

Лабораторная работа №4

Математическая модель гармонических колебаний

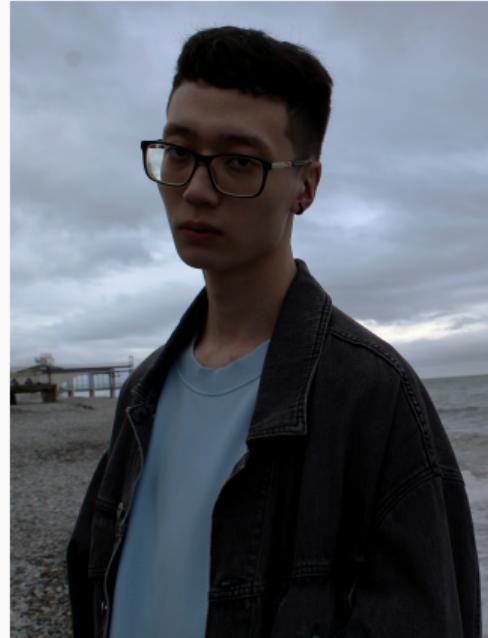
Ким М. А.

03 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Ким Михаил Алексеевич
- студент уч. группы НФИбд-01-20
- Российский университет дружбы народов
- 1032201664@pfur.ru
- <https://github.com/exmanka>



Вводная часть

Актуальность

- Необходимость навыков моделирования реальных математических задач, построение графиков.

Объект и предмет исследования

- Язык программирования Julia
- Язык моделирования Modelica
- Математическая модель гармонических колебаний

Цели и задачи

- Продолжить знакомство с функционалом языков Julia и OpenModelica.
- Смоделировать математические модели гармонических колебаний с помощью данных языков.
- Построить графики состояния систем в соответствии с поставленными задачами.

Материалы и методы

- Языки программирования:
 - Julia
 - OpenModelica

Процесс выполнения работы

Формулировка задания

Формулировка задания

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

1. Колебания гармонического осциллятора без затуханий и без действий внешней силы $\ddot{x} + 2.5x = 0$.
2. Колебания гармонического осциллятора с затуханием и без действий внешней силы $\ddot{x} + 10\dot{x} + 11x = 0$.
3. Колебания гармонического осциллятора с затуханием и под действием внешней силы $\ddot{x} + \dot{x} + x = 3 \sin(t)$.

На интервале $t \in [0; 65]$ (шаг 0.05) с начальными условиями $x_0 = -1, y_0 = 2$.

Pluto.jl

Код задания №1

```
- begin
-   import Pkg
-   Pkg.activate() ⓘ
-   using DifferentialEquations
-   using LaTeXStrings
-   import Plots
- end
```

Activating project at `~/.julia/environments/v1.8` ⓘ

T

Период времени

```
- begin
-   const ω₀² = 2.5
-
-   "Начальные условия: u₀[1] == x₀, u₀[2] == y₀"
-   u₀ = [-1, 2]
-
-   "Период времени"
-   T = (0.0, 65.0)
- end
```

F!

Правая часть нашей системы, p, t не используются. u[1] == x, u[2] == y

```
- "Правая часть нашей системы, p, t не используются. u[1] == x, u[2] == y"
- function F!(du, u, p, t)
-   du[1] = u[2]
-   du[2] = - ω₀² * u[1]
- end
```

```
prob = ODEProblem with uType Vector{Int64} and tType Float64. In-place: true
  timespan: (0.0, 65.0)
  u₀: 2-element Vector{Int64}:
    -1
    2
prob = ODEProblem(F!, u₀, T)
```

