Optimality of the Delaunay Triangulation in $\mathbb{R}^d$

V T Rajan
Manufacturing Research Department, IBM Research Division,
T J Watson Research Center, NY 10598, USA.

Abstract

In this paper we present an optimality result for the Delaunay triangulation of a set of points in $\mathbb{R}^d$. We also show that some of the well known properties of the Delaunay triangulation in $\mathbb{R}^2$ can, when appropriately defined, be generalized to the Delaunay triangulation in $\mathbb{R}^d$. In particular, we show that (a) the maximum min-containment radius (the radius of the smallest sphere containing the simplex) of the Delaunay triangulation of a point set in $\mathbb{R}^d$ is less than the maximum min-containment radius of any other triangulation of the point set, (b) if a valid triangulation consists of only self-centered triangles (a simplex whose circumcenter falls inside the simplex) then it is the Delaunay triangulation, and (c) there exists an incremental flip algorithm (one that modifies the triangulation locally to make it Delaunay) that can generate the Delaunay triangulation for any point set. We further show that the Delaunay triangulation can be seen as the optimum solution to a continuum optimization problem.

1 Introduction

The *Delaunay triangulation* of a set of points in $\mathbb{R}^d$ is defined to be the triangulation[1] such that the circumcircle of every triangle[2] in the triangulation contains no point from the set in its interior. Such a unique triangulation exists for every point set in $\mathbb{R}^d$, and it is the dual of the Voronoi diagram [3].

In $\mathbb{R}^2$ it has been studied extensively and large number of its properties are known [1, 3, 11]. (a) Among all the valid[3] triangulations of a set of points in $\mathbb{R}^2$, the Delaunay triangulation lexicographically maximizes the minimum angle, and also minimizes the maximum circumradii. (b) If every triangle in a triangulation is non-obtuse then it is the Delaunay triangulation. (c) There exists a flip algorithm [9] which looks at the four vertices of two adjacent triangles and modifies the triangulation to ensure that it is locally Delaunay. This algorithm transforms any triangulation to the Delaunay triangulation in $O(n^2)$ time and can be used as an incremental algorithm. (d) Optimal $O(nlogn)$ time divide and conquer and plane sweep algorithms are known and elegant data structures to support their implementation exist [7, 8, 11].

In three and higher dimensions, very few results are known [3]. There exists a "lifting" transformation (discussed below) that allows the Delaunay triangulation problem in $\mathbb{R}^d$ to be transformed into a convex hull problem in $\mathbb{R}^{d+1}$. The convex hull algorithms can therefore be used to obtain the Delaunay triangulation. But no min-max result, such as the ones discussed above, was known.

The Delaunay triangulation, (and its dual Voronoi diagram) has been used extensively in both design of efficient algorithms and in practical applications [10, 3]. Since, the Delaunay triangulation has some optimal properties in $\mathbb{R}^2$, and efficient global and incremental algorithms exist to construct them, they have been used in finite element mesh generation as a way of yielding "good" meshes [13]. A "good" mesh is loosely defined as the one whose elements are of uniform size and shape. We have used them in $\mathbb{R}^3$ for the same purpose though no such properties were known.

In this paper we present a new optimality result for Delaunay triangulation of a set of points in $\mathbb{R}^d$. We also show that some of the well known properties of the Delaunay triangulation in $\mathbb{R}^2$ mentioned above can, when appropriately defined, be general-

[1] Simplicial decomposition of the convex hull of the point set

[2] d-dimensional simplex (d-simplex), which is defined by its $(d+1)$ vertices.

[3] the triangles do not overlap

ized to the Delaunay triangulation in $\mathbb{R}^d$. We define a triangle to be **self-centered** if the circumcenter of the triangle lies inside or on its boundary. In $\mathbb{R}^2$ all non-obtuse triangles are self-centered and vice-versa and thus it is a generalization to $\mathbb{R}^d$ of the non-obtuse triangle. We define the **Min-Containment Sphere** of a triangle to be smallest sphere containing the triangle. We will show that for a self-centered triangle min-containment sphere is the same as the circumsphere. But for a non-self-centered triangle it is the circumsphere of one of its facets[4], its center lies on the boundary, and its radius is less than the circumradius. (See section 2.) In $\mathbb{R}^2$ the min-containment circle of an obtuse triangle is the circle with the longest edge as the diameter. (See figure 1).

