

On étudie le système représenté figure 2 dans lequel deux métronomes reposent sur une planche légère, elle-même posée sur des cannettes de soda vides. Un métronome possède un mécanisme interne d'horlogerie se composant entre autres d'un ressort que l'on remonte manuellement. Il permet d'entretenir les oscillations et de battre à un rythme régulier. La fréquence des battements est ajustable au moyen d'une masse placée sur le bras du pendule. Plus on l'éloigne du pivot, plus la fréquence diminue. La description de ces oscillations du métronome peut se faire en utilisant le modèle du pendule simple, de longueur l et de masse m , comme représenté figure 1, auquel on ajoute un terme dit de Van Der Pol. L'équation différentielle régissant l'angle θ étant alors :

$$\underbrace{\frac{d^2\theta}{dt^2} - \varepsilon \left(1 - \left(\frac{\theta}{\theta_0} \right)^2 \right)}_{\text{Terme de Van Der Pol}} + \omega_0^2 \sin \theta = 0$$

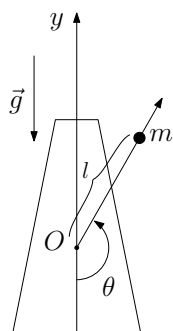


FIGURE 1 – Modélisation du métronome

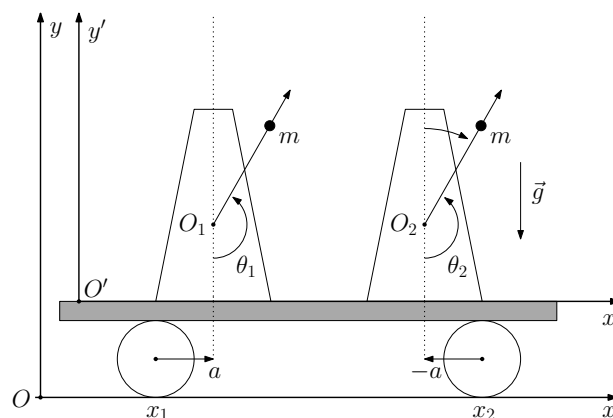


FIGURE 2 – Montage étudié

- I.– On se place dans le cas du roulement sans glissement des cannettes (dont on néglige le moment d'inertie). En considérant que le système (S) constitué par les métronomes, la planche et les cannettes peut être considéré comme pseudo isolé suivant (Ox) (on néglige la projection des forces extérieures sur (Ox)), montrer, moyennant un choix judicieux de l'origine du repère $(O, \vec{u}_x)$, que la position x de la planche vérifie :

$$x = \frac{-ml}{2m + M} (\sin \theta_1 + \sin \theta_2)$$

On supposera qu'à $t = 0$, les différents éléments du système (S) sont sans vitesse.

- II.– En appliquant le théorème du moment cinétique à chacun des métronomes dans le référentiel lié à la planche, montrer que les équations différentielles vérifiées par θ_1 et θ_2 sont :

$$\begin{cases} \frac{d^2\theta_1}{dt^2} + \omega_0^2 \sin \theta_1 - \varepsilon \left(1 - \left(\frac{\theta_1}{\theta_0} \right)^2 \right) - \frac{m}{2m + M} \cos \theta_1 \frac{d^2}{dt^2} (\sin \theta_1 + \sin \theta_2) = 0 \\ \frac{d^2\theta_2}{dt^2} + \omega_0^2 \sin \theta_2 - \varepsilon \left(1 - \left(\frac{\theta_2}{\theta_0} \right)^2 \right) - \frac{m}{2m + M} \cos \theta_2 \frac{d^2}{dt^2} (\sin \theta_1 + \sin \theta_2) = 0 \end{cases}$$

Commenter.

- III.– Dans l'approximation des petits angles et en négligeant le terme de Van Der Pol, déterminer les pulsations propres du système.
- IV.– Commenter les figures suivantes (obtenues pour $\theta_0 = 0.2 \text{ rad}$, $\omega_0 = 1 \text{ rad}\cdot\text{s}^{-1}$, $\varepsilon = 5 \times 10^{-4}$ et $\dot{\theta}_1(0) = \dot{\theta}_2(0) = 0$). Préciser le rôle du terme de Van Der Pol dans la synchronisation des pendules.

