

# **Отчёт по лабораторной работе**

**Лабораторная №4 по математическому моделированию**

**Дзахмишев Камбулат Заурович**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>6</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
<b>7</b>	<b>Выполнение лабораторной работы</b>	<b>11</b>
<b>8</b>	<b>Выполнение лабораторной работы</b>	<b>12</b>
<b>9</b>	<b>Выполнение лабораторной работы</b>	<b>13</b>
<b>10</b>	<b>Выполнение лабораторной работы</b>	<b>14</b>
<b>11</b>	<b>Выполнение лабораторной работы</b>	<b>15</b>
<b>12</b>	<b>Выполнение лабораторной работы</b>	<b>16</b>
<b>13</b>	<b>Вывод</b>	<b>17</b>

# Список иллюстраций

1.1	Мой вариант по формуле - №28. . . . .	5
2.1	Часть кода в среде OpenModelica . . . . .	6
3.1	Другая часть кода в среде Openmodelica . . . . .	7
4.1	Первый график и фазовый портрет (без затухания и внешней силы). . . . .	8
5.1	Второй график и фазовый портрет (с затуханием и без внешней силы). . . . .	9
6.1	Третий график и фазовый портрет (с затуханием и с внешней силой). . . . .	10
7.1	Код и соответствующие выводы в среде Pluto на языке Julia. . . . .	11
8.1	Код и соответствующие выводы в среде Pluto на языке Julia. . . . .	12
9.1	Код и соответствующие выводы в среде Pluto на языке Julia. . . . .	13
10.1	Код и соответствующие выводы в среде Pluto на языке Julia. . . . .	14
11.1	Код и соответствующие выводы в среде Pluto на языке Julia. . . . .	15
12.1	Код и соответствующие выводы в среде Pluto на языке Julia. . . . .	16

## **Список таблиц**

# 1 Цель работы

Вариант № 28

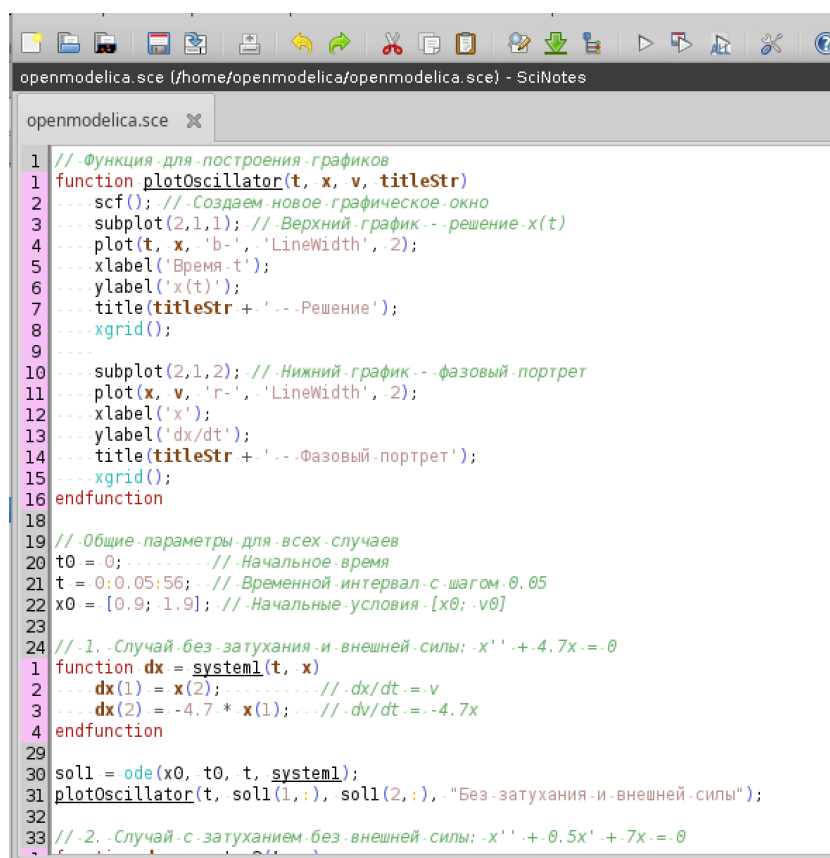
Постройте фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора для следующих случаев

