UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 401 (Fall 2025)

Numerical Methods

Lecture 15: Interpolation Using Piecewise Polynomials

Chong Liu

Department of Computer Science

Nov 10, 2025

Agenda

- Different interpolation methods:

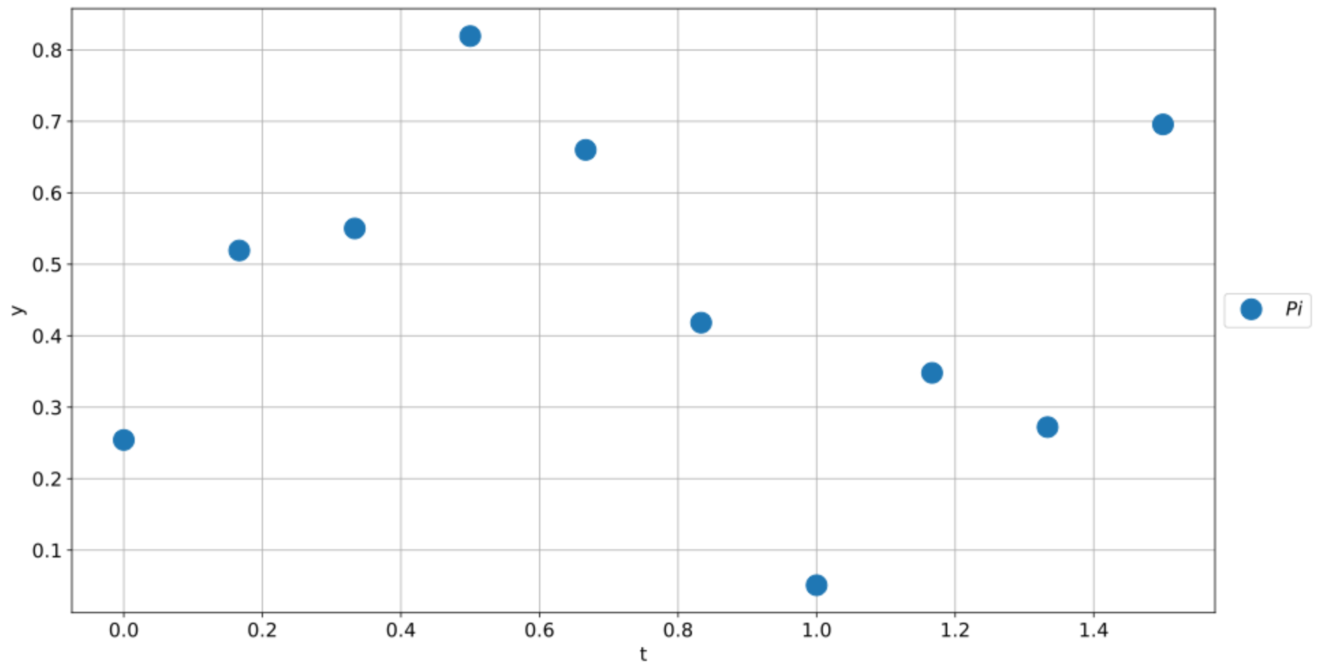
  - Polynomial interpolation: Newton interpolation

  - Piecewise polynomial interpolation
    - Piecewise linear interpolation
    - Quadratic spline interpolation
    - Cubic spline interpolation

Recap: Understanding house price change

- You observe the house price change in the past 15 months
  - You have 10 data points
  - t denotes time
  - y denotes price

- Discussion:
  - What can you do to study the price trend?
  - How can you define a function that describes all these price points?



Recap: Problem setup of interpolation

- For given data
  - $(t_1, y_1), (t_2, y_2), \dots, (t_m, y_m)$ with $t_1 < t_2 < \cdots < t_m$
- determine function $f: R \to R$ such that
  - $f(t_i) = y_i, \forall i = 1, \dots, m$

- f is **interpolating function**, or **interpolant**, for given data.

- f could be function of more than one variable, but let's focus on the 1-dimensional case first.

