Лабораторная работа №4

Модель гармонических колебаний

Кузнецов Юрий Владимирович

Содержание

# Цель работы

Рассмотреть уравнение гармонических колебаний. Построить модель гармонических колебаний при помощи OpenModellica и Julia.

# Задачи

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 ) с начальными условиями .

# Среда

* Julia – это открытый свободный высокопроизводительный динамический язык высокого уровня, созданный специально для технических (математических) вычислений. Его синтаксис близок к синтаксису других сред технических вычислений, таких как Matlab и Octave. [@unn-julia]
* OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. [@wiki-om]

# 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. [@rudn-task]

Эта модель называется линейным гармоническим осциллятором. Уравнение свободных колебаний гармонического осциллятора имеет следующий вид:

где – переменная, описывающая состояние системы (смещение грузика, заряд конденсатора и т.д.), – параметр, характеризующий потери энергии (трение в механической системе, сопротивление в контуре), – собственная частота колебаний, – время. (Обозначения , )

При отсутствии потерь в системе вместо вышекуказанного уравнения получаем уравнение консервативного осциллятора энергия колебания которого сохраняется во времени

Для однозначной разрешимости уравнения второго порядка необходимо задать два начальных условия вида

Уравнение консервативного осциллятора энергия колебания которого сохраняется во времени можно представить в виде системы двух уравнений первого порядка:

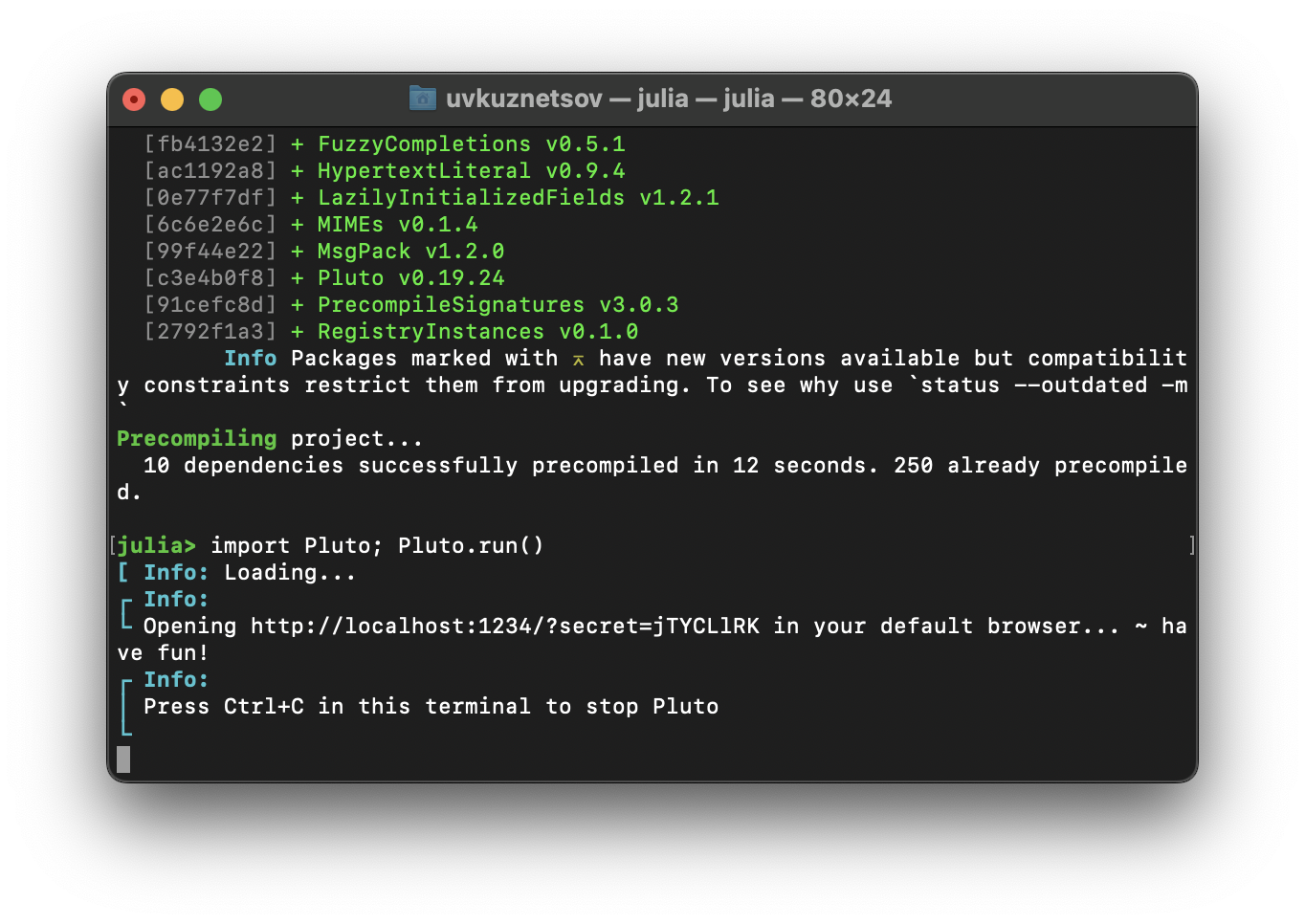
Начальные условия для системы примут вид:

Независимые переменные , определяют пространство, в котором «движется» решение. Это фазовое пространство системы, поскольку оно двумерно будем называть его фазовой плоскостью.

Значение фазовых координат , в любой момент времени полностью определяет состояние системы. Решению уравнения движения как функции времени отвечает гладкая кривая в фазовой плоскости. Она называется фазовой траекторией. Если множество различных решений (соответствующих различным начальным условиям) изобразить на одной фазовой плоскости, возникает общая картина поведения системы. Такую картину, образованную набором фазовых траекторий, называют фазовым портретом.

# Ход работы

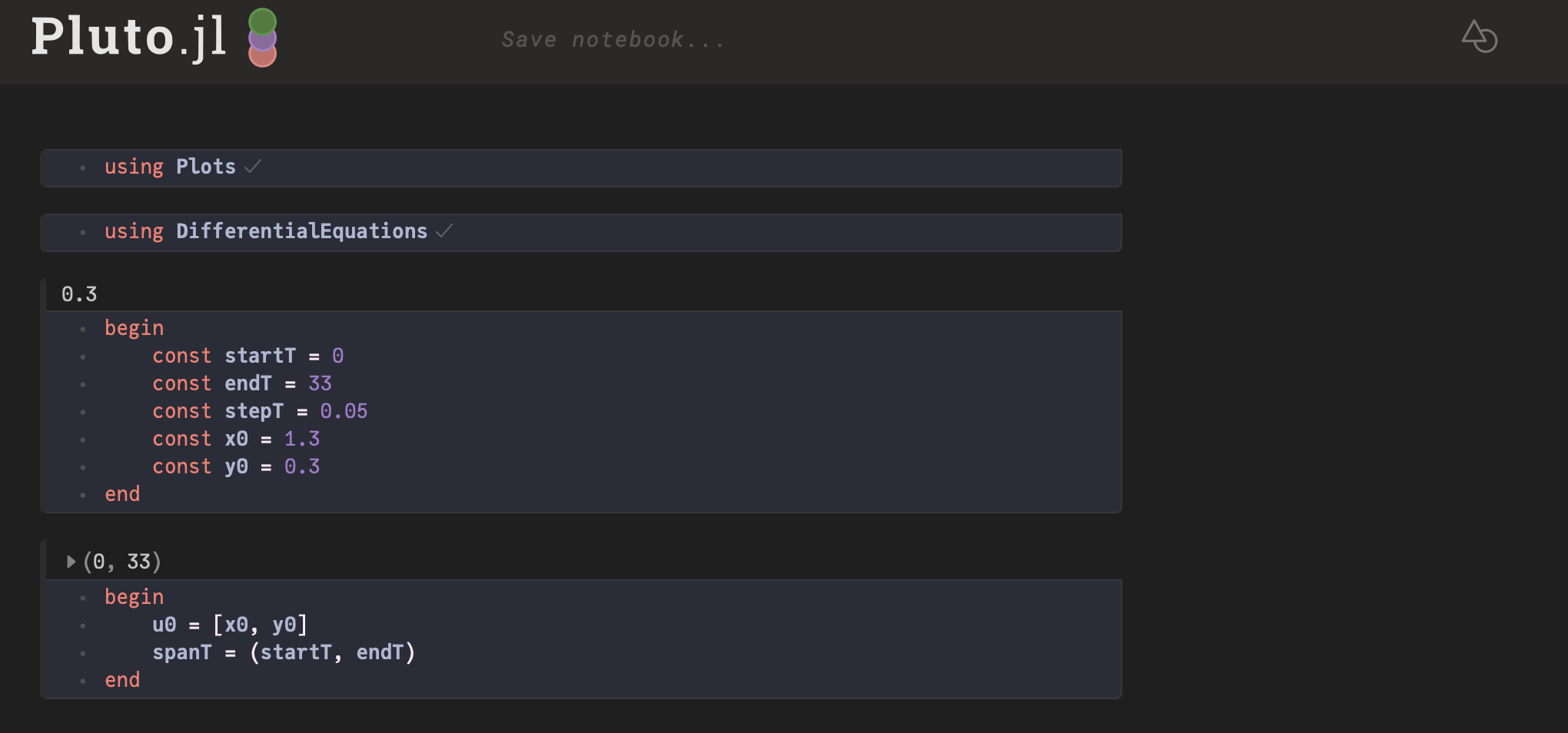
Запустим Pluto для выполнения первых задач.



Julia. Запуск Pluto

Подключаем Plots и DifferentialEquations. Объявляем начальные данные при помощи констант. Устанавливаем начальное условие для системы ДУ и промежуток времени для визуализации.

