

▼ TP2 - Génération de signaux numériques

Manipulation

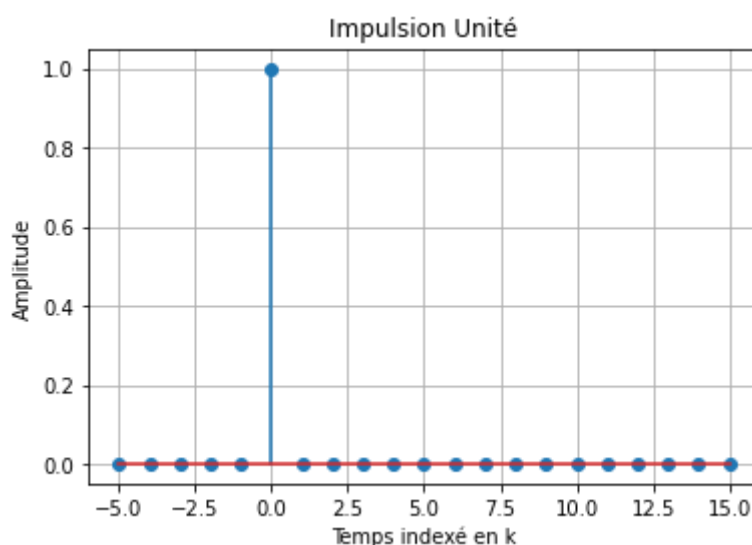
```
import numpy as np
import matplotlib.pyplot as plt
```

▼ 1. Génération des signaux numériques élémentaires

Génération et traçage de signal $d(k)$ dans le domaine temporel de k compris entre -5 et 15

```
k = np.arange(-5,16)
print(k)
d0 = np.zeros((21))
d0[5]=1
plt.figure()
plt.stem(k,d0) # Affichage numerique
plt.title('Impulsion Unité')
plt.ylabel('Amplitude')
plt.xlabel('Temps indexé en k')
plt.grid()
# Représentation temporelle de l'impulsion unité
```

[-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
 /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: UserWarning: In Matp]



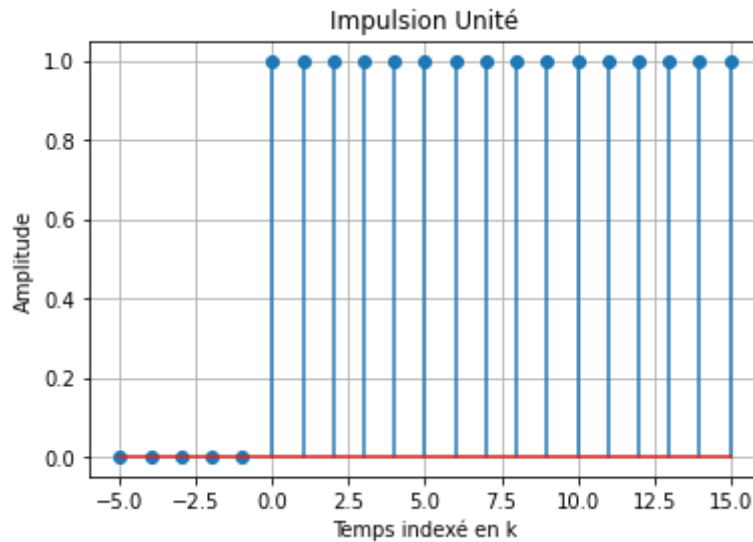
Génération et traçage de signal échlon $u(k)$ dans le domaine temporel de k compris entre -5 et 15 à l'aide de la fonction 'ones' de python.

```

k = np.arange(-5,16)
U= np.zeros(21)
U[5:21]=1
plt.figure()
plt.stem(k,U)
plt.title('Impulsion Unité')
plt.ylabel('Amplitude')
plt.xlabel('Temps indexé en k')
plt.grid()

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: In Matplotlib, the default font is sans-serif. The default font family is serif. The default font size is 10. The default font style is normal. The default font weight is normal. The default font color is black. The default font family is sans-serif. The default font size is 10. The default font style is normal. The default font weight is normal. The default font color is black.



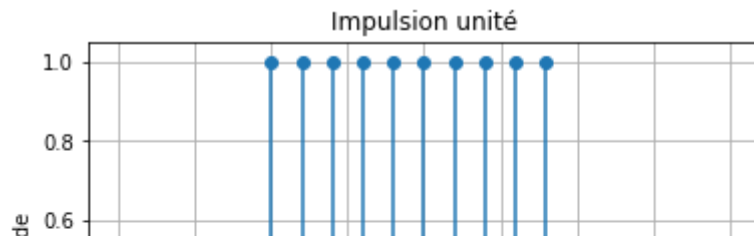
Génération et traçage de signal numérique $\text{rect}_{10}(k)$ dans le domaine temporel de k compris entre -5 et 15

```

k = np.arange(-5,16)
m= np.zeros(21)
m[5:15]=1
plt.figure()
plt.stem(k,m)
plt.title('Impulsion unité')
plt.ylabel('Amplitude')
plt.xlabel('Temps indexé en k')
plt.grid()

```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: In Matp]
"""
```

