

Отчёт по лабораторной работе

Лабораторная работа №8 (вариант 10)

Сергеев Тимофей Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Реализация модели на языке Modelica	8
4.2	Результат моделирования на языке Modelica	9
4.3	Реализация модели на языке Modelica	9
4.4	Результат моделирования на языке Modelica	10
4.5	Подключаем библиотеки, задаём коэффициенты и функцию, решающую дифференциальные уравнения. Затем зададим начальные условия.	10
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.	11
4.7	Выведем на экран полученные графы и сохраним результат в формате png	11
4.8	Результат работы программы	12
4.9	Меняем только начальную часть кода	12
4.10	Результат работы программы	13

Список таблиц

1 Цель работы

Постройте графики изменения оборотных средств фирмы 1 и фирмы 2 без учета постоянных издержек и с введенной нормировкой для случаев 1 и 2.

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]

Модель Бертрана или конкуренция по Бертрону — модель ценовой конкуренции на олигополистическом рынке, сформулированная французским математиком и экономистом Жозефом Бертреном в 1883 году. Модель описывает поведение фирм на олигополистическом рынке, конкурирующих за счет изменения уровня цен на свою продукцию. Парадоксальный вывод модели — фирмы будут назначать цену, равную предельным издержкам, как и фирмы в условиях совершенной конкуренции — назван парадоксом Бертрена. [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 значения для переменных, подставив данные из условия. После этого пропишем решение наших дифференциальных уравнений (рис. 4.1).

```
1 model lab8_1
2 Real M1, M2;
3 Real p_cr=22;
4 Real N=25;
5 Real q=1;
6 Real tau1=18;
7 Real tau2=21;
8 Real p1=15;
9 Real p2=12;
10 Real a1=p_cr/(tau1*tau1*p1*p1*N*q);
11 Real a2=p_cr/(tau2*tau2*p2*p2*N*q);
12 Real b=p_cr/(tau1*tau1*p1*p1*tau2*tau2*p2*p2*N*q);
13 Real c1=(p_cr-p1)/(tau1*p1);
14 Real c2=(p_cr-p2)/(tau2*p2);
15 Real t=time;
16 initial equation
17 M1=2.7;
18 M2=2;
19 equation
20 der(M1)=(c1/c1)*M1 - (a1/c1)*M1*M1 - (b/c1)*M1*M2;
21 der(M2)=(c2/c1)*M2 - (a2/c1)*M2*M2 - (b/c1)*M1*M2;
22 end lab8_1;
```

Рис. 4.1: Реализация модели на языке Modelica

2. Затем установим настройки симуляции (начальное (0) и конечное (30) время и шаг (0.01)) и запустим симуляцию. Получим следующий результат (рис. 4.2).

