An Introduction to the Discrete Fourier Transform

W. Smith

The purpose of this note is to introduce the novice to the discrete Fourier transform (DFT) and as such it is not intended to be comprehensive or even rigorous. It is hoped however, that sufficient isight is provided to enable newcomers to the DFT to understand and avoid some of the problems associated with its use. For a more comprehensive account, readers are referred to reference 1, which provides an excellent introduction to the DFT in general. This note may be regarded, in some part, as being a selective precis of the book. The note begins with a brief description of the general Fourier transform, followed by its adaptation to the discrete case. Lastly the application of the DFT to spectral analysis and the evaluation of correlation functions is briefly discussed.

The Fourier Transform

The Fourier transform H of a function h is given by:

$$H(f) = \int_{-\infty}^{+\infty} h(t)\ \exp(-i\ 2\pi ft)dt \qquad (1)$$

Where, by convention, h is assumed to be a function of time (t) and H a function of frequency (f). It should be noted that either or both of h and H may be complex. The inverse Fourier transform is given by:

$$H(f) = \int_{-\infty}^{+\infty} H(f)\ \exp(i\ 2\pi ft)df \qquad (2)$$

An important question at this stage concerns the actual existence of the Fourier integrals. In general it may be assumed that any function representable by a finite length curve in any chosen interval (i.e. bounded variation) will possess a Fourier transform. There are useful functions which do not comply with this criterion but nevertheless possess Fourier transforms, (such as the impulse and sampling functions mentioned below) but these are best treated here as special cases.

Perhaps the most familiar functions which comply with the above criterion are the trigonometric functions which Fourier transform (symbolised by <=>) as follows:

$$\begin{aligned} \cos(2\pi\ f_o t) &\Leftrightarrow \frac{1}{2}\left(\delta(f + f_o) + \delta(f - f_o)\right) \\ \sin(2\pi\ f_o t) &\Leftrightarrow \frac{i}{2}\left(\delta(f + f_o) - \delta(f - f_o)\right) \end{aligned} \qquad (3)$$

Where the function $\delta(f - fo)$ is the so-called impulse function, which is defined by:

$$\begin{aligned} &\delta(x - x_o) = 0 \quad \text{if } x \neq x_o \\ &\int_{-\infty}^{+\infty} \delta(x - x_o)dx = 1 \end{aligned} \qquad (4)$$

Which incidentally, has the very useful property that for any given function f(x):

$$\int_{-\infty}^{+\infty} f(x)\ \delta(x - x_o)dx = f(x_o) \tag{5}$$

Thus the simple trigonometric functions Fourier transform from continuous functions in the time domain to pairs of sharp peaks in the frequency domain. (This is in fact simpler to establish by using the property (5) in the inverse Fourier transform (2)). In view of the traditional role of the Fourier transform in harmonic analysis this result is of course, expected!

Further examples of Fourier transforms of relevance here are the sampling function:-

$$s(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nt_o) \iff S(f) = \frac{1}{t_o} \sum_{n=-\infty}^{+\infty} \delta(f - \frac{n}{t_o}) \tag{6}$$

and the (rectangular) window function:

$$w(t) = \begin{cases} 1 & |t| < T_o \\ \frac{1}{2} & |t| = T_o \\ 0 & |t| > T_o \end{cases} \iff W(f) = 2T_o \frac{\sin(2\pi T_o f)}{2\pi T_o f} \tag{7}$$

Both of these functions are needed in the adaptation of the Fourier integral to the discrete Fourier transform to be discussed later. The Fourier transforms of other mathematical functions can be found in general texts[2].

The Fourier transform has a number of useful properties which greatly assist the manipulation of the transform. These properties, which include linearity, symmetry, scaling effects and modulation and phase shifting are discussed in reference 1.

The Discrete Fourier Transform

To convert the Fourier transform from its integral representation (1) to a discrete representation amenable to digital processing, substantial modifications of the original time function are necessry. These modifications result in subtle changes in the properties of the transform that affect the accuracy and the interpretability of the result. It is instructive to examine these modifications in turn to learn of their effects and where possible, the remedies of these effects. The principal modifications referred to here are sampling and windowing in the time domain and sampling in the frequency domain.

