Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 82 (2012) 2971–2981
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original article

A local refinement algorithm for the longest-edge trisection of triangle meshes

Ángel Plaza[a], Sergio Falcón[a], José P. Suárez[b,*], Pilar Abad[b]

[a] Department of Mathematics, University of Las Palmas de Gran Canaria, Spain
[b] Department of Cartography and Graphic Engineering, University of Las Palmas de Gran Canaria, Spain

Received 6 April 2010; received in revised form 1 April 2011; accepted 7 July 2011
Available online 19 July 2011

Abstract

In this paper we present a local refinement algorithm based on the longest-edge trisection of triangles. Local trisection patterns are used to generate a conforming triangulation, depending on the number of non-conforming nodes per edge presented. We describe the algorithm and provide a study of the efficiency (cost analysis) of the triangulation refinement problem. The algorithm presented, and its associated triangle partition, afford a valid strategy to refine triangular meshes. Some numerical studies are analysed together with examples of applications in the field of mesh refinement.

Keywords: Local refinement; Triangle trisection; Longest-edge algorithms

1. Introduction

Refinement algorithms based on the longest-edge bisection of triangles were developed to deal with adaptive discretisations in 2D and 3D, [7–9,11,15,17,18,25]. These algorithms guarantee fine-quality structuring with irregular, nested triangulations, mainly as a result of the 'bounded' characteristics of the small angles of the triangles thereby created. A second reason for this guaranteed quality is the natural refinement propagation outside the targeted refinement area, leading to meshes with gradual transitions between smaller and larger elements (smoothness condition). Such refinement algorithms when applied iteratively to an initial mesh produce a sequence of nested meshes suitable for multi-grid techniques and hierarchical data structures [18,19].

For a given triangle t, iterative application of the longest-edge bisection to the triangle and its descendants produces an infinite family of triangles, called, to the effect of this paper, the family of triangles $\mathcal{F}(t)$. It has been proved that if τ is the smallest interior angle of the initial triangle t and τ_n is any interior angle of any triangle $\Delta \in \mathcal{F}$, then $\tau_n \geq \tau/2$ [21].

* Corresponding author at: University of Las Palmas de Gran Canaria, Escuela de Ingenierías Industriales y Civiles, 35017-Las Palmas de Gran Canaria, Spain. Tel.: +34 928 45 72 68; fax: +34 928 45 18 72.
E-mail address: jsuarez@dcegi.ulpgc.es (J.P. Suárez).

0378-4754/$36.00 © 2011 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.matcom.2011.07.003

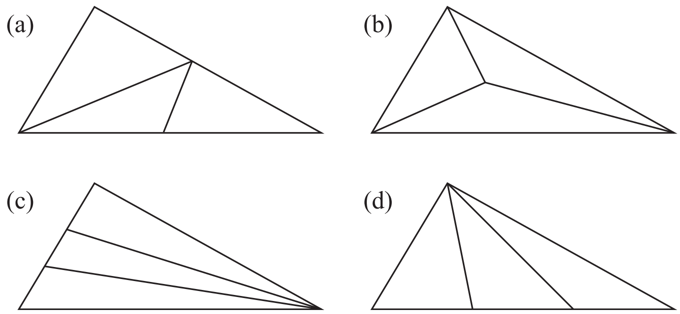


Fig. 1. (a) 3T division based on 2 edge bisections, (b) 3T division by an interior node, (c) 3T division by trisecting the shortest edge, and (d) 3T division by trisecting the longest edge (LE-trisection or 3T-LE).

In summary, longest-edge bisection offers familiar and desirable properties for mesh refinement: (i) non-degeneracy of the interior angles, (ii) finely shaped and quality meshes and (iii) smoothness condition.

However, we explore in this paper an alternative longest-edge sub-division scheme for triangles. Fig. 1 shows some of the possibilities for partitioning a triangle in three sub-triangles. All these patterns are known as *triangle trisection* in the literature [3,5], with Fig. 1(d) representing a trisection based on the longest edge, called *LE-trisection* or *3T-LE*.

We have recently proved in [16] that, for a given initial triangle t with the smallest interior angle $\tau > 0$, the LE-trisection of t produces three new triangles t_i, $i = 1, 2, 3$ such that any smallest interior angle of t_i verifies $\tau_1 \geq \tau/c_1$, where $c_1 = (\pi/3)/(\arctan(\sqrt{3}/5)) \approx 3.1403$. Moreover, there is empirical evidence with respect to the non-degeneracy of meshes obtained by iterative application of the LE-trisection and it has been shown that $\tau_n \geq \tau/6.7052025350$, independent of the value of n [16].

