Лабораторная работа №6

Задача об эпидемии

Латыпова Диана. НФИбд-02-21

Содержание

1 Цель работы 5

2 Задание 6

3 Теоретическое введение 7

4 Выполнение лабораторной работы 9

5 Выводы 17

Список литературы 18

Список иллюстраций

4.1 Julia. Случай1 12
4.2 OM. Случай1 13
4.3 Julia. Случай2 15
4.4 OM. Случай2 16

Список таблиц

1 Цель работы

- Построить графики изменения числа особей в каждой из трех групп.
- Рассмотреть два случая, как будет протекать эпидемия.

2 Задание

Вариант 46.

На одном острове вспыхнула эпидемия. Известно, что из всех проживающих на острове $(N = 6730)$ в момент начала эпидемии $(t = 0)$ число заболевших людей (являющихся распространителями инфекции) $I(0) = 46$, А число здоровых людей с иммунитетом к болезни $R(0) = 8$. Таким образом, число людей восприимчивых к болезни, но пока здоровых, в начальный момент времени $S(0) = N - I(0) - R(0)$. Постройте графики изменения числа особей в каждой из трех групп.

Рассмотрите, как будет протекать эпидемия в случае:

1. $I(0) \leq I^*$
2. $I(0) > I^*$

3 Теоретическое введение

Модель SIR является одной из базовых моделей в эпидемиологии и описывает динамику распространения инфекционных заболеваний. Она состоит из трех основных дифференциальных уравнений [1]:

Уравнение подверженности к инфекции (Susceptible):

$$\frac{dS}{dt} = \beta SI$$

где β - коэффициент заражения, который описывает вероятность передачи инфекции от инфицированного человека к восприимчивому.

Уравнение инфицирования (Infected):

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

где γ - коэффициент выздоровления, который описывает скорость выздоровления или убытия заболевших (выздоровевших или умерших).

Уравнение выздоровления (Recovered)

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

Параметр I^*. Параметр I^*представляет собой критическое значение заболевших, при котором эпидемия переходит в экспоненциальный рост или наоборот, затухает. Если $I(0) \leq I^*$, то начальные условия не приводят к экспоненциальному росту, и эпидемия подавляется. В противном случае, если $I(0) > I^*$, эпидемия может продолжиться и привести к большему числу заболевших.

Графики [2] изменения числа особей в каждой из трех групп: Построение графиков происходит путем решения системы дифференциальных уравнений SIR для различных значений времени t.

4 Выполнение лабораторной работы

Основные понятия:

$S(t)$ - количество восприимчивых к болезни, но пока здоровых людей в момент времени t.

$I(t)$ - количество заболевших (инфицированных) людей в момент времени t.

$R(t)$ - количество выздоровевших (реабилитированных) людей в момент времени t.

N - общее количество людей на острове.

Случай1: $I(0) \leq I^*$

Код на языке Julia (рис. 4.1):

```
# Подключаем необходимые библиотеки
using Plots
using DifferentialEquations

# Указываем начальные данные для моделирования эпидемии
N = 6730     # Общее количество особей на острове
I0 = 46      # Начальное количество заболевших особей
R0 = 8       # Начальное количество особей с иммунитетом
S0 = N - I0 - R0  # Начальное количество здоровых, но восприимчивых особей
alpha = 0.6  # Коэффициент заболеваемости
beta = 0.2   # Коэффициент выздоровления
```

```
# Определяем функцию, описывающую систему дифференциальных уравнений
function ode_fn(du, u, p, t)
    S, I, R = u
    # Дифференциальные уравнения модели SIR
   du[1] = 0        # Изменение количества восприимчивых особей (не изменяется)
   du[2] = -beta * u[2]      # Изменение количества инфицированных особей
   du[3] = beta * u[2]       # Изменение количества особей с иммунитетом
end

# Начальные условия и временной интервал моделирования
v0 = [S0, I0, R0]
tspan = (0.0, 60.0)

# Создаем объект, представляющий задачу дифференциальных уравнений (ODEProblem)
prob = ODEProblem(ode_fn, v0, tspan)

# Решаем дифференциальные уравнения
sol = solve(prob, dtmax = 0.05)

# Получаем решение
S = [u[1] for u in sol.u]
I = [u[2] for u in sol.u]
R = [u[3] for u in sol.u]
T = [t for t in sol.t]

# Строим графики
plt = plot(
    dpi = 600,              # Разрешение графика
    legend = :topright      # Позиция легенды
```

```
)

# График количества восприимчивых особей
plot!(
    plt,
    T,
    S,
    label = "Восприимчивые особи",  # Подпись для легенды
    color = :purple                  # Цвет графика
)

# График количества инфицированных особей
plot!(
    plt,
    T,
    I,
    label = "Инфицированные особи", # Подпись для легенды
    color = :red                     # Цвет графика
)

# График количества особей с иммунитетом
plot!(
    plt,
    T,
    R,
    label = "Особи с иммунитетом",   # Подпись для легенды
    color = :green                   # Цвет графика
)
```

```
# Сохраняем график в файл
savefig(plt, "jullab6_1.png")
```

