EE2703 : Applied Programming Lab
Assignment 9

Dharahas H
EE20B041

April 21, 2022

Assignment

Computing DFT's for non-periodic signals. Fourier analysis on signals without making assumptions that signal repeats itself after a period N.

Q1

Fourier analysis on signal :

$$f(t) = sin(\sqrt{2}t)$$

Plotting signal in 3 different ranges without windowing :

```
N = 512
t = np.linspace(-np.pi,np.pi,N+1);t = t[:-1]
t1 = np.linspace(-3*np.pi,-np.pi,N+1);t1 = t1[:-1]
t2 = np.linspace(np.pi,3*np.pi,N+1);t2 = t2[:-1]
dt = t[1] - t[0];fmax = 1/dt
y = np.sin(np.sqrt(2)*t)

plt.figure()
plt.grid()
plt.plot(t,y,'r' )
plt.plot(t1,y,'b')
plt.plot(t2,y,'b' )
plt.xlabel(r"$t\rightarrow$")
plt.ylabel(r"$y\rightarrow$")
plt.title(r"$sin(\sqrt{2}t)$")
plt.show()
```

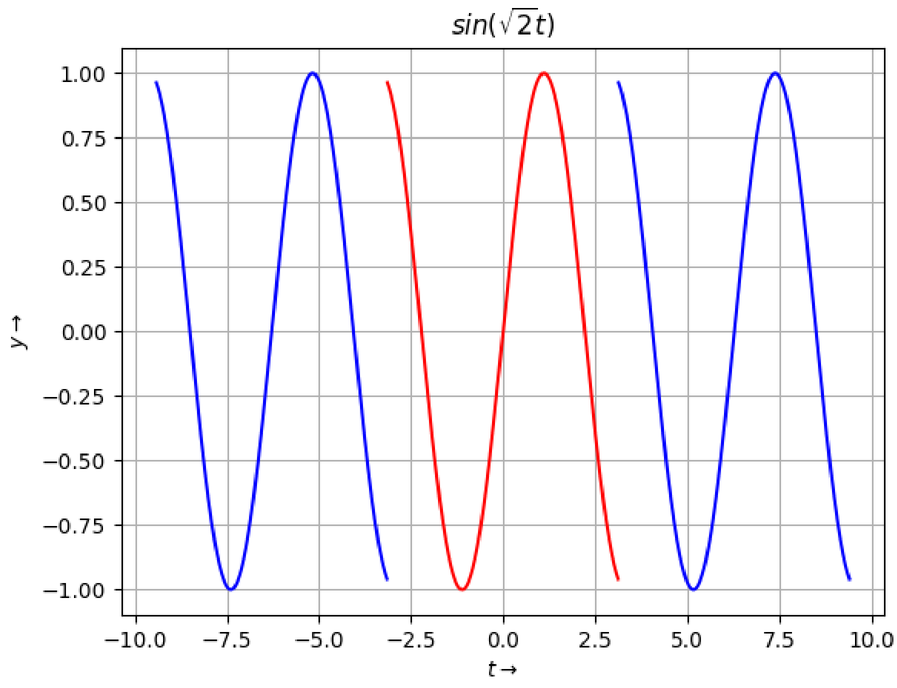

Figure 1: Signal without windowing

Now windowing the signal as follows,

$$f(t) = sin(\sqrt{2}t)w(t)$$

where,

$$w(t) = 0.54 + 0.46cos(2\pi n/(N-1))$$

code :

```
n = np.arange(N)
wnd = np.fft.fftshift(0.54+0.46*np.cos(2*np.pi*n/(N-1)))
y = y*wnd

plt.figure()
plt.grid()
plt.title(r'$sin(\sqrt{2}t)w(t)$')
plt.ylabel(r'$y\rightarrow$')
plt.xlabel(r'$t\rightarrow$')
plt.plot(t,y,'r')
plt.plot(t1,y,'b')
plt.plot(t2,y,'b')
plt.show()
```

