

# The Extended Delaunay Tessellation

Nestor Calvo<sup>1</sup>, Sergio R. Idelsohn<sup>1 and 2</sup> and Eugenio Oñate<sup>2</sup>

<sup>(1)</sup> *International Center for Computational Methods in Engineering (CIMEC-INTEC),  
Universidad Nacional del Litoral and CONICET, Santa Fe, Argentina, [sergio@ceride.gov.ar](mailto:sergio@ceride.gov.ar)*

<sup>(2)</sup> *International Center for Numerical Methods in Engineering (CIMNE),  
Universidad Politécnica de Cataluña, Barcelona, Spain, [onate@cimne.upc.es](mailto:onate@cimne.upc.es)*

---

## Abstract

*The Extended Delaunay Tessellation is presented in this paper as the unique partition of a node set into polyhedral regions defined by nodes lying on nearby Voronoï spheres.*

*Until recently, all the FEM mesh generators were limited to the generation of tetrahedral or hexahedral elements (or triangular and quadrangular in 2D problems). The reason of this limitation was the lack of any acceptable shape function to be used in other kind of geometrical elements. Nowadays, there are several acceptable shape functions for a very large class of polyhedra. These new shape functions, together with the Extended Delaunay Tessellation, gives an optimal marriage and a powerful tool to solve a large variety of physical problems by numerical methods.*

*The domain partition into polyhedra presented here does not introduce any new node nor change any node position. This makes this process suitable for Lagrangian problems and meshless methods in which only the connectivity information is used and there is no need for any expensive smoothing process.*

Keywords: Mesh Generation, Delaunay, Voronoï, polyhedral mesh, Extended Delaunay, Meshless

---

## 1. INTRODUCTION

Several numerical methods in computational mechanics, as well as other volume integration methods need to subdivide the total domain in sub-domains called “elements”. This is the case for the Finite Element Method (FEM) and the Finite Volume Method (FVM).

Once defined the total domain by the boundary surfaces, two standard algorithms may be used to perform the partition:

- a) divide the total domain into elements by an advancing front technique (AFT)<sup>[1,2]</sup>, or
- b) introduce a point distribution in the domain and then perform the partition via a Delaunay tessellation (DT)<sup>[2]</sup>.

Both methods give, as a first step, unacceptable partitions to be used in a numerical method. For this reason, the first step needs an optimization of the element shape (sometimes called smoothing or “cosmetic” process), which is an iterative problem. For this reason, both methods are unbounded in computing time: the number of operations to obtain an acceptable partition to be used in a numerical solution may not be written as a function of the number of elements or nodes.

Both methods have advantages and disadvantages, which may be summarized as:

- a) In the AFT, the boundary surfaces are easily described, but the method is very expensive compared with the DT. For this reason is unacceptable to be used in problems with permanent partition update.
- b) The big advantage of the Delaunay tessellation is that it is inexpensive and bounded in computing time. Unfortunately it needs a lot of “cosmetic” in order to be used in a numerical method to solve a 3D physical problem. For this reason, the computing time may become as large as in the AFT.

Why the problem of a domain partition is until now an open question? The main reason is that in several physical problems, it is necessary to perform a domain partition at each time step. This is for instance the case for transient problems with moving boundaries or in lagrangian formulations in fluid mechanics. In these cases the use of a bounded method is mandatory, the number of operations to achieve an acceptable partition must be fixed as a function of the number of elements or nodes. Both

of the previous methods, AFT and DT, need a process of cosmetics which increases without a fixed limit the computing time.

In this paper, a method based on the DT will be described. The problem will be: given a node distribution, find an optimal partition, regarding computing time and element shape, to be successfully used in a numerical method.

It must be noted that obtaining a node distribution in a domain is not a difficult task and it is a bounded operation. The goal of this paper will be to obtain from an arbitrary node distribution a partition of the total domain. This partition will be optimal in order to be used in a numerical method as the FEM and the number of operations to obtain this partition must be bounded with  $n^\beta$ , where  $n$  is the total number of nodes and  $\beta$  must be much less than 2. Furthermore, the partition must be unique for a given node distribution

## 2. THE DELAUNAY TESSELLATION

In order to better understand the new procedure, classical definitions will be introduced for 3 entities: Voronoï diagrams, Delaunay tessellations and Voronoï spheres.

Let a set of distinct nodes be:  $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \dots, \mathbf{n}_n\}$  in  $\mathbb{R}^3$ .

- a) The **Voronoï diagram** of the set  $\mathbf{N}$  is a partition of  $\mathbb{R}^3$  into regions  $V_i$  (closed and convex, or unbounded), where each region  $V_i$  is associated with a node  $\mathbf{n}_i$ , such that any point in  $V_i$  is closer to  $\mathbf{n}_i$  (nearest neighbor) than to any other node  $\mathbf{n}_k$ . See Figure 1 for a 2-D representation. There is a single Voronoï diagram for each set  $\mathbf{N}$ .
- b) A **Voronoï sphere** within the set  $\mathbf{N}$  is any sphere, defined by 4 or more nodes without any node inside. Such spheres are also known as empty circumspheres.
- c) The **Delaunay tessellation** within the set  $\mathbf{N}$  is a partition of the convex hull  $\Omega$  of all the nodes into regions  $\Omega_i$  such that  $\Omega = \bigcup \Omega_i$ , where each  $\Omega_i$  is the tetrahedron defined by 4 nodes of the same Voronoï sphere. Delaunay tessellations of a set  $\mathbf{N}$  are not unique, but each tessellation is dual to the single Voronoï diagram of the set.

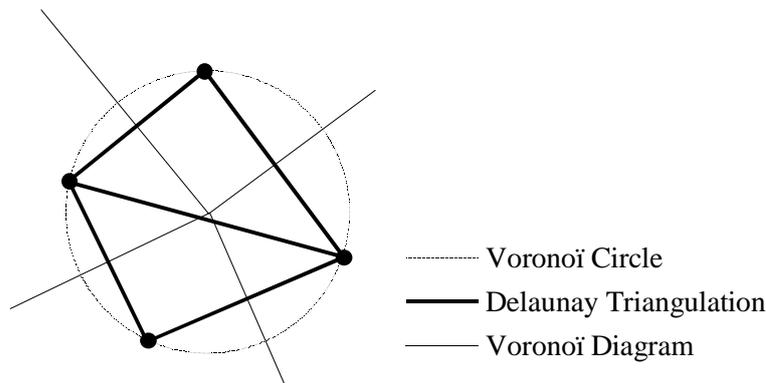


Figure 1: Voronoï diagram, Voronoï circle and Delaunay triangulation for a 4 node distribution in 2D.

