# Отчёт по лабораторной работе №3

**Модель боевых действий**

Голощапова Ирина Борисовна

# Содержание

# Список иллюстраций

# Список таблиц

# 1 Цель работы

Разобраться в алгоритме построения математической модели. Рассмотреть некоторые простейшие модели боевых действий – модели Ланчестера.

# 2 Библиография


1. Git - система контроля версий
2. Дифференциальные уравнения
3. Язык программирования - Julia
4. Решение ДУ на языке программирование Julia
5. Установка и настройка OpenModelica

# 3 Задачи лабораторной работы

1. Изучить условие задачи о моделе боевых действий
2. Провести рассуждения и вывести дифференциальные уравнения
3. Построить математическую модель
4. Определить по графику, какая армия одержит победу

# 4 Теоретическая справка

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).

В данной лабораторной работе будет представлено два случая:

1. Боевые действия между регулярными войсками
2. Боевые действия с участием регулярных войск и партизанских отрядов

## 4.1 Первый случай

В первом случае численность регулярных войск определяется тремя факторами: - скорость уменьшения численности войск из-за причин, не связанных с боевыми действиями (болезни, травмы, дезертирство); - скорость потерь, обусловленных боевыми действиями противоборствующих сторон (что связанно с качеством стратегии, уровнем вооружения, профессионализмом солдат и т.п.); - скорость поступления подкрепления (задаётся некоторой функцией от времени).

В этом случае модель боевых действий между регулярными войсками описывается следующим образом (рис. 4.1):

$$\frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t)$$
$$\frac{dy}{dt} = -c(t)x(t) - h(t)y(t) + Q(t)$$

Рис. 4.1: ДУ для первого случая

Потери, не связанные с боевыми действиями, описывают члены -a(t)x(t), члены -b(t)y(t) и -c(t)x(t) отражают потери на поле боя.

Коэффициенты b(t) и c(t) указывают на эффективность боевых действий со стороны y и x соответственно, a(t), h(t) - величины, характеризующие степень влияния различных факторов на потери.

Функции P(t), Q(t) учитывают возможность подхода подкрепления к войскам X и У в течение одного дня.

## 4.2 Второй случай

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 тем 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 (рис. 4.2):

$$\frac{dx}{dt} = -a(t)x(t) - b(t)y(t) + P(t)$$
$$\frac{dy}{dt} = -c(t)x(t)y(t) - h(t)y(t) + Q(t)$$

Рис. 4.2: ДУ для второго случая

# 5 Условие задачи (вариант №7)

Между страной Х и страной У идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями x(t)и y(t). В начальный момент времени страна Х имеет армию численностью 24 000 человек, а в распоряжении страны У армия численностью в 9 500 человек. Для упрощения модели считаем, что коэффициенты a, b, c, h постоянны. Также считаем P(t) и Q(t) непрерывные функции.

Постройте графики изменения численности войск армии Х и армии У для следующих случаев:

1. Модель боевых действий между регулярными войсками (рис. 5.1)

$$\frac{dx}{dt} = -0,3x(t) - 0,87y(t) + \sin(2t) + 1$$
$$\frac{dy}{dt} = -0,5x(t) - 0,41y(t) + \cos(3t) + 1$$

Рис. 5.1: Вариант7_1

2. Модель ведение боевых действий с участием регулярных войск и партизанских отрядов (рис. 5.2)

$$\frac{dx}{dt} = -0,25x(t) - 0,64y(t) + \sin(2t+4)$$
$$\frac{dy}{dt} = -0,2x(t)y(t) - 0,52y(t) + \cos(t+4)$$

Рис. 5.2: Вариант7_2

# 6 Выполнение лабораторной работы

## 6.1 Реализация в OpenModelica. Случай №1

Для начала реализуем данную задачу в OpenModelica.

Листинг программы для первого случая (рис. 6.1):

```
//case 1
model lab3
Real a = 0.3;
Real b = 0.87;
Real c = 0.5;
Real h = 0.41;
Real x;
Real y;
initial equation
x = 24000;
y = 9500;
equation
der(x)=-a*x-b*y+sin(2*time)+1;
der(y)=-c*x-h*y+cos(3*time)+1;
end lab3;

```

Рис. 6.1: Листинг программы для 1-го случая. OpenModelica

В результате получим следующую модель (рис. 6.2):

