

Лабораторная работа 2

Задача о погоне

Фаик Карим

Содержание

Цель работы	1
Задание.....	1
Выполнение лабораторной работы	1
Код.....	4
Находим пересечение	4
Находим пересечение	5
Выводы.....	6

Цель работы

Решить задачу о погоне, на примере лодками с браконьером и береговой охраной.

Задание

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

Выполнение лабораторной работы

1. Зададим начальные условия (вариант 62)
 $k = 14.4; n = 4.7; t_0 = 0; t_{L0} = 0; t_{K0} = 18.1$
2. Будем вести отсчет в полярных координатах. Полус у нас это место обнаружения браконьеров.
3. Чтобы найти расстояние x (расстояние после которого катер начнет двигаться вокруг полюса), необходимо составить простое уравнение. Пусть через время t катер и лодка окажутся на одном расстоянии x от полюса. За это время лодка

пройдет x , а катер $k - x$ (или $k + x$, в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как x/v или $(k - x)/4.7v$ (во втором случае $(k + x)/4.7v$). Так как время одно и то же, то эти величины одинаковы. Составим уравнения и найдем расстояние x (@fig:001)

```
julia> boat_r = Float64[0.0, 100.0]
2-element Vector{Float64}:
 0.0
100.0

julia> boat_theta = Float64[7π/4]
1-element Vector{Float64}:
 5.497787143782138

julia>

julia> # Находим пересечение

julia> intersection_r = 0
0

julia> for (i,θ) in enumerate(θ)
    if (round(θ, digits=2) == round(boat_theta[1], digits=2))
        global intersection_r = R[i]
        break
    end
end

julia>

julia> @show intersection_r
intersection_r = 31.054092976970512
31.054092976970512

julia>

julia> plt = plot(
    proj = :polar,
    aspect ratio=:equal,
    dpi=300,
    title="Задача о погоне",
    legend=true)
```

Нахождение первого решения

```

julia> for (i,θ) in enumerate(θ)
    if (round(θ, digits=2) == round(boat_θ[1], digits=2))
        global intersection_r = R[i]
        break
    end
end

julia>

julia> @show intersection_r
intersection_r = 31.054092976970512
31.054092976970512

julia>

julia> plt2 = plot(
    proj = :polar,
    aspect_ratio=:equal,
    dpi=300,
    title="Задача о погоне",
    legend=true)

julia>

julia>

julia> plot!(
    plt2,
    θ,
    R,
    label="Траектория катера",
    color=:green)

julia>

julia> plot!(
    plt2,
    boat_θ,
    boat_r,
    label="Траектория лодки",
    color=:red)

julia>

julia> plot!(
    plt2,
    boat_θ,
    [intersection_r],
    seriestype = :scatter,
    label="Точка пересечения",
    color=:blue)

julia>

julia> savefig(plt2, "lab02_2.png")
"C:\\Users\\Gigabyte\\lab02_2.png"

```

Нахождение второго решения

4. После того, как катер береговой охраны окажется на одном расстоянии от полюса, что и лодка, он должен сменить прямолинейную траекторию и начать двигаться вокруг полюса удаляясь от него со скоростью лодки v . Для этого скорость катера раскладываем на радиальную v_r и тангенциальную v_t скорости

$$v_r = \frac{dr}{dt}$$
$$v_t = r \frac{d\theta}{dt} = v\sqrt{19.25}$$

5. Давайте просчитаем траекторию движения браконьеров и движения лодки охраны и найдем точки пересечения при помощи следующего кода

Код

```
``julia
using Plots using DifferentialEquations

function F(du,u,p,t) r, θ = u du[1] = 2 du[2] = sqrt(19.25) / u[1] end

r_0 = 14.4/5.7 h = 0.1 θ_0 = 0.0 tspan = (0, 100) prob = ODEProblem(F, [r_0, θ_0], tspan) sol =
solve(prob, dtmax=h)

#Достаем значения R = [u[1] for u in sol.u] Θ = [u[2] for u in sol.u]

boat_r = Float64[0.0, 100.0] boat_θ = Float64[7π/4]
```

Находим пересечение

```
intersection_r = 0 for (i,θ) in enumerate(Θ) if (round(θ, digits=2) == round(boat_θ[1],
digits=2)) global intersection_r = R[i] break end end

@show intersection_r

plt = plot( proj = :polar, aspect_ratio=:equal, dpi=300, title="Задача о погоне",
legend=true)

plot!( plt, Θ, R, label="Траектория катера", color=:green)

plot!( plt, boat_θ, boat_r, label="Траектория лодки", color=:red)

plot!( plt, boat_θ, [intersection_r], seriestype = :scatter, label="Точка пересечения",
color=:blue)

savefig(plt, "lab02_1.png")

r_0 = 18.1/3.7 θ_0 = π
```

```

prob = ODEProblem(F, [r0, θ0], tspan) sol = solve(prob, dtmax=h)
#Достаем значения R = [u[1] for u in sol.u] θ = [u[2] for u in sol.u]
boat_r = Float64[0.0, 100.0]

