SIO221A Homework #4 (Eric Gallimore)

Table of Contents

Setup

Homework 4- 2d Spectral Analysis vs Oct 2013 Go to folder SIO 221 on our server. Within folder “MBL-Surf”, load file velb4degl.mat. This is an array of ocean surface wave data. It is dimensioned 220 points in range (each 2m apart) by 3000 points in time (each 0.625 sec apart). The data were collected by a Doppler sonar whose beam was pointing in an east-west direction (range increasing, westward). The data are the east-west component of water velocity at the sea surface, in cm/s.

```
clear();
load('velb4degl.mat');
```

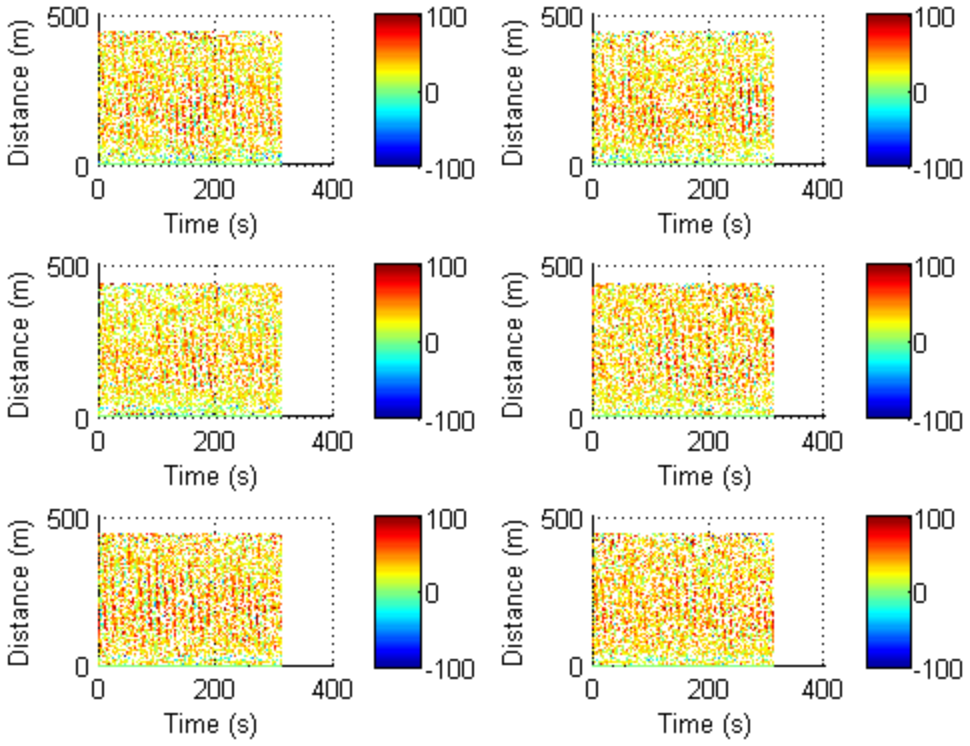
Task A - Initial Plotting

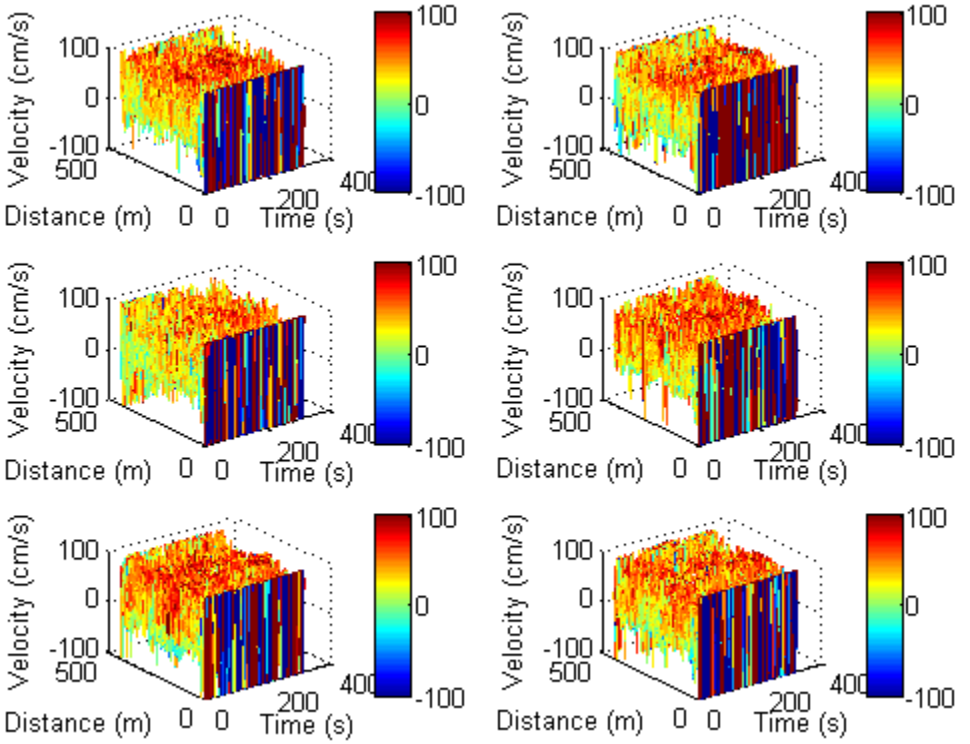
The tasks are to: A) form a 12 degree of freedom (dof) wavenumber-frequency spectral estimate of the motions B) form a 96 degree of freedom spectral estimate of the motions. C) calculate a frequency spectrum for the waves going east and a separate one for the waves going west. Do this by integrating over the appropriate quadrant of the wavenumber-frequency spectrum. Plot both spectra on the SAME plot, log-log format. Integrate each spectrum in frequency & state the variance. Please present the spectra in units of (m/s)2 / cpm / cps, plotted vs. frequency in cps and wavenumber in cpm. Is there more swell going east? Going west?

```
disp('More swell is going east (range numbers are increasing west, ');
disp('and waves are "approaching".  Based on this observation, positive');
disp('sea surface velocities also correspond to east');

% Are there more wind-waves going east? Going west? Steps: A Break the data
