

Automatic construction of quality nonobtuse boundary and/or interface Delaunay triangulations the control volume methods

Nancy Hitschfeld and María-Cecilia Rivara
Department of Computer Science, University of Chile,
Casilla 2777, Santiago, CHILE
e-mail: nancy@dcc.uchile.cl, mcrivara@dcc.uchile.cl
TR/DCC-99-10 (submitted to a journal)

Abstract.

In this paper we discuss a new algorithm that for any input quality constrained Delaunay triangulation with minimum interior angle greater than or equal to 30° , produces a quality nonobtuse boundary and/or interface Delaunay triangulation by the Delaunay insertion of a finite number of boundary and/or interface points. A boundary (interface) obtuse triangle is a triangle that has an obtuse angle opposite to a boundary (interface) edge. The output mesh might have a small number of triangles with interior angles less than 30° in the neighborhood of the triangles with boundary constrained angles.

The analysis of the algorithm considers two cases depending on the geometric complexity of the domain: (a) simple polygonal domains which may include holes and (b) polygonal domains with interfaces. In case (a) every obtuse triangle with one boundary edge is eliminated by the Delaunay insertion of one point, and every obtuse triangle having both medium size edge and longest edge (of respective lengths l and L) over the boundary and boundary constrained angle β is eliminated by building an isosceles triangle of boundary edges of lengths $l/2$ (which maintains the Delaunay triangulation) followed by the Delaunay insertion of a finite number of points N , where $N < K$, and $K = \lceil 2 + \frac{2L-l}{l\sqrt{1-\cos(\beta)}} \rceil$. In case (b), there are interface obtuse triangles either isolated or arranged into a group of adjacent interface triangles. The isolated interface obtuse triangles are destroyed in a similar way to boundary obtuse triangles of case (a), and on the contrary, the grouped interface triangles are destroyed together by Delaunay insertion of a finite number of points.

It is proved that the algorithm produces an almost (non-constrained) Delaunay triangulation in the sense that all pair of triangles sharing an interface edge satisfy the Delaunay condition. Examples of the practical behavior of the algorithm combined with a Lepp-Delaunay algorithm to produce the initial triangulation are also included.

keywords. Nonobtuse triangulation, Delaunay meshes, control volume method.

1 Introduction

The numerical solution of partial differential equations (PDEs) is invaluable in design and optimization in many fields of engineering. The spatial discretization (mesh) of the structure to be simulated, is key to the accuracy of the computed solution. An appropriate mesh should fulfill several requirements. First, it must provide a reasonable approximation of the geometry to be modeled, in particular of its boundary and internal material interfaces. Second, it is extremely important to accurately approximate all internal quantities relevant to the solution of the PDEs. Third, each cell must fulfill certain geometric constraints imposed by the numerical method: if the PDEs are solved with the finite element method, no angle must be smaller than some bound supplied *a priori*. If the equations are solved using a control volume discretization method (CVM)[1], the center of the circumcircle that surrounds each boundary (interface) element must be inside the element or inside the polygon that contains the element [2]. In case of a triangulation, this means that the angle opposite to a boundary (interface) edge must be an acute angle.

The CVM is very popular in the numerical simulation of semiconductor devices [1, 3, 4]. The meshes for the CVM can be classified into two groups: (1) nonobtuse meshes, i.e, meshes without any obtuse angle, and (2) nonobtuse boundary (interface) meshes. In the first group, we can find approaches such that nonobtuse triangulations [1, 3, 5] and rectangle based meshes [6, 7]. The second group is still less developed than the first group. One of the first approaches for simple polygonal domains is based on the sphere packing technique [5] and was presented in [8].

This paper presents a new algorithm to generate 2-D nonobtuse boundary (interface) meshes for the CVM for polygonal domains with interfaces. The algorithm receives as input any quality constrained Delaunay triangulation (CDT), whose angles are bounded by 30° and 120° and eliminates all the boundary (interface) obtuse triangles. The domain is specified by a planar straight line graph (PSLG), which can include polygons, polygons with holes, and complexes (objects made of multiple polygons); dangling edges and isolated vertices are also allowed. Even when every algorithm able to produce an initial good quality CDT could be used, we generate the CDT with the Lepp Delaunay algorithm introduced by Rivara in [9, 10], which consists of: (a) The generation of an initial CDT (which essentially uses the PSLG vertices), and (b) the use of an Lepp-Delaunay algorithm which improves the quality of the mesh so that the minimum angle is greater than or equal to 30° . The basic Lepp-Delaunay improvement strategy uses the Longest-Edge Propagation Path (Lepp) of the target triangles in order to decide which is the best point to be inserted, to produce a good-quality point distribution. This strategy is repeatedly used until the target triangle is destroyed.

The analysis of the proposed algorithm is divided into two cases depending on the domain complexity: (a) simple polygonal domains which may include holes and (b) polygonal domains with interfaces. In case (a) an obtuse triangle with one boundary edge is eliminated by the Delaunay insertion of the midpoint of the boundary edge, and an obtuse triangle with two boundary edges is eliminated by building an isosceles triangle and inserting Delaunay a finite number of points if they are required to eliminate

new boundary obtuse triangles with one boundary edge. The isosceles triangle has two boundary edges of length equal to half of the length of the smallest boundary edge of the target triangle. In case (b), in addition to the boundary obtuse triangles of case (a), there exist interface obtuse triangles both isolated or arranged into a group of adjacent interface triangles. The isolated interface obtuse triangles are destroyed in a similar way to boundary obtuse triangles of case (a), and the interface triangles arranged into a group are destroyed together by inserting Delaunay a finite number of points.

In addition, it is proved that the algorithm produces an almost (non-constrained) Delaunay triangulation in the sense that triangles lying at the interfaces satisfy the Delaunay condition.

Finally note that this kind of meshes can be very useful in semiconductor simulations when the device simulation is solved combining finite element and control volume methods [3]. This requires the combination of good quality meshes and well shaped Voronoi boxes. In particular, the minimum angle should be bounded and boundary triangles should not have obtuse angles opposite to any boundary edge or interface edge.

2 Basic concepts and definitions

This section introduces name conventions for boundary and interface obtuse triangles, the geometrical restrictions of the numerical method known as control volume, the Lepp concept and some geometrical properties.

2.1 Boundary and interface triangles

In general, we shall call a boundary triangle to any triangle that has either one, two or three boundary edges and none interface edge, and an interface triangle to any triangle that has at least one interface edge (note that an interface triangle can have a boundary edge).

In order to distinguish the different cases to be considered which depends on the number of edges that a triangle has along a boundary or an interface, the following definitions will be considered:

Definition 1 *A 1-edge boundary (interface) triangle is any triangle that has exactly one boundary (interface) edge. A 2-edge boundary triangle is any triangle that has exactly two boundary edges; and a 2-edge interface triangle is any triangle that has either 2 interface edges or one interface and one boundary edge.*

Other relevant definitions are:

Definition 2 *A boundary (interface) obtuse triangle is any triangle that has a boundary and/or interface edge opposite to its obtuse angle.*

Definition 3 *A boundary (interface) constrained angle is an angle that is defined by two boundary (interface) edges. This angle can not be modified.*

2.2 Triangulation restrictions for the control volume discretization method

The following definition describes the main restriction imposed over triangulations by the control volume discretization method (CVM).

Definition 4 Let P be any input PSLG. A triangulation τ of P is appropriate for the CVM (well-shaped) if

- (i) T is a Delaunay triangulation,
- (ii) The center of the circumcircle (Voronoi point) of each boundary triangle lies inside the boundary triangle or inside a neighboring triangle through interior edges.

The Delaunay triangulation and its dual, the Voronoi diagram fit very well with the CVM, because the Voronoi cells act as the control volumes, which are in turn used to compute the numerical integration around each mesh point. Figure 1(a) shows a well-shaped triangulation and its corresponding Voronoi diagram. The Voronoi diagram is shown with thick lines. Figure 1(b) shows a non-acceptable triangulation because it has a boundary triangle (the triangle defined by the vertices p_j, p_k, p_i) whose circumcircle center (Voronoi point v) lies outside the mesh. This occurs when the angle opposite to a boundary edge is an obtuse angle. For more information about the CVM and the restrictions on the mesh see [1, 8].

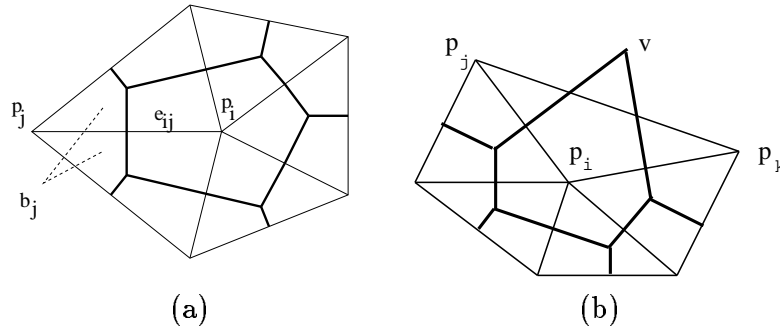


Figure 1: 2-D Delaunay triangulations and their Voronoi diagrams: (a) acceptable triangulation for the CVM, and (b) unacceptable triangulation

2.3 Basic definitions and geometrical properties

In this section, we introduce geometrical properties that we will use later to prove the correctness of the proposed algorithm to eliminate boundary (interface) obtuse triangles.