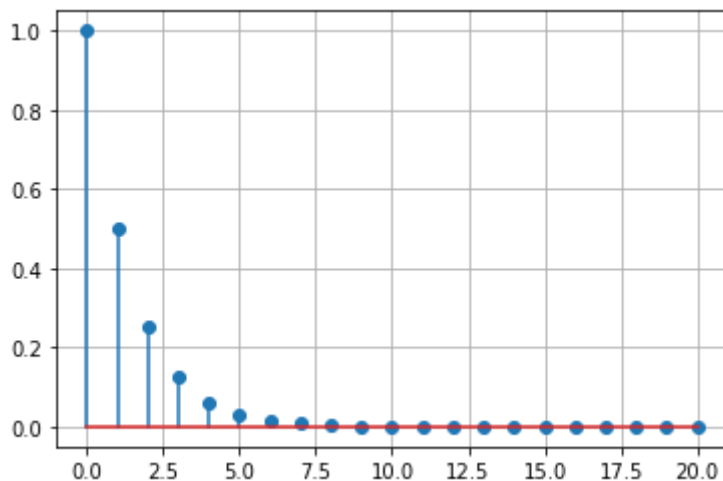


Génération et traçage de signal exponentiel $s_1(k)$ dans le domaine de k compris entre 0 et 20 avec $\alpha=1$ et $a=0.5$



```
# signal numérique exponentiel
k = np.arange(21)
S = 0.5**k
plt.figure()
plt.stem(k,S)
plt.grid()
# Représentation temporelle de l'échelon unité
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: In Matp]
"""
```

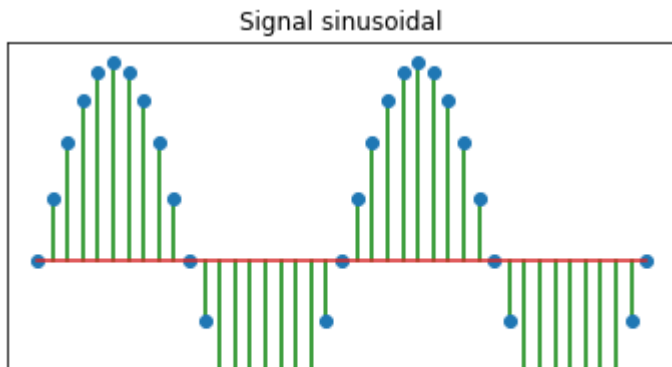


Double-cliquez (ou appuyez sur Entrée) pour modifier

```
# signal numérique sinus
k = np.arange(0,41)
N=20 ; A=2 ; phase = 0
s1=A*np.sin(2*np.pi*k/N-phase)
plt.stem(k,s1,'g')
plt.xticks([], plt.yticks([]))
plt.title('Signal sinusoidal')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: In Matp]
```

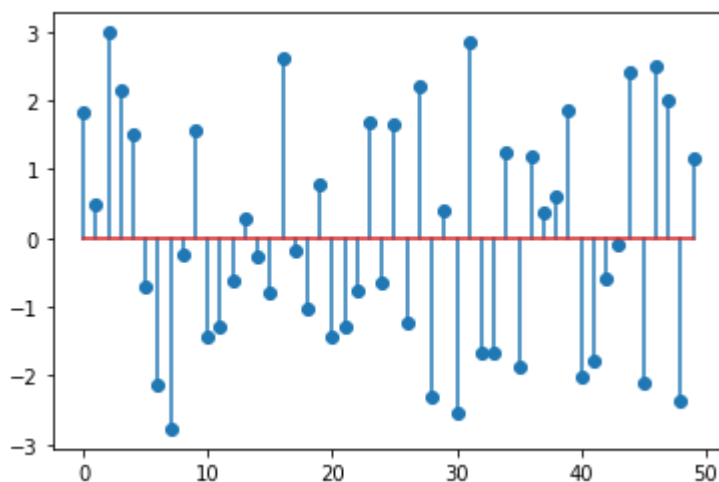
```
Text(0.5, 1.0, 'Signal sinusoidal')
```



```
# signaux numériques bruit rand
N=50
k=np.arange(N)
a,b=-3,3
bruit=(b-a)*np.random.rand(1,N)+a
bruit=bruit.flatten()
plt.figure()
plt.stem(k,bruit)
```

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

```
<StemContainer object of 3 artists>
```

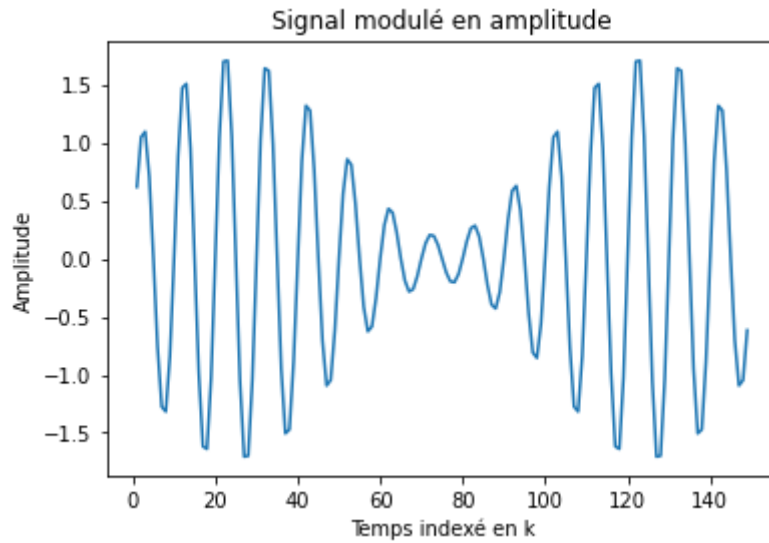


3. Génération des signaux numériques aléatoires

```
# signal de modulation d'amplitude
k=np.arange(1,150)
m=0.8; fh=0.1; fl=0.01;
xh=np.sin(2*np.pi*fh*k);
xl=np.sin(2*np.pi*fl*k);
y=(1+m*xl)*xh;
plt.figure()
plt.plot(k,y)
plt.grid; plt.ylabel('Amplitude');
```

```
plt.xlabel('Temps indexé en k')  
plt.title('Signal modulé en amplitude')
```

```
Text(0.5, 1.0, 'Signal modulé en amplitude')
```



✓ 0 s terminée à 22:05