% into six-500 point records in time. Loop through the records plotting the
% data Use mat lab routines “mesh” or “surf”. View from above, “view
% (90,90)”, to see patterns
%records.velocity = zeros(0,0,6);
for i=0:5
    records.velocity(:,:,i+1) = velb4(:,i*500+1:i*500+500);
    records.time = 0:0.625:0.625*499; % times
    records.space = 0:2:2*219; % spacing
end

% Pick just good velocities
```

```
records.v_good = records.velocity(10:201,:,:);
records.space_good = records.space(10:201);

figure(1);
clf();
for i=1:6
    subplot(3,2,i);
    surf(records.time, records.space, records.velocity(:,:,i), 'EdgeColor','none')
    xlabel('Time (s)');
    ylabel('Distance (m)');
    zlabel('Velocity (cm/s)');
    colorbar();
    view(0,90);
end
figure(2);
clf();
for i=1:6
    subplot(3,2,i);
    mesh(records.time, records.space, records.velocity(:,:,i));
    xlabel('Time (s)');
    ylabel('Distance (m)');
    zlabel('Velocity (cm/s)');
    colorbar();
end
% _ Label axes in meters & seconds NOT “scans” &”bins” !! Print me one of
% your six records, demonstrating that you can see the signal. What’s the
% wavelength of the biggest waves? The period?

disp('The period of the biggest waves is approximately 15s (by visual ');
disp('inspection.  The wavelength is about 350m (again, by visual');
disp('inspection.');
```

More swell is going east (range numbers are increasing west, and waves are "approaching". Based on this observation, positive sea surface velocities also correspond to east
The period of the biggest waves is approximately 15s (by visual inspection. The wavelength is about 350m (again, by visual inspection.


