# DELAUNAY TETRAHEDRALIZATIONS: HONOR DEGENERATED CASES

Edward Verbree

Delft University of Technology, OTB Research Institute for the Built Environment
Section GIS Technology
Delft, the Netherlands
e.verbree@tudelft.nl

**Commission IV, WG IV/8**

**ABSTRACT**:

The definition of a Delaunay tetrahedralization (DT) of a set $S$ of points is well known: a DT is a tetrahedralization of $S$ in which every simplex (tetrahedron, triangle, or edge) is Delaunay. A simplex is Delaunay if all of its vertices can be connected by a circumsphere that encloses no other vertex. An important remark made in virtually all papers on this topic is that "although any number of vertices is permitted on the sphere", a Delaunay tetrahedralization is unique "only if the points are in general position, which is the absence of degeneracy (i.e., five or more vertices possible on the circumsphere)".

For applications in which the DT must be unique and invariant invariable under either geometrical rotation or the numbering of the points of set $S$, degenerated cases are expected to be resolved or at least indicated.

This research, however, will direct to a method in which degenerated cases are *not* solved by a geometrically distorted set of input data using Steiner Points, nor is just one of the possible 'unique' DTs generated. To the contrary, an extra type of 'non-geometrical', flat, zero-volume 'tetrahedra' is purposely introduced in order to indicate the degenerated cases within the applied data structure of the tetrahedralization. The four vertices of these flat tetrahedra lie on the same circumsphere as the vertices of their neighboring tetrahedra.

Once the set $S$ of points is tetrahedralized, these flat tetrahedra provide a tool with which to discover other 'unique' DTs of the set $S$ of points. From this set of unique DTs, the one having a global optimum can be selected for a given purpose, e.g. to support data-dependent applications. Another benefit in creating flat tetrahedra is the ability to indicate nearly degenerated cases, eliminating the need for exact arithmetic.

## 1. DELAUNAY DIAGRAMS

### 1.1 Ambiguities

For any set of four co-circular points, two different diagonals can be drawn, creating two possible triangulations of the points. Hjelle and Daehlen (2006) refer to this configuration of four co-circular points with two possible triangulations that both satisfy a certain criterion as a "neutral" case for that criterion. They state: "... we often assume that neutral cases ... do not exist when developing the theory. This leaves out lengthy details and boring explanations in the proofs of the theorems". They "try to avoid these ambiguities ... by often assuming that neutral cases do not exist".

But 'neutral cases' do exist, as the Delaunay empty circle criterion allows multiple, thus non-unique, occurrences of Delaunay triangulations of the set of points $S$. Only when the points are in general position, which is the absence of degeneracy (i.e., four or more vertices being possible on the circumcircle) is the Delaunay triangulation unique.

### 1.2 Multiple Delaunay Triangulations

A set of points $S$ with degenerated cases has multiple Delaunay triangulations (DTris). One way to avoid this occurrence of multiple Delaunay triangulations is to locally avoid the triangulation of these degenerated cases and keep to the so-called Delaunay diagram. A Delaunay diagram is the result of removing all edges in the Delaunay triangulations that are sides of triangles whose vertices are co-circular with a fourth point.

This property, as expressed by Mitchell (2005), is that "the Delaunay diagram is a plane graph. If there are subsets of four or more sites that are co-circular, the faces of the Delaunay diagram may not all be triangles; some may be convex polygons".

This issue is addressed by Liu and Snoeyink (2006). They handle Delaunay diagrams in 4-D and state: "The Delaunay diagram in $d$ dimensions is the dual of the Voronoi diagram of a set of input sites. If we assume no degeneracies in the input, i.e., if no sites are co-spherical, then the diagram is a triangulation. Because this assumption is common, and can be enforced by symbolic perturbation, we often forget that Delaunay diagrams need not be triangulations". Thus, "the cells in Delaunay diagrams are generally not simplicial: they are

convex hulls of co-spherical points." Having access to the Delaunay diagram can be seen as an advantage: "instead of letting the triangulation of the cells be induced by a perturbation policy, the user directly controls the triangulations. When the input contains many degeneracies, we have another advantage: the size of the Delaunay diagram is much smaller than that of the Delaunay triangulation of the perturbed sites". They give an algorithm to allow the direct computation of 4-D Delaunay diagrams. Their implementation is "competitive against a Delaunay triangulation program when there are many degeneracies".

