Отчёт по лабораторной работе

Лабораторная работа №6 (вариант 10)

Сергеев Тимофей Сергеевич

Содержание

# 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

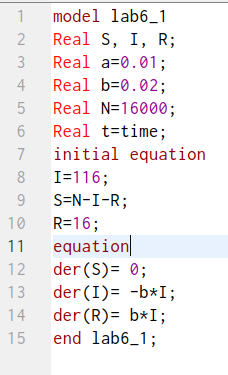
**Julia** – высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков (например, MATLAB и Octave), 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]

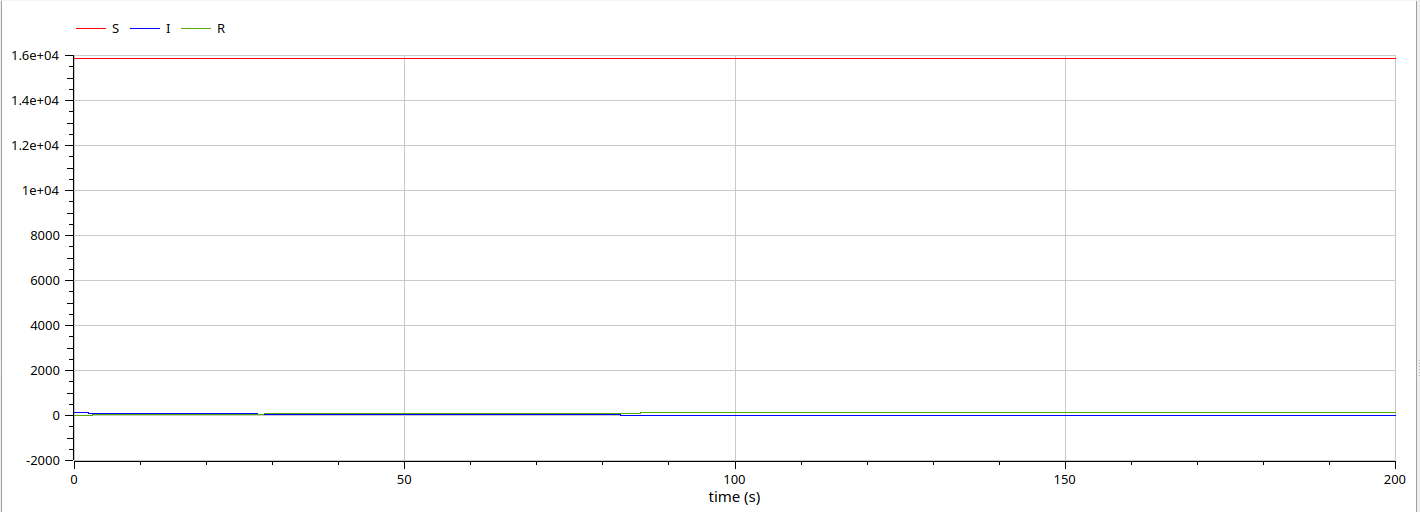
# 4 Выполнение лабораторной работы

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. ??).



