Assignment 9: Spectra of non-periodic signals

Chollangi Dheeraj Sai [EE20B029]

May 11, 2022

Overview

In this assignment, we will learn how to find transforms of non periodic functions. These kind of functions show discontinuities when they are periodically extended. These discontinuities give rise to frequency components which decay exponentialy with a factor $\frac{1}{\omega}$. We will learn how to solve this problem of different frequencies using python.

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

The asymptotic bode plots of the aperiodic sinusoid signal are shown below.

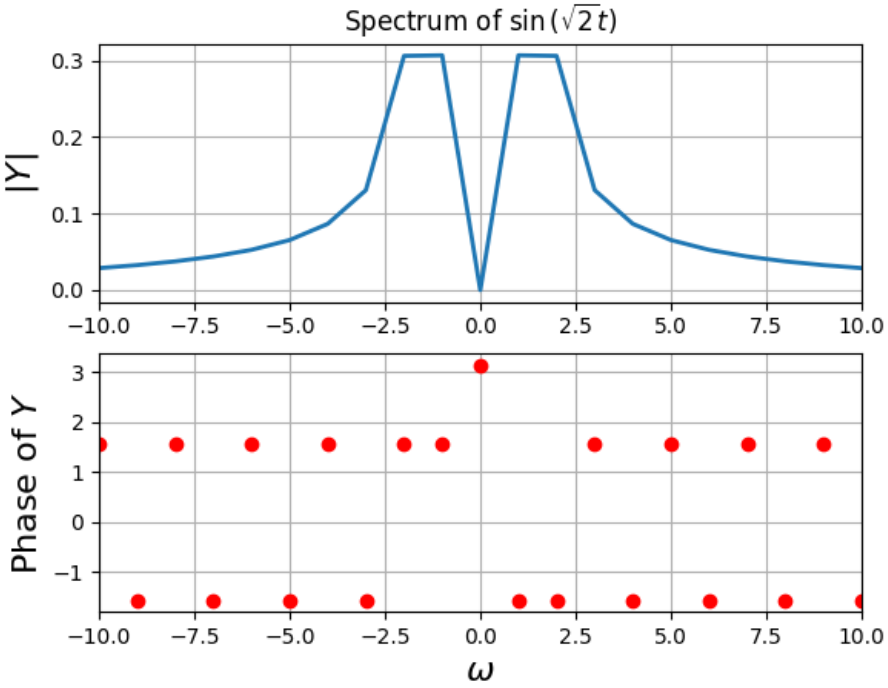


Figure 1: Spectrum of $\sin(\sqrt{2}t)$

The actual plot of the signal used to plot the DFT.

