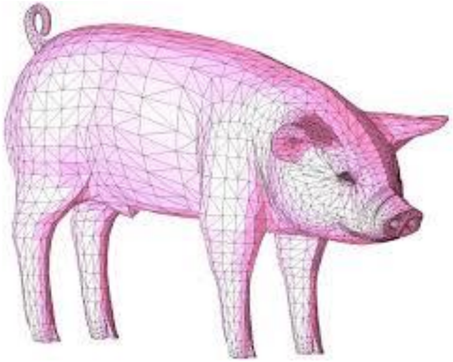
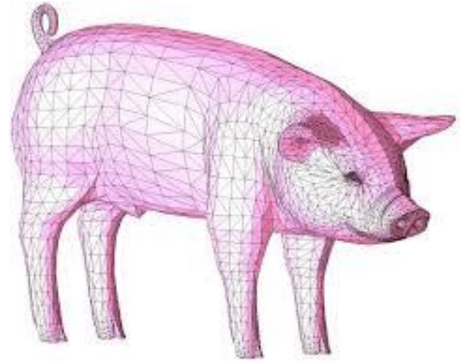
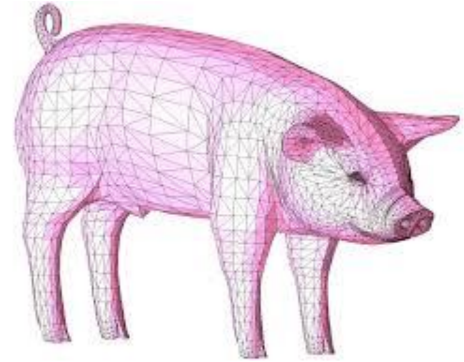
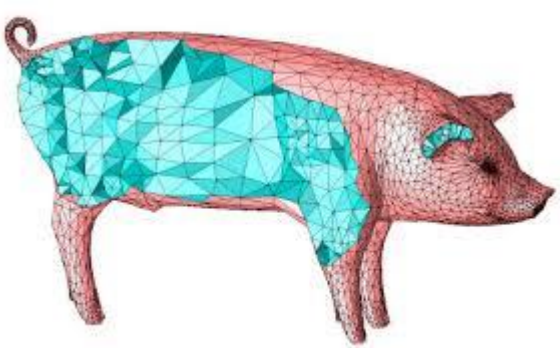
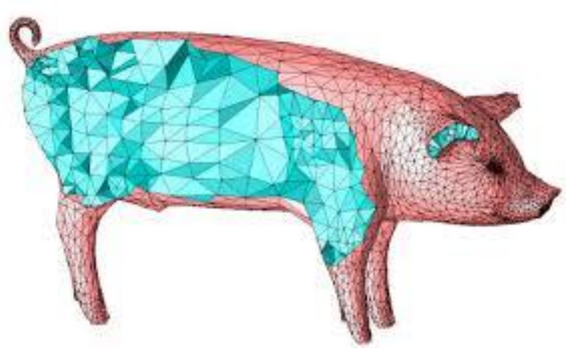
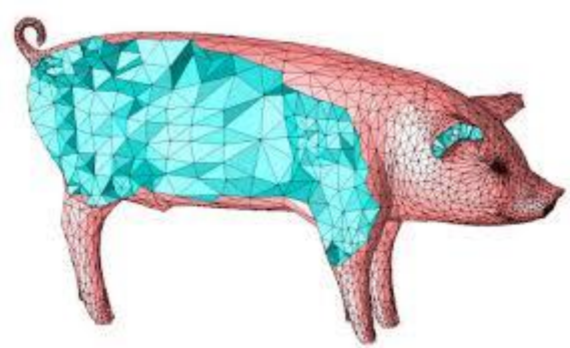
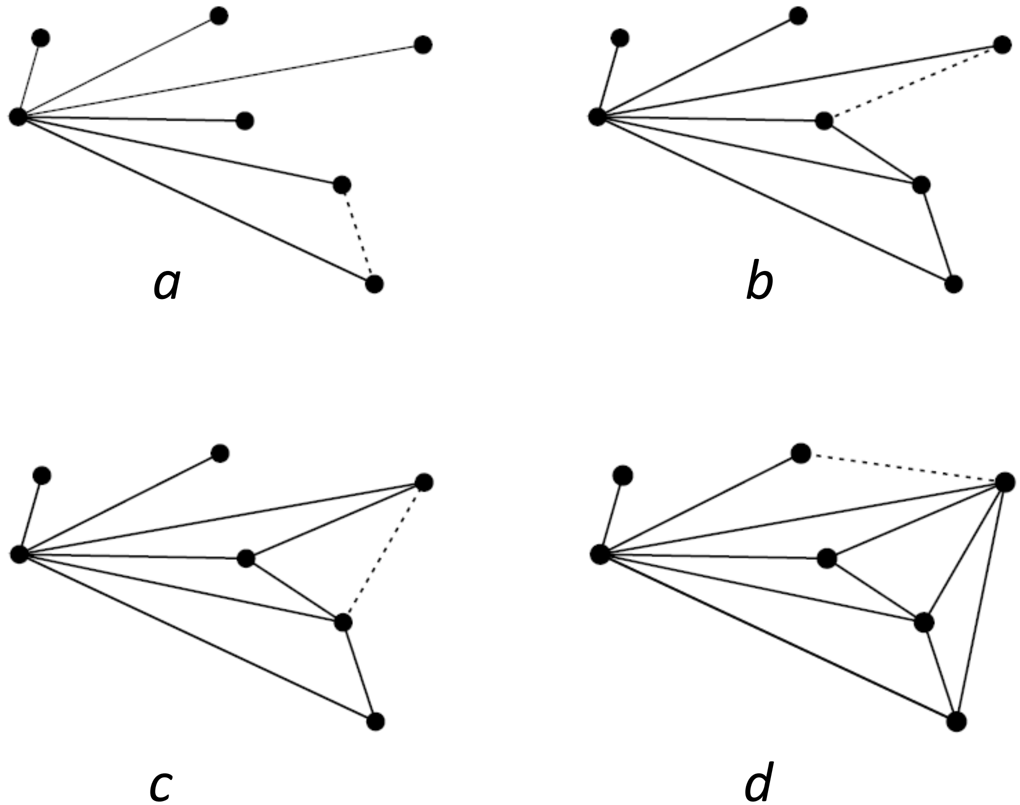




Mesh Generation

- Graham triangulation
- Delaunay triangulation
- Minimum weight triangulation
- Mesh generation

