

## PROGRAMMER EN PYTHON

### FICHE N°10 : SOMMES DE DEUX SINUSOÏDES SYNCHRONES

La sommation de deux sinusoïdes synchrones déphasées permet de visualiser le phénomène d'interférences en un point donné.

#### Capacité numérique mise en œuvre

Représenter, à l'aide d'un langage de programmation, la somme de deux signaux sinusoïdaux périodiques synchrones en faisant varier la phase à l'origine de l'un des deux.

### Présentation

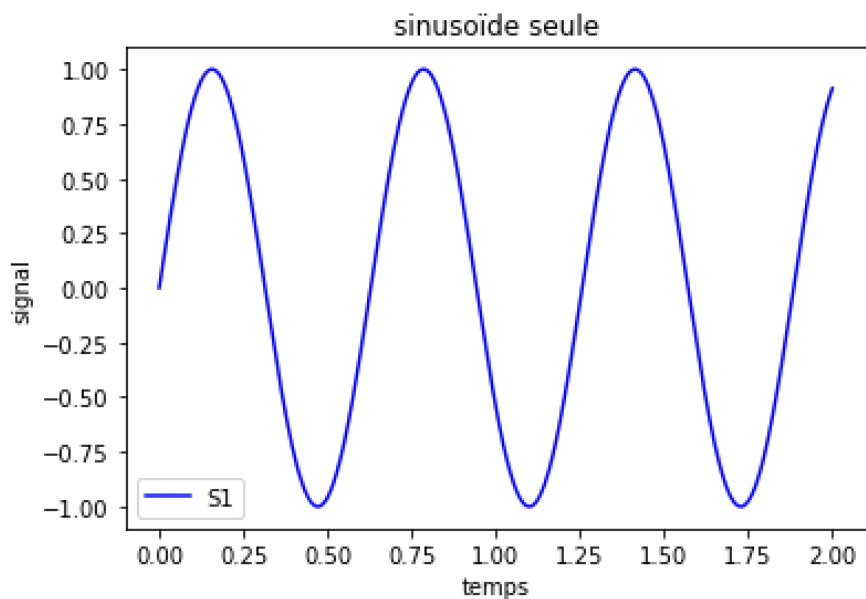
On simule deux signaux sinusoïdaux de même amplitude (on pourra faire varier l'amplitude pour aller plus loin) et de même période. On observe la somme de ces deux signaux pour divers déphasages. Dans la proposition ci-dessous on pourra observer l'effet de différentes valeurs du déphasage en modifiant « à la main » avant d'utiliser la fonction « for ».

### Construction d'une sinusoïde

Le script est le suivant :

```
9 import numpy as np
10 import matplotlib.pyplot as plt
11
12 w=10
13 lt=np.linspace(0,2,1000)
14 def s1(t):
15     return np.sin(w*t)
16
17
18 """
19 plt.plot(lt,s1(lt),'b-',label='s1')
20
21 plt.xlabel("temps")
22 plt.ylabel("signal")
23 plt.title("sinusoïde seule")
24 plt.legend()
25 plt.show()
26
```

Il renvoie à :

