Lesson dtvd3d: some extras

GEO1004:
3D modelling of the built environment

https://3d.bk.tudelft.nl/courses/geo1004

3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

While I have your attention

1. **Intro to Q5** wednesday 6 May @ 11:00—12:30 (room P)
2. **Geomatics AI taskforce** friday 8 May @ 15:30—17:00
    1. Who wants to join? Max 3-4 students ==> DM me on discord
3. **Evangelia Palli** is the new Geomatics SA

Incremental construction

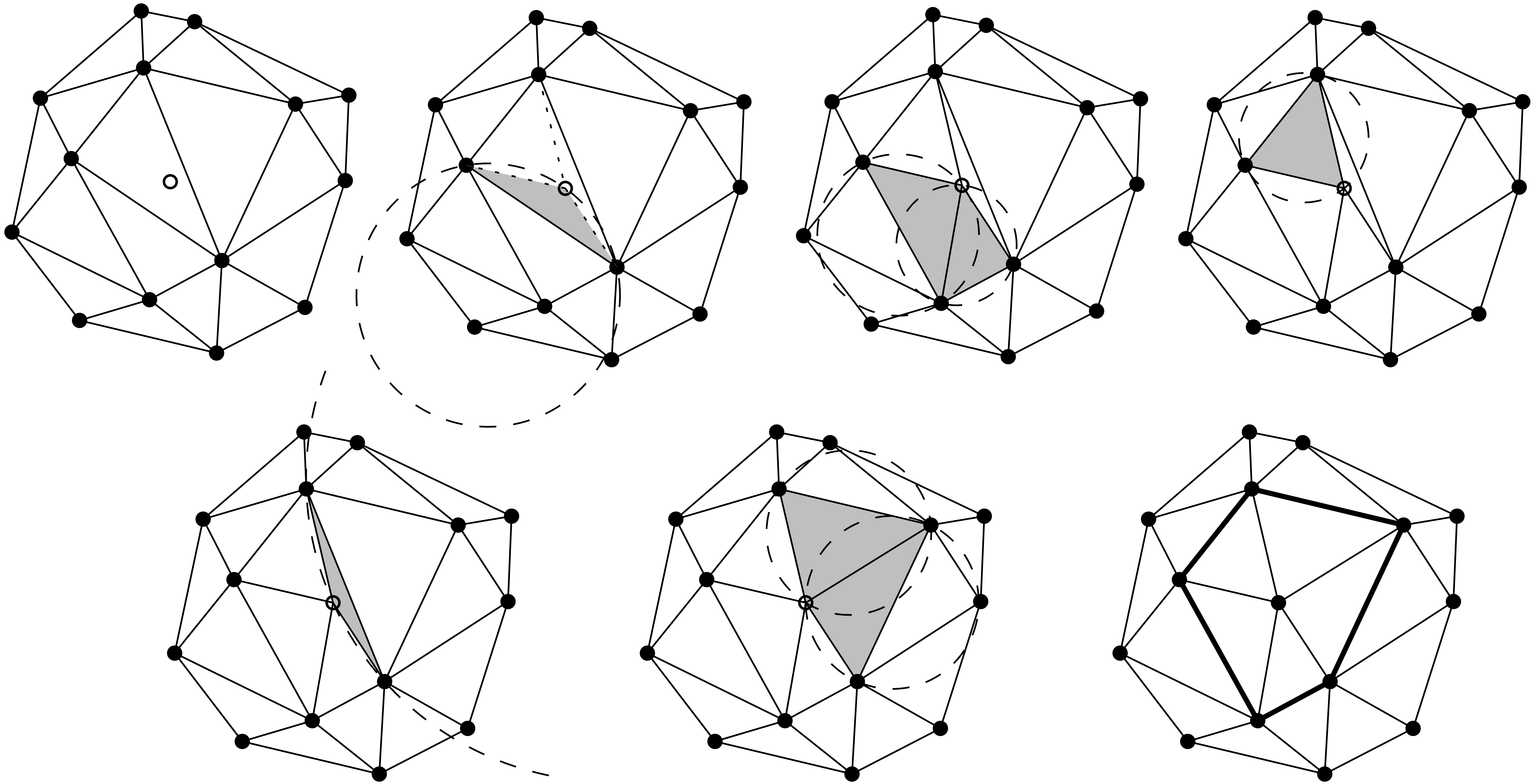


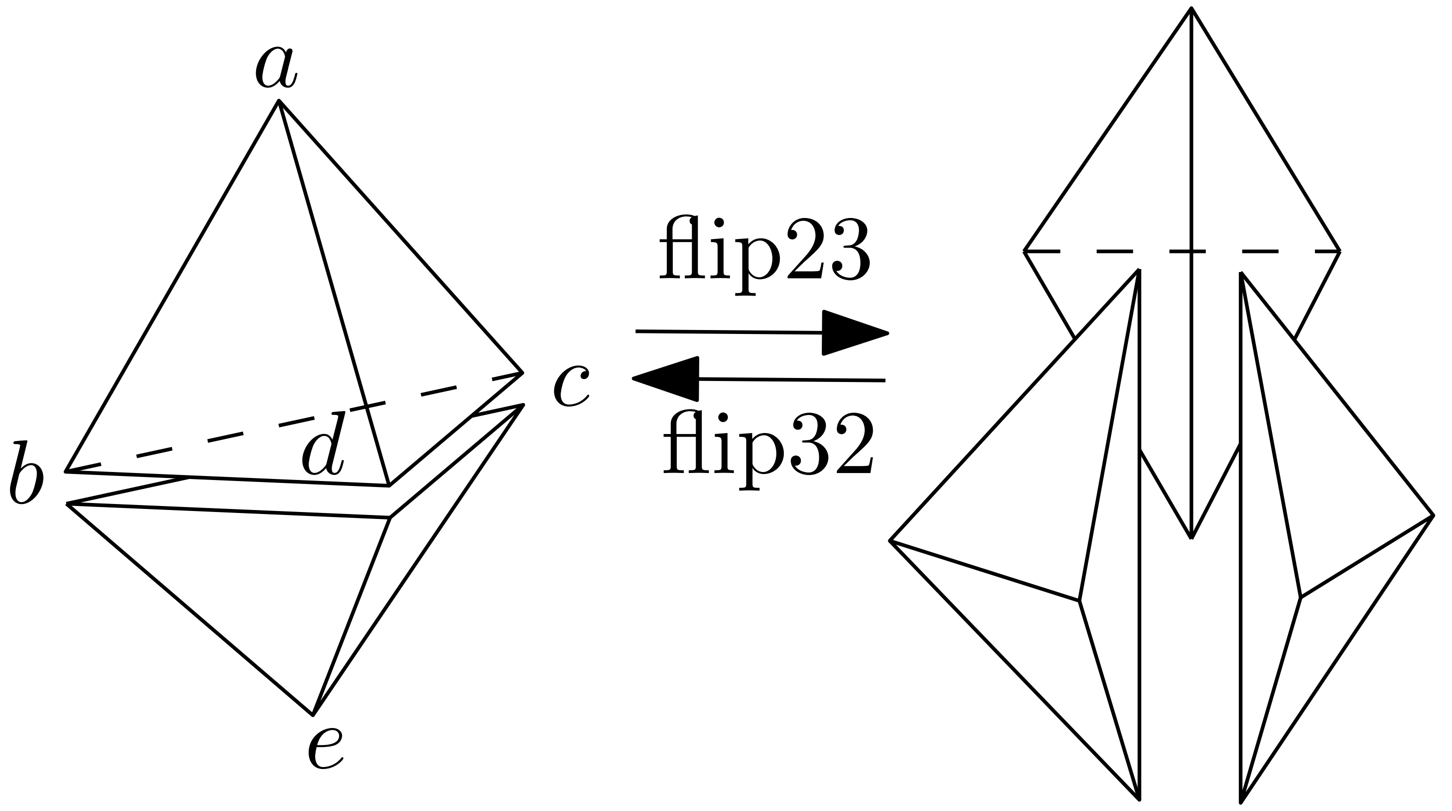
Figure 7: Step-by-step insertion, with flips, of a single point in a DT in two dimensions.

