An introduction to discrete Fourier transforms

Isao Sasano

This document is largely based on the reference book [1] with some parts slightly changed.

1 Discrete Fourier Transform (DFT)

Let $f(x)$ be a periodic function of period 2π. Assume that $f(x)$ is given only in terms of values at the following N points on the range $[0, 2\pi]$:

$$x_l = \frac{2\pi l}{N} \qquad (l = 0, 1, \dots, N-1). \tag{1}$$

We say that $f(x)$ is being **sampled** at these points. We now would like to find a linear combination of complex exponential functions $\{e^{ikx} | 0 \le k \le N-1\}$

$$\sum_{k=0}^{N-1} F_k e^{ikx}$$

that **interpolates** $f(x)$ at the nodes (1).

$$f(x_l) = \sum_{k=0}^{N-1} F_k e^{ikx_l} \qquad (l = 0, 1, \dots, N-1)$$

Let $f_l = f(x_l)$. Then we would like to find the coefficients $F_0, \dots, F_{N-1}$ such that the following equation holds.

$$f_l = \sum_{k=0}^{N-1} F_k e^{ikx_l} \qquad (l = 0, 1, \dots, N-1) \tag{2}$$

We multiply the both sides of the equation (2) by e^{-imx_l}, where $0 \le m \le N-1$, and sum over l from 0 to $N-1$.

$$\begin{aligned}
\sum_{l=0}^{N-1} f_l e^{-imx_l} &= \sum_{l=0}^{N-1}\sum_{k=0}^{N-1} F_k e^{ikx_l} e^{-imx_l} \\
&= \sum_{l=0}^{N-1}\sum_{k=0}^{N-1} F_k e^{i(k-m)x_l} \\
&= \sum_{l=0}^{N-1}\sum_{k=0}^{N-1} F_k e^{i(k-m)2\pi l/N} \\
&= \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} F_k e^{i(k-m)2\pi l/N} \\
&= \sum_{k=0}^{N-1} F_k \sum_{l=0}^{N-1} e^{i(k-m)2\pi l/N}
\end{aligned}$$

Let $r = e^{i(k-m)2\pi/N}$. Then

$$e^{i(k-m)2\pi l/N} = \left(e^{i(k-m)2\pi/N}\right)^l = r^l.$$

So the above sum is written as follows.

$$\sum_{l=0}^{N-1} f_l e^{-imx_l} = \sum_{k=0}^{N-1} F_k \sum_{l=0}^{N-1} r^l$$

When $k = m$, we have $r = e^0 = 1$, so the sum $\sum_{l=0}^{N-1} r^l$ is calculated as follows.

$$\sum_{l=0}^{N-1} r^l = \sum_{l=0}^{N-1} 1 = N$$

When $k \ne m$, we have $r \ne 1$, so the sum $\sum_{l=0}^{N-1} r^l$ is calculated as follows

$$\sum_{l=0}^{N-1} r^l = \frac{1-r^N}{1-r} = 0,$$

since

$$r^N = (e^{i(k-m)2\pi/N})^N = e^{i(k-m)2\pi} = 1.$$

So we obtain the following equality.

$$F_k \sum_{l=0}^{N-1} r^l = \begin{cases} F_m N & k = m \\ 0 & k \neq m \end{cases}$$

So we obtain

$$\sum_{k=0}^{N-1} F_k \sum_{l=0}^{N-1} r^l = F_m N.$$

Since $\sum_{l=0}^{N-1} f_l e^{-imx_l} = \sum_{k=0}^{N-1} F_k \sum_{l=0}^{N-1} r^l$ we obtain

$$\sum_{l=0}^{N-1} f_l e^{-imx_l} = F_m N.$$

By dividing by N we obtain

$$F_m = \frac{1}{N} \sum_{l=0}^{N-1} f_l e^{-imx_l}.$$

By writing k for m we obtain

$$F_k = \frac{1}{N} \sum_{l=0}^{N-1} f_l e^{-ikx_l} = \frac{1}{N} \sum_{l=0}^{N-1} f_l e^{-i2\pi kl/N} \qquad k = 0, \ldots, N-1 \tag{3}$$

since $x_l = \frac{2\pi l}{N}$. The sequence $F_0, \ldots, F_{N-1}$ is called the **discrete Fourier transform** of the given signal $f_0, \ldots, f_{N-1}$.

