

# **Лабораторная работа №5**

**Модель хищник-жертва**

Ильин Андрей Владимирович

# Содержание

1	Цель работы	4
2	Задачи	5
3	Среда	6
4	Теоретическое введение	7
5	Выполнение лабораторной работы	9
6	Анализ результатов	22
7	Выводы	23
	Список литературы	24

## Список иллюстраций

5.1	Julia. Запуск Pluto . . . . .	9
5.2	Julia. Скрипт (1). Модель “Хищник-жертва” (при начальных условиях $x = 7, y = 29$ ) . . . . .	10
5.3	Julia. Скрипт (2). Модель “Хищник-жертва” (при начальных условиях $x = 7, y = 29$ ) . . . . .	12
5.4	Julia. Модель. Графики функций изменения численности хищников и изменения численности жертв (при начальных условиях $x = 7, y = 29$ ) . . . . .	13
5.5	Julia. Модель. График зависимости изменения численности хищников от изменения численности жертв (при начальных условиях $x = 7, y = 29$ ) . . . . .	14
5.6	Julia. Скрипт. Модель “Хищник-жертва” (стационарное состояние)	15
5.7	Julia. Модель. Графики функций изменения численности хищников и изменения численности жертв (стационарное состояние) . . . .	16
5.8	Julia. Модель. График зависимости изменения численности хищников от изменения численности жертв (стационарное состояние)	17
5.9	Modelica. Скрипт. Модель “Хищник-жертва” (при начальных условиях $x = 7, y = 29$ ) . . . . .	18
5.10	Modelica. Модель. Графики функций изменения численности хищников и изменения численности жертв (при начальных условиях $x = 7, y = 29$ ) . . . . .	18
5.11	Modelica. Модель. График зависимости изменения численности хищников от изменения численности жертв (при начальных условиях $x = 7, y = 29$ ) . . . . .	19
5.12	Modelica. Скрипт. Модель “Хищник-жертва” (стационарное состояние) . . . . .	20
5.13	Modelica. Модель. Графики функций изменения численности хищников и изменения численности жертв (стационарное состояние)	20
5.14	Modelica. Модель. График зависимости изменения численности хищников от изменения численности жертв (стационарное состояние) . . . . .	21

# 1 Цель работы

Рассмотреть модель хищник-жертва. Построить вышеуказанную модель средствами OpenModelica и Julia.

## 2 Задачи

Для модели «хищник-жертва»:

$$\begin{cases} \frac{dx}{dt} = -0.81x(t) + 0.048x(t)y(t) \\ \frac{dy}{dt} = 0.76y(t) - 0.038x(t)y(t) \end{cases}$$

1. Построить график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях:  $x_0 = 7, y_0 = 29$ .
2. Найти стационарное состояние системы.

## 3 Среда

- Julia – это открытый свободный высокопроизводительный динамический язык высокого уровня, созданный специально для технических (математических) вычислений. Его синтаксис близок к синтаксису других сред технических вычислений, таких как Matlab и Octave. [1]
- OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. [2]

## 4 Теоретическое введение

Простейшая модель взаимодействия двух видов типа «хищник — жертва» - модель Лотки-Вольтерры. Данная двухвидовая модель основывается на следующих предположениях [3]:

1. Численность популяции жертв  $x$  и хищников  $y$  зависят только от времени (модель не учитывает пространственное распределение популяции на занимаемой территории);
2. В отсутствии взаимодействия численность видов изменяется по модели Мальтуса, при этом число жертв увеличивается, а число хищников падает;
3. Естественная смертность жертвы и естественная рождаемость хищника считаются несущественными;
4. Эффект насыщения численности обеих популяций не учитывается;
5. Скорость роста численности жертв уменьшается пропорционально численности хищников;

$$\begin{cases} \frac{dx}{dt} = ax(t) - bx(t)y(t) \\ \frac{dy}{dt} = -cy(t) + dx(t)y(t) \end{cases}$$

В этой модели  $x$  – число жертв,  $y$  - число хищников. Коэффициент  $a$  описывает скорость естественного прироста числа жертв в отсутствие хищников, - естественное вымирание хищников, лишенных пищи в виде жертв. Вероятность взаимодействия жертвы и хищника считается пропорциональной как количеству жертв, так и числу самих хищников ( $xy$ ). Каждый акт взаимодействия уменьшает

популяцию жертв, но способствует увеличению популяции хищников (члены  $-bxy$  и  $dxu$  в правой части уравнения).

