Task 3

Due: Feb 16th, 11:59pm

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal
import scipy.integrate as integrate
import scipy.io as io
from scipy.fftpack import fft
```

Question 1

1a. Compute an analytic y(t) which is the convolution of f(t) and g(t):

1.a. $f(t) = \begin{cases} 5 & |t| < 3. \\ 0 & \text{otherwise.} \end{cases}$ $\quad g(t) = \begin{cases} 2 & |t| < 3 \\ 0 & \text{otherwise.} \end{cases}$

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) \, d\tau.$$

$\because$ Piecewise function $\Rightarrow$ Break into 4 zones: $(-\infty, -6). [-6, 0]. (0, 6]. (6, \infty).$

$$\Rightarrow f*g = \cancelto{0}{\int_{-\infty}^{-6} \cdots d\tau} + \int_{-6}^{0} f(\tau) g(t-\tau) \, d\tau + \int_{0}^{6} f(\tau) g(t-\tau) \, d\tau + \cancelto{0.}{\int_{6}^{\infty} \cdots d\tau}$$

$$= \begin{cases} (t+3-(-3)) \times 10 & t \in [-6, 0] \\ (3-(t-3)) \times 10 & t \in (0, 6] \\ 0 & \text{otherwise.} \end{cases} = \begin{cases} 10t + 60 & t \in [-6, 0] \\ -10t + 60 & t \in (0, 6]. \\ 0 & \text{otherwise.} \end{cases}$$

1b. Write a code to numerically compute y(t) and plot y(t). Please use for-loop and do not use conv.

```
f_t = lambda t: 5 if abs(t)<=3 else 0
g_t = lambda t: 2 if abs(t)<=3 else 0
```

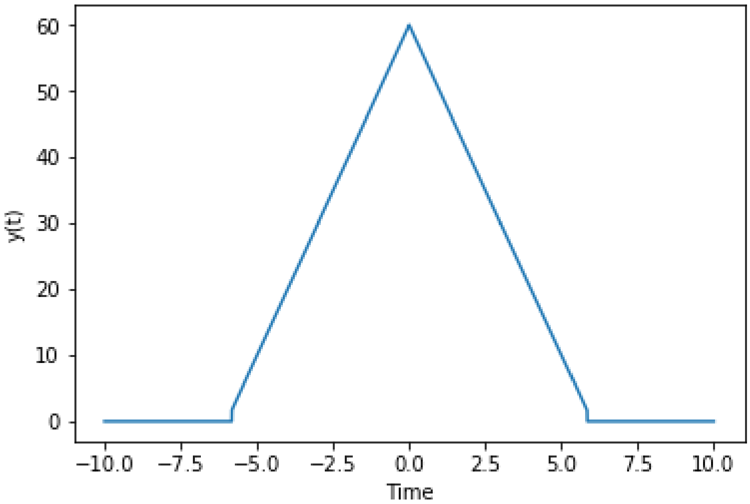
```
step_scale = 1000
timesteps = np.arange(-10,10, 1/step_scale)
```

```
conv_result = []
for t in np.arange(-10, 10, 1/step_scale):
    g_tao = lambda tao: 2 if abs(t-tao)<=3 else 0
    f_g_tao = lambda tao: f_t(tao)*g_tao(tao)
    conv_result.append(integrate.quad(f_g_tao, -10, 10)[0])

plt.plot(timesteps,conv_result)
plt.xlabel('Time')
plt.ylabel('y(t)')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\integrate\quadpack.py:385:
IntegrationWarning: The maximum number of subdivisions (50) has been achieved.
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties.  If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges.  Perhaps a special-purpose integrator should be used.
  warnings.warn(msg, IntegrationWarning)
```

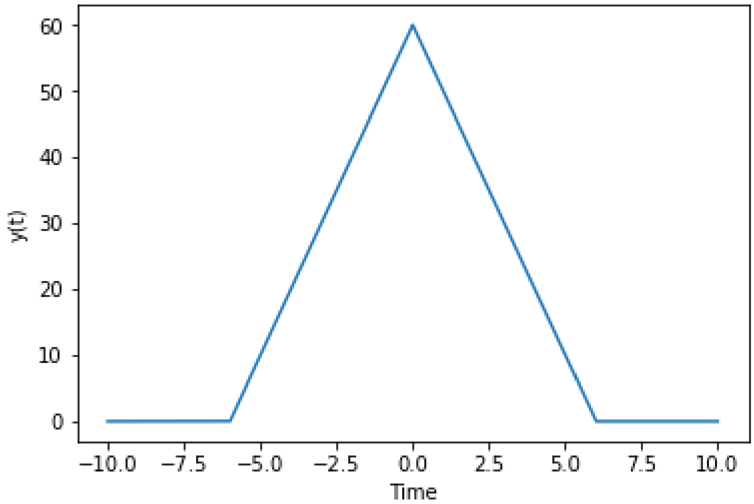
```
Text(0,0.5,'y(t)')
```



1c

