

---

# SIO221A Homework #3 (Eric Gallimore)

## Table of Contents

1) Forming a spectral estimate of Vertical Displacement: .....	1
1. Plot the data and check for “glitches”. This is always the first step. ....	1
2. Choose a clean segment of data at least 6000 points long. ....	3
200m case: Fourier transform and form the spectral estimate .....	3
500m case: Fourier transform and form the spectral estimate .....	5
2) Increasing statistical stability: .....	7
3) Forming a spectral estimate of Vertical Velocity: .....	9
4) Comparing Spectral Estimates .....	12

## 1) Forming a spectral estimate of Vertical Displacement:

Produce estimates of the power spectrum of isopycnal vertical displacement using isopycnals whose mean depths are 200 m and 500 m.

```
% OK, first, load the data
load('farfield_Disprelim.mat')

% Now, find the indexes of the 200m and 500m isopycnals.
index_200m = find(depth >= 200, 1);
index_500m = find(depth >= 500, 1);

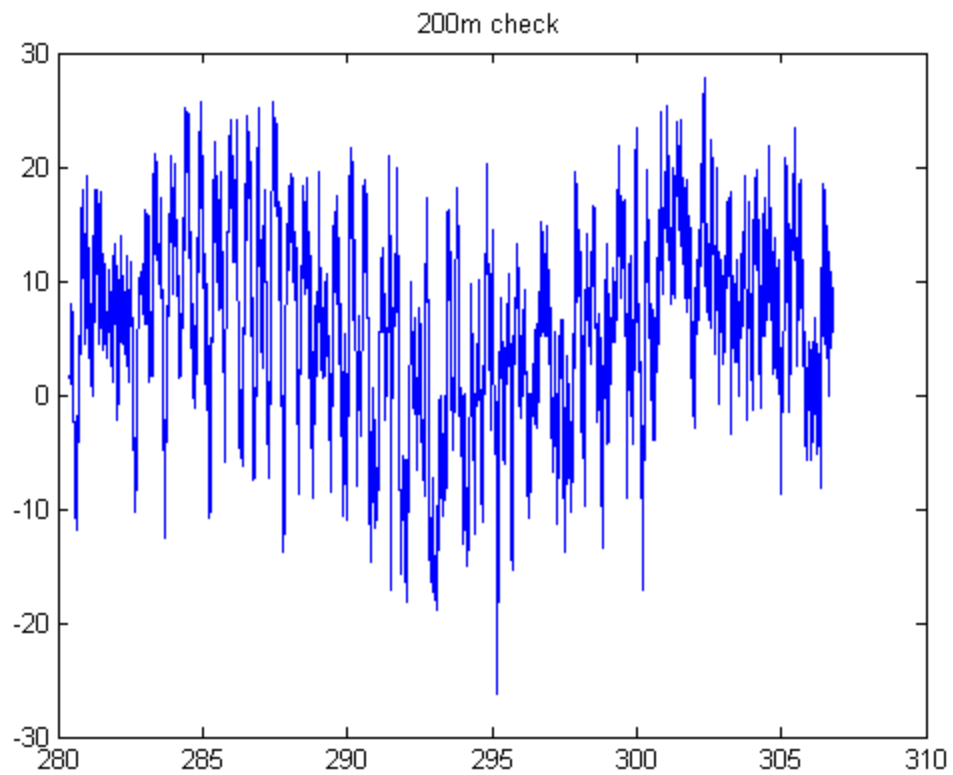
% Here's how:
%
%
```

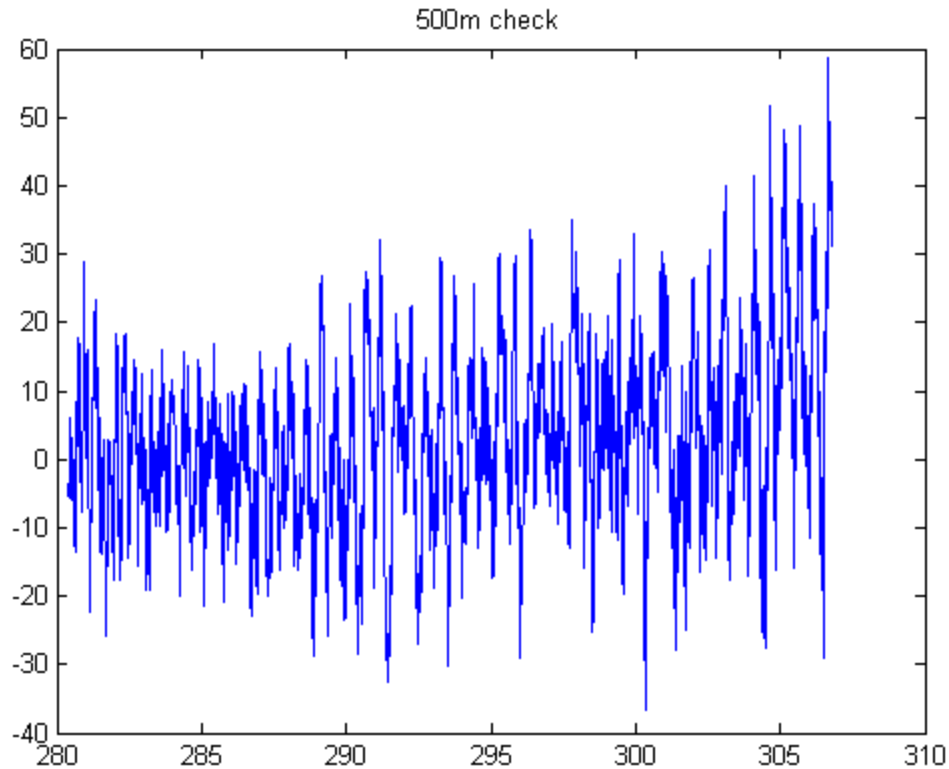
### 1. Plot the data and check for “glitches”. This is always the first step.

```
% Plot the 200m case. This is just a quick check, so we aren't producing a
% publication-worthy plot.
figure(1);
clf();
plot(time, d(index_200m, :));
title('200m check');

% and repeat for the 500m case.
figure(2);
clf();
plot(time, d(index_500m, :));
title('500m check');
```

