Introduction to Micrometeorology
Assignment 3 - Turbulence and Spectra

Jaime Liew s141777
Young Kwon s163075

1 Introduction

This assignment focuses on spectral analysis of time series. Up until now, the main method of analyzing turbulence in wind was through basic statistical analysis in the time domain. Spectral analysis is performed in the frequency domain, and is useful for determining the periodic behaviour of wind, and the energy content available at each frequency.

2 Spectra and the Discrete Fourier Transform

The power spectral density of a time series shows the energy content of a signal at each frequency. These frequencies can often be related to specific time scales. For example for a spectrum covering a year's worth of data, higher frequencies are associated with the variations in wind speeds that happen within minutes while the lower frequencies to the left of a spectrum are related to the variations that happen within days. To obtain the power spectrum of a continuous time series, the Fourier transform of the signal's autocovariance, $R_X(t)$, is taken:

$$S_X(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R_X(\tau) e^{-i\omega\tau} d\tau$$

where ω is the angular frequency in radians per second. As the autocovariance must be calculated, this method for finding the power spectrum is computationally expensive. An equivalent way of computing the spectrum is:

$$S_X(\omega) = \frac{1}{2\pi T} \left| \int_0^T X(\tau) e^{-i\omega\tau} d\tau \right|^2$$

This method does not require the autocovariance, saving computational power. This assignment deals with discrete time series, which has an equivalent expression:

$$S_X(\omega) = \frac{1}{2\pi N f_s} \left| \sum_{n=0}^{N-1} X_n e^{-i\frac{2\pi l n}{N}} d\tau \right|^2$$

where the sum term is the discrete Fourier transform (DFT). The discrete Fourier transform can be calculated using the fast Fourier transform algorithm (FFT), which is able to reduce the computational time of the transformation from $\mathcal{O}(n^2)$ time to $\mathcal{O}(n \log n)$ time. It is worth noting that the spectrum can only be calculated up to half of the sampling frequency, $f_s = 1/\Delta t$, which is also known as the Nyquist frequency, f_N. Since the Fourier transform returns a symmetric spectrum from $-f_N$ to f_N, only the positive half is considered in this assignment. To conserve spectrum power, the one sided spectrum is doubled in all figures.

2.1 Log Smoothing Algorithm

Looking at Figure 2, the blue lines represent the spectral power density of a two hour wind speed data set. As can be seen from the graph, the fluctuations of the spectrum increase dramatically as frequency increases. To mitigate this problem, the spectrum is filtered by dividing the spectrum into logarithmically spaced bins, and representing the logarithmic midpoint as the mean of each bin. For logarithmic spacing in base 10, the logarithmic mid point of a bin is located $10^{-\frac{1}{2}} \approx 31.6\%$ along the length of the bin interval. These smoothed spectral estimates are shown in orange in Figure 2. 15 bin divisions are made for each decade.

2.2 K Averaging

By assuming stationarity in the time series, it is possible to split a long time series into multiple smaller series and take the average of each of their power spectra. In this case, K is a predetermined value that specifies the number of divisions that are to be made:

$$S_X(\omega) = \langle S_{X,k}(\omega) \rangle$$

This strategy of K averaging helps to smooth out the graph of the spectrum, especially in the high frequency range. It should be noted that taking the spectrum of shorter time series does not change the domain of the spectrum. Instead, it reduces the spectral resolution as the number of frequency bins in the spectrum is equal to the length of the time series signal. Information can be lost for large values of K as the series length is less, and therefore the spectrum has a lower spectral resolution.

3 2 Hour Sprogøs Data Set

Wind velocity data is collected over a 2 hour period with a sampling frequency of $f_s = 10Hz$. As seen in Figure 1, there are a number of large outlier values, which were omitted from the analysis. Ideally, data points would not be removed from a time series when performing spectral analysis as it introduces discrepancies in the sample spacing. However, this effect is considered negligible due to the large number of data points.