```
c = signal.convolve(list(map(f_t, timesteps)),list(map(g_t, timesteps)),
'same')/step_scale
plt.plot(timesteps, c)
plt.xlabel('Time')
plt.ylabel('y(t)')
```

```
Text(0,0.5,'y(t)')
```



Question 2

2a. Proof the convolution theorem and explain the meaning of these relationships in your words.

a). ① $\because h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau) \cdot x(t-\tau)\, d\tau$

$\quad$ & $F(u) = \int_{-\infty}^{\infty} u\, e^{-i2\pi f t} dt.$

$\Rightarrow F(h(t) * x(t)) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} h(\tau) x(t-\tau)\, d\tau \right] e^{-i2\pi f t} dt.$

let $v = t - \tau$. then

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\tau) x(v)\, e^{-i2\pi f(\tau + v)}\, d\tau\, dv.$$

$$= \underbrace{\int_{-\infty}^{\infty} h(\tau) e^{-i2\pi f(\tau)} d\tau}_{\Downarrow \atop H(f)} \underbrace{\int_{-\infty}^{\infty} x(v) e^{-i2\pi f(v)} dv}_{\Downarrow \atop X(f)} = H(f) \cdot X(f).$$

$F(x(t) \cdot h(t)) = \int_{-\infty}^{\infty} x(t) h(t)\, e^{-i2\pi f t} dt.$

& $x(t) = \int_{-\infty}^{\infty} X(f)\, e^{i2\pi f t} dt \quad h(t) = \int_{-\infty}^{\infty} H(t) e^{i2\pi f t} dt.$

$$\therefore F = \int_{-\infty}^{\infty} X(f_1) \int_{-\infty}^{\infty} H(f_2) \underbrace{\int_{-\infty}^{\infty} e^{-i2\pi(f-f_1-f_2)t}\, dt.}_{\hookrightarrow\ \delta(f-f_1-f_2)}\, df_2\, df_1$$

$$= \int_{-\infty}^{\infty} X(f_1) \underbrace{\int_{-\infty}^{\infty} H(f_2)\, \delta(f-f_1-f_2)\, df_2}_{\hookrightarrow\ H(f-f_1)}\, df_1$$

$$= \int_{-\infty}^{\infty} X(f_1)\, W(f-f_1)\, df_1 = X(f) * W(f).$$

Meaning: If a signal (daughter-signal) is a convoluted result from two signals (parent signals), then the fourier transform of the daughter-signal is the product of the fourier transform of its two parents. Additionaly, convoluting two frequencies together results in the fourier tranform of the product of their time-domain functions

2b. Compute a Fourier transform of the triangular function in both analytic and numeric ways (Note that this function is not a periodic)

$$F(x(t)) = \int_{-\infty}^{\infty} x(t)\, e^{-i2\pi f t}\, dt$$

$$= \int_{-1}^{0} (1+t)\, e^{-i2\pi f t}\, dt + \int_{0}^{1} (1-t)\, e^{-i2\pi f t}\, dt.$$

$$= \int_{-1}^{1} e^{-i2\pi f t}\, dt + \int_{-1}^{0} t\, e^{-i2\pi f t}\, dt + \int_{1}^{0} t\, e^{-i2\pi f t}\, dt.$$

$$\int e^{-i2\pi f t}\, dt = \frac{1}{-i2\pi f} \cdot e^{-i2\pi f t}$$

$$\int t\, e^{-i2\pi f t}\, dt = \frac{t\, e^{-i2\pi f t}}{-i2\pi f} - \frac{1}{-i2\pi f} \int e^{-i2\pi f t}\, dt$$

$$= \frac{t\, e^{-i2\pi f t}}{-i2\pi f} - \frac{1}{(-i2\pi f)^2} e^{-i2\pi f t} = \frac{e^{-i2\pi f t}}{(2\pi f)^2} - \frac{t\, e^{-i2\pi f t}}{i2\pi f}.$$

$$= \cancel{\frac{e^{i2\pi f} - e^{-i2\pi f}}{i2\pi f}} + 2 \times \left[\frac{1}{(2\pi f)^2} - 0\right] - \left(\frac{e^{i2\pi f}}{(2\pi f)^2} + \cancel{\frac{e^{i2\pi f}}{i2\pi f}}\right) - \left(\frac{e^{-i2\pi f}}{(2\pi f)^2} - \cancel{\frac{e^{-i2\pi f}}{i2\pi f}}\right)$$

$$= \frac{2 - e^{i2\pi f} - e^{-i2\pi f}}{(2\pi f)^2} = \frac{2 - 2\cos(2\pi f t)}{(2\pi f)^2}$$

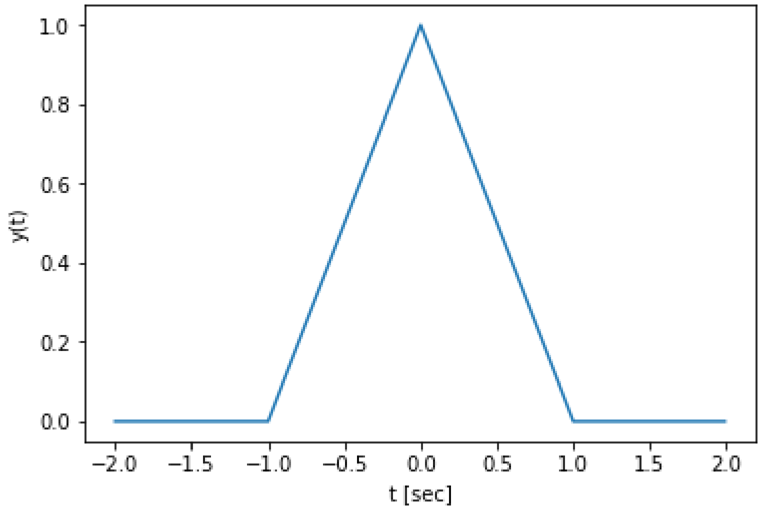
$$\because 1 - \cos(2\pi f t) = 2\sin^2(\pi f t). \quad \therefore = \frac{4\sin^2(\pi f t)}{(2\pi f)^2} = \frac{\sin^2(\pi f t)}{(\pi f)^2}$$

QUESTION:

Numerical means using fft or using integrate.quad???

