

Отчет по лабораторной работе 4

НФИбд-02-18

Оразклычев Довлет

Содержание

| | | |
|---|--------------------------------|----|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 17 |
| 5 | Ответы на вопросы | 18 |

List of Tables

List of Figures

| | | |
|-----|---------------------------------------|----|
| 2.1 | Задание лабораторной работы | 6 |
| 3.1 | График первого осцилятора | 9 |
| 3.2 | График второго осцилятора | 10 |
| 3.3 | График третьего осцилятора | 11 |

1 Цель работы

Построить модели осциллятора

2 Задание

(рис. 2.1)

Вариант № 6

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

1. Колебания гармонического осциллятора без затуханий и без действий внешней силы $\ddot{x} + 8x = 0$
2. Колебания гармонического осциллятора с затуханием и без действий внешней силы $\ddot{x} + 4\dot{x} + 3x = 0$
3. Колебания гармонического осциллятора с затуханием и под действием внешней силы $\ddot{x} + 3\dot{x} + 6x = \sin(0.5t)$

На интервале $t \in [0; 45]$ (шаг 0.05) с начальными условиями $x_0 = -1, y_0 = 0$

Figure 2.1: Задание лабораторной работы

3 Выполнение лабораторной работы

Для начала мы импортируем библиотеки для построения кода и вводим наши переменные:

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

from scipy.integrate import odeint

w = math.sqrt(6)
g = 3

t0 = 0
tmax = 50
dt = 0.05
```

Теперь мы создаем список значений t , которое мы будем использовать чтобы вычислять поточечно значения “Численность армии”:

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

Обратите внимание, что я также добавил элемент t_{\max} в конец списка. Дело в том, что функция `np.arange` заполняет от нуля до $t_{\max} - dt$, поэтому надо добавлять еще один элемент отдельно.

Теперь создаем непрерывные функции и систему уравнений:

```
def p(t):  
    return (math.sin(t*0.5))  
  
def syst(x, t):  
    return x[1], -w * w * x[0] - g * x[1] - p(t)
```

Создаем вектор значений наших данных и запускаем команду odeint, которая найдет значения поточечно.

```
v0 = (-1, 1)  
  
yf = odeint(syst, v0, t)  
  
x = []  
y = []  
  