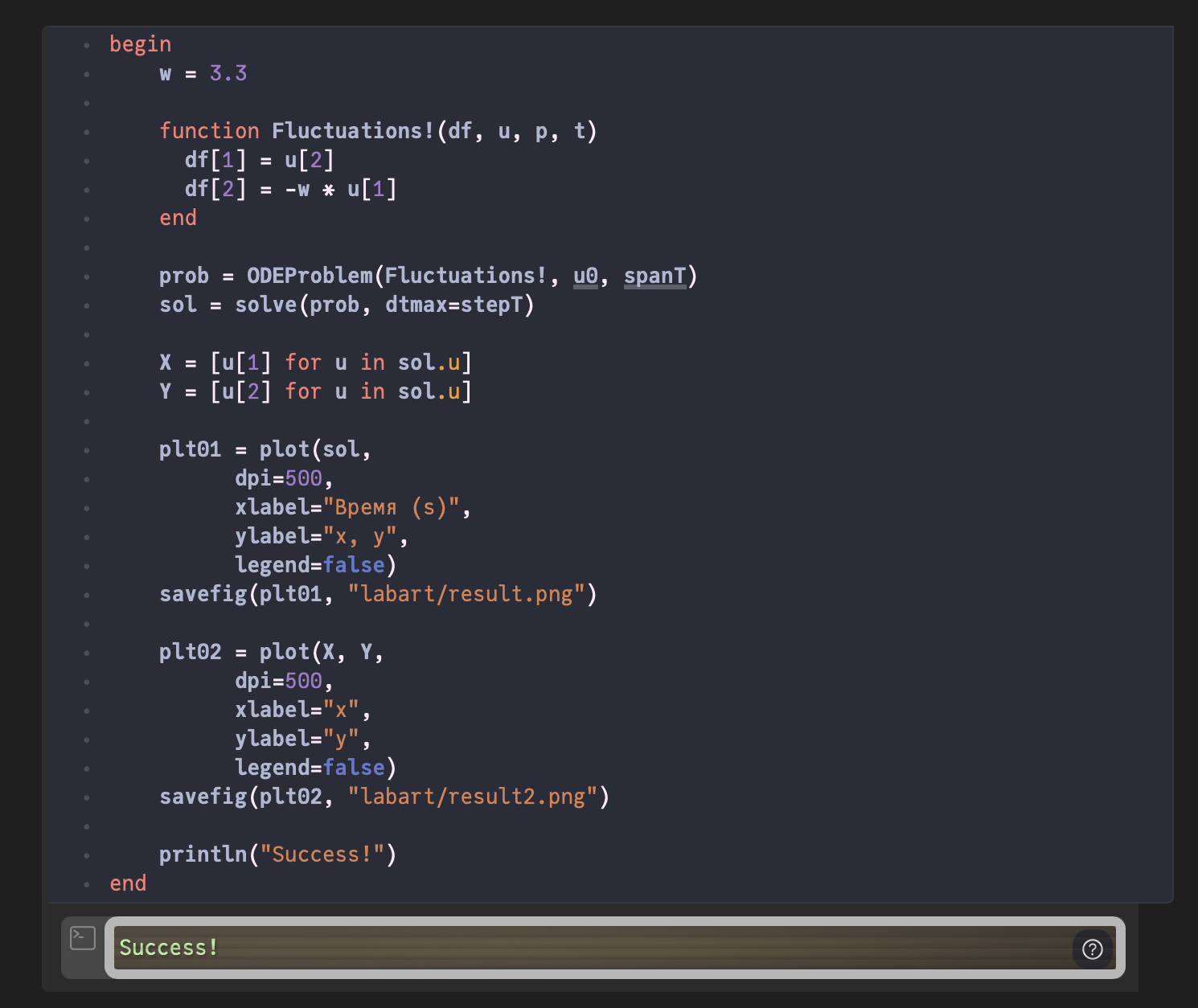
using Plots  
using DifferentialEquations  
  
const startT = 0  
const endT = 33  
const stepT = 0.05  
const x0 = 1.3  
const y0 = 0.3  
  
u0 = [x0, y0]  
spanT = (startT, endT)



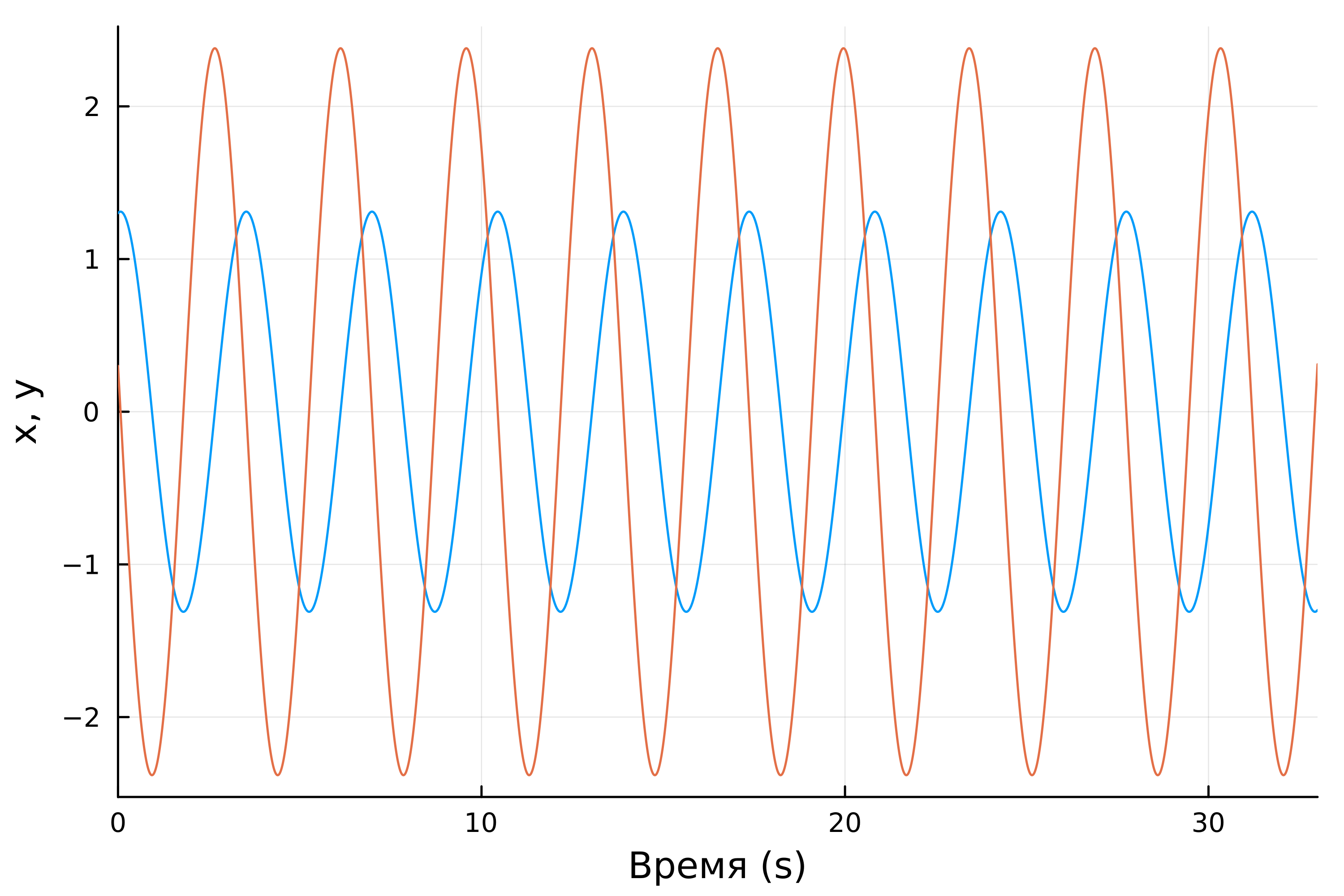
Julia. Начало написания скрипта для моделирование колебания гармонического осциллятора

В следующей ячейке Pluto построим фазовый портрет и решение уравнения гармонического осциллятора. Воспользуемся данным списком, чтобы построить фазовый портрет.

