Lecture 7:

Recap

Last time we took another look at Fourier coefficients to verify that the coefficients that we assert should represent our data really do work, and in doing so, we defined the Kronecker delta, δ_{nm}. Then we talked about 2 (out of 3) really key concepts for Fourier transforms:

1. First derivatives in the time domain (or the space domain) can be represented in the frequency (or wavenumber) domain by simple multiplication (e.g. by $i\sigma$).

2. Convolution, which is what we do when we filter data, can be carried out in the frequency (or wavenumber) domain through simple multiplication. Thus the convolution becomes

$$f * g(\tau) = \int_{-\infty}^{\infty} f(t)(\tau - t)\, dt \tag{1}$$

   and its Fourier transform is:

$$\mathcal{F}(f)\mathcal{F}(g) = \int_{-\infty}^{\infty} f * g e^{-i\sigma t}\, dt \tag{2}$$

   We're going to need slightly more compact notation for Fourier transforms, so let's try capital letters:

$$F(\sigma)G(\sigma) = \int_{-\infty}^{\infty} f * g e^{-i\sigma t}\, dt \tag{3}$$

3. That leaves Parseval's theorem for us to consider next.

Before we move on to Parseval's theorem, we'll talk about the example spectra that we started discussing last time ....

Three great traits of the Fourier transform (part 2)

3. Parseval's theorem: Total variance in the time domain equals total variance in the frequency domain

Last time we talked about convolution:

$$y(t) = h * x = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)d\tau. \tag{4}$$

You could also convolve a signal with itself, or with other data. What happens if I convolve my data ($x(t)$) with the time reversal of itself ($x(-t)$)?

$$y(t) = \int_{-\infty}^{\infty} x(\tau)x(t + \tau)d\tau. \tag{5}$$

More conventionally we might write:

$$y(\Delta t) = \int_{-\infty}^{\infty} x(t)x(\Delta t + t)dt. \tag{6}$$

So we're looking at the data multiplied by itself for a time lag Δt. At zero lag, this is the variance, and as we vary Δt we're looking at the lagged covariance for different time lags.

We can use this to consider the connection between the variance and the squared Fourier transform. What if we Fourier transform the autocovariance? It will be analogous to the Fourier transform of the convolution that we just calculated:

$$x_1(t)x_2(t) = \frac{1}{2\pi}x_1(t)\int_{-\infty}^{\infty} X_2(\sigma)e^{i\sigma t}\,d\sigma \tag{7}$$

so

$$\begin{aligned}
\int_{-\infty}^{\infty} x_1(t)x_2(t)dt &= \int_{-\infty}^{\infty}\left[\frac{1}{2\pi}x_1(t)\int_{-\infty}^{\infty} X_2(\sigma)e^{i\sigma t}\,d\sigma\right]dt &(8)\\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} X_2(\sigma)\left[\int_{-\infty}^{\infty} x_1(t)e^{i\sigma t}dt\right]d\sigma &(9)\\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} X_1^*(\sigma)X_2(\sigma)\,d\sigma. &(10)
\end{aligned}$$

Here we end up with the complex conjugate of X_1, because the exponentail was $e^{i\sigma t}$ (i.e. the form for the inverse Fourier transform) instead of $e^{-i\sigma t}$ (the form for the forward Fourier transform). That could be changed to represent a reversal in sign of t, but that would give us the complex conjugate of the Fourier transform.

Put succinctly, if $x_1 = x_2$:

$$\int_{-\infty}^{\infty} x^2(t)dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(\sigma)|^2\,d\sigma \tag{11}$$

This is Parseval's relationship. The factor of 2π is a normalization that comes about because of our notation. If we had used a definiton of Fourier transform in terms if $e^{-i2\pi xf}$, then the inverse Fourier transform wouldn't have a factor of 2π and we wouldn't be puzzling over it here.

In thinking about the time domain vs the frequency domain, one thing to keep in mind is the distinction between integrating over all time (on the left in the above equation) and integrating over all space (on the right). This implies that we're going to need to keep track of our frequency inforamtion carefully. In essence the Fourier coefficients in X (e.g. $|a_m|^2$) do not have the same units as the time domain values in x^2, because x is integrated in time and $|a_m|$ is integrated in frequency. If the total integral of x^2 is equal to the total integral of $|X|^2$, then we're going to need to adjust by factors of $\delta\sigma$, and this will influence how we label our axes.