for i in range(len(yf)):  
    x.append(yf[i][0])  
    y.append(yf[i][1])
```

Теперь создаем график и выводим на экран. график будет красного цвета с обозначением “х”. Размер графика 10 на 10 единиц.

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

И получаем: (рис. 3.1)

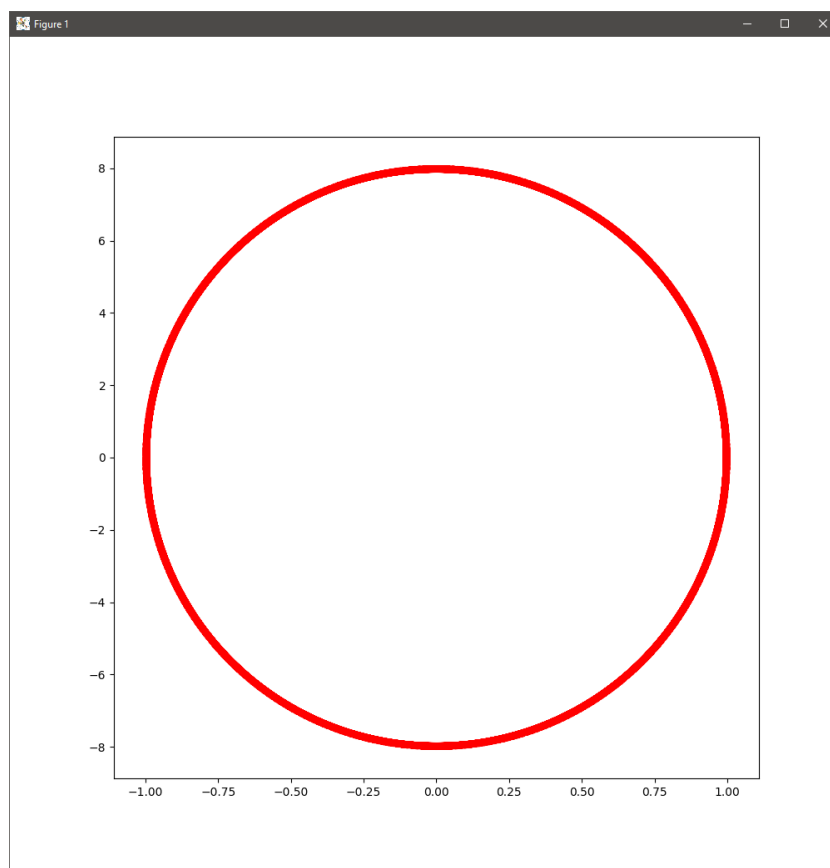


Figure 3.1: График первого осцилятора

(рис. 3.2)

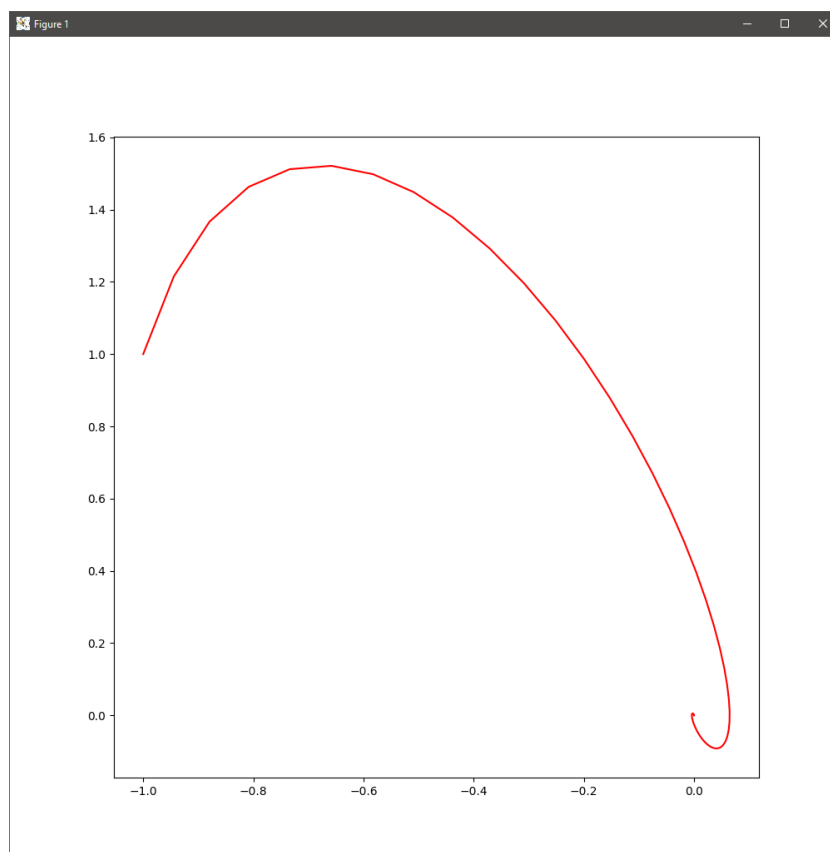


Figure 3.2: График второго осцилятора

(рис. 3.3)

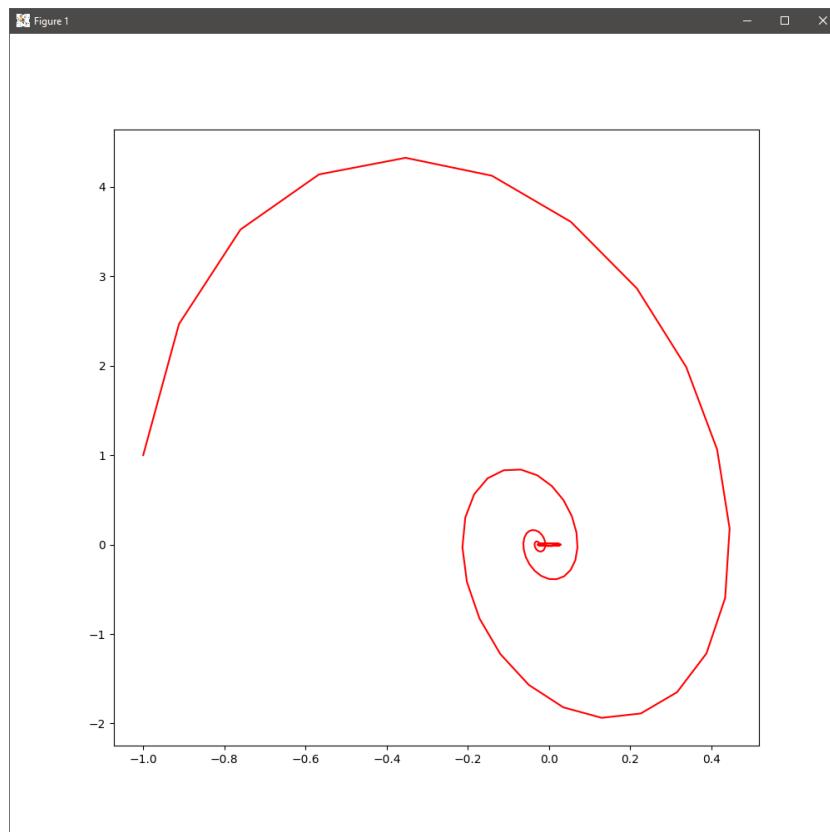


Figure 3.3: График третьего осцилятора

Код на Python для случая 1:

```
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 2:

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

from scipy.integrate import odeint

w = 3
g = 4.00

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 3:

```

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

from scipy.integrate import odeint

```

```

w = 6
g = 3.00

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))

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])

```

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


4 Выводы

Построили код на Python для решения и вывода на экран моделей осцилляторов.

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