# Lec 26: Piecewise Interpolation – Cubic Splines

# Hermite Cubic Interpolation

# Hermite Cubic Interpolation

We now seek a piecewise cubic polynomial $\mathbb{p}$ where $p_i(x)$ on $[x_i, x_{i+1}]$ is written in shifted power form as

$$p_i(x) = c_{i,1} + c_{i,2}(x - x_i) + c_{i,3}(x - x_i)^2 + c_{i,4}(x - x_i)^3 \,.$$

- If the slopes at endpoints are additionally given, *i.e.*,

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

  then we can solve for the four unknown coefficients $c_{i,j}$:

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

- This is called **Hermite cubic interpolation**.

# Convergence: Error Analysis

**Theorem 1 (Error Theorem for Hermite Cubic Interpolation)**

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

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

*where*

$$x_j = a + jh \quad \text{and} \quad h = \frac{b-a}{n-1}.$$

*Then*

$$\|f - p\|_\infty \leqslant \frac{1}{384} \|f^{(4)}\|_\infty h^4.$$

# Cubic Splines

# Cubic Splines

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

- At interior breakpoints: for $j = 2, 3, \cdots, n-1$
  - matching values: $p_{j-1}(x_j) = p_j(x_j)$ [$(n-2)$ eqns]
  - matching first derivatives: $p'_{j-1}(x_j) = p'_j(x_j)$ [$(n-2)$ eqns]
  - matching second derivative: $p''_{j-1}(x_j) = p''_j(x_j)$ [$(n-2)$ 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 $p_{n-2}(x) \equiv p_{n-1}(x)$.

# Convergence: Error Analysis

## Theorem 2 (Error Theorem for Clamped Cubic Splines)

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

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

*with the exact boundary conditions*

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

*in which*

$$x_j = a + jh \quad \text{and} \quad h = \frac{b-a}{n-1}.$$

*Then*

$$\|f - \wp\|_\infty \leqslant \frac{5}{384} \|f^{(4)}\|_\infty h^4.$$

# 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.

# Appendix: Derivation of Cubic Spline Algorithm

# Cubic Spline: Problem Set-Up

Given $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$, find a piecewise polynomial $\mathbb{p}(x) = p_i(x)$ on $[x_i, x_{i+1}]$ with

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

satisfying[1]

1. $\mathbb{p}(x_i) = y_i$ for $i = 1, \ldots, n$;
2. $p_i(x_{i+1}) = p_{i+1}(x_{i+1})$ for $i = 1, \ldots, n-2$;
3. $p_i'(x_{i+1}) = p_{i+1}'(x_{i+1})$ for $i = 1, \ldots, n-2$;
4. $p_i''(x_{i+1}) = p_{i+1}''(x_{i+1})$ for $i = 1, \ldots, n-2$.

[1] Let us not worry about the boundary conditions yet.

# Connection to Hermite Cubic Interpolation

**Key Observation.** If $c_{i,j}$'s are set to be

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

as in the Hermite cubic interpolation for some constants $\sigma_1, \sigma_2, \ldots, \sigma_n$ to be determined, then $\mathbb{p}(x)$ satisfies the first three requirements from the previous slide.

**Reduction.** Determine $\sigma_1, \sigma_2, \ldots, \sigma_n$ so that the fourth requirement is satisfied.

# Derivation of a Linear System for Cubic Splines (1)

Using ($\star$), write out the fourth requirement $p''_{i-1}(x_i) = p''_i(x_i)$ in terms of $\sigma_{i-1}, \sigma_i, \sigma_{i+1}$, where $i \in \mathbb{N}[2, n-1]$.

# Derivation of a Linear System for Cubic Splines (2)

Express the system of $n-2$ equations for $\sigma_1, \ldots, \sigma_n$ as a matrix equation $X\boldsymbol{\sigma} = \mathbf{r}$, where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)^{\mathrm{T}}$ and $X \in \mathbb{R}^{n\times n}$ and $\mathbf{r} \in \mathbb{R}^n$ are to be found[2].

[2] Since two equations are still missing, leave the first and last rows of $X$ and $\mathbf{r}$ empty for now.

