



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
**(ДВФУ)**

---

## **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Департамент математического и компьютерного моделирования**

### **Реферат**

по дисциплине

«Уравнения математической физики»

по теме:

«Математическая модель переноса вирусов в виде ОДУ»

направление подготовки 02.03.01 «Математика и компьютерные науки»

направление образовательной программы

«Сквозные цифровые технологии»

Выполнил студент

гр. Б9119-02.03.01сцт

Петров С.Д.

(Ф.И.О.)

(подпись)

Проверил

Алексеев Г. В.

(Ф.И.О.)

(подпись)

«01» \_\_ февраля 2023г.

г. Владивосток  
2023

## Оглавление

1. Введение .....	3
2. Модель “SIR” .....	3
2.1. Введение.....	3
2.2. Описание процесса .....	3
2.3. Обозначение переменных.....	4
2.4. Построение математической модели .....	4
2.5. Программная реализация .....	5
2.5.1. Код реализации .....	6
2.5.2. Графики.....	7
2.5.3. Анализ графиков .....	8
2.6. Вывод.....	8
3. Модель “WIRiv” .....	8
3.1. Введение.....	8
3.2. Описание процесса .....	9
3.3. Обозначение переменных.....	9
3.4. Построение математической модели .....	10
3.5. Программная реализация .....	10
3.5.1. Код реализации .....	11
3.5.2. Графики.....	14
3.5.3. Анализ графиков .....	14
3.6. Вывод.....	15
4. Заключение .....	15

# 1. Введение

Эпидемии издавна угрожали человечеству, и только в XX веке были разработаны эффективные средства борьбы с инфекциями.

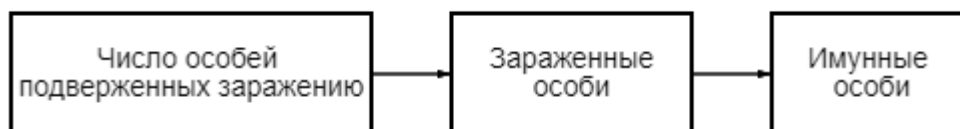
Проблема вирусной заболеваемости и возникновение эпидемий требует оперативного решения и активно исследуется во всем мире, в том числе – и с использованием математического аппарата и компьютеров. Математическое компьютерное моделирование используется для исследования механизмов распространения болезни. Такие модели позволяют прогнозировать и оценивать динамики передачи заболеваний. Это позволяет анализировать и контролировать ситуацию, связанную с распространением вируса, а также предугадывать серьёзные последствия и принимать соответствующие меры по их устранению.

## 2. Модель “SIR”

### 2.1. Введение

Данная модель была разработана в 1927 году и показывает распространение вируса в популяции особей. В модели заболевание передается при контакте здоровой особи с зараженной.

### 2.2. Описание процесса



Пусть в момент времени 0 имеются:

- $S_0$  – число особей, подверженных заражению
- $I_0$  – число зараженных особей

- $R_0$  – число особей, получивших иммунитет

Восприимчивые особи из множества  $S_0$  инфицируются при контакте с особями из множества  $I_0$  со скоростью  $c$ .

Инфицированные особи получают иммунитет со скоростью  $w$ .

## 2.3. Обозначение переменных

Обозначим переменные, необходимые для данной модели:

- $c$  – скорость заражения
- $w$  – скорость иммунизации
- $S(t)$  – число особей, подверженных заражению
- $I(t)$  – число зараженных особей
- $R(t)$  – число особей, получивших иммунитет

## 2.4. Построение математической модели

Для начала определим, как меняется число особей, подверженных заражению. Оно может измениться от контакта с инфицированными.

$$\dot{S} = -cSI$$

**Комментарий:** из этого уравнения видно, что число подверженных заражению особей уменьшается в зависимости от числа инфицированных, не инфицированных и скорости заражения. Таким образом, моделируется заражение особей при контакте.

