On Good Triangulations in Three Dimensions*

Tamal K. Dey *Chanderjit L. Bajaj* *Kokichi Sugihara*

Department of Computer Science
Purdue University
West Lafayette, IN 47907

Abstract

In this paper, we give an algorithm that triangulates the convex hull of a three dimensional point set with guaranteed quality tetrahedra. Good triangulations of convex polyhedra are a special case of this problem. We also give a bound on the number of additional points used to achieve these guarantees and report on the techniques we use to produce a robust implementation of this algorithm under finite precision arithmetic.

1 Introduction

Triangulation of a point set or a polyhedron is an important problem with applications for finite element simulations in CAD/CAM. Though a number of algorithms exist for triangulating a point set or a polyhedron in two and three dimensions [1, 6, 11, 13], few of them address the problem of guaranteeing the shape of the triangular elements. To reduce ill-conditioning as well as discretization error, finite element methods require triangular meshes of bounded aspect ratio [2, 12]. By aspect ratio of triangles or tetrahedra, one may consider the ratio of the radii of the circumscribing circle to that of inscribing circle (spheres in case of tetrahedra).

In 2D, there are basically two approaches known so far to produce guaranteed quality triangulations. The first approach, based on *Constrained Delaunay Triangulations*, was first suggested by Chew [7]. He guarantees that all triangles produced in the final triangulation have angles between 30° and 120°. In [8], we improved this algorithm with minor modifications to guarantee the boundary triangles to have better angle bounds. There is another approach based on *Grid Overlaying* which was first used by Baker, Grosse, and Raferty in [3] to produce a non-obtuse triangulation of a polygon. In [8], we proposed a simpler method based on this grid approach to triangulate a polygon with good angles. Recently, in [5], Bern, Eppstein, and Gilbert give algorithms for producing good triangulations which uses a special type of a grid that simulates the planar subdivision with the quadtree. Though several good heuristics have been published, to date there is no known algorithm that triangulates the convex hull of a three dimensional point set with guaranteed quality tetrahedra. This paper presents some results on good triangulations of the convex hull of a point set in three dimensions. The problem allows one to introduce new points to generate good tetrahedra with the restriction that all points are added only inside or on the boundary of the convex hull. Good triangulations of convex polyhedra are a special case of this problem. Our main results are as follows: (i) a 3D triangulation algorithm based on Delaunay triangulations as used by Chew [7] in 2D, to produce triangulations that do not have four out of five possible types of bad tetrahedra, and (ii) a bound on the number of additional points used to achieve this guarantee. We also report on the techniques we use to produce a robust implementation of this 3D triangulation algorithm in the presence of numerical errors under finite precision arithmetic.

2 Preliminaries

2.1 Characterizing Bad Tetrahedra

In three dimensions, a tetrahedron that is not of bounded aspect ratio can be degenerate or bad in three possible ways as described in [4]. The following two parameters ω, κ characterize bad tetrahedra as follows. Let $\omega = \frac{R}{L}$ and $\kappa = \frac{L}{I}$, where R is the radius of the circumscribing sphere of a tetrahedron, L

*Supported in part by ARO Contract DAAG29-85-C0018 under Cornell MSI, NSF grant DMS 88-16286 and ONR contract N00014-88-K-0402.

and l are the lengths of its longest and shortest edges respectively.

Category(i): $\omega = O(1), \kappa \gg 1$.
Category(ii): $\omega \gg 1$.
Category(iii): $\omega = O(1)$, $\kappa = O(1)$.

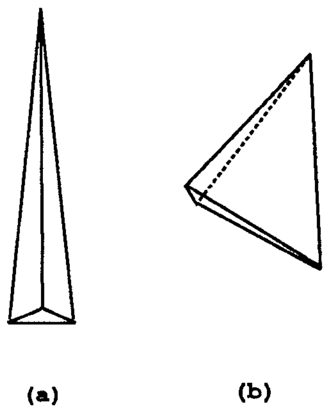


Figure 1: Category(i) tetrahedra

Definition: A sliver is a tetrahedron that is formed by four almost coplanar points and whose solid angles are very close to zero.

Category(i) corresponds to tetrahedra that have a very short edge relative to other edges and have circumscribing spheres that do not have an arbitrarily large radius compared to the length of the longest edge. Specifically, category(i) consists of type(i) and type(ii) tetrahedra. Type(i) tetrahedra are needle-like tetrahedra in which one of the solid angles is highly acute and the face opposite to it has a negligible area (Figure 1(a)). Type(ii) tetrahedra are slivers with a very short edge (Figure 1(b)).

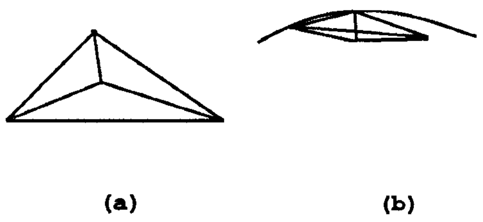


Figure 2: Category(ii) tetrahedra

Category(ii) corresponds to tetrahedra that have a circumscribing sphere with arbitrarily large radius compared to the longest edge. Specifically, category(ii) consists of type(iii) and type(iv) tetrahedra. Type(iii) tetrahedra are flat tetrahedra which have one of the solid angles highly obtuse (Figure 2(a)). Type(iv) tetrahedra are slivers which lie very close to the surface of their large circumscribing spheres (Figure 2(b)). Category(iii) consists of type(v) tetrahedra. Type(v) tetrahedra are slivers whose edges have lengths within a constant factor of each other and which do not have a close incidence with the surface of the circumscribing sphere (Figure 3). We present an algorithm that triangulates the convex hull of a three dimensional point set with the guarantee that type(i) through type(iv) tetrahedra are not generated.

