Interpolation and Data Fitting

Prof. Dr. Renato Pajarola
Visualization and MultiMedia Lab
Department of Informatics
University of Zürich

Copyrights

- Most figures of these slides are copyright protected and come from the book ***Mathematical Principles for Scientific Computing and Visualization*** or from other indicated sources
- You understand that the slides contain copyright protected material and therefore the following conditions of use apply:
  - The slides may be used for personal teaching purposes only
  - Publishing the slides to any public web site is not allowed
  - Sharing the slides with other persons or institutions is prohibited

Overview

1. Piecewise constant and linear interpolation
2. Interpolation in 2D
3. Smooth polynomial interpolation
4. Polynomial least squares interpolation

Piecewise Constant and Linear Interpolation

Interpolation

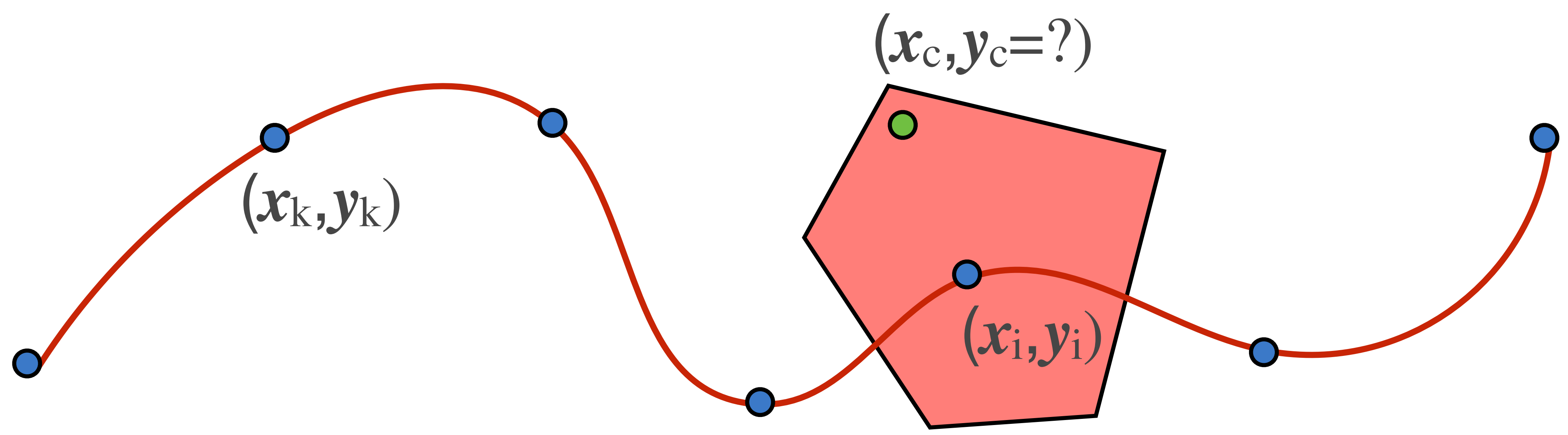
- Interpolation is a method to estimate, construct new data points from a given set of discrete known data points
  - Over a bounded range of the independent variable attribute(s) of the input data points
    - extrapolation used to estimate data points outside the range of the known data points
  - An interpolation function exactly passes through all known data points
  - Different from an approximation which only approximates them, thus passes nearby instead
- Transformation of data from a given (source) distribution to another (target) resolution is a frequent task of data preprocessing
  - Filling-in of continuous data values in between scattered sample data points
    - e.g. temporal domain, filling in intermediate values of temperature between observed values
    - e.g. spatial domain, image resizing or resampling (up-/down-)

Piecewise Constant Interpolation

- Available data points often represent a discrete sampling of a continuous phenomenon
  - Observed **output values** y_i are given for a few known **input data points** $x_i \in X$
- Given a distance metric $|x_b - x_a|$ between any two data points a and b in the input domain allows a generic *nearest neighbor interpolation* for any **new** data point c with input parameters x_c as:

$$y_c = y_i \text{ where } \forall x_{k \neq i} \in X: |x_c - x_i| < |x_c - x_k|$$

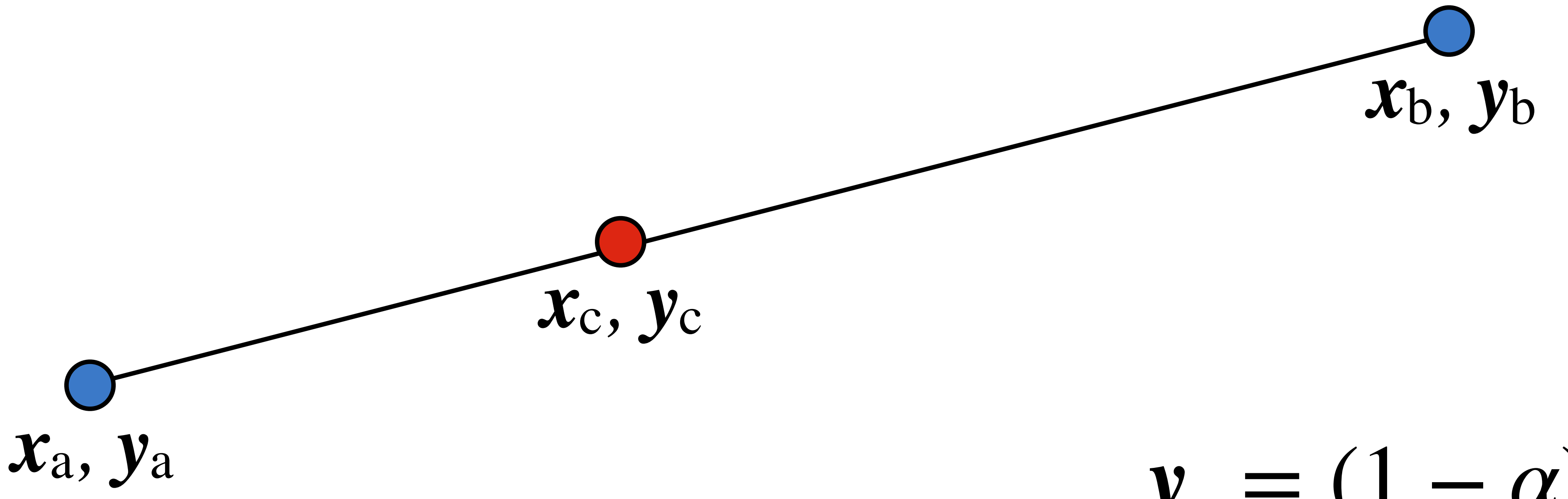
  - New unknown value is inherited from nearest available data point in any dimensions (e.g. $x_i \in \mathbb{R}^d$)
- Continuous interpolation requires (piecewise) linear or higher-order interpolation functions
  - Linear interpolations exhibit only C_0 continuity



Piecewise Linear Interpolation

- Prediction of unknown values from nearby sample locations by interpolation
  - Linear interpolation of new **data value** y_c at a **position** x_c from given data values y_b and y_b on a straight line between two known sample input data points x_a and x_b
    - x_a, x_b and x_c are all on a line $\in \mathbb{R}^D$

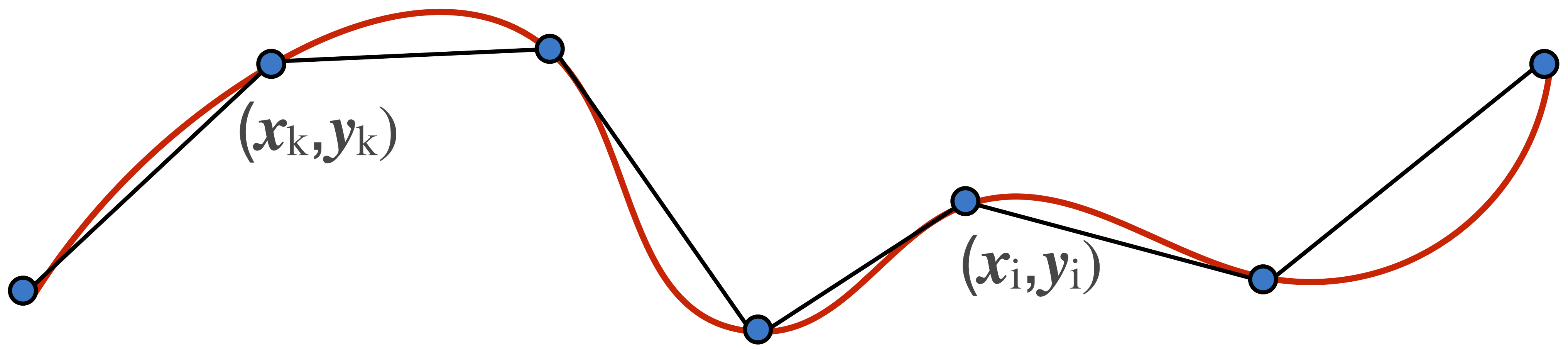
$$y_c = y_a + (y_b - y_a) \cdot \frac{|x_c - x_a|}{|x_b - x_a|}$$



$$\alpha = \frac{|x_c - x_a|}{|x_b - x_a|}$$

$$y_c = (1 - \alpha) \cdot y_a + \alpha \cdot y_b$$

- Linear interpolations exhibit only C_0 continuity
  - Discontinuity in the slope, first and higher derivatives

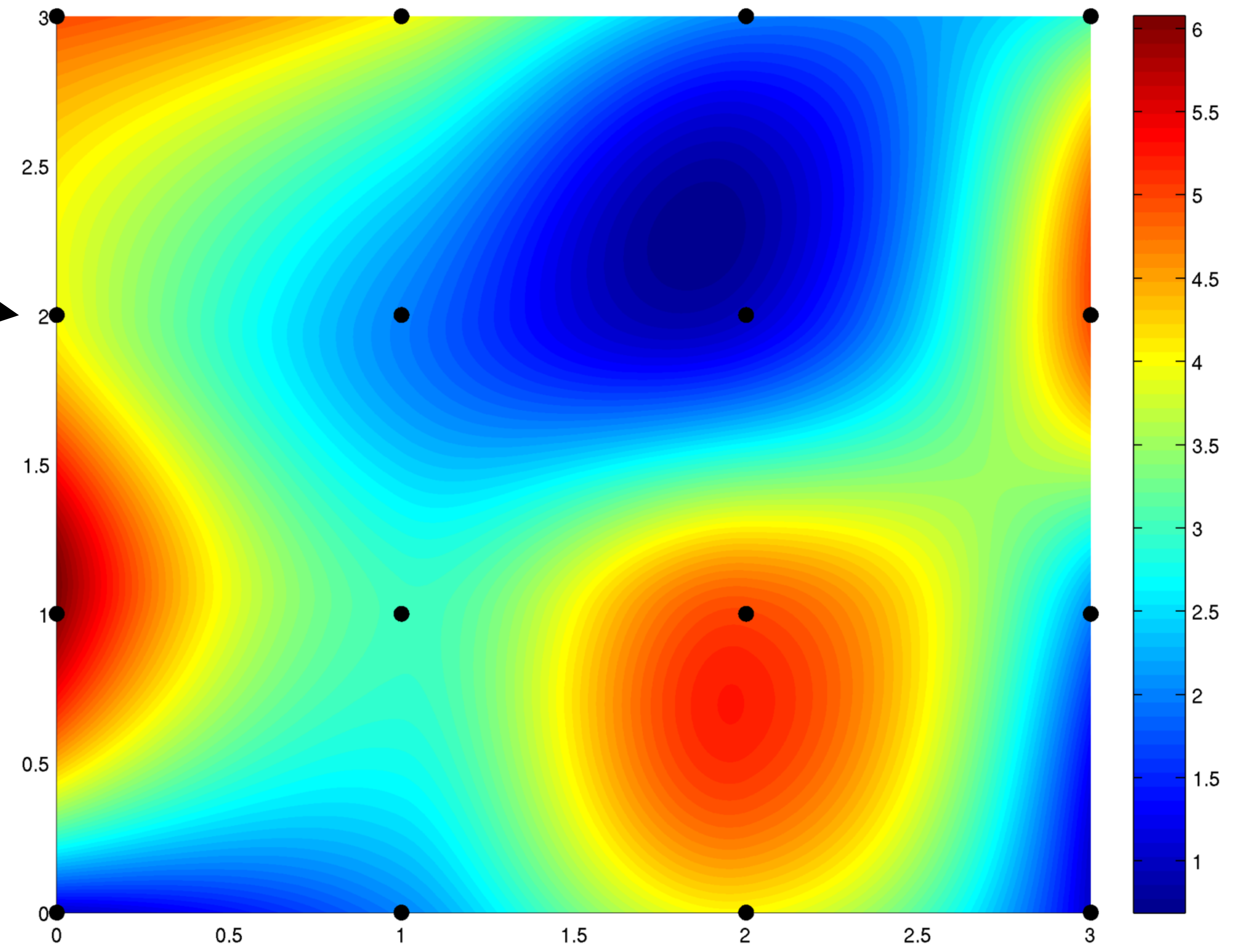


Interpolation in 2D

What Does “2D” Mean?

