Lecture 8: Fast Fourier Transform (FFT)

Note Title

Introductory Remarks:

* Various programs exist, e.g., MATLAB, R, Mathematica, ..., as well as the public domain source codes, e.g., FFTPACK @ netlib, ...
Perhaps, the most popular one is:
FFTW available from http://www.fftw.org

* It's known by Gauss!
See the article by M. Heideman et al. (1985).

* $W_N^* \mathbf{f}$ (forward) or $W_N \mathbb{F}$ (inverse) cost $O(N^2)$ if you use the conventional matrix-vector multiplications.
$\Rightarrow$ Too expensive for large N.

* W_N has a very special structure and FFT algorithms fully utilize that to achieve $O(N \log_2 N)$ operations.

In this lecture, we use the following notation / convention for convenience:

$$F[k] = \mathcal{D}_N\{\mathbf{f}\}[k] = \sum_{\ell=0}^{N-1} f[\ell]\,\omega_N^{-k\ell}.$$

In other words, we use the notation $f[\ell]$ for f_ℓ, $F[k]$ for F_k, $\ell, k = 0, 1, \cdots, N-1$, there's no normalizing const. $\frac{1}{\sqrt{N}}$.

(We can always normalize by $1/\sqrt{N}$ later.)

Note first:

$$\omega_N^{(k+mN)\cdot(l+nN)} = \omega_N^{kl} \quad \text{by periodicity}$$

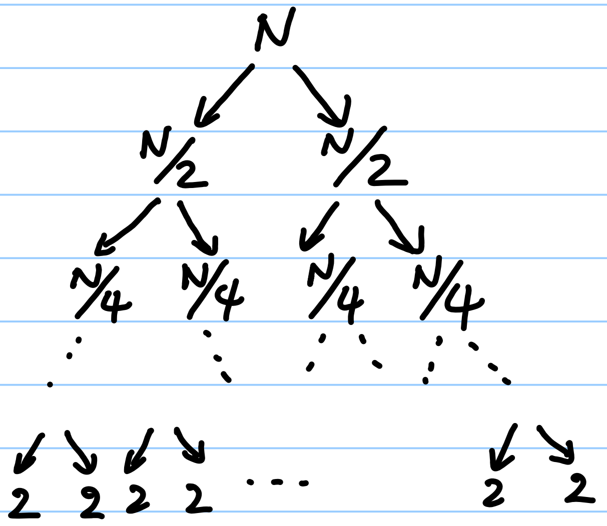
$$\omega_N^{k\cdot l} \in \{\underbrace{\omega_N^0}_{=1}, \omega_N^1, \cdots, \underbrace{\omega_N^{N/2}}_{=-1.}, \cdots, \omega_N^{N-1}\}$$

⇒ For each fixed N, we can store these N complex numbers as a table once & for all!

★ The Basic Idea of FFT (Cooley-Tukey, 1965)

It's hierarchical in nature (V. Rokhlin's comment: "it's created by God.")

Say $N = 2^n$.



	mat-vec comp
level 0	$O(N^2)$
level 1	$O(2\cdot(\frac{N}{2})^2)$
level 2	$O(4\cdot(\frac{N}{4})^2)$
⋮	⋮
level $n-1$	$O(2^{n-1}\cdot 2^2)$ $= O(2N)$

$n = \log_2 N.$

But in reality, we cannot decrease the complexity by ½ at each level.

∃ many variants of FFT. We will only describe the following.

☆ Decimation in Time (DIT) Algorithm

Suppose $N = 2^n$ as usual.

The heart of the matter: Split an input seq. into two subseq.'s of even & odd indices of the original!

