Lecture 18:

Recap

Last time we looked at the transfer function, which we framed as the Fourier transformed form of a regression coefficient. Today we'll pick up a few final details, looking at zero padding and multi-taper methods, and hopefully rotary spectra.

Multi-tapers and spectral peaks

Spectra can come in two flavors. Some have distinct single peaks, associated with tides. Some have large-scale structure associated with the general red structure of the ocean. If we want to find exactly the right peaks, then we can try different strategies to what we use when we want to find the general structure.

When we have narrow peaks, they aren't always easy to differentiate, particularly if our sampling is a bit coarse compared with the signals we'd like to detect. Consider the following case of a sinusoidal cycle that might or might not be well sampled, depending how long our instruments survive:

```
time=1:.5:120;
A=2*cos(2*pi*time/30)+cos(2*pi*time/60);
B=A(1:200);
C=[A(1:200) zeros(1,40)];

plot(time,A,time(1:200),B,'LineWidth',3)
set(gca,'FontSize',16)
xlabel('time','FontSize',16)
ylabel('amplitude','FontSize',16)

fA=fft(A);
fB=fft(B);
fC=fft(C);

frequency1=(0:120)/120;
frequency2=(0:100)/100;

loglog(frequency1,abs(fA(1:121)).^2,frequency2,abs(fB(1:101)).^2,...
  frequency1,abs(fC(1:121)).^2,'LineWidth',3)
set(gca,'FontSize',16)
xlabel('frequency','FontSize',16)
ylabel('spectral density','FontSize',16)
legend('full record','truncated record','zero padding')
```

When you look at this example, you might conclude that without perfect sampling of full sinusoidal cycles, we'll never find the correct spectral peaks. In essence this is a windowing problem. When we have narrow peaks, they aren't always easy to differentiate, particularly if our sampling is a bit coarse compared with the signals we'd like to detect. See the slides for an example (from Rob Pinkel.)

If we don't have adequate resolution what are our options?

1. Possibility 1. Pad the short record with zeros to make it as long as we want. Since resolution is $f = 1/(N\Delta t)$. In this case, we'll see the impact of a sinc function bleeding into the frequencies that we'd like to resolve. Clearly this doesn't fully solve our problem.

2. Possibility 2. Obtain a longer record. This will be critical if we really want to resolve our signal.

Even if our record is norminally long enough, we also need to figure out how to optimize our detection of spectral peaks. Earlier we looked at the impact of windows, and examples from the Harris (1978) study showed how much impact a good windowing strategy can have in identifying spectral peaks. (For continuous spectra, windowing approaches work well.)

Formally, you'll recall that we can represent our record length problems using a convolution of our data with a finite width filter:

$$\hat{X}(f_n) = \int_{-\infty}^{\infty} X(f_m) W(f_n - f_m)\, df_m, \tag{1}$$

where

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

This means that the spectrum is essentially convolved with $W(2\pi fT)^2$. But as we noted earlier, we can switch from a boxcar window to a triangle window or something a bit more Gaussian than that and cut down on the sidelobes in our window to obtain a cleaner spectrum, although we have to widen the central peak of the window in the frequency domain, which means de-emphaizing the beginning and end of the data series.

What if we want to improve our resolution. Consider Rob Pinkel's example of a record equivalent to

$$x = 100\cos(2\pi 20.5/10001) + 80\cos(2\pi 30.4/100001) + 100\cos(2\pi 40.8/100001) + 10\cos(2\pi 50.3/100001) \tag{3}$$

The quality of our spectral estimate will depend on the length of our record. (Why is that? The resolution is the lowest resolved frequency.) So what can we do to improve resolution. One strategy would be to pad our record with zeros to make it as long we want. That buys us something, but it gives us plenty of spectral ringing.

If we want to optimize resolution, we can try a multitaper approach. (See for example Ghil et al, Reviews of Geophysics, 2001). In a multitaper approach, we replace our single window with a set of tapers. The tapers are designed to minimize spectral leakage, and they are referred to as "discrete prolate spheroidal sequences" or "Slepian" tapers (after Slepian, who studied them). Tapers are what we've been calling windows—they pre-multiply the data, Fourier transforms are computed, and then the spectrum is computed as a weighted sum of all of the squared Fourier transforms. This effectively averages over an ensemble of windows to minimize variance. This is very effective for extracting narrow peaks that would otherwise be undetectable. Matlab has a multi-taper method package ('pmtm'), but if you really want this to work, you probably want to dig into the guts of the algorithm a bit further.

