A Short Note on Fourier Analysis

C. Radcliffe

The Discrete Fourier Transform

The Fourier transform provides a transformation between the time and frequency domain. As shown in (1) and (2) of Bergland's paper, for continuous, non-periodic signals, the Fourier transform $X(f)$ and the time signal $x(t)$ are related by

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt \tag{1}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{i2\pi ft} df \tag{2}$$

The methods of Fourier analysis are often employed to analyze signals and linear systems. The first application usually seen is the use of Fourier series to describe periodic signals. We assume that we are given a function $f(t)$ defined on an interval of length T (for convenience we will assume that this interval is symmetric with respect to the origin). For periodic signals, (1) and 92) become

$$X(f) = \int_{0}^{T} x(t) e^{-i2\pi ft} dt \tag{3}$$

$$x(t) = \sum_{n=-\infty}^{\infty} X(f) e^{i2\pi ft} f_0 = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(f) e^{i2nf_0 t} \tag{4}$$

where the fundamental frequency, $df \Rightarrow \Delta f = f_0 = \frac{1}{T}$ and all other frequencies, $f = nf_0$ are harmonics of the fundamental frequency.

For N data sampled over time T, the time each sample is taken, $t = k\,\Delta t$ where $\Delta t = T/N$. If these samples are taken over one period of the periodic waveform, N unique values of $X(f)$ at frequencies $f = nf_0$ can be computed uniquely

$$X(f) = X(nf_0) = X(n) = \sum_{k=0}^{N-1} x(k\Delta t) e^{-i2\pi (nf_0)(k\Delta t)} \Delta t \tag{5}$$

$$x(k\Delta t) = x(k) = \left(\frac{1}{T}\right) \sum_{n=0}^{N-1} X(nf_0) e^{i2\pi (nf_0)(k\Delta t)} \tag{6}$$

In both expressions above, the fundamental frequency, $f_0 = \frac{1}{T}$ and $\Delta t = T/N$ so that

$$i2\pi(nf_0)(k\Delta t) = i2\pi(n/T)(kT/N) = i2\pi(nk/N) \tag{7}$$

and (5) and (6) become

$$X(n) = \left(\frac{T}{N}\right)\sum_{k=0}^{N-1} x(k)\,e^{-i2\pi(nk/N)} \tag{8}$$

$$x(k) = \left(\frac{1}{T}\right)\sum_{n=0}^{N-1} X(n)\,e^{i2\pi(nk/N)} \tag{9}$$

Notice that in the above results, we can normalize the result by defining

$$\hat{X}(n) = X(n)/T \tag{10}$$

and rewrite (8) and (9) as

$$\hat{X}(n) = \left(\frac{1}{N}\right)\sum_{k=0}^{N-1} x(k)\,e^{-i2\pi(nk/N)} \tag{11}$$

$$x(k) = \sum_{n=0}^{N-1} \hat{X}(n)\,e^{i2\pi(nk/N)} \tag{12}$$

The normalized transform $\hat{X}(n)$ is referred to as the Discrete Fourier Transform (DFT) and normally is used without a "hat". This result is the result of equations (3) and (4) in Bergland's paper. Even when the weighting functions $e^{-i2\pi(nk/N)}$ are pre-computed and stored, the DFT (11) requires N multiply-addition pairs to compute for each value of $X(n)$. Since there are N values of $X(n)$ to be computed, N^2 multiply-addition pairs are required for the complete DFT computation. For large N, the result is long computation times.

An important property of the DFT is that $x(k) = x(k+N)$ and $X(n) = X(n+N)$; that is, both $x(k)$ and $X(n)$ are periodic, with period N. This follow directly since $e^{i2\pi p} = 1$ where p is an integer.

The Fast Fourier Transform

The Cooley-Tukey algorithm for the Fast Fourier Transform reduces N^2 arithmetic operations required for a DFT to only NM operations for the FFT if $N = 2^M$. For $N = 2^{10} = 1024$, the number of operations is reduced by approximately a factor of 100.

