Proceedings of the XXVI Iberian Latin-American Congress on Computational Methods in Engineering CILAMCE 2005
Brazilian Assoc. for Comp. Mechanics & Latin American Assoc. of Comp. Methods in Engineering
Guarapari, Espírito Santo, Brazil, 19th–21st October 2005

Paper CIL14-0916

A STRATEGY FOR IMPROVING THE ROBUSTNESS OF THREE-DIMENSIONAL ADVANCING-FRONT ALGORITHMS

Joaquim B. Cavalcante-Neto
joaquimb@lia.ufc.br
Computer Graphics Group (CRAb), Federal University of Ceara (UFC), Brazil

Paul A. Wawrzynek
wash@fac.cfg.cornell.edu
Cornell Fracture Group (CFG), Cornell University, USA

Luiz Fernando Martha
lfm@tecgraf.puc-rio.br
Computer Graphics Technology Group (Tecgraf), Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil

Anthony R. Ingraffea
ari1@cornell.edu
Cornell Fracture Group (CFG), Cornell University, USA

Abstract. *This work presents a back-tracking strategy for improving the robustness of advancing-front algorithms for mesh generation in three-dimensional models. It can be observed, in many advancing-front based algorithms for three dimensions, that a number of regions that can not be meshed is left after the mesh generation process, which stops the algorithms from converging. These regions are represented by polyhedra and are formed by elements of the current front, which are faces in three dimensions. These regions are usually disconnected, since they are defined by adjacent elements that form a closed loop, and they can not be meshed, even with the insertion of additional new nodes. A back-tracking procedure is applied to all of these regions, in order to guarantee robustness in mesh generation. It is devised for three-dimensional cases, because this problem has no counterpart in two dimensions. Examples of generated meshes using the back-tracking procedure are presented in order to validate the strategy proposed in this work.*

Keywords: *Three-dimensional Mesh Generation, Advancing-Front, Back-tracking*

1. INTRODUCTION

This work presents a back-tracking strategy for improving the robustness of algorithms for advancing-front mesh generation in three-dimensional models. It can be observed, in many advancing-front based algorithms for three dimensions, that a number of regions that can not be meshed is left after the mesh generation process, which stops the algorithms from converging. These regions are represented by polyhedra and are formed by elements of the current front, which are faces in three dimensions. These regions are usually disconnected, since they are defined by adjacent elements that form a closed loop,

and they can not be meshed, even with the insertion of additional new nodes. The back-tracking procedure is applied to all of these regions, in order to guarantee robustness in mesh generation.

There are some techniques in the literature that have addressed, in some way, the problem of robustness in mesh generation for advancing-front based algorithms in three dimensions. There is a variety of solutions, from simple to very complicated ones, but in the overall the idea is somehow to fix the advance of the front, in order to obtain convergence. Usually, these solutions are heuristic attempts to avoid the problem of missing closure of the advancing-front algorithm. The necessity of these procedures arise from the fact that, unlike triangulation in two dimensions, the discretization of any given volume into tetrahedra is not formally ensured, unless some additional steps are performed.

One example of a technique for trying to improve the robustness in mesh generation is the work of Chan and Anastasiou (1997), which uses local mesh regeneration based on the deletion of sliver tetrahedra in a post-processing step. The procedure proposed in the present work has the same objective of that step, but uses a different strategy. Rassineux (1998) also optimizes the mesh by reconstruction of sub-volumes that are obtained by the deletion of a group of tetrahedra. Other technique devised to address the problem was proposed by Moller and Hansbo (1995), where the front was modified and updated during the process. This idea, by the way, is the most common one in algorithms that try to address this problem. Although these techniques present good results in many cases, since they are empirical procedures the guaranteed robustness is hard to achieve.

The strategy proposed in this work is based on a back-tracking idea already used by the authors in mesh generation (Cavalcante-Neto et al., 2001) and mesh optimization (Cavalcante-Neto et al., 2004) with satisfactory results. It incorporates some improvements, such as a better determination of the disconnected regions to be meshed, which increases the robustness of the method. Also, this work formally describes the back-tracking process, which was not done in these previous works. Finally, the back-tracking in the work by Cavalcante-Neto et al. (2001) was only part of a particular advancing-front algorithm for three-dimensional mesh generation presented by the authors, whereas the procedure proposed in this work was generalized and it can be applied in any advancing-front algorithm where the problem of disconnected regions occurs, fixing this problem and generating a mesh.