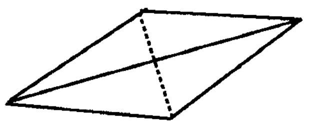


Figure 3: Category(iii) tetrahedra

2.2 2D Algorithm

The core of the algorithm presented in this paper consists of the Delaunay triangulation which is the straight line dual of the Voronoi diagram. In two dimensions, the circumscribing circle of a triangle in the Delaunay triangulation of a point set does not contain any other points inside it. Similarly, in three dimensions, the circumscribing sphere of a tetrahedron in the Delaunay triangulation does not contain any other points inside it. This property of the Delaunay triangulation was utilized by Chew in two dimensions to produce good triangulations. He introduced the centers of those circumscribing circles which maintain a certain minimum distance from the three vertices of the corresponding triangle. Of course, the edges of the boundary have to satisfy certain length criteria. In his algorithm, Chew used edge lengths in between d and $\sqrt{3}d$ where any pair of input points is at least d units away from each other. In the modified algorithm of [8], we require edge lengths in between d and $1.5d$. This gives two distinct advantages.

1. It is easier to divide edges between d and $1.5d$ in practice.
2. The triangles that have circumcenters outside the boundary have better bounds on their angles.

We present below this modified algorithm for good triangulation in two dimensions.

Algorithm 2D-TRI:
Input: Finite number of points in the plane within a polygonal boundary. The vertices of the polygonal boundary are included in the input point set.

Input Conditions: There exists a quantity d, such that no two given points are closer than d and no boundary edge is greater than $1.5d$ and less than d.

```
begin
    Construct the Delaunay triangulation
    of the given point set.
Repeat
    Add the circumcenter $v_l$ of a
    triangle $g = \Delta p_i p_j p_k$ satisfying
    the following property:
    $v_l$ is at a distance of at least d from all
    three points $p_i, p_j, p_k$.
    Update the current triangulation by constructing
    the Delaunay triangulation
    of the augmented point set.
Until there is no such triangle.
end
```

For a simple polygonal boundary with a certain lower bound (39°) on the minimum internal angles at the vertices, it is always possible to choose a d to satisfy the input conditions of the algorithm *2D-TRI*. Algorithm *2D-TRI* produces a planar triangulation T that has the following properties.

Property 1: All edges in T have lengths in between d and $2d$ and in particular all boundary edges have lengths in between d and $1.5d$.

Property 2: The circumscribing circle of all triangles in T has radius less than d.

2.3 Geometric Lemmas

We use the following geometric lemmas in the next section.

Lemma 2.1: Let T be a Delaunay triangulation of a point set in two dimensions. Let R be the maximum radius of all circumscribing circles of Delaunay triangles in T. The radius of any empty circle whose center lies inside T is less than or equal to R.

Proof: See Theorem 6.15[14]. ♣

Definition: Let c be a circle drawn on the surface of a sphere s. Let $\overline{p_1 p_2}$ be the axis which is perpendicular to the supporting plane of c and which passes through the center of c. This axis intersects s at p_1 and p_2. The points p_1, p_2 are called the *poles* corresponding to the circle c.

Lemma 2.2: Let c be a circle with radius less than r drawn on the surface of a sphere s. Let the distance between c and its nearest pole be greater than d. The radius R of s must satisfy the condition $R < \frac{r^2+d^2}{2d}$.

Proof: Consider the circle c as shown in Figure 4 with the nearest pole p_1. Let a, b be the centers of s and c respectively. Obviously, $|\overline{ab}| < (R-d)$. Consider the right angled triangle Δabt where t is a point on the circle c. Since the radius of c is less than r, we have $|\overline{bt}| < r$. Hence, $|\overline{at}^2| = R^2 = |\overline{ab}|^2 + |\overline{bt}|^2 < (R-d)^2 + r^2$ giving $R < \frac{r^2+d^2}{2d}$. ♣

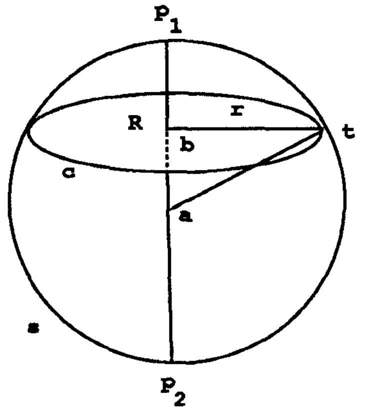


Figure 4: Lemma 2.2

3 3D Algorithm

We assume that a finite number of points is given in three dimensional space. We call the boundary of the convex hull of these points the *boundary*. In what follows, by the convex hull of a point set, we mean its interior along with its boundary. A point is called an *internal point* if it is not on the boundary and is called a *boundary point* otherwise. The facets of the boundary are referred to as *boundary facets* and the edges on the boundary facets are called *boundary edges*.

Algorithm 3D-TRI:
Input: Finite number of points in three dimensional space.

```
begin
    Let $d_1$ be the minimum of the distances
    between two points.
    Let $d_2$ be the minimum distance from an
    internal point to a boundary facet.
    Let $d_3$ be the minimum distance between
    two nonadjacent boundary facets.
    Let $r = \frac{1}{6}min\{d_1, d_2, d_3\}$.
    Triangulate each facet of the boundary using
    algorithm 2D-TRI in such a way that
    every edge has length in between $r$ and $2r$ and
    every boundary edge
    has length in between $r$ and $1.5r$.
    Let $P$ be the current point set.
    Construct a 3D Delaunay triangulation $T(P)$
    of the point set $P$.
repeat
    Add the center $v$ of the circumscribing sphere
    of a tetrahedron $t_i$ in $T(P)$
    satisfying the following properties:
    (i) all four vertices of $t_i$ are at a distance of
    at least $2r$ from $v$,
    (ii) the center $v$ lies inside the boundary.
    Set $P = P \cup v$.
    Update the Delaunay triangulation $T(P)$.
until there is no such tetrahedron.
end
```

With the above choice of r and with the assumption that all the face-angles of the facets on the boundary satisfy the minimum angle criterion, it is possible to triangulate them by *2D-TRI* maintaining the edge lengths as stated. In the following Lemma, we prove that the above procedure terminates.

