Fast Fourier Transforms

BVP $-\partial_x^2 u = f$

$u(0) = u(1)$
and
$u'(0) = u'(1)$

$\underset{\text{Finite Diff.}}{\overset{\text{discretize}}{\Longrightarrow}}$

$$\underbrace{\begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ -1 & & & -1 & 2 \end{bmatrix}}_{A_n} \underbrace{\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}}_{\underline{u}_n} = \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}}_{\underline{f}_n}$$

In general, A_n has bandwidth m for m^{th}-order diff.

But, A_n has another structure - "circulant" matrix

KEY: A_n is diagonalized by Discrete Fourier Transform

$$A = V \Lambda V^*$$

V, V^* have $\mathcal{O}(n \log n)$ matvecs

Fast Algorithm:

Key Step
Apply V, V^* fast to vector f_n

1) Compute $\hat{\underline{f}}_n = V^* \underline{f}_n$ — FFT $\mathcal{O}(n \log n)$

2) Solve $\Lambda \hat{\underline{u}}_n = \underline{f}_n$ — diagonal $\mathcal{O}(n)$

3) Solve $V^* \underline{u}_n = \hat{u}_n$ — (Inverse) FFT $\mathcal{O}(n \log n)$

Radix-2 FFT: Start with $n = 2^\ell$, $\omega_n = e^{2\pi i/n}$

$$\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{n-1} \end{bmatrix} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega_n^0 & \omega_n^0 & \omega_n^0 & \cdots & \omega_n^0 \\ \omega_n^0 & \omega_n^1 & \omega_n^2 & \cdots & \omega_n^{(n-1)} \\ \omega_n^0 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega_n^0 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{bmatrix}$$

$$\underline{\hat{f}}_n \qquad\qquad V_n \qquad\qquad \underline{f}_n$$

$\Rightarrow$ V is a dense $n \times n$ matrix, but has a <u>lot</u> of algebraic structure!

<u>$n=1$</u> $\hat{f}_0 = 1\ f_0$ $\qquad V_1 = 1$

<u>$n=2$</u>
$$\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$$

<u>$n=4$</u>
$$\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \hat{f}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Purple : Green $\qquad\qquad$ Yellow : Orange

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \pm \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix} = \pm \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}}_{\text{diagonal scaling}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

2x2 DFT $= V_2$!

Basic Idea: "Build up" $n \times n$ DFT recursively from smaller DFTs, carefully re-using computation.

We need 3 facts to do this systematically.

__Fact 1:__ DFT is polynomial evaluation at the roots of unity

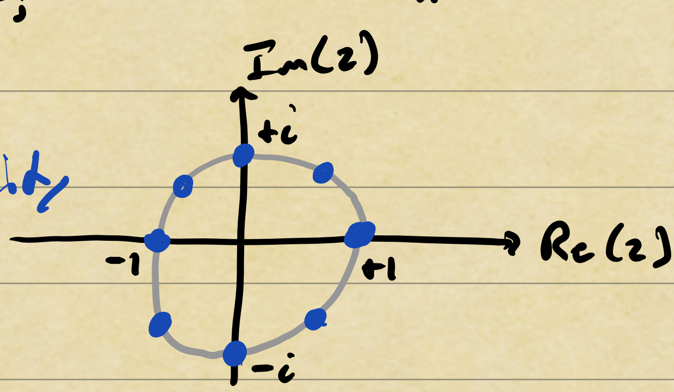
output coefficients

$$\underbrace{\hat{f}_j}_{\text{output coefficients}} = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} f_k e^{-2\pi i jk/n} = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} f_k z_j^k$$

{ input coefficients }

where $z_j = e^{-2\pi i j/n} = \omega_n^j$ = n^{th} roots of unity

$(n=8)^{th}$ roots of unity



$\Rightarrow$ if $p(z) = \sum_{k=0}^{n-1} f_k z^k$, then $\hat{f}_j = \frac{1}{\sqrt{n}} p(z_j)$

degree n-1 poly

<u>Fact 2:</u> Polynomial splits into even/odd parts.

$$p(z) = \overset{\overset{1}{\shortparallel}}{f_0 z^0} + f_1 z^1 + f_2 z^2 + f_3 z^3 + \dots$$

$$= \sum_{k=0}^{\frac{n}{2}-1} f_{2k} z^{2k} \quad + \quad z \sum_{k=0}^{\frac{n}{2}-1} f_{2k+1} z^{2k}$$

$$= \underbrace{p_e(z^2)}_{\text{even}} \quad + \quad z\, \underbrace{p_o(z^2)}_{\text{odd}}$$

$\Rightarrow$ p_e and p_o are deg $\frac{n}{2}-1$ polynomials

$\Rightarrow$ DFT splits into evaluation of two deg $\frac{n}{2}-1$ polynomials at squared roots of unity.

