SIAM J. SCI. STAT. COMPUT.
Vol. 10, No. 4, pp. 718-741, July 1989

007

THREE-DIMENSIONAL TRIANGULATIONS FROM LOCAL TRANSFORMATIONS*

BARRY JOE†

Abstract. A new algorithm is presented that uses a local transformation procedure to construct a triangulation of a set of n three-dimensional points that is pseudo-locally optimal with respect to the sphere criterion. It is conjectured that this algorithm always constructs a Delaunay triangulation, and this conjecture is supported with experimental results. The empirical time complexity of this algorithm is $O(n^{4/3})$ for sets of random points, which compares well with existing algorithms for constructing a three-dimensional Delaunay triangulation. Also presented is a modification of this algorithm for the case that local optimality is based on the max-min solid angle criterion.

Key words. three-dimensional triangulation, Delaunay triangulation, max-min solid angle criterion, computational geometry, mesh generation

AMS(MOS) subject classifications. 65N50, 68Q20, 68U05

1. Introduction. The three-dimensional triangulation problem is as follows. Given n three-dimensional points, connect them into nonoverlapping tetrahedrons that fill the convex hull of the points. There are many ways to triangulate the n points. Special triangulations include the Delaunay triangulation and the triangulation satisfying the max-min solid angle criterion. Algorithms for constructing a Delaunay triangulation in k-dimensional space for $k \geqq 2$ are given by Bowyer [2], Watson [14], and Avis and Bhattacharya [1]. For the three-dimensional case, the estimated time complexity is $O(n^{4/3})$ for Bowyer's algorithm and higher for the other two algorithms. Applications of three-dimensional triangulations include finite-element mesh generation (Nguyen [11], Cavendish, Field, and Frey [3]), where it is usually desired to avoid small angles in triangulations, and interpolation and contouring (Petersen, Piper, and Worsey [12]).

In this paper, we investigate a local transformation procedure for three-dimensional triangulations that is analogous to the procedure of Lawson [8] for two-dimensional triangulations, and use this local transformation procedure in a new algorithm for constructing three-dimensional pseudo-locally optimal triangulations, where local optimality is based on either the sphere criterion (satisfied by Delaunay triangulations) or the max-min solid angle criterion. We conjecture that our algorithm always constructs a Delaunay triangulation in the case of the sphere criterion.

In § 2, preliminary definitions and results are given. In § 3, the main theoretical results are presented for the sphere criterion. In § 4, an algorithm and data structure are given for constructing a pseudo-locally optimal triangulation with respect to the sphere criterion, assuming that no four vertices are co-planar. In § 5, this assumption is removed and the algorithm is extended. In § 6, optimal triangulations with respect to the max-min solid angle criterion and their computation are discussed. In § 7, experimental results are presented for a Fortran implementation of the algorithms. In § 8, concluding remarks and open problems are given.

2. Preliminaries. Let S be a set of $n \geqq 4$ three-dimensional points (or vertices) that are not all co-planar. A triangulation of S is valid if and only if (a) the four

* Received by the editors March 21, 1988; accepted for publication (in revised form) September 16, 1988. This work was partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

† Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2H1.

vertices of any tetrahedron are not co-planar; (b) any tetrahedron *abcd* contains no points of $S-\{a, b, c, d\}$; (c) the intersection of the interior of any two tetrahedrons is empty; and (d) a triangular face is either on the boundary of the convex hull of S (and occurs in exactly one tetrahedron), or it is common to exactly two tetrahedrons. For any valid triangulation of S, let V_b and V_i be the number of boundary and interior vertices, respectively; let E_b and E_i be the number of boundary and interior edges, respectively; let F_b and F_i be the number of boundary and interior faces (triangles), respectively; and let T be the number of tetrahedrons. For any triangulation of S, V_b, V_i, E_b, and F_b are the same. V_b is the number of vertices on the boundary of the convex hull of S and $V_i = n - V_b$. E_b and F_b are constant because all two-dimensional triangulations of the same vertices have the same number of edges and triangles.

However, different triangulations of S may have different values for E_i, F_i, and T. The above quantities satisfy the following relations (Fuhring [5]):

(1) (a) $T=(F_b+2F_i)/4$,
(b) $F_b=2V_b-4$,
(c) $F_i=V_b+2(E_i-V_i)-4$,
(d) $E_b=3F_b/2$.

Substituting (b) and (c) into (a) results in

$$T=V_b+E_i-V_i-3. \tag{2}$$

Clearly, E_i may be at most $n(n-1)/2=O(n^2)$. From (1 (c)) and (2), it can be seen that F_i and T may be at most $O(n^2)$ as well. Note that if E_i is increased by one, then F_i is increased by two and T is increased by one. It is not too difficult to construct a family of triangulations for which E_i, F_i, and T are all proportional to n^2 (see § 7).

For two-dimensional triangulations, the local transformation procedure is as follows. If two adjacent triangles of the triangulation form a strictly convex quadrilateral, then swap the common edge for the other diagonal edge of the quadrilateral to form two new triangles. Lawson [8] proves that given any two triangulations T_1 and T_2 of a set of two-dimensional points, there exists a finite sequence of local transformations (edge swaps) by which T_1 can be transformed to T_2. Lawson [9] uses this local transformation procedure in an algorithm for constructing a two-dimensional Delaunay triangulation of n points in an estimated average time of $O(n^{4/3})$.

For three-dimensional triangulations, the analogous local transformation procedure is based on the observation that a strictly convex hexahedron formed from five vertices can be triangulated in two ways, the first containing two tetrahedrons and the second containing three tetrahedrons. This is illustrated in Fig. 1, where the five vertices are a, b, c, d, and e; (i) contains the two tetrahedrons *abcd* and *abce*, and (ii) contains the three tetrahedrons *abde*, *acde*, and *bcde*. Note that (i) contains interior face *abc* and no interior edges while (ii) contains three interior faces *ade*, *bde*, *cde*, and interior edge *de*. The *local transformation procedure* is that if two (three) adjacent tetrahedrons of the triangulation form a strictly convex hexahedron as in Fig. 1, then replace the tetrahedrons by the other possible triangulation of the hexahedron containing three (two) tetrahedrons. This local transformation procedure can be considered to be a face "swap," where one interior face is "swapped" for three interior faces or vice versa. In the next two sections, we describe how this local transformation procedure can be used to construct a (nearly) Delaunay triangulation.

Two special three-dimensional triangulations of n vertices are the Delaunay triangulation and the triangulation satisfying the max-min solid angle criterion. A Delaunay triangulation satisfies the *sphere criterion*: the circumsphere of the four

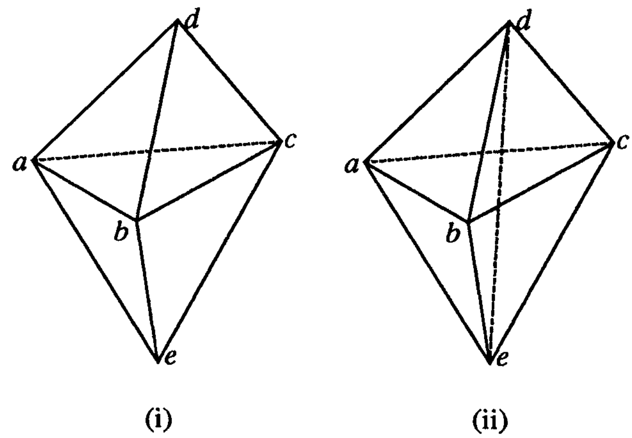

FIG. 1. *Two possible triangulations of strictly convex hexahedron.* (i) *Two tetrahedrons abcd and abce.* (ii) *Three tetrahedrons abde, acde, and bcde.*

vertices of any tetrahedron of the triangulation contains no vertices in its interior. A Delaunay triangulation is unique if no five vertices are co-spherical. The Delaunay triangulation is also the dual of the Voronoi tessellation (Bowyer [2], Watson [14]). The Voronoi tessellation of n vertices is a collection of n convex regions such that each region contains the points closer to one vertex than all the other vertices.

A tetrahedron contains twelve planar angles (three in each of the four triangular faces), six dihedral angles (one at each of the six edges), and four solid or trihedral angles at the vertices. The planar and dihedral angles are straightforward to compute. The definition and computation of a solid angle, e.g., at vertex d of tetrahedron $abcd$, are as follows. The solid angle at d is the surface area on the unit sphere formed by projecting each point on face abc to the surface of the unit sphere with d at its centre. In general, a solid angle can be defined as a double integral. In the special case of a tetrahedron, the solid angle (or spherical excess) at d can be computed as $\alpha+\beta+\gamma-\pi$ (Gasson [6]), where α, β, and γ are the dihedral angles at edges ad, bd, and cd, respectively (α, β, and γ are also the spherical angles at the projection of a, b, and c, respectively, on the unit sphere).

A triangulation satisfies the *max-min solid angle criterion* if over all possible triangulations of the vertices, the minimum of the solid angles at all vertices of all tetrahedrons is maximized. For two-dimensional triangulations, the circle and max-min angle criteria are identical, i.e., a Delaunay triangulation satisfies the max-min angle criterion and vice versa (Lawson [9]). Field [4] recently conjectured that the sphere criterion and max-min solid angle criterion are identical for three-dimensional triangulations. However, the following simple example shows that this conjecture is false. Let vertices a, b, c, d, and e have the (x, y, z) coordinates $(0,0,0)$, $(2,0,0)$, $(2,2,0)$, $(1.5,0.5,2)$, and $(1.5,0.5,-0.5)$, respectively. There are two ways to triangulate these five vertices as illustrated in Fig. 1. It is straightforward to verify by calculation that triangulation (i), containing two tetrahedrons, satisfies the max-min solid angle criterion but is not Delaunay, and triangulation (ii), containing three tetrahedrons, is Delaunay but does not satisfy the max-min solid angle criterion. In § 6, we discuss the max-min solid angle criterion further.

3. Theoretical results. In this section, we present some theoretical results for three-dimensional triangulations and the sphere criterion. Some of these results are three-dimensional versions of those in Lawson [9]. We start with definitions and results concerning the local optimality of interior faces in three-dimensional triangulations. In particular, we show that if every interior face of a triangulation is locally optimal, then it is a Delaunay triangulation. Then we discuss how the local transformation procedure given in the previous section can be used to improve an arbitrary three-dimensional triangulation to a nearly Delaunay triangulation, called a pseudo-locally optimal triangulation. We give results on when this improvement process does not terminate in a Delaunay triangulation, due to the nontransformability of some nonlocally optimal faces (unlike the two-dimensional case). Finally, we give an example of a pseudo-locally optimal triangulation that is not a Delaunay triangulation.

