THE 4-TRIANGLES LONGEST-SIDE PARTITION OF TRIANGLES AND LINEAR REFINEMENT ALGORITHMS

MARÍA-CECILIA RIVARA AND GABRIEL IRIBARREN

ABSTRACT. In this paper we study geometrical properties of the iterative 4-triangles longest-side partition of triangles (and of a 3-triangles partition), as well as practical algorithms based on these partitions, used both directly for the triangulation refinement problem, and as a basis for point insertion strategies in Delaunay refinement algorithms. The 4-triangles partition is obtained by joining the midpoint of the longest side with the opposite vertex and the midpoints of the two remaining sides. By means of simple geometrical arguments we show that the iterative partition of obtuse triangles systematically improves the triangles (while they remain obtuse) in the following sense: the sequence of smallest angles monotonically increases while the sequence of largest angles monotonically decreases in an amount (at least) equal to the smallest angle of each iteration. This allows us to improve the known bound on the smallest angle (without making use of previous results), and to obtain a better a priori bound on the number of similarly distinct triangles, as a function of the geometry of the initial triangle. Numerical evidence, showing that the practical behavior of the 4-triangles partition is in complete agreement with this theory, is included. A 4-triangles refinement algorithm is also discussed and illustrated. Furthermore, we show that the time cost of the algorithm is linear independently of the size of the triangulation.

1. INTRODUCTION: THE TRIANGULATION REFINEMENT PROBLEM

During the last 10 years an increasing effort has been devoted to the development of adaptive and/or multigrid finite element methods [1, 4, 13, 14]. In this context, adaptivity of the grid is the central feature that relieves the user of critical decisions and allows use of all flexibility of the finite element method for getting a minimum number of vertices. In this sense, the generation of the discretization (triangulation) should not be a first separate step of the finite element solution process, but a dynamic adaptive process. In order to deal with singular solutions, the capability for managing local refinement of the discretization is indispensable, and mesh refinement algorithms that maintain the nondegeneracy of the elements and the conformity and smoothness of the grid are certainly desirable. Conformity refers to the requirement that the intersection of nondisjoint triangles is either a common vertex or a common side, and the smoothness condition states that the transition between small and large elements should be smooth.

The algorithms based on the longest-side bisection of triangles were developed to deal with this specific question [11, 12, 13, 14, 15, 16]. They guarantee the

construction of good-quality irregular and nested triangulations. This is due both to the boundedness condition on the small angles of the triangles generated, and to the natural refinement propagation strategy outside the target refinement area (which ensure the construction of smooth irregular triangulations). Furthermore, in spite of this inherent propagation strategy, the time cost of this kind of algorithms is linear, independently of the size of the triangulation [19].

The algorithms have been successfully used to implement adaptive (and adaptive multigrid) software in two dimensions [8, 10, 13, 14], and have been generalized and used in the 3-dimensional context [5, 7, 18]. Derefinement algorithms, suitable to refine and derefine the mesh in the course of adaptive computations, such as needed in complex time-dependent problems (which require moving regions of refinement) have also been developed [16]. Recently, the 2D algorithms have been used both for surface fitting, by means of automatic and adaptive Powell-Sabin Splines over triangulations [3], and to develop parallel refinement algorithms for complex applications [6, 22]. Longest-side based partitions of triangles have also been the basis of point insertion strategies that ensure the construction of good-quality triangulations, for local Delaunay refinement algorithms [2, 10].

It should be pointed out here that the triangulation refinement problem [17, 19] is essentially different than the classical triangulation problem in the following sense: instead of having a fixed set of points to be triangulated, one has the freedom to choose the points to be added in order to construct a mesh with a desired resolution. The construction of the mesh is dynamically performed. Furthermore, it is possible to exploit the existence of the reference triangulation (constructed for instance by means of the Delaunay algorithm) in order to reduce the computational cost to construct the refined mesh.

By introducing a subregion R to define the refinement area (usually a changing set of triangles in the adaptive context), a condition over the diameter (longest-side) of the triangles, and a resolution parameter ε to fix the desired resolution, the *Triangulation Refinement Problem* can be stated as follows: *given an acceptable triangulation of a polygonal region Ω, construct a locally refined triangulation such that the diameter of the triangles that intersect the refinement area R be less than ε.*

In the context of the triangulation refinement problem, the algorithms based on the longest-side bisection of triangles benefit from a natural point insertion strategy (longest-side midpoint insertion) which together with the boundedness condition on the small angles generated and their inherent propagation strategy, guarantee the construction of good-quality smooth irregular triangulations. On the other hand, even when the Delaunay algorithm constructs the optimal mesh (the most equilateral one) for the classical triangulation problem (given a fixed set of vertices), this algorithm does not have a natural point insertion strategy that guarantees the construction of good-quality irregular triangulations when the algorithm is iteratively used to refine a mesh. Simple experiments with the "intuitive" centroid insertion idea show that this strategy rapidly deteriorates the quality of the triangulations (especially along the boundaries), even when global refinement is performed. To illustrate this fact, Figure 1 shows three different steps of the global iterative Delaunay refinement of an initial simple triangulation (a single triangle). The centroid of each triangle of the current triangulation was iteratively added as a point to be inserted in each refinement step. After a few refinement steps, a quite unacceptable triangulation is obtained (triangulation (c) in Figure 1).
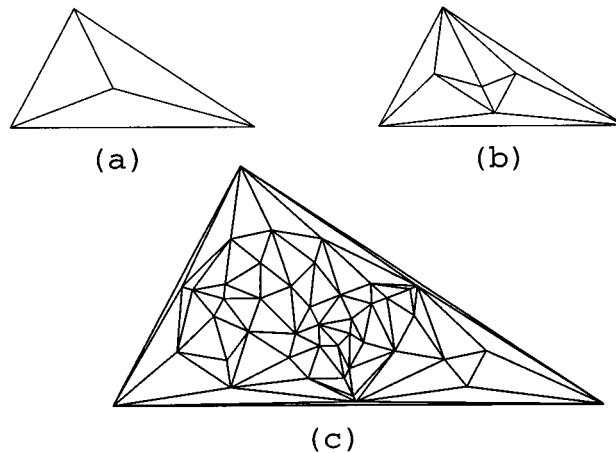
(a)   (b)

(c)

FIGURE 1

Briefly, in order to obtain robust local Delaunay refinement software, it is necessary to make use of a companion methodology that automatically introduces a point distribution that (after arbitrary steps of local refinement) guarantees the construction of smooth nondegenerate triangulations. As was pointed out previously, the longest-side based partition of triangles has been the basis to develop mixed longest-side bisection Delaunay refinement algorithms [2, 10], which ensure the construction of such triangulations. Thus, for instance, Penman and Grieve [10] use a heuristic method that chooses between either a longest-side midpoint insertion strategy, or a centroid insertion strategy, depending on the shape of the triangles involved. On the other hand, Bova and Carey [2] have developed a more sophisticated method that takes advantage of the fractal properties of the longest-side bisection algorithms [2, 19] to introduce an automatic graded distribution of points. In [10], triangulations of analogous quality to those obtained with Rivara refinement algorithms have been reported.