The strategy of the procedure is to delete an element of the front from a non-meshable region, generate a new region around this element and try to insert a new node in the centroid of this region, in order to generate new elements of the mesh. If this region does not have a kernel, which is a core where any point is visible from a straight line from all its vertices to the centroid, this region must be expanded and the process repeated, until a kernel is found. This process tries to find a meshable region, represented by a star-shaped region (polyhedron in 3D), relative to its centroid, where new elements of the mesh can be formed by the centroid and its vertices.

The remainder of this work is organized as follows. Section 2 describes the strategy in a general manner and its algorithm. Section 3 discusses the application of this strategy in three-dimensional problems. The next Section shows validation studies that attest the robustness of the technique. Finally, in Section 5 some conclusions are presented.

2. BACK-TRACKING STRATEGY

This Section describes the strategy of the back-tracking procedure and its algorithm, in a general manner. The intention of this section is to explain only the idea of the process and all the entities involved. This strategy will later be described for three dimensional problems with their specific requirements and details.

2.1 General description

Initially, it is important to define the types of elements involved in the process. The general idea of the advancing-front technique for generation of meshes is that elements of a front advance from the boundary of the model towards to its interior, forming elements of a mesh until this mesh is fully generated. An element of the front is here called a "front-element" (for instance, a face in a three-dimensional case), and an element of a mesh is here called a "mesh-element" (a tetrahedron in three dimensions, for example). Every front-element is supposed to form or to relate to one or more mesh-elements. If the front-element is on the surface mesh that defines the initial front for the model, than this front-element is related to only one mesh-element, whereas the front-element is related to more than one mesh-element if it is in the interior of the model. It is important to mention that only front-elements are involved in the advancing-front process, that is, the current front is always formed by front-elements. When a mesh-element is generated, this element is stored in a separate list of mesh-elements and the associations with the related front-elements are also stored.

It can be observed, in many advancing-front based algorithms for three dimensions, that a number of regions that can not be meshed is left after the mesh generation process, which stops the algorithms from converging. These regions are formed by a number of front-elements (for instance, this region is represented by polyhedra in three dimensions). These regions are usually disconnected, since they are defined by adjacent front-elements that form a closed loop, and they can not be meshed, even with the insertion of additional new nodes. The back-tracking strategy is applied to all of these regions, in order to guarantee robustness in mesh generation.

The idea of the strategy is then to delete a front-element in the current front (associated with a non-meshable region), generate a new region around this element and try to insert a new node in the centroid of this region, in order to generate new mesh-elements. If this region does not have a kernel, which is a core where any point is visible from a straight line from all its vertices to the centroid, this region must be expanded and the process repeated, until a kernel is found. This process tries to find a meshable region, represented by a star-shaped region, relative to its centroid, where new mesh-elements can be formed by the centroid and its vertices.

Let $\mathbf{E_f}$ be a front-element in the current front and $\mathbf{R}(\mathbf{E_f})$ the local region, formed by front-elements associated with $\mathbf{E_f}$, that can not generate mesh-elements in a valid mesh. The region $\mathbf{R}(\mathbf{E_f})$ is identified grouping the front-element $\mathbf{E_f}$ with adjacent front-elements (for instance, front-elements that share faces and edges in three dimensions). Let $\mathbf{D}(\mathbf{E_f})$ be the set of front-elements deleted and $\mathbf{A}(\mathbf{E_f})$ the set of front-elements added, during the back-tracking process, if necessary. Let $\mathbf{E_m}$ be a mesh-element related to the front-element $\mathbf{E_f}$ that can be also deleted during the back-tracking process, if necessary,

and $\mathbf{G}(\mathbf{E_m})$ the set of new mesh-elements generated for the local region $\mathbf{R}(\mathbf{E_f})$ when a kernel for this region is found. Finally, let $\mathbf{F}$ and $\mathbf{M}$ be the current list of all front-elements and mesh-elements, respectively. The procedure used to mesh the local region is as follows:

