The Poison Pump and the Spitting Fish

John Burkardt
School of Computational Science
Florida State University

....

https://people.sc.fsu.edu/∼jburkardt/presentations/...
death_map_2005_vt.pdf

Virginia Tech Student SIAM Chapter, 26 April 2005

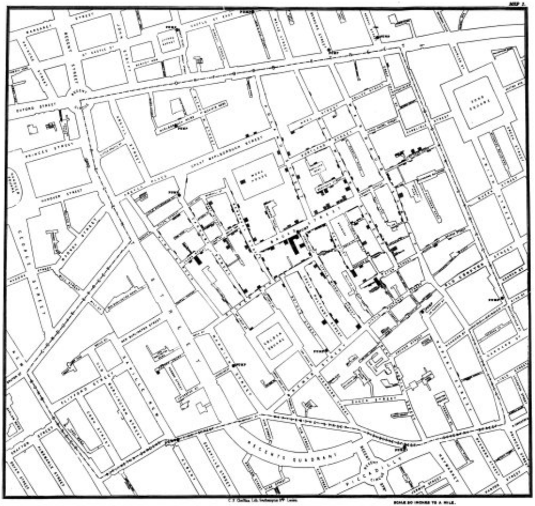
The Voronoi Diagram and the Delaunay Triangulation

A famous geometer should have said, *"All geometry is local."*

We mean that life is full of bumps and shoves; the most important influence on you is your neighbors. Far away places influence you only indirectly, through a chain of intermediaries.

The Voronoi diagram and the Delaunay triangulation are natural geometric structures that summarize this concept of nearness and neighborhood.

A Map of the Golden Square Area

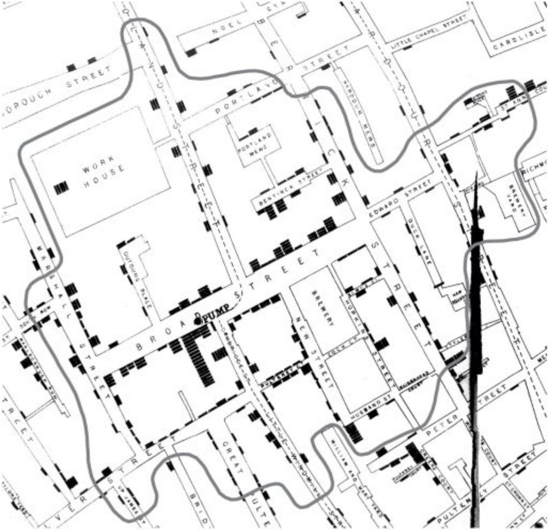


John Snow's Investigation

Dr John Snow suspected a particular water pump was the source of the cholera outbreak, but the pump water seemed relatively clean when he examined it.

- He made a map of where victims lived;
- He paced the distance to the nearest pump;
- He drew a line around the houses closest to the Golden Square pump.
- Some victims outside the line nonetheless used the pump.

The Suspected Pump and its Neighborhood



Conclusion

People minimize the distance they travel;

The path of a disease can be determined by local traffic patterns.

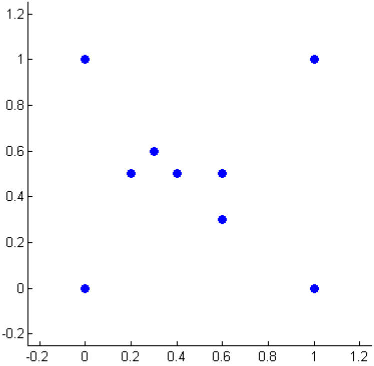
Understanding local geometry reveals natural neighborhoods and connections.

The Voronoi Diagram

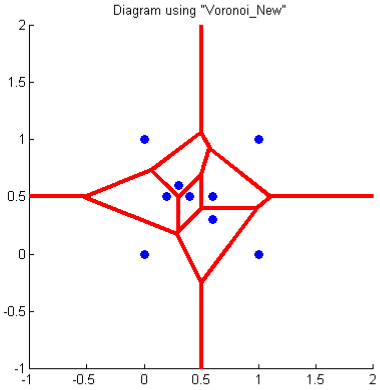
Given a set of "generator" points, associate every point in the plane with its nearest generator. The result is a kind of graph.

