Assignment 8: DFT

Abhishek Sekar
EE18B067

22nd March 2020

0.1 Introduction

- We analyse and use DFT to find the Fourier transform of periodic signals and non periodic ones using fast fourier transform algorithms which are implemented in python using fft and $fftshift$ which is used to center the fourier spectra of a discrete signal.
- The discrete Fourier transform (DFT) converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency.
- If f[n] are the samples of some continuous function $f(t)$ then we define the Z transform as

$$F(z) = \sum_{n=-\infty}^{n=\infty} f(n)z^{-n} \tag{1}$$

- Replacing z with $e^{j\omega}$ we get DTFT of the sampled function

$$F(e^{j\omega}) = \sum_{n=-\infty}^{\infty} f(n)e^{-j\omega n} \tag{2}$$

- $F(e^{j\omega})$ is continuous and periodic. $f[n]$ is discrete and aperiodic. Suppose now $f[n]$ is itself periodic with a period N, i.e.,

$$f[n+N] = f[n]$$

- Then, it should have samples for its DTFT. This is true, and leads to the Discrete Fourier Transform or the DFT:
- 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.

$$F[k] = \sum_{n=0}^{N-1} f[n]e^{-j\frac{2\pi nk}{N}} = \sum_{n=0}^{N-1} f[n]W^{nk} \tag{3}$$

$$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} F[k]W^{-nk} \tag{4}$$

- Here $W = e^{-j\frac{2\pi}{N}}$ is used simply to make the equations less cluttered and k is the sampled values of continuous variable ω at multiples of $\frac{2\pi}{N}$

- What this means is that the DFT is a sampled version of the DTFT, which is the digital counterpart of the analog Fourier Transform .In this assignment, we want to explore how to obtain the DFT, and how to recover the analog Fourier Transform for some known functions by the proper sampling of the function.

0.2 Python Code

0.2.1 Question 1:

- To find Discrete Fourier Transform DFT of $\sin(5t)$ and (AM) Amplitude Modulated signal given by $(1 + 0.1\cos(t))\cos(10t)$
- Plot and analyse the spectrum obtained for both the functions given above.
- Cross validate the spectrum obtained with what is expected.
- To compare the spectrum obtained for $\sin(5t)$, we use

$$\sin(5t) = \frac{1}{2j}e^{j5} - \frac{1}{2j}e^{-j5} \tag{5}$$

- So the fourier transform of $\sin(5t)$ using above relation is

$$\mathscr{F}(\sin(5t)) \rightarrow \frac{1}{2j}(\ \delta(\omega - 5) - \delta(\omega + 5)\ ) \tag{6}$$

- Similarly for finding Fourier Transform AM signal following relations are used

$$(1 + 0.1\cos(t))\cos(10t) \rightarrow \cos(10t) + 0.1\cos(10t)\cos(t) \tag{7}$$

$$0.1\cos(10t)\cos(t) \rightarrow 0.05(\ \cos(11t) + cos(9t)\ ) \tag{8}$$

$$(1 + 0.1\cos(t))\cos(10t) \rightarrow 0.025(e^{j11t} + e^{j9t} + e^{j11t} + e^{-j9t}) \tag{9}$$

- So we can find fourier transform from above relation
- So using this we compare the plots of Magnitude and phase spectrum obtained using DFT and analyse them.

```python
#Function to select different functions
def select(t, n):
    if(n == 1):
        return sin(5*t)       #function in example 1
    elif(n == 2):
```

```python
        return (1+0.1*cos(t))*cos(10*t)#function in example 2
    elif(n == 3):
        return pow(sin(t), 3)
    elif(n == 4):
        return pow(cos(t), 3)
    elif(n == 5):
        return cos(20*t + 5*cos(t))
    else:
        return exp(-pow(t, 2)/2)    #Gaussian


'''
Function to find Discrete Fourier Transform
 l_lim,u_lim     -> lower & upper limit for time
 points          -> Sampling rate
 n               -> code for function above(1,6)
 norm_factor     ->  only for Gaussian function
                    it is given as parameter, it is the normalizing factor.
'''


def DFT(l_lim, u_lim, points,n,norm_Factor=0):
    t =np.linspace(l_lim, u_lim,points+1)[:-1]
    y =select(t, n)
    N = points

    # DFT for gaussian function
    # norm_factor is multiplying constant to DFT

    if(norm_Factor != 0):
        Y = fftshift((fft(y))*norm_Factor)
    else:
        # normal DFT for periodic functions
        Y = fftshift(fft(y))/(N)