Newton interpolation

- For given set of data points $(t_i, y_i), i = 1, \ldots, n$, Newton basis functions are defined by

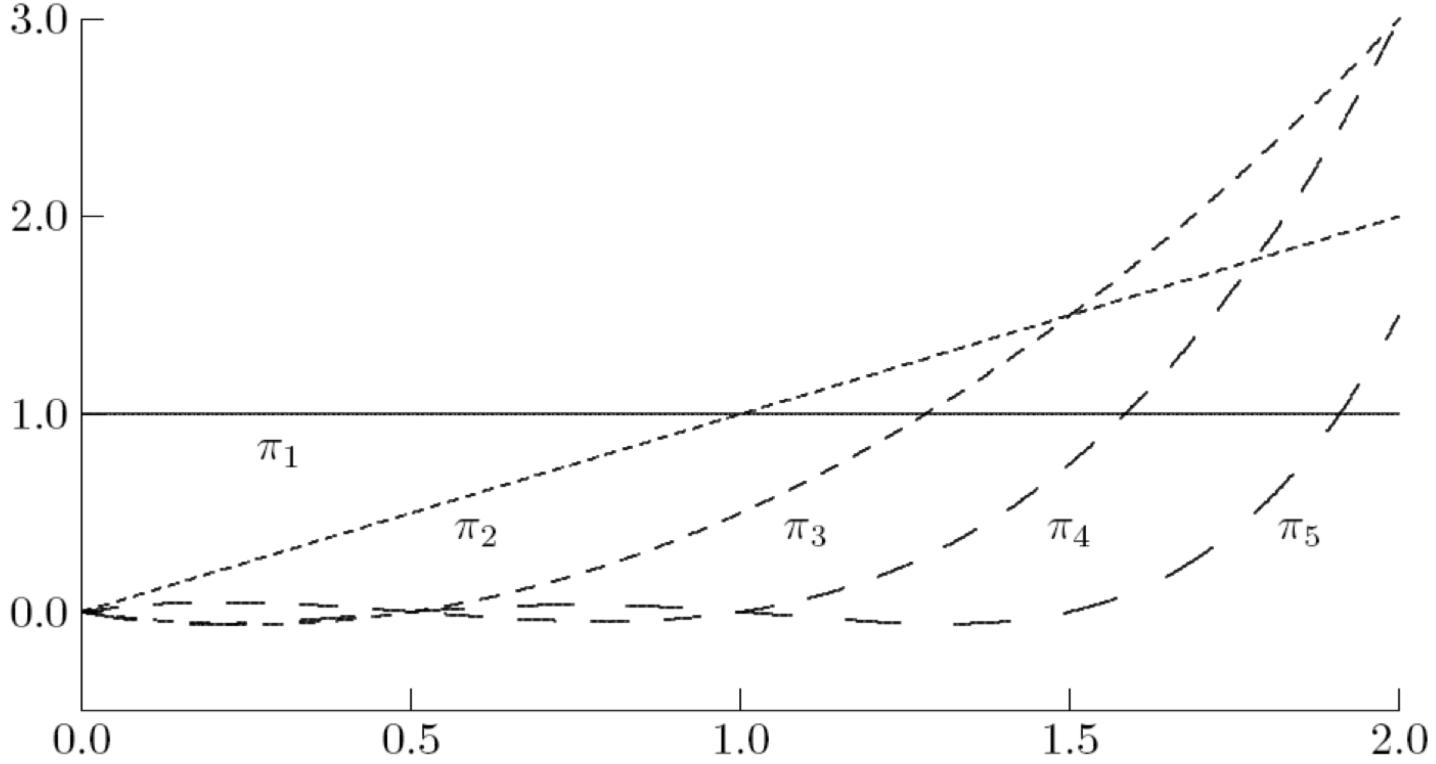
$$\pi_j(t) = \prod_{k=1}^{j-1}(t - t_k), \quad j = 1, \ldots, n$$

- Newton interpolating polynomial has form

$$\begin{aligned} p_{n-1}(t) \quad &= \quad x_1 + x_2(t - t_1) + x_3(t - t_1)(t - t_2) + \\ &\quad\ \cdots + x_n(t - t_1)(t - t_2)\cdots(t - t_{n-1}) \end{aligned}$$

  - For $i < j, \pi_j(t_i) = 0$, so basis matrix A is lower triangular, where $a_{ij} = \pi_j(t_i)$.

