

Презентация к лабораторной работе №8

Боровикова Карина Владимировна

2023, 1 апреля

RUDN University, Moscow, Russian Federation

- Важность умения работы с языками Julia и OpenModelica в части математического моделирования

- Язык Julia
- Язык OpenModelica
- Моделирование задачи конкурирования двух фирм

Цель работы:

Построить модель конкурирования дввух фирм с помощью языков Julia и OpenModelica для двух случаев

Задачи:

- Построить графики изменения оборотных средств фирмы 1 и фирмы 2 без учета постоянных издержек и с введенной нормировкой для случая 1.
- Построить графики изменения оборотных средств фирмы 1 и фирмы 2 без учета постоянных издержек и с введенной нормировкой для случая 2.

1. Рассматриваем задачу о конкурировании двух фирм (рис. 1).

Вариант 18

Случай 1. 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:

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

где $a_1 = \frac{p_{cr}}{\tau_1^2 \tilde{p}_1^2 Nq}$, $a_2 = \frac{p_{cr}}{\tau_2^2 \tilde{p}_2^2 Nq}$, $b = \frac{p_{cr}}{\tau_1^2 \tilde{p}_1^2 \tau_2^2 \tilde{p}_2^2 Nq}$, $c_1 = \frac{p_{cr} - \tilde{p}_1}{\tau_1 \tilde{p}_1}$, $c_2 = \frac{p_{cr} - \tilde{p}_2}{\tau_2 \tilde{p}_2}$.

Также введена нормировка $t = c_1 \theta$.

Рис. 1: Моделирование конкурирования двух фирм

2. Рассматриваем задачу о конкурировании двух фирм (рис. 2).

Случай 2. 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_1M_2 будет отличаться. Пусть в рамках рассматриваемой модели динамика изменения объемов продаж фирмы 1 и фирмы 2 описывается следующей системой уравнений:

$$\begin{aligned}\frac{dM_1}{d\theta} &= M_1 - \left(\frac{b}{c_1} + 0,0009 \right) M_1 M_2 - \frac{a_1}{c_1} M_1^2 \\ \frac{dM_2}{d\theta} &= \frac{c_2}{c_1} M_2 - \frac{b}{c_1} M_1 M_2 - \frac{a_2}{c_1} M_2^2\end{aligned}$$

Для обоих случаев рассмотрим задачу со следующими начальными условиями и

$$M_0^1 = 4.2, M_0^2 = 3.8,$$

параметрами: $p_{cr} = 11.4, N = 26, q = 1$

$$\tau_1 = 14, \tau_2 = 22,$$

$$\bar{p}_1 = 6.6, \bar{p}_2 = 4.5$$

Рис. 2: Моделирование конкурирования двух фирм

3. Программируем первый случай на Julia (рис. 3).

```
begin
using Plots
using DifferentialEquations

"Параметры: средняя производительность для 1 фирмы"
const M10 = 4.2 * 1000000
"Параметры: средняя производительность для 1 фирмы"
const M20 = 3.8 * 1000000
"Критическая стоимость продукта"
const pcr = 11.8 * 1000
"Начисл. отчислений: произвольного продукта"
const N = 20 * 1000
"Максимальная потребность одного человека в продукте в единицу времени"
const q = 1

"Производительность произвольного цикла фирмы"
const p1 = 14
"Производительность произвольного цикла фирмы"
const p2 = 22

"Собственная стоимость товаров в первой фирме"
const p1 = 6.6 * 1000
"Собственная стоимость товаров во второй фирме"
const p2 = 4.5 * 1000

const a1 = pcr / (q1^2 * p2^2 * N * q)
const a2 = pcr / (q2^2 * p2^2 * N * q)
const b = pcr / (q1^2 * p1^2 * q2^2 * p2^2 * N * q)
const c1 = (pcr - p1) / (q1 * p1)
const c2 = (pcr - p2) / (q2 * p2)

"Начальные условия: u[1] - P1, u[2] - P2"
u0 = [M10, M20]

"Период времени"
T = (0.0, 1*360)

function F!(du, u, p, t)
    du[1] = a[1] - (b/c1) * 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
```

Рис. 3: Первый случай на Julia

3. Программируем первый случай на OpenModelica (рис. 4).

```
model L81
  constant Real M10 = 4.2 * 1000000;
  constant Real M20 = 3.8 * 1000000;

  constant Integer pcr = 11.4 * 1000;
  constant Integer N = 26 * 1000;
  constant Integer q = 1;

  constant Integer tau1 = 14;
  constant Integer tau2 = 22;

  constant Integer p1 = 6.6 * 1000;
  constant Integer p2 = 4.5 * 1000;

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

  Real t = time / c1;
  Real M1(t);
  Real M2(t);

  initial equation
    M1 = M10;
    M2 = M20;
  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=16, Interval=0.2));
end L81;
```

Рис. 4: Первый случай на OpenModelica

4. Программируем второй случай на Julia (рис. 5).

```
begin
using Plots
using DifferentialEquations

"Оборудование" "продукты" "принадлежит" "дл" "верно"
const m10 = 5.2 * 1000000
"Оборудование" "продукты" "принадлежит" "дл" "верно"
const m20 = 2.8 * 1000000
"Принадлежит" "продукты" "продукты"
const pcr = 11.4 * 1000
"число" "потребителей" "принадлежит" "продукты"
const n = 20 * 1000
"численность" "потребителей" "единицы" "числовая" "продукты" "единицы" "численность"
const q = 1

"численность" "принадлежит" "число" "верно" "1"
const c1 = 14
"численность" "принадлежит" "число" "верно" "2"
const c2 = 22

"эффективности" "показатель" "показатель" "верно"
const p1 = 6.2 * 1000
"эффективности" "показатель" "показатель" "верно"
const p2 = 4.5 * 1000

const a1 = pcr / (c1^2 * p1^2 + p2^2 * n * q)
const a2 = pcr / (c2^2 * p2^2 + n * q)
const b = pcr / (c1^2 * p1^2 * c2^2 + p2^2 * n * q)
const c1 = (pcr - p1) / (c1^2 * p1)
const c2 = (pcr - p2) / (c2^2 * p2)

"начальные" "условия" " $u(1) = u(2) = u(3) = u(4) = u(5)$ "
u0 = [m10, m20]

"функция" "производная"
f = (u, t) ->
function F(u, v, t)
    du[1] = a[1] - (b*u[1]*v[2]) * 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
```