We always refer to the input’s dimensionality

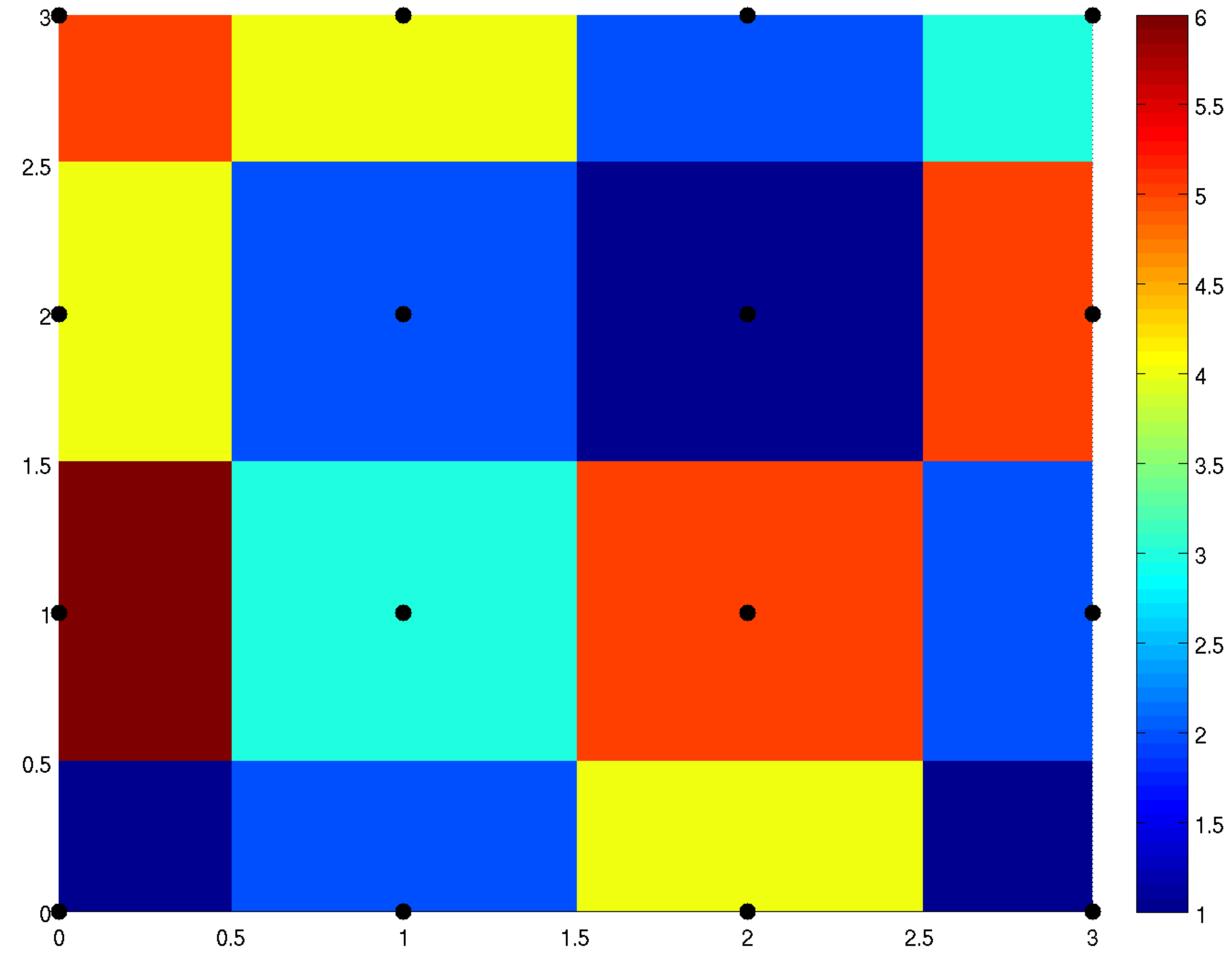
- 1D: one input x → one output y
- 2D: two inputs x, y → one output z
- *N*-dimensions: *N* inputs → still one output

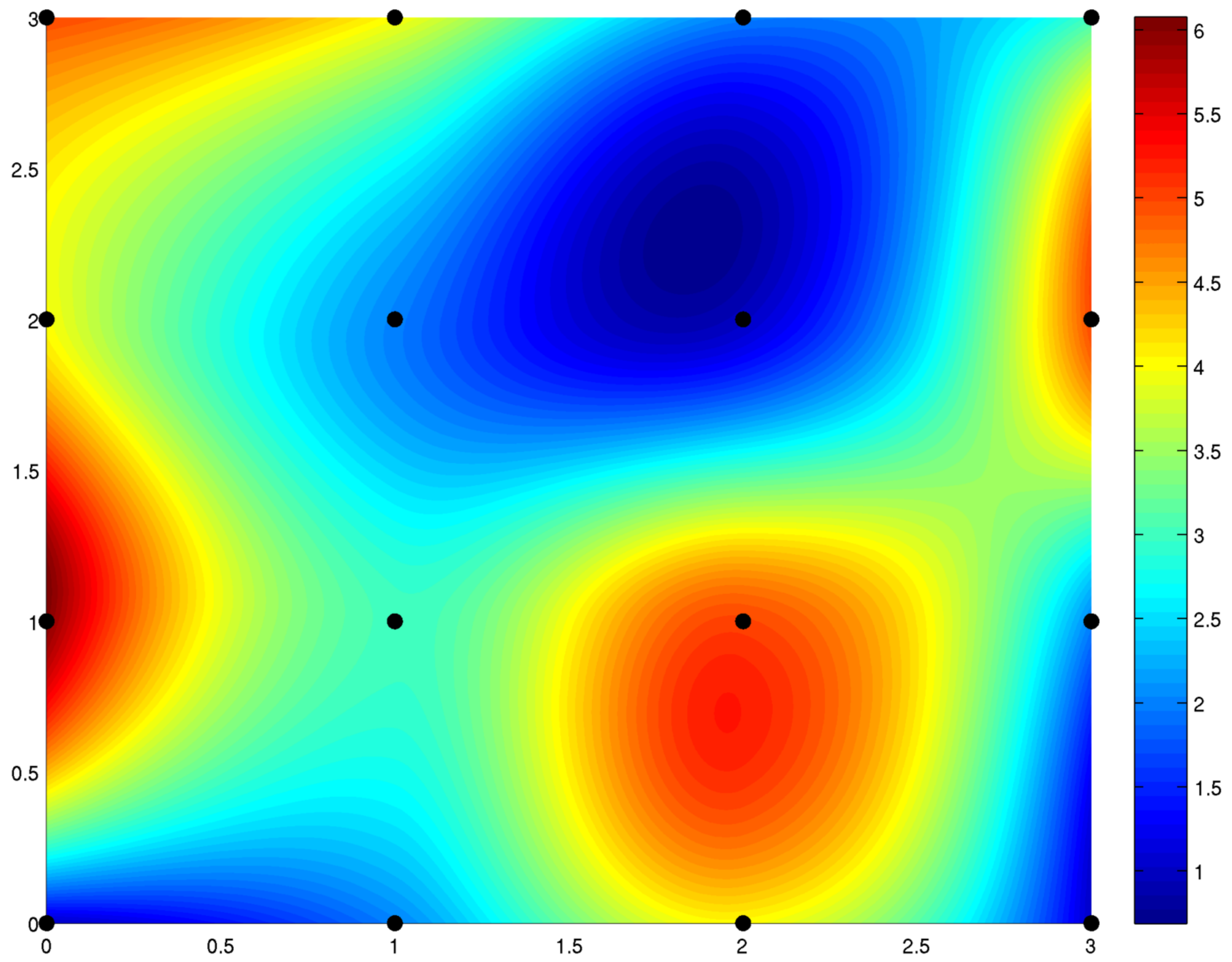


For visualization, we can map the z as e.g. color

Piecewise Constant Interpolation in 2D

- Simple piecewise constant nearest neighbor interpolation clearly exhibits sparse sample distribution
    - Patches of constant values in contrast to smooth distribution of values

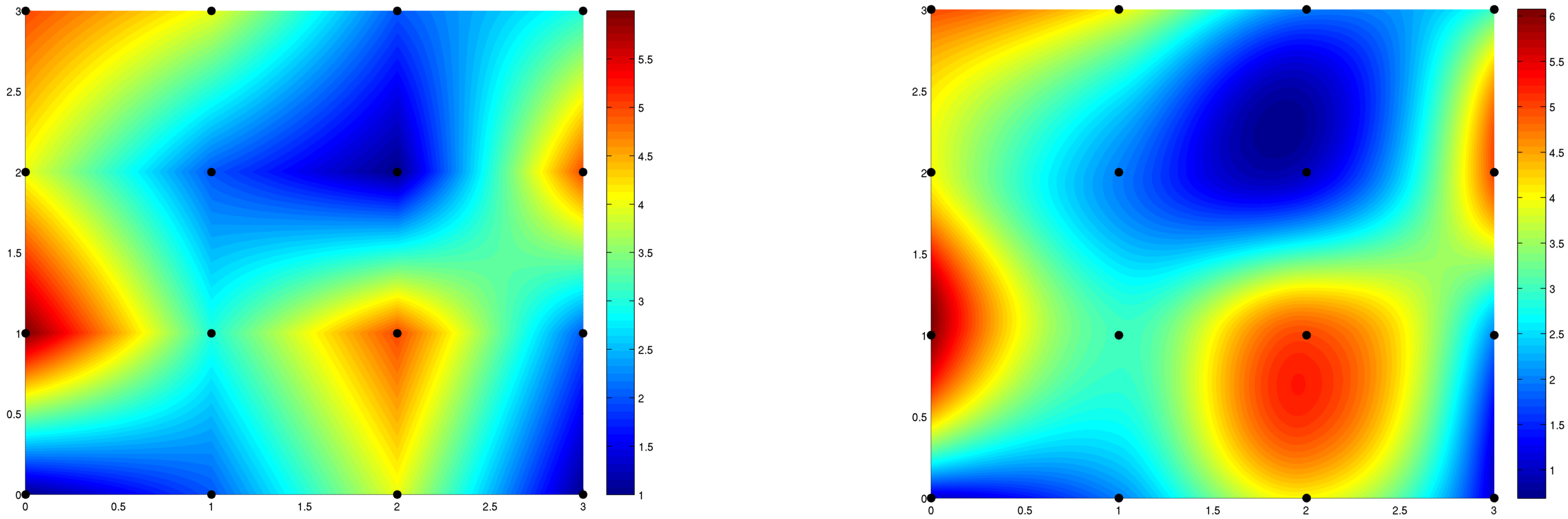




Wikimedia Commons public domain

Bilinear Interpolation in 2D

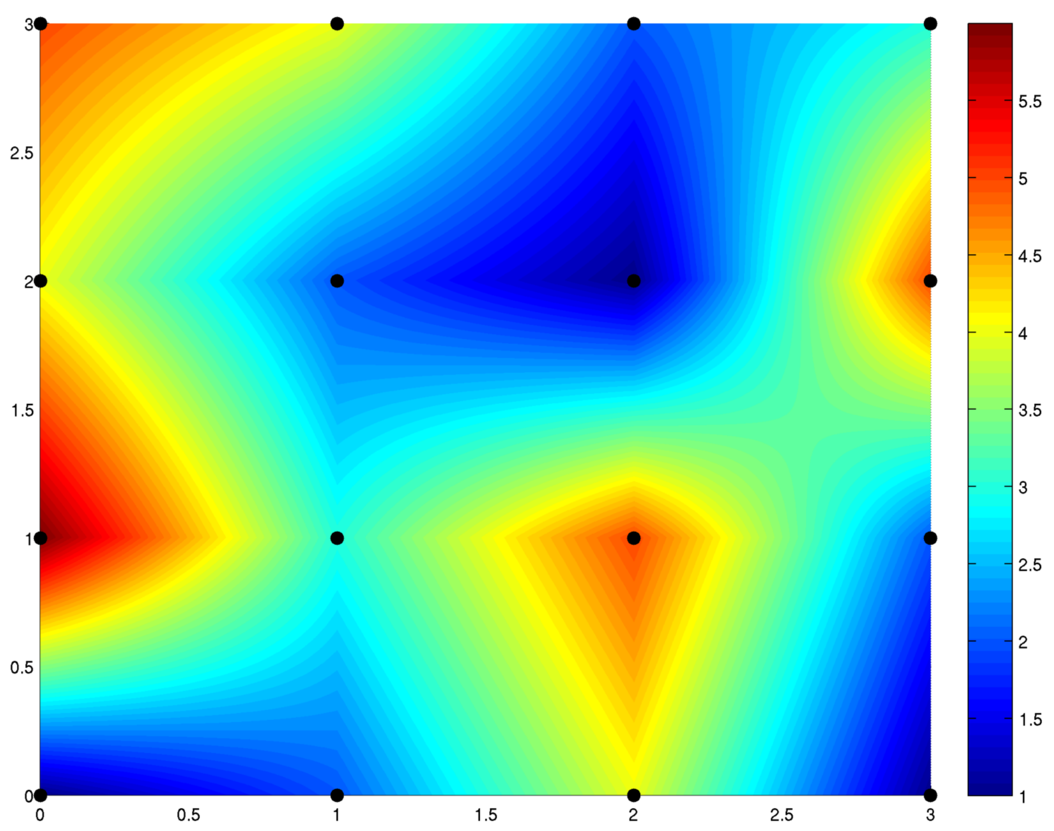
- A grid of data values can easily be interpolated using *bilinear* interpolation
- Only C_0 continuous
  - Discontinuous change in slopes