- Define two subseq.'s of $\{f[\ell]\}_{\ell=0}^{N-1}$:

$$\begin{cases} f_0[\ell] := f[2\ell] \\ f_1[\ell] := f[2\ell+1] \end{cases} \qquad \ell = 0, 1, \cdots, \frac{N}{2}-1 .$$

- Then we have

$$F[k] = \sum_{\ell=0}^{N-1} f[\ell]\,\omega_N^{-k\ell} = \sum_{\ell=0}^{N/2-1} f[2\ell]\,\omega_N^{-k\cdot 2\ell} + \sum_{\ell=0}^{N/2-1} f[2\ell+1]\,\omega_N^{-k(2\ell+1)}$$

$$\omega_N^{-k\cdot 2\ell} = e^{-\frac{2\pi i k\cdot 2\ell}{N}} = e^{-\frac{2\pi i k\ell}{N/2}} = \omega_{N/2}^{-k\ell}$$

$$= \underbrace{\sum_{\ell=0}^{N/2-1} f_0[\ell]\,\omega_{N/2}^{-k\ell}}_{\substack{= \mathcal{D}_{N/2}\{\mathbb{f}_0\}[k] \\ N/2 \text{ pt DFT} \\ =: F_0[k]}} + \omega_N^{-k} \underbrace{\sum_{\ell=0}^{N/2-1} f_1[\ell]\,\omega_{N/2}^{-k\ell}}_{\substack{= \mathcal{D}_{N/2}\{\mathbb{f}_1\}[k] \\ N/2 \text{ pt DFT} \\ =: F_1[k]}}, \quad k = 0, 1, \cdots, N-1$$

Now, note that $\mathbb{F}_0$ & $\mathbb{F}_1$ are both $N/2$-periodic.

Hence we only need to record
$F_0[0] \sim F_0[\frac{N}{2}-1]$ and $F_1[0] \sim F_1[\frac{N}{2}-1]$
i.e., N complex numbers!

In fact,

This is called ⇒ the butterfly relation

$$F[k] = \begin{cases} F_0[k] + \omega_N^{-k} F_1[k], & k = 0, 1, \cdots, \frac{N}{2}-1. \\ F_0[k-\frac{N}{2}] - \omega_N^{-(k-\frac{N}{2})} F_1[k-\frac{N}{2}], & k = \frac{N}{2}, \cdots, N-1. \end{cases}$$

$\because$ $F_0[k-\frac{N}{2}] + \omega_N^{-k} F_1[k-\frac{N}{2}]$, $k = \frac{N}{2}, \cdots, N-1$

$$= F_0[k-\tfrac{N}{2}] + \omega_N^{-(k-\frac{N}{2})} \underbrace{\boxed{\omega_N^{-\frac{N}{2}}}}_{=-1} F_1[k-\tfrac{N}{2}]$$

$$= F_0[k-\tfrac{N}{2}] - \omega_N^{-(k-\frac{N}{2})} F_1[k-\tfrac{N}{2}] \quad ///$$

This butterfly op. can also be written as

$$\begin{cases} F[k] = F_0[k] + \boxed{\omega_N^{-k} F_1[k]} & k = 0, 1, \cdots, \frac{N}{2}-1 \\ F[k+\frac{N}{2}] = F_0[k] - \boxed{\omega_N^{-k} F_1[k]} & \text{same !} \end{cases}$$

↳ requires 1 complex mult. + 2 complex additions.

