EE2703 : Applied Programming Lab
Assignment 8

P N Neelesh Sumedh
ee19b047

May 8, 2021

1 Introduction

In this assignment, we look into Discrete Fourier Transform and find the DFTs of some signals using *numpy.fft* toolbox.

2 Accuracy of numpy.fft toolbox

We find the accuracy of the numpy.fft toolbox using the following code:

```
    x=p.rand(100)
    X=p.fft(x)
    y=p.ifft(X)
    p.c_[x,y]
    print ("absolute maximum error = ",abs(x-y).max())
```

The maximum error we get is 3.330862592745259e-16. Therefore the toolbox is accurate upto 16th decimal.

3 Spectrum of $\sin(5t)$

The function $\sin(5t)$ can be written as

$$y = \sin(5t) = \frac{e^{5t} - e^{-5t}}{2j}$$

Taking the fourier transform of this we get

$$Y(j\omega) = \frac{1}{2j}[\delta(\omega - 5) - \delta(\omega + 5)]$$

As we can see in the Fourier Transform, there are two peaks at $\omega = \pm 5$ with amplitude 0.5. Plotting this using the numpy toolbox we get the plots to be

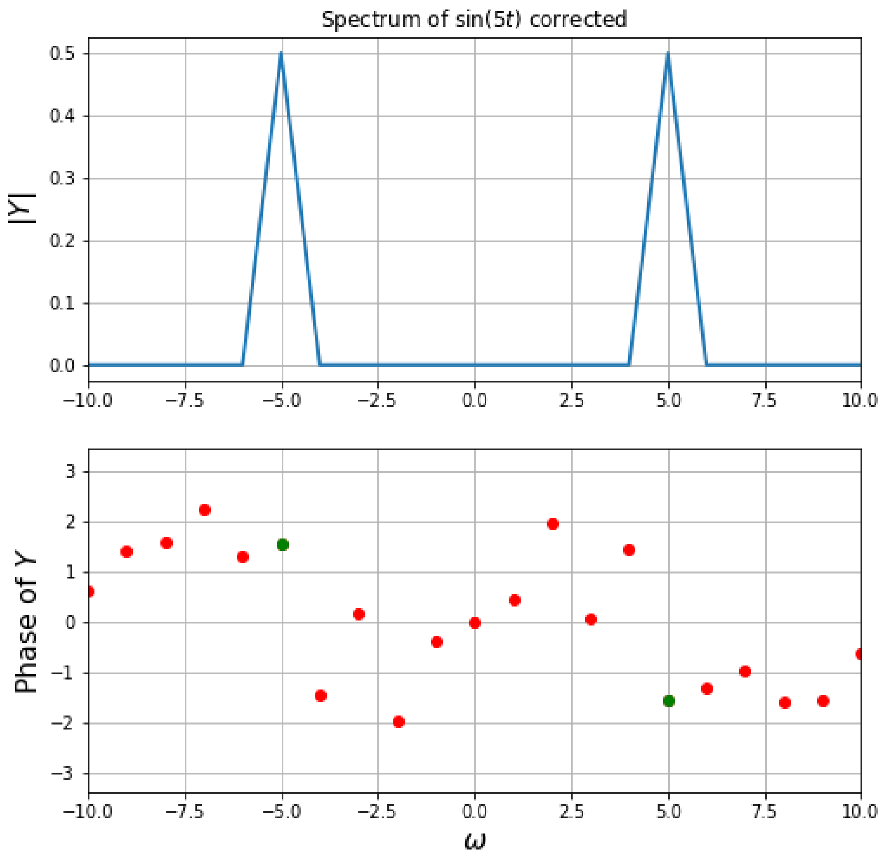

Figure 1: Spectrum of $\sin(5t)$

The code to plot the above graph is

```
x = p.linspace(0,2*p.pi,129);x=x[:-1]
y = p.sin(5*x)
w = p.linspace(-64,63,128)
Y = p.fftshift(p.fft(y))/128.0
p.figure(figsize=(8,8))
p.subplot(2,1,1)
p.plot(w,abs(Y),lw=2)
p.xlim([-10,10])
p.ylabel(r"$|Y|$",size=16)
p.title(r"Spectrum of $\sin(5t)$ corrected")
```

```
    p.grid()

    p.subplot(2,1,2)