- The plane is partitioned into regions or "Faces";
- Points equally far from two generators form Edges;
- Points equally far from three generators form Vertices;

A Simple Set of Points



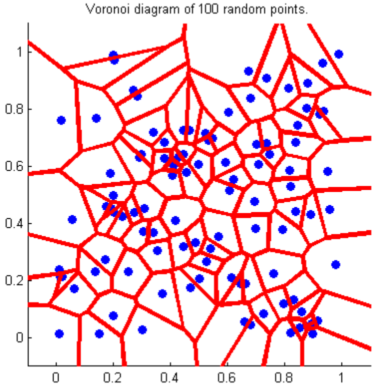
The Voronoi Diagram



Some Properties of the Voronoi Diagram

- Generator $\Rightarrow$ Face;
- Generator on convex hull $\Rightarrow$ semi-infinite face;
- Faces are convex and polygonal;
- Each face = intersection of pairwise halfplanes;
- Edges are perpendicular bisectors of neighbor lines;
- Vertices are equidistant from 3 generators;
- Each vertex defines an “empty” circle;

The Voronoi Diagram of 100 points



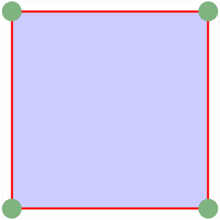
With a larger set of points, irregularities become obvious.

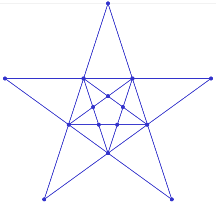
Euler's Formula for Bounded 2D Figures

Euler's formula, for a convex polyhedron in 3D, relates faces, vertices, and edges.

$$F + V = E + 2 \tag{1}$$

For a 2D bounded diagram, we need to count a single "infinite face".



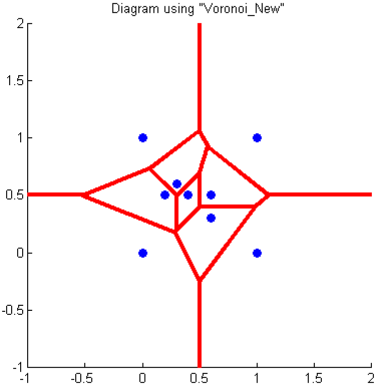


$$(1 + 1) + 4 = 4 + 2 \tag{2}$$

$$(16 + 1) + 15 = 30 + 2 \tag{3}$$

Euler's Formula for Unbounded 2D Figures

For an unbounded diagram, we need to count the infinite faces, and an extra point at infinity.



$$(5+4)+(11+1)=19+2 \tag{4}$$

How “Big” is a Voronoi Diagram

N points can have up to $N * (N - 1)/2$ pairwise connections.
Each point corresponds to a face. Each edge has two vertices (one might be infinity). Each vertex belongs to three (at least) edges (and infinity can have many more.)

$$3V \leq 2E \tag{5}$$

Substituting for V in Euler's equation gives:

$$E \leq 3 * N - 6 \tag{6}$$

Substituting for E in Euler's equation gives:

$$V \leq 2 * N - 4 \tag{7}$$

The complexity of the diagram is linear with respect to N.
Neighbors average about 6.

Voronoi Construction Algorithms

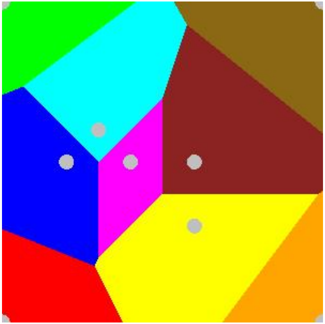
Naive approach: each Voronoi polygon V_i is the intersection of half-planes. Time is $O(N \log N)$ per polygon, so $O(N^2 \log N)$ total.

Better: *Fortune's sweepline algorithm* - pass a horizontal line from top to bottom, and construct the diagram. Work is $O(N \log N)$ (optimal).

Incremental or "local" algorithm: add one more point. Only a few Voronoi regions need to be modified.

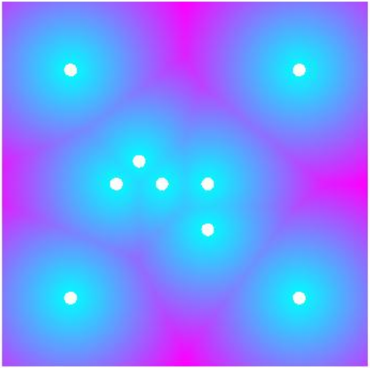
Algorithms and data structures are surprisingly complicated.