It should be emphasized here that both the longest-side bisection refinement algorithms and the mixed (longest-side bisection) Delaunay refinement algorithms are competitive linear algorithms for dealing with the triangulation refinement problem [19]. However, pure longest-side bisection-based algorithms are the most suitable ones in two important contexts: (1) in the adaptive refinement/derefinement of triangulations such as needed in complex time-dependent problems [16]; (2) in the practical use of multigrid finite element methods over irregular meshes [4, 12, 13, 14, 15, 16] (since the algorithms guarantee the construction of nested triangulations). Finally, it should be pointed out also that 3D longest-edge bisection refinement algorithms for tetrahedral meshes have shown to be very useful in practice [5, 7, 18]. However, in this context there are no mathematical results available analogous to those of this paper or to those of references [20, 21] (which were indeed the basis to initially develop the algorithms). Notice that the 3D problem has been considered an interesting open problem in the Computational Geometry field [9].

The results we include here characterize in a geometrical sense the point distribution introduced by longest-side refinement algorithms in two dimensions, and were

found looking for new paths to deal with the 3-dimensional problem. In particular, in this paper we show that the iterative 4-triangles partition of obtuse triangles systematically improves the triangles (while they remain obtuse) in the following sense: the sequence of smallest angles monotonically increases while the sequence of largest angles monotonically decreases in an amount (at least) equal to the smallest angle of each iteration. This allows us to improve the known bound on the smallest angle, without making use of the previous result on the longest side bisection [20, 21], and to obtain a better a priori bound on the number of similarly distinct triangles as a function of the geometry of the triangle. We also include numerical evidence showing that the practical behavior of the 4-triangles partition is in complete agreement with the theory. A practical 4-triangles refinement algorithm is also introduced, discussed and illustrated. Finally, we show that the time cost of this algorithm is linear independently of the size of the triangulation.

2. The 4-triangles partition: Definitions, notations and statement of the problem

Definition 1. The longest-side partition of any triangle t is obtained by joining the midpoint of the longest side of t with the opposite vertex. The 4-triangles longest-side partition is obtaining by joining the midpoint of the longest side with the opposite vertex and the midpoint of the two remaining sides (see Figure 2(b)).

Notation. For any triangle t, its sides and angles will be respectively denoted in decreasing order $r_1 \geq r_2 \geq r_3$, and $\gamma \geq \beta \geq \alpha$. In addition, r_4 will be the new side generated by joining the midpoint of the longest side and the opposite vertex (side CD in Figure 2(b)). Furthermore, $t(\alpha, \beta, \gamma)$ will be the class of similar triangles of angles $\gamma \geq \beta \geq \alpha$. Interchangeably, t will represent an element of the class $t \in t(\alpha, \beta, \gamma)$ or the class itself.

Since the first 4-triangles longest-side partition of any triangle t_0 introduces new sides parallel to the sides of t_0, the following results hold:

Proposition 1. (a) *The first 4-triangles longest-side partition of any triangle t_0 produces two triangles similar to t_0 and two (potentially) new similar triangles t_1 (see Figure 2(b)).*

(b) *The iterative 4-triangles longest-side partition of any triangle t introduces (at most) one new distinct (up to similarity) triangle for iteration.* □

In the remainder of this paper we shall always consider the triangle t_1 (the unique distinct triangle obtained after the first 4-triangles partition of t_0) which has the midpoints of r_1 and r_2 as vertices (see Figure 3).
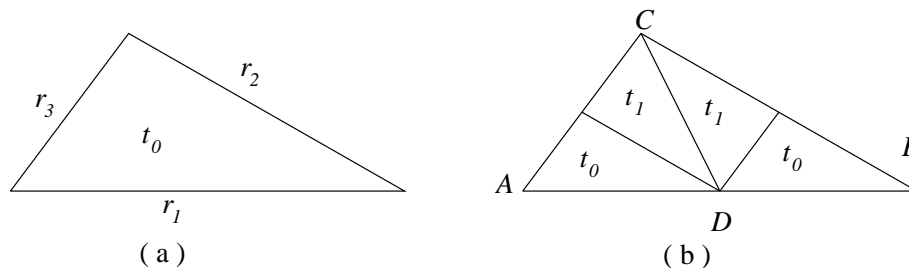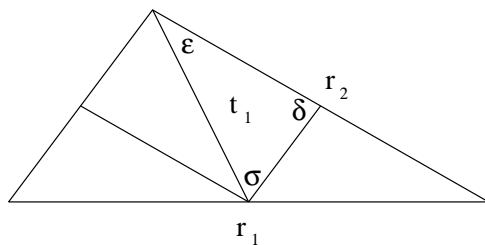

(a)   (b)

Figure 2

FIGURE 3

Notation. For the triangle t_1, we shall call ε, σ, δ the angles of t_1 having, respectively, the following vertices: the vertex of t_0 opposite to the longest side, the midpoint of r_1, and the midpoint of r_2.

The first aim of this paper is to prove that for general obtuse triangles, the iterative 4-triangles longest-side partition, produces a finite sequence of "better" triangles satisfying the properties illustrated in the following diagram until t_N becomes nonobtuse,
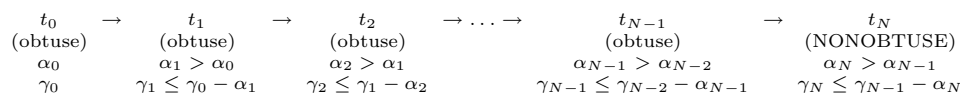
t_0	$\rightarrow$	t_1	$\rightarrow$	t_2	$\rightarrow \ldots \rightarrow$	t_{N-1}	$\rightarrow$	t_N
(obtuse)		(obtuse)		(obtuse)		(obtuse)		(NONOBTUSE)
α_0		$\alpha_1 > \alpha_0$		$\alpha_2 > \alpha_1$		$\alpha_{N-1} > \alpha_{N-2}$		$\alpha_N > \alpha_{N-1}$
γ_0		$\gamma_1 \leq \gamma_0 - \alpha_1$		$\gamma_2 \leq \gamma_1 - \alpha_2$		$\gamma_{N-1} \leq \gamma_{N-2} - \alpha_{N-1}$		$\gamma_N \leq \gamma_{N-1} - \alpha_N$

DIAGRAM 1

where α_i and γ_i are respectively the smallest and the largest angles of triangle t_i. The arrow emanating from triangle t_j to triangle t_{j+1} should be understood in the following sense: the (first) 4-triangles longest-side partition of triangle t_j produces the new (similarly distinct) triangle t_{j+1}.

