Assignment 8: The Digital Fourier Transform

J.Phani Jayanth - EE19B026

May 5, 2021

Abstract

This weeks Assignment involves the following:

- Using *numpy.fft.fft()* to plot the Discrete Time Fourier Transforms of different periodic functions
- We interpret these plots and compare them with their actual/expected spectra

Introduction

Suppose *f[n]* is a periodic sequence of samples, with a period N. Then the DTFT of the sequence is also a periodic sequence *F[k]* with the same period N.

$$F[k] = \sum_{n=0}^{N-1} f[n]exp(-2\pi \frac{nk}{N} j) = \sum_{n=0}^{N-1} f[n]W^{nk}$$

$$f[n] = \frac{1}{N} \sum_{n=0}^{N-1} F[k]W^{-nk}$$

Here W = $exp(-2\pi j/N)$. The values of *F[k]* are what remains of the Digital Spectrum $F(e^{j\theta})$. We can consider them as the values of $F(e^{j\theta})$ for $\theta = 2\pi k/N$, since the first N terms in the expression for $F(e^{j2\pi k/N})$ yield

$$F(e^{j2\pi k/N}) = \sum_{n=0}^{N-1} f[n]exp(-\frac{2\pi kn}{N} j) + ...$$

which is the same as the DFT expression. The remaining terms are the repititions of the above sum to help build up the delta function that is needed to take us from a continuous transform to discrete impulses.

In this assignment, we want to explore how to obtain the DFT, and how to recover the analog Fourier Tranform for some known functions by the proper sampling of the function.

Consider the function $y = sin(x)$. We know that,

$$y = sin(x) = \frac{e^{jx} - e^{-jx}}{2j}$$

So the expected spectrum is

$$Y(\omega) = \frac{1}{2j}[\delta(\omega - 1) - \delta(\omega + 1)]$$

Plotting the spectra of $sin(5t)$ and $(1 + 0.1cos(t))cos(10t)$

We use the *fftshift()* function to plot the DFT of *sin(5t)* and *(1+0.1cos(t))cos(10t)*. We scale the frequencty axis accordingly, to view both sides of the Y-axis. We observe peaks at 5 and -5 frequencies, with a magnitude of 0.5. The phase of these peaks are seperated by π radians. This is expected because the spectrum of $sin(5t)$ is:

$$Y(\omega) = \frac{\delta(\omega - 5) - \delta(\omega + 5)}{2j}$$

This is shown in the following figure:

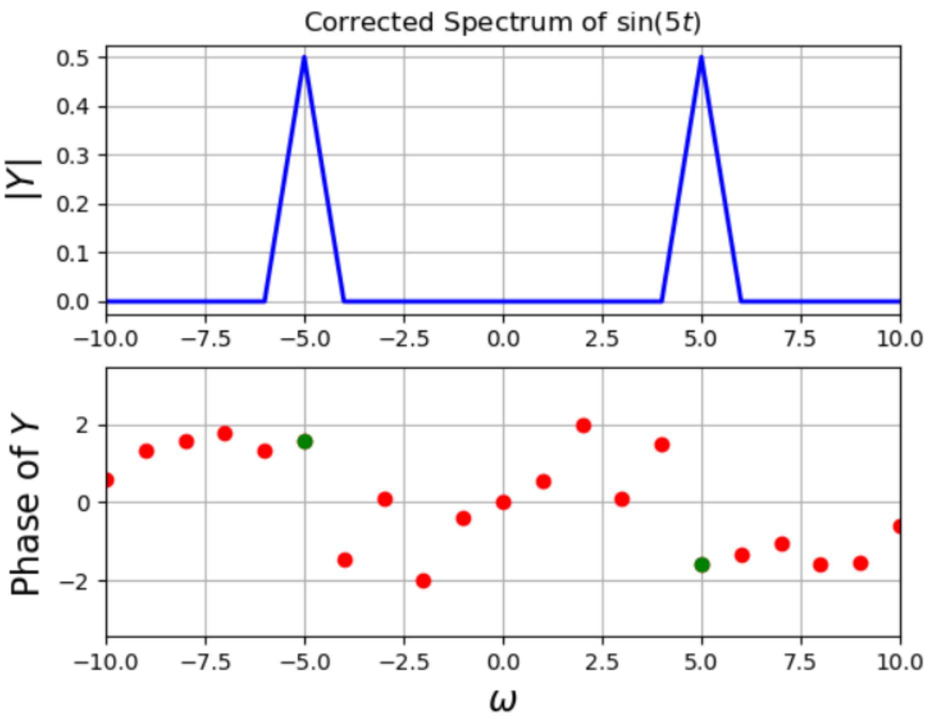

Figure 1: Spectrum of $sin(5t)$

We now plot the DFT Spectrum of $(1 + 0.1cos(t))cos(10t)$ with 128 and also with 512 points. By considering more points, we obtain a spectrum with higher accuracy. For this function, we have peaks at 10 and -10. The phase at these peaks is zero (as expected) - Figure 2 & Figure 3