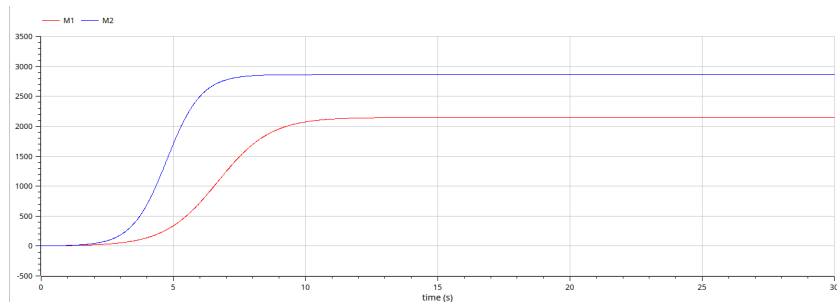


Рис. 4.2: Результат моделирования на языке Modelica

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

```

1  model lab8_2
2  Real M1, M2;
3  Real p_cr=22;
4  Real N=25;
5  Real q=1;
6  Real tau1=18;
7  Real tau2=21;
8  Real p1=15;
9  Real p2=12;
10 Real a1=p_cr/(tau1*tau1*p1*p1*N*q);
11 Real a2=p_cr/(tau2*tau2*p2*p2*N*q);
12 Real b=p_cr/(tau1*tau1*p1*p1*tau2*tau2*p2*p2*N*q);
13 Real c1=(p_cr-p1)/(tau1*p1);
14 Real c2=(p_cr-p2)/(tau2*p2);
15 Real t=time;
16 initial equation
17 M1=2.7;
18 M2=2;
19 equation
20 der(M1)=(c1/c1)*M1 - (a1/c1)*M1*M1 - (b/c1 + 0.0019)*M1*M2;
21 der(M2)=(c2/c1)*M2 - (a2/c1)*M2*M2 - (b/c1)*M1*M2;
22 end lab8_2;

```

Рис. 4.3: Реализация модели на языке Modelica

4. Затем установим настройки симуляции (начальное (0) и конечное (30) время и шаг (0.01)) и запустим симуляцию. Получим следующий результат (рис. 4.4).

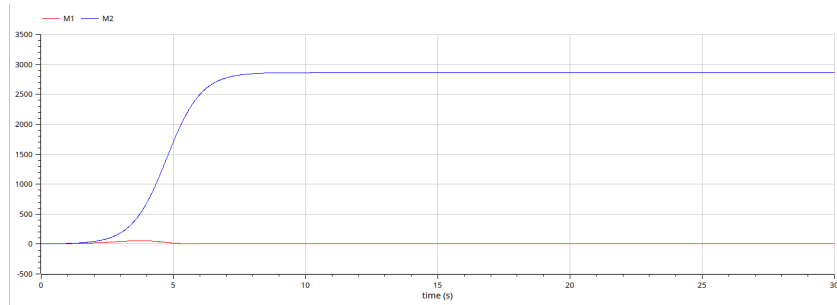


Рис. 4.4: Результат моделирования на языке Modelica

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

```

1  using DifferentialEquations
2  using Plots
3
4  p_cr=22
5  N=25
6  q=1
7  tau1=18
8  tau2=21
9  p1=15
10 p2=12
11
12 a1=p_cr/(tau1*tau1*p1*p1*N*q)
13 a2=p_cr/(tau2*tau2*p2*p2*N*q)
14 b=p_cr/(tau1*tau1*p1*p1*tau2*tau2*p2*p2*N*q)
15 c1=(p_cr-p1)/(tau1*p1)
16 c2=(p_cr-p2)/(tau2*p2)
17
18 function F!(du, u, p, t)
19     du[1]=(c1/c1)*u[1] - (a1/c1)*u[1]*u[1] - (b/c1)*u[1]*u[2]
20     du[2]=(c2/c1)*u[2] - (a2/c1)*u[2]*u[2] - (b/c1)*u[1]*u[2]
21 end
22
23 begin
24     u0 = [2.7, 2]
25     T = [0.0, 30.0]
26     prob = ODEProblem(F!, u0, T)
27 end
28

```

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

```

28
29     sol = solve(prob, dtmax=0.01)
30
31     const X = Float64[]
32     const Y = Float64[]
33
34     for u in sol.u
35         x, y = u
36         push!(X, x)
37         push!(Y, y)
38     end
39
40     plt = plot(
41         dpi = 300,
42         size = (1000,600),
43         plot_title="Модель конкуренции двух фирм"
44     )
45

```

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.

```

45
46     plot!(
47         plt,
48         sol.t,
49         X,
50         color=:blue,
51         label="M1"
52     )
53
54     plot!(
55         plt,
56         sol.t,
57         Y,
58         color=:red,
59         label="M2"
60     )
61
62     savefig(plt, "lab8_1.png")

```

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

6. Получим следующий результат (рис. 4.8).

