

- Front matter
  - Formatting pdf
  - Цель работы
  - Задание
  - Теоретическое введение
    - 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. Он имеет в своём составе сложный компилятор, обеспечивает распределённое параллельное выполнение инструкций, вычислительную точность и обширную библиотеку математических функций.
    - Возможности языка:
  - Ход работы
    - Первый способ
    - Второй способ
  - Выводы
- 

## Front matter

---

lang: ru-RU title: Лабораторная работа №2 subtitle: Задача о погоне author:

- Бабенко Артём Сергеевич institute:
- Российский университет дружбы народов, Москва, Россия

date: 16.02.2024

## Formatting pdf

---

toc: false toc-title: Содержание slide\_level: 2 aspectratio: 169 section-titles: true theme: metropolis header-includes:

- \metroset{progressbar=frametitle,sectionpage=progressbar,numbering=fraction}
- 'makeatletter'

- `'\beamer@ignorenonframefalse'`
  - `'\makeatother'`
- 

## Цель работы

---

- Научиться решать задачи поиска с помощью построения математических моделей для выбора правильной стратегии.

## Задание

---

### Вариант 3

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 7 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 3 раза больше скорости браконьерской лодки.

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

## Теоретическое введение

---

**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. Он имеет в своём составе сложный компилятор, обеспечивает распределённое параллельное выполнение инструкций, вычислительную точность и обширную библиотеку математических функций.

---

## Возможности языка:

---

множественная диспетчеризация (мультиметод): обеспечение возможности определять поведение функции при различных комбинациях типов аргументов; динамическая типизация; хорошая производительность, приближающаяся к производительности статических языков; встроенный менеджер пакетов; макросы и другие объекты метапрограммирования; функции обработки вызовов Python: пакет PyCall; функции прямой обработки вызовов C без надстроек; мощные возможности оболочки для управления другими процессами; возможности обеспечения параллелизма и распределённых вычислений; эффективная поддержка кодировки Unicode. Язык Julia распространяется бесплатно вместе с исходными кодами.

## Ход работы

---

# Первый способ

1. Принимает за  $t_0 = 0$ ,  $x_{л0} = 0$  - место нахождения лодки браконьеров в момент обнаружения,  $x_{к0} = k$  - место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения лодки.

$k = 7$  (км)

2. Введем полярные координаты. Считаем, что полюс - это точка обнаружения лодки браконьеров  $x_{л0}$  ( $\theta = x_{л0} = 0$ ), а полярная ось  $r$  проходит через точку нахождения катера береговой охраны (рис. 5.1)

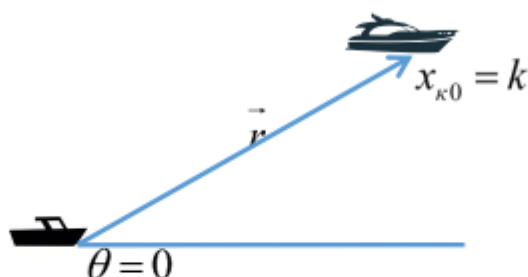


Рис.51.1. Положение катера и лодки в начальный момент времени

3. Траектория катера должна быть такой, чтобы и катер, и лодка все время были на одном расстоянии от полюса  $\theta$ , 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. 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

Время, за которое они пройдут это расстояние, вычисляется как  $x/v$  или  $(k-x)/3v$ , во втором случае  $(x+k)/3v$ . Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние  $x$  можно найти из следующего уравнения:

$$\frac{x}{v} = \frac{7-x}{3v}$$

в первом случае или

$$\frac{x}{v} = \frac{7+x}{3v}$$

во втором.

Отсюда мы найдем два значения  $x = 7/4$ ,  $x = 7/2$ , задачу будем решать для двух случаев.

