

Отчет по лабораторной работе 6

НФИбд-02-18

Оразклычев Довлет

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Вывод	14

List of Tables

List of Figures

2.1	Задание лабораторной работы	6
3.1	График зависимости численности хищников от численности жертв	9
3.2	Стационарное состояние системы	9

1 Цель работы

Постройте графики изменения числа особей в каждой из трех групп.

2 Задание

Вариант 6

На одном острове вспыхнула эпидемия. Известно, что из всех проживающих на острове ($N=12\ 000$) в момент начала эпидемии ($t=0$) число заболевших людей (являющихся распространителями инфекции) $I(0)=212$. А число здоровых людей с иммунитетом к болезни $R(0)=12$. Таким образом, число людей восприимчивых к болезни, но пока здоровых, в начальный момент времени $S(0)=N-I(0)-R(0)$.

Постройте графики изменения числа особей в каждой из трех групп. Рассмотрите, как будет протекать эпидемия в случае:

- 1) если $I(0) \leq I^*$
- 2) если $I(0) > I^*$

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 matplotlib.pyplot as plt

from scipy.integrate import odeint

a = 0.17
b = 0.046

R0 = 12
I0 = 212
N = 12000
S0 = N - I0 - R0

t0 = 0
tmax = 200
dt = 0.01
```

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 `tmax` в конец списка. Дело в том, что функция `np.arange` заполняет от нуля до `tmax - dt`, поэтому надо добавлять еще один элемент отдельно.

Теперь создаем систему уравнений:

```
def syst(x, t):  
    dx1 = 0  
    dx2 = -b * x[1]  
    dx3 = b * x[1]  
    return dx1, dx2, dx3
```

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

```
v0 = (S0, I0, R0)  
yf = odeint(syst, v0, t)
```

```
y1 = []  
y2 = []  
y3 = []
```

```
for i in range(len(yf)):  
    y1.append(yf[i][0])  
    y2.append(yf[i][1])  
    y3.append(yf[i][2])
```

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

```
plt.figure(figsize=(10, 10))  
plt.plot(t, y1, 'r', label='S(t)')  
plt.plot(t, y2, 'b', label='I(t)')
```



```
plt.plot(t, y3, 'g', label='R(t)')
plt.legend( loc = "upper right")
```

```
plt.show()
```

И получаем:

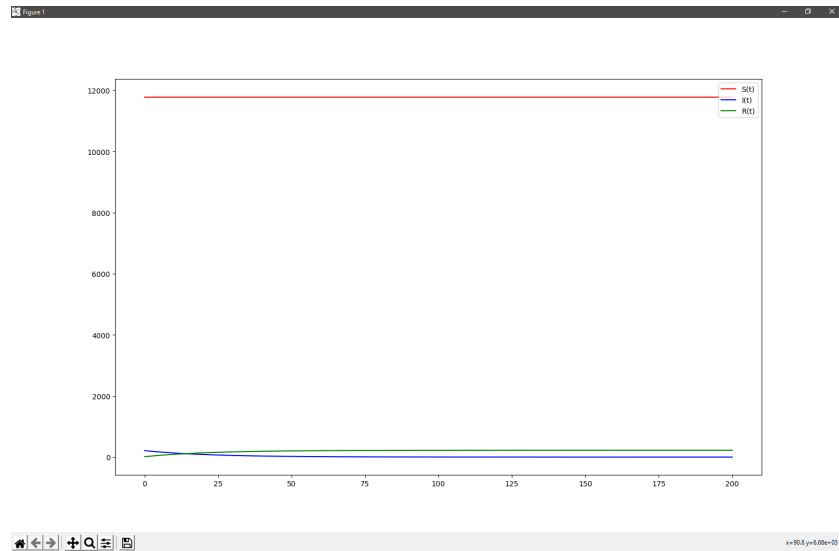


Figure 3.1: График зависимости численности хищников от численности жертв

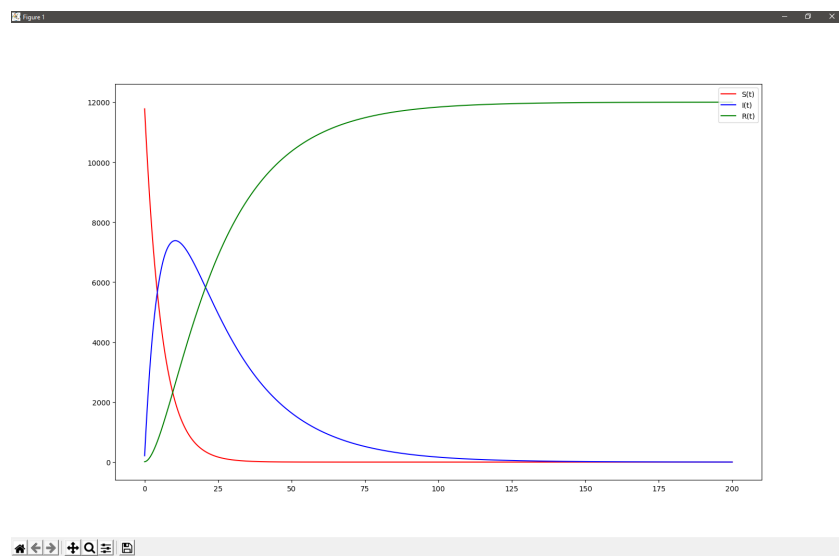


Figure 3.2: Стационарное состояние системы

Код на Python для графика 1:

```

import numpy as np
import matplotlib.pyplot as plt

from scipy.integrate import odeint

a = 0.17
b = 0.046

R0 = 12
I0 = 212
N = 12000
S0 = N - I0 - R0

t0 = 0
tmax = 200
dt = 0.01

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

# I0 < I*
def syst(x, t):
    dx1 = 0
    dx2 = -b * x[1]
    dx3 = b * x[1]
    return dx1, dx2, dx3

v0 = (S0, I0, R0)

```

```

yf = odeint(syst, v0, t)

y1 = []
y2 = []
y3 = []

for i in range(len(yf)):
    y1.append(yf[i][0])
    y2.append(yf[i][1])
    y3.append(yf[i][2])

plt.figure(figsize=(10, 10))
plt.plot(t, y1, 'r', label='S(t)')
plt.plot(t, y2, 'b', label='I(t)')
plt.plot(t, y3, 'g', label='R(t)')
plt.legend( loc = "upper right")

plt.show()

```

Код на Python для графика 2:

```

import numpy as np
import matplotlib.pyplot as plt

from scipy.integrate import odeint

a = 0.17
b = 0.046

R0 = 12
I0 = 212

```

```
N = 12000
```

```
S0 = N - I0 - R0
```

```
t0 = 0
```

```
tmax = 200
```

```
dt = 0.01
```

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

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

```
# I0 > I*
```

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

```
    dx1 = -a * x[0]
```

```
    dx2 = a*x[0] - b * x[1]
```

```
    dx3 = b * x[1]
```

```
    return dx1, dx2, dx3
```

```
v0 = (S0, I0, R0)
```

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

```
y1 = []
```

```
y2 = []
```

```
y3 = []
```

```
for i in range(len(yf)):
    y1.append(yf[i][0])
    y2.append(yf[i][1])
    y3.append(yf[i][2])

plt.figure(figsize=(10, 10))
plt.plot(t, y1, 'r', label='S(t)')
plt.plot(t, y2, 'b', label='I(t)')
plt.plot(t, y3, 'g', label='R(t)')
plt.legend( loc = "upper right")

plt.show()
```

4 Вывод

Построили код на Python для решения и вывода на экран графиков изменения числа особей в каждой из трех групп.