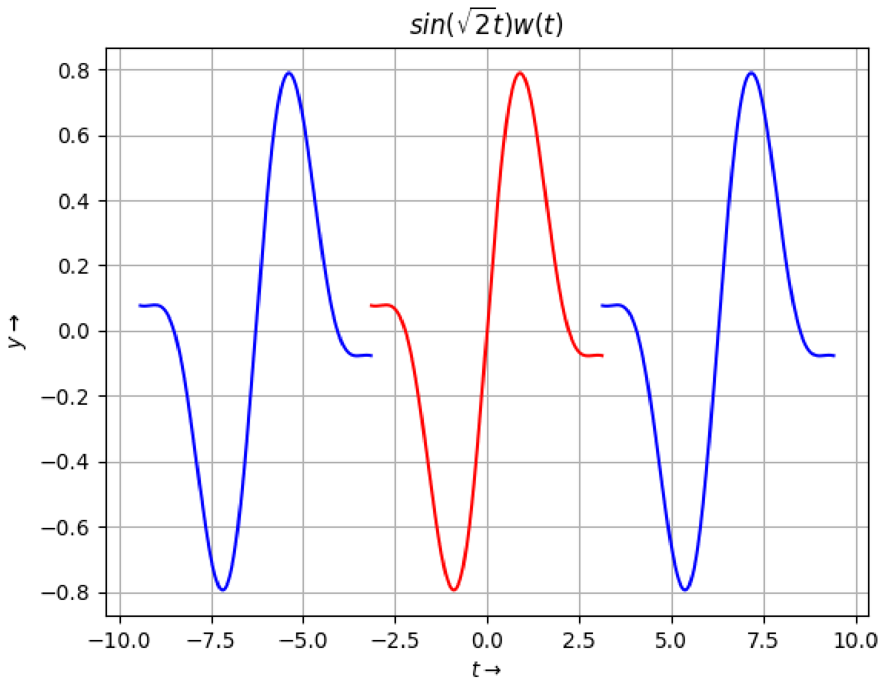

Figure 2: Signal with windowing

For the spectrum of the signal :

```
y[0]=0
y=np.fft.fftshift(y)
Y=np.fft.fftshift(np.fft.fft(y))/N
w=np.linspace(-np.pi*fmax,np.pi*fmax,N+1);w=w[:-1]

plt.figure()
plt.subplot(2,1,1)
plt.plot(w,abs(Y),'b', lw =2)
plt.plot(w,abs(Y),'bo', markersize =3)
plt.xlim([-4,4])
plt.ylabel(r"$|Y|\rightarrow$")
plt.grid(True)
plt.subplot(2,1,2)
ii = np.where(abs(Y) > 1e-2)
plt.plot(w[ii], np.angle(Y[ii]), "ro", markersize =4)
plt.plot(w,np.angle(Y),'o', markersize=3)
plt.xlim([-4,4])
plt.grid()
plt.ylabel(r"$\angle Y\rightarrow$")
plt.xlabel(r"$\omega\rightarrow$")
plt.show()
```

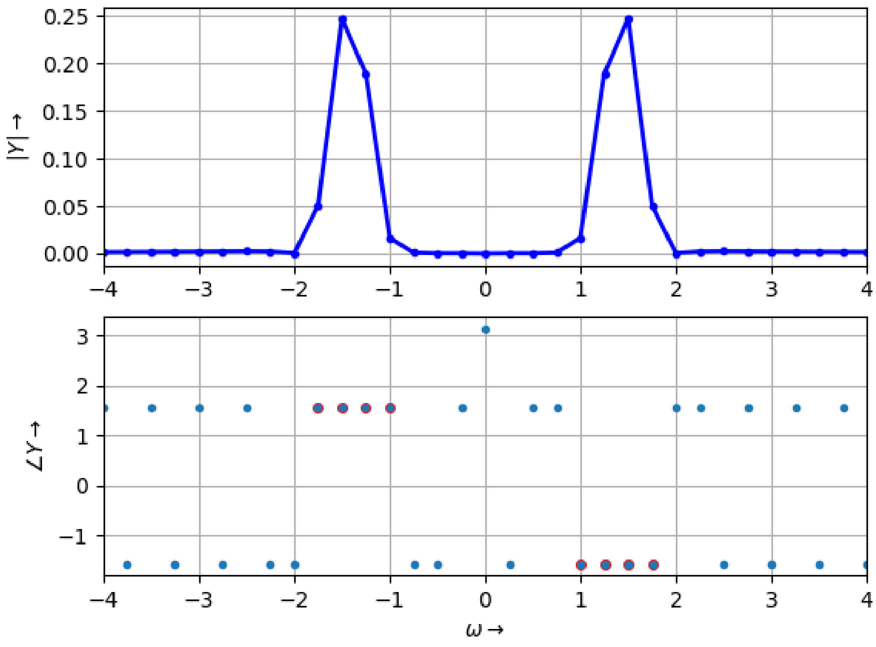

Figure 3: Spectrum of signal after windowing

Q2

Spectrum for $cos(0.86t)$ with and without hamming window.
plotting in different ranges.

```
N = 512
t = np.linspace(-np.pi, np.pi, N+1); t = t[0:-1]
t1 = np.linspace(-3*np.pi, -np.pi, N+1); t1= t1[0:-1]
```

```
t2 = np.linspace(np.pi, 3*np.pi, N+1); t2= t2[0:-1]
dt = t[1] - t[0]
fmax = 1/dt
y = (np.cos(0.86*t))**3

plt.figure()
plt.grid()
plt.plot(t,y,'r' )
plt.plot(t1,y,'b')
plt.plot(t2,y,'b' )
plt.xlabel(r"$t\rightarrow$")
plt.ylabel(r"$y\rightarrow$")
plt.title(r"$cos^3(0.86t)$")
plt.show()
```