Definition 5 The diameter circle of any edge of vertices A and B (C_{AB}) is the circle with center equal to the midpoint of edge AB and diameter AB .

The following property is a particular case of a theorem presented in [11].

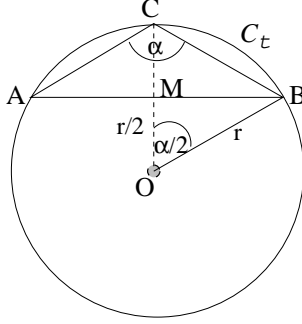


Figure 2: α is equal to 120° if and only if the distance between O and M (the midpoint of AB) is equal to $r/2$

Proposition 1 *Let $t(A,B,C)$ be any triangle of longest edge AB and circumcircle C_t . Then, the angle α opposite to AB is equal to 120° if and only if the distance between the midpoint of AB and the center of the circumcircle is equal to $r/2$, where r is the radius of the circumcircle C_t (see Figure 2).*

Proof: Since from elementary geometry all the triangles $t(A,B,C)$ with fixed edge AB and vertex C over the same arc AB of the circumcircle C_t have identical angle α opposite to AB , the result will be proved for the particular case, where $t(A,B,C)$ is an isosceles triangle of longest edge AB and smallest edges AC and BC (length of AC equal to length of BC) where the vertex C is over the smallest arc AB (note that the case where C is over the biggest arc implies that AB is not the longest edge of triangle ABC) as shown in Figure 2. Clearly in this case the triangle CBM is similar to triangle OMB where M is the midpoint of edge AB (which implies that $\cos \frac{\alpha}{2} = \frac{1}{2}$ and $\alpha = 120^\circ$) if and only if the length of edge MO is equal to $r/2$. \square

The following theorem is an extension of the known theorem of Thales and characterizes the properties of the diameter circle of any edge AB of any triangle $t(A,B,C)$. This theorem will be used in section 4 to prove Theorems 3 and 4.

Theorem 1 *Let $t(A,B,C)$ be any triangle defined by the vertices A,B,C , the circle C_{AB} the diameter circle of AB , and α the angle of vertex C (opposite to AB). Then (i) the angle α is a right angle if and only if the diameter circle of AB is identical to the circumcircle of $t(A,B,C)$ (the vertex C lies on the arc AB as stated in the theorem of Thales), (ii) the angle α is an acute angle if and only if the vertex C lies outside C_{AB} , and (iii) the angle α is an obtuse angle if and only if the vertex C lies inside C_{AB} .*

Proof: The case (i) corresponds to the theorem of Thales. In order to prove the case (ii), let us consider any triangle AQB with vertex Q in the exterior of C_{AB} as shown in Figure 3(a). Then the geometric median joining Q with the midpoint of AB intersects the arc AB in a point C' which defines a right triangle $AC'B$ whose angles of vertices A and B are respectively smaller than the angles of vertices A and B of the triangle ABQ .

This implies that angle AQB is smaller than the angle $AC'B$ and the result follows. In order to prove case (iii), let us consider any triangle AQB with vertex Q in the interior of C_{AB} as shown in Figure 3(b). The line segment defined by the geometric median joining the midpoint M of AB and C (in direction MC) intersects arc AB in a point C' (outside the triangle) which defines a right triangle $AC'B$ whose angles of vertices A and B are respectively greater than the angles of vertices A and B of the triangle ABQ . This implies that the angle of vertex Q is greater than 90° and the result follows. \square

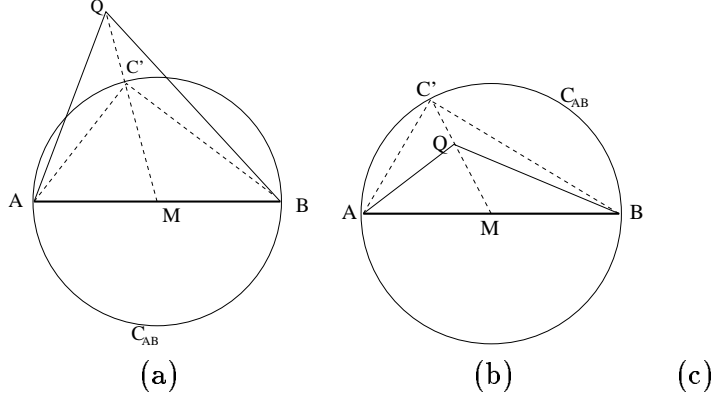


Figure 3: (a) C_t is identical to the diameter circle C_{AB} , then $\alpha = 90^\circ$ (b) C is outside C_{AB} , then $\alpha < 90^\circ$ (c) C is inside C_{AB} , then $\alpha > 90^\circ$)

2.4 The Lepp(t)

This section reviews the Lepp concept and summarizes some geometrical properties [10, 11, 9]. This applies over general conforming unstructured triangulations, where a triangulation is conforming if pairs of adjacent triangles have either a common vertex or a common edge.

Definition 6 For any triangle t_0 of any conforming triangulation τ , the Longest-Edge Propagation Path of t_0 (Lepp(t_0)) will be the ordered list of all the triangles $t_0, t_1, t_2, \dots, t_{n-1}, t_n$, such that t_i is the neighbor triangle of t_{i-1} by the longest edge of t_{i-1} , for $i = 1, 2, \dots, n$.

Proposition 2 For any conforming triangulation τ the following properties hold: (a) for any t , the Lepp(t) is always finite; (b) The triangles t_0, t_1, \dots, t_{n-1} , have strictly increasing longest edge (if $n > 1$); (c) For the triangle t_n of the Longest-Edge Propagation Path of any triangle t_0 , it holds that either: (i) t_n has its longest edge along the boundary, and this is greater than the longest edge of t_{n-1} , or (ii) t_n and t_{n-1} share the same common longest edge.

Definition 7 Two adjacent triangles (t, t^*) will be called a pair of terminal triangles if they share a common longest edge.

Definition 8 For any given triangulation τ , any interior edge l will be called a *terminal edge* in τ if this edge corresponds to the common longest edge of the two triangles that shares the edge l (l is the common edge of a pair of terminal-triangles).

Note that the Lepp of any triangle t corresponds to an associated polygon (shadowed in Figure 4), which in certain sense measures the local quality of the current point distribution induced by t . To illustrate these ideas, see Figure 4, where the Lepp of t_0 corresponds to the ordered list of triangles $(t_0, t_1, t_2, t_3, t_4)$. Moreover the pair (t_3, t_4) is a pair of terminal triangles in the mesh.

The definition 6 should be slightly modified to consider the case where the longest edge is not unique. In such a case, the longest edge that produces the shortest path should be selected.

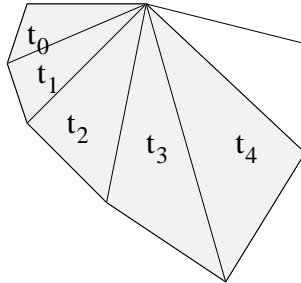


Figure 4: Longest-edge propagation path of t_0

3 Lepp-Delaunay improvement triangulation algorithm and properties

This section describes briefly the algorithm we are using to generate good quality CDT. For a detailed discussion of the algorithm and its properties [10, 9]. The algorithm receives as input any CDT τ and the value of the smallest angle δ , and produces an output triangulation whose angles are bounded between δ and $180^\circ - 2\delta$, excepting for the smallest boundary (interface) constrained angles.

The improvement algorithm uses two basic point insertion operations:

1. **Terminal-edge point insertion.** This operation refers to the Delaunay insertion of the midpoint of the terminal edge of $\text{Lepp}(t)$, whose main goal is the local improvement of the point distribution in the interior of the 2-dimensional geometry.
2. **Boundary point insertion.** This operation refers to the Delaunay insertion of the midpoint of a boundary edge, which is in turn the edge of a boundary triangle whenever this boundary triangle is the first boundary triangle with interior smallest edge in the current $\text{Lepp}(t)$. Note that the main goal of this point insertion

operation is the local improvement of the point distribution over the boundary of the geometry.

For an illustration of the use of the Terminal-edge point insertion operation, see Figure 5 where the triangulation (a) is the initial Delaunay triangulation with $\text{Lepp}(t_0) = t_0, t_1, t_2, t_3$, and the triangulation (b), (c) and (d) illustrate the complete sequence of point insertions needed to improve t_0 . In this example, the improvement (modification) of t_0 implies the automatic Delaunay insertion of three additional Steiner points. Each one of these points is the midpoint of the terminal-edge of the current $\text{Lepp}(t_0)$.

