TU Graz

Splitting a Vertex

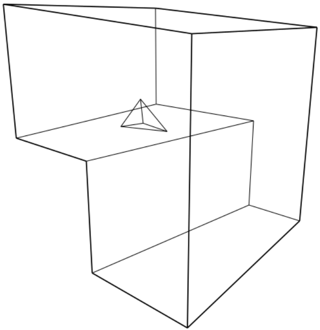
Franz Aurenhammer, Daniel Lederer,
Institute of Theoretical Computer Science

10 - 13 October, 2022

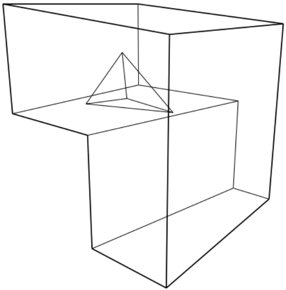
Introduction

- We are interested in straight skeletons of (non-convex) polytopes in $\mathbb{R}^3$
- Straight skeleton structure is defined by a mitered boundary offsetting process
- Shrinking a polytope in a self-parallel way until it vanishes
- Each component of the polytope traces out a certain component of the skeleton

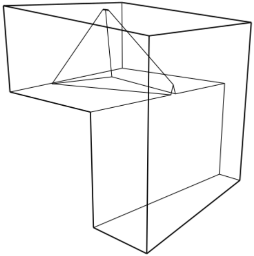
Example



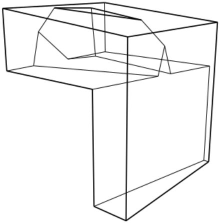
Example



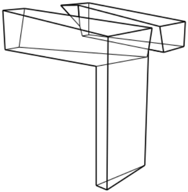
Example



Example

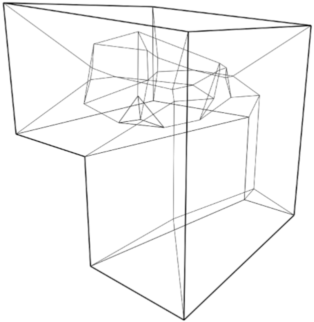


Example



Example

Example

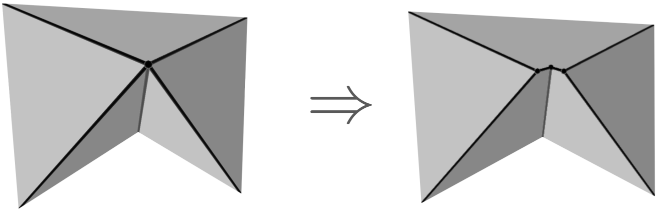


Offsetting polyhedral surface in $\mathbb{R}^3$

- Polytope vertices of degree $\geq 4 \Rightarrow$ split into degree-3 vertices

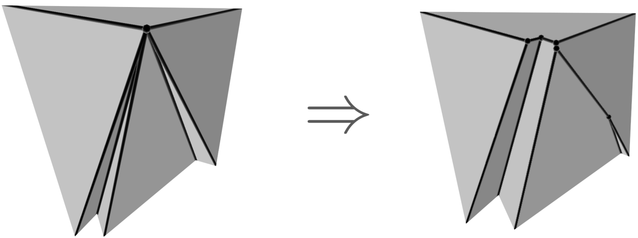
Offsetting polyhedral surface in $\mathbb{R}^3$

- Polytope vertices of degree $\geq 4 \Rightarrow$ split into degree-3 vertices



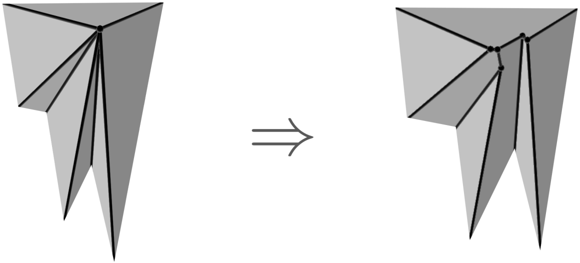
Offsetting polyhedral surface in $\mathbb{R}^3$

- Polytope vertices of degree $\geq 4 \Rightarrow$ split into degree-3 vertices



Offsetting polyhedral surface in $\mathbb{R}^3$

- Polytope vertices of degree $\geq 4 \Rightarrow$ split into degree-3 vertices



Offsetting polyhedral surface in $\mathbb{R}^3$

- Polytope vertices of degree $\geq 4 \Rightarrow$ split into degree-3 vertices
- Initially, all (high-degree) vertex splittings will happen simultaneously (= initial events)
- Later, during shrinking process $\Rightarrow$ only low-degree vertices, in generic case (= non-initial events)
  - Handled in the same way as initial events

Offsetting polyhedral surface in $\mathbb{R}^3$

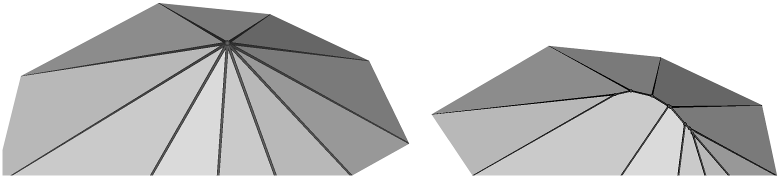
- Consider each vertex separately
- Vertex v of degree $k \Rightarrow k$ incident facets
- The k planes supporting these facets are offset by 1 unit towards the interior of the polytope
- Locally build a new *valid* surface from these offset planes

Vertex types

- In general, v can have a complex neighbourhood

Vertex types

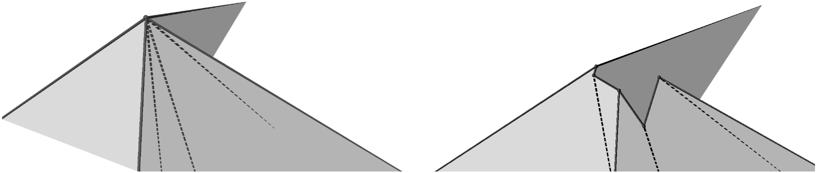
- Convex case (easiest case): all incident edges of *v* are convex



A degree-10 convex vertex and its offset surface

Vertex types

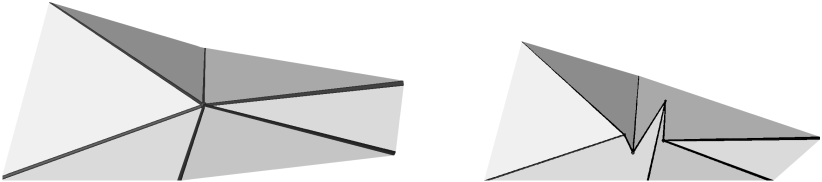
- Pointed case: *v* can be cut off from the polytope by a single plane



A degree-7 pointed vertex and its offset surface

Vertex types

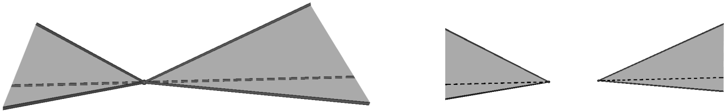
- Saddle point case: incident edges of *v* span 3-space



A degree-6 saddle vertex and its offset surface

Vertex types

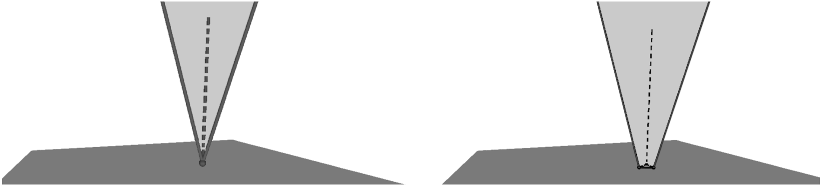
- Touching case: ϵ-sphere centered at v intersects polytope boundary in disconnected set



A degree-6 touching vertex and its offset surface

Vertex types

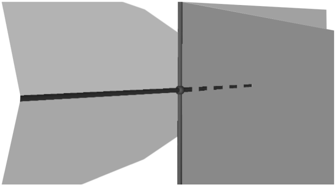
- Touching case: ϵ-sphere centered at v intersects polytope boundary in disconnected set

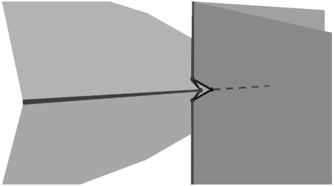


A degree-4 touching vertex and its offset surface

Vertex types

- Touching case: ϵ-sphere centered at *v* intersects polytope boundary in disconnected set





A degree-4 touching vertex and its offset surface

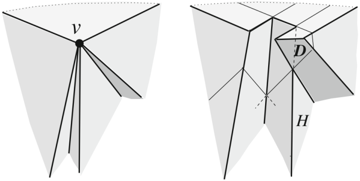
Valid offset surface

- Computing an offset surface is highly non-trivial
- What is a *valid* offset surface?

