Лабораторная работа №2

Задача о погоне

Кузнецов Юрий Владимирович

Содержание

# Цель работы

Решить задачу о погоне. Смоделировать кривую погони при помощи Julia и OpenModelica.

# Задачи

1. Провести аналогичные рассуждения и вывод дифференциальных уравнений, если скорость катера больше скорости лодки в n раз.
2. Построить траекторию движения катера и лодки для двух случаев.
3. Определить по графику точку пересечения катера и лодки

# Термины

* Задача о погоне — классических задача из области дифференциальных уравнений.
* Кривая погони — кривая, представляющая собой решение задачи о «погоне».
* 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.
* OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica.

# Теоретическая справка

Для построения кривой погони, нам необходимо знать начальные условия и уравнения кривой. Ориентируясь на рассуждения из пособия к лабораторной работы можно вывести общие формулы. Благодаря общим формулам можно будет написать программу, которая будет строить разные кривые погони в зависимости от исходных данных (расстояния и разницы в скорости).

Пусть

* n - разница в скорости, то есть скорость катера в n раз больше лодки;
* a - расстояние между катером и лодкой в момент рассеивания тумана.

Тогда общие начальные условия для первого случая выглядят следующим образом:

Для второго случая:

Уравнение кривой в общем случае выглядит следующим образом:

# Выполнение лабораторной работы.

Установим пакет в Julia необходимый для построения графика.

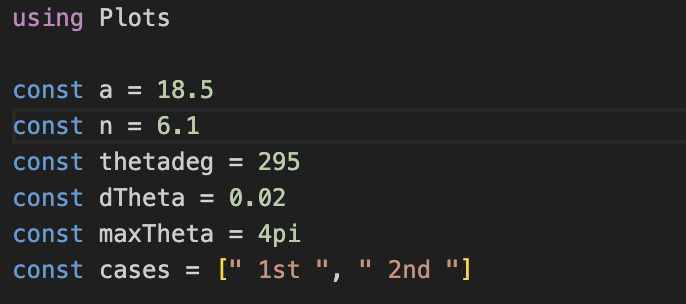
import Pkg  
Pkg.add("Plots")

Изображение выглядит как текст

Автоматически созданное описание

Установка пакета Plots

Для подключения графики импортируем пакет Plots при помощи команды using. Константы a и n - входные данные, thetadeg - направление лодки, dTheta - шаг для равномерного разбиения периода, maxTheta - длина периода построения, cases - содержит два кейса, по которым мы будем итерироваться.

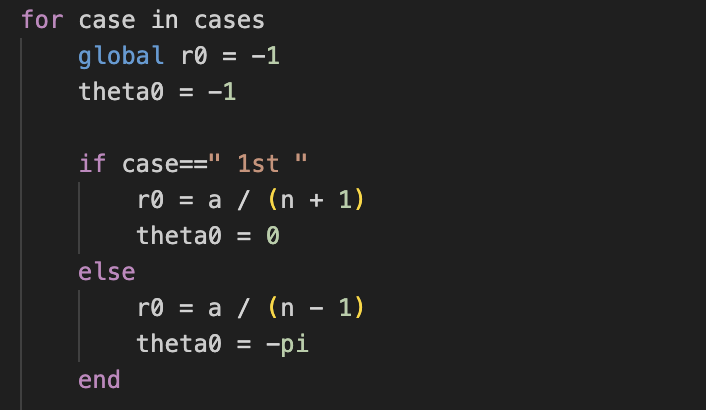


Следующие итерации будут проходить в цикле следующим образом:

for item in list  
   
end

Также будем использовать условные операторы:

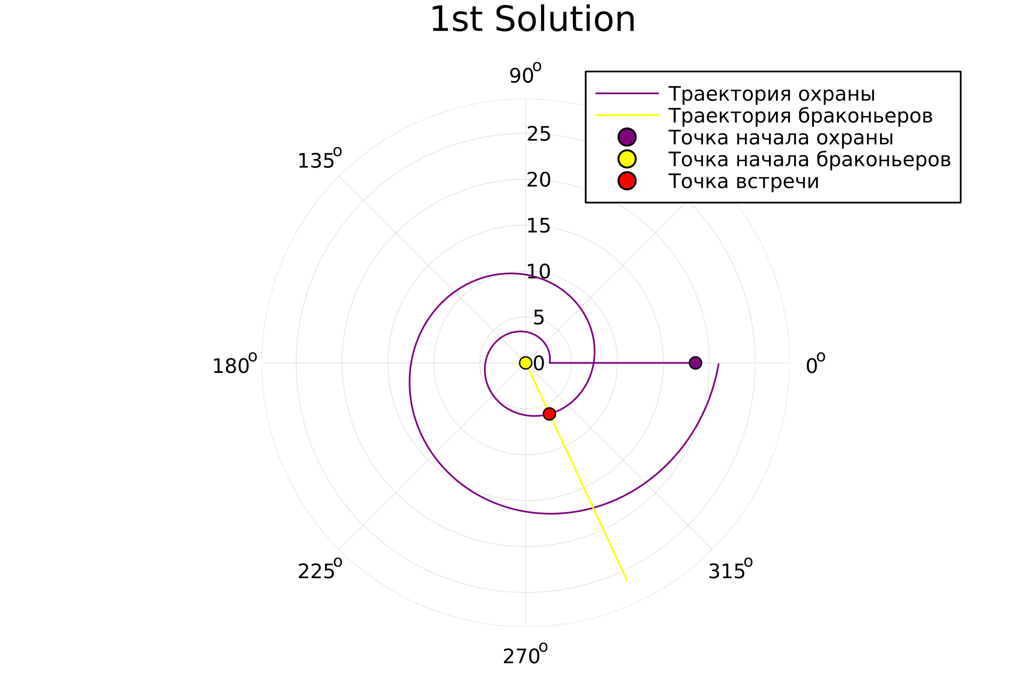
if condition  
   
else  
   
end

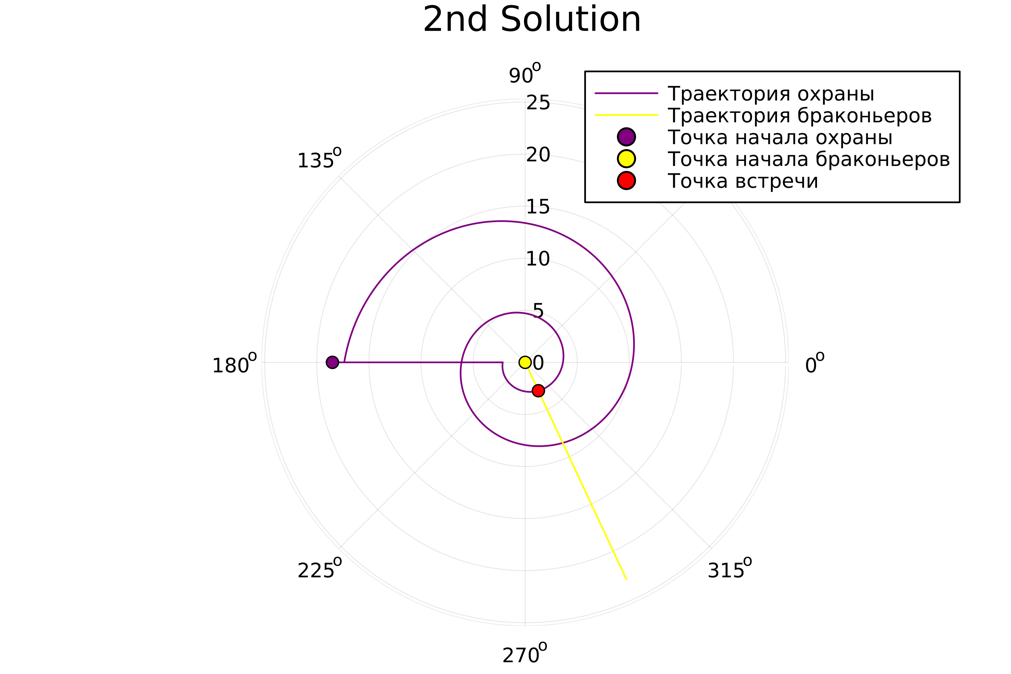


Изображение выглядит как текст

Автоматически созданное описание

Вывод программы:





На OpenModelica используя базовые настройки построение графика невозможно. В базовой настройке, графики могут выстраиваться исключительно в системе полярных координат.

# Код:

using Plots

const a = 18.5

const n = 6.1

const thetadeg = 295

const dTheta = 0.02

const maxTheta = 4pi

const cases = [" 1st ", " 2nd "]

function F(theta)

return r0 \* exp.(theta / sqrt.(n^2 - 1))

end

for case in cases

global r0 = -1

theta0 = -1

if case==" 1st "

r0 = a / (n + 1)

theta0 = 0

else

r0 = a / (n - 1)

theta0 = -pi

end

theta1 = theta0 + maxTheta

thetaHunt = theta0:dTheta:theta1

thetaPray = thetadeg \* pi / 180 + 2 \* theta0

plt = plot(proj=:polar, aspect\_ratio=:equal, dpi=500, title= case \* "Solution", legend=true)

plot!(plt, [theta0, theta0], [a, F(theta0)], label=false, color=:purple)

plot!(plt, thetaHunt, F, label=:"Траектория охраны" ,color=:purple)

plot!(plt, [0, thetaPray], [0, F(thetaPray) + 20], label=:"Траектория браконьеров", color=:yellow)

plot!(plt, [theta0], [a], seriestype=:scatter, label=:"Точка начала охраны", color=:purple)

plot!(plt, [0], [0], seriestype=:scatter, label=:"Точка начала браконьеров", color=:yellow)

plot!(plt, [thetaPray], [F(thetaPray)], seriestype=:scatter, label=:"Точка встречи", color=:red)

savefig(plt, "Lab02" \* case \* "Case.png")

end

# Вывод

Ознакомились с Julia подробнее – что дало возможность улучшить навыки в решении дифференциальных уравнений.

# Ресурсы

* Задача о погоне (https://esystem.rudn.ru/mod/resource/view.php?id=967233)
* Кривая погони (https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%B2%D0%B0%D1%8F\_%D0%BF%D0%BE%D0%B3%D0%BE%D0%BD%D0%B8)
* Julia (http://www.unn.ru/books/met\_files/JULIA\_tutorial.pdf)
* OpenModelica (https://ru.wikipedia.org/wiki/OpenModelica)