Notice that, according to the diagram, the smallest angles monotonically increase, while the largest angles are clearly decreasing in an amount (at least) equal to the smallest angle of the current iteration. This allows one to establish a sharp a priori bound for N.

Furthermore, in the case of nonobtuse triangles, at most two similarly distinct triangles are generated (by iterative use of the 4-triangles longest-side partition). This means that for each nonobtuse triangle t, there exists an obtuse triangle $\tilde{t}$ such that t generates $\tilde{t}$ and $\tilde{t}$ generates t. As a consequence, the process described in the preceding diagram can only finish in one of the three ways illustrated in Diagram 2.

In Diagram 2, t_{N-1} and t_N are the last two triangles of Diagram 1. Notice that in the three cases the process finishes with a closed loop : either two triangles are generated alternately (cases 1 and 3), or the right triangle generates itself (case 2). Only in the third case can t_{N+1} be "worse" than the preceding triangles, in the sense that a triangle can be generated having a largest angle greater than the preceding ones.

(1) t_{N-1} $\rightleftharpoons$ t_N $\gamma_{N-1} + \gamma_N = \pi$

       obtuse   nonobtuse

(2) t_{N-1} $\rightarrow$ $\overset{\frown}{t_N}$ $\gamma_N = \pi/2$

       obtuse   right-angled

(3) t_{N-1} $\rightarrow$ t_N $\rightleftharpoons$ t_{N+1} $\gamma_N + \gamma_{N+1} = \pi$

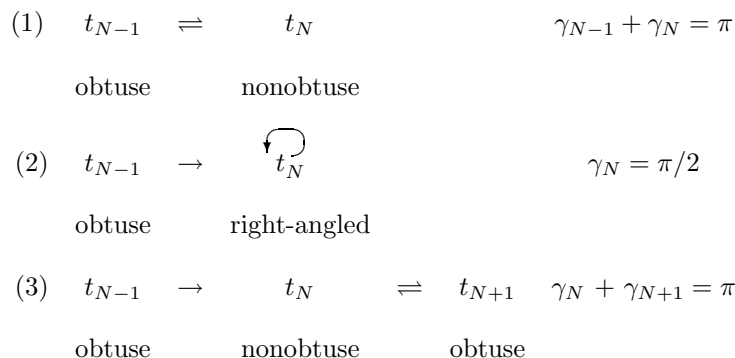   obtuse   nonobtuse   obtuse

DIAGRAM 2

3. PROPERTIES OF THE 4-TRIANGLES LONGEST-SIDE PARTITION OF NONOBTUSE TRIANGLES

The following lemma allows us to restrict the number of cases to be studied.

Lemma 1. *The first* 4*-triangles longest-side partition of any triangle* t_0 *of sides* $r_1 \geq r_2 > r_3$ *and angles* $\gamma_0 > \beta_0 > \alpha_0$ *produces a new triangle* t_1 *(of angles* $\gamma_1 \geq \beta_1 > \alpha_1$*) such that*

(a) $\gamma_1 \neq \varepsilon$,
(b) $\alpha_1 \neq \sigma$.

Proof. First consider the case $r_1 = r_2 > r_3$. This implies that t_0 is nonobtuse, $\gamma_1 = \delta > \pi/2$ and $\alpha_1 = \varepsilon$, and the result holds. For the case $r_1 > r_2 > r_3$, consider triangle t_1 equal to triangle ABC in Figure 4. In this figure, the angle ε is opposite side AB which is equal to $r_3/2$, and angle σ is opposite side BC which is equal to $r_2/2$. Since $r_3/2 < r_2/2$, it follows that $\varepsilon < \sigma$, which in turn implies the result. $\square$

Remark. An ambiguity arises in the case $r_1 \geq r_2 = r_3$, which implies that $\varepsilon = \sigma$ (t_1 is isosceles), and consequently, either $\gamma_1 = \varepsilon = \sigma$ or $\alpha_1 = \varepsilon = \sigma$. In order to be consistent with Lemma 1, in the remainder of this paper we shall assume that $\gamma_1 = \sigma$ in the first case and $\alpha_1 = \varepsilon$ in the second case. With this convention, the results of Lemma 1 are valid for any triangle t_0 of sides $r_1 \geq r_2 \geq r_3$ and angles $\gamma_0 \geq \beta_0 \geq \alpha_0$.
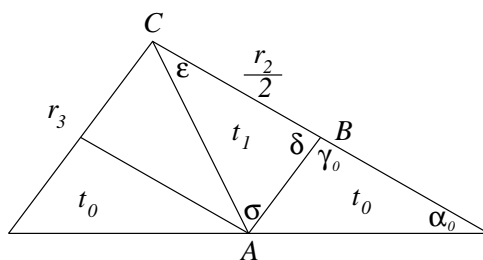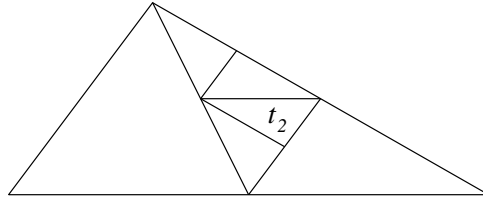

FIGURE 4

FIGURE 5

The following theorem characterizes the first partitions of nonobtuse triangles:

Theorem 1. *If t_0 is a nonobtuse triangle of largest angle $\gamma_0 \leq \pi/2$, then t_1 is an obtuse triangle such that*

(a) $\gamma_1 = \delta$,
(b) $\gamma_0 + \gamma_1 = \pi$ *and either t_1 is an obtuse triangle ($\gamma_0 < \pi/2$), or t_1 is a right triangle (if $\gamma_0 = \pi/2$),*
(c) *the 4-triangles longest-side partition of t_1 produces t_0.*