In section 3 and 5 we show the following results which correspond to the properties in $\mathbb{R}^2$ mentioned above. (a) The maximum min-containment radius of the Delaunay triangulation of a point set in $\mathbb{R}^d$ is less than the maximum min-containment radius of any other triangulation of the point set. (b) If a valid triangulation consists of only self-centered triangles then it is the Delaunay triangulation. (c) In section 4, we define a **generalized flip**, which deletes $(d+2-k)$ non-Delaunay triangles (of the $d+2$ triangles) defined by $d+2$ vertices in $\mathbb{R}^d$ and replaces them with k Delaunay ones. In section 5 we show that when a new point is inserted into a Delaunay triangulation there is a way to organize the flips to ensure that the triangulation is always valid. This leads to an incremental algorithm for the Delaunay triangulation that takes $O(n^{\lceil (d+1)/2 \rceil})$ flips. Further in section 3 we show that the Delaunay diagram is the optimal solution to a (linear) optimization problem and use this prove some of the above results.

2 An Optimization Problem over a simplex

Given $P_1, P_2, ..., P_{d+1}$ be $(d+1)$ points[5] in $\mathbb{R}^d$ that define a triangle T, consider the function $F(X)$ defined at every point X in the space:

$$\sum_{i=1}^{d+1} \lambda_i = 1 \qquad \sum_{i=1}^{d+1} \lambda_i P_i = X \qquad (1)$$

[4] A sub-simplex of the triangle. Its circumsphere is the smallest sphere passing through its vertices

[5] Throughout this paper we will assume that, unless otherwise stated, the points are in general positions. Hence, for example, no $(d+2)$ points are co-spherical

$$F(X) = \sum_{i=1}^{d+1} \lambda_i (P_i - X)^2 = \sum_{i=1}^{d+1} \lambda_i P_i^2 - X^2. \qquad (2)$$

The $(d+1)$ weights λ_i (also called the bary-centric coordinates of X) are uniquely determined by the equations (1). The equation (2) defines $F(X)$ to be the weighted average of the square of the distance to each of the vertices of the triangle. P_i^2 denotes the square of the norm of the point. For a point X inside the triangle, all of the bary-centric coordinates λ_i are positive and hence $F(X)$ is also positive. At a vertex P_i, $\lambda_i = 1$ and all other coordinates are zero, hence $F(P_i) = 0$. The following lemma give the F(X) at any point in space.

Lemma 1. The $Max_X F(X)$ occurs at the circumcenter (X_C) of the triangle, the value of $F(X)$ at that point is equal to the square of the circumradius (R^2) and the value of the function at any point X in space is given by

$$\begin{aligned} F(X) &= R^2 - (X - X_C)^2 \\ &= (R^2 - {X_C}^2) + 2X_C.X - X^2. \qquad (3) \end{aligned}$$

Proof: We can find the maximum of function $F(X)$ (equation (2)) subject to the constraints (equations (1)) using Lagrange multipliers. After some algebra we get that the maximum occurs at the point X_C that satisfies the relationship that $(P_i - X_C)^2$ is independent of i. Thus X_C is the circumcenter of the triangle and $F(X_C) = R^2$. The function $F(X)$ for an arbitrary point is given by:

$$\begin{aligned} F(X) &= \sum_{i=1}^{d+1} \lambda_i ((P_i - X_C) - (X - X_C))^2 \\ &= R^2 - 2(X - X_C)^2 + (X - X_C)^2. \end{aligned}$$

Hence the result (3).□

We can give several geometric interpretations for $F(X)$. The *Power* of a point with respect to a sphere is the square of the length of the tangent from the point to the sphere. For points inside the sphere it is negative and is minus the square of half the length of the chord with the point as the midpoint. From equation (3) we see that $F(X)$ is minus the power of the point X with respect to the circumsphere of the triangle. Hence we will take the liberty of calling $F(X)$ the **power function** of the triangle. Another interpretation is given by the lifting transformation considered by Edelsbrunner and Seidel [4]. Consider the transformation $\alpha(P) \mapsto (P, P^2)$ which lifts the points in $\mathbb{R}^d$ onto the paraboloid $z = P^2$ in $\mathbb{R}^{d+1}$. The triangle in $\mathbb{R}^d$ is lifted to a triangle lying on a plane in $\mathbb{R}^{d+1}$. $F(X)$ is the verticle distance of the plane above the paraboloid at the point X.

