EE2703 : Applied Programming Lab
Week9 : Spectra of non-periodic signals

Aditya Suresh
EE20B004

Tuesday 12^{th} April, 2022

AIM

- To Find DFT of non-periodic signals.
- Understanding the use of windowing functions in helping the implementation of DFT for non-periodic signals.
- Plot and analyse the result.

Theory

- We will use similar concepts as in Assignment 8 to calculate DFT of non periodic signals
- Sample the signal so that $f_{Nyquist}$ is met, and so that Δf is small enough. Generate the frequency axis from $f_{max}/2$ to $+f_{max}/2$, taking care to drop the last term.
- Ensure that the signal starts at t = 0^+ and ends at t = 0^-
- Use 2^k samples
- Obtain the DFT. Rotate the samples so that they go from $f_{max}/2$ to $+f_{max}/2 - \Delta f$.
- Plot the magnitude and phase of the spectrum.
- We will explore producing DFT's with windowing.You can minimize the effects of performing an DFT over a noninteger number of cycles by using a technique called windowing. Windowing reduces the amplitude of the discontinuities at the boundaries of each finite sequence acquired by the digitizer

Formula

We will derive an important property of DFT below: Let's take an asymmetric function y[n] and see how its DFT looks like

$$Y[k] = \sum_{n=0}^{N-1} y[n]exp(\frac{2\pi j}{N}kn)$$

$$= \sum_{n=1}^{N/2-1} y[n](exp(\frac{2\pi j}{N}kn) - exp(-\frac{2\pi j}{N}kn)) + y[\frac{N}{2}]exp(\pi kj)$$

$$= \sum_{n=1}^{N/2-1} -2jy[n]sin(\frac{2\pi}{N}kn) + (-1)^k y[\frac{N}{2}]$$

To produce a purely imaginary DFT we set y[N/2] to zero.

Question 1 :Plotting the given examples

1. The spectrum of $sin(\sqrt{2}t)$ is :

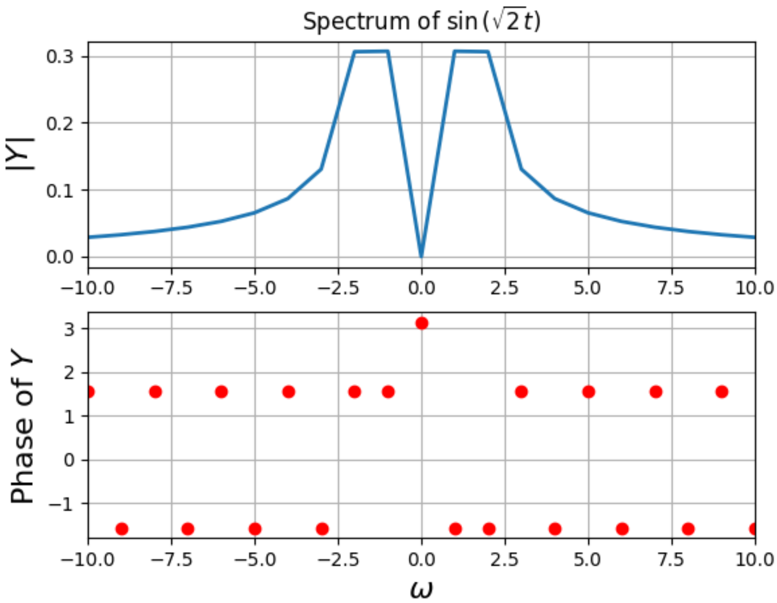

Figure 1: spectrum of $sin(\sqrt{2}t)$

2. And is obtained via the following code:

```
t = linspace(-pi,pi,65)[:-1]
dt=t[1]-t[0];fmax=1/dt
y=sin(sqrt(2)*t)
y[0]=0 # the sample corresponding to -tmax should be set zeroo
y=fftshift(y) # make y start with y(t=0)
Y=fftshift(fft(y))/64.0
w=linspace(-pi*fmax,pi*fmax,65);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-10,10])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $\sin\left(\sqrt{2}t\right)$")
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
xlim([-10,10])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
savefig("fig0.png")
show()
```