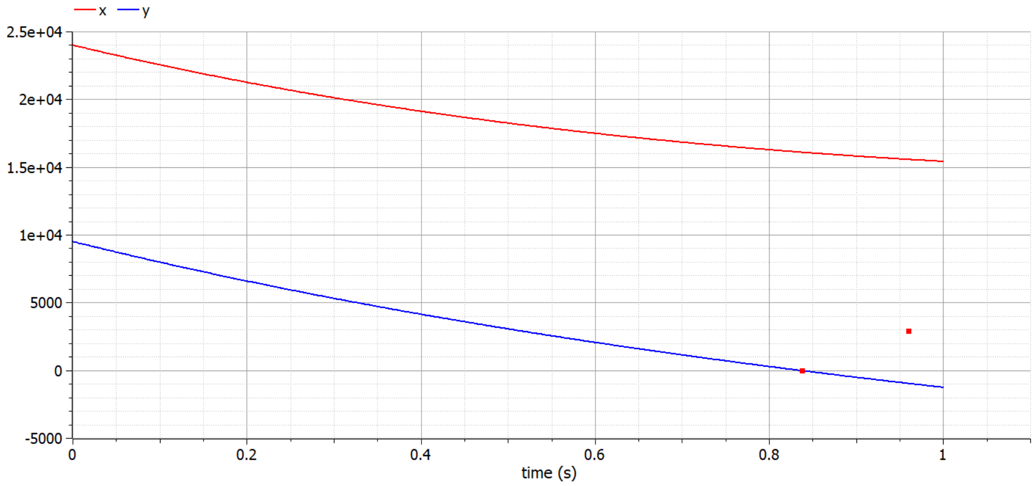


Рис. 6.2: Модель для 1-го случая. OpenModelica

Таким образом, график нам показывает, что одержит победу армия X.

## 6.2 Реализация в OpenModelica. Случай №2

Листинг программы для второго случая (рис. 6.3):

```
//case 2
model lab3_2
Real a = 0.25;
Real b = 0.64;
Real c = 0.2;
Real h = 0.52;
Real x;
Real y;
initial equation
x = 24000;
y = 9500;
equation
der(x) = -a*x-b*y+sin(2*time+4);
der(y) = -c*x*y-h*y+cos(time+4);
end lab3_2;
```

Рис. 6.3: Листинг программы для 2-го случая. OpenModelica

В результате получим следующую модель (рис. 6.4):

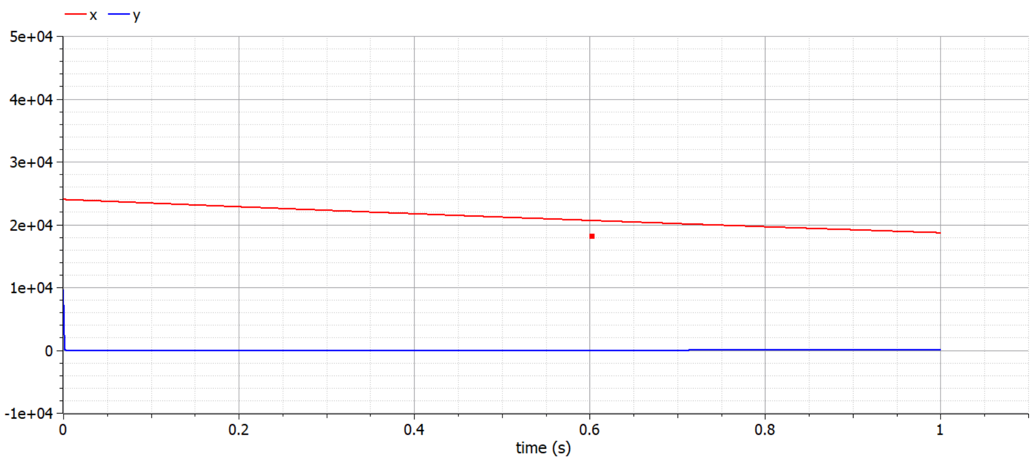


Рис. 6.4: Модель для 2-го случая. OpenModelica

На графике видно, что одержит победу армия X.

## 6.3 Реализация на Julia. Случай №1

Листинг программы для первого случая (рис. 6.5):

```julia
#case 1
using DifferentialEquations

function lorenz!(du, u, p, t)
    a,b,c,h = p
    du[1] = -a*u[1] - b*u[2] + sin(2*t) + 1
    du[2] = -c*u[1] - h*u[2] + cos(3*t) + 1
end

const x = 24000
const y = 9500
u0 = [x, y]

p = (0.3, 0.87, 0.5, 0.41)

tspan = (0.0, 1.0)
prob = ODEProblem(lorenz!, u0, tspan, p)
sol = solve(prob)

using Plots; gr()
plot(sol)

savefig("lab3_julia_1.png")
```

Рис. 6.5: Листинг программы для 1-го случая. Julia

В результате получим следующую модель (рис. 6.6):

