

# **Отчёт по лабораторной работе №**

**Предмет: Математическое моделирование**

Манаева Варвара Евгеньевна, НФИбд-01-20. 1032201197

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Математическая модель . . . . .	8
4.2	Решение с помощью программ . . . . .	10
4.2.1	OpenModelica . . . . .	10
4.2.2	Julia . . . . .	10
<b>5</b>	<b>Выводы</b>	<b>18</b>
<b>6</b>	<b>Список литературы</b>	<b>19</b>

## Список иллюстраций

2.1	“Формула вычисления варианта” . . . . .	6
2.2	“Вывод формулы” . . . . .	6
4.1	“Первая траектория: катер смотрел в схожем направлении” . . .	14
4.2	“Вторая траектория: катер смотрел в противоположном направлении” . . . . .	15
4.3	“Первая траектория: катер смотрел в схожем направлении” . . .	16
4.4	“Вторая траектория: катер смотрел в противоположном направлении” . . . . .	17

## Список таблиц

# 1 Цель работы

Изучить основы языков программирования Julia и OpenModelica. Освоить некоторые библиотеки этих языков, которые используются для решения дифференциальных уравнений и построения графиков. На основании задачи лабораторной работы продемонстрировать полученные знания.

## 2 Задание лабораторной работы

Задания лабораторной работы разделены по вариантам. Мой вариант 28 (исходя из формулы  $N_{student} \bmod K_{ofvariants} + 1$ ) (рис. 2.1, 2.2). Этот же вариант будет использоваться для всех последующих лабораторных работ.

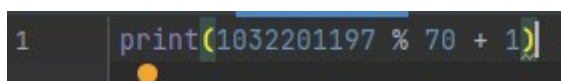


Рис. 2.1: “Формула вычисления варианта”



Рис. 2.2: “Вывод формулы”

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 15 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 4 раза больше скорости браконьерской лодки.

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 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.

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. По своим возможностям приближается к таким вычислительным средам как Matlab Simulink, Scilab xCos, имея при этом значительно более удобное представление системы уравнений исследуемого блока.

## 4 Выполнение лабораторной работы

### 4.1 Математическая модель

1. Примем за момент отсчета времени момент первого рассеивания тумана. Введем полярные координаты с центром в точке нахождения браконьеров и осью, проходящей через катер береговой охраны. Тогда начальные координаты катера  $(15; 0)$ . Обозначим скорость лодки  $v$ .
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 следующие уравнение. Пусть через время  $t$  катер и лодка окажутся на одном расстоянии  $x$  от полюса. За это время лодка пройдет  $x$ , а катер  $15 + x$  (или  $15 - x$ , в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как  $\frac{x}{v}$  или  $\frac{15-x}{4v}$  ( $\frac{15+x}{4v}$ ). Так как время должно быть одинаковым, эти величины тоже будут друг другу равны. Из этого получаем объединение из двух уравнений (двух из-за двух разных изначальных позиций катера относительно полюса):



$$\begin{cases} \frac{x}{v} = \frac{15-x}{4v} \\ \frac{x}{v} = \frac{15+x}{4v} \end{cases}$$

Из данных уравнений можно найти расстояние, после которого катер начнёт раскручиваться по спирали. Для данных уравнений решения будут следующими:  $x_1 = 3, x_2 = 5$ . Задачу будем решать для двух случаев. 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:  $v_r = \frac{dr}{dt} = v$  - радиальная скорость и  $v_\tau = r \frac{d\theta}{dt}$  - тангенциальная скорость.

$$v_\tau = \sqrt{15}v$$

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

$$\begin{cases} \frac{dr}{dt} = v \\ r \frac{d\theta}{dt} = \sqrt{15}v \end{cases}$$

с начальными условиями

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

или

$$\begin{cases} \theta_0 = \pi \\ r_0 = x_2 \end{cases}$$

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

$$\frac{dr}{d\theta} = \frac{r}{\sqrt{15}}$$

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

## 4.2 Решение с помощью программ

### 4.2.1 OpenModelica

К сожалению, OpenModelica не адаптирована к использованию полярных координат, поэтому адекватное отображение результатов данной задачи там невозможно.

### 4.2.2 Julia

#### 4.2.2.1 Программный код решения на Julia

Решить дифференциальное уравнение, расписанное в постановке задачи лабораторной работы, поможет библиотека DifferentialEquations. Итоговые изображения в полярных координатах будут строиться через библиотеку PyPlot.

