PHYS 331 – Introduction to Numerical Techniques in Physics

Homework 11: Fourier Transforms

Due Friday, April 20, 2018, at 11:59pm.

Problem 1 – Fourier Series of a Sawtooth Wave (20 points). Recall that the continuous Fourier transform, defined over an infinite range:

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{ikx} dx$$
$$= \int_{-\infty}^{\infty} f(x)\cos(kx)\,dx + i\int_{-\infty}^{\infty} f(x)\sin(kx)\,dx$$

is very similar in form to the discrete Fourier decomposition over an interval (-L, L):

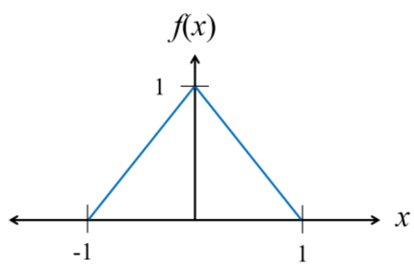
$$A_k = \frac{1}{L}\int_{-L}^{L} f(x)\cos(\pi kx/L)\,dx$$
$$B_k = \frac{1}{L}\int_{-L}^{L} f(x)\sin(\pi kx/L)\,dx \qquad (1)$$

where

$$f(x) \approx \frac{A_0}{2} + \sum_{k=1}^{k_{max}} A_k \cos(\pi kx/L) + B_k \sin(\pi kx/L), \qquad (2)$$

and the approximation is more accurate as higher orders of the sum, k_{max}, are included in the computation. The decomposition of Eq. (2) is also called the Fourier series. In the Fourier series, the basis set of cosines and sines have discrete frequencies (as opposed to sines and cosines of continuous frequencies for the continuous Fourier transform). Another difference is that, in the Fourier series, the function $f(x)$ is assumed to be periodic beyond the (-L, L) interval, as we will see in this problem.

Let us consider a sawtooth function as shown in the diagram below over the interval (-1, 1).



(a) Explicitly write the functional form of $f(x)$ based on the plot.

(b) Using your answer from part (a) and Eq. (1) and (2), solve for the coefficients A_0, A_k and B_k as a function of arbitrary k. Hint: Use symmetry to your advantage; consider rewriting the integrals using symmetry first. For full credit show your reasoning.

(c) Using your answers from part (b), write a function, `funcSawtooth(kmax)`, that inputs the integer `kmax` > 0, and displays a plot of the resulting Fourier series representation of the sawtooth function summed from k=0 to `kmax`. Features should include:

- Sufficient comments so that the grader can tell what you are doing.
- Produces an overlay plot of the Fourier series with the original sawtooth function; a legend should be used to label the curves. Both should be shown as smooth curves over the interval (-1,1).
- The plot should be sufficiently sampled in x to produce a smooth curve given the value of `kmax` used. As such, a mesh size that is dependent upon the value of `kmax` should be determined as part of your function.

- The function does not use any global variables. However, you may write additional functions that are called by `funcSawtooth` that input and output variables as needed.
- Write code that executes `funcSawtooth` for `kmax` values of 1, 2, 3, and 30. Describe your observations. Look carefully and compare and contrast the results.
- This function does not need to return any values (although you may do so for troubleshooting).

(d) Make a slightly modified function, `ModFuncSawtooth`, that computes and plots the Fourier series over the interval (-3,3), but using the same A_k and B_k values as derived for (-1,1). For simplicity, you can remove the overlay with the original function which is only defined from (-1,1). Show plots for `kmax`=1, 3, and 30. What do you observe? Is this what you expected?

Extra Credit (+5 points): Derive the continuous Fourier Transform of $f(t) = A\cos(\omega_0 t + \varphi)$ for arbitrary A, ω_0 and φ. Then, derive the Fourier series for the same $f(t)$ over an interval equal to 1 period of $f(t)$, with the interval centered at t=0 as usual. Compare your results between the series and continuous transform.

Problem 2 – The Discrete Fourier Transform (DFT) (20 points). Implement the "straightforward" DFT discussed in class that transforms an array of N values, h_k, k=0…N-1, into an array H_n, n=0…N-1, according to:

$$H_n = \sum_{k=0}^{N-1} h_k e^{-i2\pi kn/N} \tag{3}$$

In Eq. (3) the sign in the exponential is flipped in order to match the convention used in Python, so that we can compare our outputs to the built-in Python function. (Do **not** use the "divide and conquer" Fast Fourier Transform method here because it is a lot of work to obtain the speed benefits!) Write this as a function `myDFT`.

a) Write code to generate an input array h_k populated with random numbers and with N=10. Show that the results of myDFT are identical (within machine precision) to that from `numpy.fft.fft`.

b) Write code to measure the amount of time it takes for `myDFT` to execute as a function of N using randomized input arrays, h_k. I recommend using `time.time()` to capture the machine time immediately before and after a function call to `myDFT`. Plot the execution time as a function of N on a log-log plot over at least 2 decades in N (starting with N large enough that the time interval is non-zero). What is the relevant feature of the plot, and is it what you expected? Hint: You should consider sampling N on an exponential interval for this so that it doesn't take too long to collect the timing data.

Problem 3 – Fourier Spectral Analysis of Bridge Vibrations (10 points). You receive a data set of the position of the surface of a bridge in the data file "`HW11bridge.csv`". The data has been sampled at even time intervals of 25 milliseconds. The concern of the engineers is whether there may be resonances of the bridge that may eventually lead to its collapse under high winds. While the high frequencies (>1 Hz) are likely to be disturbances from passing traffic, they particularly want to know if there is any evidence of peaks in the frequency response between 0.1 - 1 Hz.

- The absolute length scale of the measurements is not needed in this problem – the goal is to identify vibrations that have a significantly higher amplitude than the noise background. However, the noise background may, in general, be frequency-dependent.
- Since the data provided has a very large N, you won't be able to use your DFT code from problem 2 because it will take too long to compute. Use the built-in `numpy.fft.fft` function in this problem instead.
- First, plot the absolute value of the DFT of the data as a function of frequency f from DC to the Nyquist frequency. As needed, produce additional plots that zoom in on different ranges of the spectrum to examine features. Make sure plots are labelled. Make a list of all vibrations found above the noise (and their approximate peak frequencies), and determine whether the engineers should be worried about bridge resonances.
- Warning: there is a lot of data to plot! You might need to play with the line settings in matplotlib to make sure you are seeing all of the features.