

Отчет лабораторной работы №4

Низамова Альфия Айдаровна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	19
	Список литературы	20

Список иллюстраций

3.1	Результат Julia 1	8
3.2	Фазовый портрет Julia 1	8
3.3	Код на OpenModelica 1	9
3.4	Результат OpenModelica 1	10
3.5	Фазовый портрет OpenModelica 1	10
3.6	Результат Julia 2	11
3.7	Фазовый портрет Julia 2	12
3.8	Код на OpenModelica 2	13
3.9	Результат OpenModelica 2	14
3.10	Фазовый портрет OpenModelica 2	14
3.11	Результат Julia 3	15
3.12	Фазовый портрет Julia 3	16
3.13	Код на OpenModelica 3	17
3.14	Результат OpenModelica 3	18
3.15	Фазовый портрет OpenModelica 3	18

Список таблиц

1 Цель работы

Целью данной лабораторной работы является ознакомление с моделью гармонических колебаний и построение их на языках программирования Julia и OpenModelica

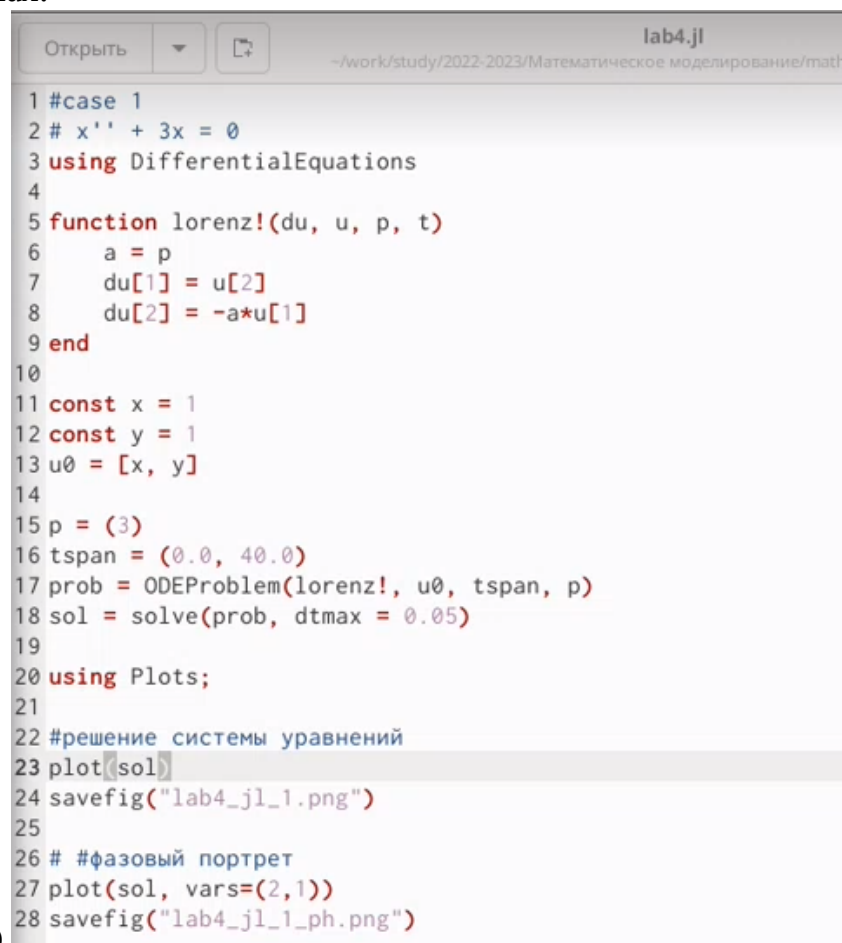
2 Задание

Постройте фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора для следующих случаев 1. Колебания гармонического осциллятора без затуханий и без действий внешней силы $x'' + 3x = 0$ 2. Колебания гармонического осциллятора с затуханием и без действий внешней силы $x'' + x' + 4x = 0$ 3. Колебания гармонического осциллятора с затуханием и под действием внешней силы $x'' + 2x' + x = \sin(2t)$

На интервале t от 0 до 40 (шаг 0.05) с начальными условиями $x_0 = 1$, $y_0 = 1$

3 Выполнение лабораторной работы

Написала код на языке Julia и Openmodelica для построения фазового портрета гармонического осциллятора и решения уравнения гармонического осциллятора для каждого случая.



```
1 #case 1
2 #  $x'' + 3x = 0$ 
3 using DifferentialEquations
4
5 function lorenz!(du, u, p, t)
6     a = p
7     du[1] = u[2]
8     du[2] = -a*u[1]
9 end
10
11 const x = 1
12 const y = 1
13 u0 = [x, y]
14
15 p = (3)
16 tspan = (0.0, 40.0)
17 prob = ODEProblem(lorenz!, u0, tspan, p)
18 sol = solve(prob, dtmax = 0.05)
19
20 using Plots;
21
22 #решение системы уравнений
23 plot(sol)
24 savefig("lab4_jl_1.png")
25
26 # фазовый портрет
27 plot(sol, vars=(2,1))
28 savefig("lab4_jl_1_ph.png")
```

