

# **Отчет по лабораторной работе №2**

**Дисциплина: Математическое моделирование**

Выполнила: Боровикова Карина Владимировна

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Теоретическое введение</b>	<b>7</b>
<b>4 Выполнение лабораторной работы</b>	<b>8</b>
<b>5 Выводы</b>	<b>17</b>
<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

4.1	Вычисления, часть 1 . . . . .	9
4.2	Вычисления, часть 2 . . . . .	10
4.3	Вычисления, часть 3 . . . . .	11
4.4	Вычисления, часть 4 . . . . .	12
4.5	Результат запуска кода на Julia для первого случая . . . . .	15
4.6	Результат запуска кода на Julia для второго случая . . . . .	16

# **Список таблиц**

# 1 Цель работы

**Цель работы:** Изучить языки Julia и OpenModelica в части моделирования решения реальных математических задач на основе 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 Теоретическое введение

Познакомимся с такими языками программирования математических вычислений как Julia и OpenModelica для выполнения лабораторной работы.

**Julia** — это высокоуровневый динамический язык программирования, разработанный для того, чтобы предоставить пользователям скорость C / C ++, оставаясь при этом таким же простым в использовании, как Python. С документацией языка можно ознакомиться на официальном сайте в разделе документации [1].

**OpenModelica** — это среда моделирования на основе Modelica 1 с открытым исходным кодом, предназначенная для промышленного и академического использования. Его долгосрочное развитие поддерживается некоммерческой организацией – Консорциумом Open Source Modelica Consortium (OSMC). Open Source Modelica Consortium является совместным проектом RISE SICS East AB и Линчёпингского университета. [2].

## 4 Выполнение лабораторной работы

1. Будем выполнять вариант 18 задачи про погоню охотников и браконьеров:

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 7.7 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 3.3 раза больше скорости браконьерской лодки.

Браконьеры и охотники видят друг друга на расстоянии  $a = 7.7$  км до наступления тумана.

Браконьеры двигаются по прямой в то время, как охотники движутся по траектории, близкой к окружности, однако увеличивают радиус, таким образом описывая спираль, чтобы нагнать браконьеров в любом случае, поскольку вторые движутся в неизвестном направлении.

3. Случай 1.

Чтобы поймать браконьеров, охотники должны преодолеть расстояние ( $a - r_0$ ) по прямой, а далее начать движение по спирали.

Тогда существует  $t$  - время, которое тратит лодка браконьеров на преодоление расстояния  $r_0$ , а также которое тратит катер охотников на преодоление расстояния ( $a - r_0$ ). Методом преобразований получим точку, с которой стартует катер охотников, начиная движение по спирали для первого случая.

Скорость катера охотников можно разложить на скорость тангенсальную и на радиальную  $V_r$ .

В таком случае  $V_r$  должна быть равной  $V_{\text{Браконьеров}}$ , чтобы катер охотников всегда находился на одном расстоянии от О, что и лодка браконьеров.

За время  $\Delta t$  охотники проходят свою траекторию по радиусу  $\Delta r$  - расстояние по прямой. Существует угол  $\Delta\Theta$  - угол между радиальной и тангенсальной скоростями охотников.

Из теоремы Пифагора и преобразований получим уравнение с разделяющимися переменными. Решив его и подставив данные в задании значения  $\Theta_0$  и  $r_0$ .

Далее прикрепляю свое решение, где я вычисляю все значения (рис. 4.1 - 4.4).