Lemma 3.1: Algorithm *3D-TRI* terminates.

Proof: Algorithm *2D-TRI* terminates since the points added by it are always at a certain distance from all other points. There can be only finitely many such points inside the given polygonal boundary. Extending this argument to Algorithm 3D-TRI, we can observe that all the circumcenters of tetrahedra that are added as new points are at a distance of at least $2r$ from all other points. There can be only finitely many such points inside the convex hull of the input points, which assures the termination of the Algorithm 3D-TRI. ♣

Lemma 3.2: Any point on a boundary facet that does not lie on a boundary edge must be at a distance of at least $\frac{\sqrt{7}}{4}r$ from all edges of that facet.

Proof: Consider a point p on a facet f. Let e be any edge of f. Note that the edge e is divided into smaller edges $e_1, e_2, ..., e_n$ through the triangulation of the boundary facets adjacent to e. Drop a perpendicular from p on the line supporting e. If the perpendicular intersects the edge e, let e_l be the edge of the triangulation on e which is intersected by it. According to property 1, all boundary edges of the triangulation of f must have lengths in between r and $1.5r$. Further, the point p is at least r units away from the end points of e_l. Thus, the minimum distance between p and e_l is at least $\frac{\sqrt{7}}{4}r$. In case the perpendicular dropped from p does not intersect e, it must intersect some other edge e' of f. In that case, the distance between p and e must be greater than the distance between p and e'. We can estimate the minimum distance between p and e by estimating the same between p and e'. While estimating the distance between p and e', if it occurs that the perpendicular dropped from p does not intersect e', we will have another edge to estimate the minimum distance between p and e'. Since there are finite number of edges and since each time we go to a next edge, its distance from p gets smaller than the previous one, there must be an edge of f which is intersected by the perpendicular dropped from p. Let e'' be the first such edge encountered in the above process. As argued above, the distance between p and e'' is at least $\frac{\sqrt{7}}{4}r$. Hence, the distance between p and e is at least $\frac{\sqrt{7}}{4}r$. Thus, any point on a boundary facet that does not lie on a boundary edge must be at a distance of at least $\frac{\sqrt{7}}{4}r$ from all edges of that facet.♣

Lemma 3.3: All edges in the triangulation produced by the algorithm *3D-TRI* have lengths greater than $l_{\min}$ where $l_{\min} = \min(r, \frac{\sqrt{7}}{2}r \sin\frac{\theta_m}{2})$. Here θ_m is the minimum dihedral angle between two adjacent boundary facets.

Proof: Initially, all internal points are at a distance of at least $6r$ units from every other point. Two boundary points, lying on non adjacent facets, are at least $6r$ units away from each other. These conditions are ensured by the particular choice of r. A boundary point is at a distance of at least r from every other point on the same facet which is ensured by the algorithm *2D-TRI.* The points added by the algorithm *3D-TRI* are always at a distance of at least

$2r$ from every other point. Thus, all points except the points on the adjacent facets are at a distance of at least r from each other. To estimate the minimum distance between any two points on the adjacent boundary facets, consider two points p_i, p_j lying on the adjacent facets f_i, f_j respectively. Let e be the edge shared by f_i and f_j. Drop a perpendicular from p_i on e. Let it meet e at p_m. Consider the triangle $\Delta p_i p_j p_m$. Let the minimum dihedral angle between any two adjacent facets be θ_m. It is easy to prove that the angle between $\overline{p_i p_m}$ and $\overline{p_j p_m}$ in the triangle $\Delta p_i p_j p_m$ must be at least θ_m. From the above discussion, it follows that $|\overline{p_i p_m}| > \frac{\sqrt{7}}{4}r$ and $|\overline{p_j p_m}| > \frac{\sqrt{7}}{4}r$. Thus, the distance between p_i and p_j is at least $\frac{\sqrt{7}}{2}r \sin\frac{\theta_m}{2}$. Hence, all edges in the final triangulation produced by the algorithm *3D-TRI* have lengths greater than $l_{\min} = \min(r, \frac{\sqrt{7}}{2}r \sin\frac{\theta_m}{2})$. ♣

Lemma 3.4: Any point p present as a vertex in the triangulation produced by the algorithm *3D-TRI* is at a distance of at least $\frac{\sqrt{7}}{4}r \sin\theta_m$ from any boundary facet on which p does not lie. Here, θ_m is a measure of angle such that all dihedral angles of the input boundary are within θ_m and $180° - \theta_m$.