3. As can be seen from above the graph the magnitude response is incorrect as we expect well defined spikes which we expect in case of sine waves is not present. To understand the reason for this we will plot extensions of $sin(\sqrt{2}t)$ using the following code:

```
t1=linspace(-pi,pi,65);t1=t1[:-1]
t2=linspace(-3*pi,-pi,65);t2=t2[:-1]
t3=linspace(pi,3*pi,65);t3=t3[:-1]
figure(2)
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)
ylabel(r"$y$",size=16)
xlabel(r"$t$",size=16)
title(r"$\sin\left(\sqrt{2}t\right)$")
grid(True)
```

```
savefig("fig10-2.png")
show()
```

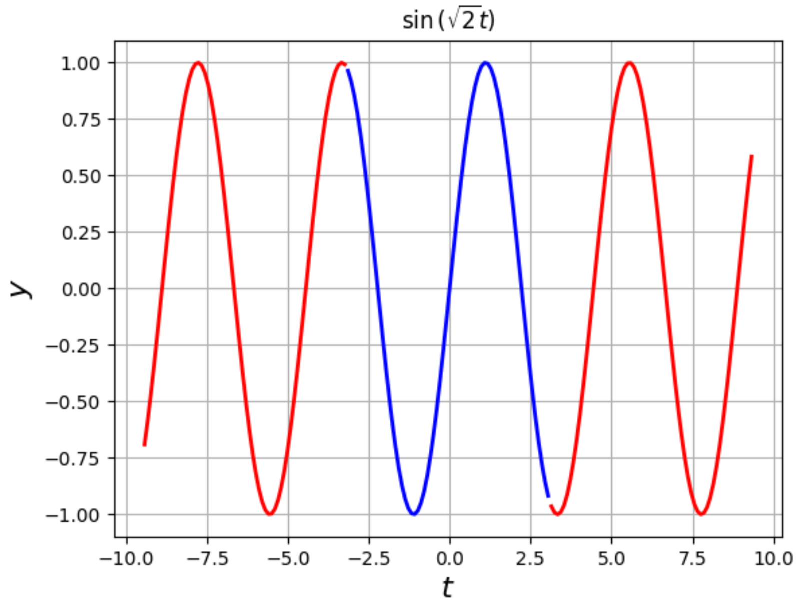

Figure 2: $sin(\sqrt{2}t)$

4. As can be seen the value of $sin(\sqrt{2}t)$ between $-\pi$ to π cant be used to replicate the function

5. Below the is the function of which we obtained DFT in Figure 1. The following is the code used for the same

```
y=sin(sqrt(2)*t1)
figure(3)
plot(t1,y,'bo',lw=2)
plot(t2,y,'ro',lw=2)
plot(t3,y,'ro',lw=2)
ylabel(r"$y$",size=16)
xlabel(r"$t$",size=16)
title(r"$\sin\left(\sqrt{2}t\right)$ with $t$ wrapping every $2\pi$ ")
grid(True)
savefig("fig10-3.png")
show()
```

