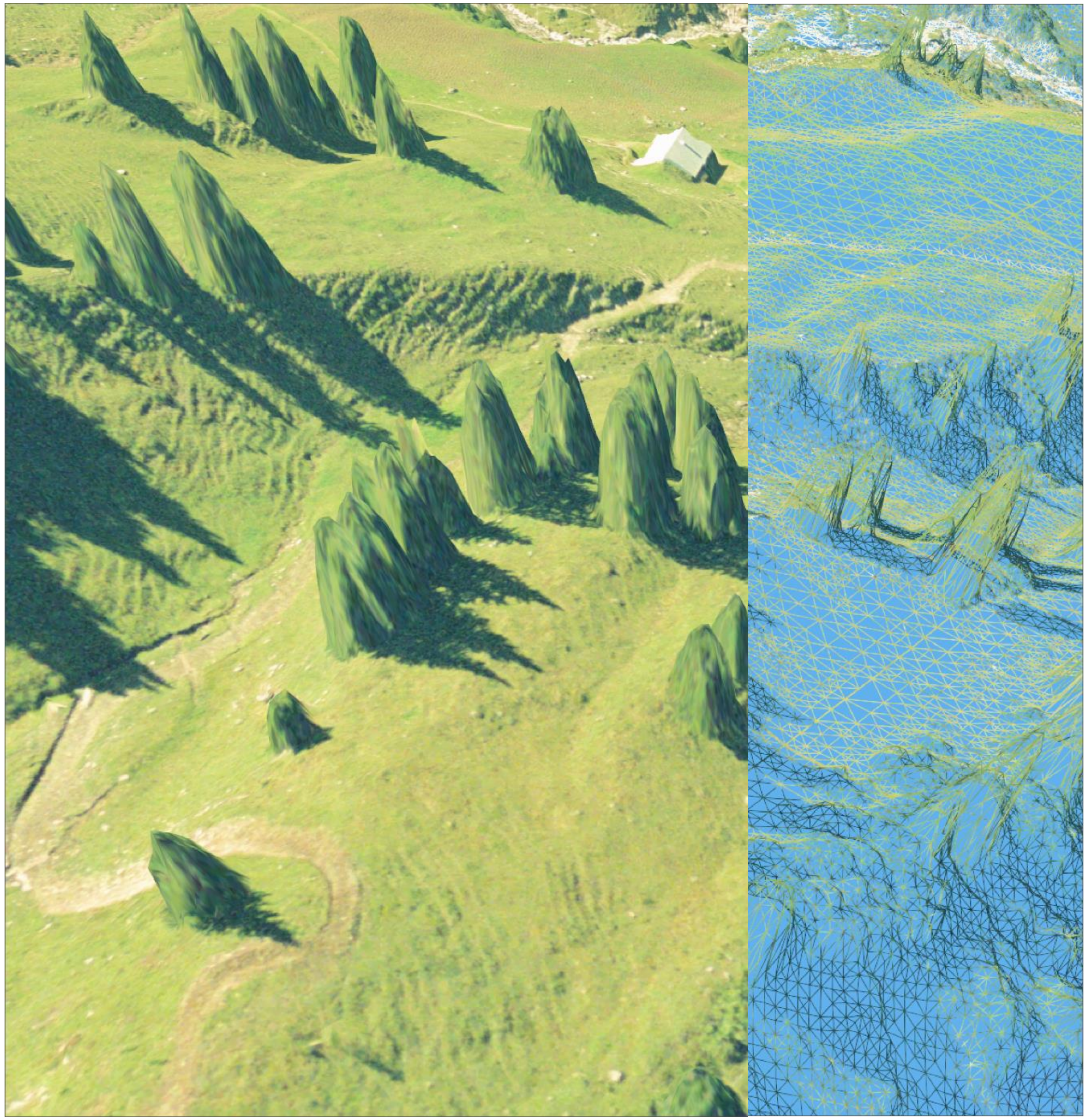
Visualization, DD2257

Prof. Dr. Tino Weinkauf

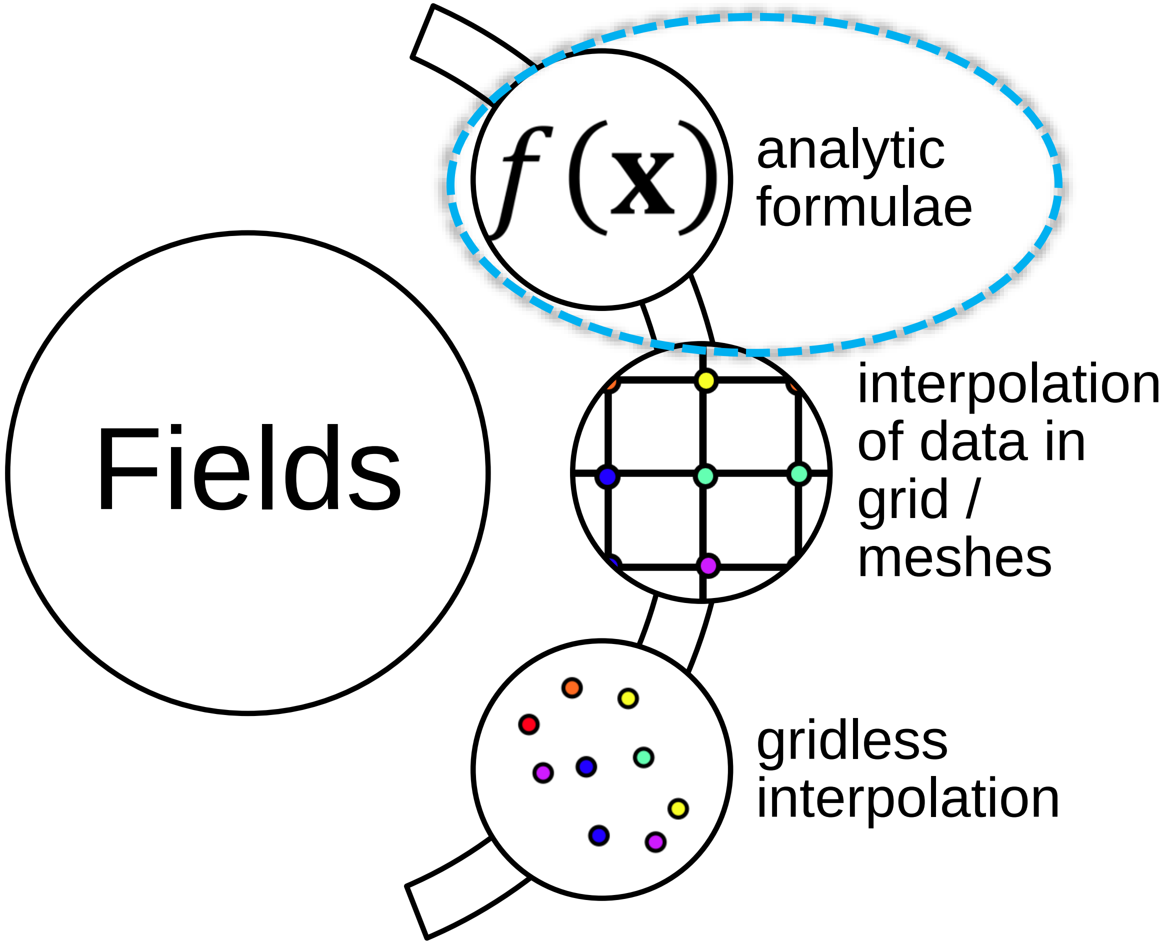
Data Description

Continuous Data



Fields

continuous representation of a variable



Fields

analytic formulae

function in 1D:

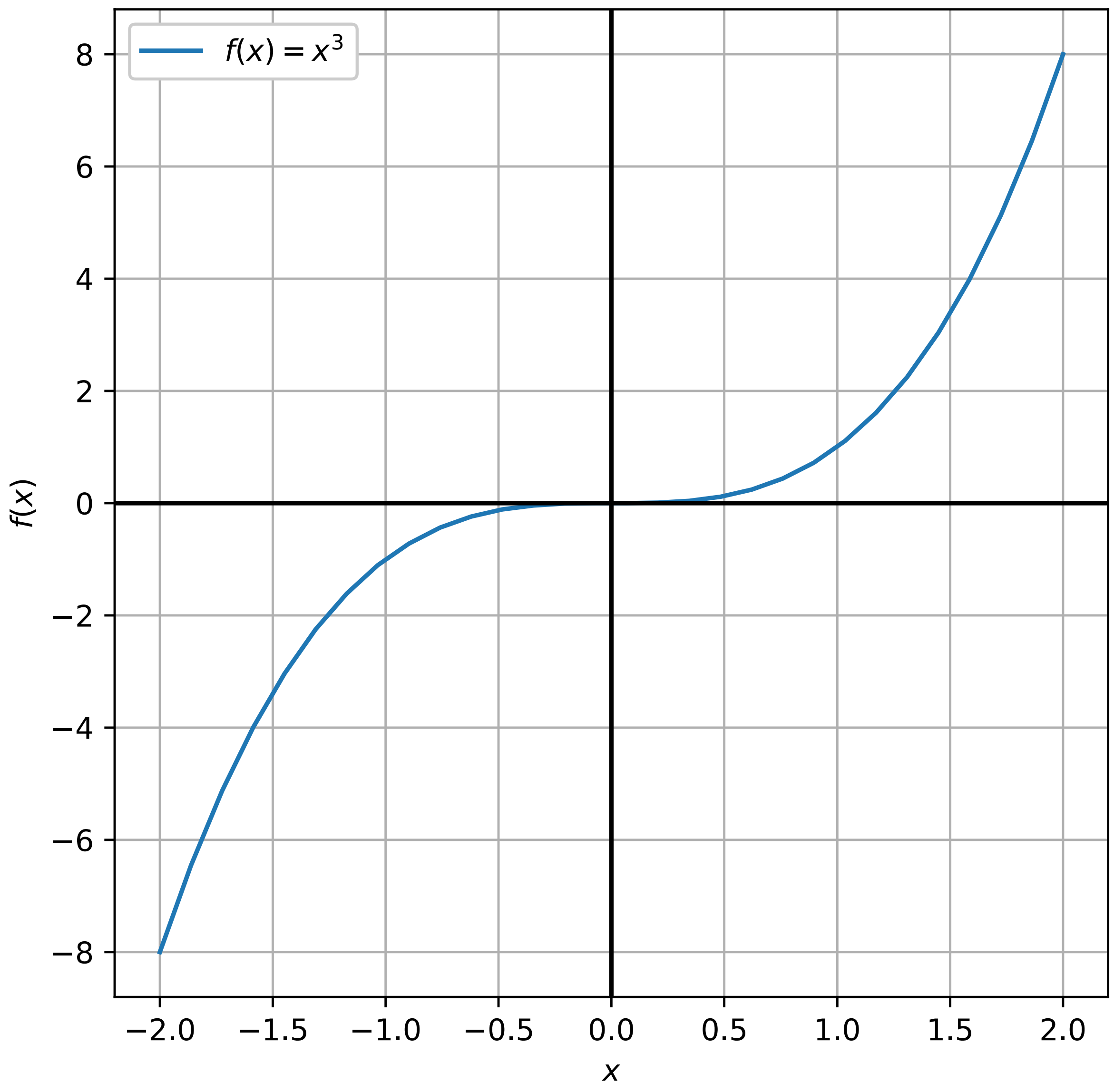
$$f(x) = x^3$$

function in 2D:

$$f(x, y) = x^2 + xy$$

function in 3D:

$$f(x, y, z) = 3x + \frac{xy}{z + 1}$$



Fields

$$f(x) = x^3$$

$$f(x, y) = x^2 + xy$$

$$f(x, y, z) = 3x + \frac{xy}{z+1}$$

$$f(\mathbf{x}) = \cdots$$

$$\mathbf{x} \in E^n$$

observation space can be 2D, 3D, …
easier to describe with a single, bold $\mathbf{x}$

scalar field

$$s : \mathbb{E}^n \to \mathbb{R}$$

$$s(\mathbf{x})$$
$$\text{with } \mathbf{x} \in \mathbb{E}^n$$

$$s(x, y) = 2xy + 4y^2$$

2D scalar field

vector field

$$\mathbf{v} : \mathbb{E}^n \to \mathbb{R}^m$$

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} c_1(\mathbf{x}) \\ \vdots \\ c_m(\mathbf{x}) \end{pmatrix}$$
$$\text{with } \mathbf{x} \in \mathbb{E}^n$$

$$\mathbf{v}(x, y) = \begin{pmatrix} u(x, y) \\ v(x, y) \end{pmatrix}$$

2D vector field

tensor field

$$\mathbf{T} : \mathbb{E}^n \to \mathbb{R}^{m \times b}$$

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} c_{11}(\mathbf{x}) & \dots & c_{1b}(\mathbf{x}) \\ \vdots & & \vdots \\ c_{m1}(\mathbf{x}) & \dots & c_{mb}(\mathbf{x}) \end{pmatrix}$$
$$\text{with } \mathbf{x} \in \mathbb{E}^n$$

$$\mathbf{T}(x, y) = \begin{pmatrix} a(x, y) & b(x, y) \\ c(x, y) & d(x, y) \end{pmatrix}$$

