3

Further Developments and Applications of the Finite Fourier Transform

In actual applications, most mathematical methods have to deal with finite data sets. Thus it is not surprising that the *finite* Fourier transform is the main tool among transforms in applied research. Two topics in communication science have been selected to illustrate the use of the finite Fourier transform: signal filters and windows in Section 3.1 and signal detection in the presence of noise in Section 3.2. These make use of the operations of convolution and correlation. The implementations of these techniques would be impossible without present-day computers and an efficient algorithm for the numerical work. The fast Fourier transform (FFT) operating principles are given in Section 3.3. Finally, in Section 3.4 we let the dimension of the vector space grow without bound. In this way we arrive at the Fourier series and integral transforms which are the subjects of Parts II and III. The sections are mutually independent except for Section 3.2, which relies somewhat on concepts developed in Section 3.1. Otherwise, they can be read in any order. The References should be consulted if the reader wishes a wider picture of the applied technology.

3.1. Convolution: Filters and Windows

The operation of convolution between the components of two vectors in $\mathscr{V}^N$ does not commonly appear in ordinary vector analysis but is quite important in the applications of the Fourier transform to communication

theory and technology. We shall first introduce this operation in a rather general setting and then particularize to the case of interest as applied to signal filtering and windowing.

3.1.1. The Coordinate-by-Coordinate Product Relative to a Basis

Let **f** and **g** be two vectors in $\mathscr{V}^N$, with coordinates $\{f_n\}_{n=1}^N$ and $\{g_n\}_{n=1}^N$ in the $\boldsymbol{\varepsilon}$-basis (see Sections 1.1 and 1.2). Construct now a vector $\mathbf{h} \in \mathscr{V}^N$ whose coordinates in the same basis are

$$h_n = f_n g_n, \qquad n = 1, 2, \ldots, N, \tag{3.1}$$

i.e., simply the coordinate-by-coordinate product of the first two vectors. We denote thus

$$\mathbf{h} \coloneqq \mathbf{f}(\boldsymbol{\varepsilon})\,\mathbf{g}, \tag{3.2}$$

defining a mapping from $\mathscr{V}^N \times \mathscr{V}^N$ into $\mathscr{V}^N$ which we call the *product* of vectors **f** and **g** *relative* to the $\boldsymbol{\varepsilon}$-basis. To determine the coordinates $\{\bar{h}_m\}_{m=1}^N$ of **h** in (3.2) in any *other* $\bar{\boldsymbol{\varepsilon}}$-basis obtained from the first one through a transformation **V** (see Section 1.2), we perform

$$\begin{aligned}
\bar{h}_m &= \sum_n (V^{-1})_{mn} h_n = \sum_n V^{-1}_{mn} f_n g_n \\
&= \sum_n V^{-1}_{mn} \sum_k V_{nk} \bar{f}_k \sum_l V_{nl} \bar{g}_l \\
&= \sum_{k,l} C^{(V)}_{m,k,l} \bar{f}_k \bar{g}_l,
\end{aligned} \tag{3.3}$$

where

$$C^{(V)}_{m,k,l} \coloneqq \sum_n V^{-1}_{mn} V_{nk} V_{nl} \tag{3.4}$$

are the *coupling* coefficients for the $\bar{\boldsymbol{\varepsilon}}$-basis coordinates.

The definition of the product (3.1)–(3.2) is quite simple. It does not appear in ordinary three-dimensional vector analysis since it does not seem to have found any meaningful application. In Fourier analysis, we shall see that it is quite useful.

Exercise 3.1. Show that the *bilinear* product (3.1)–(3.2) is commutative, associative, and distributive with respect to vector addition. Perform the proof in the $\boldsymbol{\varepsilon}$- and $\bar{\boldsymbol{\varepsilon}}$-bases. What symmetries are implied for the coupling coefficients (3.4)?

3.1.2. Coupling Coefficients and Convolution

If the transformation **V** in (3.3)–(3.4) is the Fourier transform, the coupling coefficients are particularly simple:

$$\begin{aligned} C^{(F)}_{m,k,l} &= \sum_n F^*_{nm} F_{nk} F_{nl} \\ &= N^{-3/2} \sum_n \exp[2\pi i n(m - k - l)/N] \\ &= N^{-1/2} \delta_{m,k+l}. \end{aligned} \tag{3.5}$$

The expression for $\tilde{h}_m$ is then called the *convolution* of $\tilde{f}_n$ and $\tilde{g}_n$:

$$\tilde{h}_m = N^{-1/2} \sum_n \tilde{f}_n \tilde{g}_{m-n} = N^{-1/2} \sum_n \tilde{f}_{m-n} \tilde{g}_n =: N^{-1/2} (\tilde{f} * \tilde{g})_m, \tag{3.6}$$

where all indices are counted modulo N.

3.1.3. Product in the Fourier Basis

If the product (3.2) is now made relative to the φ-basis,

$$\mathbf{k} = \mathbf{f}(\varphi)\,\mathbf{g}, \tag{3.7a}$$

namely,

$$\tilde{k}_m = \tilde{f}_m \tilde{g}_m, \qquad m = 1, 2, \ldots, N, \tag{3.7b}$$

then the ε-basis coordinates of the vectors involved can be found using the coupling coefficients (3.4) for the inverse Fourier transform. These are only the complex conjugates of (3.5), so that

$$k_n = N^{-1/2} \sum_m f_m g_{n-m} = N^{-1/2} \sum_m f_{n-m} g_m =: N^{-1/2} (f * g)_n. \tag{3.8}$$

These formulas have been collected in Table 1.1 at the end of Chapter 1.

Exercise 3.2. Using the Schwartz inequality, show that for (3.7)–(3.8)

$$|k_n|^2 \leqslant N^{-1} \|\mathbf{f}\|^2 \|\mathbf{g}\|^2. \tag{3.9}$$

Note for the product (3.1)–(3.2) relative to any basis $\boldsymbol{\delta}$ this implies that

$$\|\mathbf{f}(\boldsymbol{\delta})\mathbf{g}\| \leqslant N^{1/2} \|\mathbf{f}\|\, \|\mathbf{g}\|. \tag{3.10}$$

3.1.4. Signals

In discussing applications in signal filtering we shall first consider the product (3.7) of two vectors relative to the φ-basis and define what we mean

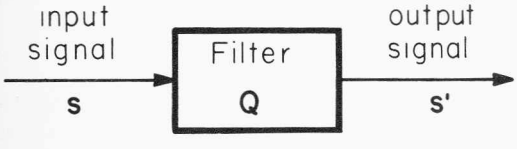


Fig. 3.1. Signal filtering.

here by a *signal vector* **s** and a *filter* $\mathbb{Q}$, showing then that the convolution (3.7)–(3.8) describes the output of the signal through the filter (see Fig. 3.1).

A *signal* **s** is an N-dimensional vector whose coordinates in the $\boldsymbol{\varepsilon}$-basis represent the input data to a "black box" system. This can be a telephone conversation, a space probe coded message, or any other form of information

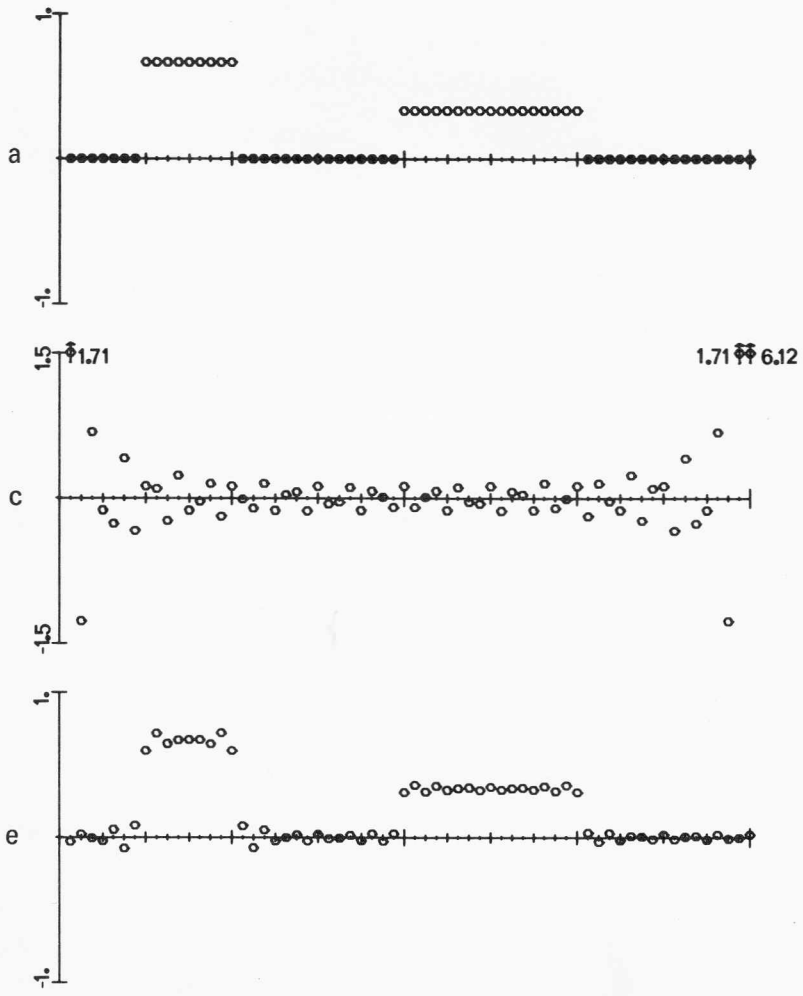
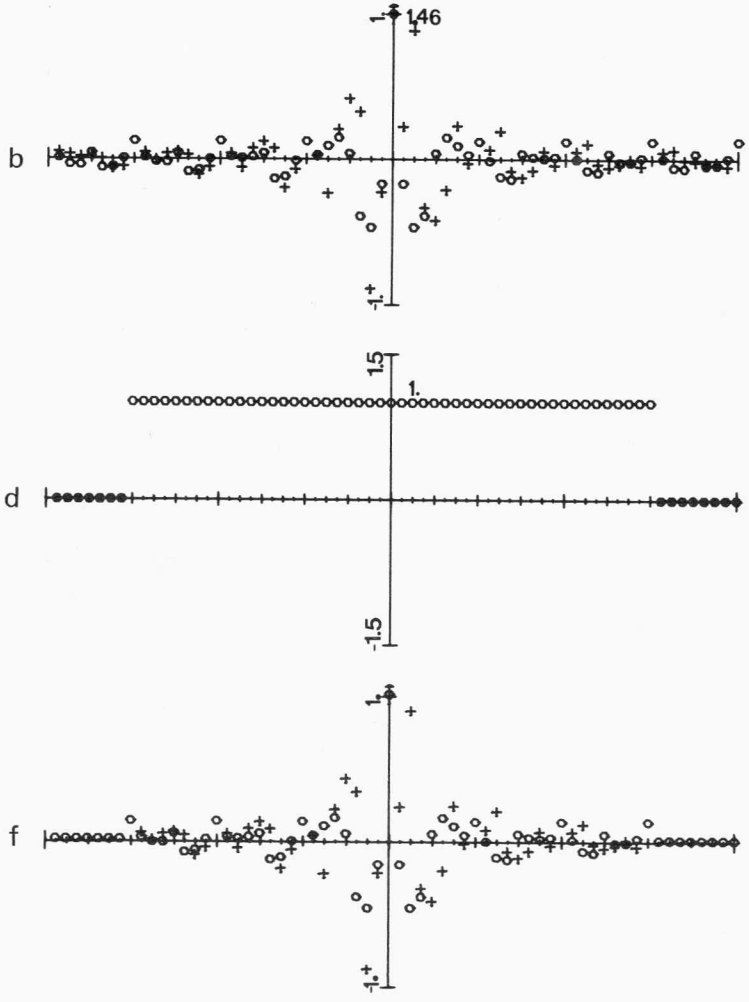


Fig. 3.2. (a) Signal. (b) Fourier transform of the signal. Real components are indicated by open circles, while imaginary components are denoted by crosses. As is customary, we are representing the Fourier-transformed components—the *frequency domain*—as extending on both sides of the $m = 0 \equiv N$ component. (c) Filter. (d) Transfer function of the filter [Fourier transform of (c)]. This is a low-pass filter which annuls the high-frequency components. Its transfer

which has finite length and which from the point of view of experiment can be taken to consist of a finite—albeit large—number of discrete data values. The consideration of discrete rather than continuous signals is here motivated by our mathematical construct but in practice corresponds to the impossibility of experimentally handling an actual infinity of data points. In Fig.



function is real and symmetric under reflections $m \leftrightarrow -m$; correspondingly, (c) exhibits the same characteristics. The product of (b) and (d) is (f), whose inverse Fourier transform is (e), the output filtered signal; (e) is thus the convolution of (a) and (c). Note that the suppression of the high-frequency components of the signal results in oscillations of the output in the neighborhood of its "discontinuities."