(1) Its facets must come from the k offset planes

(2) It must *fit* to the shrunk polytope

(3) Each of its facets must be fully visible from v



Conditions (1) and (2) are obviously necessary. Condition (3) is needed, too, because certain facets would move to the exterior of the polytope, otherwise.

Valid offset surface

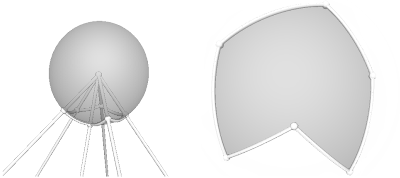
- Not unique in general
- There are exponential many valid offset surfaces, in k
- All valid offset surfaces are contained in the so-called offset arrangement $\mathcal{A}(v)$

Recap: Offset arrangement $\mathcal{A}(v)$

- Dissection of the space by the k offset planes for v
- Representation of intersections: Vertices, edges, facets, cells (*see slides from last time for more details*)
- Combinatorial structure of $\mathcal{A}(v)$ is independent from the offset distance
- $\mathcal{A}(v)$ can be used to prove that a valid offset surface always has to exist:

Existence proof for a valid offset surface

- Arrangement cell is *relevant* $\Leftrightarrow$ its radial projection from v fully lies within the *spherical polygon* S for v
- S is created by intersecting a (small) sphere, centered at v, with the polytope
- *See slides from last time for more details*



Existence proof for a valid offset surface

- Take all relevant unbounded cells of $\mathcal{A}(v)$
- If boundary of their union contains **unbounded** facets only $\Rightarrow$ valid offset surface (everything visible from v)
- Otherwise, surface contains bounded facets (**orphan facets**)
    - Add bounded cells that fit to such orphans until all facets become unbounded $\Rightarrow$ valid offset surface
    - This is always possible by the structure of $\mathcal{A}(v)$

Algorithmic viewpoint

- Proof above gives an algorithm for constructing valid offset surfaces:
  - For a vertex v with degree > 3, compute all cells of $\mathcal{A}(v)$
  - Merge all relevant unbounded cells
  - If resulting surface contains orphan facets, then add (relevant) bounded cells until no orphan facets remain

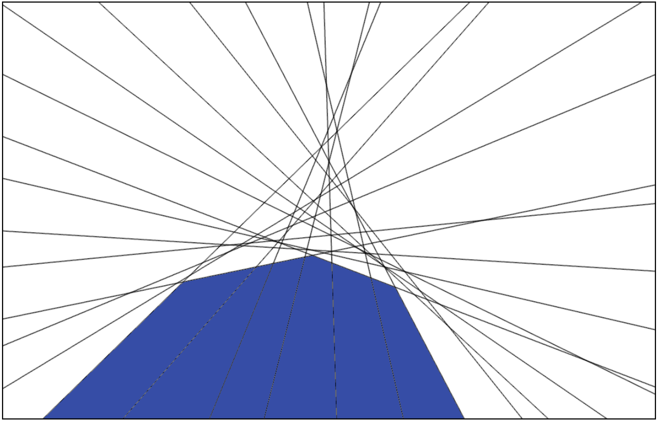
Software demonstration

- In the following examples, the cell adding process stops when the first valid offset surface is found:
  - `saddle_3.obj`
  - `kiev_17.obj`
  - `saddle_20.obj`
  - `convex_spinning_tops.obj`
  - `convex_spinning_tops_inverse.obj`

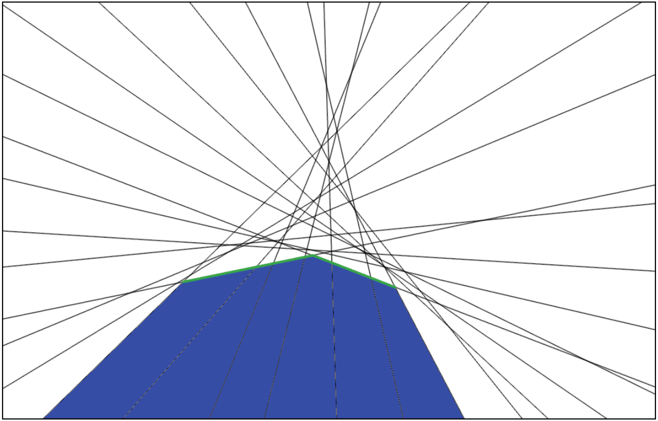
Problem: Cell Adding Process

- **Order** of adding cells matters
- If cell order is chosen unluckily
  - We cannot get rid of orphan facets
  - Non-valid offset surface
- The higher the degree of v, the more cell candidates there can be
  - Higher chance for wrong cell adding order

