Assignment 8:
The Digital Fourier Transform

Saurav Sachin Kale, EE19B141

May 5, 2021

1 Aim

- To plot the discrete fourier transform of given functions using the fft functions in PyLab Module in Python
- Understand the effect of sampling rate, time interval being transformed to frequency domain on the Discrete Fourier Transform (DFT)
- Finding the time interval for the lowest error in the calculated fourier transform and the DFT obtained from fft function.

2 Procedure

2.1 Analysis of the signal $sin(5t)$

We are given the signal $sin(5t)$. If we extract the frequency components we get:

$$sin(5t) = \frac{1}{2j}e^{j5t} - \frac{1}{2j}e^{-j5t} = -\frac{j}{2}e^{j5t} + \frac{j}{2}e^{-j5t}$$

Expectations for the magnitude plot are two peaks of hieght 0.5 at 5 and -5. For the phase plot, we expect a phase of $-\frac{\pi}{2}$ for the peak at 5 and $\frac{\pi}{2}$ for the peak at -5. We find the frequency spectrum by using the fft() function, and without any changes plot the magnitude and phase plot of the result we get.

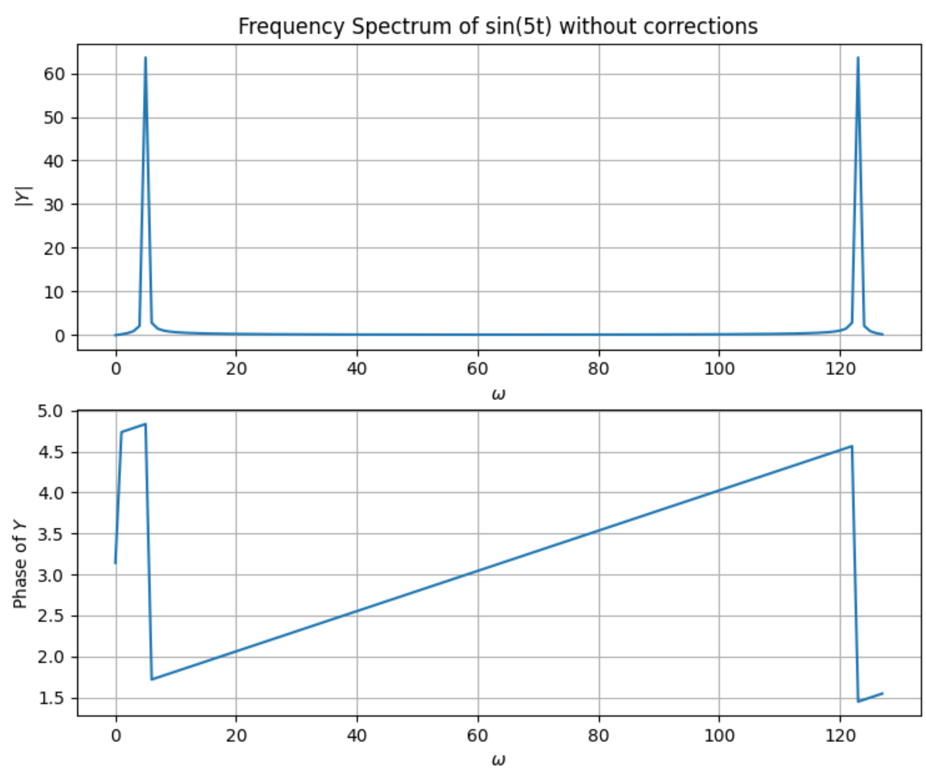

Figure 1: Frequency Spectrum of $sin(5t)$ without any changes

But the results are not as expected. This is because of the following reasons:

- The peaks are not where they were expected: fft plots the frequency function from 0 to 2π without centering the zero mark. Therefore our x axis is not indicative of frequency, it is simply the number of samples. We need to shift the plot and center the 0 point so we get x axis going from $-\pi$ to π. We will do that using a function called fftshift(). This function recenters the 0 mark in the output of fft().
- The values of magnitude are not as expected: fft() function output is basically a plot of how many samples correspond to a given frequency component. Here, since amplitude of 5 and -5 is same and we have 128 samples, we have got exactly 64 samples on each component. So we need to normalize this with respect to the total number of samples. We do this by dividing the output by the number of samples (here 128).

