

Лабораторная работа № 2

Задача о погоне (Вариант 9)

Сулицкий Богдан Романович НФИбд-02-20

Содержание

Цели работы	5
Задания [1]	6
Теоретическое введение [2]	7
Выполнение лабораторной работы	8
Код на Julia[3]	8
Вывод	13
Список литературы	14

Список иллюстраций

1	Подключение библиотек и создание переменных	8
2	Функция уравнение и нахождение точки пересечения	9
3	Функция визуализации Pyplot	9
4	Функция визуализации Pyplot	10
5	Математическая модель I случая	11
6	Математическая модель II случая	12
7	Точки пересечения для двух случаев	12

Список таблиц

Цели работы

Целью данной работы является построение математической модели задачи о погоне.

Задания [1]

1. Записать уравнение, описывающее движение катера, с начальными условиями для двух случаев (в зависимости от расположения катера относительно лодки в начальный момент времени).
2. Построить траекторию движения катера и лодки для двух случаев.
3. Найти точку пересечения траектории катера и лодки.

Теоретическое введение [2]

Julia - высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений.[3]

Кривая погони — кривая, представляющая собой решение задачи о «погоне», которая ставится следующим образом. Пусть точка равномерно движется по некоторой заданной кривой. Требуется найти траекторию равномерного движения точки такую, что касательная, проведённая к траектории в любой момент движения, проходила бы через соответствующее этому моменту положение точки.

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 6,7 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 2,7 раза больше скорости браконьерской лодки.

Выполнение лабораторной работы

Код на Julia[3]

Подключаем нужные библиотеки и создаем переменные расстояния, разницы скорости и угла соприкосновения.[1]

```
using PyPlot
using DifferentialEquations

s = 6.7 # расстояние
v = 2.7 # разница в скорости
span = (0, 50)
angle = 5*pi/4 # угол соприкосновения
```

Рис. 1: Подключение библиотек и создание переменных

С помощью Differential Equations[3] создадим функцию уравнения и опишем метод нахождения точки пересечения.(2)


```

function f(du, u, p, t) # функция уравнения
    du[1] = 1
    du[2] = sqrt(v*v - 1) / u[1]
end

touchPoint(k, a) = for (i,k) in enumerate(k)
    if (round(k, digits = 3) == round(a, digits = 3))
        return intersection = r[i]
    end
end
end

```

Рис. 2: Функция уравнение и нахождение точки пересечения

Создадим функцию отображения математических моделей с помощью Pyplot.(3)

```

function draw() # отображение
    PyPlot.axes(polar="true")
    scatter(angle, intersection, color="red")
    plot(t, r, linestyle="-", color="green")
    plot([0, angle], [0, span[2]], color="k")
    println("Точка соприкосновения: (", angle, " ; ", intersection, ")")
    show()
    clf()
end

```

Рис. 3: Функция визуализации Pyplot

Решим ОДУ и вызовем функцию Pyplot.(4)

```

r0 = s / (v + 1) # случай 1
t0 = 0.0
ode = ODEProblem(f, [r0,t0], span)
sol = solve(ode, dtmax=0.001)
r = [u[1] for u in sol.u]
t = [u[2] for u in sol.u]
intersection = touchPoint(t, angle)
draw()

r0 = s / (v - 1) # случай 2
t0 = -2pi/3
ode = ODEProblem(f, [r0,t0], span)
sol = solve(ode, dtmax=0.001)
r = [u[1] for u in sol.u]
t = [u[2] for u in sol.u]
intersection = touchPoint(t, angle)
draw()

```

Рис. 4: Функция визуализации Pyplot

Результаты(5-7)

