ELSEVIER

Comput. Methods Appl. Mech. Engrg. 141 (1997) 335–354

Computer methods in applied mechanics and engineering

3D mesh adaptation. Optimization of tetrahedral meshes by advancing front technique

A. Rassineux

ERIN-ESSTIN, Université Henri Poincaré, Nancy 1, 2, rue Jean Lamour, 54500, Vandoeuvre-les-Nancy, France

Received 4 January 1996; revised 15 May 1996

Abstract

The objective of the paper is to present an automatic mesh generator which builds a mesh which respects the element densities prescribed as a result of an error sensitivity analysis. The desired new mesh is fully characterized by the distribution of the refinement levels on the initial mesh. Therefore, the most important criterion in the choice of a mesh generator comes from the ability to produce a mesh with imposed density while meeting 'acceptable' shape quality requirements. In this spirit, a new method for tetrahedron mesh generation and optimization with respect to both a shape and a size criterion is presented. The advancing front technique is used to mesh the whole volume. Then, optimization procedures make astute local use of the algorithm used to mesh the complete model. The process is iterative. The corresponding method involves extracting sub-volumes or shells from the tetrahedron mesh volume, which are then remeshed by an advancing front technique to improve their quality. The sub-volumes are constructed by determining the set of tetrahedra which touch the same node, edge or face. Local transformations are extended to non-convex sub-regions which gives a higher quality mesh. Industrial examples of relatively complex volumes are given, demonstrating that a high quality and optimized mesh can be obtained by the proposed method.

1. Introduction

One of the problems of the Finite Element Method is that the accuracy of the analysis is highly dependent on the density of the mesh. If the discretization in regions of high stress gradients is not convenient, the difference between the finite element solution and the exact one can be significant. Therefore, in order to control the quality of the analysis, error estimation procedures can be applied to provide information on optimal mesh density [1–4]. The process involved in mesh adaptation techniques is iterative. Once an initial coarse mesh is created, a first solution is obtained, and an error sensitivity analysis can be performed in order to calculate the density of the optimal mesh. This initial coarse mesh, often denoted as 'natural mesh', is the result of the mesh generator when no control parameter is specified. In that particular case, local densities of elements are controlled by the densities of the surface triangles and the shape of the elements can be improved if necessary. When the densities of the optimal mesh have been computed, a new surface mesh which respects the prescribed density is generated and the last requirement is to mesh the volume with respect to both size and shape of the elements.

Many approaches have been presented in a 2D context in which the desired size on surfaces can be obtained by subdivision [5] or remeshing [6,7] but very few direct approaches have been extended successfully to 3D models [8]. The most difficult aspect is the problem of filling the volume while taking account of the density inside the domain and while meeting 'acceptable' shape quality requirements. A number of publications on the 3D advancing front method [9–11] have been presented but the extension of the advancing front technique to optimization purposes with respect to a shape or a size criterion has to our knowledge never been proposed. It is on the basis of these considerations that we introduce a new 3D meshing approach, that can be used for both mesh generation and optimization purposes. We will concentrate essentially on mesh optimization procedures.

0045-7825/97/$17.00
PII S0045-7825(96)01116-4

2. The advancing front technique

The main steps and difficulties of the advancing front technique are recalled hereafter. 2D advancing front methods have proved their worth. 3D frontal methods have been the subject of fewer publications than Delaunay type approaches [12–16].

A plausible explanation for this is that the frontal method is not based on theory, but merely on strategies, convictions and the intuition of the developer. Delaunay type approaches are founded on a demonstrable mathematical base [16]. Furthermore, the causes of failure of the advancing front technique are both numerous and difficult to identify. Development costs for 3D approaches are high, while their empirical character and many unsolved aspects discourage developers from selecting this type of method.

Boundary data for 3D triangulation comprise a set of triangular facets. Lo [17] has proposed a method for determining automatically closed boundary surfaces, and checking the orientability and closure of these surfaces. The surfaces are then classified as exterior and interior boundary surfaces, and each triangular element is given a correct orientation. At the start of the process, the front is equal to the set of triangular elements forming the boundary. The algorithm [9–11,18,19] is iterative and can be summarized as follows:

Provided the front is not empty:

(1) Select an element of the front.
(2) Identify points close to this element, or generate new points.
(3) Sort the points according to defined priorities (quality, convergence).
(4) Create a tetrahedron with the first point which satisfies the validity tests.
(5) Update the front.

The validity test including the intersection test has been fully described by Löhner and Parikh [9], Bonet and Peraire [19] and Jin and Tanner [10]. We will also concentrate essentially on the selection of a candidate node, and on the organisation of the generation front.

We note that convergence of the algorithm depends mainly on the choice of heuristic node selection programs and the strategies adopted for generation organisation.

Generation of a tetrahedron on a selected face and a given set of nodes is not always guaranteed and the difficulty of 3D automatic mesh generation is not only restricted to the creation of a well-distributed set of nodes, and connection of this set while avoiding any intersection between elements. This topic is far more complex than a common geometrical problem. For 2D, a Delaunay or max–min angle criterion can be verified over the entire domain. However, no similar criterion can be extended to 3D problems [16,20]. The following extremely simple example shown in Fig. 1 illustrates one of the main obstacles encountered with the 3D advancing front method.

The object to be meshed is a prism, the skin faces of which cannot be modified. F is the selected face. Node D is not a candidate node, as the tetrahedron created would be flat. Node A cannot be selected as edge BC is intersected. Node B cannot be selected as edge AD is intersected. The solution involves of generating a node inside the prism. This situation occurs quite often when elements are created inside the volume. If the shell to be meshed is flat, node generation may lead to the creation of highly distorted elements.

Another difficulty encountered with a 3D advancing front mesh generator is the creation of an infinite series of nodes. The above configuration shows that during the element creation process, interior nodes must be created to prevent the procedure from failing. When a node is generated, new faces are created and the same situation may occur again. When a candidate node cannot be found in the vicinity of a selected face, a new node is usually created close enough to avoid any element overlap. Newly created faces are smaller than the current

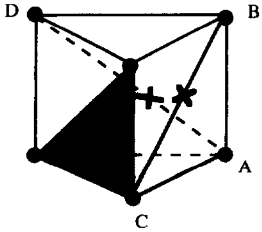


Fig. 1. Prism configuration.

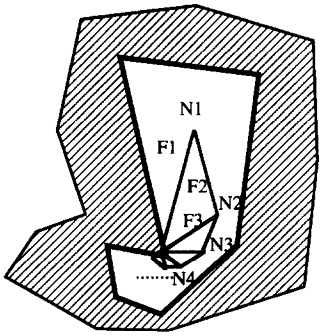


Fig. 2. An infinite series of tetrahedron coiled round an edge or point is created.

face. The volume of the meshable area decreases. If the same situation reappears, the tetrahedrons decrease in volume, and an infinite series of tetrahedrons coiled round a point edge, but which do not intersect, is generated (Fig. 2). A single check on the quality of the elements does not provide relevant information about the infinite loop, since the quality of the tetrahedrons is still acceptable.

The absence of a global space connection criterion must be offset by an efficient connection heuristic approach. The local connection principle applies to the frontal method. A node is sought or created to form a tetrahedron with a face. The faces of the tetrahedron thus created must not intersect other existing faces. Some of the methods presented employ a connection criterion which takes account of integration of the new element in a set of faces [21]. This type of approach indeed facilitates convergence at the next step, although it is essential to provide auxiliary solutions, in order to ensure convergence if the quality criterion of the tetrahedron to be created, or the integration criteria of the new element is or are not met. Complicated configurations can frequently be solved by degrading mesh quality below an acceptable threshold.

