Отчет по лабораторной работе №2

Задача о погоне

Махорин Иван Сергеевич

Содержание

[1 Цель работы 1](#_Toc159027113)

[2 Задание 1](#_Toc159027114)

[3 Выполнение лабораторной работы 1](#_Toc159027115)

[4 Моделирование с помощью Julia 3](#_Toc159027116)

[5 Выводы 7](#_Toc159027117)

[6 Список литературы 7](#_Toc159027118)

# 1 Цель работы

Решить задачу о погоне и изучить основы языка программирования Julia.

# 2 Задание

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

# 3 Выполнение лабораторной работы

Расчитаем свой вариант по формуле и получаем наш вариант №59 (рис. 3.1).



Рис. 1: Получение нужного номера варианта

1. На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 20,3 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 5,2 раза больше скорости браконьерской лодки.
2. Примем за момент отсчета времени момент первого рассеивания тумана. Введем полярные координаты с центром в точке нахождения браконьеров и осью, проходящей через катер береговой охраны. Тогда начальные координаты катера (20,3; 0). Обозначим скорость лодки v.
3. Траектория катера должна быть такой, чтобы и катер, и лодка все время были на одном расстоянии от полюса. Только в этом случае траектория катера пересечется с траекторией лодки. Поэтому для начала катер береговой охраны должен двигаться некоторое время прямолинейно, пока не окажется на том же расстоянии от полюса, что и лодка браконьеров. После этого катер береговой охраны должен двигаться вокруг полюса удаляясь от него с той же скоростью, что и лодка браконьеров.
4. Пусть время t - время, через которое катер и лодка окажутся на одном расстоянии от начальной точки.

$$ t = {{x }\over{v}} $$

$$ t = {{20,3-x}\over{5,2 v}} $$

$$ t = {{20,3+x}\over{5,2 v}} $$

Из этих уравнений получаем объедиение двух уравнений:

$$ \left[ \begin{array}{cl} {{x}\over{v}} = {{20,3-x}\over{5,2 v}}\\ {{x}\over{v}} = {{20,3+x}\over{5,2 v}} \end{array} \right. $$

Решая это, получаем два значения для x:

– тангенциальная скорость

– радиальная скорость

$$ v = {dr\over dt} $$

$$ v\_\tau = {{\sqrt{((5,2\*v)^2-v^2)}}} = {\sqrt{651}\*v \over{5}} $$

$$ \left\{ \begin{array}{cl} {dr\over dt} = v \\ r{d\theta\over dt} = {\sqrt{651}\*v \over{5}} \end{array} \right. $$

или

Итоговое уравнение после того, как убрали производную по t:

$$ {dr\over d\theta} = {5r\over\sqrt{651}} $$

# 4 Моделирование с помощью Julia

1. Скачиваем Julia (рис. 4.1).