Proof: If p is an inner point, we already know p is at least r units away from every boundary facet. By the choice of r, any point on a boundary facet is at least r units away from any other nonadjacent facet. We prove that if p lies on a boundary facet but not on a boundary edge, it is at a distance of at least $\frac{\sqrt{7}}{4}r \sin\theta_m$ from all adjacent facets. Let p lie on f_i and let f_j be any facet adjacent to f_i. In Lemma 3.2, we proved that the distance of p from any line supporting an edge of the facet f_i , is at least $\frac{\sqrt{7}}{4}r$. Let l be the distance of p from the line where f_i and f_j meet. The distance d of p from f_j is given by $d = l \sin\theta$ where θ is the dihedral angle between f_i and f_j. Putting the minimum value of l and θ gives the lower bound on d. Thus, the distance of a point from any facet that does not contain it, is at least $d_{\min} = \min(r, \frac{\sqrt{7}}{4}r \sin\theta_m) = \frac{\sqrt{7}}{4}r \sin\theta_m$. ♣

4 Qualities of Tetrahedra

Definition: A tetrahedron in the final triangulation is said to have a ***good circumcenter*** if the center of its circumscribing sphere lies inside or on the boundary (convex hull boundary). Conversely, a tetrahedron is said to have a ***bad circumcenter*** if the center of its circumscribing sphere lies outside the boundary.
We classify the tetrahedra with bad circumcenters into two classes, namely class A and class B.

Definition: A tetrahedron t with a bad circumcenter is called a class A tetrahedron if it satisfies the following property. There exists a facet f intersected by the circumscribing sphere s of t in such a way that the foot of the perpendicular dropped from the center of s on the supporting plane of f lies inside f. Any other tetrahedron with a bad circumcenter is called a class B tetrahedron. See figure 5 and figure 6.

Assuming lower and upper bounds on the dihedral angles between adjacent boundary facets, we can prove that all tetrahedra produced by *3D-TRI* cannot be in category(i) or category(ii). Though we cannot avoid category(iii) tetrahedra, occurrences of them in practice are rare, as stated in [4]. Finally, in most of the cases these category(iii) tetrahedra can often be avoided by introducing a suitable point inside the circumscribing sphere. See [4]. In what follows, we assume that all dihedral angles between adjacent boundary facets are greater than θ_m and less than $180° - \theta_m$.

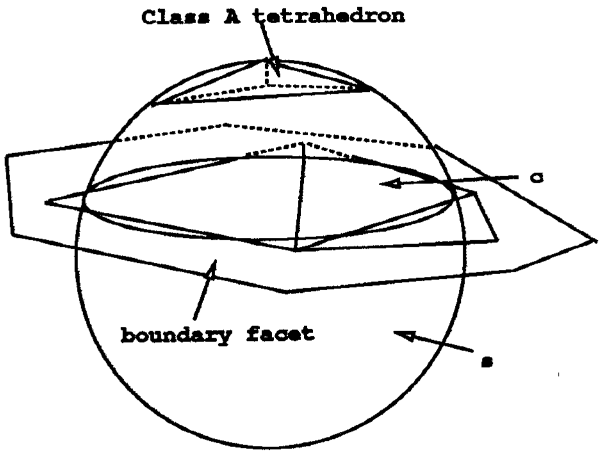


Figure 5: class A tetrahedron

Lemma 4.1: No tetrahedron with good circumcenter can be in category(i) or category(ii).

Proof: All tetrahedra in the final triangulation having good circumcenters must have circumscribing spheres with radii less than $2r$, because otherwise these circumcenters would have been introduced as new points. Hence, all these tetrahedra have edges of length less than $4r$. By Lemma 3.3, all edges have lengths greater than $\min(r, \frac{\sqrt{7}}{2}r \sin\frac{\theta_m}{2})$. Thus, κ for

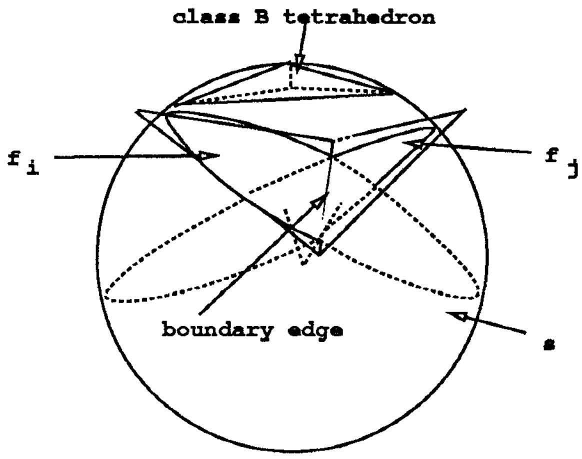


Figure 6: class B tetrahedron

these tetrahedra can be at most $\max(4, \frac{8}{\sqrt{7}\sin\frac{\theta_m}{2}})$. Assuming a lower bound on the dihedral angles of the input boundary, we get κ for these tetrahedra to be of $O(1)$ which violates the condition for category(i) tetrahedra. Further, ω for these tetrahedra can be at most $\max(2, \frac{4}{\sqrt{7}\sin\frac{\theta_m}{2}}) = O(1)$ which prohibits them to be in category(ii). ♣

Lemma 4.2: No class A tetrahedron can be in category(i) or category(ii).

Proof: Let t be a class A tetrahedron with the circumscribing sphere s. By the definition of class A tetrahedron, there exists a boundary facet f such that the foot of the perpendicular dropped from the center of s on the supporting plane of f lies inside f. Let c be the circle of intersection of S with the supporting plane of f. By Lemma 3.4, a vertex p of t that does not lie on f must be at a distance of at least $\frac{\sqrt{7}}{4}r\sin\theta_m$ from f where θ_m is defined as before. The center of the circle c lies inside f. Thus, the center must lie inside the triangulation T of f produced by the algorithm *2D-TRI*. Further, c must be an empty circle since s does not include any point of f inside it. See figure 5. By property 2, all triangles of T have circumscribing circles of radii less than r. Hence, according to Lemma 2.1, c must have a radius less than or equal to r. The vertex p lying on s must be at a distance of at least $\frac{\sqrt{7}}{4}r\sin\theta_m$ from c. Further, the vertex p and the center of s lie on the opposite sides of c. This implies c is at a distance of at least $\frac{\sqrt{7}}{4}r\sin\theta_m$ from its nearest pole. Thus, according to Lemma 2.2, s must have a radius less than or equal to k_1r where $k_1 = (\frac{\sqrt{7}\sin\theta_m}{8} + \frac{2}{\sqrt{7}\sin\theta_m})$. This puts an upper bound of $2k_1r$ on the lengths of the edges of t_i. By Lemma 3.2, all edges of t_i are greater than k_2r where $k_2 = \min(1, \frac{\sqrt{7}}{2}\sin\frac{\theta_m}{2})$. Hence, ω, κ for t_i is $O(1)$ assuming a lower bound on θ_m (A lower bound on θ_m puts lower and upper bounds on the dihedral angles between adjacent boundary facets). This prohibits it to be in category(i) or category(ii). ♣

