Assignment-9 : Discrete Fourier Transform

Sai Gautham Ravipati - EE19B053

May 24, 2021

Contents

1 Introduction

Fast Fourier Transform (FFT) provides a faster way of implementation of Discrete Fourier Transform (DFT). The same is implemented using numpy.fft() module, which is imported to local name-space using pylab. Throughout the assignment the same has been used to find the CTFT's of certain periodic functions and at the end CTFT of non band-limited Gaussian signal is computed, by varying several properties like number of points sampled, the length of the time-spectrum considered etc. The rest of the analysis follows.

2 Fast Fourier Transform

Given a Continuous time signal, DFT is a sampled version of the DTFT, which is the digital version of the Continuous Fourier Transform. The number of samples of DFT are same as that of number of samples of continuous signal being considered. The FFT algorithm improves the speed with which DFT is computed from $O(N^2)$ to $O(NlogN)$, where N is the size of the data. Module $numpy.fft()$ is used to compute the DFT and $numpy.ifft()$ is used to compute the inverse of the same, both loaded using pylab to the local name-space. The following lines of code computes FFT and inverse FFT, for 128 random samples, and inverts them back.

1. Input Code :

```
x = rand(128)
X = fft(x)
y = ifft(X)
c_[x,y]
print('The error after inverting back is {:.2e}'.format(abs(x-y).max()))
print('Original and Inverted samples:')
print(c_[x,y])
```

2. Output :

```
The error after inverting back is 3.37e-16
Original and Inverted samples:
[[0.87601469+0.00000000e+00j 0.87601469+5.55111512e-17j]
 [0.56018187+0.00000000e+00j 0.56018187-1.73472348e-17j]
 ...
 [0.9104371 +0.00000000e+00j 0.9104371 -1.38777878e-17j]
 [0.59068279+0.00000000e+00j 0.59068279-2.08166817e-17j]]
```

Clearly the inverted values have finite imaginary values due to the CPU precision limitation. The maximum absolute error is of the order 1e-16.

3 Spectrum of $Sin(5t)$

The function $Sin(5t)$ can be expressed as the difference of complex sinusoids as follows :

$$sin(5t) = \frac{e^{j5t} - e^{-j5t}}{2j} \tag{1}$$

By taking the Continuous time Fourier transform, the spectrum is defined by,

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

So we can expect to two impulses located as $w = 1, -1$ with magnitude 0.5 and phase -$\frac{\pi}{2}$ and $\frac{\pi}{2}$ respectively. Using the following lines of code, the spectrum shown in Fig 1 has been plotted.

```
t_1a = linspace(0,2*pi,128)              #Generating 128 points in 0 to 2*pi
Y_1a = fft(y_1a(t_1a))                   #Using fft to find the fourier transform
figure()                                 #Generates new figure
subplot(2,1,1)                           #Sub-plot to plot the magnitude spectrum
plot(abs(Y_1a),lw=2)                     #Magnitude plot
grid(True)                               #Setting grid
title(r'Spectrum of $sin(5t)$')          #Phase plot
ylabel(r'$|Y|$', size = 12)              #Setting ylabel
subplot(2,1,2)                           #Sub-plot to plot the phase spectrum
ylabel(r'Phase of Y', size = 12)         #Setting ylabel
xlabel(r'$k$',size = 12)                 #Setting xlabel
plot(unwrap(angle(Y_1a)),lw=2)           #Phase plot
grid(True)                               #Setting grid
savefig('EE2703_ASN9/ex1.jpg')           #Saving the plot at file_location
```

