Lecture 10:

Reading: Bendat and Piersol, Ch. 11.5.1-11.5.2

Recap

We've now examined how to compute spectra and how to use the χ^2 distribution to put error bars on our spectra. We've also talked about detrending and a bit about windowing. And Matthew went over the basic issue of aliasing. Now, we'll need to fill in some details to really understand both aliasing and windowing.

Where does aliased energy go?

You left off with the example of a 12 hour signal, sampled at 9 hour increments. The Nyquist frequency was 20 cycles per 15 days or $f_{Ny} = 20f$, and the frequency of the 12-hour oscillation was 30 cycles per 15 days or $30f = f_{Ny}+10f$. This will alias into its counterpart $f_Ny-10f = 10f$ which in this case is 10 cycles per 15 days or 36 hours, as illustrated in Figure 1.

Just in case you want to replicate this, here's the code that I used:

```
time=0:9:15*24;
time_dense=0:15*24;
plot(time_dense,cos(time_dense*2*pi/12),time,cos(time*2*pi/12),...
     'o','LineWidth',2)
hold on
plot(time_dense,cos(time_dense*2*pi/36),'r','LineWidth',3)
legend('"True" 12-hour oscillation',...
    'Measurements at 9-hour increments',...
     'Inferred 36-hour oscillation')
set(gca,'FontSize',16)
xlabel('Time (hours)','FontSize',16)
ylabel('Amplitude','FontSize',16)
```

There's a simple rule for determining alias frequencies. Take the frequency of the true signal minus the Nyquist frequency, and subtract that from the Nyquist frequency. So if I measure at 1 Hz, I have a Nyquist frequency of 1 cycle per 2 seconds. If I want to know what happens to a signal with a frequency of 1 cycle/1.1 seconds, I can figure it out this way. For the frequency f_{true} = 1 cycle/1.1 seconds = 10 cycles/11 seconds, with a Nyquist frequency of 5.5 cycles/11 seconds, then the difference will be (10-5.5)cycles/11 seconds = 4.5 cycles/11 seconds, so the resulting alias will be f_{alias} = (5.5-4.5)cycles/11 seconds = 1 cycle/11 seconds. (Ignore what I said in class, which was based on the wrong formulation....) To test this out:

```
t_time=0:.05:200;
test_time2=0:200;
signal=sin(2*pi*test_time/1.1);
signal2=sin(2*pi*test_time2/1.1);
plot(test_time,signal,test_time2,signal2)
```

So now you can test yourself. What are the alias frequencies for the following:

1. A 12-hour oscillation is sampled every 10 hours?

2. A 12-hour oscillation is sampled every 15 hours?

3. A 12.4 hour oscillation is sampled every 9.9 days?

The first of these cases is straightforward, but the second two cases are tricky because the aliased frequencies fold back and forth across resolved range of frequencies. Here's a basic algorithm.

1. Compute the Nyquist frequency and the frequency of the signal that might be aliased.
2. Compute the ratio: f_{signal}/f_{Ny} and truncate this to be a whole integer M less than or equal to the ratio.
3. If M is odd, then we're folding down relative to the Nyquist frequency. $\Delta = f_{signal} - M f_{Ny}$, and $f_{alias} = f_{Ny} - \Delta$.
4. If M is even, then we're folding up relative to frequency zero. $\Delta = f_{signal} - M f_{Ny}$, and $f_{alias} = \Delta$.

By this rule, the 12-hour oscillation sampled at 15-hour intervals aliases to 60 hours.

Using aliasing strategically

Finally, let's look at a very real world example. The TOPEX/Poseidon/Jason altimeters pass over the same ground tracks every 9.9156 days. Sea surface height is strongly influenced by the M2 tide, which has a period of 12.4206 hours. What frequency does 12.4206 hours alias into? In this case, the altimeter Nyquist frequency is nowhere near the tidal frequency, and the aliased signal folds back and forth along the x-axis several times. To compute this, we first need to compute frequencies in common units:

```
f_sampling=1/(9.9156); % cycles per day
f_M2=24/12.4206;       % cycles per day
f_Nyquist=f_sampling/2;
M=floor(f_M2/f_Nyquist) % compute the integer ratio of the two frequencies.
                        % In examples above, this was one.
alias = f_M2 - floor(f_M2/f_Nyquist)*f_Nyquist;
1/alias

%  Note: if M is odd then reset
if(rem(M,2)!=0) alias=f_Nyquist-alias; end
```

This calculation shows that M is 38 and the tidal energy aliases into a 62.1068 day period. Note that if M is odd, you have to do the calculation slightly differently. As an exercise, you can check what happens if you round of the numbers and use 9.9 for the altimeter sampling frequency and 12.4 for the tidal cycle.

