

Отчёт по лабораторной работе №3

Модель боевых действий

Артамонов Тимофей Евгеньевич

Содержание

1	Цель работы	3
2	Теоретическое введение	4
3	Постановка задачи	6
4	Задание	7
5	Выполнение лабораторной работы	8
6	Выводы	13

1 Цель работы

- Рассмотреть модели Ланчестера и научиться их строить
- Написать аналог программы на языке OpenModelica

2 Теоретическое введение

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). Рассмотрим три случая ведения боевых действий: 1. Боевые действия между регулярными войсками 2. Боевые действия с участием регулярных войск и партизанских отрядов

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 связано 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

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

Потери, не связанные с боевыми действиями, описывают члены $a(t)x(t)$ и $h(t)y(t)$, члены $b(t)y(t)$ и $c(t)x(t)$ отражают потери на поле боя. Функции $P(t)$, $Q(t)$ учитывают возможность подхода подкрепления к войскам X и Y .

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 площа-

дям, 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

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

Рассмотренные простейшие модели соперничества соответствуют системам обыкновенных дифференциальных уравнений второго порядка, широко распространенным при описании многих естественно научных объектов.

3 Постановка задачи

Между страной X и страной Y идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями $x(t)$ и $y(t)$. В начальный момент времени страна X имеет армию численностью 150 000 человек, а в распоряжении страны Y армия численностью в 100 000 человек. Для упрощения модели считаем, что коэффициенты,

Таблица 3.1: Вариант 1

Коэффициенты при $x(t)$	Коэффициенты при $y(t)$
$a = 0.45$	$b = 0.85$
$c = 0.45$	$h = 0.45$

и функции $P(t) = \sin(t+8) + 1$ $Q(t) = \cos(t+8) + 1$

Таблица 3.2: Вариант 2

Коэффициенты при $x(t)$	Коэффициенты при $y(t)$
$a = 0.31$	$b = 0.79$
$c = 0.41$	$h = 0.32$

$P(t) = 2\sin(t)$ $Q(t) = 2\cos(t)$ - постоянны и непрерывны соответственно.

4 Задание

1. Постройте графики изменения численности войск армии X и армии У для Модели боевых действий между регулярными войсками
2. Постройте графики изменения численности войск армии X и армии У для Модели ведения боевых действий с участием регулярных войск и партизанских отрядов
3. Построить графики на языках Julia и OpenModelica
4. Сравнить результаты

5 Выполнение лабораторной работы

##Переписали код из примера лабораторной работы на язык Julia и посмотрели на результат работы программы. (рис. [5.1])

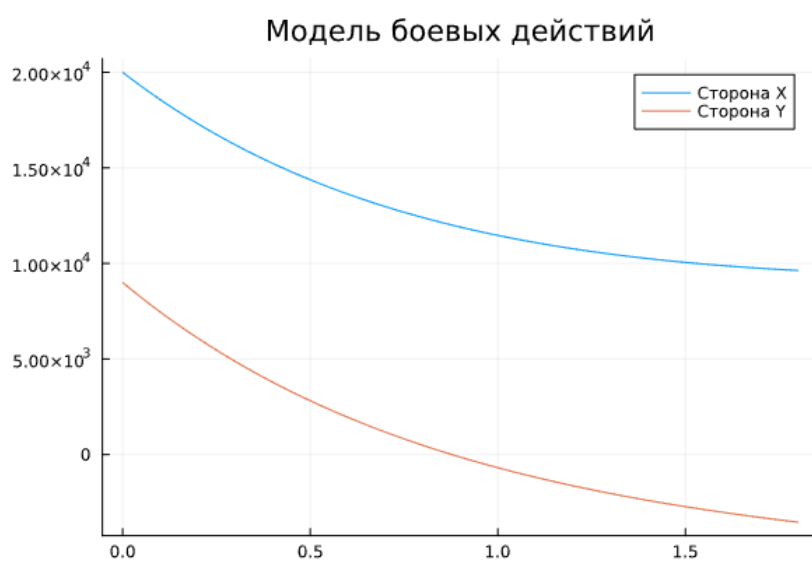


Рис. 5.1: Julia Plot 1

##Сделали то же самое для языка OpenModelica. (рис. [5.2])

