Conforming Delaunay Triangulations in 3D ⋆

David Cohen-Steiner [a], Éric Colin de Verdière [b,∗], Mariette Yvinec [a]

[a] *I.N.R.I.A., Sophia-Antipolis, France.*

[b] *Laboratoire d'informatique de l'École normale supérieure, UMR 8548 (CNRS), Paris, France*

Abstract

We describe an algorithm which, for any piecewise linear complex (PLC) in 3D, builds a Delaunay triangulation conforming to this PLC.

The algorithm has been implemented, and yields in practice a relatively small number of Steiner points due to the fact that it adapts to the local geometry of the PLC. It is, to our knowledge, the first practical algorithm devoted to this problem.

Key words: Conforming Delaunay triangulation, Delaunay triangulation, mesh
1991 MSC: 52B70
1991 MSC: 68R99

⋆ Work partially supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-2000-26473 (ECG – Effective Computational Geometry for Curves and Surfaces). This work was initiated during a visit of the second author at I.N.R.I.A., supported by I.N.R.I.A. A preliminary version of this paper appeared in the *Proceedings of the 18th ACM Symposium on Computational Geometry 2002.*

∗ Corresponding author.

Email addresses: `David.Cohen-Steiner@sophia.inria.fr` (David Cohen-Steiner), `Eric.Colin.de.Verdiere@ens.fr` (Éric Colin de Verdière), `Mariette.Yvinec@sophia.inria.fr` (Mariette Yvinec).

URLs: `http://www-sop.inria.fr/prisme/personnel/David.Cohen-Steiner/` (David Cohen-Steiner), `http://www.di.ens.fr/~colin/` (Éric Colin de Verdière), `http://www-sop.inria.fr/prisme/personnel/yvinec/` (Mariette Yvinec).

This paper appeared in *Computational Geometry: Theory and Applications*, 28 (2004), 217–233.

1 Introduction

In the following, the term *faces* denotes objects in 3D space which are either 0-dimensional faces called vertices, 1-dimensional faces called edges or 2-dimensional faces called 2-faces. The vertices are just points, the edges are straight line segments, and the 2-faces are polygonal regions possibly with holes and isolated edges or vertices included in their interior. A piecewise linear complex, called for short PLC, is a finite set $\mathcal{C}$ of faces such that:

- the boundary of any face of $\mathcal{C}$ is a union of faces of $\mathcal{C}$;
- the intersection of any two faces of $\mathcal{C}$ is either empty or a union of faces of $\mathcal{C}$.

A triangulation $\mathcal{T}$ is said to *conform* to a PLC $\mathcal{C}$ if any face of $\mathcal{C}$ is a union of faces of $\mathcal{T}$. In this paper, we propose an algorithm which, given a PLC $\mathcal{C}$, finds a set of points $\mathcal{P}$ whose Delaunay triangulation conforms to $\mathcal{C}$. The set $\mathcal{P}$ includes the vertices of $\mathcal{C}$ and a certain number of additional points which are usually called Steiner points.

This question is motivated by problems in mesh generation and geometric modeling: in these fields, it is crucial to decompose the space into a set of simplices which conforms to a given PLC, with the additional restriction that the shape of the cells must satisfy certain properties. Delaunay triangulations present several features (see, *e.g.*, [1]) which can be exploited to solve this problem, and many mesh generation algorithms make use of this concept.

The problem of computing a conforming 2D Delaunay triangulation was solved by Saalfeld [6] and Edelsbrunner and Tan [3]. The algorithm by Edelsbrunner and Tan [3] guarantees an $O(n^3)$ bound on the number of generated Steiner vertices, if n is the size of the input. Most of the further works on the subject are based on the Delaunay refinement approach pioneered by Ruppert [5] and Chew [2]. Shewchuk [7] gave an algorithm in 3D which builds a conforming Delaunay triangulation under restrictive conditions on the angles of the PLC. Murphy, Mount, and Gable [4] found a solution which works under no restriction, but produces far too many points in practice. The main interest of their paper is to show the existence of a conforming Delaunay triangulation with a finite set of vertices for any 3D PLC.

Our algorithm uses the Delaunay refinement approach. Initially, the set $\mathcal{P}$ is the set of vertices of the complex $\mathcal{C}$. Points are then added to $\mathcal{P}$ until each edge and each face of the complex $\mathcal{C}$ is a union of simplices which are in the Delaunay triangulation of $\mathcal{P}$.

The main difficulty with such a strategy is to ensure termination. Indeed, it is known that sharp edges and corners may induce cascading additions of Steiner points. To avoid this effect, we first define a protected area around edges and vertices of the PLC with a special refinement process. Outside the protected area, the PLC can be refined using Ruppert's process and the interaction between refinements in both areas can be controlled. Murphy, Mount, and Gable use a similar approach. The main difference with our work lies in the definition of the protected area. In our case, this area adapts to the local geometry of the input PLC.

The algorithm is presented in Section 2 and proved to be correct in Section 3. In Section 4, we present the details of the construction of the initial protected area, skipped in Section 2. Section 5 presents some refinements to improve the running time of the algorithm and to lower the number of vertices in the output conforming triangulation. At last, we end with experimental results in Section 6.

2 The algorithm

After a few definitions, we describe the protected area (Subsections 2.2 and 2.3). We then define the refinement process used for this area (Subsections 2.4 and 2.5). Finally, we describe the main procedure and summarize the whole algorithm.

2.1 Definitions and notations

The *circumball* of a segment ab is the ball admitting the segment ab as diameter. The *circumball* of a triangle abc is the ball admitting the circumscribing circle of abc as great circle.

An edge (resp. a triangle) is said to have the *Gabriel property* if its circumball contains no point of $\mathcal{P}$ in its interior. A point in the interior of the circumball of an edge (resp. a triangle) is said to *encroach upon* this edge (resp. this triangle).

In the following, we note $\mathrm{bd}(B)$ the boundary of a ball B, $\mathrm{int}(B)$ the interior of B and $\mathrm{circum}(ab)$ (resp $\mathrm{circum}(abc)$) the circumball of the segment ab (resp. of the triangle abc).

