EE2703:
APPLIED PROGRAMMING LAB

Assignment 9

Spectra of Non-Periodic Signals

Arjun R
EE20B016

May 12, 2022

Contents

1 Aim

Through this assignment, we aim to do the following:

- Obtain the Discrete Fourier Transform (DFT) of non-periodic functions
- Observe the effect of Hamming window in DFT computation
- Analyse spectra of different aperiodic functions

2 Introduction

- We focus on finding transforms of **non-periodic functions** in this assignment. Due to the discontinuity when periodically extended, fourier components in frequencies other than the sinusoids frequency are also produced which decay as $\frac{1}{\omega}$, due to Gibbs phenomenon.
- To tackle this, **Hamming window** is used. This make the signal square-integrable, and that the function goes sufficiently rapidly towards 0, also to make infinitely long signal to a finite signal.
- We also perform a sliding DFT on a chirped signal and plot the results.

3 Example Problems

3.1 Ex1: DFT of $sin(\sqrt{2}t)$

$sin(\sqrt{2}t)$ is non-periodic w.r.t. 2π time period. For finding the DFT, we take the signal from $-\pi$ to π. **Code implementation:**

```
t=linspace(-pi,pi,65);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
y=sin(sqrt(2)*t)
```

```
y[0]=0 # the sample corresponding to -tmax should be set zero
y=fftshift(y) # make y start with y(t=0)
Y=fftshift(fft(y))/64.0
w=linspace(-pi*fmax,pi*fmax,65);w=w[:-1]
```

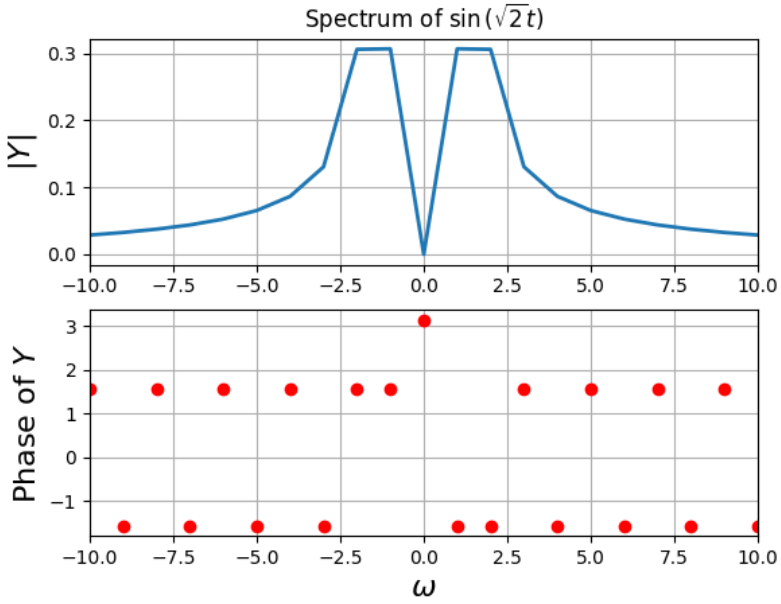

Figure 1: Spectrum of $\sin(\sqrt{2}t)$

We can observe from the plot that, instead of getting two spikes, we got two broad peaks with 2 finite values followed by a gradually decaying magnitude.