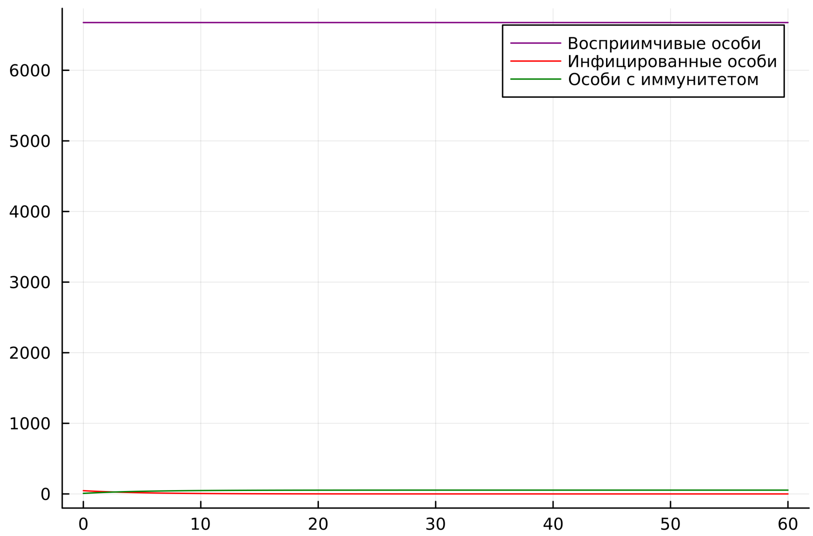

Рис. 4.1: Julia. Случай1

Код на ПО OpenModelica (рис. 4.2):

```
model lab6_1
Real S;
Real I;
Real R;
Real N = 6730;
Real alpha = 0.6;
Real beta = 0.2;
initial equation
I = 46;
R = 8;
S = N - I - R;
equation
der(S) = 0;
```

```
der(I) = -beta*I;
der(R) = beta*I;
end lab6_1;
```

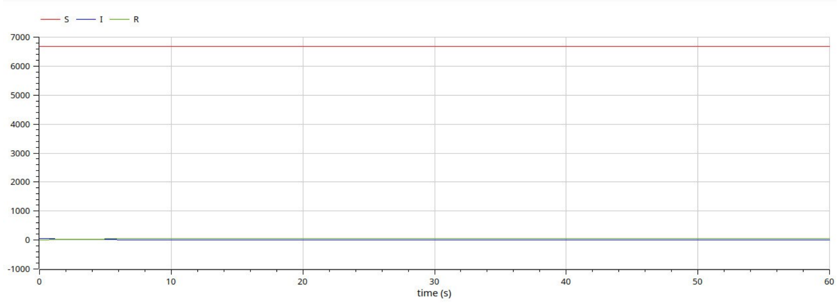

Рис. 4.2: OM. Случай1

Случай2: $I(0) > I^*$

Код на языке Julia (рис. 4.3):

```
using Plots
using DifferentialEquations

N = 6730
I0 = 46 # заболевшие особи
R0 = 8 # особи с иммунитетом
S0 = N - I0 - R0 # здоровые, но восприимчивые особи
alpha = 0.5 # коэффициент заболеваемости
beta = 0.1 # коэффициент выздоровления

function ode_fn(du, u, p, t)
    S, I, R = u
    du[1] = -alpha*u[1]
    du[2] = alpha*u[1] - beta*u[2]
    du[3] = beta*I
```

```
end

v0 = [S0, I0, R0]
tspan = (0.0, 60.0)
prob = ODEProblem(ode_fn, v0, tspan)
sol = solve(prob, dtmax = 0.05)

S = [u[1] for u in sol.u]
I = [u[2] for u in sol.u]
R = [u[3] for u in sol.u]
T = [t for t in sol.t]

plt = plot(
  dpi = 600,
  legend = :topright)
plot!(
  plt,
  T,
  S,
  label = "Восприимчивые особи",
  color = :purple)
plot!(
  plt,
  T,
  I,
  label = "Инфицированные особи",
  color = :red)
plot!(
  plt,
```

```
  T,
  R,
  label = "Особи с иммунитетом",
  color = :green)

savefig(plt, "jullab6_2.png")
```

