Отчет по лабораторной работе 8

НФИбд-02-18

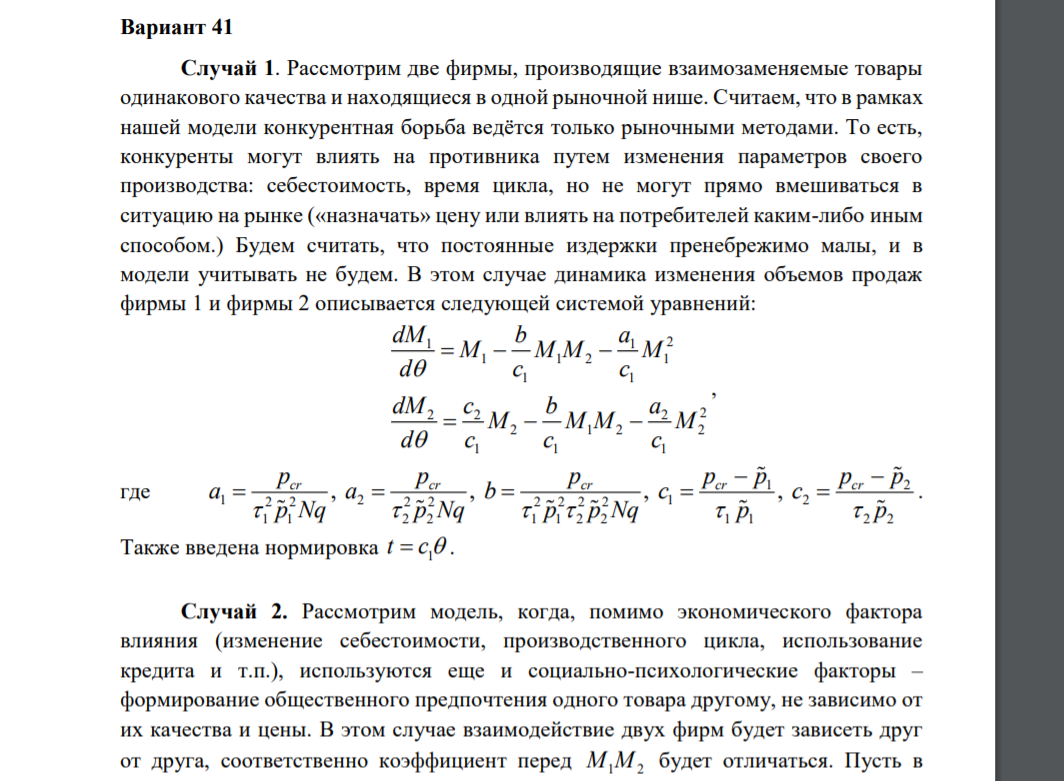
