Least squares approximation   ve smyslu nejm. čtverců

$B = C[a,b]$, A ... set of approx. functions

→ measure the distance by a norm induced by an inner product

Examples:
- $w(x)$ ... the weight function , $w(x) > 0$ [$w \geq 0$ $\int w > 0$ stačí]

$$(f,g) \overset{def}{=} \int_a^b f(x)\, g(x)\, w(x)\, dx$$

$$\| f \| = (f,f)^{1/2} = \sqrt{\int_a^b f^2(x)\, w(x)\, dx} \, .$$

or

- $B = \mathbb{R}^m$ $\min \| f - p \|_w^2 = \ldots$

$$(f,g) = \sum_{i=1}^{m} f_i g_i w_i \, , \quad w_i > 0 \, , \; i = 1, \ldots, m$$

$$= f^T W g \, , \qquad f = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}, \; g = \begin{bmatrix} g_1 \\ \vdots \\ g_m \end{bmatrix}$$

$$\| f \| = \sqrt{f^T W f} \qquad W = \mathrm{diag}(w_1, \ldots, w_m). \quad ■$$

In general, B ... a Hilbert space

$f \in B$, look for the best appr. to f in A

A ... finite dim. subspace

- We have the existence
- Uniqueness : either prove that $\| \cdot \|$ is strictly convex or it follows from the following construction

How to compute?

Theorem [P. [NS]] Let A be a linear subspace of a Hilbert space $\mathcal{H}$, let $f \in \mathcal{H}$.
$p^* \in A$ is the best approximation from A to f ⟺

$$(f - p^*, p) = 0 \quad \forall p \in A \, , \text{ i.e., } f - p^* \perp A .$$

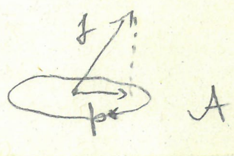
Proof. (⇒) if $(f - p^*, p) \neq 0$ for some $p \in A$

$$\Rightarrow \| (f - p^*) - \lambda p \|^2 < \| f - p^* \|^2 \quad \text{for some } \lambda \neq 0 .$$

(⇐) $f - p^* \perp A \Rightarrow \forall q \in A$:

$$\| f - q \|^2 = \| f - p^* + \underbrace{p^* - q}_{\in A} \|^2 = \| f - p^* \|^2 + \| p^* - q \|^2 \geq \| f - p^* \|^2 . \; ■$$

Remark.
- $p^* \neq q \Rightarrow \| f - q \|^2 > \| f - p^* \|^2$ → uniqueness
- $q = 0 \Rightarrow \| f \|^2 = \| f - p^* \|^2 + \| p^* \|^2$



Method of calculation

- Choose a basis of $\mathcal{A}$, $\{\phi_j : j = 0, 1, \dots, n\}$
- $p^* = \sum_{j=0}^{n} c_j \phi_j$
- It holds that

$$\left(f - \sum_{j=0}^{n} c_j \phi_j, \phi_i\right) = 0, \quad i = 0, 1, \dots, n.$$

$$\underbrace{\begin{bmatrix} a_{ij} = (\phi_i, \phi_j) \end{bmatrix}}_{A} \begin{bmatrix} c_0 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} (\phi_0, f) \\ \vdots \\ (\phi_n, f) \end{bmatrix}$$

↳ Gram matrix, symmetric & pos. definite (since ϕ_i lin. independent)

$$x^T A x = \sum_{i=0}^{n} \sum_{j=0}^{n} x_i x_j (\phi_i, \phi_j)$$
$$= \left(\sum_{i=0}^{n} x_i \phi_i, \sum_{j=0}^{n} x_j \phi_j\right) = \left\| \sum_{i=0}^{n} x_i \phi_i \right\|^2 > 0 \quad \forall x \neq 0$$

Orthogonal basis

A is diagonal $\Rightarrow$ $\boxed{p_n^* = \sum_{j=0}^{n} \frac{(\phi_j, f)}{\|\phi_j\|^2} \phi_j}$

$n \to \infty$, stop if $\|f - p_n^*\| < \delta$

↙ can be computed

$$\|f - p_n^*\|^2 = \|f\|^2 - \|p_n^*\|^2$$
$$= \|f\|^2 - \sum_{j=0}^{n} \frac{(\phi_j, f)^2}{\|\phi_j\|^2}$$

↗ increase n, add one term.