Distance (m)
Time (s)
500
0
200
400
100
-100

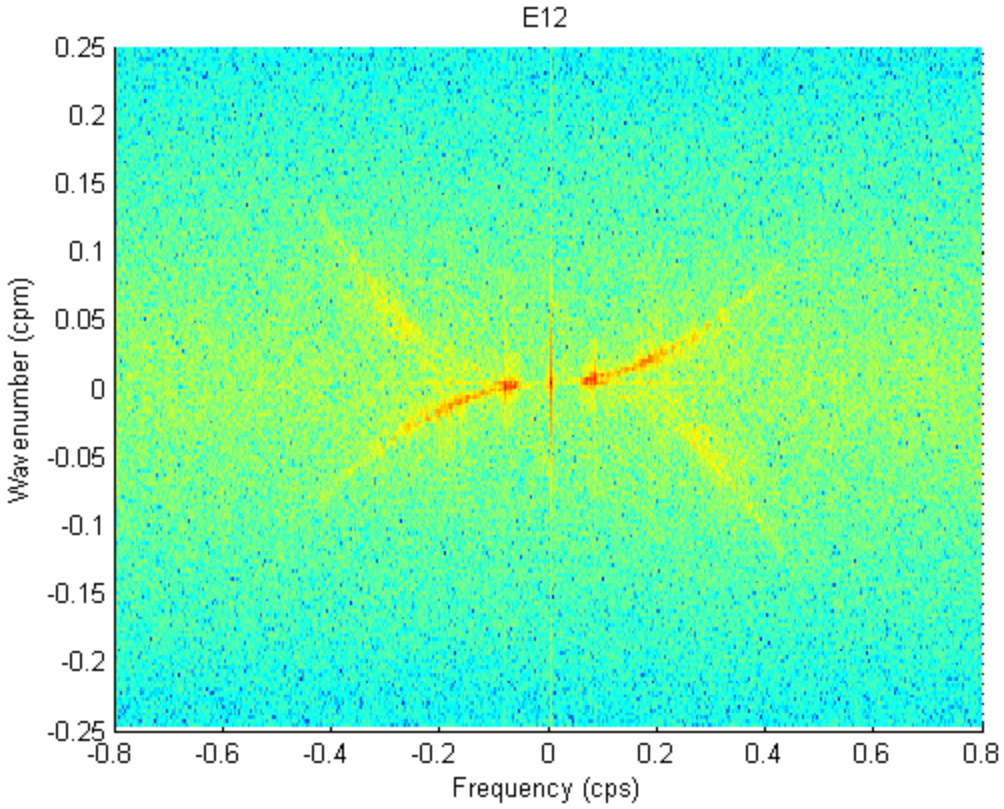

Velocity (cm/s)
Distance (m)
Time (s)
100
0
-100
500
200
400

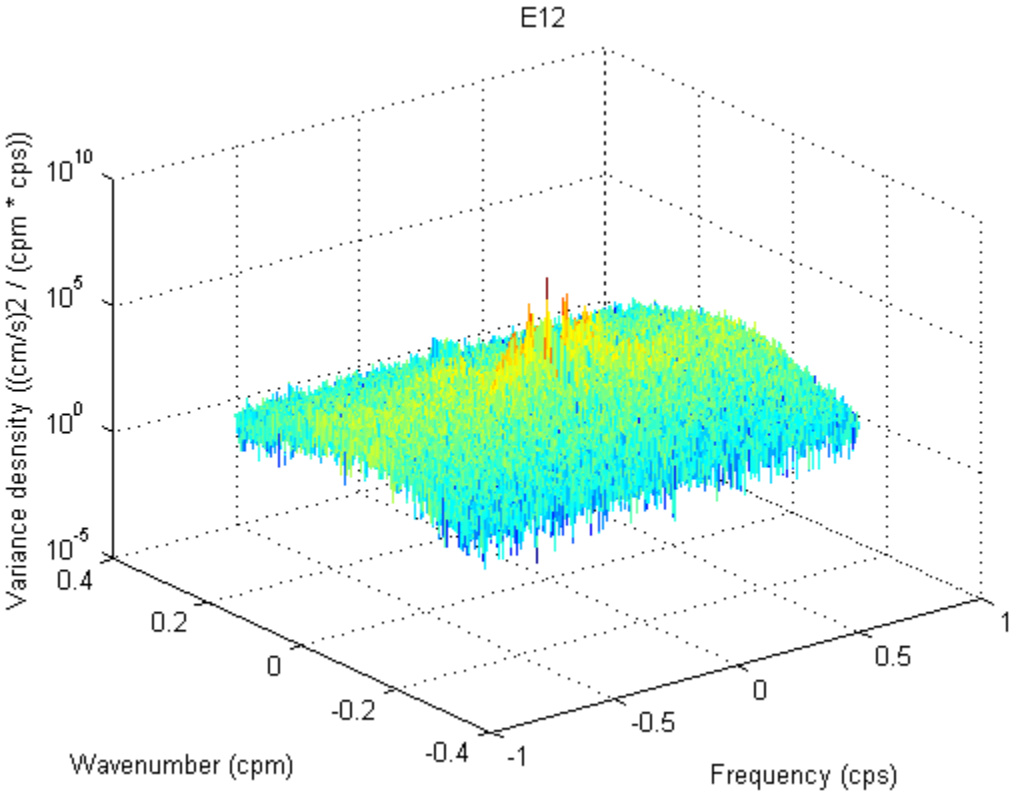
Part A - Spectra

```
% Note how noisy the data are. -Calculate the variance of the data over the
% “good” ranges, say 10-201
var_good = var(reshape(records.v_good(:,:,:), 1,1,[]))
fprintf('The variance for the "good" data (calculated using var()) \nis %d\n',...
    var_good);


% -Calculate the spectrum using the Matlab routine “fft2”
interval_space = records.space_good(2) - records.space_good(1);
interval_time = records.time(2) - records.time(1);

n_time = length(records.time);
n_space = length(records.space_good);

df_time = (1 / interval_time) / n_time;
df_space = (1 / interval_space) / n_space;

f_time = -(n_time*df_time)/2 : df_time : (n_time*df_time)/2;
f_space = -(n_space*df_space)/2 : df_space : ...
    (n_space*df_space)/2;


for i=1:6
    records.fft(:,:,i) = fft2(records.v_good(:,:,i));
    records.ehat(:,:,i) = ...
        abs(records.fft(1:length(f_space)-1,1:length(f_time)-1,i).^2) ./ ...
        (df_time * df_space * n_time^2 * n_space^2);
end



% -Average the six two-dimensional spectral estimates to get the 12 dof
% estimate requested. Now, average them.
average_ehat = mean(records.ehat,3);


% -Plot the log10 of the spectrum using “mesh”, “surf”, or “contour”. Make
% at least one plot looking from directly above (“view (90,90)”).

% We could show only one half of this spectrum... the negative
% cycles-per-second axis is redundant.  Here we show the whole thing, see
% the E96 case (below) for an example where we plot only the positive half
% of the spectrum.

figure(3);
clf();
s=surf(f_time(2:end), f_space(2:end), fftshift(records.ehat(:,:,1)), 'EdgeColor','
set(gca, 'zsc', 'log')
