Computer-Aided Design 42 (2010) 58–66

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

# Lepp terminal centroid method for quality triangulation

Maria-Cecilia Rivara *, Carlo Calderon

*Department of Computer Science, University of Chile, Blanco Encalada 2120, Santiago, Chile*

ARTICLE INFO

*Article history:*
Received 15 July 2008
Accepted 11 November 2008

*Keywords:*
Mesh generation
Triangulations
Lepp-Delaunay algorithms
Lepp midpoint
Lepp centroid

ABSTRACT

We discuss Lepp-centroid versus Lepp-midpoint algorithms for Delaunay quality triangulation. We present geometrical results that ensure that the centroid version produces triangulations with both average smallest angles greater than those obtained with the midpoint version and with bigger smallest edges, without suffering from a rare looping case associated to the midpoint method. Empirical study shows that the centroid method behaves significantly better than the midpoint version (and than the off-center algorithm for angles bigger than 25°), for geometries whose initial Delaunay triangulation have triangle smallest edges over the boundary.



## 1. Introduction

In the last two decades, the triangular mesh generation subject has evolved into an important and interdisciplinary research field. Triangulations have been used as an important methodology for many problems in different applications such as finite element analysis of complex physical problems modeled by partial differential equations, computer graphics applications, geometric modeling, geographical information systems, terrain modeling and real time rendering.

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 [1,2] and Rivara and collaborators, [3–6]. Mesh improvement properties for iterative Delaunay refinement based on inserting the circumcenter of triangles to be refined have been established by Chew, [7], Ruppert [8], and Shewchuk [9]. Applications of this form of refinement have been described by Weatherill et al. [10] and Baker [11]. Baker also published a comparison of edge based and circumcenter based refinement [12]. Algorithms based on off-center insertions have been recently presented by Üngör and collaborators [13,14]. Algorithms for uniform triangular meshes are discussed in [15]. For a theoretical review on mesh generation see the monograph of Edelsbrunner [16].

This paper discusses and compares geometrical properties and empirical behavior of Lepp-centroid and Lepp (terminal edge) midpoint methods for quality Delaunay triangulation, and is an extended and improved version of the paper presented at GMP2008 [17].

### 1.1. 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 vertex (Fig. 3). Longest edge based algorithms [18–20,3,4,6] were designed to take advantage of the following mathematical properties on the quality of triangles generated by iterative longest edge bisection of triangles [21–23], and require a reasonably good quality input triangulation to start with.

**Theorem 1.** *For any triangle $t_0$ of smallest angle $\alpha_0$.*

(i) *The iterative longest edge bisection of $t_0$ ensures that for any longest edge son $t$ of the smallest angle $\alpha_t$, it holds that $\alpha_t \geq \alpha_0/2$, the equality holding only for equilateral triangles.* (ii) *A finite number of triangle similarity classes is generated.* (iii) *The area of $t_0$ tends to be covered by quasi-equilateral triangles (for which at most 4 triangle similarity classes of good quality triangles are produced).*

#### 1.1.1. 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 the 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 (see Section 2). The algorithm was introduced in a rather intuitive basis as a generalization of previous longest edge algorithms in [4–6]. This was supported by the improvement properties of both the

* Corresponding author. Tel.: +56 2 9784945; fax: +56 2 6895531.
*E-mail addresses:* mcrivara@dcc.uchile.cl (M.-C. Rivara), carcalde@dcc.uchile.cl (C. Calderon).

0010-4485/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2008.11.004

longest edge bisection of triangles (Theorem 1) and the Delaunay algorithm, and by the result presented in Theorem 2 in the next section. Later in [5] 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 [4–6], formal proofs on algorithm termination and on optimal size property have not been fully established due to the difficulty of the analysis. Recently in [24] we have presented some geometrical improvement properties of an isolated insertion of a terminal edge midpoint $M$ in the mesh. In [25] a first termination proof is presented and several geometric aspects of the algorithm are studied.

### 1.2. Point insertion strategies for constrained edges

For constrained edges, both the circumcenter and the off-center algorithm use the following encroachment strategy: if a prospective point $P$ to be inserted is contained in the diameter circle of any constrained edge $E$, the midpoint of $E$ is inserted instead of $P$, and implies that a strict Delaunay triangulation is maintained. This imposes a strong geometry restriction: no angle less than 90° can appear in the geometry. In exchange, for the Lepp-Delaunay midpoint algorithm, a real *CDT* is allowed by only considering local information associated to the terminal triangle that contains such constrained edge. This avoids the interaction with the whole set of constrained items, which simplifies the algorithm implementation, specially when the algorithm is generalized to 3-dimensions.

### 1.3. Lepp-centroid algorithm

In order to improve the performance of the previous Lepp-midpoint algorithm, in [17] we have introduced a new Lepp-centroid algorithm for quality triangulation. For any general (planar straight line graph) input data, and a quality threshold angle $\theta$, the algorithm constructs constrained Delaunay triangulations that have all angles at least $\theta$ as follows: for every bad triangle t with smallest angle less than $\theta$, 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 this paper we prove that the centroid version produces triangulations both with average smallest angles greater than those obtained with the midpoint version, and with bigger smallest edges without suffering from a rare looping case associated to the midpoint method.

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. For the Lepp-midpoint method, 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 this method. In Section 5 we present an empirical study that compares the behavior of Lepp-centroid and Lepp-midpoint methods for geometries whose initial Delaunay triangulation have triangle smallest edges over the boundary. The centroid method computes significantly smaller triangulation than the terminal edge midpoint variant, produces globally better triangulations, and terminates for higher threshold angle $\theta$ (up to 36°). We also show that the Lepp-centroid method behaves better than the off-center algorithm for $\theta > 25°$.

