

# Лабораторная работа №3

## Модель боевых действий

---

Майсаров А. М.

15.04.2023

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Майсаров Аббас Мурадович
- студент НФИбд-01-20
- Российский университет дружбы народов
- 1032200530@pfur.ru

# **Вводная часть**

---

- Моделирование ситуации
- Наглядное представление
- Простота использования

- Сделать моделирование боевых действий
- Представить результаты в виде графиков

- Язык `Julia` и ее библиотеки: `Plots` и `DifferentialEquations` для построения графиков
- Свободное открытое программное обеспечение `OpenModelica` для моделирования ситуации

## Ход работы

---



## Построение двух простейших моделей боя

1. Модель боевых действий между регулярными войсками описывается следующим ОДУ

$$dx/dt = -ax(t) - bx(t) + P(t)$$

$$dy/dt = -cx(t) - hy(t) + Q(t)$$

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

$$dx/dt = -a(t)x(t) - b(t)x(t) + P(t)$$

$$dy/dt = -c(t)x(t)y(t) - h(t)y(t) + Q(t)$$

где:  $a, b, c, h$  - постоянные коэффициенты

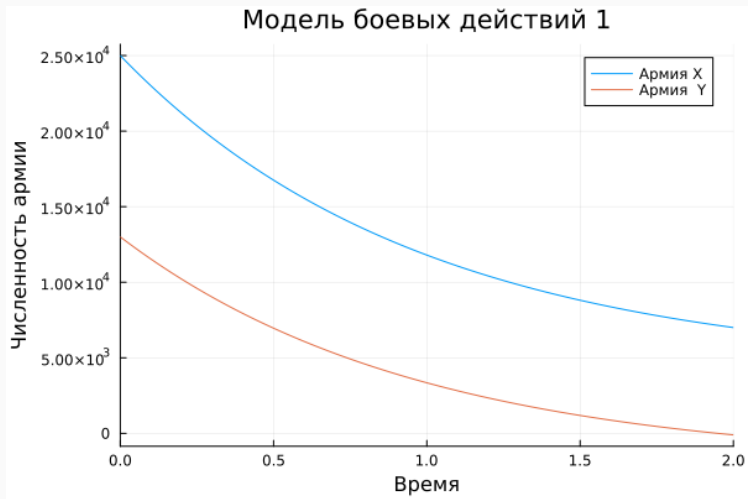
$a(t), h(t)$  - коэффициенты, описывающие потери, не связанные с боевыми действиями

$c(t), b(t)$  - коэффициенты, описывающие потери, связанные с боевыми

# Код моделирования на Julia, модель 1

```
1 using Plots
2 using DifferentialEquations
3
4 x_start = 25000
5 y_start = 13000
6 a = -0.33
7 b = -0.88
8 c = -0.44
9 h = -0.77
10 P(t) = sin(t)
11 Q(t) = cos(3*t)
12 t_start = 0
13 t_end = 0.01
14 dt = 50
15 t = collect(LinRange(t_start, t_end, dt))
16
17 function F(du, u, p, t)
18     du[1] = a*u[1] + b*u[2] + P(t)
19     du[2] = c*u[1]*u[2] + h*u[2] + Q(t)
20 end
21
22 u0 = [x_start, y_start]
23
24 prob = ODEProblem(F, u0, (t_start, t_end))
25
26 solving = solve(prob)
27
28 plt = plot(solving, vars=(0,1), title="Модель боевых действий 1", ylabel = "Численность армии", label = "Армия X")
29 plot!(solving, vars=(0,2), xlabel = "Время", label = "Армия Y")
30
31 savefig(plt, "fig004.png")
```

