Лабораторная работа №8

Модель конкуренции двух фирм

Ильин Андрей Владимирович

Содержание

Список иллюстраций

1 Цель работы

Рассмотреть модель конкуренции двух фирм. Построить вышеуказанную модель средствами OpenModellica и Julia.

2 Задачи

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

Нормировка: $t = c_1\theta$

Начальными условия и параметры: $M_0^1 = 6.8, M_0^2 = 6, p_{cr} = 35, N = 31, q = 1, \tau_1 = 18, \tau_2 = 23, p_1 = 1.5, p_2 = 8.7$

Значения $p_{cr}, p_{1,2}, N$ указаны в тысячах единиц, а значения $M_{1,2}$ указаны в млн. единиц.

3 Среда

- Julia – это открытый свободный высокопроизводительный динамический язык высокого уровня, созданный специально для технических (математических) вычислений. Его синтаксис близок к синтаксису других сред технических вычислений, таких как Matlab и Octave. [1]
- OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. [2]

4 Теоретическое введение

4.1 Случай I

Рассмотрим две фирмы, производящие взаимозаменяемые товары одинакового качества и находящиеся в одной рыночной нише. Считаем, что в рамках нашей модели конкурентная борьба ведётся только рыночными методами. То есть, конкуренты могут влиять на противника путем изменения параметров своего производства: себестоимость, время цикла, но не могут прямо вмешиваться в ситуацию на рынке («назначать» цену или влиять на потребителей каким-либо иным способом.) Будем считать, что постоянные издержки пренебрежимо малы, и в модели учитывать не будем. В этом случае динамика изменения объемов продаж фирмы 1 и фирмы 2 описывается следующей системой уравнений:[3]

$$\begin{cases} \frac{dM_1}{d\theta} = M_1 - \frac{b}{c_1}M_1M_2 - \frac{a_1}{c_1}M_1^2 \\ \frac{dM_2}{d\theta} = \frac{c_2}{c_1}M_2 - \frac{b}{c_1}M_1M_2 - \frac{a_2}{c_1}M_2^2 \end{cases}$$

4.2 Случай II

Рассмотрим модель, когда, помимо экономического фактора влияния (изменение себестоимости, производственного цикла, использование кредита и т.п.), используются еще и социально-психологические факторы – формирование общественного предпочтения одного товара другому, не зависимо от их качества и цены. В этом случае взаимодействие двух фирм будет зависеть друг от друга,

соответственно коэффициент перед $M_1 M_2$ будет отличаться. Пусть в рамках рассматриваемой модели динамика изменения объемов продаж фирмы 1 и фирмы 2 описывается следующей системой уравнений:

$$\begin{cases} \frac{dM_1}{d\theta} = M_1 - (\frac{b}{c_1} + 0.00067) M_1 M_2 - \frac{a_1}{c_1} M_1^2 \\ \frac{dM_2}{d\theta} = \frac{c_2}{c_1} M_2 - \frac{b}{c_1} M_1 M_2 - \frac{a_2}{c_1} M_2^2 \end{cases}$$

В обоих случаях: $a_1 = \frac{p_{cr}}{\tau_1^2 p_1^2 N q}$, $a_2 = \frac{p_{cr}}{\tau_2^2 p_2^2 N q}$, $b = \frac{p_{cr}}{\tau_1^2 p_1^2 \tau_2^2 p_2^2 N q}$, $c_1 = \frac{p_{cr} - p_1}{\tau_1 p_1}$, $c_2 = \frac{p_{cr} - p_2}{\tau_2 p_2}$.

5 Выполнение лабораторной работы

1. Начнем выполнения поставленных задач в Julia. Для этого запустим Pluto [4]. (рис. 5.1)

