Computer-Aided Design 46 (2014) 246–251

Contents lists available at ScienceDirect

Computer-Aided Design

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

Technical note

Longest-edge algorithms for size-optimal refinement of triangulations

Carlos Bedregal *, Maria-Cecilia Rivara

Department of Computer Science, University of Chile, Chile

HIGHLIGHTS

- We improve geometrical results on longest-edge refinement algorithms.
- We provide new results on the refinement propagation of the Lepp-bisection algorithm.
- The iterative application of the algorithm improves the quality of the triangulation.
- We perform an empirical study of the algorithm and the behavior of the propagation.
- We also review mathematical properties of the iterative longest-edge bisection.

ARTICLE INFO

Keywords:
Computational geometry
Mesh refinement algorithms
Longest-edge bisection

ABSTRACT

Longest-edge refinement algorithms were designed to iteratively refine the mesh for finite-element applications by maintaining mesh quality (assuring a bound on the smallest angle). In this paper we improve geometrical results on longest-edge refinement algorithms and provide precise bounds on the refinement propagation. We prove that the iterative application of the algorithm gradually reduces the average extent of the propagation per target triangle, tending to affect only two triangles. We also include empirical results which are in complete agreement with the theory.

1. Introduction

Mesh generation and refinement techniques are widely used in applications that require a decomposition of geometric objects for processing. The domain is typically discretized into a mesh composed of triangles. For most applications, the time required to process a geometric object depends on the size and quality of the decomposition. Thus, provably good mesh generation algorithms are desirable.

We consider a conforming 2-dimensional triangulation τ, where pairs of neighbor triangles have either a common vertex or a common edge. We also assume a quality triangulation in the sense that the minimum angle is greater than or equal to an angle parameter θ. For a triangle set $S \subset \tau$, the triangle-set refinement problem needs to produce a conforming refined triangulation τ' of quality analogous to τ, and where every triangle $t \in S$ has been refined. Note that for finite element applications, triangle-set S corresponds to the triangles in τ with unacceptable finite element error.

Longest-edge refinement algorithms were designed to deal with the iterative triangle-set refinement problem. These algorithms perform local refinement based on the bisection of triangles by their longest edge, and produce conforming triangulations that maintain the quality of the input mesh. The *local refinement* implies the refinement of the triangles in the current triangle set and some of their neighbor triangles due to refinement propagation.

In this paper we focus on the study of refinement propagation. We show that the iterative application of the algorithm asymptotically reduces the propagation to a constant value (less than three triangles in practice). We prove that the overall quality of the triangulation improves, increasing the area covered by quasi-equilateral triangles and isolating the triangles with the smallest angles.

2. Previous results on longest-edge bisection of triangles

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

We call any triangle t that behaves like an equilateral triangle with respect to its iterative longest-edge bisection (Fig. 2), a *quasi-equilateral triangle*. For quasi-equilateral triangle $t(ABC)$, the longest-edge bisection of its descendants, triangles AMC and MNC, is respectively performed by the edges AC and MC. Note that these bisections only produce edges parallel to the edges of initial triangle ABC. This implies that at most four similarly distinct triangles are produced: ABC, MBC, AMC and MNC. Therefore, further longest-edge bisections of t and its descendants will only produce triangles similar to one of these four triangles.

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

0010-4485/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cad.2013.08.040

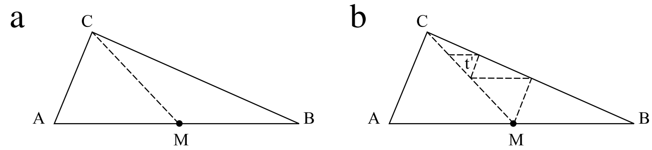

Fig. 1. (a) Longest-edge bisection of triangle $t(ABC)$, with the partition of t by longest edge AB. (b) Repetitive longest-edge bisection of some descendants of t.