sol =

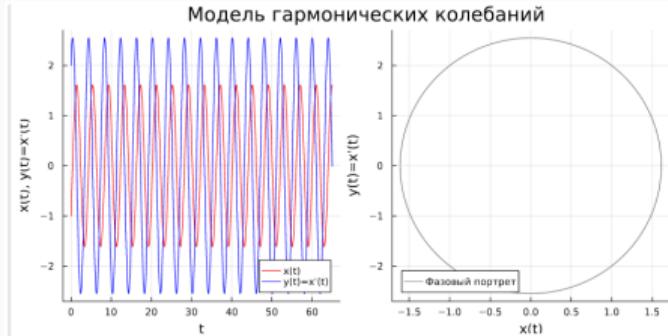
	timestamp	value1	value2
1	0.0	-1.0	2.0
2	0.05	-0.896981	2.11862
3	0.1	-0.788358	2.22401
4	0.15	-0.674811	2.31551
5	0.2	-0.557049	2.39254
6	0.25	-0.435806	2.45462
7	0.3	-0.311842	2.50138
8	0.35	-0.185929	2.5325
9	0.4	-0.0588552	2.54781
10	0.45	0.0685865	2.5472
	: more		

```
sol = solve(prob, saveat=0.05, abstol=1e-8, reltol=1e-8)
```

```
▶ [0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8]
```

```
- begin
-   const xx = []
-   const yy = []
-   for u in sol.u
-     x, y = u
-     push!(xx, x)
-     push!(yy, y)
-   end
-   Time = sol.t
-   Time
- end
```

Графики задания №1



```
begin
    fig = Plots.plot(
        layout=(1, 2),
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        # aspect_ratio=:equal,
        size=(800, 400),
        plot_title="Модель гармонических колебаний"
    )
    Plots.plot!(
        fig[1],
        Time,
        [xx, yy],
        color=[:red :blue],
        xlabel="t",
        ylabel="x(t), y(t)=x'(t)",
        label=["x(t)" "y(t)=x'(t)"]
    )
    Plots.plot!(
        fig[2],
        xx,
        yy,
        color=[:gray],
        xlabel="x(t)",
        ylabel="y(t)=x'(t)",
        label="фазовый портрет"
    )
end
```

Измененные блоки кода для задания №2. Получившиеся графики

T

Период времени

```
begin
  const  $\omega_0^2 = 11.0$ 
  const  $y \cdot 2 = 10.0$ 

  "Начальные условия:  $u_0[1] = x_0$ ,  $u_0[2] = y_0$ "
   $u_0 = [-1, 2]$ 

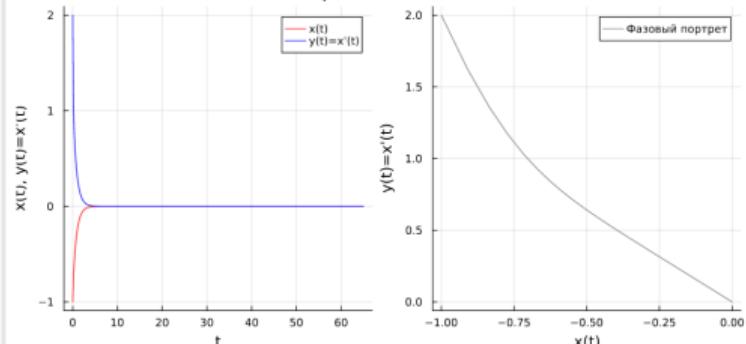
  "Период времени"
  T = (0.0, 65.0)
end
```

F!

Правая часть нашей системы, p, t не используются. u[1] – x, u[2] – y

```
"Правая часть нашей системы, p, t не используются. u[1] – x, u[2] – y"
function F!(du, u, p, t)
  du[1] = u[2]
  du[2] = -y \cdot 2 * u[2] -  $\omega_0^2$  * u[1]
end
```

Модель гармонических колебаний



Измененные блоки кода для задания №3. Получившиеся графики

T

Период времени

```
begin
  const  $\omega_0^2$  = 1.0
  const  $\gamma^2$  = 1.0

  function f(t)
    return 3 * sin(t)
  end

  "Начальные условия: u[1] -- x0, u[2] -- y0"
  u0 = [-1, 2]

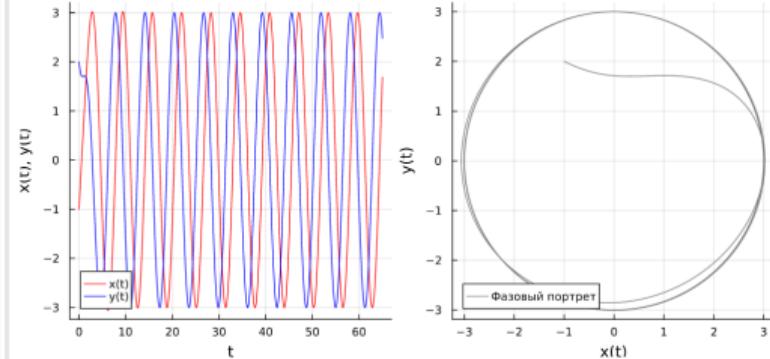
  "Период времени"
  T = (0.0, 65.0)
end
```

F!

