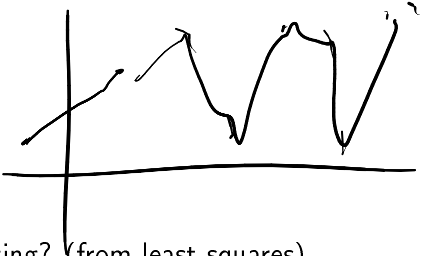
Today:

- Interpolation

Announcements:

- Exam 1 + 3
- Exam 1 + 4
- HW 10

Interpolation: Setup



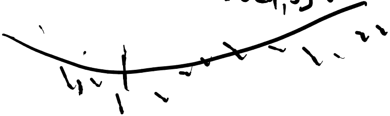
Given: $(x_i)_{i=1}^N$, $(y_i)_{i=1}^N$
Wanted: Function f so that $f(x_i) = y_i$

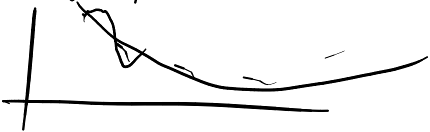
How is this not the same as function fitting? (from least squares)

→ assume underlying function exists

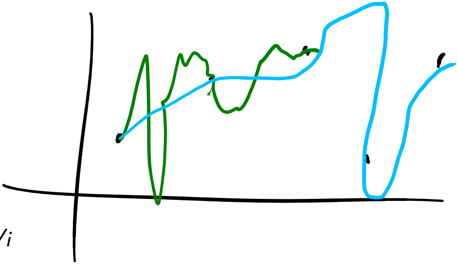
→ zero residual

→ $\max_{x\in[a,b]} |f_{\text{true}}(x) - f(x)|$





Interpolation: Setup (II)



Given: $(x_i)_{i=1}^N$, $(y_i)_{i=1}^N$
Wanted: Function f so that $f(x_i) = y_i$

Does this problem have a unique answer?

Interpolation: Importance

Why is interpolation important?

Making the Interpolation Problem Unique

Interpolant $\rightarrow$ $f(x) = \sum_{j=0}^{N_{\text{func}}-1} \alpha_j \varphi_j(x)$

$i = 1 \dots n$

$$y_i = f(x_i) = \sum_{j=0}^{n-1} \alpha_j \underbrace{\varphi_j(x_i)}_{\text{basis functions}}$$

points

$$\underbrace{\begin{pmatrix} \varphi_0(x_1) & \varphi_1(x_1) & \\ \varphi_0(x_2) & & \\ & & \end{pmatrix}}_{\text{generalized Vandermonde matrix}} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Existence/Sensitivity

Solution to the interpolation problem: Existence? Uniqueness?

same as linear system

Sensitivity?

$\sim$ conditioning of the linear system

$$\|\underset{\text{relew.}}{\text{coefficients}}\| \le \kappa(V)\, \|\overset{\text{rel. error in}}{\vec{y}}\|$$

$$\max_{x\in[a,b]} |f(x)| \le \Lambda \, \|\vec{y}\|_\infty$$

Λ: Lebesgue constant.

$\Lambda(\text{points, basis})$

For polynomial interp.: $\Lambda(\text{points})$

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
- Why not equispaced?
  Demo: Choice of Nodes for Polynomial Interpolation

Lagrange Interpolation

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

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

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

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

Lagrange Polynomials: General Form

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

Newton Interpolation

Find a basis so that V is triangular.

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

→ solve is fw/bw subst → $O(n^2)$

→ divided differences → $O(n^2)$

Why not Lagrange/Newton?

doing calculus on Lag/Newt is expensive + unpleasant

Better conditioning: Orthogonal polynomials

What caused monomials to have a terribly conditioned Vandermonde?

nearly lin. dep.

What's a way to make sure two vectors are *not* like that?

orthogonality

But polynomials are functions!

vec. $\to$ $(\vec{f}, \vec{g}) = \vec{f} \cdot \vec{g} = \sum_{i=1}^{n} f_i \cdot g_i$

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

Constructing Orthogonal Polynomials

How can we find an orthogonal basis?

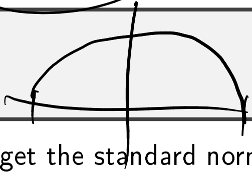
Demo: Orthogonal Polynomials — Obtained: Legendre polynomials.

But how can I practically compute the Legendre polynomials?

Chebyshev Polynomials: Definitions

Three equivalent definitions:

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

Demo: Chebyshev Interpolation (Part 1)

Chebyshev Interpolation

$$\varphi_j(x) = \cos\left(j \cos^{-1}(x)\right)$$

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

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

What is the Vandermonde matrix for Chebyshev polynomials?

DCT $\rightarrow$ Fourier transform $\rightarrow$ FFT

$O(n \log n)$

Chebyshev Nodes

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

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

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

It turns out that we were still looking for a good set of interpolation nodes.

We came up with the criterion that the nodes should bunch towards the ends. Do these do that?

Demo: Chebyshev Interpolation (Part 2)

Chebyshev Interpolation: Summary

- 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

PDE

$$\frac{\partial}{\partial x}(1) = 0$$

$$\frac{\partial}{\partial x}\left(e^{i\alpha x}\right) = i\alpha e^{i\alpha x}$$

$$\|\cdot\|_\infty = \alpha$$

In-Class Activity: Interpolation

In-class activity: Interpolation

Interpolation Error

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

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