There are two scenarios in the case of bisection and trisection of triangles: algorithms producing conforming (no hanging nodes) triangulations, and those which occasionally yield said hanging nodes. For example, if in a given triangulation, all the simplices are bisected (trisected) simultaneously, the aforementioned hanging nodes may appear, see Fig. 2. The appearance of hanging nodes is a certain disadvantage especially in such case as the use for the finite element discretisations is designed over such partitions. They also cause surface discontinuities if meshes are applied for geometric representations. In order to ensure conformity and avoid such hanging nodes, continued longest-edge sub-division of non-conforming triangles may be considered. However, the extent of the subsidiary-induced refinement due to conformity is an issue of practical concern since it has been noted that refinement of a single cell can propagate LE subdivisions to the boundary [22,23].

Several local and global refinement algorithms based on bisection have been proposed in the existing literature [10,12,17,18]. The refinement is said to be *local* if the partition is carried out on sub-set of triangles, producing so-called adaptive refined meshes. *Global*, also known as *uniform*, refinement concerns the partition of all the triangles in a mesh.

In the context of local refinement algorithms based on the bisection of the edges, an alternative to the longest-edge based algorithms is the *red-green* approach, both in 2D and in 3D [2,4]. In 2D, the conformity of the mesh is ensured

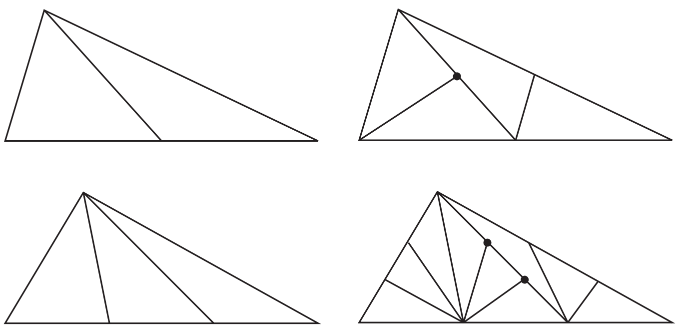


Fig. 2. Hanging nodes may appear when bisecting (or) trisecting all the triangles in a given triangulation by the longest edge.

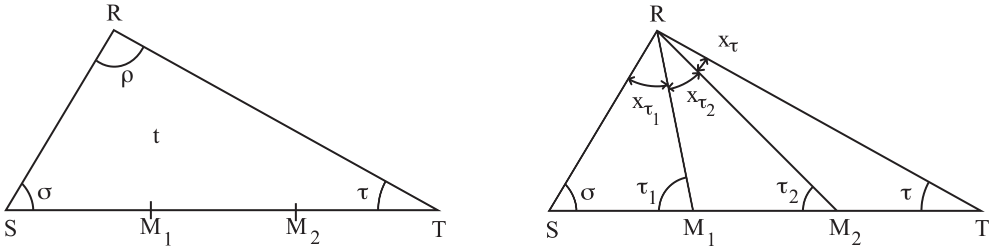


Fig. 3. Longest-edge trisection of triangle ΔRST.

by joining the hanging nodes with the opposite vertex. Partition of the non-conforming triangles is not by the longest edge, in general, in the red-green approach. In this strategy, the refinement does not extend by conformity.

In this paper, we introduce a local refinement algorithm for conforming triangle meshes based on LE-trisection.

We test the algorithm to cope with the triangle refinement problem [20]: given a valid triangulation of a poligonal region, construct a locally refined triangulation with triangles of a prescribed size in a refinement region *R*, such that the smallest angle is bounded.

A cost analysis of the algorithm is included that focuses on two critical aspects: (i) the number of inserted points to solve the triangle refinement problem, and (ii) the extension of the refinement (propagation) inherent to longest-edge based refinement. Similarly, as proved for the longest-edge bisection in [20], we give a boundary for the number *N* of points inserted in *R* to obtain triangles of a prescribed size by the longest-edge trisection. We also provide numerical evidence showing that the average propagation zone for each triangle is progressively reduced at each uniform refinement stage, and asymptotically, approaches six neighboring triangles.

Our additional contribution in this paper is to affirm that said longest-edge trisection scheme, similar to longest-edge bisection, offers fine quality properties. The proposed algorithm, thus, constitutes a valid strategy to refine triangular meshes.

2. The longest-edge trisection: definition, notations and properties

Here, we study the trisection of a triangle by its longest-edge. This partition is the counterpart of familiar longest-edge bisection and presents similar properties.

Definition 1. The longest-edge bisection of a triangle *t* is obtained by joining the midpoint of the longest-edge of *t* with the opposite vertex. We shall also refer to this partition as 2T-LE (two triangles by the longest edge).

Definition 2. The longest-edge trisection of a triangle *t* is obtained by joining the two equally spaced points of the longest-edge of *t* with the opposite vertex. We shall also refer to this partition as 3T-LE (three triangles by the longest edge).

The fractal appearance of iteratively refined meshes in the neighborhood of a critical point is clear in the case of LE-bisection local refinement but not so clear for the 3T-LE approach. However, the concept of a stable molecule associated to a node *P* as in [13,20] is highly useful.