2.2 Protecting balls

The *1-skeleton* Sk of the complex $\mathcal{C}$ is the union of the 0- and 1-dimensional faces of $\mathcal{C}$. The protected area is defined by means of a set $\mathcal{B}$ of closed balls, called protecting balls, satisfying the following requirements:

(i) the union of the balls in $\mathcal{B}$ covers the 1-skeleton Sk of the complex $\mathcal{C}$;
(ii) the balls are centered on points which are in Sk;
(iii) if two balls intersect, their centers belong to the same edge of the complex $\mathcal{C}$;
(iv) if a face of $\mathcal{C}$ intersects a ball, then it contains the center of this ball;
(v) the intersection of any three balls in $\mathcal{B}$ is empty;
(vi) any two balls are not tangent;
(vii) the center of any ball is inside no other ball.

(i) and (iv) imply that any vertex in $\mathcal{C}$ is the center of a ball in $\mathcal{B}$. We show in Section 4 how to build a set of balls satisfying these requirements. Furthermore, in Section 5, we show that there is in fact no need to cover all the edges.

2.3 Center-points, h-points, p-points, and SOS-points

We describe here a few subsets of points, included in the balls of $\mathcal{B}$, that we need to add first in the set $\mathcal{P}$. See Figure 1.

Let B be a ball in $\mathcal{B}$ with center o. Let $\mathcal{B}_B$ be the set of balls in $\mathcal{B}$ that intersect B. By condition (v), the intersections of B with the elements of $\mathcal{B}_B$ are disjoint.

We first add the center o of B. Such a point will be called a *center-point*. Then, for each element B_i of $\mathcal{B}_B$, consider the radical plane of B and B_i. It intersects the line joining the centers of B and B_i at a point h_i, which is on an edge of $\mathcal{C}$ by condition (iii). The point h_i is added to the set $\mathcal{P}$. Such points will be called *h-points*.

By condition (iv), any face of $\mathcal{C}$ which intersects $B \cap B_i$ contains the centers of B and B_i, and thus can be either the edge including the segment oo_i (o_i is the center of B_i) or a 2-face incident this edge. For each 2-face F of $\mathcal{C}$ intersecting $B \cap B_i$, we add to $\mathcal{P}$ the intersection points of F with the circle $\mathrm{bd}(B) \cap \mathrm{bd}(B_i)$. We called those points *p-points*.

Consider the plane Q of a 2-face of $\mathcal{C}$ intersecting B (and thus containing o). The edges of $\mathcal{C}$ split the disk $Q \cap B$ into one or several sectors. We focus

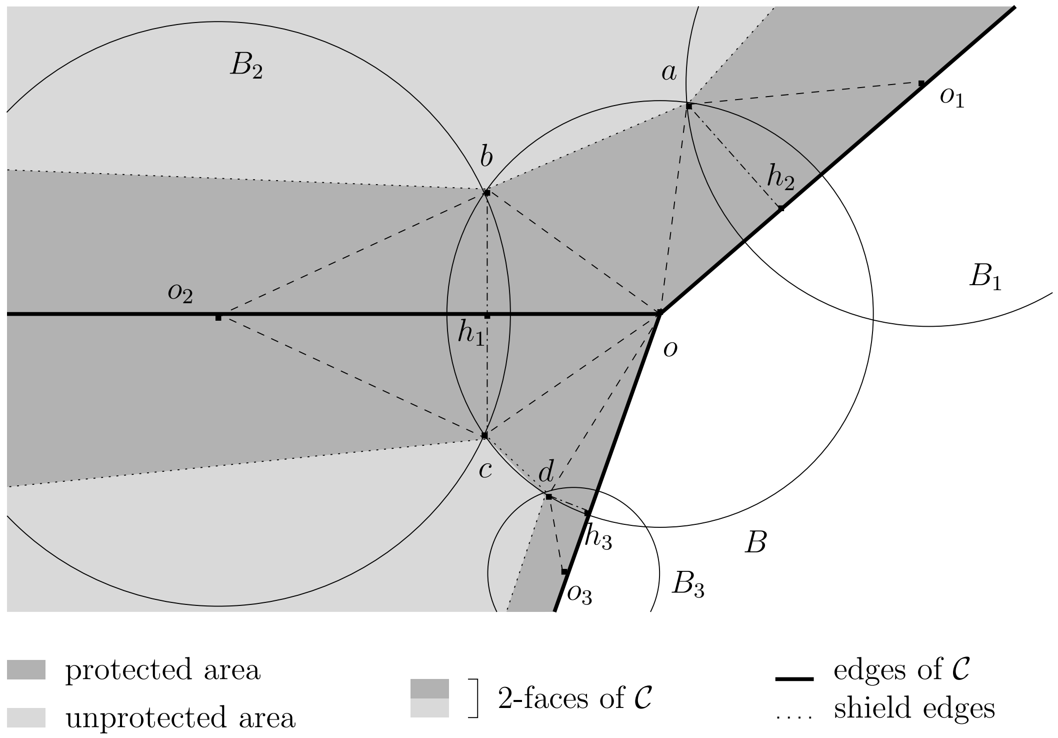

Fig. 1. The situation in the neighborhood of a ball B, incident to three other balls B_1, B_2 and B_3. There are two faces in the complex, limited by three edges, in the plane of the figure. Point h_i is added on the radical plane of B and B_i. p-points a, b, c, and d belong to the boundary of two balls and to a face, they are therefore also inserted in $\mathcal{P}$. Incident to o are four right-angled triangles (*e.g.*, oh_2a) and two isosceles triangles (*e.g.*, oab). The shield edges are ab and cd.

on sectors which are included in $\mathcal{C}$. The p-points further split these sectors in subsectors. We call *right-angled subsectors* the subsectors limited by an edge of $\mathcal{C}$ and a p-point and *isosceles subsectors* the subsectors limited by two p-points.