```

```
    w_lim = (2*pi*N/((u_lim-l_lim)))  #freq domain
    w = linspace(-(w_lim/2), (w_lim/2), (points+1))[:-1]
    return t, Y, w


'''
Function to plot Magnitude and Phase spectrum for given function
Arguments:

 thresh_val     -> value above which phase is made zero
 xlims,ylims    -> limits for x&y axis for spectrum
 plt_title -> title of plot
'''


def plot_DFT(t, Y, w, thresh_val,plt_title,xlims=[-15,15],ylims=None):

    subplot(2, 1, 1)
    plt.plot(w, abs(Y), lw=2)
    plt.xlim(xlims)
    if(ylims != None):
        plt.ylim(ylims)
    plt.xlabel("w ->")
    plt.ylabel("|Y(jw)| ->")
    plt.title(plt_title)
    plt.grid()

    ax = subplot(2, 1, 2)
    ii = where(abs(Y) > thresh_val)
    plt.plot(w[ii], angle(Y[ii]), 'go', lw=2)

    if(ylims != None):
        plt.ylim(ylims)

    plt.xlim(xlims)
```

```python
    plt.ylabel("phase(Y(jw)) ->")
    plt.xlabel("w ->")
    plt.grid()
    plt.show()


'''
DFT for sin(5t) computed in incorrect way
* like without normalizing  factor
* without centering fft of function to zero
'''

x = linspace(0, 2*pi, 128)
y = sin(5*x)
Y = fft(y)
subplot(2, 1, 1)
plt.plot(abs(Y), lw=2)
plt.title("Figure 1 : Incorrect Spectrum of sin(5t)")
plt.ylabel("|Y(jw)| ->")
plt.grid()

subplot(2, 1, 2)
plt.plot(unwrap(angle(Y)), lw=2)
plt.xlabel("w -> ")
plt.ylabel("phase(Y(jw)) ->")
plt.grid()
plt.show()


#proper computation of DFT for Sin(5t)
t, Y, w = DFT(0, 2*pi, 128,1)
plot_DFT(t, Y, w,1e-3,"Figure 2: Spectrum of sin(5t)")

#unstretched t-axis calculation for (1+0.1cos(t))cos(10t)
t, Y, w = DFT(0, 2*pi, 128,2)
```

```
plot_DFT(t, Y, w, 1e-4,"Figure 3: Incorrect Spectrum of (1+0.1cos(t))cos(10t)"

#stretched t-axis calculation for refined plot
t, Y, w = DFT(-4*pi, 4*pi, 512,2)
plot_DFT(t, Y, w, 1e-4, "Figure 4 : Spectrum of (1+0.1cos(t))cos(10t)")
```

