

Защита лабораторной работы

Лабораторная работа №5 (вариант 10)

Сергее Т.С.

09 февраля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Сергеев Тимофей Сергеевич
- Студент 3 курса группы НФИбд-02-20
- Студенческий билет №1032201669
- Российский университет дружбы народов
- 1032201669@pfur.ru

Вводная часть

- Данная работа нацелена на изучение языков программирования Julia и Modelics, созданных для выполнения математических вычислений и моделирования.

- Консоль компьютера
- Язык программирования Julia
- Язык программирования Modelica

- Для модели «хищник-жертва» постройте график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при заданных начальных условиях. Найдите стационарное состояние системы.
- Написать код на языке Julia.
- Написать код на языке Modelica для случаев.
- Составить отчёт на языке Markdown и сконвертировать его в docx и pdf.
- Подготовить презентацию на языке Markdown и защитить её.

Выполнение работы

```
1  model lab5_1
2  Real x, y;
3  Real a=0.22;
4  Real b=0.051;
5  Real c=0.33;
6  Real d=0.041;
7  Real t=time;
8  initial equation
9  x=3;
10 y=8;
11 equation
12 der(x)= -a*x + b*x*y;
13 der(y)= c*y - d*x*y;
14 end lab5_1;
```

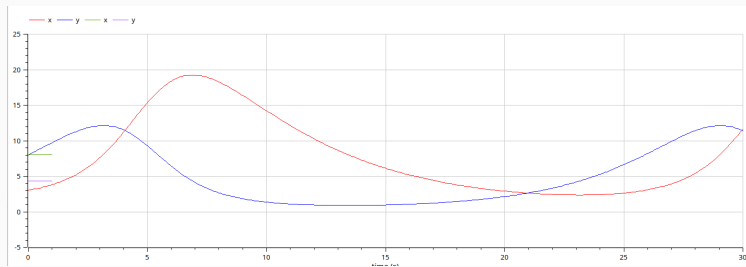


Рис. 2: Результат моделирования на языке Modelica

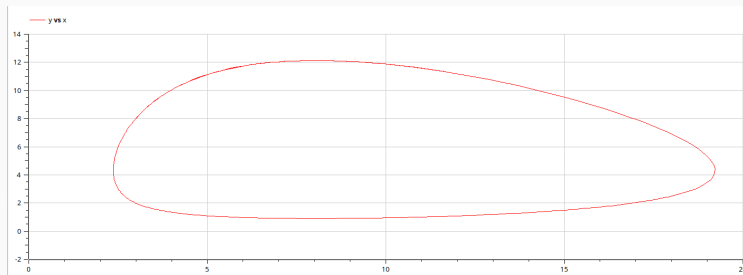


Рис. 3: Фазовый портрет на языке Modelica

```
1 model lab5_2
2 Real x, y;
3 Real a=0.22;
4 Real b=0.051;
5 Real c=0.33;
6 Real d=0.041;
7 Real t=time;
8 initial equation
9 x=c/d;
10 y=a/b;
11 equation
12 der(x)= -a*x + b*x*y;
13 der(y)= c*y - d*x*y;
14 end lab5_2;
```