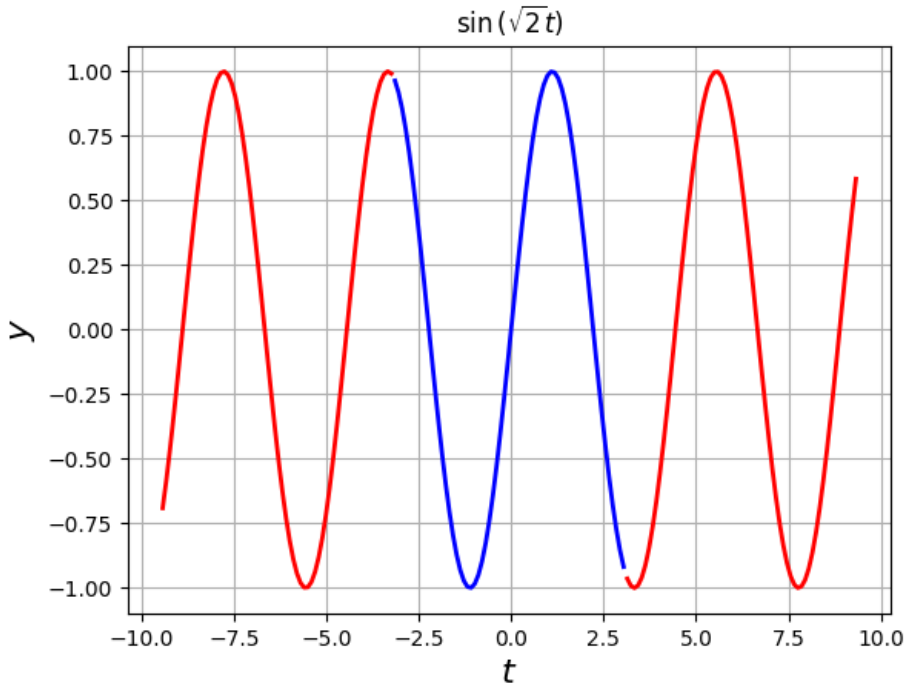


Figure 2: $\sin(\sqrt{2}t)$

Periodic extension of the signal in the interval wrapped for finding the DFT.

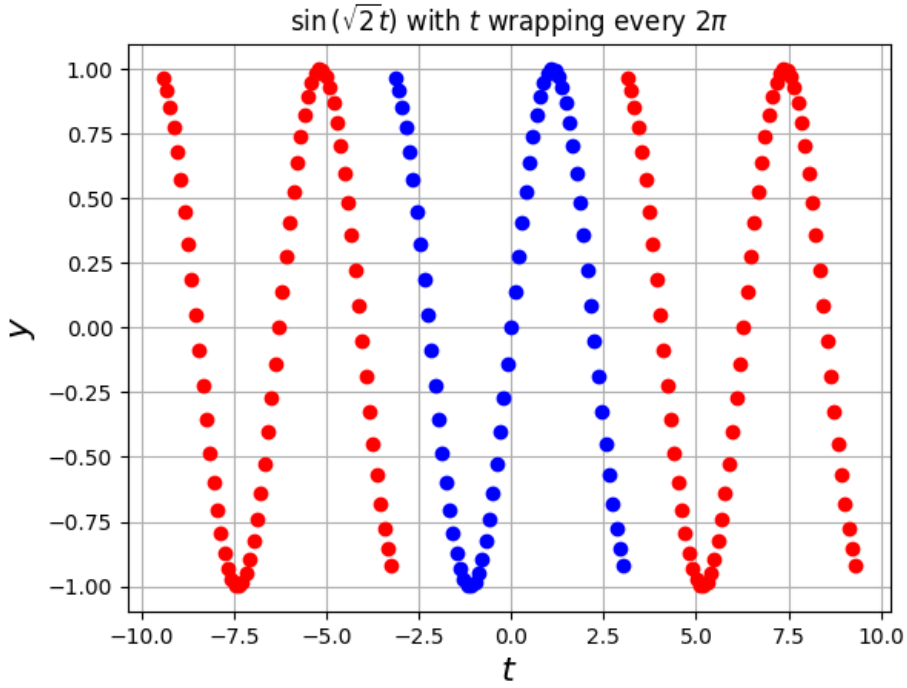


Figure 3: Spectrum of $\sin(\sqrt{2}t)$

The plot showing the decay of non harmonic components in the periodic ramp signal is shown to have a general idea of decay of frequency components during discontinuities.

