

Лабораторная работа № 6

Задача об эпидемии

Пиняева Анна Андреевна

Содержание

Цель работы

3

Теоретическое введение

3

Задание

4

Выполнение лабораторной работы

5

Построение математической модели. Решение с помощью программ

5

Julia

5

Результаты работы кода на Julia

6

Julia

7

Результаты работы кода на Julia

8

OpenModelica

9

Результаты работы кода на OpenModelica

10

Выводы

12

Список литературы

12

Цель работы

Целью данной работы является построение модели эпидемии.

Теоретическое введение

Рассмотрим простейшую модель эпидемии SIR [1]. 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, обозначаемая 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 $I(t) > I^*$, тогда инфицирование способны заражать восприимчивых к болезни особей. [2]

Задание

Вариант 29

На одном острове вспыхнула эпидемия. Известно, что из всех проживающих на острове ($N=11\ 600$) в момент начала эпидемии ($t=0$) число заболевших людей (являющихся распространителями инфекции) $I(0)=260$, А число здоровых людей с иммунитетом к болезни $R(0)=48$. 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) если $I(0) \leq I^*$
- 2) если $I(0) > I^*$

“Вариант 29”

Выполнение лабораторной работы

Построение математической модели. Решение с помощью программ

Julia

Первый случай:

```
using Plots
using DifferentialEquations

N = 11600
I0 = 260
R0 = 48
S0 = N - I0 - R0
a = 0.01
b = 0.02

function ode_fn(du, u, p, t)
    S, I, R = u;
    du[1] = 0
    du[2] = -b*u[2]
    du[3] = b*u[2]
end

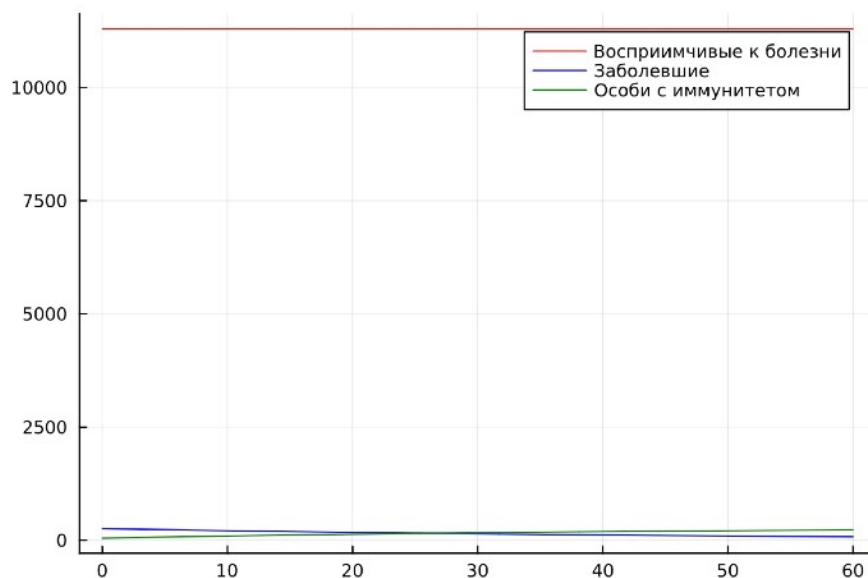
v0 = [S0, I0, R0]
tspan = (0.0, 60.0)
prob = ODEProblem(ode_fn, v0, tspan)
sol = solve(prob, dtmax = 0.05)
S = [u[1] for u in sol.u]
I = [u[2] for u in sol.u]
R = [u[3] for u in sol.u]
T = [t for t in sol.t]
```

```
plt = plot(  
    dpi = 300,  
    legend = :topright)  
  
plot!(  
    plt,  
    T,  
    S,  
    label = "Восприимчивые к болезни",  
    color = :red)  
  
plot!(  
    plt,  
    T,  
    I,  
    label = "Заболевшие",  
    color = :blue)  
  
plot!(  
    plt,  
    T,  
    R,  
    label = "Особь с иммунитетом",  
    color = :green)
```

Результаты работы кода на Julia

Построим графики численности особей трех групп S, I, R для первого случая (рис.1)

Out [2]:



“Рис.1 Графики численности особей трех групп S , I , R , построенные на Julia, для случая, когда больные изолированы”

