

Отчет по лабораторной работе №2

Дисциплина: Математическое моделирование

Выполнила: Губина Ольга Вячеславовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	20
	Список литературы	21

Список иллюстраций

4.1	Установка Julia	9
4.2	Проверка работы Julia	9
4.3	Установка OpenModelica	10
4.4	Разница случаев	13
4.5	График первого случая задачи	19
4.6	График второго случая задачи	19

Список таблиц

1 Цель работы

Цель работы: Изучить языки Julia и OpenModelica, освоить моделирование реальных математических задач с их помощью.

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

В данной работе нам предстоит познакомиться с такими языками программирования мат. вычислений как Julia и OpenModelica.

Julia — высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений.[1] Эффективен также и для написания программ общего назначения. С документацией языка можно ознакомиться на официальном сайте в разделе документации.[2]

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.

OpenModelica используется в академической среде и на производстве. В промышленности используется в области оптимизации энергоснабжения, автомобилестроении и водоочистке.

Включает блоки: - механики - электрики - электроники - электродвигатели - гидравлики - термодинамики - элементы управления и т. д.

По своим возможностям приближается к таким вычислительным средам как Matlab Simulink, Scilab xCos, имея при этом значительно более удобное представление системы уравнений исследуемого блока (фактически без существенного преобразования и без сведения к форме Коши, остаётся лишь задать начальные

условия и записать уравнения в скоростях).[3]

4 Выполнение лабораторной работы

1. Первое, что необходимо было сделать, это установить недостающие пакеты. Таким образом была произведена установка Julia через менеджер пакетов choco (рис. [4.1]) и OpenModelica через официальный установщик (рис. [4.3]).

```
PS C:\WINDOWS\system32> choco install julia
Chocolatey v1.8.0
Installing the following packages:
julia
By installing, you accept licenses for the packages.
Progress: Downloading Julia 1.8.5... 100%
Julia v1.8.5 [Approved]
Julia package files install completed. Performing other installation steps.
The package Julia wants to run 'chocolateysinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable --n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y
Installing 64-bit Julia...
Julia has been installed.
Julia installed to 'C:\Users\olga\AppData\Local\Programs\Julia-1.8.5\bin\julia.exe'
Added C:\ProgramData\chocolatey\bin\julia.exe shim pointed to 'C:\Users\olga\AppData\Local\Programs\Julia-1.8.5\bin\julia.exe'.
Julia can be automatically uninstalled.
The install of Julia was successful.
Software installed to 'C:\Users\olga\AppData\Local\Programs\Julia-1.8.5\'
Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

Рис. 4.1: Установка Julia

```
PS C:\WINDOWS\system32> julia

Documentation: https://docs.julialang.org
Type "?" for help, "]" for Pkg help.

