Marching Triangles

Misha Kazhdan

Outline

- Review
- Goal
- Interpolation
- Implementation
- Inductive triangulation

Notation

Throughout:

We represent scalars by italics/Greek letters:

$$a, \alpha \in \mathbb{R}$$

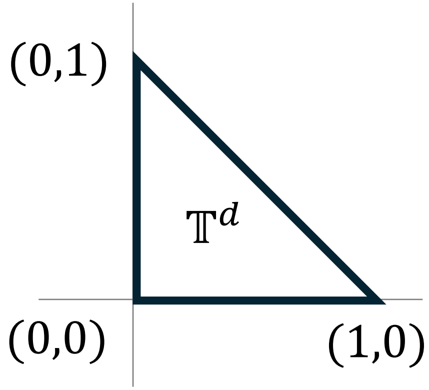
We represent vectors/matrices with bold letters:

$$\mathbf{v} \in \mathbb{R}^d \quad \text{and} \quad \mathbf{M} \in \mathbb{R}^{d \times d}$$

We represent vector/matrix coefficients with italics letters:

$$(\mathbf{v})_i \equiv v_i \in \mathbb{R} \quad \text{and} \quad (\mathbf{M})_{ij} \equiv M_{ij} \in \mathbb{R}$$

Review: simplices in $\mathbb{R}^d$



Definition:

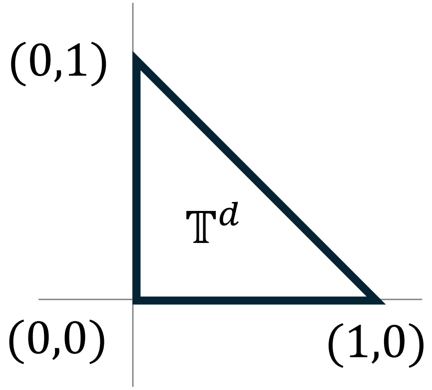
The d-dimensional *unit-right-simplex* $\mathbb{T}^d \subset \mathbb{R}^d$ is the simplex through:

$$\{(0,0,\ldots,0,0),(1,0,\ldots,0,0),(0,1,\ldots,0,0),\ldots,(0,0,\ldots,1,0),(0,0,\ldots,0,1)\}$$

$$\Updownarrow$$

$$\mathbb{T}^d = \left\{\mathbf{x} = \{x_1,\ldots,x_d\} \in \mathbb{R}^d \middle| 0 \le x_i \le 1,\ \textstyle\sum_{i=1}^d x_i \le 1\right\}$$

Review: simplices in $\mathbb{R}^d$



Definition:

The d-dimensional *unit-right-simplex* $\mathbb{T}^d \subset \mathbb{R}^d$ is the simplex:

$$\mathbb{T}^d = \left\{ \mathbf{x} = \{x_1, \dots, x_d\} \in \mathbb{R}^d \middle| 0 \le x_i \le 1,\ \sum_{i=1}^d x_i \le 1 \right\}$$

Fact:

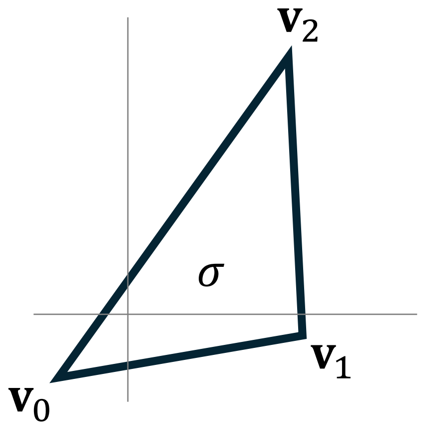
Given a set of $\{f_0, \dots, f_d\} \in \mathbb{R}$ values at the corners of the simplex, there is a unique affine function which evaluates to those values at the corners.

Proof:

Set $f: \mathbb{R}^d \to \mathbb{R}$ to be the function:

$$f(\mathbf{p}) = f_0 + \langle \mathbf{p}, \mathbf{v}_f \rangle \quad \text{with } \mathbf{v}_f = (f_1 - f_0, \dots, f_d - f_0)^\top$$

Review: simplices in $\mathbb{R}^d$



Definition:

A d-dimensional *simplex* $\sigma \subset \mathbb{R}^d$ is the convex hull of $d+1$ points* $\{\mathbf{v}_0, \dots, \mathbf{v}_d\} \subset \mathbb{R}^d$

Fact:

There is a unique affine map $\phi_\sigma\colon \mathbb{R}^d \to \mathbb{R}^d$ taking the unit-right-simplex to σ.

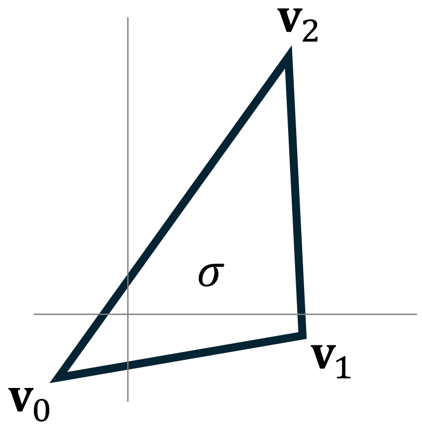
Proof:

Set ϕ_σ to be the map:

$$\phi_\sigma(\mathbf{p}) = \mathbf{v}_0 + (\mathbf{v}_1 - \mathbf{v}_0 | \cdots | \mathbf{v}_d - \mathbf{v}_0) \cdot \mathbf{p}$$

*Throughout will be assuming that geometry is "in general position"

Review: simplices in $\mathbb{R}^d$



$$\phi_\sigma(\mathbf{p}) = \mathbf{v}_0 + (\mathbf{v}_1 - \mathbf{v}_0 | \cdots | \mathbf{v}_d - \mathbf{v}_0) \cdot \mathbf{p}$$

Definition:

Given a point $\mathbf{p} \in \mathbb{R}^d$, the point

$$\mathbf{p}^\sigma = \phi_\sigma^{-1}(\mathbf{p})$$

is the *barycentric coordinates of* $\mathbf{p}$ *with respect to the simplex* σ.

Fact:

The barycentric coordinates $\mathbf{p}^\sigma = \phi_\sigma^{-1}(\mathbf{p})$ satisfy:

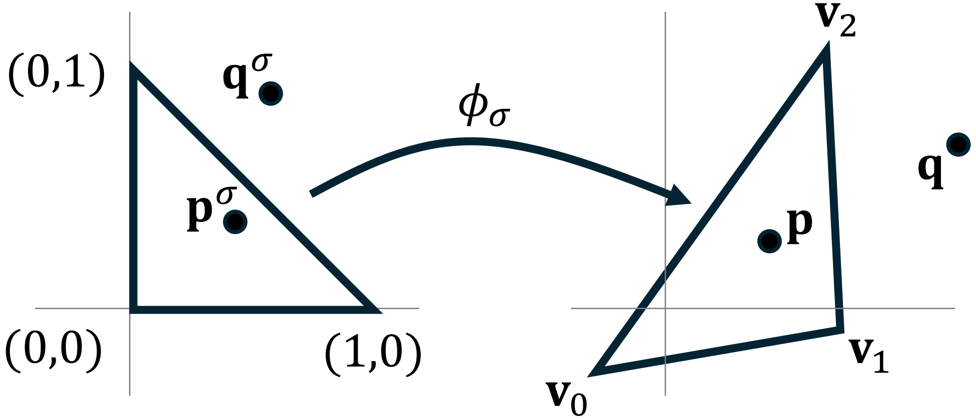
$$\mathbf{p} = \mathbf{v}_0 \cdot (1 - p_1^\sigma - \cdots - p_d^\sigma) + \mathbf{v}_1 \cdot p_1^\sigma + \cdots + \mathbf{v}_d \cdot p_d^\sigma$$

Review: barycentric coordinates

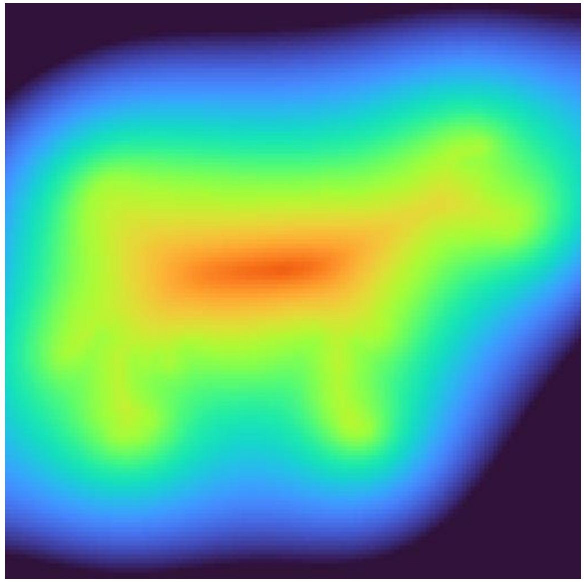
Example:

Given a d-dimensional simplex $\sigma \subset \mathbb{R}^d$ point $\mathbf{p} \in \mathbb{R}^d$, check if $\mathbf{p} \in \sigma$.

- Compute the barycentric coordinates of $\mathbf{p}$:
$$\mathbf{p}^\sigma = \phi_\sigma^{-1}(\mathbf{p})$$
- Check if the barycentric coordinates are in the unit-right-simplex $\mathbb{T}^d$:
$$0 \le p_i^\sigma \le 1 \quad \text{and} \quad \sum_{i=1}^{d} p_i^\sigma \le 1$$



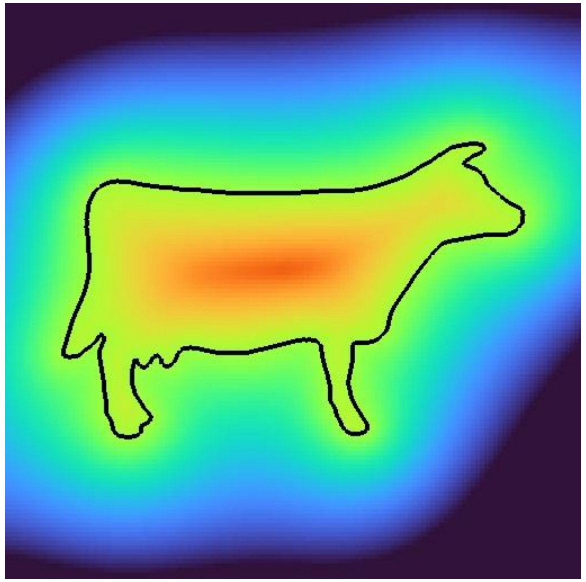
Goal (analytic)



Input:

- A real-valued function, $g: [0,1]^2 \to \mathbb{R}$
- A real level-set-value, $\alpha \in \mathbb{R}$

Goal (analytic)



Input:

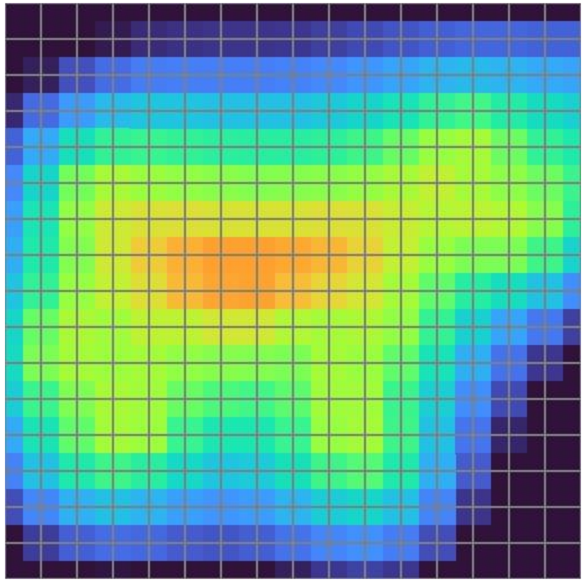
- A real-valued function, $g: [0,1]^2 \to \mathbb{R}$
- A real level-set-value, $\alpha \in \mathbb{R}$