We can show that the circumcenter of the triangle lies on the normal to each face[6] of the triangle passing through the face's circumcenter. For a non-self-centered triangle one or more of the values of the weights λ_i at the circumcenter is negative.

Lemma 2. The constrained $Max_X F(X)$, with the added constraint that X lies in (or on) the triangle T (i.e. $\lambda_i \geq 0$), occurs at the center of the min-containment sphere of the triangle and the maximum value is the square of its radius (r^2).

Proof: For a self-centered triangle, the circumsphere is the smallest sphere spanning the triangle. Because, if we move the center of the sphere away from any of the vertices, the radius of the minimum spanning sphere will increase. For a non-self-centered triangle, the constrained optimum is the point where the smallest sphere centered at the circumcenter touches the simplex. Let this sphere touch the facet f of the simplex at point X_c. The sphere centered at X_c with radius square $F(X_c)$ is the circumsphere of f. It is the smallest sphere spanning f. It also contains the remaining vertices of the simplex in its interior. Hence the result. (See figure 1.) Also notice that the constrained optimization problem is a quadratic programming problem whose dual is the minimum spanning sphere problem [10].□

We will use the following result in the next section.

Lemma 3. Let $T_1 = [P_1, ..., P_d, P_{d+1}]$ and $T_2 = [P_1, ..., P_d, P_{d+2}]$ be two self-centered triangles that lie on the opposite sides of their common face $F = [P_1, ..., P_d]$. Then the circumsphere of one does not contain the opposing vertex of the other.

Proof: Let X be a point lying on the normal to the face F and on the opposite side to P_{d+1}. If the sphere centered at a point X passing through the vertices of F contains P_{d+1} then T_1 is non-self-centered. Choose X to be circumcenter of T_2 and a proof by contradiction follows. □

3 An Optimization Problem over a set of points

Let $Sp = \{P_1, P_2, ..., P_n\}$ be a set of n points in $\mathbb{R}^d$ then consider the function $f(X)$ defined at every point X in the convex hull of Sp ($CHSp$):

$$\lambda_i \geq 0, \qquad \sum_{i=1}^{n} \lambda_i = 1, \qquad \sum_{i=1}^{n} \lambda_i P_i = X \tag{4}$$

$$F(X, \lambda) = \sum_{i=1}^{n} \lambda_i (P_i - X)^2 = \sum_{i=1}^{n} \lambda_i P_i^2 - X^2. \tag{5}$$

$$f(X) = Min_\lambda F(X, \lambda) \tag{6}$$

$F(X,\lambda)$ is the weighted average of the distance square to the points, except, now up to $(n - (d+1))$ weights λ_i can be be varied for a fixed point X. $f(X)$ is the minimum for a fixed X over this choice of weights. One (not necessarily optimal) choice of weights would be to give non-zero weights to only some set of $(d+1)$ points (whose convex hull contains X, cf. equation (4)) and set the remaining $(n-(d+1))$ weights to zero. In this case the function $F(X,\lambda) = F(X)$ where $F(X)$ is the power function of the triangle defined by the chosen $(d+1)$ points.

Theorem 1. At $Min_\lambda F(X, \lambda)$ for a fixed point X the only non-zero values of λ_i occur for the vertices of the Delaunay triangle containing the point X. Thus $f(X)$ is given by Lemma 1,

$$\begin{aligned} f(X) &= f_t(X) = R_t{}^2 - (X - X_{Ct})^2 \\ &= (R_t{}^2 - X_{Ct}{}^2) + 2X_{Ct}.X - X^2. \end{aligned} \tag{7}$$

where X_{Ct} and R_t are the circumcenter and circumradius of the Delaunay triangle containing X. (t denotes the label of this triangle.)