3.2(a) we show an example of a signal **s** with coordinates s_n, $n = 1, 2, \ldots, N$. In Fig. 3.2(b) the partial-wave content of **s** is shown: Equation (1.51b) states that

$$s_n = N^{-1/2} \sum_m \tilde{s}_m \exp(-2\pi i m n/N), \tag{3.11}$$

which displays the signal **s** as a sum of waveforms $\boldsymbol{\varphi}_m$ with amplitude proportional to $\tilde{s}_m$ (see Fig. 1.3). The quantities $p_m{}^s := |\tilde{s}_m|^2$ for $m = 1, 2, \ldots, N$ constitute the *power spectrum* of the signal **s**.

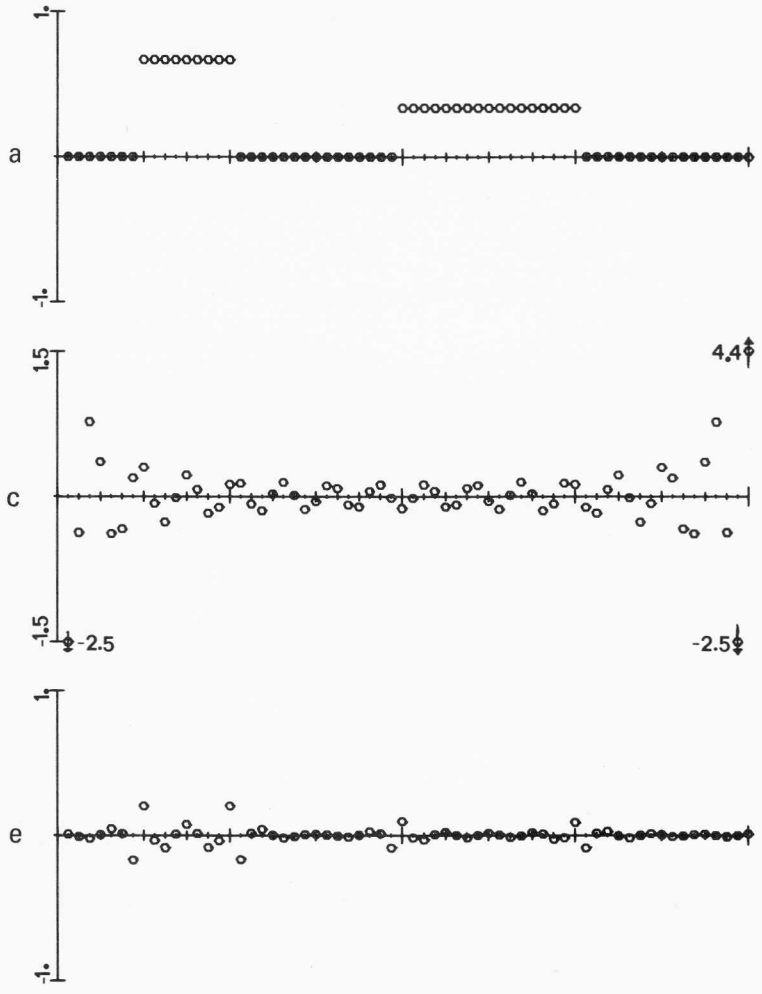
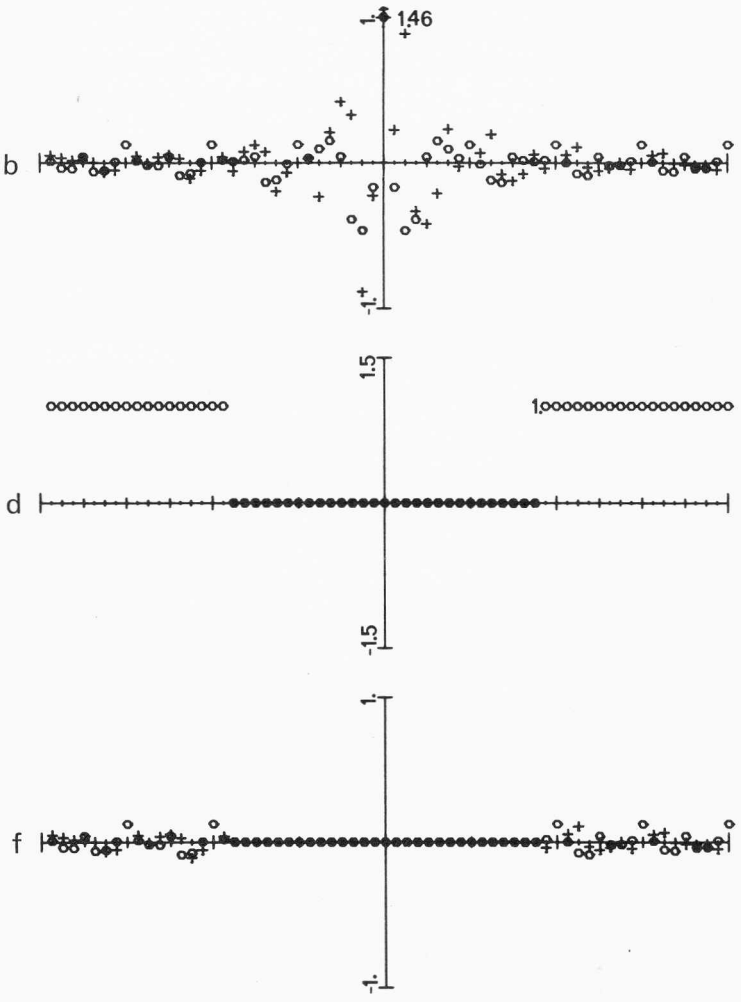


Fig. 3.3. (a) Signal (the same as in Fig. 3.2). (b) Fourier transform. (c) is the high-pass filter and (d) is its Fourier transform, i.e., the transfer function of the filter.

3.1.5. Filters

When the input signal $\mathbf{s}$ is fed into the "black box" signal processor in Fig. 3.1 it is converted into an *output* signal $\mathbf{s}'$. If a linear combination of input signals $\mathbf{s} = c_1\mathbf{s}_1 + c_2\mathbf{s}_2$, $c_1, c_2 \in \mathscr{C}$, is converted into the linear combination of the corresponding output signals $\mathbf{s}' = c_1\mathbf{s}'_1 + c_2\mathbf{s}'_2$, the box acts as a *linear operator* $\mathbb{Q}$ and $\mathbf{s}' = \mathbb{Q}\mathbf{s}$. To find the matrix $\|Q_{nm}\|$ representing $\mathbb{Q}$ in a given basis, we can test the box with *unit pulses*: We let $\mathbf{s} = \boldsymbol{\varepsilon}_n$ for



The product of (b) and (d) is (f). The output filtered signal is (e). The latter shows that under high-pass filtering it is mainly the "discontinuities" of the signal which remain.

$n = 1, 2, \ldots, N$ successively and then find the components $s'_m = (\varepsilon_m, \mathbf{s}')$ of the output signal, thus constructing $Q_{nm} = s'_m$ for $n, m = 1, 2, \ldots, N$. The testing can also be done using the waveforms $N^{-1/2} \exp(-2\pi inm/N)$, which constitute the signal $\mathbf{s}$ in (3.11). In this case, we let $\mathbf{s} = \boldsymbol{\varphi}_n$ for $n = 1, 2, \ldots, N$, find the components $\tilde{s}'_m = (\boldsymbol{\varphi}_m, \mathbf{s}')$, and construct $\mathbb{Q}$ as represented by $\tilde{Q}_{nm} = \tilde{s}'_m$ for $n, m = 1, 2, \ldots, N$. Now, if the black box is such that *waveforms of a given frequency are converted into waveforms of the same frequency*, with only a possible change of amplitude and phase, the device will here be called a *filter*. (In actual technology, the meaning of a *filter* is very often widened to include any linear operator.) In this case, for a given $\mathbf{s} = \boldsymbol{\varphi}_n$ input, we obtain an output $\mathbf{s}' = \tilde{q}_n \boldsymbol{\varphi}_n$, $\tilde{q}_n$ being a complex number, $n = 1, 2, \ldots, N$. The set of coefficients $\{\tilde{q}_n\}_{n=1}^N$ is called the *transfer function* of the filter. It is easy to see that $\mathbb{Q}$ is then represented in the $\boldsymbol{\varphi}$-basis by a *diagonal* matrix $\tilde{\mathbf{Q}} = \|\delta_{nm}\tilde{q}_n\|$, and any input signal (3.11) will produce an output $\mathbf{s}'$ with partial-wave coefficients

$$\tilde{s}'_n = \tilde{q}_n \tilde{s}_n, \qquad n = 1, 2, \ldots, N; \qquad \text{i.e.,} \quad \mathbf{s}' = \mathbf{q}(\boldsymbol{\varphi})\mathbf{s}. \tag{3.12}$$

For a particular wave input $\mathbf{s} = \boldsymbol{\varphi}_n$, when $\tilde{q}_n = 1$, the wave passes through the filter undistorted, while if $|\tilde{q}_n| > 1$ or $|\tilde{q}_n| < 1$, the wave will be enhanced or attenuated.

Exercise 3.3. Show that if $\tilde{q}_n$, the transfer function of a filter, is complex, its phase $\arg \tilde{q}_n$ determines a phase shift in the signal waveform. This shift, in units of data point separation, is $\sigma_n = -(N/2\pi n) \arg \tilde{q}_n$. Devices such that $|\tilde{q}_n| = 1$ and $\sigma_n =$ constant (modulo N) are *delay* filters. Notice that as we are working here with the tools of finite-dimensional spaces, a delay filter would pass the last part of the input to the beginning of the output.

3.1.6. Low- and High-Pass Filters

If low frequencies are enhanced and high frequencies are attenuated, i.e., if $\tilde{q}_n$ is large for n near 0 (recall the coordinates are numbered modulo N and see Fig. 1.3) and small for n near $N/2$, we have a *low-pass* filter. If high frequencies are enhanced and low ones correspondingly suppressed, the filter is a *high-pass* one. In Fig. 3.2(d) we have drawn the transfer function of a "rectangular" low-pass filter and in Fig. 3.2(c) its inverse transform. The output signal partial-wave coefficients (3.12) are shown in Fig. 3.2(f) and the output signal in Fig. 3.2(e). The latter is the *convolution* of the input signal Fig. 3.2(a) and the transform [Fig. 3.2(c)] of the transfer function. In Fig. 3.3 a rectangular high-pass filter has been applied to the same signal. Note that the power spectrum of the output signal (3.12) is simply $p_m^{s'} = p_m{}^s |\tilde{q}_m|^2$. This is unchanged for delay filters (see Exercise 3.3).

We are generally interested in upgrading the quality of signals, not in degrading it as Figs. 3.2 and 3.3 may suggest. Transmission lines or storing

devices act in many ways as filters which attenuate the high-frequency components which constitute the "fine detail" of a signal. A high-pass filter which enhances these components can be used to restore the signal to its original sharpness. The proper transfer function of this upgrading filter is determined by determining the transfer function of the degrading process. Of course, if some frequencies are entirely suppressed, they cannot be restored; the effect of noise (to be described in Section 3.2) happens also to be most important in the high-frequency region, so practical considerations exist which curtail the possibilities of these devices.

Exercise 3.4. Assume that the input signal is passed through two (or more) filters with different transfer functions $\tilde{q}_n^{(1)}$ and $\tilde{q}_n^{(2)}$. These may be placed in *series* [Fig. 3.4(a)] or in *parallel* [Fig. 3.4(b)] with a signal-summing device. Show that

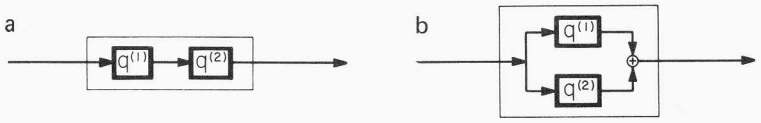


Fig. 3.4.

the filter arrays can be replaced by a single filter whose transfer function is $\tilde{q}_n = \tilde{q}_n^{(1)}\tilde{q}_n^{(2)}$ in the first case and $\tilde{q}_n = \tilde{q}_n^{(1)} + \tilde{q}_n^{(2)}$ in the second.

