EE2703: Applied Programming Lab
Assignment 8
Discrete Fourier Transform

Arjun Menon Vadakkeveedu
EE18B104
Electrical Engineering, IIT Madras

April 20, 2020

Contents

1 Introduction

This program uses numpy's fft module to compute the Discrete Fourier Transform of some common signals (sinusoids, AM signal, FM signal and Gaussian). The Discrete Fourier Transform of a discrete time sequence x[n] is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-jk\frac{2\pi}{N}n}$$

and the inverse DFT is obtained by:

$$x[n] = \sum_{n=0}^{N-1} X[k]e^{jk\frac{2\pi}{N}n}$$

The DFT can be interpreted as a sampling of the Discrete Time Fourier Transform (DTFT). As a result, the corresponding time domain sequence is periodic with period N.
Time and Frequency Bins: The time vector is sampled for a total of c cycles (each cycle is of duration 2π) and N samples are recorded. This means that the time vector is (defined as a non-causal signal in time) linspace(-cπ, cπ, N+1)[:-1]

Similarly, the frequency vector has length N and corresponds to [-π, π) in the Discrete Frequency domain and [-$F_s/2$, $F_s/2$) in the Continuous Frequency Domain- where F_s is the sampling frequency.

Mapping the DFT to the Continuous Time Fourier Transform: While the DFT can accurately determine the spectrum of bandlimited periodic continuous time signals sampled at a rate greater than the Nyquist rate, the DFT of a Gaussian will have errors due to its infinite band. Hence, by taking sufficiently large values of N and F_s, the CTFT of a Gaussian may be approximated using the DFT:

$$X(j\Omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\Omega t}dt$$

The integral may be approximated to a finite sum as (for sufficiently large N):

$$X(j\Omega) \approx \sum_{n=-N/2}^{N/2-1} x(n\Delta t)e^{-j(\Omega\Delta t)n}(\Delta t)$$

$$= (\Delta t)X[k]$$

$$= N(\Delta t) * Spectrum(x) = 2\pi c * Spectrum(x)$$

---

2 Approach

The DFT is computed using the FFT algorithm, implemented using numpy.fft module. To centre the fft vector about the 0-frequency bin, the vector is re-arranged using the fftshift function.

2.1 Important blocks used in the program

1: Generating FFT

```
def gen_fft(f, num_cycles, sample_freq, ii_cutoff):
  N = num_cycles*sample_freq
  x = np.linspace(-1*pi, pi, N+1)*num_cycles; x = x[:-1]
  y = f(x)
  y = ifftshift(y)
  Y = fftshift(fft(y))/N
  w = np.linspace(-0.5, 0.5, N+1)*sample_freq; w = w[:-1]
  ii = np.where(abs(Y)>ii_cutoff)
  return Y, w, ii
```

2: Plotting DFT characteristics:

```
def gen_plots(Y, w, ii, x_range, title, plot_gauss = False):
  fig = plt.figure()
  plt.subplot(2,1,1)
  plt.xlim(x_range)
  plt.plot(w, abs(Y), lw = 2)
  if plot_gauss == True:
    plt.plot(w, gauss_f(w))
  plt.ylabel(r"|Y|", size = 16)
  plt.grid(True)
  plt.title(title)
  plt.subplot(2,1,2)
  plt.xlim(x_range)
  plt.plot(w[ii], (np.angle(Y[ii])), 'ro', lw = 2)
```

```
  plt.ylabel(r"Phase of Y")
  plt.grid(True)
  plt.tight_layout()
  plt.show()
  if plot_gauss == True:
    plt.xlim(x_range)
    plt.plot(w, np.real(Y - gauss_f(w)))
    plt.title("Error of DFT with true Fourier Transform")
    plt.show()
  return 0
```

3: Input Format:

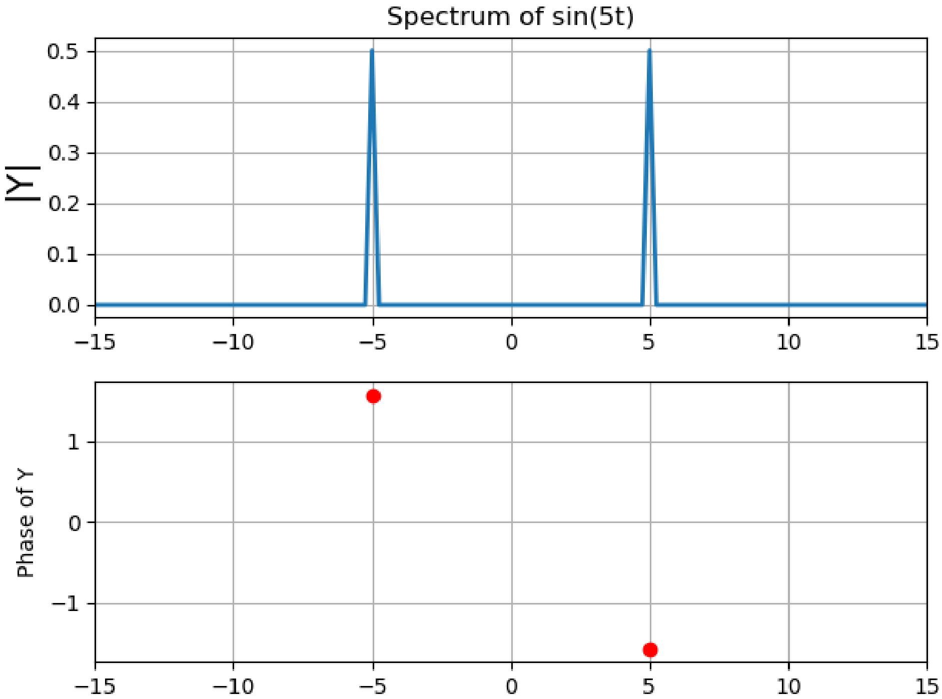
```
python3 EE18B104_Assign8.py --h
usage: EE18B104_Assign8.py [-h] [--fn_choice FN_CHOICE]

optional arguments:
  -h, --help            show this help message and exit
  --fn_choice FN_CHOICE
                        Choose time domain function:
                        1: sin(5t)
                        2: (1+0.1cos(t))cos(10t) (AMPLITUDE MODULATION)
                        3: sin^3(t), cos^3(t)
                        4: cos(20t + 5cos(t)) (FREQUENCY MODULATION)
                        5: exp(-(t^2)/2) (GAUSSIAN)
```

---

3 FFT Analysis

3.1 sin(5t)



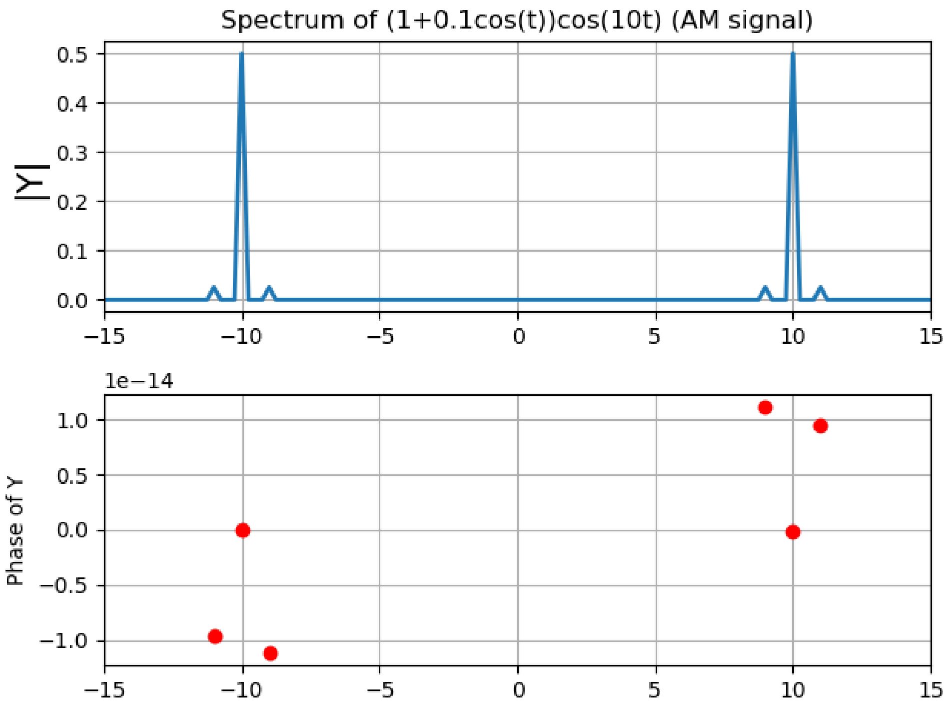
As can be seen in the figure, sin(5t) comprises of two deltas at -5 and +5 with phase $+\pi/2$ and $-\pi/2$ respectively

3.2 Ampltiude Modulation

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

$$y(t) = cos(10t) + 0.05 * (cos(11t) + cos(9t))$$

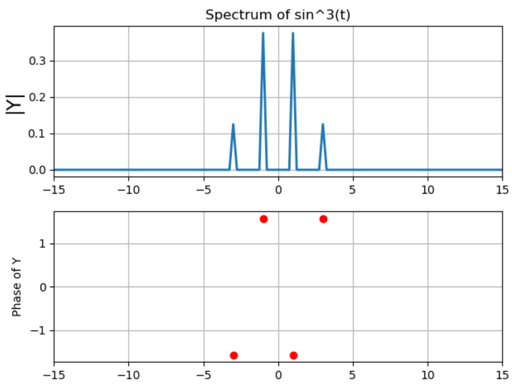
Thus, the AM signal is expected to have frequency components at ± 9, ± 10, ± 11 rad/s.



