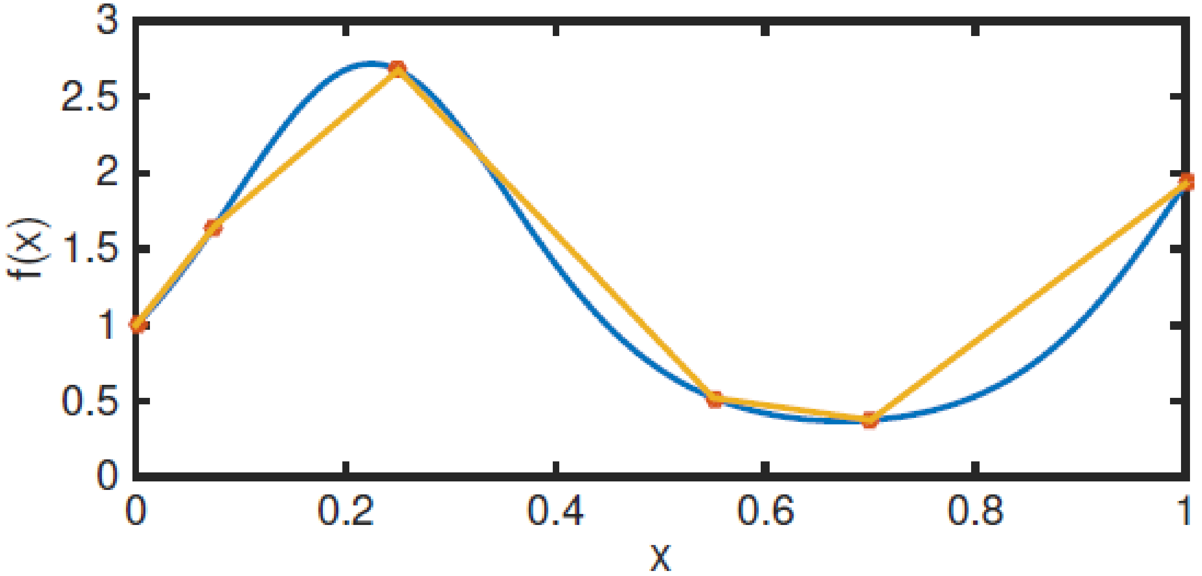
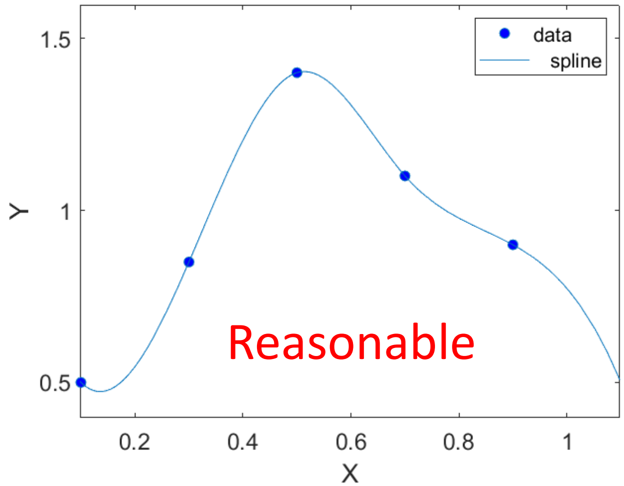
Interpolation

