

Лабораторная работа №4

Математическое моделирование

Байрамгельдыев Довлетмурат

28 февраля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Байрамгельдыев Довлетмурат
- студент 3 курса группы НФИбд-01-20
- ст. б. 1032207470
- Российский университет дружбы народов
- 1032207470@pfur.ru

Вводная часть

- Решение с помощью программных средств - наиболее распространенный на данный момент подход в решении дифференциальных уравнений. Освоение методов построения математической модели и вариантов ее программной реализации лежит в основе математического моделирования любых систем.
- Julia и OpenModelica - языки программирования, созданные для математических вычислений но использующие разные концепции (императивную и декларативную), являются распространенными инструментами для моделирования.

- Языки научного программирования Julia и OpenModelica
- Фазовые портреты гармонического осциллятора

- Изучить уравнение описывающее поведение гармонического осциллятора
- Построить фазовые портреты рассматриваемых осцилляторов

- Julia
 - DifferentialEquations
 - PyPlot
- OpenModelica

Ход работы

Три случая колебаний гармонического осциллятора

Модель для первого случая:
$$\begin{cases} \dot{x} = y \\ \dot{y} = -\omega_0^2 x \end{cases}$$

Модель для второго случая:
$$\begin{cases} \dot{x} = y \\ \dot{y} = -2\gamma\dot{x} - \omega_0^2 x \end{cases}$$

Модель для третьего случая:
$$\begin{cases} \dot{x} = y \\ \dot{y} = P(t) - 2\gamma\dot{x} - \omega_0^2 x \end{cases}$$

Программа на языке Julia, 1

```
using Plots
using DifferentialEquations

const x0 = 0
const y0 = 1
const omega = 5
const gamma = 0

P(t) = sin(14*t)

T = (0, 30)

u0 = [x0, y0]

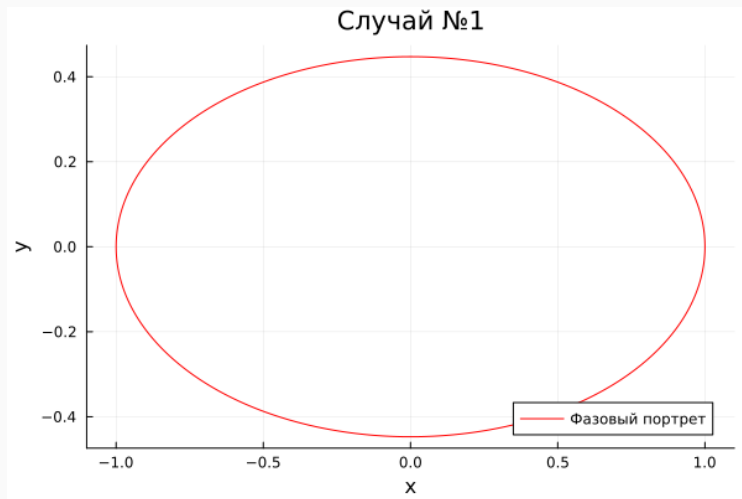
p1 = (omega)
p2 = (omega, gamma)

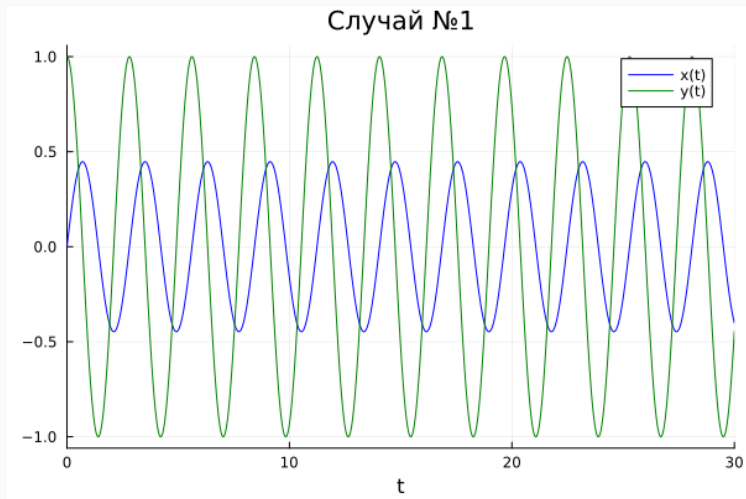
function F1(du, u, p, t)
    omega = p
    du[1] = u[2]
    du[2] = -omega*u[1]
end

prob1 = ODEProblem(F1, u0, T, p1)
sol1 = solve(prob1, dtmax=0.05)

plt = plot(sol1, vars=(2,1), color=:red, label="Фазовый портрет", title="Случай №1", xlabel="x", ylabel="y")
plt2 = plot(sol1, vars=(0,1), color=:blue, label="x(t)", title="Случай №1", xlabel="t")
plot!(plt2, sol1, vars=(0,2), color=:green, label="y(t)")

savefig(plt, "lab4_1_1.png")
savefig(plt2, "lab4_1_2.png")
```