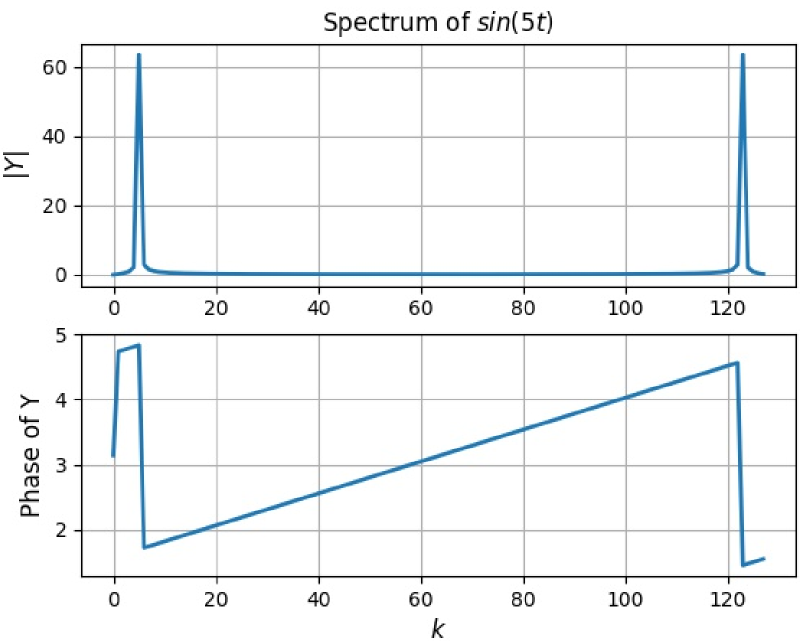

Figure 1: Spectrum of $Sin(5t)$

Clearly, there are certain deviations in the spectrum from what is expected, and some of them could be improved by certain means :

1. The spikes have a height of 64, not 0.5. We should divide by N (128) to use it as a spectrum.

2. The peaks are not at -1,1 this is because Y(0) is the first element of the array returned by fft(), fftshift() is used to make the spectrum symmetric with Y(0) in the middle and positive and negative components on the either side of Y(0).

3. The frequency is not in place and this can be corrected by removing the last point 2π as it is identical to 0, apart from using fftshift().

4. The actual phase at the spikes is not truly what was expected, although there is a difference of π.

To resolve these issues a function call has been defined, which has been presented below.

```python
def func_spectrum((low_t, up_t, N, y, y_ttl, file_name, x_range, flag):
    """
    This function generates spectrum plots for various input functions and then
    it saves them in the directory EE2703_ASN9. The inputs to the function are:

    low_t     : Lower limit of t, for which data-points are calculated
    up_t      : Upper limit of t, for which data-points are calculated
    N         : Number of data-points for which the function is calculated
    y         : Name of the function which has t as argument
    y_ttl     : Title to be added to the plot : 'Spectrum of <y_ttl>'
    file_name : Output file name of the image
    x_range   : Range of x datapoints to be displayed in the plot
    flag      : 1 - Plots only those points having magnitude > 1e-3
                    in phase spectrum
                2 - Plots all phase points in red with point mentioned
                    in 1 as green dots
                else - Plots all the phases points in red

    Output - None
    The plots shall be saved as EE2703_ASN9/<file_name>
    """
    t = linspace(low_t,up_t, N + 1)
    t = t[:-1]
    T = up_t - low_t
    w = linspace(-pi,pi, N + 1)*N/T
    w = w[:-1]

    figure()
    subplot(2,1,1)

    if (y_ttl == '$e^{-t^2/2}$'):
        <Shall be analysed later along with Gaussian Function>
    else:
        Y = fftshift(fft(y(t)))/N
        plot(w, abs(Y), lw = 2)

    xlim(x_range)
    ylabel(r"$|Y|$", size = 12)
    title(r"Spectrum of {}".format(y_ttl))
    grid(True)

    subplot(2,1,2)

    if flag == 1:
        ii = where(abs(Y) > 1e-3)
        plot(w[ii], angle(Y[ii]),'go',lw=2,markersize = 5)
    elif flag == 2:
        plot(w,angle(Y),'ro',lw=2)
        ii = where(abs(Y) > 1e-3)
        plot(w[ii], angle(Y[ii]),'go',lw=2,markersize = 5)
    else:
        plot(w,angle(Y),'ro',lw=2)

    xlim(x_range)
```

```
55     ylim([-4,4])
56     ylabel(r"Phase of $Y$", size = 12)
57     xlabel(r"$\omega$", size = 12)
58     grid(True)
59     savefig('EE2703_ASN9/' + file_name)
60 
61     return 0
```

