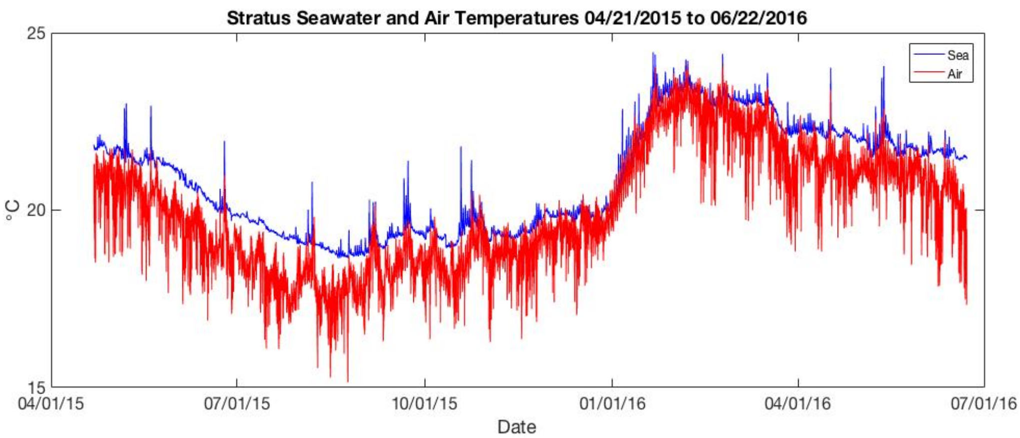
SIOC 221A Homework 7 | **Julia Dohner**
Prof. Sarah Gille | Due Date: November 21, 2017

I certify that this represents my own work and that I have not worked with classmates or other individuals to complete this assignment. -JLD

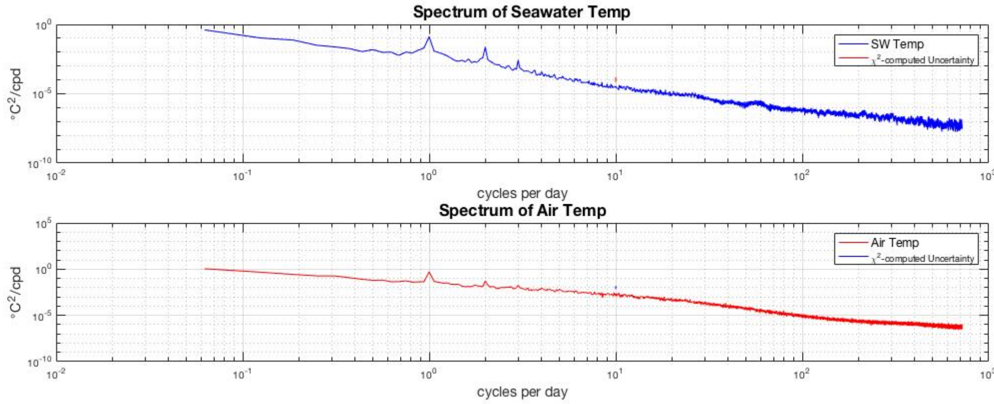
1. Inspect the data.

There do not appear to be any gaps in the data. The data is taken over the course of 427.588 days. The air temperature seems to on the whole lower than the seawater temperature. Both sets of data show a steady decline starting at the beginning of the record, but then a steady increase in temperature between January and February of 2016, then again a steady decline in temperature thereafter. There do seem to be wiggles in the data, suggesting that the temperature records are driven by oscillatory processes. The air temperature record also seems to show more variability than the seawater temperature record.



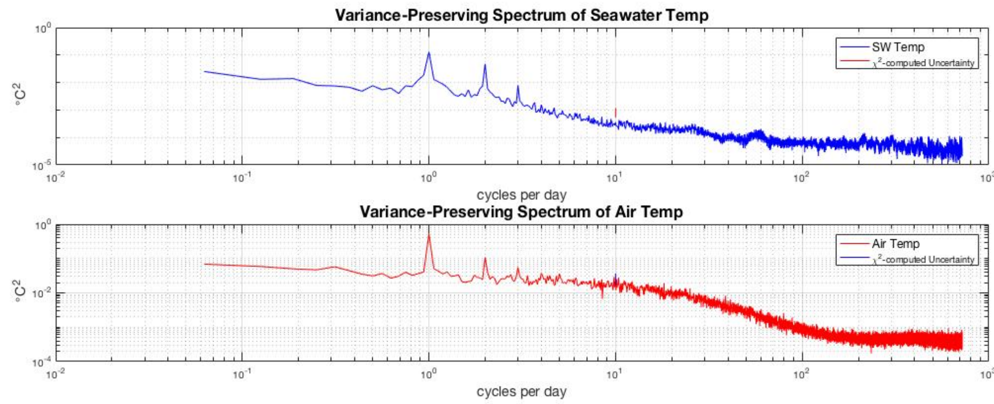
2. Compute the spectra for T_o and T_a.