% Visually, those both look good, so we will proceed.





## 2. Choose a clean segment of data at least 6000 points long.

( $t = 4 \text{ min} = 1/15 \text{ hr}$ ) If there are “glitches”, set them equal to the mean value of the record. (“?  
=mean(D(100,:))”)

```
data_200m = d(index_200m, :);  
data_500m = d(index_500m, :);  
fs = 1/(4*60); %Hz
```

```
% 3. Fourier transform and form the spectral estimate  
% Here, N points have been Fourier transformed, the record duration is N /15 hour  
%  
% Note 1: For a real time-series, the spectrum will be mirror symmetric about “ze  
%  
% Note 2: The Fourier transform routine in Matlab is normalized such that you ne  
%
```

## 200m case: Fourier transform and form the spectral estimate

```
A_200m = fft(data_200m);
```

```
fregs = 0:fs/length(data_200m):(fs/2);
% drop the DC element
fregs = fregs(2:end);

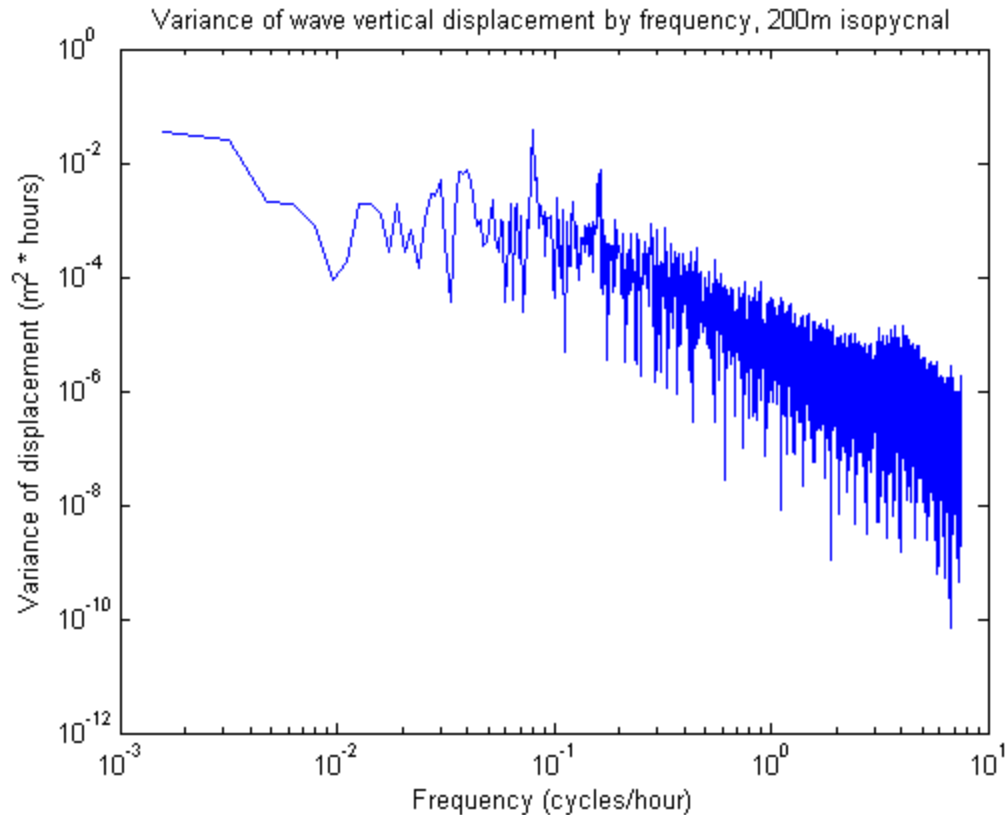
% We are dealing with a real signal, so we only care about positive, real
% frequency components. To get the power, we want to "fold" both sides of
% the spectrum together. Since they are the same, we can just take one
% side and multiply by 2. (That's probably a terrible explanation, but
% it's the one I use in my head.)
Ehat_200m_ss_real = 2 * abs(A_200m(2:length(fregs)+1)).^2 ./ (1/fs) ./ ...
    (length(data_200m)^2);

% Verify that sum(fourier coefficients) = sample variance
variance_200m = var(data_200m);
variance_200m_in_fft = sum(Ehat_200m_ss_real*(1/fs));
fprintf('Variance of 200m data (using var): %d m^2\n', variance_200m);
fprintf('Variance of 200m data (using fft): %d m^2\n', variance_200m_in_fft);

% Verify that we did this properly
if(0)
    figure(11);
    subplot(2,1,1);
    % Plot, with power in dB
    plot(fregs, 10*log10(Ehat_200m_ss_real));
    title('200m PSD (for verification of methodology)');
    % Check this against the output of pwelch
    subplot(2,1,2);
    pwelch(data_200m);
end

% Plot log10(E(f) vs. log10 (f) in cycles per hour. Label All axes.
figure(3);
clf();
% fregs is in cycles/s, so multiply by 3600 to get cycle/hour.
% The following line would plot log10(y) vs log10(x) on a linear scale.
% This works, but it means that the axes are log10(x,y), which makes no
% intuitive sense. I think that we actually want a loglog plot (see below)
%plot(log10(fregs*60*60), log10(Ehat_200m_ss_real));
loglog(fregs*60*60, Ehat_200m_ss_real);
% An FFT yields coefficients with the same units as the input signal.
ylabel('Variance of displacement (m^2 * hours)');
xlabel('Frequency (cycles/hour)');
title('Variance of wave vertical displacement by frequency, 200m isopycnal');

    Variance of 200m data (using var): 6.948121e+001 m^2
    Variance of 200m data (using fft): 6.947400e+001 m^2
```