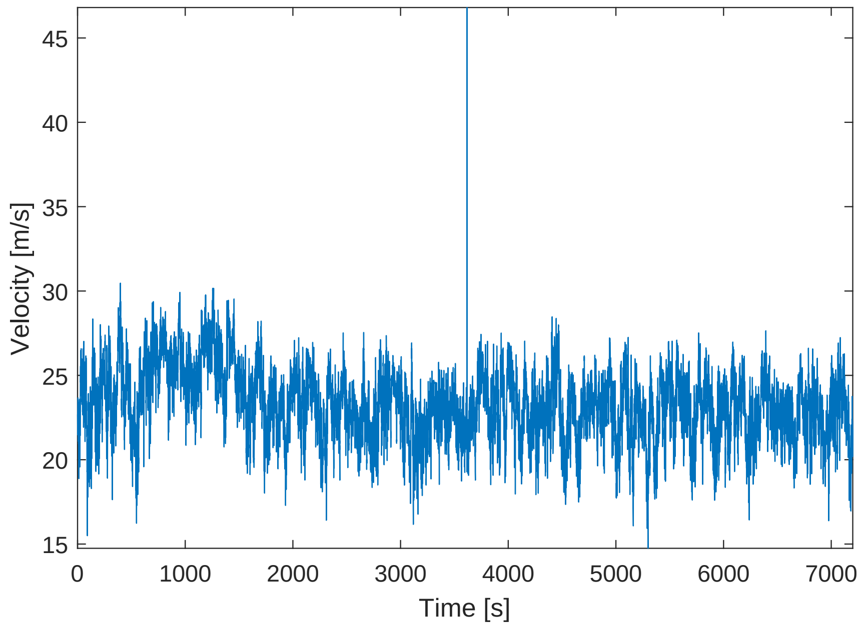

Figure 1: Time series data measured at the island of Sprogø.

The spectrum is generated for the time series, and the logarithmic smoothing algorithm is used applied. Both are plotted in Figure 2.

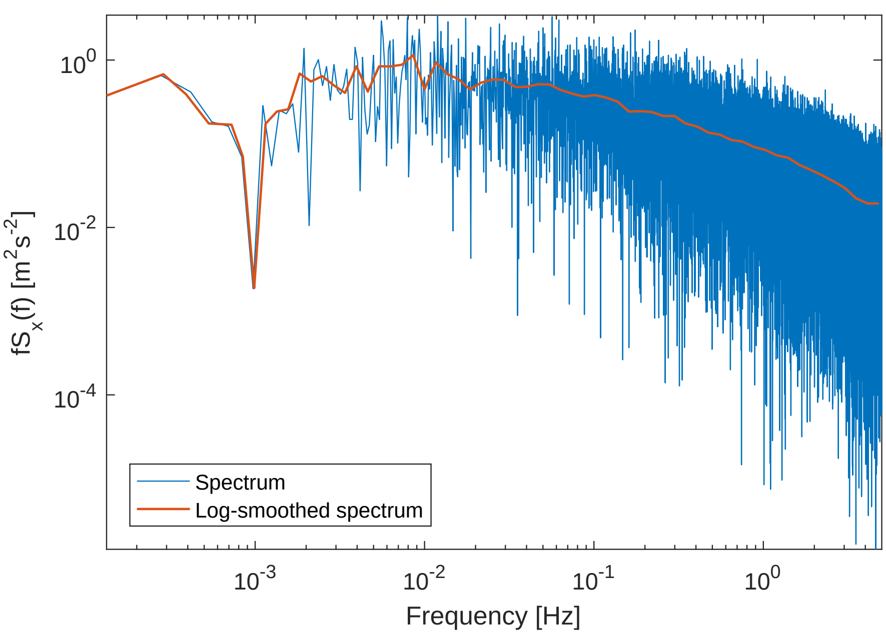

Figure 2: Spectral power density of short term wind speed time series.

The fluctuations in the spectrum can be mitigated by splitting the time series into K parts, and averaging their spectra. Figure 3 shows how the spectrum (blue line) tends to converge with larger values of K. Although higher K values produce a smoother spectrum, it also loses information as the spectral resolution decreases.

The spectrum is validated by comparing the area under the spectrum to the variance. If the spectrum calculation is consistent, then it should satisfy:

$$\int_{-\infty}^{\infty} S_X(f)df = \sigma_X^2$$

Table 1 shows how these quantity vary for different values of K, as well as the area under the log-smoothed spectra. The true value for the variance is calculated to be $\sigma_X^2 \approx 4.129$. It is clear that the K-average spectra maintain the correct variance properties as the area remains relatively constant. The log-smoothed curves do not remain consistent, however. This is due to the fact that the log-smoothing takes the average at the logarithmic midpoint, and it therefore misses the first points of the spectrum which contains a large amount of energy. The algorithm would need to be modified to add these initial points. K-averaging produces a converging spectrum while maintaining the variance properties of the spectrum.

