

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №3-4

«ДИНАМИКА СИСТЕМЫ»

ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И ОС- НОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»

ВАРИАНТ ЗАДАНИЯ №15

Выполнил(а) студент группы М8О-208Б-20

Маринин Иван Сергеевич _____
подпись, дата

Проверил и принял

Доцент каф. 802, Чекина Е.А. _____
подпись, дата

с оценкой _____

Москва, 2021

Вариант №15

Задание:

Построить анимацию движения системы, а также графики законов движения системы для разных случаев системы (поэкспериментировать с параметрами системы). Вывести уравнения, определяющие законы движения системы.

$$(m_1 + m_2)(\ddot{s} - g \sin \alpha) - m_2 \ell [\ddot{\varphi} \cos(\varphi - \alpha) - \dot{\varphi}^2 \sin(\varphi - \alpha)] = 0,$$

$$\ell \ddot{\varphi} - \ddot{s} \cos(\varphi - \alpha) + g \sin \varphi = 0.$$

Лабораторная работа №3

Для переменных задать следующие значения

12. Задавая численные значения параметров и начальные условия: $m_1 = 1$ кг, $m_2 = 0,5$ кг, $\ell = 0,5$ м, $t_0 = 0$, $s_0 = 0$, $\varphi_0 = \pi/6$, $\dot{s}_0 = 0$, $\dot{\varphi}_0 = 12\text{с}^{-1}$, составить программу решения системы дифференциальных уравнений и на ЭВМ построить зависимости $s(t)$, $\varphi(t)$, $N(t)$ для двух значений параметра α : $\alpha = \pi/12$ и $\alpha = \pi/4$.

Текст программы

```
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import sympy as sp
import math
from scipy.integrate import odeint

matplotlib.use("TkAgg")

t = np.linspace(1, 20, 1001)

#создаю функцию, которая подставляет уравнение Лагранжа
def odesys(y, t, m1, m2, l, g, alpha):
    dy = np.zeros(4)
    dy[0] = y[2]
    dy[1] = y[3]

    a11 = m1 + m2
    a12 = -m2 * l * np.cos(y[1] - alpha)
    a21 = -np.cos(y[1] - alpha)
    a22 = l

    b1 = (m1 + m2) * g * np.sin(alpha) - m2 * l * (y[3])**2 * np.sin(y[1] - alpha)
```

```

b2 = -g * np.sin(y[1])

dy[2] = (b1 * a22 - a12 * b2) / (a11 * a22 - a21 * a12)
dy[3] = (b2 * a11 - b1 * a21) / (a11 * a22 - a21 * a12)

return dy

#задаю начальные значения всех параметров
m1 = 1
m2 = 0.5
l = 5
g = 9.81
alpha = math.pi / 4

x0 = 0
phi0 = math.pi / 12
dx0 = 0
dphi0 = 12
y0 = [x0, phi0, dx0, dphi0]

Y = odeint(odesys, y0, t, (m1, m2, l, g, alpha))

x = Y[:, 0]
phi = Y[:, 1]
dx = Y[:, 2]
dphi = Y[:, 3]
X_0 = 4
a = 2.5
b = 3

#задаю функции движения тел
X_A = -a / 40 * x
Y_A = X_A
Y_B = Y_A - l * np.sin(math.pi / 1.2 - phi)
X_B = X_A + l * np.cos(math.pi / 1.2 - phi)
X_Box = np.array([-0.75, -1.3, 0.2, 0.75, -0.75])
Y_Box = np.array([-0.75, -0.25, 1.25, 0.75, -0.75])

#строю наклонную плоскость
X_Straight = [-10, 0, 10]
Y_Straight = [-10, 0, 10]

#рисую окно с телом
fig = plt.figure(figsize = [9, 5])
ax = fig.add_subplot(1, 1, 1)
ax.axis('equal')
ax.set(xlim = [-5, 5], ylim = [-5, 5])

ax.plot(X_Straight, Y_Straight)
Drawed_Box = ax.plot(X_A[0] + X_Box, Y_A[0] + Y_Box)[0]
Line_AB = ax.plot([X_A[0], X_B[0]], [Y_A[0], Y_B[0]])[0]
Point_A = ax.plot(X_A[0], Y_A[0], marker = 'o')[0]
Point_B = ax.plot(X_B[0], Y_B[0], marker = 'o', markersize = 5)[0]

#рисую окно с графиками
fig2 = plt.figure(figsize = [9, 5])
ax2 = fig2.add_subplot(2, 2, 1)
ax2.plot(t, -x)
plt.title('x(t)')

ax3 = fig2.add_subplot(2, 2, 2)
ax3.plot(t, phi)
plt.title('phi(t)')

ax4 = fig2.add_subplot(2, 2, 3)
ax4.plot(t, dx)