Let $\omega = e^{2\pi i/N}$. Note that $= e^{2\pi i/N}$ is a primitive N-th root of 1. Then $e^{-i2\pi kl/N} = \omega^{-lk}$, so

$$F_k = \frac{1}{N} \sum_{l=0}^{N-1} f_l \omega^{-lk} \qquad k = 0, \ldots, N-1.$$

Then the discrete Fourier transform is written in matrix form as follows.

$$\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{pmatrix} = \frac{1}{N} \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(N-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \cdots & \omega^{-(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{pmatrix}$$

Note that the element of l-th row and k-th column in the matrix is ω^{-lk}.

By the formula (2), we obtain

$$f_l = \sum_{k=0}^{N-1} F_k e^{ikx_l} = \sum_{k=0}^{N-1} F_k e^{i2\pi kl/N} = \sum_{k=0}^{N-1} F_k \omega^{lk} \qquad (l = 0, 1, \ldots, N-1), \quad (4)$$

which gives the transformation from the sequence $F_0, \ldots, F_{N-1}$ to the sequence $f_0, \ldots, f_{N-1}$. It is called the **inverse discrete Fourier transform**. The inverse discrete Fourier transform is written in matrix form as follows.

$$\begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{pmatrix} = \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{1} & \omega^{2} & \cdots & \omega^{N-1} \\ \omega^0 & \omega^{2} & \omega^{4} & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{(N-1)} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{pmatrix}$$

The inverse discrete Fourier transform of the discrete Fourier transform of a given signal is the signal itself, since the following equation holds.

$$\begin{aligned} &\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{1} & \omega^{2} & \cdots & \omega^{N-1} \\ \omega^0 & \omega^{2} & \omega^{4} & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{(N-1)} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}^{-1} \\ &= \frac{1}{N} \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(N-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \cdots & \omega^{-(N-1)(N-1)} \end{pmatrix} \end{aligned}$$

We do not prove this equation. Refer to textbooks like [1]. Note that A^{-1} represents the inverse matrix of A.

Example: the case for $N = 4$.
Calculate the discrete Fourier transform of the following signal.

$$\boldsymbol{f} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

Since $N = 4$, $\omega = e^{2\pi i/4} = e^{\pi i/2} = i$ and thus $\omega^{-lk} = i^{-lk}$. So the discrete Fourier transform of $\boldsymbol{f}$ is calculated as follows.

$$\begin{aligned} \frac{1}{4}\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \omega^{-3} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \omega^{-6} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \omega^{-9} \end{pmatrix}\boldsymbol{f} &= \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}\begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} \\ &= \frac{1}{4}\begin{pmatrix} f_0 + f_1 + f_2 + f_3 \\ f_0 - if_1 - f_2 + if_3 \\ f_0 - f_1 + f_2 - f_3 \\ f_0 + if_1 - f_2 - if_3 \end{pmatrix} \end{aligned}$$

Exercise 14-2 Calculate the discrete Fourier transform of the following sequence.

$$\boldsymbol{f} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 4 \\ 9 \end{pmatrix}$$

2 Fast Fourier Transform (FFT)

The discrete Fourier transform is just a multiplication of a matrix to the given sequence of signal. Naively computing the matrix multiplication requires $O(N^2)$ operations. However, when $N = 2^n$ for some natural numer $n > 1$, the discrete Fourier transform can be done by the **fast Fourier transform (FFT)**, which needs only $O(N \log_2 N)$ operations. FFT utilizes some specific properties of the matrix.

