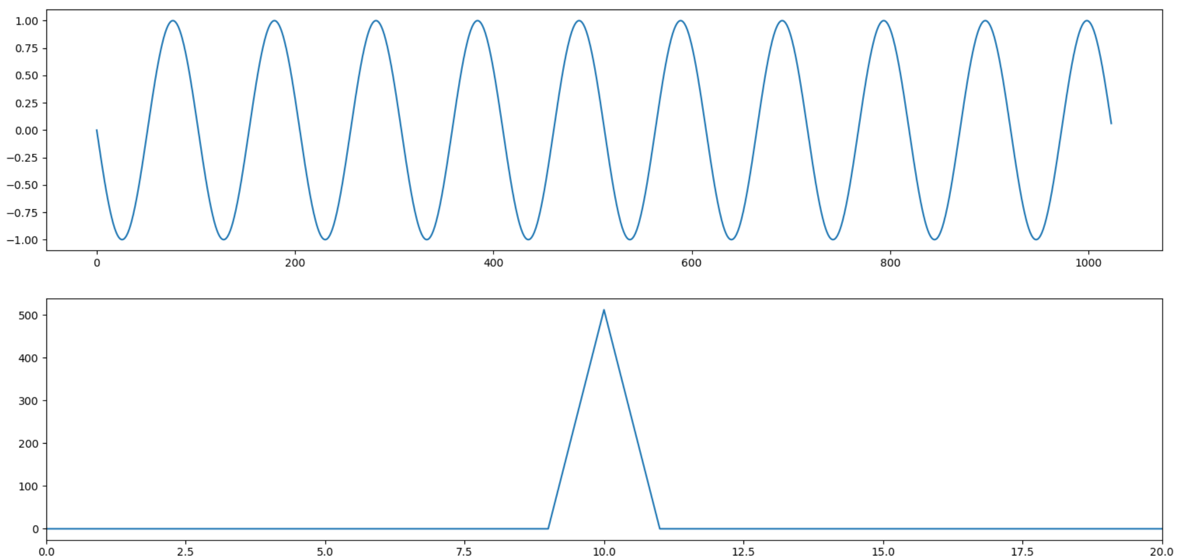
Homework 6 Solutions: *Hirak Bandyopadhyay*

Problem 1:

In this problem we have to run the fft_test.py. The output is stored in fft_test.png file.

This is the output we get. As we see in the file that the formulae is given by,



$y(x) = \sin(\frac{-2\pi f x}{N})$ and hence we can easily understand that the angular frequency is given by $\omega = \frac{2\pi f}{N}$. So the period of the oscillation is given by, $T = \frac{2\pi}{\omega} = \frac{N}{f}$. In our case N = 1024 and f = 10, so clearly, T = 102.4. In this case N corresponds to the binning to the linear frequency which is the fourier index and f corresponds to the total frequency available. Clearly, the linear frequency is given by $n = \frac{f}{N}$ in this case, so given fourier index, we can set f to get a particular linear frequency.

Problem 2:

In this problem, we have to smooth the power spectrum. In order to do that we use the fft_sunspot_demonstration.py program and try to modify it. First, we need to get the actuat data of co2 emission. We can observe that the decimal year and the intrapolated CO2 emission is in column 3 and 5 in the file we are looking at. In this way we extract the data and put it in the xinput and yinput arrays. Now, in order to do the fft, we need to meke the data size as 2^n so we use padding of the data. Now, once that is done, we define an array called ps to get the raw power spectrum of the data. Now, once we get that, we observe that the data is not of the same length of 2^n so, we redo the padding and also, to smooth the power spectrum we use the Hanning window in order to clean the sidelobes. Once that is done, we take the fft of ps and use maxfreq = 50 to smooth the data, we create a nyquest range and make that zero inside the data. Now, we take the inverse fourier transform of the data. Now, in order to recover our actual data we need to undo the windowing and padding. So, first we undo the Hanning window. We see that is the window function is not evaluated to be zero inside the range, we can just divide it to get back the normal form of the data, and clearly, by cutting of the padding we can retrive the main data back. Clearly when this procedure is done, we take the real value of the smoothed power spectrum and plot it with raw power spectrum. The output is saved as problem2a.png. We see that almost all the sidelobes are gone and we can get a clean power spectrum.

Now in the second part of the preoblem we have to clear the jitters in fourier domain. To reduce high frequency noise, we reduce the value of maxfreq to 10 to eleminate almost all the high requency noise and plot the smoothed power spectrum along with the real data. I also have plotted the clean and raw spectrum and the magnified version of it so that we can compare the actual data to the less noise one. The output is saved in problem2b.png.

Problem 3:

In this problem, we first have to subtract the linear trend of the data from the actual given dataset. So, I use gnuplot to find the linear trend in the data. I fit the data to $y = ax + b$ and from gnuplot find the coefficients. The output is give below. The program is named as “linear.gp”

#Final set of parameters Asymptotic Standard Error

#======================= ==========================

#a = 1.54159 +/- 0.008688 (0.5636%)

#b = -2711.84 +/- 17.27 (0.637%)

Once we get that, I use excel to create the y data from the file and save it to linear.dat. Now, also, as I need to subtract the linear data from the actual one, I subtract it from the raw data and save it to another file as nonlinear.dat. Now, I need to plot fourier tarnsform of the data. So I take the fourier transform of the raw data and save it to fftY. Now I take the real of thefourier transform data and plot it along with the raw non linear data. The output is saved as problem3a.png. Now, once that is done, I need to again clear up the jitters from the non linear data. In order to do that we agan use the technique of smoothing and then take the ifft of the data named as ysmooth and then take the real value. Now to add the linear trend back to the data we read the file linear.dat and take all the Co2 ppm values to another array named new_yinput. The I use a for loop to add all the elements of that array to the ysmoothedreal values we got from the non linear data after clearing out the noise. So, our data is reconstructed. To compare this data with the raw one, I real the raw file and take values as x_new and y_new . Then I plot both tha data simultaneously to compare them. The output is stored in problem3b.png

I observe that the data is significantly smooth. This gives me a good hint that the noise cancellation can be done in this process very effectively.