Output:

- The α level-set of g:

$$g^{-1}(\alpha) = \{\mathbf{p} \in [0,1]^2 | g(\mathbf{p}) = \alpha\}$$

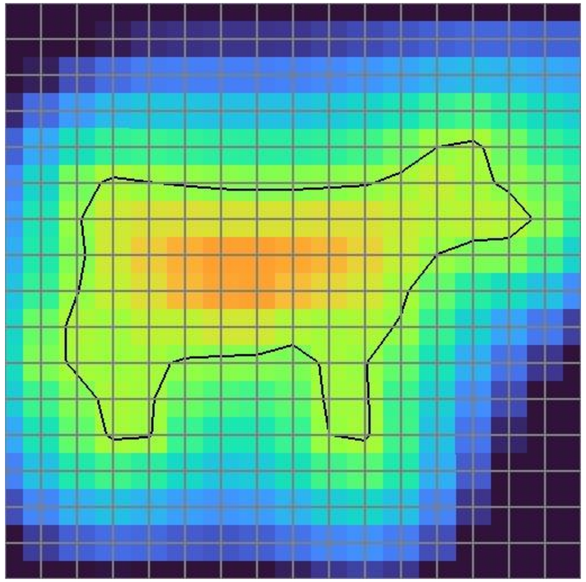
Goal (discrete)



Input:

- A regular real-valued 2D grid, $g \in \mathbb{R}^{N \times N}$, with values at corners
- A real level-set-value, $\alpha \in \mathbb{R}$

Goal (discrete)



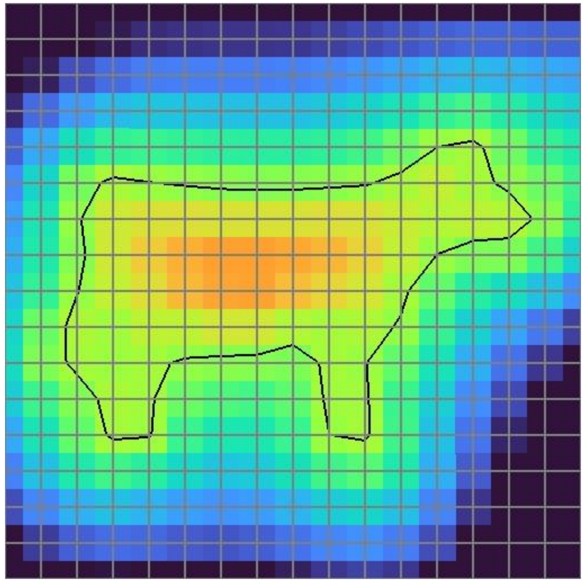
Input:

- A regular real-valued 2D grid, $g \in \mathbb{R}^{N \times N}$, with values at corners
- A real level-set-value, $\alpha \in \mathbb{R}$

Output:

- A piecewise-linear curve corresponding to the α level-set of g.
  - A set of vertices, $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\} \subset \mathbb{R}^2$
  - Set of edges $E \subset \{1, \dots, n\} \times \{1, \dots, n\}$

Approach



- Interpolate the function’s values from the corners.
- Independently extract the level-set from each cell.

Observation:

- If the interpolated function is continuous,
  the per-cell level-sets will match seamlessly.

Challenge:

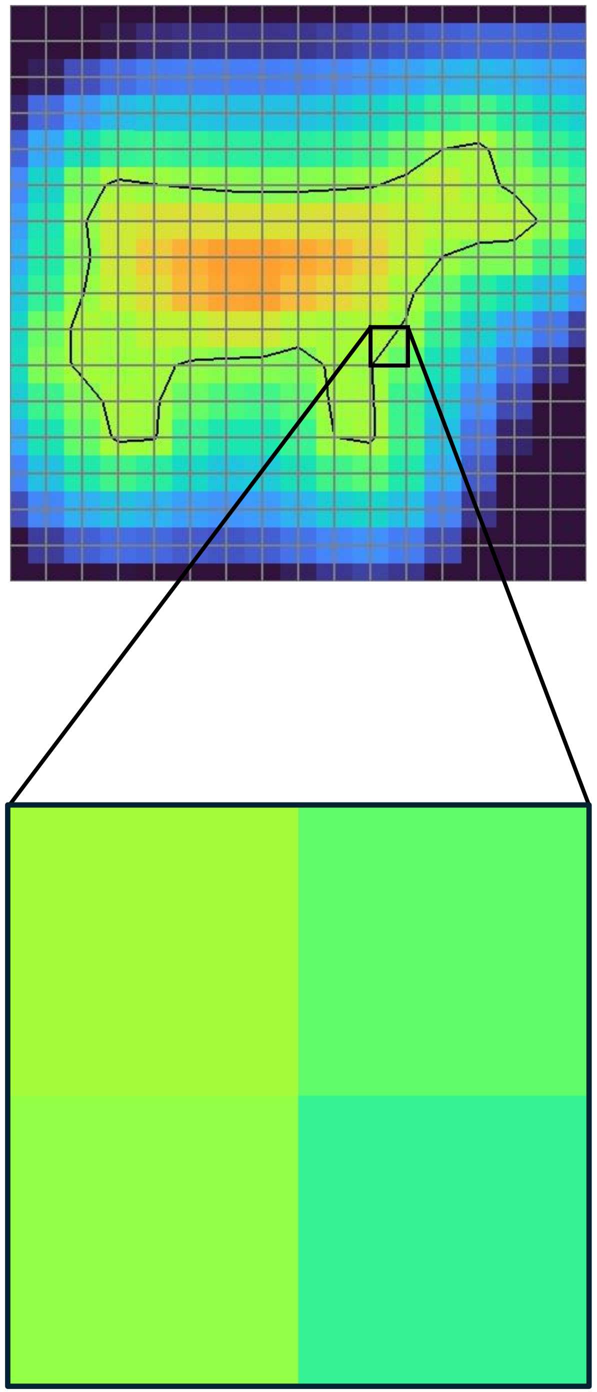
- How to interpolate from corners?

Challenge

Q: How to interpolate from corners?

A: Nearest interpolation

- ✗ Not continuous

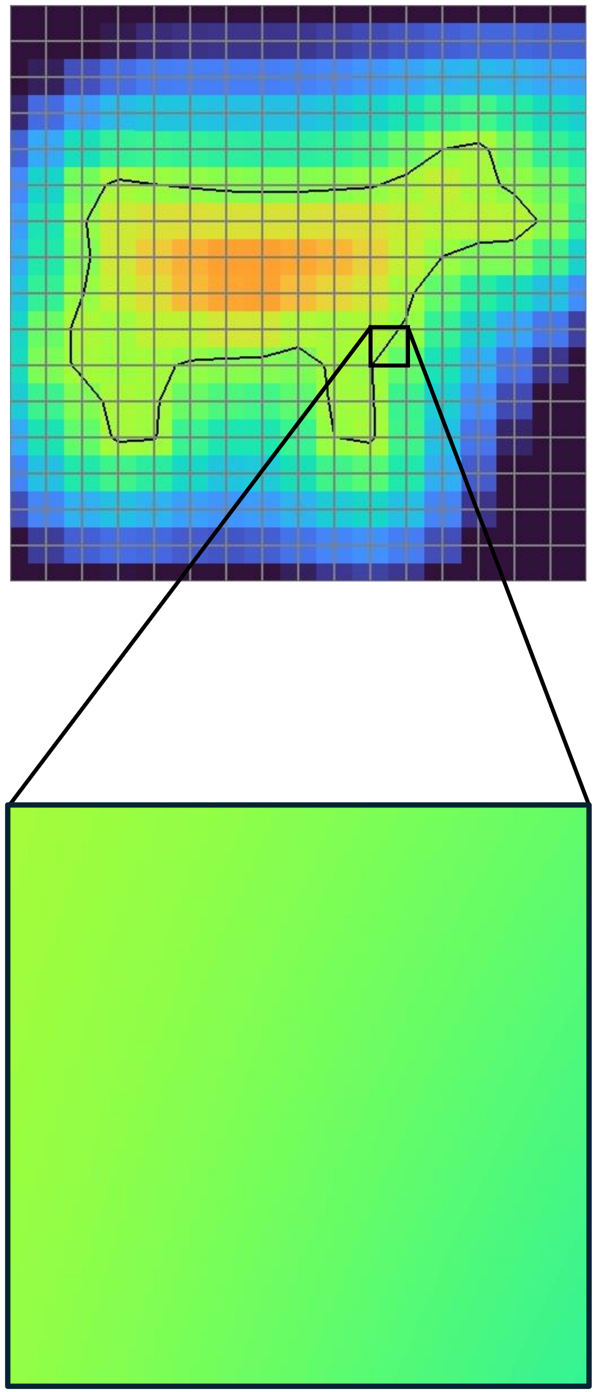


Challenge

Q: How to interpolate from corners?

A: Bi-linear interpolation

$$\begin{aligned} f(x, y) = &(1-x)\cdot(1-y) f_{00} \\ &+(1-x)\cdot y\cdot f_{01} \\ &+x\cdot(1-y)\cdot f_{10} \\ &+x\cdot y\cdot f_{11} \end{aligned}$$

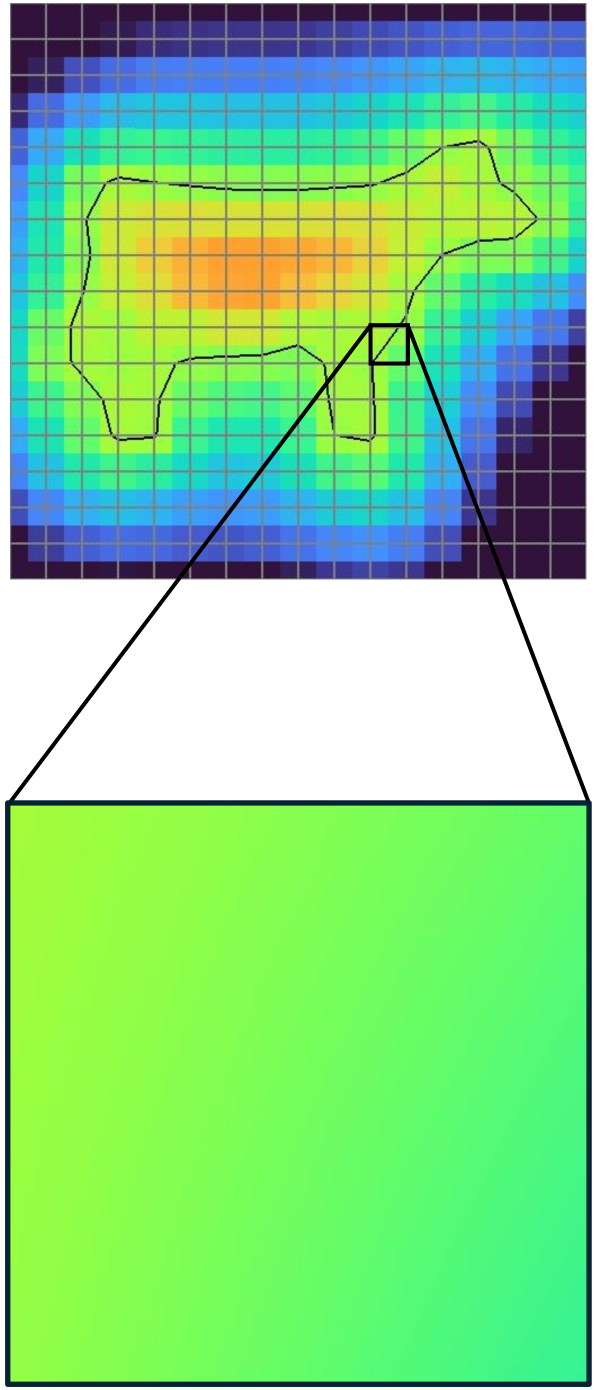


Challenge

Q: How to interpolate from corners?

