



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 182 (2000) 371–378

**Computer methods  
in applied  
mechanics and  
engineering**

www.elsevier.com/locate/cma

# All-hexahedral element meshing: Generation of the dual mesh by recurrent subdivision

Nestor A. Calvo, Sergio R. Idelsohn \*

*Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), Instituto de Desarrollo Tecnológico para la Industria Química (INTEC), Universidad Nacional del Litoral (UNL), Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Güemes 3450 – (3000) Santa Fe, Argentina*

Received 1 September 1998

## Abstract

The domain geometry is defined by means of a closed all-quadrilateral mesh. The outer mesh imposes very strong restrictions on the possible connectivities between the inner hexahedral elements. Following the guidelines of the outer topology, the inner one is almost entirely defined. Several ways may be decided for certain configurations, some of them requiring special considerations in order to achieve a valid FEM mesh. The process is entirely performed by constructing the (graph theoretical) dual of the hexahedral mesh, this means no metric information is handled until the final (positioning and smoothing) steps. The essential steps of this scheme are described by means of examples. © 2000 Elsevier Science S.A. All rights reserved.

**Keywords:** Hexahedral; Non-structured; Mesh; Generation; Dual; Finite Elements

## 1. Introduction

In the present there are a few all-quadrilateral plane and non-plane 2D-mesh generators [1–3] and many CAD programs readily available. On this ground, we have decided to start from a closed quadrilateral mesh. This mesh is the boundary of the domain and shows the desired sizes of the elements, at least near the domain boundary.

We present a method that, given a few topological constraints, produces a non-structured all-hexahedral mesh. It will be explained by means of a simple 2D example and then extended into 3D.

This paper presents the method and the main features of the software that have been implemented up to now; it is an extension of previously presented advances [4].

## 2. Recipe for a planar quadrilateral mesh

Fig. 1 was hand-made and will be used in order to explain the main features of the topology of a 2D, all-quadrilateral mesh. The actual 3D method of generation will be presented later in the paper.

We start (a) from a discretized boundary, in this case a non-simply-connected 1D mesh.

\* Corresponding author. Present address: IV WCCM Chairman, Edificio C-1, Campus Nord-UPC, E-08034 Barcelona, Spain. Tel.: +54-42-556673; fax: +54-42-550944.

E-mail addresses: ncalvo@arcride.edu.ar (N.A. Calvo), rnsergio@arcride.edu.ar (S.R. Idelsohn).

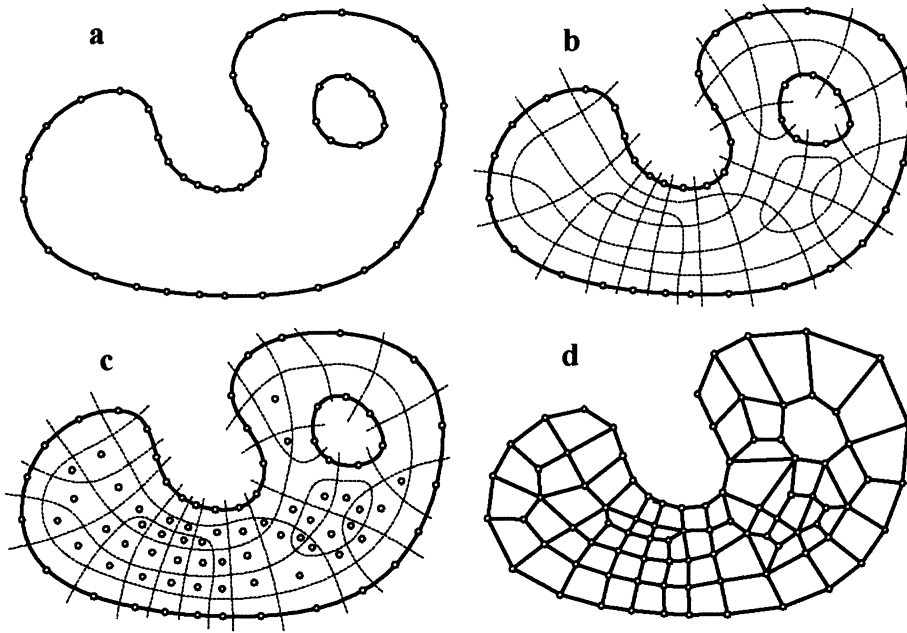


Fig. 1. Example: meshing a planar region.