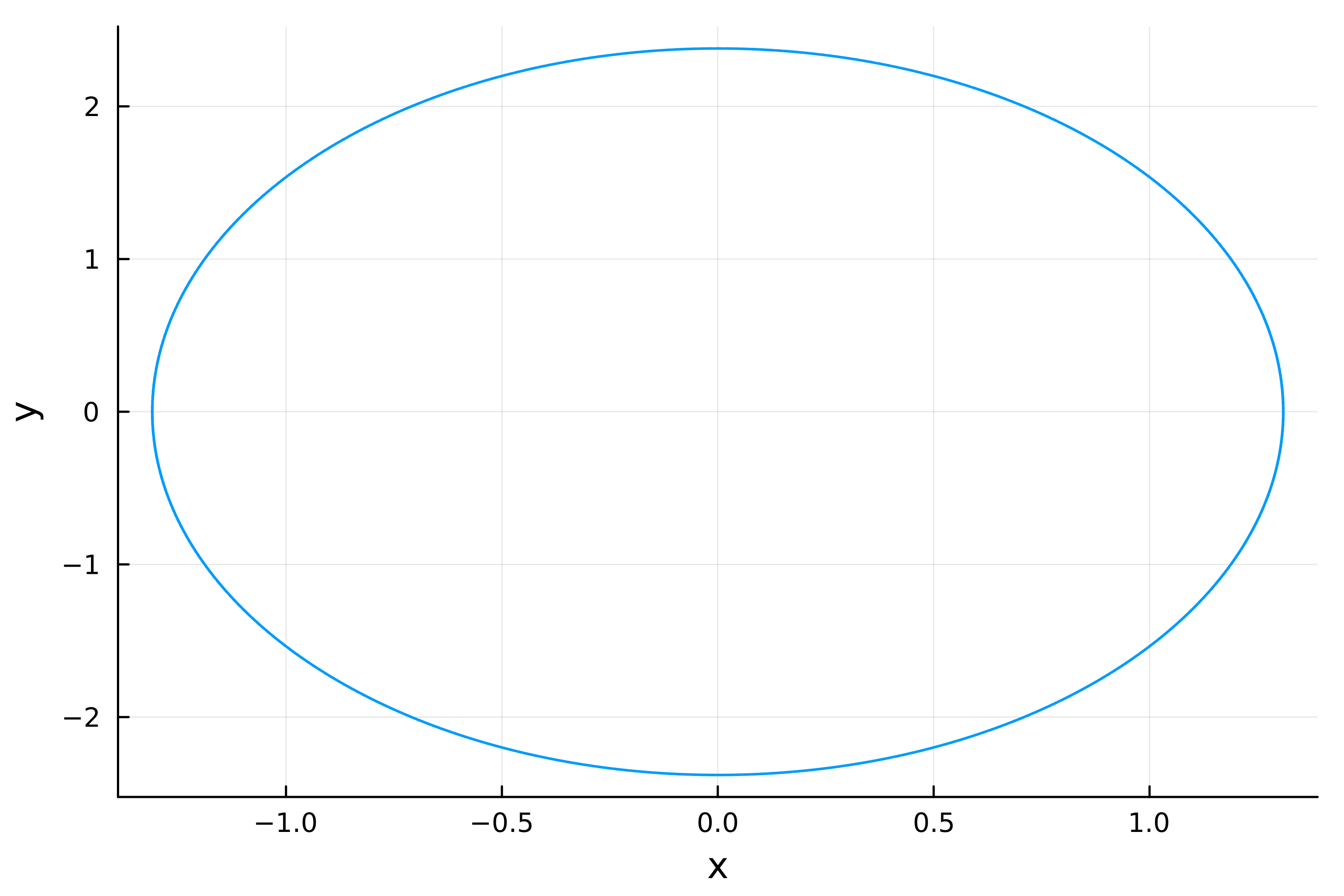
w = 3.3  
  
function Fluctuations!(df, u, p, t)  
 df[1] = u[2]  
 df[2] = -w \* u[1]  
end  
  
prob = ODEProblem(Fluctuations!, u0, spanT)  
sol = solve(prob, dtmax=stepT)  
  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
  
plt01 = plot(sol,  
 dpi=500,  
 xlabel="Время (s)",  
 ylabel="x, y",  
 legend=false)  
savefig(plt01, "labart/result.png")  
  
plt02 = plot(X, Y,  
 dpi=500,  
 xlabel="x",  
 ylabel="y",  
 legend=false)  
savefig(plt02, "labart/result2.png")  
  
println("Success!")



Julia. Скрипт. Колебания гармонического осциллятора без затуханий и без действий внешней силы



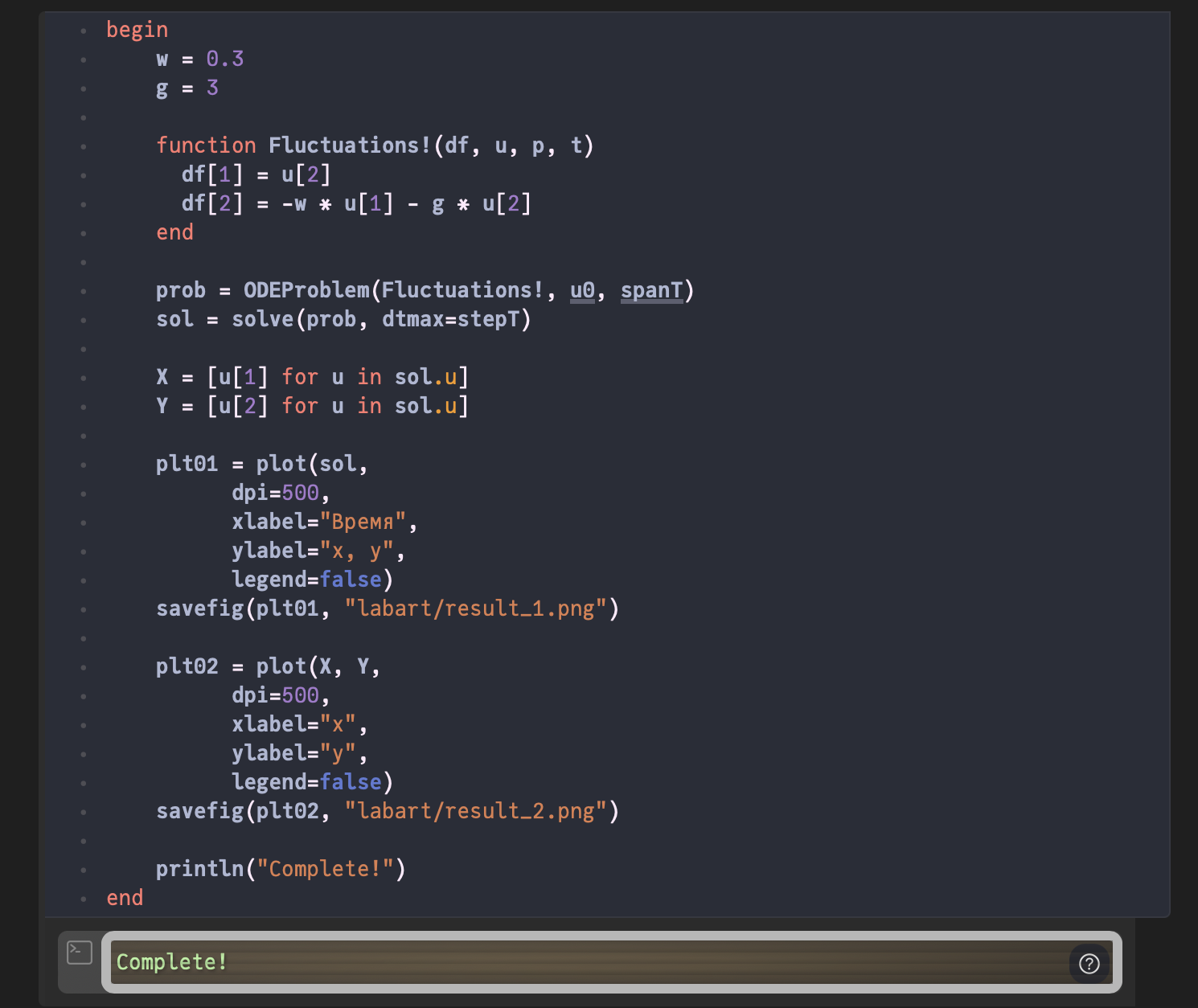
Julia. Модель. Решение уравнения гармонического осциллятора без затуханий и без действий внешней силы



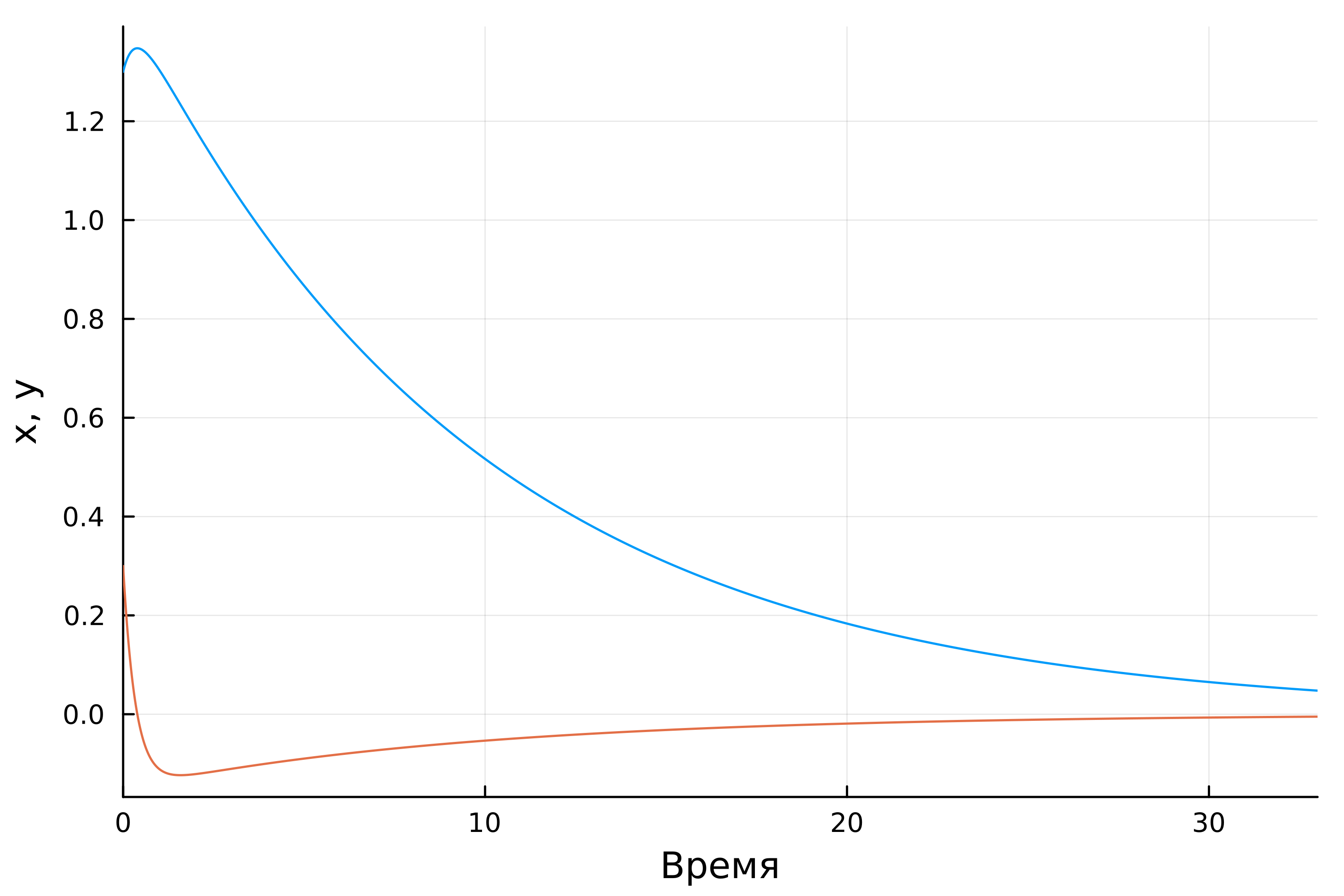
Julia. Модель. Фазовый портрет осциллятора без затуханий и без действий внешней силы

Доработаем данный скрипт, чтобы построить решение уравнения и фазовый портрет гармонического осциллятора c затуханием и без действий внешней силы. Для этого нам неободимо добавить новый параметр - затухание. Также необходимо изменить функцию системы ДУ.

