Московский авиационный институт
(национальный исследовательский университет)
Институт № 8 «Информационные технологии и прикладная математика»

Лабораторная работа №4
по курсу «Теоретическая механика»
Уравнение Лагранжа

Выполнил студент группы М8О-207Б-20
Мерц Савелий Павлович
Преподаватель: Чекина Евгения Алексеевна

Оценка:

Дата: 25/12/2021

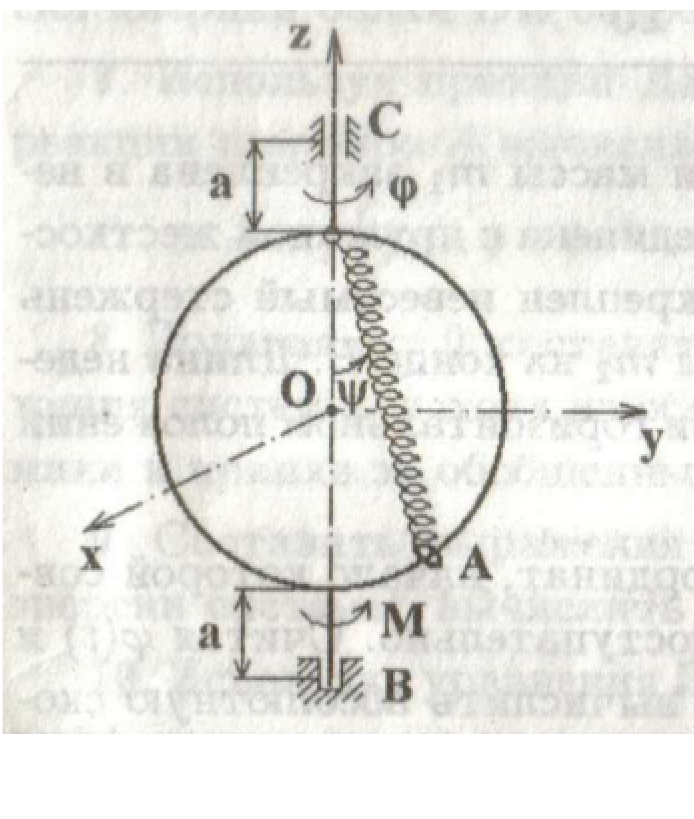
Москва, 2021

Вариант №«19»

Задание:

Написать на языку python программу визуализирующую кинематику плоского движения механической системы или сложного движения, согласно варианту, зафиксировать одну из свободных координат для анализа колебаний. Найти положения равновесия бусинки и исследовать их на устойчивость. Найти период малых колебаний в окрестности нижнего положения равновесия.

Механическая система:



Текст программы

Основная :

```
import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

from scipy.integrate import odeint

import sympy as sp

import math


def formY(y, t, fV, fOm):

    y1,y2,y3,y4 = y

    dydt = [y3,y4,fV(y1,y2,y3,y4),fOm(y1,y2,y3,y4)]

    return dydt


def formY2(y, t, fOm):

    y1,y2 = y

    dydt = [y2,fOm(y1,y2)]

    return dydt


# defining parameters

# the angle of the plane (and the prism)

alpha = 0.002

M = 1

m = 0.1

R = 0.3

c = 20

l0 = 0.7

g = 9.81


# defining t as a symbol (it will be the independent variable)

t = sp.Symbol('t')
```

```
# defining s, phi, V=ds/dt and om=dphi/dt as functions of 't'
phi=0
psi=sp.Function('psi')(t)
Vphi=0
Vpsi=sp.Function('Vpsi')(t)

l = 2 * R * sp.cos(psi)  # длина пружины
#constructing the Lagrange equations
#1 defining the kinetic energy
TT1 = M * R**2 * Vphi**2 / 4
V1 = 2*Vpsi * R
V2 = Vphi * R * sp.sin(2 * psi)
Vr2 = V1**2 + V2**2
TT2 = m * Vr2 / 2
TT = TT1+TT2
# 2 defining the potential energy
Pi1 = 2 * R * m * g * sp.sin(psi)**2
Pi2 = (c * (l - l0)**2) / 2
Pi = Pi1+Pi2
# 3 Not potential force
M = alpha * phi**2;

# Lagrange function
L = TT-Pi

# тут исследую положения устойчивости при л0 = 0.3
PI = -1*R*sp.cos(2*psi)*m*g + (c/2)*(2*R*sp.cos(psi)-l0)*(2*R*sp.cos(psi)-l0)
print(sp.diff(PI,t))
PI1 = -6.0*(0.6*sp.cos(psi) - 0.502)*sp.sin(psi) - 0.6*(6.0*sp.cos(psi) -
0.502)*sp.sin(psi) + 0.5886*sp.sin(2*psi)#первая производная ПИ
```

```
print(sp.diff(PI1,t))

PI2 = (0.3012 - 3.6*sp.cos(0))*sp.cos(0) + (3.012 - 3.6*sp.cos(0))*sp.cos(0) +
7.2*sp.sin(0)**2 + 1.1772*sp.cos(2*0)# вторая производная ПИ

print(PI2)


# тут уже проверка устойчивости при пси = 0 и л = 0.7

print(sp.diff(sp.diff(PI, t), t))

pi2 = (4.2 - 3.6*sp.cos(0))*sp.sin(0) + (4.2 - 3.6*sp.cos(0))*sp.cos(0) + (4.2
- 3.6*sp.cos(0))*sp.sin(0) + (4.2 - 3.6*sp.cos(0))*sp.cos(0) +
7.2*sp.sin(0)**2 + 0.5886*sp.sin(2*0) + 1.1772*sp.cos(2*0)

print(pi2)

# equations

#ur1 = sp.diff(sp.diff(L,Vphi),t)-sp.diff(L,phi) - M

ur2 = sp.diff(sp.diff(L,Vpsi),t)-sp.diff(L,psi)


# isolating second derivatives(dV/dt and dom/dt) using Kramer's method

# a11 = ur1.coeff(sp.diff(Vphi,t),1)

# a12 = ur1.coeff(sp.diff(Vpsi,t),1)

# a21 = ur2.coeff(sp.diff(Vphi,t),1)

a22 = ur2.coeff(sp.diff(Vpsi,t),1)

#b1 =
-(ur1.coeff(sp.diff(Vphi,t),0)).coeff(sp.diff(Vpsi,t),0).subs([(sp.diff(phi,t)
,Vphi), (sp.diff(psi,t), Vpsi)])

b2 = -ur2.coeff(sp.diff(Vpsi,t),0).subs(sp.diff(phi,t), Vpsi);

# detA = a11*a22-a12*a21

# detA1 = b1*a22-b2*a21

# detA2 = a11*b2-b1*a21

#

# dVdt = detA1/detA

domdt = b2/a22


countOfFrames = 1700
```

```
# Constructing the system of differential equations
T = np.linspace(0, 25, countOfFrames)
# Pay attention here, the function lambdify translate function from the sympy to numpy and then form arrays much more
# faster then we did using subs in previous lessons!
#fVphi = sp.lambdify([phi,psi,Vphi,Vpsi], dVdt, "numpy")
fVpsi = sp.lambdify([psi,Vpsi], domdt, "numpy")
y0 = [np.pi/6, 0]
sol = odeint(formY2, y0, T, args = (fVpsi,))

#sol - our solution
#sol[:,0] - phi
#sol[:,1] - psi
#sol[:,2] - dphi/dt
#sol[:,3] - dpsi/dt

# Построение графика и подграфика с выравниванием осей
fig = plt.figure(figsize=(17, 8))
ax1 = fig.add_subplot(1, 2, 1)
ax1.axis('equal')

phi = 0
psi = sol[:,0]
# Vphi = sol[:,2]
Vpsi = sol[:,1]

w = np.linspace(0, 2 * math.pi, countOfFrames)
conline, = ax1.plot([sp.sin(2*psi[0]) * R * sp.cos(phi), 0], [-sp.cos(2*psi[0]) * R, R], 'black')
P, = ax1.plot(sp.sin(2*psi[0]) * R * sp.cos(phi), -sp.cos(2*psi[0]) * R, marker='o', color='black')
```