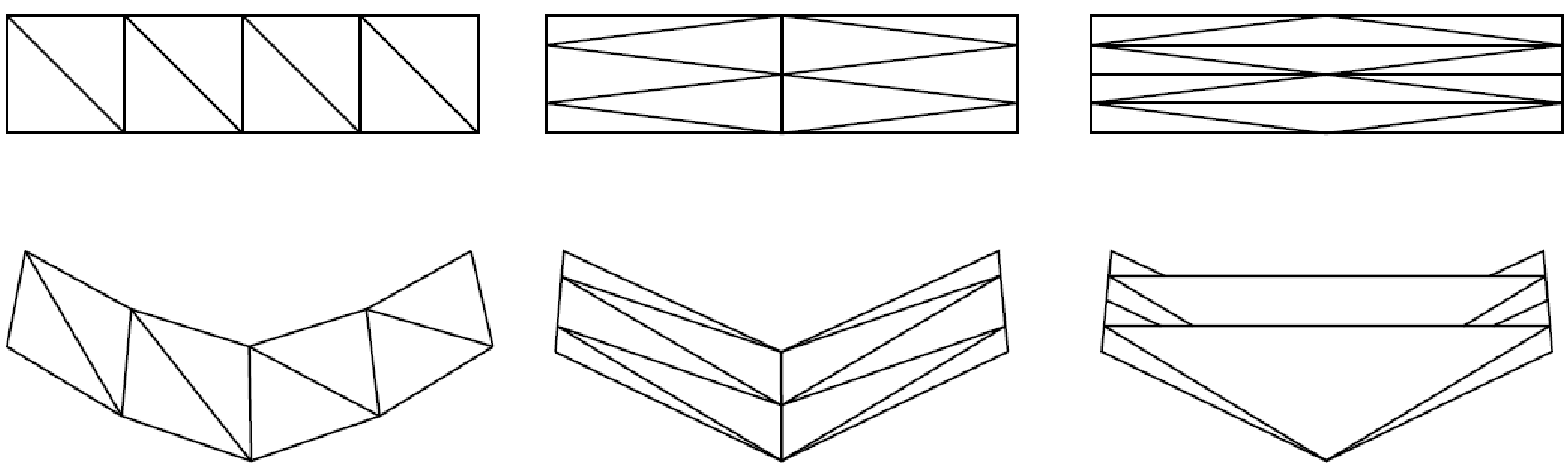




Graham Triangulation

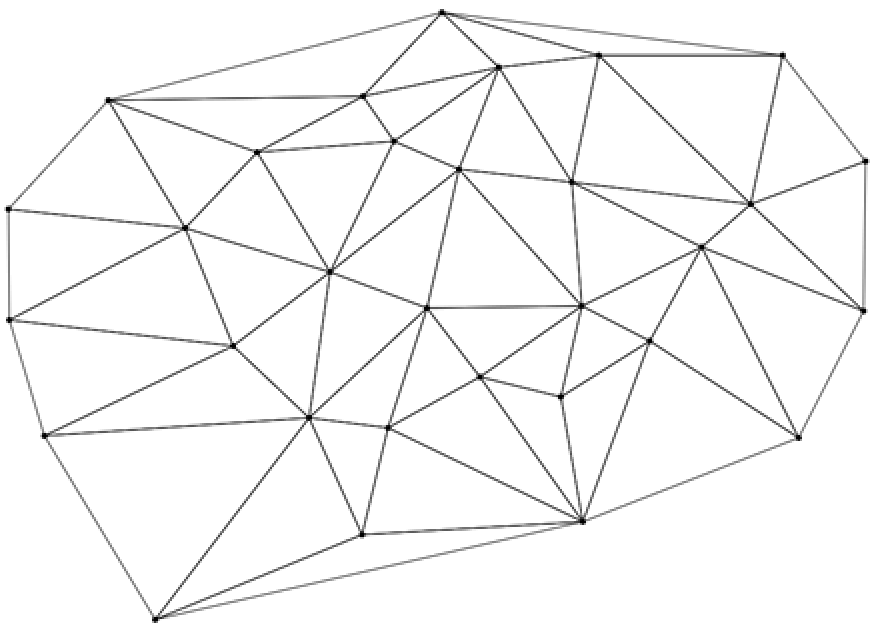


Linear Interpolation Error

$$\|f - g\|_\infty \le c_t \frac{l_{\max}^2}{6},$$
$$\|\nabla f - \nabla g\|_\infty \le c_t \frac{3 l_1 l_2 l_3}{4A}$$



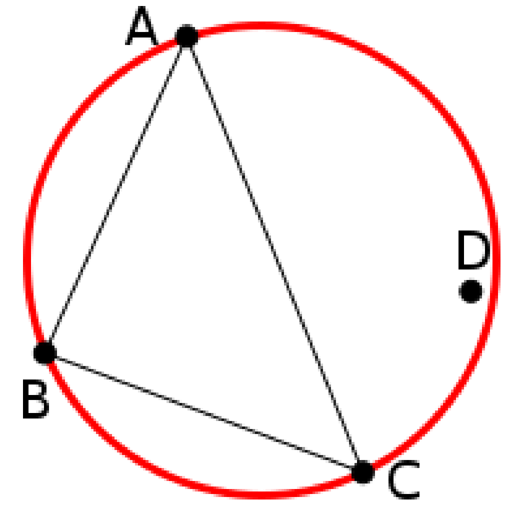
Delaunay Triangulation



Delaunay Condition

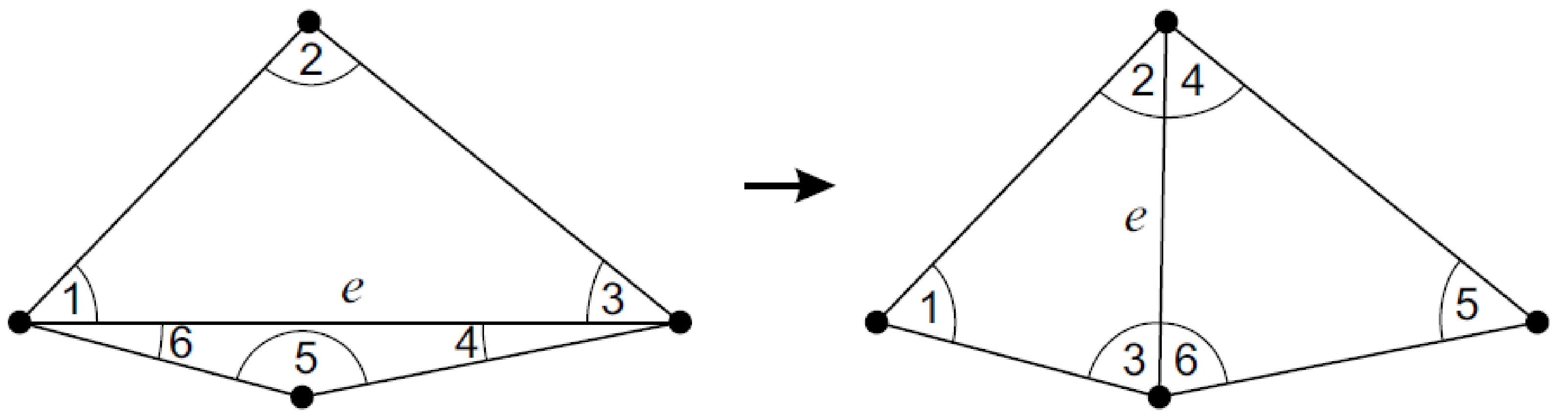
The point $D = (x, y)$ is strictly inside a circle passing through the points $A = (x_1, y_1), B = (x_2, y_2)$ and $C = (x_3, y_3)$ if

$$\begin{vmatrix} x & y & x^2 + y^2 & 1 \\ x_1 & y_1 & x_1^2 + y_1^2 & 1 \\ x_2 & y_2 & x_2^2 + y_2^2 & 1 \\ x_3 & y_3 & x_3^2 + y_3^2 & 1 \end{vmatrix} > 0$$



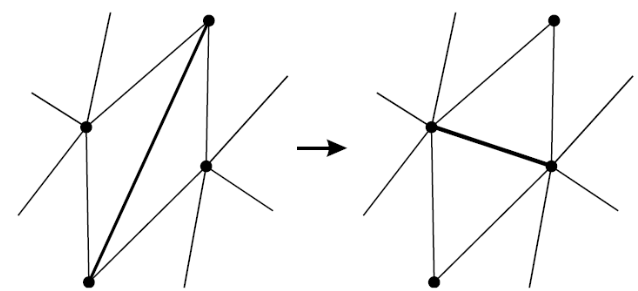
Delaunay Condition

Theorem. *The Delaunay condition holds if and only if the flip of the diagonal of any convex quadrilateral does not increase the minimum of the 6 adjacent angles.*



