Лабораторная работа №3

Математическое моделирование

Байрамгельдыев Довлетмурат

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

- Построение простейшей математической модели боевых действий — модели Ланчестера
- Визуализация модели на языках Julia и OpenModelica
- Сравнение языков Julia и OpenModelica

2 Задание

- Построить график изменения численности войск армии X и армии Y на основе модели боевых действий между регулярными войсками
- Построить график изменения численности войск армии X и армии Y на основе модели боевых действий с участием регулярных войск и партизанских отрядов
- Проанализировать графики

3 Теоретическое введение

Модель Ланчестера является простейшей моделью для описания боевых действий. Основной характеристикой соперников являются численности сторон (если какая-то из численностей обращается в ноль при положительной численности соперника, то данная сторона считается проигравшей).

Существует три случая ведения боевых действий:

- Боевые действия между регулярными войсками
- Боевые действия с участием регулярных войск и партизанских отрядов
- Боевые действия между партизанскими отрядами

В рамках нашей задачи мы будем рассматривать только первые два случая. Для описания этих случаев будут использоваться общие обозначения:

- a(t) и h(t) — коэффициенты, описывающие потери, не связанные с боевыми действиями (болезнь, дезертирство и пр.)
- b(t) и c(t) — коэффициенты, отражающие потери на поле боя
- P(t) и Q(t) — функции, учитывающие возможность подхода подкрепления к войскам

В первом случае (бой между регулярными войсками) модель имеет вид:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = -a(t)x(t) - b(t)y(t) + P(t) \\ \frac{\mathrm{d}y}{\mathrm{d}t} = -c(t)x(t) - h(t)y(t) + Q(t) \end{cases}$$

Во втором случае в борьбу добавляются партизанские отряды. Нерегулярные войска в отличии от постоянной армии менее уязвимы, так как действуют скрытно, в этом случае сопернику приходится действовать неизбирательно, по площадям, занимаемым партизанами. Поэтому 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. В результате модель принимает вид:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = -a(t)x(t) - b(t)y(t) + P(t) \\ \frac{\mathrm{d}y}{\mathrm{d}t} = -c(t)x(t)y(t) - h(t)y(t) + Q(t) \end{cases}$$

4 Выполнение лабораторной работы

Построив модель для случая боевых действий между регулярными войсками, пишем программу на языке Julia (рис. 4.1). Указываем начальные значения и коэффициенты, задаем функции возможности подхода подкрепления и функцию, описывающую нашу модель. С помощью библиотеки DifferentialEquations находим решение системы [**ode-solve?**] и визуализируем его средствами библиотеки Plots.

```
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")
```

Рис. 4.1: Программа на Julia для первого случая

Из полученного графика можно сделать вывод, что армия Y в заданных условиях является проигравшей стороной, так как численность ее армии доходит до нуля (рис. 4.2).

