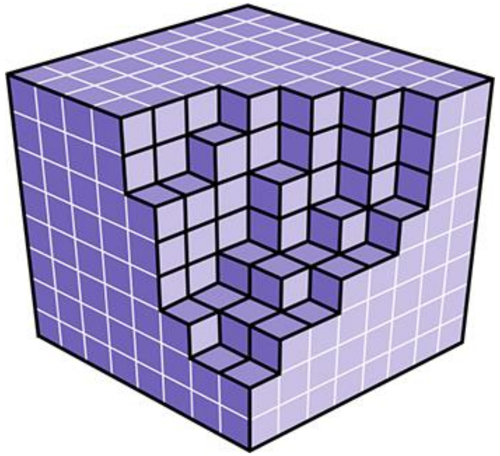
Visualization
– Interpolation

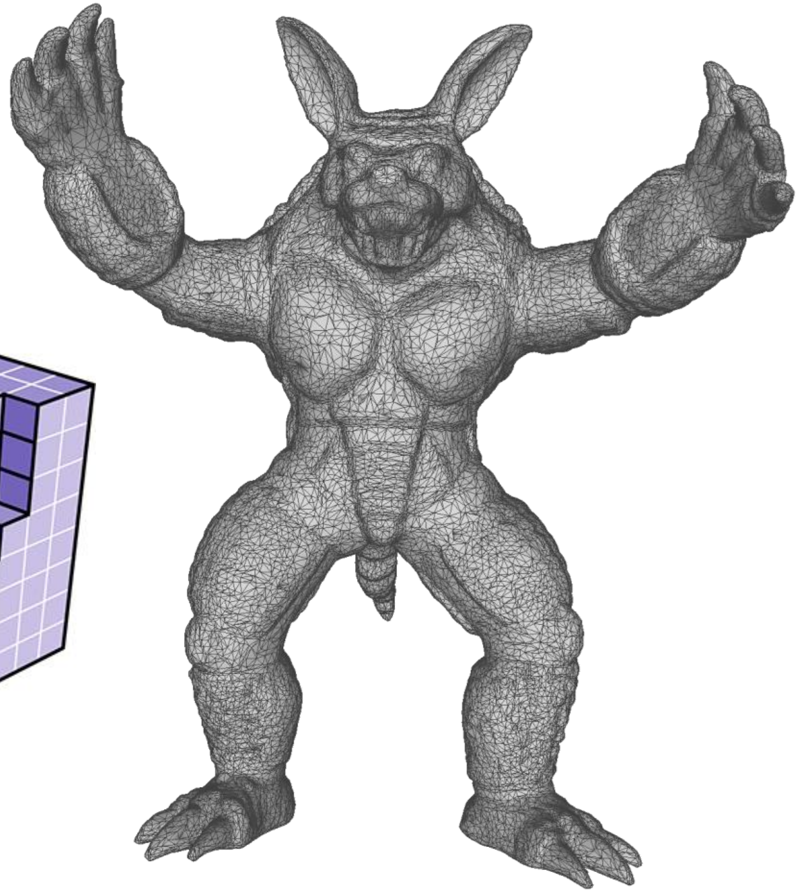
J.-Prof. Dr. habil. Kai Lawonn

Data Representation

Data representation

- Discrete representations
  - Data samples (values) typically given on meshes/grids consisting of cells
  - Compact/efficient data representation



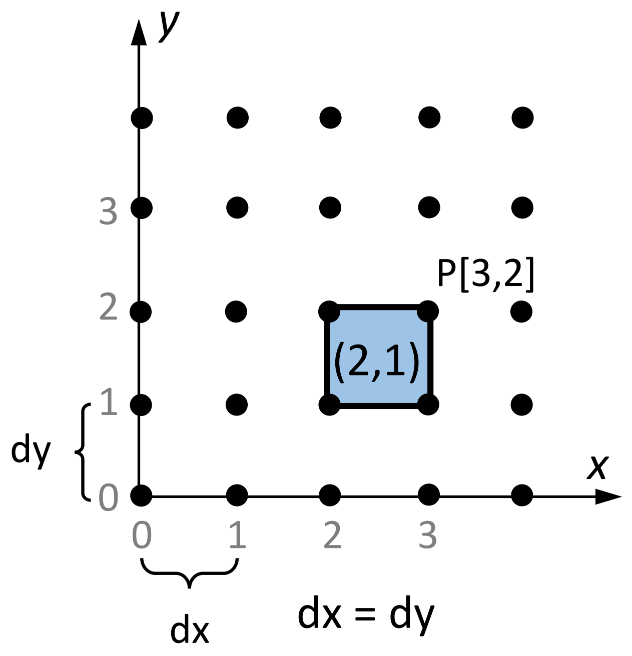


Primitives in different dimensions

dimension	cell	mesh
0D	points	
1D	lines (edges)	polyline
2D	triangles, quadrilaterals (rectangles)	2D mesh
3D	tetrahedra, prisms, hexahedra	3D mesh

Data representation

- Grids: Cartesian or equidistant grid
  - Samples at equidistant intervals along Cartesian coordinate axes
  - Neighboring samples are connected via edges
  - Cells formed by 4 (2D) or 8 (3D) samples
  - Cells and samples (grid vertices) are numbered sequentially with respect to increasing coordinates

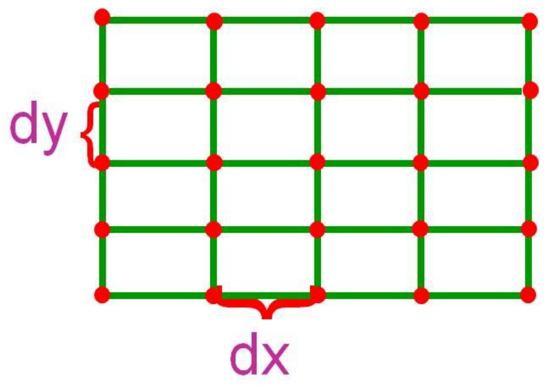


Data representation

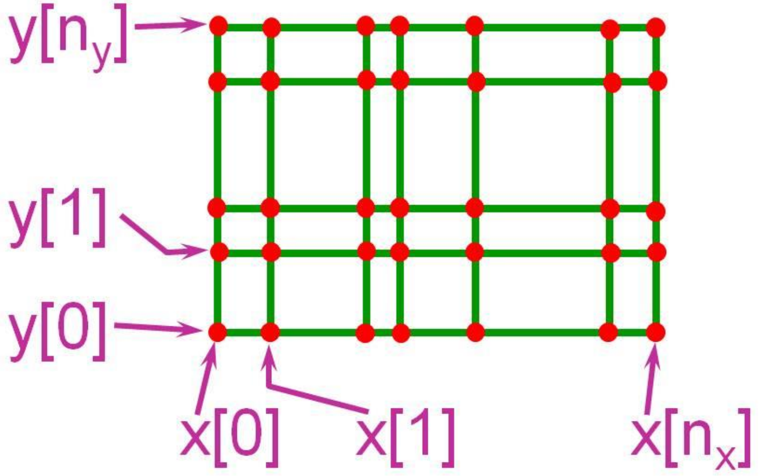
- Some properties of Cartesian grids
  - Assuming N_x and N_y vertices along *x*- and *y*-axis
  - Number of vertices = $N_x \cdot N_y$
  - Number of cells = $(N_x - 1) \cdot (N_y - 1)$
  - Vertex positions are given implicitly from indices [*i*,*j*]:
    - P[*i*,*j*].x = origin + $i \cdot$ dx
    - P[*i*,*j*].y = origin + $j \cdot$ dy
  - It is a structured grid
    - Neighboring information (topology) is given implicitly
    - Neighbors obtained by incrementing/decrementing indices

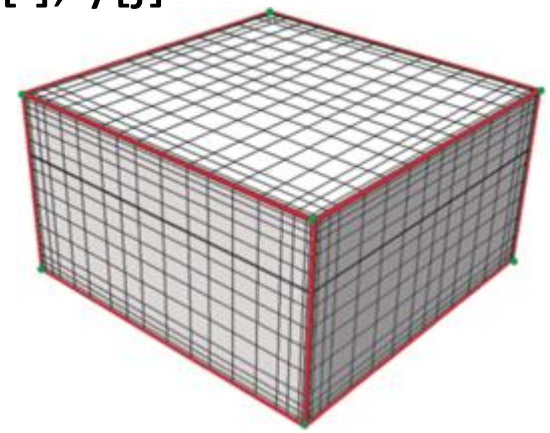
Data representation

- Uniform or Regular Grid
  - Orthogonal, equidistant grid
  - $dx \neq dy$



- Rectilinear Grid
  - Varying sample-distances x[i], y[j]

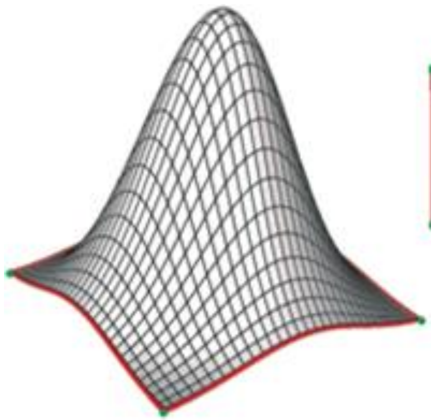


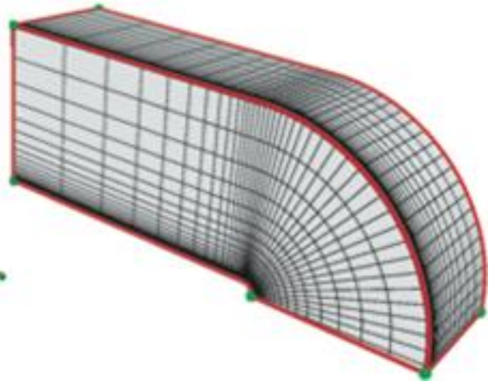


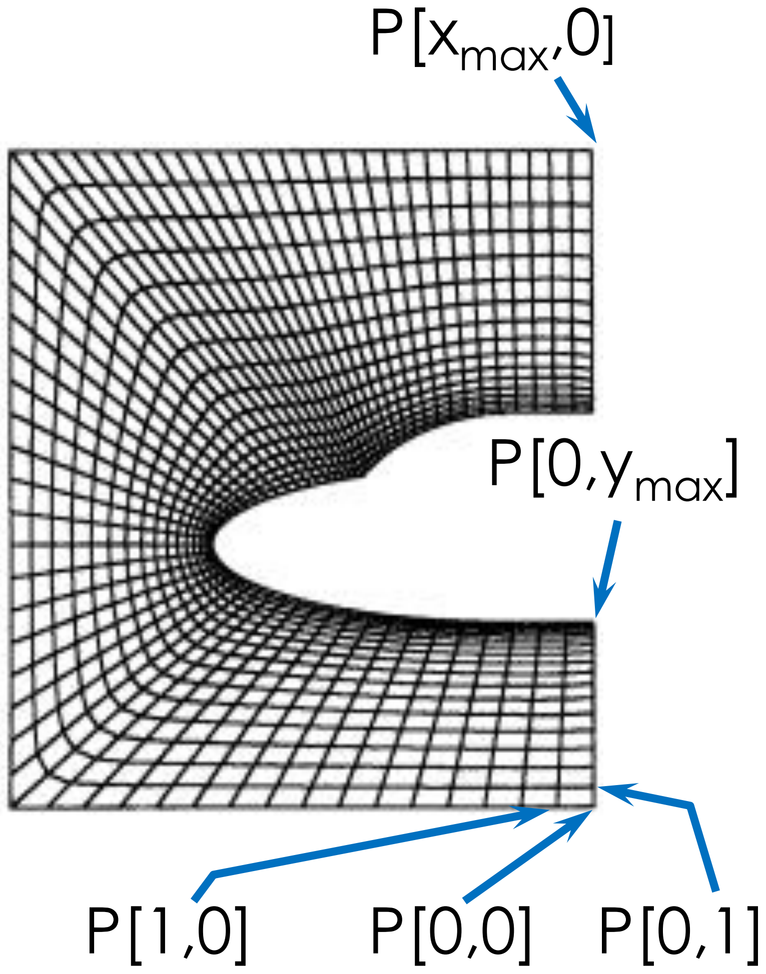
Gröller & Hauser

Data representation

- Curvilinear Grid
  - Non-orthogonal grid
  - Grid-points specified explicitly (P[i,j])
  - Implicit neighborhood relationship

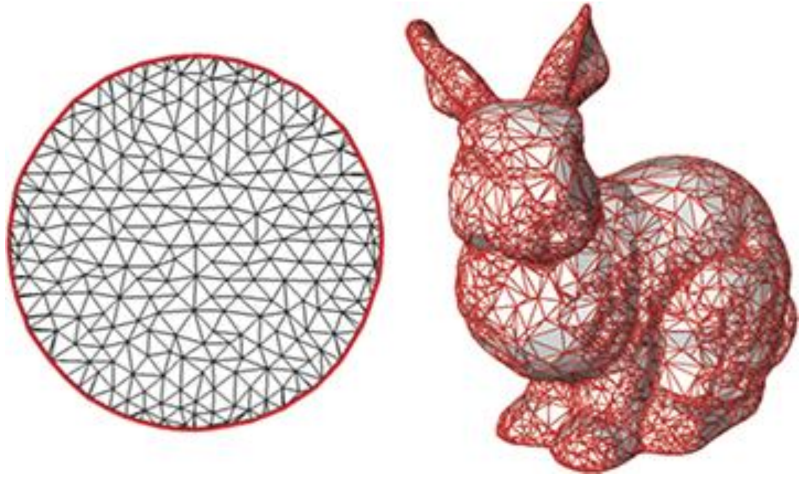


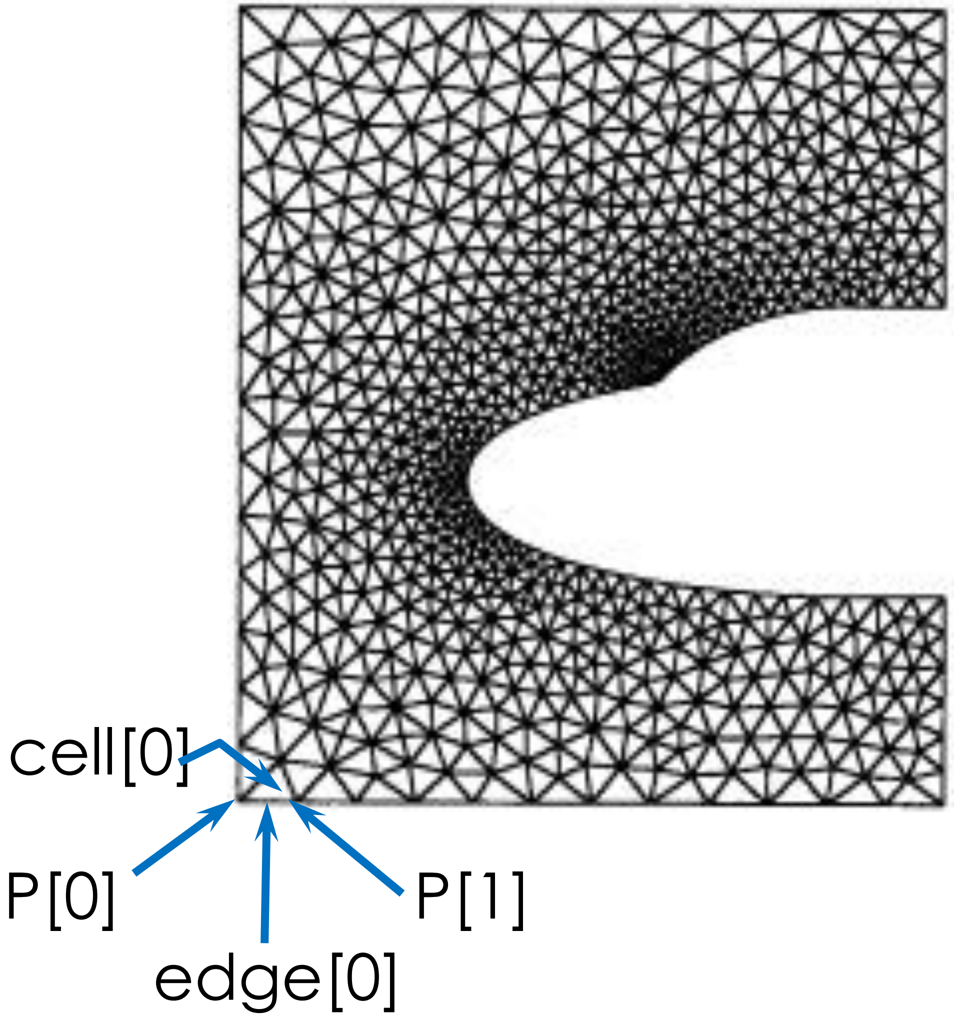




Data representation

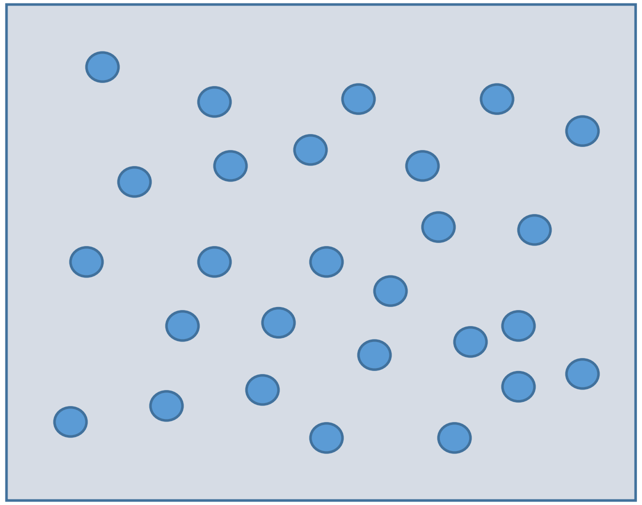
- Unstructured grid
  - Grid points and neighborhood specified explicitly
  - Cells: tetrahedra, hexahedra





Data representation

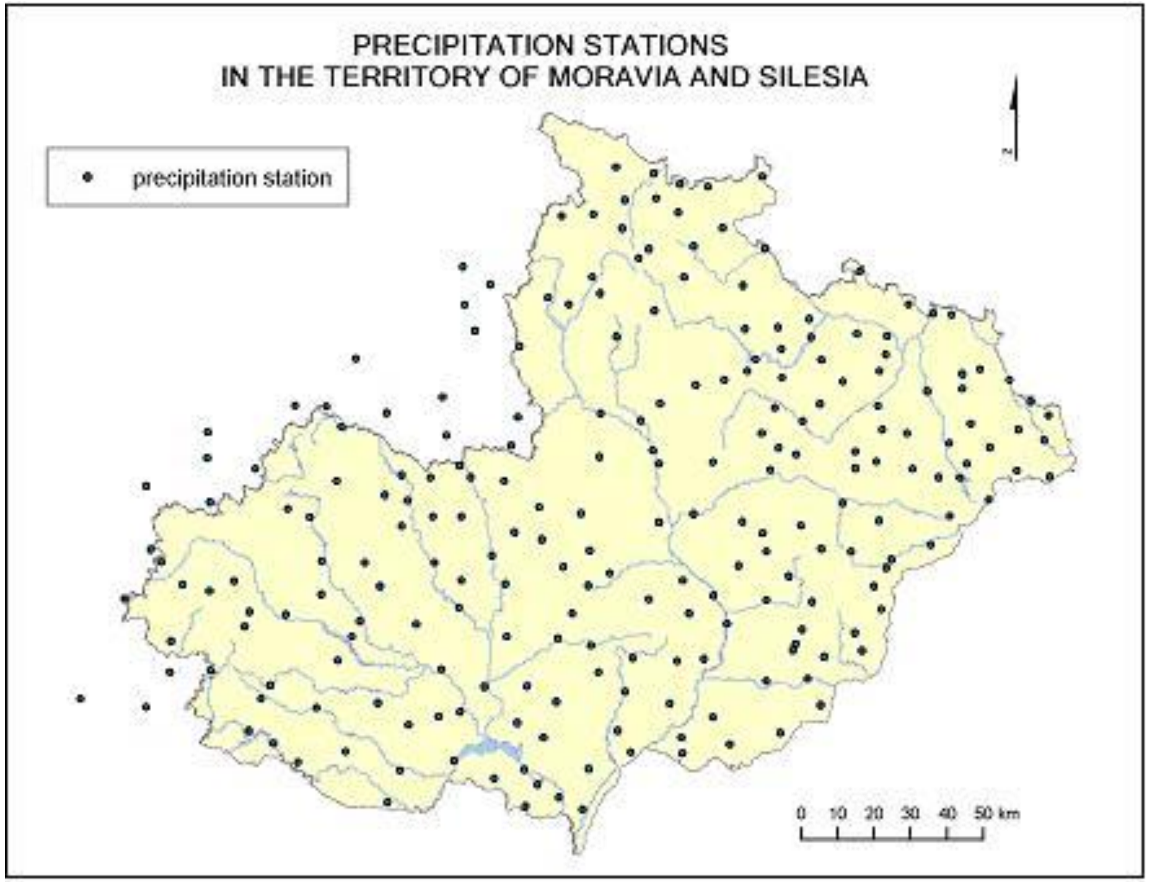
- Scattered Data
  - Grid-free data
  - Data points given without neighborhood-relationship
  - Influence on neighborhood defined by spatial proximity
  - Scattered data interpolation



Data Reconstruction

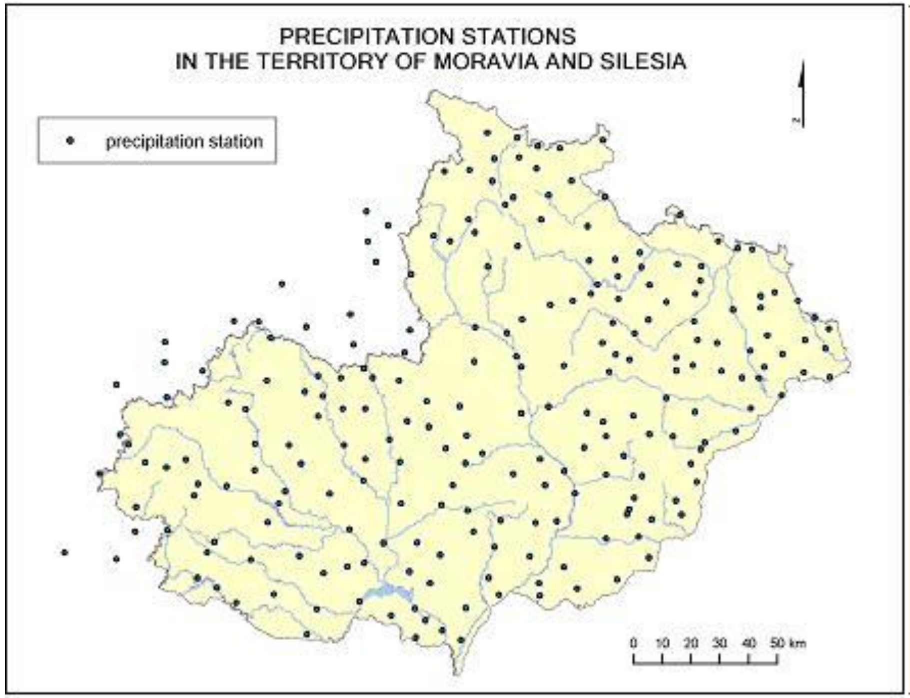
Data reconstruction

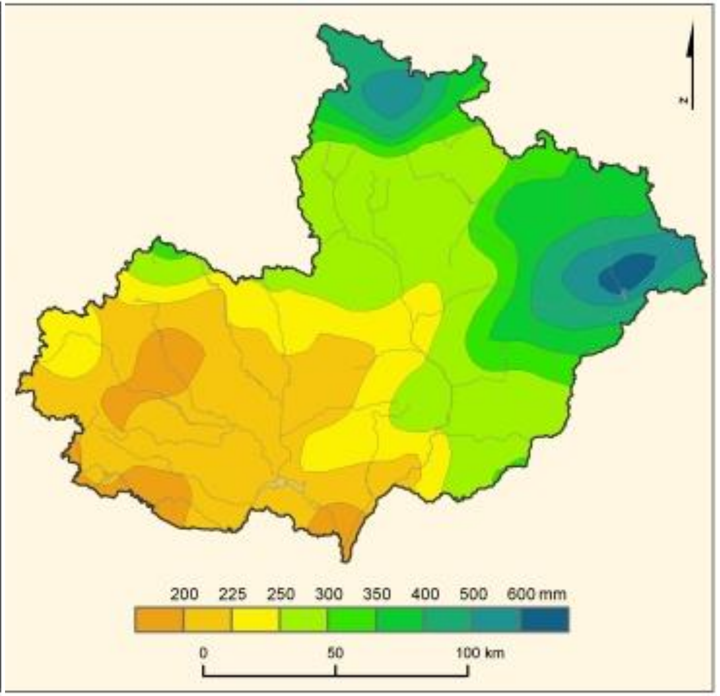
- Initial data often given at a discrete set of scattered points (samples) in the domain



Data reconstruction

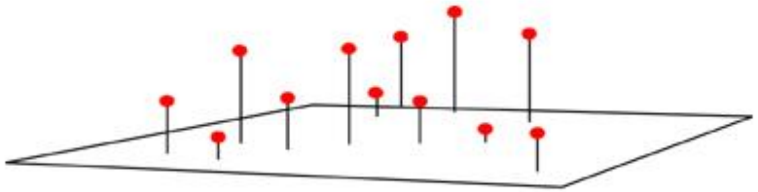
- Derive continuous representation from data given at scattered points
  - Better communication of spatial data distribution



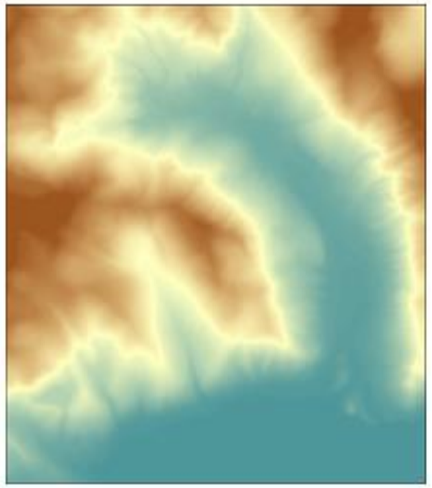


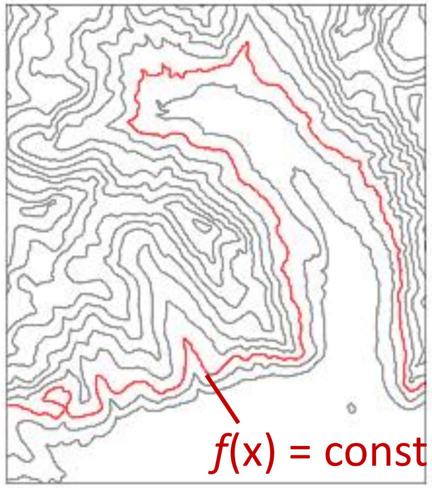
Data reconstruction

- Derive continuous representation from data given at scattered points
  - Some analysis techniques require a continuous representation



Data distribution

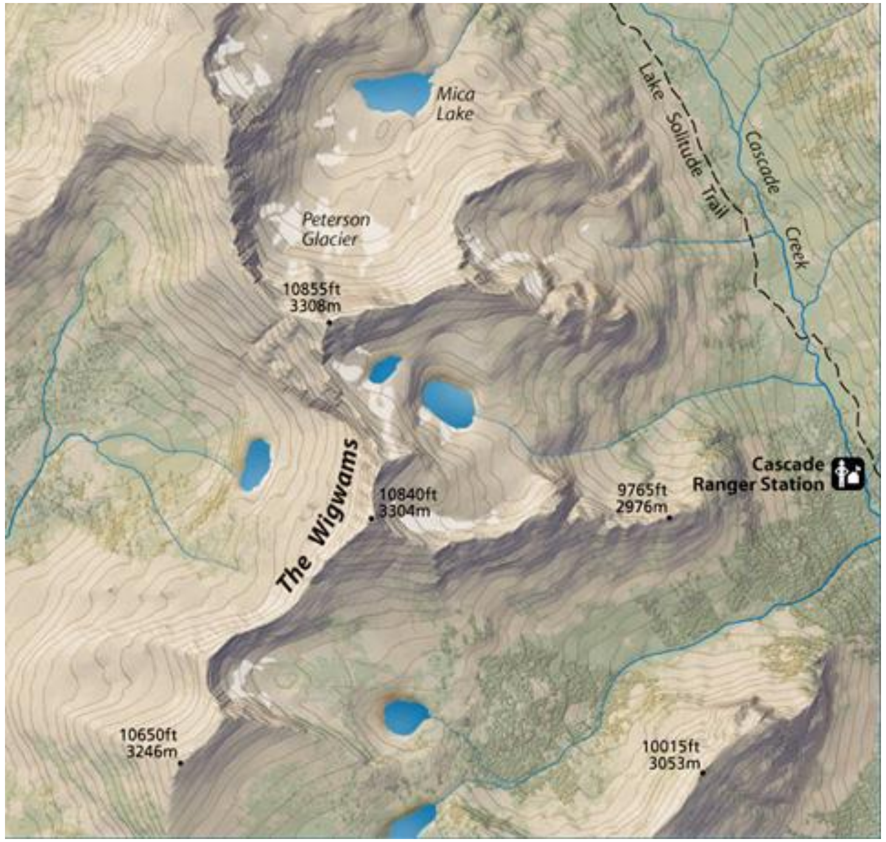


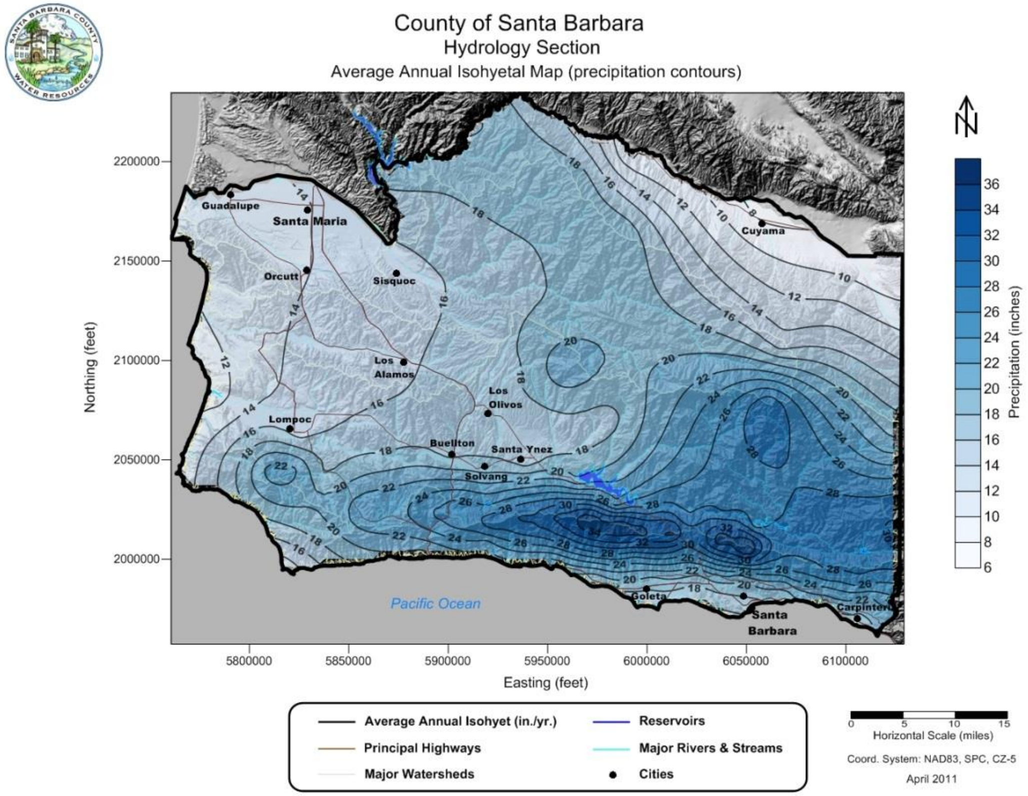


Isocontours – curves on which all points with a given value lie

Data reconstruction

- Isocontours in continuous data fields

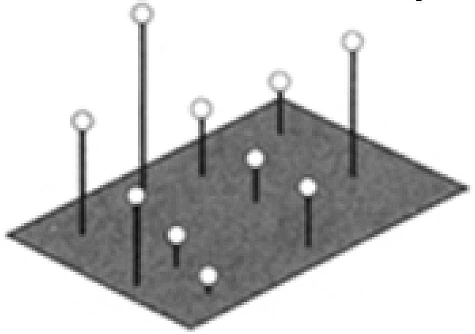




© NPS 2000

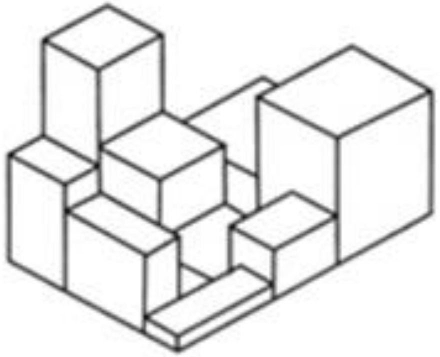
Data reconstruction

- Data reconstruction from scattered points

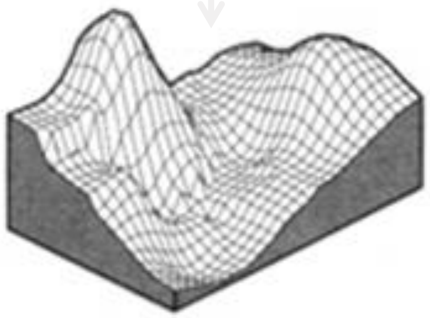


Assumes some similarity between data values inversely proportional to distance

data interpolation



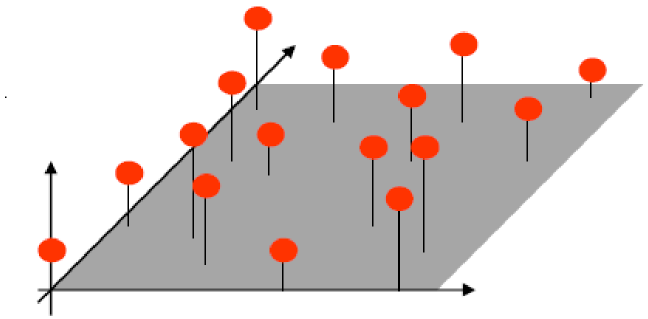
piece-wise constant interpolation



continuous interpolation

Data reconstruction

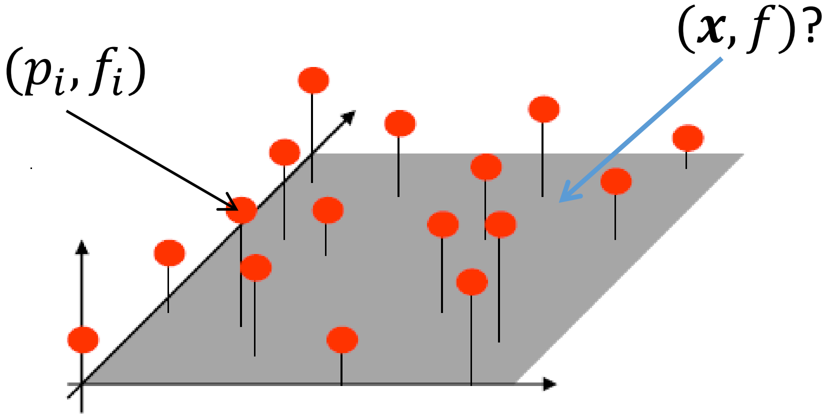
- Data reconstruction from scattered points