- 
1. Колебания гармонического осциллятора без затуханий и без действий внешней силы  $\ddot{x} + 4.7x = 0$
  2. Колебания гармонического осциллятора с затуханием и без действий внешней силы  $\ddot{x} + 0.5\dot{x} + 7x = 0$
  3. Колебания гармонического осциллятора с затуханием и под действием внешней силы  $\ddot{x} + 7\dot{x} + 0.5x = 0.5\sin(0.7t)$

На интервале  $t \in [0; 56]$  (шаг 0.05) с начальными условиями  $x_0 = 0.9, y_0 = 1.9$

**Рис. 1.1: Мой вариант по формуле - №28.**

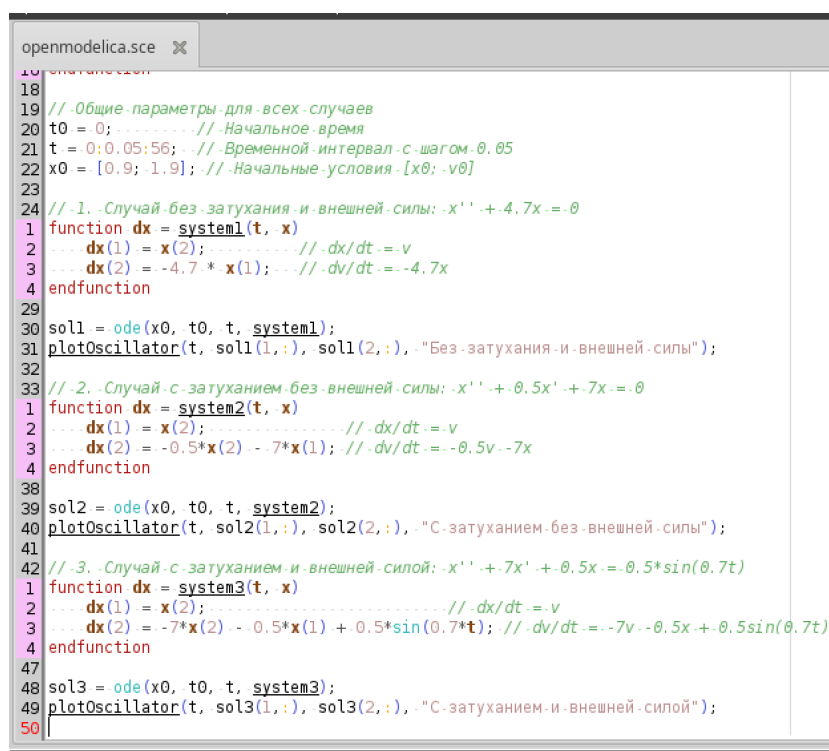
## 2 Выполнение лабораторной работы



```
1 // -Функция -для построения графиков
1 function plotOscillator(t, x, v, titleStr)
2   ....scf(); // -Создаем новое графическое окно
3   ....subplot(2,1,1); // -Верхний график - решение x(t)
4   ....plot(t, x, 'b-', 'LineWidth', 2);
5   ....xlabel('Время-t');
6   ....ylabel('x(t)');
7   ....title(titleStr + ' - -Решение');
8   ....xgrid();
9   ....
10  ....subplot(2,1,2); // -Нижний график - фазовый портрет
11  ....plot(x, v, 'r-', 'LineWidth', 2);
12  ....xlabel('x');
13  ....ylabel('dx/dt');
14  ....title(titleStr + ' - -Фазовый портрет');
15  ....xgrid();
16 endfunction
17
18 // -Общие параметры для всех случаев
19 t0 = 0; ..... // -Начальное время
20 t = 0:0.05:56; // -Временной интервал с шагом 0.05
21 x0 = [0.9; -1.9]; // -Начальные условия [x0; -v0]
22
23 // -1. -Случай без затухания и внешней силы: -x'' + -4.7x = -0
24 function dx = system1(t, x)
25   ....dx(1) = x(2); ..... // -dx/dt = -v
26   ....dx(2) = -4.7 * x(1); ..... // -dv/dt = -4.7x
27 endfunction
28
29 sol1 = ode(x0, t0, t, system1);
30 plotOscillator(t, sol1(1,:), sol1(2,:), "Без затухания и внешней силы");
31
32 // -2. -Случай с затуханием без внешней силы: -x'' + -0.5x' + -7x = -0
33
```