An efficient solution for convergence problems associated with the frontal method consists of destroying those elements which impede the tetrahedron creation process, or those elements which induce the problem. The proposed solution thus involves avoiding problems, either by eliminating each problem encountered, or by doubling back. In all cases, the 'wise' mesh generator must learn from its own errors, and avoid repeating them. Element destruction is dictated not only by convergence needs, but also by quality considerations. These processes thus form part of a global scheme, aimed at optimizing the quality of the complete mesh. Conflictual situations resulting from an insufficiently global approach (local selection criterion) are eliminated by destruction. When a convergence problem arises inside a volume, the global strategy involves destroying unsatisfactory elements, and leaving the problem on the skin of the volume to the end. Convergence strategy in fact depends on skin mesh quality alone.

The advantage of the proposed method is to adjust the quality degradations that have been temporarily accepted to accelerate algorithm convergence. Local optimization procedures can be used, once the mesh has been obtained, to enhance the mesh in areas where quality degradation has been allowed.

3. Mesh quality

We define some measures of both shape and size quality of a tetrahedron.

The shape quality criterion for a tetrahedron is defined as follows:

$$Q_e(T) = \alpha \frac{\rho}{h}, \tag{1}$$

where h is the longest edge length of the tetrahedron, and ρ is the ratio of the radius of the sphere inscribed in the element. A coefficient α is applied so that the higher criterion (equilateral element) is set at 1.

The size quality criterion for an edge of a tetrahedron is defined as follows:

$$Q_t(A) = \min\left(\frac{h_{\text{th}}}{h}, \frac{h}{h_{\text{th}}}\right) \quad \text{and} \quad 0 \leq Q_t(A) \leq 1, \tag{2}$$

h is the length of the edge, and h_{th} is the density of the mesh provided by the error estimation procedures.

The size quality criterion of a tetrahedron T is given by the lower quality of the 6 edges of the element.

$$Q_t(T) = \min(Q_t(A_i)) \quad i = 1, 6. \tag{3}$$

4. Meshing the whole volume

We consider first the functionalities of the mesh generator when no account needs be taken of local densities. The objective is to show the ability of the mesh generator to provide a good quality mesh with respect to a shape criterion even on the most complex geometries. The algorithm used to mesh the whole volume [18] is not presented in this paper but its extension to the meshing of sub-shells for optimization purposes is provided. However, the main concepts and associated tools which guarantee the success of the method are detailed thereafter.

4.1. Nodal generation by an octree technique

Nodes are created on an a priori basis, although additional nodes can be created during the connection process for solving the most complicated configurations. A conventional octree approach [12] is used for node generation. Element density is controlled by the skin mesh. The octree is built by successive insertion of weighted points inside the initial hexahedron root. Octants are divided recursively up to a solution level at which the ratio between the size of the octant containing the point to be inserted and the element size at the point does not exceed a value specified by the system. After recursive division, some octants are sub-divided to limit the level difference between adjacent octants to a factor of 2. Once the octree has been completed, nodes are generated at each vertex of the terminal octants. This provides a high degree of local control over the mesh during the element creation process.

4.2. Quality criterion

The quality criterion (shape criterion) for the global mesh process is set a priori. The value of this criterion is sufficiently high to avoid precision problems in the intersection tests, while ensuring the convergence of the algorithm.

4.3. Basic mechanisms

Basic mechanisms for creation and deletion of elements are reviewed.

Faces which have been eliminated from the generation front are disactivated. The generation front comprises all active faces.

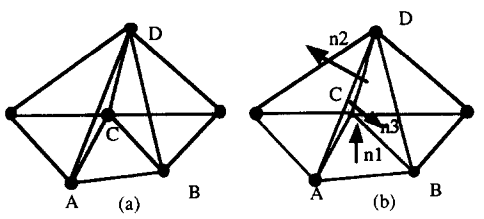
4.3.1. Tetrahedron creation

The creation of an element leads to the creation of a maximum of three new faces. Faces which share two nodes with the selected face (adjacent faces), which contains the candidate node, are deleted from the front. The current face is also eliminated.

4.3.2. Tetrahedron deletion

This mechanism is far more complex and requires particular care. The convergence of the method depends totally on the validity of generation front topology. The idea is to reinsert faces which are again visible in the generation front. When a tetrahedron is deleted, those faces of the front which do not belong to the skin, and which are connected to the tetrahedron, are deleted. Skin faces belonging to this tetrahedron (and this tetrahedron only, since a skin face can only belong to one tetrahedron) must be reactivated. Those faces of the deleted tetrahedron which also belong to another tetrahedron must also be reactivated.

The mechanism is shown in Fig. 3. (a): Tetrahedron (A, B, C, D) must be deleted. C is no longer a front node. Face (A, B, C) is a skin face and faces (A, D, C) and (C, D, B) were deleted from the generation front, when



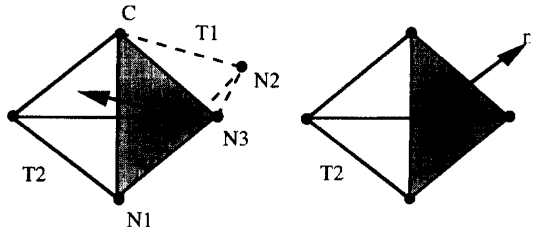


Fig. 3. Tetrahedron deletion.

Fig. 4. Face reactivation.

adjacent tetrahedra were created. Face (A, B, D) is the only face belonging to the front. (b): Tetrahedron (A, B, C, D) is deleted. The orientation of skin face (A, B, C) remains unchanged, but the orientation of interior faces must be changed where necessary, so that the normal vectors point outwards. The oriented face (A, D, C) is changed to (A, C, D), and face (C, D, B) to (C, B, D). Node C is reactivated. At the end of the process, node C is connected to three active faces and the cardinality of node C is set at 3.

4.3.3. *Face reactivation*

The mechanism is shown in Fig. 4. Face F was created at the same time as tetrahedron T1. Tetrahedron T1 and T2 share common face F. F was deleted from the generation front when T2 was created. If T1 is deleted, face F1 must be reactivated, and the orientation of F must be changed so that the normal vector points outwards. Links between faces and tetrahedra and nodal cardinality must be updated. The normal vector must be oriented in the direction of the node belonging to the deleted tetrahedron, but which does not belong to the face to be reactivated. The orientation of a face is changed by simply inverting its last two node numbers.

When faces are reactivated, nodes which were deleted from the generation front are again visible and must be considered as candidate nodes.

4.4. *Node triangulation*

We will concentrate essentially on the selection of a candidate node, and on the organisation of the generation front.

4.4.1. *Candidate node search procedure SEARCH1*

Priority candidate nodes are points which can be used to close the model, namely nodes which are located on faces sharing an edge with the faces (adjacent nodes) followed by other nodes close to the face. This order of priority is similar to that recommended by Golgolab [11]. The candidate node is one which makes it possible to create a tetrahedron which does not intersect existing faces and for which the quality criterion is maximum.