On making these changes, we obtain the following plot.

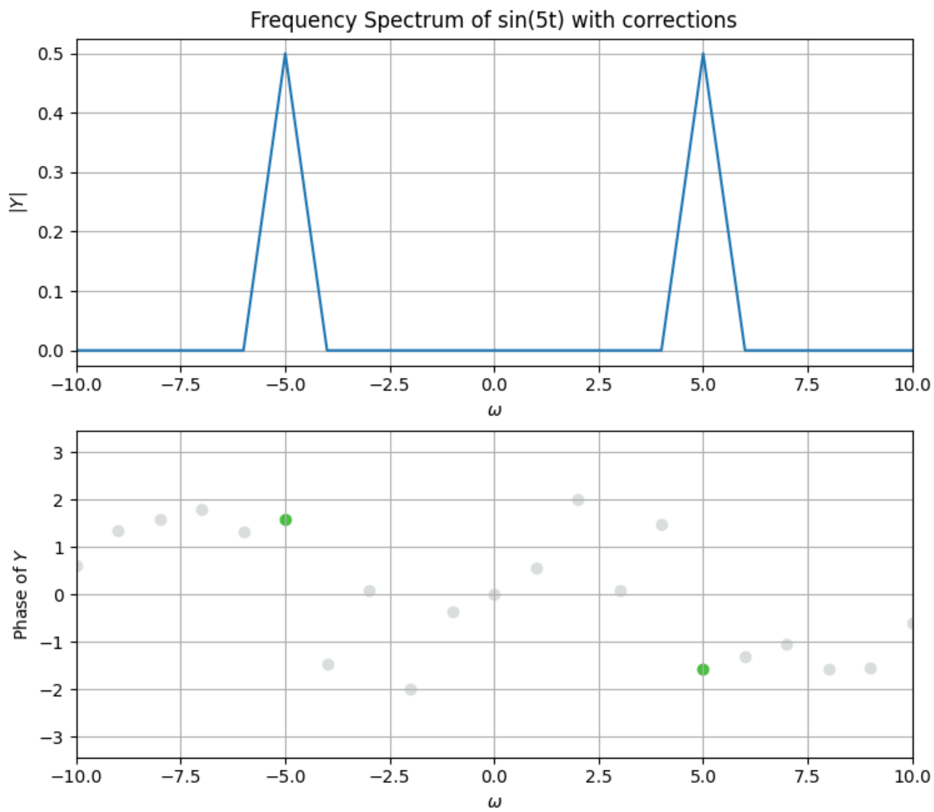

Figure 2: Frequency Spectrum of $sin(5t)$ with corrections

This plot is exactly as expected in both phase and magnitude.

2.1.1 The plotFT function

We will now create a function which given the following parameters:

- N: the number of samples per cycle

- *xinterval*: the time interval for sampling the function (one cycle)
- *signal*: the function which will be sampled

Gives back the magnitude and phase of the resulting frequency spectrum.

```
# function to generate the fourier terms from
# - a given time domain function
# - a given time interval xinterval
# - number of samples per cycle
def plotFT(signal, xinterval, N):
    # initialize the points where we will sample the function.
    # size of x is N, since N is number of samples per cycle
    x = p.linspace(xinterval[0], xinterval[1], N + 1)[:-1]
    w = p.linspace(-p.pi*N/(xinterval[1]-xinterval[0]), p.pi*N/(xinterval[1]-xinterval

    # sample the function
    y = signal(x)

    # calculate and normalize the frequency spectrum
    Y = p.fftshift(p.fft(y))/N
    phase = p.angle(Y)

    # return the magnitude and the phase
    return w, abs(Y), phase
```

We also need to redefine our x axis, since it is currently simply the number of frequencies from 0 to 128, we need to map these to the actual values of these frequencies.
Since we have N samples per cycle, let us calculate the time duration between consecutive sampling points. Let us call this time T_s. Here

$$NT_s = T_0$$

Where T_0 is the interval in which the function is being sampled.

$$T_s = \frac{T_0}{N}$$

Since the DFT exists only for periodic DT functions, our sampled function also has to be periodic with some time period, say M. Since we are using one cycle and suppose we are plotting from 0 to 2π, after M samples, we should reach 2π

$$MT_s = 2\pi$$

$$M = \frac{2\pi}{T_s} = \frac{2\pi N}{T_0}$$

Therefore, the frequencies corresponding to our samples are going from $[-\frac{\pi N}{T_0}, \frac{\pi N}{T_0})$. We make the following two observations from this analysis.