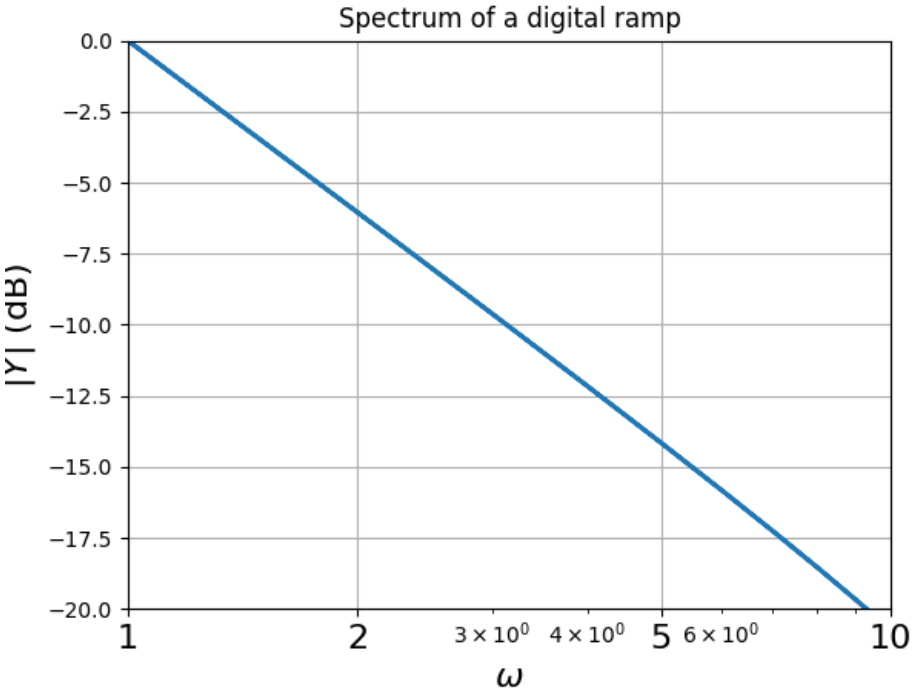


Figure 4: Spectrum of digital ramp

2 Hamming window

The hamming window removes discontinuities by attenuating the high frequency components that cause the discontinuities. The hamming window function is given by

$$x[n] = 0.54 + 0.46cos(\frac{2\pi n}{N-1}) \tag{1}$$

When we multiply the signal with the hamming window and extend it periodically, we can see that the discontinuities will almost vanish.

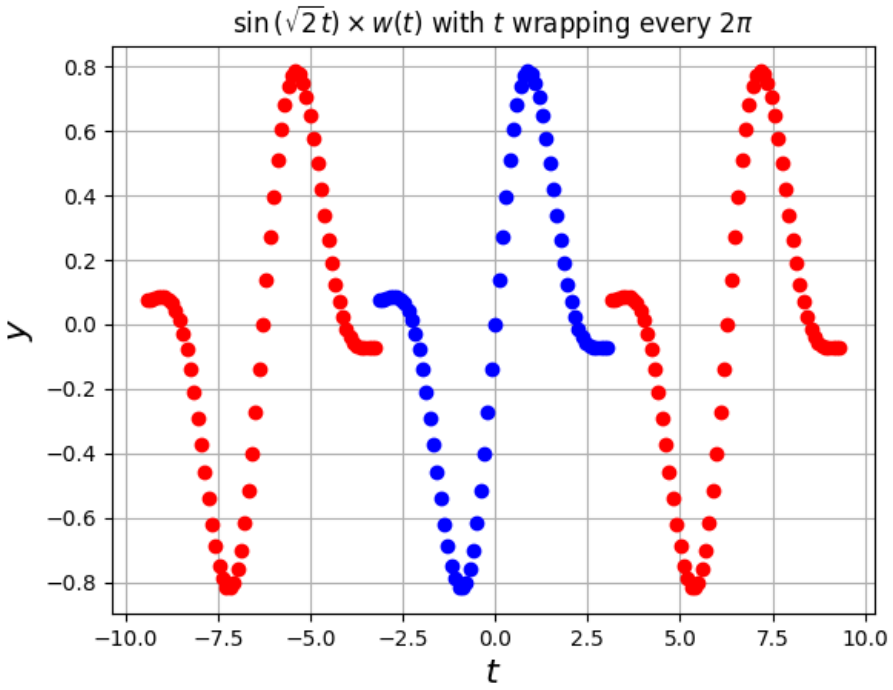


Figure 5: Spectrum of $\sin(\sqrt{2}t) * w(t)$

The spectrum that is obtained with a time period 2π is given below:

