# Chapter 2
# Digital Fourier Transforms

As discussed in Ch. 1, scalar diffraction theory is the physical basis of wave-optics simulations. A result of this theory is that propagation of electromagnetic waves through vacuum may be treated as a linear system. For monochromatic waves, the vector magnitude of the electric field in the observation plane of a system is the convolution of the vector magnitude of the electric field in the source plane and the free-space impulse response.[5] Consequently, the tools of linear-systems theory and Fourier analysis are indispensable for studying wave optics. These topics are discussed in Ch. 4 and beyond. In those chapters, discrete Fourier transforms are applied to obtain computationally efficient algorithms for the simulations. First, the basic computational algorithms must be discussed.

As in many areas of science and engineering, most problems encountered while researching complex optical systems are analytically intractable. Consequently, most calculations regarding the inner workings and performance of optical systems are performed by numerical simulation on computers. Fortunately, sampling theory and discrete-Fourier-transform (DFT) theory provide many important lessons for optics researchers who perform such simulations. With due consideration to the limitations imposed by performing computations on sampled functions, there is much to be gained from numerical simulation of optical-wave propagation.

## 2.1 Basics of Digital Fourier Transforms

This section covers the basics of computing DFTs that match the corresponding analytic results. This includes proper scaling, correct use of spatial and spatial-frequency coordinates, and use of DFT software.

### 2.1.1 Fourier transforms: from analytic to numerical

There are a few common conventions for defining the FT operation and its inverse. This book defines the continuous FT $G(f_x)$ of a spatial function $g(x)$ and its inverse as

$$G(f_x) = \mathcal{F}\{g(x)\} = \int_{-\infty}^{\infty} g(x)\, e^{-i2\pi f_x x}\, dx \tag{2.1}$$

$$g(x) = \mathcal{F}^{-1}\{G(f_x)\} = \int_{-\infty}^{\infty} G(f_x)\, e^{i2\pi f_x x}\, df_x, \tag{2.2}$$

where $x$ is the spatial variable, and $f_x$ is the spatial-frequency variable. The first step to discretize the FT is writing the integral as a Riemann sum:

$$\begin{aligned} G(f_{xm}) &= \mathcal{F}\{g(x_n)\} \\ &= \sum_{n=-\infty}^{\infty} g(x_n)\, e^{-i2\pi f_{xm} x_n} (x_{n+1} - x_n), \quad m = -\infty, \ldots \infty, \end{aligned} \tag{2.3}$$

where $n$ and $m$ are integers. Computer calculations can only work with a finite number of samples $N$, and this book discusses only even $N$ for reasons that are discussed later. Further, typical DFT software requires a fixed sampling interval. The sampling interval is $\delta$, and so $x_n = n\delta$. Then, the frequency domain interval is $\delta_f = 1/(N\delta)$ such that $f_{xm} = m\delta_f = m/(N\delta)$. Eq. (2.3) becomes

$$\begin{aligned} G\left(\frac{m}{N\delta}\right) &= \mathcal{F}\{g(n\delta)\} \\ &= \delta \sum_{n=-N/2}^{N/2-1} g(n\delta)\, e^{-i2\pi mn/N}, \qquad m = -N/2, 1 - N/2, \ldots N/2 - 1. \end{aligned} \tag{2.4}$$

The last step is to format the samples for the DFT software. Such software is available for many programming languages. Examples in this book use the MATLAB scripting language, which has DFT routines in its core function library.[7] Other programming languages such as C, C++, FORTRAN, and Java do not have DFT routines in their core libraries, but DFT algorithms are described in many books,[8] and DFT software is readily available from third-party suppliers.[9–11] MATLAB uses positive indices (also called one-based indexing). To account for only positive indices, the order of the spatial samples inside the sum must be rearranged such that

$$g_{n'} = \begin{cases} g\left[\left(n' + \frac{N}{2}\right)\delta\right] & \text{for} \quad n' = 1, 2, \ldots \frac{N}{2} + 1 \\ g\left[(n' - N - 2)\,\delta\right] & \text{for} \quad n' = \frac{N}{2} + 2, \frac{N}{2} + 3, \ldots N. \end{cases} \tag{2.5}$$

For a one-dimensional DFT, this amounts to circularly shifting the samples in the spatial domain so that the origin corresponds to the first sample, as illustrated in Fig. 2.1.

The reordering of spatial samples means that the samples in the spatial-frequency domain end up out of order, too. We denote the new index in the spatial-frequency domain as $m'$, which finally leads to the form of the DFT equation:

$$G_{m'} = \delta \sum_{n'=1}^{N} g_{n'} e^{-i2\pi(m'-1)(n'-1)/N}, \qquad m' = 1, 2, \ldots N. \tag{2.6}$$

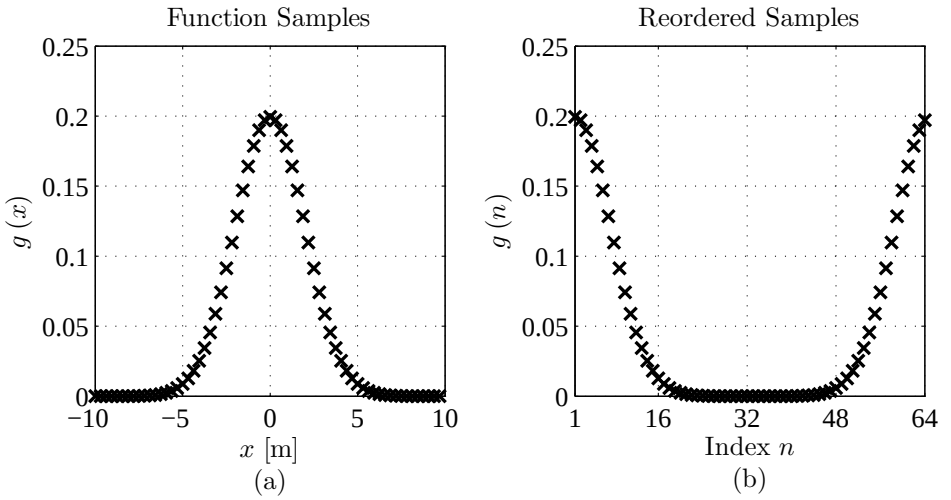


**Figure 2.1** An illustration of reordering samples in the spatial domain in preparation for the DFT. Plot (a) shows a Gaussian function in the spatial domain. Plot (b) shows the samples of plot (a) reordered. The reordering essentially circularly shifts the samples so that the origin is at the first element.

MATLAB's DFT software computes everything in Eq. (2.6) except for multiplying by $\delta$, as is typical. That is left to the user.

### 2.1.2 Inverse Fourier transforms: from analytic to numerical

Discrete IFTs (DIFTs) operate very similarly to DFTs. As before, the first step is to write the integral in Eq. (2.2) as a Riemann sum:

$$\begin{aligned} g(x_n) &= \mathcal{F}^{-1}\{G(f_{xm})\} \\ &= \sum_{m=-\infty}^{\infty} G(f_{xm})\, e^{i2\pi f_{xm}\, x_n}\, (f_{x,m+1} - f_{x,m}), \qquad n = -\infty, \ldots \infty. \end{aligned} \tag{2.7}$$

Again, with a finite number of samples $N$ and uniform sample spacing $\delta_f = 1/(N\delta)$ in the frequency domain, the sum becomes

$$\begin{aligned} g(n\delta) &= \mathcal{F}^{-1}\{G(f_{xm})\} \\ &= \delta_f \sum_{m=-N/2}^{N/2-1} G\left(\frac{m}{N\delta}\right) e^{i2\pi mn/N}, \qquad n = -N/2, 1 - N/2, \ldots N/2 - 1. \end{aligned} \tag{2.8}$$

Then, the use of positive indices results in reordering of the samples similar to what happens in the forward DFT. The result is

$$g_{n'} = \frac{1}{N\delta} \sum_{m'=1}^{N} G_{m'} e^{i2\pi(m'-1)(n'-1)/N}, \qquad n' = 1, 2, \ldots N. \tag{2.9}$$

**Listing 2.1** Code for performing a DFT in MATLAB.

```
function G = ft(g, delta)
% function G = ft(g, delta)
    G = fftshift(fft(fftshift(g))) * delta;
```

**Listing 2.2** Code for performing a DIFT in MATLAB.

```
function g = ift(G, delta_f)
% function g = ift(G, delta_f)
    g = ifftshift(ifft(ifftshift(G))) ...
        * length(G) * delta_f;
```

DFT software typically computes everything in Eq. (2.9) except for multiplying by $\delta^{-1}$.