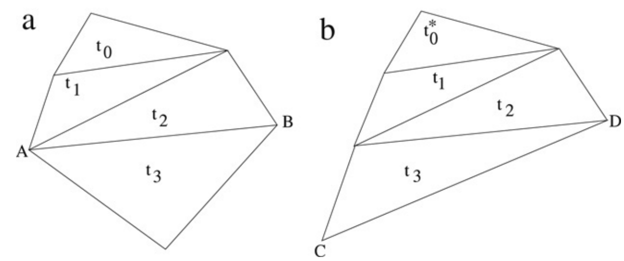


**Fig. 1.** (a) AB is an interior terminal edge shared by terminal triangles $(t_2, t_3)$ associated to Lepp$(t_0) = \{t_0, t_1, t_2, t_3\}$; (b) CD is a boundary terminal edge with unique terminal triangle $t_3$ associated to Lepp$(t_0^*) = \{t_0^*, t_1, t_2, t_3\}$.

## 2. Concepts and preliminary results

The following concepts were introduced and used in references [4–6]. An edge $E$ is called a terminal edge in triangulation $\tau$ if $E$ is the longest edge of every triangle that shares $E$, while the triangles that share $E$ are called terminal triangles. 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 Fig. 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 $t_3$.

For any triangle $t_0$ in $\tau$, 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 $j = 1, \ldots N$. Edge $E$ = longest-edge$(t_{N+1})$ = longest-edge$(t_N)$ is a terminal edge in $\tau$ and 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 $\tau$, or $E$ is shared by a unique terminal triangle $t_N$ with boundary (constrained) longest edge. See Fig. 1.

For a Delaunay mesh, an unconstrained terminal edge imposes the following constraint on the largest angles of the associated terminal triangles [3,4,6]:

**Theorem 2.** *For any pair of Delaunay terminal-triangles $t_1, t_2$ sharing a non-constrained terminal edge, largest angle $(t_i) \leq 2\pi/3$ for $i = 1, 2$. Furthermore when the equality on the bound on the largest angle holds for one of the triangles (say triangle ABC), then the other triangle is equilateral, the 4 vertices are cocircular and the distance from the terminal edge midpoint to the circumcenter is maximal and equal to half of the circumradius (see Fig. 2(b)).*

**Proof Sketch.** For any pair of Delaunay terminal triangles BAC, BAD of longest edge AB (see Fig. 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 Fig. 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:

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

**Lemma 4.** *The longest edge bisection of any bad triangle BAC produces an improved triangle $t_B$ and a bad quality obtuse triangle $t_A$.*

**Remark 5.** Note that usually $t_A$ has largest angle greater than $2\pi/3$ and it is consequently eliminated by edge swapping throughout the process.

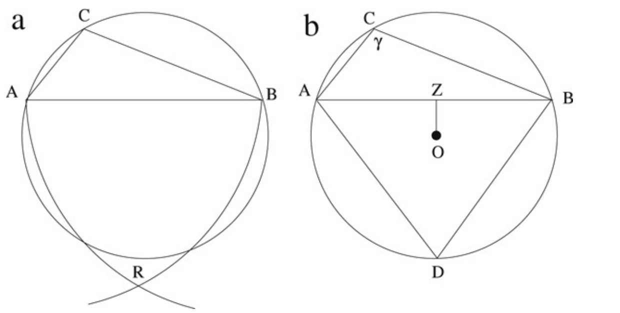


**Fig. 2.** R is the geometrical place of the fourth vertex D for Delaunay terminal triangles ABC, ABD; (b) R reduces to one point when $\gamma = 2\pi/3$ (triangle ADB equilateral).

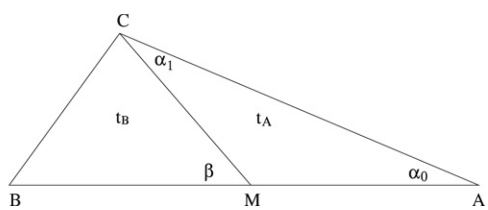


**Fig. 3.** Notation for longest edge bisection.

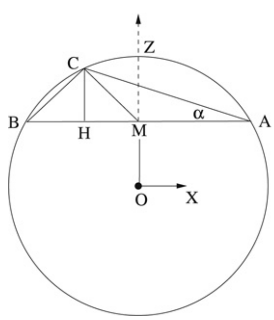


**Fig. 4.** For a bad terminal triangle BAC, Lepp-midpoint method inserts midpoint $M$.

## 3. Lepp (terminal edge) midpoint method

Given an angle tolerance $\theta_{tol}$, the algorithm can be simply described as follows: iteratively, each bad triangle $t_{bad}$ with smallest angle less than $\theta_{tol}$ in the current triangulation is eliminated by finding Lepp($t_{bad}$), a pair of terminal triangles $t_1$, $t_2$, and associated terminal edge $l(AB)$ (see Fig. 4). 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 $\theta_{tol}$.

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 Fig. 5(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 ABC.

**Constrained edge point insertion**: If CA is a constrained edge and $BA$ is a non-constrained edge, then insert midpoint $M_1$ of edge CA.

**Special loop case**. For the Lepp-midpoint method, there is a rare special loop case discussed in [5], where a triangle $MAM_1$ similar to a bad-quality triangle ABC can be obtained for a non-constrained

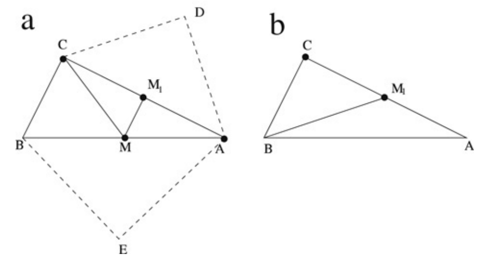


**Fig. 5.** (a) Over constrained edge CA, the insertion of $M$ and $M_1$ produces triangle $MAM_1$ similar to triangle BAC; (b) Insertion of $M_1$ avoids this situation.

edge CA. This happens when quadrilaterals BEAC and ADCM (see Fig. 5(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 [5]. This loop case can be avoided by adding some extra conditions to the algorithm. To simplify the analysis we restrict the angle tolerance to $\alpha_{limit}$ which is slightly bigger than the limit tolerance, equal to 20.7°, used to study both for the circumcenter and the off-center methods. The algorithm is given below:

**Lepp Midpoint Algorithm**
Input = a CDT, $\tau$, and angle tolerance $\theta_{tol}$
Find $S_{bad}$ = the set of bad triangles with respect to $\theta_{tol}$
**for** each t in $S_{bad}$ **do**
  **while** t remains in $\tau$ **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 $\tau$
    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**

### 3.1. Angle and edge size bounds for the Lepp-midpoint method

The results of this section improve results discussed in [17, 25]. 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 Fig. 6(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 center $C$ and radius $|\overline{CA}|$. Consequently, $M$ lies inside the circular arc $E'F'A$ of center $N = (C + A)/2$ and radius $|\overline{CA}|/2$; (2) Since $BA$ is a longest edge, $B$ lies outside the circular arc $CF$ of center $A$ and radius $|\overline{C-A}|$, and so $M$ lies outside circular arc $NF'$ of center $A$, radius $|\overline{CA}|/2$; (3) According to Theorem 2, the line $CE$ makes an angle of 120° with $CA$.

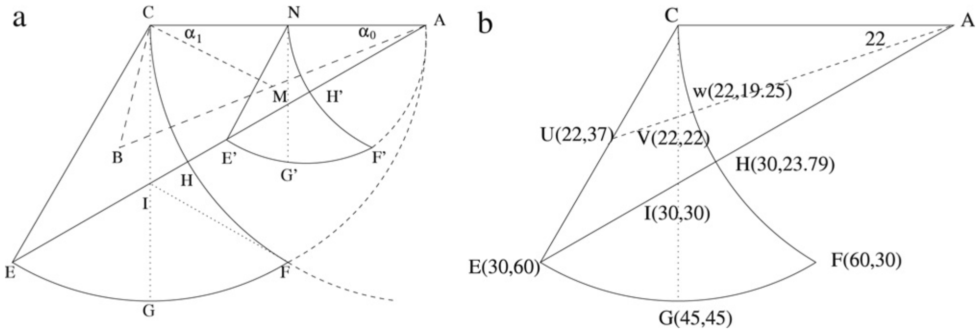


**Fig. 6.** (a) EFC and $E'F'N'$ are geometrical places for vertex B and midpoint $M$ for a terminal triangle BAC with respective smallest and largest angles of vertices A and C. (b) Distribution of angles $(\alpha_0, \alpha_1)$.

Now we use the diagram of Fig. 6(a) both to improve the bounds on $\alpha_1$ (Fig. 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 $(\alpha_0, \alpha_1)$ illustrated in Fig. 6(b). A sketch of the ideas involved in the proofs is as follows: For the right triangles ($B$ over $CG$) $\alpha_1 = \alpha_0$. The ratio $\alpha_1/\alpha_0$ decreases from 2 to 1 along line $EC$ (obtuse triangles with largest angle equal to $2\pi/3$), while the ratio $\alpha_1/\alpha_0$ 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 $\alpha_{limit}$ and 30° respectively.

The following proposition simplifies the analysis:

**Proposition 6.** *Let ABC be any Delaunay terminal triangle of longest edge AB and smallest angle $\alpha_0$, with $\alpha_0 \leq 30°$ (see Fig. 3).*

(a) *Let $d(M)$ be the minimum distance from M, midpoint of AB, to the previous vertices in the mesh. Then*
 (i) *If t is acute then $d(M) = BM$*
 (ii) *If t is obtuse then $d(M) = CM$.*
(b) *Consider the set of acute Delaunay terminal triangles with smallest edge $\alpha_0$ and fixed edge CA, then the smallest $d(M)/BC$ ratio corresponds to the isosceles triangle for which $BA = CA$.*
(c) *Consider the set of obtuse Delaunay terminal triangles with smallest angle $\alpha_0$ and fixed edge CA, then the smallest $d(M)/BC$ ratio corresponds to the most obtuse triangle of angles $2\pi/3$, $\alpha_0$, $(\pi/3 - \alpha_0)$.*

**Proof.** Consider Fig. 7 that shows the possible locations of vertex $B$ ($B$ going from $B_0$ to $B_i$) of all the Delaunay terminal triangles of smallest angle $\alpha_0$ and fixed edge $CA$. Note that $B_0, B_r, B_i$ respectively correspond to the triangle of largest angle equal to $2\pi/3$, the right triangle, and the isosceles triangle ($B_iA = CA$), whose respectively longest edge midpoints are $M_0, M_r, M_i$.

Now consider the right triangle $B_rAC$. For this triangle $B_rM_r = CM_r$ (for $\alpha_0 = 30°$ also $B_rM_r = B_rC$). Then for any acute triangle $BAC$ ($B$ going from $B_r$ to $B_i$), $CM$ increases as $BM$ and $CB$ decrease, which implies that $d(M) = BM$, and the result of (a) (i) follows. Note also that for these triangles, the worst d(M)/BM ratio is obtained for the isosceles triangle, $B = B_i$, which implies (b). On the other hand, for any obtuse triangle $BAC$ ($B$ going from $B_r$ to $B_0$), $CM$ decreases, as $BM$ and $CB$ increase, which implies that $d(M) = CM$, the result of (b) (ii) holds. Note also that the worst $d(M)/BC$ ratio is obtained for the most obtuse triangle with $B = B_0$, which implies the result of (c). □

**Lemma 7.** (I) *For acute Delaunay terminal triangles of smallest angle $\alpha_0$ (see Fig. 3), 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* $\alpha_1 \geq C_2\alpha_0$ *with* $C_2 \approx 0.866$

(c) *The ratio* $\alpha_1/\alpha_0$ *approaches* 1.0 *both when* $\alpha_0$ *decreases, and when BAC becomes a right triangle.*

(d) *Using the notation of Fig. 3,* $\beta \geq (1 + C_1)\alpha_0$ *for* $\alpha_0 \leq 30°$.

(II) *For an obtuse Delaunay terminal triangle,* $\alpha_1 > \alpha_0$.