1 случай (рис.1-4)

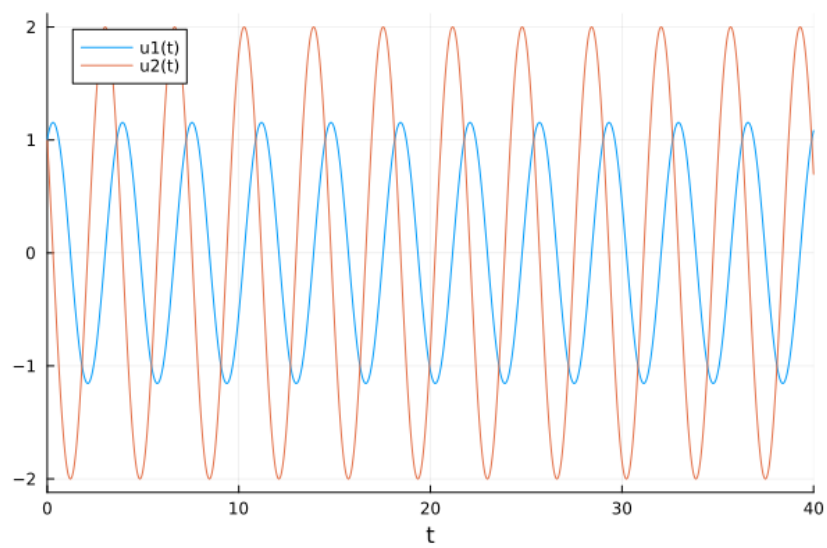


Рис. 3.1: Результат Julia 1

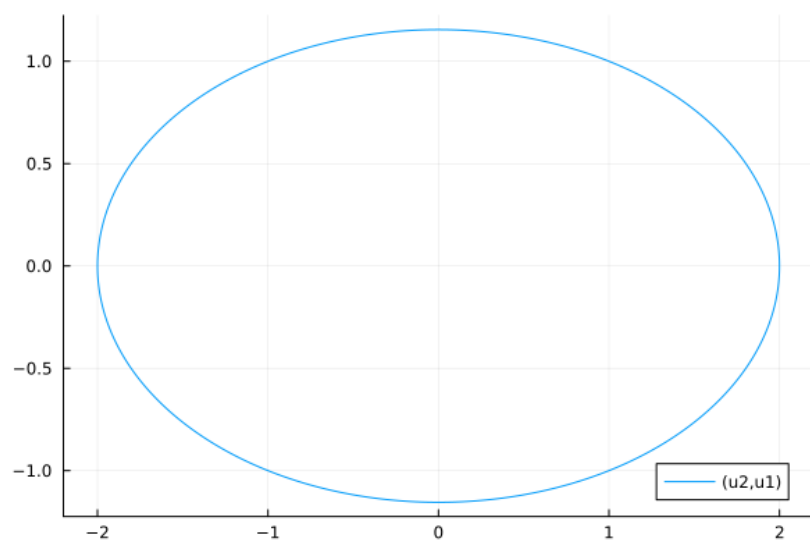
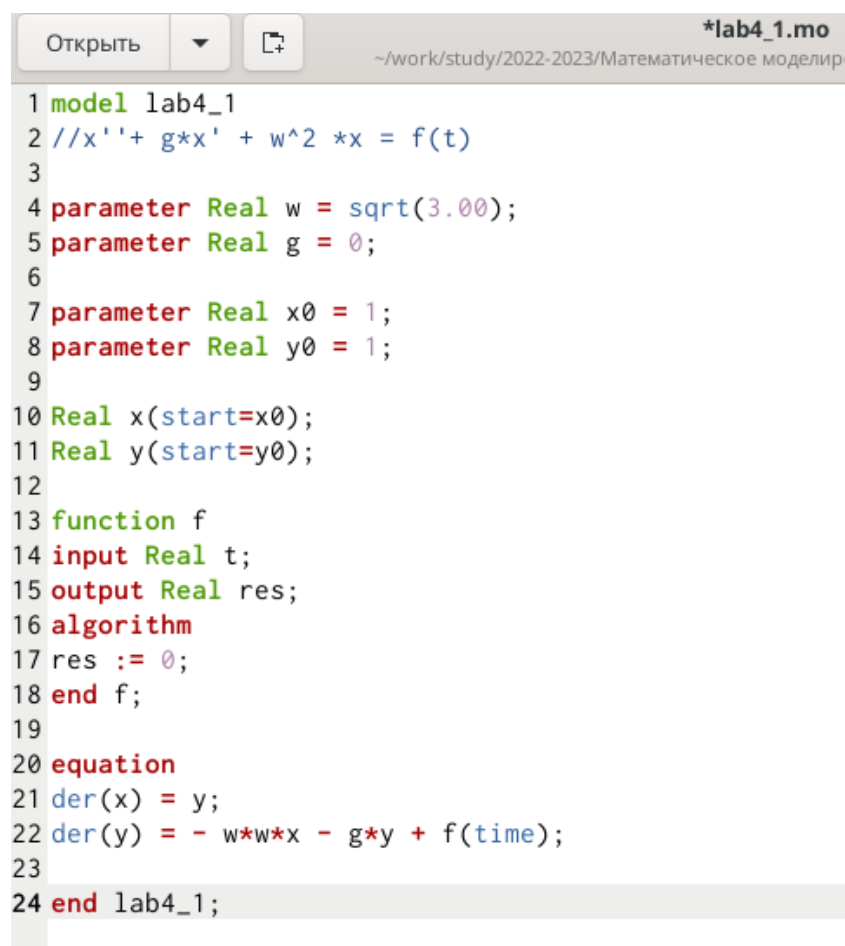