Рис. 2.1: Часть кода в среде OpenModelica

### 3 Выполнение лабораторной работы



```
openmodelica.sce
18 endfunction
19 // -Общие-параметры-для-всех-случаев
20 t0 = 0; // -Начальное-время
21 t = 0:0.05:56; // -Временной-интервал-с-шагом-0.05
22 x0 = [0.9; -1.9]; // -Начальные-условия-[x0; -v0]
23
24 // -1.-Случай-без-затухания-и-внешней-силы: -x'' + -4.7x = -0
1 function dx = system1(t, x)
2   dx(1) = x(2); // -dx/dt = -v
3   dx(2) = -4.7 * x(1); // -dv/dt = -4.7x
4 endfunction
29
30 sol1 = ode(x0, t0, t, system1);
31 plotOscillator(t, sol1(1,:), sol1(2,:), "-Без-затухания-и-внешней-силы");
32
33 // -2.-Случай-с-затуханием-без-внешней-силы: -x'' + -0.5x' + -7x = -0
1 function dx = system2(t, x)
2   dx(1) = x(2); // -dx/dt = -v
3   dx(2) = -0.5*x(2) - 7*x(1); // -dv/dt = -0.5v - 7x
4 endfunction
38
39 sol2 = ode(x0, t0, t, system2);
40 plotOscillator(t, sol2(1,:), sol2(2,:), "-С-затуханием-без-внешней-силы");
41
42 // -3.-Случай-с-затуханием-и-внешней-силой: -x'' + -7x' + -0.5x = -0.5*sin(0.7t)
1 function dx = system3(t, x)
2   dx(1) = x(2); // -dx/dt = -v
3   dx(2) = -7*x(2) - 0.5*x(1) + 0.5*sin(0.7*t); // -dv/dt = -7v - 0.5x + 0.5sin(0.7t)
4 endfunction
47
48 sol3 = ode(x0, t0, t, system3);
49 plotOscillator(t, sol3(1,:), sol3(2,:), "-С-затуханием-и-внешней-силой");
50
```

