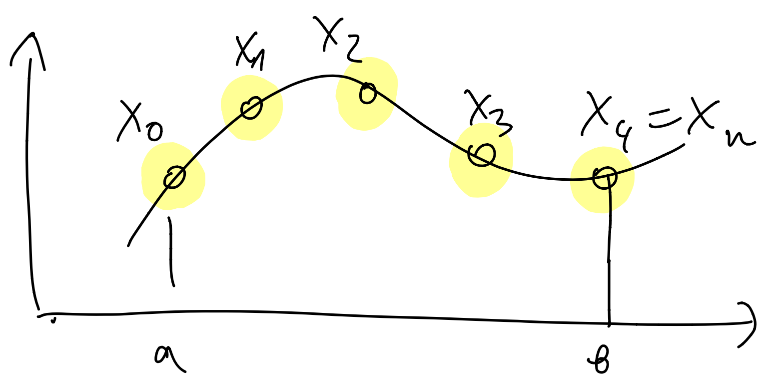
Review: Polynomial Approximation

A. Donev, Spring 2021

Interpolation

$p(x) \approx f(x)$ on $[a,b]$



$$P_n(x) = \sum_{k=0}^{n} a_k x^k \approx f(x)$$

plug

$$P_n(x_j) = f(x_j) \quad \text{interpolation}$$

You get a linear system for $\vec{a}$

$$\overleftrightarrow{V}\,\vec{a} = \vec{y} = f(x)$$

ill-conditioned for n large

$$V_{ij} = x_i^{\,j}$$

<u>Not</u> the recommended method

$$P_n(x) = \sum_{k=0}^{n} a_k L_k(x)$$

new polynomials

$$L_k \in \mathcal{P}_n$$

$\{L_k\}$ is a basis for $\mathcal{P}_n$

Best choice of basis for <u>interpolation</u> :

L_k = Lagrange polynomial

$$\begin{cases} L_k(x_k) = 1 \\ L_k(x_j) = 0, \quad j \neq k \end{cases}$$

$$\Rightarrow \quad a_k = f(x_k) = y_k$$

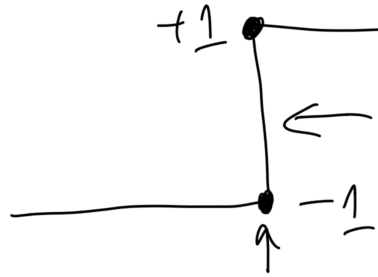
$$L_k(x) = \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)}$$

Error in <u>interpolant</u>

$$\| f - p \| \quad ?$$

$$\| f - p \|_\infty = \max_{a \le x \le b} | f(x) - p(x) |$$

pointwise



$\| f - p \|_\infty$ must be large

$$\| f - p \|_1 = \int_a^b | f(x) - p(x) | \, dx$$

$$(f, g)_{L_2} = \int_a^b \underbrace{f(x) g(x)}_{\text{real-valued}} \, dx$$

$$(f,g)_w = \int_a^b w(x) f(x) g(x)\, dx$$

weight function

$w(x) > 0$ in (a,b)

$$\|f-p\|_{L_2} = \int_{x=a}^{b} |f(x)-p(x)|^2 dx$$

If $f(x) \in \mathcal{P}_n$ then

$$p(x) = f(x) \Rightarrow \text{error} = 0$$

Interpolation is exact for polynomials of degree up to n

"Intepolation is of order $n+1$"

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi \in [a,b])}{(n+1)!} \underbrace{\prod_{k=0}^{n} (x - x_k)}_{\text{nodal polynomial}}$$

$p(x) \not\longrightarrow f(x)$ as $n \to \infty$ unless we choose nodes carefully

does NOT

even if $f(x)$ is smooth

But, if we choose nodes carefully, then

$$p(x) \xrightarrow[n \to \infty]{} f(x)$$

on $[a,b]$ for smooth $f(x)$

L_2 approximation

$$p_n^* = \underset{p(x)\in\mathcal{P}_n}{\arg\min} \|f(x)-p(x)\|^2$$

Choose an inner product

$$(f, g)$$

Define

$$\|f\|^2 = (f, f)$$

If $(f, g)_{L_2} = \int_a^b f(x)g(x)\,dx$

But other options possible

$$(f, g)_{\ell_2} = \sum_{k=0}^{n} f(x_k)g(x_k)$$

HW: Optimal ℓ_2 approximant is the interpolant

<u>Another way</u>

$$p(x) = f(x) \quad \left(\overset{=}{\underset{\smile}{\text{really}}}\right)$$

$$p(x) = \sum_{k=0}^{n} a_k \, p_k(x)$$

$\{p_k(x)\}$ is a basis for $\mathcal{P}_n$

$$\sum_{k=0}^{n} a_k \, p_k = f$$

Take inner product of both sides with p_ℓ

$$\left(P_\ell, \sum a_k P_k\right) = \left(P_\ell, f\right)$$

actual equals

$$\sum_{k=0}^{n} a_k \left(P_\ell, P_k\right) = \left(P_\ell, f\right)$$

Linear system for a_k

$$\overleftrightarrow{V}\,\vec{a} = \vec{f}$$

$$\overleftrightarrow{V}_{ij} = \left(P_i, P_j\right)$$

$$\vec{f}_k = \left(P_k, f\right)$$

Also ill-conditioned

Best choice of basis for P_n?