5. 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 две составляющие:  $v_r$  - радиальная скорость и  $v_\tau$  - тангенциальная скорость (рис. 2). Радиальная

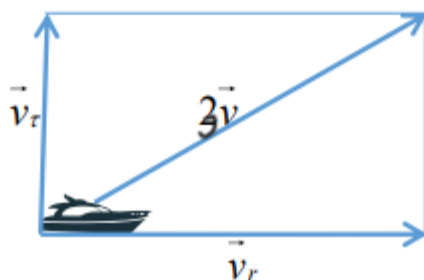
скорость - это скорость, с которой катер удаляется от полюса,  $v_r = \frac{dr}{dt}$ . Нам

нужно, чтобы эта скорость была равна скорости лодки, поэтому полагаем

$$\frac{dr}{dt} = v.$$

Тангенциальная скорость - это линейная скорость вращения катера относительно полюса. Она равна произведению угловой скорости  $\frac{d\theta}{dt}$  на

радиус  $r$ ,  $v_\tau = r \frac{d\theta}{dt}$



Из рисунка видно:  $\vec{v}_t = \sqrt{9v^2 - v^2} = 2\sqrt{2}v$  (учитывая, что

$$r \frac{d\theta}{dt} = \sqrt{3}v$$

радиальная скорость равна  $v$ ). Тогда получаем:

6. Решение исходной задачи сводится к решению системы из двух дифференциальных уравнений

$$\begin{cases} \frac{dr}{dt} = v \\ r \frac{d\theta}{dt} = 2\sqrt{2}v \end{cases}$$

$$\begin{cases} \theta_0 = 0 \\ r_0 = 7/4 \end{cases} \quad \text{или} \quad \begin{cases} \theta_0 = -\pi \\ r_0 = 7/2 \end{cases}$$

с начальными условиями

или

Исключая из полученной системы производную по  $t$ , переходим к следующему уравнению:

$$\frac{dr}{d\theta} = \frac{7}{8\sqrt{2}}$$

в первом случае и

$$\frac{dr}{d\theta} = \frac{7}{4\sqrt{2}}$$

во втором.

Начальные условия остаются прежними. Решив эти уравнения, мы получим траекторию движения катера в полярных координатах.

## Второй способ

---

Код на Julia:

```

using Plots
using DifferentialEquations

# расстояние от лодки до катера
const a = 7
const n = 3

# расстояние начала спирали
const r0 = a/(n + 1)
const r0_2 = a/(n - 1)
# интервал
const T = (0, 2*pi)
const T_2 = (-pi, pi)

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

# задача ОДУ
problem = ODEProblem(F, r0, T)

#решение
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]]

#холст1
plt = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)

#параметры для холста
plot!(plt, xlabel="theta", ylabel="r(t)", title="Задача о погоне - случай 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, "lab02_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]]

#холст2
plt1 = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)

#параметры для холста
plot!(plt1, xlabel="theta", ylabel="r(t)", title="Задача о погоне - случай 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, "lab02_02.png")

```

## Запуск файла с кодом:

Администратор: Windows PowerShell

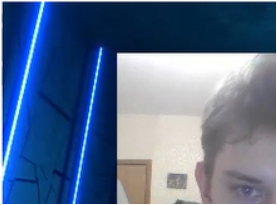
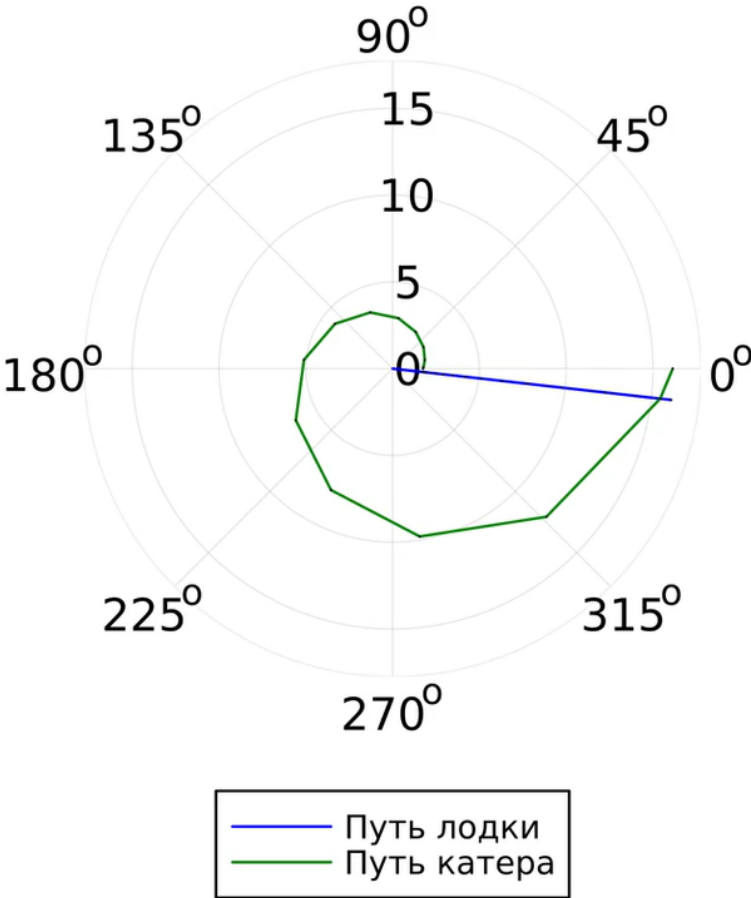
```

[11] _start()
@ Base .\client.jl:552
PS C:\Users\Admin\Documents\2023-2024\MathematicalModeling\mathmod\labs\lab02> julia lab002.jl