```
# ------------------------------------------------------------ QUESTION: Numerical means using fft or using integrate.quad???
# assume its fft
# numerically
T = 0.01
y_t = lambda t: 1-abs(t) if abs(t)<1 else 0
t = np.arange(-2, 2, T)
y = list(map(y_t, t))
plt.plot (t, y)
plt.xlabel('t [sec]')
plt.ylabel('y(t)')
```

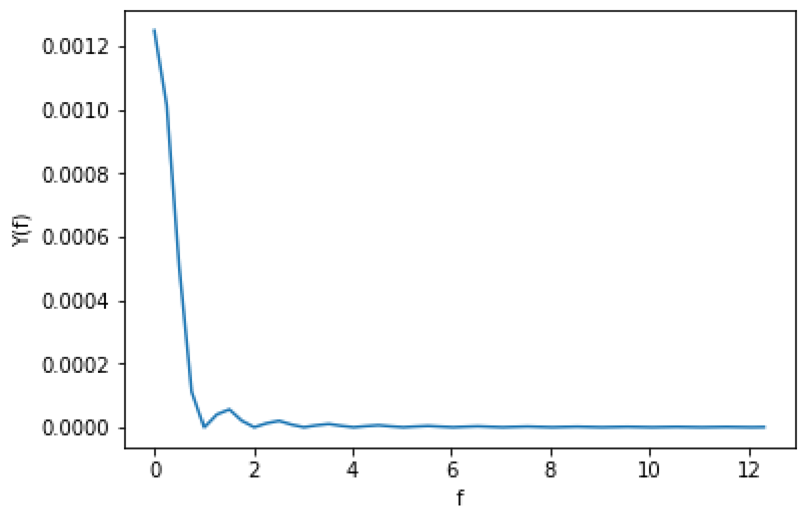
```
Text(0,0.5,'y(t)')
```




```
# fft
N = len(t)
xf = np.linspace(0.0, 1/(2*T), N/2)
fft_y = fft(y, N)/N
fft_y_clean = (2/N)*np.abs(fft_y[0:int(N/2)])
# fft_y_clean[fft_y_clean]
plt.plot(xf[0:50],fft_y_clean[0:50])
plt.xlabel('f')
plt.ylabel('Y(f)')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:3:
DeprecationWarning: object of type <class 'float'> cannot be safely interpreted
as an integer.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

```
Text(0,0.5,'Y(f)')
```



(P.s. This is just not right...?

