

Лабораторная работа №3

Математическое моделирование

Байрамгельдыев Д.

25 февраля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Байрамгельдыев Довлетмурат
- студент 3 курса группы НФИбд-01-20
- ст. б. 1032207470
- Российский университет дружбы народов
- 1032207470@pfur.ru

Вводная часть

- Необходимость предоставления командованию оснований для принятия решений
- Необходимость визуализировать данные
- Простота построения моделей

- Построить математическую модель боевых действий
- Визуализировать модель с помощью Julia и OpenModelica
- Проанализировать полученные результаты

- Средства языка **Julia** для визуализации данных
- GUI **OMEdit** для визуализации данных на **OpenModelica**
- Результирующие форматы
 - `jl`
 - `mo`
 - `png`

Ход работы

Модель для первого случая:
$$\begin{cases} \frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t) \\ \frac{dy}{dt} = -c(t)x(t) - h(t)y(t) + Q(t) \end{cases}$$

Модель для второго случая:
$$\begin{cases} \frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t) \\ \frac{dy}{dt} = -c(t)x(t)y(t) - h(t)y(t) + Q(t) \end{cases}$$

Программа на языке Julia, 1

```
using Plots
using DifferentialEquations

const x0 = 30000
const y0 = 17000

const a = 0.45
const b = 0.86
const c = 0.49
const h = 0.73

P(t) = sin(t) + 1
Q(t) = cos(t) + 2

u0 = [x0, y0]
p = (a, b, c, h)
T = (0, 1.5)

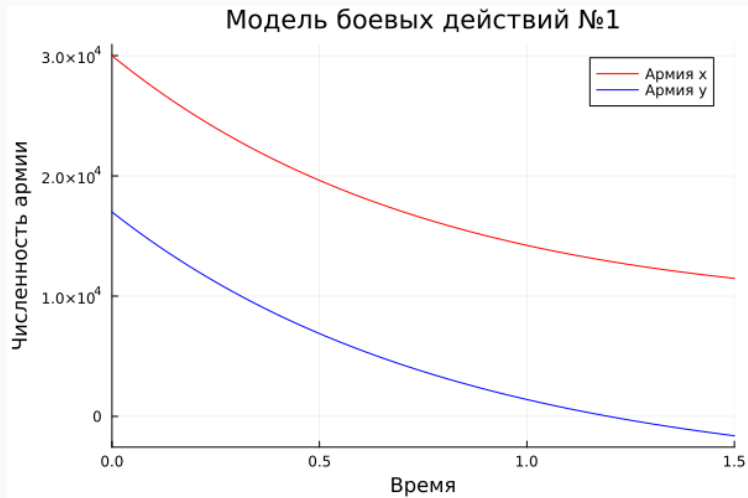
function F(du, u, p, t)
    a, b, c, h = p
    du[1] = -a*u[1] - b*u[2] + P(t)
    du[2] = -c*u[1] - h*u[2] + Q(t)
end

prob = ODEProblem(F, u0, T, p)

sol = solve(prob)

plt = plot(sol, vars=(0,1), color=:red, label="Армия x", title="Модель боевых действий №1", ylabel="Численность армии")
plot!(sol, vars=(0,2), color=:blue, label="Армия y", xlabel="Время")

savefig(plt, "lab3_1.png")
```



Программа на языке Julia, 2

```
const a = 0.34
const b = 0.81
const c = 0.22
const h = 0.91
```

```
P(t) = abs(sin(2*t))
Q(t) = abs(cos(t))
```

