A BRIEF STUDY OF DISCRETE AND FAST FOURIER TRANSFORMS

AASHIRWAD VISWANATHAN ANAND

ABSTRACT. This paper studies the mathematical machinery underlying the Discrete and Fast Fourier Transforms, algorithmic processes widely used in quantum mechanics, signal analysis, options pricing, and other diverse fields. Beginning with the basic properties of Fourier Transform, we proceed to study the derivation of the Discrete Fourier Transform, as well as computational considerations that necessitate the development of a faster way to calculate the DFT. With these considerations in mind, we study the construction of the Fast Fourier Transform, as proposed by Cooley and Tukey [7].

Contents

1. History and Introduction

The subject of Fourier Analysis is concerned primarily with the representation of functions as sums of trigonometric functions, or, more generally, series of simpler periodic functions. Harmonic series representations date back to Old Babylonian mathematics (2000-1600 BC), where it was used to compute tables of astronomical positions, as discovered by Otto Neugebauer [8]. While, trigonometric series were first used by 18th Century mathematicians like d'Alembert, Euler, Bernoulli and Gauss, their applications were known only for periodic functions of known period. In 1807, Fourier was the first to propose that arbitrary functions could be represented by trigonometric series, in the article *Memoire sur la propagation de le chaleur dans les corps solides.* Gauss is credited with the first Fast Fourier Transform (FFT) implementation in 1805, during an interpolation of orbital measurements. However, the most commonly used FFT algorithm today is named after J.W. Cooley, an employee of IBM, and J.W. Tukey, a statistician, who jointly developed an implementation of the FFT for high speed computers in 1965 [1].

In this paper, we review briefly the theory of continuous Fourier transforms, but concern ourselves mainly with Discrete Fourier Transforms (DFTs), which are of great practical importance in the analysis of discrete signals and other data. The

DFT in its most general form, as we shall see, is inefficient for machine computation, requiring N^2 complex operations for a signal containing N samples. This motivates the development of the Fast Fourier Transforms, a family of efficient implementations of the DFT for different composites of N. In particular, we examine the radix-2 Cooley-Tukey FFT algorithm, which reduces the number of complex operations required to $N \log_2(N)$.

2. Overview of the Continuous Fourier Transform and Convolutions

We begin with a discussion on the continuous Fourier transform and the **Inversion** and **Convolution theorems**, which are important in understanding the relationship between the continuous Fourier transform and the DFT.

Definition 2.1. (Fourier Series) The **Fourier series** of a function f, piecewise continuous on $[-P, P]$ and having period $2P$ is defined as

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\frac{\pi}{P}t} \tag{2.2}$$

where

$$c_n = \frac{1}{2P}\int_{-P}^{P} f(t)e^{-in\frac{\pi}{P}t}dt^1 \tag{2.3}$$

The Fourier series represents the periodic function f as a sum of oscillations with frequencies $\frac{n\pi}{P}$ and complex amplitudes c_n. It is now possible to construct an integral expression that represents any function f (not necessarily periodic) as a sum of periodic functions. We define

$$\hat{f}(P, \omega) = \int_{-P}^{P} f(t)e^{-i\omega t}dt \tag{2.4}$$

Then, from equation (2.3), $c_n = \frac{1}{2P}\hat{f}(P, \frac{n\pi}{P})$, so that equation (2.2) may be rewritten as

$$f(t) = \frac{1}{2P}\sum_{n=-\infty}^{\infty} \hat{f}(P, \omega_n)e^{i\omega_n t} = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} \hat{f}(P, \omega_n)e^{i\omega_n t}\frac{\pi}{P}, \quad \omega_n = \frac{n\pi}{P} \tag{2.5}$$

Since $\Delta\omega_n = \omega_{n+1} - \omega_n = \frac{\pi}{P}$, the expression in Equation (2.5) becomes a Riemann sum. As $P \to \infty$, and defining

$$\hat{f}(\omega) = \lim_{P\to\infty} \hat{f}(P, \omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt, \quad \omega \in \mathbb{R} \tag{2.6}$$

the limiting process results in

$$f(t) \sim \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)e^{i\omega t}d\omega \tag{2.7}$$

[1] Here, and throughout the rest of this paper, $i = \sqrt{-1}$.