- Obtain values in-between the data points…
  1. from a continuous function which interpolates the given values and varies smoothly in-between
  2. from a grid which is constructed from the given points, i.e., a triangulation

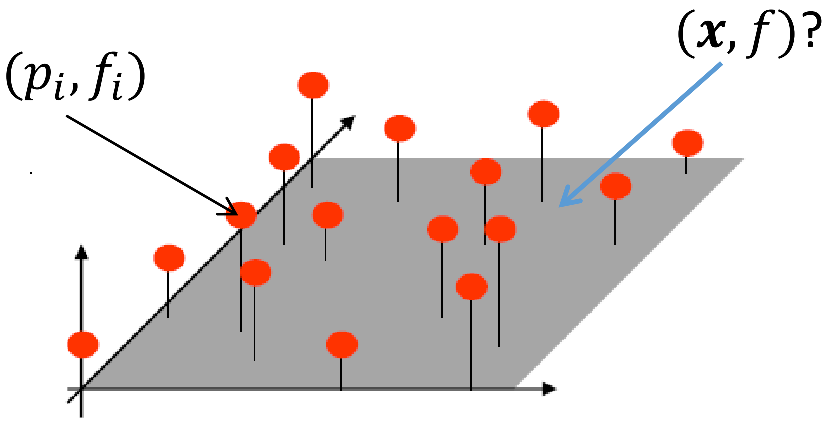
Continuous representation

- Given a set of scattered points p_i in a 2D parameter domain with scalar values f_i
  - The principles are applicable to arbitrary parameter domain dimensions (1D/2D/3D)
- **Goal:** Construct a continuous function f from given set of p_i, f_i which approximates ("follows") the given values

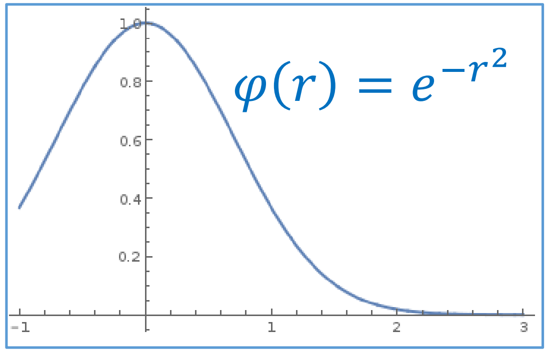


Continuous representation

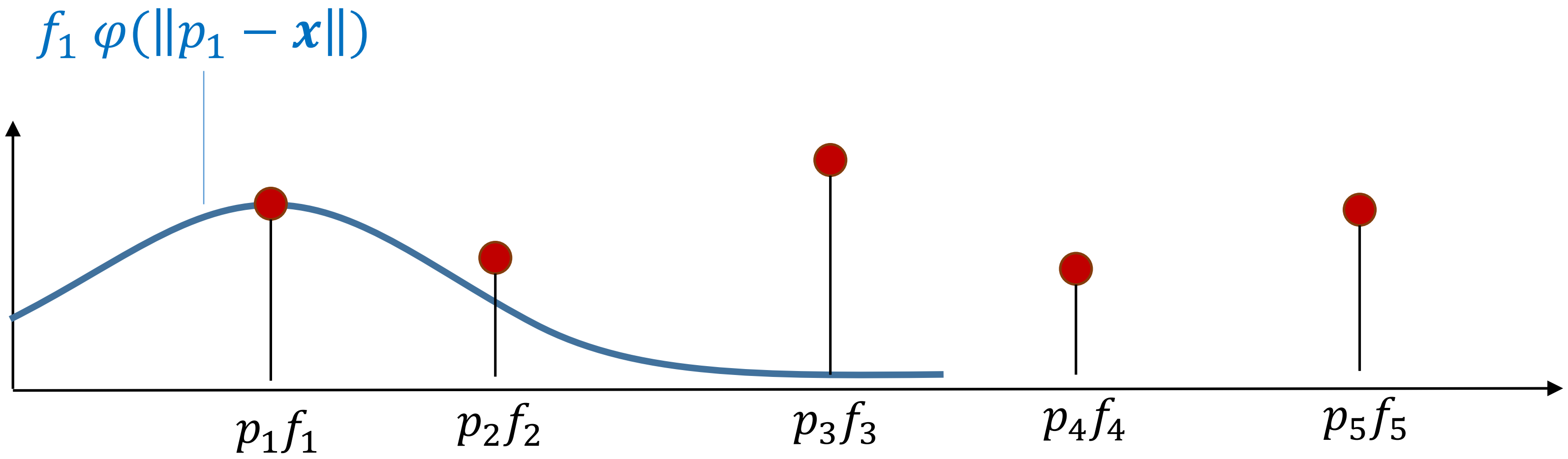
- Radial Basis Functions
  - Independent of dimension of parameter domain (1D/2D/3D)
  - Each (p_i, f_i) influences $f(\boldsymbol{x})$ based on Euclidean distance $r = ||p_i - \boldsymbol{x}||$
  - Nearby points have higher influence than far-away points



Continuous representation

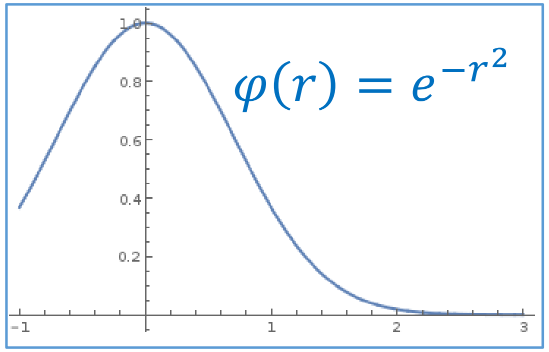


- Radial Basis Functions
  - Each radial function $\varphi(r)$ is centered around a data point p_i
  - Value of $\varphi(r)$ decreases quickly with increasing distance $r = \|p_i - \boldsymbol{x}\|$ to the function's center p_i

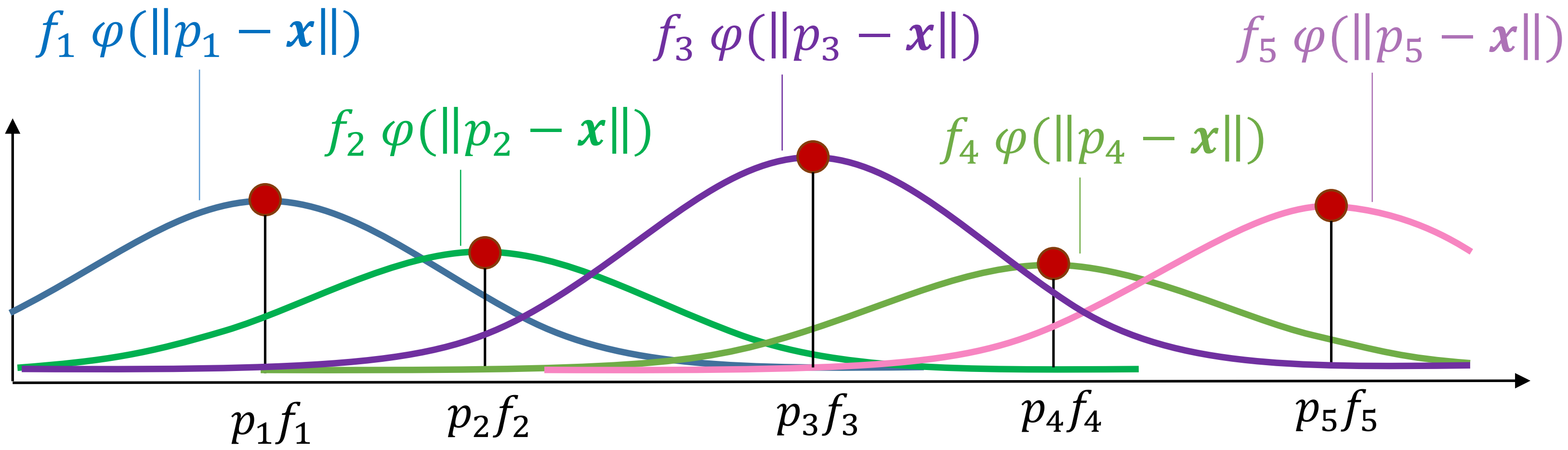


multivis.net/lecture/radial-basis.html

Continuous representation

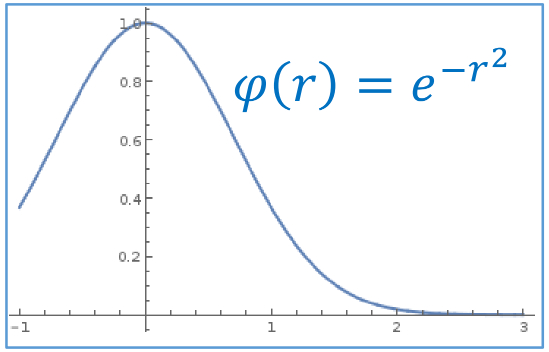


- Radial Basis Functions
  - Each radial function $\varphi(r)$ is centered around a data point p_i
  - Value of $\varphi(r)$ decreases quickly with increasing distance $r = \|p_i - \boldsymbol{x}\|$ to the function's center p_i



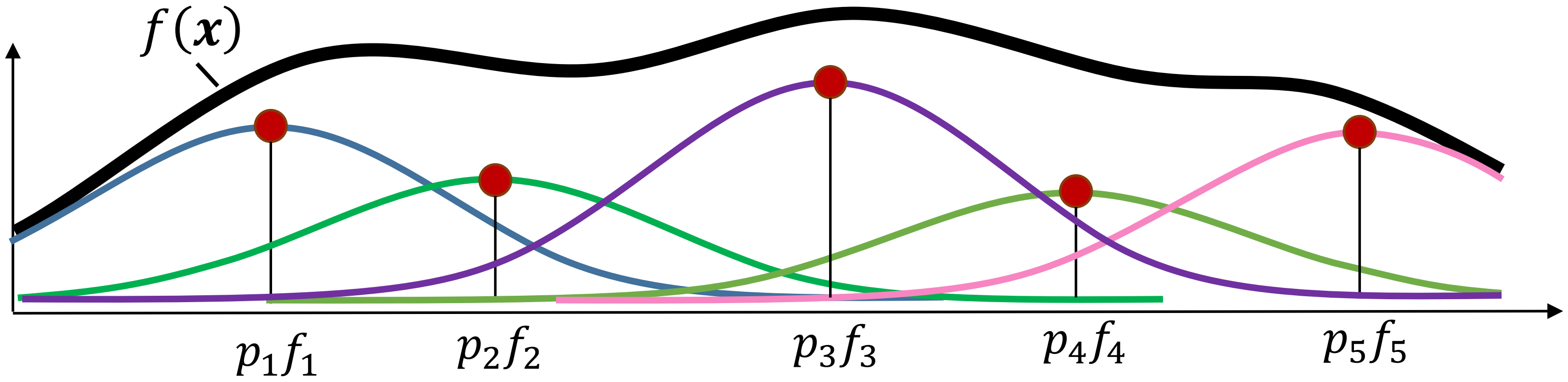
multivis.net/lecture/radial-basis.html

Continuous representation



- Radial Basis Functions
  - Each radial function $\varphi(r)$ is centered around a data point p_i
  - Value of $\varphi(r)$ decreases quickly with increasing distance $r = \|p_i - \boldsymbol{x}\|$ to the function's center p_i
  - Function *f* represented as weighted sum of *N* radial functions φ

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i \, \varphi(\|p_i - \boldsymbol{x}\|)$$

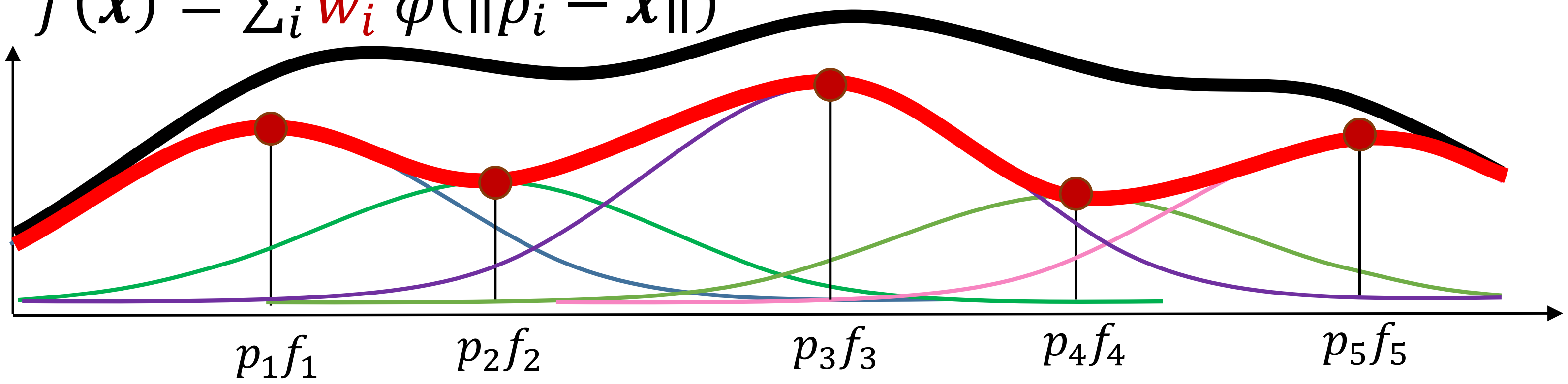


multivis.net/lecture/radial-basis.html

Continuous representation

- Radial Basis Functions
  - Instead of the **black** curve we want the **red** one, i.e., a curve which is going through the initial data points
  - This is called an interpolation
  - **Question:** How do we have to select the weights w_i so that the red curve is obtained?

$$f(\boldsymbol{x}) = \sum_i w_i \, \varphi(\|p_i - \boldsymbol{x}\|)$$



multivis.net/lecture/radial-basis.html

Continuous representation

- Radial Basis Functions – finding the weights w_i
  - For $j = 1, \dots, N$,
    specify w_i such that $f(p_j)$ interpolates the value f_j

$$f(p_j) = \sum_{i=1}^{N} w_i \varphi(\|p_i - p_j\|) = f_j$$

Continuous representation

- Example:
  - Data points: $p_1 = 1$, $p_2 = 3$, $p_3 = 4$
  - Data values: $f_1 = 1$, $f_2 = 2$, $f_3 = 0$
- Find the weights w_i such that f interpolates all points

$$f(p_j) = \sum_{i=1}^{N} w_i \varphi(\|p_i - p_j\|) = f_j$$

Continuous representation

$$f(1) = 1$$
$$f(3) = 2$$
$$f(4) = 0$$

- This results in:

$$f(x) = w_1\varphi(|1-x|) + w_2\varphi(|3-x|) + w_3\varphi(|4-x|)$$

- Yielding the equation system:

$$w_1\varphi(0) + w_2\varphi(2) + w_3\varphi(3) = 1$$
$$w_1\varphi(2) + w_2\varphi(0) + w_3\varphi(1) = 2$$
$$w_1\varphi(3) + w_2\varphi(1) + w_3\varphi(0) = 0$$

Continuous representation

$$w_1\varphi(0) + w_2\varphi(2) + w_1\varphi(3) = 1$$
$$w_1\varphi(-2) + w_2\varphi(0) + w_1\varphi(1) = 2$$
$$w_1\varphi(-3) + w_2\varphi(-1) + w_1\varphi(0) = 0$$

- We can formulate this in matrix representation:

$$\begin{pmatrix} \varphi(0) & \varphi(2) & \varphi(3) \\ \varphi(2) & \varphi(0) & \varphi(1) \\ \varphi(3) & \varphi(1) & \varphi(0) \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

Continuous representation

- Radial Basis Functions – finding the weights w_i
  - For $j = 1, \dots, N$,
    specify w_i such that $f(p_j)$ interpolates the value f_j
    $$f(p_j) = \sum_{i=1}^{N} w_i \varphi(\|p_i - p_j\|) = f_j$$
  - Yields a system of linear equations (per point) to be solved for w_i

$$\begin{bmatrix} \varphi(\|p_1 - p_1\|) & \varphi(\|p_2 - p_1\|) & \cdots & \varphi(\|p_N - p_1\|) \\ \varphi(\|p_1 - p_2\|) & \varphi(\|p_2 - p_2\|) & \cdots & \varphi(\|p_N - p_2\|) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|p_1 - p_N\|) & \varphi(\|p_2 - p_N\|) & \cdots & \varphi(\|p_N - p_N\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}$$

N equations in N unkowns

Continuous representation

- Radial Basis Functions – finding the weights w_i
  - For $j = 1, \dots, N$,
    specify w_i such that $f(p_j)$ interpolates the value f_j

$$f(p_j) = \sum_{i=1}^{N} w_i \varphi(\|p_i - p_j\|) = f_j$$

  - Yields a system of linear equations (per point) to be solved for w_i

$$\begin{bmatrix} \varphi(0) & \varphi(\|p_2 - p_1\|) & \cdots & \varphi(\|p_N - p_1\|) \\ \varphi(\|p_1 - p_2\|) & \varphi(0) & \cdots & \varphi(\|p_N - p_2\|) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|p_1 - p_N\|) & \varphi(\|p_2 - p_N\|) & \cdots & \varphi(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}$$

Continuous representation

$$w_1\varphi(0) + w_2\varphi(2) + w_1\varphi(3) = 1$$
$$w_1\varphi(-2) + w_2\varphi(0) + w_1\varphi(1) = 2$$
$$w_1\varphi(-3) + w_2\varphi(-1) + w_1\varphi(0) = 0$$

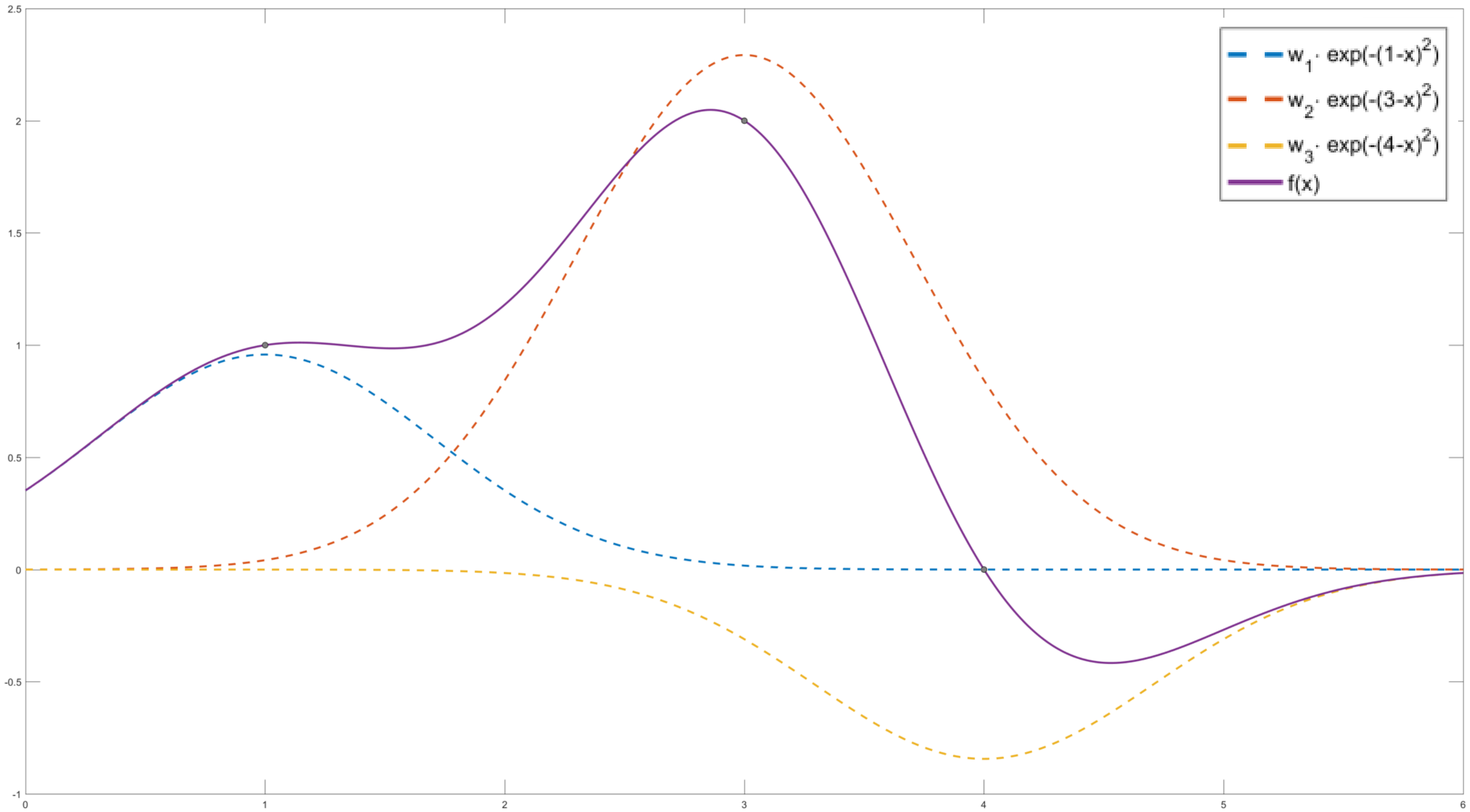
- Back to our example

$$\underbrace{\begin{pmatrix} \varphi(0) & \varphi(2) & \varphi(3) \\ \varphi(2) & \varphi(0) & \varphi(1) \\ \varphi(3) & \varphi(1) & \varphi(0) \end{pmatrix}}_{\boldsymbol{R}} \cdot \underbrace{\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}}_{\boldsymbol{w}} = \underbrace{\begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}}_{\boldsymbol{f}}$$

- If matrix $\boldsymbol{R}$ is invertible,
- $\boldsymbol{w}$=$\boldsymbol{R}$^(−1) $\boldsymbol{f}$ with solution

Continuous representation

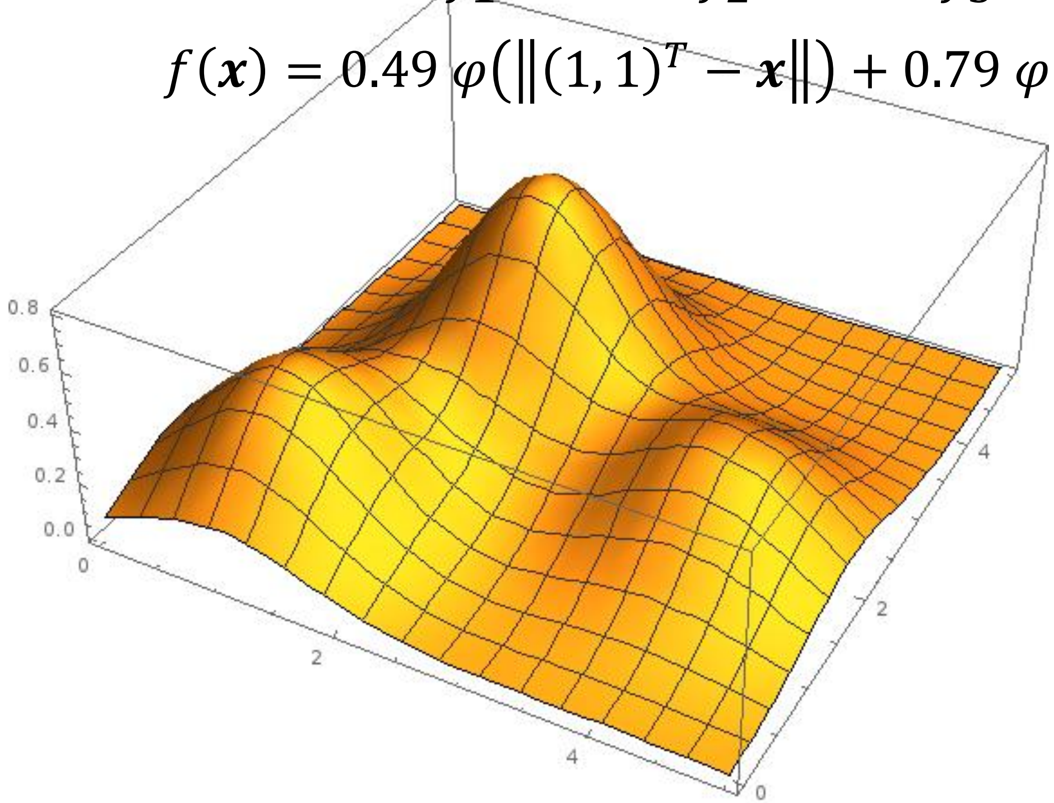
- Example $f(x) = 0.9581\varphi(\|1-x\|) + 2.2928\,\varphi(\|3-x\|) - 0.8436\,\varphi(\|4-x\|)$

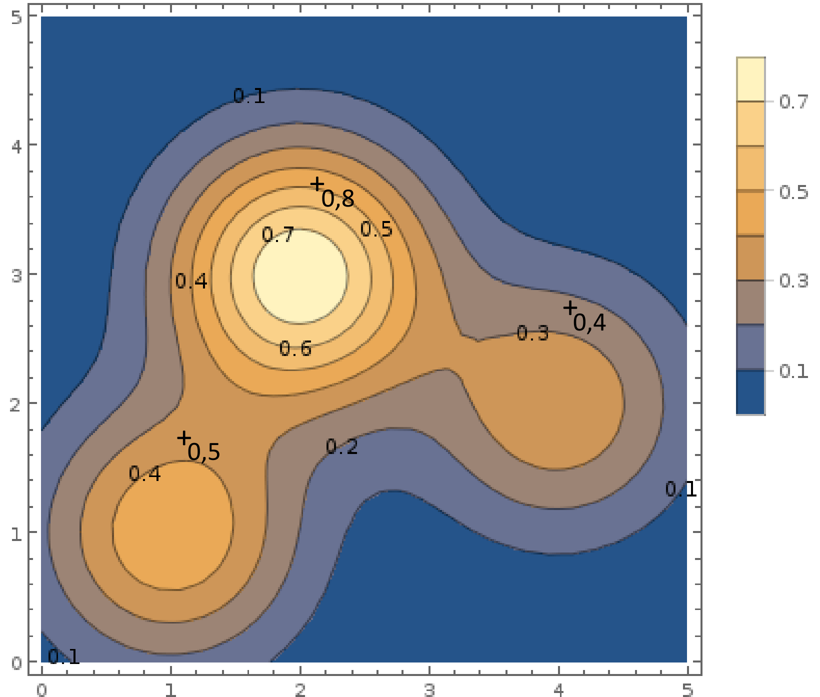


Continuous representation

- 2D Example
  - Data points: $p_1 = (1,1)^T$, $p_2 = (2,3)^T$, $p_3 = (4,2)^T$
  - Data values: $f_1 = 0.5$, $f_2 = 0.8$, $f_3 = 0.4$

$$f(\boldsymbol{x}) = 0.49\,\varphi(\|(1,1)^T - \boldsymbol{x}\|) + 0.79\,\varphi(\|(2,3)^T - \boldsymbol{x}\|) + 0.39\,\varphi(\|(4,2)^T - \boldsymbol{x}\|)$$