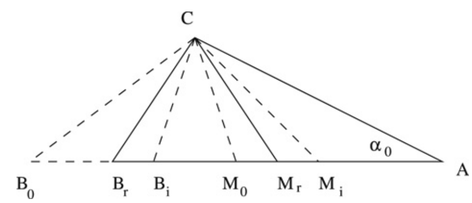


**Fig. 7.** Delaunay terminal triangles of smallest angle $\alpha_0$ and fixed second longest edge $CA$. For acute triangles, vertex B goes from $B_r$ (right triangle) to $B_i$ (isosceles triangle). For obtuse triangle, $B$ goes from $B_0$ (obtuse triangle) to $B_r$. Points $M_0$, $M_r$ and $M_i$ are midpoints of $B_0A$, $B_rA$ and $B_iA$ respectively.

**Proof.** (I) Consider the acute Delaunay terminal triangles of smallest angle $\alpha_0$, which in the Fig. 7 correspond to the triangles of vertex $B$ going from $B_r$ (right triangle) to $B_i$ (isosceles triangle). Note that for any acute Delaunay terminal triangle with fixed smallest angle $\alpha_0$, the angle $\alpha_1$ is minimum for the isosceles triangle with $B_iA = CA$. In Fig. 7, the points $M_r$, $M_i$ correspond to the terminal edge midpoints for right and isosceles triangle respectively. Now assume $B_iA = 1$ which implies that $B_iC = 2\sin(\alpha_0/2)$. Then use the cosine theorem over triangle $B_iMC$ for computing $CM_i$ and the following sine formula (applied over triangle $M_iAC$) for computing $\alpha_1$:

$$\frac{\sin\alpha_1}{\sin\alpha_0} = \frac{1/2}{CM}. \tag{1}$$

In particular, for the case $\alpha_0 = 30°$, this computation gives $\alpha_1 > 23.79°$, which implies $\alpha_1/\alpha_0 > 0.79$. Now note that for these isosceles triangles, $CM$ decreases as $\alpha_0$ decreases, and according to (1), the ratio $\alpha_1/\alpha_0$ is minimum for the biggest value of $\alpha_0$, and the result of (a) follows. The result of (b) is obtained analogously. Finally note that $CM$ tends to be equal to $BM$ when $\alpha_0$ approaches 0, which implies (c). The result of (d) follows from the fact that angle $BMC$ is equal to $\alpha_0 + \alpha_1$ for any triangle $ABC$.

(II) Follows directly from part (b) of Proposition 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 circumcircle. Note that the circumcenter O of an obtuse (acute) triangle is situated in the exterior (the interior) of the triangle. Furthermore for any non-constrained obtuse Delaunay terminal triangle t, the distance $d = MO$ from the circumcenter O to the longest edge BA (see Fig. 4) satisfies that $0 < d < r/2$, where r is the circumradius. 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 8.** *Consider any Delaunay terminal triangle t of smallest angle $\alpha$ and terminal edge AB of midpoint M (Fig. 4), and let $d(M)$ be* *the minimum distance from M to any vertex of the mesh. Then there exists constants $C_1, C_2, C_3, C_4, C_5$ 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* $C_2 > 0.96$.
(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* $d(M) \geq C_4 \mid BC \mid$ *with* $C_4 > 0.66$
  (iii) $\alpha \leq 30°$ *implies* $d(M) \geq C_5 \mid BC \mid$ *with* $C_5 = 0.5$
(c) *obtuse t with* $\alpha > 14°$ *implies* $\beta_2 > 28° > \alpha_{limit}$.

**Proof.** (a) According to part (b) of Proposition 6, for acute triangles with smallest angle $\alpha \leq \tilde{\alpha}$, the worst ratio $d(M)/BC$ holds for the isosceles triangle $B_iAC$ with $B_iA = CA$ and smallest angle $\tilde{\alpha}$. Also note that for the right triangles with $\alpha \leq 30°$, when $\alpha$ decreases, the ratio $d(M)/BC$ increases. Then for acute triangles of smallest angle $\alpha \leq \tilde{\alpha}$, it suffices to study the acute isosceles triangle of smallest angle $\tilde{\alpha}$. In particular for proving a(ii), corresponding to the case $\alpha \leq 30°$, assume $BA = 1$, compute $BC = 2\sin 15° = 0.517638$ and $MB/BC < 0.9659$, and the result follows. For $\alpha \leq \alpha_{limit}$, assume $BA = 1$, compute $BC = 2\sin(\alpha_{limit}/2) = 0.38504$ and $MB/BC < 1.32$, and the result follows.

(b) According to Proposition 6, for obtuse Delaunay terminal triangles with $\alpha \leq \tilde{\alpha}$, the worst $d(M)/BC$ ratio is obtained for the obtuse triangle of angles $2\pi/3, \tilde{\alpha}, \pi/3-\tilde{\alpha}$. For $\alpha \leq 30°$ we need to consider the simple obtuse isosceles triangles of angles 120°, 30°, 30°, for which $CM = CB/2$, and b(iii) follows. Analogous reasoning and computations gives cases b(ii) and the result of b(iii).

As shown in [25], the following theorem based on Proposition 3 and Lemmas 7 and 8 ensures that bad quality terminal triangles are quickly improved by introducing a sequence of better triangles of edge $CB$ (see Fig. 3), but not introducing points too close to the previous vertices in the mesh. Furthermore, here we also prove that, for triangle smallest angle less than 14°, no edge smaller than the existing neighbor edges is introduced in the mesh.

**Theorem 9.** *Consider the Lepp-midpoint method and* $\theta_{tol} \leq \alpha_{limit}$. *Then for any bad quality terminal triangle of smallest angle* $\alpha$, *at most a finite sequence of improved triangles* $\{t_{B_j}\}$ *(Fig. 3), can be obtained until* $t_B$ *is good (if edge swapping does not occur), such that* $\beta_j \geq (3/2)^j\alpha_0$. *Furthermore* (a) *For* $\alpha \leq 14°$, *no 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 occasionally introduced in the mesh for obtuse triangle* $t$.

