

# Лабораторная работа №2

## Задача о погоне

---

Парфенова Е. Е.

15 февраля 2024

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Парфенова Елизавета Евгеньевна
- студент
- Российский университет дружбы народов
- 1032216437@pfur.ru
- <https://github.com/parfenovae>



# Вводная часть

---

- Важность изучения языка программирования Julia
- Необходимость умения строить математические модели к разным задачам

- Составить математическую модель для задачи о погоне
- Решить задачу о погоне
- Изучить основы языка программирования Julia.

# Теоретическое введение

---

*Julia* — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков (например, MATLAB и Octave), однако имеет некоторые существенные отличия. Julia написан на Си, C++ и Scheme. Имеет встроенную поддержку многопоточности и распределённых вычислений, реализованные в том числе в стандартных конструкциях. Язык возник 23 августа 2009 года.

Основной задачей при создании была разработка универсального языка, способного работать с большим объёмом вычислений и при этом гарантировать максимальную производительность.



На официальном сайте представлен алгоритм установки Julia для разных систем. Установка и настройка версии 1.10.0 заключаются в следующем наборе команд, который загружает последнюю версию Julia в каталог с именем `julia-1.10.1`:

```
wget https://julialang-s3.julialang.org/bin/linux/x64/1.10/julia-1.10.1-linux-x86_64.tar.gz  
tar zxvf julia-1.10.1-linux-x86_64.tar.gz
```

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

*Дифференциальное уравнение* — уравнение, которое помимо функции содержит её производные.

*Полярная система координат* - двумерная система координат, в которой каждая точка на плоскости определяется двумя числами — полярным углом и полярным радиусом.

*Тангенциальная скорость* - это скорость объекта, совершающего круговое движение, то есть движущегося по круговой траектории.

*Радиальная скорость* - это скорость изменения вектора смещения между двумя точками.

## **Задание лабораторной работы**

---

## Определение варианта

Вариант определялся по формуле из ТУИСа:  $(S_n \bmod N) + 1$ , где  $S_n$  — номер студбилета,  $N$  — количество заданий.

Мой вариант - 8.

### *Задача о погоне. Вариант 8*

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 6,5 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 2,6 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

# Выполнение лабораторной работы

---



1. Примем за начальный момент времени момент обнаружения лодки браконьеров, то есть момент, когда туман рассеялся.
2. Введем полярные координаты, считая, что точка обнаружения лодки браконьеров - это полюс, а полярная ось проходит через точку нахождения береговой охраны. Тогда координаты катера  $(6,5; 0)$

3. Далее необходимо найти расстояние после которого катер начнет двигаться вокруг полюса. Так как траектория катера пересечется с траекторией лодки только в случае того, если судна будут двигаться на одном расстоянии от полюса. Поэтому некоторое время катер береговой охраны должен двигаться прямолинейно, а затем, когда окажется на том же расстоянии от полюса, что и лодка, начать двигаться вокруг полюса.

## Построение математической модели

4. Составим систему простых уравнений. За время  $t$  лодка пройдет  $x$ , а катер береговой охраны  $6.5 - x$ . Примем скорость лодки браконеров за  $v$ . Следовательно время будет равно  $\frac{x}{v}$  для лодки и  $\frac{6.5-x}{2.6v}$  или  $\frac{6.5+x}{2.6v}$  для катера. Учитывая, что время должно быть равно, получается:

$$\begin{cases} \frac{x}{v} = \frac{6.5 - x}{2.6v} \\ \frac{x}{v} = \frac{6.5 + x}{2.6v} \end{cases}$$

Решив систему, мы получили два значения  $x$ :  $x_1 = \frac{65}{36}$ , а  $x_2 = \frac{65}{16}$

5. Как только катер береговой охраны окажется на том же расстоянии от полюса, что и лодка, он начнет двигаться вокруг полюса удаляясь от него со скоростью лодки браконьеров  $v$ . Скорость  $v$  раскладывается на 2 значения:  $v_r = \frac{dr}{dt}$  - радиальная скорость и  $v_\tau = r * \frac{d\theta}{dt}$  - тангенциальная скорость.
6. Нам необходимо составить систему дифференциальных уравнений. Первое уравнение у нас уже есть:  $v_r = \frac{dr}{dt}$ . Второе уравнение мы найдем из разложения скорости на две составляющие с помощью теоремы Пифагора:

$$v_\tau = \sqrt{(2.6v)^2 - v_r^2} = \sqrt{6.76v^2 - v^2} = 2.4v$$

## Построение математической модели

Следовательно второе уравнение выглядит так:  $r * \frac{d\theta}{dt} = 2.4v$

Тогда система уравнений получается:

$$\begin{cases} \frac{dr}{dt} = v \\ r * \frac{d\theta}{dt} = 2.4v \end{cases}$$

# Построение математической модели

С начальными условиями:

(для первого случая)

$$\begin{cases} \theta = 0 \\ r_0 = \frac{65}{36} \end{cases}$$

(для второго случая)

$$\begin{cases} \theta = -\pi \\ r_0 = \frac{65}{16} \end{cases}$$

Путем математических манипуляций приводим систему к такому виду:

$$\frac{dr}{d\theta} = \frac{r}{2.4}$$

Математическая модель готова.

Julia на системы Unix устанавливается в соответствии с алгоритмом в теоретическом введении. Настройка языка программирования для дальнейшей работы включает в себя:

- добавление папки bin наше Julia в системную PATH переменную окружения
- загрузка всех необходимых библиотек

При правильной установке и настройке Julia открывается командой **julia** из КОНСОЛИ

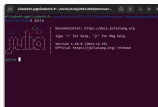


Рис. 1: Запуск Julia



# Процесс решения задачи

Для решения задачи необходимо вставить сделанный код в открытую Julia и дождаться пока сгенерируются два изображения (решения задачи для двух случаев)

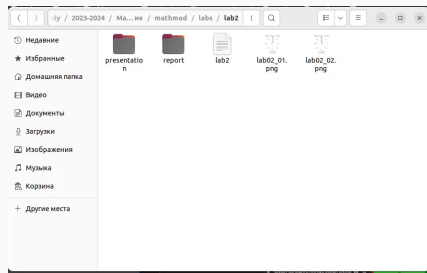


Рис. 2: Сгенерированные изображения

## Решение задачи для 1 случая

На рисунке изображены траектории движения лодки и катера для первого случая и наглядно видна точка их пересечения

Точка пересечения: 2,5 по полярному радиусу и 72 градуса по полярному углу



Рис. 3: Решение задачи для 1 случая

## Решение задачи для 2 случая

На рисунке изображены траектории движения лодки и катера для второго случая и наглядно видна точка их пересечения

Точка пересечения: 8 по полярному радиусу и 23 градуса по полярному углу



Рис. 4: Решение задачи для 2 случая

## Вывод

---

Мы успешно решили задачу о погоне для двух случаев, составив математическую модель для нашего варианта задачи. Построили два изображения и наглядно нашли точки пересечения траекторий. Также изучили основы программирования на языке Julia