SFU
muraki

WEDNESDAY

- tutorials start @ 2^{30}
- instructor office hours: WED 3^{30}–5pm, THURS 11–12
  (attn: grade concerns on CA)

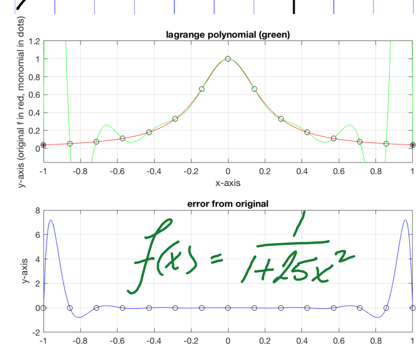
LAST DAY

- MODIFIED LAGRANGE INTERPOLANT (MLI)

$$P_{MLI}(x) = \underbrace{\left(\prod_{0}^{N} (x - x_k) \right)}_{L(x)} \left\{ \sum_{0}^{N} f_k \cdot \frac{w_k}{(x - x_k)} \right\}$$

$$w_k = \left(\prod_{j \neq k} (x_k - x_j) \right)^{-1}$$

- accuracy: error tends to $\searrow$ as $N \nearrow$
  - until finite-precision issues
  - RUNGE PHENOMENON: there are BAD functions where error $\nearrow$!



- efficiency: operation counts
  - pre-computing w_k, O( )
  - flops per $P_{MLI}(x)$, O( )
  - add extra x_{N+1}, O( )

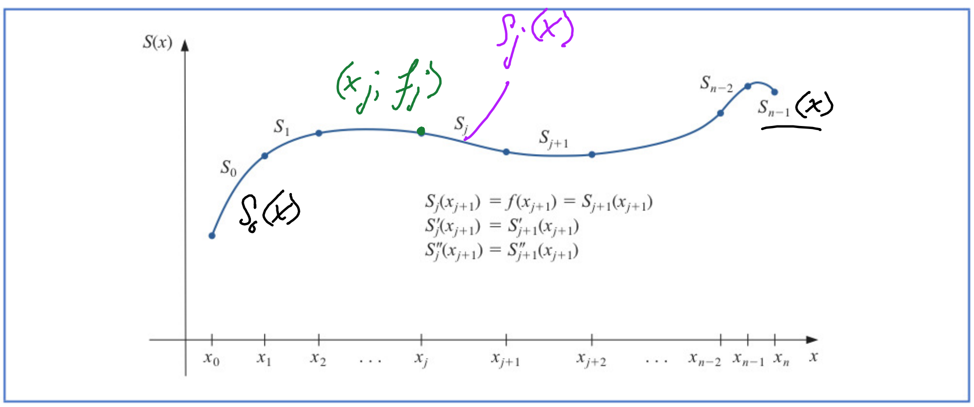
SFU muraki

- robustness: YES, for LMI & barycentric formula

NEWTON INTERPOLATION (§3.3, not a 316 method)

- key facts: linear solve for coefficients & divided-difference calculations

CUBIC SPLINE (§3.5)



Definition 3.10 Given a function f defined on $[a,b]$ and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a **cubic spline interpolant** S for f is a function that satisfies the following conditions:

A natural spline has no conditions imposed for the direction at its endpoints, so the curve takes the shape of a straight line after it passes through the interpolation points nearest its endpoints. The name derives from the fact that this is the natural shape a flexible strip assumes if forced to pass through specified interpolation points with no additional constraints. (See Figure 3.9.)

(a) $S(x)$ is a cubic polynomial, denoted $S_j(x)$, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0, 1, \ldots, n-1$;

(b) $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0, 1, \ldots, n-1$;

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* **(b)**.)

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(f) One of the following sets of boundary conditions is satisfied:

(i) $S''(x_0) = S''(x_n) = 0$ (**natural** *(or free)* **boundary**);

(ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (**clamped boundary**). ■

$$S_j(x) = f_j + b_j(x-x_j) + c_j(x-x_j)^2 + d_j(x-x_j)^3$$

$j = 0 \to N-1$

SFU muraki

a) $S_j(x_{j+1}) = f_j + b_j \cdot h_j + c_j h_j^2 + d_j h_j^3 = f_{j+1}$

b) $S_j'(x_{j+1}) = b_j + 2c_j h_j + 3d_j h_j^2 = S_{j+1}'(x_{j+1})$
$= b_{j+1}$

c) $S_j''(x_{j+1}) = 2c_j + 6d_j h_j = S_{j+1}''(x_{j+1})$
$= 2c_{j+1}$

• eliminate d_j using $d_j h_j = \frac{1}{3}(c_{j+1} - c_j)$ (eqn 3.17)

b') $b_{j+1} - b_j = (c_{j+1} + c_j) h_j$ (eqn. 3.19)