The computing time required for evaluation of all these 3 entities is of order  $n^\beta$ , with  $\beta \leq 1.333$ . Using a very simple bin organization, the computation time may be reduced to near  $n$ .

The Delaunay tessellation of a set of nodes is non-unique. For the same node distribution, different tetrahedrations are possible. Therefore, a partition based on the Delaunay tessellation is sensitive to geometric perturbations of the node positions. On the other hand, its dual, the Voronoï diagram, is unique. Thus, it makes more sense to define a partition based on the unique Voronoï diagram than on Delaunay tessellations. In Figure 2 two critical cases of Delaunay instabilities are represented. One is the case of 4 nodes on the same circle and the other is the case of a node close to a boundary. In both cases, the Voronoï diagram remains almost unchanged.

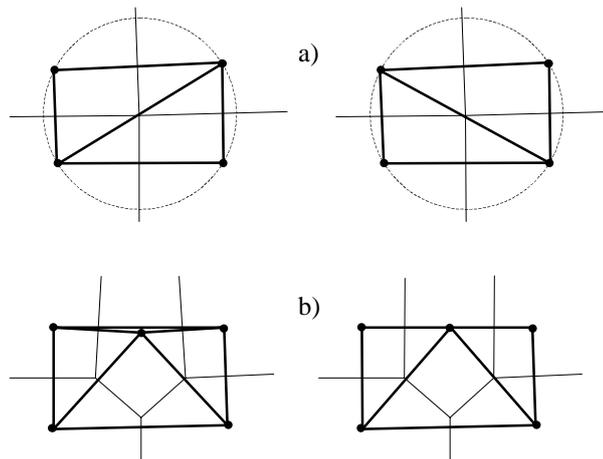


Figure 2: Instabilities on the Delaunay tessellation.  
 a) Four nodes on the same circle; b) Node close to a boundary.

Furthermore, in 3-D problems the Delaunay tessellation may generate several tetrahedra of zero or almost zero volume, which introduces large inaccuracies into the shape function derivatives to be used in a numerical method. This is the reason why a Delaunay tessellation must be improved iteratively in order to obtain an acceptable partition to be used in a numerical method. The time to obtain an acceptable partition via a Delaunay tessellation is then an unbounded operation, thus not satisfying the requirement expressed in the introduction.

### 3. THE EXTENDED DELAUNAY TESSELATION

From the point of view of the application of a domain partition to the solution of a numerical method, the best partition is that which the elements have:

- a) all its nodes on the same empty sphere (this is the concept of optimal distance between nodes), and
- b) the polyhedral elements fill the sphere as much as possible (this is the concept of optimal angle between faces).

The standard Delaunay tessellation satisfies both previous statements only for 2D problems. Unfortunately, the second statement is not necessarily satisfied in a 3D partition.

The drawbacks appear in the so-called “degenerated case”, which is the case where more than 4 nodes (or more than 3 nodes in a 2-D problem) are on the same empty sphere. For instance, in 2-D, when 4 nodes are on the same circumference, 2 different triangulations satisfy the Delaunay criterion. However, the most dangerous case appears only in 3-D. For instance, when 5 nodes are on the same sphere, 5 tetrahedra may be defined satisfying the Delaunay criterion, but some of them may have zero or almost zero volumes, called **slivers** (see Figure 3).

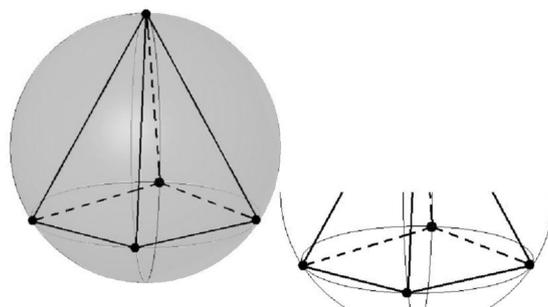


Figure 3: Five nodes on the same sphere and possible zero or almost zero volume tetrahedron (sliver) on the right.

In order to overcome the drawbacks referred to in the above paragraphs, a generalization of the Delaunay tessellation will be defined:

**Definition: The Extended Delaunay Tessellation** within the set  $N$  is the unique partition of the convex hull  $\Omega$  of all the nodes into regions  $\Omega_i$  such that  $\Omega = \cup \Omega_i$ , where each  $\Omega_i$  is the polyhedron defined by all the nodes laying on the same Voronoï sphere.

The main difference between the traditional Delaunay tessellation (DT) and the Extended Delaunay Tessellation (EDT) is that, in the latter, all the nodes belonging to the same Voronoï sphere define a unique polyhedron. With this definition, the domain  $\Omega$  will be divided into tetrahedra and other polyhedra, which are unique for a given node distribution, satisfying then one of the goals required in the Introduction.

Figure 4, for instance, is a 2-D polygon partition with a triangle, a quadrangle and a pentagon. Figure 5 is a polyhedron with all the nodes on the same sphere, which may appear in a 3-D problem.

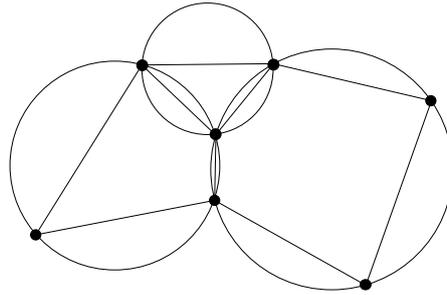


Figure 4: Two-dimensional partition in polygons. The triangle, the quadrangle and the pentagon are each inscribed on a circle

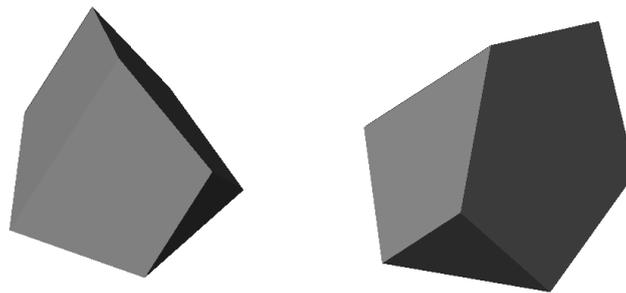


Figure 5: Eight-node polyhedron. All nodes are on the same sphere.