```

```

plt.title('dx(t)')

ax5 = fig2.add_subplot(2, 2, 4)
ax5.plot(t, dphi)
plt.title('dphi(t)')

plt.subplots_adjust(wspace=0.3, hspace=0.7)

def Kino(i):
    Point_A.set_data(X_A[i], Y_A[i])
    Point_B.set_data(X_B[i], Y_B[i])
    Line_AB.set_data([X_A[i], X_B[i]], [Y_A[i], Y_B[i]])
    Drawed_Box.set_data(X_A[i] + X_Box, Y_A[i] + Y_Box)
    return [Point_A, Point_B, Line_AB, Drawed_Box]

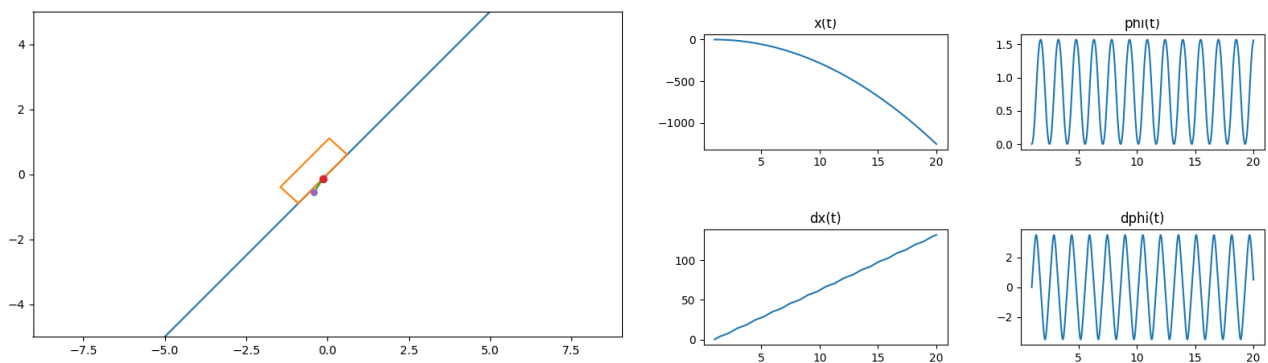
anima = FuncAnimation(fig, Kino, frames = 1000, interval = 10)
plt.show()

```

Лабораторная работа №4

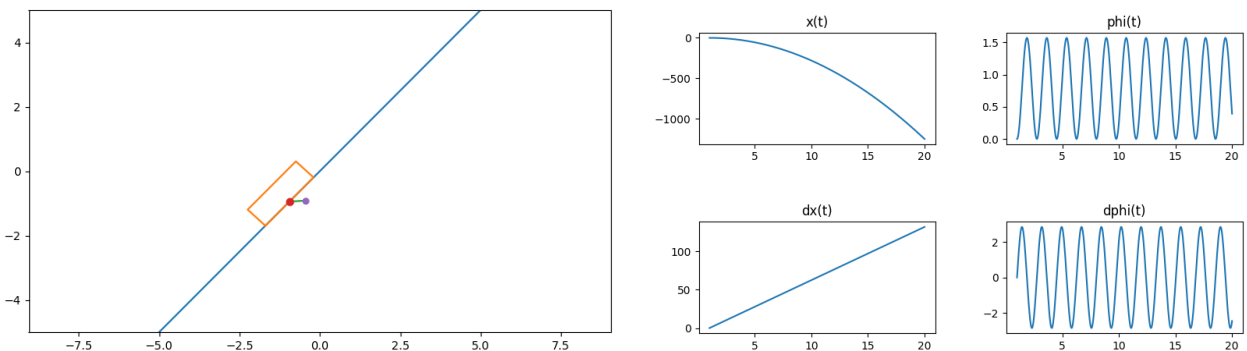
Работа программы:

- 1) При начальных параметрах: $m_1 = 1$; $m_2 = 0.5$; $l = 0.5$; $x_0 = 0$; $\phi_0 = \pi/12$; $dx_0 = 0$; $d\phi_0 = 12$



