Отчет по лабораторной работе №6

по дисциплине: Математическое моделирование

Майсаров А.М.

Содержание

Список иллюстраций

1 Цель работы

Продолжить знакомство с функционалом языка программирования Julia, дополнительных библиотек (DifferentialEquations, Plots). Продолжить ознакомление с языком моделирования Modelica и программным обеспечением OpenModelica. Используя эти средства, описать задачу об эпидемии (используя измененную математическую модель SIR).

2 Задание

На одном острове вспыхнула эпидемия. Известно, что из всех проживающих на острове ($N = 25000$) в момент начала эпидемии ($t = 0$) число заболевших людей (являющихся распространителями инфекции) $I(0) = 150$, А число здоровых людей с иммунитетом к болезни $R(0) = 15$. 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 $I(0) > I^*$

3 Теоретическое введение

Задача текущей лабораторной работы сводится к построению математической модели, достаточно сильно похожей на модель SIR. Сначала будет дан материал о модели «Susceptible-Infectious-Recovered», а далее будут рассмотрены различия данной модели и модели, используемой при выполнении лабораторной работы.

3.1 Модель SIR

Модель SIR - это математическая модель, используемая для описания распространения инфекционных заболеваний в популяции. Аббревиатура SIR означает «Susceptible-Infectious-Recovered». Из расшифровки аббревиатуры следует, что модель разделяет популяцию на три группы: восприимчивые (susceptible), инфицированные (infectious) и выздоровевшие (recovered).

В модели SIR инфекционное заболевание передается от инфицированных к восприимчивым через непосредственный контакт. Когда восприимчивый контактирует с инфицированным, есть определенная вероятность заражения, которая зависит от свойств возбудителя и сопротивляемости организма. После того, как восприимчивый заразился, он становится инфицированным, и тем самым переходит в группу infectious.

Когда инфицированный выздоравливает, он переходит в группу recovered. В отличие от других моделей, таких как SEIR, модель SIR не учитывает длительности инкубационного периода или время восстановления, и считает, что инфицированные остаются в одном состоянии до тех пор, пока не выздоровеют [**SIR?**].

Модель SIR представляется системой трех дифференциальных уравнений, которые описывают динамику численности каждой группы в зависимости от времени. Эти уравнения могут быть использованы для прогнозирования темпов распространения заболевания и оценки эффективности мер по его контролю.

1. Уравнение числа восприимчивых (S):

   $$\frac{dS}{dt} = -\frac{\beta I S}{N},$$

   где β — коэффициент интенсивности контактов индивидов с последующим инфицированием; $S(t)$ — численность восприимчивых индивидов в момент времени t; $I(t)$ — численность инфицированных индивидов в момент времени t; N — объем популяции.

   Первое уравнение описывает изменение численности восприимчивых с течением времени. Уравнение показывает, что изменение числа здоровых (и при этом восприимчивых к заболеванию) индивидуумов уменьшается со временем пропорционально числу контактов с инфицированными. После контакта происходит заражение, восприимчивый переходит в состояние инфицированного.

2. Уравнение числа инфицированных (I):

   $$\frac{dI}{dt} = \frac{\beta I S}{N} - \gamma I,$$

   где γ — коэффициент интенсивности выздоровления инфицированных индивидов.

   Второе уравнение описывает изменение числа инфицированных с течением времени. Уравнение показывает, что скорость увеличения числа заразившихся растет пропорционально числу контактов здоровых и инфицированных и уменьшается по мере выздоровления последних.

3. Уравнение числа выздоровевших (R):

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

где $R(t)$ — численность переболевших индивидов в момент времени t.

Третье уравнение демонстрирует, что число выздоровевших в единицу времени пропорционально числу инфицированных. Иначе говоря, каждый заболевший через некоторое время должен поправиться.

Стоит отметить, что сумма численностей трех групп всегда остается постоянной, т.е. $S + I + R = N$. Коэффициент $R_0 = \frac{\beta}{\gamma}$ называется **«базовым коэффициентом воспроизведения»** [**math?**]. Для каждой болезни есть собственный коэффициент R_0.