- Increasing the number of samples increases the range of the frequencies we can represent. That explains the definition of `w` in our code.
- Notice that the discrete frequency step in our DFT result is given by the total range divided by the number of samples

$$\Delta\omega = \frac{\frac{2\pi N}{T_0}}{N} = \frac{2\pi}{T_0}$$

Therefore, increasing T_0 has the effect of decreasing the discrete frequency step. We get better resolution in the frequencies on increasing T_0.

2.2 The Frequency Spectrum of an Amplitude Modulated (AM) signal

We have been given the signal $(1 + 0.1cos(t))cos(10t)$. Calculating the frequency components by hand we get:

$$(1 + 0.1cos(t))cos(10t) = (1 + \frac{0.1}{2}e^{jt} + \frac{0.1}{2}e^{-jt})(\frac{1}{2}e^{j10t} + \frac{1}{2}e^{-j10t})$$

$$\frac{1}{2}e^{j10t} + \frac{1}{2}e^{-j10t} + \frac{0.1}{4}e^{j11t} + \frac{0.1}{4}e^{j9t} + \frac{0.1}{4}e^{-j11t} + \frac{0.1}{4}e^{-j9t}$$

We expect 6 peaks, at ± 10 of amplitude 0.5, at ± 9 and ± 11 with amplitudes $\frac{1}{40} = 0.025$. For phase, all the terms will have zero phase since they are all real valued. We now use the same method as described before to plot the frequency spectrum.