    p.plot(w,p.angle(Y),'o',lw=2,color='r')
    ii = p.where(abs(Y)>1e-3)
    p.plot(w[ii],p.angle(Y[ii]),'o',lw=2,color='g')
    p.xlim([-10,10])
    p.ylabel(r"Phase of $Y$",size=16)
    p.xlabel(r"$\omega$",size=16)
    p.grid(True)
    p.savefig("test2.png")
    p.show()
```

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

The signal when expanded into complex exponential we get,

$$(1 + 0.1\cos(t))\cos(10t) = \frac{1}{2}(e^{10t} + e^{-10t}) + 0.1 \cdot \frac{1}{2} \cdot \frac{1}{2}(e^{9t} + e^{-9t} + e^{11t} + e^{-11t})$$

From the above equation, the Spectrum has peaks at $\omega = \pm 10, \pm 9, \pm 11$. At $\omega = \pm 10$, it has the maximum amplitude of 0.5 and at the remaining four frequencies it has an amplitude of 0.025. If we take less number of sample frequencies, then we get the plots to be

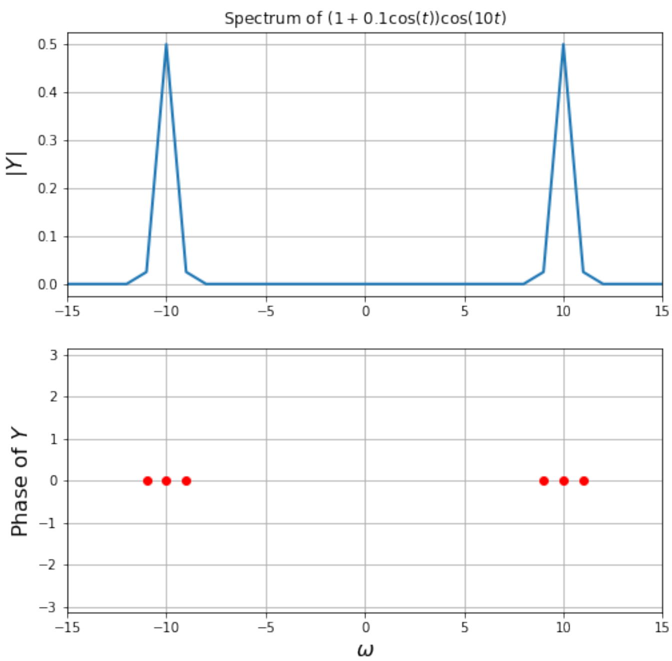

Figure 2: Spectrum of $(1 + 0.1\cos(t))\cos(10t)(inaccurate)$

Here we can see that, it does not tally with our equation. Therefore, to have accurate Spectrum, we should have sufficient number of sample frequencies. After taking sufficient number of sampling frequencies, we get the plots to be

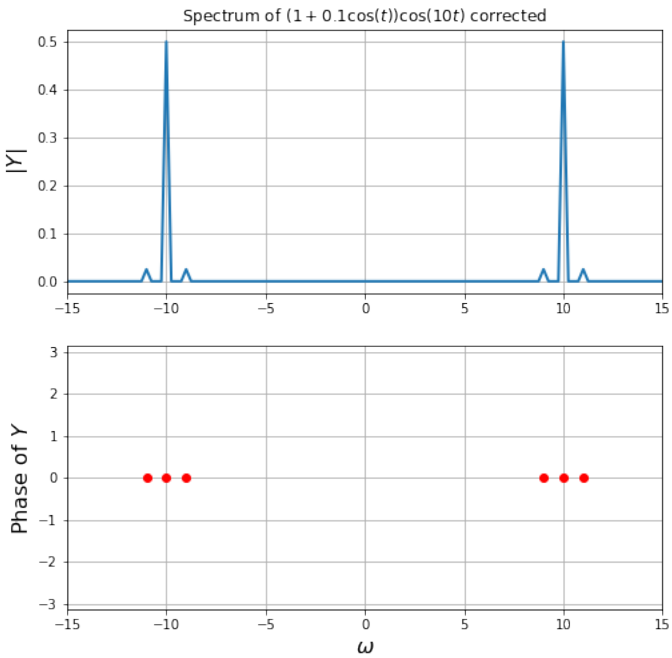

Figure 3: Spectrum of $(1 + 0.1\cos(t))\cos(10t)$

Now the plot tallies with out equation. Therefore, this is the spectrum of the signal $(1 + 0.1\cos(t))\cos(10t)$.