4.4.2. *Node generation procedure GENERATION1*

A set of candidate nodes is created on the normal to the current face passing through its centre of gravity. These nodes are subjected to the validity tests. If an intersection is detected, the next selected point is the point closer to the face. During the meshing process, the current face that has been deleted from the front can be reactivated. It is therefore essential to control the process to prevent recreating the same node a number of times, as this could lead to infinite creation/deletion loops.

4.4.3. *Deletion procedure SEARCH2*

The candidate node is the node which procedure SEARCH1 would select if no intersection is detected. The priority among the nodes remains unchanged. The quality criterion is set at PASSABLE. Therefore, the first node selected is the node adjacent to the face for which the quality criterion is maximum. The approach involves deleting tetrahedra which impede selection of this node. However skin faces cannot be deleted. The candidate node for procedure SEARCH2 is a node for which no intersection with a skin face is detected.

The procedure may be described as follows:

- Select the node by procedure SEARCH1.

- Identify faces which intersect the tetrahedron to be created.
  If a skin belongs to the selected set of faces, choose another node or store the elements in the list of tetrahedron to be deleted.
- Delete the selected tetrahedra.

4.4.4. Node generation procedure GENERATION2

The candidate node is the node which procedure GENERATION1 would select if no intersection is detected. The candidate node for procedure GENERATION2 is a node for which no intersection with a skin face is detected.

4.5. Multi frontal method

It seems obvious that the order in which faces of the front are activated has a considerable influence on the final result. However, we can observe that front management is the most neglected aspect of the advancing front method. Efficient front management is one of the keys to success. The front is traditionally sorted by order of size in order to avoid the larger elements perturbing creation of the smaller ones. Apart from the essential care required, too much emphasis is placed on the search for a candidate node rather than concentrating on front management.

Concentration of the development skills and efforts, and thus of the strategy, of a frontal mesh generator on the search for a candidate node, amounts to taking all decisions at the local level. We consider this approach inadequate. Front management strategies for working on the mesh at global level have been thus developed.

The front is subdivided into different lists according to the level of priority in which the faces are activated. The method involves priority construction of enhanced quality tetrahedra, to avoid weaker elements slowing or impeding the convergence of the algorithm.

4.5.1. Concept

Generation front organisation based on quality requirements is a new concept.

The method consists of constructing tetrahedra with optimum quality first, in order to prevent the worst tetrahedra (most extended) from slowing down, or even preventing the convergence of the algorithm. All the faces of the skin are chained. This list constitutes the initial front. The elements are sorted in ascending order of size so that the largest elements do not impede creation of the smallest.

4.5.2. Composition of the front

The multi-front is divided into a number of lists:

- Three lists (OPTI, MEDIUM and PASSABLE) correspond to minimum values for the quality criterion to be achieved (respectively 0.5, 0.2 and 0.1).
- List GENERATION1 corresponds to faces created by generation of a new node.
- List GENERATION2 corresponds to faces created by generation of a new node and deletion of tetrahedra impeding convergence of the algorithm.
- List REACTIVE corresponds to faces which are again visible, and which are therefore reinserted in the generation front.
- List WAIT includes all faces created during the meshing process, except faces created by generation of a new node.

4.5.3. Order of priority

The following element faces are processed in order of priority:

(1) Faces resulting from creation by nodal generation and deletion of elements: procedure GENERATION2.
(2) Faces resulting from creation by nodal generation only: procedure GENERATION1.
(3) Faces of the REACTIVATED front. These faces are inserted at the top of list OPTI, and therefore have priority. Areas in which elements have been deleted must be filled first.
(4) Faces with OPTI quality.
(5) Faces with MEDIUM quality.
(6) Faces with PASSABLE quality.

(7) Other faces created during the process are stored in a WAITing list. When all the other lists are empty, these faces are transferred to list OPTI, and the list is sorted by ascending order of size.

These faces are used to create a new layer of elements.

5. Mesh optimization

The optimization method presented here involves extracting shells from the tetrahedral mesh, and remeshing these sub-volumes while improving their quality (shape and size). These sub-volumes are constructed by detection of the set of tetrahedra touching the same node, edge or the same face. George et al. [8] have proposed a method for remeshing convex shells only. The approach presented here is not limited to convex shells, but can also be applied to all kinds of sub-volume. The procedures developed are extremely efficient and make natural and astute local use of the algorithm used to mesh the whole volume. The method is based on local transformations for enhancing the mesh.

5.1. Meshing of a shell

A shell is defined as the envelope of a set of tetrahedra sharing the same node or the same edge or the same face.

Mesh concepts used for both local and global approaches are similar. The order of priority of the candidate nodes (4.5.3) remains unchanged and procedures SEARCH1 (4.4.1) and SEARCH2 (4.4.3) can be applied. Priority candidate nodes are adjacent nodes followed by other nodes close to the face. Data structures (octree based) used to address the objects can be deactivated. The small number of nodes, faces and elements enables a systematic search on all the entities. No internal node is generated by a GENERATION type procedure. Therefore, lists GENERATION1 and GENERATION2 are removed from the multi-front (4.5.2). Apart from this, the order of priority in which front faces are processed is not changed. The algorithm used to mesh a sub-volume is described in Table 1.

5.2. Shape optimization

The quality criterion of the skin mesh is denoted PASSABLE (quality of the most distorted element). We select the elements the quality criterion of which is below a value fixed a priori. The initial value of the criterion is set at PASSABLE +0.1.

Sub-volumes are composed of elements sharing an edge or a face with the selected tetrahedra. The method consists of remeshing the shell constituted by the outer faces of the set of elements. The quality of a set of tetrahedra is that of the most distorted element. A local transformation is applied if the shell can be remeshed to a higher criterion. Interior nodes can be generated if necessary.

The process is iterative. Once all selected elements have been processed, the criterion is incremented (0.1), and another list of tetrahedra is constituted. When the criterion has reached value OPTI (0.5), the process is reversed and the criterion is decremented.

The optimization principle is local. While it is possible to examine all tetrahedron shell configurations locally in order to select the best, it is obvious that the same approach is inconceivable in a global context, where the main purpose is to define efficient front management and candidate node search strategies.

5.3. Size optimization

The procedures are quite similar to the one used for shape optimization purposes. The method consists of splitting edges of elements by adding new nodes (mesh refinement) or deleting existing nodes in order to increase local density (mesh coarsening). The maximum criterion OPTI at which the procedure can be applied is set to 0.6.