```

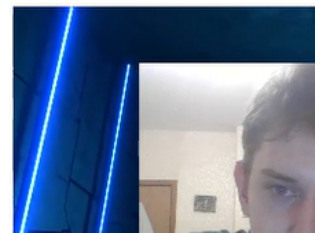
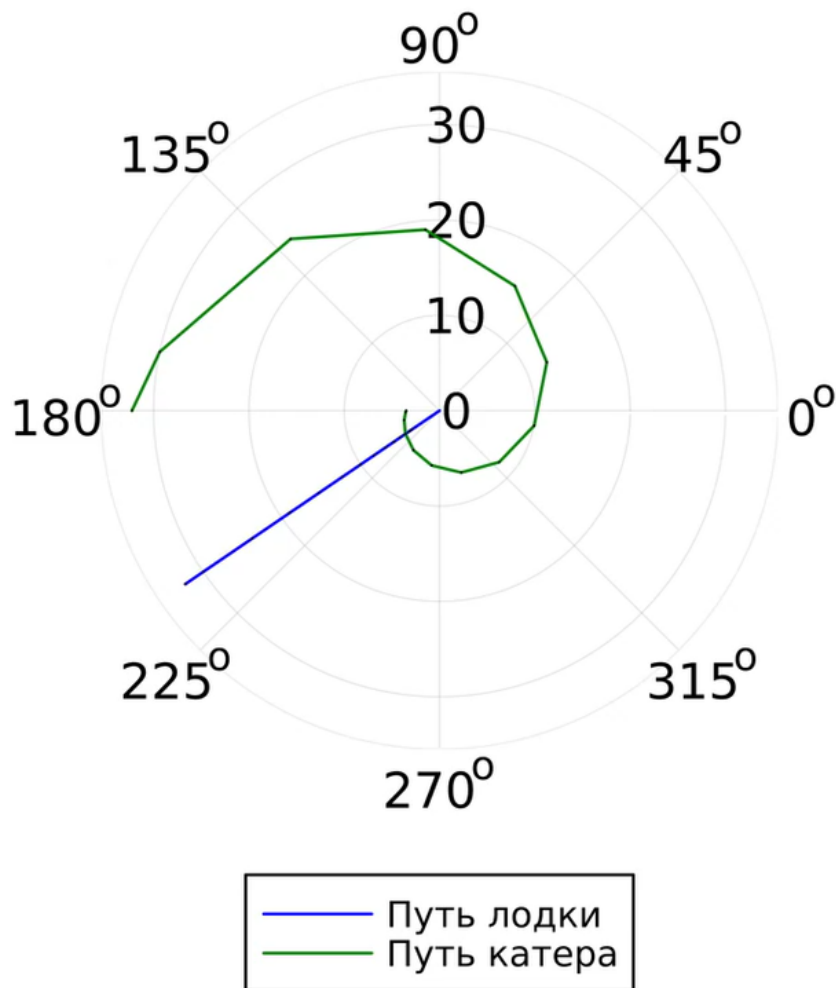
Результат: траектории движения лодки и катера для двух случаев.

# Задача о погоне - случай 1





# Задача о погоне - случай 2



По полученным изображениям вычислил точку пересечения катера и лодки.

## Выводы

Я ознакомился с языком Julia, его синтаксисом и научился решать задачи поиска с помощью построения математических моделей для выбора правильной стратегии.