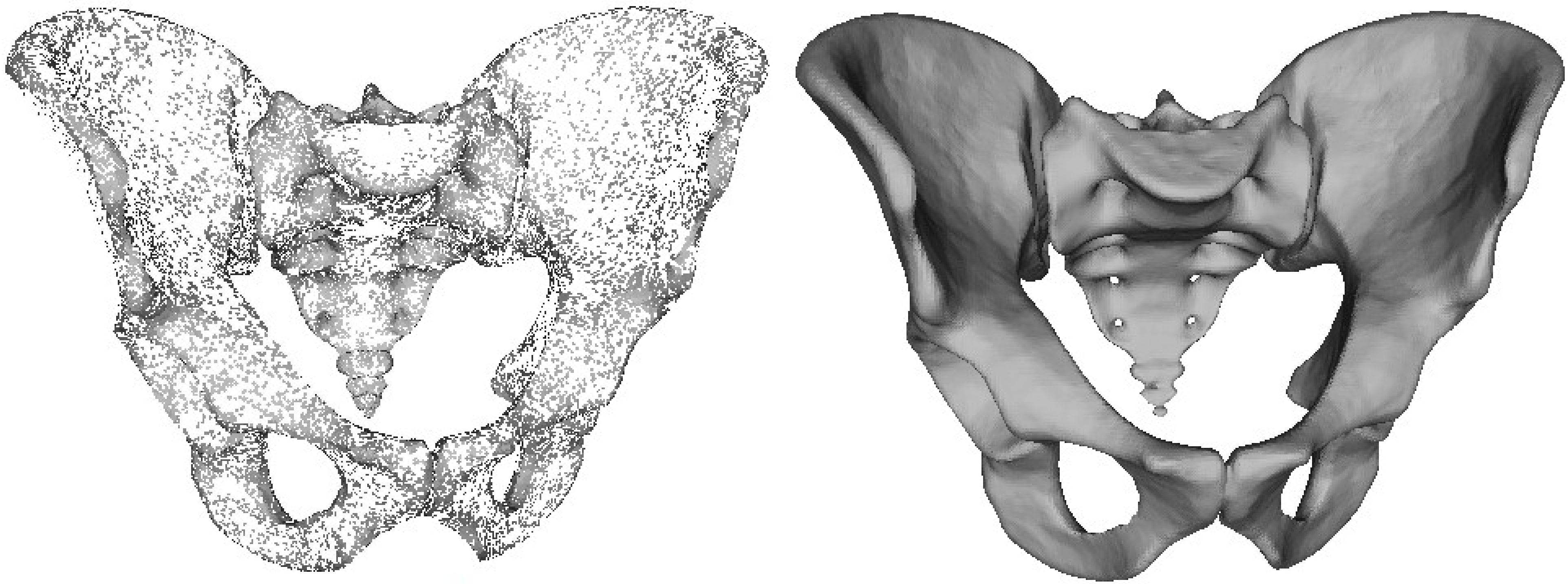
Misha Kazhdan

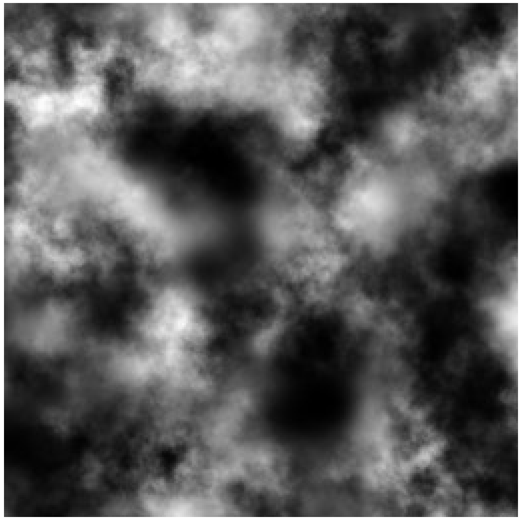


Mesh Reconstruction from Points

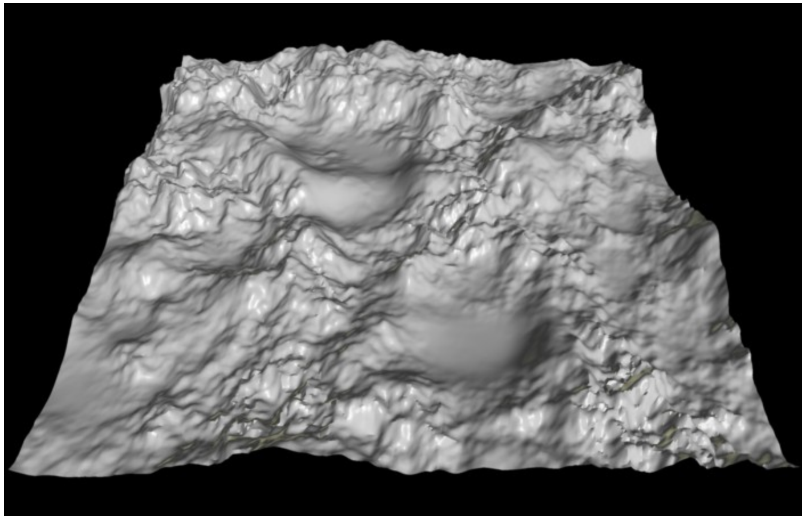
Siddhartha Chaudhuri http://www.cse.iitb.ac.in/~cs749

