

# Лабораторная работа №3

## Модель боевых действий

Кузнецов Юрий Владимирович

### Содержание

Цель работы .....	1
Задачи.....	1
Термины .....	2
Теоретическое введение .....	2
Ход работы .....	3
Вывод.....	12
Ресурсы.....	13

### Цель работы

Рассмотреть простейшие модели боевых действий, а именно модели Ланчестера. Смоделировать боевые действия средствами OpenModelica и Julia.

### Задачи

Между страной  $X$  и страной  $Y$  идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями  $x(t)$  и  $y(t)$ .

Необходимо построить:

1. Модель боевых действий между регулярными войсками;

$$\begin{cases} \frac{dx}{dt} = -0.312x(t) - 0.456y(t) + \sin(t + 3) \\ \frac{dy}{dt} = -0.256x(t) - 0.340y(t) + \cos(t + 7) \end{cases}$$

2. Модель ведение боевых действий с участием регулярных войск и партизанских отрядов.

$$\begin{cases} \frac{dx}{dt} = -0.318x(t) - 0.615y(t) + |\cos(8t)| \\ \frac{dy}{dt} = -0.312x(t)y(t) - 0.512y(t) + |\sin(6t)| \end{cases}$$

## Термины

- 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. [@unn-julia]
- OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. [@wiki-om]

## Теоретическое введение

Рассмотрим некоторые простейшие модели боевых действий – модели Ланчестера. В противоборстве могут принимать участие как регулярные войска, так и партизанские отряды. В общем случае главной характеристикой соперников являются численности сторон. Если в какой-то момент времени одна из численностей обращается в нуль, то данная сторона считается проигравшей (при условии, что численность другой стороны в данный момент положительна). [@rudn-task]

Рассмотри два случая ведения боевых действий:

1. Боевые действия между регулярными войсками;
2. Боевые действия с участием регулярных войск и партизанских отрядов;

В первом случае численность регулярных войск определяется тремя факторами:

- скорость уменьшения численности войск из-за причин, не связанных с боевыми действиями (болезни, травмы, дезертирство);
- скорость потерь, обусловленных боевыми действиями противоборствующих сторон (что связано с качеством стратегии, уровнем вооружения, профессионализмом солдат и т.п.);
- скорость поступления подкрепления (задаётся некоторой функцией от времени).

В этом случае модель боевых действий между регулярными войсками описывается следующим образом

$$\begin{cases} \frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t) \\ \frac{dy}{dt} = -c(t)x(t) - h(t)y(t) + Q(t) \end{cases}, \text{ где}$$

$-a(t)x(t)$  и  $-h(t)y(t)$  - потери, не связанные с боевыми действиями,

$-b(t)y(t)$  и  $-c(t)x(t)$  - отражают потери на поле боя,

$b(t)$  и  $c(t)$  - эффективность боевых действий со стороны  $y$  и  $x$  соответственно,

$a(t)$  и  $h(t)$  - величины, характеризующие степень влияния различных факторов на потери,

$P(t)$  и  $Q(t)$  - функции, которые учитывают возможность подхода подкрепления к войскам  $X$  и  $Y$  в течение одного дня.

Во втором случае в борьбу добавляются партизанские отряды. Нерегулярные войска в отличие от постоянной армии менее уязвимы, так как действуют скрытно, в этом случае сопернику приходится действовать неизбирательно, по площадям, занимаемым партизанами. Поэтому считается, что тем потерь партизан, проводящих свои операции в разных местах на некоторой известной территории, пропорционален не только численности армейских соединений, но и численности самих партизан. В результате модель принимает вид:

$$\begin{cases} \frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t) \\ \frac{dy}{dt} = -c(t)x(t)y(t) - h(t)y(t) + Q(t) \end{cases}, \text{ где}$$

