A BACK-TRACKING PROCEDURE FOR OPTIMIZATION OF MESHES

Joaquim B. Cavalcante-Neto
joaquimb@lia.ufc.br
joaquimb@vienna.cfg.cornell.edu
Computer Graphics Group (CRAb), Federal University of Ceara (UFC), Brazil
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
Paul A. Wawrzynek
Anthony R. Ingraffea
wash@fac.cfg.cornell.edu
ari1@cornell.edu
Cornell Fracture Group (CFG), Cornell University, USA

Abstract. *This work presents a back-tracking procedure for optimization of two and three dimensional meshes. The Back-tracking procedure is applied to all ill-shaped elements in a mesh, in order to guarantee quality in mesh optimization. Basically, the original mesh is reconstructed in regions around the ill-shaped elements. The Back-tracking procedure is based on the following steps. The initial local region for remeshing is identified grouping the ill-shaped element with adjacent elements that share edges (in 2D) and faces and edges (in 3D) with it. Then, visibility tests related to the centroid of this region are done, in order to check whether this centroid is visible from all nodes on the boundary of the region. If any boundary node is not visible from the centroid, the region is expanded with elements that share edges (in 2D) and faces and edges (in 3D) with the elements that contain the non-visible nodes. The procedure is applied again for this expanded region and continues until all nodes on the boundary of the selected region are visible from the centroid. After a region for reconstruction is selected, new elements are generated, connecting the elements on the boundary of the region to the centroid. Since the region presents a convex shape, the remeshing results in well-shaped elements. The Back-tracking procedure fixes problems related to the quality of generated meshes in general. It can be applied in both two and three dimensions, although it can be shown that in two dimensions such problems are more difficult to occur because usually a smoothing procedure gets rid of ill-shaped elements. The procedure works deleting elements ranked below a predefined shape quality measure. It is important to mention that, although the presented procedure was devised for advancing-front algorithms, it can also be used for meshes generated by different approaches, such as Delaunay based and others, with no modifications. Examples of generated meshes using the Back-tracking procedure are presented, in which the quality of these meshes are assessed in order to validate the procedure proposed in this work.*

Keywords: *Mesh Generation, Mesh Optimization, Adaptivity*

1. INTRODUCTION

This work presents a back-tracking procedure for optimization of two and three dimensional meshes. The Back-tracking procedure is applied to all ill-shaped elements in a mesh, in order to guarantee quality in mesh optimization.

There are a lot of techniques in the literature that have addressed, in some way, the problem of optimization of meshes. There is a wide variety of solutions, from simple to very complicated ones, but in the overall the idea is somehow improve the quality of the final mesh generated. Some of these solutions are applied *a priori*, that is, they try to improve the mesh quality before its generation. For instance, improving the surface mesh, which usually is the input for the volume mesh generation, in a three-dimensional advancing front technique, is a type of *a priori* solution. However, most of the solutions to improve mesh quality are performed *a posteriori*, that is, after the mesh is generated. The idea in the *a posteriori* approaches is to try to remove or change, if possible, any remaining, poorly-shaped element in the generated mesh, replacing them by better ones.

One example of a technique for optimization of meshes, is the so-called Mesh smoothing technique, which is probably the most used and and one of the most effective techniques found in the literature. This technique is employed usually *a posteriori*, and consists basically of local mesh modifications implemented to improve mesh quality. Mesh smoothing usually increases the quality of the mesh as well as smoothes the transition among elements of various sizes. In general, there are two common methods used in mesh smoothing, the Laplacian smoothing (Field, 1988; Hansbo, 1995; Freitag and Ollivier-Gooch, 1997; Cavalcante-Neto et al., 2001) and the Optimal smoothing (Borouchaki and George, 1997). Also, there are variations of them, such as the weighted Laplacian smoothing approach, where the smoothing is performed by relocating interior nodes within a patch, using a weight called the relaxation coefficient (Lewis et al., 1996). This can be very important and used very effectively, specially in three dimensions (Cavalcante-Neto et al., 2001). This weight is necessary because the smoothing technique really improves quality in two dimensions, but in three dimensions this is more difficult to guarantee. Also, one additional problem is that the relocation of the interior nodes can occasionally result in invalid elements with negative volumes, even with the use of weights. Because of this, in order to ensure that this technique is really effective, additional procedures to check if the relocation of a node still keeps the integrity of the mesh are necessary.

Another example of a very used technique for optimization of meshes is the technique based on the swapping of entities in a mesh, in order to improve its quality (Freitag and Ollivier-Gooch, 1997). These entities are usually edges and/or faces. This technique is very effective, but since only local changes are performed, considering that only local connectivity is affected, really bad meshes, specially in three dimensions, may not be improved sufficiently.