Код первой программы ( $\theta_l = \frac{7\pi}{4}$ ):

```
using PyPlot;
using DifferentialEquations;
F(u, p, T) = u / sqrt(15)
const r_1 = 30/10
const r_2 = 50/10
const T = (0, 5pi)
const kater = 7pi/4
prob1 = ODEProblem(F, r_1, T)
prob2 = ODEProblem(F, r_2, (pi, 6pi))
```

```

sol1 = solve(prob1, abstol=1e-18, reltol=1e-18)
sol2 = solve(prob2, abstol=1e-18, reltol=1e-18);

R1 = [tu[1] for tu in sol1.u]
R2 = [tu[1] for tu in sol2.u]

Q1 = [tu[1] for tu in sol1.t]
Q2 = [tu[1] for tu in sol2.t]

intersection_r1 = 0/1
intersection_r2 = 0/1
for (i, q) in enumerate(Q1)
    if (round(q, digits=2) == round(kater, digits=2))
        global intersection_r1 = R1[i]
        break
    end
end
for (i, q) in enumerate(Q2)
    if (round(q, digits=2) == round(kater, digits=2))
        global intersection_r2 = R2[i]
        break
    end
end
clf()
polar(sol1.t, sol1.u + fill(r_1, 9106))
polar(fill(kater, 19), collect(0: 10: 180))
polar(kater, intersection_r1, "g.")
xlabel("theta")
title("Первая траектория")

```

```

savefig("kater1.jpg")
clf()
polar(sol2.t, sol2.u + fill(r_2, 10426))
polar(fill(kater, 31), collect(0: 10: 300))
polar(kater, intersection_r2, "g.")
xlabel("theta")
title("Вторая траектория")
savefig("kater2.jpg")
clf()

```

Код второй программы программы ( $\theta_l = \frac{5\pi}{4}$ ):

```

using PyPlot;
using DifferentialEquations;
F(u, p, T) = u / sqrt(15)
const r_1 = 30/10
const r_2 = 50/10
const T = (0, 5pi)
const kater = 5pi/4
prob1 = ODEProblem(F, r_1, T)
prob2 = ODEProblem(F, r_2, (pi, 6pi))
sol1 = solve(prob1, abstol=1e-18, reltol=1e-18)
sol2 = solve(prob2, abstol=1e-18, reltol=1e-18);

R1 = [tu[1] for tu in sol1.u]
R2 = [tu[1] for tu in sol2.u]

Q1 = [tu[1] for tu in sol1.t]
Q2 = [tu[1] for tu in sol2.t]

```

```

intersection_r1 = 0/1
intersection_r2 = 0/1
for (i, q) in enumerate(Q1)
    if (round(q, digits=2) == round(kater, digits=2))
        global intersection_r1 = R1[i]
        break
    end
end
for (i, q) in enumerate(Q2)
    if (round(q, digits=2) == round(kater, digits=2))
        global intersection_r2 = R2[i]
        break
    end
end
clf()
polar(sol1.t, sol1.u + fill(r_1, 9106))
polar(fill(kater, 19), collect(0: 10: 180))
polar(kater, intersection_r1, "g.")
xlabel("theta")
title("Первая траектория")
savefig("kater3.jpg")
clf()
polar(sol2.t, sol2.u + fill(r_2, 10426))
polar(fill(kater, 31), collect(0: 10: 300))
polar(kater, intersection_r2, "g.")
xlabel("theta")
title("Вторая траектория")
savefig("kater4.jpg")
clf()

```

#### 4.2.2.2 Результаты работы кода на Julia

На рис. 4.1 и 4.2 изображены итоговые графики траектории движения катера и лодки для случая 1 ( $\theta_l = \frac{7\pi}{4}$ )

