Computer-Aided Design 125 (2020) 102870

Contents lists available at ScienceDirect

Computer-Aided Design

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

Terminal Triangles Centroid Algorithms for Quality Delaunay Triangulation

Maria-Cecilia Rivara *, Javier Diaz

University of Chile, Department of Computer Science, Avenida Beauchef 851, Santiago 8370456, Chile

A R T I C L E I N F O

Article history:
Received 25 May 2019
Received in revised form 21 December 2019
Accepted 23 April 2020

Keywords:
Lepp algorithms
Lepp Delaunay centroid algorithms
Quality Delaunay triangulation
Terminal triangles centroid
Delaunay terminal triangles
Delaunay terminal edges

A B S T R A C T

Two Lepp algorithms for quality Delaunay triangulation are discussed. Firstly a terminal triangles centroid Delaunay algorithm is studied. For each bad quality triangle *t*, the algorithm uses the longest edge propagating path (Lepp(t)) to find a couple of Delaunay terminal triangles (with largest angles less than or equal to 120 degrees) sharing a common longest (terminal) edge. Then the centroid of the terminal quadrilateral is Delaunay inserted in the mesh. Insertion of the midpoints of some constrained edges are also performed to assure convergence close to the constrained edges. We prove algorithm termination and that a graded, optimal size, 30 degrees triangulation is obtained, for any planar straight line graph (PSLG) geometry with constrained angles greater than or equal to 30 degrees. We also prove that the size of the final triangulation is optimal and that this size is independent of the processing order of the bad triangles in the mesh. Next, by introducing the concept of non-improvable triangles (with constrained angle $<$ 30 degrees), we generalize the algorithm to deal with PSLG geometries with N small constrained angles. Thus given a triangle size parameter δ for non-improvable triangles, the generalized algorithm constructs a quality triangulation with non constrained angles $\geq$ 30 degrees and at most N non-improvable triangles of size δ (longest edge $\leq \delta$). In practice the algorithms behave as predicted by the theory.

1. Introduction

Algorithms based on the longest edge bisection of triangles were proposed and studied in [1,2] for adaptive and multigrid finite element methods. Lepp bisection algorithm [3,4] is an efficient reformulation of previous longest edge algorithm for triangulation refinement, that for each target triangle follows the longest edge propagating path (Lepp) to find a couple of terminal triangles sharing a common longest edge (terminal edge), which are then refined by longest edge bisection. Consequently, local refinement operations are used, and conforming triangulations (where adjacent triangles either share a common edge or a common vertex) are maintained throughout the whole refinement process. Due to the properties of the iterative longest edge bisection of triangles, refined triangulations that maintain the triangulation quality (bounded smallest angle) are obtained, while the proportion of quality triangles increases as the refinement proceeds. Based on the properties of terminal triangles and terminal edges it was also proved that optimal size triangulations are obtained [4].

A Lepp Delaunay algorithm for quality Delaunay triangulation, based on the Delaunay insertion of the midpoint of the terminal edge, was introduced by Rivara [3] and further studied by Bedregal and Rivara [5]. An algorithm based on computing the centroid Q of the terminal triangles which is Delaunay inserted, was presented in [6] without proving termination, neither optimal size property. In this paper we study a tuned, order independent algorithm (where the size of the refined triangulation is almost equal independently of the triangle processing order), based on the Lepp centroid algorithm discussed in [6].

Alternative Delaunay refinement algorithms, based on selecting the circumcenter (or a point close to the circumcenter) of each skinny triangle, which is Delaunay inserted in the triangulation have been studied by Ruppert [7], Shewchuk [8], and by Erten an Üngor [9]. Lepp Delaunay algorithms and circumcenter based algorithms have analogous practical behavior, as shown in the empirical study of Ref. [6], where the Triangle software [8] (without later improvement criteria) was compared with Lepp Delaunay algorithms. It is worth noting however that Lepp based algorithms have the advantage of being order independent, in the sense that they construct triangulations of approximately the same size independently of the processing order of the bad quality triangles. Consequently they are simpler methods than circumcenter based algorithms, easy to implement and easy to parallelize. On the other hand, the implementation

* Corresponding author.
E-mail addresses: mcrivara@dcc.uchile.cl (M.-C. Rivara), jdiaz@dcc.uchile.cl (J. Diaz).

https://doi.org/10.1016/j.cad.2020.102870
0010-4485/

of circumcenter algorithms is rather cumbersome, and requires processing triangles in bad-quality order. Section 6.3 of Ref. [10] discusses several recommendations to implement Ruppert's algorithm efficiently, which include maintaining a queue of skinny and oversized triangles throughout the refinement process.

Lepp algorithms. These are longest edge algorithms formulated in terms of the concepts of terminal edges, terminal triangles and the longest edge propagating path [3–6]. An edge E is a *terminal edge* in triangulation τ if E is the longest edge of every triangle that shares E. The triangles sharing E are called *terminal triangles* (edge AB in Fig. 1(a)). If E is shared by two terminal triangles then E is an interior edge; if E is shared by a single terminal triangle then E is a boundary edge.

For any triangle t_0 in τ, the *longest edge propagating path* of t_0, $\mathtt{Lepp}(t_0)$, is the ordered sequence of increasing triangles $\{t_j\}_0^{N+1}$ such that t_j is the neighbor triangle on the longest edge of t_{j-1} and where longest_edge $t_j >$ longest_edge t_{j-1}, for $j = 1, \ldots, N$. The process ends by finding the terminal edge E and a couple of associated terminal triangles t_N, t_{N+1}. In Fig. 1(a), Lepp$(t_0) = \{t_0, t_1, t_2, t_3\}$.

For each target triangle t, the generic Lepp based algorithms find an associated local largest edge shared by a couple of terminal triangles. Then a point is selected inside the terminal triangles (terminal edge midpoint or terminal triangles centroid) and inserted in the mesh. In the Lepp bisection algorithm, the midpoint M of the terminal edge is inserted by longest edge bisection of the terminal triangles as shown in Fig. 1(b). The process is repeated until the target triangle t is destroyed. The generic algorithm is as follows:

Algorithm Generic Lepp-based algorithm
Input : triangulation τ, set S of triangles to be refined/improved
Output : Refined triangulation τ'

for each t in S **do**
  while t remains in τ **do**
    Find $\mathtt{Lepp}(t)$, terminal triangles t_1, t_2 and terminal edge E (t_2 can be null)
    Select point P inside terminal triangles, insert P in the mesh and update S
  end while
end for

In the Lepp centroid Delaunay algorithm of this paper, for each processing bad triangle t, if Lepp(t) finds a couple of non constrained Delaunay terminal triangles, the centroid Q of the quadrilateral formed by the terminal triangles is computed, and Delaunay inserted in the mesh. For constrained edges two additional operations are used. The algorithm is presented in Section 2 together with a sketch of the analysis performed over the most frequent operation, the Delaunay insertion of centroid Q.

This paper discusses a simple algorithm in 2-dimensions for the construction of quality triangulations suitable for complex practical applications such as those related with finite element methods, as well as a set of robust improvement operations that can be easily added to any meshing software to improve meshes. The following summarizes the contributions (new and improved results) of this paper.

- We prove that the algorithm produces 30° quality triangulations for any planar straight line graph (PSLG) geometry with constrained angles greater than or equal to 30° by using the improvement properties of the operations OP1, OP2, OP3 described in the next section, and the fact that the average Lepp size tends to be 2. This is a strong new result with respect to previous algorithms. Note that the proof in Ruppert's algorithm requires constrained angles $\geq 90°$, while the modified algorithm of Shewchuk requires constrained angles $\geq 60°$.

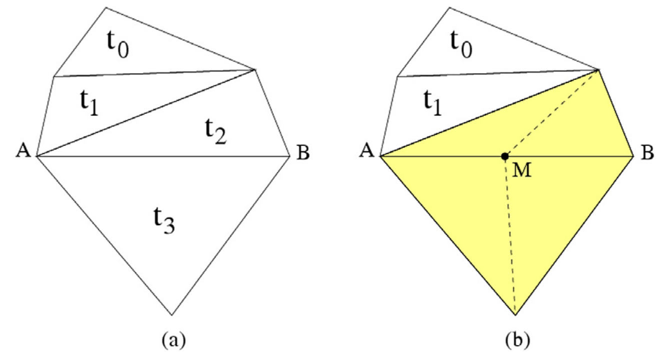

Fig. 1. (a) AB is a terminal edge shared by terminal triangles $\{t_2, t_3\}$ and $\mathtt{Lepp}(t_0) = \{t_0, t_1, t_2, t_3\}$; (b) First step of Lepp-bisection algorithm for refining t_0.

- The constrained Delaunay triangulation (CDT) of the PSLG data defines an intuitive edge distribution function, which identifies edge details and non-edge details in the PSLG geometry (smallest edges in the CDT). We prove that the algorithm constructs a graded quality triangulation around the geometry details. This also allows us to prove termination and optimal size property. Note that we use the edge distribution function instead of using the local feature size function introduced by Ruppert [7].
- To deal with constrained edges in the PSLG geometry, the algorithm does not require the edge encroachment test used in Ruppert's algorithm, but a simple test based on triangle constrained edges.
- We prove that the size of the output (quality) triangulation is independent of the processing order of the bad triangles in the mesh, which is not the case of circumcircle based algorithms.
- Due to the properties of Delaunay terminal triangles, the bad obtuse triangles with largest angle $> 120°$ cannot belong to couples of Delaunay terminal triangles, and are eliminated by edge swapping, assuming that an edge swapping Delaunay algorithm is used.
- The average Lepp size is small and tends to be 2 as the refinement proceeds. This result was proved for triangulations obtained by the Lepp bisection algorithm [5] and extends to the algorithms of this paper. This contributes to prove the optimal size property.
- Finally we generalize the tuned algorithm for PSLG geometries with non-improvable angles (constrained angles < 30 degrees), an extension of the algorithm suitable for applications involving material changes.