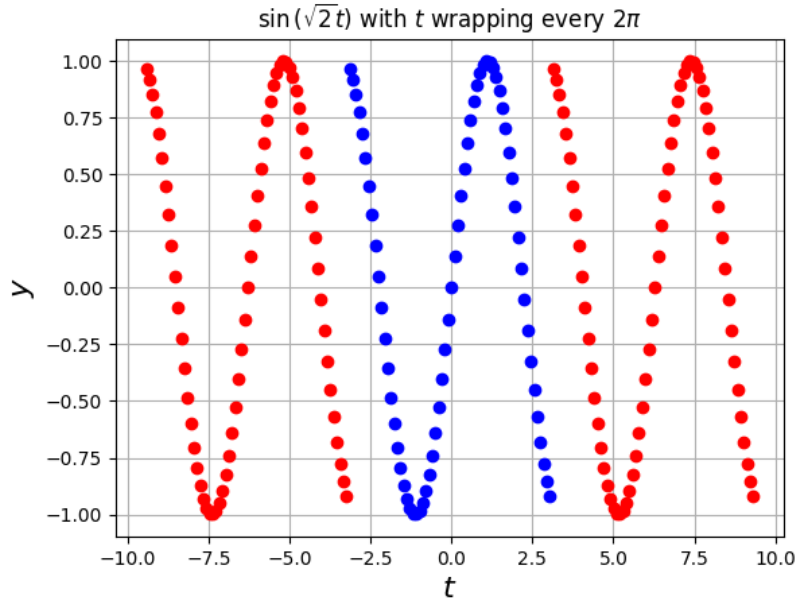

Figure 2

Since the DFT is computed over a finite time interval, discontinuities occur for these aperiodic signals. These discontinuities lead to non-harmonic components in the DFT which

decay as $\frac{1}{\omega}$. This can be seen with an example of a periodic ramp:

$$f(t) = t; -\pi < t < \pi$$

The Fourier series of f(t):

$$f(t) = 2(\frac{sint}{1} - \frac{sin2t}{2} + \frac{sin3t}{3} - ...)$$

The DFT spectra of the ramp would be as shown below:

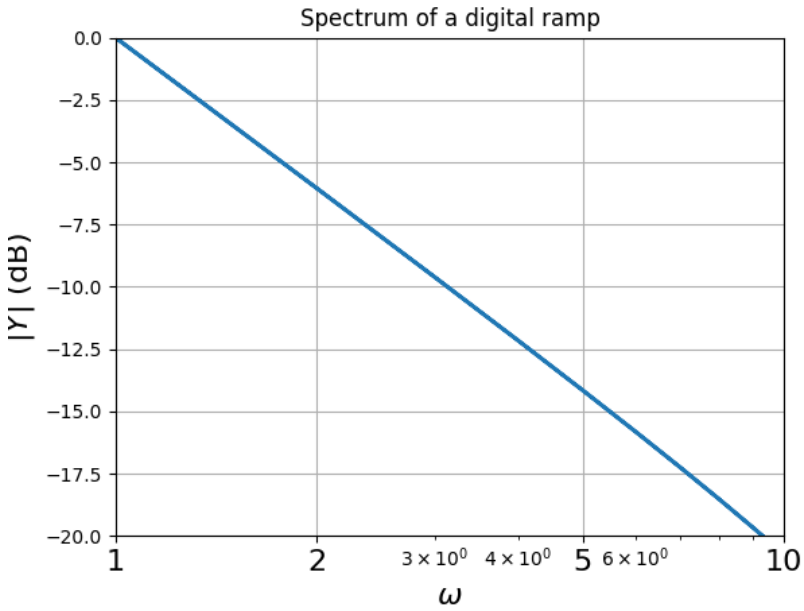

Figure 3: Spectrum of periodic ramp

Hence the coefficients decay slowly in $\frac{1}{\omega}$ fashion. In order to nullify the effect of $\frac{1}{\omega}$ decay, we use **Windowing**. Essentially, we damp the function near the periodic interval where the jump occurs, by multiplying the sequence with a window sequence w[n].

$$g(n) = f(n)w(n)$$

The window we will use is called the **Hamming window**:

$$w[n] = \begin{cases} 0.54 + 0.46cos\frac{2\pi n}{N-1} & |n| \leq \frac{N-1}{2} \\ 0 & \text{else} \end{cases} \tag{1}$$

Our time sequence $sin(\sqrt{2}t)$ windowed using the Hamming window will look as follows:

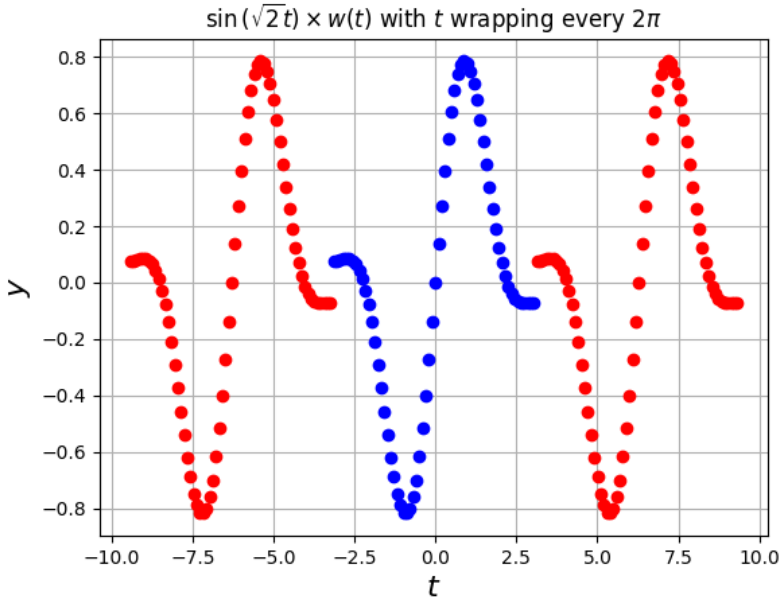

Figure 4

Now let us take the DFT of this sequence. The spectrum looks as below:

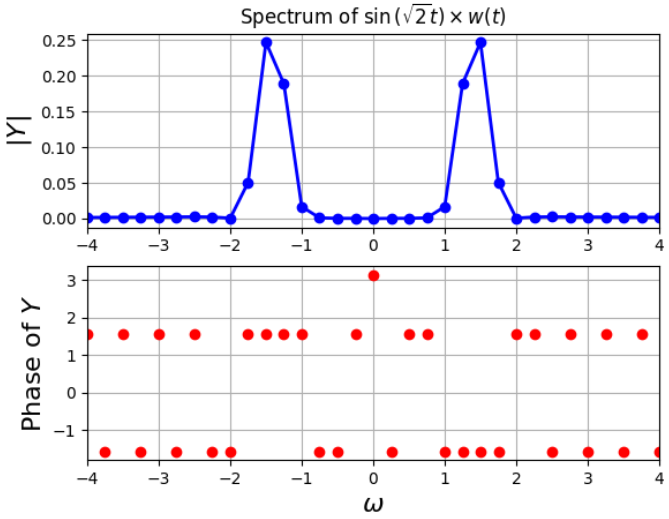

Figure 5: Spectrum of $\sin(\sqrt{2}t) * w(t)$

4 Defining necessary functions

I have used two functions namely, `spectrum` and `spectrum`$_h amming, both of which takes in the function for wh$

```
def spectrum(function, T, N, Title, xLim = 10, lineWidth = 2):
    """
    Finds DFT and plots Magnitude and Phase spectra plots
    Inputs:
    function: function to calculate DFT for
    N: number of samples to be taken
    Title: Title for the plots
    xLim: xlimits for the plots
    lineWidth: linewidth for the plots
    """
    t=linspace(-T/2, T/2, N+1);t=t[:-1]
    dt=t[1]-t[0];fmax=1/dt
    y=function(t)
    y[0]=0 # the sample corresponding to -tmax should be set zero
    y=fftshift(y) # make y start with y(t=0)
    Y=fftshift(fft(y))/float(N)
    w=linspace(-pi*fmax,pi*fmax,N+1);w=w[:-1]
    figure()
    subplot(2,1,1)
    plot(w,abs(Y),lw=lineWidth)
    xlim([-xLim, xLim])
    ylabel(r"$|Y|$",size=16)
    title("Spectrum of "+Title)
    grid(True)
    subplot(2,1,2)
    plot(w,angle(Y),'ro',lw=lineWidth)
    xlim([-xLim, xLim])
    ylabel(r"Phase of $Y$",size=16)
    xlabel(r"$\omega$",size=16)
    grid(True)
    show()
    return Y

def spectrum_hamming(function, T, N, Title, xLim = 10, lineWidth = 2):
    """
    Finds DFT after applying Hamming window and plots Magnitude and Phase spectra plots
    Inputs:
    function: function to calculate DFT for
    N: number of samples to be taken
    Title: Title for the plots
    xLim: xlimits for the plots
    lineWidth: linewidth for the plots
    """
    t=linspace(-T/2, T/2, N+1);t=t[:-1]
    dt=t[1]-t[0];fmax=1/dt
    n=arange(N)
    wnd=fftshift(0.54+0.46*cos(2*pi*n/N))
    y=function(t)
```

```
y=y*wnd
y[0]=0 # the sample corresponding to -tmax should be set zero
y=fftshift(y) # make y start with y(t=0)
Y=fftshift(fft(y))/float(N)
w=linspace(-pi*fmax,pi*fmax,N+1);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=lineWidth)
xlim([-xLim, xLim])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of "+Title)
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=lineWidth)
xlim([-xLim, xLim])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
show()
return Y
```