Julia

Второй случай:

```
using Plots
using DifferentialEquations
```

```
N = 11600
I0 = 260
R0 = 48
S0 = N - I0 - R0
a = 0.01
b = 0.02
```

```
function ode_fn(du, u, p, t)
    S, I, R = u;
    du[1] = -a*u[1]
    du[2] = a*u[1]-b*u[2]
    du[3] = b*u[2]
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 = 300,
    legend = :topright)

plot!(
    plt,
    T,
    S,
    label = "Восприимчивые к болезни",
    color = :red)

plot!(
    plt,
    T,
    I,
    label = "Заболевшие",
    color = :blue)

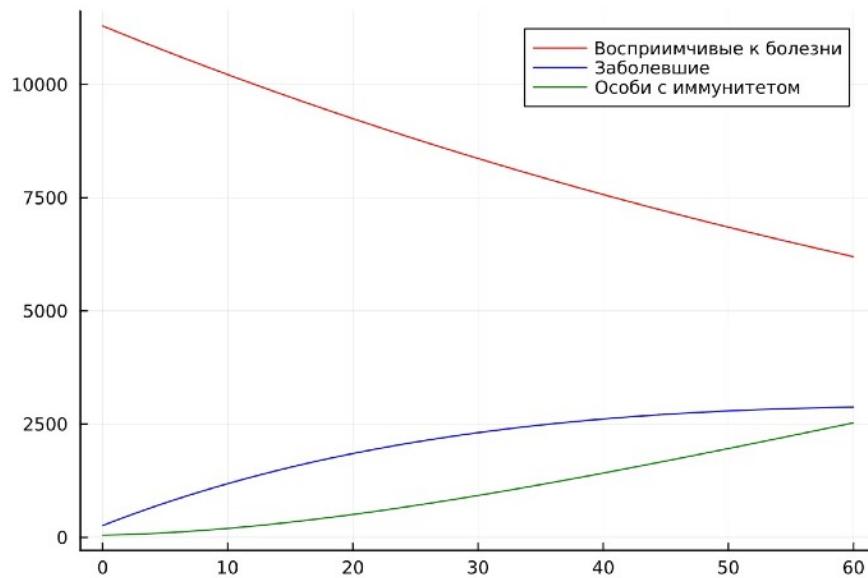
plot!(
    plt,
    T,
    R,
    label = "Особь с иммунитетом",
    color = :green)

```

Результаты работы кода на Julia

По аналогии с предыдущим построением получим графики для второго случая (рис.2)

Out[1]:



“Рис.2 Графики численности особей трех групп S , I , R , построенные на Julia, для случая, когда больные могут заражать особей группы S ”

OpenModelica

Первый случай:

```
model lab6_1
Real N = 11600;
Real I;
Real R;
Real S;
Real a = 0.01;
Real b = 0.02;
initial equation
I = 260;
R = 48;
S = N - I - R;
equation
der(S) = 0;
```

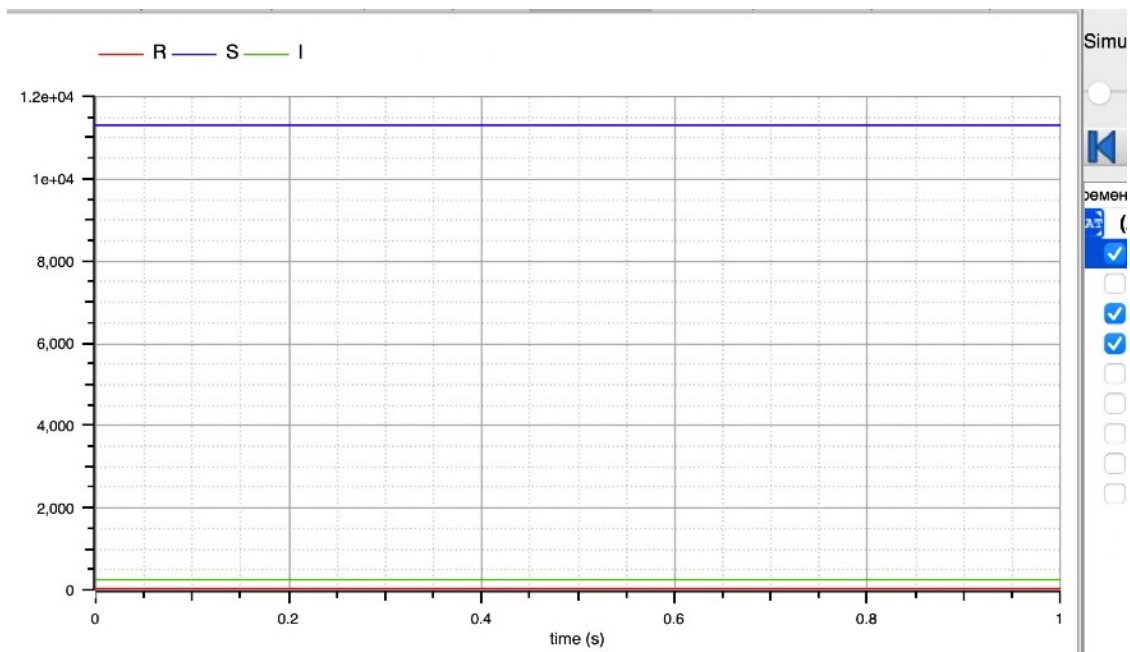
```
der(I) = -b*I;  
der(R) = b*I;  
end lab6_1;
```