Points to Meshes

- Simplest: convert a heightfield to a terrain



Regular 2D grid,
pixel value = height
above base plane

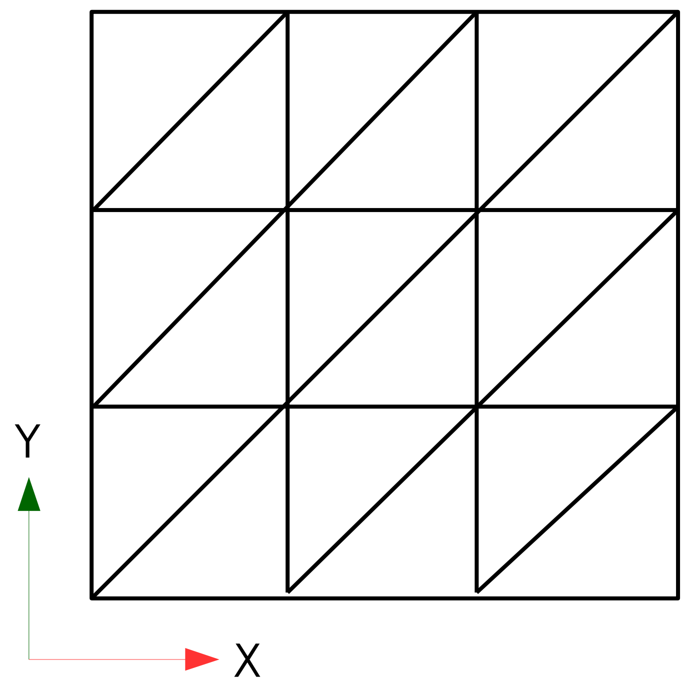


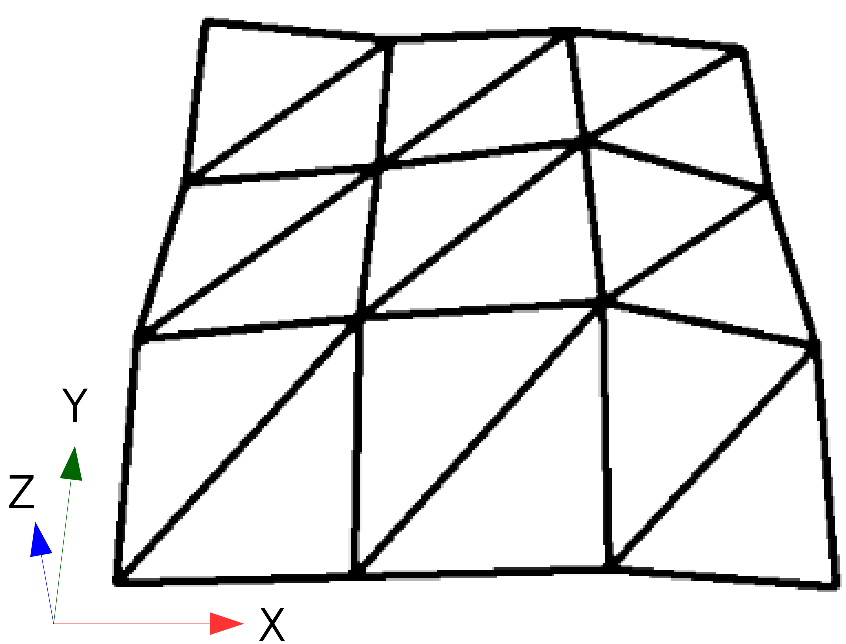
Same heightfield as a
terrain mesh

Wikipedia

Points to Meshes

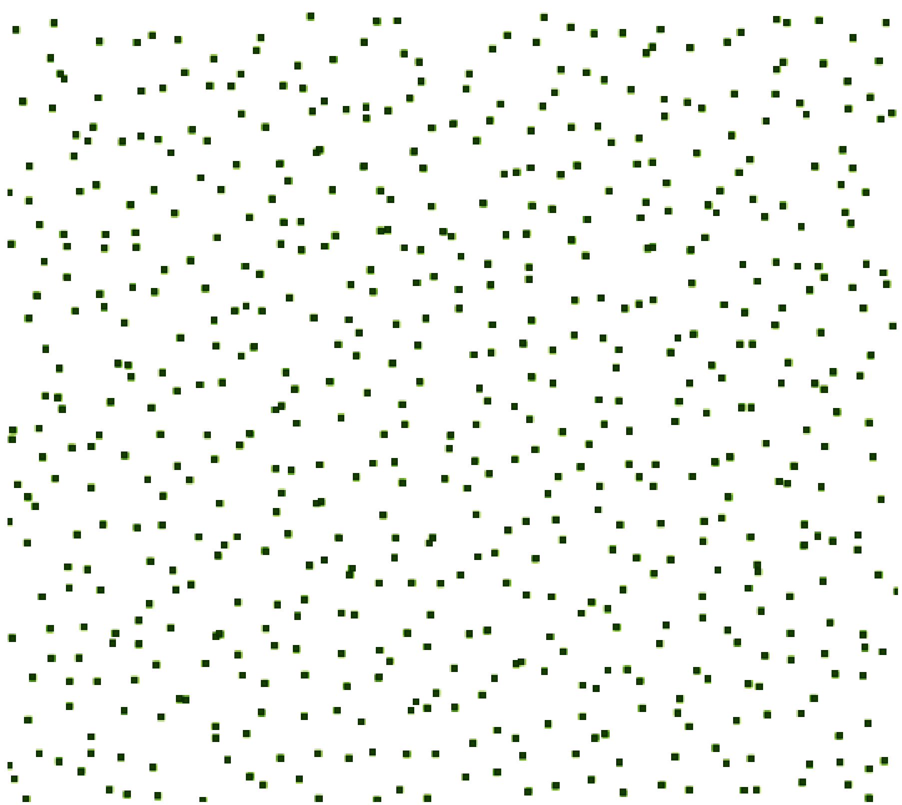
- Image defines height value at each vertex
- Raise vertices of uniform XY grid to these heights along Z



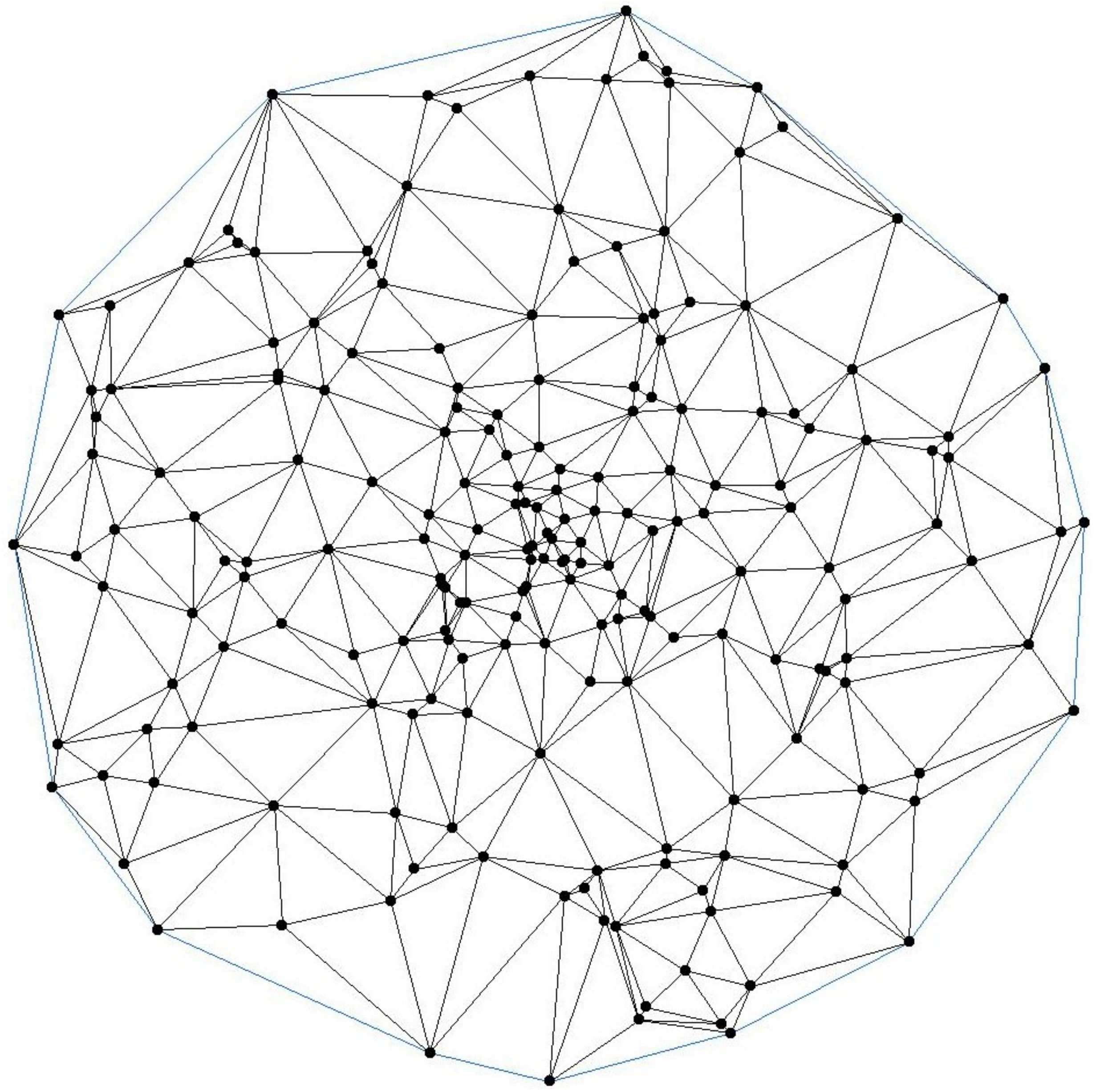


Thought for the Day #1

What if the heights are not sampled along a grid?

