Polynomial approximation

Spring 2021, A. Donev

How to approximate nonlinear functions

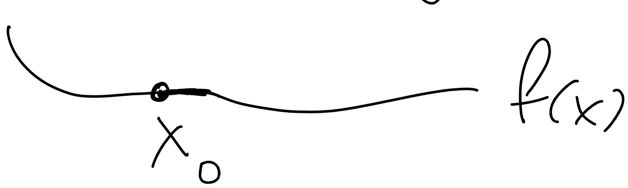
$f(x) = \exp(x)$, $\sin(x)$, $e^{\cos(x)}$

$f(x)$ can be locally approximated by a polynomial on $[a, b]$

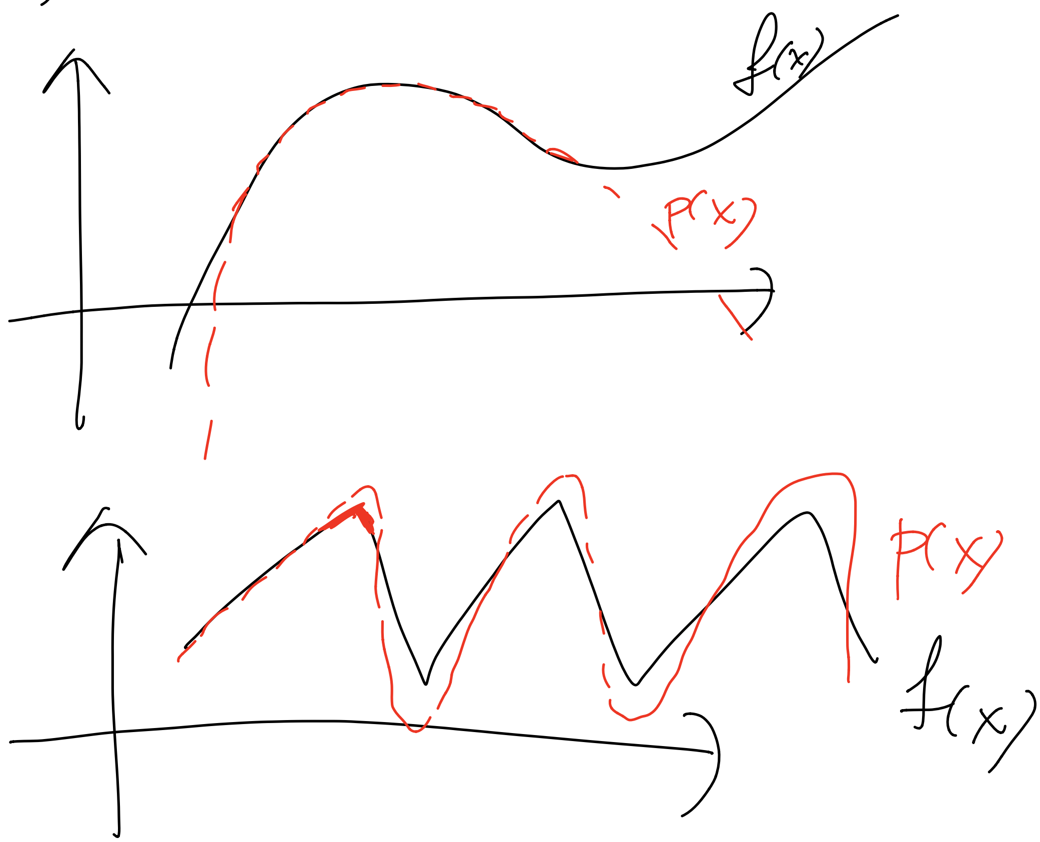
$$p(x) \approx f(x) \text{ on } [a, b]$$