K	1	10	50	100
Spectrum Area	4.129	4.130	4.133	4.136
log-smoothed Spectrum Area	3.85	3.034	2.276	1.736

Table 1: Area under spectra and log-smoothed spectra for various values of K.

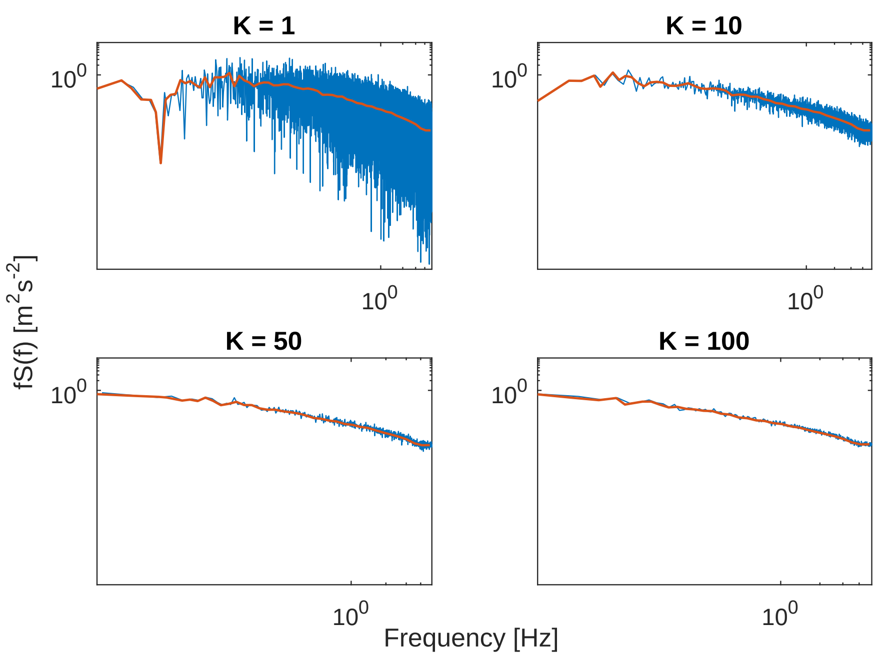

Figure 3: Spectral power density of short term wind speeds for varying values of K.

Interestingly, the spectrum closely resembles Kolmogorov's 5/3 law. This can be seen in Figure 4 as the slope of the loglog spectrum tends to $-5/3 = -1.\bar{6}$. Kolmogorov's 5/3 law states that energy per length scale is a function of the eddy length scale, and follows the power law $S_X(k) \propto k^{-5/3}$. Where $S_X(k)$ is the spacial spectrum, and $k = 1/L$ where L is the length scale of the turbulence. Although Kolmogorov's 5/3 law applies to length scales, it can be shown that length scales and time scales are proportional if Taylor's frozen turbulence hypothesis is assumed.

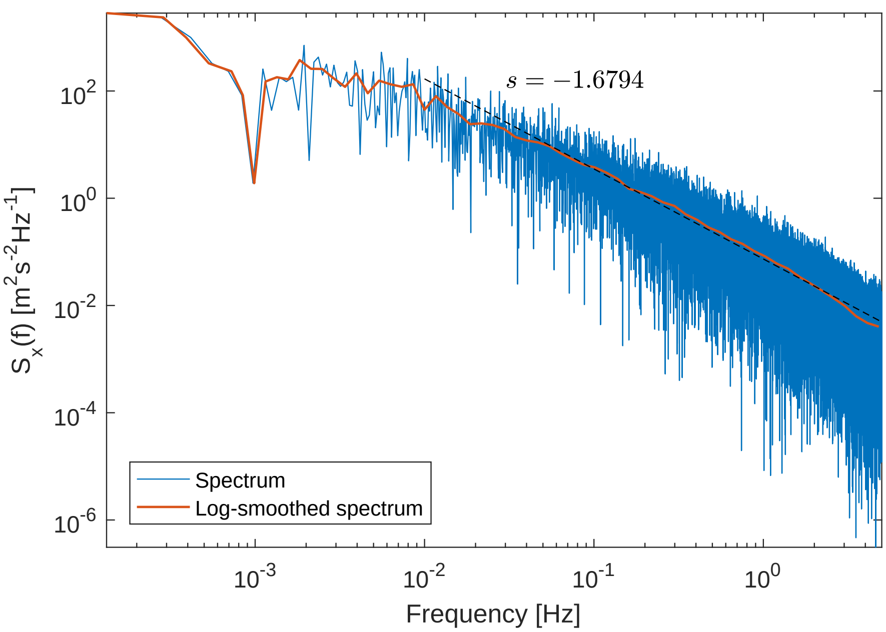

Figure 4: Spectral energy density of short term wind speed time series. For higher frequencies, the energy follows Kolmogorovs 5/3 Law.

