BP 105
78153 LE CHESNAY FRANCE

October 20, 1989

Abstract

Devoted to the mesh generation of 3D-domains, this paper briefly describes different approaches actually in progress.

A new method is introduced which can be seen as a variant of the Delaunay-Voronoï's tessellation coupled with a control of the given boundary used to defined the domain to be meshed.

1 Introduction

The availibility of automatic generator of meshes for 3D domains with complex geometry is more and more a crucial necessity. This is the solution to be able to solve physical problems formulated through P.D.E. by the use of numerical simulation based upon Finite Element.

A typical problem of mesh generation can be seen as that connected to the simulation of characteric fields associated to electrotechnic problems: one has to create a mesh for the domain of interest that can be very complex. Engineers are already able to mesh the domain of interest using actual possibilities (derivation of 3D mesh from 2D one, "mapping" method [5],[2], front advancing technique [12], "algebraic" approach or change of variables [15]), their science and a lot of time.

This paper is devoted to the presentation (section 2) of a method coupling an approach based on a variant of the Delaunay-Voronoï's technique (see [1],[3],[11] for a different approach to this method), an original algorithm for the consideration of the **specified boundaries** (section 3) and a process (section 5) for the creation of internal points to obtain good shape elements. Conclusions and examples of application are shown in section 5.

2 Partition with specified vertices

$\{P_k\}_{k=1,n}$ is a set of points in $\mathbb{R}^3$ which are not all coplanar and T_i is a tetrahedral mesh whose vertices are the i-first points of $\{P_k\}$ such that: $(H1)$ P_{i+1} is inside at least one element of T_i.

We will propose a method for the creation of T_{i+1} using local modification of T_i.

2.1 Definition of a process for insertion of points

From T_i, C_r a criterium (or a set of criteria) and C_o a constraint (or a set of constraints), the set of tetrahedrons $\mathcal{S}$ is defined such that:

$(H2)$ $\forall K \in$ S, K satisfies the criterium C_r with respect to P_{i+1}

$(H3)$ $\forall K \in$ S, K satisfies the constraint C_o

It is sufficient to be able to remesh $\mathcal{S}$ such that P_{i+1} is a vertex of this mesh (let $\mathcal{S}'$ this new local mesh) and then:

$$T_{i+1} = \{T_i - \mathcal{S}\} \cup \{\mathcal{S}'\}$$

A pertinent choice of C_r and C_o insures the validity of this local remeshing process (LRP) and can insure extra desirable properties; for instance, C_r can simply mean: $\mathcal{S}$ is star-shaped w.r. to P_{i+1}. Different possible choices of criteria and constraints are discussed in [8].

2.2 Full description of the proposed method

LRP, C_r and C_o assumed, the construction of the adequate set S is done as:

1. research of element(s) of T_i including the point P_{i+1}.
2. initialization of S with this (these) element(s)
3. research by proximity w.r.to elements in S of elements to be added in this set, i.e. those satisfying both C_r and C_o.

Step 1 of this process implies that $\forall i$, $P_{i+1} \in T_i$, this is achieved by creation of mesh T_o consisting of the 5 tetrahedrons of a "box" including all the given points $\{P_k\}$. As this property is recurrent, as long as points P_i are inserted, step 1 is based on the following three possible results: P_i is inside an element K, it can be located on a face or along an edge.

To quickly obtain this starting set, one can follow [9],[13],[1]. The creation of set $\mathcal{S}$ is then achieved by proximity (with constraint C_o) w.r.to elements already in it. When $\mathcal{S}$ has been obtained, one has to suppress its elements and to create (LRP) $\mathcal{S}'$ a new mesh of $\mathcal{S}$ whose vertices of elements include point P_{i+1}, so T_{i+1} is hold.

The method for the creation of a mesh with specified vertices in $\mathbb{R}^3$ can be seen as follows:

Step 1 : compute the extrema of coordinates of the given points,

Step 2 : derive the 8 extra points of the including box,

Step 3 : partition this box into 5 tetrahedrons,

Step 4 : insert, one after one, the given points.

Note : When suppressing elements which have at least one vertex identical to one extra points, a partition including only all given points is obtained. In [9] it is shown that this mesh is not, in general, that of the convex hull of the given set of points.