### *3.2. An explanation on reported convergence*

In practice, the Lepp-midpoint method shows slower convergence for $\theta_{tol}$ greater than 25°, than for smaller $\theta_{tol}$ values [6]. This can be explained by using the results of this section as follows. Firstly note that the worst $d(M)/BC$ ratio is obtained for obtuse Delaunay terminal triangles $t_1$ of largest angle close to $2\pi/3$ and smallest angles close to 30°. Note also that for this $t_1$, its associated terminal triangle $t_2$ is close to an equilateral triangle. Consequently, the longest edge bisection of both $t_1, t_2$, not only does not improve significantly the smallest angles, but introduces four triangles whose smallest angles remain around 30°. Even furthermore, according to part (I) (a) of Lemma 7, the longest edge bisection of either $t_1, t_2$ can produce an acute son $t^*$ of smallest angle close to 30°, whose longest edge bisection (if this becomes a terminal triangle) can in turn produce a worst son of smallest angle approximately equal to 23.7°. Note that $t^*$ cannot become a terminal triangle and can need of more point insertions for its elimination.

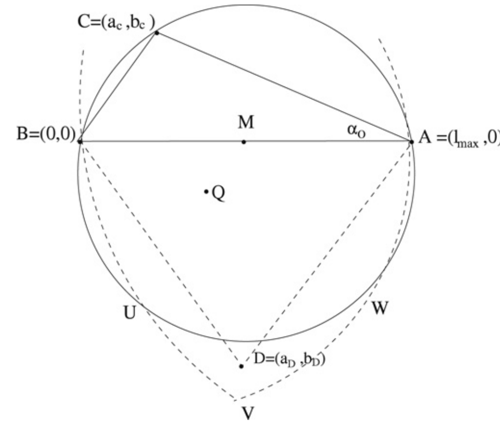


**Fig. 8.** Estimating location of centroid Q for Delaunay terminal triangles BAC, BDA, where BAC is an obtuse triangle.

## 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 discussed at the end of last section, and reported in [6] for $\theta_{tol} > 25°$. It would be also desirable that for a bad triangle $t_1$ belonging to a couple $t_1, t_2$ of Delaunay terminal triangles, the point selection strategy be a function of both $t_1, t_2$, not only of $t_1$ as happens with Lepp-midpoint algorithm. Since $t_1, t_2$ form a convex (terminal) quadrilateral we choose to insert the centroid which is always in the interior of the terminal quad. Note that this strategy is related with the smart Laplacian smoothing but without requiring an expensive optimization step [26–29].

Thus, 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 $BDA$ (Fig. 8) sharing an unconstrained terminal edge. The algorithm is given below:

**Lepp–Terminal-Centroid Algorithm**
Input = a CDT, $\tau$, and angle tolerance $\theta_{tol}$
Find $S_{bad}$ = the set of bad triangles with respect to $\theta_{tol}$
**for** each t in $S_{bad}$ **do**
  **while** t remains in $\tau$ **do**
    Find Lepp ($t_{bad}$), terminal triangles $t_1, t_2$ and terminal edge $l$.
    Triangle $t_2$ can be null for boundary $l$.
    New Select Point (P, $t_1, t_2, l$)
    Perform constrained Delaunay insertion of P into $\tau$
    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**
  **if** (second longest edge of $t_{term1}$ is constrained and second longest edge of $t_{term2}$ is constrained) **then**
    Select $P$ = midpoint of a constrained second longest edge
  **else**
    **for** $t^*$ in { $t_{term1}, t_{term2}$ } with constrained second longest edge $l^*$ **do**
      Select ConstrainedQuad ($P, t_{term1}, t_{term2}, l_{term}, l^*$)
    **end for**
  **end if**
**end if**

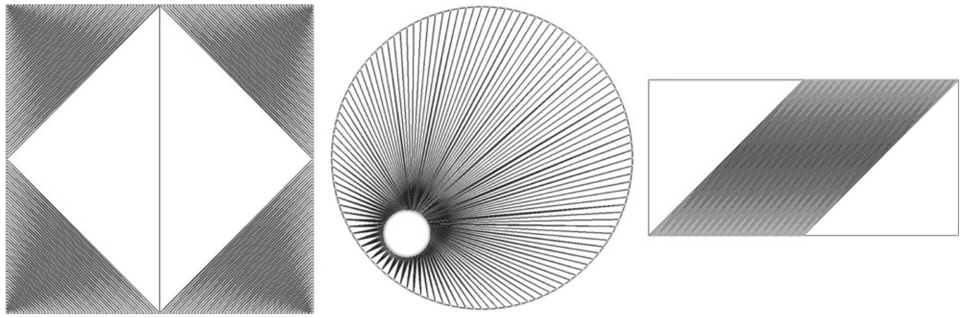

**Fig. 9.** Initial triangulations for the test problems.

**Remark 10.** For the examples presented in this paper we have used a simple Select ConstrainedQuad function. For pairs of Delaunay terminal triangles where only one of them has constrained second longest edge, we proceed as follows: if the centroid is inside the non-constrained triangle, the centroid is selected for point insertion; otherwise the midpoint of the second longest edge is selected. As discussed in Section 5, with this simple criterion, the centroid method works significantly better than the off-center algorithm for geometries such that the Delaunay triangulation of the input data has triangles with smallest edges over the boundary. In exchange, for geometries having a rougher and irregular boundary point distribution the centroid introduces more points close to the boundaries as compared with the off-center method. It is worth noting that the set of problems whose geometry has a fine boundary point distribution is an important class of problems for finite element applications, where usually the first mesh generation step consists on introducing Steiner points over the boundary. Anyway, in future research we expect to tune the algorithm also for rough boundary point distribution problems.

### *4.1. Geometrical properties of the centroid selection*

Consider a terminal quadrilateral *ACBD* formed by the union of a pair of terminal triangles *ABC*, *ABD* sharing a non-constrained terminal edge *AB* (Fig. 8). In what follows we will prove that inserting the terminal centroid $Q$, defined as the centroid of the terminal quadrilateral *ACBD*, tends to produce better new triangles $t_B$ and $t_A$ (Fig. 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 Fig. 8. This is the longest edge coordinate system introduced by Simpson [30].