Скорость изменения количества особей с иммунитетом имеет следующий вид:

$$\dot{R} = wI$$

**Комментарий:** *отсюда видно – число выздоровевших особей меняется в зависимости от числа инфицированных особей и скорости выздоровления.*

Составим уравнение, выражающее скорость изменения количества инфицированных особей:

$$\dot{I} = -\dot{S} - \dot{R}$$

**Комментарий:** *можно заметить, что число инфицированных особей увеличивается на количество только что зараженных особей и одновременно уменьшается на число только что выздоровевших особей.*

## 2.5. Программная реализация

Для реализации будем использовать язык программирования Python

## 2.5.1. Код реализации

```
class SIR:
    def __init__(self, S, I, R, c, w, h):
        self.sum = S + I + R
        self.S, self.I, self.R = S / self.sum, I / self.sum, R / self.sum
        self.c, self.w = c / self.sum, w / self.sum
        self.h = h

    def create_time_line(self, max_timer, step=1):

        S_line = []
        I_line = []
        R_line = []

        S, I, R = self.S, self.I, self.R

        for t in range(int(max_timer / self.h) + 1):

            if t % step == 0:
                S_line += [S]
                I_line += [I]
                R_line += [R]

            S, I, R = (
                S - self.c * self.h * I * S,
                I + self.c * self.h * I * S - self.w * self.h * I,
                R + self.w * self.h * I,
            )

        return (
            np.array(S_line) * self.sum,
            np.array(I_line) * self.sum,
            np.array(R_line) * self.sum,
        )

    def create_plot(self, max_timer, step):
        S_list, I_list, R_list = self.create_time_line(max_timer, step)
        steps = S_list.shape[0]
        x = range(steps)
        plt.fill_between(x, [0] * steps, S_list, label="Подверженные заражению")
        plt.fill_between(x, S_list, I_list + S_list, label="Зараженные")
        plt.fill_between(
            x, S_list + I_list, S_list + I_list + R_list, label="Выздоровевшие"
        )

        plt.xlabel("Часы")
        plt.ylabel("Особи")
        plt.legend()
        plt.show()
```

## 2.5.2. Графики

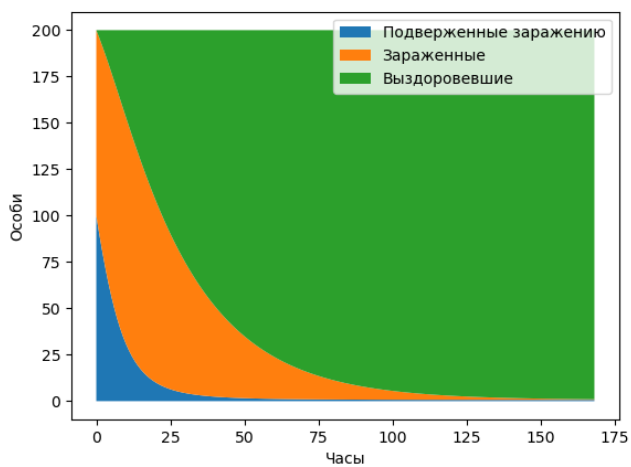


Рис. 1 График модели SIR с  $c = 40$  у. е. и средней  $w = 8$  у. е.

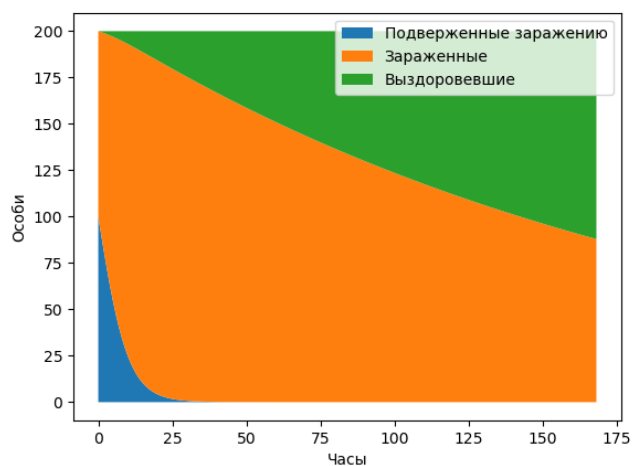


Рис. 2 График модели SIR с  $c = 40$  у. е. и  $w = 2$  у. е.