Стационарное состояние данной системы (положение равновесия, не зависящее от времени решение) будет в точке:

$$x_0 = \frac{c}{d}, y_0 = \frac{a}{b}$$

Если начальные значения задать в стационарном состоянии

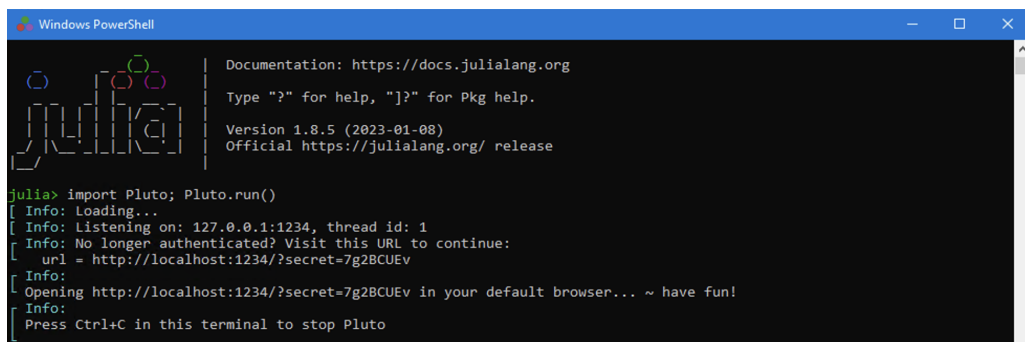
$$x(0) = x_0, y(0) = y_0$$

то в любой момент времени численность популяций изменяться не будет. При малом отклонении от положения равновесия численности как хищника, так и жертвы с течением времени не возвращаются к равновесным значениям, а совершают периодические колебания вокруг стационарной точки. Амплитуда колебаний и их период определяется начальными значениями численностей  $x(0), y(0)$ . Колебания совершаются в противофазе.



## 5 Выполнение лабораторной работы

1. Начнем выполнения поставленных задач в Julia. Для этого запустим Pluto [4]. (рис. 5.1)



```
Windows PowerShell

Documentation: https://docs.julialang.org
Type "?" for help, "]" for Pkg help.
Version 1.8.5 (2023-01-08)
Official https://julialang.org/ release

julia> import Pluto; Pluto.run()
[ Info: Loading...
[ Info: Listening on: 127.0.0.1:1234, thread id: 1
[ Info: No longer authenticated? Visit this URL to continue:
      url = http://localhost:1234/?secret=7g2BCUEv
[ Info:
[ Info: Opening http://localhost:1234/?secret=7g2BCUEv in your default browser... ~ have fun!
[ Info:
[ Info: Press Ctrl+C in this terminal to stop Pluto
```

Рис. 5.1: Julia. Запуск Pluto

2. Первым делом подключим пакеты “Plots” [5] и “DifferentialEquations” [6]. Далее объявим начальные данные при помощи констант. Также объявим начальное условие для системы ДУ и промежуток времени, на котором будет проходить моделирование. После этого объявим функцию, представляющую систему. (рис. 5.2)

# подключение пакетов

**using** Plots

**using** DifferentialEquations

# входные данные

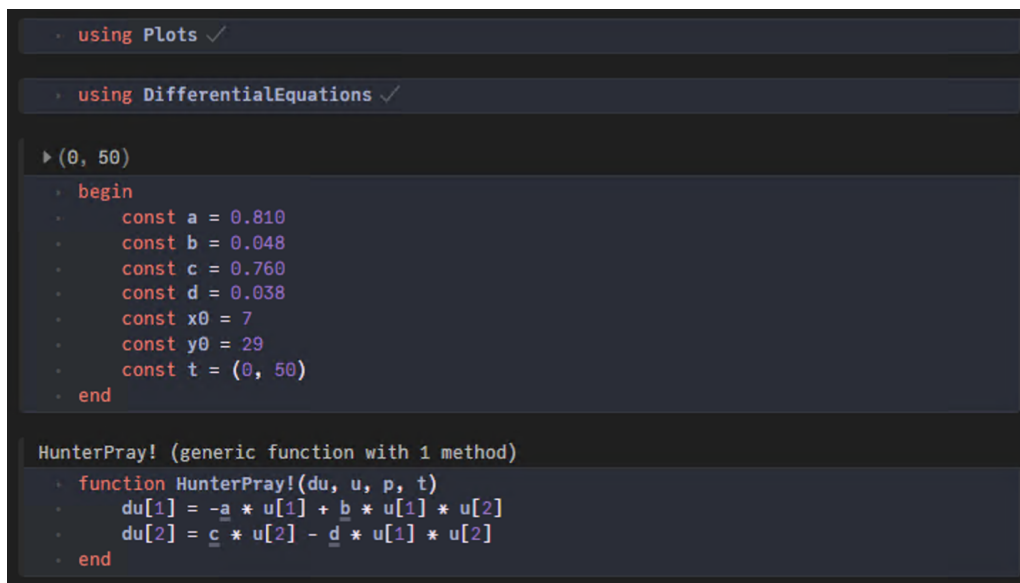
```

const a = 0.810
const b = 0.048
const c = 0.760
const d = 0.038
const x0 = 7
const y0 = 29

const t = (0, 50) # промежуток времени

function HunterPray!(du, u, p, t)
    du[1] = -a * u[1] + b * u[1] * u[2]
    du[2] = c * u[2] - d * u[1] * u[2]
end