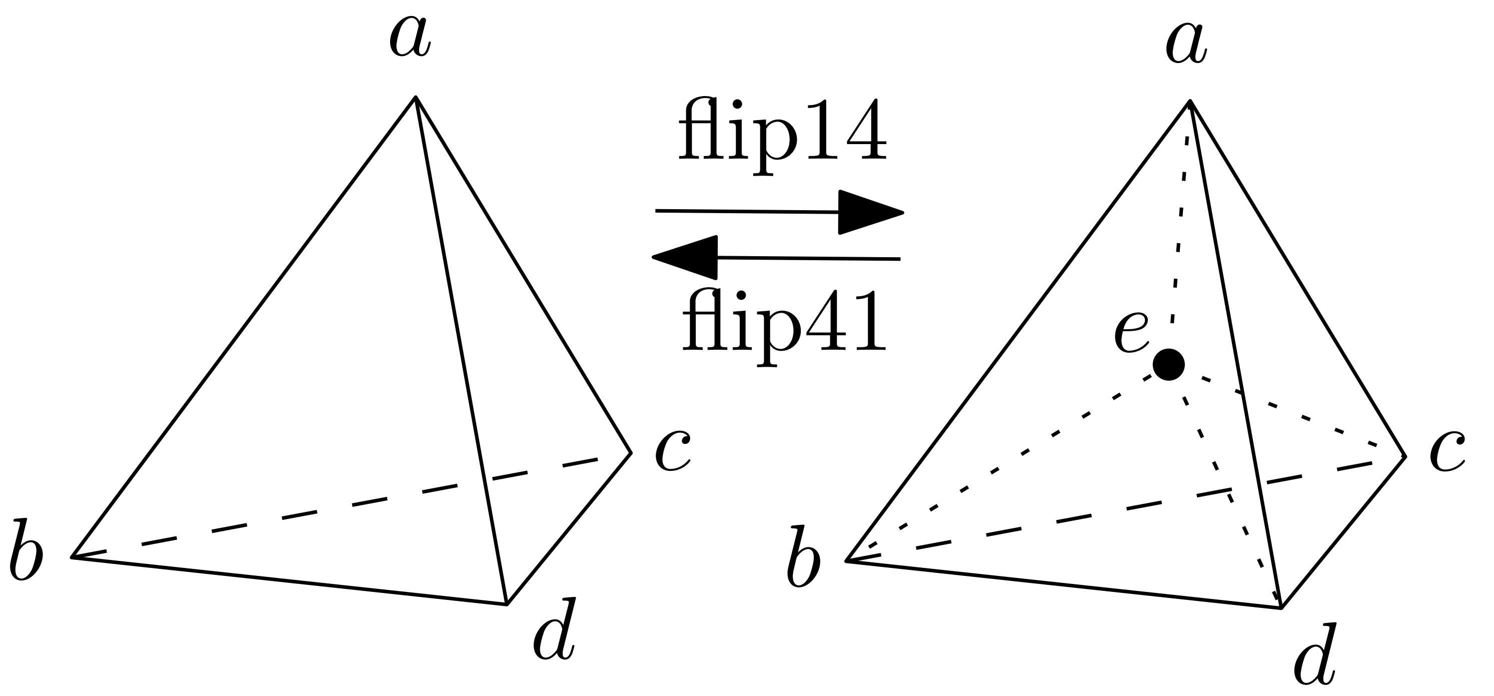
Incrementation construction

Algorithm 1: Algorithm to insert one point in a DT

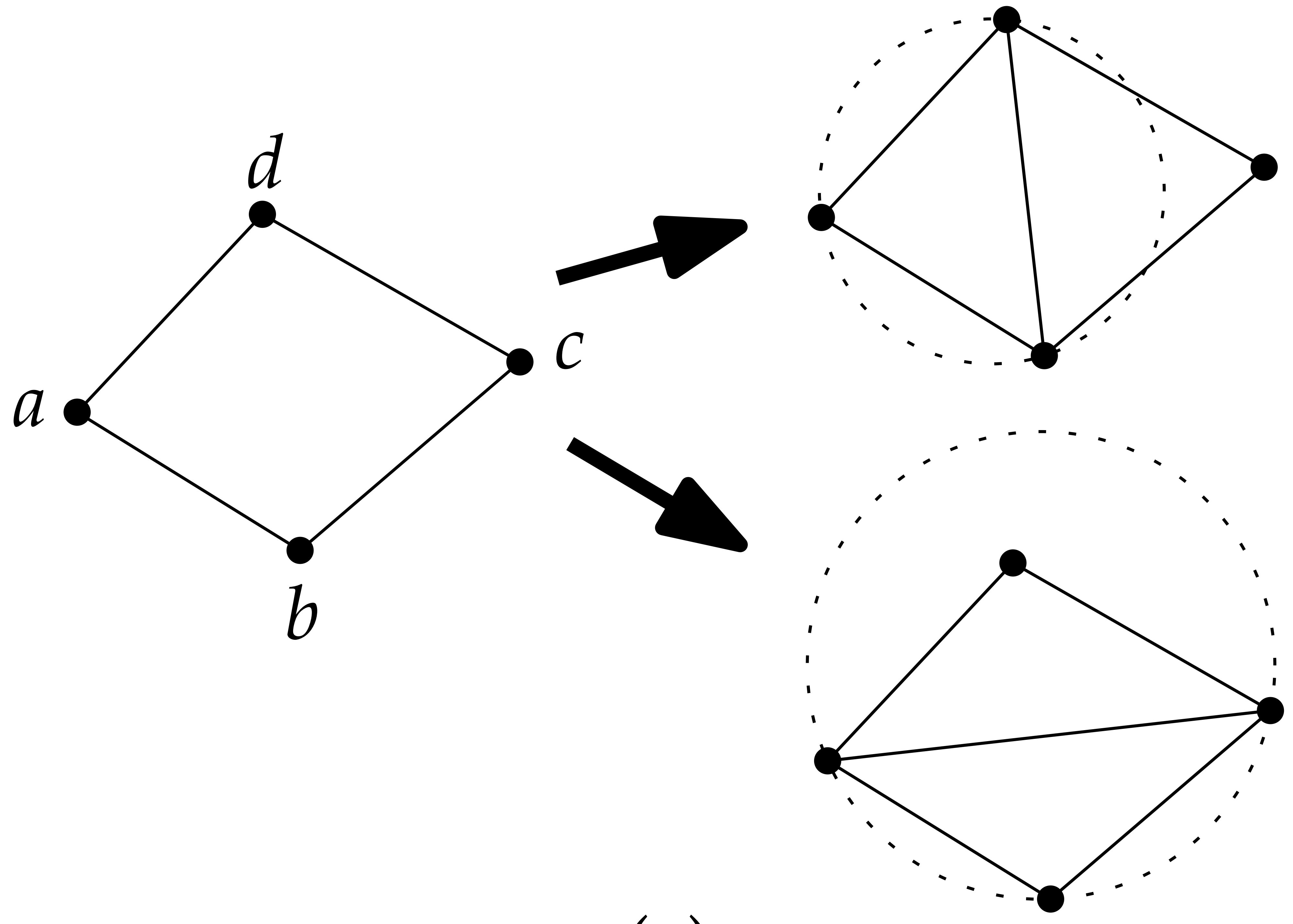
Input: A DT(S) $\mathcal{T}$ in $\mathbb{R}^3$, and a new point p to insert
Output: $\mathcal{T}^p = \mathcal{T} \cup \{p\}$

find tetrahedron τ containing p
insert p in τ by splitting it in to 4 new tetrahedra (flip14)
push 4 new tetrahedra on a stack
while *stack is non-empty* **do**
    $\tau = \{p, a, b, c\} \leftarrow$ pop from stack
    $\tau_a = \{a, b, c, d\} \leftarrow$ get adjacent tetrahedron of τ having the edge *abc* as a face
    if *d is inside circumsphere of τ* **then**
        if *configuration of τ and τ_a allows it* **then**
            flip the tetrahedra τ and τ_a (flip23 or flip32)
            push 2 or 3 new tetrahedra on stack
        else
            Do nothing