If some isosceles subsectors form an angle $\geq \pi/2$, we add some points on their bounding circular arcs to subdivide them in new subsectors forming an angle $< \pi/2$. For reasons that will be clear in Subsection 2.4, these points are called SOS-*points*. The new subsectors with angle $< \pi/2$ are still called *isosceles subsectors.*

Center-points and h-points are the only categories of points added in the interior of protecting balls. p-points and SOS-points lie on the boundaries of protecting balls. SOS-points belong to a single protecting ball while p-points belong to the intersection of two balls.

Isosceles subsectors are defined by the center o of a ball B and by two points a and b (either p-points or SOS-points) on $\mathrm{bd}(B)$. Line segments such as ab, joining two points that define an isosceles subsector, are called *shield edges.*

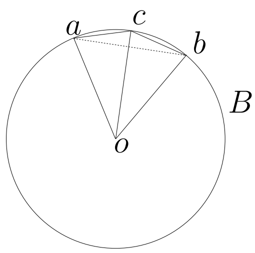

Fig. 2. The SOS strategy: We split the shield edge ab by inserting the point c on the boundary of the ball.

In the following, triangles defined by center-points and shield edges such as oab are referred to as *isosceles triangles.* Triangles spanned by a center-point, a h-point and a p-point on the boundary of some right-angled subsector are referred to as *right-angled triangles.*

Definition 1 *The* protected area *is the union of the isosceles and right-angled triangles. See the dark gray area in Figure 1. In particular, the protected area is included in the union of the protecting balls.*

Definition 2 *The* unprotected area *is the complex $\mathcal{C}$, minus the protected area.*

2.4 The "split-on-a-sphere" strategy

During the process, it will be necessary to split shield edges. Since we do not want to add more points inside the balls in $\mathcal{B}$, we use a special treatment to split such a shield edge, called the "split-on-a-sphere" strategy (SOS for short). See Figure 2.

Let ab be a shield edge to be split, in a ball B. We distinguish two cases: a and b are both SOS-points and belong to a single ball B, or at least one of these two points (for example a) is a p-point and belongs also to another ball B'.

If a and b belong only to B, let c be the midpoint of the shortest geodesic arc ab on $\text{bd}(B)$. To refine edge ab, we add c to $\mathcal{P}$ and replace the shield edge ab by two shield edges ac and cb.

If a is a p-point belonging to $\text{bd}(B) \cap \text{bd}(B')$, the idea is quite similar; however, if we do not take care, the SOS strategy could lead to cascading insertions of points, because refining an edge on B would lead to refinement of an edge on B', and so on. We thus use a strategy "à la Ruppert" [5], using circular shells. We consider the length of the segment ab, divided by two, and round it to the nearest distance d which is of the form $2^k, k \in \mathbb{Z}$ (the unit distance has been chosen arbitrarily at the beginning of the algorithm). Let c be the point of the shortest geodesic arc ab on $\mathrm{bd}(B)$ at distance d from a. We split the shield edge ab using the point c.

In both cases, the added point c belongs to the category of *SOS*-points. Note that, due to the SOS refinement strategy, the protected and unprotected areas, still defined as in Subsection 2.3, will slightly evolve during the algorithm. Each SOS refinement increases the protected area and decreases the unprotected area.

2.5 The protection procedure

This procedure adds some points to set $\mathcal{P}$ to ensure that shield edges and isosceles triangles have the Gabriel property. It uses recursively the SOS strategy and works as follows: While there is an encroached shield edge ab or an encroached isosceles triangle oab, refine the edge ab using the SOS strategy.

2.6 The whole algorithm

Let us recall that the algorithm works by adding points to set $\mathcal{P}$. We note $Dt_3(\mathcal{P})$ the 3D Delaunay triangulation of points in $\mathcal{P}$. For each plane Q of a 2-face in $\mathcal{C}$, we note $Dt_2(\mathcal{P} \cap Q)$ the 2D Delaunay triangulation of points in $\mathcal{P} \cap Q$. These triangulations are updated upon each insertion of a point in $\mathcal{P}$.

The algorithm performs the initialization step and the main procedure described below.

The Initialization Step:

- Construct and initialize the protected area (as described in 2.2 and 2.3);
- execute the protection procedure.

We will see later that the Delaunay triangulation of $\mathcal{P}$ conforms to the part of $\mathcal{C}$ which is inside the protected area. Because the algorithm maintains the Gabriel property of shield edges, in each plane Q of a 2-face F of $\mathcal{C}$, the 2D triangulation $Dt_2(\mathcal{P} \cap Q)$ conforms to the shield edges in this plane and thus to the unprotected part F_u of F. The main procedure ensures that the triangles of $Dt_2(\mathcal{P} \cap Q)$ included in F_u appear in the 3D triangulation $Dt_3(\mathcal{P})$.

The Main Procedure:
The Main Procedure consists in executing the following loop: While there is

a triangle T in the 2D Delaunay triangulation $Dt_2(\mathcal{P} \cap Q)$ of the plane Q of a 2-face F of $\mathcal{C}$ such that:

(a) T is included in the unprotected part F_u of F,
(b) T does not appear in $Dt_3(\mathcal{P})$,

refine T trying to insert its circumcenter c, that is:

- if c encroaches upon no shield edge, insert it;
- otherwise, split all the shield edges encroached upon by c using the SOS strategy, and then execute the protection procedure.

3 Proof of the algorithm

Two steps are involved for the proof of this algorithm. First, we prove invariants of the algorithm concerning the positions of the points added and the Gabriel property of some triangles and edges. After that, we are able to prove termination.

3.1 Properties maintained in the algorithm

Lemma 3 *At the beginning (and the end) of each execution of the main loop, the shield edges have the Gabriel property.*

Proof. Indeed, this is true before the first execution of the main loop, because the protection procedure, which has just been executed, ensures this property; for the same reason, this also holds after an execution of the loop leading to the split of shield edges. At last, a circumcenter is inserted in $\mathcal{P}$ only if it does not violate this property. $\square$

In the following, we define an *added* circumcenter to be a circumcenter inserted in the set $\mathcal{P}$, and a *rejected* circumcenter to be a circumcenter considered in the algorithm but not inserted because it encroaches upon some shield edge.

