Отчет по лабораторной работе 8

НФИбд-02-18

Оразклычев Довлет

Содержание

# Цель работы

Постройте графики фирм для двух случаев

# Задание

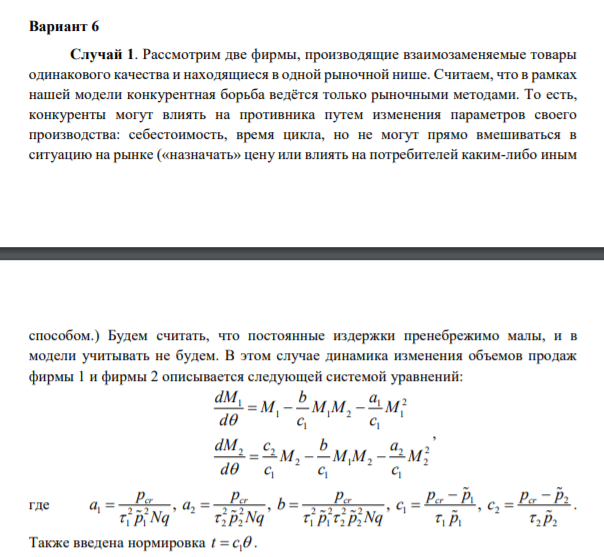


Figure 1: Задание лабораторной работы

# 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  
  
p\_cr = 20  
tau1 = 10  
p1 = 9  
tau2 = 16  
p2 = 7  
V = 10  
q = 1  
  
a1 = p\_cr / (tau1 \* tau1 \* p1 \* p1 \* V \* q)  
a2 = p\_cr / (tau2 \* tau2 \* p2 \* p2 \* V \* q)  
b = p\_cr / (tau1 \* tau1 \* tau2 \* tau2 \* p1 \* p1 \* p2 \* p2 \* V \* q)  
c1 = (p\_cr - p1) / (tau1 \* p1)  
c2 = (p\_cr - p1) / (tau2 \* p2)  
  
t0 = 0  
tmax = 30  
dt = 0.01

Теперь мы создаем список значений t, которое мы будем использовать чтобы вычислять поточечно значения:

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 f(x, t):  
 dx1 = (c1 / c1) \* x[0] - ((a1 / c1) + 0.0015) \* x[0] \* x[0] - (b / c1) \* x[0] \* x[1]  
 dx2 = (c2 / c1) \* x[1] - (a2 / c1) \* x[1] \* x[1] - (b / c1) \* x[0] \* x[1]  
 return dx1, dx2

Запускаем команду odeint, которая найдет значения поточечно.

yf = odeint(f, v0, 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)  
plt.show()

И получаем:

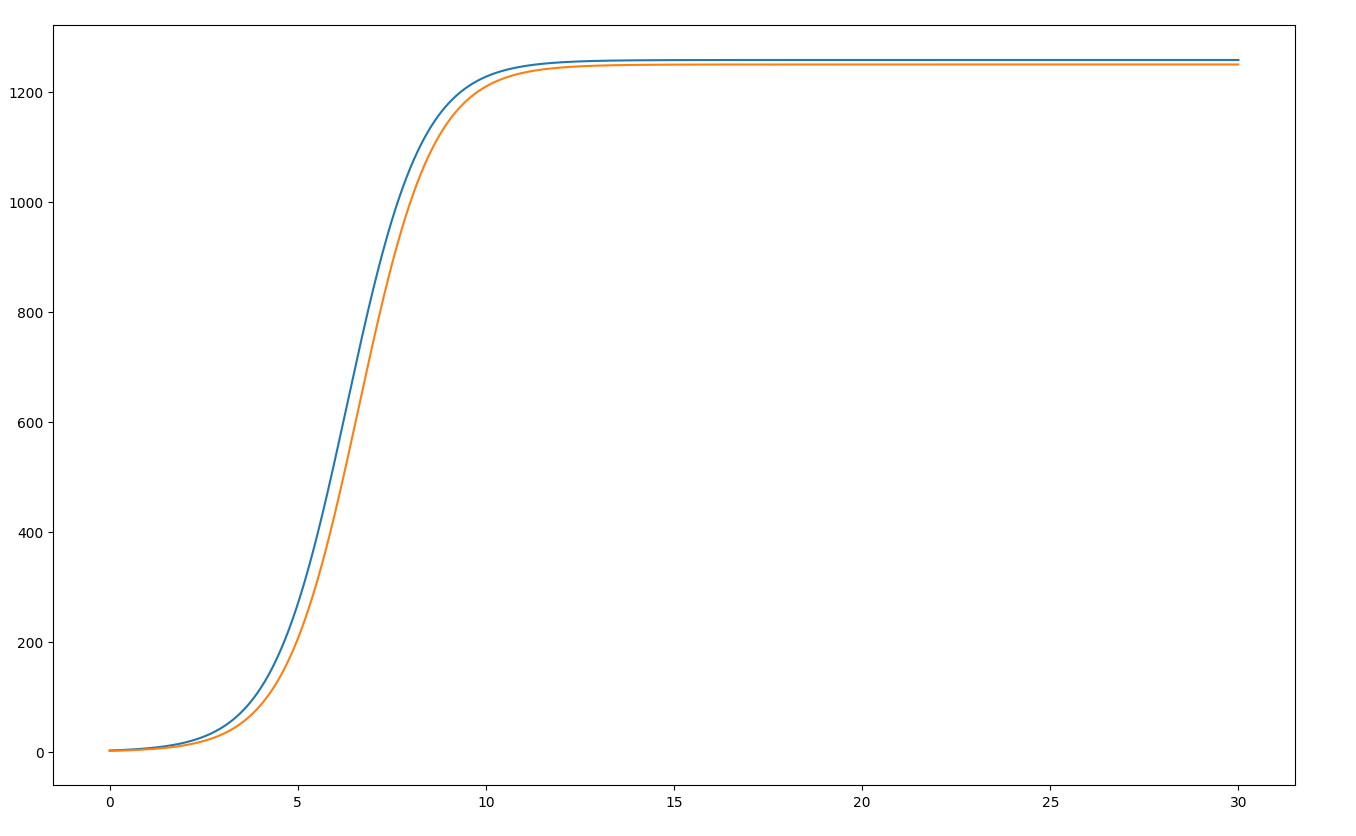


Figure 2: Случай 1

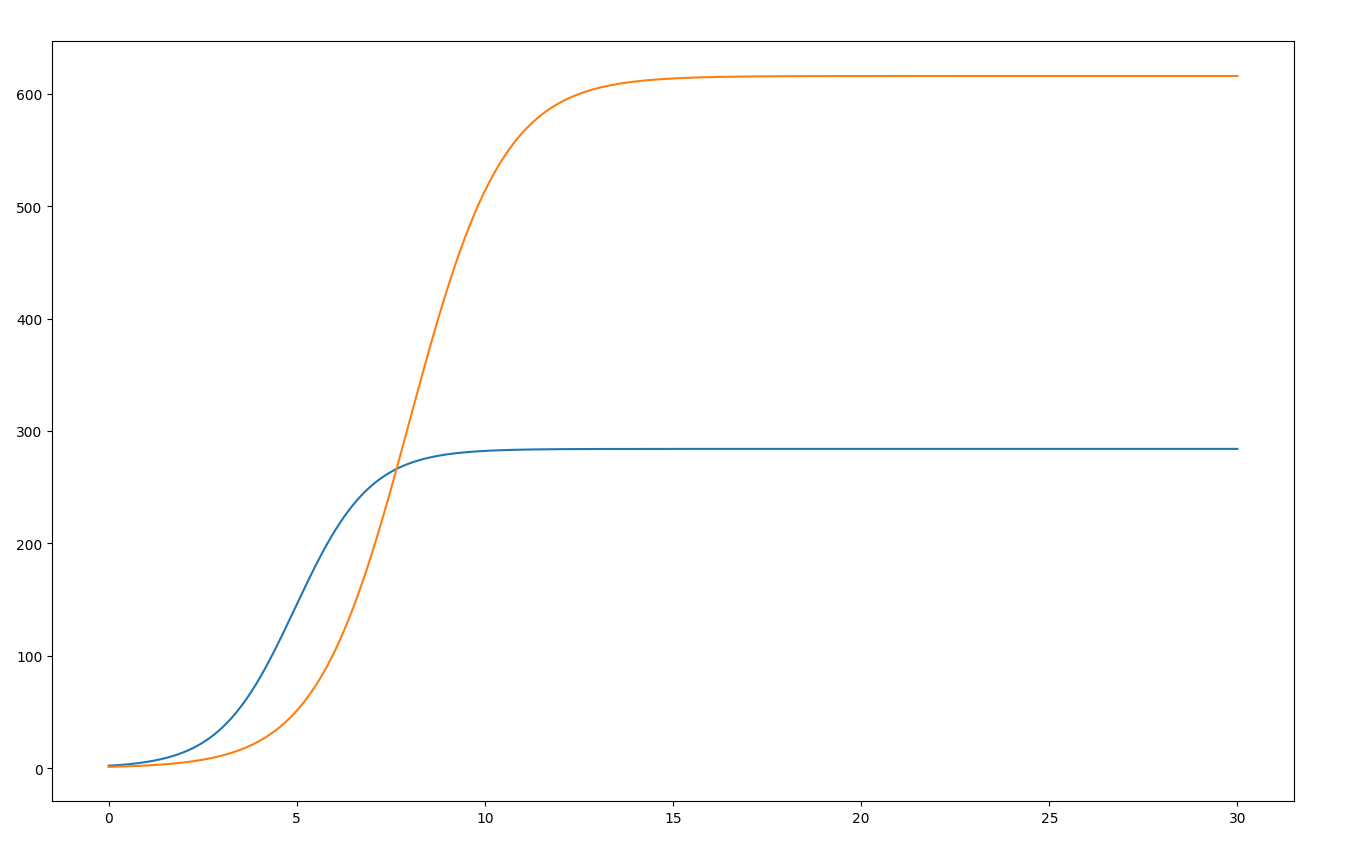


Figure 3: Случай 2

Код на Python для графика 1:

import numpy as np  
import matplotlib.pyplot as plt  
  
from scipy.integrate import odeint  
  
p\_cr = 18  
tau1 = 14  
p1 = 11  
tau2 = 17  
p2 = 9  
V = 21  
q = 1  
  
a1 = p\_cr / (tau1 \* tau1 \* p1 \* p1 \* V \* q)  
a2 = p\_cr / (tau2 \* tau2 \* p2 \* p2 \* V \* q)  
b = p\_cr / (tau1 \* tau1 \* tau2 \* tau2 \* p1 \* p1 \* p2 \* p2 \* V \* q)  
c1 = (p\_cr - p1) / (tau1 \* p1)  
c2 = (p\_cr - p1) / (tau2 \* p2)  
  
  
t0 = 0  
tmax = 30  
dt = 0.01  
  
t = np.arange(t0, tmax, dt)  
t = np.append(t, tmax)  
  
  
def f(x, t):  
 dx1 = (c1 / c1) \* x[0] - (a1 / c1) \* x[0] \* x[0] - (b / c1) \* x[0] \* x[1]  
 dx2 = (c2 / c1) \* x[1] - (a2 / c1) \* x[1] \* x[1] - (b / c1) \* x[0] \* x[1]  
 return dx1,dx2  
  
  
x0 = [2.3,1.6]  
yf = odeint(f, x0, t)  
  
  
plt.figure(figsize=(10, 10))  
plt.plot(t, yf)  