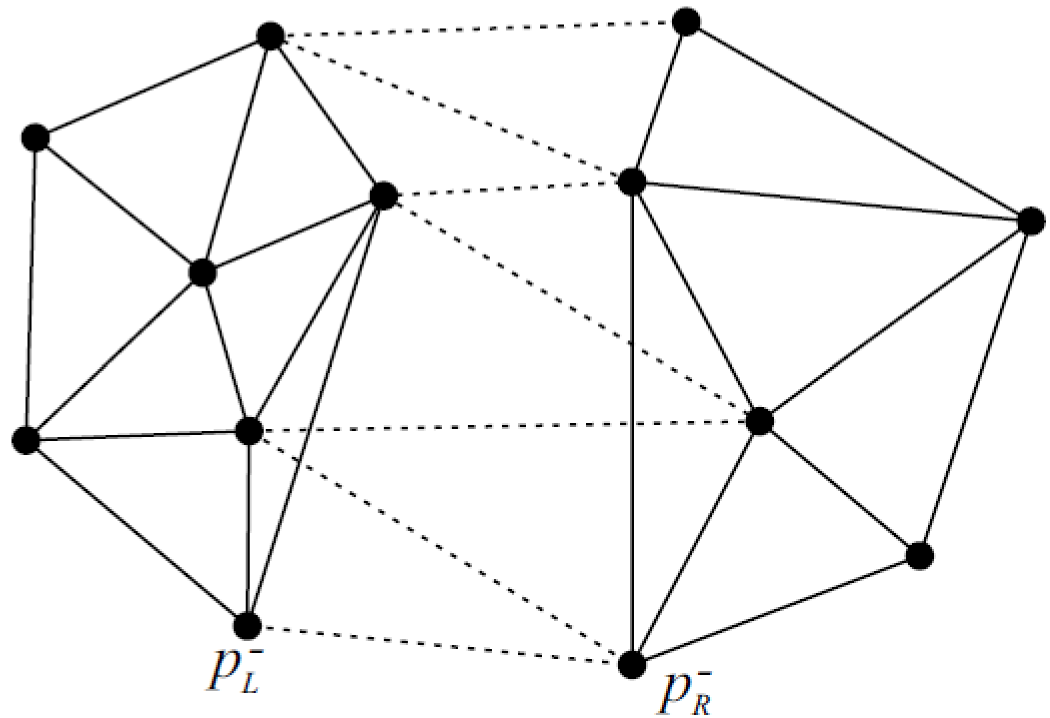
The Flip Algorithm

```
FlipTriangulation (T)
Insert all the edges of T in the queue Q
while Q ≠ NIL do
    e ← Pop(Q)
    if Check(e) then
        Flip(e)
        A ← GetAffectedEdges(T, e)
        for i ← 1 to 4 do
            if not Member(Q, A[i]) then
                Push(Q, A[i])
```



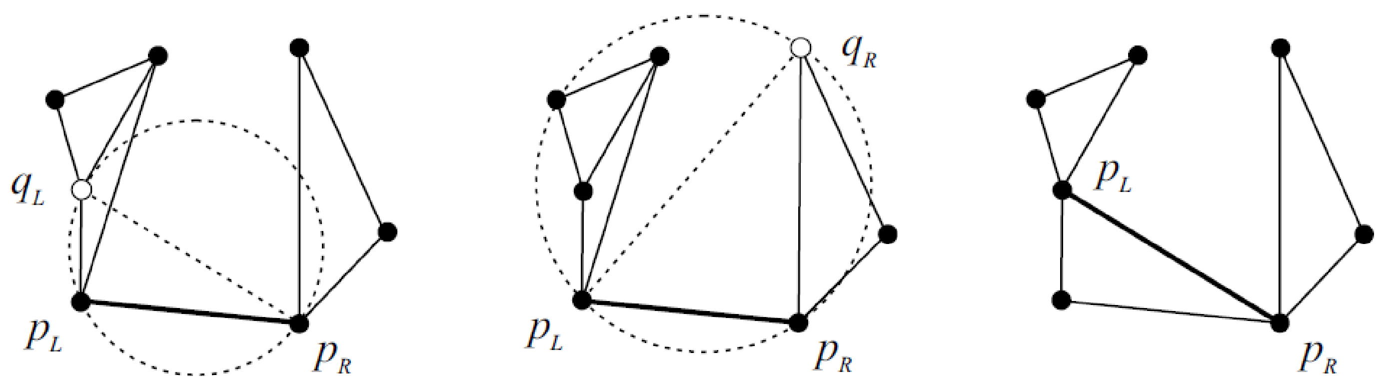
“Divide And Conquer” Approach

Binding of two triangulations computed recursively:

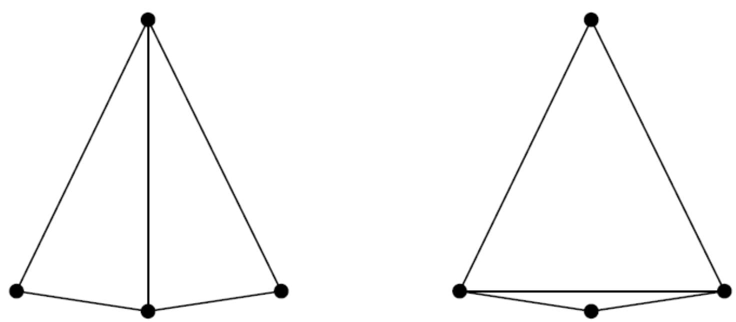


“Divide And Conquer” Approach

1. Find the point q_L such that the Delaunay condition holds for the left triangulation.
2. Find the point q_R such that the Delaunay condition holds for the right triangulation.
3. Define the new supporting edge $[q_L, p_R]$ or $[p_L, q_R]$ depending on for which of them the Delaunay condition holds for all the 4 points p_L, p_R, q_L, q_R.



Minimum-Weight Triangulation (MWT)



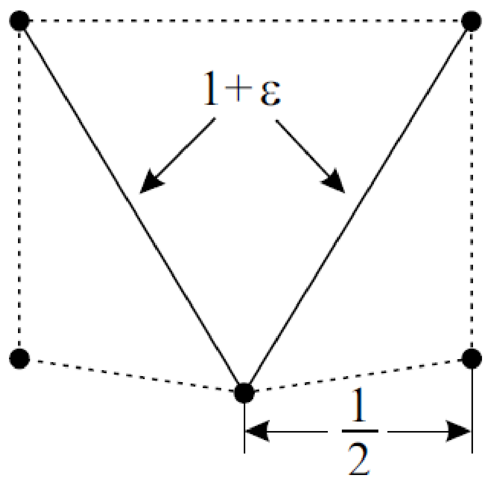
$M = l_1 + l_2 + l_3$

MINIMUM WEIGHT TRIANGULATION
Given a set $C = \{(a_i, b_i) : 1 \leq i \leq n\}$ of pairs of integers, which define positions of n points on the plane, and a positive number B, determine if there exists a triangulation of the set C such that its whole Euclidian length does not exceed B?

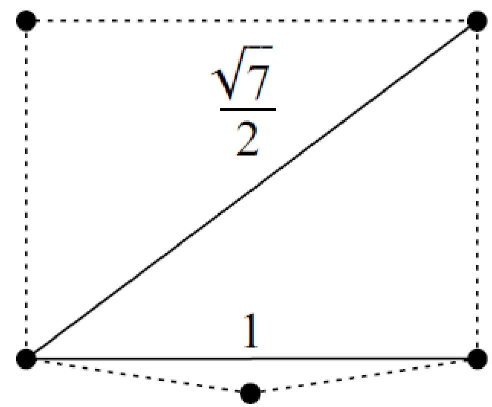
[Mulzer, Rote (2008)] **Minimum-weight triangulation is NP-hard**, *Journal of the ACM , 55 (2).*

Minimum-Weight Triangulation (MWT)

Neither the flip-algorithm nor the greedy algorithm produces the MWT:



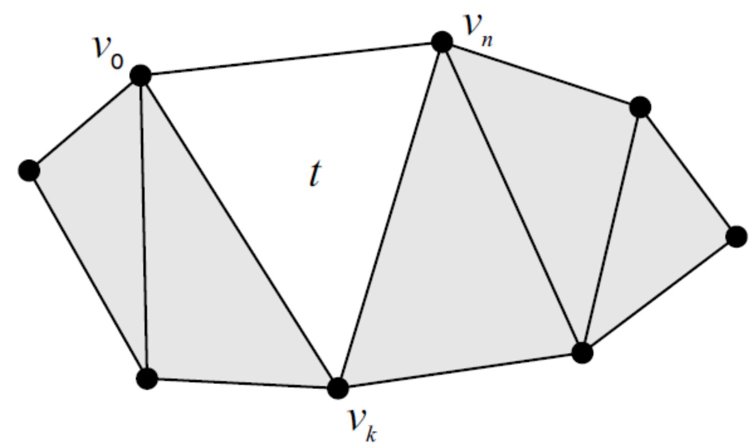
$$T^* = 2 + 2\epsilon$$



$$T = 1 + \frac{\sqrt{7}}{2} \approx 2.33$$

MWT of a Convex Polygon

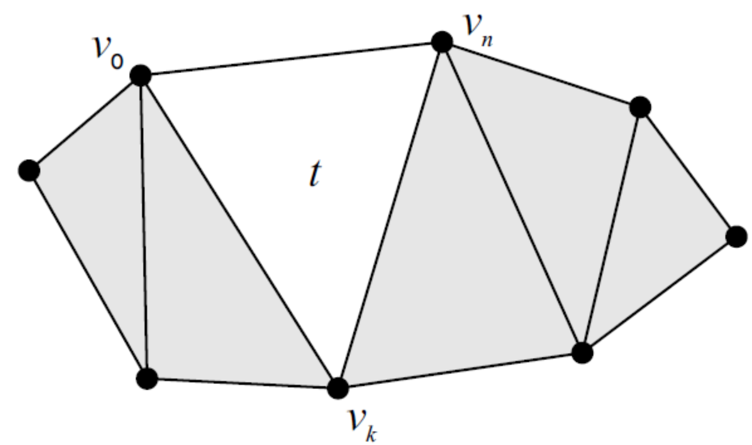
- At least one chord adjoins any edge of the polygon.
- For any chord (v_i, v_j) there exists a vertex v_k such that each of the segments (v_i, v_k) and (v_k, v_j) is either a chord or an edge of the polygon.



- Let $w[i, j]$ is the length of MWT of the polygon $v_i, \dots, v_j$, $0 \le i < j \le n$. Then the length of the whole MWT is $w[0, n]$.
- $w[i, j] = \begin{cases} 0, & i = j, \\ \min\limits_{i<k<j} \left(w[i,k] + w[k,j] + d_{i,k} + d_{k,j} \right), & i < j. \end{cases}$

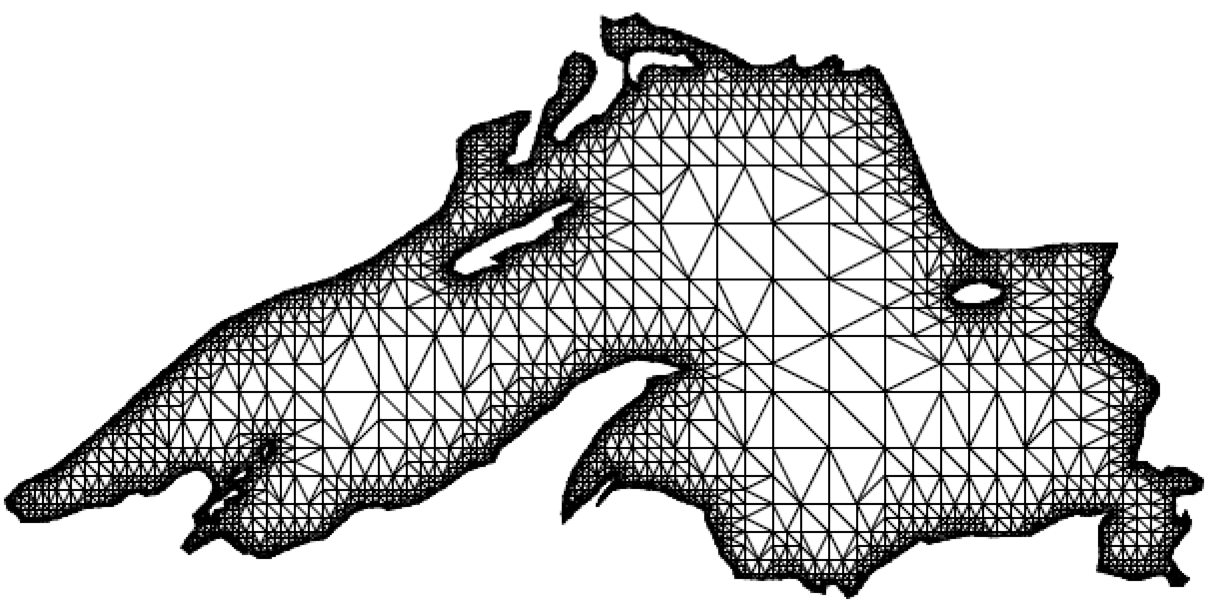
MWT of a Convex Polygon

```
MinWeightTriangulation(P)
for i ← 0 to n do
    w[i, i] ← 0
    w[i, i + 1] ← 0
    w[i, i + 2] ← 0
for m ← 3 to n do
    for i ← 0 to n − m
        j ← i + m − 1
        w[i, j] ← ∞
        for k ← i+1 to j − 1 do
            q ← w[i, k] + w[k, j] + d(i, k) + d(k, j)
            if q < w[i, j] then
                w[i, j] ← q
                e[i, j] ← k
return e
```



The MWT of a convex polygon can be computed in $O(n^3)$ time using $O(n^2)$ memory.

Mesh Generation and Refinement



Planar Straight-Line Graph

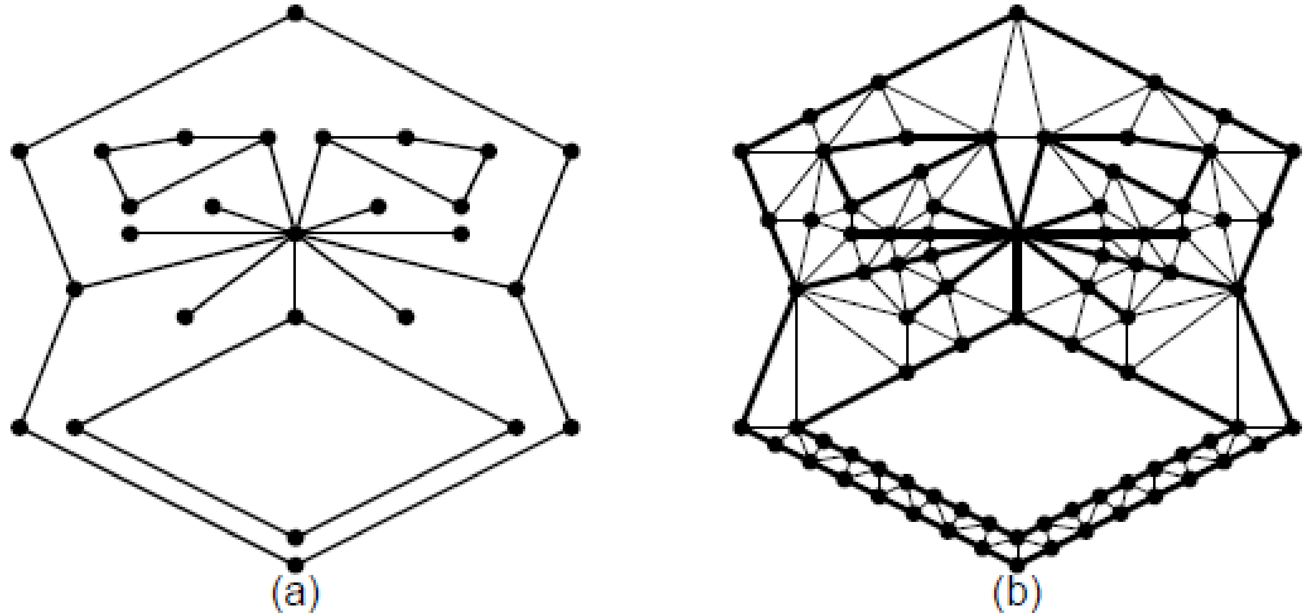
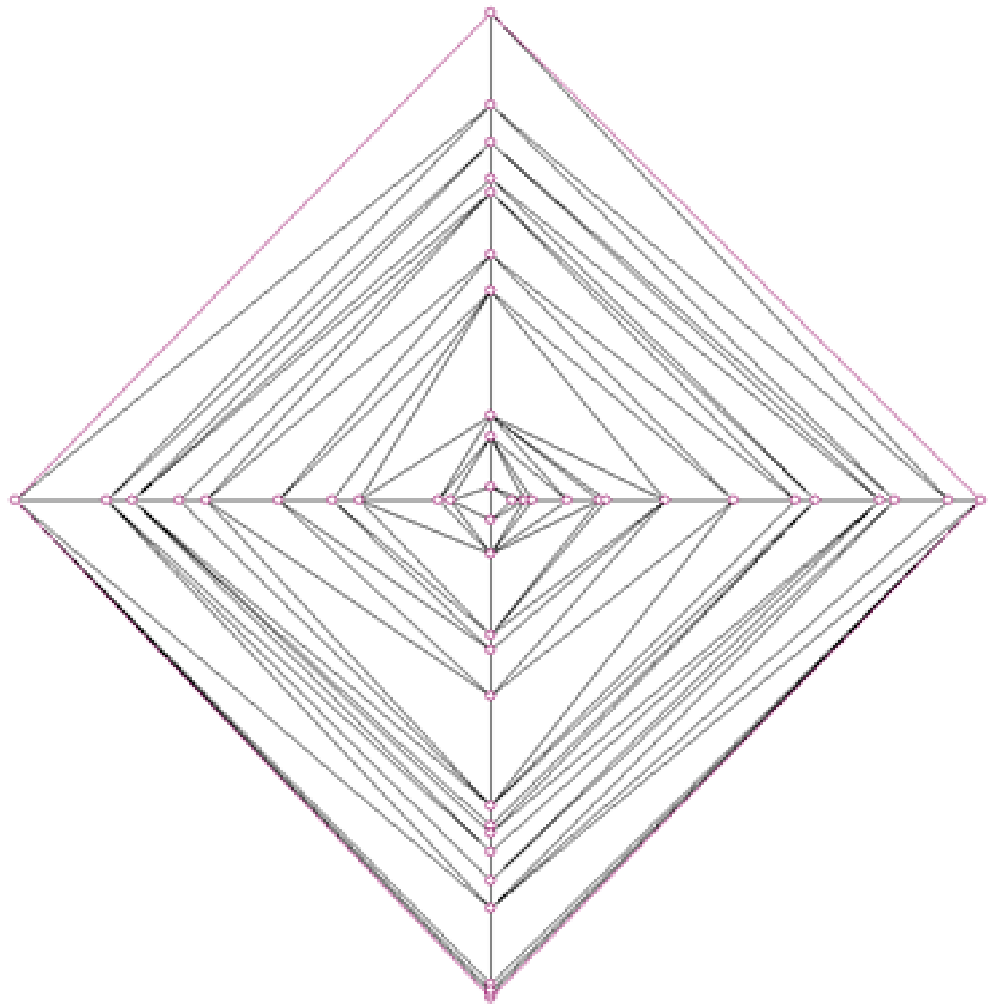