Exercise 3.5. An *averaging filter* produces an output signal $\mathbf{s}'$ whose components relate to the input signal $\mathbf{s}$ as $s'_m = (s_m + s_{m-1})/2$. Find the transfer function of such a filter to be $\tilde{q}_n = [1 + \exp(2\pi in/N)]/2$, and see that it enhances the lower frequencies. Note that, as an operator, the filter can be expressed by $\mathbb{Q} = (\mathbb{1} + \mathbb{R})/2$, where $\mathbb{R}$ is the rotation operator of Section 1.6. Such a filter will smooth out a signal and can be expected to reduce the noise (see Section 3.2).

Exercise 3.6. A *differencer filter* relates output to input by $s'_m = (s_m - s_{m-1})/2$. Find the transfer function, and see that it enhances the higher frequencies. A differencer filter will pick out *changes* in signal intensity and accentuate boundaries much like a Xerox copier when reproducing gray-tone images. Note that the second-difference operator $\mathbb{\Delta}$ of Sections 1.5 and 2.2 can be used as a filter too. Its spectrum tells us that it also enhances higher frequencies.

3.1.7. Windows

Our presentation of filtering devices has been overly optimistic. We have implied that the signal as a whole can be filtered when needed. A telephone conversation or even a speech spectrogram cannot be conveniently handled in this way. What must be done in these cases, roughly, is to break the full signal into consecutive pieces—time *windows*—each of which consists of a reasonably small set of data points which can be filtered and Fourier-analyzed separately. The process of windowing the signal corresponds mathematically

to multiplying the signal points s_n, $n = 1, 2, \ldots, N$, by a *window* function w_n, $n = 1, 2, \ldots, N$, which admits only data points between n_1 and n_2 and rejects all others: As a first example, we consider a *rectangular* window function:

$$r_n = \begin{cases} 1, & n_1 \leqslant n \leqslant n_2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

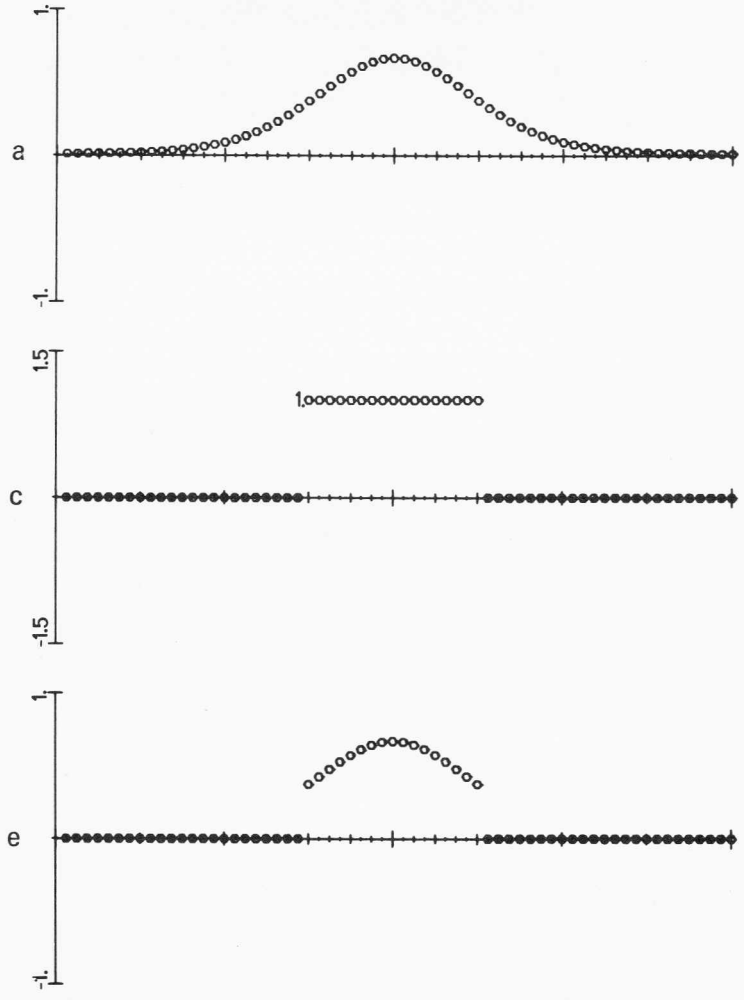
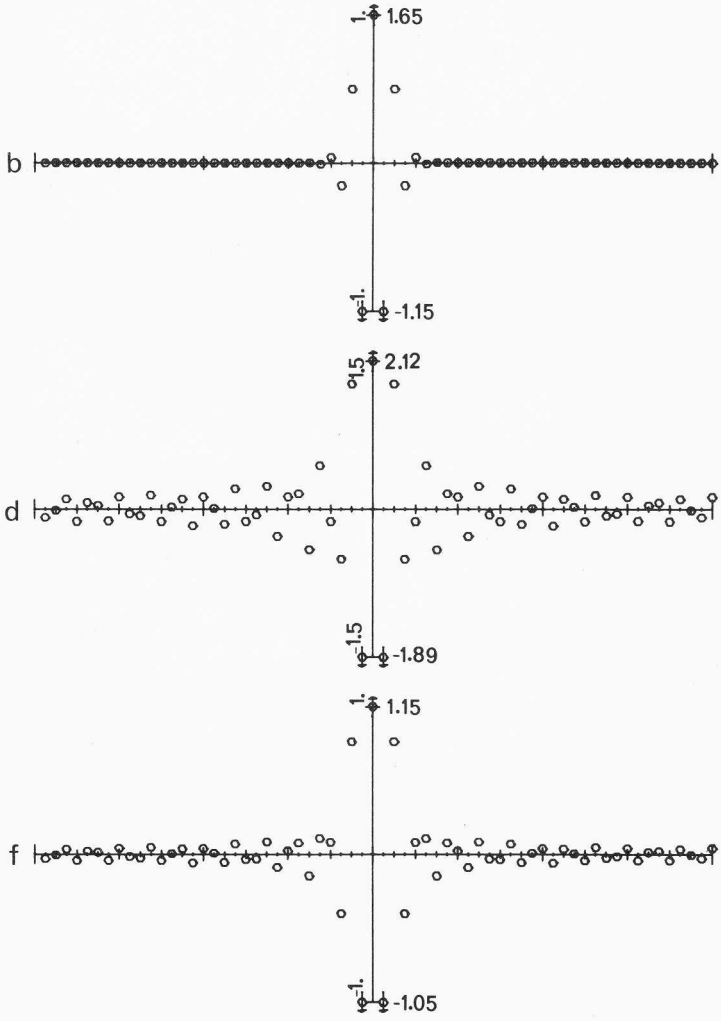


Fig. 3.5. (a) A "smooth" signal [representing the function in Eq. (2.38a)] and (b) its Fourier transform, exhibiting vanishingly small high-frequency components. (c) A rectangular "time window" and (d) its Fourier transform. The latter has

The output signal of such a windowing device is

$$s'_n = r_n s_n, \qquad \text{i.e.,} \quad \mathbf{s}' = \mathbf{r}(\varepsilon)\,\mathbf{s}. \tag{3.14}$$

In Fig. 3.5, we have used the rectangular window function [Fig. 3.5(c)] on a "smooth" signal [Fig. 3.5(a)] with little or no high-frequency components [Fig. 3.5(b)]. In chopping up a signal in this way [Fig. 3.5(e)], we are paying the price, due to the abruptness of the chop, of introducing spurious high-



significant components for all frequencies. (e) The chopped signal [the product of (a) and (c)]. (f) Fourier transform of (e) and convolution of (b) and (d). The appearance of high-frequency components is an artifact of the abrupt window function.

frequency components [Fig. 3.5(f)] which misrepresent the signal. This is an artifact of the window function we have used and can be seen to stem from the fact that the rectangular window function has a Fourier transform which is quite spread out in *side lobes*, with significant high-frequency components. The high-frequency components in Fig. 3.5(f) are a result of $\tilde{s}'_n$ being the convolution of $\tilde{s}_n$ with this spread-out window-transform function. This effect is termed *leakage*. To reduce the leakage effect it is desirable to use a window function whose Fourier transform has side lobes as small as possible.

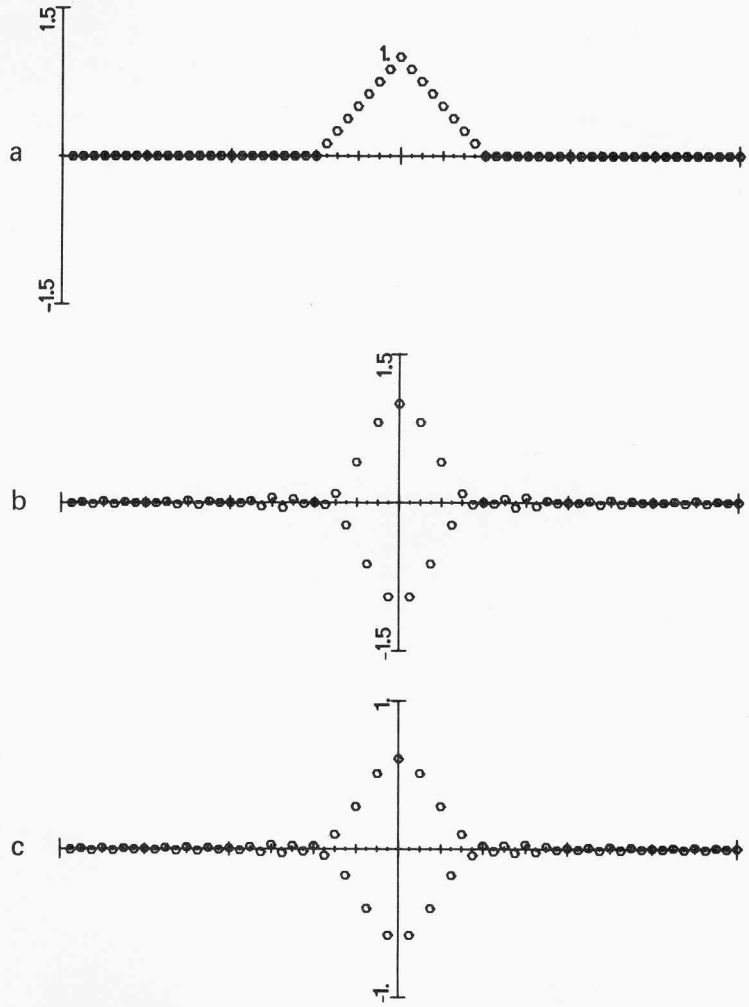


Fig. 3.6. (a) A triangular window function and (b) its Fourier transform. *Side lobes* are smaller here than in Fig. 3.5(d). (c) The smooth signal in Fig. 3.5(a) cut by this window exhibits smaller high-frequency components than in Fig. 3.5(f).

Instead of a rectangular window, a *triangular* function (Fig. 3.6) can be used, as its Fourier transform has smaller side lobes. An even better choice is the *Hanning* function, which has a $(1 - \cos \theta)$ form in its nonzero range (Fig. 3.7). The price paid for these improvements in the smoothing of the window is that there must be some window *overlap* in the description of the signal so that none of the signal components is slighted for falling at the edge of the window.

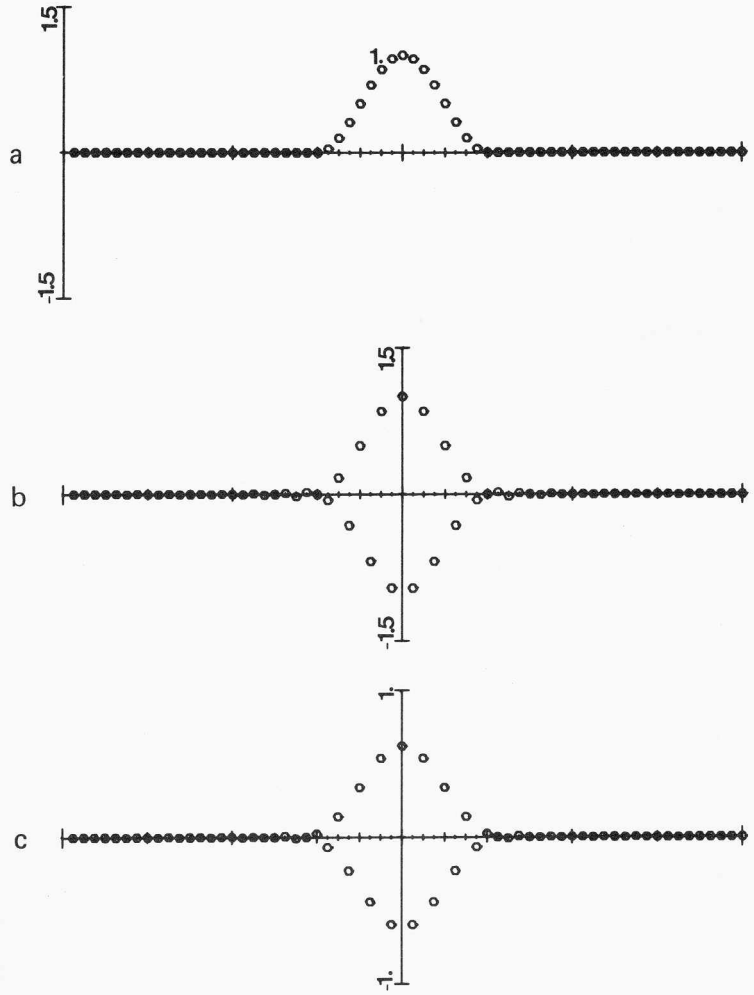


Fig. 3.7. (a) The Hanning function and (b) its Fourier transform. As the latter has negligible side lobes, the windowed "smooth" signal in Fig. 3.5(a), having basically no high-frequency components, (c), is expected to be acceptably "smooth" as well.

