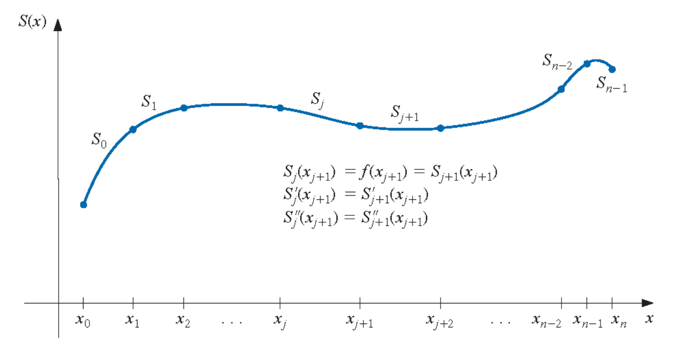
SFU

muraki

FRIDAY

- QUIZ TODAY, bring calculator
- COMPUTING WORKSHOP @ 2^{30}
- UPCOMING DATES : MON 11 FEB CA4 due
  FRI 15 FEB quiz 6
  - CANVAS lecture table has all CA + quiz dates



LAST DAY

- cubic spline (§3.5)
- NUMBERS
- $N+1$ data points, intervals, $-$ interior nodes
- cubic interpolants, coefficients
  - equation count: , $S_j(x_j) = f_j = a_j$

$(j = 0 \to N-1)$, $S_j(x_{j+1}) = f_{j+1}$

$-$, $S_j'(x_{j+1}) = S'(x_{j+1})$

$-$, $S_j''(x_{j+1}) = S''(x_{j+1})$

, endpoints (natural, clamped, not-a-knot)

$=$ equations

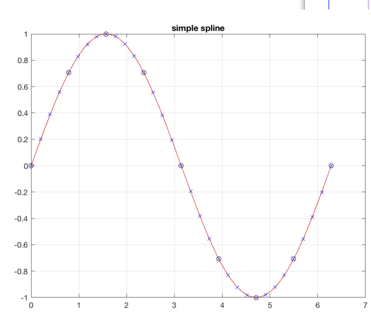
SFU

muraki

MATLAB: spline + ppval

```
%
% simple_spline.m -- djm -- 07 feb 2019
%

%
% exact curve, plotting points
%
xx = 0:pi/50:2*pi;

figure(1); clf
plot(xx,sin(xx),'r'); hold on
grid on

title('simple spline')
xlabel('x-axis')

%
% N sample points
%
N = 8;
xs = (0:N)*(2*pi/N);
ys = sin(xs);

plot(xs,ys,'ko')

%
% interpolation points
%
xi = (0:1/4:N)*(2*pi/N);
yi = spline(xs,ys,xi);

plot(xi,yi,'k:')

%
% spline coefficients
%

sp_info = spline(xs,ys)
whos

out1 = sp_info.breaks
out2 = sp_info.coefs
out3 = sp_info.pieces

%
% spline from coefficients
%
yci = ppval(sp_info,xi);
plot(xi,yci,'bx')
```

```
>> simple_spline

sp_info =

  struct with fields:

      form: 'pp'
    breaks: [0 0.7854 1.5708 2.3562 3.1416 3.9270 4.7124 5.4978 6.2832]
     coefs: [8Ã—4 double]
    pieces: 8
     order: 4
       dim: 1

  Name        Size            Bytes  Class     Attributes

  N           1x1                 8  double
  ans         1x9                72  double
  breaks      1x9                72  double
  out1        1x9                72  double
  out2        8x4               256  double
  out3        1x1                 8  double
  sp_info     1x1              1412  struct
  xi          1x33              264  double
  xs          1x9                72  double
  xx          1x101             808  double
  yci         1x33              264  double
  yi          1x33              264  double
  ys          1x9                72  double


out1 =

         0    0.7854    1.5708    2.3562    3.1416    3.9270    4.7124    5.4978    6.2832


out2 =

   -0.0849   -0.1356    1.0592         0
   -0.0849   -0.3357    0.6890    0.7071
    0.0705   -0.5359    0.0044    1.0000
    0.1569   -0.3697   -0.7068    0.7071
    0.1569   -0.0000   -0.9971    0.0000
    0.0705    0.3697   -0.7068   -0.7071
   -0.0849    0.5359    0.0044   -1.0000
   -0.0849    0.3357    0.6890   -0.7071


out3 =

     8
```



