VORONOI MESH GENERATION
TAILORED FOR URBAN FLOW SIMULATIONS

PRESENTER: ÁKOS S. SÁRKÁNY

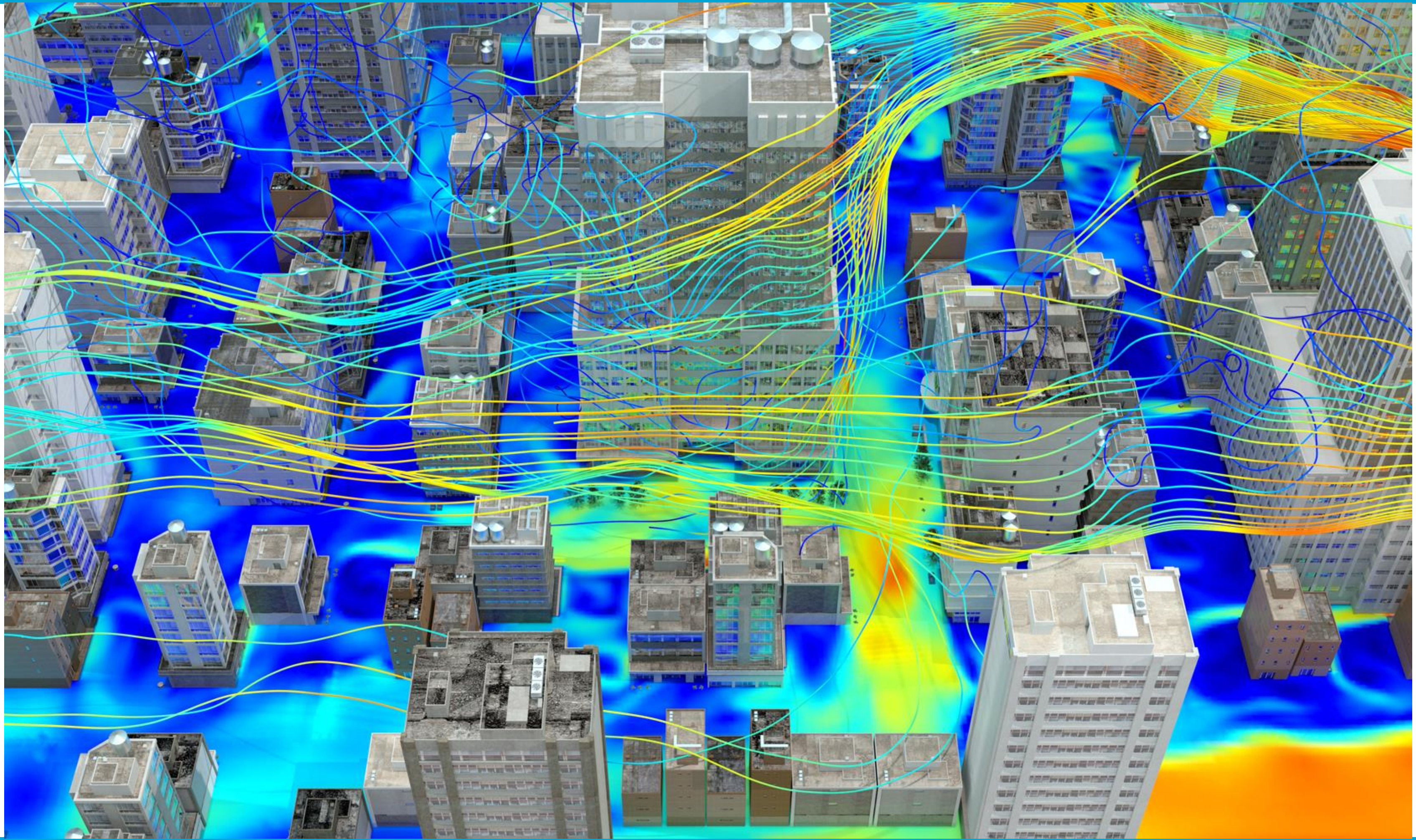
Supervisor: Dr. Clara García-Sánchez
Co-supervisor: Dr. Hugo Ledoux
Co-reader: Dr. Akshay Patil

Agenda

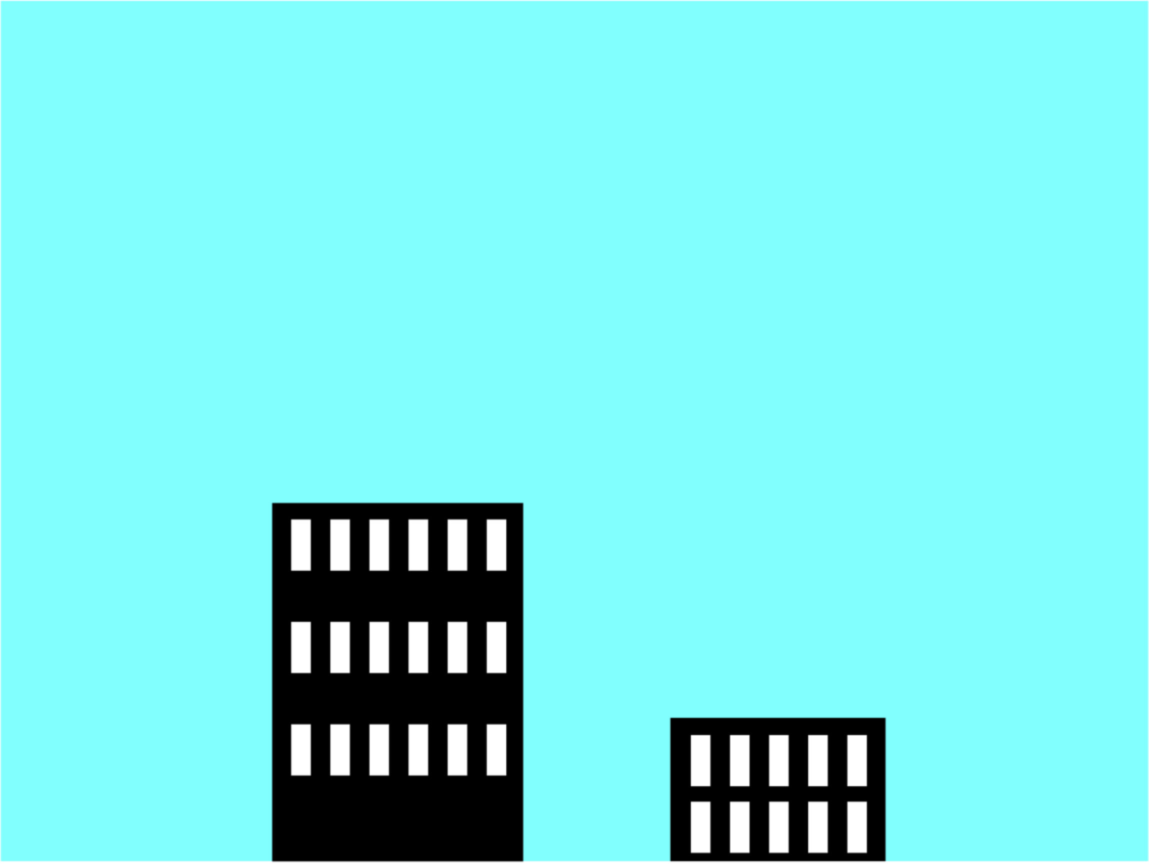
1. Introduction
2. Motivation
3. Relevant concepts from literature
4. Methodology
5. Results
6. Conclusions & Future work

INTRODUCTION

Computational Fluid Dynamics (CFD)

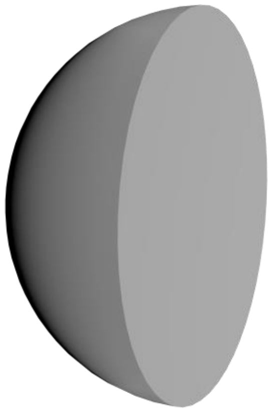


Urban CFD Domain

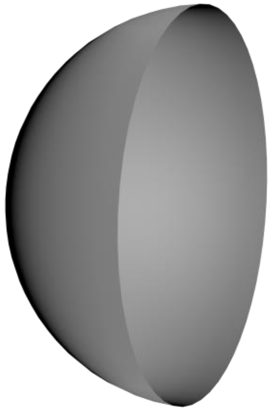


Boundary representation

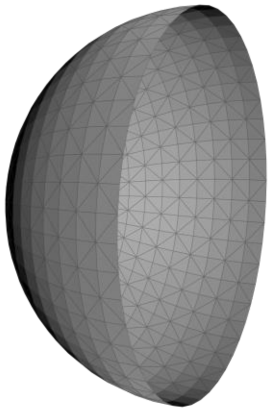
- **Boundary representation:** Efficient, lightweight, enough for most cases.
- **Piecewise Linear Complex (PLC):** Discrete set of connected faces embedded in 3D which bound the volume. No boundary loops.



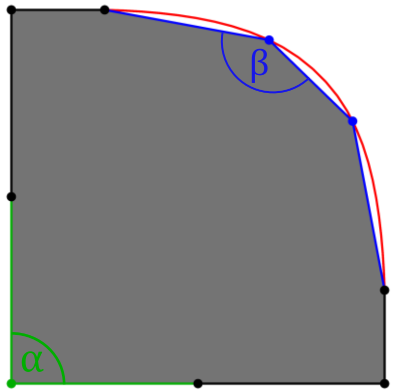
Volume



Boundary



PLC



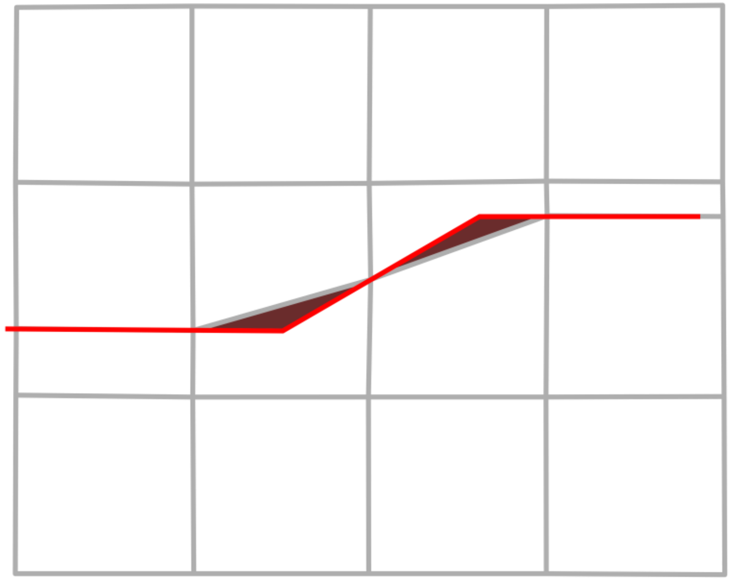
Sharp vs. Smooth

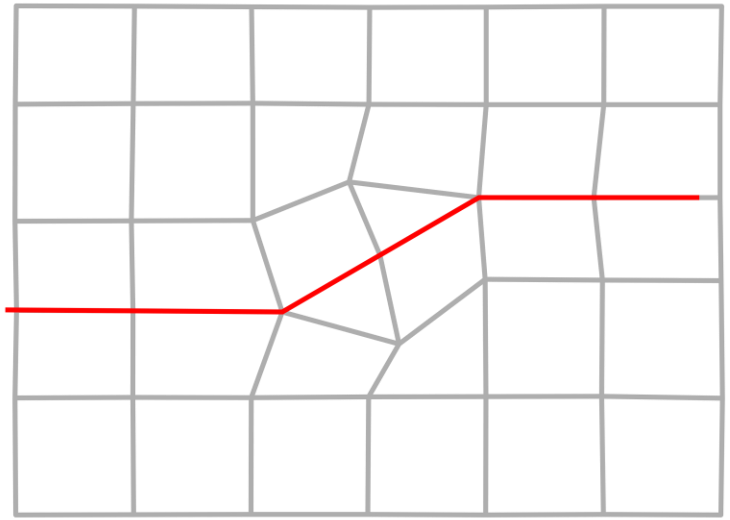
Meshing

- Divide the volume into finite number of discrete cells
- Capture flow locally

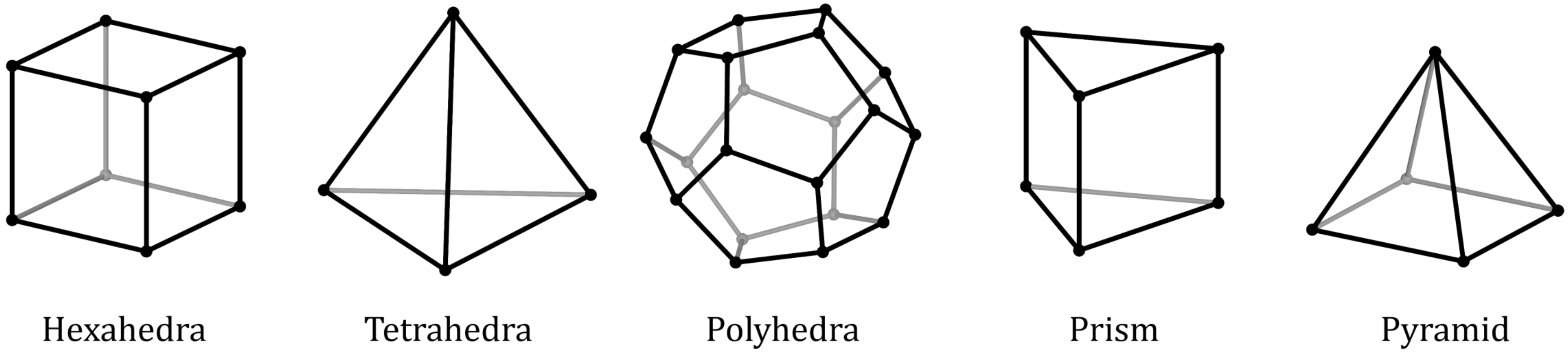
MESHING: CONFORMITY

- Cells should conform to boundaries closely.