Definition 3. For any conforming triangulation τ and any vertex *P* of τ, the stable molecule associated with vertex *P* is the partition of the plane around vertex *P*, induced by the use of some triangle refinement scheme of each triangle around the vertex *P*, such that further refinements of the triangles around *P* do not change the number of triangles sharing *P*.

Definition 4. For any conforming triangulation τ and any vertex *P* of τ, we shall call the size of the stable molecule associated with *P*, the number of straight-lined segments that define said molecule. Furthermore, we shall call the stable molecular size of the triangulation τ the maximum size of the set of stable molecules associated both with the vertices of τ, and with the vertices created by the refinement of τ.

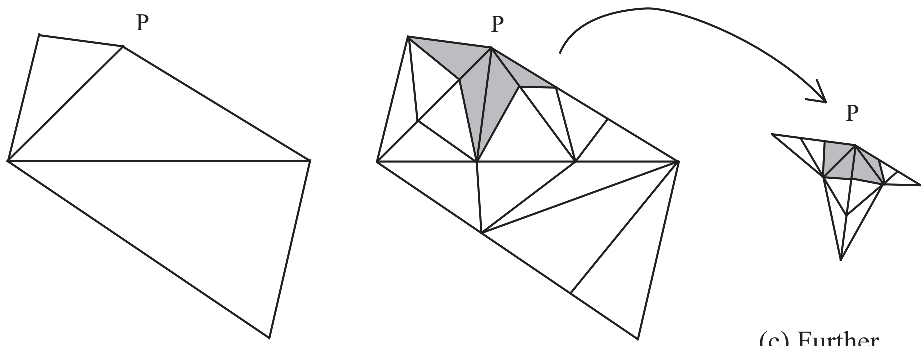


Fig. 4. Stable molecule at point P by 3T-LE local refinement around P.

In the same procedure as for longest-edge bisection local refinement algorithm, the following theorems hold for the counterpart longest-edge trisection:

Theorem 5. *Given any conforming triangulation τ, for any vertex P of τ, the arbitrary iterative use of the 3T-LE refinement algorithm to refine the triangulation around the vertex P, divides the angles converging in P in a finite number of k parts.*

Theorem 6. *Let τ be any conforming triangulation and consider any vertex P of τ. The use of the 3T-LE refinement algorithm to refine the triangulation around the vertex P produces triangulations that have the following characteristics:*

(a) *After a finite number of iterations, the algorithm produces a triangulation τ^* such that the stable molecule associated with vertex P is obtained.*
(b) *The successive iterations of the algorithm do not partition the angles of the stable molecule, but only introduce a set of new vertices distributed in geometric progression along the sides of the stable molecule of P.*

Proof. Note that by using longest-edge based algorithms for refining, the stable molecule with vertex P is achieved once the opposite edge to vertex P is no longer than the longest-edge of all the triangles sharing P. An example of the stable molecule with vertex P is shown in Fig. 4. □

2.1. The non-degeneracy property of the longest-edge trisection partition

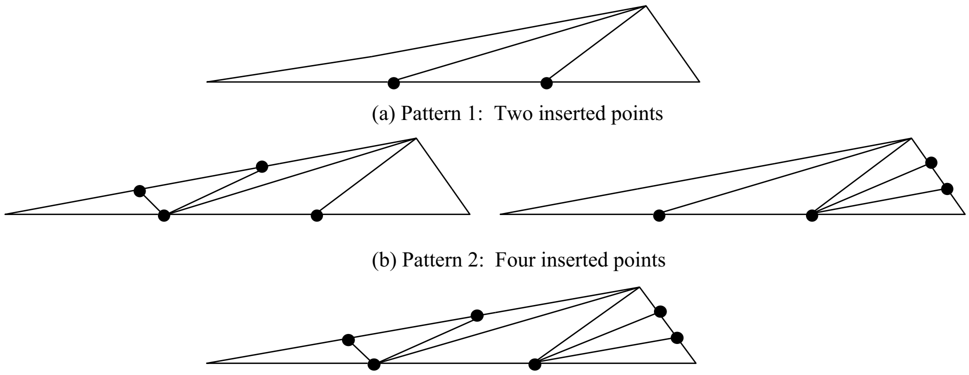
Recently, it has been proven that the iterative application of the longest-edge trisection of triangles is non-degenerate[16]. Some of these results are given below:

Theorem 7. *Let τ, σ, ρ represent the three angles in increasing order of the triangle t. Let $\tau \le \pi/3$ and $\rho = \sigma = \pi/2 - \tau/2$. Then, the minimum interior angle of the three triangles obtained by the LE-trisection of t into three new triangles, say x_τ as in Fig. 3, satisfies*

$$\tan x_\tau = \frac{\sin\tau}{3-\cos\tau} \ge \frac{\tan\tau}{3.1403}.$$

Corollary 8. *If t_i is a triangle obtained by the LE-trisection of t and θ is an interior angle of t_i, then $\theta \ge \lambda/c_1$, where* $c_1 = (\pi/3)(\arctan(\sqrt{3}/5)) \approx 3.1403$.

