IB Paper 6: Signal and Data Analysis

Handout 6: The Discrete Fourier Transform

S Godsill

Signal Processing and Communications Laboratory,
Engineering Department,
Cambridge, UK

Lent 2011

Digital Sampling and the DTFT (revisited)

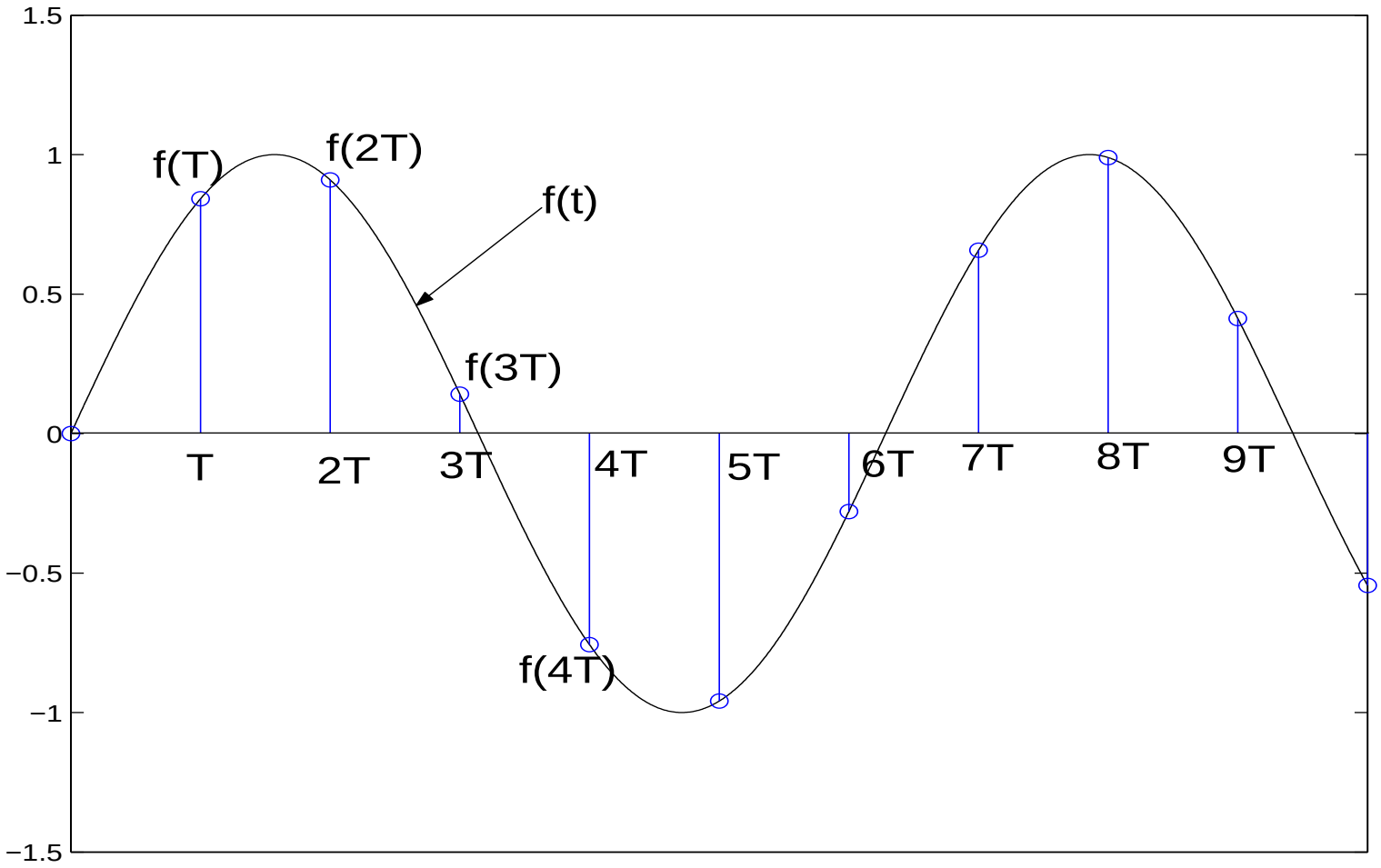

Figure 1: Digital sampling of a continuous waveform

Given the sampled values of some signal $f(t)$, i.e.

$$\{f(-\infty), \ldots f(-T), f(0), f(+T), f(2T), \ldots, f(+\infty)\}$$

the discrete time Fourier transform (DTFT) shows how to determine the frequency content:

$$F_s(\omega) = \sum_{n=-\infty}^{\infty} f(nT)\mathrm{e}^{-jn\omega T} \tag{1}$$

See handout 5. (See below for examples of DTFT of a cosine wave and DTFT of a double exponential).