Remark 2.8. We have so far omitted details regarding the convergence of the integrals described. The following discussion shall be a little more explicit in this regard.

Definition 2.9. (Continuous Fourier Transform) For a function f on $\mathbb{R}$, such that

$$\int_{-\infty}^{\infty} |f(t)|dt < \infty \tag{2.10}$$

the integral $\hat{f}(\omega) = \int_{\mathbb{R}} f(t)e^{-i\omega t}dt$ converges absolutely and is defined as the **continuous Fourier transform** or **Fourier integral** of f.

Thus, any function on a finite interval $(-P, P)$ can be represented as a sum of harmonic oscillations with discrete frequencies $\{\omega_n : \omega_n = \frac{n\pi}{P}, n \in \mathbb{Z}\}$, and any function on the infinite interval $(-\infty, \infty)$ can be constructed from harmonic oscillations with a continuous frequency spectrum $\{\omega : \omega \in \mathbb{R}\}$, with the sum replaced by an integral.

Theorem 2.11. (Inversion Theorem) *Suppose that f satisfies Equation (2.10) and is continuous, except for a finite number of finite jumps in any finite interval, and that $f(t) = \frac{1}{2}(f(t+) + f(t-))$ for all t, i.e. f is of bounded variation, then*

$$f(t_0) = \lim_{A\to\infty} \frac{1}{2\pi} \int_{-A}^{A} \hat{f}(\omega)e^{i\omega t_0}d\omega \tag{2.12}$$

for every t_0 where f has left and right derivatives.

We omit the proof of the Inversion theorem for continuous Fourier transforms in this discussion. A rigorous treatment of the theorem is given in Vretbald [3] and Stein and Sharkachi [5], while Brigham [1] discusses the conditions imposed on f in the theorem, along with examples.

Definition 2.13. (Inverse Fourier Transform) The integral expression in Equation (2.12) is defined as the **Inverse Fourier Transform** of $\hat{f}$.

The most common applications of Fourier transforms have t representing time-domain units and ω representing angular frequency domain units. Both t and ω are real variables, while $f(t)$ and $\hat{f}(\omega)$ are complex. When oscillation frequency $v = \frac{\omega}{2\pi}$ is used, the Fourier integral pair can be expressed in the symmetrical form

$$\hat{f}(v) = \int_{-\infty}^{\infty} f(t)e^{-2\pi ivt}dv \tag{2.14}$$

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(v)e^{2\pi ivt}dv \tag{2.15}$$

Equation (2.14) is commonly referred to as the **forward Fourier transform**, while equation (2.15) is referred to as the **inverse Fourier transform**. The dual processes of forward and inverse Fourier transformations are used to simplify a large number of operations (a partial list of these can be found in Table A.1 in the Appendix). We concern ourselves here with convolutions, and their continuous Fourier transforms.

Definition 2.16. (Convolution) Let f and g be two functions satisfying Equation (2.10). The **convolution** $f \star g$ is defined to be the function given by the formula

$$(f \star g)(t) = \int_{\mathbb{R}} f(t-y)g(y)dy = \int_{\mathbb{R}} f(y)g(t-y)dy, \quad t, y \in \mathbb{R} \tag{2.17}$$

Theorem 2.18. (Convolution Theorem) *Let $\mathcal{F}$ be the map from the space of functions satisfying Equation (2.10), to the space of continuous functions on $\hat{\mathbb{R}}$ that tend to 0 at $\pm\infty$, such that $\mathcal{F}[f] = \hat{f}$, as defined in Equation (2.6). Then*

$$\mathcal{F}[f \star g] = \mathcal{F}[f]\mathcal{F}[g] \tag{2.19}$$

Proof.

$$\begin{aligned}
\mathcal{F}[f \star g](\omega) &= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} f(t-y)g(y)dy \right) e^{-i\omega t} dt \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} f(t-y)g(y)e^{-i\omega(t-y+y)} dtdy \\
&= \int_{\mathbb{R}} g(y)e^{-i\omega y} dy \int_{\mathbb{R}} f(t-y)e^{-i\omega(t-y)} dt \\
&= \int_{\mathbb{R}} g(y)e^{-i\omega y} dy \int_{\mathbb{R}} f(t)e^{-i\omega(t)} dt = \mathcal{F}[f]\mathcal{F}[g]
\end{aligned}$$

We assume over here that the order of integration can be changed.