### 2.1.3 Performing discrete Fourier transforms in software

MATLAB is one of many software applications that provide DFT functionality.[9–11] Specifically, it includes the functions `fft` and `ifft` for performing one-dimensional DFTs using the fast Fourier-transform (FFT) algorithm. The FFT algorithm works only for values of $N$ that are an integer power of two. Now, this is common practice, but using powers of two is not entirely necessary anymore because of sophisticated DFT software like FFTW (Fastest Fourier Transform in the West).[9] Computational efficiency for DFTs is maximized when $N$ is a power of two, although depending on the value, other lengths can be computed nearly as fast. In any case, we restrict our discussions to only even $N$, as previously mentioned. Listings 2.1 and 2.2 give functions that compute a properly scaled FT and IFT, making use of `fft` and `ifft`. Listing 2.1 evaluates Eq. (2.6) including the reordering in both domains using the function `fftshift`. Listing 2.2 evaluates Eq. (2.9) including the reordering in both domains using the function `ifftshift`.

Listings 2.3 and 2.4 give examples of computing properly scaled DFTs, making use of `ft` and `ift`, and Figs. 2.2 and 2.3 illustrate the results. In the first example, both the spatial function and its spectrum are real and even. In the second example, the spatial function is a shifted version of that from the first example. The result of the shift is a non-zero phase in the spectrum.

Figure 2.2 shows that the DFT values for a Gaussian function match the analytic FT values closely. The most notable departure is at $f_x = 0$. However, if the original function were to be synthesized from the DFT values shown in Fig. 2.2, any error at $f_x = 0$ would only affect the mean value of synthesized function, not its structure.

Figure 2.3 shows that the DFT values for a shifted Gaussian function match the

**Listing 2.3** MATLAB example of performing a DFT with comparison to the analytic FT. The spatial function is real and even.

```
% example_ft_gaussian.m

% function values to be used in DFT
L = 5;       % spatial extent of the grid
N = 32;      % number of samples
delta = L / N; % sample spacing
x = (-N/2 : N/2-1) * delta;
f = (-N/2 : N/2-1) / (N*delta);
a = 1;
% sampled function & its DFT
g_samp = exp(-pi*a*x.^2); % function samples
g_dft = ft(g_samp, delta); % DFT
% analytic function & its continuous FT
M = 1024;
x_cont = linspace(x(1), x(end), M);
f_cont = linspace(f(1), f(end), M);
g_cont = exp(-pi*a*x_cont.^2);
g_ft_cont = exp(-pi*f_cont.^2/a)/a;
```

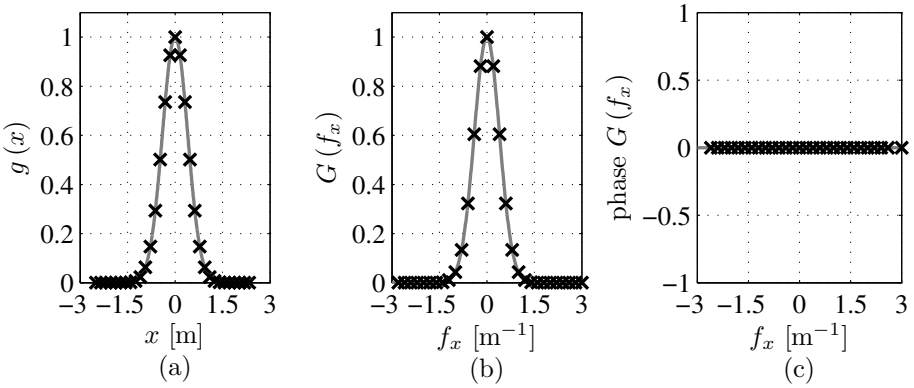


**Figure 2.2** A Gaussian function and its properly scaled DFT plotted with its analytic counterpart.

**Listing 2.4** MATLAB example of performing a DFT with comparison to the analytic FT. The spatial function is real but asymmetric.

```
% example_ft_gaussian_shift.m

L = 10;      % spatial extent of the grid
N = 64;      % number of samples
delta = L / N; % sample spacing
x = (-N/2 : N/2-1) * delta;
x0 = 5*delta;
f = (-N/2 : N/2-1) / (N*delta);
a = 1;
% sampled function & its DFT
g_samp = exp(-pi*a*(x-x0).^2); % function samples
g_dft = ft(g_samp, delta); % DFT
% analytic function & its continuous FT
M = 1024;
x_cont = linspace(x(1), x(end), M);
f_cont = linspace(f(1), f(end), M);
g_cont = exp(-pi*a*(x_cont-x0).^2);
g_ft_cont = exp(-i*2*pi*x0*f_cont) ...
    .* exp(-pi*f_cont.^2/a)/a;
```

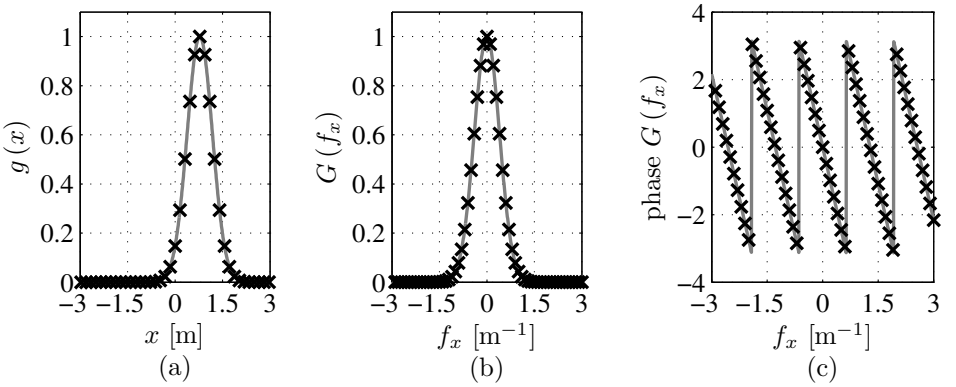


**Figure 2.3** A shifted Gaussian function and its properly scaled DFT plotted with its analytic counterpart. Plot (a) shows the spatial function and its sample values. Plot (b) shows the modulus of the analytic FT and the modulus of the DFT. Plot (c) shows the analytic phase of the FT and the phase of the DFT.

analytic FT values closely. The spatial shift moved the Gaussian pulse toward one edge of the grid. As a result, the grid had to be extended to twice the size shown in Fig. 2.2 by doubling the number of samples. Without the increased number of samples, the phase in the spatial-frequency domain would match the analytic result only in the center of the spectrum.

## 2.2 Sampling Pure-Frequency Functions

A very important issue in achieving accurate results with FTs and FT-based calculations is determining the necessary grid spacing $\delta$ and number of grid points $N$. This is an important distinction between Figs. 2.2 and 2.3. The highest significant frequency in the shifted Gaussian signal is higher than that in the centered Gaussian. Accordingly, the shifted Gaussian requires more samples to adequately represent its spectrum. The reasons for this requirement are discussed in this section.

The Whittaker-Shannon sampling theorem states that a bandlimited signal having no spectral components above $f_{max}$ can be uniquely determined by values sampled at uniform intervals of $\delta_c = 1/(2f_{max})$.[5,12] The Nyquist sampling frequency is defined as $f_c = 1/\delta_c = 2f_{max}$. The requirement for sampling frequencies higher than $f_c$ is called the Nyquist sampling criterion. Essentially, this means that there must be at least two samples per period for the highest frequency component of the signal. If the sample spacing is larger than $\delta_c$, it may not be possible to reconstruct each frequency component uniquely. This can be a problem for DFTs.

The simplest way to illustrate sampling effects is with pure sinusoidal signals. The following discussion can be extended to any Fourier-transformable signal by applying the Fourier integral representation. This section uses signals of the form

$$g(x) = \cos(2\pi f_0 x) \tag{2.10}$$

to illustrate some aspects of sampling related to this theorem. In this type of signal, the frequency is $f_0$ and the period is $T = 1/f_0$. The required grid spacing is $\delta_c = 1/(2f_0)$, corresponding to two samples per period.