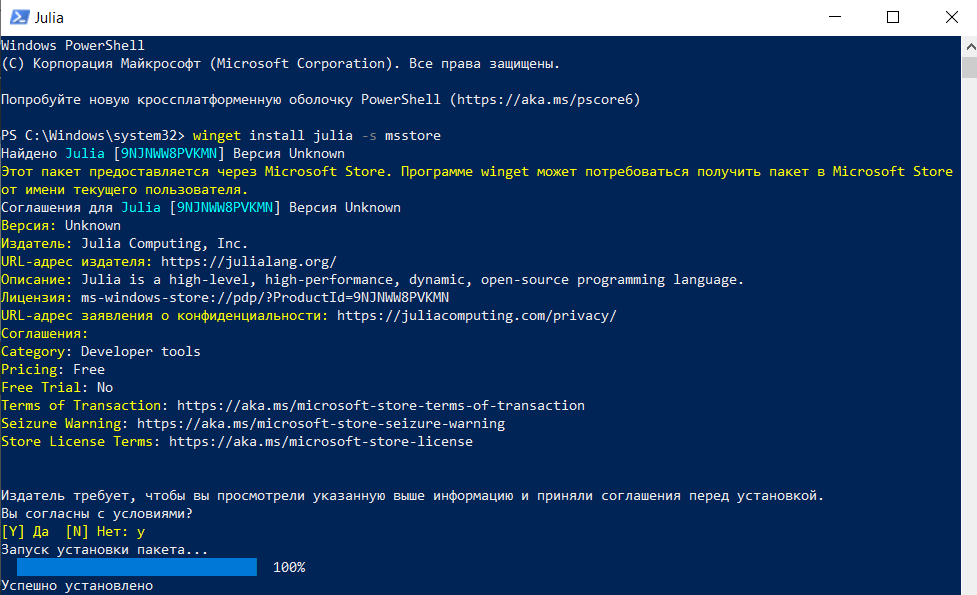


Рис. 2: Скачивание Julia

1. Запускаем Julia (рис. 4.2).

Запуск Julia

Рис. 3: Запуск Julia

1. Процесс запуска Julia (рис. 4.3).

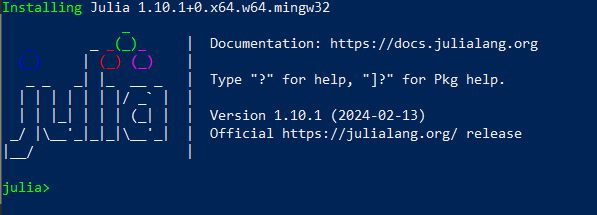


Рис. 4: Процесс запуска

1. Скачаем необходимые для работы пакеты (рис. 4.4-4.5).