Firstly consider the case of obtuse Delaunay terminal triangles. 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 radius *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 $D(a_D, b_D)$.

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 11.** *Let ABC be any obtuse Delaunay terminal triangle. Let the centroid Q have longest edge coordinates* $(a_Q, b_Q)$ *with respect to triangle ABC. Then*

$$a_{min}/4 + l_{max}/4 < a_Q < a_{max}/4 + l_{max}/2, \quad \textit{and}$$
$$b_D/4 < b_Q < (b_c + b_{min})/4.$$

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

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

**Corollary 12.** *For any terminal quad, the quad centroid is situated in the interior of the best smallest angled triangle.*

**Remark 13.** Let $t_a$, $t_b$ be terminal triangles forming a terminal quadrilateral, and sharing a terminal edge, $E$. (1) If the worst quality triangle is obtuse, then the shortest edge that results from the insertion of the terminal centroid into the mesh is in general longer than the shortest edge that results from inserting the midpoint of $E$. In fact for obtuse triangle of angles $2\pi/3, \pi/6, \pi/6$, (and equilateral terminal edge neighbor), the ratio $d(Q)/BC = 0.75$. This is the worst case for the midpoint method. (2) In exchange, if the worst quality triangle is non-obtuse, then the worst $d(Q)/BC$ ratio holds now for a pair of bad quality right triangles of smallest angle $\pi/3$, where one is the mirror image of the other one. For this case also $d(Q)/BC = 0.75$. Note however that this is a less probable case, since for other locations of vertex $D$, the $d(Q)/BC$ ratio is better.

**Theorem 14.** *The algorithm does not suffer from 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 15.** The constraint on $\theta_{tol}$ is not necessary and the analysis of the algorithm can be extended until $\theta_{tol} = 30°$.

## 5. Empirical study and concluding remarks

Note that in the previous version of this paper we had not realized that the Lepp-centroid and the off-center algorithms behave differently for different classes of geometries. In what follows we present the empirical study reported in [17] considering this new evidence.

### *5.1. Triangulations with boundary smallest edges*

We consider the 3 test problems of Fig. 9 whose (bad quality) initial triangulations are shown in this figure. They correspond to a square with 400 equidistributed boundary points (Square400), a discretized circle having a discretized circular hole close to the

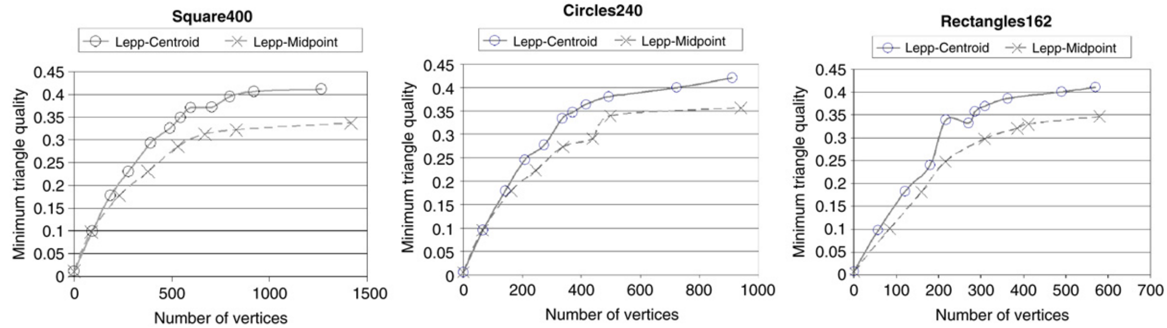


**Fig. 10.** Evolution of the minimum area quality measure as a function of the number of vertices for Lepp-centroid and Lepp-midpoint algorithms.

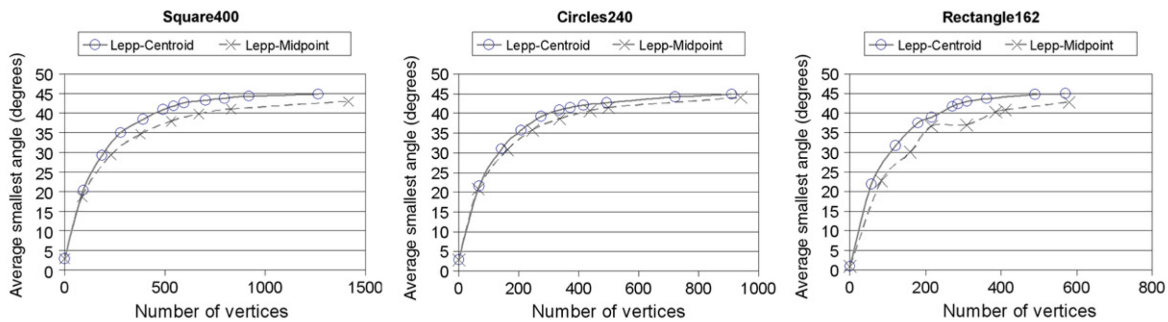


**Fig. 11.** Evolution of the average smallest angle (degrees) as a function of the number of vertices for Lepp-centroid and Lepp-midpoint algorithms.

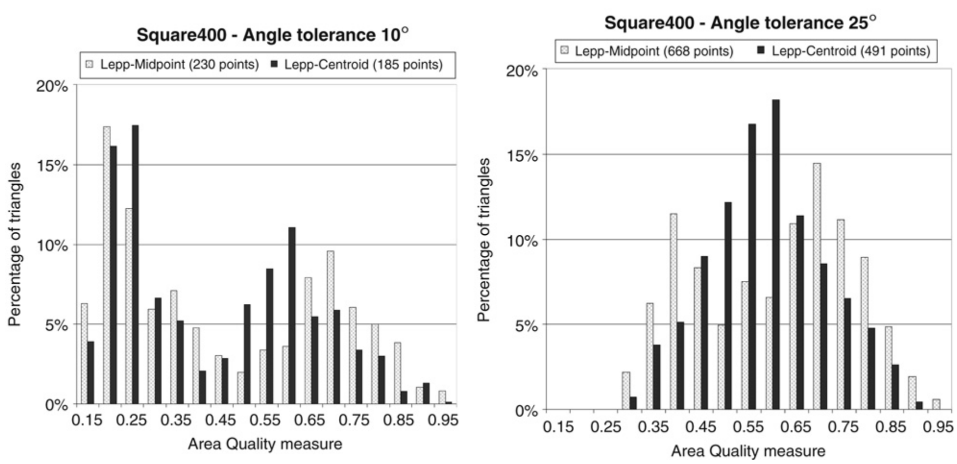


**Fig. 12.** Area quality distribution for Square 400 for $\theta_{tol} = 10°$ (midpoint 230 points, centroid 185 points) and $\theta_{tol} = 25°$ (midpoint 668 points, centroid 491 points).