Lemma 4 *Any circumcenter (added or rejected) considered by the algorithm lies in the unprotected area, outside the protecting spheres. In particular, no point is added inside the protecting spheres after the initialization step, and $\mathcal{P}$ is included in $\mathcal{C}$.*

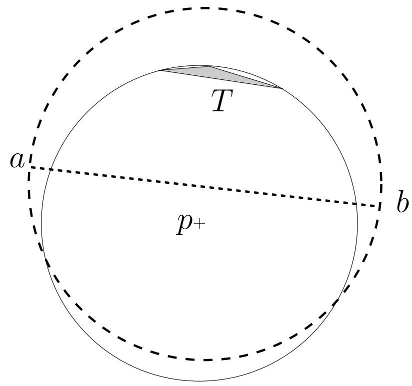

Fig. 3. The circumcenter p of a triangle T lies in the unprotected area.

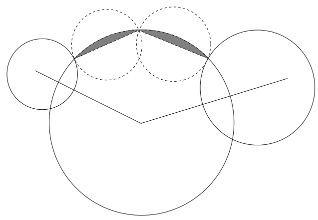

Fig. 4. The intersection of the unprotected area with the union of protecting balls is included in the circumballs of shield edges.

Proof. Let T be a triangle whose circumcenter is considered at some step of the algorithm. T lies in the unprotected area, and belongs to the 2D Delaunay triangulation $Dt_2(\mathcal{P} \cap Q)$ of the plane Q of some 2-face in $\mathcal{C}$. Let p be the circumcenter of T. Assume for contradiction that p lies outside the unprotected area. Let m be a point in T. Since shield edges enclose the connected component of the unprotected area which contains T, the segment pm must intersect a shield edge ab. The vertices a and b cannot be inside circum(T) because T belongs to $Dt_2(\mathcal{P} \cap Q)$. Hence (Figure 3), triangle T belongs to the circumball of ab, which is impossible by Lemma 3.

Moreover, since the circumballs of shield edges cover the intersection of the unprotected area with the protecting balls (see Figure 4), any added circumcenter is actually outside the protecting spheres. □

Proposition 5 *At the beginning (and the end) of each execution of the main loop, the isosceles triangles have the Gabriel property.*

Proof. The proposition is obvious after the initialization step because the protection procedure is called and enforces the Gabriel property of isosceles triangles. For the same reason, it is also the case when a circumcenter has just been rejected because it encroaches upon some shield edge.

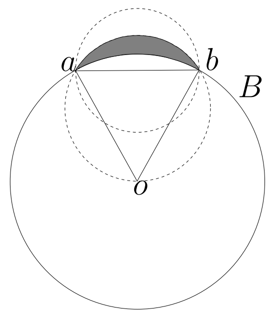

Fig. 5. The balls B, circum(oab), and circum(ab).

It remains to see that this proposition is still true when a circumcenter has just been inserted: such a circumcenter lies outside the protecting spheres (by Lemma 4) and outside the circumball of any shield edge (otherwise it is not inserted in $\mathcal{P}$). Let ab be such a shield edge, belonging to ball B. We note that the boundaries of B, circum(ab), and circum(oab) belong to a pencil of spheres. Because the angle $\widehat{aob}$ is smaller than $\pi/2$, we have circum(oab) $\subset$ circum(ab) $\cup B$ (Figure 5). The result follows. $\square$

Lemma 6 *Let B be a ball with center o, and p be a point on the boundary of B. If, at some stage of the algorithm, the segment op is encroached upon, the encroaching point is a h-point h_i on the radical plane of B and B_i, and p belongs to* $\mathrm{bd}(B) \cap \mathrm{int}(B_i)$.

Proof. The circumball of op is inside B. Therefore, op can only be encroached upon by a vertex in this ball, and not by the center of B, hence only by a h-vertex in B. Suppose that op is encroached upon by a vertex h_i, belonging to B and B_i. The encroachment condition can be rewritten $\widehat{oh_ip} > \pi/2$. Because points q in $\mathrm{bd}(B)$ that satisfy $\widehat{oh_iq} > \pi/2$ lie in $\mathrm{bd}(B) \cap \mathrm{int}(B_i)$, p belongs to $\mathrm{int}(B_i)$. $\square$

Proposition 7 *At each stage of the algorithm, the right-angled triangles have the Gabriel property.*

Proof. Suppose that a right-angled triangle oh_jp does not have the Gabriel property at some stage of the algorithm: h_j is on the radical plane between B and B_j, and p is on the boundary of B and B_j. Because the circumball of oh_jp is the circumball of op, by Lemma 6, the encroching point is a h-point, and p has to belong to the interior of a third ball B_i, which is impossible by condition (v). $\square$

Center points and h-points cut the edges of $\mathcal{C}$ in subedges. Note that Proposition 7 implies that these subedges are edges of $Dt_3(\mathcal{P})$.

3.2 Termination proof

Proposition 8 *The protection procedure always terminates.*

The proof is a straightforward consequence of the following lemma.

Lemma 9 *For each call to the protection procedure, there exists $\theta > 0$ such that no isosceles triangle with angle at the center of the ball less than θ will be split.*

Proof. Let oab be an isosceles triangle with shield edge ab in a protecting ball B. We consider in turn three kinds of possible encroaching points: points on the boundary of B (case 1), points in the interior of B (case 2), and points outside B (case 3). In each case k, we prove the existence of a value θ_k, such that neither oab nor ab can be encroached upon by a point of type k if $\widehat{aob} < \theta_k$.

Recall that the three balls B, circum(ab) and circum(oab) belong to a pencil of spheres. Because the angle $\widehat{aob}$ is smaller than $\pi/2$, we have circum$(oab) \subset B \cup$ circum(ab) and circum$(ab) \cap B \subset$ circum(oab) (see Figure 5). Therefore, it is enough to check that points on the boundary of B or outside B (cases 1 and 3) do not encroach upon ab and that points in B (case 2) do not encroach upon oab.

