

1. FONCTIONS

Corrigé F3

```

"""
oscillation libres amorties
définition de la fonction suivant la valeur de Q
"""

#chargement module
import numpy as np

# paramètres connus
xeq=-0.5           # position d'équilibre
T=0.8             # période propre
x0=1             # position initiale
v0=0             # vitesse initiale

# calcul de la pulsation propre
w = 2*np.pi/T

# facteur de qualité
Q = float(input("Valeur du facteur de qualité (Q) : "))

# fonction
if Q<=0:
    print("Q ne peut pas être négatif ou nul")

elif Q>0.5 : # amortissement faible
    wp = w*np.sqrt(1-1/(4*Q**2)) # pseudopulsation
    A = x0-xeq # coeff du cosinus
    B = (v0 + w*A/(2*Q))/wp # coeff du sinus
    def x(t):
        return xeq + np.exp(-w*t/(2*Q))*(A*np.cos(wp*t) + B*np.sin(wp*t))

elif Q<0.5 : # amortissement fort
    d1 = w*(1/(2*Q)-np.sqrt(1/(4*Q**2)-1)) # delta 1
    d2 = w*(1/(2*Q)+np.sqrt(1/(4*Q**2)-1)) # delta 2
    A = (d2*(x0-xeq)+v0)/(d2-d1) # coeff de la 1e exp
    B = (d1*(x0-xeq)+v0)/(d1-d2) # coeff de la 2e exp
    def x(t):
        return xeq + A*np.exp(-d1*t) + B*np.exp(-d2*t)

else :
    #elif Q==0.5 : # amortissement critique
    A = v0+w*(x0-xeq) # coeff A
    B = x0-xeq # coeff B
    def x(t):
        return xeq+(A*t+B)*np.exp(-w*t)

```