Lepp Terminal Centroid Method for Quality Triangulation: A Study on a New Algorithm

Maria-Cecilia Rivara and Carlo Calderon

Department of Computer Science, University of Chile
{mcrivara,carcalde}@dcc.uchile.cl

Abstract. We introduce a new Lepp-Delaunay algorithm for quality triangulation. For every bad triangle t with smallest angle less than a threshold angle θ, a Lepp-search is used to find an associated convex terminal quadrilateral formed by the union of two terminal triangles which share a local longest edge (terminal edge) in the mesh. The centroid of this terminal quad is computed and Delaunay inserted in the mesh. The algorithm improves the behavior of a previous Lepp-Delaunay terminal edge midpoint algorithm. The centroid method computes significantly smaller triangulation than the terminal edge midpoint variant, produces globally better triangulations, and terminates for higher threshold angle θ (up to $36°$). We present geometrical results which explain the better performance of the centroid method. Also the centroid method behaves better than the off-center algorithm for θ bigger than $25°$.

Keywords: mesh generation, triangulations, Lepp Delaunay algorithms, Lepp centroid.

1 Introduction

In the last decade, methods that produce a sequence of improved constrained Delaunay triangulations (CDT) have been developed to deal with the quality triangulation of a planar straight line graph D. The combination of edge refinement and Delaunay insertion has been described by George and Borouchaki [3,2] and Rivara and collaborators, [8,9,10,11]. Mesh improvement properties for iterative Delaunay refinement based on inserting the circumcentre of triangles to be refined have been established by Chew, [1], Ruppert [12], and Shewchuk [15]. Applications of this form of refinement have been described by Weatherill et al [17] and Baker [13]. Baker also published a comparison of edge based and circumcenter based refinement [14]. Algorithms based on off-center insertions have been recently presented by Üngor and collaborators [19,24]. Algorithms for uniform triangular meshes are discussed in [16]. For a theoretical review on mesh generation see the monograph of Edelsbrunner [4].

Longest edge refinement algorithms. The longest edge bisection of any triangle t is the bisection of t by the midpoint of its longest edge and the opposite

F. Chen and B. Jüttler (Eds.): GMP 2008, LNCS 4975, pp. 215–230, 2008.

vertex (Figure 3). Longest edge based algorithms [5,6,7,8,9,11] were designed to take advantage of the following mathematical properties on the quality of triangles generated by iterative longest edge bisection of triangles [18,21,22], and require of a resonably good quality input triangulation to start with.

Theorem 1. *For any triangle t_0 of smallest angle α_0.*
(i) The iterative longest edge bisection of t_0 assures that for any longest edge son t of smallest angle α_t, it holds that $\alpha_t \geq \alpha_0/2$, the equality holding only for equilateral triangles. (ii) A finite number of similarly distint triangles is generated. (iii) The area of t_0 tends to be covered by quasi equilateral triangles (for which at most 4 similarly different, good quality triangles are obtained by longest edge bisection of t_0).

Lepp (Delaunay terminal edge) midpoint method. The algorithm was designed to improve the smallest angles in a Delaunay triangulation. This proceeds by iterative selection of a point M which is midpoint of a Delaunay terminal edge (a longest edge for both triangles that share this edge) which is then Delaunay inserted in the mesh. This method uses the longest edge propagating path associated to a bad quality processing triangle to determine a terminal edge in the current mesh. The algorithm was introduced in a rather intuitive basis as a generalization of previous longest edge algorithms in [9,10,11]. This was supported by the improvement properties of both the longest edge bisection of triangles (Theorem 1) and the Delaunay algorithm, and by the result presented in Theorem 2 in next section. Later in [10] we discussed some geometrical properties including a (rare) loop case for angle tolerance greater than 22° and its management. However, while empirical studies show that the method behaves analogously to the circumcircle method in 2-dimensions [9,10,11], formal proofs on algorithm termination and on optimal size property have not been fully established due to the difficulty of the analysis. Recently in [20] we have presented some geometrical improvement properties of an isolated insertion of a terminal edge midpoint M in the mesh. In [23] a first termination proof is presented and several geometric aspects of the algorithm are studied.

Lepp-centroid algorithm. In order to improve the performance of the previous Lepp midpoint algorithm, in this paper we introduce a new Lepp-centroid algorithm for quality triangulation. For any general (planar straight line graph) input data, and a quality threshold angle θ, the algorithm constructs constrained Delaunay triangulations that have all angles at least θ as follows: for every bad triangle t with smallest angle less than θ, a Lepp-search is used to find an associated convex terminal quadrilateral formed by the union of two terminal triangles which share a local longest edge (terminal edge) in the mesh. The centroid of this terminal quad is computed and Delaunay inserted in the mesh. The process is repeated until the triangle t is destroyed in the mesh.

In section 2 we introduce the basic concepts of longest edge propagating path (Lepp), terminal edges and terminal triangles, and a relevant constraint on the largest angle of Delaunay terminal triangles. In section 3 we describe the Lepp-midpoint algorithm, discuss a special loop case that rarely occurs for angles greater than 22°, and a geometric characterization on Delaunay terminal triangles. We use this characterization to state improved angle bounds on the smallest angles, and to prove that the new points are not inserted too close to previous vertices in the mesh. In section 4 we formulate the new Lepp centroid algorithm and state geometrical results which explain the better performance of the Lepp centroid method, and guarantee that the loop case is avoided. In section 5 we present an empirical study that compares the behavior of Lepp-centroid and Lepp-midpoint methods. The centroid method computes significantly smaller triangulation than the terminal edge midpoint variant, produces globally better triangulations, and terminates for higher threshold angle θ (up to 36°). We also show that the Lepp centroid method behaves better than the off-center algorithm for $\theta > 25°$.