Proof: To prove this result we use following well known result [2, 4]. Consider the transformation $\alpha(P) \mapsto (P, P^2)$ which lifts the points in $\mathbb{R}^d$ onto the paraboloid $z = P^2$ in $\mathbb{R}^{d+1}$. The point Sp is lifted to a set of points $SP = \{(P_1, P_1{}^2), (P_2, P_2{}^2), ..., (P_n, P_n{}^2)\}$ in $\mathbb{R}^{d+1}$. Take the lower part of the convex hull of SP. Project this back into $\mathbb{R}^d$ and we get the Delaunay triangulation ($\mathcal{D}$) of Sp.

Now consider a point $(\sum_{i=1}^{n} \lambda_i P_i, \sum_{i=1}^{n} \lambda_i P_i^2) = (X, F(X,\lambda) + X^2)$, subject to the conditions in the equation (4). This is a general point inside the convex hull of SP. The minimum of F for a fixed X is given by the point with the lowest z coordinate, hence the point on the lower convex hull of SP. This triangle (d-facet in $\mathbb{R}^{d+1}$) is the Delaunay triangle containing X. □

Corollary Among all the triangles with vertices in Sp and containing the point X, the Delaunay triangle

[6] (d-1) dimensional facet of the triangle

minimizes the power function $F(X)$ of the triangle at the point (or maximizes the power of the point with respect to the circumsphere of the triangle).

Theorem 2. The maximum min-containment radius of the Delaunay triangulation is less than the maximum min-containment radius of any other triangulation of the point set.

Proof: Given any triangulation $\mathcal{T}$ of Sp we can define the function $F_{\mathcal{T}}(X)$ at each point X in $CHSp$ as power the function $F(X)$ of the point X with respect to the triangle $T \in \mathcal{T}$ that contains the point X. Let $F_{\mathcal{D}}(X)$ define the corresponding function for the Delaunay triangulation $\mathcal{D}$. Let X_T and X_D respectively represent the points in $CHSp$ where $F_{\mathcal{T}}(X)$ and $F_{\mathcal{D}}(X)$ attain their respective maxima. Let R_T and R_D be respectively the maximum min-containment radii of the triangulations $\mathcal{T}$ and $\mathcal{D}$ respectively. From the results of the previous section the square of the min-containment radius of a triangle T is the maximum of the power function over the triangle and hence its maxima for the triangulation $\mathcal{T}$ is equal to maxima of $F_{\mathcal{T}}(X)$. The same is true for the Delaunay triangulation. Hence:

$$\begin{aligned} {R_T}^2 &= F_{\mathcal{T}}(X_T) \geq F_{\mathcal{T}}(X_D) \\ &\geq F_{\mathcal{D}}(X_D) = {R_D}^2. \end{aligned} \tag{8}$$

Hence the result.□

While theorem 2 is true in all dimensions, the lexicographical version of the theorem is not true even in $\mathbb{R}^2$. A counter example to this was given by H. Edelsbrunner. (See Figure 2.) Also, notice that the triangulation $\mathcal{T}$ does not have to be valid for the Theorem 2 to hold. As long as the triangulation spans the $CHSp$, the theorem holds. If the triangulation is not valid then there may be more than one triangle covering the point X and the function $F_{\mathcal{T}}(X)$ will be multiple valued at those points. The inequality (8) still holds and hence the theorem.

Theorem 3. If a valid triangulation consists of only self-centered triangles, then it is the Delaunay triangulation of that point set.

Proof: Suppose we are given a valid triangulation of Sp consisting of only self-centered triangles. Then from Lemma 3, every pair of adjacent triangles satisfy the Delaunay condition. If we project the triangles using $\alpha(P)$ defined above, then this states that the resulting surface is convex locally. Since a surface that is locally convex everywhere is globally convex it follows that we have generated the convex hull. □

In the next section we will discuss how the local modification technique can be used to provide an incremental algorithm to generate the Delaunay Triangulation.

4 The flip procedure

Consider now the optimization problem discussed above for a set $sp = \{P_1, P_2, ..., P_{d+2}\}$ of $(d+2)$ points. Let $T_1 = [P_1, ..., P_d, P_{d+1}]$ and $T_2 = [P_1, ..., P_d, P_{d+2}]$ be two triangles that share a common face $F = [P_1, ..., P_d]$. If the circumsphere of T_1 does not contain the vertex P_{d+2} then the triangles T_1 and T_2 and their facets are Delaunay-valid with respect to the point set sp. We say that a facet is Delaunay-valid with respect to a point set if there exists a sphere that passes through the vertices of the facet and does not contain any of the points in the set in its interior. If a k-facet[7] is Delaunay-valid, then all its subfacets are Delaunay-valid.