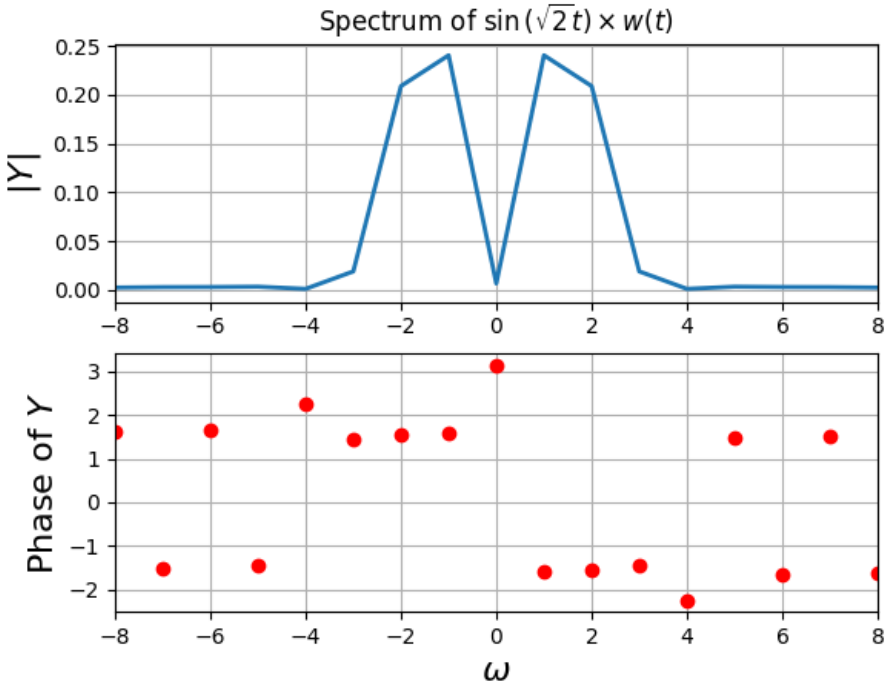


Figure 6: Spectrum of $\sin(\sqrt{2}t) * w(t)$

3 Initializing potential

The same spectrum when we use four times the number of points is is shown below. It can be seen that we get sharper and more detailed peaks.