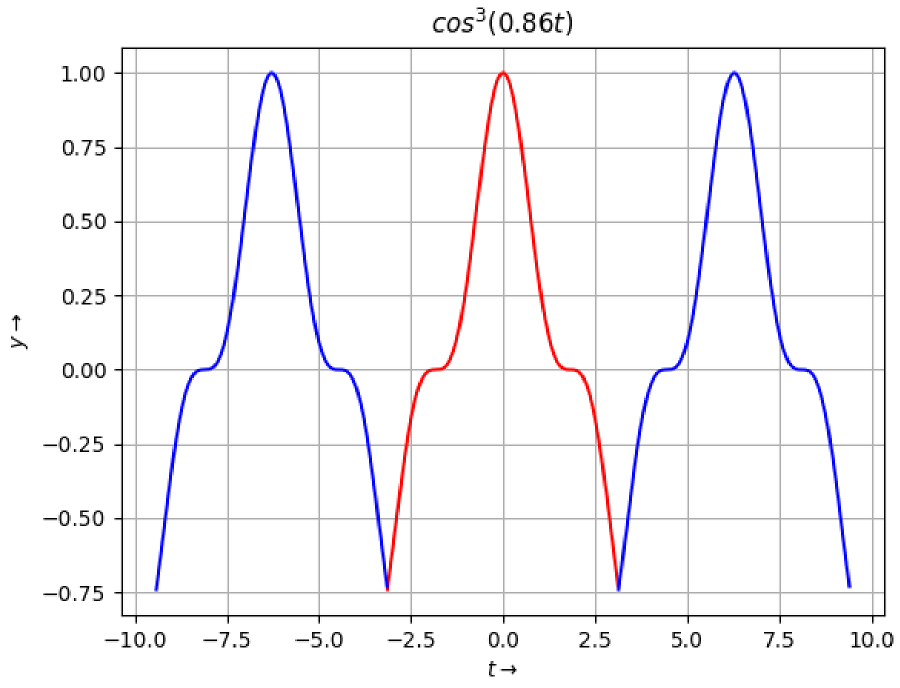

Figure 4: Signal in 3 different ranges

Spectrum of the signal :

```
N =512
```

```
t = np.linspace(-32*np.pi,32*np.pi,N+1);t = t[:-1]
dt = t[1] - t[0]
fmax = 1/dt
y = (np.cos(0.86*t))**3
y[0] = 0
Y = np.fft.fftshift(y)
Y = np.fft.fftshift(np.fft.fft(y))/N
w = np.linspace(-np.pi*fmax,np.pi*fmax,N+1);w = w[:-1]

plt.plot(w, abs(Y), 'b')
plt.xlim([-4,4])
plt.xlabel(r"$\omega\rightarrow$")
plt.ylabel(r"$|Y|\rightarrow$")
plt.title("Spectrum without Hamming window")
plt.grid()
```

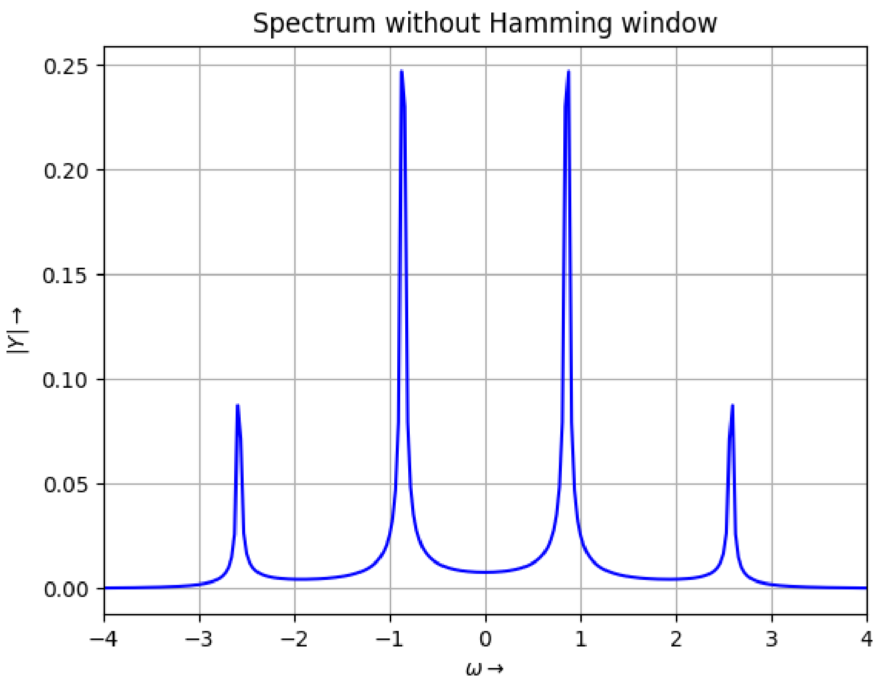

Figure 5: Spectrum without hamming window

Windowing and plotting the spectrum :

```
n = np.arange(N)
```

```
wnd = np.fft.fftshift(0.54+0.46*np.cos(2*np.pi*n/(N-1)))
yw[0] = 0
yw = y*wnd

Yw = np.fft.fftshift(yw)
Yw = np.fft.fftshift(np.fft.fft(Yw))/N

plt.plot(w, abs(Yw))
plt.xlim(-4,4)
plt.xlabel(r"$\omega\rightarrow$")
plt.ylabel(r"$|Y|\rightarrow$")
plt.title("Spectrum with hamming window")
plt.grid()
plt.show()
```

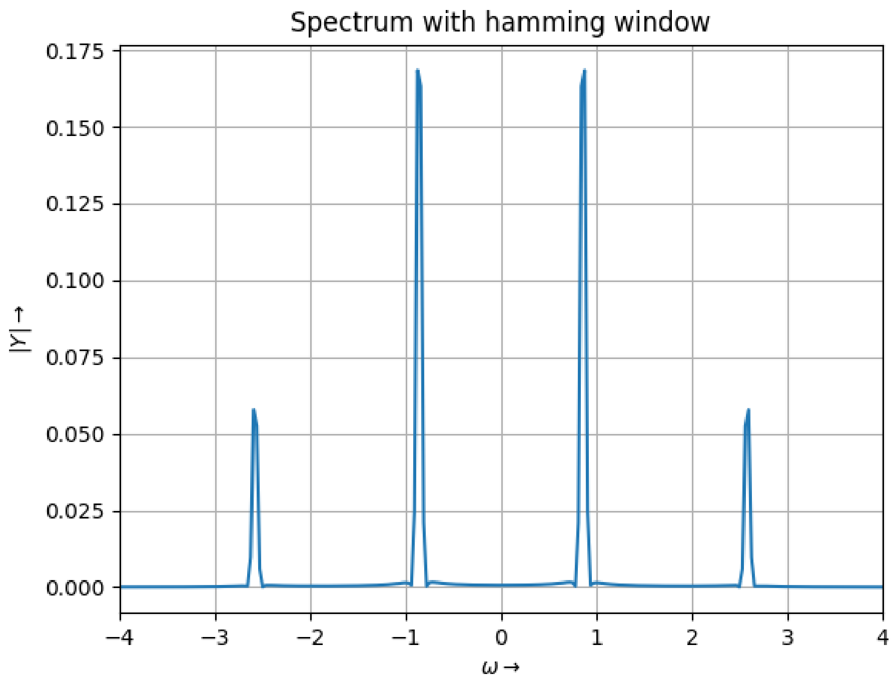

Figure 6: Spectrum with hamming window

Q3

Initialising random omega and phi:

```
omega = 0.5 + np.random.random()
phi = 2*np.pi*np.random.random()
print("omega :"+ str(omega))
print("phi :"+ str(phi))
```