□

The convolution theorem demonstrates how Fourier transformations simplify convolutions into a multiplication problem. Using the inversion theorem, the convolution $f \star g$ can be obtained from the product of the Fourier transforms of f and g.

3. The Discrete Fourier Transform (DFT)

The Discrete Fourier Transform is an approximation of the continuous Fourier transform for the case of discrete functions. Given a real sequence of $\{x_n\}$, the DFT expresses them as a sequence $\{X_k\}$ of complex numbers, representing the amplitude and phase of different sinusoidal components of the input 'signal'. As with the continuous case, inversion can be used to reconstruct the original function. However, the DFT has the additional requirement that the sequence $\{x_n\}$ be a sample of a continuous function truncated over a finite interval.

Definition 3.1. (Discrete Fourier Transform) The Discrete Fourier Transform (DFT) of a signal x may be defined by

$$X(\omega_k) = \sum_{n=0}^{N-1} x(t_n)e^{-i\omega_k t_n}, \quad k = 0, 1, ..., N-1 \tag{3.2}$$

The construction of the DFT uses the important fact that the sinusoidal terms in the DFT form an orthogonal basis of the space $\mathbb{C}^n$ (which may be normalized to obtain an orthonormal basis for $\mathbb{C}^n$). In the following discussion, we shall write $\omega_k = \frac{2\pi k}{NT}$ and $t_n = nT$, where T is the sampling interval of the signal. With these substitutions we can write $e^{-i\omega_k t_n} = e^{-2\pi i \frac{nk}{N}}$.

Theorem 3.3. *Let*

$$s_k(n) = e^{2\pi i \frac{nk}{N}}, \quad n = 0, 1, ..., N-1$$

denote the kth DFT complex sinusoid, for $k = 0, 1, ..., N-1$. *Then*

$$s_k \perp s_l, \quad k \neq l, \quad 0 \leq k, l \leq N-1$$

Specifically,

$$< s_k, s_l >= \begin{cases} N, & k = l \\ 0, & k \neq l \end{cases}$$

Proof. We take the inner product of two complex sinusoidal vectors s_k and s_l.

$$< s_k, s_l >= \sum_{n=0}^{N-1} s_k(n)\overline{s_l(n)} = \sum_{n=0}^{N-1} e^{2\pi i \frac{nk}{N}} e^{-2\pi i \frac{nl}{N}}$$
$$= \sum_{n=0}^{N-1} e^{2\pi i \frac{n(k-l)}{N}} = \frac{1 - e^{2\pi i(k-l)}}{1 - e^{2\pi i \frac{(k-l)}{N}}}, k \neq l$$

If $k = l$, we get from the second-to-last step that $< s_k, s_l >= \sum_{n=0}^{N-1} e^0 = N$. If $k \neq l$, we have $< s_k, s_l >= \frac{1-e^{2\pi i(k-l)}}{1-e^{2\pi i \frac{(k-l)}{N}}}$. The denominator is non-zero, while the numerator is zero, since $e^{2\pi i \frac{(k-l)}{N}}$ is a primitive N^{th} root of unity.

□

Remark 3.4. In the last step, we made use of the closed-form expression for the sum of a geometric series

$$\sum_{n=0}^{N-1} z^n = \frac{1 - z^N}{1 - z}$$

Remark 3.5. When $k = l$, the inner product gives us $< s_k, s_l >= N$, which gives us the norm of the DFT sinusoids $\|s_k\| = \sqrt{N}$. Normalizing the DFT sinusoids, we obtain the orthonormal set:

$$\tilde{s}_k(n) = \frac{e^{2\pi i \frac{nk}{N}}}{\sqrt{N}}$$

Theorem 3.3 demonstrates that the set of vectors $\{s_k(n) : k = 0, 1, ..., N-1, n = 0, 1, ..., N-1\}$ are orthogonal (linearly independent) and span the space $\mathbb{C}^n$, thereby forming a basis. Given a signal $x(t_n) \in \mathbb{C}^n$, its DFT is now defined by

$$X(\omega_k) =< x, s_k >= \sum_{n=0}^{N-1} x(n)\overline{s_k(n)}, \quad k = 0, 1, ..., N-1$$

which is the expression in Equation (3.2). Substituting $s_k(n) = e^{i\omega_k t_n}, t_n = nT, \omega_k = \frac{2\pi k}{NT}$, we get a more commonly written form of the DFT:

$$X(\omega_k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi i \frac{nk}{N}}, \quad k = 0, 1, ..., N-1 \tag{3.6}$$