# Tridiagonal System for Cubic Splines

**Notation.** $\Delta x_i = x_{i+1} - x_i$ and $\nabla x_i = \Delta x_{i-1} + \Delta x_i = x_{i+1} - x_{i-1}$.

$$X = \begin{bmatrix} * & * & * & \cdots & * & * & * \\ \Delta x_2 & 2\nabla x_2 & \Delta x_1 & & & & \\ & \Delta x_3 & 2\nabla x_3 & \Delta x_2 & & & \mathbf{0} \\ & & \ddots & \ddots & \ddots & & \\ \mathbf{0} & & & \Delta x_{n-2} & 2\nabla x_{n-2} & \Delta x_{n-3} & \\ & & & & \Delta x_{n-1} & 2\nabla x_{n-1} & \Delta x_{n-2} \\ * & * & * & \cdots & * & * & * \end{bmatrix},$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_{n-2} \\ \sigma_{n-1} \\ \sigma_n \end{bmatrix}, \quad \text{and} \quad \mathbf{r} = \begin{bmatrix} * \\ 3\left(y[x_1, x_2]\Delta x_2 + y[x_2, x_3]\Delta x_1\right) \\ 3\left(y[x_2, x_3]\Delta x_3 + y[x_3, x_4]\Delta x_2\right) \\ \vdots \\ 3\left(y[x_{n-3}, x_{n-2}]\Delta x_{n-2} + y[x_{n-2}, x_{n-1}]\Delta x_{n-3}\right) \\ 3\left(y[x_{n-2}, x_{n-1}]\Delta x_{n-1} + y[x_{n-1}, x_n]\Delta x_{n-2}\right) \\ * \end{bmatrix}.$$

# Implementation of Boundary Conditions (1)

- (**clamped cubic spline**) If slopes at each end are known, fill in the first and the last equation of $X\boldsymbol{\sigma} = \mathbf{r}$ with

$$\sigma_1 = y_1', \quad \sigma_n = y_n'.$$

- (**natural cubic spline**) If the second derivatives at the endpoints are known, then use

$$\begin{aligned} 2\sigma_1 + \sigma_2 &= 3y[x_1, x_2] - \frac{1}{2}\Delta x_1 y_1'' \\ \sigma_{n-1} + 2\sigma_n &= 3y[x_{n-1}, x_n] + \frac{1}{2}\Delta x_{n-1} y_n''. \end{aligned}$$

# Implementation of Boundary Conditions (2)

- (**periodic boundary condition**) If the data points come from a periodic function with period $P = x_n - x_1$ so that $\sigma_1 = \sigma_n$, then use

$$\begin{aligned} \Delta x_1 \sigma_{n-1} + 2\nabla x_1 \sigma_1 + \Delta x_{n-1}\sigma_2 &= 3\left(y[x_{n-1}, x_n]\Delta x_1 + y[x_1, x_2]\Delta x_{n-1}\right) \\ \sigma_1 - \sigma_n &= 0. \end{aligned}$$

Here, take $\nabla x_1 = x_2 - x_0 = x_2 - (x_{n-1} - P)$.

# Implementation of Boundary Conditions (3)

- (***not-a-knot* boundary condition**) If nothing is known about the endpoints, require $p_1(x) \equiv p_2(x)$ and $p_{n-2}(x) = p_{n-1}(x)$:

$$
\begin{aligned}
(\Delta x_2)^2 \sigma_1 + \left((\Delta x_2)^2 - (\Delta x_1)^2\right) \sigma_2 - (\Delta x_1)^2 \sigma_3 \\
= 2 \left(y[x_1, x_2] (\Delta x_2)^2 - y[x_2, x_3] (\Delta x_1)^2\right),
\end{aligned}
$$

$$
\begin{aligned}
(\Delta x_{n-1})^2 \sigma_{n-2} + \left((\Delta x_{n-2})^2 - (\Delta x_{n-1})^2\right) \sigma_{n-1} + (\Delta x_{n-2})^2 \sigma_n \\
= 2 \left(y[x_{n-1}, x_n] (\Delta x_{n-2})^2 - y[x_{n-2}, x_{n-1}] (\Delta x_{n-1})^2\right).
\end{aligned}
$$

**Exercise.** Derive the equations shown above.