Отчёт по лабораторной работе №4

дисциплина: Математическое моделирование

Ерёменко Артём Геннадьевич, НПИбд-02-18

Содержание

[Цель работы 1](#_Toc65861182)

[Задание 1](#_Toc65861183)

[Выполнение лабораторной работы 1](#_Toc65861184)

[Выводы 8](#_Toc65861185)

[Ответы на вопросы к лабораторной работе 8](#_Toc65861186)

# Цель работы

Построить модель гармонических колебаний с помощью Julia.

# Задание

**Вариант 4** Постройте 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

# Выполнение лабораторной работы

**1. Колебания без затуханий и без действий внешней силы**

1.1. Уравнение свободных колебаний гармонического осциллятора имеет следующий вид:

Изучил начальные условия. Первая формула - это уравнение осциллятора, при котором в системе нет потерь  
, поэтому, . Частота колебаний . . Правая часть уравнения .

1.2. Ввёл начальные условия в код на Julia:

w1 = 15^0.5  
g1 = 0\*0.5  
  
par1 = [w1,g1]  
  
x0 = [0.0,2.0]

1.3. Решение ищу на интервале (шаг не получается задать, так как в ODEProblem(prob, v0, t, p) t обязательно должно быть парой (t0,tmax)), значит, – начальный момент времени, – предельный момент времени.

1.4. Добавил в программу условия, описывающие время:

t0 = 0.0  
tmax = 55.0  
  
t = (t0,tmax)

1.5. Представил заданное уравнение второго порядка в виде системы двух уравнений первого порядка:

function dx1(dv,v,p,t)  
 w,g = p  
 x1,x2 = v  
 dv[1] = x2;  
 dv[2] = -w.\* w.\* x1;  
end

1.6. Поместил начальные условия в решатель системы уравнений:

x1 = ODEProblem(dx1,x0,t,par1)  
sol1 = solve(x1)

1.7. Переписал отдельно в , а в :

n1 = size(sol1,2)  
  
y11 = Array{Float16}(undef, n1)  
y21 = Array{Float16}(undef, n1)  
for i = 1: n1  
 y11[i] = sol1[1, i]  
 y21[i] = sol1[2, i]  
end

1.8. Построил фазовый портрет:

plot(y11,y21)

**2. Колебания c затуханием и без действий внешней силы**

2.1. Изучила начальные условия. - потеря энергии. Собственная частота колебаний . и остались такими же. Правая часть уравнения такая же, как и в п. 1.1.

2.2. Начальные значения переменных одинаковы для всех пунктов задачи. Остальные начальные условия оформил в код на Julia:

w2 = 1^0.5  
g2 = 10\*0.5  
  
par2 = [w2,g2]

2.3. Интервал решения такой же.

2.4. Представил заданное уравнение второго порядка в виде системы двух уравнений первого порядка:

function dx2(dv,v,p,t)  
 w,g = p  
 x1,x2 = v  
 dv[1] = x2;  
 dv[2] = -w.\* w.\* x1 - 2\*g.\* x2 ;  
end

2.5. Поместил начальные условия в решатель системы уравнений:

x2 = ODEProblem(dx2,x0,t,par2)  
sol2 = solve(x2)

2.6. Переписал отдельно в , а в :

n2 = size(sol2,2)  
  
y12 = Array{Float16}(undef, n1)  
y22 = Array{Float16}(undef, n1)  
for i = 1: n2  
 y12[i] = sol2[1, i]  
 y22[i] = sol2[2, i]  
end

2.7. Построил фазовый портрет:

plot(y12,y22)

**3. Колебания c затуханием и под действием внешней силы**

3.1. Изучила начальные условия. - потеря энергии. Собственная частота колебаний . и остались такими же. Правая часть уравнения .

2.2. Начальные значения переменных одинаковы для всех пунктов задачи. Остальные начальные условия оформил в код на Julia:

w3 = 1^0.5  
g3 = 3\*0.5  
  
par3 = [w3,g3]  
  
function f(t)  
 return sin(3 \* t)  
end

2.3. Интервал решения такой же.