Модель SIR может быть использована для прогнозирования темпов распространения заболевания и оценки эффективности мер по его контролю, таких как вакцинация, карантин, социальная дистанцирование и т.д. Также, в зависимости от начальных условий, коэффициента инфицирования, коэффициента выздоровления и других коэффициентов, модель может быть использована для исследования различных вариантов эпидемических сценариев.

3.2 Задача об эпидемии

Отличия модели, предлагаемой для описания в лабораторной работы, от вышеуказанной модели SIR таковы:

1. Введен дополнительный параметр: I^* — критическое значение $I(t)$, после превышения которого инфицированные способны заражать восприимчивых. До этого критического значения инфицированные не заражают восприимчивых.

2. Изменены стандартные символы, отождествляющие коэффициенты: $\beta\frac{I}{N} \to \alpha$ (коэффициент заболеваемости), $\gamma \to \beta$ (коэффициент выздоровления).
3. В соответствии с предыдущими пунктами изменена система уравнений [**rudn?**]:

$$\frac{dS}{dt} = \begin{cases} -\alpha S, \;\; I(t) > I^* \\ 0, \;\; I(t) \le I^* \end{cases}$$

$$\frac{dI}{dt} = \begin{cases} \alpha S - \beta I, \;\; I(t) > I^* \\ -\beta I, \;\; I(t) \le I^* \end{cases}$$

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

4 Выполнение лабораторной работы

4.1 Julia

4.1.1 Задание №1

1. Пишем программу, воспроизводящую модель на языке программирования Julia. (рис. 4.1)

```
using Plots
using DifferentialEquations

alfa = 0.8
betta = 0.2

T = (0.0, 30.0)

N = 25000
I0 = 150
R0 = 15
Ik = 500
S0 = N - I0 - R0

u0 = [S0, I0, R0]
```

```
function F!(du, u, p, t)
    if u[2] > Ik
        du[1] = -alfa*u[1]
        du[2] = alfa*u[1] - betta*u[2]
    else
        du[1] = 0
        du[2] = -betta*u[2]
    end
    du[3] = betta * u[2]
end

prob = ODEProblem(F!, u0, T)
sol = solve(prob, dtmax=0.1)

time = sol.t
S = []
I = []
R = []

for u in sol.u
    s, i, r = u
    push!(S, s)
    push!(I, i)
    push!(R, r)
end

plt = plot(dpi=150, size=(800, 400))
plot!(plt, time, [S, I, R], color = [:red :green :blue], xlabel = "Время", l
savefig(plt, "lab6v1img.png")
```

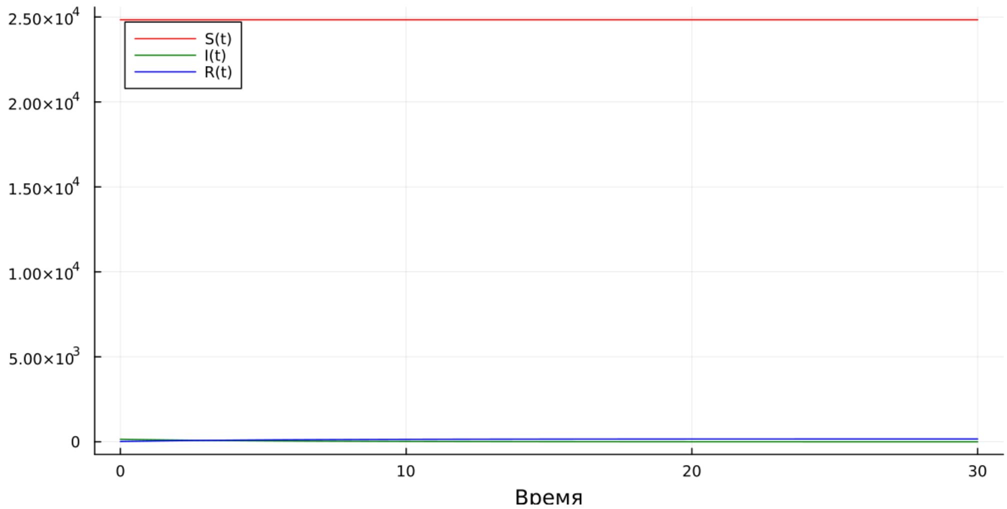

Рис. 4.1: График модели SIR Julia (при $I(t) > I^*$)

