SPLINE INTERPOLATION

Spline Background

- Problem: high degree interpolating polynomials often have extra oscillations.
  Example: “Runge” function \( f(x) = \frac{1}{1+4x^2}, x \in [-1, 1] \).

- Piecewise Polynomials provide alternative to high degree polynomials: approximation interval \([a, b]\) is subdivided into pieces \([x_1, x_2]\), \([x_2, x_3]\), \ldots, \([x_{n-1}, x_n]\), with \(a = x_1 < x_2 < \cdots < x_n = b\), and a low degree polynomial is used to approximate \( f(x) \) on each subinterval.
  Example: piecewise linear approximation \( S(x) \)

  \[
  S(x) = f(x_j) + (x - x_j) \frac{f(x_{j+1}) - f(x_j)}{x_{j+1} - x_j}, \quad \text{if} \; x \in [x_j, x_{j+1}]
  \]

- Splines are piecewise polynomial approximations, connected at \( x_j \)'s with various continuity conditions.
CUBIC SPLINE INTERPOLATION

Cubic Interpolating Splines for \( a = x_1 < \cdots < x_n = b \)
with given data \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\).

- Properties of Cubic Interpolating Spline \( S(x) \),
  a) \( S(x) \) is composed of cubic polynomial pieces \( S_j(x) \)
    
    \[ S(x) = S_j(x) \text{ if } x \in [x_j, x_{j+1}], \quad j = 1, 2, \ldots, n-1. \]

  b) \( S(x_j) = y_j, \quad j = 1, \ldots, n. \) (interpolation)
  c) \( S_{j-1}(x_j) = S_j(x_j), \quad j = 2, \ldots, n-1 \) (\( S \in C[a, b] \)).
  d) \( S'_j(x_j) = S'_j(x_j), \quad j = 2, \ldots, n-1 \) (\( S \in C^1[a, b] \)).
  e) \( S''_j(x_j) = S''_j(x_j), \quad j = 2, \ldots, n-1 \) (\( S \in C^2[a, b] \)).
  f) two end conditions: examples
    i) \( S''(x_1) = S''(x_n) = 0 \) (natural or free spline);
    ii) \( S'(x_1) = f'(x_1), \quad S'(x_n) = f'(x_n) \)
        (complete or clamped spline);
    iii) \( S'''_1 = S'''_{n-1} = 0 \) (parabolically terminated);
    iv) \( S'''_1(x_2) = S'''_2(x_2), \quad S'''_{n-2}(x_{n-1}) = S'''_{n-1}(x_{n-1}) \)
        (not-a-knot).

Note: if

\[ S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3, \]

condition a) provides \( 4(n - 1) \) free parameters;

b)-f) give \( n + 3(n - 2) + 2 = 4(n - 1) \) constraints.
CUBIC SPLINE INTERPOLATION

- Example: $n = 3$, natural, data (1,2), (2,3), (3,5).

\[
S_1(x) = 2 + \frac{3}{4}(x - 1) + \frac{1}{4}(x - 1)^3, \\
S_2(x) = 3 + \frac{3}{2}(x - 2) + \frac{3}{4}(x - 2)^2 - \frac{1}{4}(x - 2)^3,
\]
CUBIC SPLINE INTERPOLATION

Cubic Interpolating Spline Construction

- Spline Linear System: let \( h_j = x_{j+1} - x_j \); start with

\[
S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3
= y_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3.
\]

c) \( S_{j+1}(x_{j+1}) = S_j(x_{j+1}) \), implies

\[
y_{j+1} = y_j + b_jh_j + c_jh_j^2 + d_jh_j^3, \quad \frac{\Delta y_j}{h_j} = b_j + c_jh_j + d_jh_j^2,
\]

where \( \Delta y_j = y_{j+1} - y_j \).
Notice \( S''_j(x) = 2c_j + 6d_j(x - x_j) \), so

e) \( S''_{j+1}(x_{j+1}) = S''_j(x_{j+1}) \), implies

\[
2c_{j+1} = 2c_j + 6d_jh_j; \quad d_jh_j = (c_{j+1} - c_j)/3,
\]

with extra unknown \( c_n = S''_{n-1}(x_n)/2 \) added. Then

\[
\frac{\Delta y_j}{h_j} = b_j + c_jh_j + \frac{(c_{j+1} - c_j)h_j}{3} = b_j + \frac{(c_{j+1} + 2c_j)h_j}{3};
\]

\[
\frac{3\Delta y_{j+1}}{h_{j+1}} - \frac{3\Delta y_j}{h_j} = 3\Delta b_j + (c_{j+2} + 2c_{j+1})h_{j+1} - (c_{j+1} + 2c_j)h_j.
\]

Also \( S'_j(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2 \), so
d) \( S'_{j+1}(x_{j+1}) = S'_j(x_{j+1}); \quad b_{j+1} = b_j + 2c_jh_j + 3d_jh_j^2; \)

\[
\Delta b_j = 2c_jh_j + (c_{j+1} - c_j)h_j = (c_{j+1} + c_j)h_j
\]

\[
\frac{3\Delta y_{j+1}}{h_{j+1}} - \frac{3\Delta y_j}{h_j} = c_jh_j + 2c_{j+1}(h_j + h_{j+1}) + c_{j+2}h_{j+1}
\]
CUBIC SPLINE INTERPOLATION

