EE2703 : Applied Programming Lab
Experiment 8

Ch Venkat Sai
EE18B042

April 22, 2020

Overview

The experiment is about:
Learning the concepts of DFT and also using it in python with a command called fft and making certain required changes to the corresponding plots to make them match them with the expected plots.

Theory

A finite energy function of the type f(t) has a Fourier Transform (and its inverse):

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-st}dt \tag{1}$$

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

If f(t) is periodic with period 2, the Fourier Transform collapses to the Fourier Series:

$$f(t) = \sum_{n=-\inf}^{\inf} c_n e^{jnt} \tag{3}$$

$$c_n = \frac{1}{2\pi}\int_{0}^{2\pi} f(t)e^{-jnt}dt \tag{4}$$

We can invert this picture and say, suppose f [n] are the samples of some function f(t), then we define the Z transform as:

$$F(z) = \sum_{n=-\inf}^{\inf} f(n)z^{-n} \tag{5}$$

Replacing z with $e^{j\theta}$ we get

$$F(e^{j\theta} = \sum_{n=-\inf}^{\inf} f(n)e^{-jn\theta} \tag{6}$$

So clearly F(z) is like the periodic time function that gives rise to the fourier series whose coefficients are the samples f[n]. $F(e^{j\theta})$ is continuous and periodic. f [n] is discrete and aperiodic. Suppose now f [n] is itself periodic with a period N, i.e.,

$$f(n) = f(n+N) \tag{7}$$

Then, it should have samples for its DTFT. This is true, and leads to the Discrete Fourier Transform or the DFT:
Suppose f [n] is a periodic sequence of samples, with a period N. Then the DTFT of the sequence is also a periodic sequence F[k] with the same period N. So we have:

$$F[k] = \sum_{n=0}^{N-1} f(n)W^{nk} \tag{8}$$

$$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} F[k]W^{-nk} \tag{9}$$

Here $W = e^{\frac{-2\pi j}{N}}$ is used to make the equations less cluttered. The values F[k] are what remains of the Digital Spectrum $F(e^{j\theta}$. We can consider them as the values of $F(e^{j\theta})$ for $\theta = \frac{2k}{N}$

Q1

We now do an example of sin(5x). We use the fft command to het the DFT. The spikes give a height of 64, not 0.5. We should divide by N to use it as a spectrum.
The problem is that the DFT treats the position axis as another frequency axis. So it expects the vector to be on the unit circle starting at 1. Our position vector started at 0 and went to 2π, which is correct. The fft gave an answer in the same value. So we need to shift the π to 2π portion to the left as it represents negative frequency. This can be done with a command called fftshift.
We plot the magnitude spectrum and phase spectrum with omega.

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

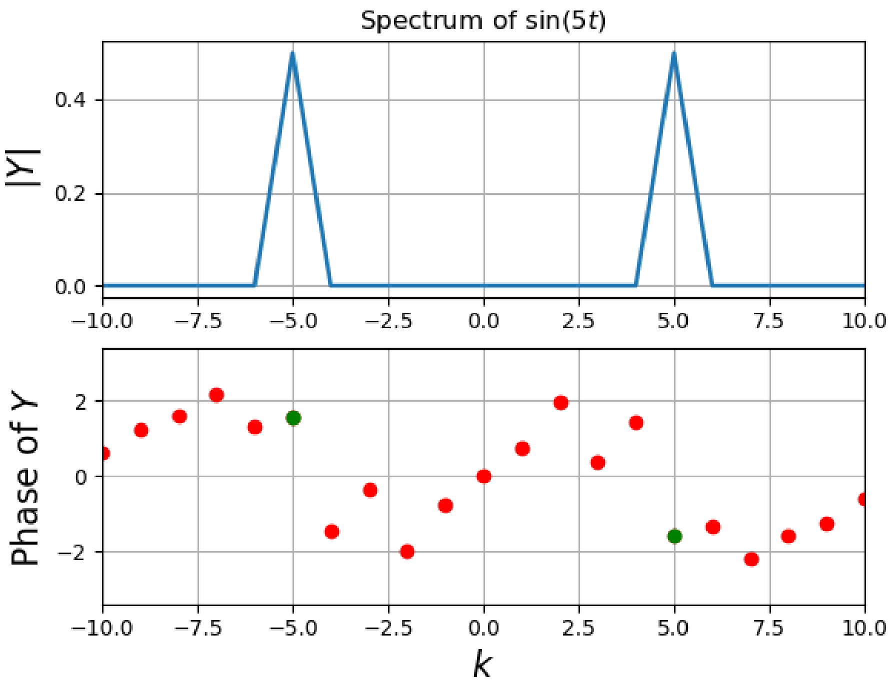


Figure 1: sin(5x)

```
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
ii=where(abs(Y)>1e-3)
plot(w[ii],angle(Y[ii]),'go',lw=2)
xlim([-10,10])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$k$",size=16)
grid(True)
show()
```

The plots are as shown in figure 1.

