# ECE 351 - Lab 9

Khoi Nguyen
https://github.com/3khoin

4 November 2021

# Contents

# 1 Introduction

The goal of this lab was to use fast Fourier transforms in Python.

# 2 Equations

Task 1:

$$x(t) = cos(2\pi t)$$

Task 2:

$$x(t) = 5sin(2\pi t)$$

Task 3:

$$x(t) = 2cos((2\pi * 2t) - 2) + sin^2((2\pi * 6t) + 3)$$

From Lab 8:

$$x(t) = \sum_{k=1}^{\infty} \frac{4}{k\pi} sin^2(\frac{k\pi}{2}) sin(k\omega_o t)$$

# 3 Methodology

We first implemented the Fast Fourier Transform as a user-defined function with the provided code and added return values. We then plotted Task 1, 2, and 3 equations (listed in Equations) with a domain of 0 $\leq$ t ¡ 2 s. In the same subplot, we plotted the magnitude and phase of the equations using the user-defined FFT, with a sampling frequency of fs = 100. We then modified our FFT function to filter out phases of X_mag ¡ 1e-10 and replotted the equations. We created the same format of plot for the Fourier series approximation of the square wave from Lab 8 for N = 15, with a new domain of 0 $\leq$ t ¡ 16 s, and T = 8.

# 4 Results

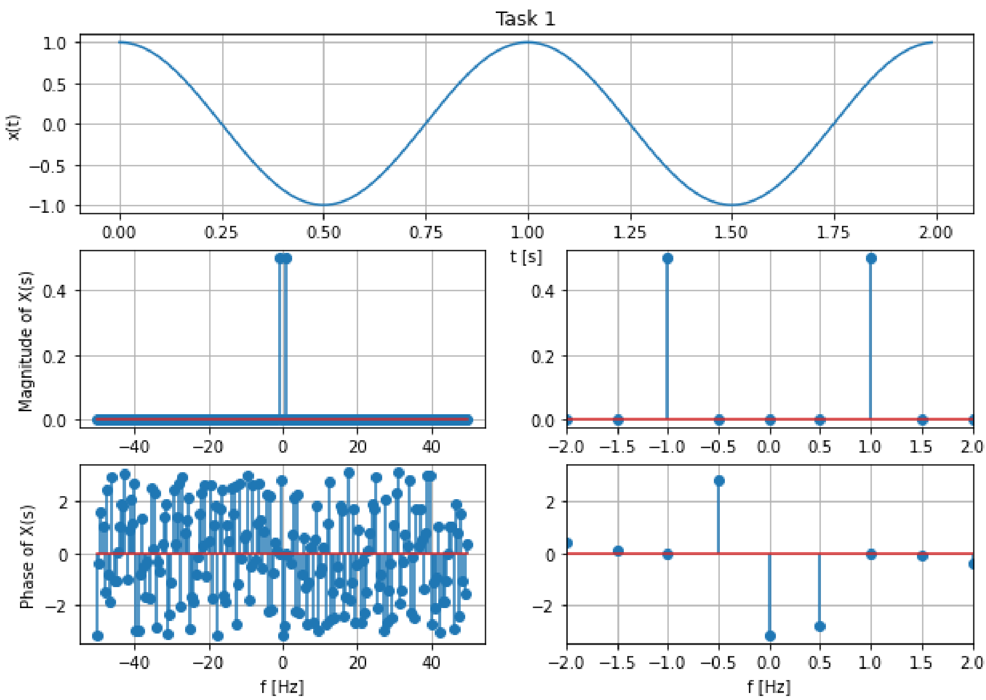
The Fast Fourier Transform user-defined function is listed down below, as well as the modified function that filters all elements of X_mag ¡ 1e-10.