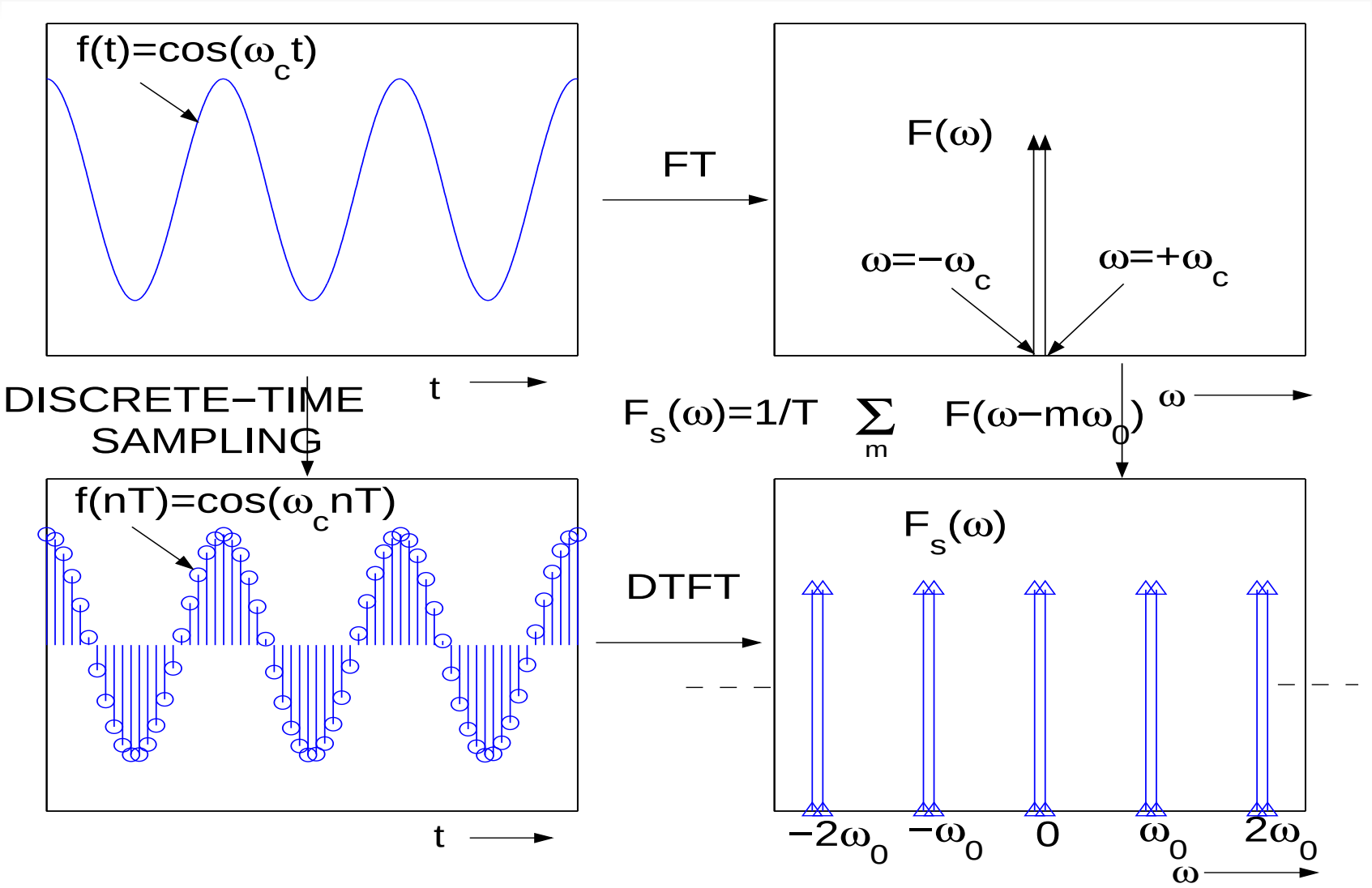

Figure 2: FT and DTFT of a cosine wave with frequency ω_c

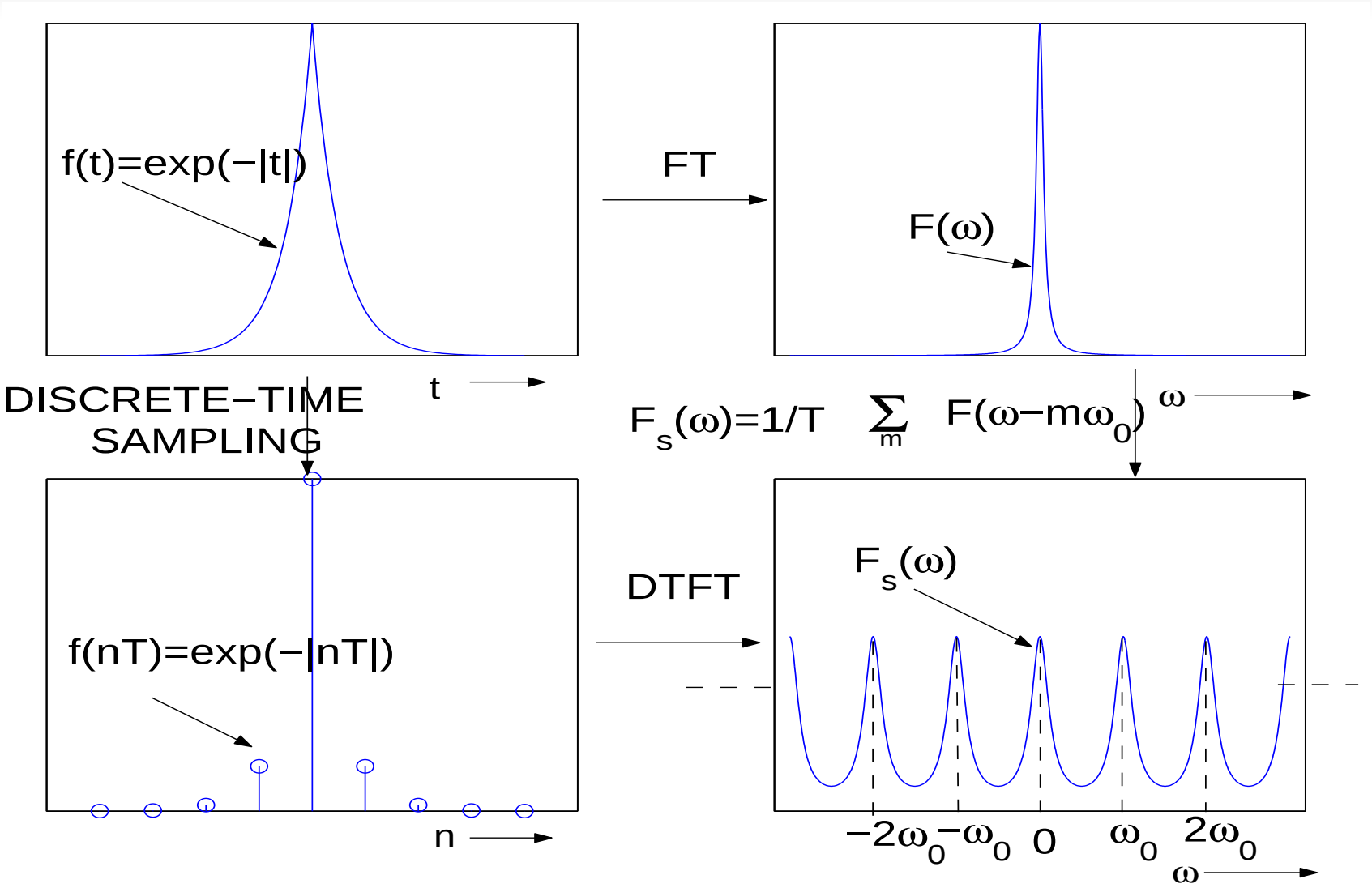

Figure 3: FT and DTFT of a double exponential

This is fine in theory, but in practice there are two reasons why you can't compute the DTFT of a signal:

- We don't have access to all of the data points from $-\infty$ to $+\infty$ (even if we did, the computing time would be infinite).
- $F_s(\omega)$ is defined over the continuous range of frequencies from $-\infty$ to $+\infty$. Hence it is impossible to calculate and store $F_s(\omega)$ for all frequency values.