This paper is organized as follows. In Section 2 we present the algorithm based on three simple mesh operations, as well as a sketch on the algorithm analysis based on studying the improvement properties of three intermediate mesh operations, not used in the implementation, but required in the analysis. In Section 3 we review results on the iterative longest edge bisection of triangles and on the properties of Lepp algorithms. In Section 4 we state improvement properties on the longest edge bisection of triangles with largest angles $\leq 120°$ (the only triangles involved in the mesh operations). In Section 5 we discuss improvement properties of the operations performed over triangles with constrained edges. In Section 6 we present improvement results for the simple centroid insertion over Delaunay terminal triangles. In Section 7 we analyze the algorithm integrating the results of the previous sections. In Section 8 we present empirical results in agreement with the theory. In Section 9 we generalize the algorithm for geometries with constrained angles $< 30°$.

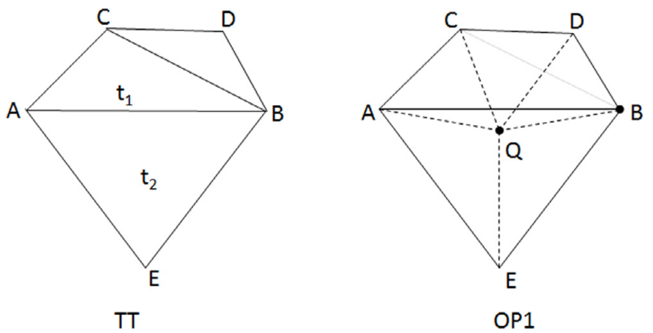

Fig. 2. Operation OP1 applied over non constrained terminal triangles (t_1, t_2) in triangulation TT.

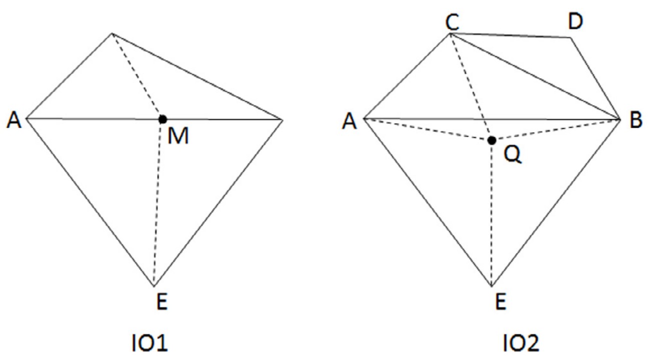

Fig. 3. Intermediate operations used in the analysis of Operation OP1. Operation IO1 performs longest edge bisections; Operation IO2 performs simple centroid insertion.

2. The algorithm and sketch of the analysis

The (tuned) Lepp Delaunay centroid algorithm uses the following three simple mesh operations:

- OP1 Over a couple (t_1, t_2) of non constrained Delaunay terminal triangles (see Fig. 2 TT), the centroid Q of the quadrilateral formed by (t_1, t_2) is computed and Delaunay inserted in the mesh as shown in Fig. 2 OP1.
- OP2 Over a couple of Delaunay terminal triangles where one of the triangles is bad and has a constrained second longest edge E, the midpoint of E is selected and constrained Delaunay inserted in the mesh.
- OP3 For constrained terminal edge, the longest edge bisection of the terminal triangles (or triangle) is performed.

The operation OP2 was added to the algorithm discussed in Ref. [6], and contributes to prove the stronger results on the algorithm. A preliminary version of this paper was presented in the 27th International Meshing Roundtable [11].

2.1. The algorithm for geometries with constrained angles $\geq 30°$

Assuming an input constrained Delaunay triangulation (CDT) of a PSLG data with constrained angles $\geq 30°$; and angle tolerance $\theta_{tol} \leq 30°$, the algorithm is as follows.

Algorithm Tuned Terminal_Triangles_Centroid_Delaunay _Algorithm
Input: CDT τ associated with PSLG data, angle tolerance θ_{tol}
Output: Refined triangulation τ_f with angles $\geq \theta_{tol}$.

Find S set of bad quality triangles
for each t in S (while $S \neq \varnothing$) **do**
  while t remains unrefined **do**
    Use Lepp(t) to find Delaunay terminal triangles t_1, t_2 and terminal edge E
    if E is constrained (this includes t_2 null) **then**
      Perform Constrained Delaunay insertion of midpoint of E
    else
      if there exists t (t_1 or t_2) such that $\alpha_t < \theta_{tol}$ and second longest edge L is constrained **then**
        Perform constrained Delaunay insertion of midpoint of L
      else
        Compute centroid Q of terminal triangles, and perform constrained Delaunay insertion of Q
      end if
    end if
    Update S
  end while
end for

2.2. Sketch of the algorithm analysis

In the analysis we assume that the Delaunay insertion of the centroid Q is performed by using an edge swapping Delaunay function as described by de Berg et al. [12] in section 9.3, page 192.

Let us consider the most frequent operation OP1 in the algorithm. Then assume that by processing a bad quality triangle t with smallest angle $\alpha_o < 30°$, the algorithm computes Lepp(t) finding a couple of non constrained terminal triangle (t_1, t_2) as shown in Fig. 2 TT, where t_1 is a bad triangle. Then operation OP1 applies and the centroid Q of the quadrilateral formed by the terminal triangles (t_1, t_2) is computed and Delaunay inserted in the mesh as shown in Fig. 2 OP1, which improves the involved triangles.

To study the improvement properties of the operation OP1 we need to consider three intermediate operations, not performed by the algorithm, but required in the analysis.

- IO1 The longest edge bisection of the triangles (t_1, t_2), which introduces midpoint M and produces the triangulation of Fig. 3 IO1.
- IO2 the simple insertion of centroid Q obtained by joining Q with the vertices of the quadrilateral AEBC Fig. 3 IO2, which indeed corresponds to the Laplacian smoothing over vertex M in triangulation of Fig. 3 IO1.
- IO3 Delaunization of the triangulation obtained by operation IO2 by applying the necessary edge swapping operations to produce a full Delaunay triangulation. This completes the operation OP1.

Important Remark. Each triangle belonging to a couple of Delaunay terminal triangles has largest angle $\leq 120°$. This implies that every bad triangle with largest angle $> 120°$ is eliminated by Delaunay swapping of its longest edge. In this sense, the Delaunay edge swapping operation plays an important role in the algorithm, by performing the implicit elimination of too obtuse bad triangles in the mesh.

The algorithm analysis is roughly as follows. The operation IO1 produces both a triangle better than the terminal triangle and a bad obtuse triangle. The next operation IO2 improves the new triangles, especially the obtuse ones, while the Delaunay operation IO3 further improves the mesh by producing more equilateral triangles and eventually by eliminating (by edge swapping) the too obtuse triangles (with angle $> 120°$).

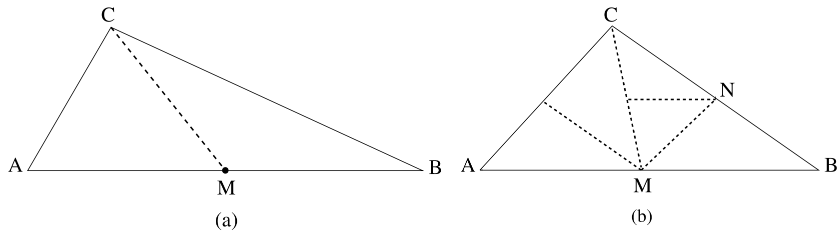

Fig. 4. (a) Longest-edge bisection of triangle $t(ABC)$ (b) First longest edge bisections that define a quasiequilateral triangle $t(ABC)$.

3. Previous results

The iterative longest edge bisection of individual triangles was studied by Rosenberg and Stenger [13] and by Stynes [14,15]. This process produces a finite number of non-similar triangles with bounded smallest angle, while the proportion of good triangles (quasiequilateral triangles) increases as the refinement proceeds. The following definitions are in order.

Definition 1. Given a triangle $t(ABC)$ of vertices A, B, C, and edges $AB \geq BC \geq CA$, the longest-edge bisection of t (or simply bisection of t) is performed by joining the midpoint M of AB with the opposite vertex C (see Fig. 4(a)).

Definition 2. Triangle $t(ABC)$ of edges $AB \geq BC \geq CA$ is *quasiequilateral* if $AC \geq max\{AB/2, CM\}$ and $MC \geq BC/2$ (see Fig. 4 (b)).

Note that for quasiequilateral triangles (see Fig. 4(b)) after the first median MC is introduced, the next longest edge bisections only produce medians parallel to the edges of the initial triangle ABC, which implies that at most, four similarly distinct triangles are produced. Furthermore the following results hold [13–15]:

A1. Given any triangle t_0 of smallest angle α_0, the iterative longest edge bisection of t_0 and its descendants produces a finite set $S(t_0)$ of similarly distinct triangles. Furthermore each triangle t in $S(t_0)$ has smallest angle α_t such that $\alpha_t \geq \alpha_0/2$.

A2. For any quasiequilateral triangle t_{qeq}, the triangle set $S(t_{qeq})$ has at most, four similarly distinct triangles, all of which are also quasiequilateral.

A3. For any non quasiequilateral triangle t_0, consider the sequence of triangle sets Q_j defined as follows: $Q_0 = \{t_0\}$, and for $j \geq 1$, Q_j is obtained by longest edge bisection of the triangles of Q_{j-1}. Then the triangle sets Q_j improve with j as follows: both the percentage of quasiequilateral triangles and the area of t_0 covered by these triangles, monotonically increase as the iterative refinement proceeds.

Lepp bisection algorithm. As discussed in Section 1, the Lepp bisection algorithm only performs longest edge bisections of couples of terminal triangles sharing a common longest (terminal) edge.