Continuous representation

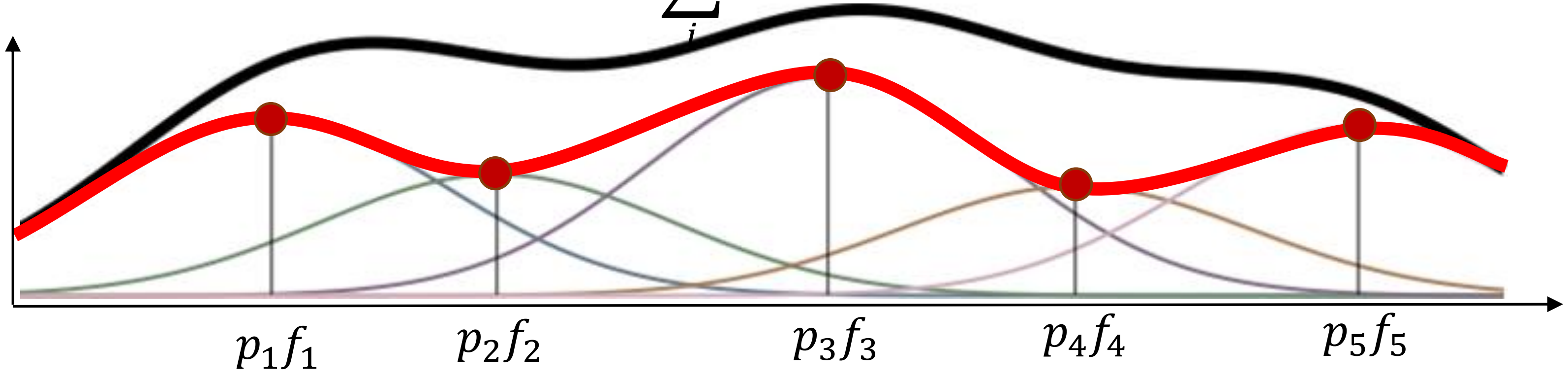
- Drawbacks of radial basis functions
  - Every sample point has influence on whole domain
  - Adding a new sample requires re-solving the equation system
  - Computationally expensive (solving a system of linear equations)

- What can we do?
  - Find a different radial function
  - Give up finding a smooth reconstruction
    - Try finding a piecewise (local) reconstruction function

Continuous representation

- Radial Basis Functions
  - Instead of the **black** curve we want the **red** one, i.e., the curve which is going through the initial data points
  - This is called an interpolation
  - **Question:** How do we have to select the radial function φ so that the red curve is obtained?

$$f(\boldsymbol{x}) = \sum_i f_i \varphi(\|p_i - \boldsymbol{x}\|)$$



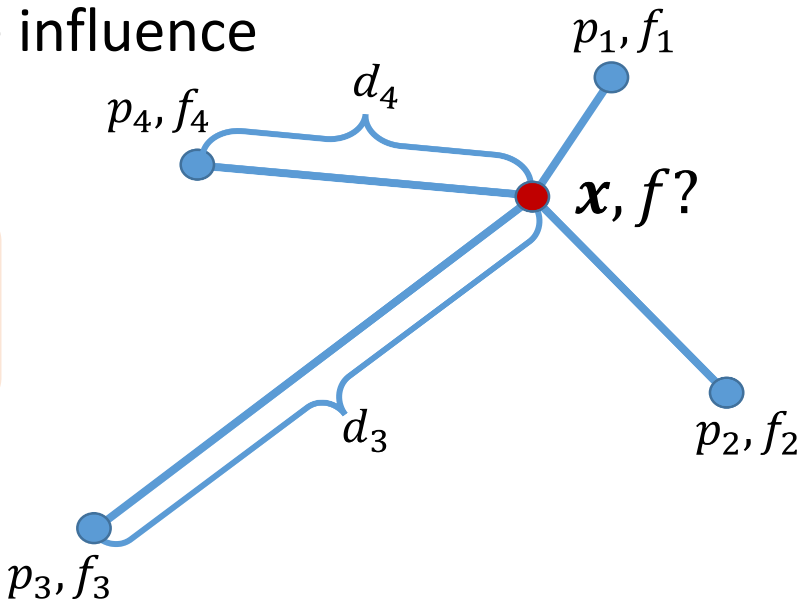
Continuous representation

- Inverse distance weighting
  - Sample positions p_i and values f_i
  - Assumption: Nearby points are more similar than those further away → they have more influence

$$f(\boldsymbol{x}) = \sum_i f_i \, \varphi(\|p_i - \boldsymbol{x}\|)$$

$$d_i = \|p_i - \boldsymbol{r}\|, \qquad \varphi(r) = \frac{1}{r^2} \Big/ \sum_{i=1}^{N} \frac{1}{d_i^2}$$

Attention for $d_i < \epsilon$



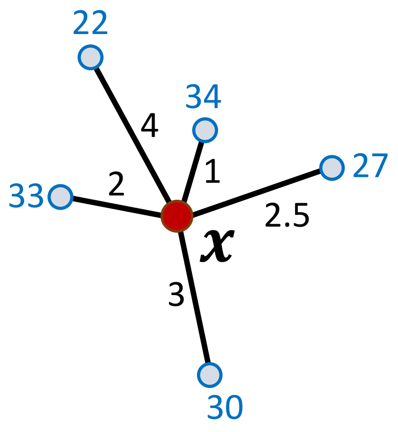
multivis.net/lecture/radial-basis.html

Continuous representation

- Inverse distance weighting
  - Sample positions p_i and values f_i
  - Assumption: Nearby points are more similar than those further away → they have more influence

$$f(\boldsymbol{x}) = \sum_i f_i\, \varphi(\|p_i - \boldsymbol{x}\|) =$$

$$= \sum_{i=1}^{N} \frac{f_i}{\|p_i - \boldsymbol{x}\|^2} \Big/ \sum_{i=1}^{N} \frac{1}{\|p_i - \boldsymbol{x}\|^2}$$

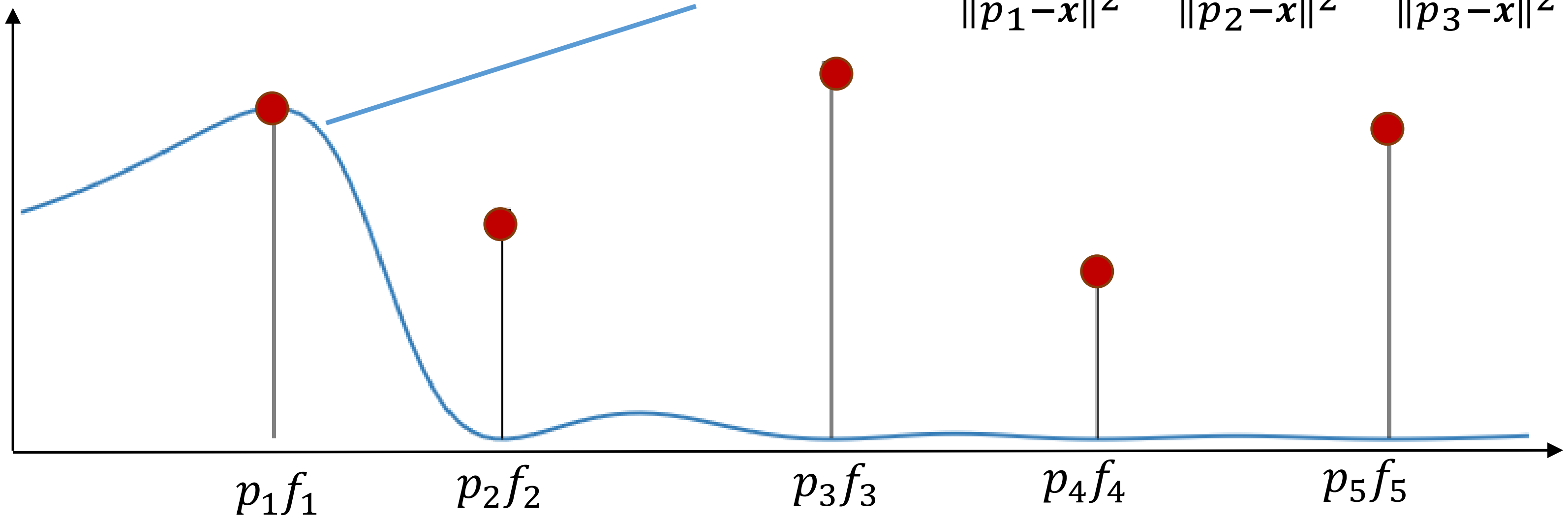


$$f(\boldsymbol{x}) = \frac{\frac{22}{4^2} + \frac{34}{1^2} + \frac{27}{2.5^2} + \frac{30}{3^2} + \frac{33}{2^2}}{\frac{1}{4^2} + \frac{1}{1^2} + \frac{1}{2.5^2} + \frac{1}{3^2} + \frac{1}{2^2}} = 32.38$$

Continuous representation

- Inverse distance weighting
  - Sample positions p_i and values f_i
  - Assumption: Nearby points are more similar than those further away

$$f_1\,\varphi(\|p_1-\boldsymbol{x}\|)=\frac{\frac{f_1}{\|p_1-\boldsymbol{x}\|^2}}{\frac{1}{\|p_1-\boldsymbol{x}\|^2}+\frac{1}{\|p_2-\boldsymbol{x}\|^2}+\frac{1}{\|p_3-\boldsymbol{x}\|^2}+\frac{1}{\|p_4-\boldsymbol{x}\|^2}+\frac{1}{\|p_5-\boldsymbol{x}\|^2}}$$

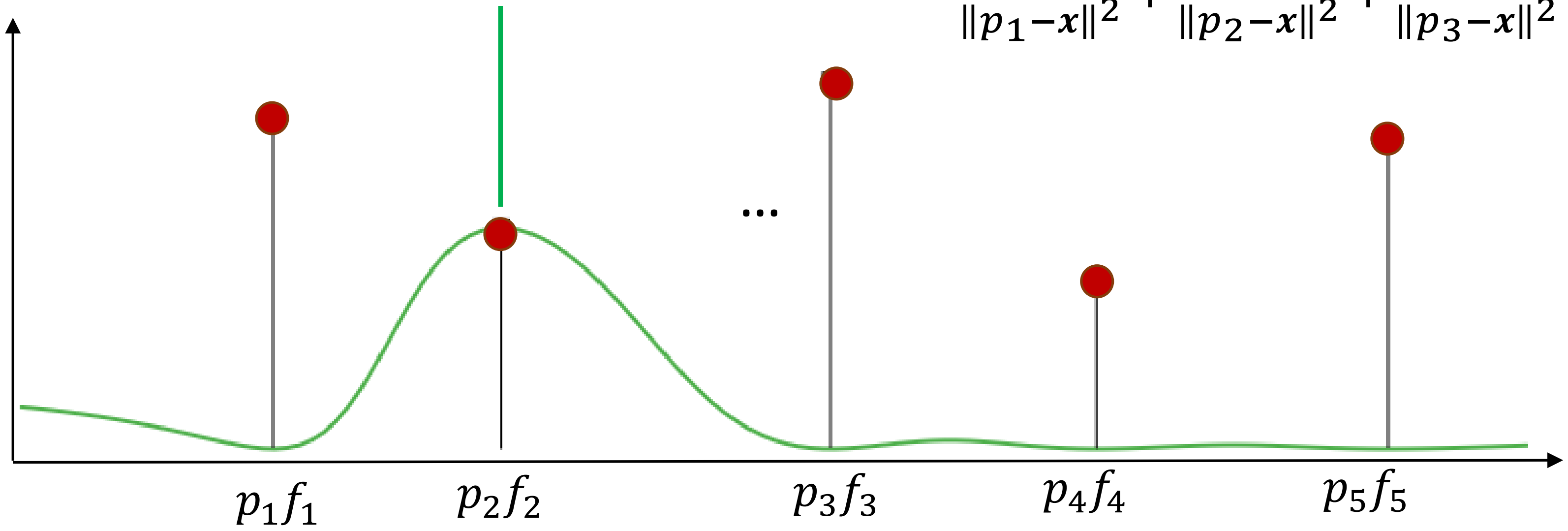


function is zero at p_i, $i \neq 1$

Continuous representation

- Inverse distance weighting
  - Sample positions p_i and values f_i
  - Assumption: Nearby points are more similar than those further away

$$f_2\,\varphi(\|p_2 - \boldsymbol{x}\|) = \frac{\frac{f_2}{\|p_2-\boldsymbol{x}\|^2}}{\frac{1}{\|p_1-\boldsymbol{x}\|^2}+\frac{1}{\|p_2-\boldsymbol{x}\|^2}+\frac{1}{\|p_3-\boldsymbol{x}\|^2}+\frac{1}{\|p_4-\boldsymbol{x}\|^2}+\frac{1}{\|p_5-\boldsymbol{x}\|^2}}$$

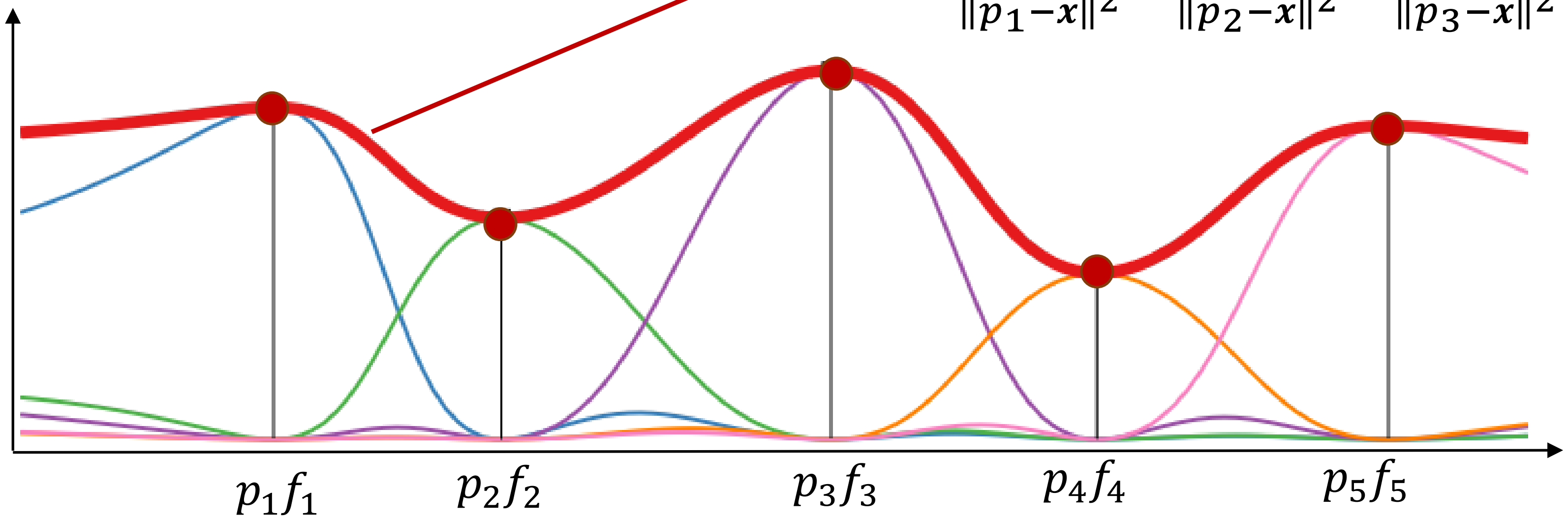


function is zero at $p_i, \quad i \neq 2$

Continuous representation

- Inverse distance weighting
  - Sample positions p_i and values f_i
  - Assumption: Nearby points are more similar than those further away

$$f(\boldsymbol{x}) = \frac{\frac{f_1}{\|p_1-\boldsymbol{x}\|^2} + \frac{f_2}{\|p_2-\boldsymbol{x}\|^2} + \frac{f_3}{\|p_3-\boldsymbol{x}\|^2} + \frac{f_4}{\|p_4-\boldsymbol{x}\|^2} + \frac{f_5}{\|p_5-\boldsymbol{x}\|^2}}{\frac{1}{\|p_1-\boldsymbol{x}\|^2} + \frac{1}{\|p_2-\boldsymbol{x}\|^2} + \frac{1}{\|p_3-\boldsymbol{x}\|^2} + \frac{1}{\|p_4-\boldsymbol{x}\|^2} + \frac{1}{\|p_5-\boldsymbol{x}\|^2}}$$



Continuous representation

- Inverse distance weighting
  - We no longer have to solve a system of linear equations to find the weights
  - However, every sample point still has global influence

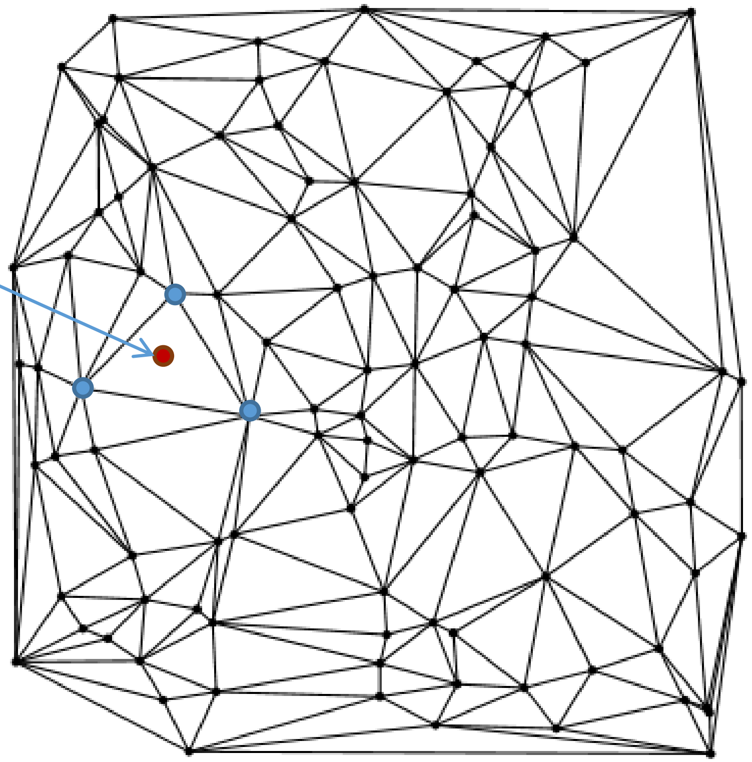
- What can we do?
  - Give up smooth reconstruction by constructing a grid from the given points, i.e., a triangulation

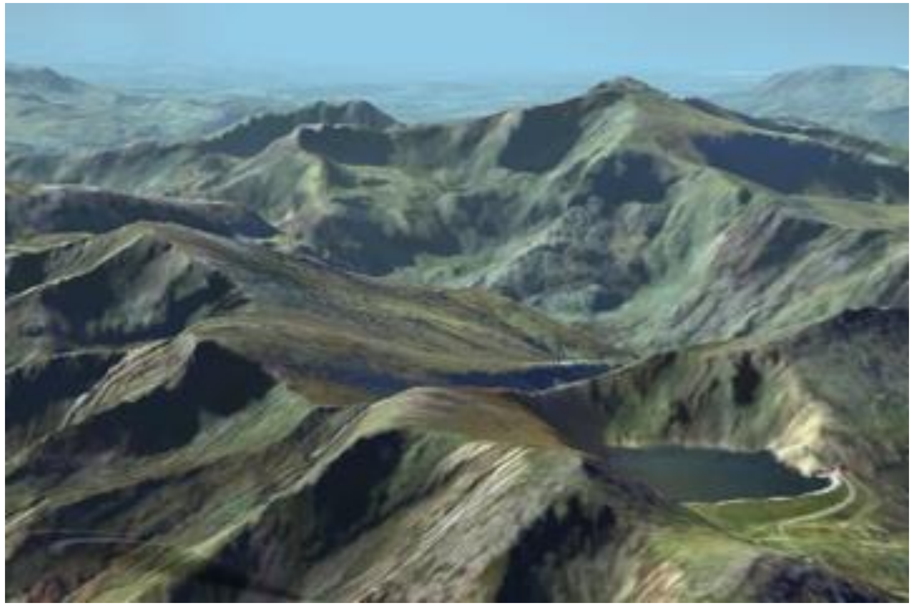
Triangulation

Generating a triangulation

- Try finding a piecewise (local) reconstruction function
  - Connect the points so that a triangulation is obtained
  - Interpolate locally within the triangles

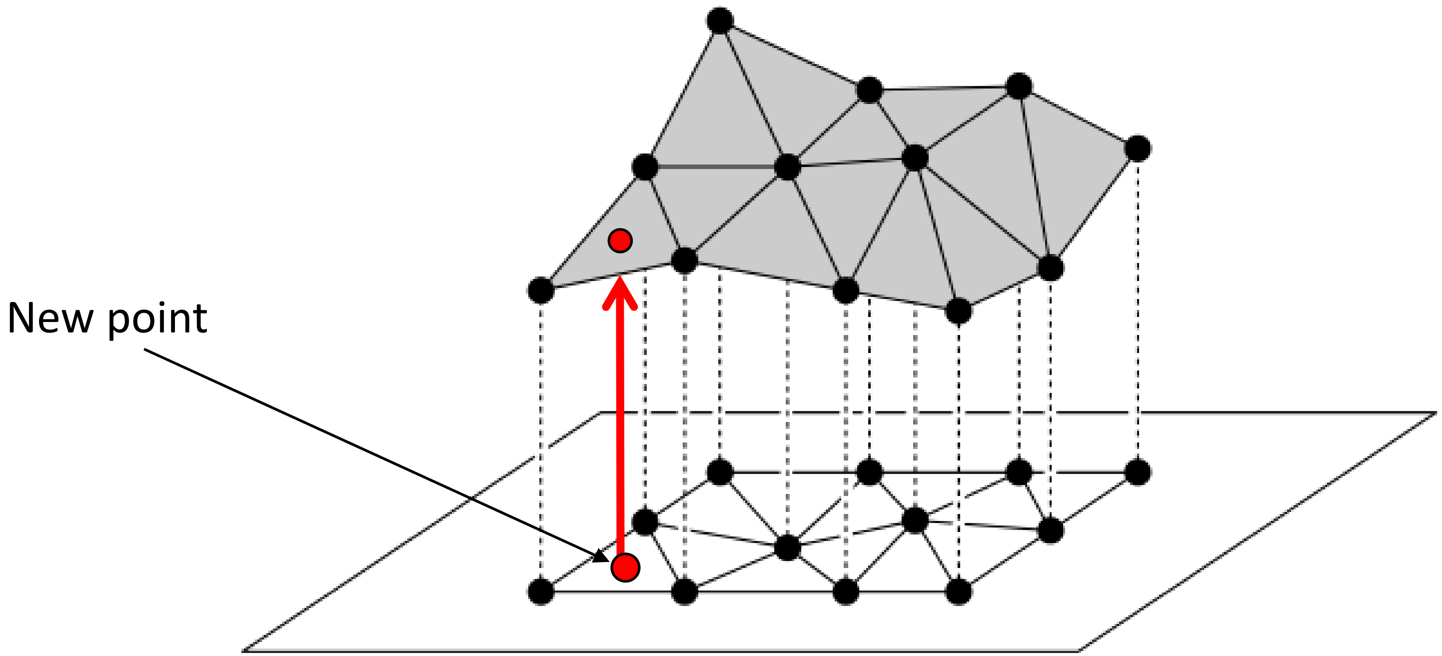
Value obtained by only considering values at triangle corners





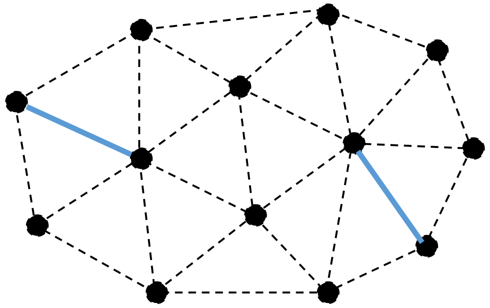
Generating a triangulation

- Once a triangulation is given
  - Let the scalar values at vertices be interpolated across the triangles
  - I.e., piecewise linear interpolation of values at interior points



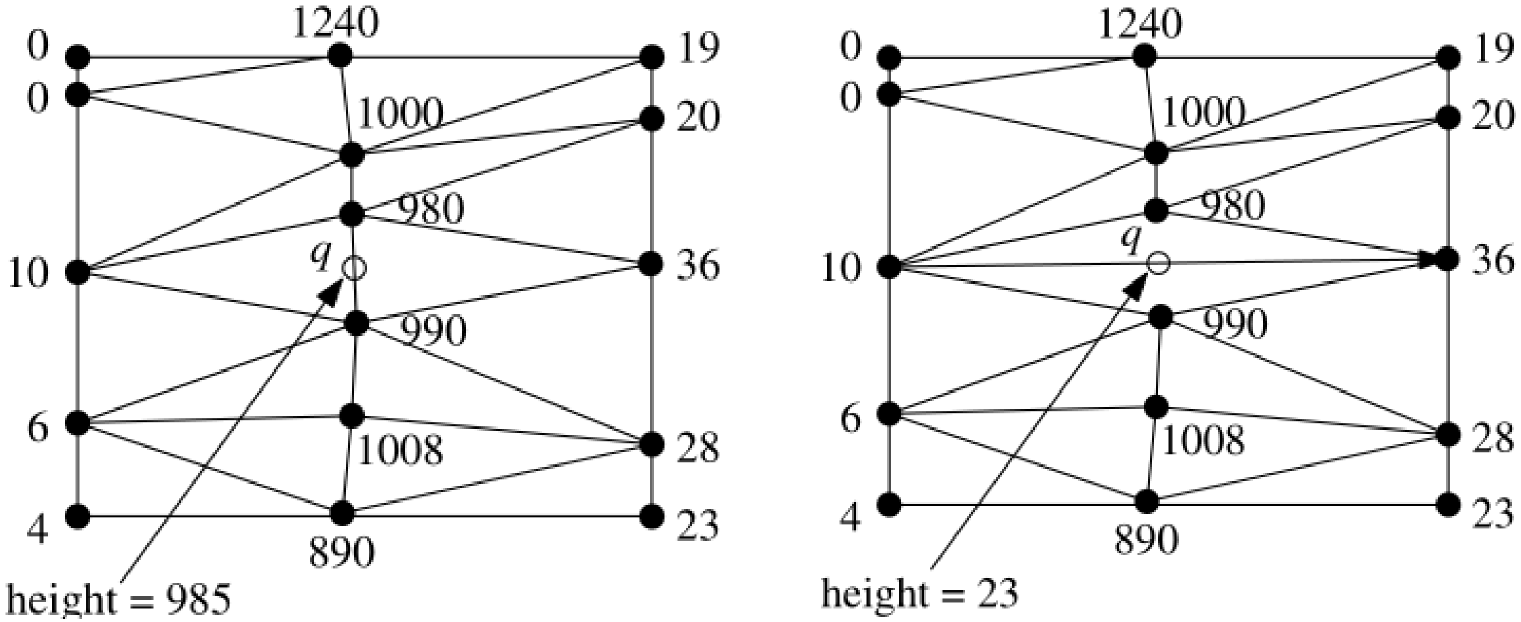
Generating a triangulation

- Given irregularly distributed positions without connectivity information
- For a set of points many triangulations exist
- The challenge is to find the connectivity so that a “good” triangulation is generated



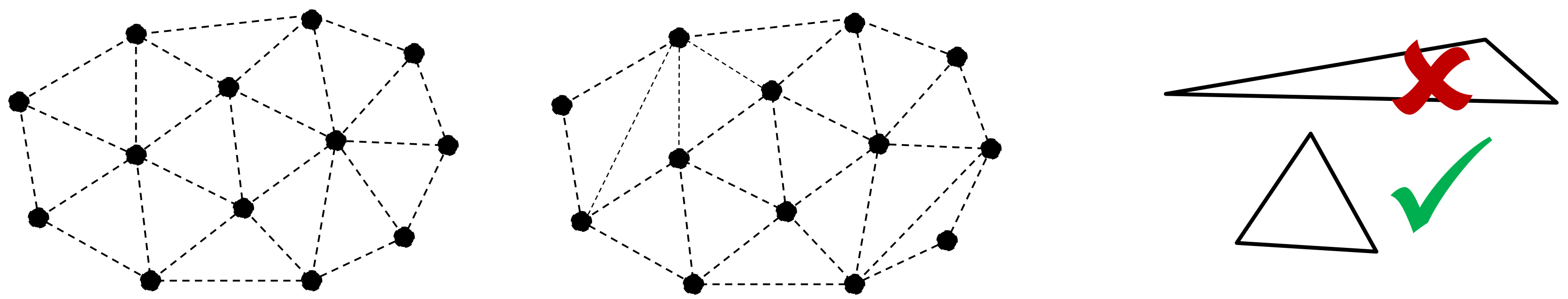
Generating a triangulation

- What is a good triangulation?