2 Concepts and Preliminary Results

An edge E is called a terminal edge in triangulation τ if E is the longest edge of every triangle that shares E, while the triangles that share E are called terminal triangles [9,10,11]. Note that in 2-dimensions either E is shared by two terminal triangles t_1, t_2 if E is an interior edge, or E is shared by a single terminal triangle t_1 if E is a boundary (constrained) edge. See Figure 1 where edge AB is an interior terminal edge shared by two terminal triangles t_2, t_3, while edge CD is a boundary terminal edge with associated terminal triangle t_3.

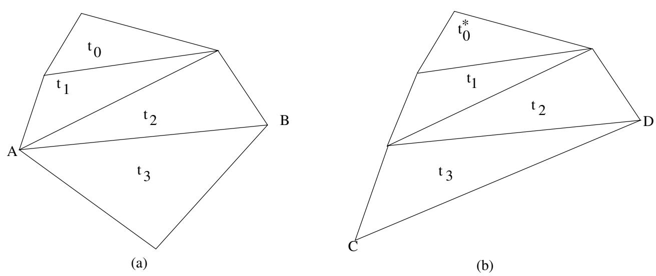

Fig. 1. (a) AB is an interior terminal edge shared by terminal triangles (t_2, t_3) associated to Lepp(t_0) $=\{t_0, t_1, t_2, t_3\}$; (b) CD is a boundary terminal edge with unique terminal triangle t_3 associated to Lepp(t_0^*) $= \{t_0^*, t_1, t_2, t_3\}$

For any triangle t_0 in τ, the longest edge propagating path of t_0, called *Lepp(t_0)*, is the ordered sequence $\{t_j\}_0^{N+1}$, where t_j is the neighbor triangle on a longest edge of t_{j-1}, and longest-edge $(t_j) >$ longest-edge (t_{j-1}), for *j=1,... N*. Edge $E =$ longest-edge(t_{N+1}) = longest-edge(t_N) is a terminal edge in τ and

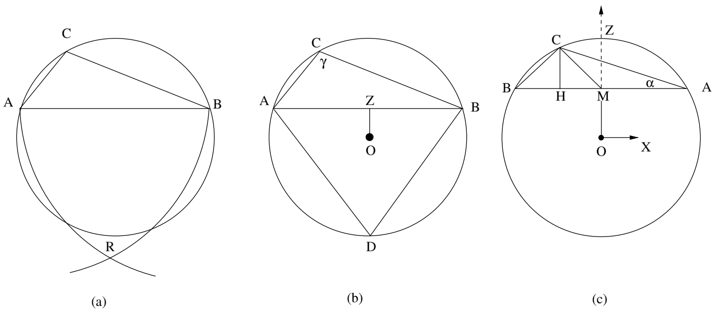

Fig. 2. R is the geometrical place of the fourth vertex D for Delaunay terminal triangles ABC, ABD; (b) R reduces to one point when $\gamma = 2\pi/3$ (triangle ADB equilateral); (c) For a bad terminal triangle BAC, Lepp-midpoint method inserts midpoint M

this condition determines N. Consequently either E is shared by a couple of terminal triangles (t_N, t_{N+1}) if E is an interior edge in τ, or E is shared by a unique terminal triangle t_N with boundary (constrained) longest edge. See Figure 1.

For a Delaunay mesh, an unconstrained terminal edge imposes the following constraint on the largest angles of the associated terminal triangles [8,9,11]:

Theorem 2. *For any pair of Delaunay terminal-triangles t_1, t_2 sharing a non-constrained terminal edge, largest angle(t_i) $\leq$ 2π/3 for $i = 1, 2$.*

Proof. For any Delaunay terminal triangles BAC of longest edge AB (see Figure 2(a)), the third vertex D of the neighbor terminal triangle ABD must be situated in the exterior of circumcircle CC(BAC) and inside the circles of center A, B and radius $\overline{AB}$. This defines a geometrical place R for D which reduces to one point when $\measuredangle\ BCA = 2\pi/3$ where OZ = r/2 (see Figure 2(b)), implying that $R = \phi$ when angle $BCA > 2\pi/3$ □

For a single longest edge bisection of any triangle t, into two triangles t_A, t_B, the following result holds:

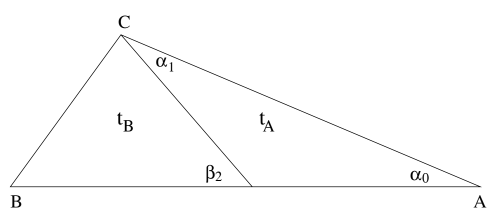

Fig. 3. Notation for longest edge bisection

Proposition 1. *For the longest edge bisection of any triangle t (see Figure 3), where $BC \leq CA \leq BA$, its holds that: a) $\alpha_1 \geq \alpha_0/2$ which implies $\beta_2 \geq 3\alpha_0/2$; b) If t is obtuse, $\alpha_1 \geq \alpha_0$ which implies $\beta_2 \geq 2\alpha_0$*

Lemma 1. *The longest edge bisection of any bad triangle BAC produces an improved triangle t_B and a bad quality triangle t_A. Usually t_A has largest angle greater than $2\pi/3$ and it is consequentely eliminated by edge swapping.*

3 Lepp (Terminal Edge) Midpoint Method

Given an angle tolerance θ_{tol}, the algorithm can be simply described as follows: iteratively, each bad triangle t_{bad} with smallest angle less than θ_{tol} in the current triangulation is eliminated by finding Lepp(t_{bad}), a pair of terminal triangles t_1, t_2, and associated terminal edge l. If non-constrained edges are involved, then the midpoint M of l is Delaunay inserted in the mesh. Otherwise the constrained point insertion criterion described below is used. The process is repeated until t_{bad} is destroyed in the mesh, and the algorithm finishes when the minimum angle in the mesh is greater than or equal to an angle tolerance θ_{tol}.

