INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 40, 3313–3324 (1997)

NEW LONGEST-EDGE ALGORITHMS FOR THE REFINEMENT AND/OR IMPROVEMENT OF UNSTRUCTURED TRIANGULATIONS

MARÍA-CECILIA RIVARA*

Department of Computer Science, University of Chile, Casilla 2777, Santiago, Chile

ABSTRACT

In this paper I introduce a new mathematical tool for dealing with the refinement and/or the improvement of unstructured triangulations: the Longest-Edge Propagation Path associated with each triangle to be either refined and/or improved in the mesh. This is defined as the (finite) ordered list of successive neighbour triangles having longest-edge greater than the longest edge of the preceding triangle in the path. This ideal is used to introduce two kinds of algorithms (which make use of a Backward Longest-Edge point insertion strategy): (1) a pure Backward Longest-Edge Refinement Algorithm that produces the same triangulations as previous longest-edge algorithms in a more efficient, direct and easy-to-implement way; (2) a new Backward Longest-Edge Improvement Algorithm for Delaunay triangulations, suitable to deal (in a reliable, robust and effective way) with the three important related aspects of the (triangular) mesh generation problem: mesh refinement, mesh improvement, and automatic generation of good-quality surface and volume triangulation of general geometries including small details. The algorithms and practical issues related with their implementation (both for the polygon and surface quality triangulation problems) are discussed in this paper. In particular, an effective boundary treatment technique is also discussed. The triangulations obtained with the LEPP–Delaunay algorithm have smallest angles greater than 30° and are, in practice, of optimal size. Furthermore, the LEPP–Delaunay algorithms naturally generalize to three-dimensions. © 1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Engng., **40**, 3313–3324 (1997)

No. of Figures: 8. No. of Tables: 0. No. of References: 13.

KEY WORDS: longest-edge refinement; triangulation improvement; surface and volume triangulation

1. INTRODUCTION

During the last 15 years the triangular mesh generation problem has evolved into an important and interdisciplinary research field. In this general context, the following three related problems (in two and three-dimensions) should be consistently considered: (1) triangular mesh refinement; (2) triangular mesh improvement; (3) automatic generation of good-quality surface and volume triangulation.[1–5] In effect, many applications on numerical simulation, solid modelling and computer graphics require that complicated geometric objects be decomposed in simpler pieces for further processing. Furthermore, in the adaptive finite element setting, the availability of mesh refinement algorithms capable of modifying the mesh in the course of computations is a critical aspect of the entire numerical solution process. A difficult related problem (especially difficult in

* Correspondence to: M.-C. Rivara, Department of Computer Science, University of Chile, Casilla 2777, Santiago, Chile

CCC 0029–5981/97/183313–12$17.50

Received 12 July 1996
Revised 20 January 1997

three-dimensions) is the automatic construction of good quality, adapted to the geometry, triangulations. In this case, a set of non-vertex (Steiner) points should be added to produce a quality triangulation to be later refined.

Two approaches, currently used to deal with some aspects of these questions, are combined in this paper: the approach based on the known Delaunay[2,3] algorithm, which assures, at least in two-dimensions, the construction of the most equilateral triangulation at the optimal time cost of $O(N \log N)$ for a given set of N vertices, with careful management of its associated non-robustness issues; and the approach based on longest-edge refinement/derefinement algorithms for general triangulations which guarantees the robust construction of good-quality irregular and nested triangulations with linear time complexity, provided that an initial good-quality triangulation is used.[1,8–10]

Both approaches have been recently used to deal with some aspects of the refinement of constrained Delaunay triangulations. The Delaunay algorithm has been the basis to produce (automatic) good-quality 2-D triangulations of complex polygons,[3–5] by adding non-vertex points in the circumcenter of the worst triangles of the current triangulation, which generalizes Paul Chew's idea. However, as Ruppert points out in his paper, these ideas do not extend to three-dimensions: "It seems that significant new ideas are necessary in order to get bounded aspect-ratio tetrahedra using a Delaunay triangulation based approach". On the other hand, longest-edge bisection techniques have been successfully used for the automatic refinement of quality Delaunay triangulations with linear time complexity.[6,7]