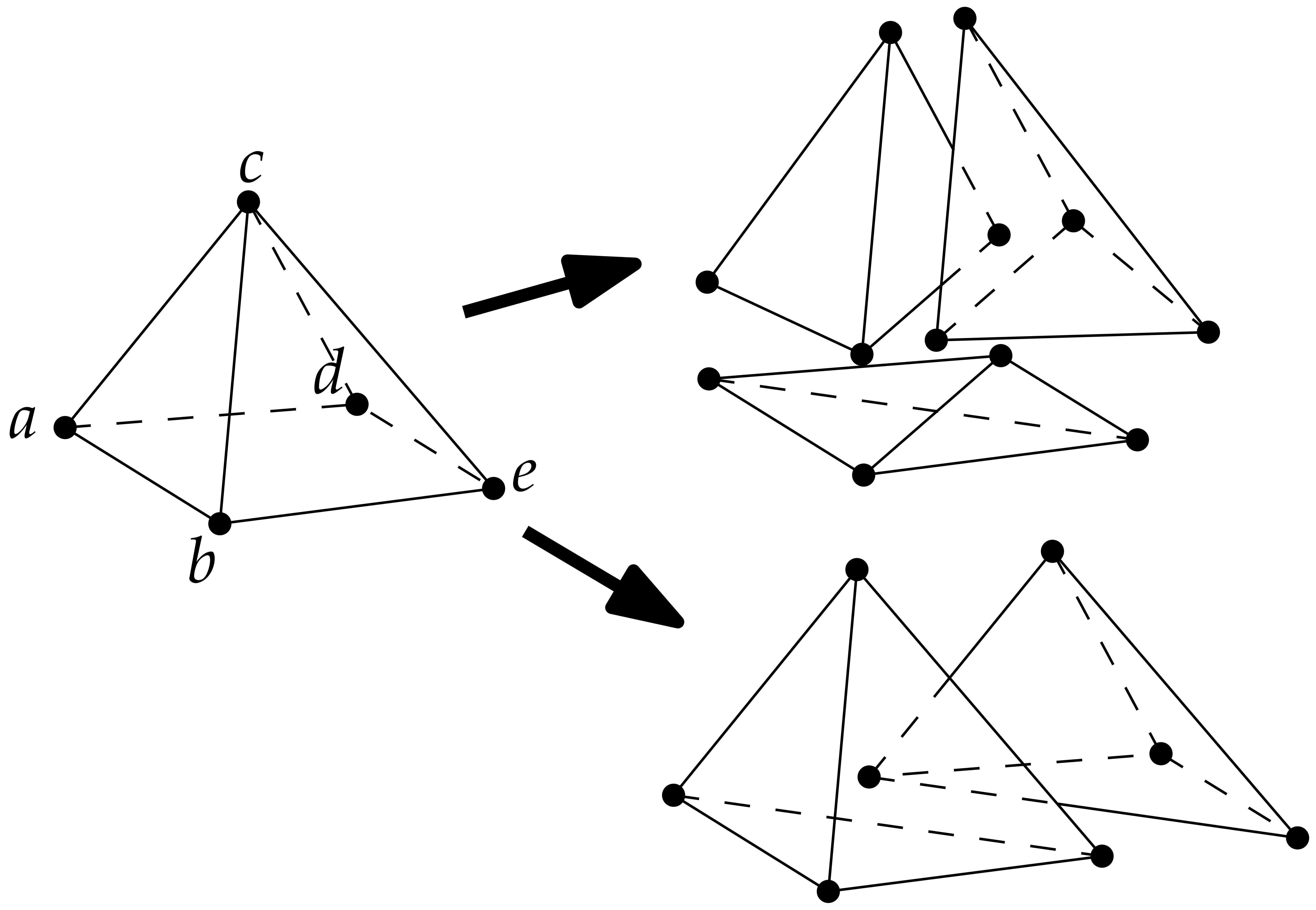




Slivers in DT



(a)



(b)

3D DT == 4D convex hull

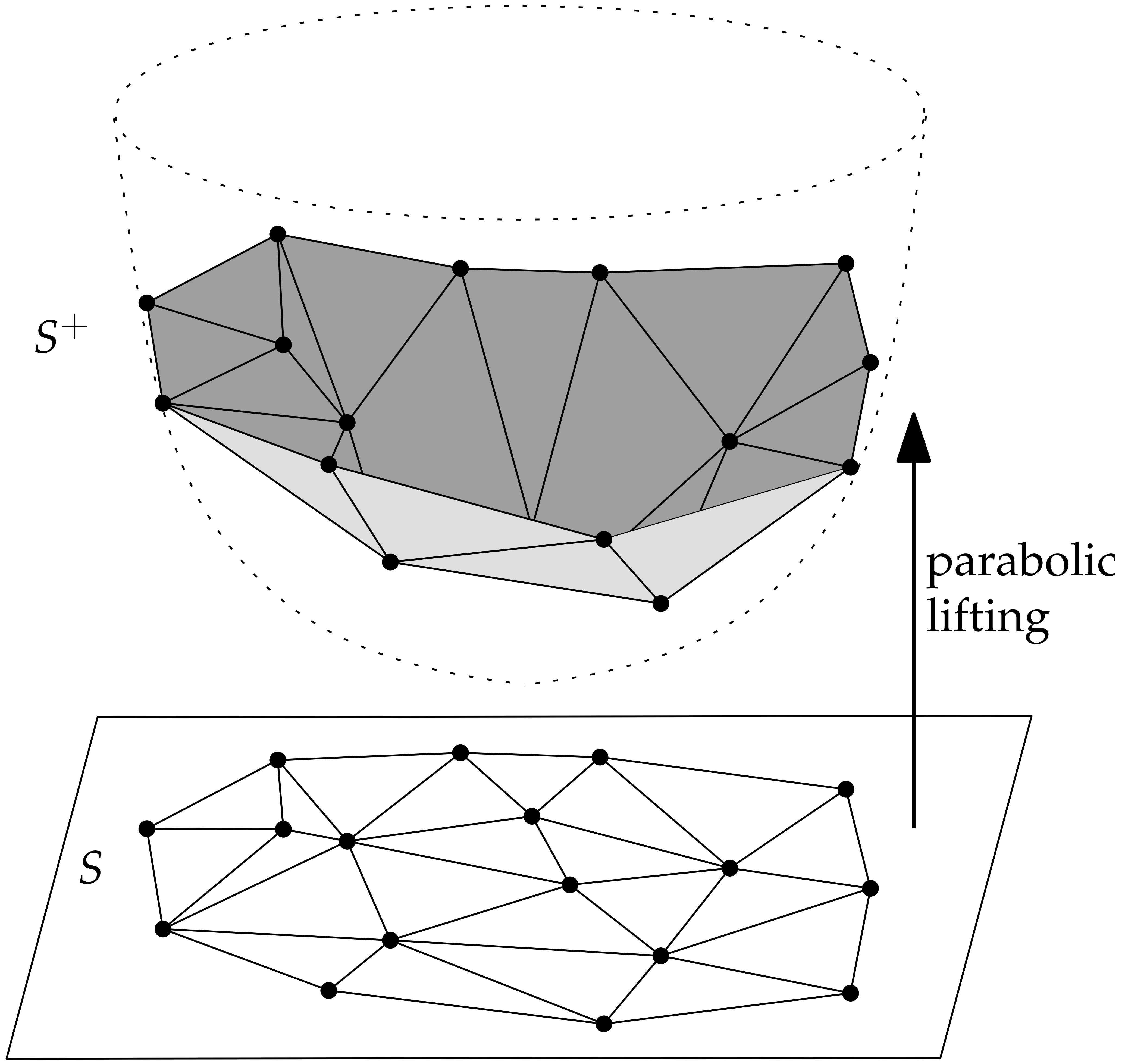


Figure 6: The parabolic lifting map for a set S of points $\mathbb{R}^2$.

3D == no “fixed” orientation

3.3.2 Predicates

The ‘orientation’ of points in three dimensions is somewhat tricky because, unlike in two dimensions, we can not simply rely on the counter-clockwise orientation. In three dimensions, the orientation is always relative to another point of reference, ie given three points we cannot say if a fourth one is left of right, this depends on the orientation of the three points.