Программа на языке Julia, 2

```
using Plots
using DifferentialEquations

const x0 = 0
const y0 = 1
const omega = 5
const gamma = 2

P(t) = sin(14*t)

T = (0, 30)

u0 = [x0, y0]

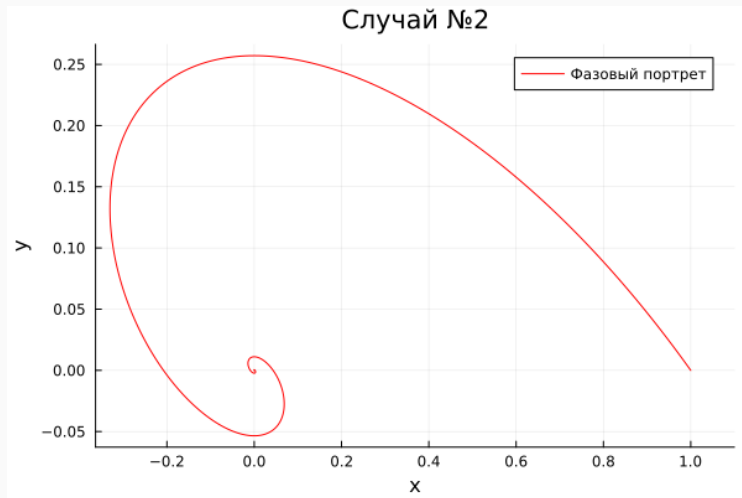
p1 = (omega)
p2 = (omega, gamma)

function F2(du, u, p, t)
    omega, gamma = p
    du[1] = u[2]
    du[2] = -gamma*du[1]-omega*u[1]
end

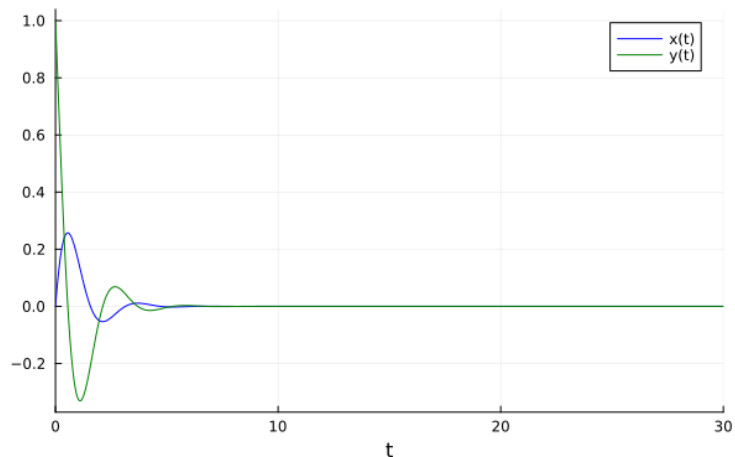
prob2 = ODEProblem(F2, u0, T, p2)
sol2 = solve(prob2, dtmax=0.05)

plt = plot(sol2, vars=(2,1), color=:red, label="Фазовый портрет", title="Случай №2", xlabel="x", ylabel="y")
plt2 = plot(sol2, vars=(0,1), color=:blue, label="x(t)", title="Случай №2", xlabel="t")
plot!(plt2, sol2, vars=(0,2), color=:green, label="y(t)")

savefig(plt, "lab4_2_1.png")
savefig(plt2, "lab4_2_2.png")
```



