

Лабораторная работа №4

Модель гармонических колебаний

Шубнякова Дарья НКНбд-01-22

1. Вводная часть

2. Основная часть

3. Результаты

1. Вводная часть

1.1 Цели и задачи

Ознакомиться с задачей гармонических колебаний. Реализовать данную модель на языке Julia, а так же в среде OMEdit.

Постройте фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора для следующих случаев. Наш вариант 13.

2. Основная часть

2.1 Выполнение лабораторной работы

2.1.1 Прописываем наш код на языке Julia в JupiterNotebook.

[2]:

```
using DifferentialEquations, Plots

# Общие параметры
tspan = (0.0, 75.0)
tsteps = 0.0:0.05:75.0
u0 = [1.0, 0.0] # Начальные условия:  $x=1$ ,  $v=0$ 

# Случай 1: Без затухания, без внешней силы
function oscillator1!(du, u, p, t)
    x, v = u
    w = sqrt(6.5)
    du[1] = v #  $dx/dt = v$ 
    du[2] = -w^2 * x #  $dv/dt = -\omega^2 x$ 
end

# Случай 2: С затуханием, без внешней силы
function oscillator2!(du, u, p, t)
    x, v = u
    gamma = 4.0
    w = sqrt(5.0)
    du[1] = v #  $dx/dt = v$ 
    du[2] = -gamma*v - w^2*x #  $dv/dt = -\gamma v - \omega^2 x$ 
end

# Случай 3: С затуханием и внешней силой
function oscillator3!(du, u, p, t)
    x, v = u
    gamma = 3.0
    w = sqrt(7.0)
    du[1] = v #  $dx/dt = v$ 
    du[2] = -gamma*v - w^2*x + sin(2*t) #  $dv/dt = -\gamma v - \omega^2 x + \sin(2t)$ 
end

# Решение всех трех случаев
prob1 = ODEProblem(oscillator1!, u0, tspan)
sol1 = solve(prob1, TsitF(), saveat=tsteps)
```

2.2 Продолжение кода.

```
prob3 = ODEProblem(oscillator3!, u0, tspan)
sol3 = solve(prob3, Tsit5(), saveat=tsteps)

# Построение решений во времени
plt_time = plot(layout=(3,1), size=(800, 900), dpi=300)

plot!(plt_time[1], sol1.t, sol1[1,:], label="x(t)", linewidth=2,
      title="Случай 1: Без затухания, без внешней силы", xlabel="Время", ylabel="x(t)")
plot!(plt_time[1], sol1.t, sol1[2,:], label="v(t)", linewidth=2)

plot!(plt_time[2], sol2.t, sol2[1,:], label="x(t)", linewidth=2,
      title="Случай 2: С затуханием, без внешней силы", xlabel="Время", ylabel="x(t)")
plot!(plt_time[2], sol2.t, sol2[2,:], label="v(t)", linewidth=2)

plot!(plt_time[3], sol3.t, sol3[1,:], label="x(t)", linewidth=2,
      title="Случай 3: С затуханием и внешней силой", xlabel="Время", ylabel="x(t)")
plot!(plt_time[3], sol3.t, sol3[2,:], label="v(t)", linewidth=2)

# Построение фазовых портретов
plt_phase = plot(layout=(1,3), size=(1200, 400))

plot!(plt_phase[1], sol1[1,:], sol1[2:], label="Фазовая траектория", linewidth=2,
      title="Случай 1: Фазовый портрет", xlabel="x", ylabel="v", legend=false)
scatter!(plt_phase[1], [1], [0], label="Начальная точка", color=:red)

plot!(plt_phase[2], sol2[1:], sol2[2:], label="Фазовая траектория", linewidth=2,
      title="Случай 2: Фазовый портрет", xlabel="x", ylabel="v", legend=false)
scatter!(plt_phase[2], [1], [0], label="Начальная точка", color=:red)

plot!(plt_phase[3], sol3[1:], sol3[2:], label="Фазовая траектория", linewidth=2,
      title="Случай 3: Фазовый портрет", xlabel="x", ylabel="v", legend=false)
scatter!(plt_phase[3], [1], [0], label="Начальная точка", color=:red)

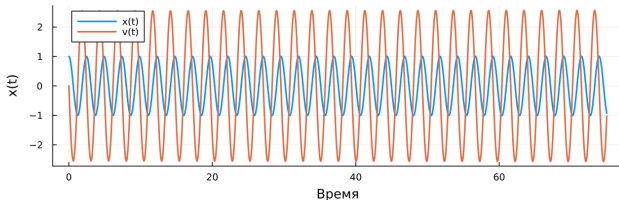
# Отображение графиков
display(plt_time)
display(plt_phase)
```

