Отчёт по лабораторной работе

Лабораторная работа №6 (вариант 10)

Сергеев Тимофей Сергеевич

Содержание

1 Цель работы **5**

2 Задание **6**

3 Теоретическое введение **7**

4 Выполнение лабораторной работы **8**

5 Выводы **17**

Список литературы **18**

Список иллюстраций

4.1 Реализация модели на языке Modelica 9
4.2 Результат моделирования на языке Modelica 10
4.3 Реализация модели на языке Modelica 11
4.4 Результат моделирования на языке Modelica 12
4.5 Подключаем библиотеки, задаём коэффициенты и функцию, решающую дифференциальные уравнения. Затем зададим начальные условия. 12
4.6 Выполним функцию с данными значениями. Создадим три пустых массива, в которые мы передадим полученные значения. Затем с помощью функционала библиотеки Plots создадим поле для вывода результата. 13
4.7 Выведем на экран полученные графы и сохраним результат в формате png 14
4.8 Результат работы программы 15
4.9 Меняем только начальную часть кода 15
4.10 Результат работы программы 16

Список таблиц

1 Цель работы

Построить графики изменения числа особей в каждой из трех групп во время эпидемии: все проживающие на острове, заболевшие люди, здоровые люди с иммунитетом. Рассмотреть, как будет протекать эпидемия в случаях, когда число заоблевших не превышает критическое значение и когда превышает.

2 Задание

- Написать код на языке Julia.
- Написать код на языке Modelica для случаев.
- Составить отчёт на языке Markdown и сконвертировать его в docx и pdf.
- Подготовить презентацию на языке Markdown и защитить её.

3 Теоретическое введение

Julia – высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков (например, MATLAB и Octave), однако имеет некоторые существенные отличия. Julia написан на Си, C++ и Scheme. Имеет встроенную поддержку многопоточности и распределённых вычислений, реализованные в том числе в стандартных конструкциях. [1]

OpenModelica – свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. Активно развивается Open Source Modelica Consortium, некоммерческой неправительственной организацией. Open Source Modelica Consortium является совместным проектом RISE SICS East AB и Линчёпингского университета. [2]

Modelica -— объектно-ориентированный, декларативный, мультидоменный язык моделирования для компонентно-ориентированного моделирования сложных систем, в частности, систем, содержащих механические, электрические, электронные, гидравлические, тепловые, энергетические компоненты, а также компоненты управления и компоненты, ориентированные на отдельные процессы. [3]

4 Выполнение лабораторной работы

1. Рассмотрим код на языке Modelica. Объявим переменные и коэффициенты типа Real (потому что это тип с плавающим знаком, наиболее подходящий для решения дифференциальных уравнений). Затем введём начальные значение для переменных, означающих число людей на острове (N), число заболевших людей (I), число здоровых людей с иммунитетом (R), подставив данные из условия. После этого пропишем решение наших дифференциальных уравнений (рис. 4.1).

```
model lab6_1
Real S, I, R;
Real a=0.01;
Real b=0.02;
Real N=16000;
Real t=time;
initial equation
I=116;
S=N-I-R;
R=16;
equation
der(S)= 0;
der(I)= -b*I;
der(R)= b*I;
end lab6_1;
```

Рис. 4.1: Реализация модели на языке Modelica

2. Затем установим настройки симуляции (начальное (0) и конечное (200) время и шаг (0.01)) и запустим симуляцию. Получим следующий результат (рис. 4.2).