Figure 2.4 shows such a sinusoidal signal. This particular signal, shown by the solid gray line, has a frequency of $6\ \mathrm{m}^{-1}$. Samples of the signal, separated by $\delta_0 = 1/12\ \mathrm{m} = 0.0833\ \mathrm{m}$, are shown in the black $\times$'s. The samples are located at all of the peaks and troughs of the signal. Now, if we were given these samples without knowledge of the signal from which they were drawn, could we uniquely identify the signal? Actually, there are are many other sinusoidal signals that could have produced these samples. For example, $\cos(4\pi f_0 x)$ could produce the samples shown; however, there is no frequency lower than $f_0$ that could have produced these samples. Further, the only signal satisfying the Nyquist criterion is $f_0$. Now, we realize that if we are given the samples and the fact that they satisfied the Nyquist criterion, we could certainly identify the signal uniquely.

As a counter-example, we consider sinusoidal signals that are sampled on grids that do not satisfy the Nyquist criterion. Figure 2.5 shows two such signals. In

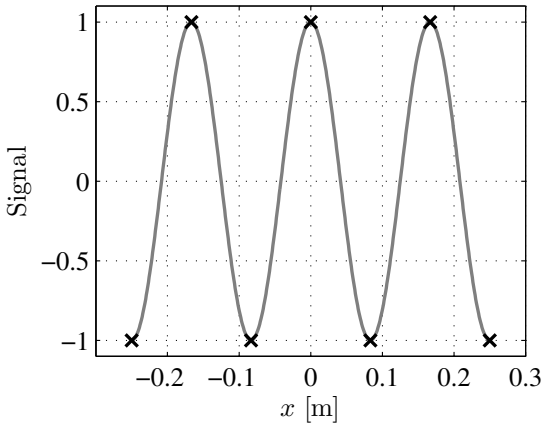


**Figure 2.4** Example of a sinusoidal signal (gray line) that is properly sampled. There is no lower frequency that could produce the samples shown.

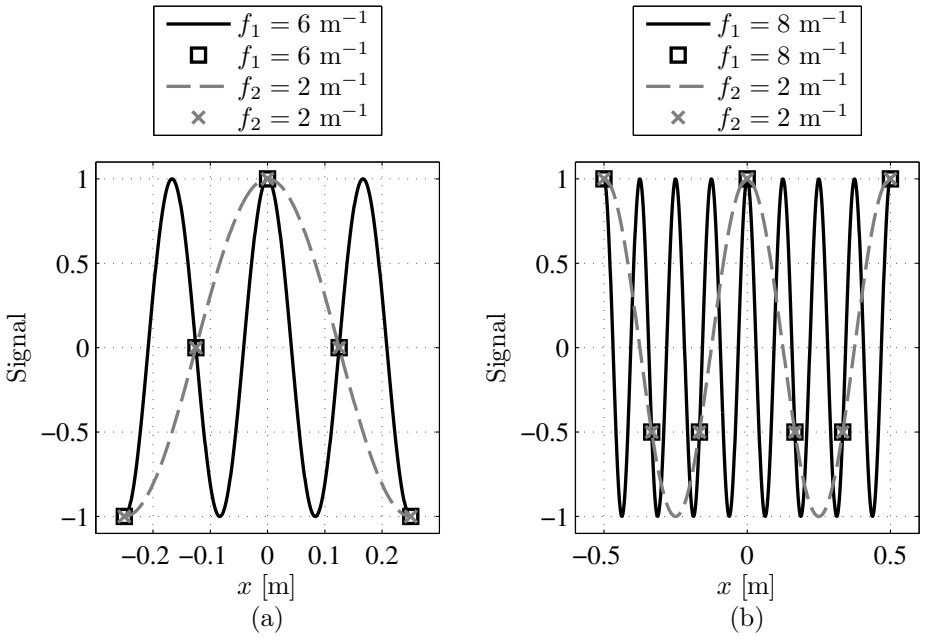


**Figure 2.5** Example of a sinusoidal signal (gray line) that is sampled too coarsely. Samples taken from both frequencies are identical.

plot (a), the solid black line shows a cosine signal with frequency $f_1 = 6\ \mathrm{m}^{-1}$. Properly sampling this signal would require a sample spacing of $1/12\ \mathrm{m} = 0.0833\ \mathrm{m}$. The black squares show samples of this signal that are separated by $\delta = 1/8\ \mathrm{m} = 0.125\ \mathrm{m}$. Now, let us consider the other signal in plot (a). The gray dashed line shows a signal with frequency $f_2 = 2\ \mathrm{m}^{-1}$, and the gray $\times$s show its samples. The samples from the two different frequencies are identical! In the previous example of a properly-sampled function, only frequencies that are multiples of the original, in this case $f_1$, could produce the given samples. None of those harmonics would be properly sampled, though. Now, when the signal is undersampled, there is at least one lower (and properly-sampled) frequency that could produce the given samples. If we were given these samples and someone asked us to identify the signal's frequency, and we answered with a properly-sampled signal (satisfying the Nyquist criterion), like $2\ \mathrm{m}^{-1}$, we would be incorrect.

This is not a rare occurrence; plot (b) shows another undersampled example with $f_1 = 8\ \mathrm{m}^{-1}$ sampled with a grid spacing of $1/6\ \mathrm{m} = 0.167\ \mathrm{m}$. Again, the gray dashed line shows a signal with frequency $f_2 = 2\ \mathrm{m}^{-1}$, and its samples shown in gray $\times$s are identical to those taken from the higher frequency. When the grid spacing is too coarse, the improperly-sampled, high-frequency sinusoids appear as properly-sampled, lower frequencies. This effect is called aliasing.

Returning to other signals that can be written as a sum or integral of sinusoids, we need to know the highest frequency component and then compute the grid spacing from there. If the highest frequency is properly sampled, so are all of the lower frequencies. This seems like a simple solution, but there are many examples in this book that are not so straightforward, and even cases in which we can (and probably should) relax this constraint. The next section gives a more detailed treatment.

## 2.3 Discrete vs. Continuous Fourier Transforms

DFT pairs differ from their continuous counterparts in three important ways:

- spatial domain sampling,
- a finite spatial grid,
- and spatial-frequency-domain sampling.

These three properties result in three distortions to continuous FT pairs when they are computed discretely:

- aliasing in the spatial-frequency domain,
- rippling and smearing in the spatial-frequency domain,
- and virtual periodic replication in the spatial domain.

These effects are illustrated more formally here in a development that closely follows the approach of Brigham.[8] Let a known FT pair be

$$g(x) \Leftrightarrow G(f_x), \tag{2.11}$$

and let the sampled versions of these functions be

$$\widetilde{g}(x) \Leftrightarrow \widetilde{G}(f_x), \tag{2.12}$$

respectively. The next few equations develop the sampled FT pair. Figure 2.6 shows the graphical development. The figure uses

$$g(x) = \exp(-a|x|) \tag{2.13}$$

$$G(f_x) = \frac{1}{a}\frac{2}{1+(2\pi f_x/a)^2} \tag{2.14}$$

as the example FT pair to illustrate the effects of discretization. This is for illustration purposes; the effects would be the same for any other FT pair. Plots of Eqs. (2.13) and (2.14) are shown in Figs. 2.6 (a) and (b) for $a = 10\,\mathrm{m}^{-1}$. The peak value of the spectrum is $0.2$.

To begin accounting for discretization, $g(x)$ is sampled by multiplication with a $\mathrm{comb}$ function with spacing $\delta$. Multiplication in the spatial domain is equivalent to convolution in the spatial-frequency domain (for a discussion of convolution, see Ch. 3), which transforms the pair in Eq. (2.11) into

$$g(x)\frac{1}{\delta}\mathrm{comb}\left(\frac{x}{\delta}\right) \Leftrightarrow G(f_x) \otimes \mathrm{comb}(\delta f_x). \tag{2.15}$$

Figures 2.6(c) and (d) show the impact of sampling in the spatial domain for $\delta = 0.0375$ m. This results in periodic replication in the spatial-frequency domain. This is visible in the tails of the frequency spectrum that lift up at large positive and negative frequencies. That is an artifact that is not present in the analytic spectrum shown in Fig. 2.6(b).