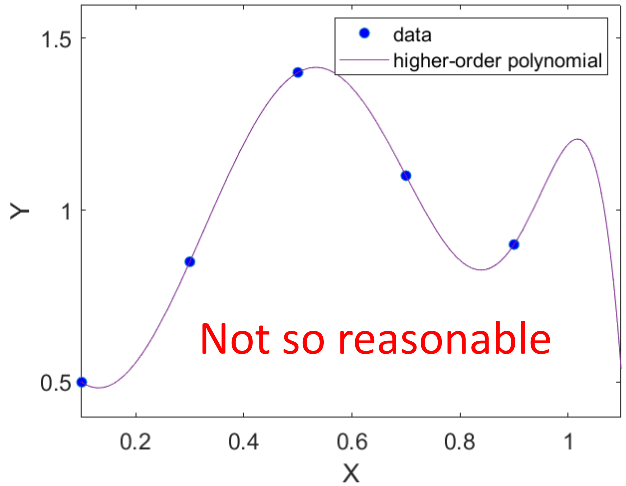


Interpolation

- Building block for various complex algorithms
  - Differentiation
  - Integration
  - Solution of differential equations
  - Signal processing
    - Trigonometric interpolation
  - Approximation theory
    - Approximate discrete data
    - Complex function
- Key idea
  - Find a curve that passes through the data points
  - Data might come from any field
- We have two approaches to interpolation
  - Polynomial interpolation
  - Piecewise interpolation

In interpolation, function (interpolant) must match the data exactly





$(x_i, y_i), \qquad i = 1, \dots m, \qquad$ with $x_1 < x_2 < \cdots x_m$

we seek a function $f: \mathbb{R} \to \mathbb{R}$ such that $\quad f(x_i) = y_i, \qquad i = 1, \dots m$

We call function f the interpolant.

For complex interpolation problems additional information might be included

- Slope of interpolant at data points
- Monotonicity
- Convexity

Function Fitting

Fitting the data exactly is not always desirable

- When data have uncertainty or sources of error

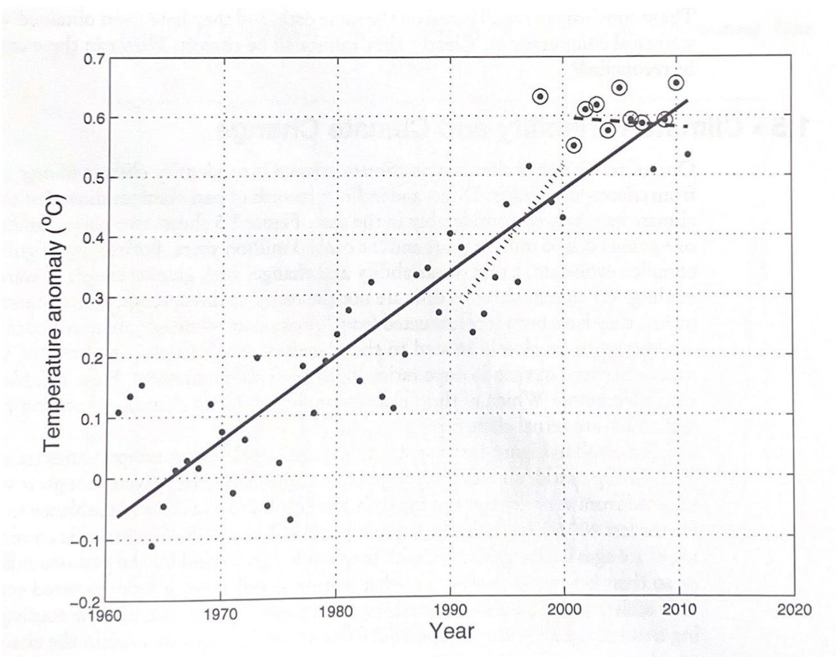
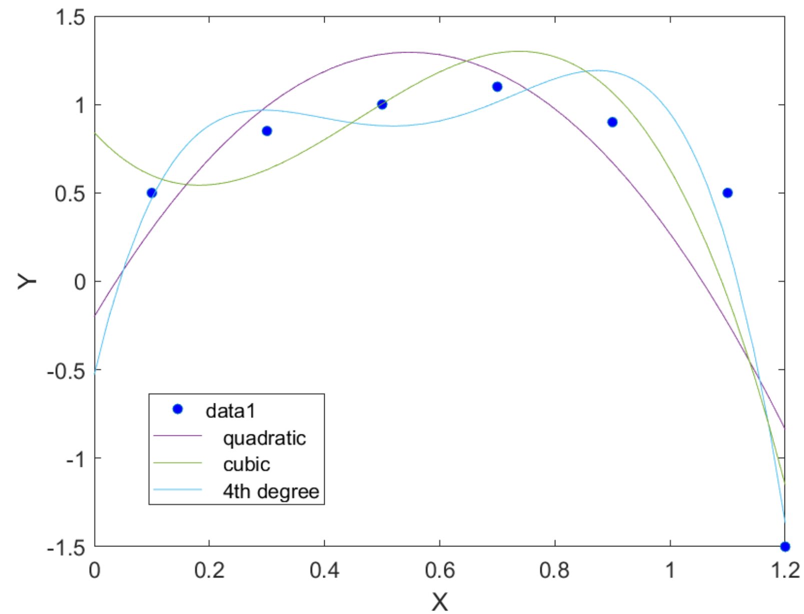