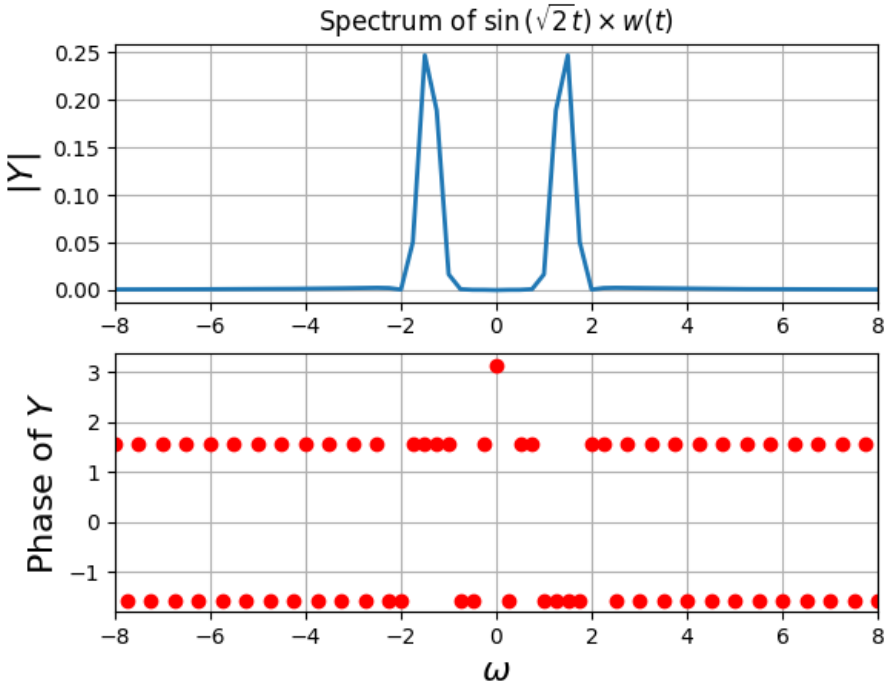


Figure 7: Spectrum of $\sin(\sqrt{2}t) * w(t)$

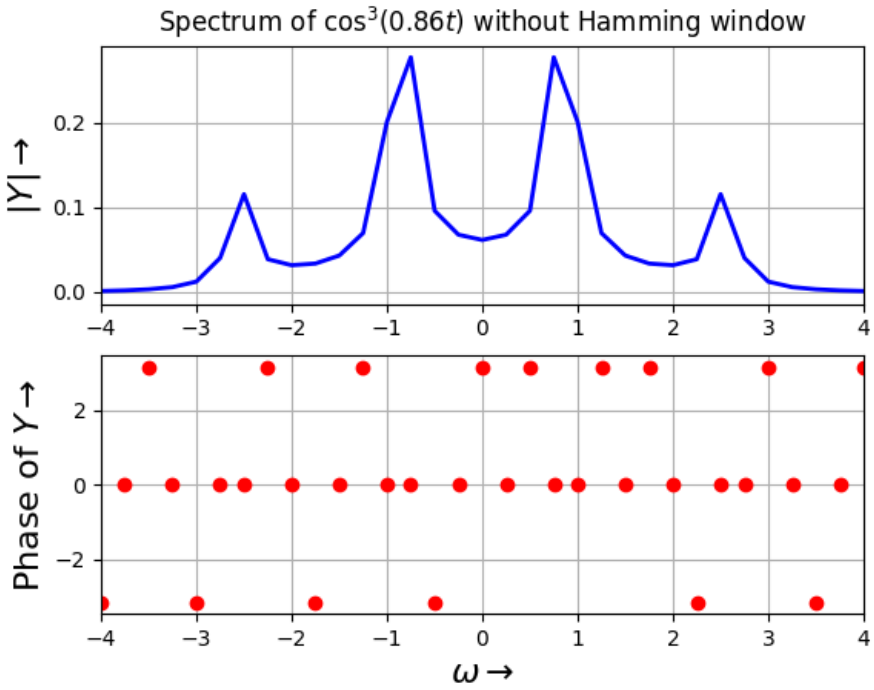
4 DFT of $cos^3(0.86t)$

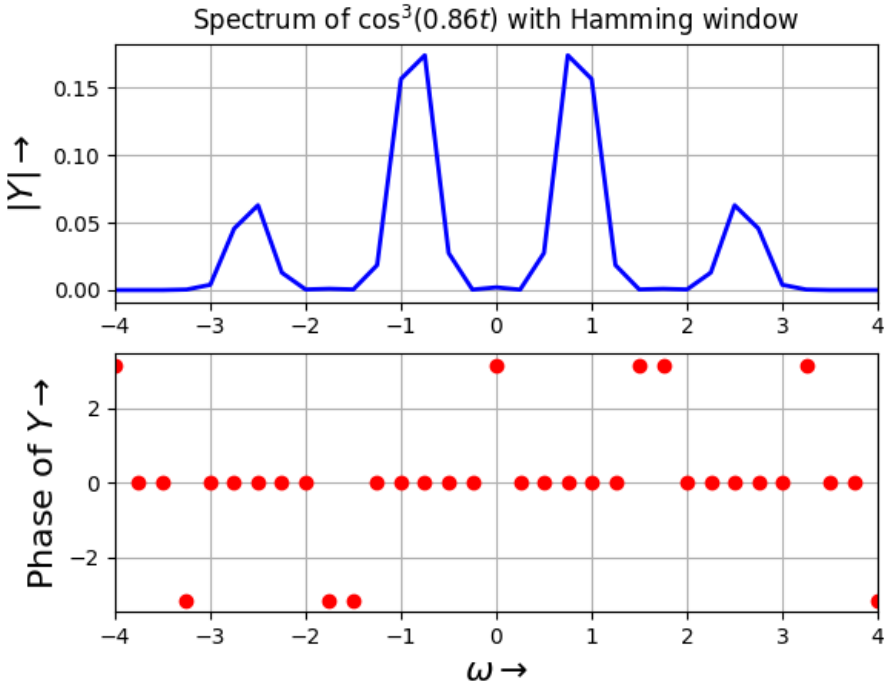
Using the below code will plot the DFT of $cos^3(0.86t)$. We are considering both the cases, one in which we are not multiplying with the hamming window and the other in which we are multiplying with the hamming window.

```
def assymptotic_plots(w, Y, xlimit, Title, ylabel1, ylabel2, Xlabel):
    figure()
    subplot(2, 1, 1)
    plot(w, abs(Y), "b", lw=2)
    xlim([-xlimit, xlimit])
    ylabel(ylabel1, size=16)
    title(Title)
    grid()
    subplot(2, 1, 2)
    plot(w, angle(Y), "ro", lw=2)
    xlim([-xlimit, xlimit])
    ylabel(ylabel2, size=16)
    xlabel(Xlabel, size=16)
    grid()
```

```
y = cos(0.86 * t) ** 3
y1 = y * wnd
y[0] = 0
y1[0] = 0
y = fftshift(y)
y1 = fftshift(y1)
Y = fftshift(fft(y)) / 256.0
Y1 = fftshift(fft(y1)) / 256.0

assymptotic_plots(
    w,
    Y,
    4,
    r"Spectrum of $\cos^{3}(0.86t)$ without Hamming window",
    r"$|Y|\rightarrow$",
    r"Phase of $Y\rightarrow$",
    r"$\omega\rightarrow$",
)
assymptotic_plots(
    w,
    Y1,
    4,
    r"Spectrum of $\cos^{3}(0.86t)$ with Hamming window",
    r"$|Y|\rightarrow$",
    r"Phase of $Y\rightarrow$",
    r"$\omega\rightarrow$",
)
```


