

Отчёт по лабораторной работе №3

Модель боевых действий

Van I

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Модель боевых действий между регулярными войсками на Julia .	11
4.2	Модель ведение боевых действий с участием регулярных войск и партизанских отрядов на Julia	11
4.3	Модель боевых действий между регулярными войсками на OpenModelica	13
4.4	Модель ведение боевых действий с участием регулярных войск и партизанских отрядов на OpenModelica	13

Список таблиц

1 Цель работы

- Построение модели боевых действий на языках Julia и OpenModelica.
- Решение ОДУ 1 порядка с помощью графика.
- Рассмотрение модели боевых действий между регулярными войсками.
- Рассмотрение модели ведения боевых действий с участием регулярных войск и партизанских отрядов.

2 Задание

Существуют три модели боя.

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

В нашей работе мы рассмотрим только первых две модели.

Проверим, как работает модель в различных ситуациях, постройте графики $y(t)$ и $x(t)$ в рассматриваемых случаях.

3 Теоретическое введение

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 действиями

$P(t), Q(t)$ - функции, учитывающие возможность подхода подкрепления к войскам

Подробнее о модели боевых действий в [1,2]

4 Выполнение лабораторной работы

Напишем код на Julia с помощью средств языка и библиотек Plots и DifferentialEquations. Решаем наши ОДУ [3], строим график.

1)

```
#variant 50
using Plots
using DifferentialEquations

x0 = 80000
y0 = 60000

a = -0.21
b = -0.855
c = -0.455
h = -0.32

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

function F(du, u, p, t)
    x,y = u
    du[1] = a*u[1] + b*u[2] + P(t)
```



```

        du[2] = c*u[1] + h*u[2] + Q(t)
    end

    u0 = [x0,y0]
    tspan = (0.0, 3.0)
    pr = ODEProblem(F, u0, tspan)
    sol = solve(pr)

    xx = [u[1] for u in sol.u]
    yy = [u[2] for u in sol.u]
    T = [t for t in sol.t]
    plott = plot(dpi=500, title = "Combat model", legend=true)
    plot!(plott, T, xx, label = "X army size", color=:brown)
    plot!(plott, T, yy, label = "Y army size", color=:green)

    savefig(plott, "lab3_1jl.png")

```

2)

```

#variant 50
using Plots
using DifferentialEquations

x0 = 80000
y0 = 60000

a = -0.267
b = -0.687
c = -0.349
h = -0.491

```

```

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

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

u0 = [x0,y0]
tspan = (0.0, 0.002)
pr = ODEProblem(F, u0, tspan)
sol = solve(pr)

xx = [u[1] for u in sol.u]
yy = [u[2] for u in sol.u]
T = [t for t in sol.t]
plott = plot(dpi=500, title = "Combat model", legend=true)
plot!(plott, T, xx, label = "X army size", color=:brown)
plot!(plott, T, yy, label = "Y army size", color=:green)

savefig(plott, "lab3_2jl.png")

```

Первый полученный график показывает нам - численность армии Y проигрывает (рис. 4.1).

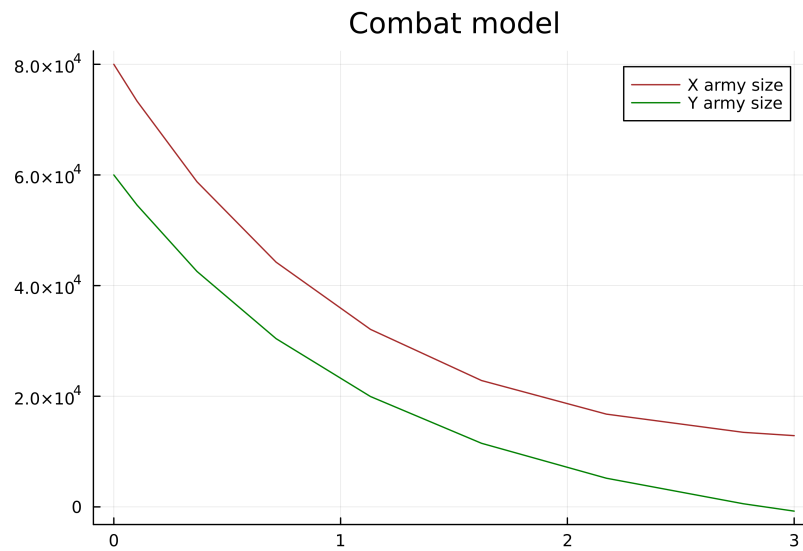


Рис. 4.1: Модель боевых действий между регулярными войсками на Julia

На втором графике проигрывает армия Y.