In order to avoid numerical problems, which may hide polyhedra with more than 4 nodes, the polyhedra are defined by all the nodes of the same sphere and nearby spheres. The proximity of the spheres is governed by a parameter  $\delta$  (see the Appendix).

The parameter  $\delta$  avoids the possibility of having near zero volume tetrahedra. When  $\delta$  is large, the number of polyhedra with more than 4 nodes will increase, and the number of tetrahedra with near zero volume will decrease, and vice versa.

Then, the idea of the Extended Delaunay Tessellation is to avoid the tetrahedral partition when more than 4 nodes are in the same empty sphere (or near the same). When this is the case, the best element to be used in a numerical method is the polyhedron formed with all these nodes.

The EDT allows the existence of a domain partition which: a) is unique for a node distribution, b) is formed without polyhedra with near zero volume, and c) is obtained in a bounded time of order  $n$ . Then, it satisfies all the goals stated previously.

It must be noted that until quite recently, all the mesh generators were limited to the generation of tetrahedral or hexahedral elements (or triangular and quadrangular in 2D problems). The reason of this limitation was the lack of any acceptable shape function to be used in other kind of geometrical elements. Nowadays, several acceptable shape functions for any kind of polyhedron are easily generated (see for instance references 3 and 4 or the Appendix II). These new shape functions, together with the Extended Delaunay Tessellation presented in this paper, gives an optimal marriage and a powerful tool to solve a large variety of physical problems by numerical methods.

#### 4. THE BOUNDARY SURFACE

One of the problems in mesh-generation is the correct definition of the domain boundary. Sometimes, boundary nodes are explicitly defined as special nodes, which are different from internal nodes. In other cases, the total set of nodes is the only information available and the algorithm must recognize the boundaries. Such is the case for instance, of the Lagrangian formulation in fluid mechanics problems in which, at each time step, a new node distribution is obtained and the free-surface must be recognized from the node positions.

The use of Voronoï diagrams or Voronoï spheres may make it easier to recognize boundary nodes. By considering that the node follows a variable distribution, with  $h(\mathbf{x})$  as the minimum distance between two nodes, the following criterion has been used:

*All nodes defining an empty circumsphere with a radius  $r(\mathbf{x})$  larger than  $\alpha h(\mathbf{x})$ , are considered as boundary nodes.*

In this criterion,  $\alpha$  is a parameter close to, but greater than one. Note that this criterion is coincident with the Alpha Shape concept<sup>[5,6]</sup>.

Once a decision has been made concerning which of the nodes are on the boundaries, the boundary surface must be defined. It is well known that, in 3-D problems, the surface fitting a number of nodes is not unique. For instance, four boundary nodes on the same sphere may define two different boundary surfaces, one concave and the other convex.

In order to avoid this undefined boundary, the boundary surfaces will be defined with:

*All the polyhedral surfaces having all their nodes on the boundary and belonging to just one polyhedron are boundary surfaces.*

The correct boundary surface may be important to define the correct normal external to the surface. Furthermore, in weak forms, the boundary surface is also important for a correct evaluation of the volume domain.

Nevertheless, it must be noted that in the criterion proposed above, the error in the boundary surface definition is bounded and proportional to  $h$ . This is the standard error of the boundary surface definition in a numerical method for a given node distribution.

#### 5. ELEMENT QUALITY INDICATOR

In order to evaluate the partition obtained with the EDT, compared with other partition as, for instance, DT, a polyhedron quality indicator will be defined.

A wrong mesh is a mesh which is not acceptable to be used in a numerical method. The main drawback of a Delaunay mesh is the presence of zero or almost zero volume polyhedra. In fact, the problem of almost zero volume is the existence of shape functions with infinite or almost infinite gradients. These infinite gradients deteriorate the numerical solution.

Classical definition of mesh quality will not be used here. The reason is that in this paper a node distribution is considered for which an optimal partition must be found without moving or removing nodes. The presence of nodes very close to each other in the node distribution is considered as a necessity of the physical problem to represent correctly a gradient in the solution and not as a wrong partition.

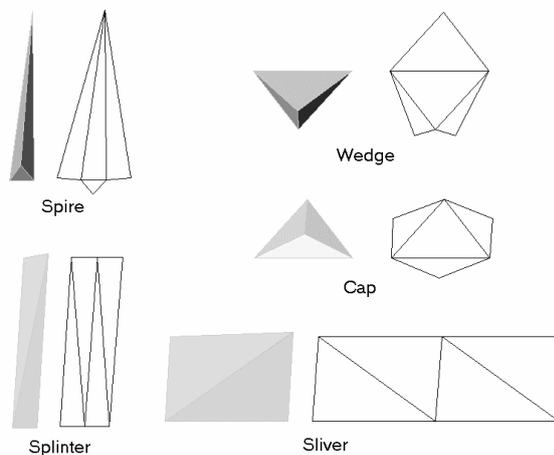


Figure 6: Perspective unfolding and naming of some bad-shaped tetrahedra.

For instance, tetrahedra currently called *spires* and *wedges* (see

Figure 6), will be considered elements of good quality because they are the best elements for a node distribution with the nearest nodes along one direction and more separated nodes in the other. These elements represent correctly the gradient “expected” in each direction to solve a particular physical problem. On the other hand, *slivers* or *splinters* as well as *caps*, will be considered wrong elements because they introduce shape functions with infinite or almost infinite gradients, which are not in agreement with the gradients expected for this node distribution.

In order to obtain a parameter to define which is a good or a wrong element to be used in a numerical method, but taking into account that the node distribution is fixed and introduced for physical needs, the gradient ratio will be defined as:

$$\gamma = \frac{\text{maximum gradient of the shape functions}}{\text{maximum gradient expected}} \quad (1)$$

The maximum gradient expected for the node distribution will be defined in each polyhedron as  $1/h_{\min}$ , where  $h_{\min}$  is the shortest distance between two nodes belonging to the polyhedron

The maximum gradient of the shape functions may be evaluated directly using the largest gradient modulus from the shape functions of the polyhedron  $|\nabla N_p|_{\max}$  (See the Appendix II for the definition of the shape functions  $N_p$ ).