2D tensor field

scalar field

$$s : \mathbb{E}^n \rightarrow \mathbb{R}$$

$$s(\mathbf{x})$$
$$\text{with } \mathbf{x} \in \mathbb{E}^n$$

$$s(x, y) = 2xy + 4y^2$$

2D scalar field

vector field

$$\mathbf{v} : \mathbb{E}^n \rightarrow \mathbb{R}^m$$

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} c_1(\mathbf{x}) \\ \vdots \\ c_m(\mathbf{x}) \end{pmatrix}$$
$$\text{with } \mathbf{x} \in \mathbb{E}^n$$

$$\mathbf{v}(x, y) = \begin{pmatrix} 2x - y \\ 2y \end{pmatrix}$$

2D vector field

tensor field

$$\mathbf{T} : \mathbb{E}^n \rightarrow \mathbb{R}^{m \times b}$$

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} c_{11}(\mathbf{x}) & \dots & c_{1b}(\mathbf{x}) \\ \vdots & & \vdots \\ c_{m1}(\mathbf{x}) & \dots & c_{mb}(\mathbf{x}) \end{pmatrix}$$
$$\text{with } \mathbf{x} \in \mathbb{E}^n$$

$$\mathbf{T}(x, y) = \begin{pmatrix} 2x & 1 \\ x + y & -y \end{pmatrix}$$

2D tensor field

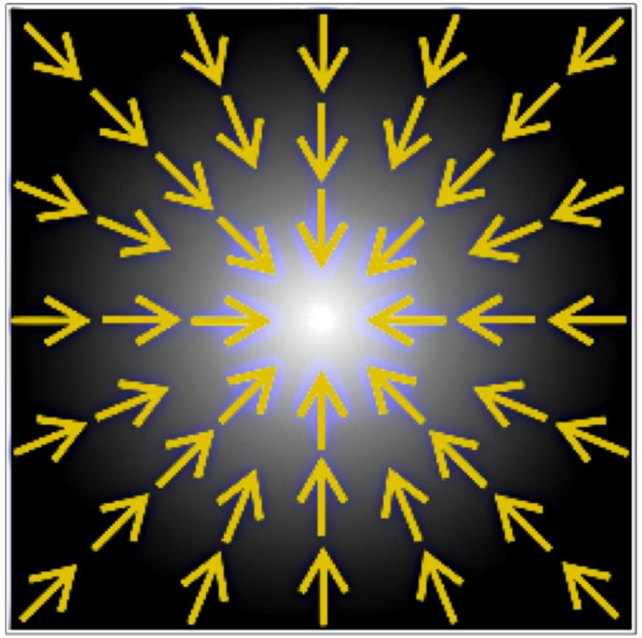
Derivatives

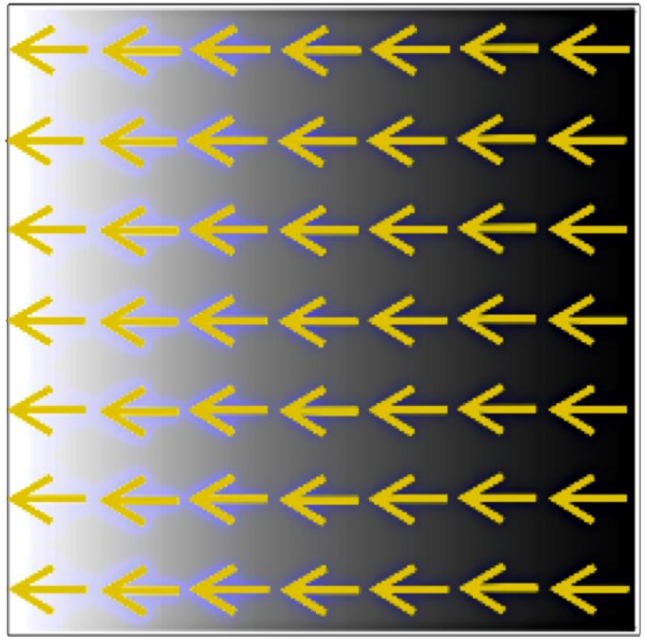
many applications

normal for volume rendering

critical point classification for vector field topology

In scalar fields: describes direction of steepest ascend





scalar field

$$s : \mathbb{E}^n \rightarrow \mathbb{R}$$

vector field

$$\mathbf{v} : \mathbb{E}^n \rightarrow \mathbb{R}^m$$

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} c_1(\mathbf{x}) \\ \vdots \\ c_m(\mathbf{x}) \end{pmatrix}$$

with $\mathbf{x} \in \mathbb{E}^n$

tensor field

$$\mathbf{T} : \mathbb{E}^n \rightarrow \mathbb{R}^{m \times b}$$

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} c_{11}(\mathbf{x}) & \dots & c_{1b}(\mathbf{x}) \\ \vdots & & \vdots \\ c_{m1}(\mathbf{x}) & \dots & c_{mb}(\mathbf{x}) \end{pmatrix}$$

with $\mathbf{x} \in \mathbb{E}^n$

The first derivative of a scalar field is a vector field called **gradient**.
It consists of the partial derivatives of the scalar function $s(\mathbf{x})$ for each dimension of the observation space.

$$s(x, y) \dashrightarrow \nabla s(x, y) = \begin{pmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial y} \end{pmatrix} = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$$

2D scalar field

gradient

scalar field	vector field	tensor field
$s : \mathbb{E}^n \rightarrow \mathbb{R}$	$\mathbf{v} : \mathbb{E}^n \rightarrow \mathbb{R}^m$	$\mathbf{T} : \mathbb{E}^n \rightarrow \mathbb{R}^{m \times b}$
$s(\mathbf{x})$ with $\mathbf{x} \in \mathbb{E}^n$	$\begin{pmatrix} c_1(\mathbf{x}) \\ \ldots \end{pmatrix}$	$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} c_{11}(\mathbf{x}) & \ldots & c_{1b}(\mathbf{x}) \\ \vdots & & \vdots \\ c_{m1}(\mathbf{x}) & \ldots & c_{mb}(\mathbf{x}) \end{pmatrix}$ with $\mathbf{x} \in \mathbb{E}^n$

The second derivative of a scalar field is a tensor field called **Hessian**. It consists of the partial derivatives of $s(\mathbf{x})$ derived twice for each dimension of the observation space.

$s(x, y)$	$\nabla s(x, y) = \begin{pmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial y} \end{pmatrix} = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$	$\nabla^2 s(x, y) = \begin{pmatrix} s_{xx} & s_{xy} \\ s_{yx} & s_{yy} \end{pmatrix}$
2D scalar field	gradient	Hessian

scalar field

$$s : \mathbb{E}^n \to \mathbb{R}$$

$$s(\mathbf{x})$$
with $\mathbf{x} \in \mathbb{E}^n$

vector field

$$\mathbf{v} : \mathbb{E}^n \to \mathbb{R}^m$$

$$\begin{pmatrix} c_1(\mathbf{x}) \end{pmatrix}$$

The first derivative of a vector field is a tensor field called **Jacobian**. It consists of the partial derivatives of $\mathbf{v}(\mathbf{x})$ for each dimension of the observation space.

$$\mathbf{v}(x, y) = \begin{pmatrix} u(x, y) \\ v(x, y) \end{pmatrix}$$

2D vector field

tensor field

$$\mathbf{T} : \mathbb{E}^n \to \mathbb{R}^{m \times b}$$

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} c_{11}(\mathbf{x}) & \dots & c_{1b}(\mathbf{x}) \\ \vdots & & \vdots \\ c_{m1}(\mathbf{x}) & \dots & c_{mb}(\mathbf{x}) \end{pmatrix}$$
with $\mathbf{x} \in \mathbb{E}^n$

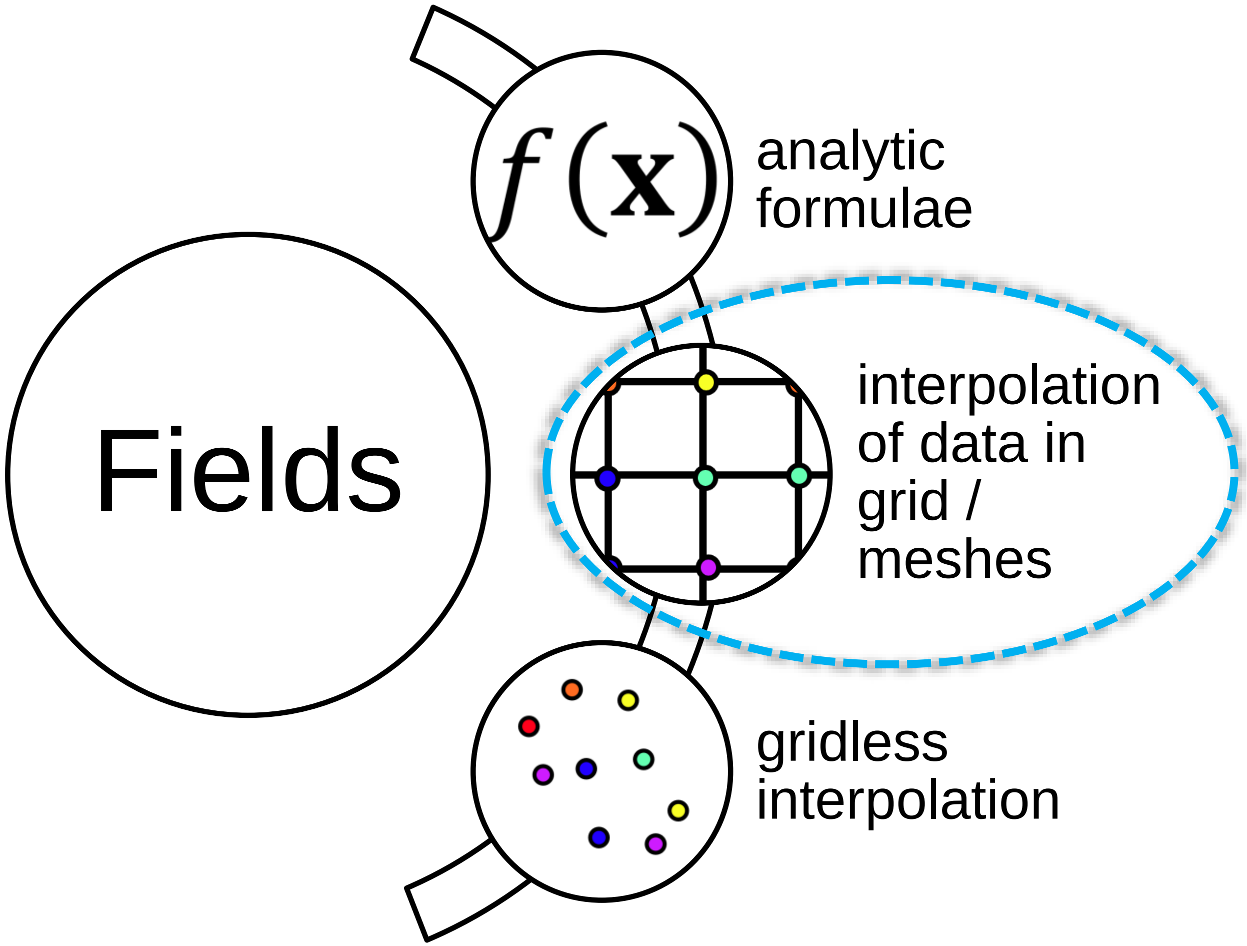
$$\nabla \mathbf{v}(x, y) = \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix}$$

