Appendix 2

The Fourier Transform

The Fourier transform is a subject that is usually reserved for advanced undergraduate mathematics courses and often doesn't sneak into the physics curriculum until somewhat late in the game. This is unfortunate, since the basic ideas are not difficult to grasp, and it is a very useful tool in many problems. Our goal in this appendix is to give a general introduction to the Fourier transform so as to provide a basis for the ideas and techniques we need for the problems discussed in this book.

A2.1 Theoretical Background

On the left of Figure A2.1 we show a hypothetical signal. It is simply a function that describes how some quantity varies with time. This quantity might be the intensity of a sound wave, the displacement of a particular part of a vibrating string, or the voltage at some point in an electronic circuit. To the eye, this particular signal appears to have an oscillatory character. Thus it should not come as a shock to learn that it was constructed by adding the five individual sine waves shown on the right in Figure A2.1. This signal $y(t)$ can thus be written as

$$y(t) = \sum_{j=1}^{5} y_j \sin(2\pi f_j t + \phi_j), \tag{A2.1}$$

where y_j is the amplitude, f_j the frequency, and ϕ_j the phase of the jth sine wave component.

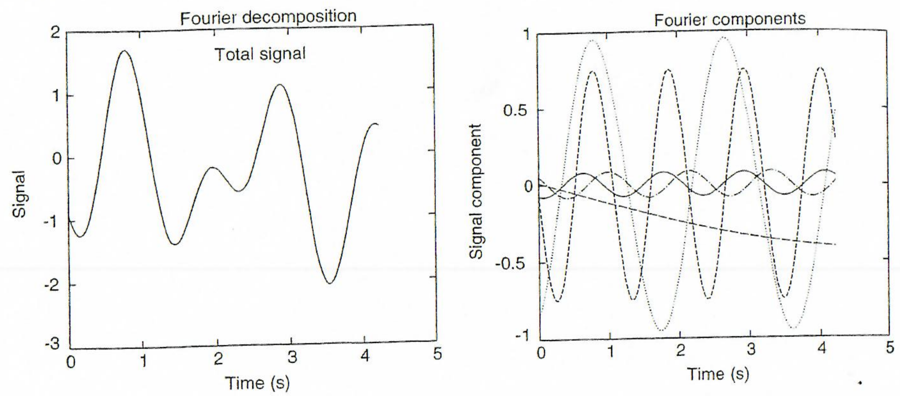

Figure A2.1: Left: hypothetical signal; right: individual sine waves whose sum yields the signal at left.

The total signal in Figure A2.1 is a very simple one, so it is probably not surprising that it can be "decomposed" into a collection of sine waves. However, it was shown by Joseph Fourier nearly 200 years ago that virtually *any* signal can be written in this way.[1] That is, *any* function, $y(t)$, can be written as a sum of sine waves. Most signals will be more complicated than the one in Figure A2.1, so the sum may involve a large (perhaps infinite) number of sine waves, but just the guarantee that such a sum exists can be extremely helpful. It is convenient to express (A2.1) as an integral over frequency, which is usually written in the form

$$y(t) = \int_{-\infty}^{\infty} Y(f)\, e^{-2\pi \iota f t}\, df\,, \tag{A2.2}$$

where $\iota = \sqrt{-1}$. The operation in (A2.2) is known as a Fourier transform. The factor $e^{-2\pi\iota f t}$ in this expression is just the sum $[\cos(2\pi f t) - \iota \sin(2\pi f t)]$. Hence, in the most general case the Fourier transform function $Y(f)$ will be complex.

While (A2.2) may have a formidable appearance, it is really just a sum of sines of cosines. It is common to refer to $y(t)$ as a function in the time domain and to its transform $Y(f)$ as existing in the frequency domain.[2] We move between these parallel universes with the Fourier transform. The so-called forward transform (A2.2) takes functions in one direction while the inverse transform,

$$Y(f) = \int_{-\infty}^{\infty} y(t)\, e^{2\pi \iota f t}\, dt\,, \tag{A2.3}$$

[1] Strictly speaking, this claim does not apply to extremely pathological signals. We will leave a discussion of such functions to the mathematicians (and sources in the references).

[2] It can also be useful to consider the Fourier transform of a function that is a function of space. In such cases the variables become space (that is, distance) and wave vector.

takes functions in the opposite direction. Thus, our signal can be described equally well by either function, $y(t)$ or $Y(f)$.