## 500m case: Fourier transform and form the spectral estimate

Yes, this should have become a function. If I had to do it one more time...

```
A_500m = fft(data_500m);

freqs = 0:fs/length(data_500m):(fs/2);
% drop the DC element
freqs = freqs(2:end);

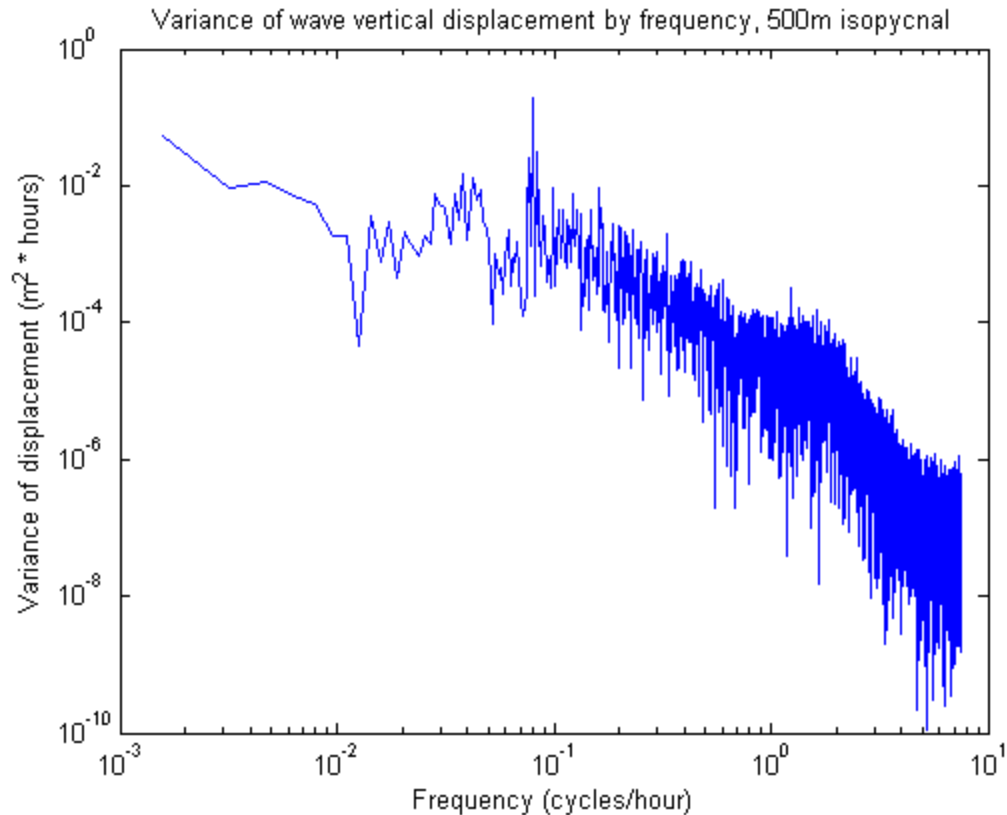
% We are dealing with a real signal, so we only care about positive, real
% frequency components. To get the power, we want to "fold" both sides of
% the spectrum together. Since they are the same, we can just take one
% side and multiply by 2. (That's probably a terrible explanation, but
% it's the one I use in my head.)
Ehat_500m_ss_real = 2 * abs(A_500m(2:length(freqs)+1)).^2 ./ (1/fs) ./ ...
    (length(data_500m)^2);

% Verify that sum(fourier coefficients) = sample variance
variance_500m = var(data_500m);
variance_500m_in_fft = sum(Ehat_500m_ss_real*(1/fs));
fprintf('Variance of 500m data (using var): %d m^2\n', variance_500m);
fprintf('Variance of 500m data (using fft): %d m^2\n', variance_500m_in_fft);
```

```
% Verify that we did this properly
if(0)
    figure(12);
    subplot(2,1,1);
    % Plot, with power in dB
    plot(freqs, 10*log10(Ehat_500m_ss_real));
    title('500m PSD (for verification of methodology)');
    % Check this against the output of pwelch
    subplot(2,1,2);
    pwelch(data_200m);
end

% Plot log10(E(f) vs. log10 (f) in cycles per hour. Label All axes.
figure(4);
clf();
% freqs is in cycles/s, so multiply by 3600 to get cycle/hour.
% The following line would plot log10(y) vs log10(x) on a linear scale.
% This works, but it means that the axes are log10(x,y), which makes no
% intuitive sense. I think that we actually want a loglog plot (see below)
%plot(log10(freqs*60*60), log10(Ehat_200m_ss_real));
loglog(freqs*60*60, Ehat_500m_ss_real);
ylabel('Variance of displacement (m^2 * hours)');
xlabel('Frequency (cycles/hour)');
title('Variance of wave vertical displacement by frequency, 500m isopycnal');

    Variance of 500m data (using var): 1.669780e+002 m^2
    Variance of 500m data (using fft): 1.669604e+002 m^2
```