(i) Sampling in the Time Domain

The continuous function of time h(t) may be converted to a discrete representation by multiplying it by the sampling function s(t) given in equation (6). The sampling function consists of an infinite train of regularly spaced 'sharp peaks' of unit area but of infinitesimal width. The peaks are separated by the time interval t_o. The result of this multiplication is to produce a set of data points, equally spaced in the time domain at intervals of t_o and with a value related to the function $h(t)$ at the corresponding abscissa. The effect of this sampling function on the Fourier transform however, is rather more complicated than a simple multiplication of the corresponding Fourier transformed functions $S(f)$ and $H(f)$. The result is in fact, a convolution integral, which may be summarised thus:

$$h(t)\ s(t) \; < \begin{array}{l} \xrightarrow[\text{domain}]{\text{time}} \quad \sum_{n=-\infty}^{n=+\infty} h(nt_o) \\[2ex] \xrightarrow[\text{domain}]{\text{frequency}} \quad \int_{-\infty}^{+\infty} S(f')H(f-f')df' \end{array} \tag{8}$$

The fact that the result in the frequency domain is a convolution integral, is of sufficient importance to warrant at least a demonstration of its truth. First it is necessary to define a function $u(t) = h(t)\ s(t)$. From the Fourier transforms of these functions (equation (1)) $u(t)$ may be written directly as:-

$$u(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} S(f)\ H(\sigma)\ \exp(i2\pi(f + \sigma)t)\ d\sigma\ df$$

Using the substitution $f' = f + \sigma$ and re-arranging gives:

$$u(t) = \int_{-\infty}^{+\infty} \left\{\int_{-\infty}^{+\infty} S(f)H(f' - f)df\right\} \exp(i2\pi f't)df'$$

By comparison with equation (2) it can be seen at once that the term written in curled brackets above is the Fourier transform of $u(t)$, i.e. equations (8) are valid.

The Fourier transform $S(f)$ as shown in equation (6) is a series of impulse functions separated by the frequency interval $1/t_o$. The convolution of this function with the Fourier transform $H(f)$ results in a <u>continuous</u> function of frequency consisting of periodic superpositions of the function $H(f)$ centred on the locations of the impulse functions (i.e. repeats of the basic function $H(f)$ set at intervals of $1/t_o$ apart):

$$U(f) = \int_{-\infty}^{+\infty} S(f')\ H(f - f')df' = \frac{1}{t_o} \sum_{n=-\infty}^{+\infty} H(f - \frac{n}{t_o}) \tag{9}$$

This periodic replication of the function $H(f)$ is interesting for a number of reasons. Firstly it is clearly an artefact of the discretization of the original function $h(t)$ as it does not arise in the Fourier transform of the continuous function. Secondly, it is clearly possible for neighbouring replications of H(f) to overlap in some circumstances. When they do they constitute an error in the Fourier transform which is known as aliasing.

If however the original function h(t) is band-limited (i.e. does not possess frequency components higher than a critical frequency f_c say) then it is possible to prevent the overlapping error by choosing a sampling interval of $1/(2f_c)$. In this case, which is known as Nyquist sampling, aliasing does not arise. In the general case, all that can be done is to choose t_o sufficiently small to reduce aliasing by widening the gaps between the impulse functions S(f) in the frequency domain.

(ii) Windowing in the Time Domain

The infinite set of data points produced by sampling of the function h(t) needs to be reduced to a finite set to allow digital processing. The simplest way to achieve this is to multiply the sampled function by the rectangular window function given in (7). This results in a truncated set of data points (and also a truncated time function!). The effect of this operation in the frequency domain is to convolve the Fourier transform of the sampled function given in (9) with the Fourier transform of the window function given in (7) (i.e. U(f) is convolved with W(f)). Since the function W(f) consists of a sharp central peak with smaller oscillations on either side, the convolution introduces ripples into the periodic function U(f). These ripples are the source of an error in the DFT known as leakage. This matter however is best left to the next section where it can be dealt with more fully. Suffice it to say at this stage that if the side oscillations of the Fourier transformed window function could be reduced (as by increasing the width of the window for instance) the problem of leakage may be reduced.

(iii) Sampling in the Frequency Domain

The effect of sampling and windowing on the function h(t) in the time domain is to produce a finite set of discrete data points. In the frequency domain however, this produced a periodic, continuous function (albeit with ripples). To allow a completely discrete formulation of the Fourier transform, it is also necessary to sample the frequency domain.