The idea of deleting elements or regions to improve mesh quality has also been addressed in the literature recently. For instance, the work of Chan and Anastasiou (1997) 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.

The technique proposed in this work, called Back-tracking, is a new procedure devised to improve mesh quality in two and three dimensions. This procedure is applied to all ill-shaped elements in a mesh, in order to guarantee quality in mesh optimization. Basically, the original mesh is reconstructed in regions around the ill-shaped elements. In summary, the Back-tracking procedure is based on the following steps. The initial local region for remeshing is identified grouping the ill-shaped element with adjacent elements that share edges (in 2D) and faces and edges (in 3D) with it. Then, visibility tests related to the centroid of this region are done, in order to check whether this centroid is visible from all nodes on the boundary of the region. If any boundary node is not visible from the centroid, the region is expanded with elements that share edges (in 2D) and faces and edges (in 3D) with the elements that contain the non-visible nodes. The procedure is applied again for this expanded region and continues until all nodes on the boundary of the selected region are visible from the centroid. After a region for reconstruction is selected, new elements are generated, connecting the elements on the boundary of the region to the centroid. Since the region presents a convex shape, the remeshing results in well-shaped elements.

The Back-tracking procedure fixes problems related to the quality of generated meshes in general. It can be applied in both two and three dimensions, although it can be shown that in two dimensions such problems are more difficult to occur because usually a smoothing procedure gets rid of ill-shaped elements. The procedure works deleting elements ranked below a predefined shape quality measure. It is important to mention that, although the presented procedure was devised for advancing-front algorithms, it can also be used for meshes generated by different approaches, such as Delaunay based and others, with no modifications.

The remainder of this work is organized as follows. Section 2 describes the Back-tracking strategy in a general manner and its algorithm. Section 3 discusses the application of this strategy in two-dimensional problems. The next Section brings the same discussion but related to three-dimensional problems. Section 5 shows validation studies that attest the robustness of the proposed technique. Finally, in Section 6 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. This strategy will later be described specifically for two and three dimensional problems with their specific requirements and details.

2.1 General description

The idea of the strategy is to delete an ill-shaped element, generate a new region around this element and try to insert a new node in the centroid of this region, in order to generate better shape new 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 non-meshed region, represented by a star-shaped polygon in 2D (or polyhedron in 3D), relative to its centroid, where new elements with better shape can be formed by the centroid and its vertices.

Let $\mathbf{E_k}$ be an ill-shaped element of the generated mesh and $\mathbf{R}(\mathbf{E_k})$ the local region for remeshing in the neighborhood of $\mathbf{E_k}$, which is identified grouping the ill-shaped element with adjacent elements that share edges (in 2D) and faces and edges (in 3D) with it. Let $\mathbf{M}(\mathbf{E})$ be the set of all elements on the current mesh (before the back-tracking modification) and $\mathbf{D}(\mathbf{E_k})$ the set of elements deleted during the back-tracking process, if necessary. Finally, let $\mathbf{F}(\mathbf{E_k})$ be the set of new elements generated for the local region $\mathbf{R}(\mathbf{E_k})$, when a kernel for this region is found. The procedure used to transform an ill-shaped region into one with a visible kernel is as follows:

- The boundary of the local region $\mathbf{R}(\mathbf{E_k})$ related to the current ill-shaped element $\mathbf{E_k}$ is identified.

- A visibility test is performed. This consists of computing the coordinates of the center of this region $\mathbf{R}(\mathbf{E_k})$ and counting the number of intersections that would occur, for each of the 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 elements, this region must be modified. This is done by removing the element(s) attached to the element that has the highest number of intersections. These elements form the set $\mathbf{D}(\mathbf{E_k})$ for the current region. This generates a new region $\mathbf{R}(\mathbf{E_k})$. Also, the set of all elements on the current mesh $\mathbf{M}(\mathbf{E})$ must be updated to account for the elements deleted. The deletion of the elements represented by $\mathbf{D}(\mathbf{E_k})$ usually results in the growth of $\mathbf{R}(\mathbf{E_k})$, since this deletion increases the region that needs to be remeshed with better elements.

- This process is repeated until the center of the region $\mathbf{R}(\mathbf{E_k})$ is visible from all vertices of this region. When this happen, the set of new elements $\mathbf{F}(\mathbf{E_k})$ are generated for the region $\mathbf{R}(\mathbf{E_k})$, by connecting the center to all region's vertices. The set of all elements on the current mesh $\mathbf{M}(\mathbf{E})$ must be updated to account for these new elements $\mathbf{F}(\mathbf{E_k})$ generated.