- The boundary of the non-meshable local region $\mathbf{R}(\mathbf{E_f})$ related to the current front-element $\mathbf{E_f}$ is identified.
- A visibility test is performed. This consists of computing the coordinates of the center of this region $\mathbf{R}(\mathbf{E_f})$ and counting the number of intersections that would occur, for each of the front-elements of this region, if lines were drawn from the center to all region's vertices.
- If there is at least one intersection for any of the region's front-elements, this region must be modified. This is done by removing the front-element that has the highest number of intersections from the region $\mathbf{R}(\mathbf{E_f})$ as well as from the front list $\mathbf{F}$ and the mesh-element $\mathbf{E_m}$ attached to this front-element from the mesh list $\mathbf{M}$. This front-element is not necessarily the same initial front-element used to determine the initial local region, and for this reason, $\mathbf{E_f}$ is updated to represent this particular front-element. A removal of a mesh-element implies in a possible adition or deletion of other front-elements from the front, represented by $\mathbf{A}(\mathbf{E_f})$ and $\mathbf{D}(\mathbf{E_f})$, respectively, since a mesh-element is usually formed by more than one front-elements (for instance, a tetrahedron is formed by four faces in three dimensions). Also, the front list $\mathbf{F}$ must be updated to account for these front-elements. All these operations usually results in the growth of the region $\mathbf{R}(\mathbf{E_f})$, since there are more inclusions than deletions of front-elements, in general.
- This process is repeated until the center of the region $\mathbf{R}(\mathbf{E_f})$ is visible from all vertices of this region. When this happen, a set of new mesh-elements $\mathbf{G}(\mathbf{E_m})$ is generated for the region $\mathbf{R}(\mathbf{E_f})$, by connecting the center to all region's vertices. The list of all mesh-elements on the current mesh $\mathbf{M}$ must be updated to account for these new mesh-elements $\mathbf{G}(\mathbf{E_m})$ generated.

It is possible that the process of findind a star-shaped region, relative to its centroid, may fail if front-elements to be removed are part of the original boundary mesh. When this occurs, the front-elements attached to internal faces with non-zero intersection counts are deleted and the element extraction procedure is restarted. If the region is still not meshable, the procedure fails and terminates. In principle, it is possible to create a boundary input mesh that forces failure of the process. Such a failure, however, has not yet been observed for "non-contrived" input, i.e., for all the realistic input boundary meshes, such as the ones shown in Section 4 and the other ones tested so far.

2.2 General algorithm

A general algorithm for the back-tracking procedure, which consists of all steps previously described, is as follows:

- Determine non-meshable region $\mathbf{R}(\mathbf{E_f})$ for the front-element $\mathbf{E_f}$.

- Perform visibility tests for region $\mathbf{R}(\mathbf{E_f})$.
- If there are intersections, update region $\mathbf{R}(\mathbf{E_f})$ by $\mathbf{R}(\mathbf{E_f})_{\mathbf{new}} = \mathbf{R}(\mathbf{E_f})_{\mathbf{old}} - \mathbf{E_f} - \mathbf{D}(\mathbf{E_f}) + \mathbf{A}(\mathbf{E_f})$. Also update the current front and mesh lists by $\mathbf{F_{new}} = \mathbf{F_{old}} - \mathbf{E_f} - \mathbf{D}(\mathbf{E_f}) + \mathbf{A}(\mathbf{E_f})$ and $\mathbf{M_{new}} = \mathbf{M_{old}} - \mathbf{E_m}$.
- Repeat the process until a kernel is found. When this happen, find the set of new elements $\mathbf{G}(\mathbf{E_m})$ and update the mesh list $\mathbf{M}$ by $\mathbf{M_{new}} = \mathbf{M_{old}} + \mathbf{G}(\mathbf{E_m})$.

3. THREE-DIMENSIONAL BACK-TRACKING

This Section describes the details for the application of the proposed technique to three-dimensional problems. Basically, the entities involved in the procedure are faces, which represent the front-elements, and tetrahedra, representing the mesh-elements, with their respective roles.

3.1 Strategy applied to 3D models

The back-tracking procedure consists of deleting a front-element from a non-meshable region and a group of front-elements in its vicinity. In three-dimensional models, the front-element is represented by a face (triangle, in simplex meshes). The objective of the back-tracking procedure is to delete faces surrounding a triangle to modify the non-meshable local region, in this case a polyhedron, that can be then meshed with tetrahedra. After the modification of the local polyhedron, an attempt is made to generate tetrahedra by inserting a new internal node in the polyhedron's center. Is this does not work, the back-tracking procedure described in Section 2 is employed.

Figures 1 and 2 are used to illustrate the back-tracking process. Figure 1 illustrates a hypothetical three-dimensional model and the aspect of the disconnected non-meshable regions that can occur. Figure 2 illustrates the transformation of a non-meshable polyhedron into a star-shaped one, relative to its centroid, after an "intersected" face is deleted, for a particular region. In this example, a line drawn from node *b* to the centroid intersects face *acd*, as it can be seen in the polyhedron on the left. This results in the removal of this face, as well as the tetrahedron formed by nodes *a*, *b*, *c* and *d*. The removal of this tetrahedron also implies in the removal of the faces *abd*, as well as in the inclusion of faces *bcd* and *abc*, resulting in the new polyhedron that can be seen on the center. This polyhedron can then be meshed and new tetrahedra are formed, as it can be seen on the right.