Рис. 3.2: Фазовый портрет Julia 1



```
1 model lab4_1
2 //x'' + g*x' + w^2 *x = f(t)
3
4 parameter Real w = sqrt(3.00);
5 parameter Real g = 0;
6
7 parameter Real x0 = 1;
8 parameter Real y0 = 1;
9
10 Real x(start=x0);
11 Real y(start=y0);
12
13 function f
14 input Real t;
15 output Real res;
16 algorithm
17 res := 0;
18 end f;
19
20 equation
21 der(x) = y;
22 der(y) = - w*w*x - g*y + f(time);
23
24 end lab4_1;
```

Рис. 3.3: Код на OpenModelica 1

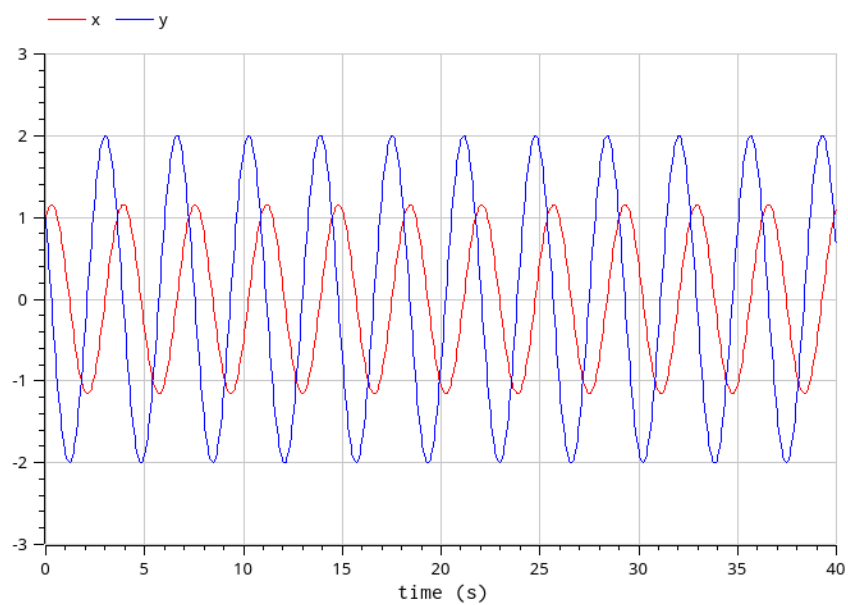


Рис. 3.4: Результат OpenModelica 1

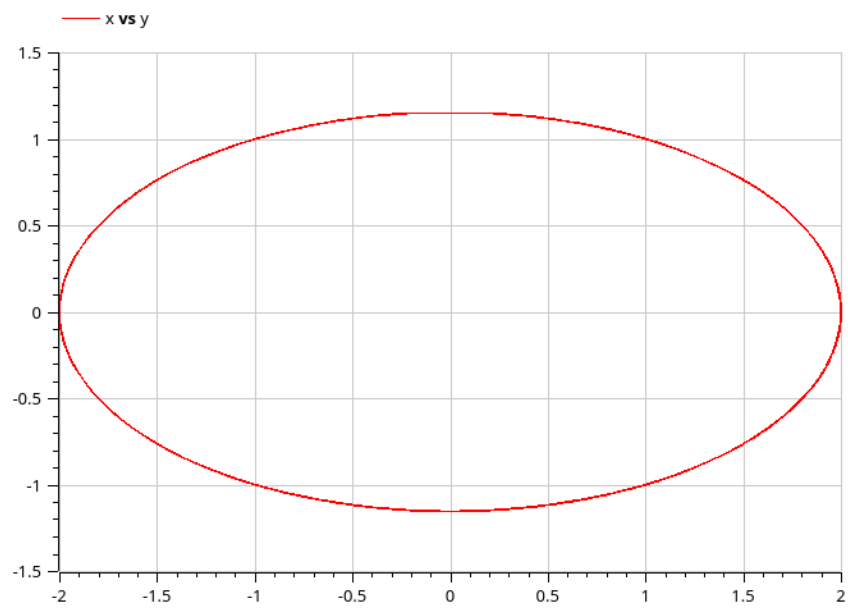


Рис. 3.5: Фазовый портрет OpenModelica 1

```

Открыть ▼
~\work\study\2022-2023\Математическое моделирование\math
*lab4.jl

1 #case 2
2 #  $x'' + x' + 4x = 0$ 
3 using DifferentialEquations
4
5 function lorenz!(du, u, p, t)
6     a, b = p
7     du[1] = u[2]
8     du[2] = -a*du[1] - b*u[1]
9 end
10
11 const x = 1
12 const y = 1
13 u0 = [x, y]
14
15 p = (sqrt(1), 4)
16 tspan = (0.0, 40.0)
17 prob = ODEProblem(lorenz!, u0, tspan, p)
18 sol = solve(prob, dtmax = 0.05)
19
20 using Plots;
21
22 #решение системы уравнений
23 plot(sol)
24 savefig("lab4_jl_2.png")
25
26 #фазовый портрет
27 plot(sol, vars=(2,1))
28 savefig("lab4_jl_2_ph.png")

```

2 случай (рис.5-8)

