EE2703 : Applied Programming Lab
Assignment - 8

Jawhar S
EE20B049

May 12, 2022

Introduction

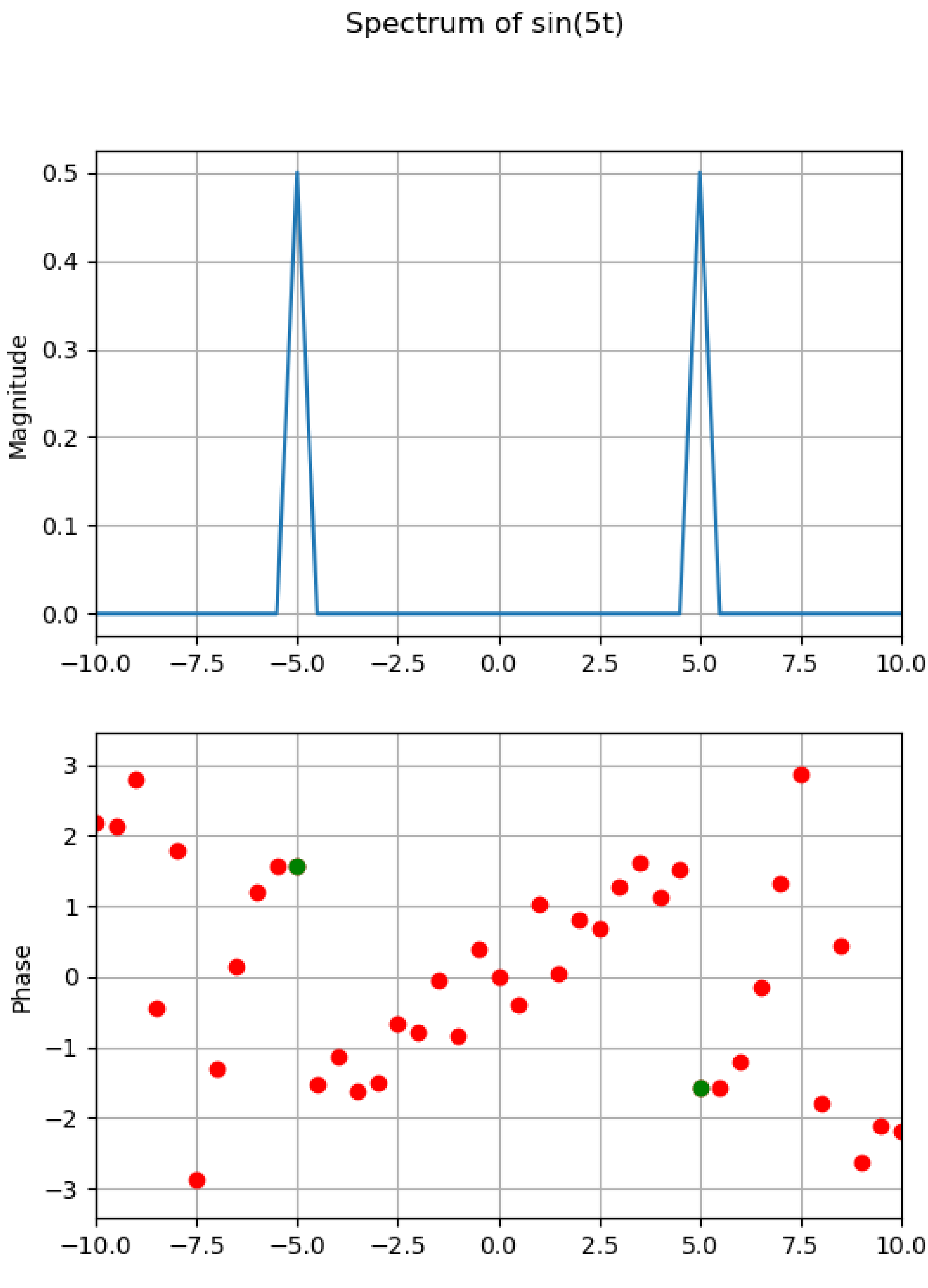
This assignment is about DFT and how it is implemented in python using Numpy's FFT module. We also attempt to approximate the continuous time fourier transform of a gaussian by windowing and sampling in time domain, and then taking the DFT.