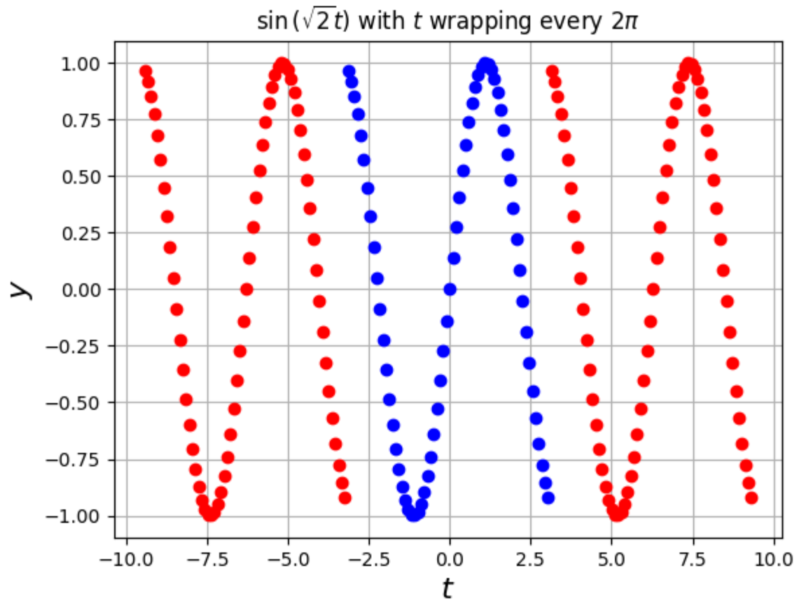

Figure 3: Periodic extension of $sin(\sqrt{2}t)$ in between $-\pi$ to π

6. The above is clearly not $sin(\sqrt{2}t)$.To understand why we dont see spikes in magnitude plot in Figure-1, Lets plot the magnitude-phase response of a ramp function, generated via the following code:

```
y=t
y[0]=0 # the sample corresponding to -tmax should be set zeroo
y=fftshift(y) # make y start with y(t=0)
Y=fftshift(fft(y))/64.0
w=linspace(-pi*fmax,pi*fmax,65);w=w[:-1]
figure()
semilogx(abs(w),20*log10(abs(Y)),lw=2)
xlim([1,10])
ylim([-20,0])
xticks([1,2,5,10],["1","2","5","10"],size=16)
ylabel(r"$|Y|$ (dB)",size=16)
title(r"Spectrum of a digital ramp")
xlabel(r"$\omega$",size=16)
grid(True)
savefig("fig10-4.png")
show()
```

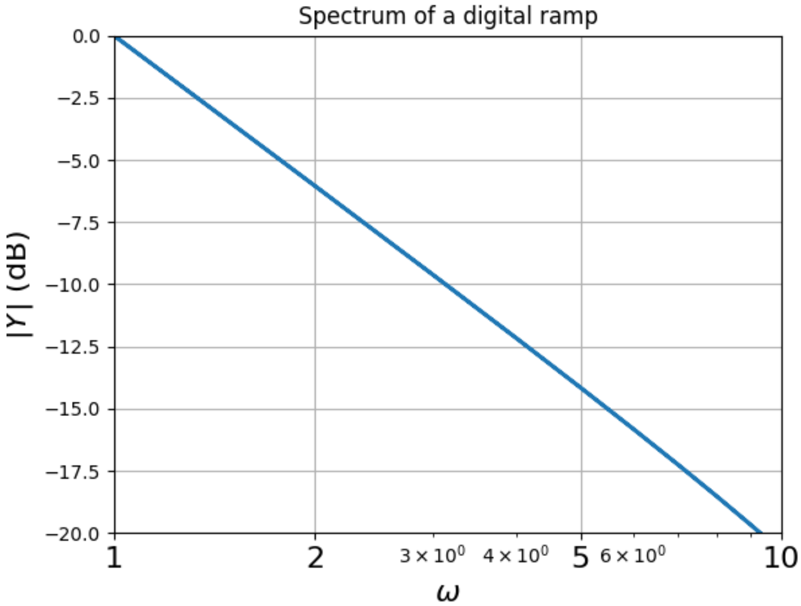

Figure 4: spectrum of digital ramp

7. As can be seen the slope of decay is -20dB , showing that Magnitude response falls as $\frac{1}{\omega}$.The big jumps at $n\pi$ force this slowly decaying spectrum, which is why we don't see the expected spikes for the spectrum of $sin(\sqrt{2}t)$

8. **Windowing**: The spikes happen at the end of the periodic interval. So we damp the function near there, i.e., we multiply our function sequence f [n] by a "window" sequence w[n].This is done by following code and corresponding time domain graph is as follows:

```
n=arange(64)
wnd=fftshift(0.54+0.46*cos(2*pi*n/63))
y=sin(sqrt(2)*t1)*wnd
figure(3)
plot(t1,y,'bo',lw=2)
plot(t2,y,'ro',lw=2)
plot(t3,y,'ro',lw=2)
ylabel(r"$y$",size=16)
xlabel(r"$t$",size=16)
title(r"$\sin\left(\sqrt{2}t\right)\times w(t)$ with $t$ wrapping every $2\p
grid(True)
```

```
savefig("fig10-5.png")
show()
```