Рис. 3.1: Другая часть кода в среде Openmodelica

## 4 Выполнение лабораторной работы

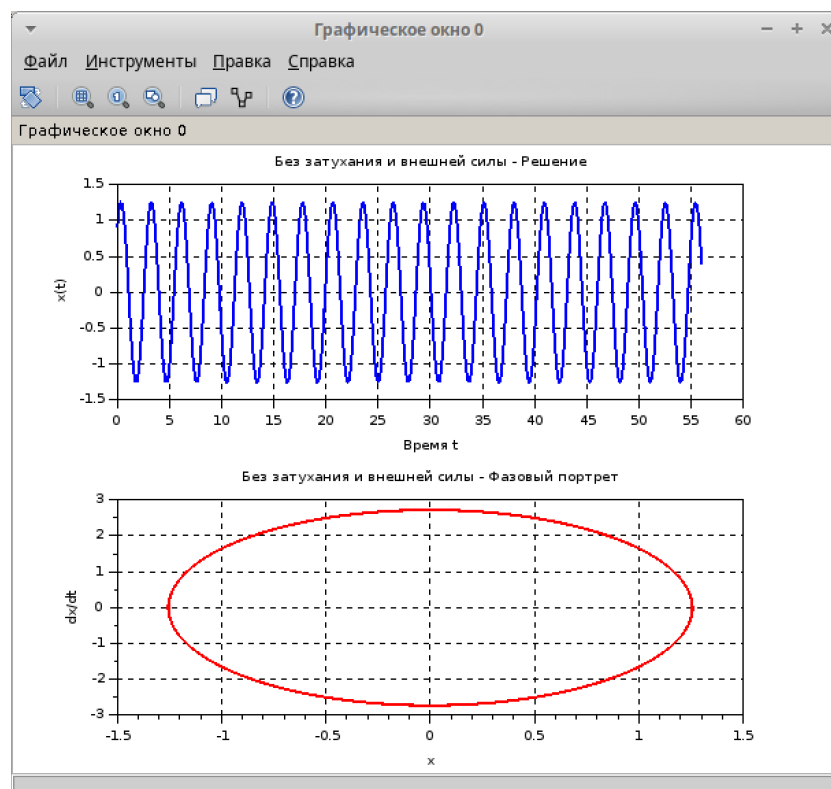


Рис. 4.1: Первый график и фазовый портрет (без затухания и внешней силы).



## 5 Выполнение лабораторной работы

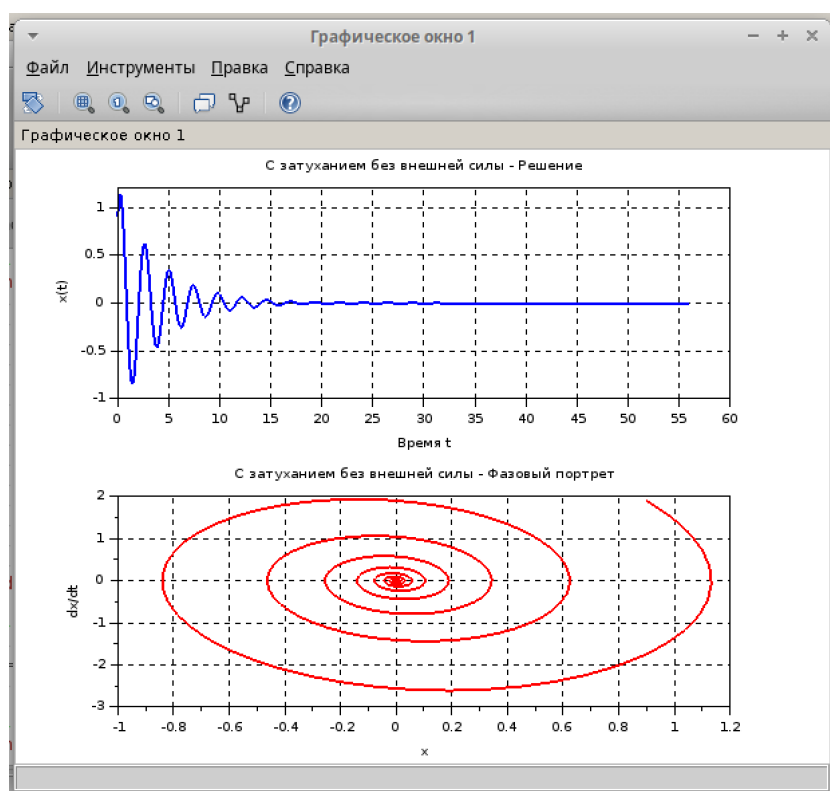


Рис. 5.1: Второй график и фазовый портрет (с затуханием и без внешней силы).