In this paper, I generalize the previous results on the longest-edge mesh refinement algorithms, as well as recent results on the combination of longest-edge bisection techniques and the Delaunay algorithm, in order to introduce a new backward longest-edge point placement strategy, based on the Longest-Edge Propagation Path (LEPP) concept, which allows to formulate two kinds of algorithms: (1) a pure Backward Longest-Edge Refinement Algorithm that produces the same triangulations as previous longest-edge algorithms in a more efficient, direct and easy-to-implement way; (2) a new Backward Longest-Edge Improvement Algorithm for Delaunay triangulations, suitable to deal (in a reliable, robust and effective way) with the three important related aspects of the (triangular) mesh generation problem: mesh refinement, mesh improvement, and automatic generation of good-quality surface and volume triangulation of general geometries including small details. The algorithms and practical issues related to their implementation, both for the quality 2-D and surface triangulation problems, are discussed in this paper.

2. LONGEST-EDGE PROPAGATION PATH OF A TRIANGLE

In this section we shall consider general conforming unstructured triangulations (where the intersection of adjacent triangles is either a common vertex or a common side). The following concepts have been implicitly used before in previous longest-edge refinement/derefinement algorithms:

Definition 1. For any triangle t_0 of any conforming triangulation T, the LEPP of t_0 will be the ordered list of all triangles $t_0, t_1, t_2, \ldots, t_{n-1}, t_n$, such that t_i is the neighbour triangle of t_{i-1} by the longest side of t_{i-1}, for $i = 1, 2, \ldots, n$. In addition, we shall denote it as the LEPP(t_0).

Proposition 1. For any triangle t_0 of any conforming triangulation of any bounded 2-D geometry Ω, the following properties hold: (a) *for any t, the* LEPP(t) *is always finite*; (b) *The triangles t_0, $t_1, \ldots, t_{n-1}$ have strictly increasing longest side* (*if $n > 1$*); (c) *For the triangle t_n of the* LEPP *of any*

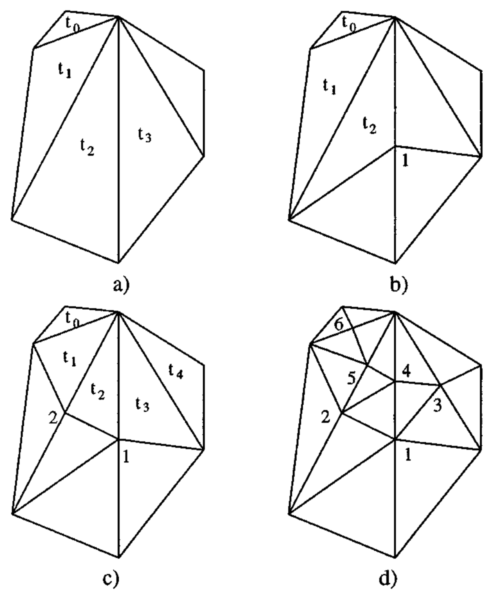

Figure 1. Backward longest-edge bisection of triangle t_0: (a) initial triangulation; (b) first step of the process; (c) second step in the process; (d) final triangulation

triangle t_0, *it holds that either* (i) t_n *has its longest edge along the boundary, and this is greater than the longest edge of* t_{n-1}, *or* (ii) t_n *and* t_{n-1} *share the same common longest edge.*

Definition 2. Two adjacent triangles (t, t^*) will be called a pair of terminal triangles if they share their respective (common) longest edge. In addition, t will be a terminal boundary triangle if its longest-edge lies along a boundary edge.

It should be pointed out here that the LEPP of any triangle t corresponds to an associated polygon, which in certain sense measures the local quality of the current point distribution induced by t. To illustrate these ideas, see Figure 1, where the longest-edge propagation path of t_0 corresponds to the ordered list of triangles (t_0, t_1, t_2, t_3). Moreover, the pair (t_2, t_3) is a pair to terminal triangles.

