

Отчет для лабораторной работы №4

НФИбд-02-18

Оразклычев Давут

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	16
5	Ответы на вопросы	17

List of Tables

List of Figures

2.1	Задание	6
3.1	Колебания гармонического осциллятора без затуханий и без действий внешней силы, График 1	9
3.2	Колебания гармонического осциллятора с затуханием и без действий внешней силы, График 2	10
3.3	Колебания гармонического осциллятора с затуханием и под действием внешней силы, График 3	11

1 Цель работы

Решение заданий: Постройте фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора

2 Задание

(рис. 2.1)

Вариант № 41

Постройте фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора для следующих случаев

1. Колебания гармонического осциллятора без затуханий и без действий внешней силы $\ddot{x} + 3.5x = 0$
2. Колебания гармонического осциллятора с затуханием и без действий внешней силы $\ddot{x} + 7\dot{x} + 3x = 0$
3. Колебания гармонического осциллятора с затуханием и под действием внешней силы $\ddot{x} + 5\dot{x} + 2x = 2\sin(6t)$

На интервале $t \in [0; 37]$ (шаг 0.05) с начальными условиями $x_0 = 1, y_0 = 1.2$

Figure 2.1: Задание

3 Выполнение лабораторной работы

Импортируем библиотеки и переменные

```
import numpy as nmp
import math
import matplotlib.pyplot as plt

from scipy.integrate import odeint
```

```
a_1 = 3.5
b_1 = 0.00
```

```
Time_null = 0
Time_Max = 37
Step = 0.05
```

Создаем список t

```
t = nmp.arange(Time_null, Time_Max, Step)
t = nmp.append(t, Time_Max)
```

Создаем функции и уравнение:

```
def p(t):
    return 0
```

```
def syst(x,t):  
    return x[1], -a_1 * a_1 * x[0] - b_1 * x[1] - p(t)
```

Создаем вектор значений

```
v0 = (1, 1.2)
```

```
yf = odeint(syst, v0, t)
```

```
x = []
```

```
y = []
```

```
for i in range(len(yf)):  
    x.append(yf[i][0])  
    y.append(yf[i][1])
```

Показать результаты на дисплее

```
plt.figure(figsize = (10,10))  
plt.plot(x,y,'r', label = 'x')  
plt.show()
```

И получаем: (рис. 3.1)

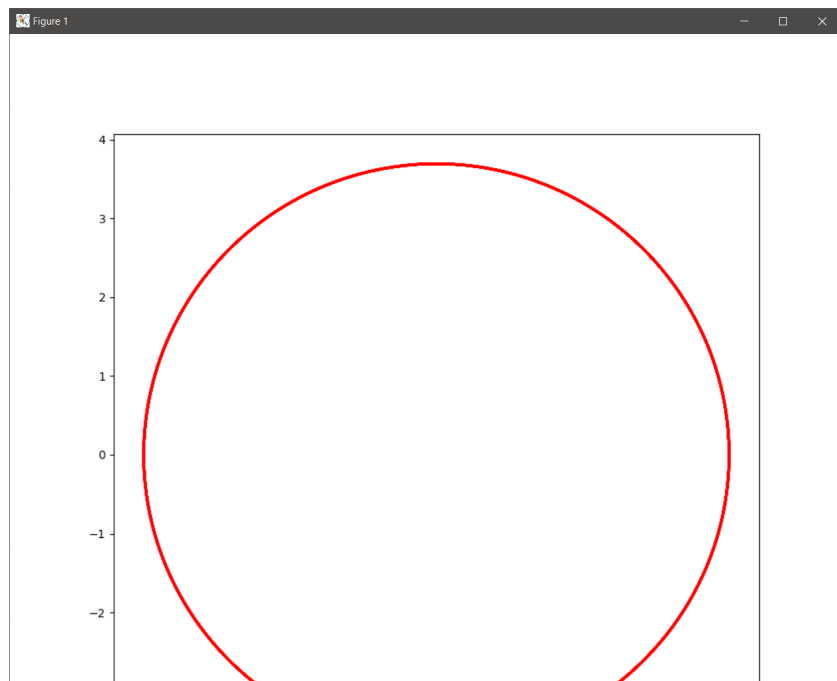


Figure 3.1: Колебания гармонического осциллятора без затуханий и без действий внешней силы, График 1