set(s, 'cdata', log10(get(s, 'cdata'))); % scale CDATA here
view(0,90);
```

```
xlabel('Frequency (cps)');
ylabel('Wavenumber (cpm)');
zlabel('Variance desnsity ((cm/s)2 / (cpm * cps))');
title('E12');

figure(13);
clf();
s=surf(f_time(2:end), f_space(2:end), fftshift(records.ehat(:,:,1)), 'EdgeColor','
set(gca, 'zsc', 'log')
set(s, 'cdata', log10(get(s, 'cdata'))); % scale CDATA here

xlabel('Frequency (cps)');
ylabel('Wavenumber (cpm)');
zlabel('Variance desnsity ((cm/s)2 / (cpm * cps))');
title('E12');



% -Normalize such that the variance of the data equals the volume under the
% spectrum (positive frequencies, positive and negative wavenumbers). The
% negative frequencies are just a mirror image of the positive frequencies.
% Negative and positive wavenumbers are mirror images as well.

% find that variance
% To get the spectral variance to match the directly calculated variance,
% include the zero-wavenumber band-for positive frequencies only, and the
% zero-frequency band for positive wavenumbers only. Exclude the
% zero-frequency-zerowavenumber mean value.

% Since we'll be summing these, we can just set the values we don't want to
% include to 0.
variance_ehat = average_ehat;
% zero the zero-wavenumber, negative frequency components:
variance_ehat(1, 251:end) = 0;
% zero the zero frequcny, negative wavenumber components:
variance_ehat(97:end, 1) = 0;
good_values = reshape(average_ehat*(df_time*df_space),1,[]);
% remove the mean
good_values = good_values(2:end);
variance_in_fft = sum(good_values);

fprintf('The variance for the "good" data (calculated using FFT) \nis %d\n',...
    variance_in_fft);
```

```
var_good =

  521.7184

The variance for the "good" data (calculated using var())
is 5.217184e+002
The variance for the "good" data (calculated using FFT)
is 5.188230e+002
```