3. A BACKWARD LONGEST-EDGE REFINEMENT ALGORITHM

The backward longest-edge refinement techniques extend both the pure longest-edge refinement algorithms for general non-Delaunay triangulations,[1,8–10] and the longest-edge refinement algorithm for Delaunay triangulations proposed in References 6 and 7. Note that these previous longest-edge algorithms guarantee the construction of refined triangulations that basically maintain the quality of the input coarse triangulation. Furthermore, for dealing with the triangulation refinement problem, they are of optimal (linear) time cost.[9,10]

The original refinement algorithms, based on the longest-edge bisection of triangles, were explicitly developed to solve the triangulation refinement problem in the adaptive finite element setting. The ideal is to exploit the knowledge one has of the reference triangulation for working only locally with the refinement area (and some neighbouring triangles). The new points introduced in the mesh are mid-points of the longest edge of (at least) one triangle of the reference mesh. In order to maintain a conforming triangulation, the local refinement of a given triangle involves refinement of the triangle itself and refinement of some of its neighbours (by the longest edge).

By using the LEPP(t) concept, a backward longest-edge refinement algorithm can be formulated,[11] where the pure longest-edge refinement of a target triangle t_0 (see Figure 1) essentially means the repetitive longest-edge partition of the pair of terminal triangles associated with the current LEPP(t_0), until the triangle t_0 itself is partitioned. Note that his backward algorithm indeed produces the final triangulation without intermediate non-conforming points in the mesh.

Backward-Longest-Edge-Bisection (t, T)
  While t remains without being bisected do
    Find the LEPP(t)
    If t*, the last triangle of the LEPP(t), is a
      terminal boundary triangle, bisect t*
    Else bisect the (last) pair of terminal triangles
      of the LEPP(t)

For an illustration of the algorithm, see Figure 1, which illustrates the refinement of the triangle t_0 over the initial triangulation of Figure 1(a) with associated LEPP$(t_0) = \{t_0, t_1, t_2, t_3,\}$. The triangulations (b) and (c) illustrate the first 2 steps of the backward-longest-edge-bisection procedure and their respective current LEPP(t_0), while that triangulation (d) is the final mesh obtained. Note that the new vertices have been enumerated in the order they were created.

Remarks

(1) The backward-longest-edge-bisection procedure is a non-recursive algorithm essentially based on refining pairs of terminal triangles (according to Definition 2). The concept of the LEPP of the triangle t has been repeatedly used (over the current triangulation) in order to find the last 2 (terminal) triangles of the path, until the initial triangle t is bisected.
(2) To solve the triangulation refinement problem, the algorithm guarantees the construction of good-quality irregular and nested triangulations, with linear time complexity, provided that an initial good-quality triangulation is used. To this end, a suitable data structure that explicitly manage the neighbour-triangle relation should be used. In addition, since at each iteration within the while loop, the LEPP(t) may or may not be shortened, and may include new triangles not previously included in the LEPP(t) (see Figure 1), the current LEPP(t) should be updated, rather than computed from scratch in order to get the linear running time.
(3) The new backward refinement algorithm produces the same triangulation as the previous recursive algorithm,[1,8] in a simpler, cleaner, easy-to-implement and more direct way.
(4) It should be pointed out also that, since the backward-longest-edge-bisection procedure is a non-recursive algorithm, based on refining pairs of terminal triangles, parallel and efficient refinement algorithms (for a variety of triangles t_0) can be imagined. Clearly, in this case, the identification of the set of pairs of terminal triangles can be made in parallel (without modifying the data structure). Furthermore, by using a suitable edge-based data structure, the refinement step (partition of the pairs of terminal triangles) can be easily parallelized.
(5) These ideas easily extend to three-dimensions (see Section 9).

