Отчёт по лабораторной работе №3

Модель боевых действий. Вар. №28

Павлова Полина Алексеевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Рассмотреть некоторые простейшие модели боевых действий – модели Ланчестера. Решить поставленные задачи с помощью Julia и OpenModelica.

2 Задание. Вариант 28

Между страной X и страной У идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями x(t) и y(t). В начальный момент времени страна X имеет армию численностью 32 888 человек, а в распоряжении страны У армия численностью в 17 777 человек. Для упрощения модели считаем, что коэффициенты a, b, c, h постоянны. Также считаем P(t) и Q(t) непрерывные функции.

Постройте графики изменения численности войск армии X и армии У для следующих случаев:

1. Модель боевых действий между регулярными войсками:

$$\frac{dx}{dt} = -0.55x(t) - 0.77y(t) + 1.5sin(3t+1) + 1$$
$$\frac{dy}{dt} = -0.66x(t) - 0.44y(t) + 1.2cos(t+1)$$

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

$$\frac{dx}{dt} = -0.27x(t) - 0.88y(t) + sin(20t)$$
$$\frac{dy}{dt} = -0.68x(t)y(t) - 0.37y(t) + cos(10t) + 1$$

3 Теоретическое введение

Законы Ланчестера (законы Осипова — Ланчестера) — математическая формула для расчета относительных сил пары сражающихся сторон — подразделений вооруженных сил

Уравнения Ланчестера — это дифференциальные уравнения, описывающие зависимость между силами сражающихся сторон A и D как функцию от времени, причем функция зависит только от A и D.

В 1916 году, в разгар первой мировой войны, Фредерик Ланчестер разработал систему дифференциальных уравнений для демонстрации соотношения между противостоящими силами. Среди них есть так называемые Линейные законы Ланчестера (первого рода или честного боя, для рукопашного боя или неприцельного огня) и Квадратичные законы Ланчестера (для войн начиная с XX века с применением прицельного огня, дальнобойных орудий, огнестрельного оружия). В связи с установленным приоритетом в англоязычной литературе наметилась тенденция перехода от фразы «модель Ланчестера» к «модели Осипова — Ланчестера». [4]

- 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).

Рассмотривается три случая ведения боевых действий:

1. Боевые действия между регулярными войсками
2. Боевые действия с участием регулярных войск и партизанских отрядов
3. Боевые действия между партизанскими отрядами

4 Задачи

1. Построить модель боевых действий между регулярными войсками на языках Julia и OpenModelica
2. Построить модель ведения боевых действий с участием регулярных войск и партизанских отрядов на языках Julia и OpenModelica

5 Выполнение лабораторной работы

5.1 Математическая модель

5.1.1 Регулярная армия X против регулярной армии Y

Рассмотрим первый случай. Численность регулярных войск определяется тремя факторами:

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

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

$$\frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t)$$
$$\frac{dy}{dt} = -c(t)x(t) - h(t)y(t) + Q(t)$$

В первом пункте нами рассматривается как раз такая модель. Она является доработанной моделью Ланчестера, так его изначальная модель учитывала лишь члены $b(t)y(t)$ и $c(t)x(t)$, то есть, на потери за промежуток времени влияли лишь численность армий и “эффективность оружия” (коэффициенты $b(t)$ и $c(t)$).

$$\frac{dx}{dt} = -ax(t) - by(t) + P(t)$$
$$\frac{dy}{dt} = -cx(t) - hy(t) + Q(t)$$

Именно эти уравнения [3] и будут решать наши программы для выполнения первой части задания. В конце мы получим график кривой в декартовых координатах, где по оси ox будет отображаться численность армии государства X, по оси oy будет отображаться соответствующая численность армии Y. По тому, с какой осью пересечётся график, можно определить исход войны. Если ось ox будет пересечена в положительных значениях, победа будет на стороне армии государства X (так как при таком раскладе численность армии Y достигла нуля при положительном значении численности армии X). Аналогичная ситуация для оси oy и победы армии государства Y.