$$(p_\ell, p_k) = 0 \text{ if } \ell \neq k$$

$$a_\ell (p_\ell, p_\ell) = (p_\ell, f)$$

$$a_\ell = \frac{(p_\ell, f)}{(p_\ell, p_\ell)}$$

General formula

How to find $\{p_k\}$?

Use Gram-Schmidt orthogonalization starting with monomials

$$\{1, x, x^2, \dots, x^n\}$$

$$\Big\downarrow GS \to \text{inner product}$$

$$\{ P_0 = 1,\ P_1(x),\ P_2(x), \dots, P_k(x) \}$$

$$P_k(x) \in \mathcal{P}_k \quad (\text{of degree } k)$$

<u>Example</u> For $(f, g)_{L_2}$ then we get Legendre polynomials

$$\begin{cases} \varphi_0(x) = 1 \\ \varphi_1(x) = x \qquad \text{on } [-1, 1] \\ \varphi_2(x) = x^2 - \frac{1}{3} \\ \dots \end{cases}$$

For $[a, b]$ general

$$t = \left(\frac{x+1}{2}\right)(b-a) + a$$

$t \in [a, b]$

$x \longrightarrow x(t)$

$$x = \frac{2(t-a)}{b-a} - 1$$

$x \in [-1, 1]$ $\quad t \in [a, b]$

or

$x \in [a, b]$ $\quad t \in [-1, 1]$

<u>Theorem</u>: The roots of $P_k(x)$ (orthogonal polynomials) are real and distinct and in $[a, b]$.

e.g. $x = \pm\sqrt{\frac{1}{3}}$

Use the roots of orthogonal polynomials for interpolation nodes.

For Legendre polynomials the roots are called <u>Gauss points</u>

$$x = \pm\sqrt{\frac{1}{3}} \quad (2 \text{ points})$$

or

$$x = \pm\sqrt{\frac{3}{5}}, \ 0 \quad (3 \text{ points})$$

Quadrature

$$(*) \ldots \int_a^b f(x)\,dx = \sum_{k=1}^{n} w_k f(x_k)$$

standard notation

$$\begin{cases} x_k \in [a,b] & \text{nodes} \\ w_k \text{ are weights} \end{cases}$$

How to find w_k ? the best / optimal

Option 1: Make $(*)$ be exact for polynomials of degree

$0, 1, 2, \ldots, n-1$

gives you n LINEAR equations for n unknowns $\{w_k\}$

$\Longleftrightarrow$ Make $(*)$ be exact for $\{1, x, x^2, \dots, x^{n-1}\}$

<u>Option 2</u>

Approximate $f(x) \approx p_{n-1}(x)$ via <u>interpolation</u>, and integrate the interpolant

$$p_{n-1}(x) = \sum f(x_k) L_k(x)$$

$$\int f(x)\,dx \approx \int \sum \underline{f(x_k)} L_k(x)\,dx$$

$$= \sum w_k f(x_k)$$

$$w_k = \int_a^b L_k(x)\,dx$$

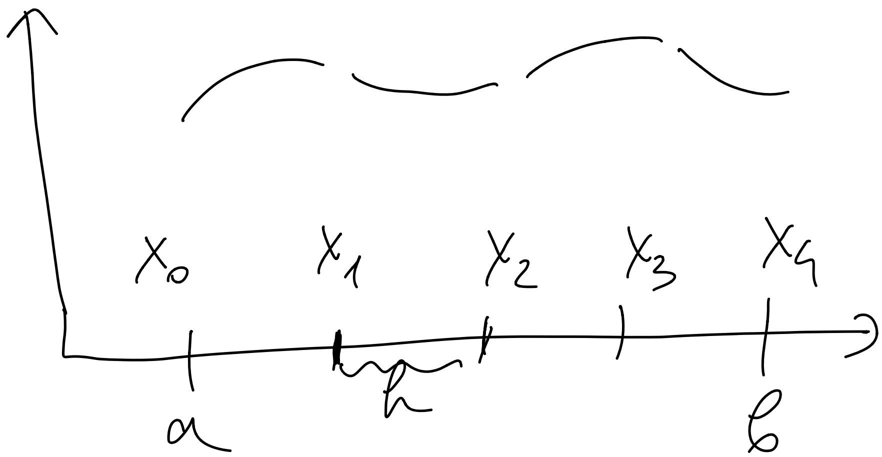
L_k: Lagrange polynomial

<u>Th</u>: If x_k are Gauss nodes, then (*) is exact for $f(x) \in P_{2n-1}$

If $n=3$, GQ is exact for polynomials of degree up to 5 $\Rightarrow$ it's <u>sixth</u> order

