

Preconditioned CG algorithm

No initial guess given, preconditioner M

$$r_0 = Ax_0 - b$$

Solve $My_0 = r_0$

$$p_0 = -y_0$$

for $k=0, 1, \dots$

$$\alpha_k = \frac{r_k^T y_k}{p_k^T A p_k}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k + \alpha_k A p_k \quad \leftarrow \text{check convergence}$$

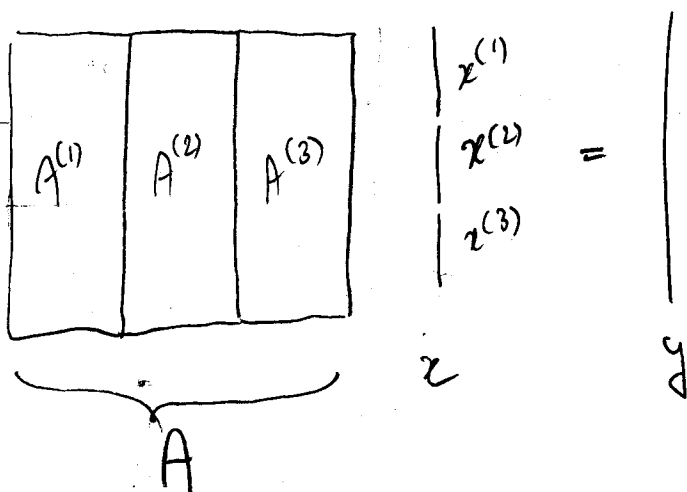
$$My_{k+1} = r_{k+1} \quad (\text{solve})$$

$$\beta_{k+1} = \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$$

$$p_{k+1} = -y_{k+1} + \beta_{k+1} p_k$$

Conjugate gradient is straightforward to parallelize if we have a parallel algo for computing matvec $y = Ax$.

One simple way is to give processor a chunk of vector x .



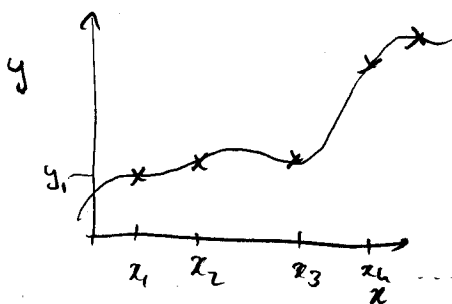
not perfect because each processor computes

$$y^{(i)} = A^{(i)} x^{(i)}$$

a lot of storage.

when A comes from disc of PDE it is easy to parallelize using "domain decomposition"

§ 8.1 Discrete least squares



polynomial interpolation does not always correspond well with the data.

Maybe here data can be better represented by a line. $y = ax + b$ even if this line does not go through all points.

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_x \approx \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_b$$

there are different ways of measuring closeness:

$$\min_x \|Ax - b\|_\infty$$

mini max problem

$$\min_x \|Ax - b\|_1$$

absolute deviation

$$\min_x \|Ax - b\|_2^2$$

least-squares

what norm we use depends on application and contrary to what your book says solving problems with $\|\cdot\|_\infty$ and $\|\cdot\|_1$ is feasible and useful (compressed sensing) but requires more knowledge of optimization and is not easy to "generalize" to infinite dimensions.
 $\rightarrow$ look at least-squares only.

We assume $A = \begin{matrix} m \\ n \end{matrix}$ $n > m$ overdetermined problem with more equations than unknowns. (underdetermined problem requires an additional technicality)

(157)

we also assume A has full column rank i.e. $\text{rank}(A) = \dim(\text{Range}(A)) = m$

Then to get least squares solution:

$$q(x) = \|Ax - b\|_2^2 = (Ax - b)^T (Ax - b) = \text{quadratic} \\ = x^T A^T A x - 2b^T A x + b^T b$$