In computing the discrete Fourier transform and the inverse discrete Fourier transform, it is essential to compute the sequence $b_0, \ldots, b_{N-1}$ from any sequence $a_0, \ldots, a_{N-1}$ as follows.

$$b_k = \sum_{l=0}^{N-1} a_l \omega^{kl} \qquad k = 0, \ldots, N-1 \tag{5}$$

Let's check this. In order to compute the inverse discrete Fourier transform $f_0, \ldots, f_{N-1}$ of the sequence $F_0, \ldots, F_{N-1}$ following (4), we set $a_k = F_k$ in the equation (5) so that we obtain $f_l = b_l$.

In order to compute the discrete Fourier transform $F_0, \ldots, F_{N-1}$ of the sequence $f_0, \ldots, f_{N-1}$, we rewrite the formula (3) as follows.

$$F_k = \frac{1}{N} \sum_{l=0}^{N-1} f_l \omega^{-kl} = \frac{1}{N} \overline{\sum_{l=0}^{N-1} \overline{f_l} \omega^{kl}}$$

Note that $\overline{z}$ is called the complex conjugate of z, defined as follows.

$$\overline{a + bi} = a - bi$$

We can show the above equation by transforming RHS to LHS as follows.

$$\begin{aligned}
\text{RHS} &= \frac{1}{N}\overline{\sum_{l=0}^{N-1}\overline{f_l}\omega^{kl}} \\
&= \frac{1}{N}\sum_{l=0}^{N-1}\overline{\overline{f_l}\omega^{kl}} \\
&= \frac{1}{N}\sum_{l=0}^{N-1}f_l\overline{\omega^{kl}} \\
&= \frac{1}{N}\sum_{l=0}^{N-1}f_l\overline{\omega}^{kl} \\
&\qquad \text{(since } \overline{z^n} = (\overline{z}^n) \text{ for any integer } n \\
&\qquad \text{ and any complex number } z) \\
&= \frac{1}{N}\sum_{l=0}^{N-1}f_l(\omega^{-1})^{kl} \qquad (\text{since } \overline{\omega} = \omega^{-1}) \\
&= \frac{1}{N}\sum_{l=0}^{N-1}f_l\omega^{-kl} \\
&= \text{LHS}
\end{aligned}$$

Then we set $a_l = \overline{f_l}$ in (5) so that we obtain $F_k = \dfrac{1}{N}\overline{b_k}$.

Now we consider the cases where N is a number that satisfies

$$N = 2^n$$

for some natural number $n > 1$. In these cases we can efficiently compute the discrete Fourier transform and the inverse discrete Fourier transform.

When N is an even number, the following equations hold.

$$\omega^{N/2} = -1, \omega^{N/2+1} = -\omega, \omega^{N/2+2} = -\omega^2, \ldots, \omega^{N-1} = -\omega^{N/2-1}$$

We show these equations. Since $\omega = e^{2\pi i/N}$, we obtain

$$\omega^{N/2} = (e^{2\pi i/N})^{N/2} = e^{i\pi} = -1$$

and hence

$$\omega^{N/2+k} = \omega^{N/2}\omega^k = -\omega^k.$$

In the following we write $\omega = e^{2\pi i/N}$ by parameterizing N as follows.

$$\omega_N = e^{2\pi i/N}$$

Then the following equation holds when N is an even number.

$$\omega_N^2 = \omega_{N/2}.$$

We show this as follows.

$$\omega_N^2 = (e^{2\pi i/N})^2 = e^{4\pi i/N} = e^{2\pi i/(N/2)} = \omega_{N/2}$$

By defining

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{N-1} x^{N-1} = \sum_{l=0}^{N-1} a_l x^l, \tag{6}$$

the formula (5) can be written as follows.