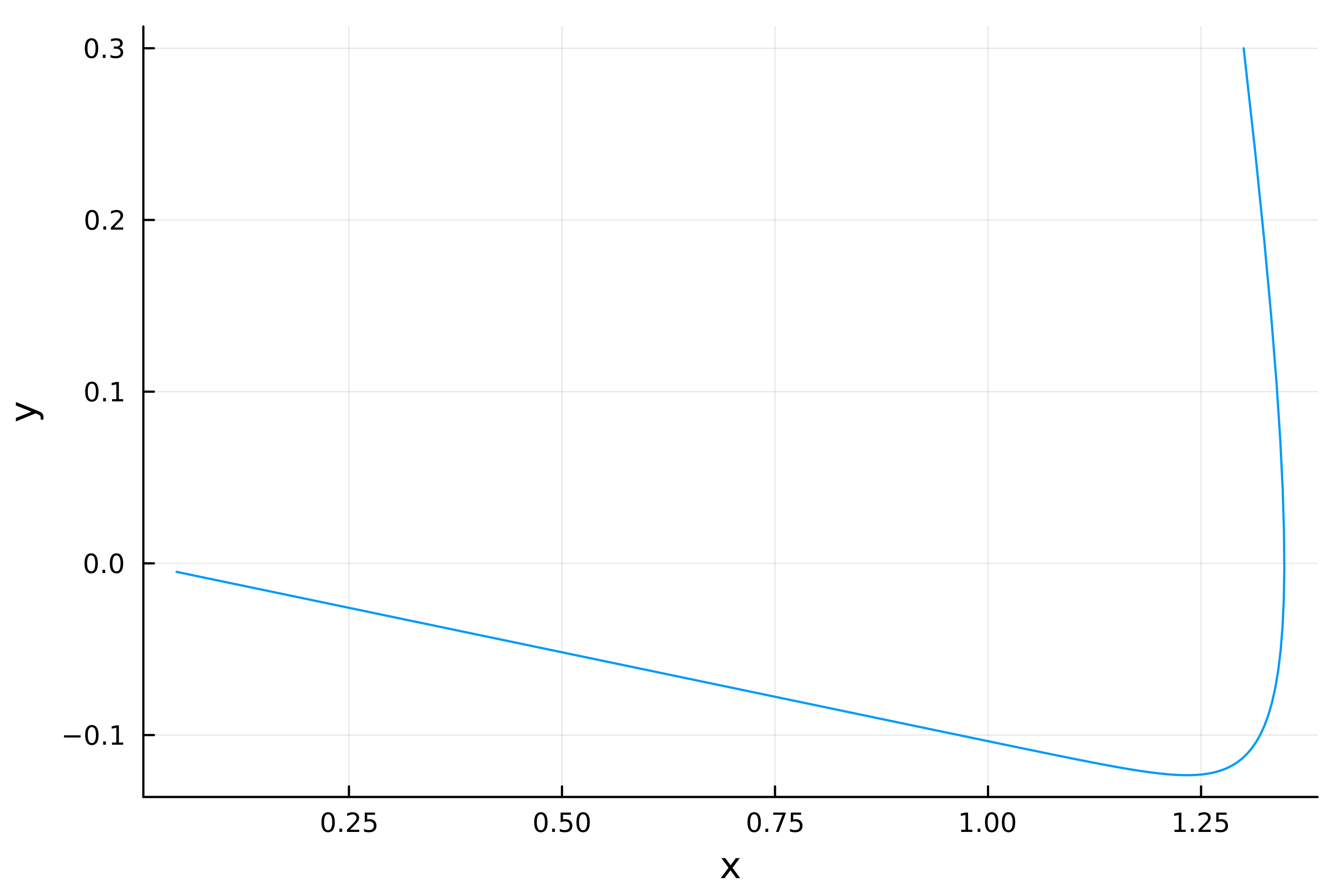
w = 0.3   
g = 3   
  
function Fluctuations!(df, u, p, t)  
 df[1] = u[2]  
 df[2] = -w \* u[1] - g \* u[2]   
end  
  
prob = ODEProblem(Fluctuations!, u0, spanT)  
sol = solve(prob, dtmax=stepT)  
  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
  
plt01 = plot(sol,  
 dpi=500,  
 xlabel="Время",  
 ylabel="x, y",  
 legend=false)  
savefig(plt01, "labart/result\_1.png")  
  
plt02 = plot(X, Y,  
 dpi=500,  
 xlabel="x",  
 ylabel="y",  
 legend=false)  
savefig(plt02, "labart/result\_2.png")  
  
println("Complete!")



Julia. Скрипт. Колебания гармонического осциллятора с затуханием и без действий внешней силы



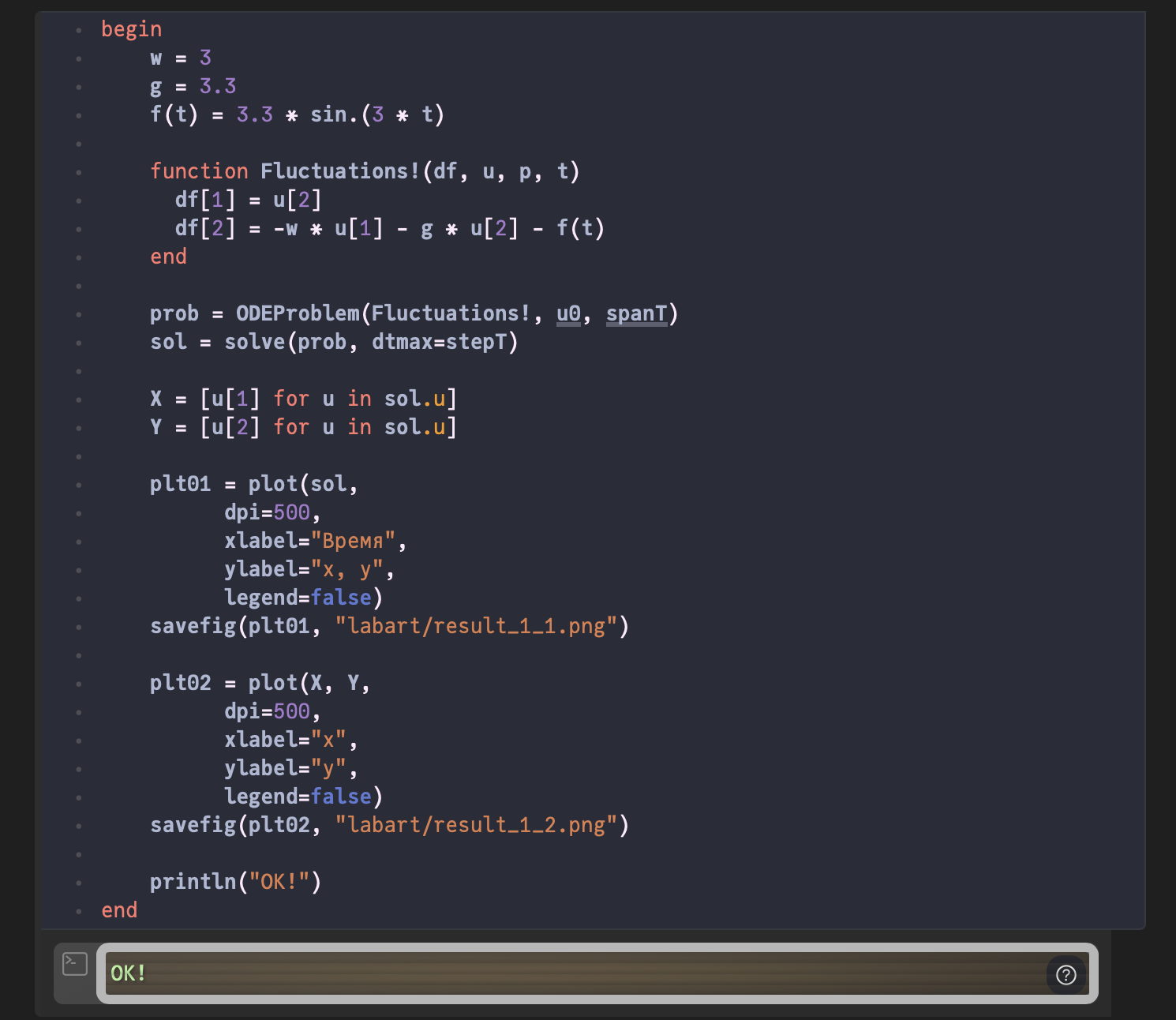
Julia. Модель. Решение уравнения гармонического осциллятора с затуханием и без действий внешней силы