Obtained values are :

$$\omega = 0.8316$$
$$\delta = 4.8778$$

Windowing and plotting the signal :

```
N = 128
t = np.linspace(-np.pi,np.pi,N+1);t = t[:-1]
dt = t[1] - t[0];fmax = 1/dt
y = np.cos(omega*t + phi)
#windowing the signal
n = np.arange(N)
wnd = np.fft.fftshift(0.54+0.46*np.cos(2*np.pi*n/(N-1)))
yw = y*wnd
#plotting the windowed signal

t1 = np.linspace(-3*np.pi, -np.pi, N+1); t1=t1[:-1]
t2 = np.linspace(np.pi, 3*np.pi, N+1); t2 = t2[:-1]
plt.figure()
plt.plot(t, yw, "r", lw= 1)
plt.plot(t1, yw, 'b', lw= 1)
plt.plot(t2, yw, 'b', lw= 1)
plt.xlabel(r"$t\rightarrow$")
plt.ylabel(r"$y\rightarrow$")
plt.title(r"Signal after windowing")
plt.grid()
plt.show()
```

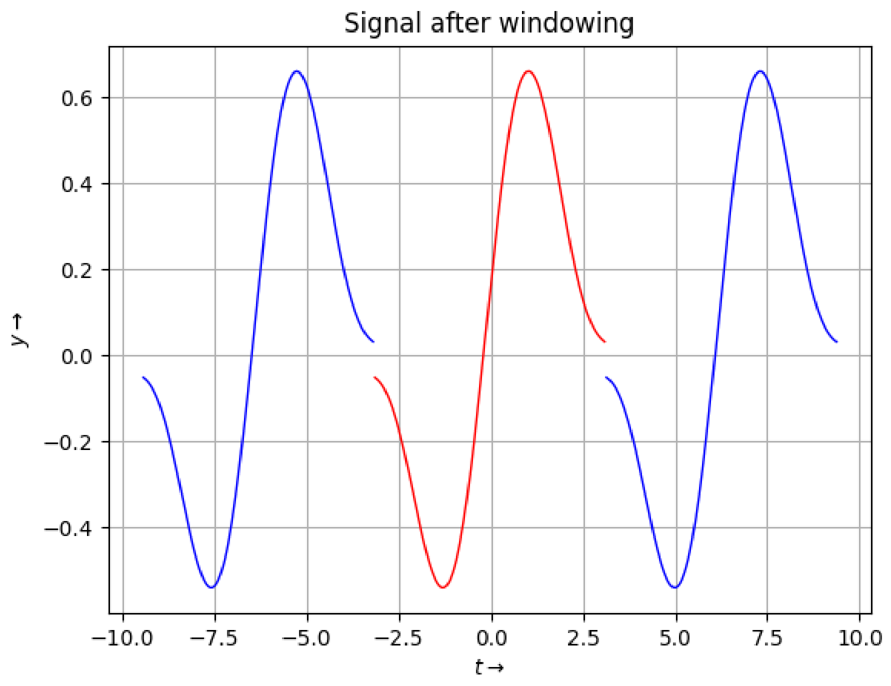

Figure 7: Signal with hamming window

Obtaining the Spectrum of the signal :

```
#DFT of the windowed signal
yw[0] = 0
Yw = np.fft.fftshift(yw)
Yw = np.fft.fftshift(np.fft.fft(Yw))/N
w = np.linspace(-np.pi*fmax, np.pi*fmax, N+1); w = w[:-1]

plt.figure()
plt.subplot(2,1,1)
plt.plot(w, abs(Yw))
plt.stem(w, abs(Yw))
plt.xlim([-10,10])
plt.xlabel(r"$\omega\rightarrow$")
plt.ylabel(r"$|Y|\rightarrow$")
plt.grid()

plt.subplot(2,1,2)
plt.plot(w, np.angle(Yw), "*", ms = 3)
```

```
ii = np.where(abs(Yw) > 1e-2)
plt.plot(w[ii], np.angle(Yw[ii]),'ro', ms= 4)
plt.xlim([-10,10])
plt.ylabel(r"$\angle Y\rightarrow$")
plt.xlabel(r"$\omega\rightarrow$")
plt.grid()
plt.show()
```