When the second longest edge CA is a constrained edge, the swapping of this edge is forbidden. In such a case, the insertion of point M would imply that the later processing of bad quality triangle MAC would introduce triangle MAM_1 (see Figure 4(a)) similar to triangle ABC implying an infinite loop situation. To avoid this behavior we introduce the following additional operation, which guarantees that M is not inserted in the mesh by processing triangle M_1BA.

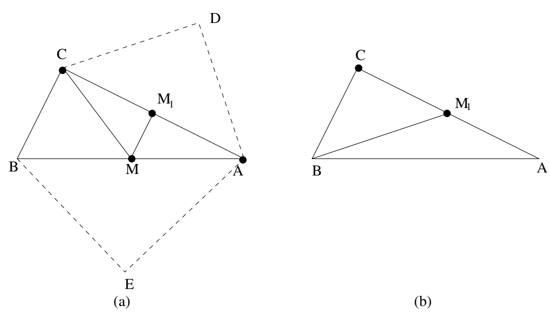

Fig. 4. (a) Over constrained edge CA, the insertion of M and M_1 produces triangle MAM_1 similar to triangle BAC; (b) Insertion of M_1 avoids this situation

Constrained edge point insertion: If CA is a constrained edge and BA is not a constrained edge, then insert midpoint M_1 of edge CA.

Special loop case. For Lepp-midpoint method, there is a rare special loop case discussed in [10], where a triangle MAM_1 similar to a bad-quality triangle

ABC can be obtained for a non-constrained edge CA. This happens when quadrilaterals BEAC and ADCM (see Figure 4(a)) are terminal quadrilaterals (where edges BA and CA are terminal edges respectively) together with some non-frequent conditions on neighbor constrained items. A necessary but not sufficient condition on the triangle ABC for this to happen is that angle $BMC \geq \pi/3$ which implies $\alpha_0 \geq \alpha_{limit} = arctan\frac{\sqrt{15}-\sqrt{3}}{3+\sqrt{5}} > 22°$ for obtuse triangle BAC [10]. This loop case can be avoided by adding some extra conditions to the algorithm. To simplify the analysis we restrict the angle tolerance to α_{limit} which is slightly bigger than the limit tolerance, equal to 20.7°, used to study both for the circuncenter and the off-center methods. The algorithm is given below:

Lepp Midpoint Algorithm
Input = a CDT, τ, and angle tolerance θ_{tol}
Find S_{bad} = the set of bad triangles with respect to θ_{tol}
for each t in S_{bad} **do**
  while t remains in τ **do**
    Find Lepp (t_{bad}), terminal triangles t_1, t_2 and terminal edge l. Triangle t_2 can be null for boundary l.
    Select Point (P, t_1, t_2, l)
    Perform constrained Delaunay insertion of P into τ
    Update S_{bad}
  end while
end for

Select Point (P, t_{term1}**,** t_{term2}**,** l_{term}**)**
if (second longest edge of t_{term1} is not constrained and second longest edge of t_{term2} is not constrained) or l_{term} is constrained **then**
  Select P = midpoint of l_{term} and return
else
  for j = 1,2 **do**
    if t_{termj} is not null and has constrained second longest edge l^* **then**
      Select P = midpoint of l^* and return
    end if
  end for
end if

Angle and edge size bounds for Lepp midpoint method. The results of this section improve results discussed in [23]. Firstly we present a characterization of Delaunay terminal triangles based on fixing the second longest edge CA and choosing the smallest angle at vertex A. The diagram of Figure 5 (a) shows the possible locations for vertex B and the midpoint M. The diagram is defined as follows: (1) Since CB is a shortest edge, B lies inside the circular arc EFA of centre C and radius $|\overline{CA}|$. Consequently, M lies inside the circular arc $E'F'A$

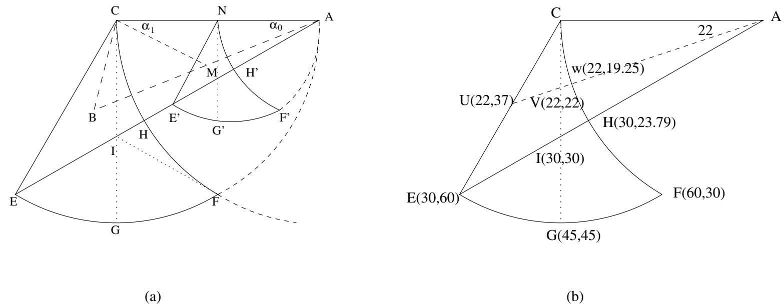

Fig. 5. (a) EFC and E'F'N' are geometrical places for vertex B and midpoint M for a terminal triangle BAC with respective smallest and largest angle of vertices A and C. (b) Distribution of angles (α_0, α_1).

of centre $N = (C + A)/2$ and radius $|\overline{CA}|/2$; (2) Since BA is a longest edge, B lies outside the circular arc CF of centre A and radius $|\overline{C - A}|$, and so M lies outside circular arc NF' of centre A, radius $|\overline{CA}|/2$; (3) According to Theorem 2, the line CE makes an angle of $120°$ with CA.