3.2 Three-dimensional algorithm

The algorithm for a three-dimensional problem, which represents the application of the general algorithm to the three-dimensional case, is as follows:

- Determine non-meshable polyhedron $\mathbf{P}(\mathbf{F_f})$ for the face $\mathbf{F_f}$.
- Perform visibility tests for polyhedron $\mathbf{P}(\mathbf{F_f})$.
- If there are intersections, update polyhedron $\mathbf{P}(\mathbf{F_f})$ by $\mathbf{P}(\mathbf{F_f})_{\mathbf{new}} = \mathbf{P}(\mathbf{F_f})_{\mathbf{old}} - \mathbf{F_f} - \mathbf{D}(\mathbf{F_f}) + \mathbf{A}(\mathbf{F_f})$, where $\mathbf{D}(\mathbf{F_f})$ is the set of deleted faces and $\mathbf{A}(\mathbf{F_f})$ is the

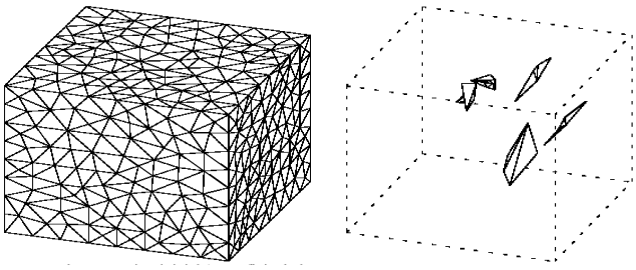

Figure 1: Hypothetical three-dimensional model: a) left: surface mesh, representing initial front; b) right: disconnected regions, representing non-meshable front.

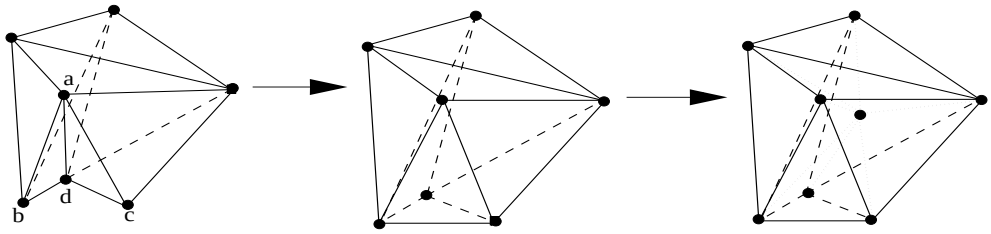

Figure 2: Transformation of a polyhedron into a convex one and generation of a mesh.

set of the included ones. Also update the current front and mesh lists by $\mathbf{F_{new}} = \mathbf{F_{old}} - \mathbf{F_f} - \mathbf{D}(\mathbf{F_f}) + \mathbf{A}(\mathbf{F_f})$ and $\mathbf{M_{new}} = \mathbf{M_{old}} - \mathbf{T_m}$, where $\mathbf{T_m}$ is the deleted tetrahedron associated with the deleted face $\mathbf{F_f}$.

- Repeat the process until a kernel is found. When this happen, find the set of new tetrahedra $\mathbf{G}(\mathbf{T_m})$ and update the mesh list $\mathbf{M}$ by $\mathbf{M_{new}} = \mathbf{M_{old}} + \mathbf{G}(\mathbf{T_m})$.

3.3 Determination of non-meshable polyhedra

One very important step in the back-tracking process is the determination of the disconnected region (polyhedron) associated with a particular front-element (face). The region has to form a closed loop, otherwise a valid mesh will not be generated for this region.

The normal and most common situation is the one illustrated by Figure 2. The polyhedron on the left represents the disconnected region. This region is found by the following procedure. Fist, the front-element (face) $\mathbf{E_f}$ is included in the region list of front-elements. This region is then formed by only one front-element at this point. After this, the other front-elements that are adjacent to $\mathbf{E_f}$ are also included in the region (that is, front-elements that share faces and edges in three dimensions). However, it is possible that some pathological situations can occur. Figure 3 illustrates situations where two regions are linked by one vertex (left) or edge (right). Although they could be treated as only one region in each case, it is indicated that the polyhedra linked by one vertex or edge be split into two and treated in a separate way. This is done by examining the adjacency of the faces. This procedure helps to improve the convergence of the back-tracking process.

