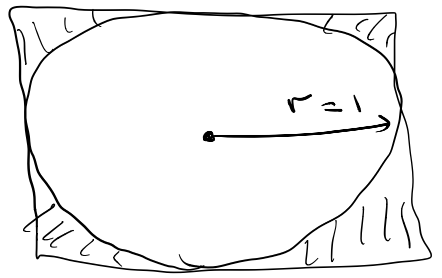
Un paquete que genera números aleatorios:

https://people.sc.fsu.edu/~jburkardt/c_src/rnglib/rnglib.html

Introducción a Monte Carlo

$\pi = r^2$ área de un círculo

Cuadrado de lado l



Insertamos un círculo de radio 1 en un cuadrado

$(\pm 1, \pm 1)$ vertices del cuadrado

$\text{Area} = l \times l$

$\pi = \text{Area del cuadrado} = l \times l$

$x^2 + y^2 = r^2 = (1)^2$

$x^2 + y^2 = 1$

$\pi = \int dx \int dy$

$y = \pm\sqrt{1-x^2}$

$$\boxed{\pi = \int_{-1}^{1} dx \int_{-\sqrt{1-x^2}}^{+\sqrt{1-x^2}} dy}$$

Seleccionar un número de puntos aleatorios que caigan dentro del círculo, la probabilidad de que caigan dentro del círculo es igual al cociente de lás areas, en este caso es $\boxed{\frac{\pi}{4}}$.

$$\text{Probabilidad}: \frac{\pi / r^2}{l * l} = \frac{\pi / (1)^2}{(2) * (2)} = \frac{\pi}{4}$$

Tenemos que generar números aleatorios entre 0 y 1:

Esto lo genera la subroutina RAN3 :

Chapter 2

Introduction to Monte Carlo

How Monte Carlo techniques are used for the solution of physical problems is not well understood by many. A general belief is that they are designed to treat statistical (and other) problems based on probability. While this is certainly, true the techniques are much more powerful and their applications are much more general than this view would imply.

In its purest form, one could perhaps describe the method in terms of a relationship between the geometric and algebraic aspects of theoretical science. Over the years powerful methods have been developed to express the solution of probabilistic problems in analytical terms. With the advent of powerful computers the statistical problems can be modeled directly, opening the possibility of inferring the solution to the analogous analytical problem, thus reversing the originally intended direction. The solution to any problem which has an analytical expression that "looks like" the solution to a probability problem can be obtained by constructing a realization of the random variable problem. A large number of problems seem to fall into this category. This explanation is perhaps a bit too abstract for an introduction. It is hoped that this point will be clearer after the example in the following section.

2.1 Preliminary Notions - - Calculating π

Since it is perhaps not clear how one actually goes about calculating something with random numbers, a simple example is presented in this section to give the reader a feeling for the method. One widespread application of Monte Carlo methods involves calculating multidimensional integrals. We start out in this introduction with a simple two-dimensional integral.

Consider a square with corners at $(\pm 1, \pm 1)$ and a circle inscribed within this square. The square has sides of length 2 and area 4, and the circle a radius of 1 and

area π. The area of the circle can be calculated analytically with the double integral:

$$\pi = \int_{-1}^{1} dx \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} dy. \tag{2.1}$$

We might imagine a problem in which it was necessary to evaluate the double integral on the right hand side of Eq. 2.1.

If we select a number of points at random, distributed uniformly in the square, we know that the probability that they will fall within the circle is equal to the ratio of the areas, in this case $\pi/4$. The computer can choose points at random in the ranges $-1 \le x \le 1$ and $-1 \le y \le 1$ and so arrive at points randomly distributed within the square*.

Under these conditions it is then a simple matter to test if a given point lies within the circle ($x^2 + y^2 \le 1$). The fraction of points that fall within the circle will be $\pi/4$ Following is a program which calculates π in this manner.

```
  NMC=1000            ! number of Monte Carlo throws
  C=0.                ! zero the number of hits in the circle
  DO 1 I=1,NMC        ! throw NMC points
  X=2.*RAN3(D)-1.     ! RAN3 gives numbers between 0 and 1 so
  Y=2.*RAN3(D)-1.     ! multiply by 2 and subtract 1 to get -1,1
  IF(X**2+Y**2.GT.1.) GOTO 1   ! if outside of circle don't
  C=C+1.              ! count another hit in circle
1 CONTINUE            ! end of MC loop
  PRINT *,4.*C/NMC    ! print answer
  END
```

This short code illustrates three of the principal characteristics of Monte Carlo integrals.