Now we use the diagram of Figure 5(a) both to improve the bounds on α_1 (Figure 3) and to bound the minimum distance from M to the previous neighbor vertices in the mesh. To this end consider the distribution of the ordered pair of angles (α_0, α_1) illustrated in Figure 5(b). As expected for the right triangles (B over CG) $\alpha_1 = \alpha_0$. Also note that the ratio α_1/α_0 decreases from 2 to 1 along line EC (obtuse triangles with largest angle equal to $2\pi/3$), while the ratio α_1/α_0 increases from 1/2 to 1 along arc F to C (isosceles acute triangles with two longest edges). Note that segment lines UW and EH correspond to fixed smallest angle equal to α_{limit} and $30°$ respectively.

Lemma 2. *For acute Delaunay terminal triangles of smallest angle α, there exist constants C_1, C_2 such that:*

a) $\alpha_0 \leq 30°$ (B in region CIH) implies $\alpha_1 \geq C_1\alpha_0$ with $C_1 \approx 0.79$.
b) $\alpha_0 \leq \alpha_{limit}$ (B in region CVW) implies $\alpha_1 \geq C_2\alpha_0$ $C_2 \approx 0.866$
c) The ratio α_1/α_0 approaches 1.0 both when α_0 decreases, and when BAC becomes a right triangle.
d) Using the notation of Figure 3, $\beta_2 \geq (1 + C_1)\alpha_0$ for $\alpha_0 \leq 30°$.

Proof for (d) note that $\beta_2 = \alpha_1 + \alpha_0$ (Figure 3) □

In order to bound the minimum distance from M to previous vertices in the mesh, we use both the properties of the longest edge bisection of a Delaunay terminal triangle BAC and the constraint on the empty circuncircle. Note that the circuncenter O of an obtuse (acute) triangle is situated in the exterior (the interior) of the triangle. Furthermore for any non constrained Delaunay obtuse triangle t, the distance $d = MO$ from the circuncenter O to the longest edge

BA (see Figure 2(c)) satisfies that $0 < d < r/2$, where r is the cicunradius. We will consider the limit cases $d = r/2$ and $d = 0$, which respectively correspond to largest angles equal to $2\pi/3$ and $\pi/2$, as well as the cases $\alpha = \alpha_{limit}$ and $\alpha = 14°$ to state bounds for obtuse and acute triangles.

Lemma 3. *Consider any Delaunay terminal triangle t of smallest angle α and terminal edge AB of midpoint M (Figure 2(c)), and let d(M) be the minimum distance from M to any vertex of the mesh. Then there exists constants C_1, C_2, C_3, C_4 such that*

a) For acute t
   (i) $\alpha \leq \alpha_{limit}$ implies $d(M) \geq C_1 \mid BC \mid$ with $C_1 > 1.3$
   (ii) $\alpha \leq 30°$ implies $d(M) \geq C_2 \mid BC \mid$ with $C_2 = 1$
b) For obtuse t
   (i) $\alpha \leq 14°$ implies $d(M) \geq C_3 \mid BC \mid$ with $C_3 > 1$
   (ii) $\alpha \leq \alpha_{limit}$ implies $d(M) \geq C_4 \mid BC \mid$ wiht $C_4 > 0.66$
   (iii) $\alpha \leq 30°$ implies $d(M) \geq C_5 \mid BC \mid$ with $C_4 = 0.5$
c) obtuse t with $\alpha > 14°$ implies $\beta_2 > 28° > \alpha_{limit}$

The following theorem based on Proposition 1 and Lemmas 2 and 3 assure that bad quality terminal triangles are quickly improved by introducing a sequence of better triangles of edge CB, but not introducing points too close to the previous vertices in the mesh. This improves the results presented in [23].

Theorem 3. *Consider $\theta_{tol} \leq \alpha_{limit}$. Then for any bad quality terminal triangle of smallest angle α, a finite sequence of improved triangles t_B (Figure 3), is obtained until t_B is good such that: (a) For $\alpha \leq 14°$, none edge smaller than the existing neighbor edges is inserted in the mesh. (b) Only at the last improvement step (when $\alpha > 14°$) a small smallest edge, at least 0.66 times the size of a previous neighbor smallest edge, can be occassionaly introduced in the mesh for obtuse triangle t.*

Remark: Note that the worst $d(M)$ value is obtained for obtuse triangles, for which in turn the angles are most improved.

4 Lepp Centroid Algorithm

The Lepp centroid method was designed both to avoid the loop situation discussed in section 3 and to improve the slower convergence reported in [11] for $\theta_{tol} > 25°$ due to the fact that good quality acute terminal triangles can produce a slightly bad triangle t_A (Figure 3). Instead of selecting an edge aligned midpoint M, we select the centroid of a terminal quad defined as the quadrilateral $ACBD$ formed by a couple of terminal triangles BAC and BDA (Figure 6) sharing an unconstrained terminal edge. The algorithm is given below:

Lepp–Terminal-Centroid Algorithm
Input = a CDT, τ, and angle tolerance θ_{tol}
Find S_{bad} = the set of bad triangles with respect to θ_{tol}
for each t in S_{bad} **do**
  while t remains in τ **do**
    Find Lepp (t_{bad}), terminal triangles t_1, t_2 and terminal edge l. Triangle t_2 can be null for boundary l.
    Select Point (P, t_1, t_2, l)
    Perform constrained Delaunay insertion of P into τ
    Update S_{bad}
  end while
end for