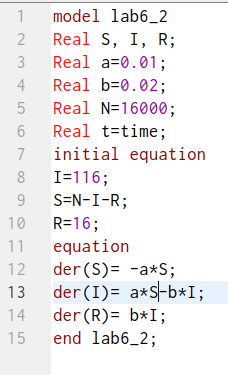
Реализация модели на языке Modelica

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



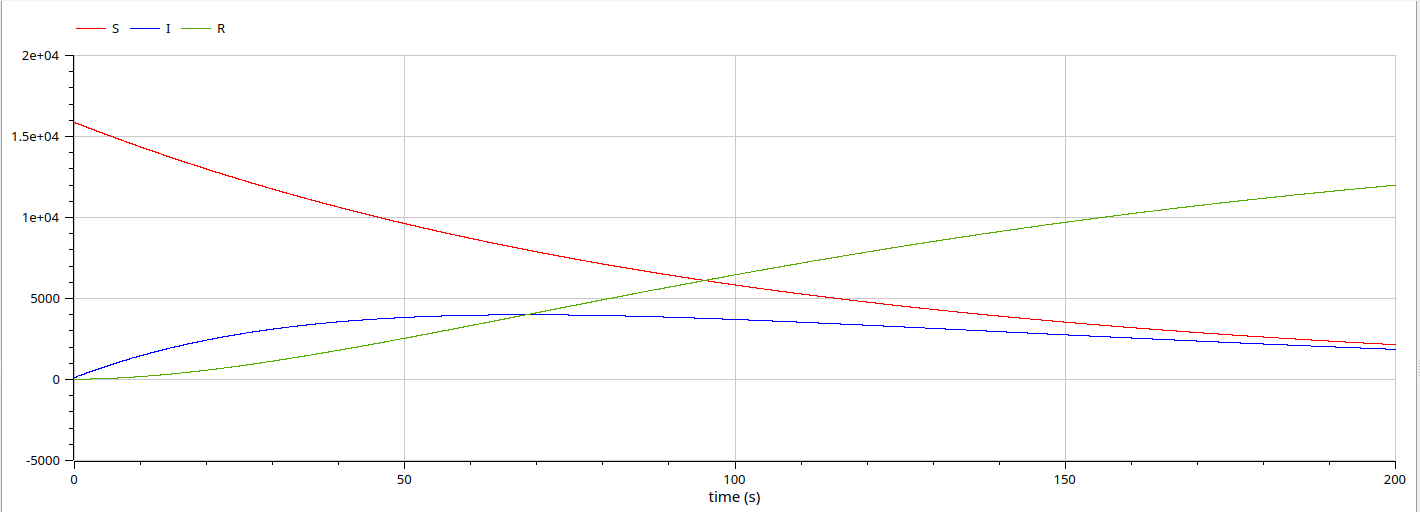
Результат моделирования на языке Modelica

1. Аналогичным образом запишем реализацию второго случая (рис. ??).



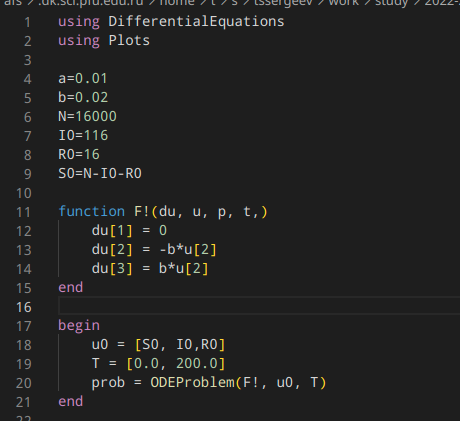
Реализация модели на языке Modelica

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

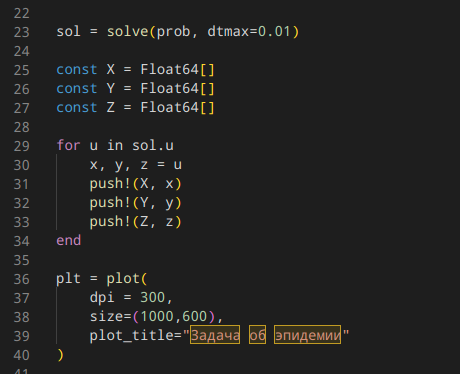


Результат моделирования на языке Modelica

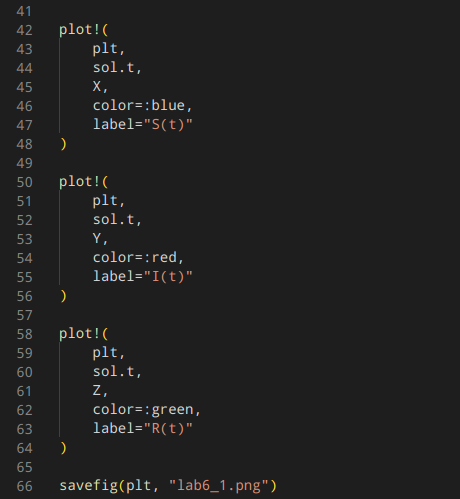
1. Теперь опишем эти случаи на языке Julia (рис. ??, ??, ??).