The quality of a polyhedron is then numerically defined as:

$$\gamma = \frac{1}{h_{\min} |\nabla N_p|_{\max}} \quad (2)$$

It must be noted that in the EDT algorithm there is not any smoothing process in which element qualities must be evaluated many times. The gradient ratio  $\gamma$  is only defined here in order to verify the quality of the resulting mesh.

The ideal element will therefore have a large gradient ratio. This will have both: the required distribution on the Voronoï sphere and an acceptable shape function for numerical computations. On the other hand, bad elements have all their nodes on the same empty sphere, but the small  $\gamma$  value is indicating high shape function gradients capable to destroy the numerical solution.

Taking into account that the computer precision nowadays is of order  $10^{-16}$ , and also the results from the numerical test performed in the next section, a gradient ratio  $\gamma > 10^{-2}$  is recommended in order to accept a mesh for numerical computations.

## 6. NUMERICAL TEST

In order to check if a domain partition is acceptable or not to be used in a numerical method, a physical problem must be solved using a defined partition.

A cube of unit side, with an internal exponential source, has been used to validate the EDT.

The problem to be solved is the classical Poisson equation:

$$\nabla^2 u = f(x, y, z) \quad (3)$$

With the internal source :

$$\begin{aligned} f(x, y, z) = & (-2 k y z (1-y)(1-z) + (k y z (1-y) (1-z) (1-2x)^2)^2 \\ & - 2 k z x (1-z)(1-x) + (k z x (1-z) (1-x) (1-2y)^2)^2 \\ & - 2 k x y (1-x)(1-y) + (k x y (1-x) (1-y) (1-2z)^2)^2 \\ & (- e^{kxyz(1-x)(1-y)(1-z)} / (1 - e^{k/64})) \end{aligned} \quad (4)$$

The boundary condition is the unknown function  $u$  set equal to zero on all the boundaries.

This problem has the following analytical solution:

$$u(x, y, z) = (1 - e^{kxyz(1-x)(1-y)(1-z)}) / (1 - e^{k/64}) \quad (5)$$

Several node distributions have been tested with 125 ( $5^3$ ), 729 ( $9^3$ ), 4,913 ( $17^3$ ), and 35,937 ( $33^3$ ) nodes, with structured and non-structured node distributions. In all cases, the numerical solution was obtained using linear finite element shape functions for tetrahedral elements and polyhedral shape functions (as defined in the Appendix II) for polyhedral elements.

For the structured node distributions, the following procedure was used to generate the nodes: Initially all the nodes are in a regular cubic array with a constant distance  $h$  between neighbor nodes.

Then, each internal node has been randomly displaced a distance  $\rho h$  (with  $\rho \ll 1$ ) in order to have an arbitrary, but structured, node distribution. Surface and edge nodes are perturbed but remaining in the surface or the edge. Corner nodes were not perturbed. In this paper, the parameter  $\rho$  was fixed to  $10^{-6}$ .

The 3D non-structured node distribution was generated using the GID pre/post-processing code<sup>[7]</sup> with a constant  $h$  distribution. GID generates the nodes using an advancing front technique, which guarantees that the minimal distance between two nearby nodes lies between  $0.707 h$  and  $1.414 h$ .

### 6.1. Extended Delaunay tessellation versus Delaunay tessellation

It must be noted that in 2D problems, both node distributions generated as described before, structured and non-structured, will give a Delaunay partition with near-constant area triangles, which is optimal for a numerical solution. Nevertheless, this is not the case in 3D problems in which, even for a constant  $h$  node distribution, many zero or near-zero volume tetrahedra (slivers) will be obtained on a standard Delaunay partition<sup>[8]</sup>. Figure 7 shows, for instance, the presence of slivers on a structured eight-node distribution. Slivers may introduce large numerical errors in the solution of the unknown functions and their derivatives, which may completely destroy the solution.

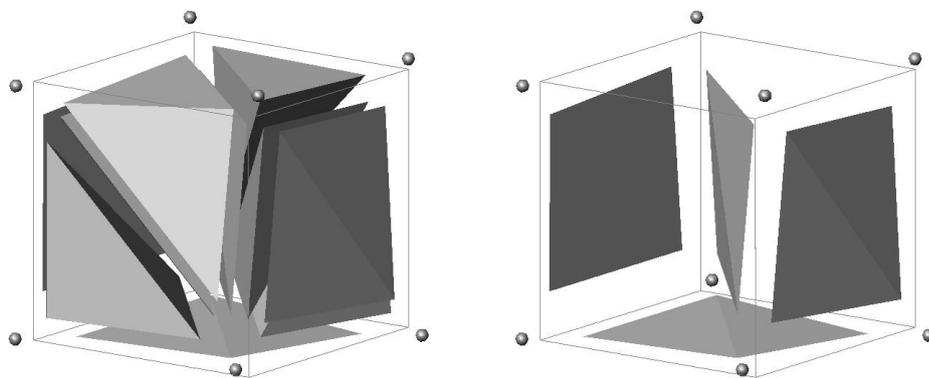


Figure 7: Presence of slivers in a Delaunay partition of a perturbed cube. Left: Tetrahedra produced by the Delaunay partition. Right: Slivers isolated.

In order to show this behavior and to show that the EDT eliminates this problem, the following tests were performed: for a fixed-node partition (e.g.  $17^3$  nodes) the  $\delta$  parameter (introduced in section 3 and in the Appendix I) was swept from 0 to  $10^{-1}$ . With  $\delta = 0$  (i.e. Voronoi spheres are never joined) the standard Delaunay tessellation (DT) is obtained. Larger values of  $\delta$  gives the Extended Delaunay Tessellation (EDT).