2c. Please explain the result in (b) using your answers for Problem 1.

Problem one shows that the convolution of two 'box' functions is a triangle function. This problem shows that the fourier transform of a triangle function is a square root of a sinc function. It should be noted that the fourier transform of a 'box' function is a sinc function. Therefore, this case proves that the fourier tranform of a convoluted function is the product of the fourier transformed individual function.

Question 3

3a, What is the meaning of the following relationship in the lecture slide? Please explain it.

This means that the sampled signal in frequency domain is a periodic function.

3b. What is the meaning of the following relationship in the lecture slide? What issues are introduced by fourier transform a discrete sequence? Please answer this question using this graph.

This means that, in frequency domain, the sampled signals (the discrete squence) is the sum of the shifted version of the original continuous function. The transformation from continuous signals to the discrete sampling sequence is equivalent to multiply a "train" of delta functions.

Issue: if the frequency of sampling is not high enough, the "true" pattern from the original (continuous) signal cannot be isolated since the shifted pattern of the original will be "merged" with the original ones. If we want to isolate the frequency pattern of the original signal, then 1/2 of the sampling frequency need to be higher than the highest frequency of the original continuous signal.

3c. What is the difference between these two functions? Xs(f) and X(k)

X_k is the discrete fourier transform of a finite length of a signal. It transforms a sequence in time domain to another sequence in the frequency domain. However, Xs(f) is a continuous function in the frequency domain obtained from a discrete periodic signal.

Question 4

```
y1 = lambda t: np.exp(-2*abs(t))*(2*np.cos(2*np.pi*3*t)+6*np.cos(2*np.pi*6*t))
y2 = lambda t: np.exp(-0.3*abs(t))*
(10*np.cos(2*np.pi*5*t)+3*np.cos(2*np.pi*8*t))
```

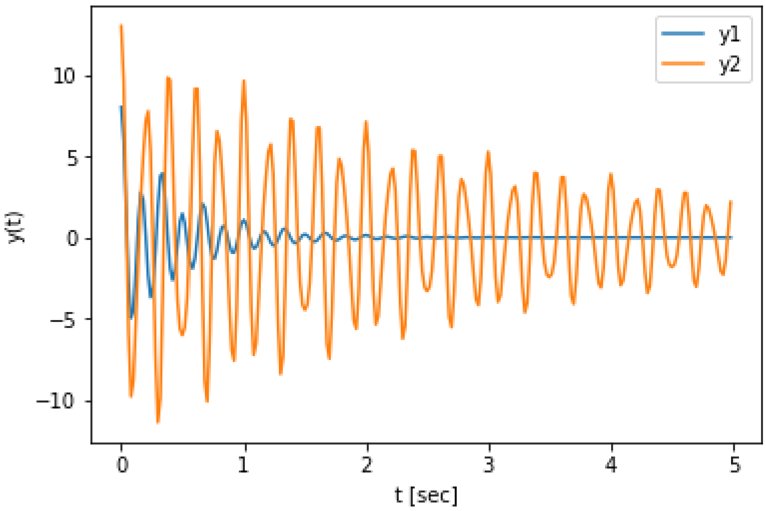
4a: z1 and z2 are discrete signals, which are obtained by digitizing y1(t) and y2(t) with a sampling rate of 50 Hz and collecting them for 5 seconds, respectively. Please plot z1 and z2 in the time domain (include a proper time axis).

```
N = 50*5
T = 1/50

t = np.arange(0, 5, 1/50)
seq1 = y1(t)
seq2 = y2(t)

plt.plot(t, seq1, label = 'y1')
plt.plot(t, seq2, label = 'y2')
plt.legend()
plt.xlabel('t [sec]')
plt.ylabel('y(t)')
```

```
Text(0,0.5,'y(t)')
```



4b. Perform the discrete Fourier transform of z1 and z2, and plot your graphs in the frequency domain (include a proper frequency axis). Plot only positive frequency signals.

