Assignment 8

Katari Hari Chandan [EE19B032]

May 2021

1 Introduction

This week's assignment deals with analysing signals using the Fast Fourier Transform(FFT) using Numpy's fft module. FFT is an implementation of the DFT.We also attempt to approximate the CTFT of a gaussian by changing the window size and number of samples until the error is below a threshold.

2 FFT and IFFT

We find the Fourier transform and invert it back to the time domain for a random signal, find maximum error to test the reconstruction

The maximum error obtained from the below code = 2.7781647035173285e-16

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

3 Spectrum of $sin(5t)$

The solution for this is already a part of the assignment. As expected the phase fro some values near the peaks is non zero. To fix this we sample the input signal at an appropriate frequency. We also shift the phase plot so that it goes from $-\pi$ to π. To do this we write a helper function

```
def dft(x_start,x_end,steps,f,xlim1,titl,ylabel1,ylabel2,xlabel1,savename,go=False):
    sampling_rate = steps/(x_end-x_start)
    x=linspace(x_start,x_end,steps+1)[:-1]
    y = f(x)
    Y=fftshift(fft(y))/float(steps)
    w=sampling_rate*(linspace(-pi,pi,steps+1)[:-1])
```

```
#plotting
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-xlim1,xlim1])
ylabel(ylabel1,size=16)
title(titl)
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
if(go):
    ii=where(abs(Y)>1e-3)
    plot(w[ii],angle(Y[ii]),'go',lw=2)
xlim([-xlim1,xlim1])
ylabel(ylabel2,size=16)
xlabel(xlabel1,size=16)
grid(True)
savefig(savename)
show()
return
```

As expected we get 2 peaks at +5 and -5 with height 0.5. The phases of the peaks at $\frac{\pi}{2}$ and $-\frac{\pi}{2}$ are also expected based on the expansion of a sine wave ie:

$$sin(5t) = 0.5(\frac{e^{5t}}{j} - \frac{e^{-5t}}{j}) \tag{1}$$

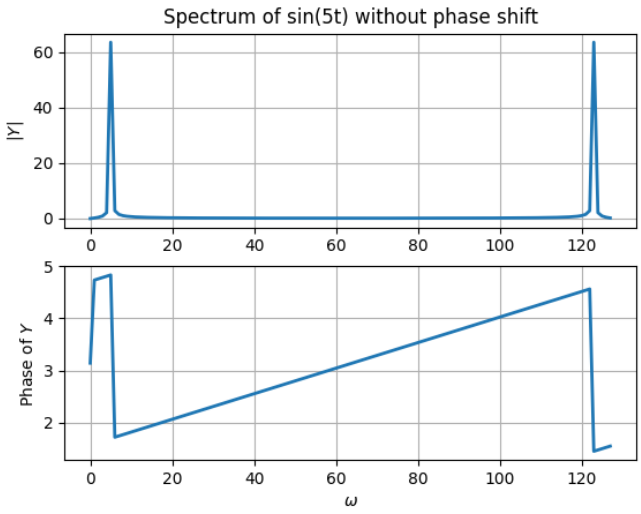

Figure 1: Spectrum of sin(5t) without Phase wrapping

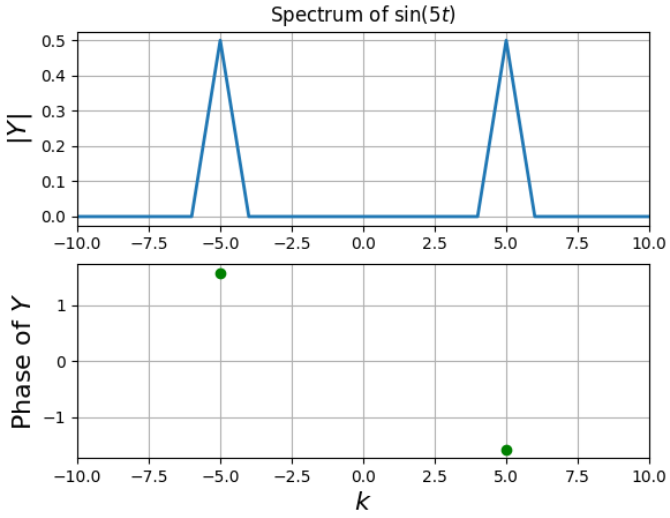

Figure 2: Spectrum of sin(5t) after phase wrapping

4 Spectrum of Amplitude Modulated Wave

Consider the signal:

$$f(t) = (1 + 0.1\cos(t))\cos(10t) \tag{2}$$

Using the same helper function as before, we get the following output: We note that 2 of the peaks have merged, we need to increase the number of samples we take. Calling the same function with a larger range and a higher number of samples we get 3 peaks. At all 3 peaks, the phase is 0 as expected for a cosine wave.