Spectrum of $sin(5t)$

The phase for some values near the peaks is non zero. To fix this we sample the input signal at an appropriate frequency. We also shift the phase plot so that it goes from $-\pi$ to π.

```
x = np.linspace(-2*np.pi,2*np.pi,257)
x=x[:-1]
y = np.sin(5*x)
Y = fft.fftshift(fft.fft(y))/256
w = np.linspace(-64,64,257)
w=w[:-1]
fig,fig1 = plt.subplots(2,1,figsize=[6,8])
fig1[0].plot(w,abs(Y))
fig1[0].set_ylabel("Magnitude")
fig1[0].set_xlim([-10,10])
fig1[1].plot(w,np.angle(Y),'ro')
fig1[1].set_ylabel("Phase")
fig1[1].set_xlim([-10,10])
ii = np.where(abs(Y)>1e-3)
fig1[1].plot(w[ii],np.angle(Y[ii]),'go')
fig.suptitle("Spectrum of sin(5t)")
fig1[0].grid("true")
fig1[1].grid("true")
plt.show()
```

As expected we get 2 peaks at +5 and -5 with height 0.5


Spectrum of sin(5t)
Magnitude
Phase

Spectrum of $(1 + 0.1cos(t))cos(10t)$

Consider the signal:

$$f(t) = (1 + 0.1\cos(t))\cos(10t) \tag{1}$$

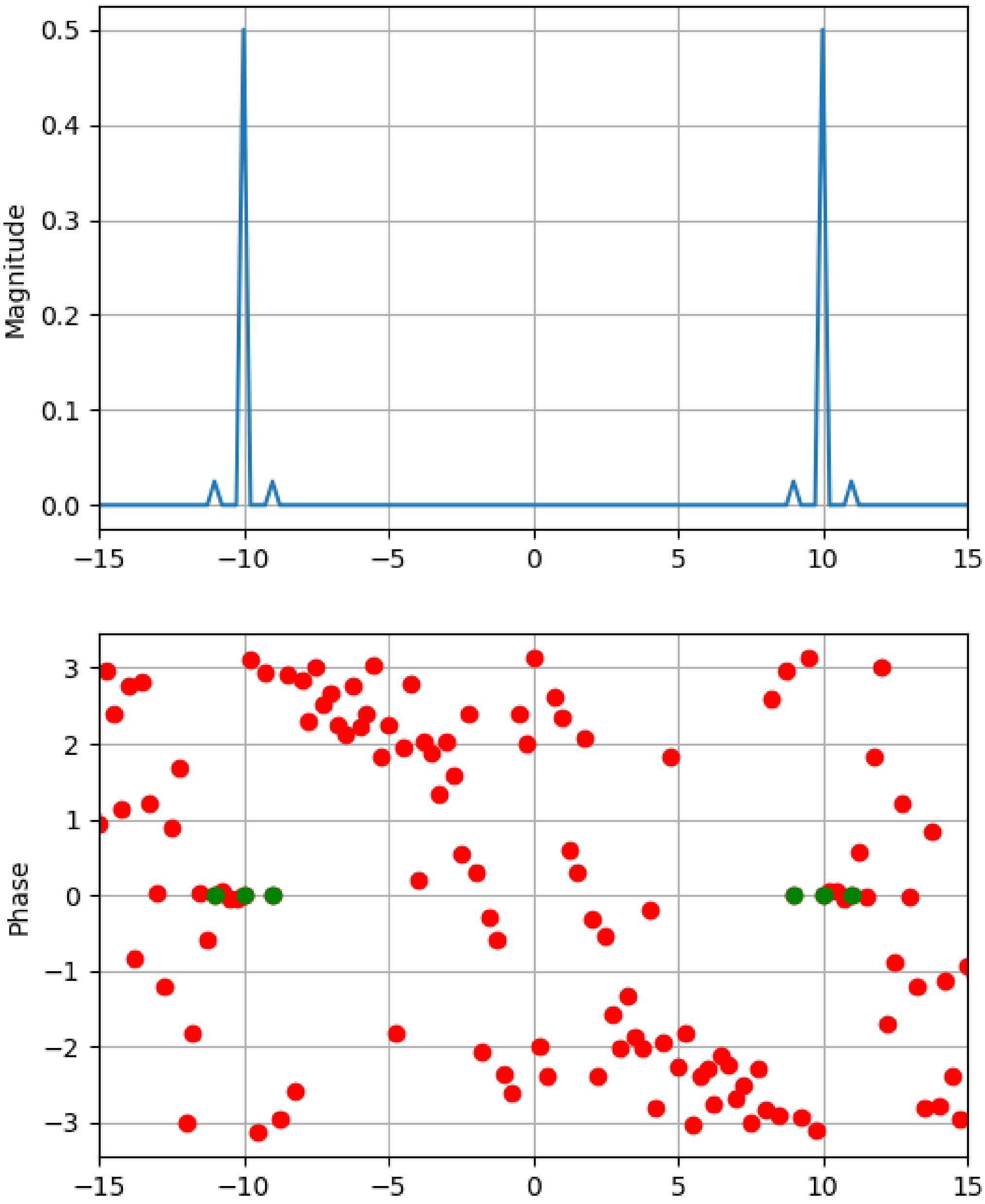
We expect a shifted set of spikes, with a main impulse and two side impulses on each side. This is because,