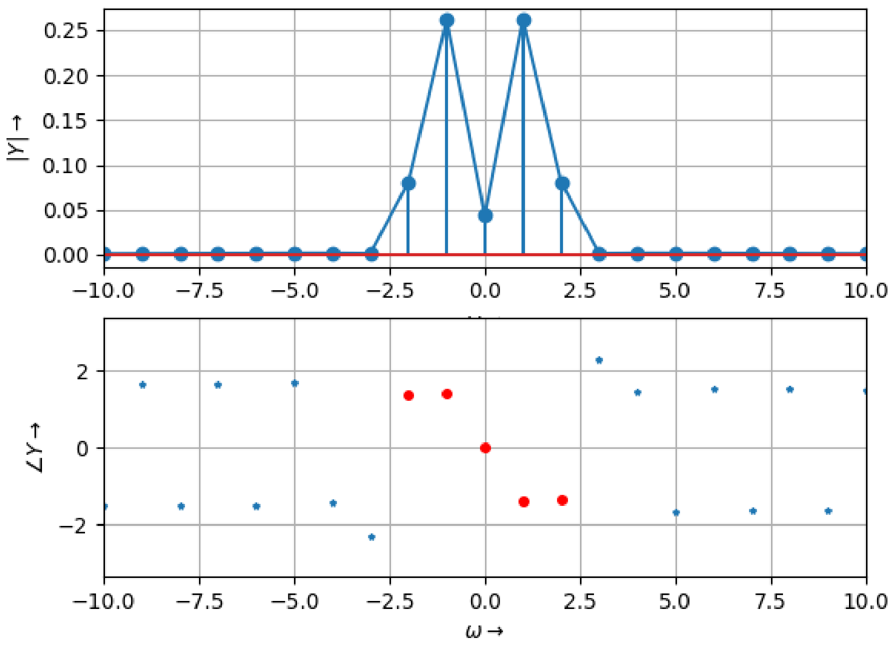

Figure 8: Spectrum of the windowed signal

Finding ω and δ :

we use weighted sum formula :

$$\omega = \frac{\sum_{k=-N/2}^{k=+N/2} |Y(k)|^{2.1} w_k}{\sum_{k=-N/2}^{k=+N/2} |Y(k)|^{2.1}}$$

On Subtracting the phase of $cos(t)$ and phase of $cos(\omega t + \delta)$ at $\omega = 1$ we will get δ. we get,

$$\omega = 0.8145$$
$$\delta = 4.8763$$

Q4

Signal is :

$$y(t) = cos(\omega t + \delta) + noise$$

Initialising ω, δ and noise :

```
N=128
omega = 0.5 + np.random.random()
phi = 2*np.pi*np.random.random()
noise = 0.1*np.random.randn(N)
print("omega :"+ str(omega))
print("phi :"+ str(phi))
```

Obtained values are :

$$\omega = 0.9951$$
$$\delta = 5.3948$$

Plotting the windowed signal:

```
t = np.linspace(-np.pi,np.pi,N+1);t = t[:-1]
dt = t[1] - t[0];fmax = 1/dt
y = np.cos(omega*t + phi) + noise
#windowing the signal
n = np.arange(N)
wnd = np.fft.fftshift(0.54+0.46*np.cos(2*np.pi*n/(N-1)))
yw = y*wnd
#plotting the windowed signal

t1 = np.linspace(-3*np.pi, -np.pi, N+1); t1=t1[:-1]
t2 = np.linspace(np.pi, 3*np.pi, N+1); t2 = t2[:-1]
plt.figure()
plt.plot(t, yw, "r", lw= 1)
plt.plot(t1, yw, 'b', lw= 1)
plt.plot(t2, yw, 'b', lw= 1)
plt.xlabel(r"$t\rightarrow$")
plt.ylabel(r"$y\rightarrow$")
plt.title(r"Signal after windowing")
plt.grid()
plt.show()
```