The inverse DFT is constructed by projecting signals $x(t_n)$ onto s_k:

$$\mathbf{P}_{s_k}(x) = \frac{< x, s_k >}{\|s_k\|^2} s_k = \frac{X(\omega_k)}{N} s_k \tag{3.7}$$

Since the $\{s_k\}$ are orthogonal and span $\mathbb{C}^n$, we have

$$x(t_n) = \sum_{k=0}^{N-1} \frac{X(\omega_k)}{N} s_k \tag{3.8}$$

which is commonly written as

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(\omega_k) e^{2\pi i \frac{nk}{N}}, \quad n = 0, 1, ..., N-1 \tag{3.9}$$

Definition 3.10. (Inverse Discrete Fourier Transform) The **Inverse DFT** of a frequency domain signal $X(\omega_k)$ is defined by the expression given in Equation (3.9)[2]

We conclude our discussion on the construction of the DFT by noting that the DFT is proportional to $\frac{<x,s_k>}{\|s_k\|^2}$, which is the set of coefficients of projection onto the sinusoidal basis set, and the inverse DFT is a reconstruction of the original signal as a superposition of its sinusoidal projections. Like the continuous Fourier transform, the DFT's main utility lies in its ability to convert operations on functions in the time domain into simpler, equivalent operations in the frequency domain. A list of common operations and their DFTs can be found in Appendix 1. As with the continuous case, we concern ourselves with discrete convolutions, and their DFTs. In the following discussion, we use $DFT_{N,k} = X(k)$ to denote the DFT of a sequence of length N.

Definition 3.11. (Discrete Convolution) The **Convolution** of two signals $x(t_n)$ and $y(t_n)$ in $\mathbb{C}^n$, denoted by $x \star y$, is defined by

$$(x \star y)(n) = \sum_{m=0}^{N-1} x(m) y(n-m) \tag{3.12}$$

Theorem 3.13. Discrete Convolution Theorem *For any $x, y \in \mathbb{C}^n$,*

$$x \star y = X \cdot Y \tag{3.14}$$

where $X(\omega_k) = DFT_{N,k}(x(n)), Y(\omega_k) = DFT_{N,k}(y(n))$.

[2]Stoer and Bulirsch [2] provide a numerical analytic proof demonstrating how this formula is the 'best' reconstruction of the original waveform.

Proof.

$$\begin{aligned} DFT_{N,k}(x \star y) &= \sum_{n=0}^{N-1} (x \star y)(n) e^{-2\pi i \frac{nk}{N}} \\ &= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(m) y(n-m) e^{-2\pi i \frac{nk}{N}} \\ &= \sum_{m=0}^{N-1} x(m) \sum_{n=0}^{N-1} y(n-m) e^{-2\pi i \frac{nk}{N}} \end{aligned}$$

The sum on the right becomes

$$\begin{aligned} \sum_{n=0}^{N-1} y(n-m) e^{-2\pi i \frac{nk}{N}} &= \sum_{n=-m}^{N-1-m} y(n) e^{-2\pi i \frac{(n+m)k}{N}} \\ &= \sum_{n=0}^{N-1} y(n) e^{-2\pi i \frac{mk}{N}} \\ &= e^{-2\pi i \frac{mk}{N}} \sum_{n=0}^{N-1} y(n) \\ &= e^{-2\pi i \frac{mk}{N}} Y(k) \end{aligned}$$

Therefore, we have

$$DFT_{N,k}(x \star y) = \left(\sum_{m=0}^{N-1} x(m) e^{-2\pi i \frac{mk}{N}} \right) Y(k) = X(k) \cdot Y(k)$$

□

This theorem is of great practical importance as it forms the basis of a large body of FFT theory. The Fast Fourier Transform provides a fast method of computing the DFT, and consequently a 'fast convolution' when Theorem (3.13) is applied.

