

Proximity Graphs

Autores: Héctor Saib Maravillo Gómez, Diego Villarreal de la Cerda, Heriberto Espino Montelongo
Correo: heriberto.espinomo@udlap.mx

Dada una nube de puntos: $P = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$, se busca construir un grafo donde se satisfice una regla de proximidad.

Objetivo: ofrecer una API para crear y comparar graficos de proximidad con utilidades de análisis y visualización. Cada construcción se define mediante una regla de proximidad con parámetros, lo que habilita barridos paramétricos y comparación de propiedades globales.

1. **Unit-Disk** $UDG(r)$.

$$\{i, j\} \in E \iff d_{ij} \leq r.$$

2. **k-Nearest Neighbor** k -**NNG** (simétrico).

$$i \rightarrow j \iff x_j \in \text{NN}_k(x_i), \quad \{i, j\} \in E \iff (i \rightarrow j) \text{ o } (j \rightarrow i).$$

3. **Gabriel** (GG).

$$m_{ij} = \frac{x_i + x_j}{2}, \quad \{i, j\} \in E \iff \forall z \neq x_i, x_j: \|z - m_{ij}\| > \frac{d_{ij}}{2}.$$

4. **Relative Neighborhood** (RNG).

$$\mathcal{L}_{ij} = \{z: \max(\|z - x_i\|, \|z - x_j\|) < d_{ij}\}, \quad \{i, j\} \in E \iff \mathcal{L}_{ij} \cap (P \setminus \{x_i, x_j\}) = \emptyset.$$

5. **β -Skeleton** ($\beta \geq 1$).

$$\mathcal{L}_{ij}^{(\beta)} = B\left(x_i, \frac{\beta d_{ij}}{2}\right) \cap B\left(x_j, \frac{\beta d_{ij}}{2}\right), \quad \{i, j\} \in E \iff \mathcal{L}_{ij}^{(\beta)} \cap (P \setminus \{x_i, x_j\}) = \emptyset.$$

$$\beta = 1 \Rightarrow \text{GG}.$$

6. **σ -Graph** y **SIG**.

$$\delta(x_i) = \min_{y \neq x_i} \|x_i - y\|, \quad r_i = \sigma \delta(x_i),$$

$$\{i, j\} \in E \iff B(x_i, r_i) \cap B(x_j, r_j) \neq \emptyset.$$

7. **Stepping Stone** ($d \geq 1, k \geq 0$).

$$\{i, j\} \in E \iff \neg \exists z \neq x_i, x_j: \|x_i - z\|^d + \|z - x_j\|^d \leq \|x_i - x_j\|^d - k.$$

8. **Delaunay**.

$$\{i, j\} \in E \iff i, j \text{ adyacentes en } \text{DT}(P) \iff \exists \text{ círculo vacío que pasa por } x_i, x_j$$

9. **Voronoi**.

$$\{i, j\} \in E \iff \text{Vor}(x_i) \text{ y } \text{Vor}(x_j) \text{ comparten arista.}$$

10. **Alpha-Shape** / **Alpha-Hull**.

$$\{p, q\} \in E_\alpha \iff \exists r: \text{rad}(p, q, r) \leq \frac{1}{\alpha} \text{ y el circuncírculo correspondiente es vacío.}$$

11. **γ -vecinos**.

$$\Phi_\gamma(i, j) = \max_{z \neq i, j} \left(\cos \angle x_i z x_j \right)^\gamma \frac{\min(\|z - x_i\|, \|z - x_j\|)}{d_{ij}}, \quad \{i, j\} \in E \iff \Phi_\gamma(i, j) < 1.$$

Topología y TDA:

α -Shape para estimar fronteras de conjuntos de puntos.

ML/IA/GNN:

Regularización laplaciana para aprendizaje semi-supervisado.

Series de tiempo:

Grafos de recurrencia para detectar cambios de régimen.

GIS y geociencias:

EMST como esqueleto de drenajes.

Bioinformática y morfometría:

β -Skeleton para cuantificar vecindad celular.