Generating a triangulation

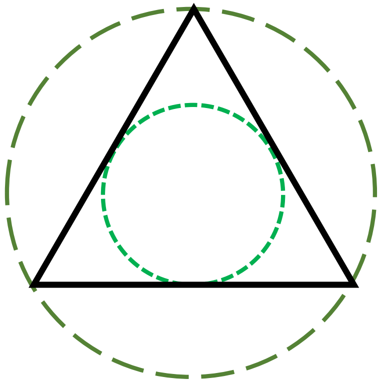
- What is a good triangulation?
  - A measure for the quality of a triangulation is the aspect ratio of the so-defined triangles
  - Avoid long, thin triangles
  - Make triangles as equilateral as possible

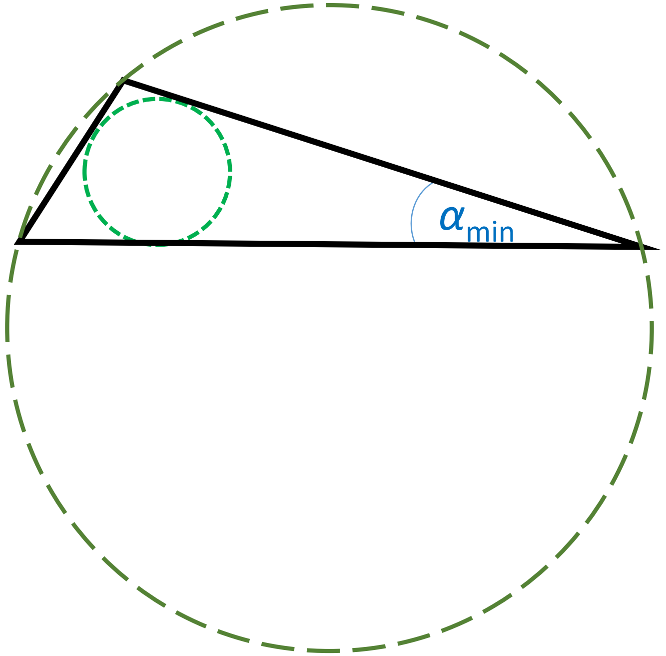


Generating a triangulation

- An “optimal” triangulation
  - Makes triangles as equilateral as possible
  - Maximizes the minimum angle in the triangulation
  - Maximizes $\frac{\text{radius of in}-\text{circle}}{\text{radius of circumcircle}}$

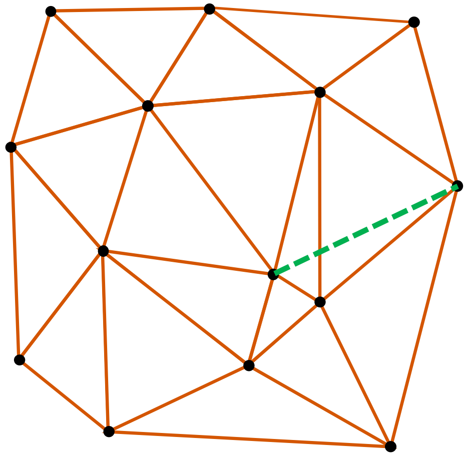
- A Delaunay triangulation is an optimal triangulation

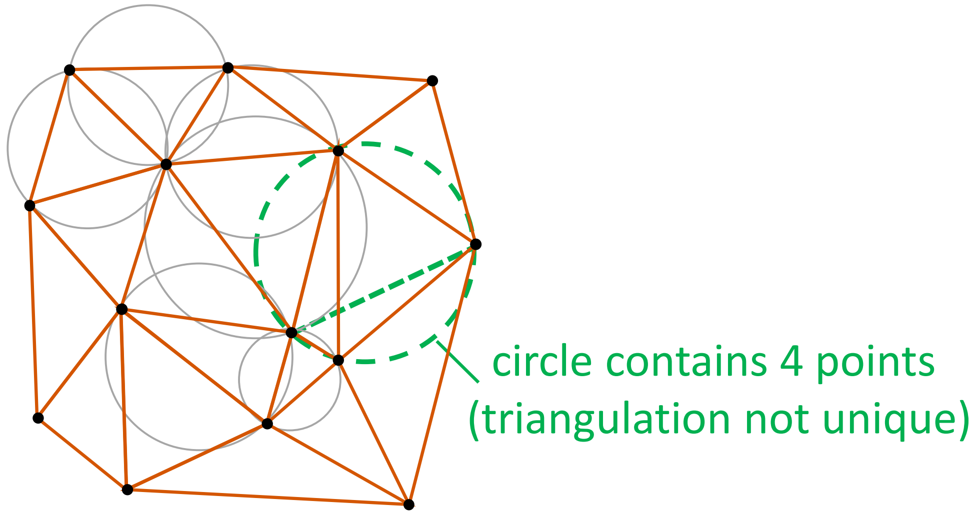




Delaunay triangulation

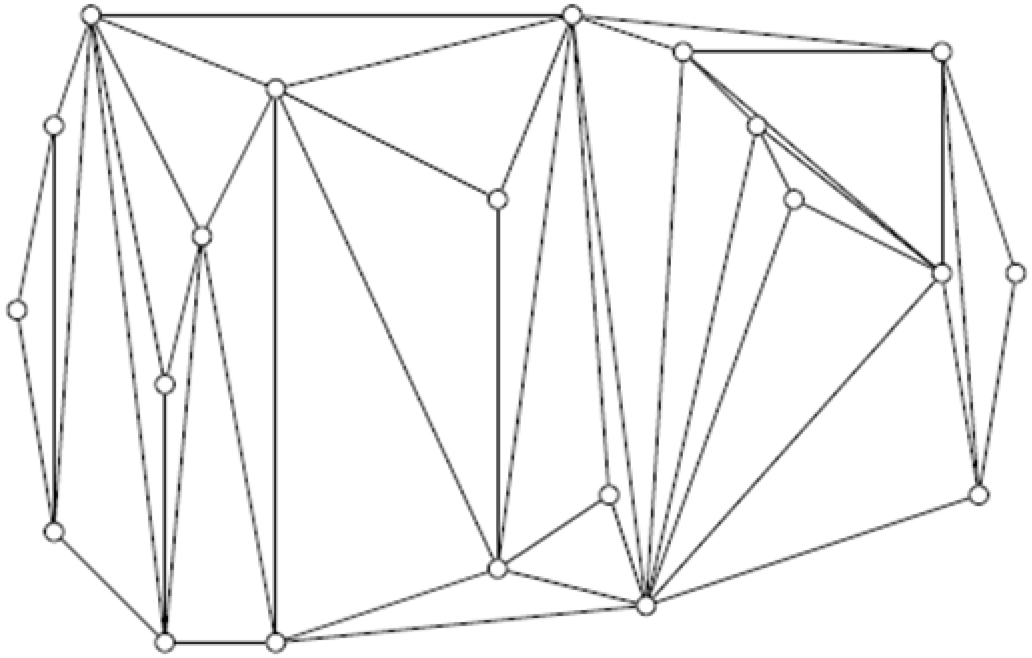
- Delaunay triangulation
  - The circumcircle of any triangle does not contain another point of the set
    - Maximizes the minimum angle in the triangulation
    - Such a triangulation is unique (independent of the order of samples) for all but trivial cases





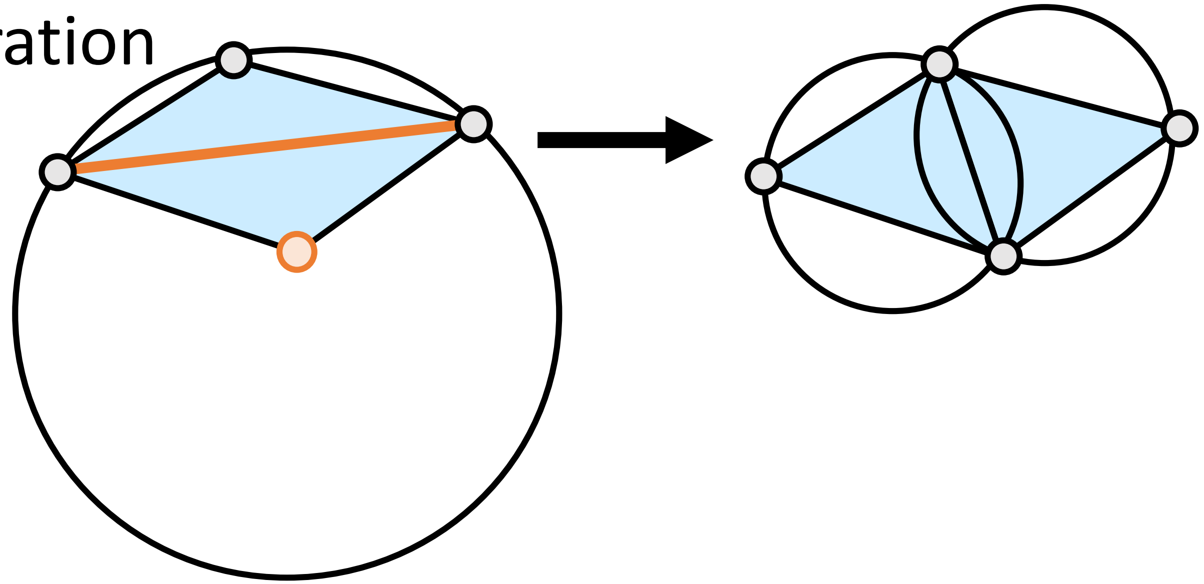
Delaunay triangulation

- How to build a Delaunay triangulation from an initial, non-optimal triangulation?
  - Can be performed by successively improving the initial triangulation via local operations



Delaunay triangulation

- Edge flip operation

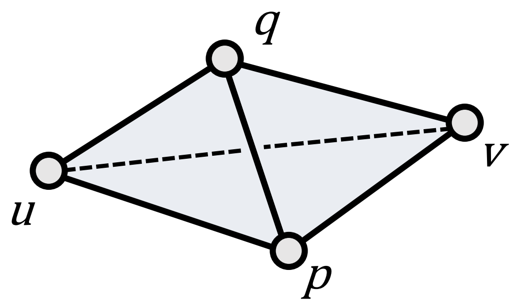


  - An edge is local Delaunay if there exists an empty circumcircle (otherwise illegal)
  - If an edge shared by two triangles is illegal, a flip operation generates a new edge that is legal!

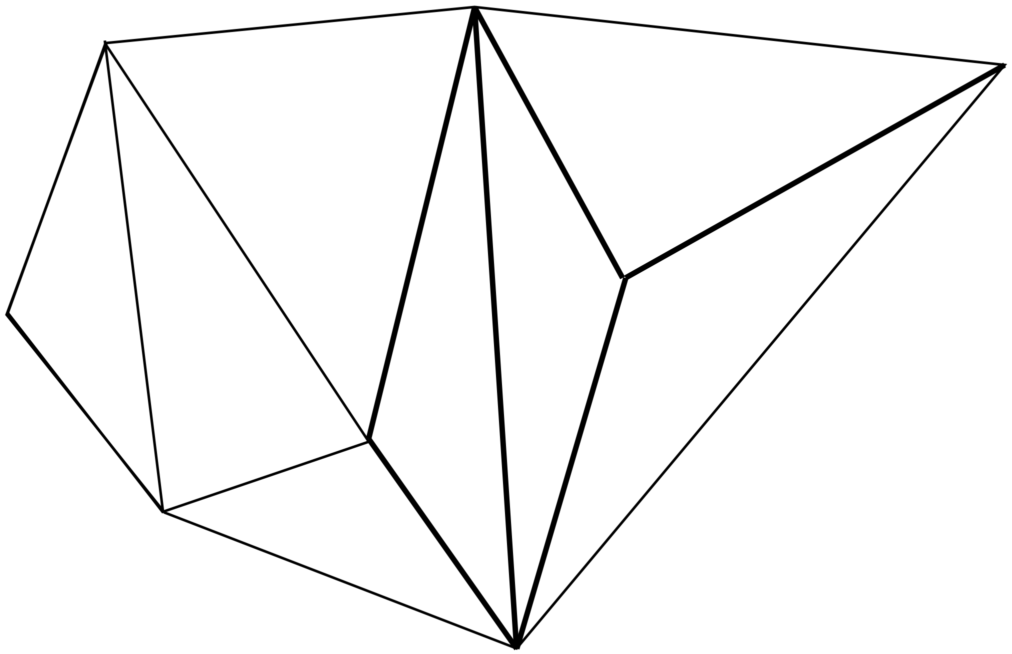
Delaunay triangulation

Edge flip algorithm:

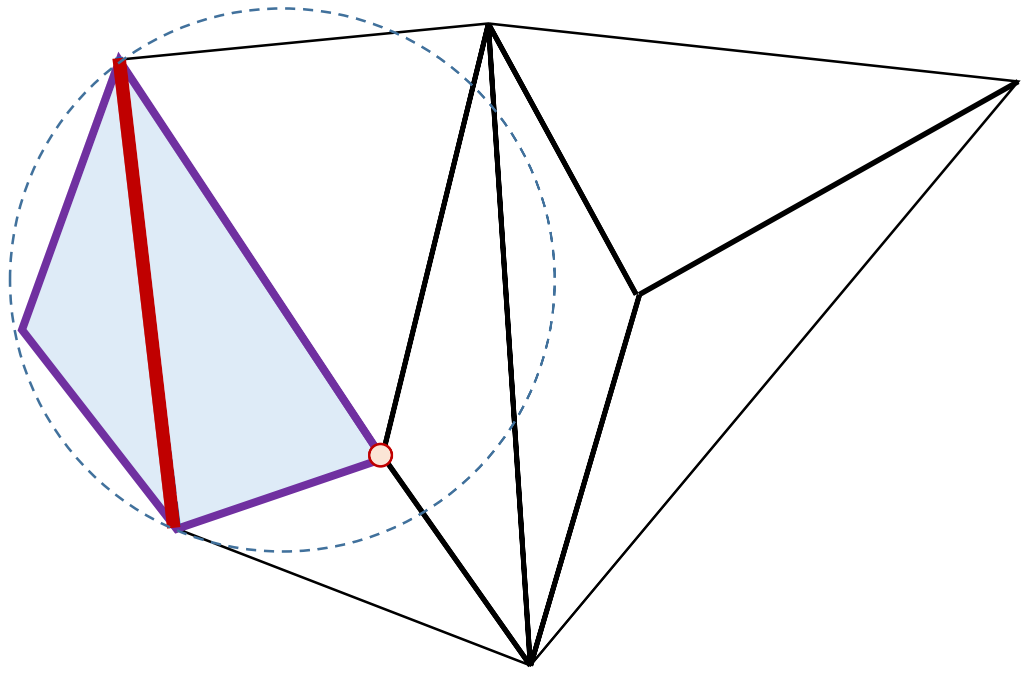
```
put all edges in stack and mark them
while stack is non-empty do
  pop edge uv from stack and unmark it
  if uv is illegal then
    substitute pq for uv    //edge flip
    for ab ∈ {up, pv, vq, qu} do
      if ab is unmarked then
        push ab on the stack and mark it
      endif
    endfor
  endif
endwhile
```