4. A BASIC LONGEST-EDGE IMPROVEMENT TECHNIQUE FOR DELAUNAY TRIANGULATIONS

Analogous to the algorithms of the previous section, this Delaunay improvement technique[11] uses the LEPP of the target triangles (to be either refined and/or improved in the mesh) in order to decide which is the best point to be inserted to produce a good-quality distribution of points.

Basic Backward-LE-Delaunay-Improvement (t, T)
While t remains without being modified do
  Find the Longest Edge Propagation Path of t
  Perform a Delaunay insertion of the point p
    (midpoint of the longest edge of the last triangle
      in the LEPP(t))

Note that we have used the word improvement instead of bisection or refinement. This is to make explicit the fact that one step of the procedure does not necessarily produce a smaller triangle. More important, however, is the fact that the procedure improves the triangle t in the sense of Theorem 2 of Section 5.

For an illustration of the algorithm see Figure 2, where the triangulation (a) is this initial Delaunay triangulation with LEPP(t_0) = $\{t_0, t_1, t_2, t_3\}$, and the triangulations (b)–(d) illustrate the complete sequence of point insertions needed to improve t_0. Note that in this example, the improvement (modification) of t_0 implies the automatic Delaunay insertion of three additional Steiner points. Each one of these points is the midpoint of the longest-edge of the last triangle of the current LEPP(t_0). It should be pointed out here that each Delaunay point insertion locally

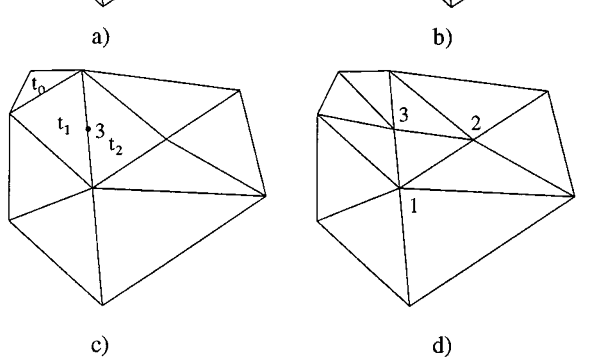

Figure 2. Backward longest-edge Delaunay improvement of triangle t_0

improves the triangulation in the current LEPP(t_0), and in this sense this algorithm improves the triangulations obtained with the pure backward longest-edge refinement procedure.

5. MATHEMATICAL PROPERTIES OF THE BACKWARD TECHNIQUES

In what follows, I shall assume that t_0 is the worst triangle in the LEPP of t_0. Note that this assumption is general enough, since if the LEPP(t_0) includes a worst triangle t_k, in this case we can first consider the smaller chain LEPP(t_k). Then for the pure longest-edge backward algorithm, the following property holds:

Theorem 1. (a) *The repetitive use of the backward longest-edge bisection procedure, in order to refine t_0 and its descendants (triangles nested in t_0), tends to produce local quasi-equilateral triangulations.* (b) *The smallest angle α_t of any triangle t obtained throughout this process, satisfies that $\alpha_t \geqslant \alpha_0/2$, where α_0 is the smallest angle of t_0.* (c) *For any conforming triangulation T, the global iterative application of the backward algorithm covers, in a monotonically increasing form, the area of t_0 with quasi-equilateral triangles.*

Proof. The algorithm produces the same triangulations as the recursive pure longest-edge refinement algorithm.[1,8–10]

For the backward longest-edge Delaunay technique, I shall assume in addition that the input triangulation (the constrained Delaunay triangulation[2] of the input geometry data), has a boundary point distribution that represents well the local feature size[5] of the geometry boundary. This assumption allows to avoid boundary troubles which will be surmounted in practice by using the boundary treatment technique of Section 6.

Theorem 2. For any Delaunay triangulation T, the repetitive use of the Backward-LE-Delaunay-Improvement technique over the worst triangles of the mesh (with smallest angle $\alpha < 30^\circ$) produces a quality triangulation of smallest angles greater than 30°.