SFU

muraki

• CUBIC SPLINE SUMMARY

•   - error from a

Theorem 3.13 Let $f \in C^4[a,b]$ with $\max_{a\le x\le b}|f^{(4)}(x)| = M$. If S is the unique clamped cubic spline interpolant to f with respect to the nodes $a = x_0 < x_1 < \cdots < x_n = b$, then for all x in $[a,b]$,

$$|f(x) - S(x)| \le \frac{5M}{384}\max_{0\le j\le n-1}(x_{j+1}-x_j)^4.$$

error.

of from

a on $\frac{d^4f}{dx^4}$. depending (deg +)

same form as

+ errors (polynomial approx)

•

d

• because cubic spline is equivalent to

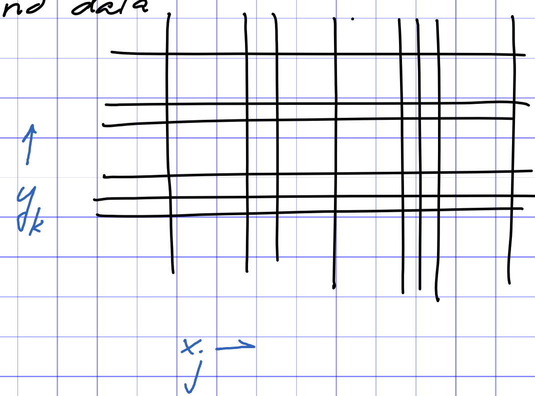
linear solve

O( )

!

MATLAB 2D INTERPOLATION

- interp2.m for lattice grid data

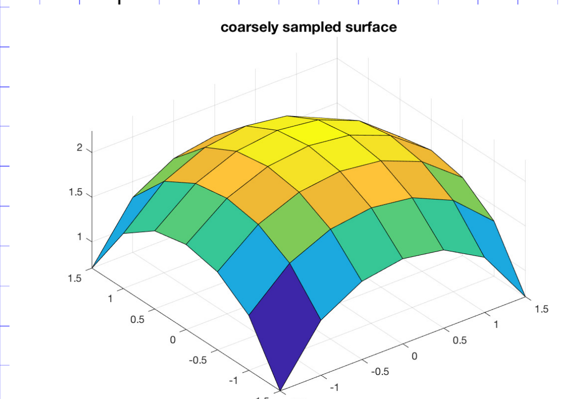


- polynomials over each rectangle

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

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

POLYNOMIAL INTERPOLATION

- NLI versus cubic spline
- NLI: simple, ONE FUNCTION to evaluate

  all derivatives of $P_{NLI}(x)$

  there are BAD $\{x_k\}$ & $f(x)$

  for large errors

  generally fails for N large

  $O(N^2)$ pre-computing for $\{w_k\}$

- cubic spline:

  MANY cubic poly, must find $x_k < x < x_{k+1}$

  only continuous S, S' & S''

  robust for large N

  $O(N)$ algorithm

SFU
muraki

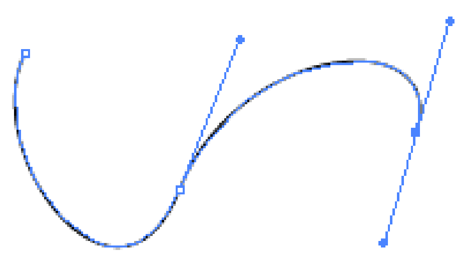
WHY DO WE STILL NEED CHOICE ?

- algorithms are seldom used in ISOLATION
  - multiple algorithm in sequence
  - for instance, derivative discontinuities in a differential equation code may be problematic ...

OTHER CHOICES

- Newton
- Hermite (what if you know $f(x_k)$ & $f'(x_k)$?)

→ BEZIER CURVES



WHAT NEXT ?

- say you have too many points, OR the data is imperfect ?

muraki

LEAST-SQUARES APPROXIMATION

· using polynomial basis functions...

SFU

muraki