Examples:

- ϕ_j ... orthogonal polynomials (in general)
- In particular → Chebyshev polynomials T_i

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

R_n ... linear operator

$$f \in C[a,b] \longmapsto R_n f = \sum_{i=0}^{n} a_i T_i, \quad a_i = \frac{2}{\pi} \langle f, T_i \rangle, \quad a_0 = \frac{1}{\pi} \langle f, T_0 \rangle$$

Chebyshev projection

↘ can be computed using FFT

It can be shown that

$$\|R_n\|_\infty = \frac{1}{\pi} \int_0^{\pi} \left| \frac{\sin((n+\frac{1}{2})\theta)}{\sin(\frac{\theta}{2})} \right| d\theta.$$

Approximation to periodic functions

- periodic f. → occur naturally in many problems
- WLG → period 2π, $f(x+2\pi) = f(x)$, $-\infty < x < \infty$. $f \in C_{2\pi}$
- A ... lin. subspace of trigonometric polynomials

$$q(x) = \frac{1}{2} a_0 + \sum_{j=1}^{m} \left[a_j \cos(jx) + b_j \sin(jx) \right]$$

dim $A = 2m+1$ → $\boxed{\mathcal{T}_m}$

- $f \in C_{2\pi}$ can be approximated arbitrarily close by a $q \in \mathcal{T}_m$:

<u>Theorem</u> $\forall f \in C_{2\pi}$ $\forall \varepsilon > 0$ $\exists q \in \mathcal{T}_m$: $\| f - q \|_\infty < \varepsilon$,

where m is a finite integer.

<u>Proof.</u> A consequence of Jackson's theorems III (later). ☒.

↓

look for the best approximation $q^* \in \mathcal{T}_m$:

Consider L_2 norm →

$$\| f - q^* \|_2 = \min_{q \in \mathcal{T}_m} \| f - q \|_2 = \left[\int_{-\pi}^{\pi} (f(x) - q(x))^2 \, dx \right]^{1/2}$$

- $1, \cos(x), \sin(x), \dots, \cos(mx), \sin(mx)$ form an orthogonal basis of $\mathcal{T}_m$ → ϕ_j → use the previous results with respect to L_2 inner product
- lin. operator S_m: $f \in C_{2\pi} \longmapsto S_m f \in \mathcal{T}_m$ → use $\frac{(\phi_j, f)}{\|\phi_j\|^2}$ coef.

$$S_m f = \frac{1}{2} a_0 + \sum_{j=1}^{m} \left[a_j \cos(jx) + b_j \sin(jx) \right]$$

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(\theta) \cos(j\theta) \, d\theta, \quad b_i = \frac{1}{\pi} \int_{-\pi}^{\pi} f(\theta) \sin(j\theta) \, dx$$

$j = 0, 1, \dots, m$ $\qquad$ $i = 1, 2, \dots, m$

→ S_m is projection

- It can be shown that

$$(S_m f)(x) = \frac{1}{\pi} \int_{-\pi}^{\pi} \underbrace{\frac{\sin[(m+\frac{1}{2})\theta]}{2 \sin(\frac{\theta}{2})}}_{\text{Dirichlet kernel}} f(x+\theta) \, d\theta$$

and

$$\| S_m \|_\infty = \frac{1}{\pi} \int_{-\pi}^{\pi} \left| \frac{\sin[(m+\frac{1}{2})\theta]}{2 \sin(\frac{\theta}{2})} \right| d\theta = \| R_m \|_\infty .$$

→ $\ln m$ → poslijenormiranje konvergencija bitke "lepši" ipgidost

$f - S_m f \perp S_m f$

- we have an infinite orthogonal system of functions ϕ_j

$$\| f - S_m f \|^2 + \underbrace{\sum_{j=0}^{2m} \frac{(\phi_j, f)^2}{\|\phi_j\|^2}}_{\|S_m f\|^2} = \| f \|_2^2$$

⇒

$$\|S_m f\|_2^2 = \frac{\pi}{2} a_0^2 + \pi \sum_{j=1}^{m} (a_j^2 + b_j^2) \le \|f\|^2 = \int_{-\pi}^{\pi} f^2(x)\,dx$$

↓ Bessel's inequality $\Rightarrow$ $a_j \to 0$, $b_j \to 0$ with a reasonable speed ⇓ $S_m f$ converges

$\Rightarrow$ Series converges

- ϕ_j .... orthog. system → dense in $C_{2\pi}$ $\Rightarrow$ is hold Parseval's identity (equality)

$$\lim_{m\to\infty} \|S_m f\|_2^2 = \|f\|_2^2, \text{ and, therefore}$$

↳ (related to best approx in L_2 norm)

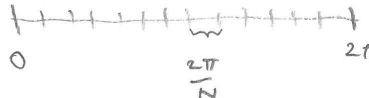
$$\|f - S_m f\|_2^2 \longrightarrow 0$$

(we have convergence in L_2 norm).