$$\nabla q(x) = 2A^T A x - A^T b$$

$$\nabla^2 q(x) = 2A^T A \text{ s.p.d because of our assumptions}$$

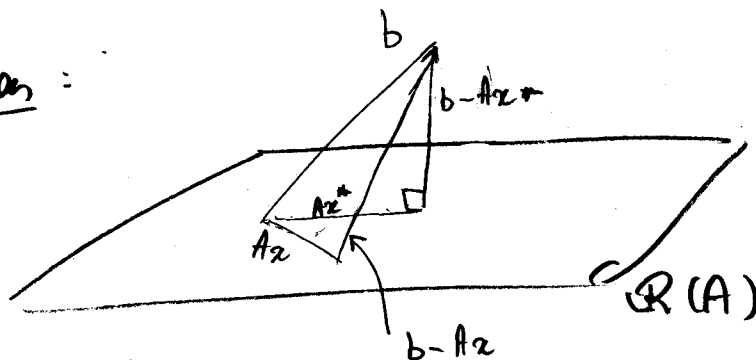
indeed: $x^T A^T A x = \|Ax\|^2 \geq 0$

and $x^T A^T A x = 0 \Rightarrow Ax = 0 \Rightarrow x = 0$
 $\uparrow$
 A full rank

Thus it suffices to find x_* s.t. $\nabla q(x_*) = 0$ i.e. solving:

$$\boxed{A^T A x = A^T b} = \text{normal eq}$$

Interpretation:



but x_* must have residuals to $R(A)$ can you read this from normal eq?

One can use least squares for fitting functions that are more complicated than lines! For example:

(158)

$$y = be^{ax}$$
$$\ln y = a \ln x + \ln b$$

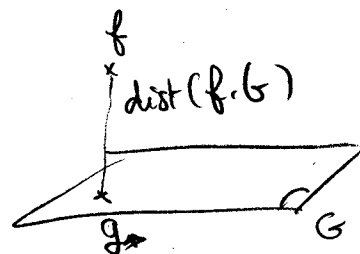
§ 8.2 Least Square approximation

General problem of best approximation:

Let E be a normed linear space

G be a subspace of E

$$\text{dist}(f, G) = \inf_{g \in G} \|f - g\|$$



The best approx of f in G is g s.t. $\|f - g\| = \text{dist}(f, G)$

One example of this problem:

$$E = C[a, b], \quad \|f\|_{\infty} = \max_{x \in [a, b]} |f(x)|$$

$$G = \Pi_m = \text{poly of degree } \leq m$$

Question: does approx exist?

Theorem: If G is a finite-dimensional subspace in a NLS E then $\forall f \in E \exists g \in G$ s.t. $\|f - g\| = \text{dist}(f, G)$.

We will confine ourselves to Normed Linear spaces where the norm comes from an inner product. This makes life easier.

Recall inner product = symm pos bilinear form

$$\left\{ \begin{array}{l} (f, g) = (g, f) \\ (\alpha f + \beta g, h) = \alpha (f, h) + \beta (g, h) \\ (f, f) > 0 \text{ if } f \neq 0 \\ \|f\| = \sqrt{(f, f)} \end{array} \right.$$

examples of inner products spaces:

$$\mathbb{R}^n \quad (x, y) = \sum_{i=1}^n x_i y_i$$

$$C_w[a, b] \quad (x, y) = \int_a^b w(x) f(x) g(x) dx.$$

where $w(x) \geq 0$ = weight function.

Inner products give notion of angle so:

$$f \perp g \Leftrightarrow (f, g) = 0$$

$$f \perp G \Leftrightarrow \forall g \in G \quad (f, g) = 0.$$

Here are some more properties of inner products that are handy and can be derived from axioms:

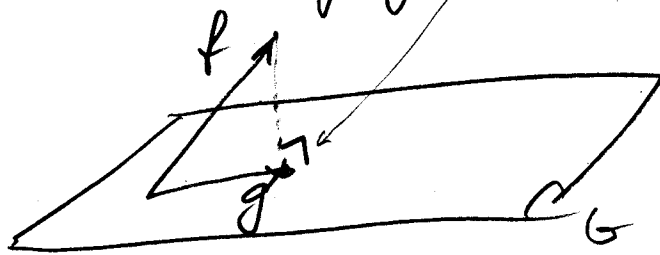
- $\|f + g\|^2 = \|f\|^2 + 2(f, g) + \|g\|^2$
- $\|f + g\|^2 = \|f\|^2 + \|g\|^2$, if $f \perp g$ (Pythagorean law)
- $|(f, g)| \leq \|f\| \|g\|$ Cauchy Schwarz inequality

- $\|f+g\|^2 + \|f-g\|^2 = 2\|f\|^2 + 2\|g\|^2.$

(163)

Theorem Let G be a subspace of an inner prod space E .