$$0.1cos(10t)cos(t) = 0.05(cos11t + cos9t) = 0.025(e11tj + e9tj + e11tj + e9tj) \tag{2}$$

The Python code to calculate the DFT for above function is as follows:

```
t = np.linspace(-4*np.pi,4*np.pi,513)
t = t[:-1]
y = (1 + 0.1*np.cos(t))*np.cos(10*t)
Y = fft.fftshift(fft.fft(y))/512
w = np.linspace(-64,64,513)
w = w[:-1]
fig,fig2 = plt.subplots(2,1,figsize=[6,8])
fig2[0].plot(w,abs(Y))
fig2[0].set_ylabel("Magnitude")
fig2[0].set_xlim([-15,15])
fig2[1].plot(w,np.angle(Y),'ro')
fig2[1].set_ylabel("Phase")
fig2[1].set_xlim([-15,15])
ii = np.where(abs(Y) > 1e-3)
fig2[1].plot(w[ii],np.angle(Y[ii]),'go')
fig.suptitle("Spectrum of (1+0.1cos(t))cos(10t)")
fig2[0].grid("true")
fig2[1].grid("true")
plt.show()
```

Spectrum of (1+0.1cos(t))cos(10t)



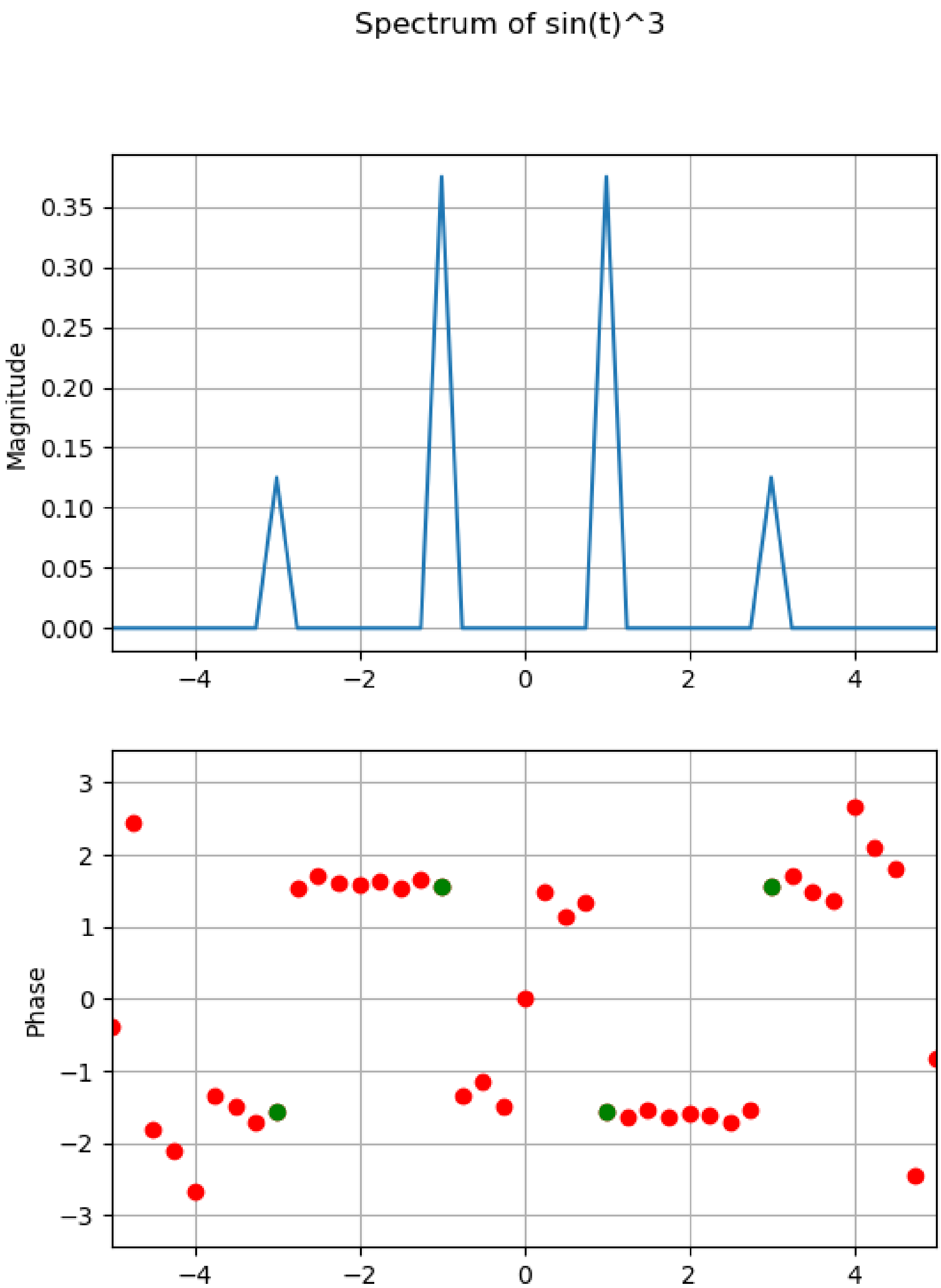
Spectrum of $sin^3(t)$

This signal can be expressed as a sum of sine waves using this identity:

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

The Python code for above function is as follows:

```
x = np.linspace(-4*np.pi,4*np.pi,513)
x=x[:-1]
y = pow(np.sin(x),3)
Y = fft.fftshift(fft.fft(y))/512
w = np.linspace(-64,64,513)
w=w[:-1]
fig,fig3 = plt.subplots(2,1,figsize=[6,8])
fig3[0].plot(w,abs(Y))
fig3[0].set_ylabel("Magnitude")
fig3[0].set_xlim([-5,5])
fig3[1].plot(w,np.angle(Y),'ro')
fig3[1].set_ylabel("Phase")
fig3[1].set_xlim([-5,5])
ii = np.where(abs(Y)>1e-3)
fig3[1].plot(w[ii],np.angle(Y[ii]),'go')
fig.suptitle("Spectrum of sin(t)^3")
fig3[0].grid("true")
fig3[1].grid("true")
plt.show()
```


Spectrum of sin(t)^3
Magnitude
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−2
0
2
4
Phase
3
2
1
0
−1
−2
−3
−4
−2
0
2
4

