Fourier Lab Instructions
Physics 218: Waves and Thermodynamics

Fall 2003, James P. Sethna

Latest revision: September 23, 2003, 8:40

Definitions: Series, Transform, and FFT

The Fourier series for periodic functions of period L is

$$\tilde{y}_m = (1/L) \int_0^L y(x) \exp(-ik_m x) dx, \qquad (FS1)$$

where $k_m = 2\pi m/L$. The Fourier series can be resummed to retrieve the original function:

$$y(x) = \sum_{m=-\infty}^{\infty} \tilde{y}_m \exp(ik_m x). \qquad (FS2)$$

The Fourier transform for functions on the infinite interval

$$\tilde{y}(k) = \int_{-\infty}^{\infty} y(x) \exp(-ikx)\, dx \qquad (FT1)$$

where now k takes on all values. We regain the original function by doing the inverse Fourier transform.

$$y(x) = (1/2\pi) \int_{-\infty}^{\infty} \tilde{y}(k) \exp(ikx)\, dk \qquad (FT2),$$

The Fast Fourier transform starts with N equally spaced data points y_ℓ, and returns a new set of complex numbers $\tilde{y}_m^{FFT}$:

$$\tilde{y}_m^{FFT} = \sum_{\ell=0}^{N-1} y_\ell \exp(-i2\pi m\ell/N), \qquad (FFT1)$$

with $m = 0, \ldots N-1$. It's essentially sampling the function $y(x)$ at equally spaced points points $x_\ell = \ell L/N$ for $\ell = 0, \ldots N-1$.

$$\tilde{y}_m^{FFT} = \sum_{\ell=0}^{N-1} y_\ell \exp(-ik_m x_\ell), \qquad (FFT2)$$

Running the Program.

Run the program *Fourier*.

```
      python /home/sethna/bin/Fourier.py
```

If that doesn't work, we may fall back to the less efficient method

```
      mkdir "jps6" [Choose a directory name instead of jps6]
      cd "jps6"
      cp /home/sethna/bin/Fourier .
      ./Fourier
```

On the left, you see a cosine wave $A\cos(k(x-x_0))$, evaluated at 32 points from $x=0$ to 20 as described above, with $k=k_1=2\pi/L$, $A=1$, and $x_0=0$. On the right, you see a Fourier series of the cosine: black is the real part, red is the imaginary part. Play with the program, trying various combinations of the real-space, Fourier-space, and parameter options.[†]

I. Sinusoidal Waves and Fourier Series.

(a) Check your predictions from the pre-lab exercise I(a) for the Fourier series for $\cos(k_1 x)$ and $\sin(k_1 x)$. Check your predictions from I(b) for δk and for $k_{N/2}$.[*] You can find where the data points are by clicking *Symbols*; you can find $k_{N/2}$ by observing the range on the Fourier series plot (which should go from $k_{-N/2}$ to $k_{N/2-1}$ for even N).

Increase the number of wavelengths, keeping the number of data points fixed. Notice that the Fourier series looks fine, but that the real-space curves quickly begin to vary in amplitude, much like the patterns formed by beating (superimposing two waves of different frequencies). By increasing the number of data points, you can see that the beating effect is due to the small number of points we sample. Of course, even for large numbers of sampled points N, at very small wavelengths (when we get close to $k_{N/2}$) these effects will still happen. Click through various numbers of wavelengths m up to and past $m=N/2$.

(b) **Wavelength.** Check your prediction from I(c) for what y_ℓ looks like for $\cos(k_N x)$ and $\cos(k_{N/2}x)$. The Fourier series for the latter should have only one spike, at the edge of the plot.

Aliasing. If you sample a function at N points with Fourier components beyond $k_{N/2}$, their contributions get added to Fourier components at smaller wave-vectors. This is called *aliasing*, and is an important source of error in Fourier methods. We always strive to sample enough points to avoid it.

(c) Check your prediction from I(d) for the transforms of $\cos(kx)$ for $k > k_{N/2}$, by typing in larger values of m.

[†] You can always return to the initial configuration by quitting the program and starting again.

[*] You can get better accuracy on the plots by zooming in: select a region with your right mouse button. The left button zooms in; clicking without dragging restores the default. Warning: the range will stay zoomed until you click: if you change parameters and see no plot, try clicking in the window.

Now, restart the program (to get the default values), and shift to the function *Packet*. Change the width σ of the packet to make it thinner. Notice that when the packet begins to look ratty (as thin as the spacing between the sampled points x_ℓ) the Fourier series hits the edges and overlaps: high frequency components are "folded over" or *aliased* into the lower frequencies.