The discrete Fourier transform (DFT) is a way to bypass these practical difficulties:

- **Step 1:** consider only data points which lie within some finite range, say $t = 0,\ T,\ 2T,\ 3T,\ \ldots,\ (N-1)T$. The DTFT is modified to:

$$F_s(\omega) = \sum_{n=0}^{N-1} f(nT)\mathrm{e}^{-jn\omega T} \tag{2}$$

This is equivalent to assuming that $f(nT) = 0$ outside the range $n = 0, \ldots, N-1$.
Provided we have 'enough' of the original continuous signal within the time window $0, \ldots, N-1$, the truncated DTFT will be similar to the full DTFT.

- **Step 2:** Calculate only over a finite grid of frequencies. Since $F_s(\omega)$ is periodic with period ω_0, there is no need to calculate any values beyond the Nyquist frequency. Since we are using N data points, it seems reasonable to evaluate $F_s(\omega)$ at N evenly spaced frequency values: $\omega = 0,\ \omega_0/N,\ 2\omega_0/N,\ \ldots,\ (N-1)\omega_0/N$:

$$F_s(m\omega_0/N) = \sum_{n=0}^{N-1} f(nT)e^{-jnm\omega_0 T/N}$$
$$= \sum_{n=0}^{N-1} f(nT)e^{-jnm2\pi/N}$$