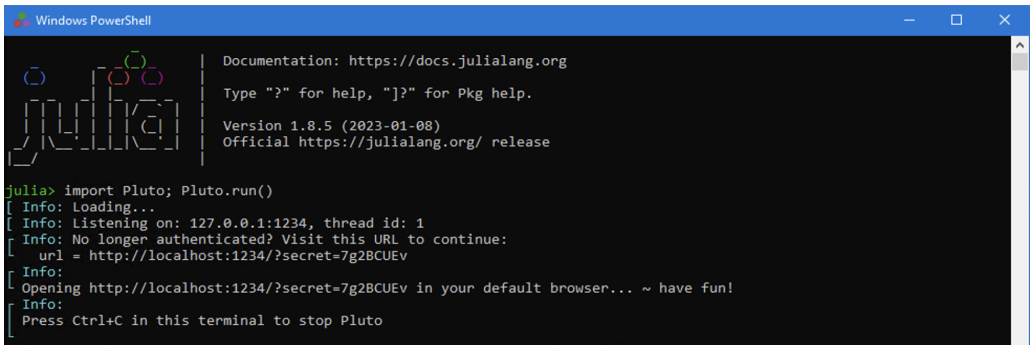

Рис. 5.1: Julia. Запуск Pluto

2. Первым делом подкючим пакеты “Plots” [5] и “DifferentialEquations” [6]. Далее объявим начальные данные верные для всех кейсов при помощи констант. Также объявим начальное условие для системы ДУ. (рис. 5.2)

```
# подключение пакетов
using Plots
using DifferentialEquations

# входные данные
const M1_0 = 6.8 * 1e6;
const M2_0 = 6 * 1e6;
```

```
const p_crit = 35 * 1e3;
const N = 31 * 1e3;
const q = 1;
const p1 = 11.5 * 1e3;
const tau1 = 18;
const p2 = 8.7 * 1e3;
const tau2 = 23;

const a1 = p_crit / (tau1 ^ 2 * p1^2 * N * q);
const a2 = p_crit / (tau2 ^ 2 * p2^2 * N * q);
const b = p_crit / (tau1 ^ 2 * p1^2 * tau2 ^ 2 * p2^2 * N * q);
const c1 = (p_crit - p1) / (tau1 * p1);
const c2 = (p_crit - p2) / (tau2 * p2);

u0 = [M1_0, M2_0]
t = (0, 150 * c1)
```

Рис. 5.2: Julia. Скрипт (1). Конкуренция двух фирм.

3. В следующей ячейке Pluto построим модель. При помощи ‘DifferentialEquations’ зададим и решим систему ДУ, после чего построим график решения и сохраним его. Далее запустим итоговый скрипт. После чего изменим коэффициент и также запустим скрипт. (рис. 5.3, 5.4, 5.5)

```
case = "1"; k = 0;
# case = "2"; k = 0.00067;

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

prob = ODEProblem(VS!, u0, t)
```

```
sol = solve(prob)

plt = plot(
  sol,
  dpi=500,
  size=(1024, 512),
  plot_title="Модель конкуренции двух фирм. Случай " * case,
  xlabel="t / c1",
  ylabel="M1(t), M2(t)",
  label=["M1 - оборотные средства предприятия #1" "M2 - оборотные средства предпр

savefig(plt, "artifacts/JL.lab08-0" * case * ".png")
println("Success")
```