Notice that the previous bound is sharp. The equality holds for an initial equilateral triangle. In addition, in Ref. [16], experimental evidence of the bound $\tau_n \ge \tau/6.7052025350$ is given, where τ_n is the minimum interior angle after n iterative trisections. The reduction factor in the minimum angle, 6.7052025350 is attained again in the case of the initial equilateral triangle.



(a) Pattern 1: Two inserted points

(b) Pattern 2: Four inserted points

(c) Pattern 3: Six inserted points

Fig. 5. The partial partitions of a triangle depending on the number of edges subdivided for the 3T-LE local refinement.

As was given in the case of the longest-edge bisection based algorithms [19], for the longest-edge trisection algorithms, the following definition also applies.

Definition 9. $T(\alpha, \delta)$ is a family of (α, δ)-irregular triangulation when for any $\tau \in T(\alpha, \delta)$ the following three conditions hold:

(i) Conformity, that is, the intersection of two non-disjointed non-identical triangles is either a common vertex or a common edge.
(ii) Smoothness, for any pair of edge-adjacent triangles $t_1, t_2 \in \tau$ with respective diameters (longest-edges) h_1, h_2:

$$\frac{\min(h_1, h_2)}{\max(h_1, h_2)} \geq \delta > 0$$

(iii) Min-angle boundedness, if α_t is the minimum angle of $t \in \tau$, then $\min_{t \in \tau} \alpha_t \geq \alpha > 0$.

where α and δ are constants.

It should be noted that the chain of triangulations obtained by iterative application of longest-edge trisection verifies said conditions of previous definition. Although here we do not provide mathematical proof of condition (ii) or the smoothness condition, we shall later offer experimental evidence based on the empirical results reported in Section 4.

3. Local refinement algorithms based on LE-trisection and cost analysis

Following the indications offered above for the LE-trisection of a triangle, a refinement algorithm has been devised. The main features of such algorithm are: first, the triangles to be refined are trisected by their longest-edge. Then, in order to assure the conformity of the new arising mesh, we use one of the partitions given in Fig. 5. It can be observed that the 3T-LE partition uses two interior, equally spaced points per edge sub-divided. Due to the different possibilities of sub-divided edges, and respecting the longest edge subdivision layout, the partition patterns listed in Fig. 5 need to be used to preserve the conformity condition (in other words, no hanging nodes).

In general, the refinement of triangular meshes involves two main tasks. The first is the sub-division of the target triangles, and the second is the propagation to successive neighboring triangles to preserve conformity. If the refinement is uniform then all the triangles in the mesh are refined and no propagation is induced on the mesh, but hanging nodes have to be avoided by applying the subdivision patterns given in Fig. 5. Refinement can also be used locally for a small group of triangles in a mesh and, in this case, a neighboring triangle refinement (propagation) is induced. In the LE schemes, the longest edge of the triangle t is first trisected (or bisected) to form three (or two) sub-triangles. Then, to preserve mesh conformity, the refinement is iterated with the associated-edge neighbor using the longest-edge strategy until the propagating path terminates. In both the 2T-LE and the 3T-LE scheme, LE refinement propagates through the neighboring triangle to t via the longest-edge of t. In [22,23] a study is given of the propagation and efficiency in practical adaptive scenarios.

The following algorithm describes the process of local refinement by the 3T-LE scheme, where the targeted triangles to be refined are given in the input list T together with the entire triangle mesh τ. The algorithm is devised to avoid hanging nodes, which, in addition, increases smoothness over the mesh, Step 2.

Step 3 subdivides those triangles with some inserted points, according to the triangle patterns given in Fig. 5. In other words, Step 3 constructs the triangles and updates the newly generated mesh τ. It can be easily noted that the algorithm is linearly complex with respect to the number of nodes.

Algorithm 3T-LE Local Refinement (τ, T)
/* Input: τ mesh, T List of triangles to be refined
/* Output: τ mesh
1. Insert 2 points in the LE of each triangle of T
2. **While** non-conforming triangles in τ**do**
  Insert two points in the LE of non-conforming triangles
  Update mesh points in τ
  End
3. **For each** triangle t with inserted points **do**
  Perform a partition of t:
    Case of 2 points: Subdivide t by Pattern 1
    Case of 4 points: Subdivide t by Pattern 2
    Case of 6 points: Subdivide t by Pattern 3
  End
Update mesh triangles in τ
End

The data structure used to store the mesh τ consists of two arrays. The node array represents points (x, y) of nodes. The mesh triangles are represented in another matrix. Each mesh triangle is described by indices to its three mesh nodes. It is worth noting that the basic adjacency relationship used in the refinement algorithm is determined by the longest edge of each triangle. Hence, we store the nodes surrounding the longest edge in the two first array positions. This is consistently maintained over each refinement step so that, each time a neighboring triangle is requested, the triangle array provides the information in constant time. To reduce storage, rather than representing edges in a separate array, these are accessed by the relative nodes of the triangle array.