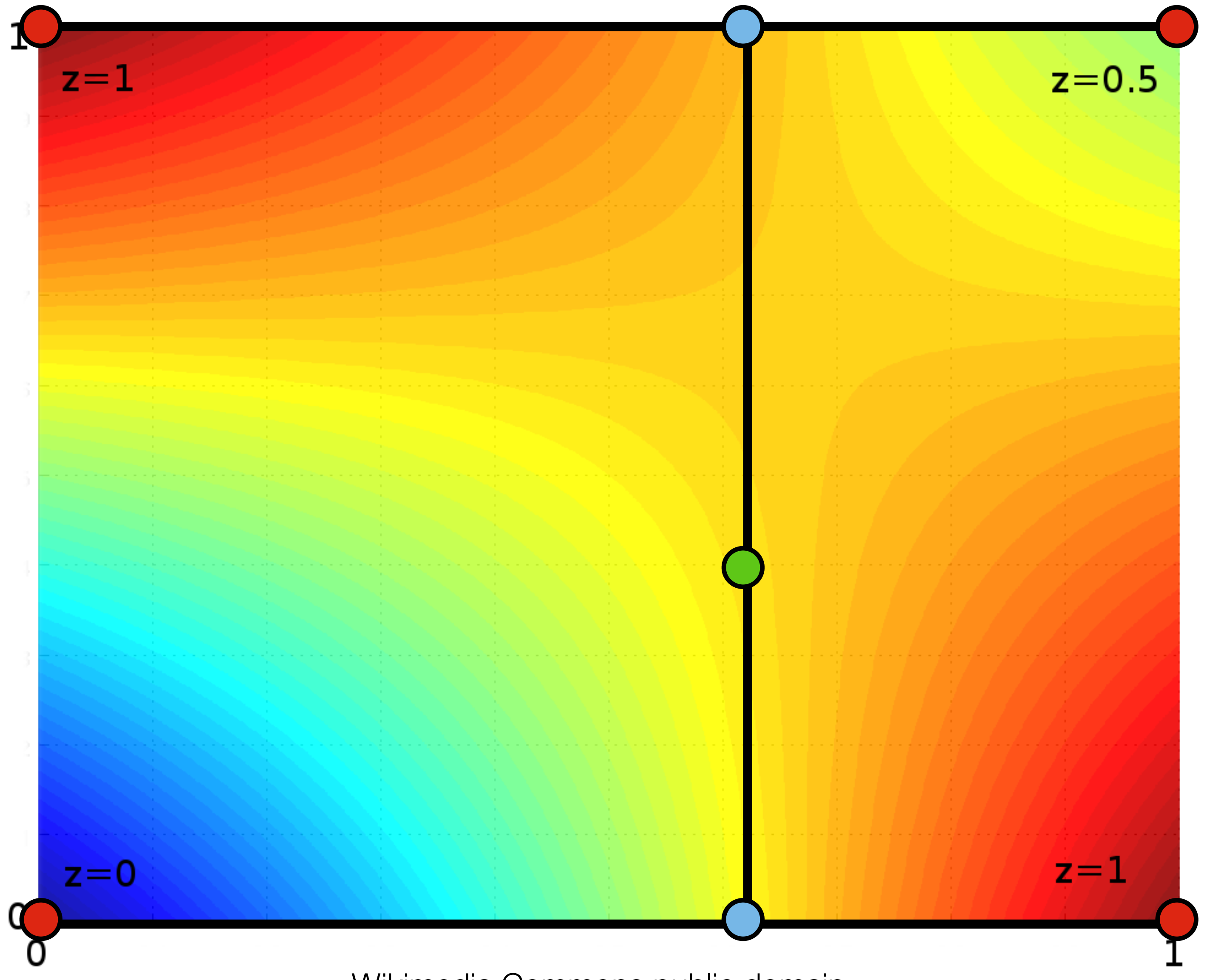
Wikimedia Commons public domain

Bilinear Interpolation in 2D

- **Bi**linear because of the linear interpolations along each dimension separately
  - Two linear interpolations are performed along two opposite boundaries of a grid cell in one dimension, e.g. horizontally
  - One final interpolation is performed across the grid cell in the other dimension, e.g. vertically
- Non-smooth interpolation, only $\mathbf{C_0}$ but not $\mathbf{C_1}$ continuity



Wikimedia Commons public domain

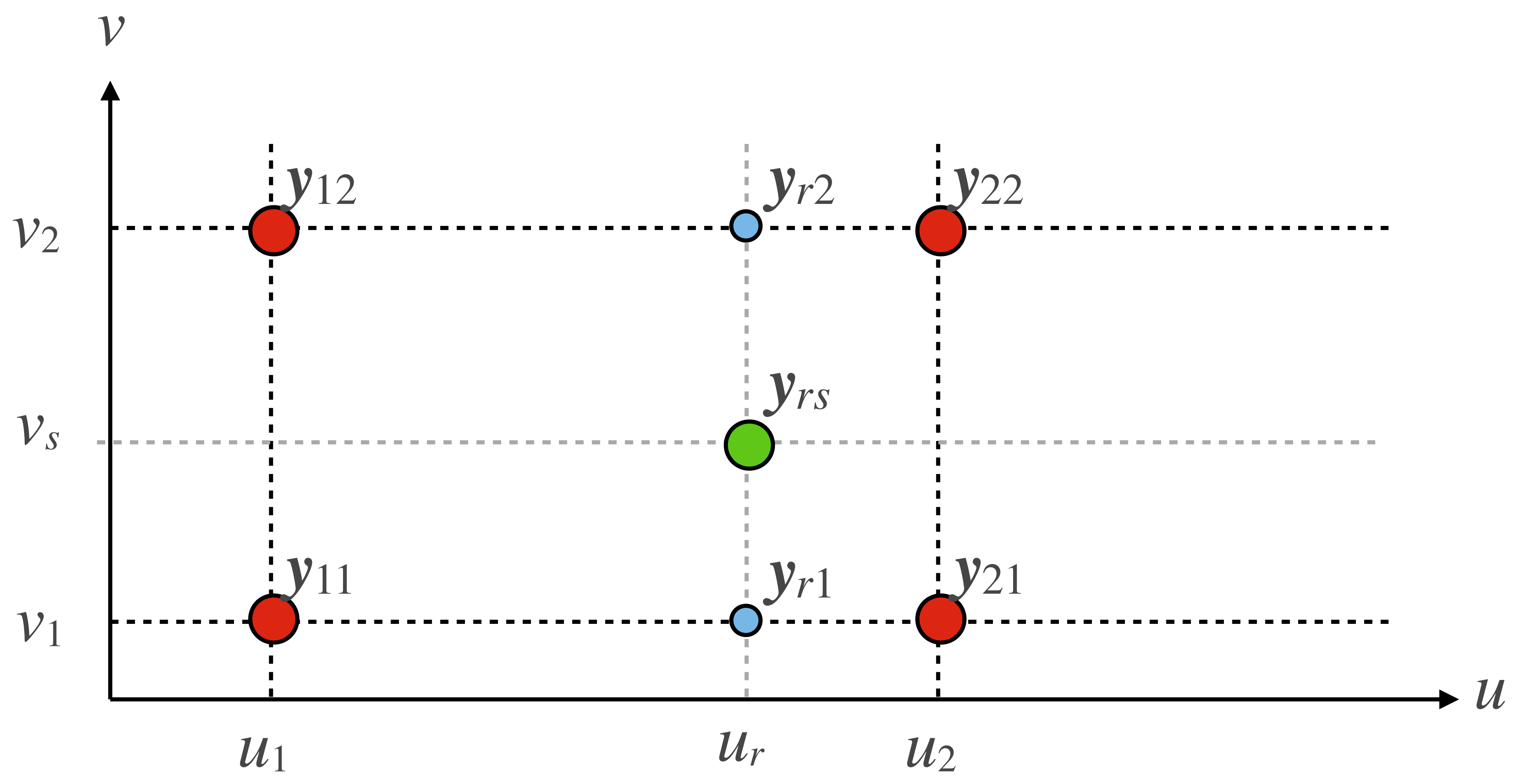


Wikimedia Commons public domain

Bilinear Interpolation in 2D

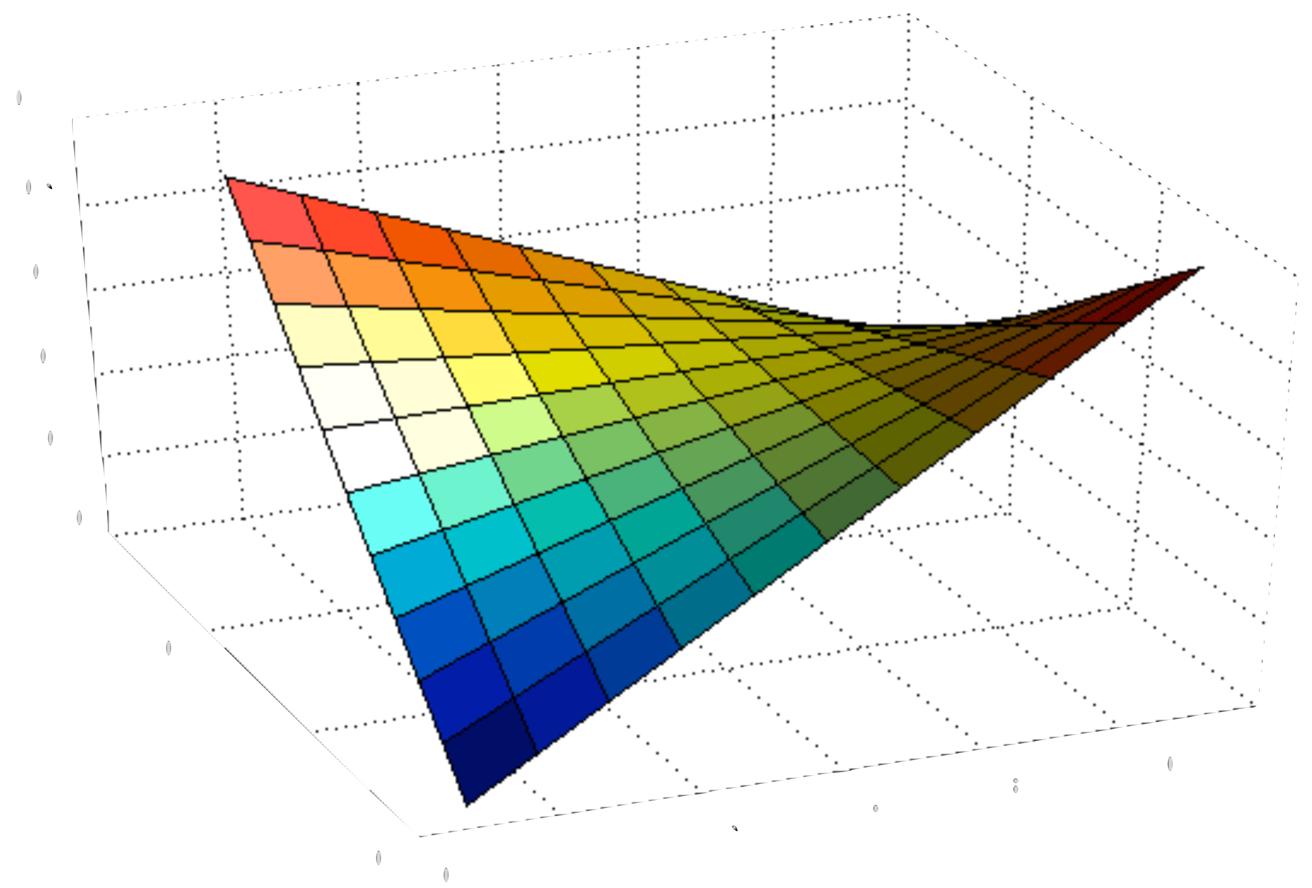
- Interpolation between four grid corners (u_1, v_1), (u_2, v_1), (u_1, v_2) and (u_2, v_2)
  - Input data points $\boldsymbol{x}_{ij}$ are parametrized in two dimensions, hence $\boldsymbol{x}_{ij} \rightarrow (u_i, v_j)$
  - Corresponding output values given by $\boldsymbol{y}_{ij}$
- New data point at parameters (u_r, v_s) with output value $\boldsymbol{y}_{rs}$ given by two successive linear interpolations

$$\boldsymbol{y}_{r2} = \boldsymbol{y}_{12} + (\boldsymbol{y}_{22} - \boldsymbol{y}_{12}) \cdot \frac{u_r - u_1}{u_2 - u_1}$$



$$\boldsymbol{y}_{r1} = \boldsymbol{y}_{11} + (\boldsymbol{y}_{21} - \boldsymbol{y}_{11}) \cdot \frac{u_r - u_1}{u_2 - u_1}$$