polynomial

Recall Taylor series



This will work "well" if $f(x)$ is smooth

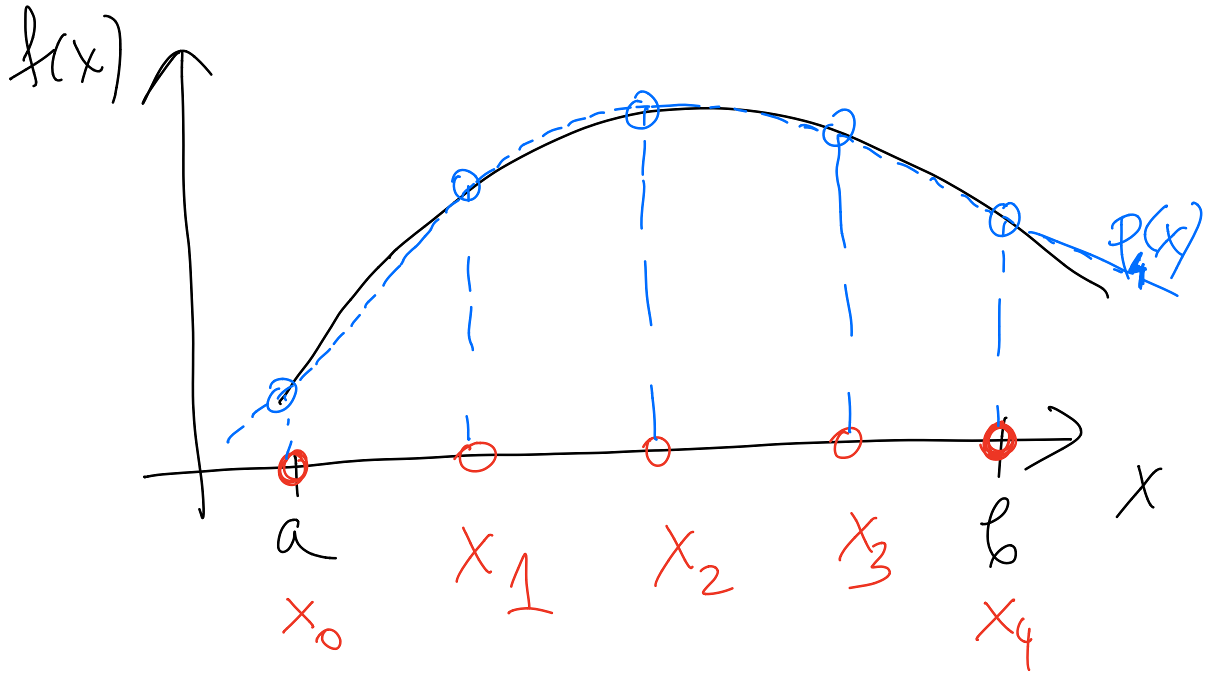


<u>Weierstrass approximation th</u>

$\forall \varepsilon > 0$, $\exists\, p(x)$ s.t.

$$\max_{a \le x \le b} |f(x) - p(x)| < \varepsilon$$

Interpolation



5 nodes

$(n+1)$ nodes

$P_n(x)$ passes through points

$(n+1)$ distinct points uniquely define a polynomial of degree n

Let's find $p_n(x)$

$$P_n = a_n \underbrace{x^n}_{\text{monomials}} + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

$$\vec{a} = [a_0, a_1, \dots, a_n]$$

$$\begin{cases} P_n(x_0) = f(x_0) = y_0 \\ P_n(x_1) = f(x_1) = y_1 \\ \dots \\ P_n(x_n) = f(x_n) = y_n \end{cases}$$

System of $n+1$ linear equations for $\vec{a}$

Interpolation

$$\sum_{k=0}^{n} a_k x_i^k = y_i$$

$i = 0, \dots, n+1$ nodes

$$\overleftrightarrow{V} \vec{a} = \vec{y}$$

Square linear system

(Fitting V was not square)

$$V = \underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}}_{\text{Vandermonde matrix}} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix}$$

$$\boxed{\overleftrightarrow{V}\,\vec{a} = \vec{y}}$$ LU factorization

Is V invertible?

$$\det(V) = \prod_{j<k} (x_k - x_j) \neq 0$$

If nodes are distinct then V is invertible

Is p_n unique? Yes

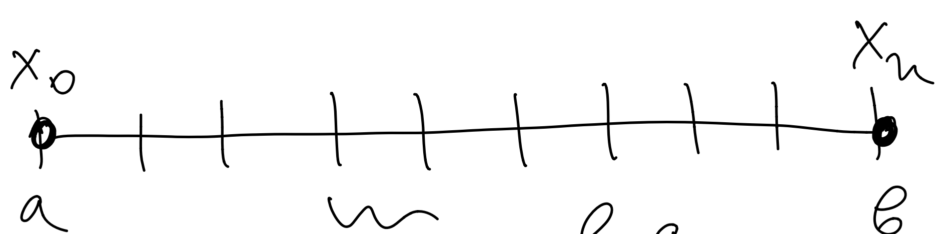
In Matlab you can do this using - polyfit