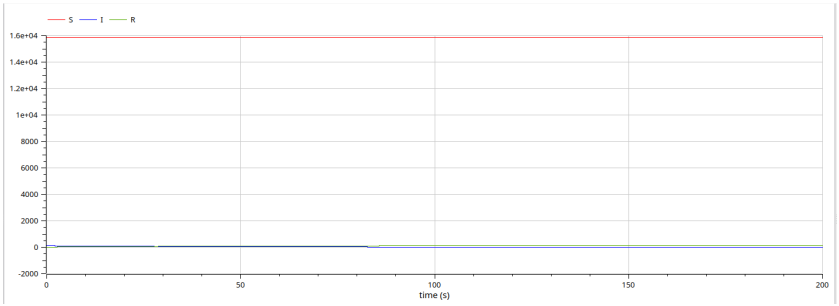

Рис. 4.2: Результат моделирования на языке Modelica

3. Аналогичным образом запишем реализацию второго случая (рис. 4.3).

```
1   model lab6_2
2   Real S, I, R;
3   Real a=0.01;
4   Real b=0.02;
5   Real N=16000;
6   Real t=time;
7   initial equation
8   I=116;
9   S=N-I-R;
10  R=16;
11  equation
12  der(S)= -a*S;
13  der(I)= a*S-b*I;
14  der(R)= b*I;
15  end lab6_2;
```

Рис. 4.3: Реализация модели на языке Modelica

4. Затем установим настройки симуляции (начальное (0) и конечное (200) время и шаг (0.01)) и запустим симуляцию. Получим следующий результат (рис. 4.4).

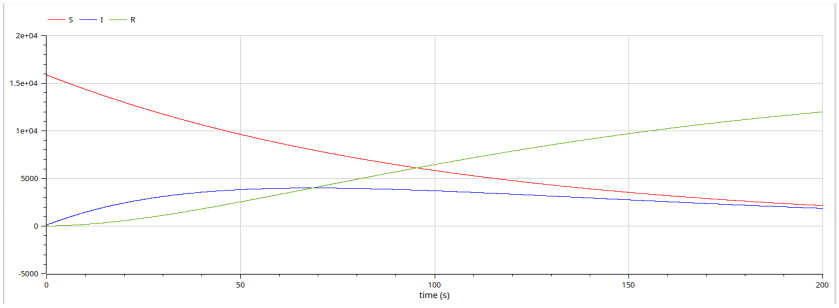

Рис. 4.4: Результат моделирования на языке Modelica

5. Теперь опишем эти случаи на языке Julia (рис. 4.5, 4.6, 4.7).

```
1   using DifferentialEquations
2   using Plots
3
4   a=0.01
5   b=0.02
6   N=16000
7   I0=116
8   R0=16
9   S0=N-I0-R0
10
11  function F!(du, u, p, t,)
12      du[1] = 0
13      du[2] = -b*u[2]
14      du[3] = b*u[2]
15  end
16
17  begin
18      u0 = [S0, I0,R0]
19      T = [0.0, 200.0]
20      prob = ODEProblem(F!, u0, T)
21  end
```

Рис. 4.5: Подключаем библиотеки, задаём коэффициенты и функцию, решающую дифференциальные уравнения. Затем зададим начальные условия.

```
22
23  sol = solve(prob, dtmax=0.01)
24
25  const X = Float64[]
26  const Y = Float64[]
27  const Z = Float64[]
28
29  for u in sol.u
30      x, y, z = u
31      push!(X, x)
32      push!(Y, y)
33      push!(Z, z)
34  end
35
36  plt = plot(
37      dpi = 300,
38      size=(1000,600),
39      plot_title="Задача об эпидемии"
40  )
```

Рис. 4.6: Выполним функцию с данными значениями. Создадим три пустых массива, в которые мы передадим полученные значения. Затем с помощью функционала библиотеки Plots создадим поле для вывода результата.

```
41
42  plot!(
43      plt,
44      sol.t,
45      X,
46      color=:blue,
47      label="S(t)"
48  )
49
50  plot!(
51      plt,
52      sol.t,
53      Y,
54      color=:red,
55      label="I(t)"
56  )
57
58  plot!(
59      plt,
60      sol.t,
61      Z,
62      color=:green,
63      label="R(t)"
64  )
65
66  savefig(plt, "lab6_1.png")
```

Рис. 4.7: Выведем на экран полученные графы и сохраним результат в формате png

6. Получим следующий результат (рис. 4.8).