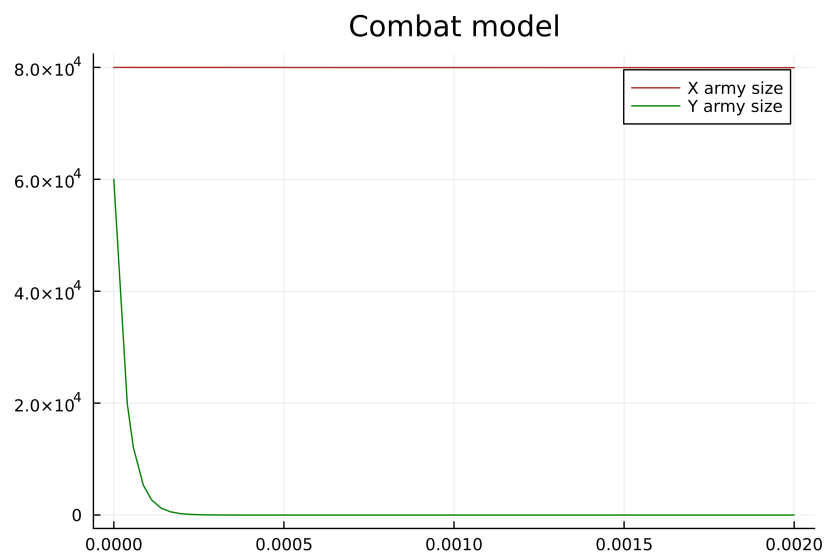


Рис. 4.2: Модель ведение боевых действий с участием регулярных войск и партизанских отрядов на Julia

Дальше переходим к написанию кода на OpenModelica. 1) Первая модель.

model lab3_1

```

parameter Integer x0 = 80000;
parameter Integer y0 = 60000;

parameter Real a = -0.21;
parameter Real b = -0.855;
parameter Real c = -0.455;
parameter Real h = -0.32;
Real x(start=x0);
Real y(start=y0);
equation
der(x) = a*x + b*y + sin(time) + 2;
der(y) = a*x + b*y + cos(time) + 2;
end lab3_1;

```

2)

```

model lab3_2
parameter Integer x0 = 80000;
parameter Integer y0 = 60000;

parameter Real a = -0.267;
parameter Real b = -0.687;
parameter Real c = -0.349;
parameter Real h = -0.491;
Real x(start=x0);
Real y(start=y0);
equation
der(x) = a*x + b*y + abs(sin(2*time));
der(y) = a*x*y + b*y + abs(2*cos(time));
end lab3_2;

```

Настраиваем нашу симуляцию.

Получаем наши графики моделей боя (рис. 4.3) (рис. 4.4).

Графики похожи на графики в Julia, значит, мы сделали все верно. Исходы боя получили аналогичным на Julia.

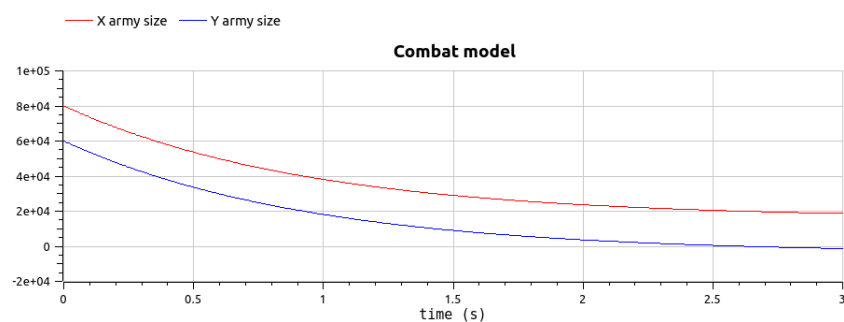


Рис. 4.3: Модель боевых действий между регулярными войсками на OpenModelica

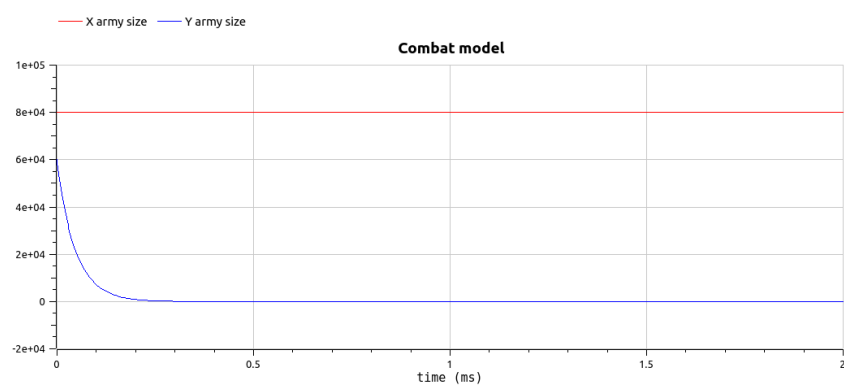


Рис. 4.4: Модель ведение боевых действий с участием регулярных войск и партизанских отрядов на OpenModelica

5 Выводы

В ходе лабораторной работы были получены навыки работы с простейшими моделями боетвий. Укрепили наши навыки работы на Julia и OpenModelica. Результат работы - графики, наглядно показывающие результат. если сравнивать эти языки программирования, то для решения этой задачи мне понравился OpenModelica. Просто, наглядно, быстро и много интересных настроек, которые легко менять для изучения ситуации.

Список литературы

1. Теоритический материал "Модель боевых действий" [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.php/1971652/mod_resource/content/2/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%96%202.pdf.
2. Законы Ланчестера [Электронный ресурс]. URL: https://wiki5.ru/wiki/Lanchester%27s_laws.
3. Решение ОДУ на OpenModelica [Электронный ресурс]. URL: <https://habr.com/ru/post/202596/>.