(1) For a plane Q of a 2-face of $\mathcal{C}$ intersecting B, we consider the circle bounding $B \cap Q$ and we denote by $S(Q, B)$ the union of arcs on this circle spanned by the isosceles triangles in Q. Notice that all the SOS-points inserted on B are located on such a set $S(Q, B)$.

If Q is the plane containing oab, no point of $S(Q, B)$ encroaches upon ab. If Q' is another plane, the distance between $S(Q, B)$ and $S(Q', B)$ is strictly positive, so there is a value $\theta_1(B, Q, Q')$ such that ab is not encroached upon by a point on $S(Q', B)$ if $\widehat{aob} < \theta_1$. Setting $\theta_1 = \min\{\theta_1(B, Q, Q')\}$ achieves the proof of case 1.

(2) The only points in a ball B which can encroach upon an isosceles triangle oab in B are the h-points in B. Suppose that a point h_i (on the radical plane of B and B_i) encroaches upon oab.

If h_i is in the plane Q of oab, we prove that encroachment is not possible. Indeed, if h_i encroaches upon oab, h_i encroaches either upon oa or upon ob. Thus a or b would belong to bd$(B) \cap$ int(B_i), by Lemma 6, which is impossible because a and b are either p-points or SOS-points.

Let us now deal with the case where h_i does not belong to the plane Q. Let $c \in S(Q, B)$; c does not belong to B_i, for otherwise h_i would belong to Q. Let us prove that h_i is not in the closed ball circum(oc). If h_i is in the interior of circum(oc), this means that oc is encroached upon by h_i, hence, by Lemma 6, c belongs to int(B_i), which is not the case. Similarly, if h_i is on the boundary of circum(oc), c belongs to B_i.

Hence, the distance between h_i and the ball circum(oc) is strictly positive. Let $\delta(B, Q, h_i)$ be the minimum (strictly positive) of this distance for $c \in S(Q, B)$. Let $\delta'(B, \theta)$ be the Hausdorff distance between circum(oc) and circum$(oa'b')$ where $oa'b'$ is an isosceles triangle with a' and b' on bd(B), axis oc and $\widehat{a'ob'} = \theta$. As $\delta'(B, \theta)$ goes to 0 when θ goes to 0, there exists $\theta_2(B, Q, h_i)$ such that $\delta'(B, \theta) < \delta(B, Q, h_i)$ for any $\theta < \theta_2(B, Q, h_i)$. It follows that oab cannot be encroached upon by h_i if $\widehat{aob} < \theta_2(B, Q, h_i)$. Setting $\theta_2 = \min\{\theta_2(B, Q, h_i)\}$ achieves the proof of case 2.

(3) Consider now the case where edge ab is encroached upon by a point p outside the ball B. At each call of the protection procedure, the set of points outside the protecting spheres is fixed. Also, the distance between two sets $S(Q_1, B_1)$ and $S(Q_2, B_2)$ which do not share a p-point is bounded from below. Thus, there is a value θ'_3 such that, if $\widehat{aob} < \theta'_3$, edge ab cannot be encroached upon by p except if p belongs to $S(Q, B')$ where Q is the plane of oab and B' intersects B. Therefore, the only case remaining to be considered is the case where a is a p-point in $Q \cap \mathrm{bd}(B) \cap \mathrm{bd}(B')$ and ab is encroached upon by a point p of $S(Q, B')$. However, in this case, we split edges incident to a using circular shells. Hence, after a few splits, the edges incident to a will have the same lengths and will be unable to encroach upon each other. Therefore, we get a value $\theta_3 \leq \theta'_3$ satisfying the desired requirement.

□

Theorem 10 *The algorithm terminates, and, once it is the case, the Delaunay triangulation of $\mathcal{P}$ conforms to the complex $\mathcal{C}$.*

Proof. It is sufficient to prove that the main procedure terminates: indeed, once it is the case, Propositions 5 and 7 show that the Delaunay triangulation of $\mathcal{P}$ conforms to the protected area of $\mathcal{C}$, and the fact that the algorithm ends precisely means that the Delaunay triangulation of $\mathcal{P}$ also conforms to the unprotected area of $\mathcal{C}$. We prove the termination of the main procedure by proving first that the number of added circumcenters is finite and second that the number of shield edges encroached upon by rejected circumcenters is finite. Because the protection procedure is already known to terminate, these two facts imply the termination of the main procedure.

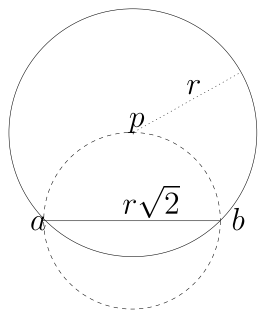

Fig. 6. The shortest shield edge ab which may be encroached upon by a rejected circumcenter p.

By construction of the protecting spheres, the unprotected area is a disjoint union of plane regions. Let F_u be such a region. As previously noticed, owing to the SOS strategy, these unprotected regions slightly evolve during the algorithm; however, they are always shrinking. Consequently, the distance between F_u and the other regions as well as the distance between F_u and the set of center-points and h-points added in the interior of the protecting balls can be bounded from below by a constant δ_F. Let T be a triangle in F_u whose circumcenter has to be inserted in $\mathcal{P}$ and let C_T be the circumcircle of T. As T does not belong to $Dt_3(\mathcal{P})$, its circumball circum(T) contains a point in $\mathcal{P}$ which is not in the plane of F_u. Such a point can be inside a protecting ball (a center-point or a h-point), on the boundary of a protecting ball (and thus on the boundary of another region), or an added circumcenter (in another region by Lemma 4). Therefore circum(T) either contains a point added in the interior of a protecting sphere or intersects another unprotected region, and the radius of C_T is thus larger than δ_F. Because T belongs to the 2D Delaunay triangulation in the plane of F_u, C_T encloses no point of $\mathcal{P}$. The area of F_u being finite, this shows that the number of added circumcenters is bounded.