```

Находим пересечение

```

for (i,θ) in enumerate(θ) if (round(θ, digits=2) == round(boat_θ[1], digits=2)) global
intersection_r = R[i] break end end

```

```
@show intersection_r
```

```

plt2 = plot( proj = :polar, aspect_ratio=:equal, dpi=300, title="Задача о погоне",
legend=true)

```

```
plot!( plt2, θ, R, label="Траектория катера", color=:green)
```

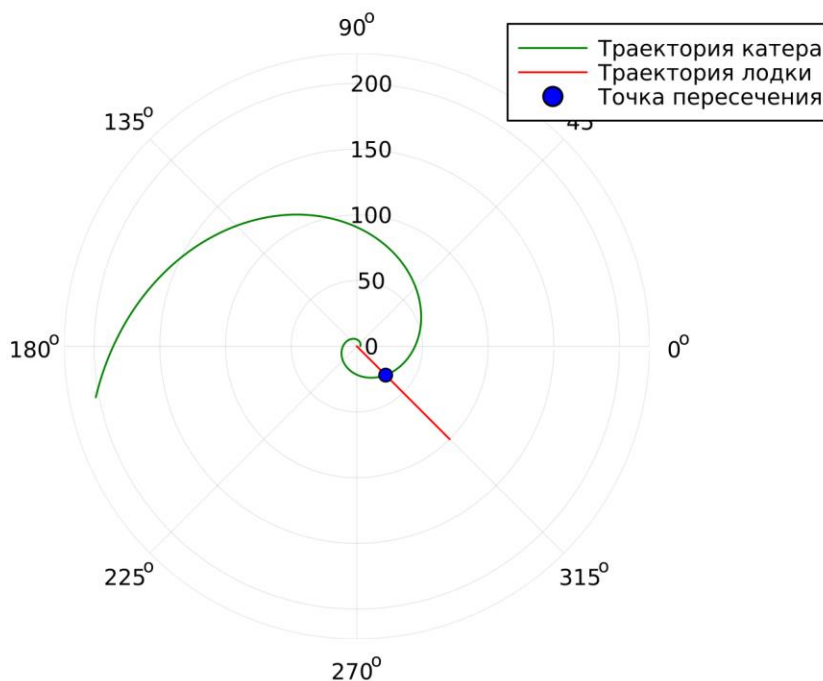
```
plot!( plt2, boat_θ, boat_r, label="Траектория лодки", color=:red)
```

```
plot!( plt2, boat_θ, [intersection_r], seriestype = :scatter, label="Точка пересечения",
color=:blue)
```

```
savefig(plt2, "lab02_2.png")
```

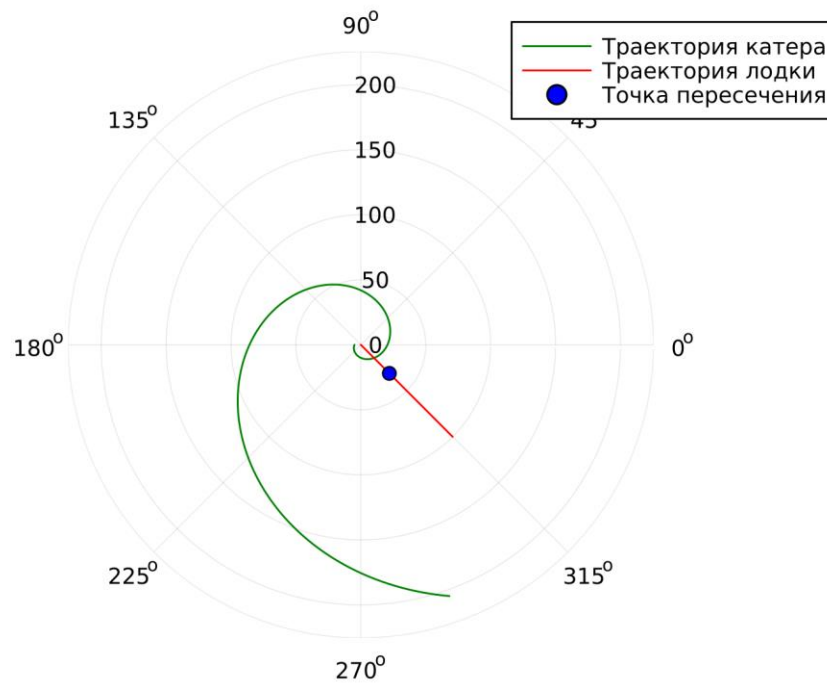
6. Далее получим следующие траектории движения

Задача о погоне



Траектория 1

Задача о погоне



Траектория 2

Выводы

По мере выполнения данной работы я научился работать с языком Julia посредством решения задачи о погоне на примере лодки с браконьерами и береговой охраны