While functions can be written in the form (A2.2), it remains to be seen why we would want to do this in the first place. As an example, let us assume that we are dealing with an electronically recorded sound signal. The fact that it can be written as a Fourier sum means that it can be viewed as (or decomposed into) a sum of "pure" tones (or sinusoids). The frequencies of these tones are the only frequencies present in the original signal, and the Fourier transform $Y(f)$ gives a direct measure of the frequencies that are present. Most sound signals are composed of many such frequencies. In some cases these components may be harmonically related, but things can also be more complicated. In any event, the Fourier transform gives a convenient way to view a signal in the frequency domain, and we will find it useful on many occasions.

Now assume that we have used the Fourier transform to separate the components of a particular signal. What can we do with these besides perform a spectral analysis? Suppose you want to play this sound signal through the speakers in your high-fidelity system. In order to determine how the speakers will respond to this signal, we can first calculate how they would respond to each Fourier component *as if* that component was the *only* signal applied to the speakers. In many physical systems of interest, the total response is just the sum of the individual responses to each component. This approach, which relies on the linearity of the system (in this case the speakers), is intimately linked to the principle of superposition and can be used in a wide variety of problems, including waves on a string (Chapter 6), quantum mechanics (Chapter 10), and heat flow (which is the problem Fourier first treated with this method).

The next issue to consider is how to actually compute a Fourier transform. That is, given the function $y(t)$, how *in practice* do we determine $Y(f)$?

A2.2 The Fast Fourier Transform (FFT)

Suppose that a signal is specified analytically; that is, you are given the functional form of $y(t)$. The Fourier transform $Y(f)$ could then be calculated by simply performing the integral (A2.3). Of course, this may not be easy, which is why many mathematics texts have been written on the topic. However, in numerical work we are almost never given the analytic form of the signal, but instead have knowledge of its amplitude at certain discrete values of t. It is often the case that the values of $y(t)$ are known (or given) at evenly spaced intervals. For example, in our simulations of the pendulum we calculated the angular position at times $t_j = j\,\Delta t$, where j was an integer and Δt was the time step. In such situations it is useful to define the discrete Fourier transform [compare with (A2.2) and (A2.3)]

$$y_j = \frac{1}{N}\sum_{k=0}^{N-1} Y_k\, e^{-2\pi\iota jk/N} \tag{A2.4}$$

$$Y_k = \sum_{j=0}^{N-1} y_j\, e^{2\pi\iota jk/N}\,.$$

Here we follow the usual convention and let the indices on y and Y run from 0 to $N-1$, where N is the number of data points. Again we can think of our data in the time domain where we have the values

y_j, or the frequency domain with Y_j. These are two equivalent ways of describing the *same* collection of data points.

It is important to note that the time step does not enter directly into the discrete transform (A2.4). This disappearance is possible because the signal values y_j were obtained at points equally spaced in time. Hence, the corresponding time values are fully specified by the index j and the time step Δt. It turns out that the frequency associated with each Y_j is $f_j = j/N\Delta t$. However, there are some subtleties here that we will come to in due course.

Returning to (A2.4), if we have N data points, that is, N values y_j, then there are N values of Y_j. That is, we have N pieces of information either way. However, this is not quite the whole story. Both the signal y_j and its transform Y_j in (A2.4) are, in general, *complex* numbers. Of course, most physical signals can be expressed as real numbers and in such cases the y_j are real. However, even here the Y_j can be complex. This can be understood in the following way. The exponential factors in (A2.4) are just a shorthand for the sum $[\cos(2\pi jk/N) \mp \iota \sin(2\pi jk/N)]$. The imaginary factor then causes Y_j to be complex. The real and imaginary parts of Y_j correspond to what are known as the cosine and sine transforms.

Since the Y_k are complex, it appears that we have $2N$ pieces of information in the frequency domain. If the y_k are real, we have only N pieces of information in the time domain. But if these are describing the same function, they must contain the same amount of information. The resolution of this apparent paradox is that the Y_j are then not all independent. We will have more to say about such issues in a moment.