Figure 1.4. *Global mean temperature anomalies for the period 1960–2011 relative to the average global mean temperature since 1880, with various trend lines and records.*



A common approach is method of least squares (will be covered separately)

Existence & Uniqueness of Interpolant

For a given set of data points (t_i, y_i), $\quad i = 1, \dots m$,

an interpolant is chosen from the space of functions spanned by a suitable set of *basis functions* $\phi_1(t), \dots \phi_n(t)$

Interpolating function f is chosen to be a linear combination of the basis functions

$$f(t) = \sum_{j=1}^{n} x_j \phi_j(t),$$

where the parameters x_j are to be determined. Additionally, we require f to interpolate the data (t_i, y_i)

$$f(t_i) = \sum_{j=1}^{n} x_j \phi_j(t_i) = y_i, \qquad i = 1, \dots m,$$

which is a system of linear equations that can be written in matrix form as

$$\boxed{\mathbf{A}x = y}$$

$$\boxed{\mathbf{A}x = y}$$

y is a vector of size *m* composed of the known data values y_i

x is a vector of size *n* are the unknown parameters to be determined

A is an $m \times n$ *basis* matrix where the entries are given by $a_{ij} = \phi_j(t_i)$. In other words, a_{ij} is the value of the *j*th *basis* function evaluated at the *i*th data point

If $n = m$ data points can be fit exactly. Need to solve a system of linear equations. Assuming A is a nonsingular matrix

If $n < m$ overdetermined system (more data than the parameters) → method of least squares. Data cannot be fitted exactly.

If $n > m$ underdetermined system (more parameters than the data) → Interpolant is nonunique. Additional properties can be useful (e.g. degree of smoothness, monotonicity, convexity)

Sensitivity of the parameters x to perturbations in the data depends on the conditioning of **A**. Choice of basis functions determines the conditioning of **A**

Example: Monomial Basis

Basis function: $\phi_j(t) = t^{j-1}, \qquad j = 1, \ldots n$

Interpolant: $p_{n-1}(t) = x_1 + x_2 t + \cdots + x_n t^{n-1}$

$$\mathbf{A}x = \begin{bmatrix} 1 & t_1 \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^{n-1} \\ \vdots & \vdots & \vdots \\ 1 & t_n & t_n^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = y$$

Three data points: (-2,-27), (0,-1), (1,0) → n=3

$$\phi_j(t) = t^{j-1}, \quad j = 1, \dots 3$$

$$f(t_i) = \sum_{j=1}^{n} x_j \phi_j(t_i) = y_i, \qquad i = 1, \dots m,$$

$$f(t_i) = \sum_{j=1}^{3} x_j t^{j-1} = y_i, \qquad i = 1, \dots 3,$$

$$\boxed{p_2(t) = x_1 + x_2 t + x_2 t^2}$$

$$Ax = \begin{bmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -27 \\ -1 \\ 0 \end{bmatrix}$$

Solve by Gauss elimination to find
$x = [-1 \quad 5 \quad -4]^T$

$$\boxed{p_2(t) = -1 + 5t - 4t^2}$$

Issues with $\mathbf{A}x = y$

- Polynomial order grows with more data
- Solving system of linear equations is $\mathcal{O}(n^3)$ operations
- Matrix **A** is often ill-conditioned for high-degree polynomials
  - As the order of polynomial increase, each polynomial becomes less distinguishable (i.e., 9^{th} order looks like 10^{th} order)

Lagrange Interpolation

For a given set of data points (t_i, y_i), $\quad i = 1, \ldots n$, let

$$l(t) = \prod_{k=1}^{n} (t - t_k) = (t - t_1)(t - t_2) \ldots (t - t_n)$$

Define barycentric weights $\quad w_j = \frac{1}{l'(t_j)} = \frac{1}{\prod_{k=1,k\neq j}^{n} (t_j - t_k)}, \qquad j = 1,2, \ldots n.$