The triangulations obtained are conforming and inherit properties A1, A2, A3 as follows: the iterative local/global use of the Lepp bisection algorithm (and previous longest edge algorithms) produces sequences of nested, refined and conforming triangulations $\{\tau_j\}$ such that B1, B2 hold:

B1. For any triangle t_0 in τ_0, the refined triangles nested in t_0 belong to a finite set $S(t_0)$ of similarly distinct triangles, all of which have smallest angle $\alpha \geq \alpha_0/2$, where α_0 is the smallest angle of t_0.

B2. The refined triangulations $\{\tau_j\}$ improve with j in the following senses: both the percentage of quasiequilateral triangles, and the area covered by these triangles, increase as the refinement proceeds.

Later Bedregal and Rivara [4] proved that there exists a close relationship between quasiequilateral triangles and terminal triangles (the proportion of terminal triangles increases as quasiequilateral triangles increases), which imply B3. Furthermore, bounds on the number of triangle partitions performed inside a triangle in a Lepp sequence, summarized in assertion B4, were stated by Bedregal and Rivara [4]. Finally assertions B3 and B4 together imply B5.

B3. The proportion of terminal triangles increases (approaching 1) as the refinement proceeds and the average length of Lepp(t) tends to be 2 as the refinement proceeds.

B4. The number of longest edge bisections performed in the interior of a triangle t to make it conforming in a refining Lepp sequence, is constant and less than 3 in most cases. This constant is bounded by $O(\log^2(1/\alpha))$ for triangles with arbitrary smallest angle α.

B5. Lepp bisection algorithm produces optimal size triangulations.

Lepp Delaunay algorithms. These allow the construction of quality Delaunay triangulations of planar straight line graph (PSLG) geometries. Starting from the bad quality Delaunay triangulation of the PSLG input data, for each bad quality triangle t, the Lepp(t) is computed to find a couple of Delaunay terminal triangles over which a point is Delaunay inserted in the mesh. It is worth noting that Delaunay terminal triangles play a crucial role in Lepp Delaunay algorithms which have the properties summarized in the next theorems [3,16,17].

Theorem 1. *For any pair of Delaunay terminal triangles t_1, t_2 sharing a terminal edge AB it holds:*

(a) Largest angle $(t_i) \leq 2\pi/3$ for i=1,2
(b) At most one of the triangles t_1, t_2 is obtuse

Proof (*Part (a) Sketch*)**.** Couples of Delaunay terminal triangles ABC, ABD (see Fig. 5) are neighbor triangles that simultaneously satisfy that AB is the common longest edge of the both triangles, and that triangles ABC and ABD are locally Delaunay, which implies that vertex D is outside the circumcircle of triangle ABC. Both conditions together imply that vertex D must belong to the shadowed region $\mathcal{R}$ limited by the circumcircle of triangle ABC and the circles of vertices A, B and radius AB. In the case that $\measuredangle ACB = 120°$, $\mathcal{R}$ reduces to one point D' (triangle $AD'B$ is equilateral). Consequently for $\measuredangle ACB$ greater than 120°, $\mathcal{R}$ is empty and assertion (a) follows. □

The Lepp Delaunay algorithms inherit the properties B3 and B5 as stated in the following theorem. For a proof see references [4,5]

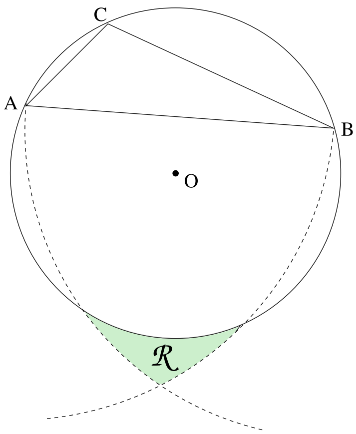

Fig. 5. Delaunay terminal triangles ABC, ABD; vertex D belongs to region $\Re$.

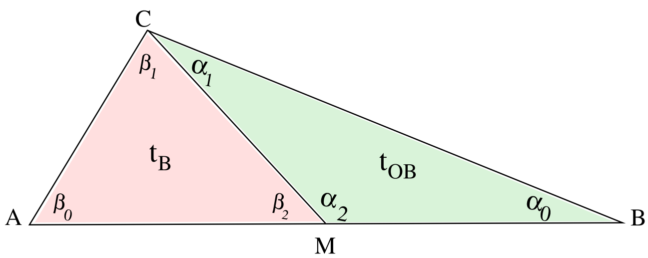

Fig. 6. Notation for longest edge bisection. Angles in longest edge bisection of triangle ABC with $AB \geq BC \geq AC$.

Theorem 2. *For the triangulations obtained by the Lepp Delaunay algorithms the following properties hold:*

(a) The proportion of terminal triangles increases (approaching 1) as the refinement proceeds and the average length of Lepp(t) tends to be 2 as the refinement proceeds.
(b) Lepp Delaunay algorithms produce optimal size triangulations (of size by a constant equal to the size of the smallest quality triangulation)

Remark. Since the algorithms of this paper insert points in the interior of couples of Delaunay terminal triangles, only triangles with largest angle less than or equal to 120° can become a terminal triangle throughout the algorithm processing. The following definition identifies the triangles that can become a terminal triangle throughout the algorithm processing.

Definition 3. We will say that t is a PD terminal triangle (potentially a Delaunay terminal triangle) if the $largestangle(t) \leq 120°$.

Corollary 1. *Non-PD terminal triangles are eliminated by swapping its longest edge in a delaunization process.*

At this point we need to emphasize the improvement property of the Delaunay edge swapping operation, stated by Sibson [18]. A proof can be found in Ref. [17].

Proposition 1. *Given any couple t_1, t_2 of non Delaunay neighbor triangles, then the swapping of the common edge produces a couple of locally Delaunay triangles such that the six new angles are better (by pairs) than the six angles of $t_1.t_2$.*

4. Angle bounds on the first bisections of triangles

Here we present revised results with respect to those of references [11]. We center the study on triangles with largest angle $\leq 120°$, and follow the notation of references [16,19].

4.1. Triangles taxonomy

Let us consider the first longest edge bisection of a triangle ABC where $AB \geq BC \geq AC$, which produces a better triangle t_B and an obtuse triangle t_{OB} (see Fig. 6). Considering this notation it is rather easy to prove the following lemma:

Lemma 1.

(a) if t is a right angled triangle then $\alpha_1 = \alpha_0$, $\beta_1 = \beta_0$, $AM = CM$
(b) if t is an acute triangle then $\alpha_1 < \alpha_0$, $\beta_1 < \beta_0$, $AM < CM$; and
(c) if t is an obtuse triangle then $\alpha_1 > \alpha_0$, $\beta_1 > \beta_0$, $AM > CM$.

These properties allow proving the assertions (b), (c), (d) of Lemma 2 [16]. The bound on α_1 follows from the strong property $A1$ of Section 3.

Lemma 2. *The following angle bounds hold [16].*

(a) $\alpha_1 \geq \alpha_0/2$, $\alpha_2 \geq 90°$, $\beta_2 \leq 90°$, $\beta_1 \geq \pi/6$, $\beta_1 \geq \alpha_1$
(b) $\beta_2 = \alpha_0 + \alpha_1 \geq 3\alpha_0/2$
(c) if t is obtuse, then $\alpha_1 > \alpha_0$ and $\beta_2 \geq 2\alpha_0$
(d) if t is acute, then $\alpha_1 < \alpha_0$ and t_B is acute

Next we introduce the taxonomy of Fig. 7, which is a variation of those discussed by Simpson and Rivara [16] and by Gutierrez et al. [19]. This is obtained by fixing the longest edge AB of triangle ABC considering $AB \geq BC \geq CA$, and studying which is the longest edge of triangle AMC and the longest edge of triangle CMN (see Fig. 6) according to the position of vertex C, which is situated in the region limited by lines AM, MX and arc AX in Fig. 7.

Note that the half circle of vertex M and radius AM separates obtuse and acute triangles. Arcs AR and MR respectively correspond to isosceles triangles with edges $AM = CM$ and edges $AC = AM$; while arc ZW corresponds to the circle of center $\tilde{N}$ (where $A\tilde{N} = AB/3$) and radius $A\tilde{N}$, corresponding to the triangles for which $CB = 2CM$.

The set of quasiequilateral triangles is the union of region $\mathcal{R}_1$ (acute triangles) and region $\mathcal{R}_2$ (obtuse triangles). By studying the boundaries of regions $\mathcal{R}_1$ and $\mathcal{R}_2$ it is easy to see that $\mathcal{R}_1 \bigcup \mathcal{R}_2$ correspond to quasiequilateral triangles and that most of these triangles (vertex C by above line SW) have the smallest angles $\geq 30°$. Only for vertex C in region SZW, the smallest angle α_0 is lightly $< 30°$. The worst case corresponds to $C = Z$ where $tg(\alpha_0(Z)) = \sqrt{7}/5 > 27.88°$). Note that most of the triangles of $\mathcal{R}_3$ also have the smallest angle $\geq 30°$.

Finally arc AW corresponds to points C for which the largest angle is equal to 120°, defined by the circle of center W' and radius WW', where points W, W' are symmetric with respect to line AB.

These results are summarized in the parts (a), (b), (c) of Lemma 3. Part (d) was proved in [16,19]

Lemma 3.

(a) Consider the taxonomy of Fig. 7. Then for quasiequilateral triangles in region TSWX, $\alpha_0 \geq 30°$;
(b) For quasiequilateral triangles in region SZW, $\alpha_0 > 27.88$.
(c) The non PD terminal triangles (with angle $>$ 120°) have vertex C in region AMW.
(d) Any triangle in region R_3 produces a quasiequilateral triangle t_B.

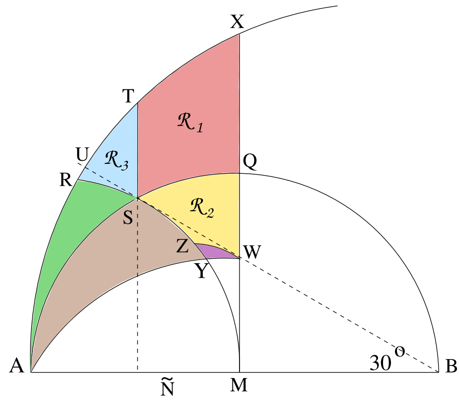

Fig. 7. Taxonomy on longest edge bisection of triangles $t(ABC)$ with $AB \geq BC \geq CA$.