Jacobian

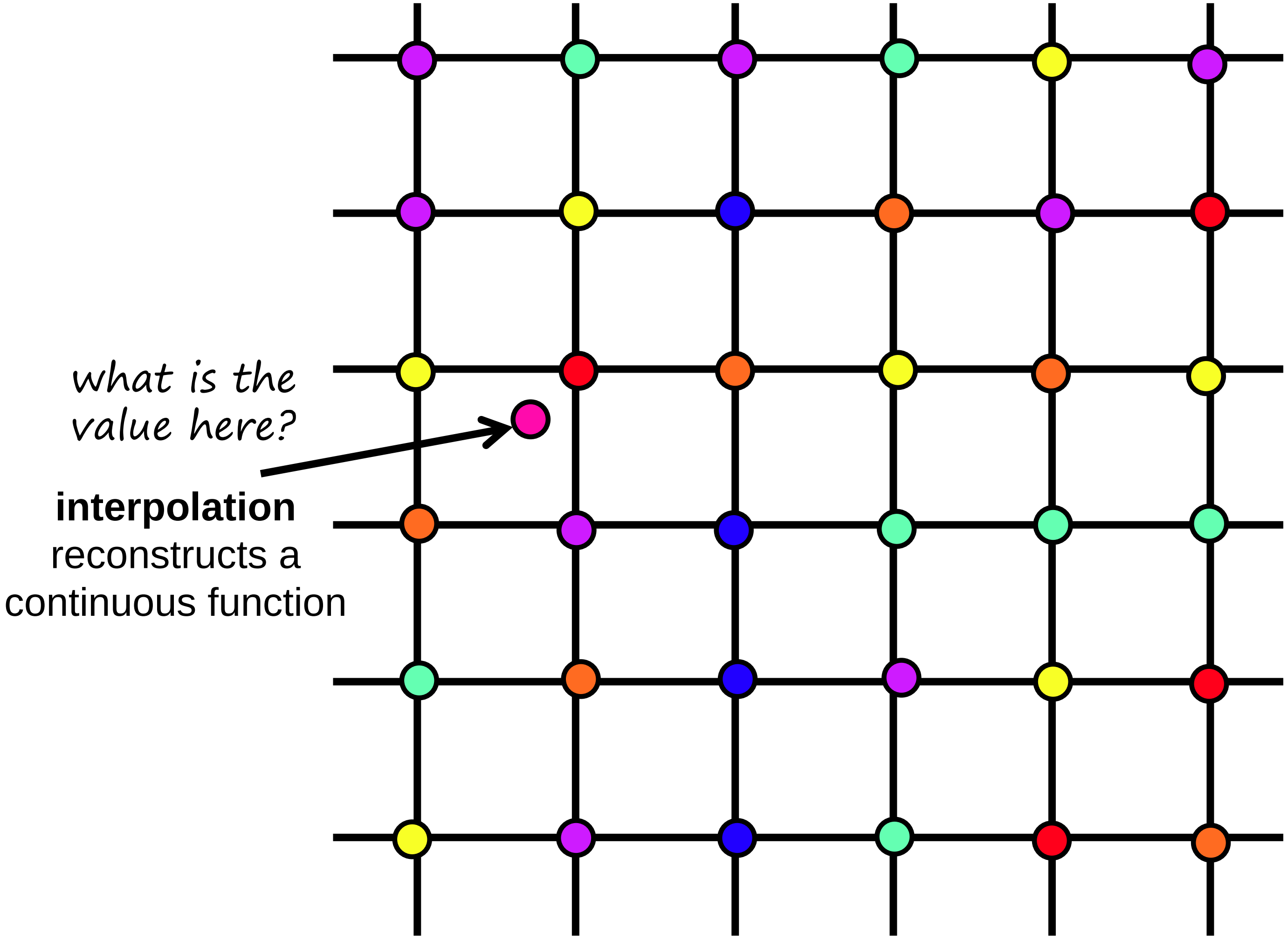
Fields

continuous representation of a variable

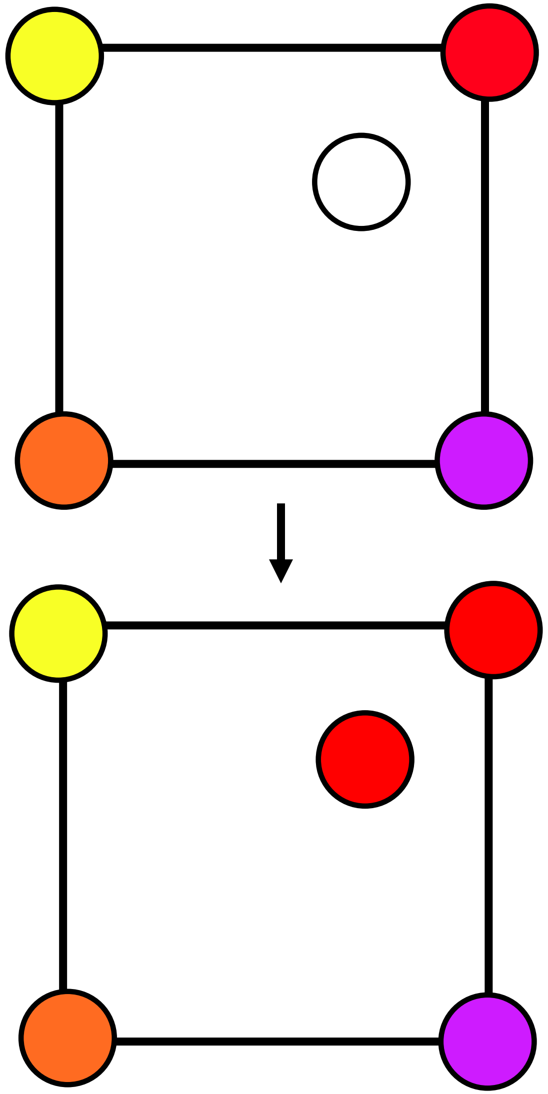
sampled data:
interpolation formulae used to create a continuous representation

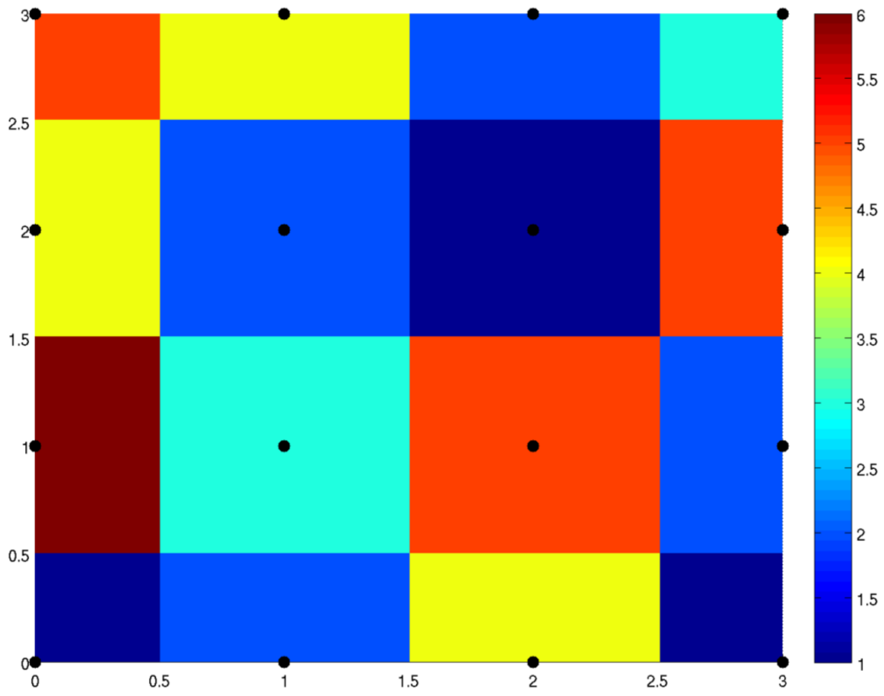


- A grid consists of a finite number of **samples**
  - The continuous signal is known only at a few points (**data points**)
  - In general, data is needed in between these points
- By **interpolation** we obtain a representation that matches the values at the data points
  - **Reconstruction** at any other point possible

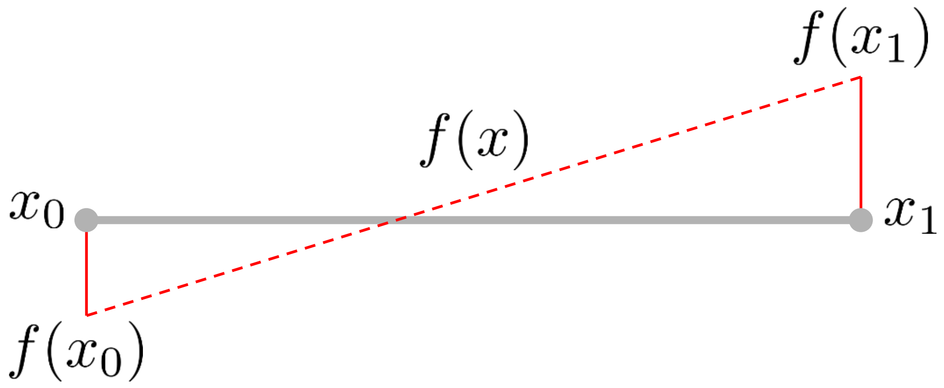


- Simplest approach: **Nearest-Neighbor Interpolation**
  - Assign the value of the nearest grid point to the sample.