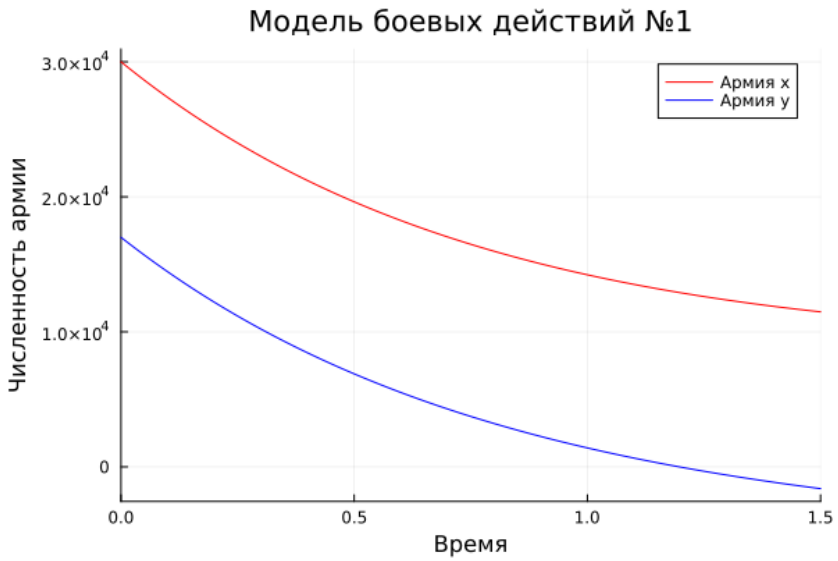

Рис. 4.2: График на Julia для первого случая

Меняем значения коэффициентов и функций P(t) и Q(t), а также слегка меняем функцию, описывающую нашу модель, чтобы она соответствовала второму случаю (рис. 4.3).

```
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
```

Рис. 4.3: Программа на Julia для второго случая

Из графика видно, что армия Y стремительно теряет в численности при указанных условиях и опять проигрывает (рис. 4.4).

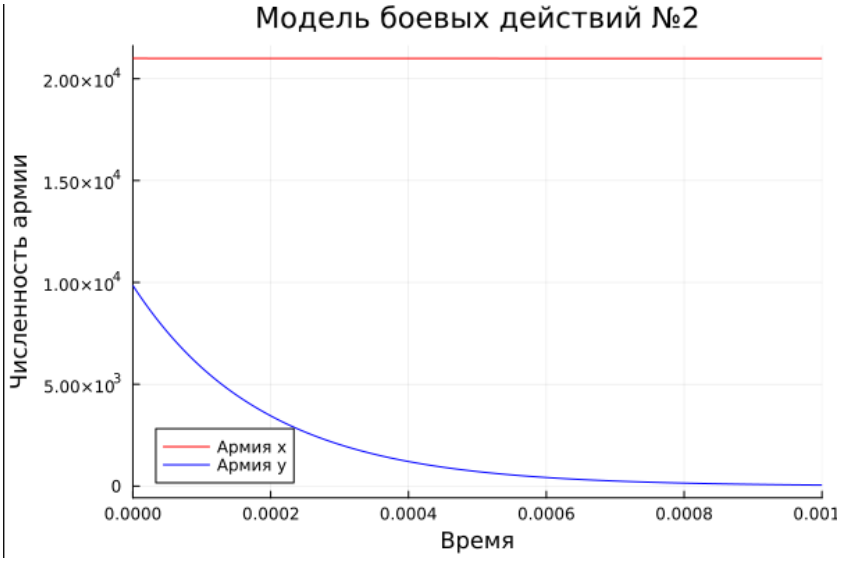

Рис. 4.4: График на Julia для второго случая

Далее описываем модель для первого случая на OpenModelica (рис. 4.5). В параметрах указываем начальные значения и коэффициенты, в разделе equation задаем функции, описывающие модель.

```
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;
```

Рис. 4.5: Программа на OpenModelica для первого случая

В установке симуляции настроим начальное и конечное время, а также размер интервала (рис. 4.6).