To evaluate $p(x)$ - polyval

$Va = y$

① If n is large, $O(n^3)$
cost is large

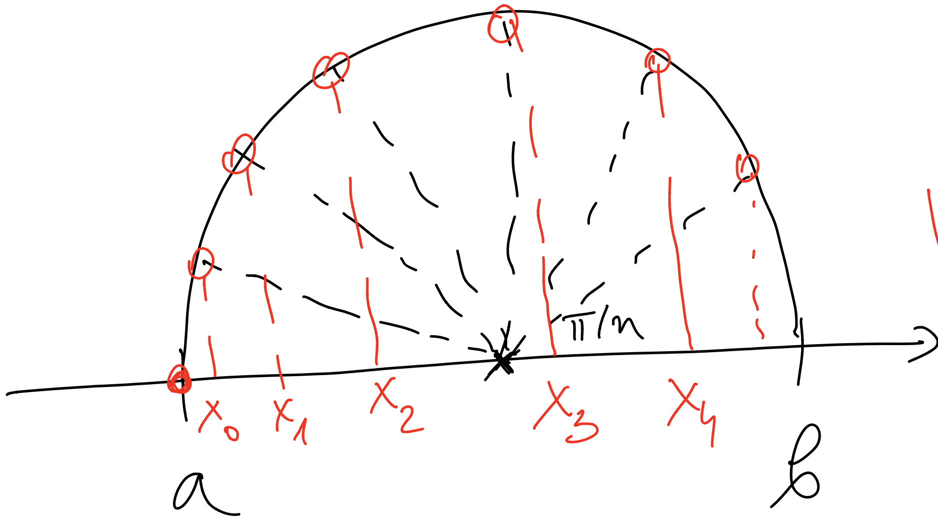
② Is V well-conditioned?
NO — ill-conditioned matrix
just like Hilbert



$$h = \frac{b-a}{n}$$

grid spacing

Equispaced grid $x_i = a + i \cdot h$



Abstract linear algebra

$\mathcal{P}_n$ – linear space of polynomials of degree n

$$\dim(\mathcal{P}_n) = n+1$$

# of linearly independent polynomials that form a basis

$$P_n = \text{span}\{b_0(x), b_1(x), \ldots, b_n(x)\}$$

$$\text{span}\{x^0, x^1, \ldots, x^n\}$$

If we use monomials as basis, then we get Vandermonde matrix

$$\vec{V} \longleftrightarrow \overleftrightarrow{V}'$$

monomials — other basis

$$\overleftrightarrow{V}' a' = y$$

$$V' = \text{Identity}$$

$$\Rightarrow a = y$$

$$P_n = \mathrm{span}\{L_0(x), L_1(x), \ldots, L_n(x)\}$$

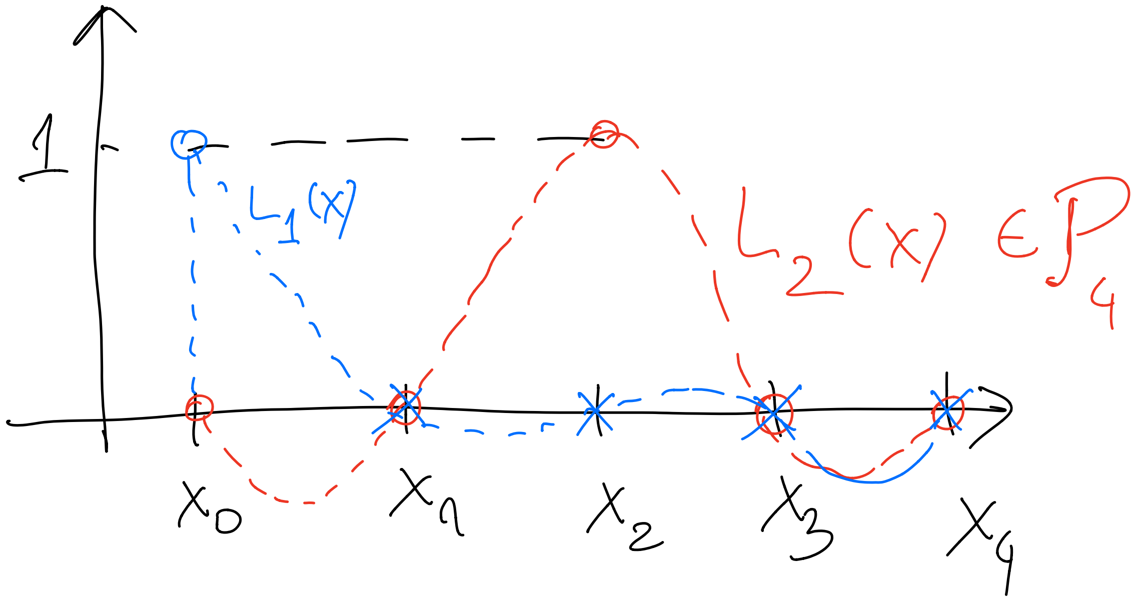
$$p(x) = \sum_{k=0}^{n} a_k L_k(x)$$

$$p(x_i) = \sum_{k=0}^{n} a_k L_k(x_i)$$

$$= \sum_{k=0}^{n} V_{ik}\, a_k$$

$$V_{ik} = \boxed{L_k(x_i) = \delta_{ik}}$$ best choice of L_k

$$\delta_{ik} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}$$ Kronecker δ symbol