Рис. 5: Второй случай на Julia

5. Программируем второй случай на OpenModelica (рис. 6).

```
model L82
  constant Real M10 = 4.2 * 1000000;
  constant Real M20 = 3.8 * 1000000;

  constant Integer pcr = 11.4 * 1000;
  constant Integer N = 26 * 1000;
  constant Integer q = 1;

  constant Integer tau1 = 14;
  constant Integer tau2 = 22;

  constant Integer p1 = 6.6 * 1000;
  constant Integer p2 = 4.5 * 1000;

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

  Real t = time / c1;
  Real M1(t);
  Real M2(t);

  initial equation
    M1 = M10;
    M2 = M20;
  equation
    der(M1) = M1 - ((b/c1)+0.0009) * M1 * M2 - (a1/c1) * M1^2;
    der(M2) = (c2/c1) * M2 - (b/c1) * M1 * M2 - (a2/c1) * M2^2;
    annotation(experiment(StartTime=0, StopTime=16, Interval=0.2));
end L82;
```

Рис. 6: Второй случай на OpenModelica

6. Запускаем код через терминал, получаем изображения для случаев на Julia (рис. 7).

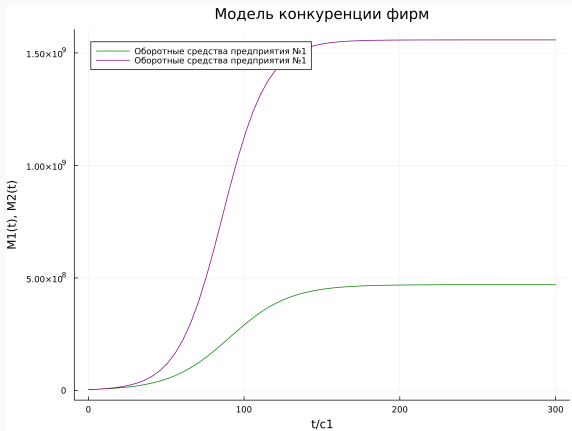


Рис. 7: Результаты выполнения кодов на Julia

7. Запускаем код через терминал, получаем изображения случаев на Julia (рис. 8).

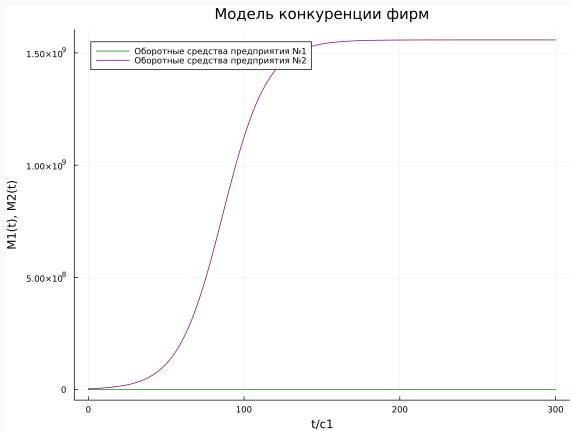


Рис. 8: Результаты выполнения кодов на Julia

9. Запускаем код через терминал, получаем изображения для случаев на OpenModelica (рис. 11).

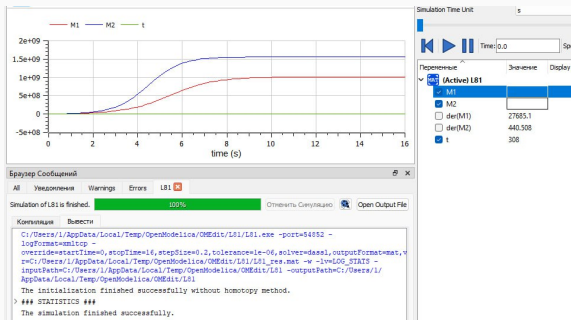


Рис. 9: Результаты выполнения кодов на Julia

9. Запускаем код через терминал, получаем изображения для случаев на OpenModelica (рис. 12).

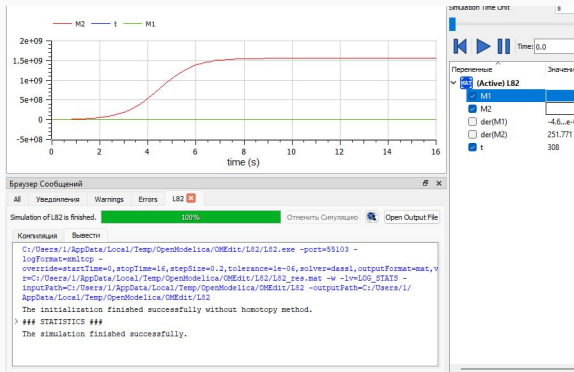


Рис. 10: Результаты выполнения кодов на Julia

Результаты

В ходе выполнения лабораторной работы мне удалось построить модель конкуренции двух фирм с помощью языков Julia и OpenModelica в двух случаях