```



```

> using Plots ✓
> using DifferentialEquations ✓

> p(0, 50)
begin
  const a = 0.810
  const b = 0.048
  const c = 0.760
  const d = 0.038
  const x0 = 7
  const y0 = 29
  const t = (0, 50)
end

HunterPray! (generic function with 1 method)
function HunterPray!(du, u, p, t)
  du[1] = -a * u[1] + b * u[1] * u[2]
  du[2] = c * u[2] - d * u[1] * u[2]
end

```

Рис. 5.2: Julia. Скрипт (1). Модель “Хищник-жертва” (при начальных условиях  $x = 7, y = 29$ )

3. В следующей ячейке Pluto построим график зависимости  $x$  от  $y$  и графики функций  $x(t), y(t)$ . При помощи ‘DifferentialEquations’ зададим и решим си-

стему ДУ, после чего построим графики функций  $x(t)$ ,  $y(t)$ . Так же создадим два списка, в которых будут храниться точки уравнений. Воспользуемся данным списком, чтобы построить график зависимости  $x$  от  $y$ . Далее запустим итоговый скрипт. (рис. 5.3, 5.4, 5.5)

```
begin
```

```
u0 = [x0, y0] # начальные условия
prob = ODEProblem(HunterPrey!, u0, t)
sol = solve(prob, dtmax=0.05)
```

```
X = [u[1] for u in sol.u]
Y = [u[2] for u in sol.u]
```

```
# используем Plots,
```

```
# чтобы построить графики
```

```
plt01 = plot(sol,
             dpi=500,
             plot_title = "Хищник-жертва",
             xlabel="Время (s)",
             ylabel="x, y",
             label=["x - хищник" "y - жертва"])
savefig(plt01, "artifacts/JL.lab05_01-1.png")
```

```
plt02 = plot(X, Y,
             dpi=500,
             plot_title = "Хищник-жертва",
             xlabel="x",
             ylabel="y",
             label="Зависимость x (хищник) от y (жертва)")
savefig(plt02, "artifacts/JL.lab05_01-2.png")
```

```
println("Success!")
```

```
begin
    u0 = [x0, y0]
    prob = ODEProblem(HunterPrey!, u0, t)
    sol = solve(prob, dtmax=0.05)

    x = [u[1] for u in sol.u]
    y = [u[2] for u in sol.u]

    plt01 = plot(sol,
        dpi=500,
        plot_title = "Хищник-жертва",
        xlabel="Время (s)",
        ylabel="x, y",
        label=["x - хищники" "y - жертва"])
    savefig(plt01, "artifacts/JL.lab05_01-1.png")

    plt02 = plot(x, y,
        dpi=500,
        plot_title = "Хищник-жертва",
        xlabel="x",
        ylabel="y",
        label="Зависимость x (хищники) от y (жертва)")
    savefig(plt02, "artifacts/JL.lab05_01-2.png")

    println("Success!")
end
```

Success!

