March 28, 2014

Piecewise polynomials

In this class:

- *Quiz*
- *Barycentric form*
- *The error polynomial*
- *ApproxFun*
- *Piecewise linear approximations*
- *Piecewise quadratics?*
- *Piecewise cubics & splines – a surprising linear system!*

Next class

High dimensional polynomials
Numerical differentiation

Next next class

More numerical differentiation

Barycentric form of interpolating polys

The Lagrange form is:

Given $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$

$$p(x) = \sum_{i=1}^{n} y_i \psi_i(x) \qquad \psi_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}$$

This takes $O(n^2)$ work to evaluate at a point x

Barycentric form of interpolating polys

The Lagrange form is:

Given $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$

$$p(x) = \sum_{i=1}^{n} y_i \psi_i(x) \qquad \psi_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}$$

This takes $O(n^2)$ work to evaluate at a point x

Let $\psi(x) = \prod_{j=1}^{n}(x - x_j)$

Then $\psi_i(x) = \frac{\psi(x)}{(x - x_i)} \frac{1}{\prod_{j \neq i}(x_i - x_j)}$

Barycentric form of interpolating polys

The Lagrange form is:

Given $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$

$$p(x) = \sum_{i=1}^{n} y_i \psi_i(x) \qquad \psi_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}$$

This takes $O(n^2)$ work to evaluate at a point x

Let $\psi(x) = \prod_{j=1}^{n}(x - x_j)$ **Let** $w_i = \frac{1}{\prod_{j \neq i}(x_i - x_j)}$

Then $p(x) = \sum_{i=1}^{n} y_i \frac{\psi(x)}{x - x_i} w_i$

Takes $O(n^2)$ work to compute w_i
but $O(n)$ work to evaluate $p(x)$

The Barycentric Form

$$p(x) = \frac{\displaystyle\sum_{i=1}^{n} \frac{w_i}{(x - x_i)} y_i}{\displaystyle\sum_{i=1}^{n} \frac{w_i}{x - x_i}}$$

Takes $O(n^2)$ work to compute w_i
but $O(n)$ work to evaluate $p(x)$
better numerical properties

To derive, use $1 = \phi(x) \displaystyle\sum_{i=1}^{n} \frac{w_i}{x - x_i}$

Let $\psi(x) = \displaystyle\prod_{j=1}^{n} (x - x_j)$

Let $w_i = \dfrac{1}{\prod_{j \neq i} (x_i - x_j)}$

Then $p(x) = \displaystyle\sum_{i=1}^{n} y_i \frac{\psi(x)}{x - x_i} w_i$

Takes $O(n^2)$ work to compute w_i
but $O(n)$ work to evaluate $p(x)$

Error in polynomial interpolation

THEOREM 8.4.1

Assume

that f is $n+1$ times cont. diff. in a region $[a, b]$, and that $x_0, \ldots, x_n$ are distinct points in $[a, b]$.

Let

$p(x)$ be the unique polynomial of degree n that interpolates f at $x_0, \ldots, x_n$.

Then

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{j=0}^{n} (x - x_i)$$

for some point ξ_x in $[a, b]$ that depends on x.

Analysis

Goes down with n

Hard to control if the n+1 derivative isn't well behaved.

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{j=0}^{n} (x - x_i)$$

A polynomial that is zero at each x_i if this is necessarily large, we'll study this!

Interpolation at Chebyshev points

THEOREM 8.5.1

Let

f be a continuous function on $[-1, 1]$

p_n its degree n interpolant at Chebyshev points

p_n^* its best approximation among n degree polynomials in the uniform error

Then

uniform error in $p_n \leq (2 + \frac{2}{\pi} \log n)$uniform error in p_n^*

p_n converges exponentially fast to f if f is smooth

Analysis

If we interpolate f at Chebyshev points, we get something close to the *best possible result*

Example. Suppose f is complicated.

and best (unknowable) degree 100 poly p_{100}^* gets $\max_x |p_n^*(x) - f(x)| \leq 10^{-5}$

computable Chebyshev degree 100 poly p_{100} gets $\max_x |p_n(x) - f(x)| \leq 5 \cdot 10^{-5}$ (2 +2/pi log(100)) <= 5