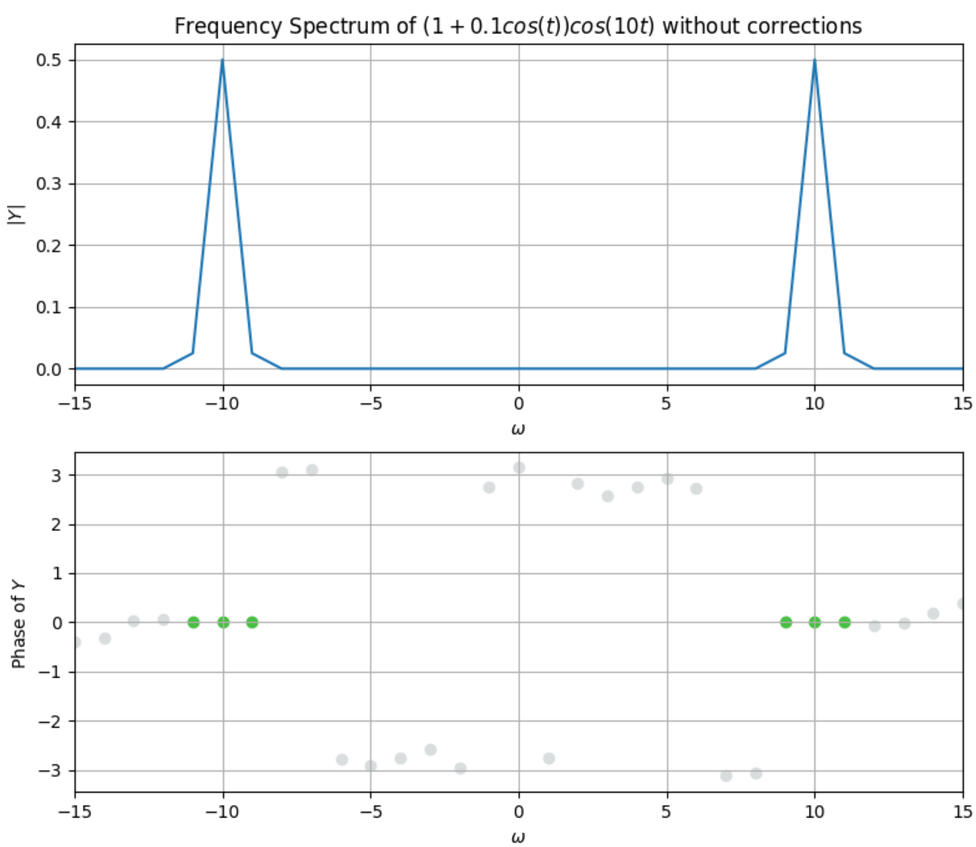

Figure 3: Frequency Spectrum of AM signal without corrections

Again, this is not as expected. This is because the time interval we have used $(0, 2\pi)$ is too low. We increase the time interval to $(-4\pi, 4\pi)$, since stretching the time interval increases the resolution of the DFT.

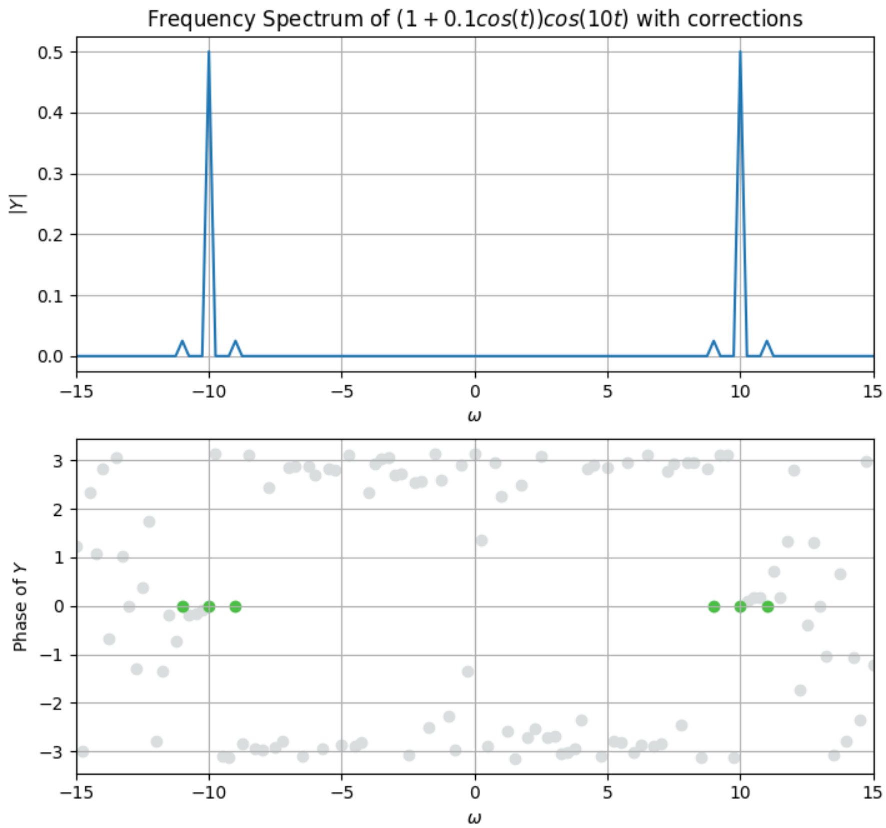

Figure 4: Frequency Spectrum of AM signal with corrections

This plot is exactly as expected.

2.3 Frequency Spectrum of $sin^3(t)$ and $cos^3(t)$

We start with the analysis of $sin^3(t)$. We split it up into its frequency components using the triple angle formula:

$$sin(3t) = 3sin(t) - 4sin^3(t)$$

$$sin^3(t) = \frac{3}{4}sin(t) - \frac{1}{4}sin(3t)$$

$$sin^3(t) = \frac{3}{8j}e^{jt} - \frac{3}{8j}e^{-jt} - \frac{1}{8j}e^{3jt} + \frac{1}{8j}e^{-3jt}$$

$$sin^3(t) = -\frac{3j}{8}e^{jt} + \frac{3j}{8}e^{-jt} + \frac{j}{8}e^{3jt} - \frac{j}{8}e^{-3jt}$$

We expect 4 peaks, two at $1, -1$ with phases $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, and two at $3, -3$ with phases $\frac{\pi}{2}$ and $-\frac{\pi}{2}$. We now plot the frequency spectrum.