Spectrum of cos^3(0.86t) without Hamming window
|Y| →
0.0
0.1
0.2
−4
−3
−2
−1
0
1
2
3
4
Phase of Y →
2
0
−2
ω →



5 Estimating ω_0 and δ

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

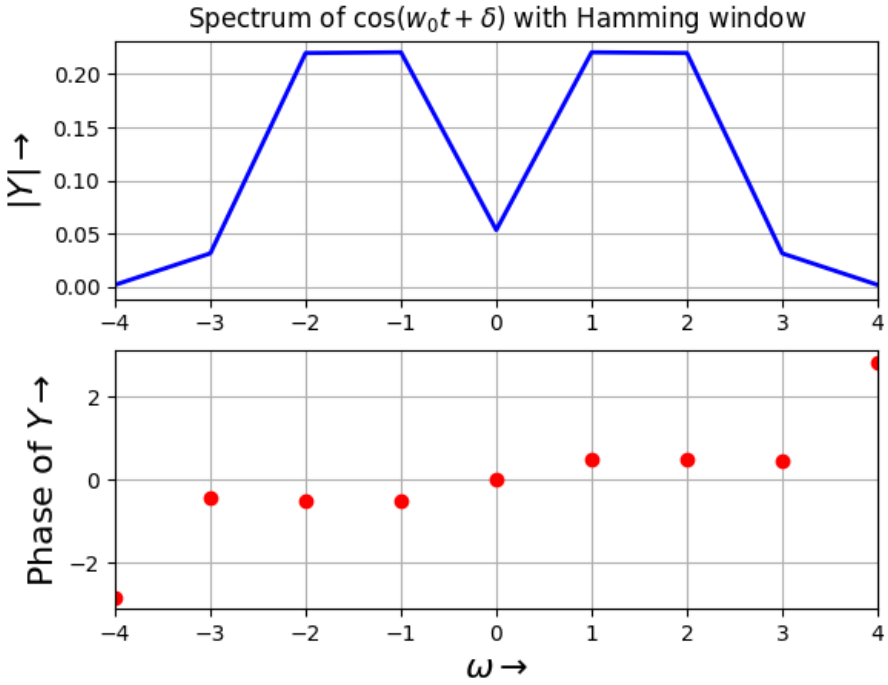


Figure 8: Fourier transform of $cos(1.5t + 0.5)$

We estimate omega by performing a Mean average of ω over the magnitude of $|Y(j\omega)|$. We can estimate δ by calculating the phase of the DFT at ω_0 nearest to the estimated ω.

```
ii = where(w > 0)
omega = sum(abs(Y[ii]) ** 2 * w[ii]) / sum(abs(Y[ii]) ** 2)
i = abs(w - omega).argmin()
delta = angle(Y[i])
```

