Лабораторная работа №3

Модель боевых действий

Кузнецов Юрий Владимирович

Содержание

# Цель работы

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

# Задачи

Между страной и страной идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями и .

Необходимо построить:

1. Модель боевых действий между регулярными войсками;
2. Модель ведение боевых действий с участием регулярных войск и партизанских отрядов.

# 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.
* OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica.

# 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 (приусловии, 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 связанно 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

и - потери, не связанные с боевыми действиями,

и - отражают потери на поле боя,

и - эффективность боевых действий со стороны у и х соответственно,

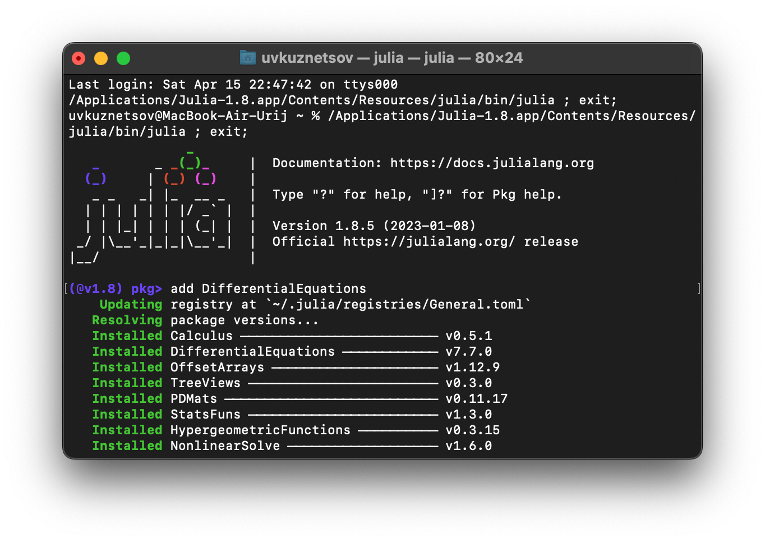
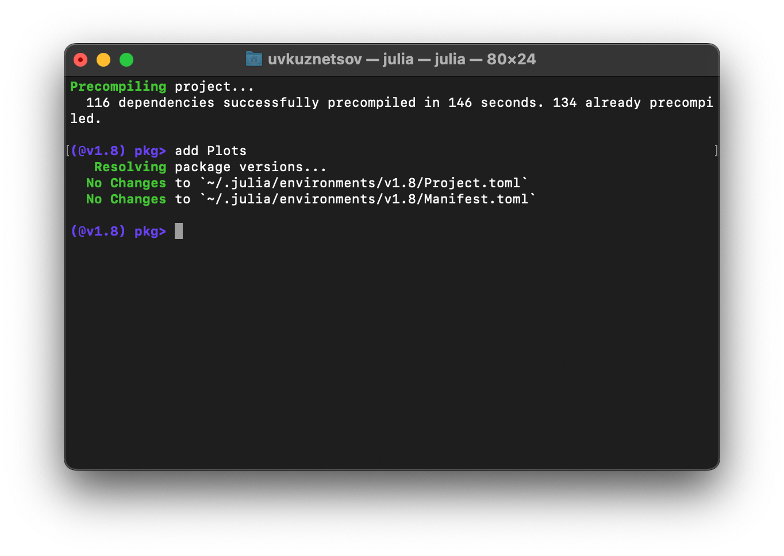
и - величины, характеризующие степень влияния различных факторов на потери,

и - функции, которые учитывают возможность подхода подкрепления к войскам Х и У в течение одного дня.