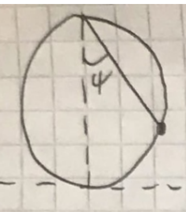
```
Circ, = ax1.plot(R * sp.cos(phi) * np.cos(w), R * np.sin(w), 'black')


#Доп графики
ax2 = fig.add_subplot(4, 2, 2)
ax2.plot(T, psi)
ax2.set_xlabel('T')
ax2.set_ylabel('psi')
ax3 = fig.add_subplot(4, 2, 4)
ax3.plot(T, Vpsi)
ax3.set_xlabel('T')
ax3.set_ylabel('Vpsi')


def anima(i):
    P.set_data(sp.sin(2*psi[i]) * R * sp.cos(phi), -sp.cos(2*psi[i]) * R)
    conline.set_data([sp.sin(2*psi[i]) * R * sp.cos(phi), 0],
[-sp.cos(2*psi[i]) * R, R])
    Circ.set_data(R * sp.cos(phi) * np.cos(w), R * np.sin(w))
    return Circ, P, conline


anim = FuncAnimation(fig, anima, frames=countOfFrames, interval=1, blit=True)
plt.show()
```

Поиск положений равновесия и периода колебаний:



$$\Pi = \frac{c(2R\cos\psi - l_0)^2}{2} + (2R - \cos(\psi)\cdot l)\cdot mg =$$

$$= \frac{c}{2}(2R\cos\psi - l_0)^2 + (2R - 2R\cos^2(\psi))mg =$$

$$= \frac{c}{2}(2R\cos\psi - l_0)^2 + 2R\,mg\sin^2\psi$$

$$\frac{\partial \Pi}{\partial \psi} = 0 \qquad 4R\,mg\sin\psi\cos\psi + c(2R\cos\psi - l_0)(-2R\sin\psi) = 0$$

$$4R\,mg\sin\psi\cos\psi - 4R^2c\sin\psi\cos\psi + 2c\,l_0R\sin\psi = 0$$

$$\sin\psi\cos\psi(4Rmg - 4R^2c) + 2c\,l_0R\sin\psi = 0$$

$$\sin\psi = 0 \qquad \cos\psi = \frac{2c\,l_0R}{4R^2c - 4Rmg} = \frac{1000}{1673}$$

$$\psi = 0, \pi \qquad \psi = \arccos\left(\frac{1000}{1673}\right)$$

$$\frac{\partial^2\Pi}{\partial\psi^2} = 4Rmg(\cos^2\psi - \sin^2\psi) - 4R^2c(\cos^2\psi - \sin^2\psi) +$$

$$+ 2c\,l_0R\cos\psi = 4Rmg\cos(2\psi) - 4R^2c\cos(2\psi) +$$

$$+ 2c\,l_0R\cos\psi = \cos(2\psi)(4Rmg - 4R^2c) + 2c\,l_0R\cos\psi$$

- при $\psi = 0$

$$(4Rmg - 4R^2c) + 2c\,l_0R = -6{,}0228 + 3{,}6 = -2{,}4228 < 0$$

неустойчивое положение

$$4Rmg - 4R^2c + 2c\,l_0R > 0$$

$$l_0 > \frac{4R^2c - 4Rmg}{2c\cdot R}$$

$l_0 > 0{,}5019$ – условие устойчивости при $\psi = 0$

при $\psi = \pi$

$$(4Rmg - 4R^2c) - 2cl_0R > 0$$

$$l_0 < \frac{(4Rmg - 4R^2c)}{2cR}$$

$l_0 < -0{,}2019$ – условие устойчивости при $\psi = \pi$

при $\psi = \arccos\left(\frac{2cl_0R}{4R^2c - 4Rmg}\right)$

$$(2\cos^2\psi - 1)(4Rmg - 4R^2c) + 2cl_0R \cdot \frac{2cl_0R}{4R^2c - 4Rmg} > 0$$