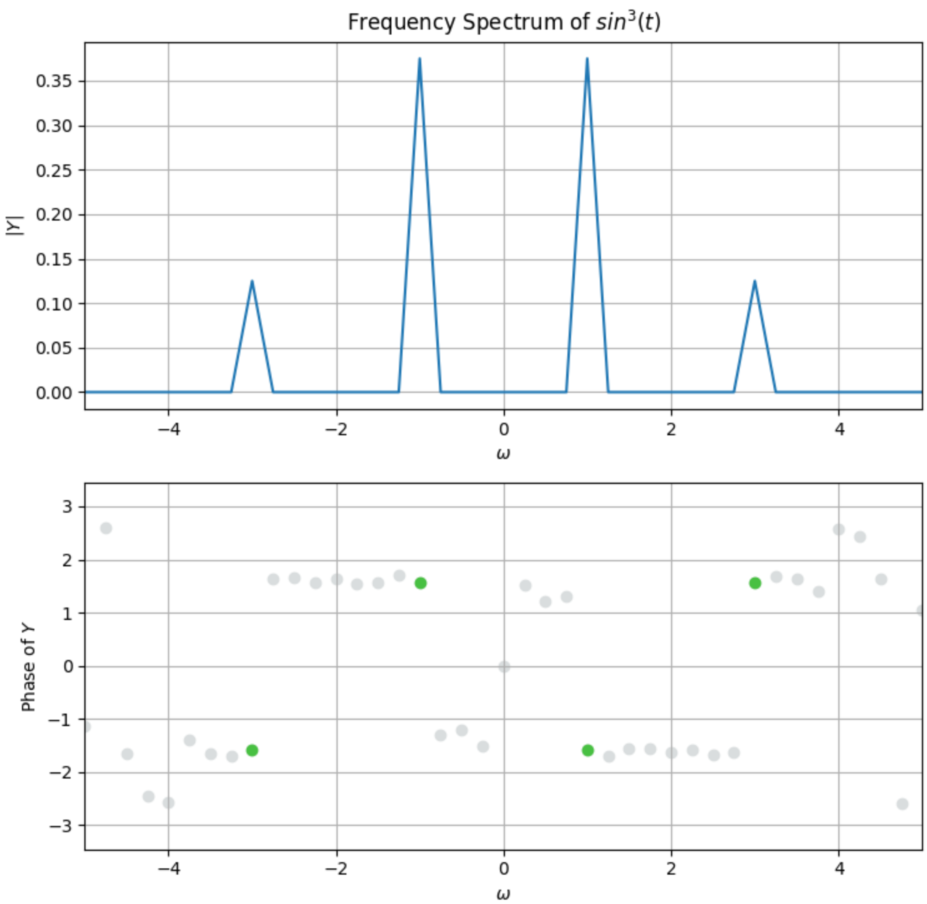

Figure 5: Frequency Spectrum for the signal $sin^3(t)$

The result is as expected.
We now analyse the signal $cos^3(t)$. Again using the triple angle formulae we obtain

$$cos(3t) = 4cos^3(t) - 3cos(t)$$

$$cos^3(t) = \frac{1}{4}cos(3t) + \frac{3}{4}cos(t)$$

$$cos^3(t) = \frac{1}{8}e^{j3t} + \frac{1}{8}e^{-j3t} + \frac{3}{8}e^{jt} + \frac{3}{8}e^{jt}$$

We expect 4 peaks, at ± 1 and ± 3 all with zero phase, since all coefficients are real valued. We now plot the frequency spectrum.