Now we want to return to the question of the frequencies associated with each Y_j. We just noted that $f_j = j/N\Delta t$. Since the values of Y_j with $j \geq N/2$ will turn out to be redundant, the highest frequency Fourier component is $Y_{N/2-1}$. The special frequency $1/2\Delta t$ is known as the Nyquist frequency and plays a very important role. If a signal is measured at time intervals spaced by Δt, then the spectral components that can be recovered with a Fourier transform are those with frequencies below $f_{\mathrm{Nyquist}} \equiv 1/2\Delta t$. If our signal were a simple sine wave at the Nyquist frequency, then we would be sampling it only twice during each period of oscillation. The amazing thing is that sampling *only* twice each period is sufficient to capture this Fourier component. This result is known as the sampling theorem. We will illustrate this and other properties of the discrete Fourier transform below. In particular, we will consider what happens if our signal contains components at frequencies above the Nyquist frequency.

The discrete Fourier transform (A2.4) is just a sum of exponential terms, so it *appears* to be very amenable to numerical evaluation. However, straightforward evaluation of the sums in (A2.4) is computationally very expensive. Each term involves the computation of the exponential factor, which must then be multiplied by y_j and added to the running total. Each sum has N terms and there are N data points, so the total number of operations is of order N^2. This is bad. It turns out that even with a very fast computer, this brute force approach would take a prohibitively long time for typical values[3] of N. For this reason a conventional numerical approach to evaluating discrete Fourier transforms is not practical.

While the simplest approach to evaluating (A2.4) would require of order N^2 operations, this does not mean that *all* approaches must involve the same number of operations. The exponential terms in (A2.4) are multiples of one another, and this makes it possible to "reuse" many of the terms in the sum. In fact, it is possible to evaluate the discrete transform with only of order $N \log N$ operations. This can

[3] $N \sim 10^6$ is not uncommon in many applications.

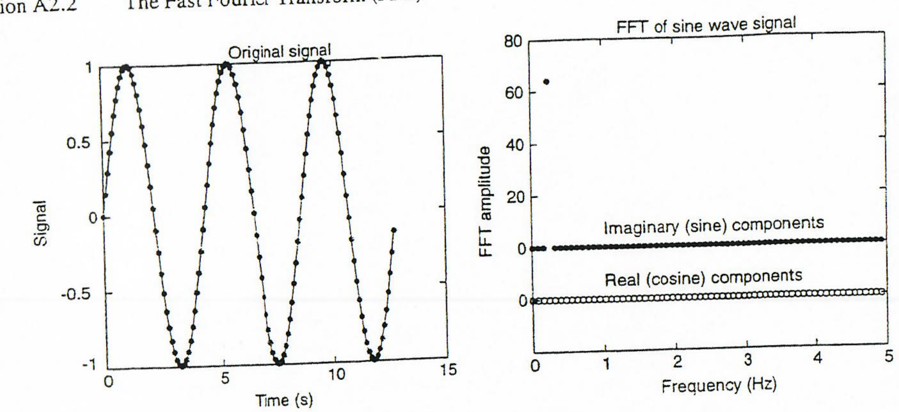

Figure A2.2: Left: pure sine wave signal, that is, $y(t) = \sin(2\pi f t)$. These 128 points are the values of the signal that were Fourier analyzed. Right: FFT of this signal. The real (cosine) and imaginary (sine) parts of the transfrom are shown seperately. We have plotted them as discrete points to emphasize that the number of Fourier amplitudes yielded by FFT is equal to the number of original data points. Hence we have 64 cosine and 64 sine components. Only one of the points in the transform is nonzero.

be a major time reduction for large transforms, and the savings are substantial even for $N \sim 1000$, a value we will find useful for the problems in this book. There are several specific algorithms of this kind, and they are known as *the fast Fourier transform* or *FFT*. The existence of the FFT has made many important calculations feasible, and it is used in technologies such as X-ray tomography. The FFT algorithm is sufficiently complicated that we will not give a full explanation here (see the references). However, in Appendix 4 we give the listing of a Fourier analysis program that employs the FFT.

To get a feeling for how Fourier analysis works in practice, we now consider a few examples. Perhaps the simplest possible signal is a pure sine wave, such as the one shown in Figure A2.2. The period here is approximately 4.3 s and has been chosen so that the total recorded signal is precisely three complete periods.[4] Note also that the 128 signal values used in the analysis[5] are the points plotted in Figure A2.2. The FFT of this signal is shown on the right in Figure A2.2. The results are 128 Fourier amplitudes, half of which are the amplitudes of the component sine waves and half of which are the cosine amplitudes. We see that all of these are zero, *except one*, the one corresponding to the sine wave we started with. Hence, the FFT tells us that our signal is composed of a single Fourier component corresponding to a sine wave with a frequency of ≈ 0.23 Hz.