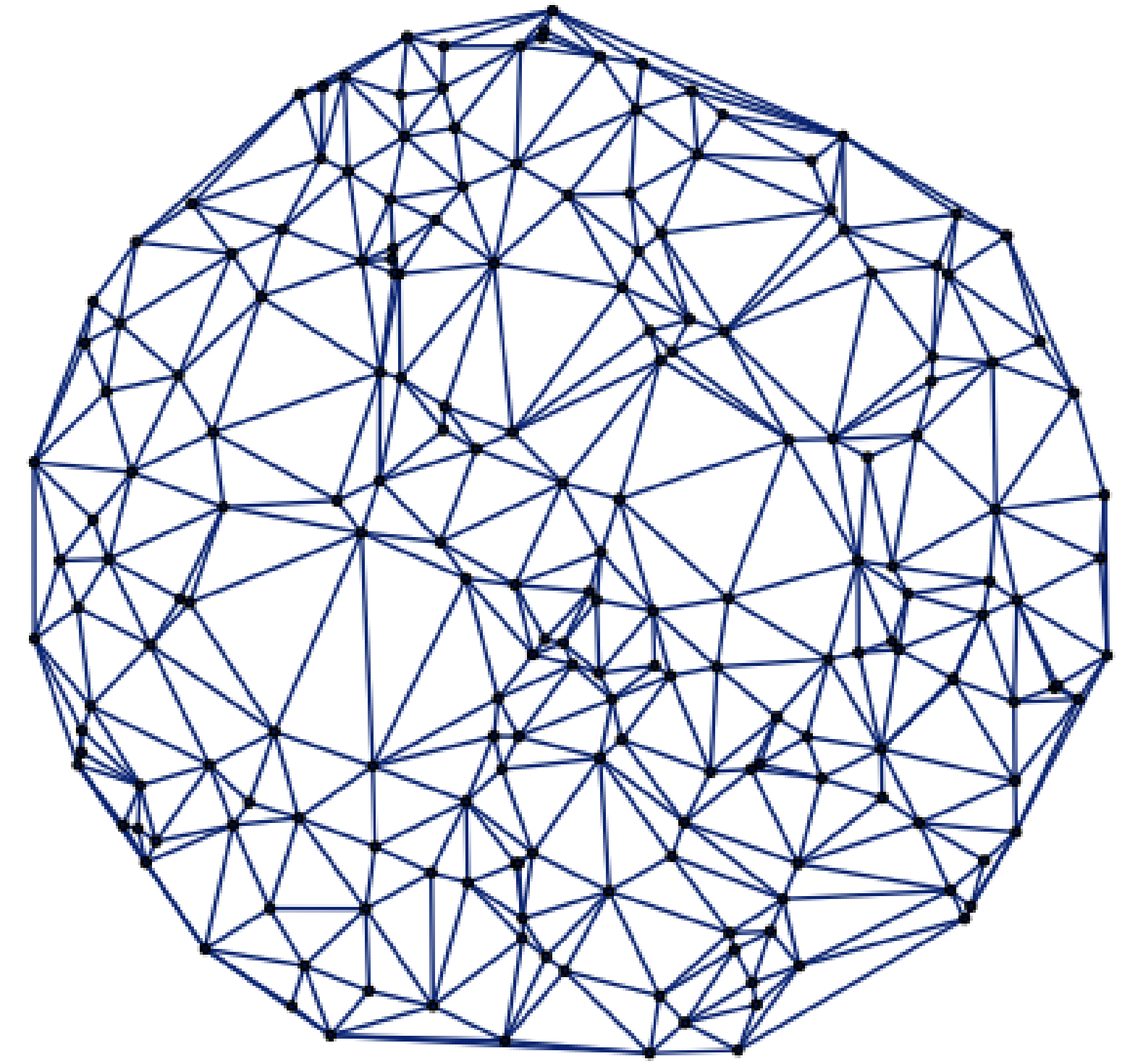
Robótica y planificación de movimiento:

Conectando configuraciones con k-NN o radio para planear rutas.