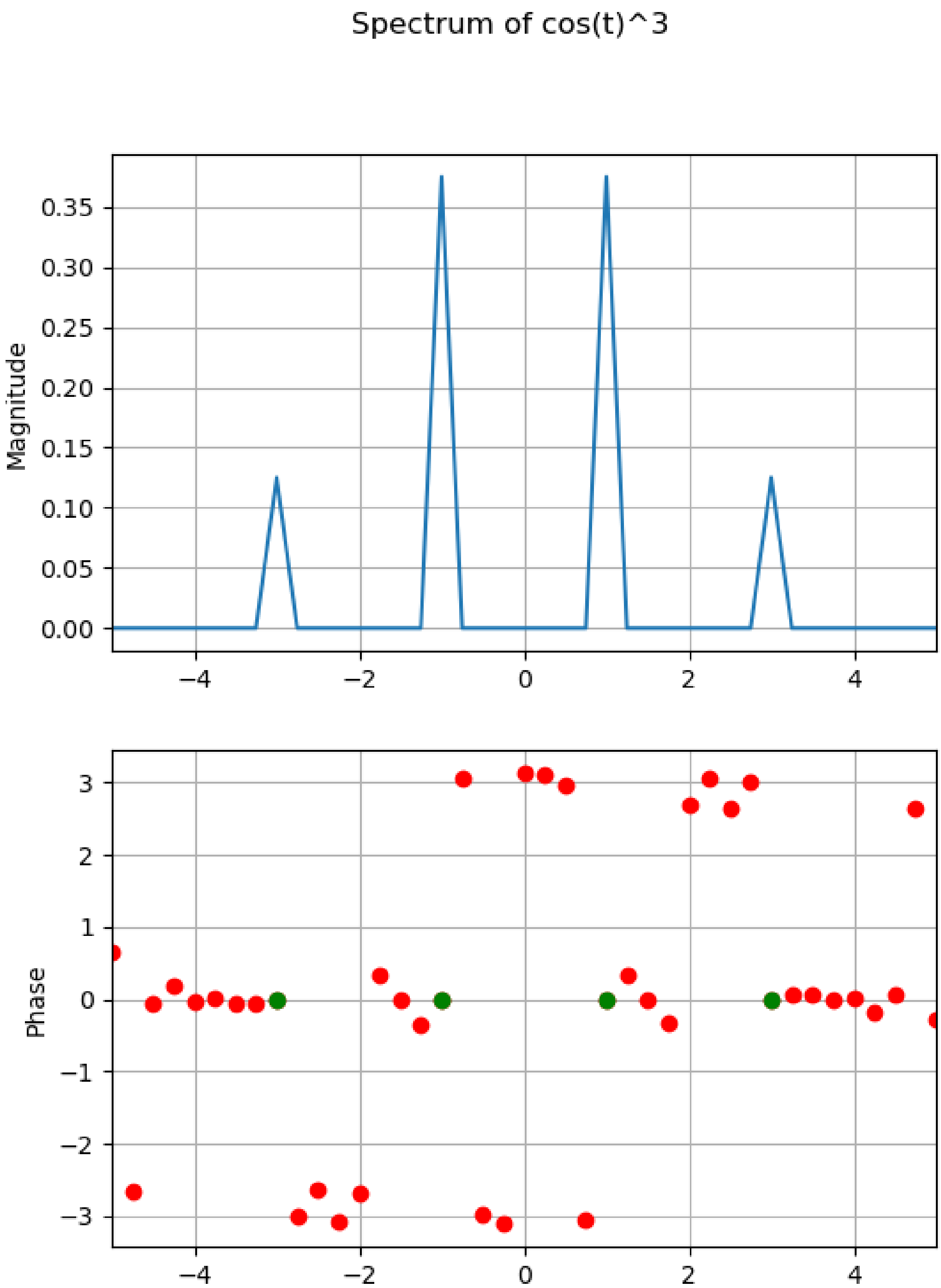
Spectrum of $cos^3(t)$

This signal can be expressed as a sum of cosine waves using this identity:

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

The Python code for above function is as follows:

```
t = np.linspace(-4*np.pi,4*np.pi,513)
t=t[:-1]
y = pow(np.cos(t),3)
Y = fft.fftshift(fft.fft(y))/512
w = np.linspace(-64,64,513)
w=w[:-1]
fig,fig4 = plt.subplots(2,1,figsize=[6,8])
fig4[0].plot(w,abs(Y))
fig4[0].set_ylabel("Magnitude")
fig4[0].set_xlim([-5,5])
fig4[1].plot(w,np.angle(Y),'ro')
fig4[1].set_ylabel("Phase")
fig4[1].set_xlim([-5,5])
ii = np.where(abs(Y)>1e-3)
fig4[1].plot(w[ii],np.angle(Y[ii]),'go')
fig.suptitle("Spectrum of cos(t)^3")
fig4[0].grid("true")
fig4[1].grid("true")
plt.show()
```



Spectrum of cos(t)^3
Magnitude
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−2
0
2
4
Phase
3
2
1
0
−1
−2
−3
−4
−2
0
2
4

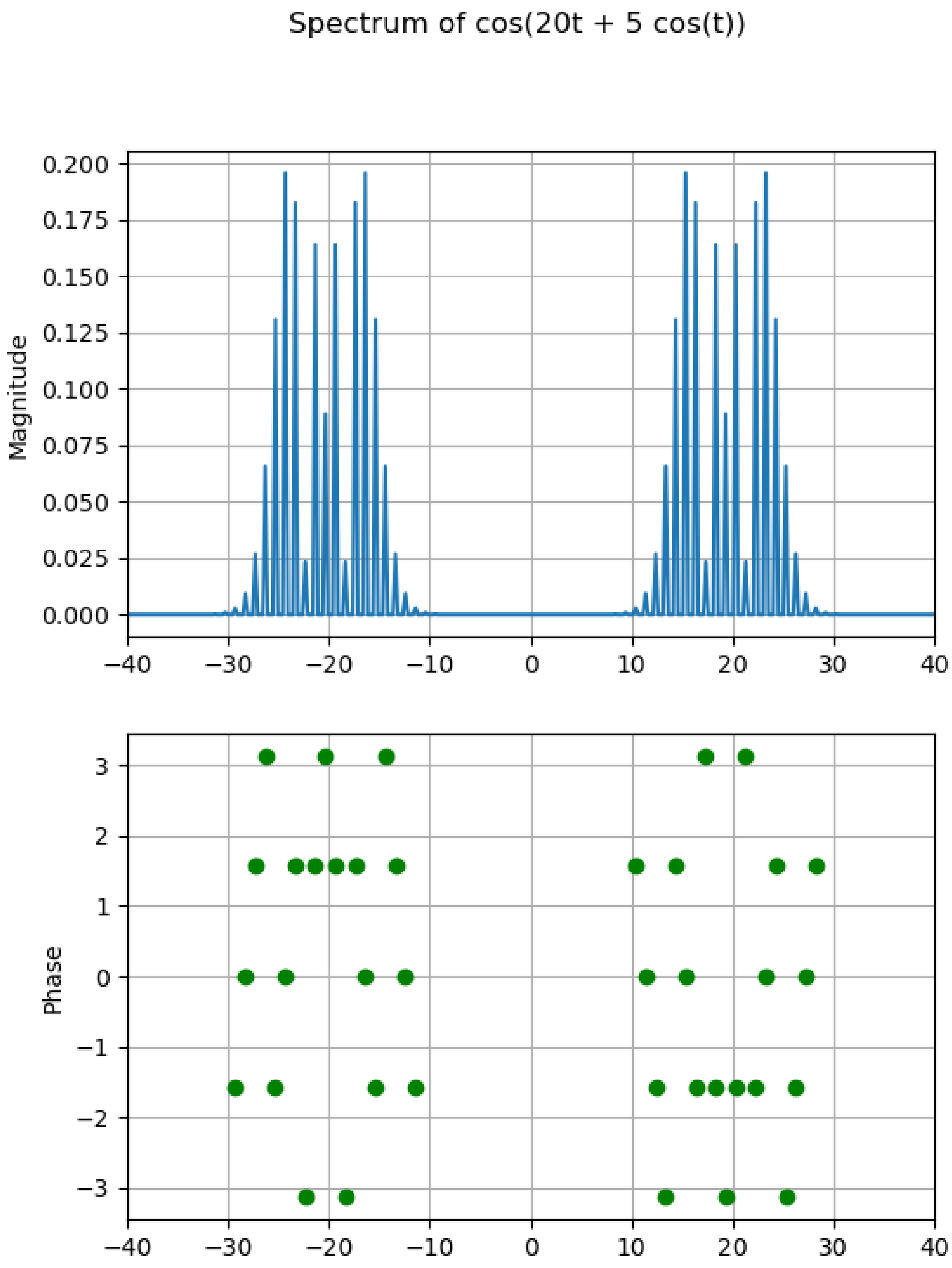
Spectrum of $cos(20t + 5cost)$

Consider the signal:

$$f(t) = \cos(20t + 5\cos(t)) \tag{5}$$

The Python code for above function is as follows:

```
x = np.linspace(-4*np.pi,4*np.pi,513)
x=x[:-1]
y = np.cos(20*x + 5*np.cos(x))
Y = fft.fftshift(fft.fft(y))/512
w = np.linspace(-64,63,513)
w=w[:-1]
fig,fig5 = plt.subplots(2,1,figsize=[6,8])
fig5[0].plot(w,abs(Y))