4. Computational Considerations

We recall the definition of the DFT of a sequence $x(t_n)$, from Equation (3.2):

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-i\omega_k t_n}, \quad k = 0, 1, ..., N-1 \tag{4.1}$$

where we have replaced ω_k by k and t_n by n for convenience of notation. Let

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

Then Equation (4.1) can be written as

$$\begin{aligned}
X(0) &= x(0)W_N^0 + x(1)W_N^0 + x(2)W_N^0 + \ldots + x(N-1)W_N^0 \\
X(1) &= x(0)W_N^0 + x(1)W_N^2 + x(2)W_N^3 + \ldots + x(N-1)W_N^{N-1} \\
&\vdots \\
X(N-2) &= x(0)W_N^0 + x(1)W_N^{N-2} + x(2)W_N^{2(N-2)} + \ldots + x(N-1)W_N^{(N-1)(N-2)} \\
X(N-1) &= x(0)W_N^0 + x(1)W_N^{N-1} + x(2)W_N^{2(N-1)} + \ldots + x(N-1)W_N^{(N-1)(N-1)}
\end{aligned}$$

This system of equations is more conveniently represented in the matrix form below:

(4.2)

$$\begin{bmatrix} X(0) \\ X(1) \\ \ldots \\ X(N-2) \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \ldots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \ldots & W_N^{N-1} \\ \ldots & \ddots & \ddots & \ddots & \ldots \\ W_N^0 & W_N^{N-2} & W_N^{2(N-2)} & \ldots & W_N^{(N-1)(N-2)} \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \ldots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \ldots \\ x(N-2) \\ x(N-1) \end{bmatrix}$$

or more compactly as

$$\mathbf{X(k)} = \mathbf{W_N^{nk} x(n)} \tag{4.3}$$

To comprehend the size and complexity of the computational task at hand, it is essential to examine Equation (4.3). Here, W_N and $x(n)$ possibly are complex quantities. Each row in the matrix $\mathbf{W_N}$ has N elements, each of which has to be multiplied by its corresponding N elements in the column matrix $\mathbf{x(n)}$, giving us N multiplications per row of $\mathbf{W_N}$. Furthermore, the matrix multiplication involves $N-1$ additions in every row of $\mathbf{W_N}$, one following every multiplication. Since there are N rows in $\mathbf{W_N}$, the required matrix computation involves N^2 complex multiplications and $N(N-1)$ complex additions to be performed. For large N, DFT computations require large quantities of time, even with high speed computers. The reduction of machine time involved in the computation of the DFT is the primary motivation behind the development of the family of algorithms that are known as Fast Fourier Transforms, which efficiently implement the DFT for highly composite transform lengths N. We proceed to examine the construction of the Cooley-Tukey FFT Algorithm, and the order of computations required in its implementation.

5. The radix-2 Cooley-Tukey FFT Algorithm

When the transform length is of arbitrary integer composite size, i.e. $N = N_1 N_2$, the Cooley-Tukey algorithm recursively rewrites the DFT in terms of smaller DFTs of sizes N_1 and N_2, so as to reduce the computation time. The two basic approaches towards implementation of Cooley-Tukey FFT are decimation in time (DIT), and to compute the Inverse DFT, decimation in frequency (DIF)[3]. The choice between DIT and DIF is made depending on the relative sizes of N_1 and N_2. The following discussion presents a radix-2 DIT FFT, in which a DFT of size N is split into two

[3]DIF is also known as the Sande-Tukey FFT algorithm.

DFTs of size $\frac{N}{2}$ at each stage of the recursion. We assume that $N = 2^\gamma$, where γ is an integer.[4]

We recall the definition of the DFT, as in Equation (3.2):

$$X(\omega_k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = 0, 1, ..., N-1 \tag{5.1}$$

where $W_N = e^{\frac{-2\pi i}{N}}$. When $N = 2^\gamma$, n and k can be represented in binary form as

$$\begin{aligned} n &= 2^{\gamma-1} n_{\gamma-1} + 2^{\gamma-2} n_{\gamma-2} + \ldots + n_0 \\ k &= 2^{\gamma-1} k_{\gamma-1} + 2^{\gamma-2} k_{\gamma-2} + \ldots + k_0 \end{aligned} \tag{5.2}$$

