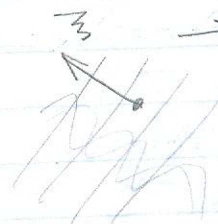
02/09. Fourier

Fourier analysis & PDE $\quad Lu = f$.

$$\Delta e^{2\pi i x\xi} = \left(-4\pi|\xi|^2\right) e^{2\pi i x \xi}$$

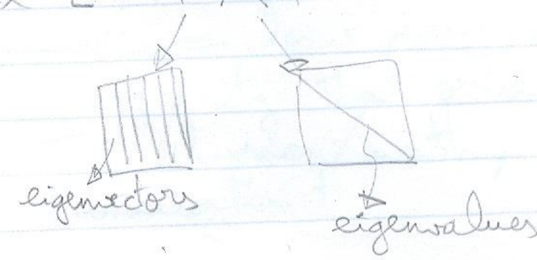


$\rightarrow$ generalized eigenfunctions

eigenvalue $-4\pi|\xi|^2$

$\rightarrow$ provide spectral decomposition of Δ in R^n.

Matrix $L = P \Lambda P^{-1}$



then $L^{-1} = P \Lambda^{-1} P^{-1}$

L symmetric: $P^{-1} = P^*$ adjoint

Back to Δ:

P^{-1} is FT: $\hat{f}(\xi) = \int_{R^n} e^{-2\pi i x\xi} f(x)\,dx$

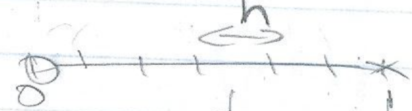
Λ^{-1} is division by $-4\pi|\xi|^2$

P is IFT: $f(x) = \int_{R^n} e^{2\pi i x \xi} \hat{f}(\xi)\,d\xi$

$$\Rightarrow \Delta^{-1} f(x) = \int_{\mathbb{R}^n} e^{2\pi i x \xi} \left(\frac{-1}{4\pi|\xi|^2} \right) \hat{f}(\xi)\, d\xi ,$$

inverts Δ w/ free space cond.

Discretization?



$x_j = jh$, $h = \frac{1}{N}$

$$\Delta_h u_j = \frac{-2u_j + u_{j-1} + u_{j+1}}{h^2} = f_j, \text{ error } O(h^2)$$

(i) periodic B.C.: $u_0 = u_N$ $\quad 0 \le j \le N-1$
(ii) Dirichlet B.C.: $u_0 = a$, $u_N = b$, $1 \le j \le N-1$
(iii) Neumann B.C.: $\frac{u_1 - u_{-1}}{2h} = a$, $\frac{u_{N+1} - u_{N-1}}{2h} = b$

with
$$\begin{cases} \frac{-2u_0 + u_1 + u_{-1}}{h^2} = f_0 \\ \frac{-2u_N + u_{N+1} + u_{N-1}}{h^2} = f_N \end{cases}$$

$\longrightarrow$ elim. u_{-1} and u_{N+1}

still, $0 \le j \le N$

All B.C. (a, b)
go in the rhs with f.

Matrices for Δ_h =

(i) $$\Delta_{h,per} = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \boxed{-1} \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ \boxed{-1} & & & -1 & 2 \end{bmatrix}$$ Toeplitz, circulant $\in \mathbb{R}^{N \times N}$

(ii) $$\Delta_{h,D} = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}$$ Toeplitz $\in \mathbb{R}^{N-1 \times N-1}$

(iii) $$\Delta_{h,N} = \frac{1}{h^2}\begin{bmatrix} 2 & -2 & & \\ -1 & 2 & -1 & \\ & -1 & \ddots & -1 \\ & & -2 & 2 \end{bmatrix}$$ $\in \mathbb{R}^{N+1 \times N+1}$

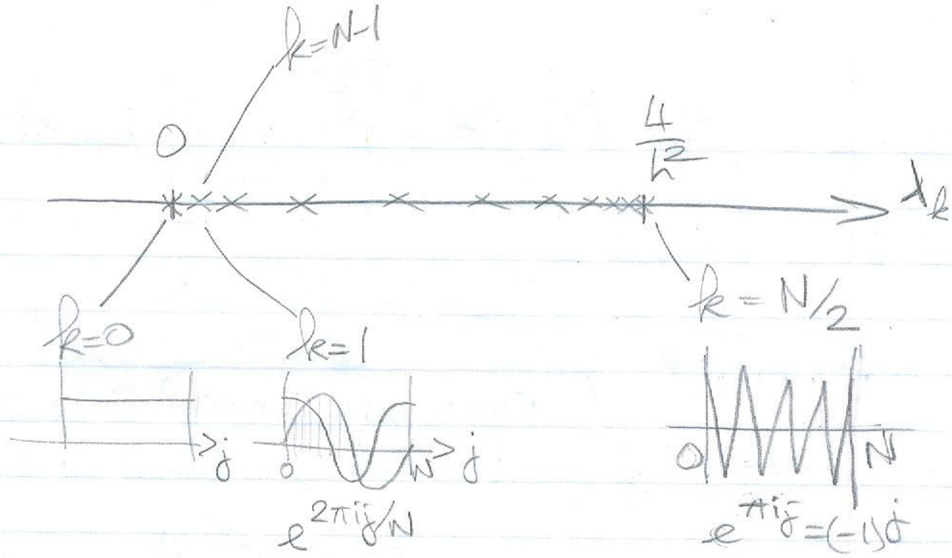
(i) $u_j = e^{2\pi i jk/N} = \omega^{jk}, \quad \omega = e^{2\pi i/N}$