Случай №2



Программа на языке Julia, 3

```
using Plots
using DifferentialEquations

const x0 = 0
const y0 = 1
const omega = 1
const gamma = 4

P(t) = sin(14*t)

T = (0, 30)

u0 = [x0, y0]

p1 = (omega)
p2 = (omega, gamma)

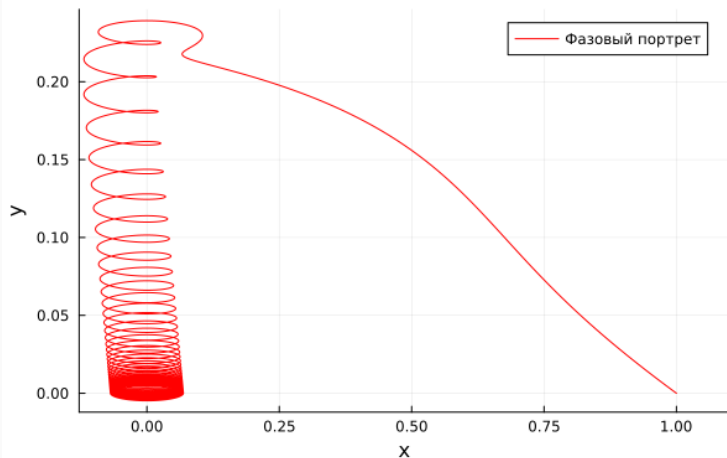
function F3(du, u, p, t)
    omega, gamma = p
    du[1] = u[2]
    du[2] = P(t) - gamma*du[1] - omega*u[1]
end

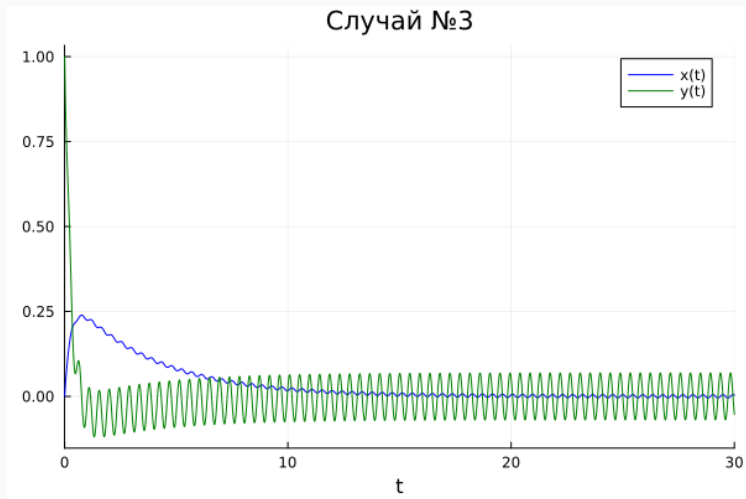
prob3 = ODEProblem(F3, u0, T, p2)
sol3 = solve(prob3, dtmax=0.05)

plt = plot(sol3, vars=(2,1), color=:red, label="Фазовый портрет", title="Случай №3", xlabel="x", ylabel="y")
plt2 = plot(sol3, vars=(0,1), color=:blue, label="x(t)", title="Случай №3", xlabel="t")
plot!(plt2, sol3, vars=(0,2), color=:green, label="y(t)")

savefig(plt, "lab4_3_1.png")
savefig(plt2, "lab4_3_2.png")
```

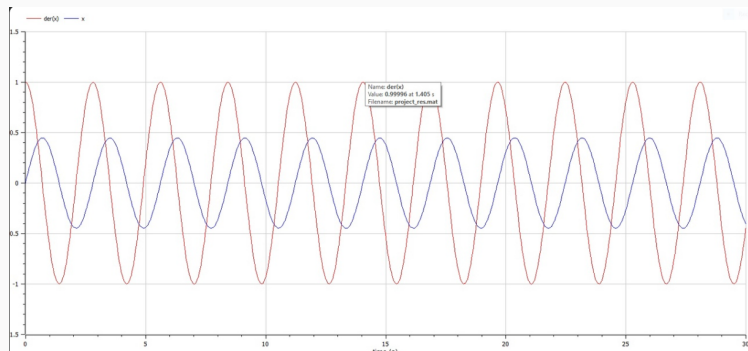
Случай №3



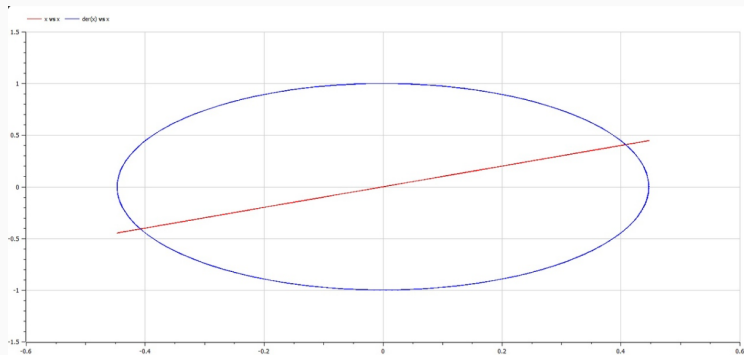


```
model Oscil
parameter Real x0 = 0;
parameter Real y0 = 1;
parameter Real omega = 5;
Real x(start=x0);
Real y(start=y0);
equation
der(x) = y;
der(y) = -omega*x;
end Oscil;
```