Второй случай:

```
model lab6_2  
Real N = 11600;  
Real I;  
Real R;  
Real S;  
Real a = 0.01;  
Real b = 0.02;  
initial equation  
I = 260;  
R = 48;  
S = N - I - R;  
equation  
der(S) = -a*S;  
der(I) = a*S-b*I;  
der(R) = b*I;  
end lab6_2;
```

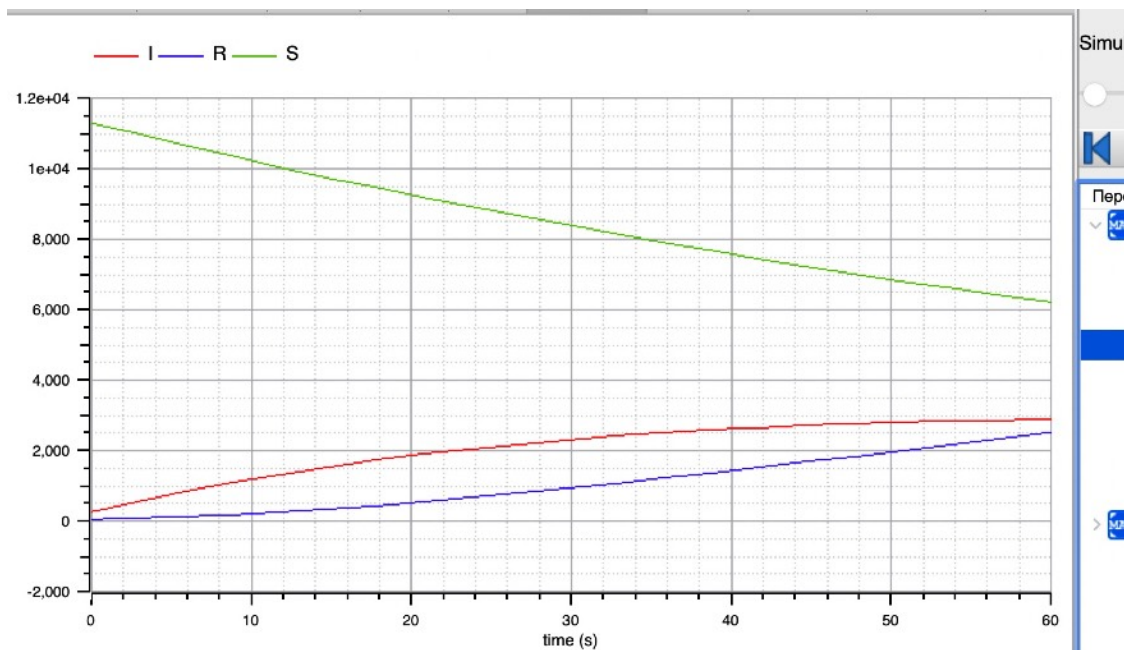
[Результаты работы кода на OpenModelica](#)

Построим графики численности особей трех групп S, I, R для первого случая (рис.3)



“Рис.3 Графики численности особей трех групп S, I, R, построенные на OpenModelica, для случая, когда больные изолированы”

По аналогии с предыдущим построением получим получим графики для второго случая (рис.4)



“Рис.4 Графики численности особей трех групп S, I, R, построенные на OpenModelica, для случая, когда больные могут заражать особей группы S”

Выводы

В итоге проделанной работы мы построили графики зависимости численности особей трех групп S, I, R для случаев, когда больные изолированы и когда они могут заражать особей группы S, на языках Julia и OpenModelica. Построение модели эпидемии на языке OpenModelica занимает значительно меньше строк, чем аналогичное построение на Julia. Кроме того, построения на языке OpenModelica проводятся относительно значения времени t по умолчанию, что упрощает нашу работу.

Список литературы

- [1] Конструирование эпидемиологических моделей. Habr: <https://habr.com/ru/post/551682/>
- [2] Руководство к лабоартпорной работе: https://esystem.rudn.ru/pluginfile.php/1971664/mod_resource/content/2/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80

%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B
0%20%E2%84%96%205.pdf