$$-\omega^{(j-1)k} + 2\omega^{jk} - \omega^{(j+1)k}$$
$$= \left(-\omega^{-k} + 2 - \omega^{k}\right)\omega^{jk}$$
$$= \left(2 - \left(e^{2\pi i k/N} + e^{-2\pi i k/N}\right)\right)\omega^{jk}$$
$$= \left(2 - 2\cos\frac{2\pi k}{N}\right)\omega^{jk}$$

$\Rightarrow \omega^{jk}$ eigenvector of $\Delta_{h,per}$ with egval $\frac{1}{h^2}\left(2 - 2\cos\frac{2\pi k}{N}\right) = \lambda_k$

$k = 0, \dots, N-1$ freq. index

$k = N$: $\omega^{jN} = e^{2\pi i j N/N} = 1 = \omega^{j0}$



$$\lambda_{N-k} = \frac{1}{h^2}\left(2-2\cos\left(\frac{2\pi(N-k)}{N}\right)\right) = \lambda_k$$

$F_{jk} = \frac{1}{\sqrt{N}}\,\omega^{jk}$ is the IFFT

$\Delta_{h,per} = F\Lambda F^*$, $\lambda_k = \frac{1}{h^2}\left(2-2\cos\frac{2\pi k}{N}\right)$

$(F^*)_{jk} = \frac{1}{\sqrt{N}}\,\omega^{-jk}$ is the FFT

$\Delta^{-1}_{h,per} = F\Lambda^{-1}F^*$

No problem with λ_0^{-1}
provided rhs f has $\sum f_j = 0$
(compatibility condition)

$\Rightarrow$ fast $O(N\log N)$ operations.

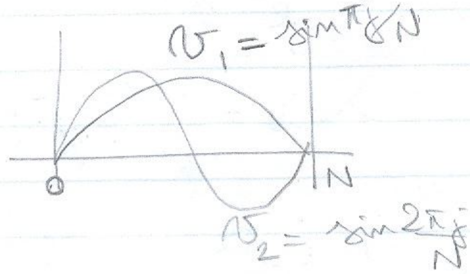
Multiplicity 2 eigenspaces: $e^{\pm 2\pi i jk/N}$
or $\cos\left(\frac{2\pi jk}{N}\right)$ & $\sin\left(\frac{2\pi jk}{N}\right)$

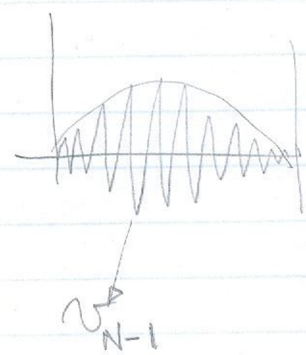
(ii) $\Delta_{h,D} = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & & \\ & & & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{N-1 \times N-1}$

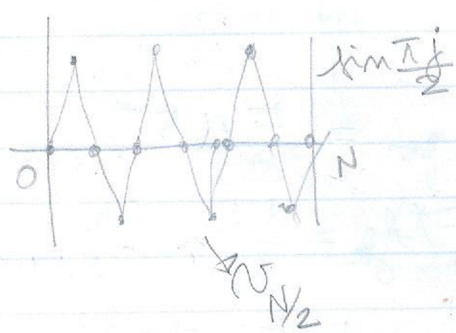
$1 \leq j \leq N-1$
$1 \leq k \leq N-1$

$v_k(j) = \sin \frac{\pi j k}{N}$

(sines are complete over a half period)







$\lambda_k = \frac{1}{h^2}\left(2 - 2\cos\frac{\pi k}{N}\right) \qquad 1 \leq k \leq N-1$

$\lambda_0 > 0, \qquad \lambda_{N-1} < \frac{4}{h^2}$

no nullspace!

$\Delta_{h,D} = S \Lambda S^*$

also DST-I

discrete sine transform, DST-I

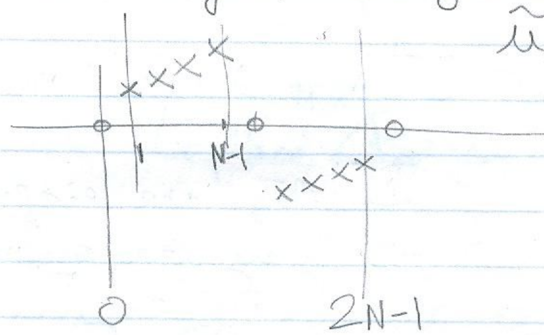
$S^{-1} = S^* = S$

$S_{jk} = c \sin\frac{\pi jk}{N} = (S^*)_{jk}$ $\qquad c = \sqrt{\frac{2}{N}}$

(so that $\sum_j |S_{jk}|^2 = 1$)

Fast DST-I: $O(N \log N)$ ops.