Lemma 4.3: Let t be a class B tetrahedron with the circumscribing sphere s. There must exist two boundary facets f_i, f_j intersected by s with the following criterion:
Let c be any circle drawn on s which is normal to the line where f_i, f_j meet. The feet of the perpendiculars dropped from the center of c on the supporting planes P_i and P_j of f_i and f_j lie outside the line segments $c \cap f_i$, $c \cap f_j$.

Proof: Consider a boundary facet f_i that has the convex hull and the center of s on opposite sides. Since t has a bad circumcenter, such a facet always exists. Consider any other facet f_j sharing an edge with f_i that has been intersected by s. Drop perpendiculars from the center of s on the supporting planes of f_i and f_j. The feet of these perpendiculars lie outside f_i, f_j since t is a class B tetrahedron. Consider the great circle c' of s whose supporting plane is normal to the edge shared by f_i and f_j. The feet of the perpendiculars dropped from the center of s on the supporting planes P_i and P_j of f_i and f_j cannot lie on the line segments $c' \cap f_i$ and $c' \cap f_j$. Two different cases are shown in figure 7. This immediately implies that the condition stated in Lemma 4.3 is true for any circle c on s that has a supporting plane parallel to that of c'.♣

Lemma 4.4: No class B tetrahedron can be in category(i) or category(ii).

Proof: Let t be a class B tetrahedron. Let the circumscribing sphere s of t intersect the boundary edge e shared by the facets f_i and f_j which satisfy the criterion as stated in Lemma 4.3. The endpoints of the edge segment e_n on e which is intersected by s cannot be inside s. Let w, y be the points where s intersects e_n. Further, let a and R denote the center and radius of s respectively.

Case(i): The tetrahedron t has a vertex p which lies neither on facet f_i nor on facet f_j. Consider the circle c on s whose supporting plane is perpendicular to e_n and which passes through p. Let R' be the radius

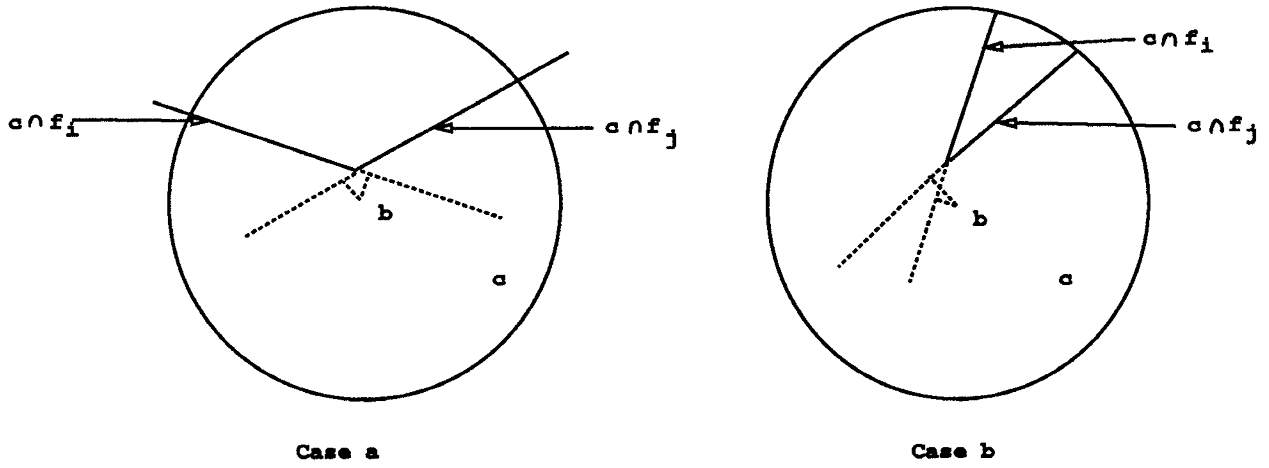
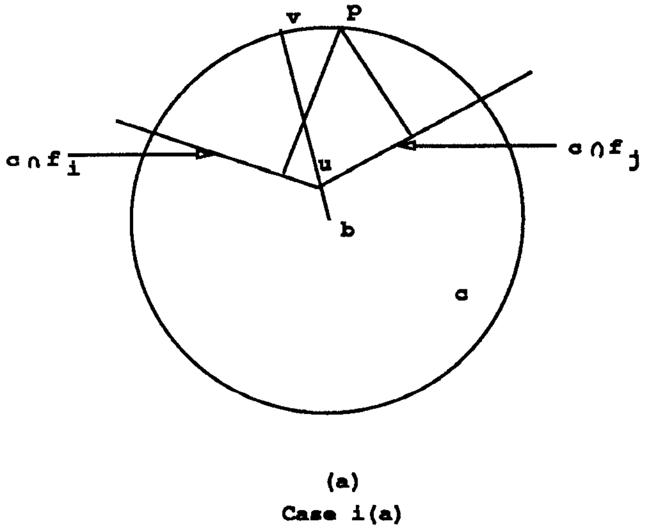


Figure 7: Lemma 4.3.



