Отчёт по лабораторной работе

Лабораторная работа №3

Серегин Денис Алексеевич

Содержание

# 1 Цель работы

При помощи Julia и Openmodelica построить графики изменения численности войск армии X и армии Y для двух случаев.

# 2 Задание

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 50,000 человек, а в распоряжении страны У армия численностью в 69,000 человек. Для упрощения модели считаем, что коэффициенты a,b,c,h постоянны. Также считаем P(t) и Q(t) непрерывные функции.

Модель боевых действий между регулярными войсками:

Модель ведение боевых действий с участием регулярных войск и партизанских отрядов :

# 3 Теоретическое введение

## 3.1 Формулировка задачи

Между страной Х и страной У идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями x(t) и y(t). Для упрощения модели считаем, что коэффициенты a,b,c,h постоянны. Также считаем P(t) и Q(t) непрерывные функции. Рассмотрим 2 модели боя.

## 3.2 Модель боевых действий между регулярными войсками

## 3.3 Модель ведение боевых действий с участием регулярных войск и партизанских отрядов

Подробнее в [1].

# 4 Выполнение лабораторной работы

## 4.1 Первая модель

Я написал программу на языке Julia для симуляции модели боевых действий с регулярной армией

begin  
 import Pkg  
 Pkg.activate()  
 using DifferentialEquations  
 using LaTeXStrings  
 import Plots  
end

function F!(du, u, p, t)  
 du[1] = -0.34u[1] - 0.72u[2] + sin(t+10)  
 du[2] = -0.89u[1] - 0.43u[2] + cos(t+20)  
end

begin  
 u0 = [50000.0, 69000.0]  
 T = (0.0, 2.0)  
 prob = ODEProblem(F!, u0, T)  
end

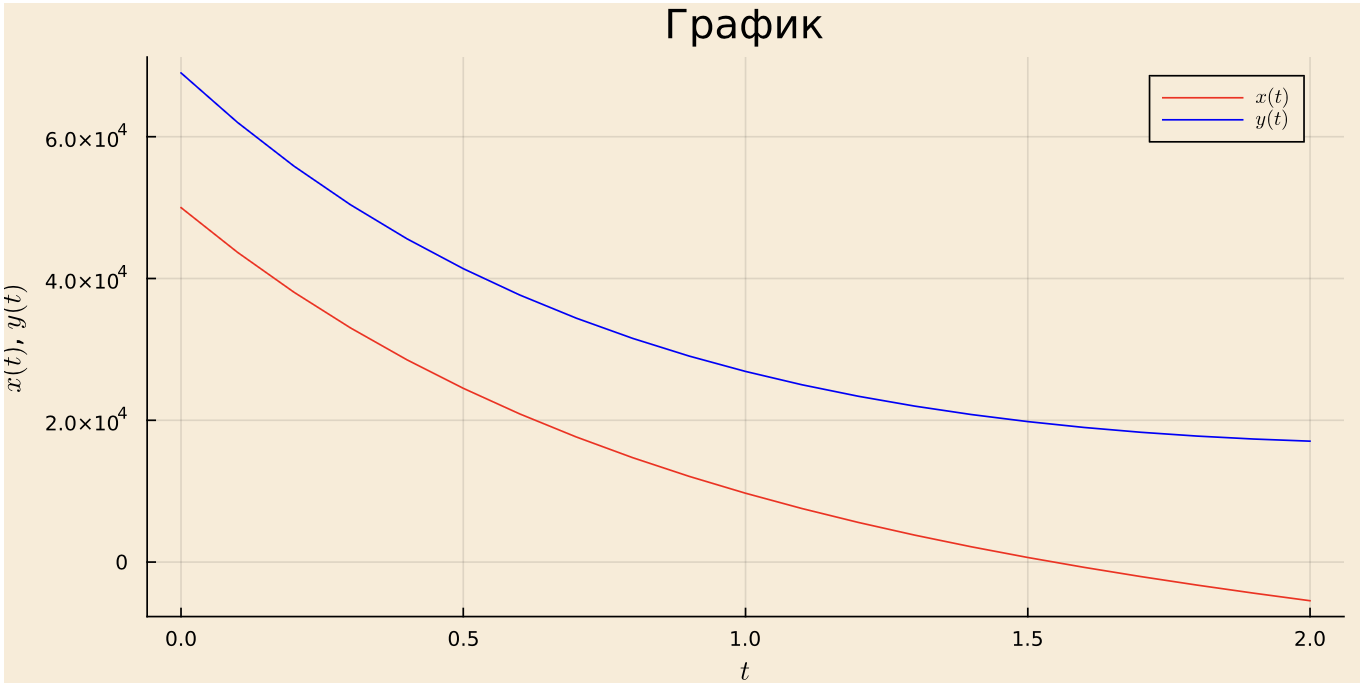
sol = solve(prob, saveat=0.1)