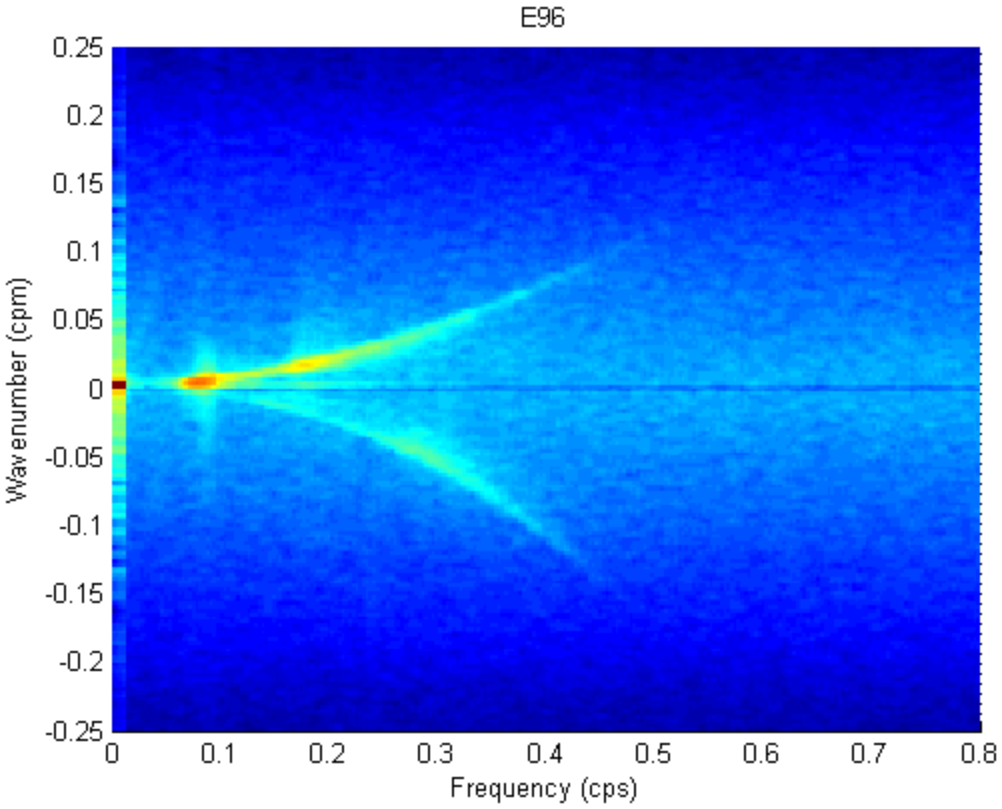

E12
Wavenumber (cpm)
Frequency (cps)
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