## 6 Выполнение лабораторной работы

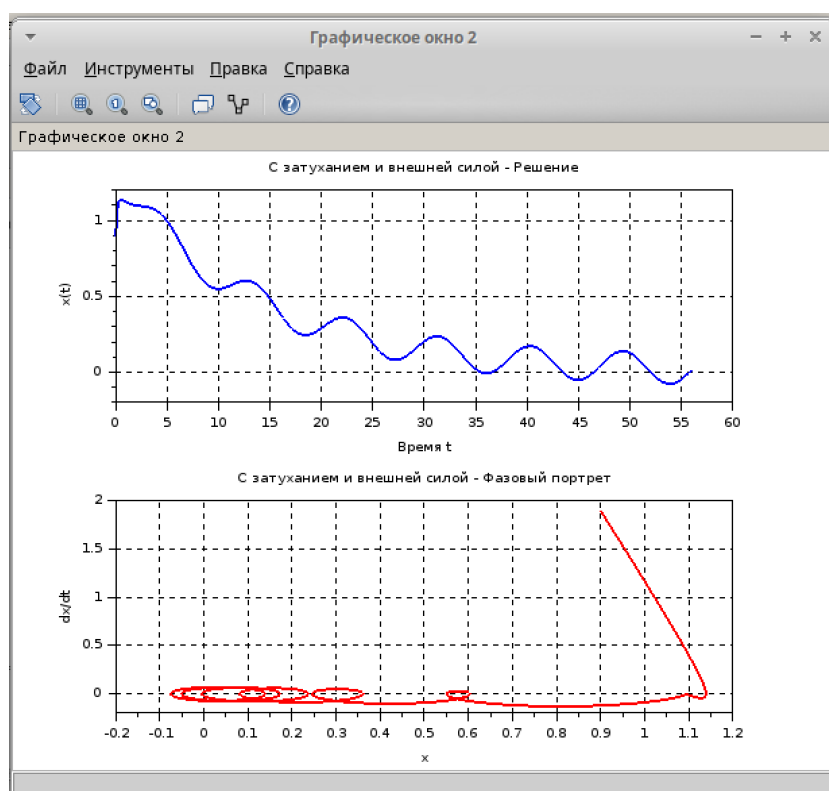


Рис. 6.1: Третий график и фазовый портрет (с затуханием и с внешней силой).

## 7 Выполнение лабораторной работы

```
using DifferentialEquations ✓

using Plots ✓

# Функция для построения графиков

plot_oscillator (generic function with 1 method)
function plot_oscillator(sol, title)
    p1 = plot(sol, idxs=(0,1), xlabel="Время t", ylabel="x(t)",
              title="Решение: $title", legend=false, linewidth=2)
    p2 = plot(sol, idxs=(1,2), xlabel="x", ylabel="x'",
              title="Фазовый портрет: $title", legend=false, linewidth=2)
    plot(p1, p2, layout=(2,1), size=(800,600))
end

# Общие параметры

tspan = ▶ (0.0, 56.0)
tspan = (0.0, 56.0)

tstep = 0.05
tstep = 0.05

x0 = ▶ [0.9, 1.9]
x0 = [0.9, 1.9] # x(0)=0.9, x'(0)=1.9
# 1. Случай без затухания и внешней силы:  $x'' + 4.7x = 0$ 

oscillator1! (generic function with 1 method)
function oscillator1!(du, u, p, t)
    x, v = u
    du[1] = v # dx/dt = v
    du[2] = -4.7 * x # dv/dt = -4.7x
end

prob1 = ODEProblem{uType Vector{Float64} and tType Float64, In-place: true,
Non-trivial mass matrix: false,
timespan: (0.0, 56.0),
u0: 2-element Vector{Float64}:
0.9
1.9}
prob1 = ODEProblem(oscillator1!, x0, tspan)
```

Рис. 7.1: Код и соответствующие выводы в среде Pluto на языке Julia.