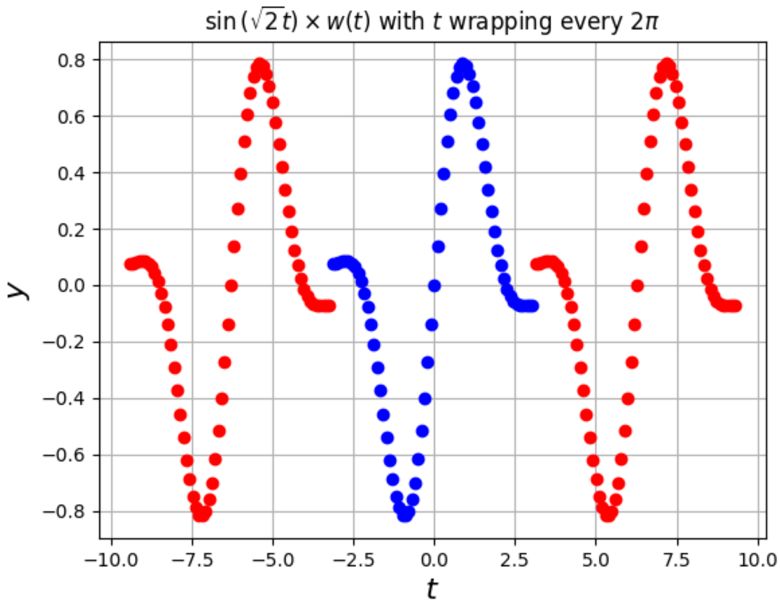

Figure 5: $sin(\sqrt{2}t)$ with windowing

9. The jump is still there, but it is much reduced. There is also a benefit of keeping some jumps- it gives us an extra 10 db of suppression.Now The DFT then will be calculated as:

```
y=sin(sqrt(2)*t)*wnd
y[0]=0 # the sample corresponding to -tmax should be set zeroo
y=fftshift(y) # make y start with y(t=0)
Y=fftshift(fft(y))/64.0
w=linspace(-pi*fmax,pi*fmax,65);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-8,8])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $\sin\left(\sqrt{2}t\right)\times w(t)$")
grid(True)
subplot(2,1,2)
```

```
plot(w,angle(Y),'ro',lw=2)
xlim([-8,8])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
savefig("fig10-6.png")
show()
```