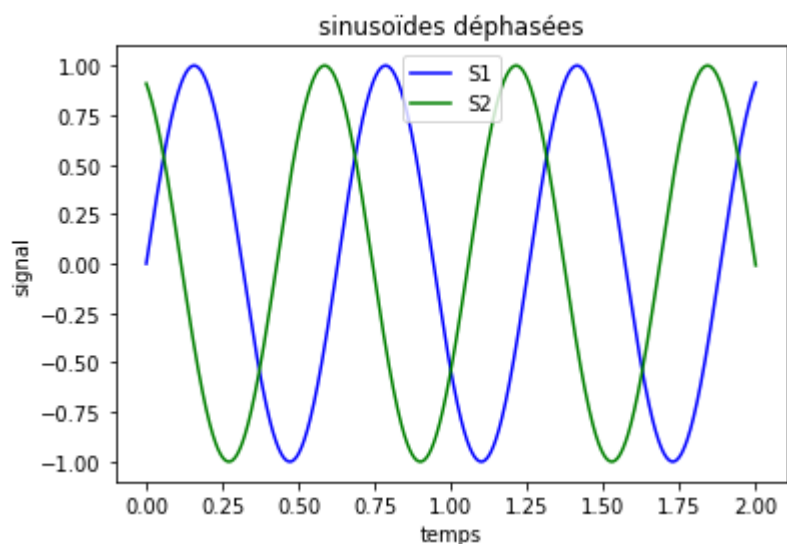


### Sommation de deux sinusoïdes déphasées

On peut ensuite ajouter une autre sinusoïde déphasée et tracer la somme des deux sinusoïdes pour une certaine valeur du déphasage :

```
27 #%%  
28 phi=2  
29 def s2(t):  
30     return np.sin(w*t+phi)  
31 plt.plot(t,s1(t),'b-',label='S1')  
32 plt.plot(t,s2(t),'g-',label='S2')  
33 plt.xlabel("temps")  
34 plt.ylabel("signal")  
35 plt.title("sinusoïdes déphasées")  
36 plt.legend()  
37  
38 plt.show()
```

Ce qui donne :



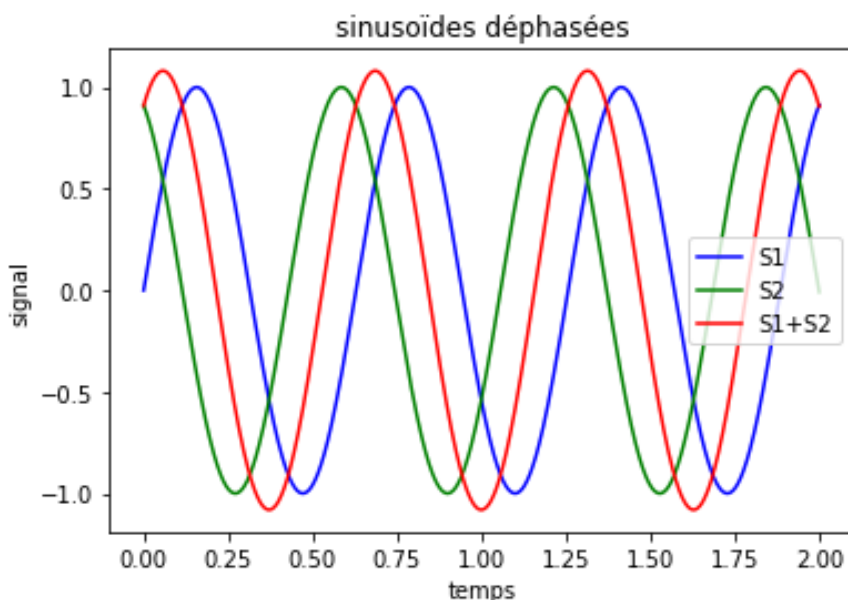
On fait la somme

```

40 phi=2
41 def somme(t):
42     return s1(t)+s2(t)
43
44 plt.plot(lt,s1(lt), 'b-',label='S1')
45 plt.plot(lt,s2(lt), 'g-',label='S2')
46 plt.plot(lt,somme(lt), 'r-',label='S1+S2')
47 plt.xlabel("temps")
48 plt.ylabel("signal")
49 plt.title("sinusoïdes déphasées")
50 plt.legend()
51 plt.show()