Table 1
Mesh of a shell

As long as the front is not empty:
If the face belongs to front OPTI or front REACTIVE:
  Provided the quality criterion set a priori is less than OPTI value.
  Search for a node leading to creation of a tetrahedron with OPTI quality **SEARCH1.**
  If a candidate node has been found:
    Create the tetrahedron and update the front.
    Delete the current face from the front.
    Store newly created faces at the bottom of list WAIT.
  If no candidate node is identified:
    Transfer the face to list MEDIUM.
  Go to SELECT.
If the face belongs to front MEDIUM (front OPTI is empty):
  Provided the quality criterion set a priori is less than MEDIUM value.
  Search for a node leading to creation of a tetrahedron with MEDIUM quality **SEARCH1.**
  If a candidate node has been found:
    Create the tetrahedron and update the front.
    Delete the current face from the front.
    Store newly created faces at the bottom of list WAIT.
  If no candidate node is identified:
    Transfer the face to list PASSABLE.
  Go to SELECT.
If the face belongs to front PASSABLE:
  PASSABLE value is the criterion set a priori.
  Search for a node leading to creation of a tetrahedron with PASSABLE quality **SEARCH1.**
  If a candidate node has been found:
    Create the tetrahedron and update the front.
    Delete the current face from the front.
    Store newly created faces at the bottom of list WAIT.
    In the global mesh approach, a node generation process may occur and the set of newly created faces are stored at the bottom of list WAIT.
  If no candidate node is identified:
    Destroy the incompatible tetrahedra: **SEARCH2.**
    Delete the selected tetrahedra.
    Create the tetrahedron and update the front.
    Store newly created faces at the bottom of list WAIT.
    Reactivated faces are stored in list REACTIVE.
  If no candidate node is identified:
    If the face is not a skin face, delete the tetrahedron connected to the face.
    Store reactivated faces in front REACTIVE.
    If the face belongs to the boundary.
    In the global mesh approach, test by generating a node and destroying the faces intersected by the tetrahedron.
    If no candidate is identified, the skin face is an intersected face and cannot therefore be destroyed.
    NO MESH OF THE SHELL CAN BE OBTAINED.
  Go to SELECT.
If the face belongs to front WAIT:
  Sort the faces by ascending order of size. These faces are inserted in front OPTI.
  Go to SELECT.
SELECT
  Extract a face from the fronts sorted in order of priority:
  REACTIVE, OPTI, MEDIUM, PASSABLE, WAIT.

5.4. Optimization procedures

5.4.1. Nodal shifting

The method was introduced by George et al. [8]. This procedure is applied to the nodes which do not belong to the boundary and can be associated to both shape or size criteria. The position of the nodes belonging to the boundary surfaces remains unchanged. The quality of the set of elements connected to a node is that of the most distorted element.

The procedure consists of shifting point P step by step to an ideal position P_{opt} with respect to the outer faces of the connected tetrahedra. The nodal shifting procedure can be written as

$$P = P + d \quad \text{with} \quad d = \alpha PP_{opt}$$

and

$$P_{opt} = \sum_{j=l}^{j=n} \beta_j P_{idj}, \qquad \beta_j = \frac{\gamma}{Qe(Tj)} \quad \text{and} \quad \gamma = \frac{1}{\sum_{j=l}^{j=n} \beta j}$$

n is the number of points connected to P. P_{idj} is the ideal position of P with respect to triangle T_j. $Qe(Tj)$ is the square of the quality criterion of the tetrahedron created with Tj and the ideal node. and βj is the value assigned to point P_{idj}. ($\Sigma\ \beta j = 1$).

Point P is moved step by step to position P_{opt} as long as the quality of the set of tetrahedra can be improved. Otherwise, the incremental step is decreased, and the process is reversed. The solution point is the centre of gravity of the ideal points of each outer face. The value of a weighted point is the inverse of the square of the quality criterion of the tetrahedron created with a face and its ideal point.

When a shape optimization process is requested, the ideal point is located on the normal to the current face passing through its center of gravity at a height corresponding to the third of the perimeter of the face. In the context of size optimization, the height is calculated with respect to the local density.

5.4.2. Meshing around an edge

The quality criterion is set a priori. One of the 6 edges of a tetrahedron is selected for deletion. Edges which belong to the skin mesh ('free edges') cannot be deleted. All tetrahedra which share this edge are determined and a volume is constituted (Fig. 5). The shell of the sub-volume is created and orientated. If the shell cannot be meshed, another edge of the tetrahedron is selected for deletion.

5.4.3. Meshing a shell by face deletion

A face of the tetrahedron is selected for deletion. However, a face which belongs to the skin mesh cannot be deleted. The face is shared by another tetrahedron.

The procedure consists of remeshing a shell comprising the outer faces of the two tetrahedra into three tetrahedra provided the quality criterion can be achieved. During the process, the face shared by the two tetrahedra is removed. If the shell cannot be meshed, another face of the tetrahedron is selected for deletion (Fig. 6).

5.4.4. Meshing around an edge by nodal insertion

This procedure consists of improving the quality of the shell by remeshing around free edges, while generating a node at the centre of the shell. Edges of the tetrahedron to be deleted are sorted by decreasing order of size. An edge (the longest) is selected. A node is created at the middle of the edge and relocated by the nodal

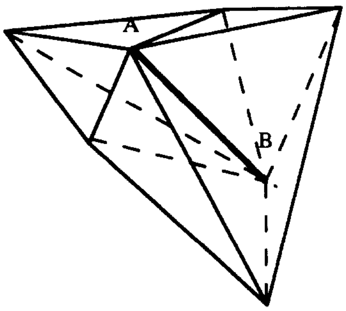


Fig. 5. Shell of tetrahedra sharing the same edge.

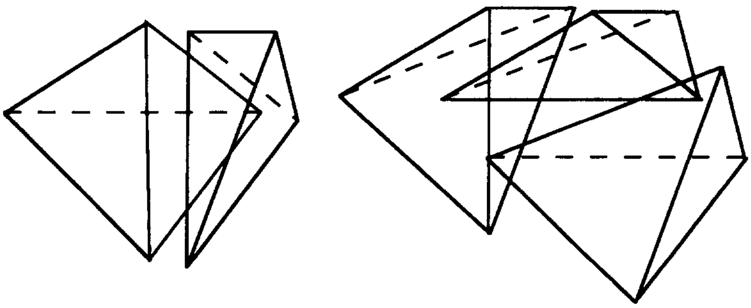


Fig. 6. Splitting of two tetrahedra into three tetrahedra.

shifting procedure. The shell around the point is meshed at a higher criterion if possible. If the shell cannot be meshed, another edge of the tetrahedron is selected for deletion.

5.4.5. Meshing by nodal deletion

This procedure consists of coarsening the mesh by remeshing the shell constituted by all tetrahedra sharing the same node. The node shared initially by all elements is eliminated from the final mesh.

5.5. Orientation procedures

Shell faces are orientated so that their normal vector point inwards. The orientation of a shell is correct if the normal vector points to the node (i) or the adjacent node (ii) which does not belong to the face (Fig. 7).

5.6. Main optimization procedures (shape and size)

The order in which the different procedures are activated has a considerable influence on the final result. In order to minimize the number of nodes, the *meshing around an edge by nodal insertion procedure* is activated only when other procedures have failed. Statistics and comparison between different heuristic programs reveal that the *meshing around an edge procedure* is the most efficient when used for shape optimization. The shape optimization procedure is summarized in Table 2. The size optimization procedure is summarized in Table 3.

This process is numerically robust, and no invalid triangulation can occur as a transformation is only performed if the sub-shell can be meshed to a higher criterion. During the process, many elements are deleted and the validity of the global mesh must be checked at each step. Extensive use is made of the mechanisms of creation, deletion and re-insertion of elements in the final mesh.

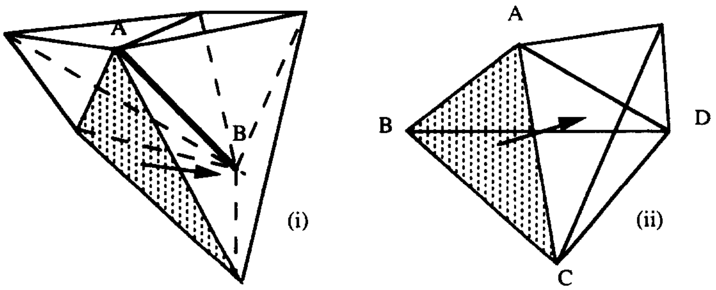


Fig. 7. Shell orientation.