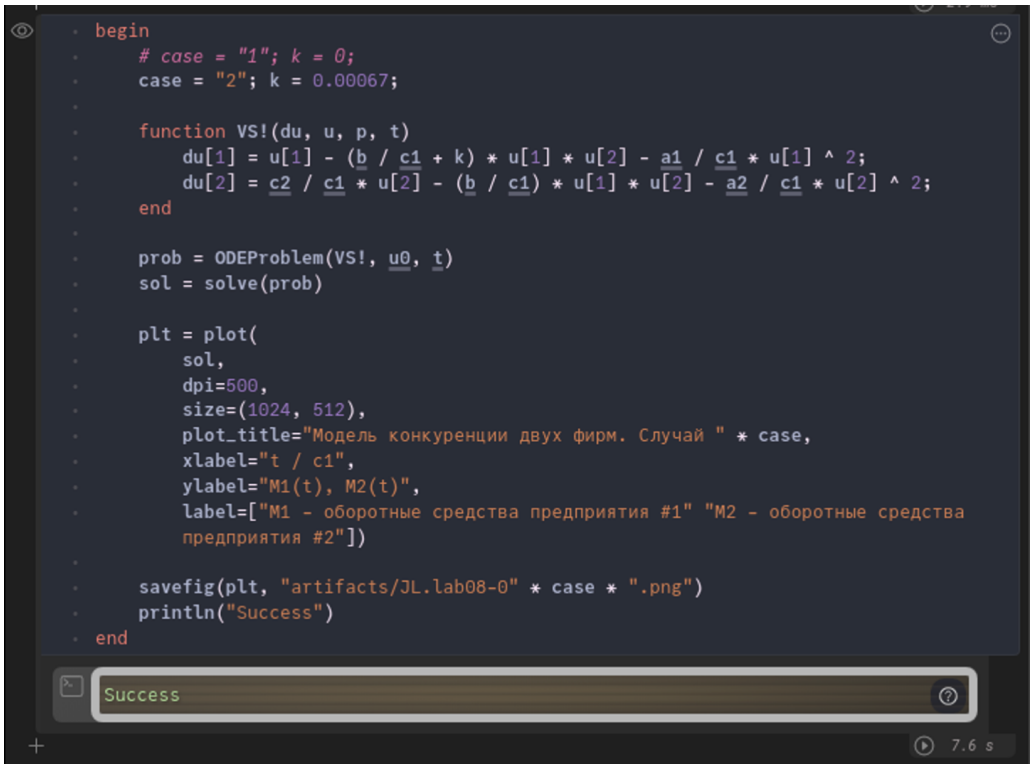

Рис. 5.3: Julia. Скрипт (2). Конкуренция двух фирм.