The cost analysis of the previous refinement algorithm is now taken into consideration. This study is elaborated on the basis of two points of views: (i) the number of inserted points used to solve the triangle refinement problem and (ii) the propagation behavior of the secondary refinements used to preserve conformity.

The triangulation refinement problem is formulated as follows: given a valid, non-degenerate triangulation of a polygonal region, construct a locally refined triangulation, with triangles of prescribed size in a refinement region R, such that the smallest (or the largest) angle is bounded.

Theorem 10 ([20]). *Consider that the longest-edge trisection refinement algorithm is used to solve the point triangulation refinement problem around any vertex P* (*a stable molecule of size k*) *of any conforming triangulation* τ. *Then, once the stable molecule is obtained, the iterative refinement around vertex P is produced until triangles with a diameter* (*longest side*) *h are obtained, introducing K vertices, where*

$$K < 2k\left[1 + \log_3\left(3 + \frac{L}{h}\right)\right]$$

k is a constant and L is the length of the longest side of the stable molecule of P.

Proof. Analogous to that in Rivara and Vemere [20] Along each side $P\overline{P}$ of the stable molecule of P, the algorithm introduces a sequence of points $P_{1a}, P_{1b}, P_{2a}, P_{2a} \ldots, P_{ja}, P_{jb}$ such that P_{1a}, P_{1a} are the two equally spaced points of side $P\overline{P}$ and P_{ja}, P_{jb} are the two equally spaced points of side PP_{j-1}. □

Theorem 11 ([20]). *The time cost of solving the point triangulation refinement problem using 3T-LE local refinement algorithm is* $O(K)$ *where* K *is bounded as in* Theorem 10.

Theorem 12 ([20]). *Let* τ_0 *be any conforming coarse triangulation (of stable molecular size* k*) of any bounded polygonal region* Ω*. Then for any circular refinement sub-region* C*, with radius* r*, with* $C \in \Omega$*, the use of the longest-edge trisection refinement algorithm over* C*, until obtaining triangles of diameter* $\tilde{h}$ *inside* C*, asymptotically introduces* N_i *points inside* C *and* N_o *points outside* C*, such that* N_i *and* N_o *are bounded as follows:*

$$N_i < k[1 + \pi(n^2 + n)]$$
$$N_0 < 2\pi kn[1 + \log_3(1 + L\tilde{h}).]$$

where $n = 2r/\tilde{h}$ *and* L *is the maximum size of all possible molecules of* Ω.

Theorem 13 ([20]). *Under the conditions of the previous theorem, and when* Ω *is a bounded region, the number of points inserted outside the refinement region* R *can be bounded as follows:*

$$N_o < K_1 n + K_2 n \log_3 n,$$

where K_1 *and* K_2 *are constants.*

In Refs. [22,23], we discuss longest-edge propagation studies. Let us consider a target triangle t to be refined by the LE-trisection partition. The conformity of the resulting mesh with adjacent elements can be preserved by continuing refinement to the neighbors, thereby inducing refinement propagation. The following definition is useful in the study of the refinement propagation:

Definition 14. The longest edge propagation path (LEPP) of a triangle t_0 is the ordered finite list of all the adjacent triangles LEPP(t_0) = $\{t_0, t_1, \ldots, t_n\}$ such that t_i is the longest-edge neighboring triangle of t_{i-1}.

The refinement propagation path finishes either in a terminating triangle pair that share their respective longest edges or at the mesh boundary.

The refinement propagation induced by the LE-trisection of a triangle extends along its longest edge, as this is the edge where new points are inserted, see pattern 1 Fig. 5(a). However, the medium and/or shortest edges may be affected by propagation if patterns 2 or 3 are employed in the refinement, see Fig. 5(b) and (c). Three propagation paths then can be identified emanating from a given triangle t. It should be noted that LEPP only depends on the underlying mesh rather than on the refinement scheme.

Two metrics denoted $M1$ and $M2$ are used to quantify propagation.

Definition 15. Let $M1(t)$ be the total number of original triangles in the propagation paths emanating from t and $M2(t)$ be the longest individual edge path from t.

Fig. 6 illustrates a situation of mesh propagation when the refining triangle is t. In this example, $M1(t) = 10$ and $M2(t) = 5$ and the propagation zone is coloured. It can be remarked that $M1$ and $M2$ measure the effect of secondary propagation in a refinement strategy, especially in the case of longest-edge based algorithms as shown in Fig. 6. In practice, however, we are not only interested in the individual triangles. Instead, representative values are preferred with the $M1$ and $M2$ mesh averages considered in this study.