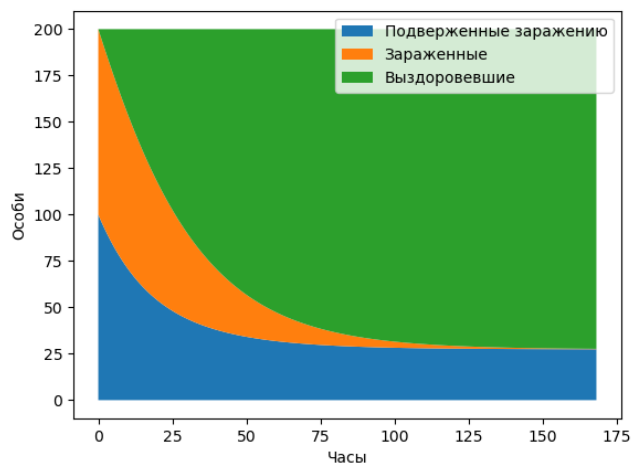


Рис. 3 График модели SIR с  $c = 15$  у. е. и  $w = 10$  у. е.

### **2.5.3. Анализ графиков**

На всех графиках отображен период в неделю с начала моделирования. Количество зараженных и незараженных особей равны 100. Шаг по времени для моделирования – 1 минута.

На Рис. 1 изображен случай, в котором из-за скорости заражения и выздоровления вся эпидемия заканчивается через 5 дней и не остается не переболевших особей.

На Рис. 2 - из-за медленной скорости выздоровления эпидемия продолжается слишком долго и ее завершение не влезает в смоделированный промежуток.

На Рис. 3 – из-за достаточно большой скорости выздоровления эпидемия заканчивается до того момента, когда все особи переболели.

## **2.6. Вывод**

Данная модель учитывает основные отношения между изменениями количества особей в разных группах( $S, I, R$ ). Однако модель не учитывает распределение особей по территории, способ передачи заболевания, а также период, после которого особь не восприимчива к вирусу.

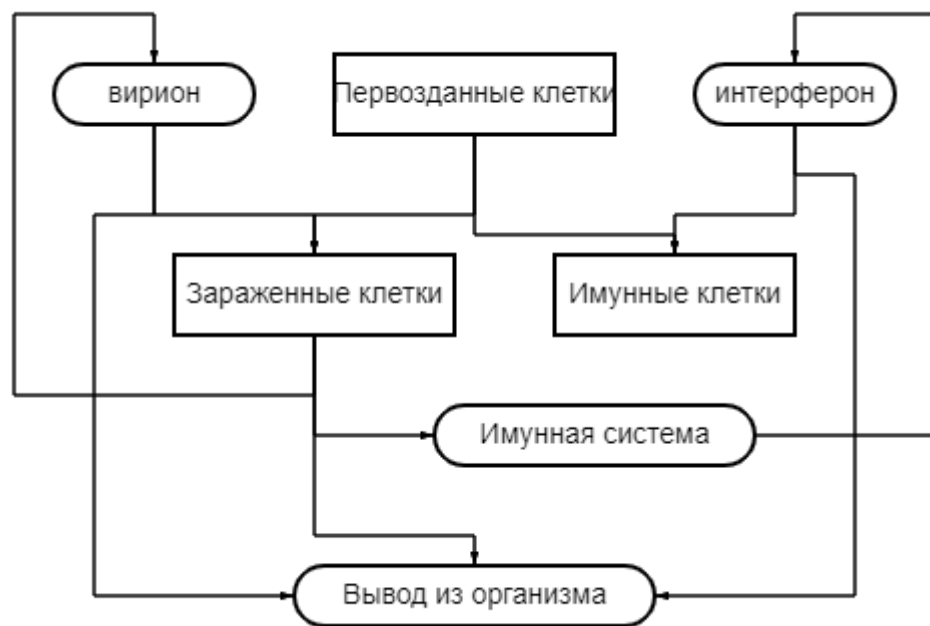
## **3. Модель “WIRiv”**

### **3.1. Введение**

Данная модель описывает процессы, происходящие внутри организма, а именно распространение вируса внутри организма на клеточном уровне.



### 3.2. Описание процесса



Пусть

$$W(0) = W_0; I(0) = I_0; R(0) = R_0; i(0) = i_0; v(0) = v_0$$

Клетки, которые были в организме изначально ( $W$ ), под воздействием вирионов или молекул интерферона могут стать зараженными или иммунными соответственно. Через определенный промежуток времени определенная доля интерферонов, вирионов и инфицированных клеток удаляется. Также инфицированные клетки производят вирионы, и на каждую инфицированную клетку организм производит интерферон.