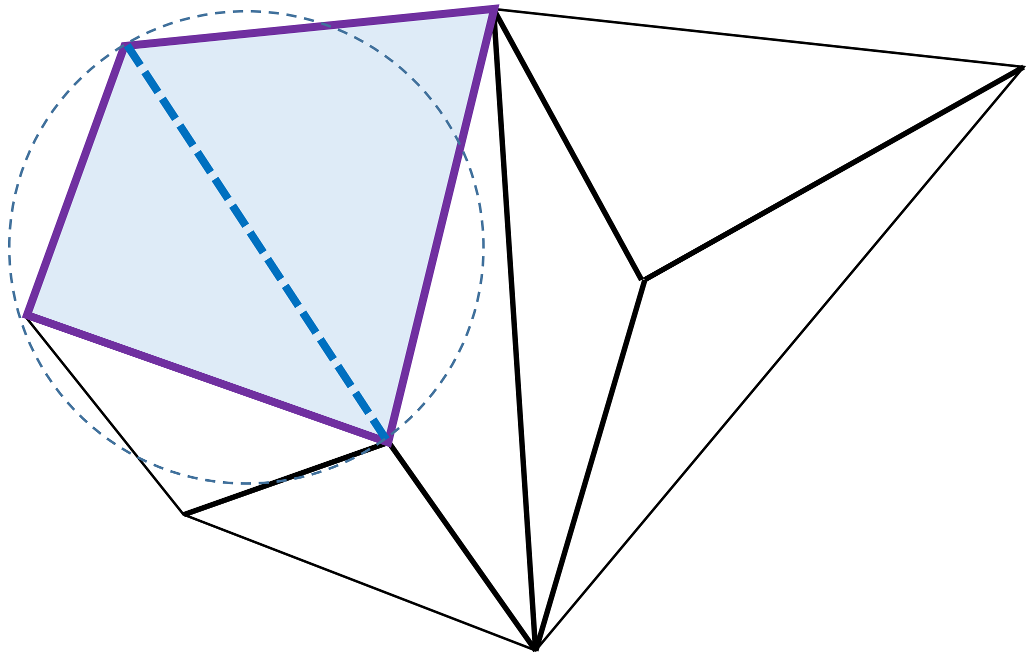
Edge flip – example



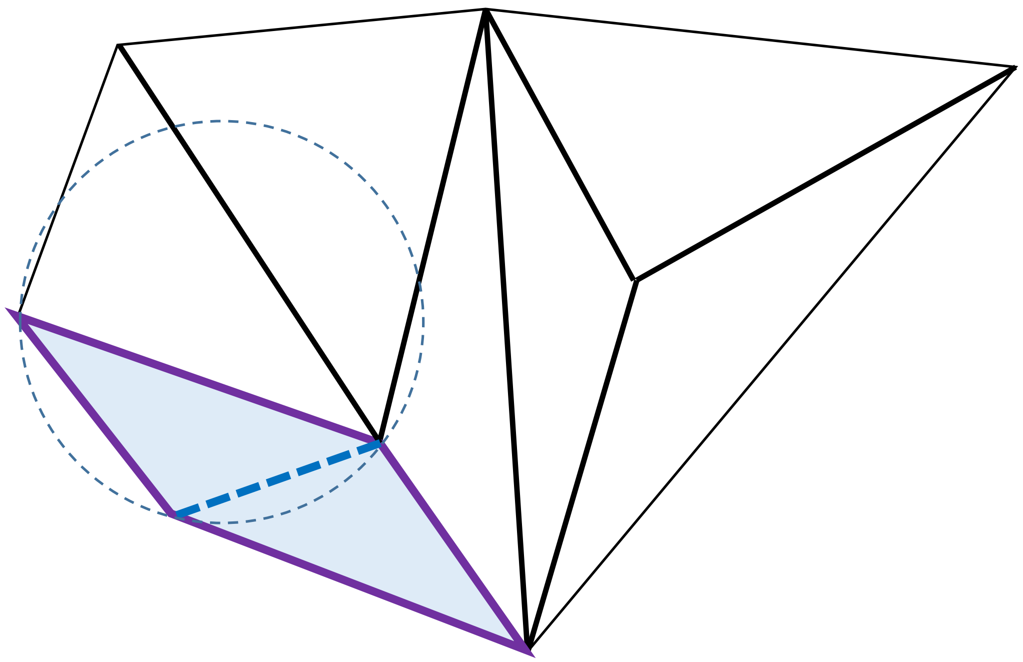
Edge flip – example



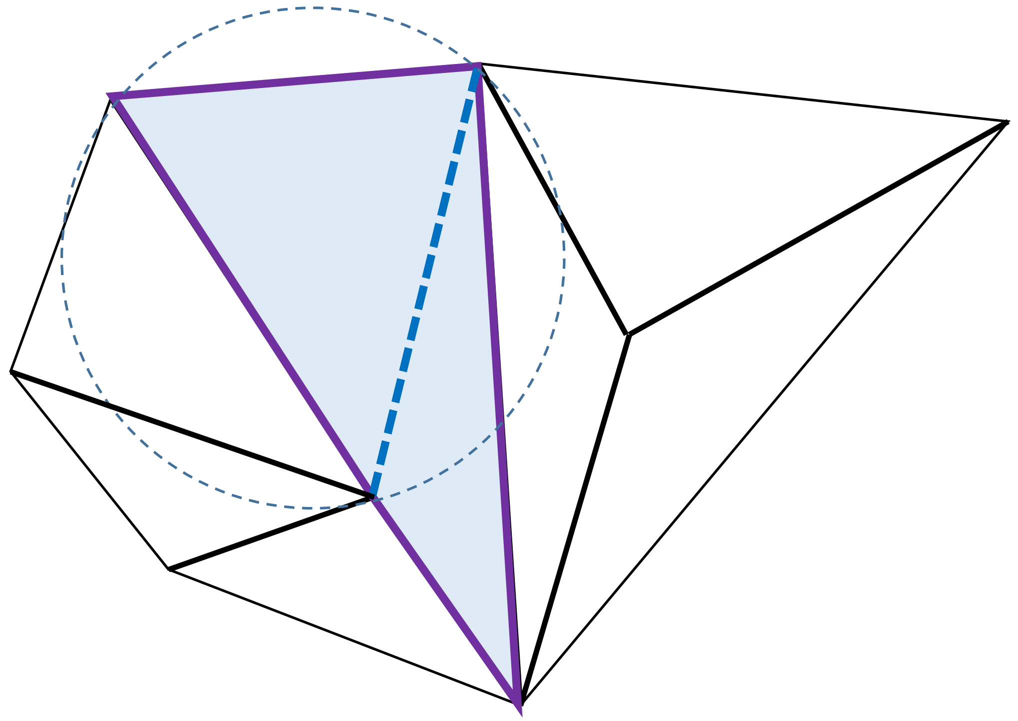
Edge flip – example



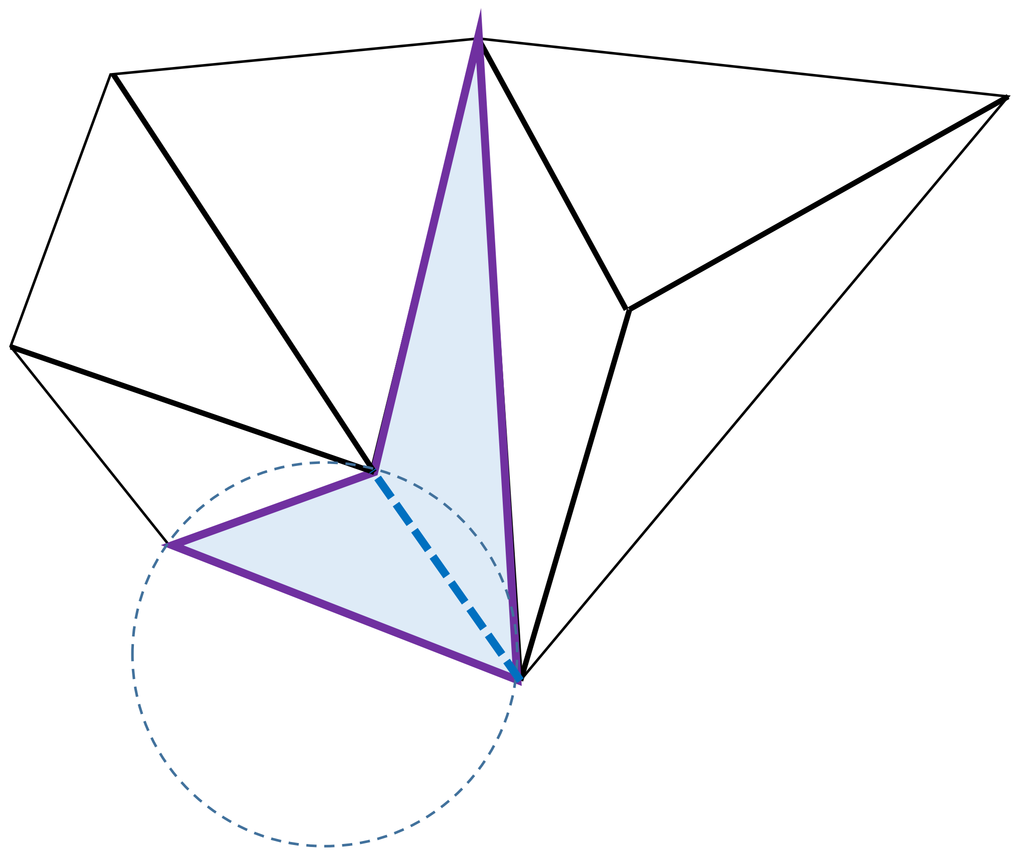
Edge flip – example



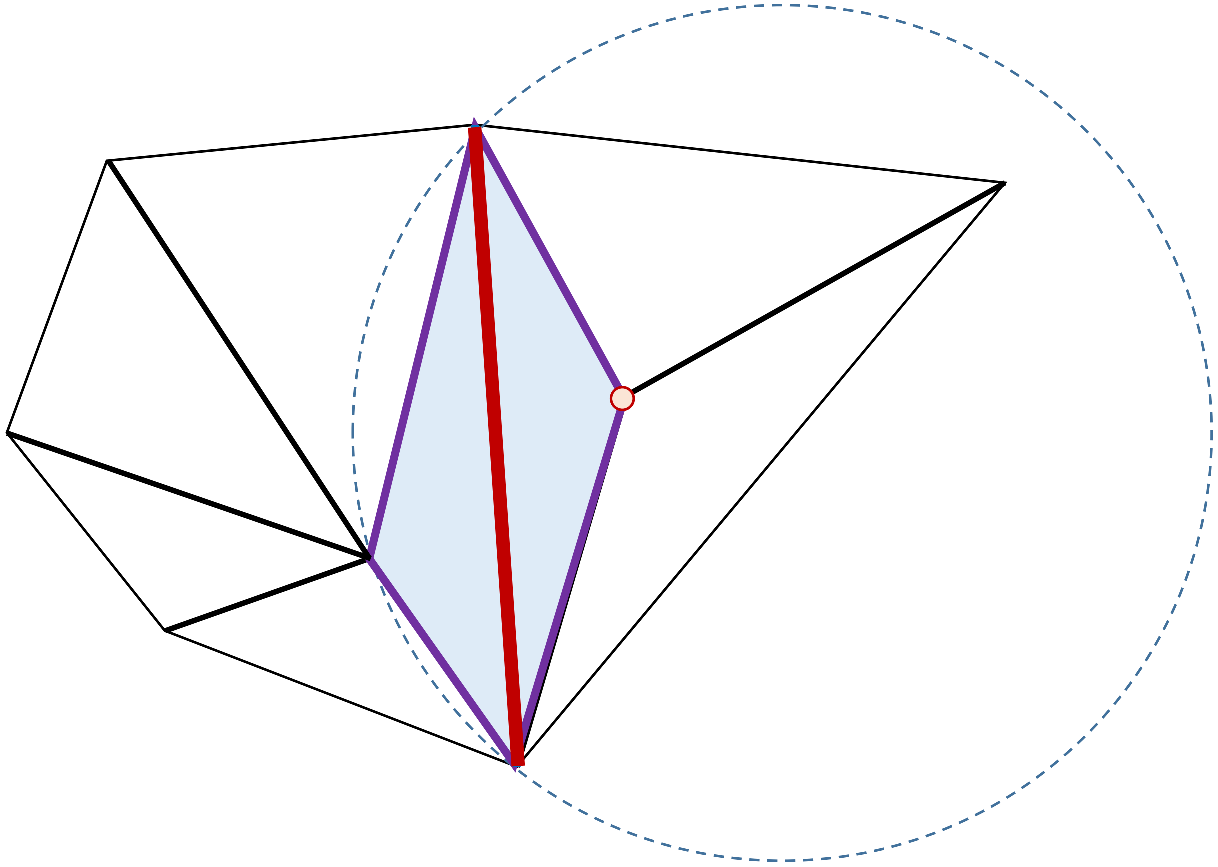
Edge flip – example



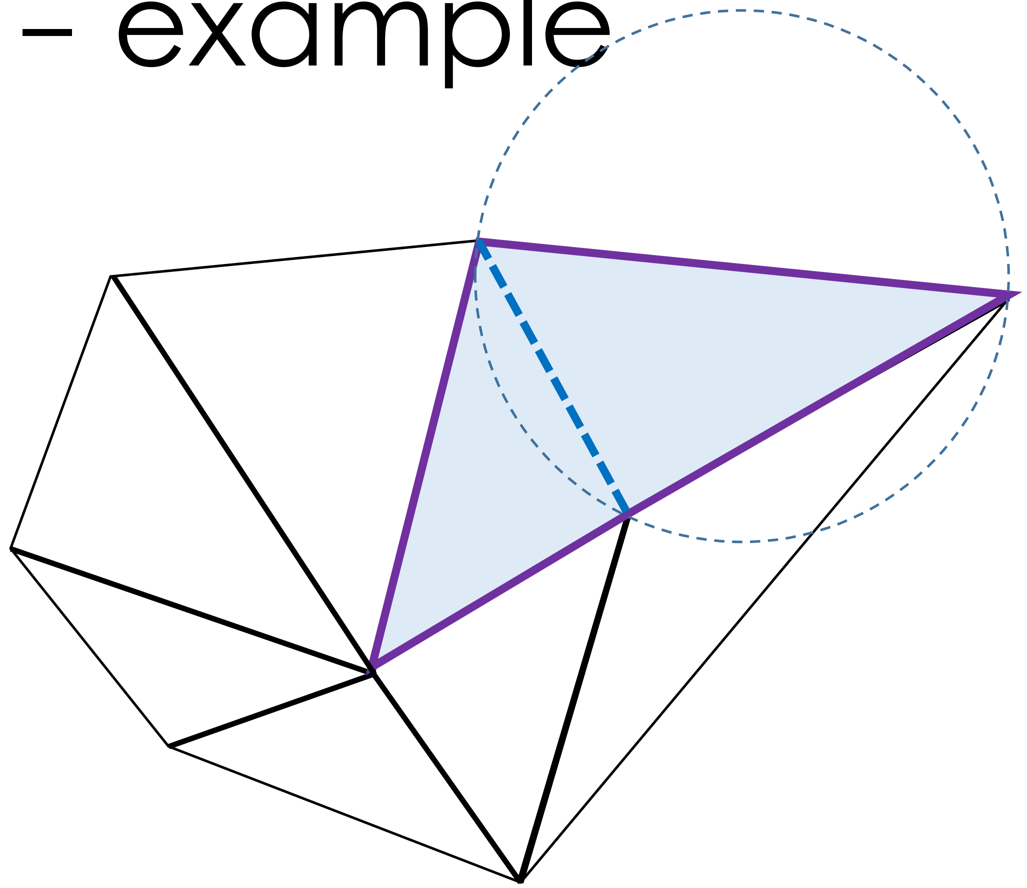
Edge flip – example



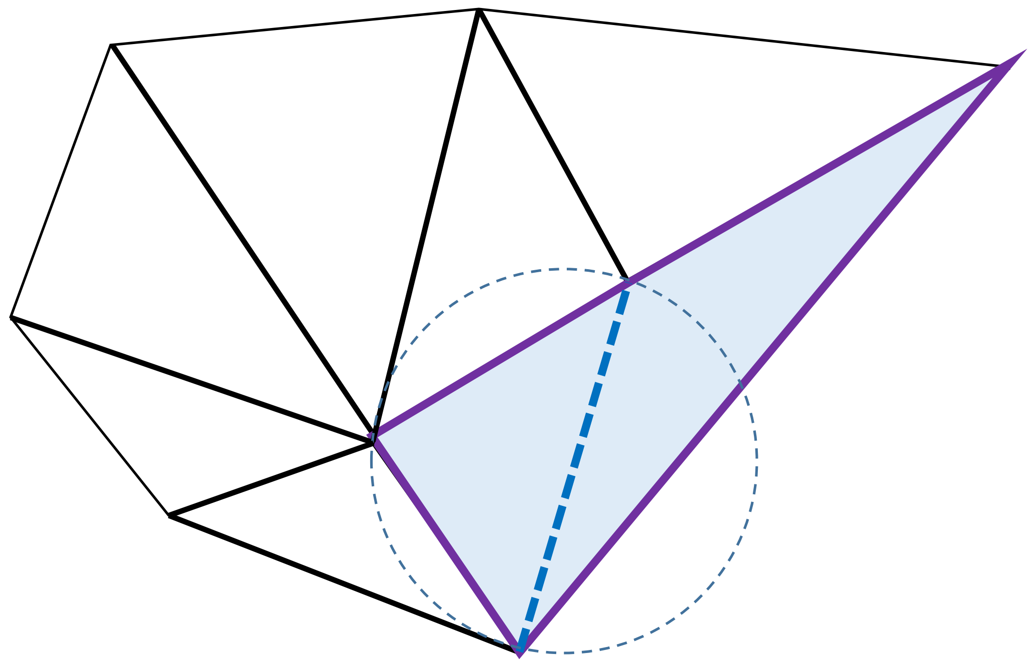
Edge flip – example



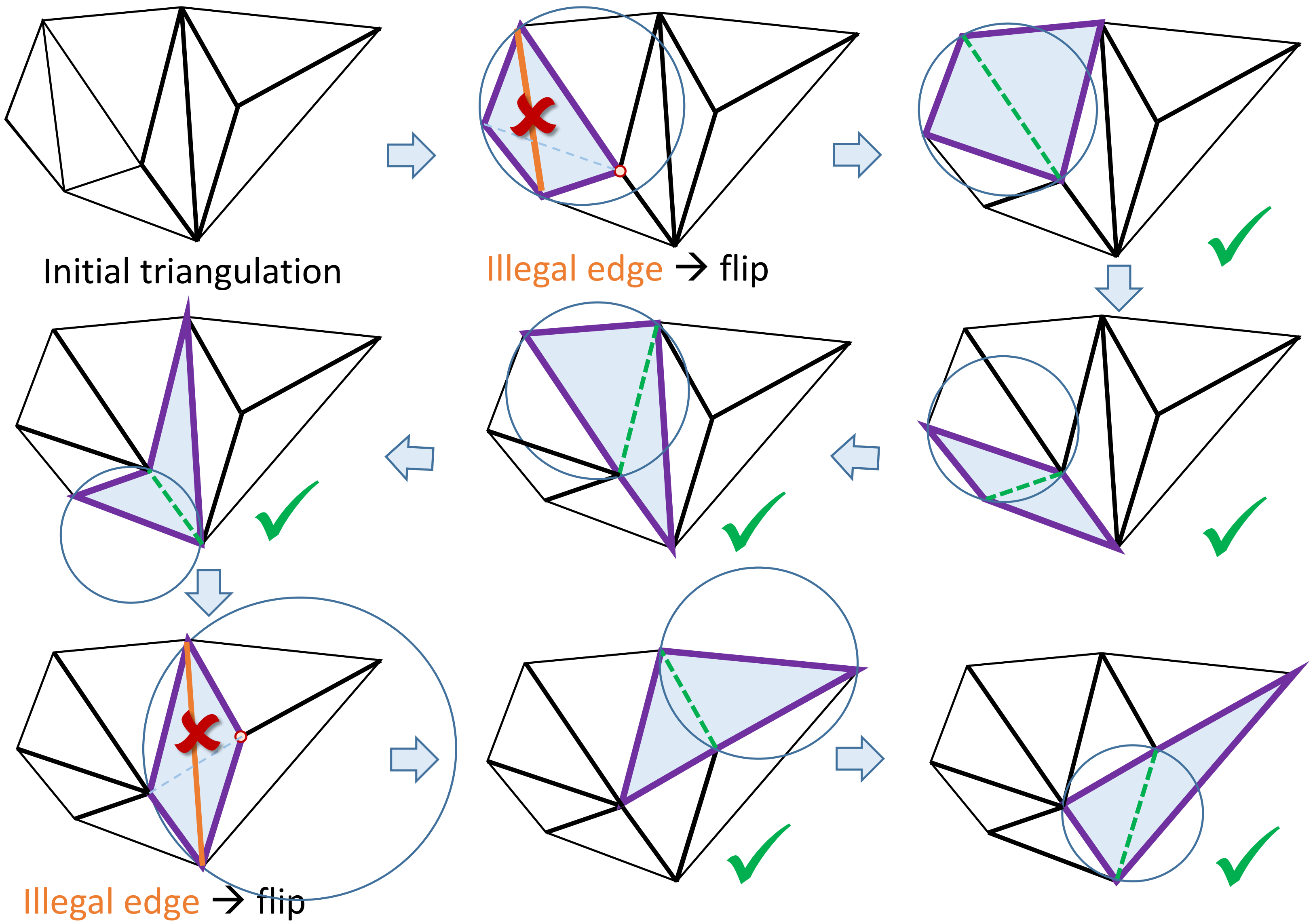
Edge flip – example



Edge flip – example

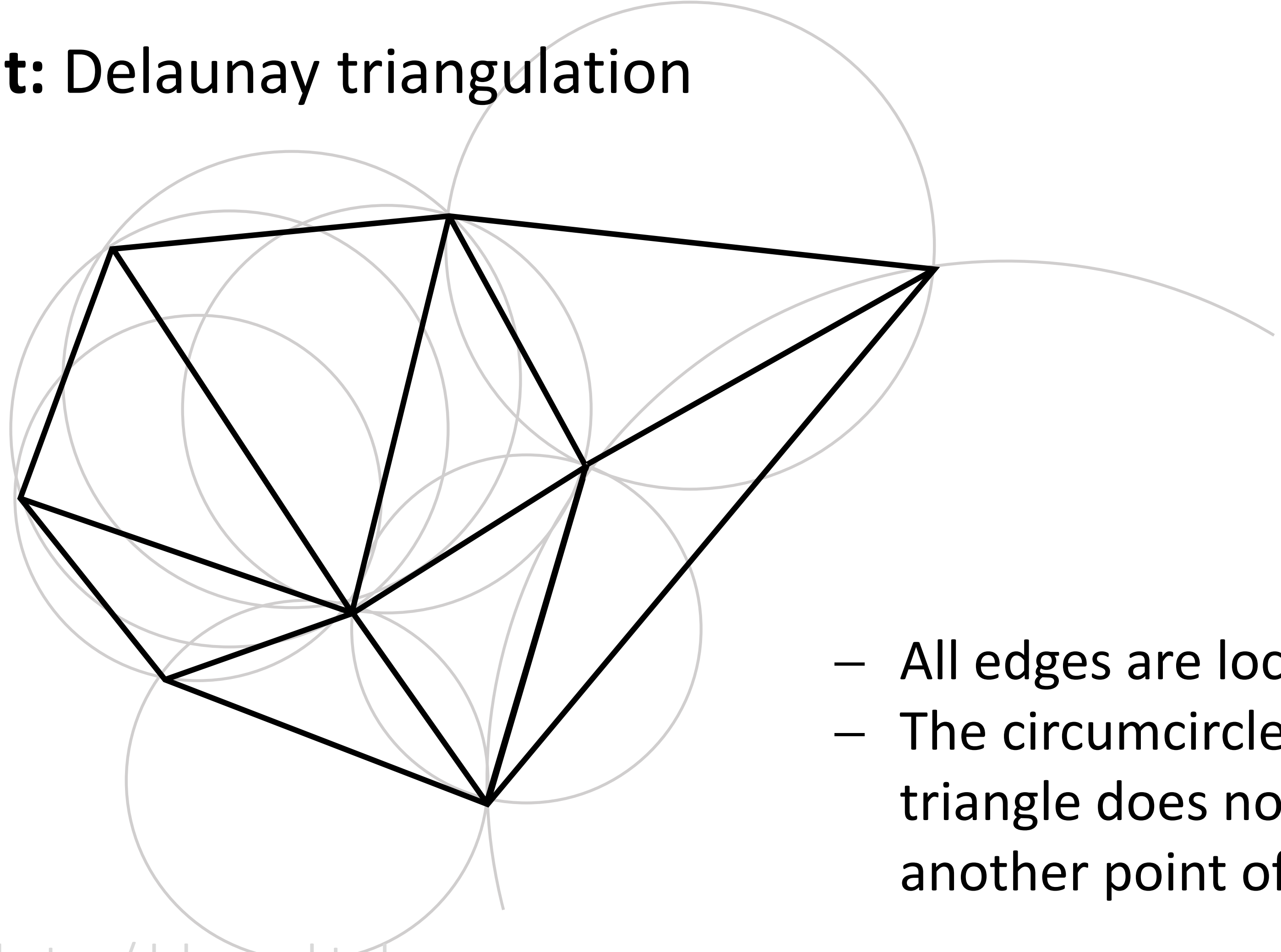


Edge flip – example



Edge flip – example

- **Result:** Delaunay triangulation

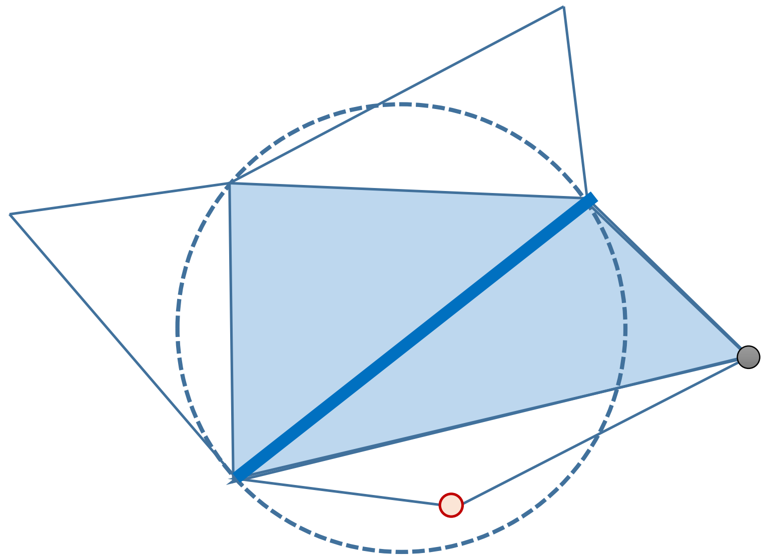


  – All edges are local Delaunay
  – The circumcircle of any triangle does not contain another point of the set

http://multivis.net/lecture/delaunay.html

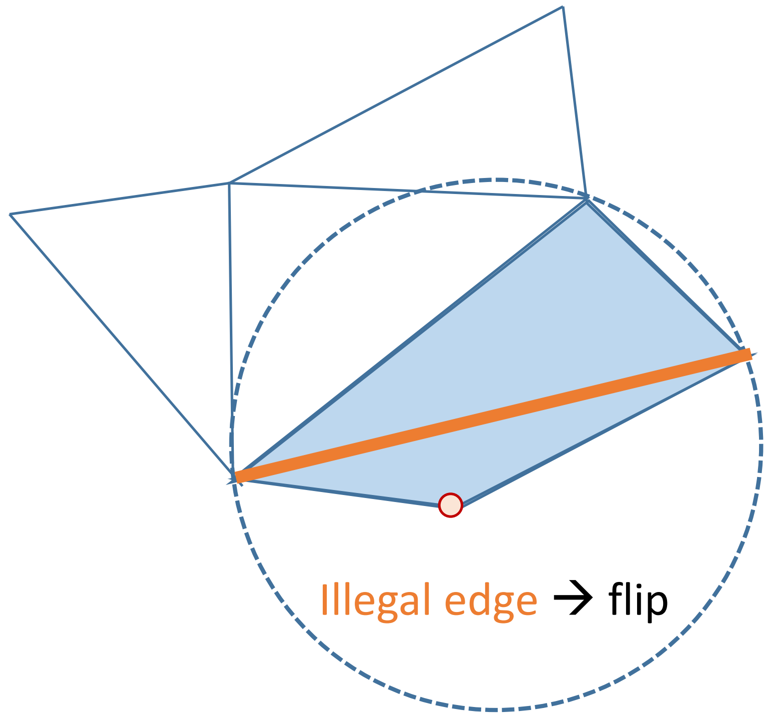
Delaunay triangulation

- Local vs. global optimality
  - Edge is locally Delaunay … but not globally



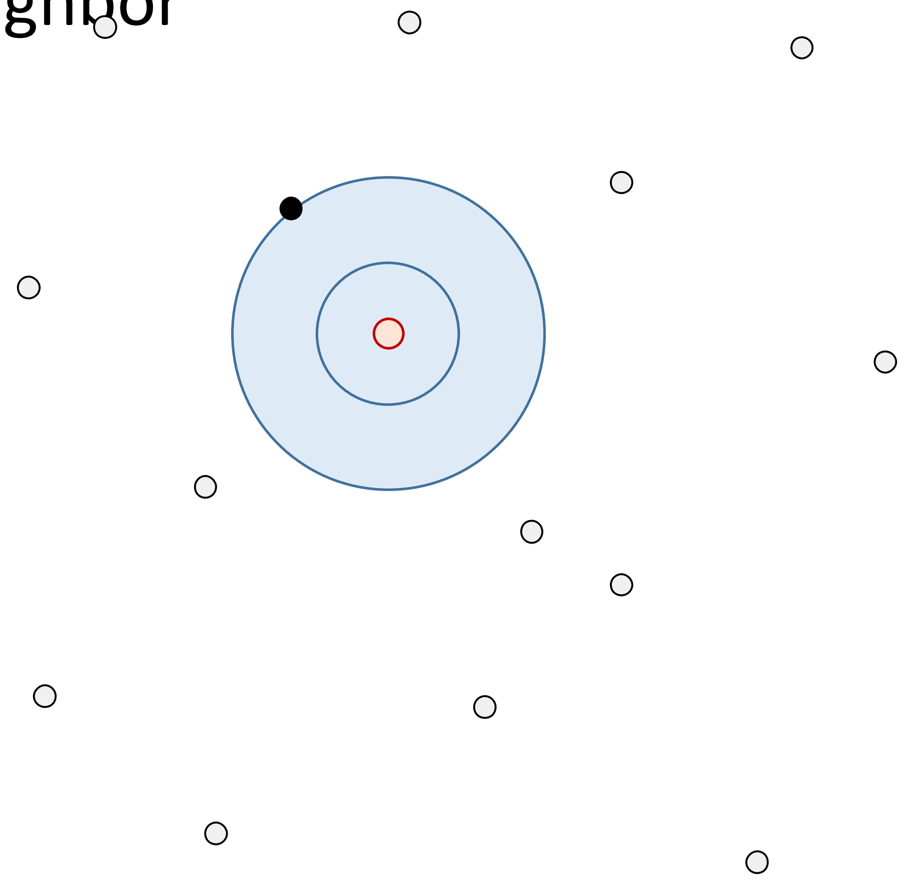
Delaunay triangulation

- Local vs. global optimality
  - If a triangulation is locally Delaunay everywhere → globally Delaunay



Voronoi diagram

- **Problem:** Looking for nearest neighbor

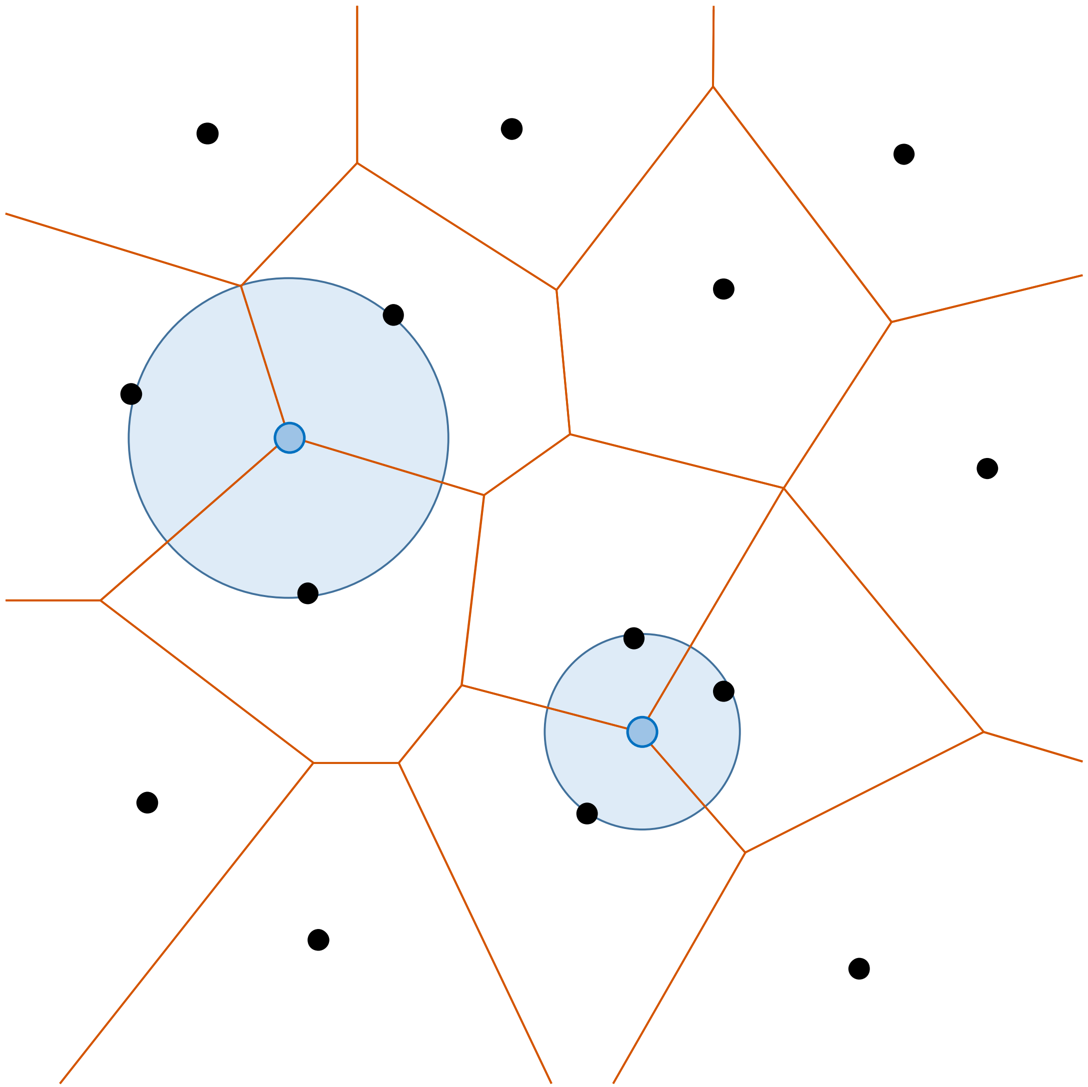


Voronoi diagram

- Partitions domain into Voronoi regions
  - Each region contains one initial sample – the Voronoi samples
  - Points in Voronoi region are closer to respective sample than to any other sample

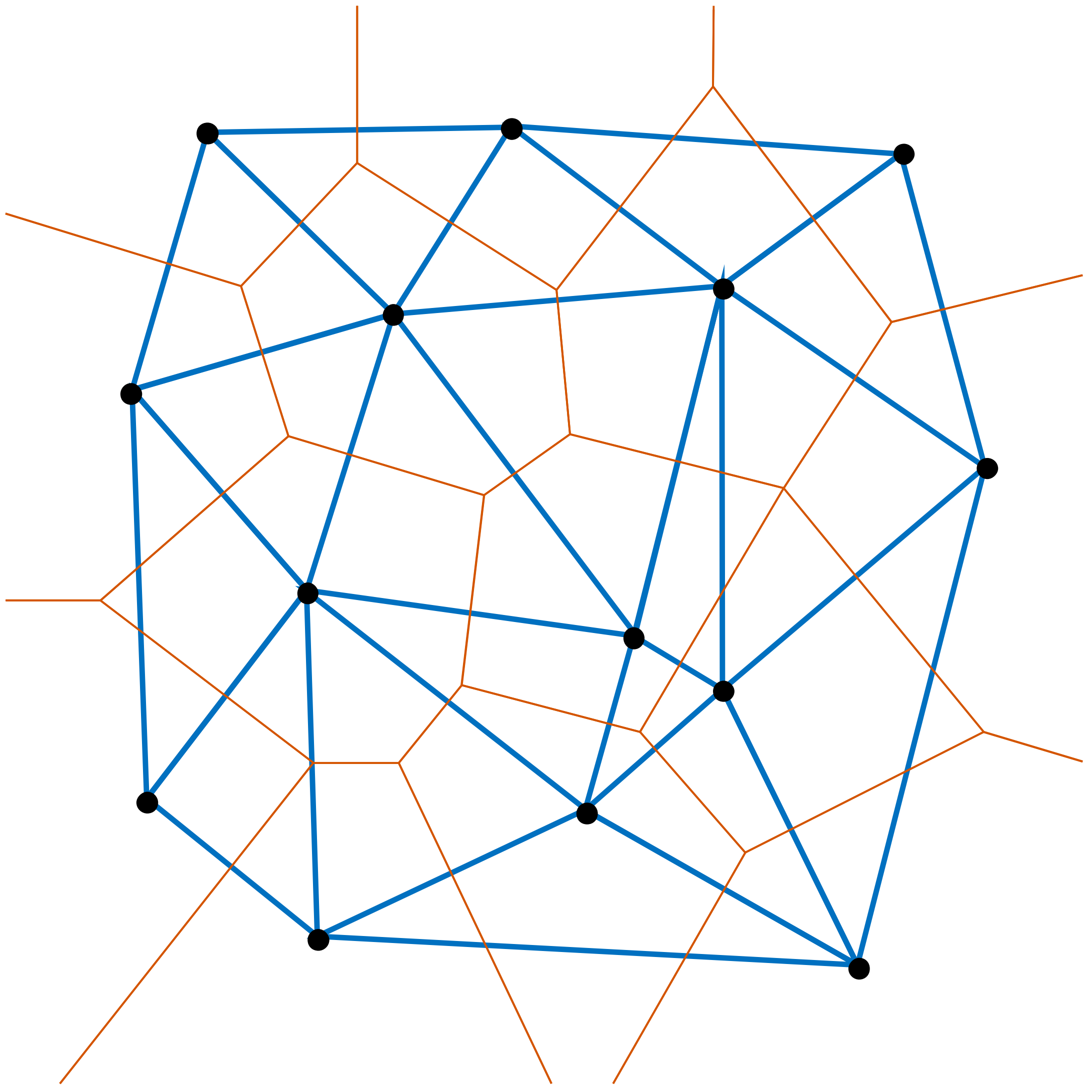
Voronoi diagram

- Centers of circumcircles of Delaunay triangulation



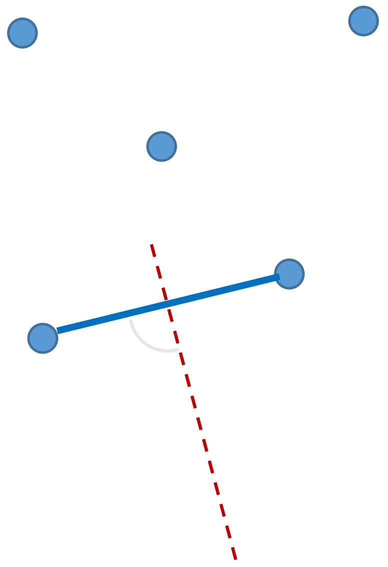
Voronoi diagram

- The geometric dual (topologically equal) of Delaunay triangulation
  - Voronoi samples are vertices in Delaunay triangulation



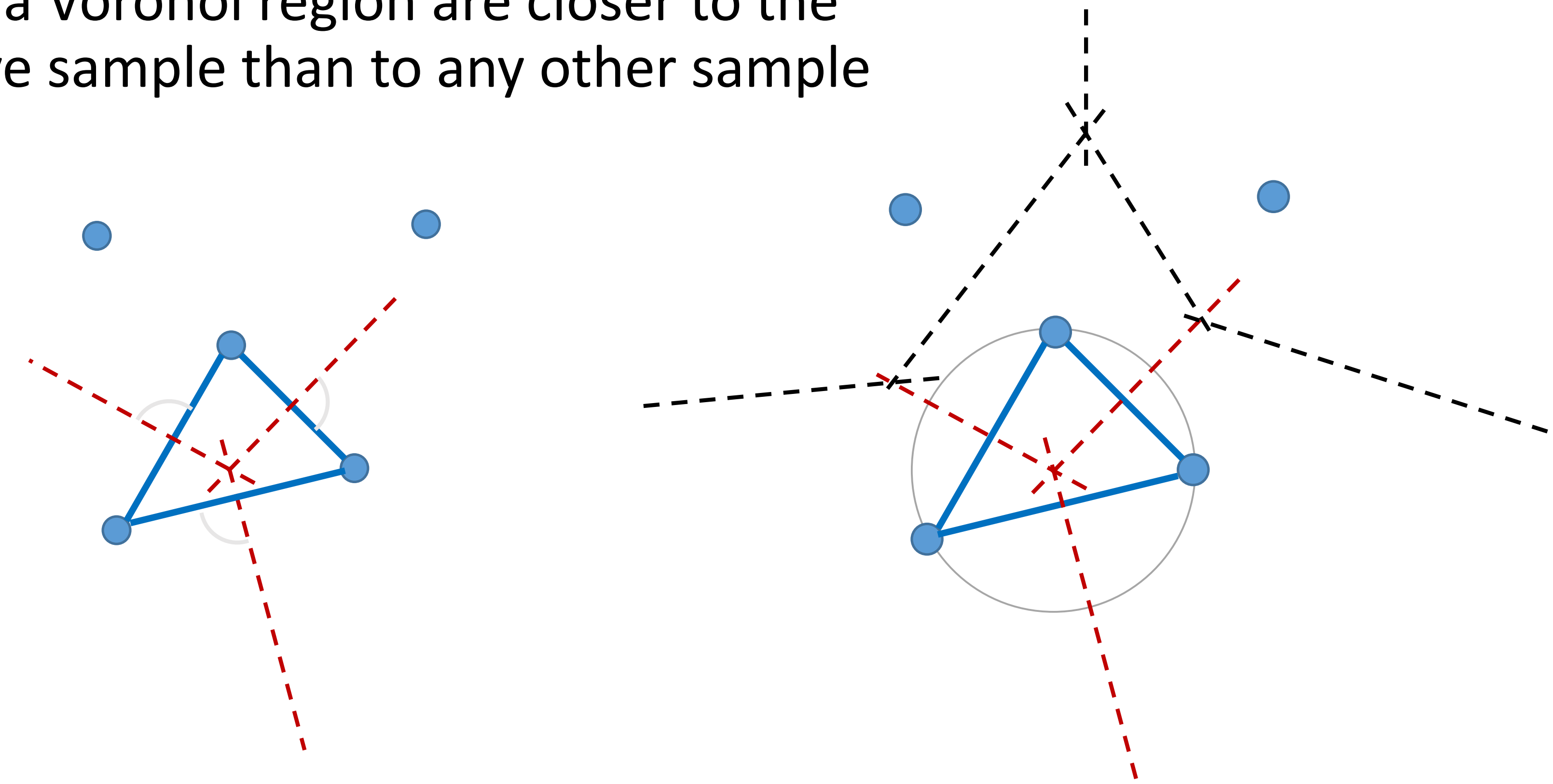
Voronoi diagram

- Construction
  - Points in a Voronoi region are closer to the respective sample than to any other sample



Voronoi diagram

- Construction
  - Points in a Voronoi region are closer to the respective sample than to any other sample

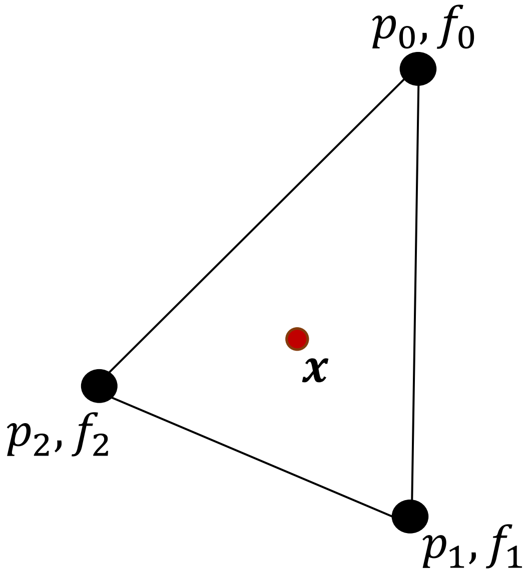


Interpolation

Data interpolation

- How to interpolate inside a triangle
  - The triangle lives in a (N = 2)D plane;
    it has $N + 1$ points (x_i, y_i) with values f_i
  - Can we find a function f that interpolates f_i at the points p_i, *i.e.,*

$$f(p_i) = f_i, \qquad i = 0, \ldots, N$$



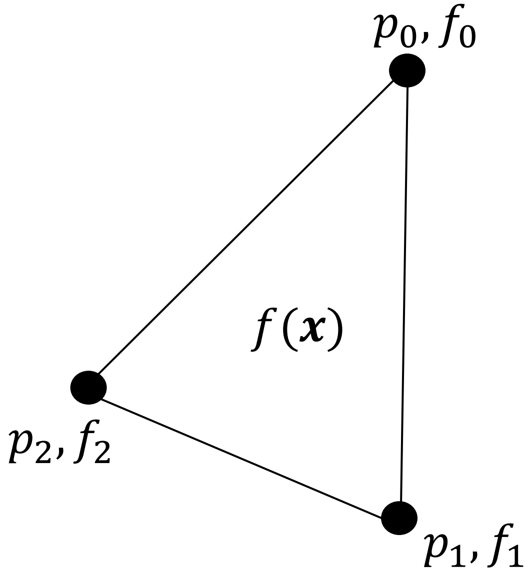
(interpolation constraint)

  - If so, then the value at any point $\boldsymbol{x}$ can be interpolated by evaluating $f(\boldsymbol{x})$

Data interpolation

- There is a unique linear function that satisfies the interpolation constraint
- A linear function can be written as

$$f(\boldsymbol{x}) = a + bx + cy$$



  - The unknown coefficients a, b, c can be obtained by solving the system

$$\begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}$$

Data interpolation

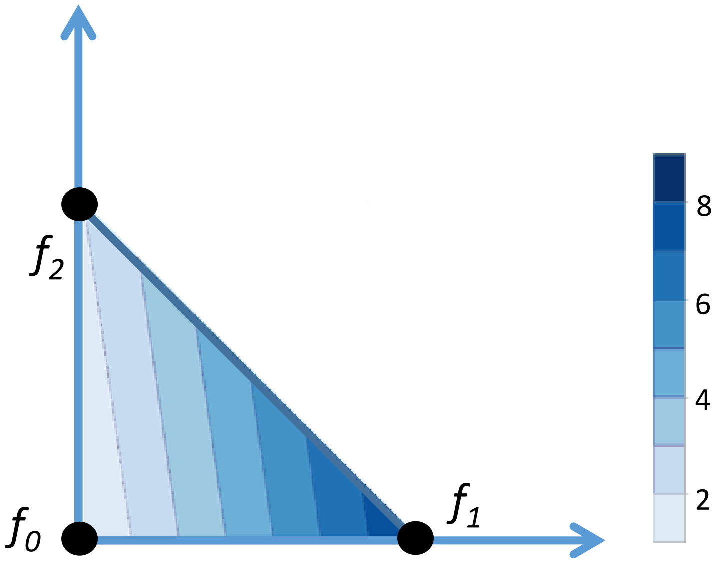
- Example

$$(x_0, y_0) = (0,0), \quad (x_1, y_1) = (1,0), \quad (x_2, y_2) = (0,1)$$
$$f_0 = 1, \quad f_1 = 8, \quad f_2 = 2$$

- Obtain a, b, c by solving the system

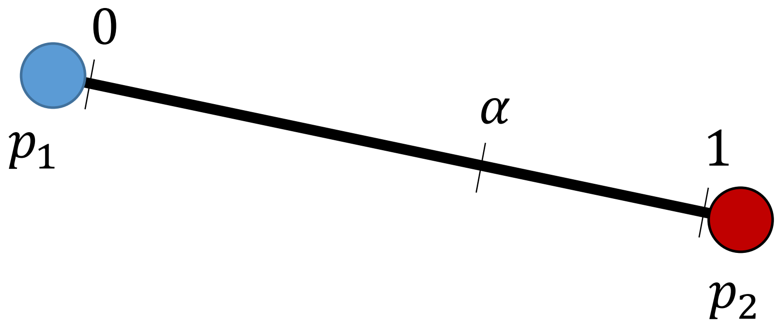
$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 2 \end{bmatrix}$$

$$a = 1, b = 7, c = 1$$
$$f(x, y) = 1 + 7x + 1y$$



Data interpolation

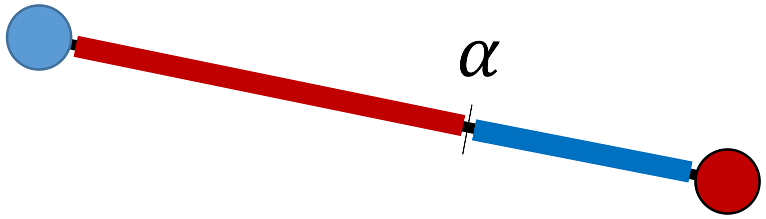
- Barycentric interpolation
  - Another way to interpolate inside a triangle, which yields the same linear interpolation as before
  - But let's solve a simpler problem first



  - We want to define a color for every $\alpha \in [0, 1]$

Data interpolation

- How do we come up with an equation?

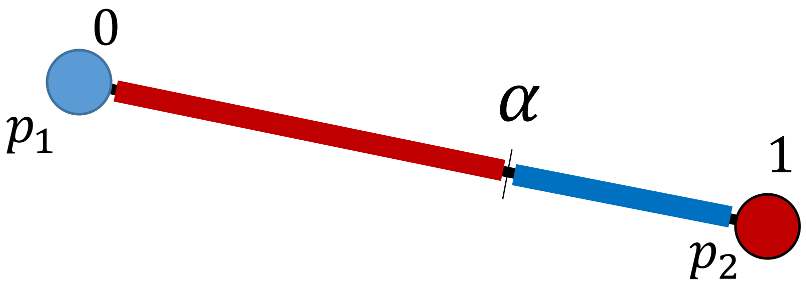


The further α is from the blue point, the more red we want
The further α is from the red point, the more blue we want

Percentage red = (length of red segment) / (total length)
Percentage blue = (length of blue segment) / (total length)

Data interpolation

- How do we come up with an equation?