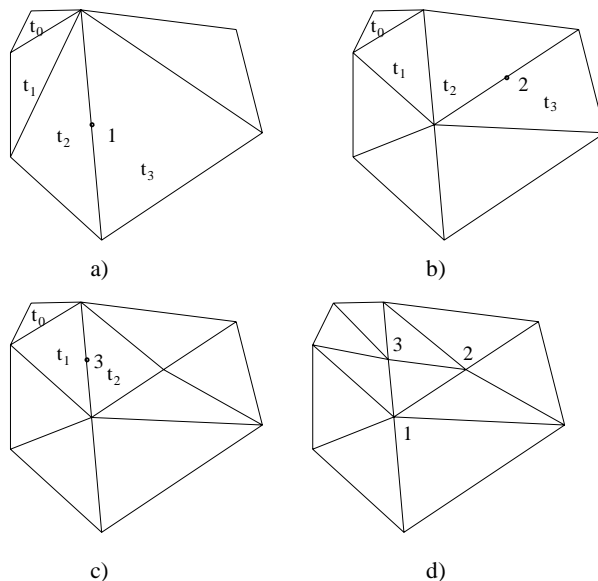


Figure 5: Backward Longest-Edge Delaunay improvement of triangle t_0

For an illustration of the Boundary point insertion operation consider the simple example of Figure 6(a). In this case the naive use of the Lepp, i.e, the insertion of the midpoint of the terminal-edge would produce undesirable interior points (as shown in Figure 6(b)). The boundary point insertion operation as described above produces an adequate point distribution as shown in Figure 6(c).

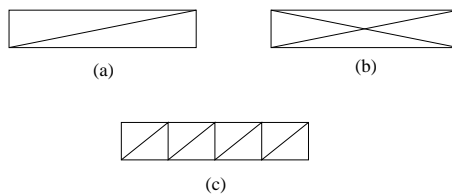


Figure 6: Boundary treatment technique

The overall improvement algorithm can be formulated as follows:

Lepp-Delaunay-quality-triangulation (τ, δ)
Input: A CDT τ of any PSLG domain; and a tolerance parameter ($\delta < 30^\circ$)
Find S, the set of the bad triangles t of τ (of smallest angle less than δ)
for each t in S **do**
 Lepp-Delaunay-Improvement (τ, t)
 Update the set S (by adding the new small-angled triangles and eliminating those destroyed throughout the process)
end for
Lepp-Delaunay-Improvement (τ, t)
while t remains without being modified **do**
 Find the Lepp(t)
 Find the first boundary triangle t^* in Lepp(t)
 with boundary edge l not equal to the smallest edge of t^*
 if l exists **then**
 select p , the midpoint of l
 else
 select p midpoint of the terminal-edge of Lepp(t)
 end if
 Perform the Delaunay insertion of p
end while

Remarks

1. The set S of τ does not consider triangles with boundary constrained angles because they cannot be improved in the mesh.
2. δ is a parameter less than or equal to 30° that can be easily adjusted.
3. As it was pointed out in [9, 10, 12] the processing order of the triangles in the set S is irrelevant from a practical point of view. Furthermore, the algorithm has a kind of selfcorrective property in the sense that the initial (nonordered) processing of any small subset of S indeed destroys and improves a big subset of the worst triangles of S.

We have used the word improvement instead of bisection or refinement. This is to explicit the fact that one step of the procedure does not necessarily produce a smaller triangle. More important however, is the fact that the Lepp-Delaunay-Improvement algorithm improves the triangle in the sense of Theorem 2 [10, 11].

Theorem 2 *For any Delaunay triangulation τ , the repetitive use of the Lepp-Delaunay-quality triangulation algorithm (with threshold parameter $\delta = 30^\circ$) produces a quality triangulation of smallest angle greater than or equal to 30° , excepting occasionally some isolated angles $22.2^\circ < \alpha < 30^\circ$ related with nonfrequent geometric conditions and boundary restrictions.*

Remark: In practice almost every CDT can be improved using the previous algorithm, with threshold parameter $\delta = 30^\circ$, producing a mesh whose internal angles are bounded by 30° and 120° .

4 Nonobtuse boundary Delaunay triangulations

In this section we present the algorithm to eliminate boundary (interface) obtuse triangles and prove its properties. In particular, it is proved that the resulting triangulation is a non-constrained Delaunay triangulation over the interface edges (triangles sharing an interface edge satisfy the Delaunay condition).

We present the algorithm divided into two cases according to the domain complexity: (a) simple polygonal domains which may include holes, and (b) polygonal domains with interfaces (PSLG inputs). In case (a) only isolated 1-edge and 2-edge boundary obtuse triangles must be considered and, in case (b), apart from the triangles of case (a), 1-edge and 2-edge interface obtuse triangles either isolated or inside a group of adjacent of interface triangles must be handled.

4.1 Nonobtuse boundary triangulation of simple polygonal domains

Triangulations of simple polygonal domains present two cases of boundary obtuse triangles: triangles with 1-boundary edge and triangles with 2-boundary edges. In this section, we prove that the elimination of 1-edge boundary obtuse triangles is done by the Delaunay insertion of one point, and the elimination of 2-edge boundary obtuse triangles requires the Delaunay insertion of a finite number of points that depends on the geometry of the target triangle.

4.1.1 1-edge boundary obtuse triangles

In order to demonstrate that 1-edge boundary obtuse triangles can be eliminated by the Delaunay insertion of one point, we first characterize the 1-edge boundary obtuse triangles, and then we demonstrate that the Delaunay insertion of the boundary edge midpoint eliminates the obtuse angle not generating new boundary obtuse triangles.

The 1-edge boundary obtuse triangles are described in the following theorem:

Theorem 3 *Let τ be any quality CDT of any PSLG geometry with internal angles bounded by 30° and 120° . Then any 1-edge boundary obtuse triangle $t(A, B, C)$ in τ of boundary longest-edge AB has vertex C located in the region R , limited by C_{AB} and the lines l_1, l_2 (respectively intersecting C_{AB} in the points F and G as shown in Figure 7), where lines l_1 and l_2 are defined so that the angles FBA and GAB are equal to 30° , respectively, and C_{AB} is the diameter circle of AB .*

Proof: In order to define an obtuse triangle with all angles greater than 30° , the vertex C must be inside the region R because of (1) if the vertex C would be outside the diameter circle C_{AB} , the angle of vertex C would be acute (see Theorem 1); and (2) if the vertex C would be located under l_1 or under l_2 , the angle FBA or GAB would be less than 30° . Note that the largest obtuse angle 120° is produced when C becomes equal to E . \square .

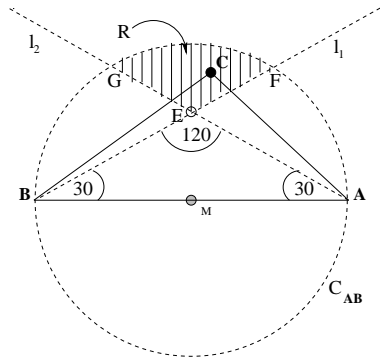


Figure 7: R defines the geometric region for the vertex C so that ABC is a quality obtuse triangle

The elimination of any 1-edge boundary obtuse triangle is done as described in the following theorem:

Theorem 4 *Let τ be any quality CDT of any PSLG geometry with angles bounded by 30° and 120° . Let $t(A,B,C)$ be a 1-edge boundary obtuse triangle in τ with obtuse angle of vertex C and unique boundary edge AB . Then the Delaunay insertion of the midpoint M of AB eliminates the obtuse angle BCA not generating new boundary obtuse triangles.*

Proof: In order to prove this theorem, we shall show that (1) the insertion of the midpoint M of AB generates two nonobtuse boundary triangles CBM and CMA of respective boundary edges BM and MA as shown in Figure 8; and (2) in case that edge swapping of edges AB and/or BC are required, the 1-edge boundary triangles of boundary edges MA or BM (see Figure 9) are nonobtuse boundary triangles.

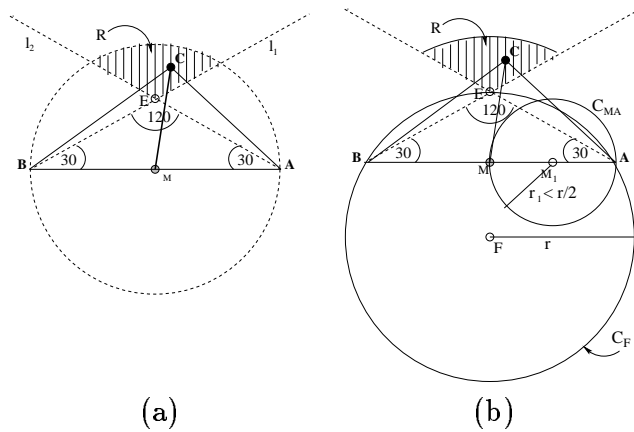


Figure 8: (a) R defines the geometric region for the vertex C for the quality obtuse triangle ABC (b) Diameter circle C_{MA} does not intersect R assuring that the triangle CMA is a nonobtuse triangle

Clearly only one of the two triangles CBM and CMA in Figure 8(a) generated by the bisection of the boundary edge AB can be an obtuse triangle of vertex C. Without loss of generality, let assume that CBM is the nonobtuse boundary triangle and CMA is an obtuse angle. According to Theorem 1(iii), this implies that C is in the interior of the diameter circle C_{MA} of radius r_1 equal to half the length of edge MA (see Figure 8(b)). This fact is impossible because the shortest distance d between any point C in R and the center M_1 of C_{MA} (produced when C lies on l_1 and CM_1 is orthogonal to l_1) is greater than r_1 (by elementary geometry d is equal to $\frac{3}{2}r_1$).