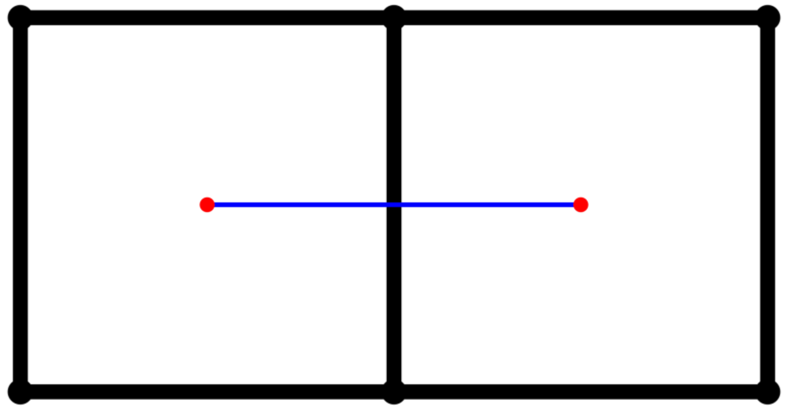


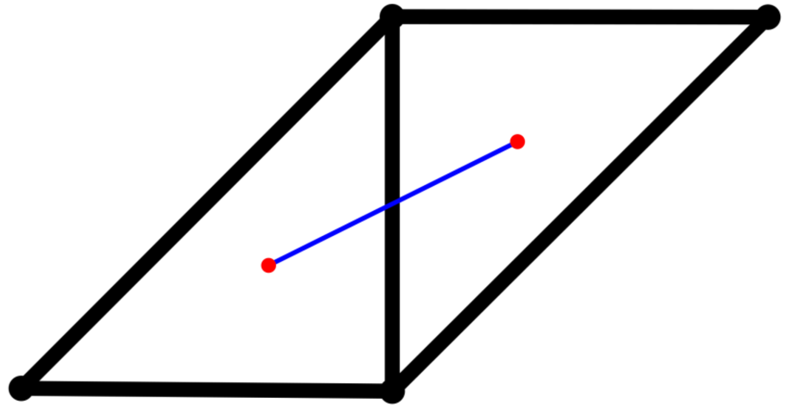


Common cell shapes in 3D

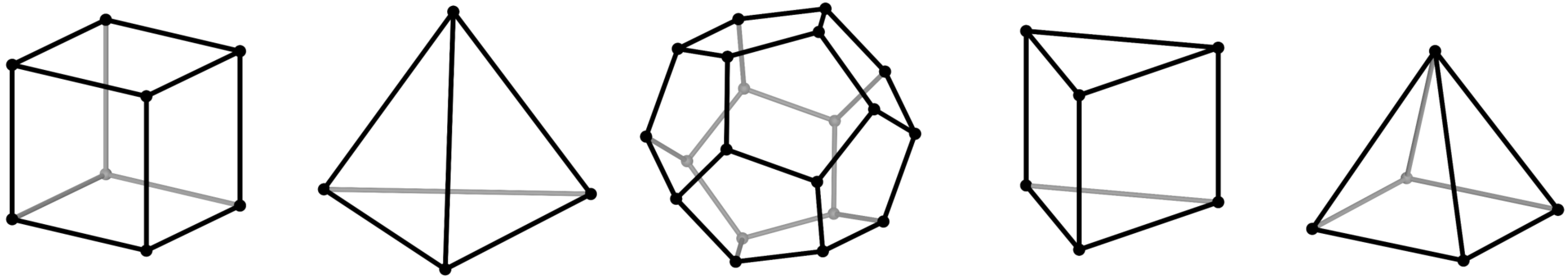


Orthogonal connections



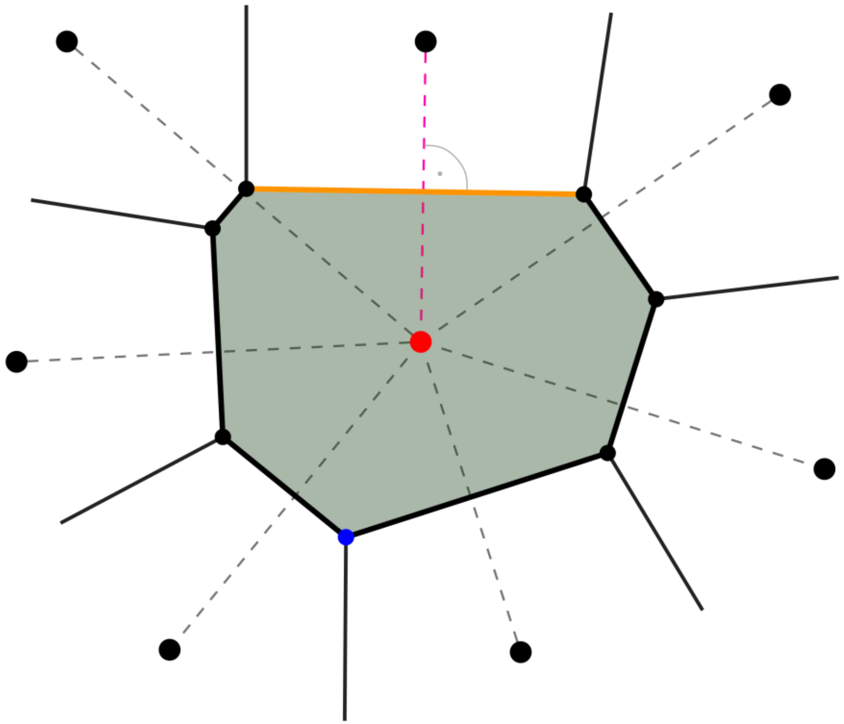


COMMON CELL SHAPES IN 3D

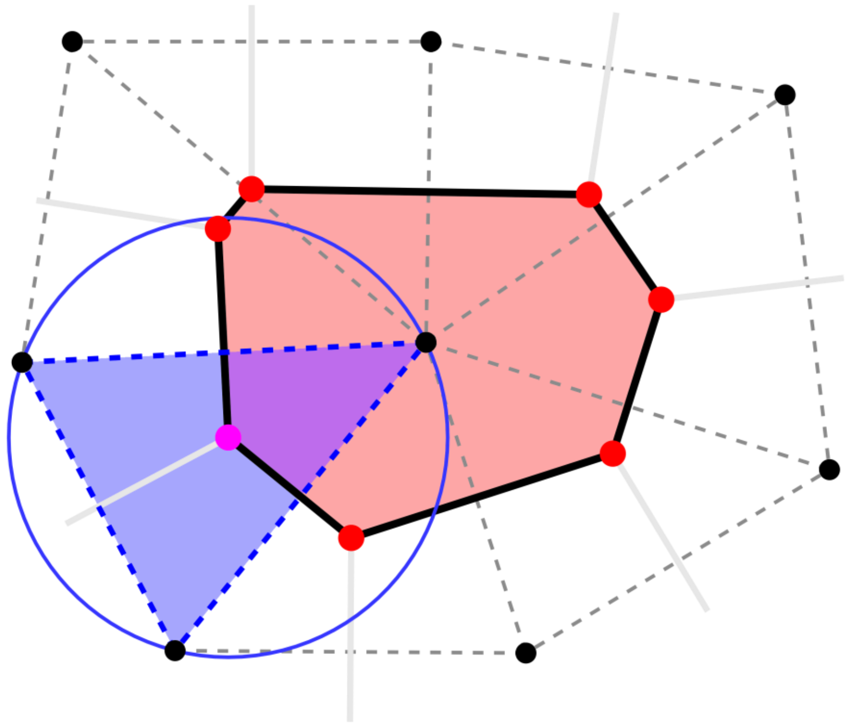


Hexahedra Tetrahedra Polyhedra Prism Pyramid

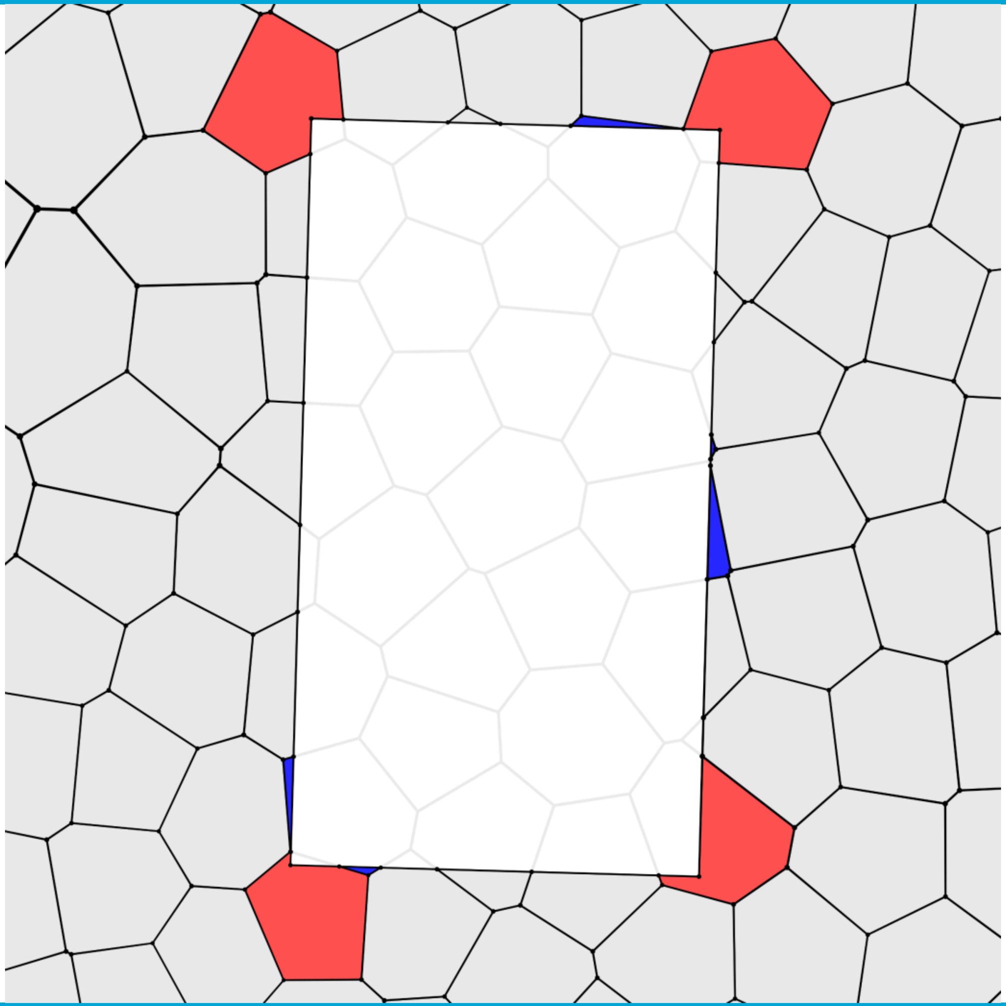
VORONOI CELL STRUCTURE



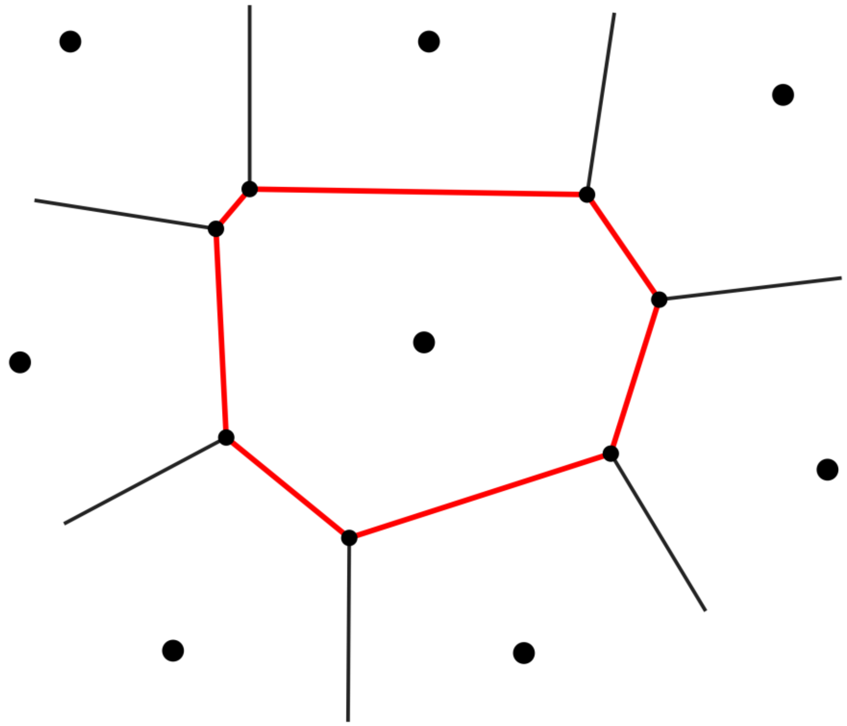
VORONOI CELL STRUCTURE



Clipping

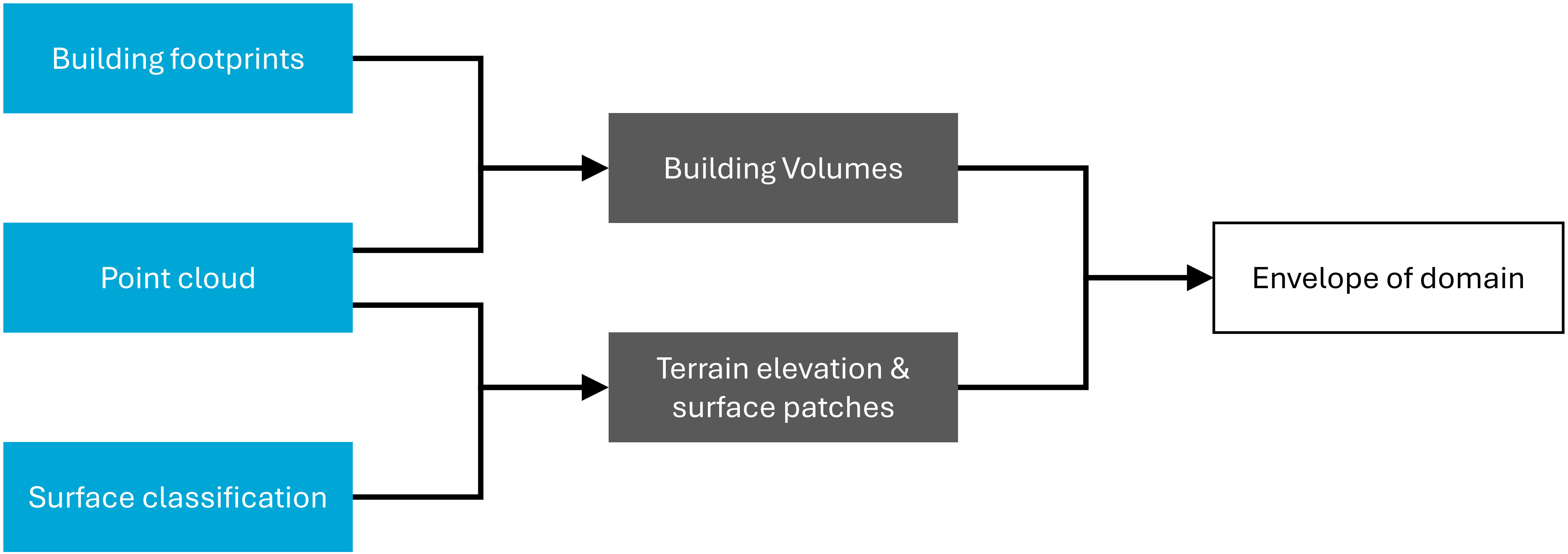


VORONOI CELL STRUCTURE

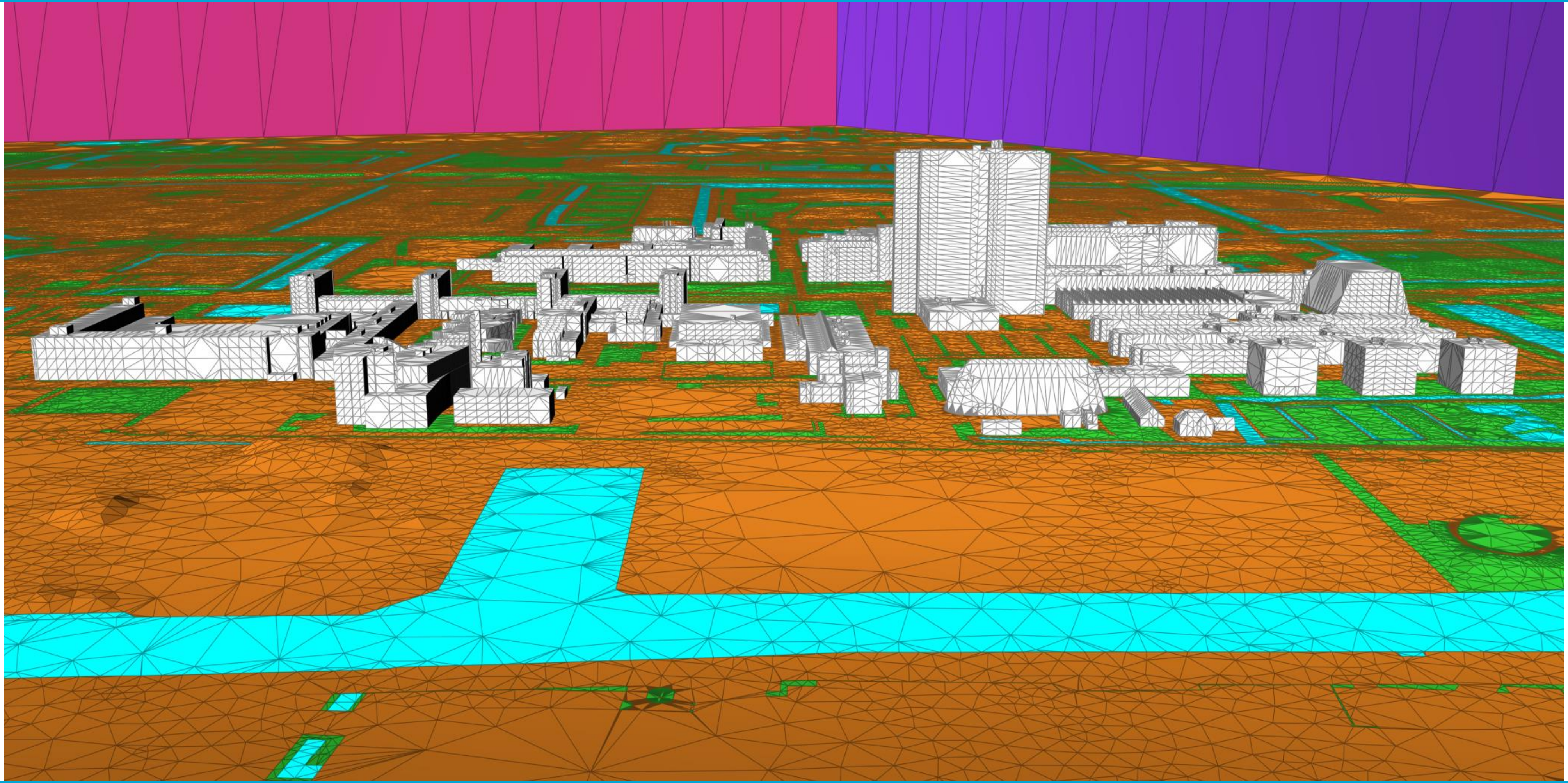


MOTIVATION

Automatic reconstruction with City4CFD



CITY4CFD OUTPUT



Missing link

- City4CFD outputs *boundary representation only*
- Currently: `snappyHexMesh` for meshing
- Limitations:
  - Does not conform well compared to running time
  - Manual configuration require

Goal

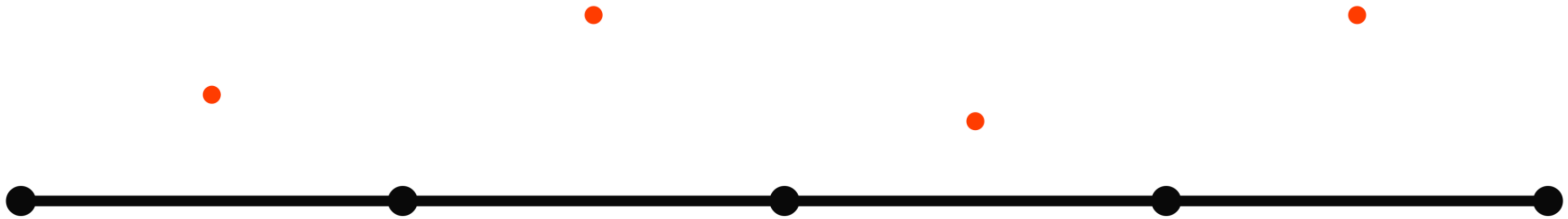
- Improve meshing in context of urban CFD
- To develop a mesher with Voronoi approach
  - Produces a mesh conforming more closely to input boundary
  - Maintains boundary patch information
  - Does not use clipping
  - Meshing requires limited manual configuration

Scope

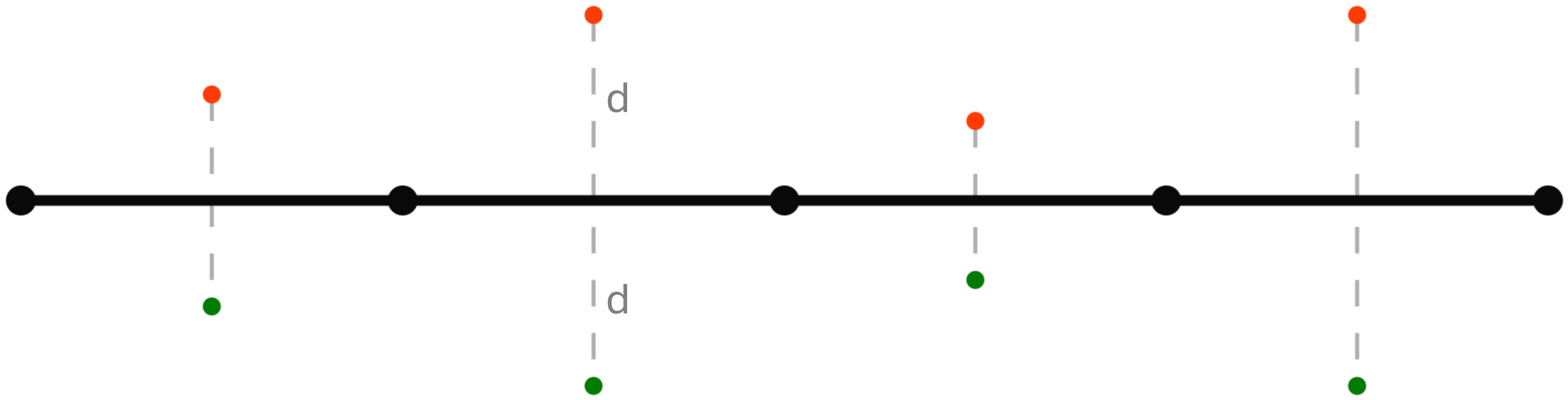
- **Focus:**
  - Derive workflow of Voronoi based meshing tailored for urban CFD
  - Implement prototype
- **Not in scope:**
  - Fix input geometry
  - Not 2-manifold cases