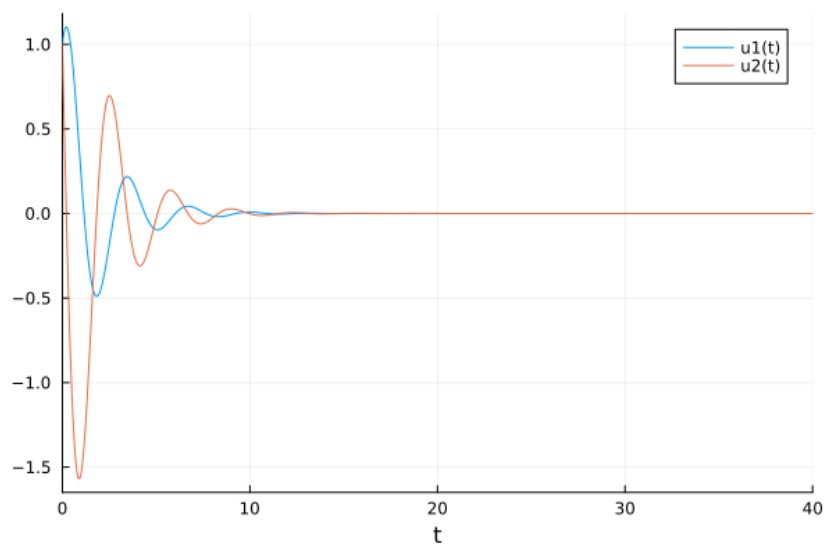


Рис. 3.6: Результат Julia 2

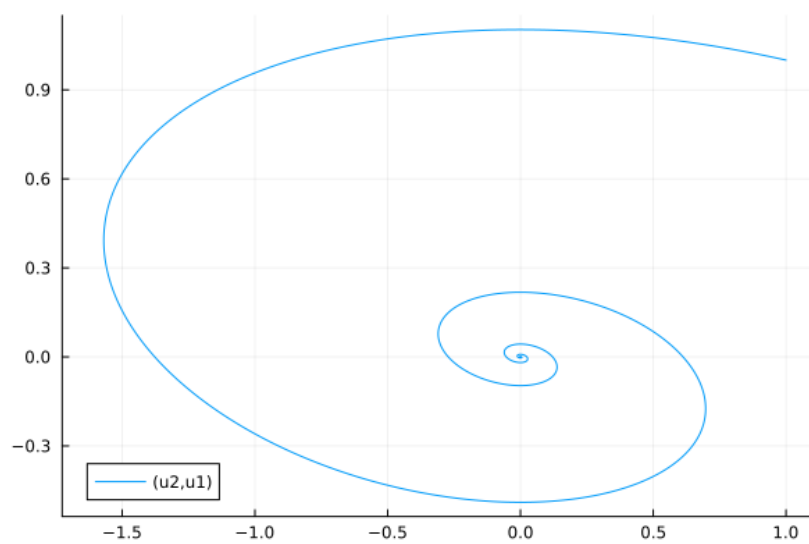
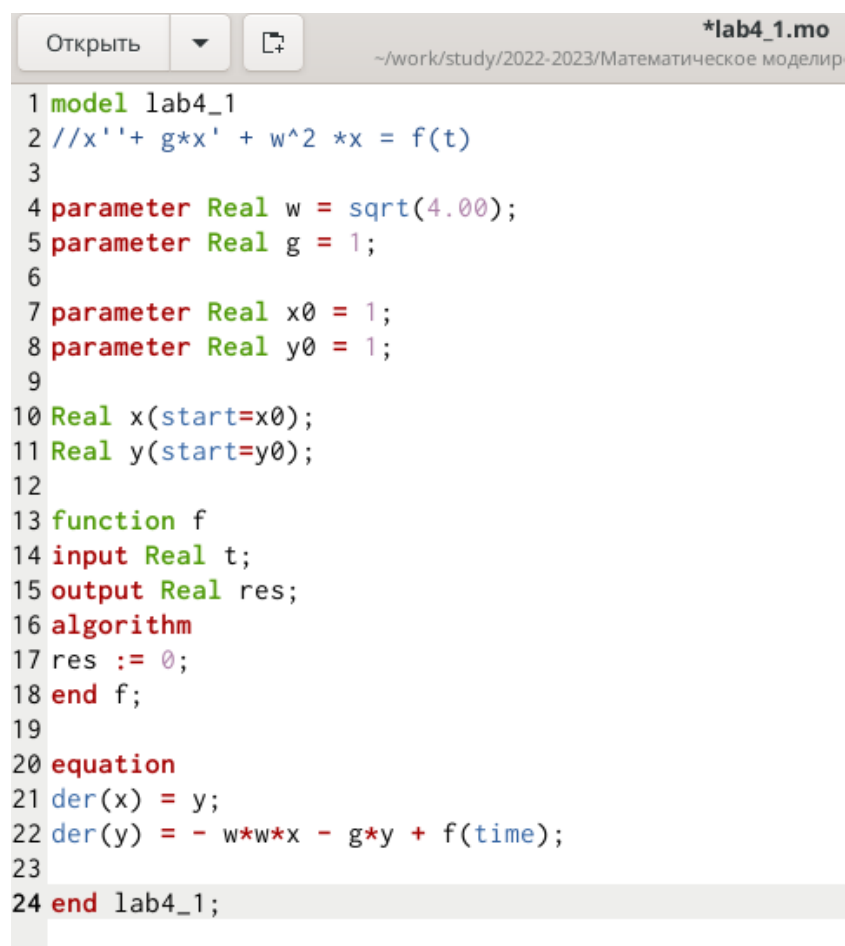


Рис. 3.7: Фазовый портрет Julia 2



```
1 model lab4_1
2 //x'' + g*x' + w^2 *x = f(t)
3
4 parameter Real w = sqrt(4.00);
5 parameter Real g = 1;
6
7 parameter Real x0 = 1;
8 parameter Real y0 = 1;
9
10 Real x(start=x0);
11 Real y(start=y0);
12
13 function f
14 input Real t;
15 output Real res;
16 algorithm
17 res := 0;
18 end f;
19
20 equation
21 der(x) = y;
22 der(y) = - w*w*x - g*y + f(time);
23
24 end lab4_1;
```