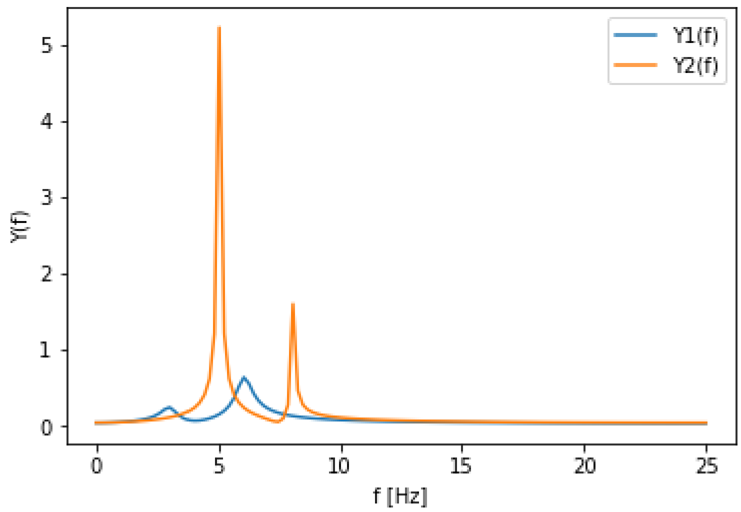
```
fft_y1 = fft(seq1)
fft_y2 = fft(seq2)

xf = np.linspace(0.0, 1/(2*T), N/2)
# ------------------------------- how to find f???
# (i copied from this youtube: https://www.youtube.com/watch?v=twi_KN1mL_E)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:4:
DeprecationWarning: object of type <class 'float'> cannot be safely interpreted
as an integer.
  after removing the cwd from sys.path.
```

```
plt.plot(xf, (2/N)*np.abs(fft_y1[0:int(N/2)]), label = 'Y1(f)')
plt.plot(xf, (2/N)*np.abs(fft_y2[0:int(N/2)]), label = 'Y2(f)')
plt.legend()
plt.xlabel('f [Hz]')
plt.ylabel('Y(f)')
```

```
Text(0,0.5,'Y(f)')
```



4c. Please compare the shape of the frequency curves of z1 and z2. Which frequency curve is thinner (more narrow)? For example, compare the frequency curve at f1 in both graphs. Which one is thinner? Please explain your answer. What makes the difference?

In frequency domain, the frequency spike for y2 is thinner (more narrow) and more sharp. From the plot for the time domain function, since the decay rate for y1 is larger, it was essentially "squeezed" in the time domain thus was widen the frequency domain.

Question 5

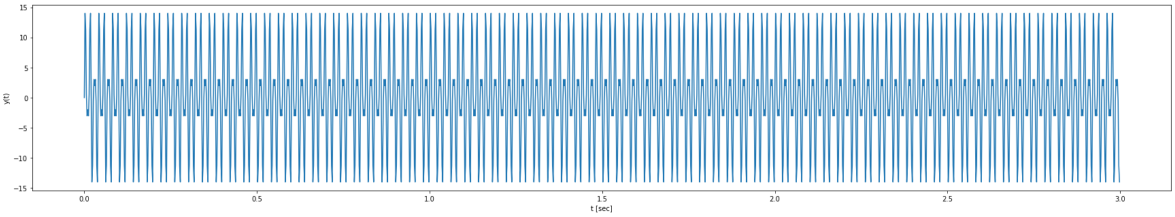
```
q5_y = lambda t:
3*np.sin(2*np.pi*25*t)+10*np.sin(2*np.pi*75*t)+5*np.sin(2*np.pi*125*t)
```

5a. y1 is a discrete signal, which is obtained by digitizing y(t) with a sampling rate of 500 Hz for 3 seconds. Please plot y1 in the time domain (include a proper time axis).

```
T = 1/500
N = 3/T
```

```
t = np.arange(0, N*T, T)
plt.figure(figsize=(30,5))
plt.plot(t, q5_y(t))
plt.xlabel('t [sec]')
plt.ylabel('y(t)')
```

```
Text(0,0.5,'y(t)')
```