For simplicity, we assume for now that no four vertices are co-planar among the n vertices to be triangulated. (This assumption will be removed in § 5.) This means that the triangulation of five vertices can be three different configurations. The first two configurations are illustrated in Fig. 1 (the boundary of the convex hull contains five vertices). The third configuration occurs when the boundary of the convex hull contains four vertices; in this case, the triangulation of the five vertices consists of four tetrahedrons. For example, if vertex e is not on the boundary of the convex hull of vertices a, b, c, d, and e, the four tetrahedrons are *abce*, *abde*, *acde*, and *bcde*; there are four interior edges and six interior faces.

DEFINITION 1. Let *abcd* and *abce* be two tetrahedrons sharing common face *abc* with d and e on opposite sides of *abc*. Then interior face *abc* is said to be *locally optimal* (with respect to the sphere criterion) if the circumsphere of tetrahedron *abcd* does not contain e in its interior. (Note that the circumsphere of *abcd* contains e in its interior if and only if the circumsphere of *abce* contains d in its interior. This follows from the fact that the intersection of the circumspheres of *abcd* and *abce* is the circumcircle of triangle *abc*.)

LEMMA 1. (a) *Let a, b, c, d, e be five vertices of a convex hexahedron as in Fig.* 1. *Then either the interior face abc is locally optimal or the three interior faces ade, bde, cde are all locally optimal. Only one of these two cases holds if the five vertices are not co-spherical.*

(b) *Let abde, acde, bcde be three tetrahedrons in the configuration of Fig.* 1(ii). *Then the three interior faces ade, bde, cde are either all locally optimal or all not locally optimal.*

(c) *Let abce, abde, acde, bcde be four tetrahedrons in the third configuration. Then the six interior faces, abe, ace, ade, bce, bde, cde are all locally optimal.*

Proof. In part (a), either Fig. 1(i) or Fig. 1(ii) must be a Delaunay triangulation since these are the only two possible triangulations. A Delaunay triangulation satisfies the sphere criterion so all its interior faces are locally optimal. In the case that the five vertices are not co-spherical, only one of the triangulations can be Delaunay; the non-Delaunay triangulation does not satisfy the sphere criterion, and hence the circumsphere of at least one of its tetrahedrons contains a vertex in its interior, implying that at least one of its interior faces is not locally optimal since there are only five vertices. Therefore part (a) holds.

Suppose the non-Delaunay triangulation is Fig. 1(ii). Without loss of generality, let the circumsphere of tetrahedron *abde* contain c in its interior. Then *ade and bde* are not locally optimal. Hence the circumsphere of tetrahedron *acde* contains b in its interior, and *cde* is not locally optimal. Therefore part (b) holds.

Part (c) follows from the fact that there is only one possible triangulation in the third configuration, so it must be a Delaunay triangulation and satisfy the sphere criterion. □

THEOREM 1. *A three-dimensional triangulation T is a Delaunay triangulation* (i.e., *satisfies the sphere criterion*) *if and only if every interior face of T is locally optimal.*

Proof. The "only if" part is clearly true by definition.

Suppose every interior face of T is locally optimal. We will show by contradiction that the sphere criterion is satisfied. Suppose $abcd$ is a tetrahedron in T such that the circumsphere of $abcd$ contains vertex p in its interior. Without loss of generality, suppose p is on the opposite side of face abc from d. Then abc must be an interior face. Let $abce$ be the other tetrahedron with face abc. Then $abce$ does not contain p and abc is locally optimal, i.e., e is not in the interior of the circumsphere S of $abcd$. Let S' be the circumsphere of $abce$. If e is on S, then S' clearly contains p in its interior. Suppose e is exterior to S. Let $R = (\text{interior of } S) \cap H$, where H is the half-space containing p that is determined by the plane containing a, b, and c. Since the intersection of S and S' is the circumcircle of triangle abc, S' must contain R in its interior, and thus S' must contain p in its interior.

The above argument can be repeated with tetrahedron $abce$ replacing $abcd$, etc. The result is a sequence of connected nonoverlapping tetrahedrons such that the circumsphere of each tetrahedron contains p in its interior. Let this sequence of tetrahedrons be $a_0b_0c_0d_0$, $a_1b_1c_1d_1, \cdots$, where $a_ib_ic_id_i$ and $a_{i+1}b_{i+1}c_{i+1}d_{i+1}$ share common face $a_ib_ic_i$, i.e., $a_{i+1}b_{i+1}c_{i+1}d_{i+1} \equiv a_ib_ic_ie_i$, where e_i is either a_{i+1}, b_{i+1}, or c_{i+1}. From the argument in the previous paragraph, e_i and p are on the same side of $a_ib_ic_i$ for all i. Since there is a finite number of tetrahedrons, the sequence must contain a cycle, i.e., $a_jb_jc_jd_j \equiv a_kb_kc_kd_k$ for some $j < k$. But this results in a contradiction, since it is not possible to have a cycle of connected tetrahedrons such that e_i and p are on the same side of $a_ib_ic_i$ for $i = j, \cdots, k-1$. □

DEFINITION 2. Let $abcd$ and $abce$ be two tetrahedrons sharing interior face abc in a triangulation T. Then face abc is said to be *transformable* if either (i) the two tetrahedrons are in the configuration of Fig. 1(i), i.e., line segment de intersects the interior of triangle abc, or (ii) the boundary of the convex hull of a, b, c, d, e contains all five vertices, de does not intersect triangle abc (i.e., $abcd \cup abce$ is not convex), and the third tetrahedron needed to fill the convex hull of the five vertices is present in T. If the third tetrahedron in case (ii) is not present in T, then abc is *not transformable,* i.e., the local transformation procedure cannot be applied.

The third tetrahedron in case (ii) is either $abde$, $acde$, or $bcde$. It is $abde$ if ab intersects the interior of triangle cde; it is $acde$ if ac intersects the interior of triangle bde; it is $bcde$ if bc intersects the interior of triangle ade (see Fig. 2). If $abcd$, $abce$ are in the third configuration, i.e., either a, b, or c is not on the boundary of the convex

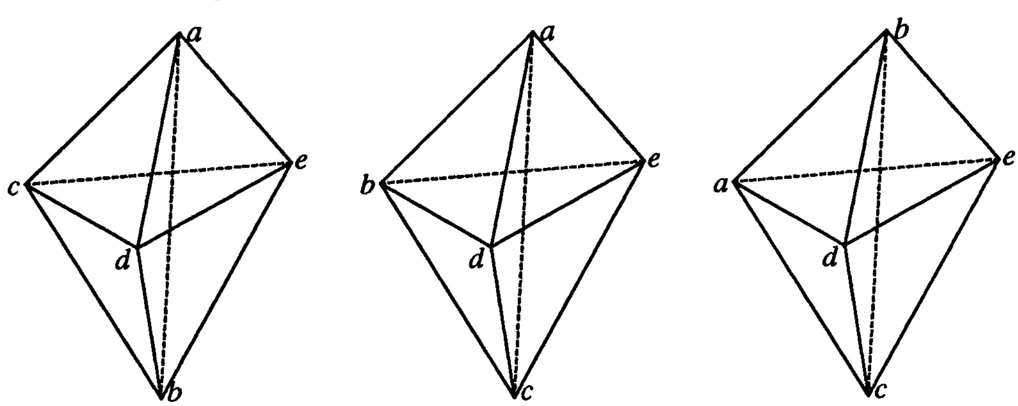

FIG. 2. *Three possible labelings of vertices in Definition* 2(ii).

hull of the five vertices, then ab does not intersect triangle cde, ac does not intersect triangle bde, and bc does not intersect triangle ade (see Fig. 3). Therefore the configuration of $abcd$, $abce$ can be determined by line segment and triangle intersection tests.

DEFINITION 3. A triangulation T is said to be *pseudo-locally optimal* (with respect to the sphere criterion) if every nonlocally optimal interior face in T is not transformable. Note that a Delaunay triangulation is pseudo-locally optimal.

LEMMA 2. *Let a, b, c, d, e be the vertices of a strictly convex hexahedron that can be triangulated by tetrahedrons $T_1 = \{abcd, abce\}$ or $T_2 = \{abde, acde, bcde\}$ as in Fig.* 1. *Let r_1 (r_2) be the minimum of the radii of the circumspheres of the tetrahedrons in T_1 (T_2). Then $r_k \leqq r_{3-k}$ if and only if T_k is a Delaunay triangulation of the five vertices, where $k = 1$ or* 2.

Proof. Consider spheres expanding at the same rate from centres a, b, c, d, and e. Let v be the first point (Voronoi vertex) where four or more expanding spheres intersect. If v is the intersection of all five spheres, then the five vertices are co-spherical, $r_1 = r_2$, and both T_1 and T_2 are Delaunay triangulations. Suppose only four expanding spheres intersect at v. Then the tetrahedron formed from the centre of these four spheres has the circumsphere with the smallest radius among the five possible tetrahedrons, and it belongs to the (unique) Delaunay triangulation. □

LEMMA 3. *Let T be a triangulation with m tetrahedrons, and let $R = (r_1, r_2, \cdots, r_m)$ be the nondecreasing sequence of circumradii of tetrahedrons in T. Suppose two or three adjacent tetrahedrons in T form a strictly convex hexahedron* (*as in Fig.* 1) *such that the one or three interior faces are not locally optimal, and the local transformation procedure is applied to these tetrahedrons. Let $R' = (r'_1, r'_2, \cdots, r'_{m\pm1})$ be the nondecreasing sequence of circumradii of tetrahedrons in the resulting triangulation T'. Then R' is lexicographically less than R* (*the shorter sequence can have an arbitrary number added at the end*).