1. The program is relatively easy to write once the algorithm is understood.

2. The technique is simple to extend to higher dimensions. In this case to extend the calculation to 3 dimensions (to compute the ratio of the volume of a sphere to that of a cube, $\frac{4\pi}{3\cdot 8} = \frac{\pi}{6}$) we only need to add the statement "`Z=2.*RAN3(D)-1.`" and alter the "`IF`" statement to read "`IF(X**2+Y**2+Z**2.GT.1.)GOTO 1`".

*At the base of all Monte Carlo calculations lies a subroutine called the "random number generator" which calculates a random number uniformly distributed between 0 and 1. Here we use such a program, written in Fortran so it is transportable, called RAN3. It must be called with an argument so that Fortran knows that it is a subroutine, and not a variable, but this argument plays no other role. See the Appendix, Section A.4 for this subroutine and a description of its limitations.

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.

8.2 Sampling: Nyquist Theorem

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 is 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 δ. 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

$$-N\delta,\ -(N-1)\delta,\ \ldots,\ \delta,\ 0,\ \delta,\ \ldots,\ (N-1)\delta,\ N\delta.$$

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

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

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

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

We may now ask how well 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

$$F_s(\omega) = \frac{\delta}{2\pi}\sum_{n=-N}^{N}\int_{-\infty}^{\infty} d\omega' F(\omega') e^{-i\omega' n\delta} e^{i\omega n\delta} \tag{8.3}$$

$$= \frac{\delta}{2\pi}\int_{-\infty}^{\infty} d\omega' F(\omega') g(\omega - \omega') \tag{8.4}$$

where

$$g(\omega - \omega') = \sum_{n=-N}^{N} e^{i(\omega-\omega')n\delta}. \tag{8.5}$$

The function, $g(\omega - \omega')$, is periodic and has the same value for

$$\omega\delta = \omega'\delta + 2\pi m; \qquad m = 0,\ 1,\ -1,\ 2,\ -2, \ldots$$

Defining $\omega_s = \frac{2\pi}{\delta}$ we see that

$$g(\omega - \omega' + m\omega_s) = g(\omega - \omega'); \;\; m = 0,\ -1,\ 1,\ -2,\ 2, \ldots \tag{8.6}$$

so that we can break the integral into segments,

$$F_s(\omega) = \frac{\delta}{2\pi}\sum_{m=-\infty}^{\infty}\int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} d\omega' F(\omega' - m\omega_s) g(\omega - \omega'). \tag{8.7}$$

If we look at the sum

$$g(x) \equiv \sum_{n=-N}^{N} e^{ixn\delta} = 1 + 2\sum_{n=1}^{N}\cos(nx\delta) \tag{8.8}$$

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

$$F_s(\omega) = \sum_{m=-\infty}^{\infty} F(\omega - m\omega_s). \tag{8.9}$$

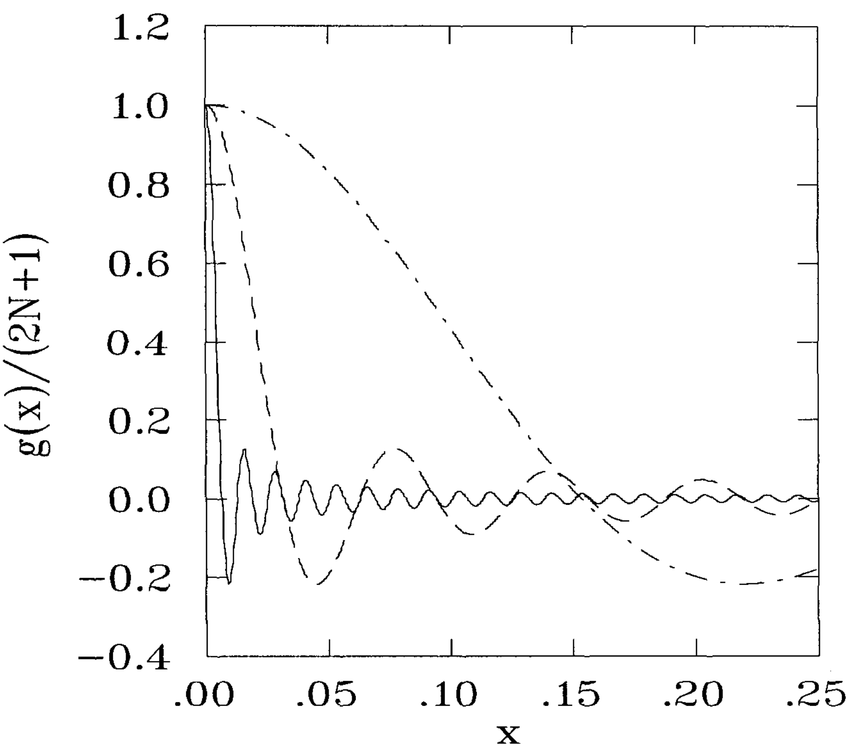


Fig. 8.1 Graph of the sum in Eq. 8.8 for N=20 (dash-dot curve), N=100 (dashed curve) and N=500 (solid curve).