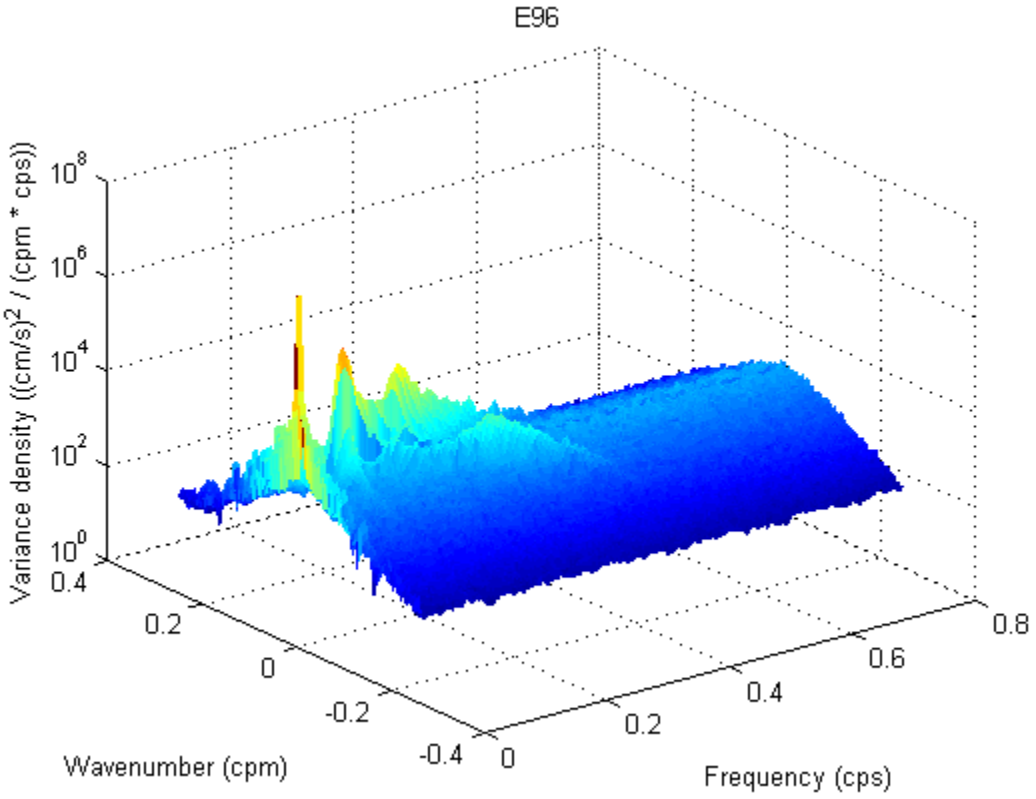

E12
Variance desnsity ((cm/s)2 / (cpm * cps))
10^10
10^5
10^0
10^-5
0.4
0.2
0
-0.2
-0.4
Wavenumber (cpm)
-1
-0.5
0
0.5
1
Frequency (cps)

Part B

Steps: B -To achieve a 96 degree of freedom spectral estimate we need to average eight of the 12 dof estimates together. Looking at the 12 dof estimate, decide how to smooth it so as to lose minimal information Create an array a = ones (nfreq, mk)./(nfreq•mk) , such that nfreq•mk?8 and convolve the spectrum with it: “E96=conv2(E12, a)”

```
% We can pick and choose nfreq and mk to adjust smoothing in the time
% and spatial dimensions.  (rows=space, columns=time)
a = ones(2,4)./(4*2);

% Low-pass the E12 estimate.
e96 = conv2(average_ehat, a);

% Plot the log10 of this spectrum and decide which waves are going which
% way. Be sure to label the axes of all plots and indicate wavenumbers and
% frequency scales, as well as a scale for the log spectral magnitude.
figure(5);
clf();
e96_shifted = fftshift(e96);

% The negative part of the time spectrum is a mirror of the positive side,
% so we can "fold" it when plotting.
f_time_positive = f_time(f_time>=0);
e96_shifted_positive_time = 2 * e96_shifted(:, 252:end-1);

s=surf(f_time_positive, f_space, e96_shifted_positive_time, 'EdgeColor','none');
set(gca, 'zsc', 'log')
set(s, 'cdata', log10(get(s, 'cdata'))); % scale CDATA here
view(0,90);

xlabel('Frequency (cps)');
ylabel('Wavenumber (cpm)');
zlabel('Variance density ((cm/s)^2 / (cpm * cps))');
title('E96');

figure(15);
clf();
s=surf(f_time_positive, f_space, e96_shifted_positive_time, 'EdgeColor','none');
set(gca, 'zsc', 'log')
set(s, 'cdata', log10(get(s, 'cdata'))); % scale CDATA here

xlabel('Frequency (cps)');
ylabel('Wavenumber (cpm)');
zlabel('Variance density ((cm/s)^2 / (cpm * cps))');
title('E96');
```

