Assignment No 9

Gali Ushasri EE20B033

1 Aim

In this assignment, we continue our analysis of signals using Fourier Transforms. This time, we focus on finding transforms of non periodic functions.

2 Examples

The worked examples in the assignment are given below: Spectrum of $sin(\sqrt{2}t)$ is given below

```
t=linspace(-pi,pi,65);t=t[:-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)
```

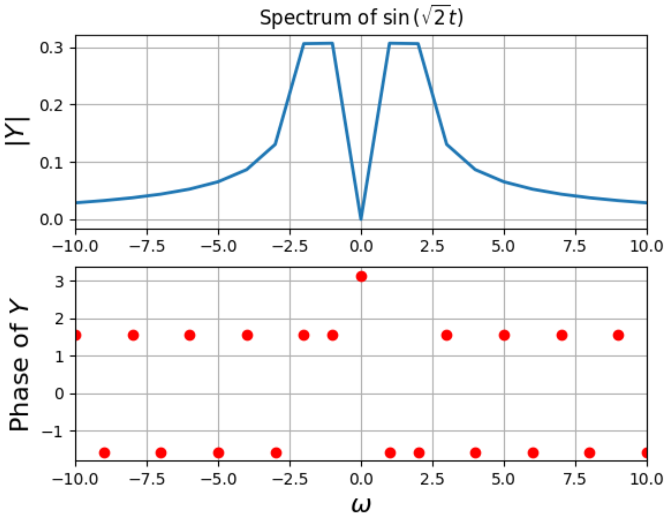


Figure 1: Spectrum of $\sin(\sqrt{2}t)$

Original function for which we want the DFT:

```
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]
# y=sin(sqrt(2)*t)
figure()
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)
```

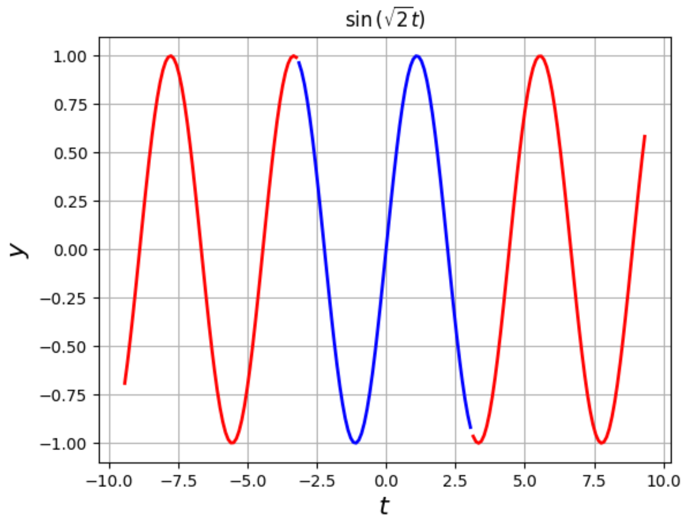


Figure 2: $\sin(\sqrt{2}t)$

As the DFT is computed over a finite time interval, we have actually plotted the DFT for this function

```
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]
y=sin(sqrt(2)*t1)
figure()
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)
```

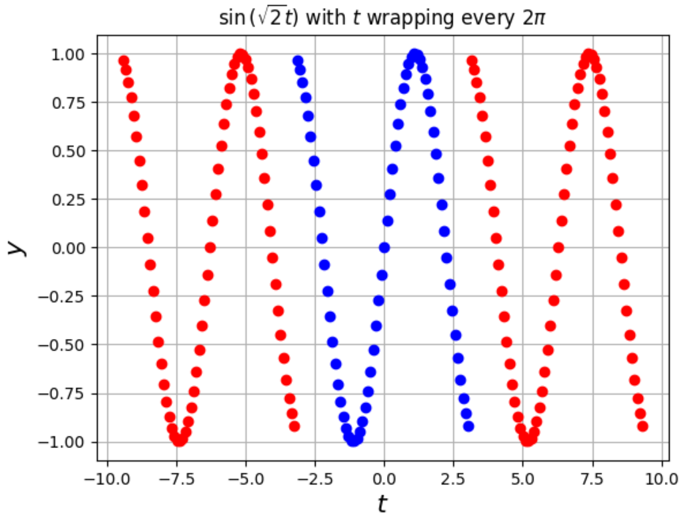


Figure 3: Spectrum of $\sin(\sqrt{2}t)$

These discontinuities lead to non harmonic components in the FFT which decay as $\frac{1}{\omega}$. To confirm this, we plot the spectrum of the periodic ramp.

```
t=linspace(-pi,pi,65);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
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)
```