Figure 1: A PSLG and a mesh generated by Ruppert's Delaunay refinement algorithm.

Delaunay Triangulation

Sometimes can be extremely bad

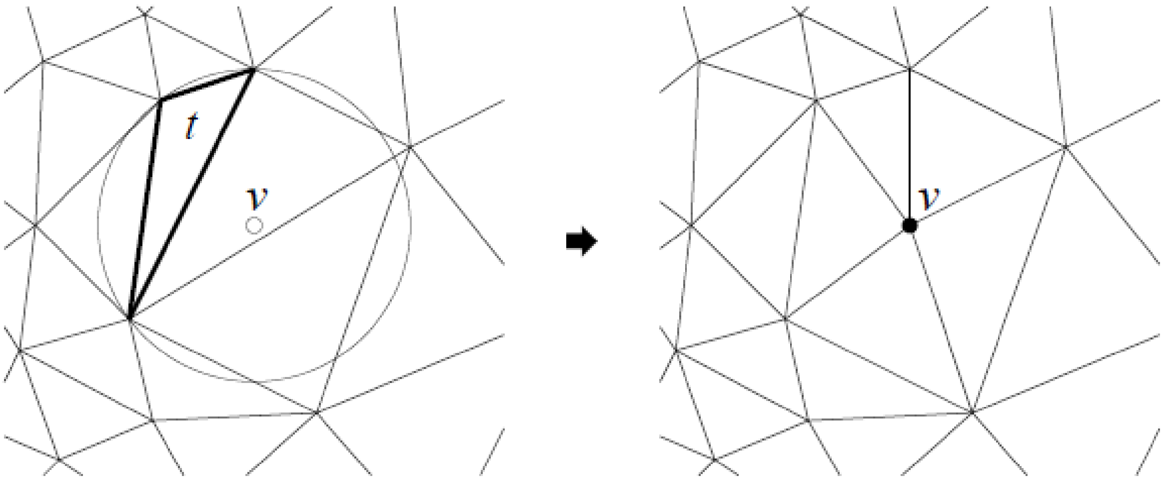


The Idea

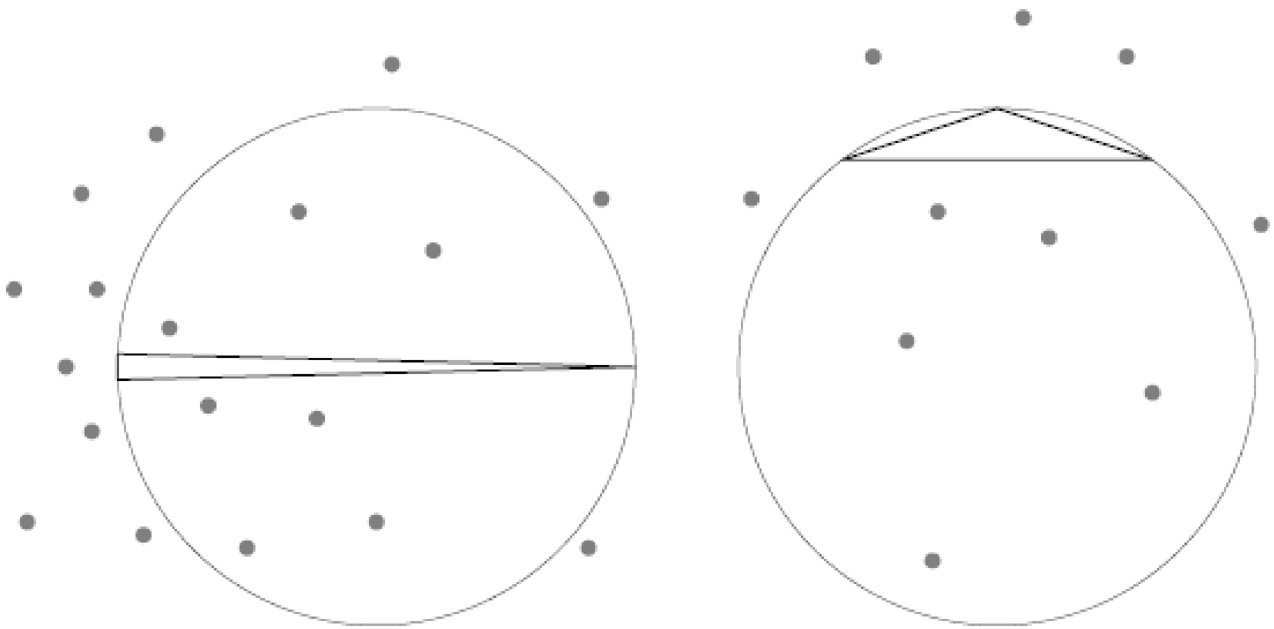
- Insert Steiner points at the circumcenters to destroy the **bad** triangles.
- A triangle is bad if it holds

$$\frac{r}{l} \geq B,$$

where r is the circumradius, l is the minimum edge length, B is some threshold.