3 Full respect of a specified boundary. Elimination of the exterior of a domain

3.1 Position of the problem

In fact we have to create a mesh of a domain D from its meshed boundary (say a list of triangular faces) whose vertices compose the set of points we can consider by method of section 2. The produced tetrahedral partition is a mesh of the enclosing box whose vertices are endpoints of the box, eventual extra points and points of the boundary.

As we consider domains with arbitrary geometry (non convex domains for instance), we have to suppress all exterior elements. This step clearily requires the existence of the boundary in the mesh to have exterior and interior defined. Regrettably the mesh due to previous method, in general, does bf not contain as faces all the given faces (see [6],[4]). So we need a special phase to be able to insure such property. To solve this crucial problem, we can locally mesh the boundary in a different manner to insure the existence of this boundary mesh as a result of previous process [1],[13] or locally differently mesh by the boundary using a sequence of local swaps [6] to suppress elements crossing the boundary.

0018-9464/90/0300-0771$01.00 © 1990 IEEE

The first method is suitable when a global creation of meshes is supposed, the second is well adapted to a modular conception of these, this is why it is advised in this paper.

3.2 General principle of the method

At least one edge (or one face) of the specified boundary does not exist in the mesh produced by method of section 2, so let us define a set of geometrical modifications for the purpose of a) creating an edge or a face which is missing and on the other hand b) suppressing an undesirable edge or face.

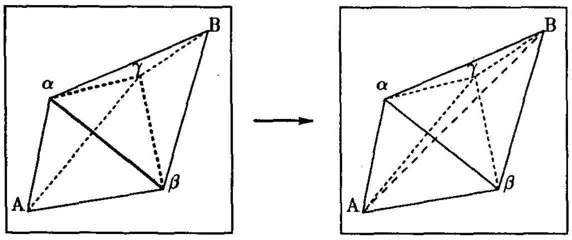

Figure 1: Example of process $2 \rightarrow 3$

Figure 1 shows the creation of an edge and the suppression of a face.

3.3 Definitions and local modifications

Two meshes are said to be **equivalent** if: first they partition the same domain, and second the boundary of this domain is partitioned in the same way in each of these.

Let us first consider the case of a missing edge and denote AB such an edge, we can classify the elements of the mesh w.r.to line AB as:

1. elements including point A (B) as vertex,
2. elements including at least one face intersected by line AB,
3. elements including at least one edge intersected by line AB,
4. elements with possibly A (or B) as vertex but with an empty intersection with line AB.

From these cases, let us define the following sets:

- a **pipe** is the ranked sub-set consisting of elements in situation 1) and 2).
- a **local shell** is the set of elements in situation 3).
- a **shell** is the set of elements such that at least one local shell exists.

Similarly, when specified edges are present, for the case of one missing face, we define the possible situations w.r.to this and more precisely w.r.to the **triangle** consisting of its 3 edges:

- this triangle is intersected by only one edge: the resulting set is a local shell with only 3 elements.
- this triangle is intersected by several edges: to each of them is associated a local shell whose number of elements is not known.

For each missing item (edge and then face) we separate the actual mesh into two sub-sets: a) the set of elements not affected, in the previous sense, by the missing item; it will not be modified and b) the set of elements affected by the missing item: a pipe, a shell or one or several local shells.

The proposed method consists of local modifications of this last set in order to obtain a new equivalent mesh (cf definition) such that missing items are created and undesirable items are suppressed. It is based on the following modifications:

From 2 tetrahedrons adjacent by a face and such that the line joining the 2 opposed vertices is internal it is possible to create 3 tetrahedrons ($2 \rightarrow 3$) recovering the same domain. In this equivalent mesh is included as edge the previous line and the undesirable face is removed.

The opposed process($3 \rightarrow 2$), with similar assumptions, consists of the creation of 2 tetrahedrons from 3 initial ones and produces opposite effect.

Various others modifications are developed in [8].

3.4 Creation of missing edges

For each missing edge the affected set of elements is either a pipe or a shell.

Proposition 3.1. The use of the previous processes makes possible the creation of an equivalent mesh of a pipe such that the missing edge is generated.