4.1.2 Задание №2

1. Изменено значение I^*, которое теперь меньше $I(0)$. Получаем новый график. (рис. 4.2)

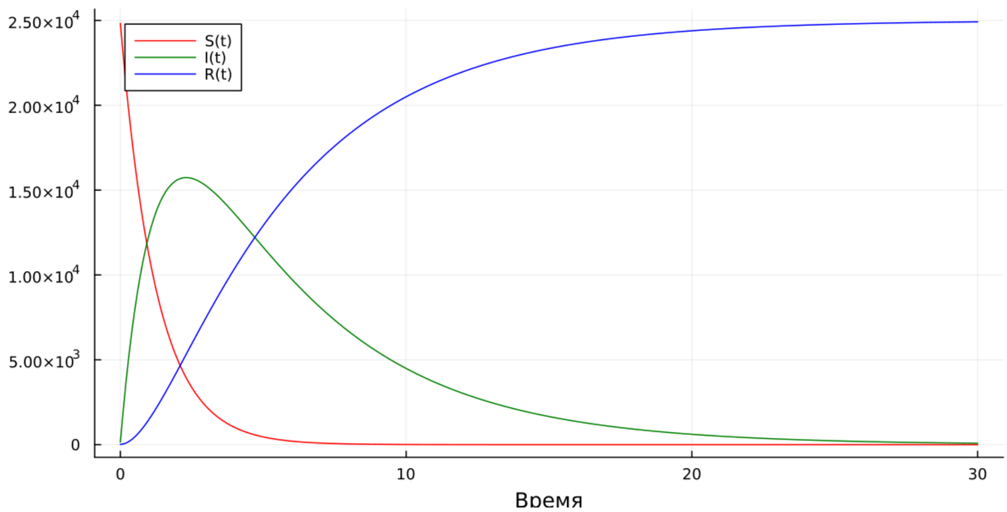

Рис. 4.2: График модели SIR Julia (при $I(t) <= I^*$)

4.2 Modelica

4.2.1 Задание №1

1. По аналогии с Julia пишем программу, воспроизводящую измененную модель SIR на языке моделирования Modelica с использованием ПО OpenModelica. Получаем график. (рис. 4.3)

```
model labv1
  constant Real alfa = 0.8;
  constant Real betta = 0.2;
  constant Integer N = 25000;
  constant Integer Ik = 500;
  Real S(time);
  Real I(time);
  Real R(time);
initial equation
  I = 150;
  R = 15;
  S = N - I - R;
equation
  if I > Ik then
    der(S) = -alfa*S;
    der(I) = alfa*S - betta*I;
  else
    der(S) = 0;
    der(I) = -betta*I;
  end if;
  der(R) = betta*I;
end labv1;
```

