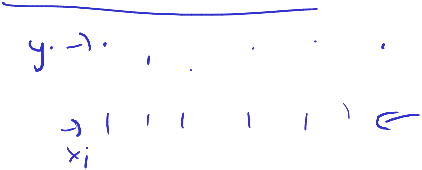
- Examlet 5
- 4CH: Assignment 1 due today



n points

n basis functions $(\varphi_i)_{i=1}^{n}$

$$V\vec{\alpha}=\vec{y}$$

$$p_{n-1}(x)=\sum \alpha_i \varphi_i(x)$$

Vdm: equispaced + monomials

gen Vdm: general nodes + general modes

Modes and Nodes (aka Functions and Points)

Both function basis and point set are under our control. What do we pick?

Ideas for basis functions:

- Monomials $1, x, x^2, x^3, x^4, \ldots$
- Functions that make $V = I \to$ 'Lagrange basis'
- Functions that make V triangular $\to$ 'Newton basis'
- *Splines* (piecewise polynomials)
- *Orthogonal polynomials*
- Sines and cosines
- 'Bumps' (*'Radial Basis Functions'*)

Ideas for points:

- Equispaced
- *'Edge-Clustered'* (so-called Chebyshev/Gauss/... nodes)

Specific issues:

- Why *not* monomials on equispaced points?
  Demo: Monomial interpolation [cleared]
- Why not equispaced?
  Demo: Choice of Nodes for Polynomial Interpolation [cleared]

Lagrange Interpolation

Find a basis so that $V = I$, i.e.

$$\varphi_j(x_i) = \begin{cases} 1 & i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Three nodes: x_1, x_2, x_3

$$\varphi_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}$$

$$\varphi_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}$$

$$\varphi_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

Lagrange Polynomials: General Form

$$\varphi_j(x) = \frac{\prod_{k=1,k\neq j}^{m}(x - x_k)}{\prod_{k=1,k\neq j}^{m}(x_j - x_k)}$$

Write down the Lagrange interpolant for nodes $(x_i)_{i=1}^m$ and values $(y_i)_{i=1}^m$.

$$p_{n-1}(x) = \sum_{i=1}^{n} y_i \, \varphi_i(x)$$

"interpolating polynomial (of degree n-1)"

Newton Interpolation

Find a basis so that V is triangular.

$$\varphi_j(x) = \prod_{k=1}^{j} (x - x_k)$$

$\Rightarrow$ V triangular

$\Rightarrow$ solve (@ $O(n^2)$ cost) with fw/bw subst.

$$\begin{matrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{matrix} \qquad \frac{y_2 - y_1}{x_2 - x_1} \qquad \frac{y_3 - y_2}{x_3 - x_2}$$

$\rightarrow$ coefficients can be read from divided differences

Why not Lagrange/Newton?

unfriendly for calculus

Chebyshev Polynomials: Definitions

$(f,g) = \int_{-1}^{1} f(x)\,g(x)\,w(x)\,dx$

$w=1$ leads to Legendre

Three equivalent definitions:

$w(x)=$

- Result of Gram-Schmidt with weight $1/\sqrt{1-x^2}$. What is that weight?

  (Like for Legendre, you won't exactly get the standard normalization if you do this.)

- $T_k(x) = \cos(k\cos^{-1}(x))$
- $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ plus $T_0 = 1$, $T_1 = x$

DLMF

Chebyshev Interpolation

What is the Vandermonde matrix for Chebyshev polynomials?

What nodes?

$$x_i = \cos\left(\frac{i}{k}\pi\right) \quad i = 0 \dots k)$$

Chebyshev nodes of the second kind.

$$V_{ij} = \cos\left(j \cdot \cos^{-1}\left(\cos\left(\frac{i}{k}\pi\right)\right)\right)$$

$$= \cos\left(\frac{j \cdot i}{k}\pi\right)$$

V is the matrix of the discrete cosine transform (DCT), a variant of the Fourier transform.

∃ Fast Fourier Transform (FFT), so that matvecs cost $n \log n$

Chebyshev Nodes

Might also consider roots (instead of extrema) of T_k:

$$x_i = \cos\left(\frac{2i-1}{2k}\pi\right) \quad (i = 1\ldots, k).$$

Vandermonde for these (with T_k) can be applied in $O(N \log N)$ time, too.