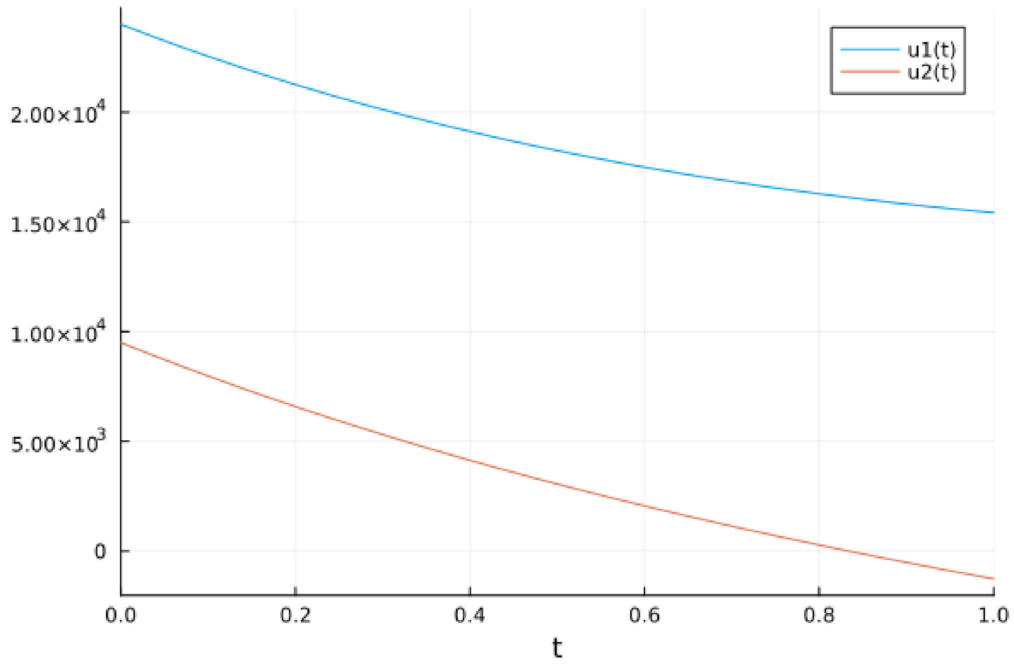


Рис. 6.6: Модель для 1-го случая. Julia

## 6.4 Реализация на Julia. Случай №2

Листинг программы для второго случая (рис. 6.7):

```
#case 2
using DifferentialEquations

function lorenz!(du, u, p, t)
    a,b,c,h = p
    du[1] = -a*u[1] - b*u[2] + sin(2*t+4)
    du[2] = -c*u[1]*u[2] - h*u[2] + cos(t+4)
end

const x = 24000
const y = 9500
u0 = [x, y]

p = (0.25, 0.64, 0.2, 0.52)

tspan = (0.0, 1.0)
prob = ODEProblem(lorenz!, u0, tspan, p)
sol = solve(prob)

using Plots; gr()
plot(sol)

savefig("lab3_julia_2.png")
```

Рис. 6.7: Листинг программы для 2-го случая. Julia

В результате получим следующую модель (рис. 6.7):

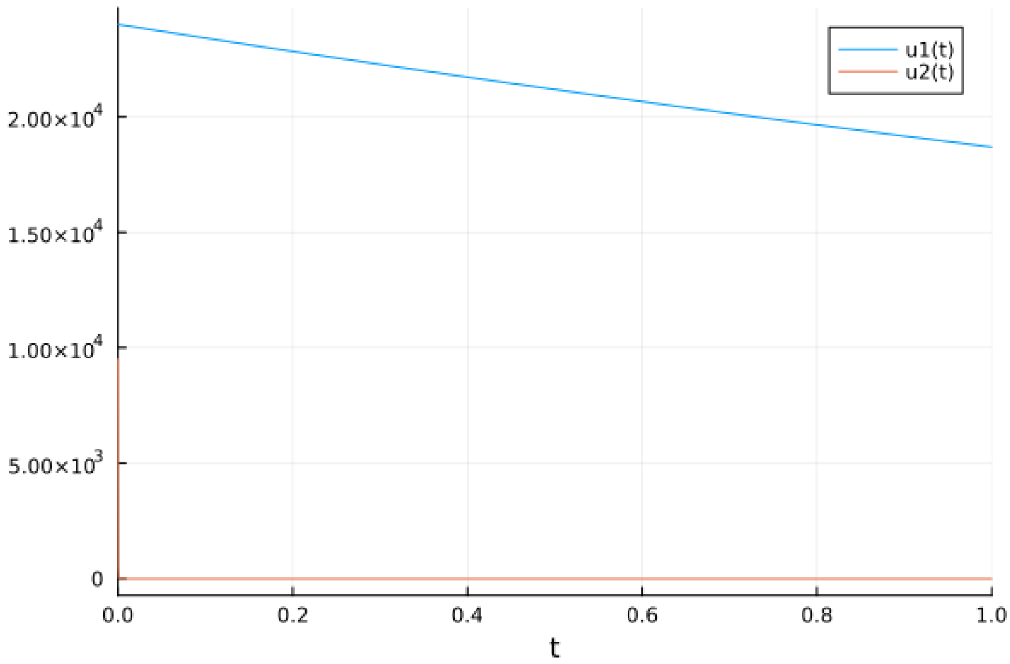


Рис. 6.8: Модель для 2-го случая. Julia

# 7 Выводы

В ходе лабораторной работы нам удалось построить модель боевых действий на двух языках программирования: OpenModelica и Julia, а также с помощью построенных графиков определить, какая из двух армий одержит победу.