Lecture 7:

In discrete case, a differential equation can be discretized as:

$$D\vec{u} = \vec{g}$$

where $\vec{u} = \begin{pmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_N) \end{pmatrix}$ = values of u at N points $\{x_1, x_2, \dots, x_N\}$

$\vec{g} = \begin{pmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_N) \end{pmatrix}$ = values of g at N points $\{x_1, x_2, \dots, x_N\}$

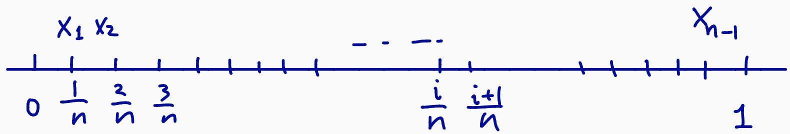
D = $N \times N$ matrice approximating the differential operator.

<u>Question:</u> Can we "transform" $\vec{u}$ and $\vec{g}$ to turn the (BIG) linear system to SIMPLE algebraic equation?

<u>Answer:</u> YES! Discrete Fourier Transform!!

$$\frac{d^2 f}{dx^2}(x) = g(x) \quad 0 < x < 1 \quad \text{with } f(0) = 1 \; ; \; f(1) = 2 .$$

Discretize $(0, 1)$:



Approximation of $\frac{d^2 f}{dx^2}$:

$$f(x_{i+1}) \approx f(x_i) + \frac{1}{n} f'(x_i) + \frac{1}{2!}\frac{1}{n^2} f''(x_i)$$

(Taylor's expansion)

$$+ \; f(x_{i-1}) \approx f(x_i) - \frac{1}{n} f'(x_i) + \frac{1}{2!}\frac{1}{n^2} f''(x_i)$$

$$f(x_{i+1}) + f(x_{i-1}) \approx 2f(x_i) + \frac{1}{n^2} f''(x_i)$$

$$\therefore \frac{d^2 f}{dx^2}(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{\left(\frac{1}{n^2}\right)}$$

$$\therefore \frac{d^2 f}{dx^2}(x_i) = g(x_i) \iff \left\{ \begin{array}{l} \vdots \\ \dfrac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{\left(\frac{1}{n^2}\right)} = g(x_i) \\ \vdots \end{array} \right.$$

$$D\vec{f} = \vec{g} \; ; \; \vec{f} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-1}) \end{pmatrix} ; \; \vec{g} = \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_{n-1}) \end{pmatrix}$$

for $i = 1, 2, \ldots, n-1$

Question: Extension to discrete case (Computational Math.)

Answer: Discrete Fourier Transform

Goal: ① Define discrete Fourier Transform (DFT)

② Use DFT to solve discretized differential eqt.

Definition: (Discrete Fourier Transform) Given $f_0, f_1, \ldots, f_{n-1} \in \mathbb{C}$, then the discrete Fourier Transform (DFT) is defined as:

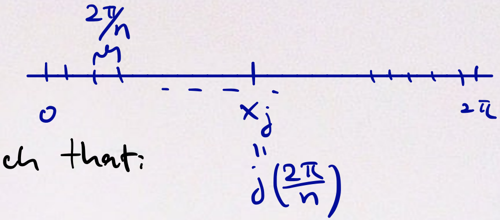
$$\vec{c} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} \in \mathbb{C}^n \quad \text{where} \quad c_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j e^{-i\left(\frac{2jk\pi}{n}\right)} \quad \text{for } k = 0, 1, 2, \ldots, n-1$$

The inverse discrete Fourier Transform recovers the original signal:

$$f_j = \sum_{k=0}^{n-1} c_k e^{i\left(\frac{2jk\pi}{n}\right)} \quad \text{for } j = 0, 1, 2, \ldots, n-1$$