Make sure you understand the Fourier series for cosines and sines with various numbers of wavelengths: how x_0 changes cosines to sines, how the wave-vector k is related to L and the number of wavelengths, how the real and imaginary parts vary with the phase of the wave, and why the imaginary parts of the Fourier series for $\sin(kx)$ have the signs that they do.

One often needs to take Fourier series of functions which are not periodic in the interval. Set the number of data points N to 256 (powers of two are faster) and compare $m = 20$ with an "illegal" non-integer value $m = 20.5$.[‡] Notice that the Fourier series looks pretty complicated! Each of the two peaks has broadened into a whole staircase. Try looking at the power spectrum (which is proportional to $|\tilde{y}|^2$), and again compare $m = 20$ with $m = 20.5$.

This is a numerical problem known as *windowing*, and there are various schemes to minimize its effects as well.

II. Gaussian Packets and Fourier Transforms

Restart *Fourier* at the starting parameter choices, and select the function *Packet* and *Fourier Transform.* Set the number of wavelengths m to zero: the graph on the left now shows $G_0(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-x^2/2)$. Increase the number of data points to 256.

Change the width of the Gaussian in real space and see that the width in Fourier space gets narrow as the width in real space gets wider.

(a) **Widths.** Starting with the Gaussian with $\sigma = 1$, zoom in to measure the width of its Fourier transform at some convenient height. (The full width at half maximum, FWHM, is a sensible choice.) Change σ to 2 and to 0.1, and measure the widths, to verify that the Fourier space with goes inversely with the real width. (Did you set m to zero?)

(b) **Uncertainty Principle.** Compute the product $\Delta x \, \Delta k$ of the FWHM of the Gaussians in real and Fourier space.

This is related to the Heisenberg uncertainty principle, $\Delta x \, \Delta p \sim \hbar$, which you will learn about in quantum mechanics.

In general, if you translate a function sideways (changing x_0), the Fourier transform changes in a particularly simple way. As you showed explicitly for the Gaussian in problem (3.5), the Fourier transform of $y(x - x_0)$ is $\exp(-ikx_0)$ times the Fourier transform of the untranslated function $\tilde{y}(k)$.

[‡] Be sure to hit Enter after typing in your number: it doesn't register the change until you do so.

Hold down the button for x_0 and watch the Fourier components change. How does the power spectrum change? Try also multiplying the real-space function by a cosine by changing the number of wavelengths m away from zero again.

III. Fourier Transform of White Noise.

White light is a mixture of light of all frequencies. White noise is a mixture of sound of all frequencies, with constant average power per unit frequency. The hissing noise you hear on radio and TV between stations is approximately white noise: there are a lot more high frequencies than low ones, so it sounds high-pitched.

What kind of time signal would generate white noise? Select the function *White Noise*. On the left you see a series of random numbers: each $y_\ell = y(\ell L/N)$ is chosen independently as a random number.[**] Change the number of data points to, say, 1024.

The Fourier transform of the white noise looks amazingly similar to the original signal. It is different, however, in two important ways. First, it is complex: there is a real part (black) and an imaginary part (red). The second is for you to discover.

(a) Zoom in near $k = 0$ on the Fourier plot, and describe how the Fourier transforms of the noisy signal are different from random. In particular, what symmetry do the real and imaginary parts have? Can you show that this is true for any real function $y(x)$?

Now select the *Power Spectrum* for the right graph. The average power at a certain frequency in a time signal $f(t)$ is proportional to $|\tilde{f}(k)|^2$, which is what we plot on the right side. Check that the power is noisy, but on average is crudely independent of frequency. (You can check this best by varying the random number seed.) White noise is usually due to random, uncorrelated fluctuations in time.

IV. Fourier Transform Twice.

Try exploring what happens when you take the Fourier transform twice. Select *Random Function*, and *Double*.

(a) Notice that the double transform is again defined over the same range $(0, L)$ as was the original function $y(x)$. Can you describe how the two are related to one another? It may help to try different random number seeds, or to continuously change x_0. Evaluate the ratio of the maxima of the two functions. Can you figure out what the formula for this constant scale factor is?

Try to figure out why this occurs, by considering the similarities and differences between the formulas (FT1) and (FT2) of the Fourier and inverse Fourier transforms. This can also give a hint for the formula for the scale factor for part (a) above.

[**] We choose the numbers with probability given by the Gaussian distribution $G_0(y)$, but it would look about the same if we took numbers with a uniform probability in, say, the range $(-1, 1)$.