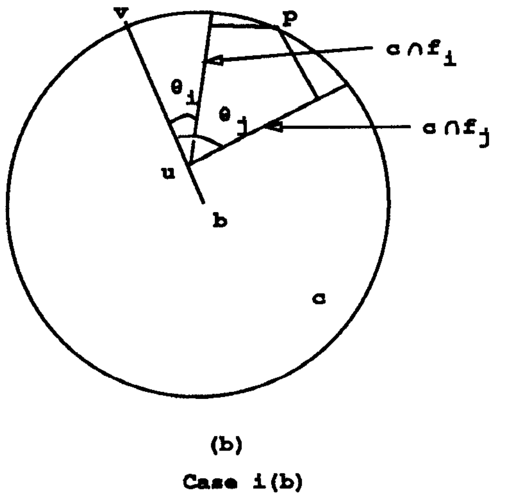


Figure 8: Lemma 4.4, case (i).

of c. Join the center b of c with the point u where c meets e_n. Extend the line $\overline{bu}$ beyond u until it intersects the boundary of c at v as shown in figure 8. Let $|\overline{bu}| = x$. Certainly, $|\overline{uv}| = R' - x$. Let d denote the minimum distance of p from the two facets f_i and f_j. There are two subcases as shown in figure 8. In subcase $i(a)$, the center of c lies in the sides of the planes containing f_i, f_j which are opposite to those containing the convex hull. It is not difficult to see that in this subcase $d \leq |\overline{uv}| = R' - x$. Since, $R \geq R'$, we have $d \leq R - x$. To estimate a lower bound on x, drop a perpendicular $\overline{az}$ from the center a of s on e_n. This perpendicular has the same length as $\overline{bu}$. Consider the triangle Δawy. We observe that $|\overline{az}| = \sqrt{R^2 - \frac{|\overline{wy}|^2}{4}}$. Since e_n can have a length of at most $1.5r$, we have $x = |\overline{az}| \geq \sqrt{R^2 - \frac{9r^2}{16}}$. Thus, $d \leq R - \sqrt{R^2 - \frac{9r^2}{16}}$. We already know $d \geq \frac{\sqrt{7}}{4}r \sin\theta_m$ (Lemma 3.4). Hence,

$$\begin{aligned}\frac{\sqrt{7}r}{4}\sin\theta_m &\leq R - \sqrt{R^2 - \frac{9r^2}{16}},\\ R &\leq \frac{7\sin^2\theta_m + 9}{8\sqrt{7}\sin\theta_m}r.\end{aligned}$$

Now, consider the subcase $i(b)$. In this subcase, one of the supporting planes of f_i and f_j has the center of c and the convex hull on its opposite sides and the other one has them on same side. Without loss of generality, assume that the supporting plane of f_i has them on same side as shown in figure 8(b). The line segments $c \cap f_i$ and $c \cap f_j$ make angles less than equal to 90° with $\overline{uv}$. Otherwise, f_i, f_j do not satisfy the criterion as stated in Lemma 4.3. In this subcase, we have $d \leq R - x$ since the distance of v from the supporting plane of f_j is greater than that of p from the same plane. Thus, in both subcases $i(a)$ and $i(b)$, we have,

$$R \leq \frac{7\sin^2\theta_m + 9}{8\sqrt{7}\sin\theta_m}r.$$

Case(ii): All vertices of the tetrahedron t lie either on f_i or on f_j. This immediately implies that one of the vertices of t lies on f_i but not on f_j and another on f_j but not on f_i. Consider the vertex p_i lying on f_i but not on f_j. Let c be the circle passing through p_i with the supporting plane being perpendicular to e_n. As in the previous case, let b be the center of c, u be the foot of the perpendicular dropped from b to e_n, and v be the point of intersection of the line $\overline{bu}$ and the circle c such that u is in between b and v. Again, we have two subcases as shown in figure 9. Consider the subcase $ii(a)$. We have $|\overline{p_iu}| \leq \frac{|\overline{uv}|}{\cos\theta_i}$, where θ_i is the angle between $\overline{p_iu}$ and $\overline{uv}$. We proved in lemma 3.2 that the distance of any point on a boundary facet that does not lie on any of its edges is at least $\frac{\sqrt{7}}{4}r$ away from any of its edges. Thus, $|\overline{p_iu}| \geq \frac{\sqrt{7}}{4}r$. Hence, $\frac{\sqrt{7}}{4}r \leq \frac{R'-x}{\cos\theta_i} \leq \frac{R-x}{\cos\theta_i}$, where $x = |\overline{bu}|$. Similarly, considering the vertex p_j of t lying on f_j but not on f_i, we can prove that $\frac{\sqrt{7}}{4}r \leq \frac{R-x}{\cos\theta_j}$, where θ_j is the angle between $f_j \cap c$ and $\overline{uv}$. The angle $\theta = \theta_i + \theta_j$ is the dihedral angle between f_i and f_j. Since one of θ_i, θ_j is less than or equal to 90° and the cos function decreases monotonically from 0° to 90°, we have $\frac{\sqrt{7}}{4}r \leq \frac{R-x}{\cos\frac{\theta}{2}}$. By the same argument as in case(i), we get $x \geq \sqrt{R^2 - \frac{9}{16}r^2}$. Hence,

$$\begin{aligned}\frac{\sqrt{7}}{4}r &\leq \frac{R - \sqrt{R^2 - \frac{9}{16}r^2}}{\cos\frac{\theta}{2}}\\ R &\leq \frac{\frac{9}{16}r^2 + \frac{7}{16}r^2\cos^2\frac{\theta}{2}}{\frac{\sqrt{7}}{2}r\cos\frac{\theta}{2}}.\end{aligned}$$