[Gauss quadrature is the "best" you can do

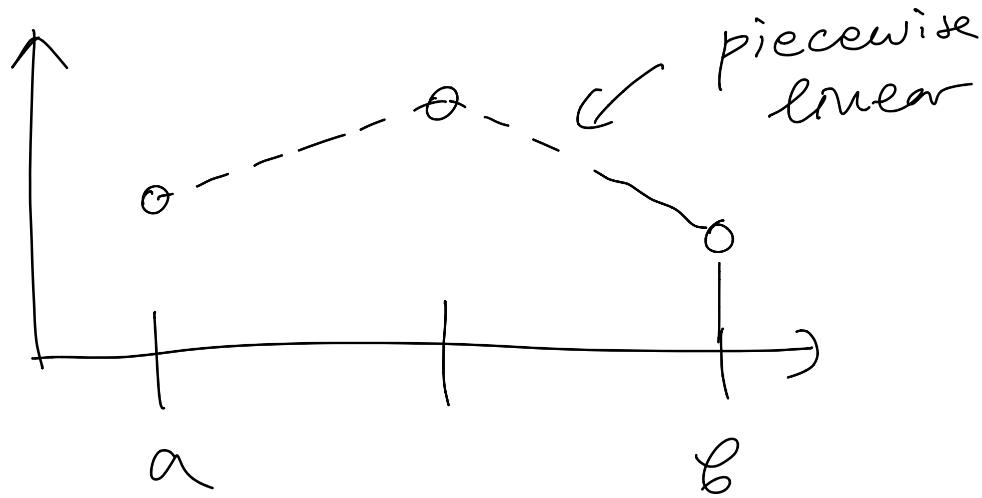
Piecewise polynomial approximation



Approximate $f(x)$ on <u>each</u> subinterval $[x_{k-1}, x_k]$ by a polynomial of some degree

$$\int_a^b f(x)\,dx = \sum_{k=1}^{n+1} \int_{x_{k-1}}^{x_k} P_k(x)\,dx$$

Option 1: Piecewise polynomial Interpolant



For quadrature, this gives

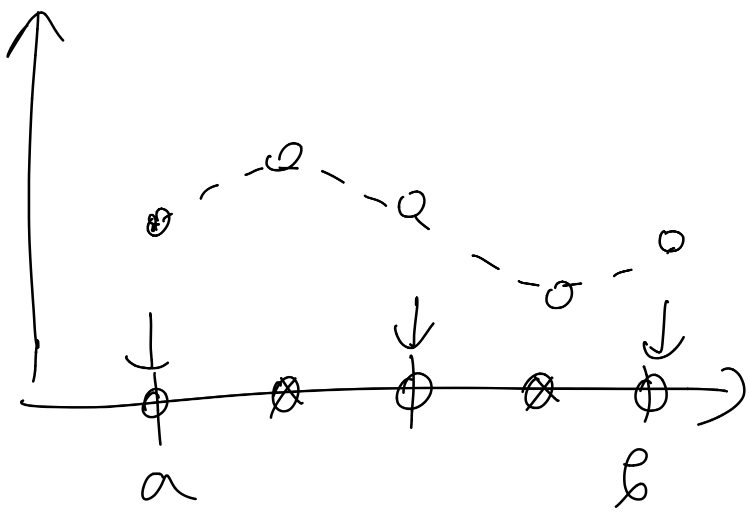
composite trapezoidal rule

$$\int_a^b f(x)\,dx \simeq h \sum_{j=0}^{n} f(x_j)$$

$$- \frac{h}{2}\left(f(a)+f(b)\right)$$

error $\sim O(h^2) \sim O\left(\frac{1}{n^2}\right)$

So double the points gives one quarter the error



This gives

(composite) Simpson formula

$$\int_a^b f(x)\,dx = \frac{b-a}{6}\left[f(a)+f(b)+4f\left(\frac{a+b}{2}\right)\right]$$

$$\int_a^b f(x)\,dx = \frac{h}{6}\left[f(a)+f(b)\right] + \frac{h}{3}\sum_{k=1}^{n-1} f(x_k) + \frac{2h}{3}\sum_{k=0}^{n-1} f(x_{k+1/2})$$

$$\text{error} \simeq O(h^4) \text{ or } O\left(\frac{1}{n^4}\right)$$

$n \to 2n$

$\text{error} \to \text{error}/16$