4 21 Year Sprogøs Data Set

The data set used in assignment 1 is used to perform spectral analysis. This data set contains 21 years of wind speeds with sampling periods of 10 minutes. The data set is initially sanitized to remove invalid readings.

The spectrum and the log-smoothed spectrum is plotted in Figure 5. The frequencies of a year ($3.17 \times 10^{-8} Hz$) and a day ($1.16^{-5} Hz$) are indicated with dotted lines in order to illustrate the annual and diurnal peaks. There is a clear annual peak due to the seasons. A small diurnal peak can also be seen in the log-smoothed spectrum.

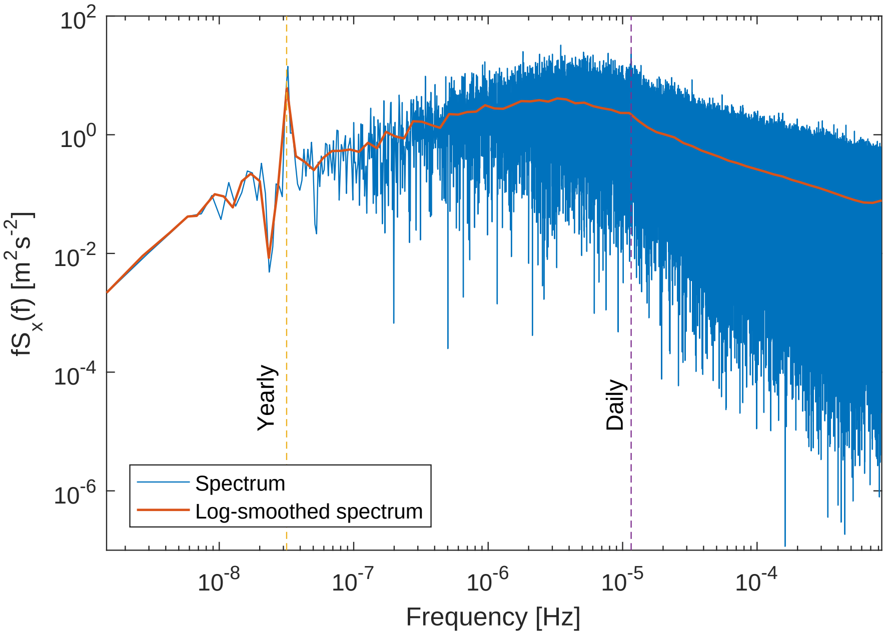

Figure 5: Spectral power density of long term wind speed time series.

From looking at Figure 5, it is clear that the power is higher in the low frequency region and tapers off in the high frequency region of the spectrum. Intuitively this makes sense. The low frequency region represents the power from the usual expected wind speeds throughout months or years of wind at this site, and the wind speeds on these larger time scales are what comprise most of the power generated by the wind turbine. In the high frequency region, the power is related to the fluctuating wind speeds that last on smaller time scales such as minutes or hours. Since this turbulent behaviour is fleeting, the power contributed by these wind oscillations are expected to be low.

5 Autoregressive Process

Spectral analysis was performed on an autoregressive process. It was established in the previous assignment that the autocorrelation of an autoregressive process is:

$$R_X = \frac{\alpha^{|t|}}{1-\alpha^2}$$

The absolute value of the time lag, t, is taken to represent the symmetric nature of the autocorrelation function. This relation can be used to find the theoretical value for the autoregressive spectrum. The spectrum as a function of frequency, f, which has units of Hertz is:

$$S_X(f) = \int_{-\infty}^{\infty} R_X(\tau)e^{-i2\pi f\tau}d\tau$$

Substituting the expression for R_X into this equation gives:

$$S_X = \int_{-\infty}^{\infty} \frac{\alpha^{|\tau|}}{1-\alpha^2} e^{-i2\pi f\tau} d\tau$$

This expression can be decomposed into the co-spectrum and quadrature spectrum given Euler's formula, which states $e^{ix} = cos(x) + isin(x)$.