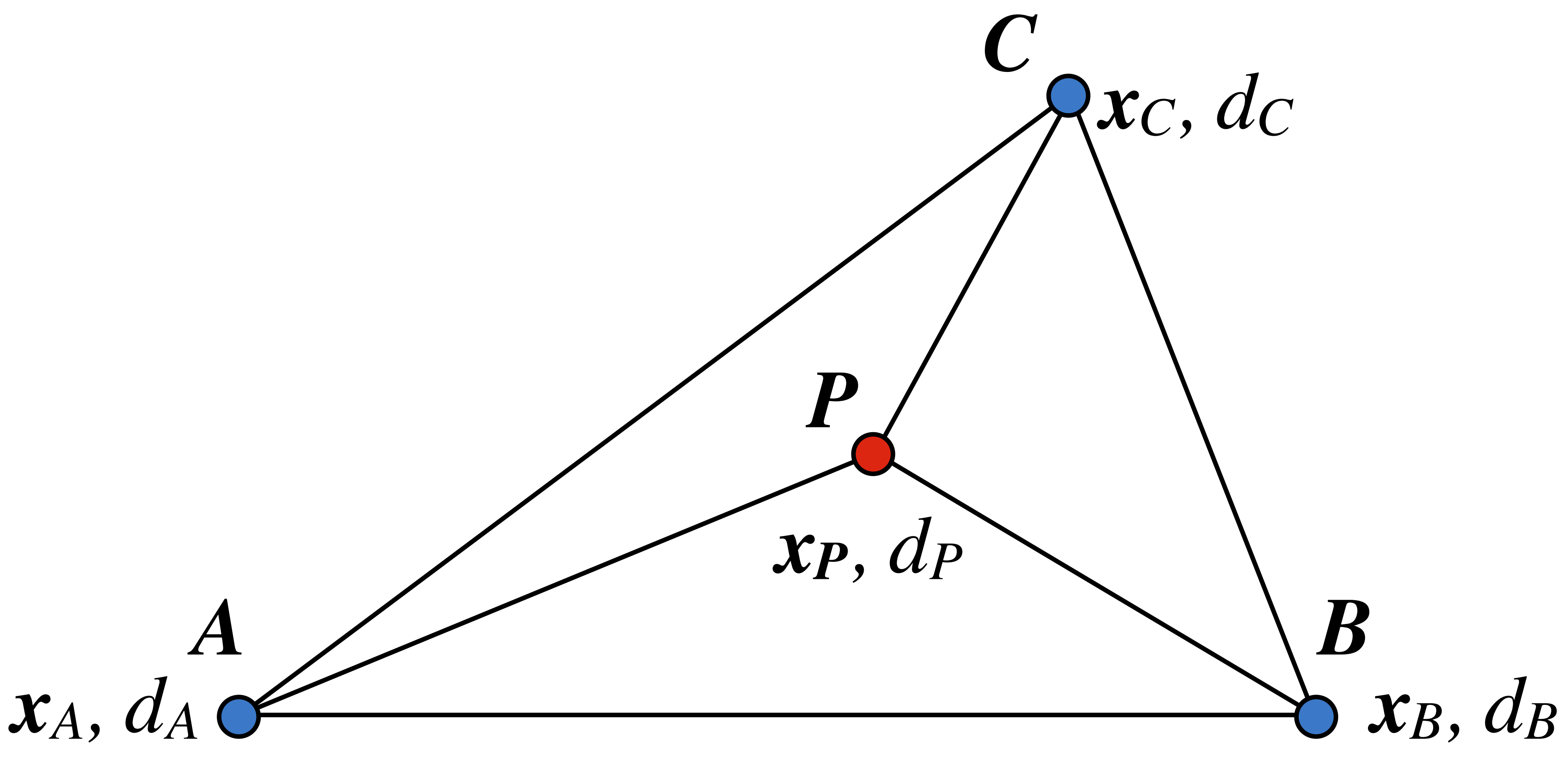
$$\boldsymbol{y}_{rs} = \boldsymbol{y}_{r1} + (\boldsymbol{y}_{r2} - \boldsymbol{y}_{r1}) \cdot \frac{v_s - v_1}{v_2 - v_1}$$



Piecewise Linear Interpolation of Irregular Sample Points in 2D

- Barycentric coordinates $\boldsymbol{\lambda}$ proportional to the signed triangle areas formed by $\boldsymbol{P}$ and corners $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$
  - $\boldsymbol{x_P} = \boldsymbol{\lambda_1\, x_A + \lambda_2\, x_B + \lambda_3\, x_C}$
  - with $\boldsymbol{\lambda_1 + \lambda_2 + \lambda_3 = 1.0}$

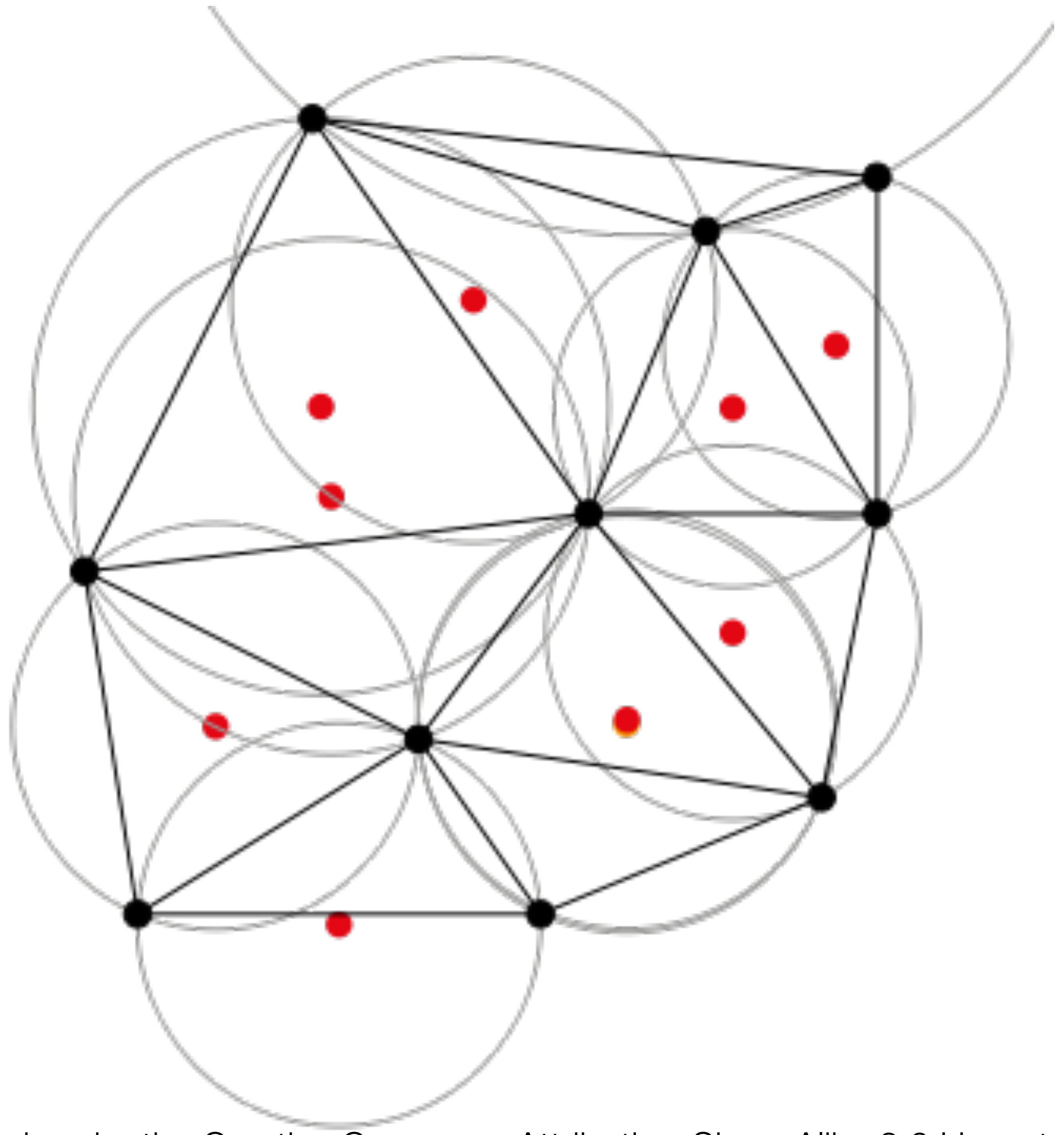
$$\vec{\lambda} = f(\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C)$$



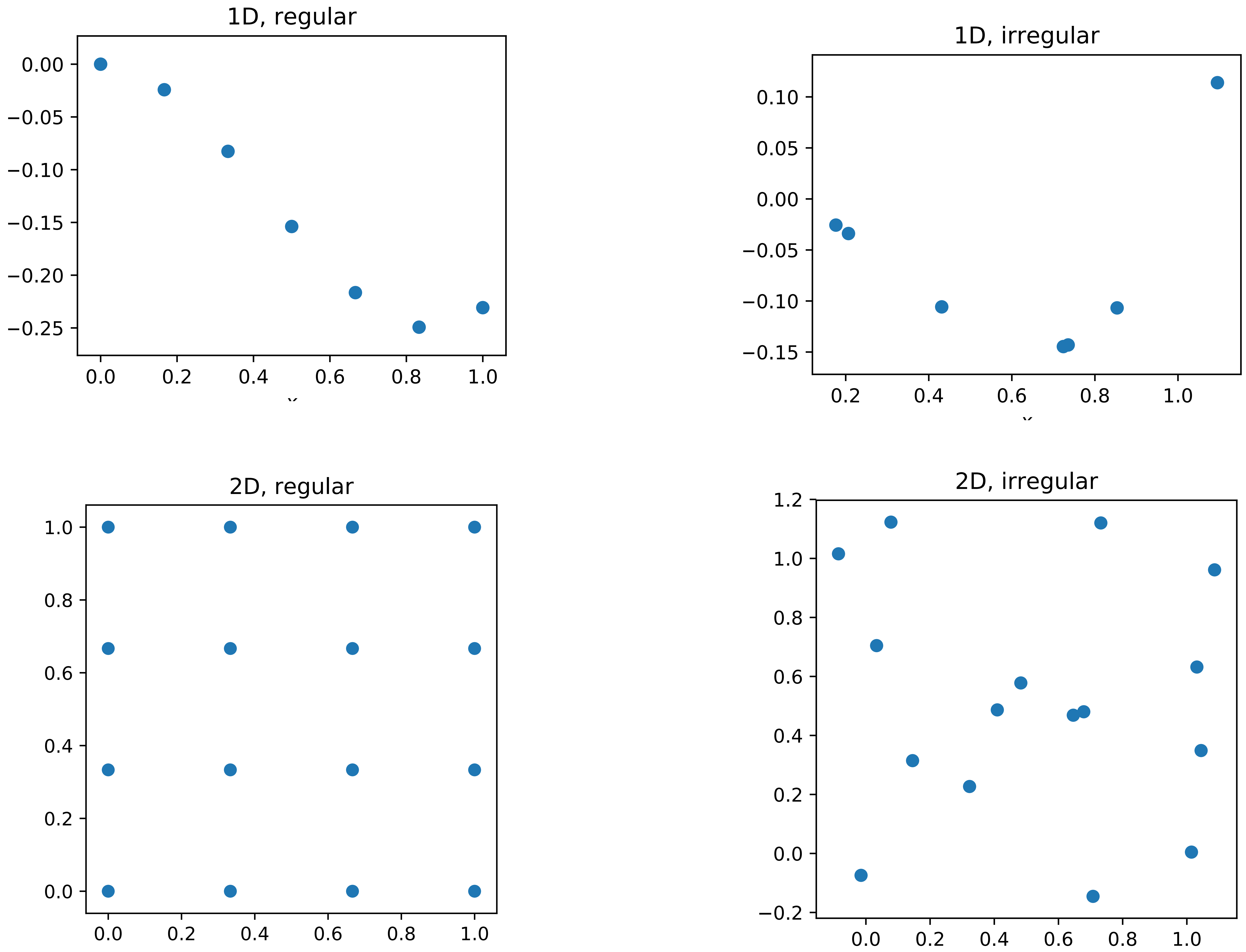
$$d_P = \lambda_A \cdot d_A + \lambda_B \cdot d_B + \lambda_C \cdot d_C$$

- Linear interpolation of any corner attribute d within a triangle defined by barycentric interpolation

- Scattered 2D points can be triangulated and linearly interpolated
  - e.g. using Delaunay triangulation
    - extension to higher dimensions as well