Во втором случае в борьбу добавляются партизанские отряды. Нерегулярные войска в отличии 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:

# Ход работы

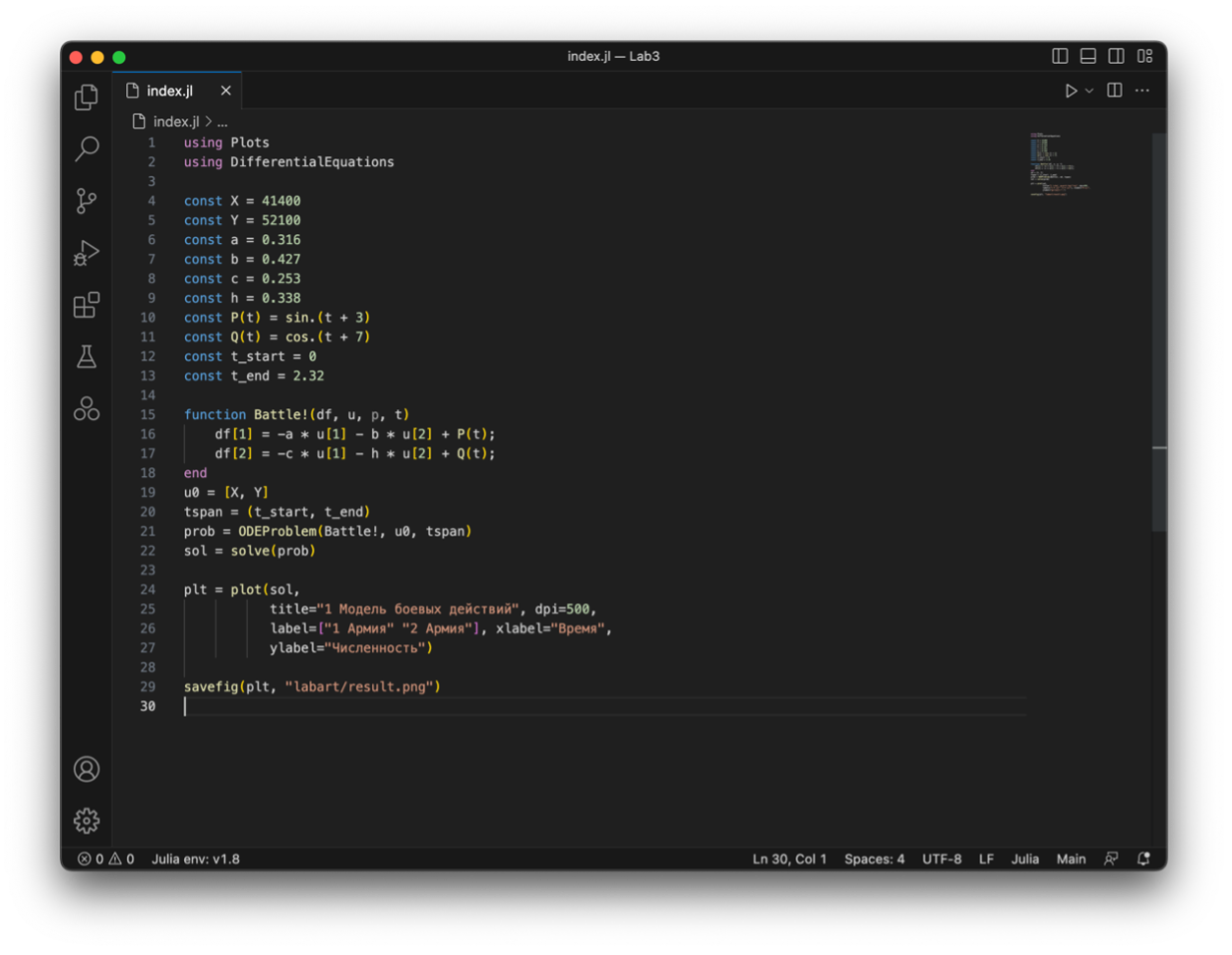
Установим пакет DifferentialEquations в Julia необходимый для работы с дифференциальными уравнениями.