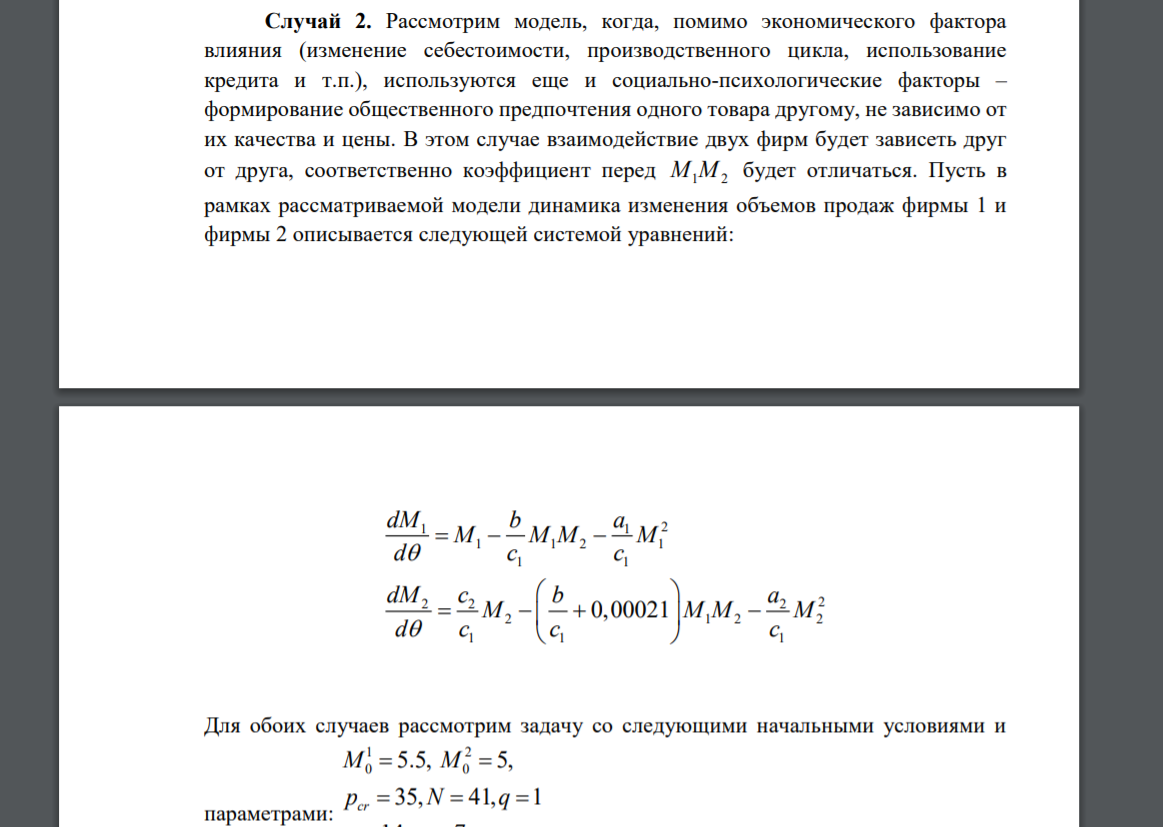
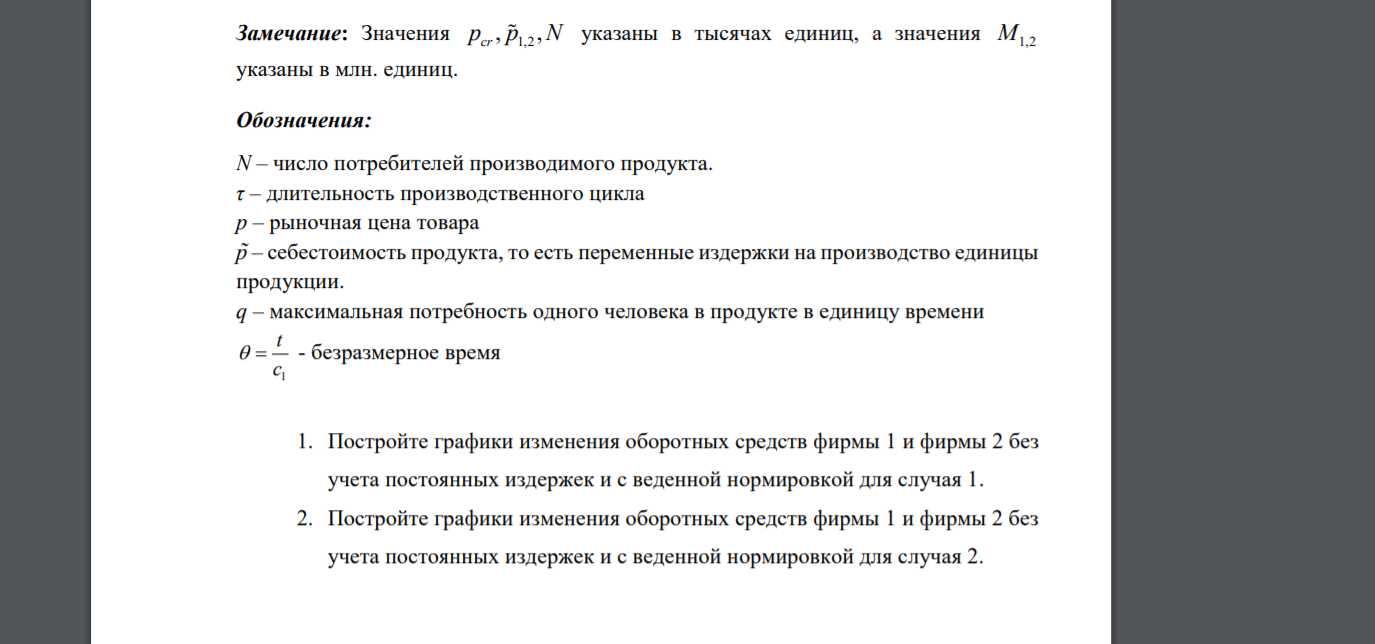
Оразклычев Давут

