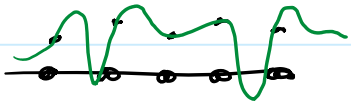
Midterm 2 review

Tuesday, November 4, 2025 9:30 PM

Interpolation

- If two polynomials, both degree n or less, agree at n+1 or more points, then these are in fact the same polynomial

- Runge phenomenon is when a high degree polynomial interpolant oscillates wildly near boundaries



Can be fixed by using Chebyshev nodes which cluster near boundaries

- Error thm: if $f \in C^{n+1}[a,b]$, $\{x_0, \dots, x_n\}$ distinct nodes in $[a,b]$, and p_n is the degree n-or-less polynomial interpolant of f at these nodes, then $\exists\, \xi \in [a,b]$ s.t.

$$f(x) = p(x) + f^{(n+1)}(\xi)\frac{1}{(n+1)!}(x-x_0)(x-x_1)\cdot\dots\cdot(x-x_n)$$

Independent of uniform v. chebyshev nodes, or how we find p_n

- Finding p_n given $\{(x_0,y_0), \dots, (x_n,y_n)\}$

how you did it in 1890 → ① Newton Divided Differences

② Lagrange interpolating polynomial,

how you do a small example on an exam →

$$L_{n,k}(x) = \prod_{i \neq k} \frac{x - x_i}{x_k - x_i}$$

$$p_n(x) = \sum_{k=0}^{n} y_i \cdot L_{n,k}(x)$$

UNSTABLE

Cost: $O(n^2)$ every time we evaluate p_n

how a computer does it → ③ Barycentric Formula variation

Tuesday, November 4, 2025 10:07 PM

③ Barycentric

$$w_k = \frac{\text{const}}{\prod_{i \neq k} (x_k - x_i)}$$ ← anything. Choose to limit overflow/underflow

precompute all weights $O(n^2)$

$$p_n(x) = \frac{\sum_{k=0}^{n} \frac{y_k \cdot w_k}{x - x_k}}{\sum_{k=0}^{n} \frac{w_k}{x - x_k}}$$

costs $O(n)$ to evaluate at each new x

④ Monomial basis /

⑤ Generic polynomial basis $\phi_0, \dots, \phi_n$ for $\mathcal{P}_n$:

$p_n(x) = \sum_{k=0}^{n} a_k \phi_k(x)$, $p_n(x_i) = y_i$, $i = 0, \dots, n$

So solve

$$\begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \dots & \phi_n(x_0) \\ \vdots & & & \\ \phi_0(x_n) & \phi_1(x_n) & \dots & \phi_n(x_n) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix}$$

- lower triangular for Newton D.D. basis
- diagonal for Lagrange basis
- dense, ill-conditioned Vandermonde for monomial basis

- Our two use cases :

① $\{(x_i, y_i)\}_{i=0}^{n}$ just data, want to fit a curve
We don't have control over locations x_i

② We have a function f, want to approximate it by p_n
We choose nodes x_i, then set $y_i = f(x_i)$

- Hermite interpolation

Use case ②, $y_i = f(x_i)$ and $\tilde{y}_i = f'(x_i)$, $i = 0, 1, \dots, n$

Insist $p(x_i) = y_i$ and $p'(x_i) = \tilde{y}_i$ $2n+2$ constraints

p is degree $2n+1$ so it has $2n+2$ coefficients

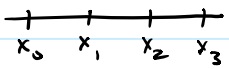
technique to construct fit looks like a fancier version of Lagrange

Tuesday, November 4, 2025 10:33 PM

• Splines

piecewise polynomials...

ex. $n=3$



$$S(x) = \begin{cases} P_0(x) & x_0 \le x < x_1 \\ P_1(x) & x_1 \le x < x_2 \\ P_2(x) & x_2 \le x \le x_3 \end{cases}$$

↑ "2" does not refer to degree 2

... that interpolate...

... and require S to be continuous (linear splines)

... and S' is continuous (quadratic splines)] +1 boundary condition

... and S'' is continuous (cubic splines)] +2 boundary conditions "B.C."