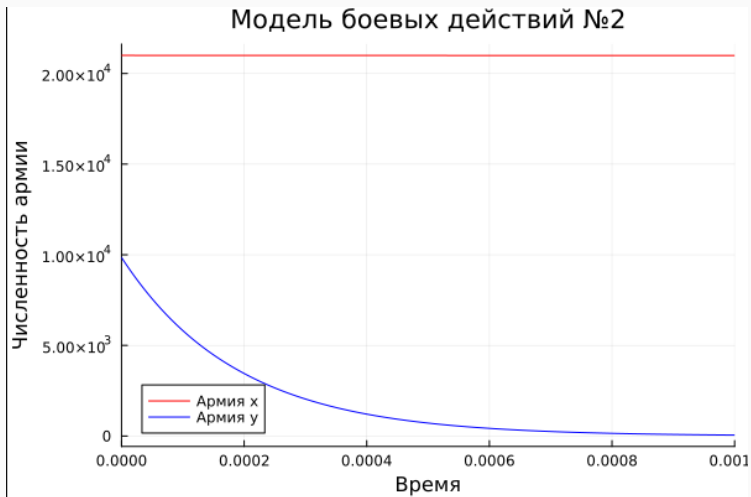
```
u0 = [x0, y0]
p = (a, b, c, h)
```

```
T = (0, 0.001)
```

```
function F(du, u, p, t)
    a, b, c, h = p
    du[1] = -a*u[1] - b*u[2] + P(t)
    du[2] = -c*u[1]*u[2] - h*u[2] + Q(t)
end
```

```
T = (0, 1.5)
```

```
function F(du, u, p, t)
    a, b, c, h = p
    du[1] = -a*u[1] - b*u[2] + P(t)
    du[2] = -c*u[1] - h*u[2] + Q(t)
end
```



```
model lab03
parameter Real a( start=0.45);
parameter Real b( start=0.86);
parameter Real c( start=0.49);
parameter Real h( start=0.73);
Real x(start=30000);
Real y(start=17000);

equation
  der(x)=-a*x-b*y+2*sin(time+1);
  der(y)=-c*x-h*y+2*cos(time+2);

  annotation(experiment(StartTime=0, StopTime=1, Tolerance=1e-6, Interval=0.05));
end lab03;
```

Установки Симуляции - Battle

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

Интервал Симуляции


Начальное Время: secs

Конечное Время: secs

☐ Число Интервалов:

☒ Interval: secs

Интегрирование

Метод: 

Точность:

Якобиан:

DASSL/IDA Options

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

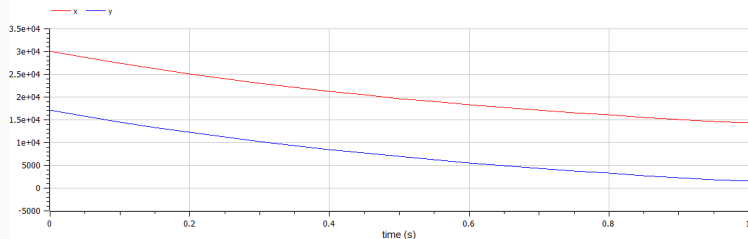
☐ Save translation flags inside model i.e., __OpenModelica_commandLineOptions annotation

☐ Save simulation flags inside model i.e., __OpenModelica_simulationFlags annotation

☒ Симулировать

OK Отмена

Программа на языке OpenModelica, 2



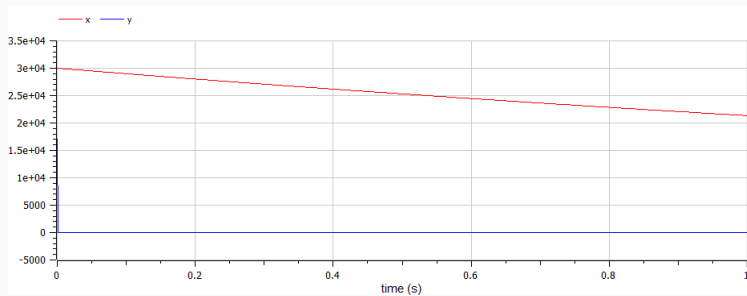
```
model lab03_1
```

```
parameter Real a( start=0.34);
parameter Real b( start=0.81);
parameter Real c( start=0.22);
parameter Real h( start=0.91);
Real x(start=30000);
Real y(start=17000);

equation
  der(x)=-a*x-b*y+sin(2*time);
  der(y)=-c*x*y-h*y+cos(time);

  annotation(experiment(StartTime=0, StopTime=1, Tolerance=1e-6, Interval=0.002));
end lab03_1;
```


График на языке OpenModelica, 2



Результаты

- Приобретены навыки работы с Julia и OpenModelica
- Построена математическая модель боевых действий
- Построены графики, визуализирующие решение задачи
- OpenModelica является более подходящим языком для работы с данной моделью-