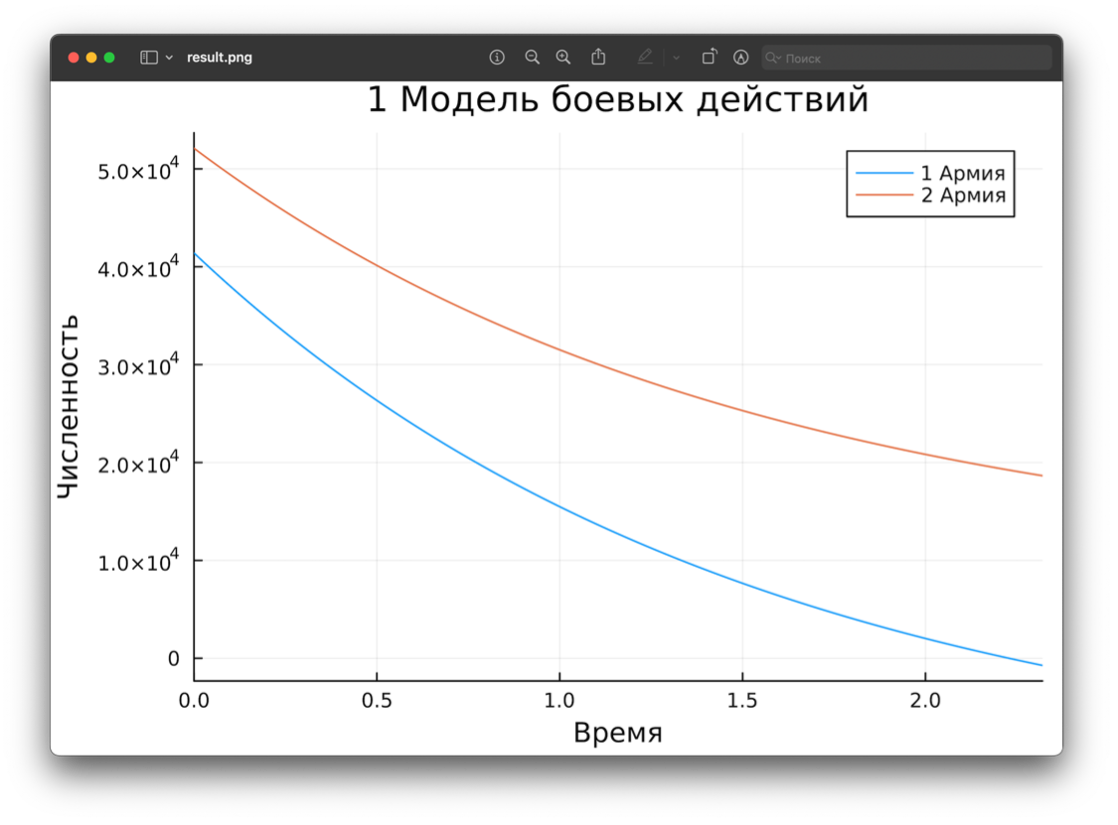
 

Напишем код для моделирования боевых действий между войсками. Подключаем Plots и DifferentialEquations , после чего объявим константы (начальные данные). Используя DifferentialEquations составим и решим систему однородных ДУ. Визуализируем модель.

Код:

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")

