

Лабораторная работа №5

“Низамова Альфия Айдаровна. НФИбд-01-20”¹

4 марта, 2023, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель работы:

Целью данной лабораторной работы является ознакомление с моделью Лотки-Вольтерры и построение его на языках программирования Julia и OpenModelica

Задачи:

Задачи:

1. Изучить условие задачи
2. Написать код на языке программирования Julia и найти стационарное состояние
3. Написать код на языке программирования OpenModelica и найти стационарное состояние
4. Изучить полученные графики

Ход работы лабораторной работы

Условие задачи:

Задание(рис.1)

Вариант 2

Для модели «хищник-жертва»:

$$\begin{cases} \frac{dx}{dt} = -0.13x(t) + 0.042x(t)y(t) \\ \frac{dy}{dt} = 0.33y(t) - 0.03x(t)y(t) \end{cases}$$

Постройте график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях: $x_0 = 7$, $y_0 = 12$. Найдите стационарное состояние системы.

Написала код на языке Julia и Openmodelica для построения графика зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях: $x_0 = 7$, $y_0 = 12$

```
1 using Plots
2 using DifferentialEquations
3
4 function ode_fn(du, u, p, t)
5     a, b, c, d = p
6     du[1] = -a*u[1] + b * u[1] * u[2]
7     du[2] = c * u[2] - d * u[1] * u[2]
8 end
9
10 x0 = 7
11 y0 = 12
12
13 p = (0.13, 0.042, 0.33, 0.03)
14 u0 = [x0, y0]
15 tspan = (0.0, 400.0)
16 prob = ODEProblem(ode_fn, u0, tspan, p)
```

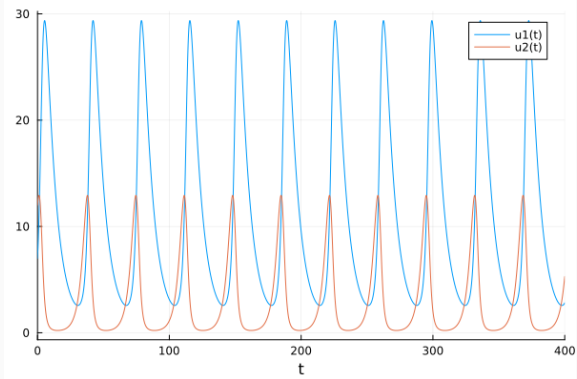
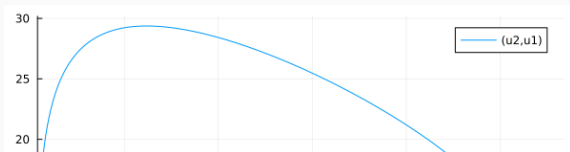



Рис. 1: Результат Julia



Стационарное состояние на Julia

```
1 using Plots
2 using DifferentialEquations
3
4 function ode_fn(du, u, p, t)
5     a, b, c, d = p
6     du[1] = -a*u[1] + b * u[1] * u[2]
7     du[2] = c * u[2] - d * u[1] * u[2]
8 end
9
10 x0 = 0.33 / 0.03
11 y0 = 0.13 / 0.042
12
13 p = (0.13, 0.042, 0.33, 0.03)
14 u0 = [x0, y0]
15 tspan = (0.0, 400.0)
16 prob = ODEProblem(ode_fn, u0, tspan, p)
17 sol = solve(prob, dtmax=0.05)
18
19 #решение системы уравнений
20 plot(sol)
21 savefig("lab5_jl_1.png")
22
23 # #фазовый портрет
24 plot(sol, vars=(2,1))
```

Написала код на языке OpenModelica Код на

```
1 model lab5
2
3 Real a = 0.13;
4 Real b = 0.042;
5 Real c = 0.33;
6 Real d = 0.03;
7
8 Real x;
9 Real y;
10
11 initial equation
12 x = 7;
13 y = 12;
14
15 equation
16 der(x) = -a*x + b*x*y;
17 der(y) = c*y - d*x*y;
18
```

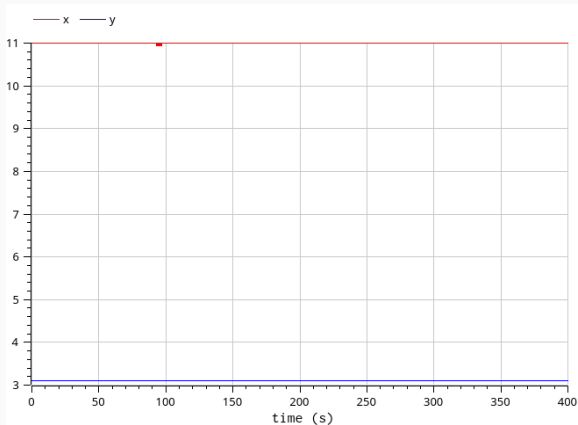
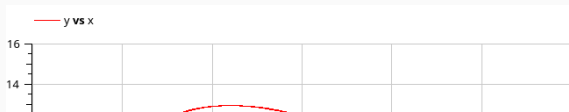


Рис. 4: Результат OpenModelica



Стационарное состояние на OpenModelica

```
1 model lab5
2
3 Real a = 0.13;
4 Real b = 0.042;
5 Real c = 0.33;
6 Real d = 0.03;
7
8 Real x;
9 Real y;
10
11 initial equation
12 x = c / d;
13 y = a / b;
14
15 equation
16 der(x) = -a*x + b*x*y;
17 der(y) = c*y - d*x*y;
18
```

Выводы

Мы познакомились с моделью Лотки-Вольтерры и построили его на языках программирования Julia и OpenModelica