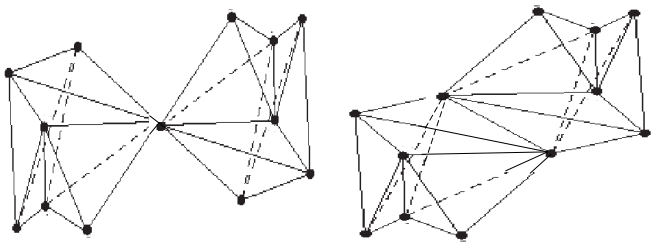

Figure 3: Two pathological situations: a) left: two polyhedra linked by one vertex; b) right: two polyhedra linked by one edge.

4. THREE-DIMENSIONAL EXAMPLES

In this Section, a study of the robustness of the proposed strategy is presented. Also, although the objective of this work is only to propose a technique to improve the robustness of advancing-front based three-dimensional algorithms, it is shown that the technique produces meshes with good quality.

4.1 Validation studies

In this study, three models are considered: a portion of a housing, a portion of a spiral-bevel gear and a portion of a turbofan hub. These models are shown through Figs. 4-12. All three examples contain small surface cracks. In a simple way, cracks can be seen as flaws that happen in some structures. The presence of cracks implies that a more complex modeling is necessary. Cracks are usually idealized as having no volume, that is, the surfaces representing the two sides of a crack are topologically distinct, but geometrically coincident. This means that nodes on opposite sides of crack faces may have identical coordinates. Moreover, the mesh on the crack's region is more refined, that is, it is common for the elements near the crack front to be two or three orders of magnitude smaller than other elements in the problem. The existence of cracks is only to show that the proposed technique can handle any constraints on the generated mesh, in the same way that it would do with "normal" meshes. Also, it shows that the technique can improve robustness of the mesh generation even for complex models.

Table 1 shows some statistics related to the number of disconnected regions that occurred for the examples and the time spent to process them by the back-tracking technique, in order to generate the mesh. It can be seen that, although the number of regions can be reasonably high, they are processed in a rapid way. The reason is that, since they are local regions with few elements, it is very fast to deal with them. The times were measured using a 1.333 GHz AMD Athlon(tm) processor.

4.2 Quality measurements

Although the objective of this work is to propose a technique to improve the robustness of advancing-front based three-dimensional algorithms, it can be shown, for the same models, that the technique produces meshes with good quality. One possible problem, that could occur, would be the situation where so many faces would be deleted for

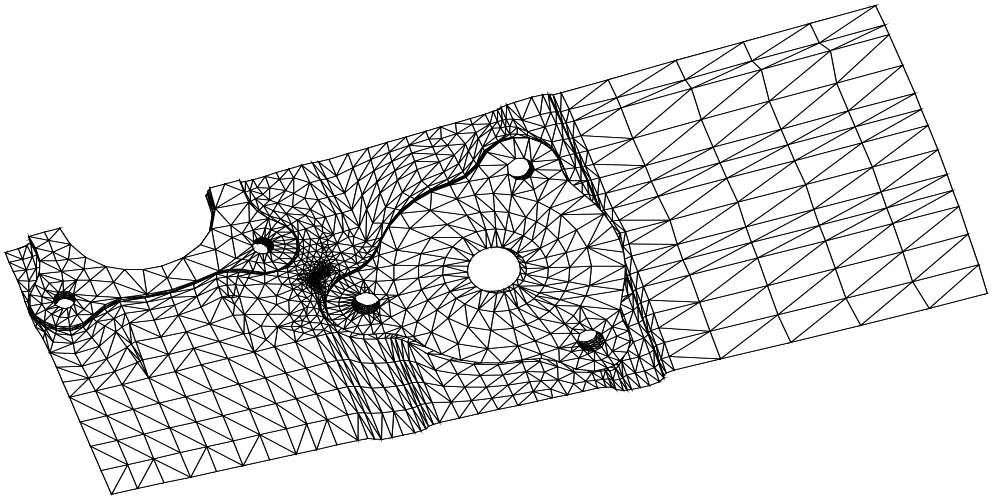

Figure 4: Mesh for housing example.

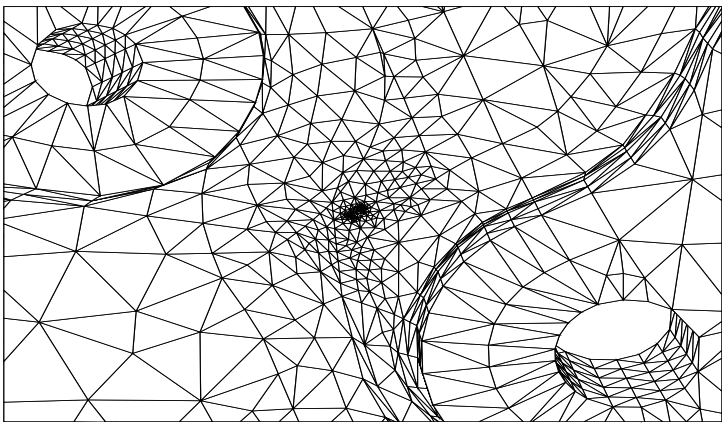

Figure 5: Mesh detail showing an embedded crack for housing example: detail of the crack region.

a particular region, creating a star-shaped region with a undesirable number of faces and many tetrahedra formed by faces linked to the very same node in the centroid. Although this is a possible scenario, this has not been observed because a mesh is usually generated after few steps of the back-tracking process.