It is worth noting that $M1$ and $M2$ can be applied to general triangled meshes. Nevertheless when calculating mesh averages after the application of iterative (local or uniform) refinement, the resulting values are dependent on the sub-division scheme, as triangle patterns differ from LE-bisection to LE-trisection.

The numerical tests in Section 4 verify the theoretical predictions made here.

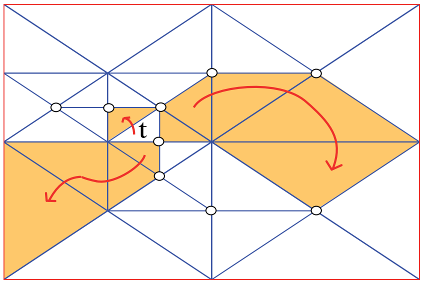


Fig. 6. An example of the propagation zone (shaded) for a triangle t. $M1(t) = 10$ and $M2(t) = 5$.

Table 1
Example 1. Regular rectangular triangulation with a sub-region of a rectangle at the base. Reported inserted points.

$h = 0.3/3^i$	N_i	N_o	8×2^{3i}	$N_i^{1/2}\log_3 N_i^{1/2}$
$i = 1$	62	26	64	29
$i = 2$	502	130	512	126
$i = 3$	4470	440	4096	511
$i = 4$	39,316	1432	32,768	1909
$i = 5$	350,262	4324	262,144	6877

4. Empirical results and tests

In this section, numerical results are offered to illustrate the practical behavior of the algorithm in concordance with the previously given cost analysis.

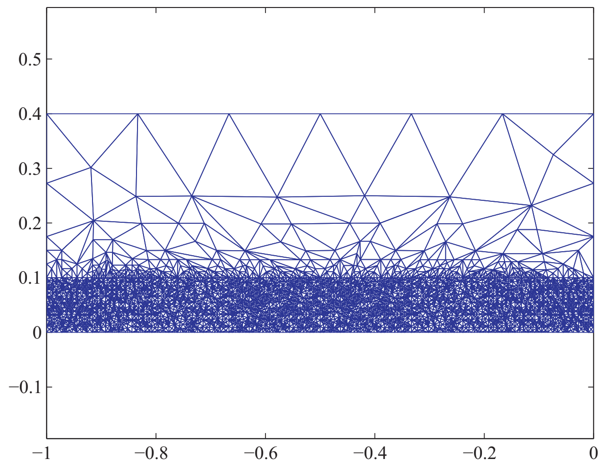
To perform the numerical studies, we have considered two triangle meshes to solve the triangulation refinement problem. The first example is a rectangular domain Ω where an interior sub-region R is targeted for refinement. This sub-region R is a sub-rectangle located along the base of Ω. We construct a locally refined triangulation until the triangles of prescribed size $(0.3)/(3^i)$ $(i = 1, 2 \ldots)$ are obtained. The second example is a square domain with an interior square-shaped sub-region R at its center. The meshes are refined in both examples until the prescribed sizes are achieved for $i = 3$ as shown in Fig. 7(a) and (b).

For these two examples, we first compare the number of vertices introduced inside R with the expected values given by the equations of Theorems 12 and 13, see Tables 1 and 2. Effectively, in practice, it can be observed how the number of inserted points inside (N_i) and outside (N_o) is asymptotically bounded by functions $C_i \times 2^{3i}$ ($C_1 = 8$ and $C_2 = 4$) and $N_i^{1/2}\log_3 N_i^{1/2}$.

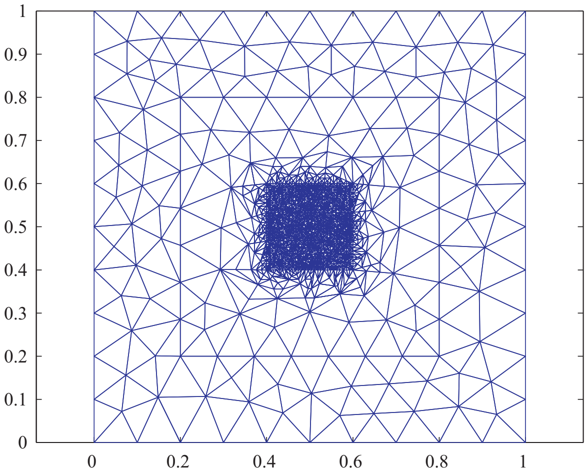
In order to understand the effect of propagation in LE-trisection refinement, we carried out another numerical experiment with the same triangle meshes. As we are interested in the metrics for all the triangles in a mesh, we will focus on the average of $M1$ and $M2$, and so, we obtain a pair of values $M1$, $M2$ for each studied mesh. Tables 3 and 4 report the metrics for the studied meshes. In average, the evolution of $M1$ approaches approximately 6. In other words, the propagation by the LE-trisection may extend up to 6 neighboring triangles, with an associated largest path extending up to 3 triangles. In fact, these values bound the secondary refinements induced by the longest-edge based algorithms.