Assuming an upper bound on $\theta \leq (180° - \theta_m)$ we have,

$$R \leq \frac{7\sin^2\frac{\theta_m}{2} + 9}{8\sqrt{7}\sin\frac{\theta_m}{2}}r.$$

Now, consider the subcase $ii(b)$. The angles between $\overline{uv}$ and the line segments $c \cap f_i$ and $c \cap f_j$ are less than 90° since otherwise f_i, f_j violate the condition of Lemma 4.3. Without loss of generality assume that $\theta_i < \theta_j$. The distance between v and $c \cap f_j$ is greater than that between p_i and $c \cap f_j$. This implies $d \leq R - x$ giving the same upper bound on R as we derived in case(i).

Thus, all class B tetrahedra have a circumscribing sphere of radius $k_1 r$ where $k_1 = O(1)$ assuming lower and upper bounds on the dihedral angles between adjacent boundary facets. This with the fact that edges of all tetrahedra have lengths greater than $k_2 r$ where $k_2 = O(1)$ (recall Lemma 3.3), makes ω and κ of these tetrahedra to be of $O(1)$ and thus prohibits them to be in category(i) or category(ii). ♣

The following Theorem is immediated from Lemmas 4.1, 4.2, and 4.4.

Theorem 4.1: Algorithm *3D-TRI* triangulates the convex hull of a three dimensional point set with the guarantee that no tetrahedron of type(i) through type(iv) are generated assuming lower and upper bounds on the dihedral angles between adjacent boundary facets of the convex hull.

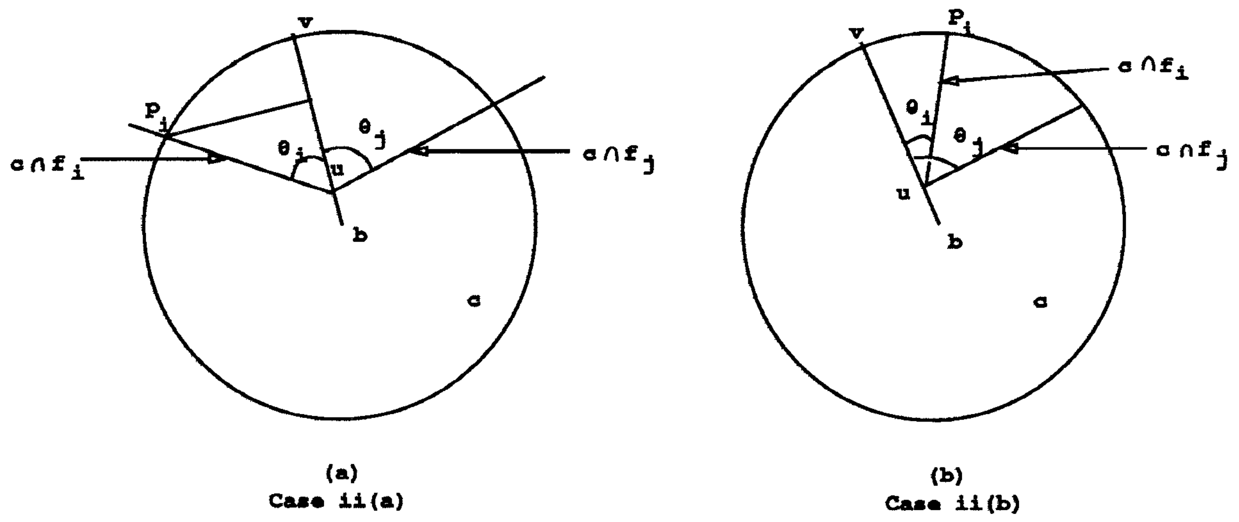


Figure 9: Lemma 4.4, case (ii).

5 Complexity

Algorithm *3D-TRI* produces tetrahedra whose edges are greater than l_{min} as defined in Lemma 3.3. The circumscribing sphere of each such tetrahedron must have a volume of $\Omega(l^3_{min})$. Let V be the volume of the convex hull of the given point set. Let n and n_o be the number of points present in the input and output respectively. Certainly, $n_o = O(\frac{V}{l^3_{min}})$. Consider a triangulation T of the input point set where $|T| = O(n)$. Such a triangulation always exists. See [11]. Let L be the largest edge length in T. All tetrahedra in T have a volume less than L^3. Thus, $V = O(nL^3)$. This gives an upper bound of $O(n\frac{L^3}{l^3_{min}})$ on n_o. Putting $A = \frac{L}{l_{min}}$, we have $n_o = O(nA^3)$. The quantity A captures the notion of how badly distributed the input point set is.

The basis of *3D-TRI* is the incremental Delaunay triangulation algorithm. We use Watson's algorithm [17] for this purpose. In this algorithm, all tetrahedra whose circumscribing spheres contain the inserted point inside are removed. The new point is connected to the triangles present in the boundary of the union of all removed tetrahedra to produce new triangulation. In *3D-TRI* we introduce the circumcenters of tetrahedra that satisfy specific properties as new points. We maintain a queue of all such tetrahedra throughout the algorithm. This queue supports deletion and addition of an element in logarithmic time. Thus, we can pick a tetrahedron t_i whose circumcenter is to be added in $O(\log n_o)$ time. We can determine all tetrahedra to be removed and to be added in $O(n_o)$ time once we have chosen t_i. This is because there are at most $O(n_o)$ tetrahedra to be removed and added for each insertion and they form a connected component together. Updating the queue for these removed and added tetrahedra takes $O(n_o \log n_o)$ time which dominates the time complexity for a single insertion. Thus, inserting all valid circumcenters takes $O(n_o^2 \log n_o)$ time. Algorithm *2D-TRI* cannot take more than $O(n_o^2)$ time [8]. Hence, *3D-TRI* takes $O(n_o^2 \log n_o) = O(n^2 A^6 \log n \log A)$ time and $O(n_o) = O(nA^3)$ space.