New Select Point (P, t_{term1}, t_{term2}, l_{term})
if l_{term} is constrained **then**
  Select P = midpoint of l_{term} and return
end if
if (second longest edge of t_{term1} is not constrained and second longest edge of t_{term2} is not constrained) **then**
  Select P = centroid of quad (t_{term1}, t_{term2}) and return
else
  for j = 1,2 **do**
    if t_{termj} is not null and has constrained second longest edge l^* **then**
      if t_{termj} does not contain the centroid **then**
        Select P = midpoint of l^* and return
      else
        Select P = centroid of quad $((t_{term1}, t_{term2})$
      end if
    end if
  end for
end if

Geometrical properties of the centroid selection. Consider a terminal quadrilateral $ACBD$ formed by the union of a pair of terminal triangles ABC, ABD sharing the terminal edge AB (Figure 6). In what follows we will prove that inserting the terminal centroid Q, defined as the centroid of the terminal quadrilateral $ACBD$, always produce a better new triangle t_B (Figure 3). Consider a coordinate system based on a bad quality triangle ABC of longest edge BA, with center in B and x-axis over BA as shown in Figure 6. This is the longest edge coordinate system introduced by Simpson [25]. As discussed in the proof of Theorem 2, D must lie in the 'triangle' UVW defined by circumcircle $CC(ABC)$ and the lens VA and VB which are arcs of the circles of radious BA and respective centers A and B. The coordinates of the vertices are $B(0,0)$, $A(l_{max}, 0)$, $C(a_c, b_c)$ and $D(a_D, b_D)$.

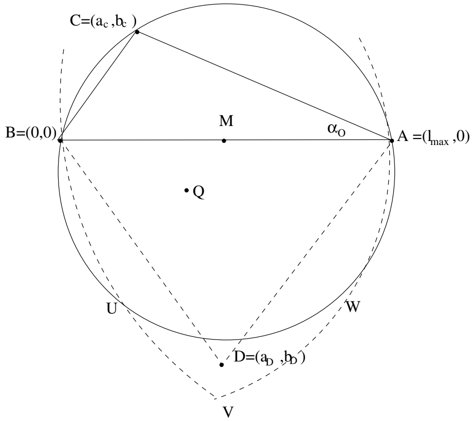

Fig. 6. Estimating location of centroid Q

Let $U(a_{min}, b_{min})$ $[W(a_{max}, b_{min}]$ be the leftmost $[rightmost]$ intersection point of the lens and $CC(ABC)$ as shown. The following lemma bounds the location of Q by parameters determined by triangle ABC.

Lemma 4. *Let the centroid Q have longest edge coordinates* (a_Q, b_Q) *with respect to triangle ABC. Then*

$$a_{min}/4 + l_{max}/4 < a_Q < a_{max}/4 + l_{max}/2, \text{ and } b_V/4 < b_Q < (b_c + b_{min})/4$$

Proof. *The proof follows from computing the centroid coordinates*

$$a_Q = (a_C + a_D + l_{max})/4 \text{ and } b_Q = (b_c + b_D)/4$$

Corollary 1. *For any terminal quad involving a pair of (obtuse acute) terminal triangles, the quad centroid is situated in the interior of the acute triangle.*

Corollary 2. *Let* t_a, t_b *be terminal triangles forming a terminal quadrilateral, and sharing a terminal edge, E. The shortest edge that results from the insertion of the terminal centroid into the mesh is longer than the shortest edge that results from inserting the midpoint of E.*

Theorem 4. *The algorithm does not suffer of the special loop case (section 3).*

Proof. The centroid Q is not aligned with the vertices of the terminal edge excepting the case of right isosceles triangles sharing a longest edge.

Remark 1. The constraint on θ_{tol} is not necessary and the analysis of the algorithm can be extended until $\theta_{tol} = 30°$.

5 Empirical Study and Concluding Remarks

We consider the 3 test problems of Figure 7 whose (bad quality) initial triangulations are shown in this figure. They correspond to a square with 400

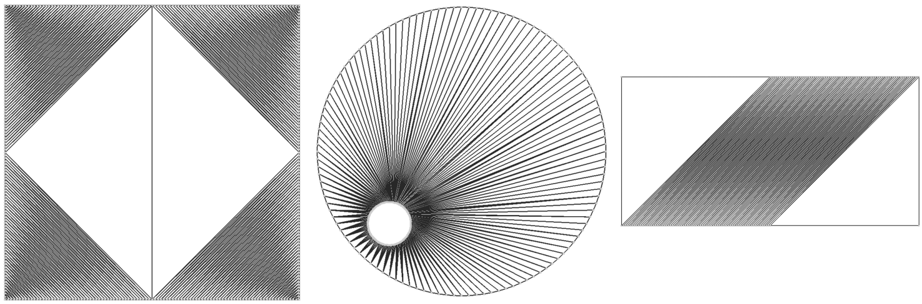

Fig. 7. Initial triangulations for the test problems

equidistributed boundary points (Square400) and discretized circle having a discretized circular hole close to the boundary with 240 boundary points (Circle 240), a rectangle having 162 boundary points distributed as shown in Figure 7 (Rectangle 162). We also consider an A test case having 21 boundary points, provided at the Triangle site. The initial triangulations are Delaunay excepting the triangulation of the Rectangle 162 case, which is an example proposed by Edelsbrunner [4] for the point triangulation problem.