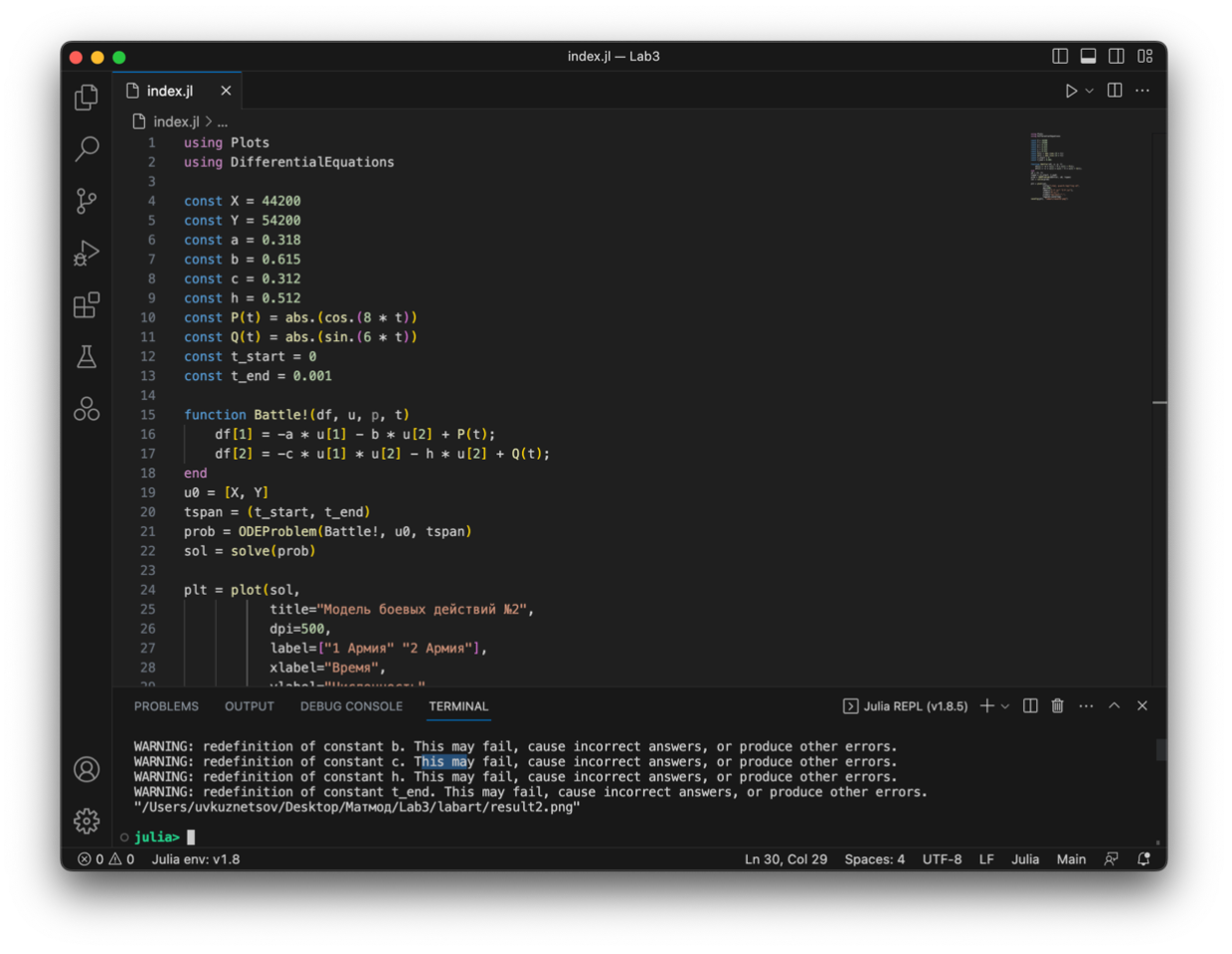




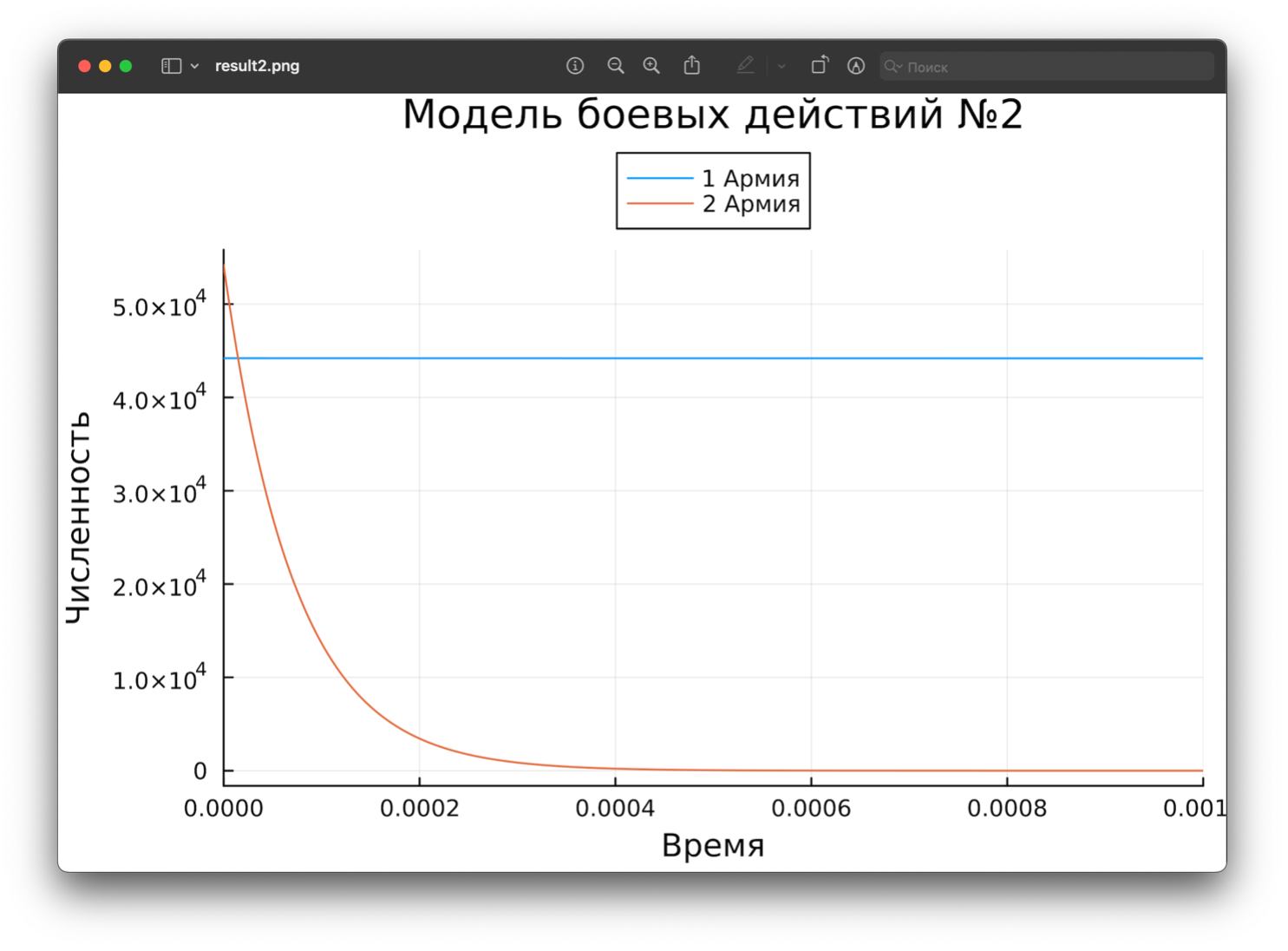
Изменяем код, чтобы построить скрипт для визуализации. Изменяем начальные данные и функцию, описывающую систему ОДУ.