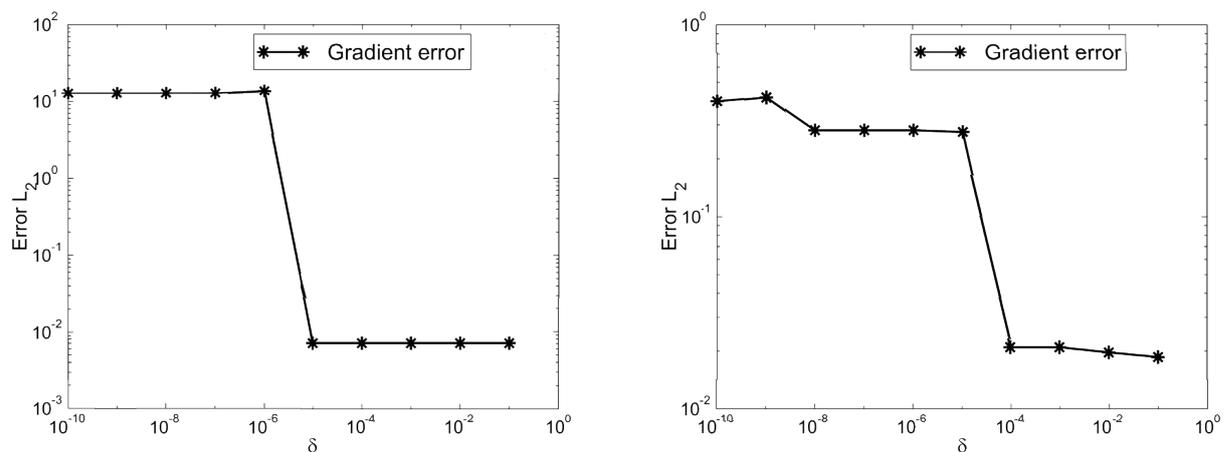


Figure 8: Cube with exponential source. Error of the derivative in  $L^2$ . Left: Structured node distribution. Right: Non-structured node distribution.

Figure 8 shows the error in  $L_2$  norm for the derivative of the solution of the 3D problem stated in equations (1) and (2). This has been done both for structured and non-structured distributions against the  $\delta$  parameter. It can be shown that in both cases the errors are very large ( $\sim 10^1$ ) for  $\delta < 10^{-6}$  and very

small ( $\sim 10^{-2}$ ) for  $\delta > 10^{-5}$ . Larger  $\delta$  do not change the results. Hereafter, the EDT will be considered for  $\delta > 10^{-4}$ .

This example is very important because it is showing that, for a given node distribution, a tetrahedrization using the standard Delaunay concept do not work. Mesh generators currently use edge-face swapping or another cosmetic algorithms to overcome the presence of wrong elements. All those operations are unbounded in computing time. The idea of joining similar spheres, even for a very small  $\delta$  parameter, solves this problem on a very simple way. The wrong tetrahedra are automatically joined to form polyhedra with optimal shapes.

The results of Figure 8 show also that  $\delta$  must be large compared with the computer precision (e.g.  $\sim 10^{-5}$  for a computer precision of order  $10^{-16}$ ) but also means that  $\delta$  is not a parameter to be adjusted in each example as results do not change by setting  $\delta$  larger than  $10^{-4}$ . In fact, all our examples were carried out with a fixed  $\delta = 10^{-2}$ .

All the polyhedra obtained with that fixed value of  $\delta$  had qualities  $\gamma > 10^{-2}$ , thus acceptable gradients and low error.

## 6.2. Good and wrong elements

In order to show the performance of the EDT to generate good elements, the gradient ratio of the node distribution of the previous example was evaluated.

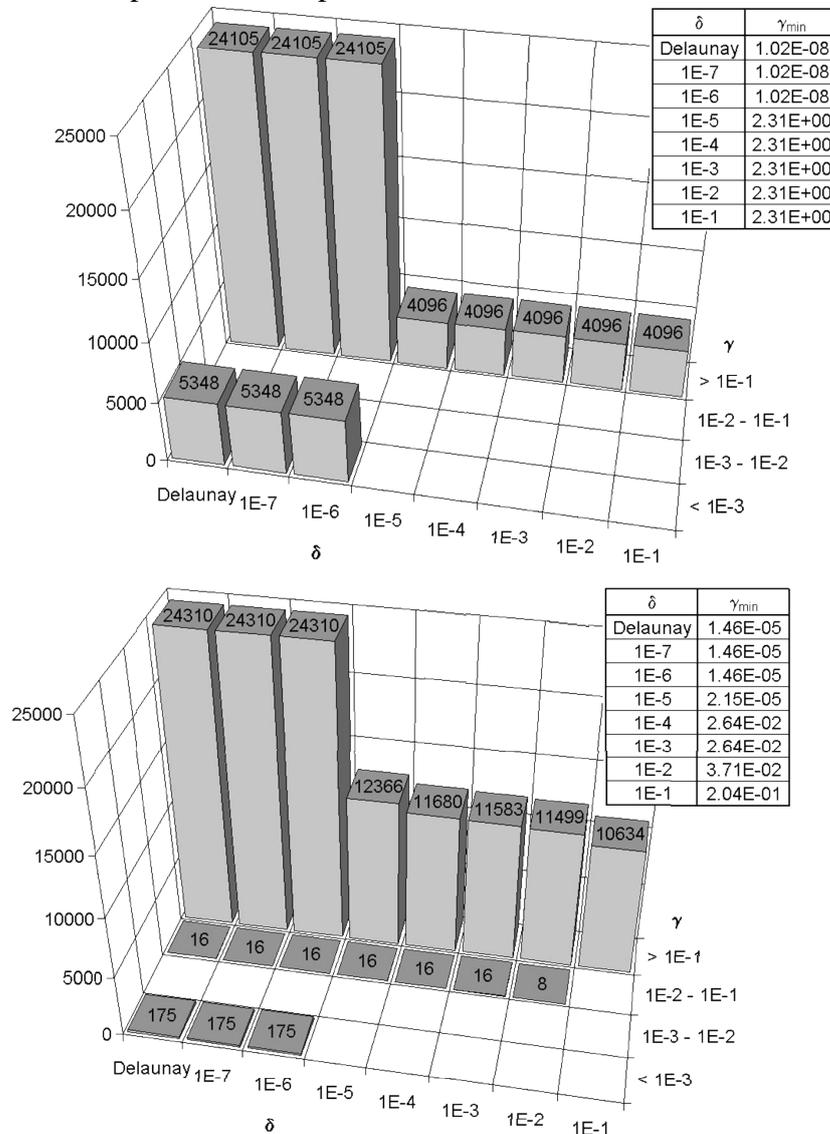


Figure 9: Number of polyhedra by volume ratio for different  $\delta$ . Top: Structured node distribution. Bottom: Non-structured distribution made by GID.

In the Figure 9, for a fixed  $\delta$ , the heights of the columns represent the number of elements having the same gradient ratio.