Newton basis functions



In-class exercise: Newton interpolation

- Use Newton interpolation to determine interpolating polynomial for three data points $(-2, -27), (0, -1), (1, 0)$

- Solution:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & t_2 - t_1 & 0 \\ 1 & t_3 - t_1 & (t_3 - t_1)(t_3 - t_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -27 \\ -1 \\ 0 \end{bmatrix}$$

$$x = \begin{bmatrix} -27 & 13 & -4 \end{bmatrix}^T$$

$$p(t) = -27 + 13(t+2) - 4(t+2)t$$

Newton interpolation

- Solution x to triangular system $Ax = y$ can be computed by forward-substitution in $O(n^2)$ operations

- Resulting interpolant can be evaluated in $O(n)$ operations

Updating Newton interpolation

- If $p_j(t)$ is polynomial of degree $j-1$ interpolating j given points, then for any constant x_{j+1},

$$p_{j+1}(t) = p_j(t) + x_{j+1}\pi_{j+1}(t)$$

  - is polynomial of degree j that also interpolates same j points
- Free parameter x_{j+1} can then be chosen so that $p_{j+1}(t)$ interpolates y_{j+1},

$$x_{j+1} = \frac{y_{j+1} - p_j(t_{j+1})}{\pi_{j+1}(t_{j+1})}$$

  - Newton interpolation begins with constant polynomial $p_1(t) = y_1$ and then successively incorporates each remaining data point into interpolant

Convergence of interpolation

- If data points are discrete sample of continuous function, how well does interpolant approximate that function between sample points?

If f is smooth function, and p_{n-1} is polynomial of degree at most $n-1$ interpolating f at n points $t_1, \ldots, t_n$, then

$$f(t) - p_{n-1}(t) = \frac{f^{(n)}(\theta)}{n!}(t - t_1)(t - t_2)\cdots(t - t_n)$$

where θ is some (unknown) point in interval $[t_1, t_n]$

Convergence of interpolation

- Theorem:

If $|f^{(n)}(t)| \leq M$ for all $t \in [t_1, t_n]$, and
$h = \max\{t_{i+1} - t_i \; : \; i = 1, \ldots, n-1\}$, then

$$\max_{t \in [t_1, t_n]} |f(t) - p_{n-1}(t)| \leq \frac{Mh^n}{4n}$$

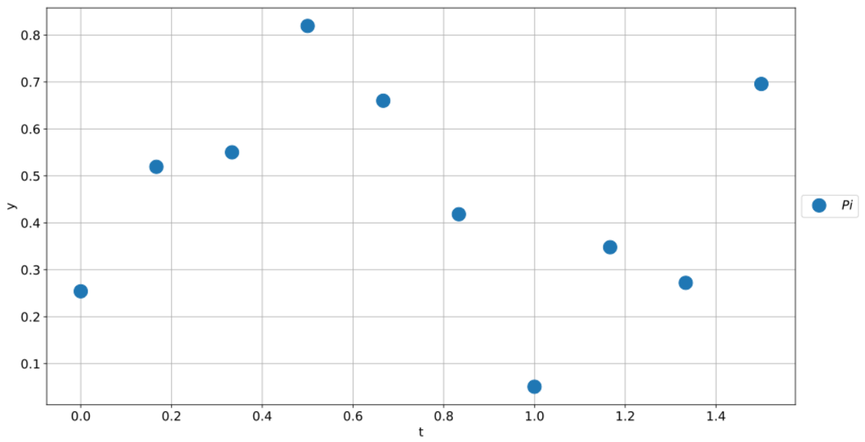
Error diminishes with increasing n and decreasing h, but only if $|f^{(n)}(t)|$ does not grow too rapidly with n