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")

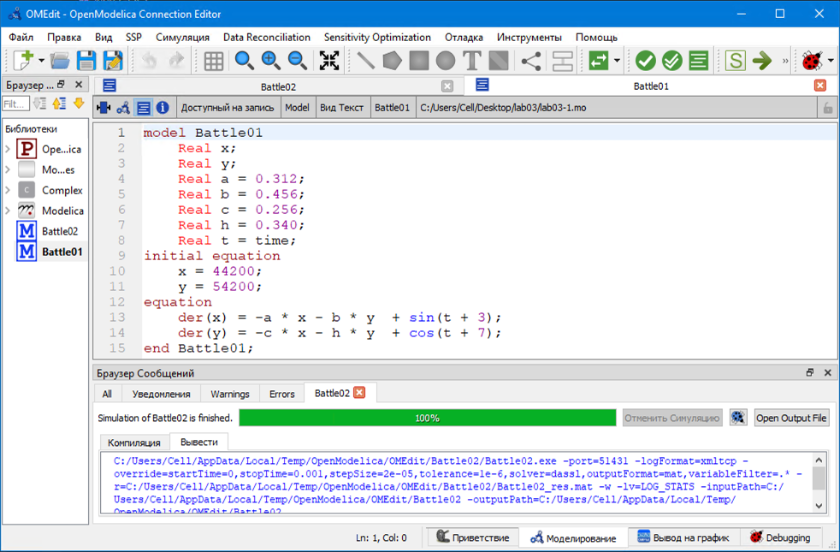


После запуска получаем визуализации модели боевых действий с участием регулярных войск и партизанских отрядов.