Q2

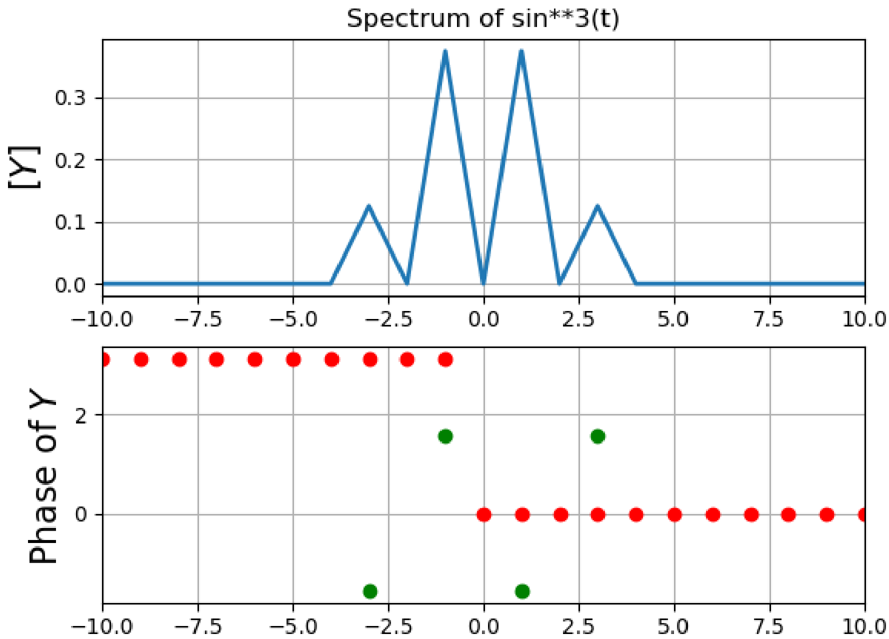
Having obtained the plots for sin(5x) we now try to obtain the plots of the spectrum of $\sin^3 x$ and $\cos^3 x$

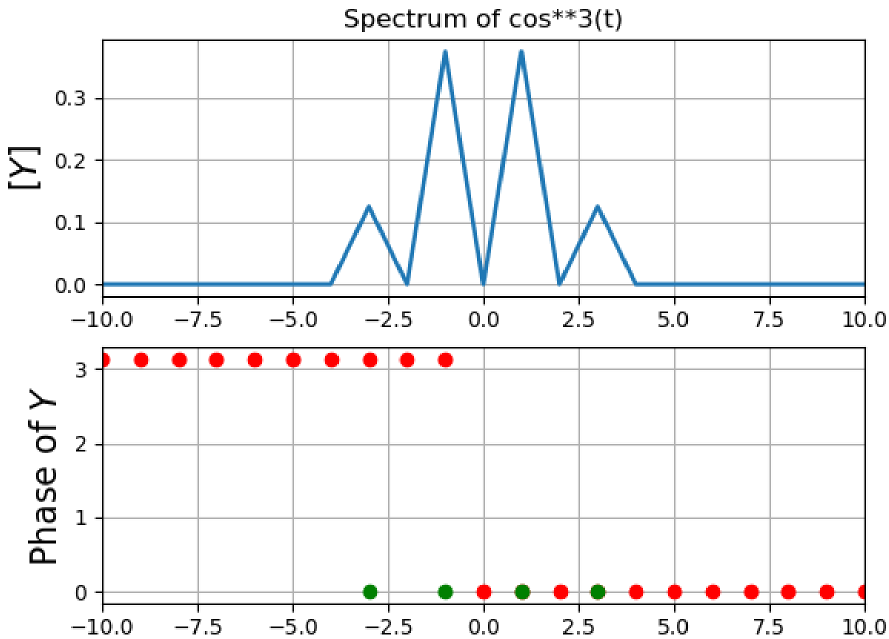
```
y=sin(x)*sin(x)*sin(x)
```

```
Y=fftshift(fft(y))/128
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
ylabel(r"$[Y]$",size=16)
title("Spectrum of sin**3(t)")
xlim([-10,10])
grid(True)
subplot(2,1,2)
plot(w,angle(w),'ro',lw=2)
ii=where(abs(Y)>1e-3)
plot(w[ii],angle(Y[ii]),'go',lw=2)
ylabel(r"Phase of $Y$",size=16)
xlim([-10,10])
grid(True)
show()

y=cos(x)*cos(x)*cos(x)
Y=fftshift(fft(y))/128
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
ylabel(r"$[Y]$",size=16)
title("Spectrum of cos**3(t)")
xlim([-10,10])
grid(True)
subplot(2,1,2)
plot(w,angle(w),'ro',lw=2)
ii=where(abs(Y)>1e-3)
plot(w[ii],angle(Y[ii]),'go',lw=2)
ylabel(r"Phase of $Y$",size=16)
xlim([-10,10])
grid(True)
show()
```