To achieve this, the truncated and sampled function in the time domain is convolved with a sampling function similar to that given in (6) which has peaks spaced at intervals of T_o (the width of the window function). The effect in the frequency domain is to sample the periodic function H(f) at intervals of $1/T_o$ in frequency. However, as might be expected, this operation of sampling the frequency domain results in a periodic replication of the function in the time domain as it is described in the interval $[-T_o/2,\ T_o/2]$. Thus as far as the discrete Fourier transform is concerned the functions h(t) and H(f) are both periodic functions in their respective domains, the former with a period T_o, and the latter with the period $1/t_o$.

It should be noted that the choice of sampling interval in the frequency domain (i.e. $1/T_o$) is such that the same number of data points are considered in each domain. Also, the possibility of aliasing in the time domain resulting from sampling at this interval in the frequency domain is avoided provided that the window function is purposely chosen to possess extremities that do not coincide with the first and last data points of the time domain function.

The result of applying all these operations is to produce the discrete version of the Fourier integral:

$$\hat{H}(n/(Nt_o)) = \sum_{k=0}^{N-1} \hat{h}(kt_o)\exp(-i2\pi nk/N) \qquad (10)$$

$$n=0,\ldots,N-1$$

The converse of which may be similarly produced.

$$\hat{h}(kt_o) = \frac{1}{N} \sum_{n=0}^{N-1} \hat{H}(n/(Nt_o))\exp(i2\pi nk/N) \qquad (11)$$

$$k=0,\ldots,N-1$$

where $\hat{H}$ and $\hat{h}$ represent the sampled functions H(f) and h(t) respectively. The number of data points in each domain is N. (Note that to enable the formulae to be useable on a computer, the summation indices have been taken to run from 0 to N-1 instead of from -N/2 to N/2 as might be expected from the discussion so far. This in fact results in a phase shift of the Fourier transform, but this is of no practical significance).

The problem of leakage mentioned in the previous section arises from the windowing of the time domain function but it manifests itself in the artificial periodicity that the DFT imposes on the functions h(t) and H(f). If for instance the natural periodicity of the function h(t) is T_o' and the chosen window function imposes a periodicity of T_o (where $T_o \neq T_o'$) then it will be observed that the DFT resolves the frequency dependence of H(t) into a major and several minor frequency components instead of the single frequency component expected. Hence the use of the term 'leakage'. The origin of these minor components are the side oscillations possessed by the Fourier transformed window function W(f), which via the process of convolution introduce spurious peaks into the Fourier transform H(f). If however the period T_o' equals T_o then it transpires that the peaks of the frequency sampling function coincide exactly with the zero points between the side oscillations of W(f) and hence, in this special case, the DFT does not give rise to spurious peaks[1].

For a general function, where the periodicity is unknown leakage is clearly a problem, as spurious peaks may easily arise in the frequency dependence of the function. A much favoured cure for this problem is the use of window functions which have suppressed or non-existant side oscillations in the frequency domain. While this strategy certainly reduces leakage it should be noted that not all window functions are equally good and all of them have drawbacks in their use. Fortunately Harris[3] has carried out a thorough examination of many window functions. His review of them is highly recommended.

Application to Spectral Analysis

The object of the exercise here is a straightforward application of the DFT to a sampled function with a view to resolving its frequency components. This is a requirement frequently encountered in molecular dynamics calculations, where the data are invariably produced in a convenient discretized form. The problems arising in this application have been outlined already. They are aliasing and leakage.

Aliasing may be reduced if the sampling rate to is kept small, which means that the time step in the molecular dynamics calculation must be reasonable. Fortunately, of necessity, the time step will be sufficiently small to allow adequate modelling of the dynamics and so an adequate sampling rate should naturally be available. It may be worth noting however that if the phenomenon of interest is band limited, then selecting a sampling rate near to the Nyquist sampling rate of $1/(2f_c)$ should lead to more efficient data processing provided that this sampling interval is equivalent to several time steps.

Leakage is a particular problem if spurious peaks arise to confuse the interpretation of the results. In this case, the use of a suitable window function is essential. Harris[3] recommends a number of window functions for such applications. Perhaps the best of these (allowing good resolution of neighbouring frequency peaks) are the Blackman-Harris windows, which have the form

$$w(k) = \sum_{j=0}^{3} a_j \cos(2\pi jk/N) \tag{12}$$

Where the coefficients a_0 to a_3 are constants tabulated by Harris[3]. In the time domain these functions suppress discontinuities at the extremities of the truncated time function (or equivalently have suppressed side oscillations in the Fourier transformed window function) thus removing spurious peaks from the frequency spectrum.

