EE2703 : Applied Programming Lab
Assignment 8
The Digital Fourier Transform

Debojyoti Mazumdar
EE20B030

May 2, 2022

0.1 Abstract

Aim of this assignment is to know about DFTs and how to get them using numpy.fft.

0.2 Introduction

In this assignment, we will understand more about DFTs. We will use the numpy.fft module to generate the DFTs of the given example time domain functions.

0.3 Assignment

0.3.1 Question 1

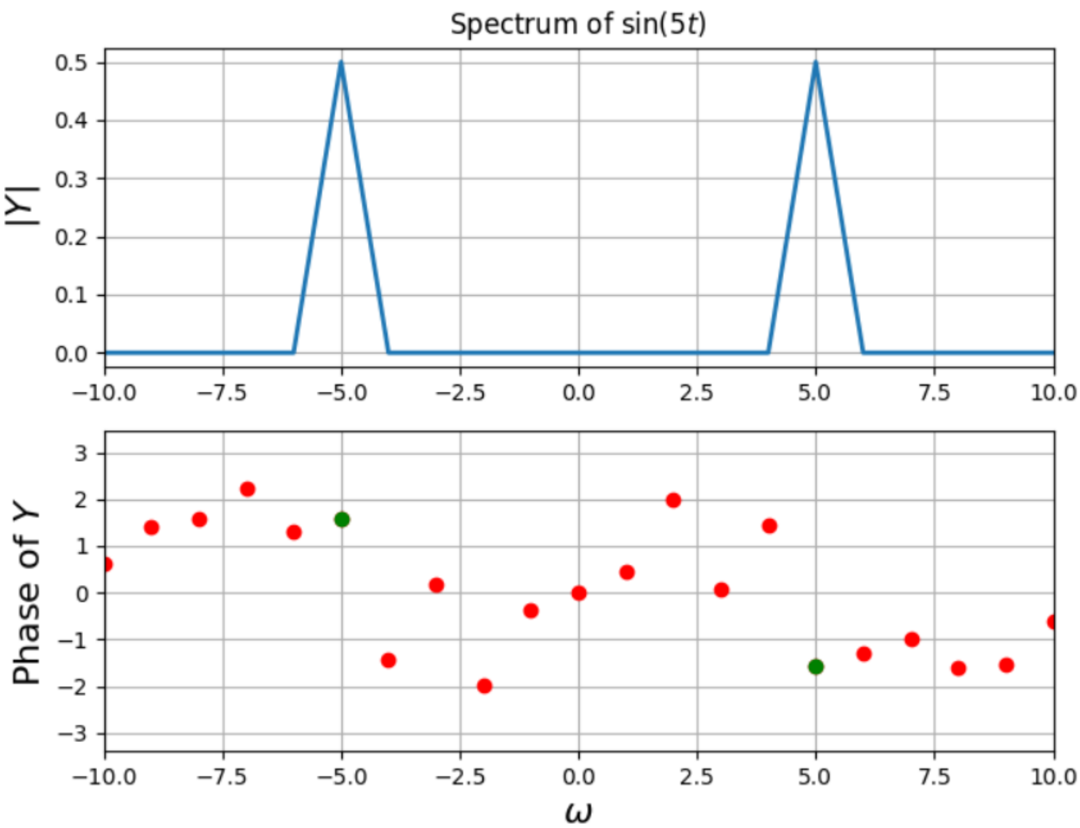
Here, we are going to plot the spectrum and the phase of the fourier transform of sin(5t). We know that,

$$y = sin(5t) = \frac{e^{j5t} - e^{-j5t}}{2j}$$

And the Fourier transform of this function is :

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

Now using the code given in the lab manual we get the spectrum and the phase of the fourier transform as shown below:



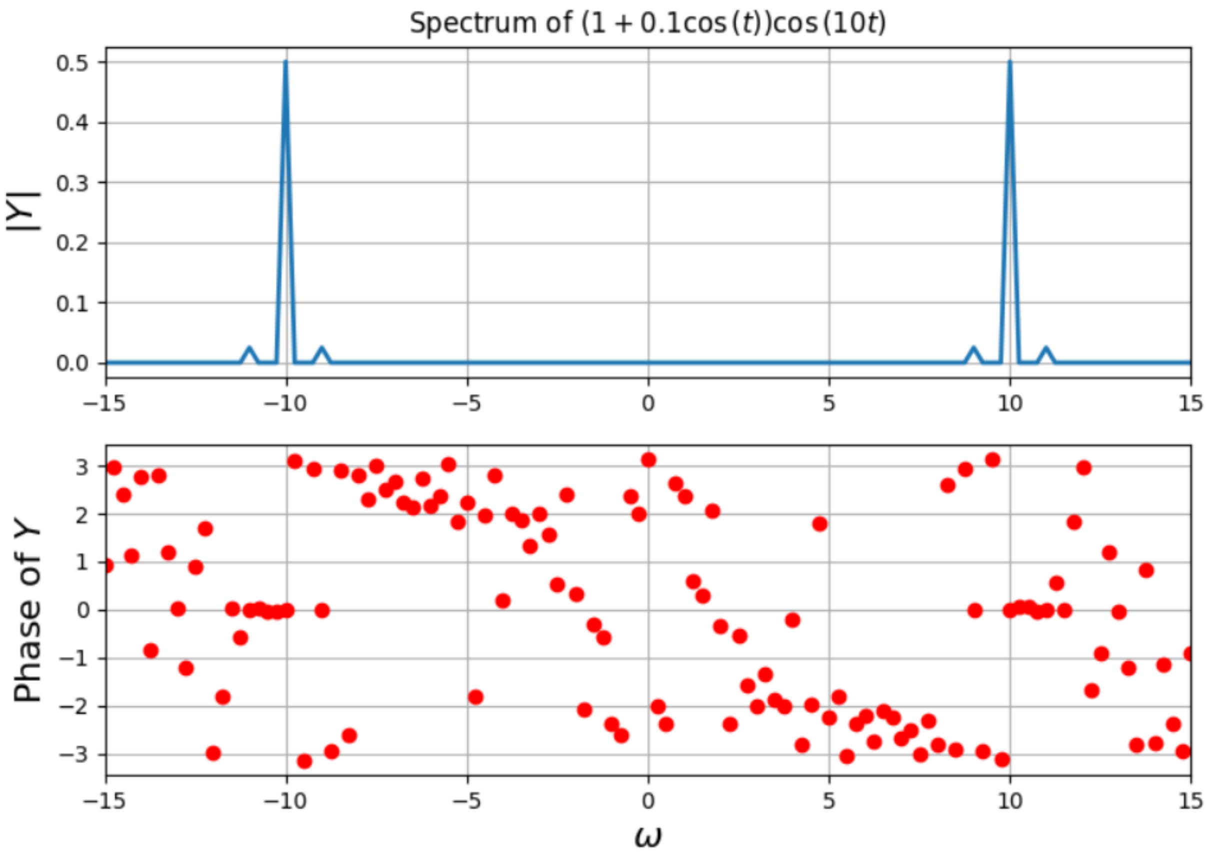
Then, we are now going to plot the spectrum and the phase of the fourier transform of the following function:

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

The fourier transform of the above function is :

$$Y(\omega) = \frac{20\delta(\omega - 10) + \delta(\omega - 11) + \delta(\omega - 9) + 20\delta(\omega + 10) + \delta(\omega + 11) + \delta(\omega + 9)}{40}$$

Now following the code given in the manual we get the following spectrum and the phase of the fourier transform as shown below:



0.3.2 Question 2

Now we plot the spectrum of the fourier transform of $\sin^3(t)$.

We know

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

Which can be also written as follows

$$y(t) = \frac{3e^{jt} - 3e^{-jt}}{8j} - \frac{e^{3jt} - e^{-3jt}}{8j}$$

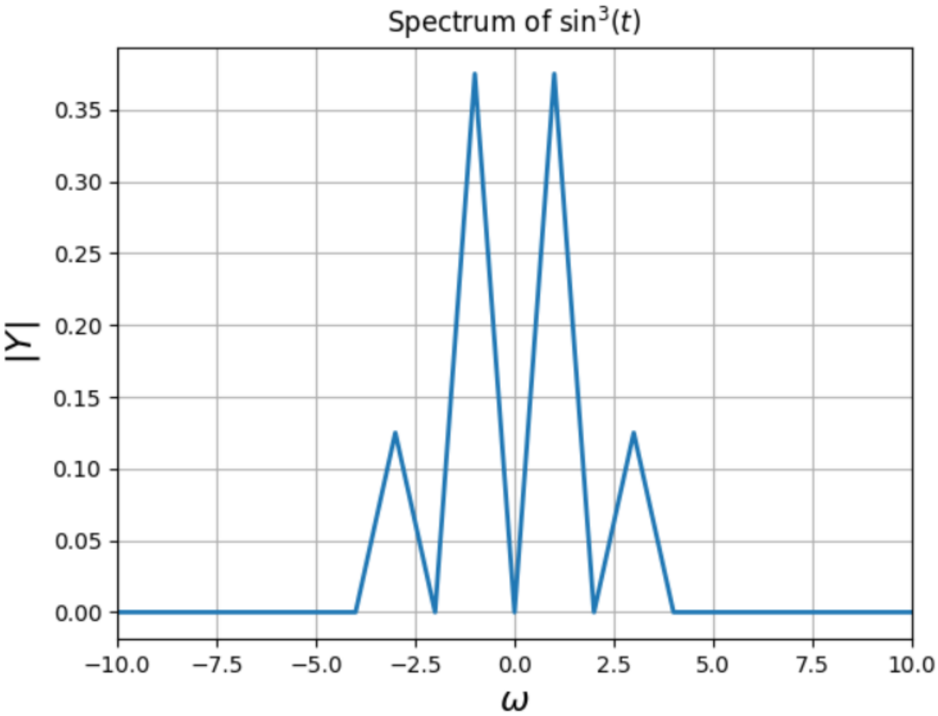
Now the fourier transform of the above function is

$$Y(\omega) = \frac{3\delta(\omega-1) - 3\delta(\omega+1)}{8j} + \frac{\delta(\omega+3) - \delta(\omega-3)}{8j}$$

Now using the code used in Question-1 example-1 but changing only y as follows

```
y = sin(t)**3
```

We get the spectrum as shown below.



Now we plot the spectrum of the fourier transform of $\cos^3(t)$.

We know

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

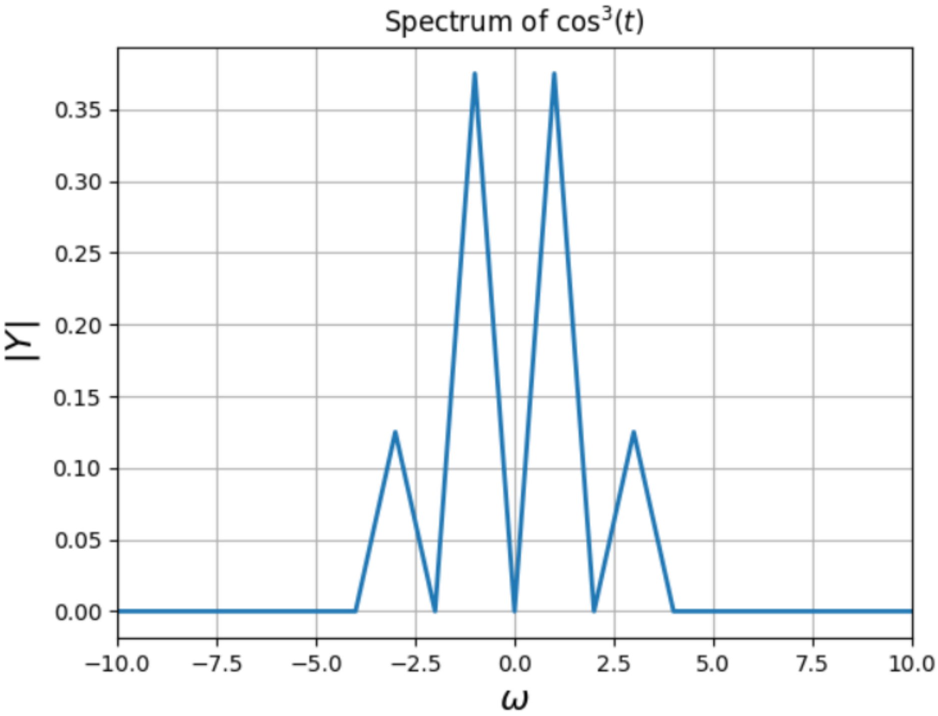
Which can be also written as follows

$$y(t) = \frac{3e^{jt} + 3e^{-jt}}{8} + \frac{e^{3jt} + e^{-3jt}}{8}$$

Now the fourier transform of the above function is

$$Y(\omega) = \frac{3\delta(\omega-1) + 3\delta(\omega+1)}{8} + \frac{\delta(\omega+3) + \delta(\omega-3)}{8}$$

Now doing the same thing as done in the previous part we get the following spectrum of fourier transform as shown below.



0.3.3 Question 3

Now we plot the fourier transform of the following function.

$$y(t) = \cos(20t + 5\cos(t))$$

So we write the following code to generate the fourier transform.

```
t = linspace(-4*pi, 4*pi, 513)
t = t[:-1]
y = cos(20*t+5*cos(t))
Y = fftshift(fft(y))/512.0
```

Now to plot the values we need to generate ω. So we write the following code.

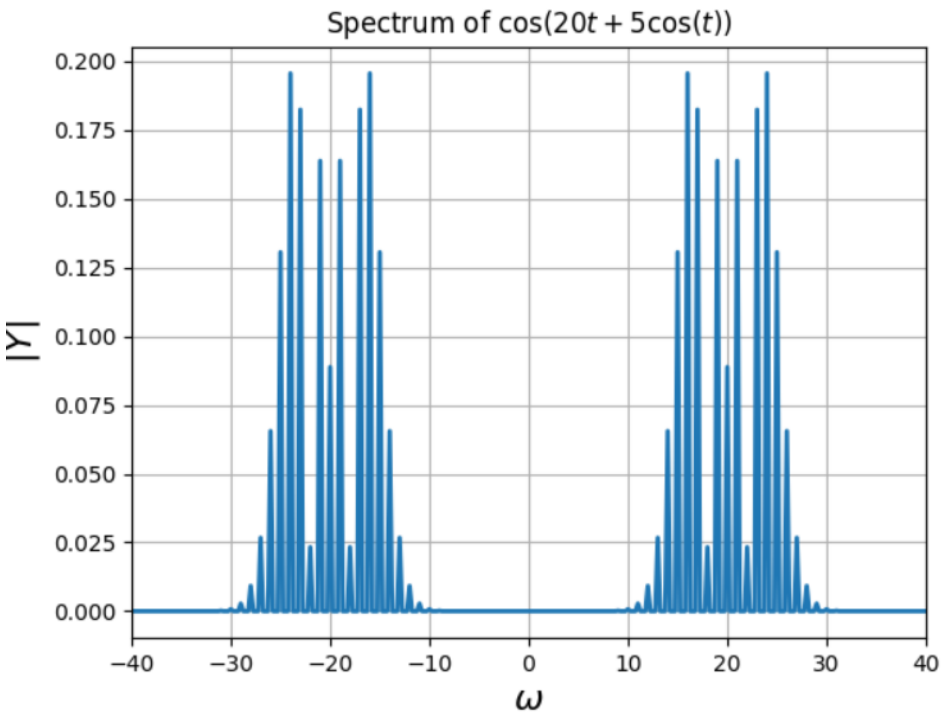
```
w = linspace(-64, 64, 513)[:-1]
```

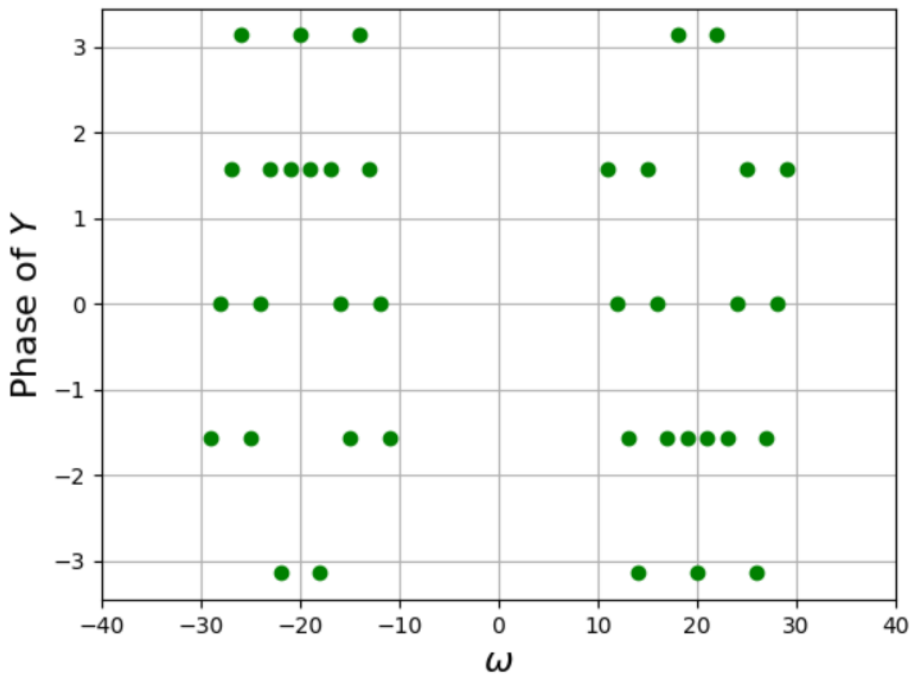
Now to plot the values we write the following code.

```
figure()
plot(w, abs(Y), lw=2)
xlim([-40, 40])
ylabel(r"$|Y|$", size=16)
title(r"Spectrum of $\cos(20t+5\cos(t))$")
grid(True)
show()
ii = where(abs(Y) > 1e-3)
plot(w[ii], angle(Y[ii]), "go", lw=2)
xlim([-40, 40])
```

```
ylabel(r"Phase of $Y$", size=16)
xlabel(r"$\omega$", size=16)
grid(True)
show()
```

We get the following as the spectrum of the given function.





We can see that the carrier frequency is ω=20 and the modulating signal is $5\cos(t)$. So $5cos(t)$ produces infinite side bands but the strength of the side bands reduces the frequency goes away from the f_c(carrier frequency) as we can see from the plot.

We can also see that the phase spectra is a mix of different phases in the range [-π,π].Depending on the phase contributed by $5cos(t)$, the carrier signal($y(t)$) will represent a *sine* or a *cosine*.

0.3.4 Question 4

Now we have to plot the fourier transform of the following gaussian function.

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

The fourier transform of the above $f(t)$ is as follows.

$$Y(\omega) = \frac{1}{\sqrt{2\pi}} e^{\,-\omega^2/2}$$

Now we write the code to generate a timespace that will give error less than 10^{-15} with the actual fourier transform as above.

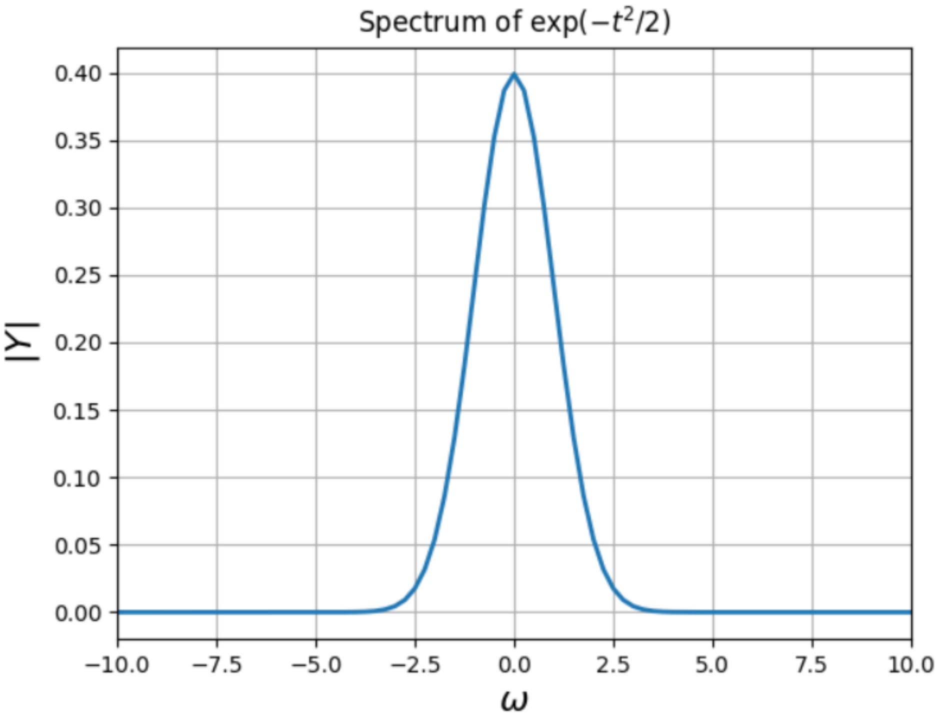
```
T = 2*pi
N = 128
tolerance = 1e-15
error = tolerance+1

while error > tolerance:
    t = linspace(-T/2, T/2, N+1)[:-1]
    w = N/T * linspace(-pi, pi, N+1)[:-1]
    y = exp(-0.5*t**2)

    Y = fftshift(fft(y))*T/(2*pi*N)
    Y_actual = (1/(sqrt(2)*pi))*exp(-0.5*w**2)
    error = mean(abs(abs(Y)-Y_actual))

    T = T*2
    N = N*2
```

Running the above code we get the best window size as 16π. Then we plot the spectrum using the same code we used for plotting the spectrum in questions 1,2 and 3. Then we get the following spectrum.



0.4 Conclusions

- Hence we analysed on how to find DFT for various types of signals and how to estimate normalising factors for Gaussian functions and hence recover the analog Fourier transform using DFT ,also to find parameters like window size and sampling rate by minimizing the error with tolerance upto 10^{-15}.
- We used fast Fourier transform method to compute DFT as it improves the computation time from $\mathcal{O}(n^2)$ to $\mathcal{O}(nlog_2(n))$.
- FFT works well for signals with samples in 2^k, as it divides the samples into even and odd and goes on dividing further to compute the DFT.That is why we take the number of samples in the problems as powers of 2.