普通物理期中 (電腦模擬)

11127137 黃乙家

所有模擬程式碼皆放在 Github 上，
連結：https://github.com/ja-errorpro/GeneralPhysicsMidtermExam

1 低軌衛星題

```
# 1_B.py 計算同步衛星高度
import math
R = 6.4e6
T = 86400
g = 9.8
pi = math.pi
H = (g*(R**2)*(T**2)/(4*(pi**2)))**(1/3) - R
print(H, "m, or ", H/1000, "km")
```

```
# 1_C.py 計算低空衛星周期

import math

R = 6.4e6

T = 86400

g = 9.8

pi = math.pi

H_B = (g*(R**2)*(T**2)/(4*(pi**2)))**(1/3) - R

H_C = 5e5

T_C = math.sqrt(((R+H_C)**3)/((R+H_B)**3)*(T**2))

print(T_C, "s")
```

```
# 1_D.py 計算衛星覆蓋面積
import math
R = 6400
pi = math.pi
H_B = 35940
H_C = 500
S_B = 2*pi*(R**2)*(1-R/(R+H_B))
S_C = 2*pi*(R**2)*(1-R/(R+H_C))
print("S_B: ", S_B, "km^2")
print("S_C: ", S_C, "km^2")
```

2 化學碰撞學說題

```
# 2_B.py 畫出 Activation Energy 與 m 的關係圖
import matplotlib.pyplot as plt
import numpy as np

T = 323
k = 1.38e-23
m_A = np.linspace(1,100,100) # 設 m_A = 1~100 kg
m_B = np.linspace(50,100,100) # 設 m_B = 50~100 kg
E_act = 3*k*T - (3*k*T * (m_A + 50 - 2 * (m_A * 50)**0.5) ) / ( 2 * (m_A + 50) ) # 計算 Activation Energy
plt.plot(m_A,E_act, label = 'm_A')
plt.plot(m_B,E_act, label = 'm_B')
plt.plot(m_A + m_B,E_act, label = 'm_A + m_B')
plt.legend()
plt.xlabel('m (kg)')
plt.ylabel('Activation Energy (J)')
plt.show()
```