2.3 Получаем первые две модели: Случай без затухания, без внешней силы и Случай с затуханием, но без внешней силы.

```
# Сохранение результатов
savefig(plt_time, "harmonic_oscillators_time.png")
savefig(plt_phase, "harmonic_oscillators_phase.png")

println("Графики сохранены в файлы:")
println("- harmonic_oscillators_time.png - временные зависимости")
println("- harmonic_oscillators_phase.png - фазовые портреты")
```

Случай 1: Без затухания, без внешней силы

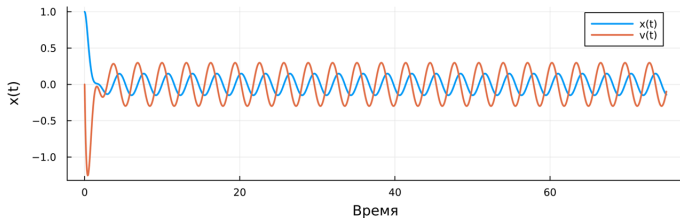


Случай 2: С затуханием, без внешней силы

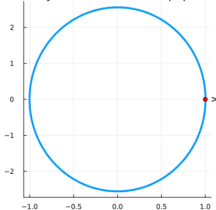


2.4 Получаем третий случай: с затуханием и внешней силой, а также фазовые портреты для всех трех случаев.

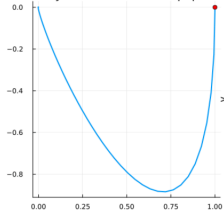
Случай 3: С затуханием и внешней силой



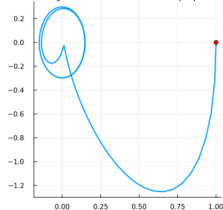
Случай 1: Фазовый портрет



Случай 2: Фазовый портрет



Случай 3: Фазовый портрет

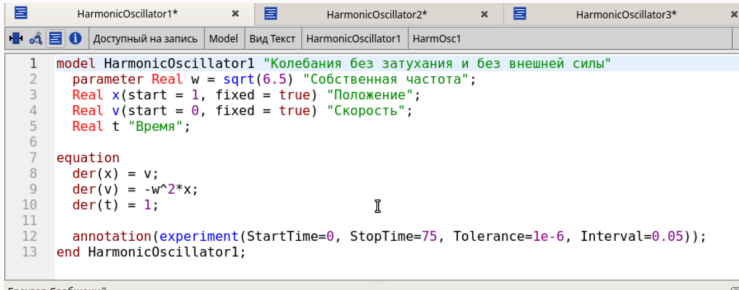


Графики сохранены в файлы:

– harmonic_oscillators_time.png – временные зависимости

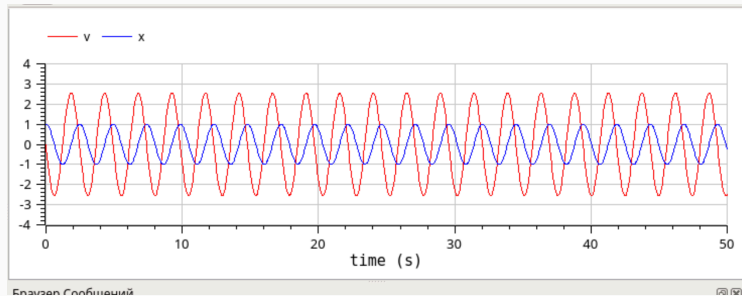
harmonic_oscillators_phase.png – фазовые портреты

2.5 Прописываем код в реде OpenModelica для первого случая: без затухания и без внешней силы.

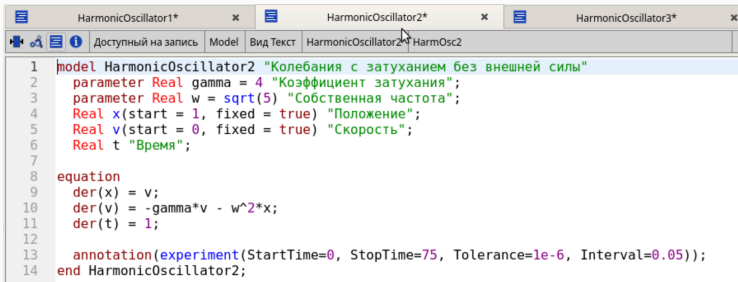


```
1 model HarmonicOscillator1 "Колебания без затухания и без внешней силы"
2   parameter Real w = sqrt(6.5) "Собственная частота";
3   Real x(start = 1, fixed = true) "Положение";
4   Real v(start = 0, fixed = true) "Скорость";
5   Real t "Время";
6
7   equation
8     der(x) = v;
9     der(v) = -w^2*x;
10    der(t) = 1;
11
12    annotation(experiment(StartTime=0, StopTime=75, Tolerance=1e-6, Interval=0.05));
13  end HarmonicOscillator1;
```

2.6 Получаем такой график.

