

Отчёт по лабораторной работе

Лабораторная работа №3 (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 первого случая на языке Modelica	9
4.2	Результат симуляции для первого случая	9
4.3	Реализация второго случая на языке Modelica	10
4.4	Результат симуляции для второго случая	10
4.5	Подключаем библиотеки, задаём значение коэффициентов, задаём нашу функцию, решающую дифференциальные уравнения. Затем зададим значения наших войск и времени.	11
4.6	Выполним функцию с данными значениями. Создадим два пустых массива, в которые мы передадим полученные значения. Затем с помощью функционала библиотеки Plots создадим поле для вывода результата.	11
4.7	Выведем на экран полученные графы и сохраним результат в формате png	12
4.8	Результат работы программы для первого случая	13
4.9	Подключаем библиотеки, задаём значение коэффициентов, задаём нашу функцию, решающую дифференциальные уравнения. Затем зададим значения наших войск и времени.	14
4.10	Выполним функцию с данными значениями. Создадим два пустых массива, в которые мы передадим полученные значения. Затем с помощью функционала библиотеки Plots создадим поле для вывода результата.	14
4.11	Выведем на экран полученные графы и сохраним результат в формате png	15
4.12	Результат работы программы для первого случая	16

Список таблиц

1 Цель работы

Построить математическую модель боевых действий двух стран - X и Y . Рассмотреть два случая: с участием партизанских отрядов и без.

2 Задание

- Написать код для первого и второго случая на языке Julia.
- Написать код для первого и второго случая на языке Modelica.
- 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]

Моделирование боевых действий является важнейшей научной и практической задачей, направленной на предоставление командованию количественных оснований для принятия решений. [3]

4 Выполнение лабораторной работы

1. Рассмотрим код на языке Modelica с для первого случая. Объявим переменные и коэффициенты типа Real (потому что это тип с плавающим знаком, наиболее подходящий для решения дифференциальных уравнений). Затем инициализируем x и y , подставим значения армий, данные нам в условии. После этого пропишем решение наших дифференциальных уравнений (рис. 4.1).


```

1  model lab31
2  Real x, y;
3  Real a=0.45;
4  Real b=0.86;
5  Real c=0.49;
6  Real h=0.73;
7  Real t=time;
8  initial equation
9  x=21200;
10 y=9800;
11 equation
12 der(x)= -a*x - b*y + sin(t+1);
13 der(y)= -c*x - h*y + cos(t+2);
14 end lab31;

```

Рис. 4.1: Реализация первого случая на языке Modelica

2. Результат симуляции для первого случая (рис. 4.2).

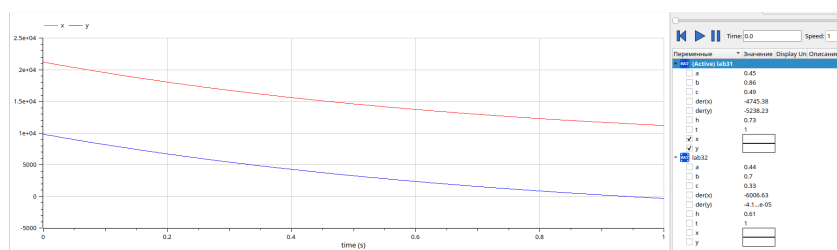


Рис. 4.2: Результат симуляции для первого случая

3. Аналогично пишется код и для второго случая, меняются коэффициенты и второе дифференциальное уравнение (рис. 4.3).

```

1  model lab32
2  Real x, y;
3  Real a=0.44;
4  Real b=0.7;
5  Real c=0.33;
6  Real h=0.61;
7  Real t=time;
8  initial equation
9  x=21200;
10 y=9800;
11 equation
12 der(x)= -a*x - b*y + sin(2*t);
13 der(y)= -c*x*y - h*y + cos(t)+1;
14 end lab32;

```

Рис. 4.3: Реализация второго случая на языке Modelica

4. Результат симуляции для первого случая (рис. 4.4).