Piecewise polynomial interpolation

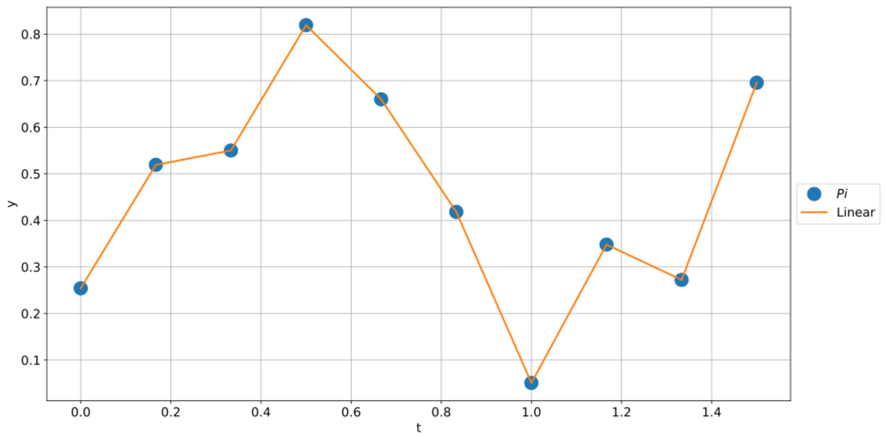
- Motivation:
  - Fitting single polynomial to large number of data points is likely to yield unsatisfactory behavior in interpolant
- Main advantage:
  - Large number of data points can be fit with low-degree polynomials
- How:
  - Given data points (t_i, y_i), different function is used in each subinterval $[t_i, t_{i+1}]$
    - t_i is called knot or breakpoint, at which interpolant changes from one function to another

Piecewise polynomial interpolation

- Discussion: Could you provide an example of a piecewise polynomial interpolation?



- Simplest example is piecewise linear interpolation, in which successive pairs of data points are connected by straight lines
  - Discussion: what are the drawbacks of linear interpolation?



Spline interpolation

- A spline is a smooth piecewise polynomial function.
  - Two popular model:
    - Quadratic spline, Cubic spline
- Quadratic spline interpolation
  - each segment is a **second-degree polynomial** function.
  - Formally, we have data points $(t_i, y_i), i = 1, \dots, n$
  - For each interval $[t_i, t_{i+1}]$, we define a quadratic polynomial
    - $f_i(t) = a_i + b_i(\mathrm{t} - t_i) + c_i\left(t - t_i\right)^2.$
    - There are $n - 1$ such polynomials (one per interval).
  - Discussion: how many coefficients need to be determined? How many equations do we need?
    - $3(n - 1)$

Conditions of quadratic spline interpolation

- The spline passes through all data points:
  - Discussion: How many conditions under this requirement?
    - 2(n-1)
  - $f_i(t_i) = y_i, f_i(t_{i+1}) = y_{i+1}$.
- The spline should be smooth at internal nodes:
  - $f_i'(t_{i+1}) = f_{i+1}'(t_{i+1}), i = 1, \ldots, n-2$.
  - Discussion: How many conditions under this requirement?
    - n-2
- We need one extra equation to close the system. Common choices:
  - Natural: assume $f_1''(t_1) = 0$,meaning the curve starts flat.
  - Or clamped: specify the slope at one endpoint.