Proposition 3.2. A local shell can be equivalently remeshed by the insertion of one internal point. The new mesh does not contain the edge from which the shell was due.

Proposition 3.3. : An equivalent mesh such that all the specified edges are existing is obtained when applying the two previous results to every missing edges (the boundary is not crossed).

Proofs of previous propositions are detailed in [8].

3.5 Creation of missing faces

From the mesh obtained at the previous step we analyse the possible situations of elements affected by a missing face.

In [6], it is shown that a situation reduced to a shell with only 3 elements can be solved by $3 \rightarrow 2$ (directly or after remeshing):

Proposition 3.4. : A local shell with 3 elements due to the intersection of an edge with the triangle associated to a missing face can be remeshed in such a way that this face is created.

Proposition 3.5. : From a situation including several local shells it is possible to create equivalent meshes such that the number of its shells decreases.

When this number is 1 proposition 3.4 achieves the proof.

By application of these results for each of the missing faces, we create an equivalent mesh fitting exactly the given boundary. As a consequence notions of exterior and interior can be accessed.

3.6 Recognition and elimination of the exterior of an object:

The theory of graphs is used for this task, an original algorithm is proposed in [8] for the recognition of all the connected components of the domain and the suppression of exterior elements.

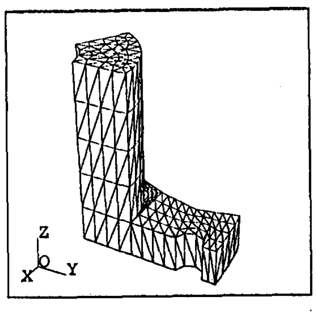

Figure 2: Boundary mesh after suppression of exterior

4 Quality and creation of internal points

The mesh produced at this step (denoted as the boundary mesh, see figure 2) is a tetrahedral partition of the domain whose vertices are the points of the specified boundary and some points created through the regeneration of the boundary. This mesh is not, in general, adequate for Finite Element computation (see figure 5), so we have to modify it to obtain good shape elements.

4.1 Controlling spaces

Let Ω be a **controlling space** such that mesh D is included in Ω, to each point P of Ω and for the direction d of the sphere S^2 is associated a functional h which defines the desirable step of the mesh: $h(P,d) : \Omega \times S^2 \rightarrow R$

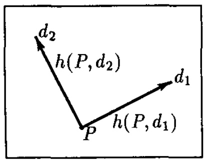

Figure 3: Example of ponctual control

The aim of this section is to create a new equivalent mesh such that in all of its points, h is satisfied.

Note: If h(P,d) is independent of d, the control is said to be **isotropic**. If not it is said to be **anisotropic**.

If no special definition is provided for the space Ω we advise to define the controlling space from the boundary mesh. The relative location of points on the boundary provides the value of h so the controlling space is strongly related to the prescribed data. To obtain the stepsize h at a point P not on the boundary, several choices are pertinent, one realistic can be written as:

$$h(P,d) = \left(\sum_{i=1,4} b_i h(P_i,d)^p \right)^{\frac{1}{p}}$$

where P_i are the vertices of the tetrahedron T including point P, b_i is a weight associated to P_i, p a ratio for the control of refinements from the boundary.

To insure desirable properties for the final mesh of domain D, the controlling space Ω can be fully defined. Basically Ω is any partition including D for which the functional h is defined.

4.2 Notion of quality

To each element T of D is associated the quantity Q defined as:
$Q(T) = r/lmax$ where

$$lmax = 0.5 \max_{\text{edge AB}} \left(\frac{l_{AB}}{h\left(A, \frac{B-A}{\|A-B\|}\right)} + \frac{l_{BA}}{h\left(B, \frac{A-B}{\|B-A\|}\right)} \right)$$

A and B are the 2 endpoints of the considered edge and r is the radius of the bigger sphere B(P,r), w.r.to functional h, inscribed in the tetrahedron T.

Note: for isotropic case and for h constant this definition is quite equivalent to the classical one.

To obtain a mesh D of good quality, i.e. such that:

$$\forall T \in D \quad Q(T) = \frac{kr}{lmax} \text{ is close to 1. w.r.to the controlling space } \Omega$$

we advise first to use local modifications applied to some elements of D and then to create internal points.