The Nyquist frequency for each of the records is 1 cycle per every 2 minutes, which converts to 720 cycles per day. The frequency resolution for both of the full records is 1/427.588 days, or 0.0023 cpd. After splitting the data into 16-day chunks, the frequency resolution changes to 1/16 days, or 0.0625 cpd. In the seawater record, the spectral peaks are located at 1 cpd and 2 cpd, which correspond to the diurnal cycle and the semi-diurnal tidal frequencies, respectively. The peak at 1 cpd is much more prominent in the air temperature spectra (and the peak at 2 cpd practically non-existent), because the air temperature is strongly influenced by the diurnal cycle (and much less so, if at all, by tides). In both spectra, the slope drops 1/2 order of magnitude in y for a one order of magnitude increase in x, corresponding to a spectral slope of $f^{-1/2}$. But after the 100 cpd mark, the slope of the air temperature spectrum decreases more rapidly, changing to roughly a slope of f^{-1}. The amplitudes are greater in the air temperature spectrum than the seawater temperature spectrum. There is also more noise at higher frequencies in the air temperature spectrum, which is possibly because the air temperature is much more variable and strongly influenced by eddies and other high frequency perturbations that pass through at a higher rate than in seawater (which I picture to be much more slower-moving and more slowly mixed compared to air).



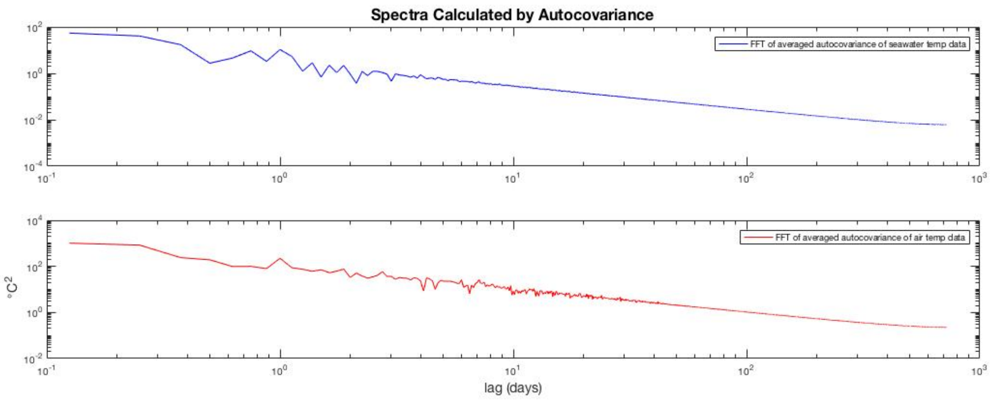
3. Show a variance preserving version of your spectra.

The peaks at lower frequencies (1 and 2 cpd) seem to become more prominent in the variance-preserving versions of my spectra. In addition, the spectra do not attenuate as much at higher frequencies since the spectra is being multiplied by greater values as the frequencies increase. In addition to the more prominent peaks at lower frequencies, there seems to be more noise at higher frequencies as well in the variance-preserving spectra.



4. Bonus: Compute the autocovariance of your data and use this to produce spectra.

Autocovariance is useful with regards to gaps in your data. Though the spectra will be affected by the locations of gaps in the data, autocovariance at least offers a formalism that allows us to track where these gaps are and where they show up in the resultant spectra. In addition, autocovariance is useful because it allows us to put an uncertainty on the autocovariance. Though this won't improve our Fourier transforms, it allows us to appropriately weight the data based on its uncertainty in a least squares fit, for example. A significant drawback to the autocovariance method is that it's computationally inefficient, especially when compared to the fast Fourier transform calculation.



