

Лабораторная работа №1

Задача о погоне

Роман Владимирович Иванов

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11

Список таблиц

Список иллюстраций

1 Цель работы

Построение математической модели для выбора правильной стратегии при решении задачи о погоне.

2 Задание

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

3 Выполнение лабораторной работы

1. Известны начальные данные задачи: лодка обнаруживается на расстоянии $s = 16.6$ км от катера, и скорость катера в 4.4 раза больше чем скорость лодки. Для того, чтобы описать уравнение движения катера необходимо, во-первых, выразить

$$x_1 = \frac{s}{n + 1},$$

где (x_1 - начальное расстояние между лодкой и катером для 1го случая)

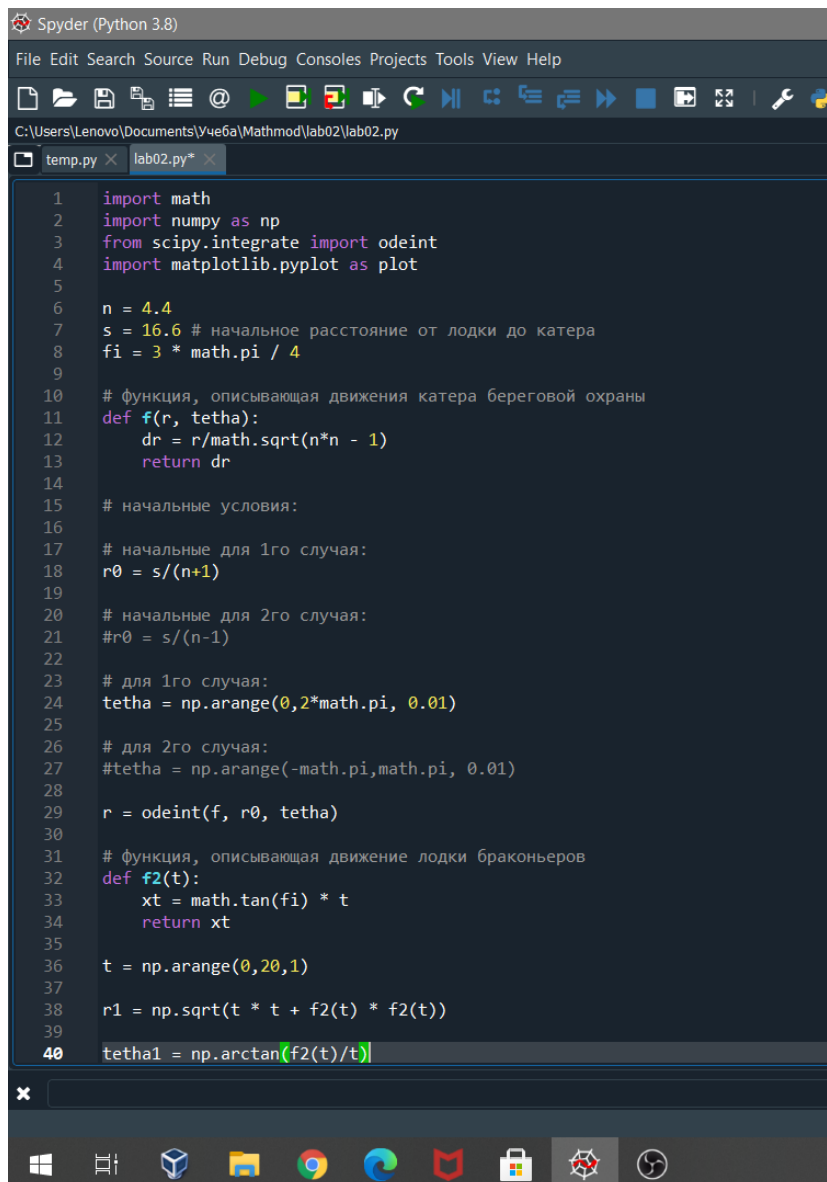
$$x_2 = \frac{s}{n - 1},$$

где (x_2 - начальное расстояние между лодкой и катером для 2го случая)

Во-вторых, нужно выразить дифференциальное уравнение в общем виде

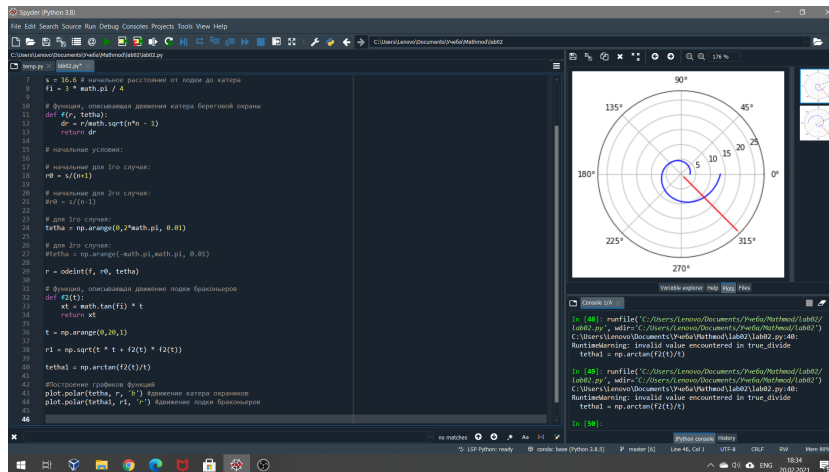
$$x_1 = \frac{dr}{d\theta} = \frac{r}{\sqrt{n^2 - 1}}$$

Ниже представлен код программы, в котором присутствуют два вышеописанных уравнения(рис. @fig:001)