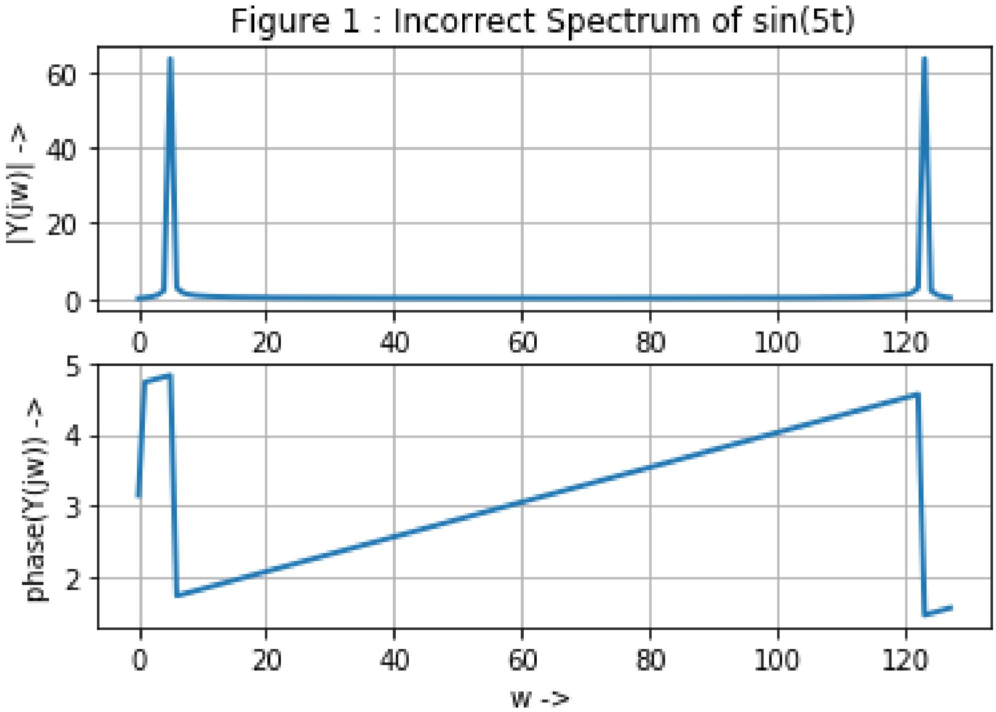

Figure 1: Incorrect Fourier spectrum plots of $sin(5t)$

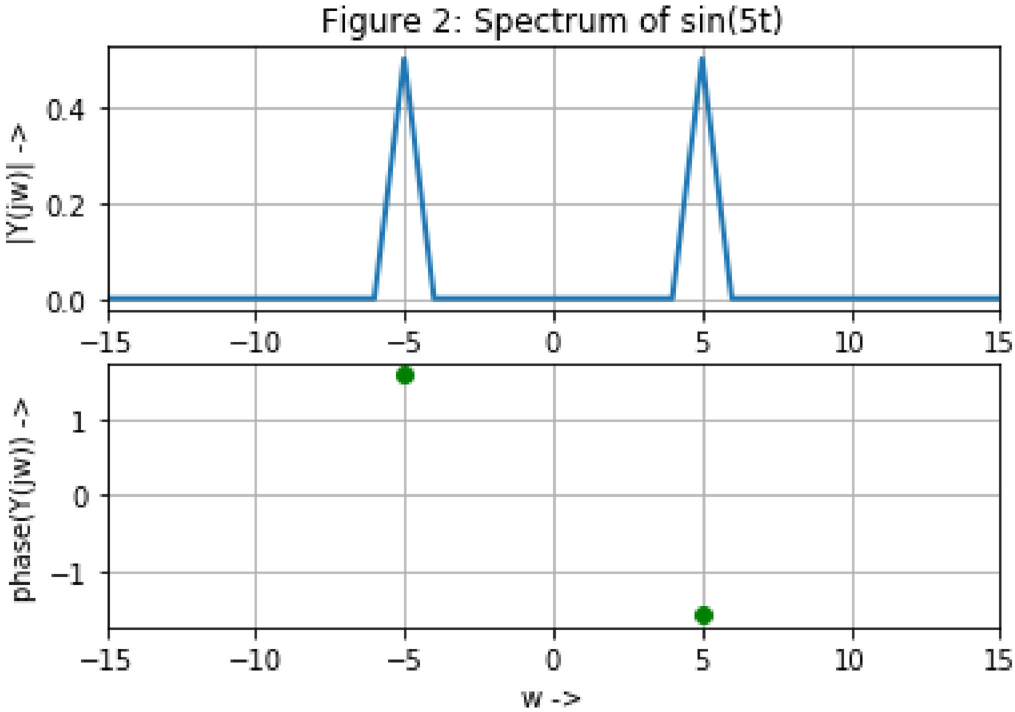

Figure 2: Correct Fourier spectrum plots of $sin(5t)$

Results and Discussion :

- The initial plot is not wrong, but is poorly labelled in x and y axes, so can be practically assumed to be incorrect. The correct plot is obtained by using the $fftshift()$ function, which shifts the fft() function output between a window of interest.
- As we observe the plot frequency contents are of $\omega = 5rads^{-1}$, $-5rads^{-1}$
- Since everything consists of sin terms, the phase is zero and π alternatively. For amplitude of the spectra we analyse the fourier transform of $sin(5t)$ which is derived above.