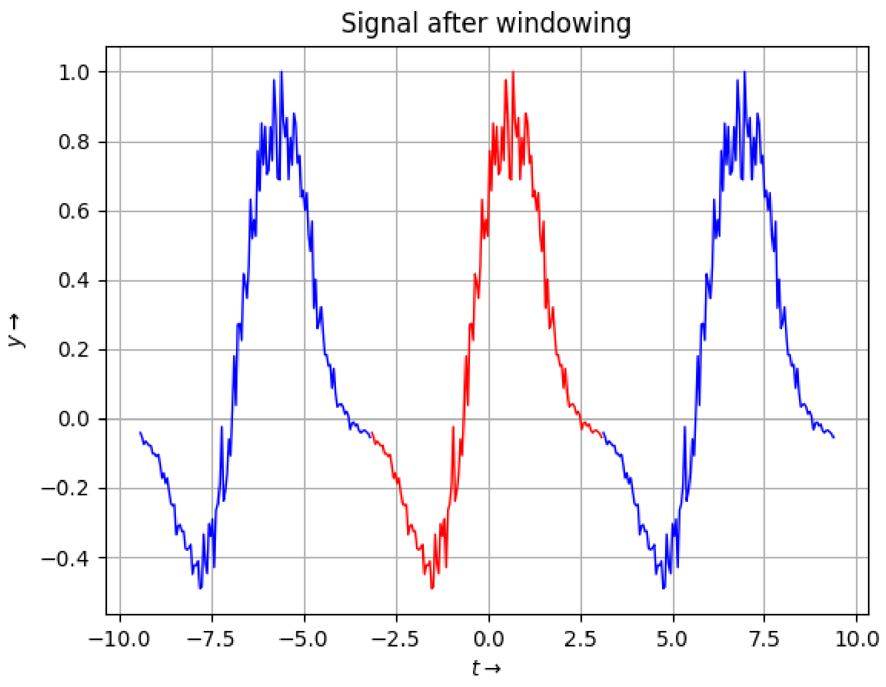

Figure 9: Windowed signal

Spectrum of the signal with hamming window :

```
#DFT of the windowed signal
 yw[0] = 0
 Yw = np.fft.fftshift(yw)
 Yw = np.fft.fftshift(np.fft.fft(Yw))/N
 w = np.linspace(-np.pi*fmax, np.pi*fmax, N+1); w = w[:-1]

 plt.figure()
 plt.subplot(2,1,1)
 plt.plot(w, abs(Yw))
 plt.stem(w, abs(Yw))
 plt.xlim([-10,10])
 plt.xlabel(r"$\omega\rightarrow$")
 plt.ylabel(r"$|Y|\rightarrow$")
 plt.title("DFT response")
 plt.grid()

 plt.subplot(2,1,2)
 plt.plot(w, np.angle(Yw), "*", ms = 3)
 ii = np.where(abs(Yw) > 1e-2)
 plt.plot(w[ii], np.angle(Yw[ii]),'ro', ms= 4)
 plt.xlim([-10,10])
 plt.ylabel(r"$\angle Y\rightarrow$")
 plt.xlabel(r"$\omega\rightarrow$")
 plt.grid()
 plt.show()
```

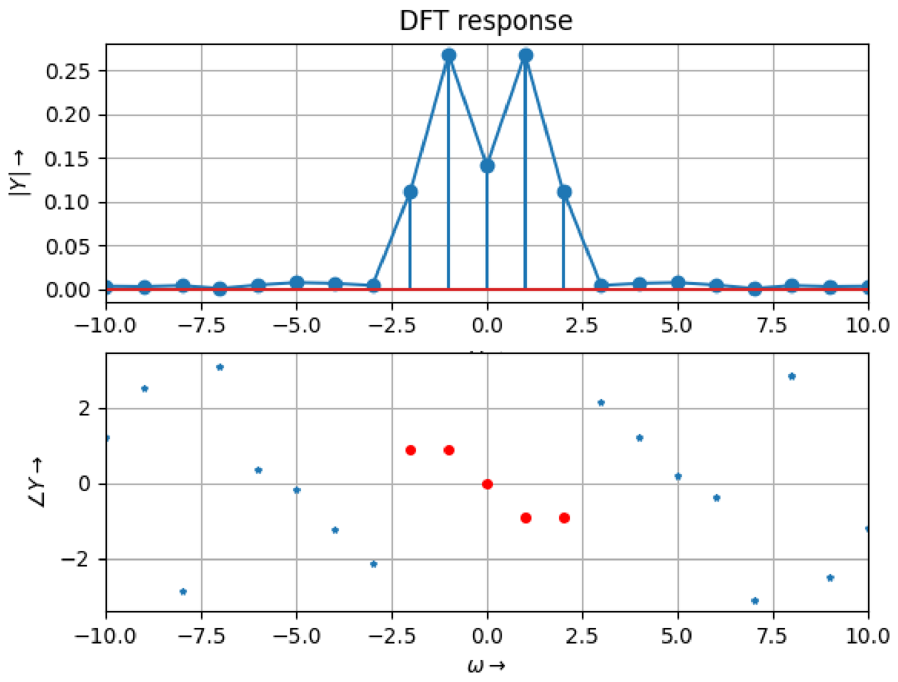

Figure 10: Spectrum of the signal

Finding ω and δ :

we use weighted sum formula :

$$\omega = \frac{\sum_{k=-N/2}^{k=+N/2} |Y(k)|^{2.6} w_k}{\sum_{k=-N/2}^{k=+N/2} |Y(k)|^{2.6}}$$

On Subtracting the phase of $cos(t)$ and phase of $cos(\omega t + \delta)$ at $\omega = 1$ we will get δ. we get,

$$\omega = 1.0327$$
$$\delta = 5.1657$$

Q5

Signal is given as :

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

Signal without hamming window:

```
N = 1024
t = np.linspace(-np.pi, np.pi, N +1); t = t[:-1]
y = (np.cos(16*(1.5+t/(2*np.pi))*t))
dt = t[1] - t[0]
fmax = 1/dt
#plotting th signal in 3 diff range
t1 = np.linspace(-3*np.pi, -np.pi, N+1); t1 = t1[:-1]
t2 = np.linspace(np.pi, 3*np.pi, N+1); t2 = t2[:-1]

plt.figure()
plt.plot(t, y, "r", lw= 1)
plt.plot(t1, y, 'b', lw= 1)
plt.plot(t2, y, 'b', lw= 1)
plt.xlabel(r"$t\rightarrow$")
plt.ylabel(r"$y\rightarrow$")
plt.title(r"Signal without windowing")
plt.grid()
plt.show()
```