Rewriting Equation (5.1) we get

$$X(k_{\gamma-1}, k_{\gamma-2}, \ldots, k_0) = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \cdots \sum_{n_{\gamma-1}=0}^{1} x(n_{\gamma-1}, n_{\gamma-2}, \ldots, n_0) W_N^p \tag{5.3}$$

where

$$p = (2^{\gamma-1} k_{\gamma-1} + 2^{\gamma-2} k_{\gamma-2} + \ldots + k_0)(2^{\gamma-1} n_{\gamma-1} + 2^{\gamma-2} n_{\gamma-2} + \ldots + n_0) \tag{5.4}$$

Since $W_N^{a+b} = W_N^a W_N^b$, we rewrite W_N^p as

$$\begin{aligned} W_N^p = {} & W_N^{(2^{\gamma-1} k_{\gamma-1} + 2^{\gamma-2} k_{\gamma-2} + \ldots + k_0)(2^{\gamma-1} n_{\gamma-1})} W_N^{(2^{\gamma-1} k_{\gamma-1} + 2^{\gamma-2} k_{\gamma-2} + \ldots + k_0)(2^{\gamma-2} n_{\gamma-2})} \\ & \times \ldots \times W_N^{(2^{\gamma-1} k_{\gamma-1} + 2^{\gamma-2} k_{\gamma-2} + \ldots + k_0) n_0} \end{aligned} \tag{5.5}$$

Now consider the first term of Equation (5.5)

$$\begin{aligned} W_N^{(2^{\gamma-1} k_{\gamma-1} + 2^{\gamma-2} k_{\gamma-2} + \ldots + k_0)(2^{\gamma-1} n_{\gamma-1})} &= W_N^{2^\gamma (2^{\gamma-2} k_{\gamma-1} n_{\gamma-1})} W_N^{2^\gamma (2^{\gamma-3} k_{\gamma-2} n_{\gamma-1})} \\ &\times \ldots \times W_N^{2^\gamma (k_1 n_{\gamma-1})} W_N^{2^{\gamma-1} (k_0 n_{\gamma-1})} \\ &= W_N^{2^{\gamma-1} (k_0 n_{\gamma-1})} \end{aligned}$$

since

$$W_N^{2^\gamma} = W_N^N = e^{-2\pi i} = 1 \tag{5.6}$$

Similarly, the second term of Equation (5.5) yields

$$\begin{aligned} W_N^{(2^{\gamma-1} k_{\gamma-1} + 2^{\gamma-2} k_{\gamma-2} + \ldots + k_0)(2^{\gamma-2} n_{\gamma-2})} &= W_N^{2^\gamma (2^{\gamma-3} k_{\gamma-1} n_{\gamma-2})} W_N^{2^\gamma (2^{\gamma-4} k_{\gamma-2} n_{\gamma-2})} \\ &\times \ldots \times W_N^{2^{\gamma-1} (k_1 n_{\gamma-2})} W_N^{2^{\gamma-2} (k_0 n_{\gamma-2})} \\ &= W_N^{2^{\gamma-2} (2k_1 + k_0)(n_{\gamma-1})} \end{aligned}$$

As we proceed through the terms of Equation (5.5), we add another factor which does not cancel by the condition $W_N^{2^\gamma} = 1$. This process continues until we reach the last term in which there is no cancellation.

Using these relationships, Equation (5.3) can be rewritten as

---

[4] Since applications usually are free to choose their sample lengths, this is not a major constraint, and analogous techniques still work for the general case.

$$X(k_{\gamma-1}, k_{\gamma-2}, \ldots, k_0) = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \cdots \sum_{n_{\gamma-1}=0}^{1} x(n_{\gamma-1}, n_{\gamma-2}, \ldots, n_0)$$
$$\times W_N^{2^{\gamma-1}(k_0 n_{\gamma-1})} \times W_N^{2^{\gamma-2}(2k_1+k_0)(n_{\gamma-1})} \times \ldots$$
$$\times W_N^{(2^{\gamma-1}k_{\gamma-1}+w^{\gamma-2}k_{\gamma-2}+\ldots+k_0)n_0}$$