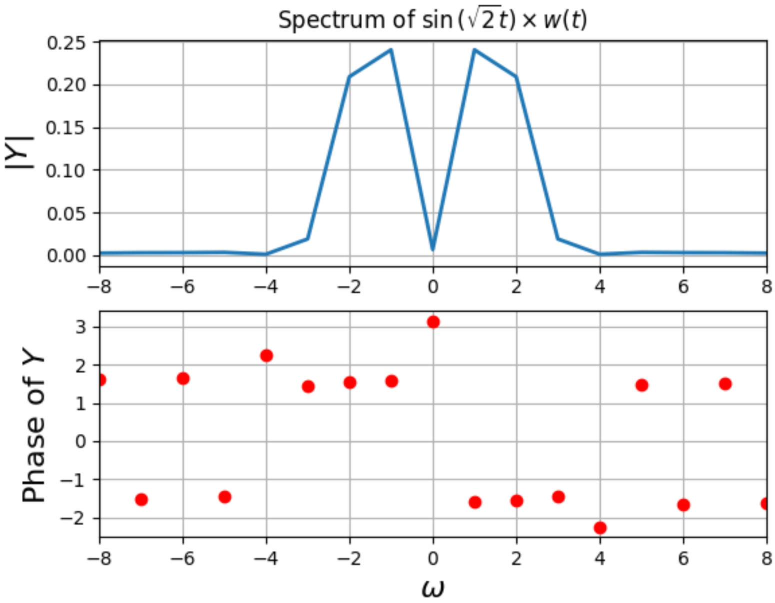

Figure 6: DFT of $sin(\sqrt{2}t)$ with windowing

plotting the same with increased number of samples:

```
t4=linspace(-4*pi,4*pi,257);t4=t4[:-1]
dt4=t4[1]-t4[0];fmax1=1/dt4
n1=arange(256)
wnd1=fftshift(0.54+0.46*cos(2*pi*n1/256))
y=sin(sqrt(2)*t4)
y=y*wnd1
y[0]=0 # the sample corresponding to -tmax should be set zeroo
y=fftshift(y) # make y start with y(t=0)
Y=fftshift(fft(y))/256.0
w1=linspace(-pi*fmax1,pi*fmax1,257);w1=w1[:-1]
figure()
```

```
subplot(2,1,1)
plot(w1,abs(Y),'b',w1,abs(Y),'bo',lw=2)
xlim([-4,4])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $\sin\left(\sqrt{2}t\right)\times w(t)$")
grid(True)
subplot(2,1,2)
plot(w1,angle(Y),'ro',lw=2)
xlim([-4,4])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
savefig("fig10-7.png")
show()
```

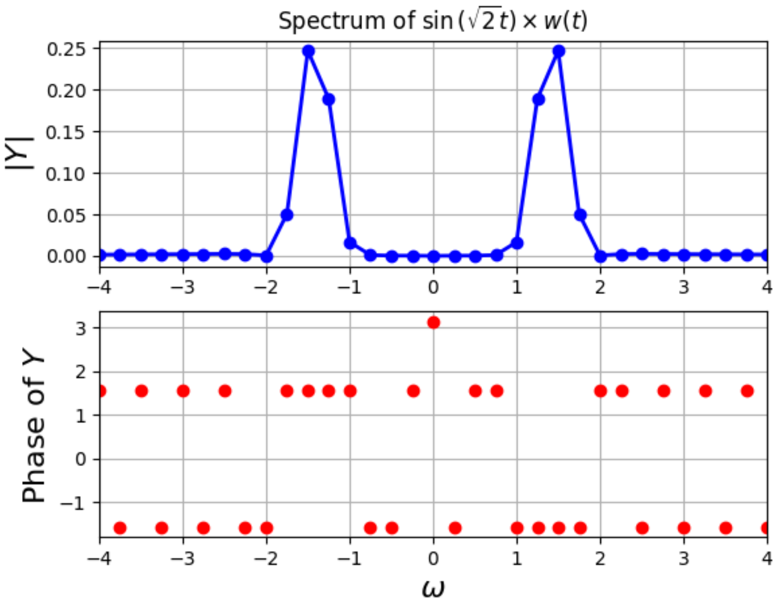

Figure 7: DFT of $sin(\sqrt{2}t)$ with windowing(increased samples)

Question 2: Spectrum of $cos^3(\omega_0 t)$

1. First we plot the spectrum of $cos^3(\omega_0 t)$ without windowing.The following is the code used and the plot obtained

```
t = linspace(-4*pi,4*pi,257); t = t[:-1]
w0 = 0.86
dt = t[1]-t[0]; fmax = 1/dt
n = arange(256)
wnd = fftshift(0.54+0.46*cos(2*pi*n/256))
y = (cos(w0*t))**3
y[0] = 0
y = fftshift(y)
Y=fftshift(fft(y))/256
w = linspace(-pi*fmax,pi*fmax,257); w = w[:-1]
figure(2)
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-4,4])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $\cos^{3}\left(\omega_{0}t\right)$")
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
xlim([-4,4])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
savefig("fig1.png")
```