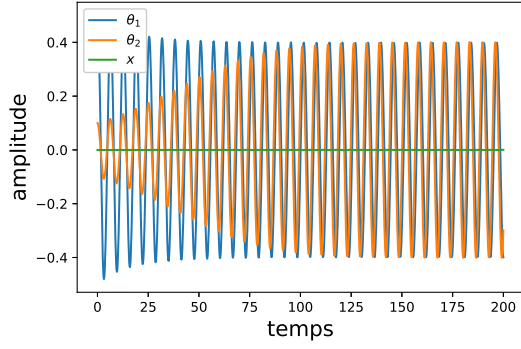


FIGURE 3 – $\theta_1(0) = 0.5 \text{ rad}$, $\theta_2(0) = 0.1 \text{ rad}$

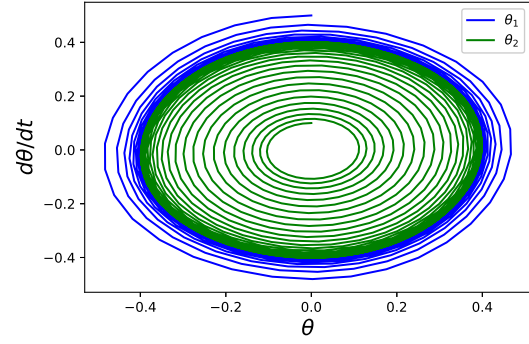


FIGURE 4 – $\theta_1(0) = 0.5 \text{ rad}$, $\theta_2(0) = 0.1 \text{ rad}$

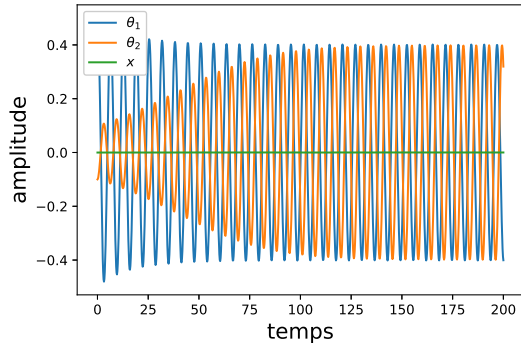


FIGURE 5 – $\varepsilon = 0.05$, $\theta_1(0) = 0.5 \text{ rad}$, $\theta_2(0) = -0.1 \text{ rad}$

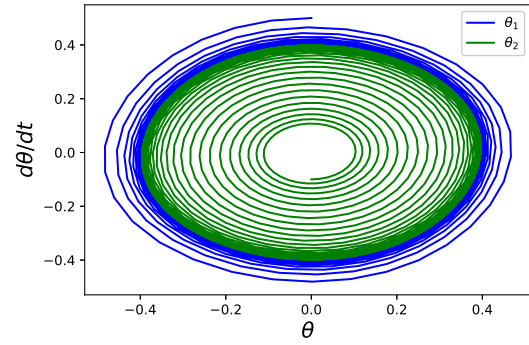


FIGURE 6 – $\varepsilon = 0.05$, $\theta_1(0) = 0.5 \text{ rad}$, $\theta_2(0) = -0.1 \text{ rad}$

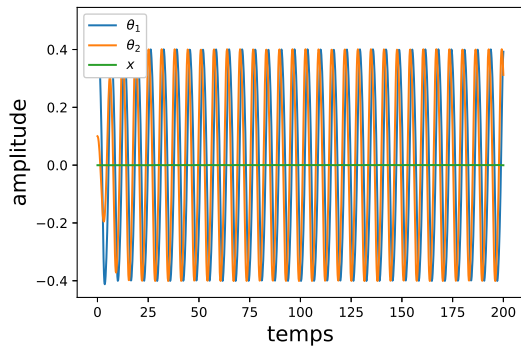


FIGURE 7 – $\varepsilon = 0.5$, $\theta_1(0) = 0.5 \text{ rad}$, $\theta_2(0) = 0.1 \text{ rad}$