## Ход работы

Установим пакет DifferentialEquations в Julia необходимый для работы с дифференциальными уравнениями.



```

using Plots
using DifferentialEquations

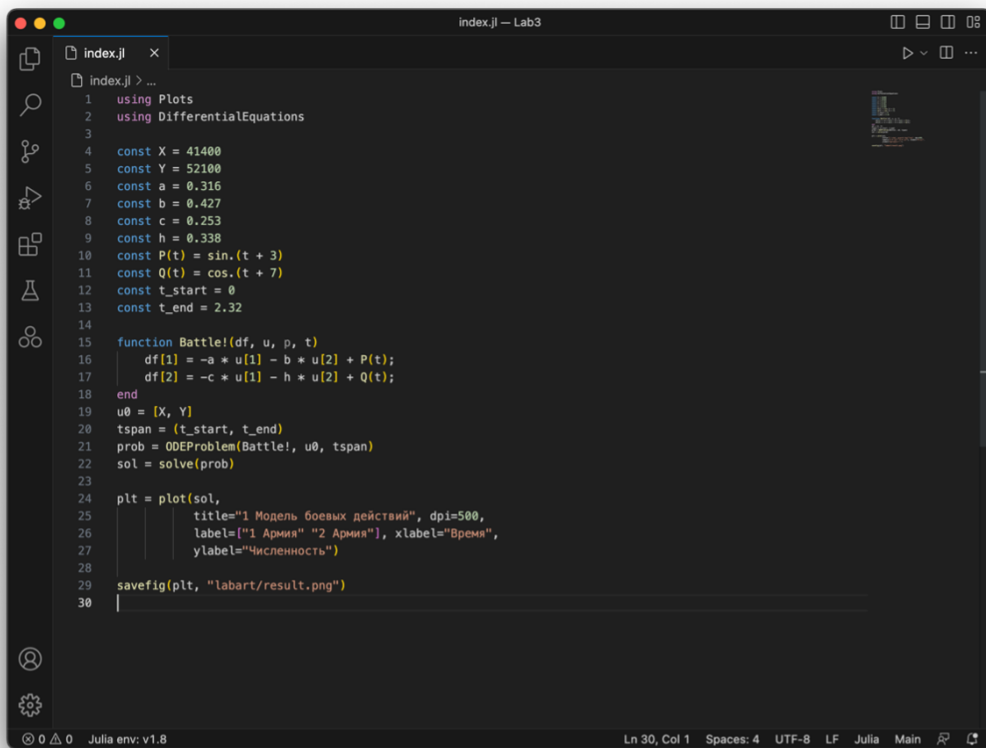
const X = 41400
const Y = 52100
const a = 0.316
const b = 0.427
const c = 0.253
const h = 0.338
const P(t) = sin.(t + 3)
const Q(t) = cos.(t + 7)
const t_start = 0
const t_end = 2.32

function Battle!(df, u, p, t)
    df[1] = -a * u[1] - b * u[2] + P(t);
    df[2] = -c * u[1] - h * u[2] + Q(t);
end
u0 = [X, Y]
tspan = (t_start, t_end)
prob = ODEProblem(Battle!, u0, tspan)
sol = solve(prob)

plt = plot(sol,
            title="1 Модель боевых действий", dpi=500,
            label=["1 Армия" "2 Армия"], xlabel="Время",
            ylabel="Численность")

savefig(plt, "labart/result.png")

```

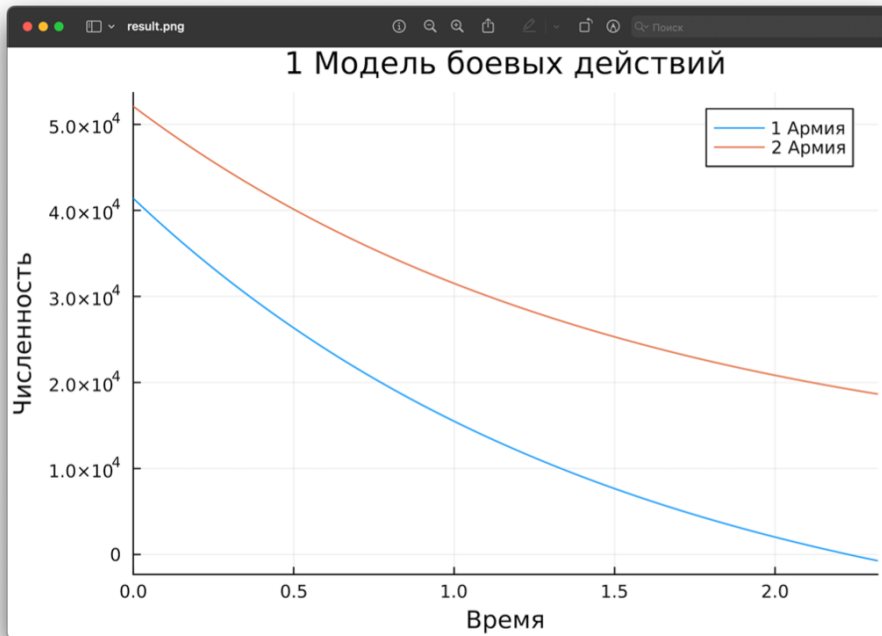


```
index.jl x
index.jl > ...
1 using Plots
2 using DifferentialEquations
3
4 const X = 41400
5 const Y = 52100
6 const a = 0.316
7 const b = 0.427
8 const c = 0.253
9 const h = 0.338
10 const P(t) = sin.(t + 3)
11 const Q(t) = cos.(t + 7)
12 const t_start = 0
13 const t_end = 2.32
14
15 function Battle!(df, u, p, t)
16     df[1] = -a * u[1] - b * u[2] + P(t);
17     df[2] = -c * u[1] - h * u[2] + Q(t);
18 end
19 u0 = [X, Y]
20 tspan = (t_start, t_end)
21 prob = ODEProblem(Battle!, u0, tspan)
22 sol = solve(prob)
23
24 plt = plot(sol,
25           title="1 Модель боевых действий", dpi=500,
26           label=["1 Армия" "2 Армия"], xlabel="Время",
27           ylabel="Численность")
28
29 savefig(plt, "labart/result.png")
30 |
```