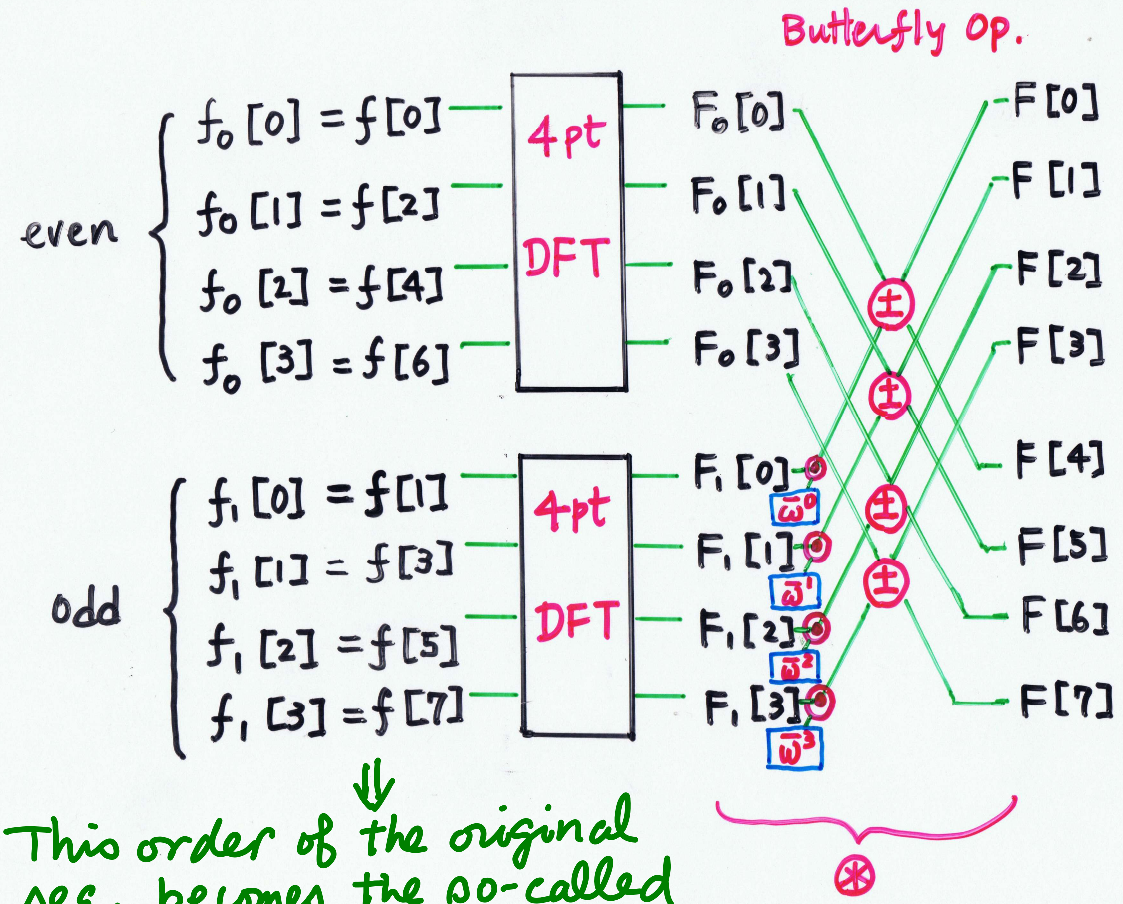
- This split procedure is repeated recursively.

- at the bottom level $n-1$:

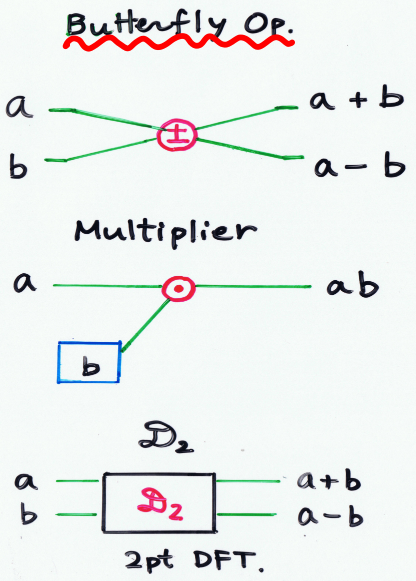
$$\omega_2 = e^{\frac{2\pi i}{2}} = -1. \qquad \begin{cases} F[0] = f[0] + \omega_2^0 f[1] = f[0] + f[1] & \text{sum} \\ F[1] = f[0] + \omega_2^1 f[1] = f[0] - f[1] & \text{difference} \end{cases}$$

Let's look at an 8-pt FFT as an example!