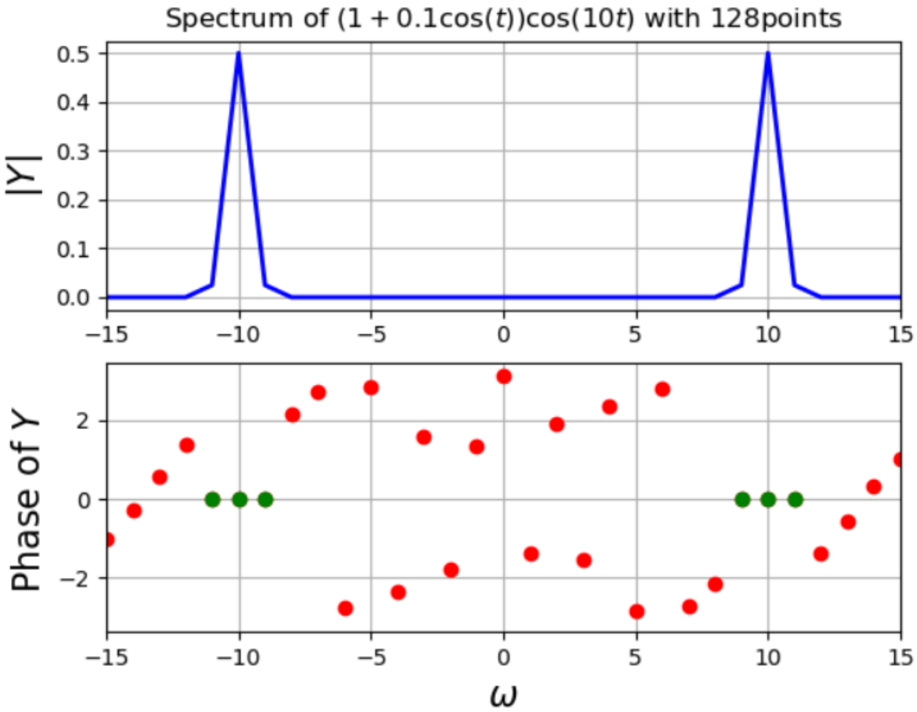

Figure 2: Spectrum of $(1 + 0.1cos(t))cos(10t)$ with 128 sampled points

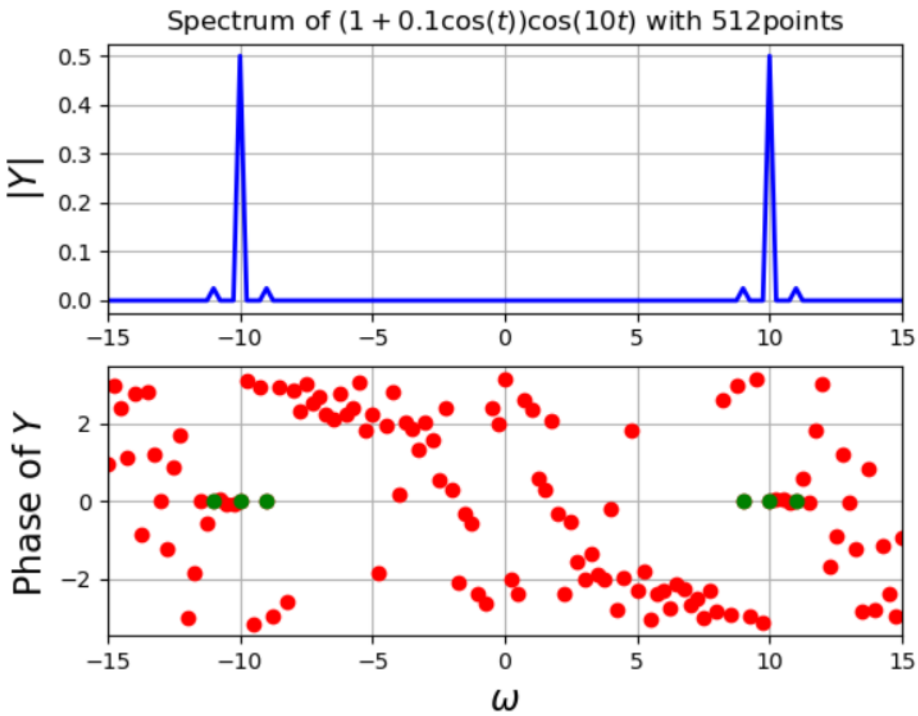

Figure 3: Spectrum of $(1 + 0.1cos(t))cos(10t)$ with 512 sampled points

We can see that by considering higher number of a sampled points, a more accurate spectrum is achieved.

The code is as follows:

```
# CORRECTED FFT of sin(5x)
x=linspace(0,2*pi,129);x=x[:-1]
y=sin(5*x)  # Defining the function
Y=fftshift(fft(y))/128.0  # Corrected DFT
w=linspace(-64,63,128)
figure()
subplot(2,1,1)
plot(w,abs(Y),'b',lw=2)   # Magnitude spectrum
xlim([-10,10])
ylabel(r"$|Y|$",size=16)
title(r"Corrected Spectrum of $\sin(5t)$")
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2) # Phase spectrum
ii=where(abs(Y)>1e-3)
plot(w[ii],angle(Y[ii]),'go',lw=2)  # Phase for points with magnitude > 0.001
xlim([-10,10])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
show()

# DFT of (1+0.1cos(t))cos(10t) using 128 points
t=linspace(0,2*pi,129);t=t[:-1]
y=(1+0.1*cos(t))*cos(10*t) # Defining the function
Y=fftshift(fft(y))/128.0   # Finding DFT
w=linspace(-64,63,128)
figure()
subplot(2,1,1)
plot(w,abs(Y),'b',lw=2)  # Magnitude spectrum
xlim([-15,15])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $(1+0.1cos(t))cos(10t)$ with 128points")
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)  # Phase spectrum
ii=where(abs(Y)>1e-3)
plot(w[ii],angle(Y[ii]),'go',lw=2)  # Phase spectrum for points with magnitude >
    0.001
xlim([-15,15])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
show()
```

Spectra of $sin^3(t)$ and $cos^3(t)$

We know that:

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

Therefore, in the spectrum of $sin^3(t)$, we expect peaks at ω = -3,-1,+1,+3. The peaks at ω = -1 and +1 have a magnitude of 0.75 and those at ω = -3 and +3 have a magnitude of 0.25. Also, the phase at ω = -1,+3 is $\pi/2$ and at ω = -3,+1 is $-\pi/2$ - Figure 4.
Similarly in the spectrum of $cos^3(t)$ also, we expect peaks at ω = -3,-1,+1,+3. The peaks at ω = -1 and +1 have a magnitude of 0.75 and those at ω = -3 and +3 have a magnitude of 0.25. Also, the phase at all these peaks would be zero - Figure 5.

