# Re-triangulation of existing surface meshes with high curvatures 

Antonio Carlos MIRANDA, Luiz MARTHA*, Paul WAWRZYNEK, Anthony INGRAFFEA<br>*Department of Civil Engineering and Tecgraf - Computer Graphics Technology Group<br>Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro RJ 22453-900, Brazil<br>lfm@tecgraf.puc-rio.br


#### Abstract

This work describes an automatic algorithm for unstructured mesh regeneration on arbitrarily shaped threedimensional surfaces. The arbitrary surface may be: a triangulated mesh, a set of points, or an analytical surface (such as a collection of NURBS patches). To be generic, the algorithm requires the implementation of three abstract methods. The first, given a point location, returns the desired characteristic size of a triangular element at this position. The second method, given the current edge in the boundary contraction algorithm, locates the ideal apex point that forms a triangle with this edge. And the third method, given a point in space and a projection direction, returns the closest point on the geometrical supporting surface. This work also describes the implementation of these three methods to re-mesh an existing triangulated mesh that might present regions of high curvature. In order to test the efficiency of the proposed algorithm of surface mesh generation and implementation of the three abstract methods, results of performance and quality of generated triangular element examples are presented.


## 1. Introduction

This paper describes an algorithm for generating triangle finite-element meshes on surfaces of arbitrary shape and curvatures. It is an extension of a previously proposed algorithm (advancing-front technique) for generating unstructured meshes in three-dimensional [1, 2] and two-dimensional [3, 4] domains. Algorithms have been described in the literature previously that generate 3D surface meshes by re-meshing existing triangulations [57], or by using analytical surface descriptions for geometrical support [8-10]. The present algorithm generalizes the type of geometrical support that may be used by making use of three generic functions that abstractly represent the supporting surface. In a particular case, this paper also describes an implementation of the three generic methods for re-meshing existing surface triangulations and demonstrates the quality of the generated meshes.

To facilitate a smooth transition between regions and regions of high curvatures, the present algorithm employs an octree data structure. However, unlike some authors, e.g. Rassineux [11], who use a quadtree/octree procedure to generate internal nodes prior to element generation, here it is used to provide local policies used to define the discretization of the surface mesh and to define the sizes of triangular elements to be generated during the advancing-front procedure. The authors feel that this approach tends to provide better control over the quality of the generated mesh and to decrease the number of heuristic, clean-up procedures.

## 2. Description of the algorithm

The input data for the present algorithm is a polygonal description of the boundary of the surface patch to be meshed and a 3D supporting surface that is abstractly represented by three generic methods defined as follow:
$\mathbf{1}^{\text {st }}$ Method - Given a point location, the method returns the desired characteristic size of an ideal equilateral triangular element at this position. The length of the triangle's side is considered as its characteristic size.
$\mathbf{2}^{\text {nd }}$ Method - Given the current base edge in the boundary contraction algorithm, the method locates the ideal apex point that forms a new triangle. The method has two input arguments: the height of the candidate equilateral triangle and a unit vector in the edge perpendicular direction. This unit vector is a "surface intersection direction" that defines a plane perpendicular to the base edge at its mid point. These arguments are used to determine an optimal triangle apex point location.
$\mathbf{3}^{\text {rd }}$ Method - Given a point in space, the method returns the closest point on the geometrical supporting surface. The method also receives a surface projection vector as additional information. This method is used only in the final stage of the surface mesh generation for local mesh improvement.

### 2.1 Advancing-front procedure

In this algorithm, as in its ancestors [1-4], the advancing-front process is divided into two phases to ensure the generation of valid triangulations. In the first phase, a geometry-based element generation is pursued to generate elements of optimal shapes. After this ideal phase is exhausted and no more optimal elements can be generated, a topology-based element generation takes place, creating valid, but not necessarily well shaped, elements in the remaining region. The required steps for the advancing-front procedure are as follows.

### 2.1.1 Front contraction (geometry-based element generation)

Ideally, the entire mesh will be generated in the geometry-based phase. In this phase, for each base edge on the advancing front, the following is performed:

- The optimal location $\mathbf{N} \mathbf{1}$ for the vertex of an equilateral triangle to be formed is determined employing the $\mathbf{1}^{\text {st }}$ and $2^{\text {nd }}$ Methods. Using the base edge middle point (M) as input, $\mathbf{1}^{\text {st }}$ Method returns the target triangle characteristic size (the length of the equilateral triangle side) at this point location. With this size, the height of the candidate triangle is obtained. Then, using the Normal x Tangent vector at the middle point $\mathbf{M}$ as the required unit vector (surface intersection direction) in the edge perpendicular direction and the triangle height, $\mathbf{2}^{\text {nd }}$ Method returns the desired $\mathbf{N} \mathbf{1}$ location on the support surface.
- If no existing node is inside the optimal region, a new node is inserted at the optimal location $\mathbf{N} \mathbf{1}$ and an element is generated using this node. If only one node exists in the region, this node is used to generate the element.
- Additional geometric checks are performed to ensure that the edges of the new triangular element do not intersect any existing edge of the advancing front and that the new triangle apex does not lie inside any other existing triangle. In both cases, the new element is rejected. These checks are not trivial in 3D space. However, they are performed in a local 2D system on the plane of the new triangular element, avoiding complex geometry checks in 3D space. The new edges and adjacent elements edges are transformed to the local 2D system on the plane of the new triangular element and all checks are made in this local system.


### 2.1.2. Front contraction (topology-based element generation)

The objective of this phase of the algorithm is to force the generation of valid triangles, even if a measure of the quality of the shape of the element does not fall within the allowable range used in the previous phase. The topology-based element-generation phase starts when a boundary edge fails twice in trying to generate an optimal element. The list of rejected edges of the previous phase is transformed into a list of active edges and, similarly to the geometry-based phase, a list of rejected edges is created for edges that eventually fail in generating valid triangles.
In the topology-based element-generation phase, any node close to the current base edge is selected and stored in a priority queue of candidate nodes. The node that has the maximum included angle with respect to the base edge is chosen for the generation of the new triangle. If the edges of this triangle do not intercept any other edge of the current advancing front, the element is created and the boundary is contracted accordingly. The topologybased phase ends when the lists of active and rejected edges are empty. This phase always generates a valid, even though non-optimal, mesh.

### 2.2 Local mesh improvement

A smoothing technique is used to improve mesh quality by relocating nodes within a patch. A general formulation for this technique is given by equation (1), which is a generic form of a weighted Laplacian function. The smoothing procedure is repeated twice for all internal nodes. In general, the smoothing of surface mesh equation will move nodes to a position off of the surface. The $\mathbf{3}^{\text {rd }}$ Method is employed after each smoothing procedure as a "pull back" operation, moving the target node back to the geometric supporting surface. In theory, Laplace smoothing and the "pull back" procedure can cause mesh "folding". A check is made and a node is not moved if doing so will cause an invalid mesh. In practice, there has been no need to enforce this restriction for any of the test cases.

## 3. Application: re-meshing

To demonstrate the performance and the efficiency of the proposed surface meshing algorithm, this section describes an implementation of the three generic methods for re-meshing existing surface triangulations.

### 3.1. First generic method implementation

The first generic method of the proposed surface mesh generation obtains the desired characteristic size of a triangular element given the position of a point on the surface. It is important that the algorithm presents a smooth transition between regions with elements of highly varying sizes. Therefore, an auxiliary background data structure (octree) is used to store the distribution of characteristic triangle sizes in space.
In the current application the size of the leaf cells in a region of a domain are used as the size of the desired characteristic element size in this region. The creation of the background octree from an input surface follows five steps: (a) Octree initialization based on given boundary edges. (b) Refinement to force maximum cell size. (c) Refinement to provide minimum size disparity for adjacent cells. (d) Refinement to account for surface curvatures. (e) Refinement to provide minimum size disparity for adjacent cells for the updated octree.

### 3.2. Second generic method implementation

Two different approaches to this task are described in the literature. Lohner [6] tries to find a host triangle, where the ideal node lies, using a neighbor-to-neighbor search. If this fails, octrees are employed. Finally, if this approach fails again, a brute force search over all the surface elements is performed. Carlos et al. [5] use a procedure that guarantees that the new triangle sides have exactly the desired length. This procedure considers the equation of a sphere, and finds the intersections between the sphere and the mesh. The approach used in this work is similar to this one, employing the circle equation in a 2D coordinate system.

### 3.3. Third generic method implementation

The third generic method returns the point on the geometric supporting surface that is closest to a given point near to, but not necessarily on, the surface. During the nodal smoothing phase, points may be moved off the surface. This method is used to move the points back to the surface. As input, the method receives the current node location and its current normal vector. The implementation of this function is quite simple: First, a local search based on the auxiliary R-Tree structure creates a list of all neighborhood triangles of the geometric supporting surface around the given node. Second, the node is projected on each neighbor triangle plane, using the given projection direction. Finally, the selected new node location is the one that lies inside one of the neighbor triangles.