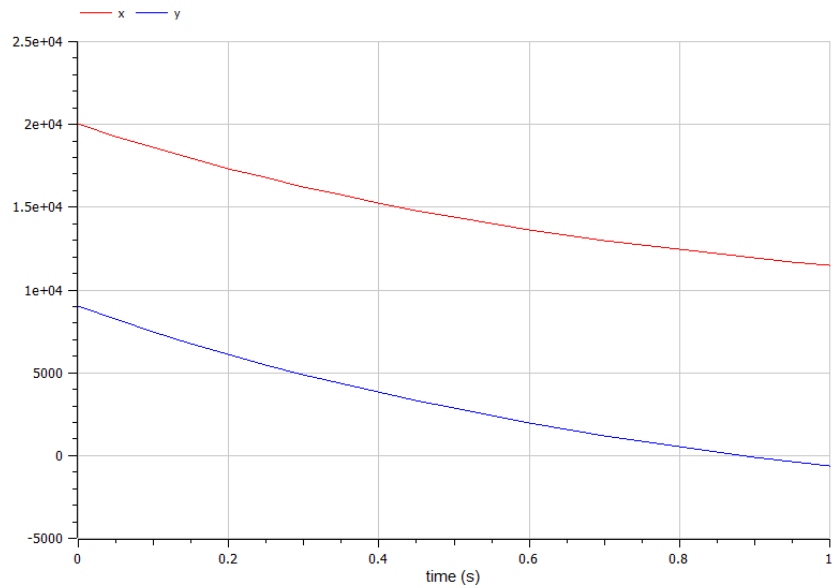


Рис. 5.2: OM Plot 1

##Подставили свои данные для случая с регулярными войсками для Julia. (рис. [5.3])

```

1  using DifferentialEquations, Plots
2  x0 = 150000
3  y0 = 100000
4
5  a = 0.45 # 45
6  b = 0.85 # 85
7  c = 0.45 # 45
8  h = 0.45 # 45
9
10
11
12  tspan = (0.0, 3)
13
14  function syst!(du, y, p, t)
15  du[1] = - a*y[1] - b*y[2] + sin(t+8) + 1 # sin(t+8)+ 1
16  du[2] = - c*y[1] - h*y[2] + cos(t+8) + 1 # cos(t+8)+ 1
17  end
18
19  v0 = [x0, y0]
20  problema = ODEProblem(syst!, v0, tspan, 0)
21
22  y = solve(problema)
23
24  u1 = Vector{Float64}()
25  u2 = Vector{Float64}()
26
27
28  for i in range(1, length(y.t))
29  push!(u1, y.u[i][1])
30  push!(u2, y.u[i][2])
31  end
32
33
34
35  plot(y.t, u1, label = "Сторона X", title = "Модель боевых действий")
36  plot!(y.t, u2, label = "Сторона Y")
37

```

Рис. 5.3: Julia code 1

##Нарисовали график для первого случая. (рис. [5.4])