3 馬尾題

```python
# 3_C_1.py 求角度微分方程解
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

g = 9.8
l = 1
def diff(y, t):
    omega, theta = y
    return np.array([-(g/l)*np.sin(theta), omega])
t = np.linspace(0, 10, 1000)
theta_0 = 50 / 180 * np.pi
ret = odeint(diff, [0, theta_0 ], t)

plt.plot(t, ret[:, 0])
plt.plot(t, ret[:, 1])
plt.show()
```

```
# 3_C_2.py 模擬週期

import numpy as np

import matplotlib.pyplot as plt

from scipy import special


theta_0 = np.linspace(0, np.pi, 100)

L = 1

g = 9.8

omega_0 = np.sqrt(g/L)

T_0 = 2*np.pi/omega_0

T = 4*np.sqrt(L/g)*special.ellipk(np.sin(theta_0/2))

plt.plot(theta_0, T/T_0)

plt.xlabel(r'$\theta_0$')

plt.ylabel(r'$T/T_0$')

plt.show()
```

```
# 3_D_1.py 推導微分方程
from sympy import *
from sympy import Derivative as D

var("x1 x2 y1 y2 L1 L2 m1 m2 dtheta1 dtheta2 ddtheta1 ddtheta2
    t g tmp")

var("theta1 theta2", cls=Function)



sublist = [
    (D(theta1(t), t, t), ddtheta1),
    (D(theta1(t), t), dtheta1),
    (D(theta2(t), t, t), ddtheta2),
    (D(theta2(t), t), dtheta2),
    (theta1(t), theta1()),
    (theta2(t), theta2())
]

x1 = L1 * sin(theta1(t))
y1 = -L1 * cos(theta1(t))
x2 = x1 + L2 * sin(theta2(t))
y2 = y1 - L2 * cos(theta2(t))
```

```
vx1 = diff(x1, t)
vy1 = diff(y1, t)
vx2 = diff(x2, t)
vy2 = diff(y2, t)

L = m1/2 * (vx1**2 + vy1**2) + m2/2 * (vx2**2 + vy2**2) - m1 *
    g * y1 - m2 * g * y2

def lagrange(L, v):
    dv = D(v(t),t)
    a = L.subs(dv, tmp).diff(tmp).subs(tmp, dv)
    b = L.subs(dv, tmp)
    b = b.subs(v(t),v())
    b = b.diff(v())
    b = b.subs(v(), v(t))
    b = b.subs(tmp, dv)
    c = a.diff(t) - b
    c = c.subs(sublist)
    c = trigsimp(simplify(c))
    c = collect(c,
        [theta1(),theta2(),dtheta1,dtheta2,ddtheta1,ddtheta2])
    return c
```

```
eq1 = lagrange(L, theta1)
eq2 = lagrange(L, theta2)

print("eq1 = ", eq1)
print("eq2 = ", eq2)
```

```
# 3_D_2.py 模擬動畫
import matplotlib
matplotlib.use('WXAgg')
import matplotlib.pyplot as plt

import numpy as np
from scipy.integrate import odeint
from math import *
import wx



g = 9.8
class DoublePendulum(object):
    def __init__(self,m1,m2,L1,L2):
        self.m1, self.m2 = m1, m2
        self.L1, self.L2 = L1, L2
        self.init_stat = np.array([0.0, 0.0, 0.0, 0.0])

    def equations(self,w,t):
        m1, m2, L1, L2 = self.m1, self.m2, self.L1, self.L2
        theta1, theta2, v1, v2 = w
        dth1 = v1
        dth2 = v2
        eq1a = (m1+m2)*L1*L1
        eq1b = m2*L1*L2*cos(theta1-theta2)
        eq1c = L1*(m2*L2*dth2*dth2*sin(theta1-theta2) +
            (m1+m2)*g*sin(theta1))

        eq2a = L1*m2*L2*cos(theta1-theta2)
        eq2b = L2*L2*m2
        eq2c = m2*L2*(-L1*dth1*dth1*sin(theta1-theta2) +
            g*sin(theta2))

        dv1, dv2 = np.linalg.solve([[eq1a, eq1b], [eq2a, eq2b]],
            [-eq1c, -eq2c])

        return np.array([dth1, dth2, dv1, dv2])

def double_pendulum_odeint(pendulum, l, r, step):
    t = np.arange(l,r,step)
    trk = odeint(pendulum.equations, pendulum.init_stat, t)
    theta1, theta2 = trk[:,0], trk[:,1]
    L1 = pendulum.L1
    L2 = pendulum.L2
    x1 = L1*np.sin(theta1)
    y1 = -L1*np.cos(theta1)
    x2 = x1 + L2*np.sin(theta2)
    y2 = y1 - L2*np.cos(theta2)
    pendulum.init_stat = trk[-1,:].copy()
    return [x1, y1, x2, y2]

fig = plt.figure(figsize=(6,6))

line1, = plt.plot([0,0],[0,0],"-o")
line2, = plt.plot([0,0],[0,0],"-o")
plt.axis("equal")
plt.xlim(-5,5)
plt.ylim(-5,5)

print('模擬雙擺運動(若直接按下Enter則使用預設值)：')

m1 = input('請輸入m1質量[1.0]：')
if m1 == '':
    m1 = 1.0
m2 = input('請輸入m2質量[1.0]：')
if m2 == '':
    m2 = 1.0
L1 = input('請輸入L1長度[1.0]：')
if L1 == '':
    L1 = 1.0
```

```
L2 = input('請輸入L2長度[1.0]：')
if L2 == '':
    L2 = 1.0

pendulum = DoublePendulum(m1, m2, L1, L2)
theta1 = input('請輸入初始theta1角度(徑度)[1.0]：')
if theta1 == '':
    theta1 = 1.0
theta2 = input('請輸入初始theta2角度(徑度)[1.0]：')
if theta2 == '':
    theta2 = 1.0
pendulum.init_stat[:2] = theta1, theta2

x1,y1,x2,y2 = double_pendulum_odeint(pendulum, 0, 30, 0.02)
plt.plot(x1,y1,label="m_1")
plt.plot(x2,y2,label="m_2")

plt.title("m1 = %s, m2 = %s, L1 = %s, L2 = %s, theta1 = %s,
    theta2 = %s" % (m1, m2, L1, L2, theta1, theta2))

idx = 0

def update_line(event):
    global x1,y1,x2,y2,idx
    if idx == len(x1):
        idx = 0
        x1, y1, x2, y2 = double_pendulum_odeint(pendulum, 0, 30, 0.02)
    line1.set_xdata([0,x1[idx]])
    line1.set_ydata([0,y1[idx]])
    line2.set_xdata([x1[idx],x2[idx]])
    line2.set_ydata([y1[idx],y2[idx]])
    fig.canvas.draw()
    idx += 1

id = wx.ID_ANY
actor = fig.canvas.manager.frame
actor.Bind(wx.EVT_TIMER, update_line, id=id)
timer = wx.Timer(actor, id)
timer.Start(1)

plt.legend()

plt.show()
```