RELEVANT CONCEPTS FROM LITERATURE

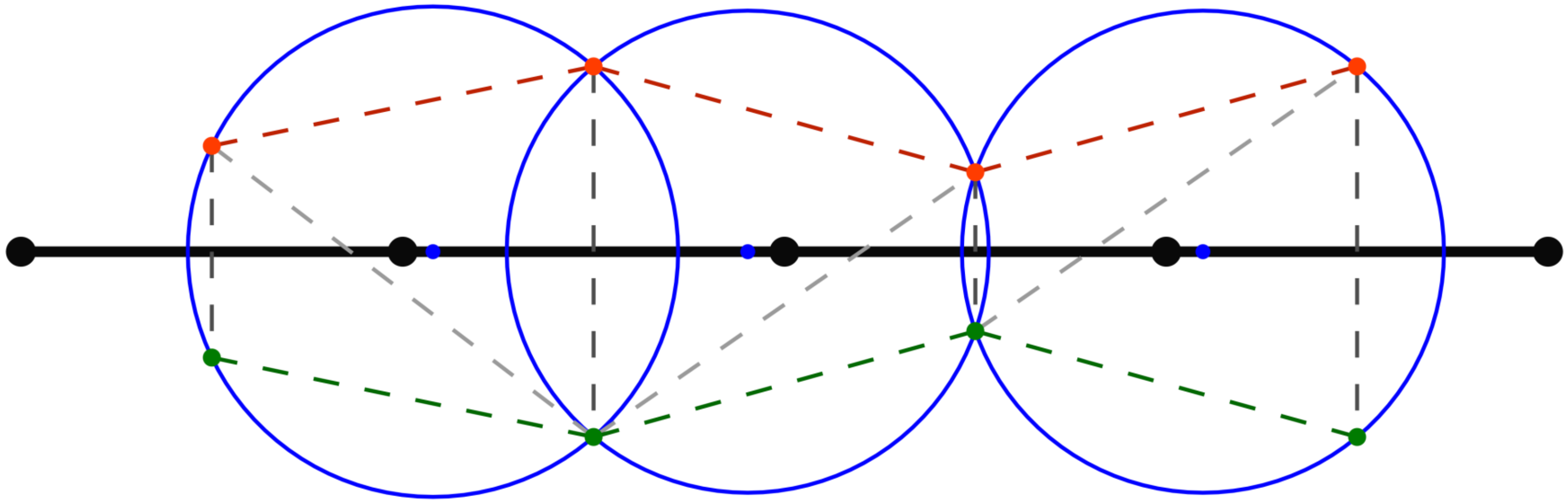
Intuitive approach: Mirroring



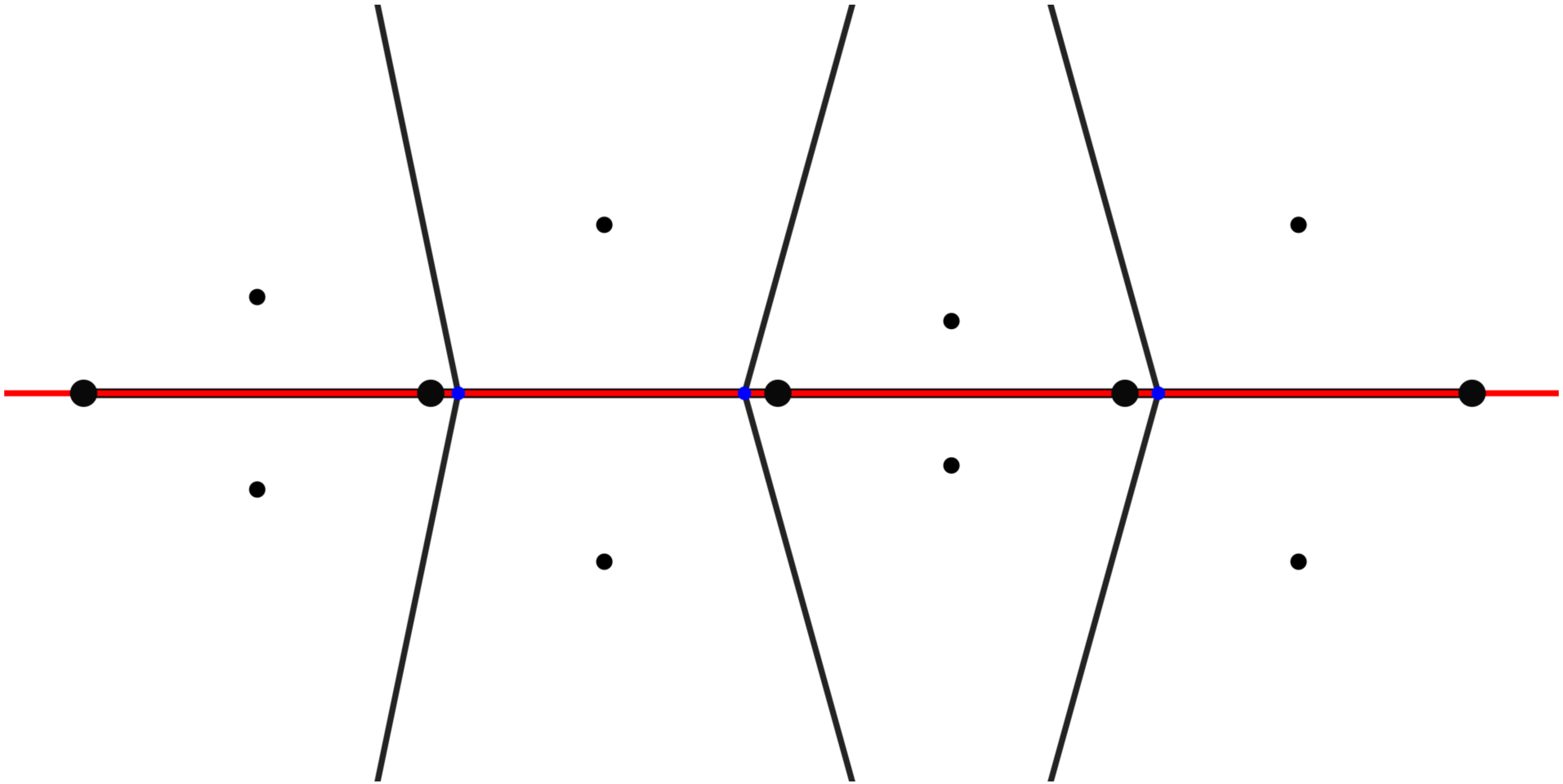
Intuitive approach: Mirroring



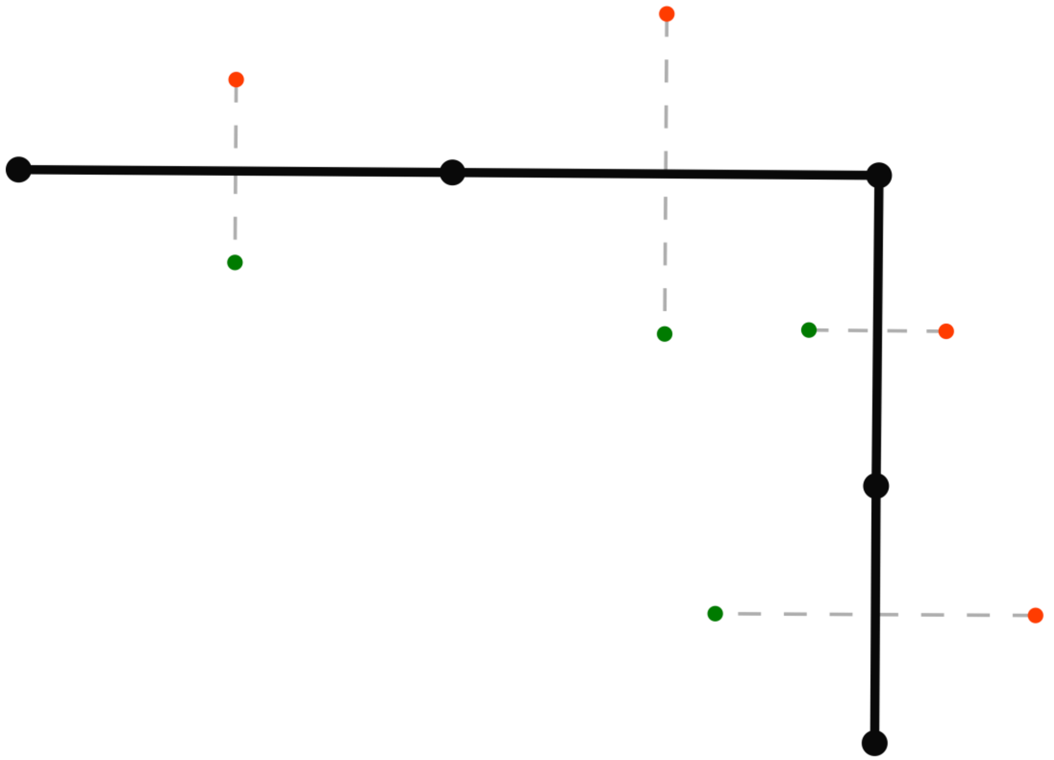
Intuitive approach: Mirroring



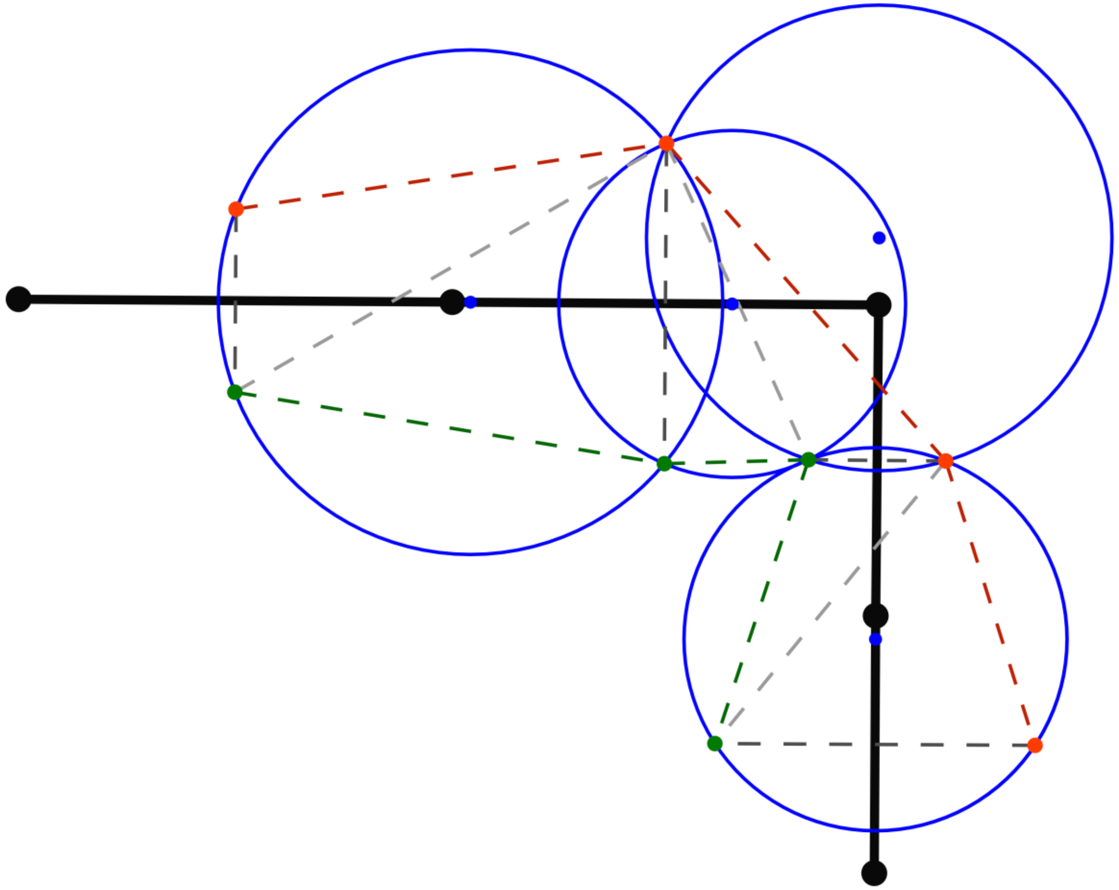
Intuitive approach: Mirroring



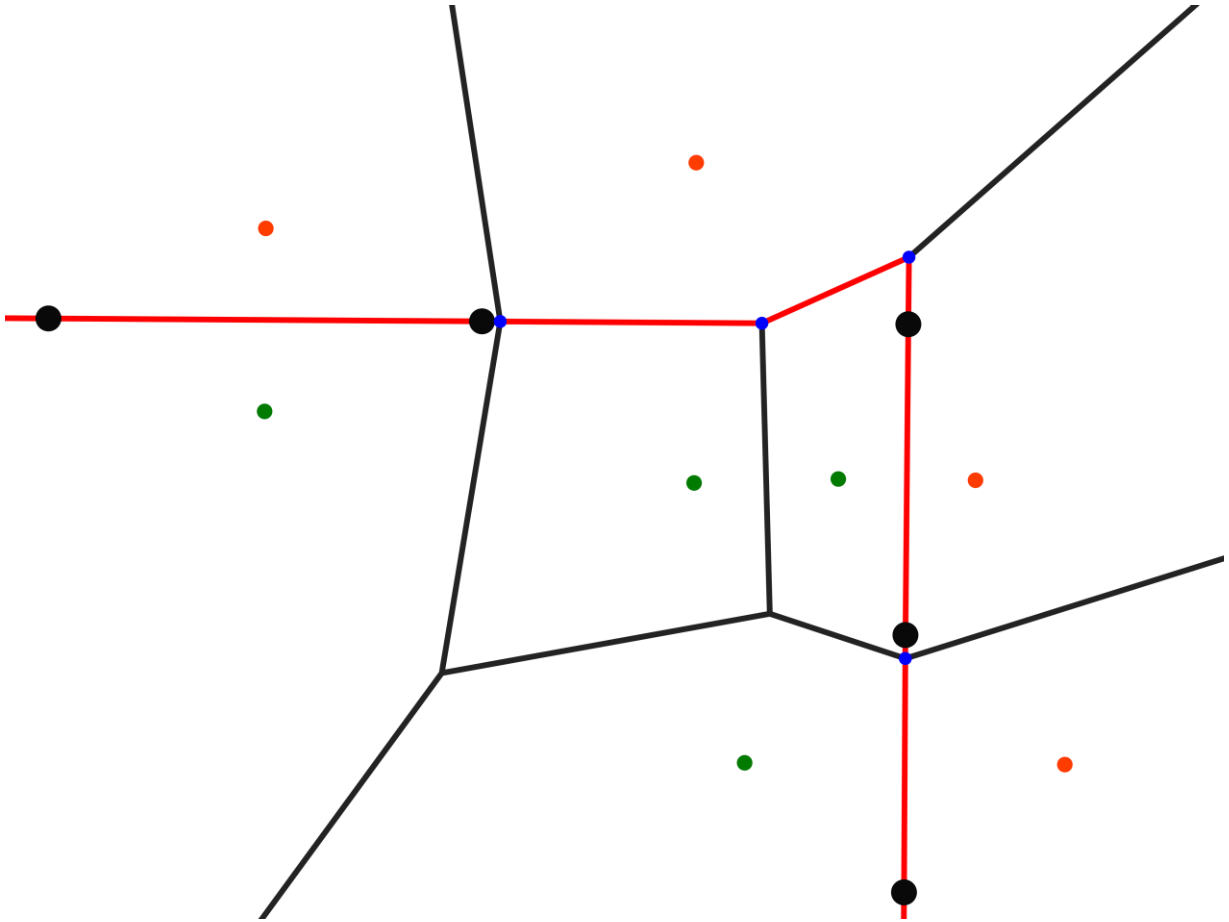
Mirroring: An incorrect edge case



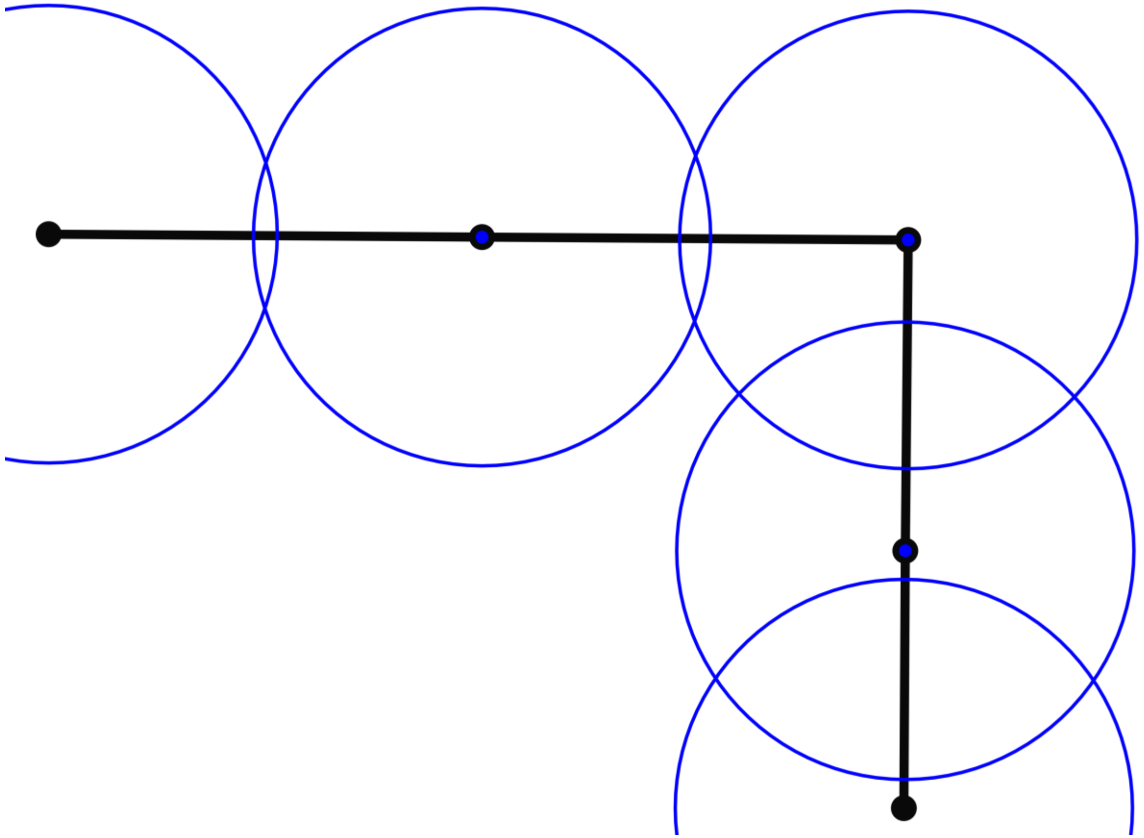
Mirroring: An incorrect edge case



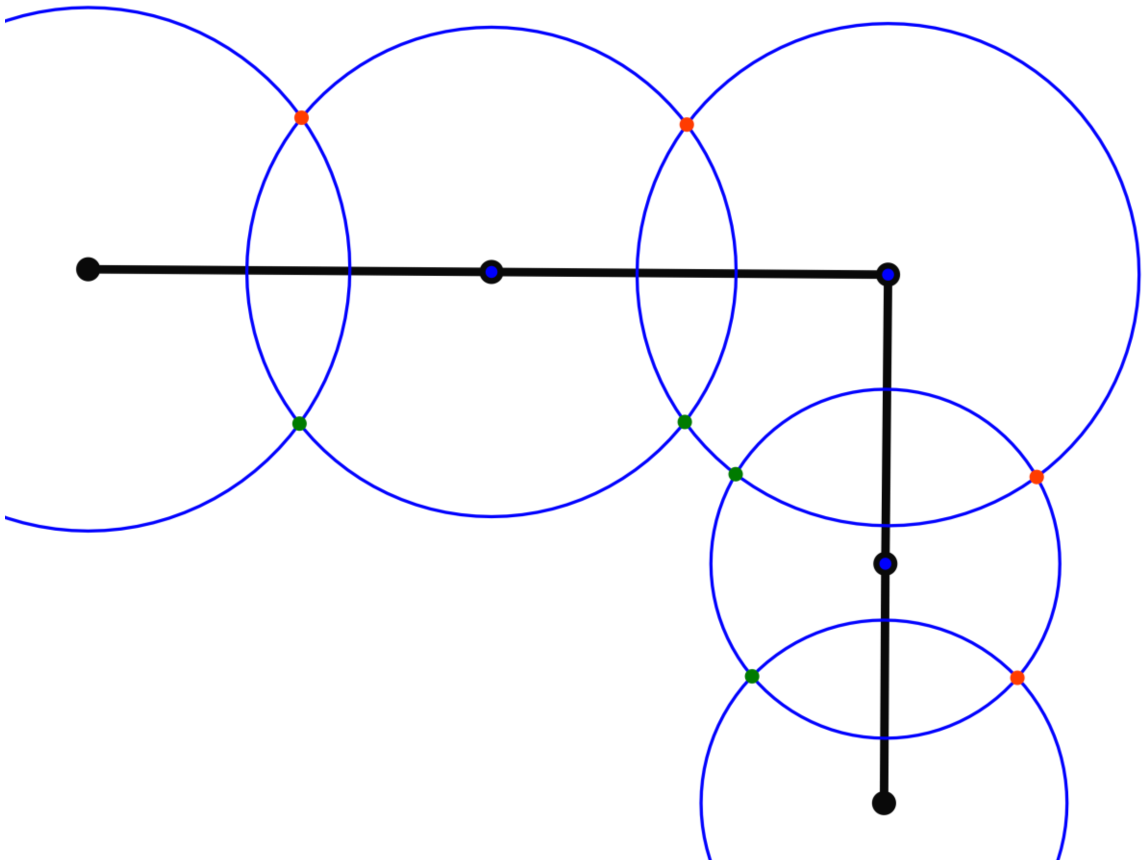
Mirroring: An incorrect edge case



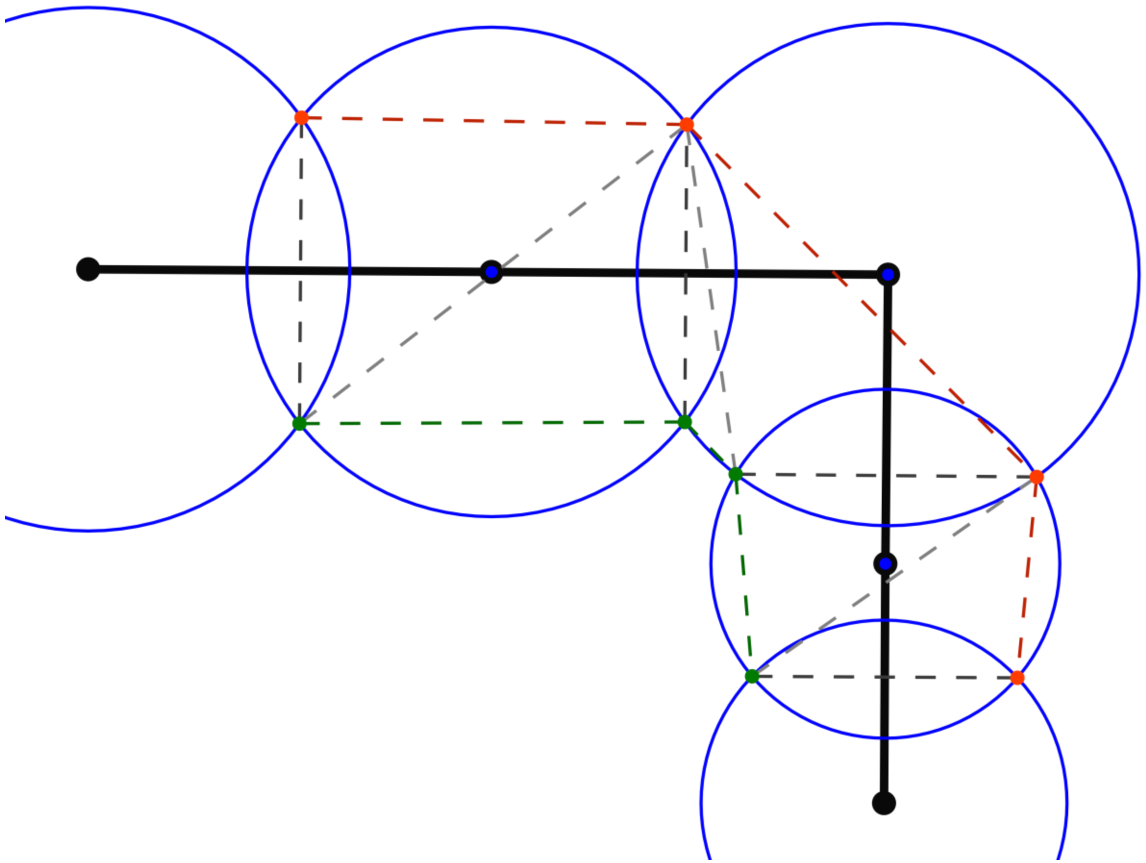
Working approach for seed placement



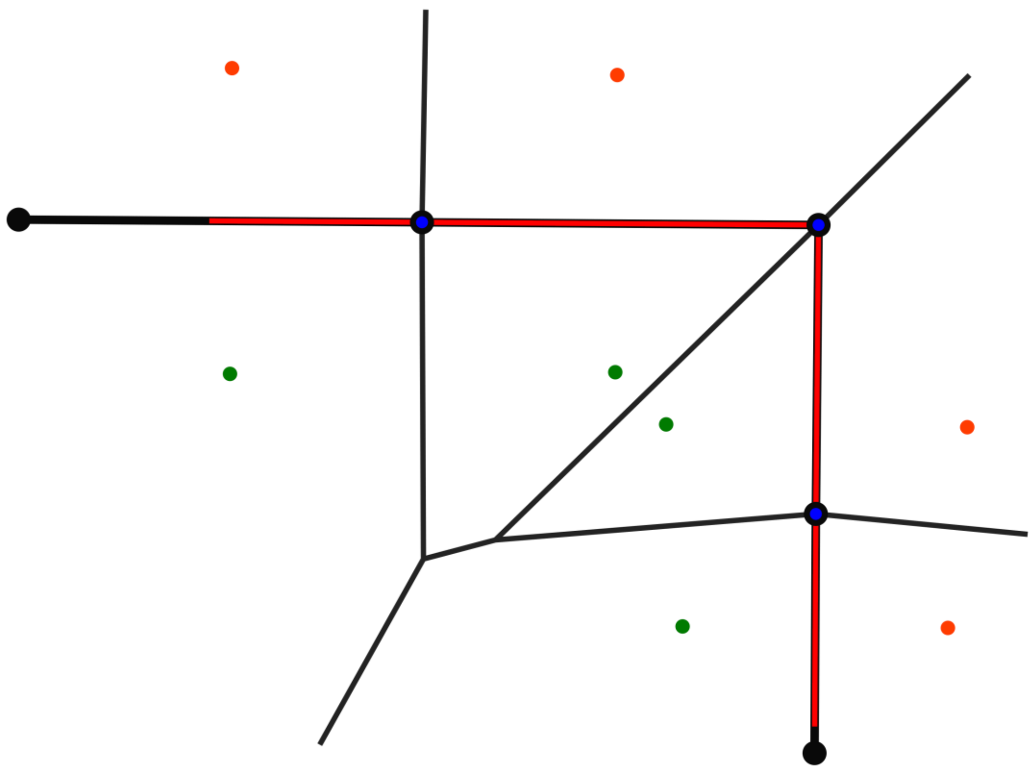
Working approach for seed placement



Working approach for seed placement

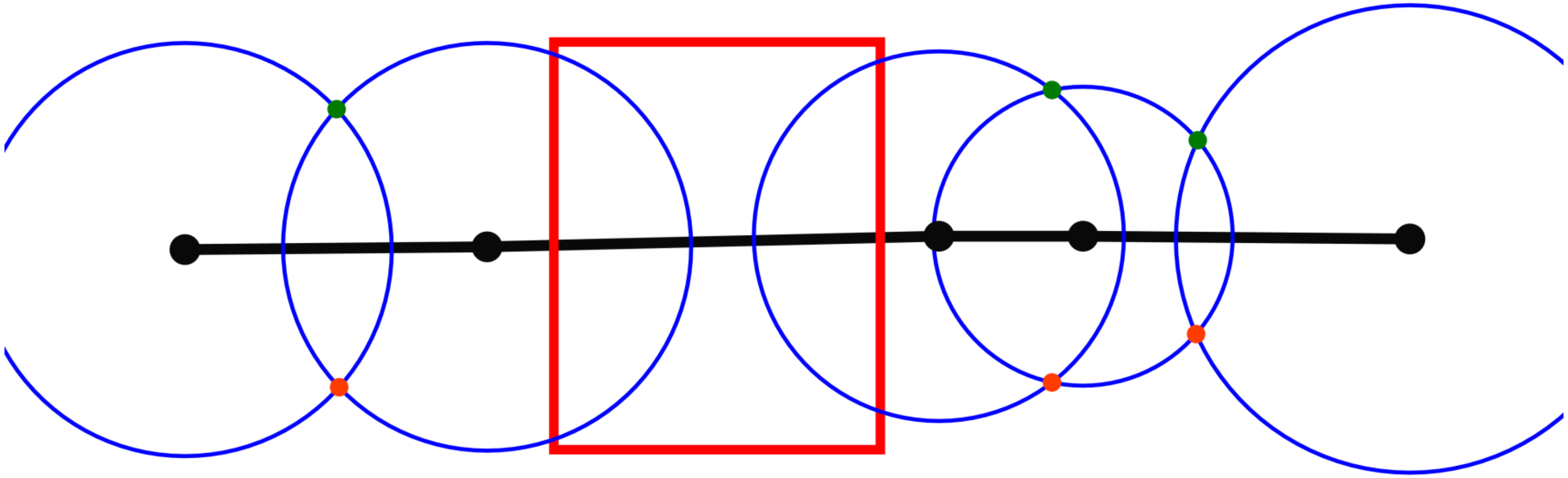


Working approach for seed placement