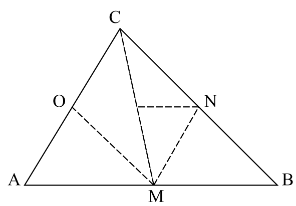

Fig. 2. Quasi-equilateral triangle $t(ABC)$ and its first descendants.

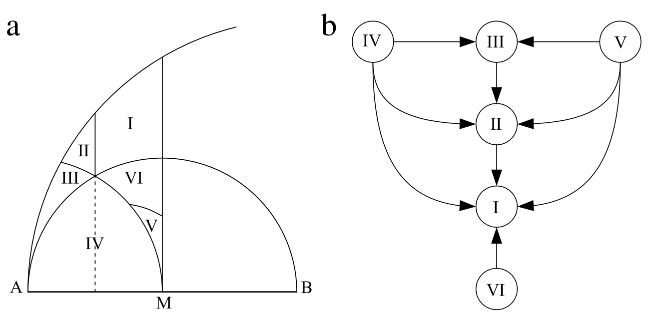

Fig. 3. (a) Regions defining the classification of triangles $t(ABC)$. Virtual vertex C lies in one of the regions defining a triangle t with $AB \geq BC \geq CA$. (b) Directed graph showing the transition of the new triangles generated by recursive longest-edge bisection.

The mathematical properties of the iterative longest-edge bisection of triangles were studied by Rosenberg and Stenger [1], Stynes [2], and Adler [3]. These can be summarized as follows: (a) for any triangle t of smallest angle α, the iterative longest-edge bisection of t and its descendants produce triangles t' whose smallest interior angles are always greater than or equal to $\alpha/2$ (Fig. 1(b)); (b) every triangle generated is similar to one of a finite number of associated non-similar triangles; (c) the global iterative bisection (the bisection of all the triangles in the preceding level) covers, in a monotonically increasing form, the area of t with quasi-equilateral triangles.

More recently, Gutierrez et al. [4] studied the complexity of the iterative bisection of a triangle, introducing a taxonomy of triangles based both on the evolution and the number of non-similar triangles produced. They defined six classes of triangles, considering the geometric position where vertex C of a triangle $t(ABC)$ lies, considering that $AB \geq BC \geq CA$ (Fig. 3(a)). Note that the half-circle of center M and radius AM correspond to right-angled triangles. Triangles from Regions I and VI correspond to quasi-equilateral triangles. In [4] it was proved that the generation of new triangles stops when a quasi-equilateral triangle is obtained, which happens in $O\left(\frac{1}{\alpha}\right)$ steps, with α the smallest angle of t. This behavior is summarized in Fig. 3(b). The graph shows the path between regions in the generation of new non-similar triangles by longest-edge bisection, as well as the convergence to quasi-equilateral triangles.

The following lemmas summarize these results.

Lemma 2.1. *The iterative bisection of a triangle $t(ABC)$ generates a finite number of non-similar triangles that move throughout the six regions of Fig. 3 (a). Furthermore, Region I or VI triangles generate at most four non-similar triangles.*

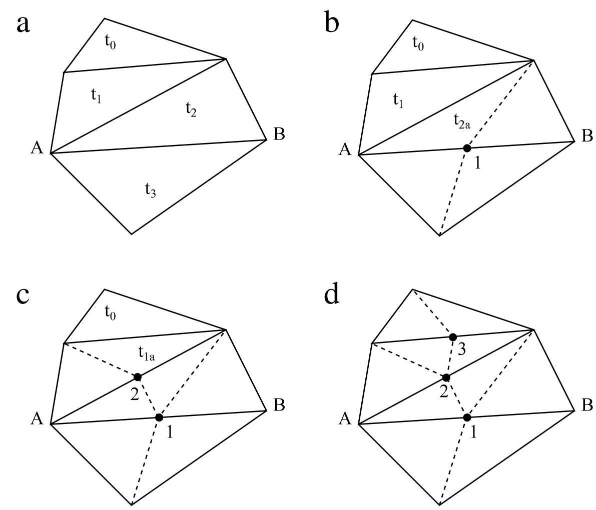

Fig. 4. (a) AB is an interior terminal edge shared by terminal triangles $\{t_2, t_3\}$ of $\texttt{Lepp}(t_0) = \{t_0, t_1, t_2, t_3\}$. (b) Vertex 1 is added by the bisection of terminal triangles $\{t_2, t_3\}$. (c) Vertex 2 is added by the bisection of terminal triangles $\{t_1, t_{2a}\}$. (d) Final triangulation.