## 8 Выполнение лабораторной работы

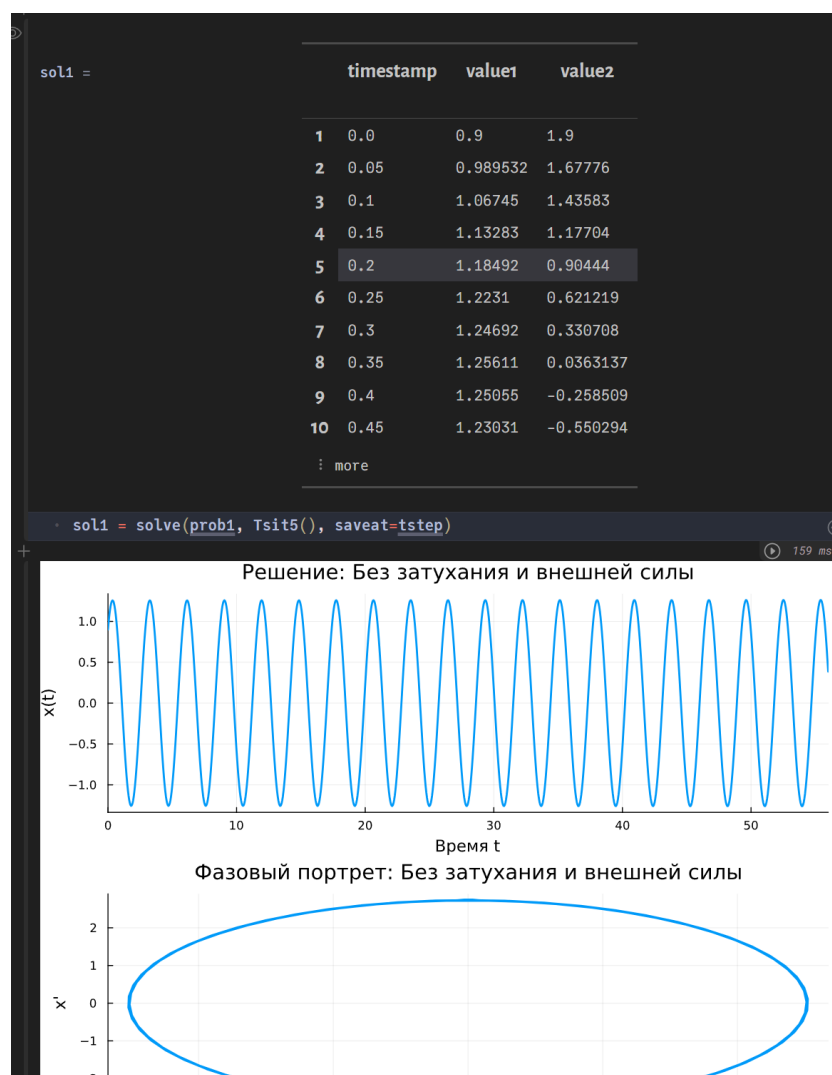


Рис. 8.1: Код и соответствующие выходы в среде Pluto на языке Julia.

## 9 Выполнение лабораторной работы

```
plot_oscillator(sol1, "Без затухания и внешней силы")
# 2. Случай с затуханием без внешней силы:  $x'' + 0.5x' + 7x = 0$ 

oscillator2! (generic function with 1 method)
function oscillator2!(du, u, p, t)
    x, v = u
    du[1] = v           #  $dx/dt = v$ 
    du[2] = -0.5*v - 7*x #  $dv/dt = -0.5v - 7x$ 
end

prob2 = ODEProblem{uType Vector{Float64} and tType Float64, In-place: true,
Non-trivial mass matrix: false,
timespan: (0.0, 56.0),
u0: 2-element Vector{Float64}:
0.9
1.9}
prob2 = ODEProblem{oscillator2!, x0, tspan}

sol2 =
```

	timestamp	value1	value2
1	0.0	0.9	1.9
2	0.05	0.985751	1.52657
3	0.1	1.05237	1.13602
4	0.15	1.09919	0.735508
5	0.2	1.12588	0.332187
6	0.25	1.13248	-0.0669059
7	0.3	1.11937	-0.454991
8	0.35	1.08727	-0.825645
9	0.4	1.0372	-1.17289
10	0.45	0.970458	-1.49133
	⋮ more		

```
sol2 = solve(prob2, Tsit5(), saveat=tstep)
```

Рис. 9.1: Код и соответствующие выходы в среде Pluto на языке Julia.