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

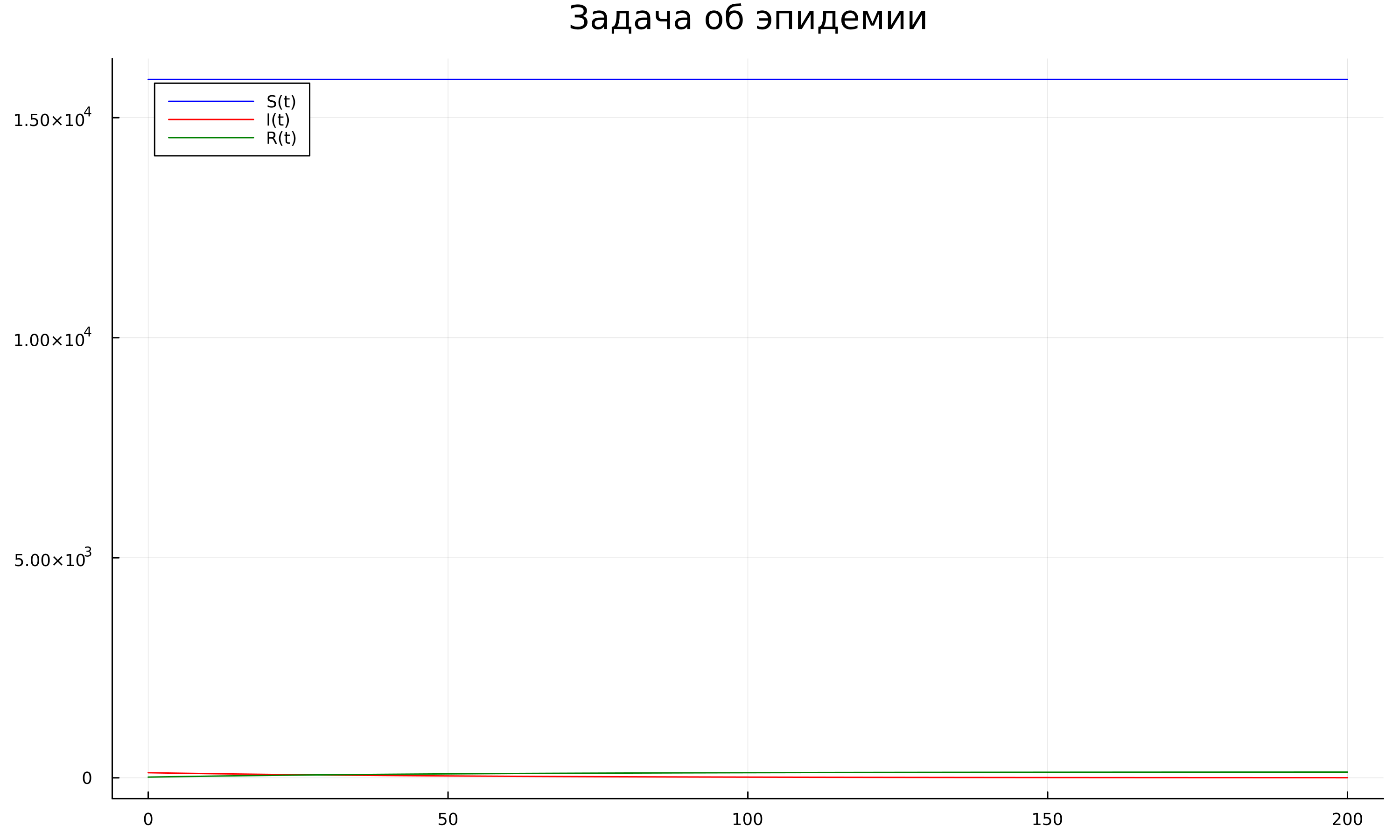


Выполним функцию с данными значениями. Создадим три пустых массива, в которые мы передадим полученные значения. Затем с помощью функционала библиотеки Plots создадим поле для вывода результата.



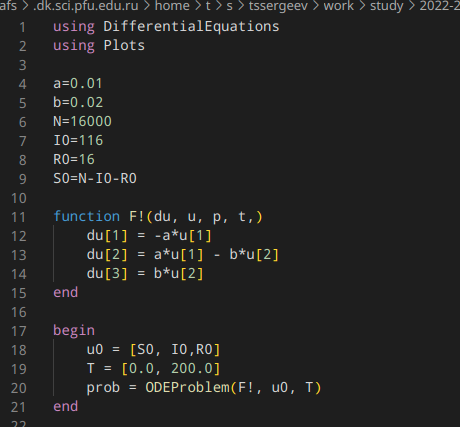
Выведем на экран полученные графы и сохраним результат в формате png

1. Получим следующий результат (рис. ??).