Crossing the boundaries near the middle point of each segment we trace lines that intersect each other and with the boundary at near-right angles. The lines must intersect the domain limits twice, or must be closed and inner lines. As is shown in b, the intersections subdivide each line in almost equal segments. The lines may go by any way, joining which ever pair of exterior segments.

To each region formed between the lines, we assign an interior point, near the centroid, in step c (Fig. 1) The exterior regions traverse the domain boundaries, and the corresponding points (nodes) are the given ones.

Tracing a segment between neighboring regions (d) the mesh becomes apparent.

We have hand-made a mesh by tracing its 'dual' graph.

The concept of dual throughout this paper is a  $D$ -dimensional extension of the graph theoretical duality between a graph and a map: to each  $d$ -dimensional element of the mesh, there is a corresponding  $(D - d)$ -dimensional one of the dual and vice versa.

The quality of the elements is related to the previously unexplained perpendicularity and equipartition of the lines. Strictly speaking, the input and output segments of the lines crossing the boundary, must have half the length of the inner ones, because there is only one half of each exterior region inside the domain. More generally, one can divert from equipartition in order to control the mesh density. With the same criterion, one can trace (as is shown) inner closed lines. In fact, this will be the method used to accomplish with the specified density.

If we look at one entire dual line with its attached quadrilateral, the line cuts the thread of elements in a 1D segment mesh. Every quad of the thread is followed by another one or it meets the domain frontier.

This process shows a simple demonstration that the boundary of a quadrilateral mesh must 'be even' i.e. to have an even number of segments: every line that crosses the boundary does so at two points. When, as in the example, the domain has holes, the sum of all its connected limits must be even, but they may be individually odd in number.

Every inner closed line borders a submesh, and has automatically the parity of the sum of its interior borders. If there are no internal borders, it is even. This fact will be used later, in the 3D case.

### 2.1. Some warnings

In an FEM mesh, two neighboring elements may share sub-elements, for example two hexahedra may share a face, but they cannot share another face, or a segment or node that are not part of the shared face.

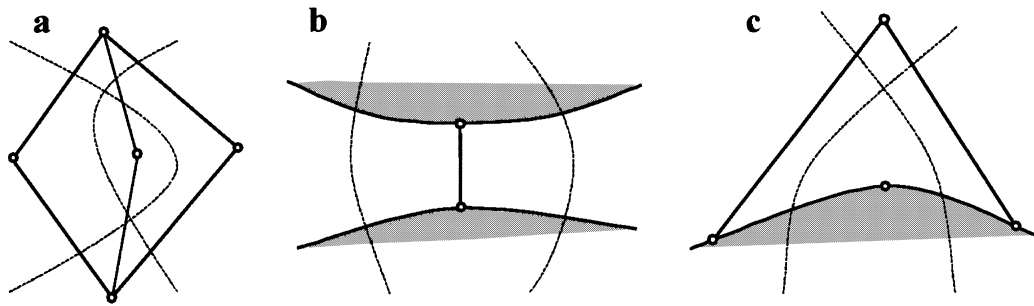


Fig. 2. Digon, two-node region and a malformed element.

This also forces the shared sub-element to be complete: if the two hexahedra of the example are sharing two segments they must belong to a complete shared face.

This condition may be expressed as *'the higher dimensional shared sub-element must be unique'*. In terms of the dual, surprisingly, the condition is identical.

Thus, the minimal region of the dual is triangular and we cannot admit (the frequently appearing) 'digonal' regions with two segments sharing its two endpoints in its boundary.

Fig. 2a shows these kinds of 'monsters'. They are topologically correct in the sense that they give quadrilateral but they are not suitable mesh elements. Even so, such two segment-sharing elements are frequently seen at the tips of acute forms in non-planar 2D meshes.

Another problem is shown in Fig. 2b there are two nearby zones of the boundary (the exterior appears hatched). The dual lines define a single region with two nodes, a situation that will give us an invalid mesh. We may say that each dual line must be intersected at least once. This avoids the related problem of lines going through adjacent external segments without crossing other lines.

As the generator will work with only the topology of the mesh until the final steps, some angular malformations may appear at concave or straight border edges, giving elements with inner angles equal to or greater than  $180^\circ$ . In the 2D case, it can be found (Fig. 2c) when two dual segments starting from adjacent boundary 1D elements meet at its inner vertex.