fig5[0].set_ylabel("Magnitude")
fig5[0].set_xlim([-40,40])
fig5[1].set_ylabel("Phase")
fig5[1].set_xlim([-40,40])
ii = np.where(abs(Y)>1e-3)
fig5[1].plot(w[ii],np.angle(Y[ii]),'go')
fig.suptitle("Spectrum of cos(20t + 5 cos(t))")
fig5[0].grid("true")
fig5[1].grid("true")
plt.grid("true")
plt.show()
```


Spectrum of cos(20t + 5 cos(t))
Magnitude
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−40
−30
−20
−10
0
10
20
30
40
Phase
3
2
1
0
−1
−2
−3
−40
−30
−20
−10
0
10
20
30
40

Spectrum of Gaussian Function

The Fourier transform of a signal $x(t)$ is defined as follows:

$$X(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{6}$$

As Gaussian Function tends to 0 for large values of t, we can approximate the above equation as shown below. And the appropriate values of T will be calculated using iterations

$$X(\omega) \approx \frac{1}{2\pi}\int_{-T/2}^{T/2} x(t)e^{-j\omega t}dt \tag{7}$$

The expression for the Gaussian is :

$$x(t) = e^{\frac{-t^2}{2}} \tag{8}$$

The CTFT is given by:

$$X(j\omega) = \frac{1}{\sqrt{2\pi}}e^{\frac{-\omega^2}{2}} \tag{9}$$

Below is the code for obtaining the spectrum of the Gaussian.

