

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

Лабораторная работа №2
по курсу «Теоретическая механика и основы компьютерного
моделирования»
Анимация точки

Выполнил студент группы М8О-210Б-21

Минеева Светлана Алексеевна

Преподаватель: Бардин Борис Сабирович

Оценка:

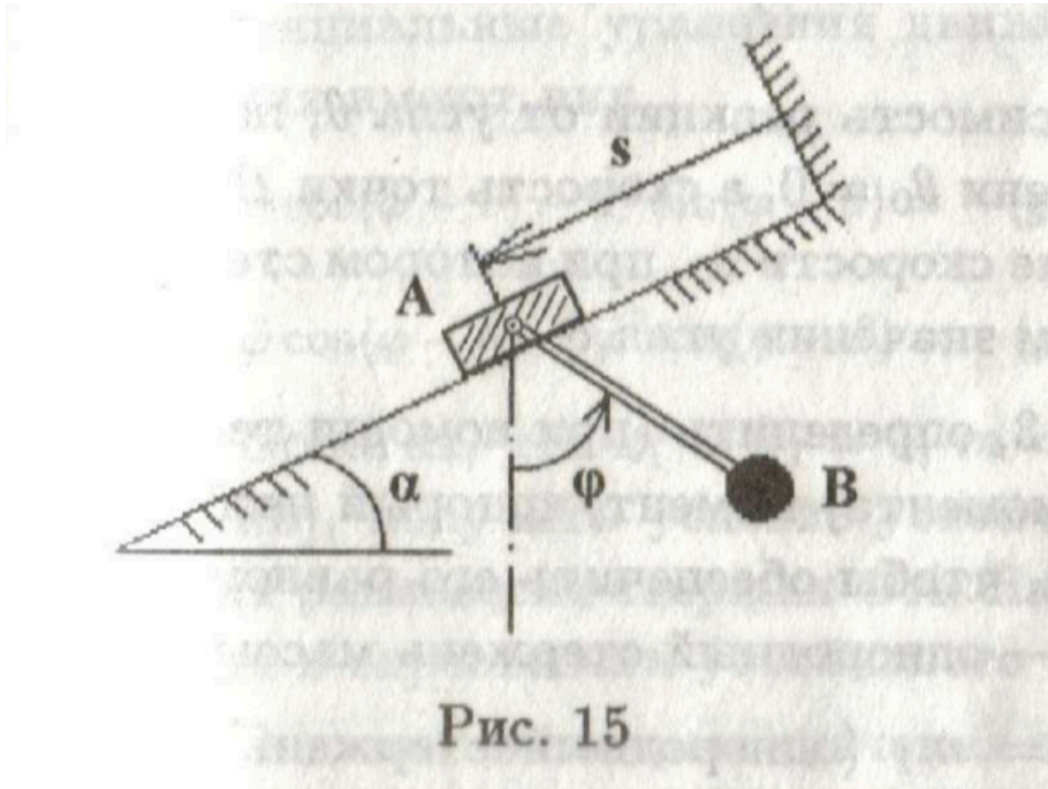
Дата: 10.10.2022

Москва, 2022

Вариант № 15

Задание: Реализовать анимацию движения механической системы в среде Python.

Механическая система:



Текст программы:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import matplotlib
import math

matplotlib.use("TkAgg")

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

x = np.cos(t)
phi = np.sin(t)
alpha = math.pi / 4
l = 3 #длина стержня
a_side = 2 #длина тела
b_side = 1 #высота тела
dia = (a_side ** 2 + b_side ** 2) ** (1 / 2)
betta = np.arctan(b_side / a_side)
proc = (dia / 2) * np.cos(alpha + betta)
gec = (dia / 2) * np.sin(alpha + betta)
```

```

X_A = -x * np.cos(alpha)
Y_A = -x * np.sin(alpha)
X_B = X_A - l * np.sin(phi)
Y_B = Y_A - l * np.cos(phi)