Proof. The proof is based on the properties of both the longest-edge refinement algorithms and the Delaunay triangulation. In effect, note that the pure backward-longest-edge-bisection procedure essentially adds to the current set of vertices, the midpoint of the longest-edge of the last greatest triangle of the LEPP(t), which in turn is inserted by longest-edge partition of the associated pair of terminal triangles of the current LEPP(t). This work produces nested triangles, and an adequate point distribution which guarantees that the percentage of good-quality triangles (and the area covered by these triangles) increases throughout the process (part(c) of Theorem 1). However, some bad triangles still remain in the mesh due to the fact that the longest-edge algorithms produce stable molecules[10] around the vertices (after a small number of partitions, the angles that share a vertex are fixed and not refined anymore). This is mainly due to the fact that nested triangulations are obtained.

When the Backward-LE-Delaunay-Improvement procedure is used in exchange, the midpoint of the longest-edge of the last greatest triangle of the LEPP(t) is also added, which improves the point distribution in the sense of the pure longest-edge algorithms. Furthermore, since this point is Delaunay inserted in the current triangulation, this local procedure improves the current triangulation in the following two senses: (1) the most equilateral mesh for the set of vertices is obtained; (2) the worst angles of the (non-fixed) molecules are eliminated (by edge swapping). If the triangle t is not destroyed throughout the process, the new LEPP(t) is found over a locally improved triangulation; and as a consequence, the addition of the mid-point of the longest-edge of the last greatest triangle of the new LEPP(t), improves even more the current point distribution; while that the Delaunay insertion of this point again improves the triangulation in the two senses stated before; and so on. This process guarantees that the Backward-LE-Delaunay algorithm produces good-quality triangulations with smallest angles greater than 30°. Note that, smallest angles of more than 30° cannot be assured, since the longest-edge Delaunay partition of equilateral triangles can produce angles of 30°.

Remarks.

(1) It should be pointed out here that, even when Theorem 2 guarantees the construction of quality triangulations, it says nothing about the size of these triangulations. More mathematical results in this sense are certainly needed.
(2) In practice, the 2-D triangulations obtained are size-optimal (they are of analogous quality as those obtained with the circumcenter point insertion strategy).

6. A LONGEST-EDGE BOUNDARY TREATMENT TECHNIQUE

It is worth pointing out here that the basic LEPP-point-insertion algorithm of the preceding section performs well in practice whenever the input geometry has an adequate initial distribution of boundary points. This is intuitively due to the fact that, in this case, for each t, the pair of terminal triangles associated with the LEPP(t) are interior triangles (the edges of the last triangle of the path are not along the geometry boundary) and consequently interior points (far enough from the boundary) are inserted.

Enough care should be taken in exchange, when the LEPP(t) finishes in the geometry boundary. To illustrate this idea consider the simple example of Figure 3(a). In this case the naive use of the LEPP point insertion algorithm would produce undesirable interior points (as in Figure 3(b)). To avoid this effect, the following boundary treatment technique is introduced in this paper (where t is the last greatest triangle of the current LEPP):

Boundary-Treatment-Procedure (T, t, P)
  If t has a boundary edge l, and l is not the smallest edge of t,
    then select P, the midpoint of l
  Else select P, the midpoint of the longest-edge of t

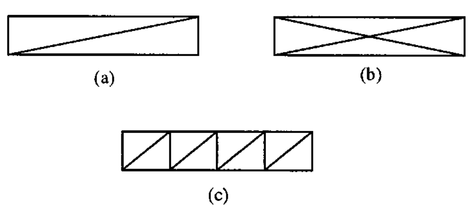

Figure 3. (a) Initial triangulation of the boundary vertices; (b) undesirable triangulation obtained without using the boundary treatment procedure; (c) final triangulation obtained by using the boundary treatment procedure

Remark. The boundary-treatment-procedure introduces in practice a boundary point distribution which adapts naturally and automatically to the local feature size[5] of the geometry boundary (as in Figure 3(c)).