[4] While not absolutely necessary, we will employ the units of seconds (s) and (hertz) Hz in the following discussion, as an aid in appreciating the connection between the time and frequency domains.

[5] It turns out that the algorithm that actually evaluates the FFT requires that the number of data points it receives be a power of 2. See the discussion of our FFT program for more on this.

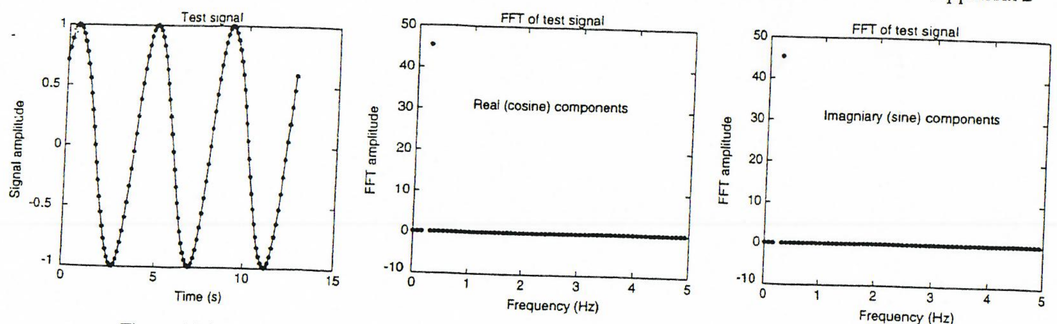

Figure A2.3: Left: test signal that is just a sine wave that is phase shifted by $\pi/4$. The dots are, again, the data points that were used in the FFT; center and right: FFT of this test signal.

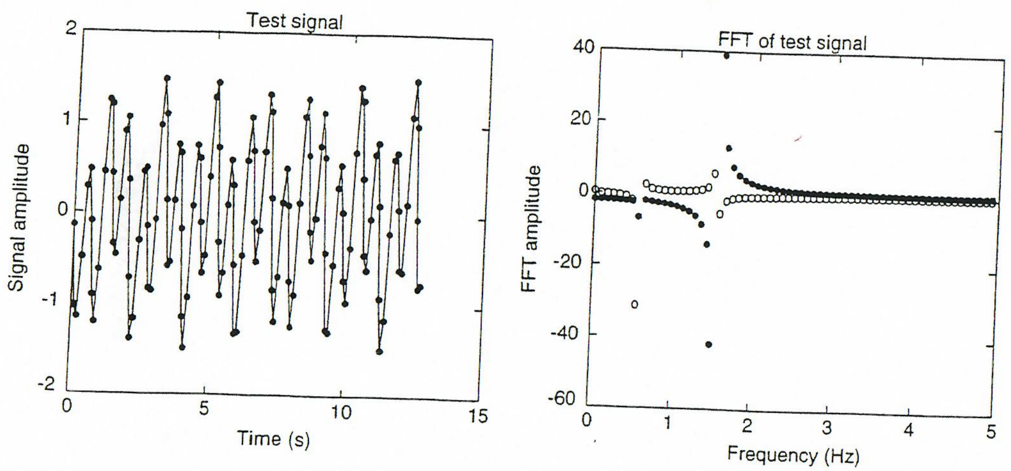

Figure A2.4: Left: test signal which is the sum of two sine waves with different frequencies, amplitudes, and phases; right: FFT of this test signal. The filled circles show the real (cosine) components, while the open circles are the imaginary (sine) amplitudes.

Next we consider essentially the same signal, but now we shift it along the time axis by adding a phase factor of $\pi/4$. That is, our signal is the function $y(t) = \sin(2\pi ft + \pi/4)$ shown in Figure A2.3. It is again sampled 128 times at intervals $\Delta t = 0.1$ s. The FFT is now slightly more complicated, with one nonzero Fourier sine component and one nonzero cosine component. These correspond to writing the signal as the sum of a sine and cosine, that is, $\sin(2\pi ft + \pi/4) = [\sin(2\pi ft) + \cos(2\pi ft)]/\sqrt{2}$. This is precisely what we find in the FFT result and shows why an FFT often yields nonzero sine and cosine components at each frequency. They are both needed if we are to be able to describe a signal with an arbitrary phase [recall the phase factors in (A2.1)].