4.2. Characterization of triangles t_B, t_{OB}

Here we study the triangles t_B, t_{OB} obtained by longest edge bisection of triangle t with angles $\leq 120°$. Remember that the angles α_0, α_1 are obtained by longest edge bisection of $t(ABC)$ as shown in Fig. 6. In addition we call $\alpha_0(t_B)$, $\alpha_0(t_{OB})$ to the smallest angles of t_B, t_{OB}.

Lemma 4. *Given a triangle t, then:*

(a) *If t is an acute triangle with $\alpha_0 \leq 30°$, then $\alpha_1 \geq 0.79\alpha_0$, $\beta_2 \geq 1.79\alpha_0$ and $\alpha_0(t_B) \geq 1.79\alpha_0$. Furthermore α_0 approaches α_1 as α_0 decreases.*
(b) *If t is an obtuse triangle with largest angle $\leq 120°$ and smallest angle $\alpha_0 \leq 30$, then $\beta_0 > 30°$, $\beta_1 > 90° - \alpha_0 > 60°$, $\beta_2 = 2\alpha_0$ and $\alpha_0(t_B) \geq Min\ \{30, 2\alpha_0\}$.*
(c) *If $\alpha_0 \geq 30°$, then $\alpha_0(t_B) \geq 30°$ and t_B is quasiequilateral.*
(d) *If t is quasiequilateral with $\alpha_0 \geq 30°$, then $\alpha_0(t_B) \geq 30°$ and $\alpha_0(t_{OB}) > 27.88°$.*
(e) *If $\alpha_0 > 16.8°$, then $\alpha_0(t_B) \geq 30°$.*

Proof. The proof of assertion (a) follows by studying the case of the acute triangles of region *UAS* in Fig. 7, where the worst case corresponds to point *U* for which $\alpha_1 \approx 23.79°$. For more details see Ref. [16,17]. Assertion (b) follows from Lemma 2. Parts (c), (d), follow from Lemma 3. Assertion (e) follows from part (a) of this lemma. □

The following Corollary and Lemma summarizes improvement results for the longest edge bisection of PD-terminal triangles (angles $\leq 120°$), which are the only triangles involved in the operations OP1, OP2 applied over couples of Delaunay terminal triangles in the algorithm. The non PD terminal triangles correspond to the obtuse triangles of region *WAM* in Fig. 7, with largest angle $> 120°$.

Corollary 2. *Given a PD-Terminal triangle t, then*

(a) *t_B is a PD-terminal triangle*
(b) *If $\alpha_0 \leq 30°$, the t_B is such that $\alpha_0(t_B) \geq Min\{30°, B\alpha(t_0)\}$, where either B = 1.79 if t is acute, or B = 2 if t is obtuse*
(c) *If t is quasi equilateral then t_B, t_{OB} are quasiequilateral*
(d) *For triangles with $\alpha \geq 30°$ (vertex C by above edge UB in Fig. 8), t_B is quasiequilateral*

In Lemma 5 we further characterize t_{OB} triangles associated to PD-terminal triangles.

Lemma 5. *Given any PD-terminal triangle t, then*

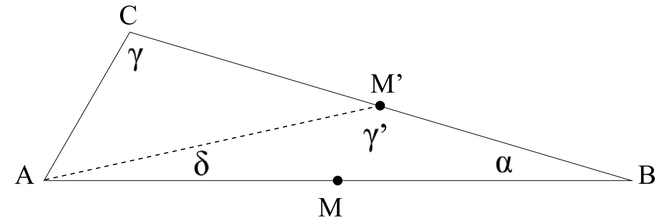

Fig. 8. For constrained second longest edge *CB* of a bad quality terminal triangle, the midpoint M' of *CB* is constrained Delaunay inserted in the mesh instead of M.

(a) *If t is acute and $\alpha_0 \leq 30°$, then t_{OB} is a non-PD terminal triangle.*
(b) *If t is obtuse and $\alpha_0 > 22°$, then t_{OB} can be a PD terminal triangle. If $\alpha < 22°$ the t_{OB} is a non-PD terminal triangle.*

Proof. Part (a) follows from the fact that for acute triangles $\alpha_1 < \alpha_0$, which in turn implies that $\beta_2 = \alpha_1 + \alpha_2 < 60°$ and consequently t_{OB} is a non-PD terminal triangle. Part (b) follows from the fact that for obtuse triangles, $\alpha_1 > \alpha_0$. In [17] it was proved that largest angle equal to 120° and $\alpha_1 + \alpha_0 = 60°$ implies that $\alpha_0 > 22°$. Thus, only for some triangles with $\alpha_0 > 22°$ it can hold $\alpha_1 + \alpha_0 > 60°$ and t_{OB} can be a PD terminal triangle. □

Remark. The longest edge bisection of a triangle close to the equilateral one, introduces two quasiequilateral triangles t_B and t_{OB}, such that $\alpha_0(t_B) \geq 30°$ and where t_{OB} can have a smallest angle $27.88 \leq \alpha_1 \leq 30°$.

5. Operations over triangles with constrained edges

5.1. Triangles with constrained second longest edge

Here we study the operation OP2. For bad quality terminal triangle ABC, where AB ≥ BC ≥ AC with a constrained second longest edge *CB*, the constrained Delaunay insertion of the midpoint M' of CB in Fig. 8 is performed,

For PD terminal triangles the following properties hold:

Theorem 3. *Let t (ABC) be any PD terminal triangle (largest angle $\leq 120°$) with smallest angle $< 30°$, and second longest edge CB of midpoint M' (Fig. 8). Then the bisection of t by the edge CB produces a non-PD terminal triangle $AM'B$.*

Proof. Firstly consider largest angle $120° \leq \gamma \leq 90°$. Then γ is also the largest angle of triangle CAM' which implies that $AM' > CM' = M'B$. This in turn implies that $\delta < \alpha < 30°$ and $\gamma' > 120°$.

Next consider an isosceles acute triangle with edges AB = CB and smallest angle α. In this case $\gamma = (180° - \alpha)/2$ which is also the largest angle of triangle CAM' and the result also follows. Finally for any acute triangle with $\alpha < 30°$, its largest angle $\gamma > (180 - \alpha)/2$ and using a reasoning analogous to the previous ones, the result follows. □

Corollary 3. *The operation OP2 always performs swapping of the longest edge AB to delaunize the mesh (see Fig. 9). This improves the mesh by destroying the bad (non-PD) triangle $AM'B$ by Delaunay swapping of edge AB.*

It is worth noting that the insertion of the midpoint of a constrained second longest edge CB avoids the insertion of points unnecessarily close to the constrained edges. Thus for very long

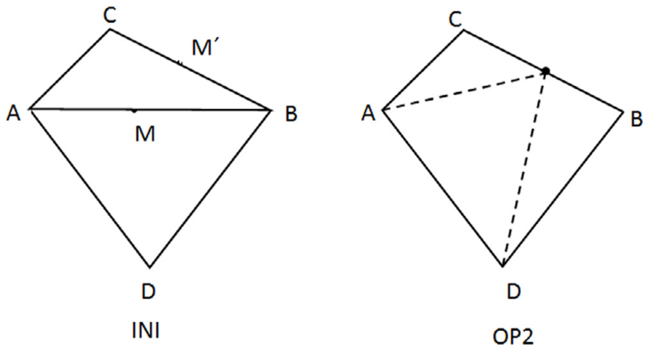

Fig. 9. Operation OP2 (Delaunay insertion of midpoint M') performs swapping of edge AB.

polygons (such as a for long quadrilaterals), the algorithm repeatedly introduces a midpoint of either a terminal edge or of a constrained second longest edge, as shown in Fig. 10.

Note that the previous Lepp Delaunay midpoint algorithm requires an analogous operation to guarantee convergence [3]. The previous Lepp Delaunay centroid algorithm of references [6] does not use this operation, but introduces more points than the tuned algorithm close to constrained edges.

5.2. Triangles with constrained smallest angle

The following Lemma assures that for triangle t with 30° constrained angle the algorithm produces a finite number of quality triangles in the interior of t when triangle t is reached throughout an interior Lepp.

Lemma 6. *Let t be any triangle with 30° constrained angle. Then (a) If t is obtuse, the tuned algorithm produces quality triangles in the interior of t; (b) If t is acute the tuned algorithm inserts three points in the constrained edges, as shown in Fig. 11, to produce quality triangles in the interior of t, excepting triangle EP_3D, which is improved using the simple centroid insertion operation inside Delaunay terminal triangle of edge ED, if this is a good triangle.*

Proof (*a*)**.** Note that, according to Lemma 1(a), the longest edge bisection of a 30° right angled triangle produces a good triangle with $\alpha_1 = \alpha_0 = 30°$ (see Fig. 6). Thus the refinement propagation will introduce one midpoint over the constrained longest edge and eventually one midpoint over the constrained second longest edge (if the propagation arrives by this edge). If t is obtuse then according to part (c) of Lemma 1, $\alpha_1 > \alpha_0 = 30°$, and analogously to the right triangle case, the result follows.