g is best approx to f in $G \iff f-g \perp G$



proof: $\Leftarrow$ assume $f-g \perp G$. Let $h \in G$:

$$(*) \quad \|f-h\|^2 = \underbrace{\|f-g\|^2}_{\perp G} + \underbrace{\|g-h\|^2}_{\in G} = \|f-g\|^2 + \|g-h\|^2 \geq \|f-g\|^2$$

$\Rightarrow$ Let g be best approx of f in G . Let $h \in G$ and $\lambda > 0$:

$$0 \leq \|f-g+\lambda h\|^2 - \|f-g\|^2$$

$$= \cancel{\|f-g\|^2} + 2\lambda (f-g, h) + \lambda^2 \|h\|^2 - \cancel{\|f-g\|^2}$$

$$= \lambda (2(f-g, h) + \lambda \|h\|^2)$$

$$\left. \begin{array}{l} \text{Letting } \lambda \rightarrow 0: \quad (f-g, h) \geq 0 \\ \text{Because } h \text{ is arbitrary in } G \text{ we must also have:} \\ \quad (f-g, -h) \geq 0 \end{array} \right\} \Rightarrow \begin{array}{l} (f-g, h) = 0 \\ \Rightarrow f-g \perp G \end{array}$$

Notice $(*)$ implies uniqueness. Since if g, h are both best approx,

$$\begin{aligned} \|f-h\|^2 &= \|f-g+g-h\|^2 = \|f-g\|^2 + \|g-h\|^2 = \|f-g\|^2 \\ &\Rightarrow \|g-h\|=0 \Rightarrow g=h \end{aligned}$$

An example with polynomials (see also example 8.2.1)

Find the best approx to $f(x) = \sin x$ by a polynomial

$$g(x) = c_1 x + c_2 x^3 + c_3 x^5 \text{ on } [-1, 1]$$

with norm:

$$\|f\| = \left(\int_{-1}^1 |f(x)|^2 dx \right)^{1/2}$$

Here $G = \text{span} \{x, x^3, x^5\}$, since x, x^3, x^5 are all lin indep.
we can enforce orthogonality of residual on each basis function.

$$(g - f, g_i) = 0, \quad i = 1, 2, 3, \quad \begin{aligned} g_1(x) &= x \\ g_2(x) &= x^3 \\ g_3(x) &= x^5 \end{aligned}$$

$$(g, g_i) = (f, g_i), \quad i = 1, 2, 3$$

$$\left(\sum_{j=1}^3 c_j g_j, g_i \right) = (f, g_i), \quad i = 1, 2, 3$$

$$\sum_{j=1}^3 G(g_j, g_i) = (f, g_i)$$

$$G c = F$$

$$c = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

$$F = \begin{pmatrix} (f, g_1) \\ (f, g_2) \\ (f, g_3) \end{pmatrix} = \begin{bmatrix} 8 \sin(1) - \cos(1) \\ -3 \sin(1) + 5 \cos(1) \\ 65 \sin(1) - 101 \cos(1) \end{bmatrix}$$

$$G = \begin{bmatrix} (g_1, g_1) & (g_1, g_2) & (g_1, g_3) \\ (g_2, g_1) & (g_2, g_2) & (g_2, g_3) \\ (g_3, g_1) & (g_3, g_2) & (g_3, g_3) \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{5} & \frac{1}{7} \\ \frac{1}{5} & \frac{1}{7} & \frac{1}{9} \\ \frac{1}{7} & \frac{1}{9} & \frac{1}{11} \end{bmatrix}$$

↑
in this case

Concl is: $c_1 \approx -0.99998$
 $c_2 \approx -0.16651$
 $c_3 \approx 0.00802$

However this is not a good idea as systems tend to be ill-conditioned (i.e. large condition number, so very sensitive to errors)

→ Better to use orthogonal systems why?

If we have an orthonormal basis of functions of G then:

$$G = \left[(g_i, g_j) \right]_{i,j=1}^n = I = \text{nothing is easier to solve than this!}$$

Gram matrix

What if we do not have an orthonormal basis?