Proof. This lemma follows from Lemma 2. □

LEMMA 4. *Let T be a triangulation containing tetrahedrons $abcd$ and $abce$ such that interior face abc is not locally optimal and is not transformable. Without loss of generality, assume that the "missing" third tetrahedron is $abde$, i.e., ab intersects the interior of triangle cde. Then there exists another interior face abf, $f \neq c$, that is not locally optimal.*

Proof. Let $\bigcirc abcd$ denote the open ball, which is the interior of the circumsphere of tetrahedron $abcd$. Let the tetrahedrons containing edge ab be $abcd$, $abdf_1$, abf_1f_2, $\cdots$, abf_ke, $abec$, $k \geqq 1$ in circular order about ab. (See the left configuration of Fig. 2 and imagine that there are more vertices between d and e.) The fact that abc is not locally optimal implies that $\bigcirc abcd$ contains e. If face abd is not locally optimal, then the lemma holds. Hence, suppose abd is locally optimal, i.e., $\bigcirc abcd$ does not contain f_1. Then the part of $\bigcirc abcd$ on the opposite side of abd from c is in $\bigcirc abdf_1$, thus $\bigcirc abdf_1$ contains e. This argument can be repeated with face abf_1, abf_2, $\cdots$ in place of abd. If faces abd, abf_1, $\cdots$, abf_{k-1} are all locally optimal, then $\bigcirc abcd$, $\bigcirc abdf_1$, $\cdots$,

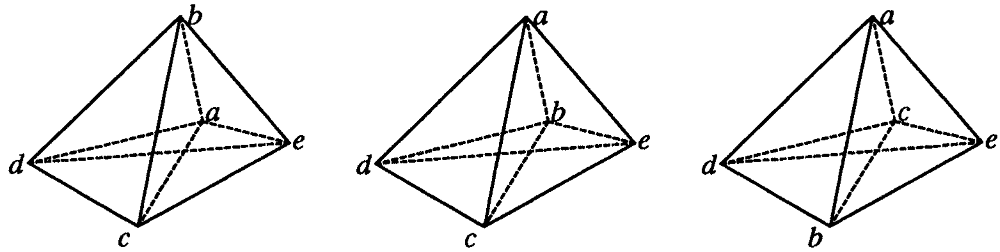

FIG. 3. *Three possible labelings of vertices in third configuration.*

$\bigcirc abf_{k-1}f_k$ all contain e. But that $\bigcirc abf_{k-1}f_k$ contains e implies that face abf_k is not locally optimal. □

DEFINITION 4. Let $abc|de$ denote a pair of adjacent tetrahedrons $abcd$, $abce$ sharing interior face abc such that abc is nonlocally optimal and nontransformable and ab intersects the interior of face cde. We call $abc|de$ a NLONT-*configuration.* Let $C=[a_0b_0c_0|d_0e_0, a_1b_1c_1|d_1e_1, \cdots, a_mb_mc_m|d_me_m]$ denote a sequence of NLONT-configurations. Then C is said to be a *connected* NLONT-*sequence* if $a_ib_ic_i$ and $a_{i+1}b_{i+1}c_{i+1}$ are distinct and share edge a_ib_i (which means that a_ib_i and $a_{i+1}b_{i+1}$ share at least one vertex) for $i=0, 1, \cdots, m-1$; and C is said to be a *connected* NLONT-*cycle* if, in addition, $a_0b_0c_0|d_0e_0=a_mb_mc_m|d_me_m$. Note that the smallest cycle length is $m=2$, which can occur only if $a_0b_0=a_1b_1$.

An example of a connected NLONT-cycle is given below in Fig. 4: $C=[452|67, 475|28, 247|15, 452|67]$, where a vertex is indicated by an integer from 1 to 8. Note that edge 45 intersects the interior of face 267, edge 47 intersects the interior of face 528, and edge 24 intersects the interior of face 715.

LEMMA 5. *Let T be a non-Delaunay triangulation. If T is pseudo-locally optimal, then T contains a connected* NLONT-*cycle.*

Proof. Since T is not Delaunay, it contains at least one nonlocally optimal interior face by Theorem 1. Since T is pseudo-locally optimal, all nonlocally optimal faces are not transformable. So T contains a NLONT-configuration $a_0b_0c_0|d_0e_0$. By Lemma 4, there exists another interior face $a_0b_0f_0$ that is not locally optimal, so T must contain another NLONT-configuration $a_1b_1c_1|d_1e_1$, where $a_1b_1c_1=a_0b_0f_0$. This argument can be repeated for $a_1b_1c_1|d_1e_1$ in place of $a_0b_0c_0|d_0e_0$, then for $a_2b_2c_2|d_2e_2$, etc. The result is a connected NLONT-sequence $[a_0b_0c_0|d_0e_0, a_1b_1c_1|d_1e_1, \cdots]$. Since there is a finite number of faces in T, there must exist j and k such that $0\leqq j<k$ and $a_jb_jc_j=a_kb_kc_k$, i.e., T contains a connected NLONT-cycle. □

THEOREM 2. *Every non-Delaunay triangulation can be transformed to a pseudo-locally optimal triangulation by a finite sequence of local transformation procedures applied to nonlocally optimal transformable interior faces.*

Proof. Let T_0 be a non-Delaunay triangulation, and let $T_0, T_1, T_2, \cdots$ be a sequence of triangulations where T_{i+1} is obtained from T_i by applying the local transformation procedure to a nonlocally optimal transformable interior face of T_i if such a face exists; otherwise, the sequence terminates at T_i. Let R_i be the nondecreasing sequence of circumradii of tetrahedrons in T_i. From Lemma 3, R_{i+1} is lexicographically less than R_i for all i. Since the R_i are lexicographically decreasing as i increases, it is not possible for the sequence of triangulations to contain a cycle, so the sequence must terminate in a pseudo-locally optimal triangulation T_m. □

COROLLARY 1. *Let T_0 be a non-Delaunay triangulation, and let $T_0, T_1, \cdots, T_m$ be a sequence of triangulations where for $i<m$, T_i is not pseudo-locally optimal and T_{i+1} is obtained from T_i by applying the local transformation procedure to a nonlocally optimal transformable interior face of T_i, and T_m is pseudo-locally optimal. If T_m does not contain a connected* NLONT-*cycle, then T_m is a Delaunay triangulation.*

Proof. This corollary follows from Lemma 5 and Theorem 2. □

If every non-Delaunay triangulation is not pseudo-locally optimal, then it would be straightforward to derive an algorithm to construct a Delaunay triangulation using the local transformation procedure. Unfortunately, we have found an example of a triangulation of eight vertices that is pseudo-locally optimal but not Delaunay. By Lemma 5, this triangulation must contain at least one connected NLONT-cycle. The eight vertices are given in Table 1, the tetrahedrons in the pseudo-locally optimal non-Delaunay and Delaunay triangulations are given in Table 2, and the NLONT-

TABLE 1
Vertex coordinates.

Index	x	y	z
1	0.054	0.099	0.993
2	0.066	0.756	0.910
3	0.076	0.578	0.408
4	0.081	0.036	0.954
5	0.082	0.600	0.726
6	0.085	0.327	0.731
7	0.123	0.666	0.842
8	0.161	0.303	0.975

TABLE 2
Pseudo-locally optimal non-Delaunay triangulation (left) and Delaunay triangulation (right). A tetrahedron is described by its four vertex indices.

1	2	3	5	1	2	3	5
1	2	4	6	1	2	5	6
1	2	4	7	1	2	6	8
1	2	5	6	1	3	4	6
1	2	7	8	1	3	5	6
1	3	4	6	1	4	6	8
1	3	5	6	2	3	5	7
1	4	7	8	2	5	6	8
2	3	5	7	2	5	7	8
2	4	5	6	3	4	6	8
2	4	5	7	3	5	6	7
3	4	6	8	3	6	7	8
3	5	6	7	5	6	7	8
3	6	7	8				
4	5	6	8				
4	5	7	8				
5	6	7	8				

configurations of the former triangulation are given in Table 3. The four tetrahedrons in the connected NLONT-cycle formed from the first three entries of Table 3 are illustrated in Fig. 4.

However, we conjecture that a non-Delaunay triangulation can be transformed to a Delaunay triangulation by a finite sequence of local transformation procedures. From the above example, some of the local transformation procedures may have to be applied to locally optimal transformable interior faces. This is a special case of the following conjecture that holds for two-dimensional triangulations. Unfortunately, the approach of Lawson [8] for proving the two-dimensional version of this conjecture does not extend to the three-dimensional case.

CONJECTURE 1. Given two different triangulations T_1 and T_2 of the same n three-dimensional vertices, T_2 can be obtained from T_1 by a finite sequence of local transformation procedures.

4. Algorithm and data structure. Based on the results of the previous section, we present an algorithm and data structure for constructing a pseudo-locally optimal triangulation of n three-dimensional vertices $v_1, v_2, \cdots, v_n$ (we are still assuming that no four vertices are co-planar). In our algorithm, the n vertices are first sorted in lexicographical order of their coordinates. In the general step, a pseudo-locally optimal

TABLE 3
NLONT-configurations. The first three form a connected NLONT-cycle. There are also three connected NLONT-cycles of length 2.

a	b	c	d	e
4	5	2	6	7
4	7	5	2	8
2	4	7	1	5
4	5	6	2	8
4	7	8	1	5
2	4	1	6	7

triangulation T_{i-1} of the first $i-1$ vertices has been constructed, and the ith vertex is added to form a preliminary triangulation of the first i vertices. Note that the ith vertex is outside the convex hull of the first $i-1$ vertices. Then the local transformation procedure is applied to nonlocally optimal transformable faces until a pseudo-locally optimal triangulation T_i of the i vertices is obtained. We now present the pseudocode for our algorithm, called TRSPH1.

In the "for" loop labelled (A) in the pseudocode, a preliminary triangulation is constructed by adding tetrahedrons with vertex v_i to T_{i-1} to fill the convex hull of the first i vertices. The new tetrahedrons are of the form $v_av_bv_cv_i$, where $v_av_bv_c$ is a boundary face of T_{i-1} but is not on the boundary of the convex hull of the first i vertices. This condition holds if and only if v_i and w are on opposite sides of $v_av_bv_c$, where w is any point in the interior of the convex hull of the first $i-1$ vertices, e.g., w is the centroid of the first tetrahedron.

Stack S is used to store the interior faces that are not known to be locally optimal or nontransformable. If $v_av_bv_cv_i$ is a new tetrahedron, then $v_av_bv_c$ may be nonlocally optimal so it is put on S. If $v_av_bv_cv_i$ and $v_av_bv_dv_i$ are adjacent new tetrahedrons, then $v_av_bv_i$ may be a nonlocally optimal interior face, but it is not put on S because the union of $v_av_bv_cv_i$ and $v_av_bv_dv_i$ is not convex, so either $v_av_bv_i$ is not transformable or tetrahedron $v_av_bv_cv_d$ is present in the triangulation, in which case $v_av_bv_cv_i$, $v_av_bv_dv_i$, $v_av_bv_cv_d$ are replaced by the two tetrahedrons $v_av_cv_dv_i$, $v_bv_cv_dv_i$ if either interior face $v_av_bv_c$ or $v_av_bv_d$ is determined to be nonlocally optimal.