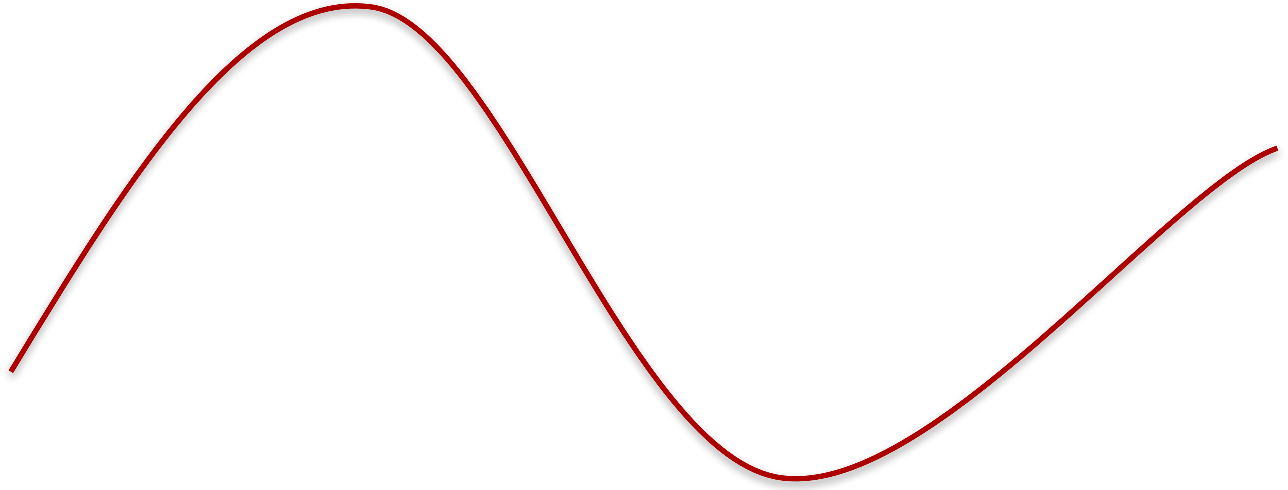
and if *f* doesn't have discontinuous derivatives (e.g. sin, cos, exp, …), best approx gets small "very fast"

ApproxFun demo

Piecewise polynomial approximation

Used all over!