A: Bi-linear interpolation

✓Continuous

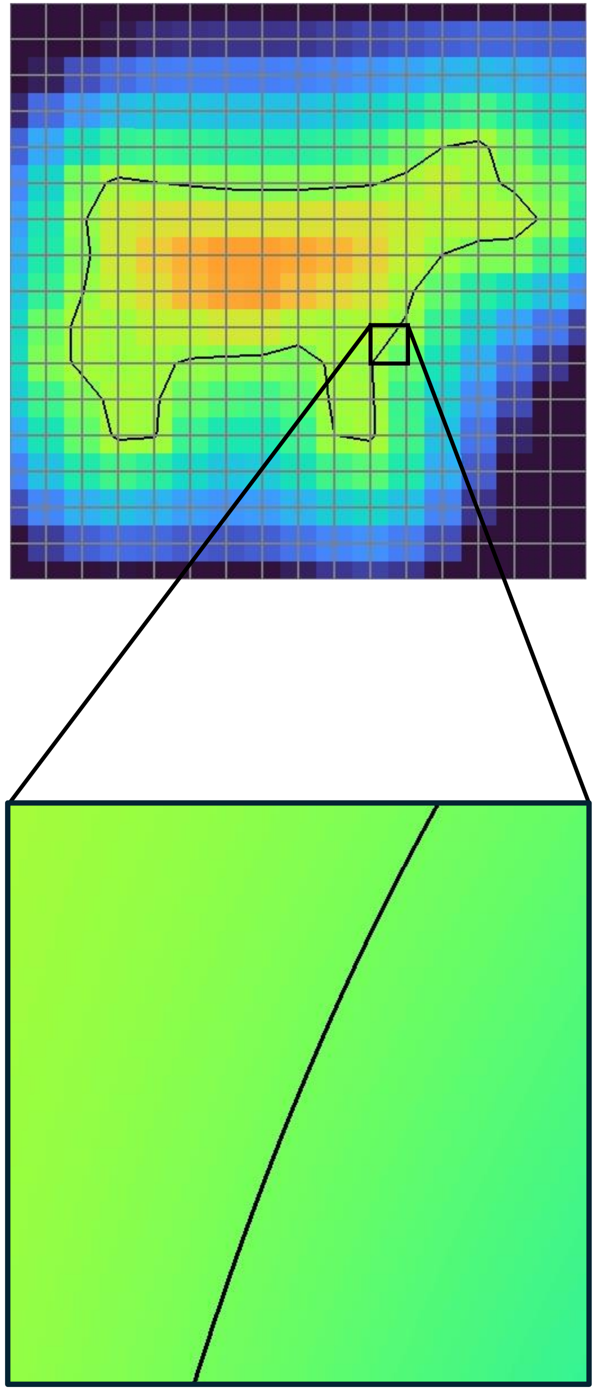


Challenge

Q: How to interpolate from corners?

A: Bi-linear interpolation

- ✓ Continuous
- ✗ Level-sets are curved
  (the level-set gets more complex as the dimension of the domain is increased)

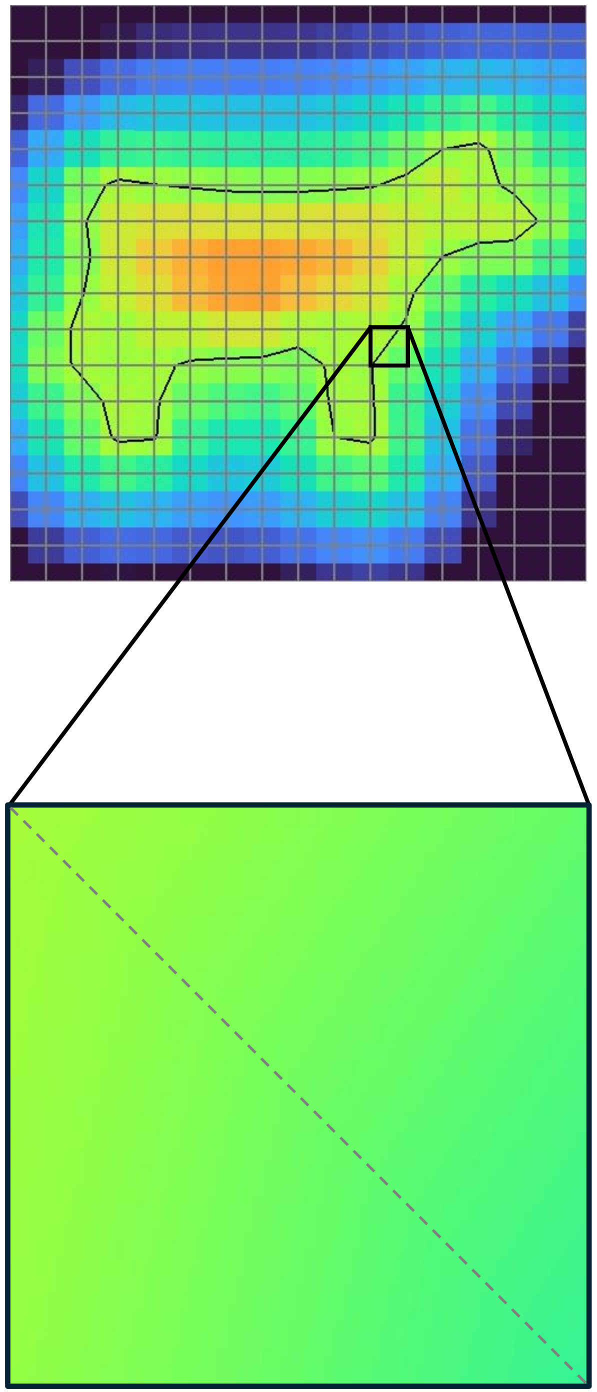


Challenge

Q: How to interpolate from corners?

A: Add a diagonal and linearly interpolate within each triangle

$$f(x, y) = (1 - x - y) \cdot f_{00} + x \cdot f_{10} + y \cdot f_{01}$$

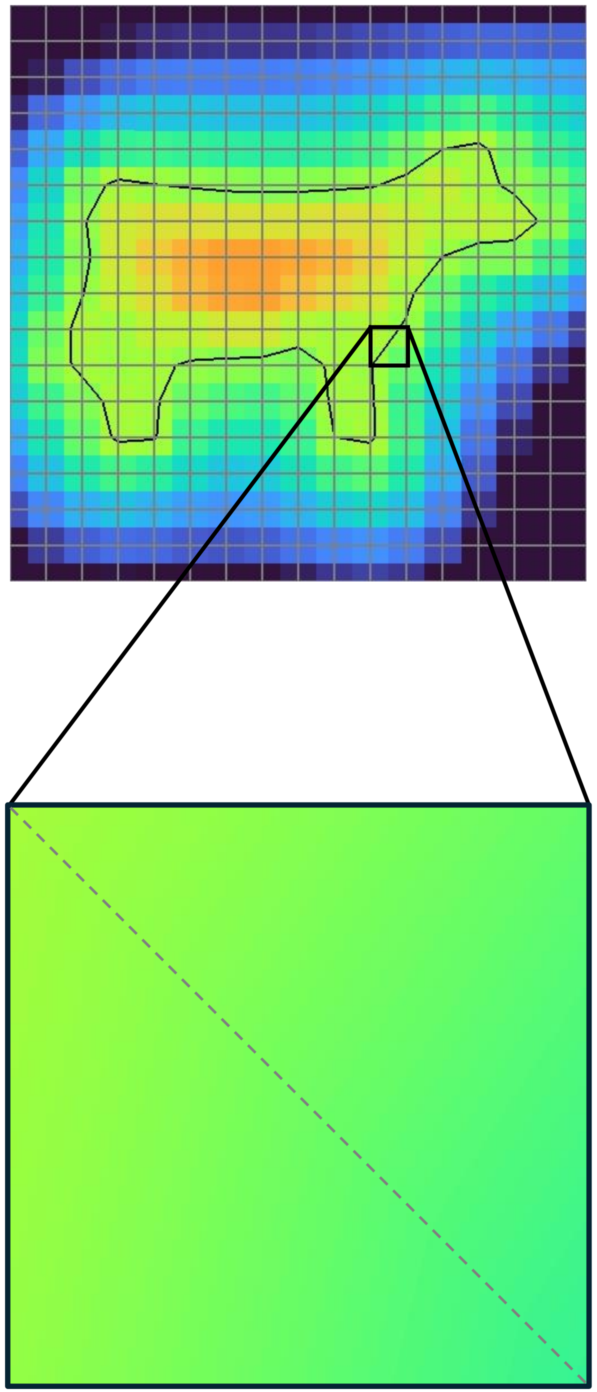


Challenge

Q: How to interpolate from corners?

A: Add a diagonal and linearly interpolate within each triangle

✓Continuous



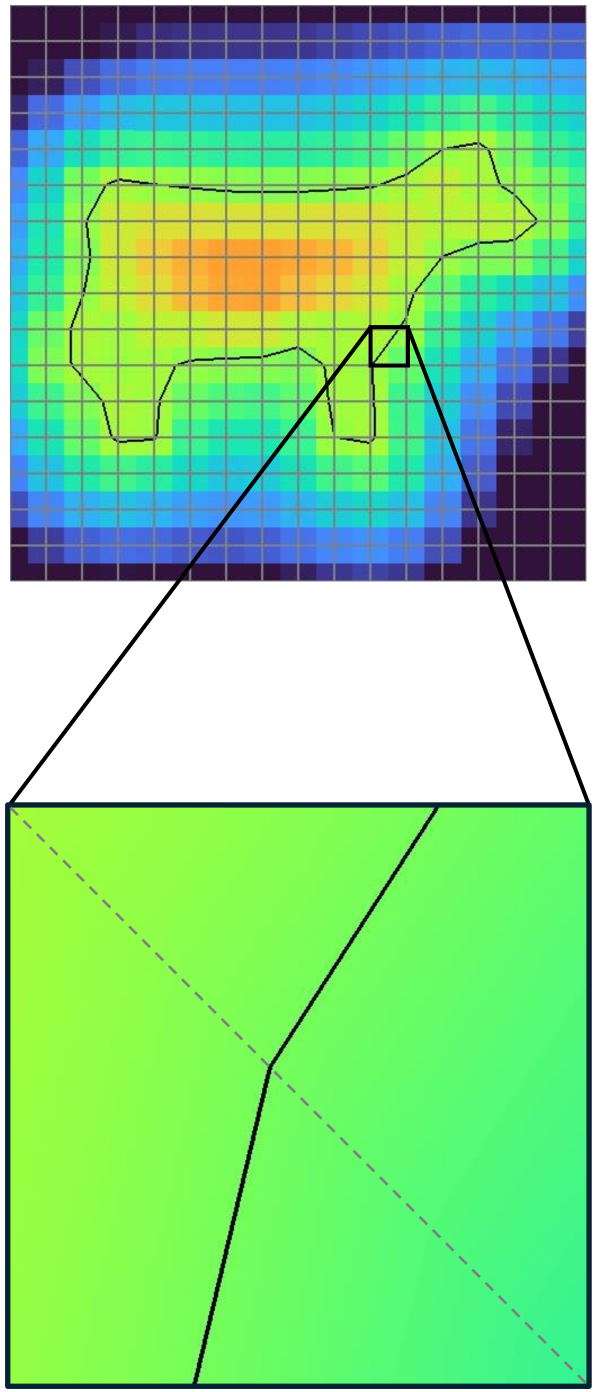
Challenge

Q: How to interpolate from corners?