2.4. Представил заданное уравнение второго порядка в виде системы двух уравнений первого порядка:

function dx3(dv,v,p,t)  
 w,g = p  
 x1,x2 = v  
 dv[1] = x2;  
 dv[2] = -w.\* w.\* x1 - 2\*g.\* x2 - f(t);  
end

2.5. Поместил начальные условия в решатель системы уравнений:

x3 = ODEProblem(dx3,x0,t,par3)  
sol3 = solve(x3)

2.6. Переписал отдельно в , а в :

n3 = size(sol3,2)  
  
y13 = Array{Float16}(undef, n1)  
y23 = Array{Float16}(undef, n1)  
for i = 1: n2  
 y13[i] = sol3[1, i]  
 y23[i] = sol3[2, i]  
end

2.7. Построил фазовый портрет:

plot(y13,y23)

**4. Сборка программы**

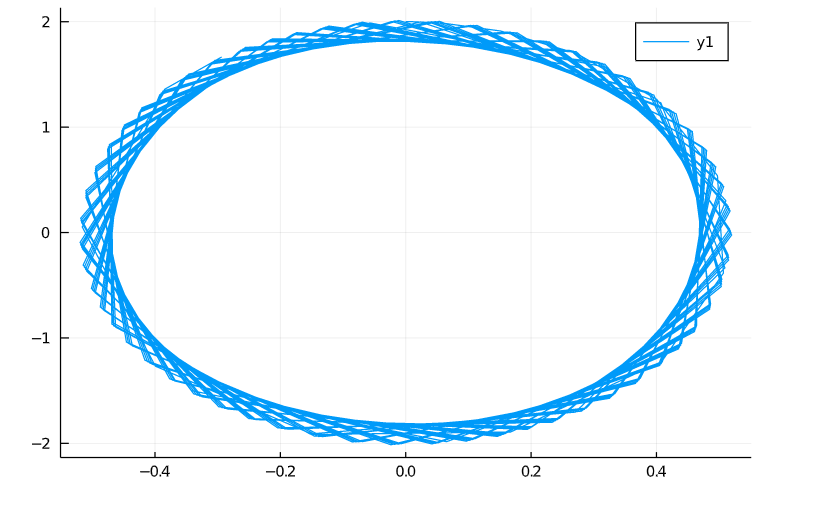
4.1. Код целиком:

using DifferentialEquations, Plots   
#Параметры осциллятора  
#x'' + 2\*g\* x' + w^2\* x = f(t)  
#w - частота  
#g - затухание  
w1 = 15^0.5  
g1 = 0\*0.5  
  
w2 = 1^0.5  
g2 = 10\*0.5  
  
w3 = 1^0.5  
g3 = 3\*0.5  
  
  
#Правая часть уравнения f(t)  
function f(t)  
 return sin(3 \* t)  
end  
#/Вектор-функция f(t, x)  
#/для решения системы дифференциальных уравнений  
#/x' = y(t, x)  
#/где x - искомый вектор  
function dx1(dv,v,p,t)  
 w,g = p  
 x1,x2 = v  
 dv[1] = x2;  
 dv[2] = -w.\* w.\* x1;  
end  
function dx2(dv,v,p,t)  
 w,g = p  
 x1,x2 = v  
 dv[1] = x2;  
 dv[2] = -w.\* w.\* x1 - 2\*g.\* x2 ;  
end  
function dx3(dv,v,p,t)  
 w,g = p  
 x1,x2 = v  
 dv[1] = x2;  
 dv[2] = -w.\* w.\* x1 - 2\*g.\* x2 - f(t);  
end  
#Точка, в которой заданы  
#начальные условия  
par1 = [w1,g1]  
par2 = [w2,g2]  
par3 = [w3,g3]  
  
t0 = 0.0  
tmax = 55.0  
#Вектор начальных условий  
#x(t0) = x0  
x0 = [0.0,2.0]  
#Интервал на котором будет  
#решаться задача  
t = (t0,tmax)  
#Решаем дифференциальные уравнения  
#с начальным условием x(t0) = x0  
#на интервале t  
#с правой частью, заданной y  
#и записываем решение в матрицу x  
x1 = ODEProblem(dx1,x0,t,par1)  
sol1 = solve(x1)  
#Количество столбцов в матрице  
n1 = size(sol1,2)  
#Переписываем отдельно  
#x в y1, x' в y2  
y11 = Array{Float16}(undef, n1)  
y21 = Array{Float16}(undef, n1)  
for i = 1: n1  
 y11[i] = sol1[1, i]  
 y21[i] = sol1[2, i]  
end  
#Рисуем фазовый портрет: зависимость x(x')  
  
plot(y11,y21)