© 0 0 0 Julia env: v1.8 Ln 30, Col 1 Spaces: 4 UTF-8 LF Julia Main

*Julia. Скрипт для моделирования боевых действий между регулярными войсками*

После запуска кода – получаем визуализацию 1 модели боевых действий.



*Julia. Модель боевых действий между регулярными войсками*

Изменяем код, чтобы построить скрипт для визуализации. Изменяем начальные данные и функцию, описывающую систему ОДУ.

Код:

```
using Plots
using DifferentialEquations

const X = 44200
const Y = 54200
const a = 0.318
const b = 0.615
const c = 0.312
const h = 0.512
const P(t) = abs.(cos.(8 * t))
const Q(t) = abs.(sin.(6 * t))
const t_start = 0
const t_end = 0.001

function Battle!(df, u, p, t)
    df[1] = -a * u[1] - b * u[2] + P(t);
    df[2] = -c * u[1] * u[2] - h * u[2] + Q(t);
end
u0 = [X, Y]
tspan = (t_start, t_end)
prob = ODEProblem(Battle!, u0, tspan)
```

```

sol = solve(prob)

plt = plot(sol,
            title="Модель боевых действий №2",
            dpi=500,
            label=["1 Армия" "2 Армия"],
            xlabel="Время",
            ylabel="Численность",
            legend=:outertop)
savefig(plt, "labart/result2.png")