- **Linear Interpolation** (in 1D domain)
  - Domain points x, scalar function $f(x)$



General:

$$f(x) = \frac{x_1 - x}{x_1 - x_0} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \qquad x \in [x_0, x_1]$$

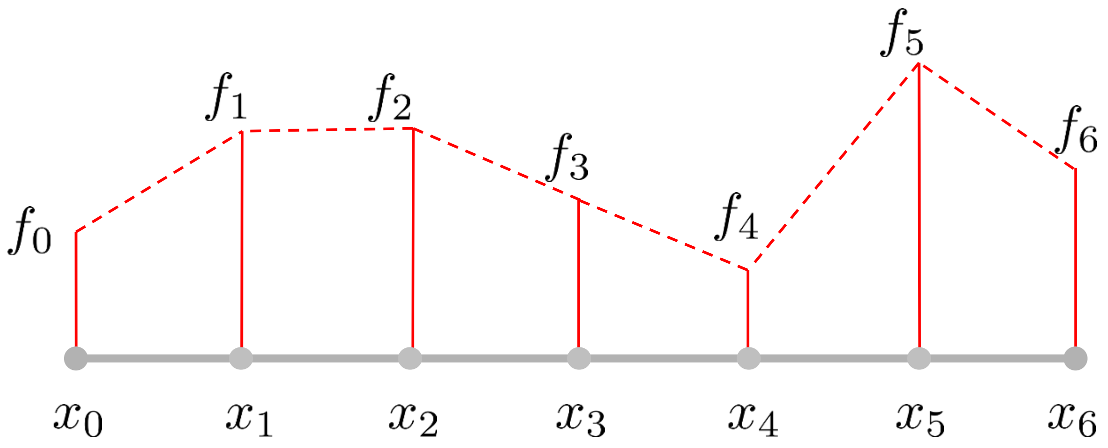
Special Case:

$$f(x) = (1-x)\ f(0) + x\, f(1) \qquad\qquad x \in [0,1]$$

$$= \begin{bmatrix}(1-x) & x\end{bmatrix} \begin{pmatrix} f(0) \\ f(1) \end{pmatrix} = \begin{bmatrix}1 & x\end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{pmatrix} f(0) \\ f(1) \end{pmatrix}$$

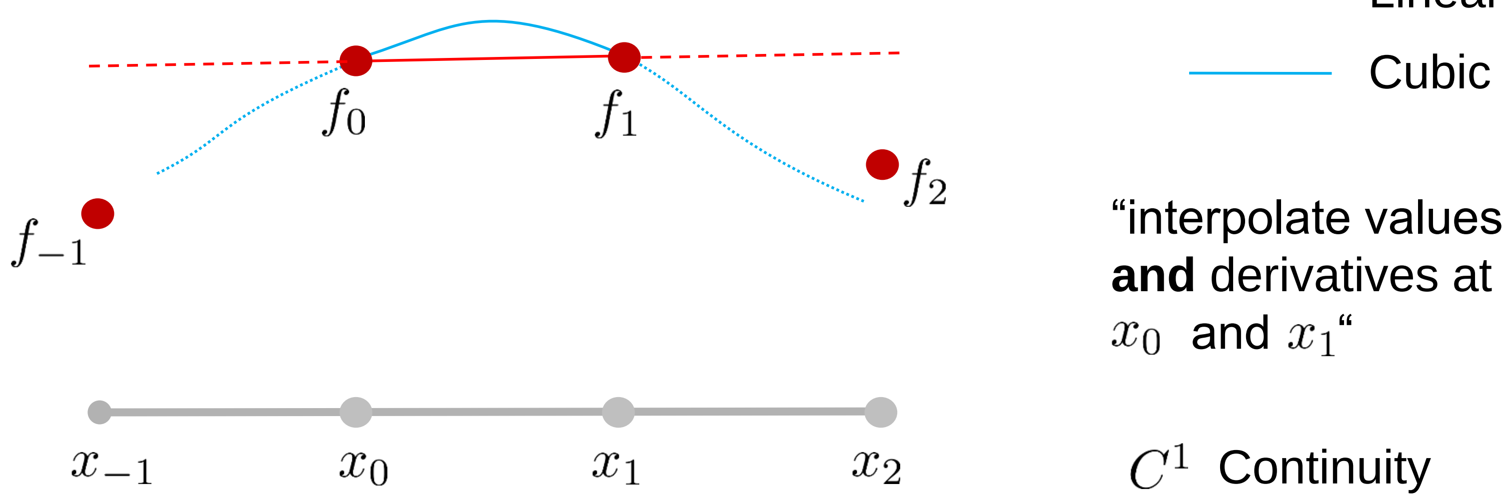
Basis Coefficients

- **Linear Interpolation** (in 1D domain)
  - Sample values $f_i := f(x_i)$



- C^0 Continuity (discontinuous first derivative)
  - Use higher order interpolation for smoother transition, e.g., **cubic** interpolation

- **Cubic Hermite Interpolation** (in equidistant 1D domain)
  - $x_{i+1} = x_i + 1$



"interpolate values **and** derivatives at x_0 and x_1"

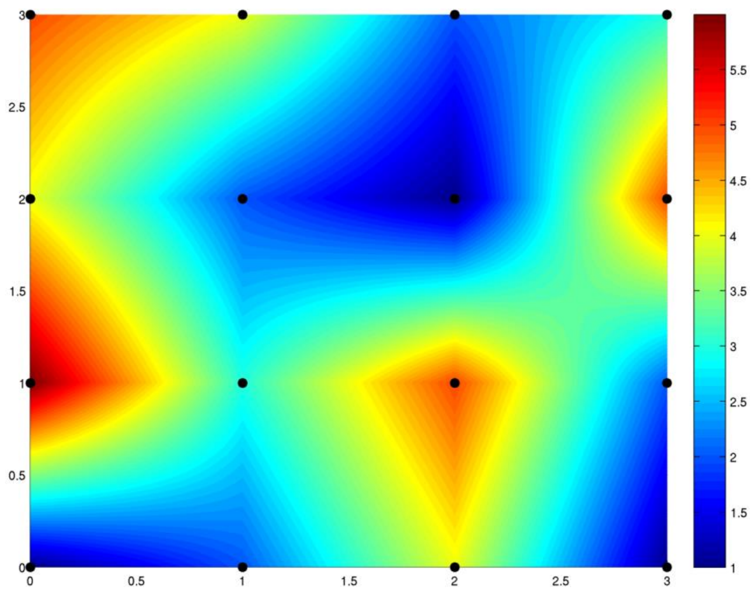
C^1 Continuity (discontinuous second derivatives)

Linear $$f(x) = \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix}$$

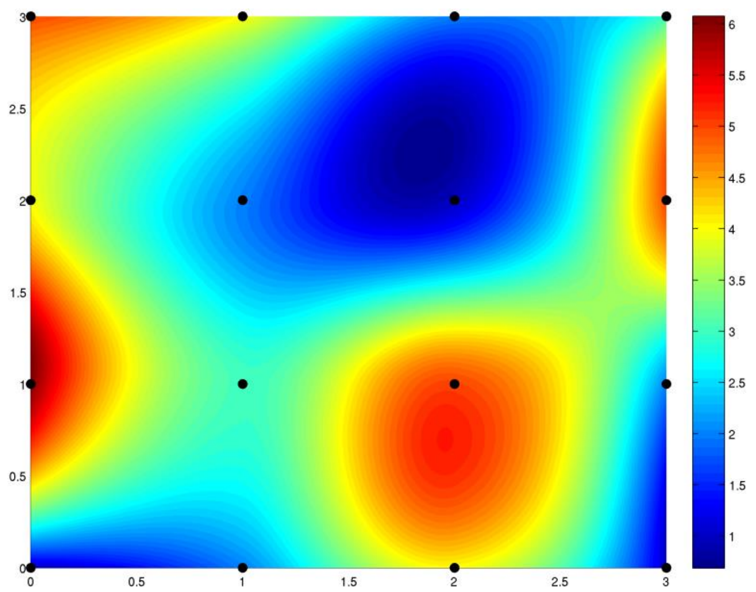
Cubic (Catmull Rom) $$f(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 1 & -\frac{5}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \end{bmatrix} \begin{pmatrix} f_{-1} \\ f_0 \\ f_1 \\ f_2 \end{pmatrix}$$

- **Interpolation in 2D, 3D, 4D, …**

- **Tensor Product Interpolation**
  - Perform linear / cubic … interpolation in each x,y,z … direction **separately**



Bi-Linear



Bi-Cubic

- **Tensor Product Interpolation**
  - Extend interpolation from 1D to higher dimensions
  - Coefficients f_i , associated basis functions $b_i(x)$ (linear / cubic / …)

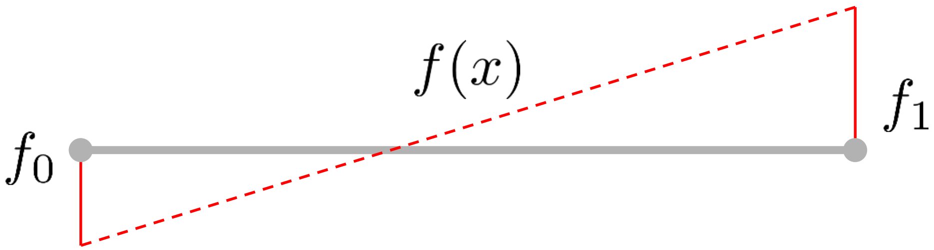
1D
$$f(x) = \sum_{i=0}^{n} b_i(x)\, f_i \qquad = \begin{bmatrix} b_0(x) & \cdots & b_n(x) \end{bmatrix} \begin{bmatrix} f_0 \\ \vdots \\ f_n \end{bmatrix}$$

2D, “bi-”
$$f(x, y) = \sum_{j=0}^{m} \sum_{i=0}^{n} b_i(x) b_j(y)\, f_{ij}$$
$$= \begin{bmatrix} b_0(x) & \cdots & b_n(x) \end{bmatrix} \begin{bmatrix} f_{00} & \cdots & f_{0m} \\ \vdots & \ddots & \vdots \\ f_{n0} & \cdots & f_{nm} \end{bmatrix} \begin{bmatrix} b_0(y) \\ \vdots \\ b_m(y) \end{bmatrix}$$

3D, “tri-”
$$f(x, y, z) = \sum_{k=0}^{p} \sum_{j=0}^{m} \sum_{i=0}^{n} b_i(x) b_j(y) b_k(z)\, f_{ijk}$$

- Example: **Linear** Tensor Product Interpolation
  - Number of basis functions / coefficients $m = 1, n = 1, p = 1$

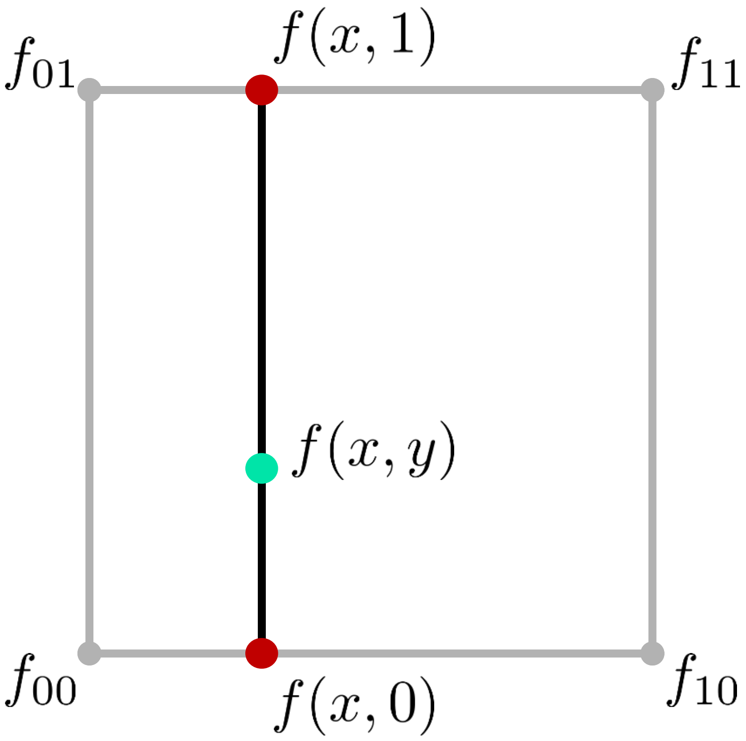
1D, linear $$f(x) = \sum_{i=0}^{n} b_i(x)\, f_i = (1 - x)\, f_0 + x\, f_1$$



very important

- Example: **Linear** Tensor Product Interpolation
  - Number of basis functions / coefficients $m = 1, n = 1, p = 1$