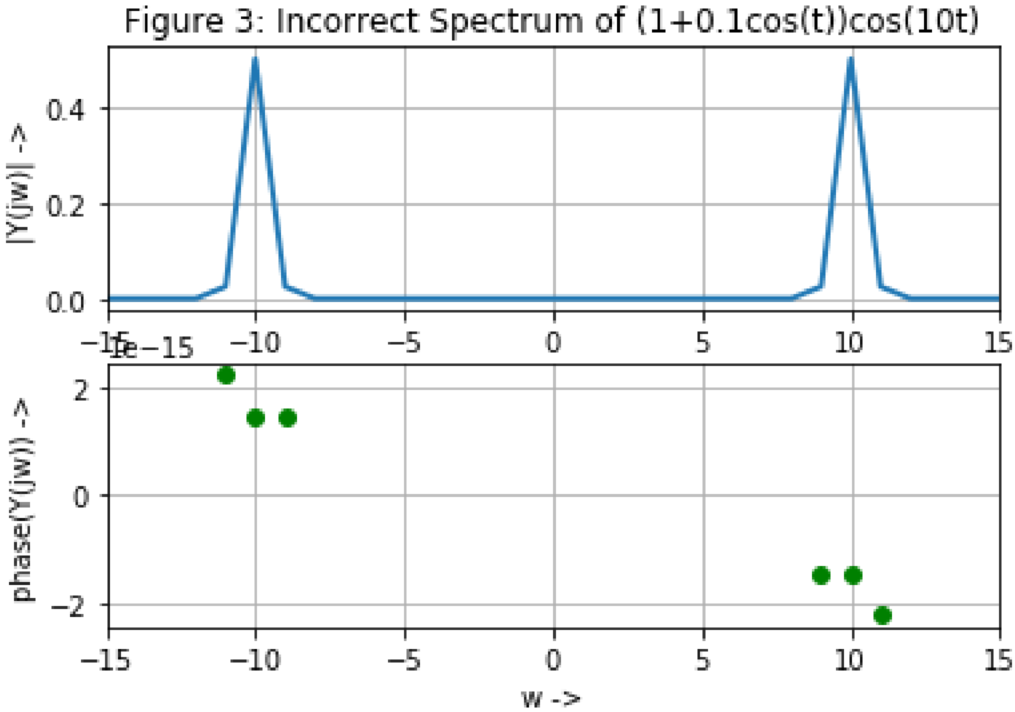

Figure 3: Incorrect Fourier spectrum plots of $(1 + 0.1\cos(t))\cos(10t)$

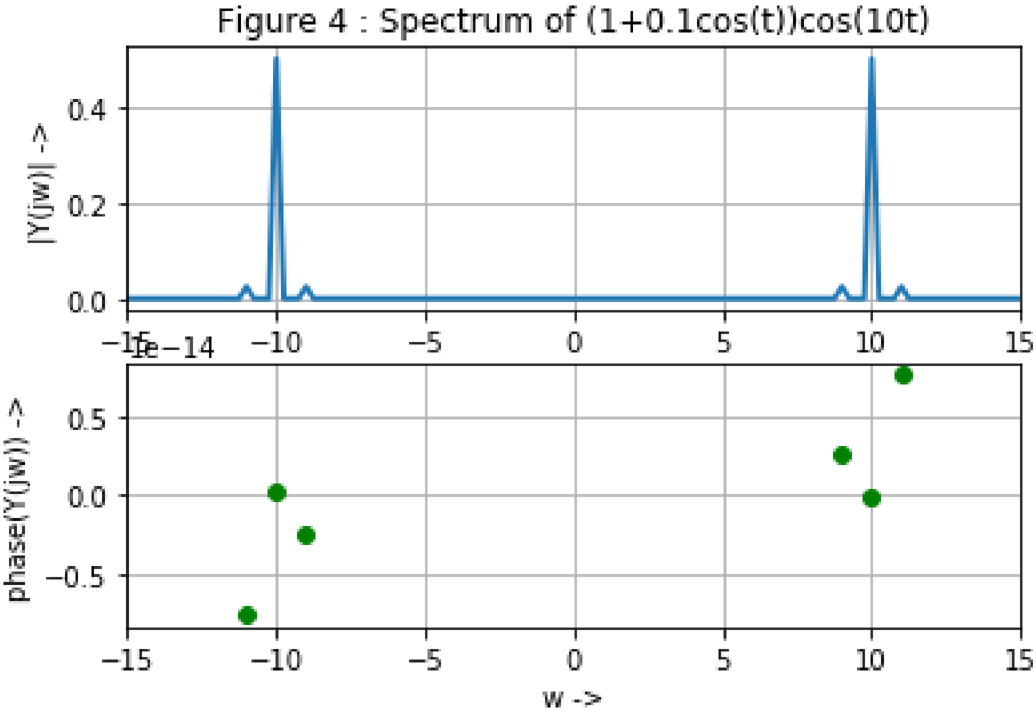

Figure 4: Correct Fourier spectrum plots of $(1 + 0.1\cos(t))\cos(10t)$

Results and Discussion :

- The first plot is partly incorrect because we did not take enough window size for the plots to be perfectly shown, thus the smaller frequency of 2π gets hidden. To bring them out we must increase number of points in time domain, for the smaller frequencies to show up, whose output is shown in second plot.

- As we observe the plot it has center frequencies of $\omega = 10, -10$ from carrier signal and as expected we get side band frequencies at $\omega = \pm 9, \pm 11$ .Since amplitude of the message signal is changed by a carrier signal $\cos(10t)$. It is called as amplitude modulation. And the amplitude of the side band frequencies are obtained from the fourier transform expression.

- Phase spectra is 0 since only cos terms are present.

0.2.2 Question 2:

- To find Discrete Fourier Transform DFT of $\sin^3(t)$ and $\cos^3(t)$
- Plot and analyse the spectrum obtained for both the functions given above.
- Cross validate the spectrum obtained with what is expected.
- To compare the spectrum obtained for $\sin^3(t)$, we use

$$\sin^3(t) = \frac{3}{4}\sin(t) - \frac{1}{4}\sin(3t) \tag{10}$$

- So the fourier transform of $\sin^3(t)$ using above relation is

$$\mathscr{F}(\sin^3(t)) \rightarrow \frac{3}{8j}(\ \delta(\omega-1) - \delta(\omega+1)\ ) - \frac{1}{8j}(\ \delta(\omega-3) - \delta(\omega+3)\ ) \tag{11}$$

