Problem Set 7 Solutions: MAE 127

1. Compute frequency spectra for the Alpine daily minimum temperatures that you used in problem set 4. Divide the year-long record into 12 30-day long segments (ignoring the last 6 days of the year), so that you can compute spectra with error bars. Compute your spectrum with and without a Hanning window. (For a Hanning window to make sense, you'll need to subtract the mean from the data. You can do this separately for each segment.) Show your method, and plot your spectra with error bars. How would you describe these spectra?

To compute spectra, we'll first load the data, divide it into segments, and compute the fft. The spectrum is the squared fft.

```
load temperature.mat

% first we reshape the data as a 30 by 12 array and compute the fft
% of each column
%  ah has no overlap
%  ah2 uses a 50% overlap between segments
data=reshape(alpine_min(1:360),30,12);
data=data-ones(30,1)*mean(data);
a=fft(data);
ah=fft(data.*(hanning(30)*ones(1,12)));
data2=[reshape(alpine_min(1:360),30,12) reshape(alpine_min(16:345),30,11)];
data2=data2-ones(30,1)*mean(data2);
ah2=fft(data2 .*(hanning(30)*ones(1,23)));

% now we take half the record, find absolute values,
% multiply by 2, and normalize by N
% I also multiply the filtered spectra by (30/sum(hanning(30))^2 = (30/15.5)^2
% to make up for the energy lost in applying the Hanning filter.
% and I scale spectrum_ah2 by 30*23 instead of 30*12
spectrum_a=sum(abs(a(1:16,:)).^2,2)/360;
spectrum_a(2:end-1)=2*spectrum_a(2:end-1);
spectrum_ah=sum(abs(ah(1:16,:)).^2,2)/360*(30/15.5)^2;
spectrum_ah(2:end-1)=2*spectrum_ah(2:end-1);
spectrum_ah2=sum(abs(ah2(1:16,:)).^2,2)/(30*23)*(30/15.5)^2;
spectrum_ah2(2:end-1)=2*spectrum_ah2(2:end-1);

% compute the error bar
nu1=2*12;
err_low_1 = nu1/chi2inv(.05/2,nu1);
err_high_1 = nu1/chi2inv(1-.05/2,nu1);
nu2=2*23;
err_low_2 = nu2/chi2inv(.05/2,nu2);
err_high_2 = nu2/chi2inv(1-.05/2,nu2);
```

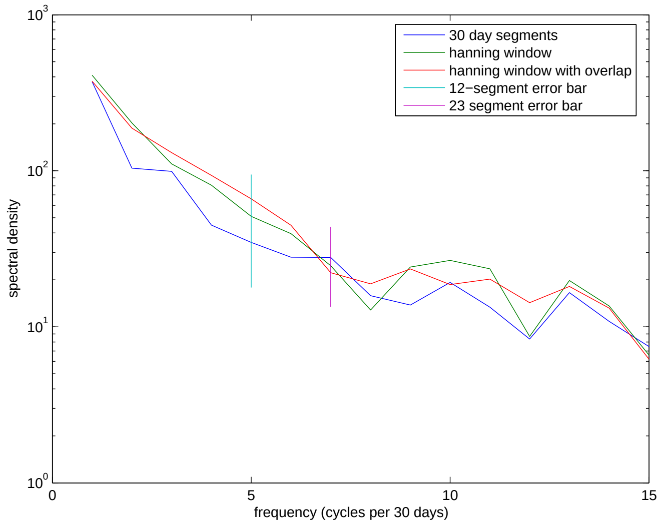

Figure 1: Spectra of Alpine minimum temperature record with and without Hanning window and overlapping segments.

```
% plot the results
semilogy(1:15,[spectrum_a(2:end) spectrum_ah(2:end) spectrum_ah2(2:end)],...
  [5 5],[err_low_1 err_high_1]*spectrum_a(6),[7 7],...
  [err_low_2 err_high_2]*spectrum_ah2(8))
xlabel('frequency (cycles per 30 days)'); ylabel('spectral density')
legend('30 day segments','hanning window','hanning window with overlap', ...
'12-segment error bar','23 segment error bar')
```

The resulting spectra, shown in Figure 1 are red. All three spectra agree within error bars, so in this case windowing is not wildly changing the interpretation of our records.

