EE2703: Applied programming Lab
Assignment 9 : The Digital Fourier Transform

Vishnu Varma V
EE19B059

April 28, 2021

1 Introduction

In this Assignment we implement Digital Fourier Transform using the Fast Fourier Transform(fft) module in numpy.

2 Error

The following code displays what is the order of error, when fft of a signal is taken and inverse fft of the that signal is taken we essentially get the same signal but there would be error we see what would the order of this be :

```
from pylab import *
x=rand(100)
X=fft(x)
y=ifft(X)
c_[x,y]
print(abs(x-y).max())
```

This gives the error to be: 3.336112599886741e-16
We can notice that the error is very small and we notice that x is purely real while ifft is very slightly complex this is due to the finite accuracy of the CPU so that the ifft could not exactly undo what fft did.

3 Sinusoid

Now we consider taking the fft of $sin(x)$

$$y = sin(x) = \frac{e^{jx} - e^{-jx}}{2j}$$

The expected spectrum would be :

$$Y(w) = \frac{\delta(w-1) - \delta(w+1)}{2j}$$

Executing the following code:

```
from pylab import *
x=linspace(0,2*pi,128)
y=sin(5*x)
Y=fft(y)
figure()
subplot(2,1,1)
plot(abs(Y),lw=2)
grid(True)
subplot(2,1,2)
plot(unwrap(angle(Y)),lw=2)
grid(True)
show()
```

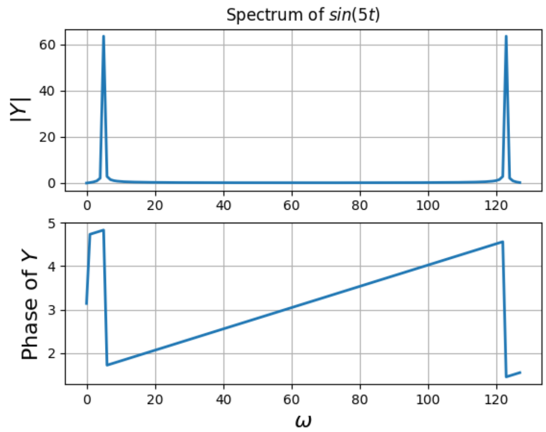

Figure 1: Spectrum of $sin(5t)$

• We get spikes as expected. But not where we expected. There is energy at nearby frequencies as well.
• The spikes have a height of 64, not 0.5. We should divide by N to use it as a spectrum.
• The phase at the spikes have a phase difference of π, which means they are opposite signs, which is correct
• The actual phase at the spikes is near but not exactly correct.
• We haven't yet got the frequency axis in place
So we use the following code to get all the aspects we want correctly.

```
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 $\sin(5t)$")
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"$\omega$",size=16)
grid(True)
show()
```

Using the above code we get the expected plot as follows:

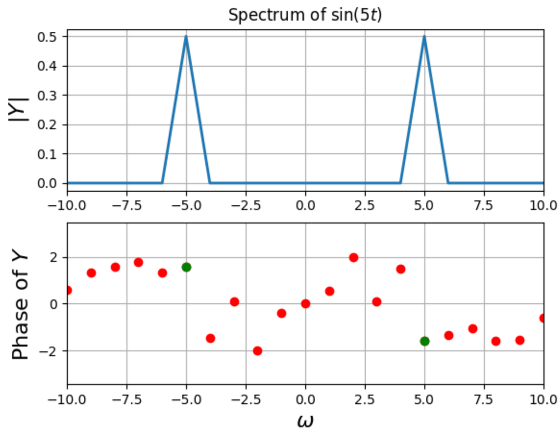

Figure 2: Spectrum of $sin(5t)$

4 Amplitude Modulation

Now we consider the amplitude modulated signal as follows:

$$f(t) = (1 + 0.1cos(t))cos(10t)$$

Using the same code as above replacing the function sin(5t) with the Amplitude modulated signal, but we expect three spikes but using the same parameters we get only one broader spike on both sides so we have to do some changes to get the plot we desire. The Problem was we did not allow for enough frequencies so we stretch the t axis and increase the number of points considered to 513 while neglecting the last one, This will give us tighter spacing between frequency samples, but the same time spacing means the same sampling frequency. The Plot obtained is as follows:

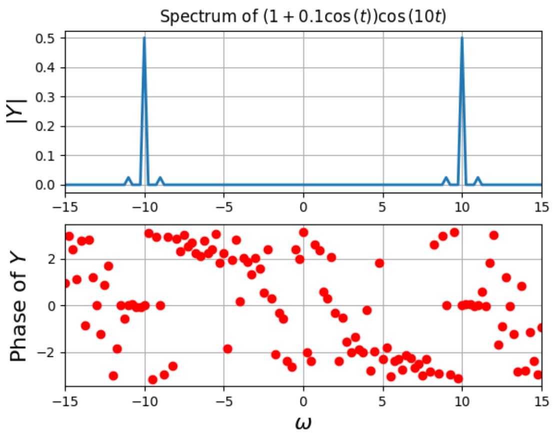

Figure 3: Spectrum of $(1 + 0.1cos(t))cos(10t)$

5 Assignment

Now we consider the sinusids $sin^3(t)$ and $cos^3(t)$ In the frequency domain we expect 4 impulses, with two impulses on each side and the impulses with higher frequency will have frequency thrice the frequency of the impulses with lower frequency. The Spectrum obtained is plotted as follows:

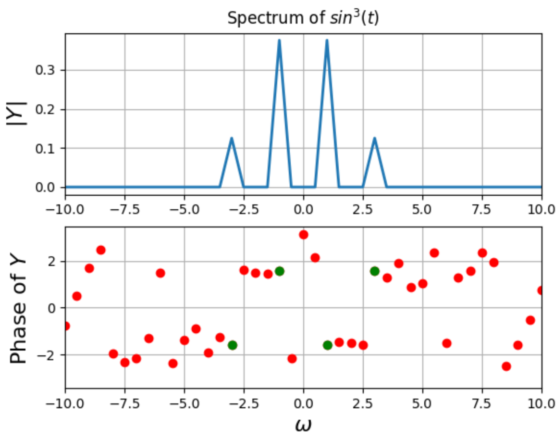

Figure 4: Spectrum of sin^3t

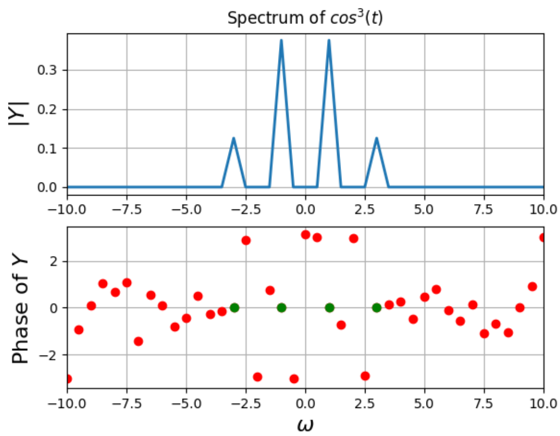

Figure 5: Spectrum of $cos^3 t$

6 Frequency Modulation

Now we consider the frequency modulated signal $cos(20t + 5cos(t))$, and plot the phase points only when the magnitude is greater than 10^{-3}. We can observe that the number of peaks have increased, and the sidebands are comparable to the main signal. Using the same code with this frequency modulated signal as input would yield the following plot:

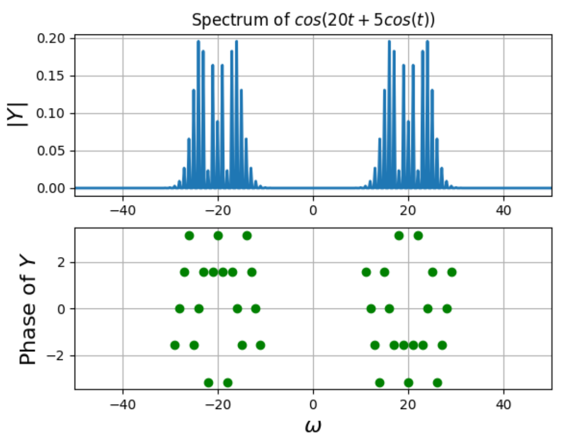

Figure 6: Spectrum of $cos(20t + 5cos(t))$

7 Gaussian

The given function is the gaussian function which is given by :