Performing each of the summations separately and labeling the intermediate results, we obtain

$$x_1(k_0, n_{\gamma-2}, \ldots, n_0) = \sum_{n_{\gamma-1}=0}^{1} x_0(n_{\gamma-1}, n_{\gamma-2}, \ldots, n_0) W_N^{2^{\gamma-1}(k_0 n_{\gamma-1})}$$
$$x_2(k_0, k_1, n_{\gamma-3}, \ldots, n_0) = \sum_{n_{\gamma-2}=0}^{1} x_1(k_0, n_{\gamma-2}, \ldots, n_0) W_N^{2^{\gamma-2}(2k_1+k_0)(n_{\gamma-1})}$$
$$\vdots$$
$$x_\gamma(k_0, k_1, \ldots, k_{\gamma-1}) = \sum_{n_0=0}^{1} x_{\gamma-1}(k_0, k_1, \ldots, n_0) W^{(2^{\gamma-1}k_{\gamma-1}+w^{\gamma-2}k_{\gamma-2}+\ldots+k_0} n_0$$
$$X(k_{\gamma-1}, k_{\gamma-2}, \ldots, k_0) = x_\gamma(k_0, k_1, \ldots, k_{\gamma-1})$$

This set of recursive equations represents the FFT as proposed by Cooley and Tukey for $N = 2^\gamma$. The direct evaluation of the DFT for an input sequence of length N requires N^2 multiplications, as shown above. In the radix-2 FFT algorithm, there are γ summations, each representing N equations. Of these N equations, each contains two complex multiplications. However, one of these multiplications is always with unity, and so it may be skipped. Since we have γ summations representing N equations, each having one complex multiplication, this gives us $N\gamma = N \log_2(N)$ operations, a considerable improvement from the direct evaluation of the DFT.

6. Acknowledgments

It is a pleasure to thank my mentors, Yan Zhang and Jessica Lin, for guiding me through the study of Fourier transforms and and for helping me develop my understanding through both examples and theory.

References

[1] E. Oran Brigham. The Fast Fourier Transform. Prentice-Hall, Inc, 1974.
[2] J. Stoer and R. Bulirsch. Introduction to Numerical Analysis. Springer-Verlag New York, Inc. 1993.
[3] Anders Vretbald. Fourier Analysis and its Applications. Springer-Verlag New York, Inc. 2003.
[4] Henri J. Nussbaumer. Fast Fourier Transform and convolution algorithms. Springer-Verlag New York, Inc. 1982.
[5] Elias M. Stein and Rami Sharkachi. Fourier Analysis: an introduction. Princeton University Press, 2003.

[6] Julius O. Smith. Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications. W3K Publishing, 2007.
[7] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. Math. Computat., vol. 19, pp. 297-301, 1965.
[8] Otto Neugebauer. The Exact Sciences in Antiquity. Dover Publications, 1969.

7. Appendix 1

In the following table, $x \Longleftrightarrow X$ shall denote that X and x are Fourier transform pairs, i.e. X is the Fourier transform of x, which can be retrieved from X via the inverse Fourier transform.

Table A.1

Fourier Transform	Property	Discrete Fourier Transform
$x(t) + y(t) \Longleftrightarrow X(\omega) + Y(\omega)$	Linearity	$x(t_n) + y(t_n) \Longleftrightarrow X(\omega_k) + Y(\omega_k)$
$X(t) \Longleftrightarrow x(\omega)$	Symmetry	$\frac{1}{N}X(t_n) \Longleftrightarrow x(-\omega_k)$
$x(t - t_0) \Longleftrightarrow X(\omega)e^{-2\pi i\omega t_0}$	Time Shifting	$x(t_n - t_m) \Longleftrightarrow X(\omega_k)e^{-2\pi i\frac{mk}{N}}$
$x(t)e^{-2\pi i\omega_0 t} \Longleftrightarrow X(\omega - \omega_0)$	Frequency Shifting	$x(t_n)e^{-2\pi i\frac{km}{N}} \Longleftrightarrow X(\omega_k - \omega_m)$
$x_e(t) \Longleftrightarrow R_e(\omega)$	Even Functions	$x_e(t_n) \Longleftrightarrow R_e(\omega_k)$
$x_o(t) \Longleftrightarrow iI_o(\omega)$	Odd Functions	$x_o(t_n) \Longleftrightarrow iI_o(\omega_k)$
$x(t) \star y(t) \Longleftrightarrow X(\omega)Y(\omega)$	Odd Functions	$x(t_n) \star y(t_n) \Longleftrightarrow X(\omega_k)Y(\omega_k)$