Отчёт по лабораторной работе №4

Вариант 67

Бабков Дмитрий Николаевич

# Цель работы

Смоделировать колебания гармонического осциллятора в разных условиях

# Задание

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

1. Колебания гармонического осциллятора без затуханий и без действий внешней силы
2. Колебания гармонического осциллятора c затуханием и без действий внешней силы
3. Колебания гармонического осциллятора c затуханием и под действием внешней силы

На интервале (шаг 0.05) с начальными условиями

# Теоретическое введение

Движение грузика на пружинке, маятника, заряда в электрическом контуре, а также эволюция во времени многих систем в физике, химии, биологии и других науках при определенных предположениях можно описать одним и тем же дифференциальным уравнением, которое в теории колебаний выступает в качестве основной модели. Эта модель называется линейным гармоническим осциллятором.

Уравнение свободных колебаний гармонического осциллятора имеет следующий вид:

где - переменная, описывающая состояние системы, – параметр, характеризующий потери энергии, – собственная частота колебаний, t – время.

# Выполнение работы

## Julia

Открыв Pluto.jl я приступил к написанию кода. Сначала я подключил библиотеки Plots и DiffetentialEquations:

using Plots  
using DiffetentialEquations

Далее я ввёл данные, приведённые в условии задачи:

tspan = (0, 33)  
dt = 0.05  
x0 = [1.3]  
y0 = [0.3]

Задал функцию, являющуюся дифференциальным уравнением для первого случая:

function harmonicOscillator(ddu, du, u, ω, t)  
 ddu .= -3.3 \* u  
end

С помощью методов SecondOrderODEProblem и solve решил это дифференциальное уравнение по начальным данным и вывел значения x и y в соответствующие массивы:

prob = SecondOrderODEProblem(harmonicOscillator, y0, x0, tspan)  
sol = solve(prob, dtmax = dt)  
diffX = [u[1] for u in sol.u]  
diffY = [u[2] for u in sol.u]

Далее с помощью метода plot я вывел фазовый портрет осциллятора, где ось x соответствует переменной x, а ось y - переменной y:

plt = plot(  
 diffY,  
 diffX,  
 xlabel = "x",  
 ylabel = "y"  
)



Аналогичным образом задал и решил дифф. уравнения для второго и третьего случаев:

function harmonicOscillator2(ddu, du, u, ω, t)  
 ddu .= -3\*du -0.3\*u  
end  
  
prob2 = SecondOrderODEProblem(harmonicOscillator2, y0, x0, tspan)  
sol2 = solve(prob2, dtmax = dt)  
diffX2 = [u[1] for u in sol2.u]  
diffY2 = [u[2] for u in sol2.u]  
  
function harmonicOscillator3(ddu, du, u, ω, t)  
 ddu .= - 3.3 \* du .- 3 \* u .+ 3.3 \* sin(3 \* t)  
end  
  
prob3 = SecondOrderODEProblem(harmonicOscillator3, y0, x0, tspan)  
sol3 = solve(prob3, dtmax = dt)  
diffX3 = [u[1] for u in sol3.u]  
diffY3 = [u[2] for u in sol3.u]

И вывел их фазовые портреты:

plt2 = plot(  
 diffY2,  
 diffX2,  
 xlabel = "x",  
 ylabel = "y"  
)  
  
plt3 = plot(  
 diffY3,  
 diffX3,  
 xlabel = "x",  
 ylabel = "y"  
)