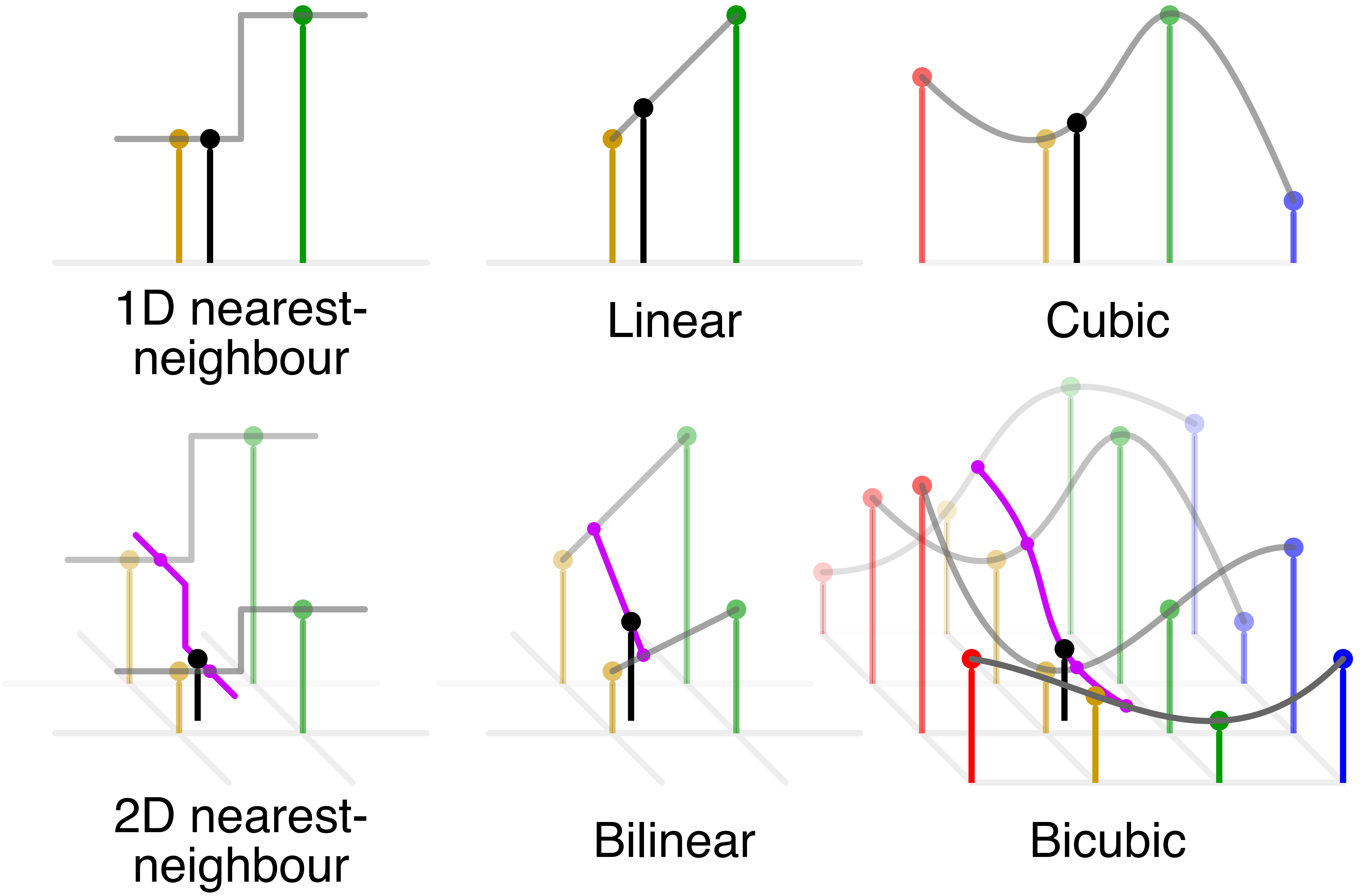
Regular and Irregular 1D and 2D Cases



Smooth Interpolation

Smooth Interpolation

- Continuity may not only be important for the interpolated function value but also with respect to the function's slopes
- Higher order polynomial interpolation or approximation methods support (any) desired continuity levels
    - Interpolated data distribution can be derived analytically
        - for analyzing slope, curvature, edges etc.

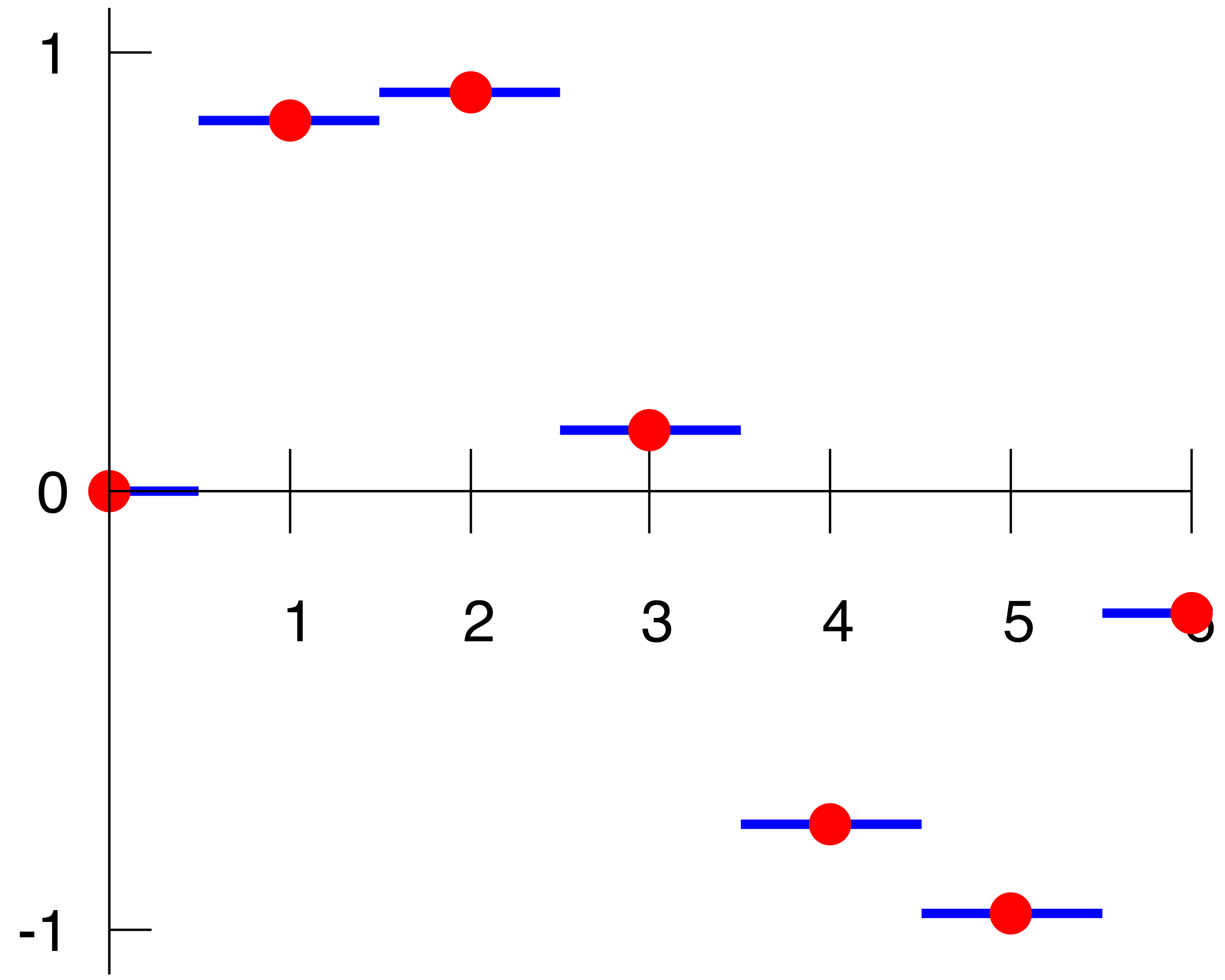


Discontinuity is bad: e.g. infinitely high frequencies appear (remember lecture on sampling & quantization)

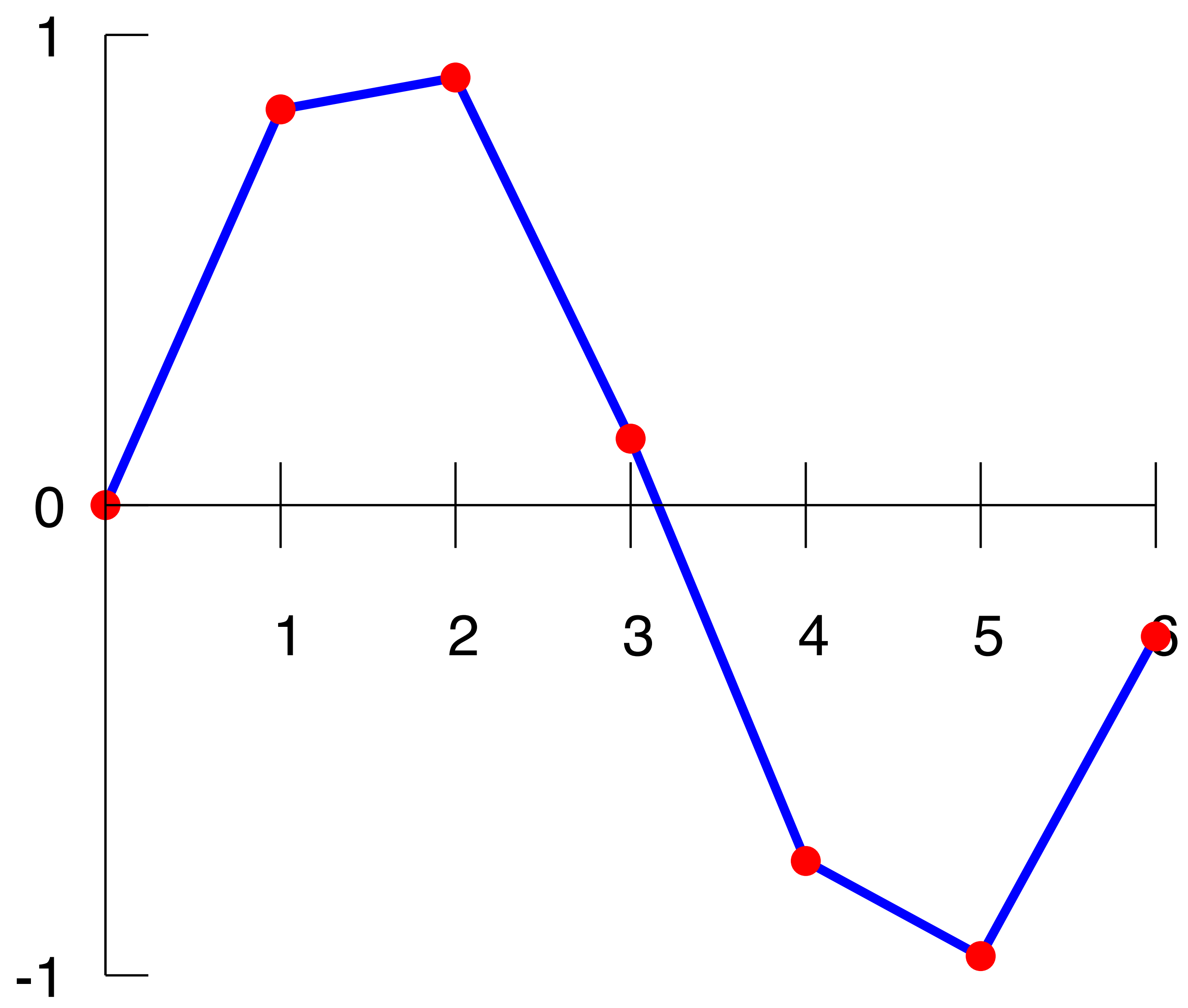
Review: Piecewise Constant or Linear 1D Interpolation

- Piecewise constant interpolation sets $f(\boldsymbol{x})$ to the value $\boldsymbol{y}_i$ of nearest element i, with smallest distance $|\,\boldsymbol{x} - \boldsymbol{x}_i\,|$

- In 1D, linear interpolation computes $f(\boldsymbol{x})$ from nearby values $\boldsymbol{y}_i$ and $\boldsymbol{y}_{i+1}$ for $\boldsymbol{x}_i \leq \boldsymbol{x} \leq \boldsymbol{x}_{i+1}$ as