boundary with 240 boundary points (Circle 240) and a rectangle having 162 boundary points distributed as shown in Fig. 9 (Rectangle 162). The initial triangulations are Delaunay excepting the triangulation of the Rectangle 162 case, which is an example proposed by Edelsbrunner [16] for the point triangulation problem. Note that for these problems, the initial Delaunay triangulation has bad triangles with boundary smallest edges.

### *5.2. Centroid versus midpoint for triangulations with boundary smallest edges*

We used the test cases of Fig. 9 to compare the algorithms. We ran every case for different $\theta_{tol}$ values, with the input meshes of Fig. 9, until reaching the maximum practical tolerance angle $\theta_{tol}$. 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 $\theta_{tol}$. The Lepp centroid works for $\theta_{tol}$ up to 36° while the previous midpoint method works for $\theta_{tol} \leq 30°$ for these examples. It 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 $\theta_{tol}$ value. To illustrate this see Figs. 10 and 11 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 Fig. 12

**Table 1**

| Number of points added | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Minimum mesh angle (°) | 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 | – |

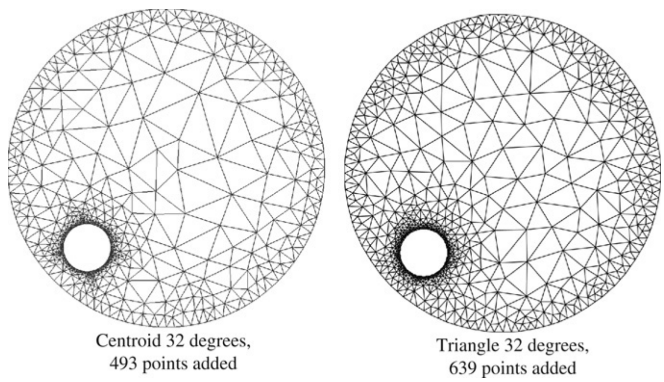


**Fig. 13.** Triangulations obtained with Lepp-centroid and Triangle methods, $\theta_{tol} = 32°$.

which shows the evolution of the area quality distribution for the Square 400 test problem for $\theta_{tol}$ equal to 10°, 25°.

The normalized minimum edge size in the mesh (wrt the smallest edge in the initial mesh) behaves accordingly to the theoretical results of Section 3 (see [17]) even for the Lepp-midpoint method. For all these problems the average Lepp size remains between 3 and 5, this value being 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 Fig. 13.

## 6. A comparison with Triangle for triangulations with boundary smallest edges

As reported previously in [6] the Lepp-midpoint method showed a behavior analogous to the circumcenter algorithm implemented in a previous Triangle version [9]. Here we use the current Triangle version based on the off-center point selection [13,14], to perform a comparison with the Lepp-centroid method for the test problems of Fig. 9 and the *A* test case having 21 boundary points provided at the Triangle site. The evolution of the minimum angle in the mesh as a function of the number of vertices is shown in Fig. 14 for the same set of $\theta_{tol}$ 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 $\theta_{tol}$ up to 36°, while that the Triangle method worked for $\theta_{tol}$ up to 35° but increasing highly the number of points inserted for $\theta_{tol}$ bigger than 25°. Only for the small A-test-case, where basically boundary points are inserted, both algorithms have more similar behavior. A quantitative view of the behavior of both algorithms is given in the Table 1 which shows the number of points added

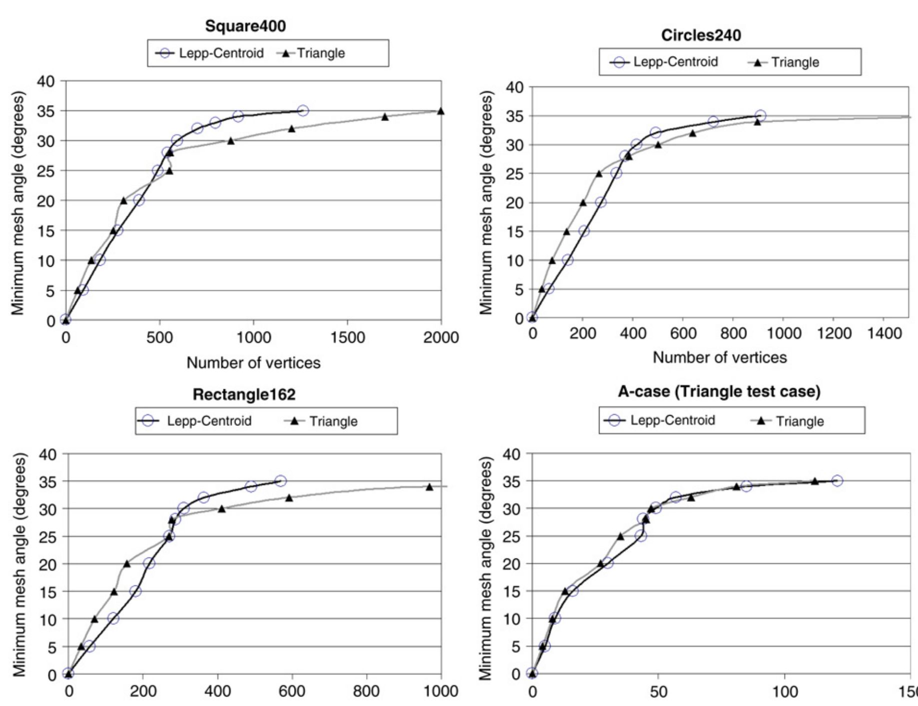


**Fig. 14.** Evolution of the minimum angle in the mesh as a function of the number of vertices for Lepp-centroid and Triangle methods.

to the mesh for all the test problems and different values of $\theta_{tol}$. The triangulations obtained for the Circles240 case with the Lepp-centroid and Triangle methods for $\theta_{tol} = 32°$ are shown in Fig. 13.