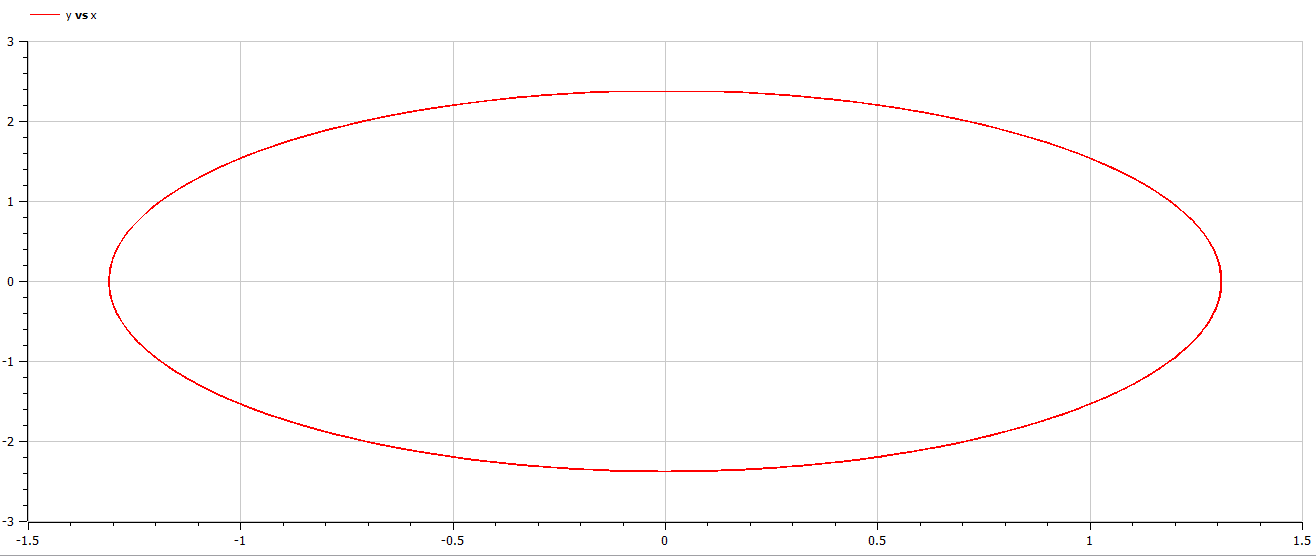
 

## OpenModelica

Открыв OpenModelica я написал код для трёх случаев, задав начальные значения и , а также дифференциальные уравнения гармонического осциллятора для трёх случаев:

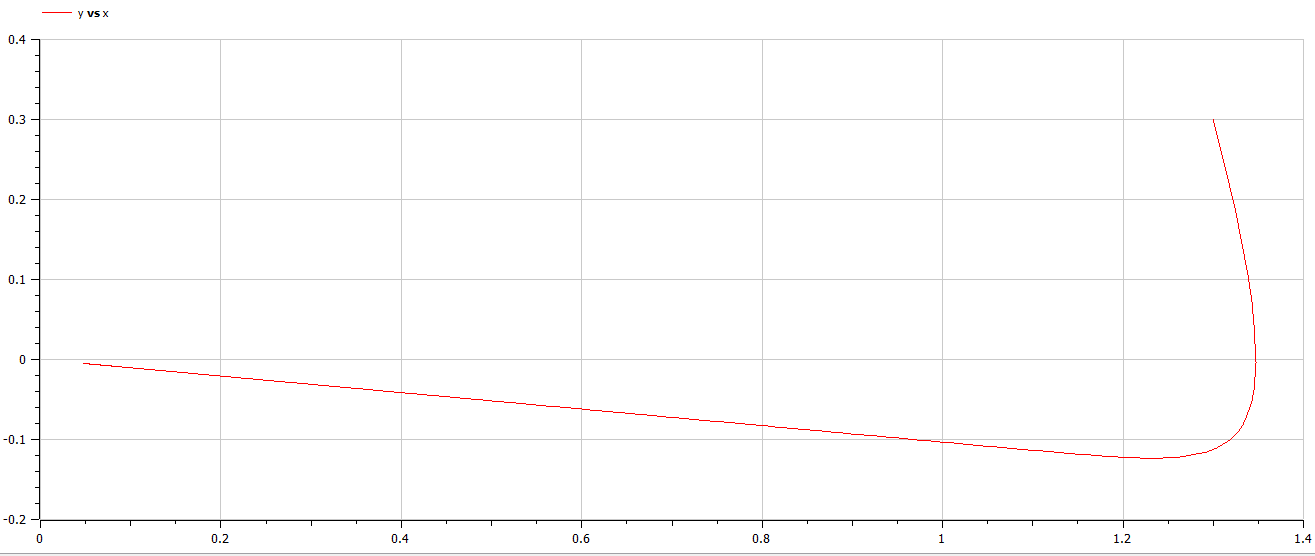
* Первый случай:

model lab04\_1  
 Real x;  
 Real y;  
initial equation  
 x = 1.3;  
 y = 0.3;  
equation  
 der(x) = y;  
 der(y) = -3.3 \* x;  
end lab04\_1;