Рис. 3.8: Код на OpenModelica 2

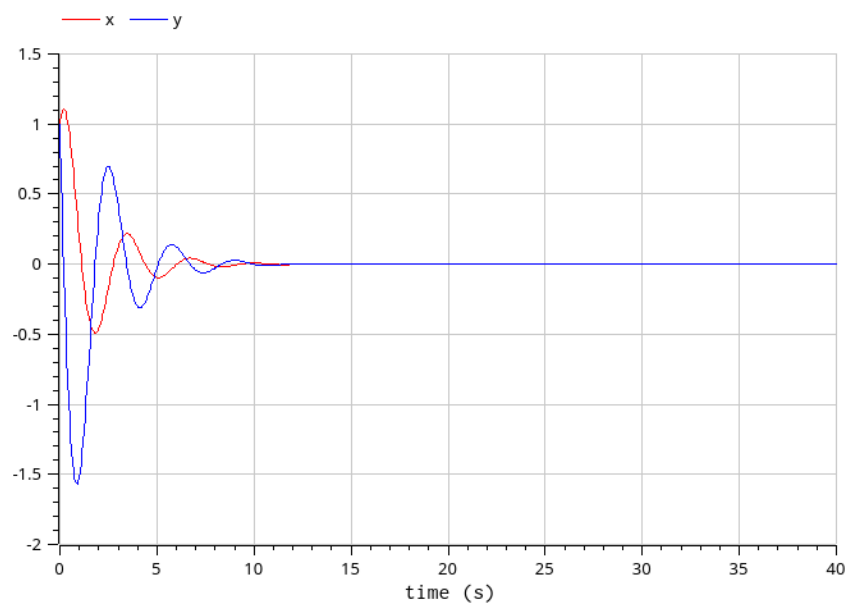


Рис. 3.9: Результат OpenModelica 2

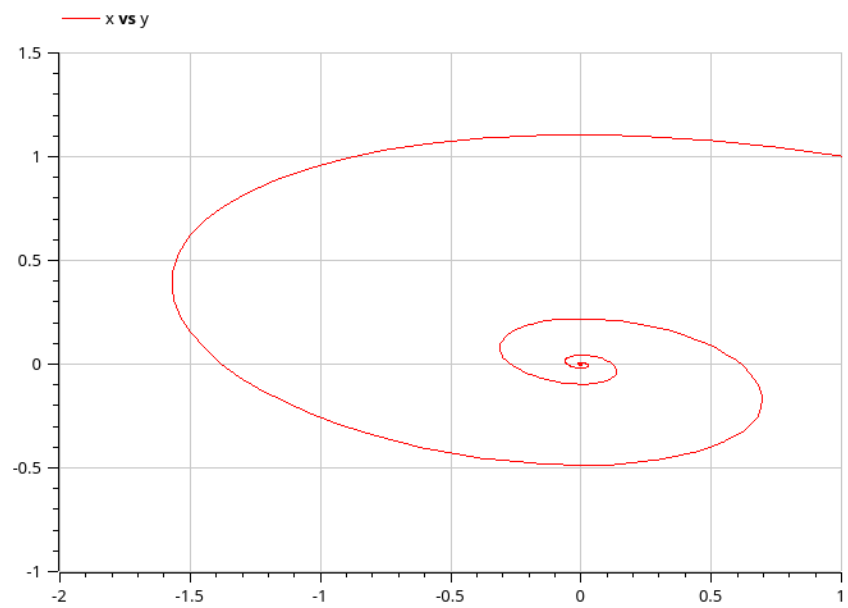


Рис. 3.10: Фазовый портрет OpenModelica 2

```

lab4.jl
~/work/study/2022-2023/Математическое моделирование/math

1 #case 2
2 #  $x'' + 2x' + x = \sin(2t)$ 
3 using DifferentialEquations
4
5 function lorenz!(du, u, p, t)
6     a, b = p
7     du[1] = u[2]
8     du[2] = -a*du[1] - b*u[1] + sin(2*t)
9 end
10
11 const x = 1
12 const y = 1
13 u0 = [x, y]
14
15 p = (sqrt(2), 1)
16 tspan = (0.0, 40.0)
17 prob = ODEProblem(lorenz!, u0, tspan, p)
18 sol = solve(prob, dtmax = 0.05)
19
20 using Plots;
21
22 #решение системы уравнений
23 plot(sol)
24 savefig("lab4_jl_3.png")
25
26 # #фазовый портрет
27 plot(sol, vars=(2,1))
28 savefig("lab4_jl_3_ph.png")

```

3 случай (рис.9-12)