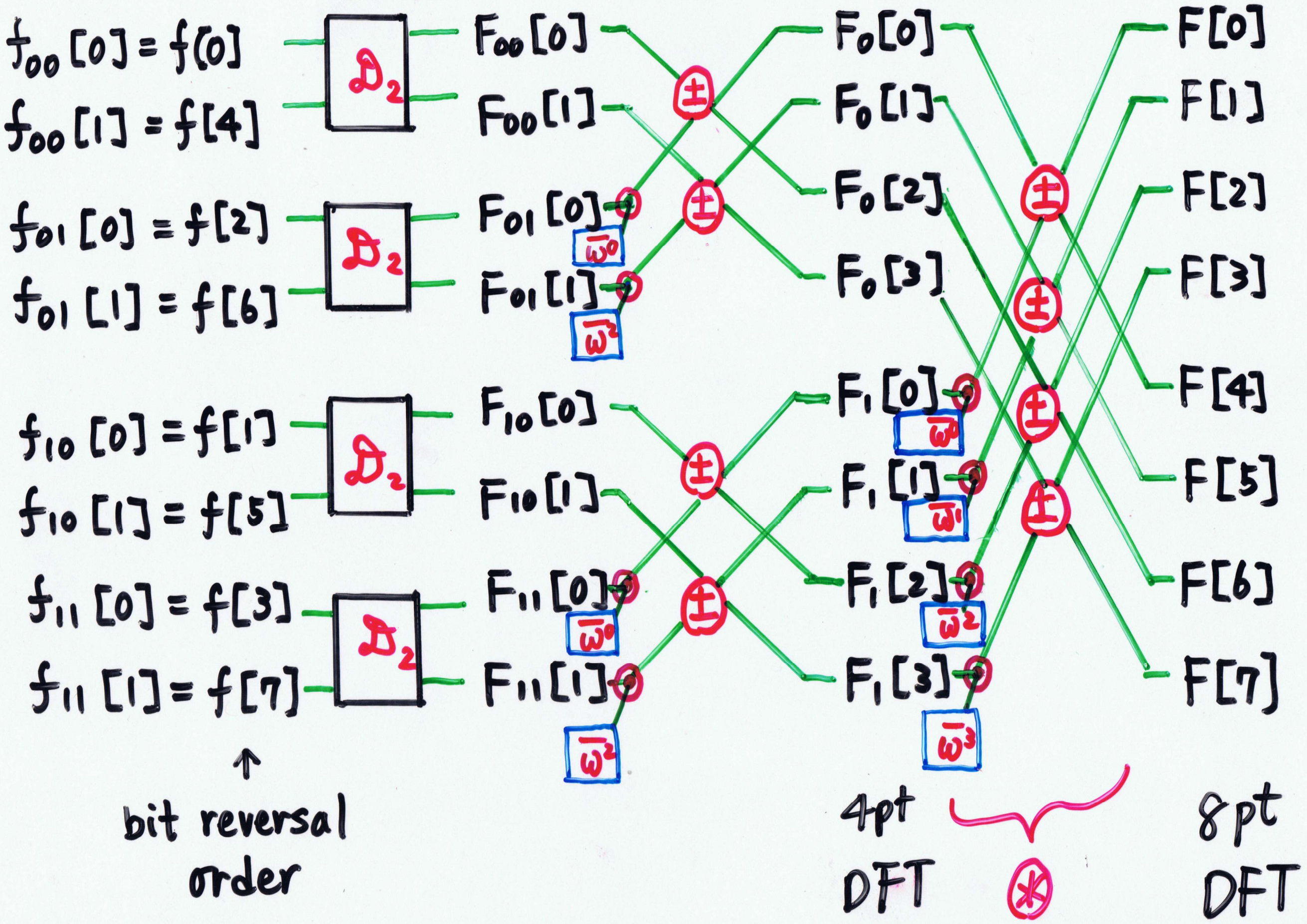
1 level FFT (N = 8)



This order of the original seq. becomes the so-called bit-reversal order if we go down to the bottom level $n-1$.



Full level FFT (N=8)



$\omega_4^1 = \omega_8^2$

- The essence of the Cooley-Tukey alg. (1965)

$\left\{ \begin{array}{l} \text{reordering stage (even-odd, bit reversal)} \\ \text{combine stage (butterfly op's)} \end{array} \right.$

This is a bottom-up recursive procedure:

$$2\text{pt} \to 4\text{pt} \to \cdots \to 2^{n-1}\text{pt} \to 2^{n}\text{pt}$$

Bit Reversal Operation

- If $\ell = b_{m-1} b_{m-2} \cdots b_0$ (binary expansion of ℓ), then $\bar{\ell} = b_0 b_1 \cdots b_{m-2} b_{m-1}$ is called the bit-reversed number of ℓ.

- If $f[\ell] = f[b_{m-1} \cdots b_0]$, then $\mathcal{D}_2$ op. at the bottom level is done between $f_{b_0 b_1 \cdots b_{m-2}}[0]$ & $f_{b_0 b_1 \cdots b_{m-2}}[1]$.