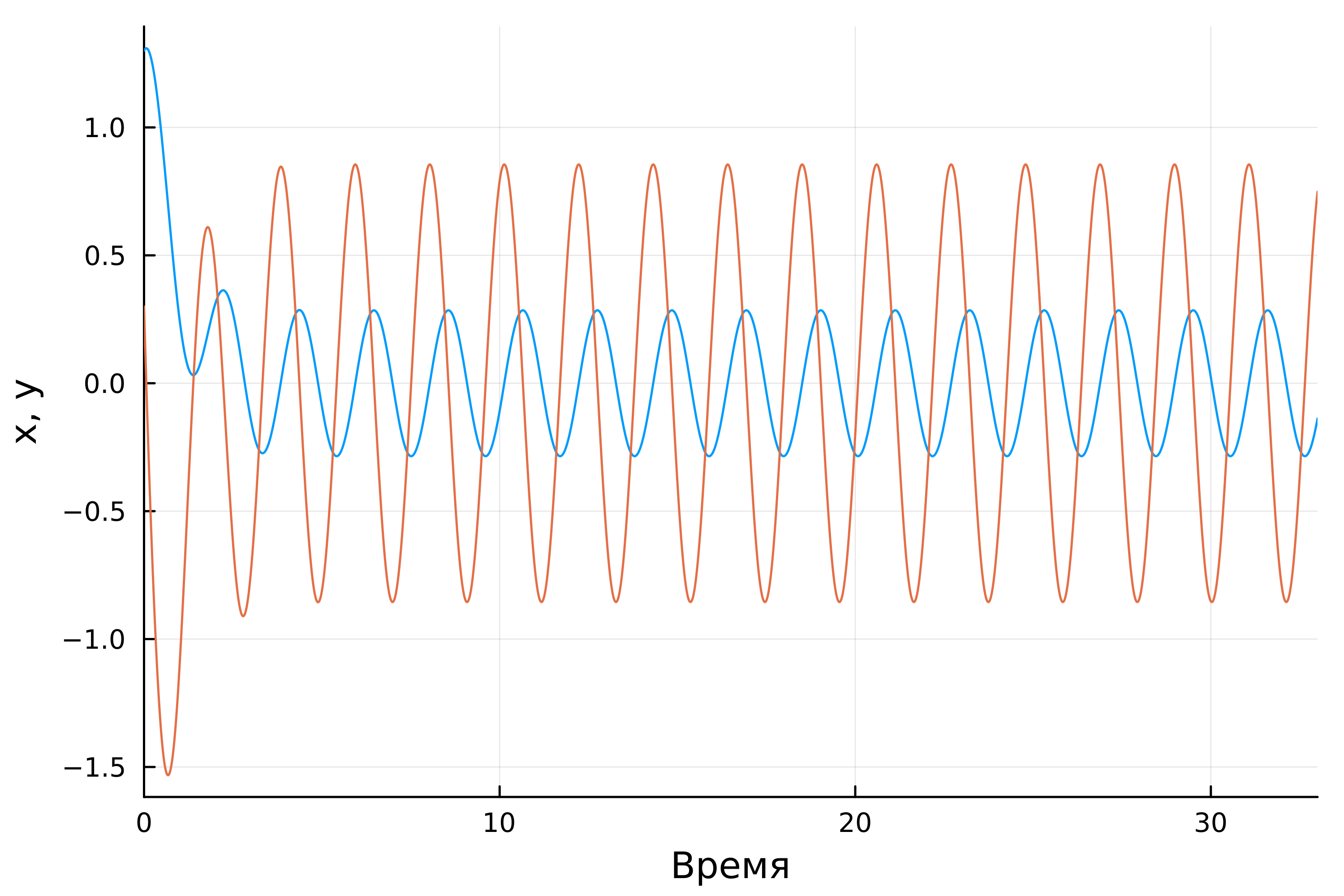
Julia. Модель. Фазовый портрет осциллятора с затуханием и без действий внешней силы

Еще раз доработаем скрипт, чтобы построить решение уравнения и фазовый портрет гармонического осциллятора c затуханием и под действием внешней силы. Для этого нам неободимо добавить новый параметр - функция внешней силы. Также необходимо изменить функцию системы ДУ.

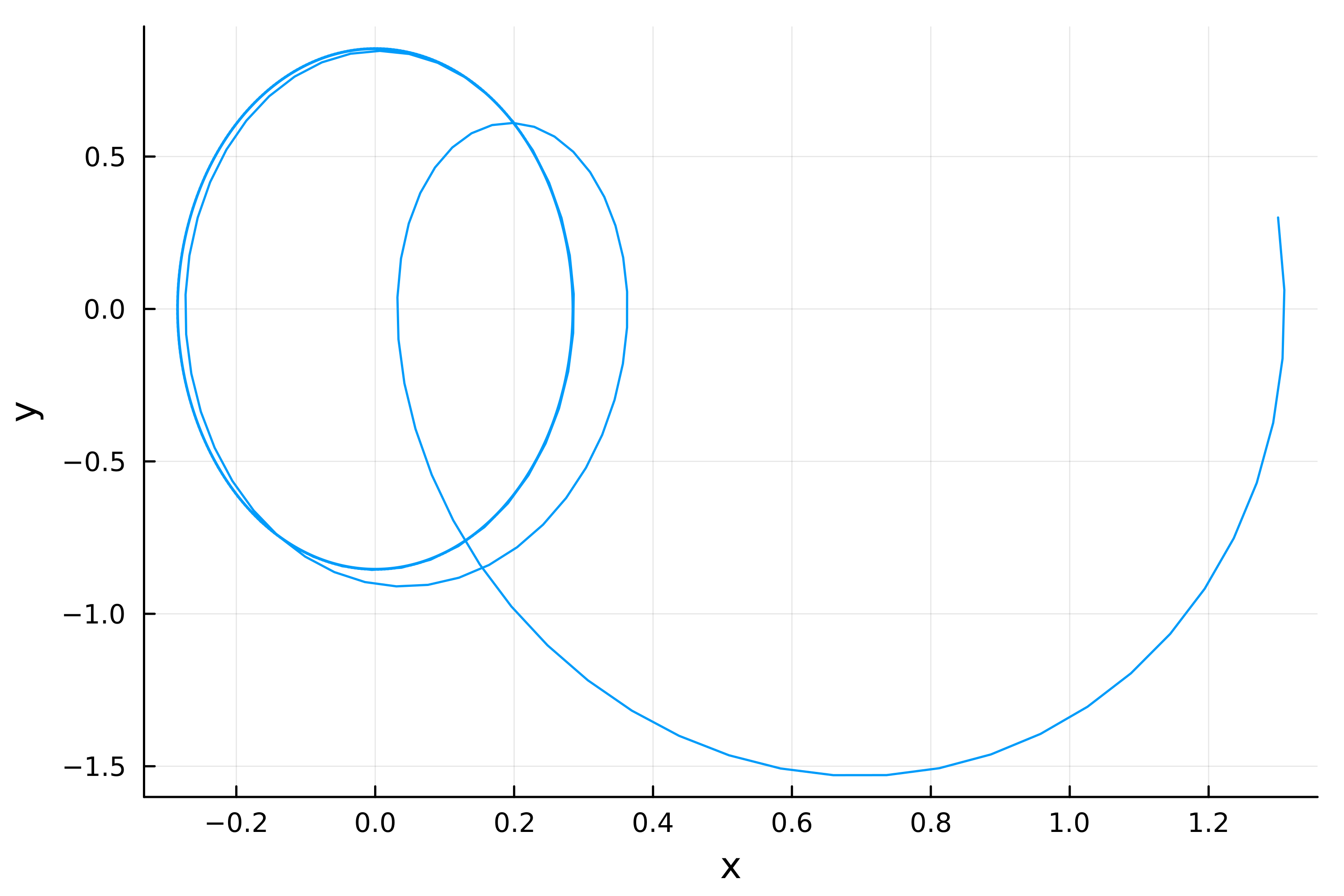
w = 3   
g = 3.3   
f(t) = 3.3 \* sin.(3 \* t)   
  
function Fluctuations!(df, u, p, t)  
 df[1] = u[2]  
 df[2] = -w \* u[1] - g \* u[2] - f(t)   
end  
  
prob = ODEProblem(Fluctuations!, u0, spanT)  
sol = solve(prob, dtmax=stepT)  
  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
  
plt01 = plot(sol,  
 dpi=500,  
 xlabel="Время",  
 ylabel="x, y",  
 legend=false)  
savefig(plt01, "labart/result\_1\_1.png")  
  
plt02 = plot(X, Y,  
 dpi=500,  
 xlabel="x",  
 ylabel="y",  
 legend=false)  
savefig(plt02, "labart/result\_1\_2.png")  
  
println("OK!")



Julia. Скрипт. Колебания гармонического осциллятора с затуханием и под действием внешней силы