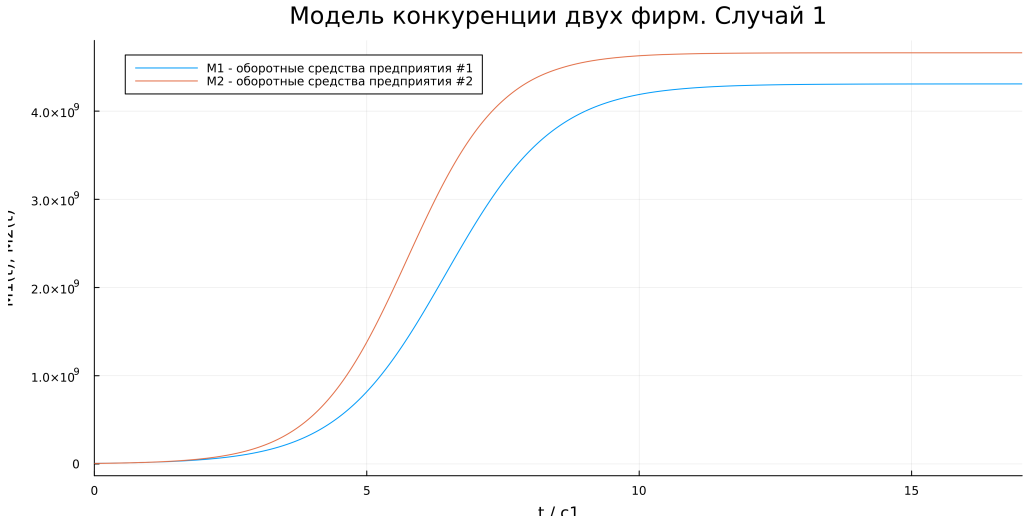

Рис. 5.4: Julia. Модель. Конкуренция двух фирм. Случай I

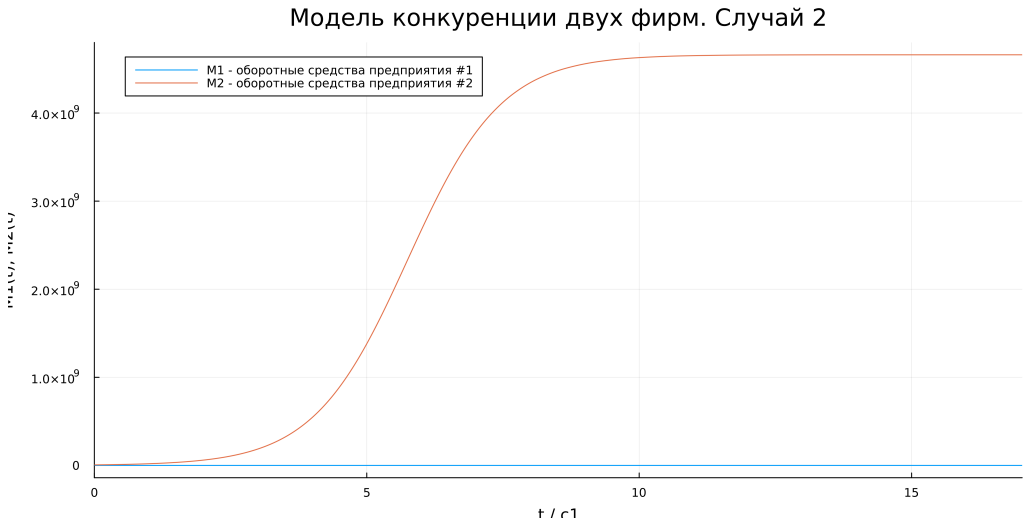

Рис. 5.5: Julia. Модель. Конкуренция двух фирм. Случай II

7. Напишем скрипт на modellica для решения 1-ой задачи. После чего запустим его и сохраним график. (рис. 5.6, 5.7)

```
model lab08_1
  constant Real M1_0 = 6.8 * 1e6;
  constant Real M2_0 = 6 * 1e6;
```

```
  constant Integer p_crit = 35 * integer(1e3);
  constant Integer N = 31 * integer(1e3);
  constant Integer q = 1;
  constant Real p1 = 11.5 * integer(1e3);
  constant Integer tau1 = 18;
  constant Real p2 = 8.7 * integer(1e3);
  constant Integer tau2 = 23;

  constant Real a1 = p_crit / (tau1 ^ 2 * p1^2 * N * q);
  constant Real a2 = p_crit / (tau2 ^ 2 * p2^2 * N * q);
  constant Real b = p_crit / (tau1 ^ 2 * p1^2 * tau2 ^ 2 * p2^2 * N * q);
  constant Real c1 = (p_crit - p1) / (tau1 * p1);
  constant Real c2 = (p_crit - p2) / (tau2 * p2);

  Real t = time / c1;
  Real M1(t);
  Real M2(t);
initial equation
    M1 = M1_0;
    M2 = M2_0;
equation
  der(M1) = M1 - (b / c1) * M1 * M2 - a1 / c1 * M1 ^ 2;
  der(M2) = c2 / c1 * M2 - (b / c1) * M1 * M2 - a2 / c1 * M2 ^ 2;
  annotation(experiment(StartTime = 0, StopTime = 15, Interval = 0.01));
end lab08_1;
```

```
model lab08_1
  constant Real M1_0 = 6.8 * 1e6;
  constant Real M2_0 = 6 * 1e6;

  constant Integer p_crit = 35 * integer(1e3);
  constant Integer N = 31 * integer(1e3);
  constant Integer q = 1;
  constant Real p1 = 11.5 * integer(1e3);
  constant Integer tau1 = 18;
  constant Real p2 = 8.7 * integer(1e3);
  constant Integer tau2 = 23;

  constant Real a1 = p_crit / (tau1 ^ 2 * p1^2 * N * q);
  constant Real a2 = p_crit / (tau2 ^ 2 * p2^2 * N * q);
  constant Real b = p_crit / (tau1 ^ 2 * p1^2 * tau2 ^ 2 * p2^2 * N * q);
  constant Real c1 = (p_crit - p1) / (tau1 * p1);
  constant Real c2 = (p_crit - p2) / (tau2 * p2);

  Real t = time / c1;
  Real M1(t);
  Real M2(t);
initial equation
    M1 = M1_0;
    M2 = M2_0;
equation
  der(M1) = M1 - (b / c1) * M1 * M2 - a1 / c1 * M1 ^ 2;
  der(M2) = c2 / c1 * M2 - (b / c1) * M1 * M2 - a2 / c1 * M2 ^ 2;
  annotation(experiment(StartTime = 0, StopTime = 15, Interval = 0.01));
end lab08_1;
```