## 2. NUMERICAL PRECISION, EXACT ARITHMETIC AND ROBUSTNESS

### 2.1 Degenerated cases

For most applications, degenerated or 'neutral' cases are not a problem in the sense of the possible existence of multiple occurrences of Delaunay triangulations (in 2-D) or Delaunay tetrahedralizations Tets (in 3-D) for the same set of input points. However, due to the nature of floating-point arithmetic, which approximates real numbers to the closest possible value, numerical instability will cause severe problems in the robustness of an implementation. This problem is also addressed by Alliez et al. (2000): "Bugs that occur due to improper detection or handling of degenerate cases can be difficult to reproduce and, therefore, to track down and fix. Furthermore, it is tedious and time-consuming to design test suites to cover all the degenerated cases. Thus, as computational geometry turns more into implementation of its algorithms, it is not surprising that there is much interest in methods to automatically handle degeneracies".

The core of each Delaunay implementation is the InCircle (in 2-D) or the InSphere (in 3-D) predicate. The in-circle test will determine whether or not a point $p$ lies inside, on, or outside the unique circumscribed circle passing through the three non-co-linear points $a$, $b$, and $c$ in $R^2$.

In Guibas and Stolfi (1985) it is shown that the InCircle test is reduced to the computation of the determinant of a matrix. A positive value for the InCircle test holds true in cases in which the orientation of the triangle is counter-clockwise; if the orientation is clockwise, this value will be negative. An identical translation to every point will not change the result of the determinant the origin can be moved to any one of the points and the InCircle predicate is implemented as the determinant of a 3x3 matrix.

This InCircle test generalizes to 3-D, where this InSphere test is reduced to the computation of the determinant of the 4x4 matrix. If the InSphere returns a positive value, this means that point $p$ is inside the sphere by virtue of the four non-spherical points $a$, $b$, $c$, and $d$ in $R^3$; a negative value indicates that point $p$ is outside the sphere. If InSphere returns to 0, point $p$ is 'on' the circle/sphere.

### 2.2 Epsilon

Although it is not necessary to know the value of an InCircle/InSphere test, but merely whether it is positive or negative, degenerated and nearly degenerated cases will result in near-zero values of their determinants, either negative or positive. One way to deal with this uncertainty is to apply a widely used method for solving this numerical non-robustness: the *tolerance* or *epsilon*. But according to Ledoux (2006), this will not solve the problem of how to define the "optimal tolerance". "Tweaking the tolerance to fix a given problem was usually easy, the problem being that it was also creating another problem somewhere else." Geometrically speaking, this 'somewhere else' should be taken literally, as point $p$ being considered 'inside' one circumsphere will cause it to be 'outside' another circumsphere.

### 2.3 Adaptive exact arithmetic

To handle the numerical non-robustness of degenerated and nearly degenerated cases, in practical terms "it should also be pointed out that the implementation of a perturbation scheme is effective only if exact arithmetic is used for all the predicates involved" Ledoux (2006). Ledoux retains a solution credited to Shewchuk (1997) "in which fast geometric predicates are developed and implemented that are 'adaptive', which means that their speed is inversely proportional to the degree of uncertainty of the result. This method works like a filter that will activate exact arithmetic only when it is needed, but since the exact result of the predicate is not sought, as soon as the sign of the determinant can be correctly computed, the process stops".

This approach is also followed by Si (2008) in his implementation of Tetgen by Si (2007): "The robustness of the implementation is achieved by using the adaptive exact floating-point arithmetic of Shewchuk (1997) to perform the two geometric predicates, i.e., the three-dimensional orientation test (orient3d) and the point-in-sphere test (InSphere). In addition, a simple symbolic perturbation approach Shewchuk (1998) is implemented in the point-in-sphere test to handle degenerated inputs".

## 3. PERTURBATIONS

### 3.1 Geometrical perturbations

If the DT has to be unique, a degenerated case can be resolved by some arbitrarily small perturbation of the set of points $S$, it can be ensured that this perturbation will not cause a new degeneracy, as discussed by Alliez et al. (2000). They developed "problem-specific schemes for resolving degeneracies by a combination a perturbing the problem and perturbing the world. In both techniques, a parameter ε is introduced and when ε goes to zero, the limit case is our original problem. Perturbing the problem approach transforms the definition of the problem while perturbing the world approach introduces a global transformation of the world of data."