begin  
 Time = sol.t  
 const X = Float64[]  
 const Y = Float64[]  
 for u in sol.u  
 x, y = u  
 push!(X, x)  
 push!(Y, y)  
 end  
 X, Y  
end

begin  
 fig = Plots.plot(  
 layout=(1),  
 dpi=150,  
 grid=:xy,  
 gridcolor=:black,  
 gridwidth=1,  
 background\_color=:antiquewhite,  
 # aspect\_ratio=:equal,  
 size=(800, 400),  
 plot\_title="График",  
 )  
 # )  
 Plots.plot!(  
 fig[1],  
 Time,  
 [X Y],  
 xlabel=L"$t$",  
 ylabel=L"$x(t)$, $y(t)$",  
 color=[ :red :blue ],  
 label=[L"$x(t)$" L"$y(t)$"]  
 )  
  
end

## 4.2 График

В результате работы кода получился график. (рис. ??)



Полученный график

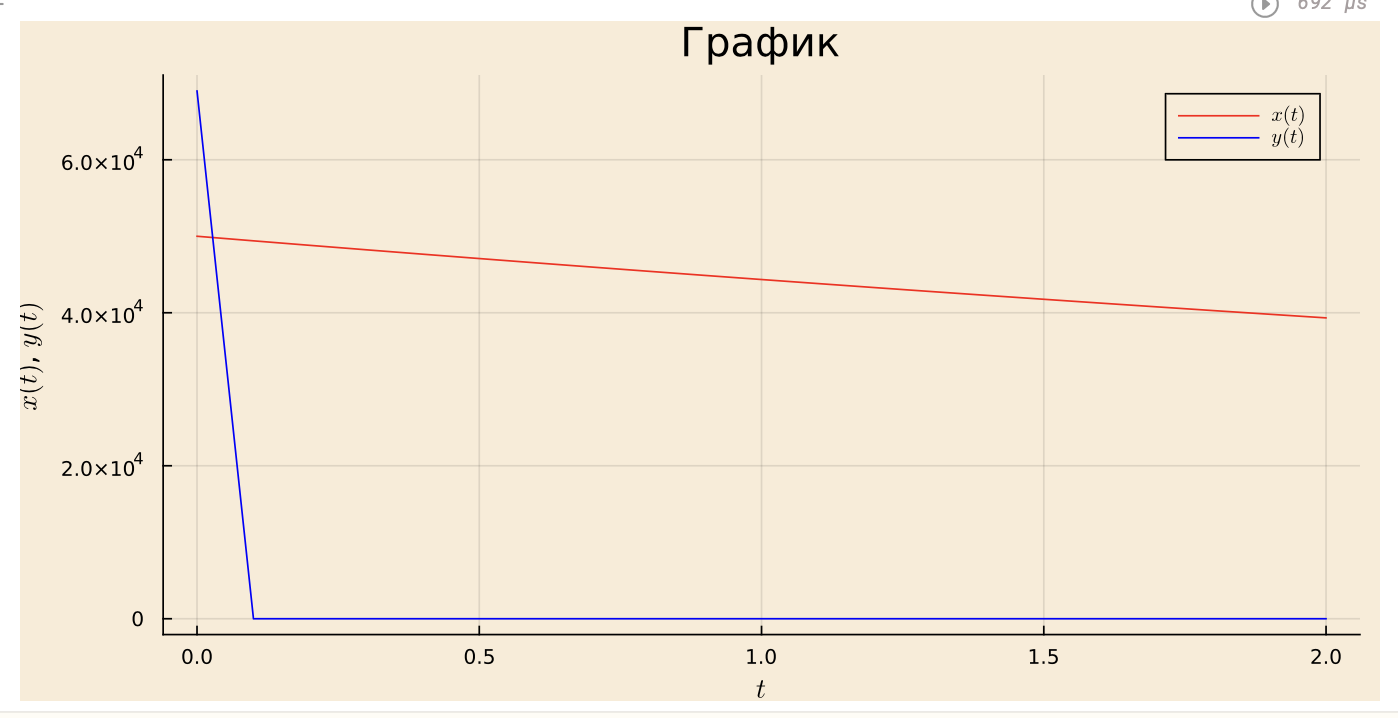
## 4.3 Вторая модель

Далее я поменял значения в уравнении чтобы получить вторую модель боевых действий с участием партизанских отрядов.

function F!(du, u, p, t)  
 du[1] = -0.12u[1] - 0.51u[2] + sin(20t)  
 du[2] = -0.3u[1]u[2] - 0.61u[2] + cos(13t)  
end

## 4.4 График

В результате работы кода получился следующий график. (рис. ??)



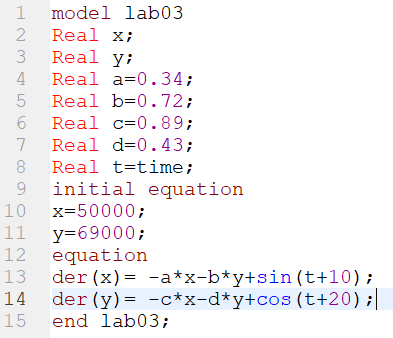
Полученный график

## 4.5 Выполнение в OpenModelica

Далее я создал модель в приложении “openmodelica connection editor”

## 4.6 Первая модель

Я создал класс модели, задал параметры уравнения. (рис. ??)