• Extend u_j, $1 \le j \le N-1$ to $\tilde{u}_j$, $0 \le j \le 2N-1$



$$= \begin{cases} 0 & j = 0 \\ u_j & 1 \le j \le N-1 \\ 0 & j = N \\ u_{2N-j} & N+1 \le j \le 2N-1 \end{cases}$$

• Apply length-2N FFT to $\tilde{u}_j$, get U_k, $0 \le k \le 2N-1$

• Restrict

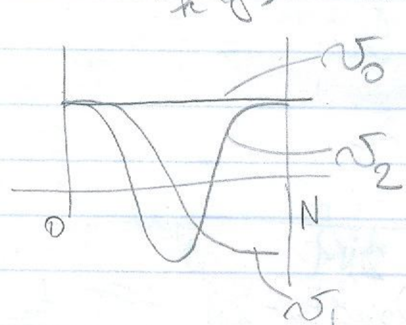
$$U_k = \tilde{U}_k / (-2i), \quad 1 \le k \le N-1$$

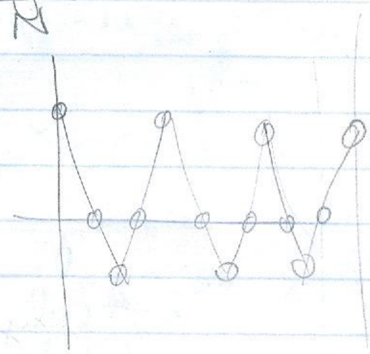
(real)

(iii) $\Delta_{h,N} = \frac{1}{h^2}\begin{bmatrix} 2 & -2 & & \\ -1 & \ddots & & -1 \\ & & -2 & 2 \end{bmatrix} \in R^{N+1 \times N+1}$

$0 \le j,k \le N$

$v_k(j) = \cos\frac{\pi jk}{N}$





$$\lambda_k = \frac{1}{h^2}\left(2 - 2\cos\frac{\pi k}{N}\right) \qquad 0 \le k \le N$$

⚠ $\lambda_0 = 0$, nullspace

$\Delta_{h,N} \neq \Delta^*_{h,N}$ so v_k are not orthogonal.

but $D^{-1}\Delta_{h,N} D$ is symm, with $D = \mathrm{diag}(\sqrt{2}, 1, \ldots, 1, \sqrt{2})$

$D^{-1} v_k$ are eigenvectors of $D^{-1}\Delta_{h,N} D$
$\Rightarrow$ they are orthogonal
(divide 1st & last component by $\sqrt{2}$)

$$\Delta_{h,N} = C \Lambda C^{-1}$$

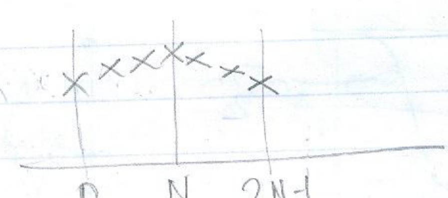
C: IDCT-I

C^{-1}: DCT-I

Fast $O(N \log N)$ algo:

- Extend u_j $\quad 0 \le j \le N$ to $\tilde{u}_j$ $\quad 0 \le j \le 2N-1$,

$$\tilde{u}_j = \begin{cases} u_j, & 0 \le j \le N \\ u_{2N-j}, & N+1 \le j \le 2N-1 \end{cases}$$



- Apply length-$2N$ FFT to $\tilde{u}_j \to \tilde{U}_k$
- Restrict $U_k = \tilde{U}_k / 2$ $\quad$, $0 \le k \le N$.
(real)

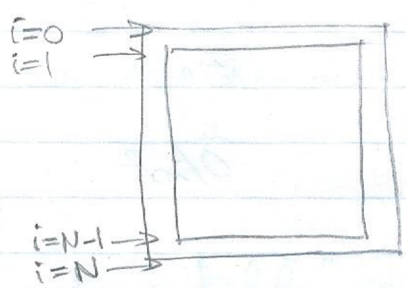
Next: 2D, + Helmholtz
spectral methods Nullspace, pseudo-inverse
other BC - Schur complements.
no worries about convergence.

Ref. Strang, SIAM Rev., 41-1, 135-147 (1999)
Persson's Laplace code.

02/14 2D Laplace / Poisson.

$$\begin{cases} -\Delta u = f, & (x,y) \in [0,1]^2 \\ u = g, & x = 0,1 \text{ or } y = 0,1 \end{cases}$$

$x_{ij} = (ih, jh)$, $h = 1/N$
$1 \le i,j \le N-1$



$$-\Delta_h u_{ij} = \frac{4u_{ij} - u_{i-1,j} - u_{i+1,j} - u_{i,j+1} - u_{i,j-1}}{h^2}$$

(error $O(h^2)$)

