EE2703 : Applied Programming Lab
ASSIGNMENT 9

RAKSHIT PANDEY
EE20B106

May 13, 2022

INTRODUCTION

In this week's assignment we will continue our analysis of signals using the fourier transforms. In previous assignment we found the DFT of periodic signals, whereas in this assignment we will do analysis for non periodic signals.

AIM OF THE ASSIGNMENT

The main of this assignment is to analyze the non periodic signals using DFT. While analyzing various non periodic functions we will have discontinuities in the DFT because of the Gibb's Phenomenon. This will be resolved by using the technique of Hamming window. We use this windowed transform signal to analyze signals containing sinusoids of unknown frequencies and then extract it's phase and frequency.

IMPORTING IMPORTANT LIBRARIES

In order to do this assignment in python we need to import some important libraries. These libraries will enable to work with arrays and plot various 2D and 3D curves.

Code

```
from pylab import *
from mpl_toolkits.mplot3d import Axes3D

```

Q1. Worked out examples

For this question we analyze some worked out examples and plot them for understanding them better.

1. Spectrum of $\sin\sqrt{2}t$

If we directly plot the spectrum and phase of $\sin\sqrt{2}t$, we get:

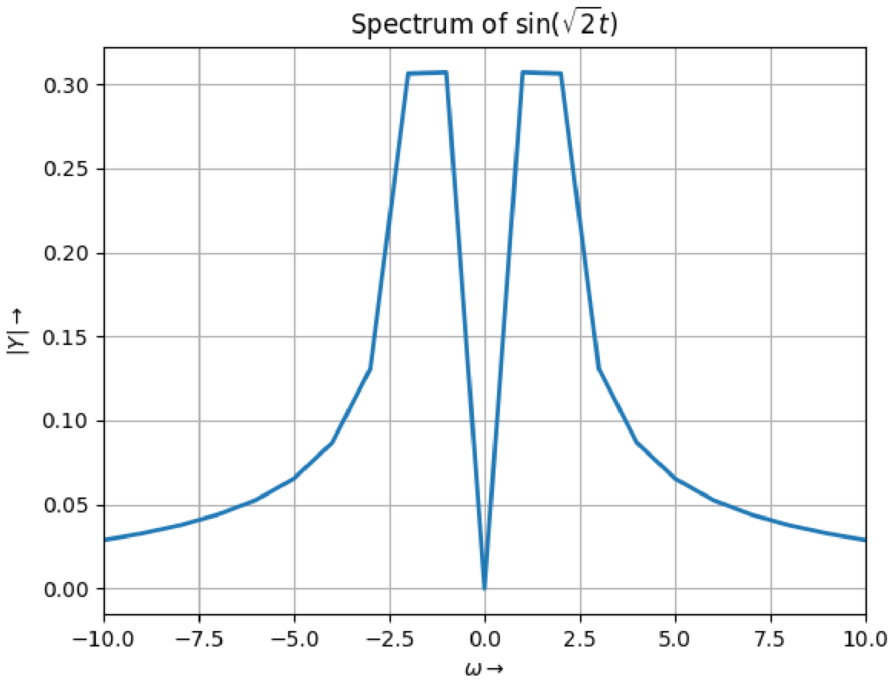

Figure 1: Spectrum of $\sin \sqrt{2}t$

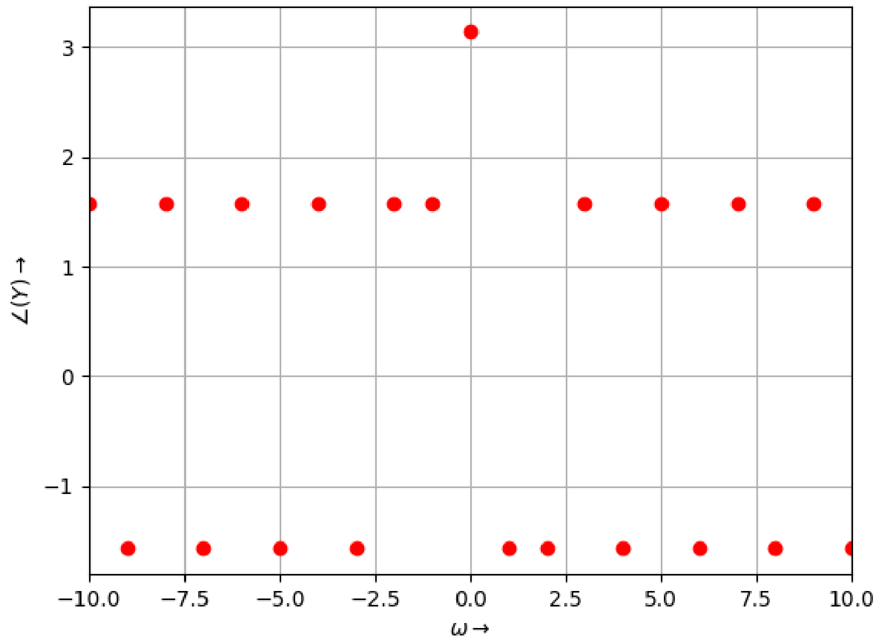

Figure 2: Phase of $\sin \sqrt{2}t$

2. Extended sinusoidal function

The original function for which we wanted the DFT was:

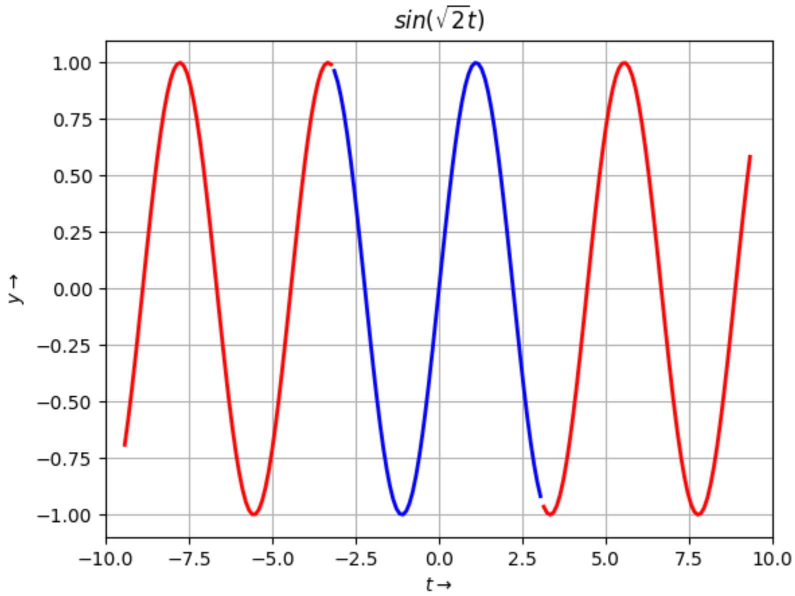

Figure 3: $\sin\sqrt{2t}$ function over multiple time periods

3. $\sin\sqrt{2t}$ with t wrapping every 2π

The DFT is plotted for finite interval hence we plot DFT for the given function:

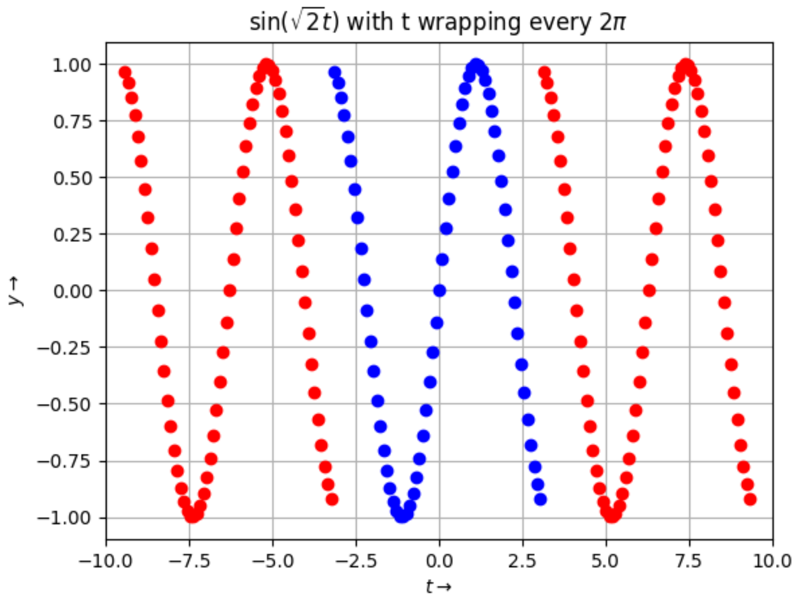

Figure 4: $\sin\sqrt{2t}$ with t wrapping every 2π

4. Spectrum of a digital ramp signal

The discontinuities occur due to the Gibbs's phenomenon. These discontinuities decay as $\frac{1}{\omega}$. We want to analyze this fact, hence we plot the spectrum of a digital ramp signal and analyze this as follows:

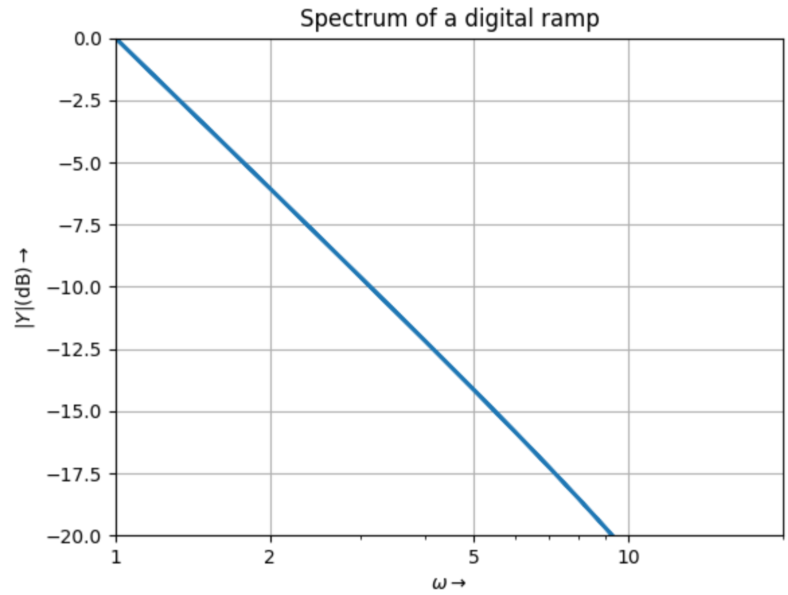

Figure 5: Spectrum of Digital ramp signal

5. Hamming window

So we can clearly observe that the spikes are happening at the end of the periodic interval. So we damp the function near there, i.e we multiply the function sequence f[n] by a window sequence w[n]:

$$g(n) = f(n)w(n)$$

The window which we use is called a Hamming window:

$$w[n] = 0.54 + 0.46cos(\frac{2\pi n}{N-1})$$

If we use the hamming window on our sin $\sqrt{2t}$ function, we get the following curve:

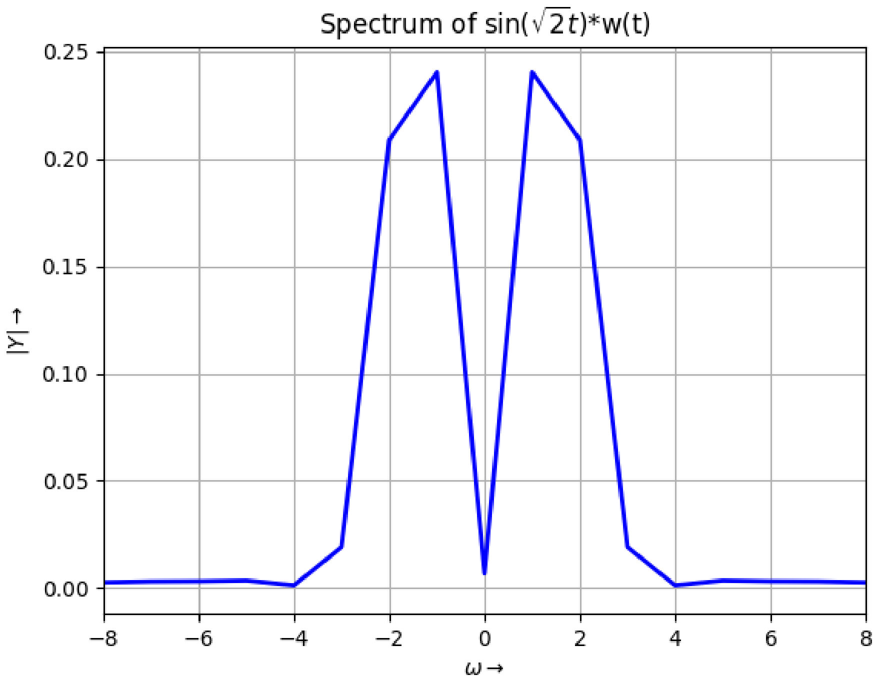

Figure 6: Spectrum of $\sin\sqrt{2}t * w(t)$

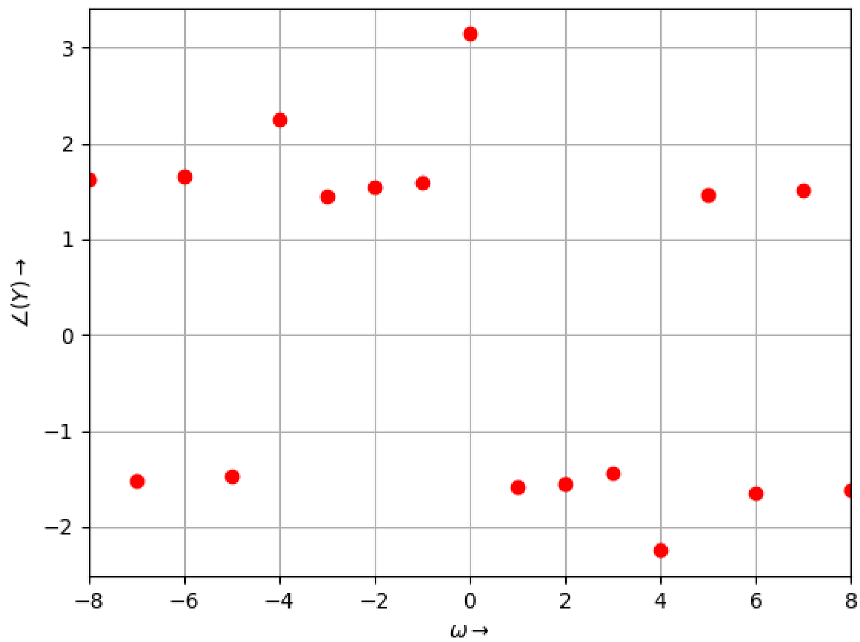

Figure 7: Phase of $\sin\sqrt{2}t * w(t)$

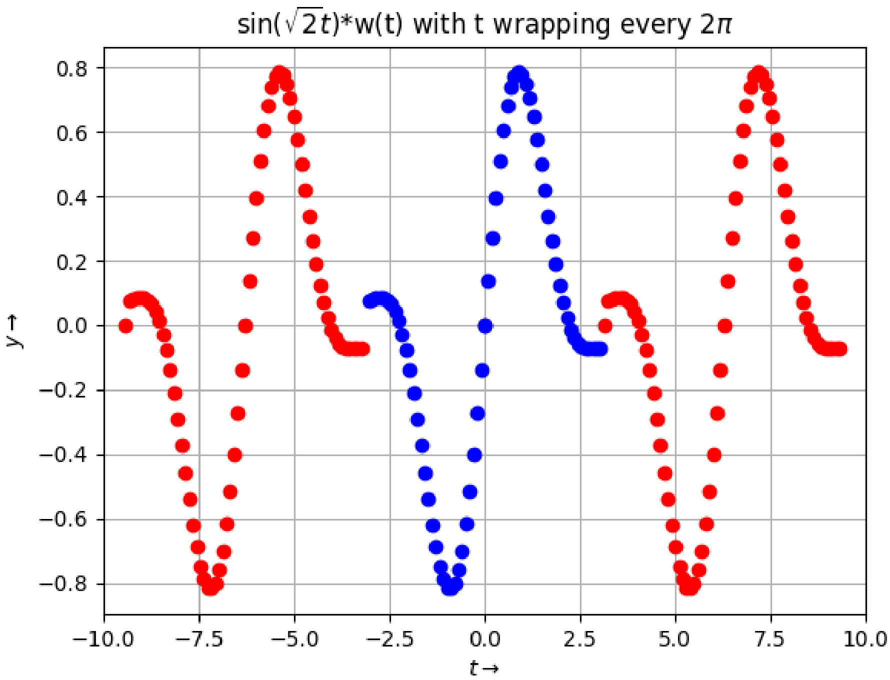

Figure 8: $\sin\sqrt{2t}$*w(t) with t wrapping every 2π

If we use more number of points, in this case 4 times the original number of points, we should get the better result.

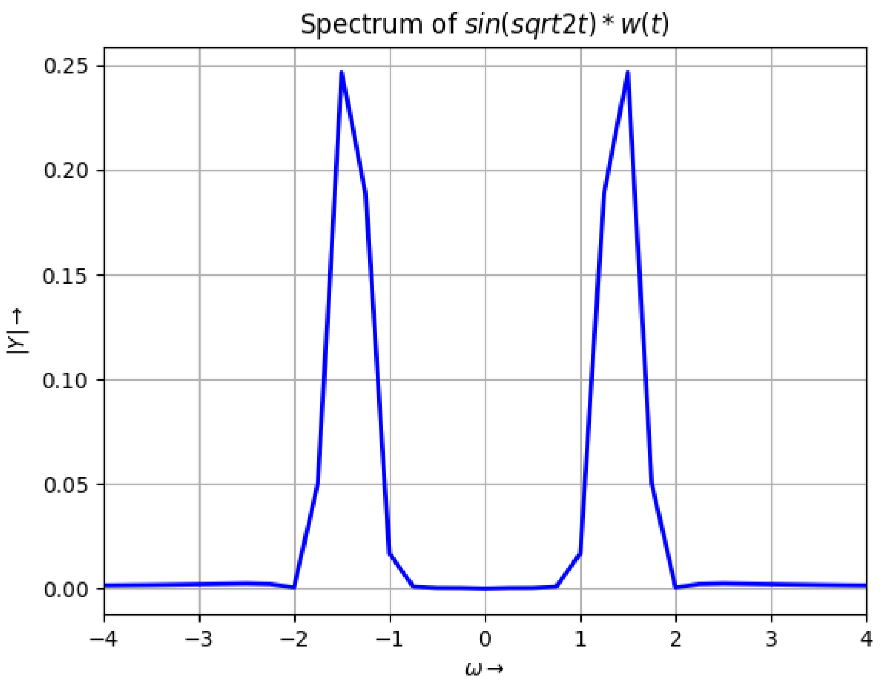

Figure 9: Spectrum of $\sin\sqrt{2t}$*w(t)