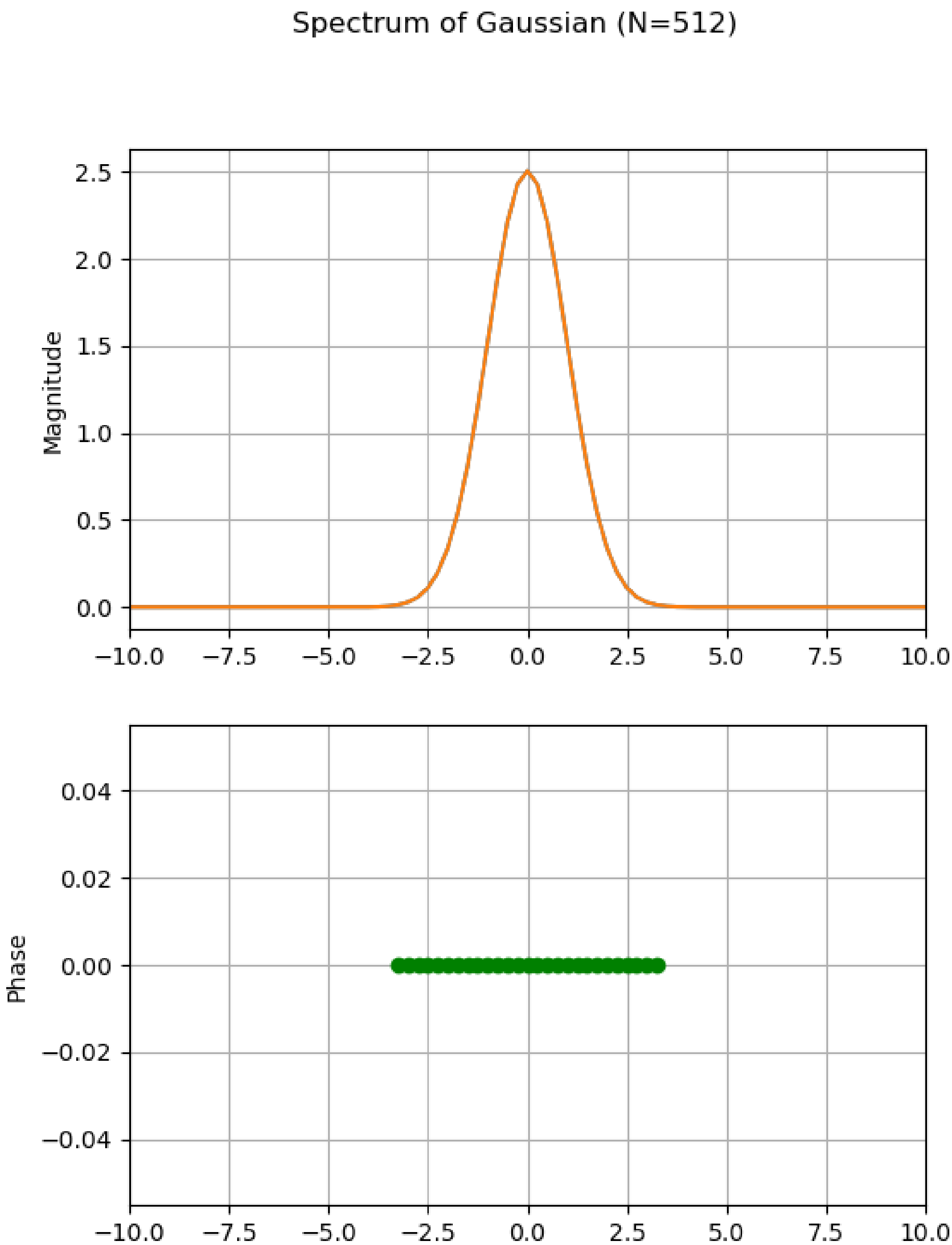
```
t = np.linspace(-4*np.pi,4*np.pi,513)
t=t[:-1]
y = np.exp(-0.5*pow(t,2))
w = np.linspace(-64,64,513)
w=w[:-1]

Y = fft.fftshift(abs(fft.fft(y)))/512
# Normalise the gaussian transfer function
Y = Y*np.sqrt(2*np.pi)/max(Y)
Y2 = np.exp(-w**2/2)*np.sqrt(2*np.pi)
# Calculate error in estimate
err = abs(Y-Y2).max()

fig,fig6 = plt.subplots(2,1,figsize=[6,8])
fig6[0].plot(w,abs(Y2),w,abs(Y))
fig6[0].set_ylabel("Magnitude")
fig6[0].set_xlim([-10,10])
fig6[1].set_ylabel("Phase")
fig6[1].set_xlim([-10,10])
ii = np.where(abs(Y)>1e-2)
```

```
fig6[1].plot(w[ii],np.unwrap(np.angle(Y[ii])),'go')
fig.suptitle("Spectrum of Gaussian (N=512)")
fig6[0].grid("true")
fig6[1].grid("true")
plt.show()
print(err)
```

Various time ranges were tried, the time range giving the best plot is chosen and is plotted below.



Spectrum of Gaussian (N=512)
Magnitude
Phase

Conclusion

We analysed the Discrete Fourier Transforms of sinusoids, amplitude and frequency modulated signals using FFT library in python. The pure sinusoids contained impulses at given frequncies. The frequency modulated wave contains many frequencies. Also we verified that DFT of a gaussian is gaussian function in w