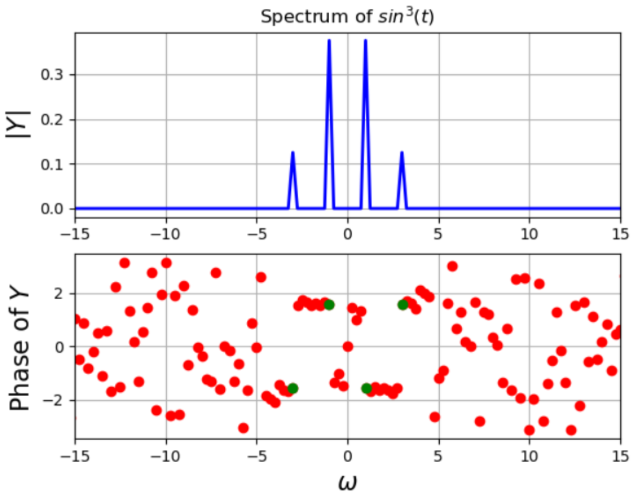

Figure 4: Spectrum of $sin^3(t)$

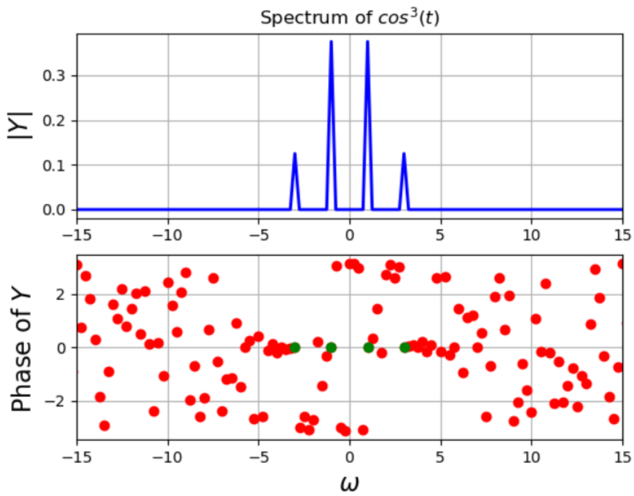

Figure 5: Spectrum of $cos^3(t)$

The Code

In the code, a function ***plotSpectrum()*** is defined to plot the DFT Spectra of a function passed into it. We also define other required parameters required in plotting the spectra of all the required functins.

```
def plotSpectrum(index,N,r,w,xlimit,heading,allphase=True,gauss=False,actgauss=
   False):
    t = linspace(-r,r,N+1);t=t[:-1]
    if actgauss == False:
        y = func(index,t) # Defining the function
    if gauss == False and actgauss == False:
        Y = fftshift(fft(y))/N    # Finding DFT
    elif gauss == True or actgauss == True:
        y = func(index,t)
        Y = fftshift(abs(fft(y)))/N  # Finding DFT
        Y = Y*sqrt(2*pi)/max(Y)
    w = linspace(-w,w,N+1);w=w[:-1]
    if actgauss == True:
        Y_ = sqrt(2*pi)*exp(-w*w/2)
        print(f'Maximum error in the Computed Spectrum of Gaussian function = {
   max(abs(abs(Y)-abs(Y_)))}')
    figure()
    subplot(2,1,1)
    if actgauss == False:
        plot(w,abs(Y),'b',lw=2)  # Magnitude spectrum
    elif actgauss == True:
        plot(w,abs(Y_),'b',lw=2)  # Magnitude spectrum
    xlim([-xlimit,xlimit])
    ylabel(r"$|Y|$",size=16)
    title(heading)
    grid(True)
    subplot(2,1,2)
    if allphase == True:
        if actgauss == False:
            plot(w,angle(Y),'ro',lw=2)  # Phase spectrum
            ii = where(abs(Y)>1e-3)
        elif actgauss == True:
            plot(w,angle(Y_),'ro',lw=2)  # Phase spectrum
            ii = where(abs(Y_)>1e-3)
    elif allphase == False:
        ii = where(abs(Y)>1e-3)
    plot(w[ii],angle(Y[ii]),'go',lw=2)  # Phase spectrum for points with
   magnitude > 0.001
    xlim([-xlimit,xlimit])
    ylabel("Phase of $Y$",size=16)
    xlabel(r"$\omega$",size=16)
    grid(True); show()
```