It's Easy to Plot



Pixel plots are very easy to make.

A 3D Contour Plot



A contour plot of distance to nearest generator.

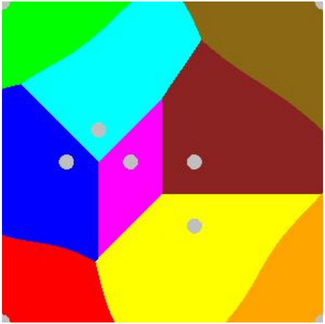
A 3D Surface Plot

A side view, using a different "mountain chain".

Generalizations

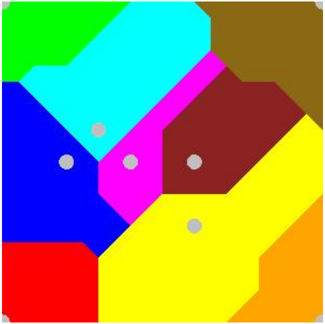
- any reasonable distance can be used;
- easily extended to higher dimensions;
- a nonuniform density or weight factor can be included;
- region can be finite; region can be a manifold;
- generators could be lines, or polygons;
- generators could be an *infinite set* of points; using each point on the boundary of a region generates the *medial axis*.

Pixel Plot with L3 distance

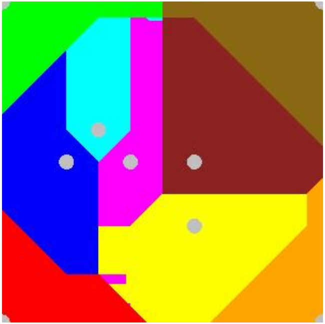


The edges are no longer straight.

Pixel Plot with $L\infty$ distance

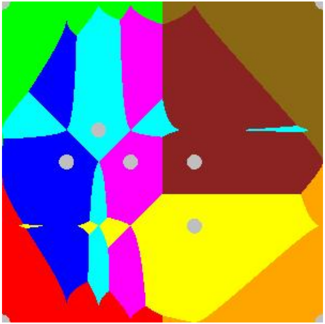


Pixel Plot with L1 distance



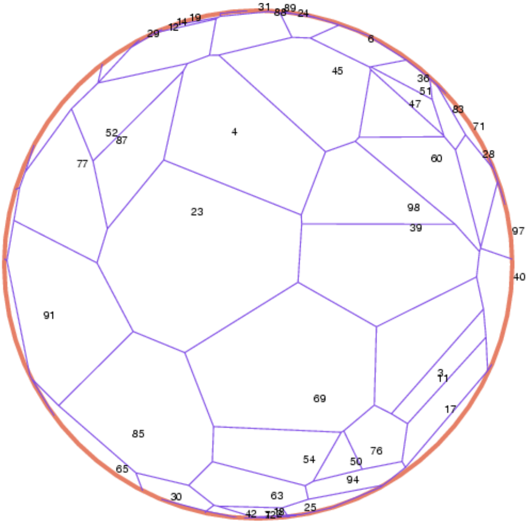
The “edges” can have finite thickness.

Pixel Plot with $L_{\frac{1}{2}}$ distance



The regions are no longer connected.

