

Защита лабораторной работы

Лабораторная работа №4 (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

- Построить фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора.
- Написать код на языке Julia и Modelica для случаев:
 - без затуханий и без действий внешней силы;
 - с затуханием и без действий внешней силы;
 - с затуханием и под действием внешней силы;
- Составить отчёт на языке Markdown и сконвертировать его в docx и pdf.
- Подготовить презентацию на языке Markdown и защитить её.

Выполнение работы

```
1  model lab4_1
2  Real x, y;
3  Real t=time;
4  initial equation
5  x=2;
6  y=0;
7  equation
8  der(x)= y;
9  der(y)= -7*x;
10 end lab4_1;
```

Рис. 1: Реализация первого случая на языке Modelica

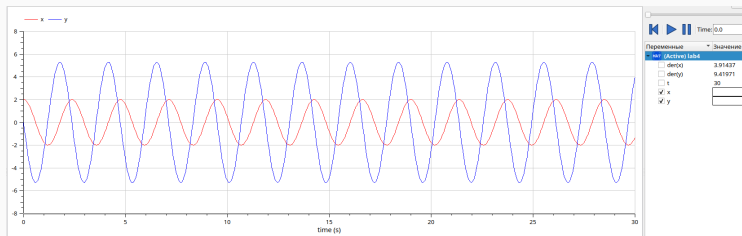


Рис. 2: Результат первого случая на языке Modelica

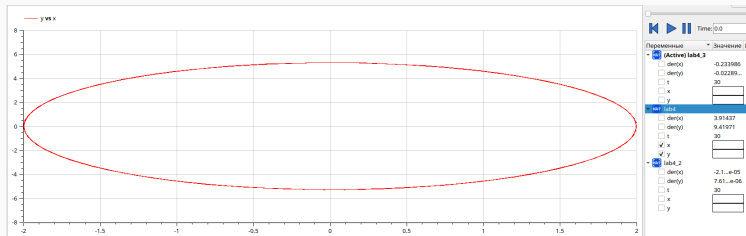


Рис. 3: Фазовый портрет первого случая на языке Modelica

```
1 using DifferentialEquations
2 using Plots
3
4 function F!(du, u, p, t,)
5     du[1] = u[2]
6     du[2] = -7*u[1]
7 end
8
9 begin
10     u0 = [2, 0]
11     T = [0.0, 30.0]
12     prob = ODEProblem(F!, u0, T)
13 end
14
```

Рис. 4: Подключаем библиотеки, задаём функцию, решающую дифференциальные уравнения. Затем зададим начальные условия.

```

14
15 sol = solve(prob, dtmax=0.05)
16
17 const X = Float64[]
18 const Y = Float64[]
19
20 for u in sol.u
21     x, y = u
22     push!(X, x)
23     push!(Y, y)
24 end
25
26 plt = plot(
27     dpi = 300,
28     size=(1000,600),
29     plot_title="Решение уравнения гармонического осциллятора"
30 )
31

```

Рис. 5: 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.

```
31
32 plot!(
33     plt,
34     sol.t,
35     X,
36     color=:blue,
37     label="x(t)"
38 )
39
40 plot!(
41     plt,
42     sol.t,
43     Y,
44     color=:red,
45     label="y(t)"
46 )
47
48 savefig(plt, "lab41.png")
```

Рис. 6: Выведем на экран полученные графы и сохраним результат в формате png

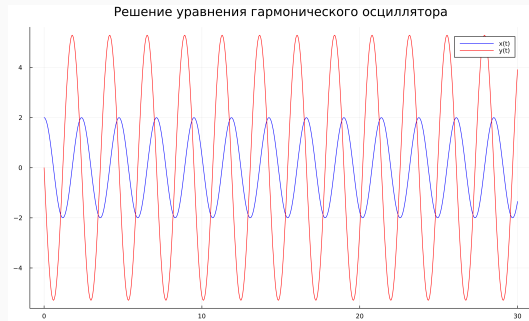


Рис. 7: Результат работы программы для первого случая

```
26 plt = plot(  
27     dpi = 300,  
28     size=(1000,600),  
29     plot_title="Решение уравнения гармонического осциллятора"  
30 )  
31  
32 plot!(  
33     plt,  
34     X,  
35     Y,  
36     color=:green,  
37     label="Фазовый портрет"  
38 )  
39  
40 savefig(plt, "lab41_portret.png")
```

Рис. 8: Меняем только последнюю часть кода

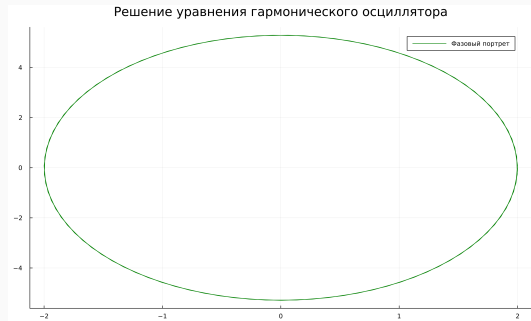


Рис. 9: Фазовый портрет для первого случая

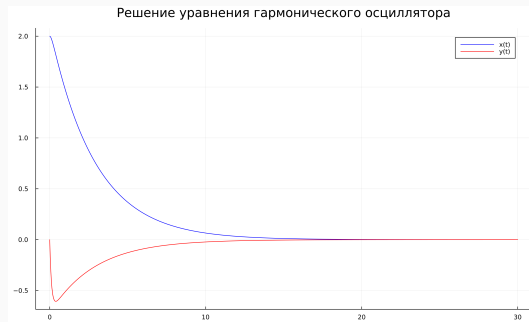


Рис. 10: Результат работы программы для второго случая

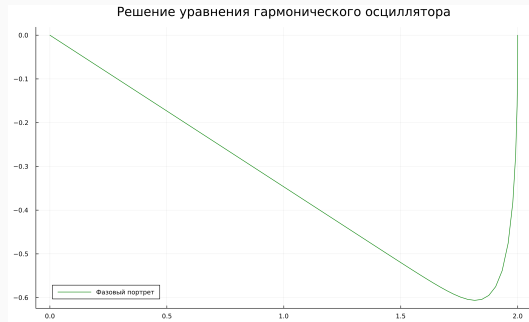


Рис. 11: Фазовый портрет для второго случая

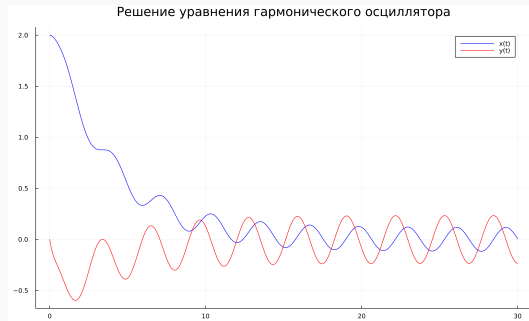


Рис. 12: Результат работы программы для третьего случая

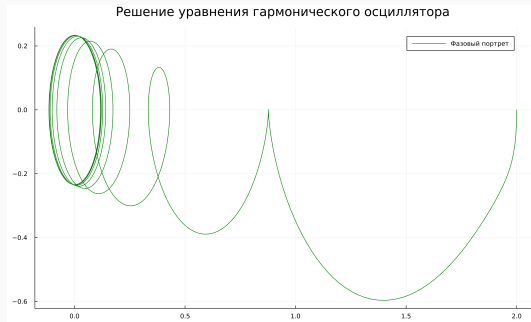


Рис. 13: Фазовый портрет для трктькго случая

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 графика.

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