$u_{ij} = g_{ij}$ when $i = 0$, or N
or $j = 0$, or N

Typical row. $-1 \; \cdots \; -1 \;\; 4 \;\; -1 \; \cdots \; -1$

$$-\Delta_{h,D} = \frac{1}{h^2}\begin{bmatrix} T & -I & & \\ -I & \ddots & & \\ & & \ddots & -I \\ & & -I & T \end{bmatrix} \in \mathbb{R}^{(N-1)^2 \times (N-1)^2}$$

$$T = \begin{bmatrix} 4 & -1 & \\ -1 & \ddots & -1 \\ & -1 & 4 \end{bmatrix}, \quad I \in \mathbb{R}^{N-1 \times N-1}$$

$$K = \begin{bmatrix} 2 & -1 & \\ -1 & \ddots & -1 \\ & -1 & 2 \end{bmatrix} \in \mathbb{R}^{N-1 \times N-1}$$

$$-\Delta_{h,D} = \frac{1}{h^2}\begin{bmatrix} 2I & -I & \\ -I & \ddots & -I \\ & -I & 2I \end{bmatrix} + \frac{1}{h^2}\begin{bmatrix} K & & \\ & \ddots & \\ & & K \end{bmatrix}$$

$$= \underbrace{\frac{K}{h^2}}_{\partial^2/\partial x^2} \otimes I + I \otimes \underbrace{\frac{K}{h^2}}_{\partial^2/\partial y^2}$$

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & & \vdots \\ a_{m1}B & \cdots & a_{mm}B \end{bmatrix}$$: Kronecker product.

(Replicate B with pattern of A)

$$(A \otimes B)_{ij;mn} = A_{im} \otimes B_{jn}$$

$$\Rightarrow \left(-\Delta_{h,D}\right)_{ij;mn} = \frac{K_{im}\delta_{jn}}{h^2} + \frac{\delta_{im}K_{jn}}{h^2}$$

(Separability)

$$\frac{1}{h^2}K v_k = \lambda_k v_k \qquad v_k(j) = \sqrt{\frac{2}{N}} \sin \frac{\pi j k}{N}$$

$$\lambda_k = \frac{1}{h^2}\left(2 - 2\cos\frac{\pi k}{N}\right).$$

Claim: $(-\Delta_{h,D})(v_{k_1} \otimes v_{k_2})$

$$= (\lambda_{k_1} + \lambda_{k_2})(v_{k_1} \otimes v_{k_2})$$

Pf: $(-\Delta_{h,D})_{ij;mn} v_{k_1}(m) v_{k_2}(n)$

$$= \frac{1}{h^2}\left(K_{im}\delta_{jn} + K_{jn}\delta_{im}\right) v_{k_1}(m) v_{k_2}(n)$$

$$= \frac{1}{h^2}\left(\lambda_{k_1} v_{k_1}(i) v_{k_2}(j) + \lambda_{k_2} v_{k_2}(j) v_{k_1}(i)\right)$$

$$= \frac{1}{h^2}(\lambda_{k_1} + \lambda_{k_2}) v_{k_1}(i) v_{k_2}(j)$$

$\square$

Gives spectral dec. $-\Delta_{h,D} = P \Lambda P^{-1}$

$$(-\Delta_{h,D})^{-1} = P \Lambda^{-1} P^{-1}$$

(invertible OK because $\lambda_{k_1} > 0$, $\lambda_{k_2} > 0$ so $\lambda_{k_1} + \lambda_{k_2} > 0$)

DST-I algorithm for $-\Delta_{h,D} u = f$

1) Expand $\sum_{i,j} v_{k_1}(i) v_{k_2}(j) f_{ij} = \hat{f}_{k_1,k_2}$

2) Invert $(\lambda_{k_1} + \lambda_{k_2})^{-1} \hat{f}_{k_1,k_2} = \hat{u}_{k_1,k_2}$

3) Synthesize $u_{ij} = \sum_{k_1,k_2} v_{k_1}(i) v_{k_2}(j) \hat{u}_{k_1,k_2}$

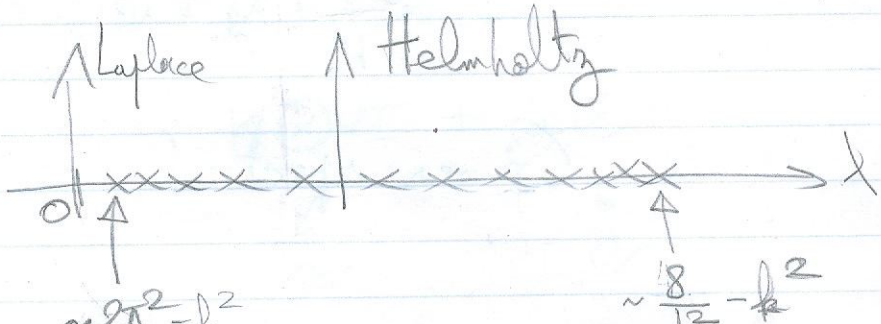
Next: Helmholtz: k fixed;

$$\begin{cases} -\Delta u - k^2 u = f & (x,y)\in[0,1]^2 \\ u = g & x=0,1 \text{ or } y=0,1 \end{cases}$$

$$(-\Delta_{h,D} - k^2 I)(v_{k_1}\otimes v_{k_2})$$
$$= (\lambda_{k_1}+\lambda_{k_2}-k^2)(v_{k_1}\otimes v_{k_2})$$

Make sure k^2 is not an eigenvalue, (or else f has no comp. along $v_{k_1}\otimes v_{k_2}$)