This is a nice illustration that if you understand how aliasing works, and what the dominant energy might be, you can use the aliasing strategically to infer a signal from sparsely sampled data.

Although the M2 tide (and internal tide) from the TOPEX/Poseidon/Jason altimeter series are one of the clearest examples, this comes up in other cases too, particularly to extract diurnal and tidal cycles from more sparsely sampled data.

What does aliasing mean for noise at the high frequency end of the spectrum?

Finally, think about this scenario. Suppose we have a spectrum that is basically red, like most geophysical spectra, but at high frequencies or wavenumber, it is white. Unfortunately we don't fully sample the high frequencies. What happens to the energy beyond the Nyquist frequency?

This unsampled high frequency energy will alias in to our signal. (Imagine folding it over on our resolved spectrum.

Oversampling The term oversampling came up in class. It's used in a couple different ways (see Wikipedia). One meaning is just the concept of sampling at a much higher rate than the Nyquist frequency. That's obviously a good way to reduce aliasing and improve peak resolution. The second meaning is used for digitial-to-analog conversion and involves resampling information at a higher frequency than the Nyquist frequency of the original digital signal. When converted to analog, this oversampling process introduces some aliased or reflected signals that are subsequently filtered out.

Some further convolution examples

Let's return to convolution for one more example. We've pointed out that convolution in the time domain is equivalent to multiplication in the frequency domain.

What if I compute the convolution of a boxcar filter with itself?

$$y(\tau) = \int_{-\infty}^{\infty} h(t)h(\tau - t)dt \quad \text{... \textit{definition of convolution}} \tag{1}$$

$$= \int_0^1 h(\tau - t)dt \quad \text{for } 0 < \tau - t < 1 \quad \text{... \textit{adjust integration limits to represent} } h(t) \tag{2}$$

$$= \int_0^1 h(\tau - t)dt \quad \text{for } -\tau < -t < 1 - \tau \quad \text{... \textit{subtract} } \tau \tag{3}$$

$$= \int_0^1 dt \quad \text{for } \tau - 1 < t < \tau \quad \text{... \textit{reverse sign from} } -t \text{ \textit{to} } +t \tag{4}$$

$$= \begin{cases} \int_0^\tau dt = \tau, & \text{for } 0 < \tau \leq 1 \\ \int_{\tau-1}^1 dt = 1 - (\tau - 1) = 2 - \tau & \text{for } 1 < \tau \leq 2 \end{cases} \tag{5}$$

The final step requires some examination of the limits of integration. We require that t be between 0 and 1, but τ can have a broader range. We can impose limits on t or on τ, somewhat separately. First, we worry only about the upper limit. If $t < \tau$, and $0 < t \leq 1$, then we can consider a case when $0 < \tau \leq 1$, while imposing the lower limit of the integral. This gives us limits of integration from 0 to τ for $0 < \tau \leq 1$. If we consider only the lower limit, so that $\tau - 1 < t$ with $0 < t \leq 1$ then τ can be between 1 and 2, with integration limits from $\tau - 1$ to 1. In any case, the results produce a triangle.

This is definitely easier to consider graphically, and it will show that boxcar filter convolved with itself becomes a triangle filter.

What if we repeat this again and again? It's actually analogous to what we considered for the central limit theorem. Applying a filter to a filter to a filter will eventually give us a Gaussian.

The Fourier Transform of a Boxcar

When we deal with data records of finite length, we're always taking finite sized segments of data, sort of like Fourier transforming a boxcar filter. Usually we think of a Fourier transform of this form:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi f t}\, dt. \tag{6}$$

If $x(t)$ is a boxcar filter, this becomes:

$$\int_{-\infty}^{\infty} h(t) e^{-i2\pi f t}\, dt = \int_{-1/2}^{1/2} e^{-i2\pi f t}\, dt = \left.\frac{e^{-i2\pi f t}}{-i2\pi f}\right|_{-1/2}^{1/2} = \frac{e^{i2\pi f/2} - e^{-i2\pi f/2}}{i2\pi f} = \frac{\sin(\pi f)}{\pi f} \tag{7}$$

This is the sinc function, which can be written as:

$$\text{sinc}\,(x) = \frac{\sin(x)}{x} \tag{8}$$

or (for digital signal processing)

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \tag{9}$$

We'll try to use the second definition, the "normalized sinc function", since we're trying to use a frequency f that needs to be multiplied by 2π.