Table 2
Example 2. Square triangulation with an interior square-shaped sub-region at its center. Reported inserted points.

$h = 0.3/3^i$	N_i	N_o	4×2^{3i}	$N_i^{1/2}\log_3 N_i^{1/2}$
$i = 1$	16	0	32	10
$i = 2$	216	58	256	71
$i = 3$	1900	250	2048	299
$i = 4$	17,228	850	16,384	1165
$i = 5$	154,372	2618	131,072	4272



(a) Example 1. Local refinement inside region (-1,0)-(0,0.1). Final mesh with 4778 points and 9322 triangles at step i=3



(b) Example 2.Local refinement inside region (0.4,0.4)-(0.6,0.6). Final mesh with 2238 points and 4434 triangles at step i=3

Fig. 7. Example of local refinement based on 3T-LE partition. Prescribed size $(0.3)/(3^i)$.

Said values coincide with those obtained for another refinement method, that is, the four triangle longest edge (4T-LE) method, in which $M1$ and $M2$ approach 5 and 2 respectively, see [22].

To complete the numerical test, we tested the mesh's smooth condition for the examples. In other words, for any pair of edge-adjacent triangles $t_1, t_2 \in \tau$ with respective diameters (longest-edges) h_1, h_2: $(\min\ (h_1, h_2))/(\max\ (h_1, h_2)) \geq 0$.

It is interesting to note that int the local refinement examples, we generate an unbalanced refinement tree with, typically, some elements at root level zero, some at the next level one, and so on to the 'leaves' at the finest refinement

Table 3
Example 1. Regular rectangular triangulation with a sub-region of a rectangle at the base. Propagation results.

$h = 0.3/3^i$	$Av(M1)$	$Av(M2)$	$\frac{\min(h_1,h_2)}{\max(h_1,h_2)}$
$i = 1$	6.2068	3.0121	0.3484
$i = 2$	6.2145	3.0492	0.3484
$i = 3$	6.2396	3.1701	0.2676
$i = 4$	6.2275	3.1698	0.1376
$i = 5$	6.2101	3.1097	0.1198

Table 4
Example 2. Square triangulation with an interior square-shaped sub-region at the center. Propagation results.

$h = 0.3/3^i$	$Av(M1)$	$Av(M2)$	$\frac{\min(h_1,h_2)}{\max(h_1,h_2)}$
$i = 1$	5.7777	2.8082	0.5303
$i = 2$	5.8217	2.8281	0.2835
$i = 3$	5.8902	2.8819	0.2835
$i = 4$	5.9821	2.9012	0.1694
$i = 5$	6.0128	3.0172	0.1420

Table 5
LE-bisection and LE-trisection comparison.

	LE-bisection	LE-trisection
Triangles generation	2	3
Complexity, K = inserted points	Linear in K	Linear in K
Boundeness of interior small angle	$\tau/2$	$\tau/6.7052$
Propagation. Inserted points	$N_o < K_1 n + K_2 n \log_2 n$	$N_o < K_1 n + K_2 n \log_3 n$
Propagation. Inserted triangles	$Av(M1) \simeq 5$; $Av(M2) \simeq 2$	$Av(M1) \simeq 6$; $Av(M2) \simeq 3$
Smooth condition $\frac{\min(h_1,h_2)}{\max(h_1,h_2)} \geq \delta$	$\delta \simeq 0$	$\delta \simeq 0$

level. This naturally destroys the 'regularity' of the mesh so one might expect to encounter relatively low values for the smooth condition. Column four in Tables 3 and 4 shows the values for (min (h_1, h_2))/(max (h_1, h_2)).

To end this section, Table 5 gives a summary of the main features treated in the paper for the LE-trisection refinement algorithm and its counterpart LE-bisection.

5. Final remarks

A new algorithm has been introduced here for the local refinement of triangular meshes, using trisection based on the longest edge. This trisection method can be regarded as a natural extension of the well-known longest-edge bisection of triangles.

We have described the algorithm and provided a study of the efficiency and cost analysis for the triangulation refinement problem. The longest-edge trisection scheme offers a valid strategy to refine triangular meshes, on account of several of its properties:

(i) The non-degenerate nature of the partition.
(ii) The existence of bound for the number of inserted points.
(ii) The linear performance of the algorithm, at the number of nodes.
(iv) The existence of bounds for the propagation refinement.

It is remarkable how LE-trisection replicates the characteristics of the better known LE-bisection. However, LE-trisection of triangles may be considered to be a limit case of n-section. It is not difficult to see, at least numerically, how, for example, longest-edge 4-section produces degenerate triangulation and, thus, is unsuited to numerical computations.

This alone would be sufficient reason for using a refinement algorithm based on LE-trisection, such as we have presented here.