### 3.3. Обозначение переменных

Обозначим переменные, необходимые для данной модели:

- $W(t)$  – количество первозданных клеток
- $I(t)$  – количество инфицированных клеток
- $R(t)$  – количество клеток, получивших иммунитет
- $i(t)$  – количество молекул интерферона
- $v(t)$  – количество вирионов

- $g_i, g_v$  – количество интерферонов и вирионов на инфицированную клетку
- $\mu_i, \mu_v, \mu_I$  – доля удаляемого интерферона, вириона, инфицированных клеток за единицу времени
- $\alpha_i, \alpha_v$  – доля первозданных клеток, вступивших в контакт с интерфероном или вирионом
- $\beta_i, \beta_v$  – доля интерферонов или вирионов, которые уходят на взаимодействие с первозданными клетками

### 3.4. Построение математической модели

Запишем уравнение изменения числа первозданных клеток:

$$\dot{W} = -\alpha_v v W - \alpha_i i W$$

Уравнение, отображающее скорость изменения зараженных особей:

$$\dot{I} = \alpha_v v W - \mu_I I$$

Уравнение скорости изменения количества резистентных клеток:

$$\dot{R} = \alpha_i i W$$

Уравнение, выражающее скорость изменения числа вирионов:

$$\dot{v} = -\mu_v v + g_v * I - \beta_v v W$$

Уравнение, отображающее скорость изменения числа молекул интерферона:

$$\dot{i} = -\mu_i i + g_i I - \beta_i i W$$

**Комментарий:** заметим, что в последних двух уравнениях производная может иметь как отрицательный, так и положительный знак. Что значит, число вирионов и интерферонов может увеличиваться или уменьшаться.

### 3.5. Программная реализация

Для вычислений необходимо перевести число клеток в проценты от общего числа клеток. Так же необходимо поступить и с числом вирионов и интерферонов. В таком случае произведение параметров в уравнениях не будет «взрываться».

### 3.5.1. Код реализации

```
import numpy as np

import matplotlib.pyplot as plt

class WIRiv:

    def __init__(self, W, I, R, i, v, g, mu, alpha, beta, h, max_interferon=1000):

        self.ceil_sum = W + I + R

        self.W = W / self.ceil_sum

        self.I = I / self.ceil_sum

        self.R = R / self.ceil_sum

        self.molec = max_interferon

        self.i = i / self.molec

        self.v = v / self.molec

        self.g_i = g["i"]

        self.g_v = g["v"]

        self.m_i = mu["i"]

        self.m_I = mu["I"]

        self.m_v = mu["v"]

        self.a_i = alpha["i"]

        self.a_v = alpha["v"]

        self.b_i = beta["i"]

        self.b_v = beta["v"]

        self.h = h

    def create_time_line(self, max_timer, step=1):
```

```

W_line = []
I_line = []
R_line = []

i_line = []
v_line = []

W = self.W
I = self.I
R = self.R

i = self.i
v = self.v

for t in range(int(max_timer / self.h) + 1):

    if t % step == 0:
        W_line += [W]
        I_line += [I]
        R_line += [R]

        i_line += [i]
        v_line += [v]

    W, I, R, i, v = (
        W - self.a_i * i * W - self.a_v * v * W,
        I + self.a_v * v * W - self.m_I * I,
        R + self.a_i * i * W,
        i + self.g_i * I - self.b_i * i * W - self.m_i * i,
        v + self.g_v * I - self.b_v * v * W - self.m_v * v,
    )

```

```

return (
    np.array(W_line) * self.ceil_sum,
    np.array(I_line) * self.ceil_sum,
    np.array(R_line) * self.ceil_sum,
    np.array(i_line) * self.molec,
    np.array(v_line) * self.molec,
)

def create_plot(self, max_timer, step):
    W_line, I_line, R_line, i_line, v_line = self.create_time_line(max_timer, step)

    steps = W_line.shape[0]
    x = range(steps)

    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))

    ax1.fill_between(x, [0] * steps, W_line, label="Подверженные заражению")
    ax1.fill_between(x, W_line, I_line + W_line, label="Зараженные")
    ax1.fill_between(
        x, W_line + I_line, W_line + I_line + R_line, label="Выздоровевшие"
    )

    ax1.legend()

    ax2.plot(x, i_line, label="Интерферон")
    ax2.plot(x, v_line, label="Вирион")

    ax2.legend()
    plt.show()