Computing spectra

Having gotten this far—we can Fourier transform, we know some of the properties of the Fourier transform, we've looked at Parseval's theorem—we need to stop beating around the bush and produce some spectra.

How do we take our data and produce a meaningful measure of the power per unit frequency? Here's a basic approach:

1. First, we know we're going to need to Fourier transform our data, and plot the squared amplitudes. We'll only need to analyze the first $N/2+1$ of the Fourier coefficients, and we'll look at the amplitudes of these values. (The second half of the Fourier coefficients are complex conjugates of the first half of the record and correspond to negative frequencies.) The frequencies corresponding to the first $N/2+1$ coefficients will run from 0 cycles per N points to $N/2$ cycles per N points. So a first step is to compute:

```
a=fft(data)
N=length(data);
amp=abs(a(1:N/2+1).^2; % for even N
amp=abs(a(1:(N+1)/2).^2; % for odd N
```

2. Second, since we're only taking half the record, we've thrown out half the energy in the original data (except at frequency 0), so we'll need to put that back in.

   ```
   amp(2:end-1)=2*amp(2:end-1); % for even N
   amp(2:end)=2*amp(2:end); % for odd N
   ```

3. Having gotten this far, we'd better check that Parseval's theorem is working for us. We'll need to see that the energy in the initial record matches the energy in the Fourier transform. If we don't worry about units, then in Matlab we just need to divide by N to make our normalization work.

   ```
   amp=amp/N;
   ```

   If we want to make this make sense for the temporal and spatial sampling of our observations we might need to scale our results to reflect real time and frequency units. (Fortunately multiplicative scalings won't influence the shape of our spectra, so we can always delay sorting out the details.)

4. We can plot what we have. Typically we use a loglog axis (or at least semilogy). We'll need to compute our frequencies thoughtfully. So for example if we have measurements every 361 seconds, then we might want to convert to cycles per hour by scaling the frequency appropriately.

   ```
   scale = 1/361 * 3600; % to convert from 1 cycle/2 pts to 0.5 cycles
                          % per 361 s to 0.5 cycles * 3600/361 /hr
   frequency=(0:N/2)/N * scale; % for N even
   frequency=(0:N/2-1)/N * scale; % for N odd
   loglog(frequency,amp)
   ```

5. Now, having gotten this far, we have a problem in that our results are way too noisy. We'll have a hard time distinguishing signal from peak. So clearly we're going to need more realizations. To do this, one common practice is chop our data into multiple segments. As a first step, we can just cut the data into M segments of N/M points each. For example:

   ```
   N=length(data); M=6; p=N/M;
   data=reshape(data,N/M,M); % this gives us an array with N/M points
             % per column and M columns
   b=fft(data);  % this computes the fft for each column
   amp_b=abs(b(1:p/2+1,:)).^2; amp_b(2:p/2,:)=2*amp_b(2:p/2,:)/p;
   loglog(frequency,mean(amp_b,2))
   ```

   Here you can debate whether you should be plotting the sum (to conserve energy) or the average (to represent a mean spectrum for a data set of length p. In most cases you'll want the average.

Now the critical question? How many degrees of freedom does this record have? Is this M to represent M segments? Maybe you can think of it that way, but by convention, we get one degree of freedom for the real part and one for the imaginary part, so 2 per segment. We'll need this to compute error bars, but let's start by noting that our error bars are not the same as the standard error of the mean. We're computing the sum of M squared quantities, and that's going to depend on something that looks like a χ^2 distribution.

When we compute spectra from segments, clearly there are tradeoffs: if I have N data points total, I can have lots of segments with few points in each segment, or few segments with more points per segment. The Nyquist frequency will be the same whatever I choose, since that's determined by the interval between observations. But the low-frequency limit will differ, as will the increment between frequencies (which is determined by the lowest resolved frequency). There's no rule for how to handle this, and your decisions will depend whether you want small uncertainties or high resolution in frequency space.