Next, representing $g(x)$ on a grid of finite size $L$ changes the pair into

$$g(x)\frac{1}{\delta}\mathrm{comb}\left(\frac{x}{\delta}\right)\mathrm{rect}\left(\frac{x}{L}\right) \Leftrightarrow G(f_x) \otimes \mathrm{comb}(\delta f_x) \otimes [L\ \mathrm{sinc}(Lf_x)]. \tag{2.16}$$

Figures 2.6(e) and (f) show the impact of the finite sample width, $L = 0.6\,\mathrm{m}$. In the spatial domain, the tails of $g(x)$ are lost. In the spatial-frequency domain, the spectrum is multiplied by $L$ and convolved with a $\mathrm{sinc}$ function, which causes rippling and smearing.

Finally, the result of the DFT is an array of the sampled values of $G(f_x)$. This makes one final modification to the FT pair so that

$$\widetilde{g}(x) = \left[g(x)\frac{1}{\delta}\mathrm{comb}\left(\frac{x}{\delta}\right)\mathrm{rect}\left(\frac{x}{L}\right)\right] \otimes \left[\frac{1}{L}\mathrm{comb}\left(\frac{x}{L}\right)\right] \tag{2.17}$$

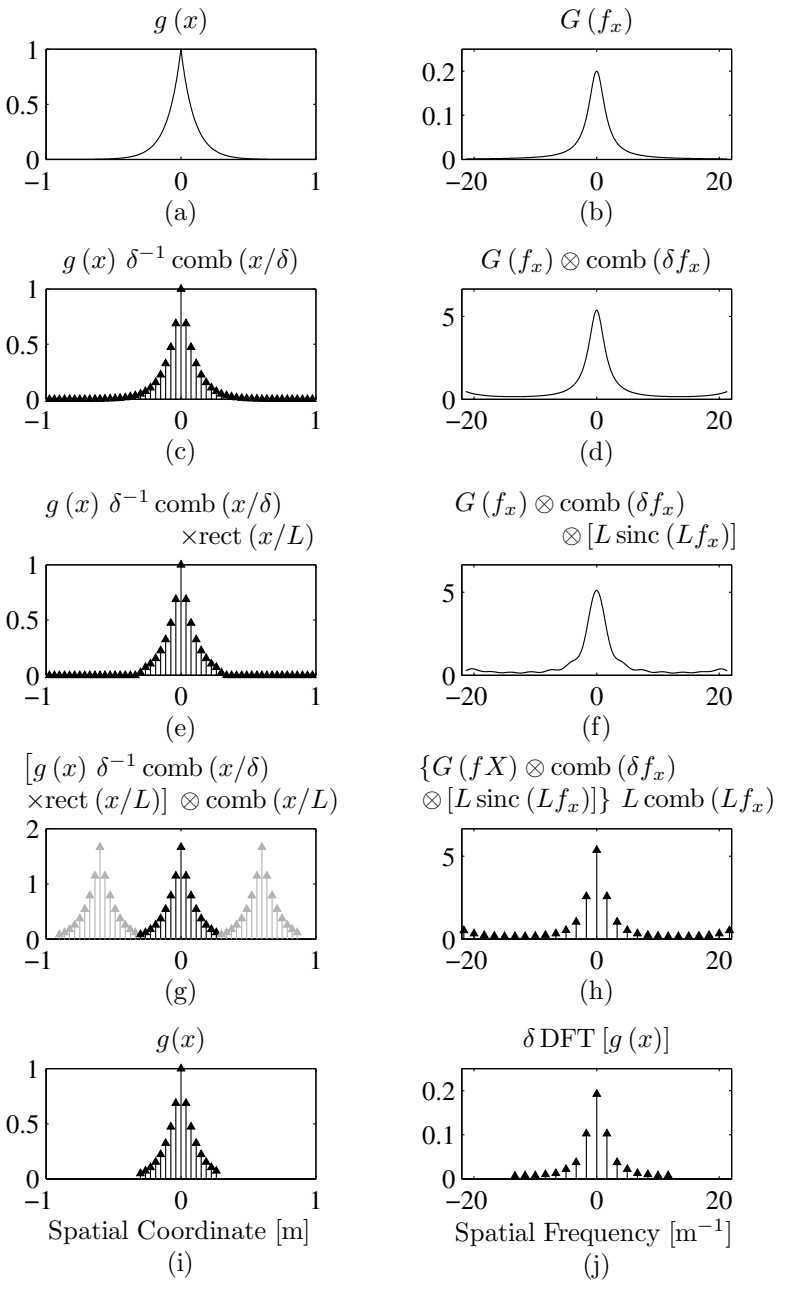


**Figure 2.6** Graphical development of the DFT from the analytic FT.

$$\widetilde{G}(f_x) = [G(f_x) \otimes \mathrm{comb}(\delta f_x) \otimes L\ \mathrm{sinc}(L f_x)] \times \mathrm{comb}(L f_x). \qquad (2.18)$$

The impact of sampling the spatial-frequency domain is shown in Fig. 2.6(g). The result is virtual periodic replication in the spatial domain. The term 'virtual' is used because there are actually no samples in the periodically replicated region.

Figures 2.6(i) and (j) show the final DFT pair. Plot (i) shows only the samples from the spatial domain that input to the DFT algorithm, and Plot (j) shows the output from the `ft` function.

To provide a clarification, the reader should note that one effect has not been discussed yet. Figure 2.6(h) shows a frequency function that still has an infinite number of samples. One would logically expect that we should go a step further and account for the finite number of samples with multiplication by a $\mathrm{rect}$ function in the frequency domain. This would imply that the spatial-domain function is rippled and broadened by convolution with a $\mathrm{sinc}$ function. However, we are considering a forward FT so that we start with the black samples shown in Fig. 2.6(g), which begin undistorted by any such convolution. Now, if we were to consider a discrete IFT, we could simply treat plots (a), (c), (e), (g), and (i) as the frequency-domain function. The IFT differs from the forward FT by only a sign in the exponential, which does not affect these distortions. Consequently, if we start with an undistorted frequency-domain function and perform a discrete IFT, the spatial-domain function would be periodically replicated, rippled, and sampled like in plots (b), (d), (f), (h), and (j).

## 2.4 Alleviating Effects of Discretization

When we want to use a DFT to approximate a continuous FT $G(f_x)$ of a known function $g(x)$, the FT pair that is actually used is $\widetilde{g}(x)$ and $\widetilde{G}(f_x)$ as given by Eqs. (2.17) and (2.18). The result $\widetilde{G}(f_x)$ of the DFT is a sampled, rippled, and aliased version of the desired analytic result. These effects may be reduced, but usually not eliminated. The rippling may be reduced by increasing the spatial grid size $L$, and the aliasing may be reduced by decreasing the spatial grid spacing $\delta$.

Figures 2.7, 2.8, and 2.9 illustrate the results of various attempts to limit rippling and aliasing (as compared to Fig. 2.6). In producing Fig. 2.7, a larger grid has been used by increasing $\delta$ while keeping $N$ the same. As a result, the factor $L\,\mathrm{sinc}(L f_x)$ became narrower, thereby reducing the rippling. This can be seen by comparing Fig. 2.7(f) to Fig. 2.6(f). Unfortunately, increasing $\delta$ means that the factor $\mathrm{comb}(\delta f_x)$ now has a narrower spacing, leading to increased aliasing, which is visible in Fig. 2.7(d). Conversely, in producing Fig. 2.8, more samples have been used so that $N$ has increased, $\delta$ has decreased, and $L$ remains the same. This approach reduces the aliasing by spreading out the $\mathrm{comb}(\delta f_x)$ factor, but without improving the rippling. The reduced aliasing is evident in Fig. 2.8(d), and the unchanged rippling is visible in Fig. 2.8(f). Finally, in learning a lesson from Figs. 2.7 and 2.8, smaller $\delta$ and larger $L$ were used in producing Fig. 2.9. This approach