$\rightarrow$ 2) $\times(\lambda_{k_1}+\lambda_{k_2}-k^2)$



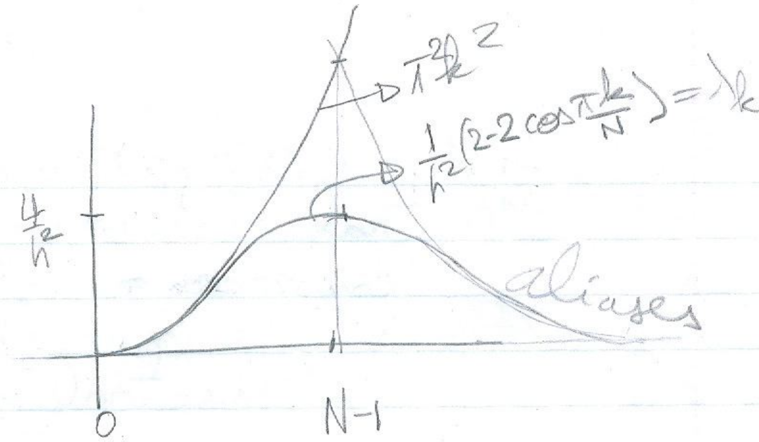
Rmk: Spectral method:

$$\lambda_k = \frac{1}{h^2}\left(2 - 2\cos\left(\pi \frac{k}{N}\right)\right)$$
$$\sim \frac{1}{h^2} \quad \frac{\pi^2 k^2}{N^2} = \pi^2 k^2.$$

Spectral: use $\lambda_k = \pi^2 k^2$

because $v_k(j) = \sin\left(\frac{j\pi k}{N}\right) \Rightarrow v_k(x) = \sin(\pi k x)$

1D:

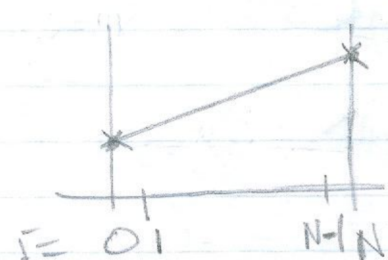


$$\lambda_k - \pi^2 k^2 = \frac{1}{h^2} O\left(\frac{k^4}{N^4}\right) = O(h^2 k^4)$$

(comparable to $\pi^2 k^2$ when $k \sim 1/h$)

→ Switch to spectral for Helmholtz. Differentiates $v_k(j)$ exactly.

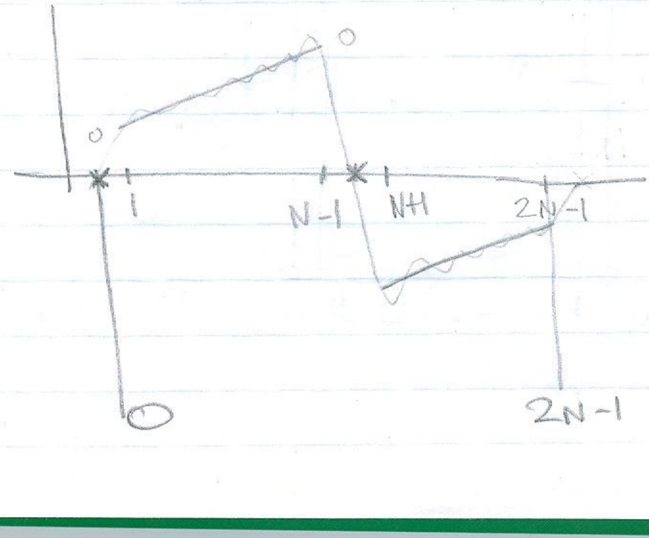
Rmk: Expansion in sines may converge slowly!



$$u_j = \sum \hat{u}_k \sin \frac{\pi j k}{N}$$

$$1 \le j \le N-1$$

↓ expand in FFT the mirror-extended

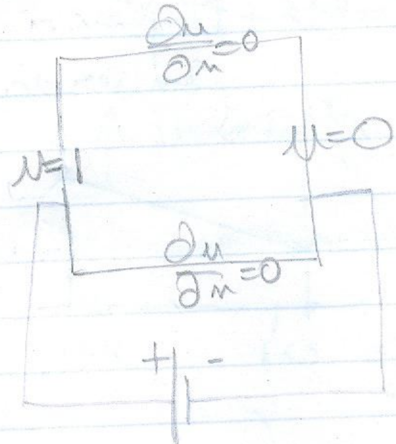


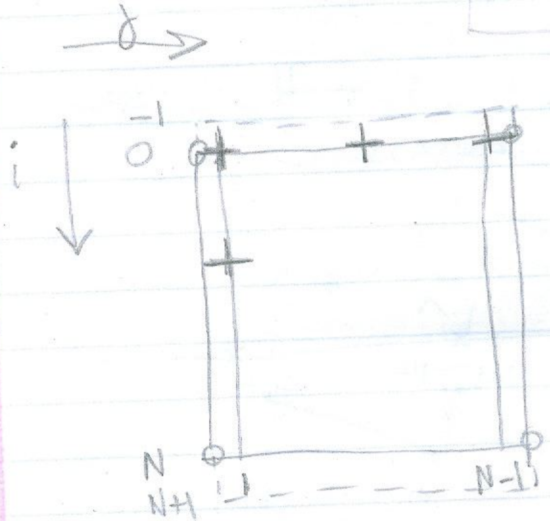
Spectral methods: because convergence is fast.