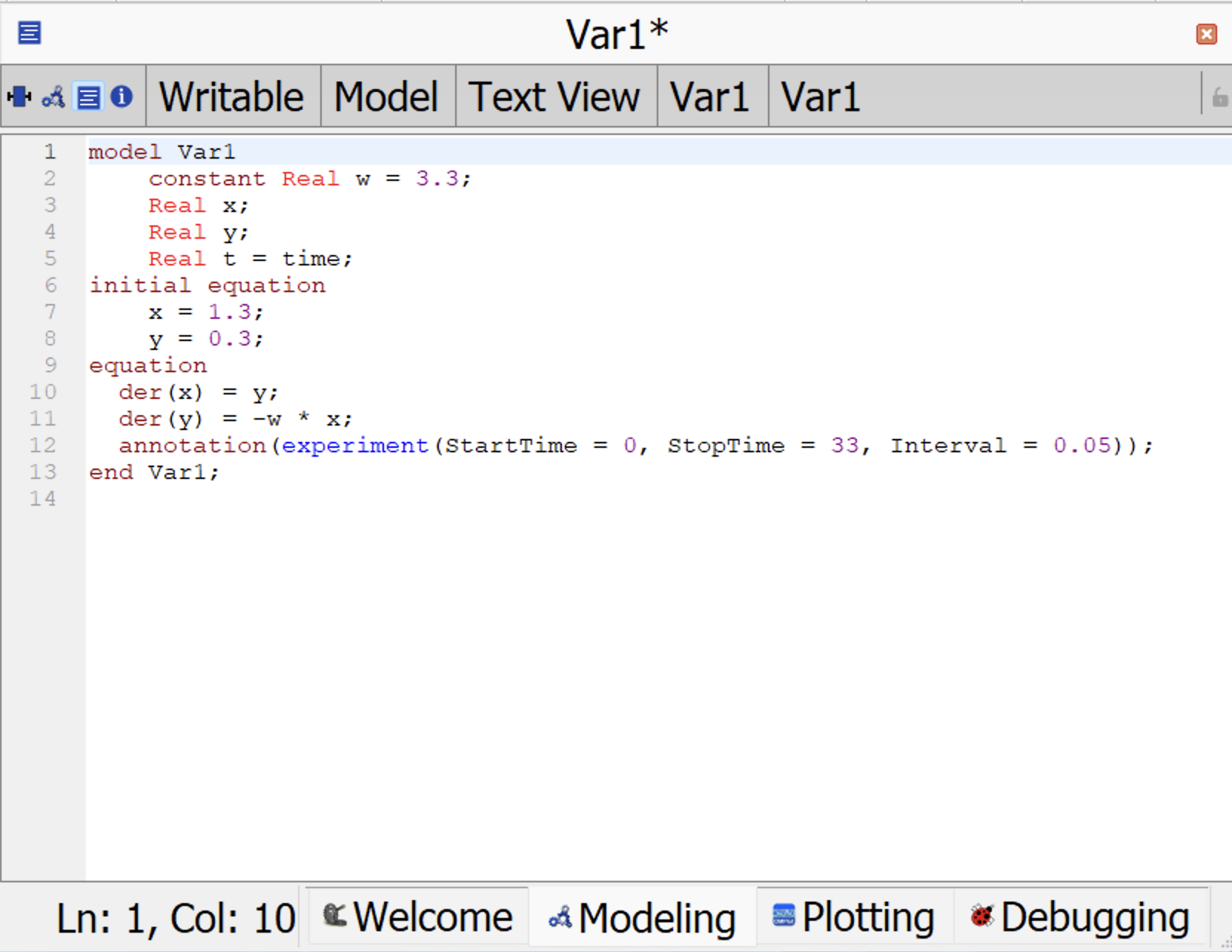
Julia. Модель. Решение уравнения гармонического осциллятора с затуханием и под действием внешней силы



Julia. Модель. Фазовый портрет осциллятора с затуханием и под действием внешней силы

Построим модель колебания гармонического осциллятора без затуханий и без действий внешней силы на Modelica.

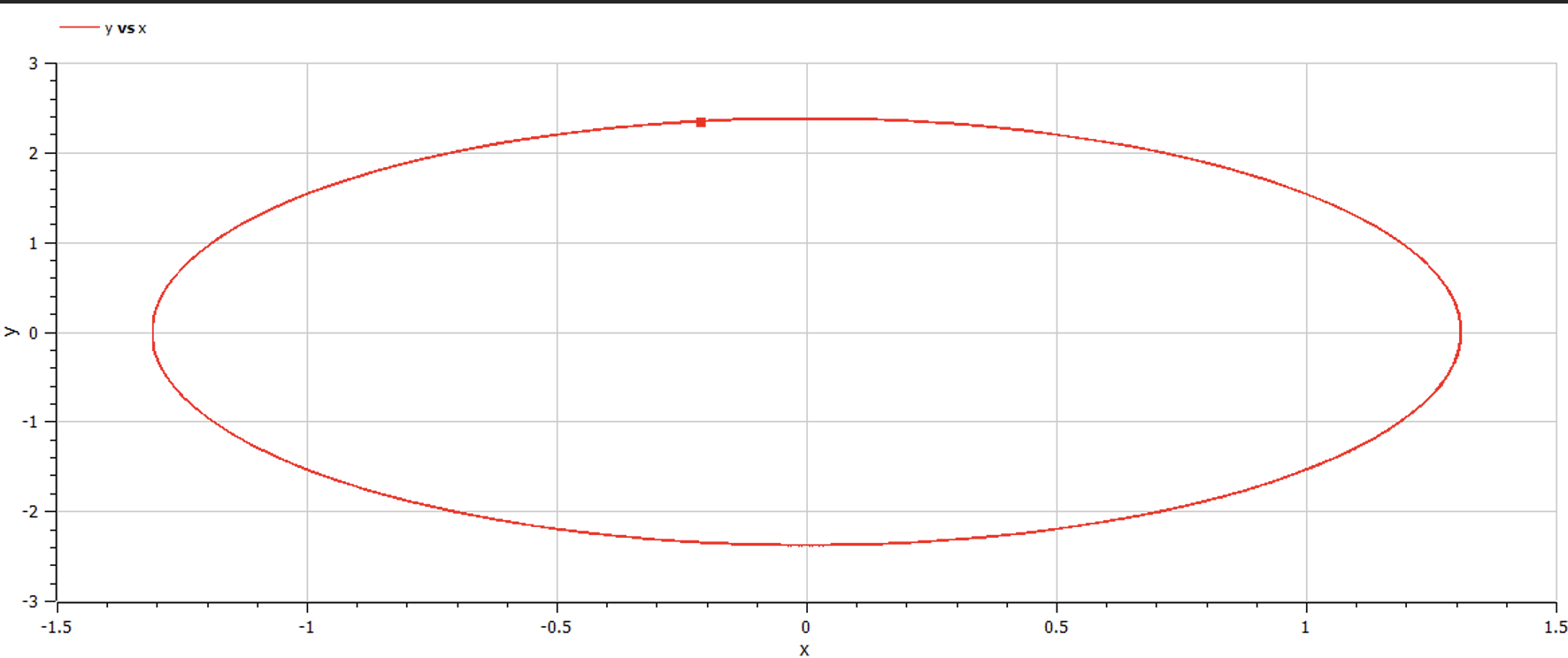
model Var1  
 constant Real w = 3.3;  
 Real x;  
 Real y;  
 Real t = time;  
initial equation  
 x = 1.3;  
 y = 0.3;  
equation  
 der(x) = y;  
 der(y) = -w \* x;  
 annotation(experiment(StartTime = 0, StopTime = 33, Interval = 0.05));  
end Var1;



Modelica. Скрипт. Колебания гармонического осциллятора без затуханий и без действий внешней силы



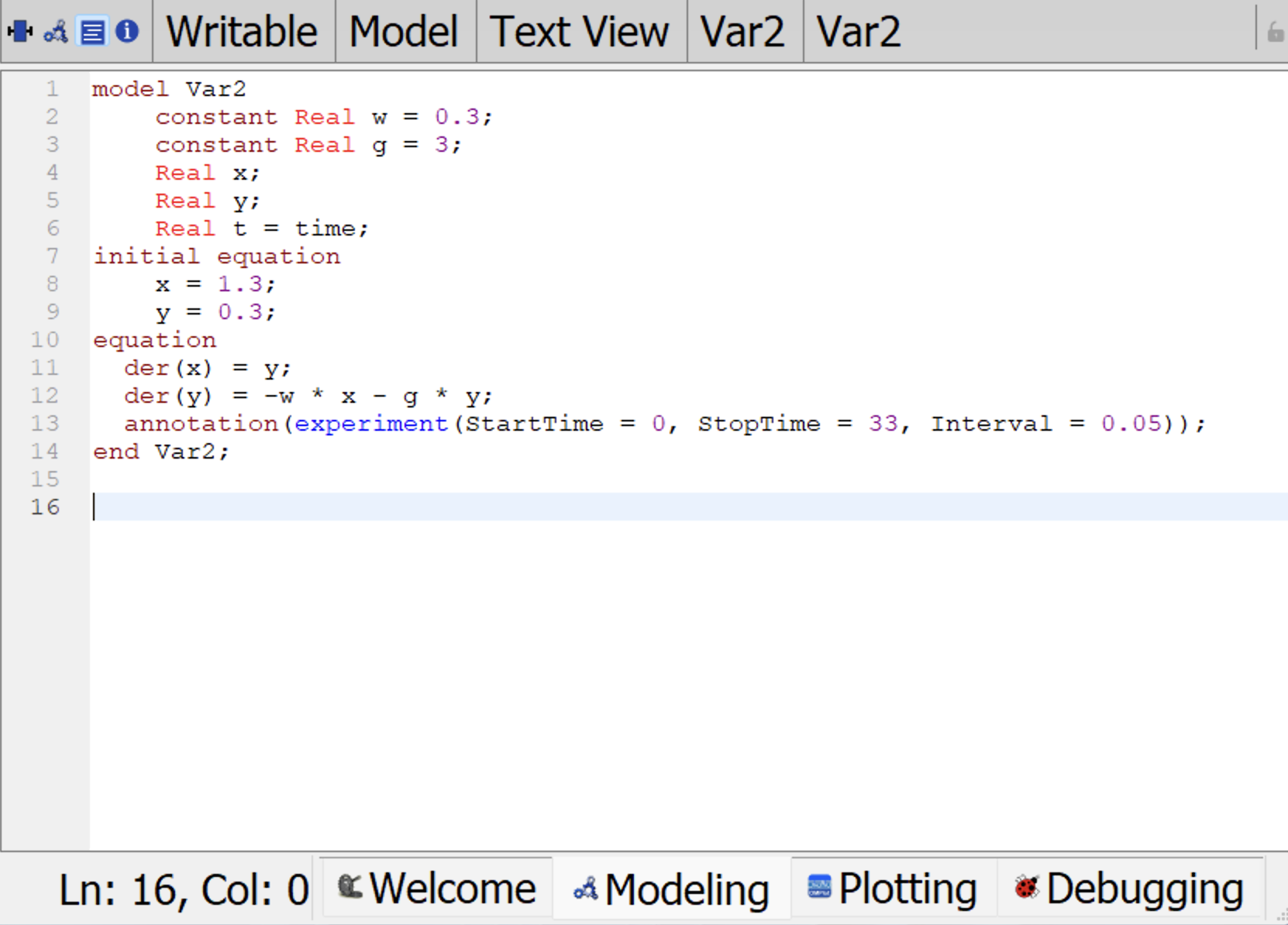
Modelica. Модель. Решение уравнения гармонического осциллятора без затуханий и без действий внешней силы