• Natural Splines: $S''(x_1) = S''(x_n) = 0$, so $c_1 = c_n = 0$

Linear system equations are a “tridiagonal” system

\[
\begin{align*}
    c_1 &= 0 \\
    c_1 h_1 + 2c_2 (h_1 + h_2) + c_3 h_2 &= \frac{3\Delta y_2}{h_2} - \frac{3\Delta y_1}{h_1} \\
    c_2 h_2 + 2c_3 (h_2 + h_3) + c_4 h_3 &= \frac{3\Delta y_3}{h_3} - \frac{3\Delta y_2}{h_2} \\
    \vdots \\
    c_{n-3} h_{n-3} + 2c_{n-2} (h_{n-3} + h_{n-2}) + c_{n-1} h_{n-2} &= \frac{3\Delta y_{n-2}}{h_{n-2}} - \frac{3\Delta y_{n-3}}{h_{n-3}} \\
    c_{n-2} h_{n-2} + 2c_{n-1} (h_{n-2} + h_{n-1}) + c_n h_{n-1} &= \frac{3\Delta y_{n-1}}{h_{n-1}} - \frac{3\Delta y_{n-2}}{h_{n-2}} \\
    c_n &= 0.
\end{align*}
\]

which can be solved uniquely for $c_j$’s with $O(n)$ work;

$d_j = (c_{j+1} - c_j)/(3h_j)$, $b_j = \Delta y_j/h_j - c_j h_j - d_j h_j^2$

can be used to find remaining coefficients for $S_j(x)$’s.

Note: if all $h_j = h$, a simpler tridiagonal system.
CUBIC SPLINE INTERPOLATION

- Example: \( n = 3 \), natural, with data \((1,2), (2,3), (3,5)\), so \( h_1 = h_2 = 1, \Delta y_1 = 1, \Delta y_2 = 2 \).

Only one equation \( 2c_2(1 + 1) = 3(2 - 1) \), so \( c_2 = 3/4 \), \( d_1 = 1/4 \), \( d_2 = -1/4 \);
\( b_1 = 1 - 0 - 1/4 = 3/4 \), \( b_2 = 2 - 3/4 + 1/4 = 3/2 \).

\[
S_1(x) = 2 + \frac{3}{4}(x - 1) + \frac{1}{4}(x - 1)^3,
\]
\[
S_2(x) = 3 + \frac{3}{2}(x - 2) + \frac{3}{4}(x - 2)^2 - \frac{1}{4}(x - 2)^3.
\]

- Example: “Runge” function \( f(x) = \frac{1}{1+4x^2}, \ x \in [-1, 1] \).
CUBIC SPLINE INTERPOLATION

• Clamped Splines: let $S'(x_1) = y'_1$, $S''(x_n) = y'_n$, so $y'_1 = b_1$, $y'_n = b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2$

Using $\frac{\Delta y_j}{h_j} = b_j + \frac{(c_{j+1}+2c_j)h_j}{3}$, $3h_jd_j = (c_{j+1} - c_j)$, “1st” and “n-th” equations become

$2c_1h_1 + c_2h_1 = \frac{3\Delta y_1}{h_1} - 3y'_1$, and

$c_{n-1}h_{n-1} + 2c_nh_{n-1} = 3y'_n - \frac{3\Delta y_{n-1}}{h_{n-1}}$.

Linear system equations are a “tridiagonal” system

\[
\begin{align*}
2c_1h_1 + c_2h_1 &= \frac{3\Delta y_1}{h_1} - 3y'_1 \\
c_1h_1 + 2c_2(h_1+h_2) + c_3h_2 &= \frac{3\Delta y_2}{h_2} - \frac{3\Delta y_1}{h_1} \\
&\vdots \\
c_{n-2}h_{n-2} + 2c_{n-1}(h_{n-2}+h_{n-1}) + c_nh_{n-1} &= \frac{3\Delta y_{n-1}}{h_{n-1}} - \frac{3\Delta y_{n-2}}{h_{n-2}} \\
c_{n-1}h_{n-1} + 2c_nh_{n-1} &= 3y'_n - \frac{3\Delta y_{n-1}}{h_{n-1}}.
\end{align*}
\]

which can be solved (uniquely) for $c_j$’s with $O(n)$ work;

$d_j = (c_{j+1} - c_j)/(3h_j)$, $b_j = \Delta y_j/h_j - c_jh_j - d_jh_j^2$

can be used to find remaining coefficients for $S_j(x)$’s.
• Example: \( n = 3 \), clamped, with data \((1,2), (2,3), (3,5)\), and \( y'_1 = 1, y'_3 = 2 \). Three equations:

\[
2c_1 + c_2 = 3(2 - 1) - 3 = 0, \\
c_1 + 4c_2 + c_3 = 3, \\
c_2 + 2c_3 = 6 - 3(2) = 0,
\]

so \( c_1 = c_3 = -1/2, c_2 = 1; \)