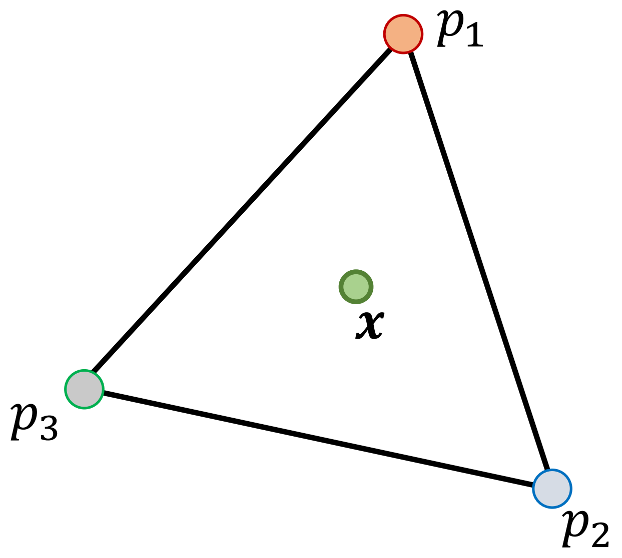
The further α is from the blue point, the more red we want
The further α is from the red point, the more blue we want

Percentage red = α
Percentage blue = $1 - \alpha$

$$f(\alpha) = \alpha \cdot p_2 + (1 - \alpha) \cdot p_1$$

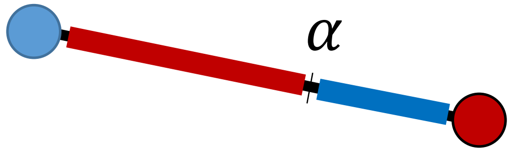
Data interpolation

- Barycentric interpolation
  - Now what about triangles?



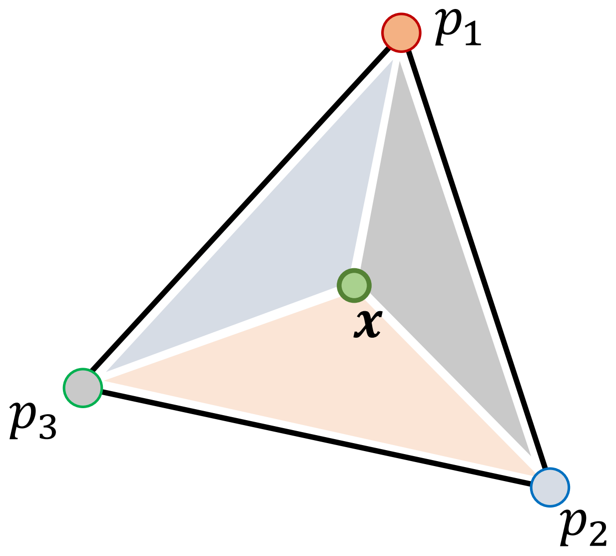
What's the interpolated value at the point $\boldsymbol{x}$?

In 1D we used ratios of lenghts



Data interpolation

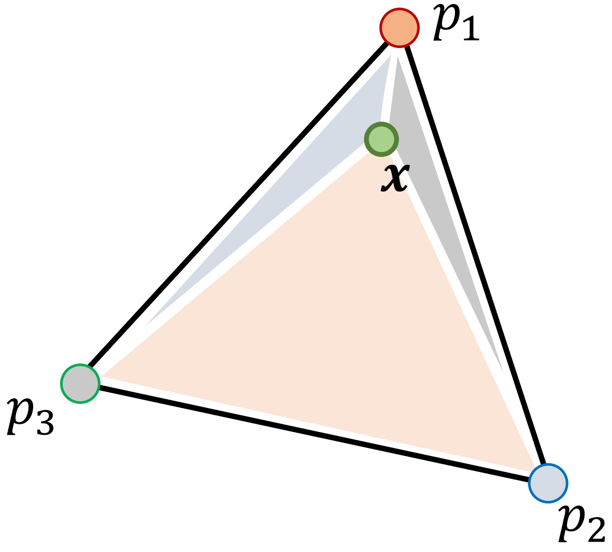
- Barycentric interpolation
  - Now what about triangles?



What about ratios of 2D areas?

Data interpolation

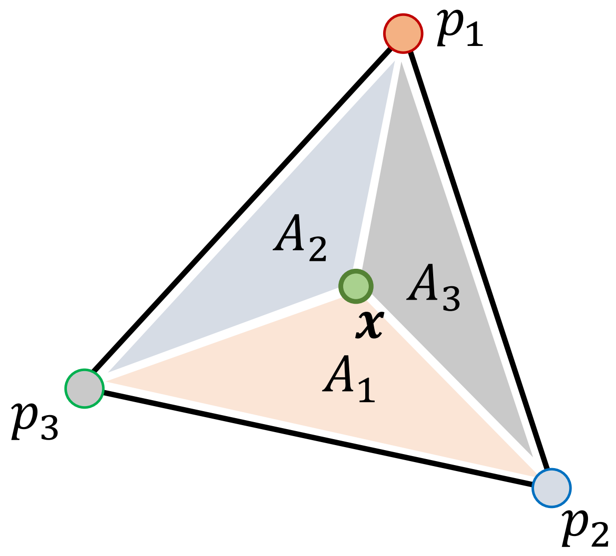
- Barycentric interpolation
  - Now what about triangles?



As $\boldsymbol{x}$ approaches the red point, the red area (for example) covers more of the triangle

Data interpolation

- Barycentric interpolation
  - Just like before:



$\alpha_1 = {}^{A_1}/_{A}$ percentage red

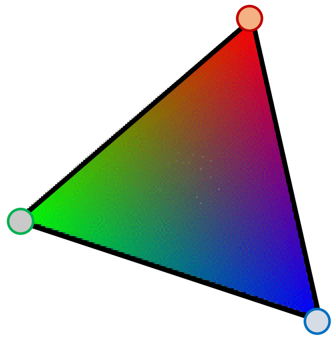
$\alpha_2 = {}^{A_2}/_{A}$ percentage blue

$\alpha_3 = {}^{A_3}/_{A}$ percentage green

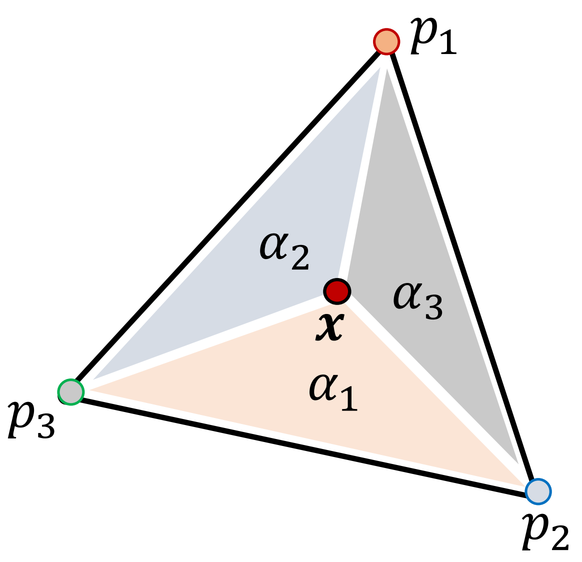
A … area of whole triangle

$$\boldsymbol{x} = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1$$



Data interpolation



$$\alpha_1 = \frac{\text{area}(\Delta \boldsymbol{x} p_2 p_3)}{\text{area}(\nabla p_1 p_2 p_3)}$$

$$\alpha_2 = \frac{\text{area}(\Delta p_1 \boldsymbol{x} p_3)}{\text{area}(\nabla p_1 p_2 p_3)}$$

$$\alpha_3 = \frac{\text{area}(\Delta p_1 p_2 \boldsymbol{x})}{\text{area}(\nabla p_1 p_2 p_3)}$$

Barycentric interpolation

$$\boldsymbol{x} = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1$$

Inside triangle criteria

$$0 \le \alpha_1 \le 1$$

$$0 \le \alpha_2 \le 1$$

$$0 \le \alpha_3 \le 1$$

Data interpolation

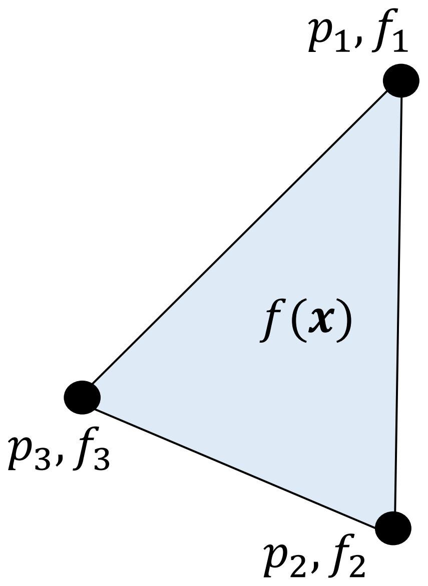
- Barycentric interpolation
  - Every point $\boldsymbol{x}$ in a triangle can be written as a barycentric combination of the vertices p_i:

$$\boldsymbol{x} = \sum_i \alpha_i p_i \qquad \text{with} \qquad \sum_i \alpha_i = 1$$

(α_i … barycentric coordinates)

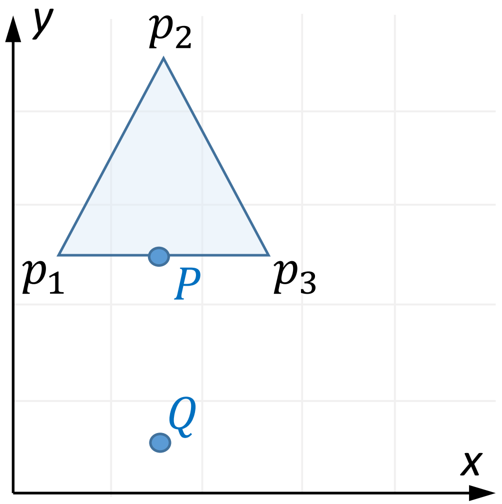
  - If α_i are known, then $f(\boldsymbol{x})$ can be interpolated from values f_i at the vertices via

$$f(\boldsymbol{x}) = \alpha_1 f_1 + \alpha_2 f_2 + \underbrace{(1 - \alpha_1 - \alpha_2)}_{\alpha_3} f_2$$



Data interpolation

Example: Given a triangle with vertices $p_1 = (0.5, 2.5)$, $p_2 = (1.5, 4.5)$ and $p_3 = (2.5, 2.5)$. Compute the barycentric coordinates of the points $P = (1.5, 2.5)$ and $Q = (1.5, 0.5)$ with respect to the triangle.



Point P:

$$\alpha_2 = 0 \;\rightarrow \alpha_1 = \alpha_3 = 0.5$$

Point Q:

$$I\text{: } 1.5 = 0.5\,\alpha_1 + 1.5\,\alpha_2 + 2.5\,\alpha_3$$ ← x coordinates

$$II\text{: } 0.5 = 2.5\,\alpha_1 + 4.5\,\alpha_2 + 2.5\,\alpha_3$$ ← y coordinates

$$III\text{: } 1 = \alpha_1 + \alpha_2 + \alpha_3$$

$$I - II\text{: } 1 = -2\alpha_1 - 3\alpha_2$$

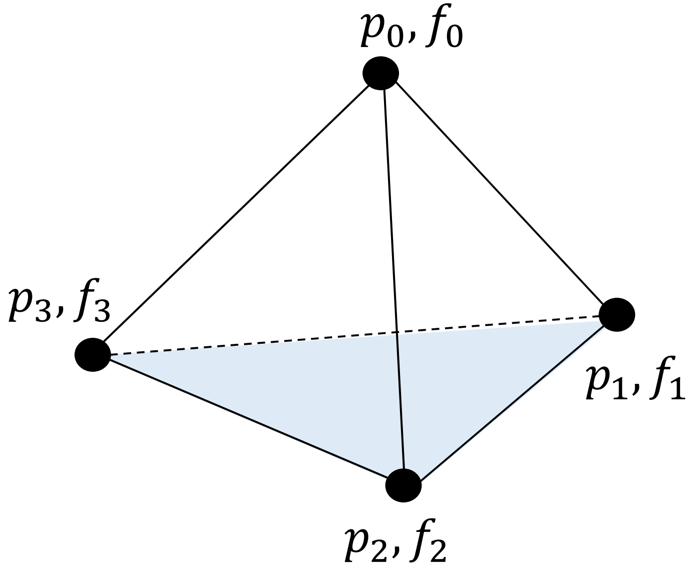
$$II - 2.5\,III\text{: } -2 = 2\alpha_2 \quad \rightarrow \alpha_2 = -1$$

$$\alpha_2 \rightarrow I'\text{: } -2 = -2\alpha_1 \quad \rightarrow \alpha_1 = 1 \quad \rightarrow \alpha_3 = 1$$

http://kehrer.bplaced.com/lecture/barycentric.html

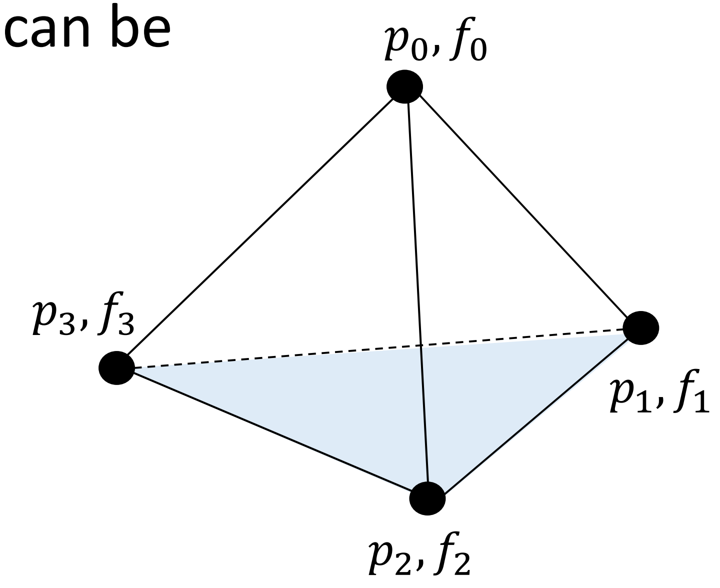
Data interpolation

- Interpolation of scalar values in a tetrahedron
  - A unique linear interpolation function $f(\boldsymbol{x}) = a + bx + cy + dz$ exists which interpolates the scalar values at the vertices
  - Solve for coefficients a, b, c, d via
    $$\begin{bmatrix} 1 & x_0 & y_0 & z_0 \\ 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$



Data interpolation

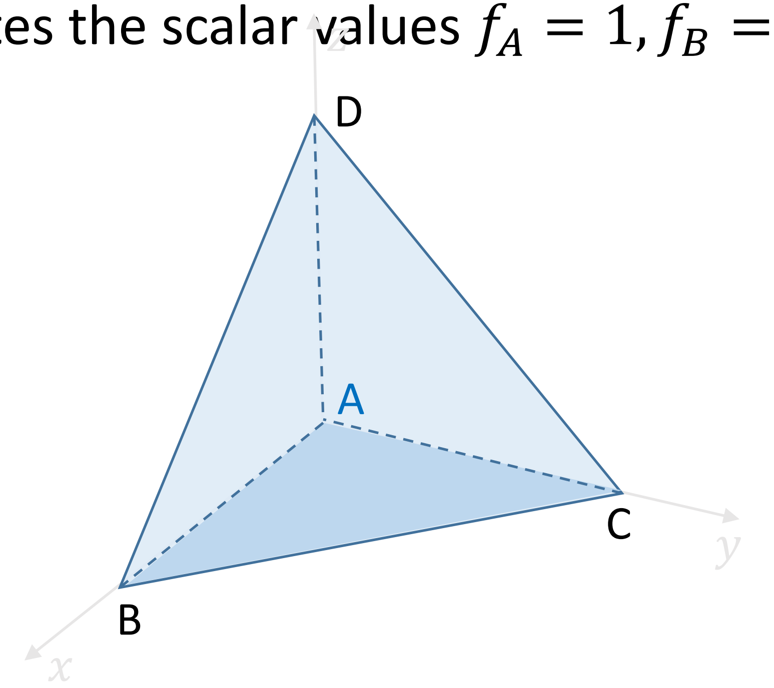
- How to get the gradient inside the tetrahedron?
  - Given the linear interpolation function
$$f(\boldsymbol{x}) = a + bx + cy + dz$$
  - The gradient of the interpolated scalar field can be obtained by
$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} = \begin{pmatrix} b \\ c \\ d \end{pmatrix}$$
  - The gradient is constant within the tetrahedron



Data interpolation

Example: For the tetrahedron with vertices A = (0,0,0), B = (1,0,0), C = (0,1,0), D = (0,0,1), compute the linear interpolation function $f(x, y, z) = a + b \cdot x + c \cdot y + d \ \cdot z$ which interpolates the scalar values $f_A = 1, f_B = 0, f_C = 0, f_D = 1$ at the corresponding vertices.

$$A \rightarrow f: 1 = a$$
$$B \rightarrow f: 0 = a + b \ \rightarrow b = -1$$
$$C \rightarrow f: 0 = a + c \ \rightarrow c = -1$$
$$D \rightarrow f: 1 = a + d \rightarrow d = 0$$
$$\rightarrow f(x, y, z) = 1 - x - y$$



Compute the gradient of the interpolated scalar field at the center of the tetrahedron.

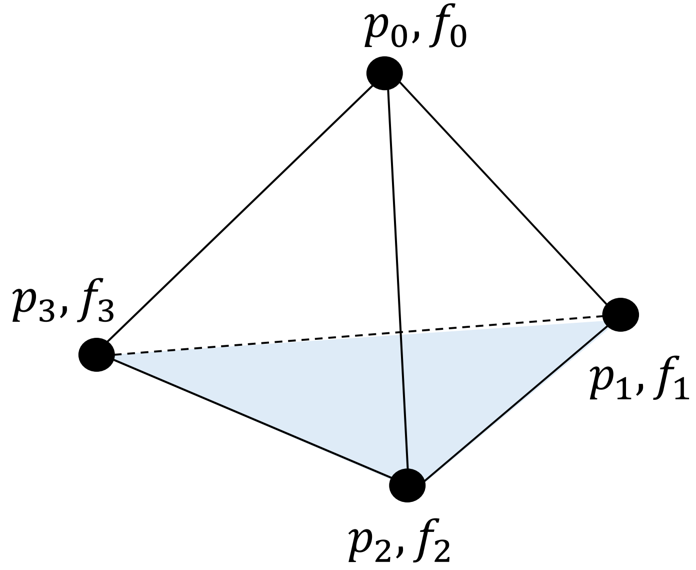
$$\nabla f = (-1, -1, 0)^T$$

Data interpolation

- Barycentric interpolation in 3D
  - Scalar values can be interpolated by means of barycentric coordinates:

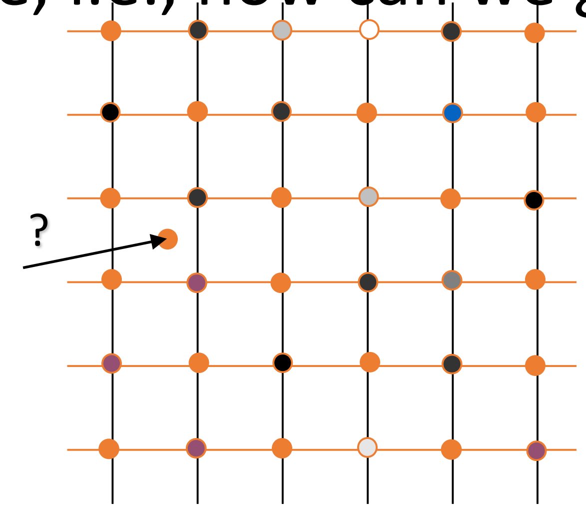
$$\boldsymbol{x} = \alpha_0 p_0 + \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3$$
$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$$

$$\rightarrow f(x, y, z) = \alpha_0 f_0 + \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3$$



Interpolation on grids

- **Problem**: assume data values are given only at vertices of a Cartesian grid
- How can we hide the underlying grid structure, i.e., how can we get a continuous data distribution over the spatial domain?



- This can be done via interpolation!

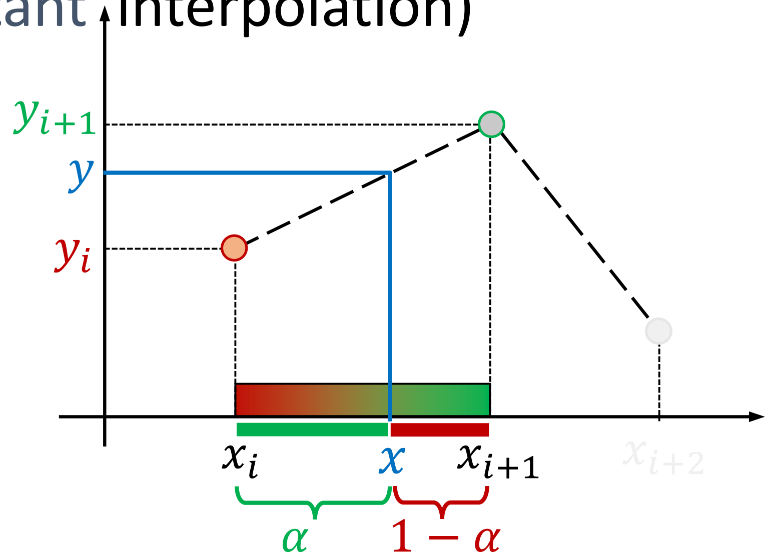
Interpolation on grids

- Piecewise linear interpolation
  - Simplest approach (except for piece-wise constant interpolation)
  - Data points: $(x_1,y_1),\ldots,(x_N,y_N)$
  - For any point x with

$$x_i \leq x \leq x_{i+1}$$

evaluate $\boxed{f(x) = (1-\alpha)y_i + \alpha y_{i+1}}$

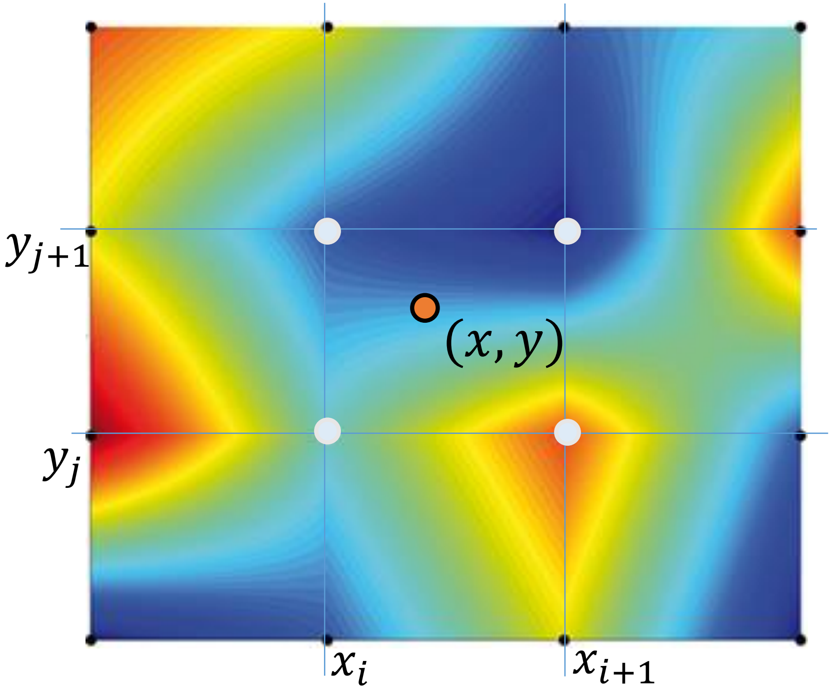
where $\alpha = \frac{x-x_i}{x_{i+1}-x_i} \in [0,1]$



The further x is away from red point, the more green we want

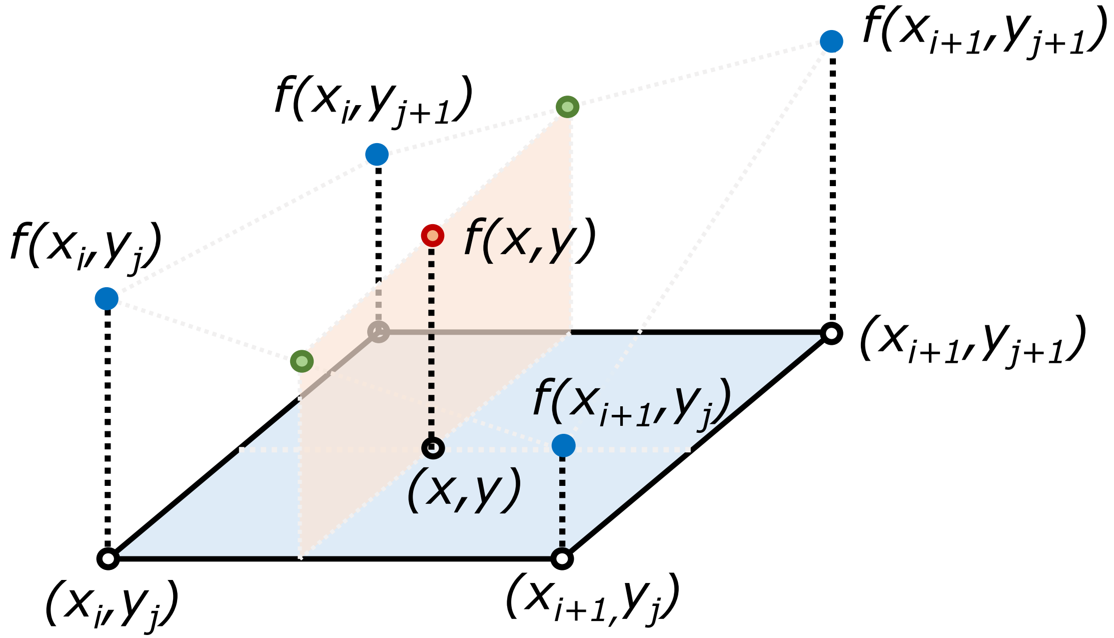
Interpolation on grids

- Linear interpolation
  - C^0 continuity at segment boundaries
    - Tangents don't match at segment transition
  - Easily extendible to 2D
    - 2D cell consisting of 4 data points $(x_i, y_j), \ldots, (x_{i+1}, y_{j+1})$ with scalar values $f_{k,l} = f(x_k, y_l)$
    - Bilinear interpolation of points (x, y) with $x_i \leq x \leq x_{i+1}$ and $y_j \leq y \leq y_{j+1}$



Interpolation on grids

- Bilinear interpolation on a rectangle



Interpolation on grids

- Bilinear interpolation on a rectangle

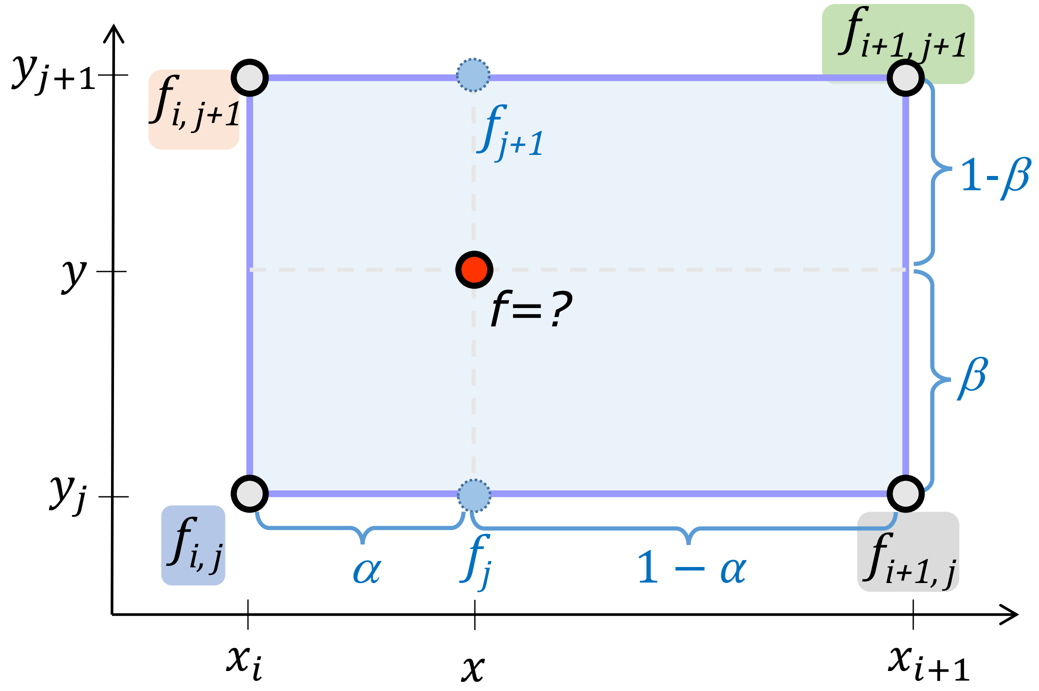
$$f(\alpha, \beta) = ?$$

Interpolate horizontally

$$f_j = (1-\alpha)\, f_{i,j} + \alpha\, f_{i+1,j}$$

$$f_{j+1} = (1-\alpha)\, f_{i,j+1} + \alpha\, f_{i+1,j+1}$$

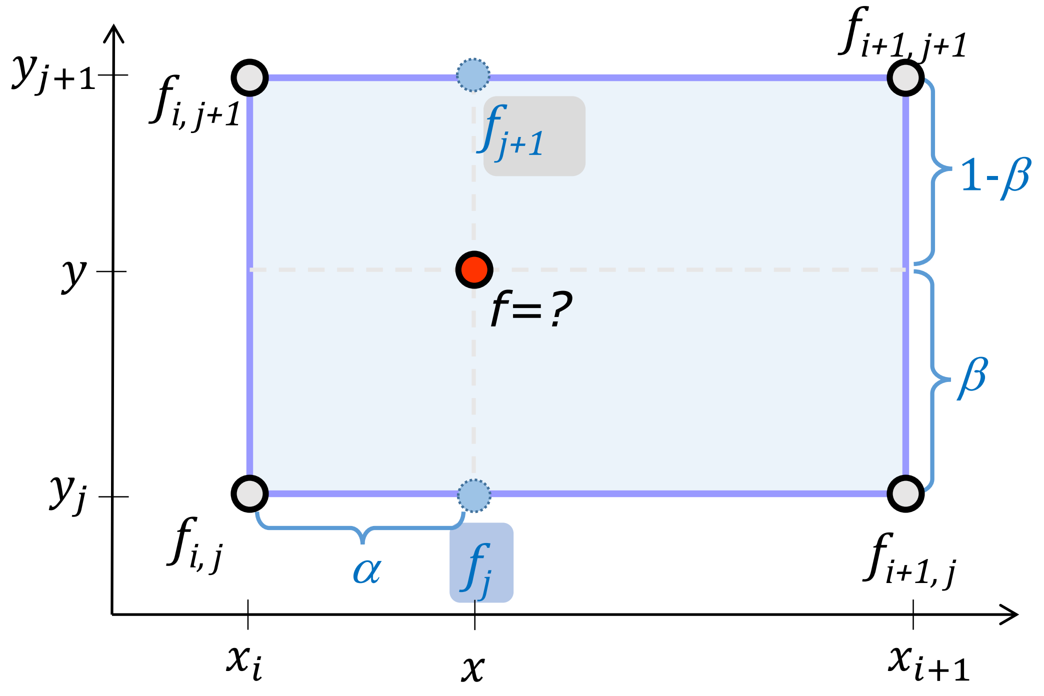
$$\alpha = \frac{x - x_i}{x_{i+1} - x_i}$$