Exercise 3.7. Consider *amplitude modulation* of a *carrier wave* φ_c by a signal $\mathbf{s}^0$. The total input signal would then be $\mathbf{s}$,

$$s_n = (\varphi_c)_n s_n{}^0 = s_n{}^0 N^{-1/2} \exp(-2\pi i c n/N). \tag{3.15}$$

Show that the partial-wave coefficients of $\mathbf{s}$ are those of $\mathbf{s}^0$ but shifted by c units: $\tilde{s}_{m+c} \sim \tilde{s}_m{}^0$. Amplitude modulation can be used to transmit a very "smooth" signal, constituted only by low frequencies, through a communication line which strongly attenuates these frequencies. Shortwave AM radio, for instance, uses the transmission properties of electromagnetic waves of appropriately high frequency for the coding of low-frequency signals. FM, on the other hand, codes the signal into the Fourier transform components $\tilde{s}_n$ with proper time windowing.

We have tried to give an inkling of how the Fourier transform and convolution appear in communication. Clearly, to go into more details would take us to a very broad field. The reader interested in this area will definitely benefit from browsing through the books by Lee (1960) and Schwartz and Shaw (1975) and that of Jenkins and Watts (1968) on signal processing and applications of spectral analysis as well as the book by Brigham (1974) on basic Fourier transform applications, which also contains a good list of the source literature. A delightful field of application is that of speech analysis and synthesis. A very readable article by Flanagan (1972) and a book by Flanagan (1971) are suggested.

3.2. Correlation: Signal Detection and Noise

Signal detection in the presence of noise is one of the most important problems in communication. The concepts developed in Fourier analysis will be used to state some of the relevant variables and to broadly outline the strategy of solution. We start by defining the *correlation* of a string of signal data.

3.2.1. Correlation

Consider a sesquilinear operation mapping $\mathscr{V}^N \times \mathscr{V}^N$ into $\mathscr{V}^N$ relative to a basis—for definiteness we shall consider here the $\boldsymbol{\varphi}$-basis—as the component-by-component product

$$\tilde{k}_m = \tilde{f}_m^* \tilde{g}_m, \qquad m = 1, 2, \ldots, N, \quad \mathbf{f}, \mathbf{g}, \mathbf{k} \in \mathscr{V}^N, \tag{3.16}$$

Except for the complex conjugation in the first factor, this operation is basically the product introduced in Section 3.1, and its properties are quite similar. The distinct usefulness of (3.16) appears when we translate it to a

relation between the ε-basis coordinates of the vectors involved. These can be found through the inverse Fourier transform

$$k_n = \sum_m F_{nm}\tilde{k}_m = \sum_m F_{nm}\tilde{f}_m^*\tilde{g}_m$$
$$= \sum_m F_{nm}\sum_k F_{mk}f_k^*\sum_l F_{ml}^* g_l. \tag{3.17}$$

Exchanging sums and using (3.5) with an appropriate relabeling of indices, we find

$$k_n = N^{-1/2}\sum_m f_m^* g_{n+m} = N^{-1/2}\sum_m f_{m-n}^* g_m =: N^{-1/2}(fcg)_n,$$
$$n = 1, 2, \ldots, N, \tag{3.18}$$

which we define as the *correlation between* **f** and **g**.

Exercise 3.8. Show that in terms of the rotation operators $\mathbb{R}^n$ of Section 1.6 the correlation (3.18) can be written as

$$k_n = N^{-1/2}(fcg)_n = N^{-1/2}(\mathbf{f}, \mathbb{R}^n\mathbf{g}). \tag{3.19}$$

Exercise 3.9. Using the result of Exercise 3.8, the fact that $\mathbb{R}^n$ is a unitary operator and the Schwartz inequality show that the norm of the correlation vector in (3.18) satisfies

$$\|\mathbf{fcg}\| \leqslant N^{1/2}\|\mathbf{f}\|\,\|\mathbf{g}\|. \tag{3.20}$$

This is the analogue of a similar result on convolution given in (3.10).

The correlation assigns a set of numerical values to the "closeness" between the signal **f** and the signal **g**; if these are real and such that f_m and g_m have generally the same sign, $(fcg)_0$ will be a sum of generally positive terms and hence large. If we find some component l among the $(fcg)_n$ to be unusually large in comparison with the others, we can conclude either that f_m and g_{m-l} have generally the same sign or that a large component or components in **f** have met its or their counterpart in **g**. The number l gives the *lag* between the two.

3.2.2. Autocorrelation

Examine now the case when $\mathbf{f} = \mathbf{g}$, the *autocorrelation* function of **f** being the k_n in (3.18). Of course

$$k_0 = N^{-1/2}(fcf)_0 = N^{-1/2}\|\mathbf{f}\|^2, \tag{3.21}$$

but what happens to k_n for n "close" to 0? If the signal **f** is such that the f_m are a "smooth" or slowly varying function of m, $f_{m\pm1}$ will still have generally the same phase and magnitude as f_m, and so will $f_{m\pm2}$, etc. The correlation function k_n is thus expected to have a more or less broad real peak around

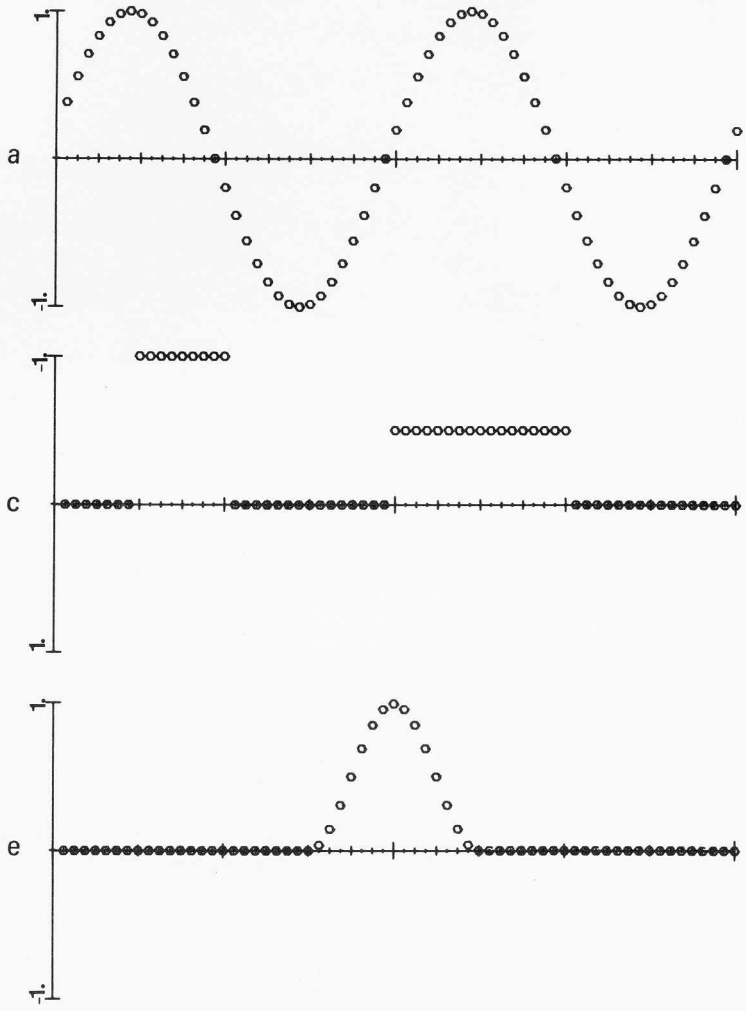


Fig. 3.8. (a) Periodic signal and (b) its autocorrelation. (c) "Two-peak" signal [Fig. 3.2(a)] and (d) its autocorrelation. (e) "Smooth" nonperiodic signal and (f) its autocorrelation.

$n = 0$. The width is determined by the distance j at which $f_{m\pm j}$ still has the same phase on the average as f_m, before sign cancellations start occurring in the sum (3.18). If now the signal f_m is *periodic* in m with period P (P divisor of N), then $f_m = f_{m+lP}$ for l integer and $k_{lP} = k_0$. The correlation function will exhibit peaks spaced by P units and will itself be periodic. In Fig. 3.8(a) is a periodic signal and in Fig. 3.8(b) its correlation function; other signals [Figs. 3.8(c) and (e)] also have characteristic correlation functions [Figs. 3.8(d) and (f)].

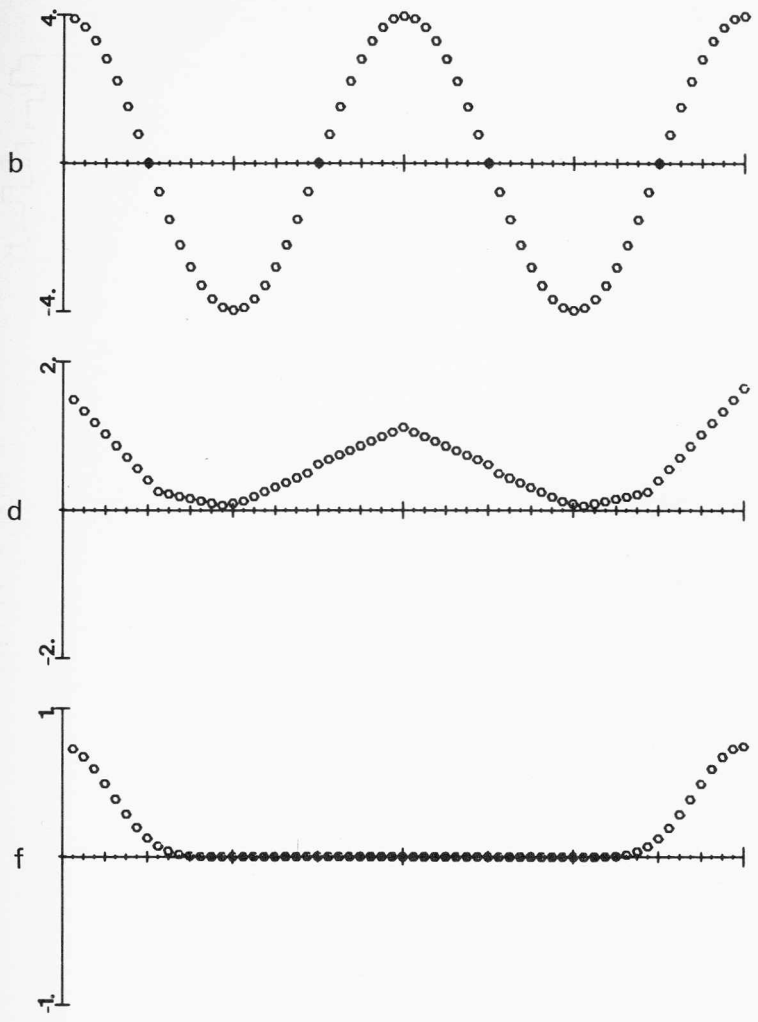


Fig. 3.8 (*continued*)

Exercise 3.10. Show that the autocorrelation function $k_n = (fcf)_n$ is even in n and cannot have a value *larger* than k_0 in (3.21). You can use the Schwartz inequality on (3.19).

Exercise 3.11. Show that *the Fourier transform of the autocorrelation function is the power spectrum* $p_n{}^f = |\tilde{f}_n|^2$ of **f**.