Рис. 5.3: Julia. Скрипт (2). Модель “Хищник-жертва” (при начальных условиях  $x = 7, y = 29$ )

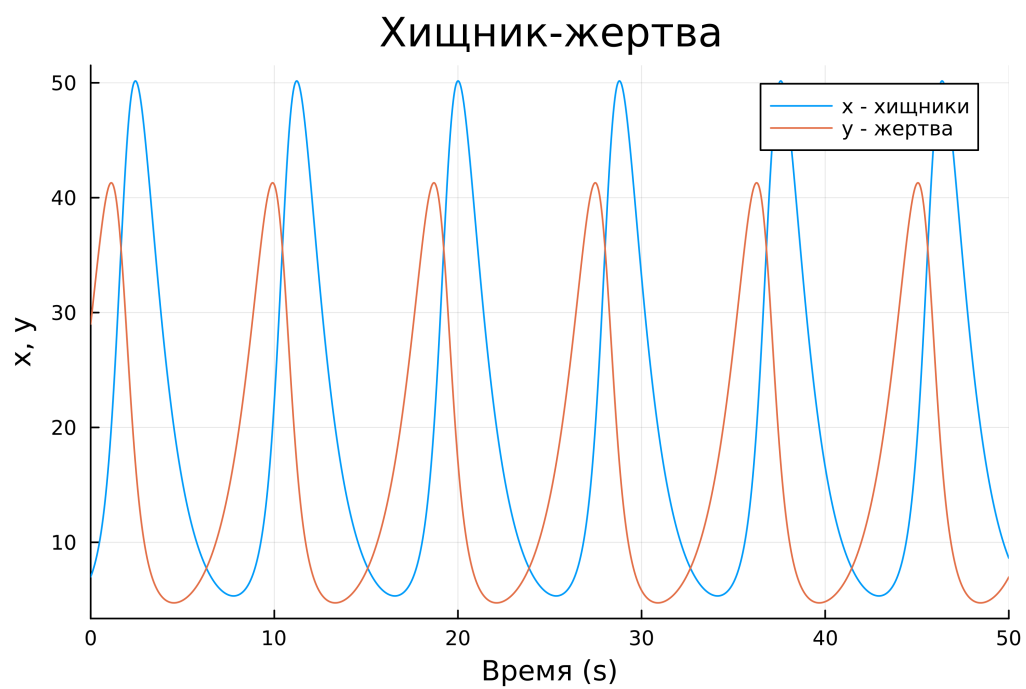


Рис. 5.4: Julia. Модель. Графики функций изменения численности хищников и изменения численности жертв (при начальных условиях  $x = 7$ ,  $y = 29$ )

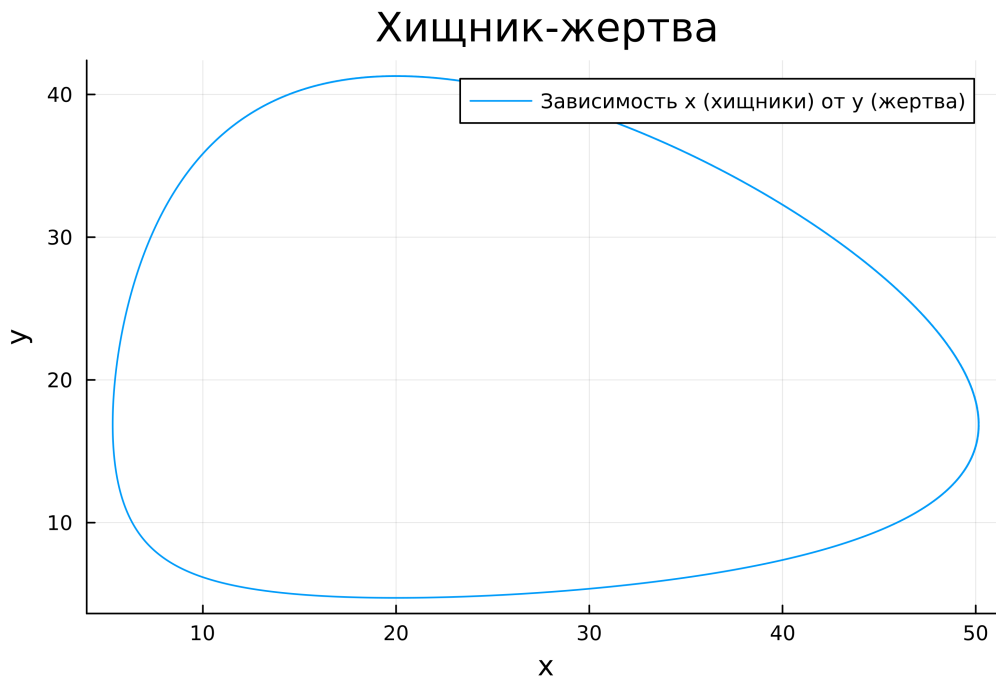


Рис. 5.5: Julia. Модель. График зависимости изменения численности хищников от изменения численности жертв (при начальных условиях  $x = 7$ ,  $y = 29$ )