It is possible that the face $v_av_bv_c$ referred to in the statement labelled (B) is no longer in the triangulation since three interior faces are replaced by one interior face

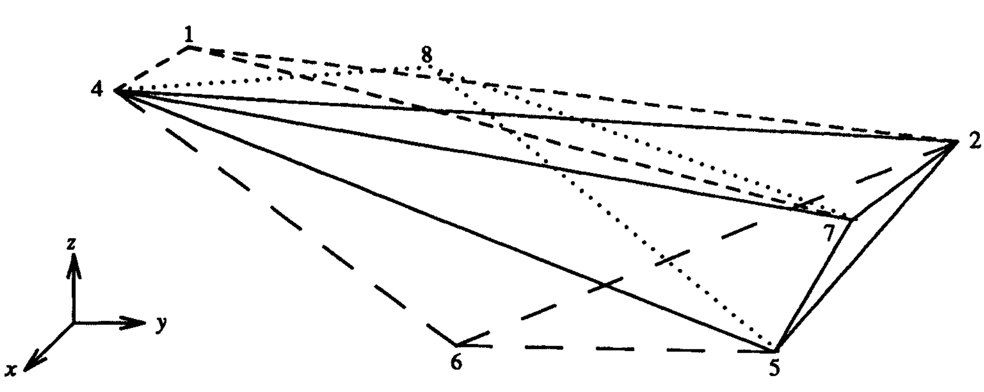

FIG. 4. *Connected* NLONT-*cycle formed from the first three entries of Table* 3. *Four tetrahedrons are*: 2457, 2456, 4578, 1247.

```
ALGORITHM TRSPH1
Sort v_1, v_2, ..., v_n into lexicographical order
Form first tetrahedron v_1 v_2 v_3 v_4
Compute centroid w = (v_1 + v_2 + v_3 + v_4)/4
Initialize S to empty stack
for i := 5 to n do
     Let T_{i-1} = current triangulation of first
       i-1 vertices (T_{i-1} is pseudo-locally optimal)
(A)  for each boundary face v_a v_b v_c of T_{i-1} do
          if v_i is on the opposite side of v_a v_b v_c from w then
               Add tetrahedron v_a v_b v_c v_i to triangulation
               Push interior face v_a v_b v_c on stack S
          endif
     endfor
     while stack S is not empty do
          Pop interior face v_a v_b v_c from stack S
(B)       if v_a v_b v_c is still in triangulation then
               Find the two tetrahedrons v_a v_b v_c v_d, v_a v_b v_c v_e sharing face v_a v_b v_c
(C)            if the circumsphere of v_a v_b v_c v_d contains v_e in its interior then
                    transform := true
                    if v_a v_b v_c v_d ∪ v_a v_b v_c v_e is a convex hexahedron then
                         Replace v_a v_b v_c v_d, v_a v_b v_c v_e by the three tetrahedrons
                              v_a v_b v_d v_e, v_a v_c v_d v_e, v_b v_c v_d v_e
                    else
                         if the third tetrahedron needed to fill the convex
                              hull of v_a, ..., v_e is present in the triangulation then
                              Relabel vertices so that three tetrahedrons are v_a v_b v_d v_e,
                                   v_a v_c v_d v_e, v_b v_c v_d v_e
                              Replace v_a v_b v_d v_e, v_a v_c v_d v_e, v_b v_c v_d v_e by v_a v_b v_c v_d, v_a v_b v_c v_e
                         else
                              transform := false
                         endif
                    endif
                    if transform then
(D)                      for each of faces v_a v_b v_d, v_a v_b v_e, v_a v_c v_d,
                              v_a v_c v_e, v_b v_c v_d, v_b v_c v_e do
                              Push face on stack S if it is an interior face and it is
                                   not yet in S
                         endfor
                    endif
               endif
          endif
     endwhile
endfor
```

in the case of two tetrahedrons replacing three tetrahedrons in the local transformation procedure. In the statement labelled (C), a test is made to see whether or not $v_av_bv_c$ is locally optimal. If not, then the local transformation procedure is applied if $v_av_bv_c$ is transformable (the configuration of the two tetrahedrons $v_av_bv_cv_d$, $v_av_bv_cv_e$ can be determined as described in Definition 2 and the paragraph after it). If the local transformation procedure is applied to $v_a v_b v_c$, then a boundary face of the convex hexahedron formed by the two or three tetrahedrons may no longer be locally optimal, so it is placed on stack S if it is an interior face of the triangulation and it is not yet in S (see statement (D)).

From the above discussion and the pseudocode, it is clear that on completion of the ith step with S empty, the triangulation of the first i vertices, T_i, is pseudo-locally optimal. If T_{i-1} is a Delaunay triangulation of the first $i-1$ vertices, then it seems likely that T_i does not contain a connected NLONT-cycle and is also Delaunay, hence we have the following conjecture.

CONJECTURE 2. Algorithm TRSPH1 constructs a Delaunay triangulation for all sets of three-dimensional vertices.

We now describe the data structure for our three-dimensional triangulation algorithm. The vertex coordinates are stored in an array VC where $VC[i].x$, $VC[i].y$, and $VC[i].z$ are the coordinates of the ith vertex, v_i. The faces and tetrahedrons are changing throughout the algorithm, and there are searching operations on the faces (e.g., find the two tetrahedrons sharing interior face $v_a v_b v_c$). Hence the faces are stored in a hash table HT with direct chaining, where $HT[i]$ is the head pointer of the linked list of faces with hashing function value i. A new face is added at the front of a linked list, since it is more likely to be referenced again. Let $a<b<c$ be the three indices in VC of the three vertices v_a, v_b, v_c of a face. A satisfactory hashing function is $h(a, b, c) = (an^2+bn+c) \bmod M$, where the hash table size, M, is a prime number. (For descriptions of hashing and linked lists, see any data structure book, e.g., Standish [13].)

We store the elements of the hash table linked lists in an array FC of face records with origin index 1, so that some fields of the face record can be used for two purposes depending on whether the face is an interior or boundary face. The fields of $FC[i]$ are a, b, c, d, $e|flink$, $stlink|blink$, $htlink$, where $0<a<b<c$ are the three vertex indices of a face; v_d and possibly v_e are the fourth vertices of the one or two tetrahedrons with (boundary or interior, respectively) face $v_a v_b v_c$; $stlink$ indicates whether or not interior face $v_a v_b v_c$ is in stack S and in the former case, it is also a pointer to the next face in S; $flink$ and $blink$ are forward and backward pointers for a doubly linked list of boundary faces (since the boundary faces must be traversed and updated in step (A) of the pseudocode); and $htlink$ is the pointer to the next element in the hash table linked list.

All four link fields represent positive indices in the FC array or zero for end of list, but due to the double use of some fields, the actual values stored in the $flink$, $blink$, and $stlink$ fields are slightly modified. If $ptr \geqq 0$ is the real pointer value, then $-ptr$ is stored in the $flink$ or $blink$ field and $ptr+2$ is stored in the $stlink$ field. For interior faces (with a positive integer in the $e|flink$ field), $stlink=1$ is used to indicate that the face is not in stack S and $stlink>1$ is used to indicate that the face is in S as well as the pointer to the next element of S. If a face is in stack S but no longer in the triangulation, then b is set to zero to indicate this and the face record is deleted when it reaches the top of S. We also use the a field to maintain an avail linked list of deleted face records (with nonpositive values to indicate pointers as for $flink$), so that a new face record can be obtained from the avail list if it is nonempty or the end of array otherwise.

With this data structure, searching, insertion, and deletion of faces are straightforward, and each operation should take constant time with a sufficiently large hash table size. At the end of the algorithm, the list of tetrahedrons in the triangulation can be obtained by sequentially traversing the array FC. Since each tetrahedron appears four times in the data structure, duplicates can be avoided as follows. For each new tetrahedron, search for the other three representations and use the *stlink*|*blink* field to indicate whether the tetrahedron determined by the d or e field has been listed already.

An example of the *VC* array is given in Table 1. An example of the *FC* and *HT* arrays is given in Fig. 5 for the four tetrahedrons in Fig. 4, where any face appearing in only one tetrahedron is taken to be a boundary face. The number of vertices is $n=8$, the hash table size is $M=5$, and the hashing function is $h(a,b,c)=(an^2+bn+c) \bmod M$. *TOP*, *HEAD*, and *TAIL* are scalar variables that are pointers to the top of stack S and the head and tail of the doubly linked list of boundary faces.

An obvious variation of algorithm TRSPH1 is to first construct an initial triangulation T_I as in step (A), then to put all interior faces of T_I in stack S, and finally to apply local transformation procedures to nonlocally optimal transformable faces of S as in the main "while" loop of TRSPH1. If the interior faces are added to stack S by sequentially traversing the array *FC* in the forward (backward) direction, then we call this algorithm TRSPH2 (TRSPH3, respectively). Note that algorithm TRSPH1 can be interpreted as constructing T_I first (although T_I never actually exists during the algorithm) and then processing the interior faces in a different order from algorithms TRSPH2 and TRSPH3. The order of processing the interior faces in TRSPH1 is closer to that in TRSPH3 than TRSPH2, since in TRSPH3 faces created closer to the beginning of the construction of T_I are closer to the top of stack S initially. Since T_I is in general not close to a Delaunay triangulation, it seems likely that TRSPH2 and TRSPH3 have a greater chance than TRSPH1 of ending up with a connected NLONT-cycle and a pseudo-locally optimal triangulation that is not Delaunay. In § 7, we report on experiments that compare these algorithms.

5. Degeneracy. In this section, we describe the extensions to the results of § 3 and algorithm TRSPH1 of § 4 when we remove the assumption that no four vertices are co-planar. Definition 1 and Theorem 1 of § 3 (about locally optimal faces) still hold when subsets of four co-planar vertices are allowed.

The degenerate configurations for two tetrahedrons *abcd* and *abce* sharing common face *abc* with d and e on opposite sides of *abc* are illustrated in Fig. 6, where the vertices of *abc* are labelled so that a, b, d, and e are co-planar and c lies on a different plane. In Fig. 6(i), quadrilateral *adbe* is strictly convex and the other triangulation of the five vertices contains tetrahedrons *acde* and *bcde*. In Fig. 6(ii), quadrilateral *adbe* degenerates to a triangle. In Fig. 6(iii), quadrilateral *adbe* is nonconvex and tetrahedron *acde* must be added to fill the convex hull of the five vertices. In the latter two cases, there are no other possible triangulations of the five vertices.