$$S_X(f) = \int_{-\infty}^{\infty} \frac{\alpha^{|\tau|}}{1-\alpha^2} cos(2\pi f\tau) d\tau - i \int_{-\infty}^{\infty} \frac{\alpha^{|\tau|}}{1-\alpha^2} sin(2\pi f\tau) d\tau$$

Since R_X is an even function, the second integral is odd. Since the bounds of integration are symmetric, this integral evaluates to zero. Similarly, the first integral can be expressed as a one sided integral as it is even. What is left is a purely real expression for the spectral power density, which is the case for all autospectra of real time series:

$$S_x(f) = \frac{2}{1-\alpha^2} \int_0^{\infty} \alpha^{\tau} cos(2\pi f\tau) d\tau$$

Given that

$$\int_0^{\infty} a^x cos(bx) dx = -\frac{ln(a)}{ln^2(a) + b^2} \qquad \text{for} \quad a < 1$$

It can be shown that

$$S_X(f) = \frac{-2ln(\alpha)}{(1-\alpha^2)(ln^2(\alpha) + 4\pi^2 f^2)}$$

The spectral power density is found for autoregressive processes with varying values of α (Figure 6), and are compared with the theoretical spectrum derived above. The sampling frequency is assumed to be $f_S = 1Hz$, and the signal length is $n = 10^5$.

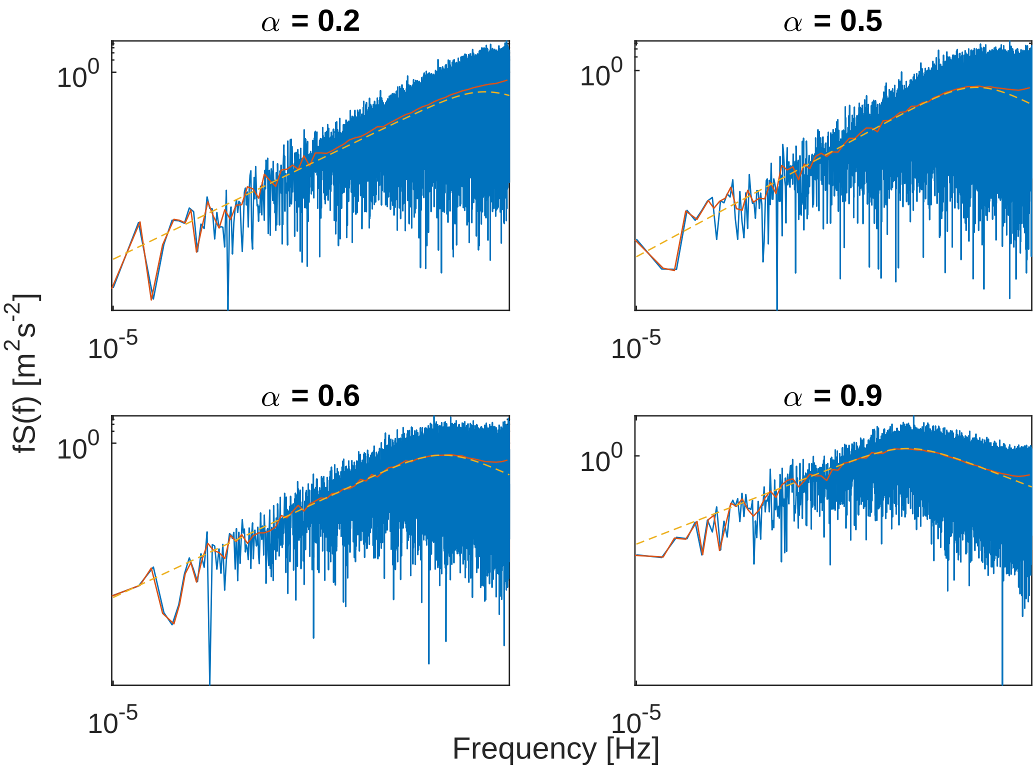

Figure 6: Spectral power density of autoregressive processes with varying α.

The calculated spectrum closely follows the theoretical spectrum function. Comparing the plots in Figure 6 to the spectra in Figures 2 and 5, it is clear that the autoregressive process does not closely match the wind speed data spectra. Wind speeds tend to have a drop in power density for higher frequencies, whereas the autoregressive process continues to increase for the majority of frequencies.