Lepp centroid versus Lepp midpoint. We used the test cases of Figure 7 to compare the algorithms. We ran every case for different θ_{tol} values, with the input meshes of Figure 7, until reaching the maximum practical tolerance angle θ_{tol}. We studied the mesh quality both with respect to the smallest angle and with respect to the area quality measure defined as $q(t) = CA(t)/l^2$, where $A(t)$ is the area of triangle t, l is its longest edge and C is a constant such that $q(t) = 1$ for the equilateral triangle. For both algorithms we studied the evolution of the minimum smallest angle, the minimum area quality measure, the average smallest angle, the normalized minimum edge size (wrt the minimum edge size in the input mesh) and the average Lepp size. As expected the smallest angle and the area quality measure show analogous behavior.

The empirical study shows that for every test problem, the Lepp centroid method computes significantly smaller triangulations than the Lepp midpoint variant and terminates for higher threshold angle θ_{tol}. The Lepp centroid works for θ_{tol} up to 36° while the previous midpoint method works for $\theta_{tol} \leq 30°$ for these examples. If is worth noting that the Lepp centroid method produces globally better triangulations having both significantly higher average smallest angle and a smaller percentage of bad quality triangles than the Lepp midpoint variant for the same θ_{tol} value. To illustrate this see Figure 8 and Figure 9 which respectively show the behavior of the minimum area quality measure and of the average smallest angles for both algorithms. This is also illustrated in Figure 11 which shows the evolution of the area quality distribution for the Square 400 test problem for θ_{tol} equal to $10°, 25°$.

The normalized minimum edge size in the mesh (wrt the smallest edge in the initial mesh) is shown in Figure 10. Note that this parameter behaves better

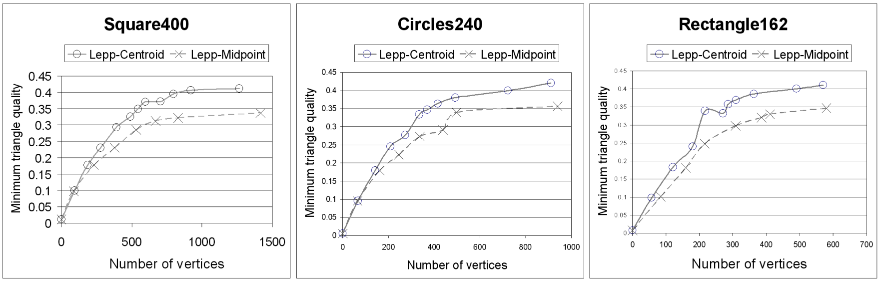

Fig. 8. Evolution of the minimum area quality measure as a function of the number of vertices for Lepp-centroid and Lepp-midpoint algorithms

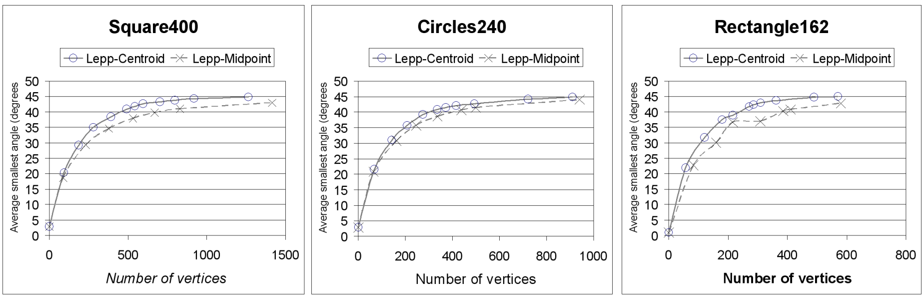

Fig. 9. Evolution of the average smallest angle (degrees) as a function of the number of vertices for Lepp-centroid and Lepp-midpoint algorithms

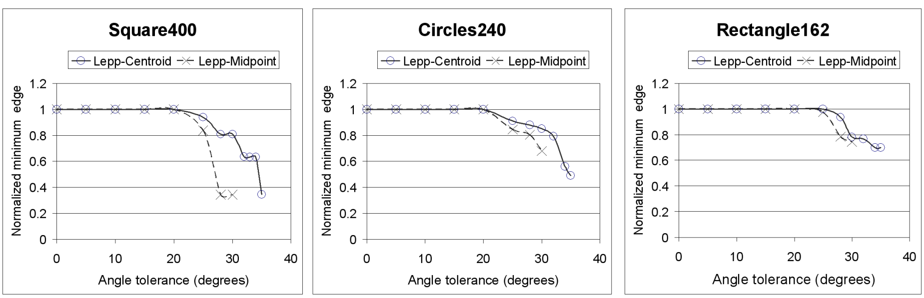

Fig. 10. Normalized minimum edge size (wrt the smallest edge in the initial mesh) as a function of the angle tolerance θ_{tol} for centroid and midpoint algorithms

than predicted by the theoretical results of section 3, even for the Lepp midpoint method. For all these problems the average Lepp size remains between 3 to 5, being this value slightly higher for the Lepp centroid method. Finally the triangulation obtained with the Lepp-centroid method (and for the Triangle method) for the Circles 240 test problem for $\theta_{tol} = 32°$ is shown in Figure 12.