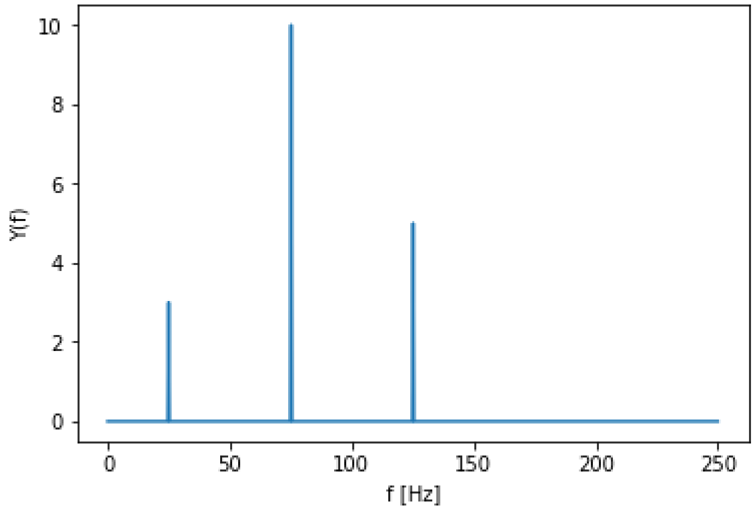
5b. Perform the discrete Fourier transform of y1 and plot your graph in the frequency domain (include a proper frequency axis). Plot only a positive frequency signal.

```
fft_y = fft(q5_y(t))
xf = np.linspace(0.0, 1/(2*T), N/2)

plt.plot(xf, (2/N)*np.abs(fft_y[0:int(N/2)]))
# plt.legend()
plt.xlabel('f [Hz]')
plt.ylabel('Y(f)')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
DeprecationWarning: object of type <class 'float'> cannot be safely interpreted
as an integer.
```

```
Text(0,0.5,'Y(f)')
```

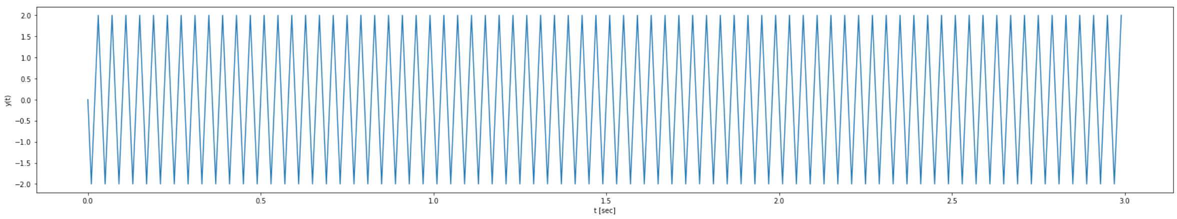


5c. y2 is a discrete signal, which is obtained by digitizing y(t) with a sampling rate of 100 Hz for 3 seconds. Please plot y2 in the time domain (include a proper time axis).

```
T = 1/100
N = 3/T
```

```
t = np.arange(0, N*T, T)
plt.figure(figsize=(30,5))
plt.plot(t, q5_y(t))
plt.xlabel('t [sec]')
plt.ylabel('y(t)')
```

```
Text(0,0.5,'y(t)')
```



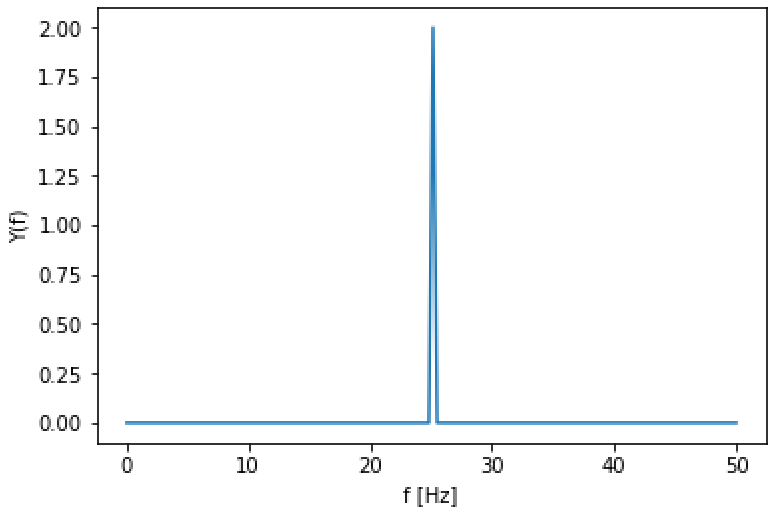
5d.Perform the discrete Fourier transform of y2 and plot your graph in the frequency domain (include a proper frequency axis). Plot only a positive frequency signal.

```
fft_y = fft(q5_y(t))
xf = np.linspace(0.0, 1/(2*T), N/2)

plt.plot(xf, (2/N)*np.abs(fft_y[0:int(N/2)]))
# plt.legend()
plt.xlabel('f [Hz]')
plt.ylabel('Y(f)')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
DeprecationWarning: object of type <class 'float'> cannot be safely interpreted
as an integer.
```

```
Text(0,0.5,'Y(f)')
```