The autocorrelation function k_n, we have seen, gives a numerical value of the "degree of similarity" between a signal and its image shifted

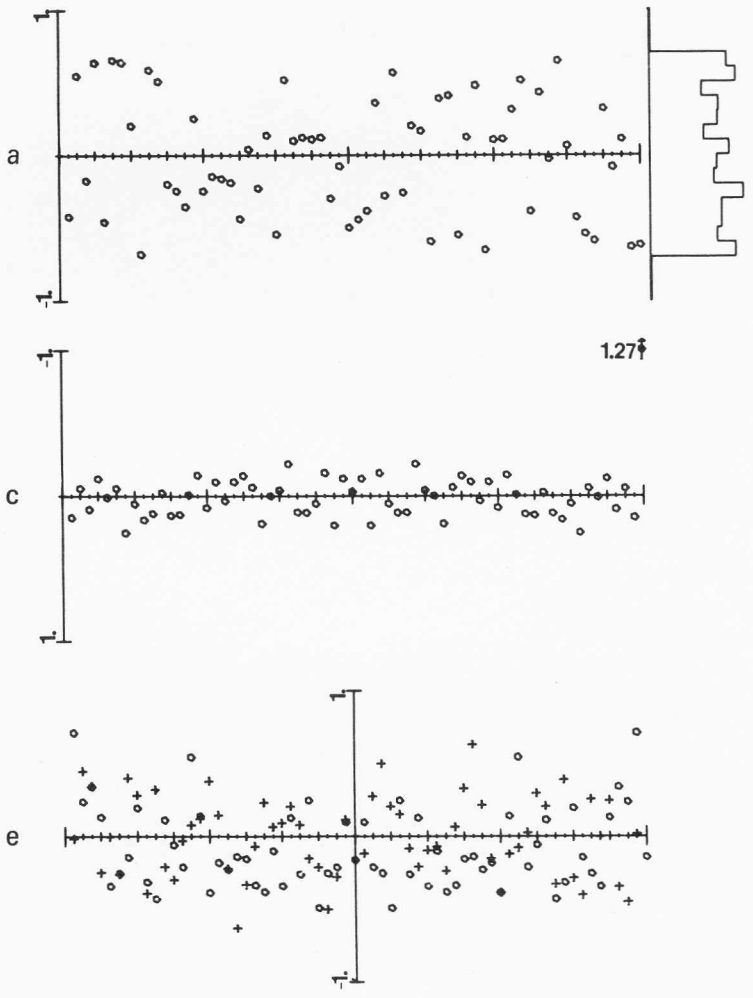
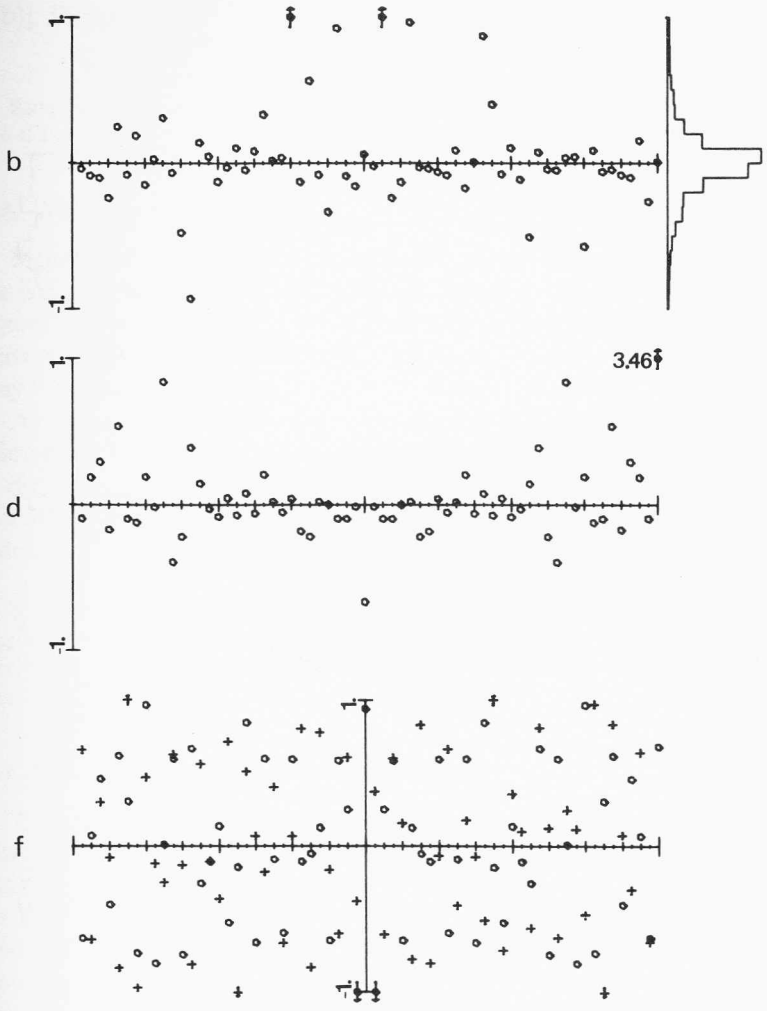


Fig. 3.9. (a) Constant-density noise and (b) Gaussian noise. The histograms to the right divide the ordinate range in 20 "bins," showing the characteristics of the distribution. (The latter were built on the basis of 1024 points rather than 64, as in the figures, in order to reduce random fluctuations.) (c) and (d) are

by n units. Suppose now we construct a "signal" $\boldsymbol{\nu}$ whose values $\boldsymbol{\nu}_m$, $m = 1, 2, \ldots, N$, are extracted from a random-number table or computer generator. Since no two values of the list are causally related, we can expect the autocorrelation function to have only a large k_0 value, but all other k_n's to fluctuate randomly. In Fig. 3.9(a) this is shown for a vector $\boldsymbol{\nu}$ constructed by a computer intrinsic "function" which produces a random sequence of real



the autocorrelations of (a) and (b), respectively. Notice the peak at the $N \equiv 0$ component of the correlation vector and the otherwise uneventful noise-like appearance of all other components. (e) and (f), Fourier transforms of (a) and (b), also have a noise-like character, showing comparable contributions from each frequency range.

numbers between -1 and 1 with constant *probability density*. This means that as the list of generated numbers tends to infinity, the proportion of those which fall in any interval $(v - \Delta v/2, v + \Delta v/2) \subset (-1, 1)$ is indepenent of the value of v. A histogram to the right of the figure shows this. The autocorrelation function in Fig. 3.9(c) is seen to exhibit only the peak at k_0. The same happens with random sequences with Gaussian probability densities [Fig.

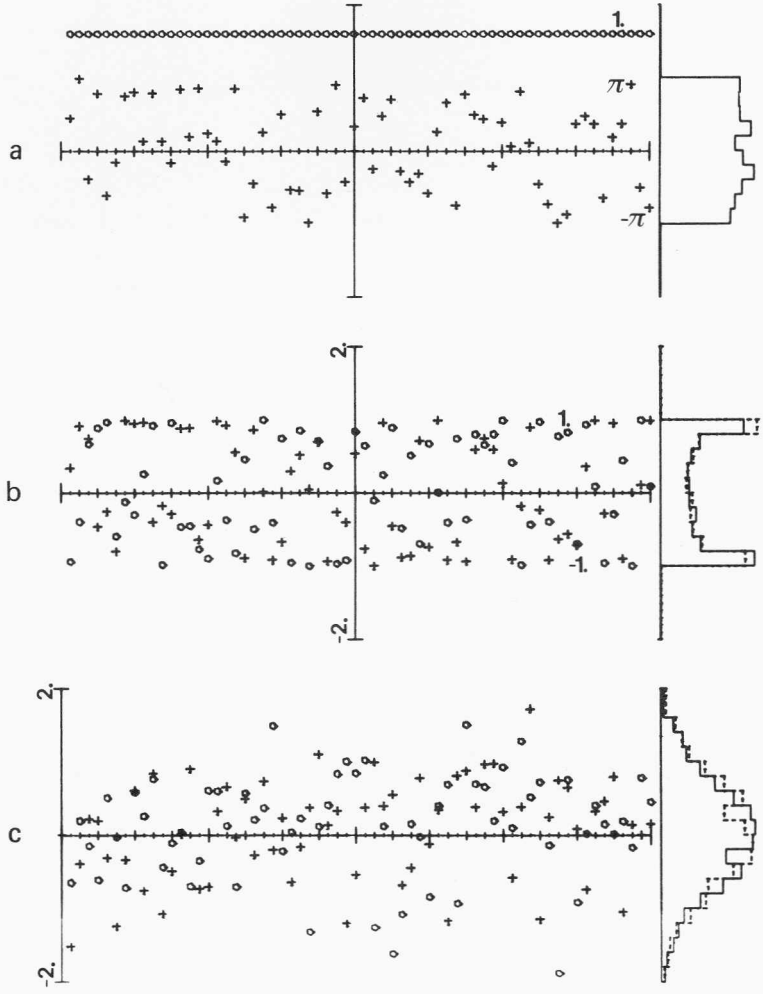


Fig. 3.10. White noise (a) built by requiring that all Fourier partial-wave coefficients have unit modulus (open circles) while their phases (crosses) be randomly distributed in $(-\pi, \pi)$ with constant density (histogram at the right). In terms of the real and imaginary parts of the Fourier coefficients (b), the probability density has a csc πx shape (see the histograms at the right—unbroken lines for the real parts and dotted lines for the imaginary parts). By inverse Fourier transformation, white noise (c) is obtained. As before, all histograms were built with 1024-component vectors, while the figures have only 64 points.

3.9(b)]. Figures 3.9(e) and (f) are the Fourier transforms of Figs. 3.9(a) and (b).

Exercise 3.12. Are there reasons to expect that as N grows without bound, $k_n \to 0$ for $n \neq 0$?

3.2.3. White Noise

Signals with random components are generically referred to as *noise*. This is a good working definition which describes the kind of background "signal" produced by the thermal agitation of electrons in radio or radar receivers and amplifiers. A broader "definition" of noise in communication is any "unwanted" part of the signal; of course this varies from case to case.

As Fig. 3.9 suggests, the definition of noise is not unique. For standardization purposes in filtering, it is common to define *white noise* as that which has *the same power spectrum at all frequencies*, i.e., such that $p_n^{\nu} = |\bar{\nu}_n|^2 =$ constant, so that only the *phase* of individual Fourier coefficients takes a random sequence of values. This is shown in Fig. 3.10.

Exercise 3.13. Note that a filter $\mathbb{Q}$ can change the characteristics of the noise input $\boldsymbol{\nu}$. Consider an averager and a differencer filter, and examine the correlation of the output. See that for these two cases $k_1^{\nu'} = \pm\frac{1}{4}k_0^{\nu} +$ random terms.

3.2.4. Signal Detection and Filtering of Noise

Noise is the part of the input signal $\mathbf{s}$ we usually want to get rid of. We consider $\mathbf{s} = \mathbf{s}_0 + \boldsymbol{\nu}$, $\mathbf{s}_0$ being the "true" signal and $\boldsymbol{\nu}$ the noise. In detecting signals $\mathbf{s}_0$ we should separate clearly two kinds of situations: first, when we have a fair idea of what $\mathbf{s}_0$ should be and we are interested in *detecting* the presence or absence of the signal, and second, when $\mathbf{s}_0$ is unknown and only its overall characteristics—as distinct from those of noise—can be used for *filtering* $\mathbf{s}$. The first situation corresponds, for instance, to radar technology, while the second was typical of early telephony.