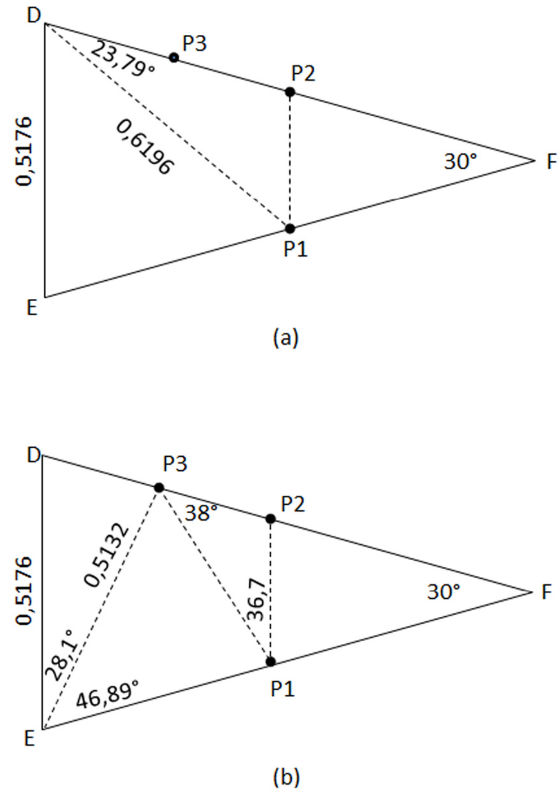

Fig. 11. Acute isosceles triangle with 30° constrained smallest angle (edges EF, DF are constrained).

(b) Consider the acute isosceles triangle DEF of Fig. 11 with constrained smallest angle 30° and longest edges DF, EF of length equal to 1. If triangle DEF becomes terminal, a vertex P_1 is constrained Delaunay inserted over constrained longest edge EF (or DF) producing triangles DEP_1 and DP_1F. Note that DE = 2 sin 15° ≈ 0.5176. Next, using the cosine theorem we find that $DP_1 \approx 0.6196$ and (smallest angle) $\angle FDP_1 \approx 23.79°$. Then by processing triangle DP_1F, the algorithm will perform constrained Delaunay insertion of point P_2 over constrained longest edge DF, which introduces new bad triangle DP_1P_2 whose processing will find terminal triangles DP_1P_2, DEP_1. Since triangle DP_1P_2 has constrained second longest edge DP_2, the point P_3 midpoint of DP_2 will be Delaunay inserted producing the triangles of Fig. 11(b). Using

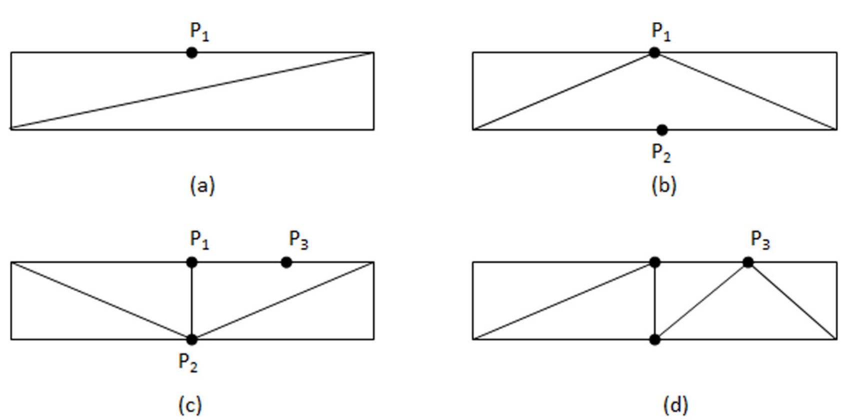

Fig. 10. For long quadrilateral both the insertion over a constrained second longest edge (operation OP2) and over a constrained longest edge (operation OP3) are repeatedly used.

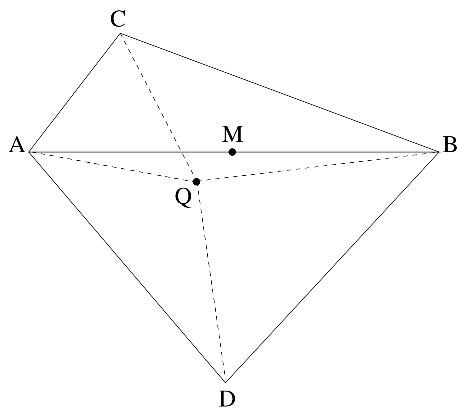

Fig. 12. Centroid refinement of terminal triangles *ABC*, *ADB*.

repeatedly the cosine theorem to compute edges and angles we find that triangles P_3EP_1 and $P_2P_3P_1$ are good triangles of smallest angles 46.89° and 36.7 approximately, while triangle DEP_3 is a bad triangle with smallest angle $\approx$ 28.1°, longest edge DE and non constrained second longest edge EP_3. Consequently this will be improved by using the standard centroid insertion operation where Q will be inside the neighbor Delaunay terminal triangle if this is good. □

6. Improvement properties of the simple centroid insertion

Here we study the intermediate operation IO2, corresponding to the simple insertion of the centroid Q over a couple of Delaunay terminal triangles as shown in Fig. 12. This is performed by joining Q with the four vertices, instead of performing longest edge bisections. This operation corresponds to a Laplacian smoothing of the terminal edge midpoint *M* (inserted by longest edge bisection of the terminal triangles) which improves the triangles obtained by longest edge bisection. It is well known that the Laplacian smoothing works well for convex geometries [20,21], and couples of terminal triangles always define a convex quadrilateral. The following theorem states further improvement results for couples of Delaunay terminal triangles.

Theorem 4. *Let us consider any couple of Delaunay terminal triangles* (t_1, t_2) *where* t_1*(ABC) is bad and worst than the triangle* t_2*(ADB) as shown in* Fig. 13. *Then, considering the coordinate system of center A and x-axis over edge AB, with length of AB equal to 1, it holds that:*

(a) The centroid Q of (t_1, t_2) *is situated inside the better triangle* t_2 *and* $y_Q < 0$. *Furthermore* $-0.217 < y_Q < 0$
(b) For any triangle t_1, *it holds that* $0.25 + x_C/4 < x_Q < 0.375 + x_D/4 < 0.625$
(c) If t_1 *is obtuse then* $x_Q < 0.375 + x_I/4$, *where I is the right intersection point of the circumcircle of* $t_1(ABC)$ *and the arc of circle of center A and radius AB, as shown in* Fig. 5

Proof. The first part of assertion (a) comes from Theorem 1. The lower bound on y_K comes from the case of terminal triangles of angles (30°, 30°, 120°) and (60°, 60°, 60°).

Assertion (b) comes from computing x_Q using the coordinate system described in the body of this theorem. Thus, $x_Q = 0.25 + (x_C + x_D)/4$. Since triangle t_1 has largest angle $\leq$ 120°, then $x_C < 1/2$, and the first part of the upper bound of assertion (b) follows. Since $x_D < 1$, the 0.625 bound follows. The lower bound is quite direct.

Next we prove assertion (c). If t is obtuse, then $x_D < x_I$, where I is the right intersection point of the circles of Fig. 5, and assertion (c) follows. □

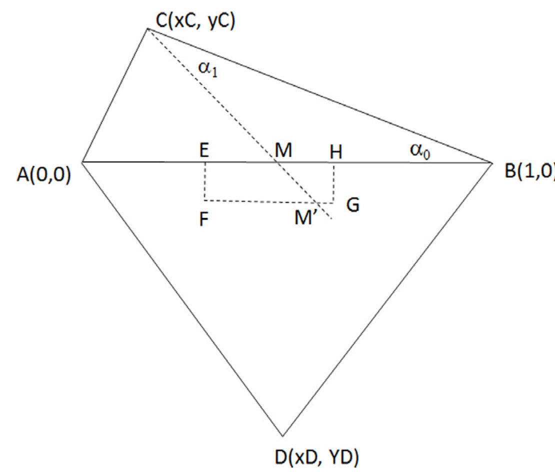

Fig. 13. Delaunay terminal triangles $t_1(ABC), t_2(BAD)$. Centroid Q is inside rectangle EFGH.

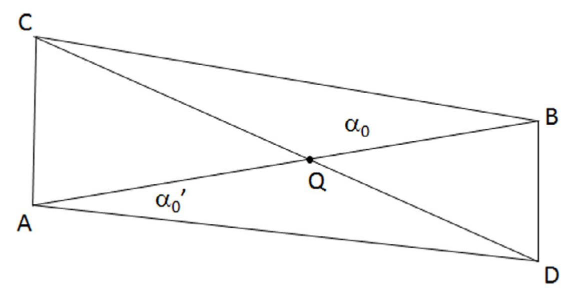

Fig. 14. Operation OP1 over two bad skinny Delaunay terminal triangles with opposite and almost equal smallest angles α_0, α_0'.

Corollary 4.

(a) For Q inside polygon MEFM', the simple insertion of Q (operation IO2) improves the angles α_0, α_1.
(b) For Q inside polygon MM'GH, the simple insertion of Q (operation IO2) improves the angle α_0.

Note that another improvement case occurs when the Delaunay terminal triangles correspond to two acute bad quality triangles with opposite and almost equal smallest angles as shown in Fig. 14. Here the centroid Q (close to the terminal edge) is inserted by using operation IO2 producing two better acute triangles CAQ and QDB and two very obtuse triangles (largest angle > 120°) QAD and CQB, which are eliminated by swapping of their respective longest edges.

7. Analysis of the tuned terminal triangles centroid Delaunay algorithm of Section 2.1

Consider a general PSLG (planar straight line graph) geometry, defined by a set of points, edges and eventually polygonal objects defining exterior boundaries and interior holes. Any PSLG geometry has edge details and non-edge details. Edge details are small edges in the PSLG data, while non edge details are defined by two close isolated interior points, an isolated point close to an input edge, two edges with close points, constrained angles either over the boundaries or interior to the geometry, and vertices over these angles. For an illustration see Fig. 15(a).