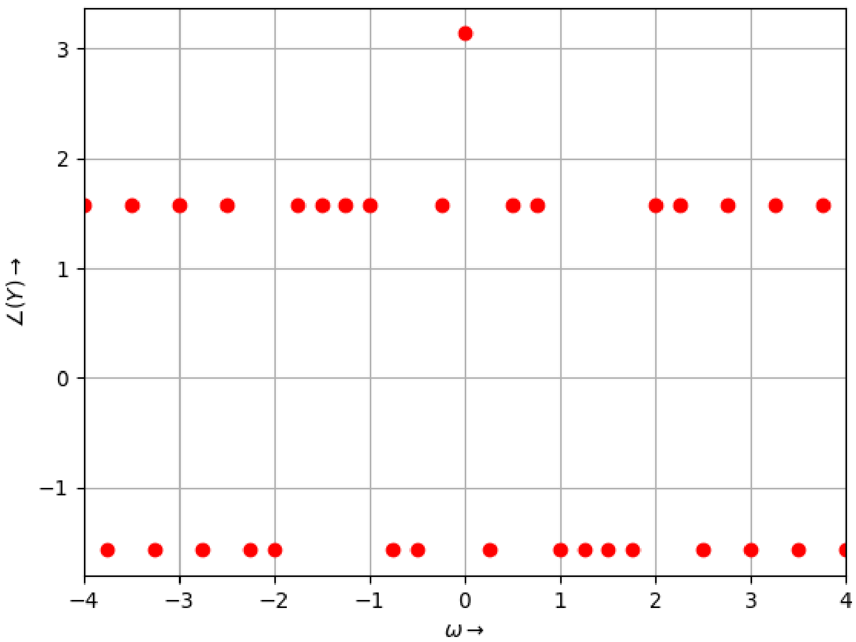

Figure 10: Phase of $\sin\sqrt{2t}$*w(t)

All the above curves are implemented in python as follows:

Code

```
t1 = linspace(-pi,pi,65); t1=t1[:-1]
dt = t1[1]-t1[0]
fmax=1/dt
y1 = sin(sqrt(2)*t1)
y1[0] = 0
w1 = linspace(-pi*fmax,pi*fmax,65); w1=w1[:-1]
y1 = fftshift(y1)
Y1 = fftshift(fft(y1))/64

figure(1)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|\rightarrow$")
title("Spectrum of $\sin(\sqrt{2}t)$")
xlim([-10,10])
plot(w1,abs(Y1),lw=2)
grid()

figure(2)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$\angle(Y)\rightarrow$")
xlim([-10,10])
```

```
plot(w1,angle(Y1),'ro',lw=2)
grid()

#Original function for which we want DFT
t2 = linspace(-3.0*pi,-1.0*pi,65); t2=t2[:-1]
t3 = linspace(pi,3.0*pi,65); t3=t3[:-1]

figure(3)
xlabel(r"$t\rightarrow$")
ylabel(r"$y\rightarrow$")
title("$sin(\sqrt{2}t)$")
xlim([-10,10])
plot(t1,sin(sqrt(2)*t1),'b',lw=2)
plot(t2,sin(sqrt(2)*t2),'r',lw=2)
plot(t3,sin(sqrt(2)*t3),'r',lw=2)
grid()

#function plot with t wrapping every 2*pi
figure(4)
xlabel(r"$t\rightarrow$")
ylabel(r"$y\rightarrow$")
title("$\sin(\sqrt{2}t)$ with t wrapping every 2$\pi$")
xlim([-10,10])
plot(t1,sin(sqrt(2)*t1),'bo',lw=2)
plot(t2,sin(sqrt(2)*t1),'ro',lw=2)
plot(t3,sin(sqrt(2)*t1),'ro',lw=2)
grid()

#For spectrum of a digital ramp
y2 = t1
y2[0]=0
y2=fftshift(y2)
Y2=fftshift(fft(y2))/64

figure(5)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|$(dB)$\rightarrow$")
title("Spectrum of a digital ramp")
xlim([1,20])
ylim([-20,0])
semilogx(abs(w1),20*log10(abs(Y2)),lw=2)
xticks([1,2,5,10],["1","2","5","10"])
grid()

#sin(sqrt(2)t)*w(t)
n=arange(64)
wnd=fftshift(0.54+0.46*cos(2*pi*n/63))
y=sin(sqrt(2)*t1)*wnd

figure(6)
xlabel(r"$t\rightarrow$")
```

```python
ylabel(r"$y\rightarrow$")
title("$\sin(\sqrt{2}t)$*w(t) with t wrapping every 2$\pi$")
xlim([-10,10])
plot(t1,y,'bo',lw=2)
plot(t2,y,'ro',lw=2)
plot(t3,y,'ro',lw=2)
grid()

#Spectrum of sin(sqrt(2)*t)*w(t)
y[0]=0
y=fftshift(y)
Y=fftshift(fft(y))/64
w = linspace(-pi*fmax,pi*fmax,65); w=w[:-1]

figure(7)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|\rightarrow$")
title("Spectrum of $\sin(\sqrt{2}t)$*w(t)")
xlim([-8,8])
plot(w,abs(Y),'b',lw=2)
grid()

figure(8)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$\angle(Y)\rightarrow$")
xlim([-8,8])
plot(w,angle(Y),'ro',lw=2)
grid()

# Spectrum plot with more number of points and narrower window of analysis
t4=linspace(-4*pi,4*pi,257);t4=t4[:-1]
dt2=t4[1]-t4[0];fmax2=1/dt2
n2=arange(256)
wnd2=fftshift(0.54+0.46*cos(2*pi*n2/256))
y4=sin(sqrt(2)*t4)
y4=y4*wnd2
y4[0]=0
y4=fftshift(y4)
Y4=fftshift(fft(y4))/256.0
w4=linspace(-pi*fmax2,pi*fmax2,257);w4=w4[:-1]

figure(9)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|\rightarrow$")
title("Spectrum of $sin(sqrt{2}t)*w(t)$")
xlim([-4,4])
plot(w4,abs(Y4),'b',lw=2)
grid()

figure(10)
xlabel(r"$\omega\rightarrow$")
```

```
ylabel(r"$\angle(Y)\rightarrow$")
xlim([-4,4])
plot(w4,angle(Y4),'ro',lw=2)
grid()
```

Q2. Spectrum of $\cos\omega_0 t$

In this question we were supposed to find the DFT of $\cos\omega_0 t$ for ω_0=0.86 for two cases, with and without Hamming window.

1. Without Hamming Window

This is done in python as follows:

```
##Spectrum of cos(0.86*t)**3 with and without hamming window
y5=cos(0.86*t4)**3
y5[0]=0
y5=fftshift(y5)
Y5=fftshift(fft(y5))/256

## Plots without windowing
figure(11)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|\rightarrow$")
title("Spectrum of cos^${3}$(0.86t) without hamming window")
xlim([-6,6])
plot(w4,abs(Y5),marker='o',color='b',lw=2)
grid()

figure(12)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$\angle(Y)\rightarrow$")
xlim([-6,6])
plot(w4,angle(Y5),'ro',lw=2)
grid()
```

we get the following plots:

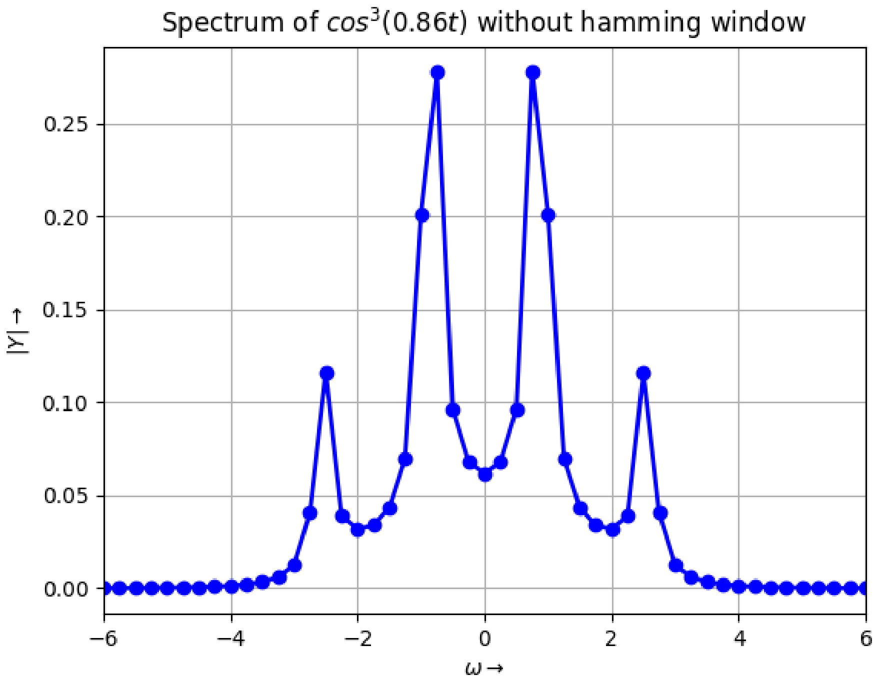

Figure 11: Spectrum of $\cos^3(0.86t)$ without Hamming window

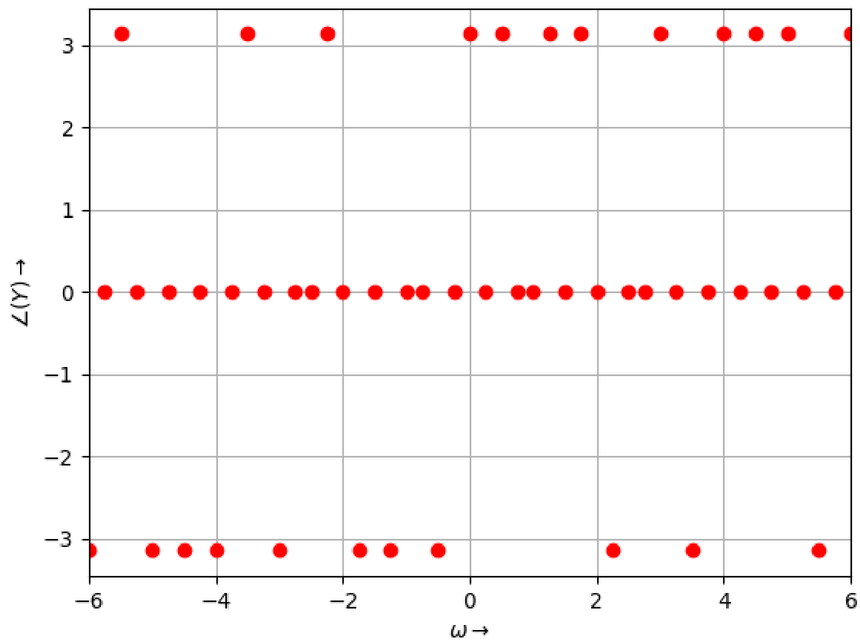

Figure 12: Phase plot of $\cos^3(0.86t)$ without Hamming window

1. **With Hamming Window**

This is done in python as follows:

```
y5=y5*wnd2
y5[0]=0
y5=fftshift(y5)
Y5=fftshift(fft(y5))/256

## Plots with windowing
figure(13)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|\rightarrow$")
title("Spectrum of cos^{3}(0.86t) with hamming window")
xlim([-6,6])
plot(w4,abs(Y5),marker='o',color='b',lw=2)
grid()

figure(14)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$\angle(Y)\rightarrow$")
xlim([-6,6])
plot(w4,angle(Y5),'ro',lw=2)
grid()
```

We get the following plots:

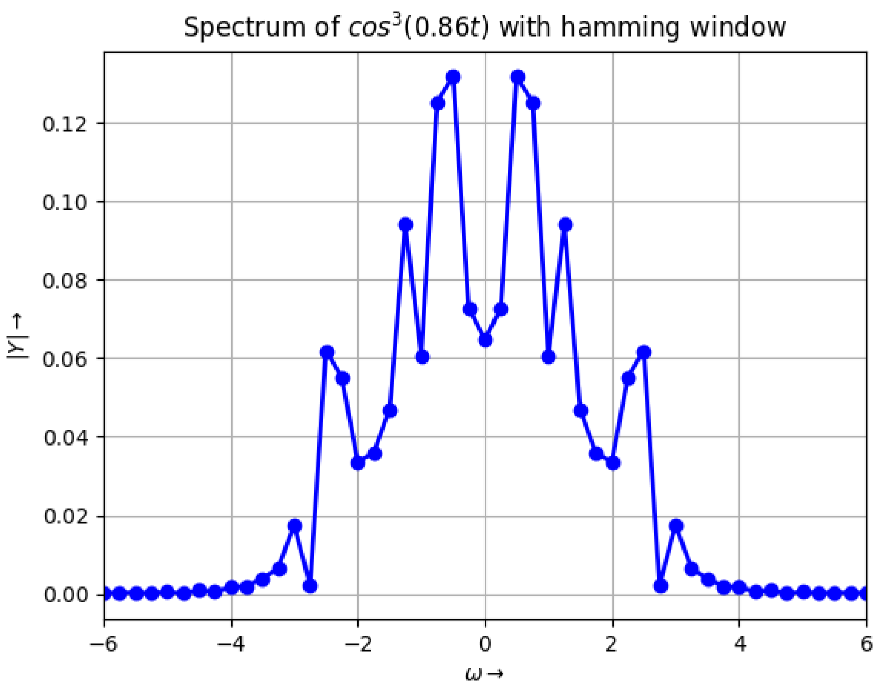

Figure 13: Spectrum of $\cos^3(0.86t)$ with Hamming window

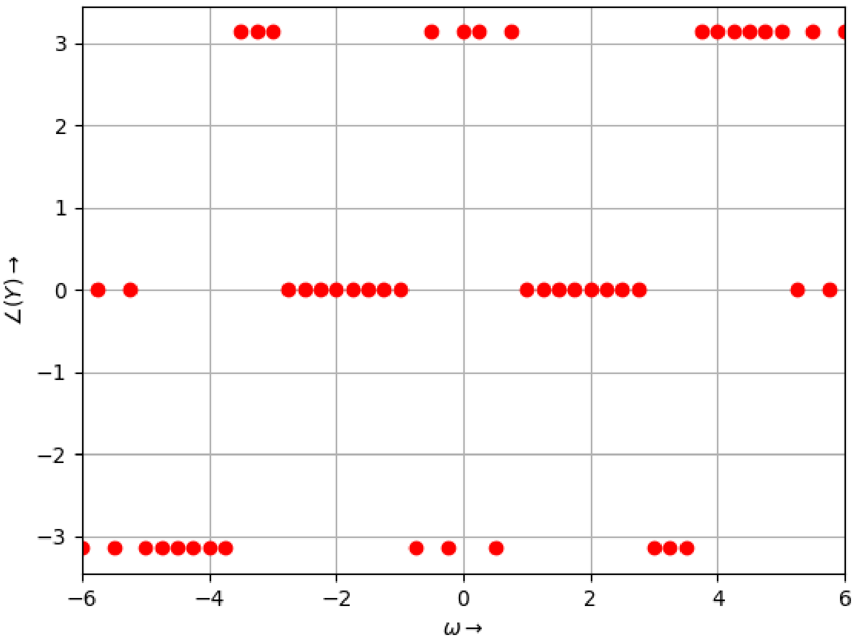

Figure 14: Phase plot of $\cos^3(0.86t)$ with Hamming window

Q3. Estimating ω_0 and δ

In this question we need to estimate ω and δ for the signal $\cos(\omega t + \delta)$ for 128 samples between $[-\pi,\pi]$. For this problem we estimate ω using a weighted average and for δ we consider a window on each half of ω and extract the mean slope value because a circular shift in time domain a sequence results in the linear phase of the spectra.

This is done in python as follows:

```
# Let w0 = 1.5 and delta = 0.5.
w0 = 1.5
d = 0.5

t6 = linspace(-pi,pi,129)[:-1]
dt6 = t6[1]-t6[0]; fmax6 = 1/dt6
n6 = arange(128)
wnd3 = fftshift(0.54+0.46*cos(2*pi*n6/128))
y6 = cos(w0*t6 + d)
y6 = y6*wnd3
y6[0]=0
y6 = fftshift(y6)
Y6 = fftshift(fft(y6))/128.0
w2 = linspace(-pi*fmax6,pi*fmax6,129); w2 = w2[:-1]

figure(15)
```

```
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|\rightarrow$")
title(r"Spectrum of $\cos(w_0t+\delta)$ with Hamming window")
xlim([-4,4])
plot(w2,abs(Y6),'b',lw=2)
grid()

figure(16)
xlabel(r"$\omega\rightarrow$")
ylabel(r"Phase of $Y\rightarrow$")
title(r"Spectrum of $\cos(w_0t+\delta)$ with Hamming window")
xlim([-4,4])
plot(w2,angle(Y6),'ro',lw=2)
grid()

# w0 is calculated by finding the weighted average of all w>0. Delta is found by calculating the phase at w closest to w0.
ii = where(w2>=0)
w_cal = sum(abs(Y6[ii])**2*w2[ii])/sum(abs(Y6[ii])**2)
i = abs(w2-w_cal).argmin()
delta = angle(Y6[i])
print("--------------------------------------------------------------------------")
print("Calculated value of w0 without noise: ",w_cal)
print("Calculated value of delta without noise: ",delta)
print("--------------------------------------------------------------------------")
```

For this we obtain the following curves:

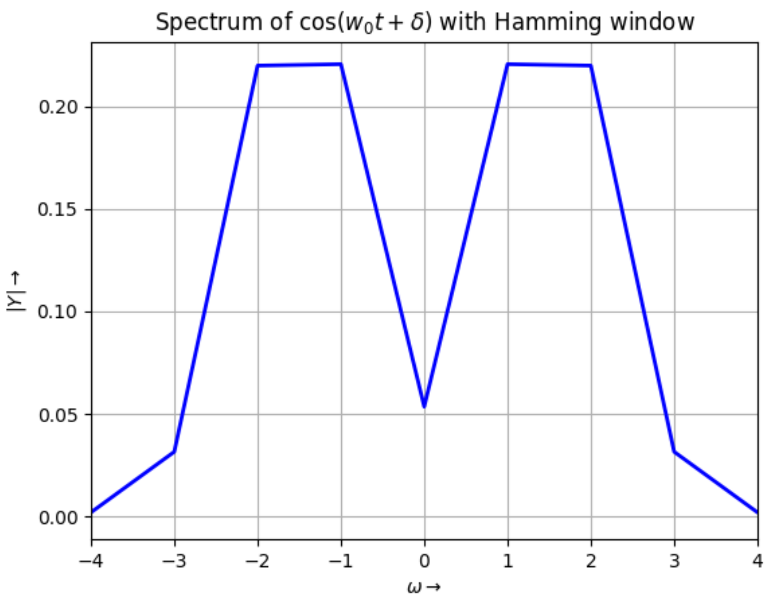

Figure 15: Spectrum of $\cos(\omega_0 t + \delta)$

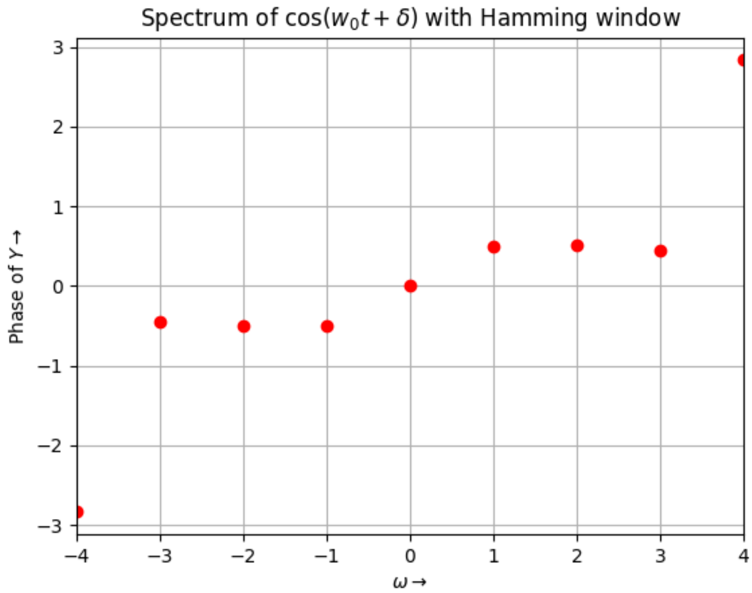

Figure 16: Phase plot of $\cos(\omega_0 t + \delta)$

We obtain the following ω_0 and δ from our computations:

```
-------------------------------------------------------------------------------
Calculated value of w0 without noise:  1.4730276250507859
Calculated value of delta without noise:  0.5018760117245951
-------------------------------------------------------------------------------
```