The discrete Fourier series operator

- practical realization → approximate the integrals by estimates (coef a_j, b_j)

$$\frac{1}{\pi}\int_{-\pi}^{\pi} g(\theta)\,d\theta = \frac{1}{\pi}\int_0^{2\pi} g(\theta)\,d\theta \approx \frac{1}{\pi}\sum_{k=0}^{N-1} g\left(\frac{2k\pi}{N}\right)\cdot\frac{2\pi}{N} = \frac{2}{N}\sum_{k=0}^{N-1} g\left(\frac{2k\pi}{N}\right)$$



so that

$$(*) \quad a_j \approx \frac{2}{N}\sum_{k=0}^{N-1} f\left(\frac{2k\pi}{N}\right)\cos\left(\frac{2k\pi j}{N}\right), \quad b_j \approx \frac{2}{N}\sum_{k=0}^{N-1} f\left(\frac{2k\pi}{N}\right)\sin\left(\frac{2k\pi j}{N}\right)$$

$j = 0, \dots, m$ $\qquad$ $j = 1, \dots, m$.

- work with vectors

$$\vec{f} = \begin{bmatrix} f(0) \\ f(\frac{2\pi}{N}) \\ \vdots \\ f(\frac{2\pi(N-1)}{N}) \end{bmatrix}, \quad \vec{c}_j = \begin{bmatrix} \cos(\frac{2\pi\cdot 0\cdot j}{N}) \\ \vdots \\ \cos(\frac{2\pi\cdot(N-1)\cdot j}{N}) \end{bmatrix}, \quad \vec{s}_j \begin{bmatrix} \sin(\frac{2\pi\cdot 0\cdot j}{N}) \\ \vdots \\ \sin(\frac{2\pi\cdot(N-1)\cdot j}{N}) \end{bmatrix}$$

i.e.

$$a_j \approx \frac{2}{N}\langle \vec{c}_j, \vec{f}\rangle, \quad b_j \approx \frac{2}{N}\langle \vec{s}_j, \vec{f}\rangle.$$

- It can be shown that $\vec{c}_0, \vec{c}_1, \dots, \vec{c}_m, \vec{s}_1, \dots, \vec{s}_m$, $m < \frac{N}{2}$, are orthogonal.

<u>Proof.</u> E.g. ($\cos\theta\cos\varphi = \frac{\cos(\theta-\varphi)+\cos(\theta+\varphi)}{2}$)

$$\langle \vec{c}_\ell, \vec{c}_j \rangle = \sum_{k=0}^{N-1} \cos\left(\frac{2\pi k j}{N}\right)\cos\left(\frac{2\pi k \ell}{N}\right) = \frac{1}{2}\sum_{k=0}^{N-1}\cos\left(\frac{2\pi k}{N}(j-\ell)\right) + \frac{1}{2}\sum_{k=0}^{N-1}\cos\left(\frac{2\pi k}{N}(j+\ell)\right)$$

$= 0$ if $j \ne \ell$

$= \frac{N}{2}$ if $j = \ell$ ∎

- integruji přesně $\sin(j\theta)$ a $\cos(j\theta)$ $j \ne mN$
- optimality of the approximation $(*)$ in the discrete least squares sense :

- Given points $\{ \frac{2\pi k}{N}, k = 0, \dots, N-1 \}$
  Find the best $q \in \mathcal{T}_m$: $\sum_{k=0}^{N} \left[f\left(\frac{2\pi k}{N}\right) - q\left(\frac{2\pi k}{N}\right) \right]^2$ is minimal.

↙ solve

$$\min \left\| \vec{f} - \left(\frac{\alpha_0}{2} \vec{c}_0 + \sum_{j=1}^{m} \alpha_j \vec{c}_j + \sum_{j=1}^{m} \beta_j \vec{s}_j \right) \right\|^2$$

if $m < \frac{N}{2} \Rightarrow \vec{c}_j, \vec{s}_j$ orthog.