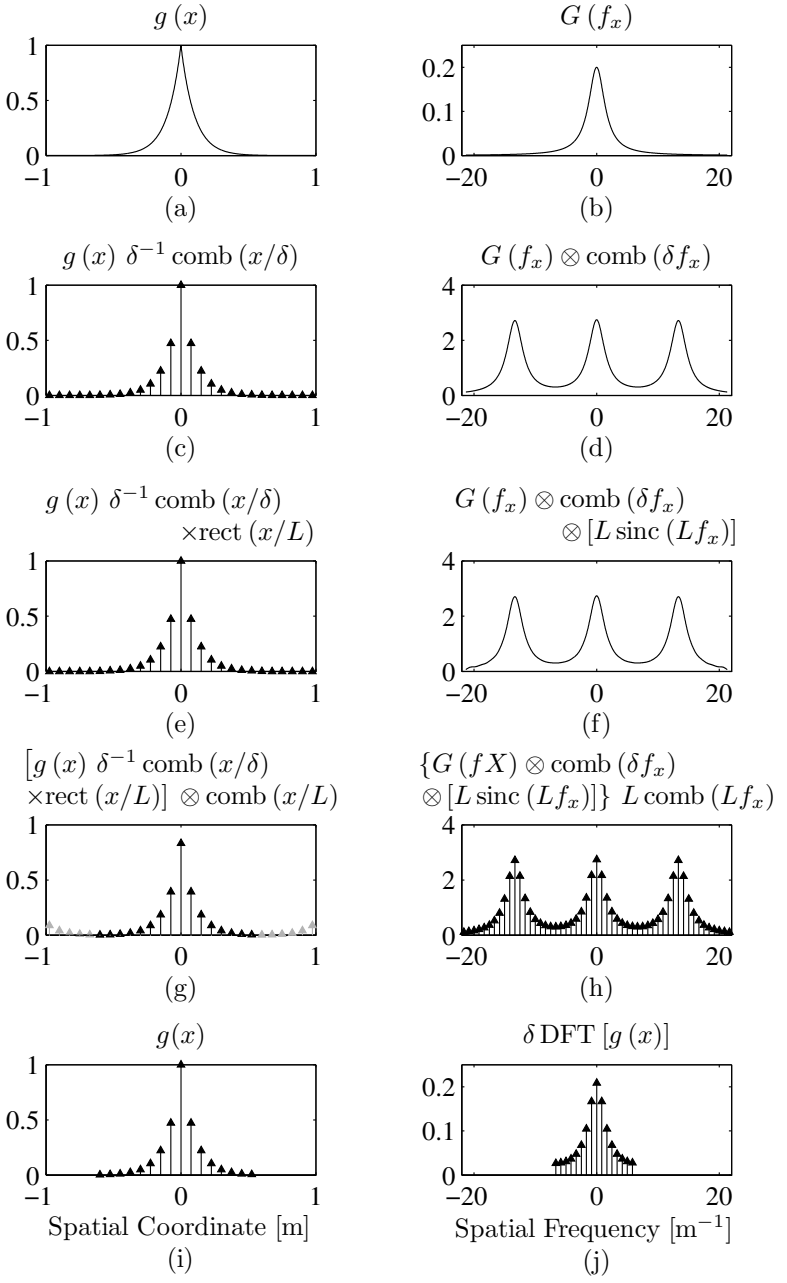


**Figure 2.7** Same as Fig. 2.6, but with a larger grid.

reduces aliasing and rippling at the same time, which is clearly the best approach. The drawbacks are the additional memory and computations required.

Unlike the graphical example above, some functions are strictly bandlimited. This means that the function $g(x)$ that we want to transform has a maximum fre-