6 Implementation Issues

We consider the problem of numerical errors under finite precision arithmetic while implementing the algorithm *3D-TRI*. The basic numerical computation in the incremental Delaunay triangulation is the insphere test. This test tells us whether a point is inside the circumscribing sphere of a tetrahedron or not. In presence of numerical error this test may provide inaccurate answers. This, in turn, may cause the program to fail. To overcome this problem, we use rules that guide the answers of the insphere tests to satisfy the following topological constraints. While inserting a point we require that the boundary of the union of removed tetrahedra is connected and has a skeleton of a triangulated planar graph of genus zero. Furthermore, not all tetrahedra incident on a vertex are removed so that the vertex becomes an isolated vertex.

These topological criteria can be checked without any numerical error.

Another difficulty that arises under finite precision arithmetic is the following. With numerical errors, the computed points on the boundary facets may not be exactly coplanar, and without proper care they may form very thin tetrahedra. While constructing the triangulation of the point set obtained by triangulating all boundary facets we take into account the topological constraint that the points generated on a boundary facet are coplanar. The details of this robust Delaunay triangulation will appear in [16]. Currently, the implementation is being carried out on SUN workstations in AKCL.

7 Conclusion

The good triangulation algorithm of convex polyhedra together with the convex decomposition algorithm of nonconvex polyhedra [9] gives a method for good triangulations of nonconvex polyhedra as well. However, this method has the limitation that the convex polyhedra produced by the convex decomposition algorithm may be very bad in shape. An algorithm that achieves good triangulations directly of nonconvex polyhedra is more practical.

Though, in our algorithm we avoided type(i) through type(iv) tetrahedra, we could not avoid some special type of slivers i.e., type(v) tetrahedra. Our immediate goal is to find a new method or modify this algorithm so that we can avoid these slivers too. The difficulty with the avoidance of these slivers comes from the fact that an upper bound on the radius of circumscribing sphere and a lower bound on lengths of the edges of a tetrahedron do not prohibit it to be a type(v) tetrahedron. A lower bound on the radius of the inscribing sphere together with an upper bound on the radius of the circumscribing sphere of a tetrahedron avoids such tetrahedra. But, currently we are unable to achieve both these bounds simultaneously.

References

[1] D. Avis and H. ELGindy, (1986), "Triangulating Simplicial Point Sets in Space", *Proc. 2nd. Ann. ACM Symposium on Computational Geometry*, pp. 133-141.

[2] Babuska and A.K.Aziz, (1976), "On the Angle Condition in the Finite Element Method", *SIAM Numerical Analysis*, 13, 214-226.

[3] B. S. Baker, E. Grosse, and C.S. Rafferty, (1988), "Nonobtuse Triangulation of Polygons", *Discrete and Computational Geometry*, 3, 147-168.

[4] T.J. Baker, (1989), "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation", *Engineering with Computers*, 5, 161-175.

[5] M. Bern, D. Eppstein, and J. Gilbert, (1990), "Provably Good Mesh Generation", *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science*, pp. 231-241.

[6] B. Chazelle and L. Palios, (1990), "Triangulating a Non-convex Polytope", *Discrete and Computational Geometry*, 5, pp. 505-526.

[7] L. P. Chew, (1989), "Guaranteed-Quality Triangular Meshes", Technical Report TR-89-983, Cornell University.

[8] T. Dey, (1990), "Good Triangulations in Plane", *Proc. of Second Canadian Conference in Computational Geometry*, 102-106.

[9] T. Dey, (1991), "Triangulation and CSG Representation of Polyhedra with Arbitrary Genus", to appear in the *Proc. of 7th Annual Symposium on Computational Geometry*, to be held at North Conway, New Hampshire, 10-12 June, 1991.

[10] H. Edelsbrunner,(1989), "Spatial Triangulations with Dihedral Angle Conditions", *Proc. of Intl. Workshop on Discrete Algorithms and Complexity, Fukuoka, Japan*, 83-89.

[11] H. Edelsbrunner, F.P. Preparata and D.B. West, (1986), "Tetrahedrizing Point Sets in Three Dimensions", Tech. Report UIUCDCS-R-86-1310.

[12] I. Fried, (1972), "Condition of Finite Element Matrices Generated from Nonuniform Meshes", *AIAA J.*, 10, pp. 219-221.

[13] B. Joe., (1989), "Three-dimensional Triangulations from Local Transformations", *SIAM J. Sci. Stat. Comput.*, 10, pp. 718-741.

[14] F.P. Preparata, and M.I. Shamos, (1986), "Computational Geometry, An Introduction", *Springer-Verlag*.

[15] D.T. Lee, and A.K. Lin,(1986), "Generalized Delaunay triangulation for planar graphs", *Discrete and Computational Geometry*, 1, 201-217.

[16] K. Sugihara, T. Dey, and C. Bajaj, (1991), "A Robust Method for Finding Geometric Intersection of Three-Dimensional Objects", in preparation.

[17] D. F. Watson, (1981), "Computing the n-Dimensional Tesselation with Applications to Voronoi Polytopes", *The Computer Journal*, 24, pp. 167-172.