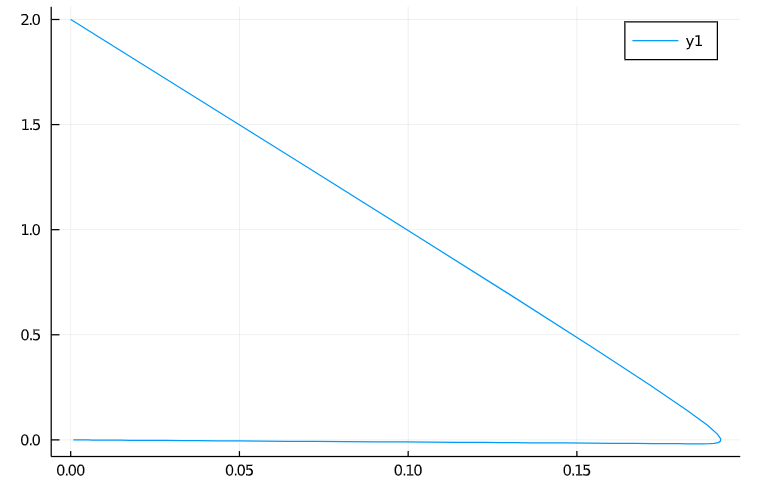
x2 = ODEProblem(dx2,x0,t,par2)  
sol2 = solve(x2)  
#Количество столбцов в матрице  
n2 = size(sol2,2)  
#Переписываем отдельно  
#x в y1, x' в y2  
y12 = Array{Float16}(undef, n2)  
y22 = Array{Float16}(undef, n2)  
for i = 1: n2  
 y12[i] = sol2[1, i]  
 y22[i] = sol2[2, i]  
end  
#Рисуем фазовый портрет: зависимость x(x')  
  
plot(y12,y22)

x3 = ODEProblem(dx3,x0,t,par3)  
sol3 = solve(x3)  
#Количество столбцов в матрице  
n3 = size(sol3,2)  
#Переписываем отдельно  
#x в y1, x' в y2  
y13 = Array{Float16}(undef, n3)  
y23 = Array{Float16}(undef, n3)  
for i = 1: n3  
 y13[i] = sol3[1, i]  
 y23[i] = sol3[2, i]  
end  
#Рисуем фазовый портрет: зависимость x(x')  
  
plot(y13,y23)

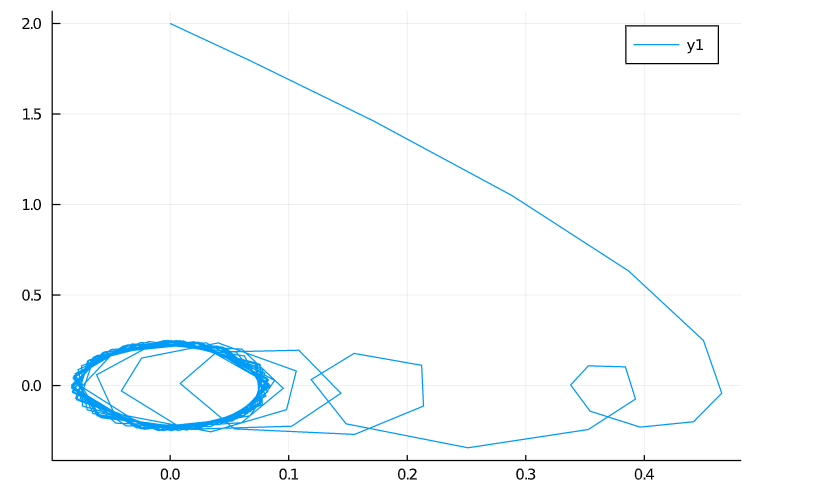
4.2. Получил фазовые портреты гармонического осциллятора (см. рис. @fig:001, @fig:002 и @fig:003):



Колебания без затуханий и без действий внешней силы



Колебания c затуханием и без действий внешней силы



Колебания c затуханием и под действием внешней силы

# Выводы

Построил модель гармонических колебаний с помощью Julia.

# Ответы на вопросы к лабораторной работе

*1. Запишите простейшую модель гармонических колебаний*

*2. Дайте определение осциллятора*

Осциллятор — система, совершающая периодичные колебания.

*3. Запишите модель математического маятника*

*4. Запишите алгоритм перехода от дифференциального уравнения второго порядка к двум дифференциальным уравнениям первого порядка*

От дифференциального уравнение 2-го порядка:

Делаем замену:

Получаем систему уравнений:

*5. Что такое фазовый портрет и фазовая траектория?*

Фазовый портрет — зависимость величин друг от друга. Эти величины также описывают состояние системы.

Фазовая траектория — график нарисованный точками состояний.