Aliasing: delocalized artifacts in spectral methods.

Next: Hybrid Dirichlet-Neumann problem:

$$\begin{cases} -\Delta u = f & (x,y) \in [0,1]^2 \\ u = g_1 & x = 0,1 \\ \dfrac{\partial u}{\partial y} = g_2 & y = 0,1 \end{cases}$$





$0 \leq i \leq N$

$1 \leq j \leq N-1$

y-derivatives (i): $\Delta^{1D}_{h,N} = \frac{1}{h^2}\begin{bmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -2 & 2 \end{bmatrix}$

x-derivatives (j): $\Delta^{1D}_{h,D} = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & & \\ & & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}$

$$\Delta_{h,DN} = \underbrace{\Delta^{1D}_{h,D}}_{N-1 \times N-1} \otimes \underbrace{I}_{N+1 \times N+1} + \underbrace{I}_{N-1 \times N-1} \otimes \underbrace{\Delta^{1D}_{h,N}}_{N+1 \times N+1}$$

Hybrid DST-DCT algo:

1) Expand: DST along x
   DCT along y
2) Invert:

$$\lambda^{-1}_{k_1 k_2} = (\lambda_{k_1} + \lambda_{k_2})^{-1}$$

3) Synthesize: DST along x
   IDCT along y.

Rmk: Full Neumann: similar

Do not divide by $\lambda_0 = 0$:
Pseudo-inverse.

$$\Delta^{+}_{h,N} = P \Lambda^{+} P^{-1}$$

$\hookrightarrow \mathrm{diag}\left(0, \frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots \frac{1}{\lambda_N}\right)$

Rmk. One-sided difference

$$\left.\frac{\partial u}{\partial x}\right|_0 = \frac{3u_0 - 4u_1 + u_2}{2h}$$

Elim with $\dfrac{-u_0 + 2u_1 - u_2}{h^2} = (f, \text{tang})$

$$\frac{\frac{2}{3}u_1 - \frac{2}{3}u_2}{h^2} = (\dots)$$

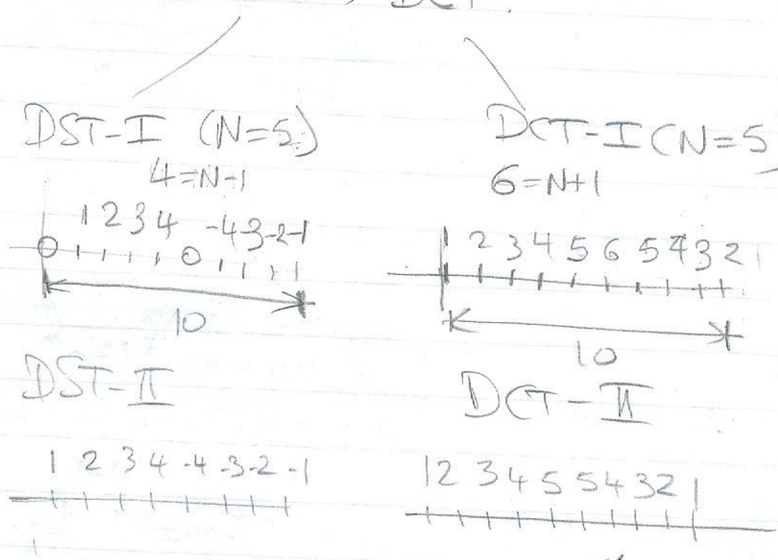
Rescale as $\dfrac{2u_1 - 2u_2}{h^2} = 3\times(\dots)$

→ same as earlier

Next: (DST, DCT), FFT

02/16 FFT, incl. parallel
- Butterfly diag.
- Sparse matrix products

Motivation:
- convolutions
- DST, DCT.



(Image proc.)
JPEG

History: Gauss 1805
Danielson-Lanczos 1942 ($N=2^n$)
Cooley-Tukey 1965 (N comp.)

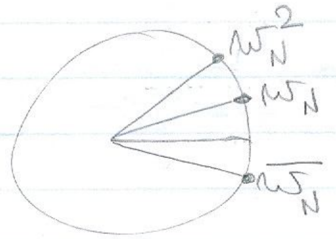
Radix-2 FFT.

$$\omega = e^{-2\pi i/N} \qquad (\omega_N)$$

$$\begin{cases} c_k = \sum_{j=0}^{N-1} f_j \omega^{jk} & \text{DFT} \\ f_j = \frac{1}{N} \sum_{k=0}^{N-1} c_k \bar{\omega}^{jk} & \text{IDFT} \end{cases}$$

$$c = F_N f$$
$$f = \frac{1}{N} F_N^* c.$$

Basic facts: $\omega_N^N = 1$



$\boxed{\omega_N^2 = \omega_{N/2}}$

$\overline{\omega_N} = \omega_N^{-1}$

$(\omega_N^k)^N = 1 \qquad \omega_N^{N/2} = -1$

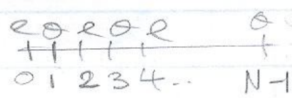
(N roots of unity)

Basic idea: multiscale recursion

Assume $N = 2^m$.