# eq'n	
$2n$	n segments, interpolation gives 2 linear equations per segment
quadratic $+ n-1$	$(n-1)$ interior nodes, self-consistent derivatives here: $\lim_{x \to x_1^-} S'(x) = \lim_{x \to x_1^+} S'(x)$ i.e. $P_0'(x_1) = P_1'(x_1)$ etc.
cubic $+ n-1$	... self-consistent second derivatives

degrees of freedom, linear: $2n$

quadratic: $3n$

cubic: $4n$

For C^2 cubic splines, common B.C. include

- Natural / free, $S''(x_0) = 0$ and $S''(x_n) = 0$
- clamped $S'(x_0) = f'(x_0)$, $S'(x_n) = f'(x_n)$ } if known. CAREFUL! Scipy let's you do this without specifying $f'(x_0)$. It assumes $f'(x_0) = 0$
- not-a-knot $P_0'''(x_1) = P_1'''(x_1)$, $P_{n-2}'''(x_{n-1}) = P_{n-1}'''(x_{n-1})$
- periodic $S'(x_0) = S'(x_n)$, $S''(x_0) = S''(x_n)$

— How to solve:

Key: write $P_j = a_j + b_j(x-x_j) + c_j(x-x_j)^2 + d_j(x-x_j)^3$

Find all $\{a_j, b_j, c_j, d_j\}$. Some are easy: $a_j = f(x_j)$ ✓

By hand, just work through

On a computer, reduce to a tridiagonal system, $n \times n$, which can be solved in $O(n)$ time.

— Theory:

If $f \in C^4[a,b]$ and $\|f^{(4)}\|_{L^\infty} \le M$ and S is the clamped cubic spline interpolant, then $\forall x \in [a,b]$

$$|f(x) - S(x)| \le M \cdot \frac{5}{384} \cdot \max_{0 \le j \le n-1} (x_{j+1} - x_j)^4$$

i.e. uniformly spaced, spacing $h = O(1/n)$,

error $= O(h^4) = O(1/n^4)$

Same is true for not-a-knot

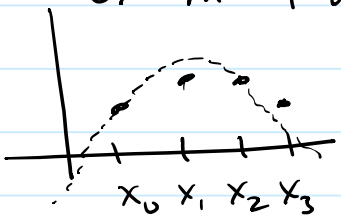
not as good as interpolation w/ a degree n polynomial ... but FAST ($O(n)$ not $O(n^2)$) and no Runge phenomenon.

Tuesday, November 4, 2025 10:53 PM

- discrete ℓ^2

ex: $m=4$ datapoints, want p to be degree 2 or less

to help w/ generalization/ extrapolation, prevent overfitting



Probably can't interpolate

$$\min_{\vec{a}\in\mathbb{R}^3} \sum_{i=0}^{3} (\hat{y}_i - y_i)^2, \quad \hat{y}_i = p(x_i), \quad p(x) = a_0 + a_1 x + a_2 x^2$$

Find $\vec{a} \in \mathbb{R}^3$

$$X = \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \in \mathbb{R}^{4\times 3}, \quad \hat{\vec{y}} = X\cdot\vec{a}$$

Vandermonde matrix

so $\min_{\vec{a}} \underbrace{\| X\vec{a} - \vec{y} \|_2^2}_{g(a)}$

Set $\nabla g(\vec{a}) = 0$ to get

$$X^T X\vec{a} - X^T\vec{y} = 0, \text{ i.e. } X^T X\vec{a} = X^T\vec{y}$$

(NORMAL EQUATIONS)

↑ solve for $\vec{a}$

or, if weighted $\min \sum w_i (\hat{y}_i - y_i)^2$

then solve $X^T W X\vec{a} = X^T W\vec{y}$, $W = \text{diag}(\vec{w})$

- discrete ℓ^1 or ℓ^∞

are linear programs, ie. convert to form $\min_{\vec{z}} c^T\vec{z}$ s.t. $A\vec{z} = \vec{b}$, $\vec{b} \geq \vec{0}$