Let us now show that the total number of edges encroached upon by rejected circumcenters is finite. For this purpose, consider a shield edge encroached upon by the center p of a circumcircle C in a region F_u. C being empty and of radius larger than δ_F, it is easy to show that the shield edge has length at least $\delta_F\sqrt{2}$ (see Figure 6). Thus the number of those edges is finite. □

4 Construction of the protecting balls

We have to build the set $\mathcal{B}$ of protecting balls satisfying the conditions described in Subsection 2.2. The efficiency of the algorithm really depends on this construction: the less balls there are, the less points will be produced in $\mathcal{P}$.

Definition 11 *Let $\mathcal{C}$ be a PLC. The* local feature size *of a point p with respect to $\mathcal{C}$ is the distance between p and the union of faces of $\mathcal{C}$ that do not contain p.*

Let $\mathit{lfs}(p)$ denote the local feature size of point p with respect to the PLC which is given as input of the algorithm. We address the following construction of the enclosing balls. Let α be a real, $0 < \alpha < \frac{1}{2}$ (typically $\alpha = 0.4$).

First, for each vertex v of the PLC, construct a ball of radius $\alpha \cdot \mathit{lfs}(v)$.

Then, on each edge e, do the following. While e is not completely covered by balls, consider a maximal open line segment $a_1 a_2$ in e and outside the union of the balls in the current set $\mathcal{B}$. Point a_i ($i = 1, 2$) is an intersection of ball B_i (with center o_i and radius r_i) with edge e. We will insert a ball between B_1 and B_2. Let o be the midpoint of $a_1 a_2$. Insert a new ball B in $\mathcal{B}$, of center o and radius r, with:

$$r = \min\left\{\alpha \cdot \mathit{lfs}(o), oa_1 + \frac{r_1}{2}, oa_2 + \frac{r_2}{2}\right\}.$$

To ensure condition (vi), if $r = oa_1$, we replace r by $(1-\varepsilon)r$ where ε is a small positive constant.

Lemma 12 *This construction terminates.*

Proof. Consider an edge e, whose vertices have just been protected by two spheres. Let A be the union of the (open) line segments which are in e minus the union of the current set of balls. Call A_0 the set A just after the protection of the endpoints of e. The distance $d = \min\{\mathit{lfs}(p) | \ p \in \overline{A_0}\}$ is strictly positive (the lfs function is continuous on $\overline{A_0}$, and lfs does not vanish on $\overline{A_0}$). The insertion of a new ball:

- either increases by one the number of connected components of A and decreases the measure of A by at least $2(1-\varepsilon) \cdot \alpha \cdot d$ (hence this case can happen only a finite number of times),
- or decreases by one the number of connected components of A (without increasing the measure of A).

The result follows. $\square$

Conditions (i), (ii), (iv), (vi) and (vii) are obviously satisfied. (iii) follows from the fact that if two points o and o' do not belong to the same edge, oo' is larger than or equal to $\mathit{lfs}(o)$ and $\mathit{lfs}(o')$. If two balls B and B', centered at o and

o' with radii r and r', are in $\mathcal{B}$, then $r < \frac{1}{2} lfs(o)$ and similarly for r'. Thus $r + r' < oo'$, hence the balls cannot intersect.

(v) is also true. Indeed, if three balls intersect, their centers must be vertices of a triangle in $\mathcal{C}$. But it follows from our construction that two balls centered on vertices of the PLC cannot intersect because $\alpha < \frac{1}{2}$.

Hence we have:

Proposition 13 *This construction of $\mathcal{B}$ is correct.*

5 Improvements

5.1 Speeding up the protection procedure

The following proposition shows that when the protection procedure is called from the main procedure, there is no need to check whether isosceles triangles have the Gabriel property.

Proposition 14 *After the initialization process, enforcing Gabriel property for shield edges in the protection procedure is enough to ensure Gabriel property for isosceles triangles.*

Proof. Upon termination of the initialization step, all isosceles triangles have the Gabriel property. Suppose that, at some stage of the algorithm, a point encroaches upon some isosceles triangle oab without encroaching ab. Let B be the ball containing oab. Since circum(oab) is included in the union of B and circum(ab) (Figure 5), the encroaching point must be inside B.

Hence it is sufficient to show that no isosceles triangle is encroached upon by a vertex inside its protecting ball during the algorithm. By contradiction, let $T = oab$ be the first isosceles triangle encroached upon by a vertex in B. Since no point is inserted inside the balls during the main procedure, T must be a triangle which results from the splitting of some triangle $T' = oac$. The encroaching point can thus only be a h-point h_i lying inside B. Arguing that circum(oab), circum(oac), and circum(oa) belong to a sphere pencil and comparing their radii, we deduce (Figure 7) that circum(oab) $\subset$ circum(oac) $\cup$ circum(oa). However, h_i does not belong to circum(oac) because $T' = oac$ was not encroached upon by h_i, nor to circum(oa) (by Lemma 6). Therefore h_i does not belong to circum(oab), which yields the contradiction. $\square$

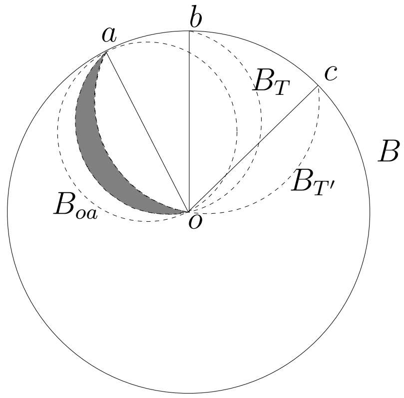

Fig. 7. circum(oab) $\subset$ circum(oac) $\cup$ circum(oa).