Фазовый портрет на языке OpenModelica, 1

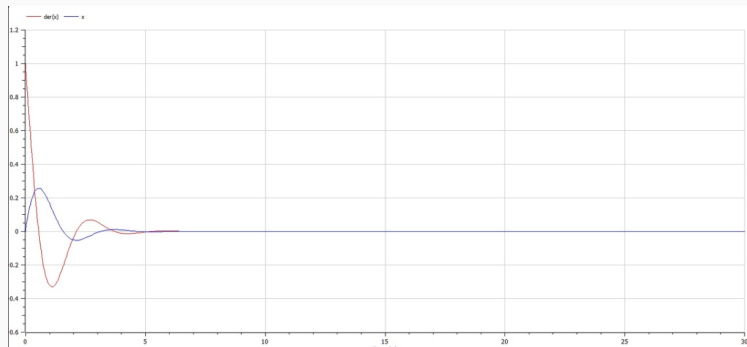


Решение уравнения на языке OpenModelica, 1

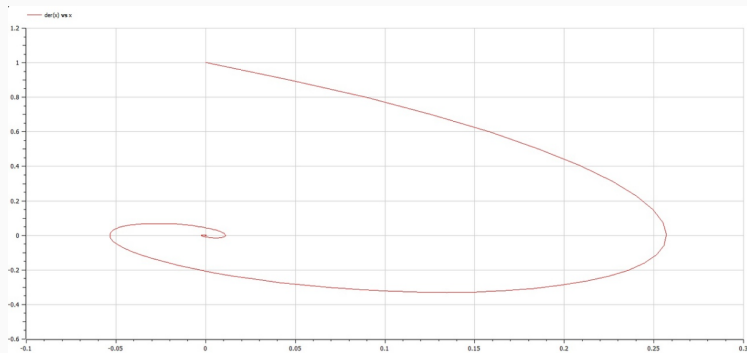


```
model Oscil
parameter Real x0 = 0;
parameter Real y0 = 1;
parameter Real omega = 5;
parameter Real gamma = 2;
Real x(start=x0);
Real y(start=y0);
equation
der(x) = y;
der(y) = -gamma*der(x)-omega*x;
end Oscil;
```

Фазовый портрет на языке OpenModelica, 2

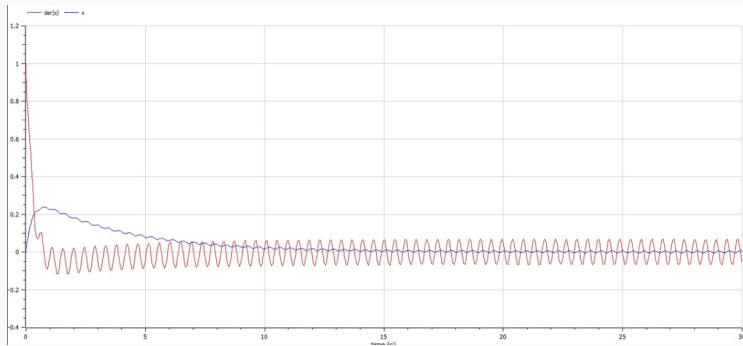


Решение уравнения на языке OpenModelica, 2

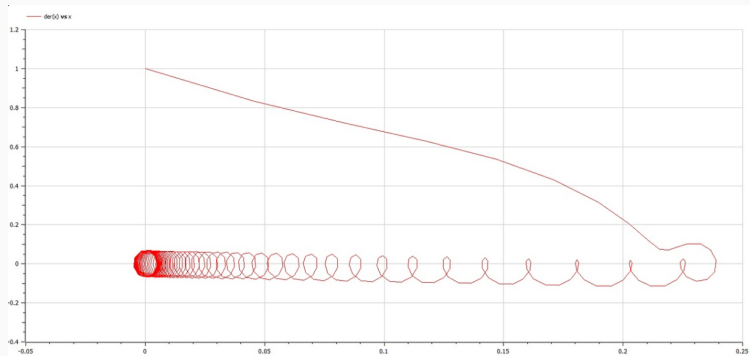


```
model Oscil
parameter Real x0 = 0;
parameter Real y0 = 1;
parameter Real omega = 1;
parameter Real gamma = 4;
Real P;
Real x(start=x0);
Real y(start=y0);
equation
P = sin(14*time);
der(x) = y;
der(y) = P-gamma*der(x)-omega*x;
end Oscil;
```

Фазовый портрет на языке OpenModelica, 3



Решение уравнения на языке OpenModelica, 3



Результаты

- Приобретены навыки работы с Julia и OpenModelica
- Построена модель колебаний гармонического осциллятора
- Построены графики фазового портрета и решения уравнения гармонического осциллятора
- OpenModelica — более подходящий язык для работы с данной моделью-