plt.show()

Код на Python для графика 2:

import numpy as np  
import matplotlib.pyplot as plt  
  
from scipy.integrate import odeint  
  
p\_cr = 20  
tau1 = 10  
p1 = 9  
tau2 = 16  
p2 = 7  
V = 10  
q = 1  
  
a1 = p\_cr / (tau1 \* tau1 \* p1 \* p1 \* V \* q)  
a2 = p\_cr / (tau2 \* tau2 \* p2 \* p2 \* V \* q)  
b = p\_cr / (tau1 \* tau1 \* tau2 \* tau2 \* p1 \* p1 \* p2 \* p2 \* V \* q)  
c1 = (p\_cr - p1) / (tau1 \* p1)  
c2 = (p\_cr - p1) / (tau2 \* p2)  
  
t0 = 0  
tmax = 30  
dt = 0.01  
  
t = np.arange(t0, tmax, dt)  
t = np.append(t, tmax)  
v0 = [2, 1]  
  
  
def f(x, t):  
 dx1 = (c1 / c1) \* x[0] - ((a1 / c1) + 0.0015) \* x[0] \* x[0] - (b / c1) \* x[0] \* x[1]  
 dx2 = (c2 / c1) \* x[1] - (a2 / c1) \* x[1] \* x[1] - (b / c1) \* x[0] \* x[1]  
 return dx1, dx2  
  
  
yf = odeint(f, v0, t)  
  
plt.figure(figsize=(10, 10))  
plt.plot(t, yf)  
plt.show()

# Вывод

Построили код на Python для решения и вывода на экран графиков для 2 случаев.