Wikimedia Commons public domain



Smooth Polynomial Interpolation in 1D

- Find polynomial $p(\boldsymbol{x})$ such that:

$$p(\boldsymbol{x}_0) = \boldsymbol{y}_0$$

$$p(\boldsymbol{x}_1) = \boldsymbol{y}_1$$

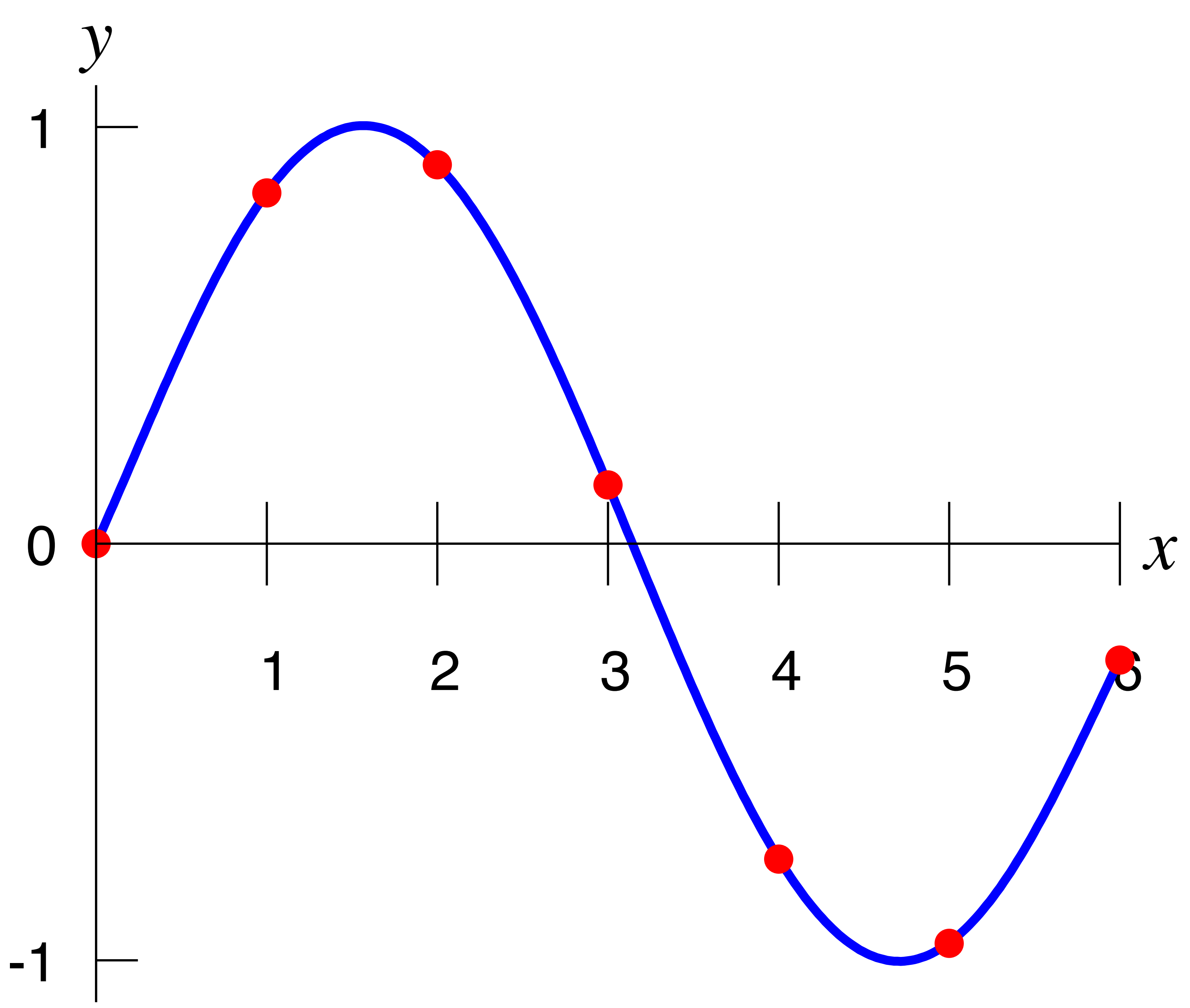
$$p(\boldsymbol{x}_2) = \boldsymbol{y}_2$$

$$p(\boldsymbol{x}_3) = \boldsymbol{y}_3$$

- E.g. cubic polynomial $p(\boldsymbol{x}) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$
  - Several equations of the form $\boldsymbol{y_i} = a_0 + a_1 x_{\boldsymbol{i}} + a_2 x_{\boldsymbol{i}}^2 + a_3 x_{\boldsymbol{i}}^3$
  - Solve system of four equations with four unknowns a_i and four given pairs (x_i, y_i)
- Matrix form:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \boldsymbol{y} = \mathbf{X} \cdot \boldsymbol{a} \qquad \Rightarrow \boldsymbol{a} = \mathbf{X}^{-1} \cdot \boldsymbol{y}$$

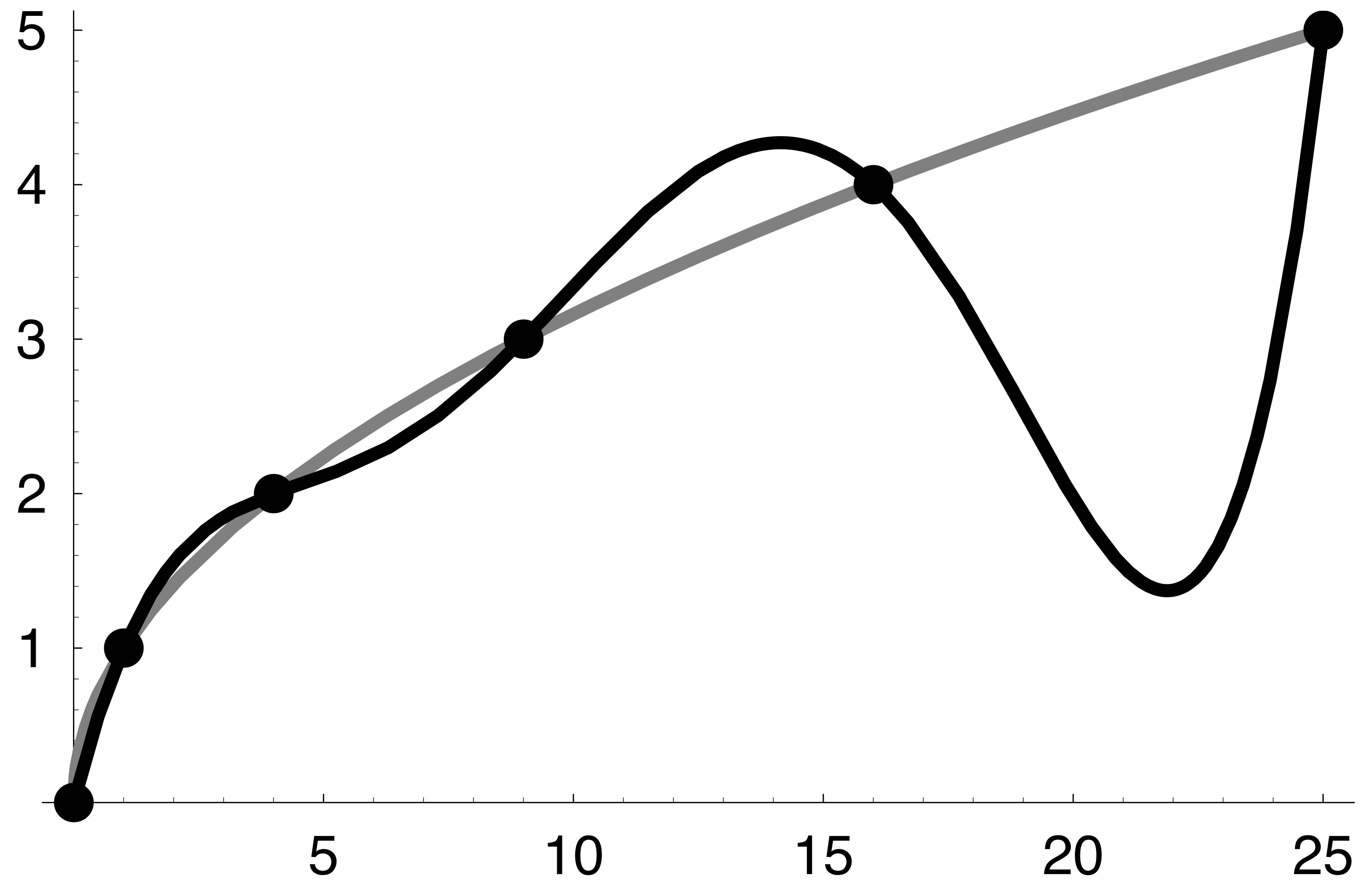
Vandermonde matrix



Wikimedia Commons public domain

Overfitting

- Given any number of n data points (x_i, y_i) a degree n-1 polynomial can be fitted to exactly interpolate them
- Forcing strict interpolation can be dangerous
  - Overfitting can lead to unwanted and fatal oscillations
- Hence careful controlled approximation may be more useful than strict interpolation



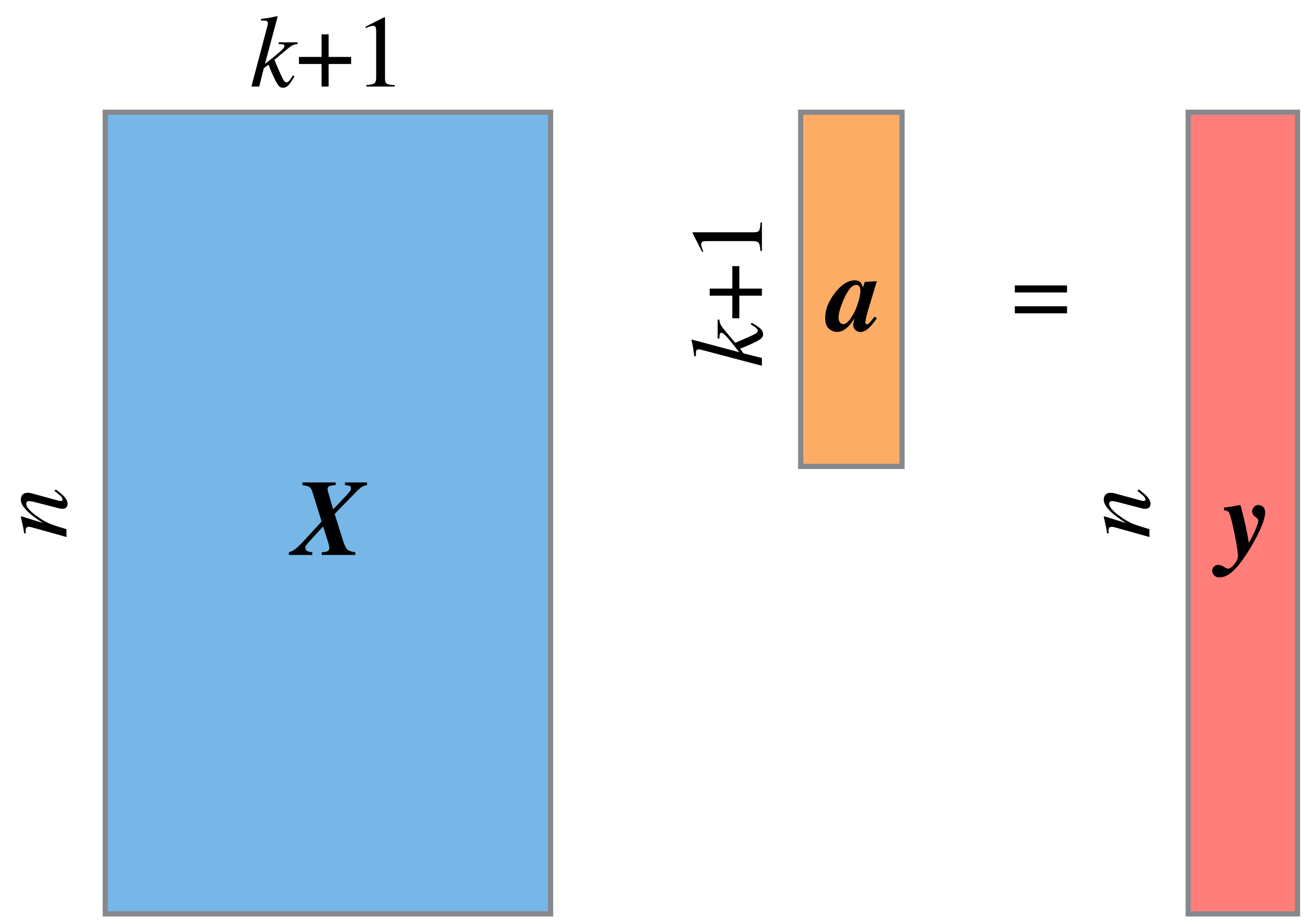
Fitting degree-4 polynomial to $\sqrt{x}$

Polynomial Least-Squares Approximation

- In contrast to interpolation, polynomial approximation is more reliable and typically leads to robust data fitting results
- Task: Given n data points $(\boldsymbol{x_i}, \boldsymbol{y_i})$, find a fixed degree k polynomial $p(\boldsymbol{x})$ that for each $p(\boldsymbol{x_i})$ as close as possible to $\boldsymbol{y_i}$
- Leads to overdetermined system of equations $\boldsymbol{X} \cdot \boldsymbol{a} = \boldsymbol{y}$
  - Tall matrix $\boldsymbol{X}$ and value vector $\boldsymbol{y}$
- Normal equations $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} \cdot \boldsymbol{a} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$ can be solved efficiently
  - Using Gaussian elimination or SVD
- Resulting polynomial is the best least squares approximation