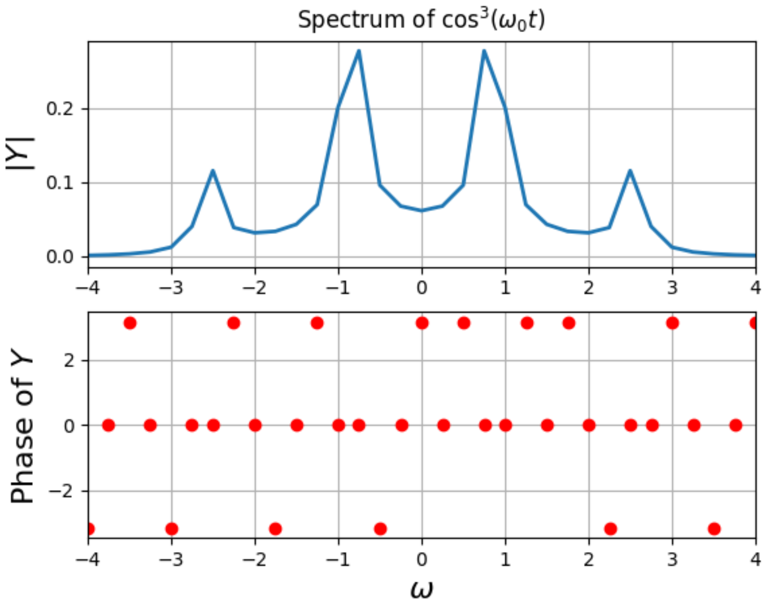

Figure 8: DFT of $cos^3(\omega_0 t)$

2. And the same for with windowing:

```
y = cos(w0*t)**3
y = y*wnd
y[0] = 0
y = fftshift(y)
Y = fftshift(fft(y))/256.0
figure(3)
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-4,4])
ylabel(r"$|Y|\rightarrow$",size=16)
title(r"Spectrum of $\cos^{3}(\omega_{0}t)$ with Hamming window")
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
xlim([-4,4])
ylabel(r"Phase of $Y\rightarrow$",size=16)
xlabel(r"$\omega\rightarrow$",size=16)
grid(True)
savefig("fig2.png")
```

```
show()
```

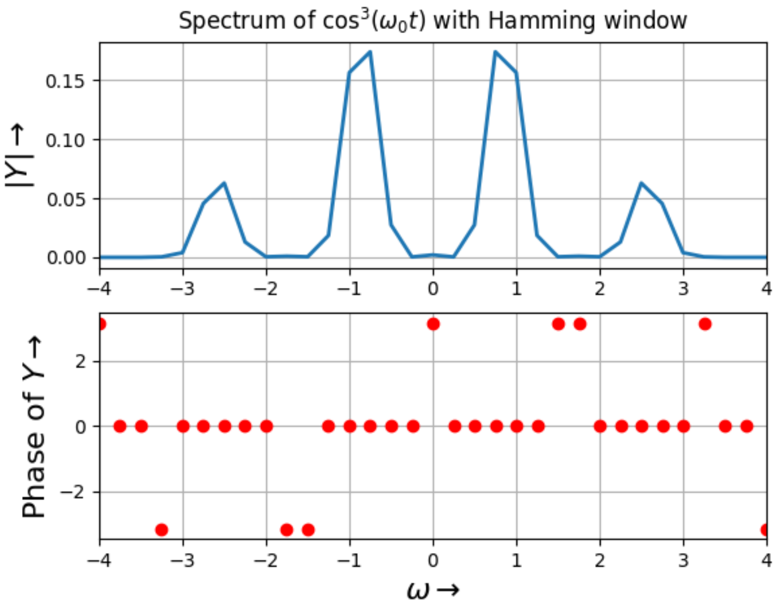

Figure 9: DFT of $cos^3(\omega_0 t)$ with windowing

Question 3: Calculating ω_0 and δ

1. We obtain ω using weighted average of Magnitude plot,i.e.:

$$\frac{\sum_{\omega>0} \omega |Y(\omega)|^2}{\sum_{\omega>0} |Y(\omega)|^2}$$

```
indexes = where(w>=0)
w_cal = sum(abs(Y[indexes])**2*w[indexes])/sum(abs(Y[indexes])**2)
```

2. similarly δ can be found by finding angle at ω closest to ω_0

```
i = abs(w-w_cal).argmin()
delta = angle(Y[i])
```

3. The values hence obtained are:
   Calculated value of ω_0 without noise: 1.4730276250507859
   Calculated value of δ without noise: 0.5018760117245951