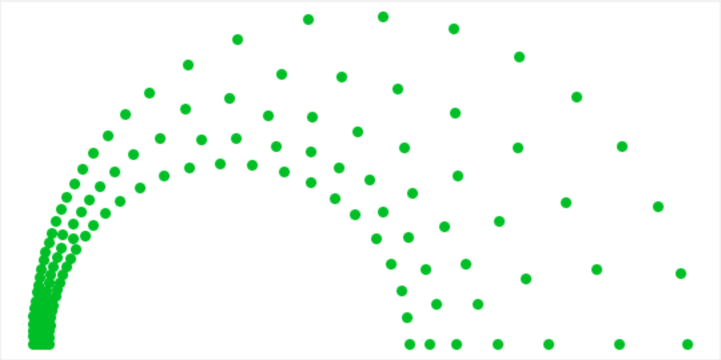
Voronoi Diagram on a Sphere



Simple Voronoi Applications

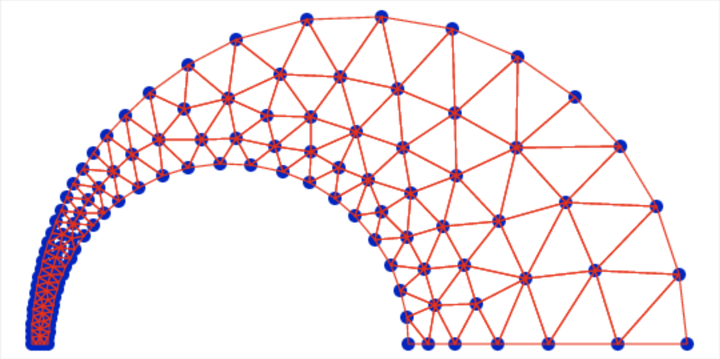
- what area is surveyed by each forest fire tower?
- if we put a new store **here**, how many customers will it get?
- can a robot move from A to B without hitting any obstacles?
- what is the largest empty circle I can draw in a set of points?

Triangulations



Given N points, we can draw many planar graphs (no edge crossing allowed.) A *maximal planar graph* is one for which no more edges can be added. A maximal planar graph is a *triangulation*, that is, it decomposes the convex hull of the N points into disjoint triangles.

A Sample Triangulation



This is a triangulation of the previous set of points. It is a **Delaunay triangulation**.

The Delaunay Triangulation

An edge of a triangulation is “locally Delaunay” if replacing it by the other diagonal would result in a smaller minimum angle.

A triangulation is a Delaunay triangulation iff every edge is locally Delaunay.

Every triangulation has a minimum angle. The Delaunay triangulation has the *maximum* minimum angle.

The Empty Circumcircle

The Delaunay triangulation has two interesting "empty circle" conditions.

Triangle: the circumcircle of each triangle is "empty".

Edge: every edge is contained in some "empty" circle.

An Incremental Delaunay Algorithm

Initial triangulation D_0: one triangle T_0 contains all the points.

- for points p_1 to p_n do:
  - find T in D_{i-1} containing p_i;
  - initialize triangulation D_i by replacing T by three subtriangles with p_i as one vertex;
  - while any edge (a, b) not locally Delaunay:
    - flip (a, b) to other diagonal (c, d).

    end while

  end

This incremental algorithm is also “local”.

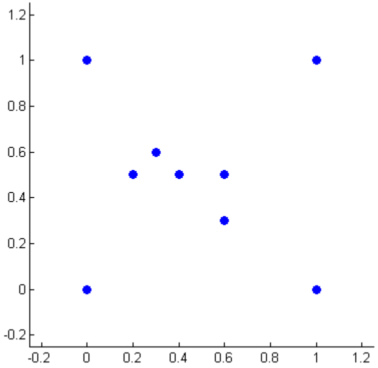
Simple Delaunay Applications

- where is the nearest neighbor P_i to a random point (x, y)?
- estimate a function $Z(x, y)$ from values at scattered points;
- generate a triangulation suitable for finite element calculations.

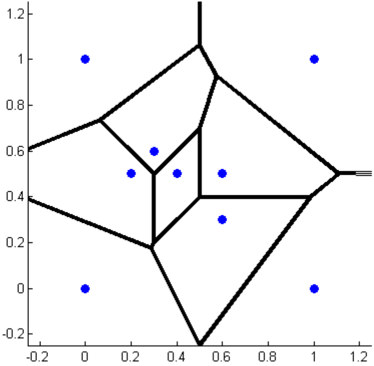
Voronoi and Delaunay are Duals

Voronoi	$\Longleftrightarrow$	Delaunay
Generators	$\Longleftrightarrow$	Nodes
Edge separates generators	$\Longleftrightarrow$	Edge joins nodes
Vertex	$\Longleftrightarrow$	Triangle