Table 2
Summary of shape optimization procedure

SHAPE OPTIMIZATION (cmin)
Apply the **NODAL SHIFTING** procedure to all interior nodes.
Determine the set of tetrahedra the quality of which is below cmin.
For each tetrahedron of the set.
{
If the tetrahedron is not deleted.
Choose a free edge of the tetrahedron.
Determine the shell comprising the outer faces of the tetrahedra which share the edge.
Apply the **MESHING AROUND AN EDGE** procedure.
If procedure fails, choose another edge.
If the shell has been meshed go to **UPDATE**.
MESHING AROUND AN EDGE procedures have failed.
Choose a face of the tetrahedron (which does not belong to the skin).
Determine the shell comprising the outer faces of the two tetrahedra which share the face.
Apply the **MESHING OF A SHELL BY FACE DELETION** procedure.
If procedure fails, choose another face.
If the shell has been meshed go to **UPDATE.**
MESHING OF A SHELL BY FACE DELETION procedures have failed.
Choose a free edge of the tetrahedron.
Determine the shell constituted by the outer faces of the tetrahedra sharing the edge.
Insert a point to break the edge and apply **NODAL SHIFTING** procedure to that node.
Apply the **MESHING AROUND AN EDGE BY NODAL INSERTION** procedure.
If procedure fails, choose another edge.
If the shell has been meshed go to **UPDATE**.
If all the procedures have failed, the tetrahedron cannot be deleted.
}
UPDATE: delete all tetrahedra in the sub-volume.
Create new elements.

Table 3
Summary of size optimization procedure

SIZE OPTIMIZATION (cmin)
Apply the **NODAL SHIFTING** procedure associated with a size criterion to all interior nodes.
Determine the set of tetrahedra the quality of which is below cmin.
For each tetrahedron of the set.
If the tetrahedron is not deleted.
For each edge.
If the mesh must be refined.
Determine the shell constituted by the outer faces of the tetrahedra sharing the edge.
Insert a point to break the edge and apply **NODAL SHIFTING** procedure to that node.
Apply the **MESHING AROUND AN EDGE BY NODAL INSERTION** procedure.
If procedure fails, choose another edge.
If the shell has been meshed go to **UPDATE**.
If the mesh must be coarsened.
Determine the shell constituted by the outer faces of the tetrahedra sharing a node of the selected edge.
Apply the **MESHING BY NODAL DELETION** procedure.
If procedure fails, choose the other node of the edge or another edge.
If the shell has been meshed go to **UPDATE**.
IF ALL THE PROCEDURES HAVE FAILED, THE TETRAHEDRON CANNOT BE DELETED.
UPDATE: delete all tetrahedra in the sub-volume.
Create new elements.

5.7. Coupling shape and size optimization

The presence of poor quality faces (flattened triangles, sharp angles or excessive edge length ratio) is the cause of many precision problems in connection with intersection tests. A poor quality face leads to elements of equally poor quality, while a local problem can be propagated, degrading the global shape quality of the mesh, or preventing convergence. Therefore, we truly believe that shape quality must be controlled during the size optimization process.

We suppose that a first mesh of the volume has been completed and that the skin mesh of the volume is consistent with the density provided by the error sensitivity analysis.

The process which combines both shape and size optimization can be described as follows:

(1) A full cycle of shape optimization is made initially.
As long as the size quality can be improved, the two following steps must be repeated.

(2) Size optimization cycle.
The initial value of criterion cmin is set at 0.1
When cmin is less than 0.6
  Increment quality criterion (cmin = cmin + 0.1)
  Execute procedure optimization-size (cmin)
When cmin is greater than 0.1
  Decrement quality criterion (cmin = cmin − 0.1)
  Execute procedure optimization-size (cmin).

(3) Apply shape optimization procedures meshing around an edge and meshing a shell by face deletion only. The existing pattern of nodes remains unchanged and no new node is created.

6. Optimization of convex shells

George et al. [8] have presented a method which can be applied for improving the quality of convex shells only. These shells must have less than 12 nodes.

The method is based on the following notions:

AB is a free edge. Shell C_{AB} is constituted by the n tetrahedra sharing edge AB. The shell comprising the outer faces of the tetrahedra can be written as

$$C_{AB} = (M_i A M_{i+1} B)_{i=1,\,n} .$$

If shell C_{AB} is convex, the different meshes of C_{AB} can be written as

$$C_{AB} = (M_i A M_{i+1} B)_{i=1,\,n} = (M_j M_k M_l A)(M_j M_k M_l B)_{j,\,k,\,l} = 1, n,$$

provided no elements overlap. The n M_i points constitute a polygon. This polygon is not necessarily planar, and is referred to as a pseudo-polygon.

The method consists of finding all triangulations of the pseudo-polygon, then joining points A and B located on both sides of the polygon in order to obtain the tetrahedron mesh. The quality of the shell is improved by choosing the triangulation of the polygon for which the quality is optimal.

Each triangulation of the pseudo-polygon is composed of a set of $n-2$ triangles (Fig. 8).

T_n is the number of all different triangles which can be created from all triangulations

$$T_n = \frac{n(n-1)(n-2)}{6} .$$

$2T_n$ different tetrahedra can be created. The quality of each tetrahedron is then evaluated. The different meshes of the shell are represented by a set of pointers to the $2T_n$ tetrahedra. The number of faces has been restricted to 20 (for a 10-node pseudo-polygon, there are already 1430 triangulations and 120 tetrahedra).

The different mesh combinations do not depend on the position of the nodes, and are therefore pre-programmed.

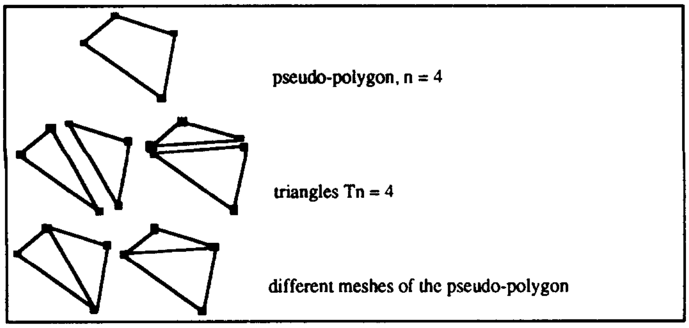


Fig. 8. Triangulations of a pseudo-polygon ($n=4$).

This method has also been applied [22] and the following examples show that a better quality mesh can be obtained with our method based on the advancing front approach. This can be explained by the fact that our approach is not restricted to convex shells and that there is no limitation of the number of faces constituting the shells.

However, we observe that the cost of this method is higher than that of the pre-programmed approach.

7. Examples

7.1. Meshes of complex models. Validation of shape optimization procedures

The following examples provide a significant overview of the main difficulties encountered with automatic 3D meshing and show the efficiency of the mesh generator in its 'natural' use and of the shape optimization procedures. For each test, the following characteristics have been calculated: Minimum surface mesh quality, maximum/minimum edge length ratio, and CPU time for both meshing and optimization processes (Silicon Graphics Indigo 2).

The tests show that a high quality mesh can be obtained.

EXAMPLE 1: Mines (Courtesy of CGES, Mines de Paris) (Table 4). We discretize a polymetallic mass shown in Fig. 9. The surface was meshed using a GOCAD software procedure [23]. The shape of the object is extremely irregular and the gradation of the surface mesh is very stiff.