Правая часть нашей системы, p, t не используются. u[1] -- x, u[2] -- y

```
"Правая часть нашей системы, p, t не используются. u[1] -- x, u[2] -- y"
function F!(du, u, p, t)
  du[1] = u[2]
  du[2] = - $\gamma^2$  * u[2] -  $\omega_0^2$  * u[1] + f(t)
end
```

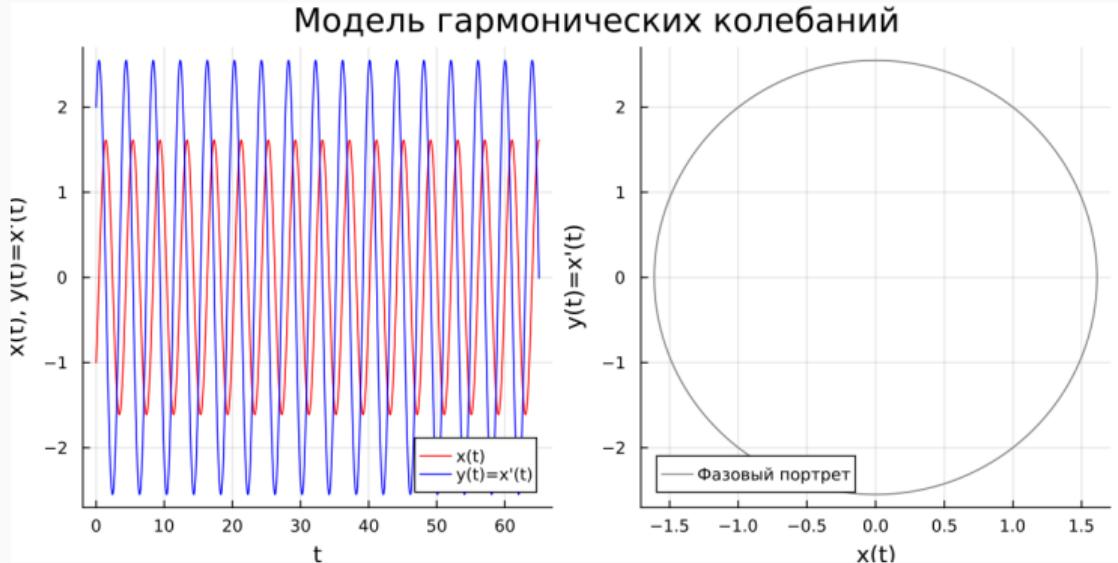
Модель гармонических колебаний



Julia

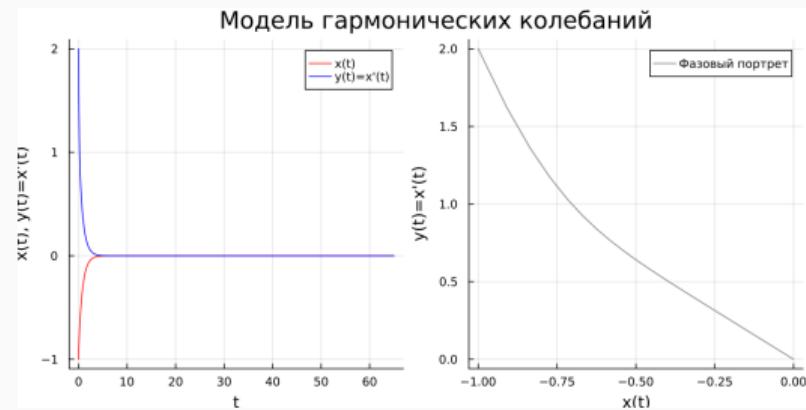
Код задания №1. Получившиеся графики

```
1  using DifferentialEquations
2  using Plots
3
4  const w0^2 = 2.5
5
6  "Начальные условия: u[1] -- x0, u[2] -- y0"
7  u0 = [-1, 2]
8
9  "Период времени"
10 T = (0.0, 65.0)
11
12 "Правая часть нашей системы, p, t не используются. u[1] -- x, u[2] -- y"
13 function F!(du, u, p, t)
14     du[1] = u[2]
15     du[2] = - w0^2 * u[1]
16 end
17
18
19 prob = ODEProblem(F!, u0, T)
20 sol = solve(prob, saveat=0.05, abstol=1e-8, reltol=1e-8)
21
22 const xx = []
23 const yy = []
24 for u in sol.u
25     x, y = u
26     push!(xx, x)
27     push!(yy, y)
28 end
29 Time = sol.t
30
31 fig = Plots.plot(
32     layout=(1, 2),
33     dpi=150,
34     grid=:xy,
35     gridcolor=:black,
36     gridwidth=1,
37     # aspect ratio=:equal,
38     size=(800, 400),
39     plot_title="Модель гармонических колебаний"
40 )
41
42 Plots.plot!(
43     fig[1],
44     Time,
45     [xx, yy],