We also define a function *func()*, which requires a numerical index as the input to return the required function whose spectrum is to be plotted by the *plotSpectrum()* function.

```
def func(index,t):
    if index == 0:
        return (1+0.1*cos(t))*cos(10*t)
    elif index == 1:
        return sin(t)*sin(t)*sin(t)
    elif index == 2:
        return cos(t)*cos(t)*cos(t)
    elif index == 3:
        return cos(20*t + 5*cos(t))
    elif index == 4:
        return exp(-t*t/2)
```

Therefore, to plot the spectrum of $sin^3(t)$ and $cos^3(t)$, the code is as follows:

```
plotSpectrum(1,512,4*pi,64,xlimit=15,heading=r"Spectrum of $sin^3(t)$")
plotSpectrum(2,512,4*pi,64,xlimit=15,heading=r"Spectrum of $cos^3(t)$")
```

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

We take the frequency-modulated function $cos(20t + 5cos(t))$ and plot its DFT Spectrum. We consider an interval of $(-8\pi, 8\pi)$ and also plot the phase only for the points where the magnitude is significant (greater than 10^{-3}). As expected, we obtain the spectrum around the carrier frequency of 20 as shown in the figure below.

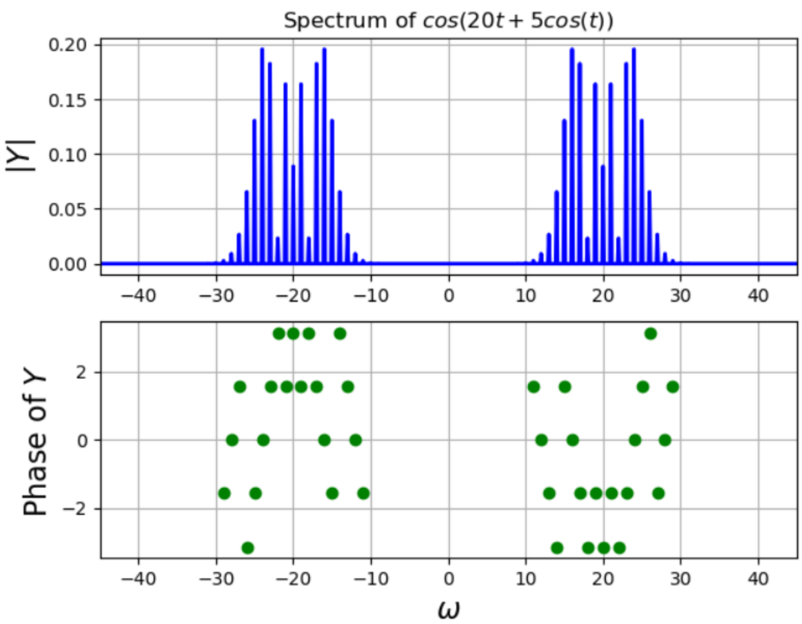

Figure 6: Spectrum of $cos(20t + 5cos(t))$

To obtain the plot, we just pass the function to the *plotSpectrum()* function as discussed before:

```
plotSpectrum(3,2048,8*pi,128,xlimit=45,heading=r"Spectrum of $cos(20t+5cos(t))$",
    allphase=False)
```

Spectrum of Gaussian Function

We plot the spectrum of the Gaussian function, $e^{-t^2/2}$. But since this function is aperiodic, it is not band-limited in the frequency domain. Hence the spectrum is different for different values of time limit we consider and also the spectrum might be slightly inaccurate.
The actual spectrum of the Gaussian function is also Gaussian (in ω), and is given by:

$$F(e^{j\omega}) = \sqrt{2\pi}e^{-\omega^2/2}$$

The phase at all points will be zero since the transform is real. For plotting the spectrum, we consider two different time ranges: $(-4\pi, 4\pi)$ and $(-8\pi, 8\pi)$.