X_Box = np.array([-proc, proc + (b_side * np.sin(alpha)), proc, -proc -
(b_side * np.sin(alpha)), -proc])
Y_Box = np.array([-gec, gec - (b_side * np.cos(alpha)), gec, -gec + (b_side *
np.cos(alpha)), -gec])

fig = plt.figure(figsize=[15, 15])
ax = fig.add_subplot(1, 2, 1)
ax.axis('equal')
ax.set(xlim=[-5, 5], ylim=[-5, 5])

ax.plot(X_A - proc, Y_A - gec, color='grey') #построение линии наклонной
плоскости
Drawed_Box = ax.plot(X_A[0] + X_Box, Y_A[0] + Y_Box, color='blue')[0]
#построение прямоугольного тела
Line_AB = ax.plot([X_A[0], X_B[0], ], [Y_A[0], Y_B[0]], color='black')[0]
#построение стержня
Point_A = ax.plot(X_A[0], Y_A[0], marker='o', color='blue')[0] #построение
точки центра масс тела
Point_B = ax.plot(X_B[0], Y_B[0], marker='o', markersize=10, color='red')[0]
#построение точечного груза

#Построение графика зависимости абциссы точки центра масс тела от параметра t
ax1 = fig.add_subplot(4, 2, 2)
ax1.plot(t, X_A)
plt.title('X of the Blue Point', fontdict={'fontsize': 10})
plt.xlabel('t values', fontdict={'fontsize': 9})
plt.ylabel('x values', fontdict={'fontsize': 9})

#Построение графика зависимости ординаты точки центра масс тела от параметра
t
ax2 = fig.add_subplot(4, 2, 4)
ax2.plot(t, Y_A)
plt.title('Y of the Blue Point', fontdict={'fontsize': 10})
plt.xlabel('t values', fontdict={'fontsize': 9})
plt.ylabel('y values', fontdict={'fontsize': 9})

#Построение графика зависимости абциссы точки центра масс точечного груза от
параметра t
ax3 = fig.add_subplot(4, 2, 6)
ax3.plot(t, X_B)
plt.title('X of the Red Point', fontdict={'fontsize': 10})
plt.xlabel('t values', fontdict={'fontsize': 9})
plt.ylabel('x values', fontdict={'fontsize': 9})

#Построение графика зависимости ординаты точки центра масс точечного груза от
параметра t
ax4 = fig.add_subplot(4, 2, 8)

```

```

ax4.plot(t, Y_B)
plt.title('Y of the Red Point', fontdict={'fontsize': 10})
plt.xlabel('t values', fontdict={'fontsize': 9})
plt.ylabel('y values', fontdict={'fontsize': 9})

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

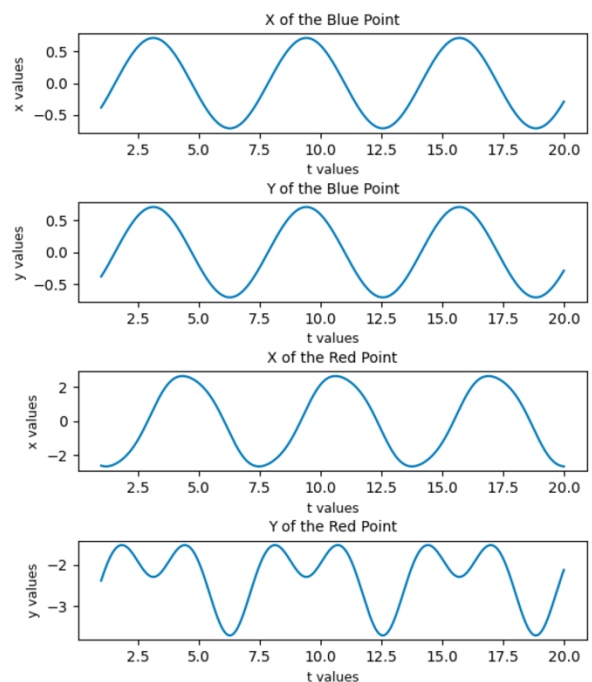
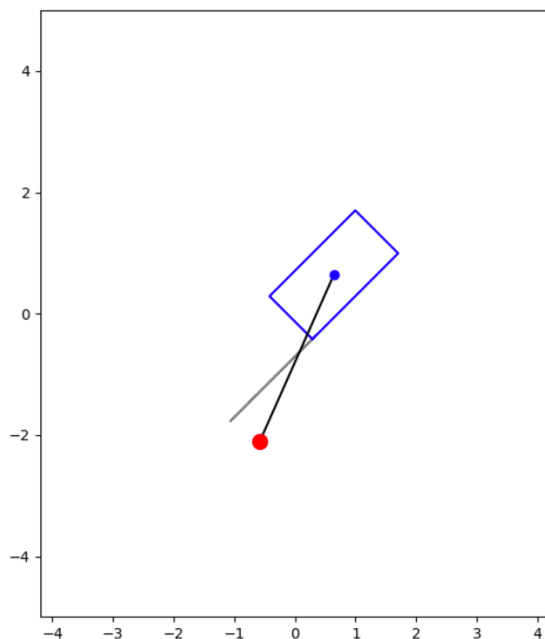
#Функция анимации
def anima(i):
    Point_A.set_data(X_A[i], Y_A[i])
    Point_B.set_data(X_B[i], Y_B[i])
    Line_AB.set_data([X_A[i], X_B[i], ], [Y_A[i], Y_B[i]])
    Drawed_Box.set_data(X_A[i] + X_Box, Y_A[i] + Y_Box)
    return [Point_A, Point_B, Line_AB, Drawed_Box]