Remember that convolution in the time domain corresponds to multiplication in the frequency domain. In this case, we're doing the opposite. We multiplied our data by a boxcar window in the time domain, and that's equivalent to convolving with a sinc function in the frequency domain:

$$\hat{X}(f) = \int_{-\infty}^{\infty} X(g) W(f-g)\, dg \tag{10}$$

where

$$W(f) = \frac{\sin(2\pi f T)}{2\pi f T} = \text{sinc}\,(2fT)\,, \tag{11}$$

for a record length of 2T.

Corresponding, the spectrum is convolved with $W(f)^2 = \text{sinc}(2fT)^2$. This has a central peak of width $\Delta f = \pm 1/(2T)$. The window will have a maximum value

$$|W(f)|^2 < (2\pi f T)^{-2} \tag{12}$$

The side lobes of the window are definitely problematic. And it turns out that we're not stuck with them. If we can widen the central peak of W (the Fourier transform of our window), we can lower the impact of the side lobes. To do this, we'll want to forego the rectangular window in the time domain and choose something that lets us attenuate the beginnings and ends of each segment of our data. What if we chose a triangle window? That will already give us fewer side lobes.

But we can keep going to find a window that looks more like an exponential. Leading possibilities:

1. Cosine taper:

$$w(t) = \cos^{\alpha}\left(\frac{\pi t}{2T}\right) \tag{13}$$

   with $\alpha = [1, 4]$.

2. Hanning window or "raised cosine" window (developed by von Hann):

$$w(t) = \cos^2\left(\frac{\pi t}{2T}\right) = \frac{1+\cos(\pi t/T)}{2} = 0.5 + 0.5\cos(\pi t/T) \tag{14}$$

3. Hamming window. This variant of the Hanning window was developed by Hamming.

$$w(t) = 0.54 + 0.46\cos(\pi t/T) \tag{15}$$

The Hamming window has less energy in the first side lobe but more in the distant side lobes.

Some other options include a Blackman-Harris window or a Kasier-Bessel window, and Harris (1978, Use of windows for harmonic analysis, *Proc. IEEE*) provides detailed discussion of options.

So how do you use a window?

1. First you must demean your data—otherwise, the window will shift energy from the mean into other frequencies. If you're working in segments, you should demean (and detrend) each segment before you do anything further.

2. Second, for a segment with N points, multiply by a window that is N points wide.

3. Since the window attentuates the impact of the edge of each segment, you can use segments that overlap (typically by 50%). This will give you (almost) twice as many segments, so instead of ν degrees some larger number.

4. Now Fourier transform, scale appropriately (e.g. by $\sqrt{8/3}$ for a Hanning window, to account for energy attenuation) and compute amplitudes.

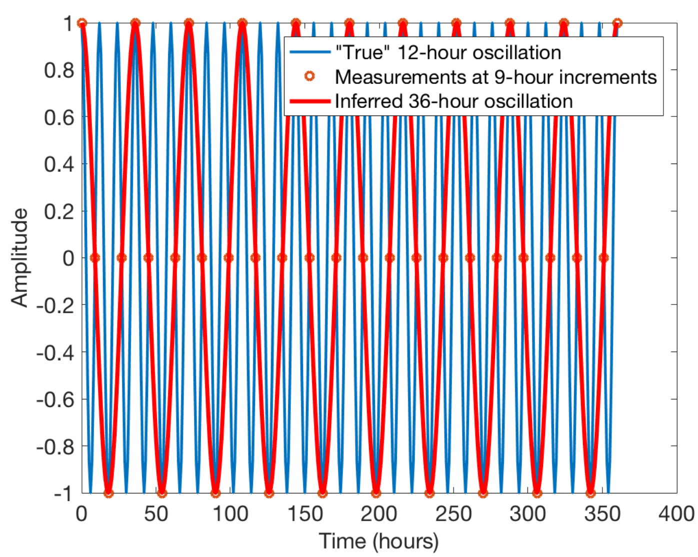

Figure 1: Example of 12-hour signal aliased by 9-hour sampling.