Today:

- Interpolation

Announcements:

- Examlet 3
- Examlet 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)

\[
\text{max}_{x \in \mathbb{R}^N} \| f(x) - y \|_2
\]

\[
\rightarrow \text{assume interpolated function exists}
\]

\[
\rightarrow \text{zero residual}
\]

\[
\rightarrow \max_{x \in \mathbb{R}^N} \| f(x) - y \|_2
\]
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

\[ f(x) = \sum_{j=0}^{N_{\text{free}} - 1} \alpha_j \psi_j(x) \]

\[ \begin{bmatrix} y_0(x_0) & y_1(x_0) & \ldots & y_{n-1}(x_0) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \]

point

generalized Vandermonde matrix
Existence/Sensitivity

Solution to the interpolation problem: Existence? Uniqueness?

same as linear system

Sensitivity?

The conditioning of the linear system is given by

$$\|\text{coefficients}\| \leq k(V) \| \frac{\partial}{\partial y} \|$$

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

\(\Lambda\): Lebesgue constant.

\(\Lambda\) (points, basis) (for polynomial interp.)
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 $V = I \rightarrow$ ‘Lagrange basis’
- Functions that make $V$ triangular $\rightarrow$ ‘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
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.

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

\[ \rightarrow \text{solve is fw/bsw subst } \rightarrow O(n^2) \]

\[ \rightarrow \text{divided differences } \rightarrow O(n^2) \]

Why not Lagrange/Newton?

Doing calculus on Lag/Newton is expensive + unpleasant.
Better conditioning: Orthogonal polynomials

What caused monomials to have a terribly conditioned Vandermonde?

near lin. dep.

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

Orthogonality

But polynomials are functions!

\[ \text{vec. } \rightarrow (f, g) = f \cdot g = \sum_{i=1}^{n} f_{i} \cdot g_{i} \]

\[ \text{func. } \rightarrow (f, g) = \int_{-1}^{1} f(x) g(x) \, dx \]
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 $T_0 = 1$, $T_1 = x$

**Demo:** Chebyshev Interpolation  (Part 1)
Chebyshev Interpolation

\[ \phi_j(x) = \cos \left( j \cos^{-1}(x) \right) \]

\[ V_{ij} = \cos \left( j \left( \frac{i}{n} \pi \right) \right) \]

\[ x_i = \cos \left( \frac{i}{n} \pi \right) \]

What is the Vandermonde matrix for Chebyshev polynomials?

\[ \text{DCT} \rightarrow \text{Fourier Transform} \rightarrow \text{FFT} \]

\[ O(n \log n) \]
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/](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
\frac{\partial}{\partial x} (A) = 0

\frac{\partial}{\partial x} (e^{i2x}) = i2e^{i2x}

\mathbf{\mathbb{I}} \cdot \mathbf{\mathbb{I}}_{\nu_0} = 2
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 \( \{x_i\} (i = 1, \ldots, n) \) in that interval, then for each \( x \) in the interval there exists \( \xi \) in that interval such that

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