## Overview

- Interpolation
  - orth. poly
  - taking derivs. / integrating
- eigenvalues

solvable using solve_tri $\to U_B^{-1} W$

$$PA = LU \rightsquigarrow A = P^T L U$$

Q:

$$U A L \times B P = Z$$

$$\hookrightarrow U A L \times B = Z P^T \quad \cdot P^T$$

$$\hookrightarrow A L \times B = U^{-1} Z P^T \quad / U^{-1}$$

$$\hookrightarrow U_A L_A L \times U_B L_B = U^{-1} Z P^T / U_A^{-1}.$$

$$\hookrightarrow L_A L \times U_B L_B = (U_A^{-1} Z) P^T$$

$P_A^T L_A U_A$

scipy.linalg.solve_triangular

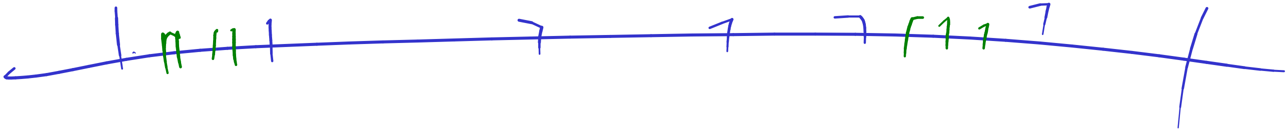
$$W U_B^{-1}$$

$$= ((W U_B^{-1})^T)^T$$

$$= (U_B^{-T} W^T)$$

lower $\Delta$ $\leftarrow$ solve tri

**Demo:** Choice of Nodes for Polynomial Interpolation



## Interpolation: Choosing Basis Function and Nodes

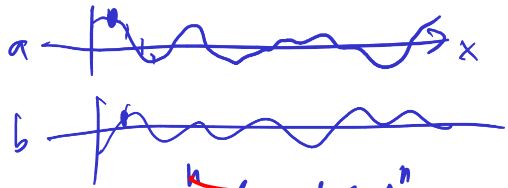
Both function basis and point set are under our control. What do we pick?

Ideas for basis functions:

$\vec{a} \cdot \vec{b} = 0$

- 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')

orthog $\left( x^2 + 5x + 7 = a(x),\ x^3 - 5 = b(x) \right.$




"$a(x) \cdot b(x)$"

Ideas for nodes:

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

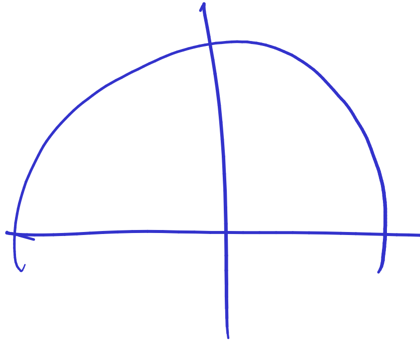
"dot product" of functions $= (a, b) = \int_5^{17} a(x) \cdot b(x)\, dx$

## Better Conditioning: Orthogonal Polynomials

- What caused monomials to have a terribly conditioned Vandermonde?
- What's a way to make sure two vectors are *not* like that?
- But polynomials are functions!

- But how can I practically compute the Legendre polynomials?

## Another Family of Orthogonal Polynomials: Chebyshev



Three equivalent definitions:

- Result of Gram-Schmidt with weight $1/\sqrt{1-x^2}$
  - What is that weight?
- $T_k(x) = \cos(k\cos^{-1}(x))$
- $T_k(x) = 2x\,T_k(x) - T_{k-1}(x)$

**Demo:** Chebyshev interpolation part I

- What are good nodes to use with Chebyshev polynomials?

$x_i$

$T_k(x_i) = \cos(k \cdot \cos^{-1}(x_i))$

$x_i = \cos(y_i) \leadsto \cos(k \cdot y_i)$

## Chebyshev Nodes

$0 \longrightarrow \pi$

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

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

$\longrightarrow (-1, 1)$

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 II

## Calculus on Interpolants

- Suppose we have an interpolant $\tilde{f}(x)$ with $f(x_i) = \tilde{f}(x_i)$ for $i = 1, \dots, n$:

$$\tilde{f}(x) = \alpha_1 \varphi_1(x) + \cdots + \alpha_n \varphi_n(x)$$

  How do we compute the derivative of $\tilde{f}$?

$$\tilde{f}'(x) = \alpha_1 \varphi_1'(x) + \cdots + \alpha_n \varphi_n'(x)$$

- Suppose we have function values at nodes $(x_i, f(x_i))$ for $i = 1, \dots, n$ for a function $f$. If we want $f'(x_i)$, what can we do?

$$V\vec{\alpha} = \vec{f} \rightsquigarrow \vec{\alpha} = V^{-1}\vec{f}$$

$$\text{point} \downarrow \underbrace{\begin{pmatrix} \varphi_1'(x) & & \varphi_n'(x) \\ \\ \varphi_1'(x_n) & & \varphi_n'(x_n) \end{pmatrix}}_{V'} \vec{\alpha} = \begin{pmatrix} \alpha_1 \varphi_1'(x) + \cdots + \alpha_n \varphi_n'(x_1) \\ \\ \\ \end{pmatrix}$$

$$D = V' V^{-1}$$

$$V'V^{-1}\vec{f} \sim D\vec{f}$$

## About Differentiation Matrices

- How could you find coefficients of the derivative?

- Give a matrix that finds the second derivative.

**Demo:** Taking derivatives with Vandermonde matrices