However, this solution could cause 'slivers', nearly flat tetrahedra with close to zero volume. These 'slivers' can be removed in a refinement step, but at the cost of extra Steiner Points. Besides, geometrical perturbations of the input data set $S$ should be avoided in any case.

### 3.2 Symbolic perturbations

Apart from perturbing the problem by introducing a geometrical ε is the addition of a rule to the empty circumcircle criterion, e.g. a symbolic perturbation, as shown by Edelsbrunner and Mücke (1990) and adapted by Shewchuk (2000). An

implementation of the symbolic perturbation scheme in the InSphere test is described well by Ledoux (2006): "First, the points are perturbed only when they are co-spherical; if they are not, then InSphere returns either a positive or negative result. When InSphere returns zero, simply move upward the lifted point with the highest index by adding an arbitrary amount (the value is not important) to its fourth coordinate … and re-compute InSphere. If the four remaining points are affinely independent, then the new result will be non-zero; if not then the points with the second highest index must be moved upward by a smaller value, and InSphere recalculated. This continues until the result of InSphere is non-zero".

A symbolic perturbation uses the highest index (node number) of the vertices as a criterion. This solution is not invariant against the renumbering of the vertices; two geometrically identical sets of vertices but a different indexing might result in two different Delaunay triangulations.

### 3.3 Preferred diagonals

Another possibility is to opt for a preferred direction for one of the diagonals as an edge of the triangulation, as discussed by Dyken and Floater (2006). But although these perturbing schemes result in a valid output, the derived DT remains just one of the possible DTs and not a unique DT. This solution is not invariant against a geometrical rotation of the input data set; i.e., if the *x*- and *y*-coordinates are reversed, so will all edges of the degenerated, co-circular cases.

Yet another option, supported by exact arithmetic, is to add another rule in addition to the empty circumsphere criterion. A symbolic point perturbation, or a preferred direction of the created triangles, will opt for one of the possible tetrahedral subdivisions having five or more points on one circumsphere. But as the degenerated cases will not be solved, this will result in only one of the possible 'unique' DTs with respect to the ordering or the geometry of the points.

## 4. RESEARCH RESULTS

### 4.1 Indicate degenerated cases

This research demonstrates a method in which degenerated cases are *not* solved by a geometrically distorted set of input data at the cost of Steiner Points, and in which more than one of the possible 'unique' DTs is generated. An extra type of 'non-geometrical', flat, zero-volume, 'tetrahedra' is purposely introduced so as to indicate the degenerated cases (fig. 1) within the applied data structure of the tetrahedralization. The four vertices of these flat tetrahedra lie on the same circumsphere as the vertices of their neighboring tetrahedra (fig. 2).

Once the set *S* of points is tetrahedralized, these flat tetrahedra provide a tool with which to discover other 'unique' DTs of the set *S* of points. From this set of 'unique' DTs, the one having a global optimum can be selected for a particular purpose, e.g. to support data-dependent applications. Another benefit is that 'flat tetrahedra' are created, also to indicate nearly degenerated cases, eliminating the need for exact arithmetic.
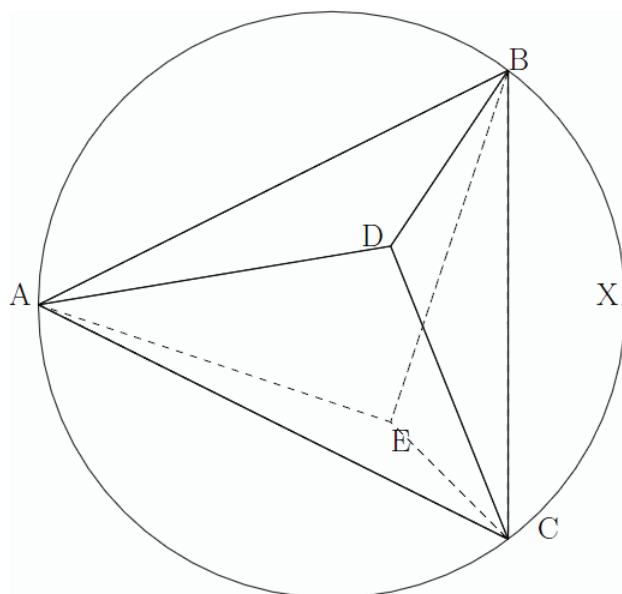


