

# Лабораторная работа №3

## Модель боевых действий

Шубнякова Дарья НКНбд-01-22

## **Содержание**

<b>1</b>	<b>Цель работы</b>	<b>3</b>
<b>2</b>	<b>Задание</b>	<b>3</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>3</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>4</b>
<b>5</b>	<b>Выводы</b>	<b>7</b>

## 1 Цель работы

Ознакомиться с задачей боевых действий и релизовать ее на языке Modelica в OMEdit, а так же на языке Julia.

## 2 Задание

Релизовать две модели:

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

## 3 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). Рассмотрим 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 В первом случае численность регулярных войск определяется тремя факторами:  $\square$  скорость уменьшения численности войск из-за причин, не связанных с боевыми действиями (болезни, травмы, дезертирство);  $\square$  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).

## 4 Выполнение лабораторной работы

Прописываем код на языке Julia(рис. 1).

```
[2]:  
  
using DifferentialEquations, Plots  
  
# Параметры задачи  
x0 = 40000.0 # Начальная численность армии X  
y0 = 69000.0 # Начальная численность армии Y  
tspan = (0.0, 50.0) # Временной интервал  
  
# Модель 1: Регулярные войска  
function regular_combat!(du, u, p, t)  
    x, y = u  
    du[1] = -0.331*x - 0.771*y - sin(t + 10) + 1 # dx/dt  
    du[2] = -0.401*x - 0.731*y - cos(t + 20) + 1 # dy/dt  
end  
  
# Модель 2: С партизанскими отрядами  
function partisan_combat!(du, u, p, t)  
    x, y = u  
    du[1] = -0.37*x - 0.73*y - 2*sin(6*t) # dx/dt  
    du[2] = -0.28*x - y - 0.82*y - 2*cos(4*t) # dy/dt  
end  
  
# Решение для регулярных войск  
u0 = [x0, y0]  
prob_regular = ODEProblem(regular_combat!, u0, tspan)  
sol_regular = solve(prob_regular, Tsit5(), reltol=1e-6, abstol=1e-6)  
  
# Решение для модели с партизанами  
prob_partisan = ODEProblem(partisan_combat!, u0, tspan)  
sol_partisan = solve(prob_partisan, Tsit5(), reltol=1e-6, abstol=1e-6)  
  
# Построение графиков  
plt1 = plot(sol_regular, idxs=[1, 2], label=["Армия X" "Армия Y"],  
            title="Регулярные войска", xlabel="Время", ylabel="Численность",  
            linewidth=2, legend=:right)  
  
plt2 = plot(sol_partisan, idxs=[1, 2], label=["Армия X" "Армия Y"],  
            title="С партизанскими отрядами", xlabel="Время", ylabel="Численность",  
            linewidth=2, legend=:right)
```

Рисунок 1

Продолжение кода(рис. 2).

```

# Отображение графиков
plot(plt1, plt2, layout=(2,1), size=(800, 600), dpi=300)

# Сохранение результатов
savefig("combat_models_comparison.png")
println("Графики сохранены в файл: combat_models_comparison.png")

# Анализ результатов
println("Регулярные войска - конечные значения:")
println("Армия X: ", round(sol_regular[1,end], digits=0))
println("Армия Y: ", round(sol_regular[2,end], digits=0))

println("\nC партизанами - конечные значения:")
println("Армия X: ", round(sol_partisan[1,end], digits=0))
println("Армия Y: ", round(sol_partisan[2,end], digits=0))

```

Графики сохранены в файл: combat\_models\_comparison.png  
Регулярные войска - конечные значения:  
Армия X: -364763.0  
Армия Y: 184940.0

С партизанами - конечные значения:  
Армия X: -0.0  
Армия Y: 0.0

Рисунок 2

Получаем два таких графика на выходе, они сохраняются в папку с lab3.ipynb(рис. 3).