↙

solve the linear LS problem $Ax \approx b \rightarrow A^TAx = A^Tb$ $\quad A = \underbrace{\left[\frac{\vec{c}_0}{2}, \vec{c}_1, \dots \vec{c}_m, \vec{s}_1, \dots \vec{s}_m \right]}_{\text{orthog}}$

$\Rightarrow$ optimal α_j and β_j are given by

$$\alpha_j = \frac{2}{N} \langle \vec{c}_j, f \rangle, \quad \beta_j = \frac{2}{N} \langle \vec{s}_j, f \rangle. \quad ▨$$

FAST FOURIER TRANSFORMATION

- Assumptions : N is power of 2 , $m < \frac{N}{2}$, $m \approx \frac{N}{2}$
  - N: number of discrete points in $[0, 2\pi)$
  - m: ↳ degree of trig. polynomial
  - $m \approx \frac{N}{2}$ → the most effective

- Goal : compute $\langle \vec{c}_j, \vec{f} \rangle \rightarrow \boxed{\alpha_j}$ and $\langle \vec{s}_j, f \rangle \rightarrow \boxed{\beta_j}$ $\rightarrow$ $O(N^2)$ operations in the standard way
  $j = 0, \dots, m$ $\qquad j = 1, \dots, m$

<u>Idea</u>

- A motivation : $\underbrace{\text{Consider}}_{2N\text{ op}}$ α_j and $\alpha_{\frac{N}{2}-j}$

$j \qquad \frac{N}{2}-j \qquad \frac{N}{2}$

$$\alpha_j = \frac{2}{N} \sum_{k=0}^{N-1} f\left(\frac{2\pi k}{N}\right) \cos\left(\frac{2\pi k}{N} \cdot j\right) = \underbrace{\sum_{k \text{ even}}}_{2N\text{op}} + \underbrace{\sum_{k \text{ odd}}}_{2N\text{op}}$$

$$\alpha_{\frac{N}{2}-j} = \frac{2}{N} \sum_{k=0}^{N-1} f\left(\frac{2\pi k}{N}\right) \cos\left(\frac{2\pi k}{N}\left(\frac{N}{2} - j\right)\right)$$

$$= \frac{2}{N} \sum_{k=0}^{N-1} f\left(\frac{2\pi k}{N}\right) (-1)^k \cos\left(\frac{2\pi k}{N} j\right) = \sum_{k \text{ even}} - \sum_{k \text{ odd}}$$

$\left.\begin{matrix} 4N \\ \text{versus} \\ 2N \end{matrix}\right\}$ op.

$\Rightarrow$ we can compute $\sum_{k \text{ even}}$ and $\sum_{k \text{ odd}}$ $\lessgtr \begin{matrix} \alpha_j \\ \alpha_{\frac{N}{2}-j} \end{matrix}$ $\rightarrow$ only N operations instead of $2N$

- A general framework

$$\begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_m \end{bmatrix} = \frac{2}{N} \begin{bmatrix} \langle \vec{c}_0, \vec{f} \rangle \\ \vdots \\ \langle \vec{c}_m, \vec{f} \rangle \end{bmatrix} = \frac{2}{N} \left[\cos\left(\frac{2\pi jk}{N}\right) \right] \vec{f}, \quad \begin{bmatrix} 0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} = \left[\sin\left(\frac{2\pi jk}{N}\right) \right] \vec{f}$$

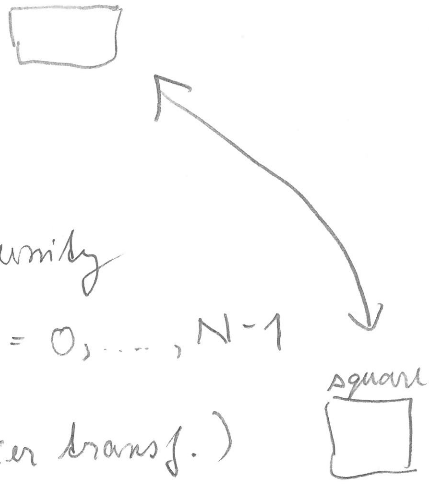
$j \rightarrow$; $j = 0, \dots, m$; $\uparrow k = 0, \dots, N-1$

$$\begin{bmatrix} \vec{c}_0^{\,T} \\ \vdots \\ \vec{c}_m^{\,T} \end{bmatrix} \qquad \begin{bmatrix} 0^T \\ \vec{s}_1^{\,T} \\ \vdots \\ \vec{s}_m^{\,T} \end{bmatrix}$$

Hence

$$\begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_m \end{bmatrix} - i \begin{bmatrix} 0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} = \begin{bmatrix} e^{-i\frac{2\pi jk}{N}} \end{bmatrix} \vec{f}$$

$\rightarrow$ $j = 0{:}m$, $\uparrow$ $k = 0, \dots, N-1$



Define $\boxed{\omega_N = e^{-\frac{2\pi i}{N}}}$ .... the Nth root of unity

DFT matrix F_N : $F_{j+1,k+1} = \omega_N^{jk}$, $j, k = 0, \dots, N-1$

Want to find $\boxed{\vec{y} = F_N \vec{f}}$ (discrete Fourier transf.)