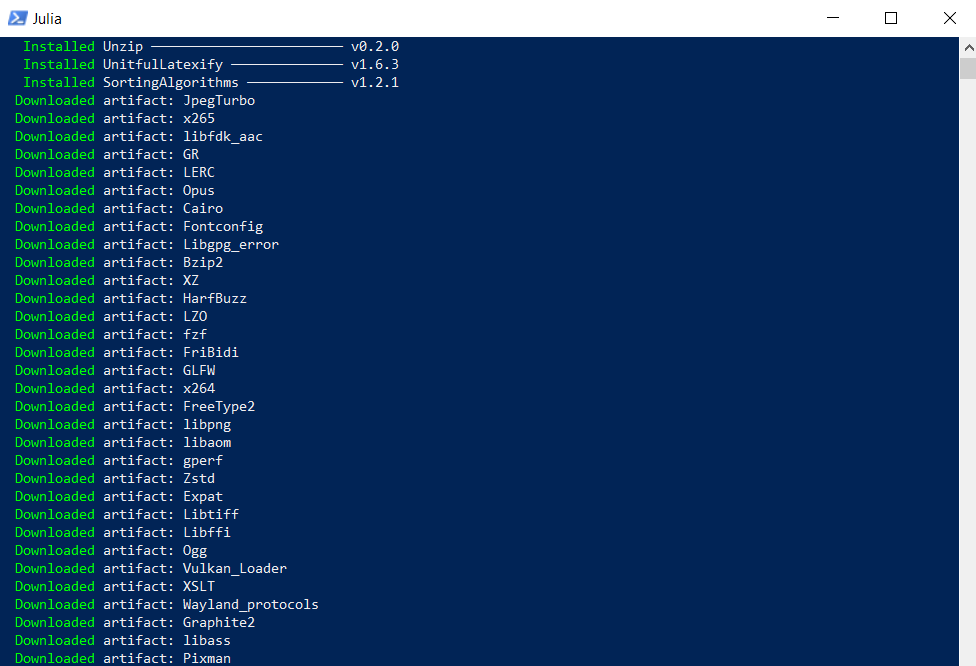


Рис. 5: Скачивание необходимых для работы пакетов

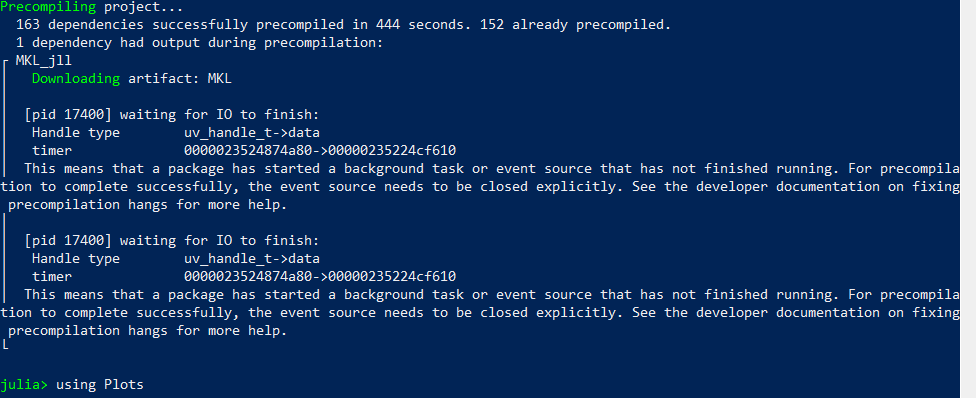


Рис. 6: Скачивание необходимых для работы пакетов

1. Код для файла lab2.jl:

using Plots using DifferentialEquations

const a = 20.3 const n = 5.2

const r0 = a/(n + 1) const r0\_2 = a/(n - 1)

const T = (0, 2\*pi) const T\_2 = (-pi, pi)

function F(u, p, t) return u / sqrt(n\*n - 1) end

problem = ODEProblem(F, r0, T)

result = solve(problem, abstol=1e-8, reltol=1e-8) [**show?**] result.u [**show?**] result.t

dxR = rand(1:size(result.t)[1]) rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]

plt = plot(proj=:polar, aspect\_ratio=:equal, dpi = 1000, legend=true, bg=:white)

plot!(plt, xlabel=“theta”, ylabel=“r(t)”, title=“Случай номер 1”, legend=:outerbottom) plot!(plt, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label=“Путь лодки”, color=:blue, lw=1) scatter!(plt, rAngles, result.u, label=““, mc=:blue, ms=0.0005) plot!(plt, result.t, result.u, xlabel=”theta”, ylabel=“r(t)”, label=“Путь катера”, color=:green, lw=1) scatter!(plt, result.t, result.u, label=““, mc=:green, ms=0.0005)

savefig(plt, “lab2\_01.png”)

problem = ODEProblem(F, r0\_2 , T\_2) result = solve(problem, abstol=1e-8, reltol=1e-8) dxR = rand(1:size(result.t)[1]) rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]

plt1 = plot(proj=:polar, aspect\_ratio=:equal, dpi = 1000, legend=true, bg=:white)

plot!(plt1, xlabel=“theta”, ylabel=“r(t)”, title=“Случай номер 2”, legend=:outerbottom) plot!(plt1, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label=“Путь лодки”, color=:blue, lw=1) scatter!(plt1, rAngles, result.u, label=““, mc=:blue, ms=0.0005) plot!(plt1, result.t, result.u, xlabel=”theta”, ylabel=“r(t)”, label=“Путь катера”, color=:green, lw=1) scatter!(plt1, result.t, result.u, label=““, mc=:green, ms=0.0005)

savefig(plt1, “lab2\_02.png”)

1. Запуск кода (рис. 4.6).

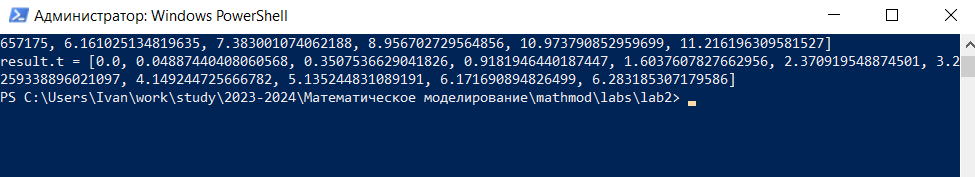


Рис. 7: Запуск кода

1. Просмотр результата работы (рис. 4.7-4.8).

