References

1. *The Finite Element Method, Basic Concepts and Applications*, Darrell W. Pepper and Juan C. Heinrich, Hemisphere Publishing Corporation ISBN 1-56032-104-0 (1992)

2. *Finite Element Analysis and Applications*, R. Wait and A. R. Mitchell, John Wiley & Sons, ISBN 0-471-90677-8 (1985)

3. *An Introduction to the Finite Element Method with Applications to Nonlinear Problems*, R. E. White, John Wiley & Sons, ISBN 0-471-80909-8 (1985)

Chapter 8

Digital Signal Processing

8.1 Fundamental Concepts

While the processing of signals has been done (and continues to be done) with the use of analog devices, the somewhat newer area of digital processing offers the advantage of being able to process many times, and in many different ways, a given signal. This has obvious merit in image processing, where one would like to look at the result and be able to change it according to the effect desired.

The first problem in treating the data is that, while the original signal was essentially continuous, the digital signal must be put on a discrete time mesh. Clearly some information must be lost in this procedure and we must be able to quantify and understand the errors involved. The signal is digitized by a sampling procedure where the size of the signal is measured at a number of time points spaced by T. We shall suppose that this part of the process has been carried out (normally by a special electronic processor) and that the discrete data exist at a series of points

$$-NT,\ -(N-1)T,\ \ldots,\ -T,\ 0,\ T,\ \ldots,\ (N-1)T,\ NT.$$

If the original signal is represented by $f(t)$, the sampled function is

$$f(nT),\quad -N \leq n \leq N.$$

Since we cannot use the usual (continuous) Fourier transform to obtain the spectral representation of the function (because we don't have the points at all values of the time) we shall make use of the discrete Fourier transform given by

$$F_s(\omega) \equiv T \sum_{n=-N}^{N} f(nT) e^{inwT}. \tag{8.1}$$

We may now ask in what way this discrete spectral distribution calculated from the sampled data resembles the true spectral distribution of the signal. To this end

we can replace the sampled data with the transform of its true spectral representation

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega' F(\omega') e^{-i\omega' t}. \tag{8.2}$$

Using this expression in Eq. 8.1 we have:

$$F_s(\omega) = \frac{T}{2\pi}\sum_{n=-N}^{N}\int_{-\infty}^{\infty} d\omega' F(\omega') e^{-i\omega' nT} e^{i\omega nT} \tag{8.3}$$

$$= \frac{T}{2\pi}\int_{-\infty}^{\infty} d\omega' F(\omega') \sum_{n=-N}^{N} e^{i(\omega-\omega')nT}. \tag{8.4}$$

This last sum is periodic and has the same value for

$$\omega T = \omega' T + 2\pi m; \quad m = 0,\ 1,\ -1,\ 2,\ -2, \ldots$$

Defining $\omega_s = \frac{2\pi}{T}$ we can rewrite Eq. 8.4 as:

$$F_s(\omega) = \frac{T}{2\pi}\sum_{m=-\infty}^{\infty}\int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} d\omega' F(\omega' - m\omega_s) \sum_{n=-N}^{N} e^{i(\omega-\omega')nT}. \tag{8.5}$$

If we look at the sum:

$$g(x) \equiv \sum_{n=-N}^{N} e^{ixnT} = 1 + 2\sum_{n=1}^{N}\cos(nxT) \tag{8.6}$$

we see that it is strongly peaked at x=0 (see Figure 8.1). Thus, to a good approximation, we can remove $F(\omega' - m\omega_s)$ from under the integral sign evaluated at $\omega' = \omega$. The validity of this approximation depends on the extent of the original data set. Using the exact result

$$\int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} d\omega' g(\omega - \omega') = \omega_s$$

we find:

$$F_s(\omega) = \sum_{m=-\infty}^{\infty} F(\omega - m\omega_s). \tag{8.7}$$

Considering values of ω around zero, there is some frequency at which the spectra from $m = 0$ will begin to overlap with those from $m = 1$ (see Figure 8.1). Thus, if ω_c is the value where $F(\omega)$ can be cut off (i.e. values greater than ω_c are negligible in the spectral distribution), then to avoid the problem of overlapping spectra we must have

