Fast-Fourier-Based

Poisson Solvers

1D Periodic Poisson eqn

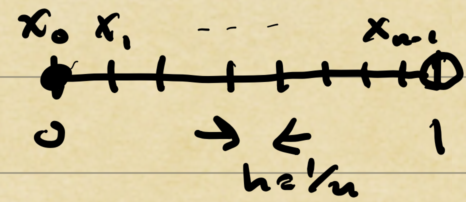
$$\int_0^1 f(x)\,dx = 0$$

$$-u_{xx} = f \qquad x \in [0,1]$$

$$u(0) = u(1), \quad u'(0) = u'(1)$$

$\Rightarrow$ solve for u

Discretize



$$-\left(\frac{u_{k+1} - 2u_k + u_{k-1}}{h^2}\right) = f_k \qquad \text{for} \quad k = 0, \dots, n-1$$

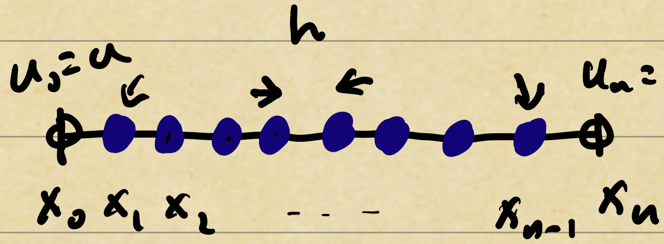
$$\underbrace{\begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ -1 & & & -1 & 2 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ u_{n-1} \end{bmatrix}}_{\underline{u}_n} = \underbrace{\begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ \vdots \\ f_{n-1} \end{bmatrix}}_{\underline{f}_n}$$

$$\hat{f}_0 = \frac{1}{n} \sum_{j=0}^{n-1} f_j \approx \int_0^1 f(x)\,dx$$

$$\underbrace{\begin{bmatrix} 0 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_{n-1} \end{bmatrix}}_{\Lambda} \begin{bmatrix} \hat{u}_0 \\ \hat{u}_1 \\ \vdots \\ \hat{u}_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{f}_1 \\ \vdots \\ \hat{f}_n \end{bmatrix}$$

Example: Dirichlet BCs

$$-u_{xx} = f \qquad x \in [0,1]$$
$$u(0) = a, \qquad u(1) = b$$



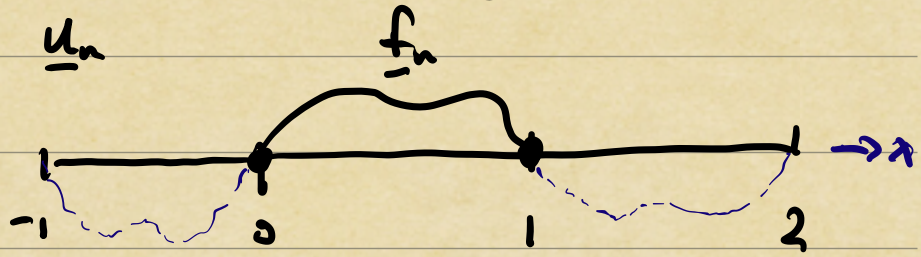
$$\frac{1}{h^2}\underbrace{\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}}_{K}\begin{bmatrix} u_1 \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 + a/h^2 \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} + b/h^2 \end{bmatrix}$$

DST - Type 1

$$K = S \Lambda S^{-1}$$

diagonal

$$\lambda_j = 2 - 2\cos\left(\frac{\pi j}{n}\right) \qquad j = 1, \dots, n-1$$



"Periodic odd Extension"

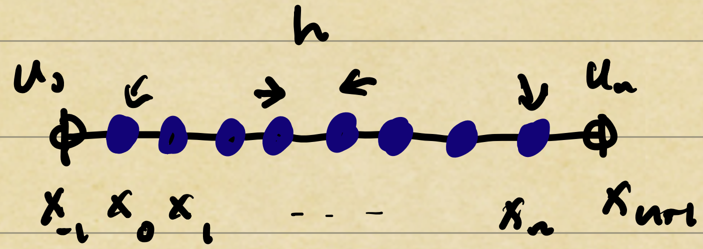
Fast Algorithm:

1) $\hat{\underline{f}}_n = S^{-1}\underline{f}_n \qquad \mathcal{O}(n\log n)$
2) $\Lambda_n \hat{u}_n = \hat{f}_n \qquad \mathcal{O}(n)$
3) $u_n = S\hat{u}_n \qquad \mathcal{O}(n\log n)$

Example: Neumann B.C.'s

$-u_{xx} = f \qquad x \in [0,1]$

$u'(0) = a, \qquad u'(1) = b$

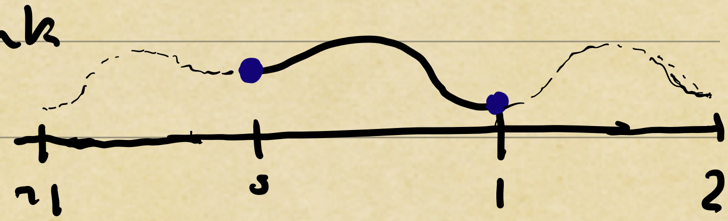


$$\frac{1}{h^2}\begin{array}{c} n+3 \\ \begin{bmatrix} -1 & 2 & 1 & & & \\ & -1 & 2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \end{bmatrix} \end{array} \begin{bmatrix} u_{-1} \\ u_0 \\ \vdots \\ u_n \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} f_0 \\ \vdots \\ f_n \end{bmatrix} \; n+1$$

Add $\quad \frac{1}{2h}\begin{bmatrix} -1 & 0 & 1 & & & \\ & & & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{-1} \\ u_0 \\ \vdots \\ u_n \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \; +2$

$$\begin{array}{cc} & n+1 \\ n+1 & \begin{bmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -2 & 2 \end{bmatrix} \end{array} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} f_0 - 2a/h \\ \vdots \\ f_n - 2b/h \end{bmatrix}$$

B = Circulant + Low-rank



$B \Rightarrow$ diagonalized by DCT - Type I

$\Rightarrow$ Analogous fast solver

Toeplitz Matrix: "Locally Translation Invariant"

$$T = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & \vdots \\ \vdots & a_1 & a_0 & \ddots & a_{-2} \\ & & a_1 & \ddots & a_{-1} \\ a_{n-1} & \cdots & & a_1 & a_0 \end{bmatrix}$$

"circulant embedding" → generalize "even/odd extension" over boundary

$$C_T = \begin{array}{c} \quad T \\ \left[\begin{array}{ccccc|ccccc} a_0 & a_{-1} & \cdots & & a_{-(n-1)} & 0 & a_{n-1} & \cdots & & a_1 \\ a_1 & a_0 & \ddots & & \vdots & a_{-(n-1)} & \ddots & & & a_2 \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & \vdots & & \ddots & & \vdots \\ a_{n-1} & \cdots & & a_1 & a_0 & a_{-1} & a_{-2} & \cdots & a_{-(n-1)} & a_{n-1} \\ \hline 0 & a_{n-1} & \cdots & & a_1 & a_0 & a_{-1} & \cdots & & a_{-(n-1)} \\ a_{-(n-1)} & \ddots & & & \vdots & a_1 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & & a_{n-1} & \vdots & \ddots & \ddots & & a_{-1} \\ a_{-1} & \cdots & & & 0 & a_{n-1} & \cdots & & a_1 & a_0 \end{array}\right] \\ \qquad\qquad T \end{array}$$

Matvec: $T\underline{v} = \begin{bmatrix} I_n & 0 \end{bmatrix} C_T \begin{bmatrix} \underline{v} \\ 0 \end{bmatrix}$

fast $C_T \underline{w} = V_n^* \Lambda_n V_n \underline{w}$

$\mathcal{O}(n \log n)$
for iterative methods