It is possible that the process of findind a star-shaped region, relative to its centroid, may fail if faces to be removed are part of the original boundary mesh. When this occurs, the 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, and the original mesh, "saved" in the beginning of the process for this particular region, is restored. 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 5 and the other ones tested so far.

Another important observation is that it is not guaranteed that the Back-tracking procedure will always improve the mesh quality, specially in 3D, since it is known that the insertion of points inside a star-shaped polyhedra can worsen the mesh, in some cases. If this happens, the original mesh for the current region is also restored. It is important to mention, however, that in the majority of the examples it was attested a significant improvement of the mesh quality.

2.2 General algorithm

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

- Determine region $\mathbf{R}(\mathbf{E_k})$ for the ill-shaped element $\mathbf{E_k}$.

- Perform visibility tests for region $\mathbf{R}(\mathbf{E_k})$.

- If there are intersections, update region $\mathbf{R}(\mathbf{E_k})$ by $\mathbf{R}(\mathbf{E_k})_{\mathbf{new}} = \mathbf{R}(\mathbf{E_k})_{\mathbf{old}} - \mathbf{D}(\mathbf{E_k})$. Also, update the current mesh $\mathbf{M}(\mathbf{E})$ by $\mathbf{M}(\mathbf{E})_{\mathbf{new}} = \mathbf{M}(\mathbf{E})_{\mathbf{old}} - \mathbf{D}(\mathbf{E_k})$.

- Repeat the process until a kernel is found. When this happen, find the set of new elements $\mathbf{F}(\mathbf{E_k})$ and update $\mathbf{M}(\mathbf{E})$ by $\mathbf{M}(\mathbf{E})_{\mathbf{new}} = \mathbf{M}(\mathbf{E})_{\mathbf{old}} + \mathbf{F}(\mathbf{E_k})$.

3. TWO-DIMENSIONAL BACK-TRACKING

This Section describes the details for the application of the proposed technique to two-dimensional problems. Basically, the entities involved in the strategy are faces, with their respective roles.

3.1 Strategy applied to 2D models

The Back-tracking procedure consists of deleting an element that is classified as poorly shaped element and a group of elements in its vicinity. In two-dimensional models, the element is represented by a face (triangle). The classification of a “bad” face is based on a specified metric for shape of triangles. For each face of the generated mesh, the quality measure is evaluated. If the value of this metric is below a threshold, the face is classified as a poorly shaped triangle.

The objective of the Back-tracking procedure is to delete faces surrounding a “bad” triangle to create a local region, in this case a polygon, that can be remeshed with better shaped triangles. This local polygon to be meshed is created by deleting all triangles adjacent to the “bad” element. This is ilustrated by means of Fig. 1, which shows a two-dimensional case. In this picture, it is shown the “bad” triangle and the polygon associated with it.

After the criation of the local polygon, an attempt is made to generate triangles by inserting a new internal node in the polygon’s center, as also shown in Fig. 1. Is this does not work, the Back-tracking procedure described in Section 2 is employed.

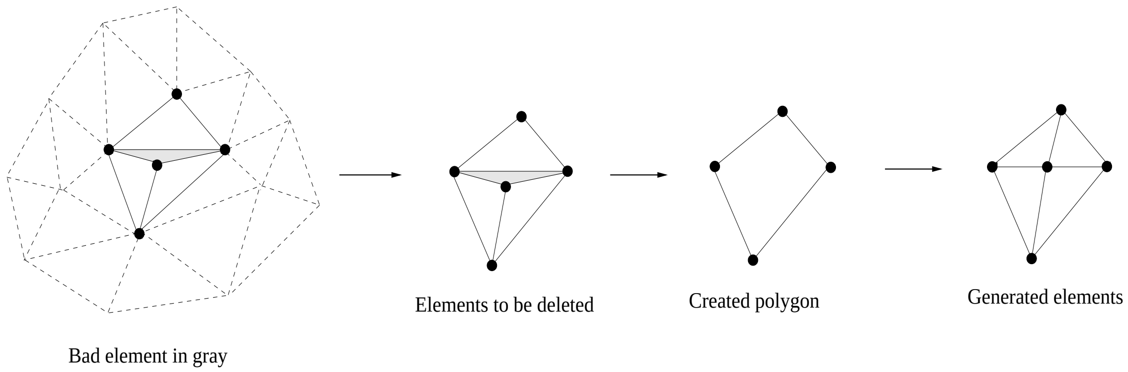

Figure 1: Two-dimensional Back-tracking procedure around a “bad” face.

