

# Лабораторная работа №6. Модель эпидемии

с/б 1032186063 | НФИбд-01-18

---

Доборщук Владимир Владимирович

20 марта 2021

RUDN University, Moscow, Russian Federation

## Цели и задачи

---

Изучить простейшую модель эпидемии, а также реализовать её программно..

- изучить теорию о модели эпидемии
- построить модель для случаев:
  - $I(t) \geq I^*$
  - $I(t) \leq I^*$

# **Ход выполнения лабораторной работы**

---

Рассмотрим простейшую модель эпидемии. Предположим, что некая популяция, состоящая из  $N$  особей, (считаем, что популяция изолирована) подразделяется на три группы. Первая группа - это восприимчивые к болезни, но пока здоровые особи, обозначим их через  $S(t)$ . Вторая группа – это число инфицированных особей, которые также при этом являются распространителями инфекции, обозначим их  $I(t)$ . А третья группа, обозначающаяся через  $R(t)$  – это здоровые особи с иммунитетом к болезни.

До того, как число заболевших не превышает критического значения  $I^*$ , считаем, что все больные изолированы и не заражают здоровых. Когда  $I(t) \geq I$ , тогда инфицирование способны заражать восприимчивых к болезни особей. Для этого случая:

$$\frac{dS}{dt} = -\alpha S$$

$$\frac{dI}{dt} = \alpha S - \beta I$$

$$\frac{dR}{dt} = \beta I$$

Для случая, когда  $I(t) \leq I$ :

$$\frac{dS}{dt} = 0$$

$$\frac{dI}{dt} = -\beta I$$

$$\frac{dR}{dt} = \beta I$$



**Вариант 14:**  $(1032186063 \bmod 70) + 1$

$$N = 19500$$

$$I_0 = 88$$

$$R_0 = 25$$

$$\alpha = 0.01, \beta = 0.02$$

## Инициализация библиотек

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4
5 from jupyterthemes import jtplot
6 jtplot.style(context='notebook', fscale=1.2,
               spines=False, gridlines='—')
```

Введём соответствующие нашему варианту начальные данные для построения модели:

```
1 N = 19500
2 t0 = 0
3 I0 = 88
4 R0 = 25
5 S0 = N - R0 - I0
6
7 alpha = 0.01
8 beta = 0.02
9
10 x0 = np.array([S0, I0, R0])
11 t = np.arange(0, 200, 0.01)
```

Создадим функцию для наших СДУ:

```
1  def dx_less(x,t):
2      dS_dt = 0.0
3      dI_dt = -beta*x[1]
4      dR_dt = beta*x[1]
5      return [dS_dt, dI_dt, dR_dt]
6
7  def dx_greater(x,t):
8      dS_dt = -alpha*x[0]
9      dI_dt = alpha*x[0] - beta*x[1]
10     dR_dt = beta*x[1]
11     return [dS_dt, dI_dt, dR_dt]
```

Воспользуемся функцией `odeint` из модуля `scipy.integrate` и решим наши СДУ, после чего выделим значения для первого и второго случая в две разные переменные

```
1 y_less = odeint(dx_less, x0, t)
2 y_greater = odeint(dx_greater, x0, t)
```

## Модель при $I(t) \leq I^*$

```
1 S = [e for e in y_less[:,0]]
2 I = [e for e in y_less[:,1]]
3 R = [e for e in y_less[:,2]]
4
5 plt.plot(t, S, label='Без иммунитета')
6 plt.plot(t, I, label='Зараженные')
7 plt.plot(t, R, label='С иммунитетом')
8 plt.ylabel('Число зараженных')
9 plt.xlabel('Время')
10 plt.legend(loc='best')
11 plt.show()
```

# Модель при $I(t) \leq I^*$

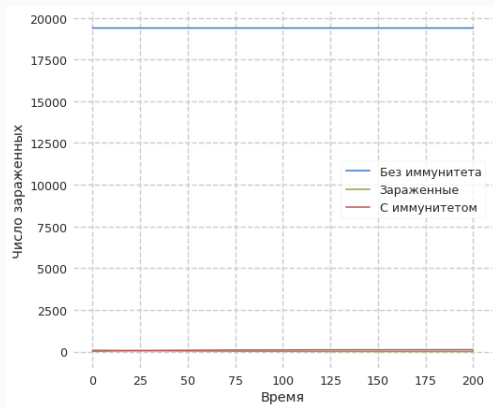


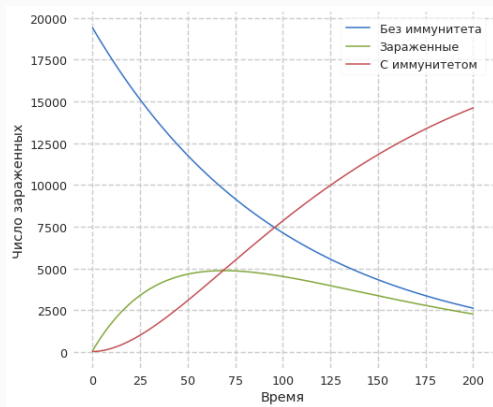
Рис. 1: Модель при  $I(t) \leq I^*$

## Модель при $I(t) \geq I^*$

```
1 S = [e for e in y_greater[:,0]]
2 I = [e for e in y_greater[:,1]]
3 R = [e for e in y_greater[:,2]]
4
5 plt.plot(t, S, label='Без иммунитета')
6 plt.plot(t, I, label='Зараженные')
7 plt.plot(t, R, label='С иммунитетом')
8 plt.legend(loc='best')
9 plt.ylabel('Число зараженных')
10 plt.xlabel('Время')
11 plt.show()
```



## Модель при $I(t) \geq I^*$



**Рис. 2:** Модель при  $I(t) \geq I^*$

## **Выводы**

---

Мы изучили простейшую модель эпидемии, после чего успешно реализовали её с помощью языка Python и дополняющих его модулей.