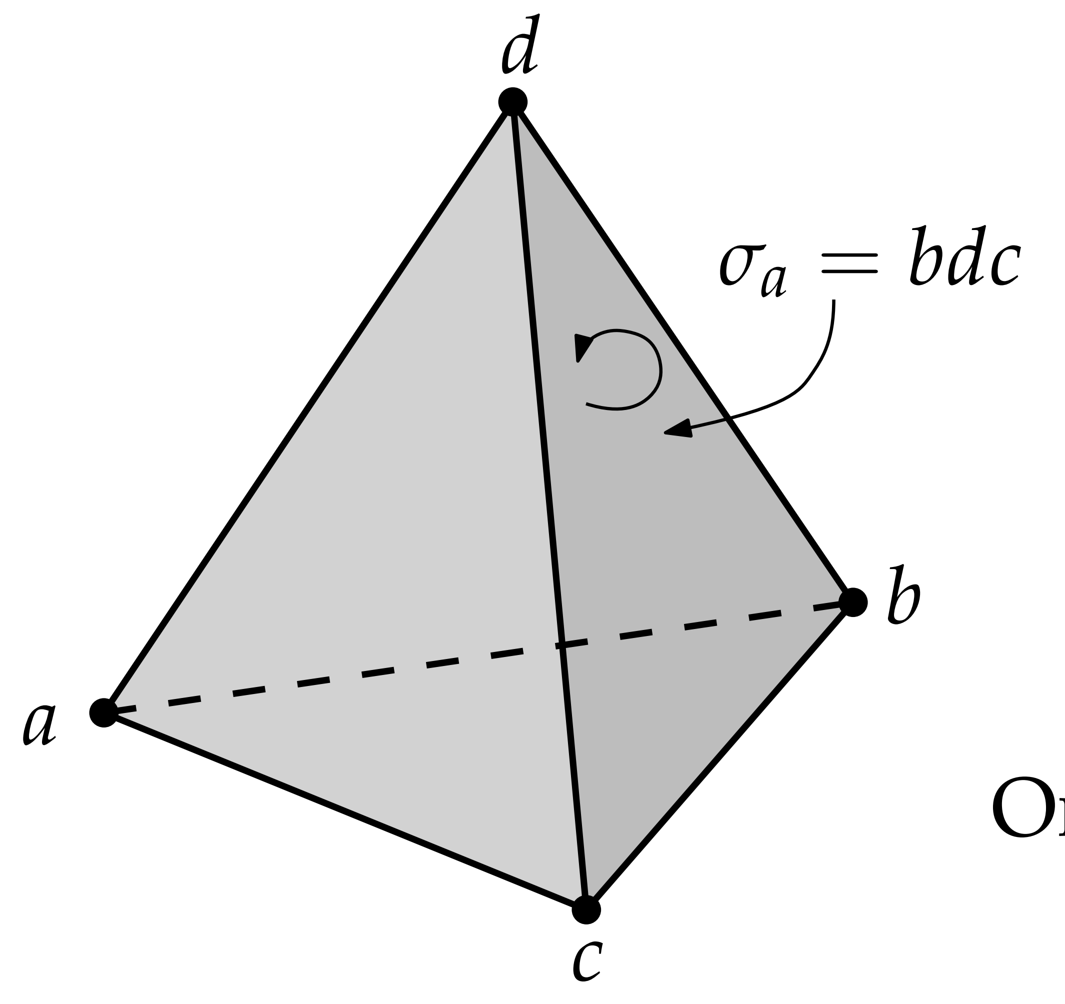
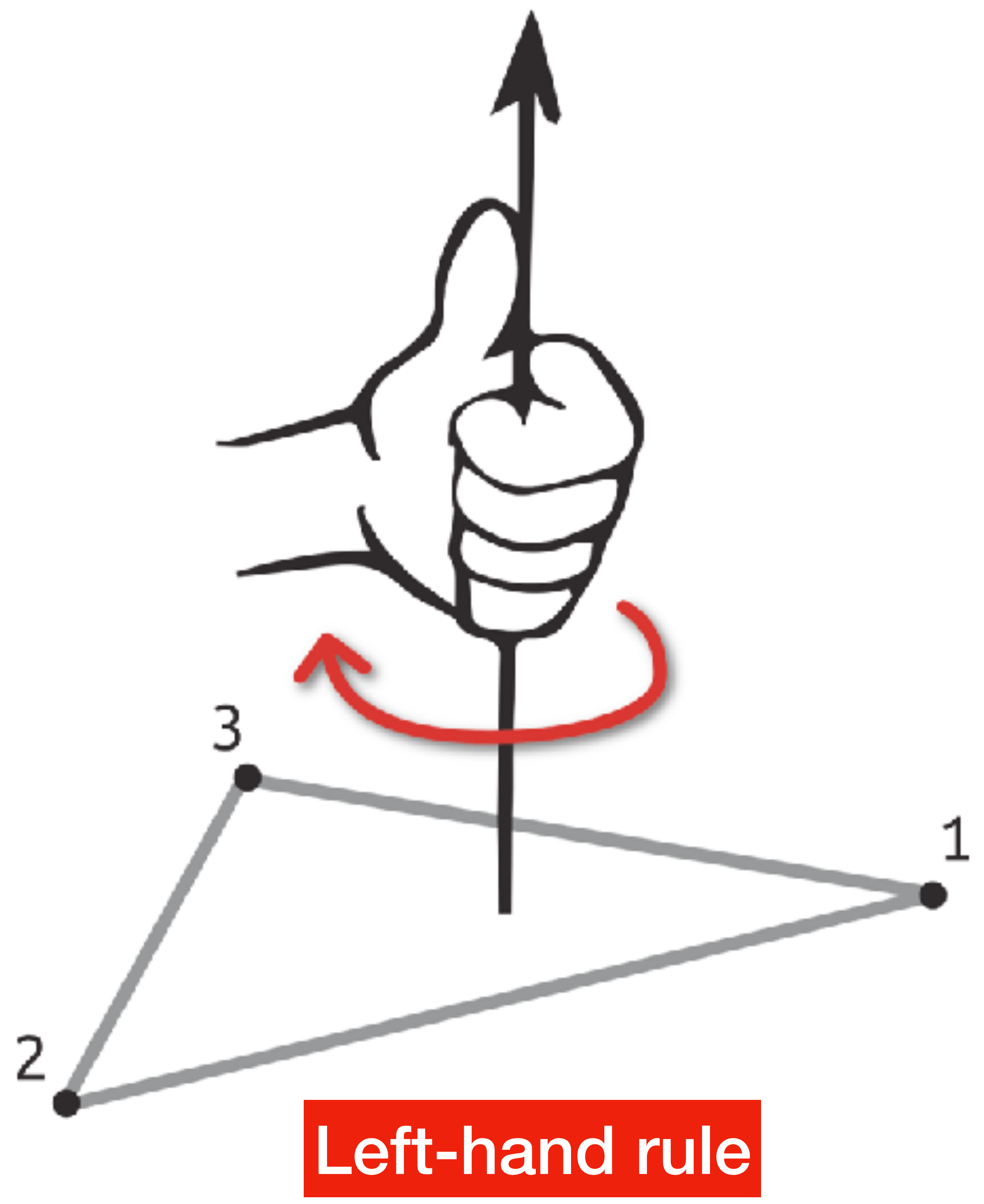