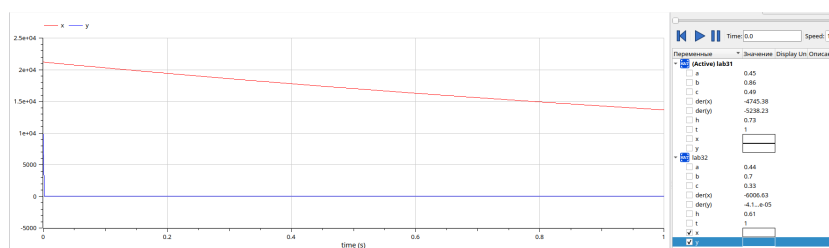


Рис. 4.4: Результат симуляции для второго случая

5. Теперь опишем первый случай на языке Julia.(рис. 4.5, 4.6, 4.7).

```

1  using DifferentialEquations
2  using Plots
3
4  a=0.45
5  b=0.86
6  c=0.49
7  h=0.73
8
9  function F!(du, u, p, t,)
10     du[1] = -a*u[1] - b*u[2] + sin(t+1)
11     du[2] = -c*u[1] - h*u[2] + cos(t+2)
12 end
13
14 begin
15     u0 = [21200, 9800]
16     T = [0.0, 1.0]
17     prob = ODEProblem(F!, u0, T)
18 end
19

```

Рис. 4.5: Подключаем библиотеки, задаём значение коэффициентов, задаём нашу функцию, решающую дифференциальные уравнения. Затем зададим значения наших войск и времени.

```

sol = solve(prob, dtmax=0.05)

const X = Float64[]
const Y = Float64[]

for u in sol.u
    x, y = u
    push!(X, x)
    push!(Y, y)
end

plt = plot(
    dpi = 300,
    size=(1000,600),
    plot_title="Модель боевых действий без участия партизанских отрядов"
)

```

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

```

plot!(
    plt,
    sol.t,
    X,
    xlabel="Время",
    color=:blue,
    label="Армия страны X"
)

plot!(
    plt,
    sol.t,
    Y,
    ylabel="Армия",
    color=:red,
    label="Армия страны Y"
)

savefig(plt, "lab31.png")

```

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

6. Получим следующий результат (рис. 4.8).

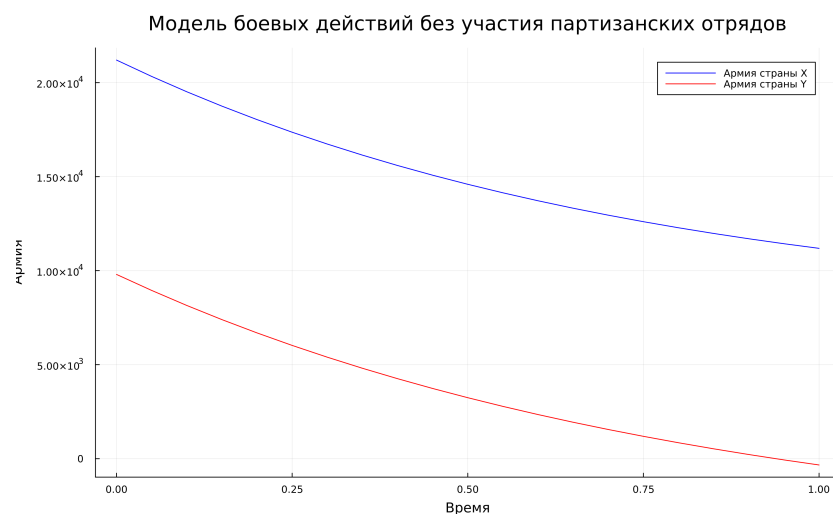


Рис. 4.8: Результат работы программы для первого случая

7. Опишем тем же образом второй случай на языке Julia.(рис. 4.9, 4.10, 4.11).

```

using DifferentialEquations
using Plots

a=0.44
b=0.7
c=0.33
h=0.61

function F!(du, u, p, t,)
    du[1] = -a*u[1] - b*u[2] + sin(2*t)
    du[2] = -c*u[1]*u[2] - h*u[2] + cos(t)+1
end

begin
    u0 = [21200, 9800]
    T = [0.0, 1.0]
    prob = ODEProblem(F!, u0, T)
end

sol = solve(prob, dtmax=0.05)

const X = Float64[]
const Y = Float64[]