7. A BASIC (QUALITY) POLYGON TRIANGULATION ALGORITHM

By combining the techniques of Sections 4 and 5, a simple and effective 2-D quality triangulation algorithm can be formulated as follows:

Quality-Polygon-Triangulation ($\wp$, ε)
Input: A general polygon $\wp$ (defined by a set of vertices and edges); and a tolerance parameter $\varepsilon (\varepsilon < 30^\circ)$
Construct T, a constrained (boundary) Delaunay triangulation of $\wp$.
Find S, the set of the worst triangles t of T (of smallest-angle $\alpha_t < \varepsilon$)
For each t in S do
  Backward-LE-Delaunay-Improvement (T, t)
  Actualize the set S (by adding the new small-angled triangles and eliminating those destroyed throughout the process)
End for

Backward-LE-Delaunay-Improvement (T, t)
While t remains without being modified do
  Find the LEPP(t), and t^* the last triangle in the LEPP(t)
  If t^* has a boundary edge l, and l is not the smallest-edge of t, then select P, the midpoint of l
  Else select P the midpoint of the longest-edge of t^*
  Perform the Delaunay insertion of P

Remarks.

(1) Note that ε is a threshold parameter less than or equal to 30° that can be easily adjusted.
(2) In practice we have worked with a constrained Delaunay triangulation of the interior of the 2-D geometry.[2]

It is worth pointing out here that different variants of the quality-triangulation-algorithm of this section can be easily imagined. In particular, we shall call basic non-ordered algorithm to the quality triangulation algorithm of this section to emphasize the fact that the triangles of the set S are maintained and processed in any order.

8. A BASIC ORDERED ALGORITHM

This is a variant of the algorithm of the previous section, where an order is previously introduced and maintained over the set of the worst triangles S: the triangles are ordered in increasing order with respect to their smallest angle and processed in this order. The order is also preserved when the S set is actualized in the For cycle.

Remarks.

(1) No relevant differences have been obtained in practice in the size and quality of the output triangulations for both basic algorithms (ordered versus non-ordered versions) and the same value of the ε parameter.
(2) It should be pointed out, however, that the ordered algorithm can be easily parameterized to produce the best mesh for a maximum number of allowable triangles or vertices. This is a desirable and important feature in complex applications.
(3) In both cases (ordered an non-ordered algorithm), a suitable data structure that explicitly manages the neighbour-triangle relation should be used. In addition, since at each iteration within the while loop, the LEPP(t) may or may not be shortened, and may include new triangles not previously included in the LEPP(t), the current LEPP(t) should be updated, rather than computed from scratch whenever the triangle t still exists in the current mesh.
(4) Finally, it should be pointed out that both algorithms have a kind of self-corrective property, in the sense that for the first triangles, the Delaunay insertion of the midpoint of the longest-edge of the last triangle of the path destroys (and improves) a big subset of the worst triangles S.

9. 3-D ALGORITHMS

In this section, I shall only introduce the 3-D concepts that allow to generalize the backward algorithms of Sections 4 and 5. Since in 3-dimensions the refinement propagates in several directions, the following definitions are in order.[12]

Definition 3. The 3D-LEPP of any tetrahedron t, is the set of all the neighbour tetrahedra (by the longest edge) having respective longest-edge greater than or equal to the longest edge of the preceding tetrahedra in the path.

Definition 4. A terminal-tetrahedra-set is the set of all the tetrahedra of the mesh that share their common longest edge.

Proposition 2. The 3*D*-LEPP *of any tetrahedron has, in the general case, a set of N terminal-tetrahedra-sets, with $N > 1$.*

By using these concepts, the backward algorithm of Section 4 generalizes to three dimensions as follows:

3D-Backward-Longest-Edge-Delaunay-Improvement (T, t)
While t remains without being modified do
  Find the set of the N terminal-tetrahedra-sets associated with
    the 3D-Longest-Edge Propagation Path of t
  Perform the Delaunay insertion of the midpoint of the longest-edge
    of each terminal-tetrahedra-set, for $n = 1, 2, \ldots, N$

It is well known that the ability of the Delaunay technique to produce quality triangulations in three dimensions is a strong function of the point placement algorithm. Preliminary research in the 3-D context shows that the basic LEPP algorithm behaves, in practice, analogously as the 2-D case. In particular, the LEPP point insertion strategy allows the fast elimination of undesirable slivers.