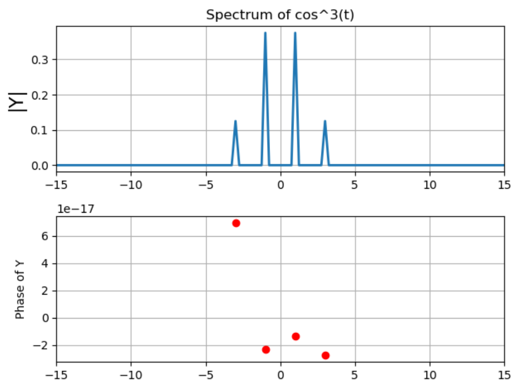
Since the signal is composed of cosines, the phase is 0 (with some finite errors due to the implementation of fft)

3.3 $\sin^3(t)$ and $\cos^3(t)$

$$sin^3(t) = 0.75sin(t) - 0.25sin(3t); cos^3(t) = 0.75cos(t) + 0.25cos(3t)$$



(a) $\sin^3$(t)



(b) $\cos^3$(t)

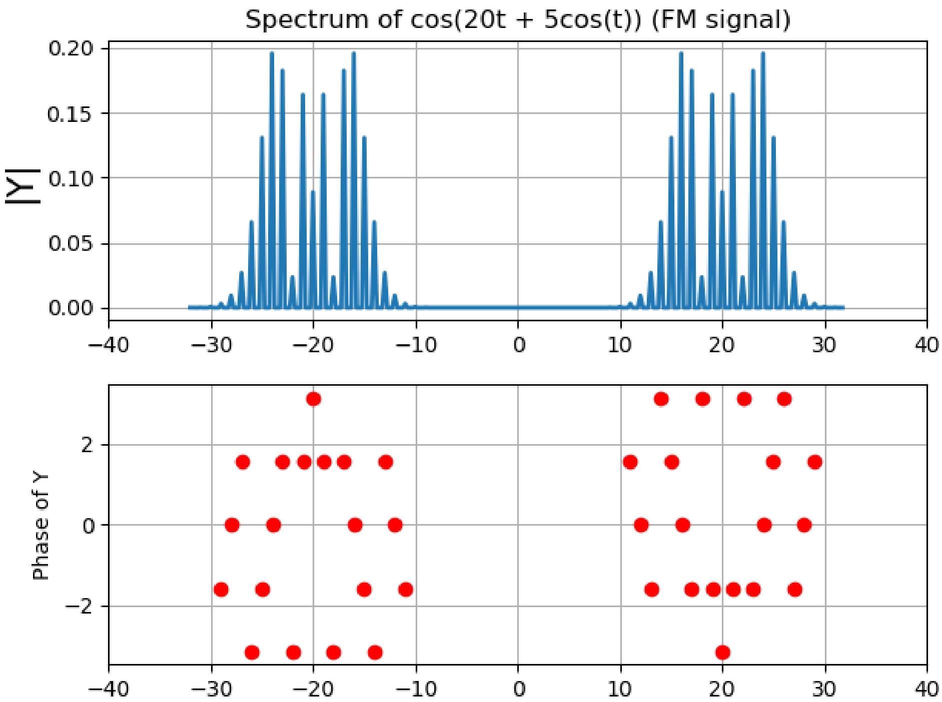
3.4 Frequency Modulation

$cos(20t + 5cos(t)) = cos(20t)cos(5cos(t)) - sin(20t)sin(5cos(t)) = \Re(e^{j20t}e^{j5cos(t)})$

The Fourier Series coefficients of $(e^{j5cos(t)}$ are given by the Jacobi-Anger expansion:

$$e^{j5cos(t)} = \sum_{n=-\infty}^{+\infty} (j^n)J_n(5)e^{jnt}$$

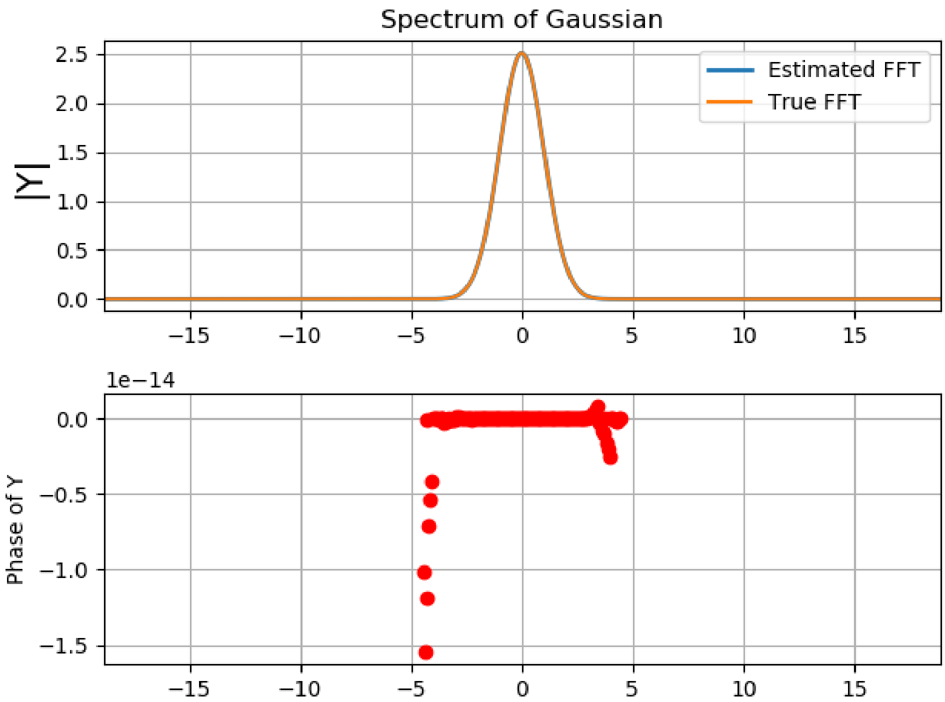
where $J_n(5)$ is the n-th order Bessel function of the first kind evaluated at 5. Since the DFT of e^{j20t} are impulses, and since the DFT of the signal is the convolution of the DFTs of the Bessel impulse train with impulses centred at ± 20.
This is clearly visible from the magnitude plot of the DFT. As for the phase, note that $J_n(5)$ varies as $(-1)^n$. About ω =20, the phase of the DFT varies as $(-j)^{mod(n-20,20)}$ (and similarly for $\omega = -20$)



3.5 Gaussian

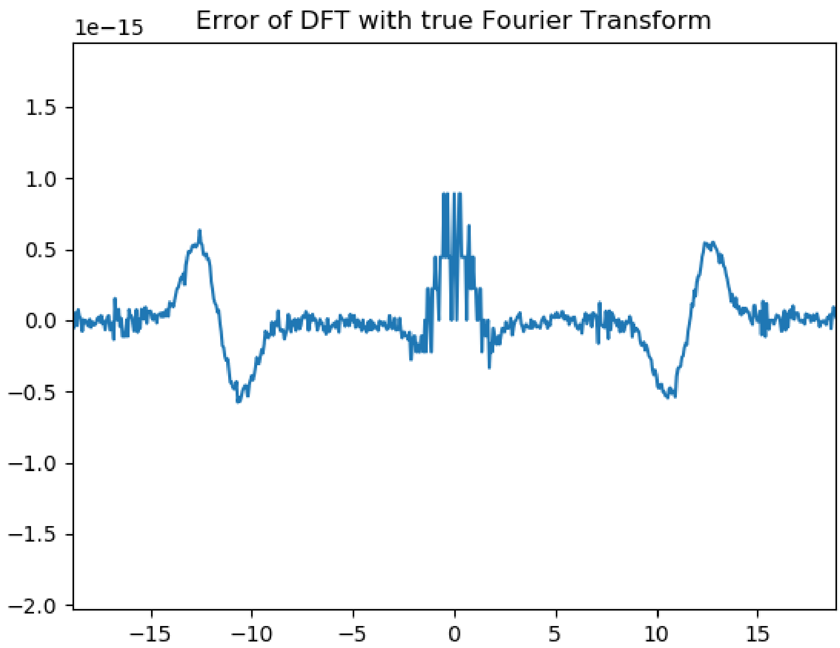
As explained above, the CTFT of the Gaussian signal is estimated using the DFT. The following plot was obtained for $F_s = 128$Hz, for a total time duration of 32π (N = 2048)

$$x(t) = e^{-t^2/2}$$
$$X(j\Omega) = \sqrt{2\pi}e^{-\Omega^2/2}$$



The error between the estimated FFT and the true FFT is quite low. As the FFT of Gaussian is real, the phase is 0

The error between the estimated and true FFT is plotted as a function of ω:



Error between the Estimated and True FFTs of Gaussian- note that in addition to this, the estimated FFT also has an imaginary part of the order of 1e-16

⋆⋆⋆