5 Spectrum of $\cos^3(t)$

The signal can be written as

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

Therefore, by observing from the previous signals, we can say that the Spectrum of $\cos^3(t)$ has peaks at $\omega = \pm 1$ and ± 3. The spectrum of the function is shown below:

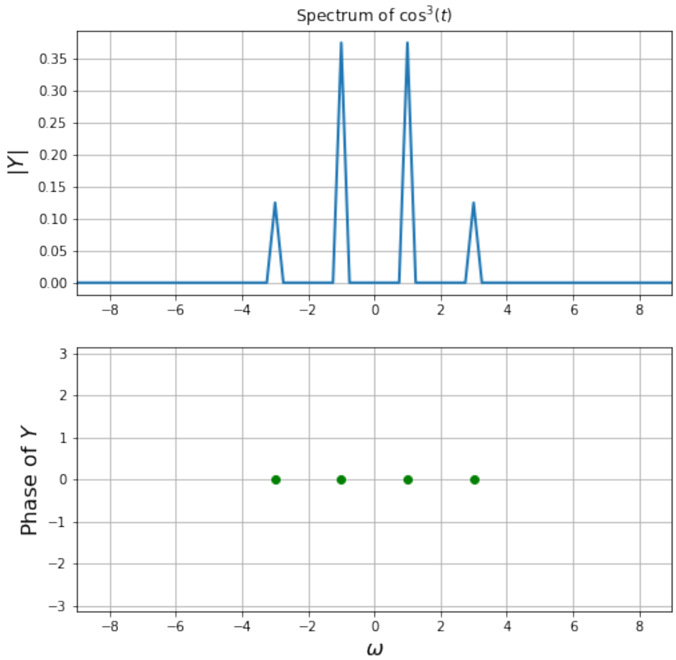

Figure 4: Spectrum of $\cos^3(t)$

As expected, we have peaks at $\omega = \pm 1$ and ± 3. The phase is zero as the signal is function of $\cos(t)$.

6 Spectrum of $\sin^3(t)$

The signal can be written as

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

The Spectrum has peaks at $\omega = \pm 1$ and ± 3. The plots are shown below.

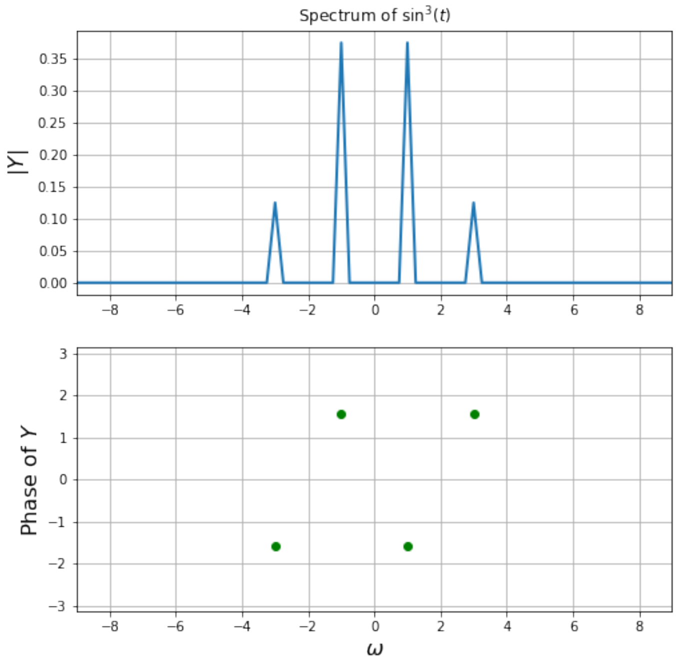

Figure 5: Spectrum of $\sin^3(t)$

As expected, the peaks are at the correct places. The phase is either $\frac{\pi}{2}$ or $-\frac{\pi}{2}$ which is also correct.