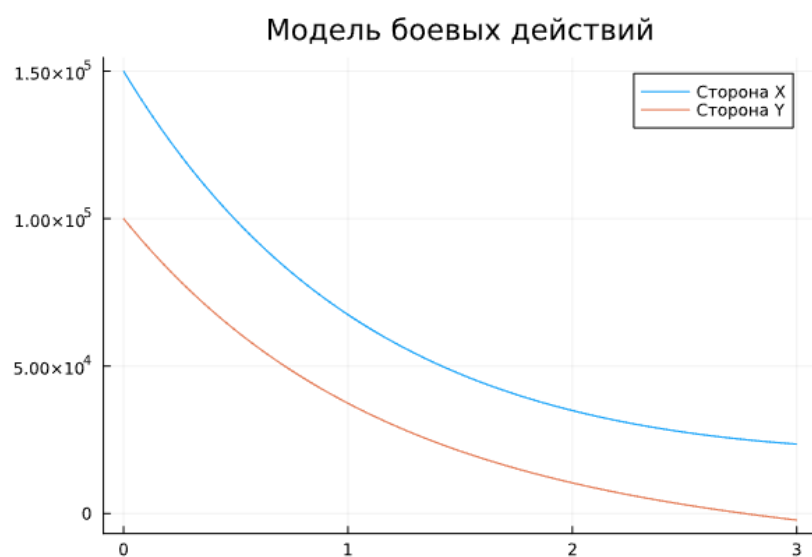


Рис. 5.4: Julia Plot 2

##Подставили свои значения для первого случая в OMEditб, в настройках модели задали промежуток и построили график. (рис. [5.5])

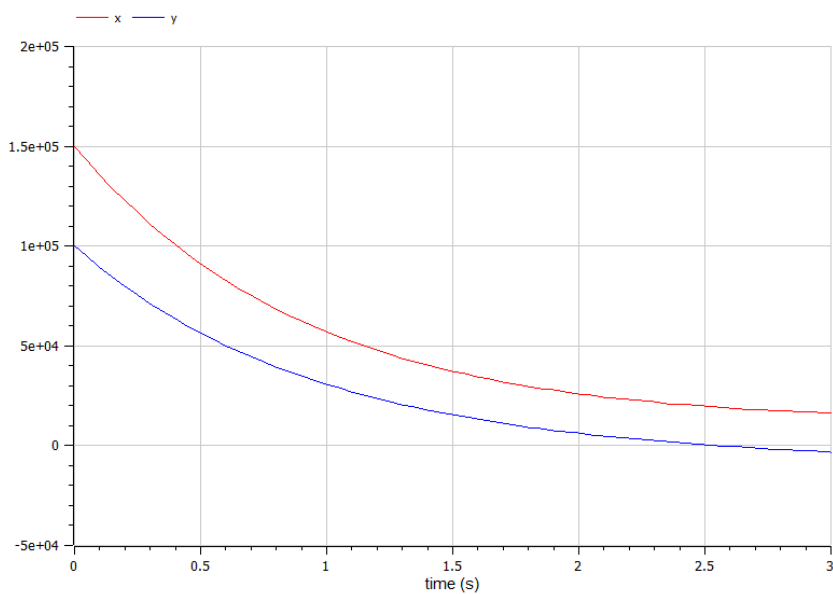


Рис. 5.5: OM Plot 2

##Подставили значения для второго случая и изменили функцию в Julia. (рис. [5.6])

```

1 using DifferentialEquations, Plots
2 x0 = 150000
3 y0 = 100000
4
5 a = 0.31 # 45
6 b = 0.79 # 85
7 c = 0.41 # 45
8 h = 0.32 # 45
9
10
11
12 tspan = (0.0, 1)
13
14 function syst!(du, y, p, t)
15     du[1] = - a*y[1] - b*y[2] + 2*sin(t) # sin(t+8)+1
16     du[2] = - c*y[1]*y[2] - h*y[2] + 2*cos(t) # cos(t+8)+1
17 end
18
19 v0 = [x0, y0]
20 problema = ODEProblem(syst!, v0, tspan, p)
21
22 y = solve(problema)
23
24 u1 = Vector{Float64}()
25 u2 = Vector{Float64}()
26
27
28 for i in range(1, length(y.t))
29     push!(u1, y.u[i][1])
30     push!(u2, y.u[i][2])
31 end
32
33
34
35 plot(y.t, u1, label = "Сторона X", title = "Модель боевых действий")
36 plot!(y.t, u2, label = "Сторона Y", xlim = [0, 0.0001])
37

```

Рис. 5.6: Julia code 2

##Построили график и приблизили так, чтобы хорошо было видно изменение функции, понадобилось приблизить до интервала (0, 0.0001). (рис. [5.7]) Видно что функция немного негладкая.