```

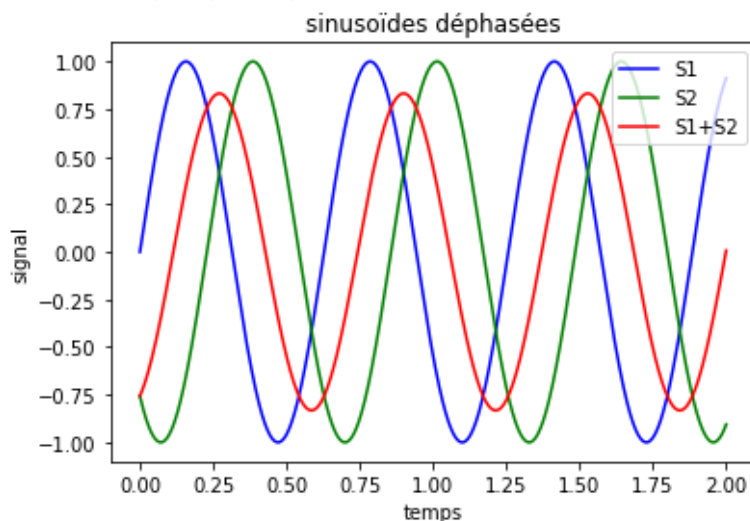
Ce qui donne :



### Évolution de la sommation avec le déphasage

On peut modifier la valeur de  $\phi$  pour visualiser l'effet du déphasage sur la somme :