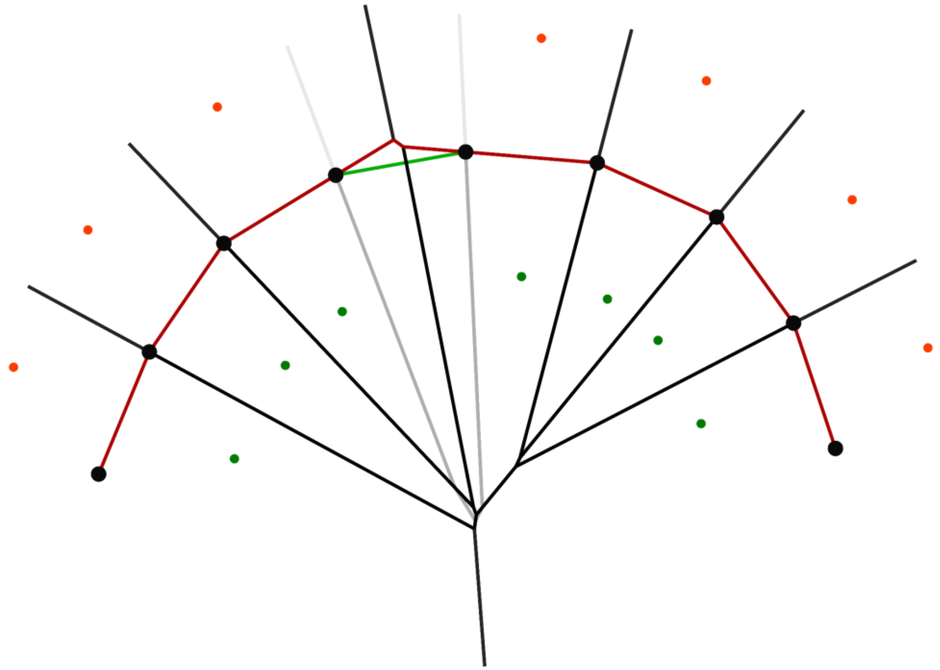
Conditions

Union of sphere must cover entire surface



Conditions

Union of sphere must cover entire surface



Conditions

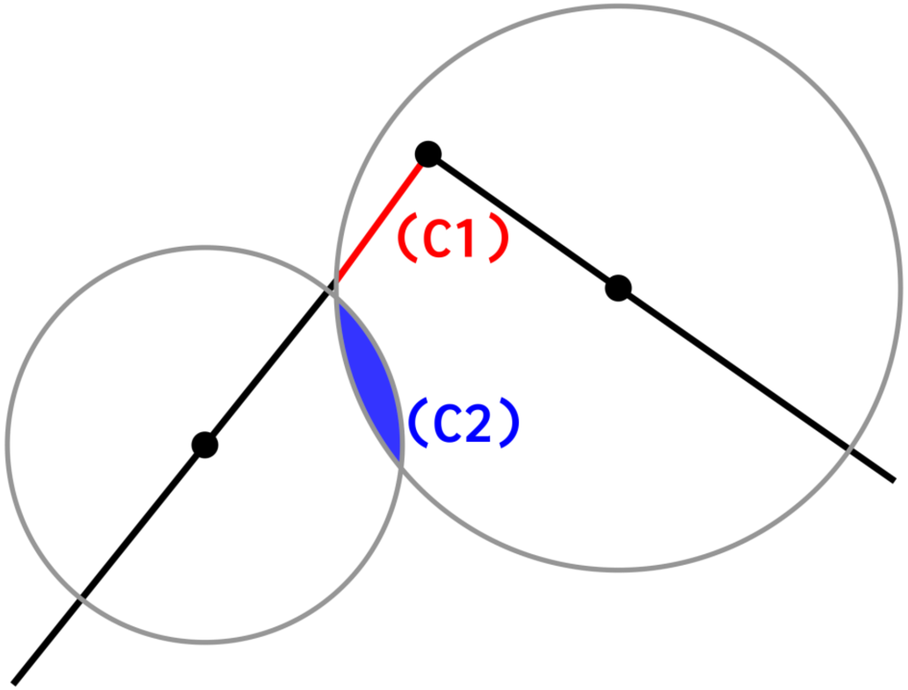
- Union of sphere must cover entire surface:
  - Sufficient vertex density w.r.t. *local feature size (lfs)* → epsilon sample
  - Appropriate sphere radii

- P is an *epsilon sample*: $\left\| x - p \right\| \leq \inf_{a \in A} \| x - a \|, \forall p \in P$
  - $x \in \mathcal{M}$
  - A is the *medial axis*

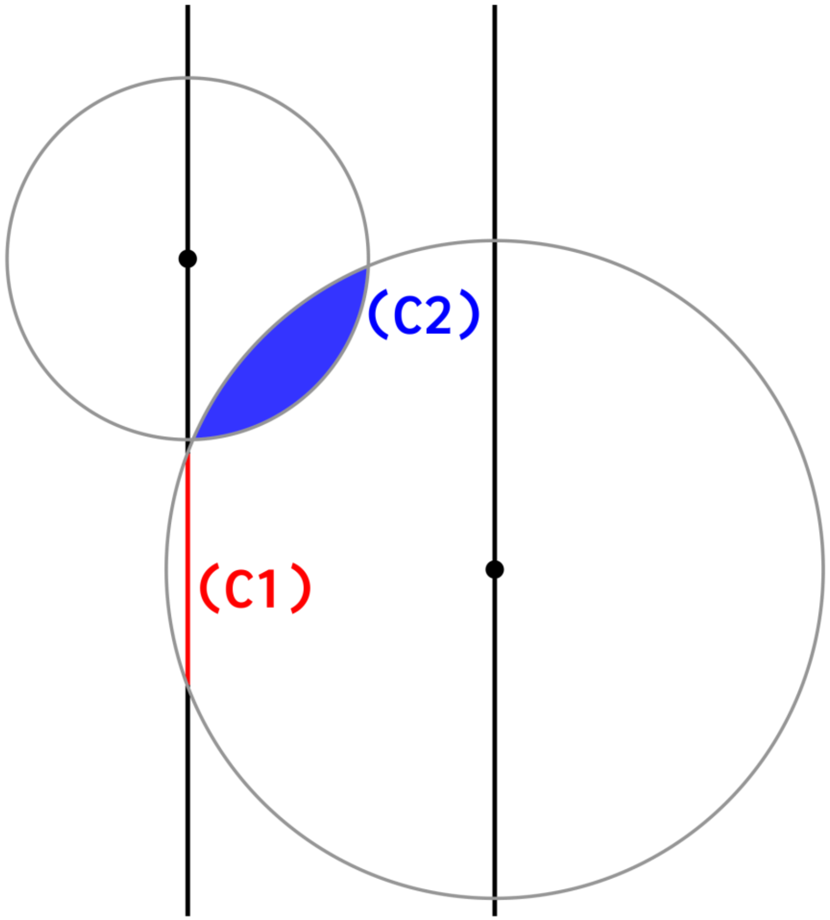
SAMPLING ON A PLC

- VoroCrust approach: *Maximum Poisson Disk Sampling (MPS)*
- Insert vertices until the spheres cover the entire surface
- Sphere radii determined by sphere conditions:
  - (C1) Smooth coverage
  - (C2) Smooth overlaps
  - (C3) Local L-Liptschitness
  - (C4) Deep-coverage