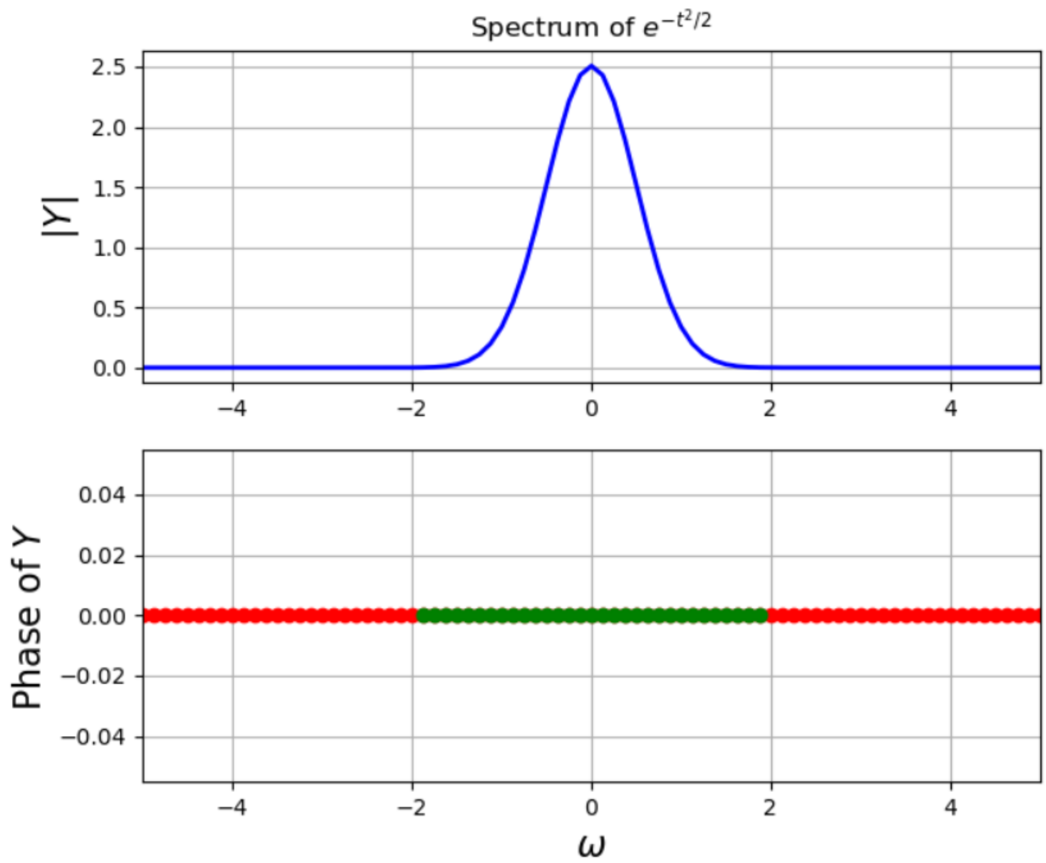

Figure 7: Spectrum of $e^{-t^2/2}$ in the interval $(-4\pi, 4\pi)$

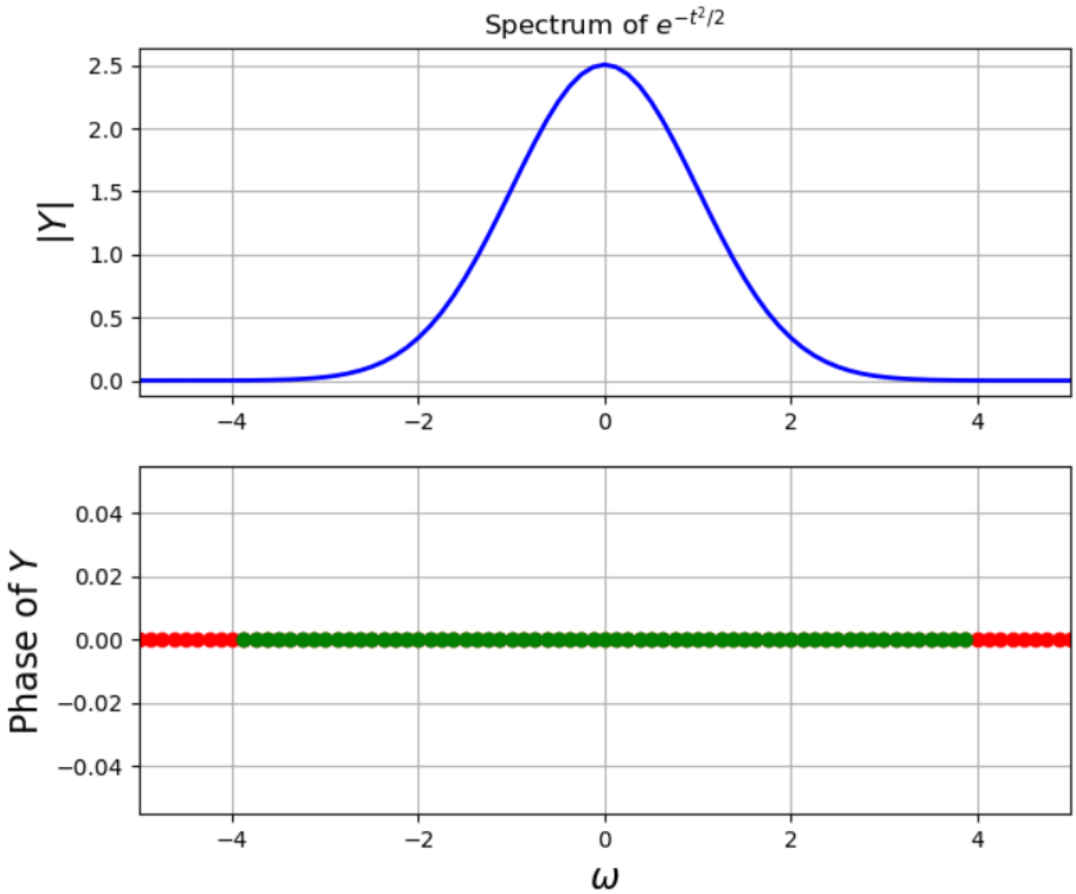

Figure 8: Spectrum of $e^{-t^2/2}$ in the interval $(-8\pi, 8\pi)$

From the above figures, we observe that the peak sharpens with an increase in sampling rate.

We also plot the actual spectrum of the Gaussian function to compare with the computed DFT (Figure 9). We observe that the spectra coincide and the deviation in both the spectra is very minute. We also compute the maximum error between the actual and the computed spectrum. The maximum error obtained is: 1.010436×10^{-15}

```
# Computing the Error:
print(f'Maximum error in the Computed Spectrum of Gaussian function = {max(abs(
    abs(Y)-abs(Y_)))}')
# Plotting
plotSpectrum(4,2**9,4*pi,2**5,xlimit=5,heading=r"Spectrum of $e^{-t^2/2}$",gauss=
    True)
plotSpectrum(4,2**9,8*pi,2**5,xlimit=5,heading=r"Spectrum of $e^{-t^2/2}$",gauss=
    True)
plotSpectrum(4,2**9,8*pi,2**5,xlimit=5,heading=r"Actual Spectrum of $e^{-t^2/2}$"
    ,actgauss=True)
```