The detection of known signals amid background noise is usually tackled by finding the correlation, in a time window, of the incoming signal. If $\mathbf{s}$ is a train of square pulses (Fig. 3.11), it has a correlation function which is quite distinct from that of $\boldsymbol{\nu}$ [Figs. 3.9(c) and (d)]. The correlation of $\mathbf{s} = \mathbf{s}_0 + \boldsymbol{\nu}$ is $\mathbf{s}_0\mathbf{c}\mathbf{s}_0 + \mathbf{s}_0\mathbf{c}\boldsymbol{\nu} + \boldsymbol{\nu}\mathbf{c}\mathbf{s}_0 + \boldsymbol{\nu}\mathbf{c}\boldsymbol{\nu}$. The shape of the first term, when present, can be recognized in Fig. 3.11. Moreover, the correlation can also be used to detect any *change* undergone by the signal. The return pulse of a radar bouncing off the surface of a planet, for instance, will yield the distance to the body by the travel-time lag; the Doppler shift due to the planet's radial velocity away or toward the observer will lengthen or shorten the pulses,

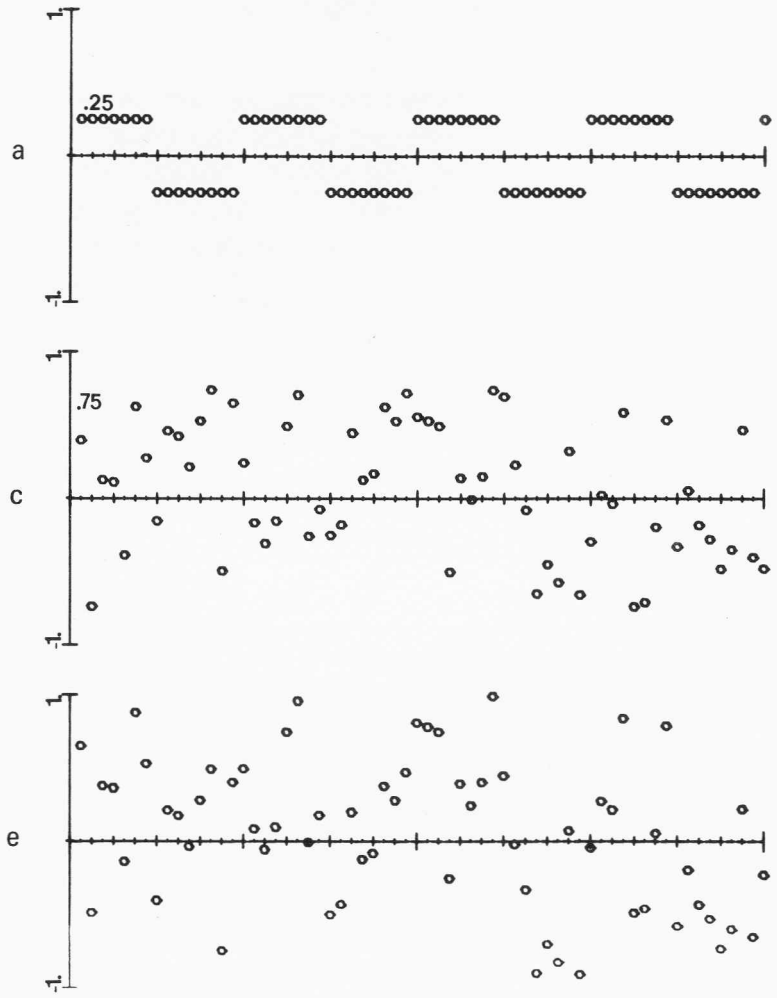
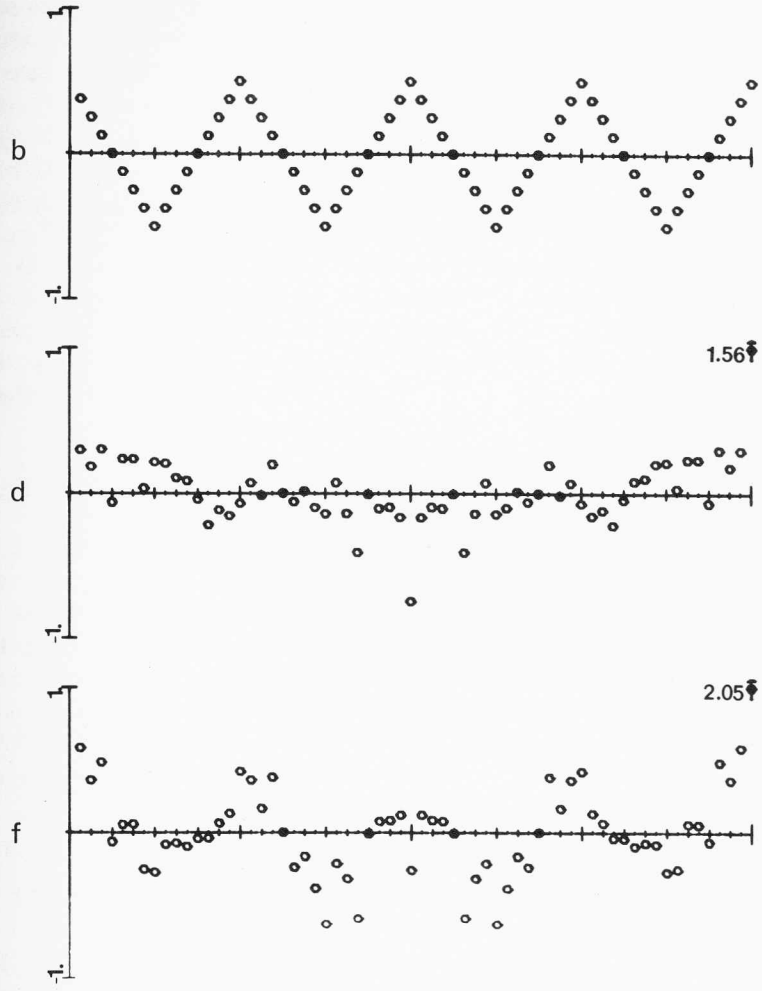


Fig. 3.11. Signal detection amid noise by correlation. (a) Periodic signal and (b) its correlation. (c) Constant-density noise and (d) its correlation. (e) Signal plus noise at a ratio of 1:3. The noise masks the signal, whose presence can

while the pulse shape will be changed by surface characteristics such as rugosity and ground reflexivity.

For the filtering of signals of which we have no *a priori* knowledge, the solution is not so clear-cut, and in fact the information of the "true" signal $\mathbf{s}_0$ is never fully retrievable. An *averager* filter (see Exercise 3.5) has a transfer function which attenuates the high-frequency components. If these are suppressed in the input (Fig. 3.12), the total noise power ($\sum_n |\tilde{v}_n|^2$) will be



nevertheless be detected by (f) correlation. The observed peaks and their periodicity match those of the signal, so we conclude that (e) contains a signal. The more data points we have, the more effective the detection by correlation becomes.

diminished to a greater extent than the total "true" signal power ($\sum_n |\delta_{0n}|^2$). The output is a "smoother" signal in which $\mathbf{s}_0$ should be recognizable. If the noise-to-signal ratio (total noise power/total "true" signal power) is large, this method—or any other filtering scheme—may not prevent loss of signal information.

A generally successful way to overcome the difficulties inherent in signal filtering is to *digitalize* the data to be transmitted, coding them into

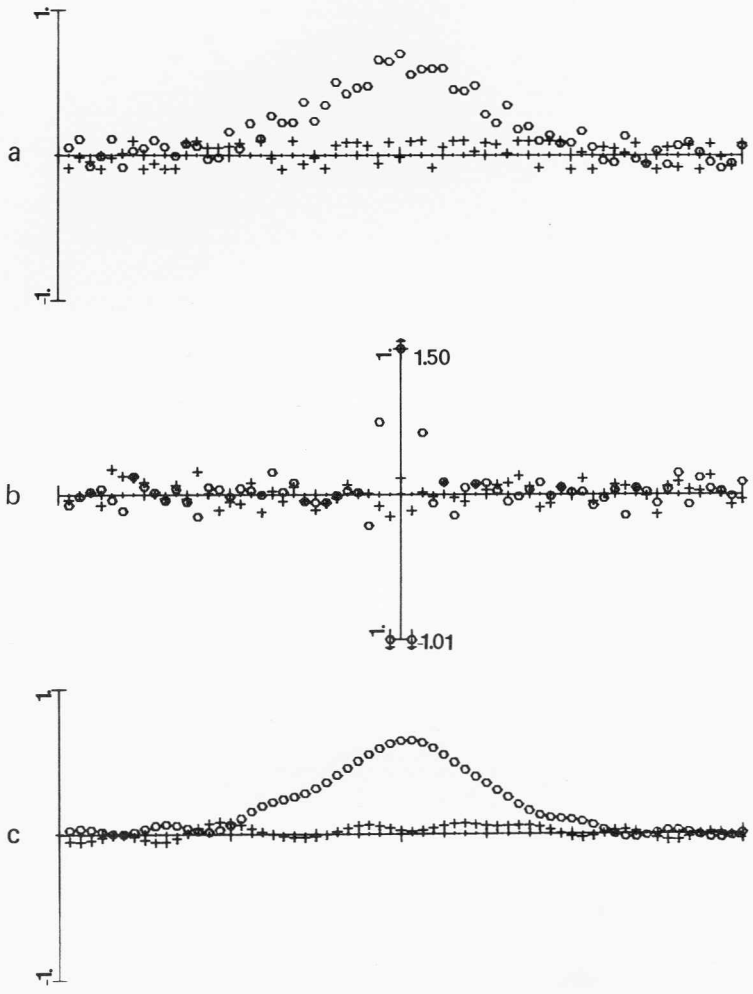


Fig. 3.12. Partial noise elimination by filtering. (a) A "smooth" signal [Fig. 3.5(a)] plus 25% white noise [Fig. 3.10(c)]. (b) The Fourier transform of the noisy signal contains large low-frequency components due to the signal [Fig. 3.5(b)] and essentially a constant high- and low-frequency noise background [Fig. 3.10(b)]. Filtering with a low-pass device whose transfer function is shown in Fig. 3.5(c) in the frequency domain, we obtain (c) the filtered signal. The small wavelets are the noise residue. Narrowing the filter's passing band would only distort the signal farther from its true shape. Broadening it would allow for more noise wavelets.

pulse sequences of "expected" shape [as in Fig. 3.11(a)] such that on arrival the message can be detected by correlation. On-the-spot planet photographs are scanned as by a TV image, but tones of gray are divided into, say, 32 values. The transmitted data will then consist of a string of numbers in this range, each in binary code whose digits, 0 or 1, are given by the absence or presence of a pulse. In this way, we trade the range of possible shades of gray (which is not too important, as 32 tones give a very accurate rendering of the picture) for protection against image degradation.

As in Section 3.1, the reader is urged to explore the source literature if he wishes to have more specialized information on the actual signal detection technology. See also the books by Papoulis (1965), Schwartz *et al.* (1966, 1970, and 1975), Gold and Rader (1969), Otnes and Enochson (1972), Stieglitz (1974), and Bloomfield (1976).

3.3. The Fast Fourier Transform Algorithm

Sections 3.1 and 3.2 point to the fact that the actual evaluation of the finite Fourier transform has a considerable range of application. Although the number of data points must in practice be finite, it can be very large, say on the order of 10^3 or 10^4, requiring a considerable amount of expensive computer time. An algorithm for the evaluation of the Fourier transform involving a drastic reduction in its computational complexity—by a factor of $N/\log_2 N$—was discovered recently by Cooley and Tukey [see Cooley and Tukey (1965); see also Cooley *et al.* (1967)].

3.3.1. Computational Complexity of the Longhand Fourier Transformation

Let us analyze the number of arithmetic operations required to calculate the Fourier transform $\{\tilde{f}_m\}_{m=1}^N$ from a given set of complex data points $\{f_n\}_{n=1}^N$. The "longhand" calculation proceeds by

$$\tilde{f}_m = N^{-1/2} \sum_{n=1}^{N} f_n \exp(2\pi i m n/N) \qquad \text{for } m = 1, 2, \ldots, N. \tag{3.22}$$

First, (a) one has to calculate $\exp(2\pi i/N)$ and then its $N - 1$ powers, as these will appear as factors in (3.22). Then (b) one has to perform the $(N-1)^2$ products of f_n's with these exponentials (for n or m equal to N the exponential factor is 1, so no product is necessary). Last, (c) there are $N(N-1)$ sums to be performed. The overall factor $N^{-1/2}$ need not be considered, as it is usually absorbed into a redefinition of the Fourier transform in actual applications.

Typical computer times required for the operations of real, single-precision, floating-point sum and product, including memory access, are on

the order of 25 μsec for the PDP-11/40, a medium-small computer. A medium-large computer such as the Burroughs 6700 requires around 7 μsec. If we round off the complexity of (3.22) as N^2 sums and N^2 multiplications, the computer work needed for N around 1000 is comprised of some 2 million complex operations. This represents some $6\frac{1}{2}$ min on the first and 2 min on the second computer. Even if machine time were unlimited and free, the Fourier transform would not often be used for real-time data analysis unless a considerably more efficient algorithm were found. The fast Fourier transform (FFT), for $N = 2^{10} = 1024$, leads to a 100-*fold* saving factor.

3.3.2. N Divisible by 2

Suppose that N is divisible by 2. The index n can be replaced by $2r + k - 1$ and the sum (3.22) split into

$$\tilde{f}_m = N^{-1/2} \sum_{k=0}^{1} \sum_{r=1}^{N/2} f_{2r+k-1} \exp[2\pi i(2r + k - 1)m/N]$$

$$= 2^{-1/2} \sum_{k=0}^{1} \exp[2\pi i(k - 1)m/N](N/2)^{-1/2} \sum_{r=1}^{N/2} f_{2r+k-1} \exp[2\pi irm/(N/2)]. \tag{3.23}$$

The second sum,

$$\tilde{f}^1_{k,m} \coloneqq (N/2)^{-1/2} \sum_{r=1}^{N/2} f_{2r+k-1} \exp(4\pi irm/N) = \tilde{f}^1_{k,N/2+m}, \tag{3.24}$$

is the $N/2$-dimensional Fourier transform, for $k = 0$, of the odd-numbered f_n's and of the even-numbered f_n's for $k = 1$. The determination of all the $\tilde{f}^1_{k,m}$'s in (3.24) involves $2(N/2)^2$ multiplications since we have two values of k and we need perform the Fourier transform only for $m = 1, 2, \ldots, N/2$. Once these have been calculated, we can *merge* the $\tilde{f}^1$'s as

$$\tilde{f}_m = 2^{-1/2}[\exp(-2\pi im/N)\tilde{f}^1_{0,m} + \tilde{f}^1_{1,m}], \qquad m = 1, 2, \ldots, N. \tag{3.25}$$

This process involves N products. The total number of multiplications in the algorithm (3.24)–(3.25) is thus $N^2/2 + N$ and about the same for sums. For large N this represents roughly a halving of the computation time.