Gram Schmidt = Given $\{v_1, v_2, \dots, v_n\}$ basis of U
then an orthonormal basis $\{u_1, u_2, \dots, u_n\}$ for U can be obtained by G.S. orthogonalization:

$$\begin{aligned} u_1 &= v_1 / \|v_1\| \\ \text{for } i=2, \dots, n \\ \left| \begin{aligned} u_i &= v_i - \sum_{j=1}^{i-1} (v_i, u_j) u_j \\ u_i &= u_i / \|u_i\| \end{aligned} \right. \end{aligned}$$

This can be shown by induction.

$u_1 = v_1 / \|v_1\|$ is an orthonormal basis of $\text{span}\{v_1\}$

assume

$\{u_1, \dots, u_i\}$ is an orthonormal basis of $\text{span}\{v_1, v_2, \dots, v_i\}$

then $u_{i+1} = v_{i+1} - \sum_{j=1}^i (v_{i+1}, u_j) u_j \neq 0$
(otherwise $v_{i+1} \in \text{span}\{v_1, \dots, v_i\}$)

By second step in loop we have $\|u_{i+1}\| = 1$
By induction hypothesis $\{u_1, \dots, u_i\}$ is orthonormal.
Now we only need to show orthogonality:

$$\begin{aligned}(u_{i+1}, u_k) &= (v_{i+1} - \sum_{j=1}^i (v_{i+1}, u_j) u_j, u_k), \quad k = 1, \dots, i \\&= (v_{i+1}, u_k) - \sum_{j=1}^i (v_{i+1}, u_j) (u_j, u_k) \\&= 0\end{aligned}$$

We saw Gram-Schmidt when we were looking at Gaussian quadratures. Specifically we used Legendre polynomials and we saw that for polynomials up to degree n , the Gram-Schmidt procedure simplifies a lot. (in fact only requirement is: $(fg, h) = (f, gh)$)

$$\begin{aligned}p_0 &= 1 \\p_1 &= x p_0 - \frac{(x p_0, p_0)}{(p_0, p_0)} p_0 \\p_2 &= x p_1 - \frac{(x p_1, p_1)}{(p_1, p_1)} p_1 - \frac{(x p_1, p_0)}{(p_0, p_0)} p_0 \\p_3 &= x p_2 - \frac{(x p_2, p_2)}{(p_2, p_2)} p_2 - \frac{(x p_2, p_1)}{(p_1, p_1)} p_1 - \frac{(x p_2, p_0)}{(p_0, p_0)} p_0 \\&\vdots\end{aligned}$$

$$p_n = x p_{n-1} - \frac{(x p_{n-1}, p_{n-1})}{(p_{n-1}, p_{n-1})} p_{n-1} - \frac{(x p_{n-1}, p_{n-2})}{(p_{n-2}, p_{n-2})} p_{n-2}$$

Legendre poly are poly $\perp$ w.r.t inner product:
 $\int_{-1}^1 f(x) g(x) dx$

Another important family of orthogonal polynomials is
Chebyshev polynomials which are $\perp$ w.r.t. inner prod.

(64)

$$(f, g) = \int_{-1}^1 f(x)g(x) \frac{dx}{\sqrt{1-x^2}}$$

It is easier to just check orthogonality than deriving it!

$$T_n(x) = \cos(n \cos^{-1} x)$$

$$(T_n, T_m) = \int_{-1}^1 \cos(n \cos^{-1} x) \cos(m \cos^{-1} x) \frac{dx}{\sqrt{1-x^2}}$$

$$\left. \begin{aligned} \text{Let } x &= \cos \theta \\ dx &= -\sin \theta d\theta \\ &= -\sqrt{1-x^2} d\theta \end{aligned} \right\} \Rightarrow (T_n, T_m) = \int_0^\pi \cos(n\theta) \cos(m\theta) d\theta = 0 \text{ if } n \neq m$$

A nice property of polynomials that we build w/ Gram Schmidt is
 that p_n is the polynomial of degree n with leading term 1

(i.e. monic: $p_n(x) = x^n + a_{n-1}x^{n-1} + \dots$)

such that $\|p_n\|$ is minimum

$$p_n = \operatorname{argmin}_{\substack{p \in \Pi_n \\ p \text{ monic}}} \|p\|$$

Proof = Since p_n is monic of degree $\leq n$ any monic poly of degree $\leq n$ can be written as:

$$q = p_n - \sum_{i=0}^{n-1} c_i p_i$$

$\|q - 0\|$ is smallest when $q \perp \text{span} \{p_0, \dots, p_{n-1}\} = \Pi_{n-1}$

but we know we can get this if $c_i = 0$, since $p_n \perp \Pi_{n-1}$.