In order to avoid one-node regions and angular errors, we will not make elements with more than one higher sub-entity on the boundary. This can be shown in 2D as making an inner 'copy' or 'offset' of the entire border as a dual closed line, the 3D case is the same thing. This kind of onion layer means one  $n$ D inner element at each of the  $(n-1)$ D elements of the boundary.

We have explained the planar situation. It is obvious how to apply these concepts to a 2D curved-surface and, if it is a closed surface, the dual lines are closed too.

### 3. Extension to 3D hexahedral meshes

In the 3D case, the boundary is a set of closed quadrilateral meshes, with all its (exterior) dual lines closed too.

The internal lines traverse threads of hexahedral elements. Here, we have 2D nets of dual lines forming internal dual surfaces.

Fig. 3a shows a hexahedral mesh in the interior of a triangular dipyrmaid. The elements attached to one of the 2D nets were extracted and shown separately and magnified. The inner 2D mesh and the effect of the 'onion layer' or 'offset' of one hexahedron for each exterior quadrilateral element can be seen.

In b, a mesh of a submarine-like body displays the exterior lines, which are the outer limit of two intersecting 2D meshes, it also shows the line at which these dual surfaces meet as a 1D thread of elements. For the sake of simplicity, we have chosen two very simple intersecting surfaces, but there are others, more complex, that can be followed visually in the same mesh.

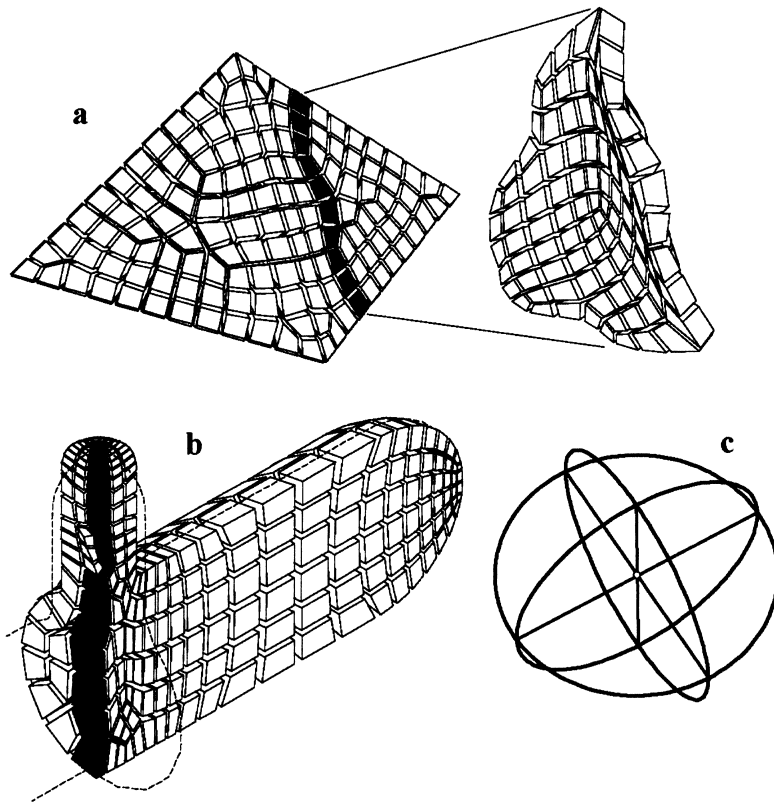


Fig. 3. Interior topology of all-hexahedra meshes.

The key of the presented process, and the key in order to understand the topology of all-hexahedral meshes, is Fig. 3c. There are three mutually intersecting circles representing the simplest form of lines from the dual of the boundary mesh.

These lines are the boundaries of internal surfaces that intersect pair wise in interior lines, the mutual intersection point is dual to a *necessary* single hexahedron.

This is also the main difference with other methods that, directly or not, have to do with the dual of the mesh [5–7]: *the given exterior mesh imposes a minimal set of elements* that must be found in the final interior mesh.

There must be one inner quadrilateral mesh for each exterior dual line. Each mesh in turn is intersected by other meshes, being the intersections 1D segment meshes whose duals are the inner lines.

It is possible, and sometimes natural, for a single inner quadrilateral mesh to be bordered by more than one external dual line; or to be closed without any boundary, like the onion layer.