- Regularize, eg. Tikhonov
- Total least squares

Tuesday, November 4, 2025 11:02 PM

- Continuous, L^2, $E = \|f - p\|_{L^2}^2$, using $\langle f, g \rangle_{L^2} = \int_a^b f(x) g(x) \underbrace{w(x)}_{\text{wt. (optional)}} dx$

think of this as the limit of $\{x_i\}_{i=0}^m$ as $m \to \infty$

Say, $p(x) = a_0 + a_1 x + a_2 x^2$,

or generally $\phi_0(x)$ $\phi_1(x)$ $\phi_2(x)$

"X" is a quasimatrix w/ ∞ rows, $[1, x, x^2]$ $\forall x \in [a,b]$ (still 3 columns)

informally, $X \in \mathbb{R}^{\infty \times 3}$

So $\underbrace{X^T X}_{G} \in \mathbb{R}^{3\times 3}$ is tractable.

formally, define $G \in \mathbb{R}^{3\times 3}$, $G_{k,j} = \langle \phi_k, \phi_j \rangle_{L^2}$

and define $\vec{b} \in \mathbb{R}^3$ (like $X^T y$)

$$b_k = \langle \phi_k, f \rangle$$

then solve $G\vec{a} = \vec{b}$ for $\vec{a} \in \mathbb{R}^3$

- orthogonal bases

If $\langle \phi_k, \phi_j \rangle_{L^2} = 0$ for $k \neq j$ (i.e. $\{\phi_j\}$ is orthogonal)

then G is a diagonal matrix, so G^{-1} is diagonal with $(G^{-1})_{k,k} = 1/G_{k,k}$, and solving $G\vec{a} = \vec{b}$ is simple: $\vec{a} = G^{-1}\vec{b}$ means

$$a_k = \frac{b_k}{G_{k,k}} = \frac{\langle \phi_k, f \rangle_{L^2}}{\|\phi_k\|_{L^2}^2}$$

Legendre polynomials are orthogonal for L^2 on $[-1,1]$ w/ $w(x) = 1$

Chebyshev polynomials are orthogonal for L^2 on $[-1,1]$ w/ $w(x) = \frac{1}{\sqrt{1-x^2}}$

For both of them, the first $n+1$ form a basis for $\mathcal{P}_n$ (vectorspace of polynomials of degree $\leq n$)

Created via Gram-Schmidt to $\{1, x, x^2, \dots\}$

Wednesday, November 5, 2025 6:06 AM

- **Gram Schmidt** (the variant that orthogonalizes, not orthonormalize)

Given a (non-orthogonal) basis $\psi_0, \psi_1, \psi_2, \ldots$

we'll construct an orthogonal basis $\phi_0, \phi_1, \phi_2, \ldots$

$$\phi_0 = \psi_0$$

$$\phi_1 = \psi_1 - \frac{\langle \psi_1, \phi_0 \rangle}{\|\phi_0\|^2} \cdot \phi_0$$

verify: $\langle \phi_0, \phi_1 \rangle = \langle \phi_0, \psi_1 \rangle - \frac{\langle \psi_1, \phi_0 \rangle}{\|\phi_0\|^2} \cdot \langle \phi_0, \phi_0 \rangle = 0$ ✓

$$\phi_2 = \psi_2 - \frac{\langle \psi_2, \phi_0 \rangle}{\|\phi_0\|^2} \phi_0 - \frac{\langle \psi_2, \phi_1 \rangle}{\|\phi_1\|^2} \phi_1$$

etc.

- **Approximation by rational functions** ch 8.4

2025, skip

- **Fourier Series**

On $[-\pi, \pi]$, $\{1\} \cup \{\cos(kx), \sin(kx)\}_{k=1}^{\infty}$ is orthogonal