Lines 22,23 does the work of removing the final 2π, line 34 performs the fftshift() and scaling the input by N which is the number of data-points being considered. The following plot is obtained when, $t \in [0, 2\pi), w \in [-64, 64)$, with 128 points being considered. Here phase of all

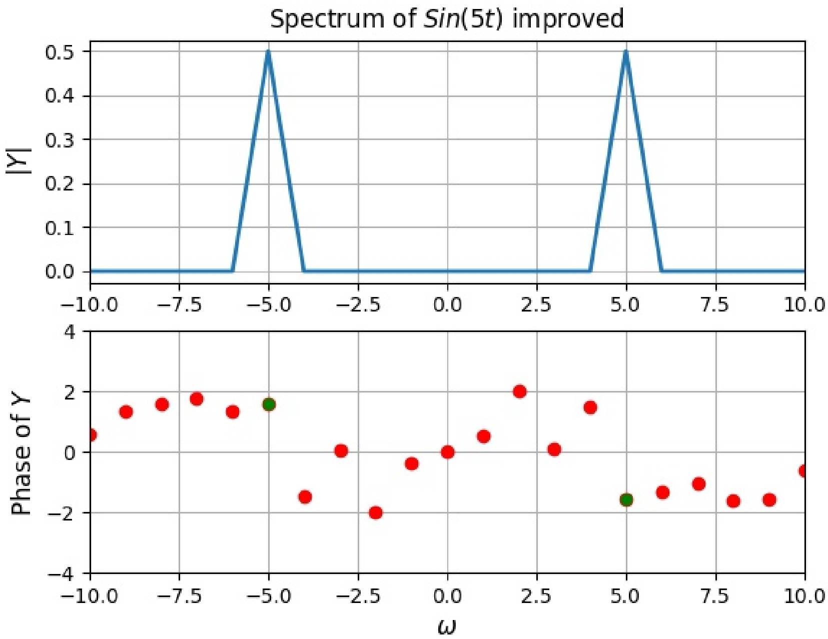

Figure 2: Improved Spectrum of $Sin(5t)$

the points has been plotted, with phase of those points with magnitude > 1e-3 in green. This can be done by using flag as 2. The points other then those in green, are insignificant as their magnitude is close to zero and such points are ignored in rest of the analysis and only green points are considered. This is done by using flag as 1. Clearly by making these modifications, the spectrum is much better then the previous one and the peaks are at the expected locations.

4 Spectrum for Amplitude Modulated Signal

Consider the signal $(1+0.1cos(t))cos(10t)$, the signal $cos(10t)$ has impulses at $w = \pm 10$, while $cos(t)cos(10t)$ leads to impulses at $w = \pm 9, \pm 11$. The transform of the above function is given by :

$$Y(\omega) = \frac{1}{2}[\delta(w-10) + \delta(w+10)] + \frac{1}{40}[\delta(w-11) + \delta(w+11) + \delta(w-9) + \delta(w+9)] \quad (3)$$

Thus we expect four smaller peaks and 2 larger peaks in the spectrum. The following plot in Fig. 3 is obtained for $t \in [0, 2\pi)$ and 128 sample points being considered.