Q4. Estimating ω_0 and δ for white gaussian noise

For this question we will add the white gaussian noise to our original cosinusoidal signal and then try to estimate the values of ω_0 and δ. This gaussian white noise can be generated in our signal with the help of rand() function of python.

$$f(t) = cos(\omega_0 t + \delta + 0.1rand(N))$$

```
y7 = (cos(w0*t6 + d) + 0.1*randn(128)) # noisy signal created using rand function
     of python
y7 = y7*wnd3
y7[0]=0
y7 = fftshift(y7)
Y7 = fftshift(fft(y7))/128.0

figure(17)
xlabel(r"$\omega\rightarrow$")
```

```
ylabel(r"$|Y|\rightarrow$")
title(r"Spectrum of $\cos(w_0t+\delta + 0.1*rand(128))$ with Hamming window")
xlim([-4,4])
plot(w2,abs(Y7),'b',lw=2)
grid()

figure(18)
xlabel(r"$\omega\rightarrow$")
ylabel(r"Phase of $Y\rightarrow$")
title(r"Spectrum of $\cos(w_0t+\delta + 0.1*rand(128))$ with Hamming window")
xlim([-4,4])
plot(w2,angle(Y7),'ro',lw=2)
grid()


# w0 is calculated by finding the weighted average of all w>0. Delta is found by
    calculating the phase at w closest to w0.
ii2 = where(w2>=0)
w_cal2 = sum(abs(Y7[ii])**2*w2[ii2])/sum(abs(Y7[ii2])**2)
i = abs(w2-w_cal2).argmin()
delta2 = angle(Y7[i])
print("Calculated value of w0 with noise: ",w_cal2)
print("Calculated value of delta with noise: ",delta2)
print("-----------------------------------------------------------------------")
```

We obtain the following spectrum and phase plots for the given function:

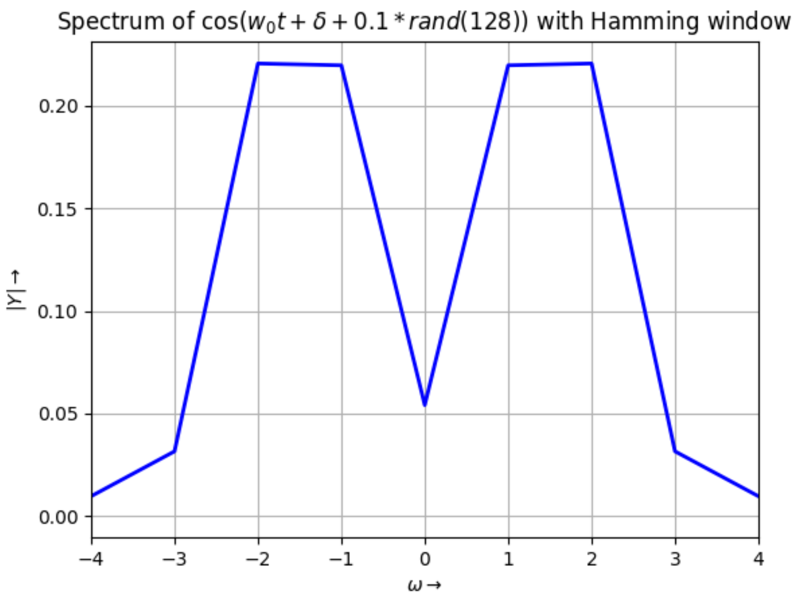

Figure 17: Spectrum of $\cos(\omega_0 t + \delta + 0.1 rand(128))$

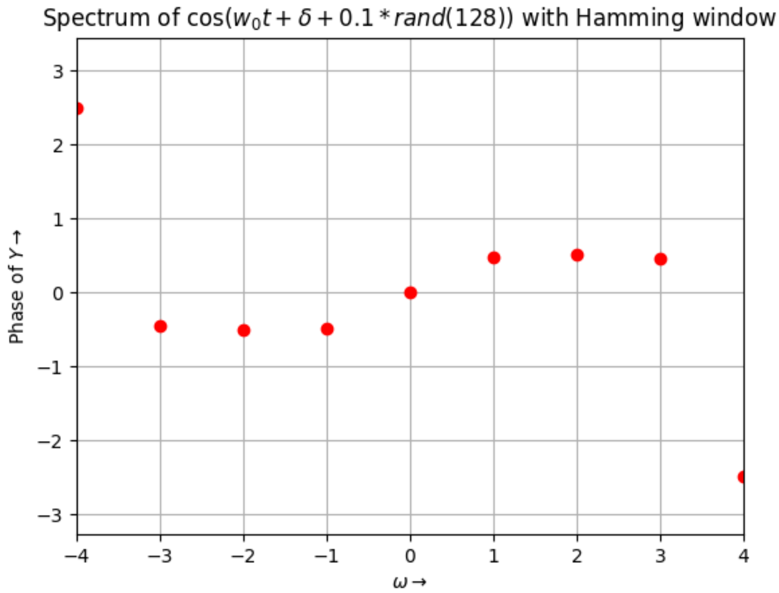

Figure 18: Phase plot of $\cos(\omega_0 t + \delta + 0.1 rand(128))$

From our computation we get the following values of ω_0 and δ:

```
 ---------------------------------------------------------------------------
Calculated value of w0 with noise:  2.034773596362756
Calculated value of delta with noise:  0.4886090045045165
----------------------------------------------------------------------------
```

Q5. Spectrum of Chirped Signal

In this question we are supposed to plot the spectrum and phase plot of chirped signal g(t)

$$g(t) = cos(16(1.5 + \frac{t}{2\pi})t)$$

for t going from [-π,π] in 1024 steps. This is done in python as follows:

```
t7 = linspace(-pi,pi,1025); t7=t7[:-1]
dt7 = t7[1] - t7[0]; fmax7 = 1/dt7
w3 = linspace(-pi*fmax7,pi*fmax7,1025); w3=w3[:-1]
n7 = arange(1024)
wnd4 = fftshift(0.54+0.46*cos(2*pi*n7/1024))
y8=cos(16*(1.5+(t7/(2.0*pi)))*t7)
y8 = y8*wnd4
y8[0]=0
y8=fftshift(y8)
```

```
Y8=fftshift(fft(y8))/1024

figure(19)
xlabel(r"$\omega\rightarrow$")
ylabel(r"$|Y|\rightarrow$")
title(r"Spectrum of chirped function with Hamming window")
xlim([-60,60])
plot(w3,abs(Y8),'b',lw=2)
grid()

figure(20)
xlabel(r"$\omega\rightarrow$")
ylabel(r"Phase of $Y\rightarrow$")
title("phase of chirped function with Hamming window")
xlim([-60,60])
plot(w3,angle(Y8),'ro',lw=2)
grid()
```