ℓ	$b_2 b_1 b_0$	$b_0 b_1 b_2$	$\bar{\ell}$	$\mathcal{D}_2$ pair
0	0 0 0	0 0 0	0	$f_{00}[0]$
1	0 0 1	1 0 0	4	$f_{10}[0]$
2	0 1 0	0 1 0	2	$f_{01}[0]$
3	0 1 1	1 1 0	6	$f_{11}[0]$
4	1 0 0	0 0 1	1	$f_{00}[1]$
5	1 0 1	1 0 1	5	$f_{10}[1]$
6	1 1 0	0 1 1	3	$f_{01}[1]$
7	1 1 1	1 1 1	7	$f_{11}[1]$

* The operation counts of the DIT alg.

- $\exists \log_2 N = n$ levels
- at each level, need $N/2$ butterfly op.

  1 $\mathbb{C}$-multi + 2 $\mathbb{C}$-add.'s
  ″
- Hence,

$$\#\text{ ops.} = \left.\begin{array}{l} \frac{N}{2}\log_2 N \ (\mathbb{C}\text{-multi.'s}) \\ + \ N\log_2 N \ (\mathbb{C}\text{-add.'s}) \end{array}\right\} \approx O(N\log_2 N)$$

☆ FFT via Matrix Factorization

The whole thing can be viewed as a clever matrix factorization!

(*) $$\mathbb{F} = W_N^{*} \mathbb{f} = \left[\begin{array}{c|c} I_{N/2} & \Omega_{N/2} \\ \hline I_{N/2} & -\Omega_{N/2} \end{array}\right] \left[\begin{array}{c} \mathbb{f}_0 \\ \hline \mathbb{f}_1 \end{array}\right]$$

where ↪ represents the butterfly op's.

$$\Omega_m := \mathrm{diag}\left(\omega_{2m}^{0}, \omega_{2m}^{-1}, \cdots, \omega_{2m}^{-(m-1)}\right) \in \mathbb{C}^{m\times m}.$$

(**) $$\begin{cases} \mathbb{F}_0 = W_{N/2}^{*} \mathbb{f}_0 = \left[\begin{array}{c|c} I_{N/4} & \Omega_{N/4} \\ \hline I_{N/4} & -\Omega_{N/4} \end{array}\right] \left[\begin{array}{c} \mathbb{f}_{00} \\ \hline \mathbb{f}_{01} \end{array}\right] \\ \mathbb{F}_1 = W_{N/2}^{*} \mathbb{f}_1 = \left[\begin{array}{c|c} I_{N/4} & \Omega_{N/4} \\ \hline I_{N/4} & -\Omega_{N/4} \end{array}\right] \left[\begin{array}{c} \mathbb{f}_{10} \\ \hline \mathbb{f}_{11} \end{array}\right] \end{cases}$$

Combining (*) & (**), we have:

$$\mathbb{F} = \left[\begin{array}{c|c} I_{N/2} & \Omega_{N/2} \\ \hline I_{N/2} & -\Omega_{N/2} \end{array}\right] \left[\begin{array}{cc|c} I_{N/4} & \Omega_{N/4} & \\ I_{N/4} & -\Omega_{N/4} & 0 \\ \hline & & I_{N/4} \quad \Omega_{N/4} \\ 0 & & I_{N/4} \ -\Omega_{N/4} \end{array}\right] \begin{bmatrix} \mathbb{f}_{00} \\ \mathbb{f}_{01} \\ \mathbb{f}_{10} \\ \mathbb{f}_{11} \end{bmatrix}$$

This can be further repeated recursively.

Let $B_{2m} := \left[\begin{array}{c|c} I_m & \Omega_m \\ \hline I_m & -\Omega_m \end{array}\right]$, then $\longrightarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ 2×2

$$\mathbb{F} = B_N \underbrace{\begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \begin{bmatrix} B_{\frac{N}{4}} & 0 & 0 & 0 \\ 0 & B_{\frac{N}{4}} & 0 & 0 \\ 0 & 0 & B_{\frac{N}{4}} & 0 \\ 0 & 0 & 0 & B_{\frac{N}{4}} \end{bmatrix} \cdots \begin{bmatrix} \boxed{B_2} & 0 & \cdots & 0 \\ 0 & B_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & B_2 \end{bmatrix}}_{\text{almost diagonal!}} \underset{\substack{\uparrow \\ \text{bit-} \\ \text{reversal} \\ \text{op.}}}{\underset{\sim}{P}^{T}} \mathbb{f}$$

- How about $\mathcal{D}_N^{-1}$? (inverse FFT)

⇒ Can use the same routine of the forward FFT.