Considering values of ω around zero, there is some frequency at which the spectra from $m = 0$ will begin to overlap with those from $m = \pm 1$ (see Figure 8.2). 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

$$\omega_s \geq 2\omega_c. \tag{8.10}$$

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

$$f(t) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} F_s(\omega)e^{-i\omega t} = \frac{\delta}{2\pi}\sum_{n=-N}^{N} f(n\delta)\int_{-\omega_c}^{\omega_c} e^{i(n\delta-t)\omega}d\omega \tag{8.11}$$

$$= \frac{\delta}{2\pi}\sum_{n=-N}^{N} 2f(n\delta)\frac{\sin(n\delta - t)\omega_c}{(n\delta - t)} = \sum_{n=-N}^{N} f(n\delta)\frac{\sin(n\delta - t)\omega_c}{(n\delta - t)\omega_c} \tag{8.12}$$

where we have assumed that $\delta = \frac{2\pi}{\omega_s} = \frac{\pi}{\omega_c}$.

If the function is evaluated at one of the input values, say $t = m\delta$ then we have

$$f(t) = \sum_{n=-N}^{N} f(n\delta)\frac{\sin(n - m)\delta)\omega_c}{(n - m)\delta\omega_c} \tag{8.13}$$

so if

$$\text{if } \ n = m \ \text{ (in the limit) } \ \frac{sin(n - m)\delta\omega_c}{(n - m)\delta\omega_c} \to 1 \tag{8.14}$$

$$\text{if } \ n \neq m \ \ \frac{sin(n - m)\delta\omega_c}{(n - m)\delta\omega_c} = \frac{sin(n - m)\pi}{(n - m)\pi} \to 0 \tag{8.15}$$

Hence, at a sampled point this expression gives the exact representation since all contributions vanish except for the term corresponding to the appropriate value.

8.3 The Fast Fourier Transform

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

$$F(\omega) = \int_0^T f(t)e^{i\omega t}dt. \tag{8.16}$$

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 δ.

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

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

$$\omega = r\epsilon; \qquad r = 0, \ 1, \ 2, \ \ldots, \ N - 1$$

such that

$$\epsilon\delta = 2\pi/N$$

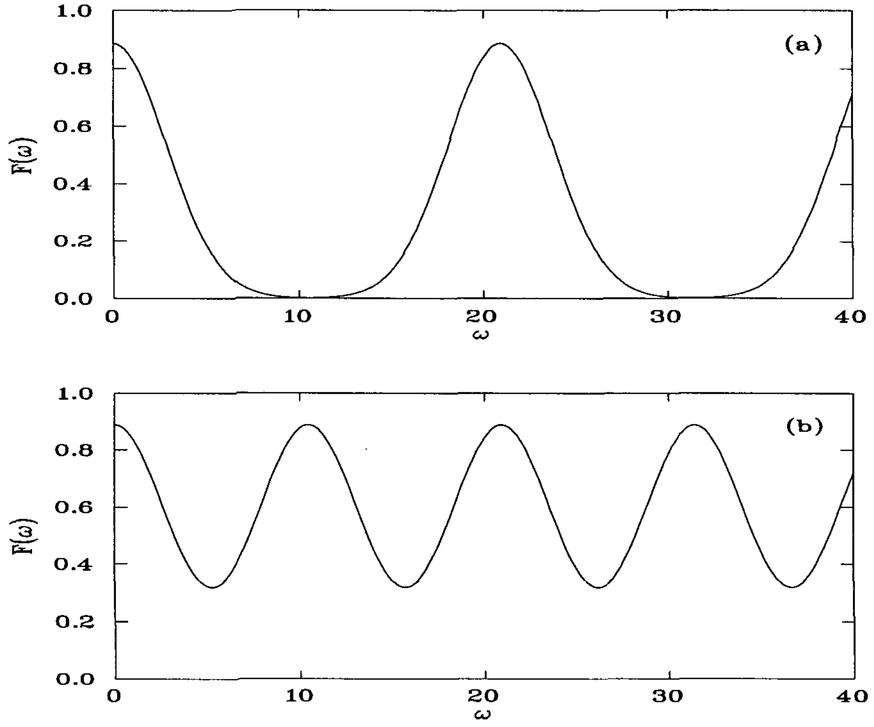


Fig. 8.2 Spectrum of a Gaussian pulse in time taken from a sample with spacing 0.2 (a) and 0.4 (b) using Eq. 8.1