Hence, an additional case to the local transformation procedure described in § 2 is as follows. If tetrahedrons *abcd* and *abce* are in the configuration of Fig. 6(i), then replace them by tetrahedrons *acde* and *bcde*, i.e., swap interior face *abc* for face *cde*.

In the three configurations of Fig. 6, the circumcircles of faces *abd* and *abe* (which are on the circumspheres of tetrahedrons *abcd* and *abce*, respectively) are in the same plane. This implies that interior face *abc* is locally optimal if and only if the circumcircle of *abd* does not contain e in its interior (i.e., edge *ab* is locally optimal in the two-dimensional triangulation of a, b, d, e). Therefore, in Figs. 6(ii) and 6(iii), *abc* is locally optimal, and in Fig. 6(i), either *abc* or *cde* is locally optimal.

DEFINITION 5 (extension of Definition 2). Let *abcd* and *abce* be two tetrahedrons in triangulation T that are in the configuration of Fig. 6(i). Then face *abc* is said to be *transformable* if either (i) *abd* and *abe* are boundary faces of T, or (ii) *abd* and *abe* are interior faces of T and there is a vertex f (on the opposite side of *abd* from c)

HEAD: 3 TOP: 1 TAIL: 13

FC

	a	b	c	d	e | flink	blink | stlink	htlink
1	2	4	5	7	6	4	0
2	2	4	7	5	1	2	0
3	2	5	7	4	−5	0	1
4	4	5	7	2	8	1	0
5	2	4	6	5	−6	−3	0
6	2	5	6	4	−7	−5	0
7	4	5	6	2	−8	−6	2
8	4	5	8	7	−9	−7	6
9	4	7	8	5	−10	−8	3
10	5	7	8	4	−11	−9	8
11	1	2	4	7	−12	−10	10
12	1	2	7	4	−13	−11	7
13	1	4	7	2	0	−12	4

HT

0	9
1	5
2	12
3	13
4	11

FIG. 5. *Illustration of FC and HT arrays for the four tetrahedrons in Fig. 4. 245, 247, 457 are interior faces, with the former two on stack S; the remaining faces are boundary faces.*

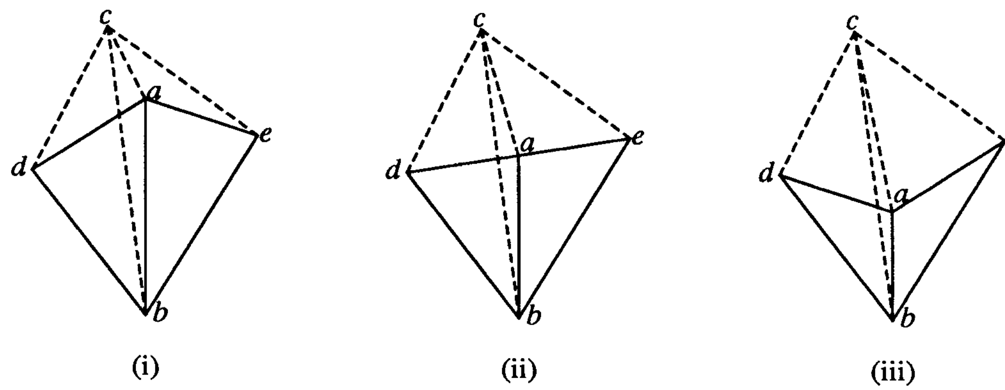

FIG. 6. *Three possible cases of two tetrahedrons with a, b, d, e co-planar.*

such that *abdf* and *abef* are tetrahedrons of *T*. Face *abc* is *not transformable*, if in case (ii), the other two tetrahedrons of *T* containing faces *abd* and *abe* are *abdf* and *abeg* where $f \neq g$.

Note that the configuration of Fig. 6(i) can be detected and distinguished from the other configurations of § 3 by the fact that *de* intersects the boundary of triangle *abc*. The main extension to algorithm TRSPH1 is to detect the configuration of Fig. 6(i) when face *abc* from stack *S* is not locally optimal and to apply the local transformation procedure to *abc* if it is transformable. In the case that *abd* and *abe* are interior faces, the local transformation procedure must be applied to both *abc* and *abf* where *f* is defined in Definition 5(ii), i.e., tetrahedrons *abcd* and *abce* are replaced by *acde* and *bcde*, and tetrahedrons *abdf* and *abef* are replaced by *adef* and *bdef*. In this case, both *cde* and *def* are locally optimal faces in the new triangulation, and the faces that may have to be put on stack *S* are *ade*, *bde*, *acd*, *ace*, *bcd*, *bce*, *adf*, *aef*, *bdf*, and *bef*. In the case that *abd* and *abe* are boundary faces, the faces that may have to be put on stack *S* are *acd*, *ace*, *bcd*, and *bce*.

The only other modification to the algorithm is a possible slight reordering of the sorted vertices to get a valid first tetrahedron (i.e., v_1, v_2, v_3, and v_4 are not co-planar). Let $v_1, \cdots, v_n$ be the sorted vertices. Let v_k, $k \geqq 3$, be the vertex of smallest index such that v_1, v_2, and v_k are not collinear. Let v_m, $m > k$ be the vertex of smallest index such that v_1, v_2, v_k, and v_m are not co-planar. Then shift the vertices to get the new ordering: $v_1, v_2, v_k, v_m, v_3, \cdots, v_{k-1}, v_{k+1}, \cdots, v_{m-1}, v_{m+1}, \cdots, v_n$. Note that with this new ordering, the vertices still satisfy the property that the ith vertex is outside the convex hull of the first $i-1$ vertices. The modifications to algorithms TRSPH2 and TRSPH3 are clearly similar.

All the remaining results in § 3 also extend to the case when subsets of four co-planar vertices are allowed. Lemmas 2 and 3 extend to the two possible triangulations in the configuration of Fig. 6(i). The extensions of Lemma 4 and Definition 4 are as follows.

LEMMA 6. *Let T be a triangulation containing tetrahedrons abcd and abce in the configuration of Fig.* 6(i) *such that interior face abc is not locally optimal and is not transformable, i.e., abdf and abeg are tetrahedrons of T where* $f \neq g$. *Then there exists another interior face abh,* $h \neq c$, *that is not locally optimal.*

Proof. The proof is similar to proof of Lemma 4. □

DEFINITION 6. In addition to the NLONT-configuration given in Definition 4, *abc*|*de* is also a NLONT-*configuration* if *abcd* and *abce* are adjacent tetrahedrons sharing interior face abc as in Fig. 6(i) such that abc is nonlocally optimal and nontransformable.

Lemma 5 still holds with the extended definition of NLONT-configuration. Theorem 2, Corollary 1, and Conjecture 1 still hold, provided T_{i+1} is obtained from T_i by two simultaneous applications of the local transformation procedure in Definition 5(ii) as discussed above.

6. Max-min solid angle criterion. In this section, we describe how our algorithms for constructing a pseudo-locally optimal triangulation with respect to the sphere criterion can be modified to construct a locally optimal triangulation with respect to the max-min solid angle criterion. The main modifications are due to the definition of "locally optimal" with respect to the max-min solid angle criterion.

DEFINITION 7. Let $abcd$ and $abce$ be two tetrahedrons sharing interior face abc in a triangulation T, where abc is a transformable face (see Definitions 2 and 5). Let T_1 contain the tetrahedrons of T that fill the convex hull of a, b, c, d, e. Let T_2 contain the tetrahedrons in the alternative triangulation of a, b, c, d, e. If abc satisfies Definition 5(ii), then let T_1 additionally contain tetrahedrons $abdf$ and $abef$, and let T_2 additionally contain tetrahedrons $adef$ and $bdef$. Then face abc is said to be *locally optimal* with respect to the max-min solid angle criterion if $s(T_1) \geqq s(T_2)$ where $s(T_i) = \min$ {solid angles at vertices of tetrahedrons of T_i}.

DEFINITION 8. A triangulation T is said to be SA-*locally optimal* if every transformable interior face in T is locally optimal with respect to the max-min solid angle criterion. (To avoid confusion with Definition 3, we use SA-*locally optimal* instead of *pseudo-locally optimal.*)

DEFINITION 9. A triangulation T of a set S of three-dimensional vertices is said to be SA-*globally optimal* if over all possible triangulations of S, the minimum of the solid angles at all vertices of all tetrahedrons is maximized in triangulation T.

Note that, unlike the case of the sphere criterion, a SA-locally optimal triangulation may not be SA-globally optimal. We have no theoretical results such as those of Theorem 1 that characterize a SA-globally optimal triangulation. It is possible that the problem of constructing a SA-globally optimal triangulation is NP-hard. The following results, which are similar to Lemma 3 and Theorem 2, indicate how an SA-locally optimal triangulation can be constructed.

LEMMA 7. Let T be a triangulation with m tetrahedrons, let a_i be the minimum of the four solid angles of a tetrahedron, and let $A = (a_1, a_2, \cdots, a_m)$ be the nondecreasing sequence of a_i values of tetrahedrons in T. Suppose the local transformation procedure is applied to a nonlocally optimal interior face (or two faces if they satisfy Definition 5(ii)) in T. Let $A' = (a'_1, a'_2, \cdots, a'_r)$ be the nondecreasing sequence of a_i values of tetrahedrons in the resulting triangulation. Then A' is lexicographically greater than A.

Proof. This lemma follows from Definition 7. □

THEOREM 3. *Every triangulation that is not* SA-*locally optimal can be transformed to a* SA-*locally optimal triangulation by a finite sequence of local transformation procedures applied to nonlocally optimal interior faces.*

Proof. This theorem follows from Lemma 7 using the same approach as in the proof of Theorem 2. □

To construct an SA-locally optimal triangulation, the algorithms given in §§ 4 and 5 must be modified, when processing a face $v_a v_b v_c$ from stack S, to first check whether $v_a v_b v_c$ is transformable before determining whether it is nonlocally optimal. The only other modification is inside the "for" loop labelled (A) in which face $v_a v_b v_i$, $v_a v_c v_i$, or

$v_bv_cv_i$ must be added to stack S if it is an interior face. The reason for this is that a face *abc* in the configuration of Fig. 6(i) may be locally optimal if *abd* and *abe* are boundary faces, but *abc* may become nonlocally optimal if *abd* and *abe* become interior faces. We call the modified algorithms TRMMSA1, TRMMSA2, TRMMSA3, with the obvious correspondence to the earlier algorithms.