(рис. 3.2)

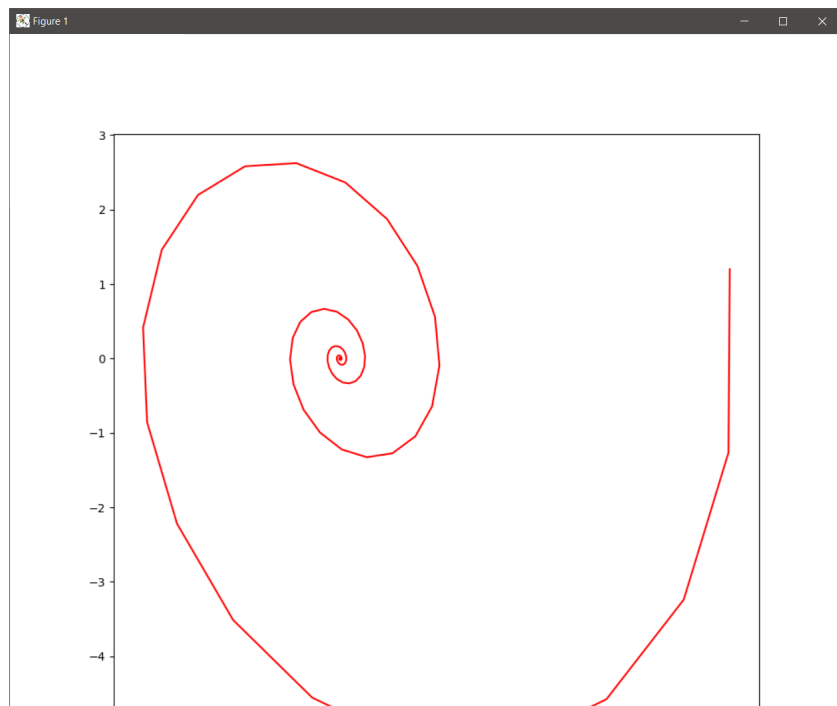


Figure 3.2: Колебания гармонического осциллятора с затуханием и без действий внешней силы, График 2

(рис. 3.3)

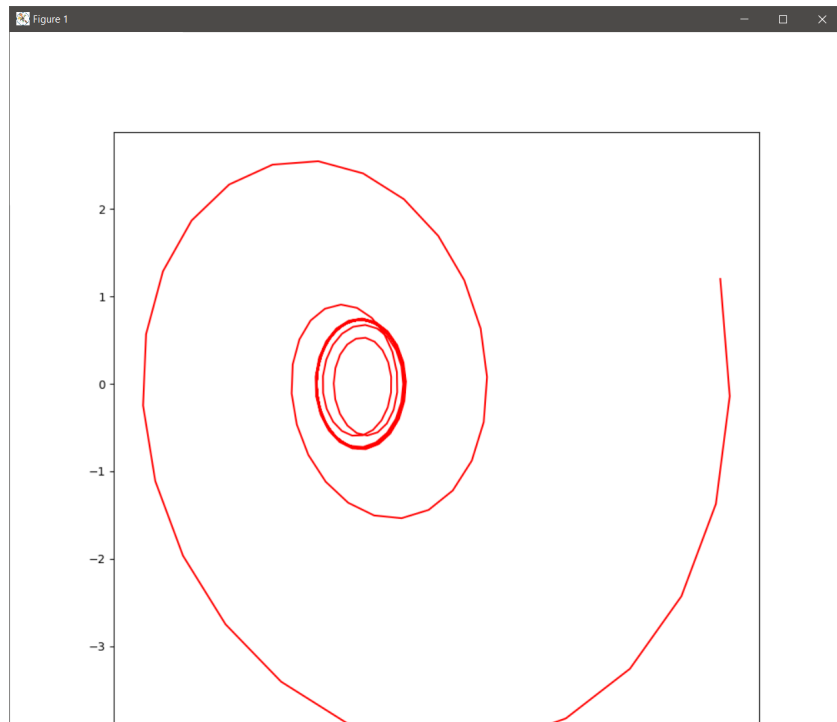


Figure 3.3: Колебания гармонического осциллятора с затуханием и под действием внешней силы, График 3