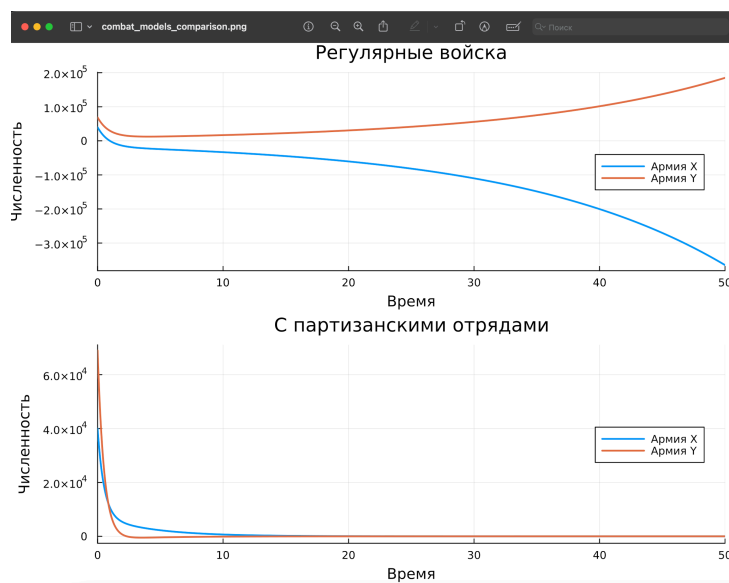


Рисунок 3

Пишем в OpenModelica код для регулярных войск(рис. 4).

```

/home/openmodelica/Downloads/RegularForces.mo - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
model RegularForces "Модель регулярных войск"
// Начальные условия
parameter Real x0 = 40000 "Начальная численность армии X";
parameter Real y0 = 69000 "Начальная численность армии Y";

// Переменные
Real x(start = x0, fixed = true) "Численность армии X";
Real y(start = y0, fixed = true) "Численность армии Y";
Real t "Время";

equation
// Система дифференциальных уравнений
der(x) = -0.331*x - 0.771*y - sin(t + 10) + 1;
der(y) = -0.401*x - 0.731*y - cos(t + 20) + 1;
der(t) = 1;
annotation(experiment(StartTime=0, StopTime=50, Tolerance=1e-6, Interval=0.1))
end RegularForces;

```

Рисунок 4

Получаем данный график(рис. 5).

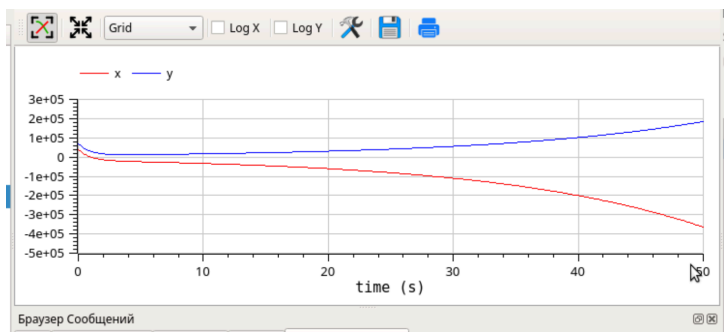


Рисунок 5

Строим модель с партизанскими войсками(рис. 6).

```

/home/openmodelica/Downloads/WithPartisans.mo - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
model WithPartisans "Модель с партизанскими отрядами"
// Начальные условия
parameter Real x0 = 40000 "Начальная численность армии X";
parameter Real y0 = 69000 "Начальная численность армии Y";

// Переменные
Real x(start = x0, fixed = true) "Численность армии X";
Real y(start = y0, fixed = true) "Численность армии Y";
Real t "Время";

equation
// Система дифференциальных уравнений
der(x) = -0.37*x - 0.73*y - 2*sin(6*t);
der(y) = -0.28*x - y - 0.82*y - 2*cos(4*t);
der(t) = 1;

annotation(experiment(StartTime=0, StopTime=50, Tolerance=1e-6, Interval=0.1));
end WithPartisans;

```

Рисунок 6

Полученный график выглядит так(рис. 7).

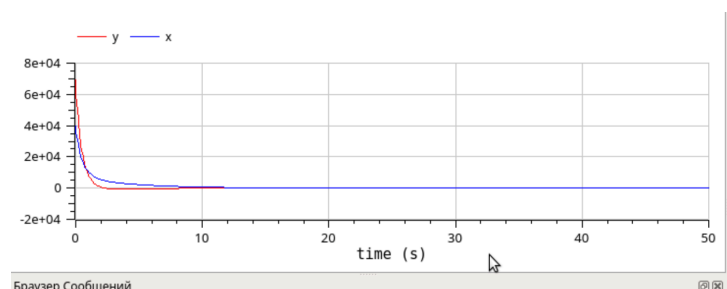


Рисунок 7

## 5 Выводы

Мы ознакомились с моделью боевых действий. Получили на выходе два файла из OpenModelica: RegularForces.mo, WithPartisans.mo. Получили картинку с графиками: combat\_models\_comparison.png, а так же код в оболочке Julia, прописанный в JupiterNotebook (lab3.ipynb).