Motivation 1: Let $f(x)$ defined on $[0, 2\pi]$

Approximate $f(x)$ by:



$$F_n(x) = \sum_{k=0}^{n-1} c_k e^{ikx}, \quad x \in [0, 2\pi] \text{ such that:}$$

$$\boxed{F_n(x_j) = f(x_j) \overset{\text{def}}{:=} f_j}, \quad x_j = \frac{2\pi}{n}. \quad (\text{for all } j = 0, 1, 2, \ldots, n-1)$$

$$(*)\begin{cases} F_n(\overset{0}{\overset{\|}{x_0}}) = c_0 + c_1 + c_2 + \ldots + c_{n-1} = f_0 \\ F_n(x_1) = c_0 + c_1 e^{ix_1} + c_2 e^{i2x_1} + \ldots + c_{n-1} e^{i(n-1)x_1} = f_1 \\ \quad \vdots \\ F_n(x_{n-1}) = c_0 + c_1 e^{ix_{n-1}} + c_2 e^{i2x_{n-1}} + \ldots + c_{n-1} e^{i(n-1)x_{n-1}} = f_{n-1} \end{cases}$$

Let $\omega = e^{i\frac{2\pi}{n}} = e^{ix_1}$, $\omega^2 = e^{i\frac{4\pi}{n}} = e^{ix_2}$, $\omega^3 = e^{ix_3}$, ... etc.

$\therefore$ (*) can be written as:

$$A_\omega = \boxed{\begin{pmatrix} 1 & 1 & - & - & - & 1 \\ 1 & \omega & \omega^2 & - & - & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & - & - & \omega^{2(n-1)} \\ & & \vdots & & & \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & & \omega^{(n-1)^2} \end{pmatrix}} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{pmatrix}$$

$$\therefore \left(A_\omega \overline{A_\omega}\right)_{j,k} = 1\cdot 1 + \omega^j \overline{\omega}^k + \omega^{2j}\overline{\omega}^{2k} + \ldots + \omega^{(n-1)j}\,\overline{\omega}^{(n-1)k}$$

$$= 1 + e^{\frac{2\pi i(j-k)}{n}} + e^{\frac{4\pi i(j-k)}{n}} \ldots + e^{\frac{2\pi i(n-1)(j-k)}{n}}$$

$$= \begin{cases} n & \text{if } j = k \\ \dfrac{1 - \left(e^{\frac{2\pi i(j-k)}{\cancel{n}}}\right)^{\cancel{n}}}{1 - e^{2\pi i(j-k)/n}} = 0 & \text{if } j \neq k \end{cases}$$

$$\therefore \quad A_\omega \overline{A_\omega} = nI = \overline{A_\omega} A_\omega$$

We have:
$$\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = A_\omega^{-1} \begin{pmatrix} f_0 \\ \vdots \\ f_{n-1} \end{pmatrix} = \frac{\overline{A_\omega}}{n} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix}$$

$$\therefore \quad c_k = \frac{1}{n}\left(f_0 + e^{-\frac{2\pi i}{n} k} f_1 + \cdots + e^{-\frac{2\pi i}{n} k(n-1)} f_{n-1} \right)$$

for $k = 0, 1, 2, \ldots, n-1$

$$\parallel$$

DFT

<u>Remark:</u> Comp cost for DFT?

n^2 multiplication
$+$
$(n-1)n$ addition $= O(n^2)$

Remark: Computational cost for DFT is:

$$\left.\begin{array}{c} n^2 \text{ multiplication} \\ + \\ n(n-1) \text{ addition} \end{array}\right\} = \mathcal{O}(n^2)$$

Example: Consider $f(t) = 5 + 2\cos\left(t - \frac{\pi}{2}\right) + 3\cos(2t)$. f is 2π-periodic. Divide $[0, 2\pi]$ by 4 partitions. Find the DFT of f (discretized by 4 points).

$$f_0 = f(0) = 8 \,;\quad f_1 = f\left(\frac{2\pi}{4}\right) = 4 \,;\quad f_2 = f\left(\frac{4\pi}{4}\right) = 8 \,;\quad f_3 = f\left(\frac{6\pi}{4}\right) = 0$$

$\therefore$ DFT: $\displaystyle c_k = \frac{1}{4}\sum_{j=0}^{3} f_j\, e^{-i\left(\frac{2jk\pi}{4}\right)}$ for $k = 0, 1, 2, 3$ or

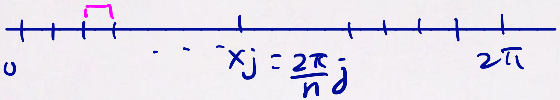
$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \frac{\overline{A_\omega}}{\underset{\substack{\shortparallel \\ 4}}{n}} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} = \frac{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}}{4} \begin{pmatrix} 8 \\ 4 \\ 8 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ -i \\ 3 \\ i \end{pmatrix}$$

$$\omega = e^{\frac{2\pi i}{4}}$$

Motivation 2: Fourier Transform $\xleftrightarrow[\text{related}]{}$ Fourier Series extended to $(-\infty, \infty)$