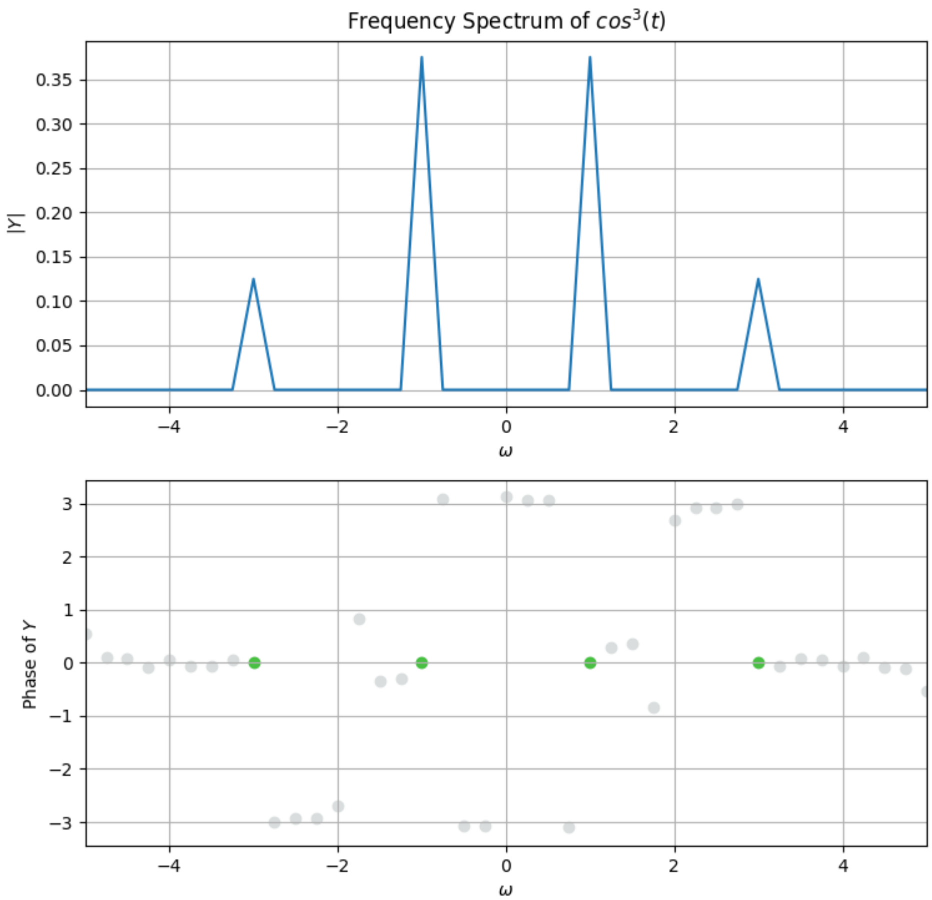

Figure 6: Frequency Spectrum for the signal $cos^3(t)$

The spectrum is exactly as we expect.

2.4 Frequency Spectrum of $cos(20t + 5cos(t))$

We consider the signal $cos(20t + 5cos(t))$. We first plot the frequency spectrum of this signal.