- Similarly $\cos^3(t)$ is given by

$$\cos^3(t) = \frac{3}{4}\cos(t) + \frac{1}{4}\cos(3t) \tag{12}$$

- So the fourier transform of $\sin^3(t)$ using above relation is

$$\mathscr{F}(\cos^3(t)) \rightarrow \frac{3}{8j}(\ \delta(\omega-1) + \delta(\omega+1)\ ) + \frac{1}{8j}(\ \delta(\omega-3) + \delta(\omega+3)\ ) \tag{13}$$

- So using this we compare the plots of Magnitude and phase spectrum obtained using DFT and analyse them.

```
#subdivision 2: plot for sin^3(t)
t, Y, w = DFT(-4*pi, 4*pi, 512,3)
plot_DFT(t, Y, w, 1e-4, "Figure 5: Spectrum of sin^{3}(t)")

#subdivision 2: plot for cos^3(t)
t, Y, w = DFT(-4*pi, 4*pi, 512,4)
plot_DFT(t, Y, w, 1e-4,"Figure 6: Spectrum of cos^{3}(t)")
```

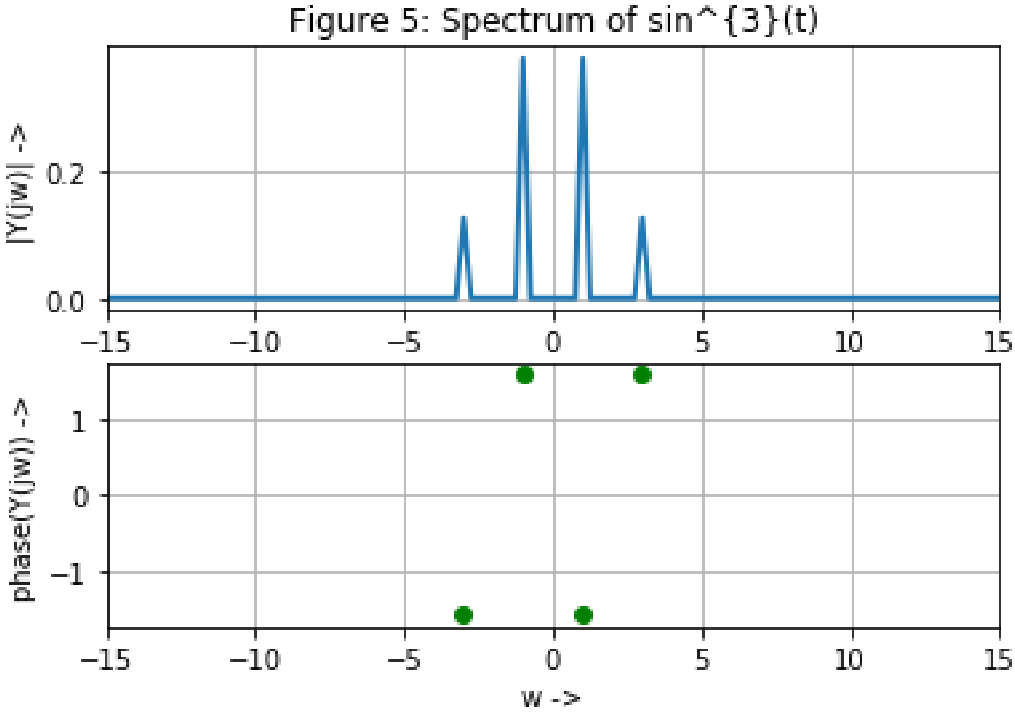

Figure 5: Spectrum of $sin^3(t)$

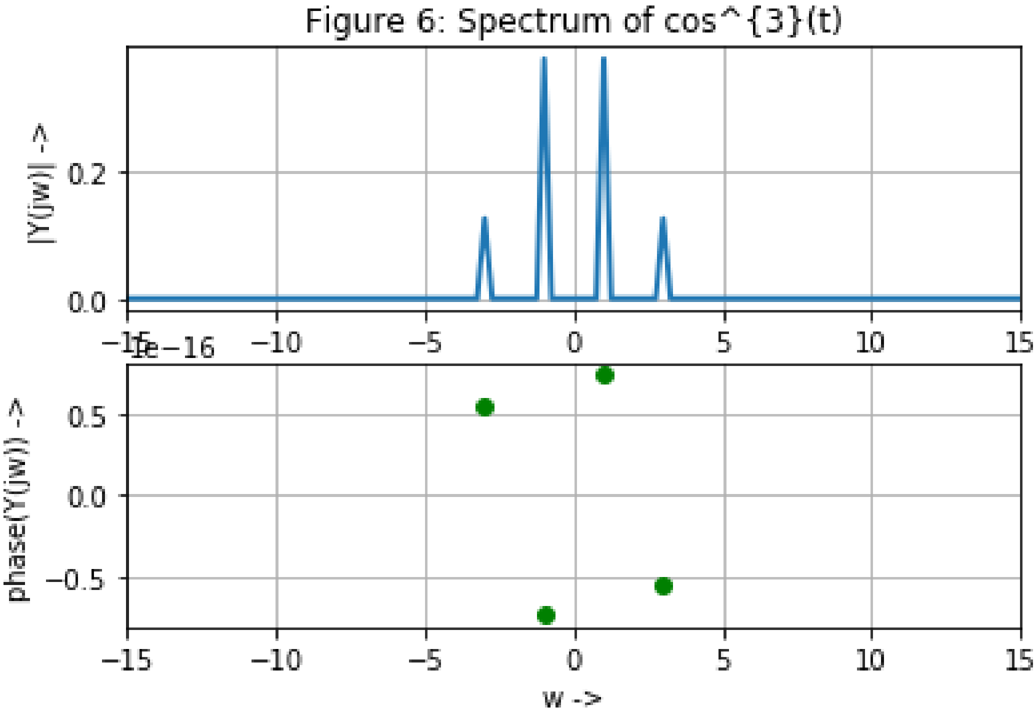

Figure 6: Spectrum of $cos^3(t)$

Results and Discussion :

- As we observe the plot frequency contents are of $\omega = 1, -1, 3, -3$ and with their amplitude in 1:3 ratio

- For $sin^3(t)$, since everything consists of cos terms so phase is zero. But due to lack of infinite computing power they are nearly zero in the order of