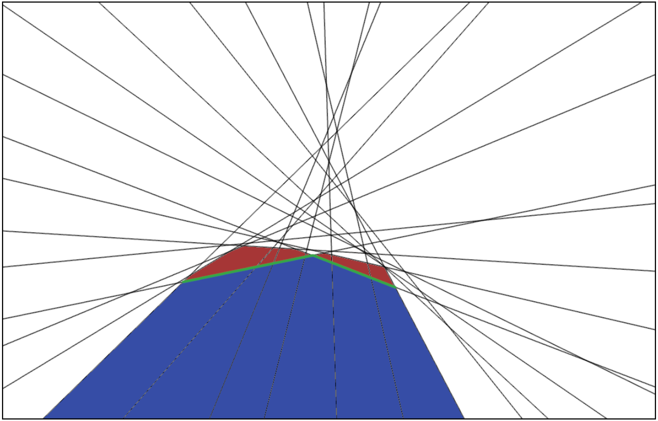
Problem example in $\mathbb{R}^2$



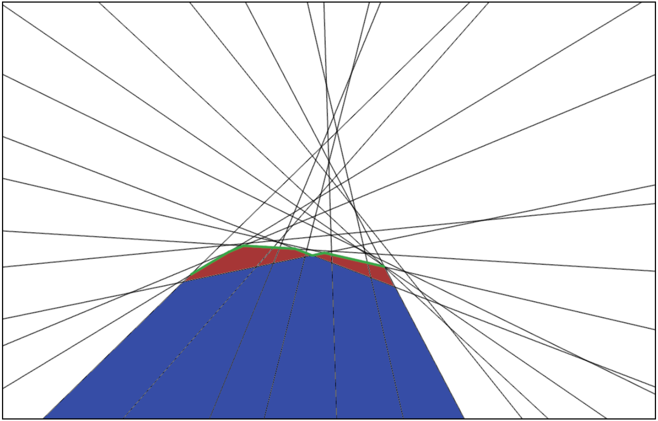
Problem example in $\mathbb{R}^2$



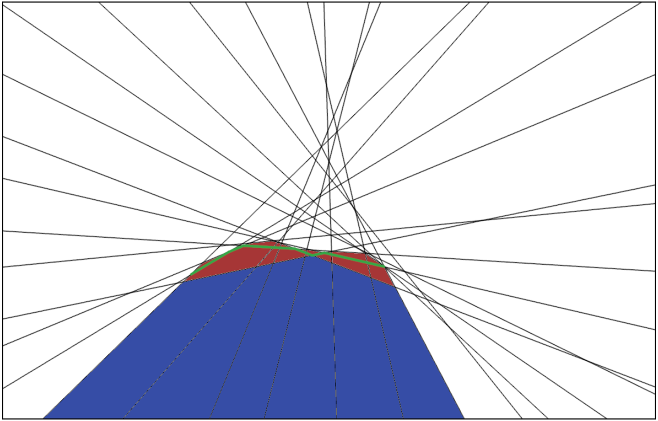
Problem example in $\mathbb{R}^2$



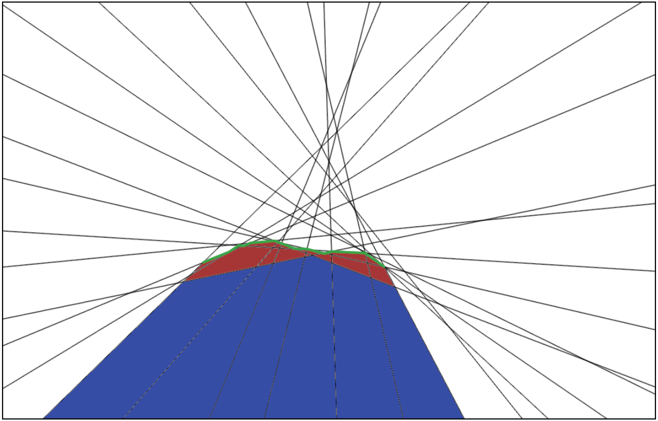
Problem example in $\mathbb{R}^2$



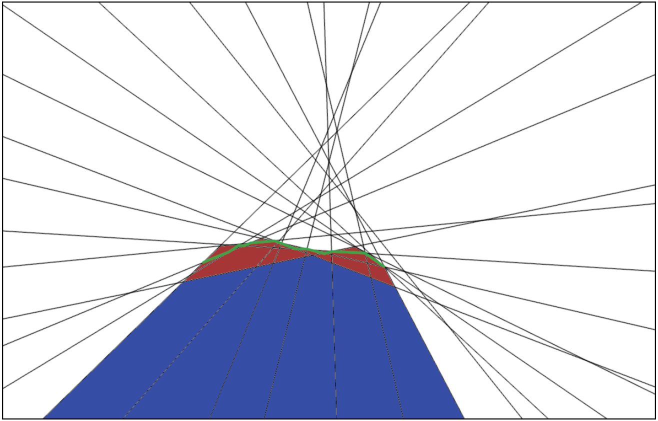
Problem example in $\mathbb{R}^2$



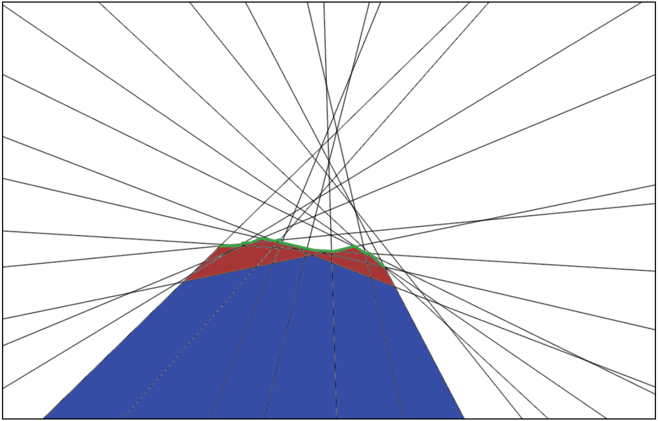
Problem example in $\mathbb{R}^2$



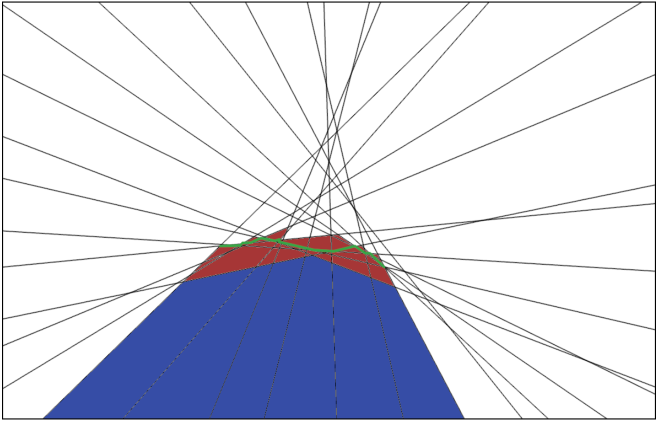
Problem example in $\mathbb{R}^2$



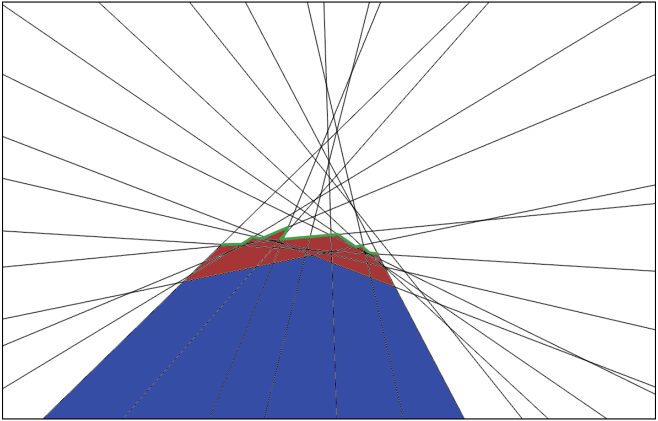
Problem example in $\mathbb{R}^2$



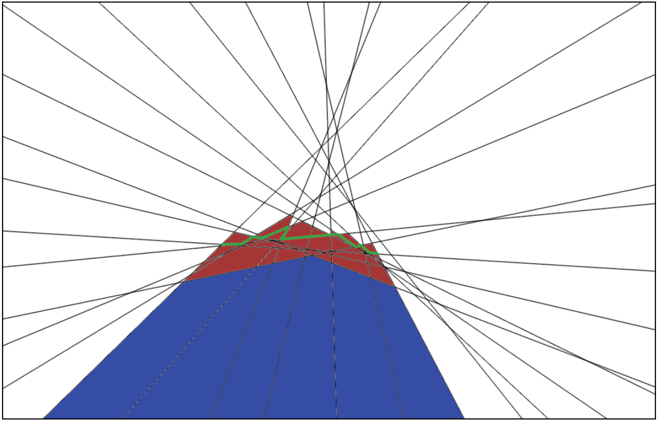
Problem example in $\mathbb{R}^2$



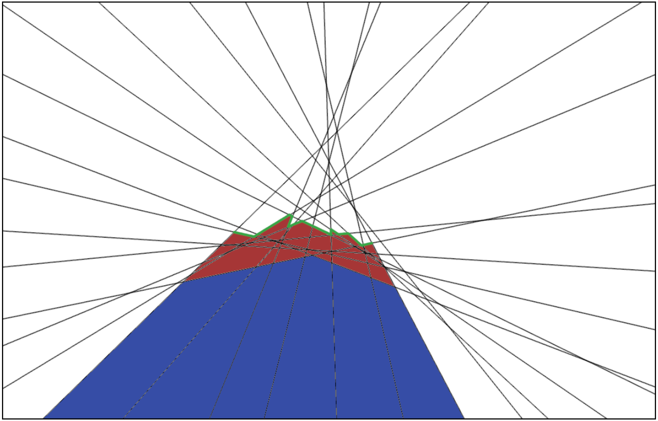
Problem example in $\mathbb{R}^2$



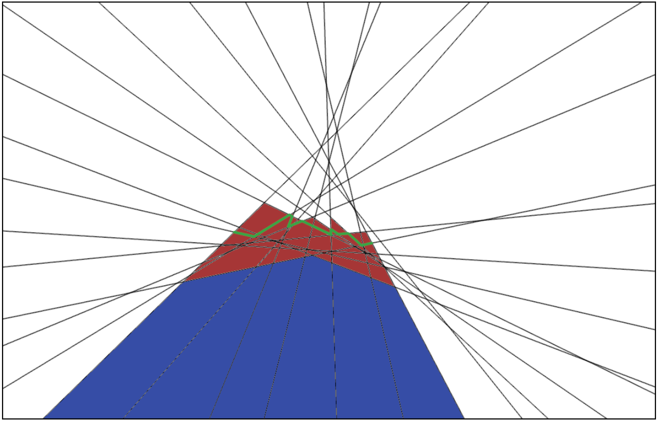
Problem example in $\mathbb{R}^2$



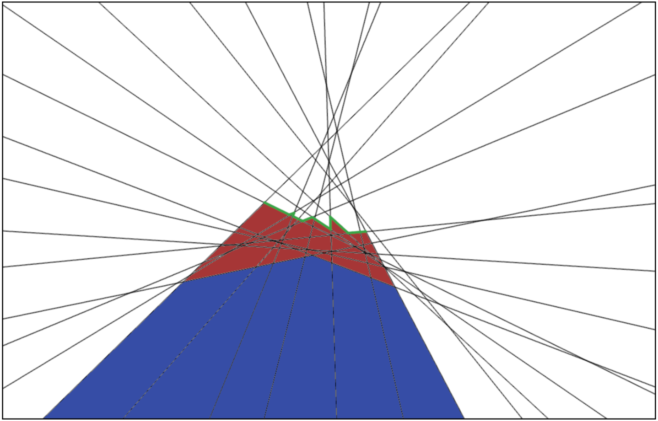
Problem example in $\mathbb{R}^2$



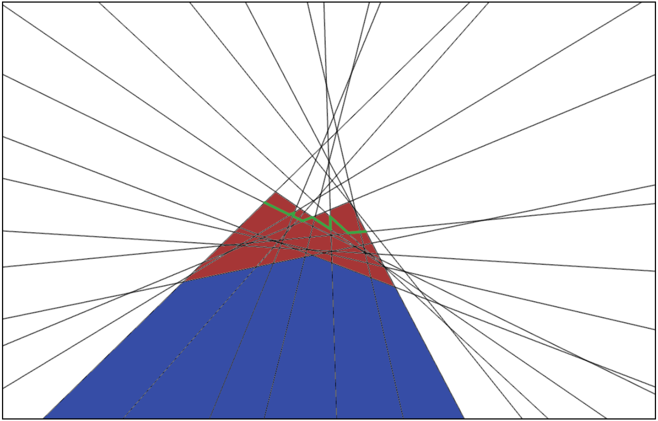