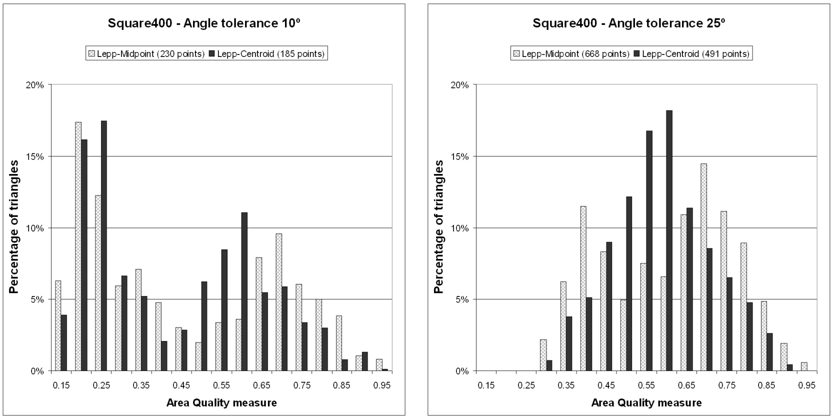

Fig. 11. Area quality distribution for Square 400 for $\theta_{tol} = 10°$ (midpoint 230 points, centroid 185 points) and $\theta_{tol} = 25°$ (midpoint 668 points, centroid 491 points)

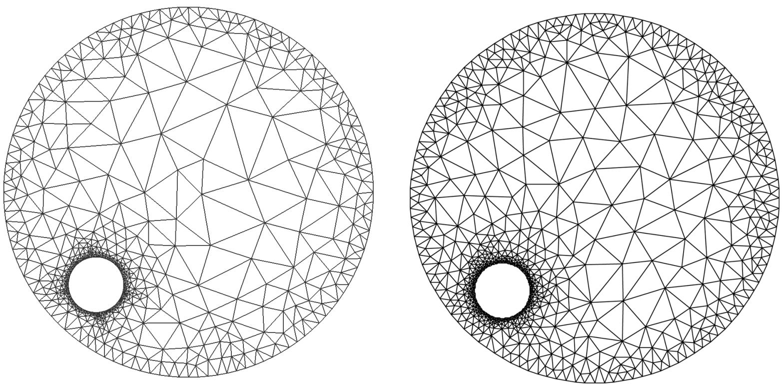

Fig. 12. Triangulations obtained with Lepp-centroid and Triangle methods, $\theta_{tol} = 32°$

A comparison with Triangle. As reported previously in [11] the Lepp midpoint method showed a behavior analogous to the circumcenter algorithm implemented in a previous Triangle version [15]. Here we use the current Triangle version based on the off-center point selection [19,24], to perform a comparison with the Lepp centroid method for the test problems of Figure 7 and the A test case. The evolution of the minimum angle in the mesh as a function of the number of vertices is shown in Figure 13 for the same set of θ_{tol} values used in the preceding subsection. Note that for all these cases the Lepp-centroid method worked well (with reasonable number of vertices inserted) for θ_{tol} up to 36°, while that the Triangle method worked for θ_{tol} up to 35° but increasing highly the number of points inserted for θ_{tol} bigger than 25°. Only for the small A-test-case, where basically boundary points are inserted, both algorithms have more

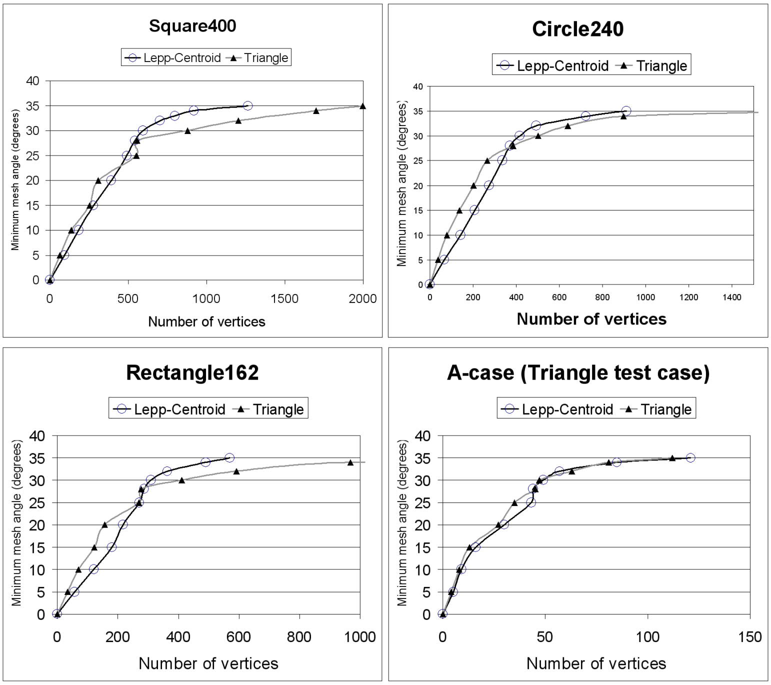

Fig. 13. Evolution of the minimum angle in the mesh as a function of the number of vertices for Lepp-centroid and Triangle methods

Number of points added								
Minimum mesh angle (degrees)	Square400		Circles240		Rectangle162		A-case	
	Centroid	Triangle	Centroid	Triangle	Centroid	Triangle	Centroid	Triangle
20	391	310	274	202	217	156	30	27
25	491	553	335	266	271	270	43	35
28	543	557	371	386	286	277	44	45
30	595	880	416	501	310	411	49	47
32	703	1202	493	639	363	592	57	63
34	919	1700	722	897	490	968	85	81
35	1264	1997	911	1811	570	4853	121	112
36	1843	-	1630	-	5473	-	239	-

similar behavior. A quantitative view of the behavior of both algorithms is given in the Table below which shows the number of points added to the mesh for all the test problems and different values of θ_{tol}. The triangulations obtained for the Circles240 case with the Lepp-centroid and Triangle methods for $\theta_{tol} = 32°$ are shown in Figure 12.