Furthermore, as in the 2-D case, the LEPP point insertion strategy can be used both to improve the efficiency and robustness of the 3-D Delaunay routine in the following 2 senses: (1) the search of the tetrahedron that contains the point is avoided, and (2) the initial (locally) non-Delaunay triangulation can be easily obtained, by longest-edge partition of the involved tetrahedra.

10. PRACTICAL BEHAVIOUR OF THE LEPP–DELAUNAY ALGORITHMS

The empirical experimentation with the 2-D algorithms[13] described in Section 6, has shown that they produce triangulations of analogous size and quality as the circumcentre algorithm: optimal size meshes are obtained with smallest angle greater than 30°. However, since the LEPP(t) is always interior to the polygon geometry, the LEPP-algorithms show better behaviour than the Ruppert algorithm around the geometry boundaries. Furthermore, the LEPP–Delaunay algorithms have the following important practical advantages: since the points inserted in the LEPP(t) are always midpoints of the longest edge of one or two known triangles of the current mesh, this knowledge can be used to improve the efficiency and robustness of the Delaunay routine. More specifically: (1) the search of the triangle that contains the points to be inserted is avoided, and (2) the initial (locally) non-Delaunay triangulation can be easily obtained by longest-edge partition of the involved triangles.

The triangulations of Figures 4–6 illustrate the practical behaviour of the LEPP-algorithms. Note that for these three examples the input was the polygon with the minimum number of vertices required to describe the geometry. Thus, the triangulation of Figure 5 was obtained automatically from an 18-vertices input polygon; the algorithm inserted the remaining vertices on the boundary. For the example of Figure 4 most of the vertices inserted are interior vertices. Figure 6 illustrates the case where an interior edge has to be respected.

The LEPP–Delaunay algorithms have been successfully used to obtain quality surface triangulations of different polyhedra. Figures 7 and 8 illustrate the polyhedra surface case. To this end, the non-ordered algorithm of Section 5 was used to triangulate the faces, combined with an adequate (iterative) boundary communication procedure between pairs of adjacent faces (both face triangulations must share the same vertices in the final surface mesh).

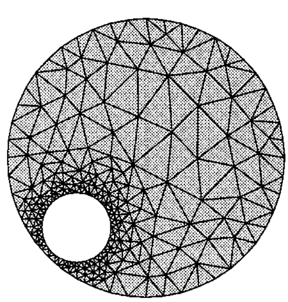

Figure 4. Automatic triangulation of the geometry (interior vertices introduced by the algorithm)

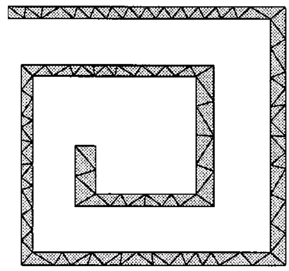

Figure 5. Automatic triangulation constructed for the input triangulation having only (18) polygon-vertices

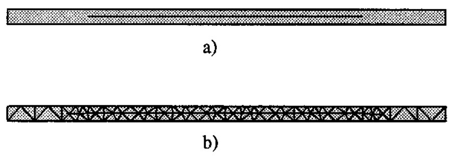

Figure 6. Triangulation of a rectangle including an interior boundary edge

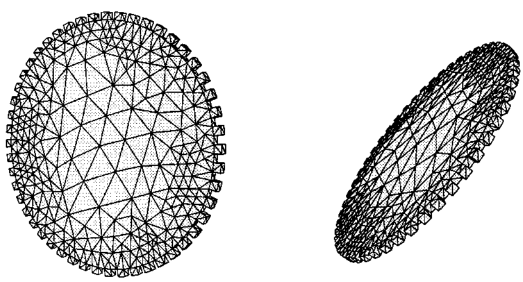

Figure 7. Automatic surface triangulation of the gear wheel (initial triangulation having only vertices over the polyhedron edges)