### 3.4 Example: performance mesh and quality

This section presents additional examples of finite-element meshes generated on 3D surfaces using the proposed algorithm. The main objectives of this section are to estimate the expected performance of the surface mesh generation algorithm and assess the quality of generated triangular elements.

Four examples of triangle surface meshes to be re-meshed with the present algorithm. The first one is an open cylinder. The second is a geological salt-dome surface. The third is a geological surface with four internal loops. The final example is a surface with many irregularities in its curvatures.

Figure 1(a) shows a plot of the elapsed processing time as a function of the number of elements generated. Using a fitting equation, one may infer that the algorithm's performance is close to $O(N \log N)$. The quality of generated meshes is presented in the form of a histogram such as the one shown in Figure 1(b). In this histogram, the horizontal axis corresponds to the $\gamma / \gamma^{*}$ quality measure in intervals represented by triangular shapes that are shown below the histogram. The vertical axis corresponds to the percentage of elements in each interval of the quality measure. These results demonstrate that the proposed algorithm generates meshes with good quality for the great majority of elements. Note that the quality of meshes in all examples is below 1.5 , an indication of very well shaped elements.

## 4. Conclusion

This paper has described an algorithm for regenerating triangle finite-element meshes on surfaces of arbitrary shape and with varying curvature. The input data for the present algorithm is a generic supporting surface and a polygonal description of the boundary on the surface patch to be meshed. The supporting surface is represented abstractly by three generic methods: The first, given a point location, returns the desired characteristic size of a triangular element at this position; The second, given the current edge in the boundary contraction algorithm, locates the ideal apex point that forms a triangle with this edge; and The third, given a point in space and a projection direction, returns the closest point on the geometrical supporting surface.

These three generic methods are used in the proposed two-pass advancing-front procedure to generate elements on the supporting surface. In the first pass, elements are generated based on geometrical criteria, which produces well-shaped elements. In the second pass, triangular elements are generated based solely on topology criteria.
The three generic methods were implemented to regenerate triangle mesh surfaces. This approach is very useful when a mesh needs to be refined, coarsened or improved. Some examples have demonstrated the quality of the generated meshes and the importance of considering surface curvatures in local mesh refinement.


Figure 1 - (a) Generation times for re-meshing the example surfaces and (b) Histogram of element quality for re-meshing the example surfaces.

## References

[1] Cavalcante Neto J B, Wawrzynek P A, Carvalho M T M, Martha L F, and Ingraffea A R (2001) An Algorithm for Three-Dimensional Mesh Generation for Arbitrary Regions with Cracks. Engineering with Computers 17(1):75-91.
[2] Miranda A C O and Martha L F (2002) Mesh Generation on High Curvature Surfaces Based on Background Quadtree Structure. Proceedings of $11^{\text {th }}$ International Meshing Roundtable 1:333-341.
[3] Miranda A C O, Cavalcante Neto J B, and Martha L F (1999) An Algorithm for Two-dimensional Mesh Generation for Arbitrary Regions with Cracks, SIBGRAPI'99. XII Brazilian Symposium on Computer Graphics, Image Processing and Vision, IEEE Computer Society Order Number PRO0481, ISBN 0-7695-0481-7, Eds.: J. Stolfi \& C. Tozzi, pp.29-38.
[4] Miranda A C O, Meggiolaro M A, Castro J T P, Martha L F, and Bittencourt T N (2003) Fatigue Life and Crack Path Predictions in Generic 2D Structural Components. Engineering Fracture Mechanics 70(10):1259-1279.
[5] Carlos J, Scheidegger E, Fleishman S, and Silva C T (1996) Direct (Re)Meshing for Efficient Surface Processing. Comp. Graph. Forum 25(3):527-536.
[6] Lohner R (1996) Regridding Surface Triangulations. Journal of Computational Physics 126(1):1-10.
[7] Shostko A A, Lohner R, Sandberg W C (1999) Surface Triangulation over Intersecting Geometries. Int. J. Numer. Meth. Engrg. 44:1359-1376.
[8] Nakahashi K, Sharov D (1995) Direct Surface Triangulation Using the Advancing Front Method. AIAA, pp. 442-451.
[9] Lau T S and Lo S H (1996) Finite Element Mesh Generation over Analytical Curved Surfaces. Computers and Structures 59(2):301-309.
[10] Lo S H and Lau T S (1998) Mesh Generation over Curved Surfaces with Explicit Control on Discretization Error. Engineering Computations: Int. J. for Computer-Aided Engineering 15(3):357-373.
[11] Rassineux A (1998) Generation and Optimization of Tetrahedral Meshes by Advancing Front Technique. International Journal for Numerical Methods in Engineering 41:651-674.