2D, “bi-linear”



$$f(x,y) = \sum_{j=0}^{m} \sum_{i=0}^{n} b_i(x) b_j(y)\, f_{ij}$$

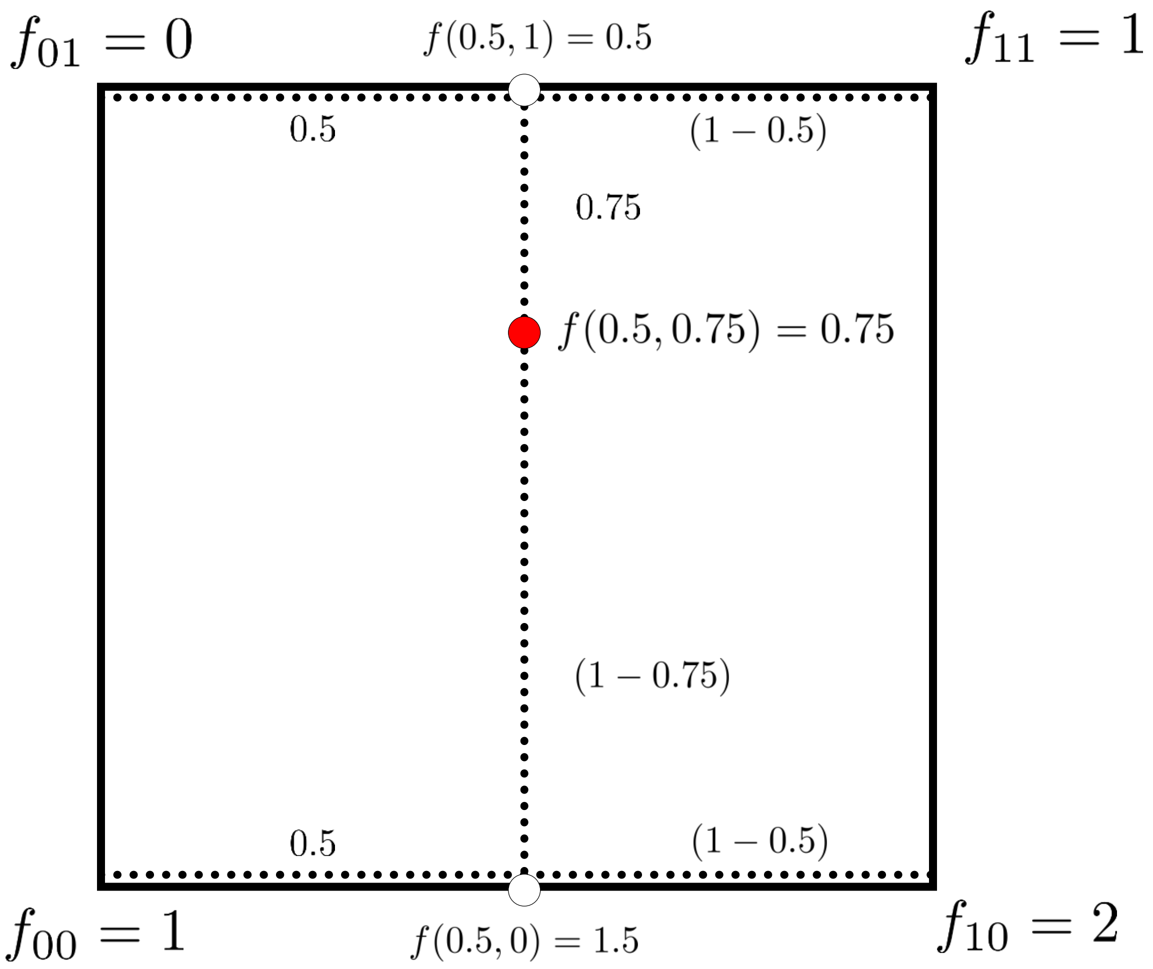
$$= (1-x)\,(1-y)\, f_{00} + x\,(1-y)\, f_{10} + (1-x)\, y\, f_{01} + x\, y\, f_{11}$$

$$= (1-y)\,((1-x)\, f_{00} + x\, f_{10}) + y\,((1-x)\, f_{01} + x\, f_{11})$$

“interpolate twice in x direction and then once in y direction”

- **Example: Bi-linear interpolation in a 2D cell**
  - Repeated linear interpolation

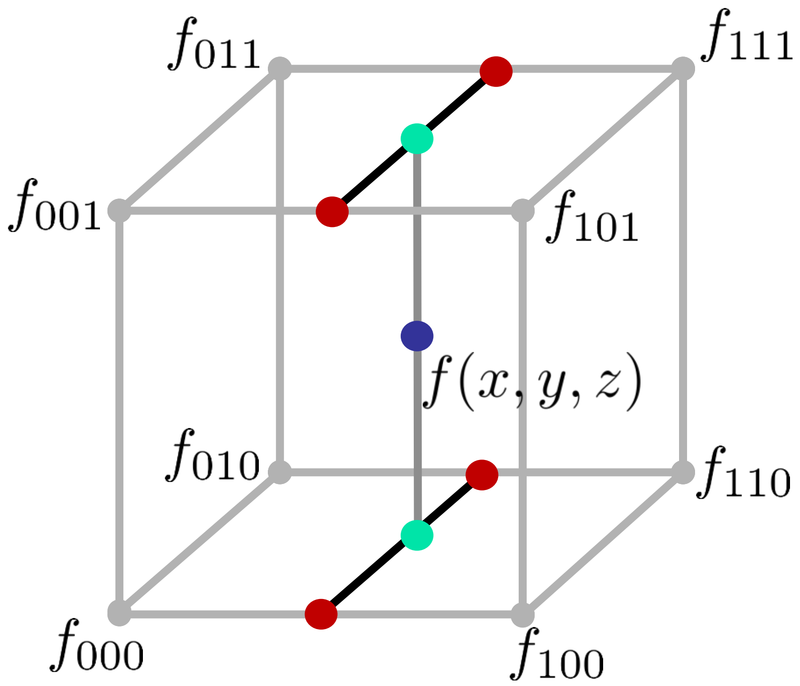
$$f(0.5, 0.75) = ?$$



- Example: **Linear** Tensor Product Interpolation
  - Number of basis functions / coefficients $m = 1, n = 1, p = 1$

3D, “tri-linear”

$$f(x, y, z) = \sum_{k=0}^{p} \sum_{j=0}^{m} \sum_{i=0}^{n} b_i(x) b_j(y) b_k(z)\, f_{ijk}$$



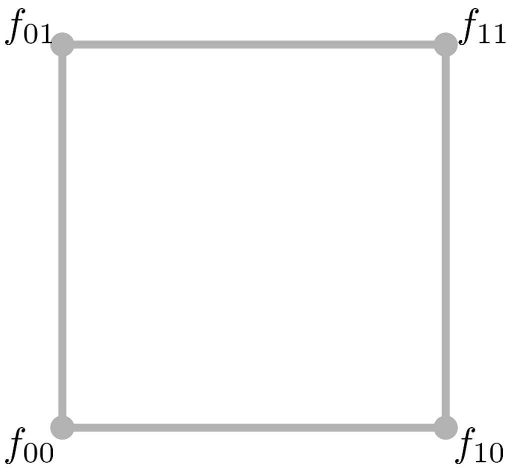
“interpolate four times in x direction, twice in y direction, and once in z direction”

$$
\begin{aligned}
f(x) = \; & (1-x) * f_0 \\
+ \; & x * f_1
\end{aligned}
$$

1D linear

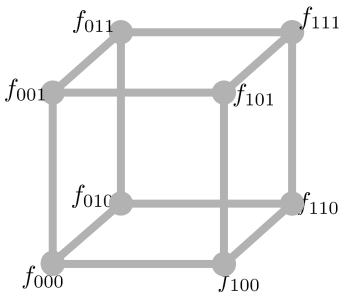
$$
\begin{aligned}
f(x,y) = \; & (1-x) * (1-y) * f_{00} \\
+ \; & x * (1-y) * f_{10} \\
+ \; & (1-x) * y * f_{01} \\
+ \; & x * y * f_{11}
\end{aligned}
$$

2D bi-linear



$$
\begin{aligned}
f(x,y,z) = \; & (1-x) * (1-y) * (1-z) * f_{000} \\
+ \; & x * (1-y) * (1-z) * f_{100} \\
+ \; & (1-x) * y * (1-z) * f_{010} \\
+ \; & x * y * (1-z) * f_{110} \\
+ \; & (1-x) * (1-y) * z * f_{001} \\
+ \; & x * (1-y) * z * f_{101} \\
+ \; & (1-x) * y * z * f_{011} \\
+ \; & x * y * z * f_{111}
\end{aligned}
$$

3D tri-linear



Two ways to estimate gradients:

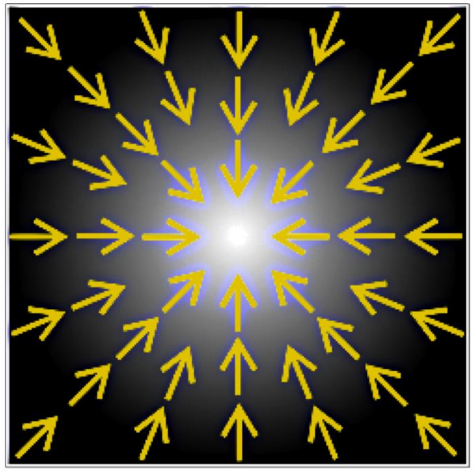
- Direct derivation of interpolation formula
- Finite differences schemes

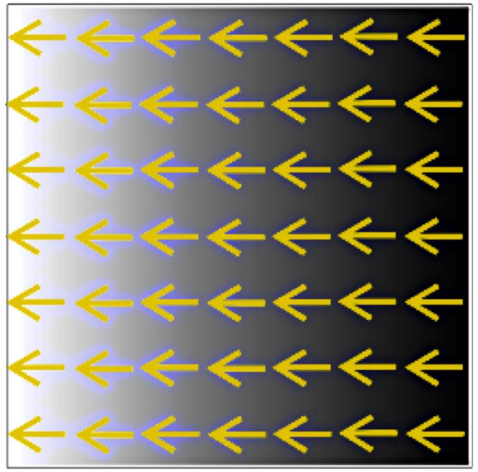
$$\nabla s(x,y) = \begin{pmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial y} \end{pmatrix} = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$$

Gradient of a 2D scalar field

$$\nabla \mathbf{v}(x,y) = \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix}$$

Jacobian of a 2D vector field





- **Field Function Derivatives**, Bi-Linear

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

derive this interpolation formula

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

“constant in x direction“

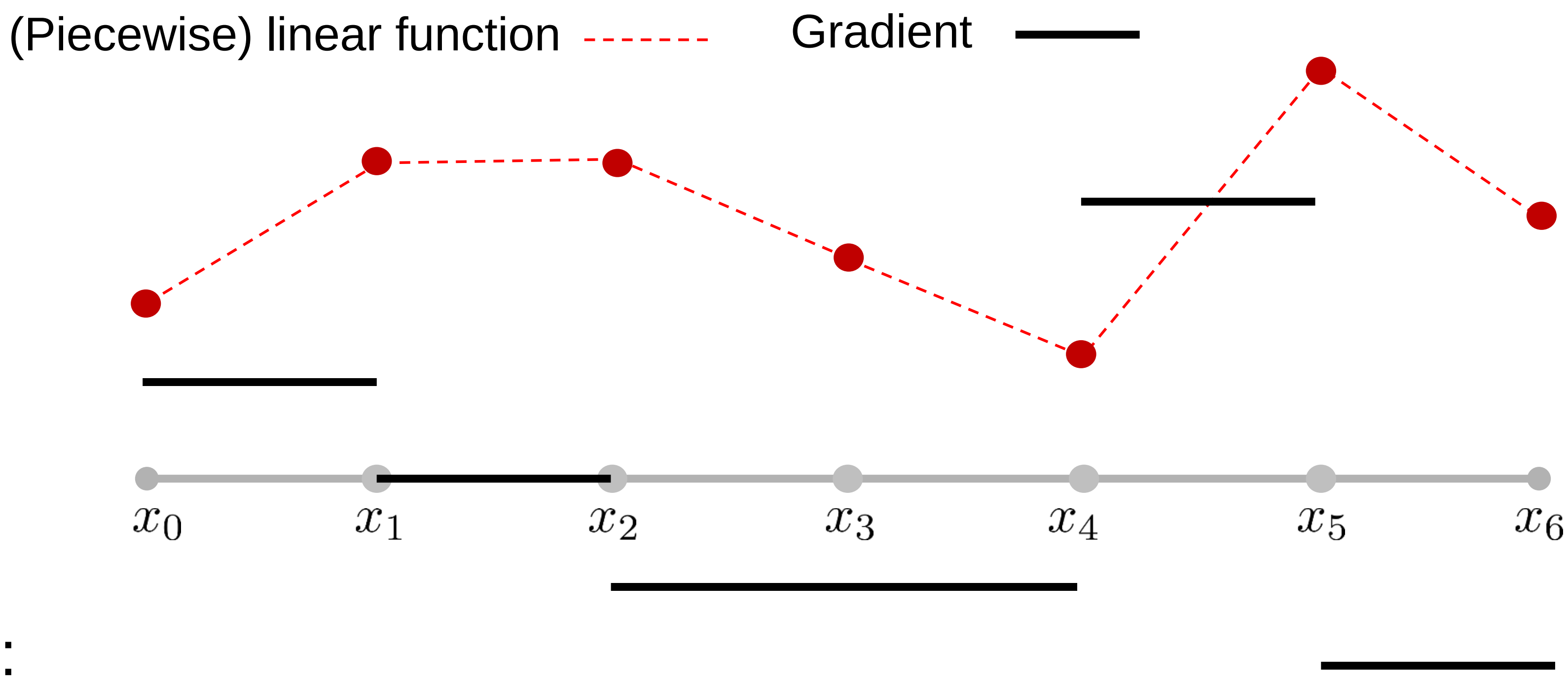
$$\frac{\partial f(x,y)}{\partial y} = \begin{bmatrix} (1-x) & x \end{bmatrix} \begin{bmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$
$$= (f_{01} - f_{00})(1-x) + (f_{11} - f_{10})\, x$$