Lemma 2.2. *The iterative bisection of triangles from Regions* II *to* V *produce a sequence of new non-similar triangles until triangles of Region* I *or* VI *are obtained.*

3. The Lepp-bisection refinement algorithm

Longest-edge refinement algorithms [5,6] were designed to deal with the local iterative refinement of triangulations as needed for finite element method applications. In order to refine a set of triangles S of a triangulation τ, these algorithms perform the longest-edge bisection of the triangles in S (and some of their neighbors and descendants) to produce a *conforming* triangulation, where pairs of neighbor triangles have either a common vertex or a common edge. Fig. 4 shows the use of the algorithm to refine the triangle $t(ABC)$ and how the refinement is propagated in order to obtain a valid (conforming) triangulation.

Longest-edge bisection refinement algorithms inherit the properties discussed in Section 2. In particular it is guaranteed that every refined triangulation has smallest angle $\geq \theta/2$, where θ is the smallest angle in the initial triangulation.

The Lepp-bisection algorithm [6,7] is an efficient formulation of pure longest-edge bisection algorithms [5], that maintains a conforming triangulation throughout the refinement process. This algorithm is based on the concepts of terminal edges and longest edge propagating path.

An edge E is called a *terminal edge* in triangulation τ if it is the longest edge of every triangle that shares E. The triangles sharing E are called *terminal triangles* (edge AB in Fig. 4(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, $\texttt{Lepp}(t_0)$, is the ordered sequence $\{t_j\}_0^{N+1}$, where t_j is the neighbor triangle on the longest edge of t_{j-1}, and $\texttt{longest\_edge}(t_j) > \texttt{longest\_edge}(t_{j-1})$, for $j = 1, \ldots, N$. The process ends by finding a terminal edge.

To refine a triangle t_0, the Lepp-bisection algorithm proceeds as follows: (1) finds $\texttt{Lepp}(t_0)$ and a pair of terminal triangles t_N and t_{N+1} which share terminal edge E; (2) performs the longest edge bisection of t_N and t_{N+1} by the midpoint of E. This process is repeated until initial triangle t_0 is refined. Algorithm 1 presents a generalization of the algorithm for the triangle set refinement problem.

Algorithm 1 Lepp-bisection algorithm

Input: A quality triangulation τ and a set S_{ref} of triangles to be refined
Output: A quality triangulation τ' such that each $t \in S_{ref}$ has been refined

```
for each t in S_ref do
  while t remains in τ do
    Find Lepp(t), terminal triangles t1, t2 and terminal
    edge l. Triangle t2 can be null for boundary l
    Select point P, midpoint of edge l
    Perform bisection by P of triangles t1, t2
    Update S_ref
  end while
end for
```

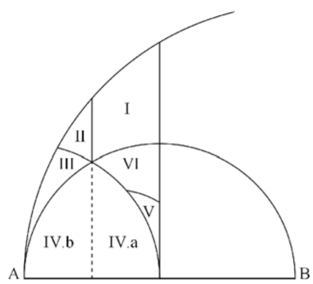

Fig. 5. Extended geometric classification of triangles $t(ABC)$ for the analysis of the behavior of longest-edge bisection algorithms.

Fig. 4 illustrates the refinement of triangle t_0. The first step computes Lepp(t_0) and terminal triangles, t_2 and t_3, which are then bisected by their longest edge (Fig. 4(b)). Next, Lepp(t_0) is recomputed with terminal triangles t_1 and t_{2a} (a sub-triangle of t_2) as shown in Fig. 4(c). Then, a last computation of Lepp(t_0) occurs, with terminal triangles t_0 and t_{1a} (a sub-triangle of t_1), whose bisection refines target triangle t_0 and stops the process (Fig. 4(d)).