It holds that $\frac{1}{N} F_N^* F_N = I \Rightarrow \frac{1}{\sqrt{N}} F_N$ is unitary

$\rightarrow$ inverse Fourier $\rightarrow$ multiply by F_N^*.

Example $N = 8$, $\omega \equiv \omega_8$

Put odd and even colums together

modulo 8

$$F_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \dots & & & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix} \rightarrow$$

$F_8(:, [1,3,5,7,2,4,6,8])$

$$\left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\ 1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\ 1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^3 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & \omega^2 & \omega^4 & \omega^6 & -\omega & -\omega^3 & -\omega^5 & -\omega^7 \\ 1 & \omega^4 & 1 & \omega^4 & -\omega^2 & -\omega^6 & -\omega^2 & -\omega^6 \\ 1 & \omega^6 & \omega^4 & \omega^2 & -\omega^3 & -\omega & -\omega^7 & -\omega^5 \end{array}\right]$$

use $\omega^2 = \omega_8^2 = \omega_4$

$$= \left[\begin{array}{c|c} F_4 & \Omega_4 F_4 \\ \hline F_4 & -\Omega_4 F_4 \end{array}\right]$$

where $\Omega_4 = \begin{bmatrix} 1 & & & \\ & \omega & & \\ & & \omega^2 & \\ & & & \omega^3 \end{bmatrix}$.

Denote cols $= [1,3,5,7,2,4,6,8]$. Then

$$y = F_8 \cdot x = F(:,\text{cols}) \cdot x(\text{cols}) = \left[\begin{array}{c|c} F_4 & \Omega_4 F_4 \\ \hline F_4 & -\Omega_4 F_4 \end{array}\right] \begin{bmatrix} x(1{:}2{:}8) \\ \hline x(2{:}2{:}8) \end{bmatrix}$$

$$= \left[\begin{array}{c|c} I_4 & \Omega_4 \\ \hline I_4 & -\Omega_4 \end{array}\right] \begin{bmatrix} F_4\, x(1{:}2{:}8) \\ \hline F_4\, x(2{:}2{:}8) \end{bmatrix}$$

Denote $y_T = \boxed{F_4\, x(1{:}2{:}8)}$
$y_B = \boxed{F_4\, x(2{:}2{:}8)}$.

It holds that
$$\boxed{\begin{aligned} y(1{:}4) &= y_T + d.*y_B \\ y(5{:}8) &= y_T - d.*y_B \end{aligned}}$$

$d = [1, \omega, \omega^2, \omega^3]^T$.

More generally, if $N = 2m$, then $y = F_N x$ is given by

$$y(1:m) = y_T + d.*y_B, \qquad d = [1, \omega_N, \dots, \omega_N^{m-1}]^T$$
$$y(m+1, N) = y_T - d.*y_B,$$

and
$$y_T = F_m\, x(1:2:N)$$
$$y_B = F_m\, x(2:2:N).$$

→ recur until $m = 1$, use $F_1 x = x$.

Algorithm → we have $x \in \mathbb{C}^N$, $N = 2^t$

```
function y = fft(x,n)
  if n=1
      y = x
  else
      m = n/2
      y_T = fft( x(1:2:n), m)
      y_B = fft( x(2:2:n), m)
      ω = exp(- 2πi/n)
      d = [1, ω, ..., ω^(m-1) ]^T
      z = d.* y_B
      y = [ y_T + z
            y_T - z ]
  end
```

→ $8\, N \log_2(N)$ operations

Fast sine and cosine transformations

p. 36

Discrete sin transform (DST)

Given real $x_1, \dots, x_{m-1}$, compute

$$y_k = \sum_{j=1}^{m-1} \sin\left(\frac{kj\pi}{m}\right) x_j, \qquad k = 1, \dots, m-1$$

Discrete cos transform (DCT)

Given real $x_0, x_1, \dots, x_m$, compute

$$y_k = \frac{x_0}{2} + \sum_{j=1}^{m-1} \cos\left(\frac{kj\pi}{m}\right) x_j + (-1)^k \frac{x_m}{2}.$$