“constant in y direction“

$$\nabla f(x,y) = \begin{pmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

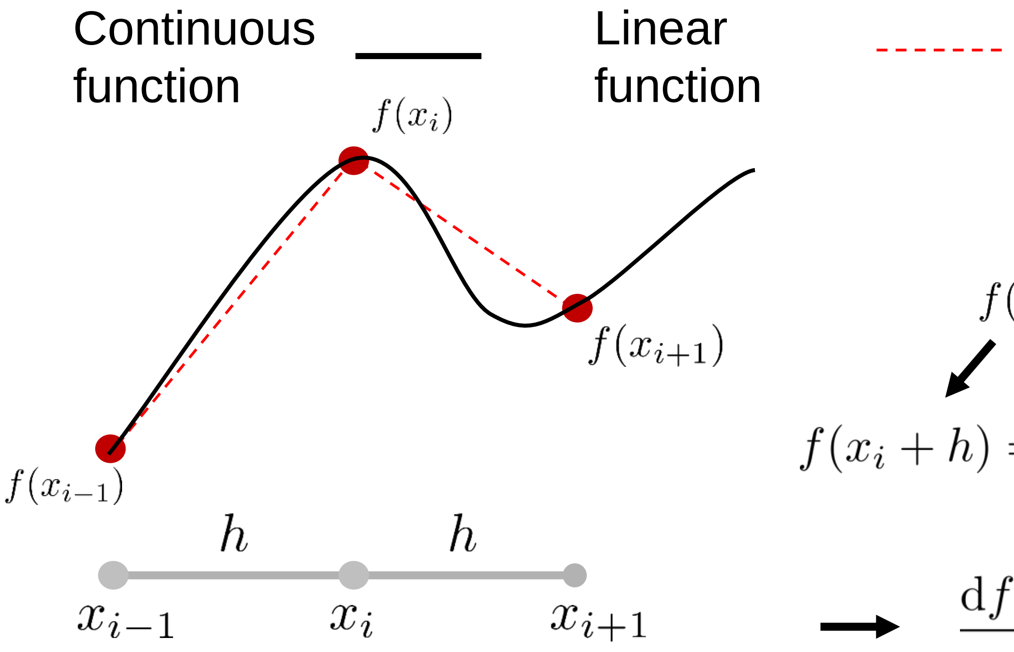
final gradient

- Problem of exact linear function differentiation: discontinuous gradients



- Solution:
  - Use higher order interpolation scheme (cubic)
  - Use **finite difference estimation**

- **Finite Differences Schemes**
  - Apply Taylor series expansion around samples



Taylor expansion

$f(x_{i+1})$

$$f(x_i + h) = f(x_i) + h\,\frac{\mathrm{d}f(x_i)}{\mathrm{d}x} + \frac{h^2}{2}\,\frac{\mathrm{d}^2 f(x_i)}{\mathrm{d}x^2} + O(h^3)$$

$\rightarrow$ $$\frac{\mathrm{d}f(x_i)}{\mathrm{d}x} \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$ **Forward difference**

$\rightarrow$ $$\frac{\mathrm{d}f(x_i)}{\mathrm{d}x} \approx \frac{f(x_i) - f(x_{i-1})}{h}$$ **Backward difference**

- **Finite Differences Schemes**

$$f(x_{i+1}) = f(x_i) + h\,\frac{\mathrm{d}f(x_i)}{\mathrm{d}x} + \frac{h^2}{2}\,\frac{\mathrm{d}^2 f(x_i)}{\mathrm{d}x^2} + O(h^3)$$

$$f(x_{i-1}) = f(x_i) - h\,\frac{\mathrm{d}f(x_i)}{\mathrm{d}x} + \frac{h^2}{2}\,\frac{\mathrm{d}^2 f(x_i)}{\mathrm{d}x^2} + O(h^3)$$

Difference

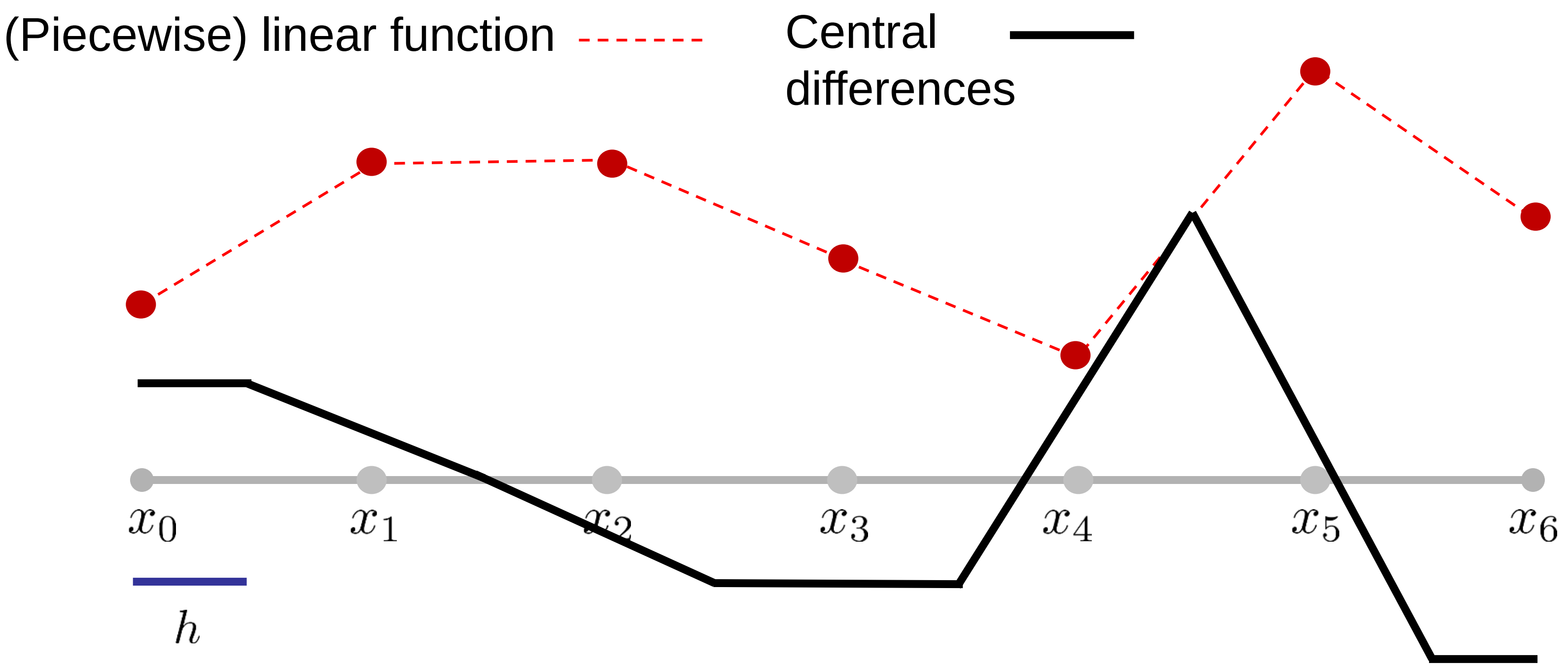
$$\longrightarrow \quad (f(x_{i+1}) - f(x_i)) - (f(x_{i-1}) - f(x_i)) = 2h\,\frac{\mathrm{d}f(x_i)}{\mathrm{d}x} + O(h^3)$$

$$\longrightarrow \quad \frac{\mathrm{d}f(x_i)}{\mathrm{d}x} \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

Central difference

- Central differences have higher approximation order than forward / backward differences

- 1D Example, linear interpolation

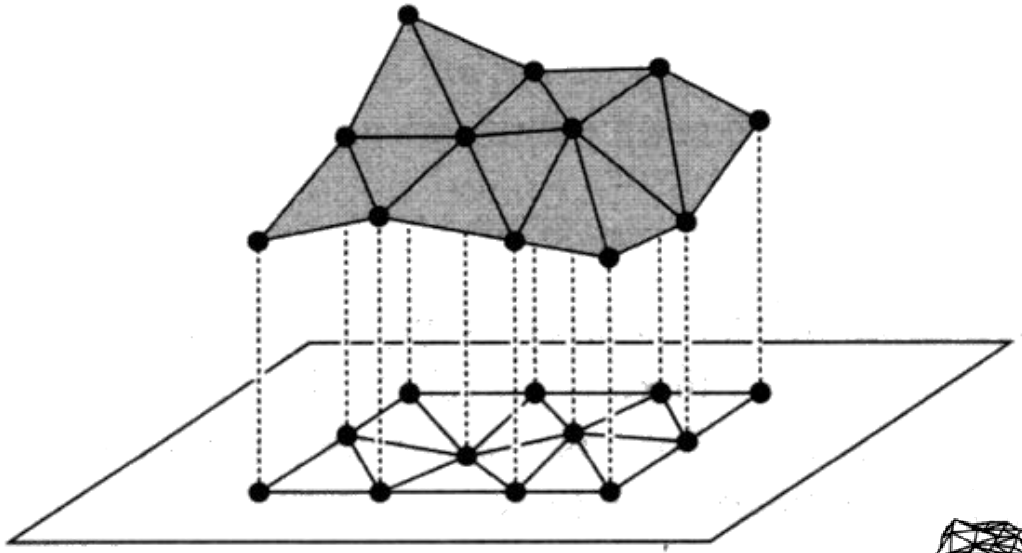


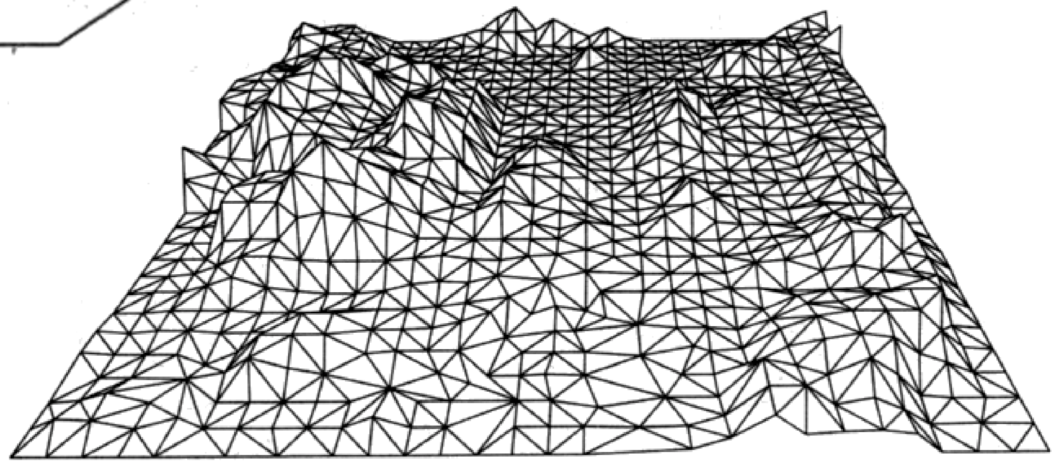
- **Finite Differences Schemes,** Higher order derivatives

$$\frac{\mathrm{d}^2 f(x_i)}{\mathrm{d}x^2} \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$$