5.2 Restricting the area where balls are required

In 2.2, the set $\mathcal{B}$ is constructed so that the balls cover the whole 1-skeleton Sk of $\mathcal{C}$. We explain here that this is not always necessary. Indeed, the balls are introduced to avoid troubles with small angles; they are thus not required at places where faces intersect with an angle large enough. This remark enables to put less balls in $\mathcal{B}$, hence to reduce the size of the output $\mathcal{P}$. We first describe the modification in the construction of the balls, and then prove that, despite this slight modification, the algorithm is still correct.

Let $e = o_1o_2$ be an edge of the PLC so that all angles between faces incident to e are $\geq \pi/2$. We modify the algorithm in the following way. Still construct balls B_1 and B_2 centered at the vertices o_1 and o_2. In $\mathcal{P}$, insert o_1, o_2, and the two intersections p_1 and p_2 of e with the boundaries of B_1 and B_2.

Consider p_1p_2 as a shield edge in the main procedure. In other words, whenever this edge would be encroached upon by the insertion of a point in $\mathcal{P}$, split this edge in the middle, to keep it protected at each stage of the algorithm. The original edge of $\mathcal{C}$ is thus not in the protected area, but the process is exactly like in the standard algorithm.

There are only minor modifications for the proof of the algorithm. The unprotected area is still bounded with shield edges. The proof of termination of the protection procedure is analogous: Lemma 9 can be adapted without difficulty to show that there also exists a length $\delta > 0$ such that the protection procedure never splits a shield edge which is a part of an edge and with length less than δ. The only difficulty is to show the following proposition.

Proposition 15 *The modified version of the main procedure always terminates.*

Proof. Let F_u be a region, in a plane Q, incident to edge e. The distance

Table 1
Experimental data.

	geological data	triceratops	umbrella
nb input vertices	7566	2832	16
nb non Delaunay faces	1045	2194	5
nb output vertices	25793	27947	122
running time (s)	83	570	0.7

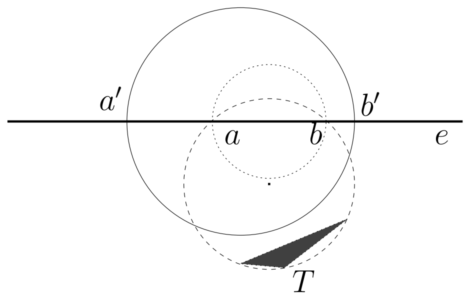

Fig. 8. In the half-space H^+ (above the edge e in the figure), the part of circum(T) is included in the part of circum(ab) which is, in turn, contained in the part of circum($a'b'$).

between F_u and the regions non-incident to e as well as the distance between F_u and the set of center-points and h-points outside Q can be bounded from below by a constant $\delta_F > 0$. Let p be the circumcenter of a triangle T in F_u, added to $\mathcal{P}$. We will show that the circumball of T cannot contain a vertex of another face incident to e, which implies that the radius of this circumball is larger than δ_F, like in the proof of Theorem 10.

Suppose for contradiction that T is encroached upon by a point p' of $\mathcal{P}$ on a face incident to e. Necessarily, because the angles of the faces of $\mathcal{C}$ are obtuse at e, the circumball of T must intersect e. Let a and b be the intersection points of the boundary of circum(T) with e. Let $a'b'$ be the unique shield edge included in e which is intersected by circum(T). (The uniqueness follows from the fact that points in $\mathcal{P}$, like a' and b', cannot lie in circum(T).) Let H be the plane orthogonal to F_u and containing e, and H^+ be the half-space bounded by H and not containing T. Clearly, circum(T) $\cap H^+ \subseteq$ circum(ab) $\cap H^+ \subseteq$ circum($a'b'$) $\cap H^+$ (see Figure 8). The point p' is in circum(T) $\cap H^+$, hence in circum($a'b'$), which means that p' encroaches upon the shield edge $a'b'$ and yields the contradiction.

The remaining part of the proof of termination of the main procedure is exactly the same as in the proof of Theorem 10. $\square$

6 Experimental results

The algorithm has been implemented and tested using the Computational Geometry Algorithms Library CGAL [1]. Results for several models are displayed in Table 1 and Figures 9, 10, 11, and 12.

Table 1 gives for each model, the number of vertices of the input PLC (*nb input vertices*), the number of 2-faces to which the Delaunay triangulation of input vertices does not conform (*non Delaunay faces*), and the number of vertices of the conforming output triangulation (*nb output vertices*). In those examples and in most cases, the number of vertices in the output conforming triangulation and the number of input vertices are in a ratio comprised between 3 to 1 and 10 to 1.

The running times, measured on a PC with 500Mhz processor, do not include the computations of local feature size values, because the current implementation uses a very slow brute force algorithm for it. We are currently designing a data structure devoted to speed up these computations.

7 Conclusion

We have presented an algorithm for computing a conforming Delaunay triangulation of any three-dimensional piecewise linear complex. The most important innovation, compared to the paper by Murphy et al. [4], is to enclose critical places by balls whose radii fit the local complexity of the complex, with the use of the local feature size. Our experimental results show that it is valuable in practice. The algorithm could be easily modified to guarantee in the resulting mesh the Gabriel property for any triangle included in a constraint. The next step currently under work is to investigate how conforming meshes with guarantees on the shape and size of the elements can be obtained. Several questions remain open: we did not try to find the time complexity of our algorithm. It would also be interesting, as in [3] in the plane, to find a bound on the output depending on the size of the initial complex and/or (like in [5]) the *lfs* function.

[1] `http://www.cgal.org/`

Acknowledgements

We thank Jean-Daniel Boissonnat for fruitful discussions and Olivier Billet for his contribution to the early stages of the implementation.

References

[1] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry.* Cambridge University Press, UK, 1998.

[2] L. P. Chew. Guaranteed-quality triangular meshes. Technical Report TR-89-983, Department of Computer Science, Cornell University, Ithaca, NY, April 1989.

[3] H. Edelsbrunner and T. S. Tan. An upper bound for conforming Delaunay triangulations. *Discrete and Computational Geometry*, 10(2):197–213, 1993.