For the evaluation of the final quality of the mesh, after the application of the back-tracking technique, a quality metric was adopted. The issue of quality metrics is well represented in the literature (Bramble and Zlamal, 1970; Ciarlet, 1970; Babuska and Aziz, 1976; Cavendish et al., 1995; Dannelongue and Tanguy, 1991; Joe, 1991; Parthasarathy et al., 1993; Weatherill and Hassan, 1994; Liu and Joe, 1994; Lewis et al., 1996; Freitag and Knupp, 1999), existing measures of all kinds. In the present work The metric α is used, which is defined by:

$$\alpha = \frac{3R_i}{R_c}, \tag{1}$$

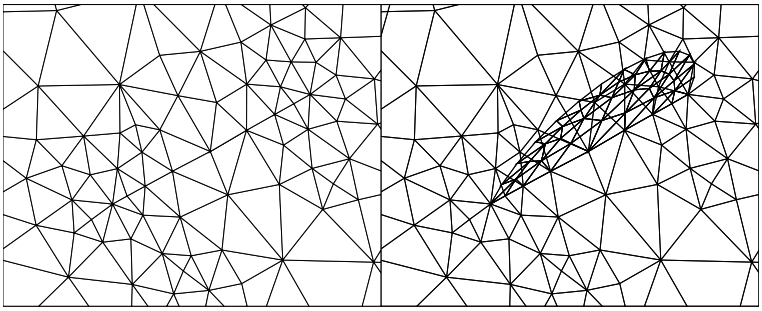

Figure 6: Mesh detail showing an embedded crack for housing example: a) left: mesh of the crack region; b) right: mesh of the crack region with the crack face mesh.

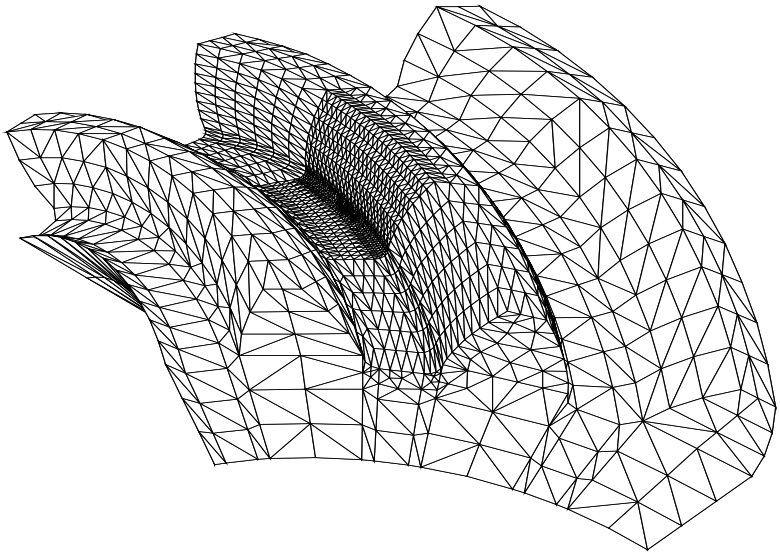

Figure 7: Mesh for gear example.

where R_i and R_c are the radii of the inscribed and circumscribed sphere, respectively. This metric is equal to 1.0 for an optimal element and sliver elements have values lower than 0.1, in general. The α metric is adopted for the evaluation of the quality of the final meshes because its interpretation is more intuitive and it is widely referenced in the literature. Table 2 shows some statistics related to α for the examples. It is shown that the majority of the elements are located in the range $[0.7, 0.8]$, which represents well-shaped elements.

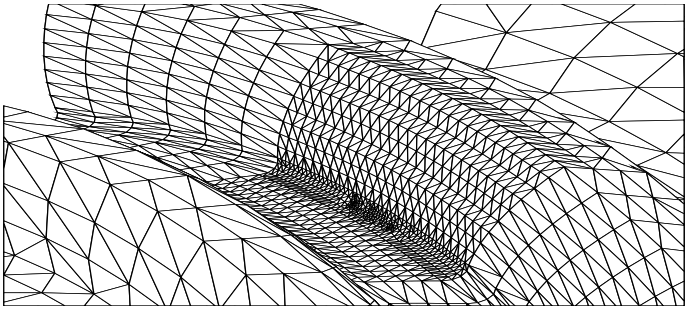

Figure 8: Mesh detail showing an embedded crack for gear example: detail of the crack region.