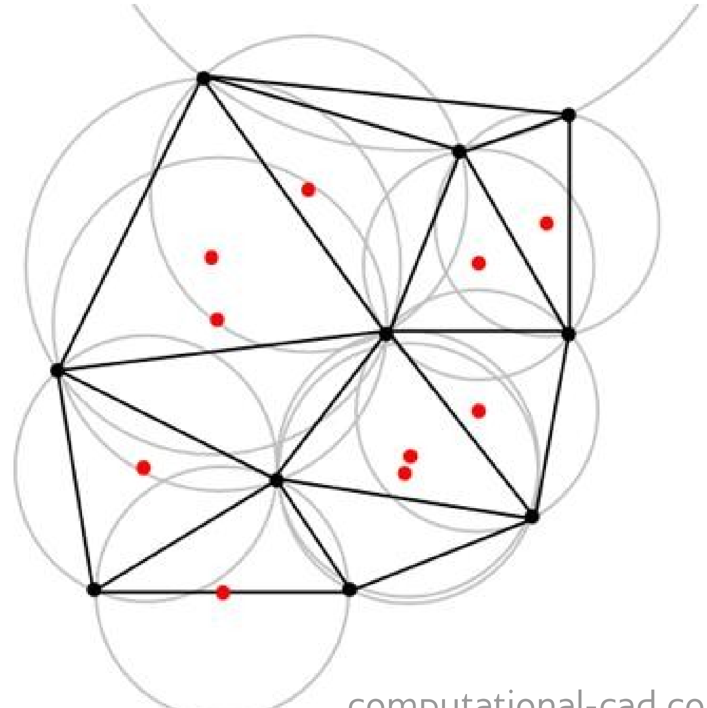


Delaunay Triangulation



Delaunay Triangulation

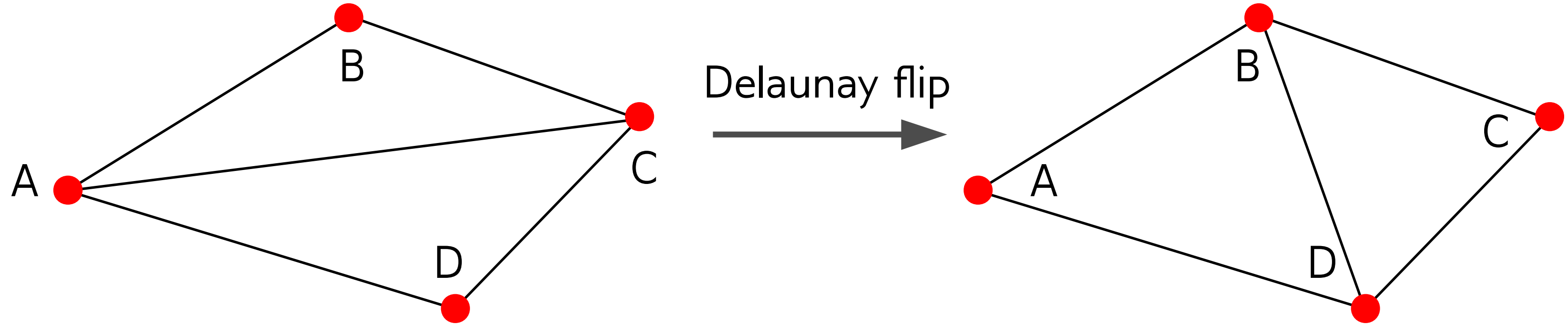
- Used to generate a planar base mesh whose vertices are a set of 2D points P
- A triangulation is **Delaunay** if no point in P lies within the circumcircle of any triangle
  - It also happens to maximize the minimum angle of any triangle, which is why it's useful



computational-cad.com

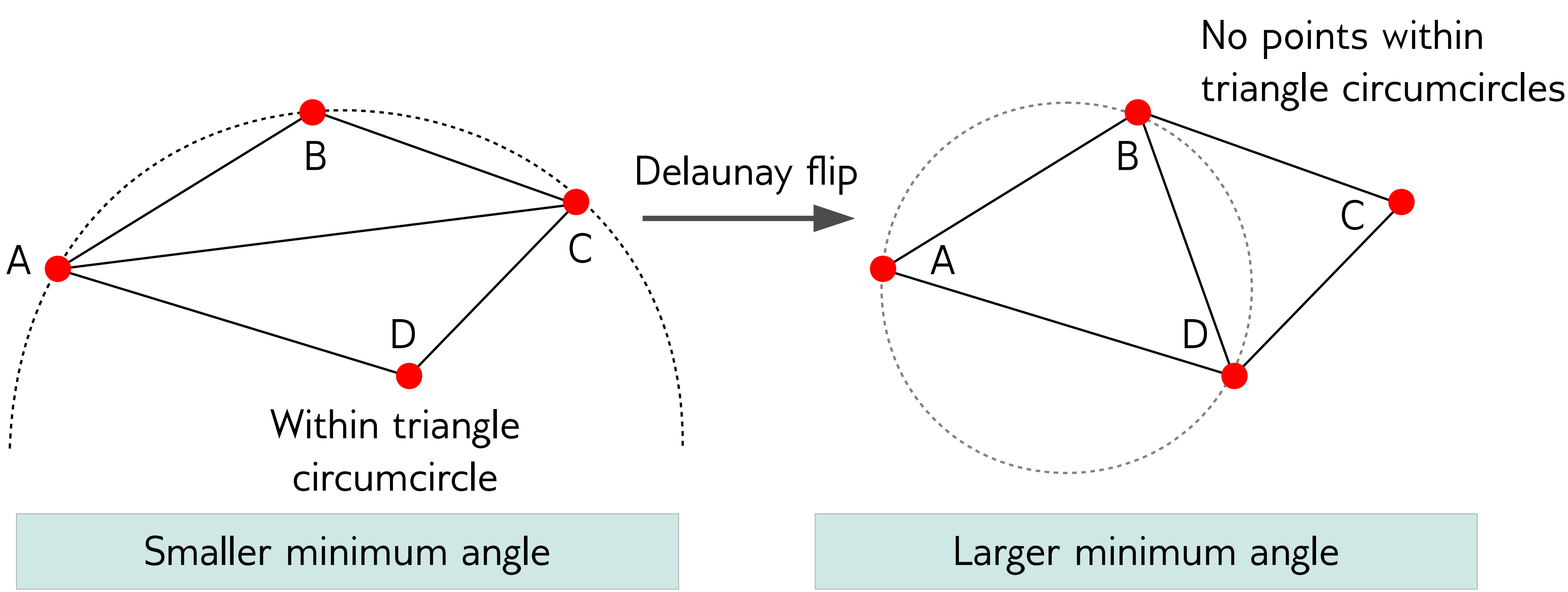
Computing the Delaunay triangulation

- Many different algorithms
- One approach: start with *any* triangulation of the points, and successively flip edges if the minimum of the 6 angles increases
  - Guaranteed to converge (since minimum angle increases)



Why this works

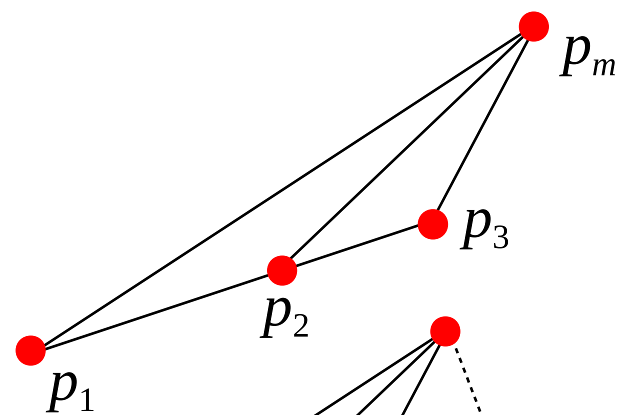
Lemma: Assume the vertices of a convex quadrilateral don't all lie on the same circle. The quad can be split into two triangles in two different ways (by drawing one of the two diagonals). Then one way satisfies the circumcircle property, the other does not. The former also has the larger minimum angle.

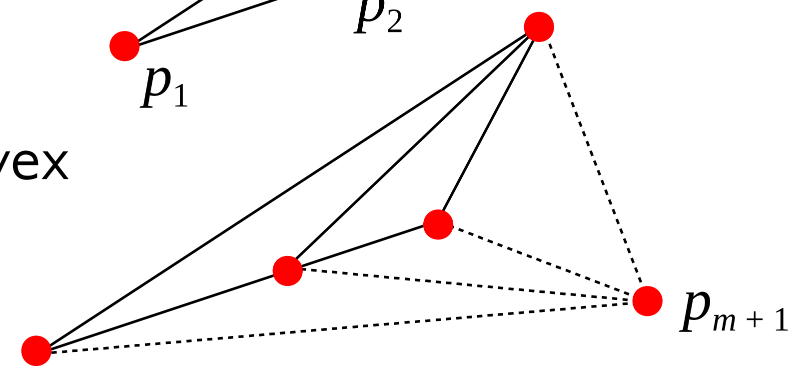


Lawson's Flip Algorithm

- Start with *any* triangulation of P...
  - Here's one way (the **scan triangulation method**)
    - Sort the points $p_1 \dots p_n$ (not all collinear and $n \geq 3$) lexicographically: $(x, y) < (u, v)$ iff $x < u$ or $(x = u$ and $y < v)$
    - Let m be the minimum index such that $p_1 \ldots p_\mathrm{m}$ are not collinear
    - Triangulate $p_1 \dots p_m$ by connecting p_m to $p_1 \dots p_{m-1}$
    - Now, for each additional point p_i, connect it to all points on the convex hull of $p_1 \dots p_{i-1}$ that it "sees"