4.3 Local modifications

- a) Suppression of a face and creation of an edge:

  Process 2 $\longrightarrow$ 3 can be applied if the worst of the 3 elements created is better than the worst of the 2 initial tetrahedrons. It will be performed as long as line AB (cf figure 1) is internal to the polyhedron $T_1 \bigcup T_2$, except if face $\alpha\beta\gamma$ is a specified one.

- b) Create a different mesh for local shells:

  This new local mesh is based upon the following process:

  - election of the elements with common edge AB,
  - determination of the polyhedron $a_1, a_2, ..., a_n$,
  - generation of a different mesh for this polyhedron,
  - creation of the new mesh by simple connection of A and B with this last mesh (cf figure 4).

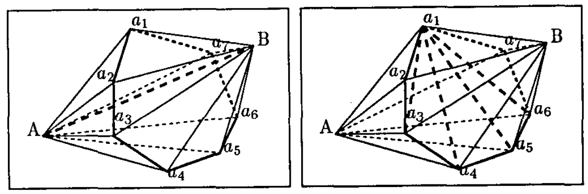

Figure 4: Example of rewriting of a shell $n_T \rightarrow 2n_T - 2$

- c) technique of barycentrage applied to free vertices by a weighted barycentrage of free vertices or by an iterative approach [7].

4.4 Creation of internal points

The method discussed in section 2 can also be used to insert internal points in an existing mesh. More precisely we run (LRP, C_r, C_o) with a constraint C_o concerning the increase in the quality.

Two problems are to be solved: a) find a criterium to decide if a point must be created and b) find a location for it

In dimension 2 it seems that the technique based on the suppression of all too big elements (cf [10],[5]) produces good results. Such a simple method is not efficient in 3 dimension since Voronoï's method (or method of the same class) can produce degenerated elements with any desirable volume. That is why a non purely local process is advised.

So we suggest as in [14], to consider first local shells (cf definition). Such a shell will be differently partioned as long as the quality of the produced mesh obtained with new points is better w.r.to Ω (location of points to be inserted will be derived from functional h). Another solution is to consider as set to be modified the ball associated to a vertex, i.e. all the tetrahedrons with this vertex as point.

4.5 Methods for regularization

For regularization purposes, local modifications and displacement of free vertices can be performed.

4.6 Results and performances

To examine the quality of the generated elements, the quantity Q(T) is measured for each element T of the mesh, see figures 5 and 6.

In histograms dotted lines and plain lines respectively describe the distribution of the given mesh of the boundary (2D) and that of the boundary mesh (3D) and then the final one. In abcissa is noted the quality (in fact the irregularity), in ordinate is denoted the number of elements with such quality. The higher is the value of the irregularity the worse is the element.

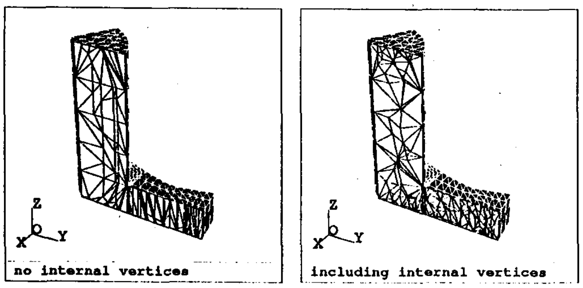

Figure 5: Cut through boundary and final mesh

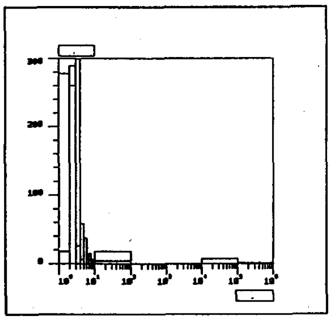

Figure 6: Histogram of quality for boundary and final mesh

5 Examples of application and conclusions

The method, summarizing the results of previous sections, has been programmed using Fortran and currently runs on various workstations (Apollo,Sun,Vax,..). The corresponding package has been tested on more than 100 different domains.

Figures (7,8) display a selection of tested domains (one can note that domains or inverse-domains have been succesfully tried).