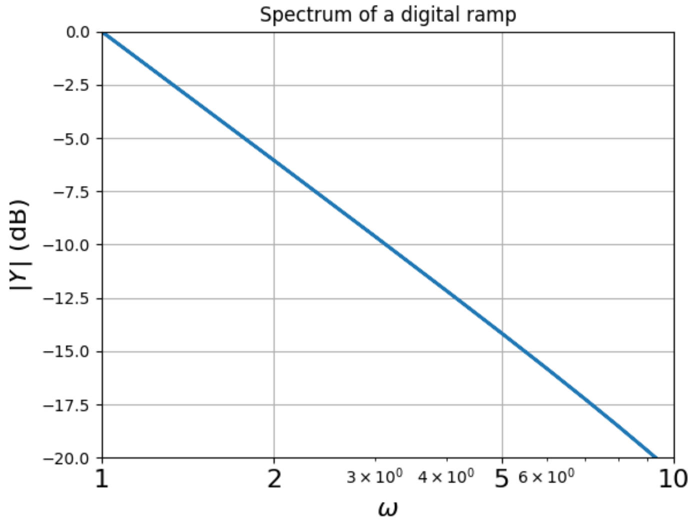


Figure 4: Spectrum of $\sin(\sqrt{2}t)$

2.1 Windowing

The hamming window removes discontinuities by attenuating the high frequency components that cause the discontinuities. The hamming window function is given by

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

We multiply our signal with the hamming window and periodically extend it. The discontinuities nearly vanish.

```
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]
n=arange(64)
wnd=fftshift(0.54+0.46*cos(2*pi*n/63))
y=sin(sqrt(2)*t1)*wnd
figure()
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 $
grid(True)
```

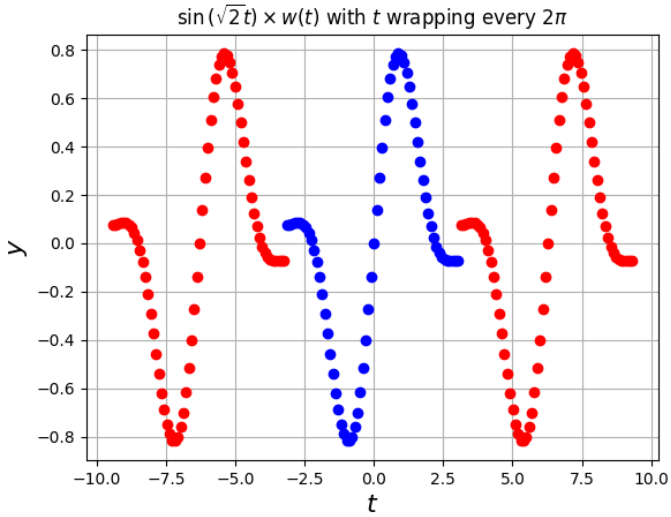


Figure 5: Spectrum of $\sin(\sqrt{2}t) * w(t)$

The spectrum that is obtained with a time period 2π is given below:

```
t=linspace(-pi,pi,65);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
n=arange(64)
wnd=fftshift(0.54+0.46*cos(2*pi*n/63))
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)
```

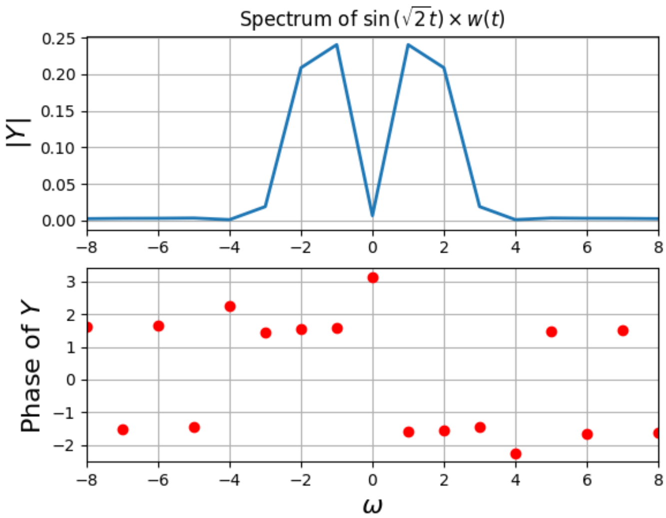


Figure 6: Spectrum of $\sin(\sqrt{2}t) * w(t)$

The spectrum that is obtained with a time period 8π has a sharper peak and is given below:

```
t=linspace(-4*pi,4*pi,257);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
n=arange(256)
wnd=fftshift(0.54+0.46*cos(2*pi*n/256))
y=sin(sqrt(2)*t)
# y=sin(1.25*t)
y=y*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))/256.0
w=linspace(-pi*fmax,pi*fmax,257);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),'b',w,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(w,angle(Y),'ro',lw=2)
xlim([-4,4])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
```