If we represent f in this basis,

$$f(x) = a_0 + \sum_{k=1}^{\infty} a_k \cdot \cos(kx) + b_k \cdot \sin(kx)$$

Even though it's ∞, we can still find Fourier coefficients a_k, b_k using orthogonality: define $\langle f, g \rangle = \int_{-\pi}^{\pi} f(x) g(x) dx$

$$\langle \sin(jx), f(x) \rangle = \underbrace{\langle \sin(jx), a_0 \rangle}_{=0} + \sum_{k=1}^{\infty} a_k \underbrace{\langle \sin(jx), \cos(kx) \rangle}_{=0} + b_k \underbrace{\langle \sin(jx), \sin(kx) \rangle}_{=0 \text{ if } j \neq k}$$

$$= a_j \cdot \|\sin(jx)\|^2$$

$$\text{So } a_j = \frac{\langle \sin(jx), f(x) \rangle}{\|\sin(jx)\|^2} !$$

Note: $\|f\|_{L^2}^2 = \langle f, f \rangle = \langle a_0 + \sum_k a_k \cos(kx) + b_k \sin(kx), a_0 + \sum_j a_j \cos(jx) + b_j \sin(jx) \rangle$

$$= a_0 \underbrace{\|1\|_{L^2}^2}_{2\pi} + \sum_k a_k^2 \underbrace{\|\cos(kx)\|_{L^2}^2}_{\pi} + b_k^2 \underbrace{\|\sin(kx)\|_{L^2}^2}_{\pi}$$

so if ↑ is finite, so is → hence $\sum a_k^2 + b_k^2 < \infty$,

so a necessary requirement is $\boxed{a_k, b_k \to 0.}$

Wednesday, November 5, 2025 9:54 AM

- Decay of Fourier coefficients

we saw if $\|f\|_{L^2} < \infty$ then $\sum a_k^2 + b_k^2 < \infty$

so $a_k, b_k \to 0$

(we are being loose on technical details)

If f is differentiable,

$$f(x) = a_0 + \sum_{k=1}^{\infty} a_k \cos(kx) + b_k \sin(kx)$$

then $$f'(x) = \sum_{k=1}^{\infty} -k \cdot a_k \cdot \sin(kx) + k \cdot b_k \cos(kx)$$

and using the same orthogonality trick

if $\|f'\|_{L^2} < \infty$, then $\sum k^2 a_k^2 + k^2 b_k^2 < \infty$

so $a_k, b_k \to 0$ fast to compensate for the growth of k^2

general rule:

The smoother f is ...

(i.e. the more derivatives it has... taking into account the periodic extension, that is)

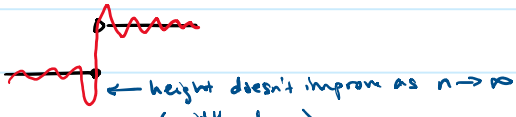
... the faster its Fourier series coefficients decay.

- Gibbs phenomenon

If f has a jump discontinuity, defining $f_n(x) = a_0 + \sum_{k=1}^{n} a_k \cos(kx) + b_k \sin(kx)$

then $\|f - f_n\|_{L^2} \to 0$

but $\|f - f_n\|_{L^\infty}$ doesn't



No "fix"

(nonequispaced nodes isn't relevant here)

- Discrete Fourier Transform

Similar but not the same as Fourier Series.

DFT $F : \mathbb{C}^N \to \mathbb{C}^N$ a linear operator w/ matrix representation

$F_{kj} = \omega^{kj}$, $k, j = 0, 1, \ldots, N-1$ where $\omega = e^{-\frac{2\pi}{N} i}$

$$F = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2N-2} \\ \vdots & & & & \\ 1 & \omega^{N-1} & \omega^{2N-2} & \cdots & \omega^{(N-1)^2} \end{bmatrix}$$

The matrix $\tilde{F} = \frac{1}{\sqrt{N}} F$ is unitary

i.e. $\tilde{F}^{-1} = \tilde{F}^*$

($A^* := \bar{A}^T$, complex conjugate transpose)

Wednesday, November 5, 2025 10:45 AM

- Fast Fourier Transform / FFT

Computing $F \cdot \vec{v}$ for a $N \times N$ DFT takes $O(N^2)$ flops naïvely, but can be reduced to $O(N \log N)$ flops using the FFT.

We did in class for when N is a power of 2, but it can be done for any N, even prime ones.

Because $F^{-1} = \frac{1}{N} F^*$, we also have a fast way to compute the inverse DFT (called the IFFT, for inverse FFT)