3.3.3. N Divisible by 2^p

The reduction in computation complexity need not stop here: the $N/2$-dimensional Fourier transform (3.24) may be subject to the same process when $N/2$ is even. We need only replace r by $2s + k_2 - 1$, $s = 1, 2, \ldots, N/4$, defining a $\tilde{f}^2_{k_2k_1}$ as the $N/4$-dimensional transform of the f_n's with $n \equiv 0, 1, 2, 3 \bmod 4$ and the merging (3.25) between the $\tilde{f}^2$'s and $\tilde{f}^1$'s. The general

recursion for N divisible by 2^p involves, first, the $(2^{-p}N)$-dimensional Fourier transform [which we write without the constant $(2^{-p}N)^{-1/2}$, which should go in front],

$$\tilde{f}^p_{k_p\cdots k_2k_1,m} := \sum_{r=1}^{2^{-p}N} f_{n_p(r,k)} \exp(2^{p+1}\pi irm/N) = \tilde{f}^p_{k_p\cdots k_2k_1,2^{-p}N+m}, \tag{3.26a}$$

where

$$n_p(r, k) := 2^p r + 2^{p-1}(k_p - 1) + \cdots + 2(k_2 - 1) + (k_1 - 1). \tag{3.26b}$$

The following p steps are the mergings (again, eliminating the factor $2^{-1/2}$),

$$\tilde{f}^{q-1}_{k_{q-1}\cdots k_1,m} = [\exp(-2^q\pi im/N)\tilde{f}^q_{0k_{q-1}\cdots k_1,m} + \tilde{f}^q_{1k_{q-1}\cdots k_1,m}],$$
$$m = 1, 2, \ldots, 2^{-q+1}N, q = p, p - 1, \ldots, 1, \tag{3.27a}$$

where the last step is

$$\tilde{f}^0_m := N^{-1/2}\tilde{f}_m, \qquad m = 1, 2, \ldots, N. \tag{3.27b}$$

The number of multiplications in the algorithm (3.26)–(3.27) is $(2^{-p}N)^2$ for the Fourier transform (3.26) and N for each merging.

3.3.4. $N = 2^\nu$

The regression in the dimension of the Fourier transform ends when it reaches 1, since then we have no sums or multiplications at all. Thus consider N to be the νth power of 2, i.e., $N = 2^\nu$. Then, for $p = \nu$,

$$\tilde{f}^\nu_{k_\nu\cdots k_2k_1} = f_{n_\nu(1,k)}, \tag{3.28a}$$

as r and m can only take the value 1, and

$$n_\nu(1, k) = 2^{\nu+1} + 2^\nu(k_\nu - 1) + \cdots + 2(k_2 - 1) + (k_1 - 1)$$
$$= 2^\nu k_\nu + 2^{\nu-1}k_{\nu-1} + \cdots + 2k_2 + k_1 + 1. \tag{3.28b}$$

It is only left for us to perform the ν mergings (3.27) for $q = \nu, \nu - 1, \ldots, 1$. As each merging involves N products and N sums, the total number of operations of each type is νN or

$$N \log_2 N. \tag{3.29}$$

The computational complexity of the *fast Fourier transform* algorithm (3.27)–(3.28) is thus significantly smaller than that of the direct formula (3.22).

3.3.5. Regression and Binary Digit Inversion

In Fig. 3.13 we have displayed graphically the regression and merging for the fast Fourier transform for $N = 8 = 2^3$. We started in the leftmost

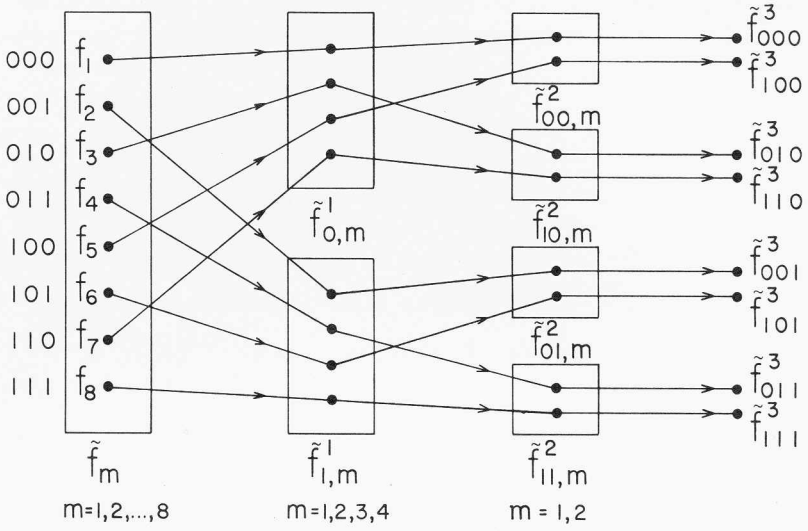


Fig. 3.13. Regression and merging of the fast Fourier transform algorithm.

column where all the $\tilde{f}_m$ depended on all the f_n. From there we proceeded to the second column having two parts, the four $\tilde{f}^1_{0,m}$'s depending on the even f_n's and the four $\tilde{f}^1_{1,m}$'s on the odd ones. From here we passed to the third column which has four pairs of $\tilde{f}^2_{k_2k_1,m}$'s, each depending on two f_n's. Last, we have eight $\tilde{f}^3_{k_3k_2k_1}$ which *are* f_n's. Note particularly that the *string of binary digits* $k_3k_2k_1$ *is the binary number representation of* $n-1$, which is written to the left of the first column. This is a general property which can be seen from (3.28b). It should also be noted that the overall shuffling of the entries in the first and last columns is such that it *inverts the digit order* of the binary representation of the row label. The merging procedure can be followed in Fig. 3.13 from right to left: each of the pair of $\tilde{f}^2_{k_2k_1,m}$'s, $m = 1, 2$, is constructed from the rightmost column entries to which it is connected, the upper link being multiplied by the phase in (3.27). Following suit, each of the $\tilde{f}^1_{k_1,m}$'s is obtained through the merging of the $\tilde{f}^2_{k_2k_1,m}$'s to which its block is linked and similarly for the $\tilde{f}_m$'s.

3.3.6. A Short Survey of the Literature

Figure 3.14 shows a FORTRAN program which calculates the direct and inverse Fourier transforms using the FFT algorithm. This program is not the ultimate in computation efficiency but should be easy to implement by the interested reader on his local computer. The software in most computing centers includes more than one version of the FFT. These are variants which follow either the *Cooley–Tukey* or the *Sande–Tukey* algorithms (Cooley and Tukey, 1965; Gentleman and Sande, 1966). Other fast algorithms

```
      SUBROUTINE FFT(X,N,NU,IT)
      COMPLEX X(N),E,T
      M=N/2
      NU1=NU-1
      K=0
      Z=IT
      U=1./SQRT(FLOAT(N))
      DO 3 I=1,NU
    1 DO 2 J=1,M
      A=INV(K/2**NU1,NU)*6.283185/N
      E=CMPLX(COS(A),Z*SIN(A))
      L=K+1
      LM=L+M
      T=X(LM)*E
      X(LM)=X(L)-T
      X(L)=X(L)+T
    2 K=K+1
      K=K+M
      IF(K.LT.N) GO TO 1
      K=0
      NU1=NU1-1
    3 M=M/2
      DO 4 K=1,N
      J=INV(K-1,NU)+1
      IF(J.LE.K) GO TO 4
      T=X(K)
      X(K)=X(J)
      X(J)=T
    4 CONTINUE
      DO 5 K=1,N
    5 X(K)=X(K)*U
      RETURN
      END
```

```
      FUNCTION INV(J,NU)
      J1=J
      INV=0
      DO 1 I=1,NU
      J2=J1/2
      INV=INV*2+J1-2*J2
    1 J1=J2
      RETURN
      END
```

Fig. 3.14. A FORTRAN IV subroutine which performs Fourier transformation through the FFT algorithm. It converts the input complex vector X of dimension N and NU = $\log_2$ N into its Fourier transform if IT = 1; if IT = −1, the vector X is converted into its inverse Fourier transform. The function INV effects the binary bit inversion. Note that the output component X(1) stands for the Nth ≡ 0th Fourier coefficient and that all other components are correspondingly shifted to one higher value.

have been developed for arbitrary N which work on similar principles (Bergland, 1967, 1968, 1969; Rader, 1968; Singleton, 1968). When the data arrays are very large and exceed the machine memory storage capacity, the use of auxiliary memory devices such as disk or tape has to be integrated properly into the algorithm. These problems have been tackled (Buijs, 1969; Singleton, 1967). Convolution and correlation of finite signals can also be profitably handled through the FFT in their many applications. The calculation of the convolution (3.8) or correlation (3.18) of two vectors involves N^2 complex products. As the FFT takes only $N \log_2 N$ operations, we may proceed to use the Fourier transform first for the two coordinate sets, multiply them in the φ-basis [Eqs. (3.7b) or (3.16)], and then Fourier-transform back. The number of operations in this roundabout way is $3N \log_2 N + N$, which is less than N^2 for $N > 16$.

The actual applications of the fast Fourier transform algorithm cover a very wide range. Some examples of what can be found in the literature are the

articles by Stockham (1966), Singleton and Poulter (1967), Welch (1967), Glisson and Black (1969), Liu and Fagel (1971), and Becker and Farrar (1972). For the reader interested in a more detailed exposition and bibliography of this rapidly growing field and its applications, we suggest the book by Brigham (1974) as well as the special issues of the *IEEE Transactions on Audio and Electroacoustics* **AU-15** (June 1967) and **AU-17** (June 1969).

3.4. The "Limit" $N \to \infty$: Fourier Series and Integral Transforms

Up to now we have dealt with complex vector spaces $\mathscr{V}^N$ with N arbitrary but finite. We shall now let N grow without bound and examine the behavior of the Fourier transform. The "limit" $N \to \infty$ is not meant to imply that $\mathscr{V}^N$ *tends* toward a "$\mathscr{V}^\infty$" since insofar as vector spaces are concerned, no convergence of the kinds familiar to the reader is defined. Yet for coordinates, inner products, and norms such a limit makes sense if focused properly. Moreover, it provides a reliable intuitive grasp of the properties of infinite-dimensional vector spaces.