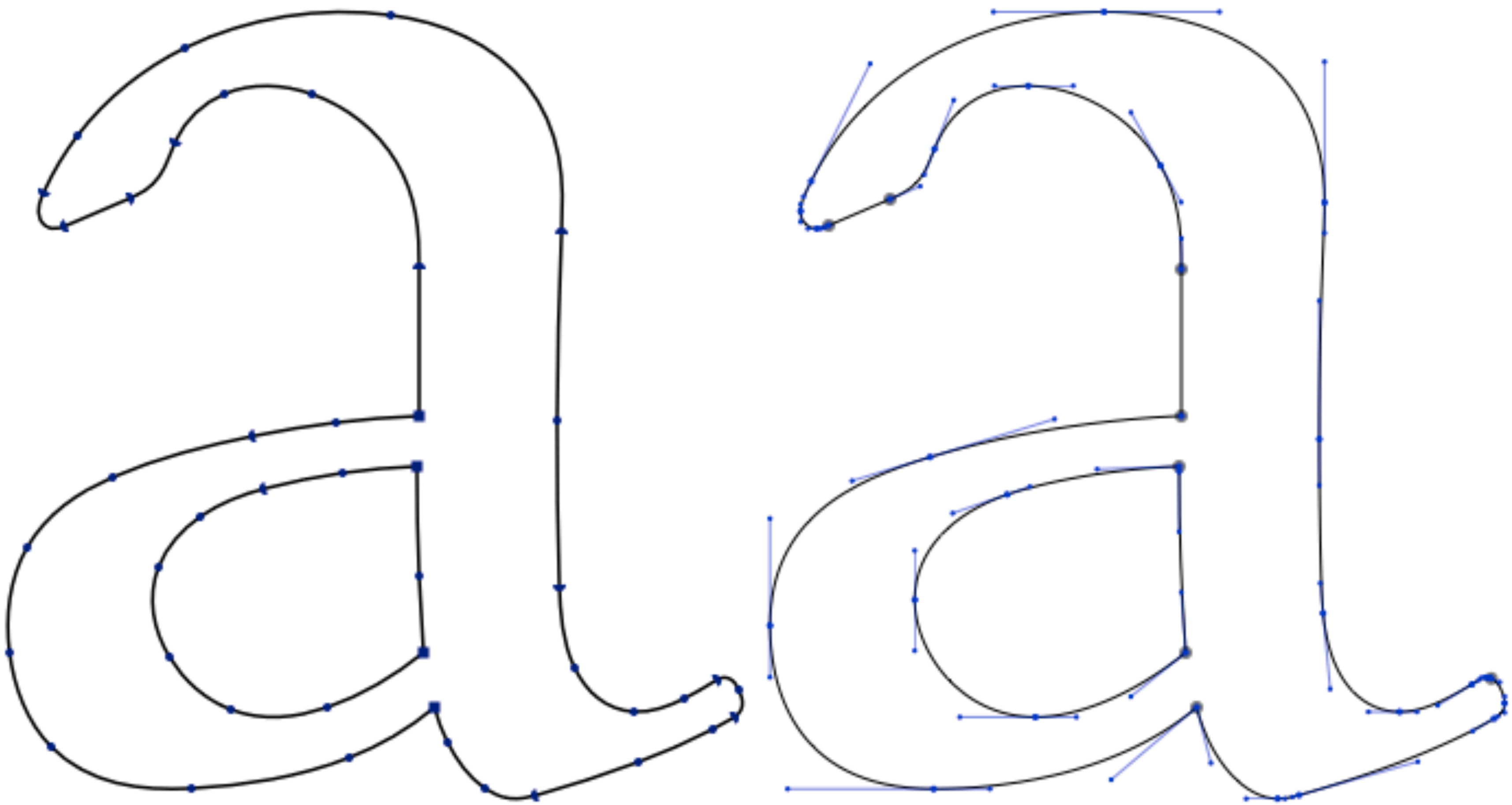
Powerpoint curves

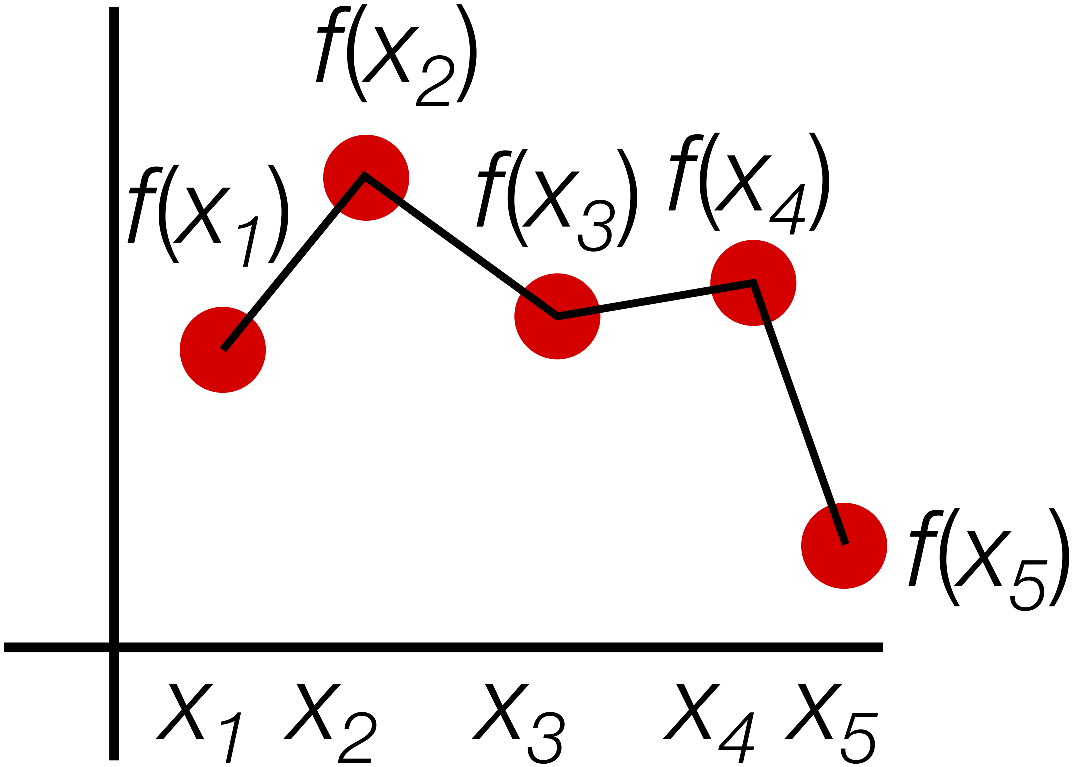


Piecewise polynomial approximation

a

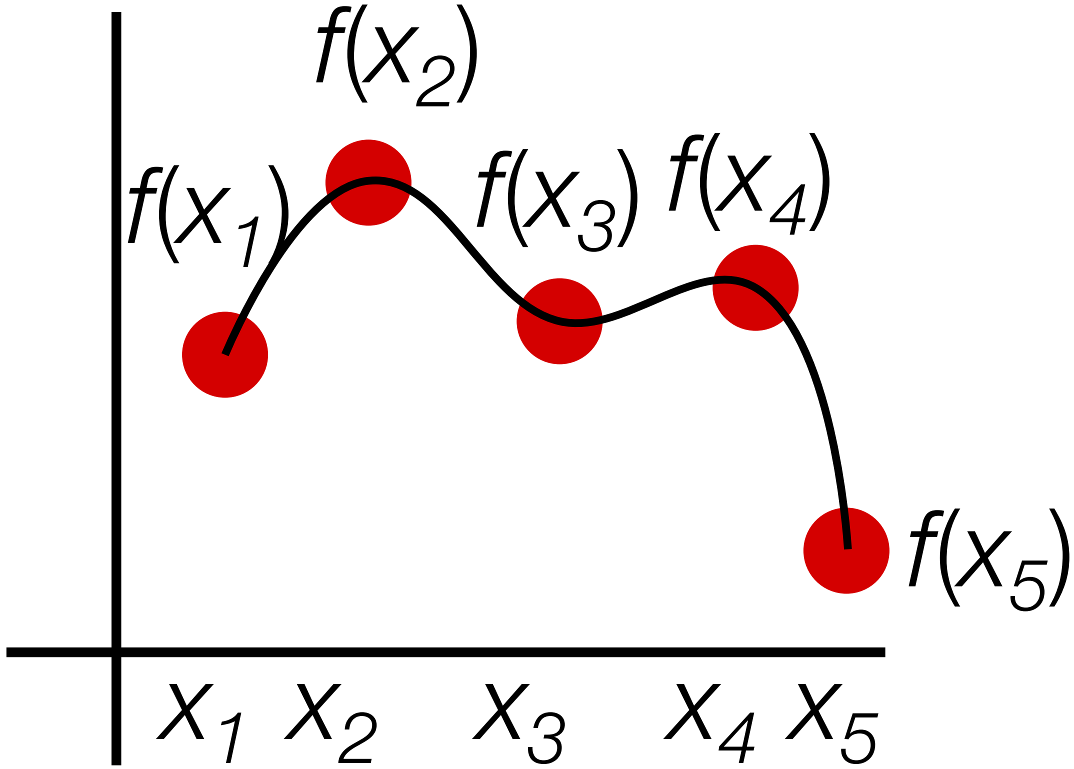
Piecewise polynomial approximation





$$\ell(x) = f(x_{i-1})\frac{x - x_i}{x_{i-1} - x_i} + f(x_i)\frac{x - x_{i-1}}{x_i - x_{i-1}}$$

In each subinterval



$$\ell(x) = f(x_{i-1})\frac{x - x_i}{x_{i-1} - x_i} + f(x_i)\frac{x - x_{i-1}}{x_i - x_{i-1}}$$

In each subinterval

Piecewise polynomial approximation

Linear Uses a set of functions values & points

Good if there are *many* points

Quadratic Same info

Can use extra point to match midpoints

Cubic Hermite Uses points, function values, and derivatives

Matches the function values and derivatives!

Cubic Splines Uses points, function values

A twice continuously differentiable interpolant!

Piecewise polynomial approximation

Linear

$$\ell(x) = f(x_{i-1})\frac{x - x_i}{x_{i-1} - x_i} + f(x_i)\frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Easy!

Quadratic

Easy!