Estimated value of ω_0 without noise = 1.5163179648582412 Estimated value of δ without noise = 0.506776265719626

6 Estimating ω_0 and δ in presence of noise

We repeat the exact same process as question 3 but with noise added to the original signal.

```
y = (cos(w0 * t + d) + 0.1 * randn(128)) * wnd
y[0] = 0
y = fftshift(y)
Y = fftshift(fft(y)) / 128.0
```

```
ii = where(w > 0)
omega = sum(abs(Y[ii]) ** 2 * w[ii]) / sum(abs(Y[ii]) ** 2)
i = abs(w - omega).argmin()
delta = angle(Y[i])
```

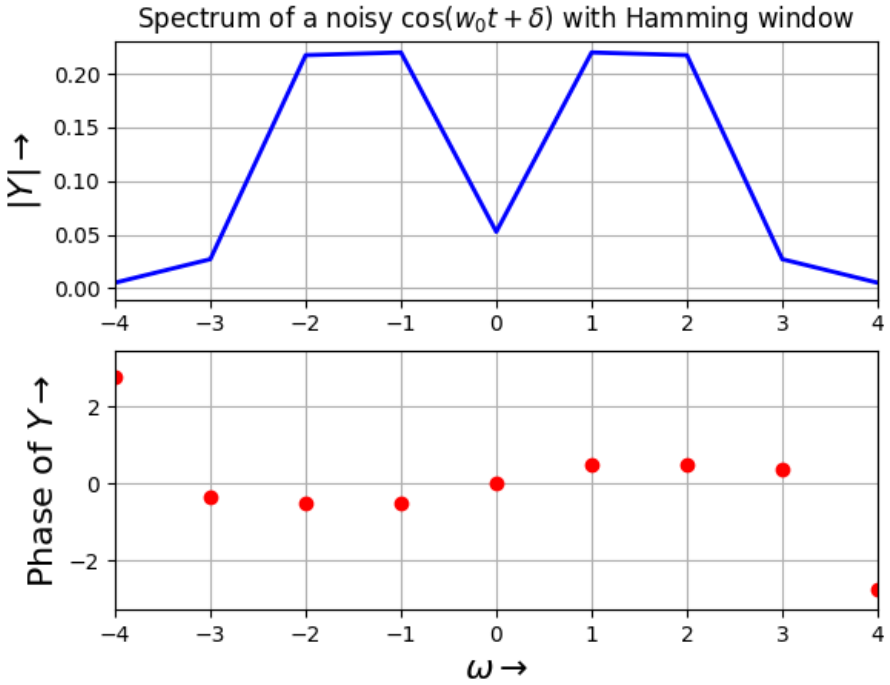


Figure 9: Fourier transform of $cos(1.5t + 0.5)$

Estimated value of ω_0 with noise: 2.080814397289841 Estimated value of δ with noise: 0.5075828642817306

7 Analysis of chirped signal

The frequency of the signal is proportional to t and is continuously changing from 16 to 32 radians per second. The code for plotting the DFT after windowing and the corresponding plot are shown below.

```
t = linspace(-pi, pi, 1025)
t = t[:-1]
dt = t[1] - t[0]
fmax = 1 / dt
n = arange(1024)
```

```
wnd = fftshift(0.54 + 0.46 * cos(2 * pi * n / 1024))
y = cos(16 * t * (1.5 + t / (2 * pi))) * wnd
y[0] = 0
y = fftshift(y)
Y = fftshift(fft(y)) / 1024.0
w = linspace(-pi * fmax, pi * fmax, 1025)
w = w[:-1]
assymptotic_plots(
    w,
    Y,
    100,
    r"Spectrum of chirped function",
    r"$|Y|\rightarrow$",
    r"Phase of $Y\rightarrow$",
    r"$\omega\rightarrow$",
)
```