[4] M. Murphy, D. M. Mount, and C. W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. *International Journal of Computational Geometry and Applications*, 11(6):669–682, 2001.

[5] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18:548–585, 1995.

[6] A. Saalfeld. Delaunay edge refinements. In *Proceedings of the 3rd Canadian Conference on Computational Geometry*, pages 33–36, 1991.

[7] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. In *Proceedings of the 14th ACM Symposium on Computational Geometry*, pages 86–95, June 1998.

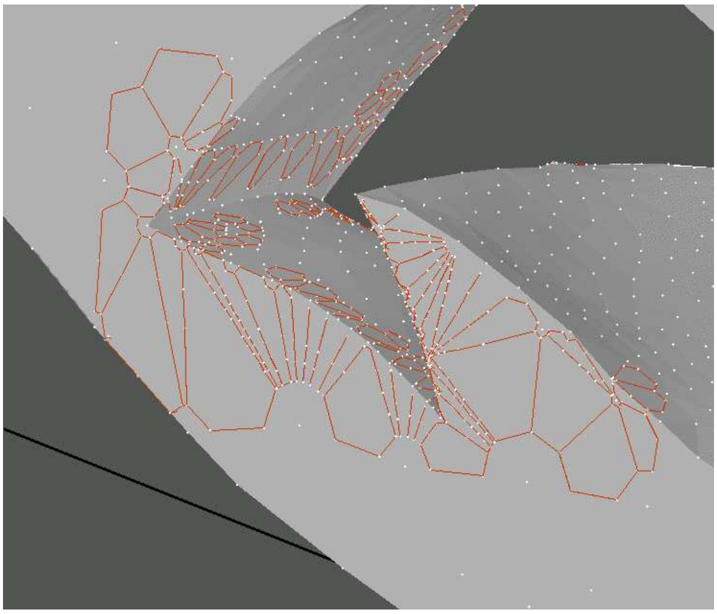

Fig. 9. Detail of a geological formation (Courtesy of T-surf and Mr. Reinsdorff). Solid line segments stand for shield edges.

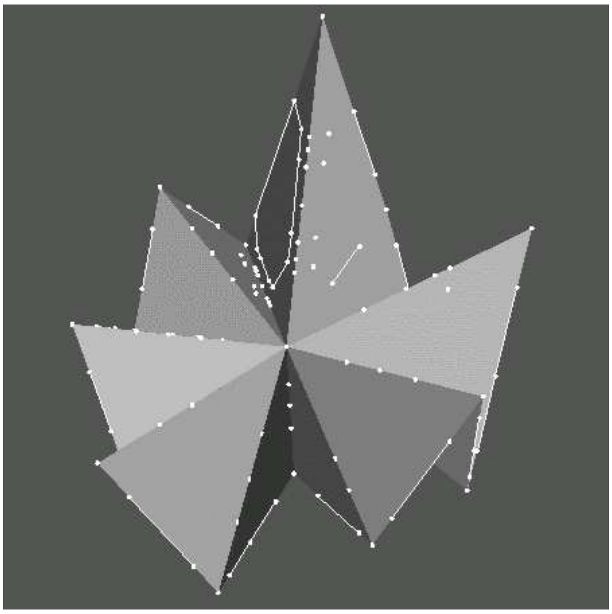

Fig. 10. Umbrella. Solid line segments stand for shield edges.

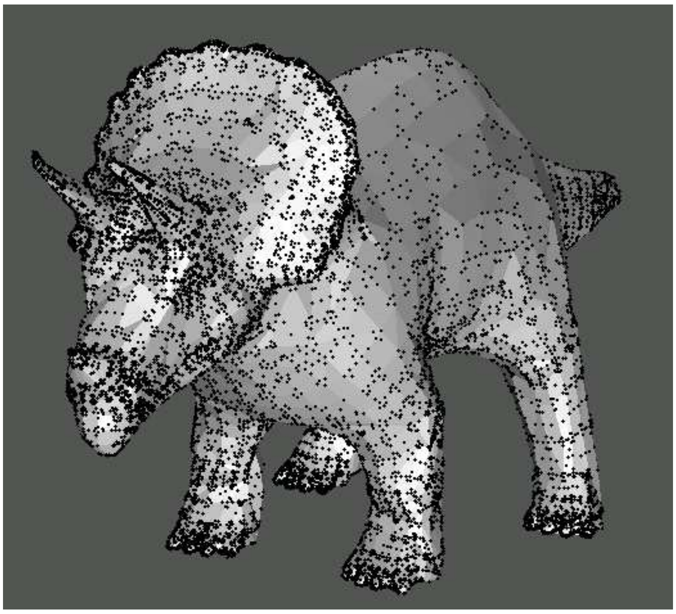

Fig. 11. Triceratops.

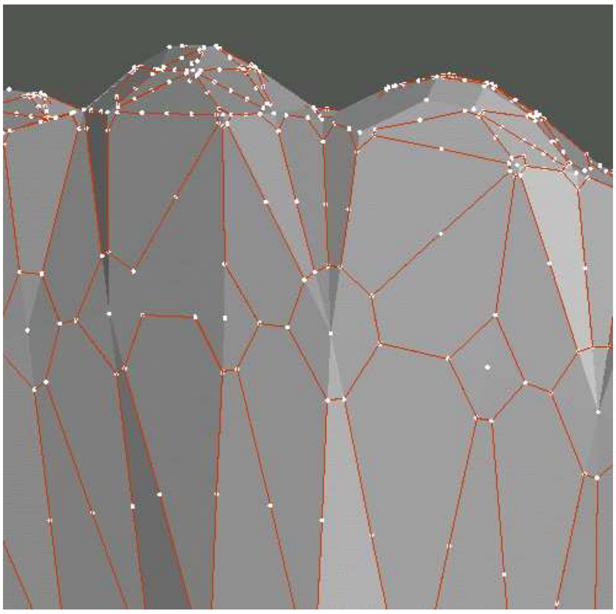

Fig. 12. Detail of the triceratops. Solid line segments stand for shield edges.