6 Conclusion

In this assignment, three different signals of wind speeds were evaluated using power spectrum analysis. This process involved converting the time series of each data set into the frequency domain by using the discrete Fourier transform which was implemented by using the fast Fourier transform (FFT) in the code. The validity of using the FFT to find the spectrum was proved by how close the values of the integrals of the power spectra were to the variance of the time series. It was possible to see how useful a tool spectrum analysis can be for breaking down the power contribution from the wind speeds at each time scale. For the autoregressive process, the theoretical calculation of the spectrum was confirmed to be equivalent to the Fourier transform of the autocovariance function by showing that the power spectrum of the time series using FFT converged towards the theoretical values. Like in the second assignment, it was shown again that an autoregressive process is not an accurate method of approximating wind speed data as the spectrum of the autoregressive process was quite different to that of the measured time series.

Appendix A Spectrum Function

```
function [freq, spectra] = makeSpectrum(x,fs)
%Takes a time series, x, with sampling frequency, fs, and
%generates the power spectral density.


spectra = abs(fft(x)).^2/(length(x)*fs);
spectra = 2*spectra(1:length(spectra)/2+1);
freq = linspace(0,fs/2,length(spectra));
end
```

Appendix B Log-Smoothing Function

```
function [specSmooth, freqSmooth] = logSmoothing(x,y,nbins)
%takes power spectral data, x (frequencies) and y (spectrum)
%and filters the spectrum by taking the average over logarithmically
%spaced frequency bins.

bins = logspace(-10,2,nbins*12);
for i = 1:length(bins)-1
    binPoints = (x>bins(i)) & (x<bins(i+1));
    specSmooth(i) = mean(y(binPoints));
    freqSmooth(i) = bins(i) + 10^(-0.5)*(bins(i+1)-bins(i));
end


freqSmooth = freqSmooth(~isnan(specSmooth));
specSmooth = specSmooth(~isnan(specSmooth));

end
```

Appendix C K-Averaging Spectrum Function

```
function [Kfreq, Kspectra] = makeMeanSpectrum(x,fs,K)
%Takes a time series, x, with sampling frequency, fs, and
%generates the mean spectrum of the signal split into K parts


N = floor(length(x)/K)

x = x(1:floor(length(x)/N)*N);
Kspectra = zeros(N,1);

for i=1:K
    Kspectra = Kspectra + ...
```

```
        1/K*1/(N*fs)*abs(fft(x(((i-1)*N+1):i*N)).^2);

end
Kspectra = 2*Kspectra(1:length(Kspectra)/2+1);
Kfreq  =linspace(0,fs/2,length(Kspectra));
end
```