Types of Bad Triangles



Needle Cap

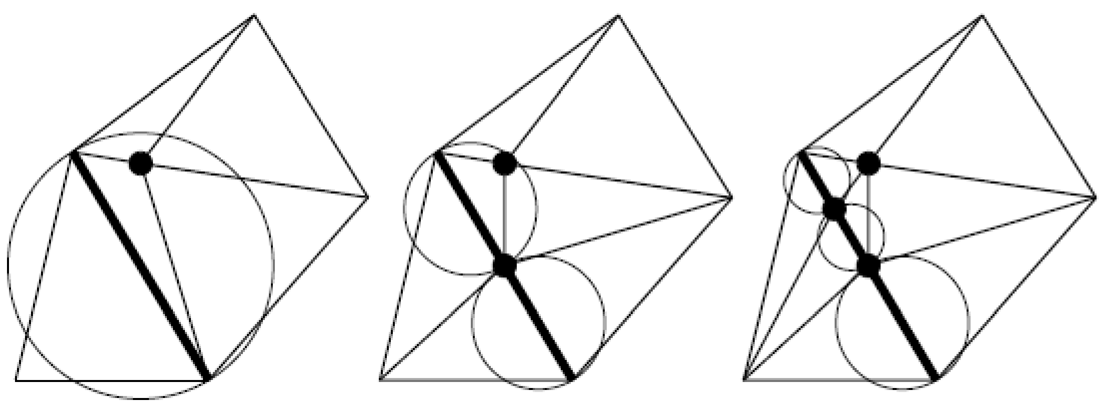
Thresholds

- The **Ruppert** Algorithm:
  - $B = \sqrt{2}$
  - Angles of the output triangles are between $20.7°$ and $138.6°$

- The **Chew** algorithm:
  - $B = 1$
  - Angles of the output triangles are between $30°$ and $120°$

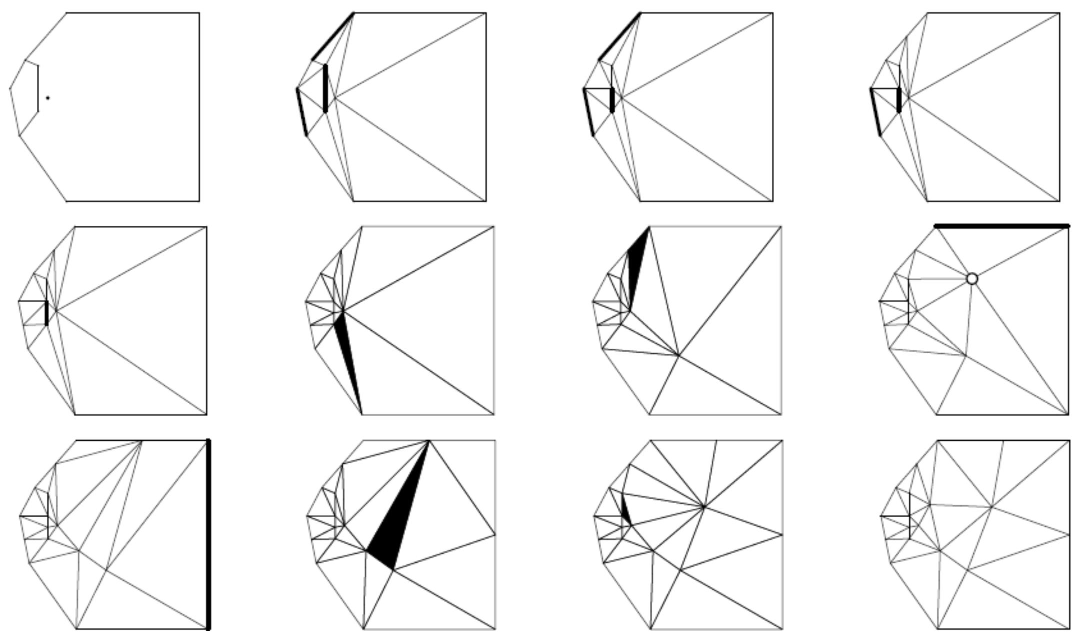
The Ruppert Algorithm

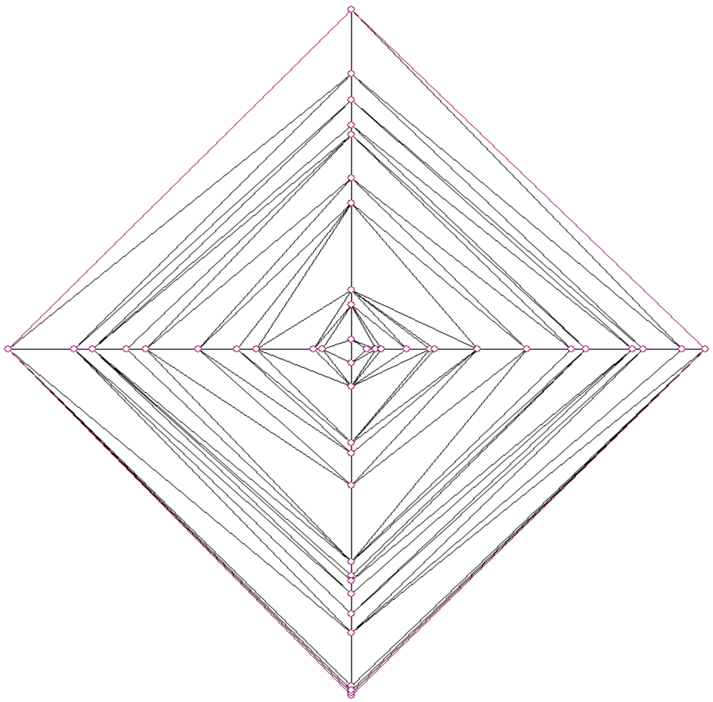
1. A vertex encroaches an edge if it is inside or on its diameter circle. An edge encroached by some vertex **is divided into two:**

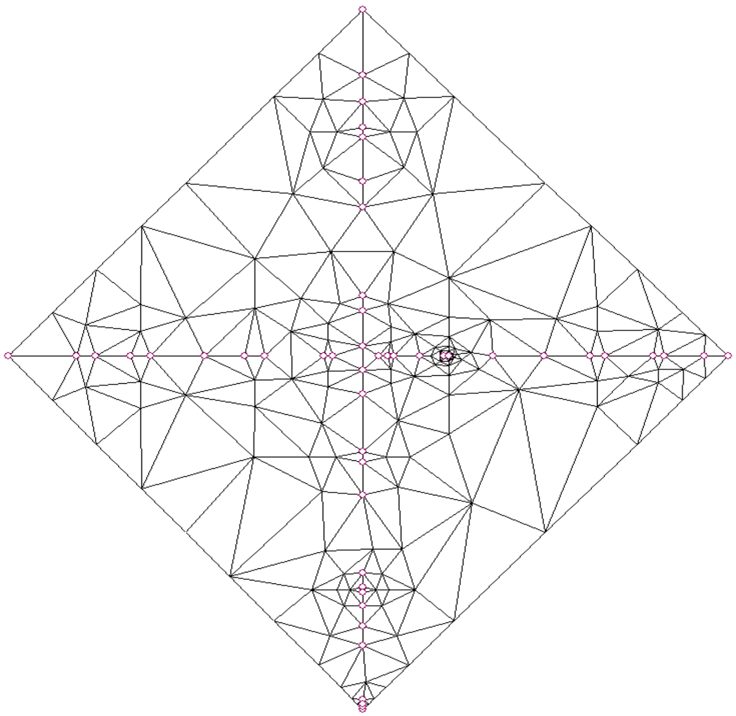


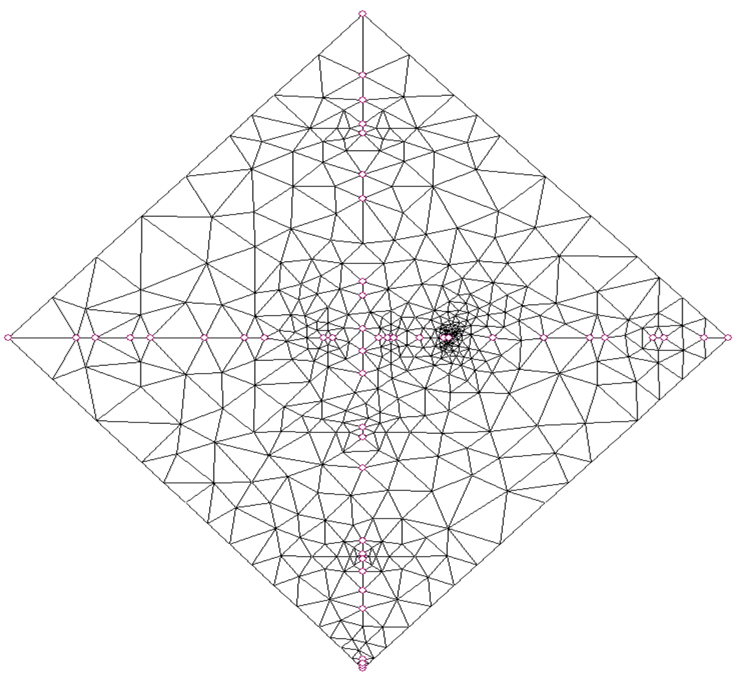
The Ruppert Algorithm

2. If no encroachment left, destroy each bad triangle by insertion of a new vertex v at its circumcenter, making a prior check (step 3):
3. If the new vertex v encroaches some edge e then cancel the insertion of v and divide e into two.



