Отчёт по лабораторной работе

Лабораторная работа №2 (вариант 10)

Сергеев Тимофей Сергеевич

Содержание

# 1 Цель работы

Построить математическую модель для выбора правильной стратегии при решении задач поиска на примере задачи преследования браконьеров береговой охраной.

# 2 Задание

* Подготовить инструменты для выполнения лабораторной работы;
* Записать уравнение, описывающее движение катера, с начальными условиями для двух случаев;
* Построить траекторию движения катера и лодки для двух случаев;
* Найти точку пересечения траектории катера и лодки.

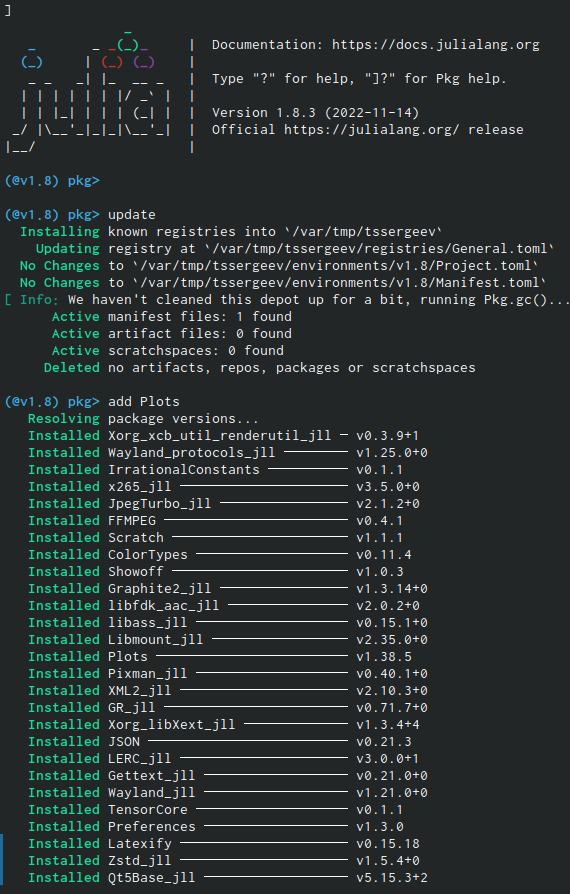
# 3 Теоретическое введение

**Julia** – высокоуровневый высокопроизводительный 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. [1]

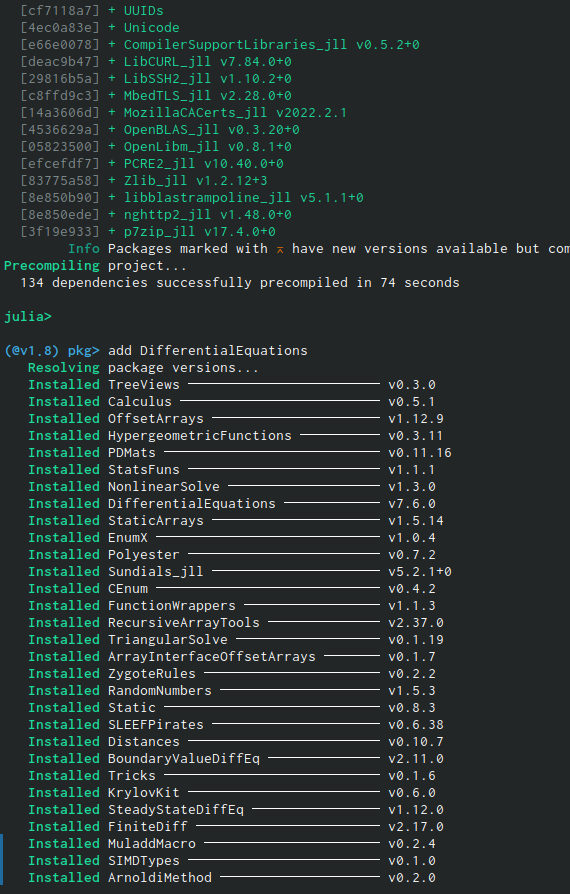
**OpenModelica** – свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. Активно развивается Open Source Modelica Consortium, некоммерческой неправительственной организацией. Open Source Modelica Consortium является совместным проектом RISE SICS East AB и Линчёпингского университета. [2]

# 4 Выполнение лабораторной работы

1. Настраиваем рабочую среду для выполнения лабораторной работы, а именно обновляем Julia, добавляем библиотеки Plots и DifferentialEquations (рис. ??, ??).