5e. If you digitize a longer-duration signal (let's say 20 seconds) with a sampling rate of 100 Hz, can you measure and extract all frequencies contained in the original signal, y(t)? Please explain your answer.

No. Since the Nyquest frequency is still lower than the frequencies we want to extract.

5f. If you digitize the signal with a sampling rate of 105 Hz for 3 seconds, can you measure and extract all frequencies contained in the original signal, y(t)? please explain your answer.

No. Since the Nyquest frequency is still lower than the frequencies we want to extract.

5g. If you digitize the signal with a sampling rate of 251 Hz for 3 seconds, can you measure and extract all frequencies contained in the original signal, y(t)? please explain your answer.

Yes. The Nyquest frequency of 251 Hz is 125.5 Hz, which is higher than the frequencies of the original signals.

Question 6

6a. Load data1.mat and plot the acceleration signal in a z direction (zvib). What is the main frequency of this wave? The sampling frequency is 10.1355 Hz.

```
mat1 = io.loadmat('data1.mat')
mat2 = io.loadmat('data2.mat')
```

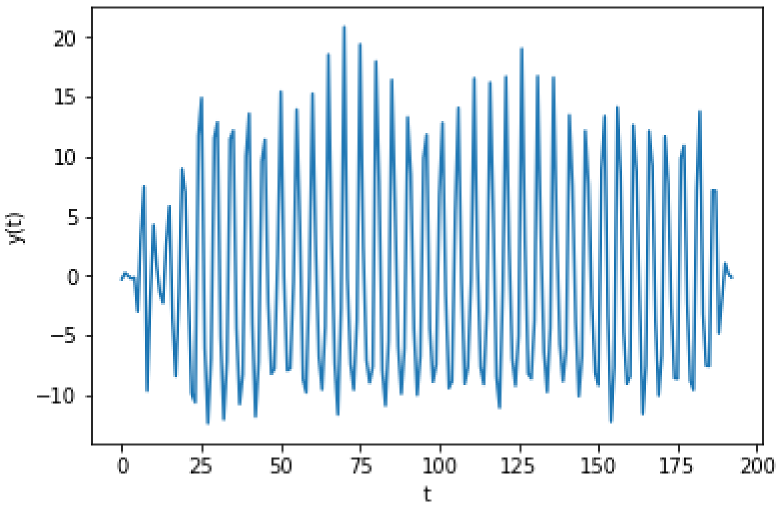
```
fs1 = mat1['fs'][0][0]
zvib1 = np.array(mat1['zvib'])
fs2 = mat2['fs'][0][0]
zvib2 = np.array(mat2['zvib'])
```

```
arr1 = []
for i in range(0, len(zvib1)):
    arr1.append(zvib1[i][0])

arr2 = []
for i in range(0, len(zvib2)):
    arr2.append(zvib2[i][0])
```

```
plt.plot(zvib1)
plt.xlabel('t')
plt.ylabel('y(t)')
```

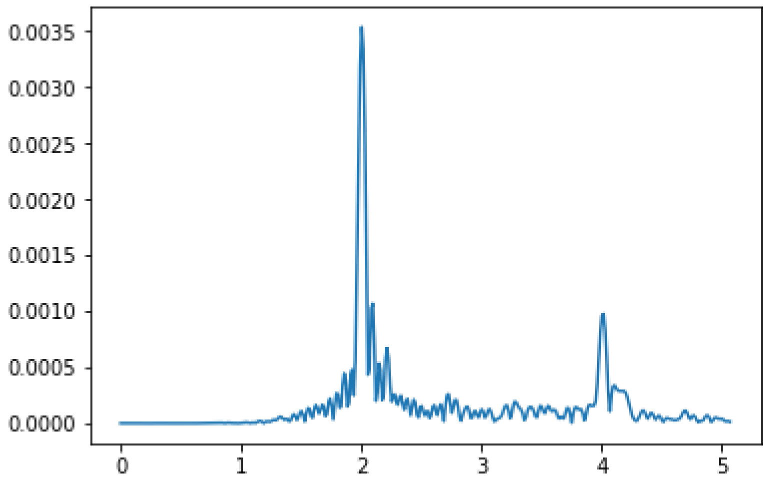
```
Text(0,0.5,'y(t)')
```




```
# signal 1 fft
n = len(arr1)*4
fft_1 = abs(fft(arr1, n))/n
xf1 = np.linspace(0.0, fs1/2, n/2)
plt.plot(xf1, (2/n)*np.abs(fft_1[0:int(n/2)]))
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:4:
DeprecationWarning: object of type <class 'float'> cannot be safely interpreted
as an integer.
  after removing the cwd from sys.path.
```

```
[<matplotlib.lines.Line2D at 0x1cf0b32fc50>]
```