Содержание

# Цель работы

Постройте графики изменения числа особей в каждой из трех групп.

# Задание

# 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  
  
p\_cr\_8 = 35  
N\_\_8 = 41  
q\_8 = 1  
  
tau1\_8 = 14  
tau2\_8 = 7  
  
p1\_8 = 6.5  
p2\_8 = 15

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

a1\_8 = p\_cr\_8 / (tau1\_8 \* tau1\_8 \* p1\_8 \* p1\_8 \* N\_\_8 \*q\_8)  
a2\_8 = p\_cr\_8 / (tau2\_8 \* tau1\_8 \* p2\_8 \* p1\_8 \* N\_\_8 \*q\_8)  
b\_8 = p\_cr\_8 / (tau1\_8 \* tau1\_8 \* tau2\_8 \* tau2\_8 \* p1\_8 \* p1\_8\*p2\_8\*p2\_8\*N\_\_8\*q\_8)  
c1\_8 = (p\_cr\_8 - p1\_8)/ (tau1\_8\*tau1\_8)  
c2\_8 = (p\_cr\_8 - p1\_8)/ (tau2\_8\*tau2\_8)  
  
t0\_8 = 0  
tmax\_8 = 30  
dt\_8 = 0.01  
  
  
t\_8 = np.arange(t0\_8,tmax\_8,dt\_8)  
t\_8 = np.append(t\_8,tmax\_8)

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 f(x,t\_8):  
 dx1\_8 = (c1\_8/c1\_8)\* x[0] - (a1\_8/c1\_8)\*x[0]\*x[0] - (b\_8/c1\_8) \* x[0] \* x[1]  
 dx2\_8 = (c2\_8/c2\_8)\* x[1] - (a2\_8/c1\_8)\*x[1]\*x[1] - (b\_8/c1\_8) \* x[0] \* x[1]  
 return dx1\_8,dx2\_8