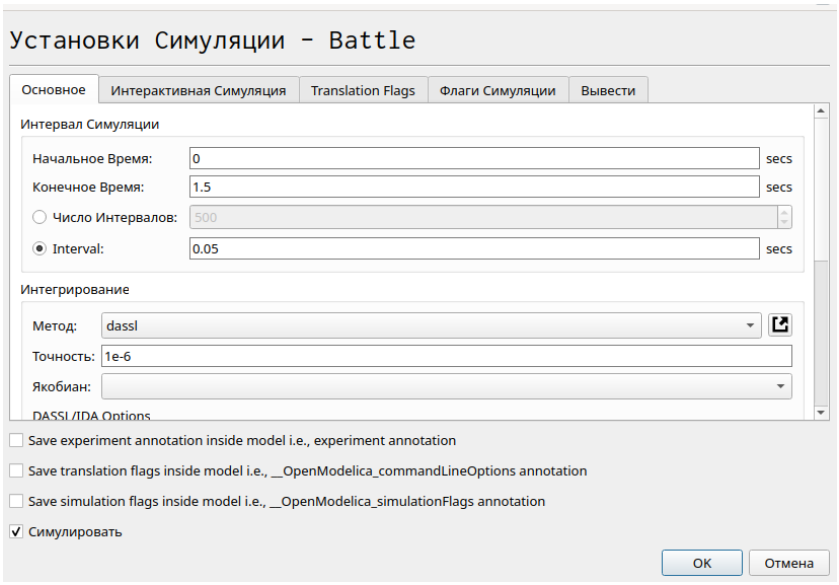

Рис. 4.6: Установка симуляции OpenModelica

Полученный график демонстрирует, что армия Y находится на проигрывающей позиции (рис. 4.7).

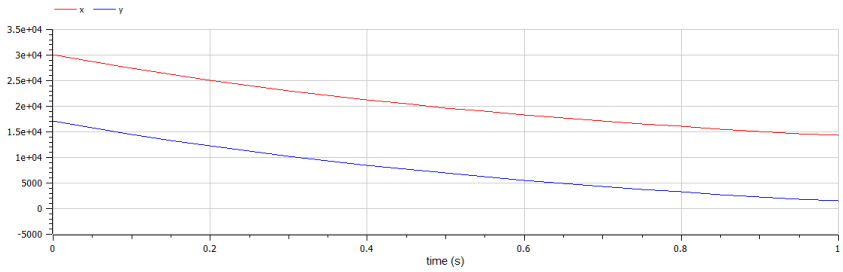

Рис. 4.7: График на OpenModelica для первого случая

Для второго случая меняем значения параметров-коэффициентов (рис. 4.8) и функции, задающие модель (рис. **??**).

```
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;
```

Рис. 4.8: Изменение параметров на OpenModelica для второго случая

График показывает, что проигрывающей стороной является армия Y, чья численность значительно упала в короткие сроки (рис. 4.9).

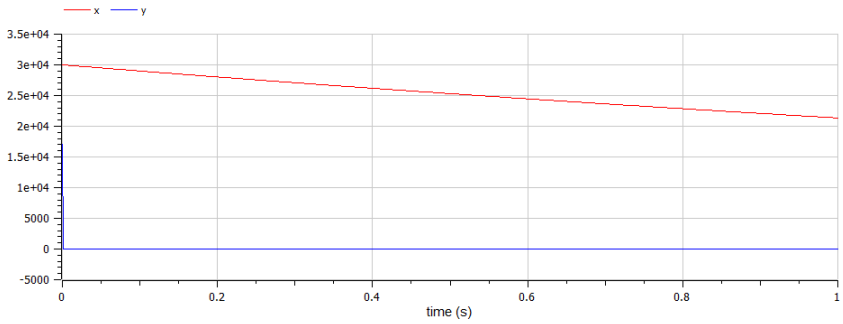

Рис. 4.9: График на OpenModelica для второго случая

5 Выводы

В ходе работы была освоена простейшая модель боевых действий — модель Ланчестера, и были применены навыки работы с Julia и OpenModelica для визуализации модели с помощью графиков. Результатом работы стали графики, демонстрирующие изменение численности двух армий на основе двух случаев ведения боевых действий, которые позволили нам сделать выводы о том, какая сторона считается проигравшей.

Сравнивая Julia и OpenModelica, отмечу, что, на мой взгляд, OpenModelica больше подходит для решения данной задачи, так как она специализируется на работе с дифференциальными уравнениями, в то время как Julia требует применения дополнительных библиотек.

Список литературы

https://cyberleninka.ru/article/n/model-lanchestera-kak-diskretnaya-upravlyaemaya-sistema/viewer

Кулябов Д. С. *Лабораторная работа №3*: https://esystem.rudn.ru/mod/resource/view.php?id=831037