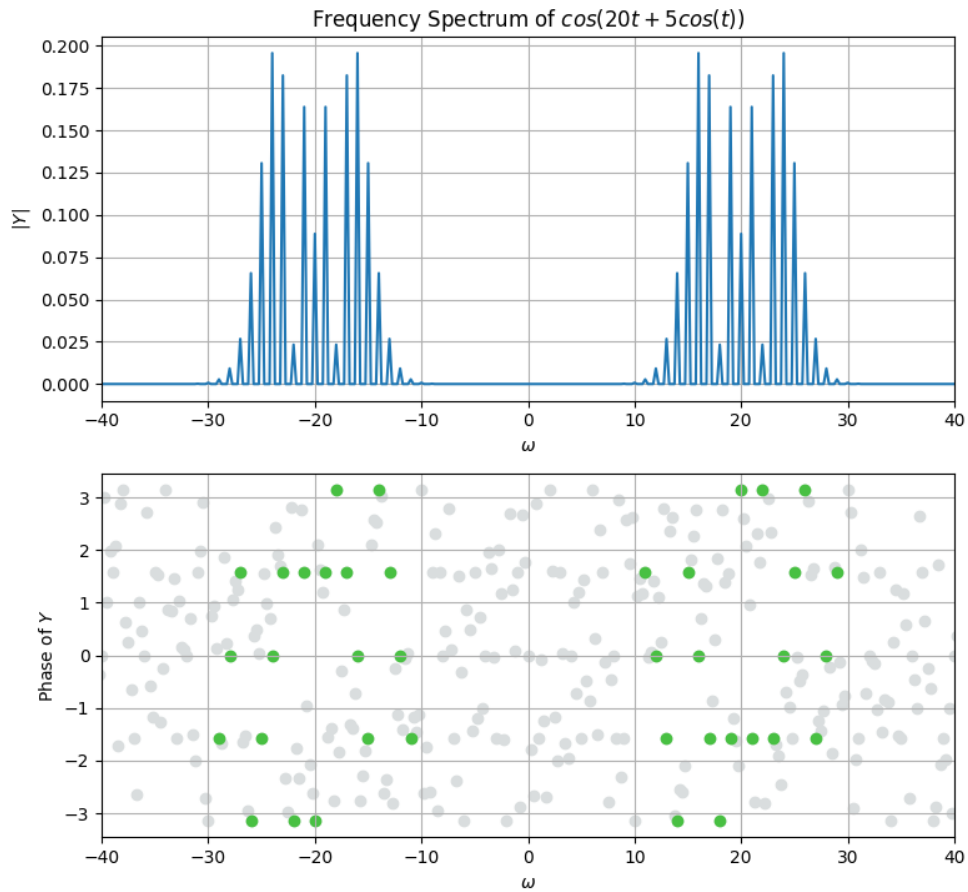

Figure 7: Frequency Spectrum for the signal $cos(20t + 5cos(t))$

We attempt to explain this spectrum. We see that this is a frequency modulated wave. We can analyse the frequency spectrum using Bessel functions.

$$cos(\omega_c t + \beta sin(\omega_m t)) = \sum_{k=-\infty}^{\infty} J_k(\beta) cos((\omega_c - k\omega_m)t)$$

$$cos(20t + 5cos(t)) = \sum_{k=-\infty}^{\infty} J_k(5) cos((20 - k)t)$$

Where J_k is the k^{th} order Bessel function. We see that we will get bands centered at ± 20. The amplitude of each component is proportional to the value of the Bessel function. Eventually at large values of k, the amplitudes die out, hence we obtain two bands centered at ± 20 and die out a short distance from it.

2.5 Frequency Spectrum of Gaussian

We now analyse the signal $e^{-\frac{t^2}{2}}$. We note that this function is not periodic. And since the DFT does not exist for a non-periodic function, we can choose an interval and assume that the function is periodic. We can

then slowly increase the time period, and the DFT will approach the Fourier Transform, because at large time periods, the function can be considered approximately aperiodic.
We know that the actual fourier transform for $e^{-\frac{t^2}{2}}$ is

$$\mathcal{F}\{e^{-\frac{t^2}{2}}\} = \sqrt{2\pi}e^{-\frac{\omega^2}{2}}$$

We will compare the error of the calculated DFT and the Fourier transform as we increase the time interval. We will plot the DFT for the intervals $[-i\pi, i\pi)$ where $i \in \{1, 2...., 11\}$. We print the maximum error between this result and the actual fourier transform.
The expected plot is another Gaussian function, with the phase being zero since it is real valued. We then plot the resulting plot.