Let now assume that the circumcircle of triangle MAC includes the point D and consequently there exist a triangle CAD (see Figure 9) which shares the edge CA with triangle CMA so that an edge swapping occurs between MD and CA (MD replaces MA) and a new 1-edge boundary triangle AMD is generated. We shall show that triangle ADM is nonobtuse of vertex D. By hypothesis, every internal angle is greater than or equal to 30° , then the vertex D must be located in the region Ω limited by lines l_3 and l_4 (see Figure 9(a)), where l_3 contains the vertex A and forms an angle of 60° with the boundary edge MA, and l_4 is defined by the vertices E and F, which respectively correspond to the vertices C and D, when the angles BCA and CDA are equal to 120° . The shortest distance d between the center M_1 of the diameter circle C_{MA} and any point D of region Ω occurs when the angle of vertex D is equal to 120° (D is on line l_4). By using Proposition 1, d is equal to $\frac{r}{2}$ (where r is the radius of the circumcircle of the triangle ABE), then D is always outside the C_{MA} because the radius of C_{MA} is less than $\frac{r}{2}$. Since D is in the exterior of C_{MA} , the triangle AMD is a nonobtuse boundary triangle (Theorem 1(ii)). The case of an edge swapping of CB is symmetric. \square

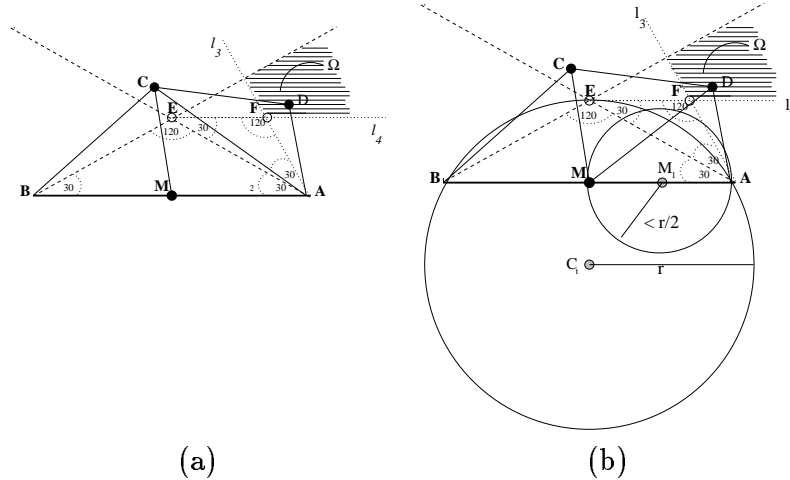


Figure 9: (a)The shadow region Ω shows the location of the vertex D so that an edge swapping is possible (b) edge swapping (AC to MD) does not produce a new boundary obtuse triangle

Corollary 5 For any quality CDT of any PSLG geometry (without interfaces) with angles lies between 30° and 120° , the number of point insertions (N_{1b}) required to eliminate N 1-edge boundary obtuse triangles is equal to N .

4.1.2 Triangles with two boundary edges

The elimination of 2-edge boundary obtuse triangles can be divided into two cases:

1. The smallest and the longest edge of the triangle are boundary edges, and the medium size edge is an internal edge, as illustrated by triangle ABC in Figure 10(a). (The vertex C must belong to the region R otherwise the length BC would be greater than the length of CA.) The strategy presented in theorem 4 also applies to this case since the Delaunay insertion of the midpoint of the edge AB can only produce an obtuse angle (δ) opposite to an internal edge, but not opposite to a boundary edge. Then, this operation does not create a new boundary obtuse triangle. (α is greater than or equal to 30° because of the previous application of the Lepp improvement procedure).

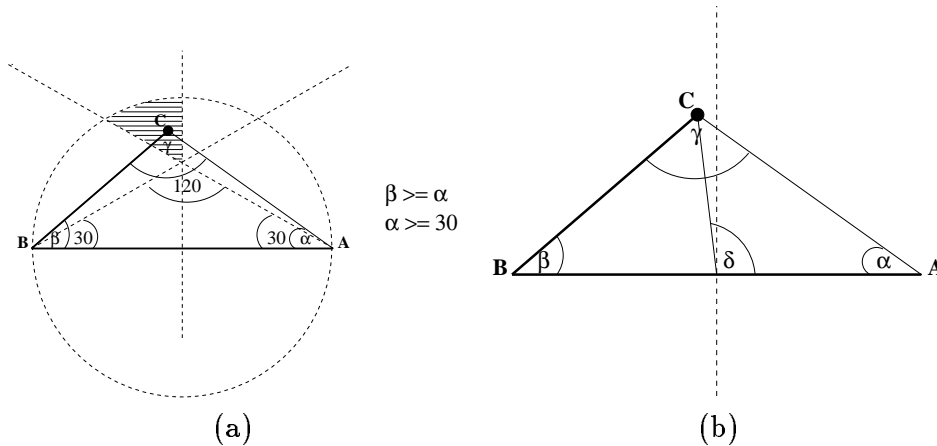


Figure 10: (a) C belongs to the Region R (b) insertion of the midpoint of AB

2. The medium size edge and the longest edge of the triangle are boundary edges, and the smallest edge is an interior edge opposite to the boundary constrained angle β (the smallest angle of the triangle). In this case, we can not apply the strategy described in theorem 4 because after two applications of the strategy, a new triangle similar to the target triangle will be obtained, as shown in Figure 11 (t_o is similar to t_4). One additional problem is that the boundary constrained angle can be less than 30° and consequently, the obtuse angle can be greater than 120° .

The essential ideas of the strategy to handle case 2 are as follows: An 2-edge boundary isosceles triangle of boundary edges equal to half the smallest boundary edge of the target triangle is constructed (triangle AMN in Figure 12(b)) by Delaunay insertion of the points

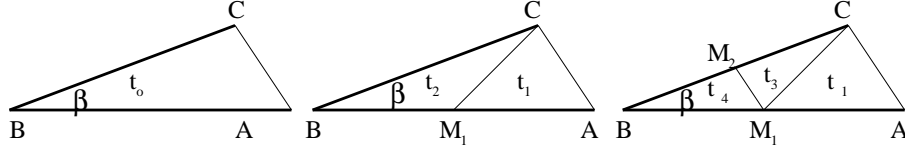


Figure 11: t_o is similar to t_4

M and N. This construction can produce an 1-edge boundary obtuse triangle t_1 , which is in turn destroyed by the Delaunay insertion of the midpoint of the longest edge of t_1 (Figure 12(c)). Since t_1 might have maximum angle greater than 120° , the elimination of t_1 can again produce a new boundary obtuse triangle t_1 , with largest angle smaller than the previous one and so on. The boundary obtuse triangles are finally eliminated after the insertion of a finite number of points. The next algorithm implements this strategy:

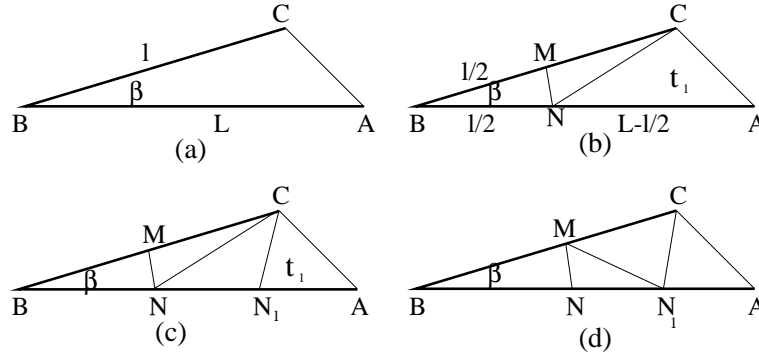


Figure 12: Elimination of 2-edge boundary obtuse triangles

Eliminate-2-edge-boundary-obtuse-triangle(t_2, τ)

Input: t_2 is a 2-edge boundary obtuse triangle with smallest interior edge and τ is the current triangulation (Figure 12)

Compute the midpoint M of the smallest boundary edge of t_2

Compute the point N so that the length of segment BM

is equal to the length of segment BN (see Figure 12(b))

Perform the Delaunay insertion of N and M (see Figure 12(b))

(This reduces to join points N and M, and points N and C)

$S = \Phi$

if triangle t_1 of vertices NAC is a 1-edge boundary obtuse triangle **then**

$S = \{t_1\}$

end if

while S is not empty **do**

Get any triangle t_1 of S

Perform Delaunay insertion of the longest edge midpoint of t_1

Update S with the new 1-edge boundary obtuse triangles

end while

Remark: Since the previous algorithm does not bisect the longest boundary edge, the angle MCN can be less than the boundary constrained angle β (see Figure 12(b)), which for the longest-edge partition of an obtuse triangle is guaranteed to be greater than β . Then, the final triangulation can have angles smaller than the boundary constrained angles in the neighborhood of the 2-edge boundary isosceles triangles.

The next theorem computes an upper bound for the number of points inserted using the previous algorithm which depends on the boundary constrained angle and on the lengths of the boundary edges. It is worth to point out that the proof of this theorem will be used later to determine upper bounds of the number of point insertions in the elimination of interface obtuse triangles.