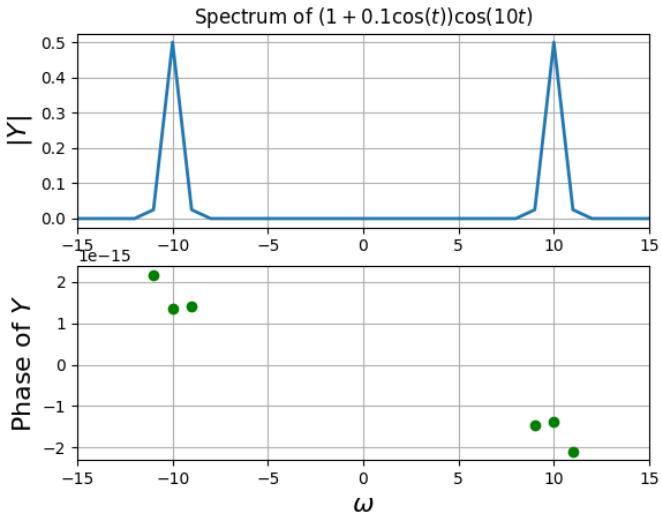

Figure 3: Spectrum of $f(t) = (1+0.1\cos(t))\cos(10t)$ with low number of samples

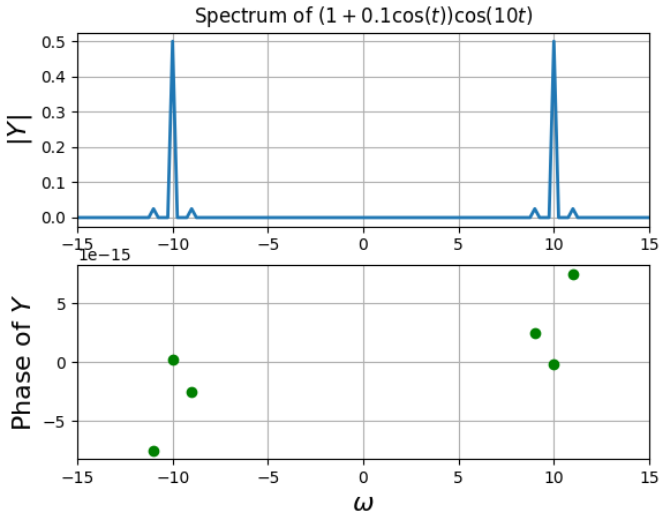

Figure 4: Spectrum of $f(t) = (1 + 0.1\cos(t))\cos(10t)$ with a higher number of samples

5 Spectrum of $sin^3(t)$

This signal can be expressed as a sum of sine waves using this identity:
$\sin^3(t) = \frac{3}{4}\sin(t) - \frac{1}{4}\sin(3t)$
We expect 2 peaks at frequencies 1 and 3, and phases similar to that expected from a sum of sinusoids.

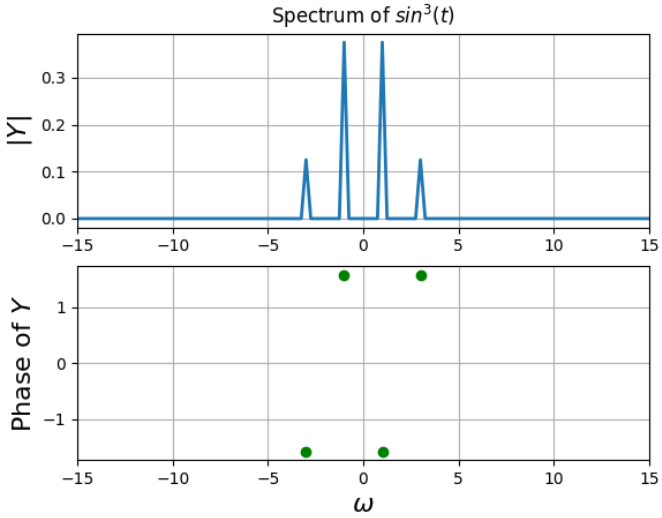

Figure 5: Spectrum of $f(t) = sin^3(t)$

6 Spectrum of $cos^3(t)$

This signal can be expressed as a sum of cosine waves using this identity:
$\sin^3(t) = \frac{3}{4}\cos(t) + \frac{1}{4}\cos(3t)$
We expect 2 peaks at frequencies 1 and 3, and phase=0 at the peaks.