Problem example in $\mathbb{R}^2$



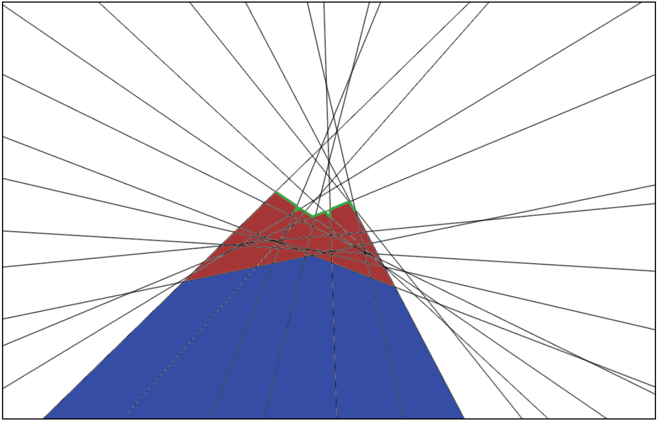
Problem example in $\mathbb{R}^2$



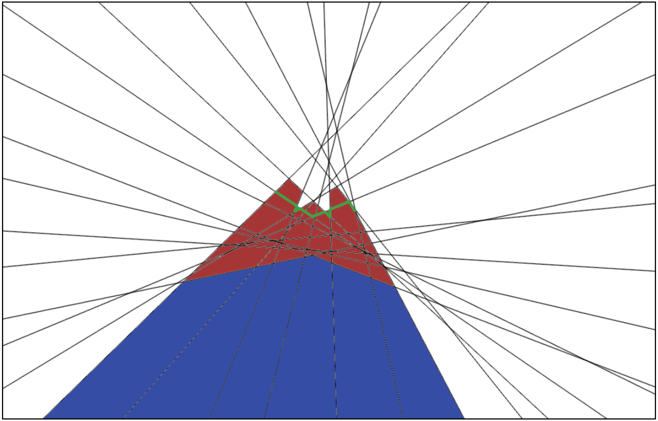
Problem example in $\mathbb{R}^2$



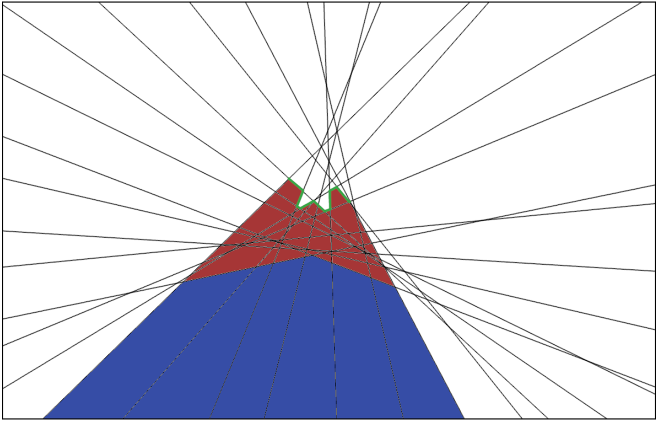
Problem example in $\mathbb{R}^2$



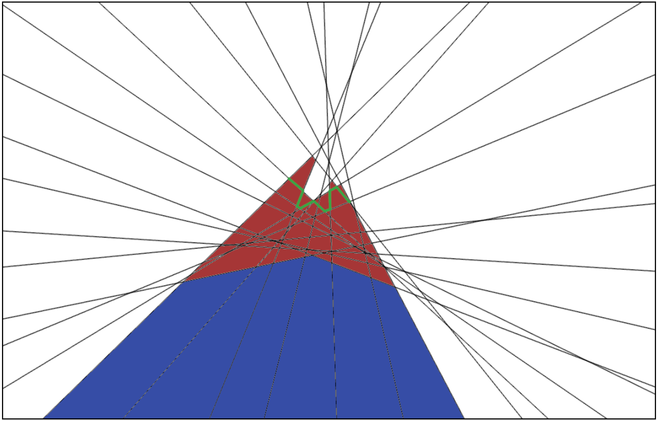
Problem example in $\mathbb{R}^2$



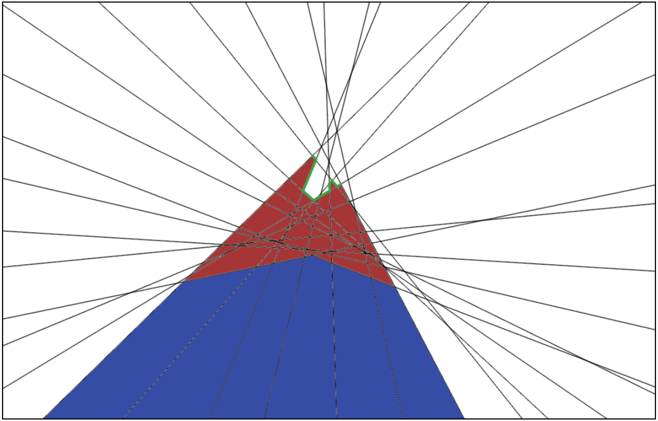
Problem example in $\mathbb{R}^2$



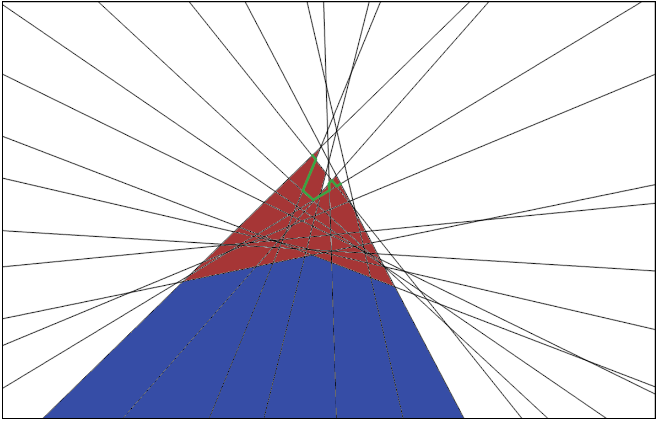
Problem example in $\mathbb{R}^2$



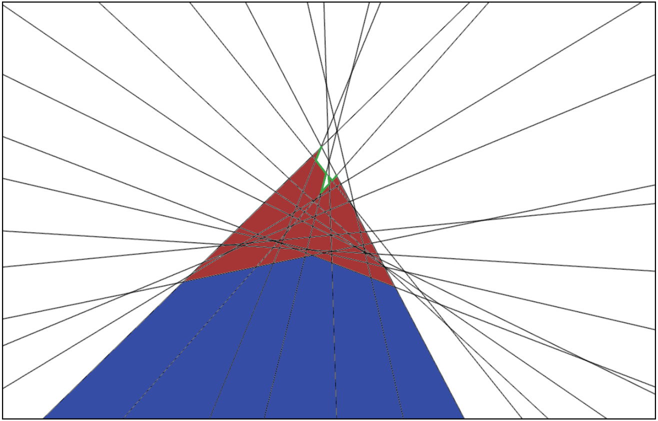
Problem example in $\mathbb{R}^2$



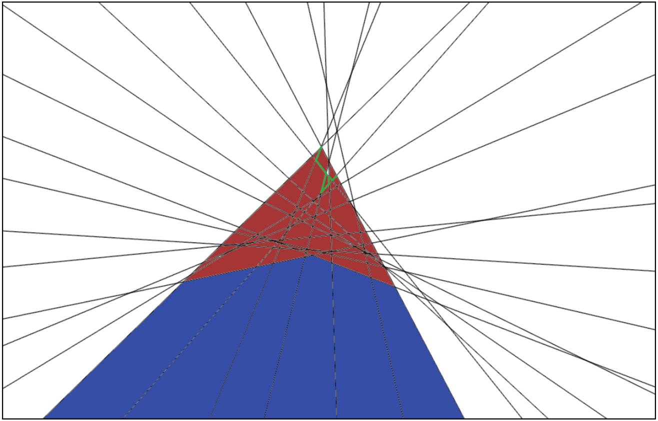
Problem example in $\mathbb{R}^2$



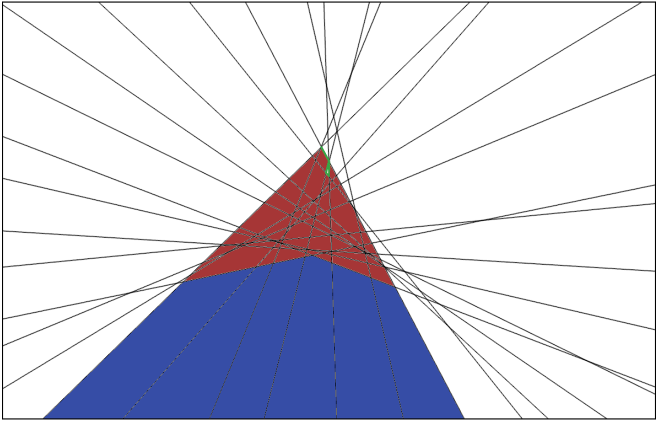
Problem example in $\mathbb{R}^2$



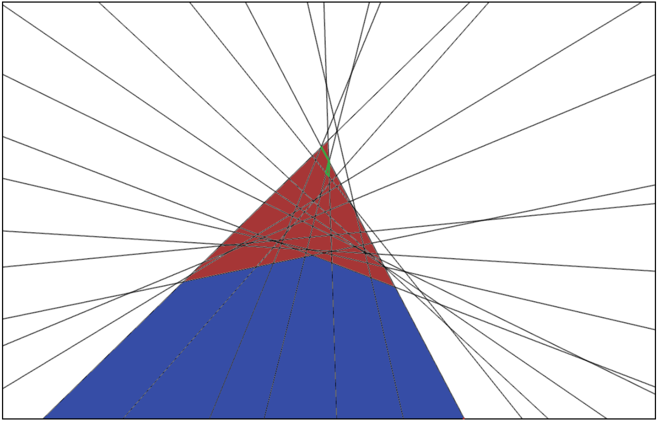
Problem example in $\mathbb{R}^2$



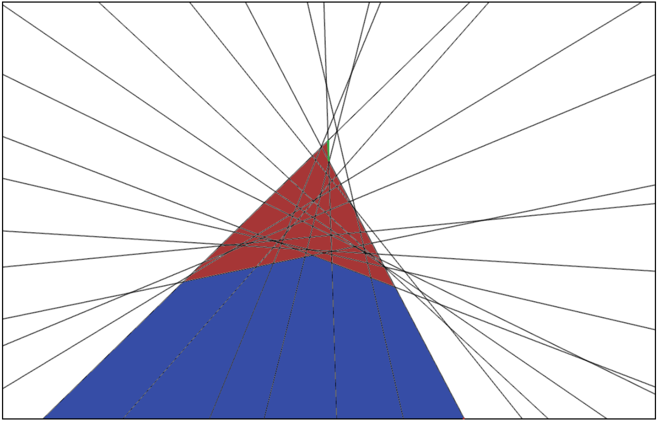
Problem example in $\mathbb{R}^2$



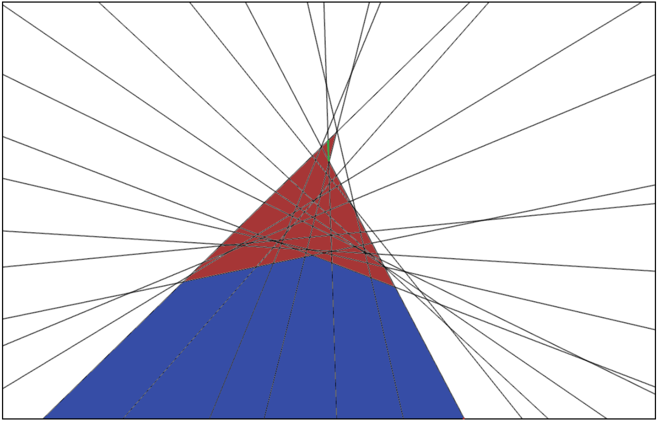
Problem example in $\mathbb{R}^2$



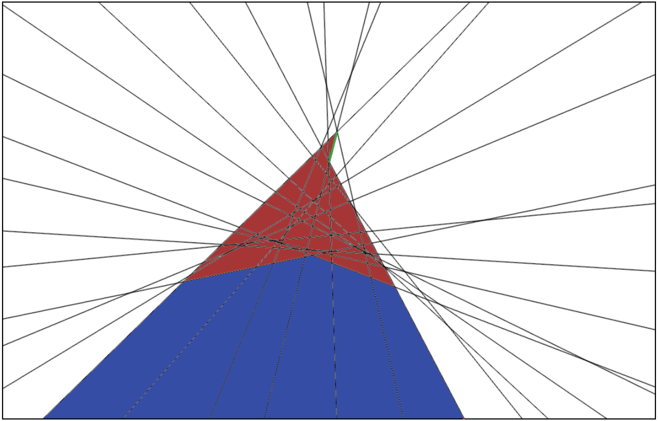