Cubic spline interpolation

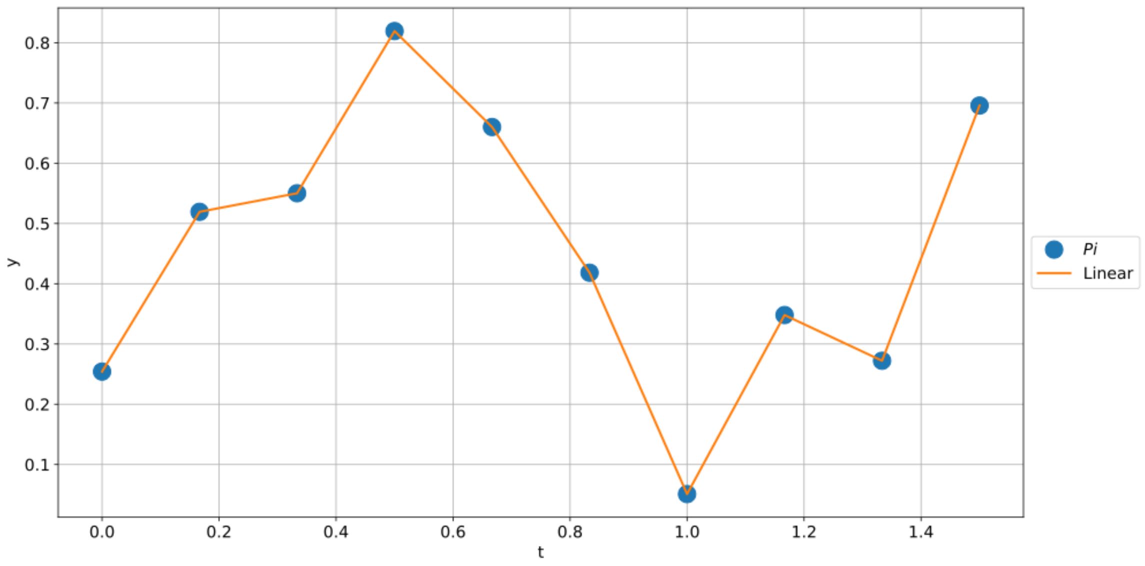
- Each segment is a third-degree polynomial function.
- Formally, we have data points $(t_i, y_i), i = 1, \ldots, n$
- For each interval $[t_i, t_{i+1}]$, we define a quadratic polynomial
  - $f_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3$
  - There are $n - 1$ such polynomials (one per interval).

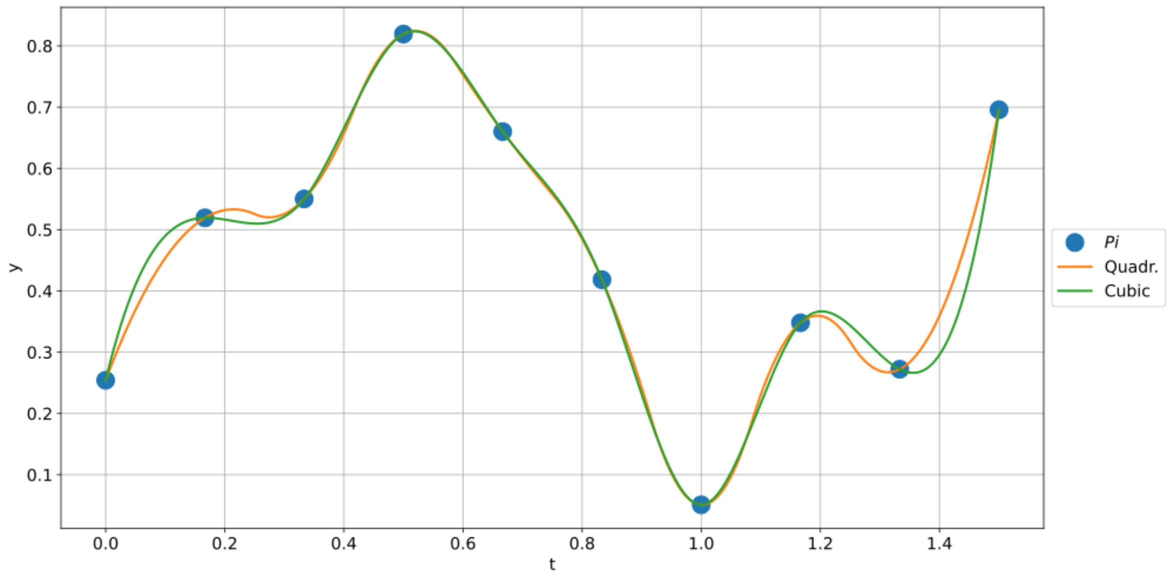
- Discussion: how many coefficients need to be determined? How many equations do we need?
  - $4(n - 1)$

Illustration of piecewise polynomial interpolation (scipy.interpolate)

- Piecewise linear

- Spline – quadratic
- Spline - cubit





Summary

- Interpolating function fits given data points exactly, which is not appropriate if data are noisy
- Interpolating function given by linear combination of basis functions, whose coefficients are to be determined
- Existence and uniqueness of interpolant depend on whether number of parameters to be determined matches number of data points to be fit
- Piecewise polynomial (e.g., spline) interpolation can fit large number of data points with low-degree polynomials
- Cubic spline interpolation is excellent choice when smoothness is important