→ they appear in F_{2m}

$$\left[F_{2m}\right]_{k+1, j+1} = \omega_{2m}^{kj} = \cos\left(\frac{kj\pi}{m}\right) - i \sin\left(\frac{kj\pi}{m}\right).$$

(8)

Denote

$$S = \left[\sin\frac{kj\pi}{m}\right]_{k,j=1,\dots,m-1} \qquad C = \left[\cos\frac{kj\pi}{m}\right]_{k,j=1,\dots,m-1}$$

$$e = \underbrace{[1,\dots,1]^T}_{m-1}, \qquad v = \underbrace{[-1,1,-1,1,\dots,(-1)^{m-1}]^T}_{m-1}$$

$$E = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix} \in \mathbb{R}^{m-1} \quad \dots \text{ exchange permutation}$$

Then [GoLo 2013, p. 37],

$$F_{2m} = \left[\begin{array}{c|c|c|c} 1 & e^T & 1 & e^T \\ \hline e & C-iS & v & (C+iS)E \\ \hline 1 & v^T & (-1)^m & v^TE \\ \hline e & E(C+iS) & Ev & E(C-iS)E \end{array}\right]$$

DST: $y = Sx$, $x \in \mathbb{R}^{m-1}$

set up $x_{sin} = \begin{bmatrix} 0 \\ x \\ \hline 0 \\ -Ex \end{bmatrix}$. Then $\frac{i}{2} F_{2m} x_{sin} = \begin{bmatrix} 0 \\ Sx \\ \hline 0 \\ -ESx \end{bmatrix}$

so that $y = i\,\tilde{y}(2:m)/2$,
where $\tilde{y} = F_{2m} x_{sin}$

DCT:

$$y = \left[\begin{array}{c|c|c} \frac{1}{2} & e^T & \frac{1}{2} \\ \frac{e}{2} & C & \frac{v}{2} \\ \hline \frac{1}{2} & v^T & \frac{(-1)^m}{2} \end{array}\right] \begin{bmatrix} x_0 \\ \tilde{x} \\ x_m \end{bmatrix} = \begin{bmatrix} \frac{1}{2}x_0 + e^T\tilde{x} + \frac{x_m}{2} \\ \frac{x_0}{2}e + C\tilde{x} + \frac{x_m}{2}v \\ \frac{x_0}{2} + v^T\tilde{x} + \frac{x_m}{2}(-1)^m \end{bmatrix} \Big\}\, m+1$$

$\tilde{x} \in \mathbb{R}^{m-1} = [x_1,\dots,x_{m-1}]$

set up

$$x_{cos} = \begin{bmatrix} x_0 \\ \tilde{x} \\ \hline x_m \\ E\tilde{x} \end{bmatrix} \in \mathbb{R}^{2m}. \text{ Then } \frac{1}{2} F_{2m} x_{cos} = \begin{bmatrix} y \\ \hline \frac{x_0}{2}e + EC\tilde{x} + \frac{x_m}{2}Ev \end{bmatrix} \Big\}\, m+1$$

so that $y = \tilde{y}(1:m+1)/2$
where $\tilde{y} = F_{2m} x_{cos}$

Björck 503
Golub - Van Loan 33
Strang 448 -> Fast Poiss. solver
Applic. of FFT
Higham 451 -> FP

Fast Fourier Transformation

Motivation :

- Cheb. projection $f_m(x) = \sum_{j=0}^{m} a_j T_j(x)$, $a_0 = \frac{1}{\pi} \int_{-1}^{1} f \frac{1}{\sqrt{1-x^2}} dx$, $a_j = \frac{2}{\pi} \int_{-1}^{1} f T_j \frac{1}{\sqrt{1-x^2}} dx$, $j \geq 1$,

  and $\int_{-1}^{1} f T_j(x) \frac{dx}{\sqrt{1-x^2}} = \boxed{\int_{0}^{\pi} f(\cos\theta) \cos(j\theta) \, d\theta} \overset{2\pi \text{ inl.}}{=} 2 \int_{-\pi}^{\pi} \dots \, d\theta$