Theorem 6 *Let t be a 2-edge boundary obtuse triangle having the boundary medium size edge and the boundary longest edge of respective lengths l and L and boundary constrained angle β . Then the algorithm produces a set of nonobtuse boundary triangles by Delaunay insertion of a number of points bounded by N , where $N = \lceil 2 + \frac{2L-l}{l\sqrt{1-\cos(\beta)}} \rceil$.*

Proof. In order to eliminate a 2-edge boundary obtuse triangle with smallest interior edge (β is the smallest angle of the triangle), first two points N_0 and M_0 are inserted so that B, N_0, M_0 is a 2-edge boundary isosceles triangle of two equal boundary edges BN_0 and BM_0 (see Figure 13). Then, the 1-edge boundary obtuse triangles generated inside the quadrilateral C, M_0, N_0, A are eliminated by the Delaunay insertion of points on the boundary edges. An upper bound of the number of point insertions is obtained by using the fact that no more point insertions are required whenever the lengths of the boundary edges of the new 1-edge boundary triangles have a size less than or equal $\sqrt{2}e$, where $e = N_0M_0$ is the smallest edge of the quadrilateral, because for any 1-edge boundary obtuse triangle with c, b, a as the respective lengths of the longest, medium size and smallest edge, holds that $c^2 \geq a^2 + b^2 \geq 2e^2$ and $c \geq \sqrt{2}e$. Subsequently, the number of point insertions on each boundary edge is bounded by $n_e = \lceil \frac{|A-N_0|}{\sqrt{2}|M_0-N_0|} \rceil = \lceil \frac{L-l/2}{l\sqrt{1-\cos(\beta)}} \rceil$ ($|A-N_0| = L - l/2$ and using the cosine theorem $\sqrt{2}|M_0-N_0| = l\sqrt{1-\cos(\beta)}$). Note that the point insertions done on the boundary edges M_0C and N_0A never destroys the isosceles triangle BM_0N_0 , because either the angle $M_jM_0N_0$ is obtuse or the angle $N_iN_0M_0$ is obtuse. This means that the internal edge M_jN_0 or N_iM_0 is the longest edge of the triangle that contains the edge M_0N_0 , then the edge swapping operation is applied over M_jN_0 or N_iM_0 . The final expression considers then twice n_e (one for each edge) and the two points M_0 and N_0 . \square

Corollary 7 *The number of points inserted (V_{2b}) to eliminate N 2-edge boundary obtuse triangles $t_j, 1 \leq j \leq N$, where each t_j has boundary longest edge L_j , boundary medium size edge l_j and boundary constrained angle β_j , is:*

$$V_{2b} \leq 2N + \sum_{j=1}^N \lceil \frac{2L_j - l_j}{l_j\sqrt{1-\cos(\beta_j)}} \rceil$$

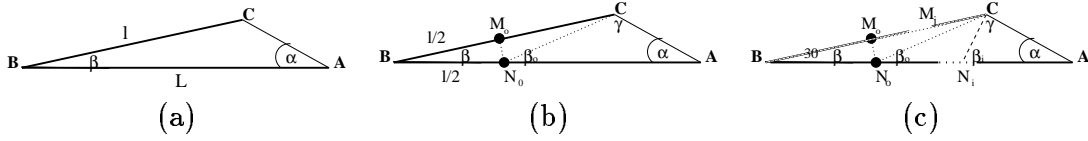


Figure 13: Boundary obtuse triangle with a constrained angle less than β

Proof: The previous expression corresponds to the sum of the points inserted in each 2-edge boundary obtuse triangle. \square

In order to show that a small number of point insertions is indeed needed in practice let us discuss a particular case of Theorem 6 where only two points are required to eliminate 2-edge boundary obtuse triangles. This case is described in the following proposition:

Proposition 3 *Let t be a 2-edge boundary obtuse triangle with smallest interior edge. If the boundary constrained angle β is greater than or equal to $\beta_0 = 32.54^\circ$, the boundary obtuse triangle is eliminated by the Delaunay insertion of the two points N and M that forms the 2-edge boundary isosceles triangle $t(B,M,N)$.*

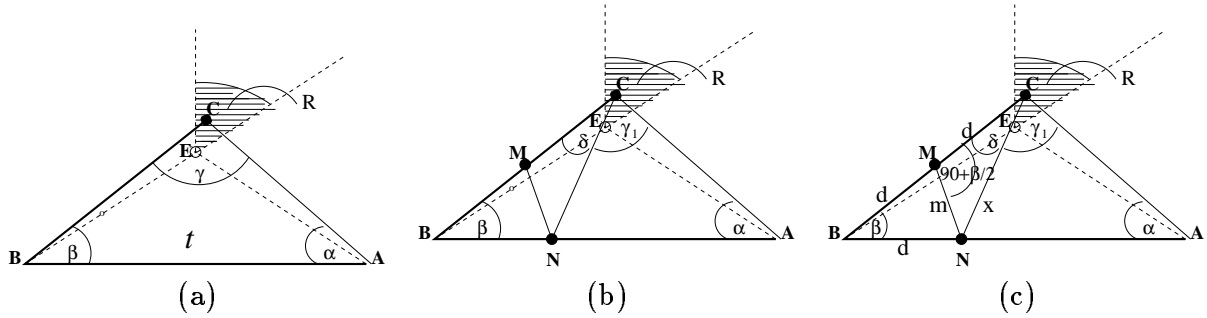


Figure 14: Parameters in the computation of a lower bound for β

Proof: Let be $t(A,B,C)$ a 2-edge boundary obtuse triangle with obtuse angle of vertex C , longest-edge AB , and medium edge BC as shown in Figure 14(a). Since β_0 is a lower bound of β , we have to identify the triangle with smallest value of β and the largest value of γ , for which the insertion of N and M generates a triangle NCA with angle $\gamma_1 \leq 90^\circ$ (Figure 14(b)). Note that the largest value of γ is used because it produces the largest value of γ_1 . By using the isosceles properties of triangle BNM and the cosine theorem, we obtain the following three equations, respectively for the angles β , $90 + \frac{\beta}{2}$ and δ :

$$m^2 = 2d^2 - 2d^2 \cos(\beta)$$

$$x^2 = m^2 + d^2 - 2md \cos\left(90 + \frac{\beta}{2}\right)$$

$$m^2 = d^2 + x^2 - 2dx \cos(\delta)$$

Let us assume that d is equal to 1, since the result is valid for any similar triangle. Then, the previous three equations allow us to compute δ for any given value of β . The value of γ_1 can be computed by adding the interior angles of the triangle t and replacing γ by $\delta + \gamma_1$. Then, the expression for γ_1 is as follows:

$$\gamma_1 = 180 - \alpha - \beta - \delta$$

For a fixed value of β , the biggest value of γ_1 is obtained when $\alpha = \beta$ (remember that for these triangles α must be greater than or equal to β). Then, the new expression for γ_1 is:

$$\gamma_1 = 180 - 2\beta - \delta \leq 90^\circ$$

Numerically, we have obtained that a lower bound of β (β_0) is 32.54° . Then, for any value of β greater than or equal to $\beta_0 = 32.54^\circ$, $\gamma_1 \geq 90^\circ$. \square

4.2 Nonobtuse boundary (interface) triangulations of PSLG inputs

The elimination of interface obtuse triangles of a quality CDT is in particular a difficult task when interface obtuse triangles are arranged into groups. In order to discuss the strategies designed to solve the different cases that arise when the domain includes interfaces, we consider four cases: (a) 1-edge interface obtuse triangles, (b) 2-edge interface obtuse triangles, (c) adjacent 2-edge obtuse triangles, and (d) 1-edge interface obtuse triangles adjacent to a 2-edge interface triangle.

4.2.1 Two 1-edge interface obtuse triangles share the interface edge

There exist two cases of 1-edge interface obtuse triangles: (1) two 1-edge interface triangles share the interface edge, and (2) a 1-edge interface obtuse triangle shares the interface edge with a 2-edge interface triangle. In this section, we consider only case (1), because case (2) requires a different strategy that will be discussed in section 4.2.4.

Figure 15 shows two 1-edge interface obtuse triangles. Note that the Delaunay insertion of the midpoint on the interface (common) edge AB destroys the two obtuse angles and does not generate new obtuse angles opposite to the interface edge because the vertices C and D are outside the diameter circles C_{MA} and C_{MB} . (Theorem 4 also applies to this case with the only difference that the insertion of one point may destroy one or two boundary obtuse angles.)

Proposition 4 *The number of vertices (V_{1i}) inserted to eliminate N 1-edge interface obtuse triangles is bounded as follows: $\frac{N}{2} \leq V_{1i} \leq N$*

Proof: The lowest value of V_{1_i} (equal to $\frac{N}{2}$) is obtained when V_{1_i} is an even number and each interface edge is shared by two interface obtuse triangles. The highest value of V_{1_i} (equal to N) is obtained when each interface edge is opposite to only one obtuse angle. \square

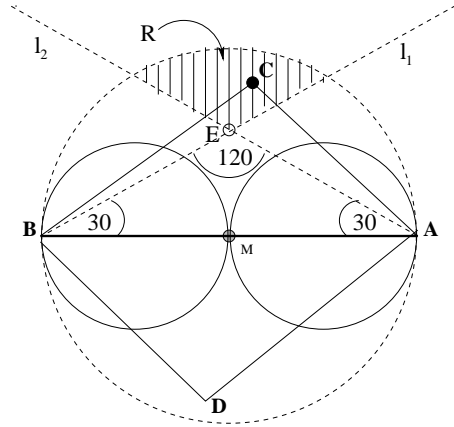


Figure 15: Obtuse angles opposite to an interface edge

4.2.2 2-edge interface obtuse triangles

Isolated 2-edge interface obtuse triangles, i.e, triangles whose interface edges are not shared with other 2-edge interface triangles, are eliminated by using the strategy applied to 2-edge boundary obtuse triangles. This strategy inserts points on the interface edges so that the local triangulation inside the 2-edge interface obtuse triangle t is a nonobtuse boundary triangulation. Since now the points are inserted on interface edges instead of boundary edges, the 1-edge interface triangles adjacent to t are destroyed and new 1-edge interface triangles generated, with angles opposite to the interface edges smaller than the original angles. Then, the new 1-edge interface triangles are nonobtuse interface triangles, because the original 1-edge interface triangles adjacent to t were also nonobtuse interface triangles.