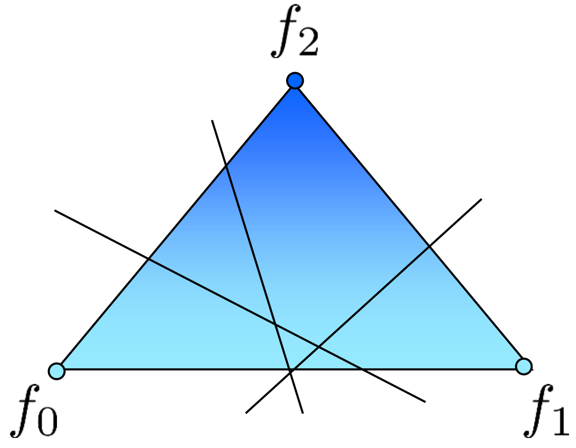
$$\frac{\partial^2 f(x_i, y_j)}{\partial xy} \approx \frac{f(x_{i+1}, y_{j+1}) - f(x_{i+1}, y_{j-1}) - f(x_{i-1}, y_{j+1}) + f(x_{i-1}, y_{j-1})}{4\, h_x\, h_y}$$

- **Piecewise Linear Interpolation in Triangle Meshes**





- **Linear Interpolation in a Triangle**
  - There is exactly one linear function that satisfies the interpolation constraint



- A linear function can be written as

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

  - Polynomial can be obtained by solving the linear system

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

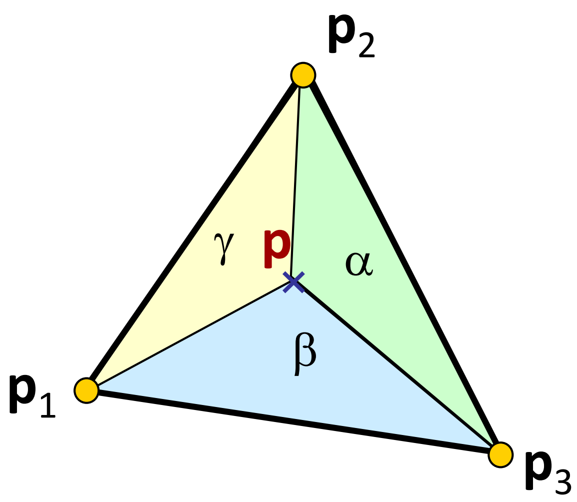
- Linear in x and y

- Interpolated values along any ray in the plane spanned by the triangle are linear along that ray

- Barycentric Coordinates:
  - Planar case:
    Barycentric combinations of 3 points

$$\mathbf{p} = \alpha \mathbf{p}_1 + \beta \mathbf{p}_2 + \gamma \mathbf{p}_3, \text{with}: \alpha + \beta + \gamma = 1$$

$$\gamma = 1 - \alpha - \beta$$



  - Area formulation:

$$\alpha = \frac{area(\Delta(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}))}{area(\Delta(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3))}, \beta = \frac{area(\Delta(\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}))}{area(\Delta(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3))}, \gamma = \frac{area(\Delta(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}))}{area(\Delta(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3))}$$

very important

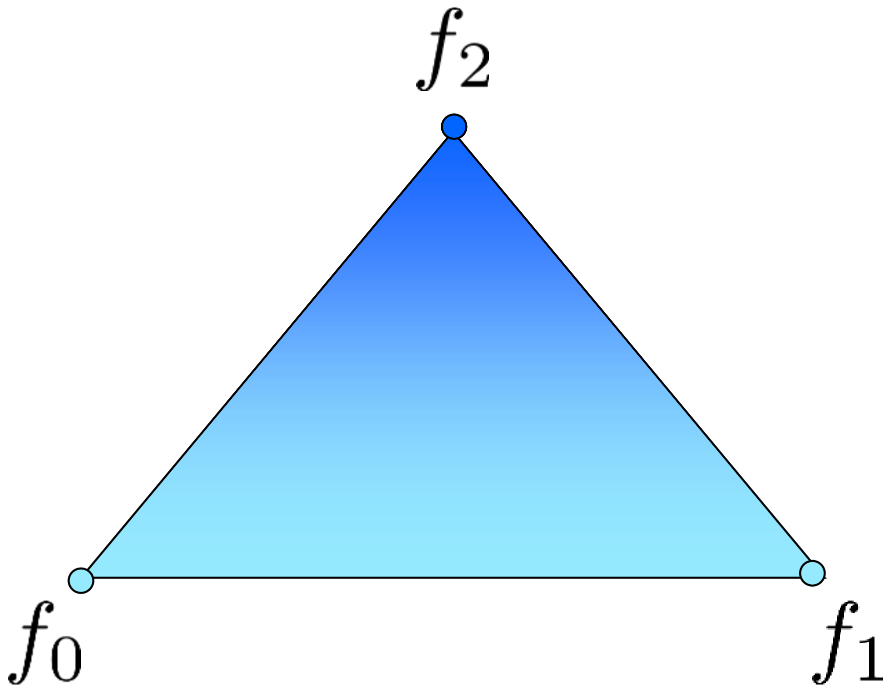
Barycentric Interpolation in a Triangle

The linear function of a triangle can be computed at any point as

$$f(x,y) = \alpha(x,y)f_0 + \beta(x,y)f_1 + \gamma(x,y)f_2$$

with $\alpha + \beta + \gamma = 1$ as barycentric coordinates.

To be inside triangle: $0 \leq \alpha, \beta, \gamma \leq 1$



- **Background on Barycentric Interpolation in a Triangle**
  - The linear function of a triangle can be computed at any point as

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

    with $\alpha_0 + \alpha_1 + \alpha_2 = 1$ (**Barycentric Coordinates**)

  - This also holds for the coordinate $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ of the triangle:

    $$\mathbf{x} = \alpha_0\,\mathbf{x}_0 + \alpha_1\,\mathbf{x}_1 + \alpha_2\,\mathbf{x}_2$$

    → Can be used to solve for unknown coefficients α_i :

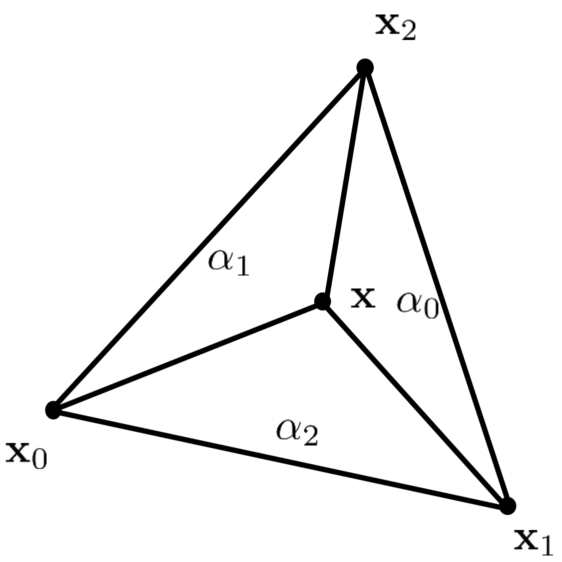
    $$\begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- **Background on Barycentric Interpolation in a Triangle**
  - Solution of $\begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ (e.g. Cramer's rule):

$$\alpha_0 = \frac{1}{2A} \det\left(\begin{bmatrix} x & x_1 & x_2 \\ y & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} \right) \qquad \alpha_0 = \frac{\text{Area}([\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2])}{\text{Area}([\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2])}$$

$$\alpha_1 = \frac{1}{2A} \det\left(\begin{bmatrix} x_0 & x & x_2 \\ y_0 & y & y_2 \\ 1 & 1 & 1 \end{bmatrix} \right) \qquad \alpha_1 = \frac{\text{Area}([\mathbf{x}_0, \mathbf{x}, \mathbf{x}_2])}{\text{Area}([\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2])}$$

$$\alpha_2 = \frac{1}{2A} \det\left(\begin{bmatrix} x_0 & x_1 & x \\ y_0 & y_1 & y \\ 1 & 1 & 1 \end{bmatrix} \right) \qquad \alpha_2 = \frac{\text{Area}([\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}])}{\text{Area}([\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2])}$$



with

$$A = \frac{1}{2} \det\left(\begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} \right)$$

Inside triangle criteria

$$0 \leq \alpha_0,\ \alpha_1,\ \alpha_2 \leq 1$$

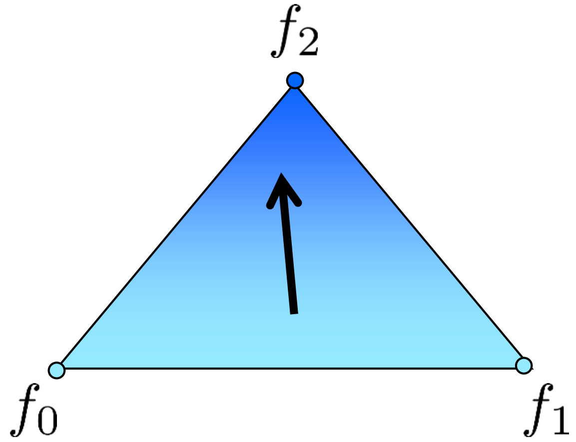
- **Barycentric Interpolation in a Tetrahedron**
- Analogous to the triangle case

Linear function in triangle/tetrahedron:

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

Gradient of a linear function:

Constant!