The very simple FFT results we have observed in our first two examples are due, in part, to the fact that we have chosen the period of the signal to precisely match the total sampling time. That is, we have sampled three *complete* periods. If the sampling time does not match the frequencies of the Fourier components, the FFT has a slightly more complicated appearance. In Figure A2.4 we consider a hypothetical signal that consists of two sine waves of different frequencies, neither of which is commensurate with the sampling time. The FFT shows large components at two frequencies that match the frequencies of the sine waves used to construct the original signal. However, we also find Fourier amplitudes that are small but nonzero over a range of frequencies. This can be understood if we recall that the frequencies of the discrete transform are $f_j = j/N\Delta t$, where j runs from 0 to $N/2 - 1$. If a frequency contained in the signal does not coincide with one of these discrete frequencies, the FFT is forced to represent the signal as a sum of components over a range of f_j. However, it is important to note that such a representation will still give a *perfect* description of the original data values.

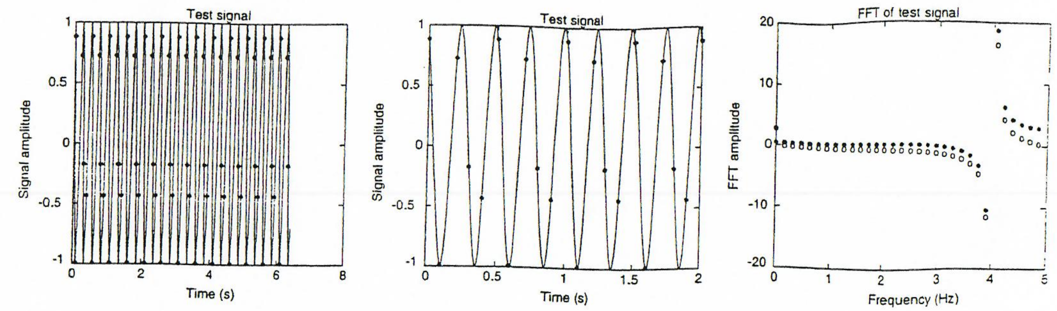

Figure A2.5: Left: test signal that is a single sine wave. The curve is what the signal would look like if it were sampled at a very large number of points, while the solid symbols show the 64 data points that were analyzed in the FFT. Center: an expanded view of the signal; right: FFT of this signal. The filled circles are the real (cosine) components, while the open circles are the imaginary (sine) components.

We have already mentioned the sampling theorem, which says essentially that the FFT will give us a perfect description of the Fourier components as long as the frequencies of these components are below the Nyquist frequency, $1/2\Delta t$. But what happens if this condition is not satisfied? In Figure A2.5 we show a pure sine wave signal that varies rapidly with time. The frequency here was 4 Hz, and the filled circles show the signal values used in the calculation. The FFT obtained using these 64 data points is also shown and exhibits a sizable component at 4 Hz, as expected. Note that the Nyquist frequency in this case was 5 Hz, so the sampling theorem says that we should, indeed, be able to successfully handle this signal.

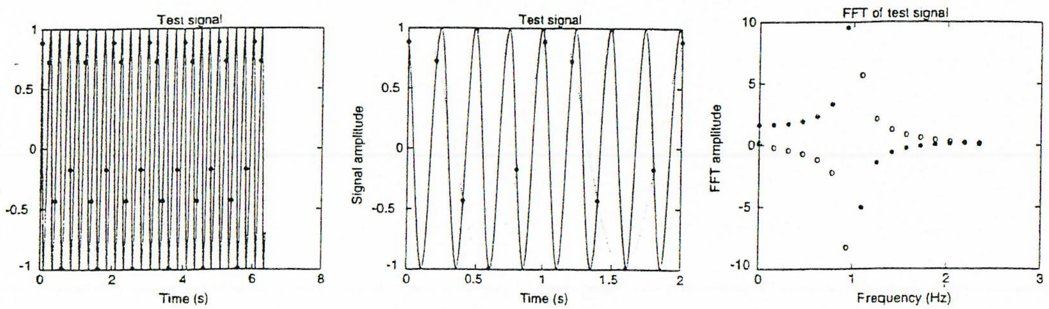

Figure A2.6: Left: test signal that is a single sine wave. The solid curve is what the signal would look like if it were sampled at a very large number of points, while the solid symbols show the 32 data points that were analyzed in the FFT. Center: an expanded view of the signal. The dotted curve shows a second sine wave, which has a much lower frequency (1 Hz) and which is seen to also pass through all of the data points. Right: FFT. The filled circles are the real (cosine) components, while the open circles are the imaginary (sine) components.