4. Study of the refinement propagation

As stated and illustrated in the preceding section, for refining an individual triangle t_0, the algorithm locally propagates the refinement to a set of largest triangles related with Lepp(t_0) in order to produce a conforming triangulation. Thus two new triangles are added by refining the target triangle t_0, and N_{prop} new triangles are introduced by refinement propagation. For an illustration see the example of Fig. 4, where $N_{prop} = 8$.

Note that N_{prop} depends on: (a) the Lepp size; and (b) the number of bisections performed inside each propagating triangle t^*, which in turn depends on the shape of t^*.

4.1. Number of bisections inside propagating triangles

Firstly, note that independent of the triangle shape, a fixed number of bisections is needed to eliminate non-conforming midpoints situated over either the longest edge, or the second longest edge of t^* as shown in Fig. 6. In exchange, for non-conforming midpoints situated over the smallest edge of t^*, some additional points in the interior and/or over the edges of t^* are introduced, depending on the geometric triangle class of t^*.

Bedregal and Rivara [8] showed that working with acceptable quality triangulations the Lepp-bisection algorithm performs a constant number of bisections per triangle. They extended the classification of Gutierrez et al. [4] to eliminate non-conforming points. (See Fig. 5.) They proved that any triangle in regions I, V and VI requires at most two bisections to be successfully refined; region II triangles require at most four bisections, while region IV.a triangles require only three. The worst case scenario occurs for triangles in regions III and IV.b, where the number of bisections is bounded by $O(\log^2 \frac{1}{\alpha})$, with α the triangle's smallest angle. Since these poor quality triangles tend to insert multiple non-conforming points over their edges (which then propagate to neighboring triangles), they generalize this bound to $O(\log^2 \frac{1}{\alpha_{min}})$ for triangulations with arbitrary smallest angle α_{min}. Lemma 4.1 summarizes these results.

Lemma 4.1. *The average number of longest-edge bisections performed by the Lepp-bisection algorithm to maintain a triangle conforming is constant. Furthermore, this constant is less than* 5 *for triangles from regions* I, II, IV.a, V *and* VI*; for region* III *and* IV.b *triangles, it is bounded by* $O(\log^2 \frac{1}{\alpha})$.

4.2. Bounding the refinement propagation

Here we analyze the evolution of Lepp(t) throughout the refinement process. Consider that, (1) whenever a triangle is bisected by the midpoint of its longest edge, a propagated refinement occurs to maintain the mesh valid, and (2) the propagation is finite since the refinement always propagates to bigger triangles.

To refine an individual triangle, the size of the propagation depends on the distribution of terminal triangles in the triangulation since they represent the end of a Lepp. Therefore, the more frequent the terminal triangles are, the shorter the average propagation of the refinement is. In the rest of this section we are interested in measuring the proportion of terminal triangles in a triangulation.

Also, the size of the propagation is directly affected by the presence of terminal triangles in the triangulation since they represent the end of a Lepp. Therefore, the more frequent the terminal triangles are, the shorter the average propagation of the refinement is. We are interested in measuring the proportion of terminal triangles in a triangulation.

In what follows, we show that the percentage of terminal triangles increases as the refinement proceeds. To this end we will take advantage of a result on the 4-Triangles refinement algorithm (a special longest-edge algorithm [9]) by Suarez et al. [10]. They showed that the iterative application of the 4-Triangles refinement algorithm on a triangulation τ increases the proportion $B(\tau) = \frac{T(\tau)}{N(\tau)}$, where $T(\tau)$ is the number of pairs of terminal triangles in the mesh τ, and $N(\tau)$ is the total number of triangles; tending to cover τ with terminal triangles, and reducing the average length of propagation to five triangles.