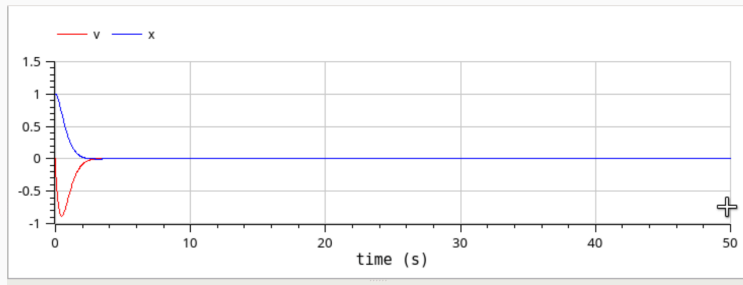


2.7 Прописываем код в реде OpenModelica для второго случая: без затухания, но с внешней силой.

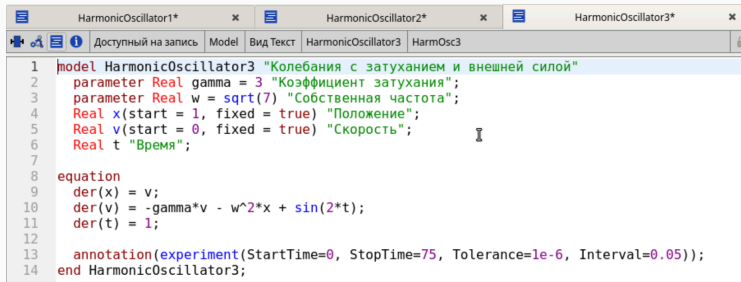


```
1 model HarmonicOscillator2 "Колебания с затуханием без внешней силы"
2   parameter Real gamma = 4 "Коэффициент затухания";
3   parameter Real w = sqrt(5) "Собственная частота";
4   Real x(start = 1, fixed = true) "Положение";
5   Real v(start = 0, fixed = true) "Скорость";
6   Real t "Время";
7
8 equation
9   der(x) = v;
10  der(v) = -gamma*v - w^2*x;
11  der(t) = 1;
12
13  annotation(experiment(StartTime=0, StopTime=75, Tolerance=1e-6, Interval=0.05));
14 end HarmonicOscillator2;
```

2.8 Получаем такой график.

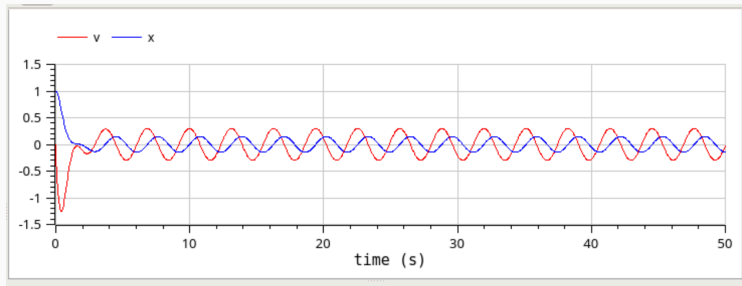


2.9 Прописываем код в реде OpenModelica для третьего случая: с затуханием и с внешней силой.



```
1 model HarmonicOscillator3 "Колебания с затуханием и внешней силой"
2   parameter Real gamma = 3 "Коэффициент затухания";
3   parameter Real w = sqrt(7) "Собственная частота";
4   Real x(start = 1, fixed = true) "Положение";
5   Real v(start = 0, fixed = true) "Скорость";
6   Real t "Время";
7
8 equation
9   der(x) = v;
10  der(v) = -gamma*v - w^2*x + sin(2*t);
11  der(t) = 1;
12
13  annotation(experiment(StartTime=0, StopTime=75, Tolerance=1e-6, Interval=0.05));
14 end HarmonicOscillator3;
```

2.10 Получаем такой график.



3. Результаты

3. Результаты

Мы реализовали модель в трех ее видах в OpenModelica и на языке Julia. На выходе получили две картинки: `harmonic_oscillators_phase.png` и `harmonic_oscillators_time.png`. Итоговый файл `lab4.ibybn` с кодом на языке Julia в JupiterNotebook. А также три файла для симуляции в OpenModelica: `HarmonicOscillator1.mo`, `HarmonicOscillator2.mo`, `HarmonicOscillator3.mo`