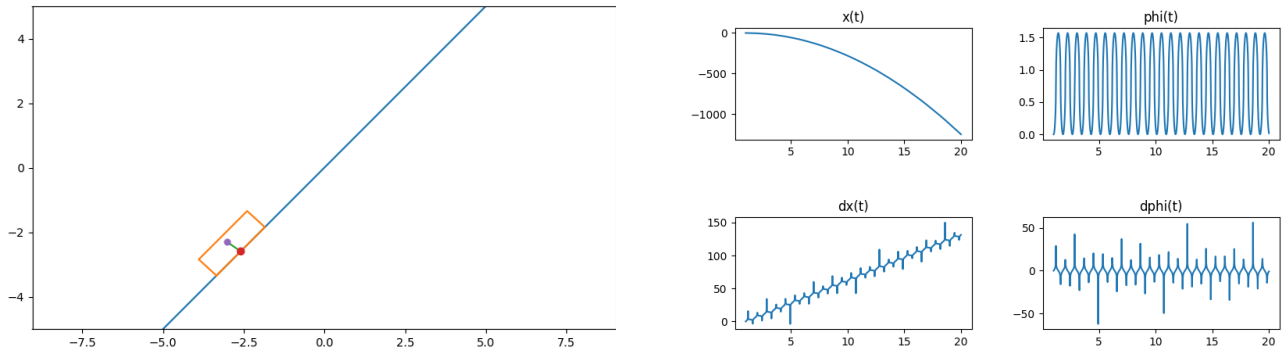
Результат : Тело движется вниз с нормальной скоростью, ускоряясь.

- 2) При параметрах: $m_1 = 1000$; $m_2 = 0.5$; $l = 0.5$; $x_0 = 0$; $\phi_0 = \pi/12$; $dx_0 = 0$; $d\phi_0 = 12$



Результат: Тело стало двигаться значительно медленней, маятник колеблется в том же режиме.

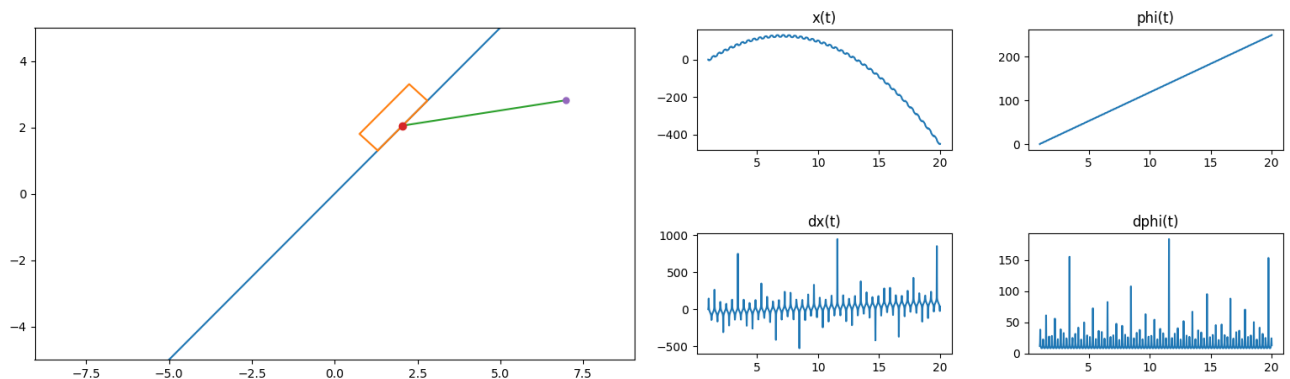
3) При параметрах: $m_1 = 1$; $m_2 = 500$; $l = 0.5$; $x_0 = 0$; $\phi_0 = \pi/12$; $\dot{x}_0 = 0$; $\dot{\phi}_0 = 12$



Результат:

В интервале от $\pi/2$ до $-\pi/2$ скорость движения маятника возросла, также маятник стал замирать в этих точках, скорость движения тела практически увеличилась.

4) При параметрах: $m_1 = 1$; $m_2 = 500$; $l = 5$; $x_0 = 0$; $\phi_0 = 0$; $\dot{x}_0 = 0$; $\dot{\phi}_0 = 12$



Результат: Из-за увеличения длины нити тело стало двигаться в противоположном направлении.

Вывод

В ходе лабораторной работы я запрограммировал уравнение Лагранжа в анимацию 2ой лабораторной работы, отладил программу, поэкспериментировали с начальными значениями и получил несколько случаев поведения системы, которые были рассмотрены с приведением графиков и скриншотов анимации.