This scheme clearly shows that the term ‘local’, in hexahedral meshing, must be treated with great caution. One cannot delete nor add a single element of the mesh like in quadrilateral or even tetrahedral meshes.

Those operations for mesh-refinement or quality-improvement, which does so by changing the mesh connectivity, must be done with entire surfaces. One may add or remove entire surfaces, or may make two of them coalesce like bubbles or change their internal course, also changing the set of internal intersections. All these operations must be done with surfaces, and among them.

#### 4. The generation method

One can consider any 2D mesh as a piece of a completely closed one. If the mesh has no holes, or if all of them are even, the closed mesh can be mapped over a sphere. In general, the supporting surface has as many holes (genus) as pairs of odd borderlines are there in the mesh.

We will deal, at this time, only with ‘spherically’ closed meshes without inner holes, mainly because the minimal set of surfaces is well defined and they have a single or simply connected boundary. If there were holes, cutting meshes would have to be given in order to reduce to zero the border genus.

In the spherical case, as can be shown in Fig. 4, any closed external dual line separates the boundary into two caps (b and c), each one with a piece of every line that crosses the ‘cutting line’.

The inner surface that belongs to the cutting line, divides the spherical mesh into two pieces, so closing the external caps with the same internal mesh (d), splits the problem into two simpler ones, b and c.

In order to obtain the inner closing mesh we made its dual by joining the ends of the trimmed crossing lines. This is accomplished like a mapping or projection of one of the caps into the surface, in the above example the cap labeled c was selected. The selection is made without testing because the two possibilities give valid outputs.

The cutting line is conserved in both halves but marked as used.

This splitting process is applied recursively until there are no more cutting lines. The fragments of used lines remain but they are not taken into account when mapping one cap into the inner surface. The figure shows one of these sequences in a, b, e, f and g.

The remainder (g) of that process is a kind of atom composed of single segments of the used lines. This atom is a 3D solid region whose main feature is that all of its vertices are trihedral. It is dual to a node of the mesh and its nodes are duals of the set of hexahedra sharing that node.

As we know the vertex set composing each line and solid, it is easy to translate the information into hexahedral elements and connectivities of the mesh.

#### 4.1. 3D warnings

One of the main advantages of the method is that, as it imposes the minimal mesh, the problem is self-solved. Well this is not the general case, the external lines are not as beautiful as in Fig. 3c or Fig. 4. Of the many difficulties found, all but one have been solved successfully.

- *Digons*: appears elsewhere when mapping the caps without the previously used lines. Its existence is easily tested and they are:
  - (1) not produced, if they are formed when two mapped lines cross twice in consecutive points, the points are not mapped and the inner lines become separated, or
  - (2) eliminated when they are formed between an inner line and the cutting line, and there are no other inner lines helping, it is surrounded with a spherical dual surface, making eight (otherwise unnecessary) nodes. Fig. 5 displays the 2D equivalent of the process.
- *Multiple intersections*: we stated that two surfaces meet at a line, in case the two external lines intercept more than twice there may be many lines. In this case, it is not imposed to join the pairs in a preferred manner. The mapping method successfully solved this kind of problem: following the lines in one cap we

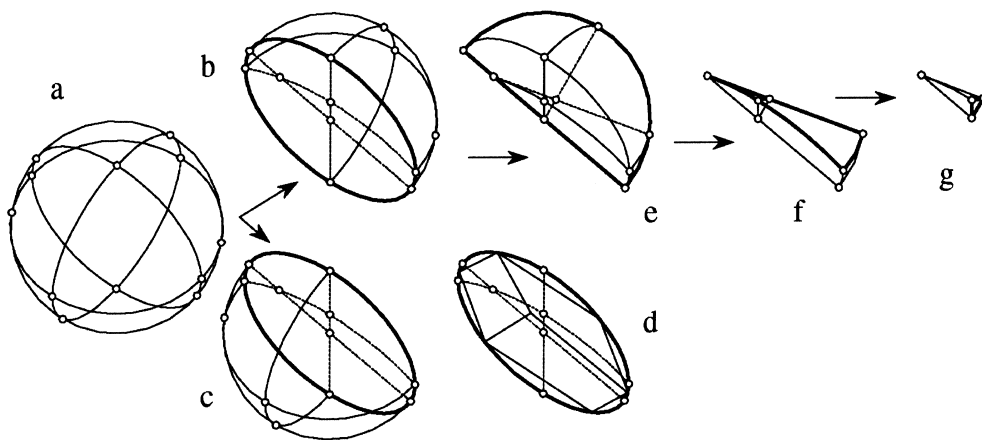


Fig. 4. Subdivision and mapping.