Figure A2.6 shows what happens when the signal is sampled with a different value of Δt. Here we used $\Delta t = 0.2$ s (N is now 32, so there are half as many Fourier components), which gives a Nyquist frequency of 2.5 Hz. This is lower than our signal frequency so we expect trouble, and we do indeed find it. The FFT now exhibits a peak at 1 Hz, far from the signal frequency of 4 Hz.

The location of this peak comes about in the following way. The frequency of the sine wave was greater than the Nyquist frequency, which means that there were fewer than two sampled points per period. In this situation the sampled points, y_k, can be described with equal precision by either of *two* different sine waves, one at the "true" frequency and one at a frequency that is *below* the Nyquist frequency. Here the "true" frequency was 4 Hz. The frequency of the other sine wave, which passes through these points, can be calculated by "reflecting" the true frequency about f_{Nyquist}. In this problem $f_{\text{true}} - f_{\text{Nyquist}} = 4 - 2.5 = 1.5$ Hz. The reflected frequency is then 1.5 Hz below the Nyquist frequency, that is, at 1 Hz, and this is where we find a peak in the FFT.

We must be careful here when referring to the "true" frequency. When samples are recorded only at intervals of $\Delta t = 0.2$ s, these two sine waves (one at 4 and one at 1 Hz) yield precisely the same signal. This is illustrated in the center part of Figure A2.6, which shows that a sine wave with a frequency of 1 Hz passes precisely through all of the data points. Thus, if all we have are these data points, who is to say which is the true frequency? This folding back of frequencies above the Nyquist frequency is known as *aliasing*.

In practice it is preferable to arrange for the Nyquist frequency to be higher than any of the Fourier components that are expected to be present in the signal. In an experiment this can be accomplished by using a low pass filter to remove components with frequencies above the Nyquist frequency, so there is no possibility of aliasing. In numerical work this is generally not a problem, since the sampling interval is usually the time step of a simulation, which should always be small compared to the characteristic time scales of the problem. This is equivalent to saying that all of the Fourier components lie below the Nyquist frequency.

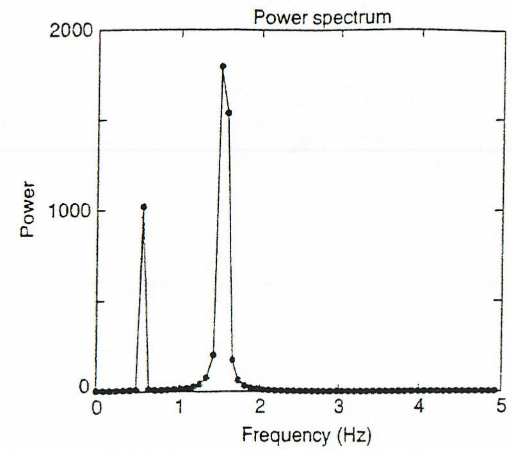

Figure A2.7: Power spectrum of signal in Figure A2.4. The solid symbols are the calculated values of the power.

A2.3 Estimation of the Power Spectrum

To this point we have always displayed the real (cosine) and imaginary (sine) parts of the FFT separately. Such a presentation has the advantage that it contains *all* of the information in the original signal. This is essential if we want to later use a backward transform to return to the time domain. However, we are often interested only in the frequencies and relative amplitudes of the Fourier components, but don't really care about their phases. In such cases there is another useful way to display the results of an FFT, known as the power spectrum. This name is used for the following reason. Suppose that $y(t)$ is an electrical signal, such as the voltage as a function of time across a resistor. The power dissipated in the resistor at a frequency f_j is proportional to the sum of the squares of the amplitudes of the cosine (real) and sine (imaginary) components at f_j

$$P_j = Y_j(\text{real})^2 + Y_j(\text{imaginary})^2 . \tag{A2.5}$$

FFT results for Y_j(real) and Y_j(imaginary) can thus be used to compute the power at each frequency, f_j; this is commonly referred to as a power spectrum. As an example, Figure A2.7 shows results for the power spectrum of the signal in Figure A2.4. Here there are just two peaks at the appropriate frequencies, and their relative sizes are proportional to the squares of the corresponding Fourier amplitudes.[6] Note that in computing the power (A2.5) we discard the phase information, since the relative magnitudes of the sine and cosine components cannot be determined solely from P. It is thus not possible to recover the original signal from knowledge of the power spectrum alone.

[6] To be precise we should really compare the areas under each peak in the power spectrum.