Here's the Matlab example, modified slightly to make a longer record:

```
fs = 1000;                  % Sampling frequency
t = (0:3*fs)/fs;              % One second worth of samples
A = [1 2];                  % Sinusoid amplitudes
```

```
f = [150;140];              % Sinusoid frequencies

xn = A*sin(2*pi*f*t) + 0.1*randn(size(t));
pmtm(xn,4,[],fs)
```

This produces an impressive two spectral peaks. Of course this example isn't too tricky. Here's what we get if we take the same data and split them into 6 non-overlapping segments, even with no windowing or detrending:

```
fa=fft(reshape(xn(1:3000),500,6));
semilogy(mean(abs(fa(1:250,:)').^2))
```

These are reassuringly similar results.

Decibels and Powers of 10

Decibels ("db", not to be confused with decibars) quantify power or variance. We often focus on orders of magnitude. But power in decibels is on a $\log_{10}$ scale, with a factor of ten normalization.

$$P_{db} = 10\log_{10}(P/P_0), \tag{4}$$

where P_0 is a reference level of power: Variance is a squared quantity so

P/P_0	P relative to P_0 (db)
1000	30
10	10
2	3
0.5	-3
0.1	-10

$$P_{db} = 20\log_{10}(V/V_0). \tag{5}$$

By convention dB is used for sound pressure and db for everything else.

Coherence from auto-covariance

We talked last time about using the auto-covariance to compute phase. What happens when we put that into practice. Here's an example:

```
lambda=10; % 10 m wavelength
V=0.3; % 0.3 m/s propagation
n2s=0.2; % noise-to-signal ratio
time=(1:5000)';
x=n2s*randn(5000,1)+cos(2*pi/lambda*V*time);
y=n2s*randn(5000,1)+cos(2*pi/lambda*V*(time)+pi/2);


% start with a segmented data approach:
xx=[reshape(x,500,10) reshape(x(251:4750),500,9)];
yy=[reshape(y,500,10) reshape(y(251:4750),500,9)];
```

```
fxx=fft(xx.*(hanning(500)*ones(1,19)));
fyy=fft(yy.*(hanning(500)*ones(1,19)));
sxx=abs(fxx(1:251,:)).^2;
syy=abs(fyy(1:251,:)).^2;
sxy=conj(fxx(1:251,:)).*fyy(1:251,:);
coher1=abs(mean(sxy,2)).^2./(mean(sxx,2).*mean(syy,2));
phase1=atan2(-imag(mean(sxy,2)),real(mean(sxy,2)));
semilogx(0:250,coher1)



% now try the autocovariance approach
cxx=xcov(x,x,'unbiased');
cxy=xcov(x,y,'unbiased');
cyy=xcov(y,y,'unbiased');
figure(1)
plot(cxy)
shg

% first just use a boxcar window
N=5000;
halfwin=250;
win=halfwin*2+1;
fxx=fft(cxx(N-halfwin:N+halfwin));
fyy=fft(cyy(N-halfwin:N+halfwin));
fxy=fft(cxy(N-halfwin:N+halfwin));
coh=abs(fxy(1:halfwin)).^2 ./abs(fyy(1:halfwin)) ./abs(fxx(1:halfwin));
phase=atan2(-imag(fxy(1:halfwin)),real(fxy(1:halfwin)));
figure(2)
semilogx(0:249,phase,0:250,phase1)
shg
figure(3)
semilogx(0:249,coh,0:250,coher1);

% now consider a Hanning window
N=5000;
halfwin=250;
win=halfwin*2+1;
fxx=fft(cxx(N-halfwin:N+halfwin).*hanning(win));
fyy=fft(cyy(N-halfwin:N+halfwin).*hanning(win));
fxy=fft(cxy(N-halfwin:N+halfwin).*hanning(win));
coh_h=abs(fxy(1:halfwin)).^2 ./abs(fyy(1:halfwin)) ./abs(fxx(1:halfwin));
phase_h=atan2(-imag(fxy(1:halfwin)),real(fxy(1:halfwin)));
figure(2)
semilogx(0:249,phase,0:250,phase1,0:249,phase_h)
shg
figure(3)
```

```
semilogx(0:249,coh,0:250,coher1,0:249,coh_h);
shg
```

If you run this code, you'll see that the coherence doesn't produce the signal that we expect. Now if you go back and increase the noise-to-signal ratio:

```
n2s=0.8
```

you'll end up with a more centralized autocovariance function, and more plausible coherence estimates. This is the warning about using auto-covariance approaches for coherence. Although you can do this, you have to be extra careful that you're not merely dividing by zero, and that the structure of the central part of the autocovariance that you Fourier transform is well behaved.