$$2 \cdot \frac{4c^2l_0^2R^2}{(4R^2c - 4Rmg)^2}(4Rmg - 4R^2c) + \frac{4c^2l_0^2R^2}{4R^2c - 4Rmg} - (4Rmg - 4R^2c) > 0$$

$$-2 \cdot \frac{4c^2l_0^2R^2}{4R^2c - 4Rmg} + \frac{4c^2l_0^2R^2}{4R^2c - 4Rmg} - (4Rmg - 4R^2c) > 0$$

$$-\frac{4c^2l_0^2R^2}{4R^2c - 4Rmg} - (4Rmg - 4R^2c) > 0$$

$$-\frac{144l_0^2}{6{,}0228} - (1{,}1772 - 7{,}2) > 0 \qquad -\frac{144l_0^2}{6{,}0228} + 6{,}0228 > 0$$

$$l_0^2 < \frac{6{,}0228^2}{144} \qquad -0{,}5019 < l_0 < 0{,}5019$$

Итог: при $\psi = 0$ положение устойчиво если $l_0 > 0{,}5019$

при $\psi = \pi$ положение не устойчиво (отриц длины не бывает)

при $\psi = \arccos\frac{2cRl_0}{4R^2c - 4Rmg}$ положение устойчиво при $-0{,}5019 < l_0 < 0{,}5019$.

Для поиска периода малых колебаний в положении равновесия $\psi = 0$ необходимо выполнить условие $l_0 > 0{,}5014$. Возьмём $l_0 = 0{,}502$

$$П = -R\,mg\cos(2\psi) + \frac{c}{2}\left(2R\cos(\psi) - l_0\right)^2 =$$

$$= -0{,}2943\cos(2\psi) + \frac{18}{5}\cos^2(\psi) - \frac{753}{125}\cos(\psi) + C =$$

$$= -0{,}2943\cos(2\psi) + \frac{9}{5}\cos(2\psi) - \frac{753}{125}\cos(\psi) + C =$$

$$= \left(1 - 2\psi^2\right)\left(\frac{9}{5} - 0{,}2943\right) - \frac{753}{125}\left(1 - \frac{\psi^2}{2}\right) + C =$$

$$= -3{,}0114\,\psi^2 + 3{,}012\,\psi^2 + C = 0{,}0006\,\psi^2 + C \Rightarrow C' = 0{,}0006$$

$$T = \frac{mR^2 \cdot 4}{2} = 2mR^2\dot{\psi}^2 \Rightarrow a = 4mR^2 = 0{,}018$$

Период малых колебаний $= 2\pi\sqrt{\frac{a}{c'}} = 2\pi\sqrt{30}$.

Результат работы:

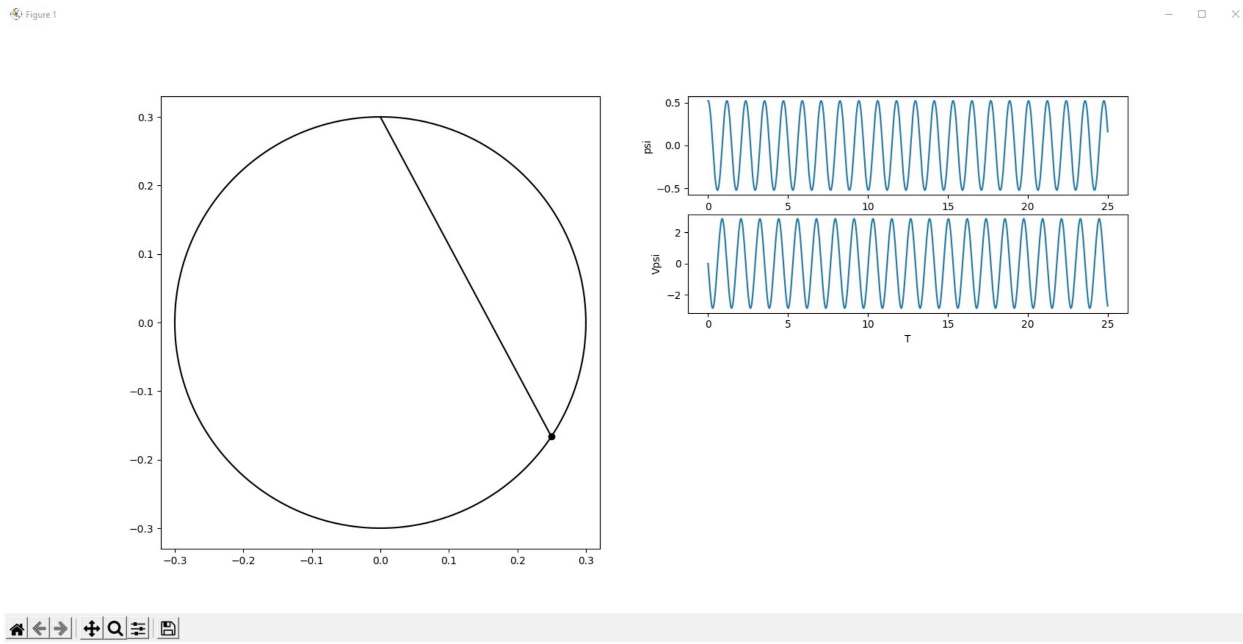


Figure 1
psi
Vpsi
T

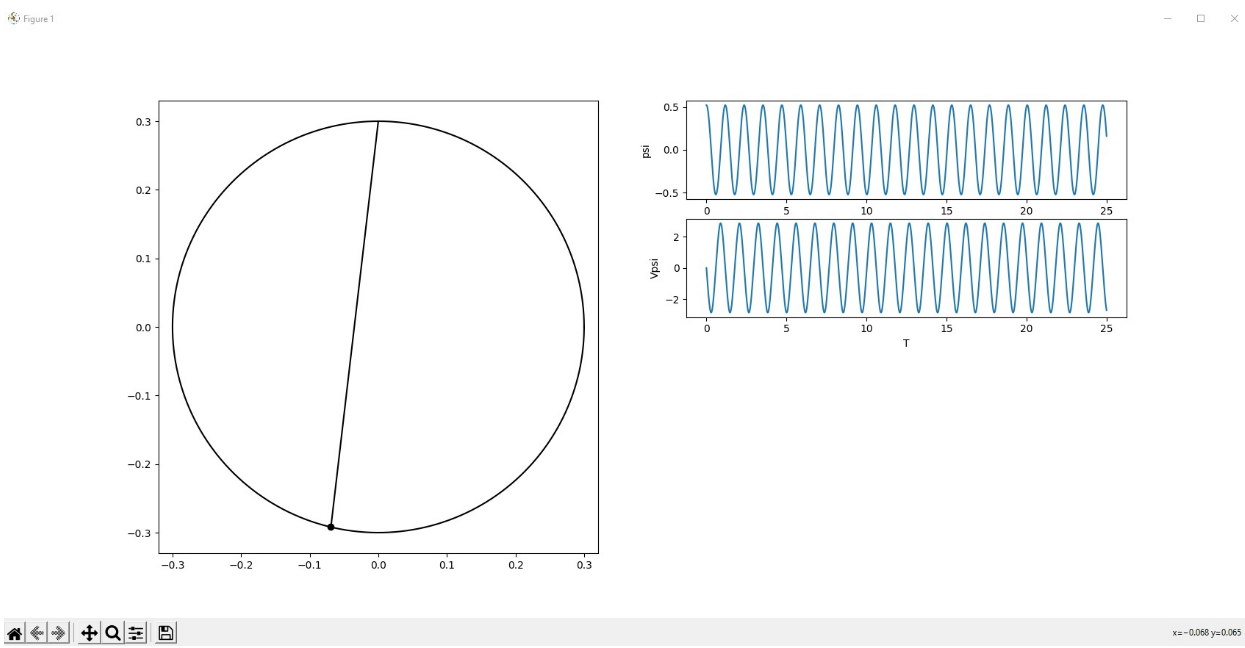


Figure 1
psi
Vpsi
T
x=−0.068 y=0.065