Рис. 5.6: Modelica. Скрипт. Конкуренция двух фирм. Случай I

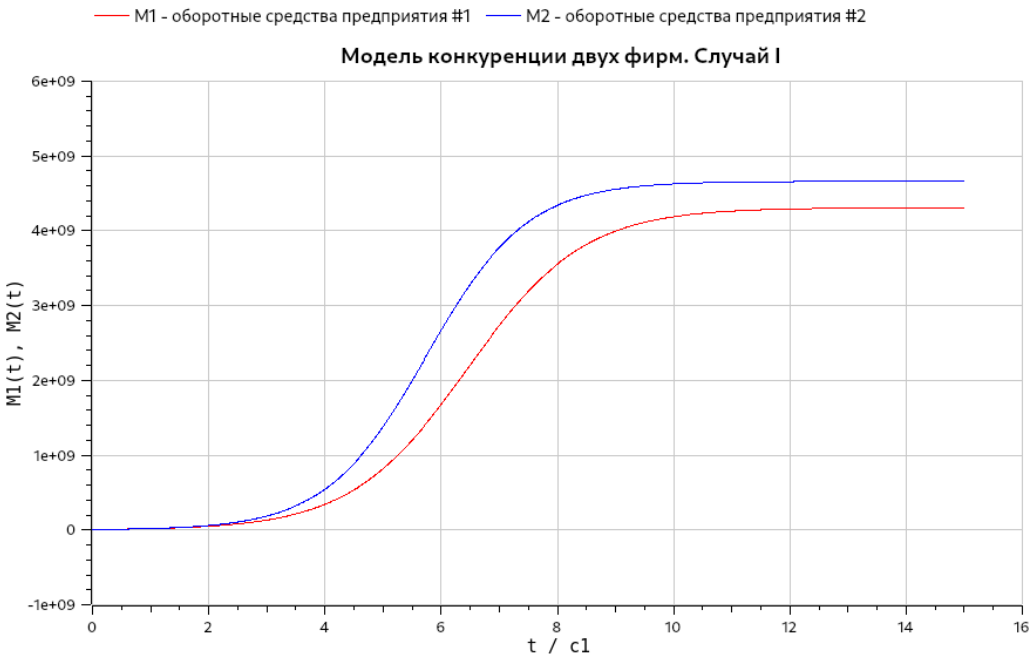

Рис. 5.7: Modelica. Модель. Конкуренция двух фирм. Случай I

8. Напишем скрипт на modellica для решения 2-ой задачи: изменим ДУ. После

чего запустим его и сохраним график. (рис. 5.8, 5.9)

```
model lab08_2
  constant Real M1_0 = 6.8 * 1e6;
  constant Real M2_0 = 6 * 1e6;

  constant Integer p_crit = 35 * integer(1e3);
  constant Integer N = 31 * integer(1e3);
  constant Integer q = 1;
  constant Real p1 = 11.5 * integer(1e3);
  constant Integer tau1 = 18;
  constant Real p2 = 8.7 * integer(1e3);
  constant Integer tau2 = 23;

  constant Real a1 = p_crit / (tau1 ^ 2 * p1^2 * N * q);
  constant Real a2 = p_crit / (tau2 ^ 2 * p2^2 * N * q);
  constant Real b = p_crit / (tau1 ^ 2 * p1^2 * tau2 ^ 2 * p2^2 * N * q);
  constant Real c1 = (p_crit - p1) / (tau1 * p1);
  constant Real c2 = (p_crit - p2) / (tau2 * p2);

  Real t = time / c1;
  Real M1(t);
  Real M2(t);
initial equation
    M1 = M1_0;
    M2 = M2_0;
equation
  der(M1) = M1 - (b / c1 + 0.00067) * M1 * M2 - a1 / c1 * M1 ^ 2;
  der(M2) = c2 / c1 * M2 - (b / c1) * M1 * M2 - a2 / c1 * M2 ^ 2;
  annotation(experiment(StartTime = 0, StopTime = 15, Interval = 0.01));
```

```
end lab08_2;
```

```
model lab08_2
  constant Real M1_0 = 6.8 * 1e6;
  constant Real M2_0 = 6 * 1e6;

  constant Integer p_crit = 35 * integer(1e3);
  constant Integer N = 31 * integer(1e3);
  constant Integer q = 1;
  constant Real p1 = 11.5 * integer(1e3);
  constant Integer tau1 = 18;
  constant Real p2 = 8.7 * integer(1e3);
  constant Integer tau2 = 23;

  constant Real a1 = p_crit / (tau1 ^ 2 * p1^2 * N * q);
  constant Real a2 = p_crit / (tau2 ^ 2 * p2^2 * N * q);
  constant Real b = p_crit / (tau1 ^ 2 * p1^2 * tau2 ^ 2 * p2^2 * N * q);
  constant Real c1 = (p_crit - p1) / (tau1 * p1);
  constant Real c2 = (p_crit - p2) / (tau2 * p2);

  Real t = time / c1;
  Real M1(t);
  Real M2(t);
initial equation
    M1 = M1_0;
    M2 = M2_0;
equation
  der(M1) = M1 - (b / c1 + 0.00067) * M1 * M2 - a1 / c1 * M1 ^ 2;
  der(M2) = c2 / c1 * M2 - (b / c1) * M1 * M2 - a2 / c1 * M2 ^ 2;
  annotation(experiment(StartTime = 0, StopTime = 15, Interval = 0.01));
end lab08_2;
```