5.1.2 Регулярная армия X против партизанской армии Y

Для второй части задания, то есть, для моделирования боевых действий между регулярной армией и партизанской армией, необходимо внести поправки в предыдущую модель. Считается, что темп потерь партизан, проводящих свои операции в разных местах на некоторой известной территории, пропорционален не только численности армейских соединений, но и численности самих партизан.

$$\frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t)$$
$$\frac{dy}{dt} = -c(t)x(t)y(t) - h(t)y(t) + Q(t)$$

Коэффициенты a, b, c и h всё так же будут положительными десятичными числами:

$$\frac{dx}{dt} = -ax(t) - by(t) + P(t)$$
$$\frac{dy}{dt} = -cx(t)y(t) - hy(t) + Q(t)$$

5.2 Решение с помощью программ

5.2.1 Julia

5.2.1.1 Программный код решения на Julia [1]

Код программы:

```
using Plots;
using DifferentialEquations;

function one(du, u, p, t)
    du[1] = -0.55*u[1] - 0.77*u[2] + 1.5*sin(3*t+1)
    du[2] = -0.66*u[1] - 0.44*u[2] + 1.2*cos(t+1)
end

function two(du, u, p, t)
    du[1] = -0.27*u[1] - 0.88*u[2] + sin(20*t)
    du[2] = (-0.68*u[1] - 0.37)*u[2] + cos(10*t)+1
end

const people = Float64[32888, 17777]
const pron1 = [0.0, 3.0]
const pron2 = [0.0, 0.0007]
```

```
prob1 = ODEProblem(one, people, pron1)
prob2 = ODEProblem(two, people, pron2)

sol1 = solve(prob1, dtmax=0.1)
sol2 = solve(prob2, dtmax=0.000001)

A1 = [u[1] for u in sol1.u]
A2 = [u[2] for u in sol1.u]
T1 = [t for t in sol1.t]
A3 = [u[1] for u in sol2.u]
A4 = [u[2] for u in sol2.u]
T2 = [t for t in sol2.t]

plt1 = plot(dpi = 300, legend= true, bg =:white)
plot!(plt1, xlabel="Время", ylabel="Численность", title="Модель боевых действий №1"
plot!(plt1, T1, A1, label="Численность армии X", color =:red)
plot!(plt1, T1, A2, label="Численность армии Y", color =:green)
savefig(plt1, "lab03_1.png")

plt2 = plot(dpi = 1200, legend= true, bg =:white)
plot!(plt2, xlabel="Время", ylabel="Численность", title="Модель боевых действий  №2
plot!(plt2, T2, A3, label="Численность армии X", color =:red)
plot!(plt2, T2, A4, label="Численность армии Y", color =:green)
savefig(plt2, "lab03_2.png")
```

5.2.2 Результаты работы кода на Julia

На рис. 5.1 и 5.2 изображены итоговые графики для обоих случаев.