Код на Python для случая 2:

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

from scipy.integrate import odeint

w = 8
g = 0.00

t0 = 0
tmax = 45
dt = 0.05
```

```
t = np.arange(t0, tmax, dt)
```

```
t = np.append(t, tmax)
```

```
def p(t):
```

```
    #return (math.sin(t*0.5))
```

```
    return 0
```

```
def syst(x, t):
```

```
    return x[1], -w * w * x[0] - g * x[1] - p(t)
```

```
v0 = (-1, 0)
```

```
yf = odeint(syst, v0, t)
```

```
x = []
```

```
y = []
```

```
for i in range(len(yf)):
```

```
    x.append(yf[i][0])
```

```
    y.append(yf[i][1])
```

```
zero = []
```

```
for i in range(len(t)):
```

```
    zero = np.append(zero, 0)
```

```
plt.figure(figsize=(10, 10))
```

```
plt.plot(x, y, 'r', label='x')  
plt.show()
```

Код на Python для случая 1:

```
import numpy as nmp  
import math  
import matplotlib.pyplot as plt  
  
from scipy.integrate import odeint  
  
a_1 = 3.5  
b_1 = 0.00  
  
Time_null = 0  
Time_Max = 37  
Step = 0.05  
  
t = nmp.arange(Time_null, Time_Max, Step)  
t = nmp.append(t, Time_Max)  
  
def p(t):  
    return 0  
  
def syst(x,t):  
    return x[1], -a_1 * a_1 * x[0] - b_1 * x[1] - p(t)  
  
v0 = (1, 1.2)
```

```

yf = odeint(syst, v0, t)

x = []
y = []

for i in range(len(yf)):
    x.append(yf[i][0])
    y.append(yf[i][1])

plt.figure(figsize = (10,10))
plt.plot(x,y,'r', label = 'x')
plt.show()

```

Код на Python для случая 3:

```

import numpy as nmp
import math
import matplotlib.pyplot as plt

from scipy.integrate import odeint

a_1 = 5
b_1 = 2.00

Time_null = 0
Time_Max = 37
Step = 0.05

```

```

t = nmp.arange(Time_null, Time_Max, Step)
t = nmp.append(t, Time_Max)

def p(t):
    return (2 * math.sin(t*6))
    return 0

def syst(x,t):
    return x[1], -a_1 * a_1 * x[0] - b_1 * x[1] - p(t)

v0 = (1, 1.2)

yf = odeint(syst, v0, t)

x = []
y = []

for i in range(len(yf)):
    x.append(yf[i][0])
    y.append(yf[i][1])

plt.figure(figsize = (10,10))
plt.plot(x,y,'r', label = 'x')
plt.show()

```

4 Выводы

Построили код на Python для решения осциллятора и решение уравнения гармонического осциллятора.

5 Ответы на вопросы

Вопросы:

1. Запишите простейшую модель гармонических колебаний
2. Дайте определение осциллятора
3. Запишите модель математического маятника
4. Запишите алгоритм перехода от дифференциального уравнения второго порядка

к двум дифференциальным уравнениям первого порядка

5. Что такое фазовый портрет и фазовая траектория?

Ответы:

1. $\sin(x)$
2. Система, которая при выведении её из положения равновесия испытывает действие возвращающей силы
3. $\ddot{\varphi} + \omega_0^2 \varphi = 0$
4. Дважды интегрируем и получаем общее решение
5.
 - a) Геометрическое представление траекторий динамической системы на фазовой плоскости
 - b) Проекция интегральной кривой