Note that the constrained Delaunay triangulation (CDT) of the input PSLG data intuitively defines an edge distribution function to which an optimal size good quality triangulation should be adapted. More specifically this identifies edge details and

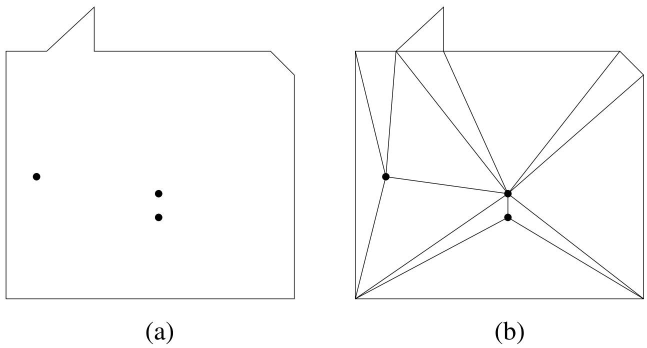

Fig. 15. (a) PSLG geometry; (b) constrained Delaunay triangulation identifies edge details and non edge details.

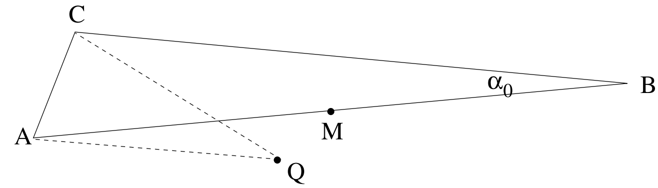

Fig. 16. Triangle ABC with $\alpha_0 < 30°$. Better triangle ACQ and CQB are obtained with respect to those obtained by longest edge bisection.

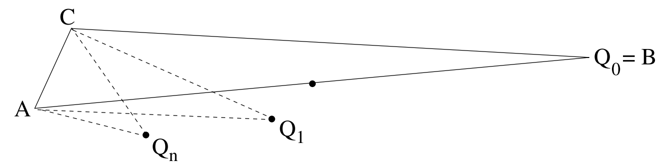

Fig. 17. Points Q_i are introduced until triangle CAQ_n is good.

non-edge details by means of skinny triangles with associated (constrained or non constrained) small edges, very obtuse triangles with largest angled vertex close to an edge data, and triangles with constrained smallest angle. Fig. 15(b) shows the constrained Delaunay triangulation of the example of Fig. 15(a). We will prove that the algorithm of Section 2.1 produces a graded quality mesh with smaller good quality triangles around the PSLG geometry details.

Theorem 5. *Consider any PSLG geometry with constrained angles $\geq$ 30° and the input constrained Delaunay triangulation τ_0 associated with the PSLG data. Then for angle tolerance $\theta_{tol} = 30°$,*

(a) The algorithm finishes with a graded 30° constrained Delaunay triangulation.
(b) The final triangulation is size optimal.

Proof. Given $\theta_{tol} = 30°$, consider the bad triangles with angles less than 30°. To prove part (a), we will study five cases of triangle processing:

Case 1. Non PD terminal triangles. Each bad quality triangle t (with largest angle $>$ 120° either with one or two bad angles) is a non PD terminal triangle which cannot become Delaunay terminal. Then according to Corollary 1, this is eliminated by swapping its longest edge, either by processing t or by processing a Lepp-neighbor bad quality triangle. This operation produces locally more equilateral triangles.

Case 2. Operation OP1 over bad PD terminal triangles. Consider a couple of non constrained Delaunay terminal triangles. Let $t(ABC)$ with $AB \geq BC \geq AC$ be the worst triangle in the couple with $\alpha_0 < 30°$. Here we will consider the three intermediate operations IO1, IO2, IO3 to perform the operation OP1.

According to part(b) of Corollary 2, the longest edge bisection of t would introduce the midpoint M of AB, a better triangle $t_B(ACM)$ with $\alpha_0(t_B) \geq 1.79\alpha_0$, and a bad obtuse triangle t_{OB}. Next, the simple centroid insertion of Q (operation IO2) corresponds to the Laplacian smoothing of point M, which according to Theorem 4 and Corollary 4, improves the worst angles of t_{OB} (introduced by the longest edge bisection) and avoids the repetition of a triangle similar to triangle ABC.

Finally the operation IO3 is performed, which delaunize the current triangulation. If triangle CQB is a non PD terminal triangle, then triangle CQB is eliminated (and improved) by swapping edge CB, either when Q is Delaunay inserted (if there exists a vertex inside the (big) circumcircle of triangle CQB), or by later processing CBQ, or by processing a bad quality neighbor triangle. If triangle CQB is a PD terminal triangle and still bad, then by processing triangle CQB this can become a terminal triangle and the centroid $\tilde{Q}$ of CQB and its neighbor triangle is inserted, which improves the angles (see Fig. 16).

Repetitive use of operation OP1. According to part (e) of Lemma 4, for $\alpha_0 < 16.8°$, triangle CAQ can still be bad. Then for small α_0, a finite sequence of points Q_i need to be inserted in the mesh until a good triangle CAQ_n is obtained (see Fig. 17). The process finishes without refining edge AC (AC is a local smallest edge), unless a close smaller edge induces neighbor refinement. See the termination analysis for more details.

Case 3. Terminal triangles with constrained edges. For Delaunay terminal triangles with the constrained terminal edge, the constrained Delaunay insertion of the terminal edge midpoint is performed and the improvement process continues. For bad triangles with constrained second edge E, the simple constrained Delaunay insertion of the midpoint of E is performed, which according to Corollary 3 improves the triangles close to constrained edges in the PSLG geometry.

Case 4. Couples of good Delaunay terminal triangles. For couples of good quality Delaunay terminal triangles with smallest angles $\geq$ 30°, the centroid Q of the terminal quadrilateral is inserted, which produces four quasiequilateral triangles better than those obtained by longest edge bisection. This is equivalent to a Laplacian smoothing of the terminal edge midpoint introduced by the longest edge bisection of the terminal triangles. This operation improves eventual angles lightly less than 30° that could have been introduced by the longest edge bisection.

Case 5. Triangles with 30° constrained angles. Here, according to Lemma 6, good quality triangles are obtained inside t by inserting a small number of points over the constrained edges.

Termination. The proof on termination is based on the fact that for skinny triangles and according to part (b) of Corollary 2, a sequence of Q_i points are added to the mesh such that the new smallest angles increase at least by a factor of $(1.79)^i$ until a good triangle is obtained. Thus the algorithm stops when every triangle of local smallest edge in τ_0 becomes good (smallest angle $\geq$ 30°), and every remaining intermediate bad quality triangle t is processed or eliminated by edge swapping; and every intermediate almost good terminal triangle is improved by centroid insertion. This produces a good quality triangulation graded around the *PSLG* geometry details. Note that the smallest edge AC is never refined, unless there exists a smaller bad quality triangle t^* such that Lepp(t^*) contains triangle AQ_nC (see Fig. 18).

Optimal size property. This follows from the termination reasoning together with the fact that the average Lepp size tends to be 2 as the refinement proceeds (part a) of Theorem 2). □

Table 1
Algorithms comparison, Key test case, $\theta_{tol} = 33^o$.

Triangle processing	Del centroid algorithm	Ruppert's algorithm [8]	
	Without order	Without order	Ordering triangles
Final Mesh size	229	450	249

Table 2
Mesh sizes for Delaunay centroid algorithm as a function of θ_{tol}.

	Superior lake	Neuss geometry	Square	Chesapeake bay	Long rectangle	Key geometry
	size(τ_0) 528	size(τ_0) 3070	size(τ_0) 9	size(τ_0) 14,262	size(τ_0) 2	size(τ_0) 54
θ_{tol}	size(τ_0)	size(τ_0)	size(τ_0)	size(τ_0)	size(τ_0)	size(τ_0)
30	1835	8338	54	36,803	19	170
33	2273	9939	65	45,883	22	229
34	2512	11,054	70	52,027	25	262
35	3017	12,742	81	63,138	27	349

Table 3
Percentage of triangles added with respect to current version of Triangle[a].

θ_{tol}	Lake superior	Neuss geometry	Square	Chesapeake Bay	Long rectangle	Key geometry
30	0.44	13.18	24.07	4.82	−15.79	22.94
33	−528	1201	16.92	2.41	0.00	10.92
34	−5.29	−2.70	20.00	3.61	−68.00	−8.78
35	$-\infty$	$-\infty$	24.69	$-\infty$	−207.41	5.44

[a]Triangle processes skinny and oversized triangles in order and uses a boundary preprocess step.

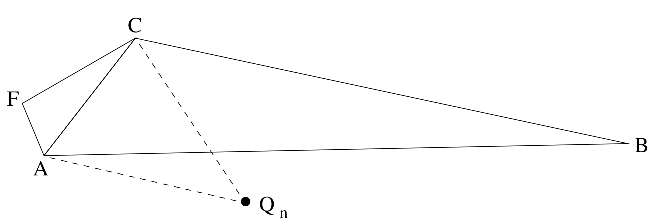

Fig. 18. Neighbor triangle $t^*(ACF)$ induces refinement of triangle AQ_nC to obtain a graded refined triangulation around edge FA.

Theorem 6. *The algorithm is order independent, in the sense that the mesh size is approximately the same by processing the bad triangles in arbitrary order.*

Proof. It is easy to see that the set of terminal edges in the triangulation induces a mesh partition so that every triangle in the partition reaches the same terminal edge. Thus the processing of several bad quality triangles in each partition (independently of their quality) find the same couple of terminal triangles. □

8. Empirical study for geometries with constrained angles ≥ 30°

In Table 1 we compare our algorithm with results reported by Shewchuk [8] on Ruppert's algorithm (without the off-center preprocess of Üngor). Next we present results on the behavior of the Delaunay centroid algorithm for the six geometries of Fig. 19. Table 2 includes final mesh sizes for $\theta_{tol} = 30°, 33°, 34°, 35°$ obtained with our algorithm. See the final triangulations for $\theta_{tol} = 30°$ for these examples in Fig. 19. Table 3 compares the number of triangles obtained with our software, with respect to those obtained with the current version of Triangle [22] which processes skinny and oversized triangles in order, and includes a boundary preprocess technique due to Üngor [9] to minimize the size of the final triangulation. A negative number means our software introduces less triangles than Triangle, while the $-\infty$ symbol means that Triangle does not converge.