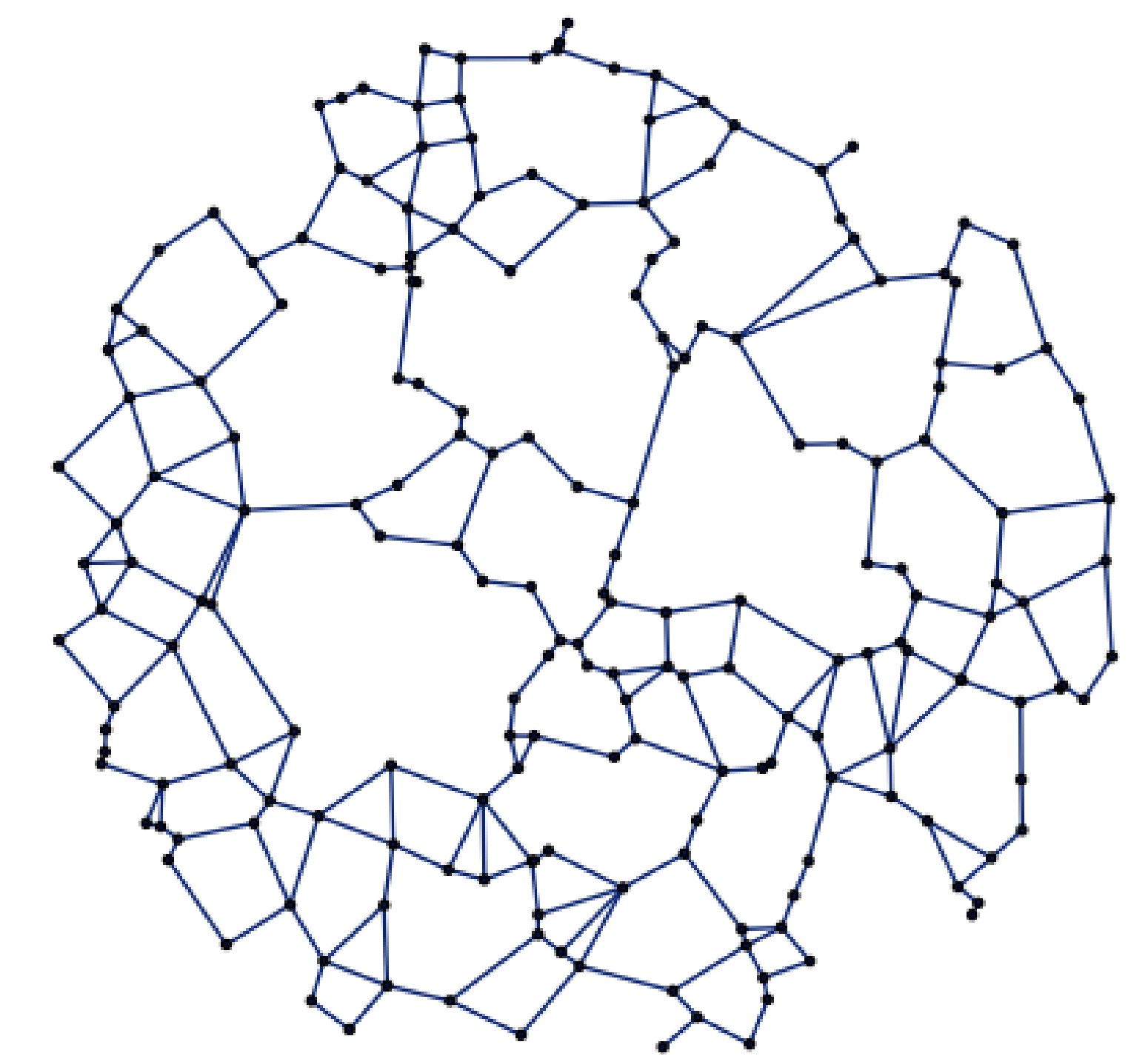
Finanzas:

MST sobre 1-p para estructura de mercado.