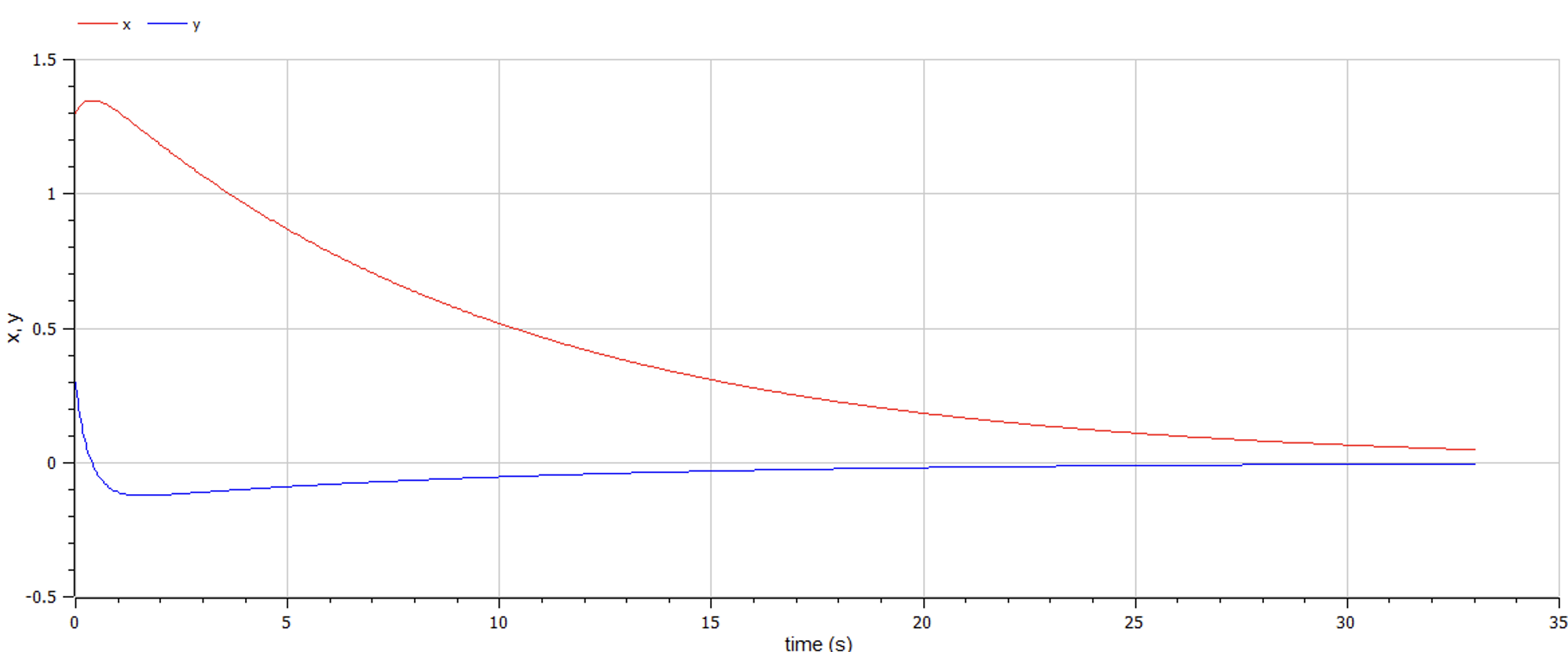
Modelica. Модель. Фазовый портрет осциллятора без затуханий и без действий внешней силы

Построим модель колебания гармонического осциллятора с затуханием и без действий внешней силы на Modelica.

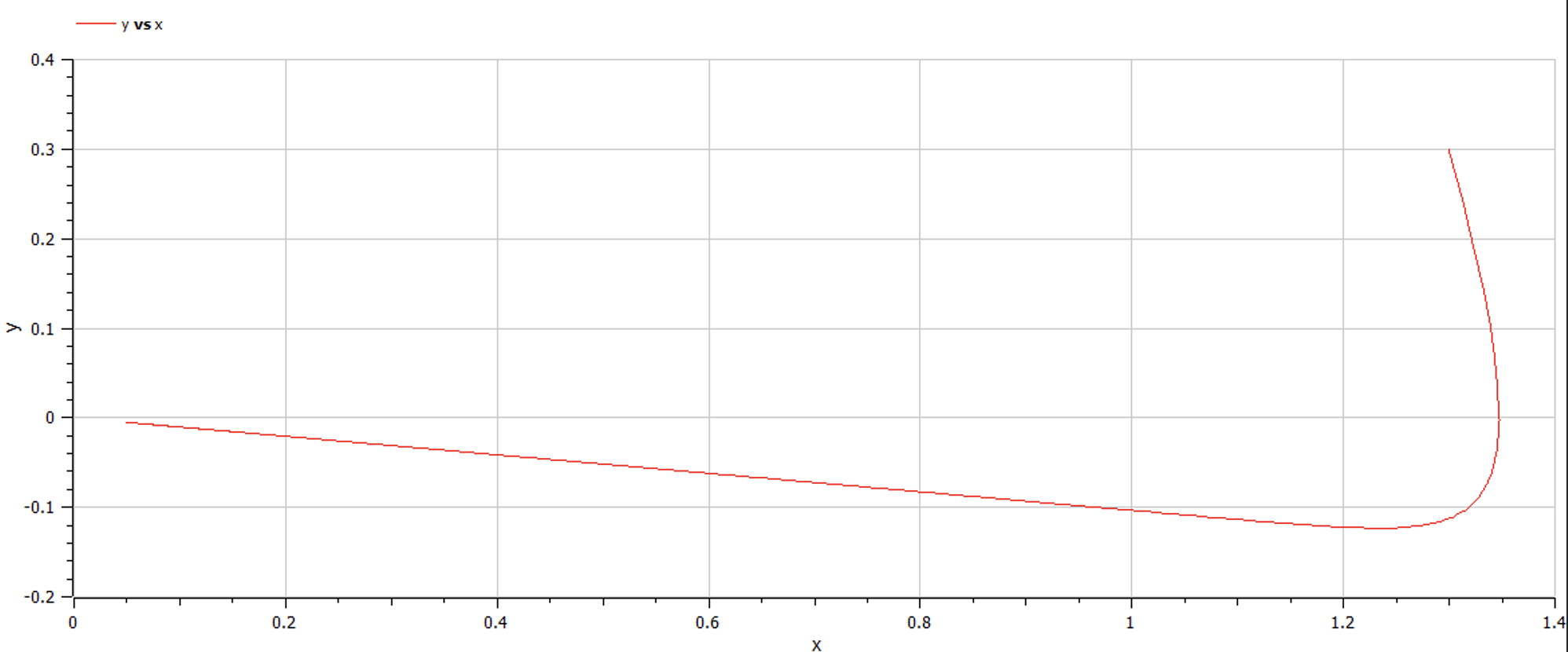
model Var2  
 constant Real w = 0.3;  
 constant Real g = 3;  
 Real x;  
 Real y;  
 Real t = time;  
initial equation  
 x = 1.3;  
 y = 0.3;  
equation  
 der(x) = y;  
 der(y) = -w \* x - g \* y;  
 annotation(experiment(StartTime = 0, StopTime = 33, Interval = 0.05));  
end Var2;



Modelica. Скрипт. Колебания гармонического осциллятора с затуханием и без действий внешней силы



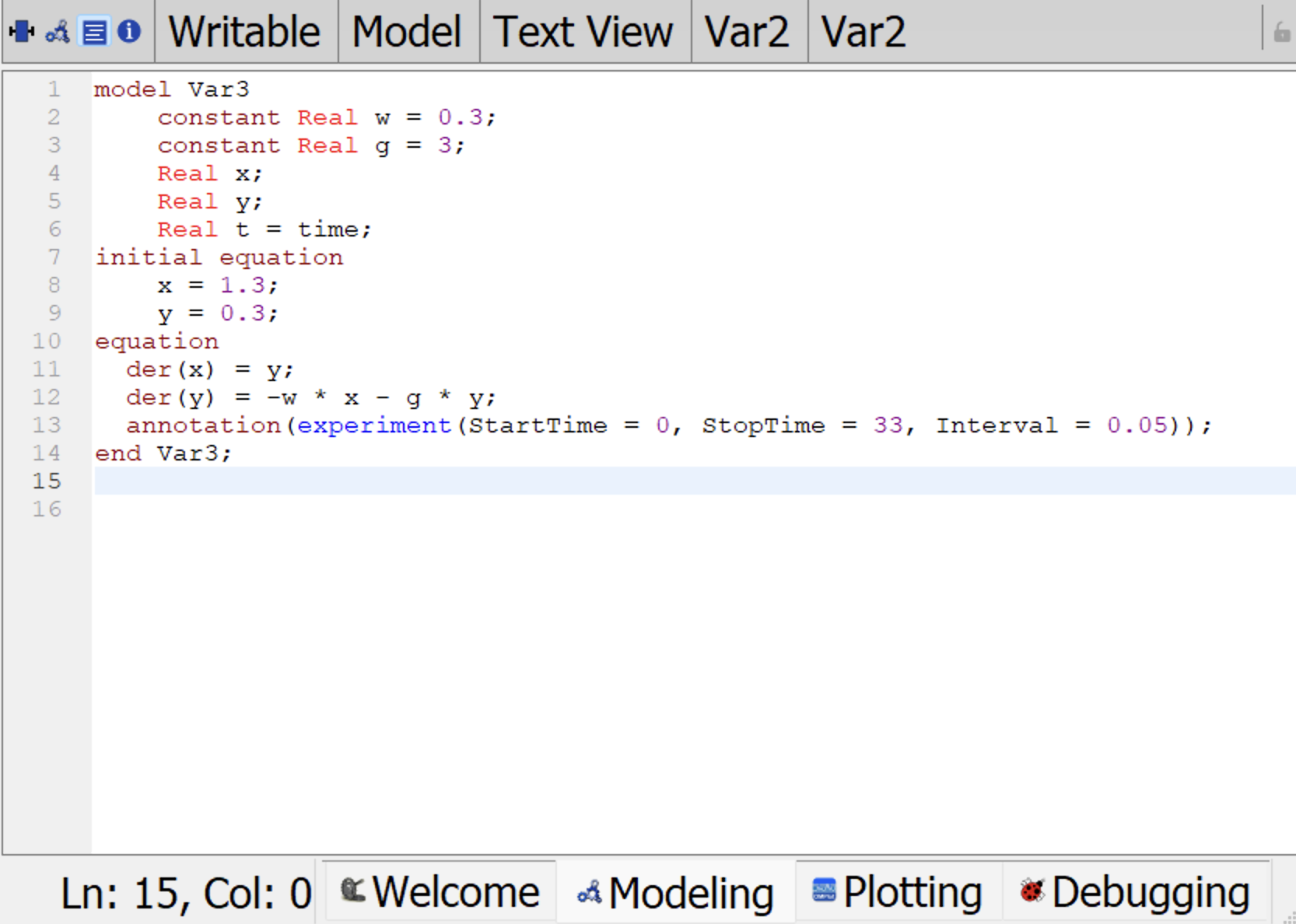
Modelica. Модель. Решение уравнения гармонического осциллятора с затуханием и без действий внешней силы