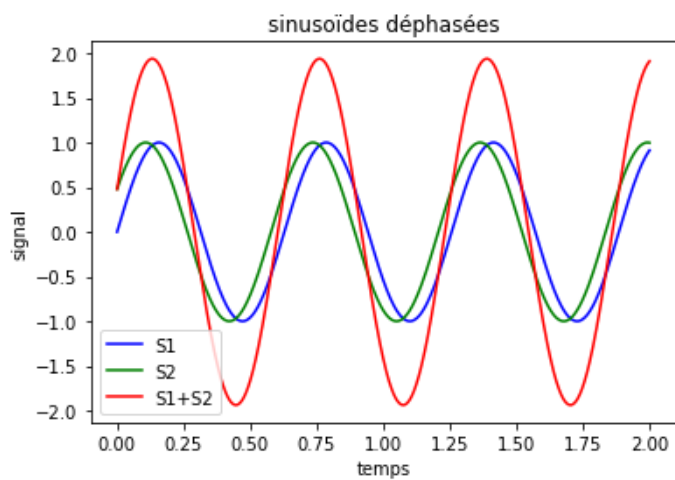
#### Par exemple pour $\phi = 4$



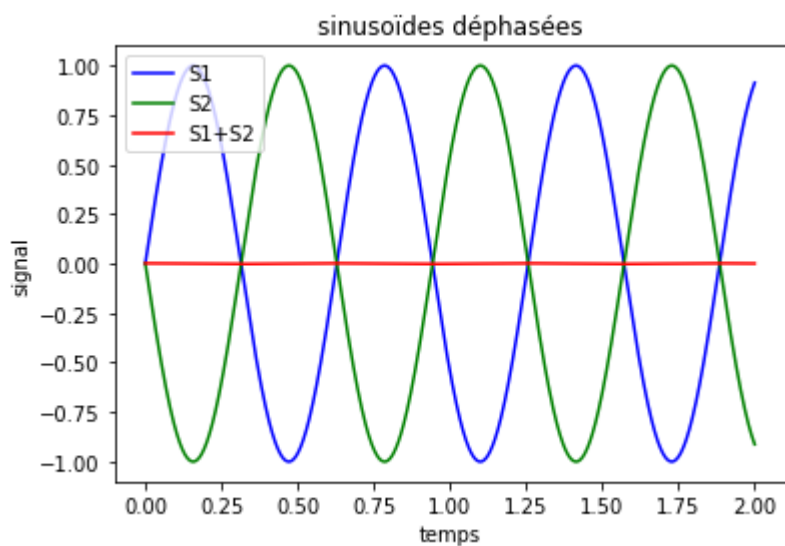
Retrouvez éducol sur



### Par exemple pour $\phi=0.5$



### Par exemple pour $\phi=3.14$



Retrouvez éducol sur

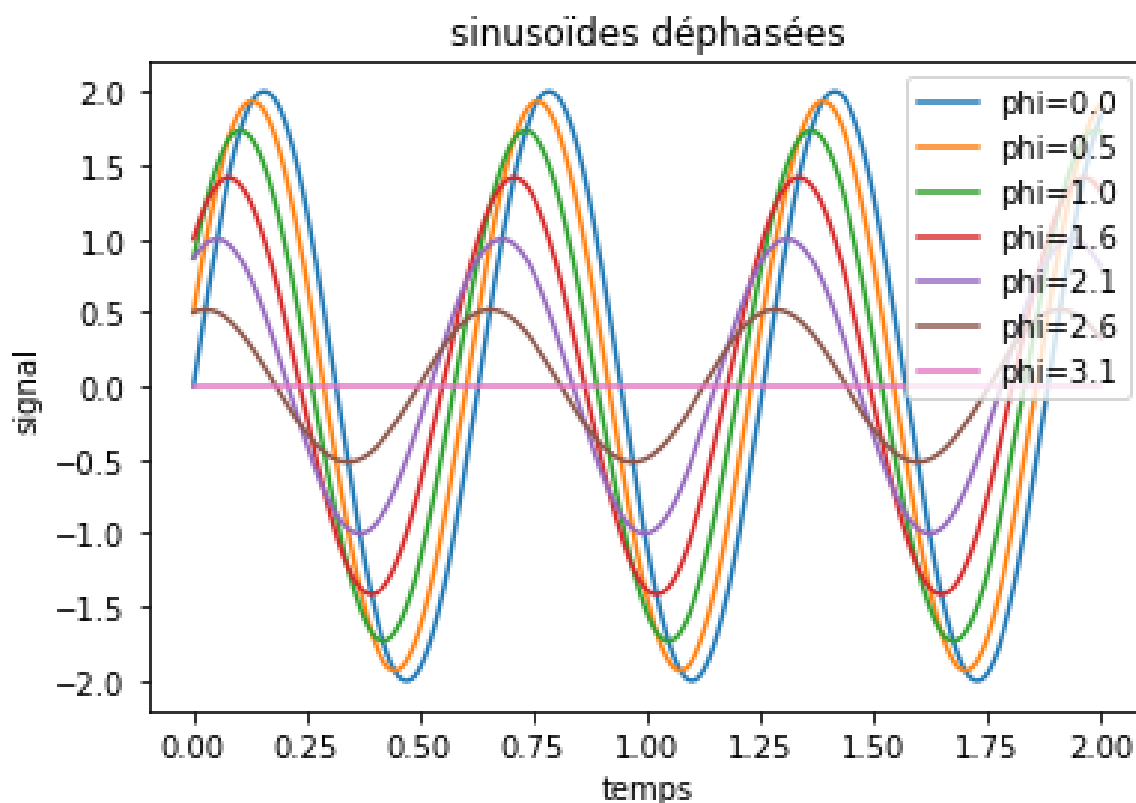


## Visualiser l'effet du déphasage sur la somme

On peut enfin utiliser une boucle **for** pour visualiser l'effet du déphasage sur la somme :

```
52 ###
53 phi=np.pi/2
54
55 plt.plot(lt,somme(lt),'r-',label='S1+S2')
56 plt.xlabel("temps")
57 plt.ylabel("signal")
58 plt.title("sinusoïdes déphasées")
59 plt.legend()
60 plt.show()
61 ###
62 lphi=np.linspace(0,np.pi,7)
63 for phi in lphi:
64     plt.plot(lt,somme(lt),label='phi='+str(round(phi,1)))
65     plt.xlabel("temps")
66     plt.ylabel("signal")
67     plt.title("sinusoïdes déphasées")
68     plt.legend()
69     plt.show()
70
```

Ce qui donne :



Retrouvez éducol sur



**Conclusion**

Le déphasage entre les deux sinusoides a un impact direct et visible sur la valeur de leur somme. Ainsi, deux sinusoides non nulles déphasées peuvent tout à fait avoir une somme nulle : on illustre ici le phénomène d'interférences destructives.