After the first optimization step at criterion 0.2, we note that we already have a mesh of acceptable quality. Further improvement of results at a higher cost can be achieved when the complete optimization procedure (5.2) is carried out. Minimum surface mesh quality: 0.24, maximum/minimum edge length ratio: >40.

Table 4
Quality statistics of Example 1

	Non-optimized	Frontal OPTI=0.2	Frontal OPTI=0.5	Convex shells
Element number	60615	62269	61596	62412
$0.5<Q<1$	39676	42556	50929	47315
%	**65**	**68**	**83**	**76**
$0.2<Q<0.5$	17369	19609	10665	15063
%	**29**	**32**	**17**	**24**
$0.1<Q<0.2$	3396	104	2	34
%	**5**	**0^+**	**0^+**	**0^+**
$Q<0.1$	174	0	0	0
%	**0^+**	**0**	**0**	**0**
Qmin	0.05	0.10	0.17	0.10
CPU Mesh	1105	1105	1105	1105
CPU optimization	–	109	1328	847

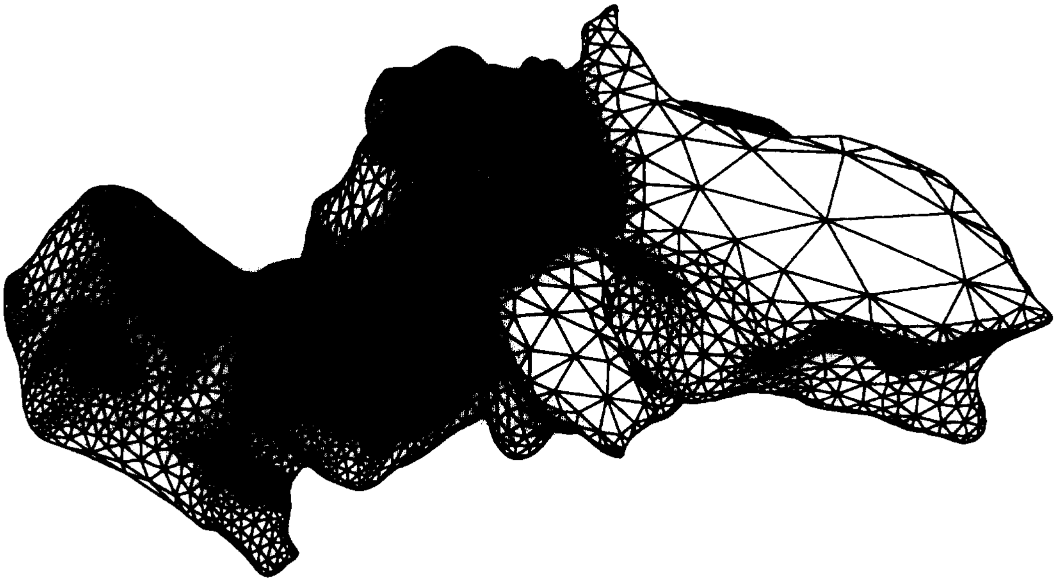


Fig. 9. Mines (Courtesy of CGES).

EXAMPLE 2: Steering gearbox (courtesy of Peugeot SA) (Table 5). At first sight, the skin mesh shown in Fig. 10 is of good quality. However, a number of connecting fillets are modelled using very elongated triangles (approximately 30). However, this local problem does not impair the global quality of the mesh. This example provides an excellent illustration of the power of the mesh algorithm and optimization procedures. Minimum surface mesh quality: 0.075, maximum/minimum edge length ratio: 4.

EXAMPLE 3: (Courtesy of SONY Research Center, Japan) (Table 6). This example combines different typical difficulties of a 3D mesh: The edge length ratio is high, the surface mesh element size progression is extremely irregular, a hundred of the 2D elements have a poor shape. Minimum surface mesh quality: 0.15, maximum/minimum edge length ratio: >40 (Fig. 11).

EXAMPLE 4: (Courtesy of Mercedes Benz) (Table 7). The object is a suspension shaft (Fig. 12). The interior shapes of the object are extremely complex. Minimum surface mesh quality: 0.12, maximum/minimum edge length ratio: 7.

Table 5
Quality statistics of Example 2

	Non-optimized	Frontal method	Convex shells
Element number	53827	61665	64433
$0.5<Q<1$	35516	49337	46108
%	**66**	**80**	**72**
$0.2<Q<0.5$	13398	12205	18121
%	**25**	**20**	**28**
$0.1<Q<0.2$	3328	122	202
%	**6**	$\mathbf{0^+}$	$\mathbf{0^+}$
$Q<0.1$	1585	1	2
%	**3**	$\mathbf{0^+}$	$\mathbf{0^+}$
Qmin	0.05	0.088	0.080
CPU Mesh	936	936	936
CPU optimization	–	1095	620

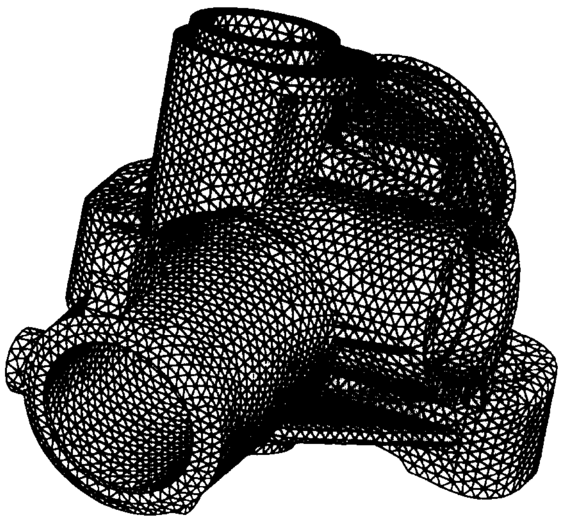


Fig. 10. Steering gearbox (courtesy of Peugeot SA).

Table 6
Quality statistics of Example 3

	Non-optimized	Frontal method	Convex shells
Element number	21525	25363	25523
$0.5<Q<1$	15354	20828	19805
%	**71**	**82**	**77**
$0.2<Q<0.5$	4312	4386	5513
%	**20**	**17**	**22**
$0.1<Q<0.2$	1047	149	197
%	**5**	**1**	**1**
$Q<0.1$	812	0	8
%	**4**	**0**	**0^+**
Qmin	0.05	0.13	0.07
CPU Mesh	348	348	348
CPU optimization	–	389	212

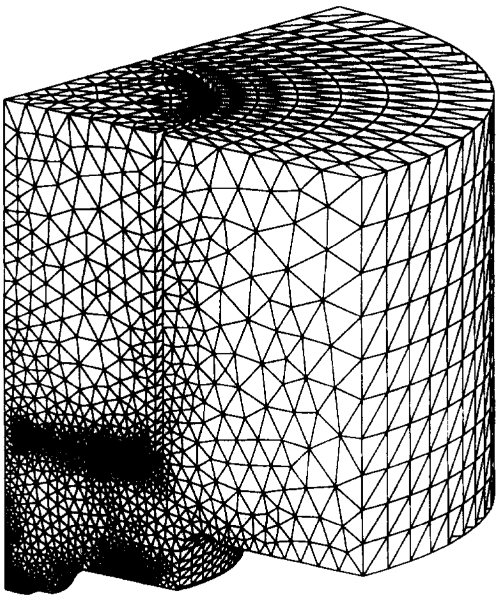


Fig. 11. Courtesy of SONY Research Center, Japan.