$$\omega_s \geq 2\omega_c. \tag{8.8}$$

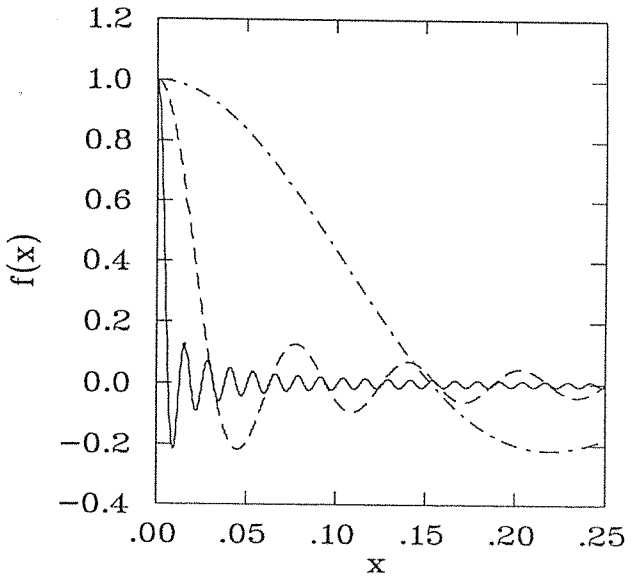

Figure 8.1: Graph of the sum 8.6 for N=20 (dash-dot curve), N=100 (dashed curve) and N=500 (solid curve).

If the sampling frequency is too low then overlap of the two regions will occur and the low and high frequencies will become confused, a condition known as aliasing. We assume that we take the maximum range possible i.e. that $\omega_s = 2\omega_c$. Given that our spectral distribution is valid within one half of the sampling frequency we may ask how to reconstruct the original signal. Using the inverse Fourier transform on $F_s(\omega)$ we find:

$$f(t) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} F_s(\omega) e^{-i\omega t} \tag{8.9}$$

$$= \frac{T}{2\pi}\sum_{n=-N}^{N} f(nT)\int_{-\omega_c}^{\omega_c} e^{i(nT-t)\omega} d\omega \tag{8.10}$$

$$= \frac{T}{2\pi}\sum_{n=-N}^{N} 2f(nT)\frac{\sin(nT-t)\omega_c}{(nT-t)} \tag{8.11}$$

$$= \sum_{n=-N}^{N} f(nT)\frac{\sin(nT-t)\omega_c}{(nT-t)\omega_c} \tag{8.12}$$

where we have used the fact that $T = \frac{2\pi}{\omega_s} = \frac{\pi}{\omega_c}$. It is clear that at the sampling points this expression gives the exact representation since all contributions vanish except for the term corresponding to the appropriate value.

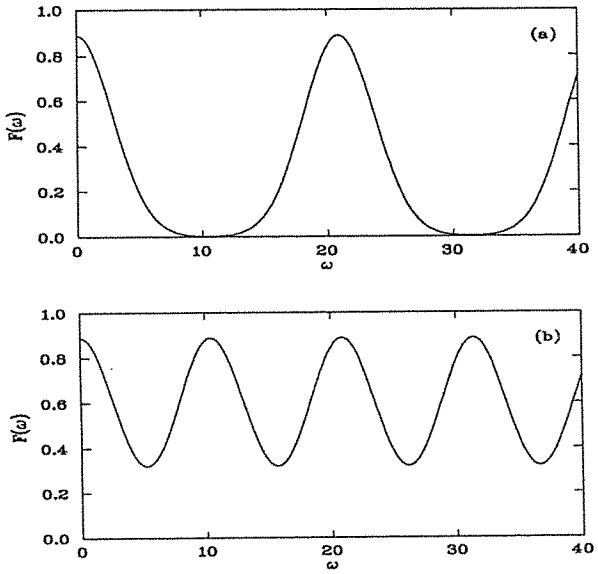

Figure 8.2: Spectrum of a Gaussian pulse in time taken from a sample with spacing 0.2 (a) and 0.4 (b) using Eq. 8.1

