Piecewise Cubic Interpolation

piecewise linear : easy / rough outcome ← "plot" in MATLAB

piecewise quad. : more work / somewhat smooth

piecewise cubic. : some more work / very smooth ←

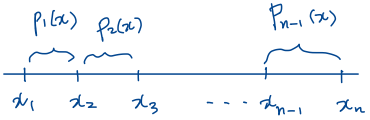
Contents

Hermite Cubic Interpolation

Problem Set-Up: General Piecewise Cubic Interpolation

We now seek a piecewise cubic polynomial which interpolates the data (x_i, y_i) for $i = 1, \ldots, n$, with $x_1 < x_2 < \cdots < x_n$, defined as

$$\mathbb{P}(x) = \begin{cases} p_1(x), & x \in [x_1, x_2) \\ p_2(x), & x \in [x_2, x_3) \\ \vdots & \vdots \\ p_{n-1}(x), & x \in [x_{n-1}, x_n] \end{cases},$$



where the ith *local* cubic polynomial p_i is written in shifted power form as

With center at x_i (the left-end point of the i^{th} interval)

$$p_i(x) = \underbrace{c_{i,1} + c_{i,2}(x - x_i) + c_{i,3}(x - x_i)^2 + c_{i,4}(x - x_i)^3}_{\text{to be determined.}}.$$

We need to determine $\boxed{4(n-1) = 4n-4}$ coefficients!

Hermite Cubic Interpolation

in addition to (x_i, y_i)

If the slopes at the breakpoints are prescribed, *i.e.*, for each $i = 1, \ldots, n-1$,

$$p_i(x_i) = y_i\,, \quad p_i'(x_i) = \sigma_i\,, \quad p_i(x_{i+1}) = y_{i+1}\,, \quad p_i'(x_{i+1}) = \sigma_{i+1}\,,$$

then we can solve for the four unknown coefficients $c_{i,j}$, $j = 1, \ldots, 4$:

$$\boxed{\begin{aligned} c_{i,1} &= y_i\,, & c_{i,3} &= \frac{3y[x_i, x_{i+1}] - 2\sigma_i - \sigma_{i+1}}{\Delta x_i}\,, \\ c_{i,2} &= \sigma_i\,, & c_{i,4} &= \frac{\sigma_i + \sigma_{i+1} - 2y[x_i, x_{i+1}]}{(\Delta x_i)^2}\,. \end{aligned}}$$

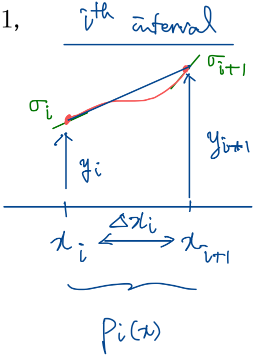
where $\Delta x_i = x_{i+1} - x_i$ and

$$y[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}.$$

(Newton's divided difference)

slope of the line connecting (x_i, y_i) and (x_{i+1}, y_{i+1})

This is called **Hermite cubic interpolation.**



$$P_i(x) = C_{i,1} + C_{i,2}(x - x_i) + C_{i,3}(x - x_i)^2 + C_{i,4}(x - x_i)^3$$

- $P_i(x_i) = \underline{C_{i,1} = y_i}$

- $P_i'(x) = C_{i,2} + 2C_{i,3}(x - x_i) + 3C_{i,4}(x - x_i)^2$

  $P_i'(x_i) = \underline{C_{i,2} = \sigma_i}$

Implementation

```
function c = hermiteCoeff(x,y,s)
% Input:
%   x,y,s    data points and slopes
% Ouput:
%   c        coefficients in matrix form
    n = length(x);
    c = zeros(n-1, 4);
    dx = diff(x);
    dy = diff(y);
    dydx = dy./dx;
    c(:,1) = y;
    c(:,2) = s;
    c(:,3) = (3*dydx - 2*s(1:n-1) - s(2:n))./dx;
    c(:,4) = (s(1:n-1) + s(2:n-1) - 2*dydx))./(dx.^2);
end
```

$$\begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} \\ C_{2,1} & C_{2,2} & C_{2,3} & C_{2,4} \\ \vdots & \vdots & \vdots & \vdots \\ C_{n-1,1} & C_{n-1,2} & C_{n-1,3} & C_{n-1,4} \end{bmatrix}$$