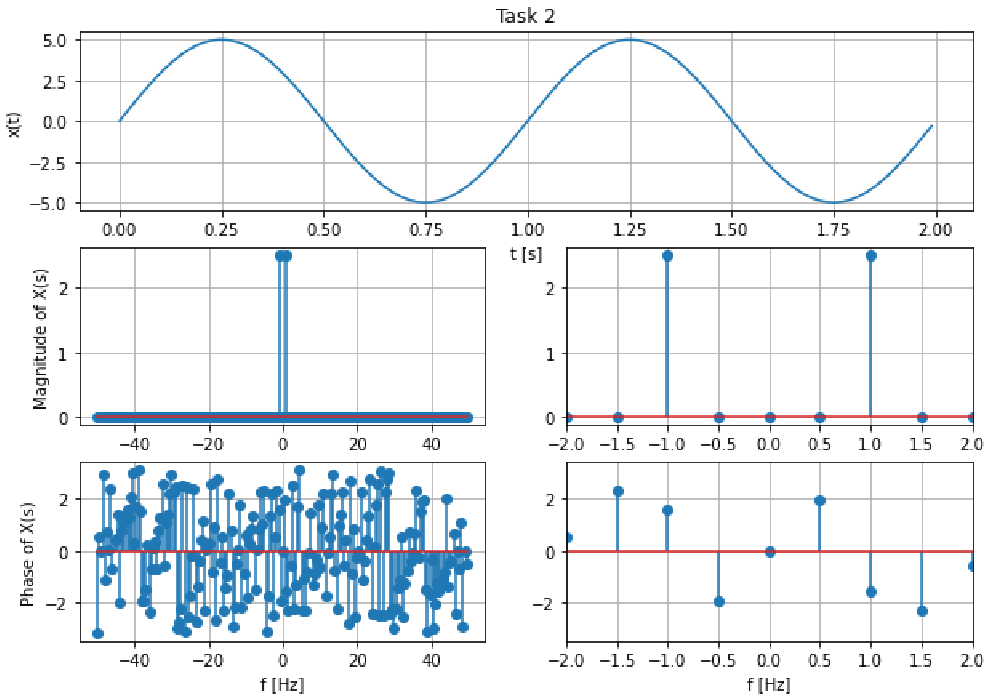
```
def fft(x):
  N = len(x)
  X_fft = scipy.fftpack.fft(x)
  X_fft_shifted = scipy.fftpack.fftshift(X_fft)

  freq = np.arange(-N/2, N/2) * fs/N

  X_mag = np.abs(X_fft_shifted)/N
  X_phi = np.angle(X_fft_shifted)
  return freq, X_mag, X_phi
```

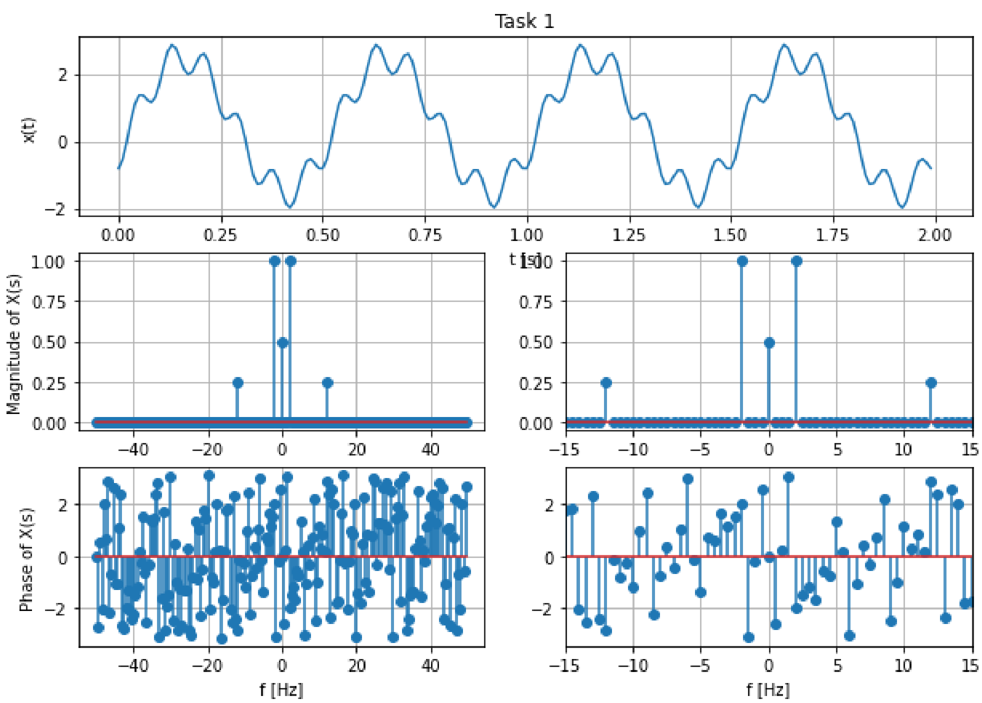
```
def fft_new(x):
  N = len(x)
  X_fft = scipy.fftpack.fft(x)
  X_fft_shifted = scipy.fftpack.fftshift(X_fft)