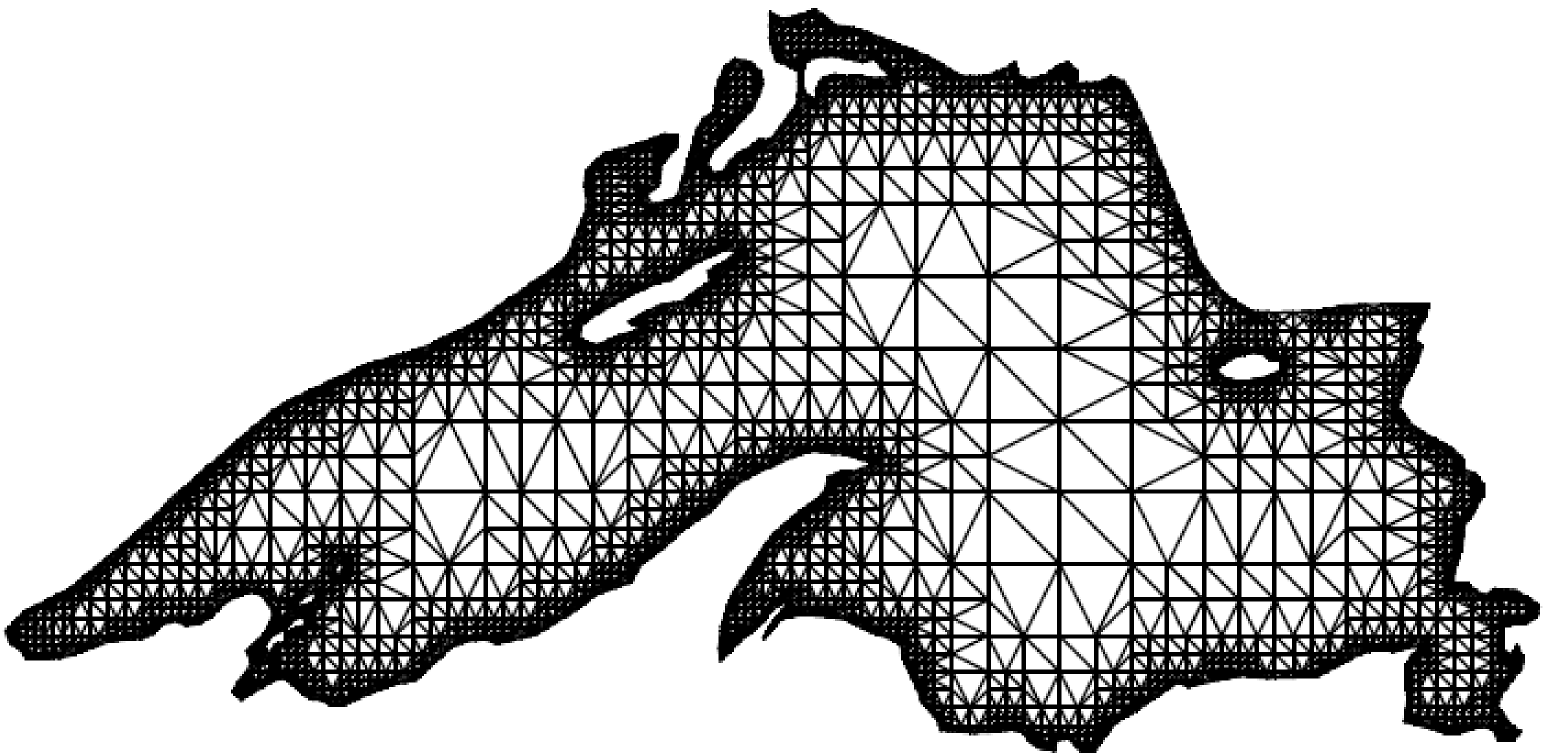






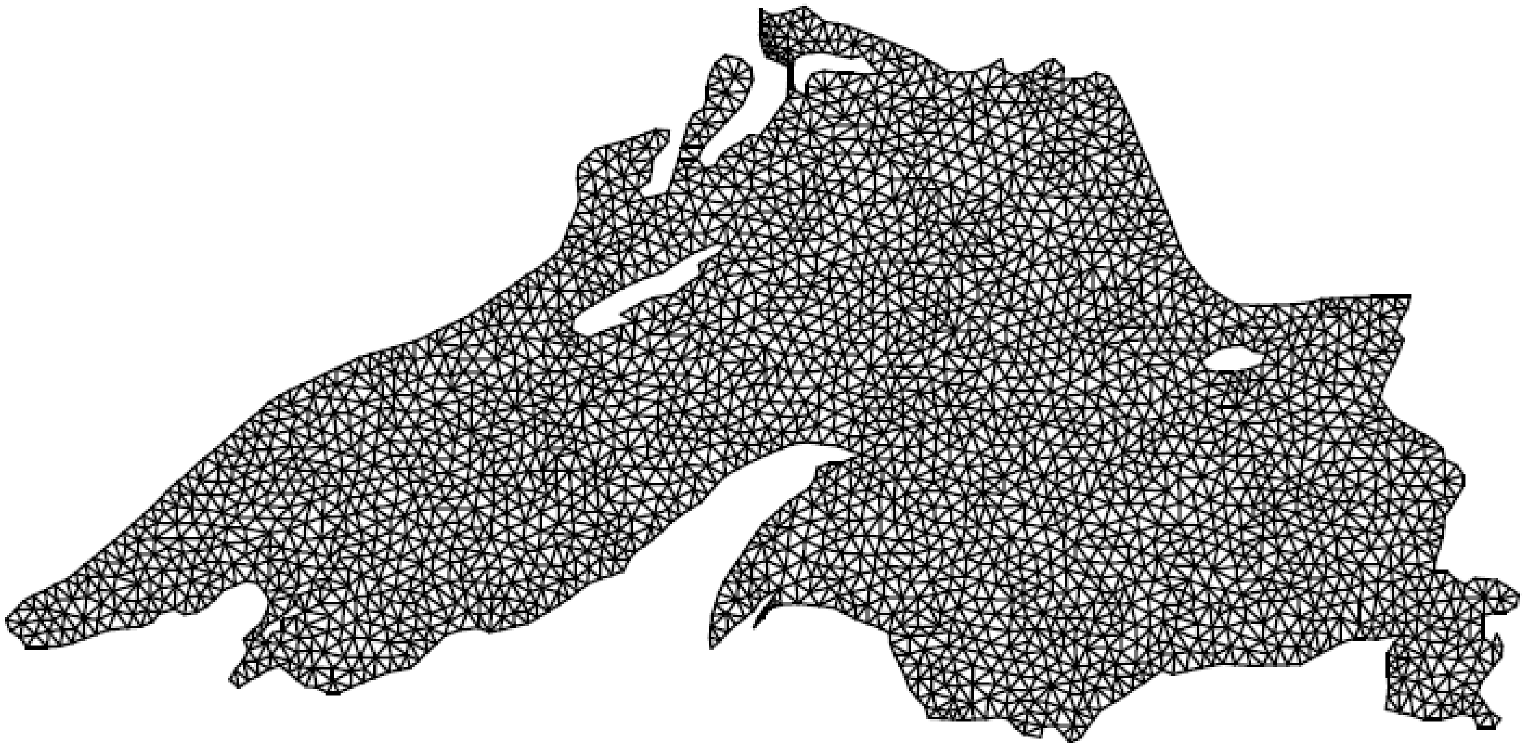
Few Classical Approaches

- Bern–Eppstein–Gilbert, quadtrees:



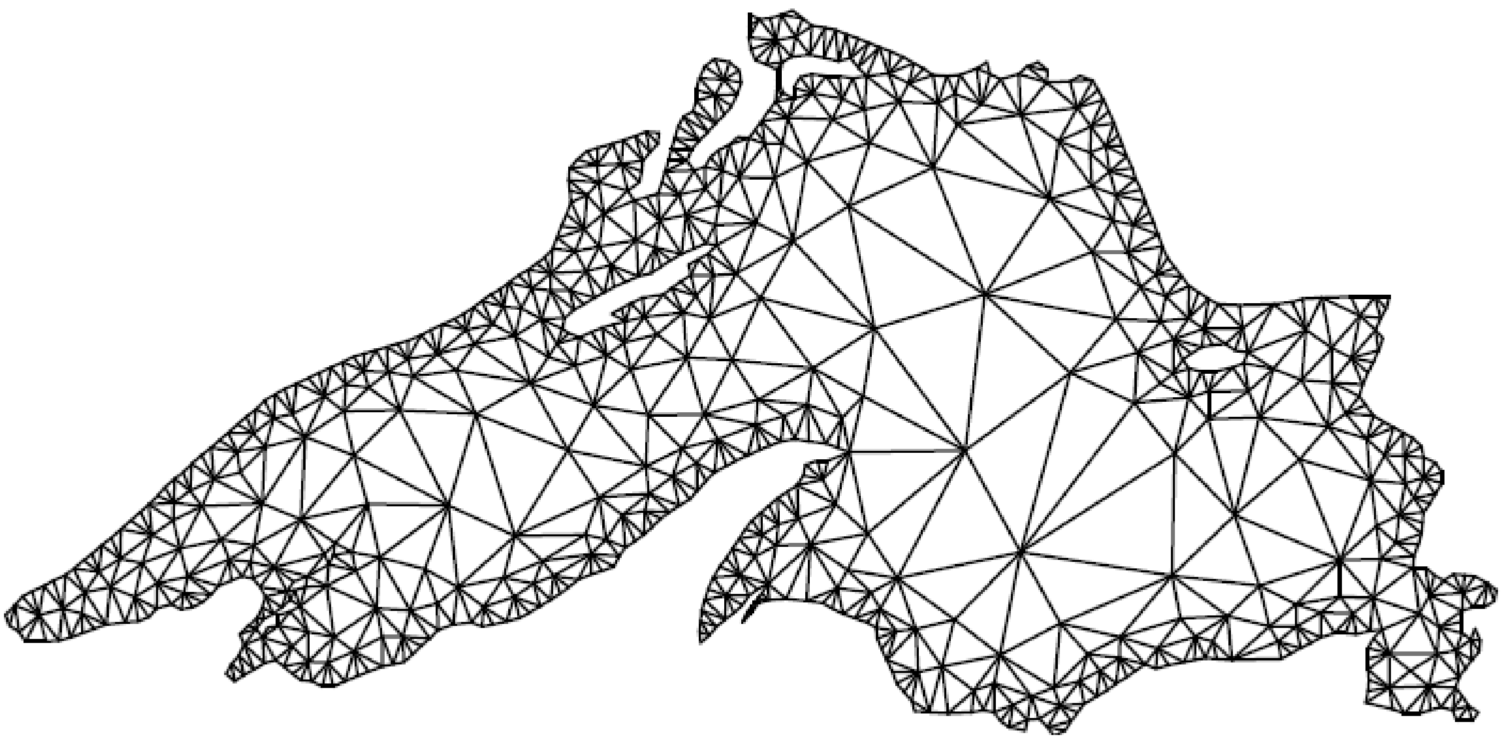
Few Classical Approaches

- The Chew algorithm:

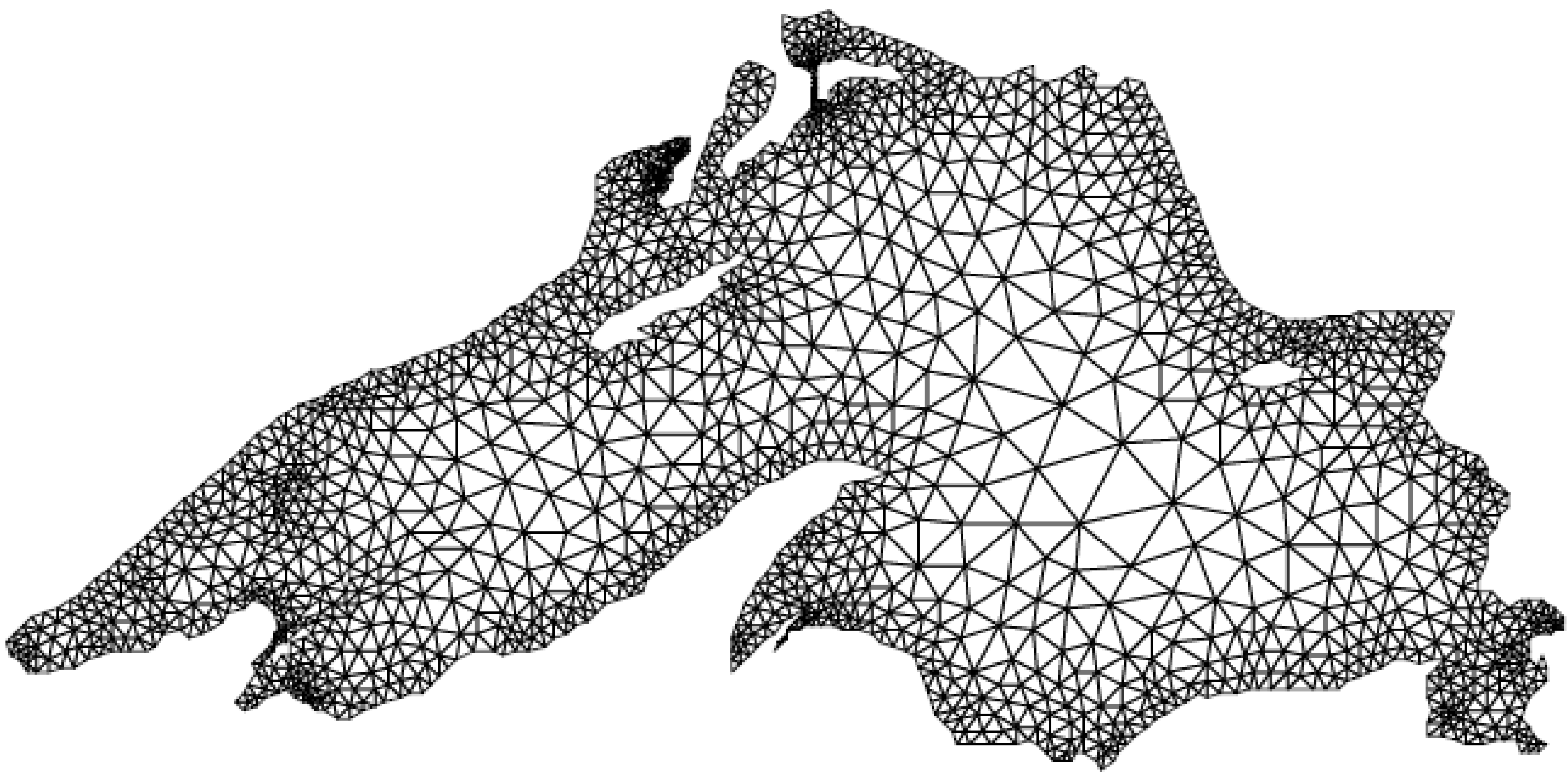


Few Classical Approaches

- The Ruppert algorithm:



Balancing



Questions?