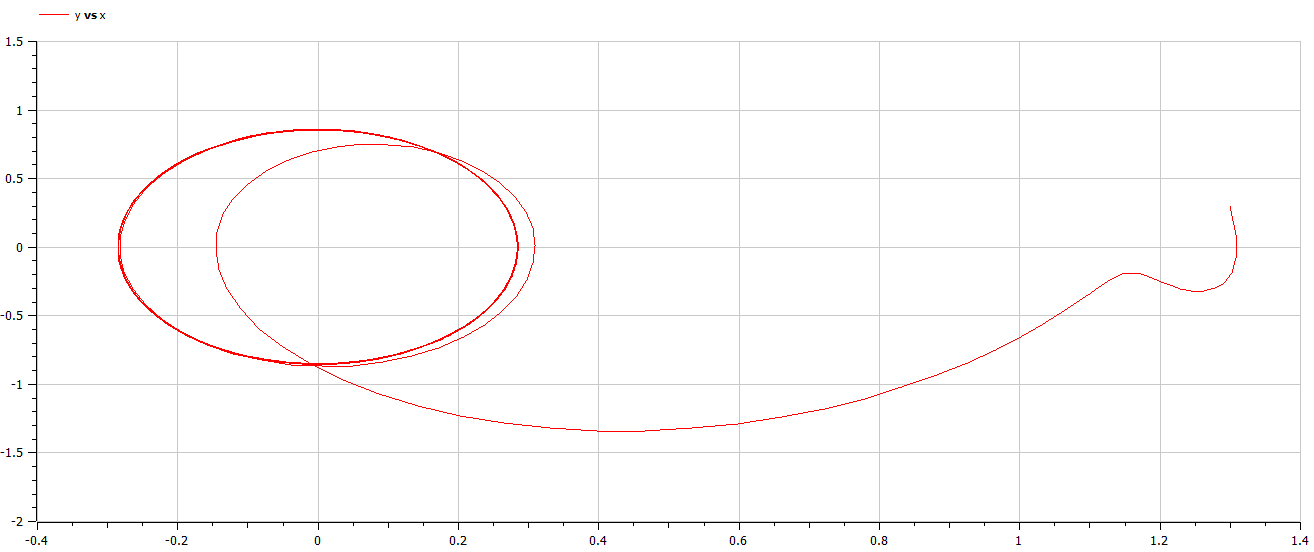
* Второй случай:

model lab04\_2  
 Real x;  
 Real y;  
initial equation  
 x = 1.3;  
 y = 0.3;  
equation  
 der(x) = y;  
 der(y) = - 3 \* y - 0.3 \* x;  
end lab04\_2;

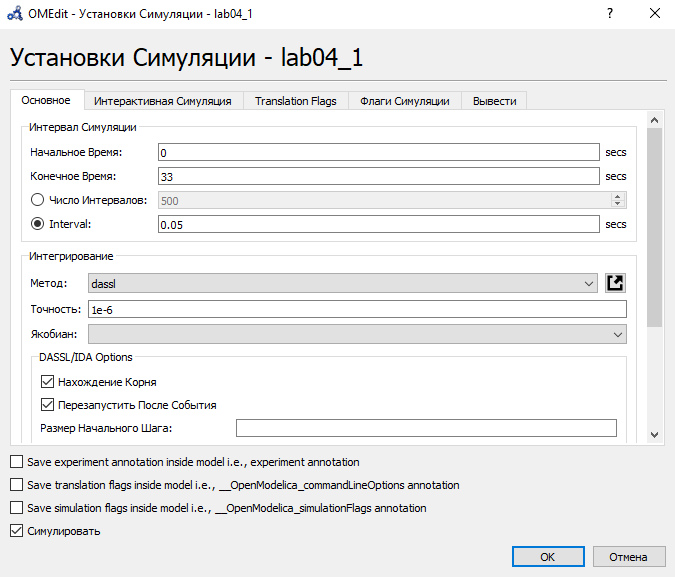


* Третий случай:

model lab04\_3  
 Real x;  
 Real y;  
initial equation  
 x = 1.3;  
 y = 0.3;  
equation  
 der(x) = y;  
 der(y) = - 3.3 \* y - 3 \* x + 3.3 \* sin(3 \* time);  
end lab04\_3;



* Задание условий симуляции (одинаково для каждого случая):



# Вывод

Модель была построена на языках Julia и OpenModelica, результаты идентичные, на OpenModelica выполнение задания и анализ полученных результатов проще.