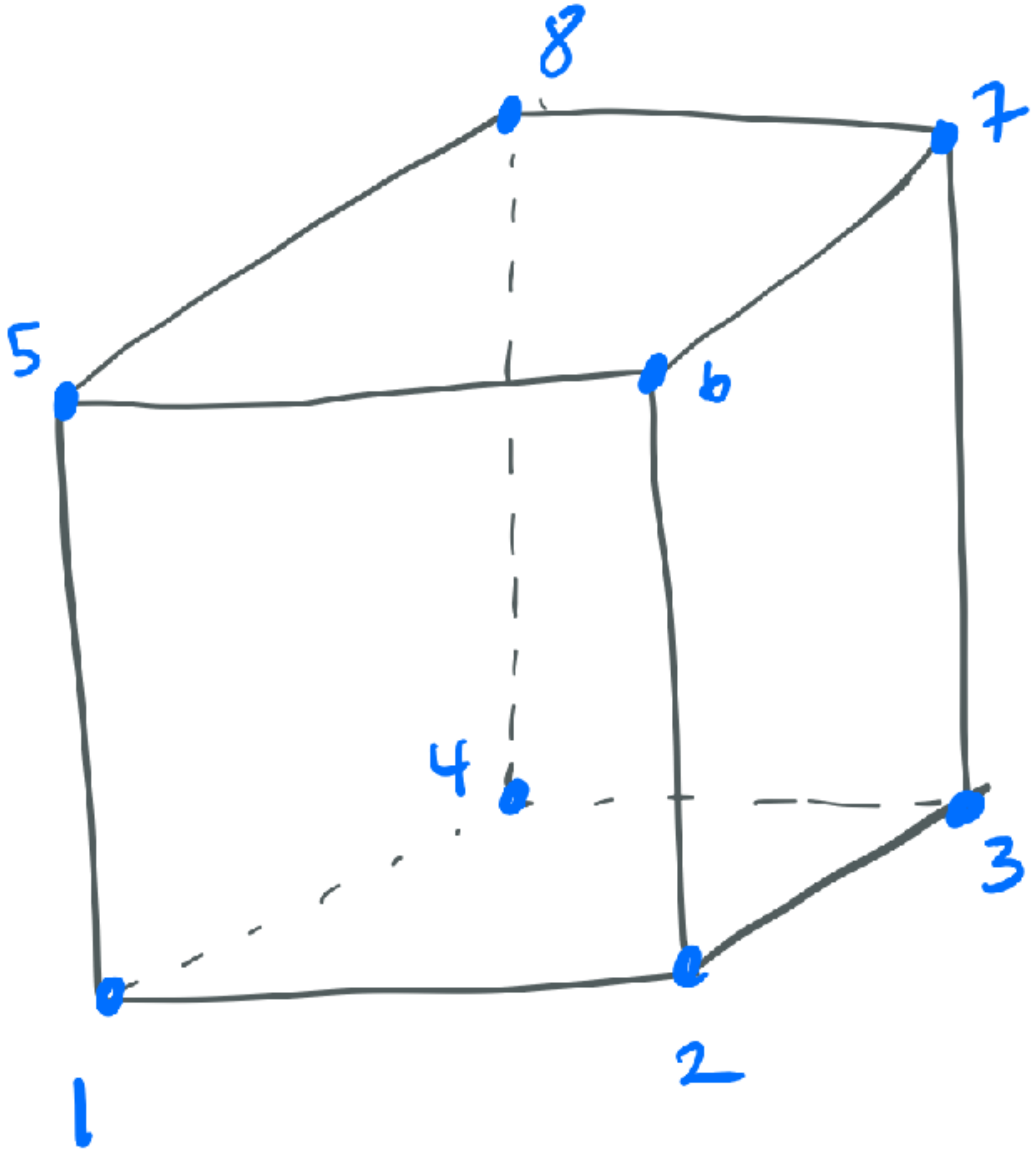
Figure 3.13: The tetrahedron $abcd$ is correctly oriented since ORIENT (a, b, c, d) returns a positive result. The arrow indicates the correct orientation for the face σ_a, so that ORIENT (σ_a, a) returns a positive result.



ORIENT can be implemented as the determinant of a matrix:

$$\text{ORIENT}(a, b, c, p) = \begin{vmatrix} a_x & a_y & a_z & 1 \\ b_x & b_y & b_z & 1 \\ c_x & c_y & c_z & 1 \\ p_x & p_y & p_z & 1 \end{vmatrix} \tag{3.2}$$

Build the OBJ of that cube



cube.obj

```
v 0 0 0
v 1 0 0
v 1 1 0
v 0 1 0
v 0 0 1
v 1 0 1
v 1 1 1
v 0 1 1

f ....
```

?

Duality VD <-> DT in 3D

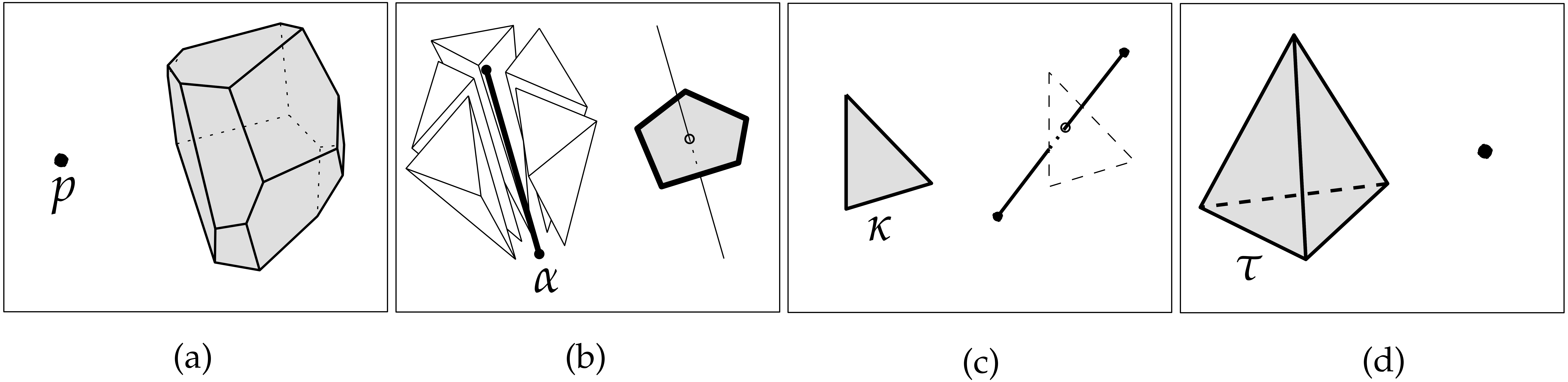


Figure 3: Duality in $\mathbb{R}^3$ between the elements of the VD and the DT.

Exercise

You have 1000 points in 3D space, and you use CGAL (or others) to construct the Delaunay tetrahedralisation.

If you want to draw one Voronoi cell of a given point, how would you proceed?

The same functions as startinpy as present (incidence + adjacency) and point location and etc.

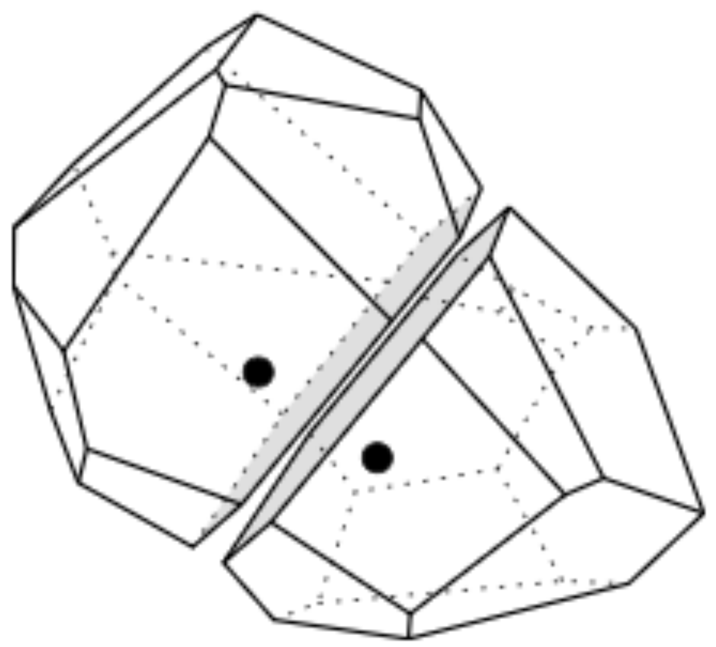


Figure 3.3: Two Voronoi cells adjacent to each other in $\mathbb{R}^3$, they share the grey face.