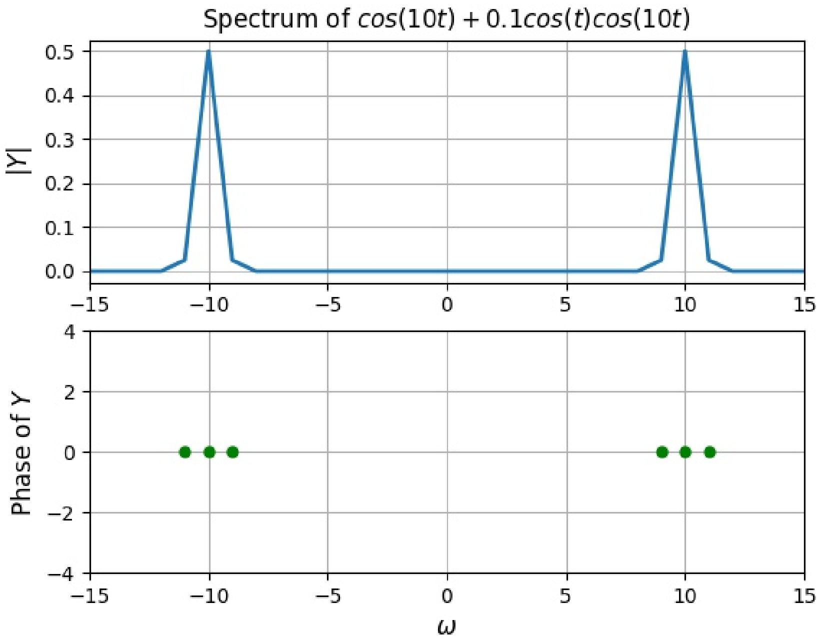

Figure 3: Spectrum of AM signal

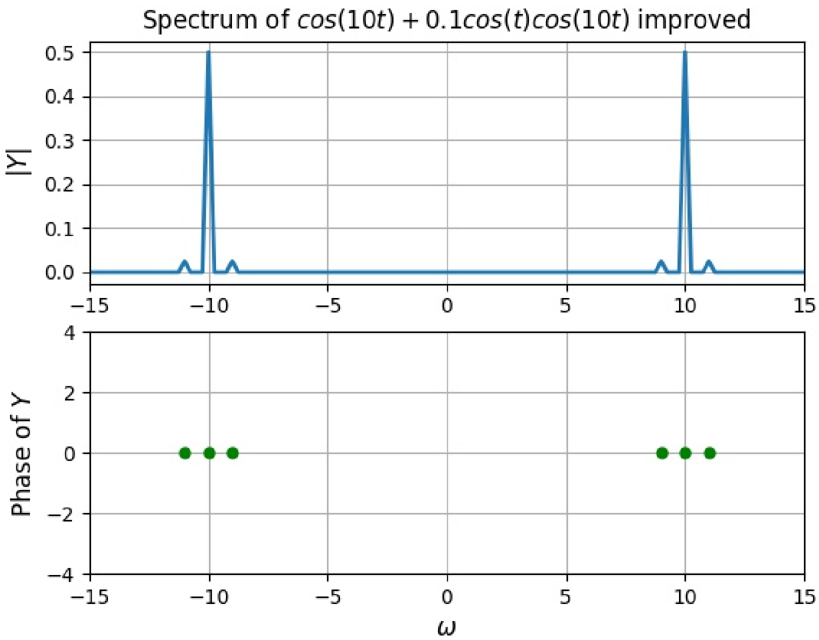

Figure 4: Improved Spectrum of AM signal

Clearly the phase spectrum indicates 3 non-zero values in the magnitude spectrum around 10 and -10. But there is an overlap between the peaks at these locations, as a result of which impulses at $w = \pm 9, 11$ are not clearly visible. The phase is consistent and is zero at these locations. To improve the magnitude spectrum, number of points sampled is increased although it gives same spacing, then t axis is stretched which provides tighter spacing between the frequency samples. The spectrum obtained for $t \in [-4\pi, 4\pi)$ and 512 sample points being considered is presented in Fig 4. The 6 impulses are clearly visible in the plot, each having a phase of 0.

5 Spectrum of $cos^3(t)$

The function $cos^3(t)$ can be expressed as the sum of $cos(t)$ and $cos(3t)$ terms. The expression is given by:

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

The transform for the same is given by :

$$Y(\omega) = \frac{1}{8}[\delta(w-3) + \delta(w+3)] + \frac{3}{8}[\delta(w-1) + \delta(w+1)] \tag{5}$$

So we expect 2 impulse pairs all having phase zero, and magnitude of those at $w = \pm 3$ being one-third that of those at $w = \pm 1$. The spectrum obtained for $t \in [-4\pi, 4\pi)$ and 512 sample points being considered is presented in Fig 5. The magnitude and phase plots are consistent with those expected, where all the impulses have phase 0, and magnitudes mentioned previously. Since $cos^3(t)$ is real and even, the spectrum is real and even.