4.2.3 Adjacent 2-edge interface obtuse triangles

Adjacent 2-edge interface obtuse triangles can be produced by (a) a chain of connected interface edges (AB, BC, DE, EF, FG and GH) where the interface constrained angle between each pair of consecutive edges is small (Figure 16(a)), and (b) several interface edges AB_i that converge to a common vertex A (Figure 16(b)).

The case (a) is solved by inserting points on the interface edges of the 2-edge obtuse interface triangles in the same way as for isolated 2-edge interface obtuse triangles. Note that in this case a sequence of point insertions might be needed over the chain of interface edges, because the Delaunay insertion of points to destroy one 2-edge interface triangle might generate a new 2-edge interface obtuse triangle. In order to illustrate this case,

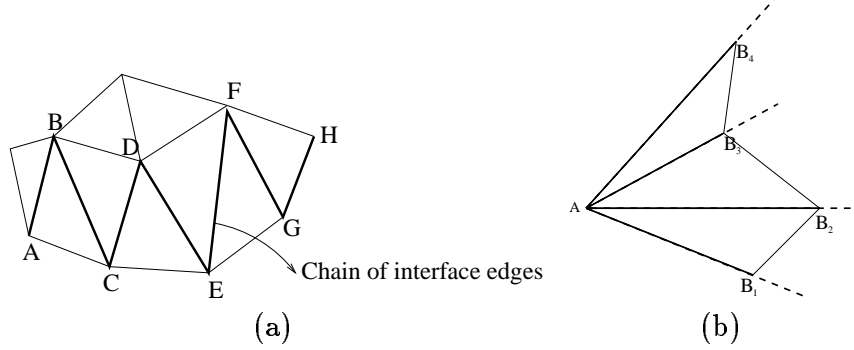


Figure 16: (a) 2-edge interface triangles formed by a chain of connected interface edges (b) interface edges that converge to a common vertex

let us assume that triangle EFG in Figure 16(a) is obtuse of vertex G. Then, the points I and J are inserted to form the 2-edge interface isosceles triangle IJF as shown in Figure 17(a). Note that the point J generates a new 2-edge interface obtuse triangle EJD which is destroyed by the Delaunay insertion of two new points K and L that forms 2-edge interface isosceles triangle KLE as shown in Figure 17(b). The number of inserted points is finite, and in the worst case involves all the interface triangles of the chain.

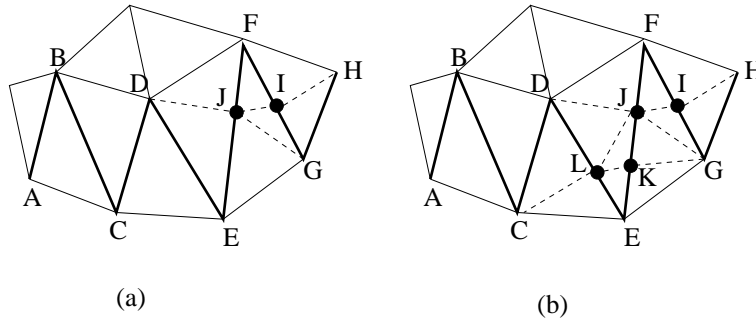


Figure 17: (a) Elimination of the 2-edge interface triangle EFG (b) Elimination of the new generated 2-edge interface obtuse triangle EJD

The case (b) requires a global strategy over all the 2-edges interface triangles of the group because the one by one elimination of 2-edge interface triangles in general produces an infinite insertion of points. Consequently, we propose the elimination of the interface obtuse triangles by computing the midpoint M of the smallest interface edge of the group and by Delaunay inserting a point N_j on each edge j so that the distance between N_j and A is equal to the distance between M and A (a set of isosceles triangles are generated around A as shown in Figure 18); we then eliminate each 1-edge interface triangle by Delaunay insertion of the midpoint of its interface edge until no new 1-edge interface obtuse triangles are generated. Since the interface obtuse triangles are adjacent, the number of points inserted on shared edges is defined by the triangle that requires the

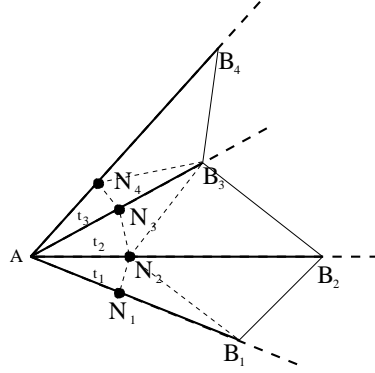


Figure 18: 2-edge interface triangles formed by interface edges converging to a common vertex A

bigger number of point insertions.

Note that the strategy of case (b) must be applied to any group of adjacent 2-edge interface triangles where at least one of them is a 2-edge interface obtuse triangle because if the point insertions destroy only the 2-edge interface obtuse triangles of the group, new 2-edge interface obtuse triangles might be generated in the adjacent interface nonobtuse triangles.

Theorem 8 *The number of vertices V_{2a} inserted to eliminate N adjacent 2-edge interface triangles that share a vertex A, where at least one of them is a 2-edge interface obtuse triangle (Figure 18) is bounded by:*

$$V_{2a}(A) \leq (N + 1)(1 + \max_{1 \leq j \leq N} NV(t_j))$$

$$NV(t_j) = \lceil \max\left(\frac{|B_j - N_j|}{|N_{j+1} - N_j|}, \frac{|B_{j+1} - N_{j+1}|}{|N_{j+1} - N_j|}\right) \rceil, 1 \leq j \leq N$$

Proof: It is easy to see that the number of point insertions required to build the N isosceles triangles requires the Delaunay insertion of $(N+1)$ points around vertex A, as illustrated in Figure 18. This step might generate one or more 1-edge interface obtuse triangles with interface edge $N_i B_i$, $1 \leq i \leq (N+1)$. Each 1-edge interface obtuse triangle is destroyed by Delaunay insertion of the midpoint of its interface edge until no one new 1-edge interface obtuse triangle is generated. If the 2-edge interface triangle were isolated (Theorem 6), the number of point insertions on each interface edge of triangle t_j would be at most $NV(t_j) = \lceil \max\left(\frac{|B_j - N_j|}{|N_{j+1} - N_j|}, \frac{|B_{j+1} - N_{j+1}|}{|N_{j+1} - N_j|}\right) \rceil, 1 \leq j \leq N$. Since each interface edge $N_i B_i$ is shared by two quadrilaterals, this normally requires two different number of point insertions $NV(t_j)$ and $NV(t_{j+1})$. Then, an upper bound to the number of point insertions on each edge is the value given by the maximum value of $NV(t_j)$. As the number of edges are $N + 1$, an upper bound for $V_{2a}(A)$ is obtained by multiplying the number of edges by $\max(NV(t_j))$. \square

4.2.4 1-edge interface obtuse triangles adjacent to 2-edge interface triangles

The elimination of a 1-edge interface obtuse triangle $t(A, B, C)$ whose interface edge (AB) also belongs to an isolated 2-edge interface (obtuse or nonobtuse) triangle $t'(A, B, C')$ with smallest interior edge (Figure 19(a)) can not be done by the insertion of one point (the midpoint M of edge AB) [13], because this point will probably generate a new 2-edge boundary obtuse triangle $t_1(A, M, C')$ (see Figure 19(b)). In this case, the elimination of t_1 would require the insertion of several points, those required to eliminate a 2-edge interface obtuse triangle. In particular, a point between A and M would be inserted to generate a new 2-edge interface isosceles triangle.

In order insert as few points as possible, t is destroyed indirectly by inserting two points (N, M) in the 2-edge interface triangle t' so that ANM is an isosceles triangle (see Figure 19(c)). Note that this strategy always eliminates the 1-edge interface obtuse triangle t and does not generate a new 1-edge interface obtuse triangle ANC even in the case N is not the midpoint of AB because the angle ACN (Figure 19(c)) is smaller than the angle ACM (Figure 19(d)). In the case that ABC' is a nonobtuse triangle, it is not necessary to insert additional points along NB y MC because neither triangle $NC'B$ (Figure 19(c)) nor triangle NBC (Figure 19(d)) are 1-edge interface obtuse triangles. In case ABC' is obtuse, the number of point insertions is bounded by the expression obtained in Theorem 6.

In the case that t is adjacent to a 2-edge interface triangle that belongs to a group of adjacent 2-edge interface triangles (case (b) of section 4.2.3), t is destroyed by generating 2-edge interface isosceles triangles around A . The number of point insertions is bounded by the number of points computed in Theorem 8.