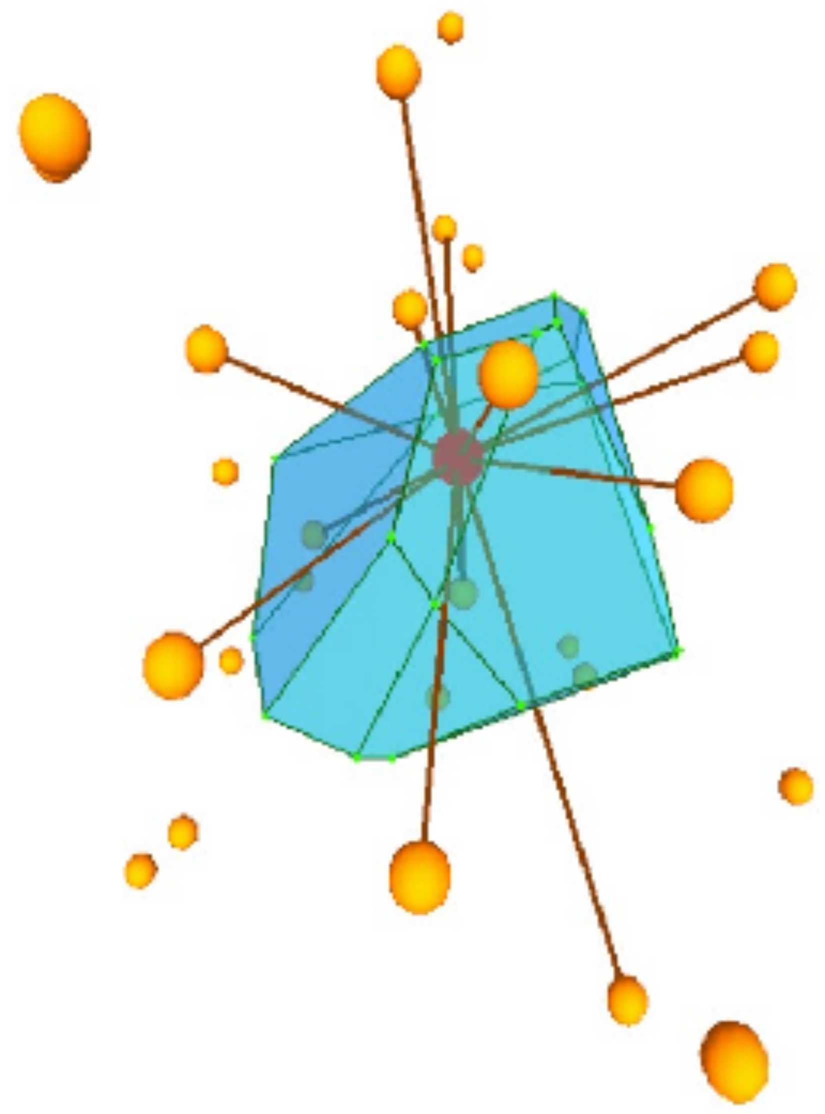
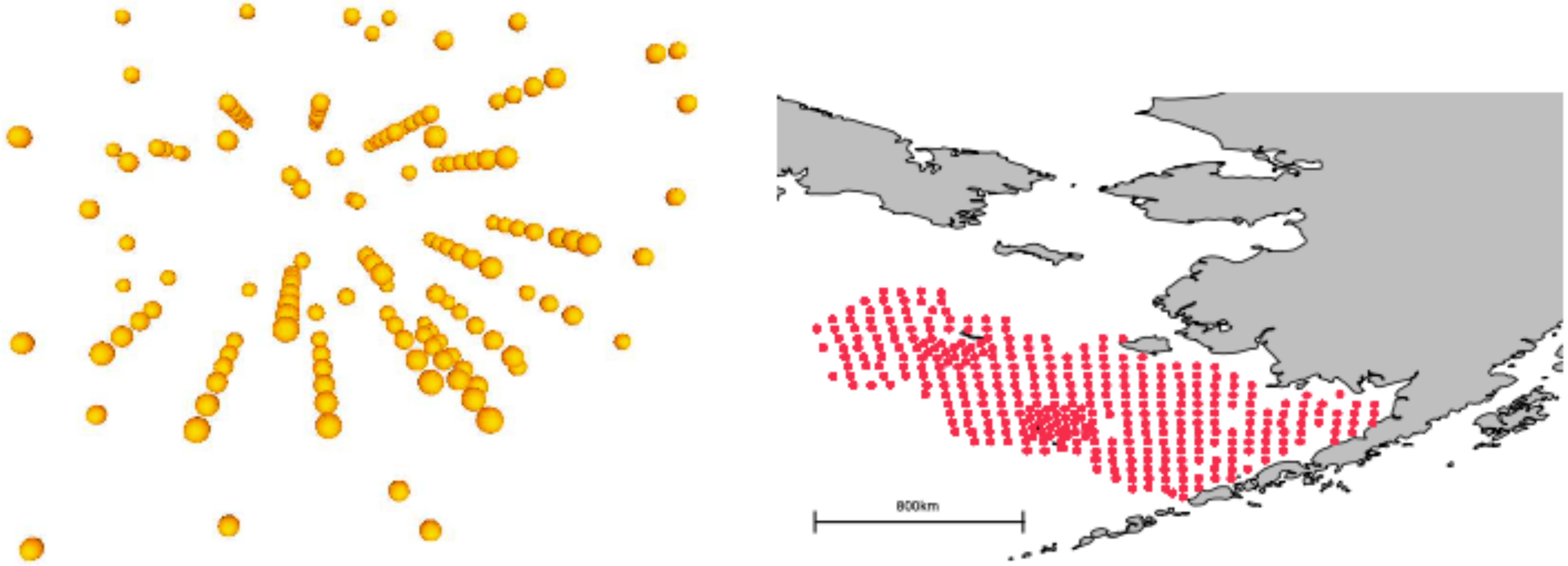


Figure 3.4: The Voronoi cell for the red vertex, the red edges are the Delaunay edges that are dual to the Voronoi facets.

App#1: 3DVD as alternative to voxels



(a) (b)

Figure 10: **(a)** Example of a dataset in geology, where samples were collected by drilling a hole in the ground. Each sample has a location in 3D space ($x - y - z$ coordinates) and one or more attributes attached to it. **(b)** An oceanographic dataset in the Bering Sea in which samples are distributed along water columns. Each red point represents a (vertical) water column, where samples are collected every 2m, but water columns are about 35km from each other.