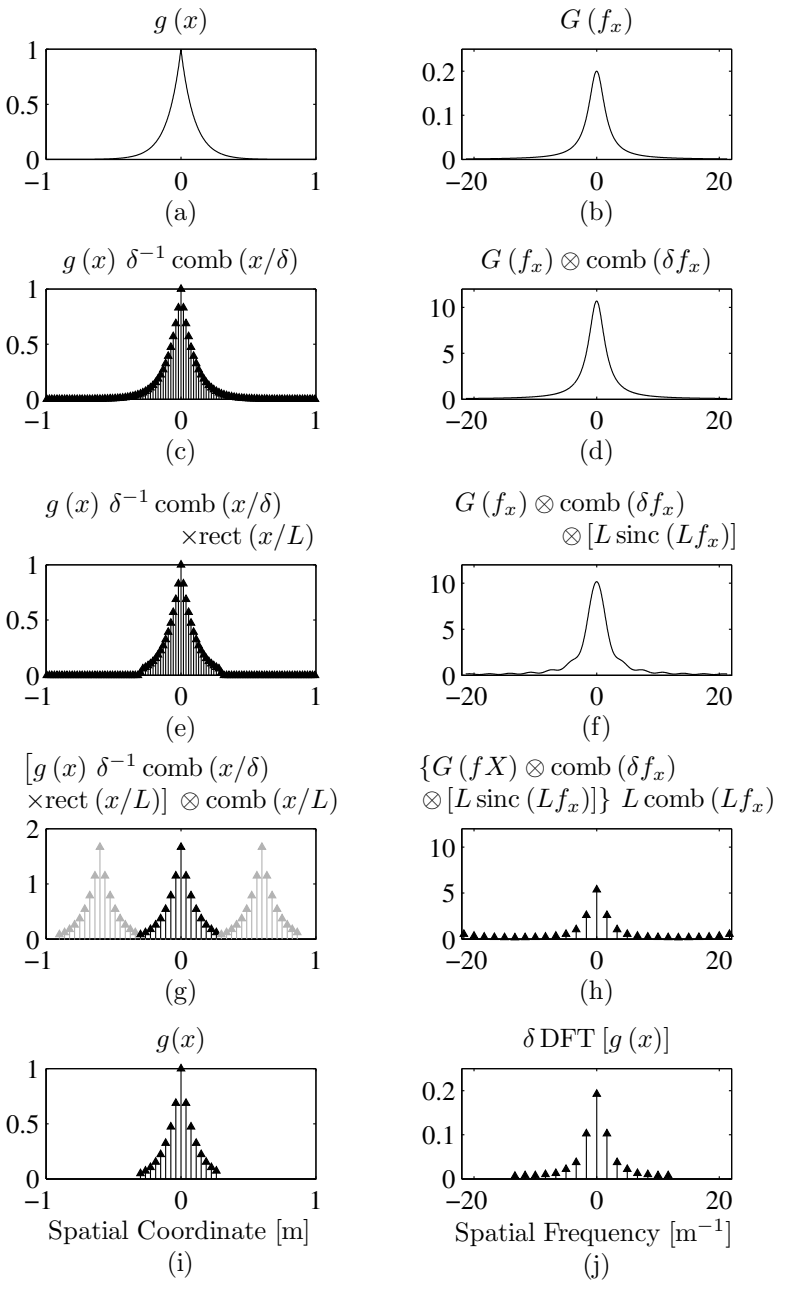


**Figure 2.8** Same as Fig. 2.6, but with more samples.

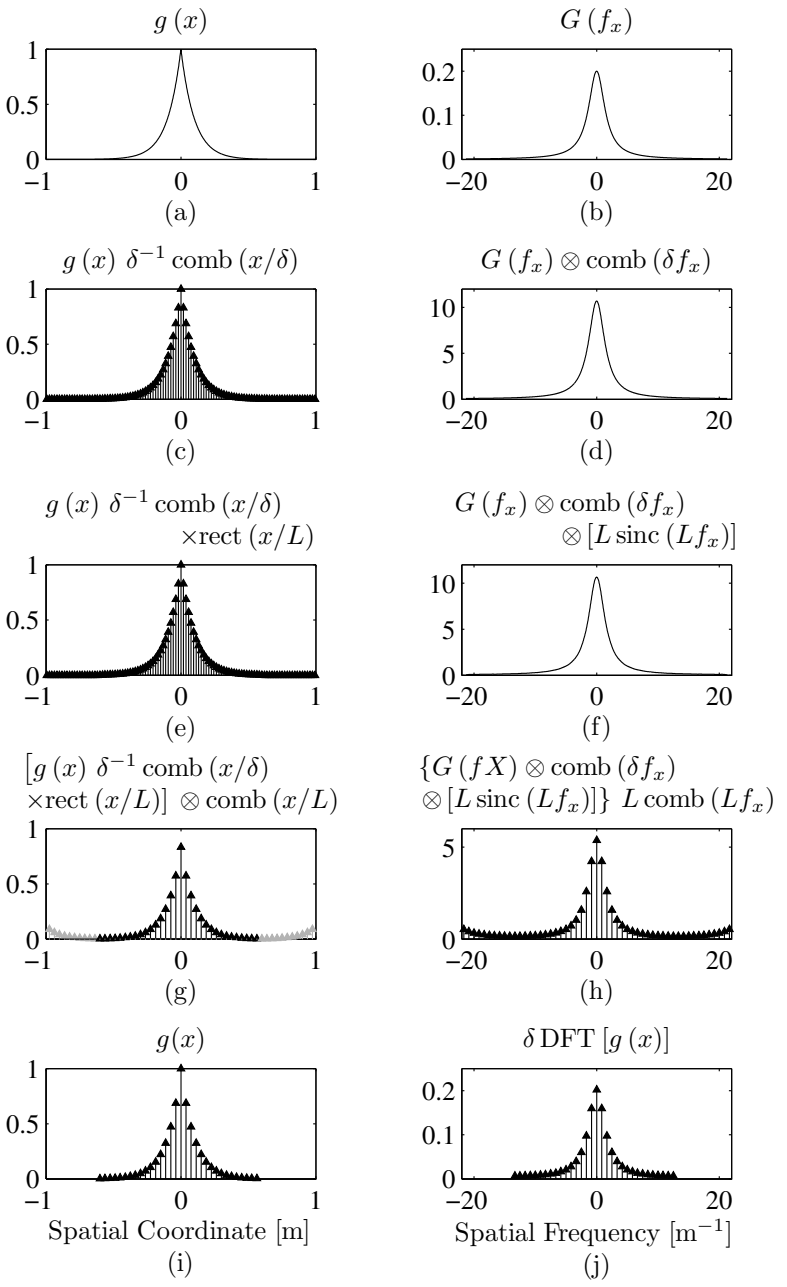


**Figure 2.9** Same as Fig. 2.6, but with more samples and larger grid.

quency $f_{x,max}$ such that

$$G(f_x) = 0 \text{ for } |f_x| > f_{x,max} \tag{2.19}$$

for some finite spatial frequency $f_{x,max}$. This frequency is called the bandwidth of $g(x)$. As discussed in Sec. 2.2, if we sample this continuous function so that there are two samples for every cycle of the highest frequency component, the continuous function can be reconstructed exactly from its spectrum. This requirement on the grid spacing can be expressed as

$$\delta \leq \frac{1}{2f_{x,max}}. \tag{2.20}$$

This is a very important consideration in the chapters covering Fresnel diffraction. Ch. 7 discusses this in detail.

Like the graphical example, sometimes signals are not strictly bandlimited, but there is a limit to how much bandwidth the user cares about. If he is simulating a system that can only sample at a rate of $f_s$, then the sampling requirement can be relaxed to

$$\delta \leq \frac{1}{f_s + f_{x,max}}. \tag{2.21}$$

This way, aliasing is present but not in the frequency range that the user cares about. The aliased frequencies wrap around from one edge of the grid to the edge of the other side, only distorting the spectrum at the highest frequencies.

## 2.5 Three Case Studies in Transforming Signals

In optics, we apply the FT to many types of signals with different types of band limits. This section highlights three different signals and how to compute their DFTs accurately. Computing the spectra of these deterministic signals provides important lessons for later when we want to compute the spectra of unknown and sometimes random signals. The three signals are a $\mathrm{sinc}$, a Gaussian, and a Gaussian $\times$ a quadratic phase. The first of these cases has a "hard" band limit like in Appendix A, while the latter two have "soft" band limits. Each case highlights different sampling considerations that become very important in later chapters.

### 2.5.1 Sinc signals

The $\mathrm{sinc}$ signal used in this book is defined in Appendix A. It is a good example of a signal that is intrinsically bandlimited such that its FT values are identically zero beyond a certain maximum frequency. It has a simple analytic FT given by

$$G(f_x) = \frac{1}{a}\,\mathrm{rect}\left(\frac{f_x}{a}\right). \tag{2.22}$$

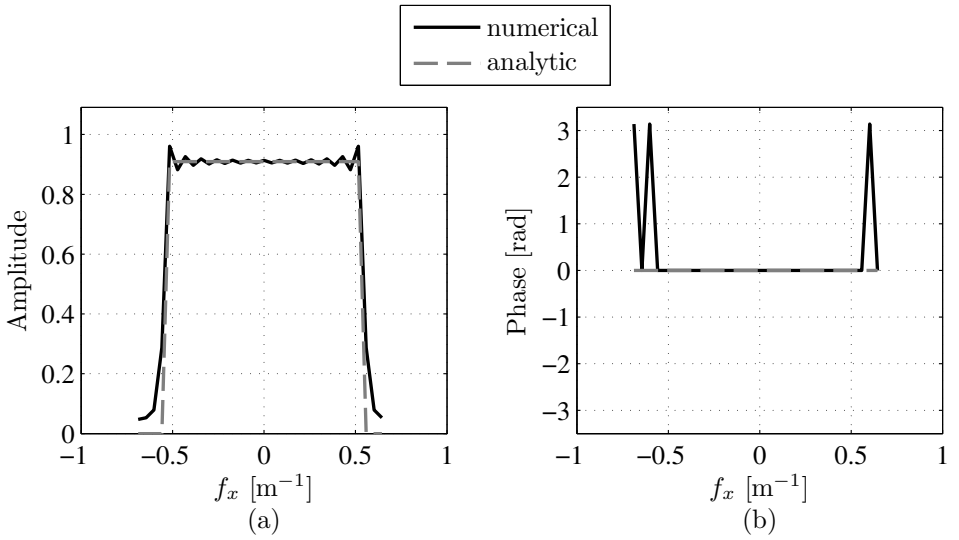


**Figure 2.10** Amplitude and phase of the DFT of a sinc signal. The grid spacing was determined by applying the Nyquist criterion.

Because we know the analytic FT of this signal, we know its maximum frequency before computing the DFT. We can then apply the Nyquist criterion to properly sample it before computing the DFT. The maximum frequency in Eq. (2.22) is $a/2$. Applying the Nyquist criterion, we get $\delta \leq 1/(2\,a/2) = 1/a$. We can try computing the DFT of a sinc signal just below (so that the frequency grid is a little broader than the spectrum) this maximum grid spacing to demonstrate how well it works.

Figure 2.10 shows the DFT of a sinc signal with $a = 1.1$. The solid black line shows the result when the grid spacing is $\delta = 0.85/a$ and $N = 32$. A slight ripple is visible in the amplitude of the DFT shown in plot (a). This is because the spatial grid has not captured the entire spatial extent of the signal. Using more samples (with fixed grid spacing) reduces this ripple. In plot (b), the phase of the DFT at the edge of the frequency grid appears to jump between the correct value, zero, and an incorrect value, $\pi$. This is because the DFT values are not exactly zero, which they should be at the edge. They are slightly negative, which is the same as saying that the phase of those points is $\pi$ radians.

### 2.5.2 Gaussian signals

The Gaussian signal used in this book is defined by

$$g(x) = \exp\left[-\pi\,(ax)^2\right]. \tag{2.23}$$

This form of the Gaussian appears in common Fourier-optics textbooks, like Goodman.[5] The Gaussian is a good example of a signal that is very nearly bandlimited,