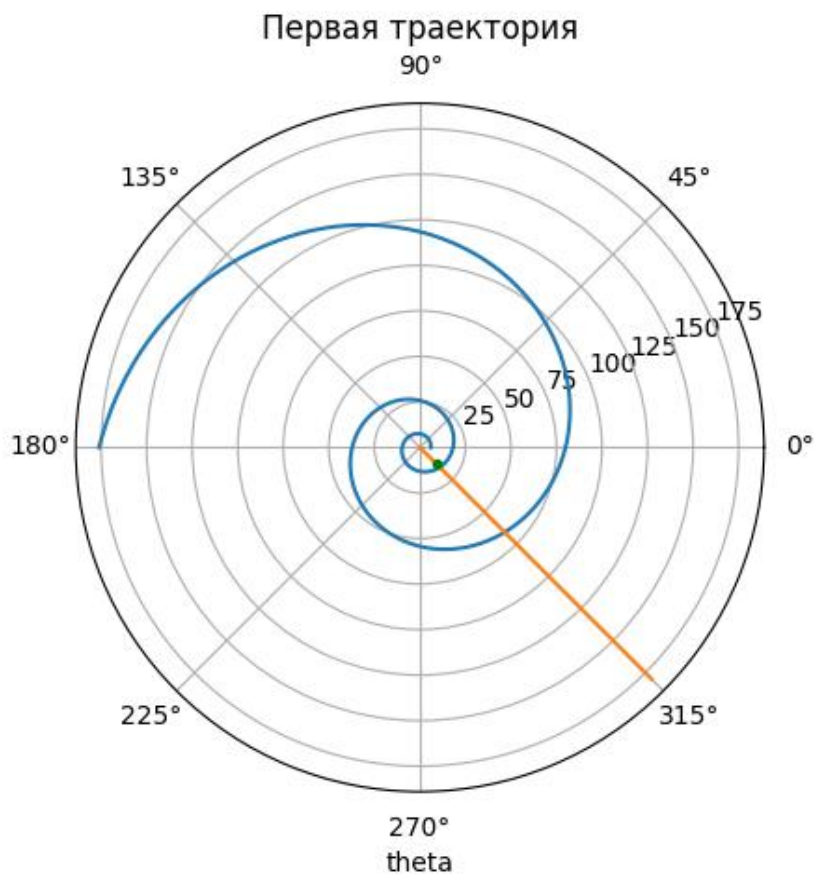


Рис. 4.1: “Первая траектория: катер смотрел в схожем направлении”

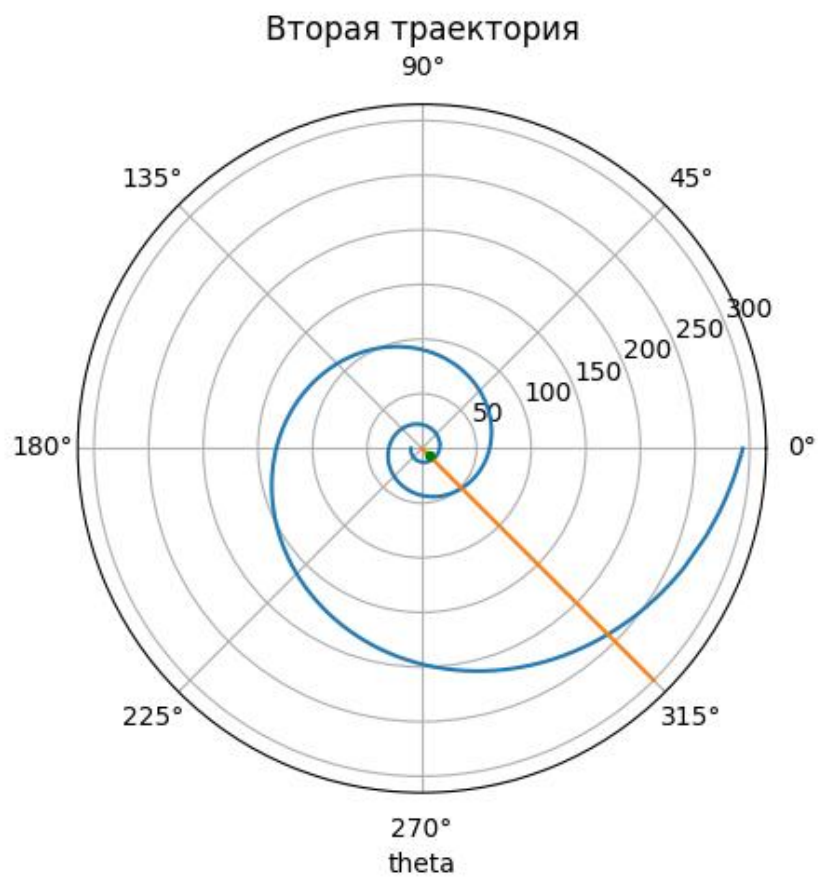


Рис. 4.2: “Вторая траектория: катер смотрел в противоположном направлении”

На рис. 4.3 и 4.4 изображены итоговые графики траектории движения катера и лодки для случая 2 ( $\theta_l = \frac{5\pi}{4}$ )