4. Изменим начальные значения в вышеуказанном скрипте, при которых будет достигаться положение равновесия (не зависящее от времени решение). (рис. 5.6, 5.7, 5.8)

```
u0 = [c/d, a/b] #!
prob = ODEProblem(HunterPrey!, u0, t)
sol = solve(prob, dtmax=0.05)
```

```
X = [u[1] for u in sol.u]
Y = [u[2] for u in sol.u]
```

```
plt01 = plot(sol,
    dpi=500,
    plot_title = "Хищник-жертва",
    xlabel="Время (s)",
```

```

        ylabel="x, y",
        label=["x - хищники" "y - жертва"]])
savefig(plt01, "artifacts/JL.lab05_02-1.png")

plt02 = scatter(X, Y,
                dpi=500,
                plot_title = "Хищник-жертва",
                xlabel="x",
                ylabel="y",
                label="Зависимость x (хищники) от y (жертва)")
savefig(plt02, "artifacts/JL.lab05_02-2.png")

println("Success!")

```



```

begin
    u0 = [c/d, a/b]
    prob = ODEProblem(HunterPrey!, u0, t)
    sol = solve(prob, dtmax=0.05)

    X = [u[1] for u in sol.u]
    Y = [u[2] for u in sol.u]

    plt01 = plot(sol,
                dpi=500,
                plot_title = "Хищник-жертва",
                xlabel="Время (s)",
                ylabel="x, y",
                label=["x - хищники" "y - жертва"]])
    savefig(plt01, "artifacts/JL.lab05_02-1.png")

    plt02 = scatter(X, Y,
                    dpi=500,
                    plot_title = "Хищник-жертва",
                    xlabel="x",
                    ylabel="y",
                    label="Зависимость x (хищники) от y (жертва)")
    savefig(plt02, "artifacts/JL.lab05_02-2.png")

    println("Success!")
end

```

Success!

Рис. 5.6: Julia. Скрипт. Модель “Хищник-жертва” (стационарное состояние)

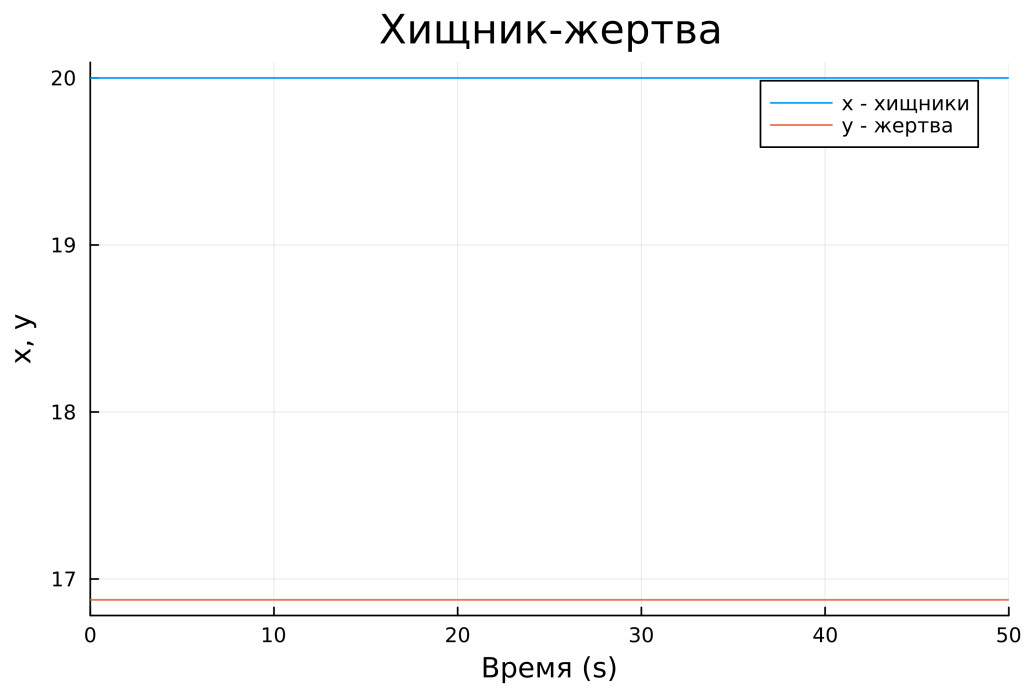


Рис. 5.7: Julia. Модель. Графики функций изменения численности хищников и изменения численности жертв (стационарное состояние)