$\rightarrow$ $\cdot - \quad = (\ \cdot + \ \cdot\ ) h_{j-1}$

a') $f_{j+1} = f_j + b_j h_j + \frac{1}{3}(2c_j + c_{j+1}) h_j^2$ (eqn 3.18)
$f_j = f_{j-1} + b_{j-1} h_{j-1} + \frac{1}{3}(2c_{j-1} + c_j) h_{j-1}^2$

MAIN CUBIC SPLINE EQUATION

+ unknowns

c') $$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}), \qquad (3.21)$$

BUT only N- equations $j = 1 \rightarrow N-$

SFU
muraki

· need 2 more equations + then

in unknowns $\{c_0 \dots c_{N-1}, \}$

$$[A]\vec{x} = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & & & \vdots \\ 0 & h_1 & 2(h_1+h_2) & h_2 & & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix}, \begin{pmatrix} c_0 \\ \vdots \\ , \\ c_N \end{pmatrix} = \vec{b}$$

p146

and **b** and **x** are the vectors

$$\vec{\mathbf{b}} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(f_2 - f_1) - \frac{3}{h_0}(f_1 - f_0) \\ \vdots \\ \frac{3}{h_{n-1}}(f_n - f_{n-1}) - \frac{3}{h_{n-2}}(f_{n-1} - f_{n-2}) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

· THE EQUATIONS → CONDITIONS (of spline)

· 3 variations (there is NO " " answer)

a) ENDS: (p146, above) $f''(x_0) = \quad = \quad , \quad f'(x_N) = \quad =$

b) ENDS: (p150 + matlab) $b_0 =$

$S'_{N-1}(x_N) =$

need deriv values at ends!

c) ENDS: (matlab) $S_0'''(x_1) = S_1'''(x_1) \rightarrow \quad =$

$S_{N-2}'''(x_{N-1}) = S_{N-1}'''(x_{N-1}) \quad =$

!

↳ my preferred logic

$S_1(x)$ holds in $< x <$ & $S_1(x_0) =$ $. ,$ etc.

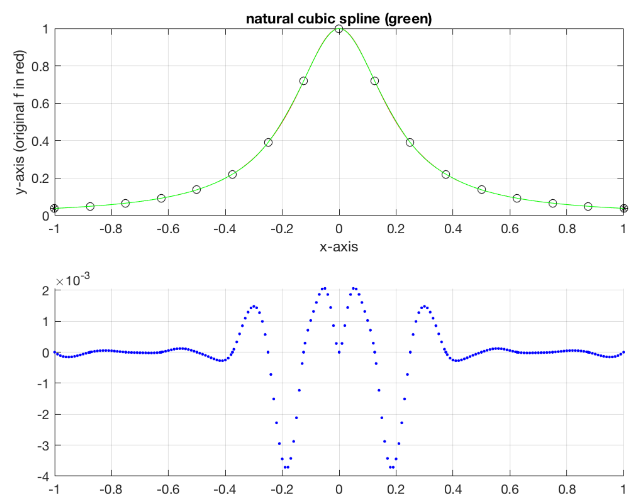
as in is NOT-a-KNOT!

CUBIC SPLINE → N×N MATRIX SOLVE

- tri-diagonal (O flops), ' −

(,)

- natural spline demo



- your observations:

SFU

muraki

Example 1 Construct a natural cubic spline that passes through the points $(1,2)$, $(2,3)$, and $(3,5)$.