The figures turn out to be like those shown in figure 2 and figure 3.


Spectrum of sin**3(t)
[Y]
Phase of Y
0.0
0.1
0.2
0.3
2
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


Spectrum of cos**3(t)
[Y]
Phase of Y
0.0
0.1
0.2
0.3
0
1
2
3
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Q3

We now try the following equation:

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

In this particular example we need more data point to actually capture the whole plot properly. So we take 512 points from -4π to 4π.

```
t=linspace(-4*pi,4*pi,513);t=t[:-1]
y=(1+0.1*cos(t))*cos(10*t)
Y=fftshift(fft(y))/512.0
w=linspace(-64,64,513)
w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-15,15])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $\left(1+0.1\cos\left(t\right)\right)\cos\left(10t\right)$")
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
xlim([-15,15])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
show()
```

The plots are as shown in fig 4.

Q4

We now try the same for cos(20t + cos(5t)).

```
y=cos(20*t + cos(5*t))
Y=fftshift(fft(y))/512.0
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-20,20])
ylabel(r"$|Y|$",size=16)
```

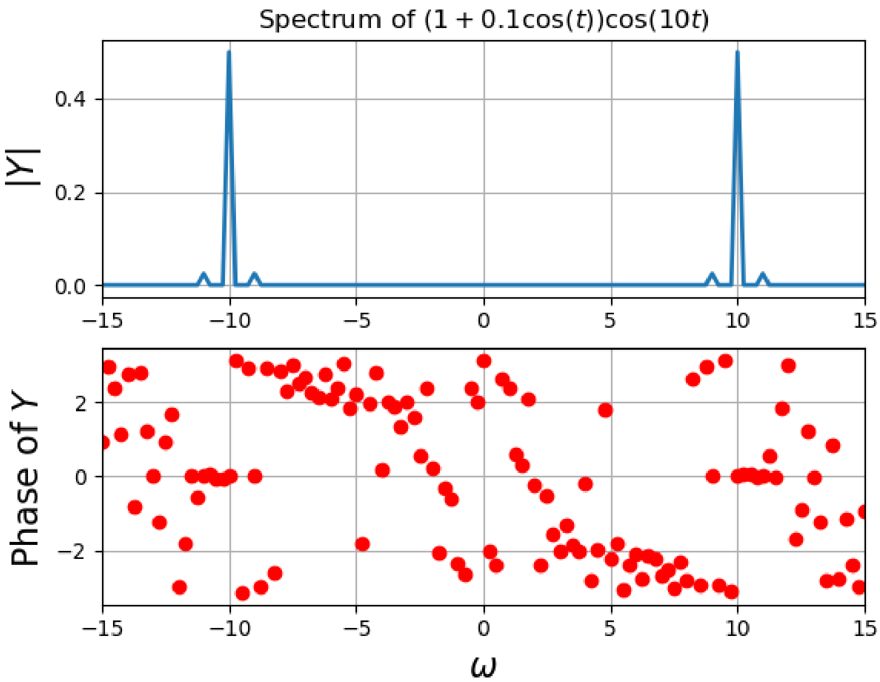


Figure 4: (1 + 0.1cos(t))cos(10t)

