

Отчёт по лабораторной работе №3

Боровикова Карина Владимировна

2023, 11 февраля

RUDN University, Moscow, Russian Federation

- Важность умения работы с языками Julia и OpenModelica в части математического моделирования

- Язык Julia
- Язык OpenModelica
- Моделирование задачи об эпидемии

Цель работы:

Целью данной лабораторной работы является создание модели эпидемии с помощью языков Julia и OpenModelica. Построить соответствующие графики двух случаев.

Задачи:

- Рассмотреть процесс распространения эпидемии в двух случаях:
 $I(0) \leq I^*$ и $I(0) > I^*$
- Построить графики изменения количества особей в каждой из трех категорий особей: $I(t)$ - инфицированные особи, $S(t)$ - восприимчивые к болезни здоровые особи, $R(t)$ - здоровые особи с иммунитетом к болезни.

1. Рассматриваем задачу об эпидемии (рис. 1).

Вариант 18

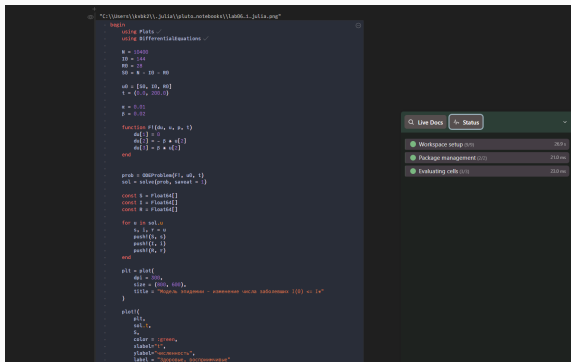
На одном острове вспыхнула эпидемия. Известно, что из всех проживающих на острове ($N=10\ 400$) в момент начала эпидемии ($t=0$) число заболевших людей (являющихся распространителями инфекции) $I(0)=144$, А число здоровых людей с иммунитетом к болезни $R(0)=28$. Таким образом, число людей восприимчивых 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 $I(0) > I^*$

Рис. 1: Моделирование эпидемии

2. Программируем первый случай на Julia (рис. 2).



```
"C:\Users\3342\julia\julia_notebooks\lab06.1_julia.png"
begin
  using Plots
  using DifferentialEquations

  N = 10000
  T0 = 14
  Tm = 25
  T0 = T - T0 - Tm

  u0 = [50, 10, 60]
  t = {0.0, 200.0}

  α = 0.02
  β = 0.02

  function v!(du, u, p, t)
    du[1] = 0
    du[2] = -β * u[2]
    du[3] = β * u[2]
  end

  prob = ODEProblem{t, u0, 1}
  sol = solve(prob, saveat = 1)

  const S = Plots[]
  const I = Plots[]
  const R = Plots[]

  for i in 1:N
    t, u, v = u
    push!(S, i)
    push!(I, i)
    push!(R, i)
  end

  plt = plot(
    sol = sol,
    size = (800, 600),
    title = "Epidemic dynamics - population size dynamics (I(t) vs t)"
  )

  plot!(
    plt,
    sol,
    t,
    color = :green,
    label = "I",
    label = "инфекция",
    label = "Infectious, not recovered"
  )
```

Q. Live Docs Status

- Workspace setup (0/1) 26.0 s
- Package management (0/1) 210 ms
- Evaluating cells (0/1) 220 ms

Рис. 2: Первый случай на Julia

3. Програмируем первый случай на OpenModelica (рис. 3).

```
1  model lab061
2      constant Integer N = 10400;
3      constant Integer I0 = 144;
4      constant Integer R0 = 28;
5      constant Integer S0 = N - I0 - R0;
6      constant Real alpha = 0.01;
7      constant Real beta = 0.02;
8      Real s(start=S0);
9      Real i(start=I0);
10     Real r(start=R0);
11     Real t = time;
12     equation
13         der(s) = 0;
14         der(i) = -beta*i;
15         der(r) = beta*i;
16         annotation(experiment(StartTime = 0, StopTime = 200.0), Documentation);
17     end lab061;
18
```

Рис. 3: Первый случай на OpenModelica

4. Програмуємо другий випадок на Julia (рис. 4).

```

"1: (yourname, your_email, jobtitle, websiteurl, linkedin_id, github_id)"
)end

using Plots
using DifferentialEquations

N = 10000
dt = 100
M0 = 0
M1 = N - M0 - m0

u0 = [M0, M1, m0]
t = [0.0, 100.0]

a = 0.01
b = 0.02

function f(t, y, p, N)
    M, I, A = y
    dM1 = -b * M * I
    dM2 = -b * M * I
    dM3 = -b * M * I
    dM = 0
end

prob = ODEProblem(f, u0, t, p)
sol = solve(prob, saveat = 1)

times = 1:length(sol)
times1 = times[1]
times2 = times[2]
times3 = times[3]

for i in 1:3
    plot(sol, t, y[i],
         label = "M",
         title = "M",
         legend = false,
         xlim = (0, 100),
         ylim = (0, 100),
         title = "Search progress - estimated virus infections (10^4)"
    )
end

plot(
    plot1,
    plot2,
    plot3,
    color = :green,
    legend = false,
    xlabel = "time (days)",
    ylabel = "infections",
    title = "Search progress - estimated virus infections"
)