- For $cos^3(t)$, since everything consists of sin terms so phase is zero and π alternatively.

0.2.3 Question 3:

- To generate the spectrum of $\cos(20t + 5\cos(t))$.
- Plot phase points only where the magnitude is significant ($> 10^{-3}$).
- Analyse the spectrums obtained.

```
#subdivision 3: plot for cos(20t + 5cos(t))
t, Y, w = DFT(-4*pi, 4*pi, 512,5)
Xlims = [-40, 40]
plot_DFT(t, Y, w, 1e-3,"Figure 7: Spectrum of cos(20t+5cos(t))",Xlims)
```

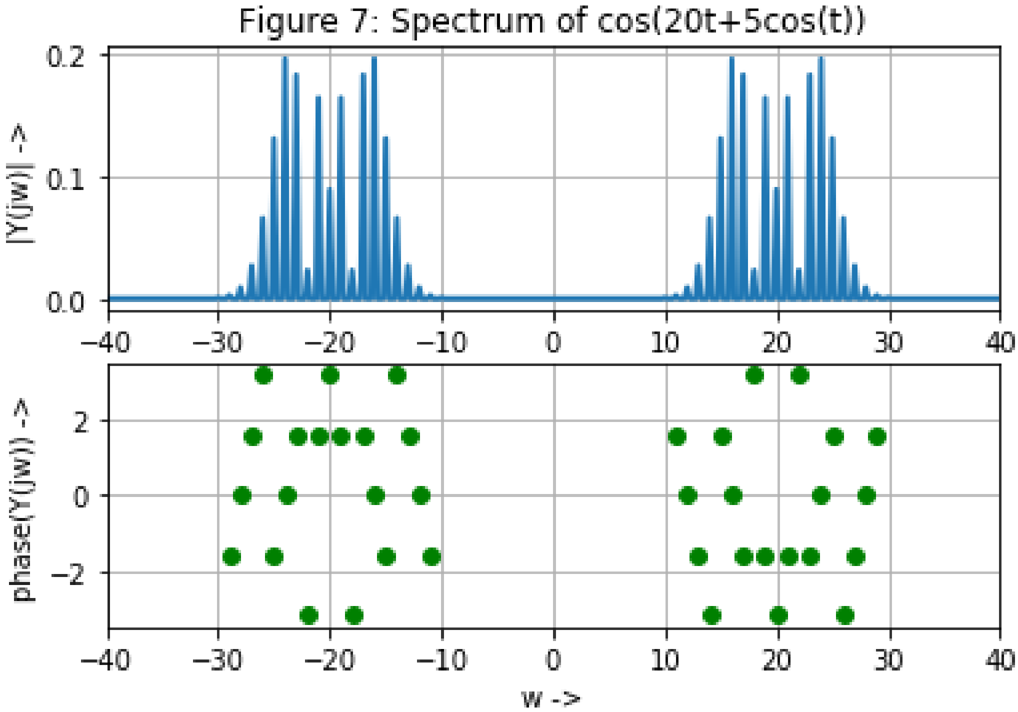

Figure 7: Spectrum of $\cos(20t + 5\cos(t))$

Results and Discussion :

- As we observe the plot that its a Phase modulation since phase of the signal is varying proportional to amplitude of the message signal being $\omega = 20$ and infinite side band frequencies which are produced by $5\cos t$. since $\cos(t)$ is infinitely long signal. But the strength of the side band frequencies decays or very small which are away from center frequency or carrier frequency component as we observe from the plot.