2. Using the Alpine minimum temperature data, verify Parseval's theorem (discussed in class on May 16.)

Parseval's theorem says that the summed energy of the original record is equal to the summed spectral energy. You can verify it by comparing the sum of the squared values in the orginal data with the sum of the squared Fourier components:

```
sum(alpine_min.^2)
sum(abs(fft(alpine_min)).^2)/366
```

Both produce 909,971 in this case.

Alternatively, you can work from the spectral segments:

```
mean(sum(data.^2))
sum(spectrum_a)
```

In this case, both produce 816.225.

3. Compute the coherence between the Alpine and San Miguel daily minimum temperature records. At what frequencies are the records most coherent? What are the statistical error bars for your coherence estimates? What is the phase of the coherence?

To compute coherence we'll need to segment both data sets and compute Fourier transforms. Here we do it using overlapping windowed data segments.

```
% segment data into overlapping pieces, demean, and apply hanning window
data=[reshape(alpine_min(1:360),30,12) reshape(alpine_min(16:345),30,11)];
data=data-ones(30,1)*mean(data);
a=fft(data .*(hanning(30)*ones(1,23)));

data2=[reshape(sanmiguel_min(1:360),30,12) reshape(sanmiguel_min(16:345),30,11)];
data2=data2-ones(30,1)*mean(data2);
b=fft(data2 .*(hanning(30)*ones(1,23)));

% compute spectra, and cross-spectra
spectrum_a=sum(abs(a(1:16,:)).^2,2)/(30*23)*(30/15.5)^2;
spectrum_a(2:end-1)=2*spectrum_a(2:end-1);
spectrum_b=sum(abs(b(1:16,:)).^2,2)/(30*23)*(30/15.5)^2;
spectrum_b(2:end-1)=2*spectrum_b(2:end-1);
C_ab= (sum(a(1:16,:).*conj(b(1:16,:)),2))/(30*23)*(30/15.5)^2;
C_ab(2:end)=2*C_ab(2:end);

% compute coherence and coherence phase
coher_ab=sqrt(abs(C_ab).^2 ./spectrum_a ./spectrum_b);
phase_ab=atan2(-imag(C_ab),real(C_ab));

% compute error estimates
alpha=0.05;
delta_coher = sqrt(1-alpha^(1/(23-1)));
delta_phase = asin((1-betainc(2*23/(2*23+alpha^2),23,.5))*...
         sqrt((1-coher_ab.^2)./(2*23*coher_ab.^2)));

subplot(2,1,1); plot(1:15,coher_ab(2:16),[1 15],[delta_coher delta_coher])
xlabel('frequency (cycles per 30 days)'); ylabel('coherence')

subplot(2,1,2); errorbar(1:15,phase_ab(2:16),delta_phase(2:16))
xlabel('frequency (cycles per 30 days)'); ylabel('phase (radians)')
```

The results, shown in Figure 2 indicate high coherence at low frequencies, but non-statistically significant coherence for frequencies greater than about 12 cycles per 30 days corresponding to periods less than 3 days. At frequencies that are statistically coherent, the phase is close to zero, implying that changes in Alpine more or less coincide with changes at San Miguel.

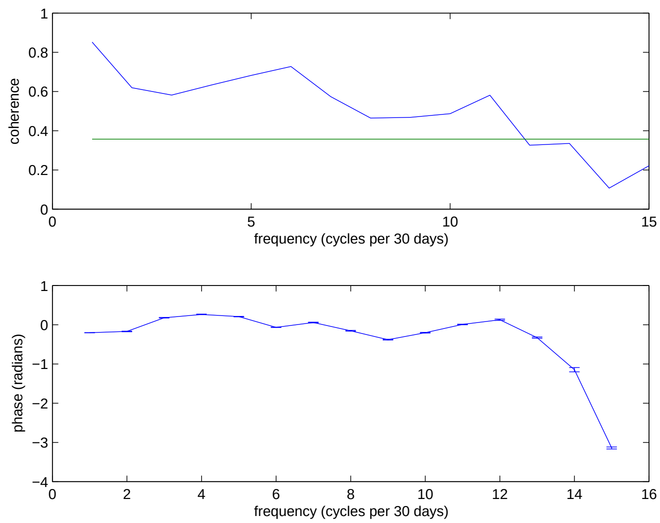

Figure 2: (top) Coherence of Alpine and San Miguel temperature minimum records (blue) with 95% significance level (green). (bottom) Phase lag between Alpine and San Miguel temperature minima. Results computed from 23 overlapping 30 day data segments.