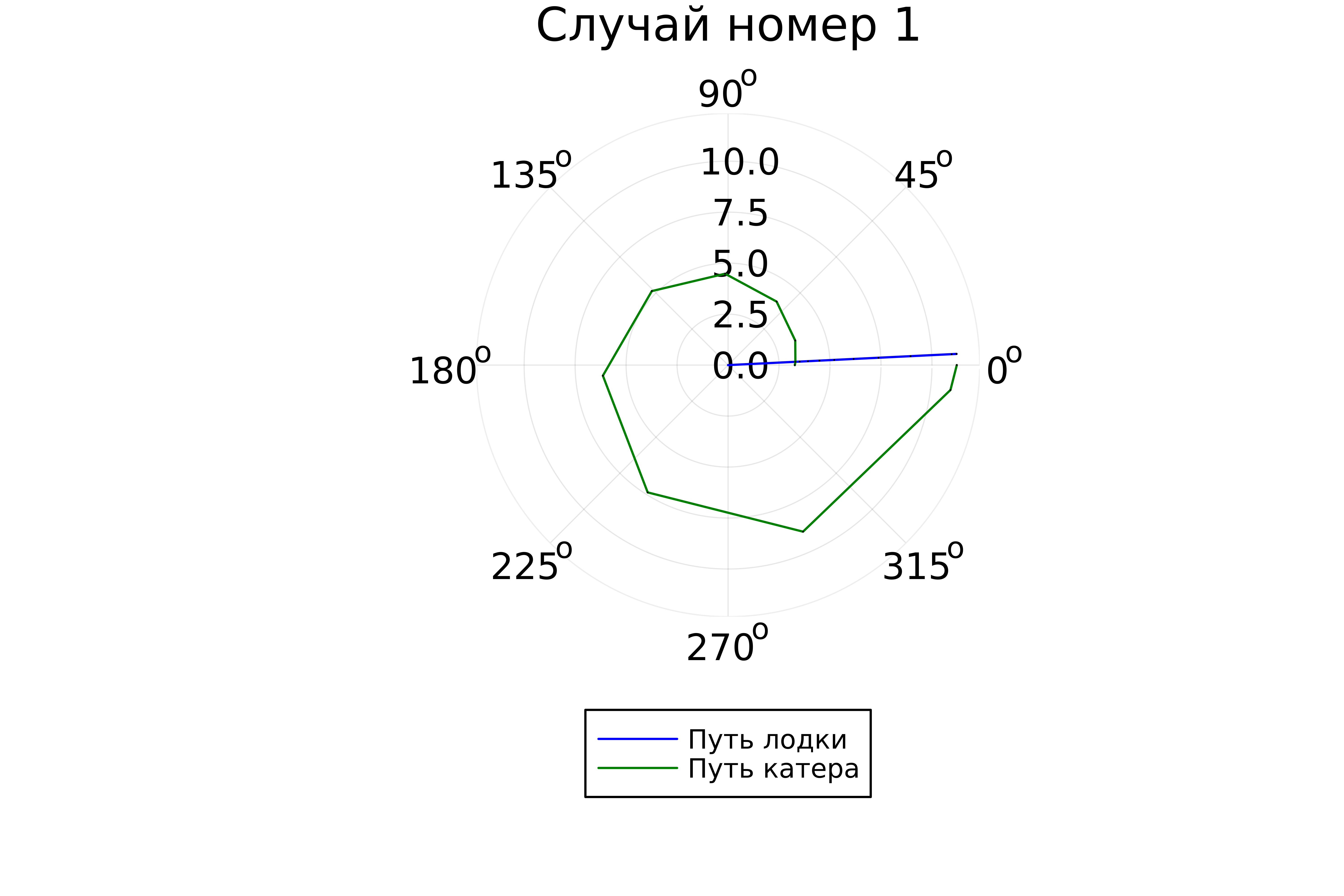


Рис. 8: Случай 1



Рис. 9: Случай 2

# 5 Выводы

Были изучены основы языка программирования Julia и его библиотеки, которые используются для построения графиков и решения дифференциальных уравнений. А также решили задачу о погоне.

# 6 Список литературы

[1] Документация по Julia: https://docs.julialang.org/en/v1/

[2] Решение дифференциальных уравнений: https://www.wolframalpha.com/