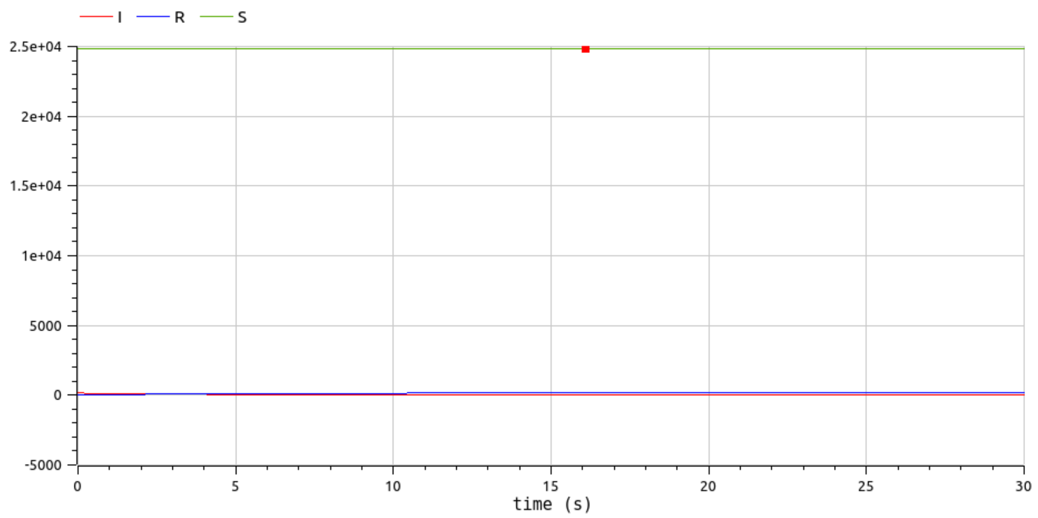

Рис. 4.3: График модели SIR Modelica (при $I(t) > I^*$)

4.2.2 Задание №2

1. По аналогии с Julia пишем программу для второго случая. Получаем новый график. (рис. 4.4)

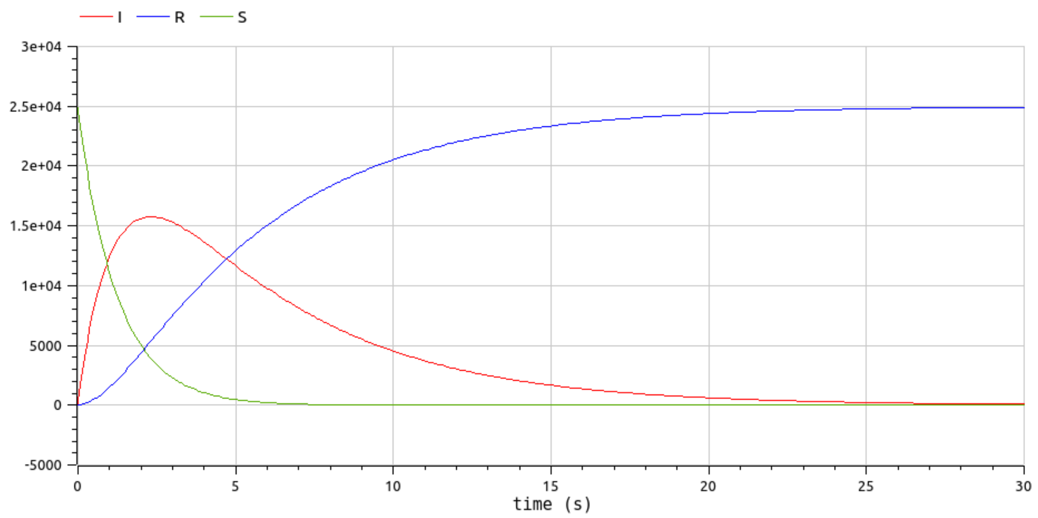

Рис. 4.4: График модели SIR Modelica (при $I(t) <= I^*$)

5 Анализ результатов

На текущем примере построения математической модели, схожей с моделью SIR, мы можем продолжить сравнивать язык программирования Julia и язык моделирования Modelica. Говоря честно, по сравнению с анализом результатов при выполнении предыдущей лабораторной работы мало что изменилось: тенденция к сглаживанию негативных моментов при выполнении лабораторной работы на языке программирования Julia продолжается. Со временем и с новыми заданиями, решаемыми при помощи библиотеки DifferentialEquations, скорость написания программ на Julia почти сравнялась с таковой скоростью при использовании Modelica.

На обоих языках одинаково просто добавляются условия в уравнения, как в текущем случае. Однако, хочется заметить, что в Modelica в разы удобнее составлять уравнения, т.к. все переменные, зависящие от времени, подписываются заданными ранее символами в отличие от Julia, где каждой переменной соответствует элемент массива. Такая реализация может запутать, особенно при условии наличие трех и более переменных, зависящих от времени и используемых в системе. Это может привести к ошибкам, связанными с усидчивостью, при написании системы.

6 Выводы

Продолжил знакомство с функционалом языка программирования Julia, дополнительных библиотек (DifferentialEquations, Plots), интерактивного блокнота Pluto, а также интерактивной командной строкой REPL. Продолжил ознакомление с языком моделирования Modelica и программным обеспечением OpenModelica. Используя эти средства, описал математическую модель, схожую с моделью SIR.

Список литературы