Lagrange basis functions for $\mathbb{P}_{n-1}$, also fundamental polynomials are given by

$$l_j(t) = l(t) \frac{w_j}{t - t_j}, \qquad j = 1,2, \ldots n.$$

Basis matrix $\boldsymbol{A}x = y$ turns out to be an identity matrix $\boldsymbol{I}$, meaning that diagonal entries are the data values y

$$\boxed{p_{n-1}(t) = \sum_{j=1}^{n} y_j l_j(t) = \sum_{j=1}^{n} y_j l(t) \frac{w_j}{t - t_j} = l(t) \sum_{j=1}^{n} y_j \frac{w_j}{t - t_j}}$$

Three data points: (-2,-27), (0,-1), (1,0) → n=3

$$\boxed{l(t) = (t - t_1)(t - t_2)(t - t_3) = (t + 2)(t - 0)(t - 1)}$$

$$w_1 = \frac{1}{(t_1 - t_2)(t_1 - t_3)} = \frac{1}{(-2 - 0)(-2 - 1)} = \frac{1}{6}$$

$$w_2 = \frac{1}{(t_2 - t_1)(t_2 - t_3)} = \frac{1}{(0 - (-2))(0 - 1)} = -\frac{1}{2}$$

$$w_3 = \frac{1}{(t_3 - t_1)(t_3 - t_2)} = \frac{1}{(1 - (-2))(1 - 0)} = \frac{1}{3}$$

$$\boxed{p_2(t) = (t + 2)(t - 0)(t - 1)\left(-27\frac{1/6}{t + 2} - 1\frac{-1/2}{t} + 0\frac{1/3}{t - 1}\right)}$$

In-class exercise

Evaluate and plot the following polynomial functions for any t and comment on your results

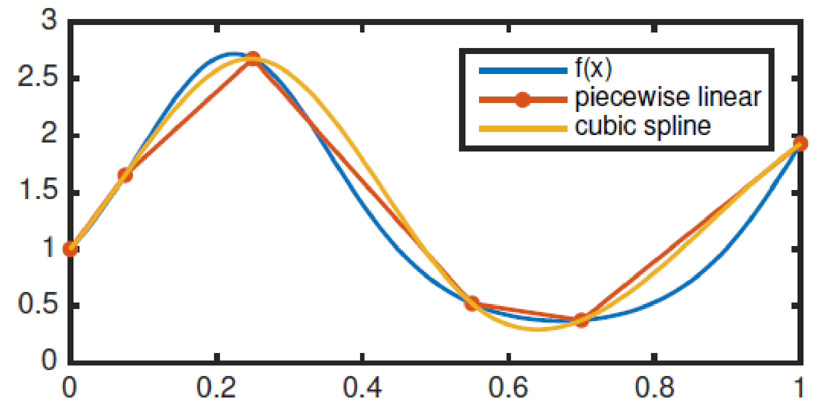
$$\boxed{p_2(t) = -1 + 5t - 4t^2}$$

$$\boxed{p_2(t) = (t+2)(t-0)(t-1)\left(-27\frac{1/6}{t+2} - 1\frac{-1/2}{t} + 0\frac{1/3}{t-1}\right)}$$

Piecewise Polynomial Interpolation

A single polynomial of degree *n* fitted to the entire data can behave poorly as data size grows. Lagrange or monomial basis.