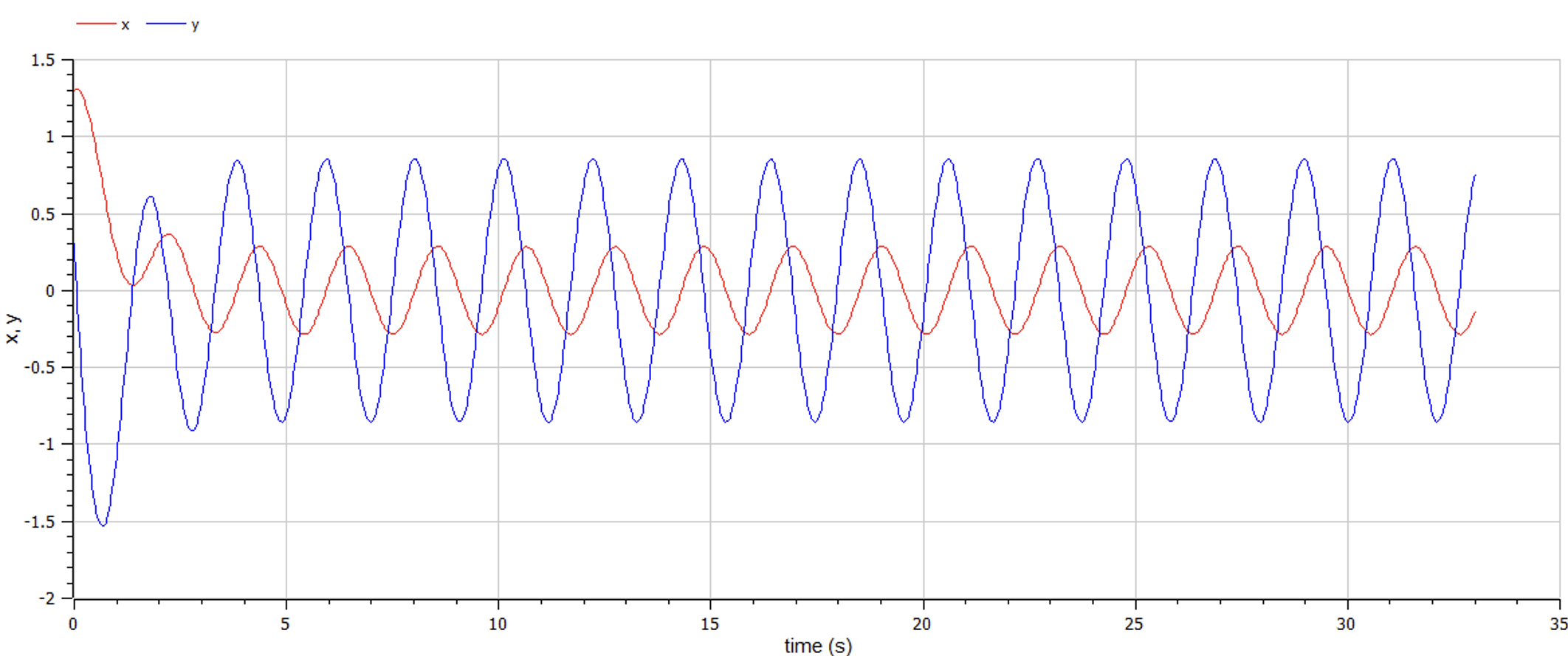
Modelica. Модель. Фазовый портрет осциллятора с затуханием и без действий внешней силы

Построим модель колебания гармонического осциллятора с затуханием и под действием внешней силы на Modelica.

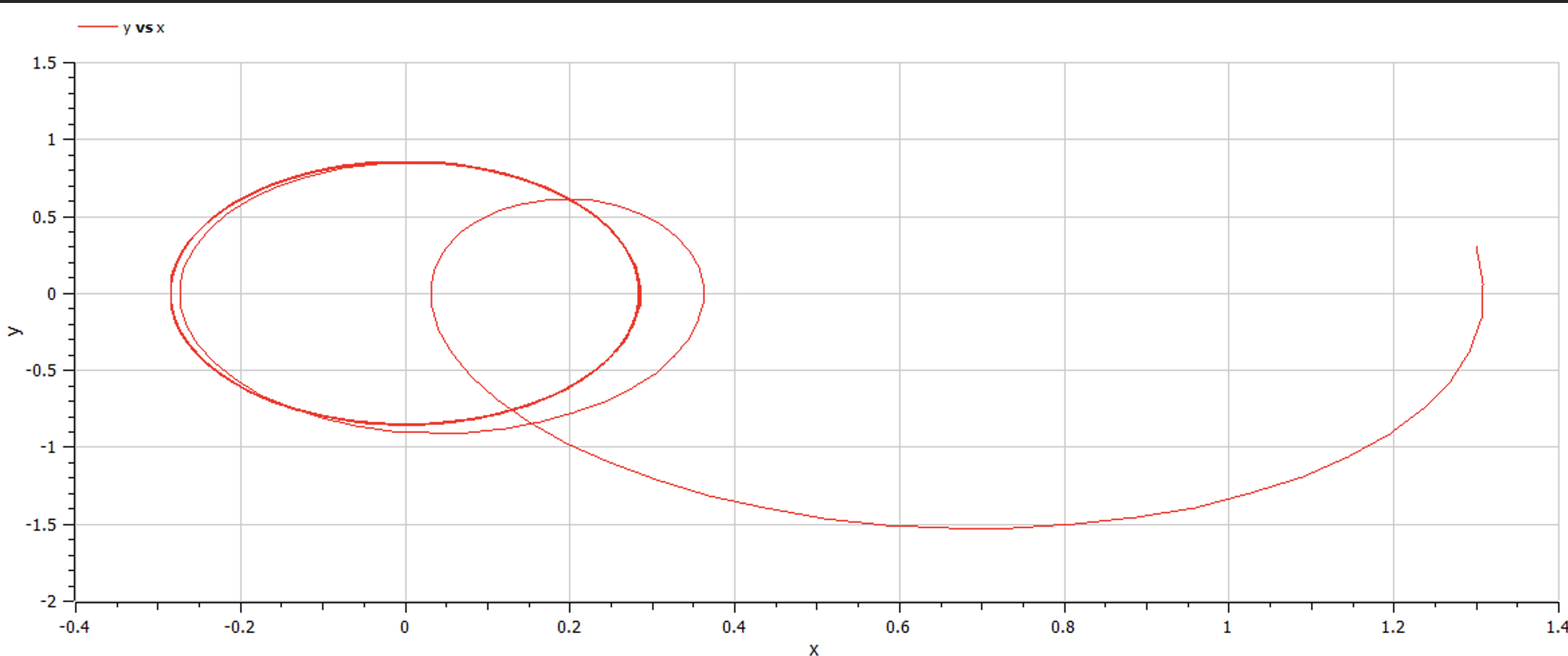
model Var3  
 constant Real w = 0.3;  
 constant Real g = 3;  
 Real x;  
 Real y;  
 Real t = time;  
initial equation  
 x = 1.3;  
 y = 0.3;  
equation  
 der(x) = y;  
 der(y) = -w \* x - g \* y;  
 annotation(experiment(StartTime = 0, StopTime = 33, Interval = 0.05));  
end Var3;



Modelica. Скрипт. Колебания гармонического осциллятора с затуханием и под действием внешней силы



Modelica. Модель. Решение уравнения гармонического осциллятора с затуханием и под действием внешней силы



Modelica. Модель. Фазовый портрет осциллятора с затуханием и без под действием внешней силы

# Вывод

Повысили навыки в написании программ на Julia. Улучшили понимание моделирования на OpenModelica. Познакомились с Pluto. Повысили навыки в решении ДУ.

# Ресурсы

* Julia. URL: http://www.unn.ru/books/met\_files/JULIA\_tutorial.pdf.
* OpenModelica. URL: https://ru.wikipedia.org/wiki/OpenModelica.
* Модель гармонических колебаний. RUDN. URL: https://esystem.rudn.ru/mod/resource/view.php?id=967241.
* Pluto. URL: https://plutojl.org/.
* Plots in Julia. URL: https://docs.juliaplots.org/latest/tutorial/.
* Differential Equations in Julia. URL: https://docs.sciml.ai/DiffEqDocs/stable/getting\_started/.