Sphere Conditions: Smooth overlaps & coverage

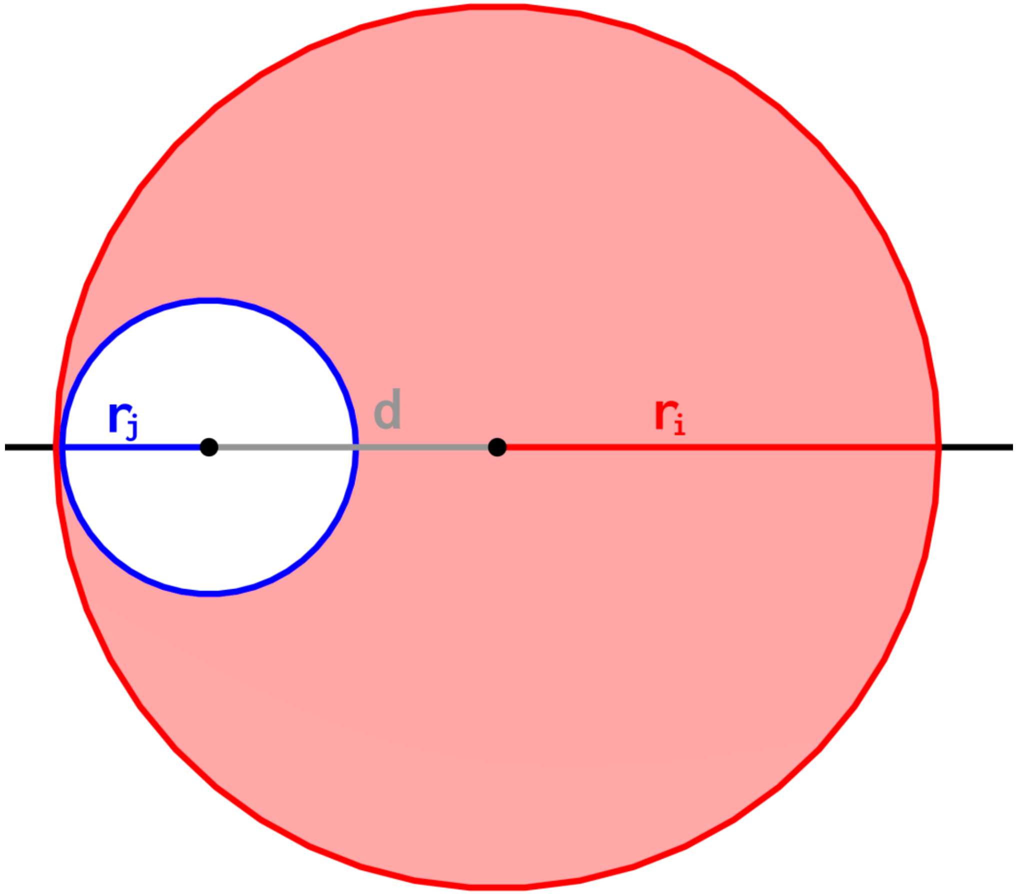


Sphere Conditions: Smooth overlaps & coverage



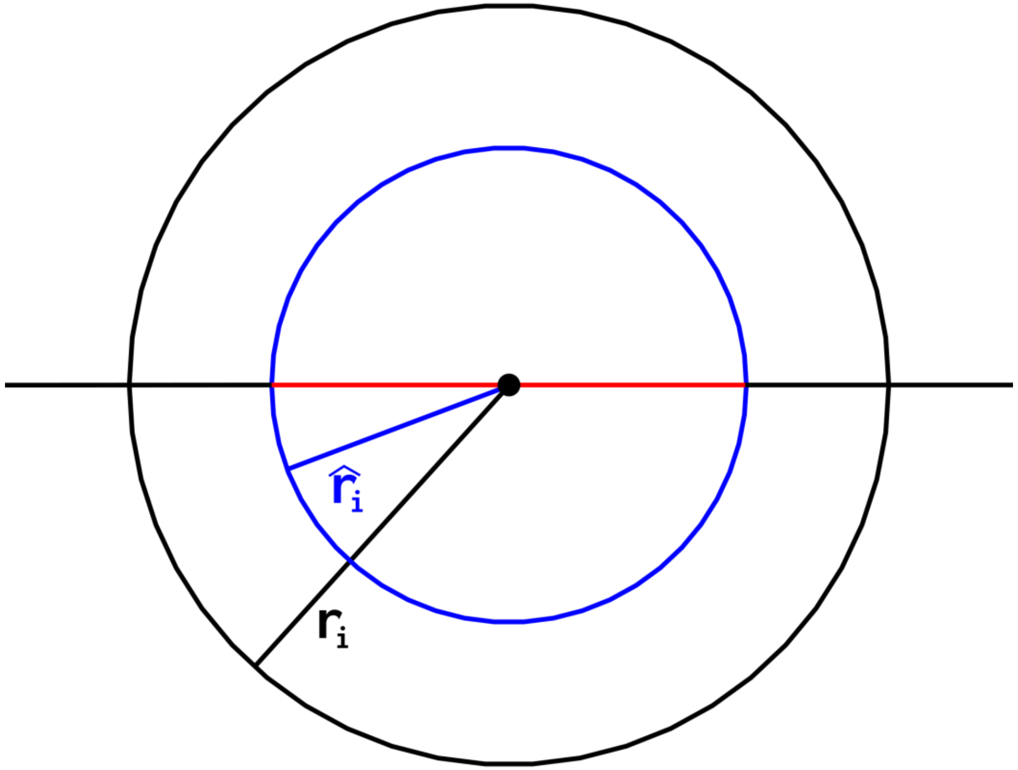
Sphere Conditions: Local L-Liptschitness

$$r_i \leq r_j + L \cdot d$$



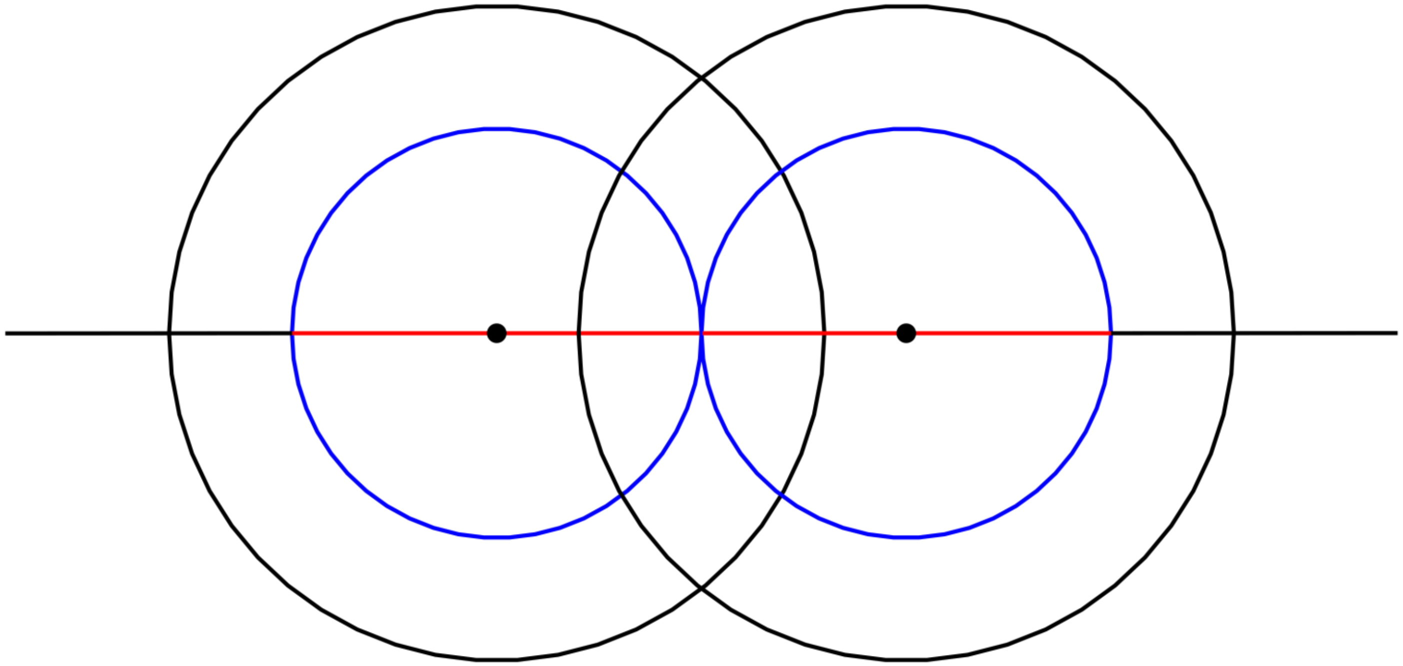
Sphere Conditions: Deep Coverage

$$\hat{r}_i = \alpha \cdot r_i$$



SPHERE CONDITIONS: DEEP COVERAGE

$$\hat{r}_i = \alpha \cdot r_i$$

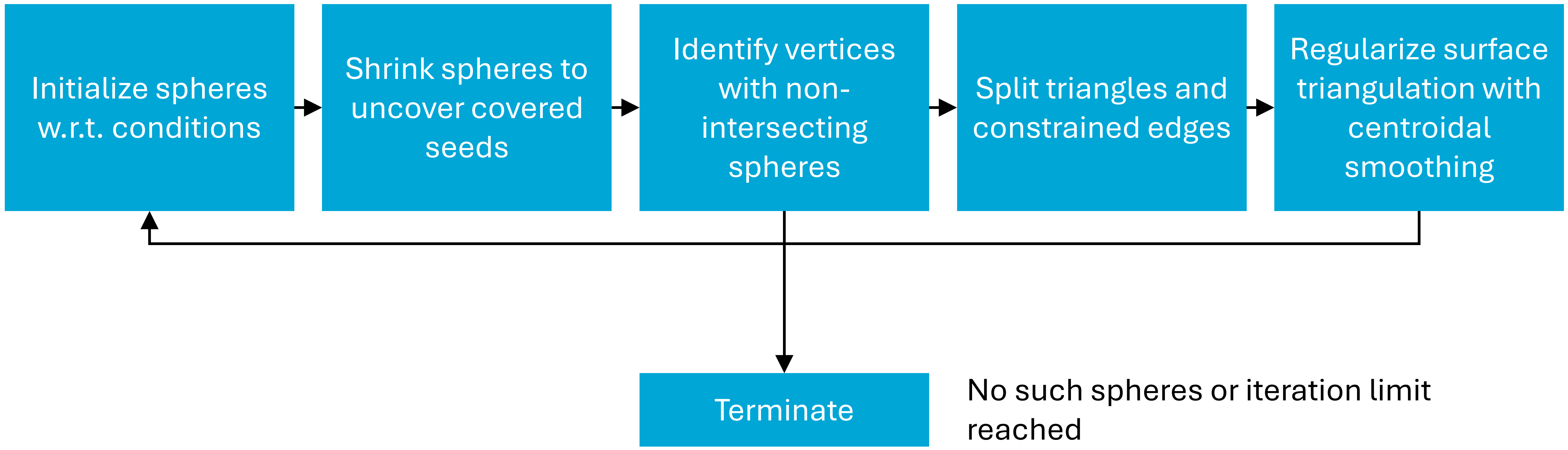


METHODOLOGY

Preprocessing

- Input file
  - 2-manifold PLC
  - With boundary information
- Configuration
  - Sharp angle threshold
  - Boundary face target area
  - Lipschitz constant
  - Iteration limits

REFINEMENT

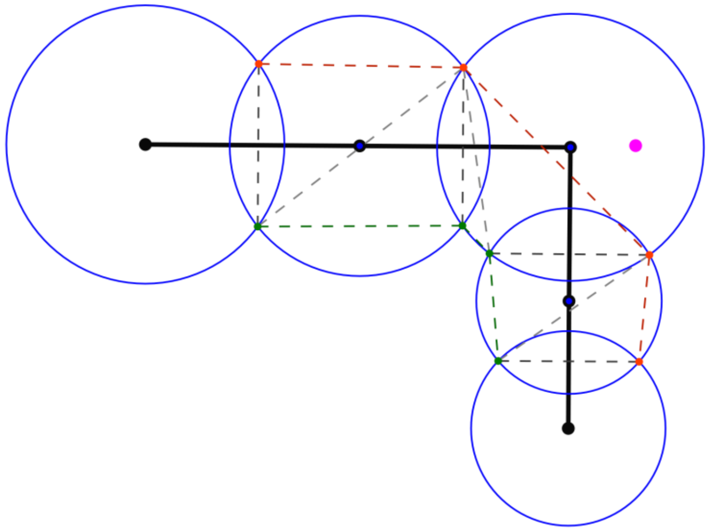

Initialize spheres w.r.t. conditions
Shrink spheres to uncover covered seeds
Identify vertices with non-intersecting spheres
Split triangles and constrained edges
Regularize surface triangulation with centroidal smoothing
Terminate
No such spheres or iteration limit reached

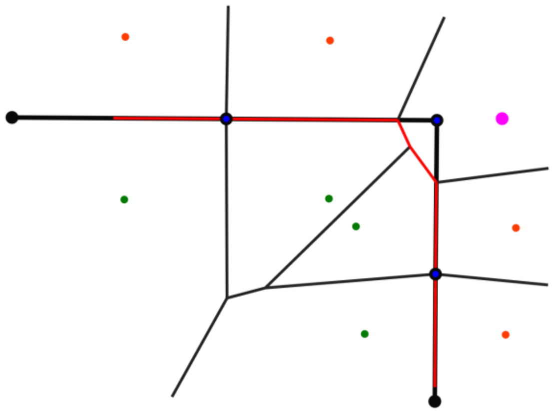
Refinement: Sphere initialization

- To maximal radii that sphere conditions allow
  - Make sure they do not deeply cover vertices
  - Enforce (C1), (C2)
  - Shrink spheres to adhere (C3) so no sphere covers others full

Refinement: Sphere shrinking

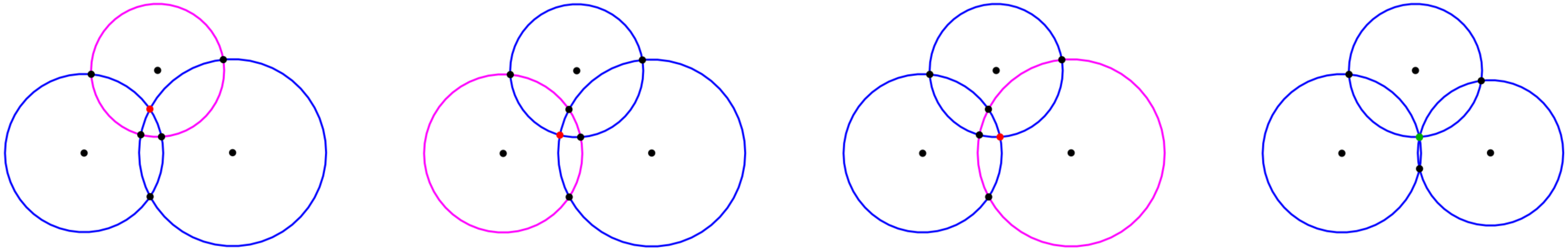
- Ensures the seeds generate a Voronoi diagram without slivers
- Ensure seeds pair correctly





REFINEMENT: SPHERE SHRINKING

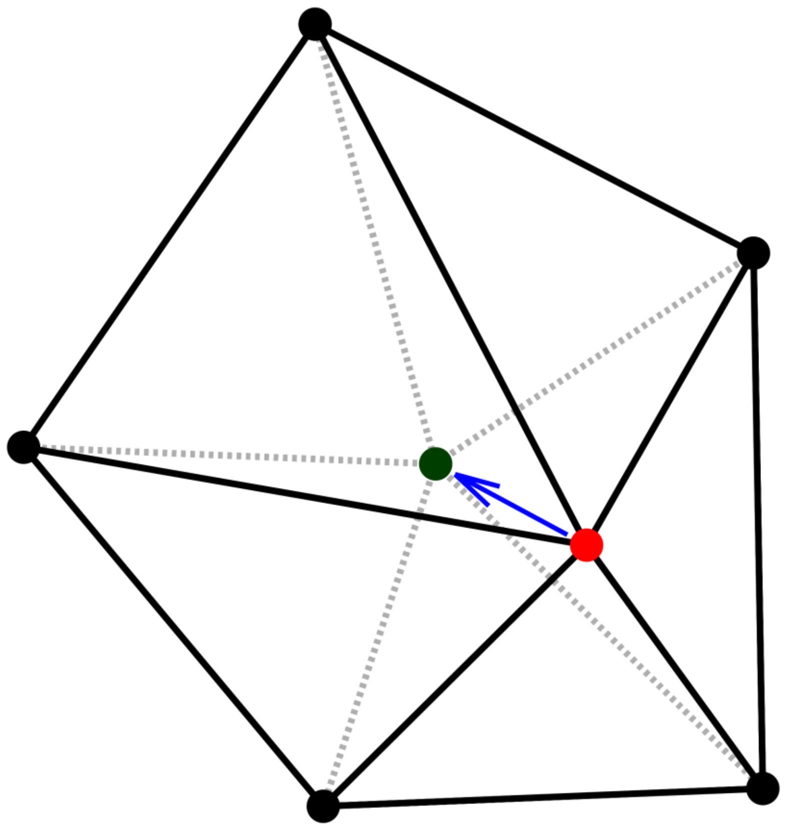
- Ensures the seeds generate a Voronoi diagram without slivers
- Ensure seeds pair correctly



REFINEMENT: SMOOTHING

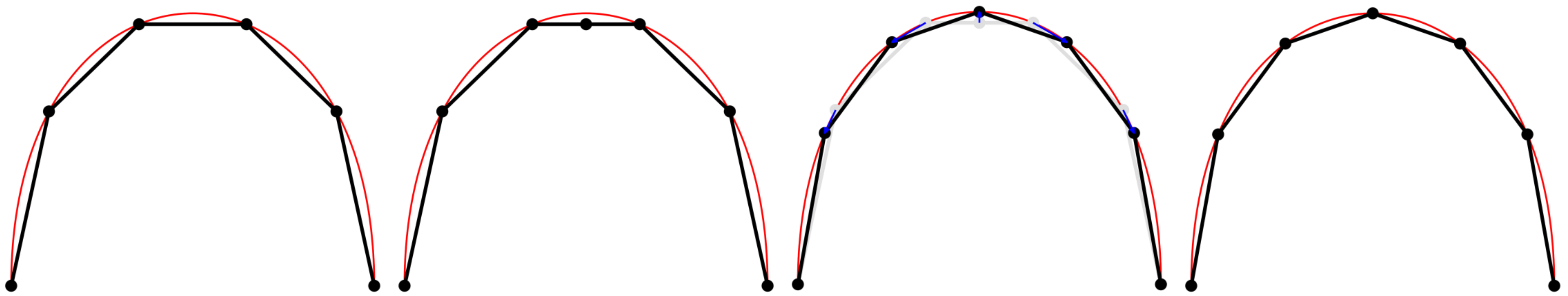
Centroidal smoothing:

- Average position of adjacent vertices
- Independent per vertex
- Fast



REFINEMENT: SMOOTHING CURVED BOUNDARIES

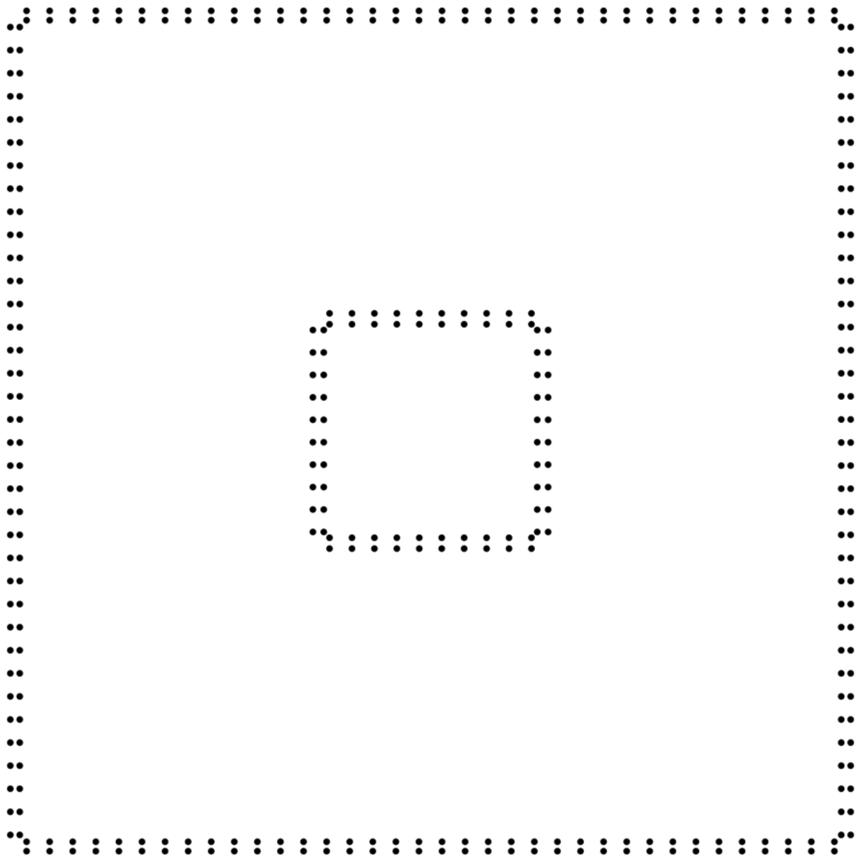
- Centroidal smoothing destroys curvature
- Use tangential Laplacian smoothing instead
- Mean curvature flow