Split into odd-even:

$$c_k = \sum_{j=0}^{N-1} f_j \omega_N^{jk}$$



$$= \sum_{j=0}^{N/2-1} f_{2j}\, \omega_N^{(2j)k} \quad \leftarrow (A)$$

$$+ \sum_{j=0}^{N/2-1} f_{2j+1}\, \omega_N^{(2j+1)k} \quad \leftarrow (B)$$

$$(A) = \sum_{j=0}^{N/2-1} f_{2j}\, \omega_{N/2}^{jk} = [\ ] \qquad 0 \le k \le N-1$$

$\rightarrow$ for $0 \le k \le \frac{N}{2} - 1$, length-$\frac{N}{2}$ DFT

$\rightarrow$ for $\frac{N}{2} \le k \le N-1$, $k = \tilde{k} + \frac{N}{2}$

$$\omega_{N/2}^{jk} = \left(\omega_{N/2}^{j}\right)^{N/2} \omega_{N/2}^{j\tilde{k}}$$

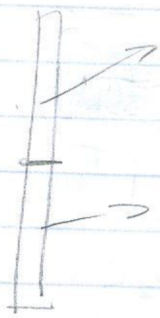
$$= \omega_{N/2}^{j\tilde{k}}$$

$\rightarrow$ concatenate two copies of length-$N/2$ DFT.

$$c_{k,A} = \begin{bmatrix} F_{N/2}(f_e) \\ F_{N/2}(f_e) \end{bmatrix}$$

$$B = \sum_{j=0}^{N/2-1} f_{2j+1}\, \omega_N^{(2j+1)k}$$

$$= \omega_N^{k} \sum_{j=0}^{N/2-1} f_{2j+1}\, \omega_{N/2}^{jk}$$



for $0 \le k \le \frac{N}{2}-1$, length-$\frac{N}{2}$ DFT $\times\, \omega_N^k$

for $\frac{N}{2} \le k \le N-1$, $k = \tilde{k} + \frac{N}{2}$

$$\omega_N^k = \omega_N^{N/2}\, \omega_N^{\tilde{k}} = -\omega_N^{\tilde{k}}$$

$$c_{k,B} = \begin{bmatrix} D\, F_{N/2}(f_o) \\ -D\, F_{N/2}(f_o) \end{bmatrix}$$

with $D = \mathrm{diag}(\omega_N^k)$

<u>Concl</u>: $c = F_N f = \begin{bmatrix} F_{N/2}(f_e) + D\, F_{N/2}(f_o) \\ F_{N/2}(f_e) - D\, F_{N/2}(f_o) \end{bmatrix}$

<u>Algo</u> for F_N from $F_{N/2}$:

function $c = F_N(f)$

Split $f \to (f_e, f_o)$

$c_e = F_{N/2}(f_e)$

$c_o = F_{N/2}(f_o)$

$$c = \begin{bmatrix} c_e + D c_o \\ c_e - D c_o \end{bmatrix}$$

Recursive; if $N=1$, $c=f$.

Matrix form:

$$\begin{bmatrix} f_e \\ f_o \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\cdots \\ \cdots 0\,1\,0 \\ \hline 0\,1\,0\,0\,0\cdots \\ 0\,0\,0\,1\,0\cdots \\ \cdots \end{bmatrix} \begin{bmatrix} f \end{bmatrix}$$

$$= \begin{bmatrix} I_e \\ I_o \end{bmatrix}$$

$$\boxed{F_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & \\ & F_{N/2} \end{bmatrix} \begin{bmatrix} I_e \\ I_o \end{bmatrix}}$$

Complexity: o,e split: N ops.
$2 \times F_{N/2}$: $2 \times C(N/2)$
assembly: $3N$ ops

$$\begin{cases} C(N) = 2\,C\left(\frac{N}{2}\right) + \underset{\nu=4}{m}N \\ C(1) = 1 \end{cases}$$

$$C(N) = mN + 2\left(m\frac{N}{2} + 2\cdot\left(m\frac{N}{4} + \cdots\right)\right)$$

$$= \underbrace{mN + mN + \cdots}_{\substack{\log_2(N) \\ \text{levels}}} + \underbrace{2^{\log_2 N}}_{N}$$

$$= O(N \log N)$$

$$F_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/4} & D_{N/4} & & \\ I_{N/4} & -D_{N/4} & & \\ & & I_{N/4} & D_{N/4} \\ & & I_{N/4} & -D_{N/4} \end{bmatrix}$$

$$\times \begin{bmatrix} F_{N/4} & & & \\ & F_{N/4} & & \\ & & F_{N/4} & \\ & & & F_{N/4} \end{bmatrix} \times \begin{bmatrix} e,o \\ \text{selector} \end{bmatrix}$$

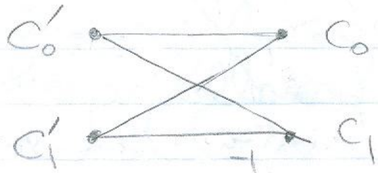
Composition of e-o selectors: bit reversal!