# График моделирования первой модели



## Код моделирования на Julia, модель 2

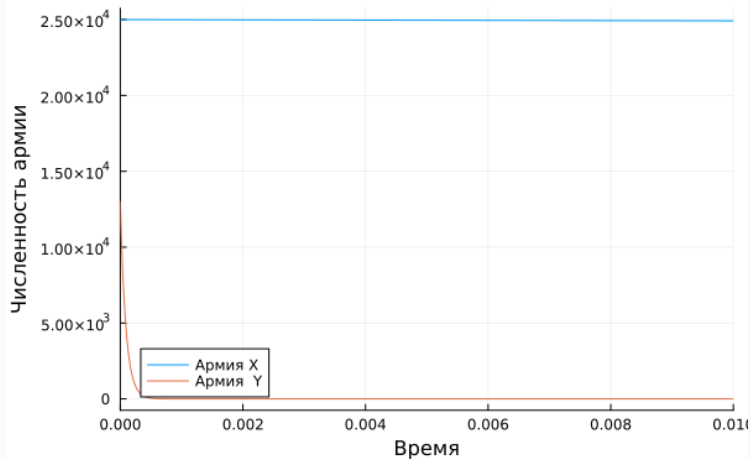
---

# Код моделирования на Julia, модель 2

```
1 using Plots
2 using DifferentialEquations
3
4 x_start = 25000
5 y_start = 13000
6 a = -0.41
7 b = -0.83
8 c = -0.29
9 h = -0.63
10 P(t) = sin(t+3)
11 Q(t) = cos(t+3)
12 t_start = 0
13 t_end = 2
14 dt = 50
15 t = collect(LinRange(t_start, t_end, dt))
16
17 function F(du, u, p, t)
18     du[1] = a*u[1] + b*u[2] + P(t)
19     du[2] = c*u[1] + h*u[2] + Q(t)
20 end
21
22 u0 = [x_start, y_start]
23
24 prob = ODEProblem(F, u0, (t_start, t_end))
25
26 solving = solve(prob)
27
28 plt = plot(solving, vars=(0,1), title="Модель боевых действий 1", ylabel = "Численность армии", label = "Армия X")
29 plot!(solving, vars=(0,2), xlabel = "Время", label = "Армия Y")
30
31 savefig(plt, "fig003.png")
```

# График моделирования второй модели

Модель боевых действий 1



## Код на OpenModelica. Первая модель

```
1  model lab3v1
2  parameter Integer x_start = 25000;
3  parameter Integer y_start = 13000;
4  parameter Real a = -0.41;
5  parameter Real b = -0.83;
6  parameter Real c = -0.29;
7  parameter Real h = -0.63;
8  Real x(start=x_start);
9  Real y(start=y_start);
10 Real P;
11 Real Q;
12 equation
13 P = sin(time + 3);
14 Q = cos(time + 3);
15 der(x) = a*x + b*y + P;
16 der(y) = c*x + h*y + Q;
17 end lab3v1;
```

## Код на OpenModelica. Вторая модель

```
1 model lab3v2
2   parameter Integer x_start = 25000;
3   parameter Integer y_start = 13000;
4   parameter Real a = -0.33;
5   parameter Real b = -0.88;
6   parameter Real c = -0.44;
7   parameter Real h = -0.77;
8   Real x(start=x_start);
9   Real y(start=y_start);
10  Real P;
11  Real Q;
12  equation
13    P = sin(time);
14    Q = cos(3*time);
15    der(x) = a*x + b*y + P;
16    der(y) = c*x*y + h*y + Q;
17  end lab3v2;
```



# Установка настроек симуляции

OMEdit - Установки Симуляции - lab3v2

## Установки Симуляции - lab3v2

Основное | Интерактивная Симуляция | Translation Flags | Флаги Симуляции | Вывести

Интервал Симуляции

Начальное Время: 0 secs

Конечное Время: 2 secs

☒ Число Интервалов: 500

☐ Interval: 0.002 secs

Интегрирование

Метод: dassl

Точность: 1e-6

Якобиан:

DASSL/IDA Options

☐ Save experiment annotation inside model i.e., experiment annotation

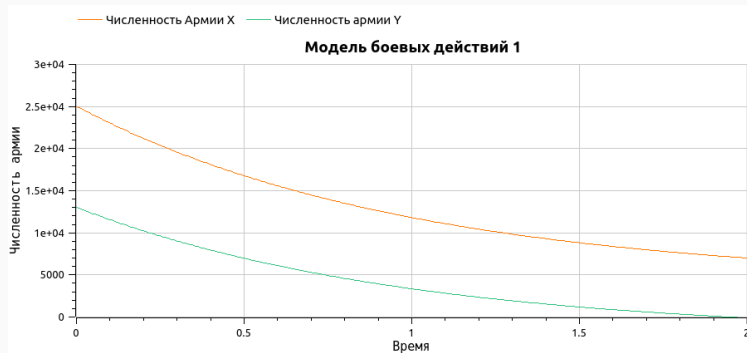
☐ Save translation flags inside model i.e., \_\_OpenModelica\_commandLineOptions annotation

☐ Save simulation flags inside model i.e., \_\_OpenModelica\_simulationFlags annotation

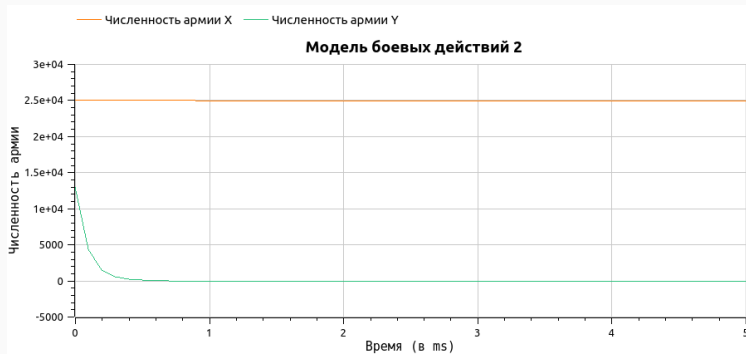
☒ Симулировать

OK Отмена

# График первой модели



## График второй модели



## Результаты

---

- Моделирование ситуации
- Ознакомление с языками
- Исследование графиков
- Сравнение языков