anim = FuncAnimation(fig, anima, frames = 1001, interval = 10)
plt.show()

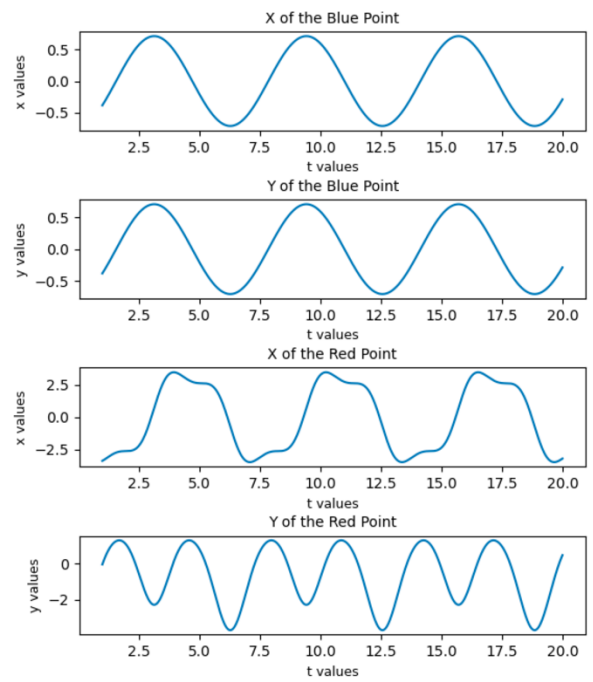
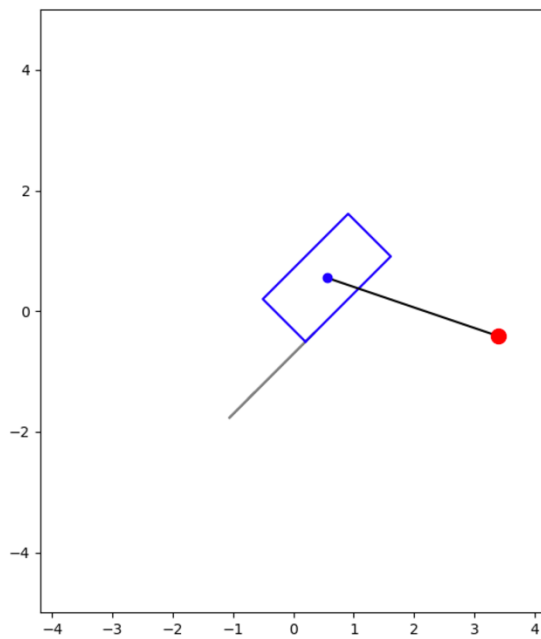
```

Результат работы программы:

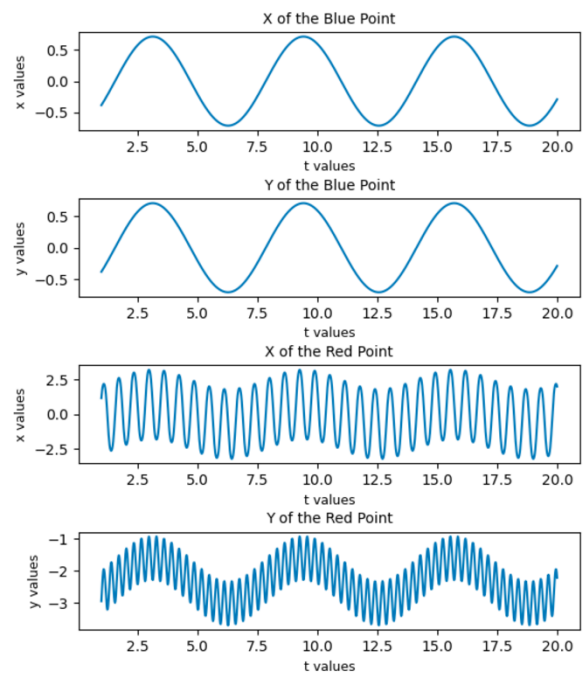
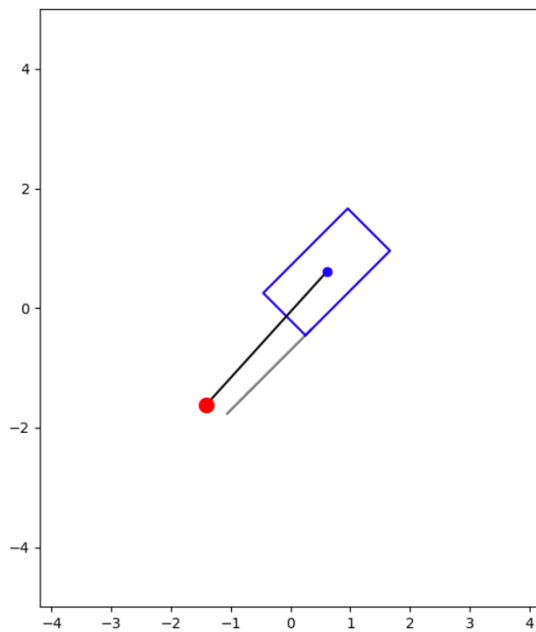
1) $x = \cos(t)$; $\phi = \sin(t)$



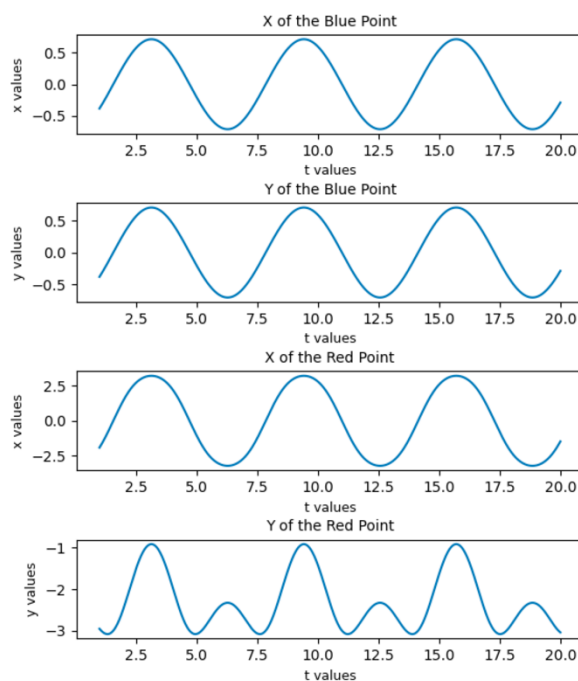