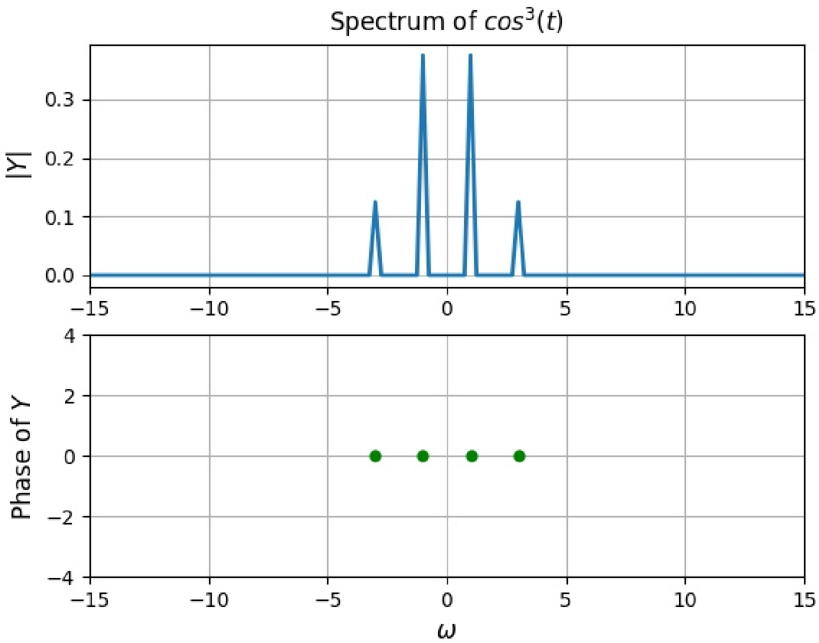

Figure 5: Spectrum of $cos^3(t)$

6 Spectrum of $sin^3(t)$

Similar to previous case, function $sin^3(t)$ can be expressed as the difference of $sin(t)$ and $sin(3t)$ terms. The expression is given by:

$$sin^3(t) = \frac{-sin(3t) + 3sin(t)}{4} \tag{6}$$

The transform for the same is given by :

$$Y(\omega) = \frac{1}{8j}[\delta(w+3) - \delta(w-3)] + \frac{3}{8j}[\delta(w-1) - \delta(w+1)] \tag{7}$$

So we expect 2 impulse pairs all having phases $\pm\ \frac{\pi}{2}$ alternatively, and magnitude of those at $w = \pm 3$ being one-third that of those at $w = \pm 1$. The spectrum obtained for $t \in [-4\pi, 4\pi)$ and 512 sample points being considered is presented in Fig 6. The magnitude and phase plots are consistent with those expected. Since $sin^3(t)$ is real and odd, the spectrum is imaginary and odd.