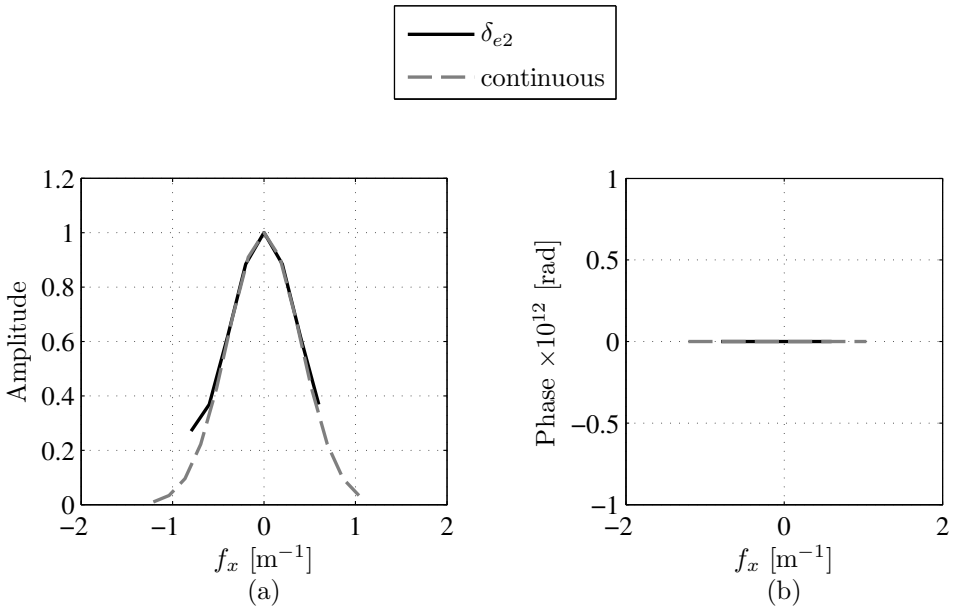


**Figure 2.11** Amplitude and phase of the DFT of a Gaussian signal. The grid spacing was determined by applying the Nyquist criterion to the $1/e^2$ frequency.

and it frequently appears in optics because laser beams often have a Gaussian amplitude profile. It has a simple analytic FT given by

$$G(f_x) = \frac{1}{|a|} \exp\left[-\pi (f_x/a)^2\right], \tag{2.24}$$

and its $1/e^2$ frequency is obviously $f_{e2} = a(2/\pi)^{1/2}$. Note that this definition of maximum frequency was arbitrary; we could always choose another definition depending on the situation.

Because we know the analytic FT of this signal, we know its maximum frequency before computing the DFT. We can then apply the Nyquist criterion to properly sample it in advance. Using the $1/e^2$ frequency as $f_{x,max}$, the corresponding maximum grid spacing is

$$\delta_{e2} = \frac{1}{2a}\sqrt{\frac{\pi}{2}}. \tag{2.25}$$

We can try computing the DFT of a Gaussian signal at this maximum grid spacing to see how well it works.

Figure 2.11 shows the DFT of a Gaussian signal with $a = 1$. The solid line shows the result when the grid spacing is $\delta_{e2}$. Aliasing is visible in the left-most sample because a little bit of the spectrum from the right side of the plot, not captured by the samples, wrapped around to the left side. Perhaps the $1/e^2$ is not quite enough to get an accurate DFT.

The value of $1/e^2$ is approximately 0.135; let us try the value $p$ instead, where $p$ has a smaller value, like 0.01. Setting the spectrum equal to $p\times$ its peak value

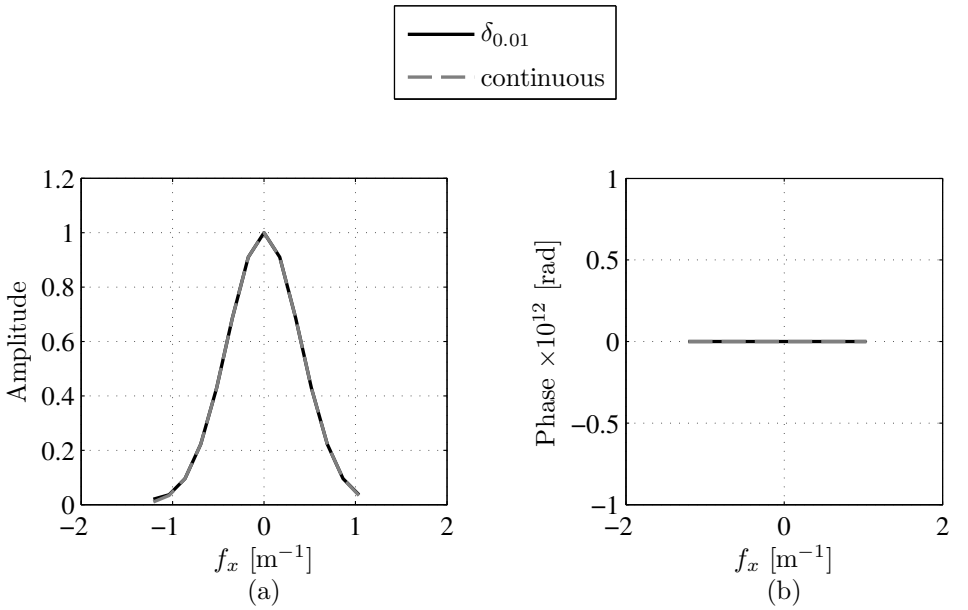


**Figure 2.12** Amplitude and phase of the DFT of a Gaussian signal. The grid spacing was determined by applying the Nyquist criterion to the 0.01 frequency.

allows us to solve for the frequency $f_{x,p}$ at this value:

$$p = \exp\left[-\pi \left(f_{x,p}/a\right)^2\right] \tag{2.26}$$

$$f_{x,p} = \left[-\left(\frac{a^2}{\pi}\right) \ln p\right]^{1/2}. \tag{2.27}$$

For example, $f_{x,0.01} = 2.1\, a/\pi^{1/2}$, and $f_{x,0.001} = 2.6\, a/\pi^{1/2}$. Figure 2.12 shows the result of using this grid spacing corresponding to $f_{x,0.01}$ as the maximum frequency. Aliasing is not visible in the amplitude plot because the portion of the spectrum that wraps around has a very small value ($0.01\times$ the peak value).

### 2.5.3 Gaussian signals with quadratic phase

In this case, we add a quadratic phase factor to the Gaussian signal. The Gaussian signal with quadratic phase is defined by

$$g\left(x\right) = \exp\left[-\pi \left(ax\right)^2\right] \exp\left[i\pi \left(bx\right)^2\right]. \tag{2.28}$$

This sort of signal arises in the propagation of Gaussian-beam waves. It is mathematically the most general and complicated of the three signals covered in these case studies. Figure 2.13 shows the real and imaginary parts of this signal for the case when $a = 0.25$ and $b = 0.57$. The quadratic phase causes it to oscillate rapidly as $|x|$ increases. The Gaussian amplitude, however, attenuates the oscillations so

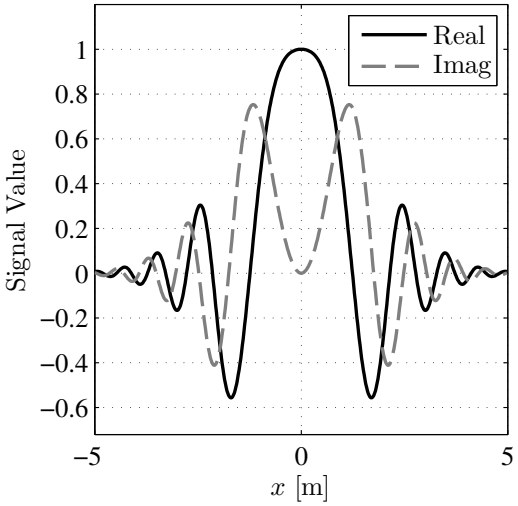


**Figure 2.13** Real and imaginary parts of a Gaussian signal with a quadratic phase.

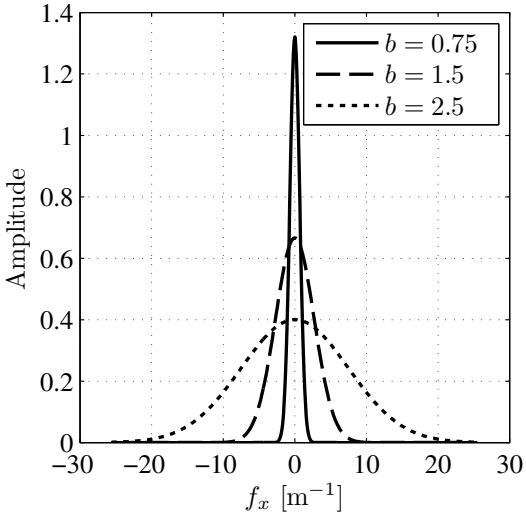


**Figure 2.14** Spectral amplitude of a Gaussian signal with a quadratic phase. Clearly, increasing the value of $b$ increases the bandwidth of the signal.

