# Chapter 13. Fourier and Spectral Applications

## 13.0 Introduction

Fourier methods have revolutionized fields of science and engineering, from radio astronomy to medical imaging, from seismology to spectroscopy. In this chapter, we present some of the basic applications of Fourier and spectral methods that have made these revolutions possible.

Say the word "Fourier" to a numericist, and the response, as if by Pavlovian conditioning, will likely be "FFT." Indeed, the wide application of Fourier methods must be credited principally to the existence of the fast Fourier transform. Better mousetraps stand aside: If you speed up *any* nontrivial algorithm by a factor of a million or so, the world will beat a path towards finding useful applications for it. The most direct applications of the FFT are to the convolution or deconvolution of data (§13.1), correlation and autocorrelation (§13.2), optimal filtering (§13.3), power spectrum estimation (§13.4), and the computation of Fourier integrals (§13.9).

As important as they are, however, FFT methods are not the be-all and end-all of spectral analysis. Section 13.5 is a brief introduction to the field of time-domain digital filters. In the spectral domain, one limitation of the FFT is that it always represents a function's Fourier transform as a polynomial in $z = \exp(2\pi i f \Delta)$ (cf. equation 12.1.7). Sometimes, processes have spectra whose shapes are not well represented by this form. An alternative form, which allows the spectrum to have poles in $z$, is used in the techniques of linear prediction (§13.6) and maximum entropy spectral estimation (§13.7).

Another significant limitation of all FFT methods is that they require the input data to be sampled at evenly spaced intervals. For irregularly or incompletely sampled data, other (albeit slower) methods are available, as discussed in §13.8.

So-called wavelet methods inhabit a representation of function space that is neither in the temporal, nor in the spectral, domain, but rather something in-between. Section 13.10 is an introduction to this subject. Finally §13.11 is an excursion into numerical use of the Fourier sampling theorem.

## 13.1 Convolution and Deconvolution Using the FFT

We have defined the *convolution* of two functions for the continuous case in equation (12.0.8), and have given the *convolution theorem* as equation (12.0.9). The theorem says that the Fourier transform of the convolution of two functions is equal to the product of their individual Fourier transforms. Now, we want to deal with the discrete case. We will mention first the context in which convolution is a useful procedure, and then discuss how to compute it efficiently using the FFT.

The convolution of two functions $r(t)$ and $s(t)$, denoted $r * s$, is mathematically equal to their convolution in the opposite order, $s * r$. Nevertheless, in most applications the two functions have quite different meanings and characters. One of the functions, say $s$, is typically a signal or data stream, which goes on indefinitely in time (or in whatever the appropriate independent variable may be). The other function $r$ is a "response function," typically a peaked function that falls to zero in both directions from its maximum. The effect of convolution is to smear the signal $s(t)$ in time according to the recipe provided by the response function $r(t)$, as shown in Figure 13.1.1. In particular, a spike or delta-function of unit area in $s$ which occurs at some time $t_0$ is supposed to be smeared into the shape of the response function itself, but translated from time 0 to time $t_0$ as $r(t - t_0)$.

In the discrete case, the signal $s(t)$ is represented by its sampled values at equal time intervals $s_j$. The response function is also a discrete set of numbers $r_k$, with the following interpretation: $r_0$ tells what multiple of the input signal in one channel (one particular value of $j$) is copied into the identical output channel (same value of $j$); $r_1$ tells what multiple of input signal in channel $j$ is additionally copied into output channel $j + 1$; $r_{-1}$ tells the multiple that is copied into channel $j - 1$; and so on for both positive and negative values of $k$ in $r_k$. Figure 13.1.2 illustrates the situation.

Example: a response function with $r_0 = 1$ and all other $r_k$'s equal to zero is just the identity filter: convolution of a signal with this response function gives identically the signal. Another example is the response function with $r_{14} = 1.5$ and all other $r_k$'s equal to zero. This produces convolved output that is the input signal multiplied by $1.5$ and delayed by $14$ sample intervals.

Evidently, we have just described in words the following definition of discrete convolution with a response function of finite duration $M$:

$$(r * s)_j \equiv \sum_{k=-M/2+1}^{M/2} s_{j-k}\ r_k \tag{13.1.1}$$

If a discrete response function is nonzero only in some range $-M/2 < k \leq M/2$, where $M$ is a sufficiently large even integer, then the response function is called a *finite impulse response (FIR)*, and its *duration* is $M$. (Notice that we are defining $M$ as the number of nonzero *values* of $r_k$; these values span a time interval of $M - 1$ sampling times.) In most practical circumstances the case of finite $M$ is the case of interest, either because the response really has a finite duration, or because we choose to truncate it at some point and approximate it by a finite-duration response function.

The *discrete convolution theorem* is this: If a signal $s_j$ is *periodic* with period $N$, so that it is completely determined by the $N$ values $s_0, \ldots, s_{N-1}$, then its