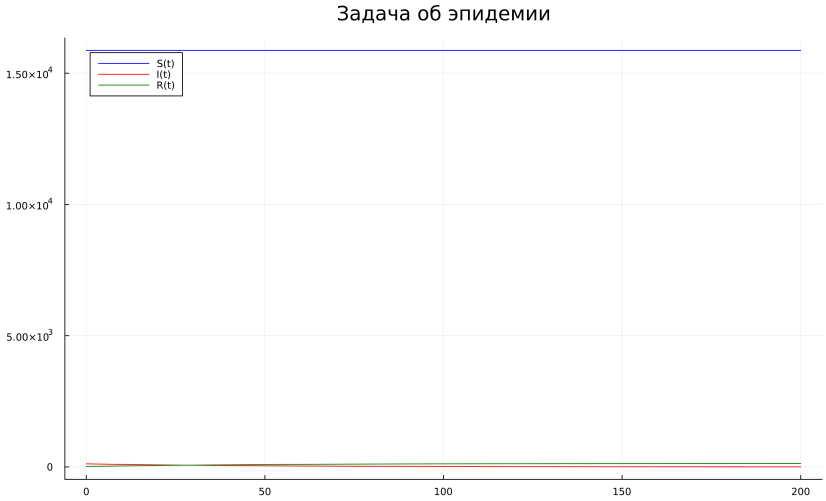

Рис. 4.8: Результат работы программы

7. Для второго случая необходимо только поменять уравнения в случае, когда число заболевших превышает критическое значение(рис. 4.9).

```
afs > .dk.sci.pfu.edu.ru > home > t > s > tssergeev > work > study > 2022-2
1   using DifferentialEquations
2   using Plots
3
4   a=0.01
5   b=0.02
6   N=16000
7   I0=116
8   R0=16
9   S0=N-I0-R0
10
11  function F!(du, u, p, t,)
12      du[1] = -a*u[1]
13      du[2] = a*u[1] - b*u[2]
14      du[3] = b*u[2]
15  end
16
17  begin
18      u0 = [S0, I0,R0]
19      T = [0.0, 200.0]
20      prob = ODEProblem(F!, u0, T)
21  end
```

Рис. 4.9: Меняем только начальную часть кода

8. Получим следующий результат (рис. 4.10).

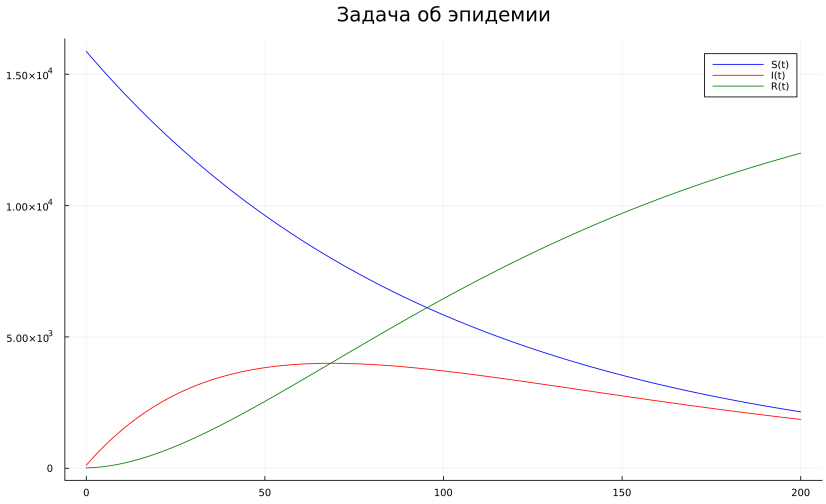

Рис. 4.10: Результат работы программы

5 Выводы

Выполнив данную лабораторную работу, мы продолжили знакомство с языками программирования Julia и Modelica. Сравнивая реализацию одной программы на этих двух языках, можно заметить, что реализация на языке Modelica заметно проще и более точно показывает результат, поскольку можно отследить значения переменных с максимальной точностью на любом отрезке времени.

Список литературы

1. Julia [Электронный ресурс]. Free Software Foundation, 2023. URL: https://ru.wikipedia.org/wiki/Julia_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F).
2. OpenModelica [Электронный ресурс]. Free Software Foundation, 2023. URL: https://ru.wikipedia.org/wiki/OpenModelica.
3. Modelica [Электронный ресурс]. Free Software Foundation, 2023. URL: https://ru.wikipedia.org/wiki/Modelica.