```
from IPython import get_ipython
get_ipython().magic('reset -sf')
#get_ipython().magic('cls')
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import freqz,dlti
from scipy.signal import butter, lfilter
from scipy import signal
import scipy.io.wavfile as wav
```

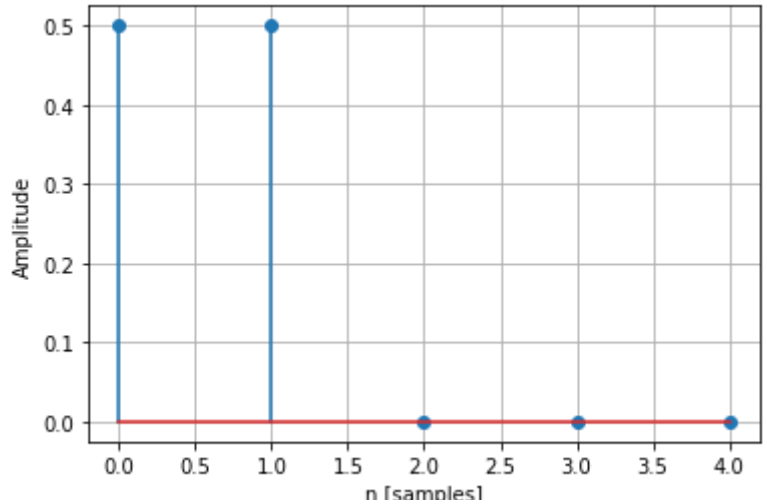
- Réponse fréquentielle

Structure de module |H(f)| et l'argument (la phase) (f) = Arg(H(f)) à l'aide de la fonction freqz sous matlab

```
#---------------------App1 RIF -------------------
num=[0.5,0.5]
den=[1,0]
h=signal.dlti(num,den)
t, h = signal.dimpulse(h, n=5)

plt.Figure()
plt.stem(t,np.squeeze(h))
plt.grid()
plt.xlabel('n [samples]')
plt.ylabel('Amplitude')
f, h = signal.freqz(num,den,worN=512,fs=1000)
plt.figure()
plt.title('Digital filter frequency response')
plt.plot(f,np.abs(h))
plt.xlabel('Amplitude Response')
plt.ylabel('Frequency (Hz)')
plt.grid()
plt.show()
```

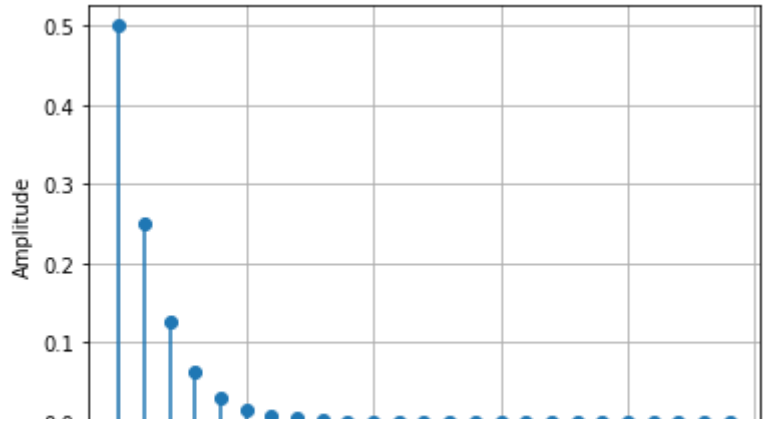
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: UserWarning: In Ma
```




```
#---------------------App2 RII -------------------
num=[0.5,0]
den=[1,-0.5]
h=signal.dlti(num,den)
t, h = signal.dimpulse(h, n=25)

plt.Figure()
plt.stem(t,np.squeeze(h))
plt.grid()
plt.xlabel('n [samples]')
plt.ylabel('Amplitude')
f, h = signal.freqz(num,den,worN=512,fs=1000)
plt.figure()
plt.title('Digital filter frequency response')
plt.plot(f,np.abs(h))
plt.xlabel('Amplitude Response')
plt.ylabel('Frequency (Hz)')
plt.grid()
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: UserWarning: In Matpl
```




```
#---------------------Application -------------------
t = np.linspace(0, 2, 201)
x = (np.sin(2*np.pi*1*t)+ 4*np.sin(2*np.pi*2.0*t))
rng = np.random.default_rng()
xn = x + rng.standard_normal(len(t))
b, a = signal.butter(3 ,0.05)
z1 = signal.filtfilt(b, a, xn)
plt.figure()
plt.subplot(311)
plt.plot(t, x, 'k',label='clean signal')
plt.legend(('noisy signal'), loc = 'best')

plt.figure()
plt.subplot(312)
plt.plot(t, xn, 'b')
plt.legend(('noisy signal'), loc = 'best')

plt.figure()
plt.subplot(313)
plt.plot(t, z1, 'r--')
plt.legend(('filtfilt'), loc = 'best')
plt.grid(True)
```