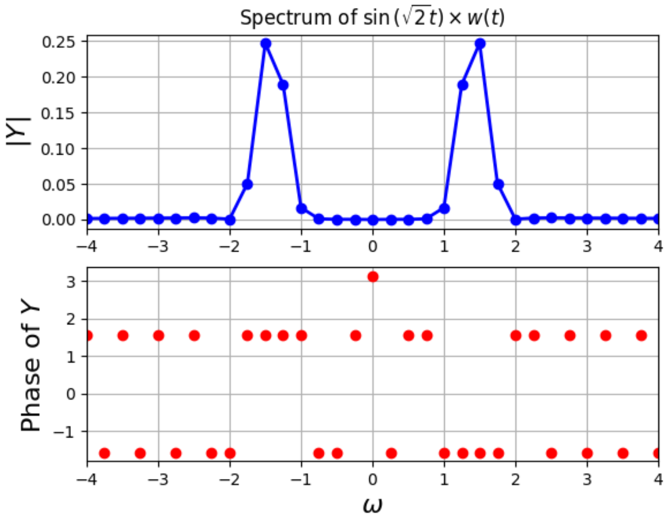


Figure 7: Spectrum of $\sin(\sqrt{2}t) * w(t)$

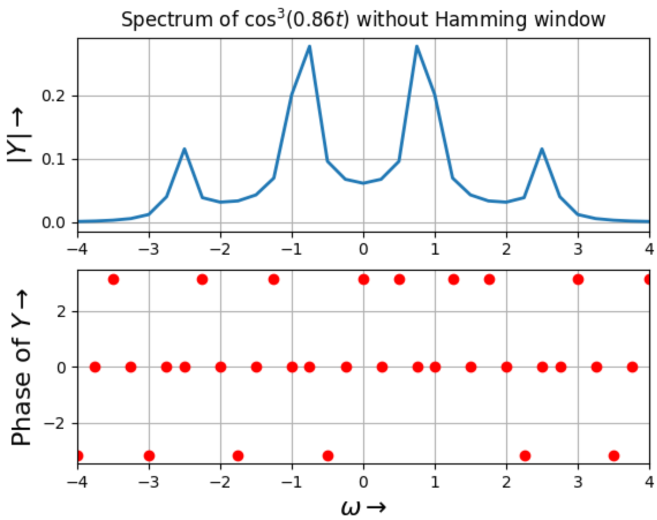
3 Questions

3.1 Question 2

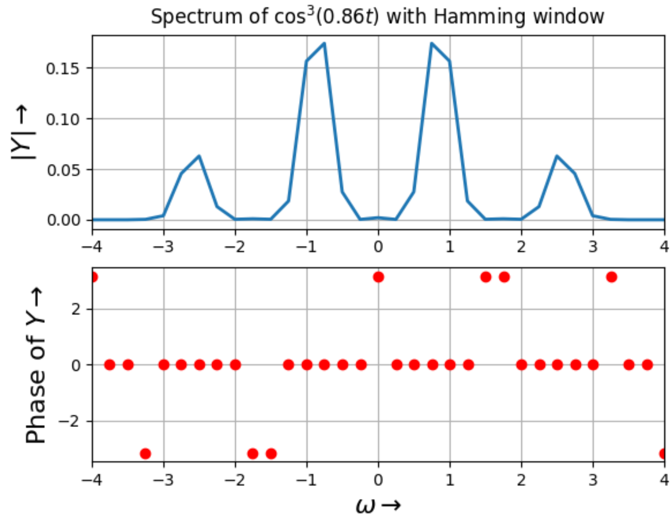
In this question, we shall plot the FFT of $cos^3(0.86t)$ The FFT without the hamming Window:

```
y = cos(0.86*t)**3
yw = y*wnd
y[0]=0
yw[0]=0
y = fftshift(y)
yw = fftshift(yw)
Y = fftshift(fft(y))/256.0                #without hamming window
Yw = fftshift(fft(yw))/256.0              #with hamming window

plot_spectrum(2,w,Y,4,r"Spectrum of $\cos^{3}(0.86t)$ without Hamming win
plot_spectrum(3,w,Yw,4,r"Spectrum of $\cos^{3}(0.86t)$ with Hamming windo
```