7. Experimental results. We implemented algorithms TRSPH1, TRSPH2, TRSPH3, TRMMSA1, TRMMSA2, and TRMMSA3 in Fortran, and ran many test problems to compare these algorithms and to determine the empirical time complexity. It is not possible to obtain an average or worst-case time complexity analytically since the complexity depends on the number of faces tested for local optimality and the number of applications of the local transformation procedure for which we have no general bounds. The implementations used double precision floating point arithmetic and were compiled using the *f*77 compiler without the optimization option (due to compiler bugs). The tests were done on a Sun 3/50 workstation with a MC68881 floating point processor running the Sun Unix 4.2 operating system.

We used 60 test problems in our experiment, consisting of 11 problems each for the five different n values 100, 200, 300, 400, 500, and a 12th problem for the n values 50, 100, 150, 200, 250. For fixed n, the description of the 12 problems is as follows. The first seven problems have vertex coordinates that are pseudorandom numbers from the uniform distribution in $[0, a_x]\times[0, a_y]\times[0, a_z]$. Problem $P1_n$, $P2_n$, $P3_n$, $P4_n$ have $a_x = a_y = a_z = 1$; problem $P5_n$ has $a_x = 0.5$, $a_y = a_z = 1$; problem $P6_n$ has $a_x = 1$, $a_y = 0.5$, $a_z = 1$; and problem $P7_n$ has $a_x = a_y = 1$, $a_z = 0.5$. The next three problems are used to test the degenerate configurations in which four vertices are co-planar. They consist of approximately $\sqrt{n}$ parallel planes, each with an average of $\sqrt{n}$ vertices. The planes are orthogonal to one of the three axes and are determined by pseudorandom uniform numbers in $[0, 1]$. The vertices on each plane have vertex coordinates that are pseudorandom uniform numbers in $[0, 1]\times[0, 1]$. In problems $P8_n$, $P9_n$, and $P10_n$, the parallel planes have the form $x = c$, $y = c$, and $z = c$, respectively.

The last two problems are not random; one does not have a unique Delaunay triangulation and the other has a Delaunay triangulation containing $O(n^2)$ tetrahedrons. In problem $P11_n$, the vertex coordinates are on a uniform grid and have the form (i, j, k), where i, j, and k are integers in the ranges 0 to $n_x - 1$, 0 to $n_y = 1$, and 0 to $n_z - 1$, respectively, and $n = n_xn_yn_z$. For $n = 100$, $n_x = 4$, $n_y = n_z = 5$; for $n = 200$, $n_x = n_y = 5$, $n_z = 8$; for $n = 300$, $n_x = 5$, $n_y = 6$, $n_z = 10$; for $n = 400$, $n_x = 5$, $n_y = 8$, $n_z = 10$; and for $n = 500$, $n_x = 5$, $n_y = n_z = 10$. The number of tetrahedrons in a Delaunay triangulation of this problem can range from $5(n_x - 1)(n_y - 1)(n_z - 1)$ to $6(n_x - 1)(n_y - 1)(n_z - 1)$, since a unit cube can be triangulated by five or six tetrahedrons.

Problem $P12_n$ has $k = \lfloor n/2 \rfloor$ vertices that are equally spaced points on the unit circle centred about the origin in the x-y plane, and $m = n - k$ points that are equally spaced in interval $[0, 1]$ of the z-axis, i.e., $v_i = (\cos(i\alpha), \sin(i\alpha), 0)$ for $i = 1, \cdots, k$ and $v_{i+k} = (0, 0, (i-1)s)$ for $i = 1, \cdots, m$ where $\alpha = 2\pi/k$ and $s = 1/(m-1)$. All tetrahedrons in the Delaunay triangulation of this problem must consist of two vertices with index $\leqq k$ and two with index $>k$, therefore the number of tetrahedrons, faces, and boundary faces in the Delaunay triangulation are $n(n-2)/4$, $n(n-1)/2$, and n, respectively, for even n.

For our experiment, we used the hashing function given in § 4 and a hash table size $M \approx 1.5n$, where M is a prime number and n is the number of vertices, so the storage complexity of the algorithms is proportional to the number of faces in the triangulation. For all algorithms and problems $P1_n$ to $P11_n$, the average number of face records compared when a face is searched in the hash table is a small constant: $\leqq 2.0$ for TRSPH1 and TRMMSA1, $\leqq 4.7$ for TRSPH2 and TRMMSA2, and $\leqq 6.1$ for TRSPH3 and TRMMSA3. For problem $P12_n$, the average mumber of face records compared during searches is up to 8.2 for TRSPH1, TRSPH2, TRSPH3, TRMMSA1, and 19.7 for TRMMSA2, TRMMSA3 when $n = 250$; the higher numbers are due to a quadratic number of faces (the hash table size should be proportional to the number of faces in order to get a constant number of comparisons, on average).

The following quantities are used to measure the performance of the algorithms (i refers to the algorithm number in TRSPHi or TRMMSAi):

NTETi— number of tetrahedrons in triangulation;
NFACi— number of faces in triangulation;
NBFCi— number of boundary faces in triangulation;
TIMi— CPU time in seconds for constructing triangulation;
TInit— CPU time in seconds for sorting vertices and producing initial triangulation T_I (or producing preliminary tetrahedrons in step (A) of TRSPH1), so TIMi—TInit is the CPU time spent in checking faces for local optimality and transformability, applying the local transformation procedure, and updating stack S;
LOPi— number of faces that are locally optimal when tested for local optimality;
LTPi— number of applications of the local transformation procedure (the degenerate case in which four tetrahedrons are replaced by four other tetrahedrons is counted as one application);
NTFi— number of faces that are nonlocally optimal and nontransformable when tested for local optimality in TRSPHi or that are nontransformable when tested for transformability in TRMMSAi;
MSAi— minimum solid angle in radians at vertices of tetrahedrons of triangulation.

Note that LOPi + LTPi + NTFi is the number of faces on stack S that are tested for local optimality (transformability) in algorithm TRSPHi (TRMMSAi).

We first describe the experimental results from running the 60 test problems for algorithms TRSPH1, TRSPH2, and TRSPH3. TRSPH1 constructed a Delaunay triangulation for all the problems (this is verified by checking that all interior faces are locally optimal). TRSPH3 failed to construct a Delaunay triangulation for only problem $P1_{300}$; for this pseudo-locally optimal triangulation, there are 34 nonlocally optimal nontransformable interior faces. TRSPH2 constructed pseudo-locally optimal non-Delaunay triangulations for 18 of the 60 problems: $P5_{200}$, $P7_{200}$, $P10_{200}$, $P1_{300}$, $P5_{300}$, $P6_{300}$, $P8_{300}$, $P1_{400}$, $P2_{400}$, $P4_{400}$, $P5_{400}$, $P9_{400}$, $P1_{500}$, $P3_{500}$, $P4_{500}$, $P5_{500}$, $P6_{500}$, $P9_{500}$. The number of nonlocally optimal faces in these triangulations are 21, 13, 20, 27, 27, 46, 12, 74, 35, 48, 68, 13, 23, 7, 57, 27, 52, and 33, respectively. It appears that algorithm TRSPH2 is more likely to construct a non-Delaunay triangulation as n increases.

We split the measurements into three categories, the average of $P1_n$ to $P10_n$, $P11_n$, and $P12_n$, since the performance of the algorithms on the last two problems is significantly different from the random problems. The measurements for the 10 random problems are approximately the same, with problems $P5_n$ and $P8_n$ always having the highest CPU times and number of faces tested for local optimality (this is probably because the lexicographical ordering of the vertices causes these two problems to have more long tetrahedrons with small solid angles in the initial triangulation T_I). Tables 4, 5, and 6 contain measurements for the average of $P1_n$ to $P10_n$. Tables 7, 8, and 9

TABLE 4

Average of problems $P1_n$ *to* $P10_n$, *algorithm* TRSPHi: *number of tetrahedrons and faces in Delaunay triangulation and* CPU *times.*

n	NTET1	NFAC1	NBFC1	TInit	TIM1	TIM2	TIM3
100	504.0	1039.2	62.4	2.62	22.71	22.19	22.15
200	1129.2	2298.7	80.6	6.59	64.20	62.56	63.07
300	1761.8	3572.7	98.2	11.91	113.97	108.88	111.08
400	2386.9	4834.4	121.2	19.38	167.74	160.96	166.12
500	3042.9	6143.3	115.0	24.11	226.73	219.80	223.91

TABLE 5

Average of problems $P1_n$ *to* $P10_n$, *algorithm* TRSPHi: *number of faces tested for local optimality.*

n	LOP1	LTP1	NTF1	LOP2	LTP2	NTF2	LOP3	LTP3	NTF3
100	3,568	1,072	474	3,363	1,012	652	3,440	1,066	494
200	10,278	3,119	1,435	9,415	2,917	1,944	9,854	3,091	1,511
300	18,160	5,557	2,654	16,385	5,105	3,416	17,346	5,484	2,727
400	26,508	8,167	3,957	23,800	7,491	5,102	25,513	8,104	4,081
500	36,098	11,139	5,444	32,867	10,383	6,974	34,749	11,028	5,603

TABLE 6

Average of problems $P1_n$ *to* $P10_n$, *algorithm* TRSPHi: *complexity of number of faces, times, and number of tests for local optimality.*

n	$\frac{\text{NFAC1}}{n}$	$\frac{\text{NBFC1}}{n^{1/3}}$	$\frac{\text{TInit}}{n^{4/3}}$	$\frac{\text{TIM1}}{n^{4/3}}$	$\frac{\text{LOP1}}{n^{4/3}}$	$\frac{\text{LTP1}}{n^{4/3}}$	$\frac{\text{NTF1}}{n^{4/3}}$
100	10.4	13.4	.00565	.0489	7.69	2.31	1.02
200	11.5	13.8	.00564	.0549	8.79	2.67	1.23
300	11.9	14.7	.00593	.0567	9.04	2.77	1.32
400	12.1	16.4	.00657	.0569	8.99	2.77	1.34
500	12.3	14.5	.00608	.0571	9.10	2.81	1.37