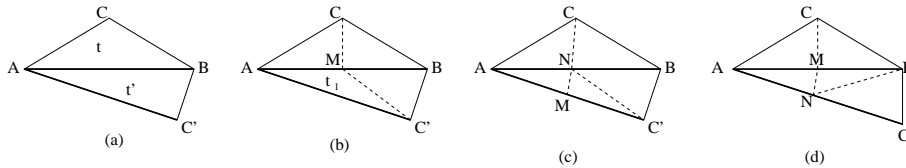


Figure 19: 1-edge interface obtuse triangle adjacent to a 2-edge interface triangle

5 Important properties of the algorithm

The following theorem follows directly from the results of theorems 4, 6, 8.

Theorem 9 *Let τ be a quality triangulation with N boundary triangles and M interface triangles, where N_b are boundary obtuse triangles and M_i are interface obtuse triangles. Then, the boundary obtuse triangles are eliminated by inserting a finite number of points.*

Furthermore, the elimination of the boundary (interface) obtuse triangles improves the mesh in the following sense:

Theorem 10 *Let τ be a nonobtuse boundary (interface) triangulation. Then, any nonobtuse boundary (interface) Delaunay triangulation is a non-constrained Delaunay triangulation with respect to triangles that share an interface edge.*

Proof: Let assume that segment AB is an interface edge and C the vertex opposite vertex edge AB. In a nonobtuse boundary Delaunay triangulation, the angles opposite to an interface edge are less than or equal to 90° . Therefore, the center of the circumcircle of t is located in the triangle ABC or in a neighboring triangle through interior edges. The same occurs for the triangle $t'(A, B, C')$ that shares AB with t . Then, the Delaunay criteria is fulfilled because both the circumcircle of t does not include C' and the circumcircle of t' does not include C. Notice that the Delaunay criteria is also fulfilled when the angle on vertex C and the angle on vertex C' are equal to 90° , because the vertices A,B,C and C' are co-circular. \square

6 Examples

This section illustrate the practical behavior of the algorithm using four test examples with different geometrical complexity: the right angled spiral of Figure 20(a); the strip geometry with "interior" interface edge of Figure 21(a), the two circle polygon with additional interior interface edges of Figure 22(a) and the polygon with several constrained angles of Figure 23(a). The geometrical information of these examples is given respectively in Tables 1, 2, 3 and 4. The first column corresponds to the CDT of the vertices, the second one to the quality mesh generated by the Lepp-algorithm, and the third column shows the result of applying the algorithm discussed in this paper. Since the examples 1 and 2 have a minimum boundary (interface) constrained angle equal to 90° , the algorithm that eliminate the boundary (interface) obtuse angles preserves the quality of the input mesh as it can be observed in Tables 1 and 2, respectively. Examples 3 and 4 have boundary (interface) constrained angles less than 15° , and as expected, few triangles with interior angles less than the boundary (interface) constrained angles are introduced in the neighborhood of the 2-edge boundary (interface) obtuse triangles. In each case the number of inserted points is in complete agreement with our theoretical results. This can be appreciated in Table 5 which shows the expected number (for example 1 and 2) or an upper bound (for example 3 and 4) of the number of point insertions versus the number of point insertions obtained in practice. An upper bound is given when the example contains 2-edge boundary (interface) obtuse triangles.

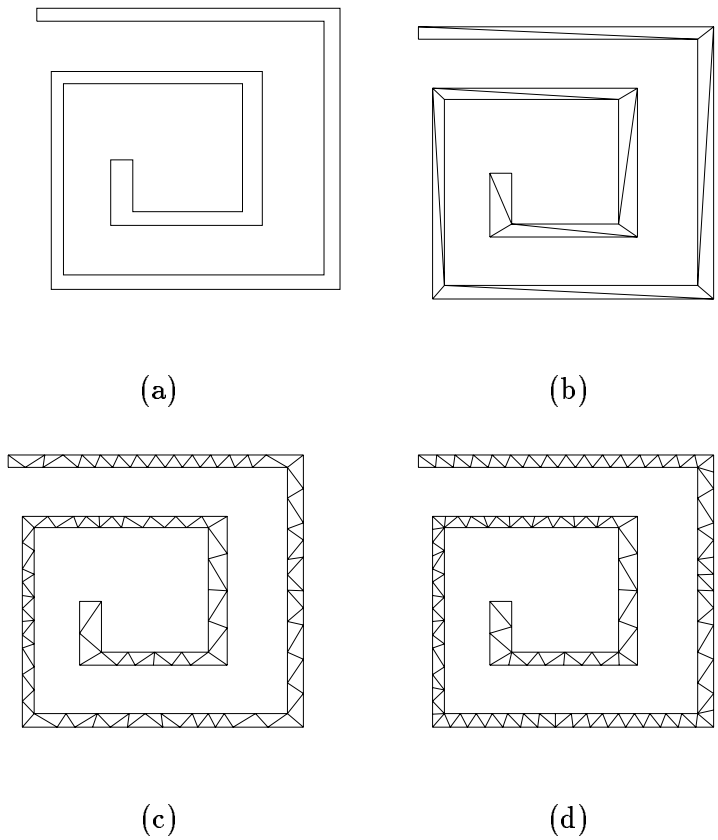


Figure 20: Example 1 (a) geometry (b) CDT of the vertices (c) quality mesh, and (d) nonobtuse boundary (interface) mesh

Example 1			
	CDT of the vertices	quality mesh ($30^\circ \leq \alpha \leq 120^\circ$)	Nonobtuse boundary (interface) mesh
Number of vertices	16	130	158
Number of triangles	18	128	156
Minimum angle	2.59	30.53	30.53
Minimum angle (average)	6.62	40.76	43.31
Maximum angle	145.53	111.03	112.52
Maximum angle (average)	126.00	83.96	80.00
Number of boundary) (interface) obtuse triangles	8	28	0

Table 1: Statistical information for the example 1 (Figure 20)

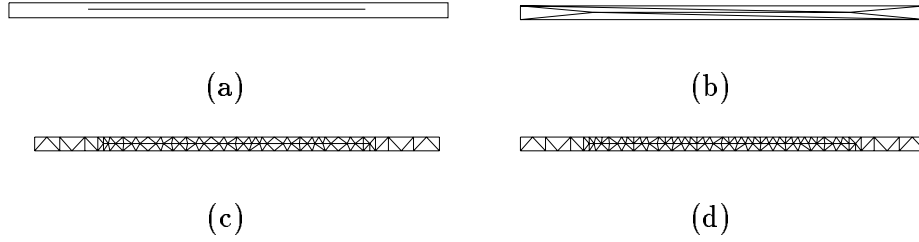


Figure 21: Example 2 (a) geometry (b) CDT of the vertices (c) quality mesh, and (d) nonobtuse boundary (interface) mesh

Example 2			
	CDT of the vertices	Quality mesh ($30^\circ \leq \alpha \leq 120^\circ$)	Nonobtuse boundary (interface) mesh
Number of vertices	6	99	116
Number of triangles	6	128	149
Minimum angle	1.00	30.77	30.77
Minimum angle (average)	4.10	43.53	44.72
Maximum angle	175.52	108.16	106.60
Maximum angle (average)	144.80	83.65	81.68
Number of boundary (interface) obtuse triangles	2	21	0

Table 2: Statistical information for the example 2 (Figure 21)

Example 3 (Minimum geometric constrained angle 14.99°)			
	CDT of the vertices	Quality mesh ($30^\circ \leq \alpha \leq 120^\circ$)	Nonobtuse boundary (interface) mesh
Number of vertices	100	272	291
Number of triangles	104	434	463
Minimum angle	0.84	30.06	12.40 ¹
Minimum angle (average)	15.73	43.15	42.39
Maximum angle	172.49	115.17	126.82
Maximum angle (average)	111.87	79.80	80.49
Number of boundary (interface) obtuse triangles	9	8	0

(1) 16 unconstrained angles less than 30° are produced

Table 3: Statistical information for the example 3 (Figure 22)

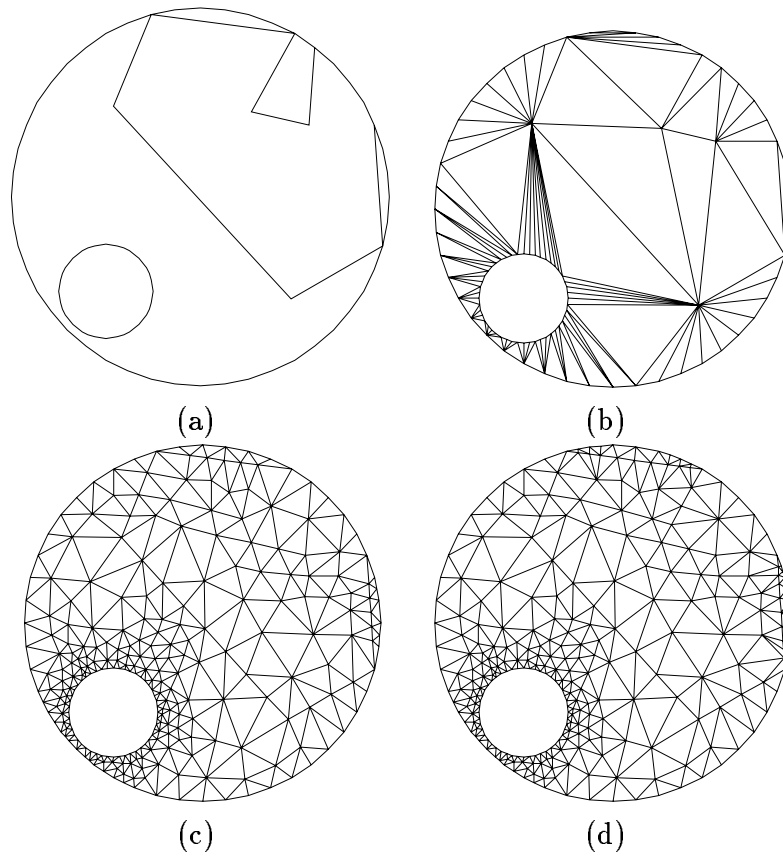


Figure 22: Example 3 (a) geometry (b) CDT of the vertices (c) quality mesh, and (d) nonobtuse boundary (interface) mesh