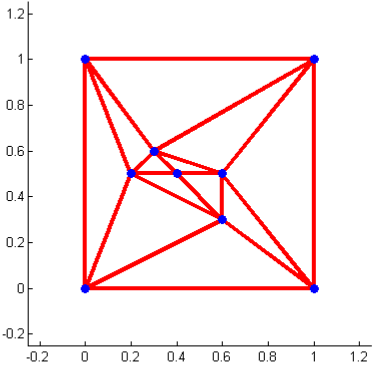
The Nodes



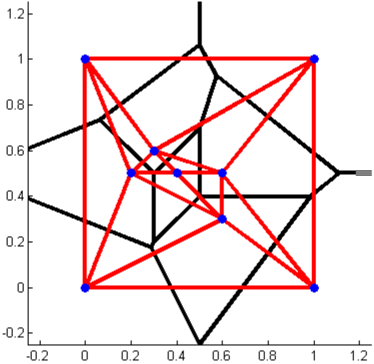
The Voronoi Diagram



The Delaunay Triangulation



Delaunay + Voronoi



The Nesting Behavior of Tilapia

Discussion

Are these fish doing geometry?

This is not a "random" Voronoi diagram!

- Each cell is about the same size;
- Each cell is about the same shape;
- The cells are "centered".

A Centroidal Voronoi Diagram

Given a set of N points $P_i \in \Omega$, a bounded subset of the plane,

let V_i be the Voronoi region associated with P_i,
let C_i be the centroid of V_i.

The points P_i generate a *centroidal Voronoi tessellation* (**CVT**) if and only if

$$P_i = C_i \qquad \forall i = 1...N$$

Does a CVT Exist?

Given region Ω and order N, there is *always* at least one CVT; finding it may be hard, because the definition is implicit.

Special examples:

- a finite hexagonal grid (stable);
- a checkerboard (unstable).

For practical calculations, an iteration is necessary.

Lloyd's Method

Initial generators are arbitrary.

Iteration:

- Construct Voronoi diagram;
- Determine Voronoi polygon for each generator;
- Determine centroid of polygon;
- Replace generator by centroid.

Nonuniform density OK, affects centroid calculation.

MacQueen's Monte Carlo Method

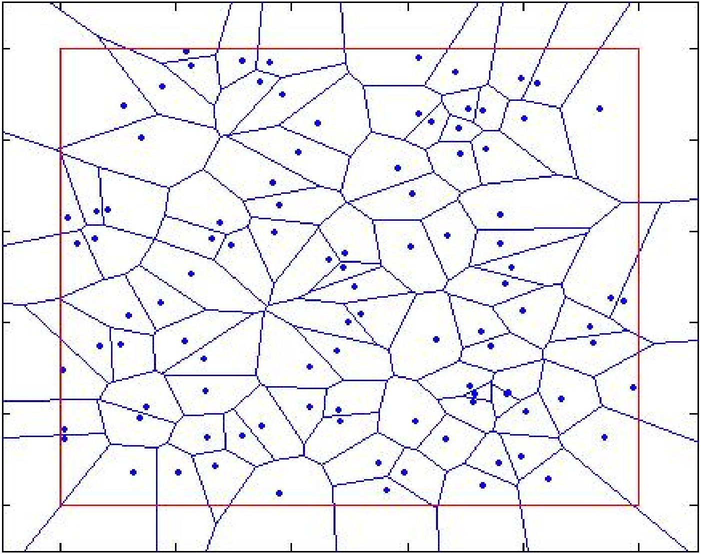
Initial generators arbitrary.

Iteration:

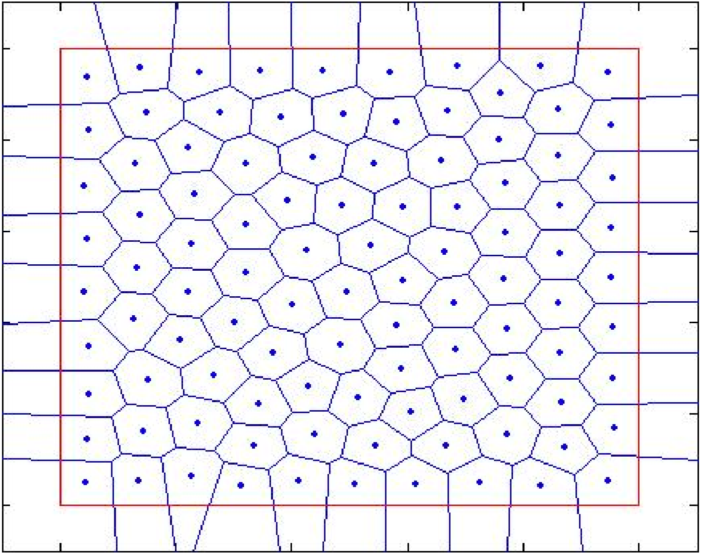
- Generate a random sample point S;
- Assign it to nearest generator P_i;
- Redefine P_i as weighted average with S;

Faster algorithm: use *thousands* of sample points.