0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Butterfly form of the recursive assembly:

$D_1 = 1$

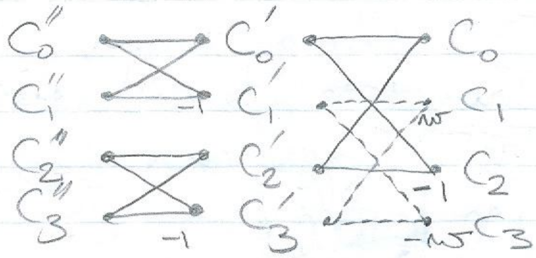
$N=2$



$$\begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} c_0' \\ c_1' \end{pmatrix}$$

$D_2 = \begin{pmatrix} 1 & \\ & \omega \end{pmatrix}$ with $\omega = -i$

$N=4$



$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \left(\begin{array}{cc|cc} 1 & & 1 & \\ & 1 & & \omega \\ \hline 1 & & -1 & \\ & 1 & & -\omega \end{array}\right) \begin{pmatrix} c_0' \\ c_1' \\ c_2' \\ c_3' \end{pmatrix}$$

Parallel FFT:

1) parallel 1D:

function $C = F_N(f)$

split

$C_e = F_N(f_e) \rightarrow$ CPU

$C_o = F_N(f_o) \rightarrow$ CPU

assemble

2) parallel multi-D: e.g.

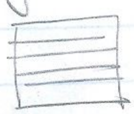
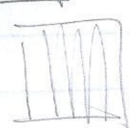
$$C_{k_1 k_2} = \sum_{j_1 j_2 = 0}^{N-1} f_{j_1 j_2} \, \omega^{j_1 k_1 + j_2 k_2}$$

$$c = (F_N \otimes F_N) f$$

- F_N along each row
- F_N along each col

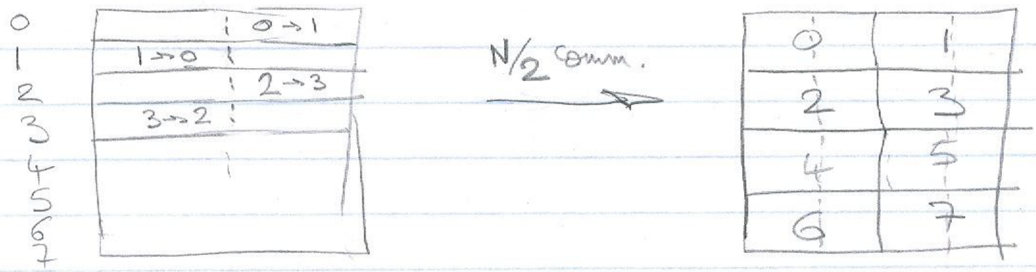
2a) shared memory:
parallelize over rows/cols

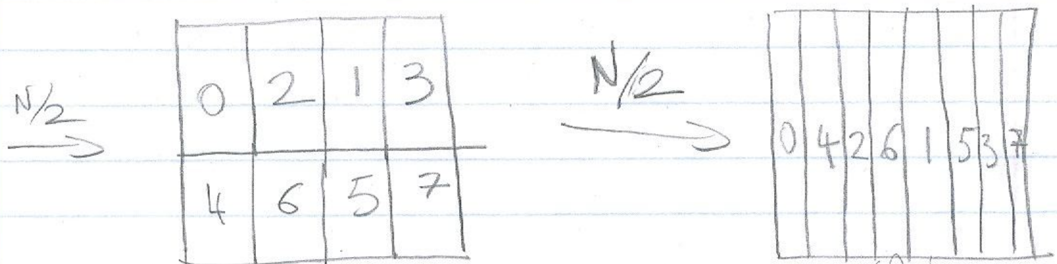
2b) distributed memory:

- Distribute over rows 
- FFT along rows 
- Transpose to dist. over cols
- FFT along cols

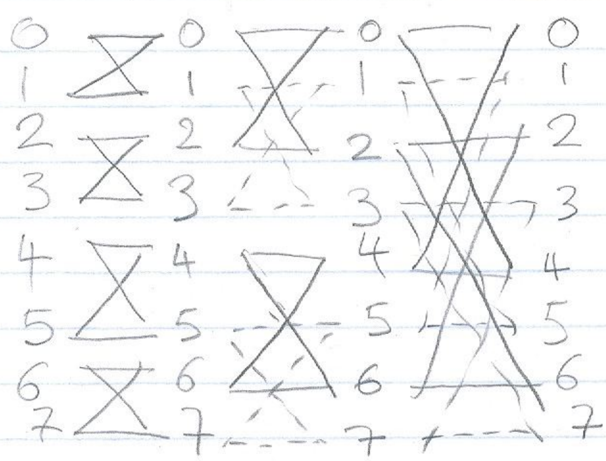
$\rightarrow$ All to all!

Transposition: Binary exchange





(bit-reversed order)



log N stages, N/2 comm. per stage

Direct communication

N stages, 1 comm. per stage