```

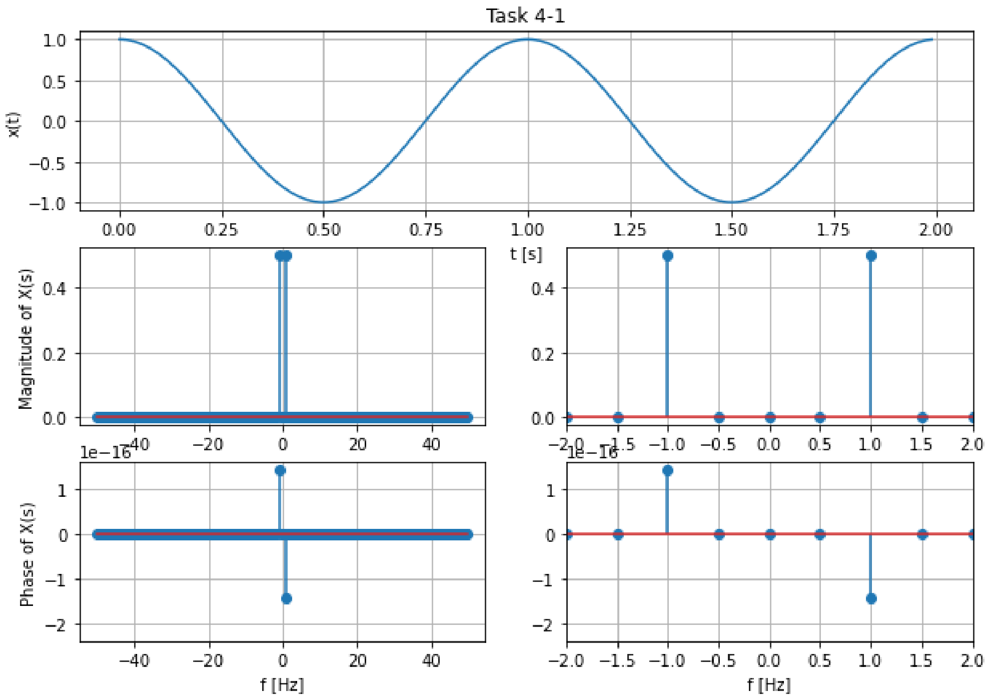
```
freq = np.arange(-N/2, N/2) * fs/N

X_mag = np.abs(X_fft_shifted)/N
X_phi = np.zeros(len(X_mag))
# New
for i in range(len(X_mag)):
  if( X_mag[i] > 1e-10 ):
    X_phi[i] = np.angle(X_fft_shifted[i])
return freq, X_mag, X_phi
```