Note that one 4-Triangles partition of a triangle is equivalent to two steps of longest-edge bisections of a quasi-equilateral triangle. This, along with Lemma 2.2, allows us to extend these results to the longest-edge bisection algorithm.

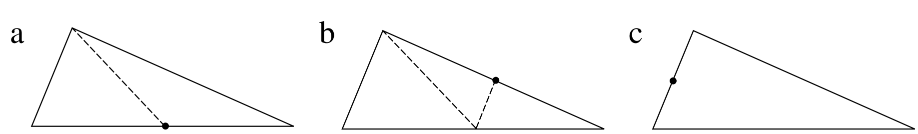

Fig. 6. Cases of longest-edge bisections to solve non-conforming midpoints. (a) The longest edge requires one bisection. (b) The second longest edge requires two bisections. (c) For the shortest, edge the number of bisections required is defined by the triangle's region.

Proposition 4.2. *For any triangulation τ, the global iterative use of the Lepp-bisection of triangles generates triangulation covered by terminal triangles.*

Proof. The iterative longest-edge bisection of triangles generates triangles from Regions I and VI (Lemma 2.2). Furthermore, the iterative longest-edge bisection of these triangles presents the same bisection pattern as the 4-Triangles partition. Therefore, the theoretical results on the propagation problem for the 4-Triangles algorithm hold for the Lepp-bisection algorithm. □

Theorem 4.3. *The global iterative application of the Lepp-bisection algorithm increases the proportion of Region* I *and* VI *triangles in the mesh, approaching* 1 *as the number of iterations increases.*

Proof. Given a triangle t with smallest angle α and largest angle γ, function F_i models the number of triangles from Regions I and VI after the i-th nested bisection of the triangles in t. The following recurrence relations represent the growth of function F_i for each similarity region:

(i) $F_i = 2F_{i-1}$, and $F_0 = 1$; for Regions I and VI triangles (See Fig. 7(a))
(ii) $F_i = 2F_{i-1} + 2F_{i-2}$, and $F_0 = 0, F_1 = 1$; for Region II triangles (See Fig. 7(b))
(iii) Region III triangles will behave like Region II for $i > 1.7 \log\left(\frac{\pi}{6\alpha}\right)$, the number of bisections needed to obtain a Region I or II triangle. (See Fig. 7(c))
(iv) Region IV: will behave like Region I, II or III triangles for $i > \left(\delta - \frac{\pi}{2}\right)/\alpha$, the number of bisections to obtain a Region I or II triangle.
(v) Region V: will not behave worse than Region III.

Function F_i only considers the nested bisection of a triangle. It does not consider the internal propagation that the algorithm performs in order to maintain the conforming mesh (if it did, that would translate into a faster appearance of quasi-equilateral triangles during a single refinement step). Therefore, F_i represents a lower bound on the algorithm's generation of Regions I and VI triangles. For Region III, IV and V triangles, Lemmas 2.1 and 2.2 ensure the convergence of the generation of Region I and VI triangles.

As the number of i iterations tends to infinity, $\frac{F_i}{N(\tau_i)}$ tends to 1, with $N(\tau_i)$ the number of triangles of triangulation τ at iteration i. $N(\tau_i)$ is represented by the recurrence relation $N_i = 2N_{i-1}$ since every triangle is bisected at each iteration, doubling the number of triangles. □

Fig. 7 shows the transition among regions during the iterative longest-edge bisection of triangles until obtaining quasi-equilateral triangles. Fig. 7(a) shows how quasi-equilateral triangles only generate other quasi-equilateral triangles. Fig. 7(b) shows the transition tree of Region II, IV.a and V triangles (the green triangle represents the refinement tree of a Region I triangle tree). Fig. 7(c) shows the transition behavior of Region III, IV.b and V triangles (the white triangle represents the refinement tree of a Region I, II or III triangle). The generation of new subtrees in Fig. 7(c) halts within a finite number of steps with the appearance of a Region I or II triangle (Lemma 2.1).