$$b_k = f(\omega_N^k) \qquad (k = 0, \ldots, N-1)$$

Note that $\omega = \omega_N$. So we obtain $b_0, \ldots, b_{N-1}$ by computing $f(\omega_N^0), \ldots, f(\omega_N^{N-1})$. Let us write this computation as $\mathrm{FFT}_N[f]$.

$$\mathrm{FFT}_N[f] = \{f(\omega_N^0), f(\omega_N^1), f(\omega_N^2), \ldots, f(\omega_N^{N-1})\}$$

where $f(\omega_N^0), f(\omega_N^1), f(\omega_N^2), \ldots, f(\omega_N^{N-1})$ represent the values to compute. The formula (6) can be rewritten as follows.

$$\begin{aligned} f(x) &= a_0 + a_2 x^2 + a_4 x^4 + \cdots + a_{N-2} x^{N-2} \\ &\quad + x(a_1 + a_3 x^2 + a_5 x^4 + \cdots + a_{N-1} x^{N-2}) \\ &= p(x^2) + x q(x^2) \end{aligned}$$

where $p(x)$ and $q(x)$ are defined as follows.

$$\begin{aligned} p(x) &= a_0 + a_2 x + a_4 x^2 + \cdots + a_{N-2} x^{N/2-1} \\ q(x) &= a_1 + a_3 x + a_5 x^2 + \cdots + a_{N-1} x^{N/2-1} \end{aligned}$$

Then $\mathrm{FFT}_N[x \mapsto p(x^2)]$ is as follows.

$$\mathrm{FFT}_N[x \mapsto p(x^2)] = \{p(1), p(\omega_N^2), p(\omega_N^4), \ldots, p(\omega_N^{2N-2})\}$$

Here it suffices to compute the first half of this sequence since the second half is the same as the first half.

$$\mathrm{FFT}_N[x \mapsto p(x^2)] = \{p(1), p(\omega_N^2), p(\omega_N^4), \ldots, p(\omega_N^{N-2})\}$$

Since $\omega_N^2 = \omega_{N/2}$, we obtain

$$\mathrm{FFT}_N[x \mapsto p(x^2)] = \{p(1), p(\omega_{N/2}), p(\omega_{N/2}^2), \ldots, p(\omega_{N/2}^{N/2-1})\}$$

and hence

$$\mathrm{FFT}_N[x \mapsto p(x^2)] = \mathrm{FFT}_{N/2}[p].$$

In the same way, we obtain

$$\mathrm{FFT}_N[x \mapsto q(x^2)] = \mathrm{FFT}_{N/2}[q].$$

Now note that the following equalities hold.

$$\begin{cases} f(\omega_N^k) &= p(\omega_{N/2}^k) + \omega_N^k q(\omega_{N/2}^k) \qquad k = 0, 1, \ldots, N/2-1 \\ f(\omega_N^{N/2+k}) &= p(\omega_{N/2}^k) - \omega_N^k q(\omega_{N/2}^k) \qquad k = 0, 1, \ldots, N/2-1 \end{cases} \tag{7}$$

This means that $f(\omega_N^k)$ for $k = 0, 1, 2, \ldots, N-1$ can be computed by using the results of $\mathrm{FFT}_{N/2}[p]$ and $\mathrm{FFT}_{N/2}[q]$. In other words, the computation $\mathrm{FFT}[f]$ can be decomposed into computations $\mathrm{FFT}_{N/2}[p]$, $\mathrm{FFT}_{N/2}[q]$, and the additions, subtractions, and multiplications in (7). Since $N = 2^n$ for some natural number $n > 1$, this decomposition can be done recursively. This is called the FFT.