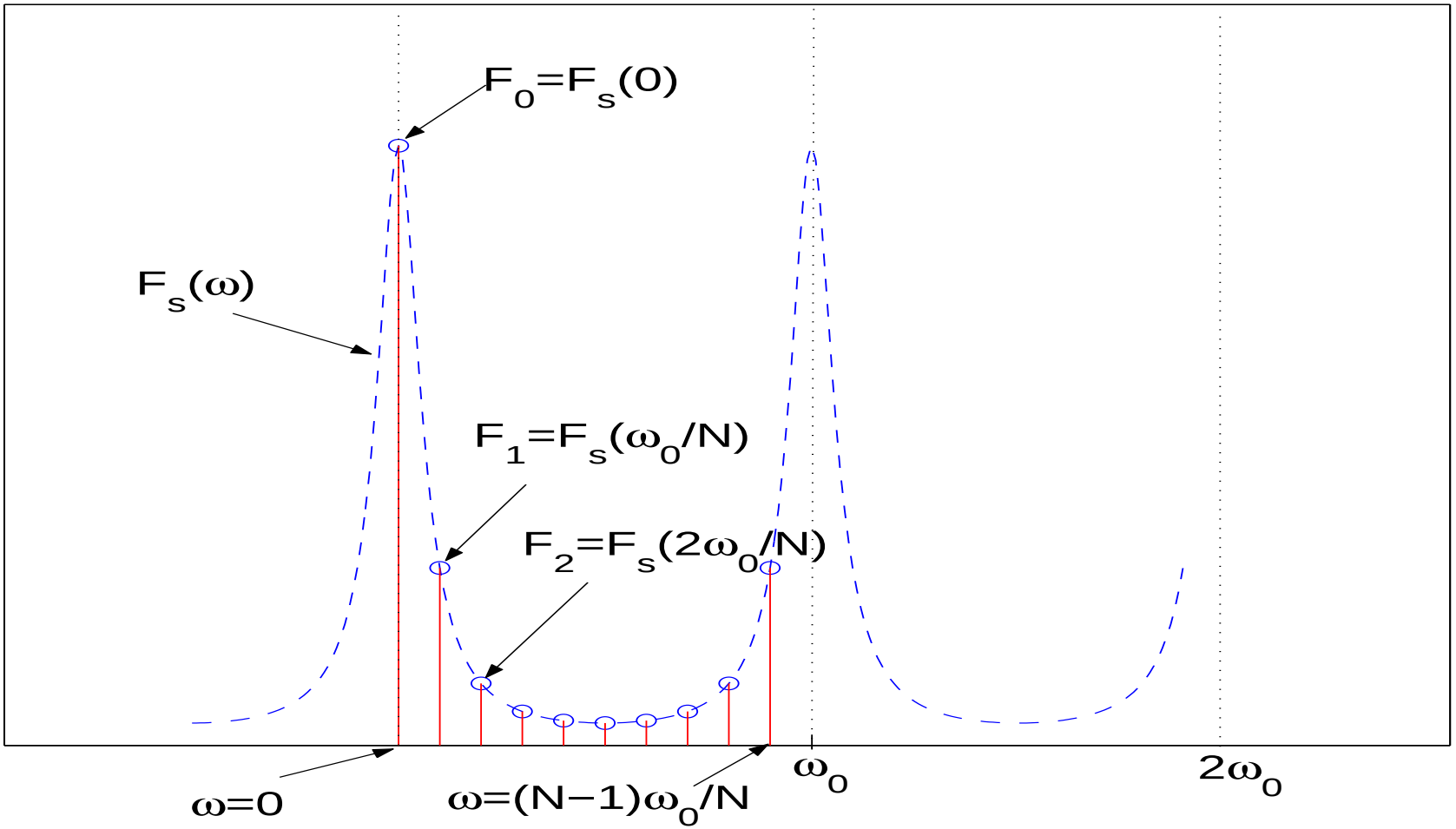

Figure 4: Sampling in frequency

This is the Discrete Fourier Transform. Define $F_m = F_s(m\omega_0/N)$ and $f_n = f(nT)$ and we have:

$$\boxed{F_m = \sum_{n=0}^{N-1} f_n e^{-jnm2\pi/N}} \tag{3}$$

Comments:

- This is now a quantity which any computer can in principle evaluate for any observed signal.
- Note that we can interpret component F_k as the frequency content of the signal at frequency $k\omega_0/N$ or $k\overline{\omega}_0$.
- The DFT is periodic, i.e. $F_{k+N} = F_k$
- For real signals $F_{-k} = F_k^*$ (Easy to check this from the DFT definition).

Inverse Discrete Fourier Transform

The grid of N frequency values chosen for the DFT is convenient in that the transform can be inverted from the N values of F_k. To derive the inverse DFT, first multiply the forward DFT by $e^{j2\pi km/N}$ and then sum over $m = 0$ to $N-1$:

$$\sum_{m=0}^{N-1} F_m e^{j2\pi km/N} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_n e^{j2\pi m\frac{k-n}{N}} \tag{4}$$

$$= \sum_{n=0}^{N-1} f_n \sum_{m=0}^{N-1} e^{j2\pi m\frac{k-n}{N}} \tag{5}$$

The inner summation over m is a Geometric Progression (GP), with factor $\rho = e^{j2\pi\frac{k-n}{N}}$.

Hence we have:

$$\sum_{m=0}^{N-1} e^{j2\pi m\frac{k-n}{N}} = \frac{1 - e^{j2\pi(k-n)}}{1 - e^{j2\pi\frac{k-n}{N}}} = \begin{cases} 0 & \text{if } k \neq n \\ N & \text{if } k = n \end{cases}$$

The second case above for $k = n$ follows immediately because $\sum_{m=0}^{N-1} e^{j0} = N$. Equation (5) now becomes

$$\sum_{m=0}^{N-1} F_m e^{j2\pi km/N} = Nf_k \qquad (6)$$

We therefore have an expression for the **Inverse Discrete Fourier Transform (IDFT)**:

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{j2\pi kn/N} \tag{7}$$

To summarise: the expressions for the DFT and the **IDFT** of a sampled signal f_n are:

$$F_k = \sum_{n=0}^{N-1} f_n e^{-j2\pi \frac{nk}{N}}, \qquad 0 \leq k \leq N-1 \tag{8}$$

$$f_n = \frac{1}{N} \sum_{k=0}^{N-1} F_k e^{j2\pi \frac{nk}{N}}, \qquad 0 \leq n \leq N-1 \tag{9}$$

Other considerations

- Fast algorithms - particularly elegant and fast methods exist for computing the DFT, known generically as the Fast Fourier Transform, or FFT algorithms. The most well-known FFT algorithms are only used when N is a power of 2. However, other classes exist, such as for N a power of 4. Study these in 3rd year module 3F3

- The effects of truncating a data sequence (ie taking a finite set of samples) and performing this periodic repetition might be to introduce some discontinuities. This will lead to high frequency components in the Fourier transform.