$$f(t) = e^{-t^2/2}$$

This function is not bandlimited, the frequency spectrum has non zero values even for large frequencies. We can approximate the Fourier Transform of the gaussian where the limits are from $-\inf$ to $\inf$ but considering a window in which this gaussian can be evaluated and making this window quite large to get a proper approximation for the fourier transform. The CTFT of this gaussian function is given by

$$F(w) = \sqrt{2}\pi e^{-w^2/2}$$

We expect the phase of this function to be zero, and the magnitude plot would be a Gaussian Function. The Error using this approximation was found to be of the order of 10^{-10} using a fixed N and varying the time limit. The function converged at time limit 2π for both N = 256 and 512. The Error seems to be decreasing as the value of N increases but it decreases slightly. Varying the limit of t in linspace showed a variation in error, and varying the limit of ω the error was found varying, varying the sampling rate also varies error. Here are the plots obtained for time limit : 2π and N: 256 and 512:

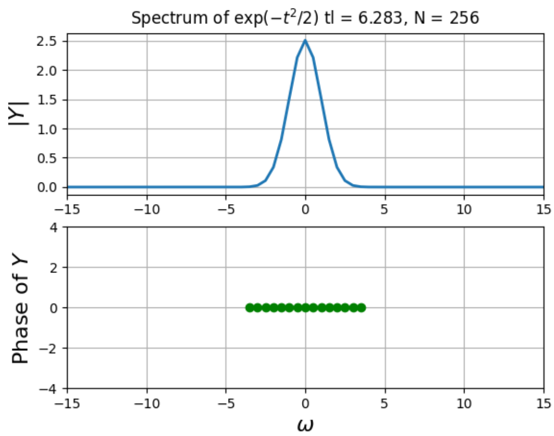

Figure 7: Spectrum of $e^{-t^2/2}$ with $N = 256$

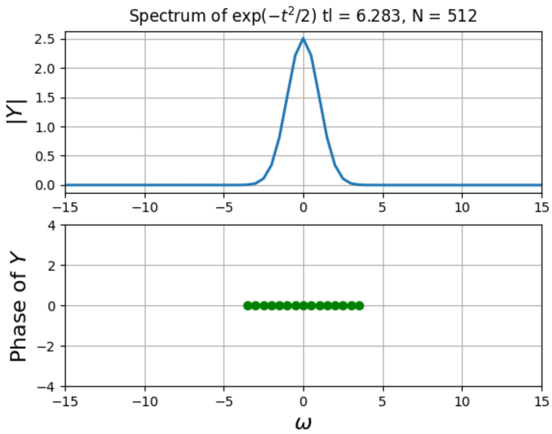

Figure 8: Spectrum of $e^{-t^2/2}$ with N as 512

The error for varied parameters and the time range for which the dft of the gaussian is accurate to 6th digit is displayed as follows:

```
For N = 256 : error = 8.382019522912287e-10 and time limit = 6.2832
Error for t limit = 6.2832, N = 256, is 8.382019522912287e-10
Error for t limit = 6.2832, N = 512, is 8.331486611723449e-10
```

8 Conclusion

We implemented Digital Fourier Transform using the Fast Fourier Transform(fft) module in numpy. The DFT's of Sinusoids, Amplitude Modulated Signals, Frequency modulated signals were found. There were impulses at the sinusoidal frequencies for normal sinusoids, while we got an impulse at carrier frequencies and sidebands for amplitude modulated signal, and there were large number of sideband frequencies for a frequency modulated signal, and for a gaussian the magnitude was Gaussian while the phase was zero. The time range for which the dft of the gaussian is accurate to 6 digits was found. Overall all the plots satisfied the expectations with very small error due to finite accuracy of the CPU, but helped us in analysing the DFT of the signals.