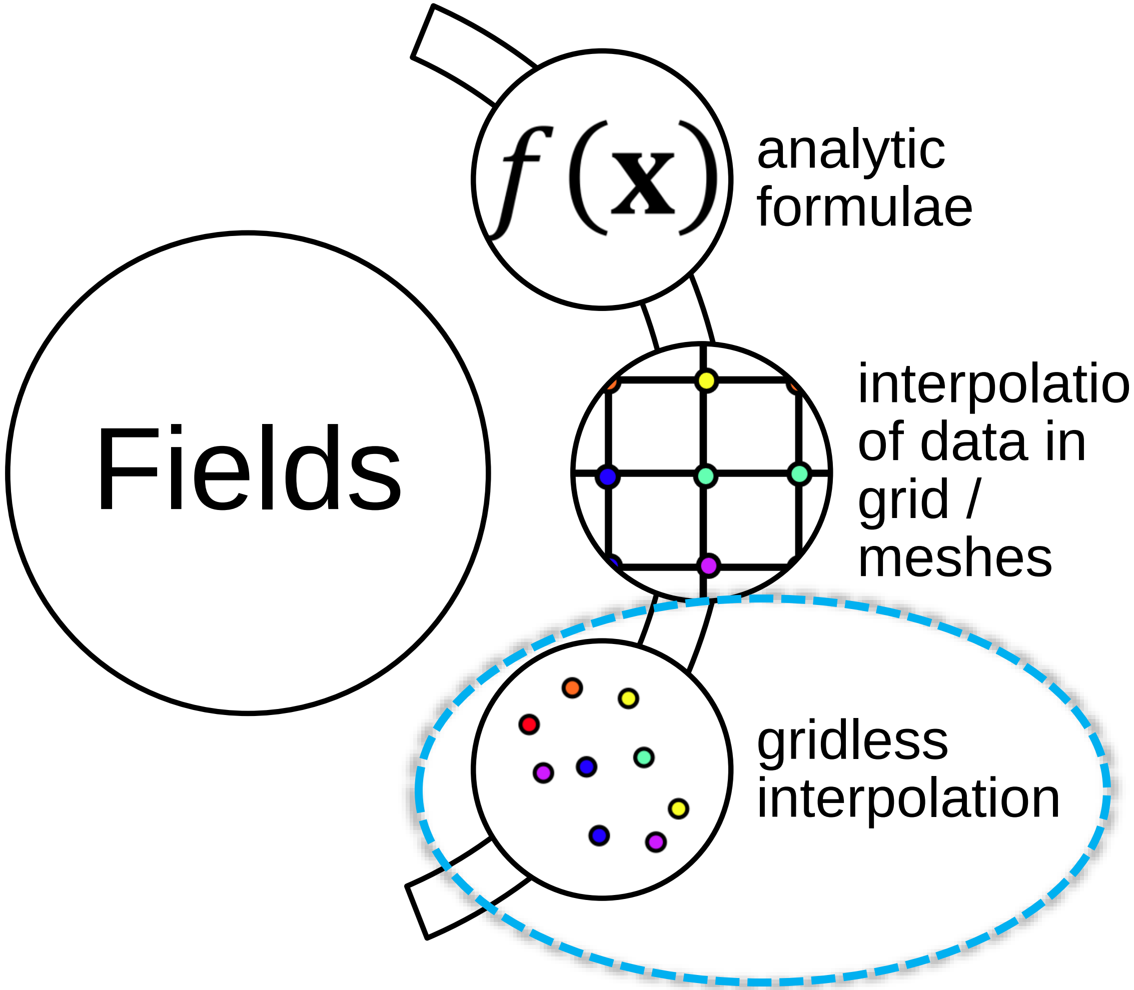
For a linear function in a triangle,
the gradient is a constant 2D vector.

For a linear function in a tetrahedron,
the gradient is a constant 3D vector.

Fields

continuous representation of a variable

sampled data:
interpolation formulae used to create a continuous representation



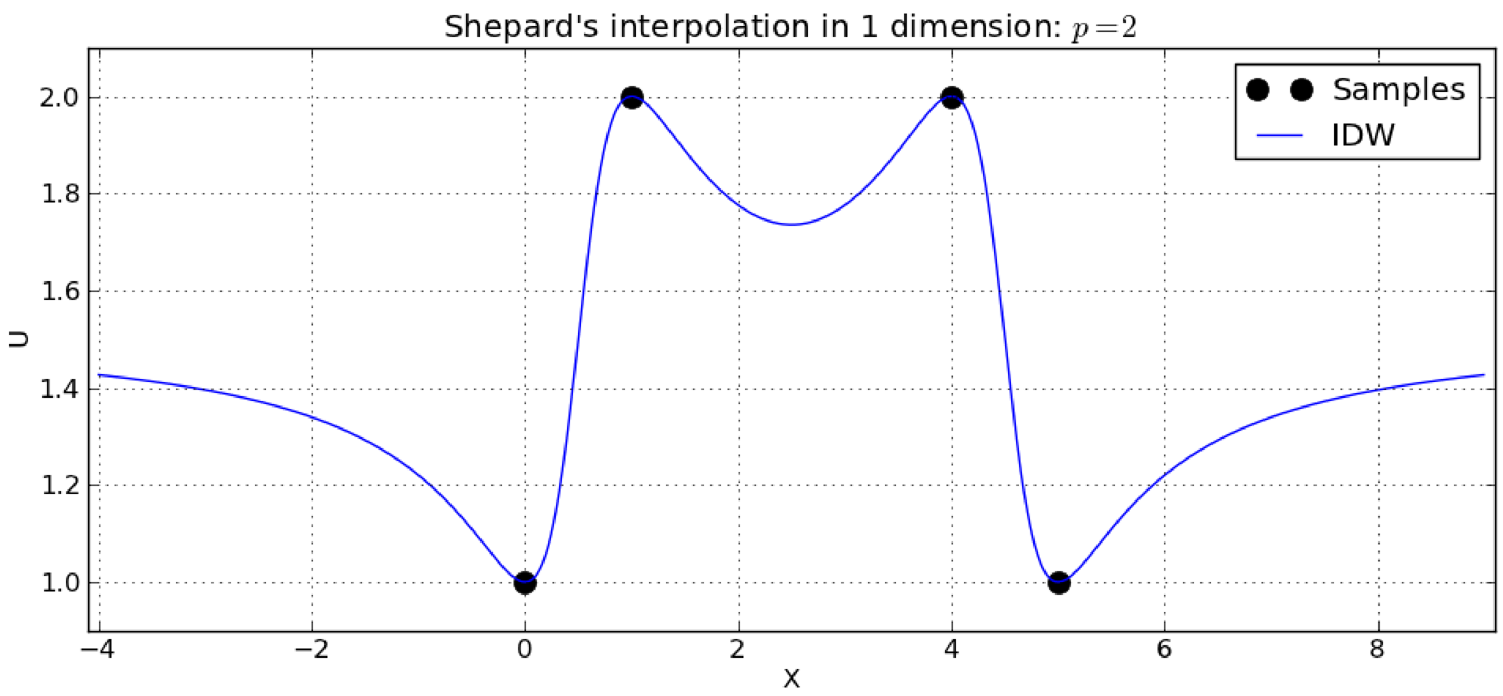
- There is a variety of further scattered data interpolation schemes, e.g., radial basis functions.

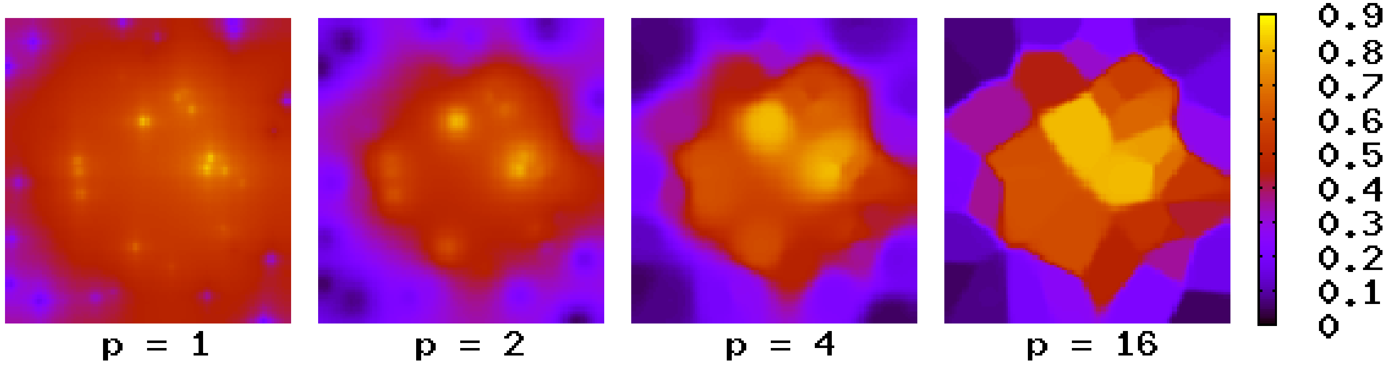
- One of them is the **Shepard approach**:

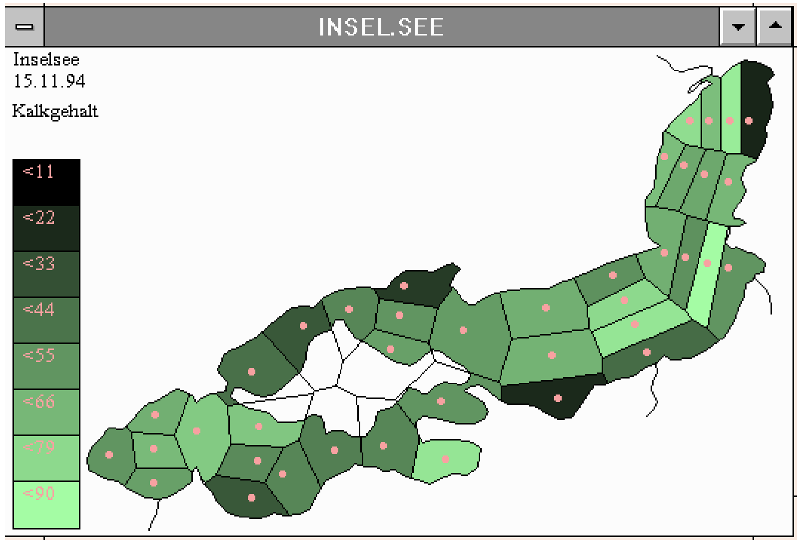
$$f(x,y) = \sum_{k=1}^{n} \frac{w_k(x,y)}{\sum_{j=1}^{n} w_j(x,y)} f_k$$

- The weight function $w_k(x,y)$ is constructed in such a way that the impact of an observation point decreases far away from the observation point, i.e.,

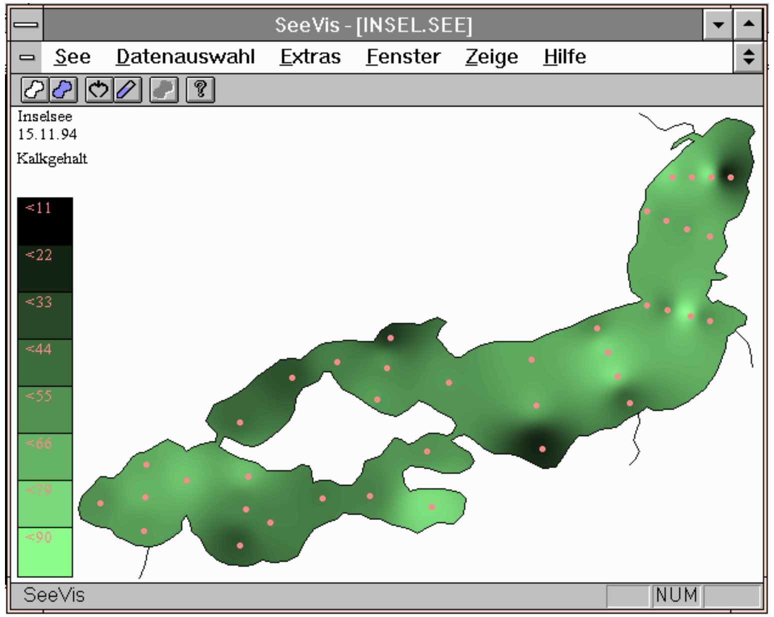
$$w_k(x,y) = \frac{1}{((x-x_k)^p + (y-y_k)^p)^{\frac{1}{p}}} = (\frac{1}{d_k})$$


Shepard's interpolation in 1 dimension: $p = 2$
Samples
IDW
u
x
2.0
1.8
1.6
1.4
1.2
1.0
−4
−2
0
2
4
6
8


p = 1
p = 2
p = 4
p = 16
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0



2D Voronoi diagram



Shepard interpolation

- For this choice of w_k , all data values have global impact.
- Using the **Franke Little weight function**, local impact can be achieved.

$$w_k(x, y) = \left(\frac{\max(r - d_k,\, 0)}{r\, d_k} \right)^2$$

Summary

- Fields are continuous representations of variables
  - analytic formulae
  - interpolation
    - in grids / meshes
    - gridless
- Interpolation
  - Linear / Cubic basis functions
  - Multidimensional interpolation (bi-linear, tri-linear …)
  - Linear interpolation on triangles and tetrahedra
  - Grid free interpolation using Shepard approach
- Gradients
  - Derivatives of (interpolation) formulae
  - Finite differences