Рис. 5.8: Modelica. Скрипт. Конкуренция двух фирм. Случай II

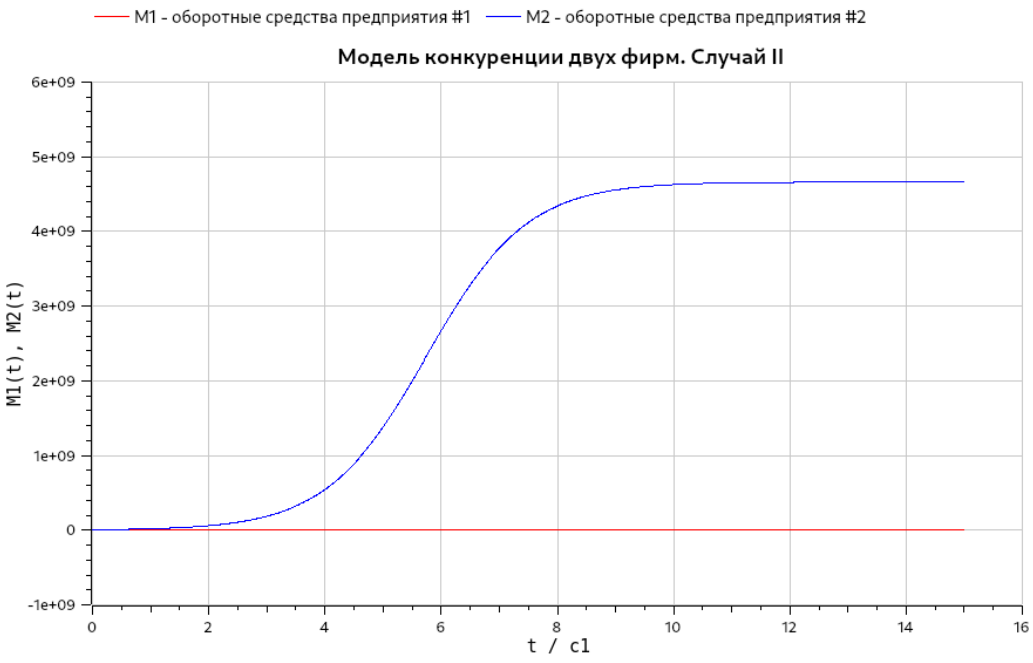

Рис. 5.9: Modelica. Модель. Конкуренция двух фирм. Случай II

6 Анализ результатов

Работа выполненна без непредвиденных проблем в соответствии с руководством. Ошибок и сбоев не произошло.

Моделирование на OMEdit было проще и быстрее, чем при использовании средств Julia. Скрипт на Modelica вышел более лакончиным, понятным и коротким. Более того OpenModelica быстрее обрабатывала скрипт и симмулировала модель. Стоит отметить, что OpenModelica имеет множество разлиных полезных инструментов для настройки с симмуляцией и работой с ней.

К плюсам Julia можно отнести, что она является языком программирования, который хорошо подходит для математических и технических задач. Отметим, что скрипт на Julia выполняется долго из-за подключения пакетов, каждый раз при его запуске. При использовании Pluto, нет необходимости каждый раз с нуля выполнять скрипт, таким образом скорость выполнения может даже превышать скорость моделирования в OMEdit.

7 Выводы

Мы улучшили практические навыки в области дифференциальных уравнений, улучшили навыки моделирования на Julia, а также навыки моделирования на OpenModelica. Изучили и построили модель конкуренции двух фирм.

Список литературы

1. Julia [Электронный ресурс]. URL: http://www.unn.ru/books/met_files/JULIA_tutorial.pdf.
2. OpenModelica [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/OpenModelica.
3. Эффективность рекламы [Электронный ресурс]. RUDN. URL: https://esystem.rudn.ru/mod/resource/view.php?id=967257.
4. Pluto [Электронный ресурс]. URL: https://plutojl.org/.
5. Plots in Julia [Электронный ресурс]. URL: https://docs.juliaplots.org/latest/tutorial/.
6. Differential Equations in Julia [Электронный ресурс]. URL: https://docs.sciml.ai/DiffEqDocs/stable/getting_started/.