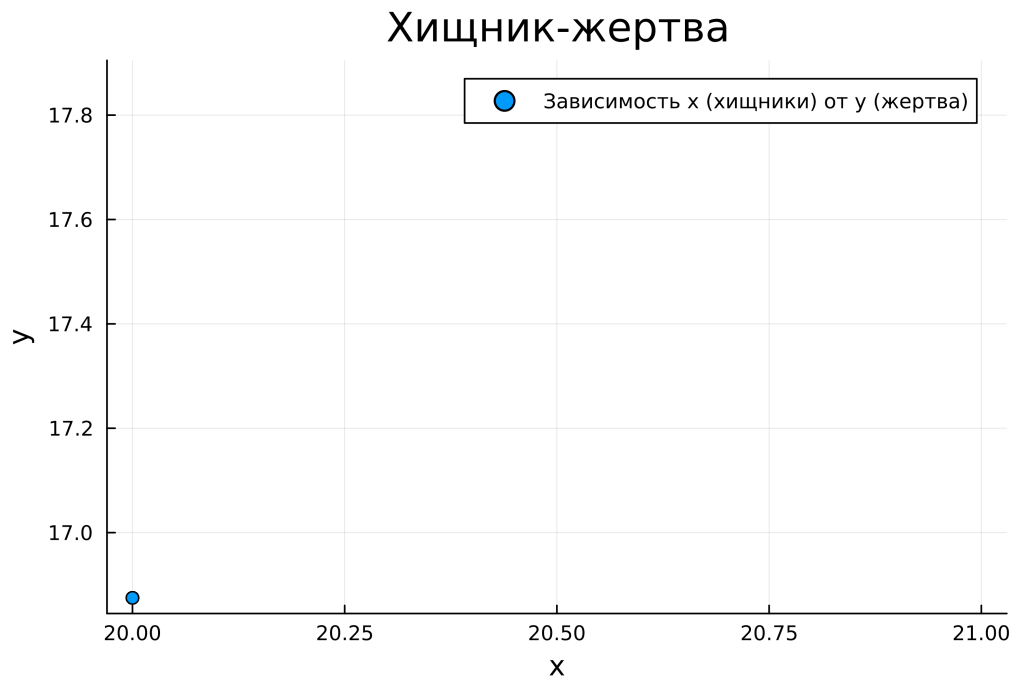


Рис. 5.8: Julia. Модель. График зависимости изменения численности хищников от изменения численности жертв (стационарное состояние)

6. Построим график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при начальных условиях на Modelica. (рис. 5.9, 5.10, 5.11)

```

model lab05_01
  constant Real a = 0.810;
  constant Real b = 0.048;
  constant Real c = 0.760;
  constant Real d = 0.038;
  Real t = time;
  Real x(t);
  Real y(t);
initial equation
  x = 7;
  y = 29;

```

equation

$$\text{der}(x) = -a * x + b * x * y;$$
$$\text{der}(y) = c * y - d * x * y;$$

annotation(experiment(StartTime = 0, StopTime = 50, Interval = 0.05));

end lab05\_01;

```
1 model lab05_01
2   constant Real a = 0.810;
3   constant Real b = 0.048;
4   constant Real c = 0.760;
5   constant Real d = 0.038;
6   Real t = time;
7   Real x(t);
8   Real y(t);
9   initial equation
10    x = 7;
11    y = 29;
12   equation
13    der(x) = -a * x + b * x * y;
14    der(y) = c * y - d * x * y;
15    annotation(experiment(StartTime = 0, StopTime = 50, Interval = 0.05));
16  end lab05_01;
17
```

Рис. 5.9: Modelica. Скрипт. Модель “Хищник-жертва” (при начальных условиях  $x = 7, y = 29$ )

