

# Fonction onde sinusoidale

August 22, 2019

## 1 Création d'une fonction onde avec Jupyter

Définissons une fonction sinusoïdale à partir de ses éléments. Commençons par établir les paramètres de l'onde.

```
In [1]: import numpy as np
import matplotlib.pyplot as plt

A = 1.0
print ("amplitude A = %5f m" % A)
f = 2.0
print ("fréquence f = %.5f 1/s" % f)
phi = 0.0
print ("phase phi = %.5f rad" % phi)
c = 4.0
print ("vitesse c = %.5f m/s" % c)
```

```
amplitude A = 1.000000 m
fréquence f = 2.000000 1/s
phase phi = 0.000000 rad
vitesse c = 4.000000 m/s
```

Nous pouvons calculer d'autres paramètres à partir des précédentes.

```
In [2]: omega = 2.0 * np.pi * f
print ("pulsation omega = %.5f rad / s" % omega)
T = 1.0 / f
print ("période T = %.5f s" % T)
lambda = c * T
print ("longueur d'onde lambda = %.5f m" % lambda)
k = 2 * np.pi / lambda
print ("nombre d'onde k = %.f rad / m" % k)
```

```
pulsation omega = 12.56637 rad / s
période T = 0.50000 s
longueur d'onde lambda = 2.00000 m
nombre d'onde k = 3 rad / m
```

Avec ces données, allons représenter notre onde aux instants  $t = 0, 0.1, 0.2, 0.3$  s. Rappelons que une période  $T$  est égale à 0.5 s.

```
In [3]: x = np.arange(-1.0, 5.0, 0.01)

t = 0.0
psi_00 = A * np.cos(omega * t - k * x + phi)
plt.plot(x, psi_00, 'yellow')

t = 0.1
psi_01 = A * np.cos(omega * t - k * x + phi)
plt.plot(x, psi_01, 'orange')

t = 0.2
psi_02 = A * np.cos(omega * t - k * x + phi)
plt.plot(x, psi_02, 'red')

t = 0.3
psi_03 = A * np.cos(omega * t - k * x + phi)
plt.plot(x, psi_03, 'black')

plt.legend(['t = 0.0', 't = 0.1', 't = 0.2', 't = 0.3'])

plt.show()
```