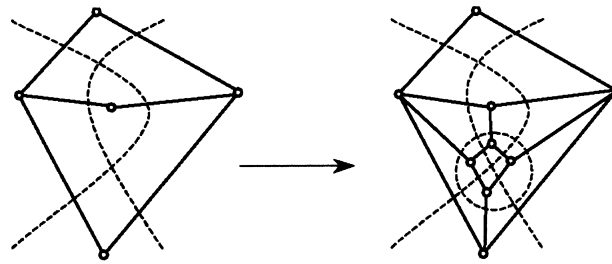


Fig. 5. Elimination of a digon.

determine which pair of points must be joined with a new line. The process develops valid meshes regardless of the chosen mapping-cap. At the mapped cap, those lines which are crossing many times the cutting line, split into pair wise-intersecting simple lines. In the other cap, the problem should be at worst as the preceding one.

- *Mesh quality*: It is impossible to know whether two lines are the non-simply-connected boundary of an internal cylindrical surface, or two boundaries from two cap-like internal surfaces. The method ‘trends’ to choose the second approach; thus giving larger elements and requiring further refinement in order to meet the desired element sizes. Currently, we are interested in obtaining a topologically valid mesh, we select then a good position for each node, but the surface operations needed to refine the mesh, changing its connectivity, are not yet implemented.

Now the bad news:

- *Self-intersections*: the lines may also self-intersect, severe cases like the *single line* shown in Fig. 6, are not yet solved.

Self-intersected (SI) lines are not chosen as cutting lines. An external SI line defines an SI internal surface with lines joining the external self-intersections. They impose a very intricate mapping in which the caps are not well defined and the surface lines of SI appear at different places being the same.

This means that the total number of given external SI's must be necessarily even to be compatible with an internal all-hexahedral mesh. If they are not individually evenly self-intersecting, then they define non-simply-connected boundaries of internal non-simple surfaces.

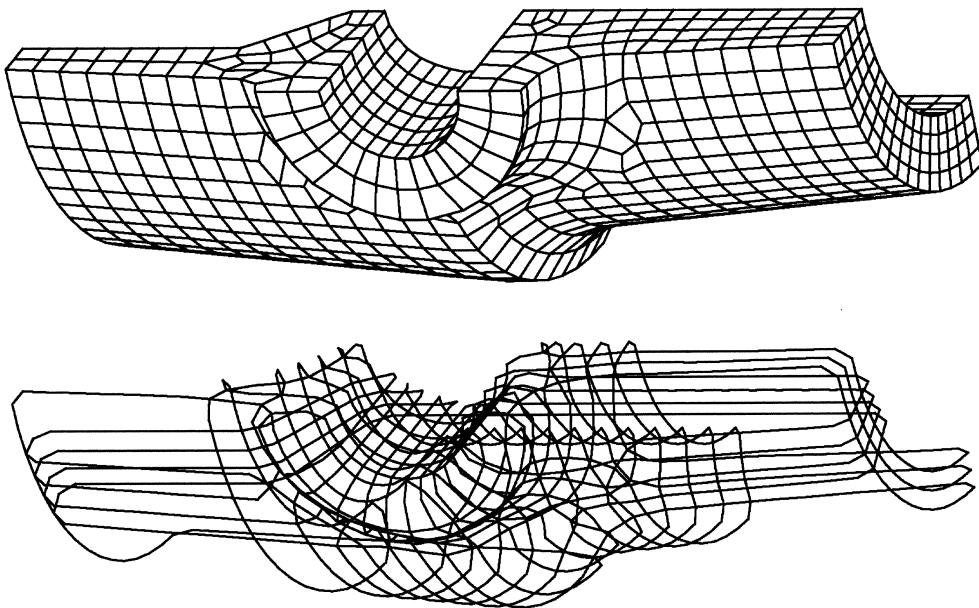


Fig. 6. Self-intersecting exterior line.