A: Add a diagonal and linearly interpolate within each triangle

✓Continuous

✓Level-sets are straight

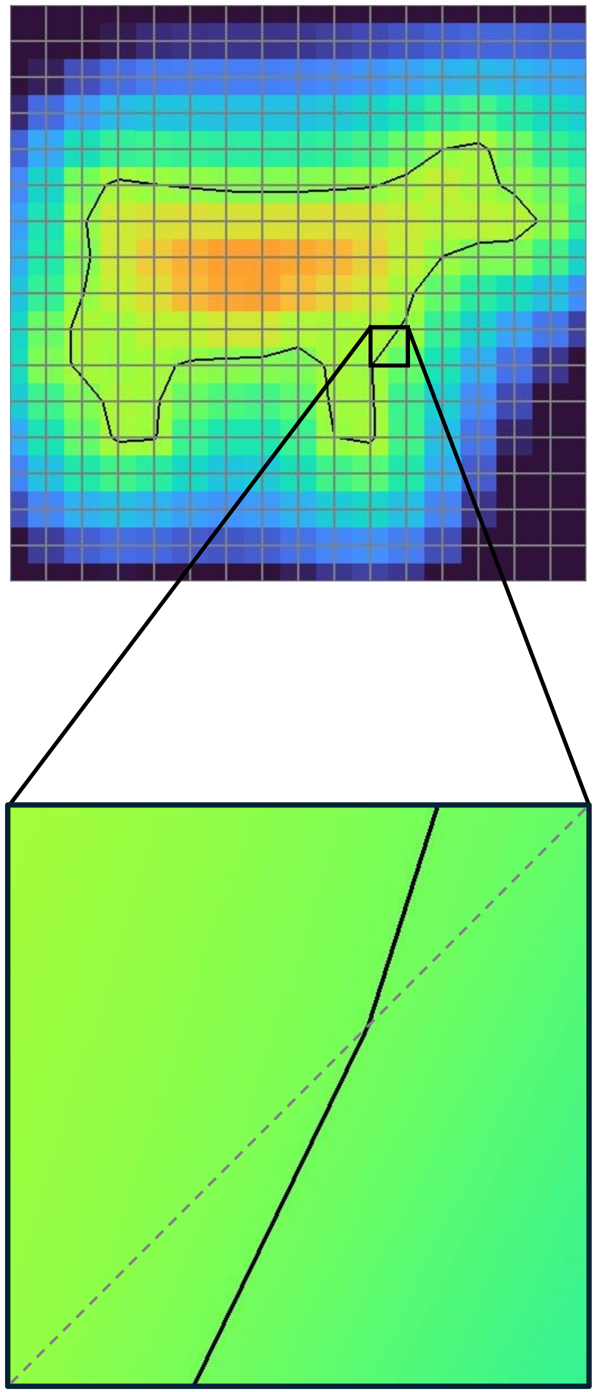


Challenge

Q: How to interpolate from corners?

A: Add a diagonal and linearly interpolate within each triangle

- ✓Continuous
- ✓Level-sets are straight
- ✗ Diagonal choice is arbitrary



Challenge

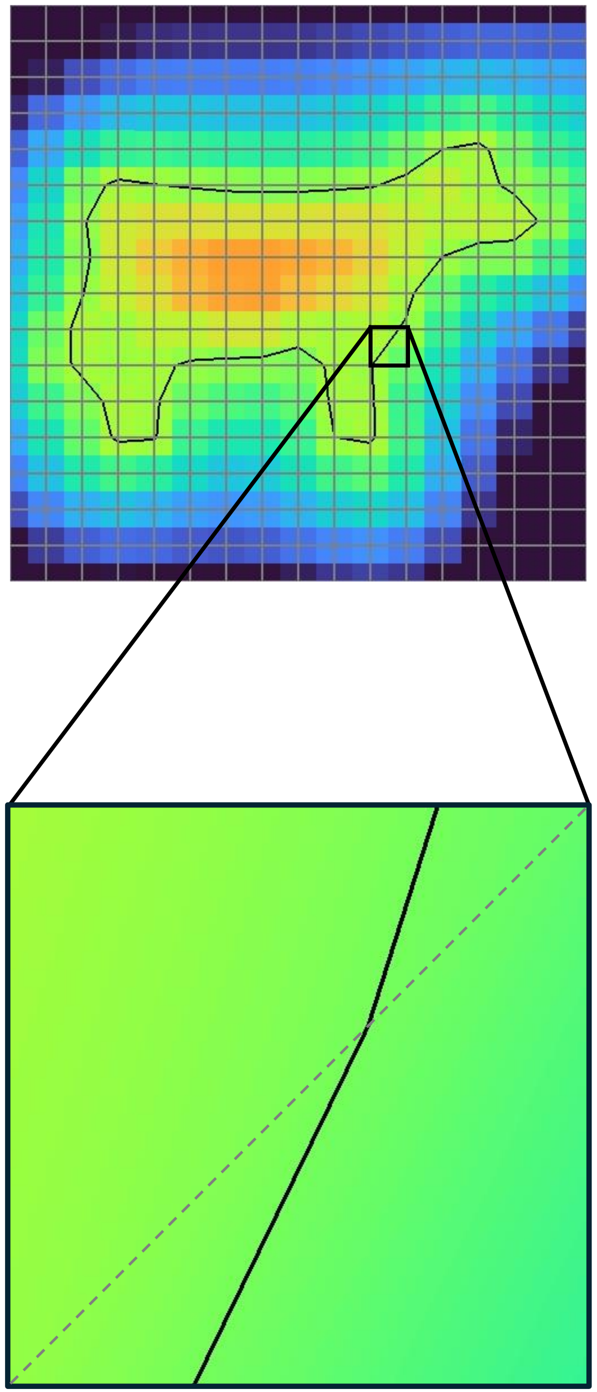
Q: How to interpolate from corners?

A: Add a diagonal and linearly interpolate within each triangle

- ✓Continuous
- ✓Level-sets are straight
- ✗ Diagonal choice is arbitrary

Note:

The functions will be continuous along the shared edge between edge-adjacent triangles.



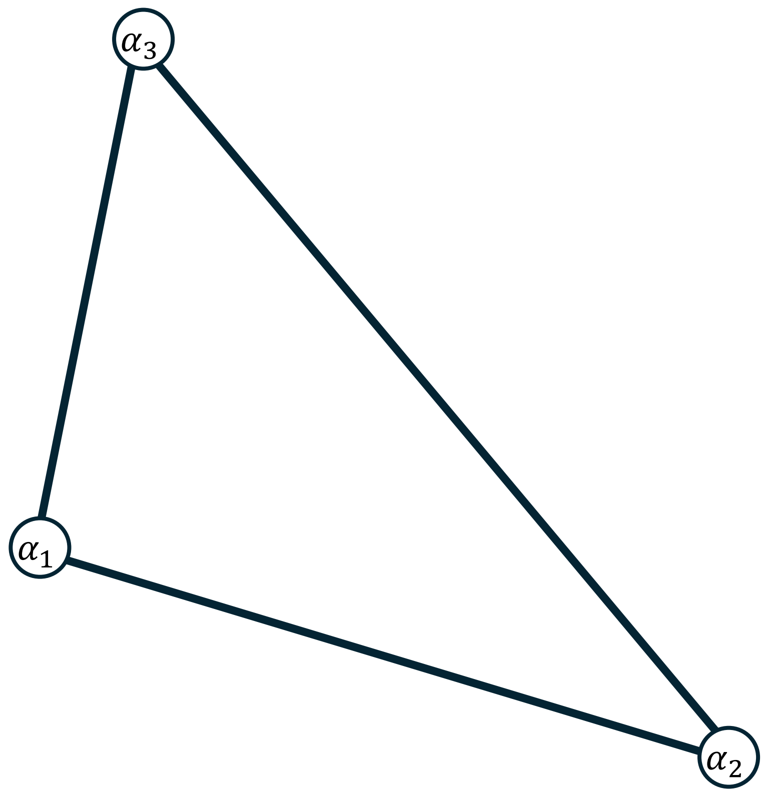
Implementation

Input:

- A triangle $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, with $\mathbf{v}_i \in \mathbb{R}^2$
- Values $(\alpha_1, \alpha_2, \alpha_3)$, with $\alpha_i \in \mathbb{R}$
- A level-set-value $\alpha \in \mathbb{R}$

Output:

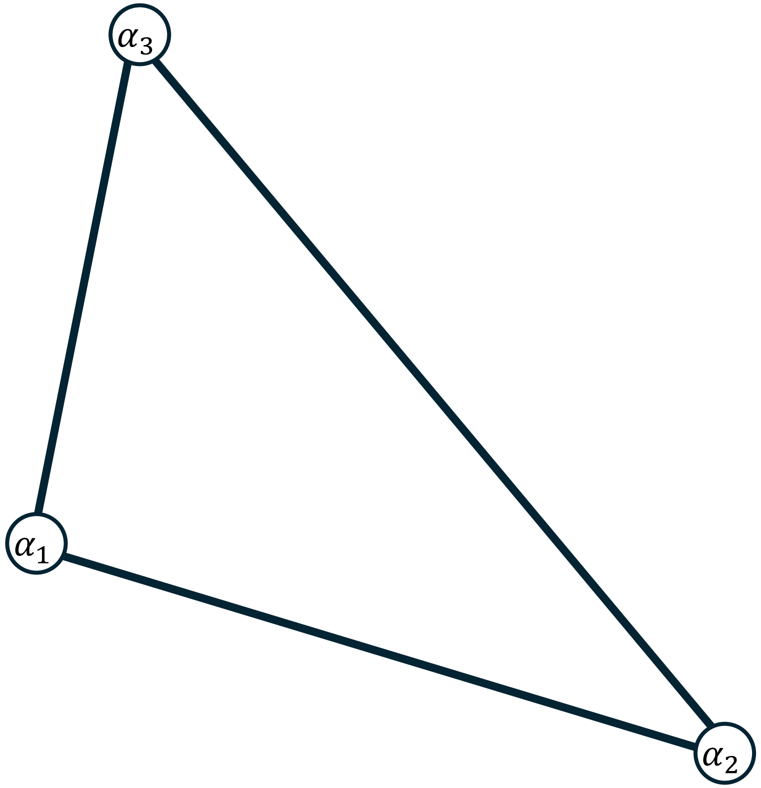
- The geometry of the α level-set within the triangle



Implementation

Assuming general position, there are four cases:

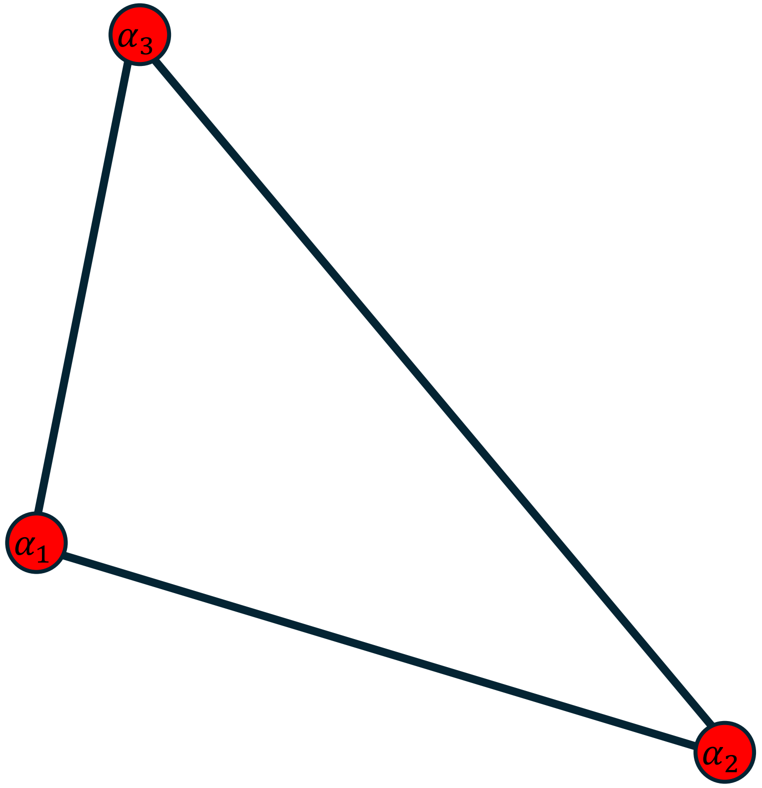
- $\alpha_1, \alpha_2, \alpha_3 > \alpha$
- $\alpha_{i+1}, \alpha_{i+2} > 0$ and $\alpha_i < \alpha$
- $\alpha_i > \alpha$ and $\alpha_{i+1}, \alpha_{i+2} < \alpha$
- $\alpha_1, \alpha_2, \alpha_3 < \alpha$



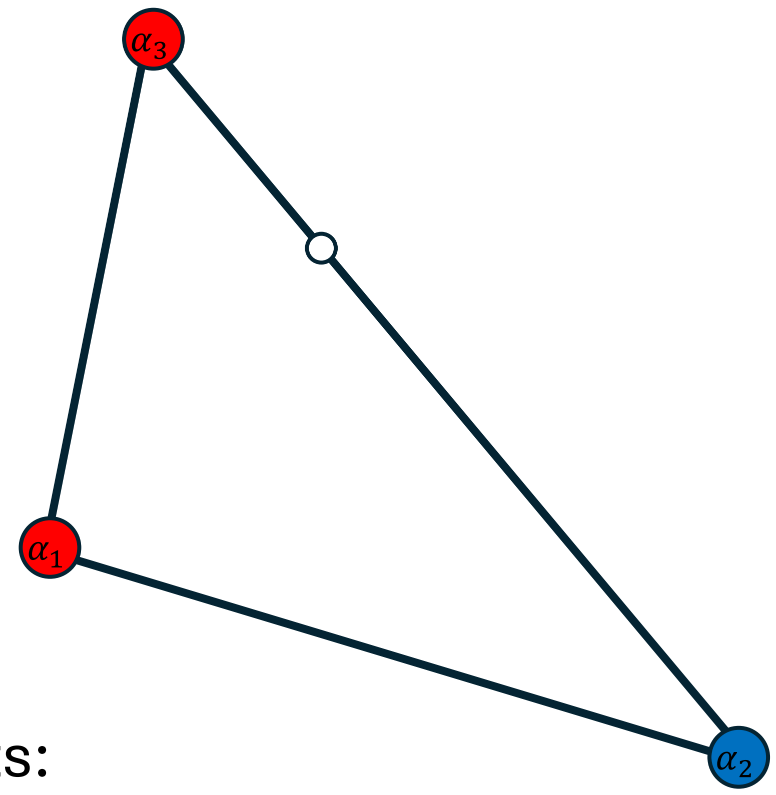
Implementation

$\alpha_1, \alpha_2, \alpha_3 > \alpha$:

- The α level-set does not pass through the triangle



Implementation



$\alpha_{i+1}, \alpha_{i+2} > \alpha$ and $\alpha_i < \alpha$:

- There is an α level-set-vertex along edge $\overline{\mathbf{v}_i \mathbf{v}_{i+1}}$

Computing the position of the level-set-vertex:

- The vertex is the weighted average of the end-points:

$$\mathbf{v} = (1 - s) \cdot \mathbf{v}_i + s \cdot \mathbf{v}_{i+1}$$

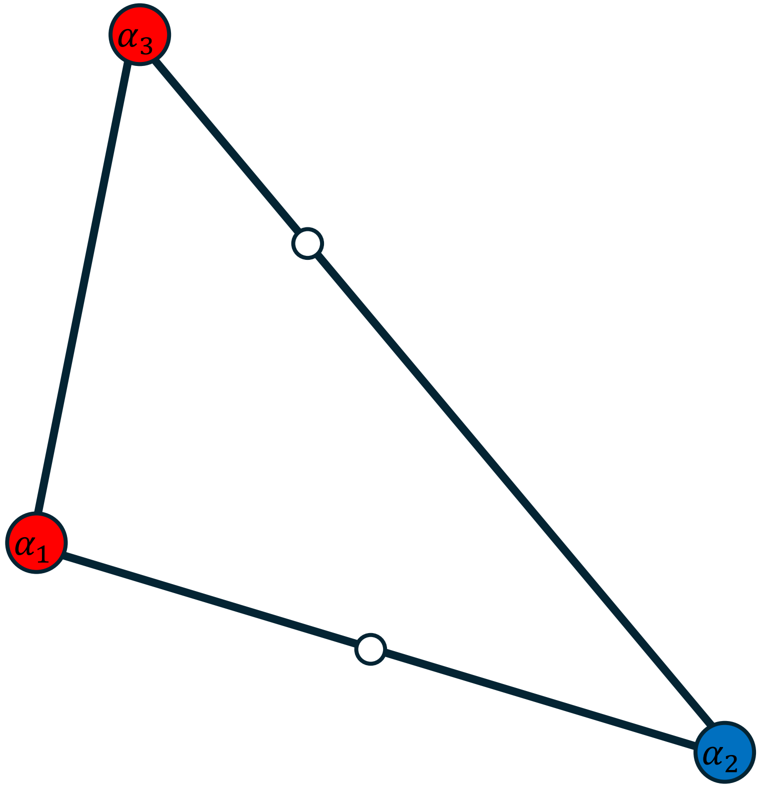
- The weights are the same that make the weighted average of end-point values equal the level-set-value:

$$\alpha = (1 - s) \cdot \alpha_i + s \cdot \alpha_{i+1}$$
$$\Downarrow$$
$$s = \frac{\alpha - \alpha_i}{\alpha_{i+1} - \alpha_i}$$

Implementation

$\alpha_{i+1}, \alpha_{i+2} > \alpha$ and $\alpha_i < \alpha$:

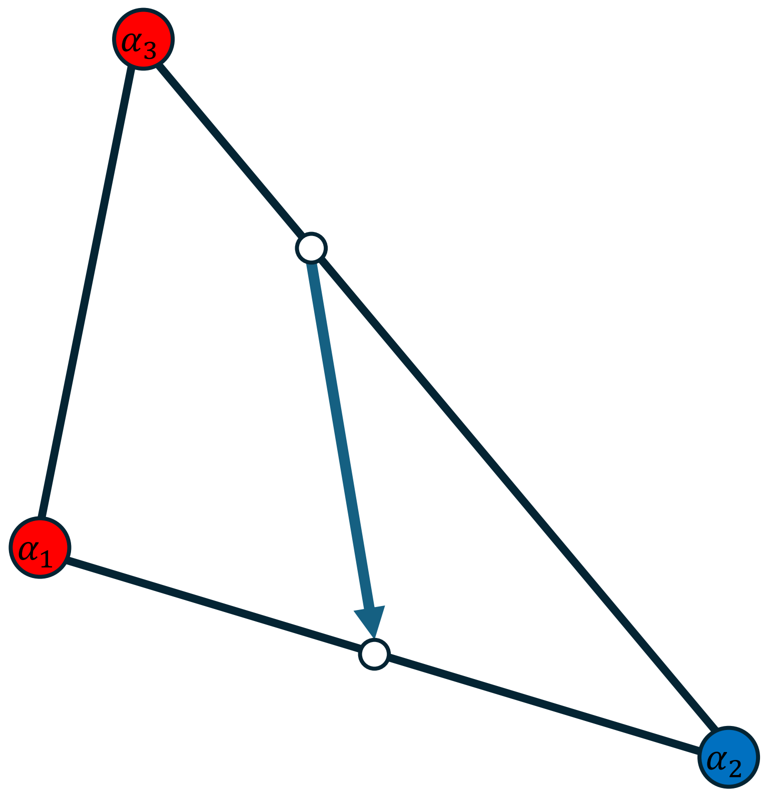
- There is an α level-set-vertex along edge $\overline{\mathbf{v}_i \mathbf{v}_{i+1}}$
- There is an α level-set-vertex along edge $\overline{\mathbf{v}_i \mathbf{v}_{i+2}}$



Implementation

$\alpha_{i+1}, \alpha_{i+2} > \alpha$ and $\alpha_i < \alpha$:

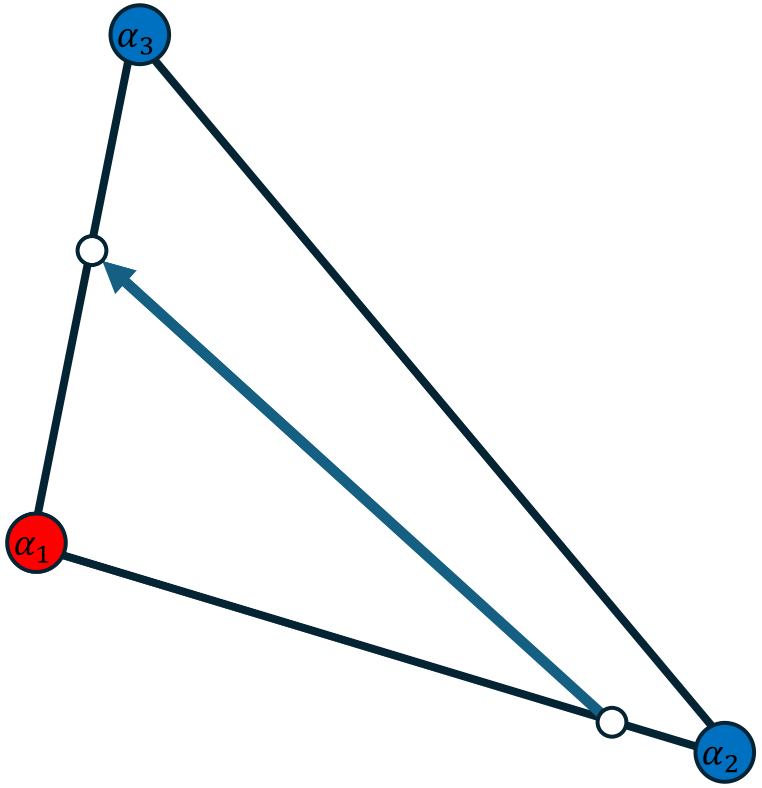
- There is an α level-set-vertex along edge $\overline{\mathbf{v}_i \mathbf{v}_{i+1}}$
- There is an α level-set-vertex along edge $\overline{\mathbf{v}_i \mathbf{v}_{i+2}}$
- There is an α level-set-edge connecting the level-set-vertices



Implementation

$\alpha_i > \alpha$ and $\alpha_{i+1}, \alpha_{i+2} < \alpha$:

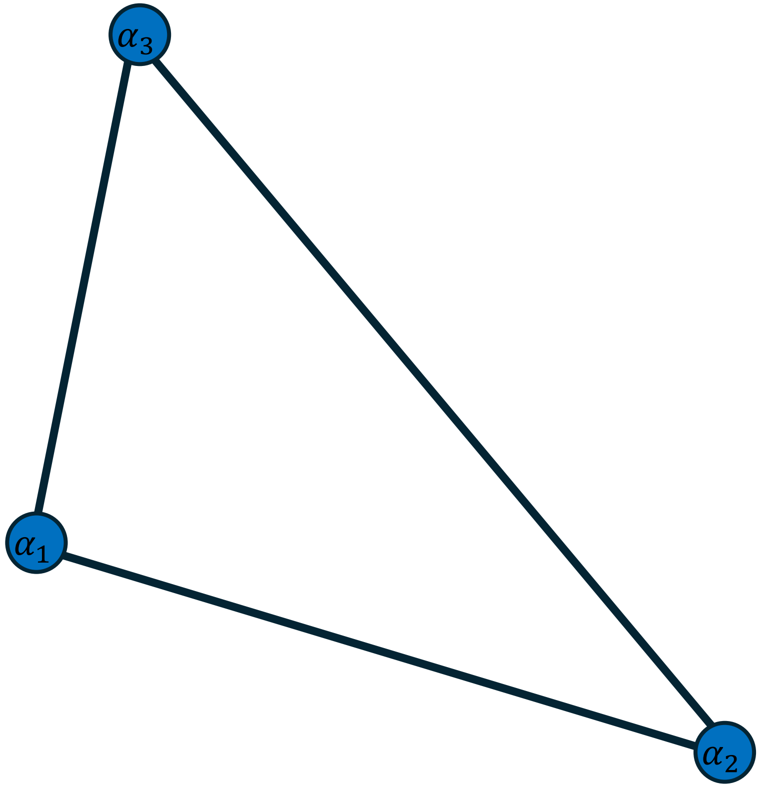
- There is an α level-set-vertex along edge $\overline{\mathbf{v}_i \mathbf{v}_{i+1}}$
- There is an α level-set-vertex along edge $\overline{\mathbf{v}_i \mathbf{v}_{i+2}}$
- There is an α level-set-edge connecting the level-set-vertices



Implementation

$\alpha_1, \alpha_2, \alpha_3 < \alpha$:

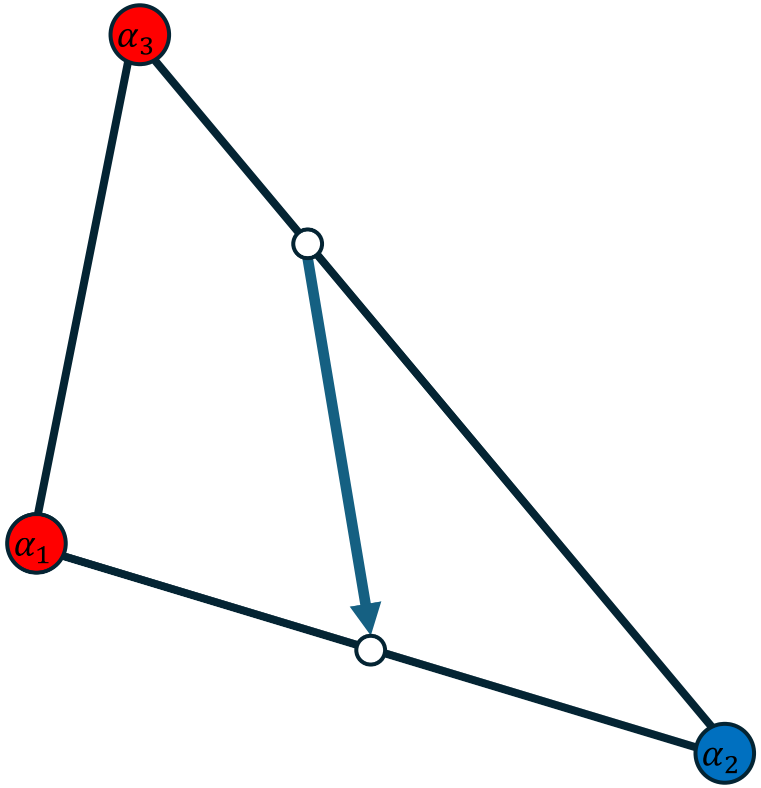
- The α level-set does not pass through the triangle



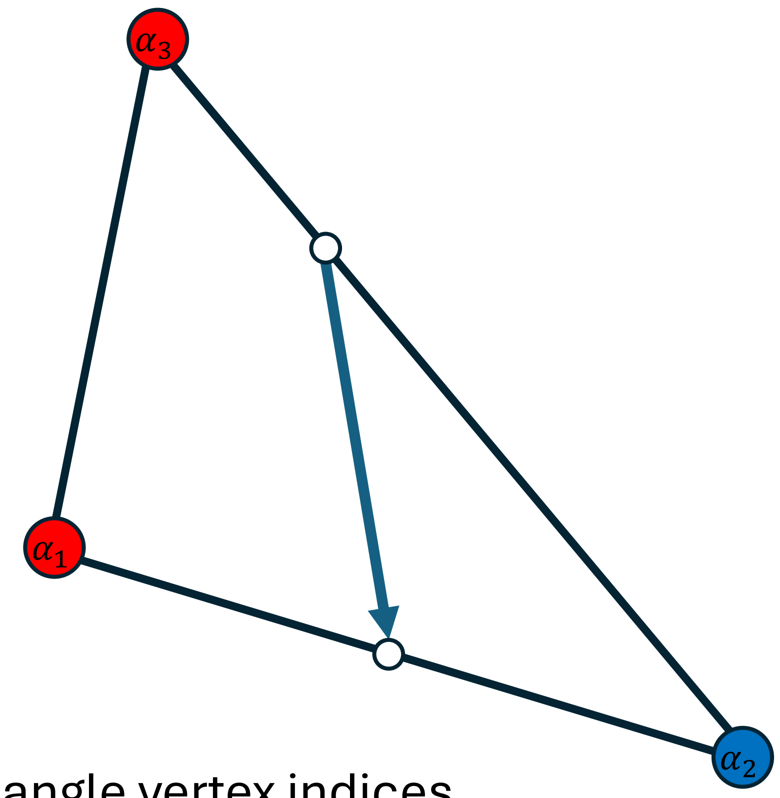
Implementation

Technical details:

- Represent the geometry by having edges store vertex **indices**, not positions.



Implementation



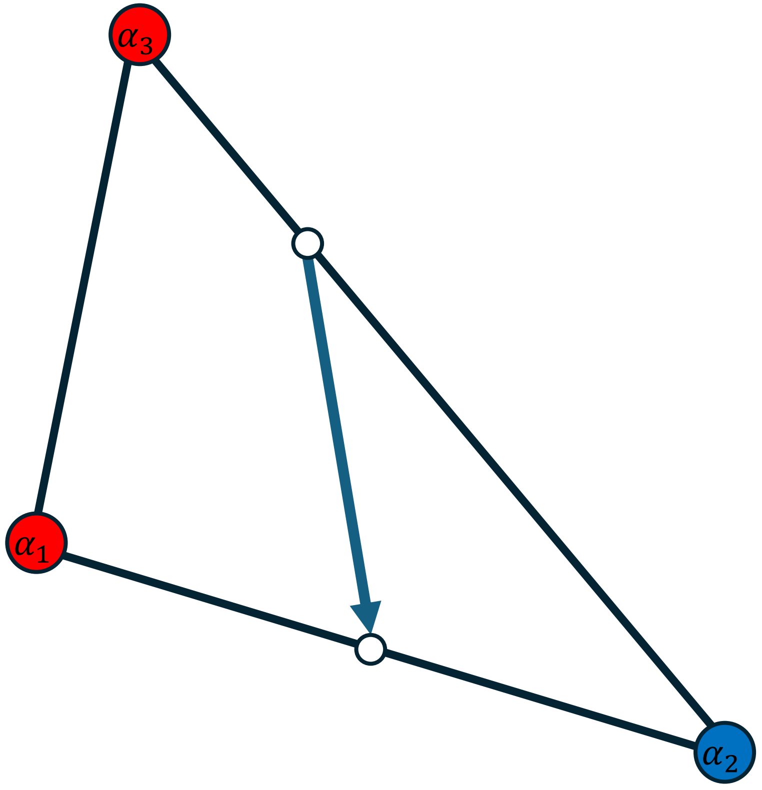
Technical details:

- Avoid generating the same level-set-vertex twice (once for each triangle adjacent to the edge)
  - Triangle vertices are at the corners of the grid
    ⇒ Linearize the triangle vertex indices
  - A level-set-vertex is defined by a pair of triangle vertices
    ⇒ Represent a level-set-vertex by a pair of (linearized) triangle vertex indices
    ⇒ Use an associative array (e.g. `std::map`) to map level-set-vertices to linear indices.

Implementation

Technical details:

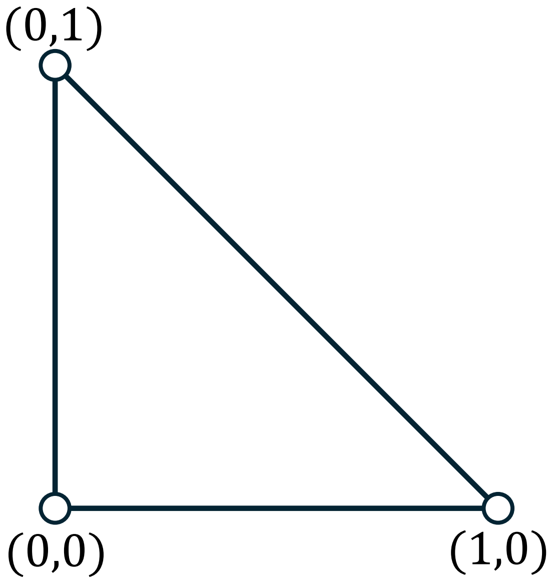
- Orient the edges consistently
  (e.g. so that the corners values less than α are always on the left)



Implementation

Technical details:

- Perform the calculation over the unit-right-simplices
  - $\Rightarrow$ Gives the position of the vertices in barycentric coordinates
- Transform barycentric coordinates to world coordinates

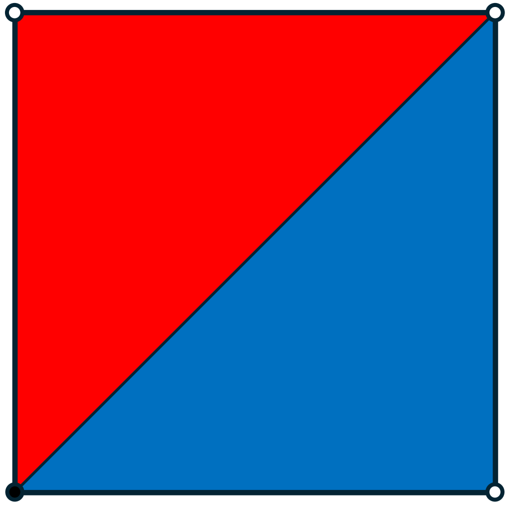


Inductive triangulation

By induction:

Decompose a d-dimensional into $d!$ d-dimensional simplices.