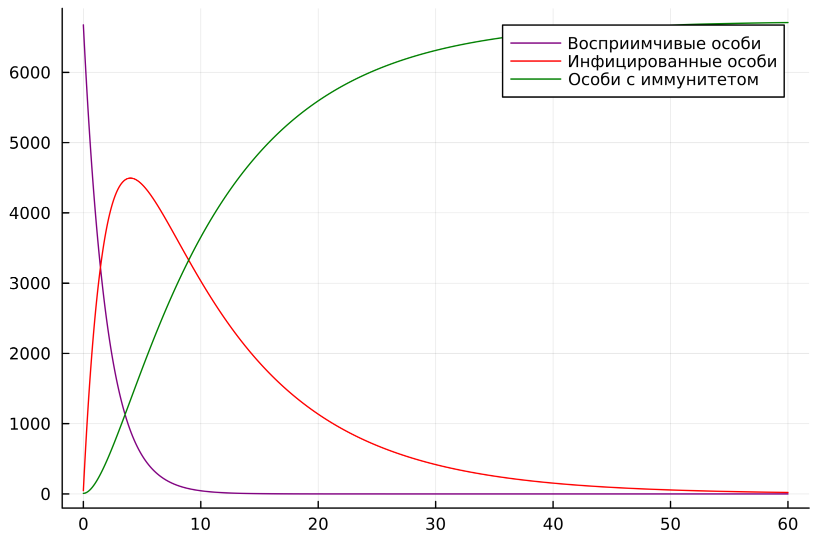

Рис. 4.3: Julia. Случай2

Код на ПО OpenModelica (рис. 4.4):

```
model lab6_2
Real S;
Real I;
Real R;
Real N = 6730;
Real alpha = 0.4;
Real beta = 0.1;
initial equation
I = 46;
```

```
R = 8;
S = N - I - R;
equation
der(S) = -alpha*S;
der(I) = alpha*S - beta*I;
der(R) = beta*I;
end lab6_2;
```

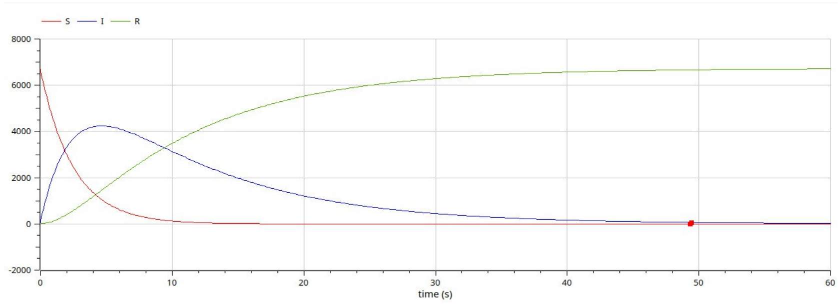

Рис. 4.4: ОМ. Случай2

Анализ. Сравнивая смоделированную задачу на языке программирования Julia и на ПО OpenModelica, можем заметить, что на ПО ОМ коды гораздо меньше и легче в плане их написания, при том, что в конечном итоге имеем абсолютно одинаковые графики.

5 Выводы

Я построила графики изменения числа особей в каждой из трех групп, рассмотрела два случая, как будет протекать эпидемия. Смоделировала задачу об эпидемии на языке программирования Julia и на ПО OpenModelica.

Список литературы

1. Моделирование распространения вирусной инфекции [Электронный ресурс]. САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, 2021. URL: https://dspace.spbu.ru/bitstream/11701/32530/1/VKR_Modelirovanie_rasprostranenia_virusnoj_infekcii__Uzakova_A.S.pdf.
2. Решение дифференциальных уравнений [Электронный ресурс]. Кафедра Технологии воды и топлива НИУ МЭИ, 2012. URL: http://twt.mpei.ac.ru/ochkov/mathcad_14/Chapter6rus/index.html.