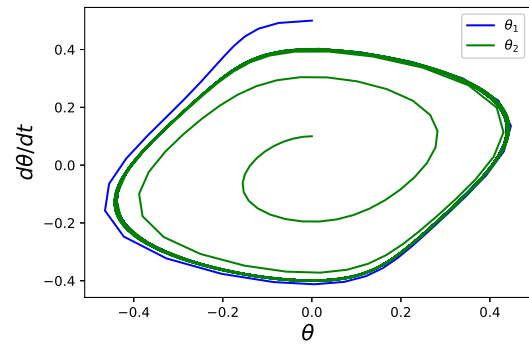


FIGURE 8 – $\varepsilon = 0.5$, $\theta_1(0) = 0.5 \text{ rad}$, $\theta_2(0) = 0.1 \text{ rad}$

```

from __future__ import division
import matplotlib.pyplot as plt
from math import sin
from math import cos
from scipy import *
from pylab import *
from scipy.integrate import odeint      # Module de résolution des équations
différentielles

wo=1
epsilon=0.5
m=0.05
M=100
x0=0.2
alpha=m/(2*m+M)

def deriv(syst, t):
    [xA,xB,vxA,vxB] = syst              # Variables
    dxAdt=vxA                          # Equation différentielle 1
    dxBdt=vxB                          # Equation différentielle 3
    dvxAdt=((-wo**2*sin(xA)+epsilon*(1-(xA/x0)**2)*vxA-
    alpha*cos(xA)*(sin(xA)*vxA**2+sin(xB)*vxB**2))*(1-alpha*(cos(xB))**2)+(-
    wo**2*sin(xB)-epsilon*(1-(xB/x0)**2)*vxB-
    alpha*cos(xB)*(sin(xA)*vxA**2+sin(xB)*vxB**2))*(alpha*cos(xA)*cos(xB)))/(1-
    alpha*(cos(xA))**2-alpha*(cos(xB))**2) # Equation différentielle 5
    dvxBdt=((-wo**2*sin(xB)+epsilon*(1-(xB/x0)**2)*vxB-
    alpha*cos(xB)*(sin(xA)*vxA**2+sin(xB)*vxB**2))*(1-alpha*(cos(xA))**2)+(-
    wo**2*sin(xA)-epsilon*(1-(xA/x0)**2)*vxA-
    alpha*cos(xA)*(sin(xA)*vxA**2+sin(xB)*vxB**2))*(alpha*cos(xA)*cos(xB)))/(1-
    alpha*(cos(xA))**2-alpha*(cos(xB))**2) # Equation différentielle 7
    return [dxAdt,dxBdt,dvxAdt, dvxBdt] # Dérivées des variables

# Paramètres d'intégration
start = 0
end = 200
numsteps = 1000
t = linspace(start,end,numsteps)

# Conditions initiales et résolution
xA0= 0.5
vxA0 = 0
xB0= 0.1
vxB0 =0

syst_CI=array([xA0,xB0,vxA0,vxB0])      # Tableau des CI
Sols=odeint(deriv,syst_CI,t)            # Résolution numérique des équations
différentielles

# Récupération des solutions
[xA,xB,vxA,vxB] = Sols . T              # Décomposition du tableau des solutions :
Affectation avec transposition

x=-alpha*(sin(xA)+sin(xB))
# Graphiques des solutions

plt.figure()
plot(t,xA, label=r"$\theta_1$")          # Solution numérique
plot(t,xB, label=r"$\theta_2$")          # Solution numérique
plot(t,x, label=r"$x$")
#xlim(0, 10)                             # Limites de l'axe des
abscisses
xlabel(u"temps", fontsize=16)             # Label de l'axe des abscisses
#ylim(-1, 1)                             # Limites de l'axe des

```

```

ordonnées
ylabel(u"amplitude", fontsize=16)          # Label de l'axe des ordonnées
#axis('equal')
legend()
plt.savefig('plot.eps')
show()
#display plot.png
plt.close()

plt.plot(vxA,xA, color='b', label=r"$\theta_1$")
plt.plot(vxB,xB, color='g', label=r"$\theta_2$")
xlabel(r"$\theta$", fontsize=16)          # Label de l'axe des abscisses
#ylim(-1, 1)                             # Limites de l'axe des
ordonnées
ylabel(r"$d\theta/dt$", fontsize=16)      # Label de l'axe des ordonnées
#axis('equal')
legend()
plt.savefig('plot2.eps')
show()
#display plot.png
plt.close()

```