Problem example in $\mathbb{R}^2$



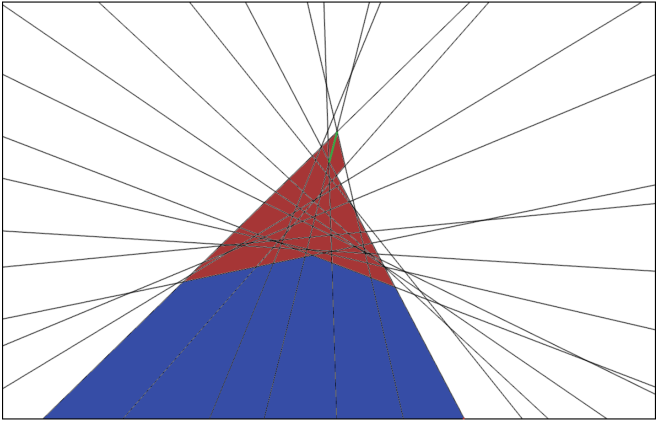
Problem example in $\mathbb{R}^2$



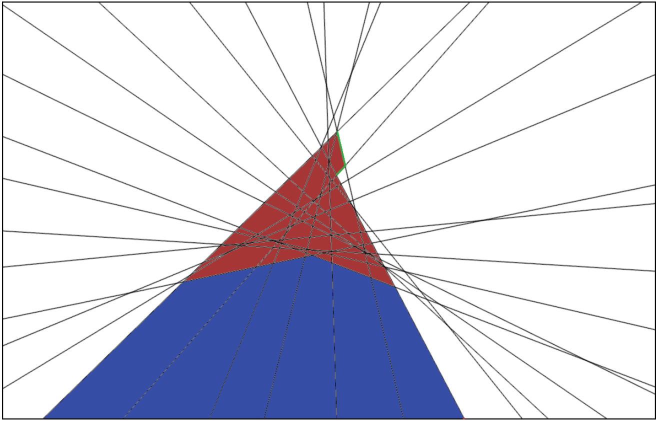
Problem example in $\mathbb{R}^2$



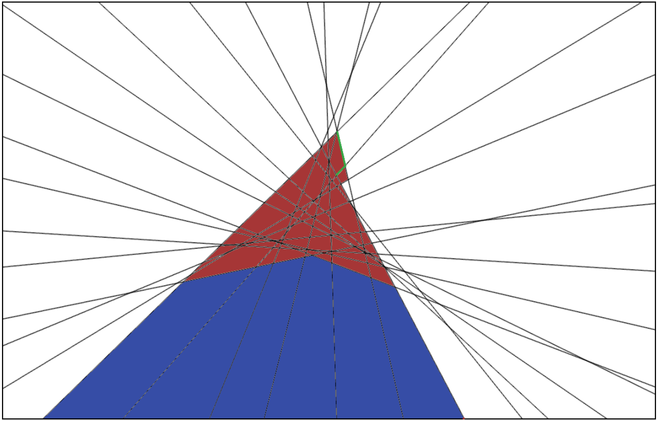
Problem example in $\mathbb{R}^2$



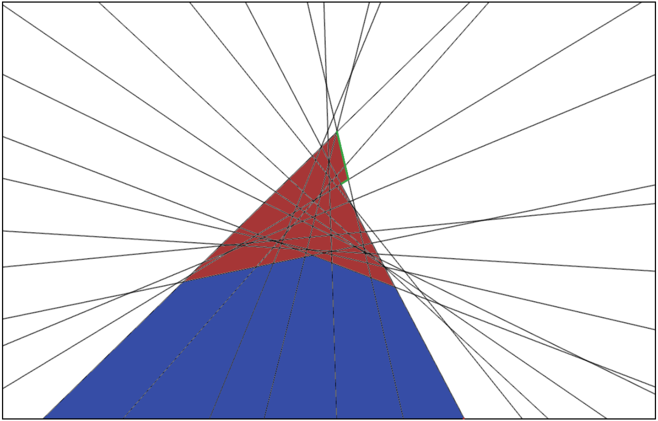
Problem example in $\mathbb{R}^2$



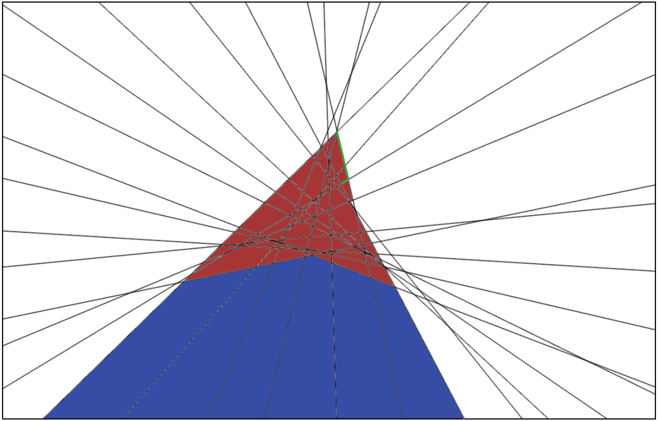
Problem example in $\mathbb{R}^2$



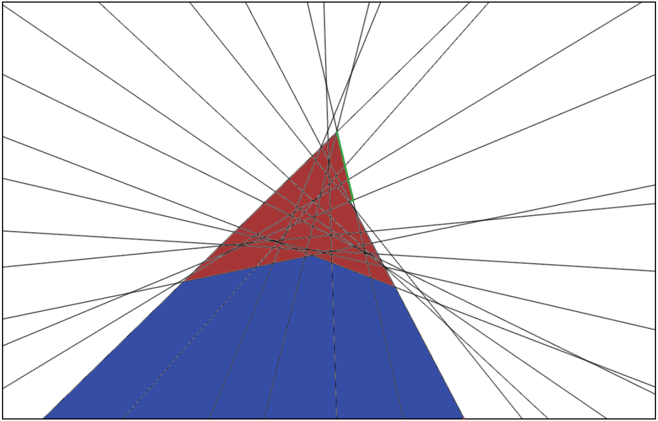
Problem example in $\mathbb{R}^2$



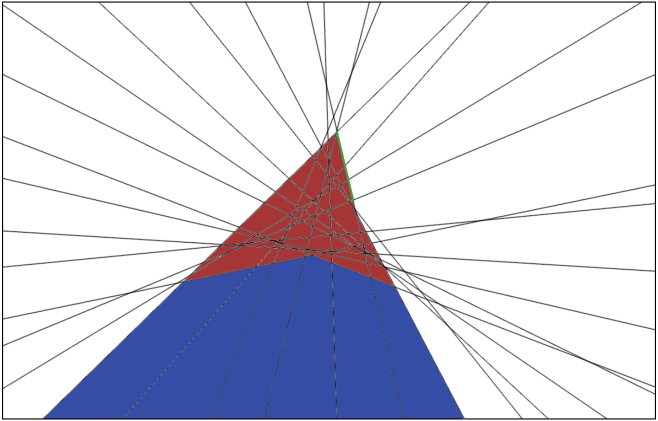
Problem example in $\mathbb{R}^2$



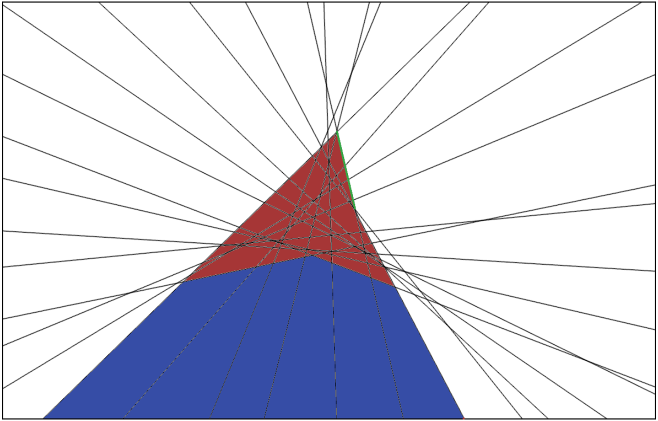
Problem example in $\mathbb{R}^2$



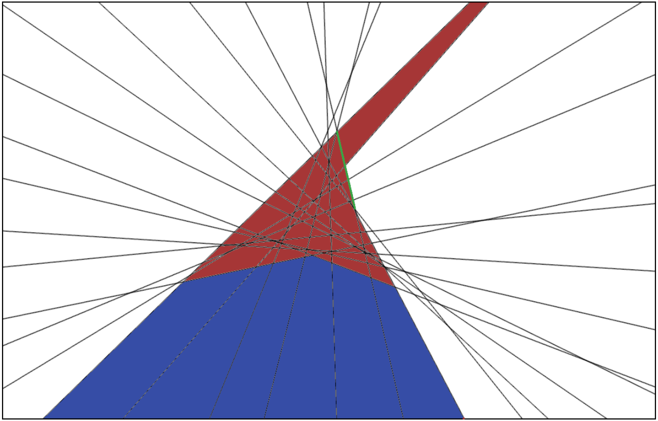
Problem example in $\mathbb{R}^2$



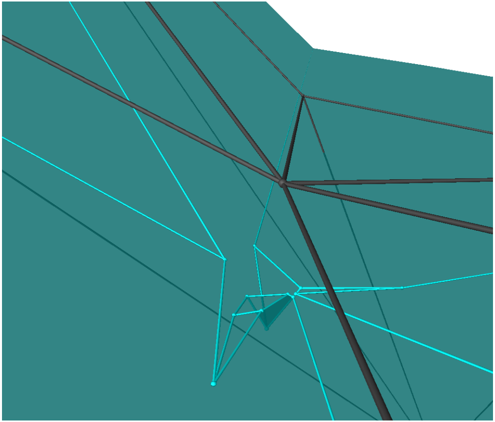
Problem example in $\mathbb{R}^2$



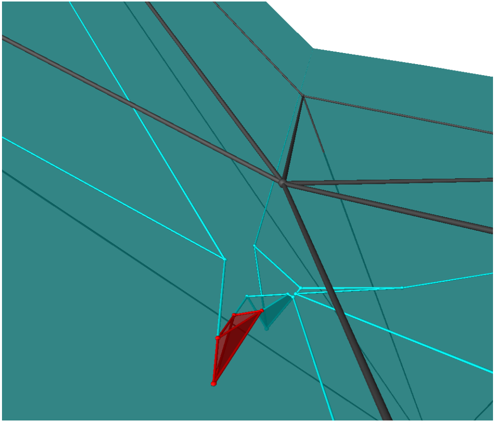
Problem example in $\mathbb{R}^2$



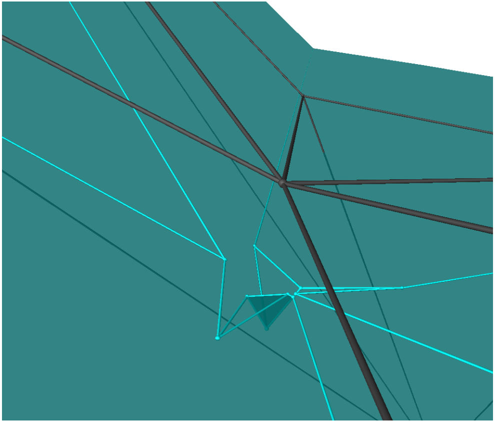
Problem example in $\mathbb{R}^3$