5 Assignment Problems

5.1 P1: DFT of $cos^3(0.86t)$

In this question, we shall obtain the DFT spectrum of $cos^3(\omega_0 t)$ for $\omega_0 = 0.86$.
The FFT without the Hamming Window:

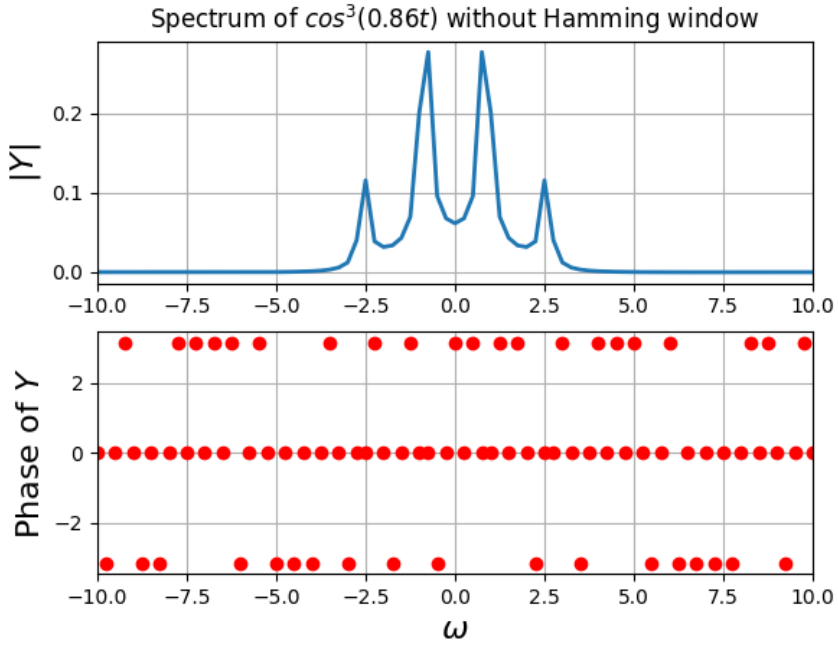

Figure 6

The FFT with the Hamming Window:

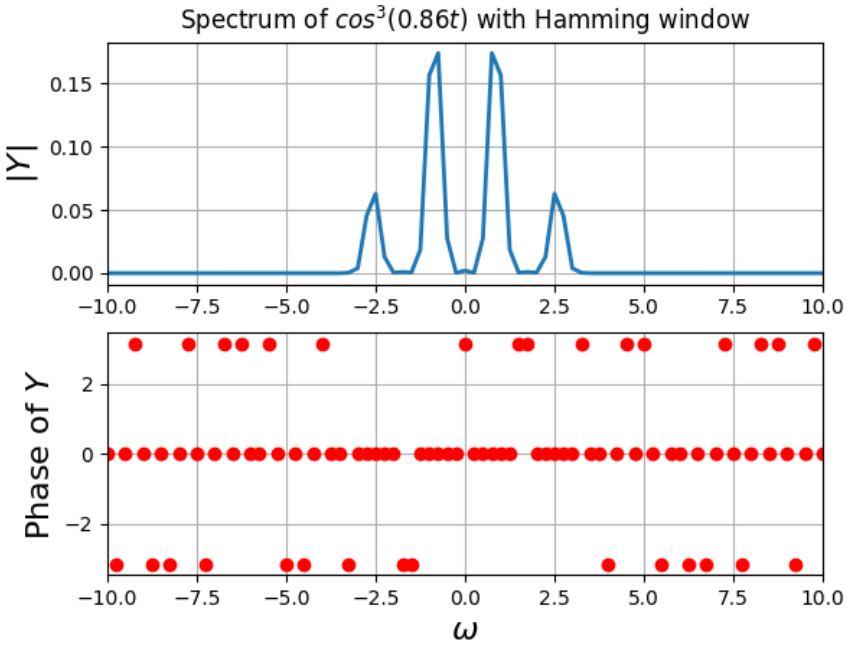

Figure 7

We notice that a lot of the energy is stored in frequencies that aren't a part of the signal. After windowing, these frequencies are attenuated and hence the peaks are sharper in the windowed function. It is still not an impulse because the convolution with the Fourier transform of the windowed function smears out the peak.

5.2 P2: DFT of $\cos(\omega t + \delta)$ and estimating ω and δ

- The resolution is not enough to obtain the ω_0 directly if the spectrum is obtained. Since the resolution of the frequency axis is not enough, the peak won't be visible clearly. So a statistical derivation is necessary to estimate value of ω_0. Thus, we can compute ω_0 by taking a weighted average of all the ω weighted with the magnitude of the DFT.
- δ can be found by calculating the phase of the DFT at ω_0 nearest to estimated ω using the above statistic.
- This works because the phase of $\cos(\omega_0 t + \delta)$ when $\delta = 0$ is 0, so when it's not, it's δ, so we can estimate it by this approach.

We estimate ω and δ for a signal $\cos(\omega t+\delta)$ for 128 samples between $[-\pi, \pi)$. We then extract the digital spectrum of the signal and find the two peaks at $\pm\omega_0$, and estimate ω and δ.

Code implementation:

```
random.seed(0)
w0 = 0.5 + random.random() # 0.5-1.5
d = random.random()*(2*pi) # 0-2pi

T = 8*pi
```

```
N = 256
t=linspace(-T/2, T/2, N+1);t=t[:-1]
def fn3(t):
    return cos(w0*t+d)

Y_3 = spectrum_hamming(fn3, T, N, r"$cos(\omega_0t+\delta)$", 5)
# Weighted average
fmax = N/T
w = linspace(-pi*fmax,pi*fmax,N+1);w=w[:-1]
ii = where(w>0)
w0_est = sum(w[ii]*abs(Y_3[ii]))/sum(abs(Y_3[ii]))
i = abs(w - w0_est).argmin()
d_est = w[i]
```

5.2.1 Without white noise:

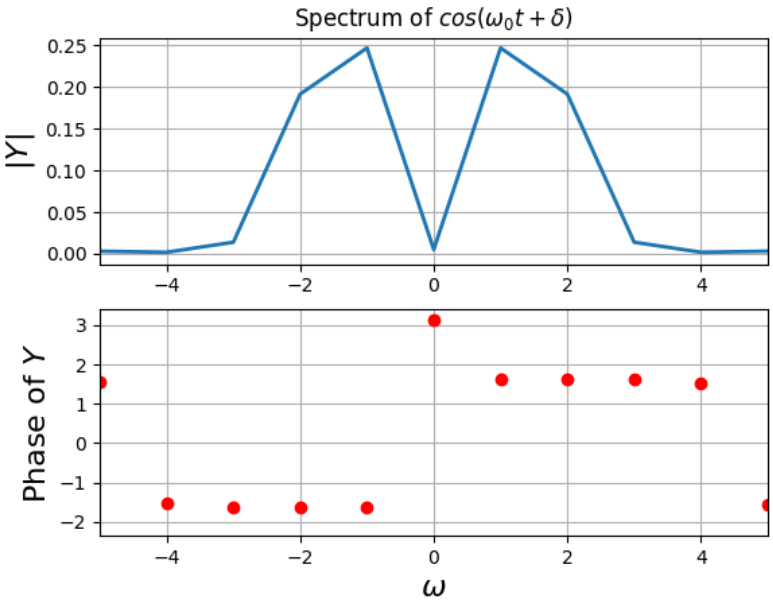

Figure 8: DFT of $\cos(\omega t + \delta)$

```
   Output:
Without noise:
Actual w0:  1.3444218515250481
Calculated w0:  1.3184174061564733
Actual delta:  1.6207753144767914
Calculated delta:  1.6201735116557918
```

5.2.2 With white noise:

We repeat the exact same process but with noise added to the original signal.