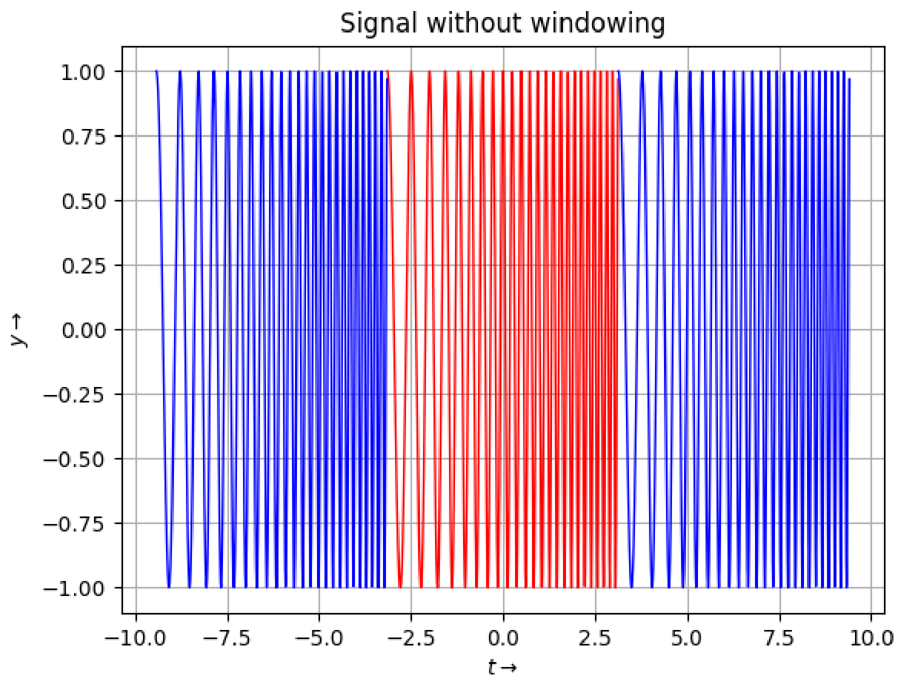

Figure 11: Signal in different ranges

Signal with hamming window :

```
#windowing the signal
n = np.arange(N)
wnd = np.fft.fftshift(0.54+0.46*np.cos(2*np.pi*n/(N-1)))
yw = y*wnd
#plotting windowed signal
plt.figure()
plt.plot(t, yw, "r", lw= 1)
plt.plot(t1, yw, 'b', lw= 1)
plt.plot(t2, yw, 'b', lw= 1)
plt.xlabel(r"$t\rightarrow$")
plt.ylabel(r"$y\rightarrow$")
plt.title(r"Signal after windowing")
plt.grid()
plt.show()
```

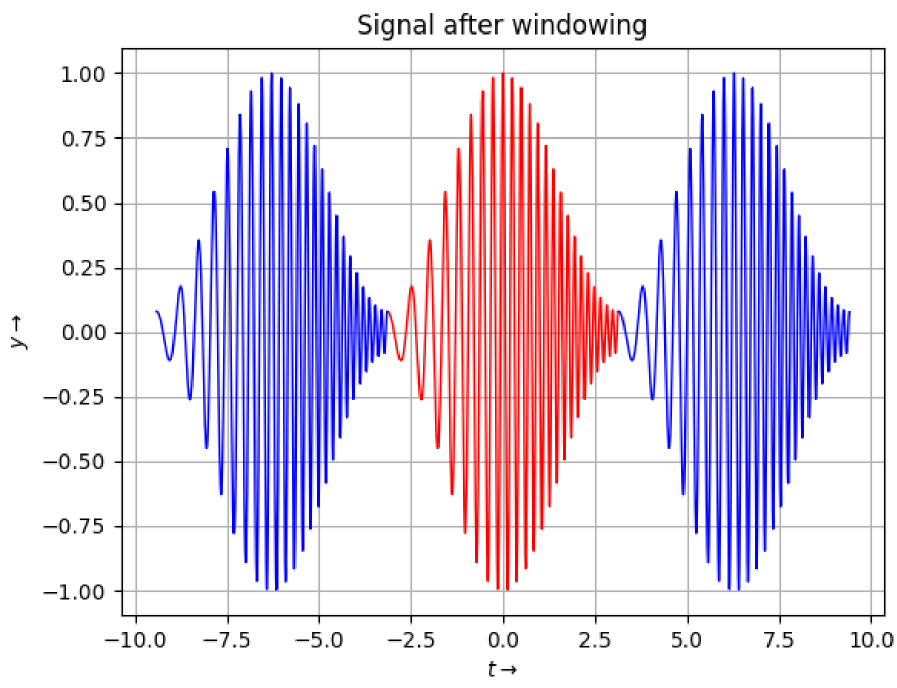

Figure 12: Signal after windowing

Spectrum of the signal :

```
#DFT of the windowed signal
yw[0] = 0
Yw = np.fft.fftshift(yw)
Yw = np.fft.fftshift(np.fft.fft(Yw))
w = np.linspace(-np.pi*fmax, np.pi*fmax, N+1); w = w[:-1]
# plotting the DFT of windowed signal

plt.figure()
plt.subplot(2,1,1)
plt.plot(w, abs(Yw))
plt.stem(w, abs(Yw), markerfmt = 'none')
plt.xlim([-50,50])
plt.xlabel(r"$\omega\rightarrow$")
plt.ylabel(r"$|Y|\rightarrow$")
plt.title("DFT response")
plt.grid()
plt.subplot(2,1,2)
plt.plot(w, np.angle(Yw), "+", ms = 1)
ii = np.where(abs(Yw) > 25)
plt.plot(w[ii], np.angle(Yw[ii]),'ro', ms= 2)
plt.xlim([-50,50])
plt.ylabel(r"$\angle Y\rightarrow$")
plt.xlabel(r"$\omega\rightarrow$")
plt.grid()
plt.show()
```