3.4.1. $(2N + 1)$-Dimensional Spaces

For the following, it will prove convenient to consider $(2N + 1)$-dimensional spaces $\mathscr{V}^{2N+1}$ where basis vectors are numbered by indices with the range $-N, -N + 1, \ldots, -1, 0, 1, \ldots, N - 1, N$. The Fourier transforms between the coordinates of a vector in the $\boldsymbol{\varepsilon}$- and $\boldsymbol{\varphi}$-bases [Eqs. (1.51)] become

$$\tilde{f}_m = (2N + 1)^{-1/2} \sum_{n=-N}^{N} f_n \exp[2\pi inm/(2N + 1)], \tag{3.30a}$$

$$f_n = (2N + 1)^{-1/2} \sum_{m=-N}^{N} \tilde{f}_m \exp[-2\pi inm/(2N + 1)]. \tag{3.30b}$$

Recall that, to start with, Eqs. (1.51) defined the range of the indices as congruent modulo the dimension of the space. We shall now introduce a new indexing system for the vectors in the $\boldsymbol{\varphi}$-basis, defining

$$x \coloneqq \pi(2m + 1)/(2N + 1), \tag{3.31}$$

so that, for $m = -N, \ldots, N$, x will range in steps of

$$\Delta x = 2\pi/(2N + 1) \tag{3.32}$$

from $-\pi + \Delta x$ to π, and while the numbers m are considered modulo $2N + 1$, the numbers x are considered modulo 2π. We shall also define the set of quantities related to the $\boldsymbol{\varphi}$-basis coordinates of $\mathbf{f}$ as

$$f(x) \coloneqq [(2N + 1)/2\pi]^{1/2} \tilde{f}_m, \tag{3.33}$$

which can be seen as a function f of $2N + 1$ equidistant points on a circle. When N grows without bound these points will become *dense* on the interval $(-\pi, \pi]$; the point $-\pi$ is excluded from the interval as it is congruent with π. Changing dummy indices, Eqs. (3.30) appear as

$$f(x) = (2\pi)^{-1/2} \sum_{n=-N}^{N} f_n \exp(inx) \exp[-in\pi/(2N + 1)], \tag{3.34a}$$

$$f_n = (2\pi)^{-1/2} \sum_{x=-\pi+\Delta x}^{\pi} \Delta x f(x) \exp(-inx) \exp[in\pi/(2N + 1)]. \tag{3.34b}$$

The substitutions (3.31)–(3.33) can also be made for the Parseval identity, Eq. (1.43), which now reads

$$(\mathbf{f}, \mathbf{g}) = \sum_{n=-N}^{N} f_n^* g_n = \sum_{x=-\pi+\Delta x}^{\pi} \Delta x f(x)^* g(x). \tag{3.35}$$

3.4.2. Fourier Series

The reader can see that Eqs. (3.34) and (3.35) lend themselves quite naturally to the limit $N \to \infty$: the sums over x have the right form to be turned into Riemann integrals. Some precautions must be taken, though. We introduce first, for every function $h(x)$ of the (discrete) variable x, a *step function* $h_{(N)}(x')$ over the *continuous* variable x' by

$$h_{(N)}(x') = h(x), \qquad x' \in (x - \Delta x/2, x + \Delta x/2]. \tag{3.36}$$

See Fig. 3.15. By this device, the $\sum_x \Delta x \cdots$ can be turned into $\int_{-\pi}^{\pi} dx' \cdots$. We can assume (at this stage) that the limit of $h_{(N)}(x')$ as $N \to \infty$ is a "proper" function $h(x')$ of x', e.g., a continuous function with a finite number of discontinuities so that it is Riemann-integrable. Next, Eq. (3.35) for $\mathbf{f} = \mathbf{g}$ states that the sum of the now-infinite series in the middle term must equal

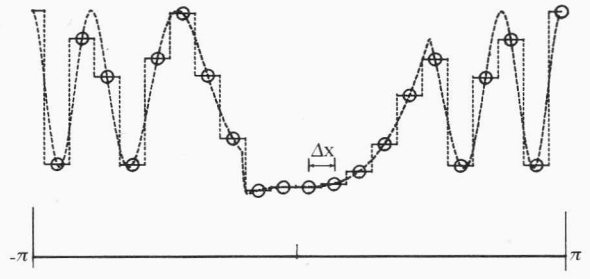


Fig. 3.15. An N-step function in $(-\pi, \pi]$ approximating a continuous function in the limit $N \to \infty$.

the assumed definite value of the integral. The coefficients $\{f_n\}_{n=-\infty}^{\infty}$ must satisfy some summability condition. They cannot all be equal, for instance. A finite number may be nonzero, or we may ask for appropriate decrease conditions for f_n as $n \to \infty$. In particular, we shall agree *not* to allow any f_n "near to" N to keep a finite value as $N \to \infty$. If these (admittedly vague) conditions are met, the sum in (3.34a) becomes a series where the exponential factor $\exp[\pi in/(2N+1)] \to 1$ as $N \to \infty$, and the same happens in (3.34b). The pair of equations then becomes

$$f(x) = (2\pi)^{-1/2} \sum_{n=-\infty}^{\infty} f_n \exp(inx), \tag{3.37a}$$

$$f_n = (2\pi)^{-1/2} \int_{-\pi}^{\pi} dx f(x) \exp(-inx), \qquad n = 0, \pm 1, \pm 2, \ldots. \tag{3.37b}$$

The first of these is the *Fourier expansion* of $f(x)$ in terms of the functions $\exp(inx)$, $n \in \mathscr{Z}$ (the set of integers), with *Fourier partial-wave coefficients* $\{f_n\}_{n\in\mathscr{Z}}$. These coefficients can be obtained from the original function by the second equation. Equations (3.37a) and (3.37b) are also referred to as the Fourier *synthesis* and *analysis* of the function $f(x)$. Finally, the Parseval identity (3.35) becomes

$$(\mathbf{f}, \mathbf{g}) = \sum_{n=-\infty}^{\infty} f_n^* g_n = \int_{-\pi}^{\pi} dx f(x)^* g(x). \tag{3.38}$$

The precise range of validity of the Fourier series pair (3.37) and Eq. (3.38) is given by the *Dirichlet conditions*, which will be proven independently of this construction in Section 4.2. We would only remark here that when $f(x)$ is a *trigonometric polynomial* of degree M, i.e., when the sum in (3.37a) is finite and n bounded,

$$f_M(x) = (2\pi)^{-1/2} \sum_{|n|\leq M} f_n \exp(inx), \tag{3.39}$$

then (3.37b) can be immediately verified by multiplying by $(2\pi)^{-1/2} \exp(-imx)$ and integrating x over $(-\pi, \pi)$, using

$$\int_{-\pi}^{\pi} dx \exp[i(n-m)x] = \begin{cases} [i(n-m)]^{-1} \exp[i(n-m)x]\big|_{-\pi}^{\pi} = 0, & n \neq m, \\ \int_{-\pi}^{\pi} dx = 2\pi, & n = m. \end{cases} \tag{3.40}$$

Indeed,

$$(2\pi)^{-1/2} \int_{-\pi}^{\pi} dx f_M(x) \exp(-imx)$$

$$= (2\pi)^{-1} \sum_{|n|\leq M} f_n \int_{-\pi}^{\pi} dx \exp[i(n-m)x] = f_m, \tag{3.41}$$

and (3.38) can be similarly proven using (3.37b) and (3.40). In restricting the degree of the polynomial to finite M, we have avoided the question of whether the infinite series (3.37a) converges to $f(x)$ for all x and what to do if the series diverges.

3.4.3. Basis Vectors

Since in Part II we shall tackle these questions using elements of functional analysis, let us have a closer look here at the vector space aspects of $\mathscr{V}^{2N+1}$ as N grows without bound. Parallel to the redefinition of the coordinates (3.33) of a vector $\mathbf{f}$, we define the basis vectors

$$\boldsymbol{\delta}_x := [(2N + 1)/2\pi]^{1/2}\boldsymbol{\varphi}_m, \tag{3.42}$$

where x and m are related by (3.31). These also constitute a *basis* for $\mathscr{V}^{2N+1}$, with coordinates

$$(\boldsymbol{\varepsilon}_n, \boldsymbol{\delta}_x) = (2\pi)^{-1/2} \exp(-inx) \exp[in\pi/(2N + 1)] = (\boldsymbol{\delta}_x, \boldsymbol{\varepsilon}_n)^*. \tag{3.43}$$

They are a set which is orthogonal, but not orthonormal, as

$$(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y) = \delta_{x,y}(2N + 1)/2\pi = \delta_{x,y}/\Delta x, \tag{3.44}$$

where $\delta_{x,y}$ is the Kronecker δ in the indices x and y. The coordinates of a vector $\mathbf{f}$ in the $\boldsymbol{\delta}$-basis are thus

$$f(x) = (\boldsymbol{\delta}_x, \mathbf{f}) = \Big(\boldsymbol{\delta}_x, \sum_y \Delta y f(y)\boldsymbol{\delta}_y\Big) = \sum_{y=-\pi+\Delta y}^{\pi} \Delta y f(y)(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y). \tag{3.45}$$

3.4.4. The Dirac δ

Whereas all expressions before (3.42) had a clear meaning as $N \to \infty$, step functions (3.36) being used and assumed to converge to Riemann-integrable functions, the step function of y corresponding to $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y)$ for fixed x is a rectangle of width $\Delta y = 2\pi/(2N + 1)$ and height $1/\Delta y$ (thus of *unit area*) centered in x. As $N \to \infty$, $\Delta y \to 0$. If we take Eq. (3.45) seriously, it tells us that such a "function" in the "limit" $N \to \infty$ has the properties

$$\delta(x - y) := (\boldsymbol{\delta}_x, \boldsymbol{\delta}_y) = \delta(y - x), \tag{3.46a}$$

$$\delta(x - y) = 0 \qquad \text{for } x \neq y, \tag{3.46b}$$

$$\int_{-\pi}^{\pi} dy \delta(x - y)f(y) = f(x) = (\boldsymbol{\delta}_x, \mathbf{f}). \tag{3.46c}$$

The symbol $\delta(x - y)$ *defined* by (3.46b) and (3.46c) is the *Dirac* δ. (The definition can be made slightly weaker.) It is not a true function. In the

rigorous framework of distribution theory, the second equality in (3.46c) is the definition of $\boldsymbol{\delta}_x$ as a *functional* or distribution, i.e., a mapping which assigns, to every function $\mathbf{f}$ in some class, a number $f(x)$. The intuitive development we have followed here is one of the standard approaches in mathematical physics, which views the Dirac δ as the symbol indicating the limit of a sequence of integrals containing the continuous function $f(x)$ and a rectangle function $\delta_N(x - y)$ of unit area centered on x whose width vanishes as $N \to \infty$. This is equivalent to the first equality in (3.46c); that is, it "punches out" the value of the *test function* $f(y)$ at the point x.

From the point of view of vector analysis, the function $f(x)$, $x \in (-\pi, \pi]$, can here be seen as the *coordinates* of a vector $\mathbf{f}$ in the $\boldsymbol{\delta}$-basis [second equality in (3.46c)], while its Fourier partial-wave coefficients $\{f_n\}_{n \in \mathscr{Z}}$ are the coordinates of the same $\mathbf{f}$ in the $\boldsymbol{\varepsilon}$-basis.

3.4.5. Fourier Integral Transforms

Another way in which the $N \to \infty$ "limit" of the finite Fourier transform leads to *integral* transforms is the following. Consider again the pair of equations (3.30) in $\mathscr{V}^{2N+1}$ for growing N and introduce new indexing variables in both the $\boldsymbol{\varepsilon}$- and $\boldsymbol{\varphi}$-bases as

$$q \coloneqq [2\pi/(2N + 1)]^{1/2}m, \qquad p \coloneqq [2\pi/(2N + 1)]^{1/2}n. \tag{3.47}$$

For $n, m = -N, \ldots, N$, q and p will correspondingly range over $2N + 1$ points spaced by decreasing intervals

$$\Delta q = [2\pi/(2N + 1)]^{1/2} = \Delta p \tag{3.48}$$

between, approximately, $\pm(\pi N)^{1/2}$. Now define the functions

$$f(q) = [(2N + 1)/2\pi]^{1/4}\tilde{f}_m, \qquad \tilde{f}(p) = [(2N + 1)/2\pi]^{1/4}f_n \tag{3.49}$$

on these points. Substituting these expressions into (3.30) and following the same procedure as before in defining step functions $f_{(N)}(q)$ and $\tilde{f}_{(N)}(p)$ for the continuous variables q and p (Fig. 3.15), assuming that as $N \to \infty$ these step functions converge to Riemann-integrable functions in the expanding integration interval and substituting $\int dq$ for $\sum \Delta q$, etc., we arrive at

$$f(q) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} dp\tilde{f}(p) \exp(ipq), \tag{3.50a}$$

$$\tilde{f}(p) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} dqf(q) \exp(-ipq), \qquad q, p \in \mathscr{R}. \tag{3.50b}$$

From the Parseval identity (1.43) we find similarly

$$(\mathbf{f}, \mathbf{g}) = \int_{-\infty}^{\infty} dqf(q)^*g(q) = \int_{-\infty}^{\infty} dp\tilde{f}(p)^*\tilde{g}(p). \tag{3.51}$$

These equations are the analogues of (3.37) and (3.38). The function $\tilde{f}(p)$ is the *Fourier integral transform* of $f(q)$, and the latter the *inverse* Fourier transform of the former. A closer examination of the validity of (3.50)–(3.51) for different classes of functions, not necessarily integrable in the sense of Riemann, will be undertaken in Part III. Again, orthogonal bases $\{\boldsymbol{\delta}_q\}$ and $\{\tilde{\boldsymbol{\delta}}_p\}$ can be defined so that $f(q) = (\boldsymbol{\delta}_q, \mathbf{f})$ and $\tilde{f}(p) = (\tilde{\boldsymbol{\delta}}_p, \mathbf{f})$, leading to Dirac δ's with properties (3.46b) and (3.46c) on the full real line $\mathscr{R}$.

The description of infinite-dimensional spaces as "limits" of finite-dimensional ones has been made here with the purpose of giving an intuitive grasp of the subject. In Parts II and III a physicist's à la Dirac approach will be given. We shall not embark here on a mathematically complete survey of this topic in part because of space and time but mainly because once the overall picture is drawn and the relevant pitfalls are pointed out, the tools of infinite-dimensional vector analysis can be used with the same operational facility as in the finite-dimensional case.