Interpolation on grids

- Bilinear interpolation on a rectangle

$$f(\alpha,\beta) = (1-\beta)[(1-\alpha)\, f_{i,j} + \alpha\, f_{i+1,j}] + \beta\, [(1-\alpha)\, f_{i,j+1} + \alpha\, f_{i+1,j+1}]$$
$$= (1-\beta)\, f_j + \beta\, f_{j+1}$$



with

$$f_j = (1-\alpha)\, f_{i,j} + \alpha\, f_{i+1,j}$$
$$f_{j+1} = (1-\alpha)\, f_{i,j+1} + \alpha\, f_{i+1,j+1}$$

and local coordinates

$$\alpha = \frac{x - x_i}{x_{i+1} - x_i}, \qquad \beta = \frac{y - y_i}{y_{i+1} - y_i}, \qquad \alpha, \beta \in [0,1]$$

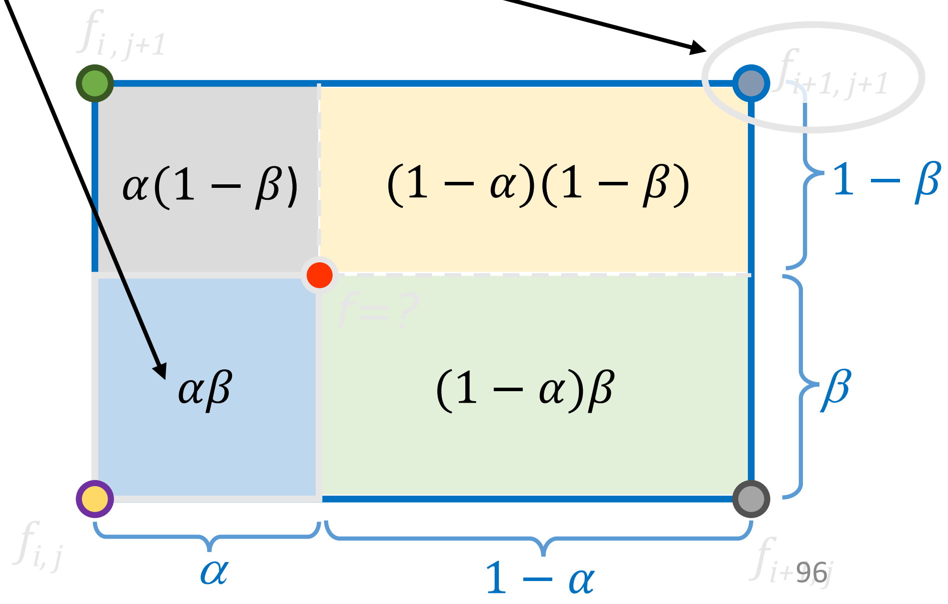
Interpolation on grids

- Geometric interpretation of bilinear interpolation

$$f(\alpha,\beta) = (1-\beta)[(1-\alpha)\, f_{i,j} + \alpha\, f_{i+1,j}] + \beta\, [(1-\alpha)\, f_{i,j+1} + \alpha\, f_{i+1,j+1}]$$

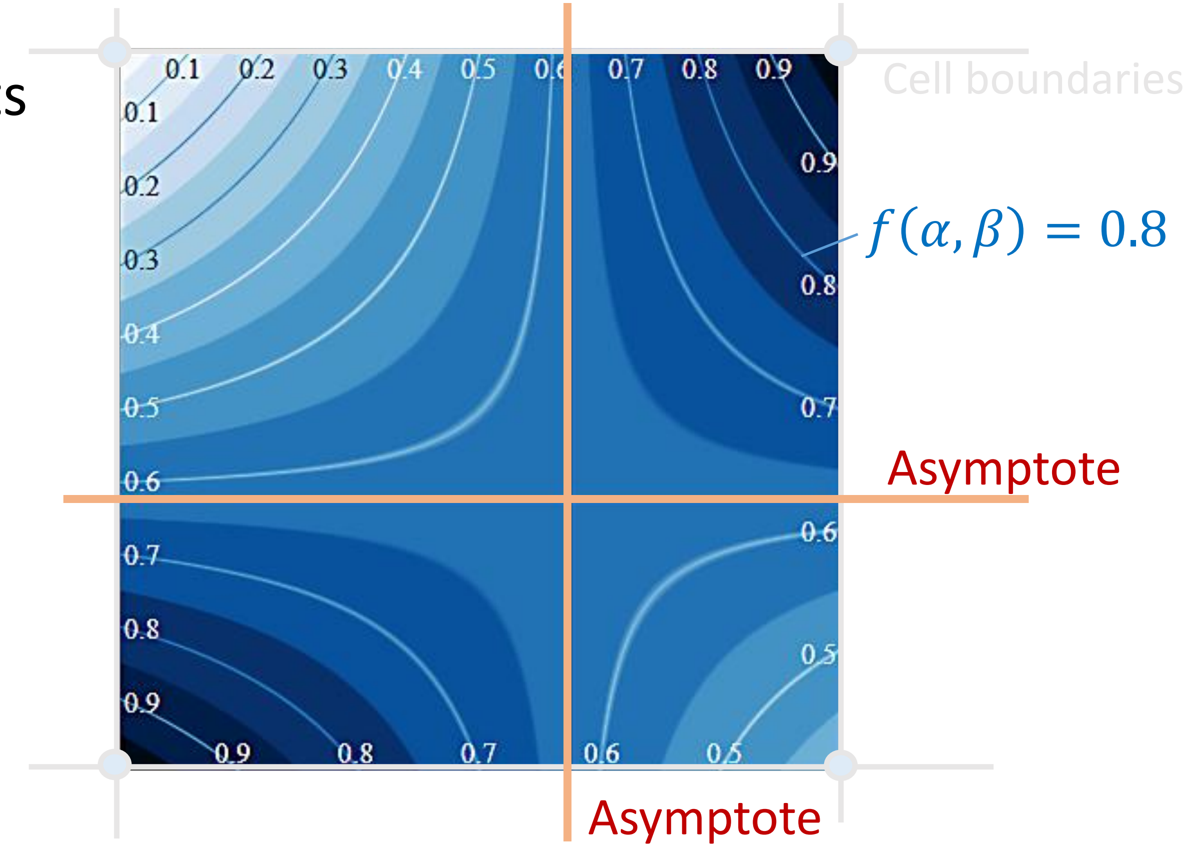
$$= (1-\alpha)(1-\beta)\, f_{i,j} + \alpha(1-\beta)\, f_{i+1,j} + (1-\alpha)\beta\, f_{i,j+1} + \alpha\beta\, f_{i+1,j+1}$$

  - Opposite points are weighted by local areas



Interpolation on grids

- When bilinear interpolation is used, isolines within a cell are hyperbolas
  - **Isoline:** curve on which all points have the same value



multivis.net/lecture/bilinear.html

Interpolation on grids

- How to evaluate the isolines?

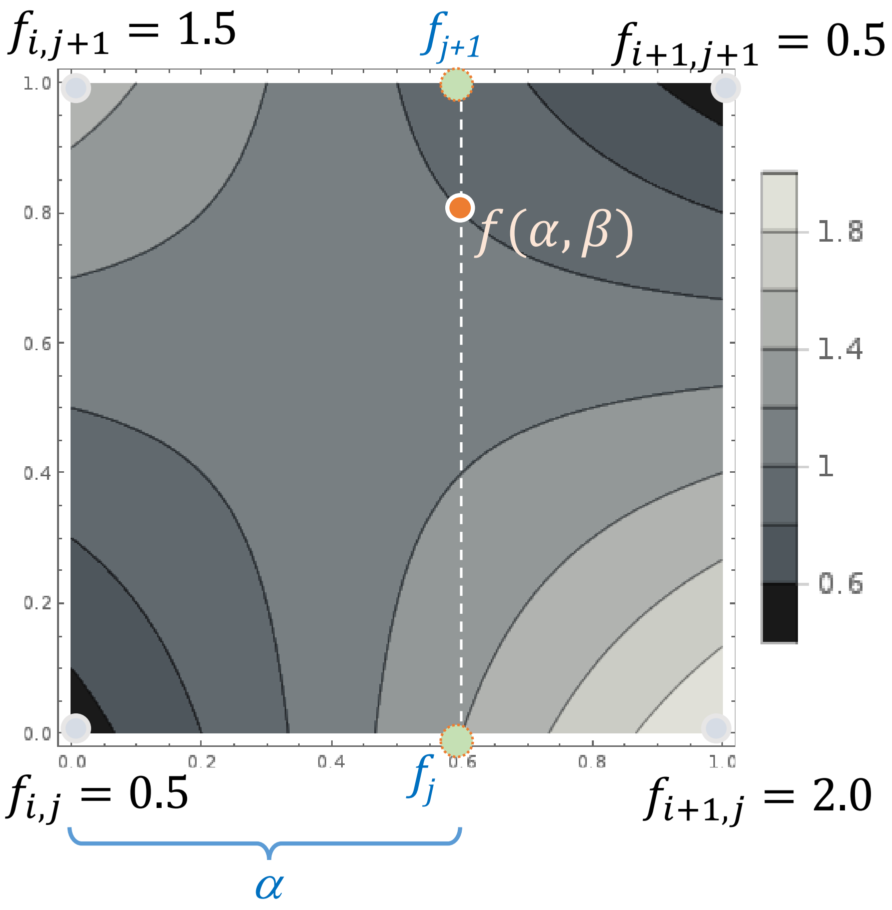
$$
\begin{aligned}
f_j &= \alpha\, f_{i+1,j} + (1-\alpha) f_{i,j} \\
&= f_{i,j} + \left(f_{i+1,j} - f_{i,j}\right)\alpha \\
&= 0.5 + 1.5\alpha
\end{aligned}
$$

$$
\begin{aligned}
f_{j+1} &= f_{i,j+1} + \left(f_{i+1,j+1} - f_{i,j+1}\right)\alpha \\
&= 1.5 - \alpha
\end{aligned}
$$

$$
\begin{aligned}
f(\alpha,\beta) &= f_j + \left(f_{j+1} - f_j\right)\beta \\
&= (0.5 + 1.5\alpha) + \\
&\quad \left(1.5 - \alpha - (0.5 + 1.5\alpha)\right)\beta
\end{aligned}
$$

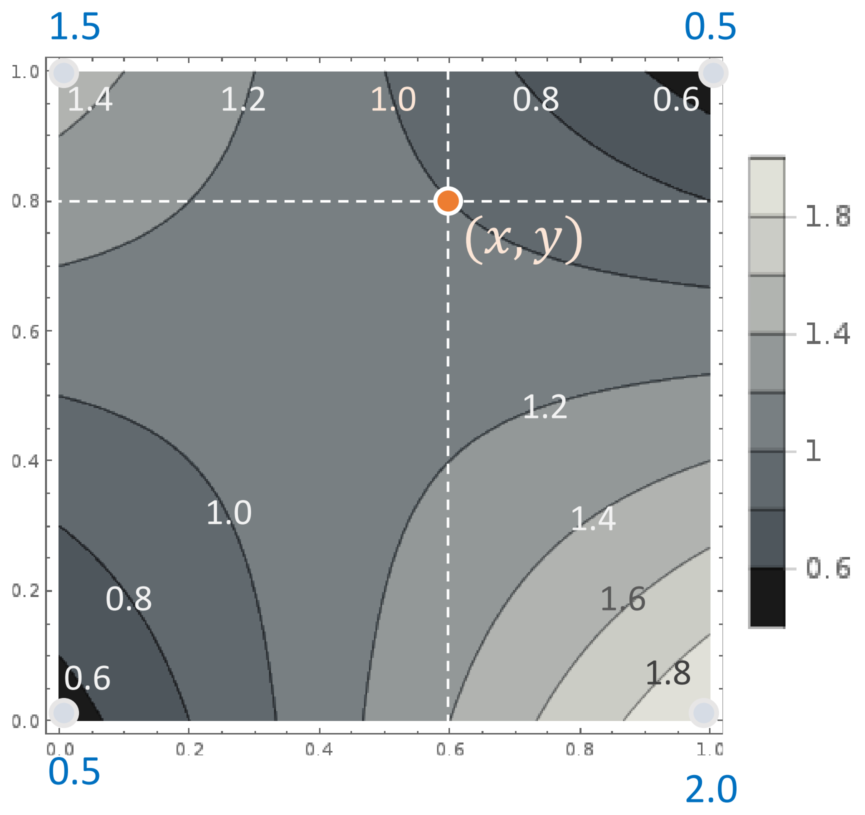
$$\boxed{f(\alpha,\beta) = 0.5 + 1.5\alpha + \beta - 2.5\alpha\beta}$$

Bi-linear interpolation function defining the scalar value at each point (α,β) within the cell



Interpolation on grids

- How to evaluate the isolines?
  - Compute y-coordinate of a point (x, y) with $x = 0.6$ which is on the iso-contour $f(x, y) = 1$



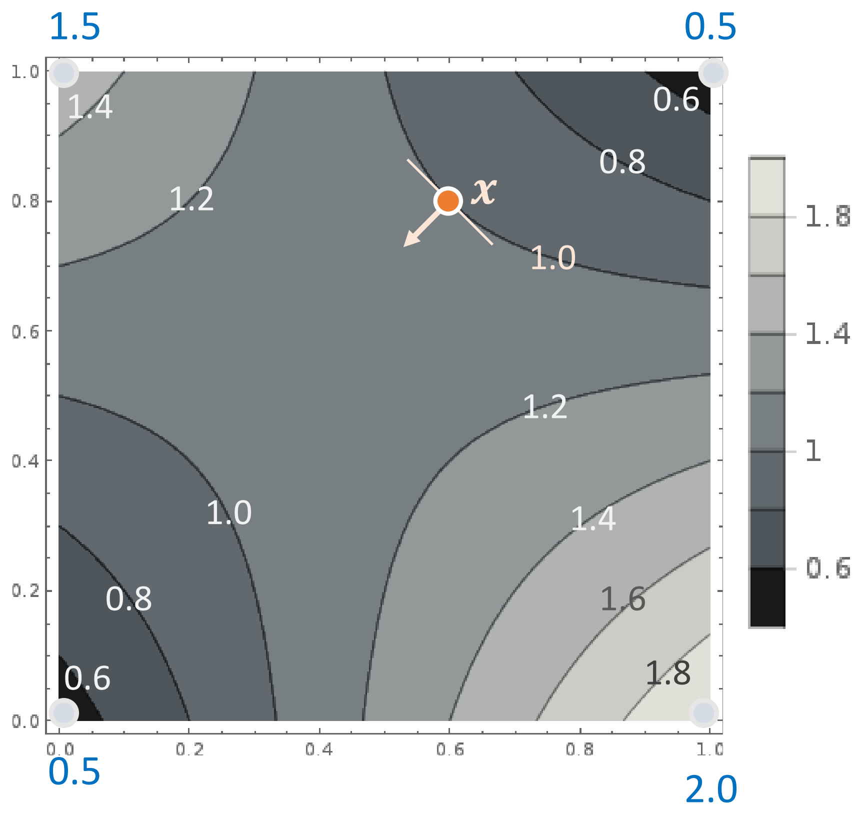
The coordinates of the point are $(0.6, 0.8)$

multivis.net/lecture/bilinear.html

Interpolation on grids

- What is the normal at a point $\boldsymbol{x}$ on an iso-surface?
  - It is the gradient at this point
  - Gradient points into direction of steepest ascent of f

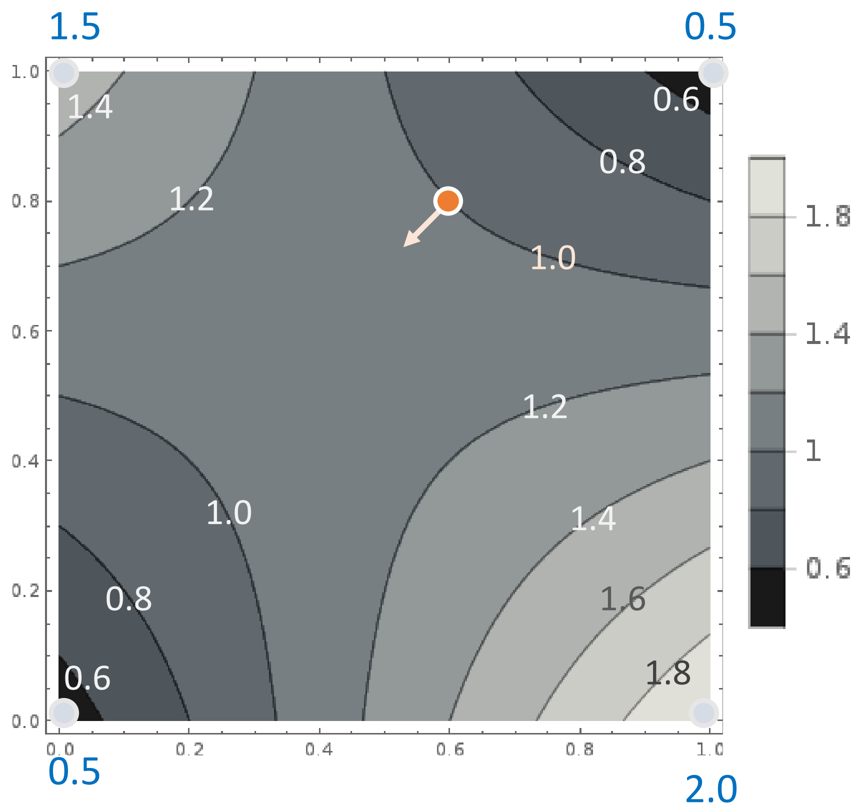
$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial}{\partial x} f(\boldsymbol{x}),\ \frac{\partial}{\partial y} f(\boldsymbol{x})\right)$$



multivis.net/lecture/bilinear.html

Interpolation on grids

- What is the normal at point $(0.6, 0.8)$ on the iso-surface?



Gradient at $(0.6, 0.8)$: $\begin{pmatrix} -0.5 \\ -0.5 \end{pmatrix}$

multivis.net/lecture/bilinear.html

Interpolation on grids

- How to evaluate the asymptotes?
  - Consider bilinear interpolation within a cell

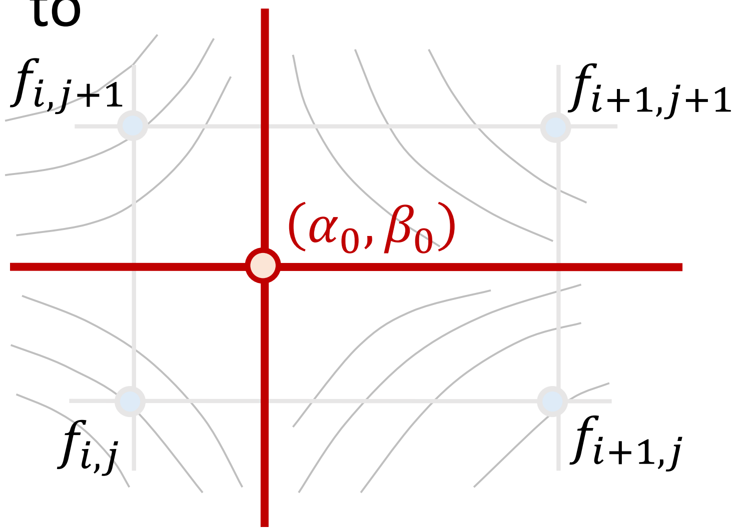
$$f(\alpha,\beta) = (1-\alpha)(1-\beta)f_{i,j} + \alpha(1-\beta)f_{i+1,j} + (1-\alpha)\beta f_{i,\,j+1} + \alpha\beta f_{\mathrm{i+1,\,j+1}}$$

  - Given the values at cell corners, transform f to

$$\boxed{f(\alpha,\beta) = \gamma(\alpha - \alpha_0)(\beta - \beta_0) + \delta}$$

Function of a hyperbola

  - δ is the function value at the intersection point (α_0, β_0) of the asymptotes



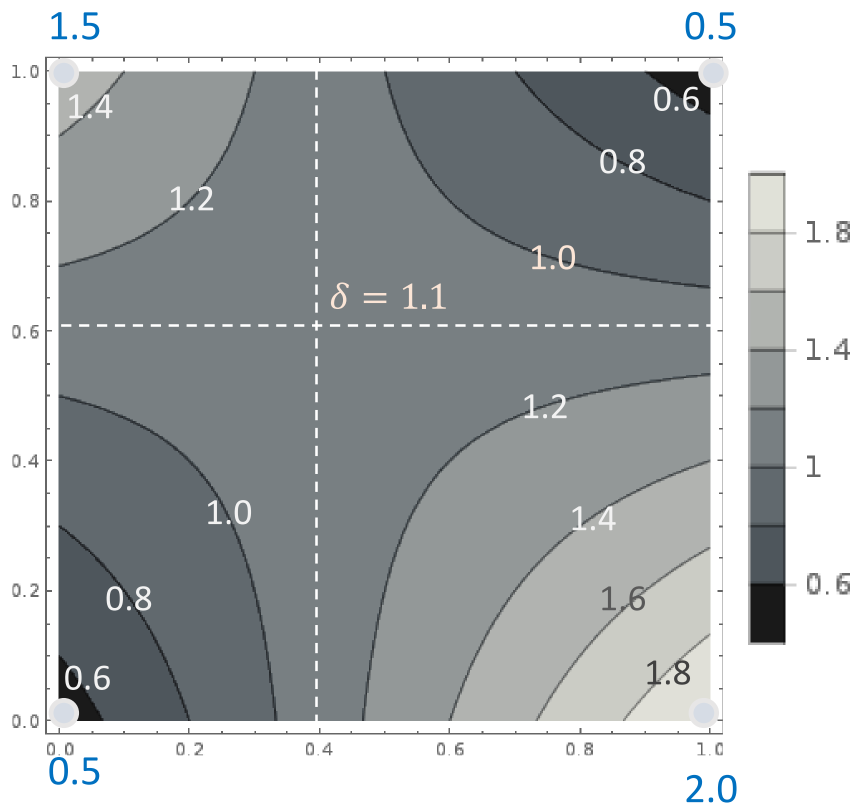
Interpolation on grids

- How to evaluate the asymptotes?

  Compute asymptotes from

  $f(\alpha, \beta) = 0.5 + 1.5\alpha + \beta - 2.5\alpha\beta$

  Get into form (hyperbola)



  - Asymptotes intersect at $(0.4, 0.6)$
  - Value at intersection: $\delta = 1.1$

Interpolation on grids

- **Summary:** Bilinear Interpolation
  - The value at each point (α, β) within the cell can be obtained by

$$f(\alpha,\beta) = (1-\alpha)(1-\beta)f_{00} + \alpha(1-\beta)f_{10} + (1-\alpha)\beta f_{01} + \alpha\beta f_{11}$$

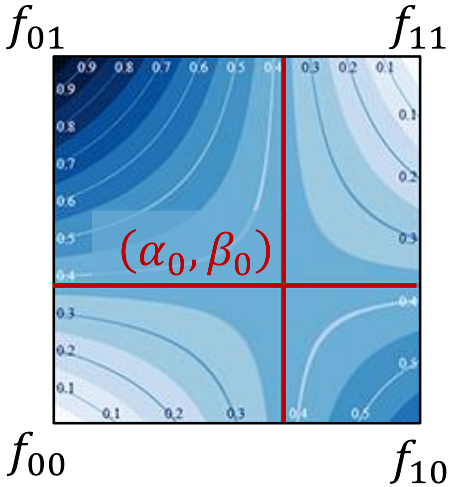
$$= A\alpha + B\beta + C\alpha\beta + D$$

where $A = f_{10} - f_{00}$, $B = f_{01} - f_{00}$,
$C = f_{00} - f_{01} - f_{10} + f_{11}$, $D = f_{00}$

  - We can evaluate the isoline for an iso-value c by setting $f(\alpha,\beta) = c$
  - The asymptotes of the hyperbolas can be computed by

$$f(\alpha,\beta) = \gamma(\alpha-\alpha_0)(\beta-\beta_0) + \delta$$

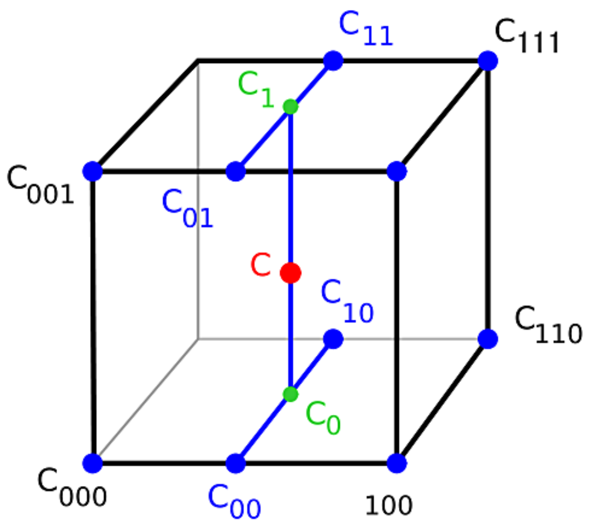
where $\gamma = C$ and $\delta = (f_{00}\, f_{11} - f_{01}\, f_{10})/C$ is the value at the intersection point (α_0, β_0) of the asymptotes with $\alpha_0 = -B/C$ and $\beta_0 = -A/C$



multivis.net/lecture/bilinear.html

Interpolation on grids

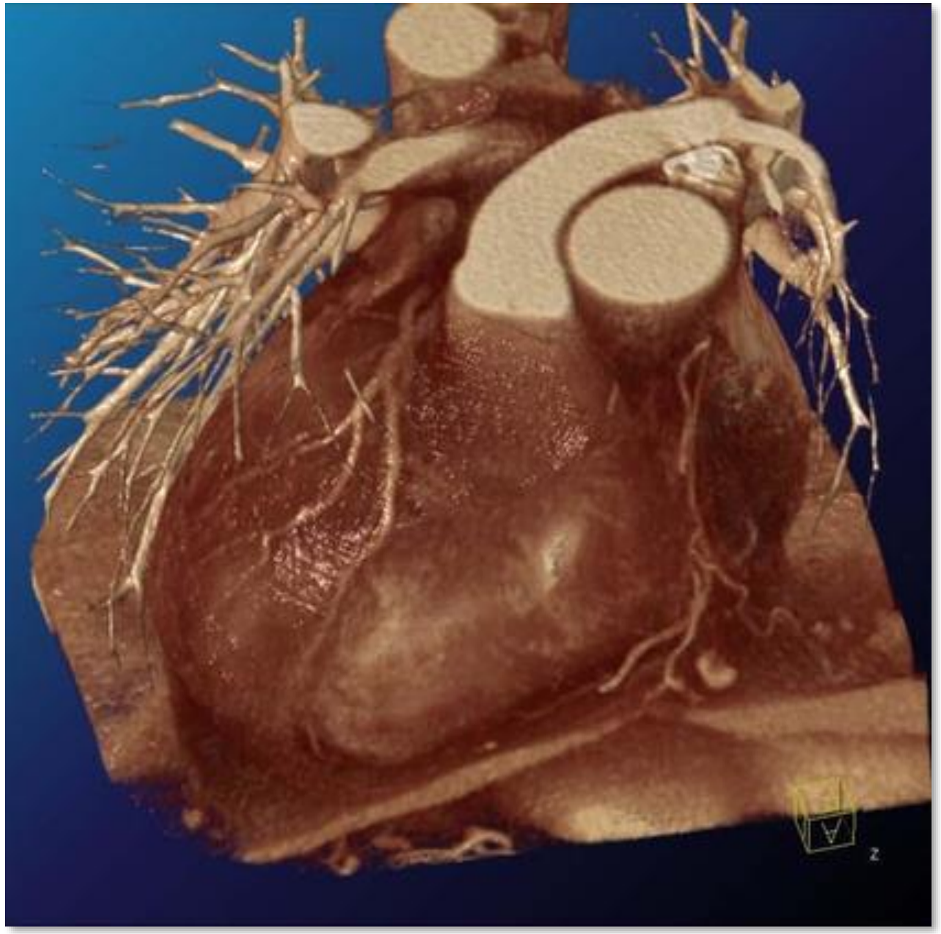
- In 3D we use trilinear interpolation:



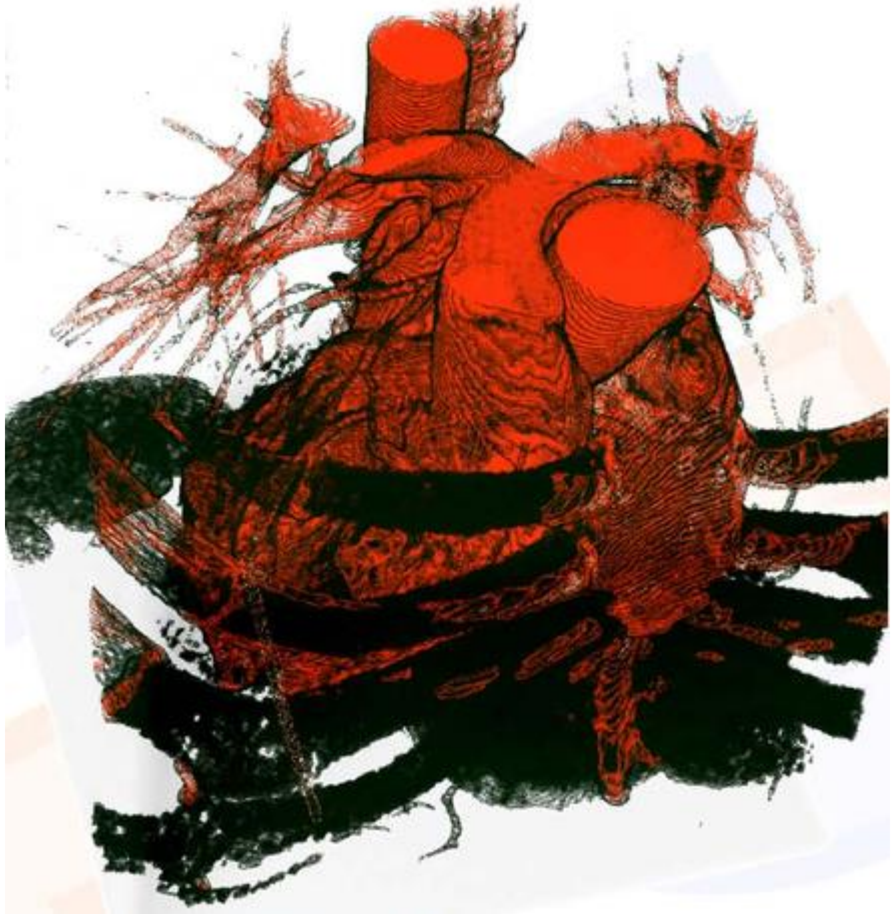
  - Apply linear interpolation of the initial data along the edges to obtain C_{00} , C_{01} , C_{10} , C_{11}
  - Interpolate linearly between C_{00} and C_{01} , and between C_{10} and C_{11} to obtain C_0 and C_1
  - Finally, interpolate between C_1 and C_0 to obtain C.