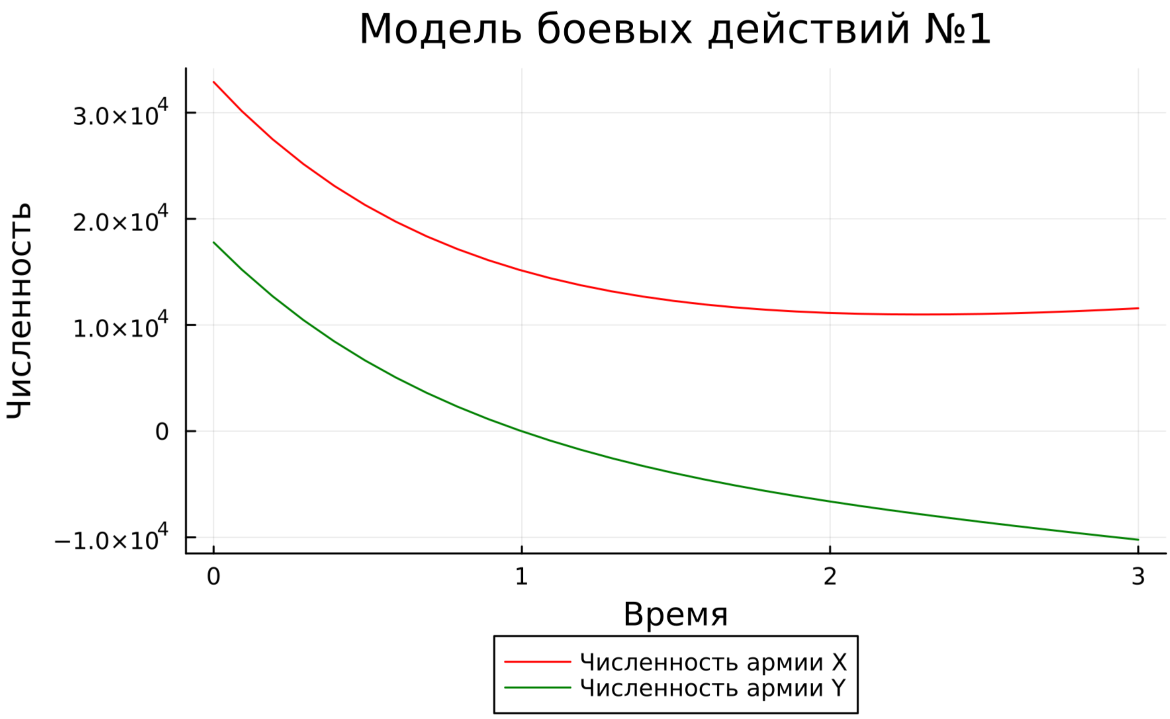

Рис. 5.1: “Полученный график Julia. Первый случай”

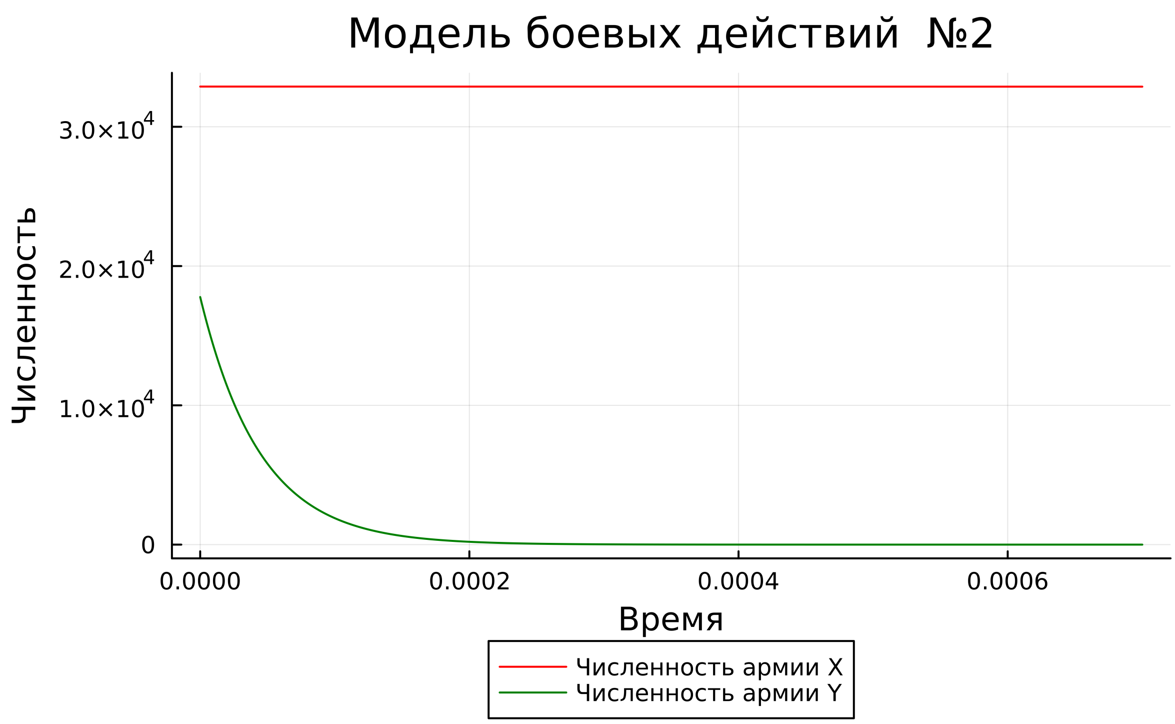

Рис. 5.2: “Полученный график Julia. Второй случай”

5.2.3 Программный код решения на OpenModelica [2]

```
model lab3_1
Real x;
Real y;
Real a = 0.55;
Real b = 0.77;
Real c = 0.66;
Real d = 0.44;
Real t = time;
initial equation
x = 32888;
y = 17777;
```

```
equation
der(x) = -a*x - b*y + 1.5*sin(3*t+1);
der(y) = -c*x - d*y + 1.2*cos(t+1);
end lab3_1;

model lab3_2
Real x;
Real y;
Real a = 0.27;
Real b = 0.88;
Real c = 0.68;
Real d = 0.37;
Real t = time;
initial equation
x = 32888;
y = 17777;
equation
der(x) = -a*x - b*y + sin(20*t);
der(y) = -c*x*y - d*y + cos(10*t)+1;
end lab3_2;
```

5.2.4 Результаты работы кода на OpenModelica

На графиках на рис. 5.3 и 5.4, построенных с помощью OpenModelica изображены графики, аналогичные графикам 5.2 и 5.3 соответственно.