TABLE 7

Problem $P11_n$, *algorithm* TRSPHi: *number of tetrahedrons and faces in Delaunay triangulation and* CPU *times.*

n	NTET1	NFAC1	NBFC1	TInit	TIM1	TIM2	TIM3
100	288	656	160	4.08	12.87	14.18	13.57
200	672	1,488	288	14.90	44.53	50.87	48.23
300	1,080	2,362	404	31.70	97.22	111.03	105.08
400	1,512	3,278	508	53.58	171.73	192.62	182.32
500	1,944	4,194	612	81.52	270.32	296.53	277.92

TABLE 8

Problem $P11_n$, *algorithm* TRSPHi: *number of faces tested for local optimality.*

n	LOP1	LTP1	NTF1	LOP2	LTP2	NTF2	LOP3	LTP3	NTF3
100	1,834	474	188	2,149	472	255	1,956	468	211
200	6,051	1,680	643	7,305	1,676	1,064	6,852	1,689	806
300	13,311	3,684	1,500	15,845	3,672	2,633	14,880	3,680	1,863
400	23,870	6,312	3,032	27,417	6,260	4,986	25,653	6,194	3,539
500	38,005	9,796	5,210	42,044	9,620	8,049	39,013	9,370	5,859

TABLE 9
Problem $P11_n$, *algorithm* TRSPHi: *complexity of number of faces, times, and number of tests for local optimality.*

n	$\frac{\text{NFAC1}}{n}$	$\frac{\text{NBFC1}}{n^{0.8}}$	$\frac{\text{TInit}}{n^{1.8}}$	$\frac{\text{TIM1}}{n^2}$	$\frac{\text{LOP1}}{n^2}$	$\frac{\text{LTP1}}{n^2}$	$\frac{\text{NTF1}}{n^2}$
100	6.56	4.02	.00103	.00129	.183	.0474	.0188
200	7.44	4.16	.00108	.00111	.151	.0420	.0161
300	7.87	4.21	.00110	.00108	.148	.0409	.0167
400	8.20	4.21	.00111	.00107	.149	.0395	.0190
500	8.39	4.24	.00113	.00108	.152	.0392	.0208

contain measurements for $P11_n$. Tables 10, 11, and 12 contain measurements for $P12_n$. The CPU times in these tables are subject to a variation of up to about 5 percent when the program is run at different times.

For the random problems, it can be seen from Table 6 that the number of faces in the Delaunay triangulation is $O(n)$, and the time complexity of algorithm TRSPH1 is approximately $O(n^{4/3})$. From equation (1)(a), the number of tetrahedrons in the

TABLE 10
Problem $P12_n$, *algorithm* TRSPHi: *number of tetrahedrons and faces in Delaunay triangulation and* CPU *times.*

n	NTET1	NFAC1	NBFC1	TInit	TIM1	TIM2	TIM3
50	600	1,225	50	1.25	7.77	8.02	8.38
100	2,450	4,950	100	5.18	31.37	34.62	35.32
150	5,550	11,175	150	11.55	73.30	79.13	73.38
200	9,900	19,900	200	19.12	116.47	129.95	131.68
250	15,500	31,125	250	30.57	184.45	207.70	208.45

TABLE 11
Problem $P12_n$, *algorithm* TRSPHi: *number of faces tested for local optimality.*

n	LOP1	LTP1	NTF1	LOP2	LTP2	NTF2	LOP3	LTP3	NTF3
50	1,385	274	0	1,608	274	0	1,719	274	0
100	5,971	1,199	0	6,959	1,199	0	7,244	1,199	0
150	13,537	2,700	0	16,067	2,700	0	16,421	2,700	0
200	24,151	4,800	0	28,810	4,800	0	29,297	4,800	0
250	38,185	7,624	0	45,521	7,624	0	46,119	7,624	0

TABLE 12
Problem $P12_n$, *algorithm* TRSPHi: *complexity of number of faces, times, and number of tests for local optimality.*

n	$\frac{\text{NFAC1}}{n^2}$	$\frac{\text{TInit}}{n^2}$	$\frac{\text{TIM1}}{n^2}$	$\frac{\text{LOP1}}{n^2}$	$\frac{\text{LTP1}}{n^2}$
50	.490	.000500	.00311	.554	.110
100	.495	.000518	.00314	.597	.120
150	.497	.000513	.00326	.602	.120
200	.498	.000478	.00291	.604	.120
250	.498	.000489	.00295	.611	.122

Delaunay triangulation has the same complexity as the number of faces. For problem $P11_n$, it can be seen from Table 9 that the number of faces in the Delaunay triangulation is $O(n)$, as expected, and the time complexity of TRSPH1 is approximately $O(n^2)$. We believe that the higher time complexity for this problem is due to the fact that the initial triangulation T_I has many long tetrahedrons with small solid angles and the Delaunay triangulation does not have any of these tetrahedrons. Note that for $P11_n$, the theoretical asymptotic complexities for NBFC1 and TInit are $O(n^{2/3})$ and $O(n^{5/3})$, respectively. For problem $P12_n$, it can be seen from Table 12 that the number of faces in the Delaunay triangulation is $O(n^2)$, as expected, and the time complexity of TRSPH1 is approximately $O(n^2)$.

For the three types of problems, the time complexities of algorithms TRSPH2 and TRSPH3 are the same as that for TRSPH1, as can be seen from calculations similar to those in Tables 6, 9, and 12. Therefore our experiment has shown that there is no advantage to using TRSPH2 or TRSPH3 over TRSPH1, since all three algorithms require approximately the same amount of CPU time and TRSPH2 and TRSPH3 have each failed to construct a Delaunay triangulation for at least one test problem.

Due to the amount of CPU time required, we chose to use a maximum value of $n=500$ in our experiment. This is large enough to determine the complexity trends given in Tables 6, 9, and 12. We have run TRSPH1 for a few problems with up to $n=5{,}000$ vertices, and have not found a counterexample to Conjecture 2. The complexities for these larger problems are similar to those given in the above tables. The CPU time required for $P1_{5{,}000}$ is approximately 73 minutes.

We now describe the experimental results from running the 60 test problems for algorithms TRMMSA1, TRMMSA2, and TRMMSA3. Table 13 contains the minimum solid angles in the triangulations of 30 of the 60 problems (to reduce the table size, we show the results for 20 of the more interesting random problems in which the entries have greater variance). MSAi is the minimum solid angle in the triangulation produced by TRMMSAi (MSA3 is not shown since MSA3 = MSA1 for all the problems). MSADel is the minimum solid angle in the Delaunay triangulation produced by TRSPH1, MSA3a is the minimum solid angle in the triangulation produced by a modified version of TRMMSA3 in which the starting triangulation is the Delaunay triangulation produced by TRSPH1 instead of the initial triangulation T_I, so MSA3a $\geqq$ MSADel is always satisfied.

The comparison of the MSA values for the 50 random problems is summarized as follows. MSA1 > MSA2 for 27 problems, MSA1 < MSA2 for 7 problems, and MSA1 = MSA2 for 16 problems. MSADel > MSA12 for 25 problems, MSADel < MSA12 for 10 problems, and MSADel = MSA12 for 15 problems, where MSA12 = max (MSA1, MSA2). MSA3a > MSADel for 9 of the 50 problems, and MSA12 > MSA3a for 8 of the 50 problems. Therefore none of the algorithms always produces a SA-globally optimal triangulation (we do not know whether a triangulation with the highest MSA value is SA-globally optimal, but we do know that the triangulations with smaller MSA values are not SA-globally optimal), and from Table 13, it can be seen that the MSA value for a SA-locally optimal triangulation can be much smaller than that for a SA-globally optimal triangulation. Since MSA3a $\geqq$ MSA12 for 52 of the 60 problems, it seems that the best approach to constructing an SA-locally optimal triangulation with a "good" MSA value is to improve the Delaunay triangulation by applying the local transformation procedure to nonlocally optimal faces (with respect to the max-min solid angle criterion).

In Tables 14, 15, and 16 we present the counts and times for algorithm TRMMSA1 (the results for TRMMSA2 and TRMMSA3 are similar). From comparison with Tables

TABLE 13
Minimum solid angles in triangulations.

Problem	MSA1	MSA2	MSADel	MSA3a
$P1_{100}$	.0001568	.0002963	.0012346	.0032361
$P4_{100}$	.0006088	.0001021	.0004123	.0004123
$P5_{100}$	.0001851	.0002813	.0000893	.0000893
$P8_{100}$	.0000115	.0000115	.0001843	.0003123
$P2_{200}$	.0002078	.0001334	.0001774	.0001774
$P4_{200}$	.0001728	.0001023	.0004117	.0004117
$P5_{200}$	.0001794	.0000120	.0000043	.0000043
$P8_{200}$	.0000136	.0000195	.0000414	.0000414
$P4_{300}$	.0000922	.0000071	.0000922	.0003906
$P6_{300}$	.0000913	.0000913	.0000515	.0001215
$P7_{300}$	.0000456	.0000036	.0000619	.0000824
$P8_{300}$	.0000048	.0000071	.0000338	.0000338
$P3_{400}$	.0000742	.0000029	.0001818	.0001818
$P5_{400}$	.0000193	.0000008	.0000205	.0000205
$P7_{400}$	.0000110	.0000096	.0000102	.0000153
$P10_{400}$	.0000008	.0000008	.0000507	.0001061
$P1_{500}$	.0000413	.0000152	.0000762	.0001843
$P4_{500}$	.0000655	.0000213	.0000299	.0000299
$P6_{500}$	.0000097	.0000419	.0001641	.0001641
$P9_{500}$	.0000223	.0000538	.0000858	.0000858
$P11_{100}$	.0050024	.0050024	.1837619	.2617994
$P11_{200}$	.0013148	.0013148	.1837619	.2617994
$P11_{300}$	.0005906	.0005906	.1837619	.2617994
$P11_{400}$	.0004443	.0004443	.1837619	.2617994
$P11_{500}$	.0003192	.0003192	.1837619	.2617994
$P12_{50}$	.0031036	.0031036	.0193708	.0193708
$P12_{100}$	.0003834	.0003834	.0098490	.0098940
$P12_{150}$	.0001134	.0001134	.0066005	.0066005
$P12_{200}$	.0000478	.0000478	.0049631	.0049631
$P12_{250}$	.0000245	.0000245	.0039765	.0039765

TABLE 14
Average of problems $P1_n$ *to* $P10_n$, *algorithm* TRMMSA1: *number of tetrahedrons, faces, tests for transformability, and* CPU *times.*

n	NTET1	NFAC1	TIM1	LOP1	LTP1	NTF1
100	460.2	951.6	12.94	530	207	866
200	1005.0	2050.3	27.84	1,159	406	1,800
300	1571.8	3192.7	45.74	1,836	670	2,920
400	2145.4	4351.4	65.63	2,481	907	4,005
500	2701.9	5461.3	82.08	3,158	1,119	5,011