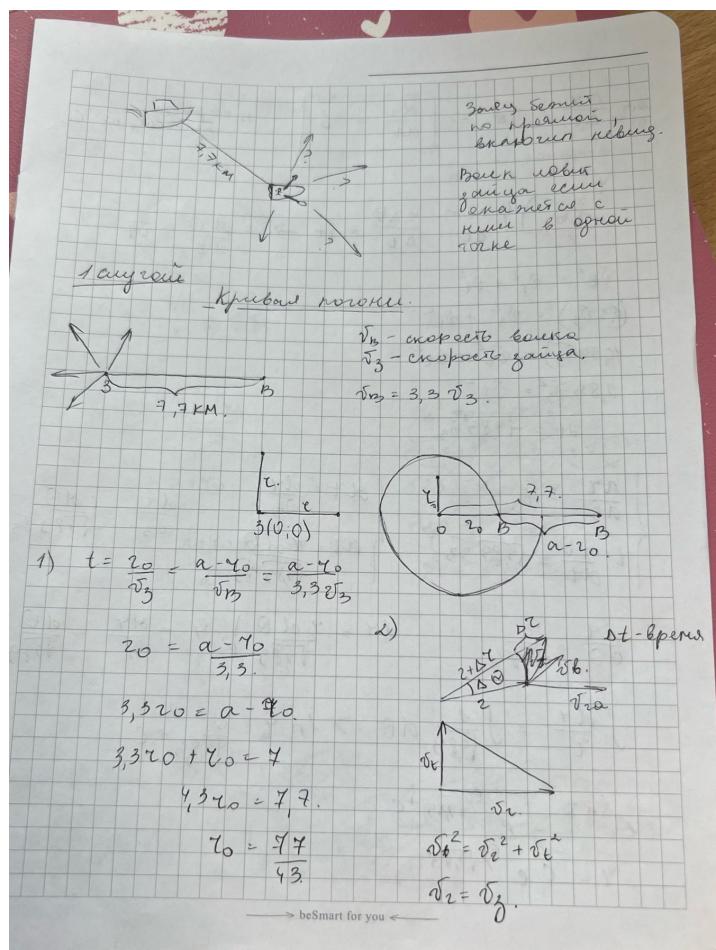


Рис. 4.1: Вычисления, часть 1

$$\lim_{\Delta t \rightarrow 0} \frac{\Delta \gamma}{\Delta t} = v_B \Rightarrow \left[ \frac{d\gamma}{dt} = v_B \right] \quad v_B = \text{расстояние} \rightarrow 0.$$

$\begin{array}{c} \gamma \\ \Delta \Theta \\ \Delta t \end{array}$

$$v_B^2 = v_x^2 + v_t^2$$

$$(3,3 v_3)^2 = v_3^2 + v_t^2$$

$$19,89 v_3^2 = v_3^2 + v_t^2$$

$$19,89 v_3^2 = v_t^2$$

$$v_t = \sqrt{19,89} v_3$$

$$\left\{ \begin{array}{l} \frac{d\gamma}{dt} = v_3 \\ \gamma \frac{d\Theta}{dt} = \sqrt{19,89} v_3 \end{array} \right. \quad \left\{ \begin{array}{l} dt = \frac{d\gamma}{v_3} \\ dt = \frac{\gamma d\Theta}{\sqrt{19,89} v_3} \end{array} \right. \quad \frac{d\gamma}{\sqrt{19,89} v_3} = \frac{\gamma d\Theta}{\sqrt{19,89} v_3}$$

$$\frac{d\gamma}{d\Theta} = \frac{\gamma}{\sqrt{19,89}} \quad d\gamma = \frac{\gamma d\Theta}{\sqrt{19,89}} \quad \frac{d\gamma}{\gamma} = \frac{d\Theta}{\sqrt{19,89}}$$

$$\int \frac{d\gamma}{\gamma} = \int \frac{1}{\sqrt{19,89}} d\Theta \Rightarrow \ln \gamma = \frac{\Theta}{\sqrt{19,89}} + C$$

