

## Filtrage d'un signal sinusoïdal par un filtre passe-bas du 1<sup>er</sup> ordre

Code Capytale : f56f-6479538

```
Entrée[7]: import numpy as np
import matplotlib.pyplot as plt
from cmath import phase
```

### Tracé du signal d'entrée sur 2 périodes

```
Entrée[8]: f=100
t = np.linspace(0, 2/f, 2000) # 2 périodes
e=4+2*np.cos(2*np.pi*f*t)
plt.plot(t,e,'b',label='e(t)')
plt.xlabel('t(s)')
plt.grid()
plt.show()
```

### Tracé du spectre en amplitude

```
Entrée[9]: plt.figure(2)
plt.vlines(0,0,4,'b',label='e(t)') # Tracé de La composante continue
plt.vlines(100,0,2,'b') # Tracé de La composante fondamentale
plt.xlabel('f(Hz)')
plt.title('spectres')
plt.ylabel('amplitude(V)')
plt.show()
```

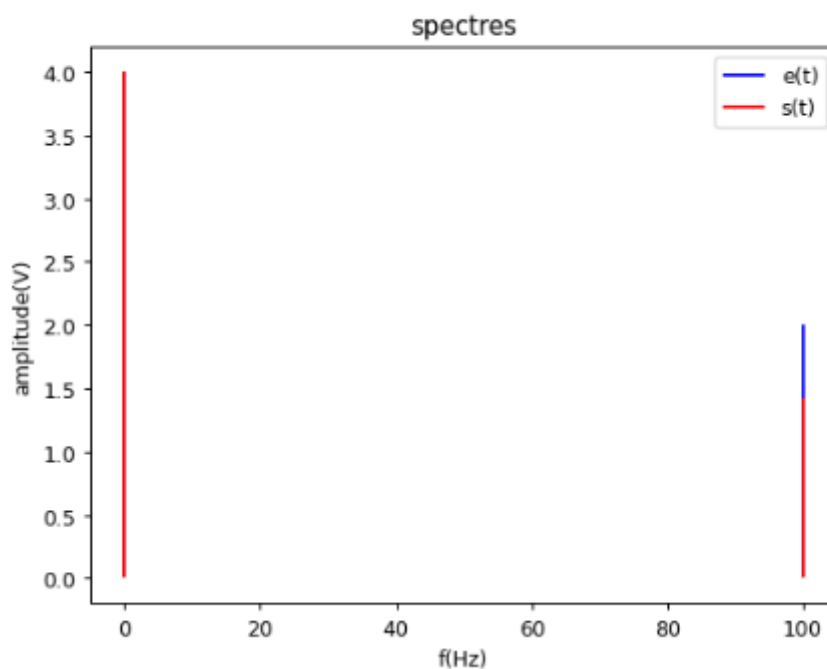
### Définir la fonction de transfert

```
Entrée[10]: def H(f,f_c):
return 1/(1+1j*f/f_c)
```

### Tracé du spectre de sortie

```
Entrée[11]: S0 =4*abs(H(0,100)) # Valeur moyenne du signal de sortie
S1 =2*abs(H(100,100)) # Amplitude de La composante fondamentale
plt.figure(2) # Pour superposer au spectre d'entrée
plt.vlines(0,0,S0,'r',label='s(t)') # Tracé de La composante continue
plt.vlines(100,0,S1,'r') # Tracé de La composante fondamentale
plt.legend()
plt.show()
```

Figure 2



## Tracé du signal de sortie

```
Entrée[12]: plt.figure(1) # Pour superposer au signal d'entrée
phi1=phase(H(100,100))
t = np.linspace(0, 2/100, 2000) # 2 périodes
s=S0+S1*np.cos(2*np.pi*f*t +phi1)
plt.plot(t,s,'r',label='s(t)')
plt.grid()
plt.legend()
plt.show()
```

Figure 1