3.2 Two-dimensional algorithm

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

- Determine polygon $\mathbf{P}(\mathbf{T_k})$ for the ill-shaped triangle $\mathbf{T_k}$.
- Perform visibility tests for polygon $\mathbf{P}(\mathbf{T_k})$.
- If there are intersections, update polygon $\mathbf{P}(\mathbf{T_k})$ by $\mathbf{P}(\mathbf{T_k})_{\mathbf{new}} = \mathbf{P}(\mathbf{T_k})_{\mathbf{old}} - \mathbf{D}(\mathbf{T_k})$, where $\mathbf{D}(\mathbf{T_k})$ is the set of deleted triangles. Also, update the current mesh $\mathbf{M}(\mathbf{T})$ by $\mathbf{M}(\mathbf{T})_{\mathbf{new}} = \mathbf{M}(\mathbf{T})_{\mathbf{old}} - \mathbf{D}(\mathbf{T_k})$.
- Repeat the process until a kernel is found. When this happen, find the set of new triangles $\mathbf{F}(\mathbf{T_k})$ and update $\mathbf{M}(\mathbf{T})$ by $\mathbf{M}(\mathbf{T})_{\mathbf{new}} = \mathbf{M}(\mathbf{T})_{\mathbf{old}} + \mathbf{F}(\mathbf{T_k})$.

4. 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 now tetrahedra, with their respective roles.

4.1 Strategy applied to 3D models

The Back-tracking strategy here is similar to the two-dimensional case, with the only differences that the elements now are tetrahedra and not triangles and the local region to be remeshed is now represented by a polyhedron and not a polygon. It is important to mention, however, that in three-dimensions it is usually much more difficult to find the kernel of the polyhedron, and sometimes it may not even exist. Because of this, it is important to “save” the original configuration of the mesh for a particular ill-shaped element, in case the kernel can not be found after a number of steps and then nothing can be done to improve the quality of this element. The number of steps is defined empirically and in this work is 10. This number is important to avoid an undesirable number of elements adjacent to the center of the region that might occur. The same care can be taken in two-dimensions, but usually the process converges well below 10 steps.

Figure 2 illustrates the transformation of a polyhedron into a star-shaped one, relative to its centroid, after an “intersected” element is deleted. In this example, a line drawn from node *b* to the centroid instersects face *acd*. This results in the removal of the tetrahedron formed by nodes *a*, *b*, *c* and *d*. The resulting polyhedron is shown in the center of the Fig. 2. This figure also shows the tetrahedral elements generated after the insertion of a node at the centroid of the new polyhedron.

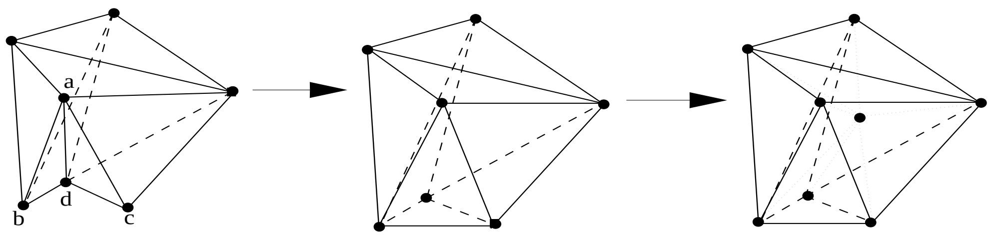

Figure 2: Transformation of a polyhedron into a convex one.

4.2 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 polyhedron $\mathbf{P}(\mathbf{T_k})$ for the ill-shaped tetrahedron $\mathbf{T_k}$.
- Perform visibility tests for polyhedron $\mathbf{P}(\mathbf{T_k})$.
- If there are intersections, update polyhedron $\mathbf{P}(\mathbf{T_k})$ by $\mathbf{P}(\mathbf{T_k})_{\mathbf{new}} = \mathbf{P}(\mathbf{T_k})_{\mathbf{old}} - \mathbf{D}(\mathbf{T_k})$, where $\mathbf{D}(\mathbf{T_k})$ is the set of deleted tetrahedra. Also, update the current mesh $\mathbf{M}(\mathbf{T})$ by $\mathbf{M}(\mathbf{T})_{\mathbf{new}} = \mathbf{M}(\mathbf{T})_{\mathbf{old}} - \mathbf{D}(\mathbf{T_k})$.
- Repeat the process until a kernel is found. When this happen, find the set of new tetrahedra $\mathbf{F}(\mathbf{T_k})$ and update $\mathbf{M}(\mathbf{T})$ by $\mathbf{M}(\mathbf{T})_{\mathbf{new}} = \mathbf{M}(\mathbf{T})_{\mathbf{old}} + \mathbf{F}(\mathbf{T_k})$.

5. VALIDATION STUDIES

In this Section, a study of the quality of the meshes generated by the proposed technique is presented. Also, this Section discusses the quality measures used in this work.