- Phase spectra is a mix of different phases from $[-\pi, \pi]$ because of phase modulation, i.e since the phase is changed continuously wrt time, the carrier signal can represent either a *sine* or *cosine* depending on the phase contribution from $cos(t)$.

0.2.4 Question 4:

- To generate the spectrum of the Gaussian $e^{-\frac{t^2}{2}}$ which is not *bandlimited* in frequency and aperiodic in time domain find Fourier transform of it using DFT and to recover the analog fourier transform from it.

$$\mathscr{F}(\ e^{-\frac{t^2}{2}}\ ) \rightarrow \frac{1}{\sqrt{2\pi}} e^{\frac{-\omega^2}{2}} \tag{14}$$

- To find the normalising constant for DFT obtained we use following steps to derive it :

- window the signal $e^{-\frac{t^2}{2}}$ by rectangular function with gain 1 and window_size 'T' which is equivalent to convolving with $Tsinc(\omega T)$ in frequency domain. So As T is very large the $sinc(\omega T)$ shrinks , we can approximate that as $\delta(\omega)$. So convolving with that we get same thing.

- Windowing done because finite computing power and so we cant represent infinetly wide signal .

- Now we sample the signal with sampling rate N,which is equivalent to convolving impulse train in frequency domain

- And finally for DFT we create periodic copies of the windowed sampled signal and make it periodic and then take one period of its Fourier transform i.e is DFT of gaussian.

- Following these steps we get normalising factor of **Window_size/(2π Sampling_rate)**

$$exp(-\frac{t^2}{2}) \longleftrightarrow \frac{1}{\sqrt{2\pi}} exp(-\frac{\omega^2}{2}) \tag{15}$$

$$rect(\frac{t}{\tau}) = \begin{cases} 1 \ for \ |t| < \tau \\ 0 \ otherwise \end{cases} \tag{16}$$

- For windowing the signal, we will multiply with the rectangular function,

$$y(t) = gaussian(t) \times rect(\frac{t}{\tau}) \tag{17}$$

- In fourier domain, its convolution (since multiplication is convolution in fourier domain)

$$Y(\omega) = \frac{1}{2\pi}(\frac{1}{\sqrt{(2\pi)}}e^{-\omega^2/2} * \frac{sin(\tau\omega)}{\omega}) \tag{18}$$

$$\lim_{\tau\to\infty} Y(\omega) = \frac{\tau}{2\pi}(\frac{1}{\sqrt{(2\pi)}}e^{-\omega^2/2} * \delta(\omega)) \tag{19}$$

$$\lim_{\tau\to\infty} Y(\omega) = \frac{\tau}{2\pi}(\frac{1}{\sqrt{(2\pi)}}e^{-\omega^2/2}) \tag{20}$$

- Now, sampling this signal with a period of $\frac{2\pi}{T_s}$, we will get (multiplication by an impulse train in fourier domain),

$$Y_{sampled} = \frac{\tau}{2\pi T_s}\frac{1}{\sqrt{(2\pi)}}e^{-\omega^2/2}\sum_{k=-\infty}^{\infty}\delta(\omega - \frac{k2\pi}{T_s}) \tag{21}$$

- Solving it further we get the multiplication factor to be,

$$const = \frac{\tau}{T_s 2\pi} \tag{22}$$

- To find the Discrete Fourier transform equivalent for Continous Fourier transform of Gaussian function by finding absolute error between the DFT obtained using the normalising factor obtained with exact Fourier transform and find the parameters such as Window_size and sampling rate by minimising the error obtained with tolerance of 10^{-15}
- To generate the spectrum
- Plot phase points only where the magnitude is significant ($> 10^{-2}$).
- Analyse the spectrums obtained.

```
#Subdivision 4: For the Gaussian
# initial window_size and sampling rate defined
window_size = 2*pi
sampling_rate = 128
# tolerance for error
tol = 1e-15

# normalisation factor derived
norm_factor = (window_size)/(2*pi*(sampling_rate))

#for loop to identify where the minimum error occurs
#window_size and sampling_rate increased to improve accuracy and overcome al

for i in range(1, 10):

    t, Y, w = DFT(-window_size/2, window_size/2,
                        sampling_rate,6, norm_factor)

    # actual transform of the gaussian
    actual_Y = (1/sqrt(2*pi))*exp(-pow(w, 2)/2)
    #comparing both to get error
    error = (mean(abs(abs(Y)-actual_Y)))

    if(error < tol):
        print("\nAccuracy of the DFT is: %g and Iterations took: %g" %
              ((error, i)))
        print("Best Window_size: %g , Sampling_rate: %g" %
              ((window_size, sampling_rate)))
        break
    else:
        window_size = window_size*2
        sampling_rate = (sampling_rate)*2
        norm_factor = (window_size)/(2*pi*(sampling_rate))