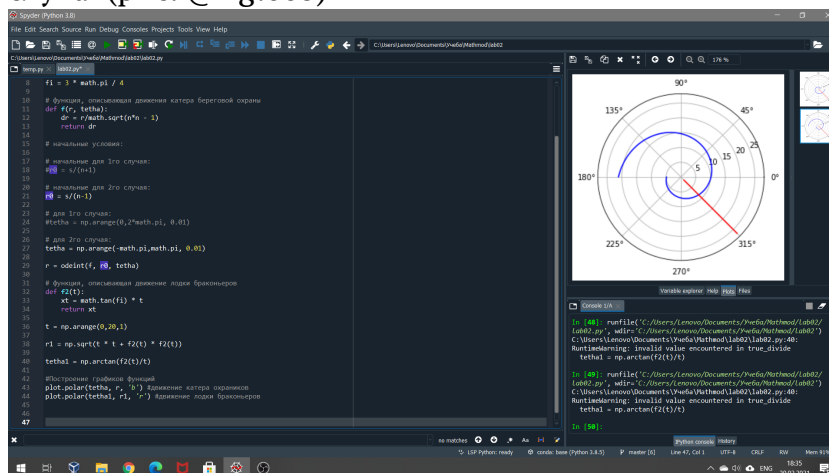


```
1 import math
2 import numpy as np
3 from scipy.integrate import odeint
4 import matplotlib.pyplot as plot
5
6 n = 4.4
7 s = 16.6 # начальное расстояние от лодки до катера
8 fi = 3 * math.pi / 4
9
10 # функция, описывающая движения катера береговой охраны
11 def f(r, theta):
12     dr = r/math.sqrt(n*n - 1)
13     return dr
14
15 # начальные условия:
16
17 # начальные для 1го случая:
18 r0 = s/(n+1)
19
20 # начальные для 2го случая:
21 #r0 = s/(n-1)
22
23 # для 1го случая:
24 theta = np.arange(0,2*math.pi, 0.01)
25
26 # для 2го случая:
27 #theta = np.arange(-math.pi,math.pi, 0.01)
28
29 r = odeint(f, r0, theta)
30
31 # функция, описывающая движение лодки браконьеров
32 def f2(t):
33     xt = math.tan(fi) * t
34     return xt
35
36 t = np.arange(0,20,1)
37
38 r1 = np.sqrt(t * t + f2(t) * f2(t))
39
40 theta1 = np.arctan(f2(t)/t)
```

2. Добавлю в код программы функцию, позволяющую строить в полярных координатах траектории движений катера и лодки для 1го случая(рис. @fig:002)

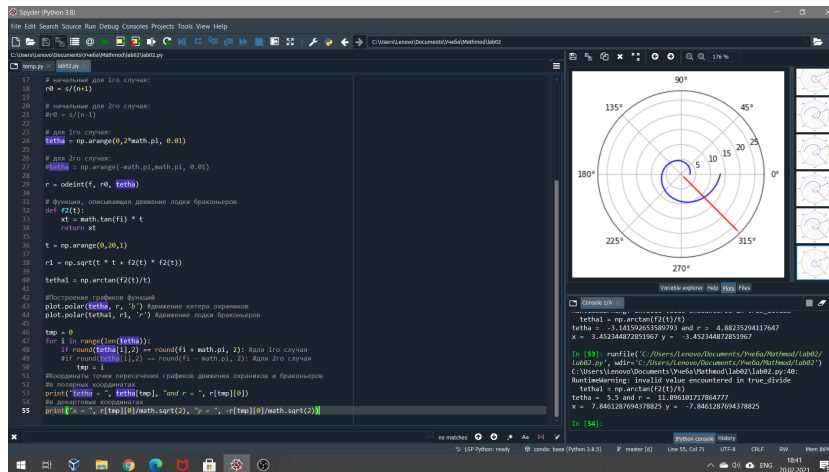


Теперь произведу изменения так, чтобы функция смогла построить в полярных координатах траектории движений катера и лодки для 2го случая(рис. @fig:003)

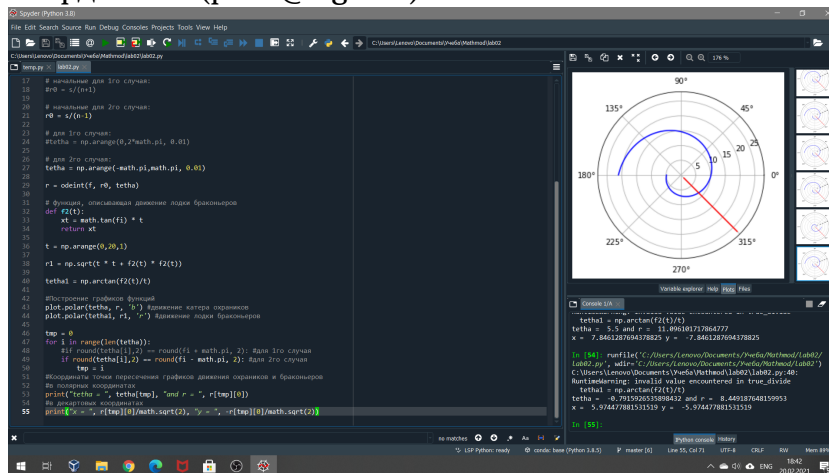


3. Найду точку пересечения траекторий катера и лодки для 1го случая. Точка пересечения данных графиков - точка, в которой радиусы и углы обоих функций совпадают (в полярных координатах).

Добавлю фрагмент к коду программе, позволяющий найти эту точку и выведу координаты этой точки в полярных и декартовых координатах(рис. @fig:004)



Теперь найду координаты 2ой точки пересечения в декартовых и полярных координатах(рис. @fig:005)



4 Выводы

Научился строить модель для выбора правильной стратегии при решении задачи о погоне.