Proof. According to Figure 4, $\gamma_0 + \delta = \pi$, and since $\gamma_0 \leq \pi/2$, clearly $\delta \geq \pi/2$, which implies $\gamma_1 = \delta$ and $\gamma_0 + \gamma_1 = \pi$. Furthermore, because of $\gamma_1 = \delta$ in Figure 4, the partition of t_1 must be necessarily performed as illustrated in Figure 5, which produces a triangle t_2 of sides parallel to the sides of t_0, so that t_2 is similar to t_1. □

Corollary 1. *If t_0 is a right triangle, then $t_1 = t_0$ and the iterative partition of t_0 only produces t_0 itself.* □

Corollary 2. *The iterative 4-triangles longest-side partition of any nonobtuse triangle with $\gamma_0 < \pi/2$ produces two distinct (up to similarity) triangles t_0 and t_1 such that triangle t_1 satisfies the properties of Theorem 1. Furthermore, t_1 only produces the triangles t_0 and t_1.* □

4. PROPERTIES OF THE 4-TRIANGLES LONGEST-SIDE PARTITION OF OBTUSE TRIANGLES

The following theorem characterizes the triangles obtained by means of the 4-triangles longest-side partition of obtuse triangles.

Theorem 2. *The 4-triangles longest-side partition of any obtuse triangle t_0 ($\gamma_0 > \pi/2$) produces a new triangle t_1 of smallest angle $\alpha_1 > \alpha_0$ and largest angle $\gamma_1 \leq \gamma_0 - \alpha_1$. Furthermore, either $\gamma_1 = \sigma$ while t_1 remains obtuse; or $\gamma_1 = \delta$ and $\gamma_0 + \gamma_1 = \pi$, when t_1 becomes nonobtuse.*

Proof. Notice that the hypothesis $\gamma_0 > \pi/2$ implies $\gamma_0 > \beta_0 \geq \alpha_0$ and $r_1 > r_2 \geq r_3$. According to Lemma 1, after one 4-triangles longest-side partition of t_0, the largest angle γ_1 of t_1 can be either σ or δ in Figure 3.

In the case $\gamma_1 = \delta$, clearly $\gamma_1 = \pi - \gamma_0 < \pi/2$ and t_1 is a nonobtuse triangle, $\alpha_1 = \varepsilon < \alpha_0$ and $\gamma_1 \leq \gamma_0 - \alpha_1$. In the case $\gamma_1 = \sigma$, the angles of t_1 are distributed as shown in Figure 6(b) (according to Lemma 1), where α_1 can be either ε or δ. We shall show that in both cases the hypothesis holds. Consider $\alpha_1 = \varepsilon$ and triangle ABC. In this triangle, the angles ε and α_0 are respectively opposite the
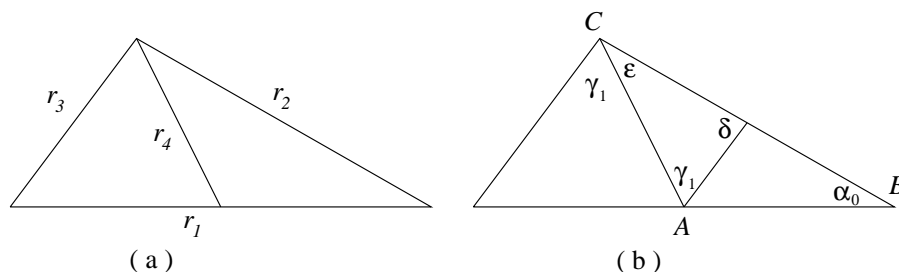
FIGURE 6

sides $r_1/2$ and r_4. Because of the position of γ_1 it follows that $r_2/2 \geq r_4$, and since $r_1 > r_2$, necessarily $r_1/2 > r_2/2 \geq r_4$, which implies $\varepsilon > \alpha_0$. Furthermore, the angle of vertex C is $\gamma_0 = \gamma_1 + \varepsilon$, which implies $\gamma_1 = \gamma_0 - \varepsilon$, and the hypothesis holds. Consider now the angles $\alpha_1 = \delta$ and α_0. In this case, $\delta = \pi - \gamma_0$, and since $\gamma_0 = \pi - \alpha_0 - \beta_0$, it follows that $\delta > \alpha_0$ and $\alpha_1 > \alpha_0$. Furthermore, the angle of vertex C is $\gamma_0 = \gamma_1 + \epsilon = \gamma_1 + \beta_1$, which implies $\gamma_1 = \gamma_0 - \beta_1 \leq \gamma_0 - \alpha_1$. □

The following theorem characterizes the iterative partition of obtuse triangles.

Theorem 3. *The iterative 4-triangles longest-side partition of any obtuse triangle t_0 produces a finite sequence of "better" distinct (up to similarity) triangles $\{t_i\}_1^N$ such that*

(a) *t_i is obtuse for $i = 1, 2, \ldots, N - 1$, and t_N is nonobtuse,*
(b) *$\alpha_j > \alpha_{j-1}$ for $j = 1, \ldots, N$,*
(c) *$\gamma_j \leq \gamma_{j-1} - \alpha_j$ for $j = 1, 2, \ldots, N$,*
(d) *the next partition of t_N only (at most) produces a new obtuse triangle t_{N+1}, and at this point the generation of new triangles stops.*

Proof. Parts (a), (b) and (c) follow from a direct iterative application of Theorem 2. Part (d) is a consequence of Theorem 1 and Corollaries 1 and 2. □

Corollary 3. *The iterative partition of any obtuse triangle t_0 produces a sequence of distinct (up to similarity) triangles $\{t_i\}_i^N$, t_i obtuse for $i = 1, \ldots, N - 1$ and t_N nonobtuse, such that $\alpha_j > \alpha_0$ and $\gamma_j < \gamma_0 - j\alpha_0$ for $j = 1, 2, \ldots, N$. In addition, only (at most) a new obtuse triangle can be obtained by partition of t_N.* □

Finally, Corollary 3 allows us to establish an a priori bound on the number of distinct triangles generated as a function of the geometry of the initial triangle.