Spectrum of $\cos^3(0.86t)$ without Hamming window
$|Y| \rightarrow$
0.0
0.1
0.2
−4
−3
−2
−1
0
1
2
3
4
Phase of $Y \rightarrow$
−2
0
2
$\omega \rightarrow$

The FFT with the hamming Window:



We notice that a lot of the energy is stored in frequencies that aren't a part of the signal. After windowing, these frequencies are attenuated and hence the peaks are sharper in the windowed function.

3.2 Question 3

We need to estimate ω and δ for a signal $\cos(\omega t + \delta)$ for 128 samples between $[-\pi, \pi)$. We estimate omega using a weighted average. We have to extract the digital spectrum of the signal and find the two peaks at $\pm\omega_0$, and estimate ω and δ.

```
w0 = 0.8
d = 0.5
t = linspace(-pi,pi,129)[:-1]
dt = t[1]-t[0]; fmax = 1/dt
n = arange(128)
wnd = fftshift(0.54+0.46*cos(2*pi*n/128))
y = cos(w0*t + d)*wnd
y[0]=0
y = fftshift(y)
Y = fftshift(fft(y))/128.0
w = linspace(-pi*fmax,pi*fmax,129); w = w[:-1]
plot_spectrum(4,w,Y,4,r"Digital Spectrum of $\cos(w_0t+\delta)$",r"$|Y|\r

ii = where(w>=0)
cal_w = sum(abs(Y[ii])**2*w[ii])/sum(abs(Y[ii])**2)
i = abs(w-cal_w).argmin()
delta = angle(Y[i])
print("Value of w0 without noise: ",cal_w)
print("Value of delta without noise: ",delta)
```

We estimate omega by performing a Mean average of ω over the magnitude of $|Y(j\omega)|$. For delta we consider a window on each half of ω (split into positive and negative values) and extract their mean slope.

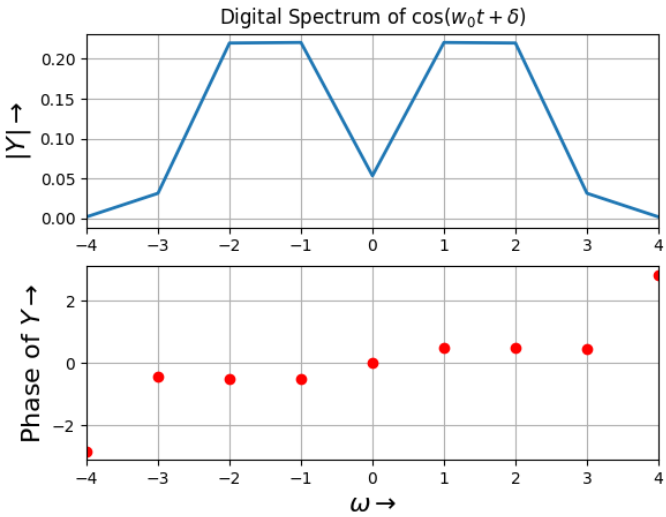


Figure 8: Fourier transform of $cos(1.5t + 0.5)$

3.3 Question 4

We repeat the exact same process as question 3 but with noise added to the original signal.

```
y = (cos(w0*t + d) + 0.1*randn(128))*wnd
y[0]=0
y = fftshift(y)
Y = fftshift(fft(y))/128.0
plot_spectrum(5,w,Y,4,r"Spectrum of a noisy $\cos(w_0t+\delta)$ with Hamm

# w0 is calculated by finding the weighted average of all w>0. Delta is f
ii = where(w>=0)
w_cal = sum(abs(Y[ii])**2*w[ii])/sum(abs(Y[ii])**2)
i = abs(w-w_cal).argmin()
delta = angle(Y[i])
print("Calculated value of w0 with noise: ",w_cal)
print("Calculated value of delta with noise: ",delta)
```