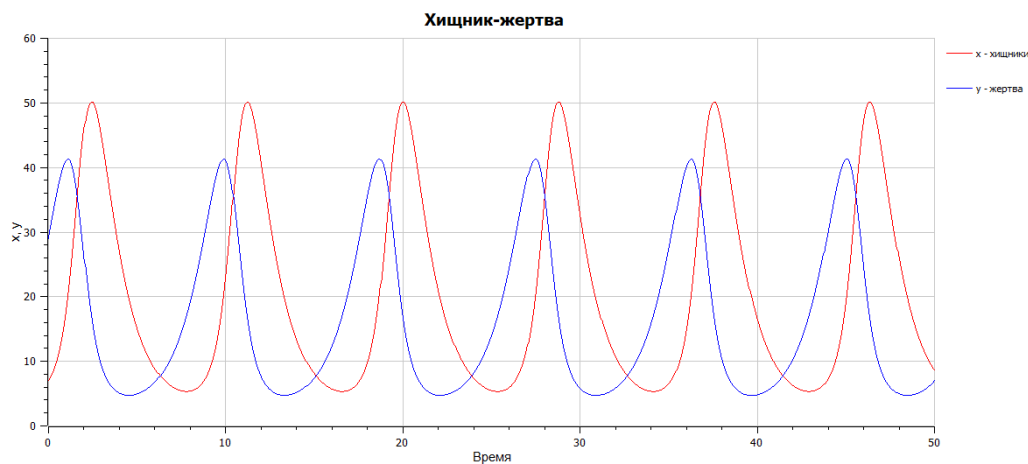


Рис. 5.10: Modelica. Модель. Графики функций изменения численности хищников и изменения численности жертв (при начальных условиях  $x = 7, y = 29$ )

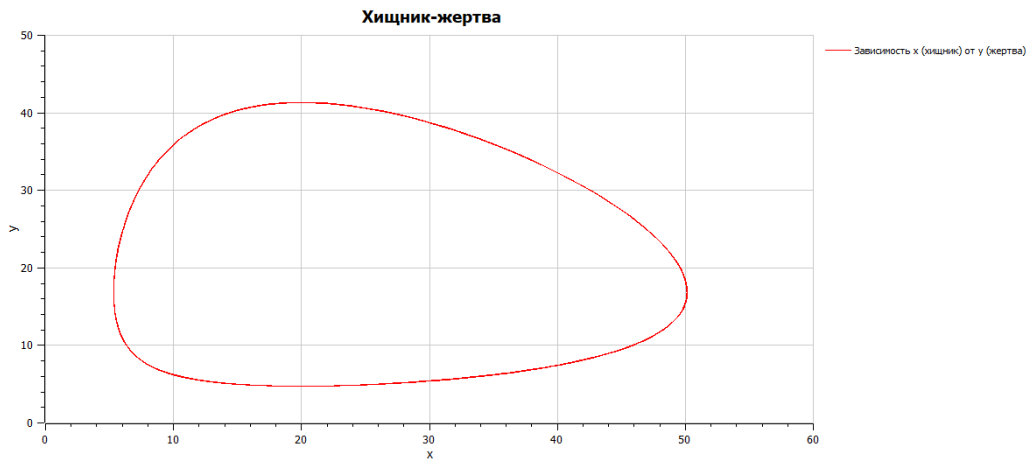


Рис. 5.11: Modelica. Модель. График зависимости изменения численности хищников от изменения численности жертв (при начальных условиях  $x = 7$ ,  $y = 29$ )

7. Построим график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв в стационарном состоянии на Modelica. (рис. 5.12, 5.13, 5.14)

```
model lab05_02
  constant Real a = 0.810;
  constant Real b = 0.048;
  constant Real c = 0.760;
  constant Real d = 0.038;
  Real t = time;
  Real x(t);
  Real y(t);
initial equation
  x = c / d;
  y = a / b;
equation
  der(x) = -a * x + b * x * y;
  der(y) = c * y - d * x * y;
```

```

    annotation(experiment(StartTime = 0, StopTime = 50, Interval = 0.05));
end lab05_02;

```

```

1  model lab05_02
2      constant Real a = 0.810;
3      constant Real b = 0.048;
4      constant Real c = 0.760;
5      constant Real d = 0.038;
6      Real t = time;
7      Real x(t);
8      Real y(t);
9  initial equation
10     x = c / d;
11     y = a / b;
12  equation
13     der(x) = -a * x + b * x * y;
14     der(y) = c * y - d * x * y;
15     annotation(experiment(StartTime = 0, StopTime = 50, Interval = 0.05));
16 end lab05_02;
17

```

Рис. 5.12: Modelica. Скрипт. Модель “Хищник-жертва” (стационарное состояние)