## 2) Increasing statistical stability:

Now, repeat “problem 1” six times, forming “averaged” spectral estimates at 200m and 500 m using times 1-1000,1001-2000,...5001-6000. Average each set of estimates together at like values of frequency, normalize to preserve variance, and plot log-log vs. frequency. Put the spectral estimates from the 200m and 500m depth zones on the same plot. Identify the tidal and buoyancy frequencies. Note a few other differences/similarities in the data.

```
% Do the 200m case.
for i=1:5
    start_idx = (i-1) * 1000 + 1;
    end_idx = start_idx + 999;
    A_partial_200m(i, :) = fft(data_200m(start_idx:end_idx));
    freqs = 0:fs/1000:(fs/2);
    % drop the DC element
    reqs = freqs(2:end);
    Ehat_partial_200m(i, :) = ...
        2 * abs(A_partial_200m(i,2:length(freqs)+1)).^2 ./ (1/fs) ./ 1000^2;
    %set(gca, 'YScale', 'log')
    %set(gca, 'XScale', 'log')
    %hold on;
    %plot(freqs*60*60, Ehat_partial_200m(i, :));
end

% Do the 500m case.
for i=1:5
```

```
start_idx = (i-1) * 1000 + 1;
end_idx = start_idx + 999;
A_partial_500m(i, :) = fft(data_500m(start_idx:end_idx));
freqs = 0:fs/1000:(fs/2);
% drop the DC element
reqs = freqs(2:end);
Ehat_partial_500m(i, :) = ...
    2 * abs(A_partial_500m(i,2:length(freqs)+1)).^2 ./ (1/fs) ./ 1000^2;
%set(gca, 'YScale', 'log')
%set(gca, 'XScale', 'log')
%hold on;
%plot(freqs*60*60, Ehat_partial_200m(i, :));
end

% Get the average values (average of columns)
Ehat_200m_average = mean(Ehat_partial_200m);
Ehat_500m_average = mean(Ehat_partial_500m);

% Plot these values.
figure(5);
clf();

loglog(freqs*60*60, Ehat_200m_average);
hold on;
loglog(freqs*60*60, Ehat_500m_average, 'r--');
ylabel('Variance of displacement (m^2 * hours)');
xlabel('Frequency (cycles/hour)');
legend('200m isopycnal', '500m isopycnal');
title('Variance of wave vertical displacement by frequency, averaged');

% Identify the tidal frequencies (identified because we expect something
% with a period of about 6 hours and something with a period of about 12
% hours).
annotation(gcf(), 'textarrow', [0.430359937402191 0.430359937402191], ...
    [0.783815028901734 0.731791907514451], 'TextEdgeColor', 'none', ...
    'String', {'Tidal Frequency (\tau approx 6h)'});
annotation(gcf(), 'textarrow', [0.341158059467919 0.341596244131456], ...
    [0.86242774566474 0.829710982658961], 'TextEdgeColor', 'none', ...
    'String', {'Tidal Frequency (\tau approx 12h)'});

% The buoyancy frequency should be somewhere around 0.5-5 cph. I don't see
% anything obvious here, but there are some possibilities.
annotation(gcf(), 'textarrow', [0.698140076668857 0.698578261332394], ...
    [0.563169344737741 0.530452581731962], 'TextEdgeColor', 'none', ...
    'String', {'Possible buoyancy', 'frequency for 500m'});
annotation(gcf(), 'textarrow', [0.811181515292782 0.811619699956319], ...
    [0.431929483787567 0.399212720781788], 'TextEdgeColor', 'none', ...
    'String', {'Possible buoyancy', 'frequency for 200m'});

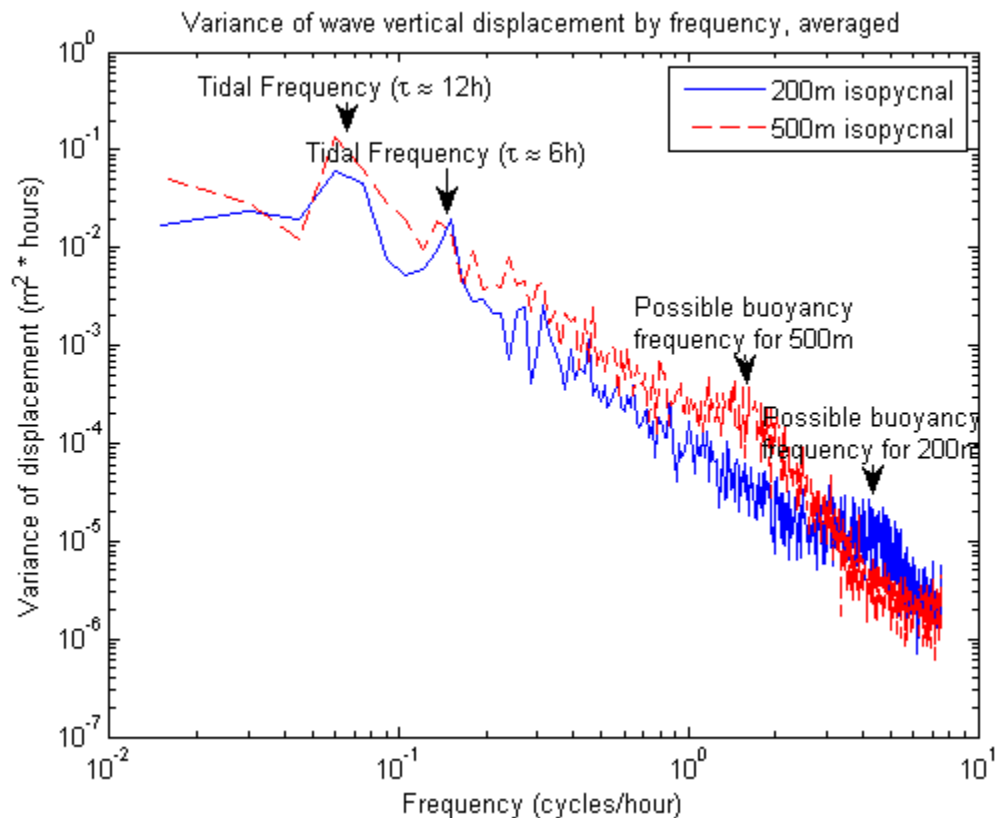
disp('Note a few other differences/similarities in the data:');
disp('In both cases, the variance is higher at lower frequencies.');
```

---



*Note a few other differences/similarities in the data:*

*In both cases, the variance is higher at lower frequencies. The variance is spread more in the 500m data. If we have correctly identified the buoyancy frequency, this may suggest that the density stratification at 500m is less stable*