5.1 Quality measures

In the proposed Back-tracking technique, the local mesh improvement procedures imply that element shape quality measures are necessary. These measures will define a threshold used to characterize if an element is considered ill-shaped. This issue 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.

Considering that in three dimensions is more difficult to guarantee that a high quality mesh is generated for these type of applications, the examples shown are three-dimensional ones, in order to attest the efficiency of the proposed technique. Because of this, the metric adopted in this work is a normalized ratio between the root mean square of the lengths of the edges of a tetrahedron, represented by:

$$S_{rms} = \sqrt{\tfrac{1}{6}\textstyle\sum_{i=0}^{5} S_i^2}, \tag{1}$$

where S_i is the length of and edge, and the volume V of the tetrahedron (Weatherill and Hassan, 1994):

$$\gamma = \frac{S_{rms}^3}{V}. \tag{2}$$

This metric generates a good quality measure and is computationally efficient. The range of valid values varies from one to infinity ($[1, \infty)$), and the optimal value for the regular tetrahedron is approximately 8.5. For each element considered to check if it is poorly shaped, this metric is evaluated. If its value is outside a pre-defined range, the element is classified as ill-shaped. The lower and upper bounds of this range are defined empirically, based on experiments and observations. In this work, the lower bound value is 5.0 and the upper bound is the “optimal” metric value of 8.5 multiplied by a factor of 30.

It is important to mention, however, that any quality measure can be used, since all the strategy needs is some metric to rank the elements and to indicate if a particular element is ill-shaped or not. Moreover, the same concept applies for two-dimensional problems, that is, it should be used a metric to define the ill-shaped two-dimensional elements of the mesh.

For the evaluation of the final quality of the mesh, after the application of the Back-tracking technique, a different metric was adopted. The metric α is used, which is defined by:

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

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, instead of the γ metric used to identify the ill-shaped elements, is adopted for the evaluation of the quality of the final meshes because its interpretation is more intuitive and it is more widely referenced in the literature. But again, any quality measure can be used for this type of study. Moreover, measures that are specific for two-dimensions can be used to evaluate the quality of two-dimensional meshes as well.

5.2 Three-dimensional examples

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. 3-11. 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.

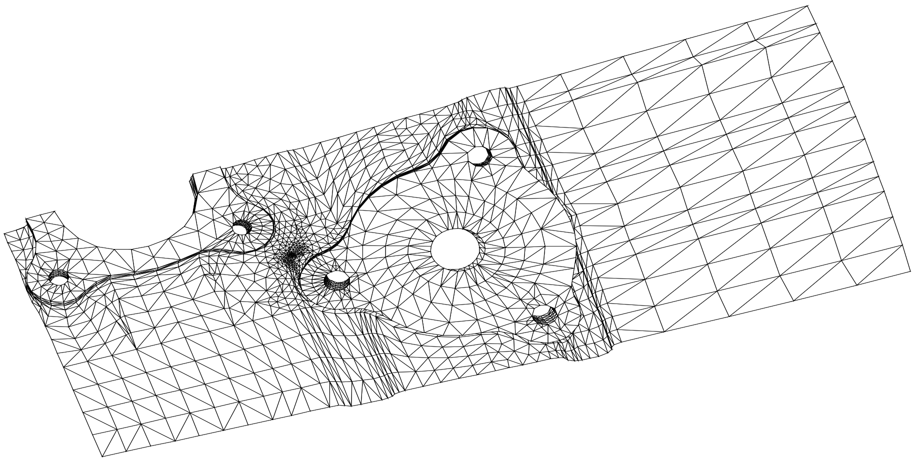

Figure 3: Mesh for housing example.

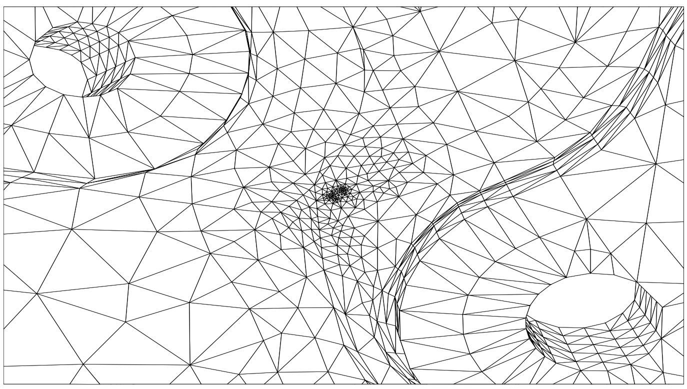

Figure 4: Mesh detail showing an embedded crack for housing example: detail of the crack region.