Corollary 4.4. *The iterative application of the Lepp-bisection algorithm gradually reduces the average extent of the propagation, tending to* 2 *triangles.*

Proof. The proof follows from the direct application of Theorem 4.3 and Proposition 4.2.

Note that Region I and VI triangles are regarded as good quality triangles, mostly having a smallest angle $\alpha \geq \frac{\pi}{6}$. Therefore, the Lepp-bisection algorithm not only reduces the work of refinement in repetitive applications, but improves the overall quality of the triangulation.

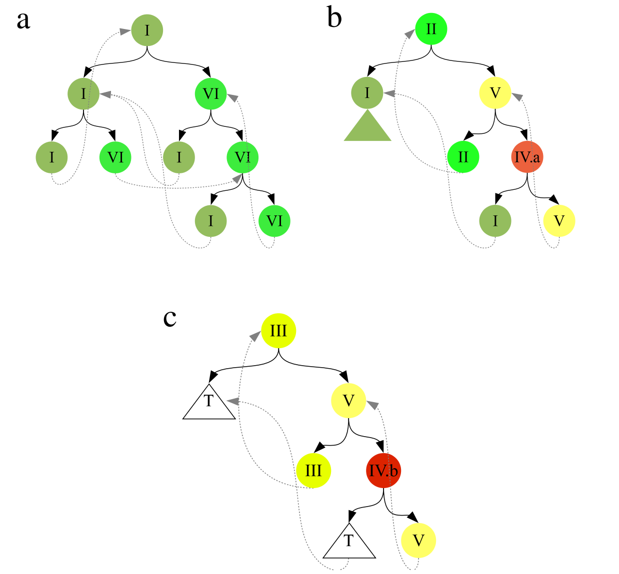

Fig. 7. Refinement trees for triangles of: (a) Region I and VI triangles. (b) Region II (IV.a and V) triangles. (c) Region III (IV.b and V) triangles. Upward gray arrows represent the appearance of a triangle similar to an ancestor. Triangles represent a subtree. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

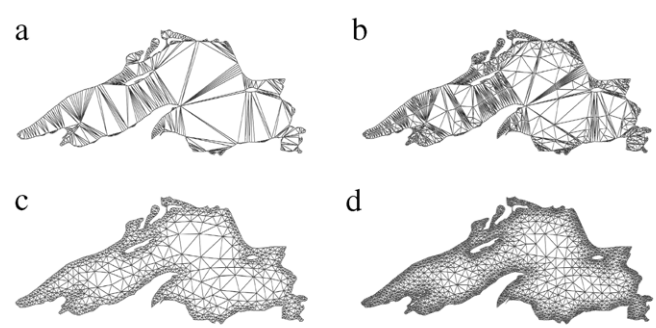

Fig. 8. Lake Superior geometry. (a) Initial triangulation `Superior-bad` (528 triangles). (b) Refined triangulation `Superior-bad` after one iteration of global refinement. (c) Initial triangulation `Superior-good` (1875 triangles). (d) Refined triangulation `Superior-good` after one iteration of global refinement.

5. Experimental results

Here we study the evolution of the quality of the triangulation throughout the iterative application of the Lepp-bisection algorithm. We compute the Lepp-size and the percentage of terminal triangles and quasi-equilateral triangles in the refined triangulations.

We tested the performance of the algorithm using two Delaunay triangulations of the Lake Superior geometry: `Superior-good`, a good quality triangulation with smallest angle $\alpha_{min} \geq 30°$ (Fig. 8(c)); and `Superior-bad`, a poor quality triangulation with smallest angle $\alpha_{min} \geq 1.6°$ (Fig. 8(a)). Although longest-edge bisection algorithms are used in practice over triangulations of acceptable quality, in order to provide a stronger empirical validation of Corollary 4.4, we also tested the algorithm's behavior over Delaunay triangulations of randomly generated points over a quadrilateral region. We generated 10 meshes of 1000, 5000 and 10,000 points (`D1k`, `D5k` and `D10k` respectively).