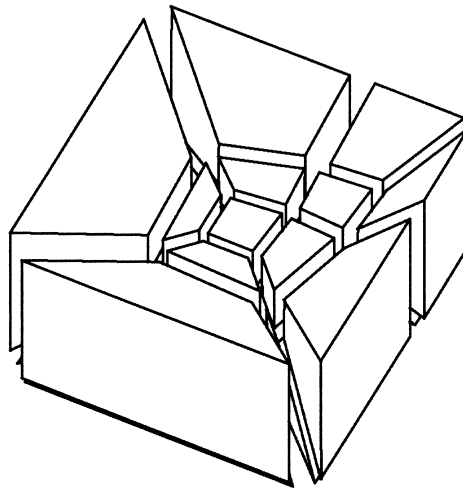


Fig. 7. Solved test example with auto-intersecting lines.

We would like to acknowledge Jordi Mola in helping us to realize that an even number of quadrilateral over a sphere implies automatically an even number of SI. This is because of the automatic parity of lines, and the fact that each SI adds ‘one’ element. In fact, this is a bi-directional assertion; also, an odd-meshed sphere implies unpaired self-intersections in the dual. Yet, we are not able to assert that this (parity) condition will suffice to be compatible with an internal all-hexahedral mesh but it is obviously a necessary condition.

There are some examples giving good meshes when the cutting (non self-intersecting) lines cut these into small pieces and the problem is disentangled. Fig. 7 shows a simple test example with some elements removed. There one can easily follow the self-intersected surface.

In the present form of the implementation, we only accept evenly SI lines; even so, we have not a solution for every case. This imposition is easily accomplished by means of the help of the program. It informs where the odd SI lines are, and the user must add or remove a few elements (into 2D it is very easy!) in order to obtain an acceptable exterior mesh.

In case there is no solution, the only way is to model the external mesh is by hand, raising the chance for the SI lines to be sliced. This requires an expert user, situation that is not acceptable for us, so we are trying to solve this problem.

We attempted to subdivide these lines into its loops but it did not work out. Now we are attempting to tackle the problem by means of strategically added simple temporary lines that will cut the offending ones, but removing those lines (and joining the solid regions) proved to be very cumbersome.

## 5. Conclusions

We have made many remarkable advances in the topic of all-hexahedral meshing. However, there is a lot of work to do. From now on, an expert user can give a mesh for many problems that were not approachable before.

The subdivision and mapping have proven to be very fast as only integral programming is involved.

Pre-positioning the nodes of the dual [4] is as inaccurate as fast but the slower repositioning of the final mesh by standard methods is secure.

Some more work is required in order to defeat the self-intersections problem, and that is where we are working now.

The next step will be the refinement and smoothing with connectivity change, but this problem seems much easier using the described technique of adding and intermixing surfaces.

As a non-discarded alternative, there is the possibility of meshing the exterior surface in an ad hoc way, in order to avoid excessive self-intersecting. If a well-shaped exterior mesh imposes a poor interior, some equilibrium must be better. Nevertheless, it seems impractical at this stage.

## References

- [1] J. Mola, J. Sarrate, A. Huerta, Extensió d'un mallador de quadrilàters per a treballar amb superfícies parametritzades, Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports. Universitat Politècnica de Catalunya, Spain, Report MA008/Set 1997.
- [2] T.D. Blacker, M.B. Stephenson, Paving: a new approach to automated quadrilateral mesh generation, *Int. J. Num. Meth. Eng.* 32 (1991) 811–847.
- [3] R.J. Cass, S.E. Benzley, R.J. Meyers, T.D. Blacker, Generalized 3-D paving: an automated quadrilateral surface mesh generation algorithm, *Int. J. Num. Meth. Eng.* 39 (1996) 1475–1489.
- [4] N. Calvo, S. Idelsohn, Generador automático de mallas de hexaedros: presentación del método y avances en la implementación, *Mecánica Computacional*, vol. XVI, Asociación Argentina de Mecánica Computacional, 1996.
- [5] M.A. Price, C.G. Armstrong, Hexahedral mesh generation by medial surface subdivision: part I. Solids with convex edges, *Int. J. Num. Meth. Eng.* 38 (1995) 3335–3359.
- [6] M.A. Price, C.G. Armstrong, Hexahedral mesh generation by medial surface subdivision: part II. Solids with flat and concave edges, *Int. J. Num. Meth. Eng.* 40 (1997) 111–136.
- [7] T. Tautges, T. Blacker, S. Mitchell, The whisker weaving algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes, *Int. J. Num. Meth. Eng.* 39 (1996) 3327–3349.