Interpolation on grids

- This is what we want…

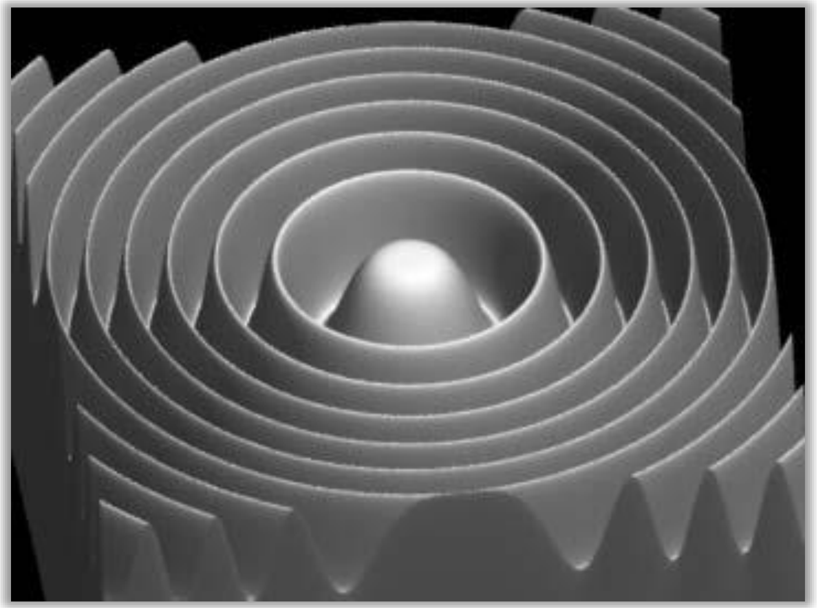


…but sometimes this is what we get



Interpolation on grids

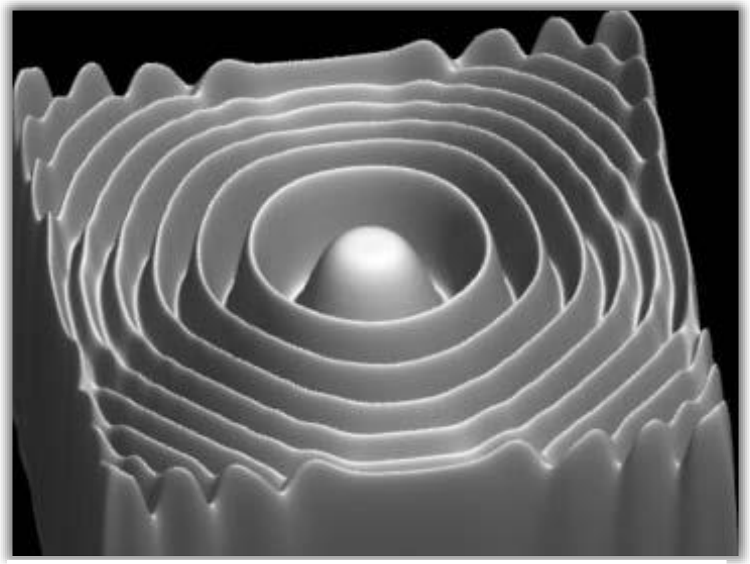
- Higher-order reconstruction/interpolation
  - Required if very high quality is needed
  - Usually tested on Marschner-Lobb function
    - High amount of its energy is near its Nyquist frequency
    - Very demanding test for accurate reconstruction



[Marschner & Lobb 94]

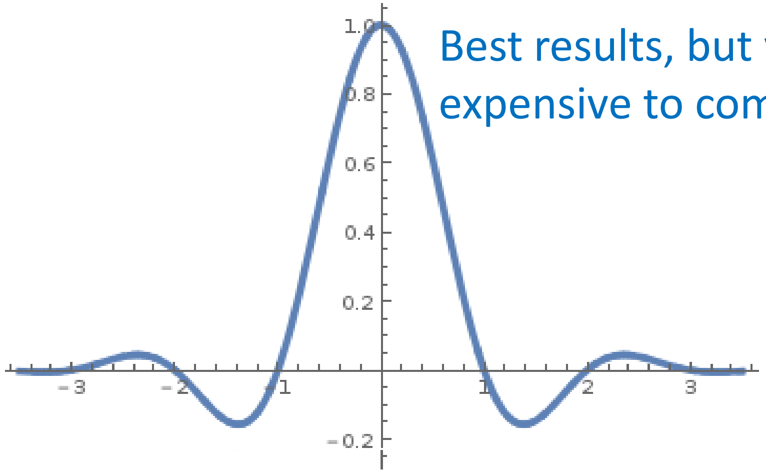
High–quality Reconstruction

Sinc function



[Marschner & Lobb 94]

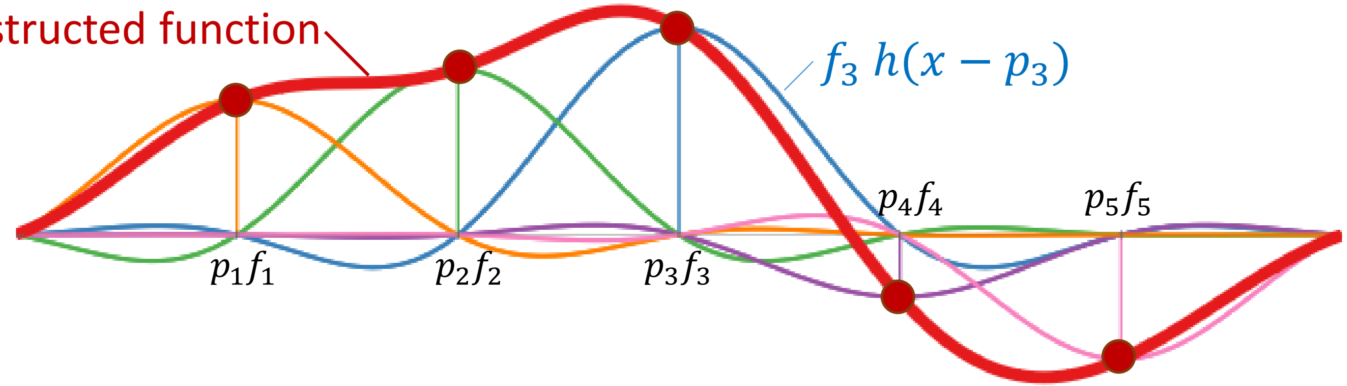
Windowed sinc (Hann window)



Best results, but very expensive to compute

"Optimal" reconstruction filter

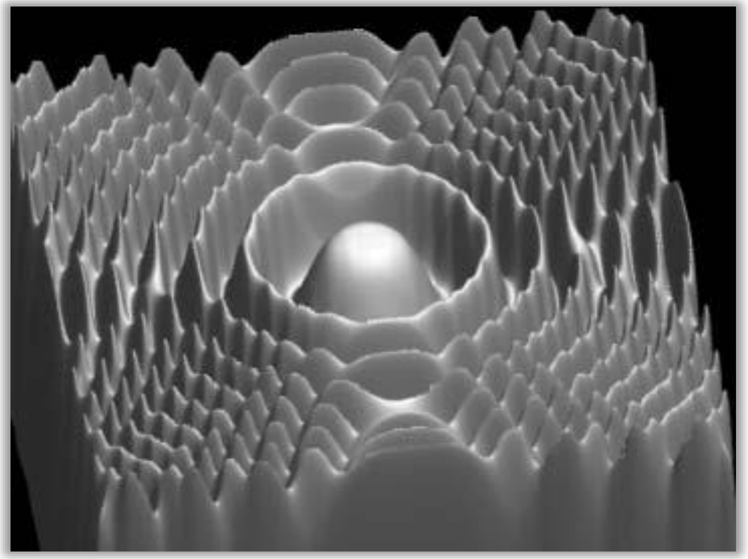
Reconstructed function



multivis.net/lecture/bilinear.html

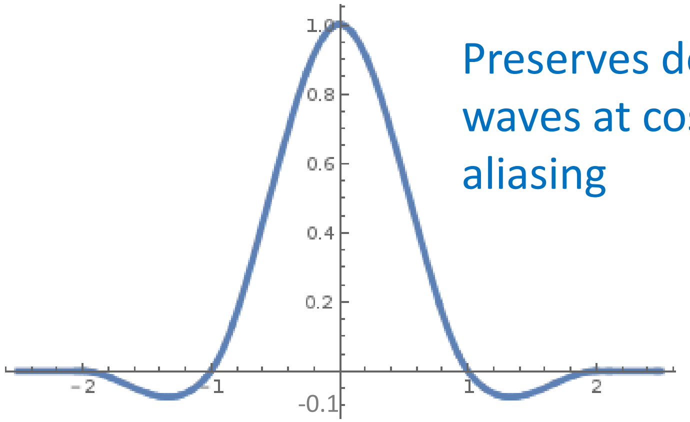
High–quality Reconstruction

Bicubic interpolation
(Catmull-Rom spline)



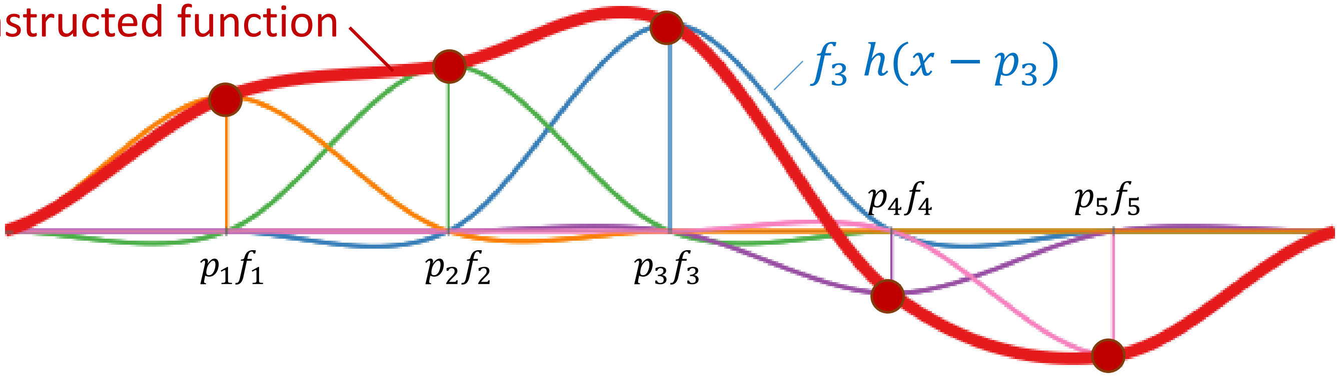
[Marschner & Lobb 94]

Reconstructed Marschner-Lobb function



Preserves depth of waves at cost of aliasing

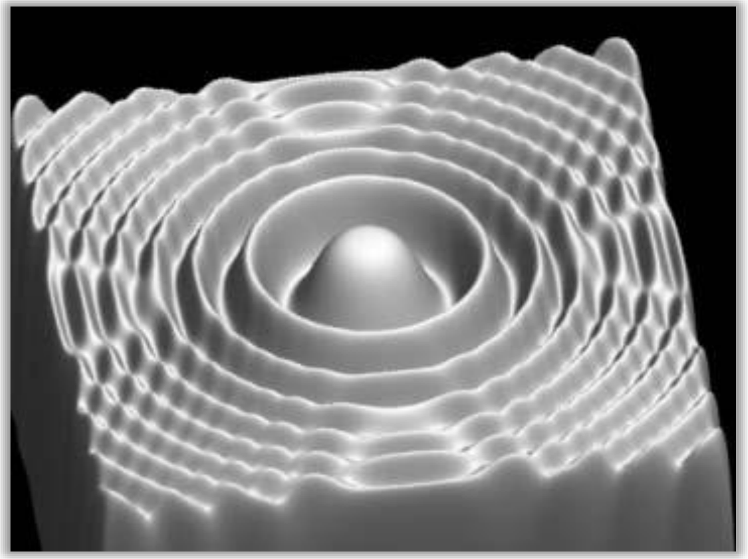
Interpolation: 1 at center and 0 at integers



multivis.net/lecture/bilinear.html

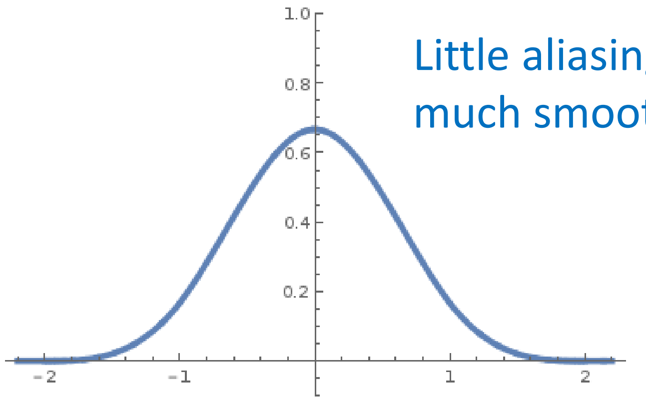
High–quality Reconstruction

Cubic B-spline
(with smoothing)



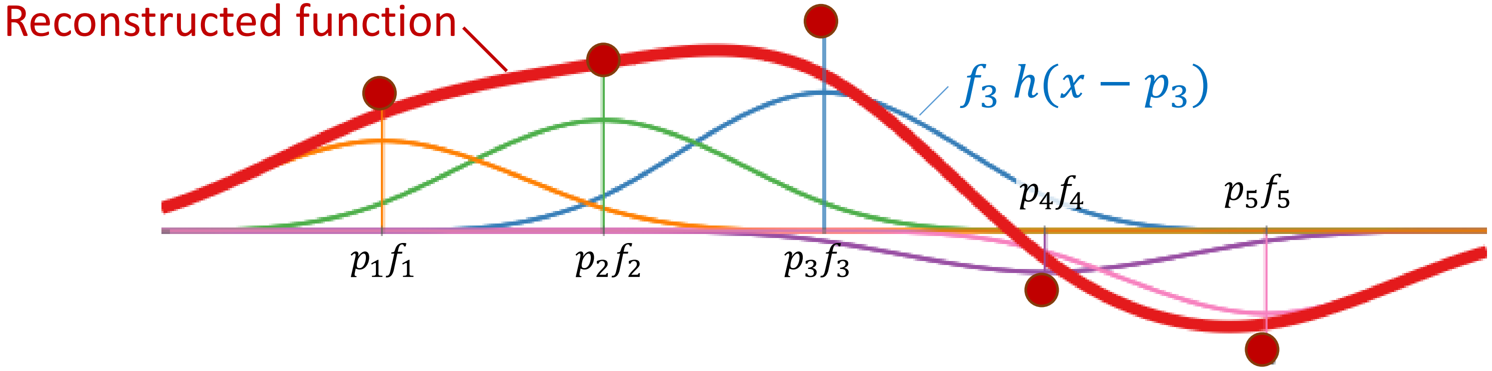
[Marschner & Lobb 94]

Reconstructed Marschner-Lobb function



Little aliasing, but much smoothing

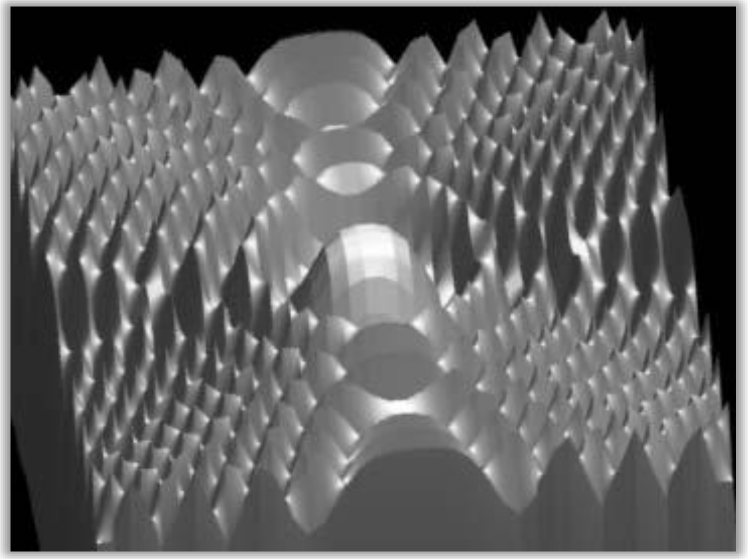
Smoothing: $^2/_3$ at center and $^1/_6$ at ± 1



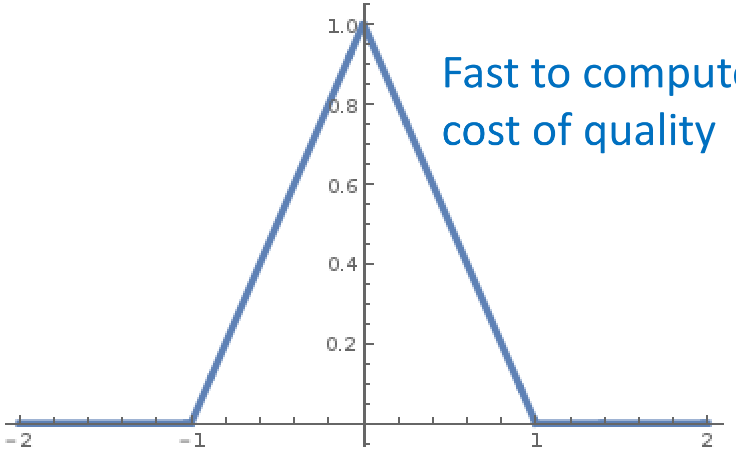
multivis.net/lecture/interpolation.html

High–quality Reconstruction

Trilinear interpolation (Tent)



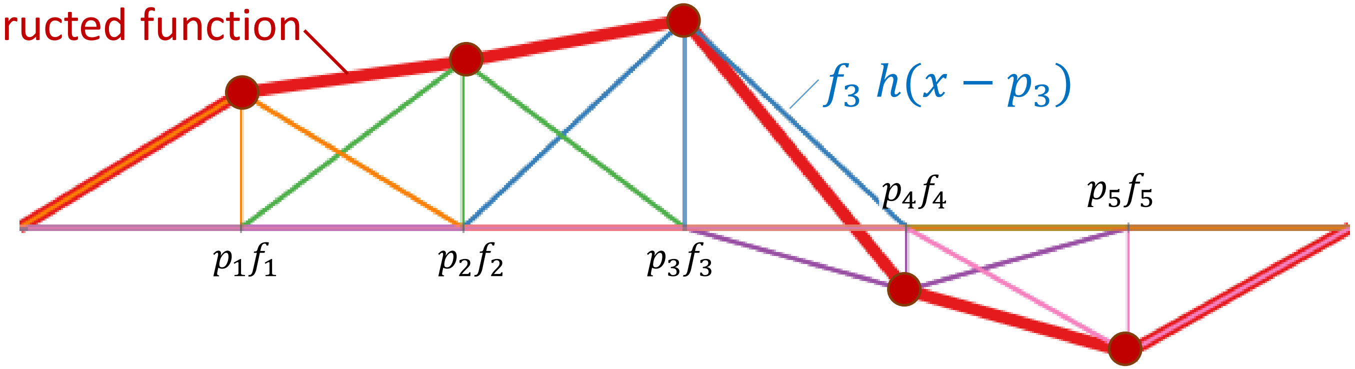
Reconstructed Marschner-Lobb function



Fast to compute at cost of quality

Interpolation: 1 at center and 0 at integers

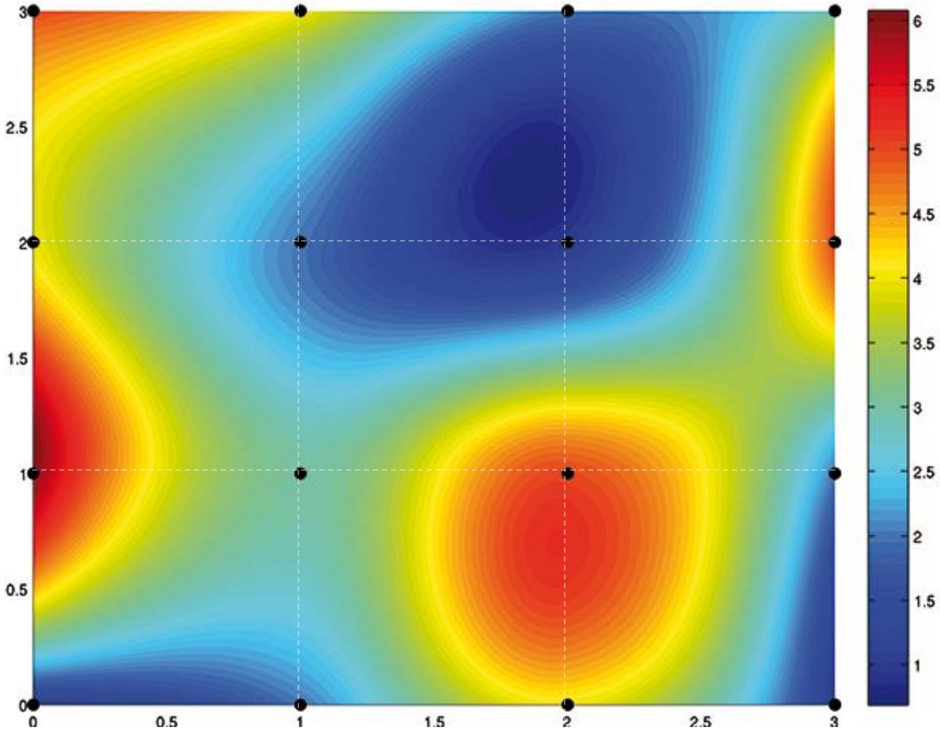
Reconstructed function



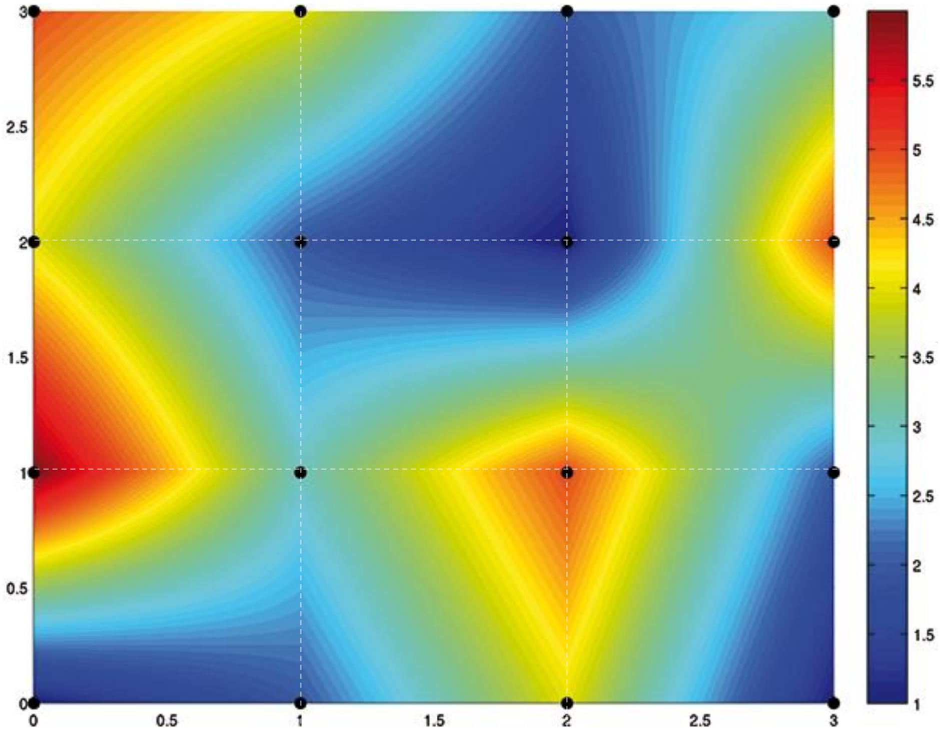
multivis.net/lecture/interpolation.html

Interpolation on grids

- Volume rendering with different reconstruction filters



Bicubic interpolation (default in Photoshop)
C^1 continuous → tangents match
at segment transition

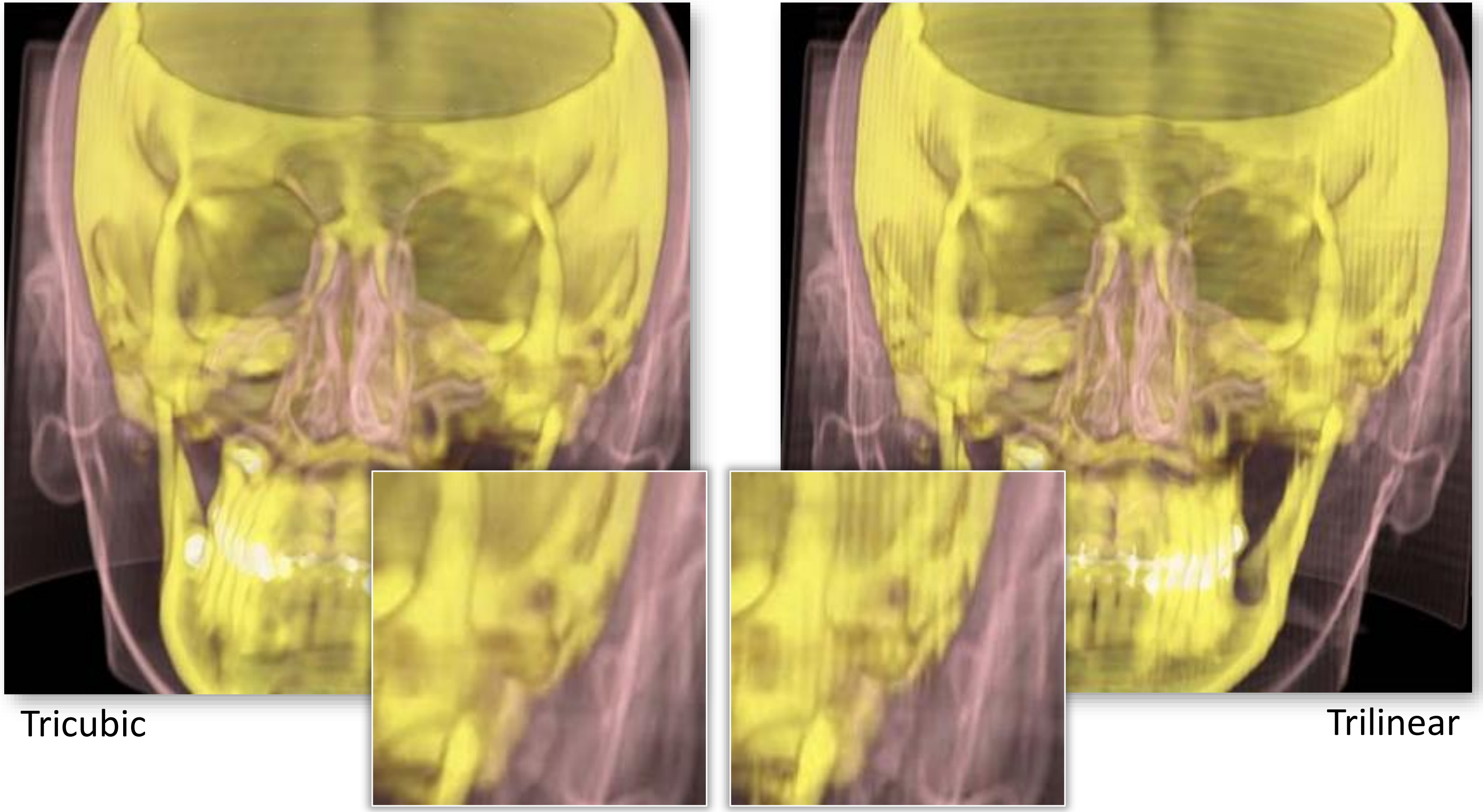


Bilinear interpolation
(C^0 continuous → values match
at segment transition)

[Wikipedia]

Interpolation on grids

- Volume rendering with different reconstruction filters



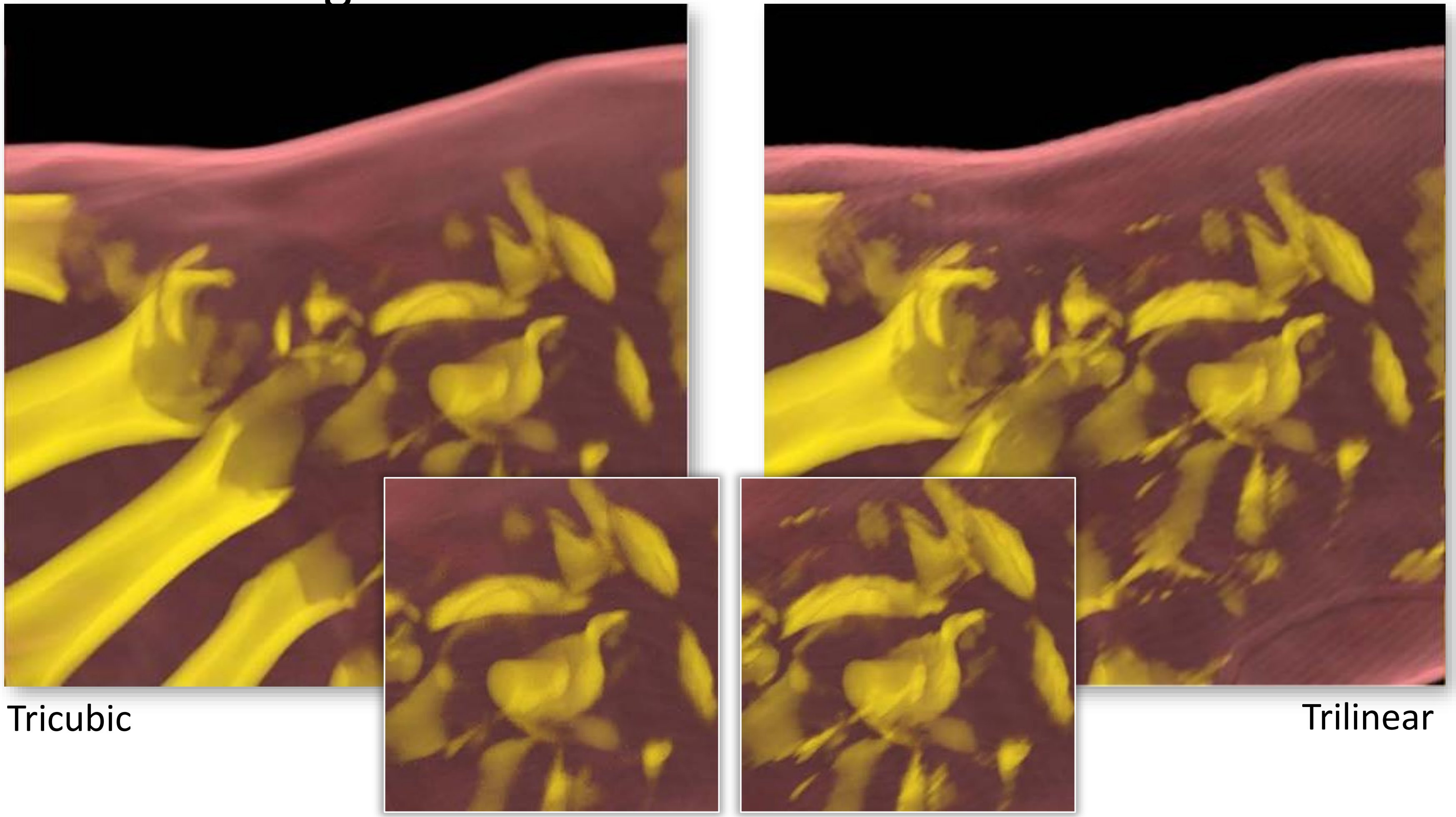
Tricubic

Trilinear

[Engel et al. 06]

Interpolation on grids

- Volume rendering with different reconstruction filters



Tricubic

Trilinear

[Engel et al. 06]

Questions???