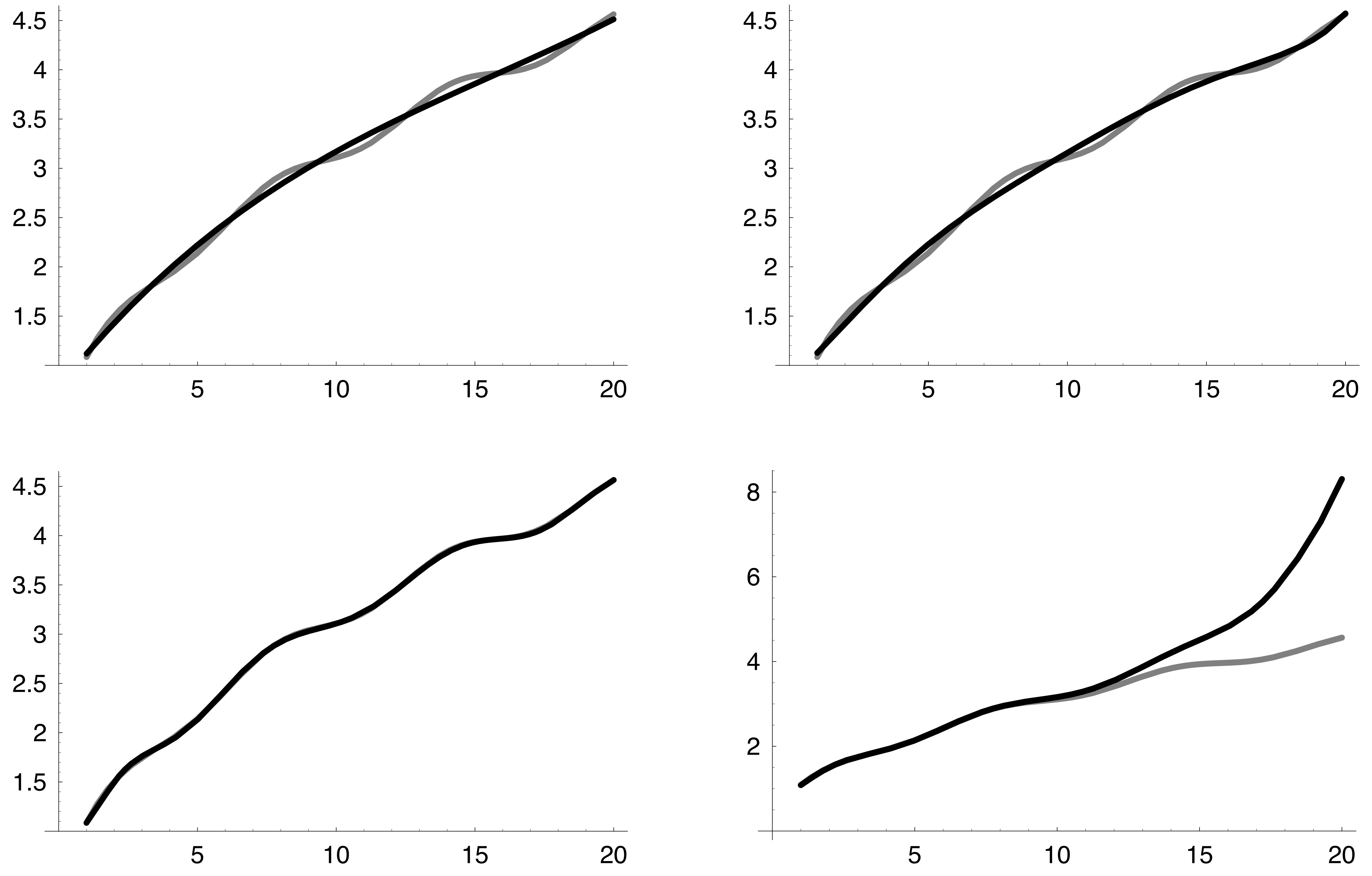
$$\varepsilon = \sum_{i=0}^{n-1} (y_i - p(x_i))^2$$

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ & \vdots & \vdots & \\ 1 & x_{n-2} & x_{n-2}^2 & x_{n-2}^3 \\ 1 & x_{n-1} & x_{n-1}^2 & x_{n-1}^3 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{bmatrix}$$



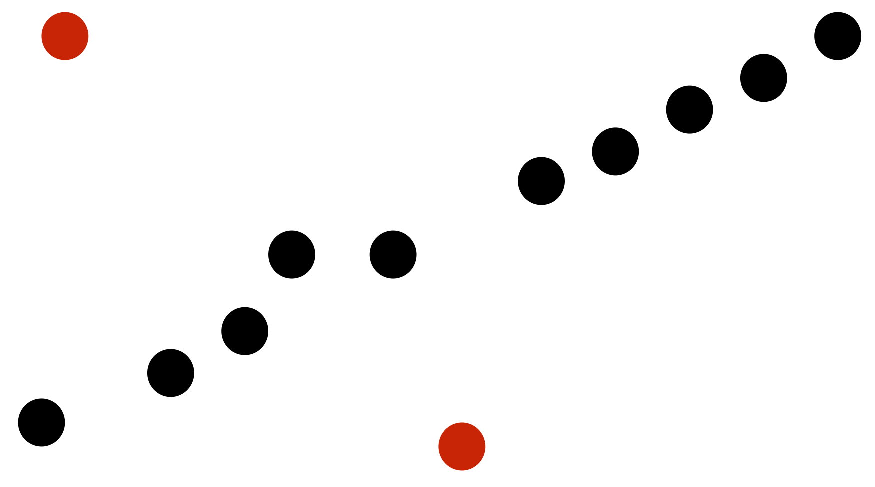
Approximation Problems

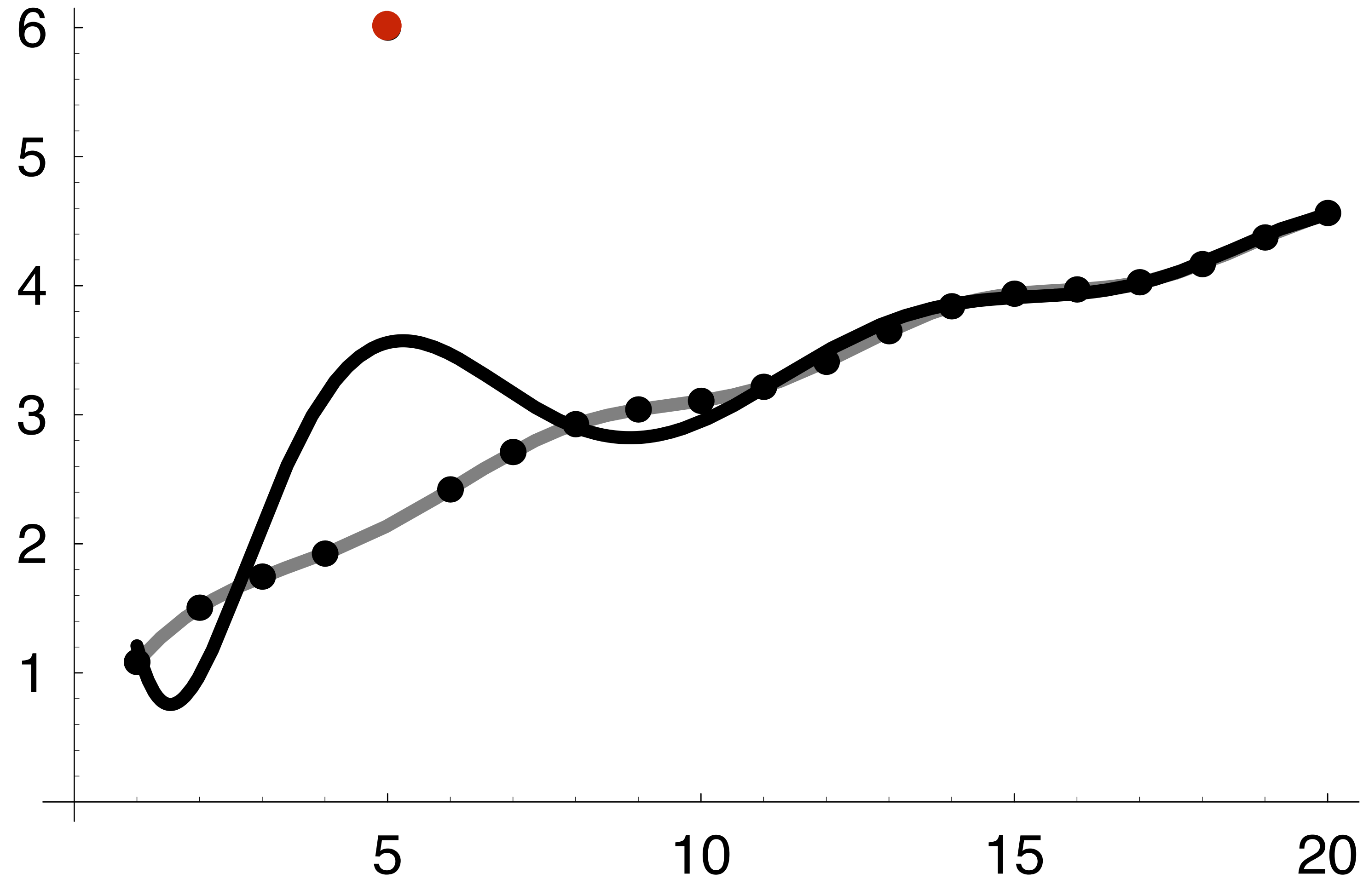
- Approximation of $\sqrt{x} + 0.1\,\sin(x)$
  - For degree $n=3$, $n=6$, $n=10$ and $n=11$
  - Instabilities may occur
    - due to high condition number of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$



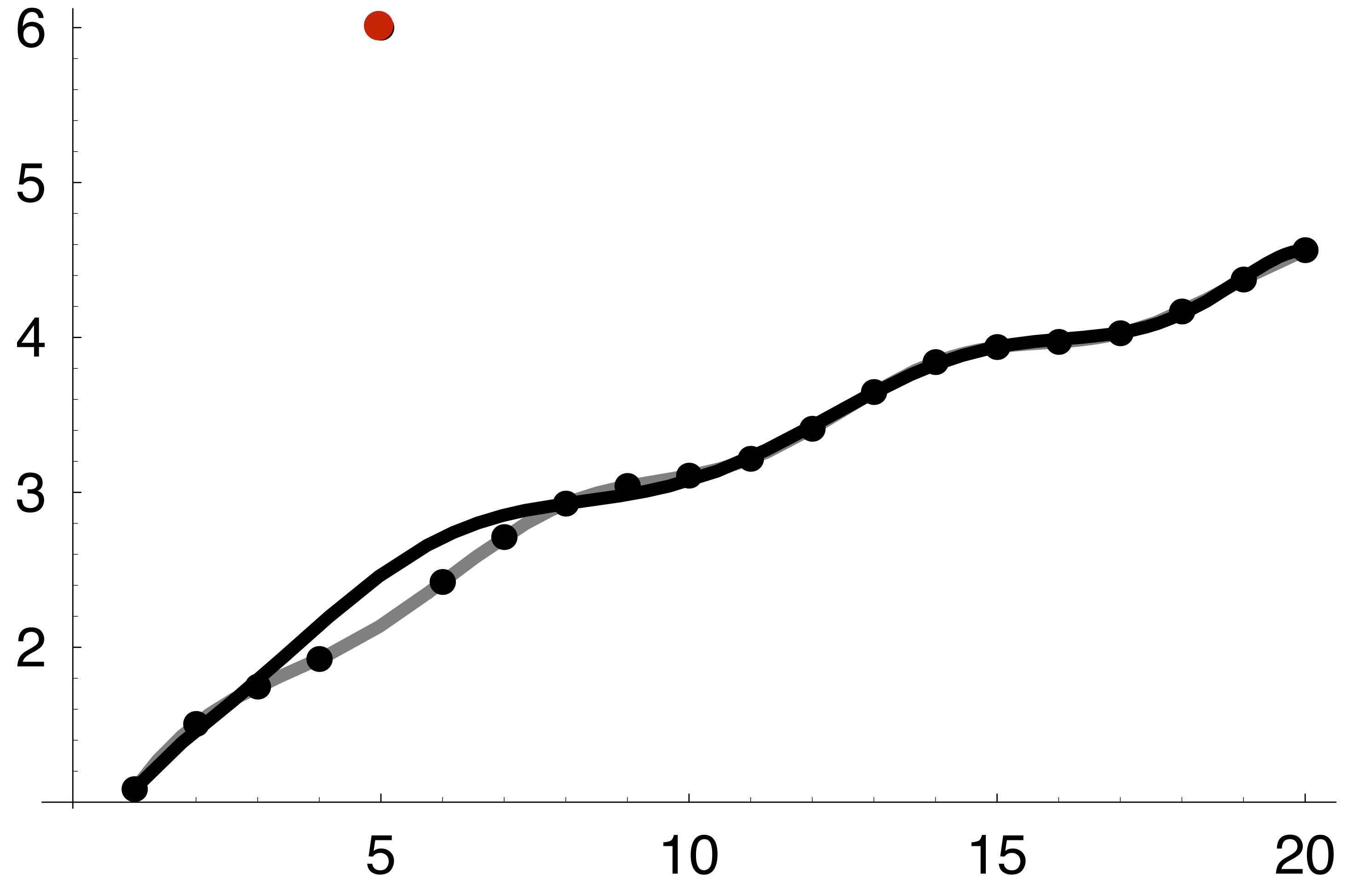
Outliers and Weighted Interpolation

- An outlier is a measurement that is less reliable or accurate
- We may know beforehand what samples are outliers
  - If we do not know, there are techniques to estimate if a sample is an inlier or outlier
- Outliers may severely affect data fitting
  - Weighted least-squares tries to follow the inliers more