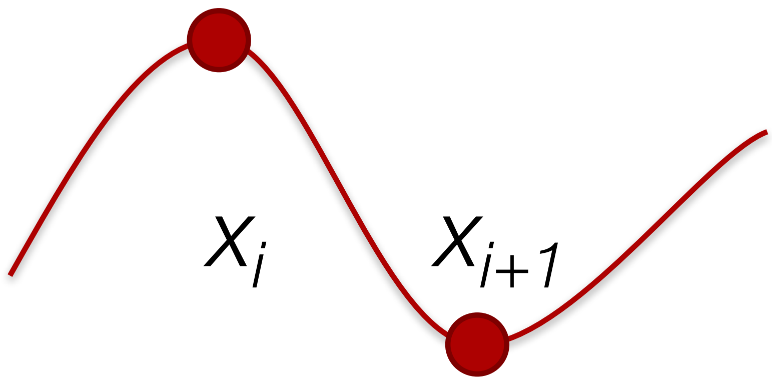
Cubic Hermite

Local work

Cubic Splines

Global work

Cubic Hermite



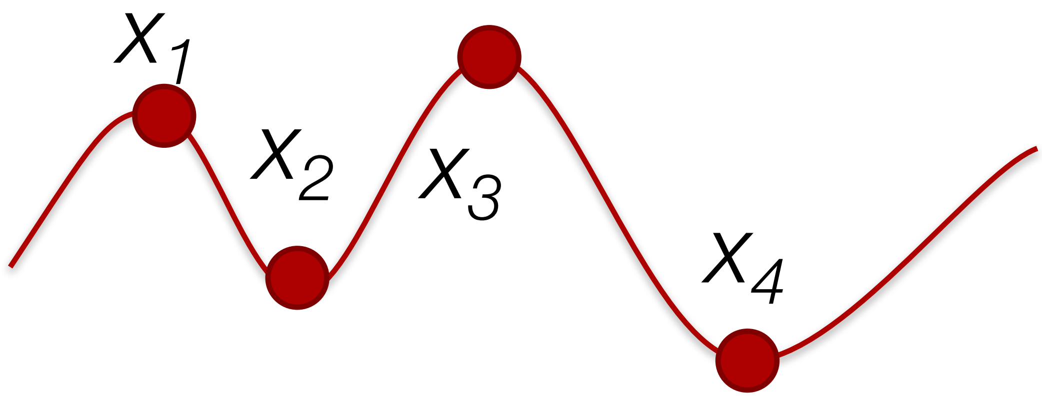
$f(x_i)$ $f'(x_i)$
$f(x_{i+1})$ $f'(x_{i+1})$

4 parameters, 4 unknowns in the cubic polynomial between x_i, x_{i+1}

Fit via differentiation.

One continuous derivative! See the book.

Cubic Splines



$f(x_i)$ $f(x_{i+1})$

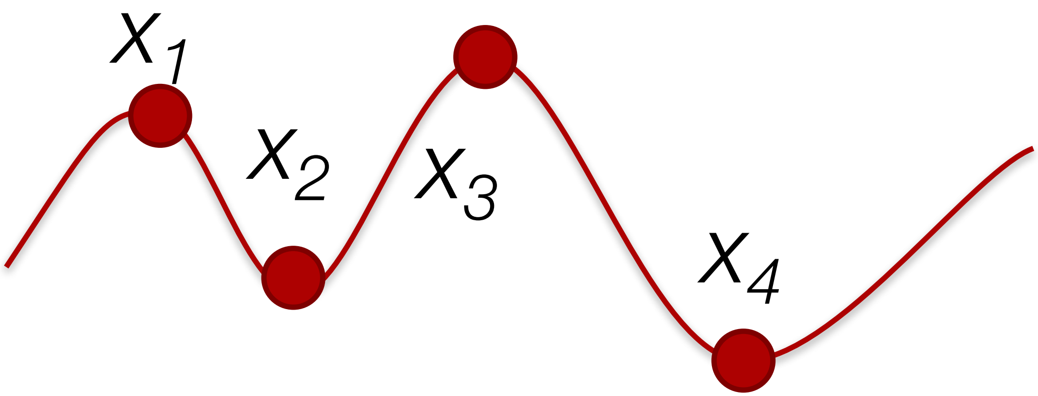
4(n-1) parameters, e.g. 4 unknowns in the cubic polynomial between x_i, x_{i+1}

Matching points gives 2(n-1) constraints

Derivatives are continuous (n-3) constraints

2nd derivatives are continuous (n-3) constraints

Cubic Splines



$f(x_i)$ $f(x_{i+1})$

$$s_i''(x) = z_{i-1}\frac{x - x_i}{x_{i-1} - x_i} + z_i \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Piecewise linear second derivative

$$\begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-2} \\ & & & \beta_{n-2} & \alpha_{n-1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{n-1} \end{bmatrix}$$

Continuous derivative gives us a linear system