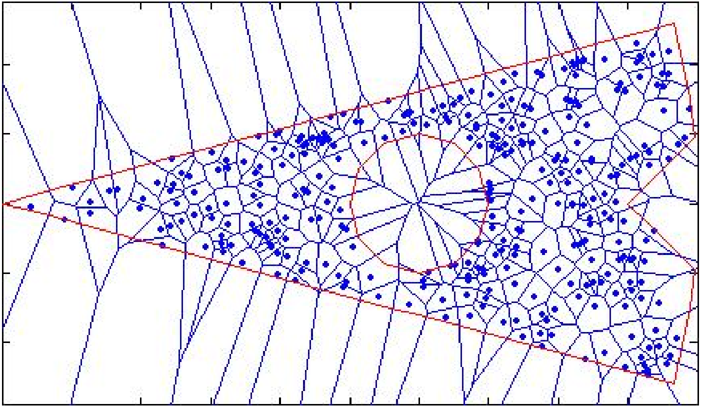
CVT in the Unit Square: Frame 1



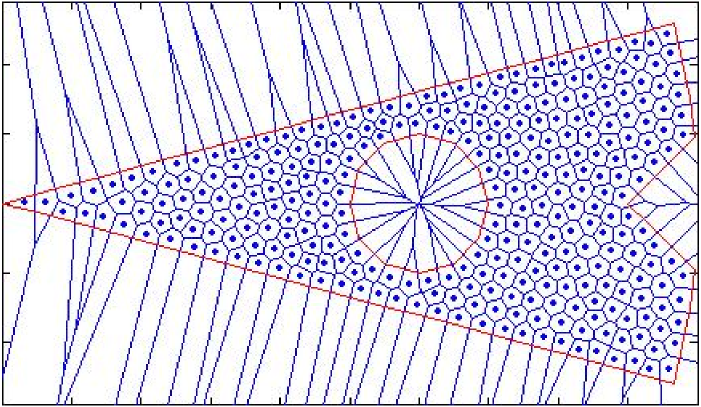
CVT in the Unit Square: Frame 100



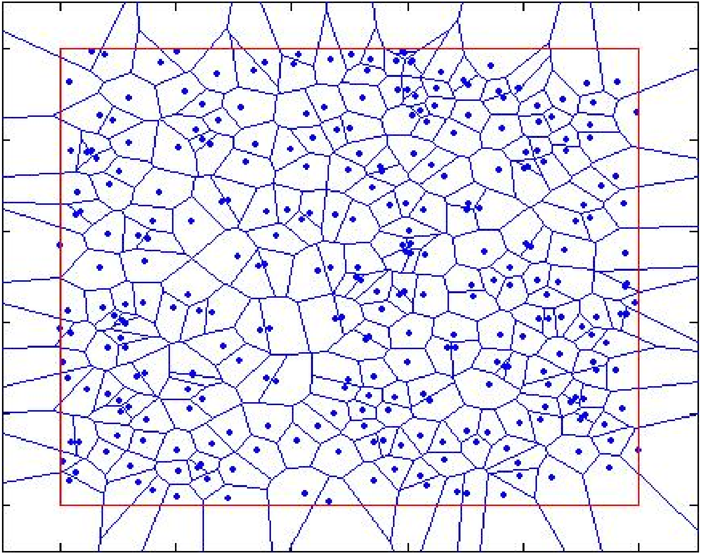
CVT in the Holey Pie: Frame 1



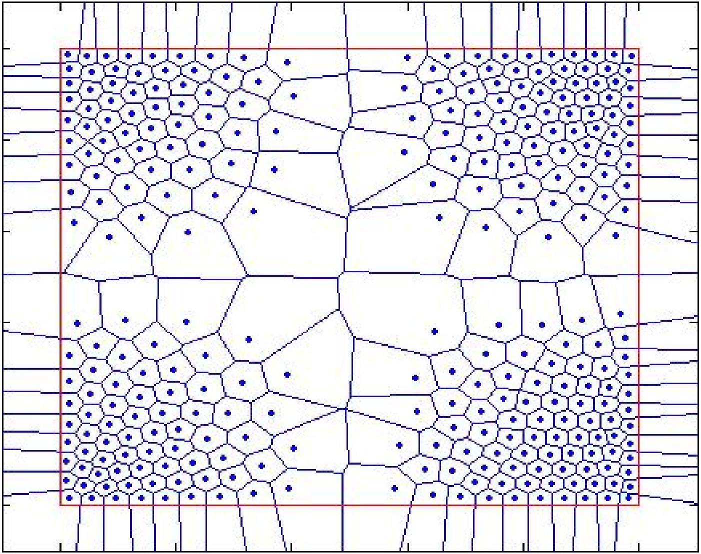
CVT in the Holey Pie: Frame 100



CVT with Nonuniform Density: Frame 1



CVT with Nonuniform Density: Frame 100



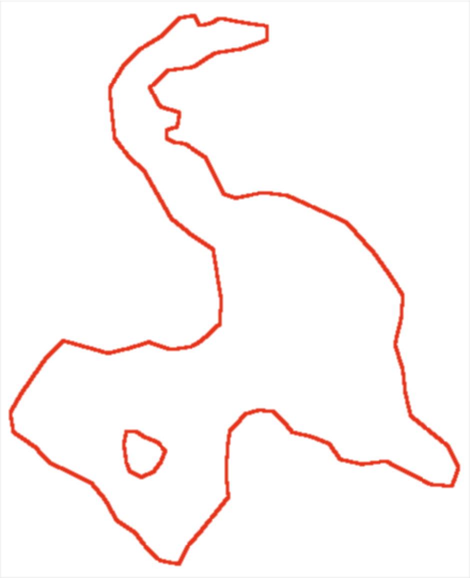
Improved Grids

Persson and Strang MATLAB code **DISTMESH**

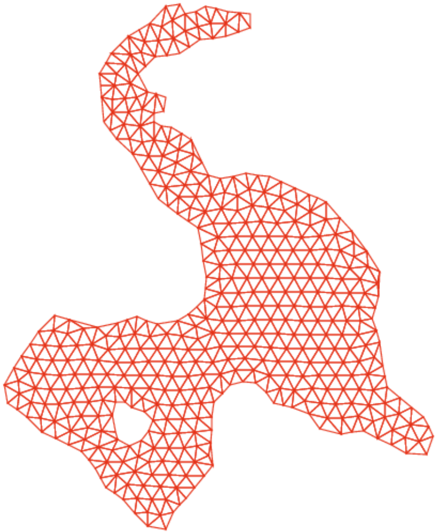
Goal - fill region and boundary with well spaced nodes and good triangulation.

