Discrete Fourier Transform to Fast Fourier Transform

What is FFT?

Did you know that Fast Fourier Transform algorithm could have stopped the nuclear arms race if it was (re)discovered earlier? Regardless of this misopportunity, this algorithm is still arguably one of the most powerful tools ever invented. Every form of digital signal processing you can think of: wifi, 5G, sonar, radar, LiDar, the music you listen to, and the last YouTube video you watched, all rely on FFT. In other words, FFT is the strongest bridge between time and frequency domain that we have.

However, to gain appreciation for FFT, we have to understand how other candidates don't compare. First of all, it's impossible for our machines to collect and store infinitely continuous data. This fact alone makes Fourier Transform (continuous in both time and frequency domain) and Discrete-Time Fourier Transform (discrete time, but continuous in frequency domain) impractical. A step up from the Discrete-Time Fourier Transform (DTFT) is the Discrete Fourier Transform (DFT), which leads to our second problem. That is DFT requires too many calculations.

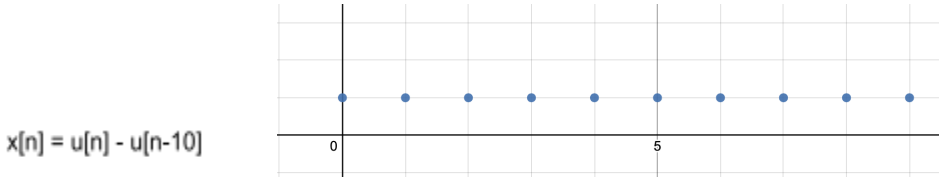
How does DFT works?

Before exploring the beauty of FFT, let's talk about its origin, DFT. DFT takes in finite number of samples in the time domain and splits out the result in frequency domain.

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-jk\omega_0 n} \quad where\ x[n],\ X[n]\ are\ of\ length\ N\ and\ \omega_0 = \frac{2\pi}{N}$$ [1]

$x[n]$ is the sample of our signal in time domain. $X[k]$ is our sample in the frequency domain. The number of samples N in time and frequency domain should be the same. ω_0 is the fundamental frequency of the signal and so, $\omega_0 k$ gives the different range of frequencies. The DFT itself is multiplying every sample with different $\omega_0 k$ – looking at every sample at each frequency. The summation tells us how much each frequency appears in the original signal.

For a better understanding, let's look at an example[2]. Suppose we have $x[n] = u[n] - u[n-10]$ and we would like to find $X[k]$.

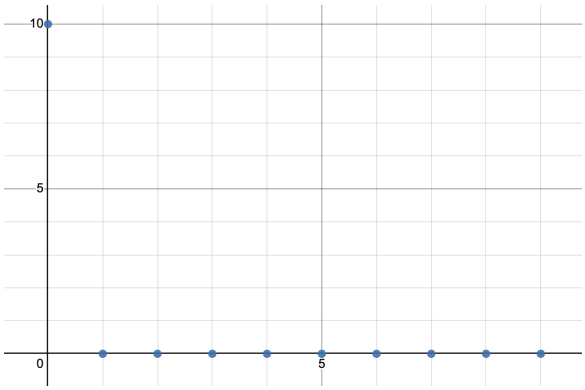


[1] https://www.princeton.edu/~cuff/ele201/kulkarni_text/frequency.pdf
[2] Prof. Achuta Kadambi's ECE 113 lecture notes

$$X[k] = \sum_{n=0}^{9} x[n]e^{-jk\frac{2\pi}{10}n} = \sum_{n=0}^{9} (u[n] - u[n-10])e^{-jk\frac{2\pi}{10}n} = \sum_{n=0}^{9} \left(e^{-jk\frac{2\pi}{10}}\right)^n = \frac{1-e^{-jk\frac{2\pi}{10}(10)}}{1-e^{-jk\frac{2\pi}{10}}}$$

$$X[0] = \lim_{k \to 0} \frac{1-e^{-j2\pi k}}{1-e^{-jk\frac{2\pi}{10}}} = 10$$

$$X[1] = X[2] = X[3] = \ldots = X[9] = 0$$



The calculation and the plot show that there are 10 samples with zero frequency and zero sample in other frequency ranges (ω_0, $2\omega_0$, $3\omega_0$..., $9\omega_0$) which matches with our original sample. Now, let's think about the complexity of DFT. In this particular example, there are 10 different frequencies. And this expression $\sum_{n=0}^{9} x[n]e^{-jk\frac{2\pi}{10}n}$ shows that each frequency is being multiplied by all 10 samples. This small example alone requires 10 x 10 = 100 complex multiplications! Hence, DFT complexity is $O\left(n^2\right)$. One thousand samples would require one million operations.

During the early years of nuclear arms race, we were using DFT. Detecting a day worth of underground nuclear testing would take three years.

How fast is FAST Fourier Transform?

FFT cuts down the number of computations by eliminating all the redundant operations. This is done by looking at even and odd indexes separately when multiplying them with a sine and cosine wave (complex exponential). Take samples at even indexes as an example, notice in figure 3[3] that if we put on top of each other the first half and the second half of the even samples, we can see that at every location that there is a sample, the result of the multiplication is the same. This means that the calculation of the second half is not required. Similar pattern happens with odd indexes. The only difference is that the sign is flipped. A simple sign changed takes zero flop. This initial round alone reduces the calculation by half.[4]

[3] showing multiplication of cosine wave at different frequencies with the samples. Will insert the figure later

[4] The Remarkable Story Behind The Most Important Algorithm Of All Time

The key to FFT is that this process can be repeated until there is only one sample left. The complexity is reduced to $O\left(Nlog_2N\right)$. One thousand samples using DFT would need one million complex multiplications, but with FFT it only takes around 9900.

There is one problem with FFT. Since the method relies entirely on the fact that total leftover samples are always divisible by 2, we have to make sure that the total number of samples is in the form 2^p where p is an integer. To achieve this, zero padding is required. It means adding zero-valued samples to the signal.

Citation:

The Remarkable Story Behind The Most Important Algorithm Of All Time

https://www.princeton.edu/~cuff/ele201/kulkarni_text/frequency.pdf

Prof. Achuta Kadambi's ECE 113 lecture notes