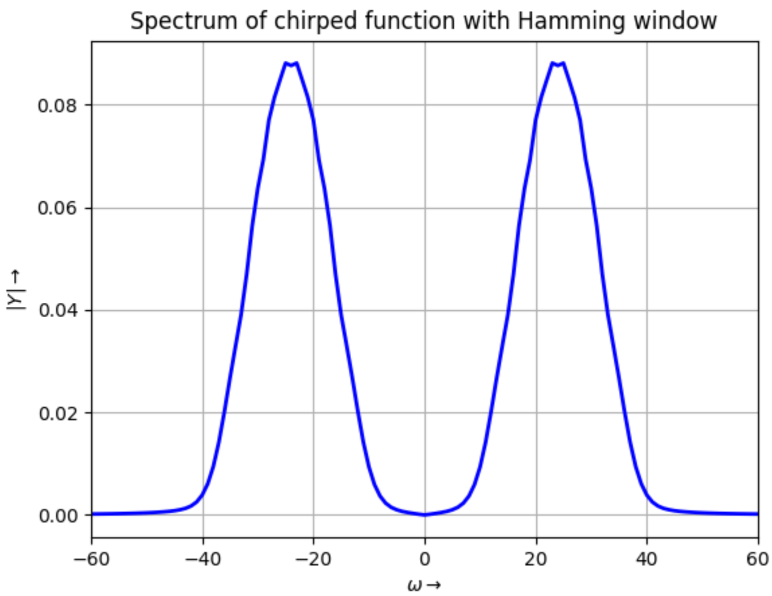

Figure 19: Spectrum of chirped function g(t)

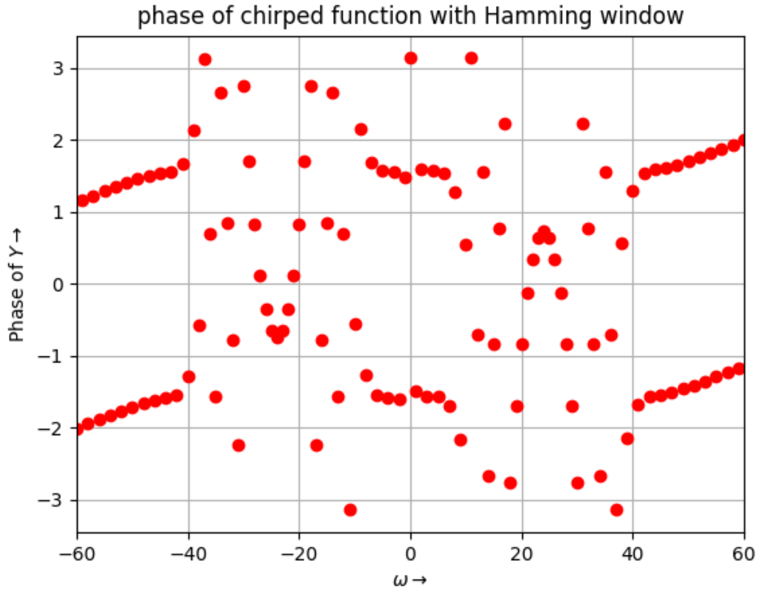

Figure 20: Phase plot of chirped function g(t)

Q6. 3D surface plot of chirped function

In this question, for the same chirped signal, we break the 1024 vector into pieces that are 64 samples wide and then extract the DFT of each and store as a column in a 2D array. Then we plot the array as a surface plot to show how the frequency of the signal varies with time.

This is done in python as follows:

```
t_array = split(t7,16)
Y_mag = zeros((16,64))
Y_phase = zeros((16,64))

for i in range(len(t_array)):
  n = arange(64)
  wnd = fftshift(0.54+0.46*cos(2*pi*n/64))
  y = cos(16*t_array[i]*(1.5 + t_array[i]/(2*pi)))*wnd
  y[0]=0
  y = fftshift(y)
  Y = fftshift(fft(y))/64.0
  Y_mag[i] = abs(Y)
  Y_phase[i] = angle(Y)

t7 = t7[::64]
w = linspace(-fmax*pi,fmax*pi,64+1); w = w[:-1]
t7,w = meshgrid(t7,w)

fig1 = figure(21)
ax = fig1.add_subplot(111, projection='3d')
```

```
surf=ax.plot_surface(w,t7,Y_mag.T,cmap='viridis',linewidth=0, antialiased=False)
fig1.colorbar(surf, shrink=0.5, aspect=5)
ax.set_title('surface plot');
ylabel(r"$\omega\rightarrow$")
xlabel(r"$t\rightarrow$")
show()
```

We obtain the following 3D surface plot as a result:

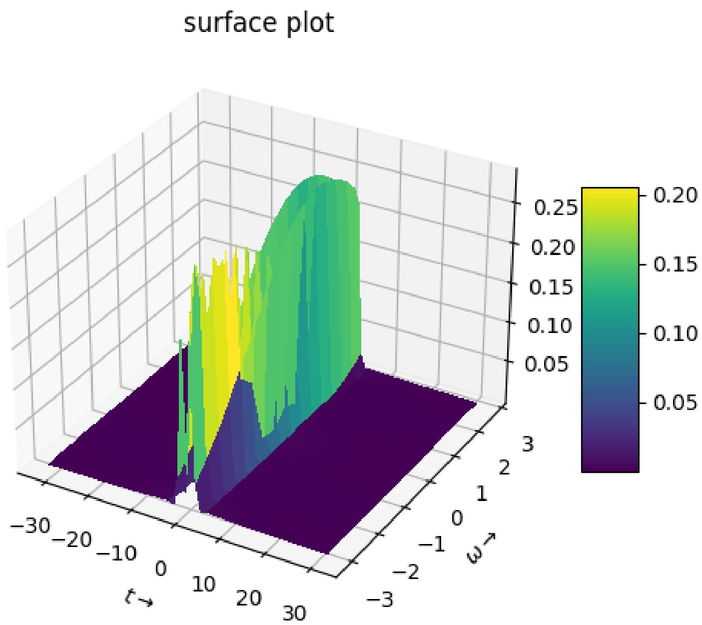

Figure 21: Windowed chopped chirped function

CONCLUSIONS

Hence by windowing we realized DFT of non periodic signals. Also the chirped function was analyzed and it's time frequency plot showed the gradual variation of peak frequency of the spectrum with time.