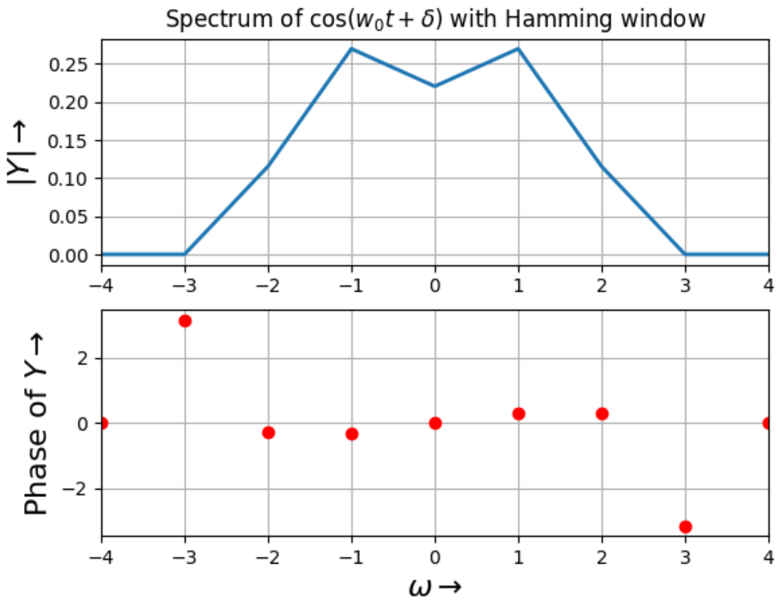

Figure 10: Spectrum of $cos(\omega_0 t + \delta)$

Question 4: Calculating ω_0 and δ with "white Gaussian noise"

1. We generate the gaussian noise using pythons 0.1*randn(N). Which is then added to y[n] , on which windowing is performed. The following is the same in code:

```
wgn = 0.1*randn(128)
y = cos(w0*t +d)*wnd + wgn*wnd
y[0]=0
y = fftshift(y)
Y = fftshift(fft(y))/128.0
```

2. Similarly calculating ω_0 and δ we get the following result:
Calculated value of w0 without noise: 1.954472250211338
Calculated value of delta without noise: 0.48891905956885434

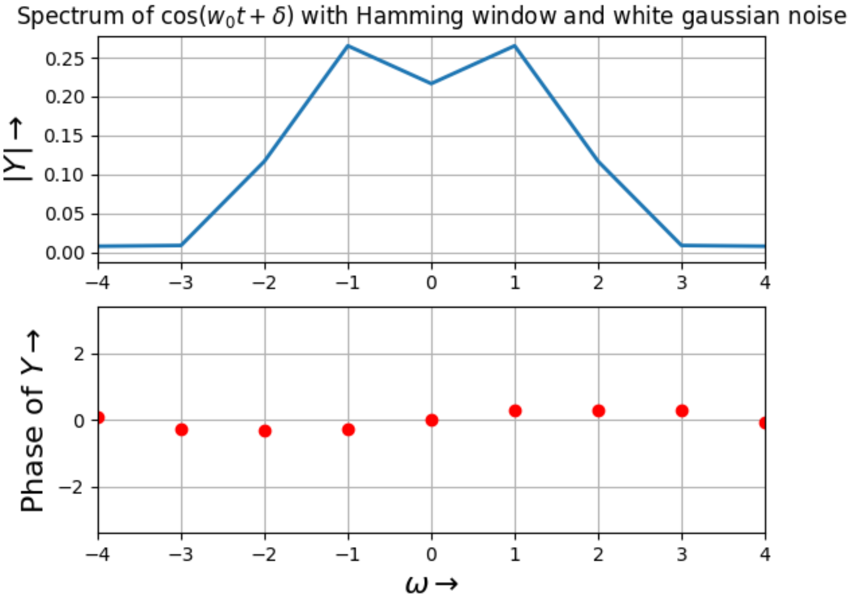

Figure 11: Spectrum of $cos(\omega_0 t + \delta)$ with noise

Question 5 : Plot DFT of chirped signal

1. We will plot the DFT of the "chirped signal" $cos(16t(1.5 + \frac{t}{2\pi}))$. The follwoing code is used for same and for plotting the DFT

```
t = linspace(-pi,pi,1025)[:-1]
dt = t[1] - t[0]; fmax = 1/dt
n = arange(1024)
wnd = fftshift(0.54 + 0.46*cos(2*pi*n/1024))
y = cos(16*(1.5+ t/(2*pi))*t)*wnd
y[0] =0
y = fftshift(y)
Y = fftshift(fft(y)) /1024
w = linspace(-pi*fmax,pi*fmax,1025)[:-1]
figure(6)
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-100,100])
ylabel(r"$|Y|\rightarrow$",size=16)
title(r"Spectrum of $\cos(16t(1.5+\frac{t}{2\pi}))$ with Hamming window")
grid(True)
```

```
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
xlim([-100,100])
ylabel(r"Phase of $Y\rightarrow$",size=16)
xlabel(r"$\omega\rightarrow$",size=16)
grid(True)
savefig("fig5.png")
show()
```