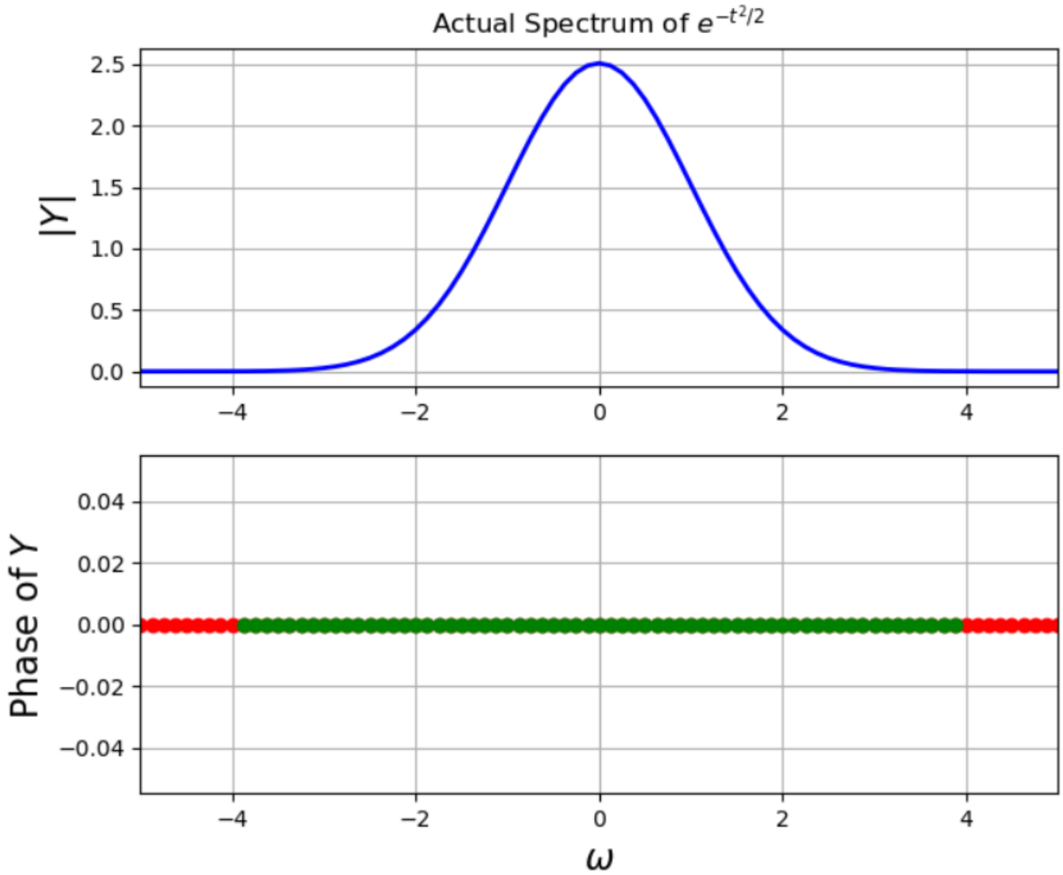

Figure 9: Actual Spectrum of $e^{-t^2/2}$ in the interval $(-8\pi, 8\pi)$

Conclusion

From this assignment, we learned how to use the *fft* library of Python for computing and plotting the Discrete Time Fourier Transforms (DFT) of various functions. We analysed sinusoidal signals, frequency and amplitude modulated signals. For sinusoidal signals, DFT consists of impulses at the sinusoid frequencies. The amplitude modulated signal had a frequency spectrum with impulses at the carrier and the side-band frequencies. The frequency moduated signal, having an infinite number of side-band frequencies, gives rise of a DFT with non-zero values for a broader range of frequencies. We have also looked at the DFT of the Gaussian function, which is also a Gaussian and have onserved that the spectrum sharpens for higher sampling rates and broadens for larger time ranges.