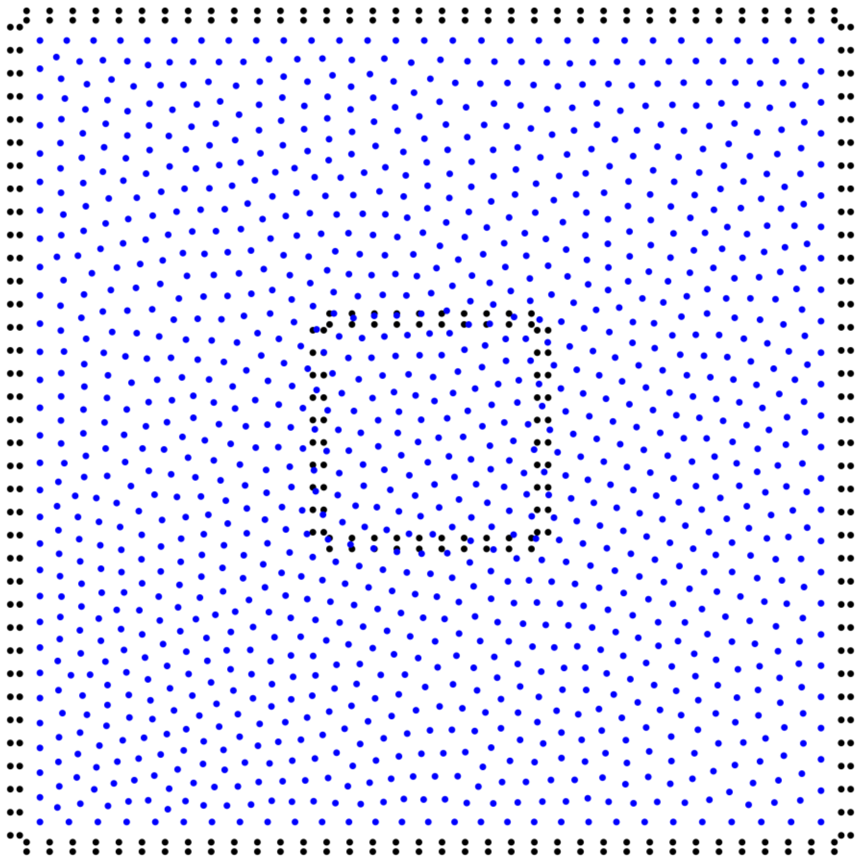
Interior samples: Random Sampling

Scatter points in the volume randomly

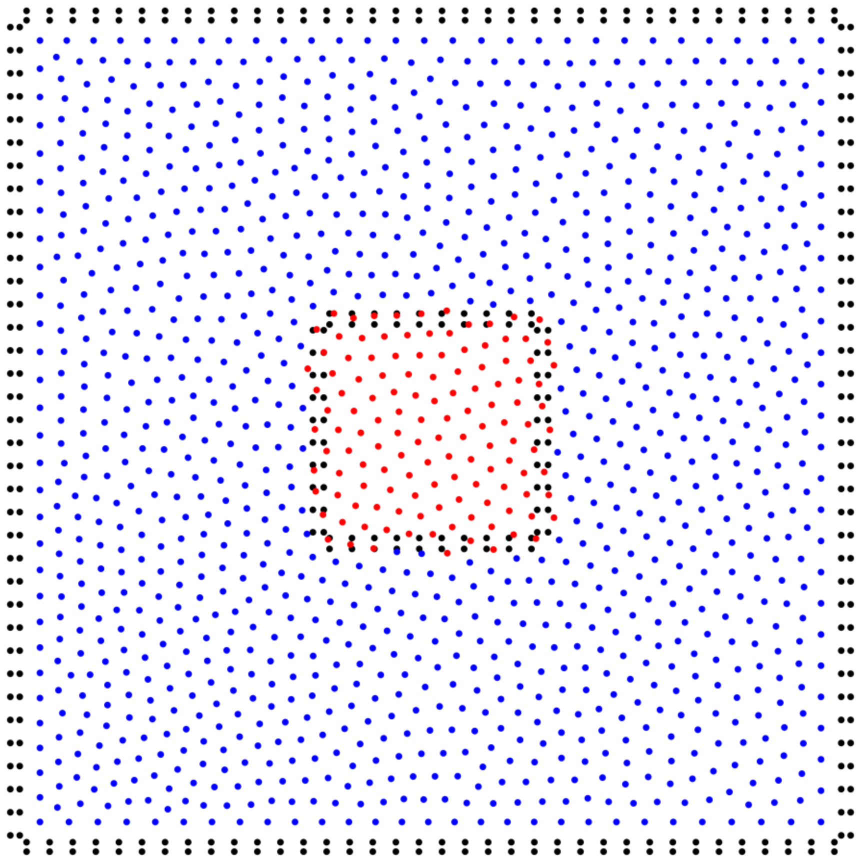
Interior samples: Random Sampling



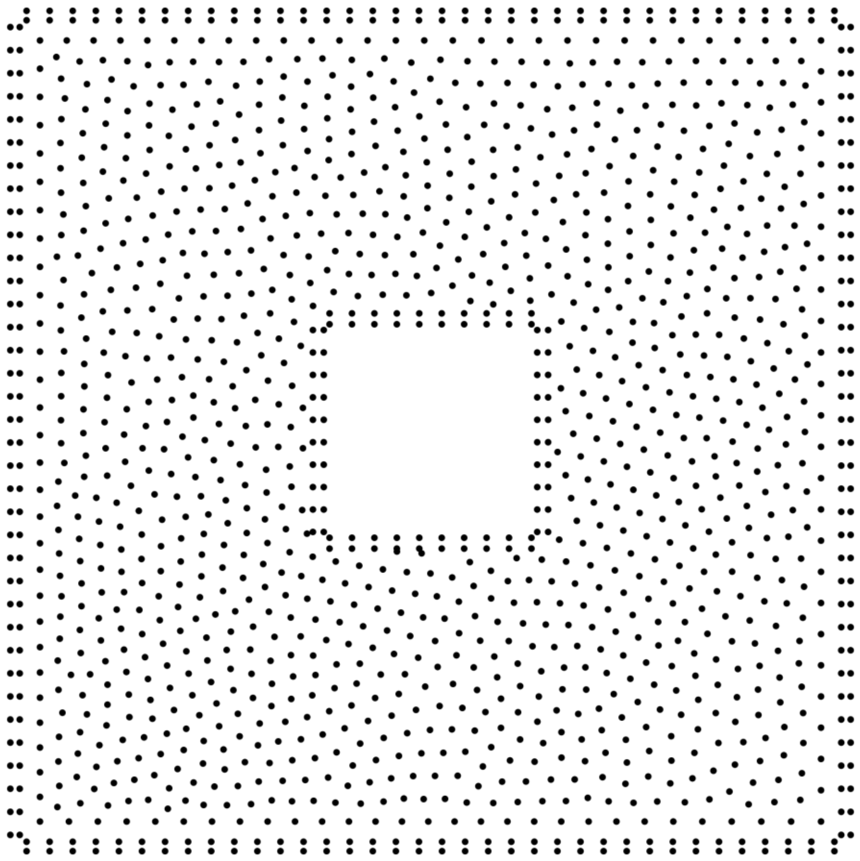
Interior samples: Random Sampling



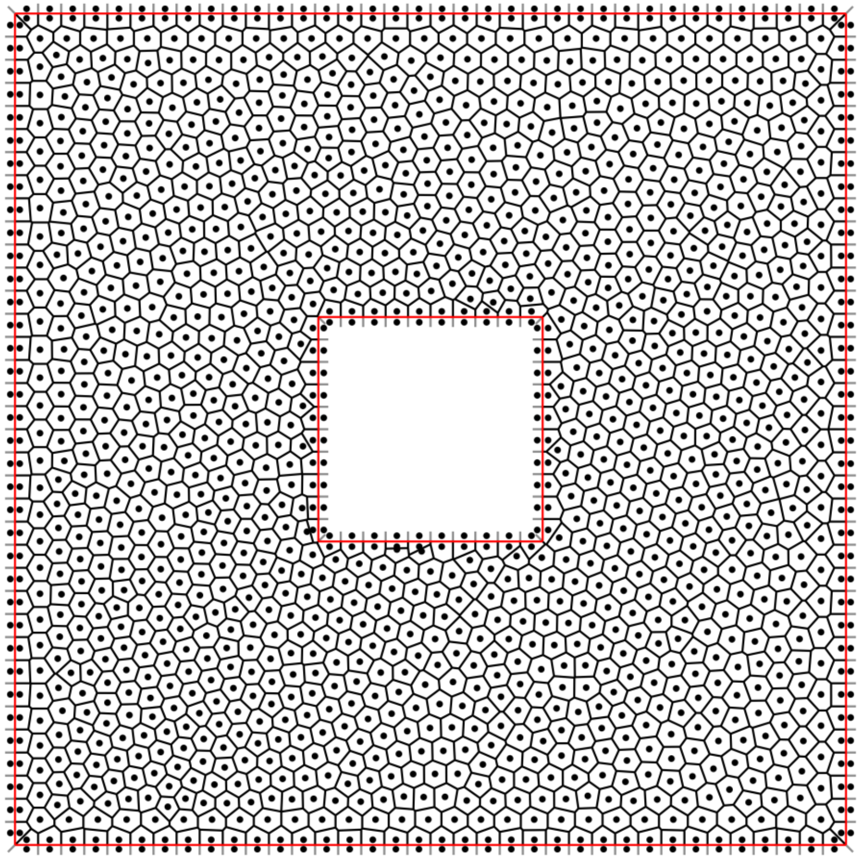
Interior samples: Random Sampling



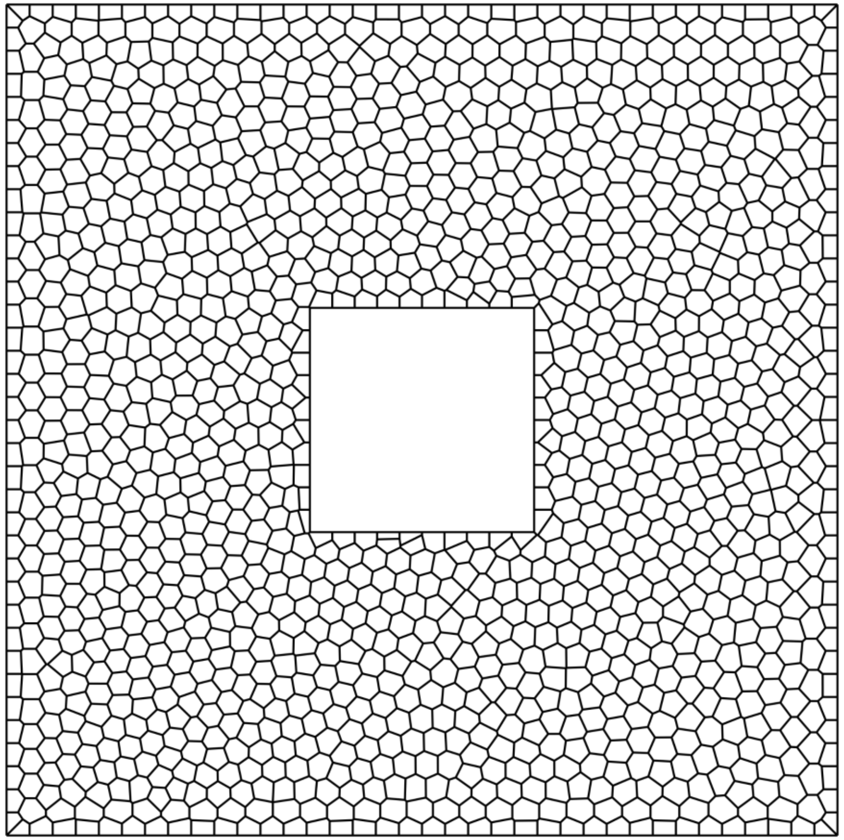
Interior samples: Random Sampling



Interior samples: Random Sampling



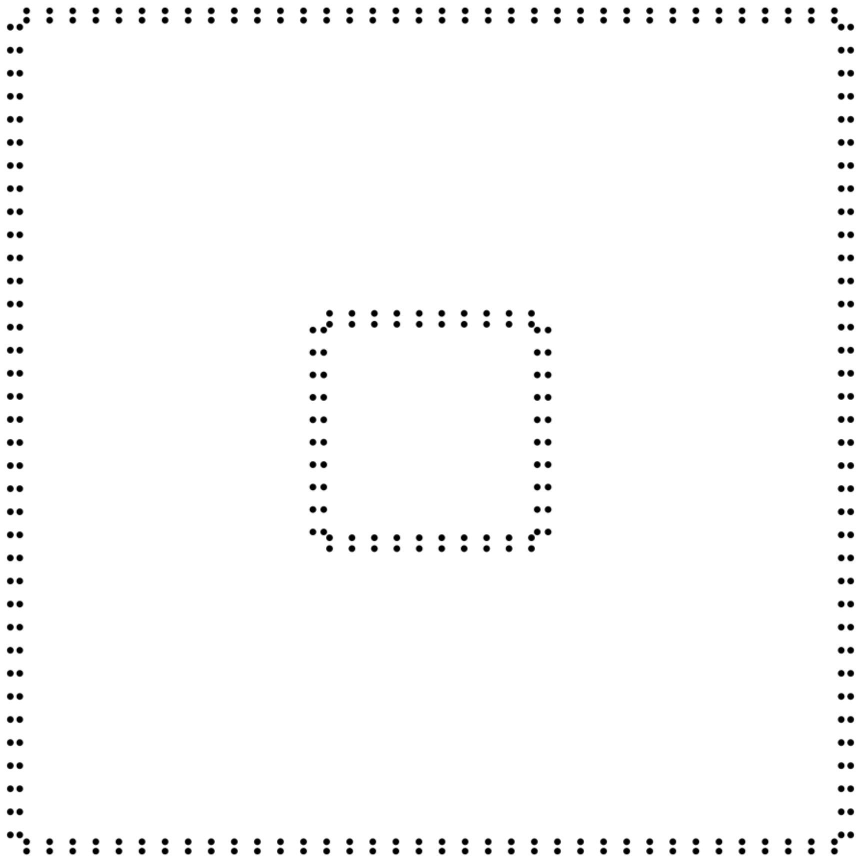
Interior samples: Random Sampling



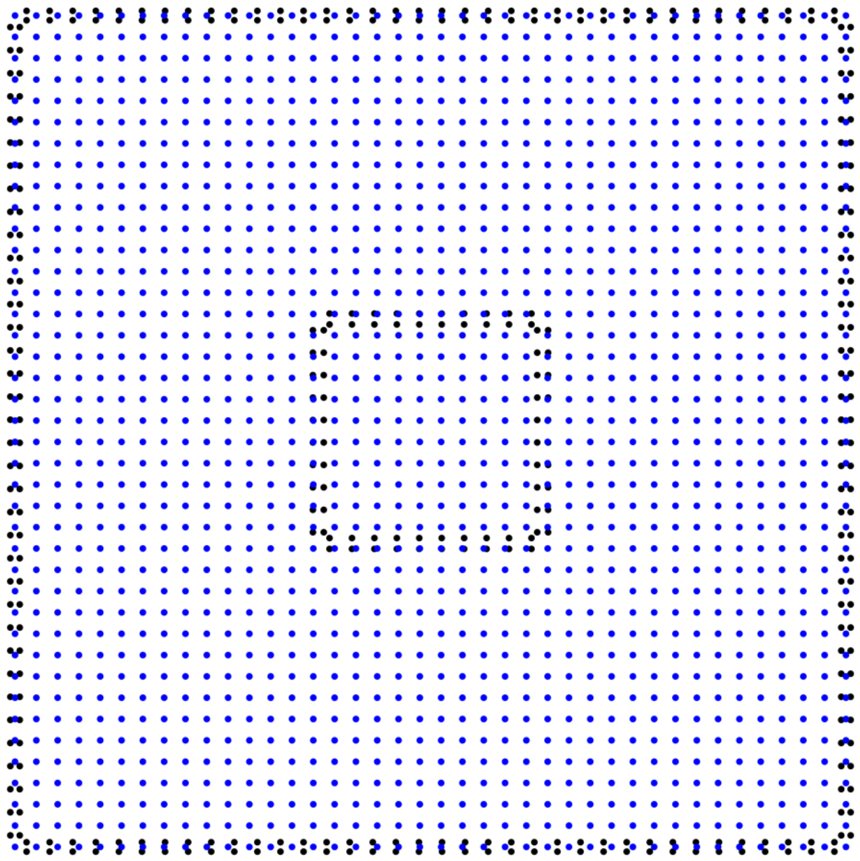
Interior samples: Hexahedral lattice

Place interior samples with uniform spacing, on a grid.