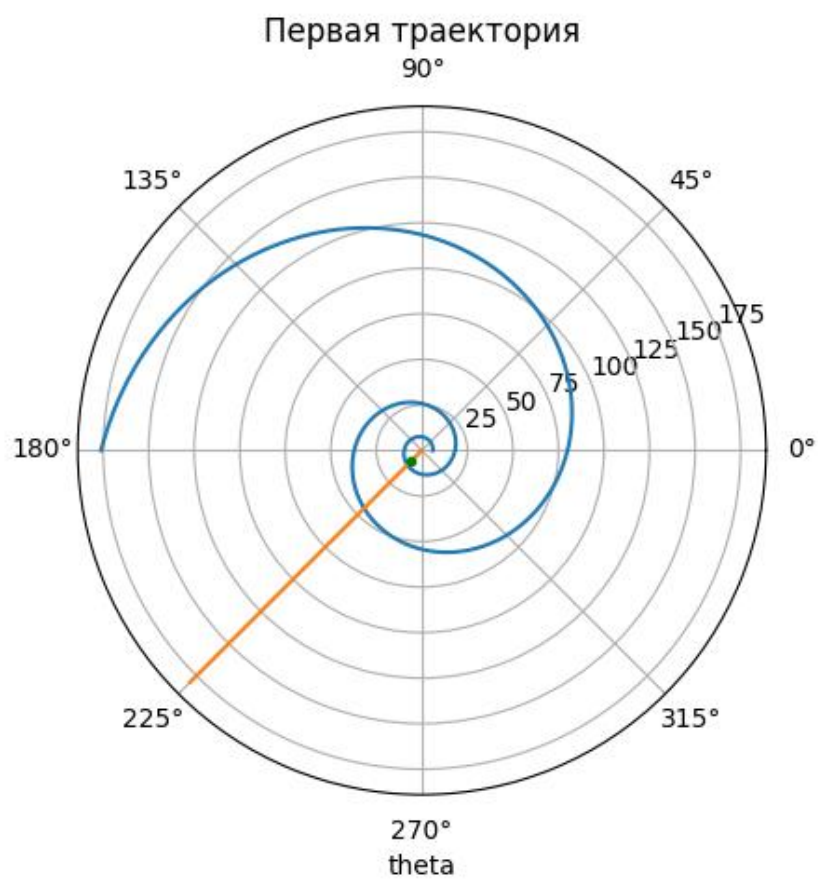


Рис. 4.3: “Первая траектория: катер смотрел в схожем направлении”



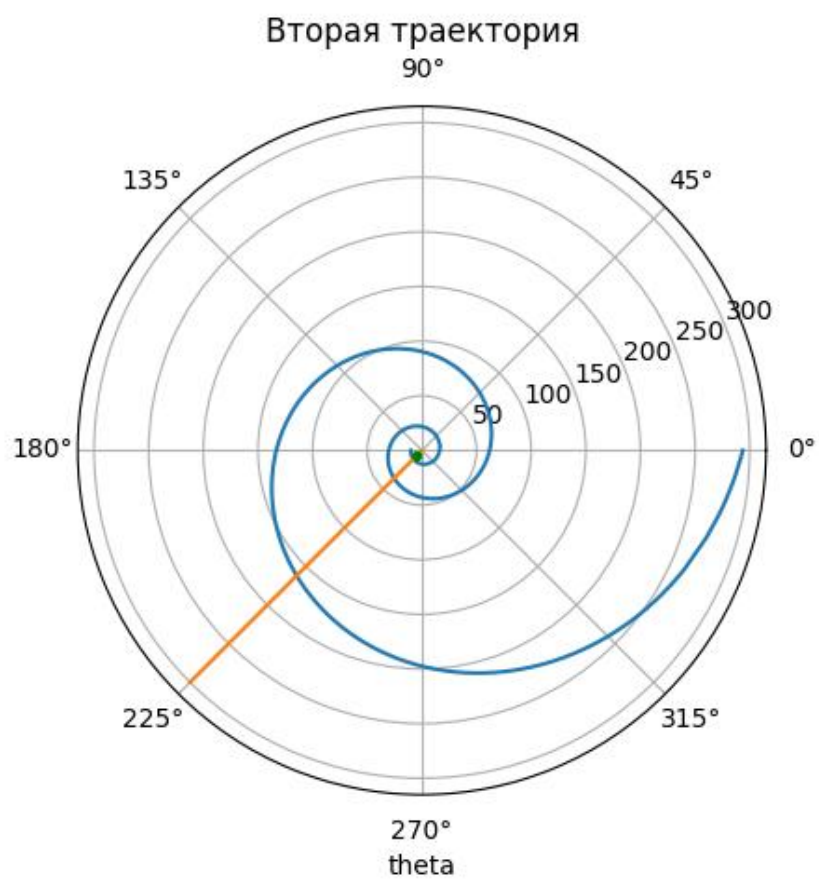


Рис. 4.4: “Вторая траектория: катер смотрел в противоположном направлении”

## 5 Выводы

Были изучены основы Julia и OpenModelica. Поскольку OpenModelica не работает с полярными координатами, она не была использована в данной лабораторной работе.

Для данной задачи на Julia был написан код, для кода были получены графики, отображающие движение катера и лодки.

Были записаны скринкасты лабораторной работы и презентации лабораторной работы.

## 6 Список литературы

1. Пример решения лабораторной работы
2. Варианты заданий лабораторной работы
3. Plotting polar curves in Python - [GeeksForGeeks](#)
4. Документация Julia 1.8
5. Документация модуля Julia Differential Equations
6. Документация модуля Julia PyPlot