The importance of Figure 9 is to show that, by joining similar spheres, the best polyhedra are automatically built. For the standard Delaunay tessellation (and also for small  $\delta$  values), there are bad tetrahedra for both the structured and the non-structured node distribution. Increasing  $\delta$ , the slivers disappear and, for the structured case, all the elements become “automatically” hexahedra (cubes), which is the optimal tessellation for this node distribution. For the non-structured case, EDT insure a mesh without any sliver when  $\delta$  is set to any value larger than  $10^{-6}$ .

### 6.3. Convergence rate

Figure 10 shows the convergence of the above-defined example, when the number of nodes is increased from  $5^3$  to  $33^3$ . The upper plots show the error in  $L^2$ -norm, both for the function and its derivatives. All the graphics shows an excellent convergence rate. It must be noted that for all the non-structured node distributions tested (and also the structured ones for  $\rho = 10^{-6}$ ), the FEM with elements generated using a Delaunay tessellation gave totally wrong results, and even times ill-conditioned matrices were frequently found during the stiffness matrix evaluation.

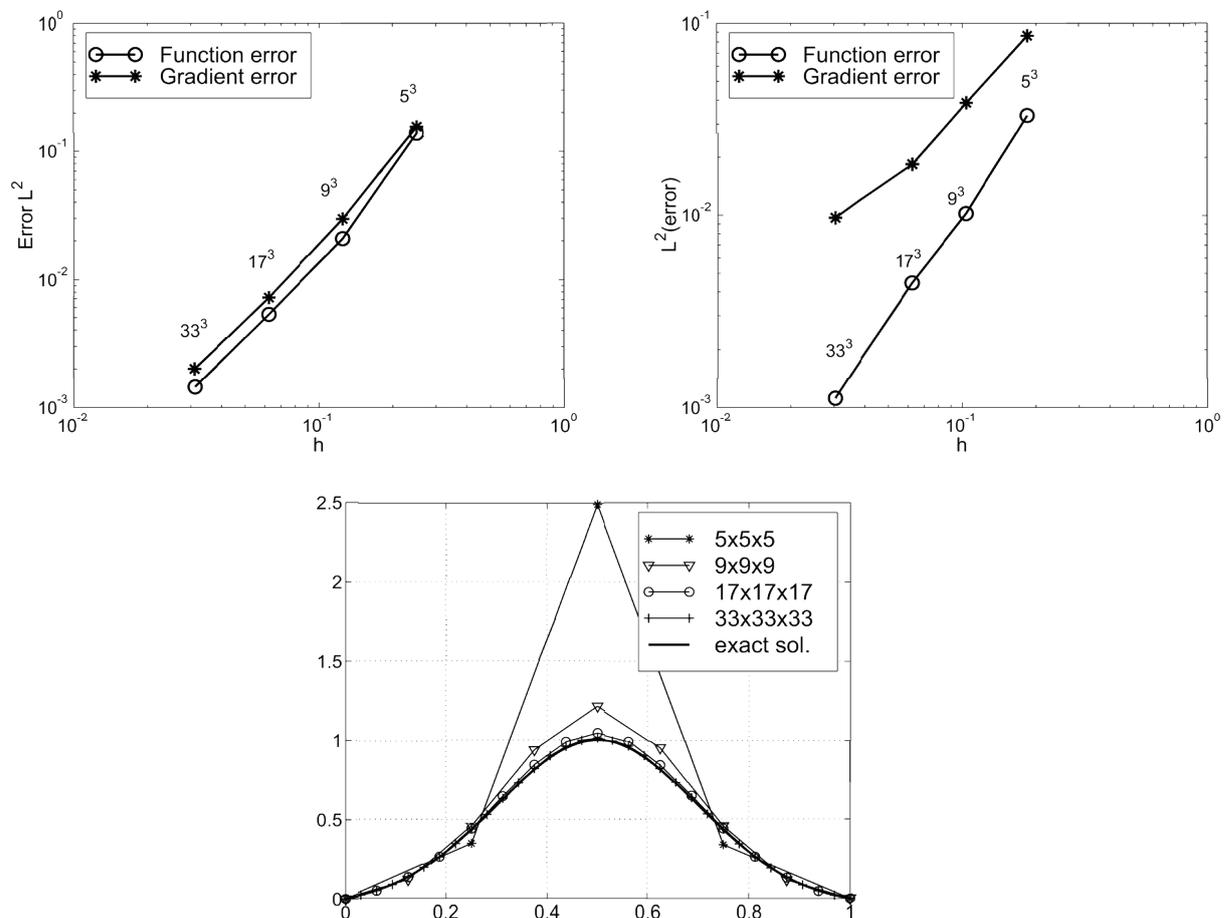


Figure 10: Cube with exponential source.

Convergence of the numerical solution and its gradient for different partitions.

Upper left: Structured node distribution. Upper right: Non-structured node distribution.

Below: Center-line solutions obtained with structured node distribution (the same results were obtained using non-structured node distribution).

#### 6.4. Computing times

In order to validate that the number of operation in the evaluation of the EDT partition is order near  $n$ , the computing time in a standard PC (Intel PIII 800Mhz) was analyzed. Figure 11 shows the time in seconds for both the structured and non-structured cases.

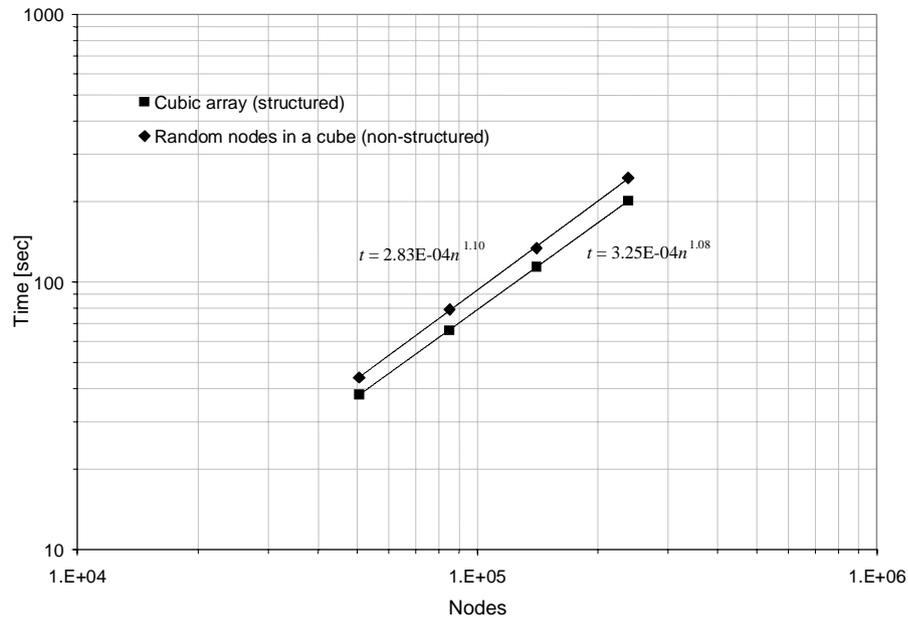


Figure 11: Time vs. number of nodes in a standard PC

A regression of the obtained results shows that the computing time is approximately:

$$t(\text{sec}) = 0.000325 n^{1.08} \quad \text{for structured meshes, and}$$

$$t(\text{sec}) = 0.000283 n^{1.10} \quad \text{for non structured,}$$

showing that the convergence exponent is even better than the 1.333 bound expected in a Delaunay tessellation.

#### 6.5. Three dimensional arbitrary geometry mesh

In order to show the performance of the method with in an arbitrary 3D geometry, a volume representing the vocal A was generated with distributions of  $n \cong 10^3$  and  $n \cong 10^5$  nodes, using the GID node generator with constant  $h$ . Figure 12. left shows, for instance, the boundary node distribution for  $n = 8452$  nodes.

Using the EDT algorithm described in section 3 with the boundary surface definition described in section 4 and an alpha shape parameter  $\alpha = 1.3$ , polyhedral partitions were found with external boundary surfaces represented in Figure 12.

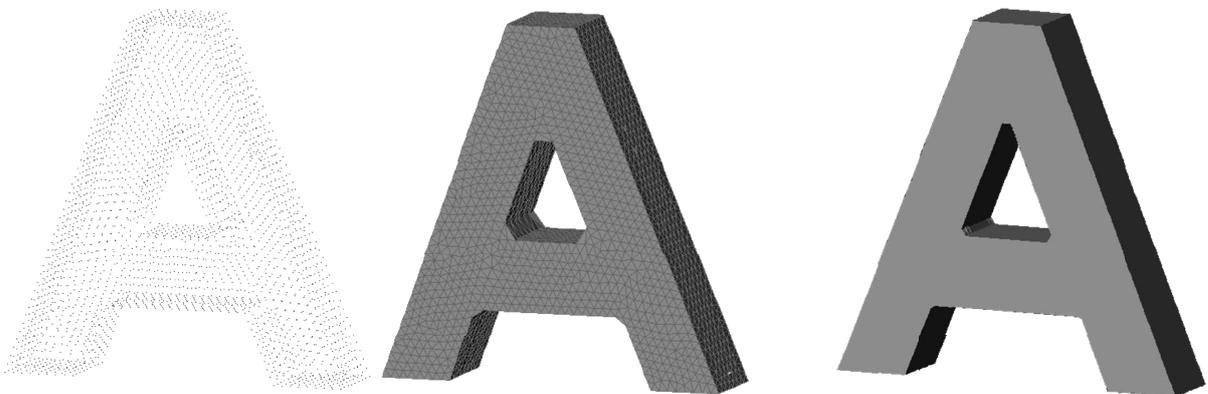


Figure 12: Three-dimensional arbitrary geometry.

Left: Boundary nodes from the point distribution generated by GID with constant  $h$  and 8452 nodes.

Center: Boundary surface for 8452 nodes. Right: Boundary surface for 106,947 nodes.

The following are the main characteristics polyhedral mesh made for the larger node set, compared with the tetrahedral mesh obtained via the standard Delaunay Tessellation.

	EDT	DT
nodes:	106,947	106,947
polyhedra:	497,424	599,934
tetrahedra:	430,262	599,934
slivers:	0	349
$\gamma_{\min}$ :	1.84e-2	2.34e-16

From a total of 106,947 nodes, the standard Delaunay Tessellation generates 599,934 tetrahedra, but 349 of them are slivers. On the other hand, EDT generate 430,262 tetrahedra and 67,162 polyhedra with more than 4 nodes. None of them have gradient ratio smaller than  $10^{-2}$ . Another interesting conclusion to take out from this example is that in the EDT mesh, 86,5% of the elements are tetrahedral and then, in these elements the definition of the shape functions (Appendix II) will be coincident with classical linear shape functions of the Finite Element Methods.

Finally, Figure 13 shows the slivers distribution in the mesh. Elements having a  $\gamma$  parameter smaller than  $10^{-2}$  have been plotted in black. Zero bad polyhedra have been found in the partition given by the EDT.

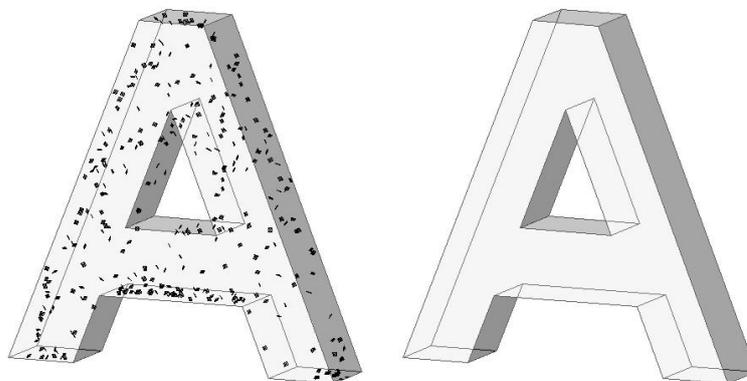


Figure 13: Wrong elements in the 106,947 nodes tessellation. Left: 349 bad tetrahedral (slivers) in the DT . Right: zero wrong polyhedra in the EDT.

## 7. CONCLUSIONS

For a given set of nodes with arbitrary 3D distribution and non-structured and variable distance between nodes, the method proposed gives a polyhedral mesh and a boundary surface with the following characteristics:

- The solution is unique for given set of parameters  $\delta$  and  $\alpha$ . The solution is not sensitive to small variation of those parameters.
- Each polyhedron has all his nodes on the same empty sphere (Optimal distance between nodes).
- Each polyhedron has an acceptable positive volume to be used in a numerical method. (The maximum gradient of the shape functions is of the same order of the expected gradient for a given node distribution).
- The boundary surface obtained may be concaves or convexes, and the correct definition depends on the local node distance between nodes  $h(\mathbf{x})$ .
- The computing time to achieve the mesh is bounded and of order  $n$ .
- The large majority of the polyhedral elements generated are tetrahedra (around 85% in an arbitrary node distribution). This allows the use of standard lineal finite element shape functions in all of them.