The algorithm is deveop4ed by first decomposing $x(k)$ into its odd and even points and evaluating $X(n)$ as the sum of the DFT's of the two sequences. Define,

$$W_N = e^{-j(2\pi/N)} \tag{13}$$

Now split the DFT into even and odd halves,

$$\begin{aligned} X(k) &= \sum_{r=0}^{(N/2)-1} x(2r) W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1) W_N^{(2r+1)k} \\ &= \sum_{r=0}^{(N/2)-1} x(2r) \left[W_N^2\right]^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1) \left[W_N^2\right]^{rk} \end{aligned} \tag{14}$$

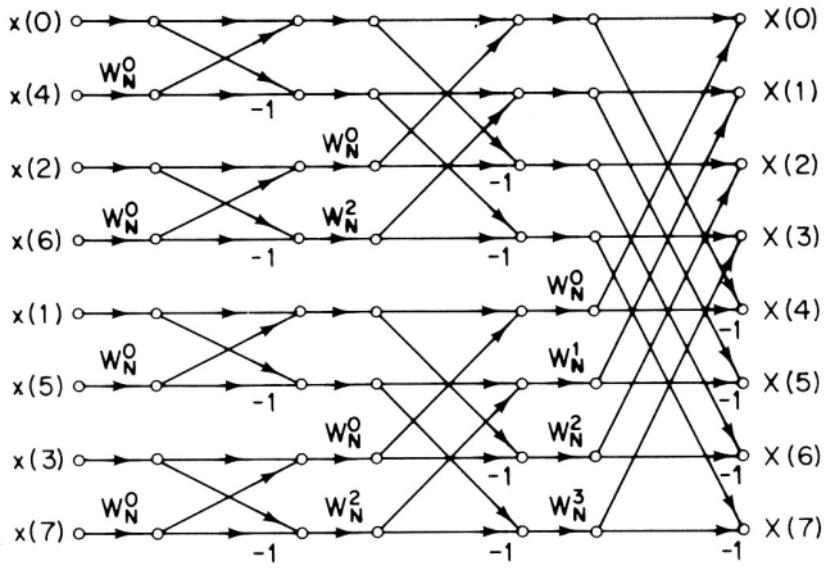
But,

$$W_N^2 = e^{-j(2\pi/N)2} = e^{-j(2\pi/(N/2))} = W_{N/2} \tag{15}$$

and therefore,

$$\begin{aligned} X(k) &= \sum_{r=0}^{(N/2)-1} x(2r) \left[W_{N/2}^{rk}\right] + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1) \left[W_{N/2}^{rk}\right] \\ &= G(k) + W_N^k H(k) \end{aligned} \tag{16}$$

The N point DFT has now been reduced to two $N/2$ point DFT's plus the operation of a weighted multiply for each of the N points in the N point DFT. This process has reduced the number of computations from N^2 to $2[N/2]^2 + N = N^2/2 + N$. Now, using similar arguments, each $N/2$ point DFT is reduced to two $N/4$ point DFT's and $N/2$ combination operations. An N point DFT would then require $\left[2\left[2(N/4)^2 + N/2\right]\right] + N = 2\left[N^2/8 + N/2\right] + N = N^2/4 + N + N$ operations. Because $N = 2^M$, this "divide by 2" process occurs M times. Thus the number of operations is reduced to MN. A flowgraph for the algorithm when N=8 (M=3) is shown below.



Flow Diagram for N=8 Point Fast Fourier Transform (FFT)

(From Eckhouse and Morris, pg 287)

This figure reveals a number of interesting properties of the FFT algorithm.

1) If the index n of each input point $x(n)$ is represented as an M bit binary number, the initial input array must be "shuffled' into *bit reversed* order before the FFT is implemented.
2) M stages are required. In the i[th] stage, $N/2$ computations of the form

$$X_i(p) = X_{i-1}(p) + W_N^r X_{i-1}(q) \tag{17a}$$

$$X_i(q) = X_{i-1}(p) + W_N^{r+(N/2)} X_{i-1}(q) \tag{17b}$$

where p and q are the indices of the *butterfly* pair.

For $i = 1$, the *input* X's are the x's shuffled as per above. Since $W_N^{N/2} = e^{-j\pi} = -1$, (17a,b) reduce to

$$X_i(p) = X_{i-1}(p) + W_N^r X_{i-1}(q) \tag{18a}$$

$$X_i(q) = X_{i-1}(p) - W_N^r X_{i-1}(q) \tag{18b}$$

This effectively halves the number of multiplications required. At each point, two complex points are operated upon to produce two results which effectively replace the original two input points. That is $X_i(p)$ and $X_i(q)$ can be stored in the same locations as $X_{i-1}(p)$ and $X_{i-1}(q)$. Thus, the FFT can be done in place and the bit-reversed input array is progressively replaced with the output FFT array. The above is the Cooley-Tukey *decimation-in-time* algorithm. A *decimation-in-frequency* algorithm is also available.

References

G.D. Bergland, "A Guided Tour of the Fast Fourier Transform", *IEEE Spectum*, vol.6, pp. 41-52, July 1969.

R.H. Eckhouse and L.R. Morris, *Minicomputer Systems: Organization, Programming and Applications (PDP-11)*, Prentice Hall, pp.285-290, 1979.