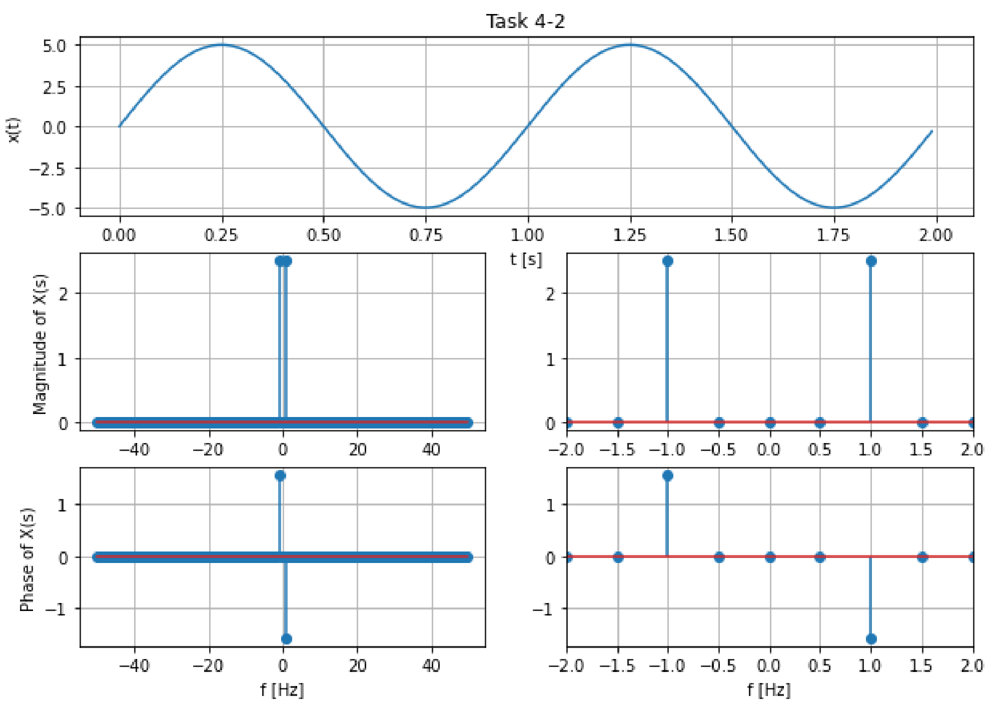
Below are the plot figures for Task 1, 2, and 3, as well as the Task 4 modified plots for the former 3 with the modified FFT function used. Finally, the Fourier series approximation of the square wave from Lab 8 is plotted with the same parameters, except with a domain of $0 \leq$ t ¡ 16 instead.



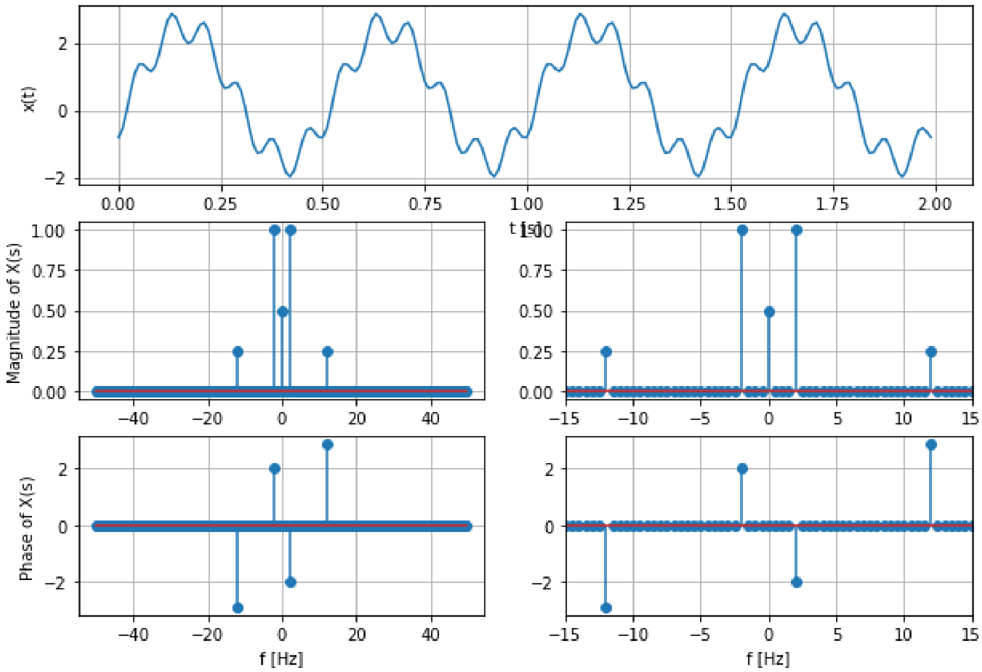

Task 2
x(t)
t [s]
Magnitude of X(s)
Phase of X(s)
f [Hz]
f [Hz]


Task 1
x(t)
t [s]
Magnitude of X(s)
Phase of X(s)
f [Hz]
f [Hz]