```

Рис. 4.9: Подключаем библиотеки, задаём значение коэффициентов, задаём нашу функцию, решающую дифференциальные уравнения. Затем зададим значения наших войск и времени.

```

for u in sol.u
    x, y = u
    push!(X, x)
    push!(Y, y)
end

plt = plot(
    dpi = 300,
    size=(800,600),
    plot_title="Модель боевых действий с участием партизанских отрядов"
)

```

Рис. 4.10: Выполним функцию с данными значениями. Создадим два пустых массива, в которые мы передадим полученные значения. Затем с помощью функционала библиотеки Plots создадим поле для вывода результата.

```

plot!(
    plt,
    sol.t,
    X,
    xlabel="Время",
    color=:blue,
    label="Армия страны X"
)

plot!(
    plt,
    sol.t,
    Y,
    ylabel="Армия",
    color=:red,
    label="Армия страны Y"
)

savefig(plt, "lab32.png")

```

Рис. 4.11: Выведем на экран полученные графы и сохраним результат в формате png

8. Получим следующий результат (рис. 4.12).

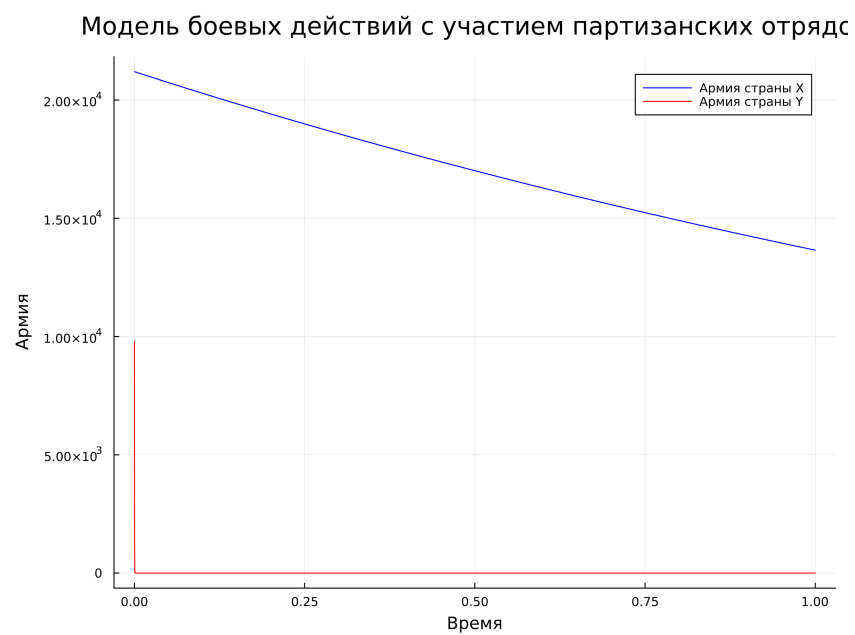


Рис. 4.12: Результат работы программы для первого случая

5 Выводы

Выполнив данную лабораторную работу, мы продолжили знакомство с языком программирования Julia, начали изучение языка 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 графика.

Список литературы

1. Julia [Электронный ресурс]. Free Software Foundation, 2023. URL: [https://ru.wikipedia.org/wiki/Julia_\(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\)](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. Модель боевых действий [Электронный ресурс]. Free Software Foundation, 2023. URL: http://crm-en.ics.org.ru/uploads/crmissues/crm_2020_1/2020_01_14.pdf.