Table 7
Quality statistics of Example 4

	Non-optimized	Frontal method	Convex shells
Element number	15965	18780	19272
$0.5<Q<1$	9807	14600	13317
%	**61**	**78**	**69**
$0.2<Q<0.5$	4663	4172	5936
%	**29**	**22**	**31**
$0.1<Q<0.2$	1059	8	19
%	**7**	$\mathbf{0^+}$	$\mathbf{0^+}$
$Q<0.1$	436	0	0
%	**3**	**0**	**0**
Qmin	0.05	0.12	0.12
CPU Mesh	324	324	324
CPU optimization	–	301	167

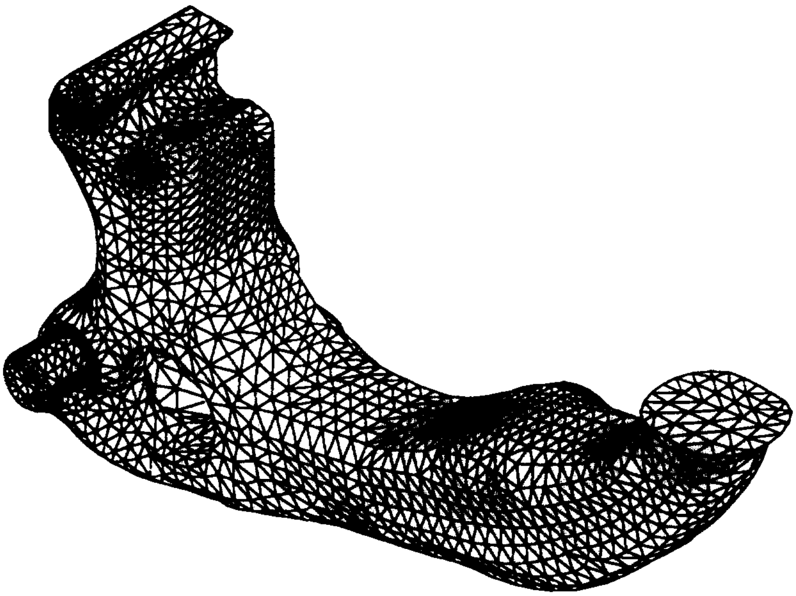


Fig. 12. Suspension shaft, courtesy of Mercedes Benz.

7.2. *Validation of size optimization procedures.*

The method proposed to demonstrate the efficiency of the procedures consists of giving an explicit element density $h(x, y, z)$ over the domain. Then, the model with respect to this distribution. The mesh generator accepts, of course, a background coarse mesh in which a distribution card of element size densities is given at each node. We could also calculate the element density using the same distribution $h(x, y, z)$ at each node of a coarse mesh in order to simulate an error sensitivity analysis and then use the coarse mesh to determine the element size inside the domain.

We suppose that the surface mesh of the volume has been meshed with respect to the imposed local densities. We consider that the quality of edges of the surface mesh is maximum, i.e. $Q_{\text{size}}(E)=1$ if E is an edge of the surface envelope.

Let Q_{opt} be the minimum quality criterion to reach.

We set this value to 0.5 for shape optimization purposes and to 0.6 for size optimization purposes.

EXAMPLE 5 (Table 8): The volume to mesh is a cube. Let L be the length of the edges. Edges have been split into three equal segments. The density at the centre is set at $L/1141$ and the element size inside the domain is

Table 8
Quality statistics of Example 5

	Shape criterion $Q_{opt} = 0.5$	Size criterion $Q_{opt} = 0.6$
Element number	2408	2408
$Q_{opt} < Q < 1$	2295	2403
%	**95**	**100**
$0.2 < Q < Q_{opt}$	113	5
%	**5**	**0**
$0.1 < Q < 0.2$	0	0
%	**0**	**0**
$Q < 0.1$	0	0
%	**0**	**0**
Qmin	0.37	0.5

given by

$$h(x, y, z) = h(r) = \left(\frac{L}{3} - h_{\text{centre}}\right) \times \frac{r}{r} + h_{\text{centre}} \quad \text{if } r < R,$$

$$h(x, y, z) = h(r) = \frac{L}{3} \quad \text{if } r \geq R \tag{4}$$

where r is the distance between a point inside the cube and the centre of the cube, and h_{centre} is the density at the centre of the cube and $R = L/2$.

The distribution of element size is assumed to be linear in the radial direction and grows from $h(r=0) = h_{\text{centre}}$ to $h(r=R) = L/3$ (Fig. 13). The initial mesh is constituted of 316 tetrahedra. A view inside the optimized mesh is provided in Fig. 14.

We can consider that nearly all elements have the desired size. This example shows that the mesh can be strongly graded while the shape quality of the elements remains quite good. The CPU time of the process is 847 s (Silicon Graphics Indigo 2).

EXAMPLE 6 (Table 9): The next example used to demonstrate the effectiveness of the adaptation approach is a cube in which the distribution function is defined in each of the eight octants of the cube.

Edges have been split into five equal segments. The element size in each octant is given by the radial distribution $h(r)$ defined previously (Fig. 15). The density at the centre of each octant is set at $L/100$ and R is set at $L/5$.

The initial mesh is constituted of 628 tetrahedra.

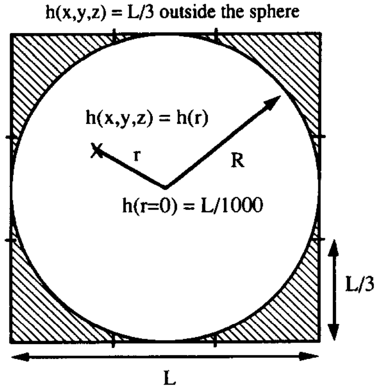


Fig. 13. Radial distribution of element size.

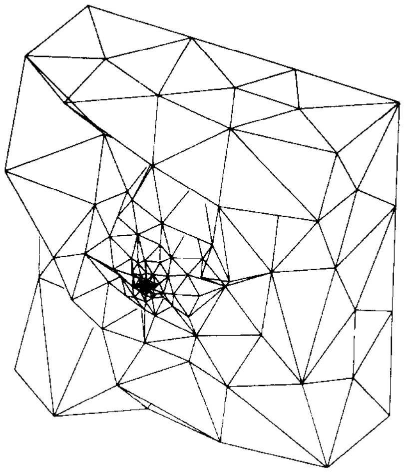


Fig. 14. Example 5: View inside the optimized mesh.

Table 9
Quality statistics of Example 6

	Shape criterion $Q_{opt}=0.5$	Size criterion $Q_{opt}=0.6$
Number of elements	8806	8806
$Q_{opt}<Q<1$	8315	8768
%	**95**	**100**
$0.2<Q<Q_{opt}$	488	38
%	**5**	**0**
$0.1<Q<0.2$	3	0
%	**0**	**0**
$Q<0.1$	0	0
%	**0**	**0**
Qmin	0.16	0.5

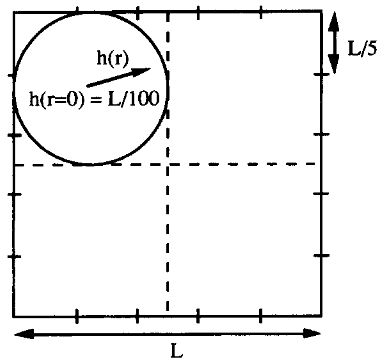


Fig. 15. Radial distribution of element size in each octant.

