simplify to the familiar Fourier series (see Section 9.3):

$$\Phi_m = \sum_{n=-N/2}^{N/2} \Psi_n e^{-ip_m x_n/\hbar} \tag{16.42}$$

$$\Psi_n = \frac{1}{N} \sum_{m=-N/2}^{N/2} \Phi_m e^{ip_m x_n/\hbar}, \tag{16.43}$$

where $\Phi_m = \Phi(p_m)$ and $\Psi_n = \Psi(x_n)$. We have not explicitly shown the time dependence in (16.42) and (16.43).

We now use the `FFTApp` class introduced in Section 9.3 to transform a wave function between position and momentum space. Note that the wavenumber $2\pi/\lambda$ (or $2\pi/T$ in the time domain) in classical physics has the same numerical value as momentum in quantum mechanics $p = h/\lambda = 2\pi\hbar/\lambda$ in units such that $\hbar = 1$. Consequently, we can use the `getWrappedOmega` and `getNaturalOmega` methods in the `FFT` class to generate arrays containing momentum values for a transformed position space wave function.

The `FFTApp` class in Listing 9.7 transforms N complex data points using an input array that has length $2N$. The real part of the jth data point is stored in array element $2j$ and the imaginary part is stored in element $2j+1$. The `FFT` class transforms this array and maintains the same ordering of real and imaginary parts. However, the momenta (wavenumbers) are in wrap-around order starting with the zero momentum coefficients in the first two elements and switching to negative momenta halfway through the array. The `toNaturalOrder` class sorts the array in order of increasing momentum. We use the `FFTApp` class in Problem 16.19.

Problem 16.19 Transforming to momentum space

(a) The `FFTApp` class initializes the wave function grid using the following complex exponential:

$$\Psi_n = \Psi(n\Delta x) = e^{in\Delta x} = \cos n\Delta x + i \sin n\Delta x. \tag{16.44}$$

Use `FFTApp` to show that a complex exponential has a definite momentum if the grid contains an integer number of wavelengths. In other words, show that there is only one nonzero Fourier component.

(b) How small a wavelength (or how large a momentum) can be modeled if the spatial grid has N points and extends over a distance L?

(c) Where do the maximum, zero, and minimum values of the momentum occur in wrap-around order? ■

After the transformation, the momentum space wave function is stored in an array. The array elements can be assigned a momentum value using the de Broglie relation $p = h/\lambda$. The longest wavelength that can exist on the grid is equal to the grid dimension $L = (N-1)\Delta x$, and this wave has a momentum of

$$p_0 = \frac{h}{L}. \tag{16.45}$$

Points on the momentum grid have momentum values with integer multiples of p_0.

Problem 16.20 Momentum visualization

Add a `ComplexPlotFrame` to the `FFTApp` program to show the momentum space wave function of a position space Gaussian wave packet. Add a user interface to control the width of the Gaussian wave packet and verify the Heisenberg uncertainty relation $\Delta x \Delta p \geq \hbar/2$. Shift the center of the position space wave packet and explain the change in the resulting momentum space wave function. ■

Problem 16.21 Momentum time evolution

Modify `TDHalfStepApp` so that it displays the momentum space wave function in addition to the position space wave function. Describe the momentum space evolution of a Gaussian packet for the infinite square well and a simple harmonic oscillator potential. What evidence of classical-like behavior do you observe? ■

The FFT can be used to implement a fast and accurate method for solving Schrödinger's equation. We start by writing (16.4) in operator notation as

$$i\hbar \frac{\partial \Psi(x,t)}{\partial t} = \hat{H}\Psi(x,t) = (\hat{T} + \hat{V})\Psi(x,t), \tag{16.46}$$

where $\hat{H}$, $\hat{T}$, and $\hat{V}$ are the Hamiltonian, kinetic energy, and potential energy operators, respectively. The formal solution to (16.46) is

$$\Psi(x,t) = e^{-i\hat{H}(t-t_0)/\hbar}\Psi(x,t_0) = e^{-i(\hat{T}+\hat{V})(t-t_0)/\hbar}\Psi(x,t_0). \tag{16.47}$$

The time evolution operator $\hat{U}$ is defined as

$$\hat{U} = e^{-i\hat{H}(t-t_0)/\hbar} = e^{-i(\hat{T}+\hat{V})(t-t_0)/\hbar}. \tag{16.48}$$

It might be tempting to express the time evolution operator as

$$\hat{U} = e^{-i\hat{T}\Delta t/\hbar} e^{-i\hat{V}\Delta t/\hbar}, \tag{16.49}$$

but (16.49) is valid only for $\Delta t \equiv t - t_0 \ll 1$, because $\hat{T}$ and $\hat{V}$ do not commute. A more accurate approximation (accurate to second order in Δt) is obtained by using the following symmetric decomposition:

$$\hat{U} = e^{-i\hat{V}\Delta t/2\hbar} e^{-i\hat{T}\Delta t/\hbar} e^{-i\hat{V}\Delta t/2\hbar}. \tag{16.50}$$

The key to using (16.50) to solve (16.46) is to use the position space wave function when applying $e^{-i\hat{V}\Delta t/2\hbar}$ and to use the momentum space wave function when applying $e^{-i\hat{T}\Delta t/2\hbar}$. In position space, the potential energy operator is equivalent to simply multiplying by the potential energy function. That is, the effect of the first and last terms in (16.50) is to multiply points on the position grid by a phase factor that is proportional to the potential energy:

$$\tilde{\Psi}_j = e^{-iV(x_j)\Delta t/2\hbar}\Psi_j. \tag{16.51}$$

Because the kinetic energy operator in position space involves partial derivatives, it is convenient to transform both the operator and the wave function to momentum space.