Overview

- Interpolation
  - orth poly
  - taking derivatives/integrals
- Eigenvalues

\[
\text{solvable} \rightarrow U_B^{-1}W
\]

\[
P_A = L_H \rightarrow A = P_\Delta H
\]

\[
A = P_\Delta H
\]

\[
U_A \perp X \lor P = Z
\]

\[
C_3: UAC \times B = Z \perp P^r
\]

\[
L, A, C \times U_B \perp C_B = U^{-1} Z \perp P^r / U_r^r
\]

\[
\text{scipy.linalg.solve_triangular}
\]

\[
W U_B^{-1} = ((W U_B^{-1})^\top)^\top
\]

\[
\begin{pmatrix}
U_0^\top & W^\top
\end{pmatrix}
\]

\[
\text{solve tri}
\]
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:

- Monomials 1, x, x^2, x^3, x^4, ...
- Functions that make \( V = I \) → ‘Lagrange basis’
- Functions that make \( V \) triangular → ‘Newton basis’
- Splines (piecewise polynomials)
- Orthogonal polynomials
- Sines and cosines
- ‘Bumps’ (‘Radial Basis Functions’)

Ideas for nodes:

- Equispaced
- ‘Edge-Clustered’ (so-called Chebyshev/Gauss/… nodes)

\[ \mathbf{a} \cdot \mathbf{b} = 0 \]

\[ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \]

\[ x^2 + 5x + 7 = q(x) \]

\[ x^3 - 5 = l(x) \]

\[ a \rightarrow x \rightarrow b \]

\[ a(x) \cdot b(x) \]

"dot product" of functions \( (a, b) = \int_a^b a(x) \cdot b(x) \, dx \)
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?

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

$$x_i = \cos(y_i) \approx \cos(k \cdot y_i)$$
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 II
Calculus on Interpolants

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

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

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

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