Concluding remarks. The results of this paper suggest that for Lepp-centroid method the algorithm analysis can be extended until 30°. In effect, in section 3 we prove that for the Lepp-midpoint method, under the constraint $\theta_{tol} \leq \alpha_{limit} \approx 22°$, and for improving a bad quality triangle ABC and its t_B sons (Figure 3), a small number of points is introduced which are not situated too close to the previous vertices in the mesh. In exchange, the results of section 4 guarantee both that the centroid method does not suffer of special looping conditions and that smallest edges bigger than those introduced by the midpoint method are introduced, which suggest that for the Lepp-centroid method the analysis can be extended for $\theta_{tol} \leq 30°$. In this paper we also provide empirical evidence that show that the Lepp-centroid method behaves in practice better than the off-center algorithm for $\theta_{tol} > 25°$.

Recently Erten and Ungor [24] have introduced algorithms that improve the off-center performance with respect to the mesh size and the angle θ_{tol} by using point selections depending on some triangle cases. We plan to improve the Lepp-based algorithms also in this direction.

Acknowledgements. Research supported by DI ENL 07/03. We are grateful with Bruce Simpson who contributed to an early formulation of this paper.

References

1. Chew, L.P.: Guaranteed-quality triangular meshes. Technical report TR-98-983, Computer Science Department, Cornell University, Ithaca, NY (1989)
2. George, P.L., Borouchaki, H.: Delaunay Triangulation and Meshing. In: Hermes (1998)
3. Borouchaki, H., George, P.L.: Aspects of 2-D Delaunay Mesh Generation. International Journal for Numerical Methods in Engineering 40, 1997 (1975)
4. Edelsbrunner, H.: Geometry and Topology for Mesh Generation. In: Cambridge Monographs on Applied and Computational Mathematics, Cambridge Univ. Press, Cambridge (2001)
5. Rivara, M.C.: Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. International Journal for Numerical Methods in Engineering 20, 745–756 (1984)
6. Rivara, M.C.: Selective refinement/derefinement algorithms for sequences of nested triangulations. International Journal for Numerical Methods in Engineering 28, 2889–2906 (1989)
7. Rivara, M.C., Levin, C.: A 3d Refinement Algorithm for adaptive and multigrid Techniques. Communications in Applied Numerical Methods 8, 281–290 (1992)
8. Rivara, M.C.: New mathematical tools and techniques for the refinement and / or improvement of unstructured triangulations. In: Proceedings 5th International Meshing Roundtable, Pittsburgh, pp. 77–86 (1996)
9. Rivara, M.C.: New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. International Journal for Numerical Methods in Engineering 40, 3313–3324 (1997)
10. Rivara, M.C., Hitschfeld, N., Simpson, R.B.: Terminal edges Delaunay (small angle based) algorithm for the quality triangulation problem. Computer-Aided Design 33, 263–277 (2001)

11. Rivara, M.C., Palma, M.: New LEPP Algorithms for Quality Polygon and Volume Triangulation: Implementation Issues and Practical Behavior. In: Cannan, S.A. (ed.) Trends unstructured mesh generation, Saigal, AMD, vol. 220, pp. 1–8 (1997)
12. Ruppert, J.: A Delaunay refinement algorithm for quality 2-dimensional mesh generation. J. of Algorithms 18, 548–585 (1995)
13. Baker, T.J.: Automatic mesh generation for complex three dimensional regions using a constrained Delaunay triangulation. Engineering with Computers 5, 161–175 (1989)
14. Baker, T.J.: Triangulations, Mesh Generation and Point Placement Strategies. In: Caughey, D. (ed.) Computing the Future, pp. 61–75. John Wiley, Chichester (1995)
15. Shewchuk, J.R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: First Workshop on Applied Computational Geometry, pp. 124–133. ACM Press, New York (1996)
16. Aurehammer, F., Katoh, N., Kokima, H., Ohsaki, M., Xu, Y.-F.: Approximating uniform triangular meshes in polygons. Theoretical Computer Science, 879–895 (2002)
17. Weatherill, N.P., Hassan, O.: Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. International Journal for Numerical Methods in Engineering (2039)
18. Rosenberg, I.G., Stenger, F.: A lower bound on the angles of triangles constructed by bisecting the longest side. Mathematics of Computation 29, 390–395 (1975)
19. Üngor, A.: Off-centers: a new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. In: Orejas, F., Spirakis, P.G., van Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 152–161. Springer, Heidelberg (2001)
20. Simpson, B., Rivara, M.C.: Geometrical mesh improvement properties of Delaunay terminal edge refinement. In: Kim, M.-S., Shimada, K. (eds.) GMP 2006. LNCS, vol. 4077, pp. 536–544. Springer, Heidelberg (2006)
21. Stynes, M.: On Faster Convergence of the Bisection Method for certain Triangles. Mathematics of Computation 33, 1195–1202 (1979)
22. Stynes, M.: On Faster Convergence of the Bisection Method for all Triangles. Mathematics of Computation 35, 1195–1202 (1980)
23. Rivara, M.C.: A study on Delaunay terminal edge method. In: Pébay, P.P. (ed.) Proceedings of the 15th International Roundtable, pp. 543–562. Springer, Heidelberg (2006)
24. Erten, H., Üngor, A.: Triangulations with locally optimal Steiner points. In: Belyaev, A., Garland, M. (eds.) Eurographics Symposium on Geometry Processing (2007)
25. Simpson, R.B.: Gometry Independence for a Meshing Engine for 2D Manifolds. International Journal for Numerical Methods in Engineering, 675–694 (2004)