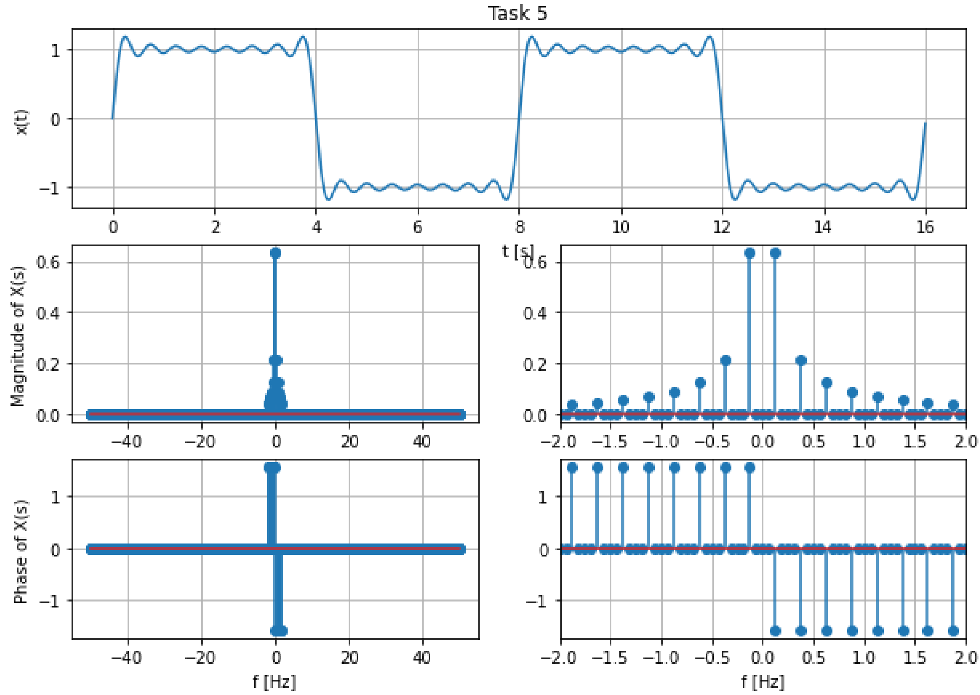

Task 4-1
x(t)
t [s]
Magnitude of X(s)
Phase of X(s)
1e−16
f [Hz]


Task 4-2
x(t)
t [s]
Magnitude of X(s)
Phase of X(s)
f [Hz]

Task 4-3




Task 5



## 5 Error Analysis

Two minor issues came to light during this lab. Firstly, a non-even time scale (eg. the maximum x value in the domain is set to 2 + step size instead of just 2) caused the graph to use the uneven intervals to create a strangely shifted and unreadable magnitude/phase plot. The values were fixed; it was realized that the upper x value being uneven made all the intervals become slightly shifted (eg. a value would be placed at 0.5 + some residue instead of just 0.5). Secondly, it had to be figured out how to plot five plots on a three-plot figure; it was found out that one could just plot the first plot with a width of two, and the others with a width of one.

## 6 Questions

1. If fs is lowered, the resolution of the graph as well as the noise in the phase plot is lowered; likewise, if increased, the graph resolution is increased, but there is also more noise in the phase plot.

2. Eliminating the small phase magnitudes allows for the viewing of only the phases directly corresponding to the important locations in the magnitude plot.

3. The Fourier transforms of cosine and sine are listed below.

$$\mathcal{F}\{cos(2\pi f_0 t)\} = \frac{1}{2}[\delta(f - f_0) + \delta(f - f_0)]$$

$$\mathcal{F}\{sin(2\pi f_0 t)\} = j\frac{1}{2}[\delta(f - f_0) - \delta(f - f_0)]$$

   For Task 1 ($cos(2\pi t)$), the resulting Fourier transform is $\frac{1}{2}[\delta(f-1) + \delta(f-1)]$. For Task 2, it becomes $\frac{5}{2}[\delta(f-1) - \delta(f-1)]$. If one looks at the magnitude plots for Task 1 and Task 2, those correspond exactly to these equations. The Task 1 plot is described by the equation here, whereas since the plot is a magnitude plot, the Task 2 plot is described by the equation here, except shifted into the positive and real domain.

4. The lab tasks, expectations, and deliverables were as per usual clear and not problematic.

## 7 Conclusion

This lab introduced the Fast Fourier transform in Python. Using the FFT function for the magnitude of a function (and, with a small modification, the phase), we can easily discern and verify the Fourier transform of the function (limited to sinusoidals in this case).