Rotary spectra

One final detail. We've discussed spectra for scalar quantities: temperature, wind speed, or atmospheric pCO_2. But what happens when we think about a vector quantity such as velocity? Of course you can treat the u and v components of velocity as separate scalars, but that might fail to capture the complexity of the overall motion. Rotary spectra provide a way to examine vector motions as a combined quantity.

To start thinking about rotary spectra, let's think about a single frequency. Imagine that you're sitting on the beach on a summer day. In the afternoon, as the land warms, wind starts blowing onshore from the ocean. This is the classic sea breeze. The wind direction reverses at night, when wind blows from the cold land to the warm ocean. Our cartoon schematics suggest that this is just an onshore/offshore circulation, but of course we live on a rotating planet, so the sea breeze, like everything else, rotates with the Earth's rotation. If we had a full anemometer, we would be able to identify this pattern of circulation.

For the diurnal cycle, as for any wind frequency, I can write the wind as:

$$u(t) = a_1 \cos(\omega t) + b_1 \sin(\omega t) \tag{6}$$
$$v(t) = a_2 \cos(\omega t) + b_2 \sin(\omega t) \tag{7}$$

Or we can represent this as a complex number:

$$U(t) = u(t) + iv(t) \tag{8}$$
$$= a_1 \cos(\omega t) + b_1 \sin(\omega t) + a_2 \cos(\omega t) + b_2 \sin(\omega t) \tag{9}$$
$$= (a_1 + ia_2)\cos(\omega t) + (b_1 + ib_2)\sin(\omega t) \tag{10}$$

That gives us a vector motion in phase with the cosine and a vector motion in phase with the sine. But we're mixing complex amplitudes with real trigonometric functions in this form.

A different way to think of this is as a rotational components in the clockwise and counter-clockwise directions:

$$U(t) = U^+ e^{i\omega t} + U^- e^{-i\omega t} \tag{11}$$
$$= U^+ \left(\cos(\omega t) + i\sin(\omega t)\right) U^- \left(\cos(\omega t) - i\sin(\omega t)\right) \tag{12}$$
$$= (U^+ + U^-)\cos(\omega t) + (U^+ - U^-) i \sin(\omega t) \tag{13}$$

Here $e^{i\omega t}$ corresponds to counter-clockwise motion and $e^{-i\omega t}$ corresponds to clockwise motion. Since both expressions for U(t) have to be equivalent, this means that

$$U^+ = \frac{a_1 + b_2 + i(a_2 - b_1)}{2} \tag{14}$$
$$U^- = \frac{a_1 - b_2 + i(a_2 + b_1)}{2}. \tag{15}$$

The magnitude of these terms will give the rotary spectral components.

We can think of these terms as defining an ellipse with major axis $|U^+| + |U^-|$ and minor axis $||U^+| - |U^-||$. At each frequency, we can define three additional parameters. Two of these depend on angles, which we define as

$$\epsilon^+ = \tan^{-1}\left(\frac{a_2 - b_1}{a_1 + b_2}\right) \tag{16}$$

$$\epsilon^- = \tan^{-1}\left(\frac{a_2 + b_1}{a_1 - b_2}\right) \tag{17}$$

Then the orientation of the ellipse is:

$$\theta = \frac{\epsilon^+ + \epsilon^-}{2}, \tag{18}$$

and the phase of the ellipse (corresponding to the time when the velocity is at a maximum) is:

$$\phi = \frac{\epsilon^+ - \epsilon^-}{2}, \tag{19}$$

Finally, we can ask whether the motion is predominantly clockwise or counterclockwise by determining the sign of $|U^+| - |U^-|$.

While these parameters can be used to assess motions at a single frequency, more broadly they can be extracted from the Fourier transform to tell us about all frequencies. To do this, we just have to remember that $a_1 + ib_1$ are the Fourier coefficients for u and $a_2 + ib_2$ represent the Fourier coefficients for v, so we can extract the values that we need.

If we separately Fourier transform u and v, then the clockwise spectrum is the positive frequencies for the Fourier transform of u plus i times the negative frequencies for the Fourier transform of v. the counterclockwise spectrum is the opposite: negative frequencies for the Fourier transform of u plus i times the positive frequencies for the Fourier tranform of u.

It's useful to consider some limiting cases. Suppose that v is zero and u us proportional to $\cos(\omega t)$. Then only a_1 is non-zero, so U^+ and U^- are both $a_1/2$, the major axis is a_1. The angles ϵ^+ and ϵ^- are both 0, so the orientation angle $\theta = 0$ and the time of maximum $\phi = 0$, consistent with cosine being a maximum when $t = 0$. You can work through other cases to see how they express themselves in rotary form.

(For a good discussion of this, check our course notes by Miles Sundermeyer (U. Mass Dartmouth).)