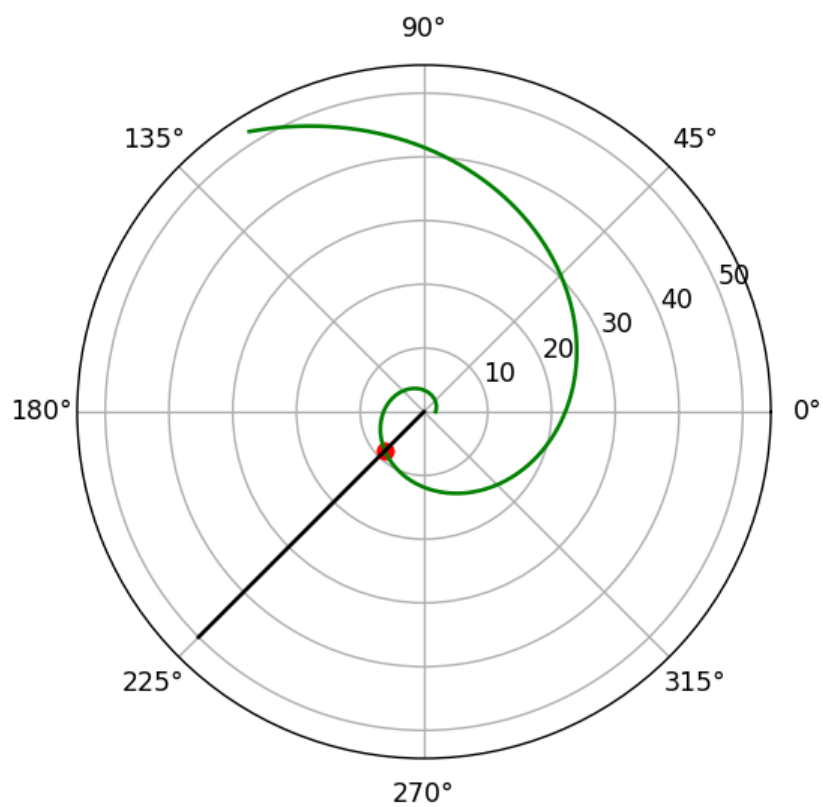


Рис. 5: Математическая модель I случая

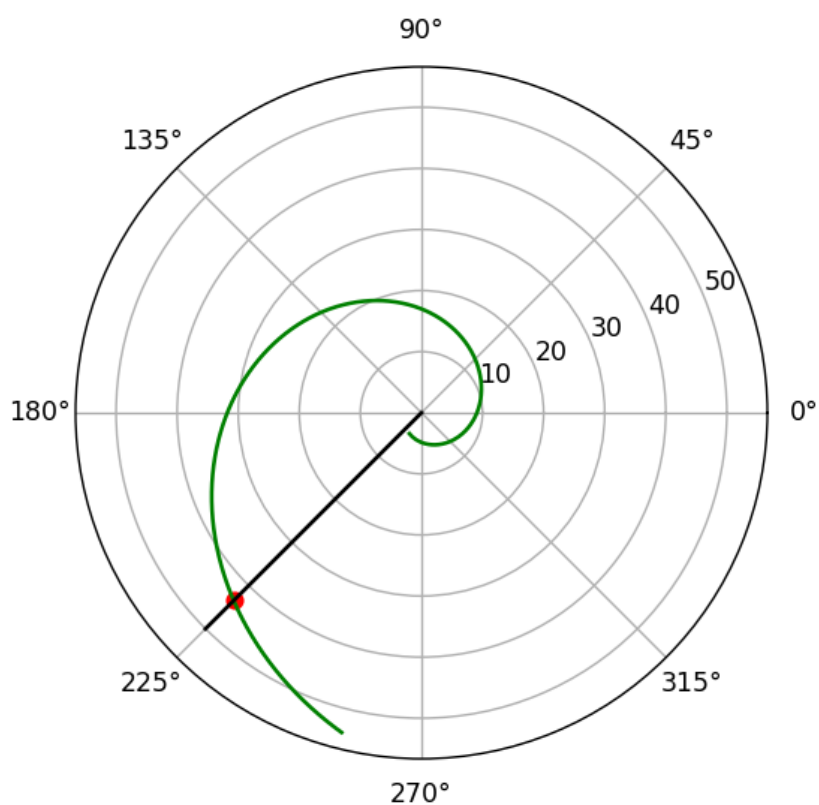


Рис. 6: Математическая модель II случая

Точка соприкосновения: (3.9269908169872414 ; 8.666532429801299)
Точка соприкосновения: (3.9269908169872414 ; 43.47517647057794)

Рис. 7: Точки пересечения для двух случаев

Вывод

В результате проделанной работы был написан код на Julia, и были построены математические модели траекторий движения катера и лодки для двух случаев.

Список литературы

1. Задания к лабораторной работе №2 (по вариантам) [Электронный ресурс]. RUDN, 2023. URL: https://esystem.rudn.ru/pluginfile.php/1971649/mod_resource/content/2/Задание%20к%20лабораторной%20работе%20№%205%20%2081%29.pdf.
2. Лабораторная работа №2 [Электронный ресурс]. RUDN, 2023. URL: https://esystem.rudn.ru/pluginfile.php/1971648/mod_resource/content/2/Лабораторная%20работа%20№%201.pdf.
3. DifferentialEquations.jl: Efficient Differential Equation Solving in Julia [Электронный ресурс]. 2023. URL: <https://docs.sciml.ai/DiffEqDocs/stable/>.