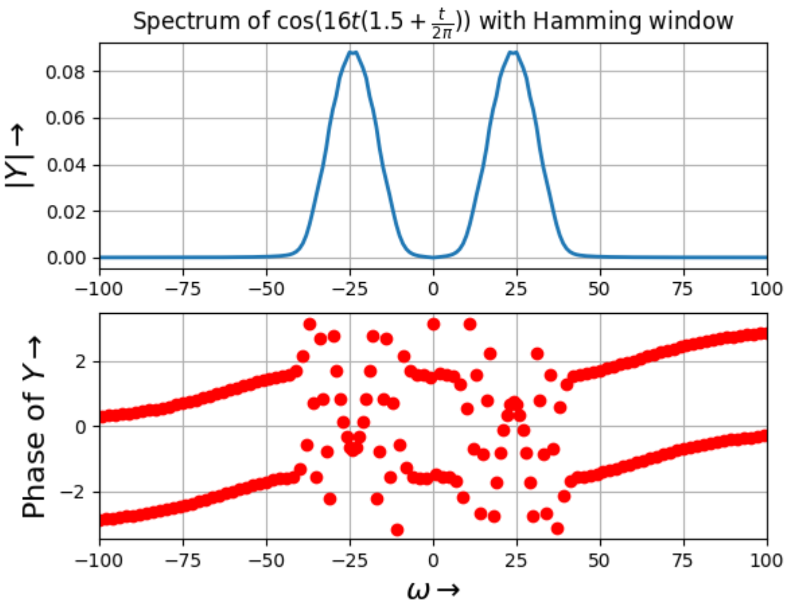

Figure 12: Spectrum of chirped signal with windowing

Qn6 : Surface Plot of chirped signal

1. We form a 2D matrix of y and corresponding DFT. We then produce the surface plot to visualize the variation of DFT with time and frequency. For thsi the following code is use:

```
n = arange(64)
wnd = fftshift(0.54 + 0.46*cos(2*pi*n/64))
Wnd = meshgrid(wnd, wnd)[0][:16]
t_arrays = np.array(split(t,16))
```

```
y_arrays = cos(16*(1.5+ t_arrays/(2*pi))*t_arrays)*Wnd
y_arrays[:0] = 0
y_arrays = fftshift(y_arrays)
Y_arrays = fftshift(fft(y_arrays))
```

2. And the corresponding surface plot is:

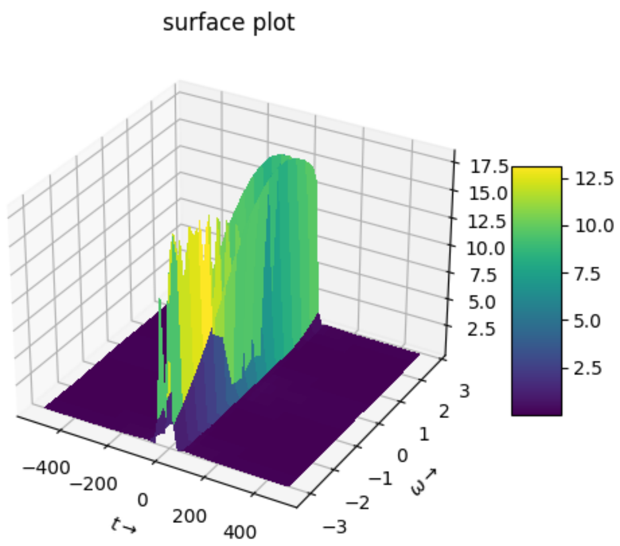

Figure 13: Surface plot of the magnitude of broken chirped signal

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.