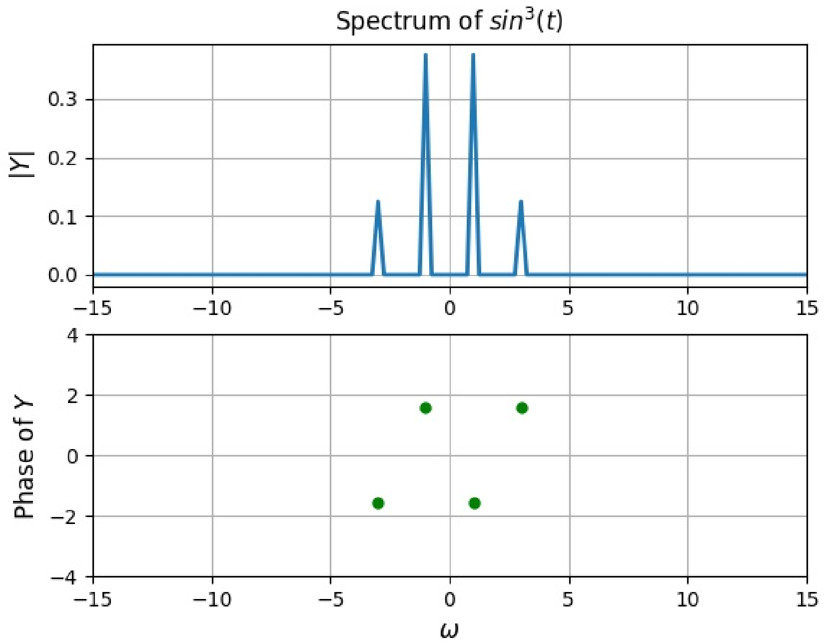

Figure 6: Spectrum of $sin^3(t)$

7 Spectrum of Frequency modulated signal

The function $cos(20t + 5cos(t))$ represents the case of a frequency modulated signal. The function can be expressed as below in terms of Bessel functions of various orders, using Chowning's version of the expansion.

$$cos(20t + 5cos(t)) = \sum_{k=-\infty}^{+\infty} J_k(5)[cos((20+k)t + \frac{k\pi}{2})] \tag{8}$$

In the expression, $J_k(x)$ is Bessel function of order k. One particular relation that is of use is given below.

$$J_{-k}(x) = (-1)^k J_k(x) \tag{9}$$

Because of the non-zero Bessel coefficients of the various frequency components, the frequency domain spectrum shall be more spread than that of amplitude modulation, with impulses around $w = \pm 20$. So the side-bands around $w = \pm 20$ also carry significant energy, unlike the case of AM. Because of equation 9, the magnitudes of impulses at $w = \pm 20 \pm k$ are the same. The structure of the phase spectrum also follows from the above equations. The spectrum obtained for $t \in [-4\pi, 4\pi)$ and 512 sample points being considered is presented in Fig 7.

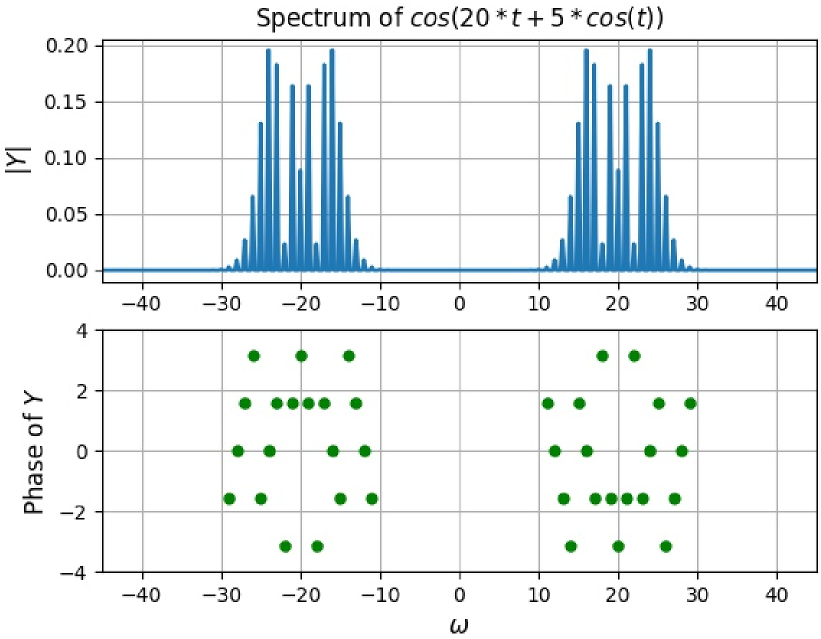

Figure 7: Spectrum of FM signal

8 CTFT of Gaussian Exponential

The Gaussian exponential, belongs to a special class of functions with self-similar transform. The transform of the Gaussian exponential $e^{\frac{-t^2}{2}}$ is given by :

$$Y(\omega) = \sqrt{2\pi}e^{\frac{-\omega^2}{2}} \tag{10}$$

The Fourier transform of a function y(t) is evaluated using the following equation :

$$Y(w) = \int_{-\infty}^{\infty} y(t)e^{-jt\omega}dt \tag{11}$$

For functions like the Gaussian and those decay with higher values of $|t|$, the integral can be approximated by considering only a period of length T, such that $t \in [-\frac{T}{2}, \frac{T}{2}]$. Suppose T is split into N intervals of length dt, then the integral can be approximated by the Riemann sum. Making the above approximations give the following set of equations.