\( d_1 = 1/2, d_2 = -1/2; b_1 = 1, b_2 = 2-1+1/2 = 3/2. \)

\[
S_1(x) = 2 + (x - 1) - \frac{1}{2}(x - 1)^2 + \frac{1}{2}(x - 1)^3, \\
S_2(x) = 3 + \frac{3}{2}(x - 2) + (x - 2)^2 - \frac{1}{2}(x - 2)^3,
\]

• Parabolically Terminated Splines: \( S''_1 = S''_{n-1} = 0, \)

so \( d_1 = d_{n-1} = 0, c_1 = c_2, c_{n-1} = c_n. \)

Linear system equations are a “tridiagonal” system

\[
c_1 - c_2 = 0 \\
c_1h_1 + 2c_2(h_1+h_2) + c_3h_2 = \frac{3\Delta y_2}{h_2} - \frac{3\Delta y_1}{h_1} \\
\vdots \\
c_{n-2}h_{n-2} + 2c_{n-1}(h_{n-2}+h_{n-1}) + c_nh_{n-1} = \frac{3\Delta y_{n-1}}{h_{n-1}} - \frac{3\Delta y_{n-2}}{h_{n-2}} \\
c_{n-1} - c_n = 0.
\]

which can be solved (uniquely) for \( c_j \)'s with \( O(n) \) work;

\[
d_j = (c_{j+1} - c_j)/(3h_j), \ b_j = \Delta y_j/h_j - c_jh_j - d_jh_j^2
\]

can be used to find remaining coefficients for \( S_j(x) \)'s.
CUBIC SPLINE INTERPOLATION

• Not-a-Knot Splines:

\[ S_1'''(x_2) = S_2'''(x_2), \quad S_{n-2}'''(x_{n-1}) = S_{n-1}'''(x_{n-1}), \text{ so} \]
\[ d_1 = d_2, \quad d_{n-2} = d_{n-1}, \text{ and } S_1 = S_2, \quad S_{n-2} = S_{n-1}. \]

Then \( (c_2 - c_1)/h_1 = (c_3 - c_2)/h_2, \)
\( (c_{n-1} - c_{n-2})/h_{n-2} = (c_n - c_{n-1})/h_{n-1}, \text{ so} \)
“1st” and “nth” equations become
\[ c_1 h_2 - c_2 (h_1 + h_2) + c_3 h_1 = 0 \]
\[ c_{n-2} h_{n-1} - c_{n-1} (h_{n-2} + h_{n-1}) + c_n h_{n-2} = 0. \]

Linear system equations are a “tridiagonal” system
\[
\begin{align*}
  c_1 h_2 - c_2 (h_1 + h_2) + c_3 h_1 &= 0 \\
  c_1 h_1 + 2c_2 (h_1 + h_2) + c_3 h_2 &= \frac{3\Delta y_2}{h_2} - \frac{3\Delta y_1}{h_1} \\
  \vdots & \vdots \\
  c_{n-2} h_{n-2} + 2c_{n-1} (h_{n-2} + h_{n-1}) + c_n h_{n-1} &= \frac{3\Delta y_{n-1}}{h_{n-1}} - \frac{3\Delta y_{n-2}}{h_{n-2}} \\
  c_{n-2} h_{n-1} - c_{n-1} (h_{n-2} + h_{n-1}) + c_n h_{n-2} &= 0.
\end{align*}
\]

which can be solved (uniquely) for \( c_j \)'s with \( O(n) \) work;
\[ d_j = (c_{j+1} - c_j)/(3h_j), \quad b_j = \Delta y_j/h_j - c_j h_j - d_j h_j^2 \]
can be used to find remaining coefficients for \( S_j(x) \)'s.
CUBIC SPLINE INTERPOLATION

Efficient Spline Evaluation

- Setup: solve linear system for $c_j$’s in $O(n)$ time
- For each evaluation point $x$, find interval $x \in [x_j, x_{j+1}]$ in $O(\log(n))$ time and evaluate $S_j(x)$ in $O(1)$ time.

Alternate formula for $S_j(x)$ (without $b_j$’s and $d_j$’s):

$$S_j(x) = \frac{c_j}{3h_j}(x_{j+1} - x)^3 + \frac{c_{j+1}}{3h_{j+1}}(x - x_j)^3 + \frac{y_j - c_jh_j}{3}(x_{j+1} - x) + \frac{y_{j+1} - c_{j+1}h_{j+1}}{3}(x - x_j).$$

- Compare with polynomial interpolation, where setup time is $O(n^2)$ and evaluation time is $O(n)$.

Error Theorem:

if $f \in C^4[a, b]$, with $\max_{x \in [a, b]} |f^{(4)}(x)| = M$, and $S(x)$ is the unique clamped spline for $f(x)$ with nodes $a = x_1 < x_2 \cdots < x_n = b$, then

$$|f(x) - S(x)| \leq \frac{5M}{384} \max_{1 \leq j < n} h_j^4.$$