7 Spectrum of $\cos(20t + 0.1\cos(t))$

Using the *numpy.fft* toolbox, the spectrum of the function can be found out. The code to find the Spectrum:

```
x = p.linspace(-4*p.pi,4*p.pi,513);x=x[:-1]
y = p.cos(20*x+5*p.cos(x))
w = p.linspace(-64,64,513); w = w[:-1]
Y = p.fftshift(p.fft(y))/512.0
p.figure(figsize=(8, 8))
p.subplot(2,1,1)
p.plot(w,abs(Y),lw=2)
p.xlim([-35,35])
p.ylabel(r"$|Y|$",size=16)
p.title(r"Spectrum of $\cos(20t + 5\cos(t))$")
p.grid()

p.subplot(2,1,2)
ii = p.where(abs(Y)>1e-3)
p.plot(w[ii],p.angle(Y[ii]),'o',lw=2,color='g')
```

```
    p.xlim([-35,35])
    p.ylabel(r"Phase of $Y$",size=16)
    p.xlabel(r"$\omega$",size=16)
    p.grid(True)
    p.ylim([-4, 4])
    p.savefig("Q3.png")
    p.show()
```

The DFT of signal $\cos(20t + 0.1\cos(t))$:

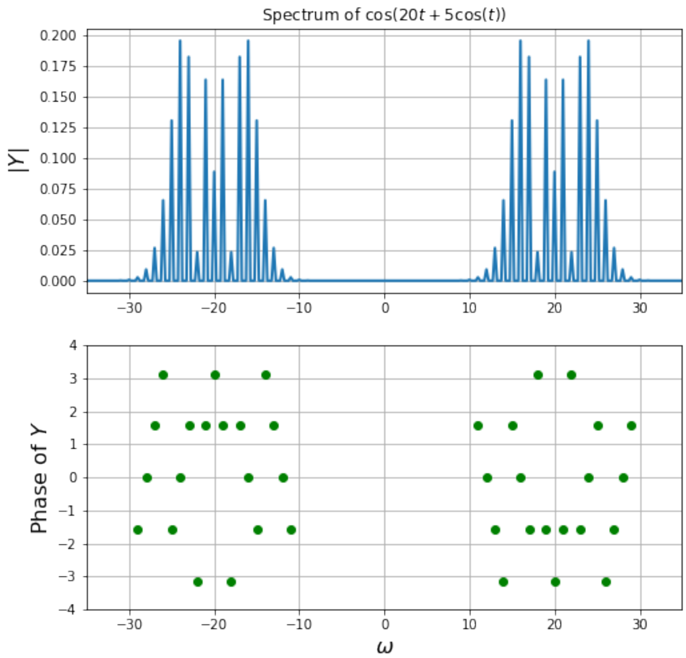

Figure 6: Spectrum of $\cos(20t + 0.1\cos(t))$

The signal has frequencies mostly around $\omega = 20$.

8 Spectrum of Gaussian

The Gaussian function is

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

After using the formula to find the DFT of the Gaussian, we get the DFT of gaussian to be