```

Рис. 4: Второй случай на Julia

5. Программируем второй случай на OpenModelica (рис. 5).

```
model lab062
  constant Integer N = 10400;
  constant Integer I0 = 144;
  constant Integer R0 = 28;
  constant Integer S0 = N - I0 - R0;
  constant Real alpha = 0.01;
  constant Real beta = 0.02;
  Real s(start=S0);
  Real i(start=I0);
  Real r(start=R0);
  Real t = time;
equation
  der(s) = -alpha*s;
  der(i) = alpha*s-beta*i;
  der(r) = beta*i;
  annotation(experiment(StartTime = 0, StopTime = 200.0), Documentation);
end lab062;
```

Рис. 5: Второй случай на OpenModelica

6. Запускаем код через терминал, получаем изображения для первого и второго случаев на Julia (рис. 6).

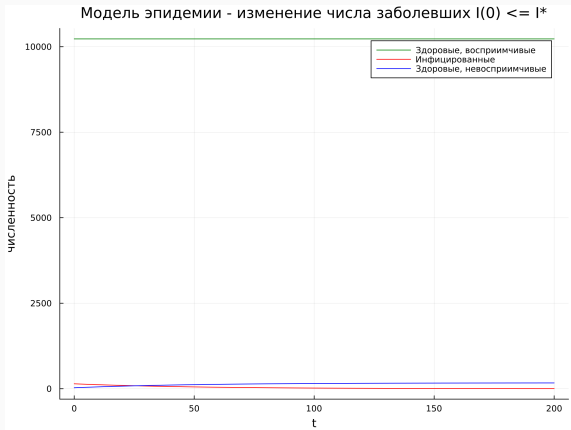


Рис. 6: Результаты выполнения кодов на Julia

7. Запускаем код через терминал, получаем изображения для первого и второго случаев на Julia (рис. 7).

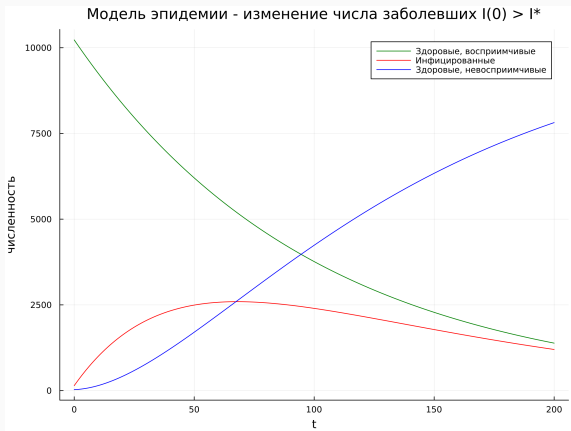


Рис. 7: Результаты выполнения кодов на Julia

8. Запускаем код через терминал, получаем изображения для первого и второго случаев на OpenModelica (рис. 8).

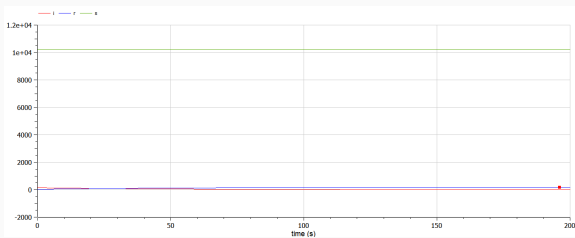


Рис. 8: Результаты выполнения кодов на Julia

9. Запускаем код через терминал, получаем изображения для первого и второго случаев на OpenModelica (рис. 9).

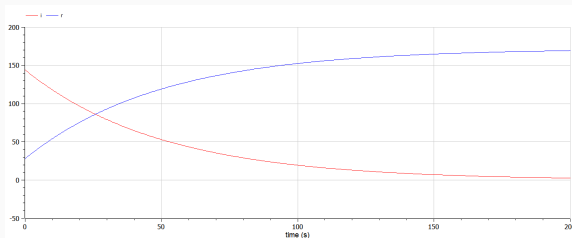
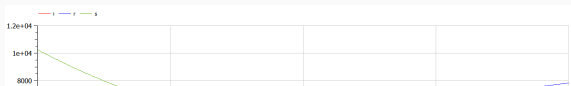


Рис. 9: Результаты выполнения кодов на Julia

10. Запускаем код через терминал, получаем изображения для первого и второго случаев на OpenModelica (рис. 9).



Результаты

В ходе выполнения лабораторной работы я построила модель задачи об эпидемии с помощью языков Julia и OpenModelica, построила графики изменения количества особей трех категорий - $S(t)$ - восприимчивые к болезни, но здоровые особи, $I(t)$ - инфицированные особи $R(t)$ - здоровые особи с иммунитетом к болезни в двух различных случаях.