App#2: spatial interpolation

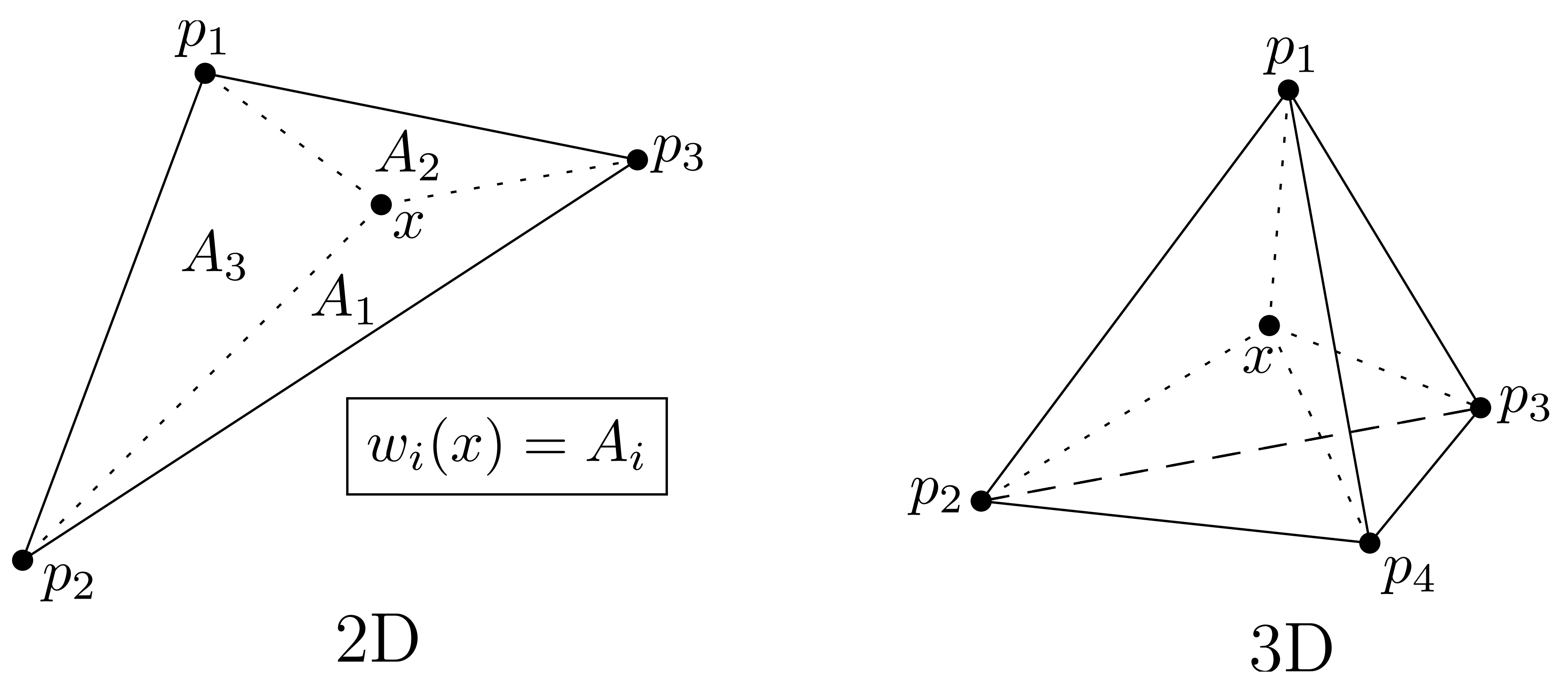
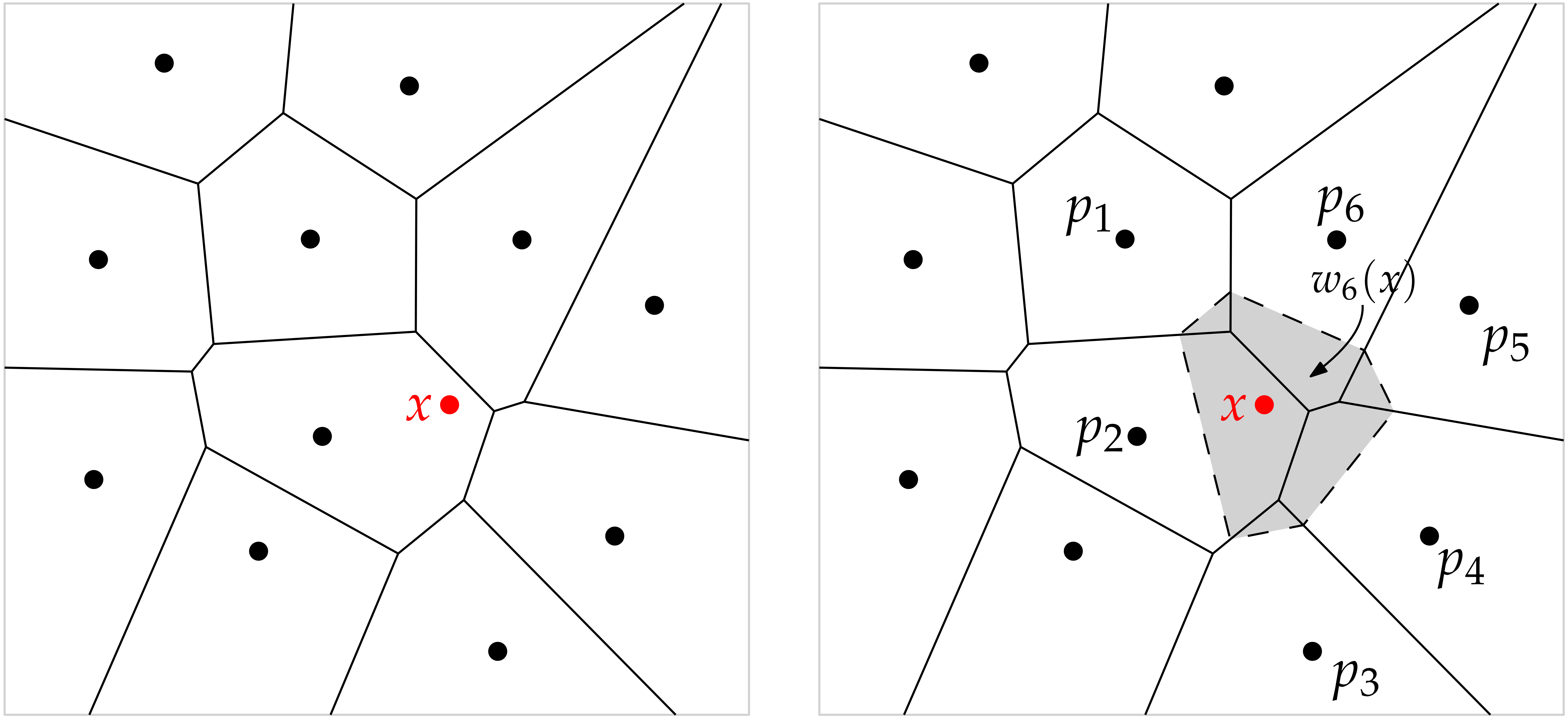


Figure 11: Barycentric coordinates in two and three dimensions. A_i represents the area of the triangle formed by x and one edge.

$$vol(\sigma) = \frac{1}{d\,!}\left| det \begin{pmatrix} v^0 & \dots & v^d \\ 1 & \dots & 1 \end{pmatrix} \right|$$

App#2: spatial interpolation



$$\boxed{w_i(x) = stolenarea}$$

App#3: visualisation with iso-surfaces

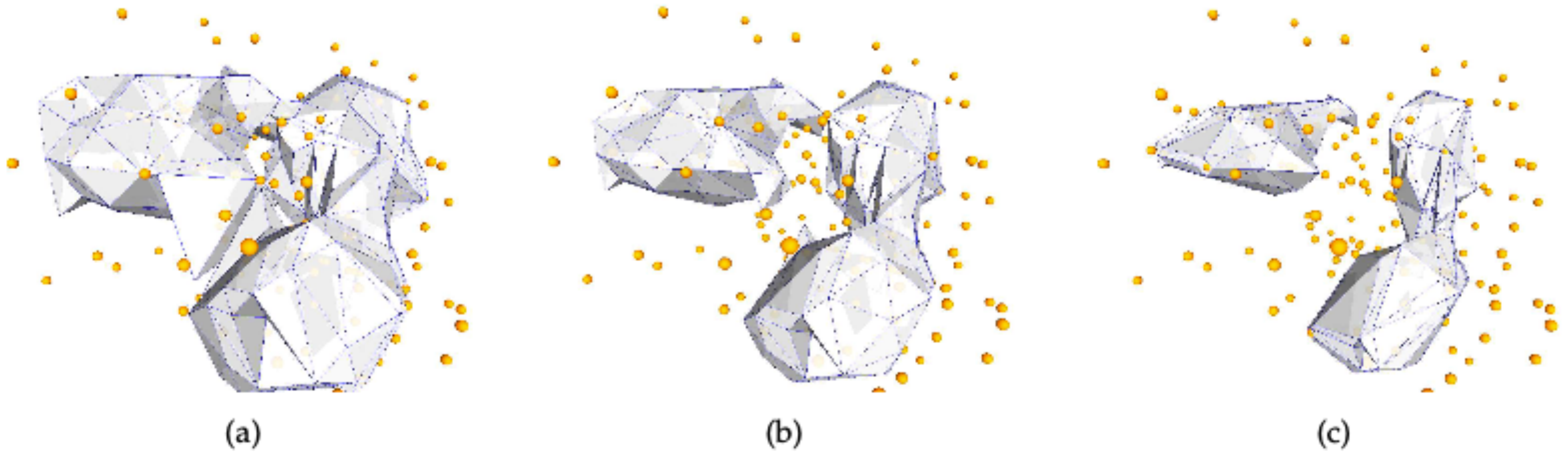
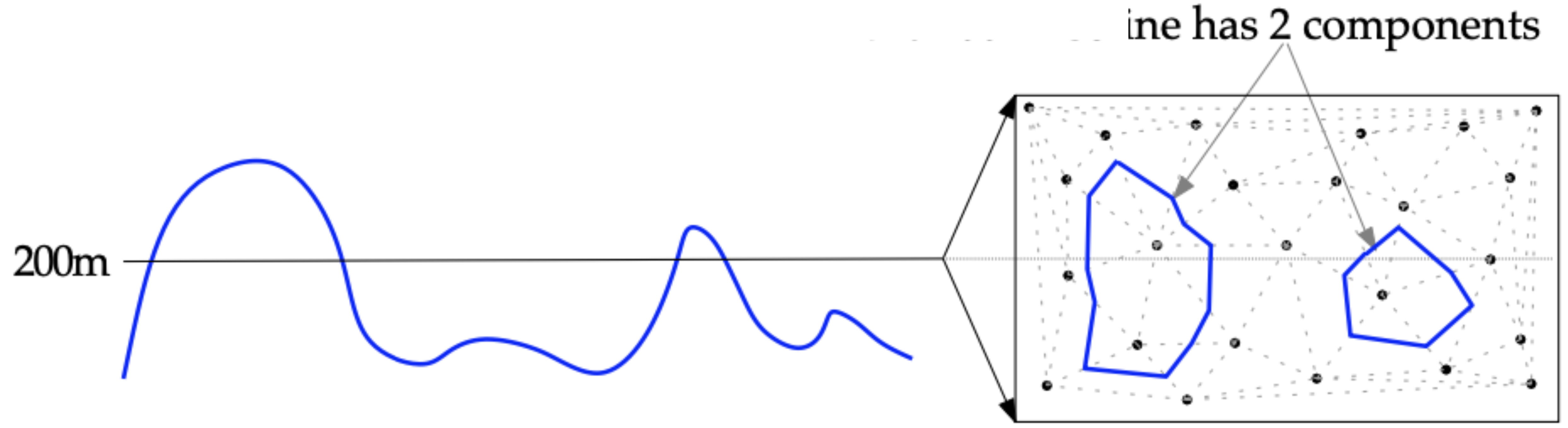