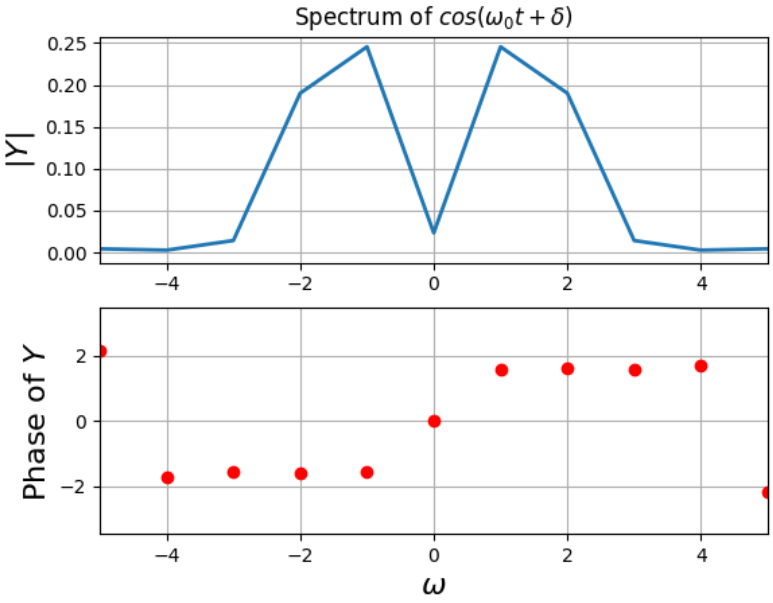

Figure 9: DFT of noisy $\cos(\omega t + \delta)$

```
   Output:
With noise:
Actual w0:  1.3444218515250481
Calculated w0:  1.314671383625852
Actual delta:  1.6207753144767914
Calculated delta:  1.567454718127288
```

5.3 P3: Chirped Signal Spectrum

- We analyze a chirped signal which is an FM signal where frequency is directly proportional to time. A chirped signal we shall consider is given by

$$f(t) = cos(16t(1.5 + \frac{t}{2\pi})) \tag{2}$$

- Its frequency continuously changes from 16 to 32 radians per second. This also means that the period is 64 samples near $-\pi$ and 32 samples near π.
- Due to Gibbs phenomenon, the frequency range is large. On windowing, only frequencies between 16 and 32 rad/s remain.

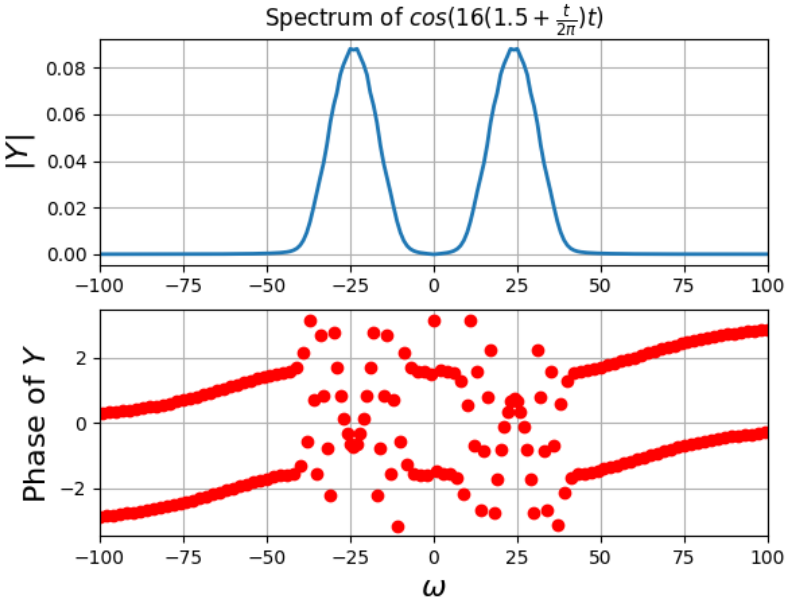

Figure 10: DFT of Windowed Chirped signal

5.4 P4: Frequency vs Time for chirped signal

- For the above mentioned chirped signal, 16 strips of 64 samples wide are made out of 1024 samples.
- We then extract the DFT of each and store as a column in a 2D array. Then, we plot the array as a surface plot to show how the frequency of the signal varies with time.
- This is a "time-frequency" plot,and we get localized DFTs. We do this for both phase and magnitude.