One can use *piecewise* Lagrange polynomials with lower order to sections of the data



The problem with *piecewise* Lagrange polynomials is that it has discontinuous slopes at boundaries of the sections, which causes problems when calculating derivatives at the data points

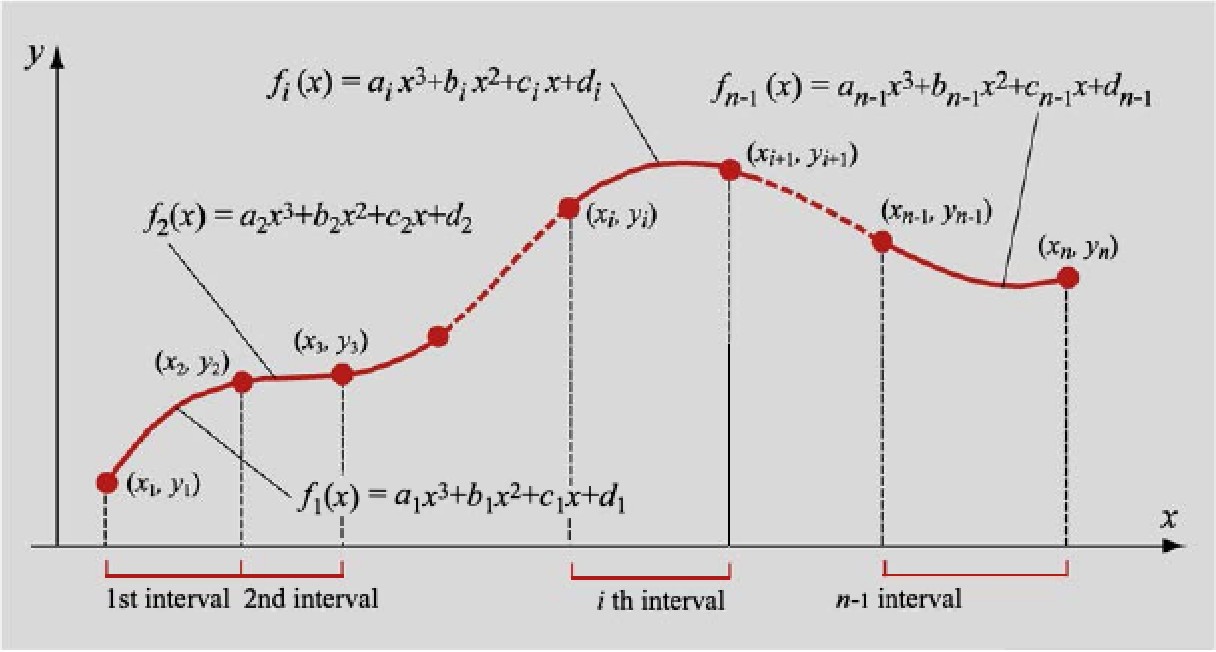
We can address this issue with cubic splines

Cubic Spline Interpolation

A *spline* is a piecewise polynomial of degree *k* that is continuously differentiable *k*-1 times.

A *linear* spline can be continuous, but it is NOT differentiable at data points

A *cubic* spline is continuous and twice differentiable at data points



Cubic Spline Interpolation

Consider $n + 1$ distinct data points (nodes) with
$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ with $x_0 < x_1 < \cdots < x_n$

Note: index starts from 0

Derive a third order polynomial for each interval $f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$

We have $n + 1$ points and n intervals → 4n unknown constants.

We need $4n$ equations or conditions

Cubic Spline Interpolation

1. Function values must be equal at the interior nodes: 2*(n-1) equations

$$f(x_{i-1}) = a_i x^3 + b_i x^2 + c_i x + d_i$$

$$f(x_{i-1}) = a_{i-1} x_{i-1}^3 + b_{i-1} x_{i-1}^2 + c_{i-1} x_{i-1} + d_{i-1}$$

2. First and last function must pass through the end points: 2 equations

$$f(x_0) = y_o \qquad f(x_n) = y_n$$

3. First derivatives at the interior nodes are equal: (n-1) equations

$$f'(x_{i-1}) = f'(x_i)$$

4. Second derivatives at the interior nodes are equal: (n-1) equations

$$f''(x_{i-1}) = f''(x_i)$$

5. Second derivatives at the end nodes are zero: 2 equations

This is known as the *natural* splines

$$f''(x_0) = 0 \qquad f''(x_n) = 0$$

Total of 4n equations/conditions

The fifth condition can be replaced with a *not-a-knot* spline

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

$$f_1'''(x_1) = f_2'''(x_1) \qquad f_{n-1}'''(x_{n-1}) = f_n'''(x_{n-1})$$

$$a_1 = a_2 \qquad a_{n-1} = a_n$$

2 equations!