### 3) Forming a spectral estimate of Vertical Velocity:

Repeat problem 2 using a series of “effective vertical velocity” Normalize correctly, use frequency in cycles per hour, velocity in meters/s. - Plot log-log, positive frequencies only. State the units, label the axes. - Discuss the physical implications of the similarity/difference between spectral estimates at the two depths.

```
% Make a time-series of velocity for 200m data
% 240 seconds between samples
w_200m = (data_200m(1:end-1) - data_200m(2:end)) ./ 240; % m/s

% And do it for 500m
w_500m = (data_500m(1:end-1) - data_200m(2:end)) ./ 240; % m/s

% Optionally look at the time series of this data to do a sanity check,
% look for glitches.
if (0)
    figure()
```

```
    plot(w_200m);
    figure()
    plot(w_500m);
end
% There are a few outliers in the 200m data, but they don't look bad enough
% to manually correct.

% Do the 200m case.
for i=1:5
    start_idx = (i-1) * 1000 + 1;
    end_idx = start_idx + 999;
    A_partial_w_200m(i, :) = fft(w_200m(start_idx:end_idx));
    freqs = 0:fs/1000:(fs/2);
    % drop the DC element
    freqs = freqs(2:end);
    Ehat_partial_w_200m(i, :) = ...
        2 * abs(A_partial_w_200m(i,2:length(freqs)+1)).^2 ./ (1/fs) ./ 1000^2;
    % The 1000 above is because we are looking at 1000 samples at a time.
    %set(gca, 'YScale', 'log')
    %set(gca, 'XScale', 'log')
    %hold on;
    %plot(freqs*60*60, Ehat_partial_200m(i, :));
end

% Do the 500m case.
for i=1:5
    start_idx = (i-1) * 1000 + 1;
    end_idx = start_idx + 999;
    A_partial_w_500m(i, :) = fft(w_500m(start_idx:end_idx));
    freqs = 0:fs/1000:(fs/2);
    % drop the DC element
    freqs = freqs(2:end);
    Ehat_partial_w_500m(i, :) = ...
        2 * abs(A_partial_w_500m(i,2:length(freqs)+1)).^2 ./ (1/fs) ./ 1000^2;
    %set(gca, 'YScale', 'log')
    %set(gca, 'XScale', 'log')
    %hold on;
    %plot(freqs*60*60, Ehat_partial_200m(i, :));