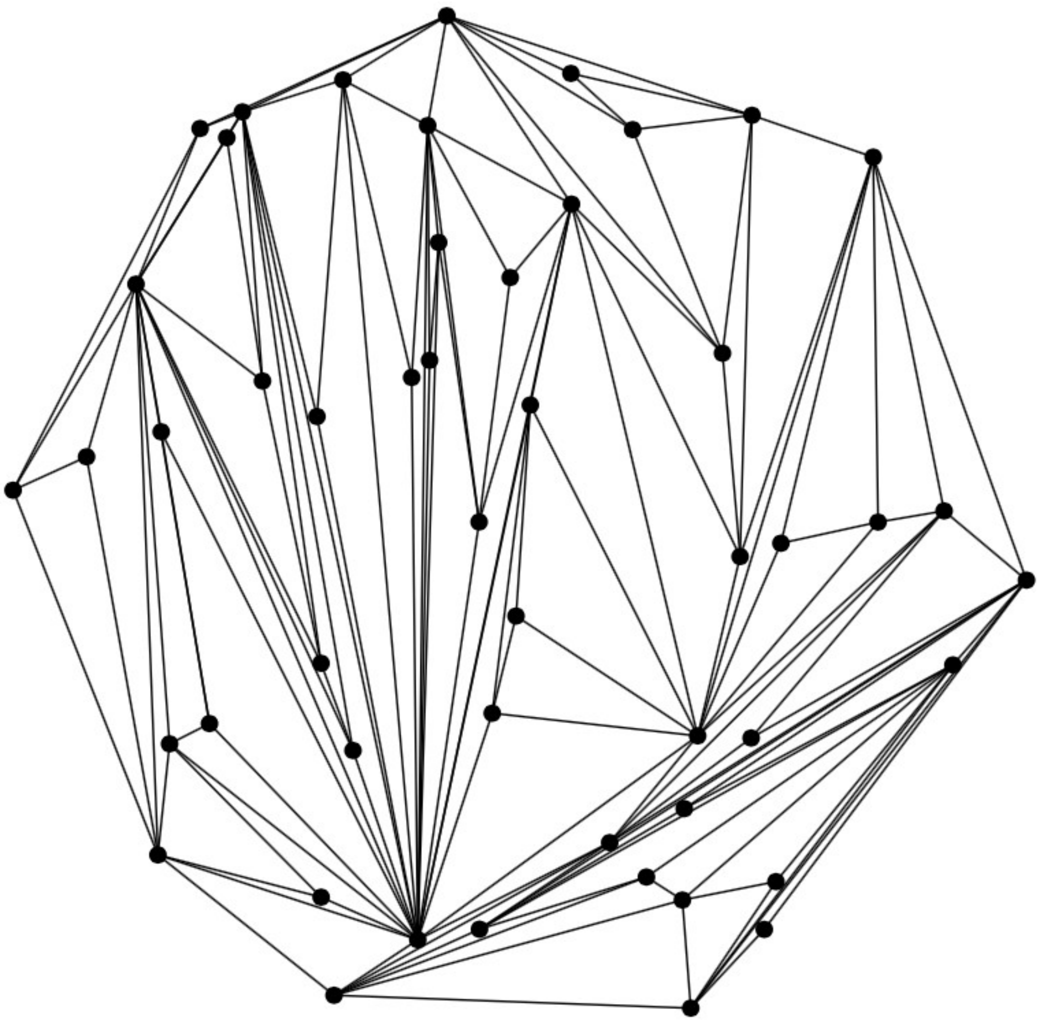
Can run in $O(n \log n)$ time





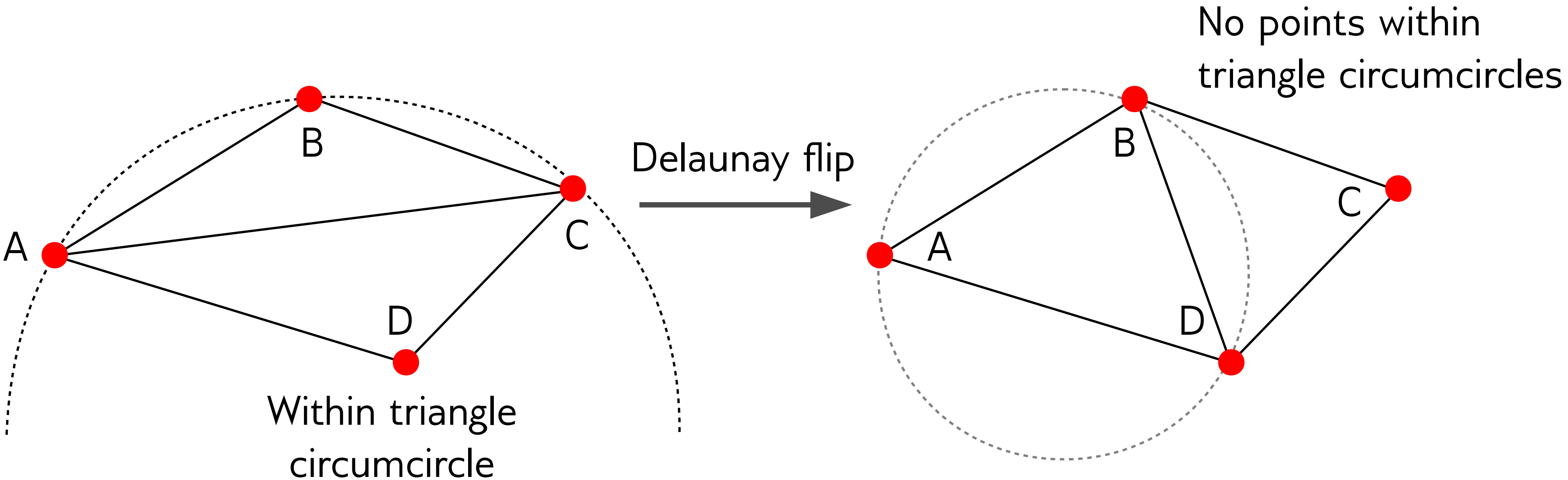
Lawson's Flip Algorithm

- Scan triangulation produces horrible triangulations



Lawson's Flip Algorithm

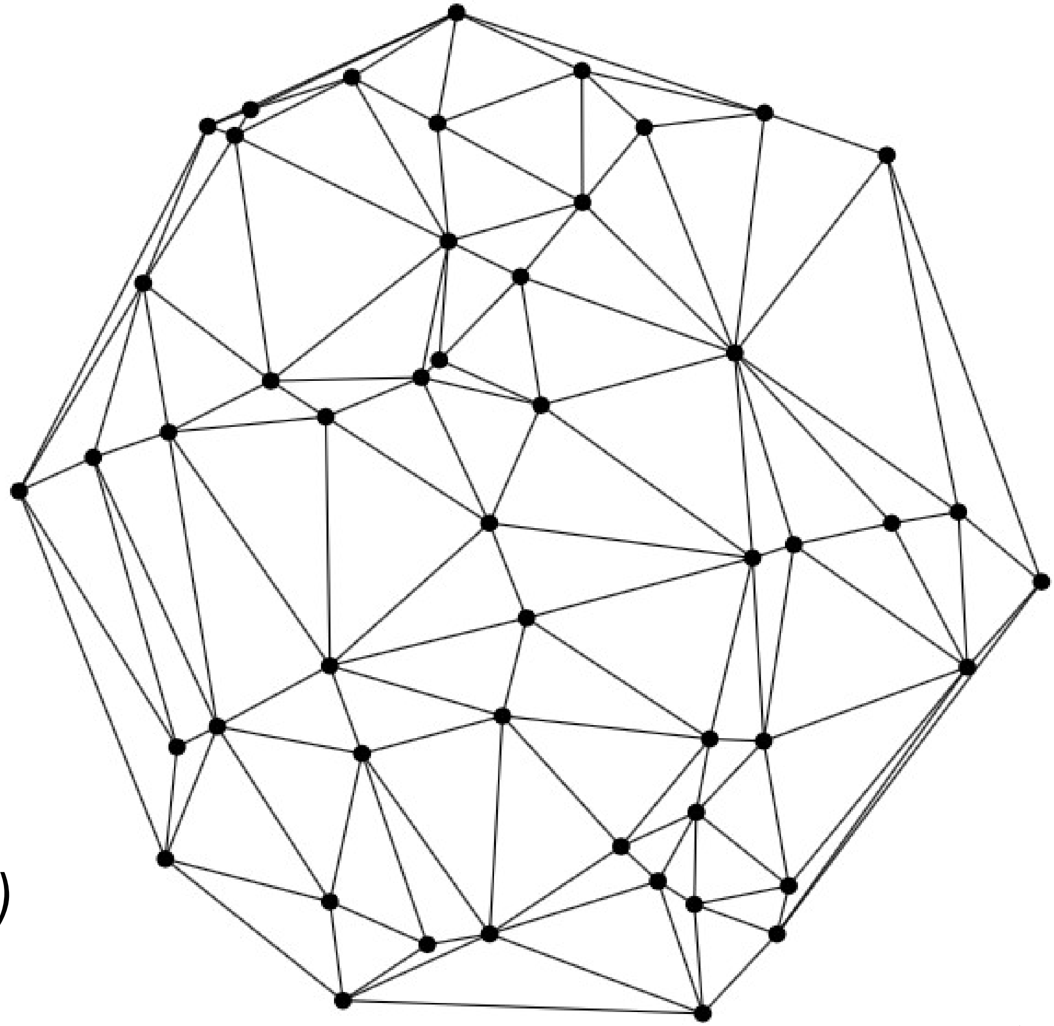
- Iteratively apply Delaunay flips to improve the triangulation
  - Identify two adjacent triangles
  - Flip the shared edge if the minimum of the 6 angles increases (i.e. if the circumcircle property can be restored)



