Phys 256 — Homework 7

Student Name

Instructions

- You may use AI assistants to help write, test, and debug code, but you must understand your implementation and use your own scientific judgment to assess its validity. **Do not use AI to write the report.**
- Submit a PDF report containing in-line code snippets, plots, numerical results, explanations, and descriptions of relevant test cases. Submit your Python code separately.
- In your report, we are looking for you to **exercise and demonstrate your independent scientific judgment.** Thoughtful reports with careful, critical discussion of the results will receive a higher grade than those that simply run the AI-generated code and paste in the results.
- Submit each coded problem in a separate file named `p1.py`, `p2.py`, etc. You may import functions from previous problems. Include all necessary files in one ZIP archive.
- Use NumPy, SciPy, and Matplotlib where appropriate. Fix random seeds or specify them so that stochastic results can be reproduced. Label all plot axes.

Problem 1: Problem 1 (3 points)

Have a look at the particular realization of the N point forward DFT with the omitted normalization coefficient:

$$C_n = \sum_{k=0}^{N-1} y_k \exp(-i2\pi kn/N)$$

Analytically prove that the forward discrete Fourier transform is periodic, i.e., $c_{n+N} = c_n$. Note: recall that $\exp(\pm i2\pi) = 1$.

Does this also prove that $c_{-n} = c_{N-n}$? Prove your point.

Problem 2: Problem 2 (2 points)

Use proof for the previous problem relationships and show that the following relationship holds for any sample set $\{y_0, y_1, \cdots y_{N-1}\}$ which has only real values (i.e., no complex part)

$$c_n = c^*_{N-n}$$

Where * depicts the complex conjugation.

Problem 3: Problem 3 (10 points)

Load the data from the file `'hw_data_for_filter.csv'` provided at the class web page. It contains a table with y vs t data points (the first column holds the time, the second holds y). These data points are taken with the same sampling rate.

1. (1 points) What is the sampling rate?
2. (1 points) What is the lowest frequency (in Hz) in this data set which we can scientifically discuss?
3. (2 points) What is the largest frequency (in Hz) in this data set which we can scientifically discuss?
4. (2 points) Make a function which convert index n to its corresponding frequency in Hz. (Do not use library implementation from the `fft` package). Recall that it should have positive and negative frequencies. Save you function to file `'p3.py'` and submit to Gradescope.

```python
def index2freq(n, Npoints=100, samplingRate=1):
    """ Calculates frequency of the n-th component of the FFT spectrum.
        Appropriate (upper half) frequencies should be negative.
        Should be able to handle numpy array of indexes or a single number provided for 'n'.
        Returns either a number or numpy array of frequencies
    """
    return frequency_n
```

Listing 1: `p3.py` template for problem 3

5. (2 points) Calculate forward DFT of the data (use python `np.fft.fft` package) and plot the resulting **amplitude** spectrum vs **frequency** of the spectral component in Hz. Since it is the complex spectrum, you should have positive and negative frequencies for your plot.
6. (2 point) Based on calculated spectrum. Does the relationship from the problem 2 hold true?

Provide relevant listings and plots with your report.

Problem 4: (5 points)

Study the spectrum from the above data file `'hw_data_for_filter.csv'`.

1. (1 points) Name two positive frequency components of the spectrum (measured in Hz not rad^{-1}) which has the largest amplitudes.
2. (1 points) Compare the amplitude of the largest component of the spectrum to the apparent size of the signal in the time domain. Does it make sense, how would you scale it based on number of samples so it is more reasonable?
3. (1 points) Consider everything else but above 2 components of the DFT as noise. Construct a low-pass filter which will pass these two components. Plot the filter frequency representation (positive and negative frequency). Explain your choice of the filter and its parameters.

4. (2 points) Apply the filter to the data Fourier representation and calculate the inverse DFT. Plot the resulting filtered data representation and raw data points in the same plot. Does your filter completely get rid of noise? If not why is it so?

Provide relevant listings and plots with your report.