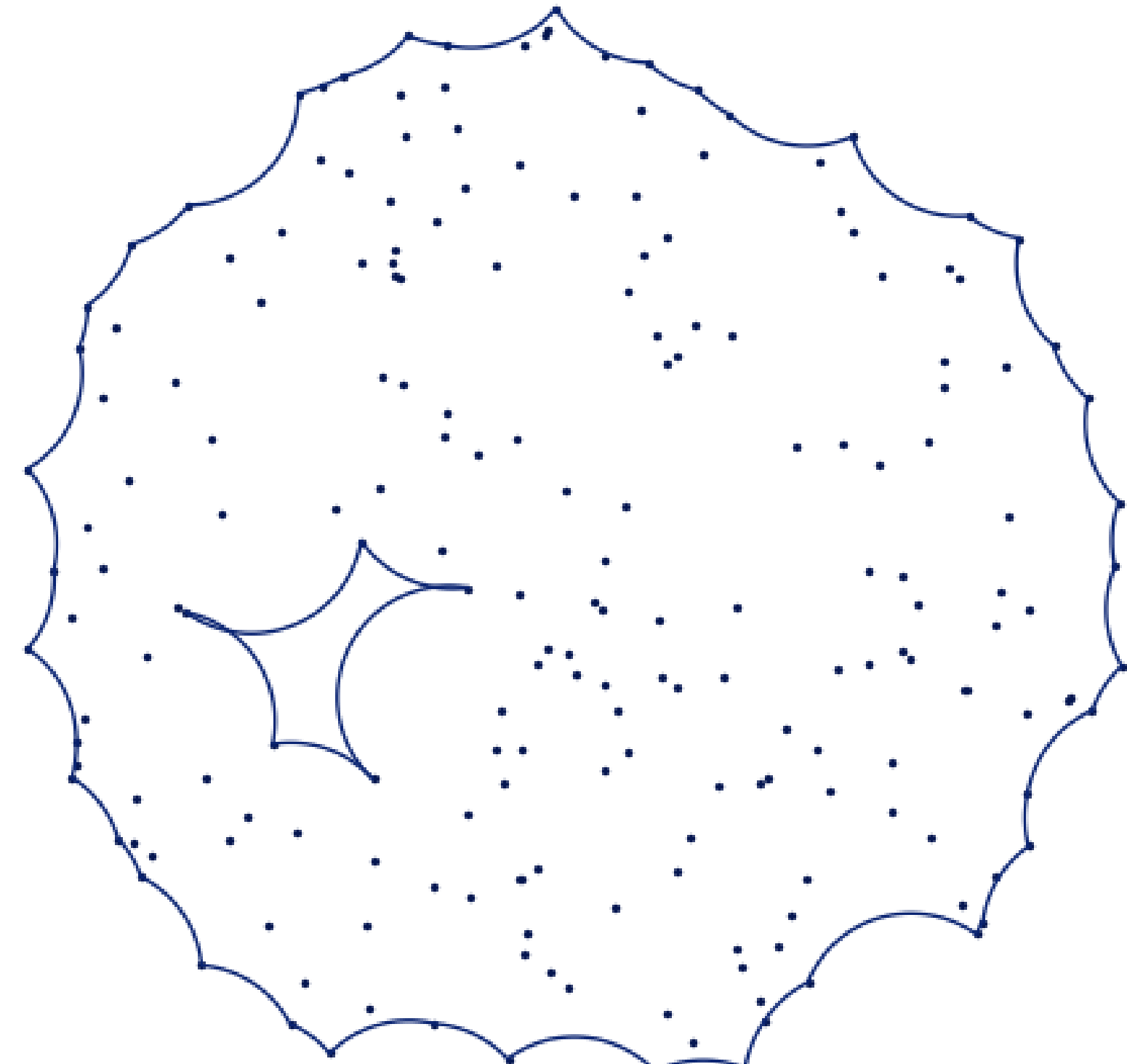
Delaunay Triangulation



β -Skeleton
 $\beta=1.5$, closed=False, type=lune



Alpha-Hull
alpha=-5



β -Skeleton
 $\beta=1.5$, closed=False, type=lune