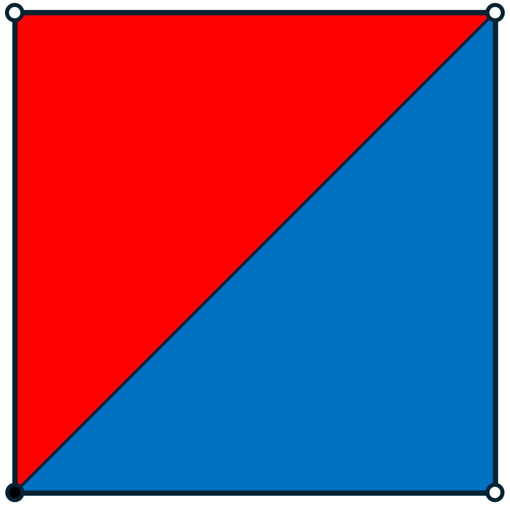
Inductive triangulation



Base case $d = 2$:

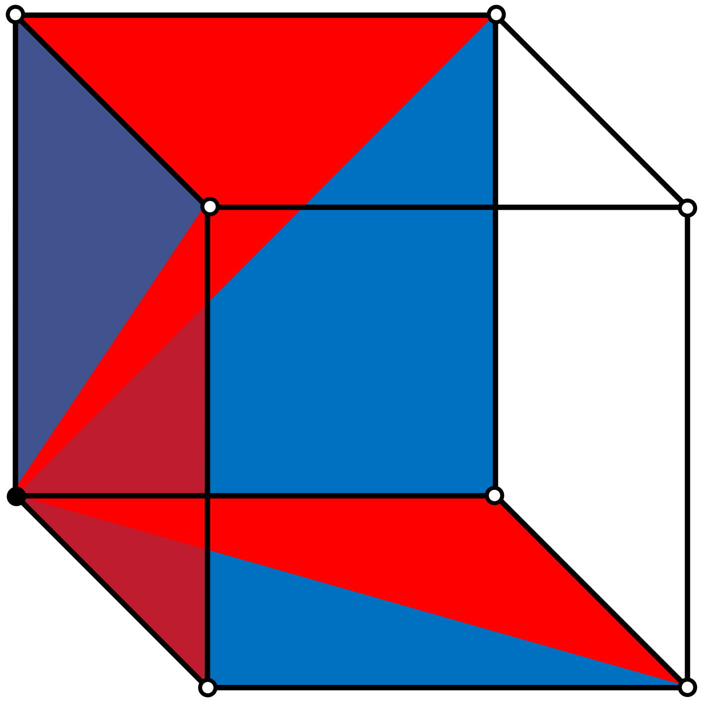
- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - $\Rightarrow$ 2-dimensional cube $\rightarrow$ $2!$ simplices

Inductive triangulation



Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - $\Rightarrow$ 2-dimensional cube $\rightarrow$ 2! simplices



Inductive case:

- Triangulate the d faces incident on the origin
  - $\Rightarrow d!\ (d-1)$-dimensional simplices

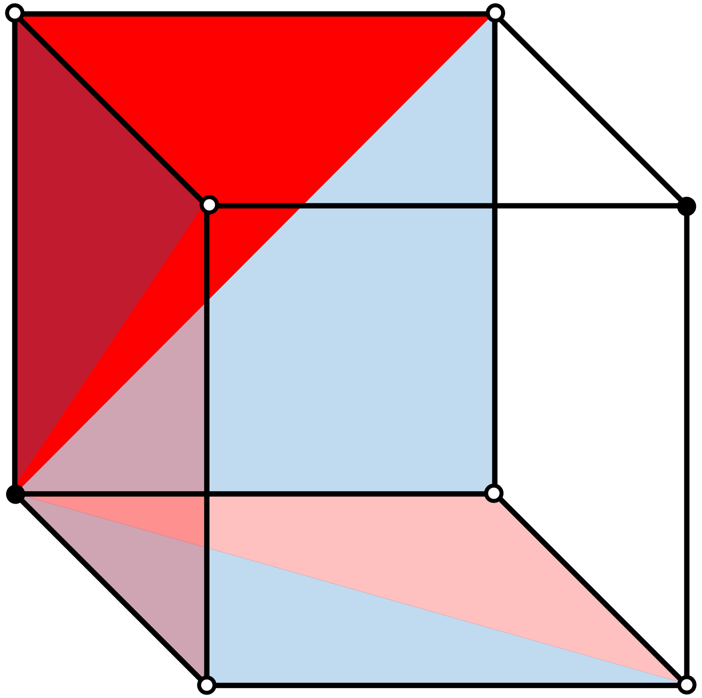
Inductive triangulation

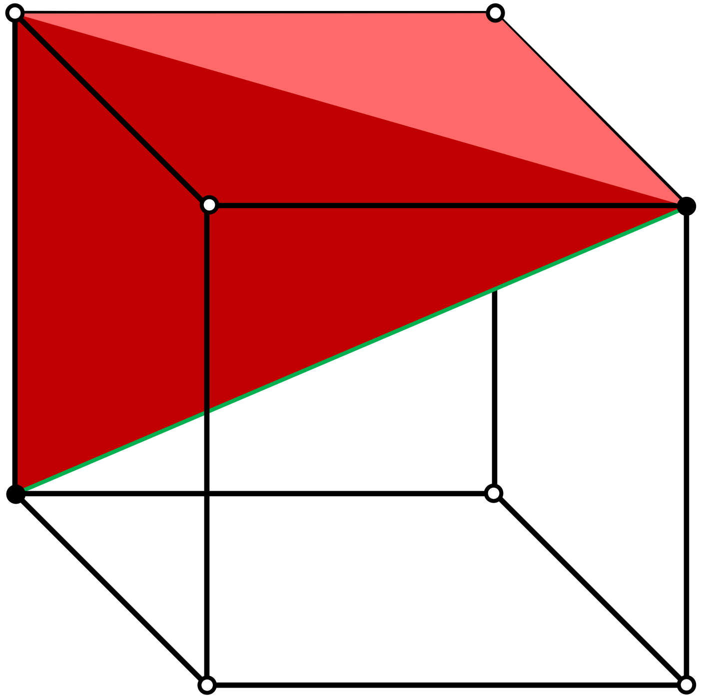
Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - $\Rightarrow$ 2-dimensional cube $\rightarrow$ 2! simplices

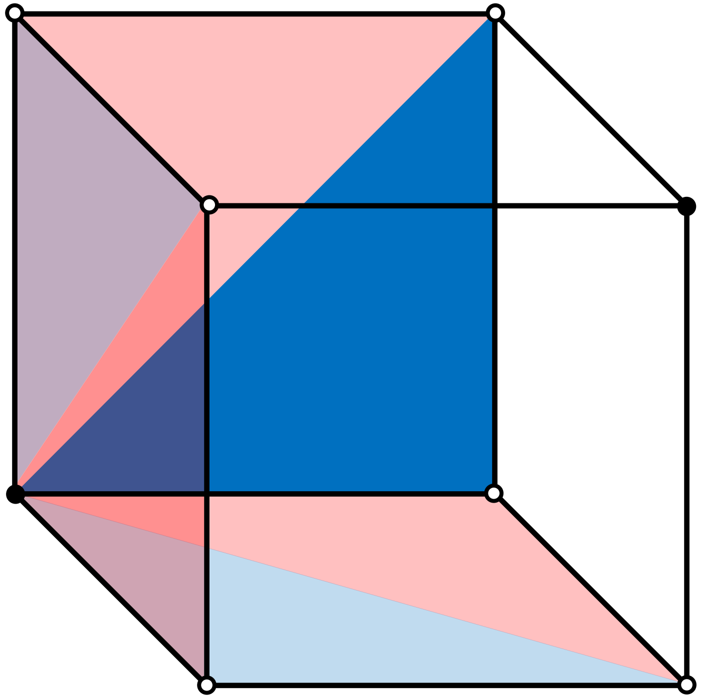
Inductive case:

- Triangulate the d faces incident on the origin
- Fuse each $(d-1)$-dimensional simplex with the point antipodal from the origin



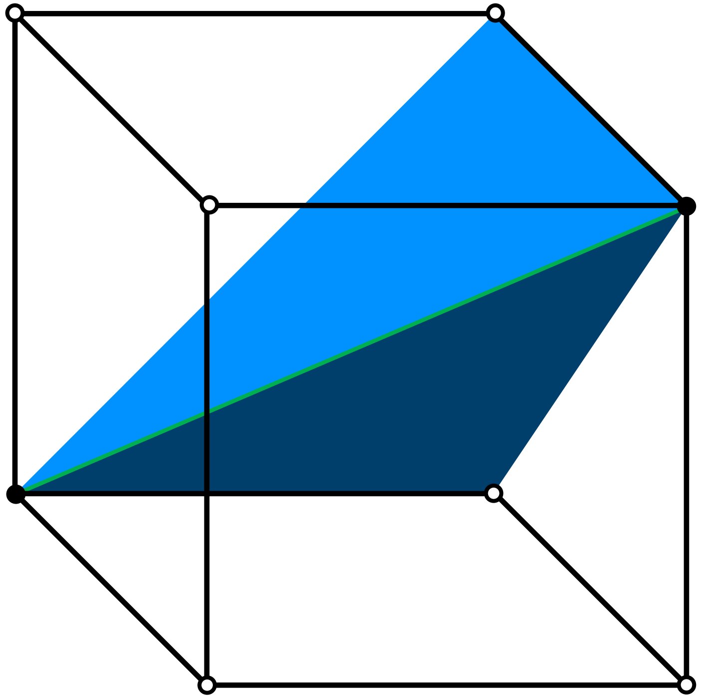


Inductive triangulation



Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - $\Rightarrow$ 2-dimensional cube $\rightarrow$ $2!$ simplices



Inductive case:

- Triangulate the d faces incident on the origin
- Fuse each $(d-1)$-dimensional simplex with the point antipodal from the origin

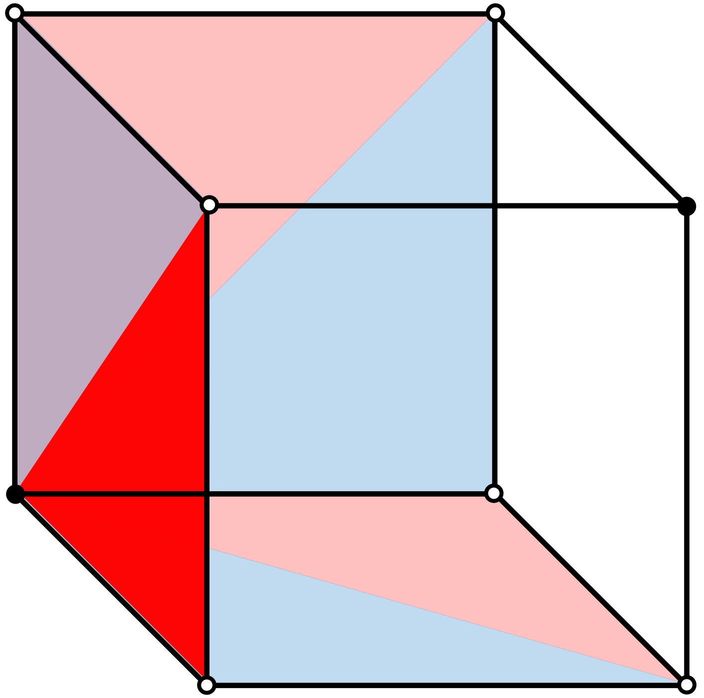
Inductive triangulation

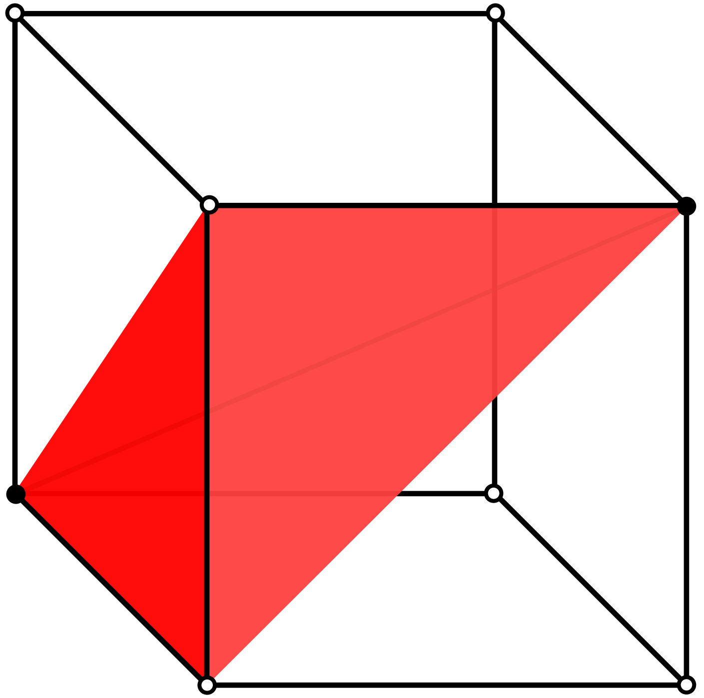
Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - $\Rightarrow$ 2-dimensional cube $\rightarrow$ $2!$ simplices