Spectra Calculated by Autocovariance
FFT of averaged autocovariance of seawater temp data
FFT of averaged autocovariance of air temp data
°C²
lag (days)

```
% file SIOC 221A HW 7
%
% author Julia Dohner
%
% due date November 21, 2017
%
% I certify that this represents my own work and that I have not worked
% with classmates or other individuals to complete this assignment. -JLD

clear all; close all;

% note: 1440 minutes/day

%% inspect the data

% create empty arrays to hold time and temp data
time = [];
swTemp = [];
airTemp = [];

% time is 04/21/2015 to 06/22/2016 (427.5882 days)
time = [time; ncread(strcat('OS_Stratus_2015_D_M.nc'),'TIME')];
swTemp = [swTemp; ncread(strcat('OS_Stratus_2015_D_M.nc'),'TEMP')];
airTemp = [airTemp; ncread(strcat('OS_Stratus_2015_D_M.nc'),'AIRT')];

% examining the time increments between adjacent measurements
t_diff = diff(time);
t_diff_mean = mean(t_diff);
minDiff = min(t_diff);
maxDiff = max(t_diff);
figure('name','Differences in measurement interval');
plot(t_diff);
% after plotting the values, I see that the t_diff values range between
% 0.0006944444 and 0.00069444452. The differences are sufficiently small
% to proceed with the Fourier transform.

% checking for NaN's in data:
numNaNsw = sum(isnan(swTemp(:)));
numNaNair = sum(isnan(airTemp(:)));

% plot the time series
date0=datenum(1950,1,1); % give reference date (first date)
time2 = double(time)+date0;
figure('name','Seawater and Air Temperatures');
plot(time2, swTemp, '-b','LineWidth',1);
hold on
plot(time2, airTemp, '-r','LineWidth',1);

set(gca,'FontSize',16);
title('Stratus Seawater and Air Temperatures 04/21/2015 to 06/22/2016');
xlabel('Date');
datetick('x','mm/dd/yy')
ylabel('\circC');
legend('\fontsize{12}Sea', '\fontsize{12}Air');

%% compute the spectra for To and Ta

% data is taken every minute (86400 (seconds in a day)*t_diff = ~60 secs)
```

```
% split into 16-day chunks to resolve M2 vs. S2 tides (need to be at least
% 14.79-day chunks)
% want 16-day chunks (so chunk length is 16*1440 = 23040)
N = 23040;
M = floor(length(time)/23040);% number of chunks
T = N/1440; % total time in days in each segment (1440 mins/day) = 16

% split into segments
swTemp3 = reshape(swTemp(1:N*M),N,M);
airTemp3 = reshape(airTemp(1:N*M),N,M);

% compute fft
swTemp4 = fft(swTemp3);
airTemp4 = fft(airTemp3);
% compute squared amplitude for half of fft
swTemp_amp = (abs(swTemp4(1:length(swTemp4)/2+1,:)).^2);% +1 for even N
airTemp_amp = (abs(airTemp4(1:length(airTemp4)/2+1,:)).^2);

% check parseval's (total variance in time domain = total variance in
% frequency domain)
df = 1/T;
dt = 1/t_diff_mean;
variance_sw = sum(swTemp3.^2).*dt;% taking sums down columns
variance_air = sum(airTemp3.^2).*dt;
sum_spec_sw = sum(swTemp_amp).*df;% taking sums down columns
sum_spec_air = sum(airTemp_amp).*df;
parsevalSW = sum_spec_sw./variance_sw;% should = 1
parsevalAir = sum_spec_air./variance_air;

% multiply by a factor of 2 to account for lost variance, excluding mean
swTemp_amp(2:end-1) = 2.*swTemp_amp(2:end-1);
airTemp_amp(2:end-1) = 2.*airTemp_amp(2:end-1);
% normalize
normalizationFactor = T/(N^2);
swTemp_amp = swTemp_amp.*normalizationFactor;
airTemp_amp = airTemp_amp.*normalizationFactor;
% average multiple segments
swTemp_mean = mean(swTemp_amp,2);
airTemp_mean = mean(airTemp_amp,2);
% get an uncertainty estimate
nu = 2*M; % DOF = 2*number of segments
err_high_sw = nu/chi2inv(0.05/2,nu);
err_low_sw = nu/chi2inv(1-0.05/2,nu);
ratio_chi2_sw = err_high_sw/err_low_sw;
err_high_air = nu/chi2inv(0.05/2,nu);
err_low_air = nu/chi2inv(1-0.05/2,nu);
ratio_chi2_air = err_high_air/err_low_air;


frequency = (0:length(swTemp_mean)-1)/(2*length(swTemp_mean)*t_diff_mean)% divided by↙
total time
figure('name','Spectra of Seawater and Air Temperatures');
subplot(2,1,1)
loglog(frequency,swTemp_mean,'-b', [10 10],[err_low_sw err_high_sw]*0.0001,'-r');
grid on;