  $x = \cos\theta$, $d\theta = -\frac{dx}{\sqrt{1-x^2}}$

  how to approx. efficiently ?   cos is symmetric

- Fourier series $f \in C_{2\pi}$, $f \approx q_m \in \mathcal{T}_m \to$ best approx in L^2 norm

$$q_m(x) = \frac{1}{2} a_0 + \sum_{j=1}^{m} [a_j \cos(jx) + b_j \sin(jx)]$$

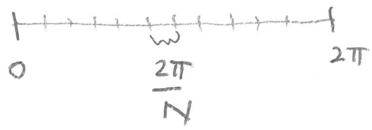
$$a_j = \boxed{\frac{1}{\pi} \int_{\pi}^{\pi} f(\theta) \cos(j\theta) \, d\theta} \qquad b_j = \frac{1}{\pi} \boxed{\int_{-\pi}^{\pi} f(\theta) \sin(j\theta) \, d\theta}$$

$j = 0, 1, \dots, m$ $j = 1, \dots, m$

- Approximate integrals by estimates (practical realisation)

$$\frac{1}{\pi} \int_{-\pi}^{\pi} g(\theta) \, d\theta = \frac{1}{\pi} \int_{0}^{2\pi} g(\theta) \, d\theta \approx \frac{1}{\pi} \sum_{k=0}^{N-1} g\left(\frac{2k\pi}{N}\right) \cdot \frac{2\pi}{N} = \frac{2}{N} \sum_{k=0}^{N-1} g\left(\frac{2k\pi}{N}\right),$$

N ... divisor par.



so that

$$a_j \approx \underbrace{\frac{2}{N} \sum_{k=0}^{N-1} f\left(\frac{2k\pi}{N}\right) \cos\left(\frac{2k\pi}{N} j\right)}_{\alpha_j}, \quad b_j \approx \underbrace{\frac{2}{N} \sum_{k=0}^{N-1} f\left(\frac{2k\pi}{N}\right) \sin\left(\frac{2k\pi}{N} j\right)}_{\beta_j}$$

$j = 0, \dots, m$ $j = 1, \dots, m$

- Work with vectors , $\alpha_j = \frac{2}{N} \langle \vec{c}_j, \vec{f} \rangle$, $\beta_j = \frac{2}{N} \langle \vec{s}_j, \vec{f} \rangle$

$$\vec{f} \equiv \begin{bmatrix} f(\frac{2\pi}{N} \cdot 0) \\ \vdots \\ f(\frac{2\pi}{N}(N-1)) \end{bmatrix}, \quad \vec{c}_j \equiv \begin{bmatrix} \cos(\frac{2\pi}{N} \cdot 0 \cdot j) \\ \vdots \\ \cos(\frac{2\pi}{N} \cdot (N-1) \cdot j) \end{bmatrix}, \quad \vec{s}_j \equiv \begin{bmatrix} \sin(\frac{2\pi}{N} \cdot 0 \cdot j) \\ \vdots \\ \sin(\frac{2\pi}{N} \cdot (N-1) \cdot j) \end{bmatrix}$$

- It can be shown that $\vec{c}_0, \vec{c}_1, \dots, \vec{c}_m, \vec{s}_1, \dots, \vec{s}_m$, $m < \frac{N}{2}$ are orthogonal.

- Assumptions : ... Page 5

Other applications

- Fast multiplication with circulant matrices, fast linear solver

$$C(a) = \begin{bmatrix} a_1 & a_5 & a_4 & a_3 & a_2 \\ a_2 & a_1 & a_5 & a_4 & a_3 \\ a_3 & a_2 & a_1 & a_5 & a_4 \\ a_4 & & \ddots & & \\ a_5 & & & & \end{bmatrix}, \quad a = \begin{bmatrix} a_1 \\ \vdots \\ a_5 \end{bmatrix}$$

It can be shown [GoLo 2013, p. 220] (book) that if $C(a) \in \mathbb{R}^{N \times N}$ is circulant, then

$$C(a) = F_N^{-1} \Lambda F_N, \quad \text{where } \Lambda = \operatorname{diag}(F_N a).$$

Then

$$C(a) \cdot v = \left(F_N^{-1}\left(\Lambda\left(F_N v\right)\right)\right).$$

(F_N^{-1} → ifft, Λ → diag, $F_N v$ → fft)

One can also solve the lin. system $C(a)\, x = b$

Fast multiplication with Toeplitz matrices (constant along diagonals)

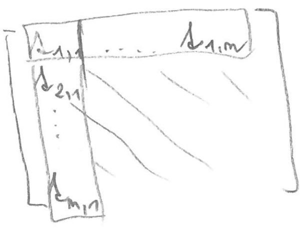
E.g.