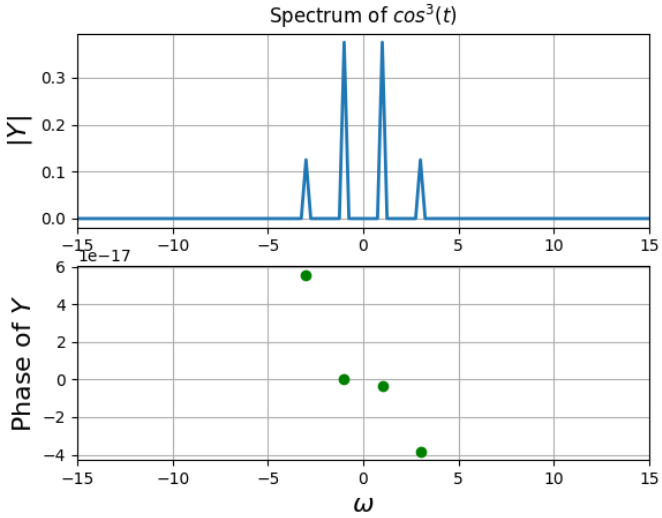

Figure 6: Spectrum of $f(t) = cos^3(t)$

7 Spectrum of Frequency Modulated Wave

Consider the signal:

$$f(t) = \cos(20t + 5\cos(t)) \tag{3}$$

Using the same helper function as before, we get the following output:

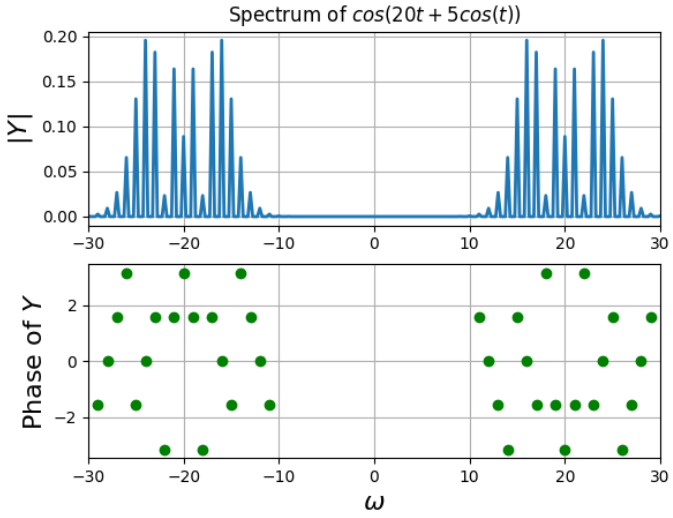

Figure 7: Spectrum of $f(t) = (1 + 0.1\cos(t))\cos(10t)$

The number of peaks has clearly increased. The energy in the side bands is comparable to that of the main signal.

8 Continuous time Fourier Transform of a Gaussian

The Fourier transform of a signal $x(t)$ is defined as follows:

$$X(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{4}$$

We can approximate this by the Fourier transform of the windowed version of the signal $x(t)$, with a sufficiently large window as Gaussian curves tend to 0 for large values of t . Let the window be of size T. We get:

$$X(\omega) \approx \frac{1}{2\pi}\int_{-T/2}^{T/2} x(t)e^{-j\omega t}dt \tag{5}$$

On writing the integral as a Riemann sum with a small time step $\Delta t = \frac{T}{N}$, We get:

$$X(\omega) \approx \frac{\Delta t}{2\pi} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} x(n\Delta t) e^{-j\omega n \Delta t} \tag{6}$$

Now, we sample our spectrum with a sampling period in the frequency domain of $\Delta\omega = \frac{2\pi}{T}$, which makes our continuous time signal periodic with period equal to the window size T. Our transform then becomes:

$$X(k\Delta\omega) \approx \frac{\Delta t}{2\pi} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} x(n\Delta t) e^{-j\frac{2\pi}{N} kn} \tag{7}$$

This form is similar to a DFT(for a finite window size). Therefore:

$$X(k\Delta\omega) \approx \frac{\Delta t}{2\pi} DFT\{x(n\Delta t)\} \tag{8}$$

We made a few approximations by using a finite window size and by using the Riemann approximation

We can improve these approximations by making the window size T larger, and by decreasing the time domain sampling period or increasing the number of samples N. We find the appropriate values for these iterative keeping the sampling frequency constant.

