Отчет по лабораторной работе 7

НФИбд-02-18

Оразклычев Довлет

Содержание

# Цель работы

Постройте графики изменения числа особей в каждой из трех групп.

# Задание

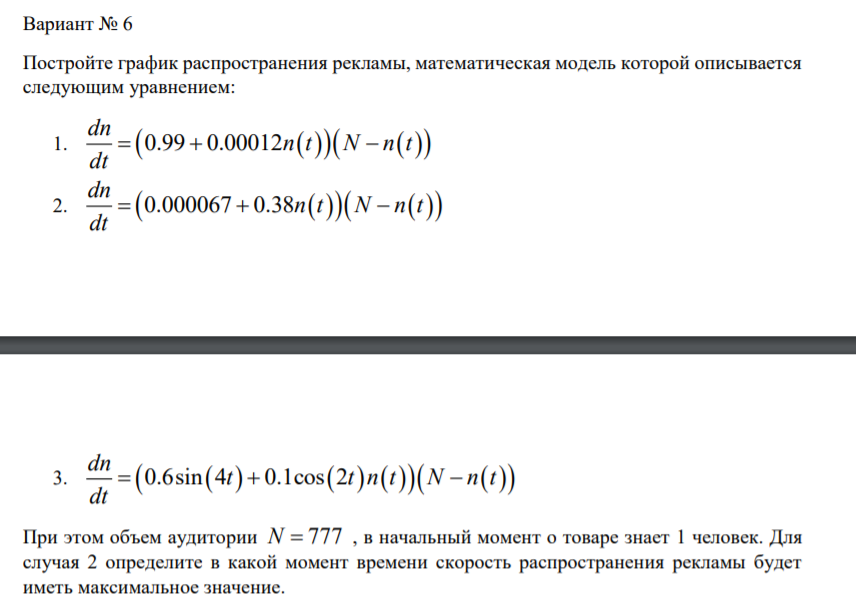


Figure 1: Задание лабораторной работы

# 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 внимаение, 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 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:

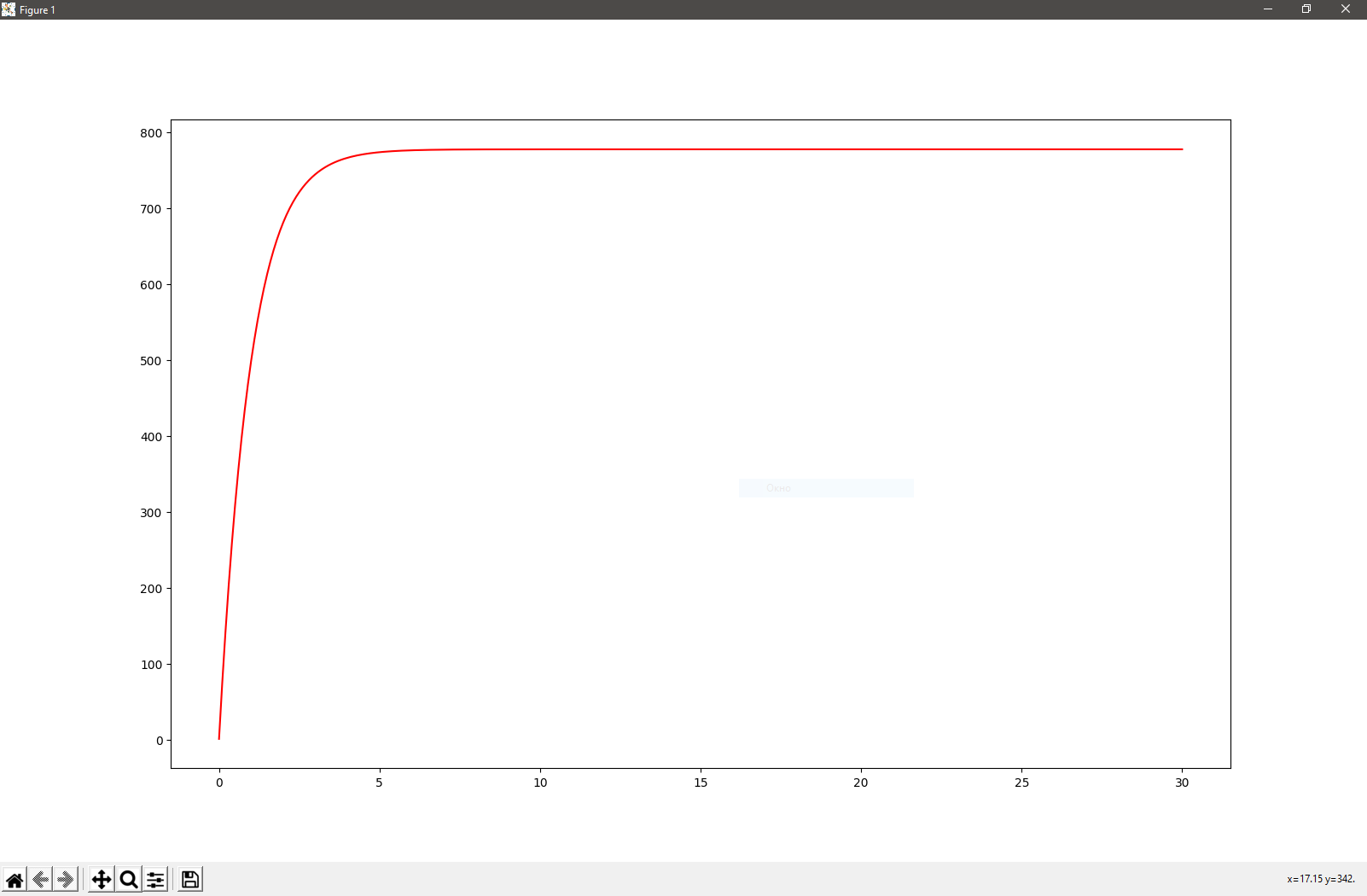


Figure 2: Случай 1

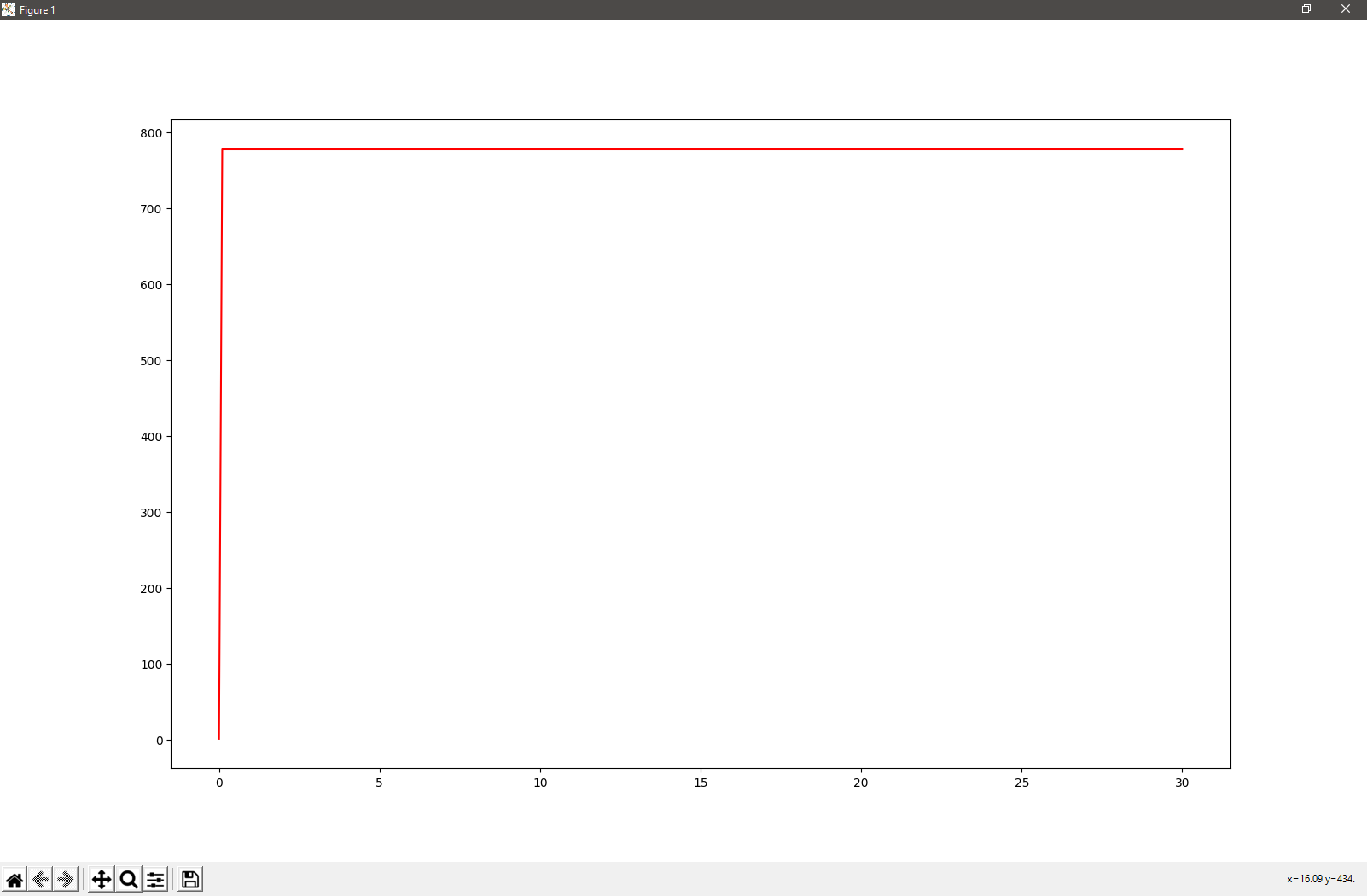


Figure 3: Случай 2

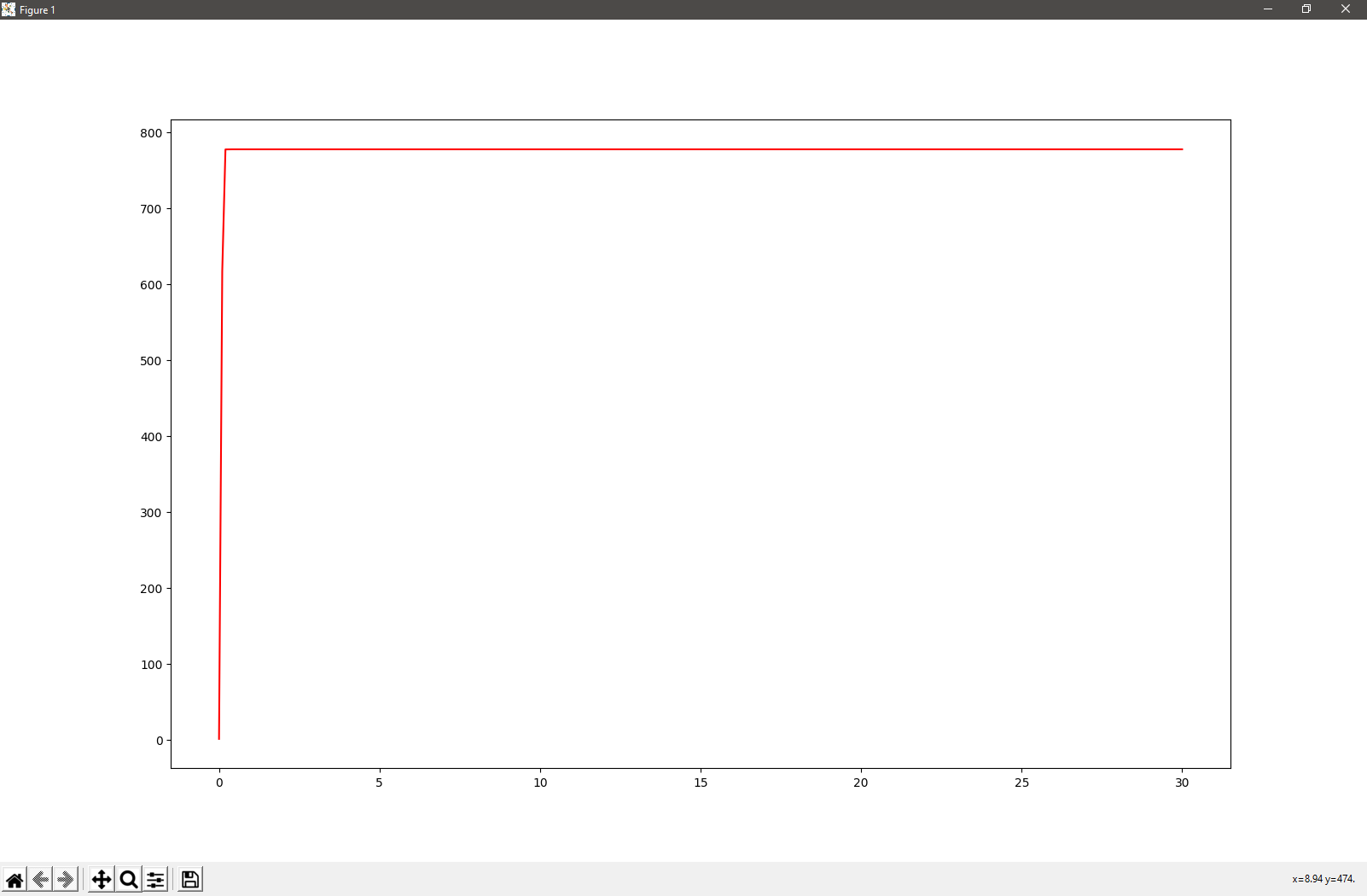


Figure 4: Случай 3

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()

# Вывод

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