Chebyshev Polynomials and interpolation

Chebyshev polynomials of the first kind can be defined recursively:

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad n \geq 1$$

Here are some of them:

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

Theorem: For x in $[-1, 1]$ we have: $T_n(x) = \cos(n \cos^{-1} x)$

proof: comes from cos addition formulae:

$$\cos(m+1)\theta = \cos\theta \cos m\theta - \sin\theta \sin m\theta$$

$$\cos(m-1)\theta = \cos\theta \cos m\theta + \sin\theta \sin m\theta$$

$$\Rightarrow \cos(m+1)\theta = 2\cos\theta \cos m\theta - \cos(m-1)\theta$$

let $\theta = \cos^{-1} x \Rightarrow$ functions $f_n(x) = \cos(n \cos^{-1} x)$ obey recurrence rel for Chebyshev poly.

Thus:

$$|T_n(x)| \leq 1 \quad -1 \leq x \leq 1$$

$$T_n\left(\cos \frac{j\pi}{n}\right) = (-1)^j \quad 0 \leq j \leq n \quad n \text{ sign changes}$$

$$T_n\left(\cos\left(\frac{2j-1}{2n}\pi\right)\right) = 0 \quad 1 \leq j \leq n \quad n \text{ zeros}$$

Monic polynomials: polynomial where coeff of highest degree is 1.

Because of recurrence formula:

$$T_n(x) = 2^{n-1}x^n + \dots$$

$\Rightarrow 2^{1-n}T_n(x)$ is a monic polynomial. (for $n > 1$)

Theorem on monic polynomials

If p is a monic polynomial of degree n then:

$$\|p\|_{\infty} = \max_{-1 \leq x \leq 1} |p(x)| \geq 2^{1-n}$$

proof: For contradiction suppose $|p(x)| < 2^{1-n}$ for $|x| \leq 1$.

let $q = 2^{1-n}T_n$ and $x_i = \cos(i\pi/n)$. q is monic and of degree n .

Then:

$$(-1)^i p(x_i) \leq |p(x_i)| < 2^{1-n} = (-1)^i \underbrace{q(x_i)}_{=(-1)^i}$$

$$\Rightarrow (-1)^i (q(x_i) - p(x_i)) > 0 \quad \text{for } 0 \leq i \leq n$$

$\Rightarrow$ polynomial $q(x) - p(x)$ changes sign n times on $[-1, 1]$

$\Rightarrow$ it must have at least n roots on $(-1, 1)$

$\Rightarrow$ contradiction since $\underset{\substack{\uparrow \\ \text{monic}}}{q(x)} - \underset{\substack{\uparrow \\ \text{monic}}}{p(x)}$ is of degree $n-1$

Now recall error estimate for interpolation on $[-1, 1]$ at some nodes x_i (168)

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i), \text{ where } \xi \in [-1, 1]$$

$$\Rightarrow |f(x) - p_n(x)| \leq \frac{1}{(n+1)!} |f^{(n+1)}(\xi)| \left| \prod_{i=0}^n (x - x_i) \right|$$

$$\leq \frac{1}{(n+1)!} \max_{|\xi| \leq 1} |f^{(n+1)}(\xi)| \left| \prod_{i=0}^n (x - x_i) \right|$$

$$\Rightarrow \max_{|x| \leq 1} |f(x) - p_n(x)| \leq \frac{1}{(n+1)!} \max_{|\xi| \leq 1} |f^{(n+1)}(\xi)| \max_{|x| \leq 1} \left| \prod_{i=0}^n (x - x_i) \right|$$

now we know from our results on Chebyshev poly that,

$$\max_{|x| \leq 1} \underbrace{\left| \prod_{i=0}^n (x - x_i) \right|}_{\text{monic poly}} \geq 2^{-n}$$

$\in \Pi_{n+1}$

and minimum is attained when $\prod_{i=0}^n (x - x_i) = \overbrace{2^{-n} T_{n+1}(x)}^{\text{monic}},$

i.e. when interp nodes are:

$$x_i = \cos\left(\frac{2i+1}{2n+2} \pi\right) \quad 0 \leq i \leq n$$

Theorem on interpolation error with Chebyshev interpolation nodes

If the nodes x_i are roots of T_{n+1} , then interp error formula gives:

$$|f(x) - p_n(x)| \leq \frac{1}{2^n (n+1)!} \max_{|t| \leq 1} |f^{(n+1)}(t)|$$

Here is another example of the use of Chebyshev polynomials.