Unweighted



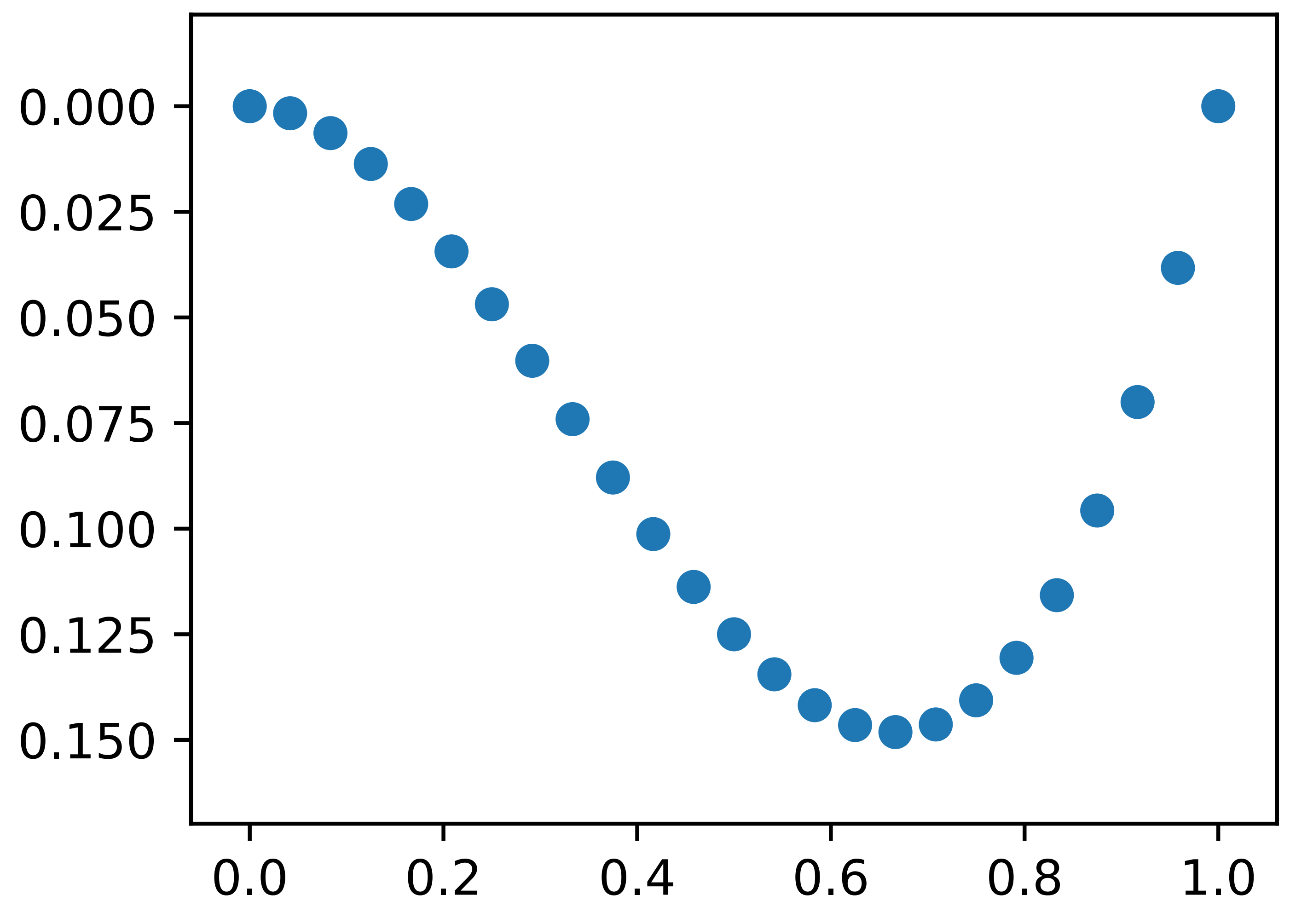
Weighted

Important

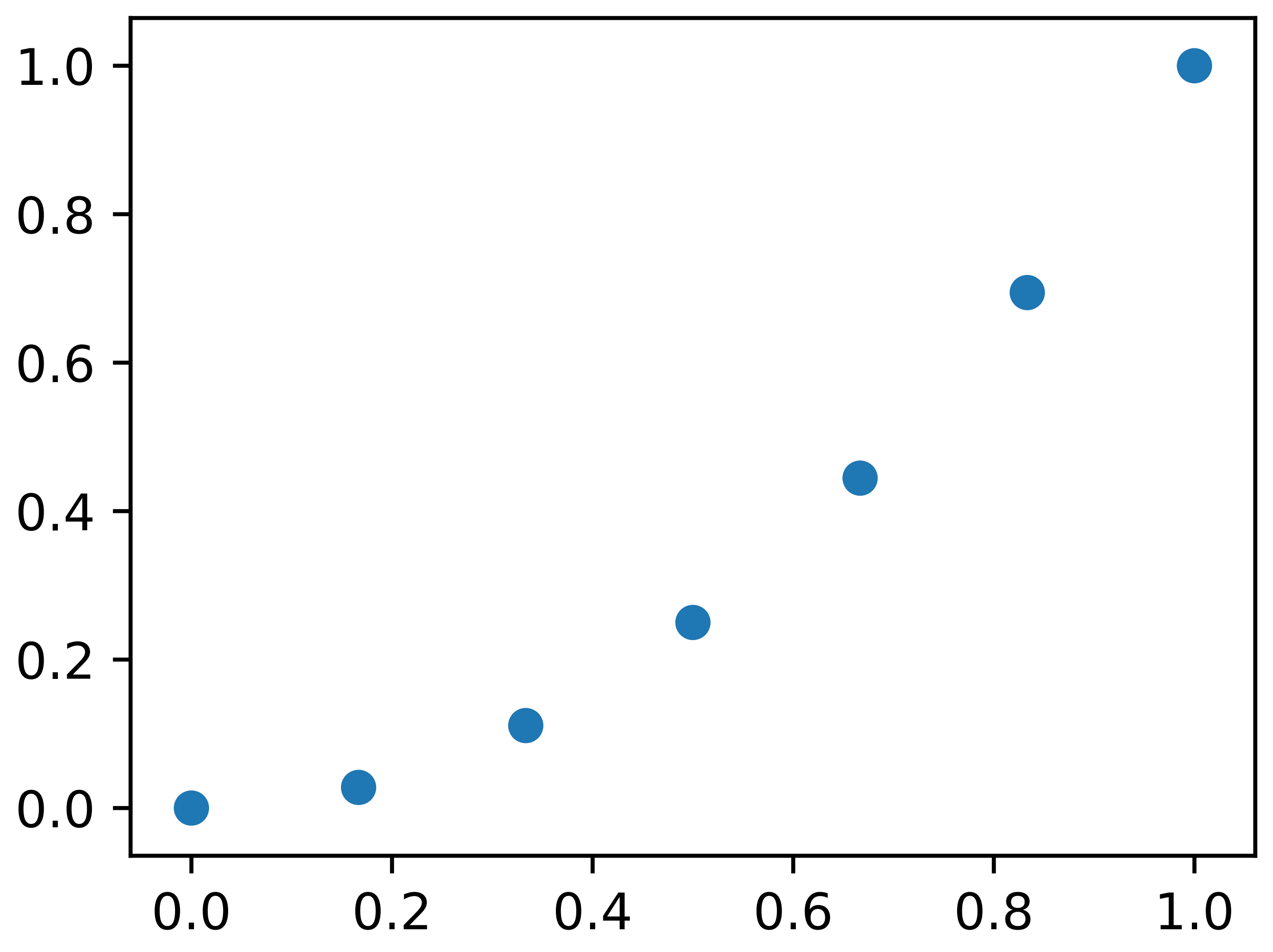
Use the right tool for the right problem!

- One has to know what interpolation fits what type of data
- Piecewise-constant interpolation: simple; no overfitting; good if enough samples available
- Low-degree polynomials: smoother; faster; less variance; less overfitting; less fitting
- High-degree polynomials: bumpier; slower; more variance; more overfitting; more fitting
- **Make sure you understand each technique's advantages, disadvantages and risks**

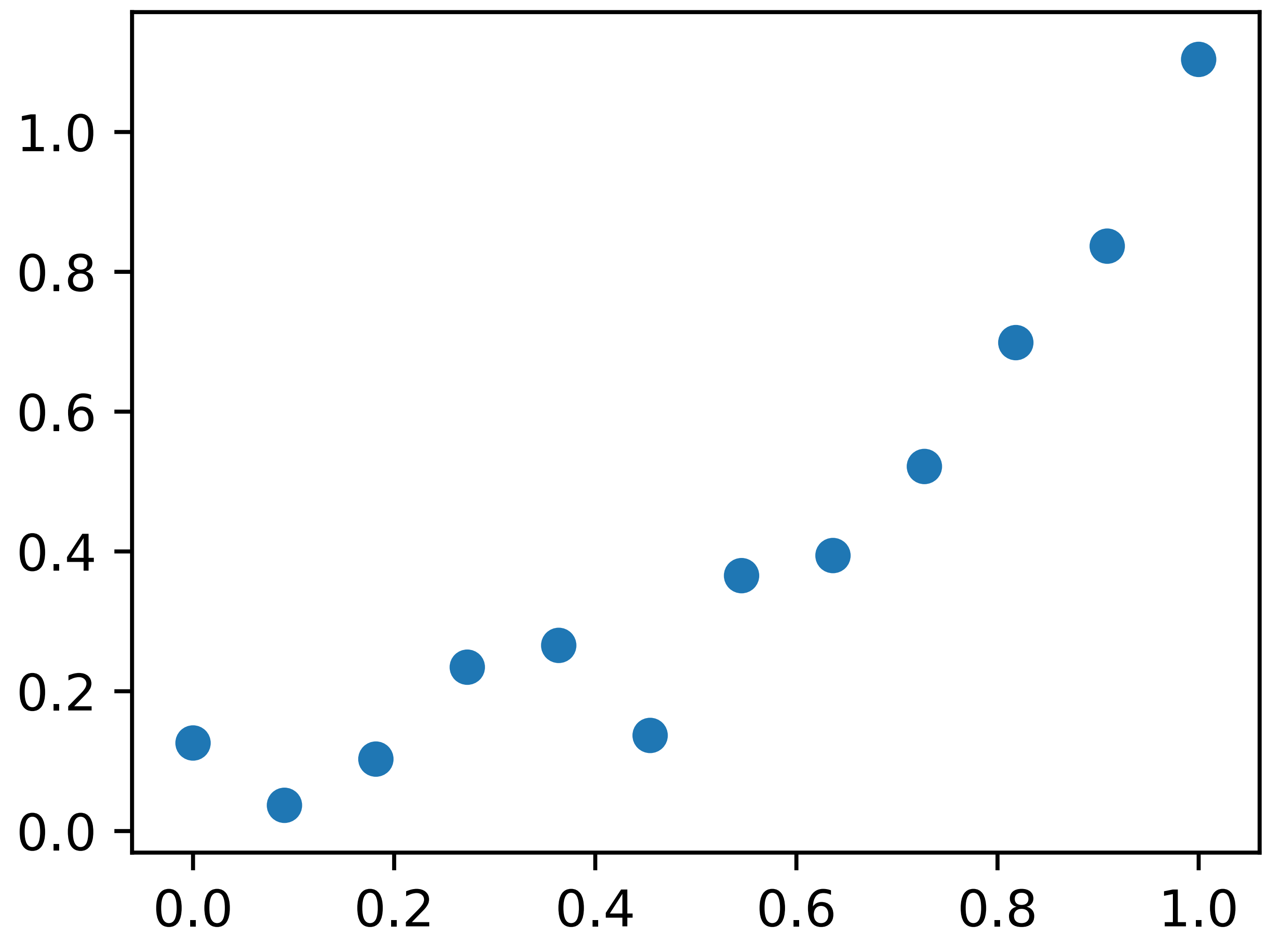
Examples



Piecewise would work



Low-degree exact



Low-degree least-squares

Recap

- **Sparse data:** set of discrete, scattered sample points (x, y)
- **Piecewise interpolation:** interpolation depends only on neighboring samples
- **Smooth interpolation:** function’s derivative (optionally high-order derivatives too) are continuous
- **Exact interpolation:** #points = #unknowns → interpolator passes *through* the given points
- **Least-squares interpolation:** #points > #unknowns → interpolator passes *near* the given points
- **Overfitting:** when a too complex model fits too poorly the true model and too well the samples
- **Outlier:** isolated noisy/unreliable measurement

- Required textbook Chapter(s): 8.1 - 8.5