It should be noted that: (i) our results are not far from those obtained by the current optimized version of Triangle; (ii) our software works properly until $\theta_{tol} = 35°$ for all the test cases, while Triangle fails for 50% of the test cases ($-\infty$ symbol) for $\theta_{tol} = 35°$; iii) Note that for the key test case and $\theta_{tol} = 33°$, our algorithm produces a final triangulation with 229 triangles against 450 triangles obtained with pure Ruppert algorithm (first-come first split bad quality triangle) and 249 triangles by always processing the worst existing triangle, as reported by Shewchuk [8].

Furthermore, for all the test cases, the average Lepp size is less than 3 from the beginning and quickly becomes less than 2.5, as the refinement proceeds. The algorithm is an easy to implement, order independent, robust method, suitable for use in adaptive finite element methods where good quality meshes are needed to assure convergence. With an adequate triangle data structure that keeps information on neighbor triangles, the refinement is of cost $O(N)$ where N is the number of points inserted.

9. Algorithm for PSLG geometries with constrained angles < 30°

In this section we consider a PSLG geometry with N constrained angles $< 30°$, which implies that the associated CDT will have N non-improvable triangles with constrained angle $< 30°$. However if we introduce and edge size parameter δ, each non-improvable triangle t can be refined until obtaining a non-improvable triangle of size δ (longest edge $< \delta$) and a set of quality triangles in the interior of t. The algorithm is as follows:

Algorithm Algorithm_For_Geometries_With_Small_Constrained _Angles
*Input***:** CDT τ associated with PSLG data, angles tolerance θ_{tol}, and finite number N of constrained angles $< 30^o$
*Output***:** Refined triangulation τ_f with non constrained angles $\geq \theta_{tol}$, and N δ-size small constrained triangles.

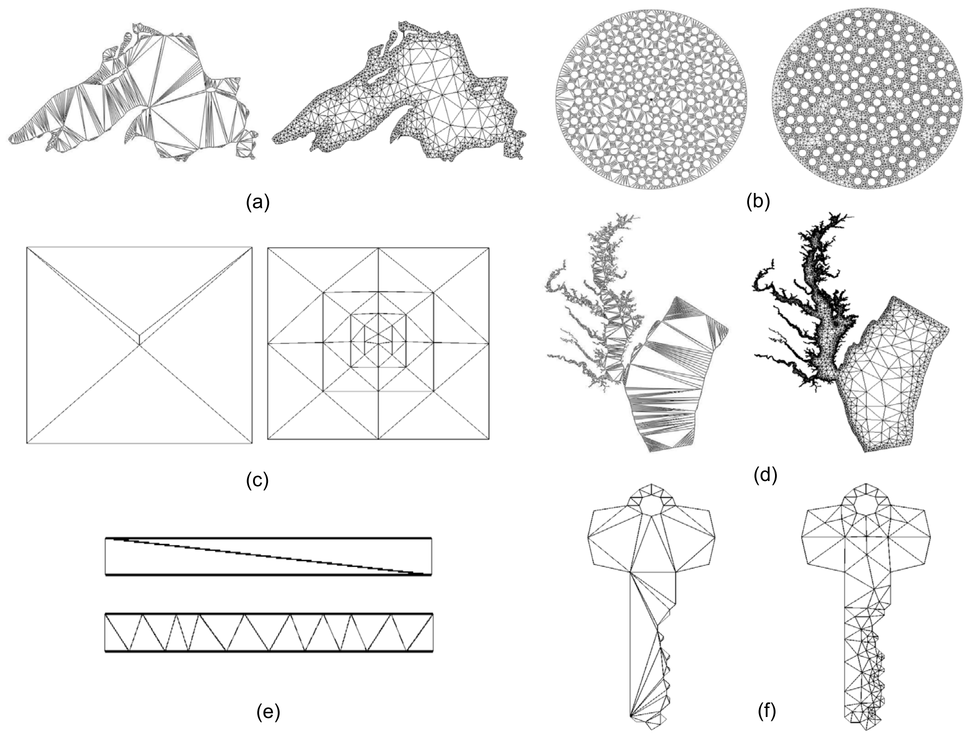

Fig. 19. Quality meshes for $\theta_{tol} = 30°$ (a) lake Superior shape; (b) Neuss shape; (c) Square with skinny triangles; (d) Chesapeake B bay shape; (e) Long rectangle; (f) Key shape.

Find S set of bad quality triangles and set $\tilde{S}$ of non-improvable triangles (constrained angle $< \theta_{tol}$)
Initialize W set of processing triangles with set S and triangles of $\tilde{S}$ with longest edge $> \delta$
for each t in W (while $W \neq \emptyset$) **do**
  while t remains unrefined **do**
    Use Lepp(t) to find Delaunay terminal triangles t_1, t_2 and terminal edge E
    if E is constrained (this includes t_2 null) **then**
      Perform Constrained Delaunay insertion of midpoint of E
    else
      if there exists t (t_1 or t_2) such that $\alpha_t < \theta_{tol}$ and second longest edge L is constrained **then**
        Perform constrained Delaunay insertion of midpoint of L
      else
        Compute centroid Q of terminal triangles, and perform constrained Delaunay insertion of Q
      end if
    end if
    Update $S, \tilde{S}$
    Update W eliminating refined triangles and adding new triangles with non constrained angle $< \theta_{tol}$; and adding new non-improvable triangles with longest edge $> \delta$
  end while
end for

Theorem 7. *Given a parameter δ, for each non-improvable triangle t, the algorithm produces a smaller non-improvable triangle of longest edge $< \delta$, and a set of quality triangles in the interior of t.*

Proof. The algorithm works until inserting points P, Q to obtain a triangle of size δ, (see Fig. 20) by processing triangles in the quadrilateral CAQP by using the edge constrained operations. □

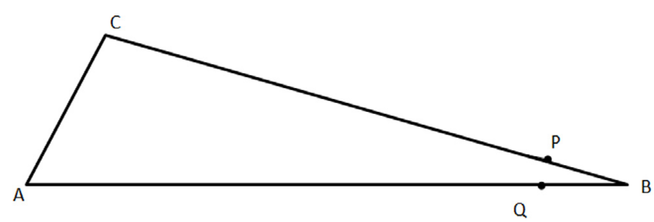

Fig. 20. For small angle ABC with constrained edges AB, CB, the algorithm finishes with a triangle PQB of size δ and quality triangles in region AQPC.

The use of the algorithm for interior small constrained angles is illustrated in Figs. 21, 22, 23, for sets of (shadowed) small constrained angles.

10. Conclusions

We have discussed a simple, easy to implement and robust Lepp algorithm for the construction of quality Delaunay triangulations. This uses three simple mesh operations over couples of Delaunay triangles sharing a common longest (terminal) edge. These are Delaunay insertion of either the centroid Q, or the midpoint of a constrained second edge, and longest edge bisection of the triangles when the terminal edge is constrained.

For any input PSLG geometry with constrained angles $\geq 30°$, the algorithm produces an optimal size triangulation (of size by a factor equal to the smallest possible one) with angles $\geq 30°$, and where the mesh size is independent of the triangle processing order. The last property makes the algorithm very appropriate for parallel implementation.

We also discussed an algorithm for geometries including constrained angles $< 30°$. Obviously these angles cannot be improved, but are isolated when refinement around them is needed. This is important for geometries with complex material changes such a those required for semiconductor applications.

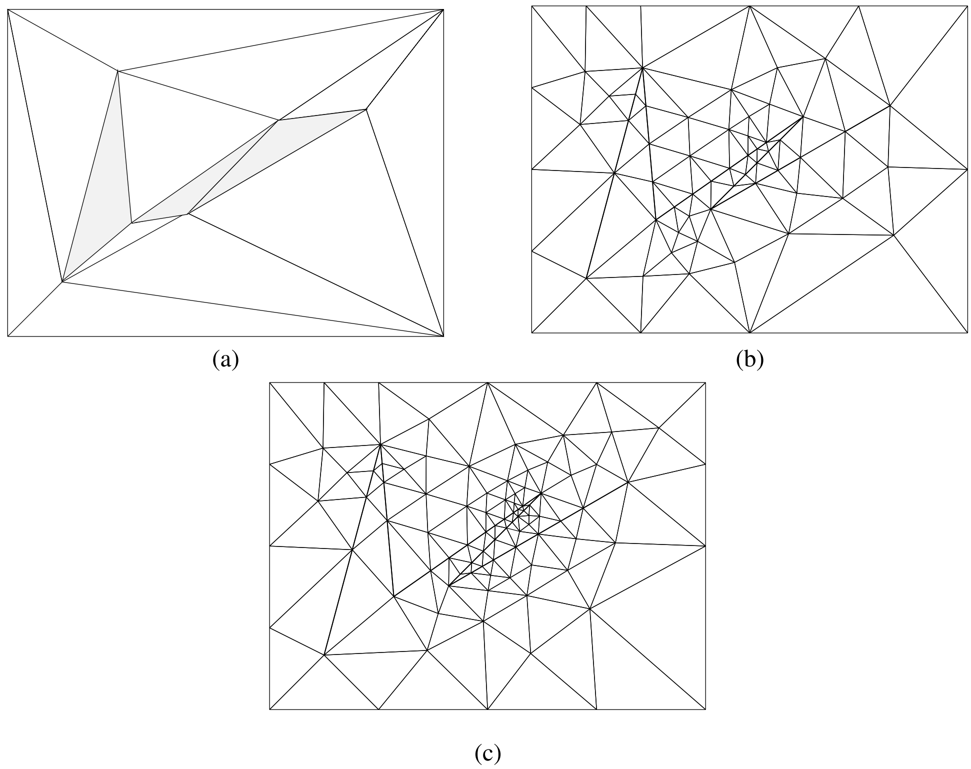

Fig. 21. Rectangle with interior (small) shadowed constrained angles of 10°, 15°, 20°. (a) Initial constrained Delaunay triangulation. Triangulations (b), (c) correspond to different δ values.