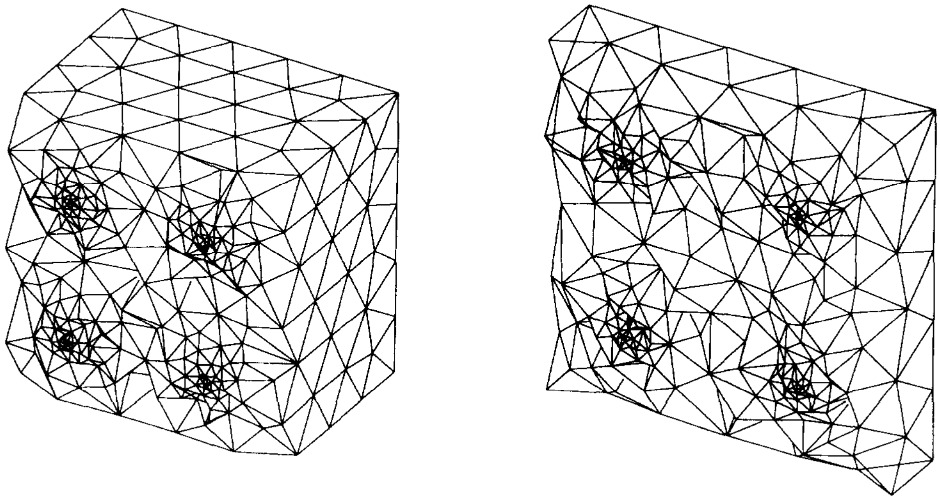


Fig. 16. Example 6: View inside the optimized mesh.

Views of the optimized mesh are provided in Fig. 16. All elements have the desired size. The CPU time of the process is 1024 s (Silicon Graphics Indigo 2).

8. Conclusions

A new mesh generation approach based on the advancing front technique has been presented. The strategy of the mesh generator involves creating, on a priority basis, those elements which lead to the creation of the best tetrahedra. Each face is assigned to a front corresponding to the quality of the best tetrahedron which can be constructed. The multi-frontal approach represents a fundamental aspect of the method. Front management and candidate node search aspects tend to privilege the closure of the volume. Elements can be destroyed in the case of non-convergence. The examples presented demonstrate the ability of the method to mesh complex volumes while meeting high quality standards. An astute application of the algorithm used to mesh the complete object has been applied to optimize to mesh with respect to a shape and a size criterion. The optimization method involves extracting shells from the tetrahedral mesh, and remeshing these sub-volumes while improving their quality. The procedures to adapt three-dimensional unstructured meshes have been described. The adaptation combines node movement, point enrichment and removal of nodes from excessively dense regions.

Acknowledgments

This work was conducted within the ERIN team of the University of Nancy and supported by FRAMASOFT+CSI, a subsidiary of the FRAMATOME Group and by *le Ministère de l'Enseignement Supérieur et de la Recherche* within the project *Saut technologique 'Nouveaux outils de conception et de modélisation en mécanique et disciplines associées'*. This assistance is gratefully acknowledged. This work contributed to the development of the SYSMESH® automatic mesh generator marketed by FRAMASOFT+CSI. I would like to thank Pr. Michel Gueury, Head of the ERIN team, Mr. P.-L. George, Director of Research at INRIA, Rocquencourt, France and Mr Jean Donéa of the Centre Commun de Recherche in Ispra for their extremely valuable help and advice.

References

[1] P. Beckers and H.G Zhong, Mesh adaptation for 2D stress analysis, 2nd Int. Conf. on Computational Structures Technology, Athens Greece, August 30–September 1, 1994, LTAS report SA-178.

[2] F. Cugnon and P. Beckers, Design of a 3-D error control procedure, September 1995, LTAS report SA-183.

[3] P. Ladeveze, J-.P. Pelle and P. Rougeot, Error estimation and mesh optimization for classical finite elements, Engrg. Comput. 8 (1991) 69–80.

[4] O.C. Zienkiewicz and J.Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, Int. J. Numer. Methods Engrg. 24 (1987) 337–357.

[5] M.C. Rivara, Selective refinement/derefinement algorithms for sequences of nested triangulations, Int. J. Numer. Methods Engrg. 24 (1987) 1343–1354.

[6] P.-L. George, Les problèmes de l'adaptation sur l'exemple d'un mailleur 2D soumis à une carte de taille, Journée d'Etudes CSMA: 'Vers l'automatisation des calculs éléments finis', Paris, June 1994.

[7] F. Noël, P. Pasquet, J.C. Léon and P. Trompette, Adaptation d'un maillage surfacique en référence un modèle NURBS, Actes du Congrès StruCome 93 (Paris 1993) 241–252.

[8] P.-L. George, F. Henot F. and E. Brière de L'Isle, Optimisation de maillages tétraèdriques, Actes du Congrès StruCome 93, Paris, 1993.

[9] R. Löhner and P. Parikh, Generation of three dimensional unstructured grids by the advancing front method, Int. J. Numer. Methods Fluids 8 (1988) 1135–1149

[10] H. Jin and R.I. Tanner, Generation of unstructured tetrahedral meshes by advancing front technique, Int. J. Numer. Methods Engrg. 36 (1993) 1805–1823.

[11] A. Golgolab, Mailleur 3D automatique pour des géométries complexes, Rapport de Recherche I.N.R.I.A, n° 1004, prog. 7, mars 1989.

[12] W.J. Schröder and M.S. Shephard, A combined octree/Delaunay method for fully automatic 3-D mesh generation, Int. J. Numer. Methods Engrg. 29 (1990) 37–55.

[13] T.J. Baker, Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation, Engrg. Comput. 5 (1989) 161–175.

[14] T.J. Baker, Development and trends in three-dimensional mesh generation, Appl. Numer. Math. 5 (1989) 275–304.

[15] P.L. George, F. Hecht and E. Saltel, Automatic mesh generator with specified boundary, Comput. Methods Appl. Mech. Engrg. 92 (1991) 269–288.

[16] D.F. Watson, Computing the n-dimensional Delaunay tesselation with application to Voronoï polytopes, Comput. J. 24(2) (1981) 167–172.

[17] S.H. Lo, Volume discretization into tetrahedra—I, verification and orientation of boundary surfaces, Comput. Struct. 39(5) (1991) 493–500.

[18] A. Rassineux, Maillage automatique triolimensionnel par une technique frontale pour la méthode des éléments finis, Ph.D. Thesis, Université de Nancy, 1995.

[19] J. Peraire and J. Bonet, An alternative digital tree (ADT) algorithm for 3D geometric searching and intersection problems, Int. J. Numer. Methods Engrg. 31 (1991), 11–17.

[20] B. Joe, Delaunay versus max-min solid angle triangulations for three-dimensional mesh generation, Int. J. Numer. Methods Engrg. 31 (1991) 987–997.

[21] S.H. Lo, Volume discretization into tetrahedra—II, 3D triangulation by advancing front approach, Comput. Struct. 39(5) (1991) 501–511.

[22] V. François, Optimisation de maillages en tétraèdres, private communication, Framasoft+CSI, 1995.

[23] M. Dolliazal, From cross-section to surfaces, GOCAD Meeting, Nancy-Vandoeuvre, June 1994.