$$T = \begin{bmatrix} 5 & 2 & 7 \\ 4 & 5 & 2 \\ 9 & 4 & 5 \end{bmatrix} \longrightarrow C = \left[\begin{array}{ccc|cc} 5 & 2 & 7 & 9 & 4 \\ 4 & 5 & 2 & 7 & 9 \\ 9 & 4 & 5 & 2 & 7 \\ \hline 7 & 9 & 4 & 5 & 2 \\ 2 & 7 & 9 & 4 & 5 \end{array}\right]$$

Toeplitz can be embedded in a circulant

In general, if $T \in \mathbb{R}^{m \times m}$, then $T = C(1{:}m, 1{:}m)$, where $C \in \mathbb{R}^{(2m-1)\times(2m-1)}$

$$C(:,1) = \begin{bmatrix} T(1{:}m, 1) \\ T(1, m{:}-1{:}2)^T \end{bmatrix}$$



$$C(:,1) = \begin{bmatrix} t_{1,1} \\ \vdots \\ t_{m,1} \\ t_{1,m} \\ t_{1,m-1} \\ \vdots \\ t_{1,2} \end{bmatrix} \qquad C = \begin{bmatrix} T & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

- Want to compute Tx

→ def $y = \begin{bmatrix} x \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{2m-1}$, then $Cy = \begin{bmatrix} Tx \\ C_{2,1} x \end{bmatrix}$.

```
function g = genT(v)
   m = length(v);
   g = fft([v; v(m:-1:2)]);
```

```
function y = mulT(x,g)
   m = length(x);
   w = fft([x; zeros(m-1,1)]);
   w = ifft(g.*w);
   y = w(1:m);
```

Example - Fast Poisson solver

[GoLo2013 : p 226]

- 2D Poisson, homogeneous boundary cond, finite differences

$$\mathbb{A} = \begin{bmatrix} T & -I & & \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & T \end{bmatrix} \qquad \begin{array}{l} I \in \mathbb{R}^{m\times m} \\ T \in \mathbb{R}^{m\times m}, \quad T = \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix} \\ m \text{ blocks} \end{array}$$

- Denote $B_m = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{m\times m}$

- Recall Kronecker product $A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,m}B \\ \vdots & & \\ a_{m,1}B & \dots & a_{m,m}B \end{bmatrix}$

- Then

$$\mathbb{A} = \begin{bmatrix} B_m & & \\ & \ddots & \\ & & B_m \end{bmatrix} + \begin{bmatrix} 2I & -I & & \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & 2I \end{bmatrix}$$

$$= (I_m \otimes B_m) + (B_m \otimes I_m)$$

- It is known that

$B_m = S_m^{-1} \Lambda_m S_m$, where S_m is the sin matrix
$S_m = \left[\sin\left(\frac{kj\pi}{m+1}\right)\right]_{k,j=1,\dots,m}$

Λ_m is diagonal
$\lambda_j^{(m)} = 2 - 2\cos\left(\frac{j\pi}{m+1}\right), \quad j = 1,\dots,m.$

- $\mathbb{A}x = c$, x … long vector
define $X = \text{reshape}(x, m, m)$
so that $x = \text{vec}(X)$
the vectorization

- Calculus
$FXG = H \iff (G^T \otimes F)\,\text{vec}(X) = \text{vec}(H)$, F, G nonsingular.

- Using this rule,

$$\mathbb{A}x = c \iff \left[(I_m \otimes B_m) + (B_m \otimes I_m)\right]\text{vec}(X) = \text{vec}(C)$$

$$\iff \boxed{B_m X + X B_m = C}$$

- Use spectral decomposition of B_m, B_m

$$S_m^{-1}\Lambda_m S_m X + X S_m^{-1} \Lambda_m S_m = C$$

$$\Lambda_m \underbrace{S_m X S_m^{-1}}_{\tilde{X}} + \underbrace{S_m X S_m^{-1}}_{\tilde{X}} \Lambda_m = \underbrace{S_m C S_m^{-1}}_{\tilde{C}}$$

- We get $\Lambda_m \tilde{X} + \tilde{X} \Lambda_m = \tilde{C}$

  (Λ_m: diag, Λ_m: diag) — compare entry by entry

$$\tilde{x}_{i,j} = \frac{\tilde{c}_{i,j}}{\lambda_i^{(m)} + \lambda_j^{(m)}} . \longrightarrow \tilde{X}$$

- compute $X = S_m^{-1} \tilde{X} S_m$ using FFT

$$x = vec(X) .$$

<u>Software</u>

FFTW, developed at MIT (Frigo, Johnson) (C)
Fastest Fourier Transform in the West

Wilkinson software Prize