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

$$F_r = \sum_{j=0}^{N-1} f_j e^{\frac{2\pi r j}{N}} = \sum_{j=0}^{N-1} f_j q^{rj} \tag{8.17}$$

where

$$q = e^{\frac{2\pi i}{N}}. \tag{8.18}$$

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

$$q^N = 1 \quad \text{and} \quad q^{\frac{N}{2}} = -1. \tag{8.19}$$

We will assume that N is a power of 2 and hence $n = \frac{N}{2}$ is an integer. In the sum 8.17 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. 8.19. 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. 8.19 to write

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

If we define

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

and

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

then

$$\begin{aligned}
F_0 &= X_0 + Y_0; & F_4 &= X_0 - Y_0\\
F_1 &= X_1 + qY_1; & F_5 &= X_1 - qY_1\\
F_2 &= X_2 + q^2 Y_2; & F_6 &= X_2 - q^2 Y_2\\
F_3 &= X_3 + q^3 Y_3; & F_7 &= X_3 - q^3 Y_3.
\end{aligned} \tag{8.20}$$

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:

$$t = q^2 \quad \text{and} \quad f_{2j} = d_j \tag{8.21}$$

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

$$\begin{aligned} X_0 &= d_0 + d_1 + d_2 + d_3 \\ X_1 &= d_0 + d_1 t + d_2 t^2 + d_3 t^3 \\ X_2 &= d_0 + d_1 t^2 + d_2 t^4 + d_3 t^6 \\ X_3 &= d_0 + d_1 t^3 + d_2 t^6 + d_3 t^9 \end{aligned} \tag{8.22}$$

or

$$\begin{aligned} X_0 &= (d_0 + d_2) + (d_1 + d_3) \\ X_1 &= (d_0 - d_2) + t(d_1 - d_3) \\ X_2 &= (d_0 + d_2) - (d_1 + d_3) \\ X_3 &= (d_0 - d_2) - t(d_1 - d_3). \end{aligned} \tag{8.23}$$

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

$$F_r = \sum_{j=0}^{2n-1} f_j q^{rj} \tag{8.24}$$

For $r < n - 1$,

$$F_r = \sum_{j\ even} f_j q^{rj} + \sum_{j\ odd} f_j q^{rj} = \sum_{j\ even} f_j q^{rj} + q^r \sum_{j\ even} f_{j+1} q^{rj}$$

$$= X_r + q^r Y_r.$$

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 $r = n + m$,

$$F_{n+m} = \sum_{j\ even} f_j q^{nj} q^{mj} + \sum_{j\ odd} f_j q^{nj} q^{mj} = \sum_{j\ even} f_j q^{mj} - q^m \sum_{j\ even} f_{j+1} q^{mj}$$

$$= X_m - q^m Y_m$$

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).

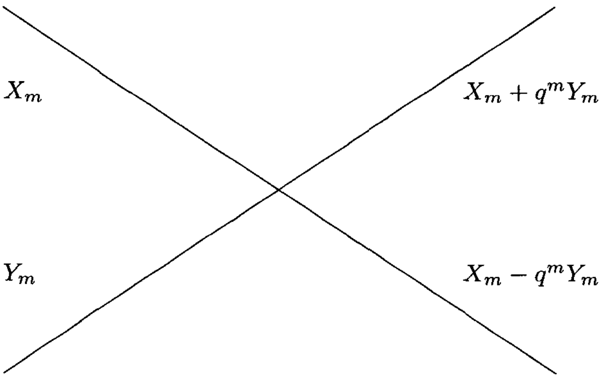


Fig. 8.3 The butterfly.

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 programmed 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
      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. 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.