Fourier coefficients: $$c_k = \frac{1}{2\pi}\int_0^{2\pi} \underbrace{f(x)}_{2\pi\text{-periodic}} e^{-ikx}\,dx$$

Divide: $\Delta x = \frac{2\pi}{n}$



We can approximate the integration:

$$c_k \approx \frac{1}{2\pi}\sum_{j=0}^{n-1} f(x_j)\, e^{-ikx_j}\,\Delta x = \frac{1}{\cancel{2\pi}}\sum_{j=0}^{n-1} f_j\, e^{-i\frac{2\pi}{n}jk}\left(\frac{\cancel{2\pi}}{n}\right)$$

$$= \frac{1}{n}\sum_{j=0}^{n-1} f_j\, e^{-i\frac{2jk\pi}{n}} \quad \text{for } k = 0, 1, 2, \ldots, n-1$$

$$= \text{DFT}$$

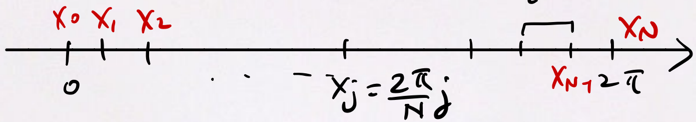
DFT = approximation of (complex) Fourier coefficient.

DFT and numerical diff eqt

Consider : $\dfrac{d^2 u}{dx^2} = f$ for $x \in [0, 2\pi]$ with periodic boundary condition $u(0) = u(2\pi)$

Suppose f is measured only at N discrete points :

$x_0, \; x_1, \; \ldots, \; x_{N-1}$ $\qquad h = \dfrac{2\pi}{N}$



Let $\vec{f} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ \\ f(x_{N-1}) \end{pmatrix} \overset{\text{def}}{=} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ \\ f_{N-1} \end{pmatrix} \in \mathbb{R}^N$ and $\vec{u} = \begin{pmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ \\ u(x_{N-1}) \end{pmatrix} \overset{\text{def}}{=} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ \\ u_{N-1} \end{pmatrix} \in \mathbb{R}^N$

(unknown)

By Taylor's expansion,

$$u(x_j+h) \approx u(x_j) + h u'(x_j) + \frac{h^2}{2} u''(x_j) \quad \text{———} \quad (1)$$

$$u(x_j-h) \approx u(x_j) - h u'(x_j) + \frac{h^2}{2} u''(x_j) \quad \text{———} \quad (2)$$

$$(1)+(2): \quad u''(x_j) \approx \frac{u(\underbrace{x_j-h}_{= x_{j-1}}) - 2u(x_j) + u(\overbrace{x_j+h}^{= x_{j+1}})}{h^2}$$

$$\therefore \quad u''(x_j) \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2}$$ (Central difference approximation)

for $j = 0, 1, 2, \ldots, N-1$

Thus:
$$\begin{pmatrix} u''(x_0) \\ u''(x_1) \\ \vdots \\ u''(x_{N-1}) \end{pmatrix} \approx \tilde{D}\vec{u} \text{ where } \tilde{D} = \frac{1}{h^2} \begin{array}{c} \begin{array}{ccccc} u_0 & u_1 & \cdots & & u_{N-1} \end{array} \\ \begin{pmatrix} -2 & 1 & \cdots & & \boxed{1} \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ \boxed{1} & & & 1 & -2 \end{pmatrix} \end{array}$$

(Use the fact that $u_0 = u_N$, $u_{-1} = u_{N-1}$)

$\therefore$ $\frac{d^2u}{dx^2} = f$ can be discretized as $\underset{M_{N\times N}}{\ni}$ $\boxed{\tilde{D}\vec{u} = \vec{f}}$ (Linear System)

Numerical differential eqt

Remark: $\tilde{D}$ is BIG matrix!!

Goal: Design numerical spectral method to solve $\tilde{D}\vec{u} = \vec{f}$.

Need to: Determine eigenvalues / eigenvectors of $\tilde{D}$