end

% Get the average values (average of columns)
Ehat_w_200m_average = mean(Ehat_partial_w_200m);
Ehat_w_500m_average = mean(Ehat_partial_w_500m);

% Plot these values.
figure(6);
clf();

loglog(freqs*60*60, Ehat_w_200m_average);
hold on;
loglog(freqs*60*60, Ehat_w_500m_average, 'r--');
% The units on this are a bit confusing. I think that this is the clearest
% way to represent them.
```

```
ylabel('Variance of velocity (m^2/s * hours)')
xlabel('Frequency (cycles/hour)');
legend('200m isopycnal', '500m isopycnal');
title('Variance of vertical velocity by frequency, averaged');

% Check that we are on the right track.
% Verify that sum(fourier coefficients) = sample variance
% The 500m numbers look a bit off, but at least they are the same OOM.
variance_w_200m = var(w_200m(1:6000));
variance_w_200m_in_fft = sum(Ehat_w_200m_average*(1/fs));
fprintf('Variance of 200m velocity data (using var): %d m^2/s\n', variance_w_200m);
fprintf('Variance of 200m velocity data (using fft): %d m^2/s\n', variance_w_200m_in_fft);
fprintf('Mean of 200m velocity data (using mean & abs): %d m/s\n', mean(abs(w_200m)));

variance_w_500m = var(w_500m(1:6000));
variance_w_500m_in_fft = sum(Ehat_w_500m_average*(1/fs));
fprintf('Variance of 500m velocity data (using var): %d m^2/s\n', variance_w_500m);
fprintf('Variance of 500m velocity data (using fft): %d m^2/s\n', variance_w_500m_in_fft);
fprintf('Mean of 500m velocity data (using mean & abs): %d m/s\n', mean(abs(w_500m)));

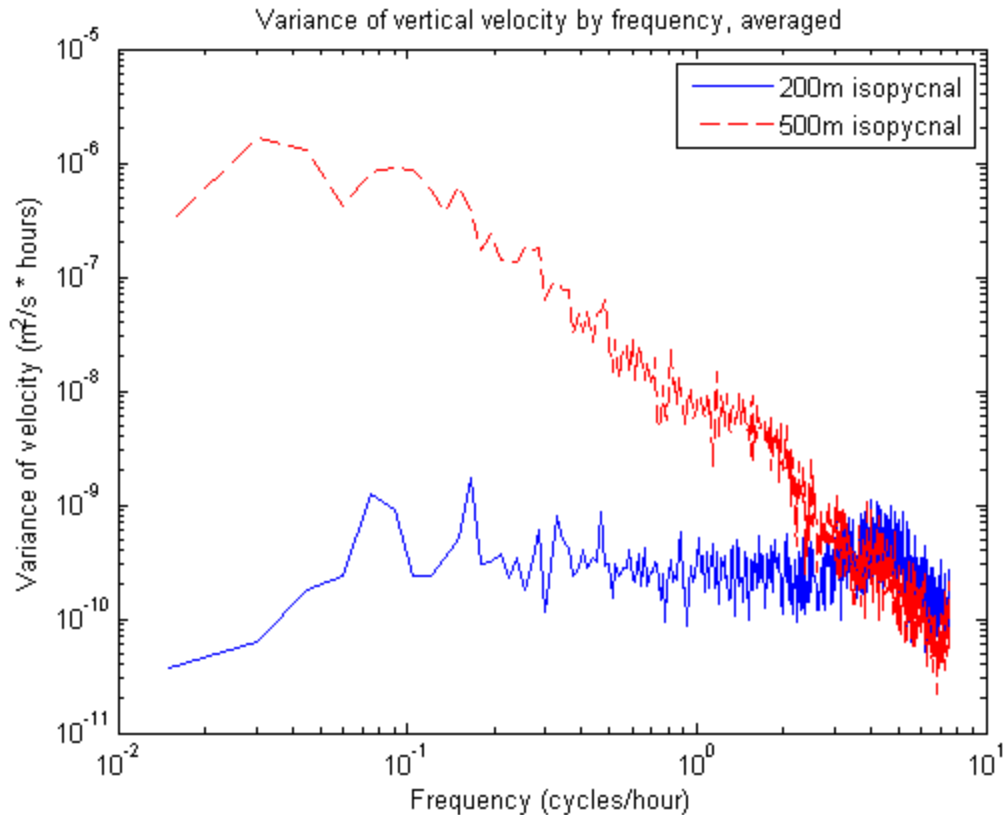
% - Discuss the physical implications of the similarity/difference between
%   spectral estimates at the two depths.
disp('The vertical velocity at 500m is much greater than the velocity at');
disp('200m (from mean).');
disp('The velocity at 500m is changing more, particularly at');
disp('low frequencies. This suggests that this depth is less stable.');
```