```

### 3.5.2. Графики

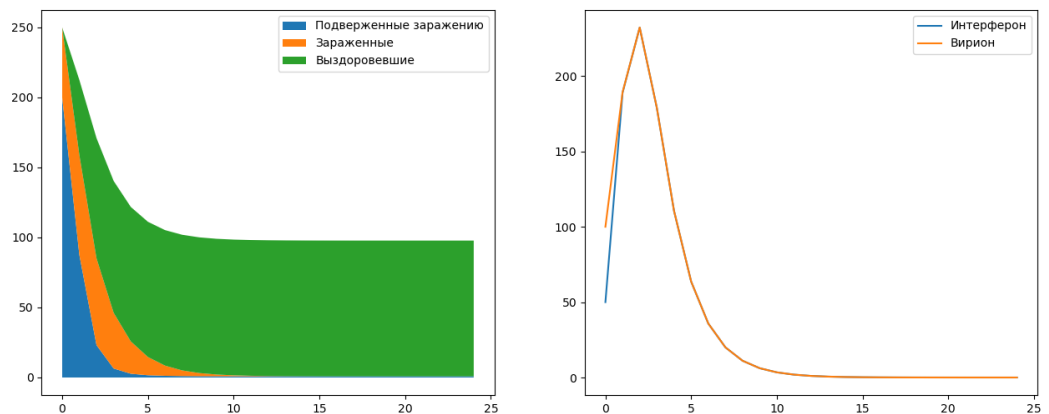


Рис. 4

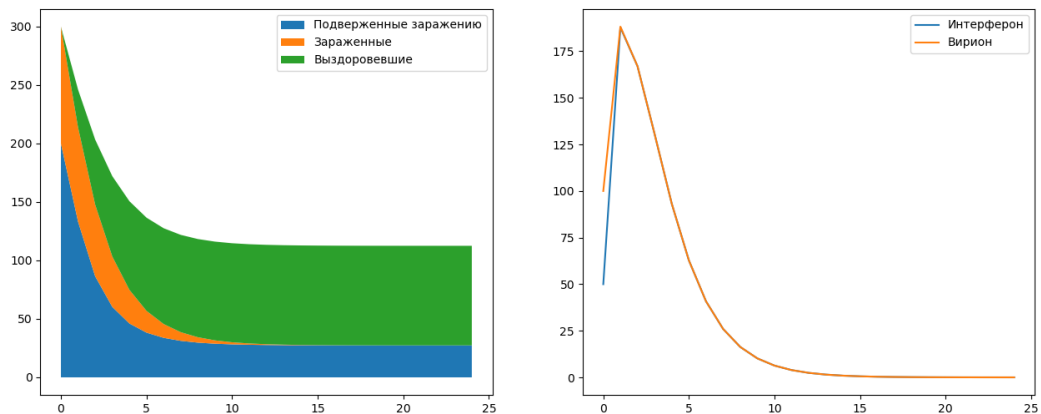


Рис. 5

### 3.5.3. Анализ графиков

На графиках отображен промежуток в 24 часа с начала моделирования.

На Рис. 4 параметры подобраны таким образом, что за первые 6 часов все клетки заражаются и далее медленно вымирают. Удачно, что при увеличении числа зараженных клеток увеличивается число интерферона, а следовательно, клетки получают иммунитет гораздо быстрее.

На Рис. 5 зараженные клетки слишком быстро вымирают, вследствие чего не все первозданные клетки получают иммунитет.

### **3.6. Вывод**

Данная модель учитывает наличие интерферона и вирионов, а также скорости их выведения.

## **4. Заключение**

В данной работе были рассмотрены математические модели по распространению вирусов и инфекций, описанные при помощи ОДУ. Были созданы реализации, помогающие познакомиться с результатами работы данных моделей.