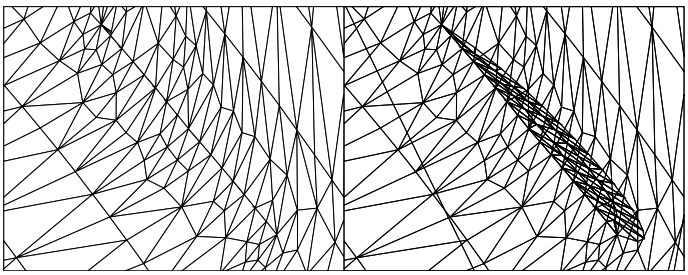

Figure 9: Mesh detail showing an embedded crack for gear example: a) left: mesh of the crack region; b) right: mesh of the crack region with the crack face mesh.

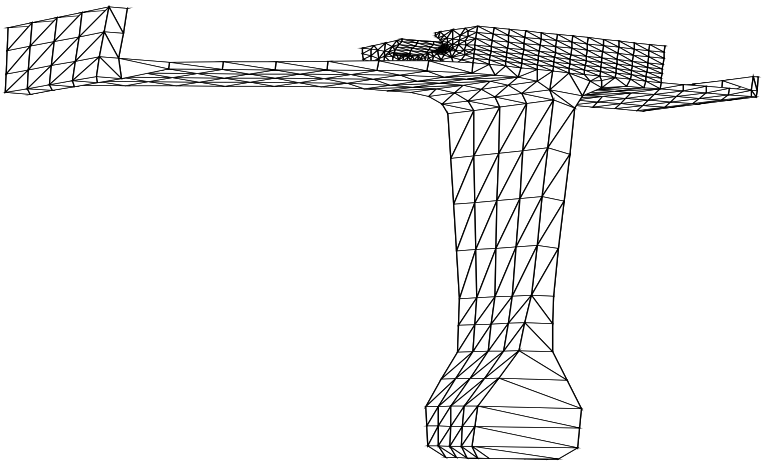

Figure 10: Mesh for turbofan example.

Table 1: Statistical values related to the number of regions for all examples

Example	# Regions	Time (sec)
Housing	438	16
Gear	573	23
Turbofan	345	14

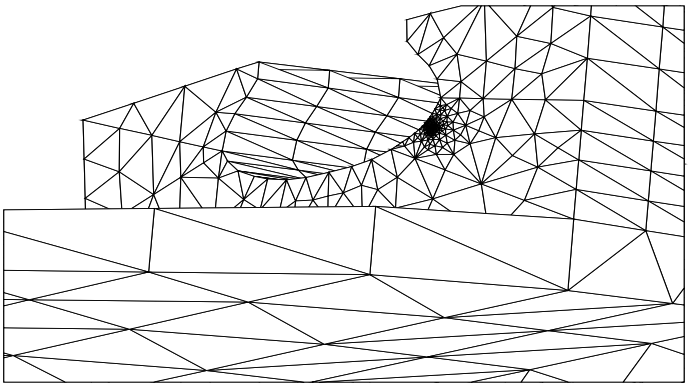

Figure 11: Mesh detail showing an embedded crack for turbofan example: detail of the crack region.

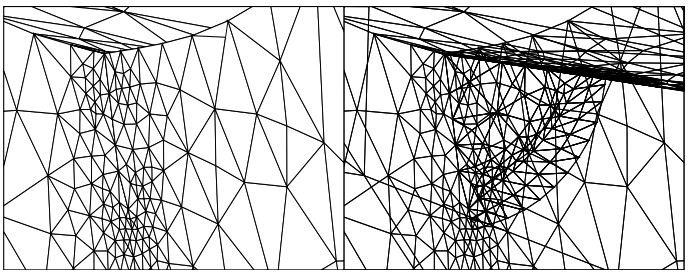

Figure 12: Mesh detail showing an embedded crack for turbofan example: a) left: mesh of the crack region; b) right: mesh of the crack region with the crack face mesh.

