

# Презентация лабораторной работы 8

---

Оразклычев Довлет<sup>1</sup>

2020-2021 г., Москва

<sup>1</sup>RUDN University, Moscow, Russian Federation

# Знакомство с задачей

---

## Вариант 6

**Случай 1.** Рассмотрим две фирмы, производящие взаимозаменяемые товары одинакового качества и находящиеся в одной рыночной нише. Считаем, что в рамках нашей модели конкурентная борьба ведётся только рыночными методами. То есть, конкуренты могут влиять на противника путем изменения параметров своего производства: себестоимость, время цикла, но не могут прямо вмешиваться в ситуацию на рынке («назначать» цену или влиять на потребителей каким-либо иным

способом.) Будем считать, что постоянные издержки пренебрежимо малы, и в модели учитывать не будем. В этом случае динамика изменения объемов продаж фирмы 1 и фирмы 2 описывается следующей системой уравнений:

$$\begin{aligned}\frac{dM_1}{d\theta} &= M_1 - \frac{b}{c_1} M_1 M_2 - \frac{a_1}{c_1} M_1^2 \\ \frac{dM_2}{d\theta} &= \frac{c_2}{c_1} M_2 - \frac{b}{c_1} M_1 M_2 - \frac{a_2}{c_1} M_2^2\end{aligned}$$

где  $a_1 = \frac{P_{\sigma}}{\tau_1^2 \tilde{p}_1^2 Nq}$ ,  $a_2 = \frac{P_{\sigma}}{\tau_2^2 \tilde{p}_2^2 Nq}$ ,  $b = \frac{P_{\sigma}}{\tau_1^2 \tilde{p}_1^2 \tau_2^2 \tilde{p}_2^2 Nq}$ ,  $c_1 = \frac{P_{\sigma} - \tilde{p}_1}{\tau_1 \tilde{p}_1}$ ,  $c_2 = \frac{P_{\sigma} - \tilde{p}_2}{\tau_2 \tilde{p}_2}$ .

Также введена нормировка  $t = c_1 \theta$ .

Figure 1: Задание

## Добавление библиотек и переменных

```
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
```

```
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] *
```

```
    dx2 = (c2 / c1) * x[1] - (a2 / c1) * x[1] * x[1] - (b / c1) * x[0] * x[1]
```

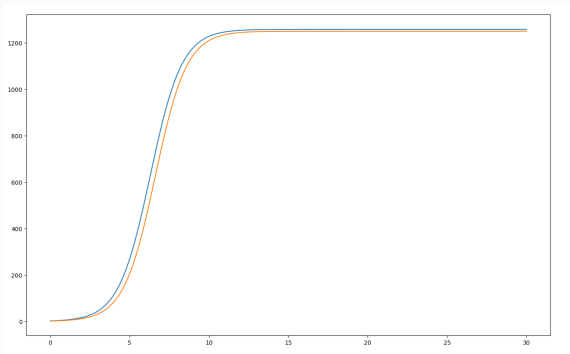
```
    return dx1, dx2
```

## Находим значения для графика

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

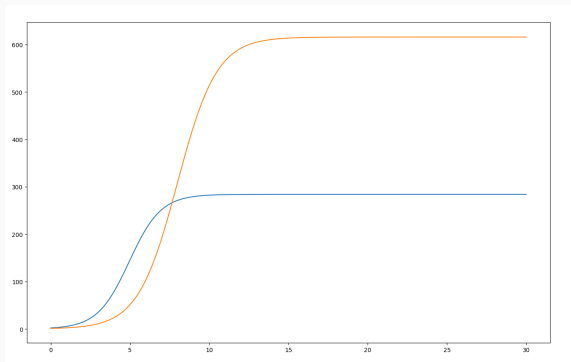
```
plt.figure(figsize=(10, 10))  
plt.plot(t, yf)  
plt.show()
```

# График №1



**Figure 2:** График №1





**Figure 3:** График №2