4 to 12, it can be seen that for TRMMSA1, a smaller percentage of the tests for transformability (local optimality) result in an application of the local transformation procedure and much larger percentage result in a nontransformable face. (This relatively large number of nontransformable faces explains why MSAi can sometimes be much smaller than MSADel.) Also, for TRMMSA1, there are much fewer tests for transformability in the random problems and problem $P11_n$ than there are for TRSPHi, and about the same number of tests for $P12_n$. Hence, the CPU time required for

TABLE 15
Problem $P11_n$, *algorithm* TRMMSA1: *number of tetrahedrons, faces, tests for transformability, and* CPU *times.*

n	NTET1	NFAC1	TIM1	LOP1	LTP1	NTF1
100	267	614	20.70	707	244	554
200	672	1,488	37.33	1,125	319	804
300	1,080	2,362	69.43	1,929	555	1,308
400	1,512	3,278	101.87	2,599	611	1,565
500	1,938	4,182	154.95	3,854	845	2,215

TABLE 16
Problem $P12_n$, *algorithm* TRMMSA1: *number of tetrahedrons, faces, tests for transformability, and* CPU *times.*

n	NTET1	NFAC1	TIM1	LOP1	LTP1	NTF1
50	532	1,089	20.47	761	199	648
100	2,122	4,294	86.60	3,079	857	2,755
150	4,760	9,595	188.80	6,738	1,891	6,379
200	8,457	17,014	335.58	11,810	3,324	11,522
250	13,201	26,527	528.35	18,644	5,285	18,036

TRMMSA1 is smaller than that for TRSPHi for the random problems and $P11_n$, and is greater for $P12_n$ since a test involving computation of solid angles takes more time than a test for determining whether a point is in a sphere. From calculations similar to Tables 6, 9, and 12, the number of faces in the triangulation produced by TRMMSA1 is $O(n)$ for the random problems, and the time complexity of TRMMSA1 appears to be approximately $O(n^{1.1})$ for the random problems, $O(n^{1.5})$ for $P11_n$, and $O(n^2)$ for $P12_n$.

Finally, to compare the CPU times for the three different types of problems and the two different types of local optimality criteria, the graphs of n versus TIM1 are given in Fig. 7.

8. Concluding remarks. We have presented an algorithm called TRSPH1 for constructing a triangulation of a set of n three-dimensional points that is pseudo-locally optimal with respect to the sphere criterion. Experimental results show that TRSPH1 always constructs a Delaunay triangulation (so far), although variations of TRSPH1 can sometimes fail to construct a Delaunay triangulation. The Delaunay triangulation of n three-dimensional random points (from the uniform distribution) is shown experimentally to contain $O(n)$ tetrahedrons and faces, and the empirical time complexity of TRSPH1 is $O(n^{4/3})$ for sets of random points, which compares well with existing algorithms for constructing a three-dimensional Delaunay triangulation (Bowyer [2], Watson [14], Avis and Bhattacharya [1]). The Delaunay triangulation of n three-dimensional points contains $O(n^2)$ tetrahedrons and faces in the worst case, and we have presented two families of problems for which TRSPH1 requires an empirical time complexity of $O(n^2)$. We have not yet found any problems for which TRSPH1 requires more than $O(n^2)$ time, so $O(n^2)$ appears to be the worst-case time complexity. This is better than the worst-case time complexity of $O(n^3)$ for Avis and Bhattacharya's algorithm (Bowyer and Watson do not discuss the worst case in their papers).

An open problem is to prove that TRSPH1 always constructs a Delaunay triangulation (Conjecture 2), or to find an example for which TRSPH1 fails to construct a

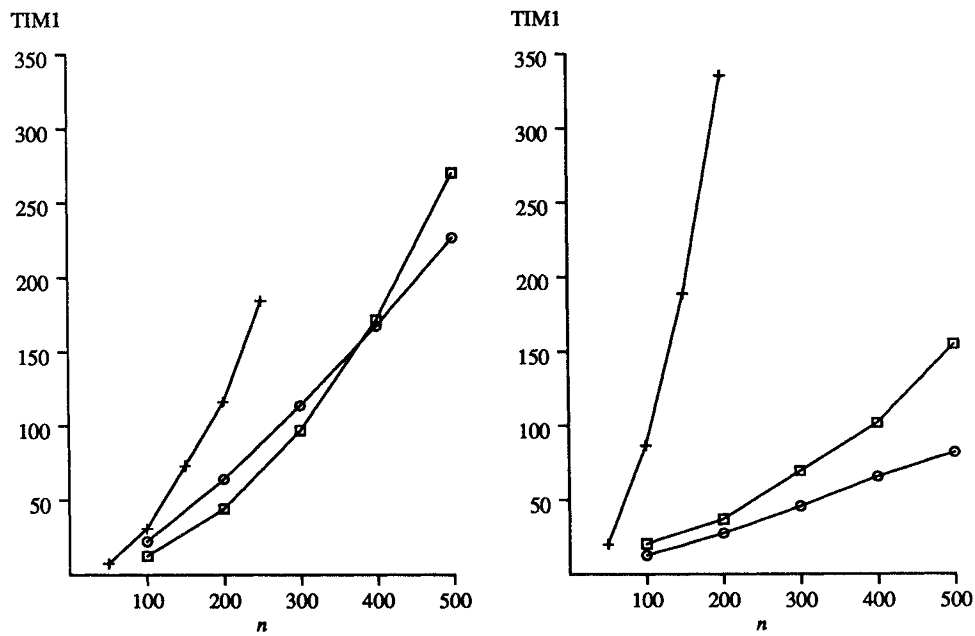

FIG. 7. *Graphs of n versus* TIM1 (*in seconds*). *Left graph is for* TRSPH1; *right graph is for* TRMMSA1. ○: *average of* $P1_n$ *to* $P10_n$; □: $P11_n$; +: $P12_n$.

Delaunay triangulation. If the latter case occurs, then open problems are to determine sufficient conditions for TRSPH1 to be successful and to determine whether the local transformation procedure can be used in a modified algorithm that always produces a Delaunay triangulation (this may involve applying the local transformation procedure to locally optimal faces, and may be related to Conjecture 1).

We believe that the approach of using the local transformation procedure to improve a triangulation, as in the TRSPHi algorithms, is especially useful if an initial triangulation that is nearly Delaunay can be constructed quickly, say in linear time. This may be possible in an application such as finite-element mesh generation in which the vertices as well as the tetrahedrons are generated. Information about the location of the generated vertices can be used to construct a "good" initial triangulation, and then it may be possible to improve this triangulation to a Delaunay triangulation in linear time. This is done in two dimensions in Joe [7] and is a subject of further research in three dimensions.

We have also introduced the max-min solid angle criterion in this paper. This criterion does not seem to have been used before, although Nguyen [11] tries to avoid small solid angles in his three-dimensional triangulation algorithm. Experimental results show that, unlike the case of the sphere criterion, a SA-locally optimal triangulation may be far from being SA-globally optimal due to many nontransformable faces. An approach to constructing a SA-locally optimal triangulation with a "satisfactory" minimum solid angle is to improve the Delaunay triangulation by applying the local transformation procedure to nonlocally optimal faces. A further research problem is to derive more theoretical results for SA-globally optimal triangulations such as determining whether the local transformation procedure can be used in their construction or whether the problem of constructing a SA-globally optimal triangulation is NP-hard.

Finally, we discuss the possibility of extending the local transformation approach to triangulations of higher dimensions. Lawson [10] has recently proved that an arbitrary dimensional version of Theorem 1 is true, shown that a set of $k+2$ points in k-dimensional space may be triangulated in at most two different ways, and characterized the different configurations of $k+2$ points from the point of view of their possible triangulations (the number of configurations increases as k increases). Hence the local transformation procedure can be defined in any dimension, but the number of different cases increases as the dimension k increases. However, we suspect that the use of the local transformation procedure to construct k-dimensional Delaunay triangulations for $k \geqq 4$ will be more difficult than the three-dimensional case (and maybe even not possible), since the configuration containing a facet may have four or more simplices so more facets are likely to be nonlocally optimal and nontransformable.

Acknowledgment. The author would like to thank the referees for their helpful comments.

Note added in proof. The author has now proved that Conjecture 2 is true.

REFERENCES

[1] D. Avis and B. K. Bhattacharya, *Algorithms for computing d-dimensional Voronoi diagrams and their duals*, in Advances in Computing Research 1, F. P. Preparata, ed., JAI Press, 1983, pp. 159–180.

[2] A. Bowyer, *Computing Dirichlet tessellations*, Comput. J., 24 (1981), pp. 162–166.

[3] J. C. Cavendish, D. A. Field, and W. H. Frey, *An approach to automatic three-dimensional finite element mesh generation*, Internat. J. Numer. Methods Engrg., 21 (1985), pp. 329–347.

[4] D. A. Field, *Implementing Watson's algorithm in three dimensions*, Proc. 2nd ACM Symposium on Computational Geometry, 1986, pp. 246–259.

[5] H. Fuhring, *The application of node-element rules for forecasting problems in the generation of finite element meshes*, Internat. J. Numer. Methods Engrg., 9 (1975), pp. 617–629.

[6] P. C. Gasson, *Geometry of Spatial Forms*, Ellis Horwood, Chichester, 1983.

[7] B. Joe, *Delaunay triangular meshes in convex polygons*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 514–539.

[8] C. L. Lawson, *Transforming triangulations*, Discrete Math., 3 (1972), pp. 365–372.

[9] ———, *Software for C^1 surface interpolation*, in Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, pp. 161–194.

[10] ———, *Properties of n-dimensional triangulations*, Computer Aided Geom. Des., 3 (1986), pp. 231–246.

[11] V. P. Nguyen, *Automatic mesh generation with tetrahedron elements*, Internat. J. Numer. Methods Engrg., 18 (1982), pp. 273–289.

[12] C. S. Petersen, B. R. Piper, and A. J. Worsey, *Adaptive contouring of a trivariate interpolant*, in Geometric Modeling: Algorithms and New Trends, G. E. Farin, ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987, pp. 385–395.

[13] T. A. Standish, *Data Structure Techniques*, Addison-Wesley, Reading, MA, 1980.

[14] D. F. Watson, *Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes*, Comput. J., 24 (1981), pp. 167–172.