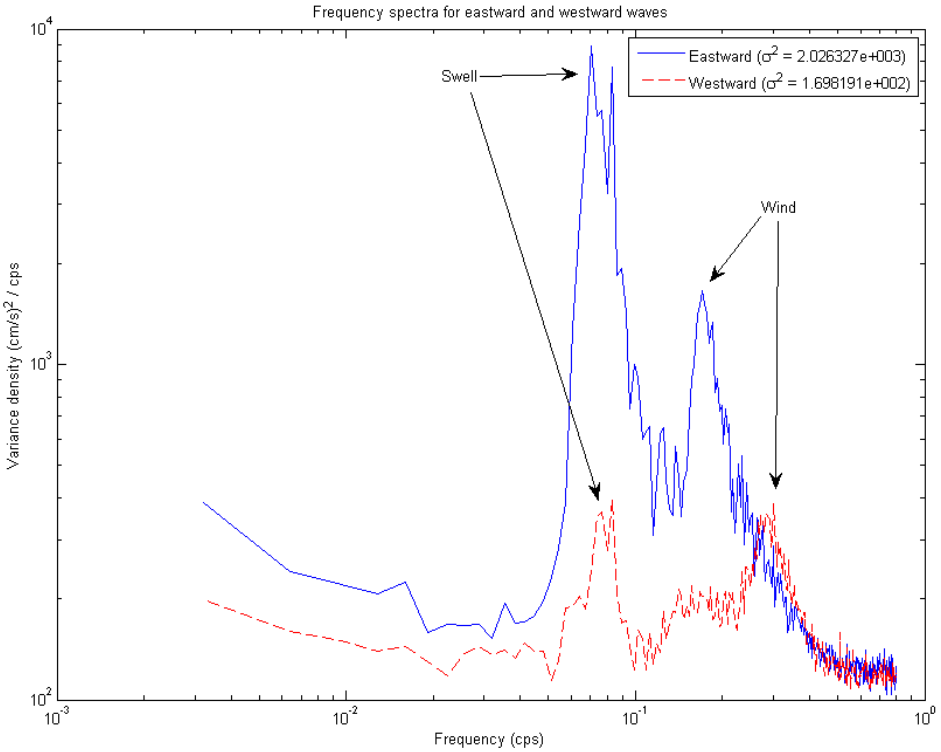

E96
Wavenumber (cpm)
Frequency (cps)
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


E96
Variance density ((cm/s)^2 / (cpm * cps))
10^8
10^6
10^4
10^2
10^0
0.4
0.2
0
-0.2
-0.4
Wavenumber (cpm)
0
0.2
0.4
0.6
0.8
Frequency (cps)

Part C

Steps: C Integrate the spectrum over wavenumber to get the frequency spectra for the eastward & westward propagating waves. Plot loglog, with both spectral on the same plot. What are the units of the frequency spectrum? Label the axes.

```
% Only use the positive time frequency (1:250).  When doing this, we need
% to "fold" that energy over to the positive side, so mulitply by 2.
positive_ehat = 2 * average_ehat(1:96, 1:250);
negative_ehat = 2 * average_ehat(97:end, 1:250);
% Integrate the wavenumbers
spec_pos = sum(positive_ehat.*df_space);
spec_neg = sum(negative_ehat.*df_space);

% Find the variance for each via integration
var_pos = sum(spec_pos.*df_time);
var_neg = sum(spec_neg.*df_time);

% Turn them into strings that can go on the plot
west_str = sprintf('Westward (\\sigma^2 = %d)', var_neg);
east_str = sprintf('Eastward (\\sigma^2 = %d)', var_pos);

f_time_positive = (0:df_time:(df_time*(length(spec_pos)-1)));


figure(6);
clf()
loglog(f_time_positive, spec_pos);
hold on
loglog(f_time_positive, spec_neg, 'r--');

xlabel('Frequency (cps)');
ylabel('Variance density (cm/s)^2 / cps');
title('Frequency spectra for eastward and westward waves');
legend(east_str, west_str);

% Label features on the plot
annotation(gcf(),'textarrow',[0.507820646506778 0.59541188738269],...
    [0.867709815078236 0.870554765291607],'TextEdgeColor','none',...
    'String',{'Swell'});
annotation(gcf(),'arrow',[0.499478623566215 0.614181438998957],...
    [0.846372688477953 0.352773826458037]);
annotation(gcf(),'textarrow',[0.758081334723671 0.713242961418144],...
    [0.695590327169275 0.617354196301565],'TextEdgeColor','none',...
    'String',{'Wind'});
annotation(gcf(),'arrow',[0.773722627737227 0.771637122002086],...
    [0.690322901849219 0.365576102418208]);
```



Published with MATLAB® 7.12