(169)

Let $P_n(x)$ be a polynomial of degree n :

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

Consider the following problem: find polynomial $P_{n-1} \in \Pi_{n-1}$

so that $\max_{|x| \leq 1} |P_n(x) - P_{n-1}(x)|$ is as small as possible.

$$\text{let } q(x) = \frac{P_n(x) - P_{n-1}(x)}{a_n} = \text{some poly} \in \Pi_n$$

$$\Rightarrow \max_{|x| \leq 1} |q(x)| \geq \frac{1}{2^{n-1}}$$

with equality happening when $q(x) = 2^{1-n} T_n(x)$

$\Rightarrow$ we should choose:

$$\frac{P_n(x) - P_{n-1}(x)}{a_n} = 2^{1-n} T_n(x)$$

$$P_{n-1} = P_n(x) - 2^{1-n} a_n T_n(x)$$

and with this choice:

$$\max_{|x| \leq 1} |P_n(x) - P_{n-1}(x)| = \frac{|a_n|}{2^{n-1}}$$

of course this procedure can be repeated. However errors are introduced at each step, so we may want to stop when $\frac{|a_n|}{2^{n-1}}$ is larger than some prescribed tolerance

Trigonometric interpolation

(170)

Idea: Instead of using monomials to interpolate a function, use trigonometric functions (especially when data are want to interpolate is periodic.)

Recall Fourier series: Consider the inner product $(f, g) = \int_{-\pi}^{\pi} f(x) g(x) dx$

The functions $1, \cos x, \cos 2x, \cos 3x, \dots$
 $\sin x, \sin 2x, \sin 3x, \dots$

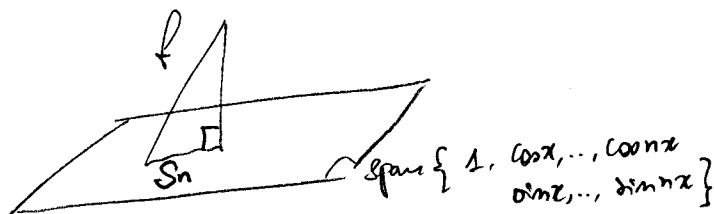
are orthogonal w.r.t. to inner product. This is because:

$$\sin nx \sin mx = \frac{1}{2} [\cos(n-m)x - \cos(n+m)x]$$

$$\cos nx \cos mx = \frac{1}{2} [\cos(n-m)x + \cos(n+m)x]$$

$$\sin nx \cos mx = \frac{1}{2} [\sin(n-m)x + \sin(n+m)x]$$

So we can find best approximation in subspace $\text{span}\{1, \cos x, \cos 2x, \dots, \cos nx, \sin x, \sin 2x, \dots, \sin nx\}$.



by simply taking: $S_n(x) = a_0 + \sum_{k=1}^n a_k \cos kx + b_k \sin kx$

$$a_0 = \frac{(f, 1)}{(1, 1)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx$$

$$a_k = \frac{(f, \cos kx)}{(\cos kx, \cos kx)} = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx dx$$

$$b_k = \frac{(f, \sin kx)}{(\sin kx, \sin kx)} = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx dx$$

If $f(x)$ is continuous: $f(x) = \lim_{n \rightarrow \infty} S_n(x) = a_0 + \sum_{k=1}^{\infty} a_k \cos kx + b_k \sin kx$

The theory for Fourier series is more elegant if we use complex exponentials:

$$e^{i\theta} = \cos\theta + i\sin\theta \quad \text{Euler's formula, } i^2 = -1$$

The Fourier series of $f(x)$ is given by:

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx} = \sum_{k=-\infty}^{\infty} c_k (e^{ix})^k \quad \leftarrow \text{trigonometric "polynomial"}$$

$$\text{where } c_k = \frac{(f, e^{ikx})}{(e^{ikx}, e^{ikx})} = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx, \quad k \in \mathbb{Z}$$

$$\text{and inner product is: } (f, g) = \int_{-\pi}^{\pi} f(x) \bar{g}(x) dx$$

$$g = a + ib, \quad \bar{g} = a - ib \quad (\text{complex conjugate})$$

By Euler's formula:

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \underbrace{(\cos kx - i \sin kx)}_{e^{-ikx}} dx$$

$$= \frac{1}{2} (a_k - ib_k) \quad k \geq 1$$

$$c_{-k} = \frac{1}{2} (a_k + ib_k)$$

$$c_0 = a_0$$

So the two formulations are equivalent since:

$$a_k = c_k + c_{-k}$$

$$b_k = \frac{c_{-k} - c_k}{i}$$

now let us look at what happens in the discrete case.

let us slightly modify inner product so that $E_k(x) = e^{ikx}$ are orthonormal. i.e. (172)

$$(E_m, E_n) = \delta_{mn} = \begin{cases} 1 & \text{if } m=n \\ 0 & \text{if } m \neq n \end{cases}$$

$$(f, g) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \overline{g(x)} dx$$

we added this 2π factor:

assume $m \neq n$.

$$\begin{aligned} (E_m, E_n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} E_m(x) \overline{E_n(x)} dx \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(m-n)x} dx = \frac{1}{2\pi} \left. \frac{e^{i(m-n)x}}{i(m-n)} \right|_{x=-\pi}^{\pi} = 0 \end{aligned}$$

We now introduce a pseudo-inner product:

$$(f, g)_N = \frac{1}{N} \sum_{j=0}^{N-1} f\left(\frac{2\pi j}{N}\right) \overline{g\left(\frac{2\pi j}{N}\right)}$$

let us check inner product axioms:

• $(f, f)_N \geq 0$, however $(f, f)_N = 0 \Rightarrow f\left(\frac{2\pi j}{N}\right) = 0 \nRightarrow f = 0$.

• $(f, g)_N = \overline{(g, f)_N}$

• $(\alpha f + \beta g, h)_N = \alpha (f, h)_N + \beta (g, h)_N$

pseudonorm: $\|f\|_N = \sqrt{(f, f)_N}$

For interpolation we will need the following result:

(17)

Theorem on pseudo-inner product.

$$(E_m, E_n)_N = \begin{cases} 1 & \text{if } m-n \text{ is divisible by } N \\ 0 & \text{otherwise.} \end{cases} \quad (N \geq 1)$$

Proof:

$$(E_m, E_n)_N = \frac{1}{N} \sum_{j=0}^{N-1} e^{\frac{i2\pi jm}{N}} e^{-\frac{i2\pi jn}{N}}$$

$$= \frac{1}{N} \sum_{j=0}^{N-1} \left[e^{\frac{i2\pi}{N}(m-n)} \right]^j$$

$$= \frac{1}{N} \frac{1 - \left[e^{\frac{i2\pi}{N}(m-n)} \right]^N}{1 - e^{\frac{i2\pi}{N}(m-n)}}$$

if $m-n$ is not divisible by N

$$= 0$$

when $m-n$ is divisible by N we do have: $(E_m, E_n)_N = 1$ since all terms in the sum are equal to one in this situation.

Interpolation with exponential polynomials

$$P(x) = \sum_{k=0}^n c_k e^{ikx} = \sum_{k=0}^n c_k (e^{iz})^k = \text{polynomials on unit circle.}$$

Theorem: The exponential polynomial that interpolates f at equally spaced nodes $x_j = \frac{2\pi j}{N}$, $j=0, \dots, N-1$ is:

$$P(x) = \sum_{k=0}^{N-1} c_k e^{ikx}, \text{ where } c_k = (f, E_k)_N$$

proof: We only need to verify that

(174)

$$P(x_l) = f(x_l), \quad l = 0, \dots, N-1$$

$$P(x_l) = \sum_{k=0}^{N-1} c_k e^{ikx_l} = \sum_{k=0}^{N-1} (f, E_k)_N e^{ikx_l}$$

$$= \sum_{k=0}^{N-1} \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) e^{-ikx_j} e^{ikx_l}$$

$$= \sum_{j=0}^{N-1} f(x_j) \underbrace{\frac{1}{N} \sum_{k=0}^{N-1} e^{ik(x_l - x_j)}}_{(E_l, E_j)_N = \delta_{lj}}$$

$$(E_l, E_j)_N = \delta_{lj}$$

$$= f(x_l).$$