```

```

index.jl x
index.jl > ...
1 using Plots
2 using DifferentialEquations
3
4 const X = 44200
5 const Y = 54200
6 const a = 0.318
7 const b = 0.615
8 const c = 0.312
9 const h = 0.512
10 const P(t) = abs.(cos.(8 * t))
11 const Q(t) = abs.(sin.(6 * t))
12 const t_start = 0
13 const t_end = 0.001
14
15 function Battle!(df, u, p, t)
16     df[1] = -a * u[1] - b * u[2] + P(t);
17     df[2] = -c * u[1] * u[2] - h * u[2] + Q(t);
18 end
19 u0 = [X, Y]
20 tspan = (t_start, t_end)
21 prob = ODEProblem(Battle!, u0, tspan)
22 sol = solve(prob)
23
24 plt = plot(sol,
25            title="Модель боевых действий №2",
26            dpi=500,
27            label=["1 Армия" "2 Армия"],
28            xlabel="Время",
29            ylabel="Численность",
30            legend=:outertop)
31 savefig(plt, "labart/result2.png")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

WARNING: redefinition of constant b. This may fail, cause incorrect answers, or produce other errors.  
 WARNING: redefinition of constant c. This may fail, cause incorrect answers, or produce other errors.  
 WARNING: redefinition of constant h. This may fail, cause incorrect answers, or produce other errors.  
 WARNING: redefinition of constant t\_end. This may fail, cause incorrect answers, or produce other errors.  
 "/Users/uvkuznetsov/Desktop/Матмод/Лаб3/labart/result2.png"

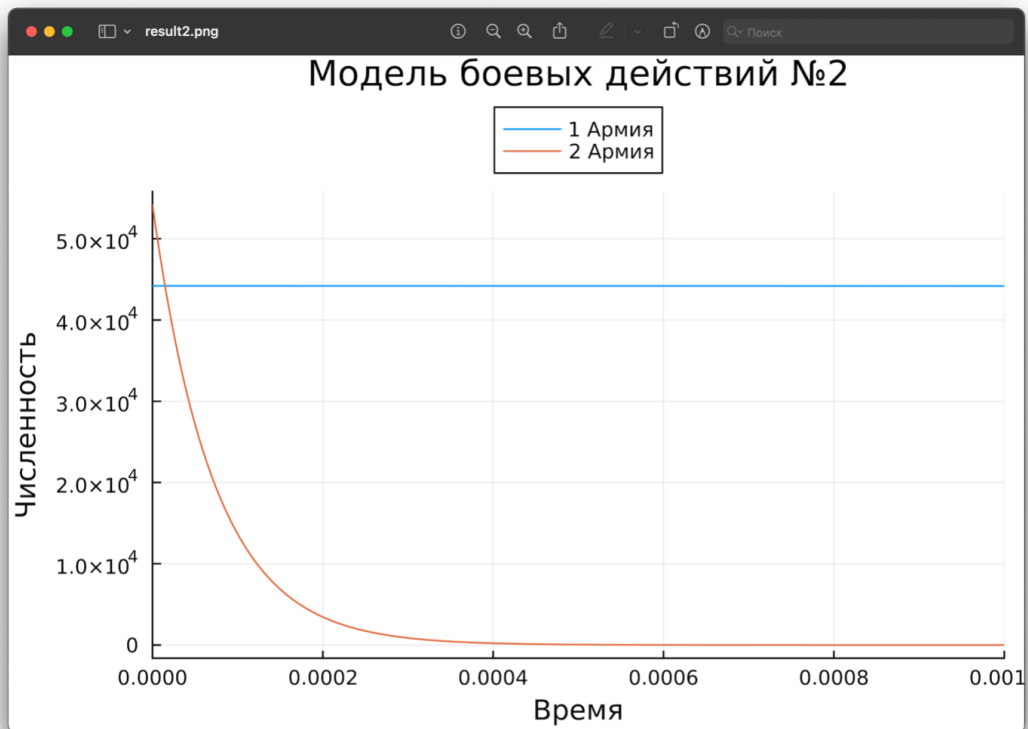
julia>

Julia env: v1.8 Ln 30, Col 29 Spaces: 4 UTF-8 LF Julia Main

*Julia. Скрипт для моделирования боевых действий с участием регулярных войск и партизанских отрядов*

После запуска получаем визуализации модели боевых действий с участием регулярных войск и партизанских отрядов.

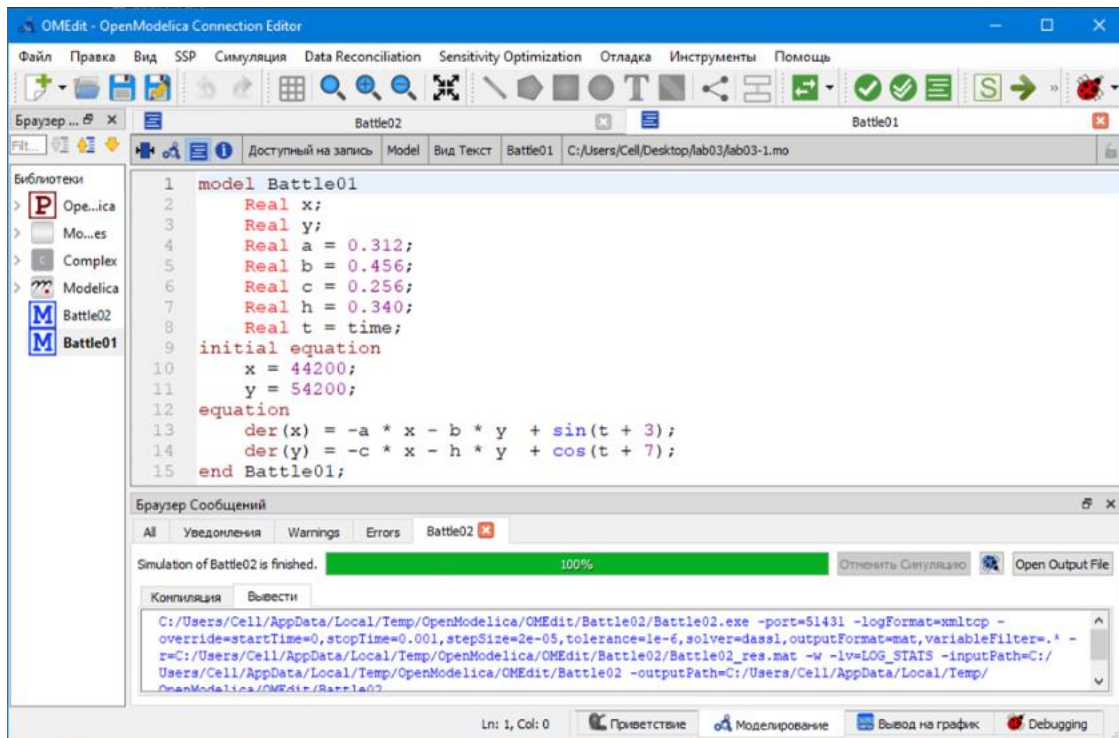




*Julia. Модель боевых действий с участием регулярных войск и партизанских отрядов*

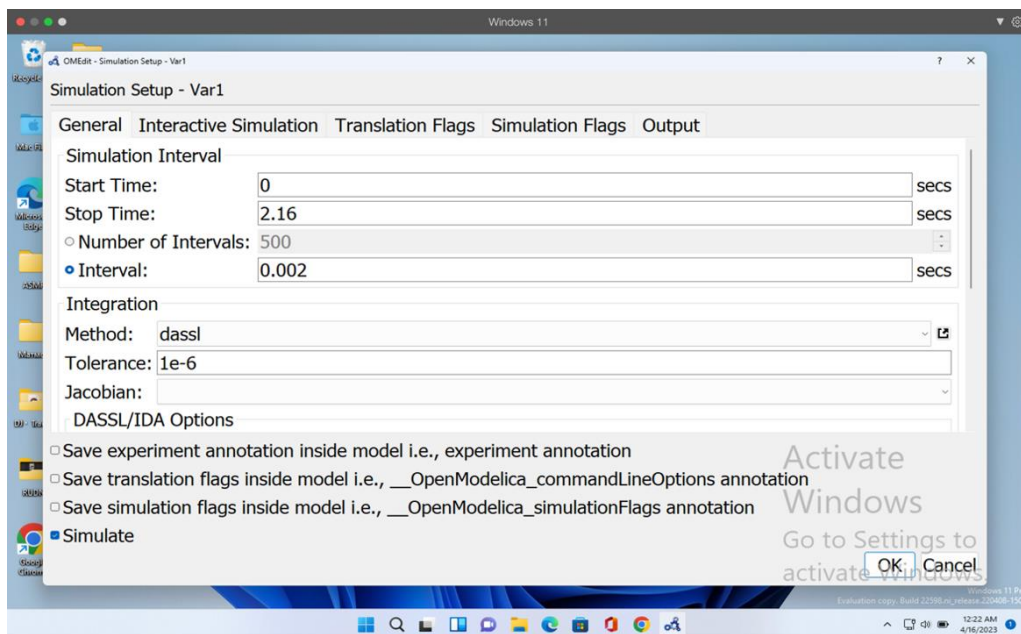
Пишем код для визуализации на OpenModelica. Для этого нам необходимо создать модель, внутри которой сначала объявим необходимые переменные, после чего проинициализируем начальные условия и в конце введем непосредственно уравнения для моделирования.