Table 1 shows the initial statistics of all the evaluated meshes. The second column refers to the number of triangles in the triangulation. The third column refers to the average smallest angle α of the triangles. The fourth column, Avg. `Lepp-set()`, refers to

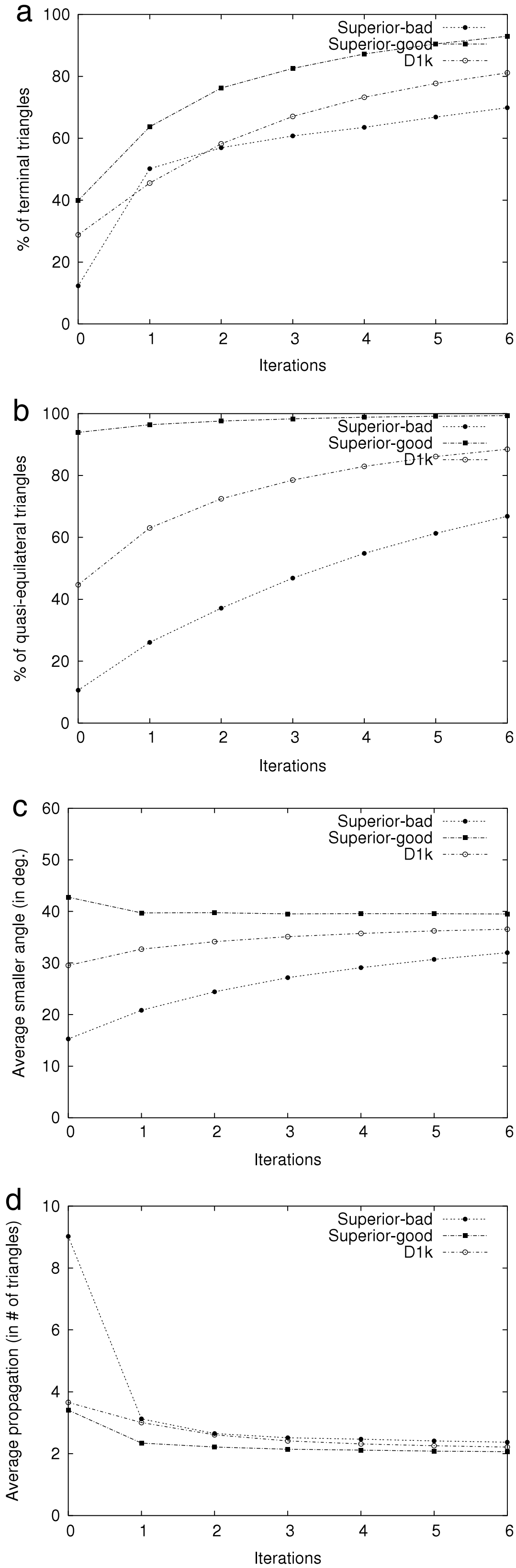

Fig. 9. Evolution of the refined triangulations for iterative global refinement. (a) Percentage of terminal triangles. (b) Percentage of quasi-equilateral triangles. (c) Average smallest angle. (d) Average number of triangles affected by propagation.

the average number of triangles locally refined due to propagation. The fifth column refers to the proportion of terminal triangles in the mesh, while the sixth column refers to the proportion of quasi-equilateral triangles (regions I and VI).

It must be noted that even when the average α shown in Table 1 is good for the triangulations of randomly generated points, these triangulations contain a small number of initial triangles with minimum angle α close to 0°. The triangulation of `Superior-bad` also contains some initial poor quality triangles (with α close to 2°).

5.1. Iterative global refinement

Starting with input triangulations of Table 1, we iteratively and globally applied the algorithm to refine every triangle of the preceding triangulation, until obtaining one million triangles. Fig. 9 shows the evolution of four statistics over the refined triangulations during this process. Since all the meshes of random points show analogous behavior we only include triangulation `nD1k`.