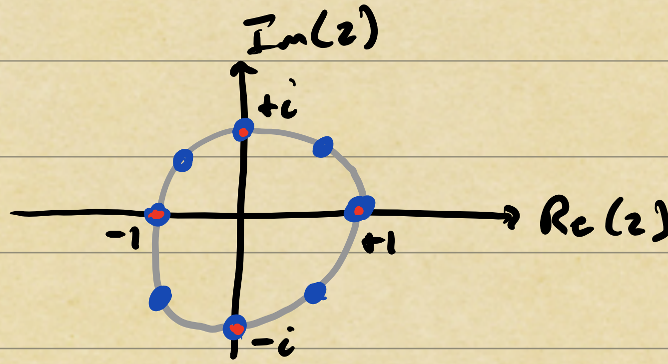
<u>Fact 3:</u> Roots of unity have special symmetries.

a) $z_j^2 = (e^{-2\pi i j/n})^2 = e^{-2\pi i j/(n/2)} = \omega_{n/2}^j$

$\Rightarrow$ $\frac{n}{2}$ roots of unity! smaller

$(n=8)^{th}$ roots of unity

$(\frac{n}{2}=4)^{th}$ roots of unity



$\Rightarrow$ $p_e(z^2)$ and $p_o(z^2)$

have form of $p(z)$, with $n \to \frac{n}{2}$

b) $$z_j^2 = e^{-2\pi i j/(n/2)} \underbrace{e^{2\pi i \frac{n/2}{n/2}}}_{=1} = e^{-2\pi i (j-n/2)/(n/2)}$$

$\Rightarrow$ only need to evaluate $p_e(z^2)$ and $p_o(z^2)$ at first $z_0, \ldots, z_{n/2-1}$ roots of unity.

c) $$z_j = e^{-2\pi i j/n} \underbrace{\left(-e^{2\pi i \frac{n/2}{n}}\right)}_{=-1} = -e^{-2\pi i (j-\frac{n}{2})/n}$$
$$= -z_{j-n/2}$$

$\Rightarrow$ Only need first $z_0, \ldots, z_{n/2-1}$ for $z p_o(z^2)$.

Let's put it altogether by calculating

$$\underline{f}_n \overset{\text{Fact 1}}{=} \begin{bmatrix} p(z_0) \\ \vdots \\ p(z_{\frac{n}{2}-1}) \\ p(z_{n/2}) \\ \vdots \\ p(z_{n-1}) \end{bmatrix} \overset{\text{Fact 2}}{=} \begin{bmatrix} p_e(z_0^2) \\ \vdots \\ p_e(z_{n/2-1}^2) \\ p_e(z_{n/2}^2) \\ \vdots \\ p_e(z_{n-1}^2) \end{bmatrix} + \begin{bmatrix} z_0 p_o(z_0^2) \\ \vdots \\ z_{\frac{n}{2}-1} p_o(z_{n/2-1}^2) \\ z_{n/2} p_o(z_{n/2}^2) \\ \vdots \\ z_{n-1} p_o(z_{n-1}^2) \end{bmatrix}$$

$$= \begin{bmatrix} p_e(z_0^2) \\ \vdots \\ p_e(z_{\frac{n}{2}-1}^2) \\ p_e(z_0^2) \\ \vdots \\ p_e(z_{n/2-1}^2) \end{bmatrix} + \begin{bmatrix} z_0 p_o(z_0^2) \\ \vdots \\ z_{\frac{n}{2}-1} p_o(z_{\frac{n}{2}-1}^2) \\ -z_0 p_o(z_0^2) \\ \vdots \\ -z_{\frac{n}{2}-1} p_o(z_{\frac{n}{2}-1}^2) \end{bmatrix}$$

Fact 2 $\Rightarrow$ copy of first $\frac{n}{2}$ entries!

Fact 1 + Fact 2 $\Rightarrow$ Again, copy!

$\frac{n}{2} \times \frac{n}{2}$ identity

$$= \underbrace{\begin{pmatrix} I_{\frac{n}{2}} & I_{\frac{n}{2}} \\ I_{n/2} & -I_{\frac{n}{2}} \end{pmatrix}}_{DFT(2) \otimes I_{n/2}} \begin{bmatrix} p_e(z_0^2) \\ \vdots \\ p_e(z_{n/2-1}^2) \\ z_0 p_o(z_0^2) \\ \vdots \\ z_{\frac{n}{2}-1} p_o(z_{n/2-1}^2) \end{bmatrix}$$

So to compute $\hat{f}_n = V f_n$, only need to compute

$$p_e(z_j^2) = \sum_{k=0}^{\frac{n}{2}-1} f_{2k} z_j^{2k} = \sum_{k=0}^{\frac{n}{2}-1} f_{2k} (\omega_{n/2}^j)^k$$

$$z_j p_o(z_j^2) = z_j \sum_{k=0}^{\frac{n}{2}-1} f_{2k+1} z_j^{2k} = z_j \sum_{k=0}^{\frac{n}{2}-1} f_{2k+1} (\omega_{n/2}^j)^k$$

for $j = 0, \dots, n/2 - 1$. These are just two DFT's of size $\frac{n}{2}$, with "twiddle factors" scaling the odd part $p_o(z_j^2) \to z_j p_o(z_j^2)$

We can do the same splitting again with the $n/2$ DFTs, and continue recursively:

$$\mathrm{DFT}(n)$$

n additions $+\ n/2$ multiplies $= \frac{3n}{2}$ flops

$$\mathrm{DFT}(n/2) \qquad \mathrm{DFT}(n/2)$$

$2 \times \frac{3n}{4} = \frac{3n}{2}$ flops

$$\mathrm{DFT}(n/4) \quad \mathrm{DFT}(n/4) \quad \mathrm{DFT}(\tfrac{n}{4}) \quad \mathrm{DFT}(n/4)$$

$\vdots$

$\log_2(n)$ levels

$$\text{Total FLOPs} = \frac{3n}{2} \log_2 n = O(n \log n)$$