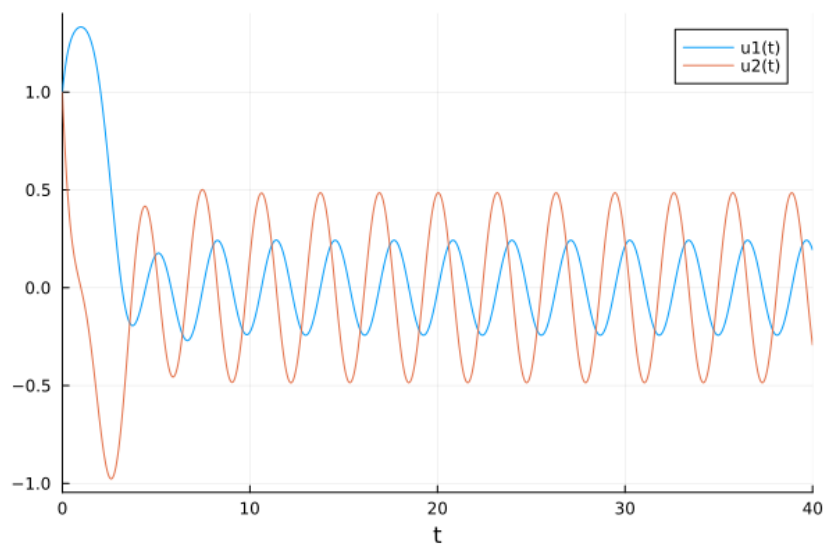


Рис. 3.11: Результат Julia 3

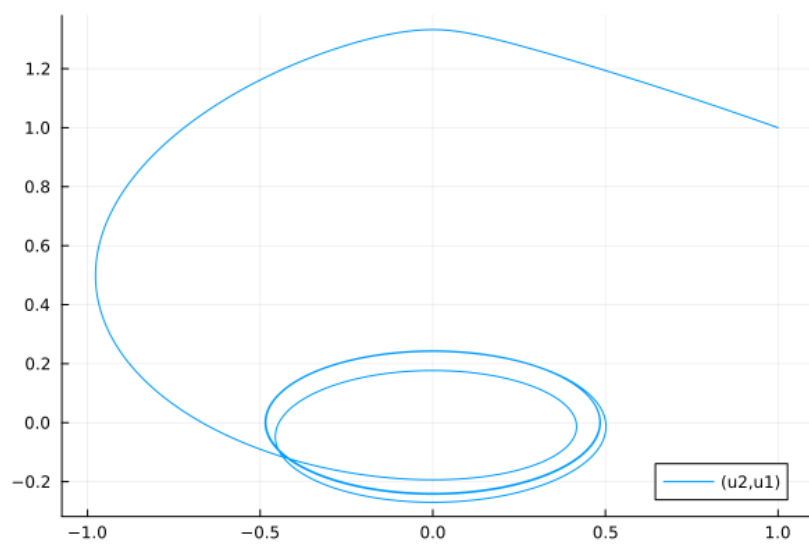
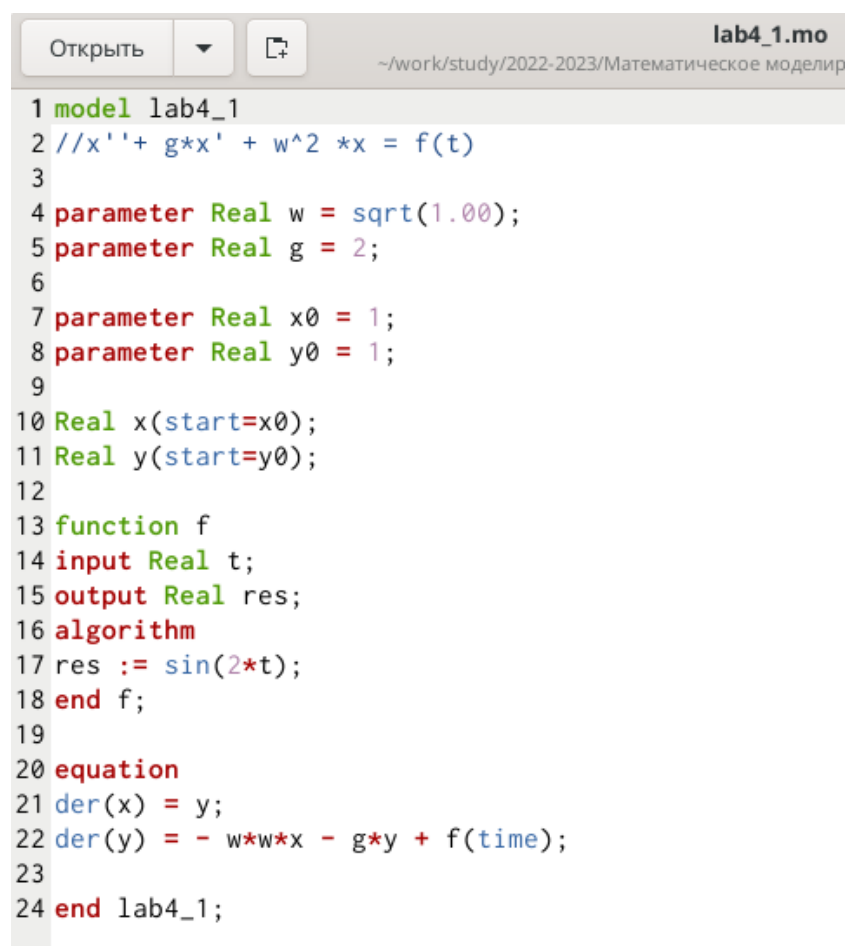