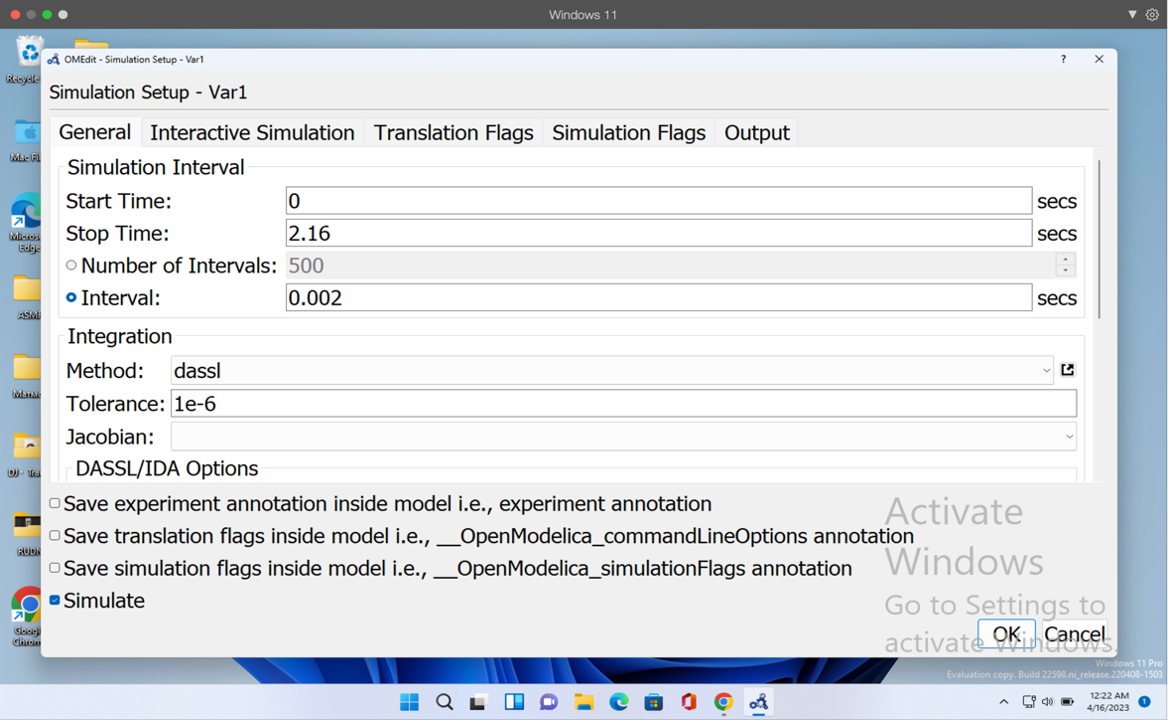


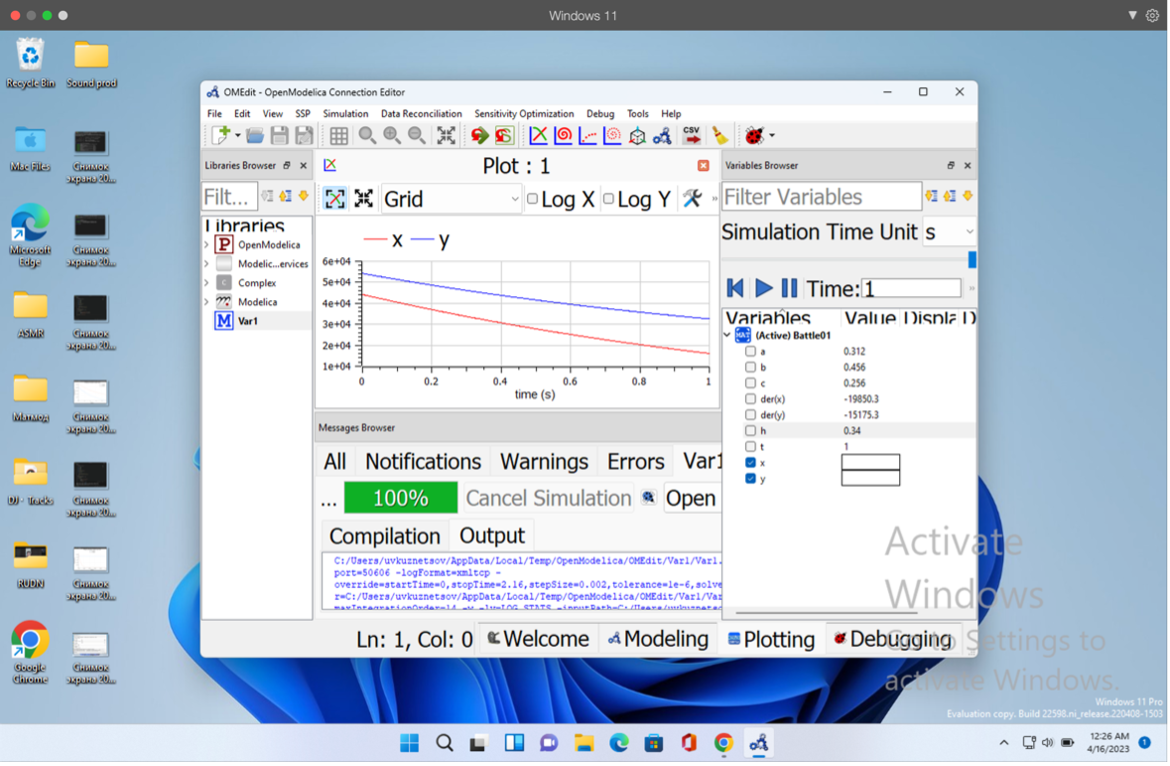
Пишем код для визуализации на OpenModelica. Для этого нам необходимо создать модель, внутри которой сначала объявим необходимые переменные, после чего проинициализирем начальные условия и в конце введем непосредственно уравнения для моделирования.

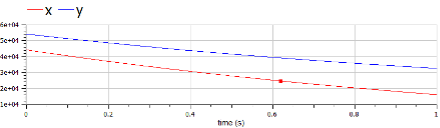
model Var1  
 Real x;  
 Real y;  
 Real a = 0.312;  
 Real b = 0.456;  
 Real c = 0.256;  
 Real h = 0.340;  
 Real t = time;  
initial equation  
 x = 44200;  
 y = 54200;  
equation  
 der(x) = -a \* x - b \* y + sin(t + 3);  
 der(y) = -c \* x - h \* y + cos(t + 7);  
end Var1;