Fig. 9(a) and (b) show that both the percentage of terminal triangles and the percentage of quasi-equilateral triangles in the mesh monotonically increases as the refinement proceeds, even for those regarded as bad quality initial triangulations.

Fig. 9(c) shows that the average smallest angle approaches about 40°. In this sense the triangulation is populated by good-quality triangles that are also easy to process even if they are affected by the propagation (recall Lemma 4.1). The results shown in Fig. 9(d) validate the reduction of the propagation length during iterative refinement since the Lepp-size quickly converges to an average of two triangles (pairs of terminal triangles).

5.2. Iterative random refinement

Here, at each step we iteratively applied the algorithm to refine one random triangle of the current triangulation for the triangulations of Table 1. Since the randomly selected triangle could have been previously refined, we randomly refine a number of triangles equal to the size of the initial triangulation to increase the probability of refining every initial triangle at least once. Table 2 presents the same statistics as Table 1 for the final refined triangulations. The initial comparison of both tables shows the increasing of good quality and terminal triangles, as well as the reduction of the Lepp-size to an average of three triangles. Note that for quality-acceptable triangulations, even when local refinement is applied, the overall quality of the refined (also quality-acceptable triangulations) is still improved: the average α increases to around 40°.

6. Conclusions

In this paper we presented a complete study on refinement propagation for the Lepp-bisection algorithm, proving that size-optimal meshes are obtained. Our experiments show that both the number of triangles generated inside propagating triangles, and the size of the propagation remain constant. The Lepp-bisection algorithm produces more quasi-equilateral triangles in each iteration. Thus, better quality triangles tend to cover the propagation path, so future iterations are processed faster with shorter longest-edge propagating paths, while the triangles with the smallest angles are isolated. In practice, longest edge algorithms are efficient techniques for fast and robust refinement of good quality triangulations.

Acknowledgment

The authors would like to acknowledge Pedro Rodriguez for the software used for the empirical work.

Table 1
Initial statistics of the triangulations evaluated.

Triangulation	Size	Avg. α (in deg.)	Avg. Lepp-set()	Terminal triang. (%)	Reg. I, VI triang. (%)
Superior-bad	528	15.28	9.02	0.12	0.11
Superior-good	1,875	42.76	3.41	39.89	93.92
D1k	1,981	29.53	3.66	28.82	44.67
D5k	9,969	30.36	3.71	29.38	46.37
D10k	19,976	30.19	3.68	29.45	45.71

Table 2
Statistics of the output triangulations after the iterative refinement of random triangles.

Triangulation	Size	Avg. α (in deg.)	Avg. Lepp-set()	Terminal triang. (%)	Reg. I, VI triang. (%)
Superior-bad	1,923	22.51	2.99	48.52	32.37
Superior-good	5,078	39.93	2.91	41.78	97.49
D1k	5,670	33.85	3.10	41.64	71.13
D5k	28,479	34.23	3.13	41.11	72.38
D10k	56,885	34.62	3.15	40.57	73.39

References

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

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

[3] Adler A. On the bisection method for triangles. Math Comp 1983;40:571.

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

[5] Rivara M-C. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. Internat J Numer Methods Engrg 1984;20(4):745–56.

[6] Rivara M-C. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. Internat J Numer Methods Engrg 1997;40(18):3313–24.

[7] Rivara M-C. Lepp-bisection algorithms, applications and mathematical properties. Appl Numer Math 2009;59(9):2218–35.

[8] Bedregal C, Rivara M-C. A study on size-optimal longest edge refinement algorithms. In: Jiao X, Weill J-C, editors. Proc. of the 21st int. meshing roundtable. Berlin, Heidelberg: Springer; 2013. p. 121–36.

[9] Rivara M-C, Iribarren G. The 4-triangles longest-side partition of triangles and linear refinement algorithms. Math Comp 1996;65(216):1485–502.

[10] Suárez JP, Plaza A, Carey GF. The propagation problem in longest-edge refinement. Finite Elem Anal Des 2005;42(2):130–51.