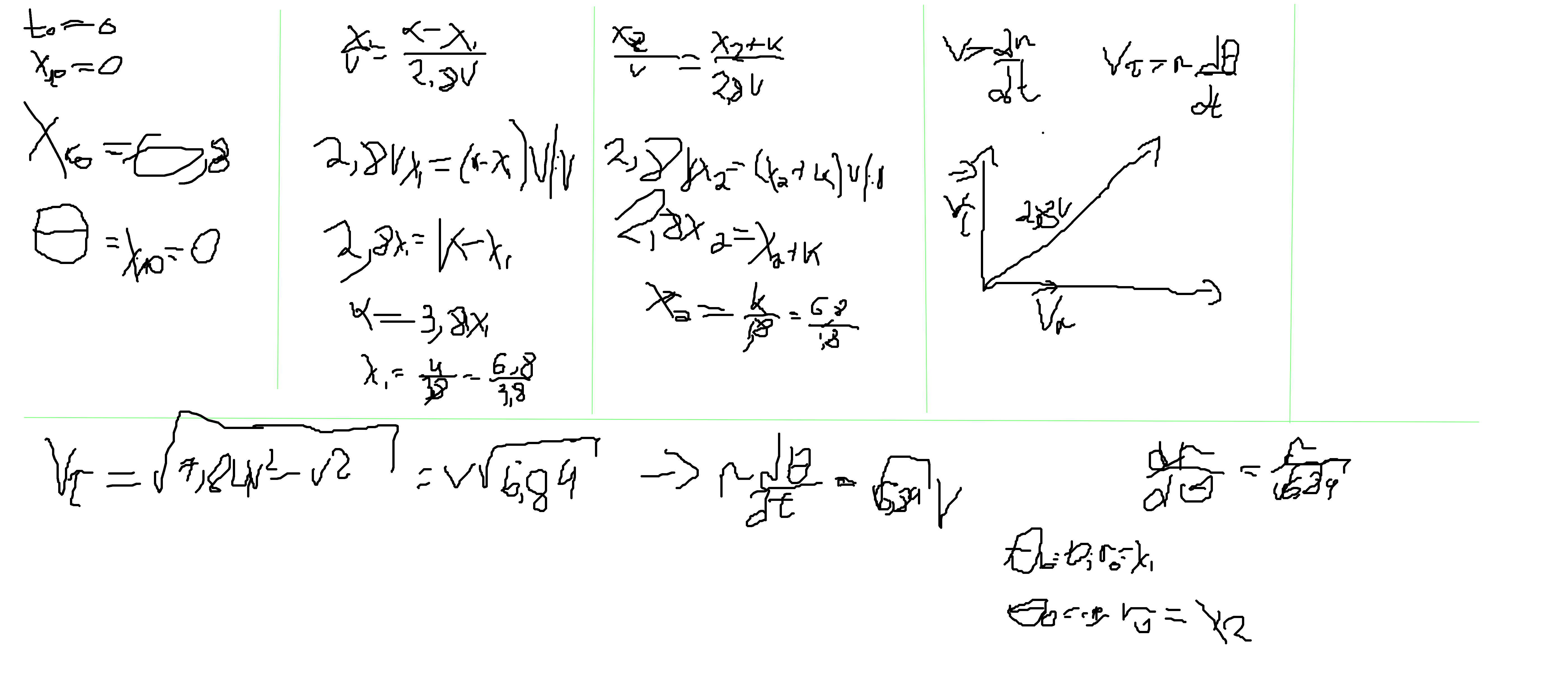


Обновляем Julia и добавляем библиотеку Plots

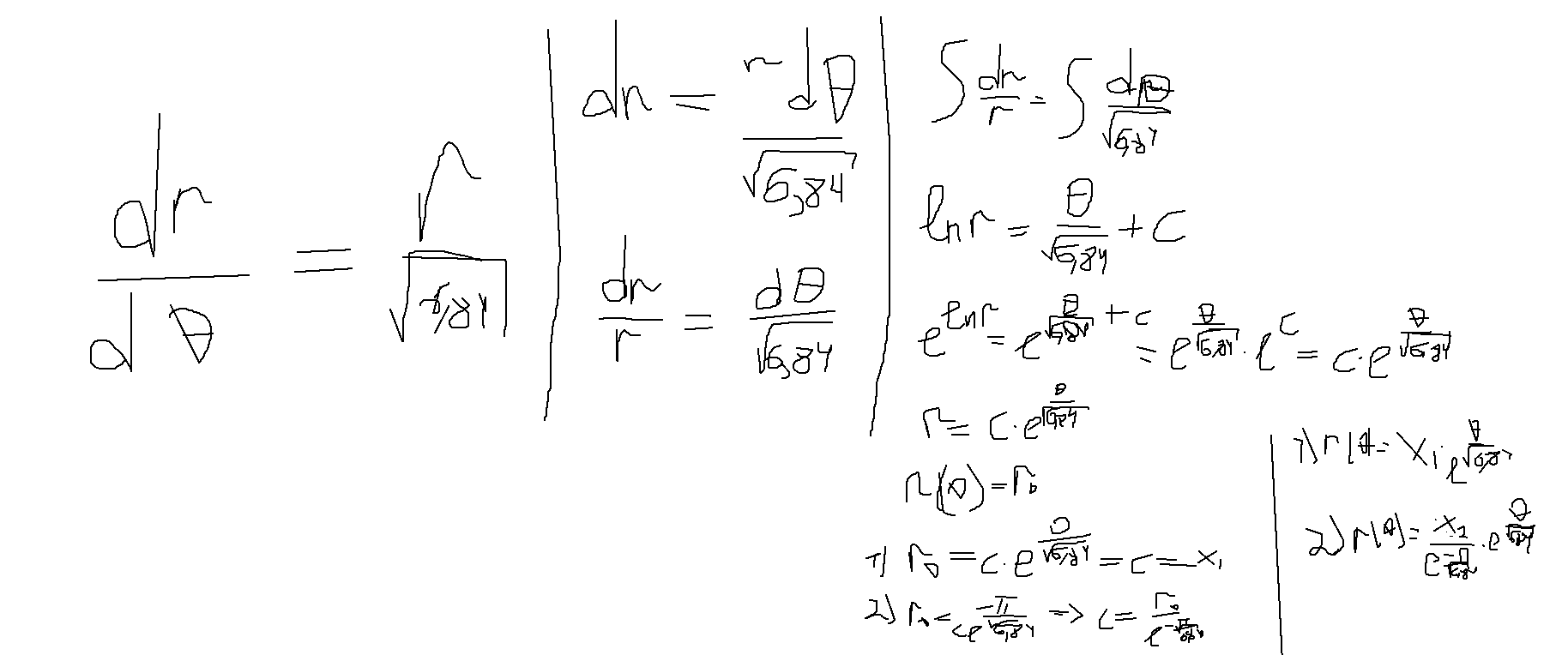


Добавляем библиотеку DifferentialEquations

1. Далее необходимо произвести математические расчёты в соответствии с материалами к лабораторной работе, но использовав свои данные, после чего дорешать получившееся уравнение. После выполнения расчётов ниже, мы получим траекторию движения катера в полярных координатах (рис. ??, ??).