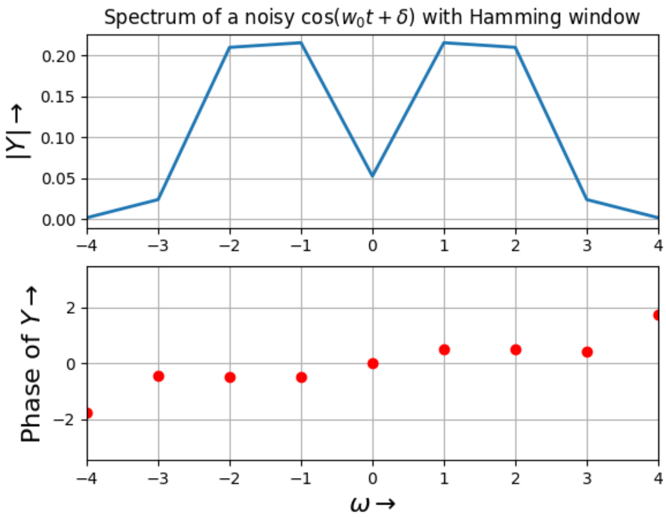


Figure 9: Fourier transform of noise + $cos(1.5t + 0.5)$

3.4 Question 5

In this question we analyze a chirp signal which is an FM signal where frequency is directly proportional to time. A chirp signal we shall consider is given by

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

The FFT of the chirp is given by: We note that the frequency response is spread between 5-50 rad/s. A large section of this range apears due to Gibbs phenomenon. On windowing, only frequencies between 16 and 32 rad/s remain.

```
t = linspace(-pi,pi,1025); t = t[:-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*t*(1.5 + t/(2*pi)))*wnd
y[0]=0
y = fftshift(y)
Y = fftshift(fft(y))/1024.0
w = linspace(-pi*fmax,pi*fmax,1025); w = w[:-1]
plot_spectrum(6,w,Y,100,r"Spectrum of chirped function",r"$|Y|\rightarrow
```

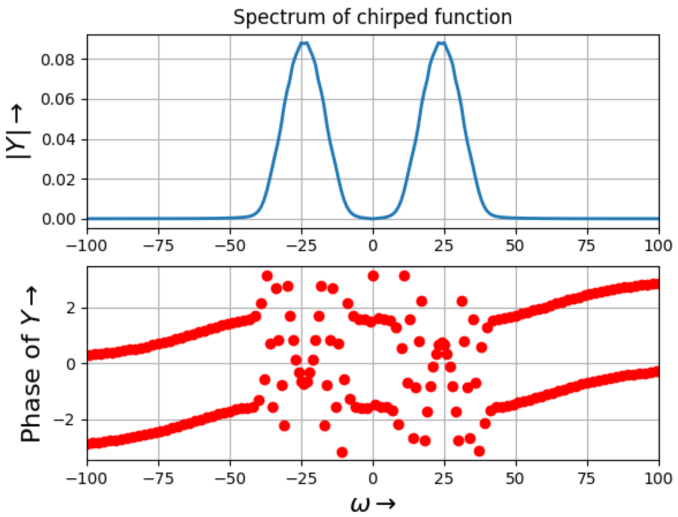


Figure 10: Chirp function fourier transform, windowed

3.5 Question 6

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

```
t_array = split(t,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)

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

fig1 = figure(7)
```

```
ax = fig1.add_subplot(111, projection='3d')
surf=ax.plot_surface(w,t,Y_mag.T,cmap='viridis',linewidth=0, antialiased=
fig1.colorbar(surf, shrink=0.5, aspect=5)
ax.set_title('surface plot');
ylabel(r"$\omega\rightarrow$")
xlabel(r"$t\rightarrow$")
```

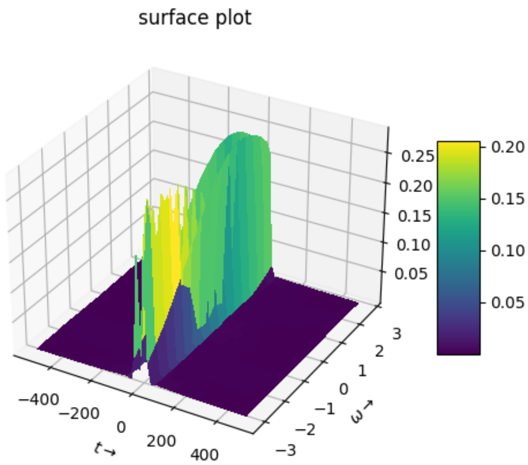


Figure 11: Chopped Chirp function