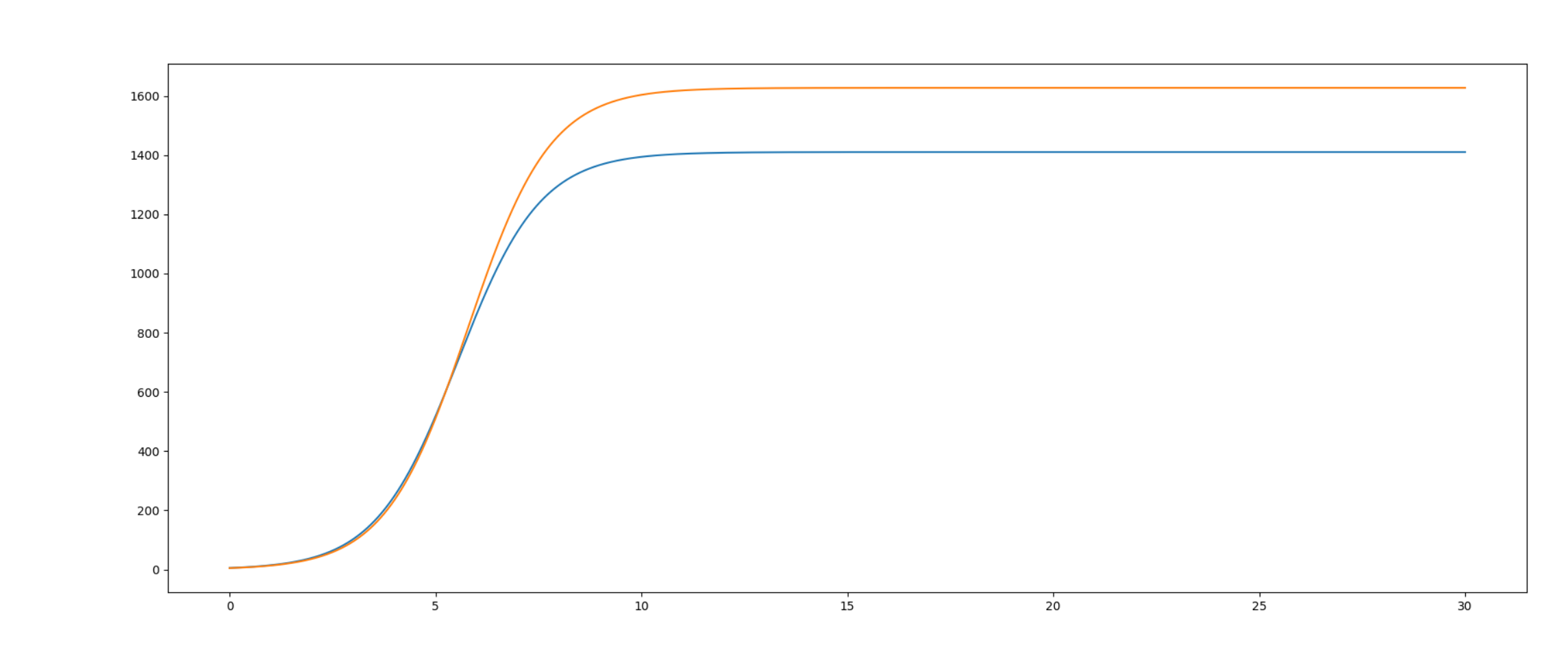
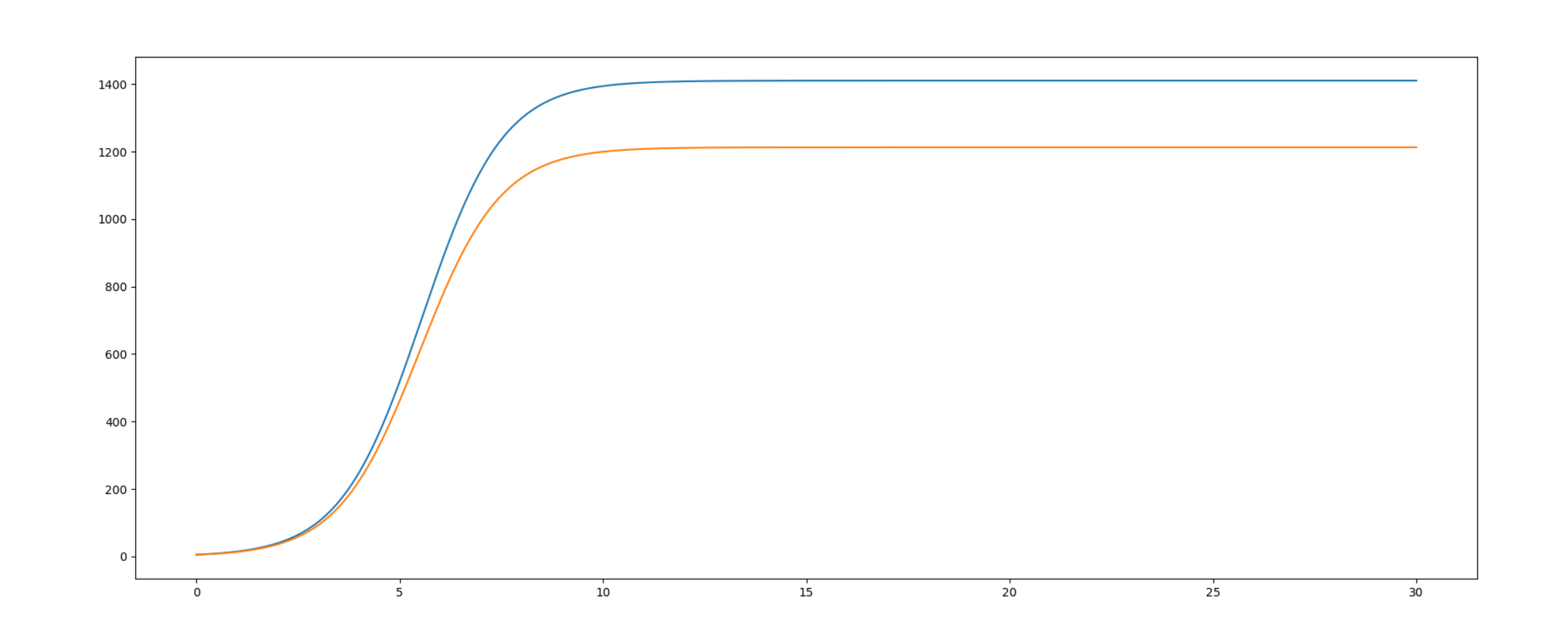
Запускаем команду odeint, которая найдет значения поточечно.