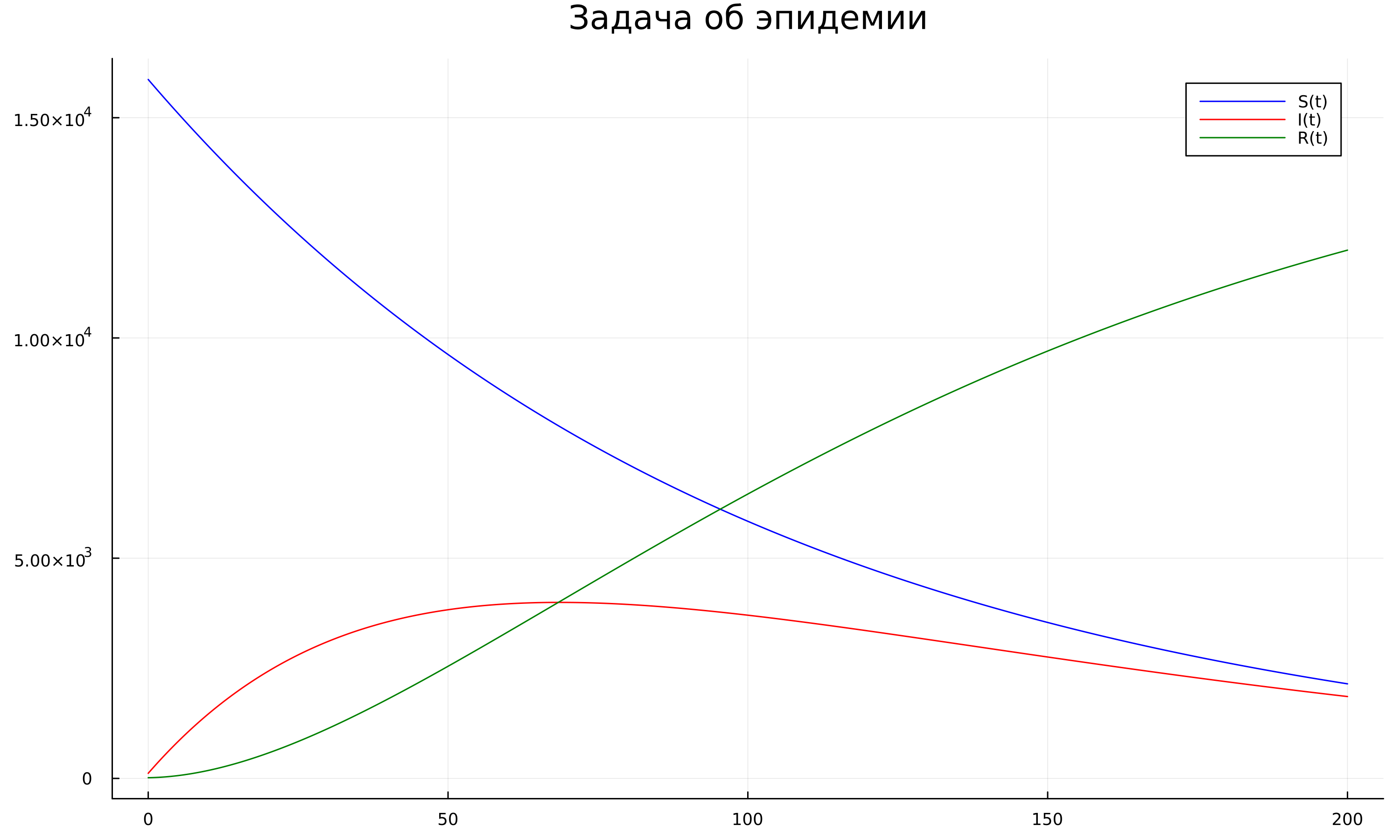
Результат работы программы

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



Меняем только начальную часть кода

1. Получим следующий результат (рис. ??).



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>.