

Лабораторная работа №6

Задача об эпидемии

Шубнякова Дарья НКНбд-01-22

Содержание

1	Цель работы	3
2	Задание	3
3	Теоретическое введение	3
4	Выполнение лабораторной работы	3
5	Выводы	7

1 Цель работы

Ознакомиться с задачей об эпидемии. Реализовать данную модель на языке Julia, а так же в среде OMEdit.

2 Задание

Построить два графика с динамикой эпидемии и относительными долями населения.

3 Теоретическое введение

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, обозначаемая через $R(t)$ – это здоровые особи с иммунитетом к болезни.

4 Выполнение лабораторной работы

Пишем код на языке Julia для реализации данной модели(рис. 1).

```
[2]: using DifferentialEquations, Plots

# Параметры модели для варианта 13
N = 19000.0 # Общая численность населения
beta = 0.3 # Коэффициент заражения
gamma = 0.1 # Коэффициент выздоровления

# Начальные условия
S0 = N - 119 - 19 # Восприимчивые: 19000 - 119 - 19 = 18862
I0 = 119.0 # Инфицированные
R0 = 19.0 # Выздоровевшие с иммунитетом

u0 = [S0, I0, R0] # Начальные условия [S, I, R]
tspan = (0.0, 200.0) # Временной интервал

# Система уравнений SIR модели
function sir_model!(du, u, p, t)
    S, I, R = u
    beta, gamma, N = p

    du[1] = -beta * S * I / N # dS/dt
    du[2] = beta * S * I / N - gamma * I # dI/dt
    du[3] = gamma * I # dR/dt
end

# Параметры модели
p = (beta, gamma, N)

# Решение системы
prob = ODEProblem(sir_model!, u0, tspan, p)
sol = solve(prob, Tsit5(), reltol=1e-6, abstol=1e-6)

# Построение графиков
plt = plot(layout=(2,1), size=(800, 600), dpi=300)
```

Рисунок 1

Продолжение кода(рис. 2).

```

# График 1: Изменение численности групп
plot!(plt[1], sol.t, sol[1,:], label="Восприимчивые (S)", linewidth=2, color=:blue)
plot!(plt[1], sol.t, sol[2,:], label="Инфицированные (I)", linewidth=2, color=:red)
plot!(plt[1], sol.t, sol[3,:], label="Выздоровевшие (R)", linewidth=2, color=:green)
plot!(plt[1], title="Динамика эпидемии (SIR модель)", xlabel="Время", ylabel="Численно

# График 2: Относительные доли
plot!(plt[2], sol.t, sol[1,:]./N, label="S/N", linewidth=2, color=:blue)
plot!(plt[2], sol.t, sol[2,:]./N, label="I/N", linewidth=2, color=:red)
plot!(plt[2], sol.t, sol[3,:]./N, label="R/N", linewidth=2, color=:green)
plot!(plt[2], title="Относительные доли населения", xlabel="Время", ylabel="Доля", leg

# Анализ порогового значения
R0_effective = beta / gamma # Базовое репродуктивное число
I_critical = N * (gamma / beta) # Критическое число инфицированных

println("Анализ эпидемии:")
println("Общая численность: N = ", N)
println("Начальные условия: S₀ = ", S0, ", I₀ = ", I0, ", R₀ = ", R0)
println("Базовое репродуктивное число: R₀ = ", round(R0_effective, digits=2))
println("Критическое число инфицированных: I* = ", round(I_critical, digits=2))

# Проверка условий
if I0 <= I_critical
    println("Случай 1: I₀ ≤ I* – эпидемия будет слабой")
else
    println("Случай 2: I₀ > I* – ожидается сильная эпидемия")
end

# Отображение графиков
display(plt)

# Сохранение результатов
savefig("epidemic_dynamics.png")
println("\nГрафик сохранен в файл: epidemic_dynamics.png")

```

Рисунок 2

Получаем два графика, необходимых для выполнения задания(рис. 3).

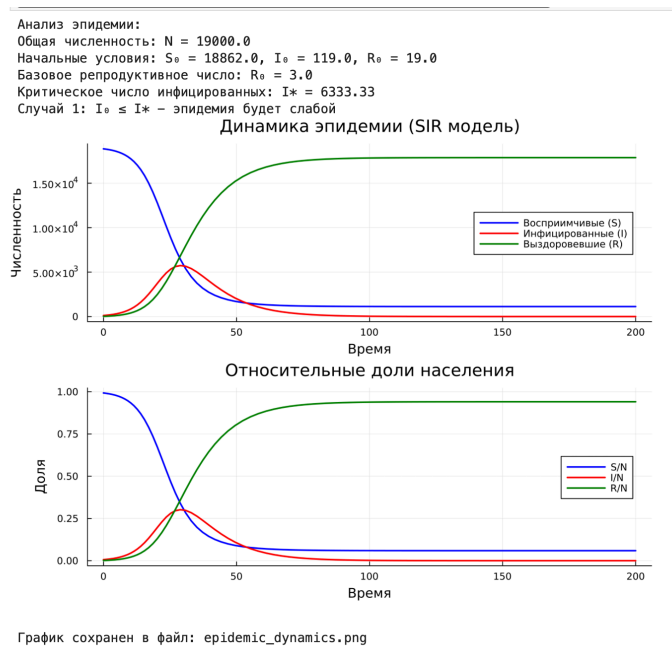


Рисунок 3

Прописываем код в среде OMEdit(рис. 4).

```
model EpidemicModel "Модель эпидемии SIR для варианта 13"
  parameter Real N = 19000 "Общая численность населения";
  parameter Real beta = 0.3 "Коэффициент заражения";
  parameter Real gamma = 0.1 "Коэффициент выздоровления";

  parameter Real S0 = N - 119 - 19 "Начальное число восприимчивых";
  parameter Real I0 = 119 "Начальное число инфицированных";
  parameter Real R0 = 19 "Начальное число выздоровевших";

  Real S(start = S0, fixed = true) "Восприимчивые";
  Real I(start = I0, fixed = true) "Инфицированные";
  Real R(start = R0, fixed = true) "Выздоровевшие";
  Real t "Время";
```

Рисунок 4

Продолжение кода(рис. 5).

```
15 equation
16 // Система уравнений SIR модели
17 der(S) = -beta * S * I / N;
18 der(I) = beta * S * I / N - gamma * I;
19 der(R) = gamma * I;
20
21 der(t) = 1;
22
23 annotation(experiment(StartTime=0, StopTime=200, Tolerance=1e-6, Interval=0.1));
24 end EpidemicModel;
```

Рисунок 5

Получаем такой график(рис. 6).

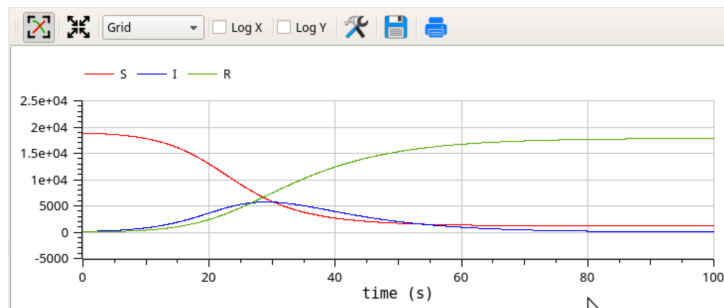


Рисунок 6

5 Выводы

Мы реализовали модель в OpenModelica и на языке Julia. На выходе получили картинку: epidemic_dynamics.png. Итоговый файл lab6.ibybn с кодом на языке Julia в JupiterNotebook. А также файл для симуляции в OpenModelica: EpidemicModel.mo