Now let's show the computational complexity of the FFT. Suppose the number of addtions, subtractions, and multiplications in the computation of $\mathrm{FFT}_N[f]$ is $T(N)$. Then the number of addtions, subtractions, and multiplications in the computation of $\mathrm{FFT}_{N/2}[p]$ and $\mathrm{FFT}_{N/2}[p]$ is $T(N/2)$ respectively. Note that we can precompute the values of ω_N^k $(k = 0, \ldots, N-1)$ before starting the computation of $\mathrm{FFT}_N[f]$. The number of additions in (7) is N in total. The number of multiplications in (7) is $N/2 - 1$ in total, since we can reuse the result of multiplication $\omega_N^k q(\omega_{N/2}^k)$ in the first equation in (7) in the multiplication in the second equation in (7) and also the multiplication when $k = 0$ is not necessary because $\omega_n^0 = 1$. So the following equality holds.

$$T(N) = 2T(\frac{N}{2}) + N + \frac{N}{2} - 1 = 2T(\frac{N}{2}) + \frac{3}{2}N - 1$$

Note that $T(1) = 0$. Recall that $N = 2^n$ for some natural number $n > 1$. We let $t_n = T(2^n)$. Then

$$\begin{aligned} t_n &= 2t_{n-1} + \frac{3}{2} \cdot 2^n - 1 \\ t_0 &= 0 \end{aligned}$$

By subtracting both sides by 1 and then dividing by 2^n in the two equations above, we obtain

$$\begin{aligned} \frac{t_n - 1}{2^n} &= \frac{t_{n-1} - 1}{2^{n-1}} + \frac{3}{2} \\ \frac{t_0 - 1}{2^0} &= -1 \end{aligned}$$

This means that $\frac{t_n - 1}{2^n}$ constitutes the arithmetic sequence with the common difference $\frac{3}{2}$ and the initial number -1. So the following equality holds.

$$\frac{t_n - 1}{2^n} = \frac{3}{2}n - 1$$

Then we have

$$\begin{aligned} t_n &= 2^n(\frac{3}{2}n - 1) + 1 \\ &= N(\frac{3}{2}\log_2 N - 1) + 1 \\ &= O(N \log_2 N) \end{aligned}$$

FFT is implmented by some programming languages and provided as libraries.

A Some equations for complex numbers

Here we show some equations for complex numbers.

Theorem 1 *For any z_1, $z_2 \in \mathbb{C}$ the following equation holds.*

$$\overline{z_1 z_2} = \overline{z_1} \cdot \overline{z_2}$$

Proof Let $z_1 = a + bi$ and $z_2 = c + di$ where a, b, c, $d \in \mathbb{R}$. Then

$$\begin{aligned} \text{LHS} &= \overline{z_1 z_2} \\ &= \overline{(a+bi)(c+di)} \\ &= \overline{(ac-bd)+(ad+bc)i} \\ &= (ac-bd)-(ad+bc)i \end{aligned}$$

$$\begin{aligned} \text{RHS} &= \overline{z_1} \cdot \overline{z_2} \\ &= \overline{(a+bi)} \cdot \overline{(c+di)} \\ &= (a-bi)(c-di) \\ &= (ac-bd)-(ad+bc)i \end{aligned}$$

□

Theorem 2 *For any z_1, $z_2 \in \mathbb{C}$ the following equation holds.*

$$\overline{z_1 + z_2} = \overline{z_1} + \overline{z_2}$$

Proof Let $z_1 = a + bi$ and $z_2 = c + di$ where a, b, c, $d \in \mathbb{R}$. Then

$$\begin{aligned} \text{LHS} &= \overline{z_1 + z_2} \\ &= \overline{(a+bi+(c+di)} \\ &= \overline{(a+c)+(b+d)i} \\ &= (a+c)-(b+d)i \end{aligned}$$

$$\begin{aligned} \text{RHS} &= \overline{z_1} + \overline{z_2} \\ &= \overline{(a+bi)} + \overline{(c+d)i} \\ &= (a-bi)+(c-di) \\ &= (a+c)-(b+d)i \end{aligned}$$

□

References

[1] Erwin Kreyszig. *Advanced Engineering Mathematics.* John Wiley & Sons Ltd., tenth edition, 2011.