$$\gamma = e^{\frac{\Theta}{\sqrt{19,89}} + C}$$

$$\gamma = C e^{\frac{\Theta}{\sqrt{19,89}}}$$

→ designed by beSmart ←

Рис. 4.2: Вычисления, часть 2

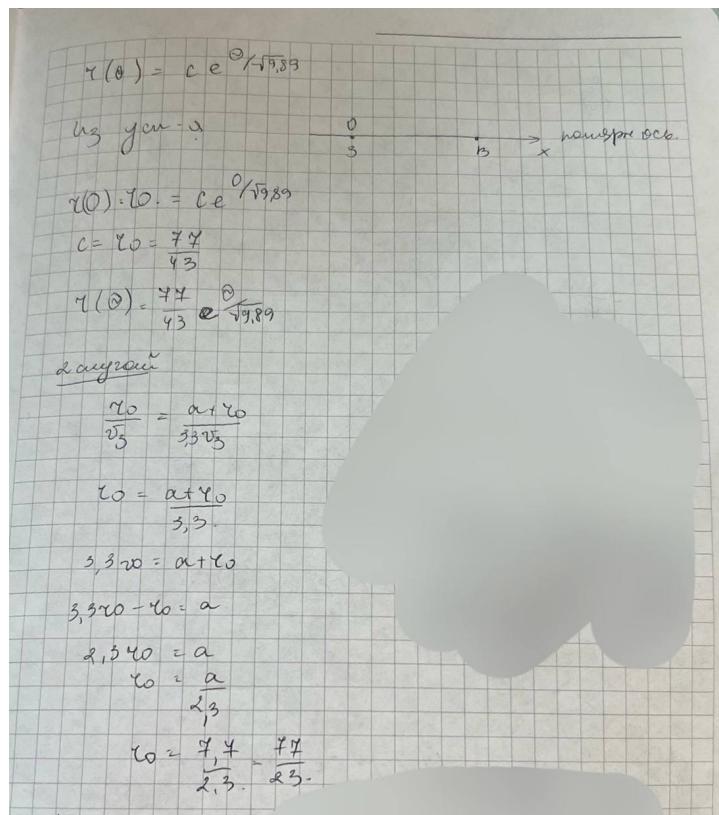


Рис. 4.3: Вычисления, часть 3

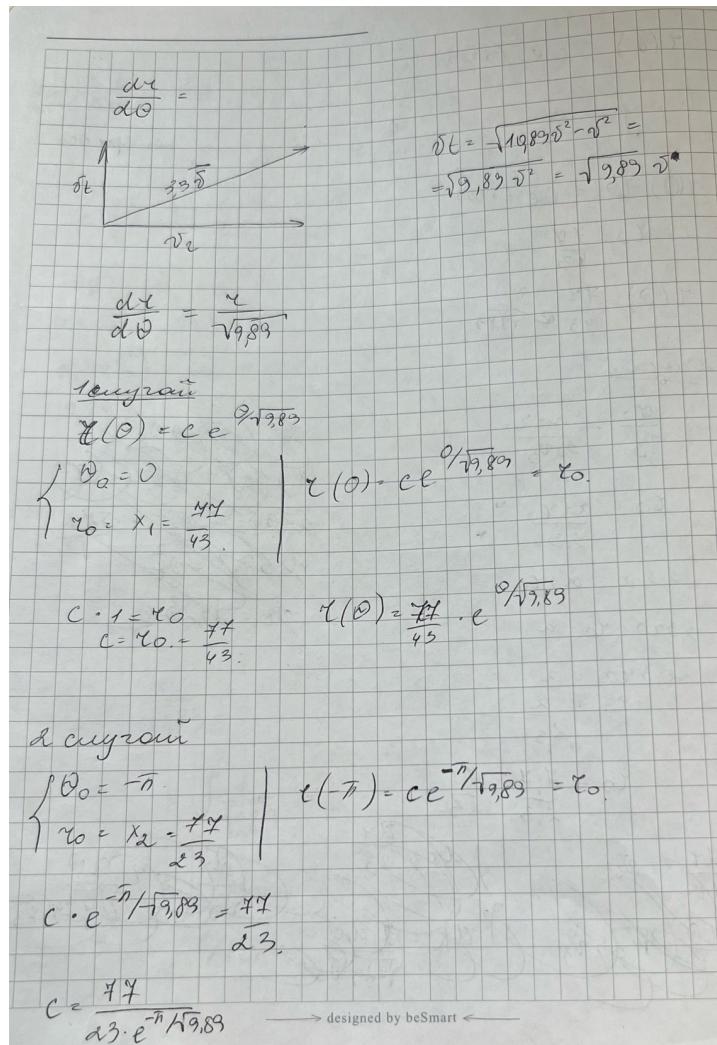


Рис. 4.4: Вычисления, часть 4

4. Аналогичным способом выполняем вычисление для второго случая.

5. Далее нам необходимо написать программу для моделирования этих ситуаций на языке Julia [1]. Ниже представлен код:

```
using Plots
```

"Расстояние на момент начала движения по спирали, случай 1"

```
const r_01 = 77/43
```

```

"Расстояние на момент начала движения по спирали, случай 2"
const r_02 = -77/23

"Расстояние вначале"
const a = 7.7

"C для случая 1"
const C_01 = 77/43

"C для случая 2"
const C_02 = 77/(23*exp(-π/sqrt(9.89)))

function r(ℳ)
    return C_02*exp(ℳ / sqrt(9.89))
end

ℳ = range(-π, 2π, 1000)

R = r.(ℳ)

plt = plot(
    proj = :polar,
    aspect_ratio=:equal,
    dpi=300,
    title="Задача преследования",
    legend=true)

```

```
plot!(  
    plt,  
    x,  
    R,  
    xlabel="x",  
    ylabel="r(t)",  
    label="Траектория катера",  
    color=:green)
```

```
plot!(  
    plt,  
    [0.0,0.0],  
    [r_02,7.7],  
    color=:green,  
    label="")
```

```
plot!(  
    plt,  
    [0.0,x[300]],  
    [0,20],  
    xlabel="x",  
    ylabel="r(t)",  
    label="Траектория лодки с бандитами",  
    color=:red)
```

```
plot!(  
    [x[300]],
```

```

[R[300]],
seriesType=:scatter,
label="Точка встречи"
)

savefig(plt, "lab02.png")

```

6. После этого запускаем код через терминал для первого случая, для этого вводим

```
julia lab02.jl
```

Где lab02.jl - название запускаемого файла, после меняем его и аналогично запускаем для второго случая (рис. 4.5 - 4.6).



Рис. 4.5: Результат запуска кода на Julia для первого случая

### Задача преследования



Рис. 4.6: Результат запуска кода на Julia для второго случая

## **5 Выводы**

Изучила языки Julia и OpenModelica в части моделирования решения реальных математических задач на основе задачи о погоне

## **Список литературы**

1. Julia - programming langiage [Электронный ресурс]. 2023. URL: <https://julialang.org/>.
2. OpenModelica environment - Introduction [Электронный ресурс]. 2023. URL: <https://openmodelica.org/>.