The expression for the Gaussian is :

$$x(t) = e^{\frac{-t^2}{2}} \tag{9}$$

The CTFT is given by:

$$X(j\omega) = \frac{1}{\sqrt{2\pi}} e^{\frac{-\omega^2}{2}} \tag{10}$$

```
def gauss(x):
    return exp(-0.5*x**2)

def expectedgauss(w):
    return 1/sqrt(2*pi) * exp(-w**2/2)

def estdft(tolerance=1e-6,samples=128,func = gauss,expectedfn = expectedgauss,wlim = 5):
    T = 8*pi
    N = samples
    Yold=0
    err=tolerance+1
    iters = 0
    while err>tolerance:
        x=linspace(-T/2,T/2,N+1)[:-1]
        w = linspace(-N*pi/T,N*pi/T,N+1)[:-1]
        y = gauss(x)
        Y=fftshift(fft(ifftshift(y)))*T/(2*pi*N)
        err = sum(abs(Y[::2] - Yold))
        Yold = Y
```

```
    iters+=1
    T*=2
    N*=2

true_error = sum(abs(Y-expectedfn(w)))
print("True error: {}".format(true_error))
print("No. of samples = "+str(N)+" time period = pi*"+str(T/pi))

mag = abs(Y)
phi = angle(Y)
phi[where(mag<tolerance)]=0

# plot estimate
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-wlim,wlim])
ylabel('Magnitude',size=16)
title("Estimate fft of gaussian")
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([-wlim,wlim])
ylabel("Phase",size=16)
xlabel("w",size=16)
grid(True)
show()


Y_ = expectedfn(w)

mag = abs(Y_)
phi = angle(Y_)
phi[where(mag<tolerance)]=0

figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-wlim,wlim])
ylabel('Magnitude',size=16)
title("True fft of gaussian")
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([-wlim,wlim])
ylabel("Phase",size=16)
xlabel("w",size=16)
grid(True)
show()

return
```

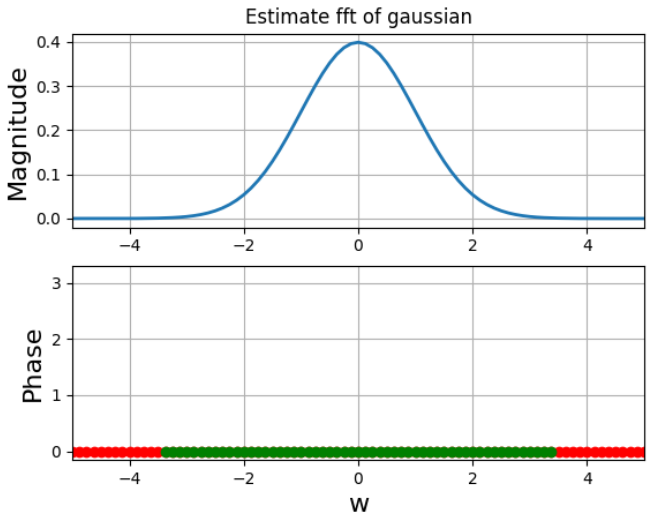

Figure 8: estimated CTFT of Gaussian

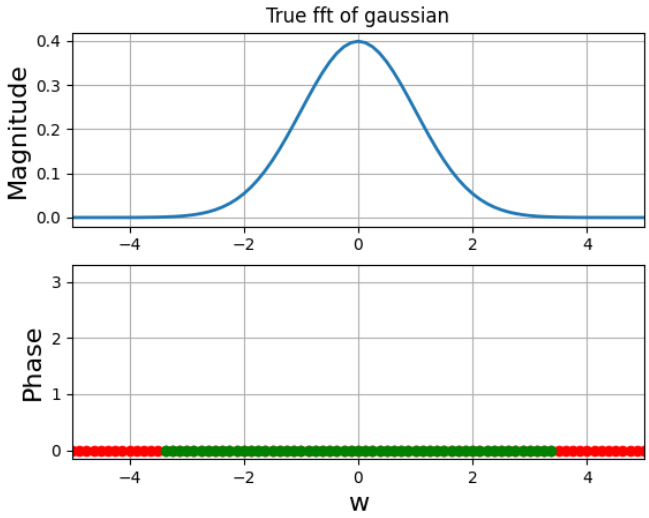

Figure 9: Expected CTFT of Gaussian

The summation of the absolute values of the error, Final number of samples and final window size are given as:

True error: 1.472553842671434e-14
samples = 512 time period = pi*32.0

9 Conclusion

We started off by testing the numpy.fft library on a random signal to check the error in in reconstructed signal,which was satisfactorily low. Then we moved on and examined the DFT's of sinusoids, AM signals, FM signals and weighted sum of sinusoids. Then we tried to iteratively estitmate the CTFT for a Gaussian signal. In each case, we utilise sampling of the DTFT to explain the spectrum or have used the discretisation of the continuous array the signal is derived from.There is a need for using ifftshift and fftshift to fix distorted phase responses.