

REGTET: A Program for Computing Regular Tetrahedralizations

Javier Bernal

National Institute of Standards and Technology, Gaithersburg MD 20899, USA,
javier.bernal@nist.gov,
WWW home page: <http://math.nist.gov/~JBernal>

Abstract. REGTET, a Fortran 77 program for computing a regular tetrahedralization for a finite set of weighted points in 3-dimensional space, is discussed. REGTET is based on an algorithm by Edelsbrunner and Shah for constructing regular tetrahedralizations with incremental topological flipping. At the start of the execution of REGTET a regular tetrahedralization for the vertices of an artificial cube that contains the weighted points is constructed. Throughout the execution the vertices of this cube are treated in the proper lexicographical manner so that the final tetrahedralization is correct.

1 Introduction

Let S be a finite set of points in 3-dimensional space (\mathcal{R}^3). By a *tetrahedralization* T for S we mean a finite collection of tetrahedra (3-dimensional triangles) with vertices in S , that satisfies the following two conditions.

1. Two distinct tetrahedra in T that are not disjoint, intersect at a common facet, a common edge, or a common vertex.
2. The union of the tetrahedra in T equals the convex hull of S .

For each point p in S let w_p be a real-valued weight assigned to p . Given p in S and a point x in \mathcal{R}^3 , the *power distance* of x from p , denoted by $\pi_p(x)$, is defined by

$$\pi_p(x) \equiv |xp|^2 - w_p,$$

where $|xp|$ is the Euclidean distance between x and p . Given a tetrahedron t with vertices in S , a point, denoted by $z(t)$, exists in \mathcal{R}^3 with the same power distance, denoted by $w(t)$, from all vertices of t . Point $z(t)$ is called the *orthogonal center* of t . Given a tetrahedralization T for S , we then say that T is a *regular tetrahedralization* for S if for each tetrahedron t in T and each point p in S , $\pi_p(z(t)) \geq w(t)$. We observe that T is unique if for each tetrahedron t in T and each point p in S that is not a vertex of t , $\pi_p(z(t)) > w(t)$. If T is unique then the *power diagram* of S [1] is the dual of T . Finally, we observe that if the weights of the points in S are all equal then the power diagram of S is identical to the *Voronoi diagram* of S [10], and the regular and *Delaurnay* [4] tetrahedralizations for S coincide.

In this paper we discuss REGTET, a Fortran 77 program for computing regular tetrahedralizations (or Delaunay tetrahedralizations in the absence of weights) with incremental topological flipping [6] and lexicographical manipulations [3]. A copy of program REGTET that includes instructions for its execution can be obtained from <http://math.nist.gov/~JBernal>

2 Incremental Topological Flipping

Let T be a tetrahedralization for S , let t be a tetrahedron in T , and let p be a point in S that is not a vertex of t . Denote the vertices of t by q_1, q_2, q_3, q_4 , and let T_1 and T_2 be the only two possible tetrahedralizations for $\{q_1, q_2, q_3, q_4, p\}$ [9]. Assume t is in T_1 , and T_1 is contained in T . A *topological flip* or simply a *flip* on T_1 is an operation that replaces T_1 with T_2 in T .

Program REGTET which is based on an algorithm by Edelsbrunner and Shah [6] constructs a regular tetrahedralization for the set S by adding the points in S one at a time into a regular tetrahedralization for the set of previously added points. A point is added by REGTET through a finite number of steps, each step involving a decision about whether a certain flip should take place and if so applying the flip. This technique is a generalization of a result for computing incrementally Delaunay triangulations in \mathcal{R}^2 [7]. By extending results for Delaunay triangulations and tetrahedralizations [8], [9], Edelsbrunner and Shah [6] justify their algorithm.

3 Lexicographical Manipulations

The incremental nature of Edelsbrunner and Shah's algorithm [6] implies that before any points in S are added a regular tetrahedralization must be first constructed by program REGTET with vertices close to infinity and underlying space equal to \mathcal{R}^3 . The vertices of this initial tetrahedralization are said to be *artificial*. Throughout the execution of the program artificial points must be treated in the proper lexicographical manner so that the final tetrahedralization does contain a tetrahedralization for S , and this tetrahedralization for S is indeed regular (since the coordinates of the artificial points can be extremely large in absolute value, it is inadvisable to identify them, thus the need to treat artificial points in a lexicographical manner).

Lexicographical manipulations that are employed in program REGTET are described and justified in [3]. At the start of the execution of the implementation a 3-dimensional cube with vertices close to infinity that contains S in its interior is identified, and a regular tetrahedralization for the set of vertices of the cube (weights set to the same number) is computed. The execution then proceeds with the incremental insertion of points in S as suggested by Edelsbrunner and Shah. However, at all times, because of the lexicographical manipulations employed in the presence of artificial points (the vertices of the cube), the artificial points are assumed to be as close to infinity as the manipulations require.

4 Flipping History

At all times during its execution, program REGTET maintains a list of all tetrahedra in the current and previous tetrahedralizations. This list is in the form of a directed acyclic graph that represents the history of the flips REGTET has performed [6], and it is used by REGTET for identifying a tetrahedron in the current tetrahedralization that contains a new point. Identifying a tetrahedron that contains a point this way is a generalization of a technique used in [7] for 2-dimensional triangulations.

5 Running Time

Program REGTET has the capability of adding the points in S in a random sequence. For some positive integer n , let n be number of points in S . Using an analysis similar to the one in [7] for 2-dimensional Delaunay triangulations, Edelsbrunner and Shah [6] show that if the points in S are added in a random sequence then the expected running time of their algorithm for computing a regular tetrahedralization for S is $O(n \log n + n^2)$. As pointed out in [6], the actual expected time could be much less, i. e. the second term (n^2) in the above expectation could be much less, depending on the distribution of the points in S . Accordingly this should be the case for sets of uniformly distributed points in a cube or a sphere. As proven for a cube in [2] and for a sphere in [5], the complexity of the Voronoi diagram, and therefore of the Delaunay tetrahedralization, for such sets is expected linear. Indeed we have obtained good running times when computing with REGTET regular tetrahedralizations for sets of uniformly distributed points in cubes: on the SGI ONYX2 (300 Mhz R12000 CPU) the running time is about 25 CPU minutes for a set of 512,000 points with random weights. A similar time was obtained for the same set without weights. Finally, REGTET has also been executed successfully and efficiently to compute Delaunay tetrahedralizations for non-uniformly distributed point sets representing sea floors and cave walls.

References

1. Aurenhammer, F.: Power diagrams: properties, algorithms and applications. *SIAM J. Comput.* **16** (1987) 78–96
2. Bernal, J.: On the expected complexity of the 3-dimensional Voronoi diagram. NISTIR 4321 (1990)
3. Bernal, J.: Lexicographical manipulations for correctly computing regular tetrahedralizations with incremental topological flipping. NISTIR 6335 (1999)
4. Delaunay, B.: Sur la sphère vide. *Bull. Acad. Sci. USSR (VII), Classe Sci. Mat. Nat.* (1934) 793–800
5. Dwyer, R. A.: Higher-dimensional Voronoi diagrams in linear expected time. *Discrete Comput. Geom.* **6** (1991) 343–367
6. Edelsbrunner, H., Shah, N. R.: Incremental topological flipping works for regular triangulations. *Algorithmica* **15**(3) (1996) 223–241

7. Guibas, L. J., Knuth, D. E., Sharir, M.: Randomized incremental construction of Delaunay and Voronoi diagrams. Springer-Verlag Lecture Notes in Computer Science **443** (1990) 414–431
8. Lawson, C. L.: Software for C^1 surface interpolation. Mathematical Software III, J. R. Rice (Ed.), Academic Press, New York (1977) 161–194
9. Lawson, C. L.: Properties of n-dimensional triangulations. Computer Aided Geometric Design **3** (1986) 231–246
10. Voronoi, G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. J. Reine Angew. Math. **134** (1908) 198–287