Solution This spline consists of two cubics. The first for the interval $[1,2]$, denoted

$$S_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3,$$

and the other for $[2,3]$, denoted

$$S_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3.$$

- isolate the necessary equations

$S_0(\ \ ) = \quad + \quad + \quad + \quad =$

$S_1(\ \ ) = \quad + \quad + \quad + \quad =$

$S_0'(\ \ ) = S_1'(\ \ )$

$\quad + \quad + \quad =$

$S_0''(\ \ ) = S_1''(\ \ )$

$\quad + \quad =$

NATURAL $S_0''(\ \ ) =$

$S_1''(\ \ ) =$

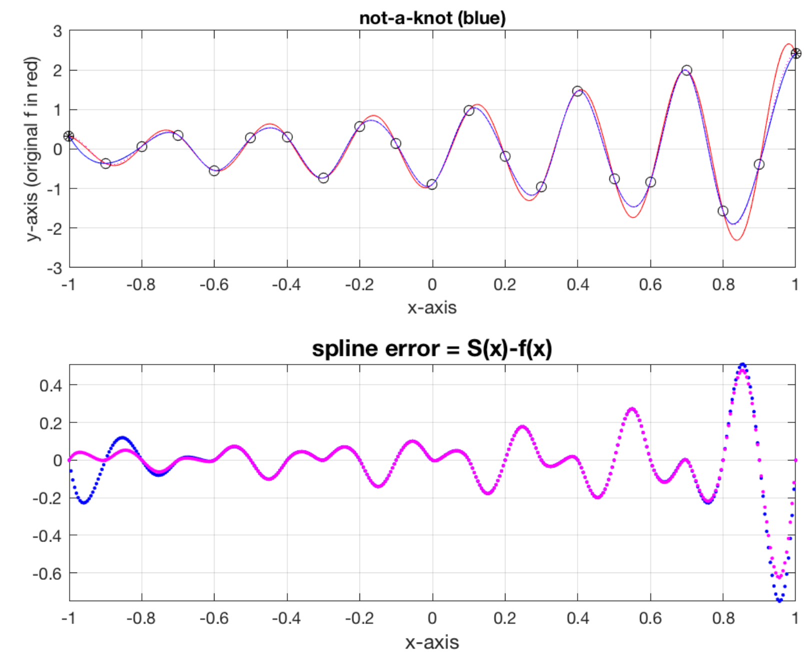
Solving this system of equations gives the spline

$$S(x) = \begin{cases} 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3, & \text{for } x \in [1,2] \\ 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3, & \text{for } x \in [2,3] \end{cases}$$

SFU

muraki

• not-a-knot + clamped demo

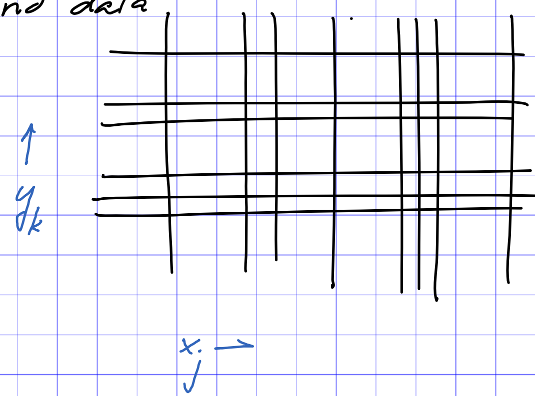


• clamped is using ZERO DERIVATIVE at ends (wrong value!)

↳ for data, we might not have access to derivative information.

MATLAB 2D INTERPOLATION

- interp2.m for lattice grid data

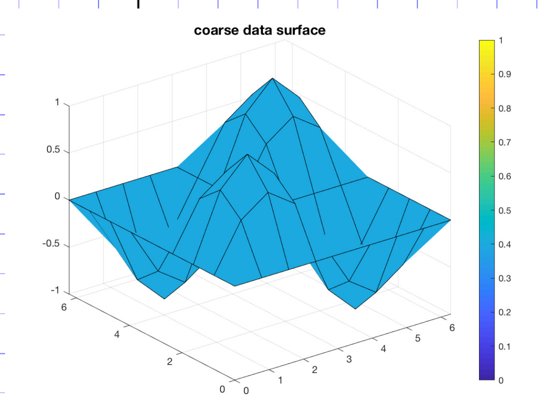


- polynomials over each rectangle

$$\text{bilinear} = p(x,y) = (ax+b)(cy+d)$$
$$= Axy + Bx + Cy + D$$

4-corners formula $= p(x,y)$

$$= \left(\tfrac{y}{\Delta y},\ 1-\tfrac{y}{\Delta y} \right) \begin{bmatrix} f(0,\Delta y) & f(\Delta x,\Delta y) \\ f(0,0) & f(\Delta x,0) \end{bmatrix} \begin{pmatrix} 1-\tfrac{x}{\Delta x} \\ \tfrac{x}{\Delta x} \end{pmatrix}$$



SFU
muraki