```
model Var1
  Real x;
  Real y;
  Real a = 0.312;
  Real b = 0.456;
  Real c = 0.256;
  Real h = 0.340;
  Real t = time;
initial equation
  x = 44200;
  y = 54200;
equation
  der(x) = -a * x - b * y + sin(t + 3);
  der(y) = -c * x - h * y + cos(t + 7);
end Var1;
```

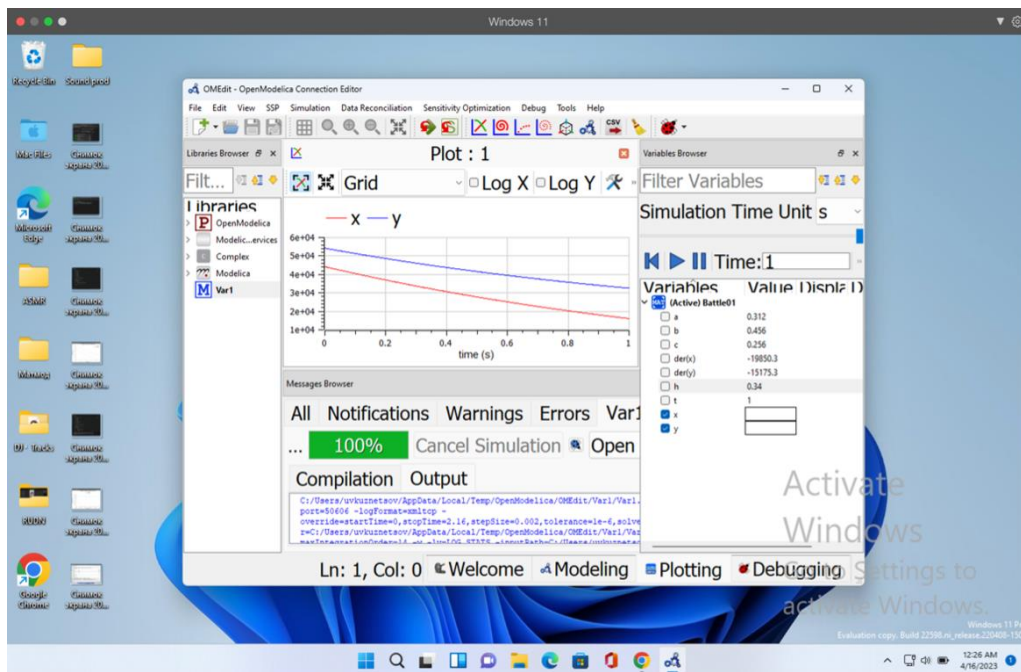


*Modelica. Скрипт для моделирования боевых действий между регулярными войсками*

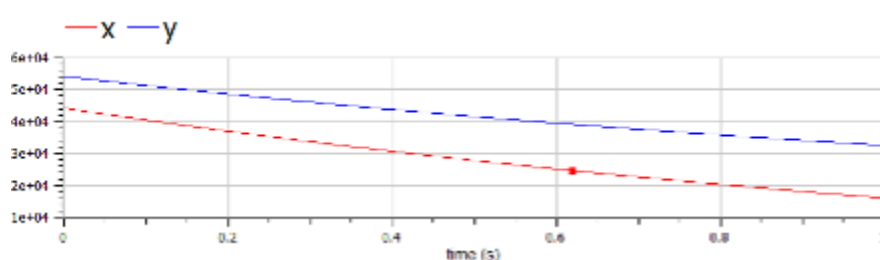
Запустим симуляцию, напомним легенду для получившейся модели и экспортируем итог.



*OMEdit. Настройка симуляции*



*OMEdit. Симуляция модели боевых действий между регулярными войсками*



*Modelica. Модель боевых действий между регулярными войсками*

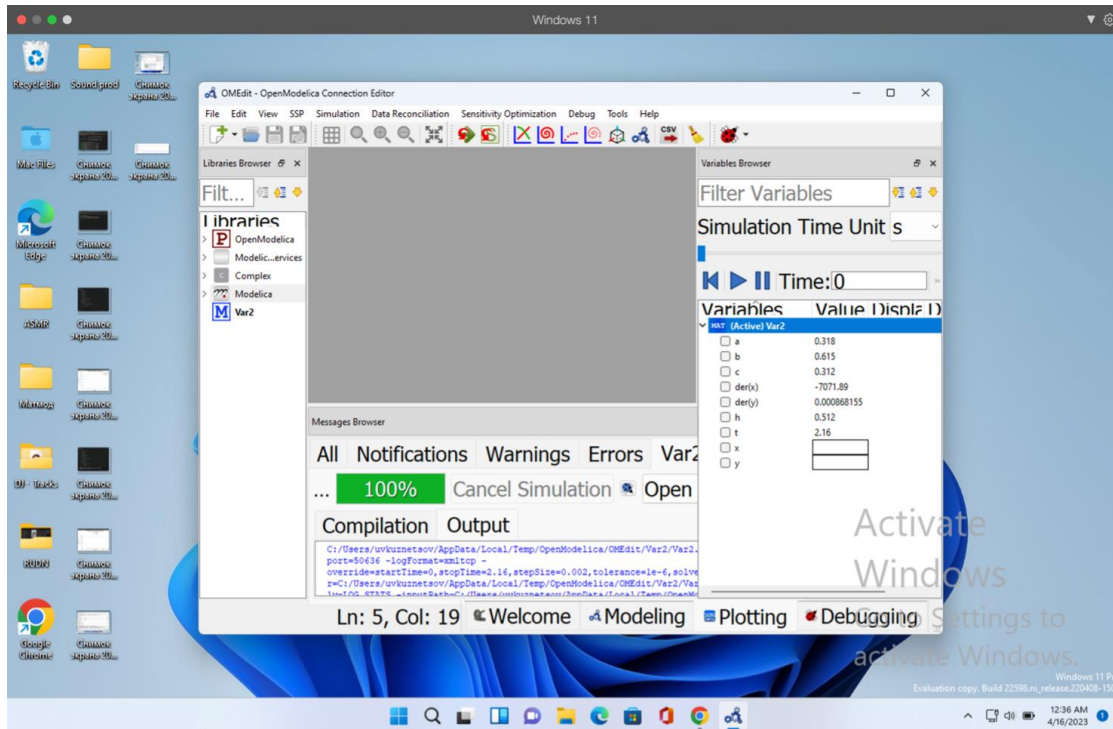
Приступим к моделированию боевых действий с участием регулярных войск и партизанских отрядов на OpenModelica. Для этого перепишем начальные данные и изменим уравнение, взяв за основу предыдущий скрипт.