Theorem 4. *For any obtuse triangle t, the number K of distinct (up to similarity) triangles generated throughout the process is bounded as follows:*

$$K \leq \lceil \tfrac{\gamma_0 - \pi/3}{\alpha_0} \rceil + 1.$$

Proof. From Corollary 3, the angle γ_N satisfies

$$\pi/3 \leq \gamma_N \leq \gamma_0 - N\alpha_0,$$

which implies $N \leq \lceil \tfrac{\gamma_0 - \pi/3}{\alpha_0} \rceil$. Since at most a new distinct (obtuse) triangle is generated in iteration $N + 1$, then $K \leq N + 1$ and the result holds. □

5. A 3-TRIANGLES PARTITION

As a by-product of the study performed in the preceding sections, we have implicitly introduced a 3-triangles longest-side partition that also improves obtuse triangles and that can be used to advantage (together with the pure longest-side bisection) to perform refinement propagation in local refinement algorithms for triangulations [12, 15].

Definition 2. The 3-triangles longest-side partition of any triangle t_0 (of sides $r_1 \geq r_2 \geq r_3$ and angles $\gamma_0 \geq \beta_0 \geq \alpha_0$) is obtained by joining the midpoint of the longest side r_1 with the opposite vertex and the midpoint of side r_2. (For an illustration see Figure 7).
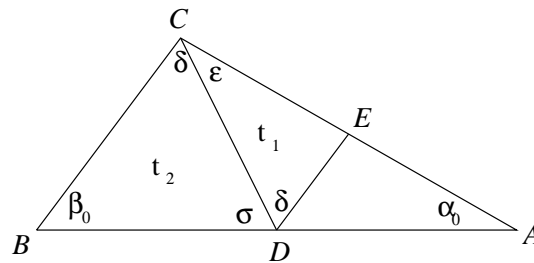

FIGURE 7

This partition also improves obtuse triangles, as stated in Theorem 5.

Theorem 5. *The 3-triangles longest-side partition of any obtuse triangle t_0 (of angles $\gamma_0 > \beta_0 > \alpha_0$) produces two new distinct (up to similarity) triangles $t_i = t_i(\gamma_i, \beta_i, \alpha_i)$ which are better than t_0 in the following sense:*

$$\gamma_i < \gamma_0 \ and \ \alpha_i > \alpha_0 \ \ for \ \ i = 1, 2.$$

Proof. Clearly, two new (distinct) triangles are generated: triangles t_1 and t_2 of Figure 7. For the triangle t_1, the result follows directly from Theorem 2. Let us consider triangle t_2. First we shall study γ_2, the largest angle of triangle t_2. Clearly, $\delta < \gamma_0$ and $\beta_0 < \gamma_0$. Moreover, t_0 is a nonisosceles, obtuse triangle, which implies $\sigma < \pi/2 < \gamma_0$. Thus, $\gamma_2 < \gamma_0$. Now we shall consider α_2, the smallest angle of triangle t_2. Clearly, $\beta_0 > \alpha_0$ and $\delta > \pi/4 > \alpha_0$ while from Theorem 2 it follows that $\sigma = \alpha_0 + \varepsilon \leq \alpha_0 + \alpha_1$, which together implies $\alpha_2 > \alpha_0$. $\square$

6. NUMERICAL EXPERIMENTS

In order to illustrate the practical behavior of the 4-triangles longest-side partition of triangles, we have included in Table 1 the sequence of distinct (up to similarity) triangles generated for six different input triangles. In this table, each triangle has been identified by its interior angles (in decreasing order), the input triangle being the first one of each column (iteration 0). For each sequence we have also included the exact number of distinct (up to similarity) triangles generated, as well as the a priori computed bound of Theorem 4.

Notice that the numerical results obtained are in complete agreement with the theory: the iterative 4-triangles longest-side partition of obtuse triangles systematically improves the triangles (both the sequence of largest angles monotonically

TABLE 1. Sequences of distinct triangles obtained by 4-triangles longest-side partition of triangles

It.	Triangle 1			Triangle 2		
	number of distinct triangles 15			number of distinct triangles 11		
	computed bound 45			computed bound 190		
	γ	β	α	γ	β	α
0	145.455	32.595	1.950	173.972	5.423	0.605
1	143.292	34.545	2.164	173.216	6.028	0.756
2	140.885	36.708	2.407	172.245	6.784	0.971
3	138.200	39.115	2.684	170.952	7.755	1.293
4	135.202	41.800	2.998	169.148	9.048	1.804
5	131.850	44.798	3.351	166.462	10.852	2.686
6	128.107	48.150	3.743	162.066	13.538	4.396
7	123.937	51.893	4.170	153.735	17.934	8.331
8	119.316	56.063	4.621	133.923	26.265	19.812
9	114.235	60.684	5.081	84.274	49.648	46.077
10	108.715	65.765	5.520	95.726	43.599	40.676
11	102.811	71.285	5.904	84.274	49.648	46.077
12	96.618	77.189	6.193			
13	90.266	83.382	6.352			
14	89.734	83.907	6.359			
15	90.266	83.382	6.352			

It.	Triangle 3			Triangle 4		
	number of distinct triangles 8			number of distinct triangles 4		
	computed bound 73			computed bound 7		
	γ	β	α	γ	β	α
0	169.900	8.572	1.527	114.624	54.900	10.475
1	167.721	10.100	2.180	102.073	65.376	12.551
2	164.371	12.279	3.349	88.250	77.927	13.824
3	158.625	15.629	5.747	91.750	74.623	13.627
4	146.921	21.375	11.704	88.250	77.927	13.824
5	117.268	33.079	29.652			
6	63.237	62.732	54.031			
7	116.763	33.270	29.967			
8	63.237	62.732	54.031			

It.	Triangle 5			Triangle 6		
	number of distinct triangles 3			number of distinct triangles 2		
	computed bound 5			computed bound 5		
	γ	β	α	γ	β	α
0	130.542	27.127	22.332	116.565	45.000	18.435
1	76.437	54.105	49.458	90.000	63.435	26.565
2	103.563	39.659	36.777	90.000	63.435	26.565
3	76.437	54.105	49.458			

decreases and the sequence of smallest angles monotonically increases). Furthermore, the partition of nonobtuse triangles produces reasonably good obtuse triangles, which in turn reproduce the same previous nonobtuse triangle, and so on, alternately, as explained in Diagram 2 of §2 (the generation of new triangles stops when a previously generated triangle is reobtained). More specifically, triangles 1 and 5 of Table 1 illustrate case (1) of Diagram 2, while triangle 6 illustrates case (2), and triangles 2, 3 and 4 illustrate case (3) of this diagram. Notice that in practice the input triangles having the smallest β_0 produce better final triangles than those having the biggest β_0. It seems this is due to the fact that the sequence of angles β is also a monotonically increasing sequence.