Lawson's Flip Algorithm

- The result is a Delaunay triangulation of P
  - It turns out that any Delaunay triangulation of P has the same minimum angle
  - … and if no four points lie on a circle, the Delaunay triangulation is unique

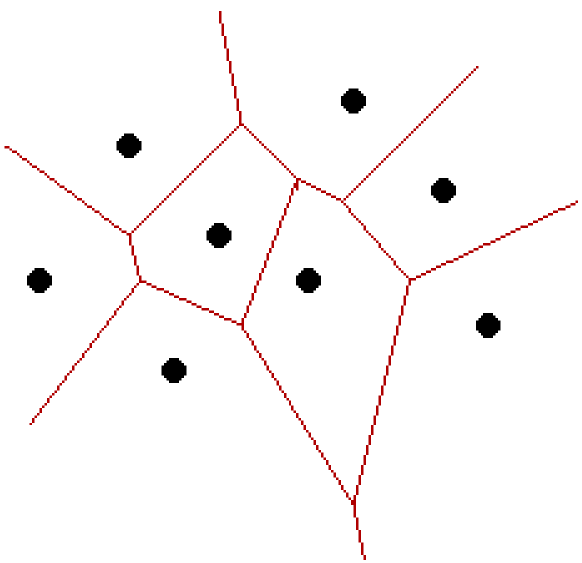
  (see handouts for proof)



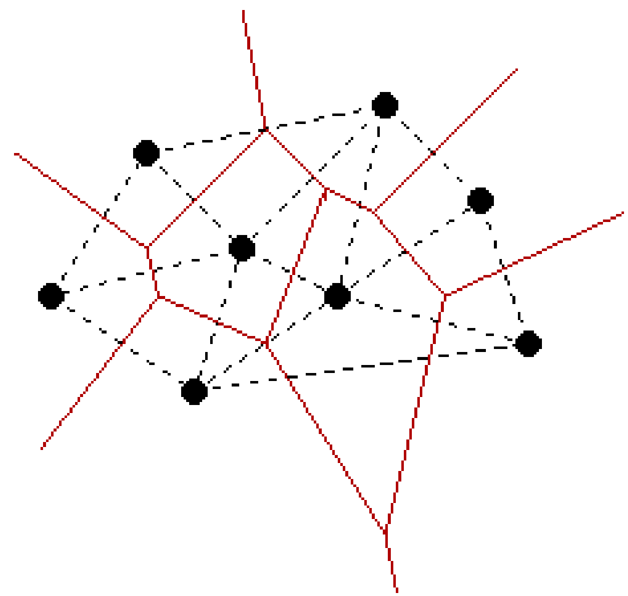
Emo Welzl

Delaunay and Voronoi

- The Delaunay triangulation is the **dual** of the Voronoi diagram of P



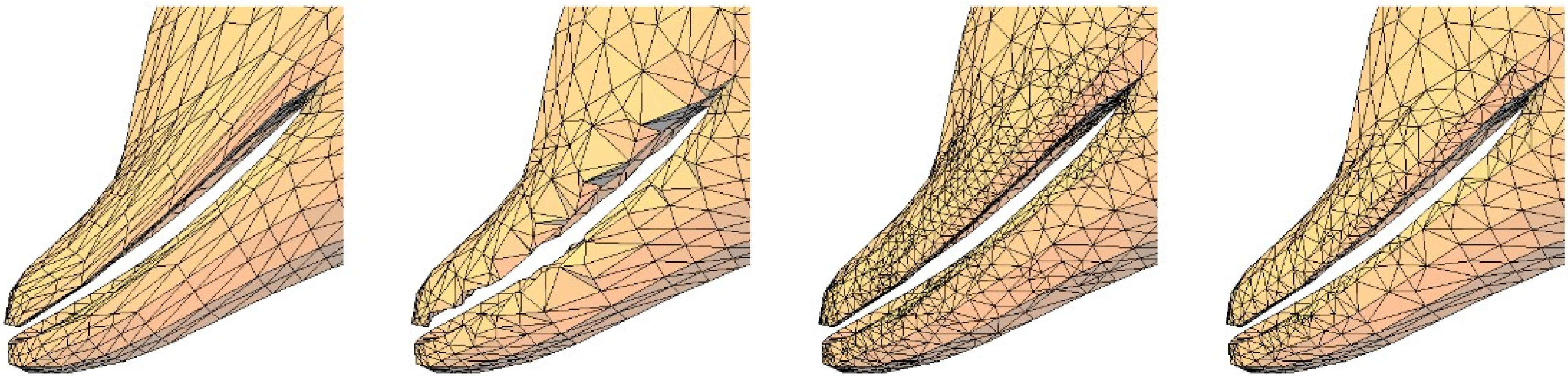
Voronoi diagram – each cell consists of points nearer the cell center than to any other point in P



(Superimposed) Delaunay triangulation. For each vertex of the Voronoi diagram, there is a Delaunay face. Two faces are adjacent if the corresponding vertices are connected by a Voronoi edge.

2D surfaces embedded in 3D

- Delaunay triangulations are defined in 2D planes
- But the Delaunay idea can be extended to 2D manifold surfaces embedded in 3D



(a) Original: $\#V = 6,002$. (b) $\#V = 6,002$; $\varepsilon = 0.385\%$. (c) $\#V = 15,560$; $\varepsilon = 0.000\%$. (d) $\#V = 7,509$; $\varepsilon = 0.071\%$.

Original mesh

Delaunay edge flipping

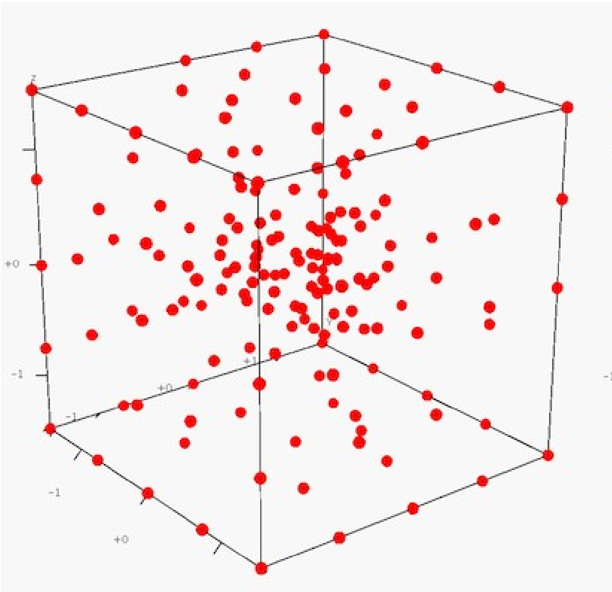
Adding vertices *after* flipping edges, to preserve geometry

Adding vertices *while* flipping edges

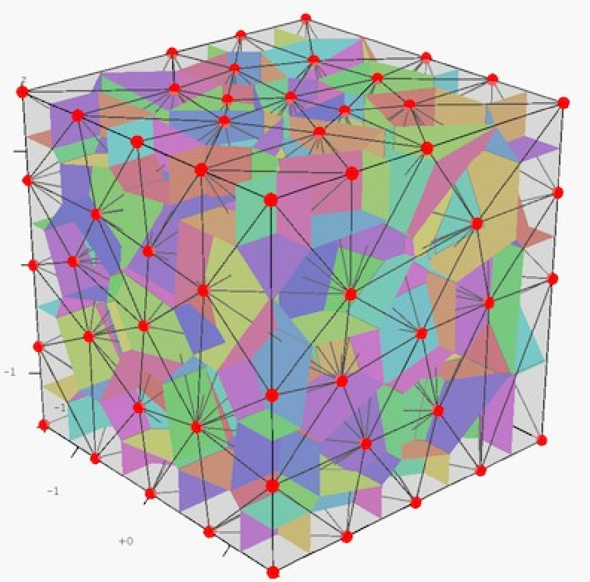
Dyer et al., "Delaunay Mesh Refinement", SGP 2007