Рис. 3.12: Фазовый портрет Julia 3



```
1 model lab4_1
2 //x'' + g*x' + w^2 *x = f(t)
3
4 parameter Real w = sqrt(1.00);
5 parameter Real g = 2;
6
7 parameter Real x0 = 1;
8 parameter Real y0 = 1;
9
10 Real x(start=x0);
11 Real y(start=y0);
12
13 function f
14 input Real t;
15 output Real res;
16 algorithm
17 res := sin(2*t);
18 end f;
19
20 equation
21 der(x) = y;
22 der(y) = - w*w*x - g*y + f(time);
23
24 end lab4_1;
```

Рис. 3.13: Код на OpenModelica 3

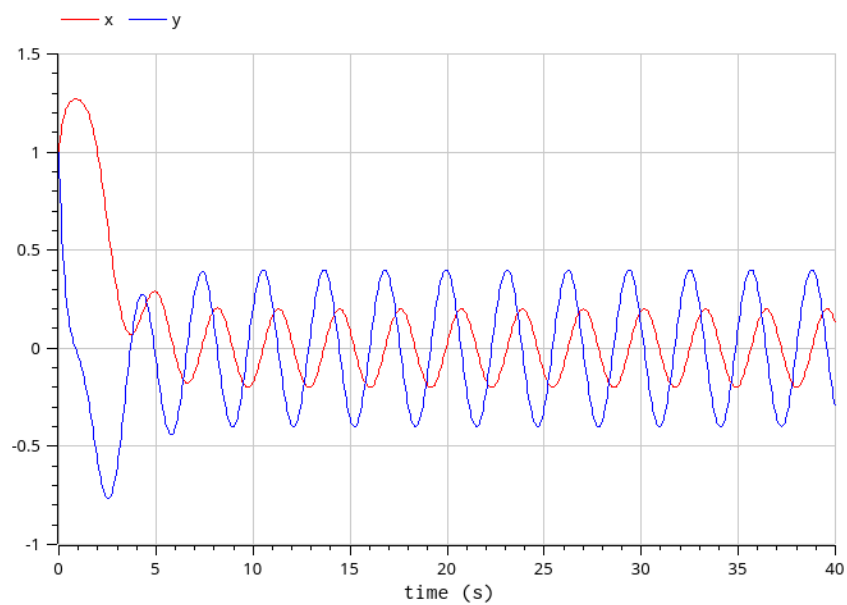


Рис. 3.14: Результат OpenModelica 3

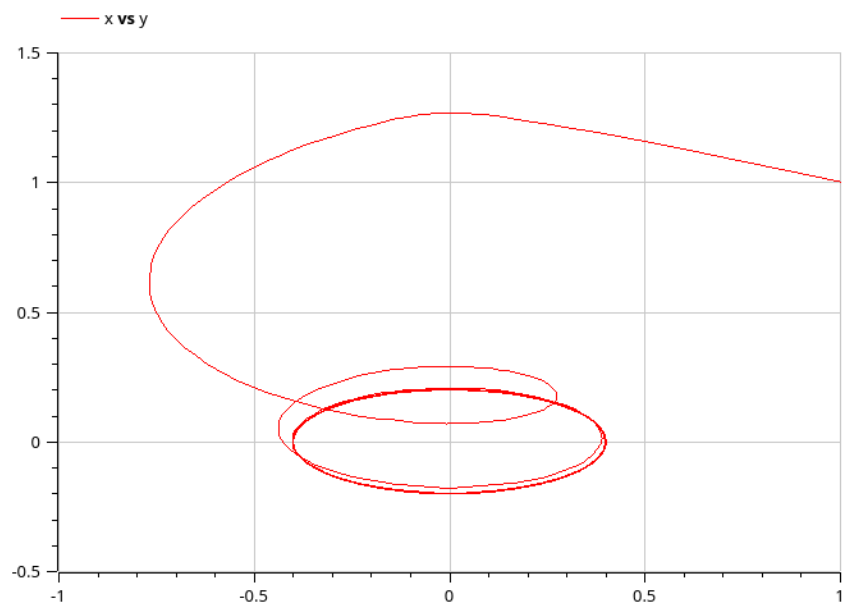


Рис. 3.15: Фазовый портрет OpenModelica 3

4 Выводы

Мы ознакомились с моделью гармонических колебаний и построили их на языках программирования Julia и OpenModelica

Список литературы