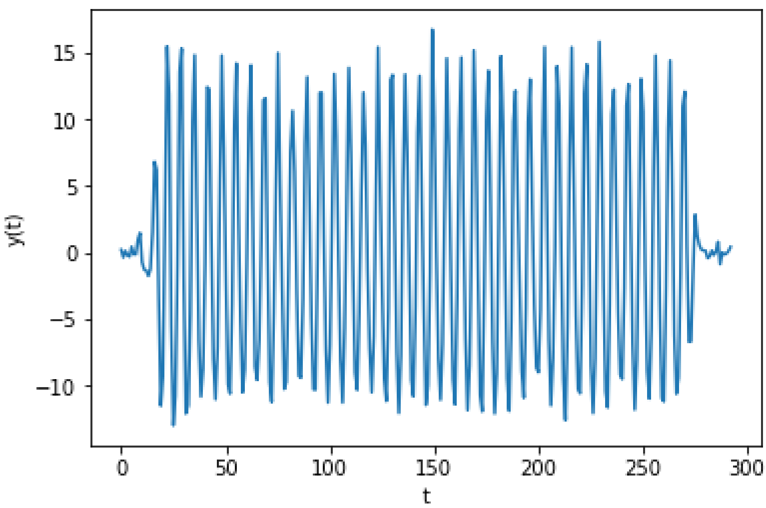


The main frequency is 2 and 4 Hz.

6b. Load data2.mat and plot the acceleration signal in a z direction (zvib). What is the main frequency of this wave? The sampling frequency is 10.1192 Hz.

```
plt.plot(zvib2)
plt.xlabel('t')
plt.ylabel('y(t)')
```

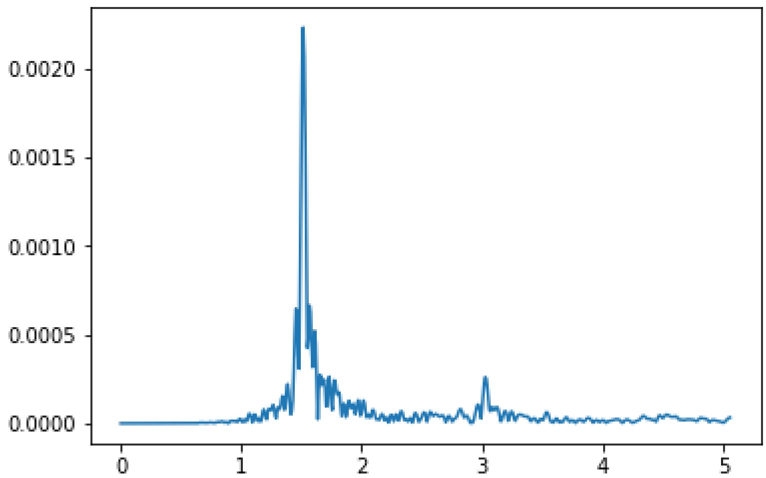
```
Text(0,0.5,'y(t)')
```



```python
# signal 2 fft
n = len(arr2)*4
fft_2 = abs(fft(arr2, n))/n
xf2 = np.linspace(0.0, fs2/2, n/2)
plt.plot(xf2, (2/n)*np.abs(fft_2[0:int(n/2)]))
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:4:
DeprecationWarning: object of type <class 'float'> cannot be safely interpreted
as an integer.
  after removing the cwd from sys.path.
```

```
[<matplotlib.lines.Line2D at 0x1cf0b3e6ef0>]
```



The main frequency is 1.5 Hz.