Perhaps the density as a function of depth is more variable around  
500m, thus causing the buoyancy frequency and the amplitude of the  
related wave components to vary more, which matches our expectation  
from the displacement data.

There is some evidence of a tidal-period related change in velocity  
at both isopycnals.

(Examining the time series data supports these conclusions.)

```
Variance of 200m velocity data (using var): 3.784630e-005 m^2/s
Variance of 200m velocity data (using fft): 3.704667e-005 m^2/s
Mean of 200m velocity data (using mean & abs): 4.575273e-003 m/s
Variance of 500m velocity data (using var): 3.281860e-003 m^2/s
Variance of 500m velocity data (using fft): 2.718612e-003 m^2/s
Mean of 500m velocity data (using mean & abs): 4.995329e-002 m/s
The vertical velocity at 500m is much greater than the velocity at
200m (from mean).
The velocity at 500m is changing more, particularly at
low frequencies. This suggests that this depth is less stable.
Perhaps the density as a function of depth is more variable around
500m, thus causing the buoyancy frequency and the amplitude of the
related wave components to vary more, which matches our expectation
from the displacement data.
There is some evidence of a tidal-period related change in velocity
at both isopycnals.
(Examining the time series data supports these conclusions.)
```



## 4) Comparing Spectral Estimates

Take the displacement spectra of problem 2 and multiply them by  $(2\pi \cdot \text{frequency})^2$ , where the frequency is in cycles per second. How do these spectra compare with those in part 3. (Plot them in the same plot). Why should these different spectral estimates compare?

```
f_Hz = 1/(4*60); %Hz

% 200m case
figure(7);