## APENDIX

### I) CRITERION TO JOIN POLYHEDRA

Consider two Voronoï spheres having nearby centers. See Figure 14 for a two dimensional reference.

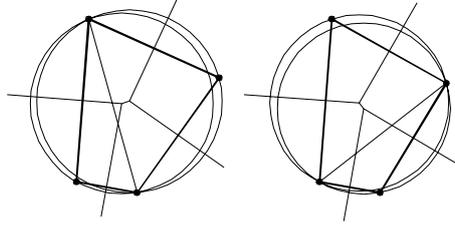


Figure 14: Four nodes in near-degenerate position showing the empty circumcircles, the Voronoï diagram and the corresponding discontinuous Delaunay triangulation.

As both Voronoï spheres are empty, they must satisfy the following relationship:

$$|r_2 - r_1| \leq \|\mathbf{c}_1 - \mathbf{c}_2\| \quad (6)$$

where  $r$  are the radii and  $\mathbf{c}$  the centers of the spheres.

Thus, two spheres are similar when their centers satisfy:

$$\|\mathbf{c}_1 - \mathbf{c}_2\| < \delta r_{rms}, \quad (7)$$

where  $\delta$  is a small non-dimensional value and  $r_{rms}$  is the root-mean-square radius.

Two polyhedra will be joined if they belong to similar spheres.

The algorithm finds all the 4-node empty spheres, and then the polyhedra are successively joined using the above criterion. It must be noted that when all the nodes of a polyhedron belongs to another polyhedron, only the last one is considered.

### II) SHAPE FUNCTIONS FOR ARBITRARY 3D POLYHEDRA

For any point within a polyhedron  $\mathbf{P}$ , there is a Voronoï cell  $\mathbf{V}(\mathbf{x})$  associated to the variable point  $\mathbf{x}$  in the Voronoï tessellation of the set  $\mathbf{P} \cup \{\mathbf{x}\}$ .

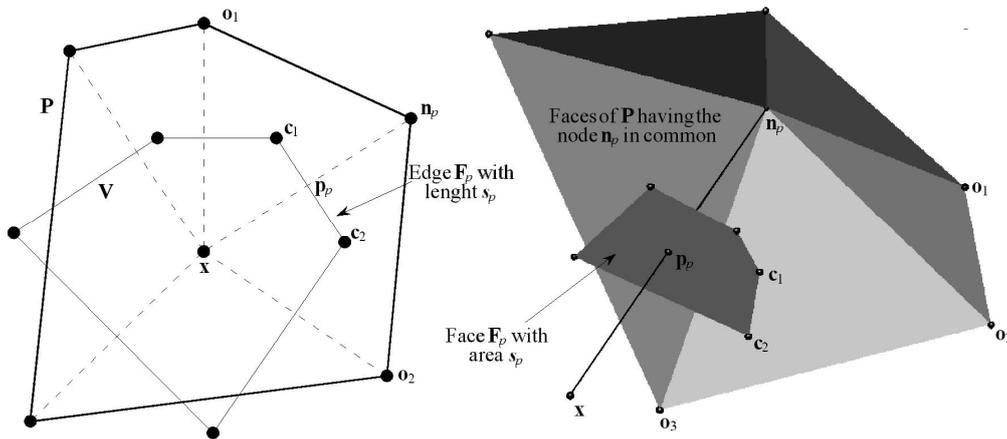


Figure 15: Elements defining 2D and 3D shape functions.

Figure 15 shows that every node  $\mathbf{n}_p \in \mathbf{P}$  has a corresponding face  $\mathbf{F}_p$  of  $\mathbf{V}$ , which is normal to the segment  $\{\mathbf{x}, \mathbf{n}_p\}$  by its midpoint. This is because  $\mathbf{V}$  is the set of points closer than  $\mathbf{x}$  than any other point.

Defining the functions:

$$\phi_p(\mathbf{x}) = s_p / \|\mathbf{n}_p - \mathbf{x}\| = s_p / h_p, \quad (8)$$

as the quotient of the Lebesgue measure  $s_p$  of  $\mathbf{F}_p$  and the distance  $h_p$  between the point  $\mathbf{x}$  and the node  $\mathbf{n}_p$ . The shape functions are:

$$N_p = \phi_p / \sum_q \phi_q, \quad (9)$$

These functions automatically satisfy the partition of unity property:

$$\sum_p N_p = 1. \quad (10)$$

## REFERENCES

- 1 Lohner R Progress in grid generation via the advancing front technique. *Engineering with Computers* 1996, **12**: 186-210
- 2 George, P. L. Automatic Mesh Generation, Applications to Finite Methods. J. Wiley & Sons, New York, 1991, ISBN 0-471-93097-0
- 3 Belikov V and Semenov A. Non-Sibsonian interpolation on arbitrary system of points in Euclidean space and adaptive generating isolines algorithm. *Numerical Grid Generation in Computational Field Simulation, Proc. of the 6<sup>th</sup> Intl. Conf.* Greenwich Univ. July 1998.
- 4 Idelsohn S, Oñate E, Calvo N and Del Pin F. The meshless finite element method. *Submitted to International Journal for Numerical Methods in Engineering.*
- 5 Edelsbrunner H and Mücke EP. Three-dimensional alpha shapes. *ACM Transactions on Graphics* 1994; **13**: 43-72.
- 6 Akkiraju N, Edelsbrunner H, Facello M, Fu P, Mücke E and Varela C. Alpha Shapes: Definition and Software. *Proceedings of the 1st International Computational Geometry Software Workshop* 1995; pp.:63-66, url: <http://www.geom.umn.edu/software/cglist/GeomDir/shapes95def/>
- 7 GID The personal pre and post processor, CIMNE, Barcelona, 2002. url: <http://gid.cimne.upc.es>
- 8 Edelsbrunner H and Damrong G. An Experimental Study of Sliver Exudation. *Proceedings, 10th International Meshing Roundtable*, Sandia National Laboratories, pp.307-316, October 7-10 2001.