that the signal is in fact nearly bandlimited. To sample this function sufficiently for computing a DFT, we first need to determine the bandwidth of the spectrum. The signal has an analytic FT given by

$$G\left(f_x\right) = \frac{1}{\sqrt{a^2 - ib^2}} \exp\left(-\pi \frac{f_x^2}{a^2 - ib^2}\right). \tag{2.29}$$

Figure 2.14 shows the impact of the curvature parameter $b$ on the width of the spectrum. The plot shows the case of $a = 0.33$ with three different values of $b$,

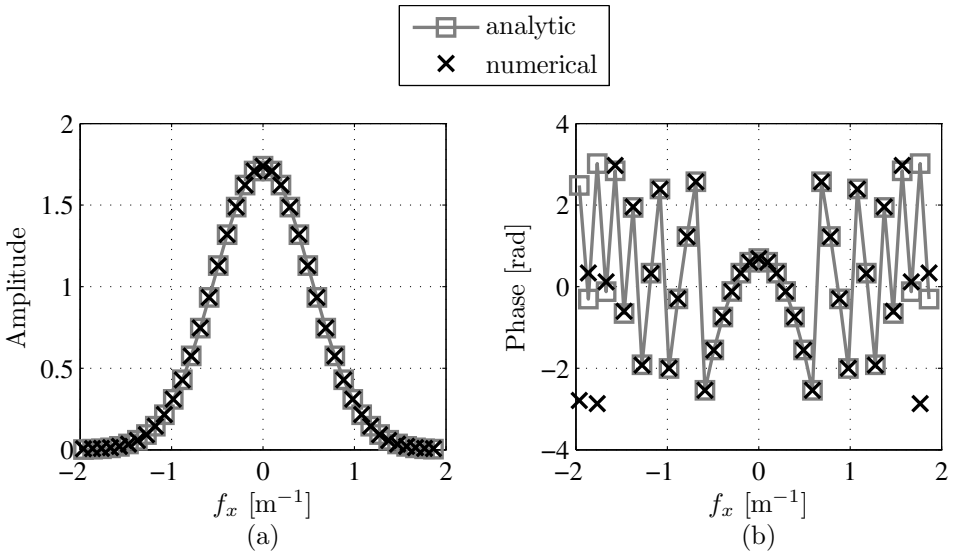


**Figure 2.15** DFT of a Gaussian signal with a quadratic phase. The frequency corresponding to $p = 0.01$ was used to compute the grid spacing.

using 0.75, 1.5, and 2.5. The three lines clearly demonstrate that as $b$ increases, so does the width of the spectrum. In fact, we can compute bandwidth from its amplitude using

$$p = \exp\left[-\pi \mathrm{Re}\left(\frac{f_{x,p}^2}{a^2 - ib^2}\right)\right]. \tag{2.30}$$

The result is

$$f_{x,p} = \left[-\left(\frac{a^2 + b^4/a^2}{\pi}\right)\ln p\right]^{1/2}. \tag{2.31}$$

Of course, Eq. (2.31) analytically confirms that $f_{x,p}$ increases with $b$. Also, note that Eqs. (2.26) and (2.27) are the $b = 0$ cases of Eqs. (2.30) and (2.31).

Figure 2.15 shows the analytic FT and DFT of a Gaussian signal with a quadratic phase. The signal has $a = 0.25$ and $b = 0.57$. It was sampled with grid spacing corresponding to the $p = 0.01$ frequency. Therefore, $f_{x,0.01} = 1.96$ m$^{-1}$ and $\delta = 1/(2f_{x,0.01}) = 0.25$ m, and only 40 samples are required. In the figure, the amplitude clearly matches well, but the DFT phase is slightly inaccurate at the edge of the spatial-frequency grid. If we were simulating a system that could sample no faster than about 1.7 m$^{-1}$, this would be all right. However, if we needed accuracy at higher spatial frequencies, we might need to do the simulation again with $p = 0.001$.

## 2.6 Two-Dimensional Discrete Fourier Transforms

We live in a four-dimensional universe (as far as we know) with three spatial dimensions plus time. Optics deals with waves traveling along one spatial dimension, and

**Listing 2.5** Code for performing a two-dimensional DFT in MATLAB.

```
function G = ft2(g, delta)
% function G = ft2(g, delta)
    G = fftshift(fft2(fftshift(g))) * delta^2;
```

we typically leave off the time dependence. That leaves us working with a function of two spatial dimensions in a plane transverse to the propagation direction. As a result, two-dimensional FTs are used frequently in optics.[8,13] In fact, they are central to the remainder of this book.

To begin studying two-dimensional FTs, we reuse the results of the previous sections with some modifications. We must rewrite Eqs. (2.1) and (2.2), generalizing to two dimensions, as

$$G(f_x, f_y) = \mathcal{F}\{g(x,y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\, e^{-i2\pi(f_x x + f_y y)}\, dx\, dy \tag{2.32}$$

$$g(x,y) = \mathcal{F}^{-1}\{G(f_x, f_y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(f_x, f_y)\, e^{i2\pi(f_x x + f_y y)}\, df_x\, df_y. \tag{2.33}$$

Then, we make the following changes to Eqs. (2.15)–(2.18):

$$g(x) \Rightarrow g(x,y) \tag{2.34}$$

$$G(f_x) \Rightarrow G(f_x, f_y) \tag{2.35}$$

$$\text{rect}\left(\frac{x}{a}\right) \Rightarrow \text{rect}\left(\frac{x}{a}\right)\text{rect}\left(\frac{y}{b}\right) \tag{2.36}$$

$$a\ \text{sinc}(af_x) \Rightarrow ab\ \text{sinc}(af_x)\,\text{sinc}(bf_y) \tag{2.37}$$

$$a\ \text{comb}(af_x) \Rightarrow ab\ \text{comb}(af_x)\,\text{comb}(bf_y) \tag{2.38}$$

This leads to (assuming same number of grid points, sample size, and spacing in $x$ and $y$ dimensions):

$$\begin{aligned}\widetilde{g}(x,y) = &\left[g(x,y)\,\frac{1}{\delta^2}\,\text{comb}\left(\frac{x}{\delta}\right)\text{comb}\left(\frac{y}{\delta}\right)\text{rect}\left(\frac{x}{L}\right)\text{rect}\left(\frac{y}{L}\right)\right] \\ &\otimes\left[\frac{1}{L^2}\,\text{comb}\left(\frac{x}{L}\right)\text{comb}\left(\frac{y}{L}\right)\right]\end{aligned} \tag{2.39}$$

$$\begin{aligned}\widetilde{G}(f_x, f_y) = &\left[G(f_x, f_y) \otimes \text{comb}(\delta f_x)\,\text{comb}(\delta f_y) \otimes L^2\ \text{sinc}(Lf_x)\,\text{sinc}(Lf_y)\right] \\ &\times \text{comb}(Lf_x)\,\text{comb}(Lf_y).\end{aligned} \tag{2.40}$$

Listings 2.5–2.6 give MATLAB code for the functions `ft2` and `ift2`, which perform two-dimensional DFTs and DIFTs, respectively. These functions are used frequently throughout the remainder of the book. They are central to two-dimensional convolution, correlation, structure functions, and wave propagation.

**Listing 2.6** Code for performing a two-dimensional DIFT in MATLAB.

```
function g = ift2(G, delta_f)
% function g = ift2(G, delta_f)
    N = size(G, 1);
    g = ifftshift(ifft2(ifftshift(G))) * (N * delta_f)^2;
```

## 2.7 Problems

1. Perform a DFT of $\mathrm{sinc}\,(ax)$ with $a = 1$ and $a = 10$. Plot the results along with the corresponding analytic Fourier transforms.

2. Perform a DFT of $\exp\left(-\pi a^2 x^2\right)$ with $a = 1$ and $a = 10$. Plot the results along with the corresponding analytic Fourier transforms.

3. Perform a DFT of $\exp\left(-\pi a^2 x^2 + i\pi b^2 x^2\right)$ with $a = 1$ and $b = 2$. Plot the results along with the corresponding analytic Fourier transforms.

4. Perform a DFT of $\mathrm{tri}\,(ax)$ with $a = 1$ and $a = 10$. Plot the results along with the corresponding analytic Fourier transforms.

5. Perform a DFT of $\exp\left(-a\left|x\right|\right)$ with $a = 1$ and $a = 10$. Plot the results along with the corresponding analytic Fourier transforms.