46     color=[:red :blue],
47     xlabel="t",
48     ylabel="x(t), y(t)=x'(t)",
49     label=[["x(t)" "y(t)=x'(t)"]]
50 )
51
52 Plots.plot!(
53     fig[2],
54     xx,
55     yy,
56     color=[:gray],
57     xlabel="x(t)",
58     ylabel="y(t)=x'(t)",
59     label="Фазовый портрет"
60 )
61
62 savefig(fig, ".../lab4_1")
```



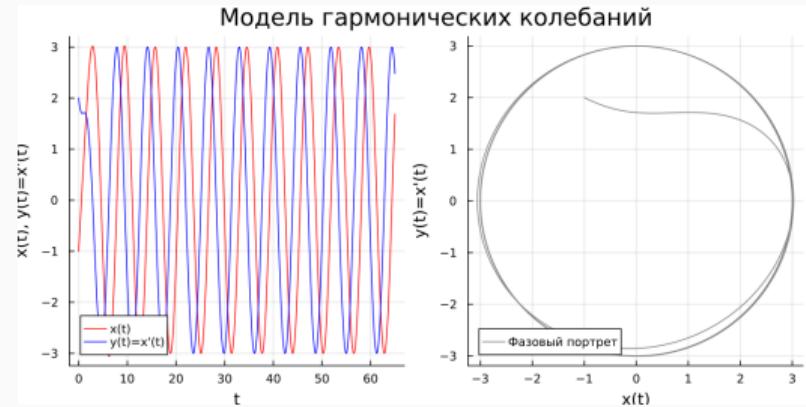
Измененные блоки кода для задания №2. Получившиеся графики

```
4 const  $\omega_0^2$  = 11.0
5 const  $\gamma^2$  = 10.0
6
7 "Начальные условия: u0[1] -- x0, u0[2] -- y0"
8 u0 = [-1, 2]
9
10 "Период времени"
11 T = (0.0, 65.0)
12
13 "Правая часть нашей системы, p, t не используются. u[1] -- x, u[2] -- y"
14 function F!(du, u, p, t)
15     du[1] = u[2]
16     du[2] = - $\gamma^2$  * u[2] -  $\omega_0^2$  * u[1]
17 end
```



Измененные блоки кода для задания №3. Получившиеся графики

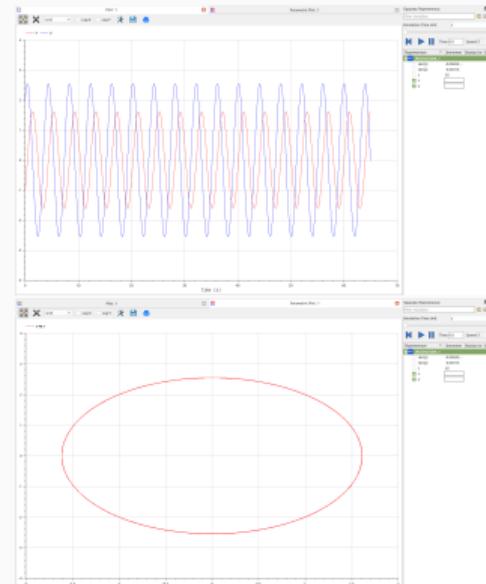
```
4 const w02 = 1.0
5 const g2 = 1.0
6 function f(t)
7     return 3 * sin(t)
8 end
9
10 "Начальные условия: u0[1] -- x0, u0[2] -- y0"
11 u0 = [-1, 2]
12
13 "Период времени"
14 T = {0.0, 65.0}
15
16 "Правая часть нашей системы, p, t не используются. u[1] -- x, u[2] -- y"
17 function F!(du, u, p, t)
18     du[1] = u[2]
19     du[2] = -g2* u[2] - w02 * u[1] + f(t)
20 end
```