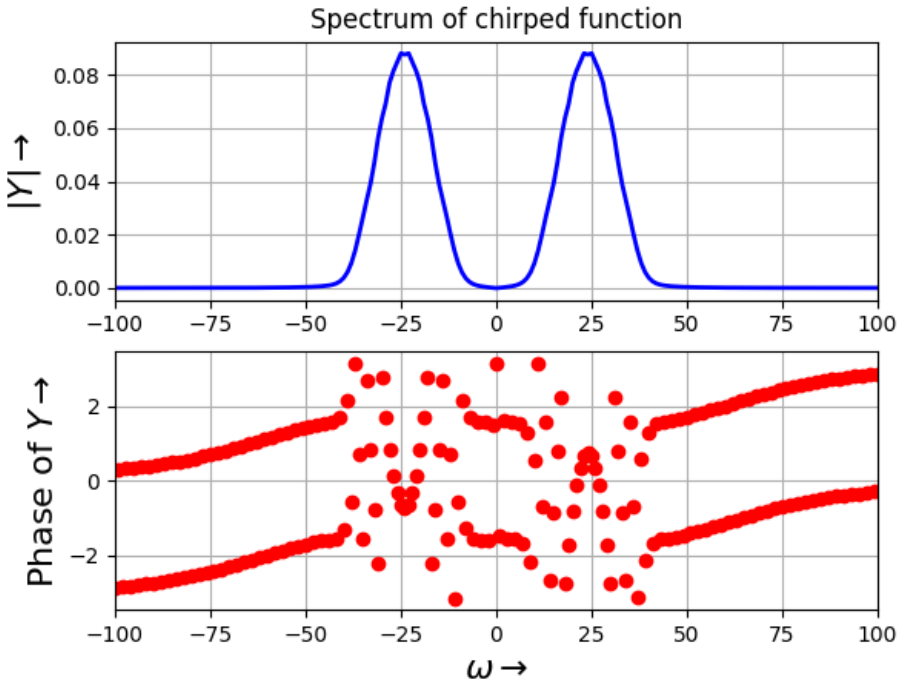


Figure 10: Chirp function fourier transform, windowed

8 Surface plot of chirped signal

For the same chirped signal, we break the 1024 vector into pieces that are 64 samples wide. Extract the DFT of each and store as a column in a 2D array. Then plot the array as a surface plot to show how the frequency of the signal varies with time.

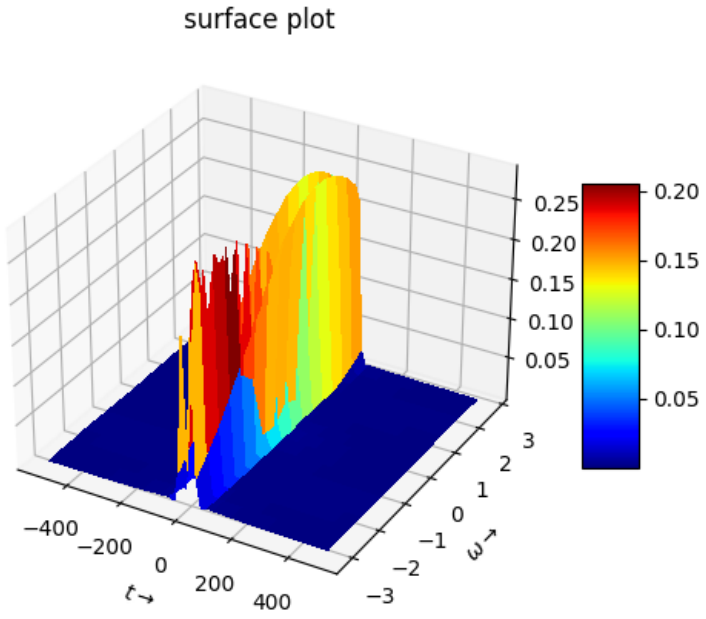


Figure 11: surface plot

9 Conclusion

- We learned how to obtain DFTs of non periodic signals.
- We learned how to use windowing functions to attenuate the non harmonic frequency components.
- We learned how discontinuities in non periodic signal introduces new frequency components in the original signal and we were able to plot the same.