Рис. 4: Реализация второго случая на языке Modelica

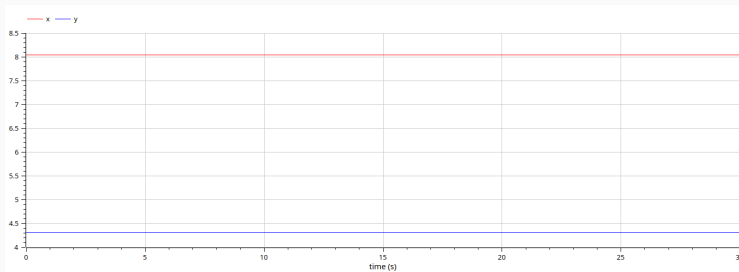


Рис. 5: Результат второго случая на языке Modelica

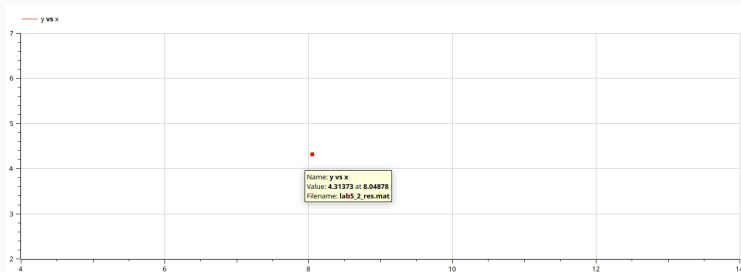


Рис. 6: Фазовый портрет второго случая на языке Modelica

```
1 using DifferentialEquations
2 using Plots
3
4 a=0.22
5 b=0.051
6 c=0.33
7 d=0.041
8
9 function F!(du, u, p, t,)
10     du[1] = -a*u[1] + b*u[1]*u[2]
11     du[2] = c*u[2] - d*u[1]*u[2]
12 end
13
14 begin
15     u0 = [3, 8]
16     T = [0.0, 30.0]
17     prob = ODEProblem(F!, u0, T)
18 end
19
```

Рис. 7: Подключаем библиотеки, задаём коэффициенты и функцию, решающую дифференциальные уравнения. Затем зададим начальные условия.

```

20 sol = solve(prob, dtmax=0.1)
21
22 const X = Float64[]
23 const Y = Float64[]
24
25 for u in sol.u
26     x, y = u
27     push!(X, x)
28     push!(Y, y)
29 end
30
31 plt = plot(
32     dpi = 300,
33     size=(1000,600),
34     plot_title="Модель «хищник-жертва»"
35 )
36

```

Рис. 8: Выполним функцию с данными значениями. Создадим два пустых массива, в которые мы передадим полученные значения. Затем с помощью функционала библиотеки Plots создадим поле для вывода результата


```
36
37 plot!(
38     plt,
39     sol.t,
40     X,
41     xlabel="x",
42     color=:blue,
43     label="Изменение численности хищников"
44 )
45
46 plot!(
47     plt,
48     sol.t,
49     Y,
50     ylabel="y",
51     color=:red,
52     label="Изменение численности жертв"
53 )
54
55 savefig(plt, "lab5.png")
```

Рис. 9: Выведем на экран полученные графы и сохраним результат в формате png

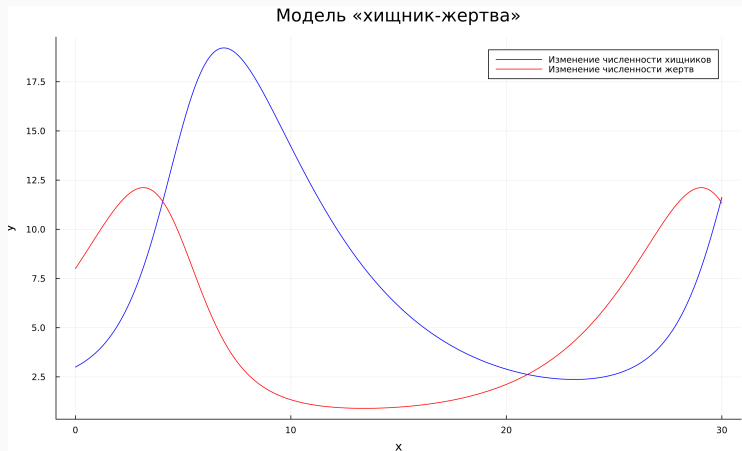


Рис. 10: Результат работы программы

```
25     for u in sol.u
26         x, y = u
27         push!(X, x)
28         push!(Y, y)
29     end
30
31     plt = plot(
32         dpi = 300,
33         size=(1000,600),
34         plot_title="Модель «хищник-жертва»"
35     )
36
37     plot!(
38         plt,
39         X,
40         Y,
41         xlabel="x",
42         color=:blue,
43         label="Фазовый портрет"
44     )
45
46     savefig(plt, "lab5_portret.png")
```