<u>Why?</u>

$$f[\ell] = \frac{1}{N}\sum_{k=0}^{N-1} F[k]\,\omega_N^{k\ell}$$ ← Note 1/N factor coming from the def. of $\mathcal{D}_N$ for this lecture.

$$\Leftrightarrow \quad N\,\overline{f[\ell]} = \sum_{k=0}^{N-1} \overline{F[k]}\,\omega_N^{-k\ell}$$

$$\Leftrightarrow \quad f[\ell] = \frac{1}{N}\,\overline{\left[\sum_{k=0}^{N-1} \overline{F[k]}\,\omega_N^{-k\ell}\right]}$$

$$\Leftrightarrow \quad \mathbf{f} = \frac{1}{N}\,\overline{\mathrm{FFT}(\overline{\mathbf{F}})}$$

///

☆ <u>FFT of a single ℝ-valued vector</u>

The FFT alg. we have discussed so far is for a 1D array of complex numbers.

Can we speed up the computation of DFT for a single ℝ-valued vector?

⇒ Yes! But how?

⇒ Split it into even & odd indices as

$$g[\ell] = f[2\ell] + i f[2\ell+1], \quad \ell = 0, 1, \cdots, \tfrac{N}{2}-1.$$

↓ FFT of length N/2

$$G[k] = \underbrace{F_0[k]}_{\in\,\mathbb{C}} + i\,\underbrace{F_1[k]}_{\in\,\mathbb{C}}, \quad k = 0, 1, \cdots, \tfrac{N}{2}-1.$$

$$F_0[k] = \sum_{\ell=0}^{N/2-1} f[2\ell]\,\omega_{N/2}^{-k\ell}, \quad F_1[k] = \sum_{\ell=0}^{N/2-1} f[2\ell+1]\,\omega_{N/2}^{-k\ell}$$

Can we reconstruct $\boldsymbol{F} \in \mathbb{C}^N$ from $\boldsymbol{G} \in \mathbb{C}^{N/2}$?

Yes! To do so, check:

$$\mathcal{D}_N\{\boldsymbol{f}\}[k] = F[k] = \sum_{\ell=0}^{N-1} f[\ell]\,\omega_N^{-k\ell}$$

$$= \sum_{\ell=0}^{N/2-1} f[2\ell]\,\omega_{N/2}^{-k\ell} + \omega_N^{-k} \sum_{\ell=0}^{N/2-1} f[2\ell+1]\,\omega_{N/2}^{-k\ell}$$

$$= F_0[k] + \omega_N^{-k} F_1[k], \quad k = 0, 1, \cdots, N-1$$

But, F_0 & F_1 are $N/2$-periodic!

So, $G[\frac{N}{2}-k] = \overline{F_0[k]} + i\,\overline{F_1[k]}$

since $F_j[\frac{N}{2}-k] = F_j[-k] = \overline{F_j[k]}, \; j = 0, 1.$

So, $F[k] = F_0[k] + \omega_N^{-k} F_1[k]$

$$= \frac{1}{2}\left\{G[k] + \overline{G[\tfrac{N}{2}-k]}\right\} - \frac{i}{2}\left\{G[k] - \overline{G[\tfrac{N}{2}-k]}\right\}\omega_N^{-k} \quad (*)$$

$k = 0, 1, \cdots, N-1$

Note $\{G[k]\}_{k=0}^{N/2-1}$ is $N/2$-periodic.

Remark

If $\boldsymbol{f} \in \mathbb{R}^N$, then

$F[0] =$ the DC comp. $\in \mathbb{R}$

$F[\frac{N}{2}] =$ the Nyquist comp. $\in \mathbb{R}$

the highest freq. comp.

So we can pack $F[0] \leftarrow F[0] + i F[\frac{N}{2}] \in \mathbb{C}$.

$F[k] \in \mathbb{C}$, $k = 1, \cdots, N/2 - 1$. No need to store $F[k]$ for $k = N/2+1, \cdots, N-1$ thanks to the symmetry.

$\Longrightarrow$ $N/2$ $\mathbb{C}$ numbers as the output!