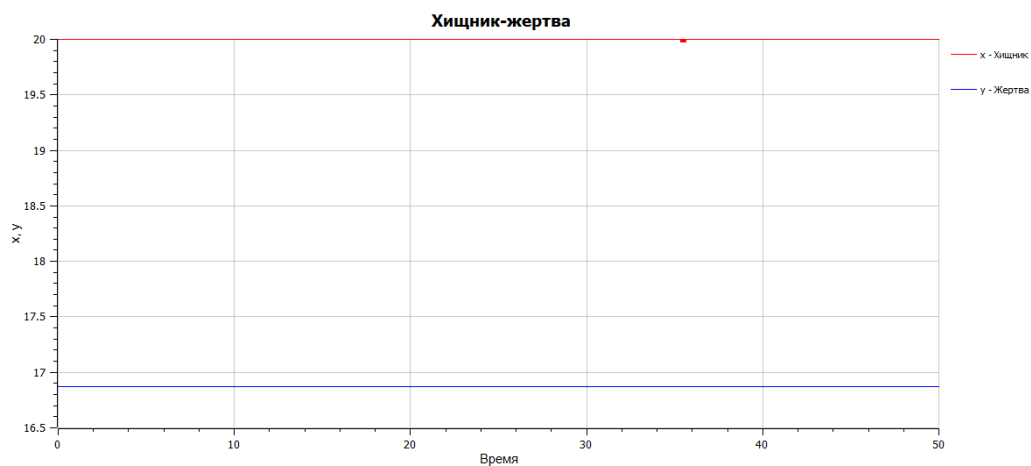


Рис. 5.13: Modelica. Модель. Графики функций изменения численности хищников и изменения численности жертв (стационарное состояние)

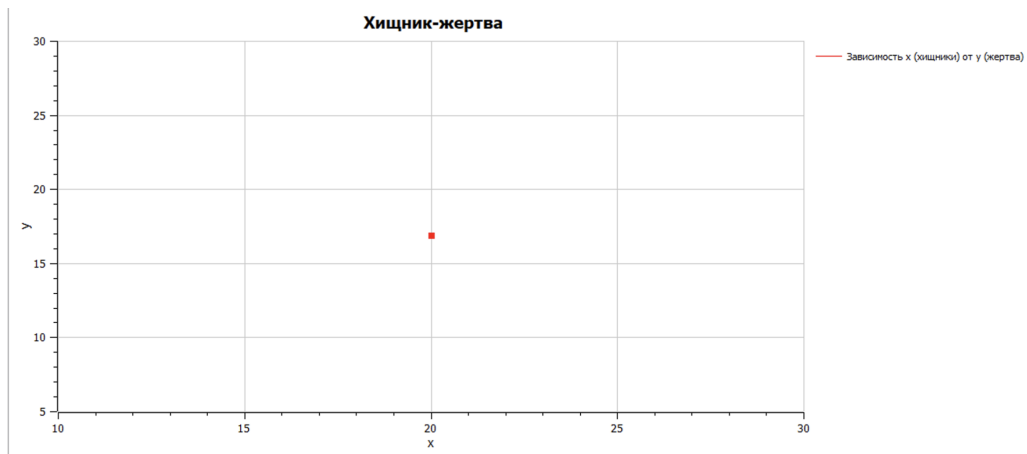


Рис. 5.14: Modelica. Модель. График зависимости изменения численности хищников от изменения численности жертв (стационарное состояние)

## 6 Анализ результатов

Работа выполнена без непредвиденных проблем в соответствии с руководством. Ошибок и сбоев не произошло.

Моделирование на OMEdit было проще и быстрее, чем при использовании средств Julia. Скрипт на Modelica вышел более лаконичным, понятным и коротким. Более того OpenModelica быстрее обрабатывала скрипт и симмулировала модель. Стоит отметить, что OpenModelica имеет множество различных полезных инструментов для настройки с симмуляцией и работой с ней. К плюсам Julia можно отнести, что она является языком программирования, который хорошо подходит для математических и технических задач.

## 7 Выводы

Мы улучшили практические навыки в области дифференциальных уравнений, улучшили навыки моделирования на Julia, также навыки моделирования на OpenModelica. Изучили модель взаимодействия двух видов типа «хищник — жертва», а именно модель Лотки-Вольтерры.

## Список литературы

1. Julia [Электронный ресурс]. URL: [http://www.unn.ru/books/met\\_files/JULIA\\_tutorial.pdf](http://www.unn.ru/books/met_files/JULIA_tutorial.pdf).
2. OpenModelica [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/OpenModelica>.
3. Модель хищник-жертва [Электронный ресурс]. RUDN. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=967245>.
4. Pluto [Электронный ресурс]. URL: <https://plutojl.org/>.
5. Plots in Julia [Электронный ресурс]. URL: <https://docs.juliaplots.org/latest/tutorial/>.
6. Differential Equations in Julia [Электронный ресурс]. URL: [https://docs.sciml.ai/DiffEqDocs/stable/getting\\_started/](https://docs.sciml.ai/DiffEqDocs/stable/getting_started/).