7. A 4-TRIANGLES REFINEMENT ALGORITHM

A 4-triangles refinement algorithm that uses 4-triangles partitions over the refinement region R (variable in the adaptive context) and both 3-triangles partitions and longest-side bisections to assure conformity of the mesh (refinement propagation), has been studied, discussed and illustrated in [15]. Software that combines this algorithm with an adaptive and multigrid (linear) finite element method has been extensively discussed in [13, 14]. Numerical experiments performed with this software to solve singular problems have shown that, in adaptive mode, the numerical rate of convergence of the finite element method approaches the maximum theoretical rate of convergence predicted for linear finite elements [13, 14].

A simpler 4-triangles refinement algorithm (whose time cost is easier to analyze) that only uses longest-side bisection of triangles in the refinement propagation step, can be schematically described as follows:

> Triangulation-refinement (τ, R, ε)
> While there exist triangles $t \in \tau$, $t \cap R \neq \varnothing$ and diam$(t) > \varepsilon$ do
>> Perform 4-triangles bisection of each t with $t \cap R \neq \varnothing$ and diam$(t) > \varepsilon$
>> For each $t \in \tau$ having 1, 2 or 3 nonconforming side midpoints P do
>> While P is nonconforming do
>>> Find the neighbor t^* of t (by the side containing P)
>>> Longest-side bisection (τ, t^*)

This algorithm makes use of the following recursive basic longest-side refinement algorithm (to refine a unique triangle t) [19]:

> Longest-side bisection (τ, t)
> Perform a longest-side bisection of t
>> (Let P be the point generated)
> While P is nonconforming do
>> Find the neighbor t^* of t (by the side containing P)
>> Longest-side bisection (τ, t^*)

Figure 8 illustrates the use of the 4-triangles refinement algorithm to perform one refinement step over the simple refinement region $R = \{t\}$ in the triangulation of Figure 8(a). Here, first a 4-triangles partition of triangle t was performed, followed by the recursive longest-side bisection of the neighboring nonconforming triangles in order to make the mesh conforming (i.e., to eliminate the nonconforming midpoints 1 and 2 in Figure 8(b)).

To conclude this section, Figures 9 and 10 illustrate the practical use of the 4-triangles refinement algorithm to solve (adaptively) the Laplace problem with Dirichlet boundary conditions over the polygonal re-entrant region of Figure 9.
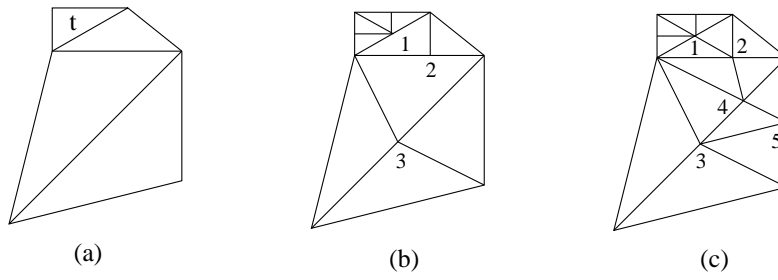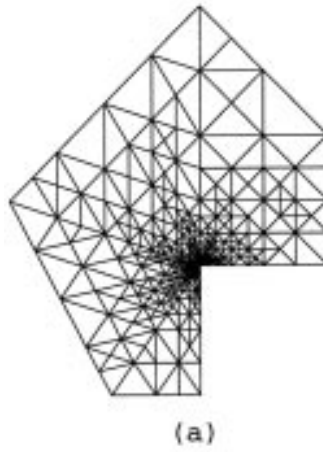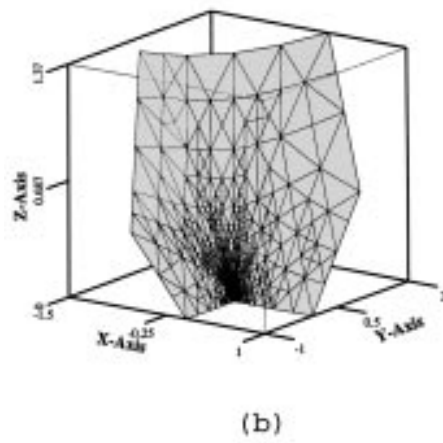
FIGURE 8

(a)

FIGURE 9

(b)

FIGURE 10

The solution of this problem is the function $u = r^{2/3} \sin(2\theta/3)$, which is singular at the origin, owing to the re-entrant corner. The triangulation of Figure 9 was adaptively constructed starting with a 5-triangles initial triangulation, while that Figure 10 corresponds to the numerical finite element solution obtained for this mesh.

8. Cost analysis: The algorithm is linear

The time cost analysis of the algorithm should study its asymptotic behavior to obtain triangles of size ε over the refinement region R. From this point of view the refinement algorithm performs two essential tasks:

(i) *Task* 1. Insertion of N vertices inside the region R (refinement of the target triangles) which ensures that the tolerance condition on the diameters of the triangles is satisfied.

(ii) *Task* 2. Insertion of K vertices outside the region R (by refinement of some nontarget triangles), which ensures the conformity of the refined mesh.

Both issues should be addressed in the time-cost analysis of the algorithm. In [19], the cost analysis of the pure longest-side bisection refinement algorithm has been performed. We shall follow here an analogous reasoning to that of this paper.

Since the time cost of each triangle partition (4-triangles partition and pure longest-side bisection) is constant, the time cost of inserting an arbitrary number of points in the input triangulation (a consequence of the partition of a set of triangles) is linear, independently of the size (number of vertices) of the input triangulation τ. This allows us to state the following results:

Theorem 6. *The time cost of performing Task 1 and Task 2 is respectively linear in N (it the number of vertices inserted inside of R) and K (the number of vertices inserted in $\Omega \backslash R$).*

Theorem 7. *The number of points inserted in R (Task 1) is optimal.*

Proof. Since a finite number of distinct (up to similarity) triangles is used throughout the entire process (which only depends on the initial triangulation) the number of points inserted inside R (to obtain triangles of size ε) is $N = kN_e$, where k is a constant and N_e is the number of vertices used in an equilateral triangulation. □

Thus, the key issues of the analysis have been reduced to answer the following question: how is the size of K compared with N? The answer is certainly related to the properties of the distribution of points introduced in the propagation step of the pure longest-side refinement algorithm, which is used to conform the mesh in the algorithm of the previous section. To this end, we need to study the behavior of the (longest-side) algorithm to refine triangulations around one vertex. In this particular case, the refinement region is $R = \{P\}$, where P is a vertex. The one-triangle longest-side refinement algorithm of the previous section is iteratively used until the diameter of the triangles that share the vertex P is less than or equal to ε. For this particular problem the following result holds [19]:

Theorem 8. *Let τ be any conforming triangulation. For any vertex P of τ, the arbitrary iterative use of the longest-side refinement algorithm to refine the trian-*
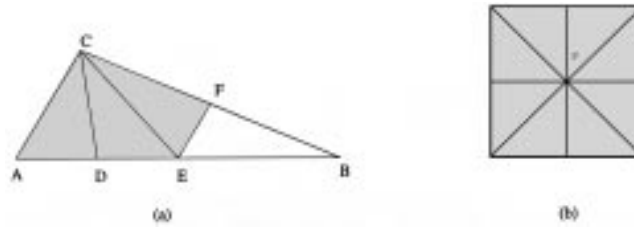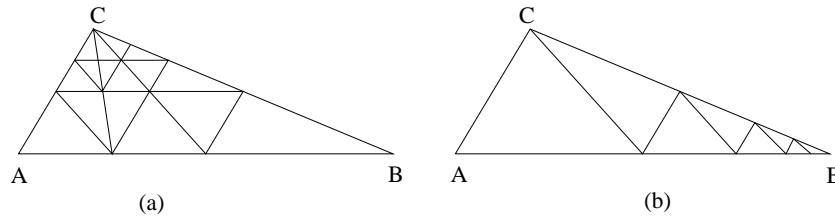
FIGURE 11

FIGURE 12

gulation around vertex P divides the angles converging in P in a finite number of k parts.

Proof. This follows directly from the fact that the angles obtained by longest-side bisection of any triangle are bounded [20]. □

Definition 3. For any conforming triangulation τ and any vertex P of τ, we shall call the *stable molecule associated* with vertex P to be the fixed partition of the plane around vertex P, induced by the iterative use of the longest-side refinement algorithm, for refining the triangulation around the vertex P. The size of the stable molecule will be the number of angles converging in P.

The examples of Figures 11 and 12 illustrate these ideas. More specifically, the shadowed polygons of Figures 11(a) and 11(b), respectively correspond to the stable molecule associated with the vertices C and P (the angles converging in C and P are not refined anymore). In Figure 12, the longest-side refinement algorithm was used over a one-triangle initial triangulation (triangle ABC) in order to respectively refine the triangulation around the vertices C and B. Notice that the first iterations of the algorithm around vertex C have constructed the stable molecule, while the next iterations of the algorithm only produce a fractal geometry around this vertex. In the case of vertex B, in contrast, the algorithm immediately "finds" the fractal geometry, since the stable molecule of vertex B is exactly defined by the sides AB and BC.

All these ideas can be summarized in Theorem 9 [19] and used to prove Theorem 10.

Theorem 9. *Let τ be any conforming triangulation and consider any vertex P of τ. The use of the longest-side refinement algorithm to refine the triangulation around the vertex P produces triangulations having 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 next 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. Part (b) follows from the fact that the triangles that share vertex P in each intermediate triangulation have longest side coincident with one of the straight line segments that defines the stable molecule, where P is one of its end points. Thus, the new points inserted are always the midpoints of these sides. $\square$

Corollary 4. *The arbitrary iterative refinement of any triangulation τ_0 produces triangulations of maximal stable molecule of size $k \leq M$, where M only depends on τ_0.*

Definition 4. The constant M of Corollary 4 will be called the stable molecular size of τ_0.

Theorem 10. *Suppose that the longest-side refinement algorithm is used to refine any triangulation τ around any vertex (of stable molecule of size k). Then, once the stable molecule is obtained, the iterative refinement around vertex P to obtain triangles of diameter (longest side) h introduces K vertices, where*

$$K \leq k[1 + log_2(1 + 2L/h)]$$

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

Proof. Along each side $P\overline{P}$ of the stable molecule of P, the algorithm introduces a sequence of points $P_1, P_2,..., P_j$ such that P_1 is the midpoint of the side $P\overline{P}$ and P_j the midpoint of the side PP_{j-1}. In order to ensure the condition on the diameters of the triangles to be obtained around the vertex P, in particular, for the longest side of the stable molecule, the algorithm should stop when $\widetilde{h} \leq \frac{h}{2}$, where $\widetilde{h}$ is the length of the side PP_j. But $\widetilde{h} \sum_{1 \leq i \leq J} 2^i = L$, which implies that $\widetilde{h}(2^J - 1) = L$ and $J = 1 + log_2(1 + L/\widetilde{h})$. Thus, along each side of the stable molecule of P (of the triangulation τ), the algorithm introduces at most J vertices, and since the stable molecule has k arms, the result follows. $\square$

Theorem 11. *The time cost of solving the Point Triangulations Refinement Problem to obtain triangles of given size is $O(K)$, where K is bounded as in Theorem 10.*

Theorem 12. *Let τ_0 be any conforming coarse triangulation (of stable molecular size M) of any bounded polygonal region Ω. Then for any circular refinement subregion C, of radius r, with $C \subset \Omega$, the use of the 4-triangles refinement algorithm to solve the Triangulation Refinement Problem over C to obtain triangles of diameter $\widetilde{h}$ 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:*

$$(a) \qquad N_i \leq k_1[1 + \pi(n^2 + n)],$$

$$(b) \qquad N_o < \pi k_2 n[1 + \log_2(1 + L/\widetilde{h})],$$

where k_1 and k_2 are constants, $n = 2r/\widetilde{h}$ and L is the maximum distance from the boundary of R to the boundary of Ω.

Proof. (a) According to Theorem 7, $N_i \leq kN_e$. In order to bound N_e, cover C with concentric circles of radius $h, 2h, 3h, ..., r$, where $h = \widetilde{h}/2$, and over each circle, consider a set of points uniformly distributed at distance h. Then

$$N_e < 1 + \frac{2\pi h}{h} + \frac{2\pi 2h}{h} + \frac{2\pi 3h}{h} + \cdots + \frac{2\pi r}{h}$$
$$= 1 + 2\pi(1 + 2 + \cdots + n) = 1 + \pi(n^2 + n),$$

where $n = r/h = 2r/\widetilde{h}$. Then $N_i \leq k_1[1+\pi(n^2 + n)]$, and the first part of the result follows.