### 6.1. 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 (Fig. 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 from special looping conditions and that smallest edges bigger than those introduced by the midpoint method are introduced. 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 $\theta_{tol} > 25°$ for initial triangulations having boundary smallest edges which is an important class of problems for finite element applications. We have also obtained empirical evidence that show that the off-center method behaves better than the centroid method for rough irregular boundary polygons. We believe that to correct this behavior we need to tune the Select ConstrainedQuad function.

Recently Erten and Üngör [14] have introduced algorithms that improve the off-center performance with respect to the mesh size and the angle $\theta_{tol}$ by using point selections depending on some triangle cases. We plan to improve the Lepp based algorithms also in this direction.

## Acknowledgements

This research was supported by DI ENL 07/03. We are grateful to Bruce Simpson who contributed to an early formulation of this paper.

## References

[1] Borouchaki H, George PL. Aspects of 2-d delaunay mesh generation. International Journal for Numerical Methods in Engineering 1997;40:1957–75.
[2] George PL, Borouchaki H. Delaunay triangulation and meshing. Hermes 1998.
[3] Rivara MC. New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations. In: International meshing roundtable. 1996. p. 77–86.
[4] Rivara MC. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. International Journal for Numerical Methods in Engineering 1997;40:3313–24.
[5] Rivara MC, Hitschfeld N, Simpson RB. Terminal edges Delaunay (small angle based) algorithm for the quality triangulation problem. Computer-Aided Design 2001;33:263–77.
[6] Rivara MC, Palma M. New lepp algorithms for quality polygon and volume triangulation: Implementation issues and practical behavior. In: SA Cannan, Saigal AMD, editors. Trends unstructured mesh generation. vol. 220. 1997. p. 1–8.
[7] Chew LP. Guaranteed-quality triangular meshes. In: Technical report TR-98-983. Computer Science Department. Cornell University; 1989.
[8] Ruppert J. A delaunay refinement algorithm for quality 2-dimensional mesh generation. Journal of Algorithms 1995;18:548–85.
[9] Shewchuk JR. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In: First workshop on applied computational geometry. ACM; 1996. p. 124–33.
[10] Weatherill NP, Hassan O. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. International Journal for Numerical Methods in Engineering 1994;2005–39.
[11] Baker TJ. Automatic mesh generation for complex three dimensional regions using a constrained delaunay triangulation. Engineering with Computers 1989;5:161–75.
[12] Baker TJ. Triangulations, mesh generation and point placement strategies. In: Caughey D, editor. Computing the future. John Wiley; 1995. p. 61–75.
[13] Üngör A. Off-centers: A new type of steiner points for computing size-optimal quality-guaranteed delaunay triangulations. In: Mathematics of computation. LNCS, vol. 2076. 2004. p. 152–61.
[14] Erten H, Üngör A. Triangulations with locally optimal steiner points. In: Belyaev A, Garland M, editors. Eurographics symposium on geometry processing. 2007.
[15] Aurenhammer F, Katoh N, Kojima H, Ohsaki M, Xu Y. Approximating uniform triangular meshes in polygons. Theoretical Computer Science 2002; 879–895.
[16] Edelsbrunner H. Geometry and topology for mesh generation. In: Cambridge monographs on applied and computational mathematics. Cambridge Univ. Press; 2001.
[17] Rivara MC, Calderon C. Lepp terminal centroid method for quality triangulation: A study on a new algorithm. In: Chen F, Jütler B, editors. GMP 2008. LNCS, vol. 4975. 2008. p. 215–30.
[18] Rivara MC. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. International Journal for Numerical Methods in Engineering 1984;20:745–56.
[19] Rivara MC. Selective refinement/derefinement algorithms for sequences of nested triangulations. International Journal for Numerical Methods in Engineering 1989;28:2889–906.
[20] Rivara MC, Levin C. A 3d refinement algorithm for adaptive and multigrid techniques. Communications in Applied Numerical Methods 1992;8:281–90.
[21] Rosenberg IG, Stenger F. A lower bound on the angles of triangles constructed by bisecting the longest side. Mathematics of Computation 1975;29:390–5.
[22] Stynes M. On faster convergence of the bisection method for certain triangles. Mathematics of Computation 1979;33:1195–202.
[23] Stynes M. On faster convergence of the bisection method for all triangles. Mathematics of Computation 1980;35:1195–202.
[24] Simpson B, Rivara MC. Geometrical mesh improvement properties of delaunay terminal edge refinement. In: Geometric modeling and processing. LNCS, vol. 4077. 2006. p. 536–44.
[25] Rivara MC. A study on delaunay terminal edge method. In: Pébay PP, editor. Proceedings of the 15th international roundtable. Springer; 2006. p. 543–62.
[26] Canann S, Stephenson M, Blacker T. Optismoothing: An optimization-driven approach to mesh smoothing. Finite Elements in Analysis and Design 1993; 13:185–90.
[27] Freitag LA. On combining laplacian and optimization-based mesh smoothing techniques. 1997.
[28] Freitag LA, Ollivier-Gooch C. Tetrahedral mesh improvement using swapping and smoothing. International Journal for Numerical Methods in Engineering 1997;40:3979–4002.
[29] Freitag LA, Plassmann P. Local optimization-based simplicial mesh untangling and improvement. International Journal for Numerical Methods in Engineering 2000;49:109–25.
[30] Simpson RB. Geometry independence for a meshing engine for 2d manifolds. International Journal for Numerical Methods in Engineering 2004;675–94.