Future projected work includes finding mathematical proof for the smoothness condition of the algorithm based on the longest-edge. In addition, a problem to be addressed is the number of dissimilarity classes appearing in the partition of triangles, as has been previously studied in the case of LE and 4T-LE, [14]. Any forthcoming work should also cover a full analysis of the different edge connections arising in LE-trisection that may improve the quality of the shape of the partition. Evidence of superior quality can be given through the use of shortest-edge connection and Delaunay criterion.

Acknowledgements

This work has been supported in part by the CICYT Project MTM2008-05866-C03-02/MTM of the Ministerio de Educación y Ciencia of Spain and by the Spanish 'Ministerio de Asuntos Exteriores y de Cooperación', Project A/025433/09.

References

[2] R.E. Bank, PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide 6.0, SIAM, Philadelphia, PA, 1990.
[3] J. Brandts, S. Korotov, M. Krizek, Simplicial finite elements in higher dimensions, Appl. Math. 52 (3) (2007) 251–265.
[4] J.-H. Choi, K.-R. Byun, H.-J. Hwang, Quality-improved local refinement of tetrahedral mesh based on element-wise refinement switching, J. Comput. Phys. 192 (1) (2003) 312–324.
[5] J. Danciger, S.L. Devadoss, D. Sheehy, Compatible triangulations and point partitions by series-triangular graphs, Comput. Geom. 34 (3) (2006) 195–202.
[7] M.T. Jones, P.E. Plassmann, Computational results for parallel unstructured mesh computations, Comput. Syst. Eng. 5 (4–6) (1994) 297–309.
[8] M.T. Jones, P.E. Plassmann, Adaptive refinement of unstructured finite-element meshes, Finite Elem. Anal. Des. 25 (1–2) (1997) 41–60.
[9] K. Kashiyama, T. Okada, Automatic mesh generation method for shallow water flow analysis, Int. J. Numer. Meth. Fluids 15 (1992) 1037–1057.
[10] S. Korotov, M. Krizek, A. Kropac, Strong regularity of a family of face-to-face partitions generated by the longest-edge bisection algorithm, Comput. Math. Math. Phys. 48 (9) (2008) 1687–1698.
[11] A. Márquez, A. Moreno-González, A. Plaza, J.P. Suárez, The 7-triangle longest-side partition of triangles and mesh quality improvement, Finite Elem. Anal. Des. 44 (12–13) (2008) 748–758.
[12] A. Plaza, G.F. Carey, Local refinement of simplicial grids based on the skeleton, Appl. Numer. Math. 32 (2) (2000) 195–218.
[13] A. Plaza, J.P. Suárez, M.A. Padrón, A non-degeneracy study of the 8-tetrahedra longest-edge partition, Appl. Numer. Math. 55 (4) (2005) 458–472.
[14] A. Plaza, J.P. Suárez, M.A. Padrón, S. Falcón, D. Amieiro, Mesh quality improvement and other properties in the four-triangle longest-edge partition, Comput. Aided Geomet. Des. 21 (4) (2004) 353–369.
[15] A. Plaza, A. Márquez, A. Moreno-González, J.P. Suárez, Local refinement based on the 7-triangle longest-edge partition, Math. Comput. Simul. 79 (8) (2009) 2444–2457.
[16] A. Plaza, S. Falcón, J.P. Suárez, On the non-degeneracy property of the longest-edge trisection of triangles, Appl. Math. Comput. 216 (3) (2010) 862–869.
[17] M.C. Rivara, Mesh refinement processes based on the generalized bisection of simplices, SIAM J. Numer. Anal. 21 (1984) 604–613.
[18] M.C. Rivara, Algorithms for refining triangular grids suitable for adaptive and multigrid techniques, Int. J. Numer. Meth. Eng. 2 (1984) 745–756.
[19] M.C. Rivara, Selective refinement/derefinement algorithms for sequences of nested triangulations, Int. J. Numer. Meth. Eng. 28 (12) (1989) 889–2906.
[20] M.C. Rivara, M. Vemere, Cost analysis of the longest-side (triangle bisection) refinement algorithm for triangulations, Int. J. Numer. Meth. Eng. 12 (3–4) (1996) 224–234.
[21] I.G. Rosenberg, F. Stenger, A lower bound on the angles of triangles constructed by bisecting the longest-side, Math. Comput. 29 (130) (1975) 390–395.
[22] J.P. Suárez, A. Plaza, G.F. Carey, The propagation problem in longest-edge refinement, Finite Elem. Anal. Des. 42 (2) (2005) 130–151.
[23] J.P. Suárez, A. Plaza, G.F. Carey, Propagation of longest-edge mesh patterns in local adaptive refinement, Commun. Numer. Meth. Eng. 24 (7) (2008) 543–553.
[25] O.C. Zienkiewicz, J.Z. Zhu, Adaptivity and mesh generation, Int. J. Numer. Meth. Eng. 32 (4) (1991) 783–810.