```
model Var2
  Real x;
  Real y;
  Real a = 0.318;
  Real b = 0.615;
  Real c = 0.312;
  Real h = 0.512;
  Real t = time;
initial equation
  x = 44200;
  y = 54200;
equation
  der(x) = -a * x - b * y + abs(cos(8 * t));
```

```

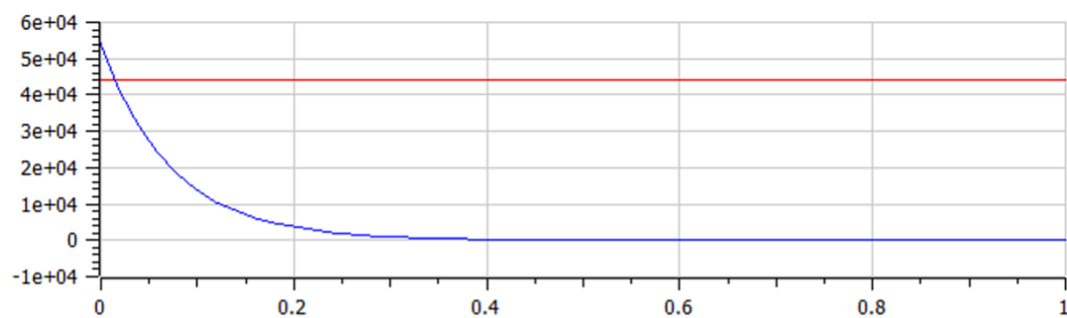
der(y) = -c * x * y - h * y + abs(sin(6 * t));
end Var2;

```



*Modelica. Скрипт для моделирования боевых действий с участием регулярных войск и партизанских отрядов*

Запустим симуляцию, пропишем легенду и экспортируем график.



*Modelica. Модель боевых действий с участием регулярных войск и партизанских отрядов*

## Вывод

Получили новые навыки в моделировании при помощи OpenModelica. Улучшили навыки в решении задач с ДУ.

## Ресурсы

- Julia. URL: [http://www.unn.ru/books/met\\_files/JULIA\\_tutorial.pdf](http://www.unn.ru/books/met_files/JULIA_tutorial.pdf).
- OpenModelica. URL: <https://ru.wikipedia.org/wiki/OpenModelica>.
- Модель боевых действий. RUDN. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=967237>.
- Plots in Julia URL: <https://docs.juliaplots.org/latest/tutorial/>.
- Differential Equations in Julia. URL: [https://docs.sciml.ai/DiffEqDocs/stable/getting\\_started/](https://docs.sciml.ai/DiffEqDocs/stable/getting_started/).