$$X(j\omega) = \sqrt{2\pi}e^{-0.5\omega^2}$$

The DFT of Gaussian is found out using *numpy.fft* toolbox. This graph is plotted along with the true spectrum found out from the mathematical equation to compare the both methods. The code to plot the estimated and true spectrum:

```
T = 8*p.pi
samp_freq = 8/512
min_error = 1e-6
error = 1+min_error
Yold = 0
while error>min_error:
    N = int(T/p.pi/samp_freq)
    x = p.linspace(-T/2,T/2,N+1);x=x[:-1]
    y = p.exp(-0.5*x**2)
    w = p.linspace(-1/samp_freq,1/samp_freq,N+1); w = w[:-1]
    Y = p.fftshift(p.fft(p.ifftshift(y)))*T/N
    error = sum(abs(Y[::2]-Yold))
    Yold = Y
    T = 2*T
    print(error)
actual_dft = p.sqrt(2*p.pi)*p.exp(-0.5*w**2)
p.figure(figsize=(8, 8))

p.subplot(2,1,1)
p.plot(w,abs(Y),lw=3.5,label='Estimate')
p.plot(w,abs(actual_dft),'--',lw=3.5,label='Expected')
p.xlim([-5,5])
p.ylabel(r"$|Y|$",size=16)
p.legend()
p.title(r"Spectrum of $e^{(\frac{-t^2}{2})}$",size = 16)
p.grid()

p.subplot(2,1,2)
ii = p.where(abs(Y)>1e-3)
p.plot(w[ii],p.angle(Y[ii]),'o',lw=2,color='g')
p.xlim([-5,5])
p.ylabel(r"Phase of $Y$",size=16)
p.xlabel(r"$k$",size=16)
p.grid(True)
p.ylim([-p.pi, p.pi])
p.savefig("fig4.png")
p.show()
```

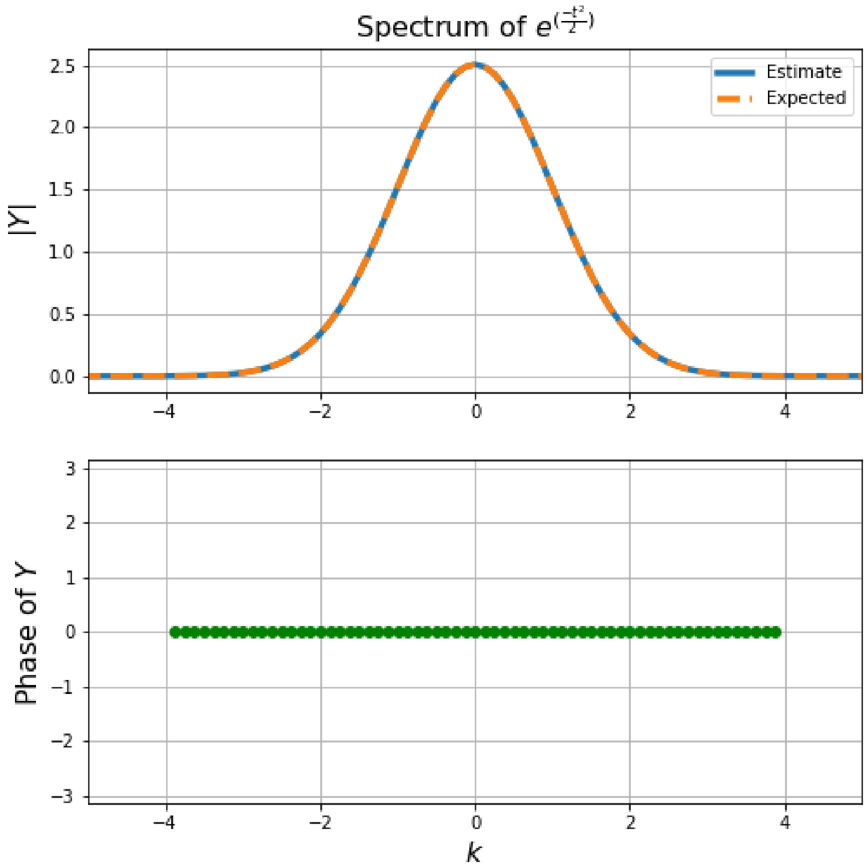

Figure 7: Gaussian Spectrum

Therefore, the estimated and the true spectrum both overlap. Thus, the spectrum estimated by the toolbox is very much accurate. The error between the true spectrum and the estimated one is 8.466806211914774e-14. The time period taken is between [-4π,4π] and the number of samples taken is 512.

9 Conclusion

Thus we have found out DFTs of some important signals using the *numpy.fft* toolbox. We found out that to get an accuarate spectrum, we should have more number of samples. Another information to add is that the DFT of a Gaussian is also a Gaussian.