Рис. 11: Меняем только последнюю часть кода

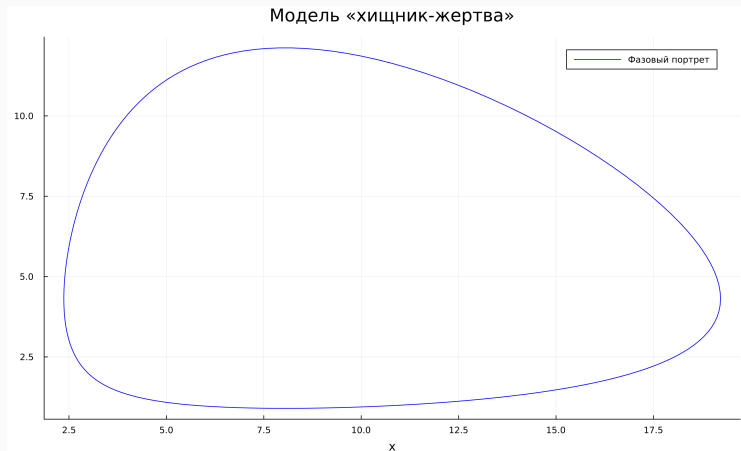


Рис. 12: Фазовый портрет

```
1 using DifferentialEquations
2 using Plots
3
4 a=0.22
5 b=0.051
6 c=0.33
7 d=0.041
8
9 function F!(du, u, p, t,)
10     du[1] = -a*u[1] + b*u[1]*u[2]
11     du[2] = c*u[2] - d*u[1]*u[2]
12 end
13
14 begin
15     u0 = [c/d, a/b]
16     T = [0.0, 30.0]
17     prob = ODEProblem(F!, u0, T)
18 end
19
20 sol = solve(prob, dtmax=0.1)
21
22 const X = Float64[]
23 const Y = Float64[]
```

Рис. 13: Изменения в коде

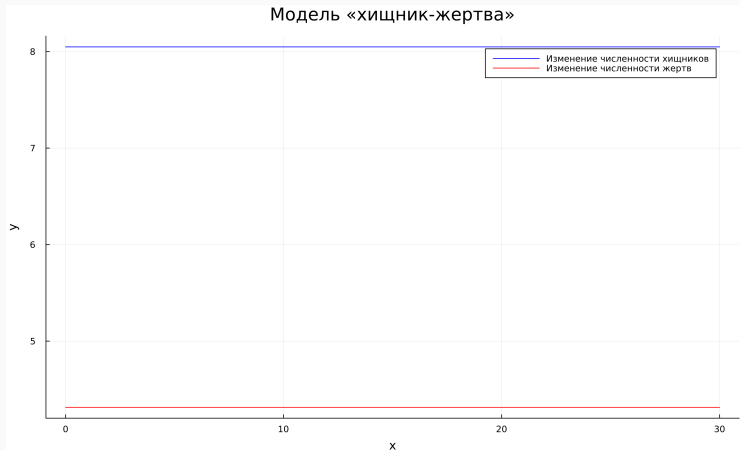


Рис. 14: Результат работы программы для второго случая

```
28     push!(Y, y)
29 end
30
31 plt = plot(
32     dpi = 300,
33     size=(1000,600),
34     plot_title="Модель «хищник-жертва»"
35 )
36
37 scatter!(
38     plt,
39     X,
40     Y,
41     xlabel="x",
42     color=:blue,
43     label="Фазовый портрет"
44 )
45
46 savefig(plt, "lab5_stat_portret.png")
```

Рис. 15: Меняем только последнюю часть кода

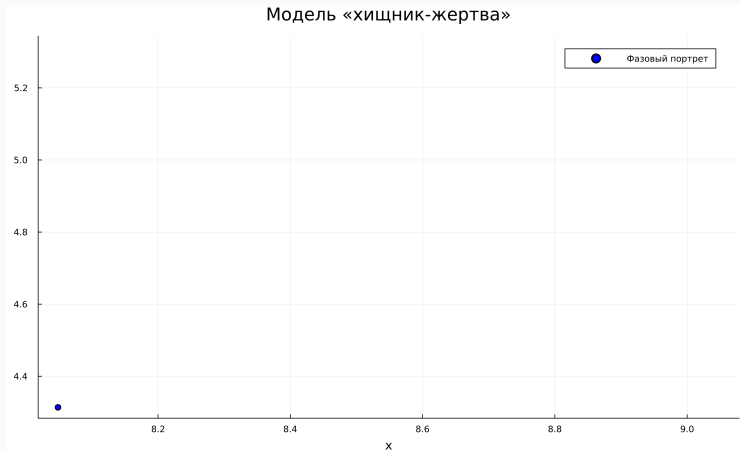


Рис. 16: Фазовый портрет для первого случая

Результаты

Выполнив данную лабораторную работу, мы продолжили знакомство с языками программирования Julia и Modelica. Сравнивая реализацию одной программы на этих двух языках, можно заметить, что реализация на языке Modelica заметно проще и более точно показывает результат, поскольку можно отследить значения переменных с максимальной точностью на любом отрезке времени.

Вовремя выполненная лабораторная работа - хорошая оценка - довольный студент -
счастливое будущее!