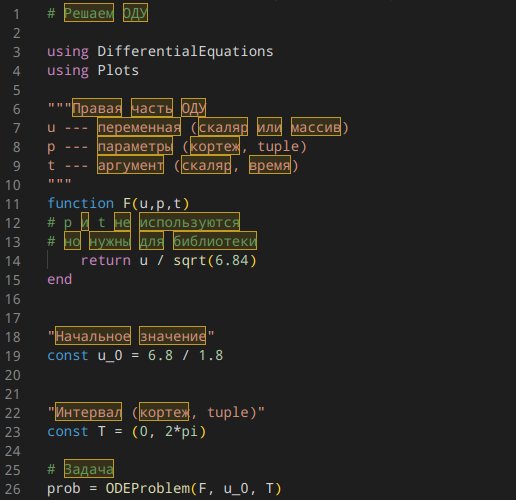


Вычисления для дальнейшего использования их в программе, часть 1

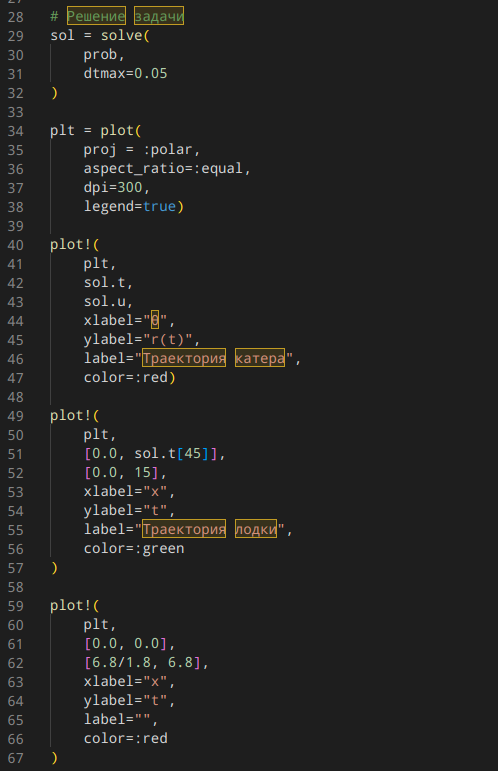


Вычисления для дальнейшего использования их в программе, часть 2

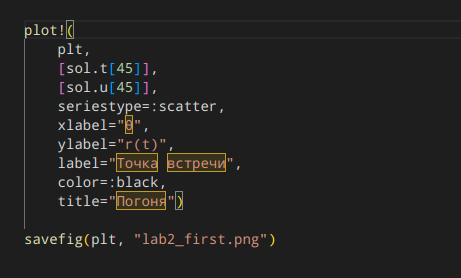
1. Теперь приступим к написанию программы на Julia (рис. ??, ??, ??).



В соответствии с теорией, рассказанной на семинаре, создаём функцию, задаём начальные значения

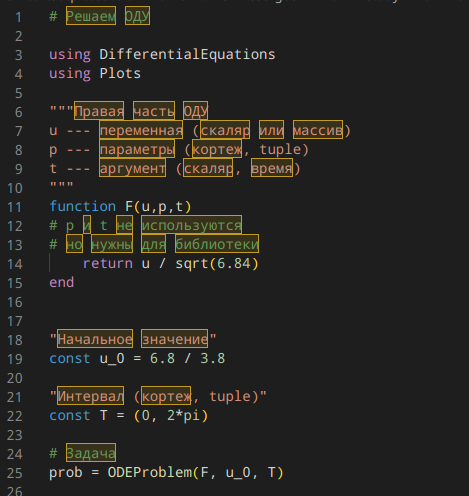


Затем переходим к непосредтсвенному добавлению элементов на основе полученных расчётов с помощью библиотеки Plots (создаём поле, задаём траектории катера, лодки а так же доходим до начала движения охраны их их первоначального положения; задаём точность, чтобы взять большее количество точек, тем самым получим более точный и плавный график



Обозначим точку, в которой охранники догонят браконьеров и сохраним полученный результат в формате png

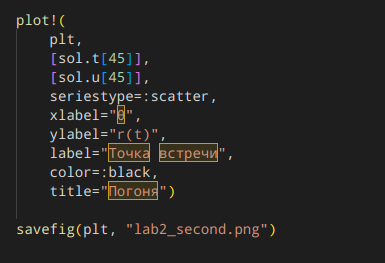
1. В зависимости от начального положения катера относительно полюса появляется второй случай (рис. ??, ??, ??).



В соответствии с теорией, рассказанной на семинаре, создаём функцию, задаём начальные значения



Затем переходим к непосредственному добавлению элементов на основе полученных расчётов с помощью библиотеки Plots (создаём поле, задаём траектории катера, лодки а так же доходим до начала движения охраны их их первоначального положения



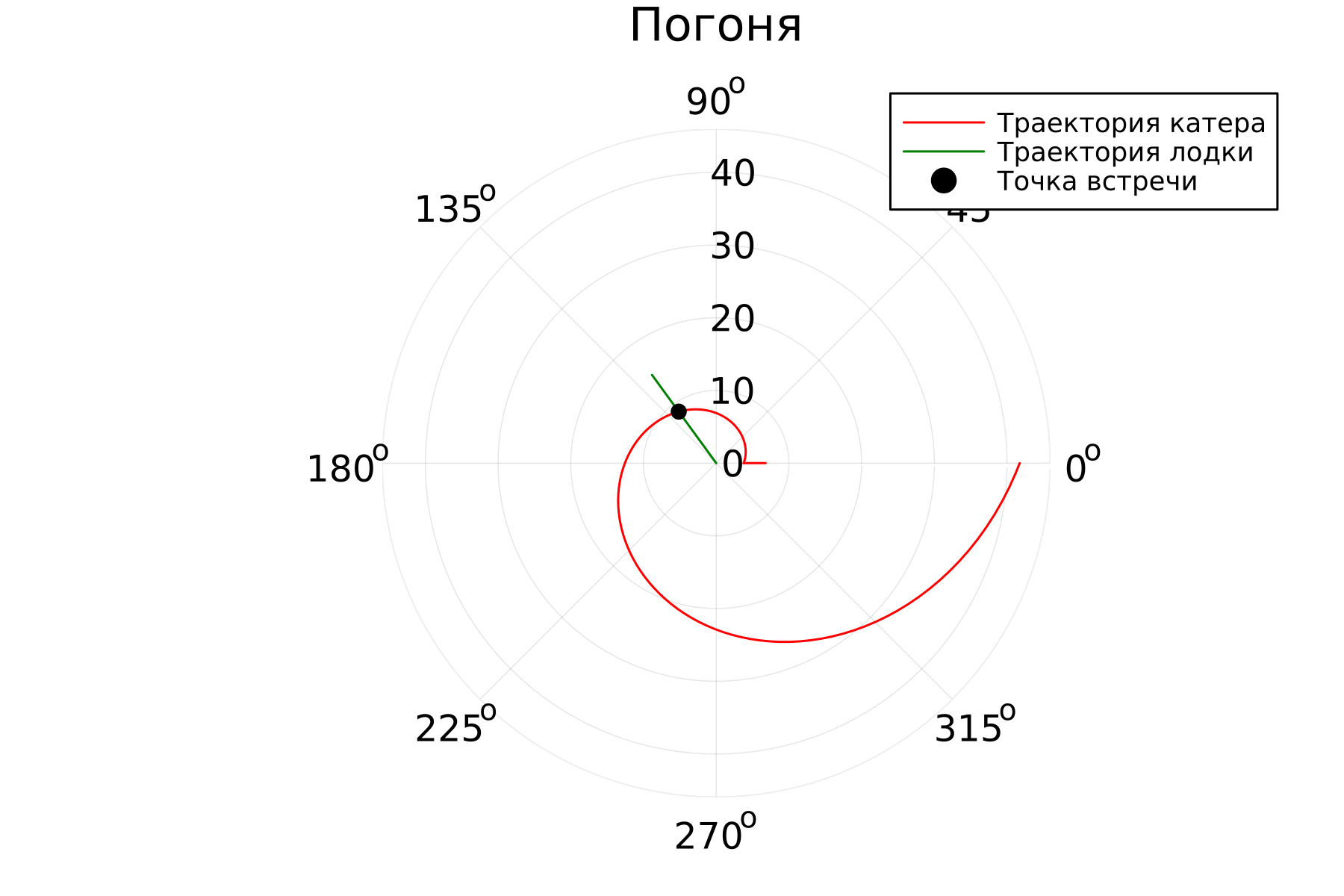
Обозначим точку, в которой охранники догонят браконьеров и сохраним полученный результат в формате png

1. Последним действием необходимо запустить и выполнить программы следующим образом (рис. ??).

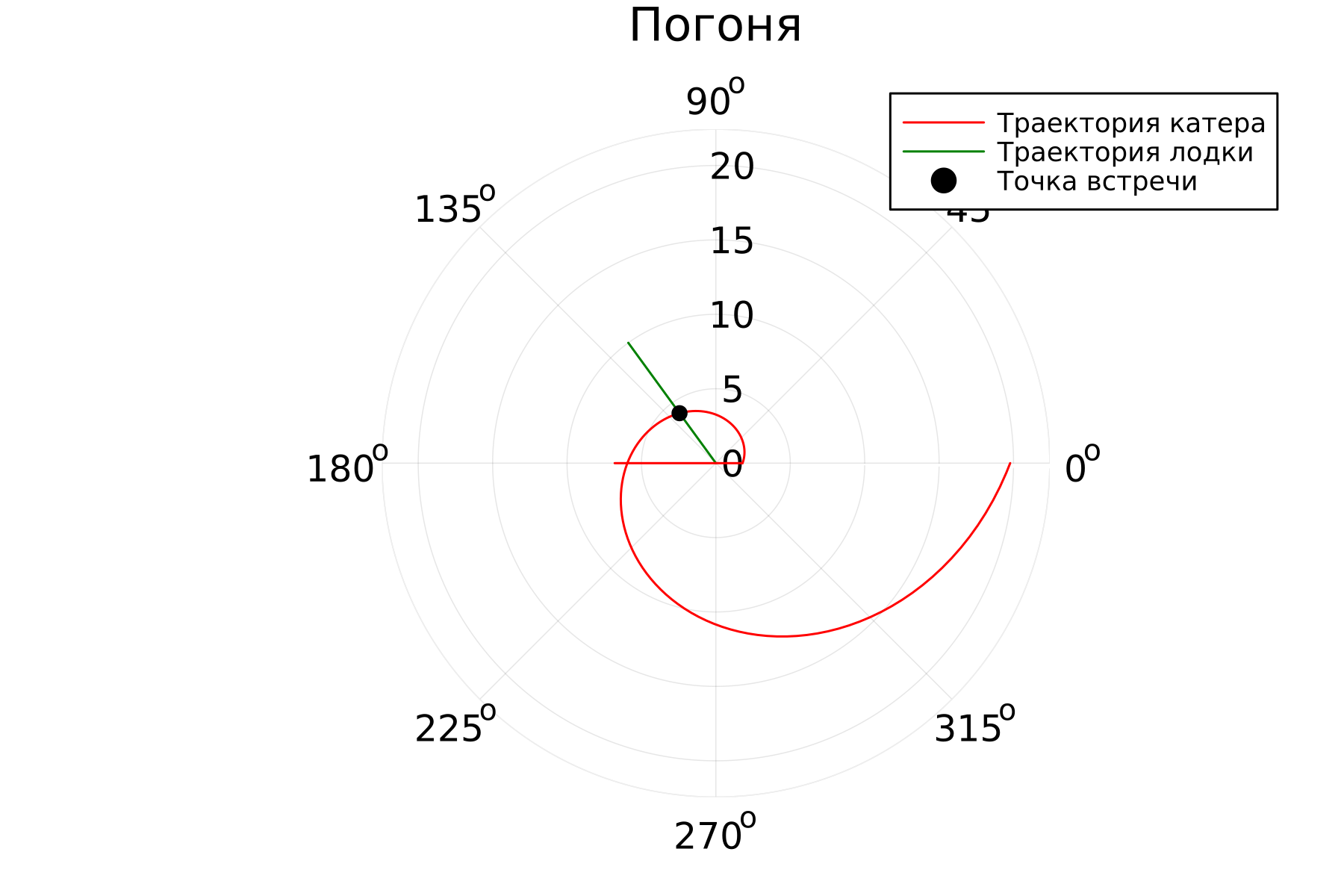
Запускаем и выполняем программы

Запускаем и выполняем программы

1. Получаем следующий результат (рис. ??, ??).



Результат первого случая



Результат второго случая

1. Результаты соответствуют поставленной задаче: сперва береговая охрана доходит до точки начала движения в зависимости от положения относительно полюса, после чего начинает движение и доходит до точки пересечения с браконьерами, которая так же обозначена на рисунке.
2. С помощью стандартных решений OpenModelica не получится представить решение данной программы, поэтому приступим к решению лабораторных работ на этом языке при выполнении следующей лабораторной работы.

# 5 Выводы

Выполнив данную лабораторную работу, мы познакомились с языком программирования Julia и с некоторыми его основами, которые понадобятся нам при выполнении следующих лабораторных работ. В дальнейших работах будем осваивать язык OpenModelica и сравнивать реализации программ на этих языках.

# Список литературы

1. Julia [Электронный ресурс]. Free Software Foundation, 2023. URL: <https://ru.wikipedia.org/wiki/Julia_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)>.

2. OpenModelica [Электронный ресурс]. Free Software Foundation, 2023. URL: <https://ru.wikipedia.org/wiki/OpenModelica>.