3D volumes

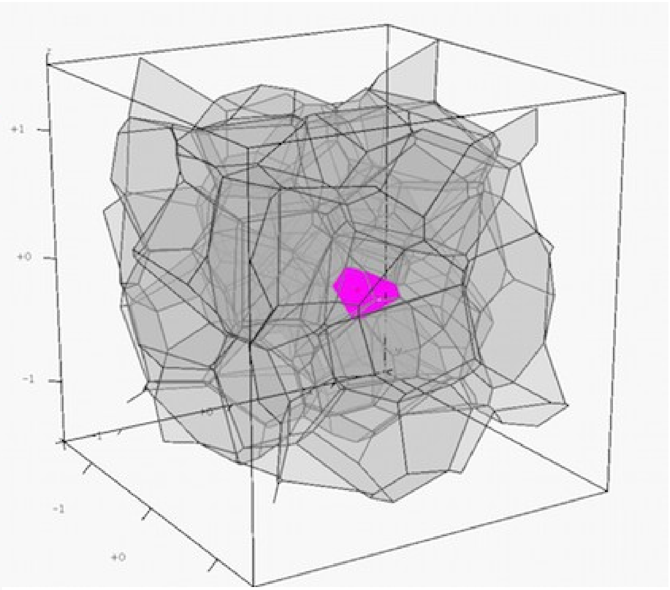
- The true 3D (volumetric) analogue of a Delaunay triangulation is a **Delaunay Tetrahedralization**



Input 3D points



Delaunay tetrahedralization (shown via edges); Voronoi faces (shown randomly colored)

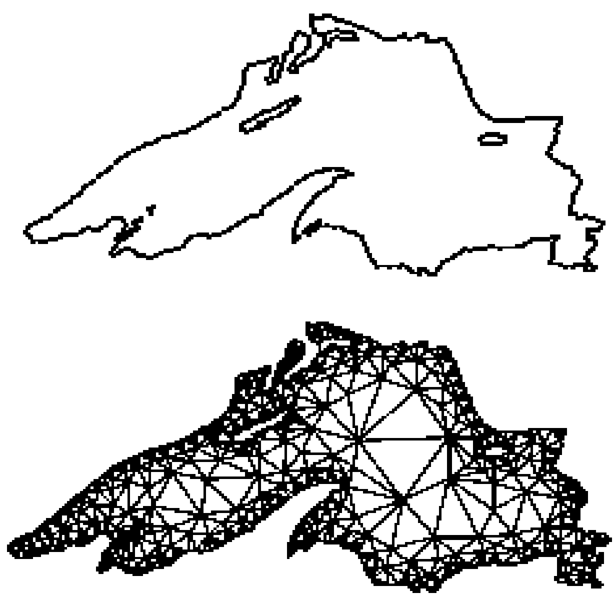


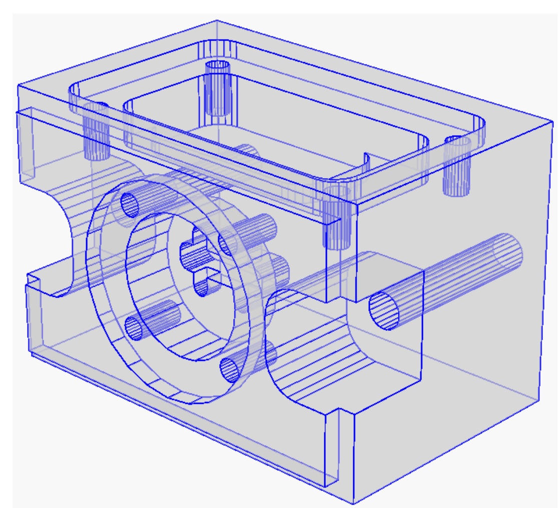
3D Voronoi diagram (only bounded cells shown)

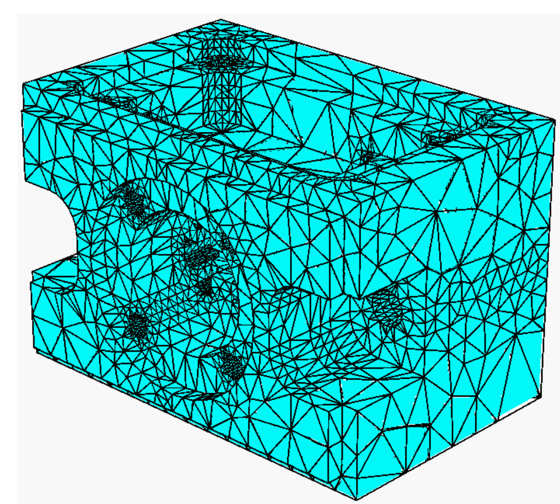
Hang Si, "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator", 2015

Boundary-Conforming Delaunay

- To generate meshes for finite element analysis and similar methods, we often want to preserve the boundary while adding vertices to the interior
    - Maximize minimum angle for high quality results



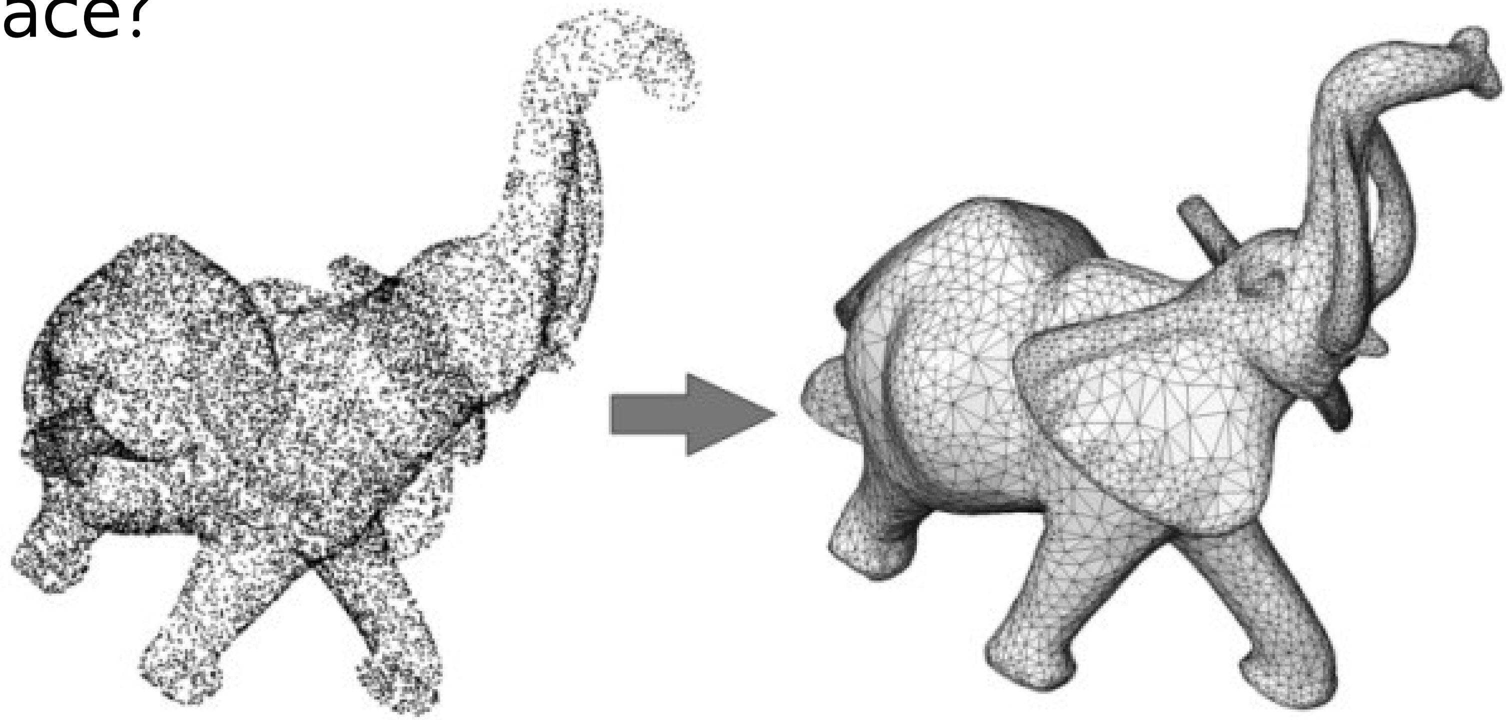




Jonathan Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator”, 1996
Hang Si, “TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator”, 2015

