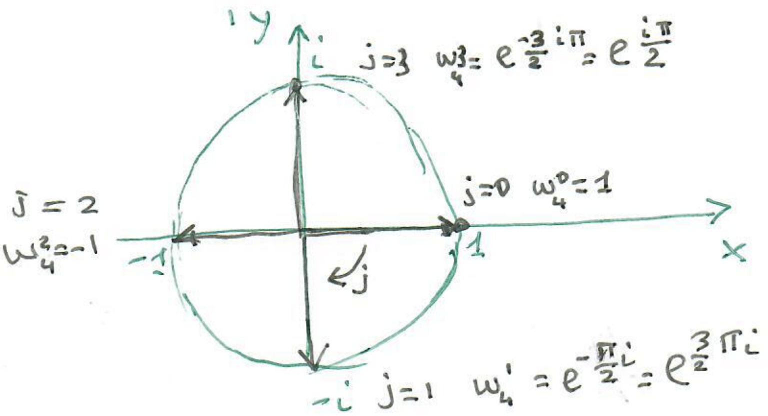
FFT: the basic Cooley-Tukey algorithm.

Preliminaries. Let $\omega_N = e^{-\frac{2\pi i}{N}}$ be the basic root N^{th} root of one, then ω_N^j for $j = 0, 1 \dots N-1$ are all distinct N^{th} roots of 1:

Indeed $(\omega_N^j)^N = e^{-\frac{2\pi i}{N} jN} = e^{-2\pi i j} = 1$

Example $N = 4$



We have the important property

$$\boxed{\omega_N^{j+N} = \omega_N^j}$$

Given a sequence of N values $\{x_0, \dots x_{N-1}\}$ with $x_j \in \mathbb{C}$ (in general) its discrete Fourier transform (DFT) is given by the sequence $\{y_0 \dots y_{N-1}\}$ of complex numbers given by

$$y_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} x_n \omega_N^{nk}$$

The FFT allow the computation of the DFT (and its inverse) with a number of operations of the order $N \log_2(N)$.
It is based on a <u>recursive decomposition</u> of the terms.

Here I present the basic algorithm, rediscovered by Cooley and Tukey in 1965, but in fact discovered more than 150 years before by Carl Friedrich Gauss!

For simplicity, we illustrate the algorithm for the case of N power of 2. The generalization to arbitrary $N>0$ is possible but it introduces additional complexity.

Let $j=2m$ and $j=2m+1$, for $m=0,1,\dots N/2$ the even and odd indexes. We may decompose the original DFT into

$$(*)\ Y_k = \sum_{m=0}^{N/2-1} x_{2m}\, e^{\frac{-2\pi i}{N/2} mk} + \sum_{m=0}^{N/2-1} x_{2m+1}\, e^{\frac{-2\pi i}{N}(2m+1)k} =$$

$$= \sum_{m=0}^{N/2-1} x_{2m}\, e^{\frac{-2\pi i}{N/2} mk} + e^{\frac{-2\pi i}{N}k} \sum_{m=0}^{N/2-1} e^{\frac{-2\pi i}{N/2} mk}\, x_{2m+1} =$$

$$= \underbrace{\boxed{\sum_{m=0}^{N/2-1} x_{2m}\, W_{N/2}^{mk}}}_{E_k} + W_N^k \underbrace{\boxed{\sum_{m=0}^{\frac{N}{2}-1} x_{2m+1}\, W_{N/2}^{mk}}}_{O_k}$$

We may now note that for $k=0,\dots \frac{N}{2}-1$ E_k and O_k define the DFF on the reduced sequences $\{x_{2m}, m=0,\dots N/2-1\}$ and $\{x_{2m+1}, m=0,\dots, N/2-1\}$. each of $N/2$ elements! Therefore, I may repeate the argumente recursively (until I arrive to empty sets).

However I need Y_k also for $k=\frac{N}{2},\dots N-1$! I can exploit that: For $k=0,\dots\frac{N}{2}-1$ I have using

$$Y_{k+\frac{N}{2}} = \sum_{m=0}^{\frac{N}{2}-1} x_{2m}\, W_{N/2}^{mk}\, W_{\frac{N}{2}}^{m\frac{N}{2}} + W_N^k\, W_N^{N/2} \sum_{m=0}^{\frac{N}{2}-1} x_{2m-1}\, W_{\frac{N}{2}}^{mk}\, W_{\frac{N}{2}}^{m\frac{N}{2}}$$

Now $W_{N/2}^{m\frac{N}{2}} = e^{-2\pi i m} = 1$! while $W_N^{N/2} = e^{-\pi i} = -1$!!

POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECN

In conclusion we have

$$y_k = E_k + w_N^k O_k \qquad k=0, \dots, N/2 - 1$$

$$y_{k+\frac{N}{2}} = E_k - w_N^k O_k$$

with E_k and O_k that are nothing else than the DFT terms on the subset of dimension N/2 formed by the even and odd-indexed components x_j.

By recursing, I reduce each time by half the number of component, until I reach the zero set and the algorith stops. It may be proved that the number of operation needed to compute the y_k is of order $\Theta(N \log_2 N)$.

The ~~FFT~~ for the inverse DFT follows the same steps.

POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECNICO DI MILANO POLITECN