- scatter nodes initially at random;
- connect Voronoi neighbors by springs **(important!)**;
- simulate how springs push nodes apart;
- springs also push nodes *out* – so push them back in!
- when done, produce Delaunay triangulation.

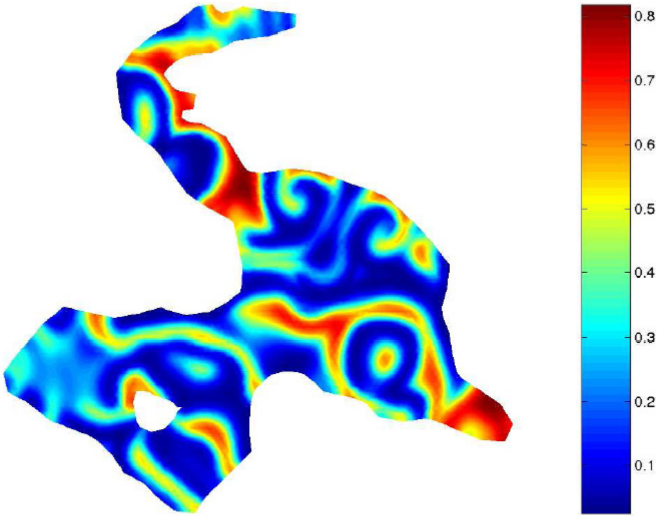
An Empty Region



The Final Mesh



A Predator Prey Calculation



Finale

In nature, squares, and infinity are rare. Triangles, hexagons and local influence are common.

The Voronoi diagram “understands” local neighborhoods.

The Delaunay triangulation “understands” the local connections between neighborhoods that create geometry.