  One way of avoiding this is to apply a window to our data sequence. A variety of window shapes can be used which better preserve the spectral information of the original (untruncated) signal and this is a large area of study known as 'windowing' or 'tapering'.

  Because we know that multiplication in the time domain is equivalent to convolution in the frequency domain, we can quantify exactly what the effect on the spectrum is of applying this window function.

Example:

Let us now give an example of calculating a DFT. Consider the four sample values

$$\{f_n\} = \{1, 2, 1, 0\}$$

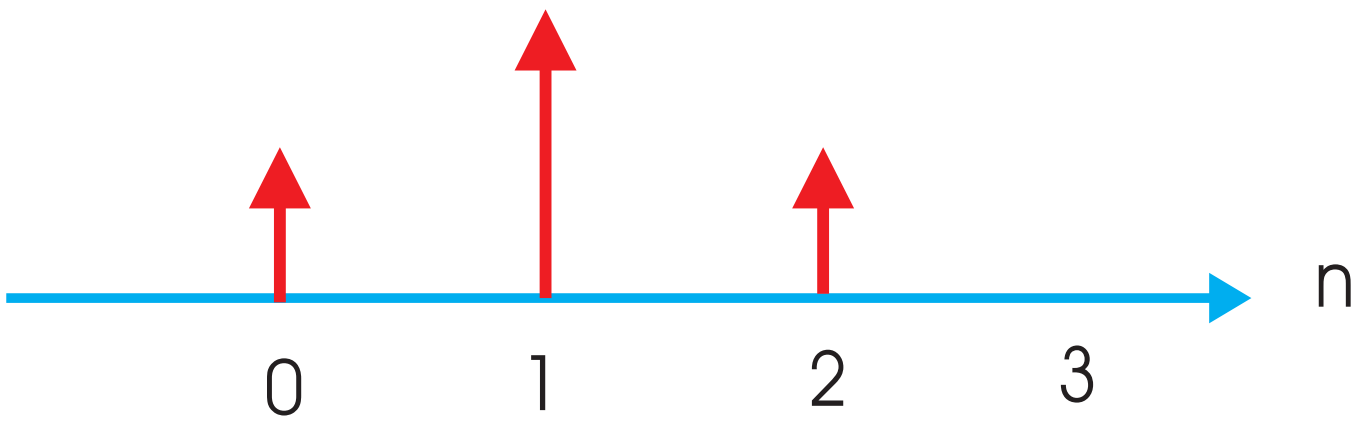

Figure 5: Time samples f_n

We now want to calculate F_k for $k = 0, 1, 2, 3$.

$$\begin{aligned} F_0 &= \{f_o e^{-j2\pi(0\times0)/4} + f_1 e^{-j2\pi(1\times0)/4} + f_2 e^{-j2\pi(2\times0)/4} \\ &+ f_3 e^{-j2\pi(3\times0)/4}\} = 1 + 2 + 1 + 0 = 4 \end{aligned}$$

$$\begin{aligned} F_1 &= \{f_o e^{-j2\pi(0\times1)/4} + f_1 e^{-j2\pi(1\times1)/4} + f_2 e^{-j2\pi(2\times1)/4} \\ &+ f_3 e^{-j2\pi(3\times1)/4}\} = 1 + 2e^{-j\pi/2} + 1e^{-j\pi} + 0 = -2j \end{aligned}$$

$$\begin{aligned} F_2 &= \{f_o e^{-j2\pi(0\times2)/4} + f_1 e^{-j2\pi(1\times2)/4} + f_2 e^{-j2\pi(2\times2)/4} \\ &+ f_3 e^{-j2\pi(3\times2)/4}\} = 1 + 2e^{-j\pi} + 1e^{-j2\pi} + 0 = 2 - 2 = 0 \end{aligned}$$

$$\begin{aligned} F_3 &= \{f_o e^{-j2\pi(0\times3)/4} + f_1 e^{-j2\pi(1\times3)/4} + f_2 e^{-j2\pi(2\times3)/4} \\ &+ f_3 e^{-j2\pi(3\times3)/4}\} = 1 + 2e^{-j3\pi/2} + 1e^{-j3\pi} + 0 = 2j \end{aligned}$$

Therefore, the DFT is given by

$$\{F_k\} = \{4, -2j, 0, 2j\}$$

You can see from this that the property $F_k^* = F_{N-k}$ is satisfied.