# Отчёт по лабораторной работе №2 Математическое моделирование

**Задача о погоне. Вариант №38**

Щербак Маргарита Романовна, НПИбд-02-21

2024

# Содержание

# Список иллюстраций

# Цель работы

Рассмотреть пример построения математических моделей для выбора правильной стратегии при решении задач поиска. С помощью примера научиться решать задачи такого типа. Ознакомиться с основами языков программирования Julia и OpenModelica. Освоить библиотеки этих языков, необходимые для визуализации данных и решения дифференциальных уравнений. Применить полученные знания к решению задачи о погоне.

# Задание

Я выполняю свой вариант лабораторной работы №38 по данной формуле $(N_{student} mod K_{ofvariants})$ + 1 = (1032216537 % 70) + 1 = 38.

Мой вариант будет применяться во всех последующих лабораторных работах.

**Задача о погоне**

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 19 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 5.1 раза больше скорости браконьерской лодки.

# Задачи

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

# Теоретическое введение

Scilab – это интерактивная система для выполнения численных вычислений, анализа данных и визуализации результатов. Она предоставляет пользователю удобное программное средство для решения математических задач различной сложности, включая решение уравнений, обработку сигналов, построение графиков и многое другое [1].

Подобно MATLAB, Scilab предоставляет широкий набор инструментов и функций, которые могут быть использованы для задач, начиная от простых математических операций и заканчивая сложными научными и инженерными вычислениями [1].

Julia – это высокоуровневый язык программирования с динамической типизацией, созданный для эффективных математических вычислений и написания программ общего назначения. В отличие от некоторых других языков, таких как MATLAB и Octave, Julia имеет свой собственный, но знакомый синтаксис. Он разработан на базе языков программирования C, C++ и Scheme. Julia обладает встроенной поддержкой многопоточности и распределенных вычислений, что делает его мощным инструментом для решения различных задач [2].

OpenModelica – это свободное программное обеспечение для моделирования и анализа сложных динамических систем, основанное на языке Modelica. OpenModelica приближается по функциональности к таким инструментам, как Matlab Simulink и Scilab xCos, но обладает более удобным представлением системы уравнений [3].

Дифференциальное уравнение – это уравнение, содержащее производные функции. Оно может быть любого порядка, и его решение представляет собой функцию (или семейство функций), в отличие от решения алгебраических уравнений, где ищется число или несколько чисел. Дифференциальное уравнение высокого порядка можно преобразовать в систему уравнений первого порядка, что облегчает его решение [4].

Тангенциальная скорость – это составляющая вектора скорости, перпендикулярная линии, соединяющей источник и наблюдателя. Она измеряется через угловое перемещение источника.

Радиальная скорость – это проекция скорости точки на прямую, соединяющую её с выбранным началом координат.

Полярная система координат – двумерная система координат, в которой каждая точка на плоскости определяется полярным углом и полярным радиусом.

# Выполнение лабораторной работы

Построим математическую модель для выбора правильной стратегии при решении задачи поиска. Необходимо определить по какой траектории должен двигаться катер, чтобы нагнать лодку.

1. Момент отсчета времени — момент первого рассеивания тумана. Введем полярные координаты с центром в точке обнаружения лодки браконьеров и осью, проходящей через точку нахождения катера береговой охраны (рис.1). Тогда начальные координаты катера (19; 0). Обозначим скорость лодки $v$.

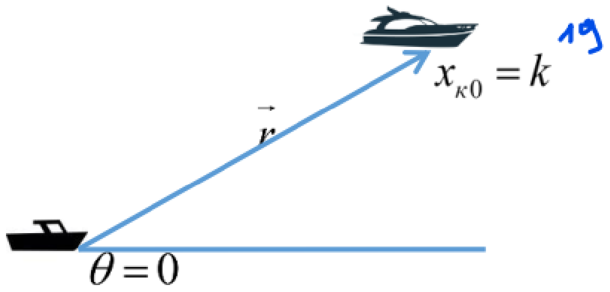



Рис. 1: Положение катера и лодки в начальный момент времени

2. 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.

3. Чтобы найти расстояние $x$ (расстояние после которого катер начнет двигаться вокруг полюса), необходимо составить простое уравнение [1]. Пусть через время $t$ катер и лодка окажутся на одном расстоянии $x$ от полюса. За это время лодка пройдет $x$, а катер $19-x$ (или $19+x$, в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как $\frac{x}{v}$ или $\frac{19-x}{5.1v}$ (во втором случае $\frac{19+x}{5.1v}$). Так как время одно и то же, то эти величины одинаковы. Из этого получаем объединение из двух уравнений (из-за двух разных изначальных позиций катера относительно полюса). Тогда неизвестное расстояние x можно найти из следующего уравнения [4]:

$$\left[\begin{array}{l} \frac{x}{v} = \frac{19-x}{5.1v} \\ \frac{x}{v} = \frac{19+x}{5.1v} \end{array}\right.$$

Отсюда мы найдем два значения $x_1 = \frac{190}{61}$, $x_2 = \frac{190}{41}$ (рис.2 - рис.3), задачу будем решать для двух случаев.

$$\frac{x}{v} = \frac{19-x}{5{,}1 \times v}$$

$$\mathbf{x = \frac{190}{61}, \ v \neq 0}$$

Рис. 2: Первое значение

$$\frac{x}{v} = \frac{19+x}{5{,}1 \times v}$$

$$x = \frac{190}{41}, \quad v \neq 0$$

Рис. 3: Второе значение

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. 4). 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, что $\frac{dr}{dt} = v$. Тангенциальная скорость – это линейная скорость вращения катера относительно полюса. Она равна $v_\tau$ = r $\frac{d\Theta}{dt}$

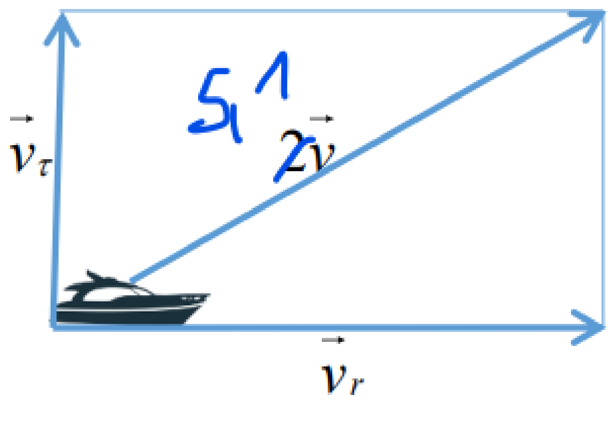



Рис. 4: Разложение скорости катера на тангенциальную и радиальную составляющие

Из рисунка видно:

$$v_\tau = \sqrt{5.1^2 v^2 - v^2} = \frac{\sqrt{2501}v}{10}$$

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

$$\begin{cases} \frac{dr}{dt} = v \\ r\frac{d\theta}{dt} = \frac{\sqrt{2501}v}{10} \end{cases}$$

с начальными условиями

$$\begin{cases} \theta_0 = 0 \\ r_0 = x_1 = \frac{190}{61} \end{cases}$$

или

$$\begin{cases} \theta_0 = -\pi \\ r_0 = x_2 = \frac{190}{41} \end{cases}$$

Исключая из полученной системы производную по $t$, можно перейти к следующему уравнению (с прежними начальными условиями):

$$\frac{dr}{d\theta} = \frac{10r}{\sqrt{2501}}$$

Решением этого уравнения с заданными начальными условиями и будет являться траектория движения катера в полярных координатах.

### OpenModelica

Оказалось, что OpenModelica не поддерживает использование полярных координат, что делает невозможным корректное отображение результатов данной задачи в этой среде.

**Julia**

Для решения дифференциального уравнения, описанного в постановке задачи лабораторной работы, можно использовать библиотеку DifferentialEquations. Для построения итоговых графиков в полярных координатах можно воспользоваться библиотекой Plots.

Я установила Julia и нужные библиотеки, проверила их установку (рис.5).

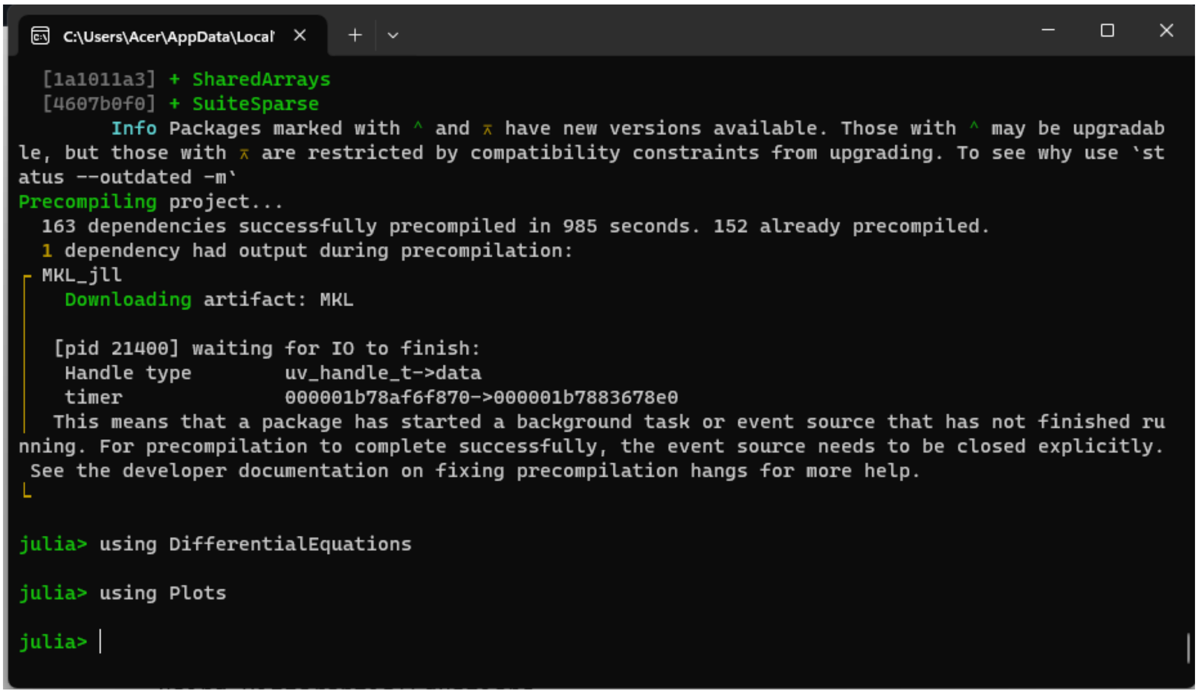



Рис. 5: Проверка установки библиотек

Однако код буду писать в Scilab. Код представлен для двух случаев (рис.6).

```
k=19; //начальное расстояние от лодки до катера
fi=3*%pi/4;
n=5.1 //отношение скорости катера к скорости лодки

//начальные условия в 1 случае
r0=k/6.1
tetha0=0;
```

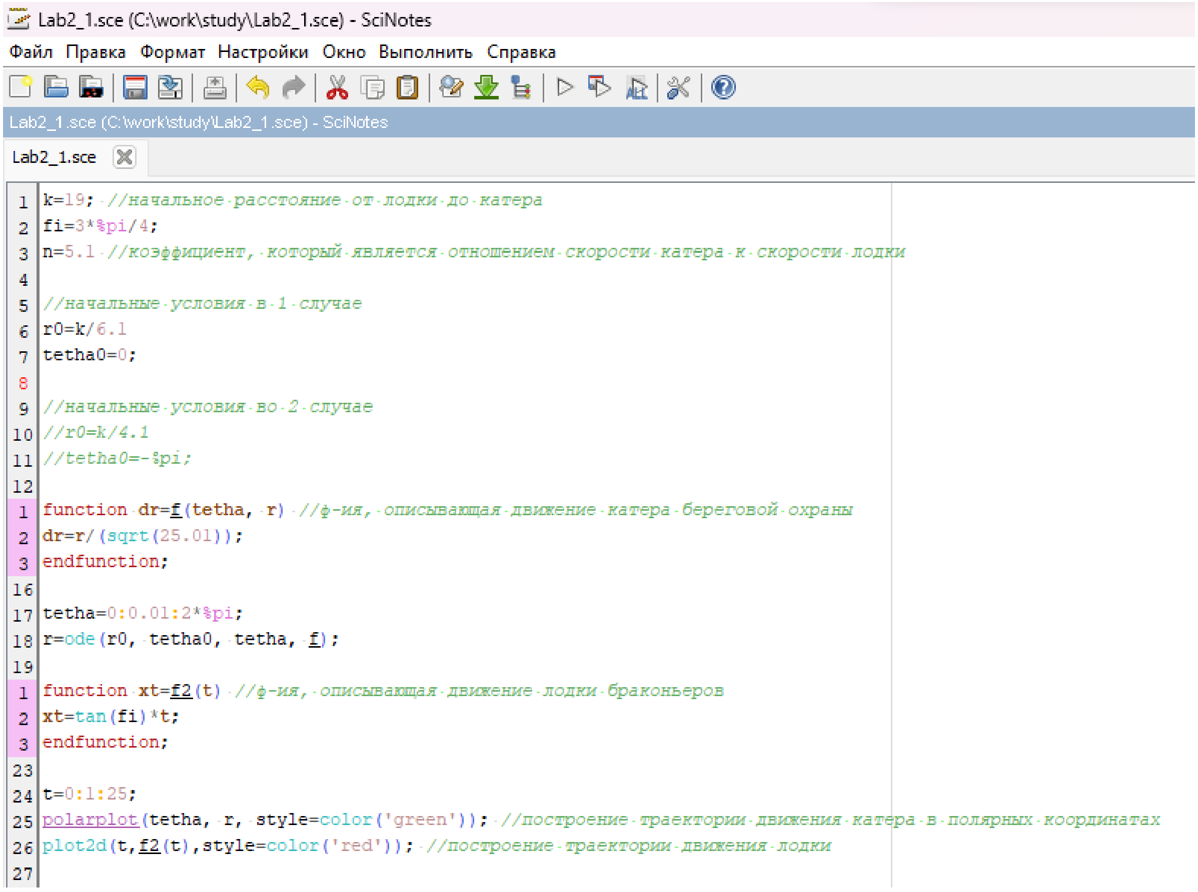
```
//начальные условия во 2 случае
//r0=k/4.1
//tetha0=-%pi;

function dr=f(tetha, r) //ф-ия, описывающая движение катера береговой охраны
dr=r/(sqrt(25.01));
endfunction;

tetha=0:0.01:2*%pi;
r=ode(r0, tetha0, tetha, f);

function xt=f2(t) //ф-ия, описывающая движение лодки браконьеров
xt=tan(fi)*t;
endfunction;

t=0:1:25;
//траектория движения катера в полярных координатах
polarplot(tetha, r, style=color('green'));
plot2d(t,f2(t),style=color('red')); //траектория движения лодки
```




```
k=19; //начальное расстояние от лодки до катера
fi=3*%pi/4;
n=5.1 //коэффициент, который является отношением скорости катера к скорости лодки

//начальные условия в 1 случае
r0=k/6.1
tetha0=0;

//начальные условия во 2 случае
//r0=k/4.1
//tetha0=-%pi;

function dr=f(tetha, r) //ф-ия, описывающая движение катера береговой охраны
dr=r/(sqrt(25.01));
endfunction;

tetha=0:0.01:2*%pi;
r=ode(r0, tetha0, tetha, f);

function xt=f2(t) //ф-ия, описывающая движение лодки браконьеров
xt=tan(fi)*t;
endfunction;

t=0:1:25;
polarplot(tetha, r, style=color('green')); //построение траектории движения катера в полярных координатах
plot2d(t,f2(t),style=color('red')); //построение траектории движения лодки
```

Рис. 6: Код в Scilab

Определила траектории катера (зелёный цвет) и лодки (красный цвет) для 1 случая (рис.7)

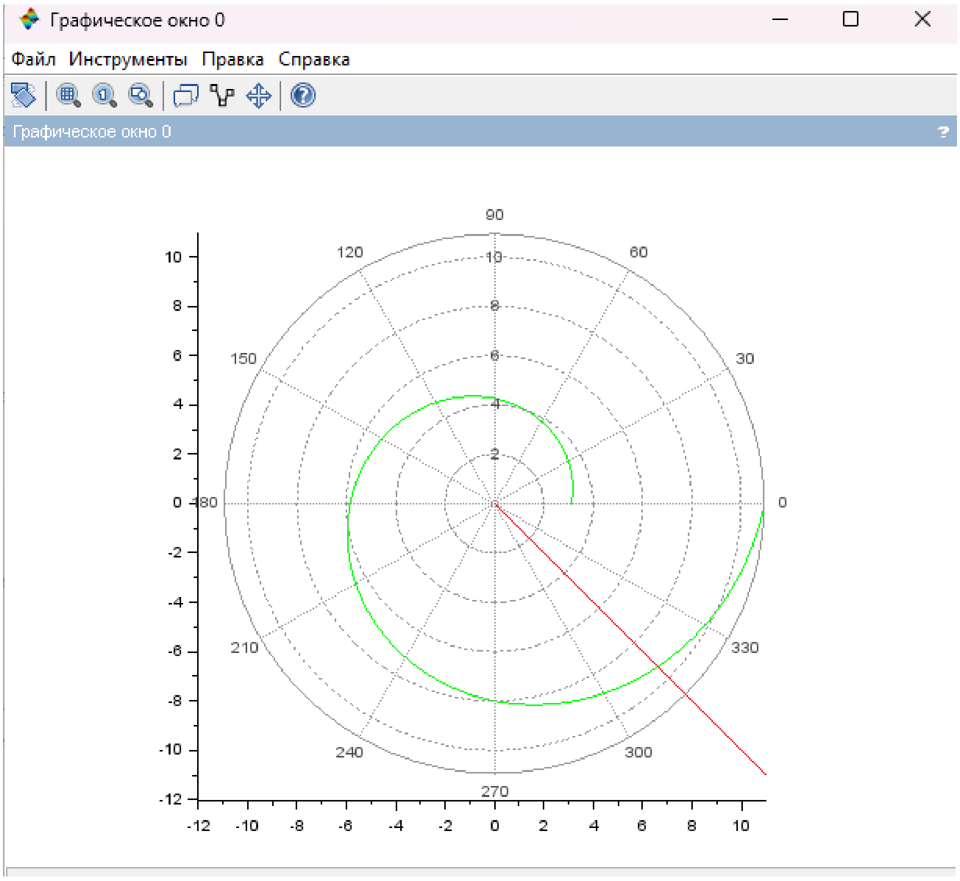



Рис. 7: Траектория катера и лодки (1 случай)

И точку пересечения катера и лодки (рис.8).

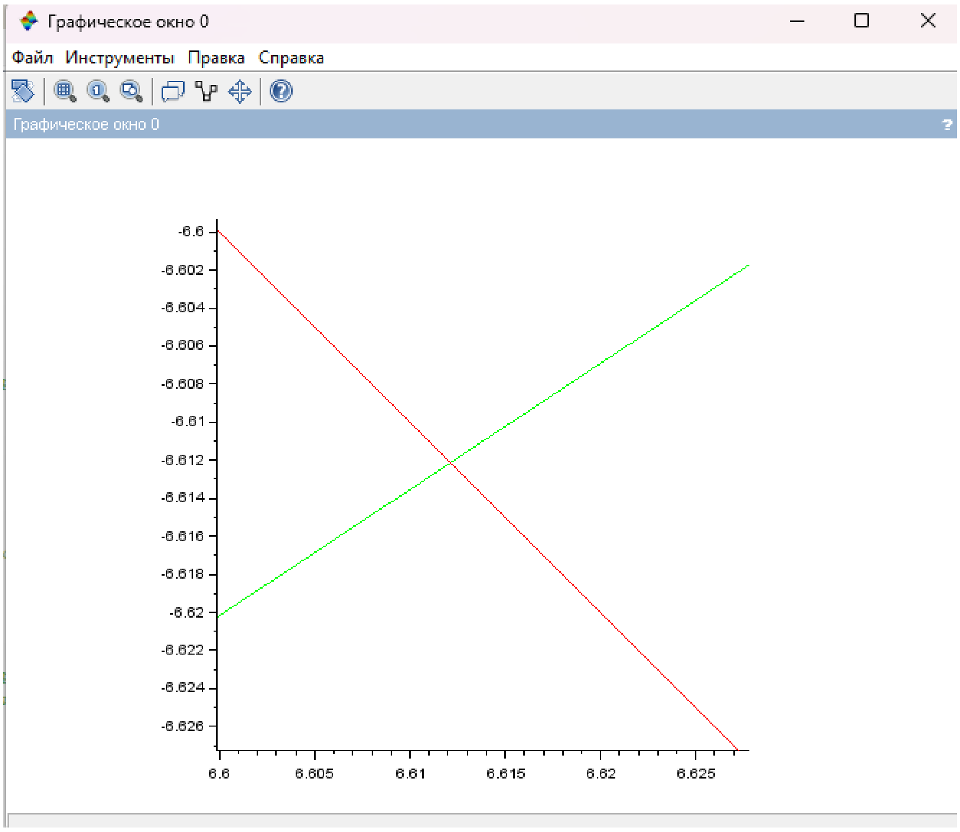



Рис. 8: Точка пересечения катера и лодки (1 случай)

Из рисунка видно, что точка пересечения $(6.6121; -6.6122)$

Определила траектории катера (зелёный цвет) и лодки (красный цвет) для 2 случая (рис.9).

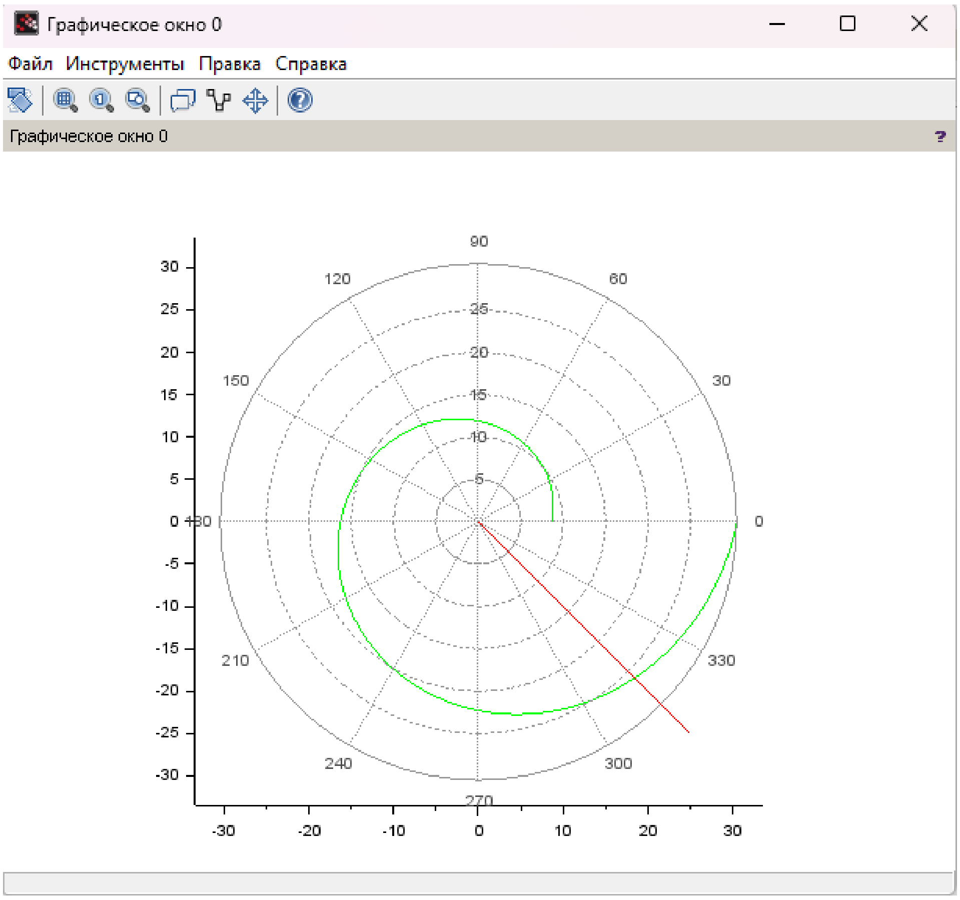



Рис. 9: Траектория катера и лодки (2 случай)

И точку пересечения катера и лодки (рис.10).

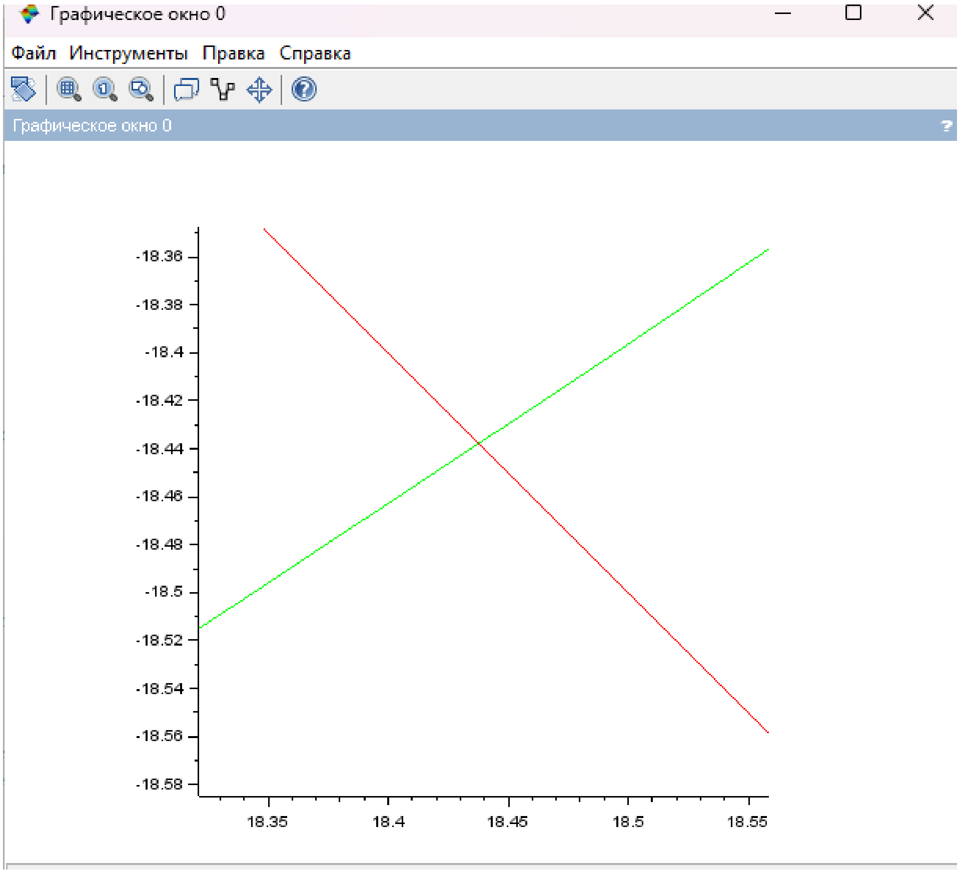



Рис. 10: Точка пересечения катера и лодки (2 случай)

Из рисунка видно, что точка пересечения $(18.4378; -18.4378)$

# Анализ полученных результатов

Так, были построены графики для обоих случаев. На них получилось отрисовать трактерию катера, траекторию лодки и получилось наглядно найти их точки пересечения. Задача о погоне решена.

# Вывод

Таким образом, в ходе ЛР№2 я рассмотрела пример построения математических моделей для выбора правильной стратегии при решении задач поиска. С помощью примера научилась решать задачи такого типа. Изучила основы языков программирования Julia и OpenModelica. Освоены библиотеки этих языков, которые используются для построения графиков и решения дифференциальных уравнений. Поскольку OpenModelica не работает с полярными координатами, она пока что не была использована в данной лабораторной работе.

# Список литературы. Библиография


1. Scilab documentation. [Электронный ресурс]. M. URL: Scilab documentation (Дата обращения: 15.02.2024).

2. Документация по Julia. [Электронный ресурс]. M. URL: Julia 1.10 Documentation (Дата обращения: 15.02.2024).

3. Документация по OpenModelica. [Электронный ресурс]. M. URL: openmodelica (Дата обращения: 15.02.2024).

4. Решение дифференциальных уравнений. [Электронный ресурс]. M. URL: wolframalpha (Дата обращения: 15.02.2024).