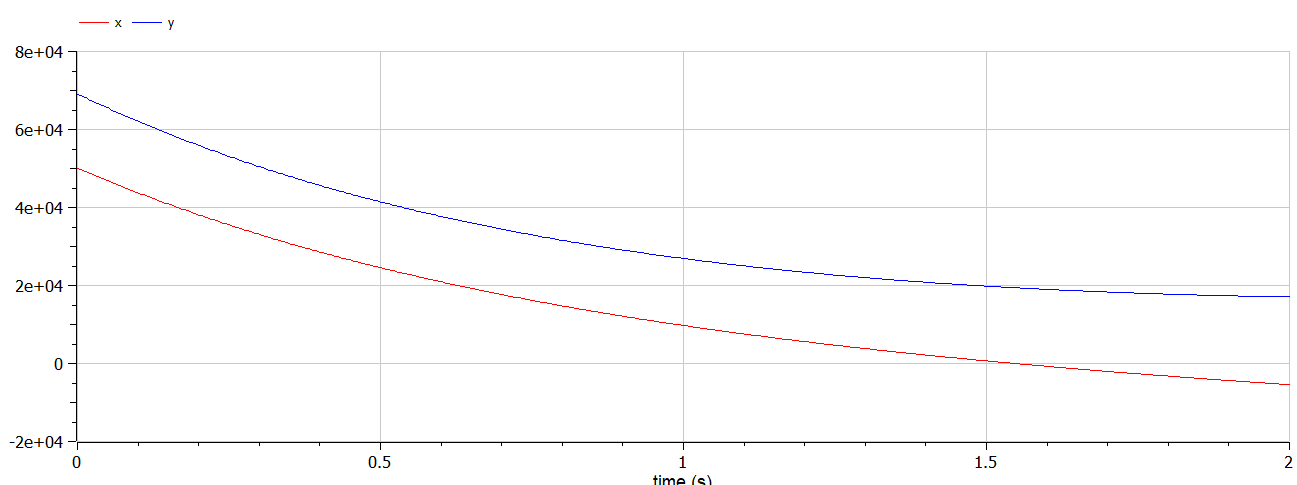


Листинг программы

## 4.7 График

В результате работы программы получился график симуляции.

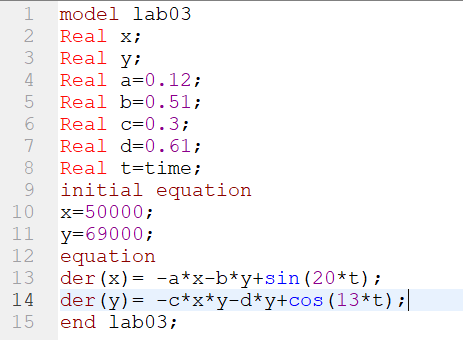
(рис. ??)



Полученный график

## 4.8 Вторая модель

Далее я исправил параметры под вторую модель. (рис. ??)

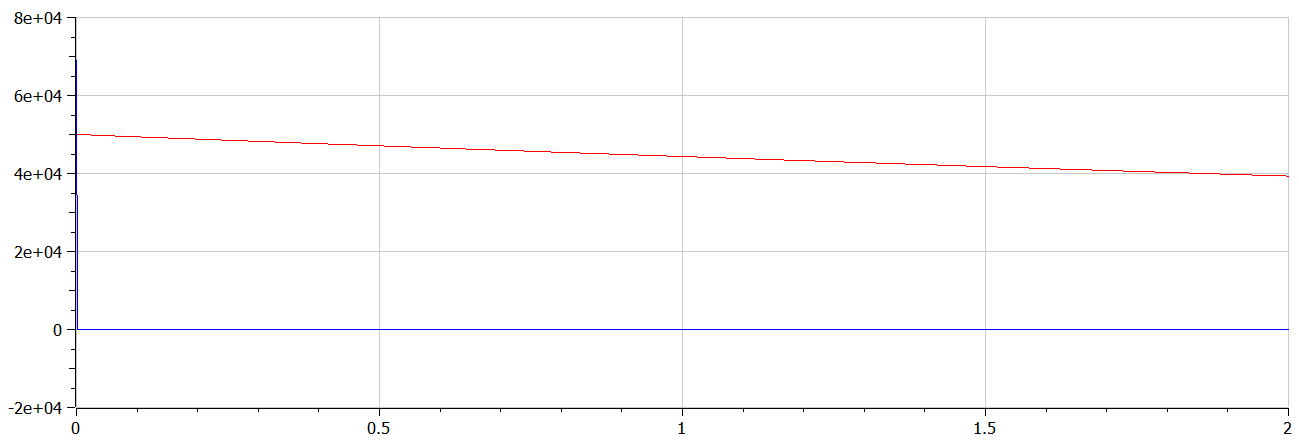


Листинг программы

## 4.9 График

В результате работы программы получился график симуляции.

(рис. ??)



Полученный график

# 5 Выводы

В результате работы я убедился в том, что результат получился одинаковый в обоих средах разработки. А также смог решить задачу и построить модели боевых действий в Julia и OpenModelica.

# Список литературы

1. Кулябов. Инструкция к лабораторной работе№3 [Электронный ресурс]. RUDN, 2023. 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>.