Table 2: Statistical values related to the quality of the mesh for all examples

Example	# Elements	α_{avg}	α_{min}	α_{max}
Housing	16463	0.675	0.025	0.729
Gear	17386	0.684	0.025	0.738
Turbofan	9628	0.668	0.018	0.733

5. CONCLUSIONS

This work presents a back-tracking strategy for improving the robustness of algorithms for advancing-front mesh generation in three-dimensional models. It can be observed, in many advancing-front based algorithms for three dimensions, that a number of regions that can not be meshed is left after the mesh generation process, which stops the algorithms from converging. These regions are represented by polyhedra and are formed by elements of the current front, which are faces in three dimensions. These regions are usually disconnected, since they are defined by adjacent elements that form a closed loop, and they can not be meshed, even with the insertion of additional new nodes. The back-tracking procedure is applied to all of these regions, in order to guarantee robustness in mesh generation.

The strategy of the procedure is to delete a front-element from a non-meshable region, generate a new region around this element and try to insert a new node in the centroid of this region, in order to generate new mesh-elements. If this region does not have a kernel, which is a core where any point is visible from a straight line from all its vertices to the centroid, this region must be expanded and the process repeated, until a kernel is found. This process tries to find a meshable region, represented by a star-shaped region (polyhedron in 3D), relative to its centroid, where new mesh-elements can be formed by the centroid and its vertices.

A number of realistic examples were shown to demonstrate the robustness of the proposed technique. Also, a distribution in the quality of the generated meshes was also shown.

Acknowledgements

The first author acknowledges the support from Brazilian agency CNPq through the grant 307148/2004-1.

REFERENCES

Babuska, I. & Aziz, A., 1976. On the angle condition in the finite element method. *SIAM Journal of Numerical Analysis*, vol. 13, n. 1, pp. 214–226.

Bramble, J. & Zlamal, M., 1970. Triangular elements in the finite element method. *The Mathematics of Computations*, vol. 24, n. 1, pp. 809–810.

Cavalcante-Neto, J., Wawrzynek, P., Carvalho, M., Martha, L., & Ingraffea, A., 2001. An algorithm for three-dimensional mesh generation for arbitrary regions with cracks. *Engineering with Computers*, vol. 17, n. 1, pp. 75–91.

Cavalcante-Neto, J., Wawrzynek, P., Martha, L., & Ingraffea, A., 2004. A back-tracking procedure for optimization of meshes. In *XXV Iberian Latin-American Congress on Computational Methods in Engineering (CILAMCE2004)*, pp. 1–14.

Cavendish, J., Field, D., & Frey, W., 1995. An approach to automatic three-dimensional finite element mesh generation. *International Journal for Numerical Methods in Engineering*, vol. 21, n. 1, pp. 329–347.

Chan, C. & Anastasiou, K., 1997. An automatic tetrahedral mesh generation scheme by the advancing front method. *Communications in Numerical Methods in Engineering*, vol. 13, n. 1, pp. 33–46.

Ciarlet, P., 1970. Orders of convergence in finite element method. *The Mathematics of Finite Element and Applications*, vol. 1, n. 1, pp. 59–82.

Dannelongue, H. & Tanguy, P., 1991. Three-dimensional adaptive finite element computations and applications to non-newtonian flows. *International Journal for Numerical Methods in Engineering*, vol. 13, n. 1, pp. 145–165.

Freitag, L. & Knupp, P., 1999. Tetrahedral element shape optimization via the jacobian determinate and condition number. In *8th International Mesh Roundtable*, pp. 247–258.

Joe, B., 1991. Delaunay versus max-min solid angle triangulation for three-dimensional mesh generation. *International Journal for Numerical Methods in Engineering*, vol. 31, n. 1, pp. 987–997.

Lewis, R., Zheng, Y., & Gethin, D., 1996. Three-dimensional unstructured mesh generation: Part 3. volume meshes. *Computer Methods in Applied Mechanics*, vol. 134, n. 1, pp. 285–310.

Liu, A. & Joe, B., 1994. Relationship between tetrahedron shape measures. *BIT*, vol. 34, n. 1, pp. 268–287.

Moller, P. & Hansbo, P., 1995. On advancing front mesh generation in three dimensions. *International Journal for Numerical Methods in Engineering*, vol. 38, n. 1, pp. 3551–3569.

Parthasarathy, V., Graichen, C., & Hathaway, A., 1993. A comparison of tetrahedron quality measures. *Finite Element Analysis and Design*, vol. 15, n. 1, pp. 255–261.

Rassineux, A., 1998. Generation and optimization of tetrahedral meshes by advancing front technique. *International Journal for Numerical Methods in Engineering*, vol. 41, n. 1, pp. 651–674.

Weatherill, N. & Hassan, O., 1994. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, vol. 37, n. 1, pp. 2005–2039.