$$a = y$$

$$\Rightarrow \quad p(x) = \sum_{k=0}^{n} y_k \underline{\underline{L_k(x)}}$$

Lagrange polynomials

$$L_k(x) = C(x-x_0)(x-x_1)\dots(x-x_n)$$

$$L_k(x) = C_k \prod_{k \neq j=0}^{n} (x - x_j)$$

$$L_k(x_k) = 1$$

$$C_k \prod_{j\neq k} (x_k - x_j) = 1$$

$$C_k = \frac{1}{\prod_{j\neq k} (x_k - x_j)}$$

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

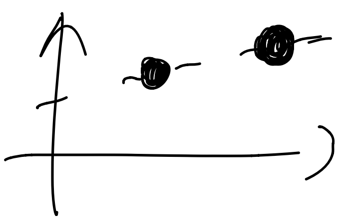
Lagrange interpolation formula

$$P(x) = \sum_{k=0}^{n} y_k \frac{\prod_{j\neq k} (x - x_j)}{\prod_{j\neq k} (x_k - x_j)}$$

Simple case: 2 points

$$p(x) = \underbrace{y_0 \cdot \frac{x - x_1}{x_0 - x_1}}_{k=0} \quad j=1$$

$$+ \underbrace{y_1 \frac{x - x_0}{x_1 - x_0}}_{k=1} \Big\} j=0$$



Are we done?

(1) How expensive is Lagrange interpolation

Evaluating $p(x)$ $O(n^2)$ evaluation

no cost to form $p(x)$

(2) Can I evaluate $p(x)$ without loosing digits?

(3) How good of an approximation is $p(x)$ to $f(x)$? math

Answers:

(1) $p_n = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$

$= a_0 + x(a_1 + \underbrace{a_2 x + \dots + a_n x^{n-1}})$

$+ a_0 + x(a_1 + x(a_2 + a_3 x + \dots + a_n x^{n-2}))$

$$= \ldots x\left(a_{n-3} + x\left(a_{n-2} + x\left(a_{n-1} + a_n x\right)\right)\right)$$

$$\left.\begin{cases} b_{n-1} = a_{n-1} + a_n x \\ b_{n-2} = a_{n-2} + b_{n-1} x \\ \vdots \\ b_0 = a_0 + b_1 x = p(x) \end{cases}\right\} 2n \text{ FLOPS}$$

$O(n)$ method — Horner's scheme

$$\boxed{\text{Evaluating Lagrange polynomial interpolant costs } O(n^2) \text{ FLOPS}}$$

Barycentric formula

$$w_k = \prod_{j \neq k} \frac{1}{x_k - x_j}$$ weights $k=0,\dots,n$

(pre-computation)

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

As long as $x \neq x_j$

this looses no digits

Accuracy of polynomial interpolation

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

$p_n(x) \equiv p(x)$

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

We need a norm

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

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

Is error small as $n \to \infty$.

$$P_{Taylor} = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2} f''(x_0)(x-x_0)^2 + \dots (x-x_0)^n$$