```

```python

Xlims = [-10, 10]
plot_DFT(t, Y, w, 1e-2,r"Figure 8: Spectrum of $e^{-\frac{t^{2}}{ 2}}$",Xlims)

# Plotting actual DFT of Gaussian
plot(w, abs(actual_Y),
     label=r"$\frac{1}{\sqrt{2}\pi} e^{\frac{\ - \omega ^{2}}{2}}$")
title("Exact Fourier Transform of Gaussian")
xlim([-10, 10])
ylabel("Y(jw) ->")
xlabel("w ->")
grid()
legend()
show()
```

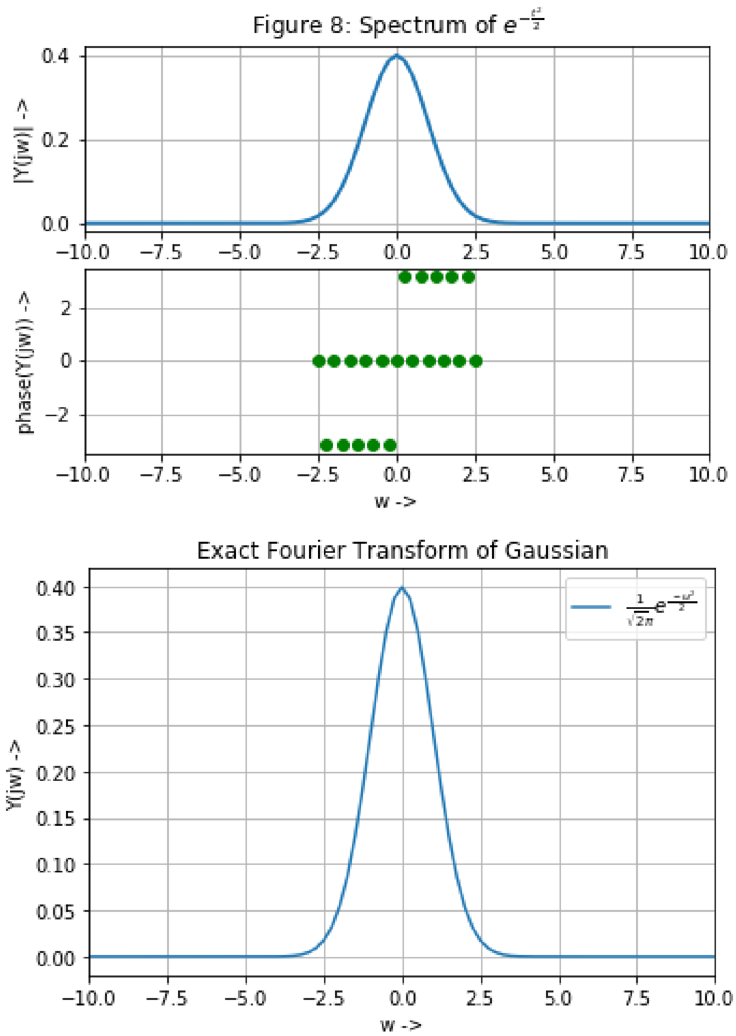

Figure 8: Spectrum of $e^{-\frac{t^2}{2}}$

Results and Discussion :

- Accuracy of the DFT is: 4.14035e-17 and Iterations took: 3
  Best Window_size: 25.1327 , Sampling_rate: 512
- As we observe the magnitude spectrum of $e^{-\frac{t^2}{2}}$ we see that it almost coincides with exact Fourier Transform plotted below with accuracy of $4.14035e^{-17}$
- To find the correct Window size and sampling rate,For loop is used to minimize the error by increasing both window_size and sampling rate as we made assumption that when Window_size is large the sinc(wT) acts like impulse $\delta(\omega)$
- so we increase window_size, similarly sampling rate is increased to overcome aliasing problems when sampling the signal in time domain.
- Similarly we observe the phase plot ,$\angle(Y(\omega) \approx 0$ in the order of 10^{-15} if we magnify and observe

0.2.5 Conclusion :

- Hence we analysed the how to find DFT for various types of signals and how to estimate normalising factors for Gaussian functions and hence recover the analog Fourier transform using DFT ,also to find parameters like window_size and sampling rate by minimizing the error with tolerance upto 10^{-15}!!
- We used fast Fourier transform method to compute DFT as it improves the computation time from $\mathcal{O}n^2 \to \mathcal{O}n\log_2(n)$.
- FFT works well for signals with samples in 2^k , as it divides the samples into even and odd and goes dividing further to compute the DFT.
- That's why we use no of samples in the problems above taken as powers of 2.