## 10 Выполнение лабораторной работы

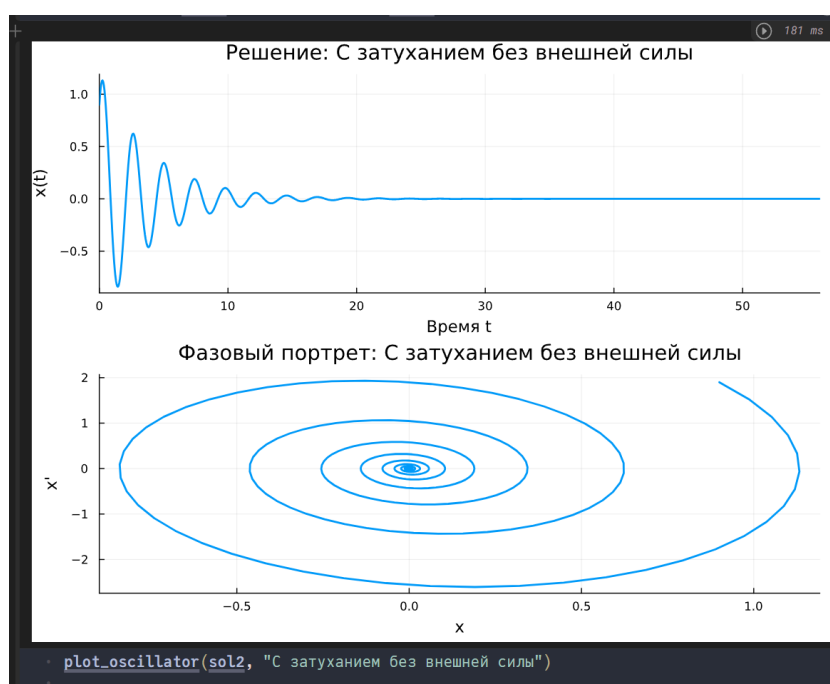


Рис. 10.1: Код и соответствующие выводы в среде Pluto на языке Julia.



## 12 Выполнение лабораторной работы

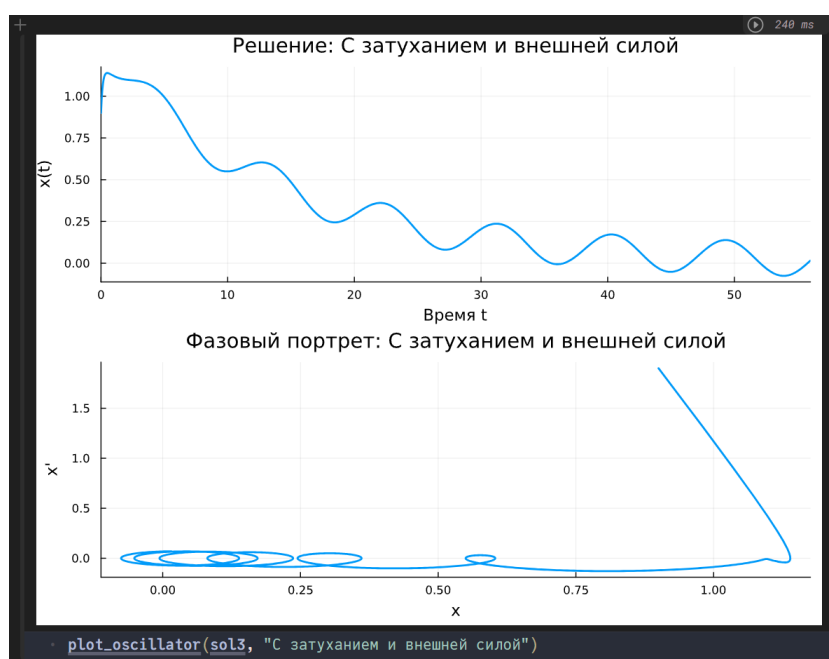


Рис. 12.1: Код и соответствующие выходы в среде Pluto на языке Julia.



## **13 Вывод**

**В ходе данной лабораторной работы построил модель гармонического осциллятора с разными параметрами в обеих средах программирования и вывел графики и фазовые портреты при соблюдении разных условий работы маятника.**