Figures 12-14 show histograms of the number of the elements generated in various ranges of α for all three models. These histograms show the distributions both before (left bar) and after (right bar) the application of the Back-tracking procedure. Table 1 shows some statistics related to α for the examples.

The histograms show that the majority of the elements are located in the range $[0.7, 0.8]$, which represents well-shaped elements, even before the application of the Back-tracking pro-

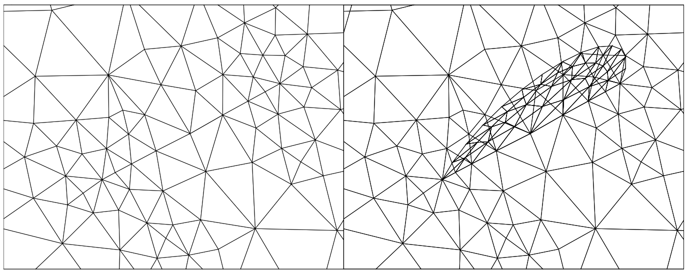

Figure 5: Mesh detail showing an embedded crack for housing example: a) left: mesh of the crack region; b) right: mesh of the crack region with the crack face mesh.

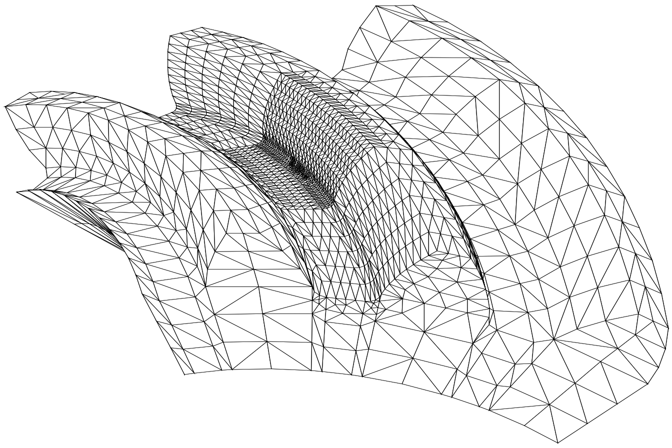

Figure 6: Mesh for gear example.

cedure. These meshes were generated by an advancing front based algorithm (Cavalcante-Neto et al., 2001) which already takes care of quality during the mesh generation. However, the number of elements in this range increases after the application of the Back-tracking procedure in a order from 2% to 4%. Although it might seem a small percentage, it must be considered that this percentage can represents a great number of elements in big meshes. Regarding the poorly-shaped elements, on the other hand, there are a significant number of elements $(1.76 - 3.77\%)$ with α values lower than 0.1, which represent undesirable elements. After the application of the Back-tracking procedure, the number of poorly-shaped elements drops significantly $(0.49 - 0.90\%)$.

Table 1: Statistical values related to the quality of the mesh for all examples

Example	Histogram	# Elements	α_{avg}	α_{min}	α_{max}
Housing	Before	16463	0.675	0.025	0.729
Housing	After	17043	0.696	0.023	0.740
Gear	Before	17386	0.684	0.025	0.738
Gear	After	16990	0.699	0.033	0.742
Turbofan	Before	9628	0.668	0.018	0.733
Turbofan	After	10046	0.692	0.022	0.741

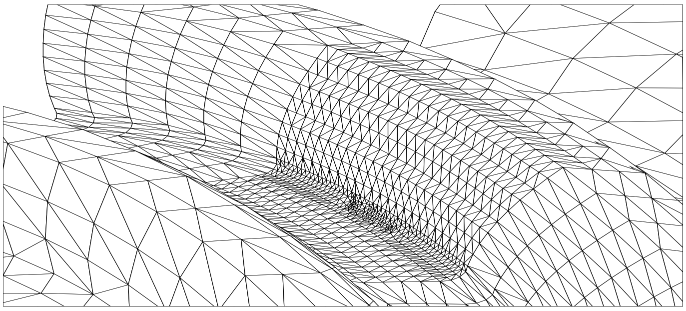

Figure 7: Mesh detail showing an embedded crack for gear example: detail of the crack region.

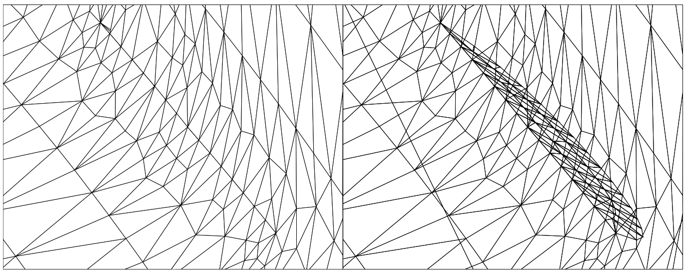

Figure 8: Mesh detail showing an embedded crack for gear example: a) left: mesh of the crack region; b) right: mesh of the crack region with the crack face mesh.