Application to Correlation Functions

In molecular dynamics work it is frequently necessary to evaluate integrals of the form:

$$c(t) = \lim_{T \to \infty} \left(\frac{1}{T}\right) \int_0^T u(\tau)\, h(\tau + t)\, d\tau \tag{13}$$

Where u and h are both functions of time. (They may in fact be the same function, in which case c(t) is known as an autocorrelation function). Such functions are used to reveal a cause-and-effect relationship between the functions u and h; by which a change in the value of one function manifests a change in the other at some time interval later.

In molecular dynamics the integral (13) is approximated by the discrete form (14).

$$c(kt_o) = \frac{1}{(N-k)} \sum_{n=0}^{N-k-1} u(nt_o)h((n + k)t_o) \tag{14}$$

$$k=0,1,\ldots,N-1$$

It has been shown however[4] that this direct method is not always the most efficient way to proceed and that by employing the DFT considerable improvements in speed can be achieved. This method hinges on the fact that the Fourier transform of integrals of the type (13) produces a simple product of Fourier transformed functions in the frequency domain (i.e. the integral is replaced by the product $H(f)\ U^*(f)$; the Fourier transforms of h and u respectively - the asterisk indicates the complex conjugate). Clearly, provided the Fourier transform can be carried out efficiently, the evaluation of the correlation function in the frequency domain is easy. Thus a proposed scheme for evaluating correlation functions might be:

(i) Fourier transform $u(t)$ and $h(t)$, obtain $U^*(f)$, $H(f)$.
(ii) Multiply $U^*(t)\ H(f)$, obtain $C(f)$. (15)
(iii) Inverse Fourier transform $C(f)$, obtain $c(t)$

Though this scheme may appear cumbersome it has to be noted that each of these steps can be accomplished very efficiently. Computer applications of the DFT are generally known as 'Fast Fourier Transforms' and with good reason. They are extremely efficient and as Futrelle and McGinty have pointed out[4] this indirect route is very much faster than the direct method once the number of data points exceeds about thirty. (The direct method requires $\approx N^2$ operations while the DFT method requires $\approx 3\ N \log_2 N$ floating point operations).

Using the DFT, the discrete correlation function (14) may be written as:

$$\hat{c}(kt_o) = \frac{1}{(N-k)(2N)} \sum_{n=0}^{2N-1} \hat{U}^*(n/(2Nt_o))\ \hat{H}(n/(2Nt_o))\ \exp(2\pi nk/(2N)) \qquad (16)$$

$$k=0,1,\ldots N-1$$

It will be noticed that the summation indices range over 2N values and not N as expected. This is because it is necessary to double the length of the $u(kt_o)$ and $h(kt_o)$ vectors in the time domain (by appending N zeros to each) to prevent spurious correlations arising. This scheme is in accordance with that given by Kestemont and Van Craen[5].

Sources of DFT routines are listed in reference no.6.

References

1. E.O. Brigham, "The Fast Fourier Transform" Prentice Hall, N.J. 1974.

2. G.A. Campbell, R.M. Foster, "Fourier Integrals for Practical Applications". Bell Telephone System Technical Publication 1931.
   R.C. Weast (Editor), "Handbook of Chemistry and Physics", C.R.C. Press 1981.

3. F.J. Harris, Proc. IEEE. 66, (1978) 51

4. R.P. Futrelle, D.J. McGinty, Chem. Phys. Lett. 12, (1971) 285.

5. E. Kestemont and J. Van Craen, J. Comput. Phys. 22, (1976) 451.

6. Sources of DFT routines:
   J.P. Christiansen, R.W. Hockney, Comput. Phys. Commun. 2, (1971) 127. The program FOUR67.
   J.A. Maruhn, Comput. Phys. Commun. 12, (1976) 147. The program FORGEN.
   N.A.G. Fortran Library Manual, Vol.1, Chapter C06.
   IBM Scientific Subroutine Package (Version III), p.275. The program FORIT.
   Cray-1 Manuals 2240203 CFFT2 Routine
   2240204 RCFFT2 Routine
   2240206 CRFFT2 Routine
   by W.P. Petersen.
   Harwell Subroutine Library - the routine FT01A/D.