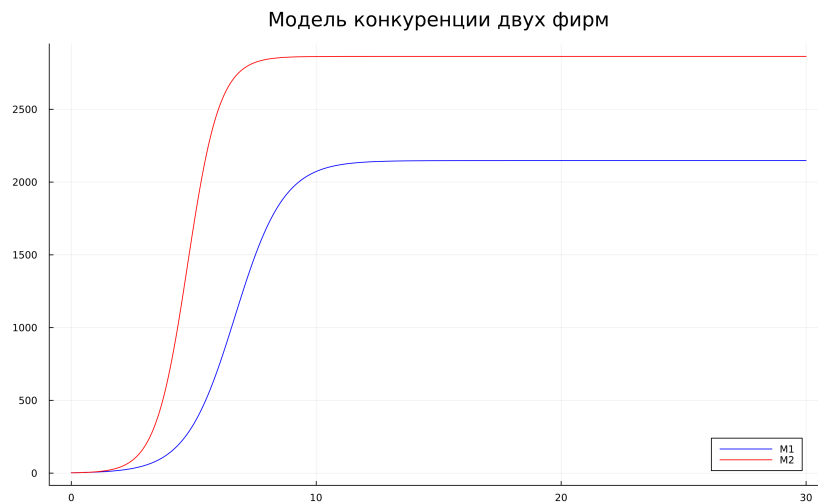


Рис. 4.8: Результат работы программы

7. Для второго случая необходимо только поменять первое уравнение(рис. 4.9).

```
tau1=18
tau2=21
p1=15
p2=12

a1=p_cr/(tau1*tau1*p1*p1*N*q)
a2=p_cr/(tau2*tau2*p2*p2*N*q)
b=p_cr/(tau1*tau1*p1*p1*tau2*tau2*p2*p2*N*q)
c1=(p_cr-p1)/(tau1*p1)
c2=(p_cr-p2)/(tau2*p2)

function F!(du, u, p, t)
    du[1]=(c1/c1)*u[1] - (a1/c1)*u[1]*u[1] - (b/c1 + 0.0019)*u[1]*u[2]
    du[2]=(c2/c1)*u[2] - (a2/c1)*u[2]*u[2] - (b/c1)*u[1]*u[2]
end

begin
    u0 = [2.7, 2]
```

Рис. 4.9: Меняем только начальную часть кода

8. Получим следующий результат (рис. 4.10).

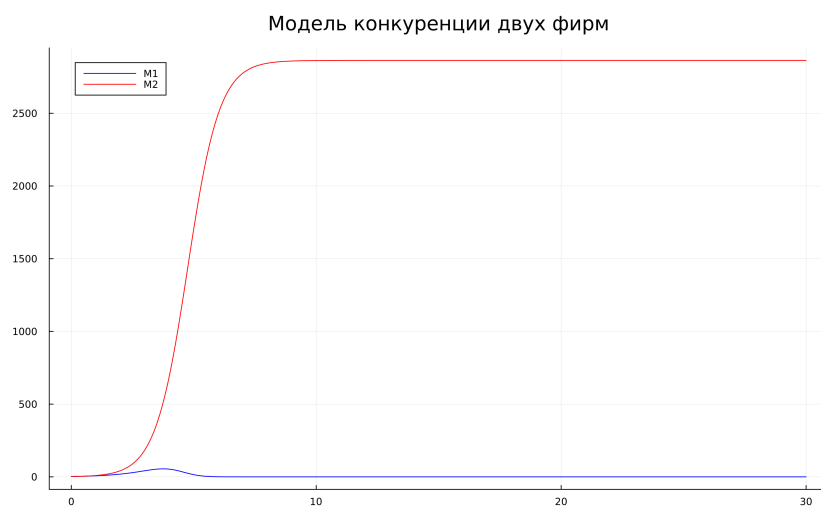


Рис. 4.10: Результат работы программы

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_\(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\)](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 Бертрана [Электронный ресурс]. Free Software Foundation, 2023. URL: https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%91%D0%B5%D1%80%D1%82%D1%80%D0%B0%D0%BD%D0%B0.