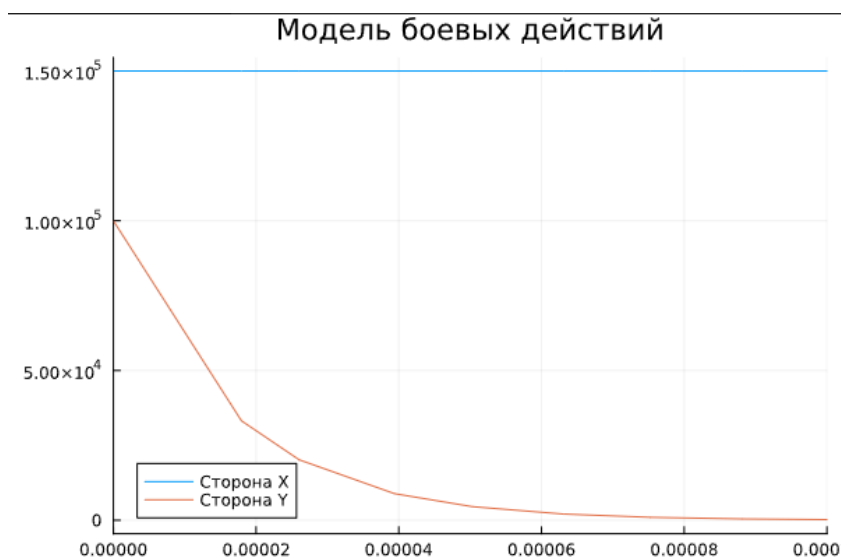


Рис. 5.7: Julia Plot 3

##Сделаем то же самое в OpenModelica. (рис. [5.8])

```
1  model lab3
2
3  Real x(start = 150000);
4  Real y(start = 100000);
5
6  Real p;
7  Real q;
8
9  parameter Real a = 0.31;
10 parameter Real b = 0.79;
11 parameter Real c = 0.41;
12 parameter Real h = 0.32;
13
14 equation
15   der(x) = - a*x - b*y + p;
16   der(y) = - c*x*y - h*y + q;
17   p = 2*sin(time);
18   q = 2*cos(time);
19
20 end lab3;
```

Рис. 5.8: OMEdit 1

##Посмотрим на график, пришлось сделать разбиение на 500 отрезков интервала (0, 0.001). (рис. [5.9])

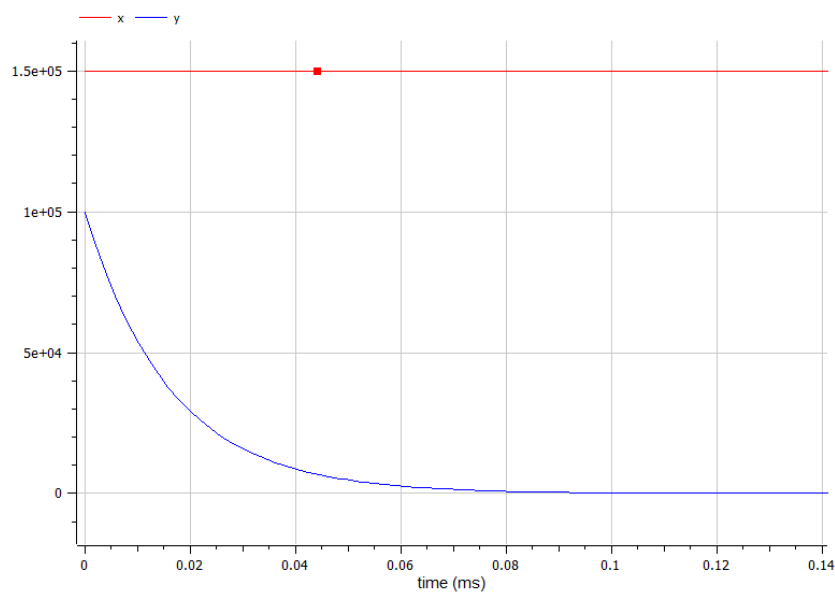


Рис. 5.9: OM Plot 3

6 Выводы

Построили модели боевых действия для 2 случаев на 2 языках. Выяснили, что на OpenModelica графики точнее. Познакомились и научились строить модели боевых действий.