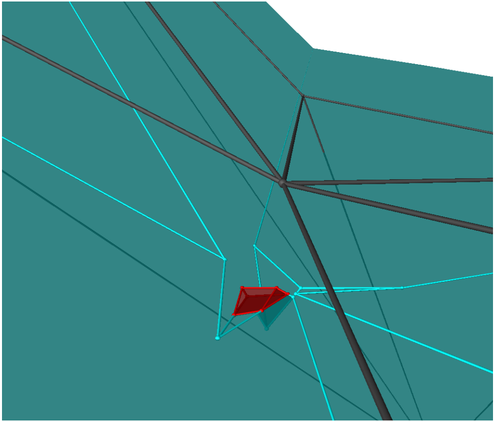
Problem example in $\mathbb{R}^3$



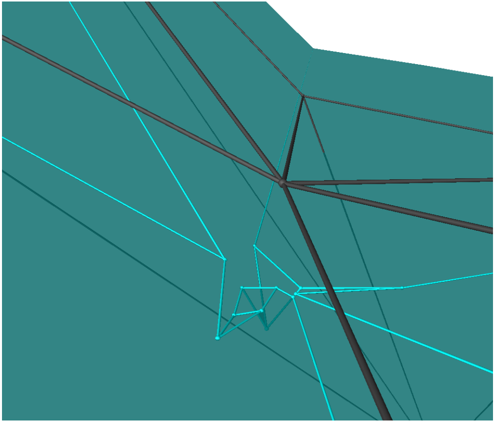
Problem example in $\mathbb{R}^3$



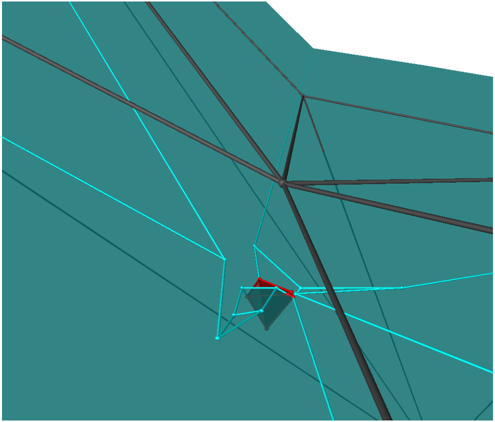
Problem example in $\mathbb{R}^3$



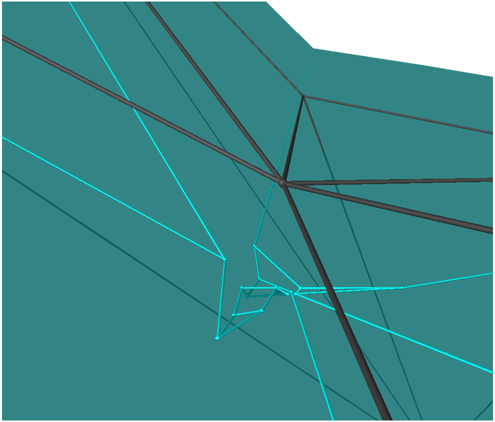
Problem example in $\mathbb{R}^3$



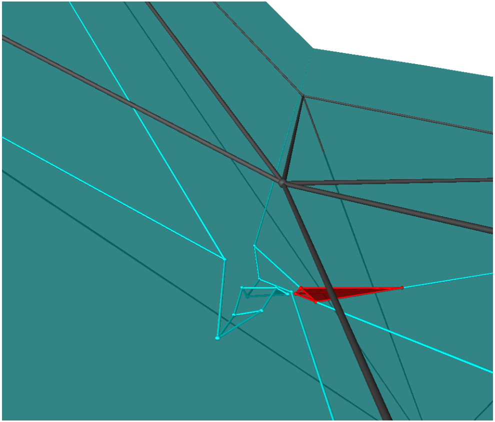
Problem example in $\mathbb{R}^3$



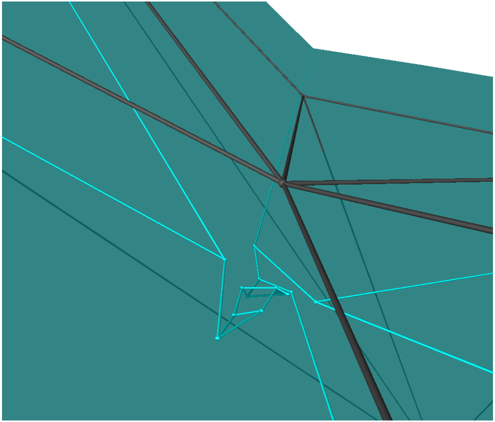
Problem example in $\mathbb{R}^3$



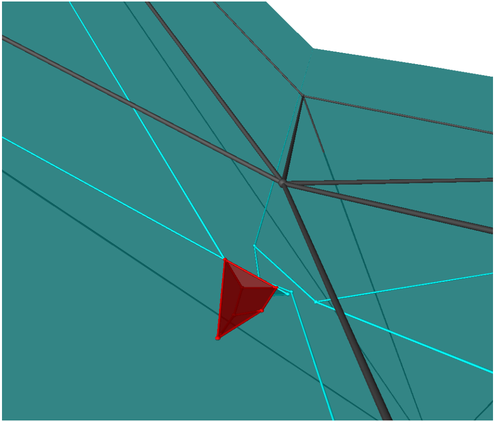
Problem example in $\mathbb{R}^3$



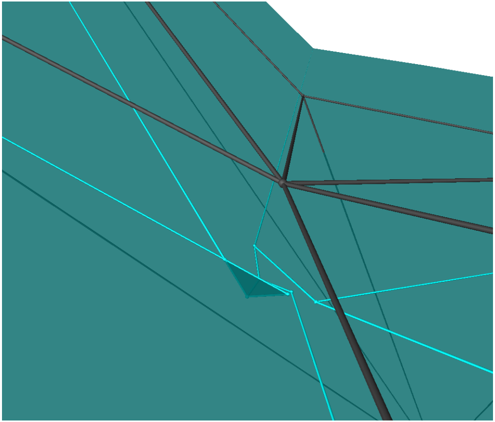
Problem example in $\mathbb{R}^3$



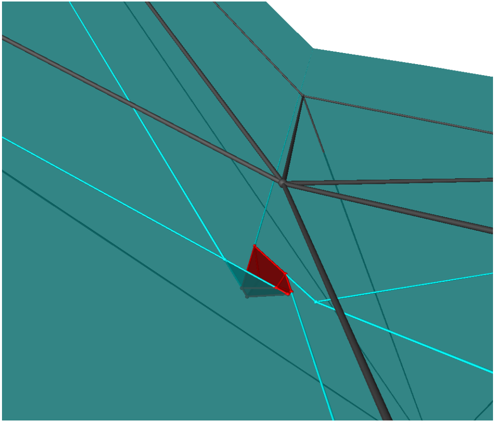
Problem example in $\mathbb{R}^3$



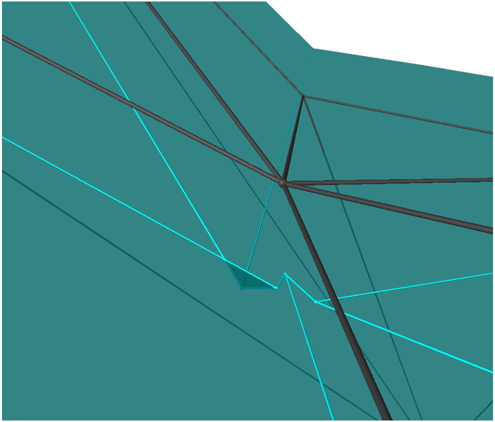
Problem example in $\mathbb{R}^3$



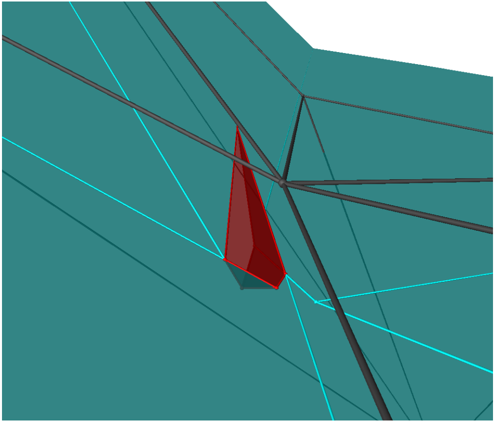
Problem example in $\mathbb{R}^3$



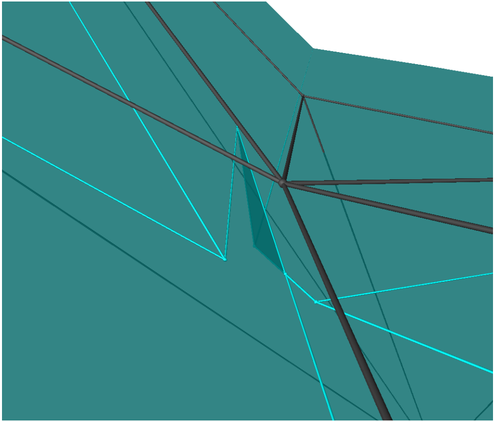
Problem example in $\mathbb{R}^3$



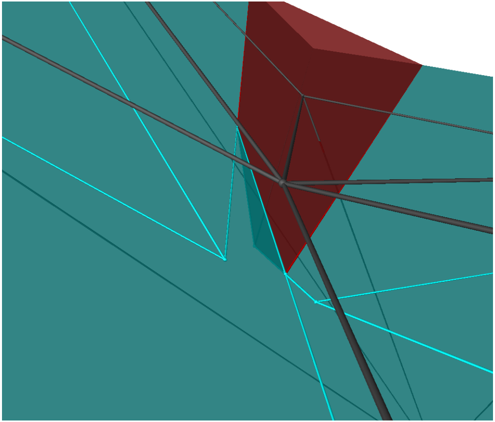