6 Conclusion

- We have emphasized the importance and necessity of windowing in the case of non-periodic series in DFT's. In particular this is to negate the effect of Gibbs phenomenon owing to the discontinuous nature of the series $\tilde{x}[n]$ realised by a discrete Fourier transform.
- By using approaches like the weighted average, we can obtain certain features of the signal such as its frequency and phase.
- We can also obtain surface plots by which we can analyse localised DFTs and can show how the spectra evolves with time.

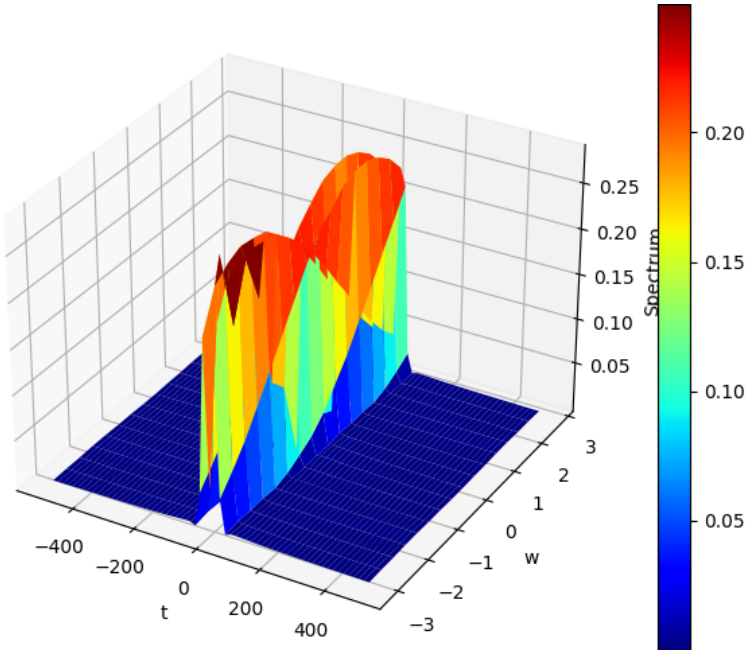

Figure 11: Chirp function, DFT Magnitude

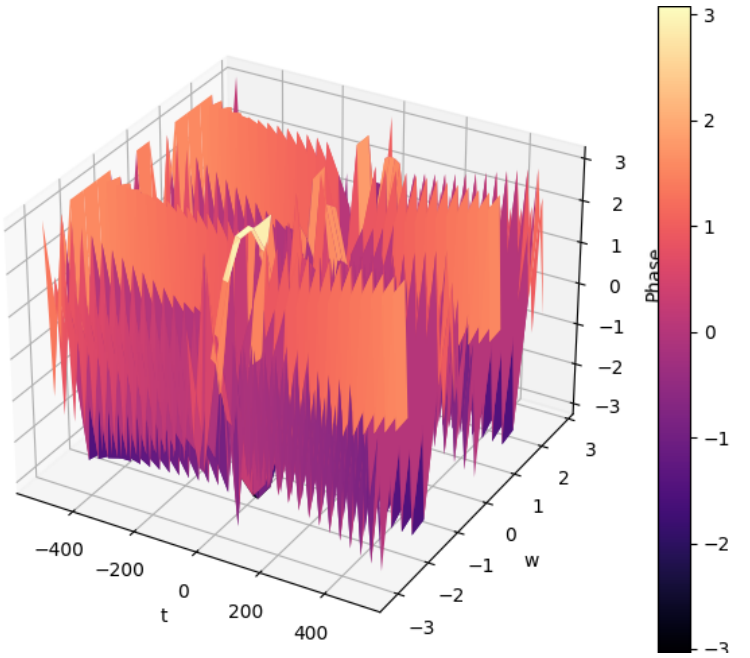

Figure 12: Chirp function, DFT Phase