Inductive case:

- Triangulate the d faces incident on the origin
- Fuse each $(d-1)$-dimensional simplex with the point antipodal from the origin





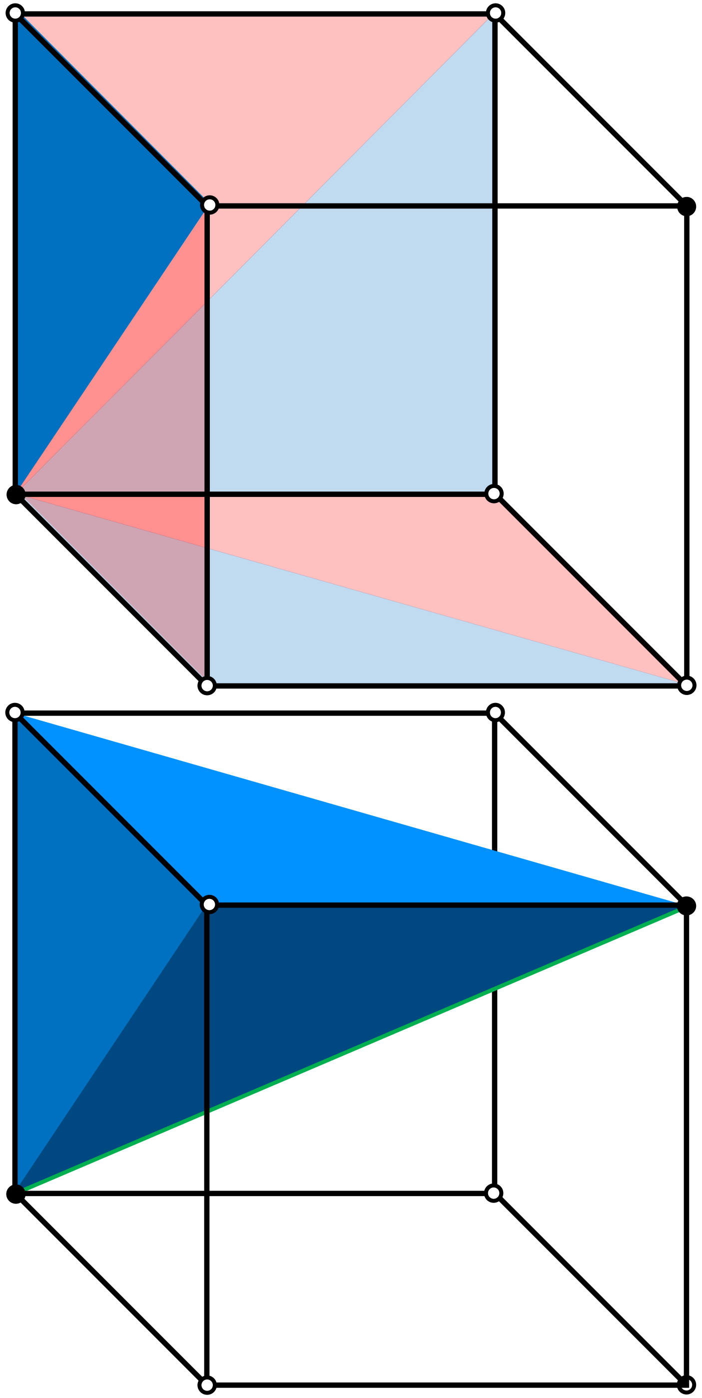
Inductive triangulation

Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - $\Rightarrow$ 2-dimensional cube $\rightarrow$ $2!$ simplices

Inductive case:

- Triangulate the d faces incident on the origin
- Fuse each $(d-1)$-dimensional simplex with the point antipodal from the origin



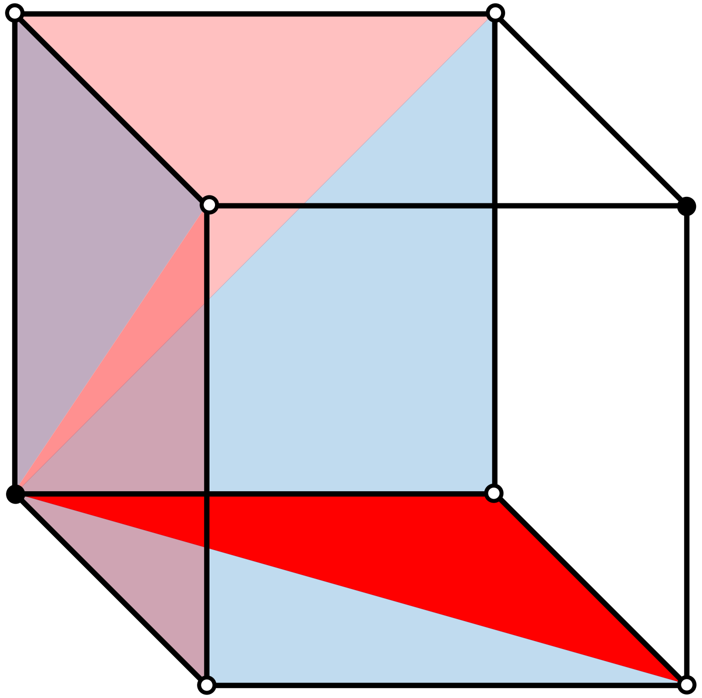
Inductive triangulation

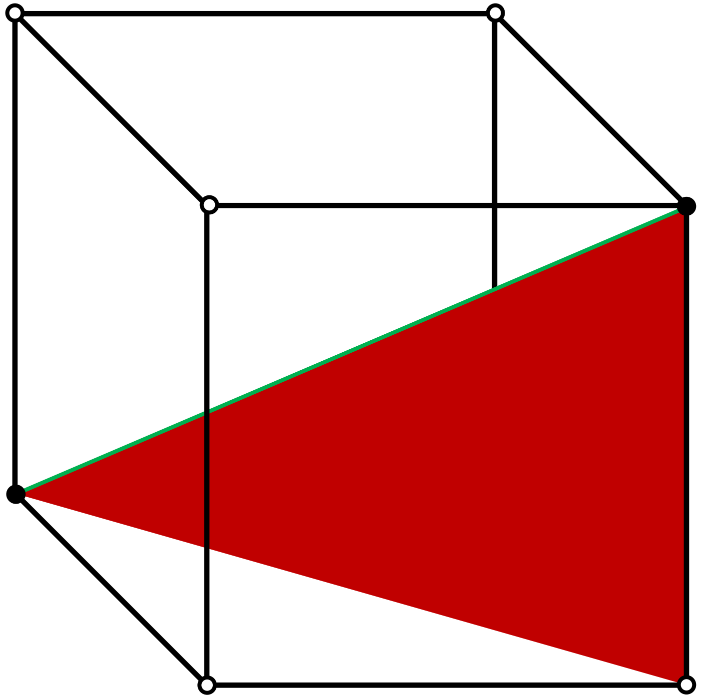
Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - $\Rightarrow$ 2-dimensional cube $\rightarrow$ 2! simplices

Inductive case:

- Triangulate the d faces incident on the origin
- Fuse each $(d-1)$-dimensional simplex with the point antipodal from the origin





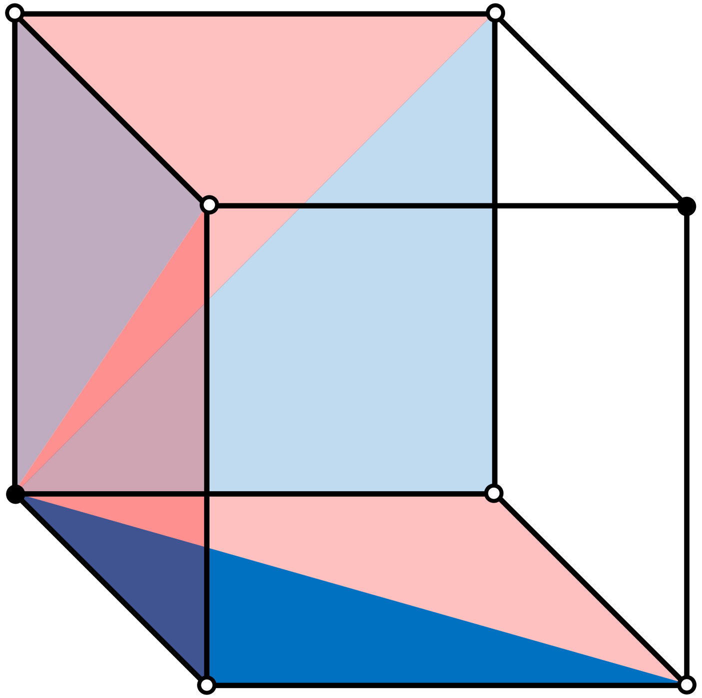
Inductive triangulation

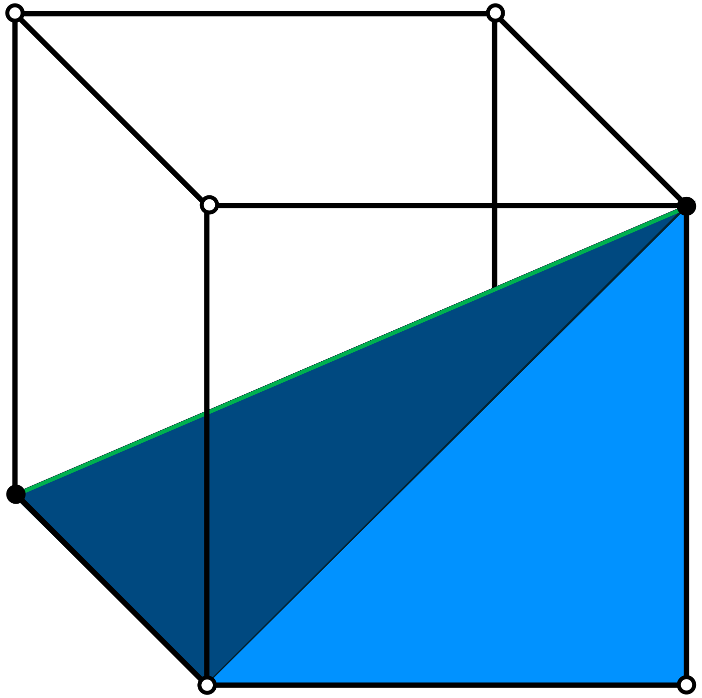
Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner
  - ⇒ 2-dimensional cube → 2! simplices

Inductive case:

- Triangulate the d faces incident on the origin
- Fuse each $(d-1)$-dimensional simplex with the point antipodal from the origin





Inductive triangulation

Base case $d = 2$:

- Triangulate the square by adding the diagonal from the origin to the opposite corner

  $\Rightarrow$ 2-dimensional cube → 2! simplices

Inductive case:

- Triangulate the d faces incident on the origin
- Fuse each $(d-1)$-dimensional simplex with the point antipodal from the origin

  $\Rightarrow$ d-dimensional cell → $d!$ simplices