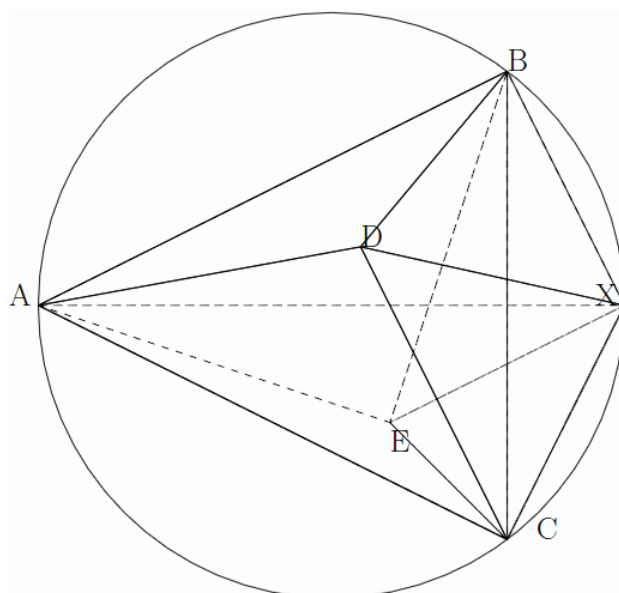Fig 1: Point X at circumspheres (A,B,C,D) and (E,C,B,A)



Fig 2: Flat tetrahedron (A,B,C,X) created.

### 4.2 Test results

The idea of introducing 'flat' tetrahedra tetrahedralization has been tested by the author through the implementation of a non-exact arithmetic Delaunay tetrahedralization and compared with the well-known exact arithmetic TetGen implementation [Si, 2007]. This testing was conducted with some challenging data sets − i.e., sets containing 25,000 points irregularly distributed over a sphere − as well as a regular cube-connected distribution. These tests prove both the stability and the speed of the tetrahedralization, which are comparable with TetGen.

## REFERENCES

Alliez et al. (2000) Pierre Alliez, Olivier Devillers, and Jack Snoeyink. Removing Degeneracies By Perturbing the Problem or Perturbing the World. *Reliable Computing*, 6: 61–79, 2000.

Dyken and Floater (2006) Christopher Dyken and Michael S. Floater. Preferred directions for resolving the non-uniqueness of Delaunay triangulations. *Computational Geometry: Theory and Applications*, 34: 96–101, 2006.

Edelsbrunner and Mücke (1990) Herbert Edelsbrunner and Ernst Peter Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9 (1): 66–104, 1990. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/77635.77639.

Guibas and Stolfi (1985) Leonidas Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi. *ACM Trans. Graph.*, 4 (2): 74–123, 1985. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/282918.282923.

Hjelle and Daehlen (2006) Oyvind Hjelle and Morton Daehlen. *Triangulations and Applications*. Spinger, 2006.

Ledoux (2006) Hugo Ledoux. *Modelling Three-dimensional Fields in Geoscience with the Voronoi Diagram and its Dual*. PhD thesis, University of Glamorgan/Prifysgol Morgannwg, 2006.

Liu and Snoeyink (2006) Yuanxin Liu and Jack Snoeyink. Sphere-based Computation of Delaunay Diagrams on Points from 4d Grids. *International Symposium on Voronoi Diagrams in Science and Engineering*, 0: 60–65, 2006. doi: http://doi.ieeecomputersociety.org/10.1109/ISVD.2006.32.

Mitchell (2005) Joe Mitchell. Notes on Voronoi and Delaunay Diagrams, 2005. URL http://www.ams.sunysb.edu/~jsbm/-courses/345/voronoi.pdf.

Shewchuk (1997) Jonathan Richard Shewchuk. Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates. *Discrete & Computational Geometry*, 18 (3): 305–368, 1997.

Shewchuk (1998) Jonathan Richard Shewchuk. A Condition Guaranteeing the Existence of Higher-Dimensional Constrained Delaunay Triangulations. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, pages 76–85, 1998.

Shewchuk (2000) Jonathan Richard Shewchuk. Sweep Algorithms for Constructing Higher-Dimensional Constrained Delaunay Triangulations. In *SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry*, pages 350–359, New York, NY, USA, 2000. ACM. ISBN 1-58113-224-7. doi: http://doi.acm.org/10.1145/336154.336222.

Si (2007) Hang Si. TetGen. http://tetgen.berlios.de, 2007.

Si (2008) Hang Si. *Three Dimensional Boundary Conforming Delaunay Mesh Generation*. PhD thesis, Technischen Universität Berlin, 2008.