8.2 The Fast Fourier Transform

Suppose that we have a time series that we wish to transform to frequency space:

$$F(\omega) = \int_0^T f(t) e^{i\omega t} dt. \tag{8.13}$$

We assume that the function $f(t)$ has been digitized as:

$$f_0,\ f_1,\ f_2,\ \ldots,\ f_{N-1}$$

with a time spacing of δt.

$$t = j\delta t; \quad j = 0,\ 1,\ 2,\ \ldots,\ N-1$$

We will find it very efficient if the function $F(\omega)$ is also calculated on a mesh with a spacing $\delta\omega$

$$\omega = r\delta\omega; \quad r = 0,\ 1,\ 2,\ \ldots,\ N-1$$

such that

$$\delta\omega\delta t = 2\pi/N$$

and the integral can be expressed in terms of the nodal values (aside from the overall factor δt) as:

$$F_r = \sum_{j=0}^{N-1} f_j e^{\frac{2\pi rj}{N}} = \sum_{j=0}^{N-1} f_j q^{rj} \tag{8.14}$$

where

$$q = e^{\frac{2\pi i}{N}}.$$

Note that for $r > N-1$ q (and hence F) repeats. In particular

$$q^N = 1 \quad \text{and} \quad q^{\frac{N}{2}} = -1. \tag{8.15}$$

We will assume that N is a power of 2 and hence $n = \frac{N}{2}$ is an integer. In the sum 8.14 the product rj will often be larger than N (and n) and so we anticipate that there will be an efficient method of calculating these sums. To see how this might come about, consider the explicit representation of the sums for the case of $N = 8$. In this case $q^8 = 1$ and $q^4 = -1$.

$$\begin{aligned}
F_0 &= f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 \\
F_1 &= f_0 + f_1 q + f_2 q^2 + f_3 q^3 + f_4 q^4 + f_5 q^5 + f_6 q^6 + f_7 q^7 \\
F_2 &= f_0 + f_1 q^2 + f_2 q^4 + f_3 q^6 + f_4 q^8 + f_5 q^{10} + f_6 q^{12} + f_7 q^{14} \\
F_3 &= f_0 + f_1 q^3 + f_2 q^6 + f_3 q^9 + f_4 q^{12} + f_5 q^{15} + f_6 q^{18} + f_7 q^{21}
\end{aligned}$$

$$\begin{aligned}
F_4 &= f_0 + f_1 q^4 + f_2 q^8 + f_3 q^{12} + f_4 q^{16} + f_5 q^{20} + f_6 q^{24} + f_7 q^{28} \\
F_5 &= f_0 + f_1 q^5 + f_2 q^{10} + f_3 q^{15} + f_4 q^{20} + f_5 q^{25} + f_6 q^{30} + f_7 q^{35} \\
F_6 &= f_0 + f_1 q^6 + f_2 q^{12} + f_3 q^{18} + f_4 q^{24} + f_5 q^{30} + f_6 q^{36} + f_7 q^{42} \\
F_7 &= f_0 + f_1 q^7 + f_2 q^{14} + f_3 q^{21} + f_4 q^{28} + f_5 q^{35} + f_6 q^{42} + f_7 q^{49}
\end{aligned}$$

We see that there are a large number of powers of q greater than N and n and we could simply reduce all powers of q to 0, 1, 2 and 3 by the use of Eq. 8.15. Rather than replace them directly, we shall make a reduction in such a manner that it can be applied generally for any power of 2. For the first four equations we will simply regroup the terms.

$$\begin{aligned}
F_0 &= (f_0 + f_2 + f_4 + f_6) + (f_1 + f_3 + f_5 + f_7) \\
F_1 &= (f_0 + f_2 q^2 + f_4 q^4 + f_6 q^6) + q(f_1 + f_3 q^2 + f_5 q^4 + f_7 q^6) \\
F_2 &= (f_0 + f_2 q^4 + f_4 q^8 + f_6 q^{12}) + q^2(f_1 + f_3 q^4 + f_5 q^8 + f_7 q^{12}) \\
F_3 &= (f_0 + f_2 q^6 + f_4 q^{12} + f_6 q^{18}) + q^3(f_1 + f_3 q^6 + f_5 q^{12} + f_7 q^{18})
\end{aligned}$$