6. CONCLUSIONS

A back-tracking procedure for optimization of two and three dimensional meshes was presented. The Back-tracking procedure is applied to all ill-shaped elements in a mesh, in order to guarantee quality in mesh optimization.

The strategy of the procedure is to delete an ill-shaped element, generate a new region around this element and try to insert a new node in the centroid of this region, in order to generate better shape new 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 non-meshed region, represented by a star-shaped polygon in 2D (or polyhedron in 3D), relative to its centroid, where new elements with better shape can be formed by the centroid and its vertices.

A number of realistic examples were shown to demonstrate the distribution in the quality of the optimized meshes. It was shown that only a small percentage ($0.49 - 0.90\%$) of the total elements are poorly shaped, after the application of the Back-tracking procedure. Also, it was shown that the presented procedure also improves the percentage of elements considered of very good shape.

Acknowledgements

The first author acknowledges fellowship provided by Brazilian agency CNPq, since part of this work was developed during his position as a Scientific and Regional Development Researcher at the Federal University of Ceara, where he is now an Associate Professor.

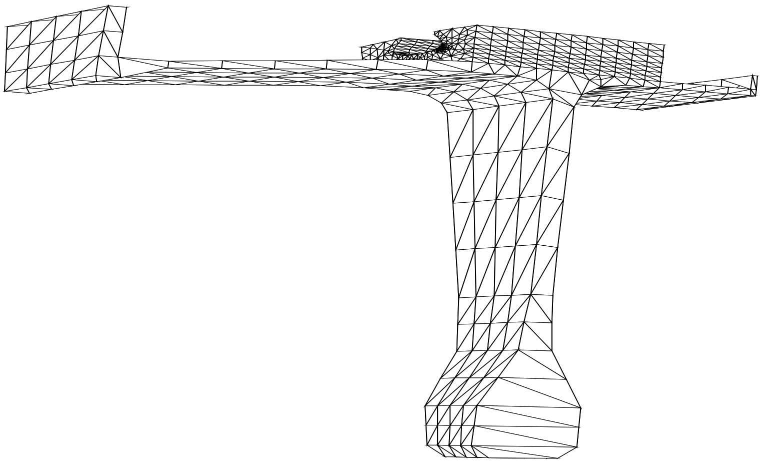

Figure 9: Mesh for turbofan example.

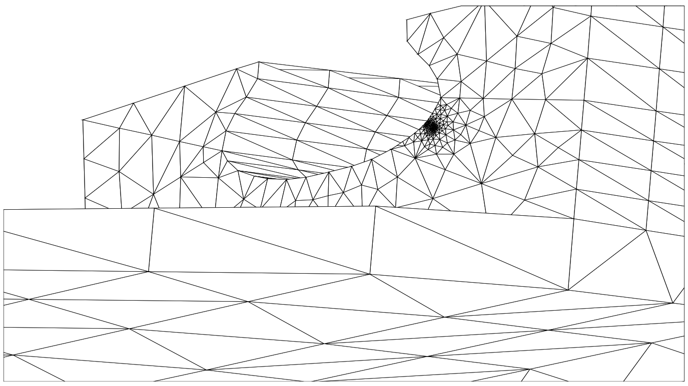

Figure 10: Mesh detail showing an embedded crack for turbofan example: detail of the crack region.

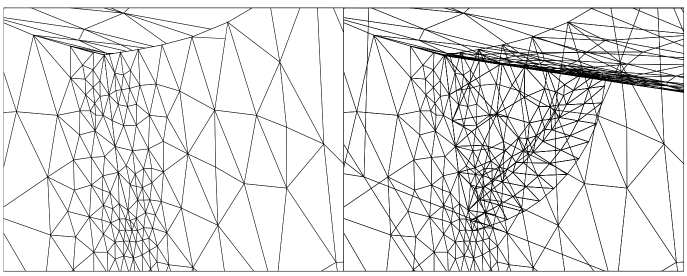

Figure 11: Mesh detail showing an embedded crack for turbofan example: a) left: mesh of the crack region; b) right: mesh of the crack region with the crack face mesh.