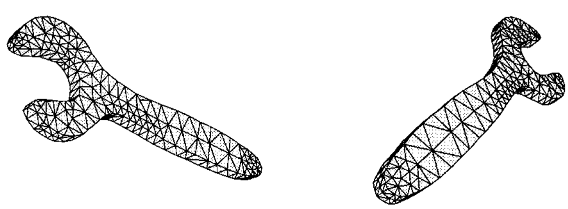

Figure 8. Automatic surface triangulation. The input triangulation had only polyhedron edge vertices

11. CONCLUSIONS

In this paper I have introduced the LEPP mathematical construct (the propagation path associated with pure longest-edge refinement algorithms) as well as a simple and effective backward LEPP-based point insertion strategy, which in turn has been used both for the design of simpler longest-edge refinement algorithms for quality triangulations; and for the improvement of bad-quality Delaunay triangulations. The LEPP–Delaunay improvement technique combined with an appropriate boundary treatment technique (introduced in this paper) produces simple and powerful LEPP–Delaunay algorithms for the quality polygon, surface and volume triangulation problems.

The 2-D LEPP–Delaunay algorithms guarantee the construction of good-quality triangulations with smallest angles greater than 30° due to the following facts: (1) the longest-edge point insertion strategy improves the quality of the point distribution (the percentage of quasi-equilateral triangles, and the area covered by these triangles, increases throughout the process); (2) For the (local) current point distribution, the Delaunay algorithm improves the mesh (the worst angles are eliminated by edge swapping). Furthermore, these algorithms naturally generalize to 3-dimensions, showing in practice an analogous behaviour as the 2-D case.

REFERENCES

1. M. C. Rivara, 'Algorithms for refining triangular grids suitable for adaptive and multigrid techniques', *Int. J. Numer. Meth. Engng*., **20,** 745–756 (1984).
2. L. P. Chew, 'Constrained Delaunay triangulation', *Algorithmica*, **4,** 97–108 (1989).
3. L. Guibas and J. Stolfi, 'Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams', *ACM Trans. Graphics*, **4,** 74–123 (1985).
4. L. P. Chew, 'Guaranteed-quality triangular meshes', *Technical Report TR-89-983*, Cornel, University, 1989.
5. J. Ruppert, 'A Delaunay refinement algorithm for quality 2-dimensional mesh generation', *J. Algorithms*, **18,** 548–585 (1995).
6. M. C. Rivara and P. Inostroza, 'A discussion on mixed (longest side midpoint insertion) Delaunay Techniques for the triangulation refinement problem', *Proc 4th Int. Meshing Roundtable*, Sandia National Labs, New Mexico, 1995, pp. 335–346.
7. M. C. Rivara and P. Inostroza, 'Using longest-side bisection techniques for the automatic refinement of Delaunay triangulations', *Int J. Numer. Meth. Engng*., **40,** 581–597 (1997).
8. M. C. Rivara, 'Selective refinement/derefinement algorithms for sequences of nested triangulations', *Int. J. Numer. Meth. Engng*., **28,** 2889–2906 (1989).
9. M. C. Rivara and G. Iribarren, 'The 4-triangles longest-side partition of triangles and linear refinement algorithms', *Math. Comput*., **65,** 1485–1502 (1996).
10. M. C. Rivara and M. Venere, 'Cost analysis of the longest-side (triangle biection) refinement algorithms for triangulations', *Eng. Comput*., **12,** 224–234 (1996).
11. M. C. Rivara, 'New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations', *Proc. 5th Int. Meshing Roundtable 96*, Pittsburgh, 1996, pp. 77–86.
12. M. C. Rivara and C. Levin, 'A 3D refinement algorithm suitable for adaptive and multigrid techniques', *Commun. Appl. Numer. Meth*., **8,** 281–290 (1992).
13. M. Palma, 'Triangulación automática de polígonos y superficies de poliedros', Departamento de Ciencias de la Computación, U. de Chile, 1996.

