

Отчет по лабораторной работе 7

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	Случай 1	8
3.2	Случай 2	9
3.3	Случай 3	9

1 Цель работы

Постройте графики изменения числа особей в каждой из трех групп.

2 Задание

Вариант № 6

Постройте график распространения рекламы, математическая модель которой описывается следующим уравнением:

1. $\frac{dn}{dt} = (0.99 + 0.00012n(t))(N - n(t))$
2. $\frac{dn}{dt} = (0.000067 + 0.38n(t))(N - n(t))$

3. $\frac{dn}{dt} = (0.6\sin(4t) + 0.1\cos(2t)n(t))(N - n(t))$

При этом объем аудитории $N = 777$, в начальный момент о товаре знает 1 человек. Для случая 2 определите в какой момент времени скорость распространения рекламы будет иметь максимальное значение.

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
```

```
N = 777
x0 = 1
```

```
t0 = 0
tmax = 30
dt = 0.1
```

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 систему уравнений:

```
def k(t):  
    return 0.99  
def p(t):  
    return 0.00012  
def f(x, t):  
    return (k(t) + p(t)*x)*(N-x)
```

Запускаем команду odeint, которая найдет значения поточечно.

```
yf = odeint(f, x0, t)
```

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, yf, 'r', label='S(t)')  
plt.show()
```

И получаем:

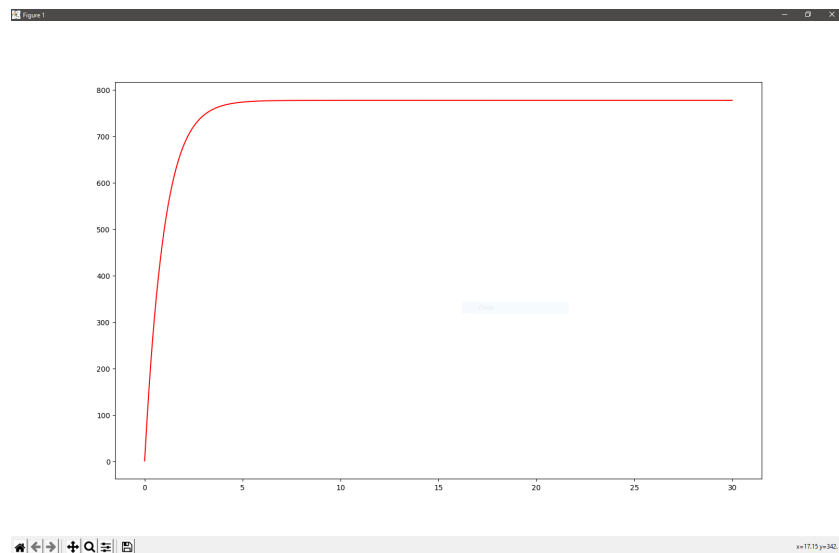


Figure 3.1: Случай 1

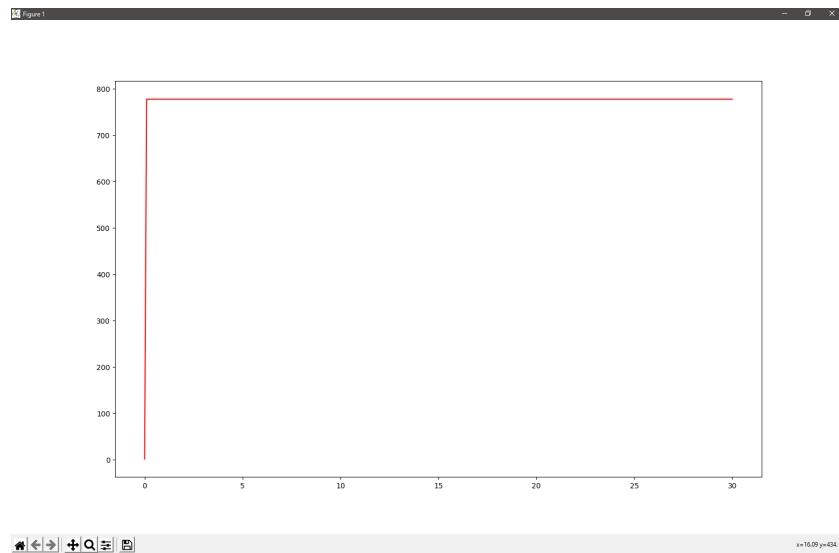


Figure 3.2: Случай 2

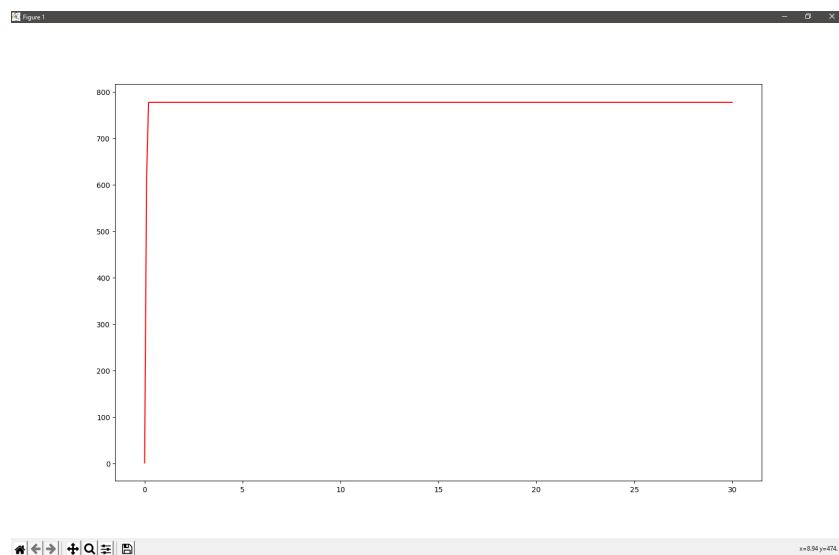


Figure 3.3: Случай 3

Код на Python для графика 1:

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

from scipy.integrate import odeint
```

```

N = 777
x0 = 1

t0 = 0
tmax = 30
dt = 0.1

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

def k(t):
    return 0.99
def p(t):
    return 0.00012
def f(x, t):
    return (k(t) + p(t)*x)*(N-x)

yf = odeint(f, x0, t)

plt.figure(figsize=(10, 10))
plt.plot(t, yf, 'r', label='S(t)')
plt.show()

```

Код на Python для графика 2:

```

import numpy as np
import matplotlib.pyplot as plt

```

```
from scipy.integrate import odeint
```

```
N = 777
```

```
x0 = 1
```

```
t0 = 0
```

```
tmax = 30
```

```
dt = 0.1
```

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

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

```
def k(t):
```

```
    return 0.000067
```

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

```
    return 0.38
```

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

```
    return (k(t) + p(t)*x)*(N-x)
```

```
yf = odeint(f, x0, t)
```

```
plt.figure(figsize=(10, 10))
plt.plot(t, yf, 'r', label='S(t)')
plt.show()
```

Код на Python для графика 3:

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

from scipy.integrate import odeint

N = 777
x0 = 1

t0 = 0
tmax = 30
dt = 0.1

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

def k(t):
    return 0.6*math.sin(4*t)

def p(t):
    return 0.1*math.cos(2*t)
```

```
def f(x, t):  
    return (k(t) + p(t)*x)*(N-x)  
  
yf = odeint(f, x0, t)  
  
plt.figure(figsize=(10, 10))  
plt.plot(t, yf, 'r', label='S(t)')  
plt.show()
```

4 Вывод

Построили код на Python для решения и вывода на экран графиков эффективности рекламы для 3 случаев.