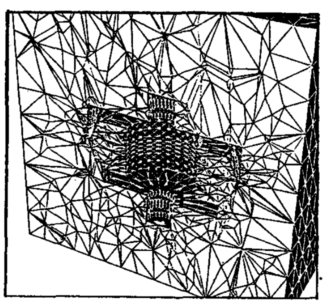

Figure 7: Exterior of an alternator:23663 elements,5371s

bf Concluding remarks:

The actual release of the program is based on a strictly isotropic control. We claim that the foundation of the method (cf section 4) is well adapted to the introduction of anisotropic controls. Similarly adaptation of meshes is possible through an adequate definition of Ω based on the solutions, its gradient or its hessien. It is obvious that such a mesh generator will give more advantage when coupled with a C.A.D. C.A.M. system.

As a final conclusion we claim we have validated the proposed method and we plan to continue the study for a better answer to above observations.

Acknowledgments This study was partially supported by D.R.E.T (Paris) we want to thank here.

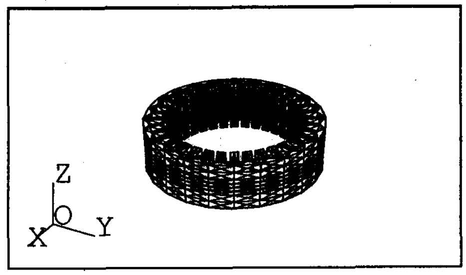

Figure 8: Stator of an alternator:13223 elements,1468s

6 Bibliography

[1] T.J.BAKER, Unstructured mesh generation and its application to the calculation of flows over complete aircraft, CFD for Aerospace problems, methods and codes, Univ. Tennessee Space Institute Workshop, March 1988.

[2] J.C.CAVENDISH, D.A. FIELD, W.H.FREY, An approch to automatic 3D finite element mesh generation, Int. Jour. Meth. in Eng., Vol 21, 1985.

[3] Z.J. CENDES, D.N. SHENTON, H. SHAHNASSER, Magnetic field computations using Delaunay tringulations and complementary finite element methods, IEEE, Trans. Magn. 21, 1985.

[4] J.L.COULOMB, Maillages 2D et 3D. Experimentation de la triangulation de Delaunay, Conférence on Automated mesh generation and adpatation, Grenoble, 1987.

[5] P.L.GEORGE, Mailleur 3D par découpage structuré d'éléments grossiers, RR n° 990, INRIA, 1989.

[6] P.L.GEORGE, F.HECHT, E.SALTEL, Automatic mesh generation with specified boundary, submitted in 1988 to Computer Methods in Applied Mechanics and Engineering.

[7] P.L.GEORGE, F.HECHT, E.SALTEL, Constraint of boundary and automatic mesh generation, in Numerical grid generation in computational fluid mechanics, Miami, 1988.

[8] P.L.GEORGE, F.HECHT, E.SALTEL, Full automatic mesh generator for 3D domains of any shape, to be submitted.

[9] P.L.GEORGE, F.HERMELINE, Maillage de Delaunay d'un polyèdre convexe en dimension d. Extention à un polyèdre quelconque, CEA-N-90, 1989.

[10] F.HERMELINE, Triangulation automatique d'un polyèdre en dimension N, RAIRO numerical analysis vol 16, n° 3, 1982.

[11] A.JAMESON, T.J.BAKER, N.P.WEATHERILL, Calculation of inviscid transonic flow over a complete aircraft. AIAA 24th Aerospace Sciences Meeting, Reno Nevada, USA, 1986.

[12] R.LOHNER, P.PARIKH, Generation of 3-D unstructured grids by the advancing front method, AIAA 26^{th} Aerospace Sciences meeting, Reno Nevada, USA, 1988.

[13] A.PERRONNET, Tétraèdrisation d'un objet multi-materiaux ou de l'exterieur d'un objet, RR 88005, LAN 189, Univ. Paris 6, 1988.

[14] J.Y.TALON, Algorithmes d'amélioration de maillages pour éléments finis en 2 et 3 dimensions, Conférence on Automated mesh generation and adaptation, Grenoble, 1987.

[15] J.F.THOMPSON, A general three dimensional elliptic grid generation system on a composite block-structure, Computer Methods in Applied Mechanics and Engineering, Vol 64, 1987.