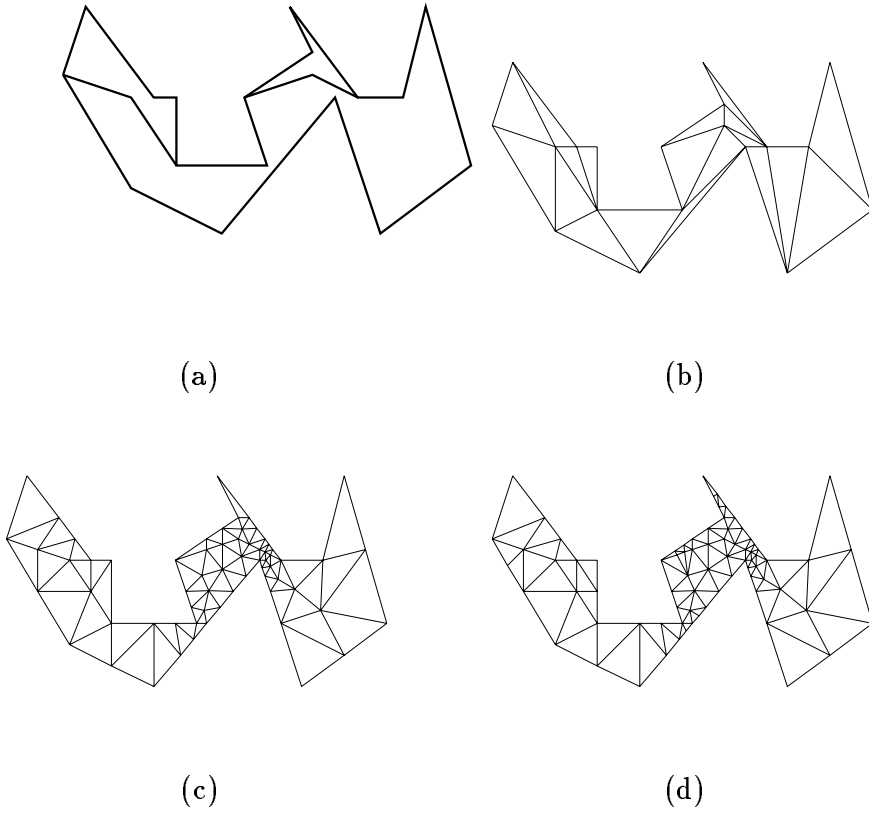


Figure 23: Example 4 (a) geometry (b) CDT of the vertices (c) quality mesh, and (d) nonobtuse boundary (interface) mesh

Example 4 (Minimum geometric constrained angle 10.30°)			
	CDT of the vertices	Quality mesh ($30^\circ \leq \alpha \leq 120^\circ$)	Nonobtuse boundary (interface) mesh
Number of vertices	19	65	77
Number of triangles	18	80	94
Minimum angle	5.19	30.34	17.91^2
Minimum angle (average)	25.91	43.34	41.88
Maximum angle	168.69	112.61	113.62
Maximum angle (average)	105.85	80.29	80.78
Number of boundary (interface) obtuse triangles	10	6	0

(2) 5 nonconstrained angles less than 30° are produced

Table 4: Statistical information for the example 4 (Figure 23)

Number of point insertions during the elimination of boundary (interface) obtuse triangles					
	N_{1b}	N_{1i}	N_{2b} or N_{2i}	Expected or	Inserted
			upper bound or		
Example 1	28	0	0	28	28
Example 2	13	8	0	$10.5 \leq N \leq 21$	17
Example 3	4	0	4	46	19
Example 4	4	0	2	25	12

Table 5: Number of point insertions needed to eliminate boundary (interface) obtuse angles

7 Conclusions

In this paper we present an automatic algorithm that for any input quality constrained Delaunay triangulation with minimum interior angle greater than or equal to 30° , produces a quality nonobtuse boundary and/or interface Delaunay triangulation by eliminating the boundary and/or interface obtuse triangles. The algorithm indeed produces a non-constrained Delaunay triangulation with respect to the interface edges. Even when any quality mesh generation algorithm guaranteeing these bounds can be used to construct the input mesh, the longest-edge Lepp-Delaunay strategy was used in this paper to construct the quality input mesh.

The proposed algorithm guarantees that: (1) if the quality input mesh has only isolated 1-edge boundary (interface) obtuse triangles, the angles of the final triangulation are bounded by 30° and 120° . (2) For general meshes with small boundary (interface) constrained angles, some few triangles not satisfying the bound can appear in the neighborhood of the 2-edge boundary (interface) isosceles triangles.

The elimination of boundary (interface) obtuse triangles introduces a finite number of points for which an upper bound can be previously computed from the input quality mesh. In particular, for meshes with boundary (interface) constrained angles greater than or equal to 32.54° and having non-grouped interface triangles, the number of inserted points is bounded by twice the number of boundary (interface) obtuse triangles.

Finally, two extensions of the results presented in this paper can be envisaged: (1) A more general algorithm able to deal with quality meshes with interior angles greater than or equal to ϵ with $\epsilon < 30^\circ$ can be designed, where depending on the value of ϵ , the elimination of 1-edge boundary (interface) obtuse triangles will require of the Delaunay insertion of more than one boundary (interface) point. (2) An iterative algorithm able to produce quality nonobtuse boundary (interface) triangulations with all interior angles greater than or equal to ϵ with $\epsilon \leq 30^\circ$ can be also designed. To this end, the boundary (interface) point insertion algorithm described in this paper followed by an interior point insertion algorithm will be repeatedly used until an acceptable quality triangulation is produced.

8 Acknowledgment

The first author thanks to Michael Murphy and Norbert Strecker for valuable interaction related with the subject. The programming of the algorithm was done by Mauricio Palma. This work was supported by Fondecyt project No 1960735 and Fondap AN-1 project.

References

- [1] M. R. Pinto. *Comprehensive Semiconductor Device Simulation for Silicon ULSI*. PhD thesis, Stanford University, 1990.

- [2] N. Hitschfeld, P. Conti, and W. Fichtner. Mixed Elements Trees: A Generalization of Modified Octrees for the Generation of Meshes for the Simulation of Complex 3-D Semiconductor Devices. *IEEE Trans. on CAD/ICAS*, 12:1714–1725, November 1993.
- [3] J. F. Bürgler. *Discretization and Grid Adaptation in Semiconductor Device Modeling*. PhD thesis, ETH Zürich, 1990. published by Hartung-Gorre Verlag, Konstanz, Germany.
- [4] N. Hitschfeld. *Grid Generation for Three-dimensional Non-Rectangular Semiconductor Devices*. PhD thesis, ETH Zürich, Series in Microelectronics, Vol. 21, 1993. PhD thesis published by Hartung-Gorre Verlag, Konstanz, Germany.
- [5] M. Bern, S. Mitchell, and J. Ruppert. Linear nonobtuse triangulation of polygons. In *Proc. 10th annu. ACM sympos. computational geometry*, pages 231–241, St.Louis, 1994.
- [6] S. Müller, K. Kells, and W. Fichtner. Automatic rectangle-based Adaptive Mesh Generation without Obtuse Angles. July 1992.
- [7] G. Garretón, L. Villablanca, N. Strecker, and W. Fichtner. A new approach for 2-d mesh generation for complex device structures. In *NUPAD V - Technical Digest*, Honolulu, USA, June 1994.
- [8] Gary L. Miller, Dafna Talmor, Shang-Hua Teng, Noel Walkington, and Han Wang. Control volume meshes using sphere packing: generation, refinement and coarsening. In *Proceedings of the 5th International meshing Roundtable*, pages 47–61, Pittsburgh, Pennsylvania, 1996.
- [9] M. C. Rivara. New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations. *Proc. 5th Int. Meshing Roundtable. Pittsburgh*, pages 77–86, 1996.
- [10] M. C. Rivara. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *Int. J. Num. Methods*, 40:3313–3324, 1997.
- [11] M. C. Rivara and N. Hitschfeld. LEPP-Delaunay algorithm: a robust tool for producing size-optimal quality triangulations. *Proc. of the 8th Int. Meshing Roundtable*, pages 205,220, October, 1999.
- [12] M. C. Rivara and M. Palma. New lepp algorithms for quality polygon and volume triangulation: Implementation issues and practical behavior. In *Trends unstructured mesh generation. Eds: S. A. Cannan, S. Saigal. AMD-Vol. 220, ASME Publication*, pages 1–8, 1997.
- [13] Michael Murphy and Carl W. Gable. Strategies for nonobtuse boundary delaunay triangulations. In *Proceedings of the 7th International Meshing Roundtable*, pages 309–319. Sandia National laboratories., October 1998.