Convergence: Error Analysis

$\|f - \mathbb{P}\|_\infty = \max_{x\in[a,b]} |f(x) - \mathbb{P}(x)|$

Theorem 1 (Error Theorem for Hermite Cubic Interpolation)

Let $f \in C^4[a,b]$ and let $\mathbb{P}(x)$ be the Hermite cubic interpolant of

$$\left(x_i, f(x_i), f'(x_i)\right), \quad \text{for } i = 1, \ldots, n,$$

where

$$x_j = a + (j-1)h \quad \text{and} \quad h = \frac{b-a}{n-1}.$$ (uniform nodes)

Then

$$\|f - \mathbb{P}\|_\infty \; \cancel{\|f - \|_\infty} \leqslant \frac{1}{384}\|f^{(4)}\|_\infty h^4.$$ (4^{th}-order accurate)

cf) PL interpolation is 2^{nd}-order accurate.

4^{th}-order accuracy

Suppose $\|f - \mathbb{P}\|_\infty \le 10^{-4}$ with $h = 0.01$.

$\times \frac{1}{2}$

What is an upper bound on $\|f - \mathbb{P}\|_\infty$ when $h = 0.005$.

$10^{-4}/16$

Drawbacks of Hermite Cubic Interpolation

continuously differentiable

- The interpolant $p(x)$ is in C^1 and so its display may be~~e~~ too crude in graphical applications.
- In other applications, there may be difficulties if $p''(x)$ is discontinuous.
- In experimental settings where y_i are measurements of some sort, we may not have the first derivative information required for the cubic Hermite process.

Cubic Splines

Cubic Splines

In technical terms, we seek $\mathbb{P} \in C^2[a,b]$.

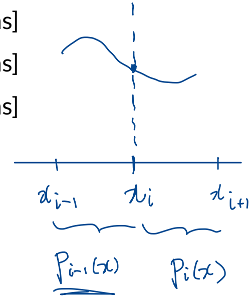
i.e., $\mathbb{P}$ is twice continuously differentiable on $[a,b]$.

Idea: Put together cubic polynomials to make the result as smooth as possible.

- At interior breakpoints: for $j = 2, 3, \cdots, n-1$

  node, $x_1, \cdots, x_n$

  - matching values: $p_{j-1}(x_j) = p_j(x_j)$ $\qquad$ $[(n-2)$ eqns$]$
  - matching first derivatives: $p'_{j-1}(x_j) = p'_j(x_j)$ $\qquad$ $[(n-2)$ eqns$]$
  - matching second derivative: $p''_{j-1}(x_j) = p''_j(x_j)$ $\qquad$ $[(n-2)$ eqns$]$



- So, together with the n interpolating conditions, we have total of $(4n-6)$ equations.
- To match up with the number of unknowns $(4n-4)$, we need to impose two more conditions on the boundary:

  1. slopes at each end (clamped cubic spline)
  2. second derivatives at the endpoints (natural cubic spline)
  3. periodic boundary condition
  4. *not-a-knot* boundary condition: $p_1(x) \equiv p_2(x)$ and $p_{n-2}(x) \equiv p_{n-1}(x)$.

MATLAB's default →

breakpoint / node

Convergence: Error Analysis

Theorem 2 (Error Theorem for Clamped Cubic Splines)

Let $f \in C^4[a, b]$ and let $p(x)$ be the cubic spline interpolant of

$$(x_i, f(x_i)), \quad \text{for } i = 1, \ldots, n,$$

with the exact boundary conditions

$$\sigma_1 = f'(x_1) \quad \text{and} \quad \sigma_n = f'(x_n),$$

in which

$$x_j = a + (j-1)h \quad \text{and} \quad h = \frac{b-a}{n-1}. \qquad \text{(uniform nodes)}$$

Then

$$\cancel{\|f - \|_\infty} \; \|f - p\|_\infty \leqslant \frac{5}{384} \|f^{(4)}\|_\infty h^4. \qquad \text{(4th-order accurate)}$$

Remarks

- Hermite cubic interpolation is about five times as accurate as cubic spline interpolation, yet both have *fourth-order accuracy*.
- Unlike the former, the latter does not require first derivatives.