$$Y(w) \approx \int_{\frac{-T}{2}}^{\frac{T}{2}} y(t)e^{-jt\omega}dt \tag{12}$$

$$Y(w) \approx \frac{T}{N} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} y(ndt)e^{-j\omega ndt} \tag{13}$$

The summation term in equation 13, is pretty much similar to the DFT of the discrete samples being considered. So CTFT of the Gaussian can be approximated as,

$$Y(w) = \frac{T}{N} * DFT[y(ndt)] \tag{14}$$

The following lines of code (block in $func - spectrum$ which was left previously for analysis), implements equation 14 and prints out the maximum error when compared to the case of analytical expression.

```
Y   = fft(ifftshift(y(t)))
Y   = fftshift(Y)/N
Y   = Y*T
Y_t = exp(-w**2/2)*sqrt(2*pi)
print("The maximum error ... is {:.3e}".format(N,low_t,up_t,max(abs(Y_t - Y))))
```

The following lines of code estimate the value of T, for a given value of N.

```
N = 256
T = 2*pi
thresh = 1e-6
err = 100

for i in range(15):
    t = np.linspace(-T/2,T/2,N+1)
    t = t[:-1]
    w = np.linspace(-N*pi/T,N*pi/T,N+1)
    w = w[:-1]
    Y   = fft(ifftshift(y_4(t)))
    Y   = fftshift(Y)/N
    Y   = Y*T
    Y_t = exp(-w**2/2)*sqrt(2*pi)
    error = max(abs(Y - Y_t))
    if error < thresh :
        T_est = T
        err   = error
        break
    T += 2*pi

if err != 100 :
    print('\nThe estimeted value of T is {:.3f}'.format(T_est))
    print('The value of error ... is {:.3e}'.format(N,T_est,err))
else :
    print("Failed to Converge")
```

The above code for a given value of N, iterates on T starting from 2π, increments it by 2π every turn until 15 turns or until error goes below 1e-6 which ever is reached first. Through this way for a given N, the value of T is found for which the approximation is valid till the 6th decimal. This keeps track of the variable err, if it is not 100, meaning error goes below 1e-6, or else it didn't converge and the same has be run for more number of iterations. Running the above code for different values of N, gave the following results, and the estimated value of T turned out to be 4π in all the cases, i.e. $t \in [-2\pi, 2\pi)$. The results of the run for $N : [128, 256, 512, 1024]$ are :

1. The value of error between true and obtained spectrum for N:128, T:12.56 is 8.583e-10

2. The value of error between true and obtained spectrum for N:256, T:12.56 is 8.382e-10

3. The value of error between true and obtained spectrum for N:512, T:12.56 is 8.331e-10
4. The value of error between true and obtained spectrum for N:1024, T:12.56 is 8.319e-10