8.4 Phase Problems

The reader will have noticed that in Section 8.2 we considered a symmetric placement of the points about (and including) zero which led to an odd number of points. The fast Fourier transform requires that the number of points be a power of two and hence even. If we want to make the transform symmetric and contain an even number of points then the origin will not be on the mesh. We can easily construct a mesh which satisfies these conditions on half integral values of the spacing,

$$t(j) = (j + \frac{1}{2})\delta; \quad j = -\frac{N}{2} \text{ to } \frac{N}{2} - 1 \tag{8.25}$$

but this process introduces an unwanted phase into the resultant spectrum. If one only wants the absolute value of the Fourier transform there is no problem, but if the full spectrum is needed, a correction will need to be applied. A particular case is the calculation of the Fourier transform of a symmetric real

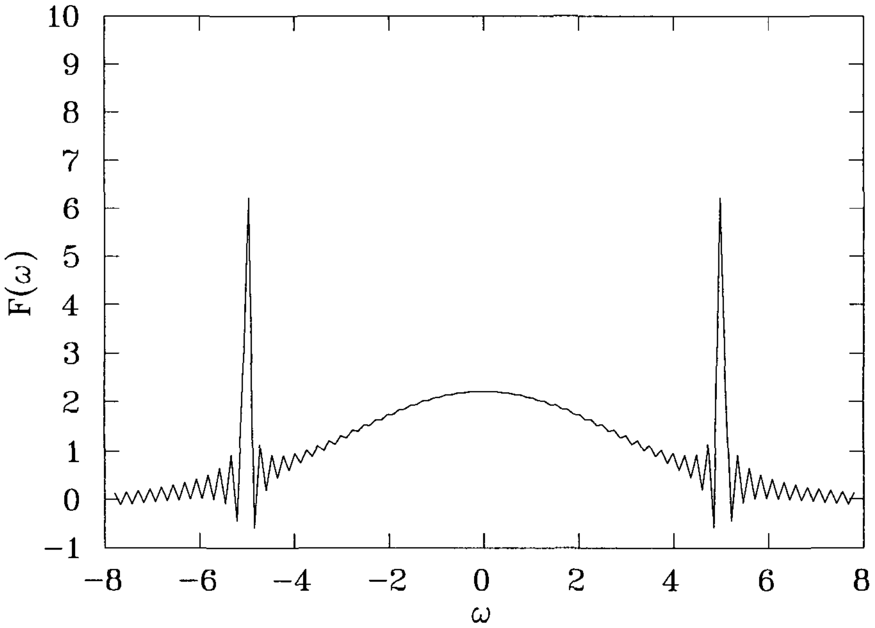


Fig. 8.4 Spectral transform for the conditions of the problem.

function. In this instance the transform itself should also be real but will not be without this correction (see the problem at the end of this chapter).

We want to calculate

$$F(\omega) = \sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} e^{i\omega(\frac{1}{2}+j)\delta} f[(\frac{1}{2}+j)\delta] \tag{8.26}$$

or, in terms of $\omega = r\epsilon$,

$$F_r = \sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} e^{ir\epsilon(\frac{1}{2}+j)\delta} f[(\frac{1}{2}+j)\delta] \tag{8.27}$$

Using the relation between ϵ and δ

$$F_r = \sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} e^{ir\frac{2\pi}{N}(\frac{1}{2}+j)} f[(\frac{1}{2}+j)\delta] \tag{8.28}$$

Changing the j index to $k = j + \frac{N}{2}$ we have

$$F_r = \sum_{k=0}^{N-1} e^{ir\frac{2\pi}{N}(\frac{1}{2}-\frac{N}{2}+k)} f[(\frac{1}{2} - \frac{N}{2} + k)\delta] \tag{8.29}$$

Now, labeling the time sequence starting from the first value with index zero

$$F_r = e^{-ir\frac{\pi(N-1)}{N}} \sum_{k=0}^{N-1} e^{ir\frac{2\pi}{N}k} f_k \tag{8.30}$$

Since the FFT calculates the sum, the transform we want can be obtained by including the premultiplying factor.

Problem

(1) Continue the construction of the FFT subroutines up to 128. Sample the function

$$f(t) = e^{-4t^2} + .1\cos 5t$$

in steps of δ=0.4 for points symmetric about $t = 0$ (i.e. at $\pm\frac{\delta}{2}$, $\pm\frac{3\delta}{2}$, etc. Use the FFT routines to calculate the spectral distribution.

Note that the initial function is real and symmetric so that we expect to obtain a real spectrum with a distribution of frequencies for the transform of the Gaussian function and sharp peaks at ± 5 for the cosine. To assure that the spectrum is real it is necessary to apply the correction discussed Section 8.4. Your result should look like Figure 8.4.

References

(1) *Introduction to Discrete-time Signal Processing*, Steven A. Tretter, John Wiley & Sons (1976) ISBN 0-471-88760-9

(2) *Digital Picture Processing*, Azriel Rosenfeld and Avinash C. Kak, Academic Press, (1976) ISBN 0-12-597360-8