x0\_8 =[5.5,5]  
yf\_8 = odeint(f,x0\_8,t\_8)

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\_8,yf\_8)  
plt.show()

И получаем:

Код:

import numpy as np  
import matplotlib.pyplot as plt  
  
from scipy.integrate import odeint  
  
p\_cr\_8 = 35  
N\_\_8 = 41  
q\_8 = 1  
  
tau1\_8 = 14  
tau2\_8 = 7  
  
p1\_8 = 6.5  
p2\_8 = 15  
  
  
a1\_8 = p\_cr\_8 / (tau1\_8 \* tau1\_8 \* p1\_8 \* p1\_8 \* N\_\_8 \*q\_8)  
a2\_8 = p\_cr\_8 / (tau2\_8 \* tau1\_8 \* p2\_8 \* p1\_8 \* N\_\_8 \*q\_8)  
b\_8 = p\_cr\_8 / (tau1\_8 \* tau1\_8 \* tau2\_8 \* tau2\_8 \* p1\_8 \* p1\_8\*p2\_8\*p2\_8\*N\_\_8\*q\_8)  
c1\_8 = (p\_cr\_8 - p1\_8)/ (tau1\_8\*tau1\_8)  
c2\_8 = (p\_cr\_8 - p1\_8)/ (tau2\_8\*tau2\_8)  
  
t0\_8 = 0  
tmax\_8 = 30  
dt\_8 = 0.01  
  
  
t\_8 = np.arange(t0\_8,tmax\_8,dt\_8)  
t\_8 = np.append(t\_8,tmax\_8)  
  
def f(x,t\_8):  
 dx1\_8 = (c1\_8/c1\_8)\* x[0] - (a1\_8/c1\_8)\*x[0]\*x[0] - (b\_8/c1\_8) \* x[0] \* x[1]  
 dx2\_8 = (c2\_8/c2\_8)\* x[1] - (a2\_8/c1\_8)\*x[1]\*x[1] - (b\_8/c1\_8) \* x[0] \* x[1]  
 return dx1\_8,dx2\_8  
  
x0\_8 =[5.5,5]  
yf\_8 = odeint(f,x0\_8,t\_8)  
  
plt.figure(figsize = (10,10))  
plt.plot (t\_8,yf\_8)  
plt.show()

# Вывод

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