while for the second ($r \geq n$) we use Eqs. 8.15 to write:

$$\begin{array}{lll} F_4 = & (f_0 + f_2 + f_4 + f_6) & -(f_1 + f_3 + f_5 + f_7) \\ F_5 = & (f_0 + f_2 q^2 + f_4 q^4 + f_6 q^6) & -q(f_1 + f_3 q^2 + f_5 q^4 + f_7 q^6) \\ F_6 = & (f_0 + f_2 q^4 + f_4 q^8 + f_6 q^{12}) & -q^2(f_1 + f_3 q^4 + f_5 q^8 + f_7 q^{12}) \\ F_7 = & (f_0 + f_2 q^6 + f_4 q^{12} + f_6 q^{18}) & -q^3(f_1 + f_3 q^6 + f_5 q^{12} + f_7 q^{18}) \end{array}$$

If we write

$$X_m = f_0 + f_2 q^{2m} + f_4 q^{4m} + f_6 q^{6m}$$

and

$$Y_m = f_1 + f_3 q^{2m} + f_5 q^{4m} + f_7 q^{6m}$$

then

$$\begin{aligned} F_0 &= X_0 + Y_0; & F_4 &= X_0 - Y_0 \\ F_1 &= X_1 + qY_1; & F_5 &= X_1 - qY_1 \\ F_2 &= X_2 + q^2 Y_2; & F_6 &= X_2 - q^2 Y_2 \\ F_3 &= X_3 + q^3 Y_3; & F_7 &= X_3 - q^3 Y_3. \end{aligned} \tag{8.16}$$

Thus the computation has been cut approximately in half since the calculations of X_m and Y_m need be done only once. This method of reduction allows a great deal more gain however. We notice that the polynomial form of X and Y is the same as the original sum except that it is one half the length and it is a function of q^2 instead of q. If we set:

$$t = q^2 \quad \text{and} \quad f_{2j} = d_j \tag{8.17}$$

(note that $t^4 = q^8 = 1$ and $t^2 = q^4 = -1$) we have

$$\begin{array}{lllll} X_0 = & d_0 & +d_1 & +d_2 & +d_3 \\ X_1 = & d_0 & +d_1 t & +d_2 t^2 & +d_3 t^3 \\ X_2 = & d_0 & +d_1 t^2 & +d_2 t^4 & +d_3 t^6 \\ X_3 = & d_0 & +d_1 t^3 & +d_2 t^6 & +d_3 t^9 \end{array} \tag{8.18}$$

or

$$\begin{array}{lll} X_0 = & (d_0 + d_2) & +(d_1 + d_3) \\ X_1 = & (d_0 - d_2) & +t(d_1 - d_3) \\ X_2 = & (d_0 + d_2) & -(d_1 + d_3) \\ X_3 = & (d_0 - d_2) & -t(d_1 - d_3). \end{array} \tag{8.19}$$

The Y_m will be calculated in the same way with an offset in the definition of d_j.

To carry out the general case we write

$$F_r = \sum_{j=0}^{2n-1} f_j q^{rj} \tag{8.20}$$

For $r < n - 1$,

$$\begin{aligned} F_r &= \sum_{j\ even} f_j q^{rj} + \sum_{j\ odd} f_j q^{rj} \\ &= \sum_{j\ even} f_j q^{rj} + q^r \sum_{j\ even} f_{j+1} q^{rj} \\ &= X_r + q^r Y_r. \end{aligned}$$

For $r \geq n$ we can again write

$$F_r = \sum_{j\ even} f_j q^{rj} + \sum_{j\ odd} f_j q^{rj}$$

but, putting $r = n + m$,

$$\begin{aligned} F_{n+m} &= \sum_{j\ even} f_j q^{nj} q^{mj} + \sum_{j\ odd} f_j q^{nj} q^{mj} \\ &= \sum_{j\ even} f_j q^{mj} - q^m \sum_{j\ even} f_{j+1} q^{mj} \\ &= X_m - q^m Y_m \end{aligned}$$

The basic procedure at each stage is to take two inputs X_m and $q^m Y_m$ and add and subtract them to produce two outputs F_m and F_{n+m}. Then those outputs become input to the next stage. This procedure is called the "butterfly" (see figure 8.3).