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

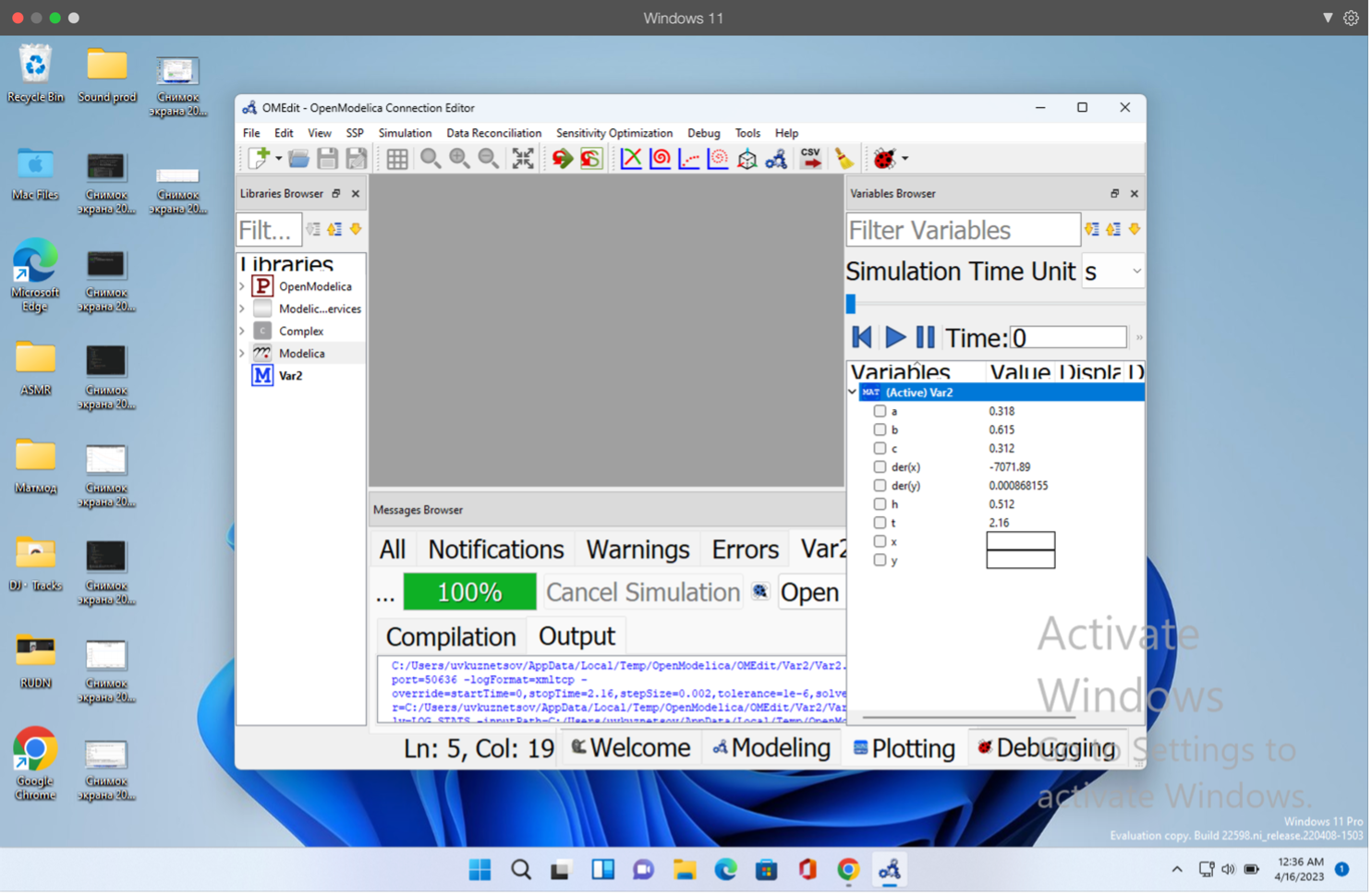




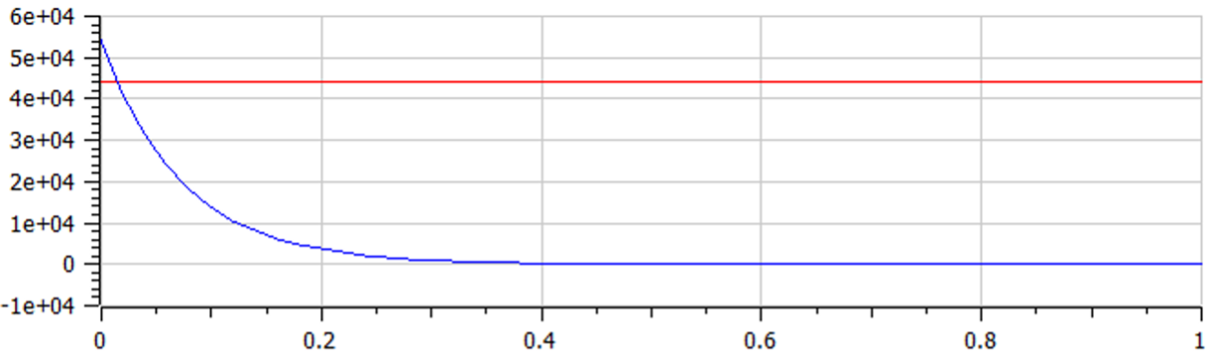


Приступим к моделированию боевых действий с участием регулярных войск и партизанских отрядов на OpenModelica. Для этого перепишем начальные данные и изменим уравнение, взяв за основу предыдущий скрипт.

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;



Запустим симуляцию, пропишем легенду и экспортируем график.



# 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.
* 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/.