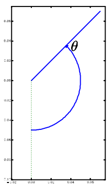
Informatique - TP 9

Pendule oscillant

Le but de ce TD est d'étudier le mouvement d'un pendule oscillant :



et notamment de comparer la solution obtenue avec l'approximation habituelle faite en physique $\sin(\theta) \underset{\theta\to 0}{\approx} \theta$ et celle obtenue lorsque l'on effectue une résolution numérique approchée.
On rappelle que l'équation différentielle régissant le mouvement du pendule est

$$\ddot{\theta} + \alpha\dot{\theta} + \frac{g}{l}\sin(\theta) = 0$$

où α (en $\mathtt{s}^{-1}$) caractérise le frottement fluide subi par le pendule, g est l'accélération de la pesanteur terrestre et l (en $\mathtt{m}$) est la longueur du pendule.
Pour l'ensemble du TD nous prendrons $g \approx 10\mathtt{m.s}^{-2}$ et $l \approx 0,1\mathtt{m}$.

I. Comparaison de l'approximation « petits angles » et de la solution numérique

Dans cette partie on prends $\alpha = 0\mathtt{s}^{-1}$.
Nous allons utiliser la méthode d'Euler pour la résolution numérique de l'équation différentielle.

I.1. Programmation de la méthode d'Euler

On rappelle que le principe de la méthode d'Euler est :

- d'approximer une dérivée $f'(t_0) = \lim\limits_{h\to 0} \dfrac{f(t_0+h)-f(t_0)}{h}$ par $f'(t_0) \approx \dfrac{f(t_0+h)-f(t_0)}{h}$ avec h « suffisamment petit » ;
- de ne calculer les valeurs des fonctions désirées que pour des temps discrets multiples de h : ici par exemple, on souhaite calculer les valeurs

$$\theta_n = \theta(t_n) \quad \text{où} \quad t_n = n \times h$$

En utilisant la méthode d'Euler, calculer $\dot{\theta}_{n+1}$ en fonction de θ_n puis θ_{n+1} en fonction de $\dot{\theta}_{n+1}$.

I.2. Comparaison

Tracer sur un même graphique la solution obtenue pour l'approximation « petits angles » et la solution numérique en prenant comme conditions initiales $\theta_0 = 0,1\mathtt{rad}$ et $\dot{\theta}_0 = 0\mathtt{rad.s}^{-1}$.
Recommencer en augmentant la valeur de θ_0.
Remarques ?

I.3. Période du mouvement en fonction de θ_0

Écrire une fonction `periode(l)` permettant dans une liste de trouver le premier élément d'indice i pour lequel `l[i-1]<=l[i]` et `l[i+1]<=l[i]`.
Utiliser cette fonction pour tracer la période du pendule en fonction de la valeur de θ_0 (on rappelle que $\dot{\theta}_0 = 0\texttt{rad.s}^{-1}$).

II. Avec forces de frottement

Refaire les questions de la partie précédente en prenant $\alpha = 1\texttt{s}^{-1}$.

Comparaison de la solution numérique et de l'approximation petits angles :

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

g=10
l=0.1
beta=g/l

def pas(theta,thetap,alpha=0,h=1e-3):
    nthetap = thetap + h*(-alpha*thetap-beta*np.sin(theta))
    ntheta = theta + h*thetap
    return ntheta,nthetap

def resNum(theta0,thetap0,fin,alpha=0,h=1e-3):
    t = 0
    theta = theta0
    thetap = thetap0
    Lt = [0]
    Ltheta = [theta0]
    while t<fin:
        t+=h
        theta,thetap = pas(theta,thetap,alpha,h)
        Ltheta.append(theta)
        Lt.append(t)
    return Lt,Ltheta

T,Theta = resNum(2.5,0,2)
plt.plot(T,Theta)
Theta2 = 0.1*np.cos(beta**0.5*np.array(T))
plt.plot(T,Theta2)
```

Période du pendule en fonction de θ_0 :

```
def periode(l):
    i=0
    while l[i]>=l[i+1]:
        i+=1
    while l[i]<l[i+1]:
        i+=1
    return temps[i]

temps=np.linspace(0.,5.,1000)
per=[]
for theta0 in np.linspace(0.1,2.5,100):
    temps, sol_num=resNum(theta0,0.,2)
    per.append(periode(sol_num))

plt.figure()
plt.plot(np.linspace(0.1,2.5,100),per)
```

Avec force de frottement :

```
plt.figure()
plt.plot(np.linspace(0.1,2.5,100),per)

plt.figure()
T,Theta = resNum(2.5,0,2,alpha=1)
plt.plot(T,Theta)
Theta2 = 0.1*np.cos(beta**0.5*np.array(T))
plt.plot(T,Theta2)

temps=np.linspace(0.,5.,1000)
per=[]
for theta0 in np.linspace(0.1,3.1,100):
    temps, sol_num=resNum(theta0,0.,2,alpha=1)
    per.append(periode(sol_num))

plt.figure()
plt.plot(np.linspace(0.1,3.1,100),per)
```