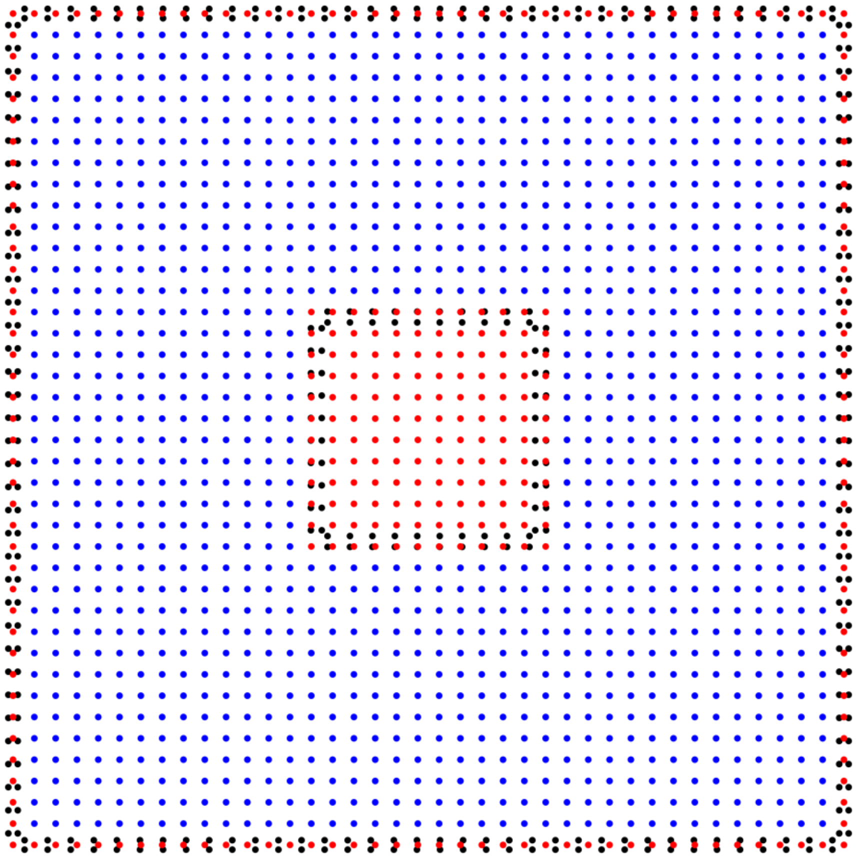
Interior samples: Hexahedral lattice



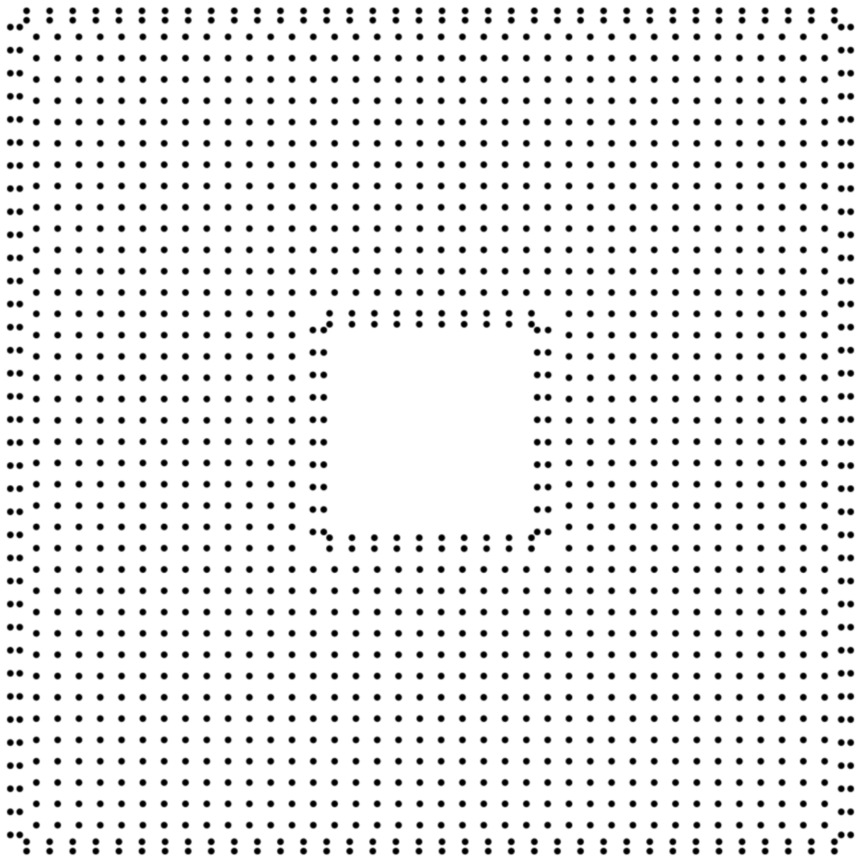
Interior samples: Hexahedral lattice



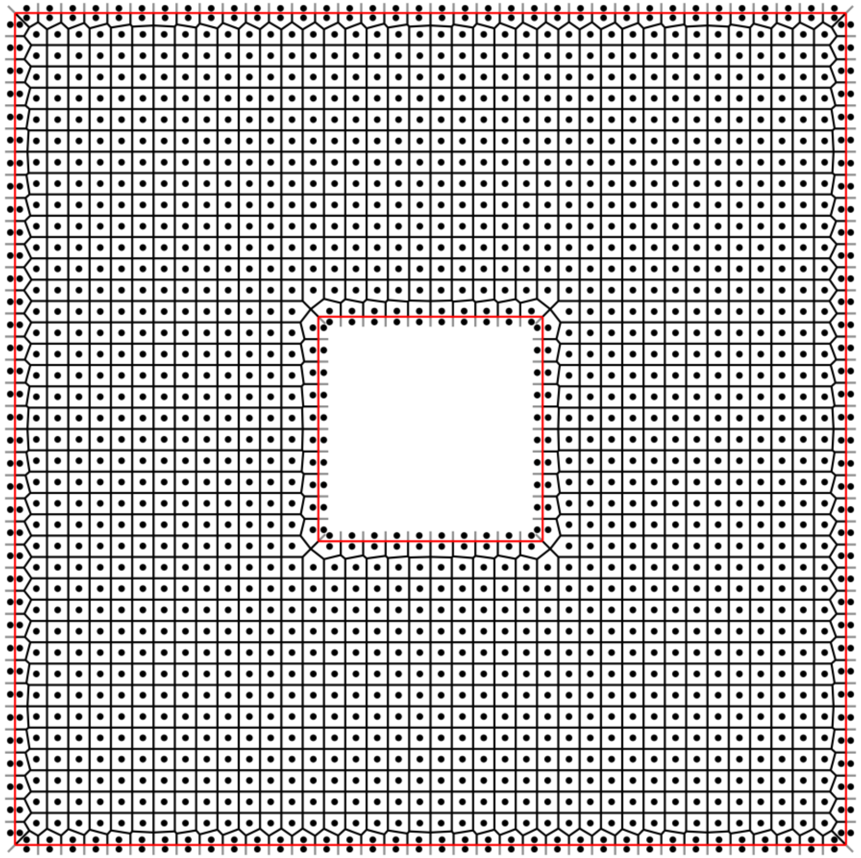
Interior samples: Hexahedral lattice



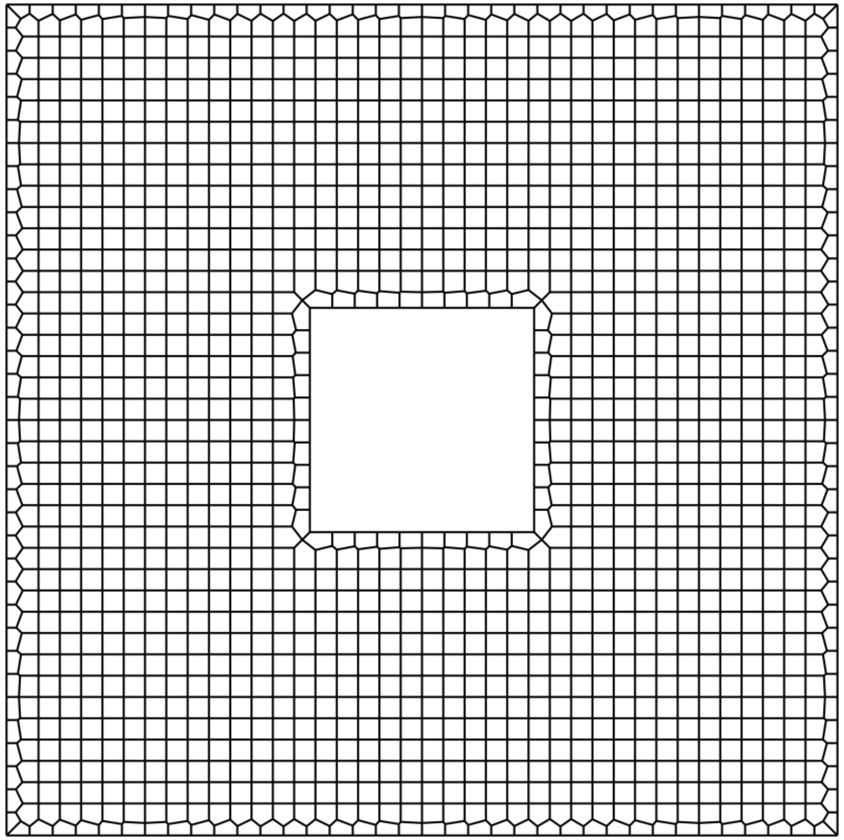
Interior samples: Hexahedral lattice



Interior samples: Hexahedral lattice



Interior samples: Hexahedral lattice



RESULTS

Implemented prototype

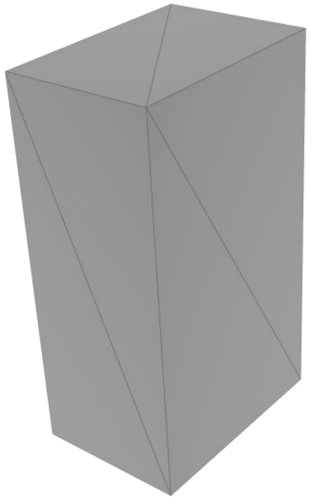
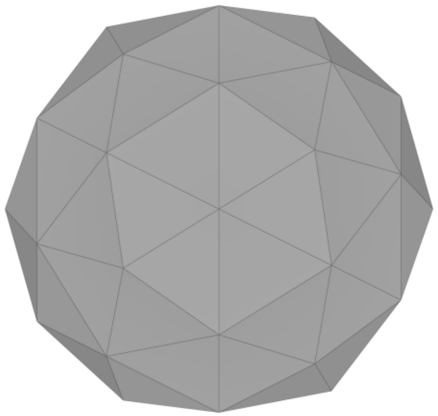
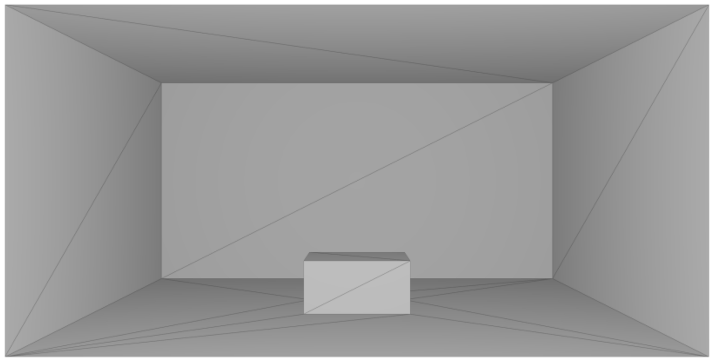
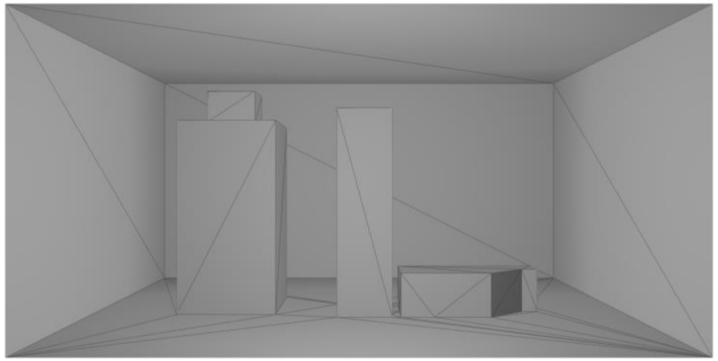
- *C++*
- Computational Geometry Algorithm Library (CGAL)
- Input: single `.obj` file with group information
- Output:
  - Voronoi mesh structure as `.obj` with patch group information
  - Seed points as a CSV file
  - Refined surface triangulation as `.obj`

Implemented prototype

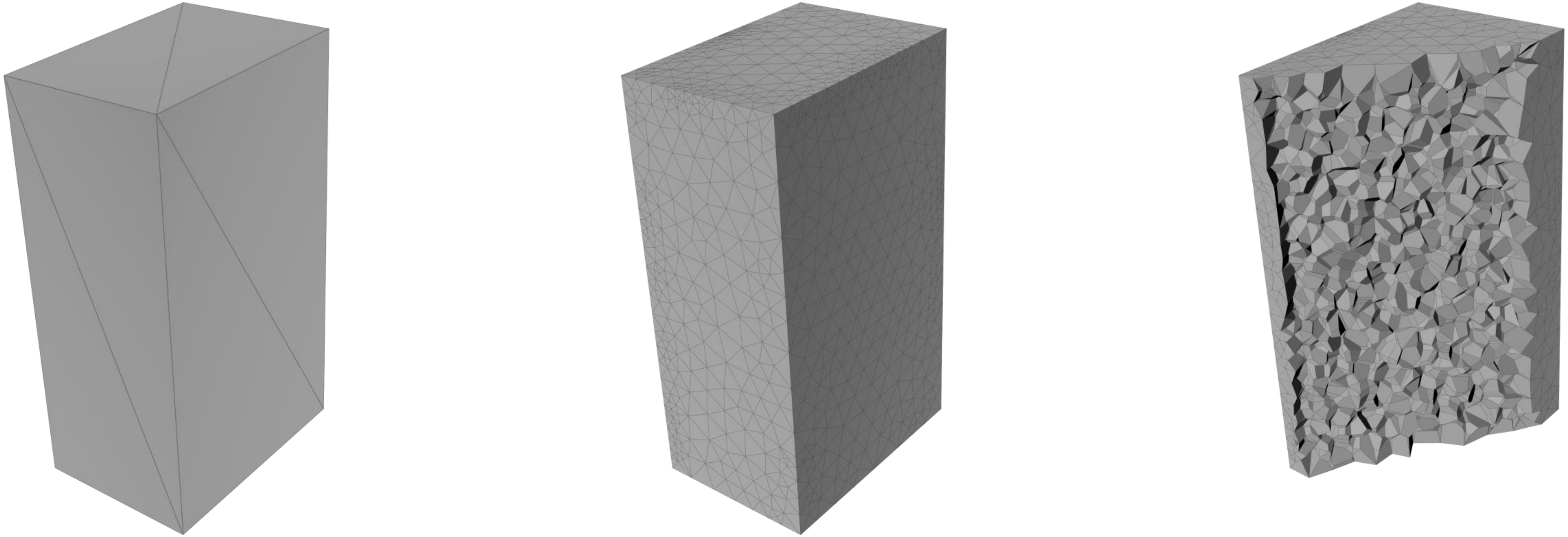
- Expects a `.obj` file structured as the output of City4CFD
- Refine surface
- Generate interior seeds
- Generate Voronoi mesh structure, tailored for OpenFOAM

TESTING THE PROTOTYPE

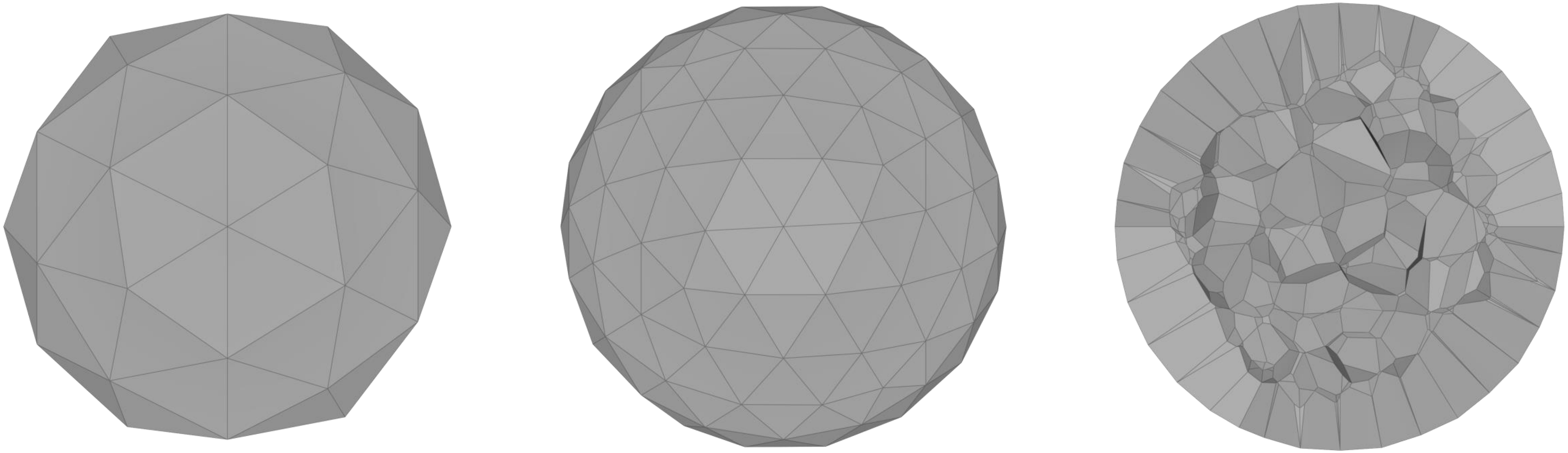
4 input geometries:

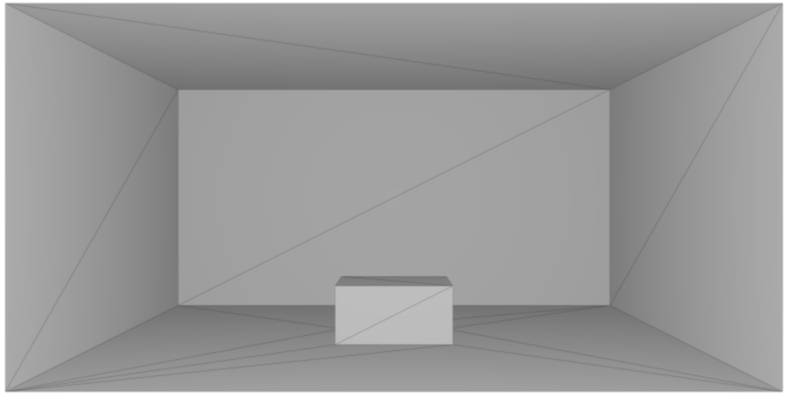
Testing the prototype: Box

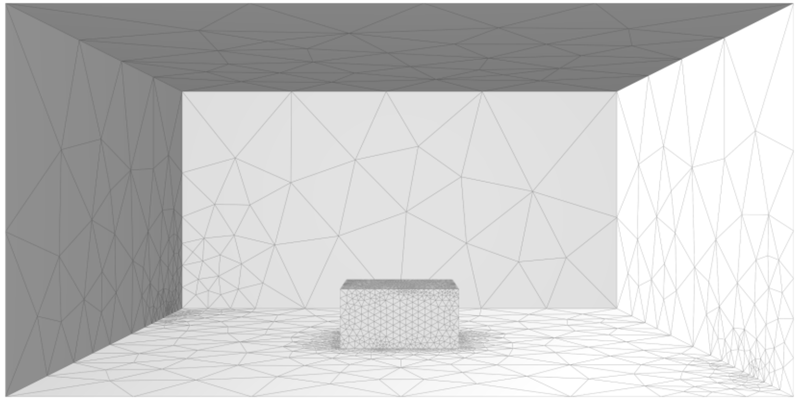


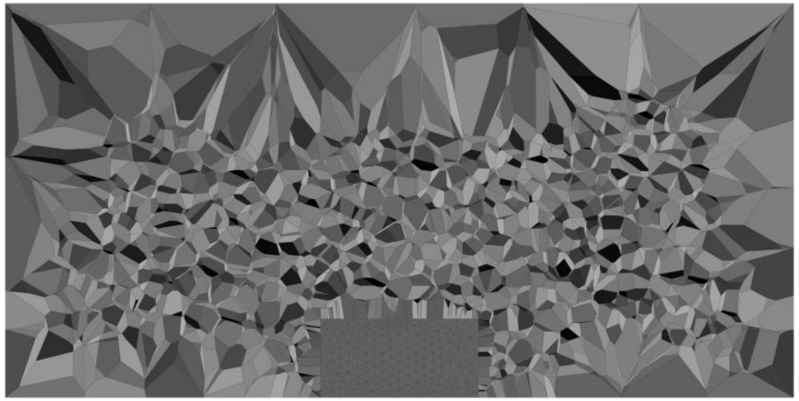
TESTING THE PROTOTYPE: SPHERE



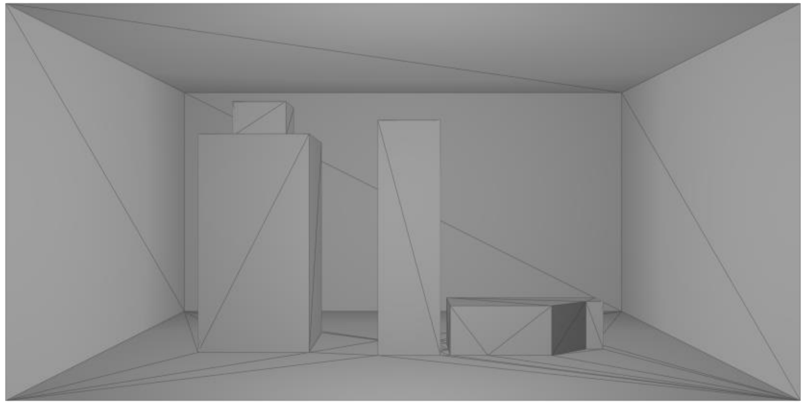
TESTING THE PROTOTYPE: ONE BUILDING

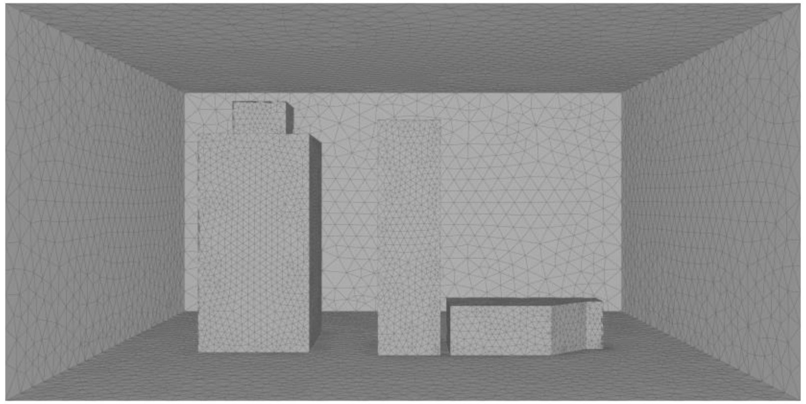


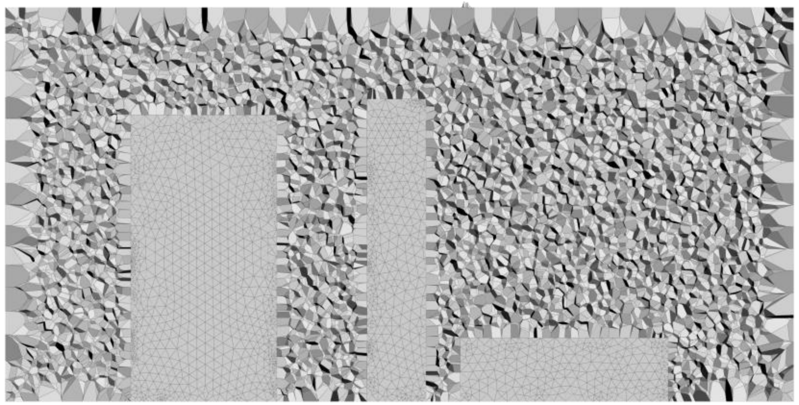




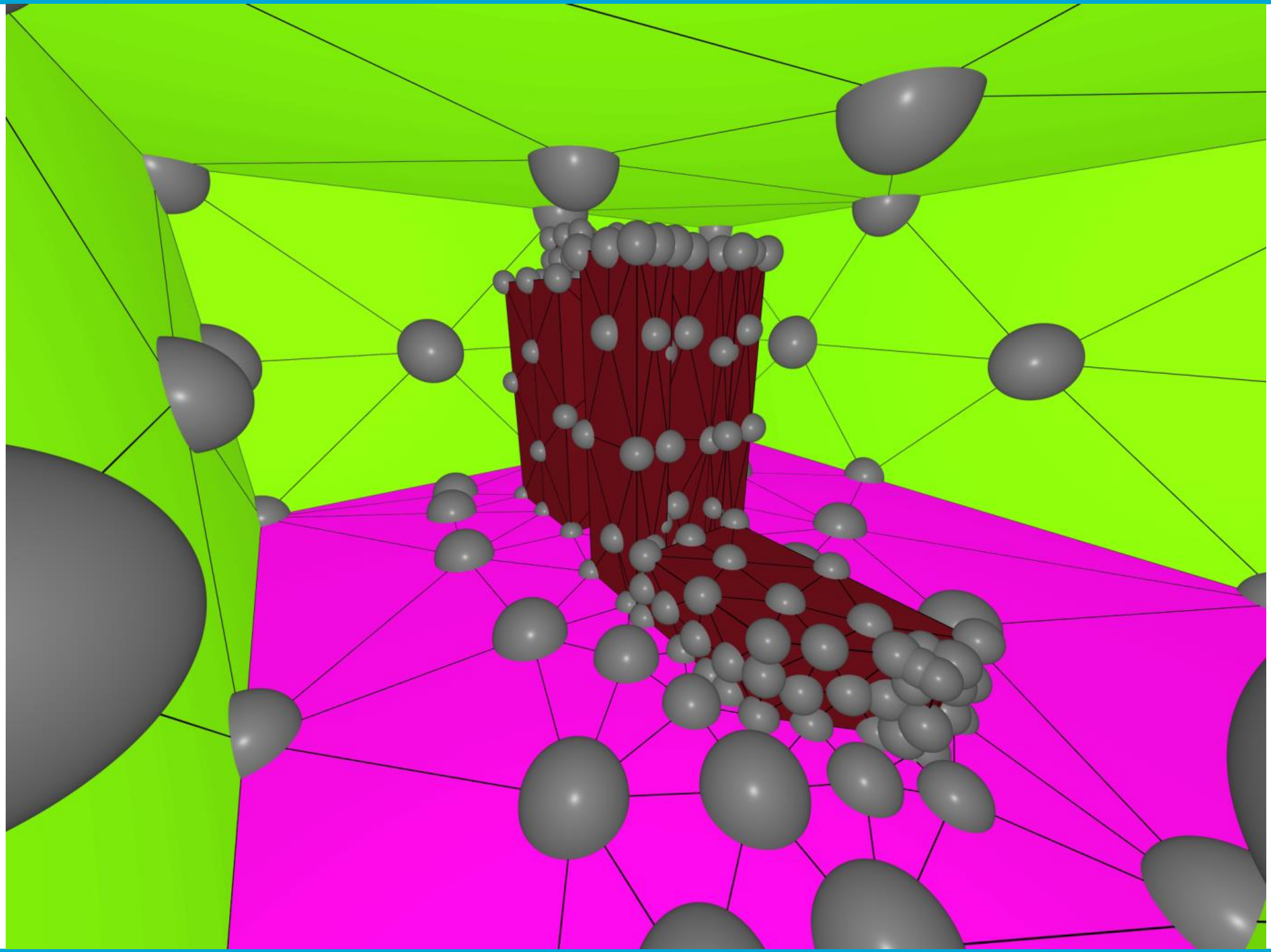
Testing the prototype: Three buildings



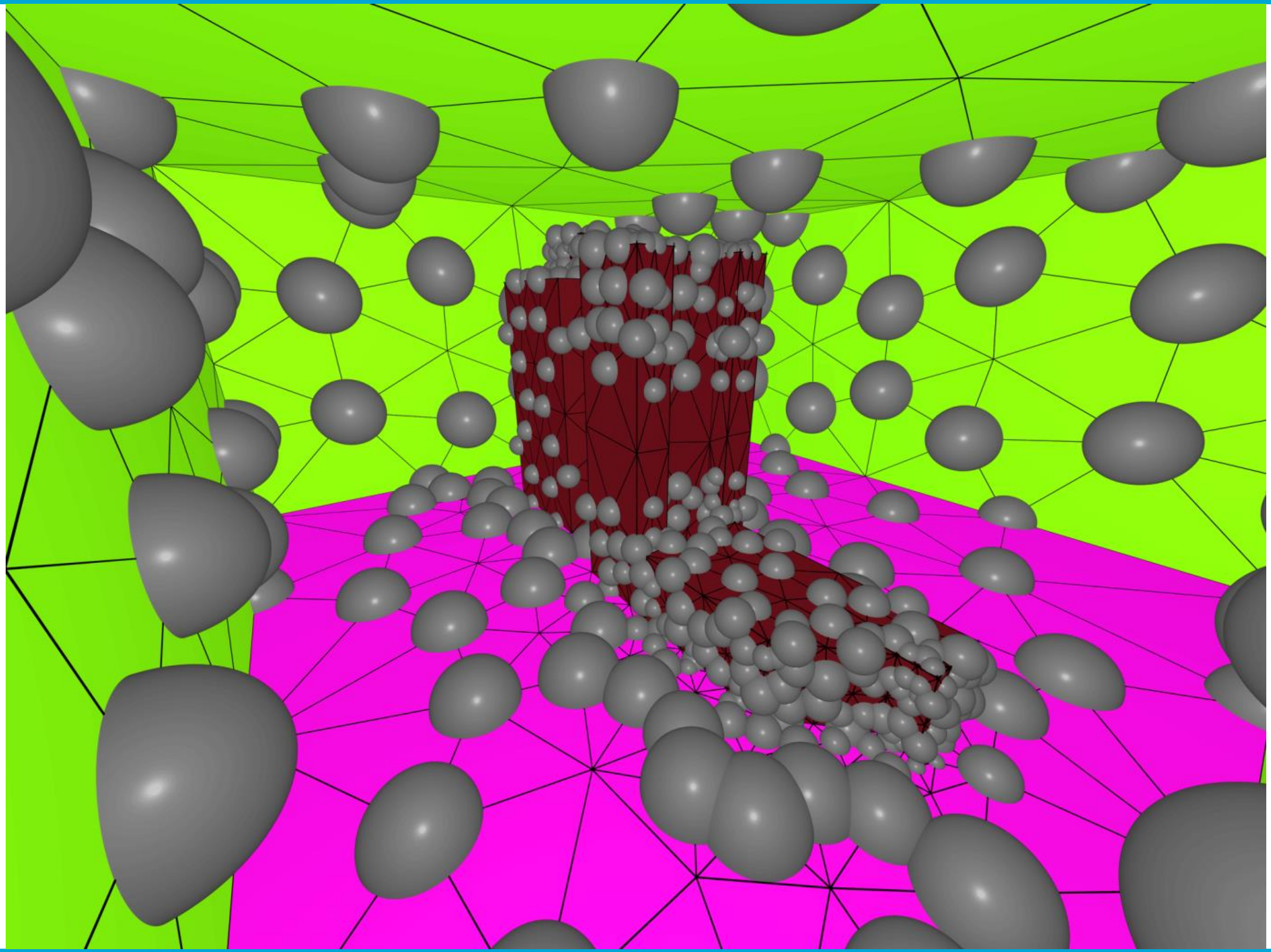




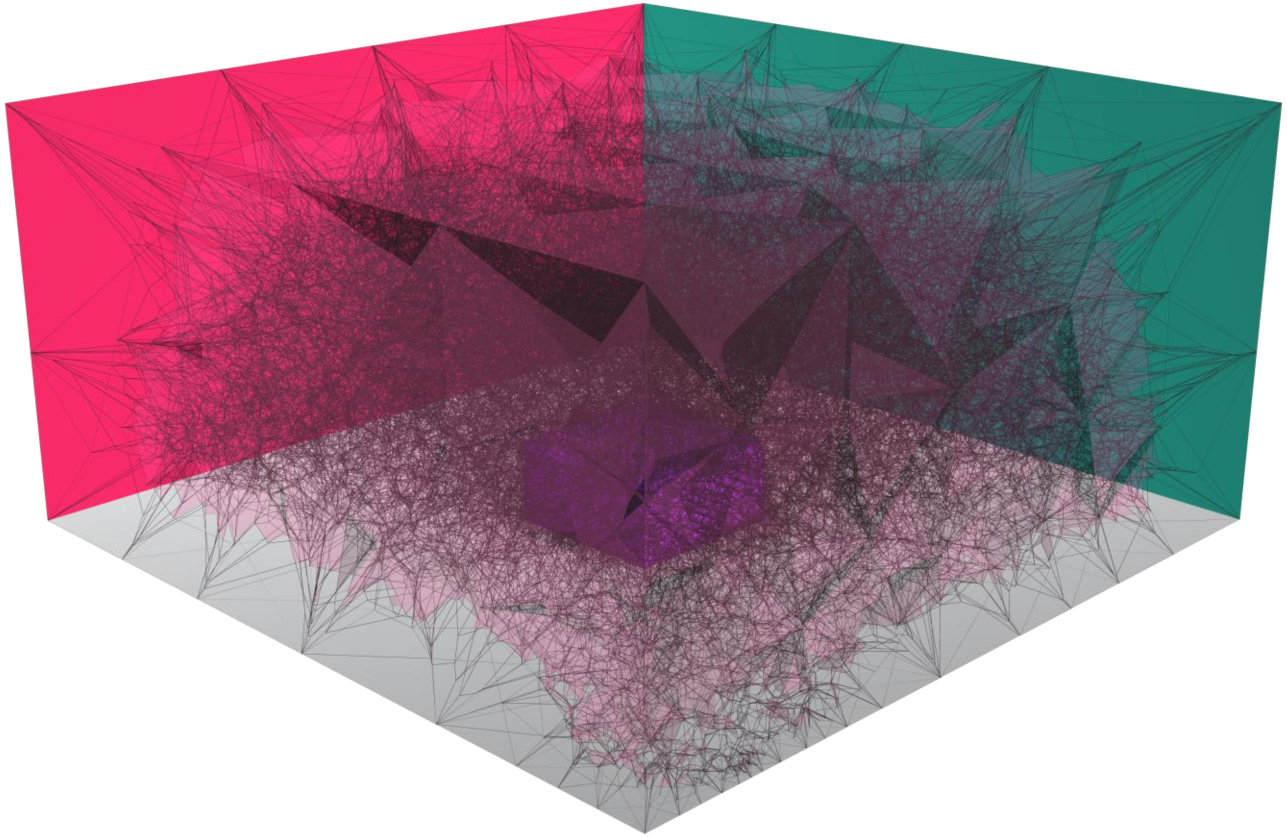
Testing the prototype: Refinement process



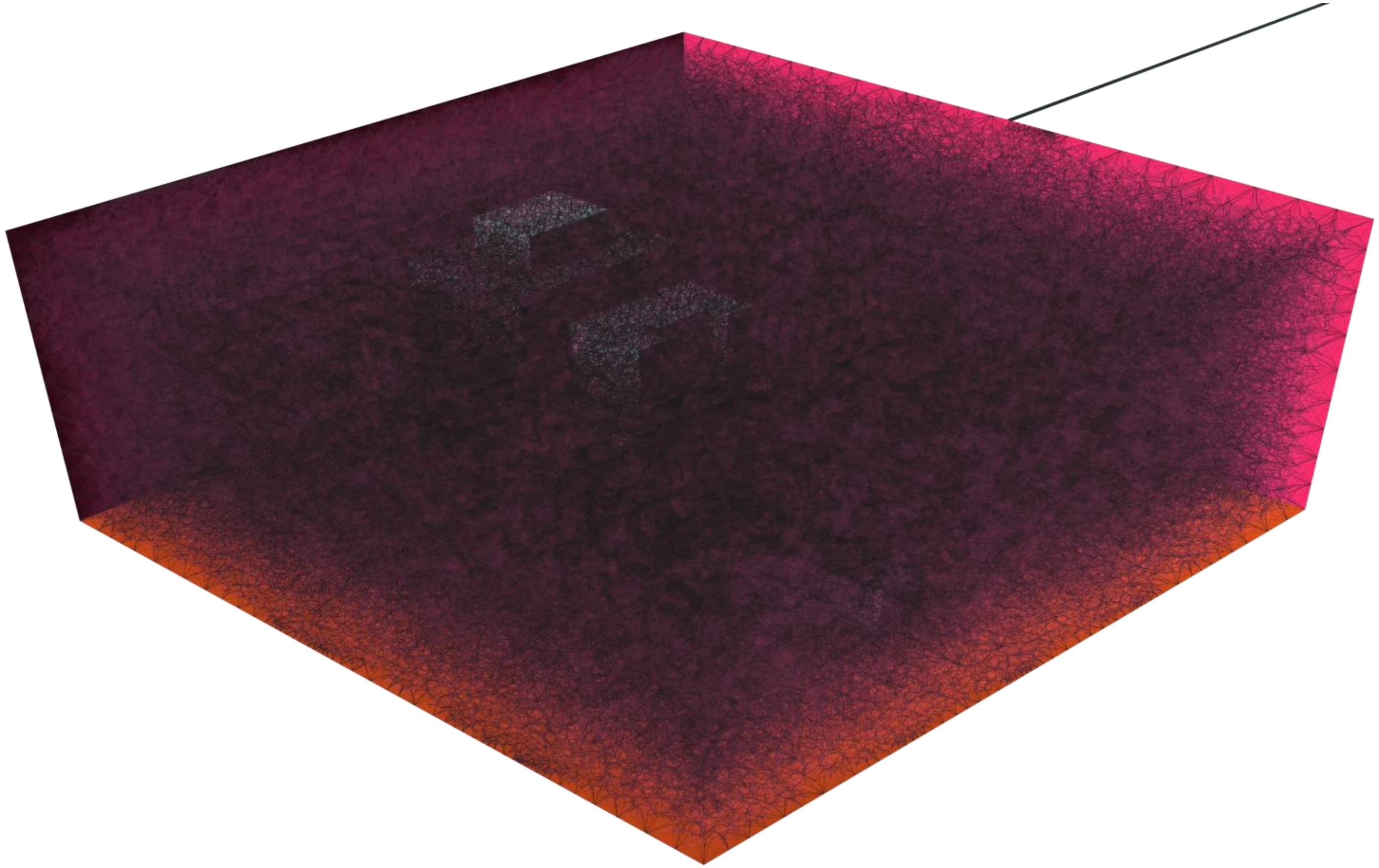
Testing the prototype: Refinement process



Testing the prototype: Patch types



Testing the prototype: Patch types



Limitations

- Only uniform sampling is implemented
- Tangential Laplacian implementation incorrectly handles certain cases leading to crashes
- Output Voronoi mesh occasionally constructed partially incorrectly
  - Inexact kernel
  - Floating point issues
  - Bug in code which uses incorrect indexing
- In OpenFOAM mesh structure but not OpenFOAM file format.

CONCLUSIONS & FUTURE WORK

CONCLUSIONS

- Successfully derives generators for a *Voronoi mesh* from *2-manifold* input PLCs which *conforms* to that boundary
- No clipping required
- Maintains patch types
- Generates Voronoi mesh tailored to be used in OpenFOAM
- Provides a solid base for Voronoi mesh generation for urban contexts

Improvements

- Upon convergence of surface refinement
  → Perfectly conforms
- If not converged
  → Conformity issues constrained to local patches

Future work

1. Fix bug in generation of the Voronoi structure.
2. Output mesh in OpenFOAM format directly.
3. Implement uniform interior sampling.
4. Implement density regulations for interior samples.
5. Empirically compare performance with `snappyHexMesh`.

THANK YOU FOR YOUR ATTENTION!

Title: *VORONOI MESH GENERATION TAILORED FOR URBAN FLOW SIMULATIONS*

Presenter: *ÁKOS S. SÁRKÁNY*