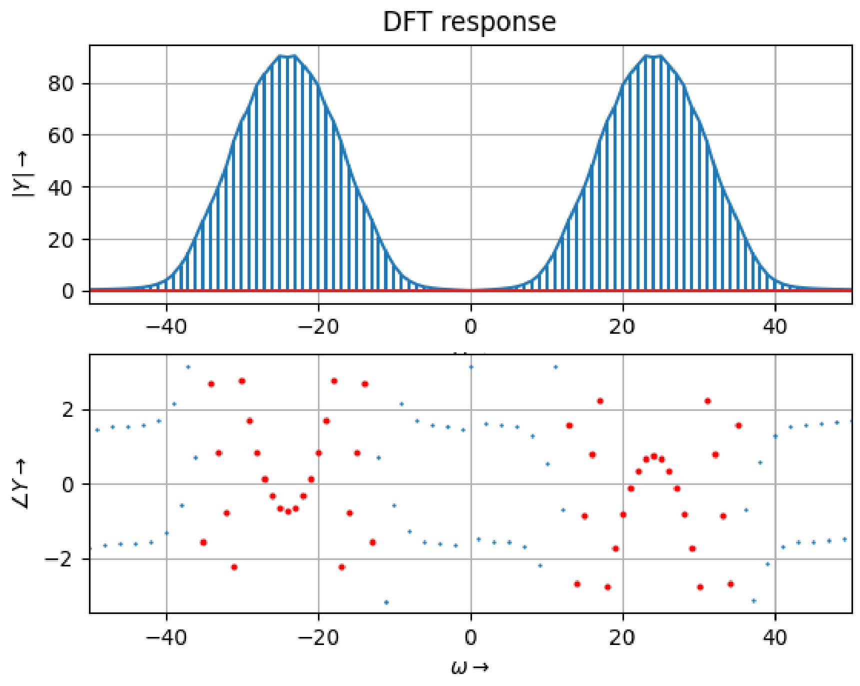

Figure 13: Signal after windowing

Q6

Plotting a surface-plot which gives inference about how magnitude response change as frequency changes with time

```
t = np.linspace(-np.pi, np.pi, 1024+1); t = t[:-1]
t = np.reshape(t,(64,-1))
Y = np.zeros_like(t,dtype=np.complex128)
```

```
for i in range(t.shape[1]):
    x = t[:,i]
    y = Y[:,i]
    y = np.cos(16*x*(1.5+x/(2*np.pi)))
    n = np.arange(64)
    wnd = np.fft.fftshift(0.54+0.46*np.cos(2*np.pi*n/(63)))
    y = y*wnd
    y[0] = 0
    y = np.fft.fftshift(y)
    Y[:,i] = np.fft.fftshift(np.fft.fft(y))/64.0

x = np.arange(t.shape[1])
y = np.arange(t.shape[0])
y,x = np.meshgrid(y,x)
Y = np.fft.fftshift(Y,axes=0)

fig = plt.figure(figsize=(16,8))
ax = plt.axes(projection='3d')
surf = ax.plot_surface(y,x,abs(Y.T), linewidth=0, cmap = plt.cm.jet,
       antialiased=False)
plt.title(r'Surface Plot of Time-Frequency Plot')
fig.colorbar(surf, shrink=0.5, aspect=5)
plt.show()
```

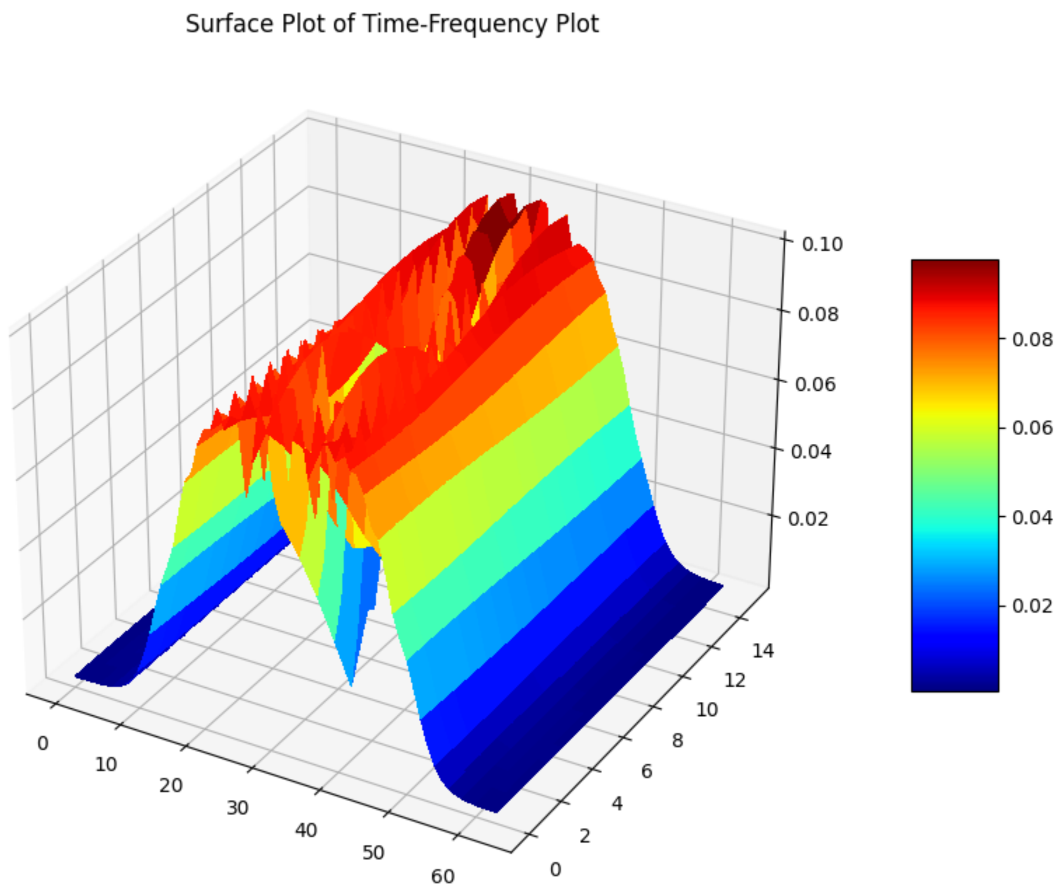

Figure 14: Surface plot : frequency - time

Conclusion

We found out the spectrum of non-periodic signals and analysed them. Used the concept of windowing to achieve periodicity.