Thus we can start at some power of 2 for N and keep reducing the problem until we arrive at the system of four elements. The calculation of the four elements might be programed as follows:

```
      subroutine four(t,x,k,i)
c t is the value of the basic exponential
c f is the initial sampled time series
c x is the vector of answers
c k is the current stride in the array f
c i is the starting point in the array f
      common /time/ f
      complex t,f(0:10000),x(0:3),s1,s2,s3,s4
      s1=f(i)+f(i+2*k)
      s2=f(i+k)+f(i+3*k)
      s3=f(i)-f(i+2*k)
      s4=f(i+k)-f(i+3*k)
      x(0)=s1+s2
      x(1)=s3+t*s4
```

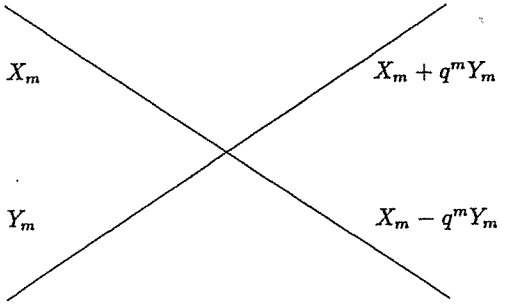

Figure 8.3: The Butterfly

```
x(2)=s1-s2
x(3)=s3-t*s4
return
end
```

Calling this program with i=0 and k=2 will calculate X in the example above and a call with i=1 and k=2 returns Y. Thus to calculate a spectrum on eight frequency points from eight time points we could use the routine "eight" with i=0 and k=1.

```
subroutine eight(q,g,k,i)
common /time/f
complex q,g(0:7),x(0:3),y(0:3),t,f(0:10000)
call four(q**2,x,2*k,i)
call four(q**2,y,2*k,i+k)
t=1.
do j=0,3
g(j)=x(j)+t*y(j)
g(j+4)=x(j)-t*y(j)
t=t*q
enddo
return
end
```

Subroutines for increasing the power of two are nearly the same.


```
subroutine sixteen(q,g,k,i)
common /time/f
complex q,g(0:15),x(0:7),y(0:7),t,f(0:10000)
call eight(q**2,x,2*k,i)
call eight(q**2,y,2*k,i+k)
t=1.
do j=0,7
g(j)=x(j)+t*y(j)
g(j+8)=x(j)-t*y(j)
t=t*q
enddo
return
end
```

The highest level routine is always called with $i = 0$ and $k = 1$. One can continue to write such subroutines as large as the size desired.

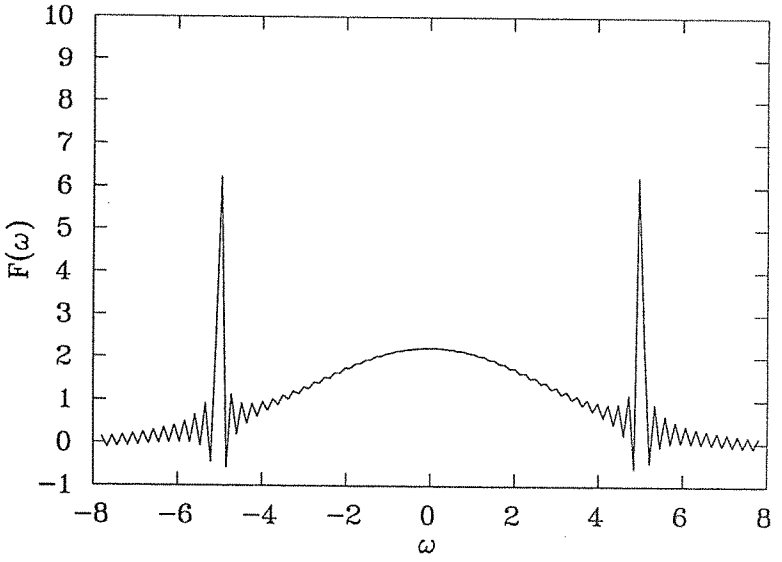

Figure 8.4: Spectral Transform for the Conditions of the Problem