Poisson Surface Reconstruction

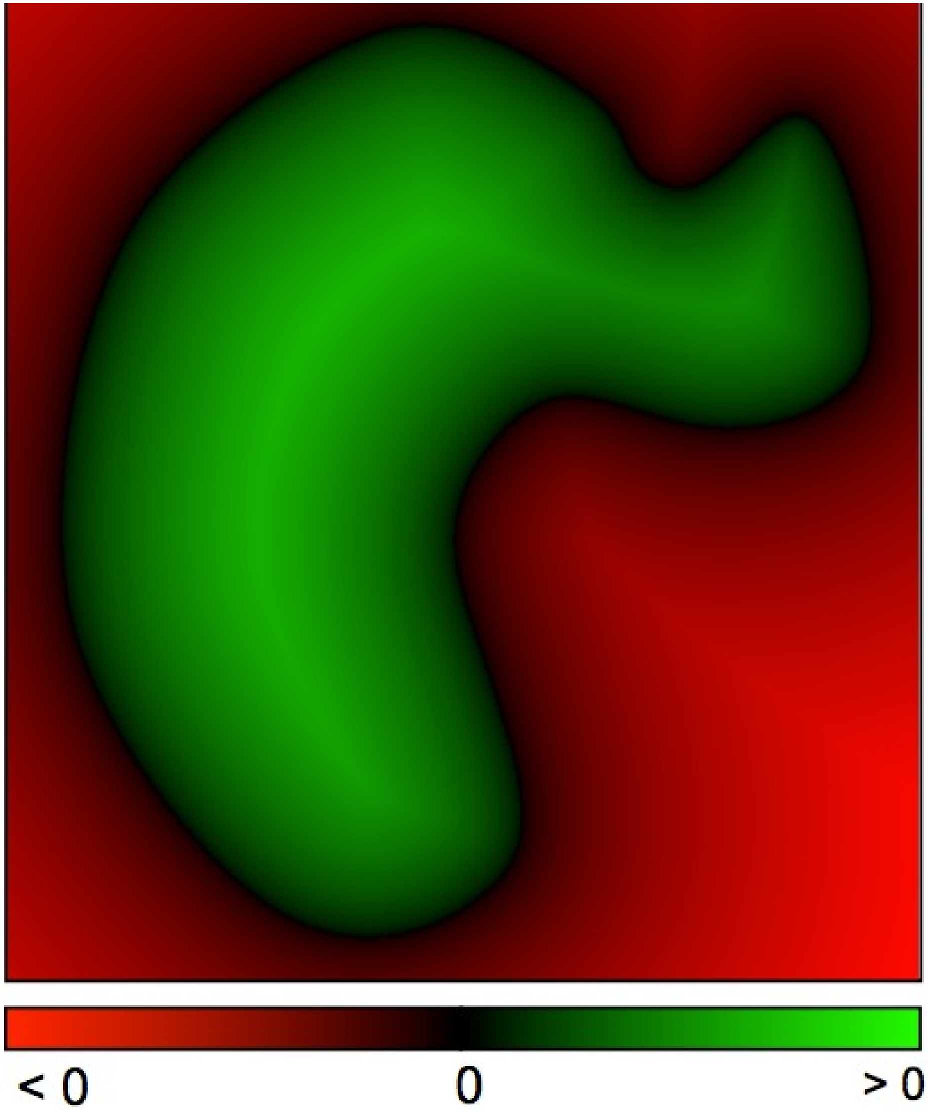
- Triangulation is sensitive to noise, sampling pattern, omissions etc
- Can we more robustly recover the underlying surface?



Slides adapted from Kazhdan, Bolitho and Hoppe

Implicit Function Approach

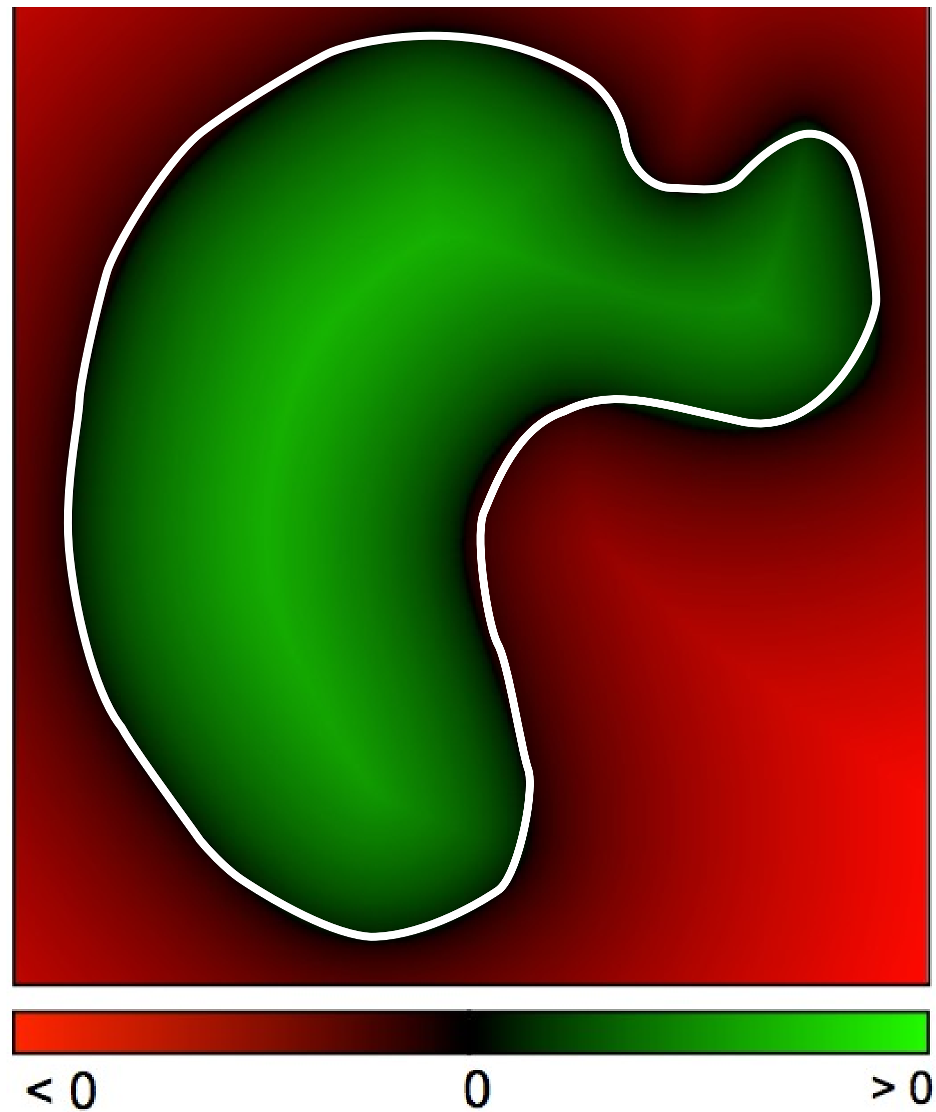
- Define a function with positive values inside the model and negative values outside



Implicit Function Approach

- Define a function with positive values inside the model and negative values outside

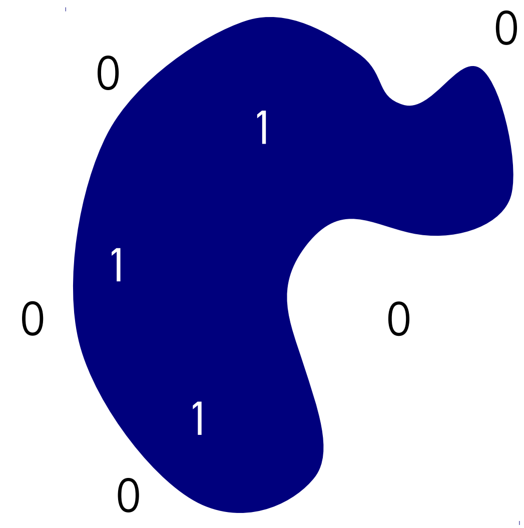
- Extract the zero-set



Key Idea

- Reconstruct the surface of the model by solving for the indicator function of the shape

$$\chi_M(p) = \begin{cases} 1 & \text{if } p \in M \\ 0 & \text{if } p \notin M \end{cases}$$