$$x_0 \in [a,b]$$

$$f(x) - P_{Taylor}(x) = \frac{f^{n+1}(\xi)}{(n+1)!}(x-x_0)^{n+1}$$

$$\xi \in [a,b] \quad , \xi(x)$$

Theorem

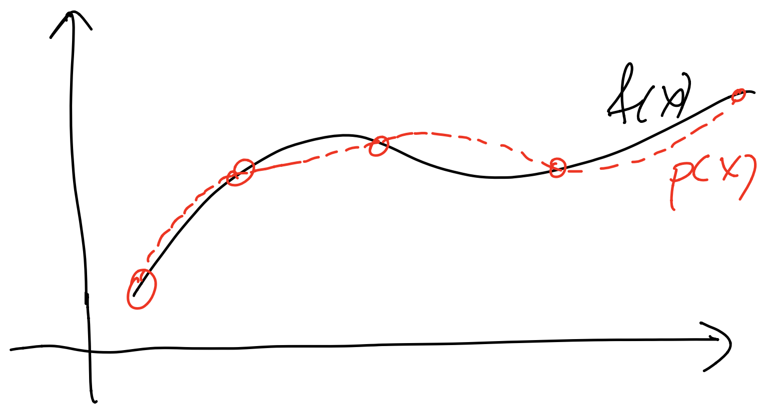
$$f(x) - p(x) = \underbrace{\frac{f^{n+1}(\xi)}{(n+1)!}}_{\text{How smooth is the function \#1}} \underbrace{\prod_{k=0}^{n}(x-x_k)}_{q(x)\ \#2 \text{ Nodal polynomial}}$$

Interpolating polynomial

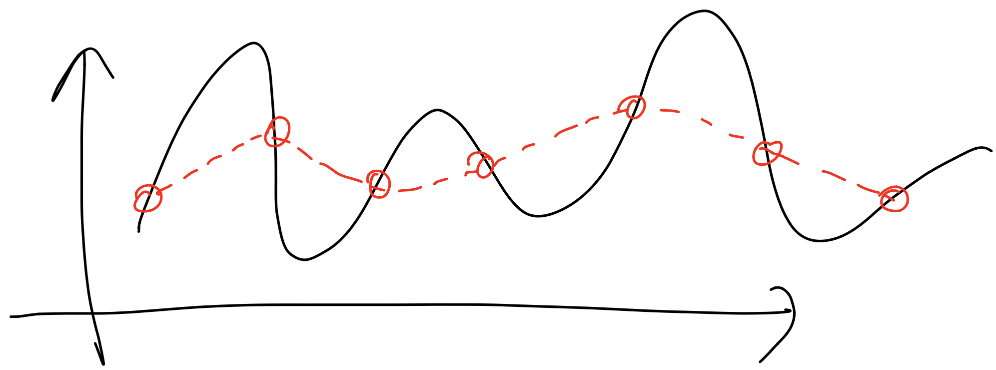
nodes

$$\xi \in [a,b]$$

#1: $\max\limits_{a \le x \le b} \left| f^{(n+1)}(x) \right|$



Smooth function is good!



#2 $\max\limits_{a \le x \le b} \left| q(x) \right|$ ← Lebesgue constant (Wiki)

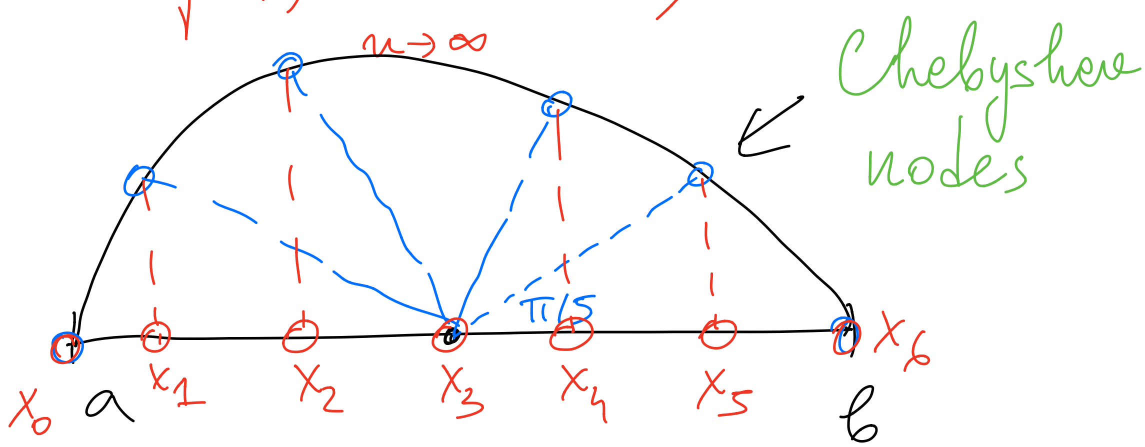
$$q(x) = \prod_{k=0}^{n} (x - x_k)$$

Depends on choice of nodes

Polynomial interpolants $p(x)$ do <u>NOT</u> converge to $f(x)$ for all smooth functions if we use equi-spaced nodes.

If $|q(x)|$ does not blow up near the end points, <u>then</u>

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



$$x_k = \cos\left(\frac{\pi}{n}\cdot k\right) \quad \text{for } [-1,1]$$

$$x_k = \frac{1}{2}(a+b) + \left(\frac{b-a}{2}\right)\cos\left(\frac{\pi k}{n}\right) \quad [a,b]$$