Edge-clustering seemed like a good thing in interpolation nodes. Do these do that?

Yes

Demo: Chebyshev Interpolation [cleared] (Part I-IV)

Chebyshev Interpolation: Summary

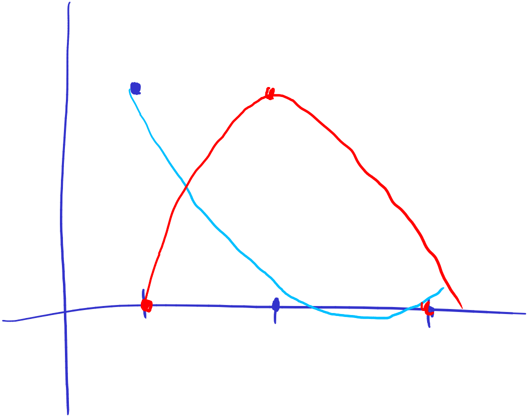
Approximation Theory
and
Approx. Practice

- Chebyshev interpolation is fast and works extremely well
- http://www.chebfun.org/ and: ATAP
- In 1D, they're a very good answer to the interpolation question
- But sometimes a piecewise approximation (with a specifiable level of smoothness) is more suited to the application

In-Class Activity: Interpolation

In-class activity: Interpolation

$f - p_{n-1}$



Truncation Error in Interpolation

If f is n times continuously differentiable on a closed interval I and $p_{n-1}(x)$ is a polynomial of degree at most n that interpolates f at n distinct points $\{x_i\}$ $(i = 1, ..., n)$ in that interval, then for each x in the interval there exists ξ in that interval such that

$\xi(x)$

$$f(x) - p_{n-1}(x) = \frac{f^{(n)}(\xi)}{n!}\underline{(x - x_1)(x - x_2)\cdots(x - x_n)}.$$

interp. error is zero at nodes

$$R(x) := f(x) - p_{n-1}(x)$$

$$Y_x(t) = R(t) - \frac{R(x)}{W(x)} W(t). \qquad W(t) = \prod_{i=1}^{n} (x - x_i)$$

Truncation Error in Interpolation: cont'd.

$$Y_x(t) = R(t) - \frac{R(x)}{W(x)} W(t) \quad \text{where} \quad W(t) = \prod_{i=1}^{n} (t - x_i)$$

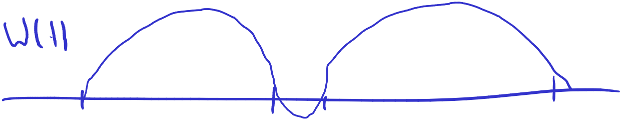
- The x_i are roots of R and W.
  $Y_x(x) = 0$. Assuming x is distinct from the x_i,
  Y_x has $n+1$ roots.
- Y_x' has n roots, ... $Y_x^{(n)}$ has at least one root in the interval. Call that root ξ.
- $p_{n-1}(x)$ is poly of degree $\le n-1$.
  $R^{(n)}(t) = f^{(n)}(t)$ $\qquad R = f - p_{n-1}$
  $Y_x^{(n)}(t) = f^{(n)}(t) - \frac{R(x)}{W(x)} n!$

- Plug in ξ

$$0 = f^{(n)}(\xi) - \frac{R(x)}{W(x)} n!$$

$$R(x) = f - p_{n-1} = \frac{f^{(n)}(\xi)}{n!} W(x)$$

$W(x)$



Error Result: Connection to Chebyshev

What is the connection between the error result and Chebyshev interpolation?

- Good nodes would make $|W(t)|$ as small as possible
- Chebyshev is the optimal node set to control $|W(x)|$.
- Cheby nodes only nearly optimal in terms of the Lebesgue const.
- Chebyshev best-approximating polynomial $\neq$ Chebyshev interpolant.

Demo: Chebyshev Interpolation [cleared] (Part V)

Error Result: Simplified Form

Boil the error result down to a simpler form.

- **Demo:** Interpolation Error [cleared]
- **Demo:** Jump with Chebyshev Nodes [cleared]