Now consider the case when the circumsphere of T_1 contains the vertex P_{d+2}. Then T_1 is not a valid Delaunay triangle and the power function $F(X)$ is not the minimal one for any point X in T_1. We can minimize the function and get $f(X)$ by giving more weight to P_{d+2}. We solve this one variable linear programming problem as follows.

Let $P_{d+2} = \sum_{i=1}^{d+1} \beta_i P_i$, $(\sum_{i=1}^{d+1} \beta_i = 1)$, then for any point X inside T_1 we have

$$\begin{aligned} X &= \sum_{i=1}^{d+1} \lambda_i P_i = \sum_{i=1}^{d+1} \lambda_i P_i + \mu(P_{d+2} - \sum_{i=1}^{d+1} \beta_i P_i) \\ &= \sum_{i=1}^{d+1} (\lambda_i - \mu\beta_i) P_i + \mu P_{d+2} \end{aligned} \tag{9}$$

The minimum possible value of μ is 0 and we know that it does not correspond to the optimum. The maximum value of μ is determined by the requirement that all the weights be non-negative. Thus $Max\mu = Min_i\mu_i = Min_i(\lambda_i/\beta_i)$ where only positive values of β_i are considered. If this minimum is achieved for point number j then the optimum and hence the Delaunay triangle spanning the point X is given by replacing P_j with P_{d+2} in T_1. Hence if there are k positive values of β then by looking at different point X in T_1 we can see that there are k valid triangles in the convex hull of the $(d+2)$ points. Similarly we can show that there are $(d+2-k)$ invalid

[7] k dimensional facet of the triangle, it is a k-simplex

triangles in the the convex hull of the $(d+2)$ points We can also show that the (k-1)-facet described by the corresponding k points is invalid, and all higher dimensional facets, and simplices incident on it are also invalid. The flip procedure consists of replacing the invalid simplices with the valid simplices. It has the effect of minimizing the function $F(X)$ with respect to the $(d+2)$ points. In the lifted space the image $(d+2)$ points form a simplex, and the flip procedure replaces the top triangles of the simplex with the bottom triangles.

We would like to develop an algorithm for generating the Delaunay triangulation from any triangulation using this procedure. Unfortunately, for $k < d$, this requires that more than two triangles be deleted. Since some of these triangles may not exist, the flip can lead to an invalid triangulation. The optimality results mentioned in the last section is still valid. But we need a way of organizing the information so that the end result is valid.

In the next section we show that if a triangulation is the Delaunay then it has enough structure so that when a new point is added to the set, we can find a sequence of valid flips which allow us to go from the existing triangulation to the new Delaunay triangulation.

5 An Incremental Algorithm

The "lifting" transformation, discussed in section 3, maps the points in the set Sp to the surface of the paraboloid $z = X^2$. Under the transformation the Delaunay triangulation mapped to a convex piecewise linear surface. We can use the results of Theorem 1 and Lemma 1 to obtain a functional form for this surface.

Lemma 4. The equation for the convex surface is given by $z = f_t(X) + X^2 = 2X_{Ct}X + ({R_t}^2 - {X_{Ct}}^2) = g(X)$ where R_t and X_{Ct} are the circumradius and the circumcenter of the Delaunay triangle containing the point X. (t denotes the label of this triangle.)

The convexity imposes an order structure on the surface [5]. This is of value in the incremental algorithm. Start from any point X_0 on the surface and shoot a ray downward in the $-z$ direction. Extend the hyperplane of each of triangle, then, the hyperplanes will strike the ray at point $(X_0, g_t(X_0))$ where $g_t(X_0) = ({R_t}^2 - {X_{Ct}}^2) + 2X_{Ct}.X_0$. These points can be sorted starting at the surface. If we start at the surface and move downward along the ray, then the triangles will become visible in the order that we meet the corresponding point along the ray. By saying a point is *visible* from another point we mean that a line joining the two points do not intersect the surface $g(X)$. The fact that $g(X)$ represents a convex surface is allows us to use the following Lemma.

Lemma 5. If from a point on the ray defined above a point P on the surface is visible, then all the points on the curve, that is formed by projecting the straight line connecting X_0 to P on the surface, are also visible.

Proof: Take a section of the surface using the 2-d plane defined by the ray and the point P. The theorem is self-evident in this plane. □

Now let us consider what effect inserting a new point at X_0 has on the surface (that is, the Delaunay triangulation). Since the surface touches the paraboloid at each of the vertices, this is equivalent to moving the z coordinate of the point X_0 from $g_O(X_0)$ (the old surface point), to ${X_0}^2$ and then retriangulating the surface to maintain its convexity. The first useful result to note is the following known Lemma [8].

Lemma 6. When a point is inserted into a Delaunay triangulation, then every new triangle or facet created during the modification to satisfy the Delaunay criteria, has the new point as one of its vertices.

Notice that the new point may render some of the old triangles and facets invalid. One way to delete these unwanted old triangles and create the new ones is to use the flip algorithm mentioned in the last section. But we need a way to organize the flips, so that they always yield a valid triangulation. This organization is provided by Lemma 5 and the discussion before it.

Instead of moving the z coordinate of the surface in one step from $g_O(X_0)$ to ${X_0}^2$, move it continuously. As the point is moved in the $-z$ direction triangles become visible. Each time a triangle becomes visible it needs to be flipped since the Delaunay criteria in $\mathbb{R}^d$ is equivalent to the convex hull condition in $\mathbb{R}^{d+1}$. Further, from Lemma 5, it follows that before any triangle becomes visible, all it faces, hence adjacent triangles also becomes visible. Thus we will always have a valid triangulation. We need to maintain a sorted (according to z intersection) list of triangles opposite to the new point, and every time we flip a triangle, we also have to enter all the new triangles that are opposite to the point. We can drop all the zintersections $< X_0^2$ from the list since they lie below the parabolic surface.

Algorithm Incremental Delaunay triangulation

- *Input* n points in $\mathbb{R}^d$.
- *Output* The Delaunay Triangulation of the n points.

1. *Initialization* Start with a suitably chosen large triangle that covers the domain.
2. *Insertion* Locate (if not known) the triangle containing the point and Insert the point there. That is replace the triangle with $(d+1)$ new triangles.
3. *In-circle test* Create a sorted list of the z intersection (with decreasing z) for each of the adjacent triangles. Drop the triangles with $z < X_0^2$.
4. *Iteration* While the list is non-empty, pop the first triangle and flip. Perform the In-circle test on each new triangle exposed and insert it into the list, if appropriate.
5. *Loop* Go back to the Insertion step until all points have been inserted.

Complexity Each flip represents the addition of a (d+1)-simplex in $\mathbb{R}^{d+1}$. Since n points in $\mathbb{R}^{d+1}$ can create $O(n^{\lceil (d+1)/2 \rceil})$ simplices, and this represents an upper bound on the number of flips. In addition we have to maintain a sorted list of the opposite triangles, and this leads to a time complexity of $O(n^{\lceil (d+1)/2 \rceil} logn)$. Further, if the point location is not known then that can take $O(n^{\lceil d/2 \rceil + 1})$ time.

The structure of the Delaunay triangulation which we exploited above is general to any triangulation in $\mathbb{R}^d$ that is obtained by orthogonal projection of the faces of a convex polytope in $\mathbb{R}^{d+1}$ [5]. Hence we can use the flip algorithm to go from any such triangulation to Delaunay triangulation. In addition, the algorithm can be used when we wish to move one or more vertices in a Delaunay triangulation and modify the triangulation to retain the Delaunay nature. This is particularly useful in Finite Element Mesh generation, where we often seek to move the vertices to improve the quality of the mesh.

6 Discussion

This paper provides an optimality result for Delaunay triangulation in $\mathbb{R}^d$. It suggests that the Delaunay triangulation is the most compact one in the sense of having the smallest maximum min-containment spheres. However, this optimality result is different from the previous ones known to be true in $\mathbb{R}^2$. Firstly, it is not lexicographically valid, secondly, the method used to prove it is different from the ones used to prove the optimality result in $\mathbb{R}^2$. Those results are proven using the flip algorithm [3]. This raises several interesting questions. Can this new method be used to prove other results? Are there optimality results that are lexicographically valid in $\mathbb{R}^d$? Can the flip algorithm be generalized to yield the Delaunay triangulation starting with any triangulation in $\mathbb{R}^d$? An $\mathbb{R}^3$ version of the incremental algorithm has been developed by Barry Joe[6].

7 Acknowledgements

I would like to thank Herbert Edelsbrunner whose talk inspired me to think about this problem, and L. Nackman whose enthusiasm encouraged me to write up the solution.

References

[1] Aurenhammer F., Voronoi diagrams - survey, Inst. for Inf. Proc., Graz Tech. Univ. report, (1988).

[2] Brown K.Q., Voronoi diagrams from convex hulls, *Inform. Process. Lett.*, 9(1970) 223-228.

[3] Edelsbrunner H., *Algorithms in Combinatorial Geometry* , Springer Verlag (1987).

[4] Edelsbrunner, H., and Seidel, R. Voronoi diagrams and arrangements. Discrete Comput. Geom. 1 (1986), 25-44.

[5] Edelsbrunner, H. An Acyclicity theorem for cell complexes. ACM Computational Geometry Conf. (1989)

[6] Edelsbrunner, H. Private communication.

[7] Fortune, S. A sweepline algorithm for Voronoi diagrams. Algorithmica, 2(2) (1987), 153-174.

[8] Guibas, L. and Stolfi J., Primitives form the manipulation of general subdivisions and the computation of Voronoi diagrams. ACM Trans. Graphics 4 (1985) 74-123.

[9] Lawson, C.L. Generation of a triangular grid with applications to contour plotting. Memo 299, Calif. Inst. Jet Propul. Lab. (1972).

[10] Preparata, F.P and Shamos, M.I, Computational Geometry - an Introduction. Springer Verlag, New York, (1985).

[11] Shamos, M.I., and Hoey, D. Closest-point problems. In Proc.16th Ann. IEEE Sympos. Found. Comput. Sci. (1975), 151-162. New Haven, Conn., (1978)

[12] Shamos, M.I. Computational Geometry. Ph.D. Thesis, Dept. Comput. Sci., Yale Univ., New Haven, Conn., (1978)

[13] Srinivasan V., Nackman L.R., Tang, J. and Meshkat S.N., Automatic Mesh Generation using the Symmetric Axis Transformation of Polygonal Domains, IBM RC 16132, (1990).

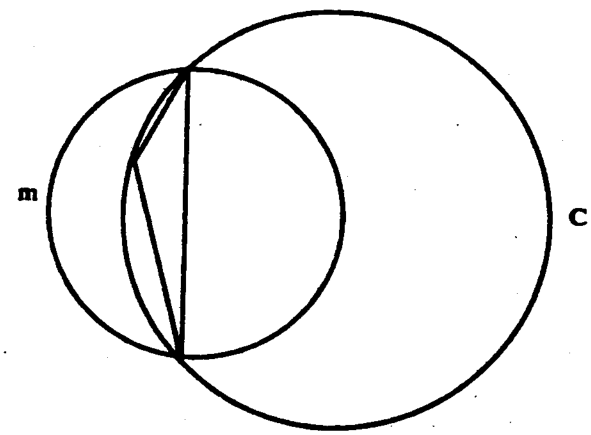

Figure 1. The min–containment circle m, and the Circum–circle M of a non–self–centered (obtuse) triangle in R^2

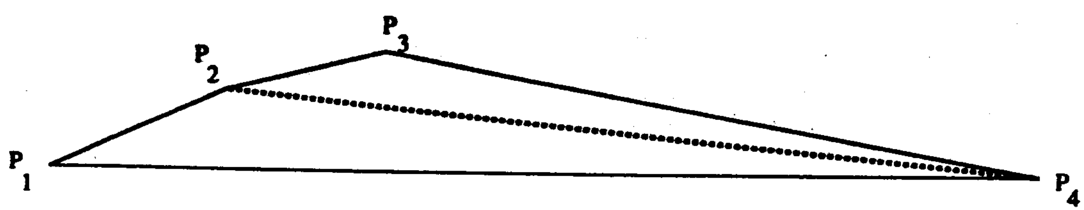

Figure 2. A Delaunay triangulation that does not lexicographically minimize the maximum min–containment radii.