(b) Notice that the refinement inside C has introduced at most $4\pi r/\widetilde{h}$ points on the boundary of C. The number of points introduced in the exterior of C can be bounded by the product of the stable molecular size M of τ_0 and the number of vertices introduced by local refinement along each vertex, assuming a stable molecule of maximal side L. Thus,

$$N_o < M2\pi n[1 + log_2(1 + L\widetilde{h})]. \quad \square$$

Corollary 5. *Under the conditions of Theorem 12, the longest-side refinement algorithm introduces an optimal number of points $N = O(n^2)$ inside the refinement region. Furthermore, since Ω is a bounded region, the time cost of the algorithm is linear in N.*

A stronger result was introduced in [19]. Specifically, it was proved that the number of points inserted outside R is of order $O(n \, log_2 \, n)$, both for circular and rectangular refinement regions, which allows us to conclude that this is true for general convex refinement regions. Empirical evidence, which is in complete agreement with the theory, even for small values of N, was also included in that paper.

9. Concluding remarks

In this paper we have discussed results which characterize, in a geometrical sense, the global point distribution introduced by the iterative 4-triangles longest-side partition of triangles. We have also discussed a practical 4-triangles refinement algorithm, proving that it has linear complexity, independently of the size of the triangulation. To this end, the "fractal" properties of the longest-side bisection algorithm (which characterize the local point distribution introduced by the algorithm) were used. Numerical experiments illustrating the practical behavior of the linear algorithm can be found in [19].

A discussion on the triangulation refinement problem and on the difficulties arising with the pure Delaunay algorithm to deal with this problem have also been included in this paper. In essence, this algorithm does not contain a natural point insertion strategy that guarantees the construction of good-quality irregular triangulations when iterative local refinement is performed. However, both the longest-side bisection based algorithms, and mixed (longest-side midpoint insertion) Delaunay algorithms are competitive linear algorithms for the triangulation refinement problem. In a forthcoming paper we shall discuss mixed Delaunay algorithms in more detail.

References

1. I. Babuška, O.C. Zienkiewcz, J. Gago and E. R. de A. Oliveira (eds.), *Accuracy estimates and adaptive refinements in finite element computations*, John Wiley & Sons, Chichester (1986). MR **87j:**65004

2. S.W. Bova and G. F. Carey, *Mesh generation/refinement using fractal concepts and iterated function systems*, Internat. J. Numer. Meth. in Engrg., 33 (1992), 287-305. MR **92j:**65137

3. P. Dierckx, S. Van Leemput and T. Vermeire, *Algorithms for surface fitting using Powell-Sabin splines*, IMA J. Numer. Anal., 12(1992), 271-299. MR **93b:**65026

4. W. Hackbush, *Multi-grid methods and applications*, Springer-Verlag, Berlin (1985). MR **87e:**65082

5. H. Kim, S. Hong, K. Choi, H. Jung and S. Hahn, *A three dimensional adaptive finite element method for magnetostatic problems*, IEEE Trans. Magnetics, 27(1991), 4081-4084.

6. L. Freitag, M. Jones and P. Plassmann, *New techniques for parallel simulation of high-temperature superconductors*, Argonne National Laboratory 1994 Preprint MCS-P44-0594.

7. S.N. Muthukrishnan, P.S. Shiakolas, R. V. Nambiar and K.L. Lawrence, *A simple algorithm for the adaptive refinement of three dimensional finite element tetrahedral meshes*, to appear AIAA Journal.

8. R.V. Nambiar, R.S. Valera, K.L. Lawrence, R. B. Morgan and D. Amil, *An algorithm for adaptive refinement of triangular meshes*, Internat. J. Numer. Meth. Engrg. 36 (1993), 499-509.

9. J. O'Rourke, *Computational Geometry Column 23*, Internat. J. Comput. Geom. Appl. 4(1994), 239-242. MR **95j:**52030

10. J. Penman and M.D. Grieve, *Self-adaptive mesh generation technique for the finite element method*, IEE Proceedings, 134(1987), 634-650.

11. M.-C. Rivara, *Mesh refinement processes based on the generalized bisection of simplices*, SIAM J. Numer. Anal. 21(1984), 604-613. MR **85i:**65159

12. _____, *Algorithms for refining triangular grids suitable for adaptive and multigrid techniques*, Internat. J. Numer. Methods Engrg., 20(1984), 745-756. MR **85h:**65258

13. _____, *Design and data structure for fully adaptive, multigrid finite-element software*, ACM Trans. Math. Software, 10(1984), 242-264. MR **86f:**65207

14. _____, *Adaptive finite element refinement and fully irregular and conforming triangulations*, In Accuracy Estimates and Adaptive Refinements in Finite Element Computations, I. Babuška, O.C. Zienkiewicz, J. Gago and E.R. de A. Oliveira (eds.), John Wiley & Sons, Chichester (1986), pp. 359-370.

15. _____, *A grid generator based on 4-triangles conforming mesh refinement algorithms*, Internat. J. Numer. Meth. Engrg., 24 (1987), 1343-1354.

16. _____, *Selective refinement/derefinement algorithms for sequences of nested triangulations*, Internat. J. Numer. Meth. Engrg., 28(1989), 2889-2906. MR **90j:**57018

17. _____, *A discussion on the triangulation refinement problem*, Proceedings of the Fifth Canadian Conference on Computational Geometry, University of Waterloo (1993), pp. 42-47.

18. M.-C. Rivara and C. Levin, *A 3D refinement algorithm for adaptive and multigrid techniques*, Comm. Appl. Numer. Methods, 8 (1992), 281-290.

19. M.-C. Rivara and M. Venere, *Cost analysis of the longest-side (triangle bisection) refinement algorithms for triangulations (1994)*, to appear Engineering with Computers.

20. I. Rosenberg and F. Stenger, *A lower bound on the angles of triangles constructed by bisecting the longest side*, Math. Comp. 29(1975), 390-395. MR **51:**11264

21. M. Stynes, *On faster convergence of the bisection method for all triangles*, Math. of Comp. 36 (1980), 1195-1201. MR **81j:**51023

22. R. Williams, *Adaptive parallel meshes with complex geometry*, In Numerical Grid Generation in Computational Fluid Dynamics and Related Fields (A.S.Arcilla, J.Hauser, P.R.Eiseman, J.F.Thompson, eds.), Elsevier Science Publishers B.V. (North-Holland) (1991). CMP 91:16

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF CHILE, CASILLA 2777, SANTIAGO, CHILE
E-mail address: `mcrivara@dcc.uchile.cl`

E-mail address: `giribar@dcc.uchile.cl`