Problem example in $\mathbb{R}^3$



Problem example in $\mathbb{R}^3$



Problem example in $\mathbb{R}^3$



Way out

- Try random order, start again when fail
- Works for all vertices of our various test polytopes
- May take longer time for very high-degree vertices
- Nice byproduct: Many valid surfaces
  - We can choose the one with minimum number of convex/reflex edges
  - Optimize other criteria

Implementation perspective

- Software solution using C++
- All algorithmic steps are implemented
- With the developed software, one can find a valid offset surface for a given event vertex
- Implementation was tested with many polytopes (several 100)
- Implementation is numerically robust

Software runtime: computing $\mathcal{A}(v)$

- In theory, $\Theta(n^3 \log n)$, where n is the number of unique offset planes ($\leq$ facet-degree of v)
- Implementation runs rather fast (on my machine ☺):
  - 10 Planes: < 0.05 s
  - 20 Planes: < 0.5 s
  - 30 Planes: ≈ 1 s
  - 40 Planes: ≈ 2.4 s
  - 50 Planes: ≈ 6 s

(20–50 Planes:) $n > 10$ rare in practice

Software runtime: Single splitting

- Theoretical runtime still needs to be investigated
- Depends on k, number of merged unbounded and bounded cells
- A few examples:
  - `convex_vertex_7.obj`: < 0.05 s
  - `kiev_7.obj`: < 0.05 s
  - `saddle_10.obj`: < 0.2 s

Software runtime: Initial splittings

- Algorithm is applied for each eligible vertex on the polytope
- A few examples:
  - `journal_verworrtakelt.obj` (66 split vertices): $\approx$ 0.8 s
  - `journal_lion.obj` (85 split vertices): $\approx$ 1.2 s
  - `journal_venus.obj` (142 split vertices): $\approx$ 1.8 s
  - `journal_bunny.obj` (144 split vertices): $\approx$ 2.2 s

Future work

- Glue the offset surfaces for the vertices to the polytope
- Speed up event detection later during the shrinking process
  - Initially, the event detection is just checking the degree of vertices
- Find a method of generating a provably sucessful cell adding order