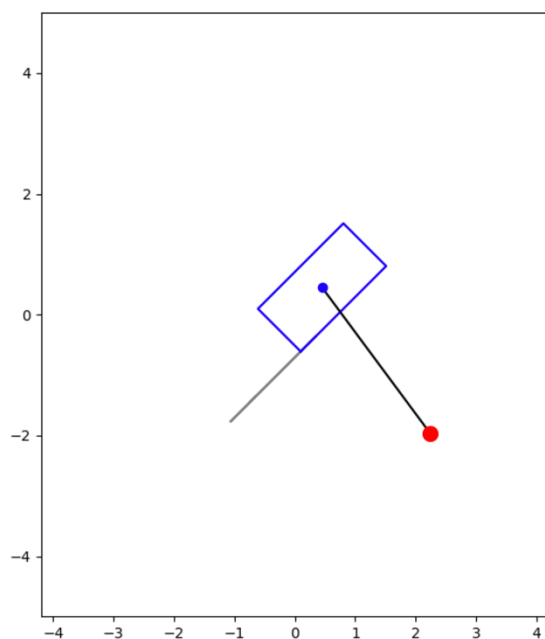
2) $x = \cos(t)$; $\phi = 2*\sin(t)$



3) $x = \cos(t)$; $\phi = \sin(10*t)$



4) $x = \cos(t)$; $\phi = \sin(t)$



Вывод: Я реализовала анимацию движения механической системы в среде Python. В ходе данной работы, я приобрела практические знания в работе с языком программирования Python и его библиотекой Matplotlib.