Appendix D Main Script

```
 %Micrometeorology
%Assignment 3 - Power spectral density
%Jaime Liew, Young Kwon

%% 2 Hour Sprogo Dataset
data1 = load('sonic1u10Hz.dat');


figure
plot(0:0.1:0.1*(length(data1)-1),data1)
axis('tight')

xlabel('Time [s]')
ylabel('Velocity [m/s]')
PlotToFileColorPDF(gcf,'figTime.pdf',15,10);

data1 = data1(data1<32);
mean1 = mean(data1);
data1 = data1 - mean1;

%% Make the spectrum.
fs1 = 10; %Hz - Sampling frequency

[freq1, spectra1] = makeSpectrum(data1,fs1);

Var1 = var(data1)
trapz(freq1,spectra1);

nbins = 15;
[specSmooth1, freqSmooth1] = logSmoothing(freq1,spectra1,nbins);
P = polyfit(log(freqSmooth1(30:end)),log(specSmooth1(30:end)),1);

%% Splitting the signal into bins of size K
Ks = [1, 10, 50,100]; %split signal into K groups
Kspec1 = {};
Kfreq1 ={};
KspecSmooth = {};
KsfreqSmooth = {};
meanVar = zeros(length(Ks),1);
area = [];
areaSmooth = [];
for i =1:length(Ks)
    [Kfreq1{i} Kspec1{i}] = makeMeanSpectrum(data1,fs1,Ks(i));
```

```
    [KspecSmooth{i} KfreqSmooth{i}] = logSmoothing(Kfreq1{i},Kspec1{i},15);


    area(i) = trapz(Kfreq1{i}, Kspec1{i})
    areaSmooth(i) = trapz( KfreqSmooth{i}, KspecSmooth{i})

end

%% Plot

close all
figure
loglog(freq1,freq1.*spectra1')
hold on

loglog(freqSmooth1,freqSmooth1.*specSmooth1,'linewidth',1)

xlabel('Frequency [Hz]')
ylabel('fS_x(f) [m^2s^{-2}]')
axis('tight')
legend('Spectrum','Log-smoothed spectrum','location','southwest')
PlotToFileColorPDF(gcf,'figSpec1.pdf',15,10);


figure

for i=1:length(Ks)
    subplot(2,2,i)
    loglog(Kfreq1{i},Kfreq1{i}.*Kspec1{i}')
    hold on
    loglog(KfreqSmooth{i},KfreqSmooth{i}.*KspecSmooth{i},'linewidth',1)
    title(['K = ' num2str(Ks(i))])
    xlim([0,fs1/2])
    ylim([1e-6,10])

end

suplabel('Frequency [Hz]','x',[.08 .12 0.85, 0.85])
suplabel('fS(f) [m^2s^{-2}]','y',[0.12 0.08 0.85, 0.85])
PlotToFileColorPDF(gcf,'figKspecs.pdf',15,10);

%%
figure
loglog(freq1,spectra1')
hold on


loglog(freqSmooth1,specSmooth1,'linewidth',1)
LINEX = linspace(0.01,fs1/2,10);
LINEY = exp(P(2))*LINEX.^(P(1));
loglog(LINEX,LINEY,'--k')
text(3e-2,150,['$s=' num2str(P(1)) '$'],'interpreter','latex','fontsize',11)

legend('Spectrum','Log-smoothed spectrum','location','southwest')
xlabel('Frequency [Hz]')
ylabel('S_x(f) [m^2s^{-2}Hz^{-1}]')
```

```
axis('tight')
PlotToFileColorPDF(gcf,'fig53Law.pdf',15,10);

%%
%%
%% 21 year Sprogo Dataset


%Sprog data
data2 = load('SprogData.mat');
data2 = data2.data(:,2);
data2 = data2(data2 < 40);
mean2 = mean(data2);
data2 = data2 - mean2;

% Make the spectrum.
fs2 = 1/(10*60); %Hz - Sampling frequency

[freq2, spectra2] = makeSpectrum(data2,fs2);

Var2 = var(data2)
trapz(freq2,spectra2)

nbins = 15;
[specSmooth2, freqSmooth2] = logSmoothing(freq2,spectra2,nbins);


% Plot


figure
loglog(freq2,freq2.*spectra2')
hold on

loglog(freqSmooth2,freqSmooth2.*specSmooth2,'linewidth',1)

fYear = 1/(60*60*24*365);
fDay = 1/(60*60*24);

xlabel('Frequency [Hz]')
ylabel('fS_x(f) [m^2s^{-2}]')
axis('tight')
plot([fYear, fYear],[1e-7,100],'--')
text(fYear- 1e-8,1e-5, 'Yearly','rotation',90)

legend('Spectrum','Log-smoothed spectrum','location','southwest')
plot([fDay, fDay],[1e-7,100],'--')
text(fDay- 3e-6,1e-5, 'Daily','rotation',90)
PlotToFileColorPDF(gcf,'figSpec2.pdf',15,10);



%% Autoregressive process
as = [0.2, 0.5, 0.6, 0.9];

figure
```

```
for i = 1:length(as)
    a = as(i);
    data3 = autoRegProcess(100000,a);
    mean3 = mean(data3)
    data3 = data3 - mean3;

    fs3 = 1; %Hz - Sampling frequency
    [freq3, spectra3] = makeSpectrum(data3,fs3);
    Var3 = var(data3);
    trapz(freq3,spectra3);

    nbins = 15;
    [specSmooth3, freqSmooth3] = logSmoothing(freq3,spectra3,nbins);

    specTheo = -4*log(a)./((1-a^2)*(log(a)^2 + 4* pi.^2 * freq3.^2));

    subplot(2,2,i)
    loglog(freq3,freq3.*spectra3')
    hold on
    loglog(freqSmooth3, freqSmooth3.*specSmooth3);
    loglog(freq3,freq3.*specTheo,'--');
    title(['\alpha = ' num2str(a)])
    axis('tight')
end
suplabel('Frequency [Hz]','x',[.08 .12 0.85, 0.85])
suplabel('fS(f) [m^2s^{-2}]','y',[0.12 0.08 0.85, 0.85])

PlotToFileColorPDF(gcf,'figSpec3.pdf',15,10);
```