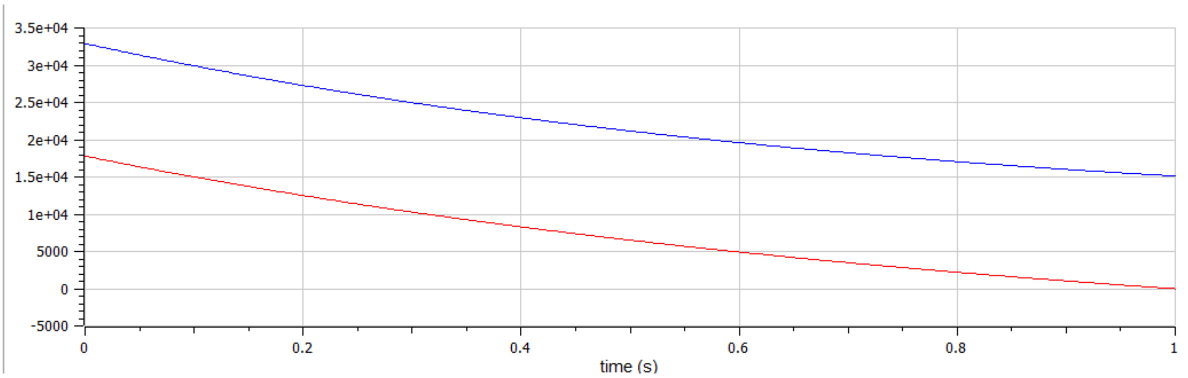

Рис. 5.3: “Полученный график OpenModelica. Первый случай”

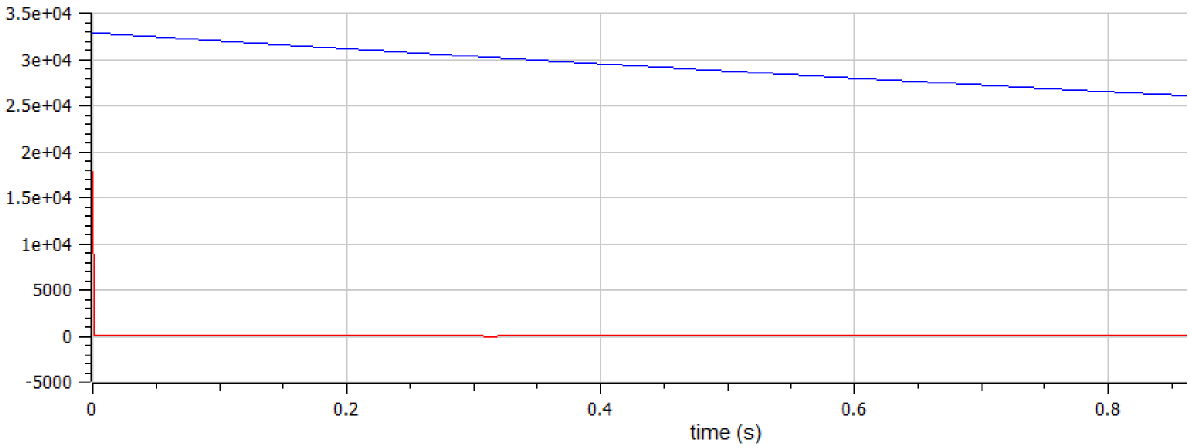

Рис. 5.4: “Полученный график OpenModelica. Второй случай”

6 Анализ полученных результатов. Сравнение языков.

Как видно из графиков, для первой модели, то есть двух регулярных армий, противостоящих друг другу, графики на Julia и OpenModelica идентичны (с поправкой на использование разных графических ресурсов, разный масштаб и т.д.).

Аналогичная ситуация верна и для графиков противостояния регулярной армии армии партизанов, которые рассматривались во второй модели.

7 Вывод

По итогам лабораторной работы я построила по две модели на языках Julia и OpenModelica. В ходе проделанной работы можно сделать вывод, что OpenModelica лучше приспособлен для моделирования процессов, протекающих во времени. Построение моделей боевых действий на языке OpenModelica занимает гораздо меньше строк и времени, чем аналогичное построение на языке Julia.

8 Список литературы. Библиография

[1] Документация по Julia: https://docs.julialang.org/en/v1/

[2] Документация по OpenModelica: https://openmodelica.org/

[3] Решение дифференциальных уравнений: https://www.wolframalpha.com/

[4] Законы Ланчестера: https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BA%D0%BE%D0%