xlabel('\fontsize{14}cycles per day')
ylabel('\fontsize{14}\circC^{2}/cpd')
title('\fontsize{16}Spectrum of Seawater Temp');
```

```
legend('\fontsize{12}SW Temp','\chi^{2}-computed Uncertainty');
subplot(2,1,2)
loglog(frequency,airTemp_mean,'-r', [10 10],[err_low_air err_high_air]*0.01,'-b');
grid on;
xlabel('\fontsize{14}cycles per day')
ylabel('\fontsize{14}\circC^{2}/cpd')
title('\fontsize{16}Spectrum of Air Temp');
legend('\fontsize{12}Air Temp','\chi^{2}-computed Uncertainty');

% For full record:
% nyquist = 1 cycle every 2 minutes (0.5 cycle per minute)
% or 1 cycle/(2/1440) = 720 cpd
% lowest frequency can resolve: 1 cycle per length of record
% length of record = 615727 minutes, so fundamental freq = 1/(615727) 1/mins
% in days: length of record = 427.588 days, so fundFreq = 1/427.588 1/days
% frequency resolution = 1/T = 1/427.588 cpd = 0.0023 cpd

% For 16-day chunks:
% nyquist = 1 cycle per 2 minutes (0.5 cycle per minute)
% or 1 cycle/(2/1440) = 720 cpd
% lowest frequency can resolve: 1 cycle per length of record
% length of record = 16 days, fundamental freq = 1/16 cpd
% frequency resolution = 1/T = 1/16 cpd = 0.0625 cpd

% drop in y per order of magnitude increase in x for both plots early
% drops 1/2 order of magnitude in y for one order magnitude increase in x
% spectral slope of f^(-1/2)
% then after 100 cpd, air temp plot drops off more quickly, with slope of
% f^-1

%% show a variance preserving version of your spectra

FswTemp_mean = frequency'.*swTemp_mean;
FairTemp_mean = frequency'.*airTemp_mean;


figure('name','Variance-preserving Spectra of Seawater and Air Temperatures');
subplot(2,1,1)
loglog(frequency,FswTemp_mean,'-b', [10 10],[err_low_sw err_high_sw]*FswTemp_mean(100),
'-r');
grid on;
xlabel('\fontsize{14}cycles per day')
ylabel('\fontsize{14} \circC^{2}')
title('\fontsize{16}Variance-Preserving Spectrum of Seawater Temp');
legend('\fontsize{12}SW Temp','\chi^{2}-computed Uncertainty');
subplot(2,1,2)
loglog(frequency,FairTemp_mean,'-r', [10 10],[err_low_air err_high_air]*FairTemp_mean
(100), '-b');
grid on;
xlabel('\fontsize{14}cycles per day')
ylabel('\fontsize{14} \circC^{2}')
title('\fontsize{16}Variance-Preserving Spectrum of Air Temp');
legend('\fontsize{12}Air Temp','\chi^{2}-computed Uncertainty');

%% compute the autocovariance of your data

% unsegmented full record:
N_Ac = floor(length(time)/8)*8;% use some number of datapoints divisible by 8
```

```
% AcSwTemp = xcorr(swTemp,swTemp)/max(xcorr(swTemp,swTemp));
% AcAirTemp = xcorr(airTemp,airTemp)/max(xcorr(airTemp,airTemp));
AcSwTemp = xcov(swTemp,swTemp,'unbiased');
AcAirTemp = xcov(airTemp,airTemp,'unbiased');
mean_AcSwTemp = AcSwTemp; %mean(AcSwTemp,2); % don't need to take mean here because ↙
you're not using segments
mean_AcAirTemp = AcAirTemp; %mean(AcAirTemp,2);
fmean_AcSwTemp = fft(mean_AcSwTemp((N/4):(N*3/4)+1)); % take middle of record
fmean_AcAirTemp = fft(mean_AcAirTemp((N/4):(N*3/4)+1)); % take middle of record
% take first half of record
fmean_AcSwTemp2 = fmean_AcSwTemp(1:(length(fmean_AcSwTemp)/2),:);
fmean_AcAirTemp2 = fmean_AcAirTemp(1:(length(fmean_AcAirTemp)/2),:);

figure('name','Spectra of Seawater and Air Temperatures Calculated by Autocovariance');
frequencyAc = (0:length(fmean_AcSwTemp2)-1)/(2*length(fmean_AcSwTemp2)*t_diff_mean); %↙
divided by total time
subplot(2,1,1)
loglog(frequencyAc,abs(fmean_AcSwTemp2),'-b')
title('\fontsize{16}Seawater Temp Spectra Calculated by Autocovariance');
legend('FFT of averaged autocovariance of seawater temp data')
xlabel('\fontsize{14}lag (days)')
ylabel('\fontsize{14} \circC^{2}')
subplot(2,1,2)
loglog(frequencyAc,abs(fmean_AcAirTemp2),'-r')
title('\fontsize{16}Air Temp Spectra Calculated by Autocovariance');
legend('FFT of averaged autocovariance of air temp data')
xlabel('\fontsize{14}lag (days)')
ylabel('\fontsize{14} \circC^{2}') % but this should probably be R, so correlation↙
(between 0 and 1)
```