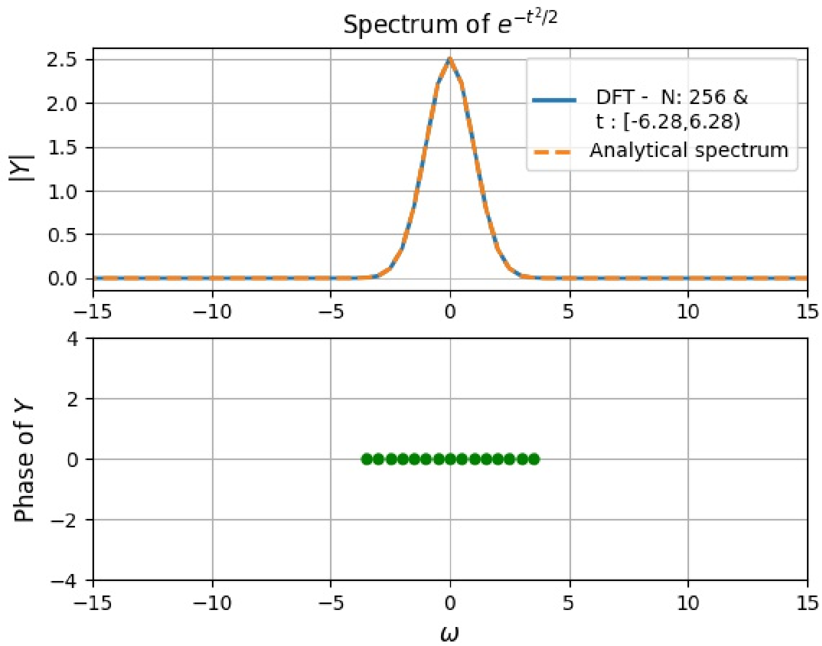

Figure 8: Spectrum of Gaussian Exponential

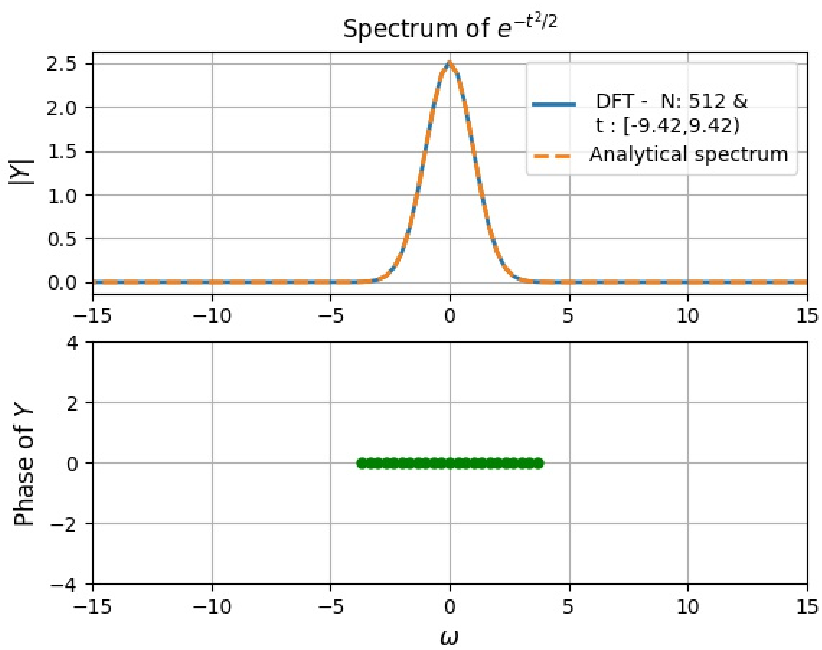

Figure 9: Spectrum of Gaussian Exponential Improved

Based on the previous results, the value of N is selected as 256 and the value of T as 4π. The plots of the spectrum obtained have been presented in Fig. 8. The error for this case is

8.382e-10. To improve the spectrum further N has been increased to 512, and the threshold value is taken as 1e-10 instead of 1e-6. This gave a better plot where T was estimated as 6π, and the error is 6.439e-15. The same has been presented in Fig. 9.

9 Conclusion

Fast Fourier Transform provides a faster method of computing the DFT of several periodic functions. In case of sinusoidal signals and AM signal, the spectrum had impulses. In case of FM signal the side-band was more spread when compared to the case of AM signal thus indicating significant energy being carried in the side-bands as well. DFT can be used to find the Fourier transform of signals like Gaussian exponential by exploiting the fact that these signals decay to zero, and considering finite period of the signal is sufficient to provide good estimate of CTFT. The error in CTFT estimated depends on number of points being considered and sample period, when chosen appropriately gave a fair estimate. Thus the fft module is very much useful to compute DFT of signals, which is a sampled version of DTFT.