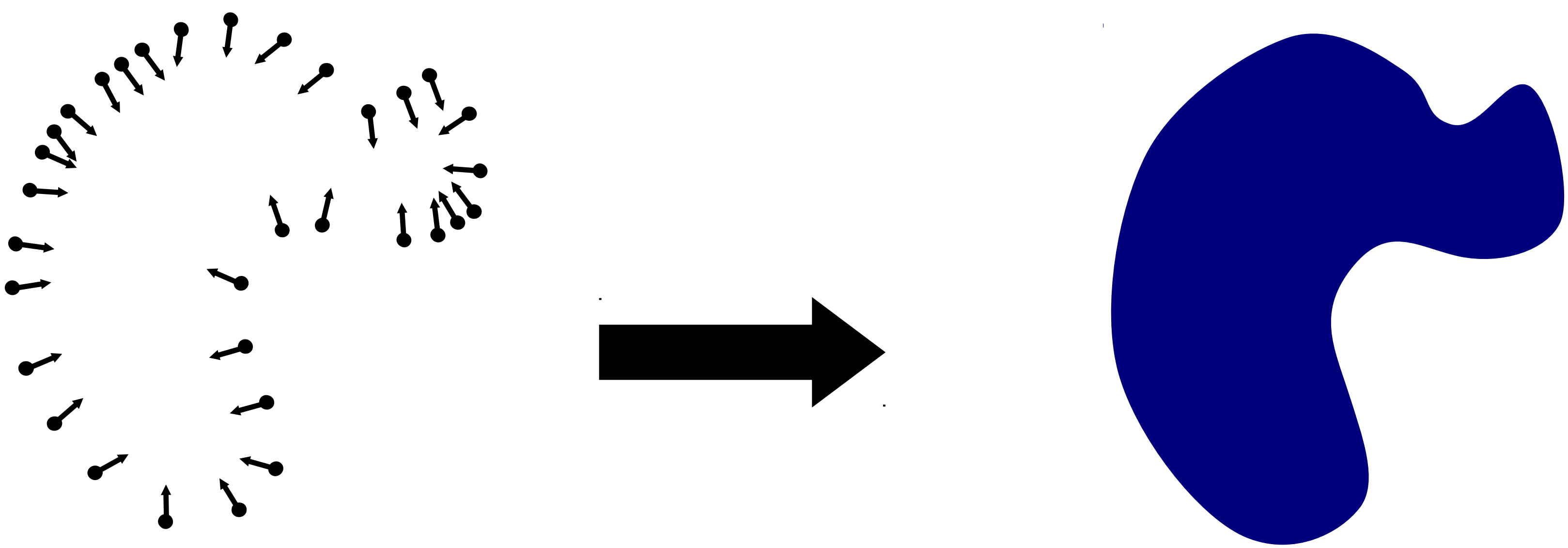
Indicator function

χ_M

In practice, we define the indicator function to be -1/2 outside the shape and 1/2 inside, so that the surface is the zero level set. We also smooth the function a little, so that the zero set is well defined.

Challenge

- How to construct the indicator function?



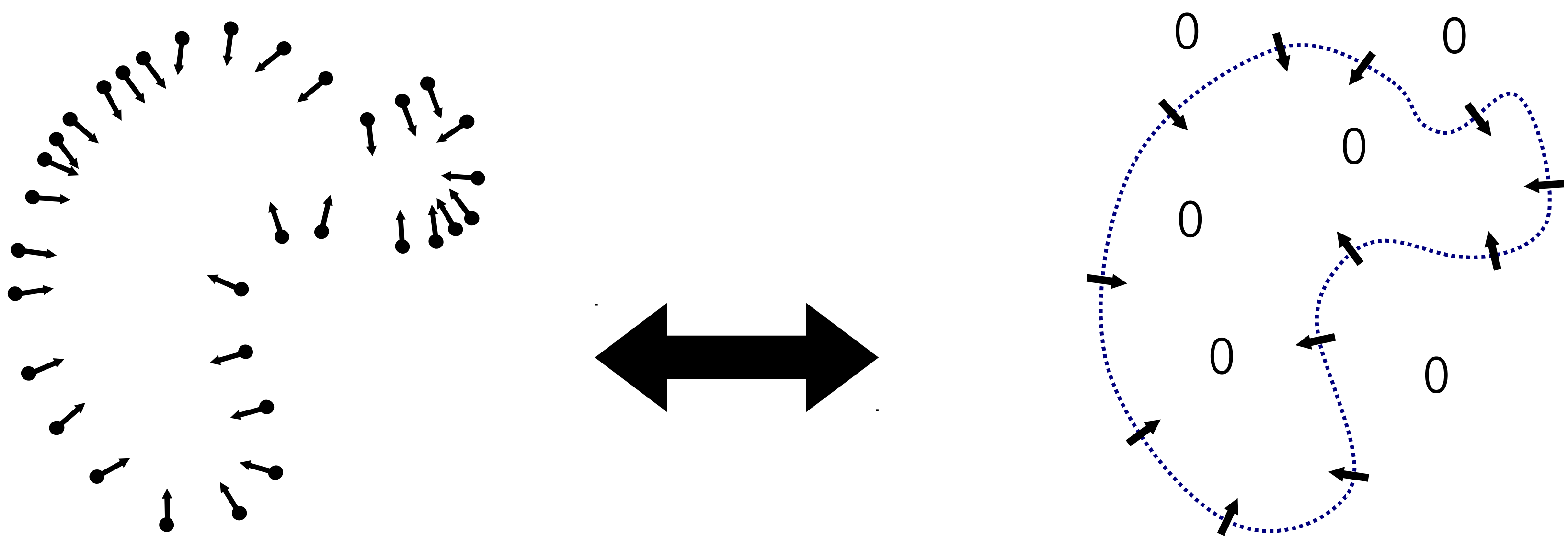
Oriented points

Indicator function

χ_M

Gradient Relationship

- There is a relationship between the normal field at the shape boundary, and the gradient of the (smoothed) indicator function



Oriented points

Indicator gradient

$\nabla \chi_M$

Integration

- Represent the point normals by a vector field V
- Find the function χ whose gradient best approximates V

$$\min_{\chi} \|\nabla \chi - V\|^2$$

Gradient $\nabla \chi = \left(\frac{\partial \chi}{\partial x}, \frac{\partial \chi}{\partial y}, \frac{\partial \chi}{\partial z}\right)$

Squared norm of function
F over domain Ω

$$\|F\|^2 = \int_{\Omega} \langle F(\mathbf{x}), F(\mathbf{x}) \rangle \, \mathrm{d}\sigma$$

Integration as a Poisson Problem

- Represent the point normals by a vector field V
- Find the function χ whose gradient best approximates V

$$\min_{\chi} \|\nabla \chi - V\|^2$$

Laplacian $\Delta \chi = \frac{\partial^2 \chi}{\partial x^2} + \frac{\partial^2 \chi}{\partial y^2} + \frac{\partial^2 \chi}{\partial z^2}$

Divergence $\nabla \cdot V = \frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}$

- Applying the divergence operator, we can transform this into a Poisson problem:

$$\nabla \cdot (\nabla \chi) = \nabla \cdot V \quad \Leftrightarrow \quad \Delta \chi = \nabla \cdot V$$

Link to Linear Least Squares

- Need to solve set of equations $A\mathbf{x} = \mathbf{b}$ in a least squares sense

$$\text{minimize } ||\mathbf{r}||^2 = ||\mathbf{b} - A\mathbf{x}||^2$$

- The directional derivative in direction $\delta\mathbf{x}$ is

$$\nabla||\mathbf{r}||^2 \,.\, \delta\mathbf{x} = 2\delta\mathbf{x}^{\mathrm{T}}(A^{\mathrm{T}}\mathbf{b} - A^{\mathrm{T}}A\mathbf{x})$$

- The minimum is achieved when all directional derivatives are zero, giving the normal equations

$$A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$$

- **Thought for the Day:** Compare this equation to the Poisson equation