From the Fourier Transform to the DFT and FFT

1. Fourier transform → DFT

The Fourier transform represents a continuous signal $x(t)$ by its frequency components:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t}\, dt.$$

In numerical applications we sample the signal at

$$t_n = n\Delta t, \qquad n = 0, \ldots, N-1,$$

and work with the finite sequence $x_n = x(t_n)$.

The discrete Fourier transform (DFT) is

$$\boxed{X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N}}, \qquad k = 0, \ldots, N-1.$$

Its inverse is

$$\boxed{x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{2\pi i k n/N}}.$$

Thus the basic transition is

$$\text{continuous signal} \quad \longrightarrow \quad \text{sampled finite sequence} \quad \longrightarrow \quad \text{DFT}.$$

A direct evaluation of all N values X_k uses about N^2 operations:

$$\boxed{\text{direct DFT: } O(N^2).}$$

2. DFT → FFT

The FFT is not a different transform. It is an efficient algorithm for computing the same DFT.

For even N, split the DFT sum into even and odd indices:

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-2\pi i k m/(N/2)} + e^{-2\pi i k/N} \sum_{m=0}^{N/2-1} x_{2m+1} e^{-2\pi i k m/(N/2)}.$$

Define the two $(N/2)$-point DFTs

$$E_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-2\pi i k m/(N/2)},$$

$$O_k = \sum_{m=0}^{N/2-1} x_{2m+1} e^{-2\pi i k m/(N/2)}.$$

Then

$$X_k = E_k + W_N^k O_k, \qquad X_{k+N/2} = E_k - W_N^k O_k,$$

where

$$W_N = e^{-2\pi i/N}.$$

These two operations form the basic FFT *butterfly*.

The crucial point is that E_k and O_k are themselves smaller DFTs. We repeat the same even/odd splitting:

$$N \longrightarrow \frac{N}{2} \longrightarrow \frac{N}{4} \longrightarrow \cdots \longrightarrow 1.$$

For the radix-2 FFT, $N = 2^p$, so there are

$$p = \log_2 N$$

levels. Each level requires $O(N)$ operations, giving

$$\boxed{\text{FFT: } O(N \log N).}$$

3. Essential picture

$$\boxed{\text{Fourier transform} \xrightarrow{\text{sampling}} \text{DFT} \xrightarrow{\text{even/odd decomposition}} \text{FFT}.}$$

The Fourier transform is the continuous mathematical transform; the DFT is its finite, sampled version; and the FFT is a fast way of evaluating the DFT. The main idea of the FFT is to reuse the results of many smaller DFTs instead of computing all N^2 terms independently.