Figure 12: An example of an oceanographic dataset where each point has the temperature of the water, and three isosurface extracted (for a value of respectively 2.0, 2.5 and 3.5) from this dataset.



App#3: visualisation with iso-surfaces

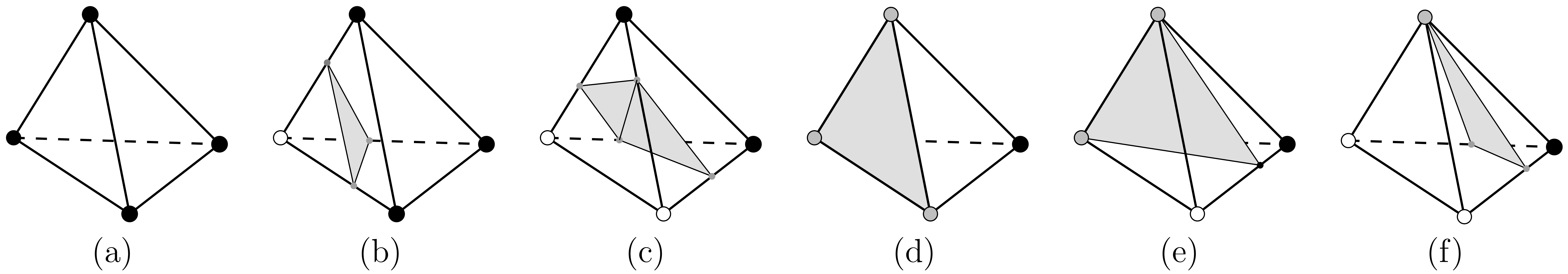
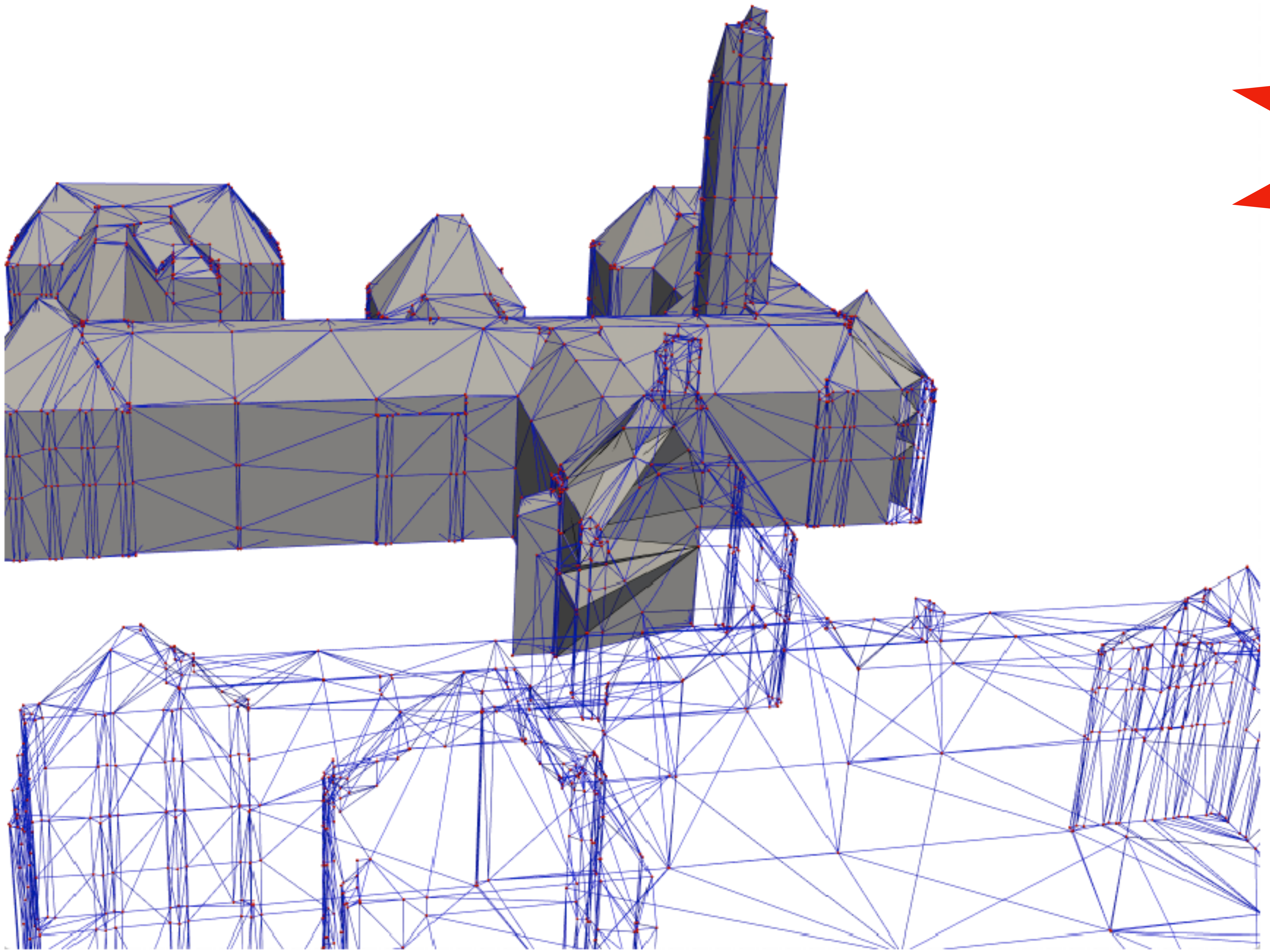


Figure 3.15: Potential isosurface (for an attribute value v) extracted for one tetrahedron. Black vertex means that the attribute of this vertex is below v; white vertex means it is above; and grey that it is equal.

Triangulating a building (or any 3D model)



Two very different cases

demo with 3DBAG buildings

- MeshLab: https://www.meshlab.net
- TetGen: http://tetgen.org
- ParaView: https://www.paraview.org
- Mapple: https://github.com/LiangliangNan/Easy3D

https://3d.bk.tudelft.nl/courses/geo1004

3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology