Assignment No 8

Subhankar Chakraborty

April 2, 2019

The DFT

The DFT, from a linear algebra perepective is a projection of a vector from the normal vector space to the Fourier Vector space. The expression of the DFT($F[k]$) for a vector of length N is given by

$$F[k] = \sum_{n=0}^{N-1} f[n]e^{-2\pi \frac{nk}{N}j}$$

Similarly the inverse DFT is given by

$$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} F[k]e^{2\pi \frac{nk}{N}j}$$

The N in the denominator comes due to the normalization factor. The DFT can be also seen as a sampled version of the DTFT, ie, a digital version of the analog quantity, which is the DTFT.

The DFT in Python

The numpy module has two inbuilt functions to handle the DFT and the IDFT.

```
numpy.fft.fft()
numpy.fft.ifft()
```

As clear from the names, the first function mentioned takes care of the DFT while the second one takes care of the IDFT.

DFT of $sin(5t)$

This section shows the plotting of the DFT of $sin(5t)$.The DTFT of the function $sin(x)$ $Y(\omega)$ is given by

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

So, for the function $sin(5t)$ we should see two peaks, one at $n = 5$ and the other at $n = -5$ (assume fftshift has already adjusted the ω axis between $[-\pi,\pi)$. The following section of code does this.

```
Y = fftshift(fft(y))/128
w = linspace(-64,63,128)
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
plot(w,angle(Y), 'ro', lw=2)
ii = where(abs(Y)>1e-3)
plot(w[ii], angle(Y[ii]), 'go', lw=2)
```

The part for the angle is to highlight the angles of the samples where we get substantial magnitude.Figure 1 shows the plots.The division by 128 is nothing but a normalization factor.

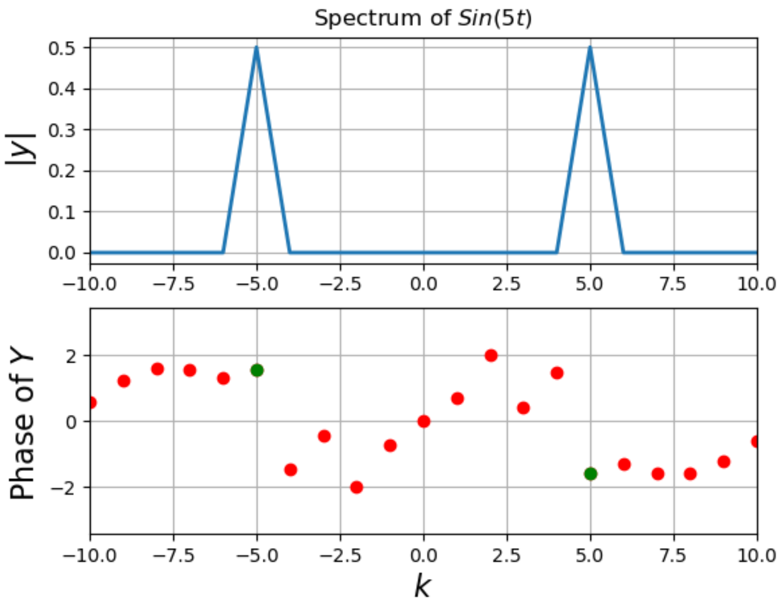


Figure 1: DFT of $Sin(5t)$

As expected, we get two peaks corresponding to n=5 and n=-5. Also we get phases of $\pi/2$ and $-\pi/2$ respectively, corresponding to the totally imaginary DTFT. We also notice only 2 green points showing that only those 2 samples have considerable magnitude.

DFT of f(t)=$(1 + 0.1 * cos(t))(cos10t)$

The function f(t) using the properties of trigonometry can be simplified as

$$f(t) = cos(10t) + 0.05cos(9t) + 0.05cos(11t)$$

So we should expect to see a peak around n=10 and some small peaks around n=9 and 11. The larger peak corresponds to the frequency component of $cos(10t)$ and the smaller ones correspond to $0.05cos(9t)$ and $0.05cos(11t)$ respectively. The code is almost the same as the previous part so is not being written again. Figure 2 shows the plot.

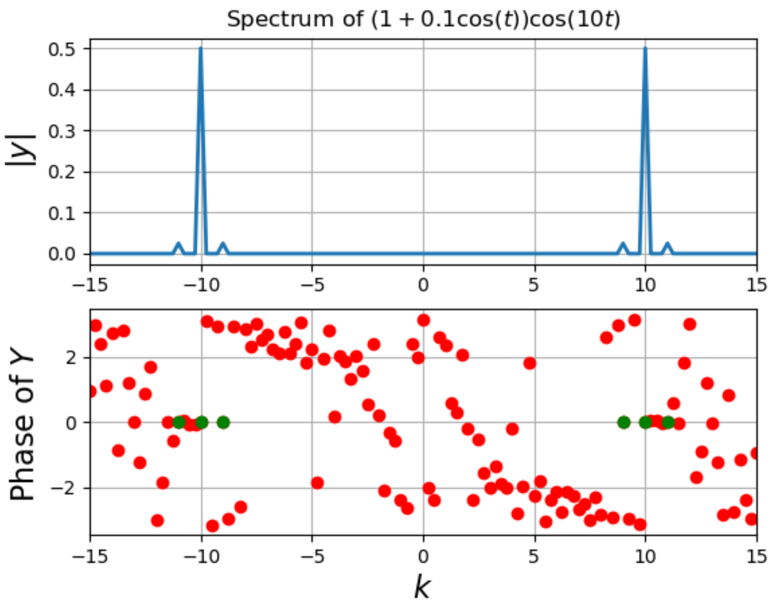


Figure 2: DFT of f(t)

For the phase response, we should expect zero phase especially at the points of magnitude greater than 10^-3 as the DTFT of a cosine is a purely real function.That is what appears in the plots.

DFT of $sin^3(t)$ and $cos^3(t)$

Using the properties of trigonometry, $sin^3(t)$ and $cos^3(t)$ can be simplified as

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

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

So for both of the above functions we should see peaks around n=1 and n=-1 and peaks of one third the magnitude at n=3 and n=-3. For the phase plots, as both terms in the RHS of $cos^3(t)$ give purely real transforms, the phase is consistently zero at points where the magnitude is above 10^{-3}. In the case of$sin^3(t)$, we get the same phase as $sin(t)$ plus more two points phase shifted by π because of the minus sign. This happens in points for n=3 and n=-3. Figure-3 and Figure-4 will make it clear.

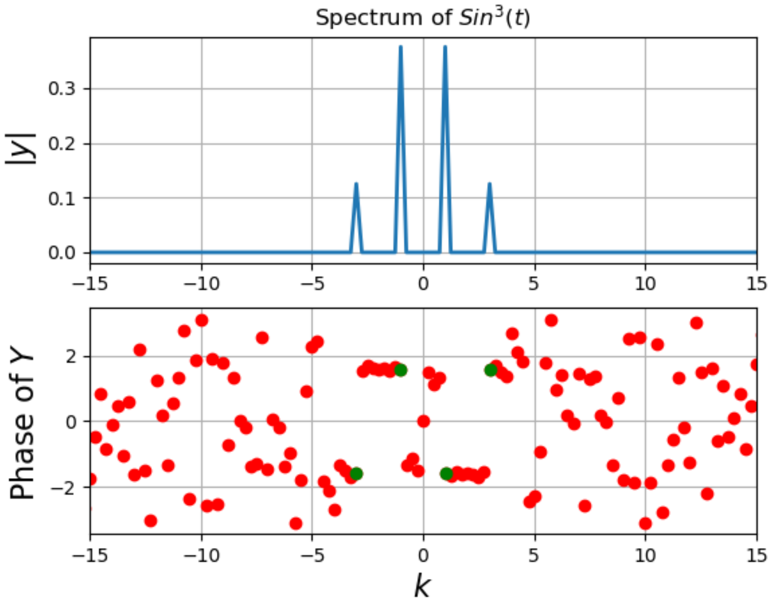


Figure 3: DFT of $sin^3(t)$

DFT of $cos(20t + 5cos(t))$

This function can be best analyzed from the plot itself. Let's analyse the magnitude first.

- The samples are around n=+-20,ie, the overall situation is like a carrier signal of frequency 20 and other signals of smaller frequencies modulated on it.
- The range of values over which the frequencies are spread depends on the magnitude of the cosine inside the cos function. So, we see most of the frequencies are around n 20+5 and 20-5.
- How closely the frequencies are spread depends on the frequency of the cosine inside the cos. Here it is 1. If we increase it, we will see

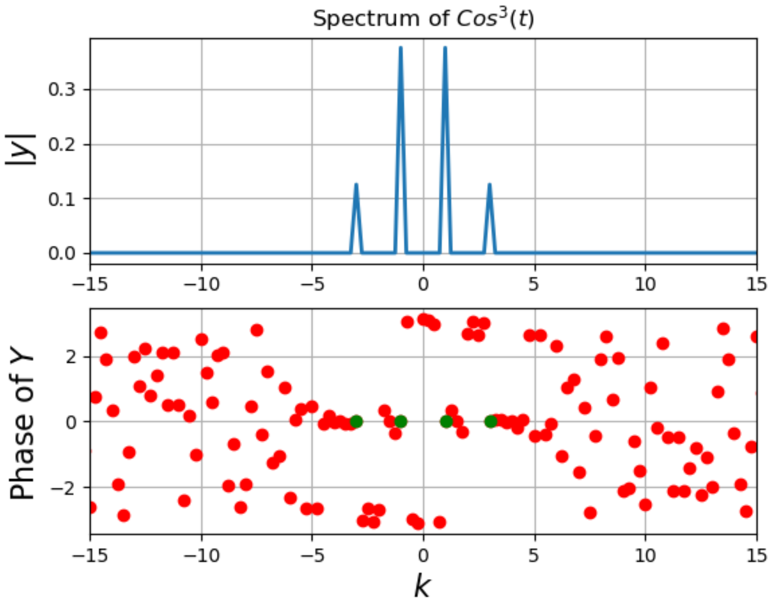


Figure 4: DFT of $cos^3(t)$

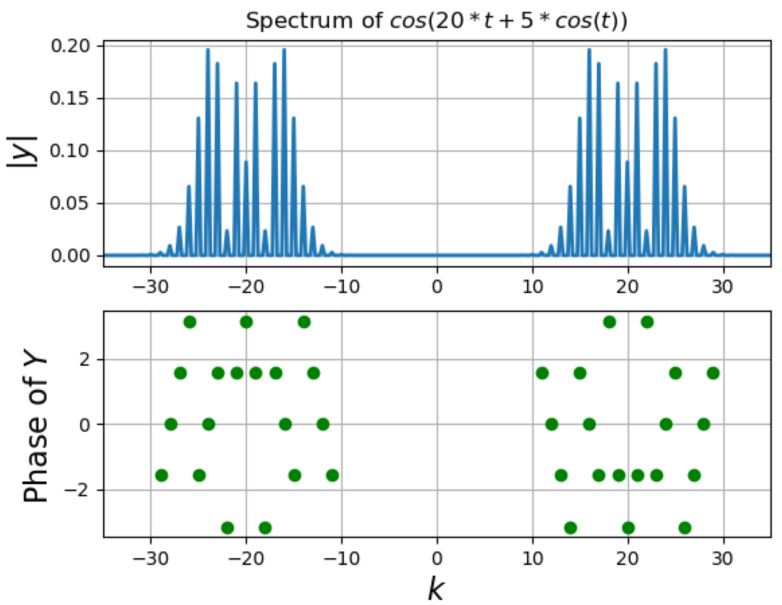


Figure 5: DFT of $cos(20t + 5cos(t))$

much more closely packed values. On decreasing it, we will get spread out values.

For the phase, we see the phase is odd symmetric, as it should be for any real signal. Plus, we see that the phase is symmetric around n=20 and n=-20.

DFT of $e^{-t^2/2}$

For the DFT of the gaussian, the following trends are observed on changing N(the number of samples) and t(the total time period).The plots below show a few examples along with the mentioned values of N and t. The plots show the following trend.

- As we increase t keeping N constant
  - The peak keeps on decreasing as the sampling frequency keeps on decreasing and we have the sampling frequency appear in the DTFT equation and hence, the DFt is scaled accordingly.

    $$X(e^{j\omega}) = X_s(\omega F_s)$$

    where the terms have their usual meaning.

    $$X_s(\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{k=\infty} X_c(\Omega - k\Omega_s)$$

  - The spectrum keeps getting more and more wide. This happens because although the number of points is same, we are sampling over a larger period. That is, we are effectively down-sampling and causing some amount of aliasing.
- As we increase N keeping t constant
  - The peak value remains almost constant.
  - The spectrum keeps getting narrower and narrower as we are introducing additional samples, which in effect is like up-sampling in the time domain.

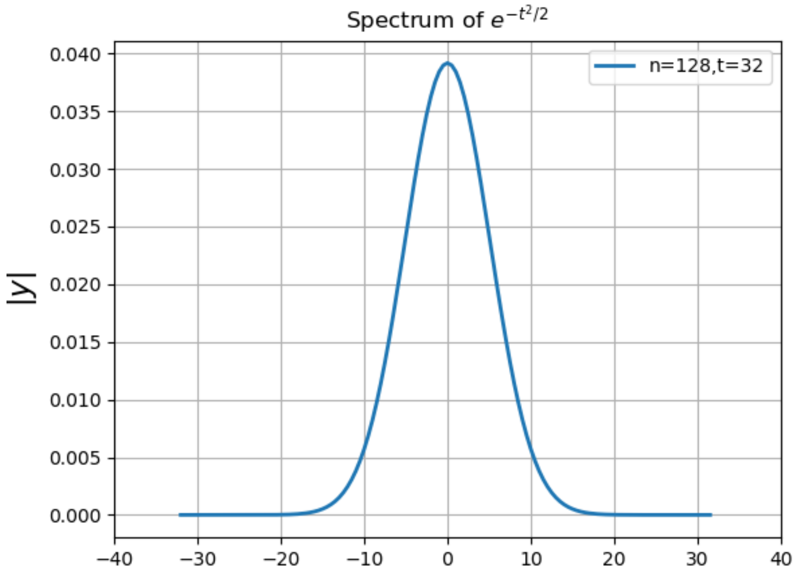
The plots below show a few examples.

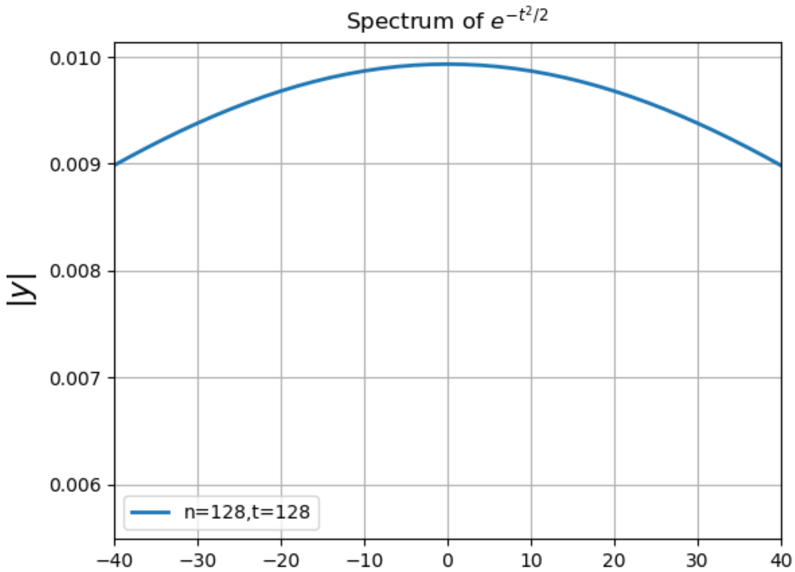
Fine tuning the Error

The DTFT of a gaussian is given by

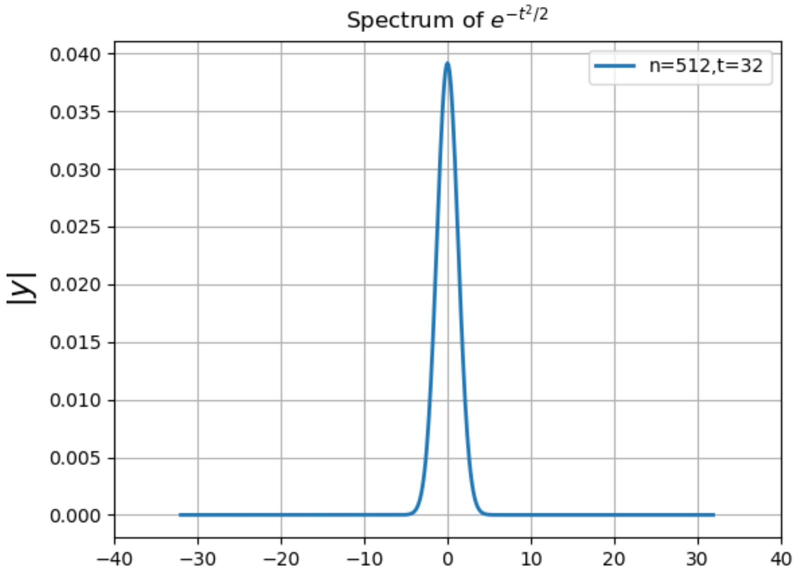
$$G(\omega) = \sqrt{2\pi}e^{-w^2\sigma^2/2}$$

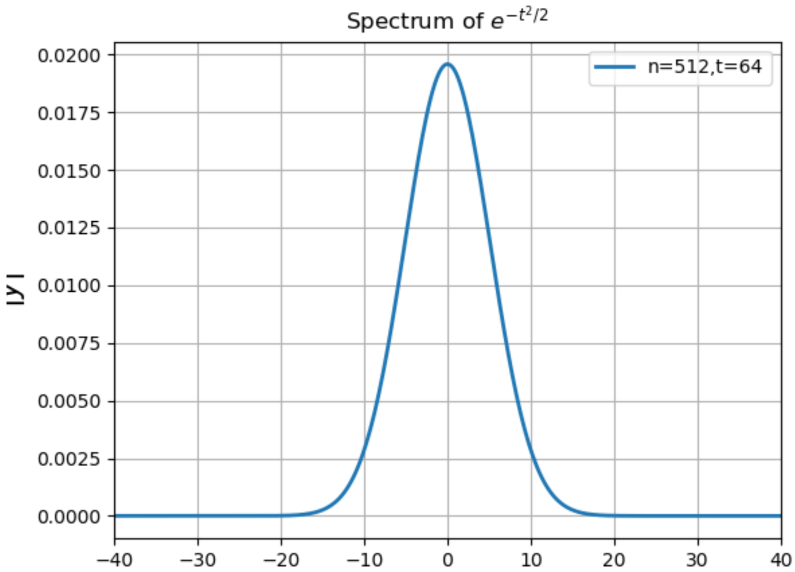
After trying out different values of N and t, and scaling the two graphs accordingly, the best curve was obtained for N=4097 and t = 80. We get an



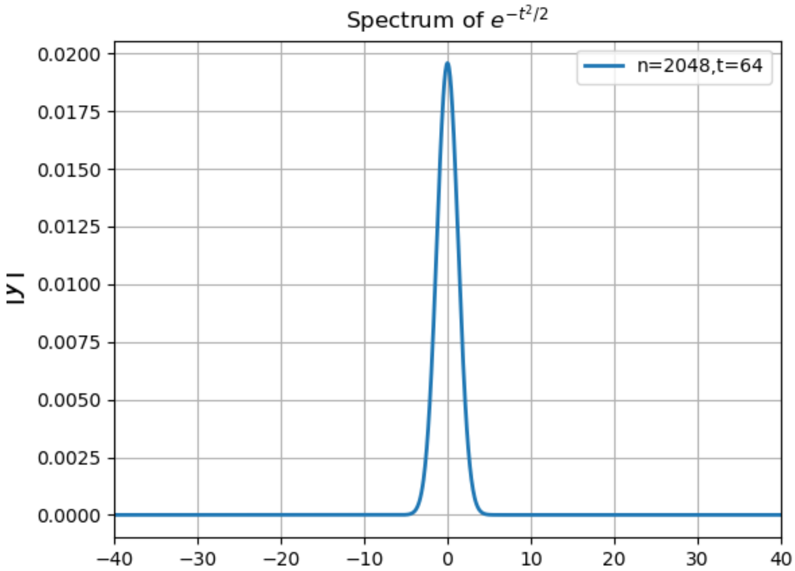


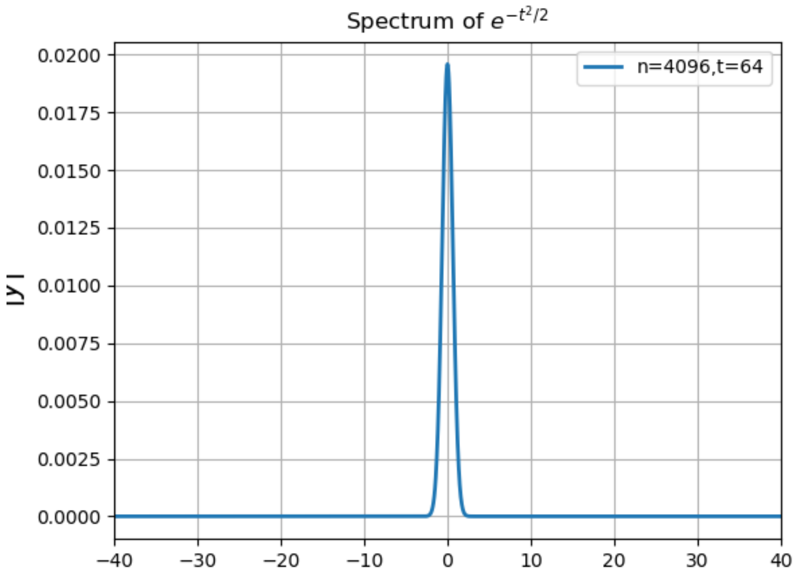
error of the order 0.6^{-5} for this pair. One thing to be noted is that the curve obtained directly from calculations from the formula was scaled accordingly to take account for all the constants that occur due to sampling and taking the DFT. The code below does that.






```
t = linspace(-x,x,N)
t = t[:-1]
y=exp(-(t**2)/2)
Y = fftshift(fft(y))/(N-1)
w = linspace(-x,x,N)
```






```
w = w[:-1]
plot(w,abs(Y),'k',lw=2)
actual_gaussian = np.sqrt(2*np.pi)*np.exp(-(w**2)/2)
scaling_factor = np.max(abs(actual_gaussian))/np.max(abs(Y))
plt.plot(w,actual_gaussian/scaling_factor,'y',alpha = 0.5, linewidth = 4)
```

\\

The plot below shows the graphs.

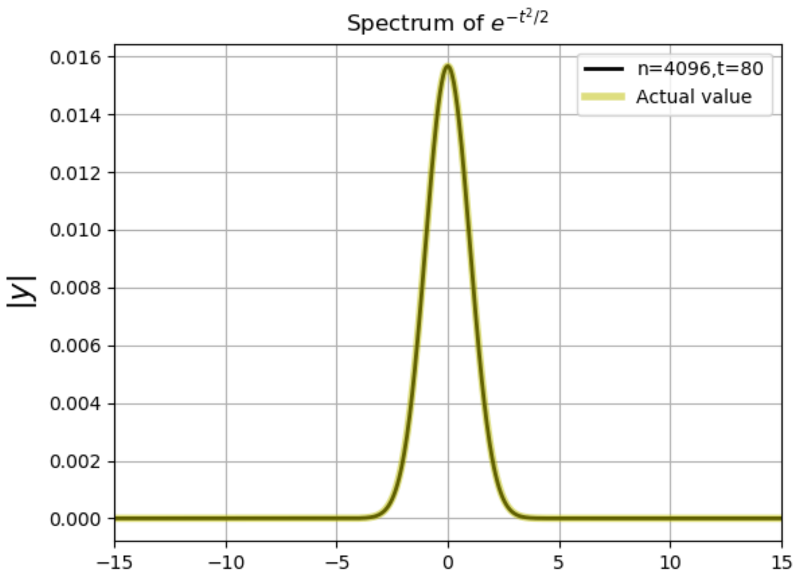


Figure 6: Best fit with the ideal DTFT.