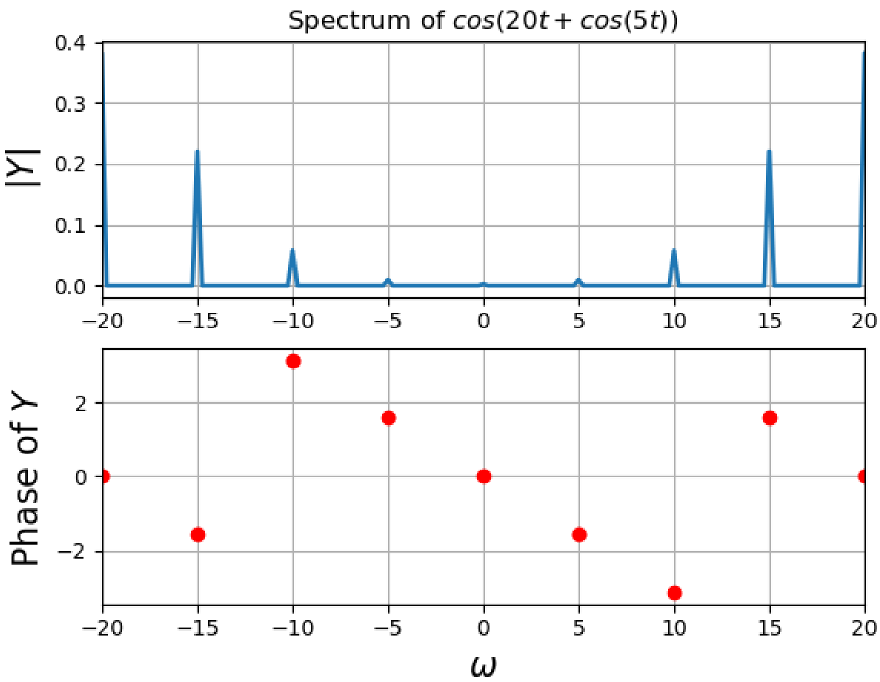


Figure 5: cos(20t + cos(5t))

```
title(r"Spectrum of $cos(20t + cos(5t))$")
grid(True)
subplot(2,1,2)
ii=where(abs(Y)>1e-3)
plot(w[ii],angle(Y[ii]),'ro',lw=2)
xlim([-20,20])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
show()
```

The plot turns out as shown in the figure 5.

Q5

Now we plot the same spectrum for $e^{\frac{-t^2}{2}}$

```
y=exp(-t**2/2)
```

```
Y=fftshift(fft(y))/512
Y=Y*8*pi
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-20,20])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $e(-t**2/2)$")
grid(True)
subplot(2,1,2)
ii=where(abs(Y)>1e-3)
plot(w[ii],angle(Y[ii]),'ro',lw=2)
xlim([-20,20])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
show()
```

The plot turns out to be as shown as in figure 6.

Conclusion

In this assignment, we explored how to obtain the DFT, and how to recover the analog Fourier Tranform for some known functions by the proper sampling of the function.

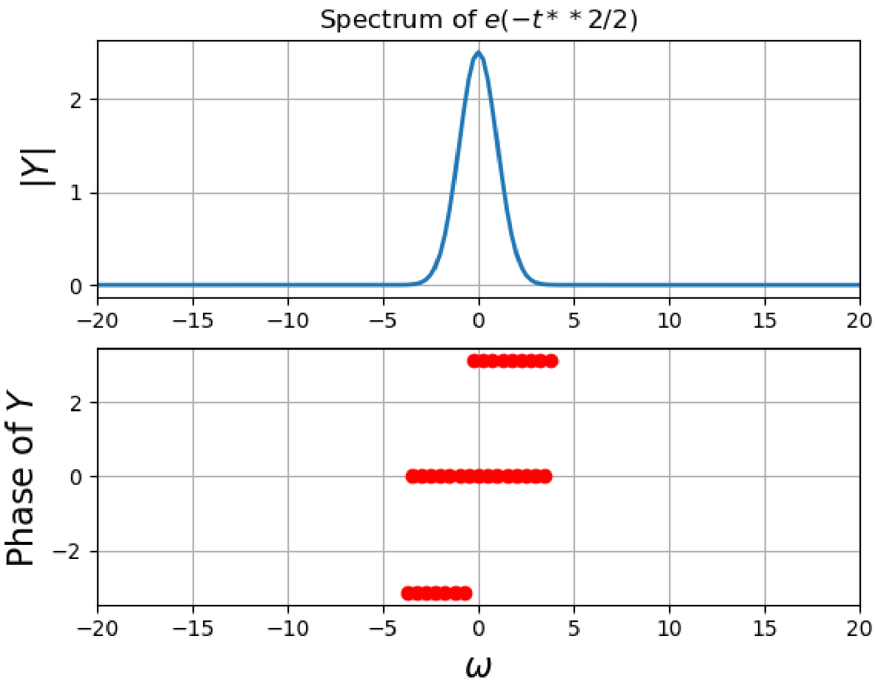


Figure 6: exp(-t**2/2)