clf();
loglog(freqs*60*60, Ehat_200m_average(1:length(freqs)) .* ((2*pi*f_Hz)^2));
hold on;
loglog(freqs*60*60, Ehat_w_200m_average, 'r--');
% I don't think we have a single axis label here.
ylabel('Variance of velocity (m^2/s * hours)');
xlabel('Frequency (cycles/hour)');
legend('From displacement spectrum', 'From velocity data');
title('Comparing spectral estimates, 200m data');

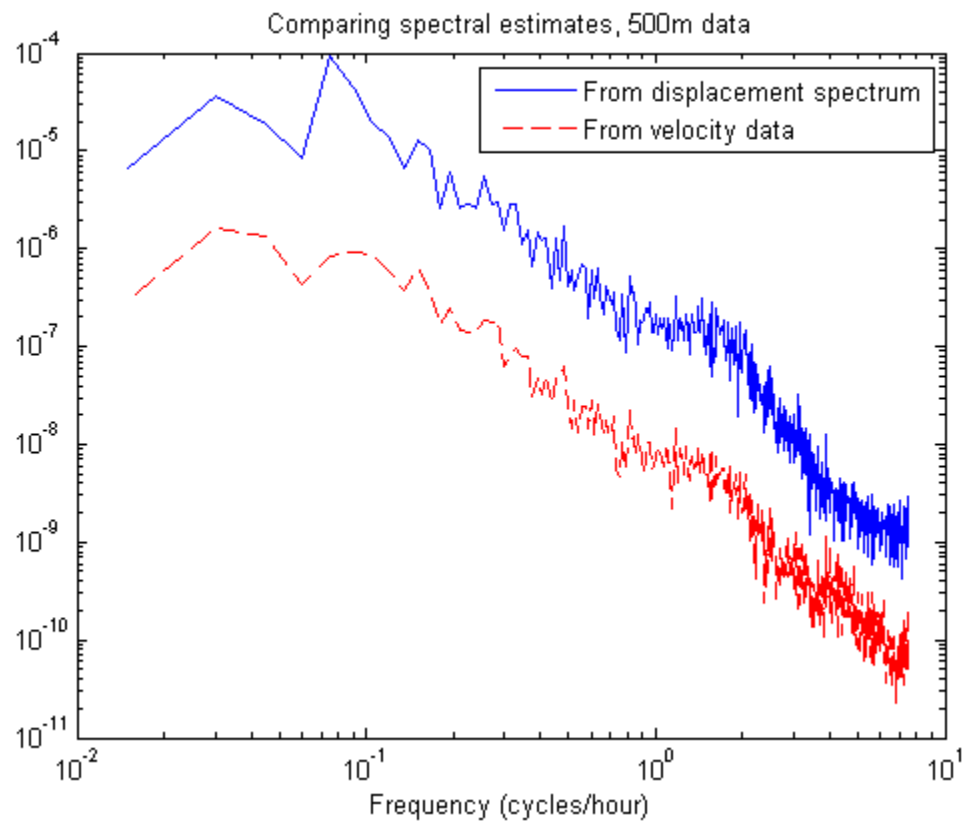
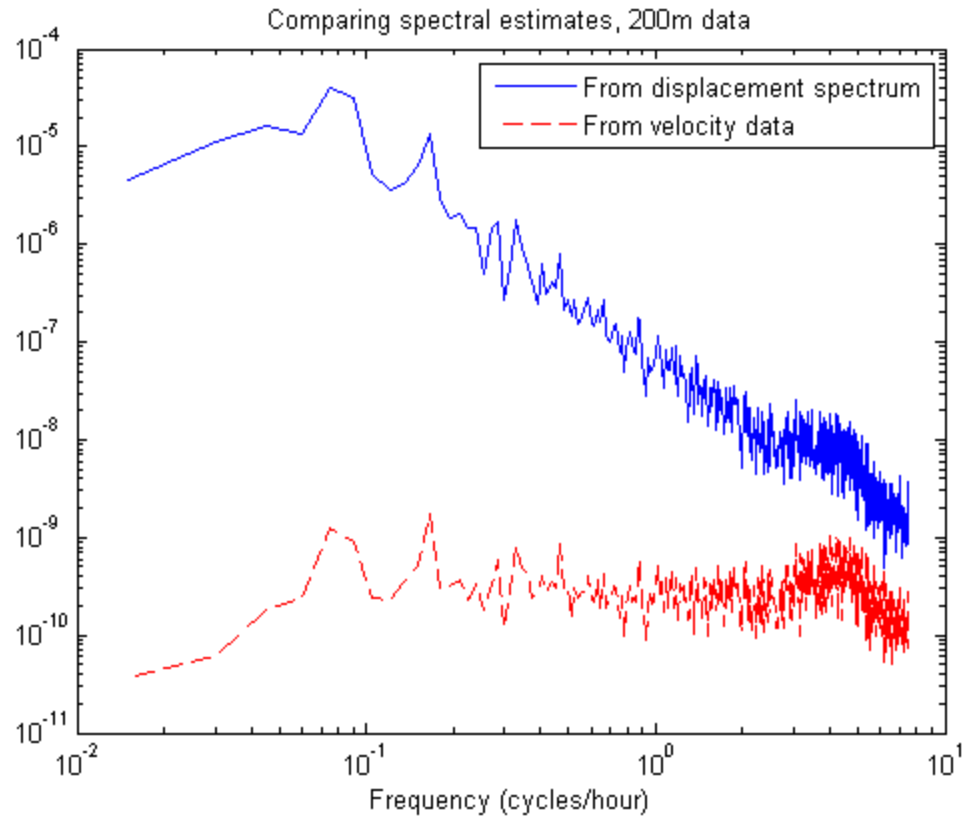
% 500m case
figure(8);
clf();
loglog(freqs*60*60, Ehat_500m_average(1:length(freqs)) .* ((2*pi*f_Hz)^2));
hold on;
loglog(freqs*60*60, Ehat_w_500m_average, 'r--');
```

```
% I don't think we have a single axis label here.
ylabel('Variance of velocity (m^2/s * hours)')
xlabel('Frequency (cycles/hour)');
legend('From displacement spectrum', 'From velocity data');
title('Comparing spectral estimates, 500m data');

disp('These spectral estimates compare because the velocity and displacement');
disp('are related by a factor of time. In a naive sense, velocity is displacement');
disp('over time. Frequency is 1/time. So, multiplying the displacement by a');
disp('frequency is the same as dividing by a unit of time, and we would expect it');
disp('to yield something with at least the units of velocity.');
```

*We need to use frequency squared because we are normalizing by the*  
*number of points squared, and the number of points is proportional to*  
*time. (Yes, that is hand waving.)*  
*I don't know why they the result isn't an exact overlap, but my*  
*intuition is that we aren't preserving mean values when doing this.*  
*I'm also still skeptical of my 200m velocity PSD, but I can't*  
*find a problem with it.*

```
%end
```



*Published with MATLAB® 7.12*