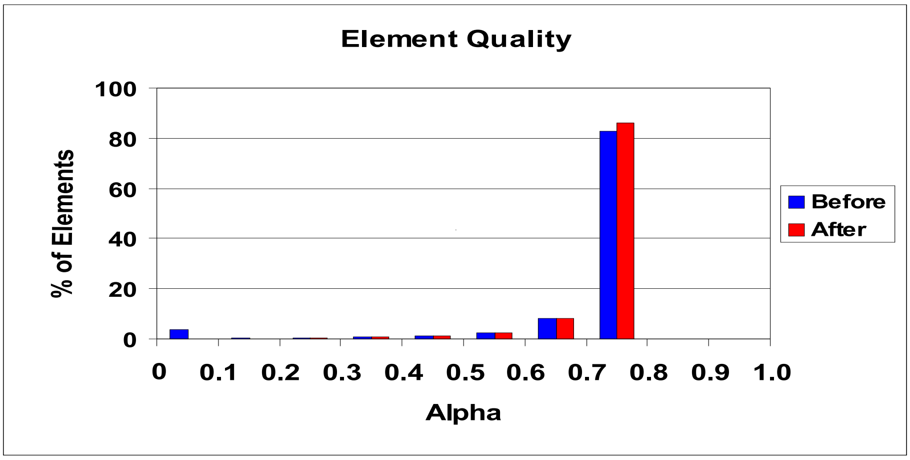

Figure 12: Histogram of element quality for housing example.

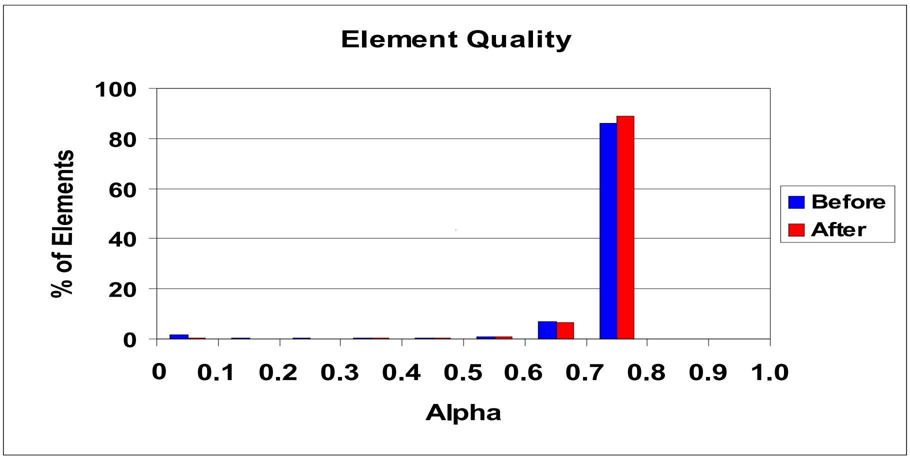

Figure 13: Histogram of element quality for gear example.

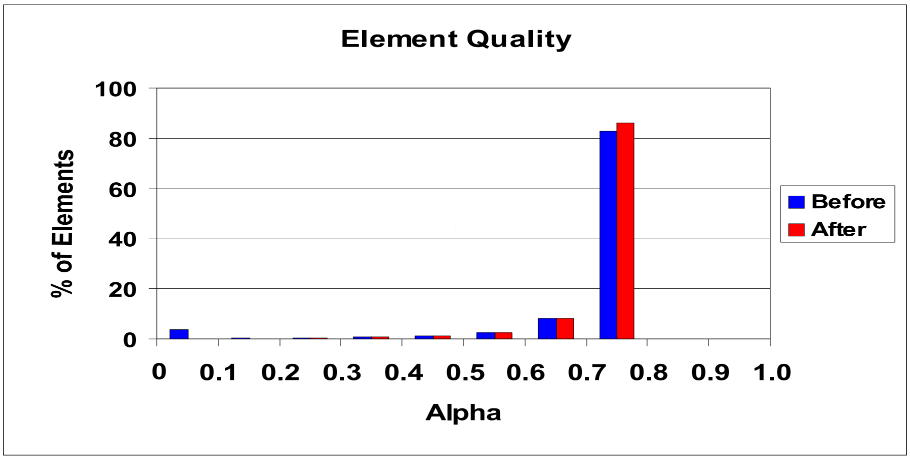

Figure 14: Histogram of element quality for turbofan example.

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.

Borouchaki, H. & George, P., 1997. Aspects of 2-d delaunay mesh generation. *International Journal for Numerical Methods in Engineering*, vol. 40, n. 1, pp. 1957–1975.

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.

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.

Field, D., 1988. Laplacian smoothing and delaunay triangulation. *Communications in Applied Numerical Methods*, vol. 4, n. 1, pp. 709–712.

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

Freitag, L. & Ollivier-Gooch, C., 1997. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, vol. 40, n. 1, pp. 3979–4002.

Hansbo, P., 1995. Generalized laplacian smoothing of unstructured grids. *Communications in Numerical Methods in Engineering*, vol. 11, n. 1, pp. 455–464.

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.

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.