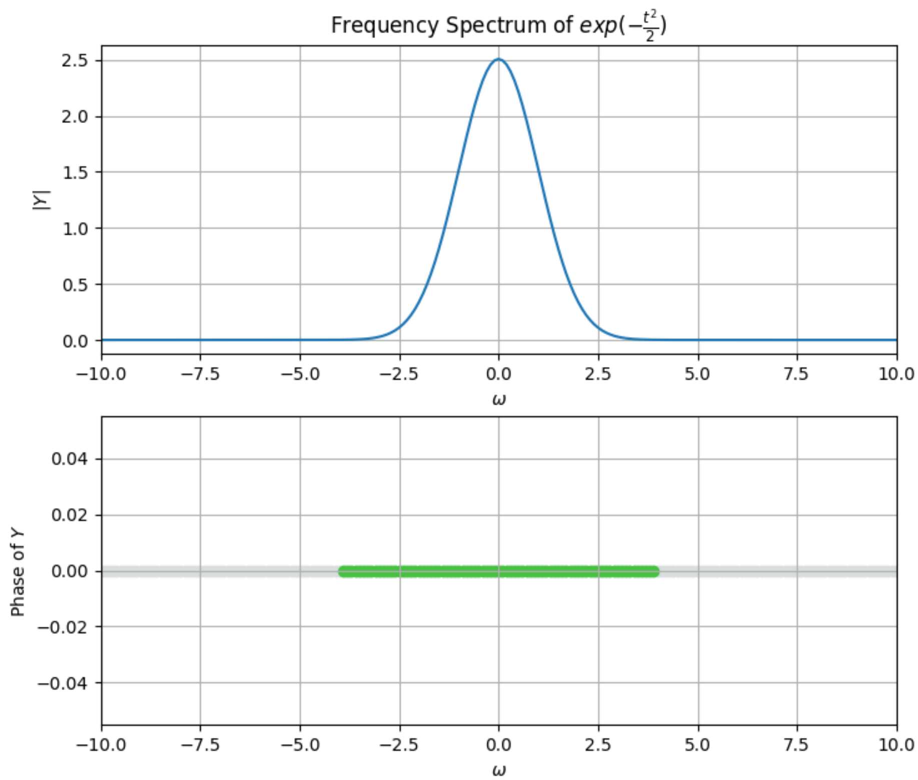

Figure 8: Frequency Spectrum of an FM wave

```
Maximum errors for time intervals
For time interval of [-1pi, 1pi) Maximum error = 0.00649697413874839
For time interval of [-2pi, 2pi) Maximum error = 1.5613434989347752e-09
For time interval of [-3pi, 3pi) Maximum error = 1.3100631690576847e-14
For time interval of [-4pi, 4pi) Maximum error = 4.620196861797216e-16
For time interval of [-5pi, 5pi) Maximum error = 7.438494264988549e-15
For time interval of [-6pi, 6pi) Maximum error = 7.105427357601002e-15
For time interval of [-7pi, 7pi) Maximum error = 8.659739592076221e-15
For time interval of [-8pi, 8pi) Maximum error = 9.67119940904726e-16
```

```
For time interval of [-9pi, 9pi) Maximum error = 4.884981308350689e-15
For time interval of [-10pi, 10pi) Maximum error = 4.440892098500626e-15
```

This is what we expected. The errors are also quite within acceptable range.

- Note that the method of normalizing the FFT result for the gaussian function is different since it is aperiodic. We have made a slight modification to our function to change the normalization method if the function passed is a gaussian.

  ```
  # gaussian curve follows a different method of normalization since it is the only 
  # periodic function we will consider.
  if normalizeGauss:
      Y = p.fftshift(abs(p.fft(y)))/N
      Y = Y * p.sqrt(2*p.pi)/max(Y)
      actualY =  p.exp(-w**2/2) * p.sqrt(2 * p.pi)
      maxError = max(abs(actualY-Y))
      phase = p.angle(Y)
      return w, abs(Y), phase, maxError
  ```

  We have chosen to take the absolute value of the fft() result. This is because the samples can be assymetric about 0. Due to sampling, since we don't have a completely accurate gaussian, some frequency components can have small negative amplitudes. Taking an absolute value and increasing the number of samples fixes this problem.

- As we increase the time interval (the time interval in one cycle) the frequency spectrum comes closer and closer to the fourier transform of the gaussian, because for a large period, we can approximate the function as aperiodic, and we see that the error hits the order of 10^{-14} or lower below an interval of 6π centered at zero.

3 Conclusions

- The frequency spectrum magnitude and phase plot was plotted and analysed for several signals including sinusoids, AM and FM signals.
- The FFT functions in the PyLab module in Python were used.
- The fourier transform of a non periodic signal was approximated with an error smaller than 10^{-14} by plotting the DFT considering it to be periodic, and then increasing the time period.