OpenModelica

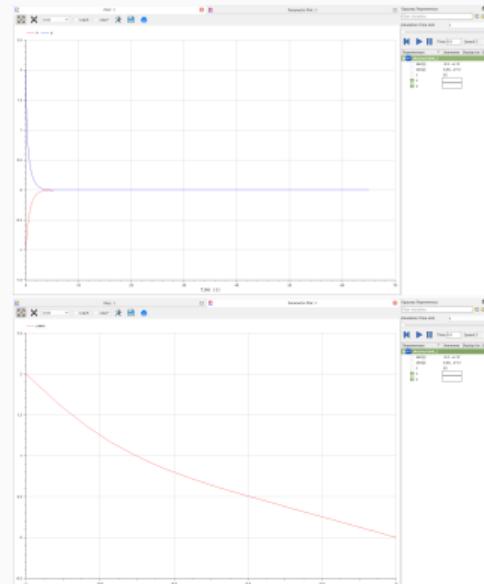
Код задания №1. Получившиеся графики

```
1 model lab4_1
2   constant Real omega_0_square = 2.5;
3   Real t = time;
4   Real x;
5   Real y;
6 initial equation
7   x = -1;
8   y = 2;
9 equation
10  der(x) = y;
11  der(y) = - omega_0_square * x;
12  annotation(experiment(StartTime=0, StopTime=65, Interval = 0.05));
13 end lab4_1;
```



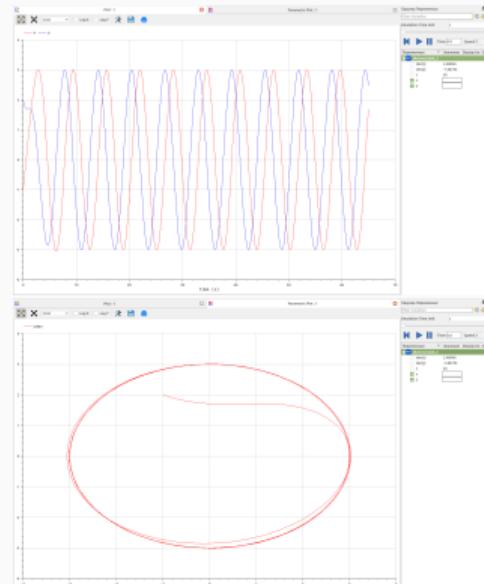
Код задания №2. Получившиеся графики

```
1 model lab4_2
2   constant Real omega_0_square = 11.0;
3   constant Real gamma_2 = 10.0;
4   Real t = time;
5   Real x;
6   Real y;
7 initial equation
8   x = -1;
9   y = 2;
10 equation
11   der(x) = y;
12   der(y) = - gamma_2*y - omega_0_square*x;
13   annotation(experiment(StartTime=0, StopTime=65, Interval = 0.05));
14 end lab4_2;
```



Код задания №3. Получившиеся графики

```
1 model lab4_3
2   constant Real omega_0_square = 1.0;
3   constant Real gamma_2 = 1.0;
4   Real t = time;
5   Real x;
6   Real y;
7 initial equation
8   x = -1;
9   y = 2;
10 equation
11   der(x) = y;
12   der(y) = - gamma_2 * y - omega_0_square * x + 3 * sin(t);
13   annotation(experiment(StartTime=0, StopTime=65, Interval = 0.05));
14 end lab4_3;
```



Результаты

Результаты

- Смоделированы математические модели гармонических колебаний с помощью данных языков.
- Построены графики состояния систем в соответствии с поставленными задачами.

Вывод

Продолжил знакомство с функционалом языка программирования Julia и языка моделирования Modelica, а также с функционалом программного обеспечения OpenModelica. Используя эти средства, построил математические модели гармонических колебаний.