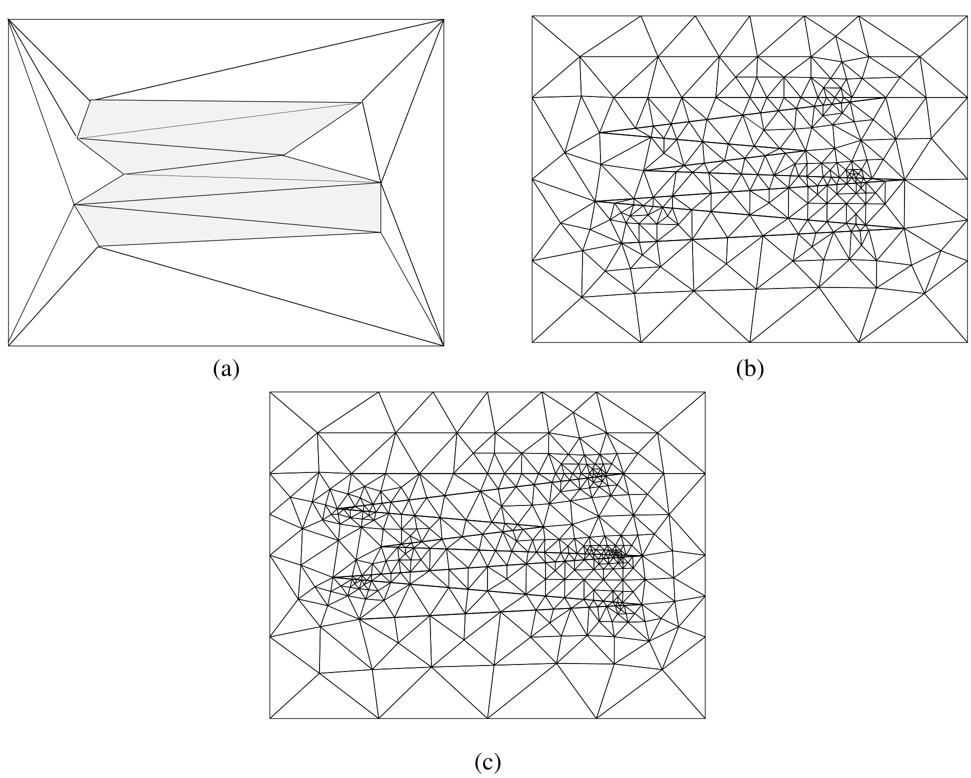

Fig. 22. Rectangle with interior (small) shadowed constrained angles of 7°, 12°, 9°, 6°, 9°, 8°. (a) Initial constrained Delaunay triangulation. Triangulations (b), (c) correspond to different δ values.

It is worth noting that we have been able to develop algorithms that produce almost Delaunay triangulations, without using explicit Delaunay point insertion operations, but simpler mesh operations based on those described in this paper. We have developed research in this direction, which will be published elsewhere.

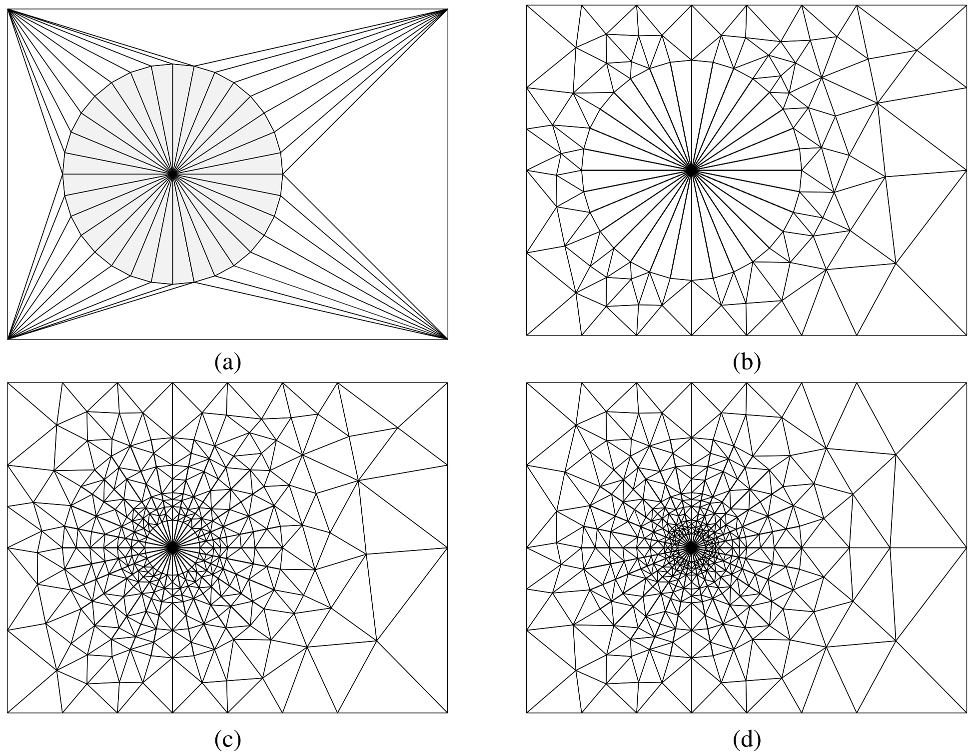

Fig. 23. (a) Initial triangulation with 32 constrained angles of 11.25° sharing the same vertex. Triangulations (b), (c), (d) correspond to different δ values.

Finally, these ideas can be generalized to 3-dimensions. In a recent paper by Balboa, Rodriguez and Rivara [23], an improvement method for tetrahedral meshes was proposed. This uses a simple centroid insertion operation that generalizes the IO2 operation of this paper, combined with a 'swapping of a terminal edge' operation, a mesh improvement operation that follows the ideas of Freitag and Oliver Gooch [24] but applied to the 3-dimensional terminal edge context.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Dedicated to Pedro Rodriguez who coauthored the IMR version of this paper. The first author is grateful to Professor Bruce Simpson for his valuable contribution to the development of this research. Work partially supported by Departament of Computer Science, University of Chile.

References

[1] Rivara MC. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. Int J Numer Methods Eng 1984;20(4):745–56.

[2] Rivara MC. Mesh refinement processes based on the generalized bisection of simplices. SIAM J Numer Anal 1984;21(3):604–13.

[3] Rivara MC. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. Int J Numer Methods Eng 1997;40(18):3313–24.

[4] Bedregal C, Rivara MC. Longest-edge algorithms for size-optimal refinement of triangulations. Computer-Aided Des 2014;46:246–51.

[5] Bedregal C, Rivara MC. New results on Lepp–Delaunay algorithm for quality triangulations. Proced Eng 2014;124:317–29.

[6] Rivara MC, Calderon C. Lepp terminal centroid method for quality triangulation. Comput-Aided Des 2010;42(1):58–66.

[7] Ruppert J. A delaunay refinement algorithm for quality 2-dimensional mesh generation. J Algorithms 1995;18(3):548–85.

[8] Shewchuk JR. Delaunay refinement algorithms for triangular mesh generation. Comput Geom 2002;22(1–3):21–74.

[9] Erten H, Ungor A. Quality triangulations with locally optimal steiner points. SIAM J Sci Comput 2009;31:2103–30.

[10] Cheng SW, Dey TK, Shewchuk JR. Delaunay mesh generation. In: Computational geometry. CRC computer and information science series, vol. 22, no. 1–3, CRC Press; 2013, p. 21–74.

[11] Rivara MC, Rodriguez-Moreno P. Tuned terminal triangles centroid delaunay algorithm for quality triangulation. In: Roca X, Loseille A, editors. Proceedings 27h international meshing roundtable, Albuquerque, USA; 2018. Lecture Notes in Computer Science, Springer; 2018.

[12] de Berg M, van Kreveld, Overmars M, Shwarzkopf O. Computational geometry: algorithms and applications. second ed.. Springer Berlin Heidelberg; 2000.

[13] Rosenberg IG, Stenger F. A lower bound on the angles of triangles constructed by bisecting the longest side. Math Comp 1975;29(130):390–5.

[14] Stynes M. On faster Convergence of the bisection method for certain triangles. Math Comp 1979;33(146):717–21.

[15] Stynes M. On faster convergence of the bisection method for all triangles. Math Comp 1980;35:1195–201.

[16] Simpson RB, Rivara MC. Geometrical mesh improvement properties of delaunay terminal edge refinement. Technical report CS-2006-16, The David Cheriton School of Computer Science, University of Waterloo; 2006, p. 536–44.

[17] Rivara MC, Hitschfeld N, Simpson RB. Terminal edges delaunay (smallest angle based) algorithm for the quality triangulation problem. Comput-Aided Des 2001;33(3):263–77.

[18] Sibson R. Locally equiangular triangulations. Comput J 1978;21:243–5.

[19] Gutierrez C, Gutierrez F, Rivara MC. Complexity of the bisection method. Theoret Comput Sci 2007;382(2):131–8.

[20] Field DA. Laplatian smoothing and Delaunay triangulations. Commun Appl Numer Methods 1988;4(6):709–12.

[21] Freitag LA. On combining laplatian and optimization-based mesh smoothing techniques, vol. 220. ASME Applied Mechanics Division-Publications-AMD; 1997, p. 37–44.
[22] Shewchuk JR. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In: Lin MC, Manocha D, editors. Applied computational geometry. Lecture notes in computer science, vol. 1148, 1996, p. 203–22.
[23] Balboa F, Rodriguez-Moreno P, Rivara MC. Terminal star operations algorithm for tetrahedral mesh improvement. In: Roca X, Loseille A, editors. Proceedings 27h International Meshing Roundtable, Albuquerque, USA, 2018. Lecture Notes in Computer Science, Springer; 2019.
[24] Freitag LA, Olliver-Gooch C. Tetrahedral mesh improvement using swapping and smoothing. Internat J Numer Methods Engrg 1997;40:3979–4002.