

Презентация по защите лабораторной работы №4

По предмету Математическое моделирование

Максимов А. А.

18 февраля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Максимов Алексей Александрович
- Российский университет дружбы народов
- https://github.com/Leximus555/study_2022-2023_mathmod/edit/master/labs

Вариант № 32

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

1. Колебания гармонического осциллятора без затуханий и без действий внешней силы $\ddot{x} + 5.2x = 0$
2. Колебания гармонического осциллятора с затуханием и без действий внешней силы $\ddot{x} + 14\dot{x} + 0.5x = 0$
3. Колебания гармонического осциллятора с затуханием и под действием внешней силы $\ddot{x} + 13\dot{x} + 0.3x = 0.8\sin(9t)$

На интервале $t \in [0; 59]$ (шаг 0.05) с начальными условиями $x_0 = 0.5, y_0 = -1.5$

получили задачу (№32)

создали программу на julia, моделирующую колебания пружинного маятника и трех случаев (без затухания, с затуханием, с затуханием и внешней силой)

```
lab41.jl
1 using Plots
2 using DifferentialEquations
3
4 println("good")
5
6 a = 5.2
7 t0 = 0
8 tmax = 59
9
10 t = collect(LinRange(t0, tmax, 1200))
11 #t = collect(LinRange(t0, tmax, dt))
12
13
14 function syst(dx,x,p,t)
15     dx[1] = x[2]
16     dx[2] = -a * x[1]
17 end
18
19 t0 = 0
20
21 x0 = [0.5 ; -1.5]
22
23 x = ODEProblem(syst, x0, (t0, tmax))
24
25 #y = ODEProblem( функция, нач, диапазон)
26
27 sol = solve(x)
28
29 plot(sol)
30
31 savefig("C:\\Users\\maks\\OneDrive\\Рабочий
32 стол\\unik2.0\\Математическое
33 Моделирование\\julia\\julia1ab4j101.png")
34
35 y1=[]
36 y2=[]
37
38 for i in t
39     push!(y1, sol(i)[1])
40     push!(y2, sol(i)[2])
41 end
42
43 plot(y1, y2)
```

```
lab42.jl
1 using Plots
2 using DifferentialEquations
3
4 println("good")
5
6 a = 14
7 b = 0.5
8 t0 = 0
9 tmax = 59
10
11 t = collect(LinRange(t0, tmax, 1200))
12
13
14 function syst(dx,x,p,t)
15     dx[1] = x[2]
16     dx[2] = -a * x[1] - b * x[2]
17 end
18
19 t0 = 0
20
21 x0 = [0.5 ; -1.5]
22
23 x = ODEProblem(syst, x0, (t0, tmax))
24
25 #y = ODEProblem( функция, нач, диапазон)
26
27 sol = solve(x)
28
29 plot(sol)
30
31 savefig("C:\\Users\\maks\\OneDrive\\Рабочий
32 стол\\unik2.0\\Математическое
33 Моделирование\\julia\\julia1ab4j102.png")
34
35 y1=[]
36 y2=[]
37
38 for i in t
39     push!(y1, sol(i)[1])
40     push!(y2, sol(i)[2])
41 end
42
43 plot(y1, y2)
```

```
lab43.jl
1 using Plots
2 using DifferentialEquations
3
4 println("good")
5
6 a = 13
7 b = 0.3
8 t0 = 0
9 tmax = 59
10
11 f = 0.9 * cos(2*pi*t)
12
13 function f(t)
14     return f
15 end
16
17 function syst(dx,x,p,t)
18     dx[1] = x[2]
19     dx[2] = -a * x[1] - b * x[2] - f(t)
20 end
21
22 t0 = 0
23
24 x0 = [0.5 ; -1.5]
25
26 x = ODEProblem(syst, x0, (t0, tmax))
27
28 #y = ODEProblem( функция, нач, диапазон)
29
30 sol = solve(x)
31
32 plot(sol)
33
34 savefig("C:\\Users\\maks\\OneDrive\\Рабочий
35 стол\\unik2.0\\Математическое
36 Моделирование\\julia\\julia1ab4j103.png")
37
38 y1=[]
39 y2=[]
40
41 for i in t
42     push!(y1, sol(i)[1])
43     push!(y2, sol(i)[2])
44 end
45
46 plot(y1, y2)
```

Поработали с Julia и OpenModelica и решили задачу.