Version 1.8.5 (2023-01-08)
Official https://julialang.org/ release

julia> 1032201737 % 70
7
julia> exit()
PS C:\WINDOWS\system32>
```

Рис. 4.2: Проверка работы Julia

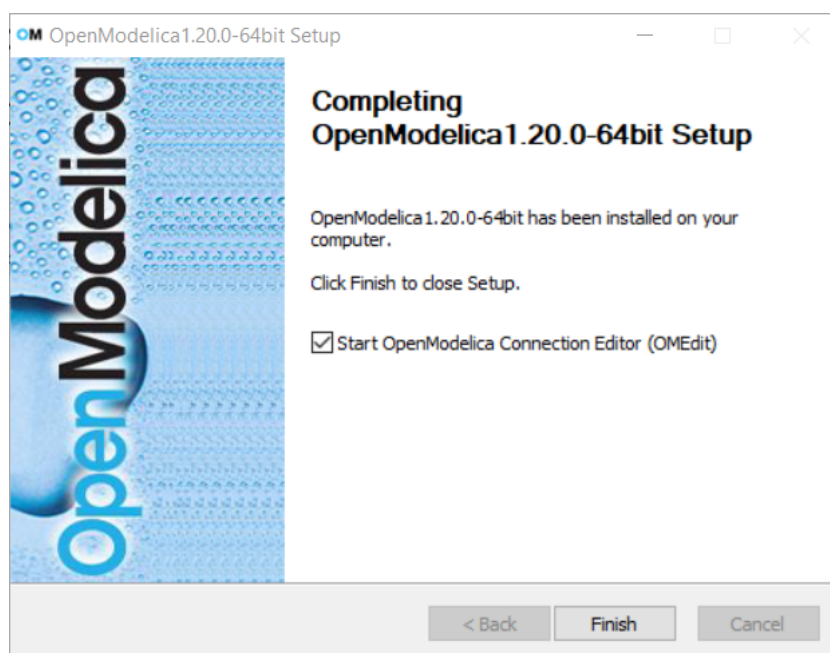


Рис. 4.3: Установка OpenModelica

2. Далее будем выполнять вариант 7 предложенных вариантов 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 6,4 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 2,4 раза больше скорости браконьерской лодки: $V_O = 2.4V_B$.

Браконьеры и охотники видят друг друга на расстоянии $a = 6.4$ до наступления тумана.

Браконьеры двигаются по прямой в то время, как охотники движутся по спирали, чтобы нагнать браконьеров в любом случае, поскольку вторые движутся в неизвестном направлении.

3. Случай 1.

Для успешного перехвата охотники должны преодолеть расстояние $a - r_0$ по прямой и только после этого начать движение по спирали.

Тогда существует время t - время, которое тратит лодка браконьеров на преодоление расстояния r_0 , более того это время, которое тратит катер охотников на преодоление расстояния $a - r_0$ (см. формулу [4.1]).

$$t = r_0/V_B = (a - r_0)/V_O = (a - r_0)/2.4V_B \quad (4.1)$$

Из формулы [4.1] следует, что [4.2]:

$$r_0/V_B = (a - r_0)/2.4V_B \quad (4.2)$$

Тогда, методом преобразований получим начальную точку, с которой стартует катер охотников, начиная движение по спирали для первого случая (см. формулу [4.3]).

$$r_0 = a/3.4 \quad (4.3)$$

После преодоления охотниками расстояния $a - r_0$ и по прошествии времени t браконьеры и охотники находятся на равном расстоянии от точки начала координат - точка начала движения браконьерской лодки. Теперь охотники могут начать движение по траектории, близкой к окружности, однако ее радиус постоянно увеличивается на Δr за время Δt .

Тогда: скорость катера охотников можно разложить на скорость тангенсальную V_t (которая двигает охотников в сторону за браконьерами, движение по касательной) и на радиальную V_r (которая двигает их от точки начала на то же расстояние, на которое движутся браконьеры Δr за время Δt).

В таком случае $V_r = V_B$, чтобы катер охотников всегда находился на одном расстоянии от , что и лодка браконьеров.

За время Δt охотники проходят свою траекторию по радиусу Δr - расстояние по прямой (формула [4.4])

$$\lim(\Delta t \rightarrow 0) \Delta r / \Delta t = V_O \quad (4.4)$$

Из этого следует (формула [4.5]):

$$dr/dt = V_r \quad (4.5)$$

Существует угол $\Delta\theta$ - угол между радиальной и тангенсальной скоростями охотников. Тогда (см. формулу [4.6]):

$$\lim(\Delta t \rightarrow 0)r\Delta\theta/\Delta t = V_t \quad (4.6)$$

Из формулы [4.6] следует, что (формула [4.7]):

$$r * d\theta/dt = V_t \quad (4.7)$$

Из теоремы Пифагора (формула [4.8]):

$$V_O^2 = V_r^2 + V_t^2 \rightarrow (2.4V_B)^2 = V_B^2 + V_t^2 \rightarrow 4.76V_B^2 = V_t^2 \quad (4.8)$$

В таком случае (формула [4.9]):

$$V_t = \sqrt{(4.76)V_B} \quad (4.9)$$

Тогда из [4.9] следует ([4.10]):

$$dr/dt = V_B \quad (4.10)$$

и [4.11]

$$r * d\theta/dt = \sqrt{(4.76)V_B} \quad (4.11)$$

Из [4.10] и [4.11] получим уравнение с разделяющимися переменными (см. формула [4.12]).

$$dr/r = d\theta/\sqrt{(4.76)} \quad (4.12)$$

Отсюда получается (формула [4.13]):

$$r(\theta) = Ce^{\frac{\theta}{\sqrt{4.76}}} \quad (4.13)$$

Из условия [4.13]:

$$C = r_0 = a/3.4 \quad (4.14)$$

Итог (формула [4.15]):

$$r(\theta) = \frac{a}{3.4} e^{\frac{\theta}{\sqrt{4.76}}} \quad (4.15)$$

4. Случай 2

Во втором случае у нас меняется расстояние, которое проходит катер охотников прежде чем начать движение по спирали, он идет в другую сторону и проходит расстояние равное $a + r_0$ - рис. [4.4].

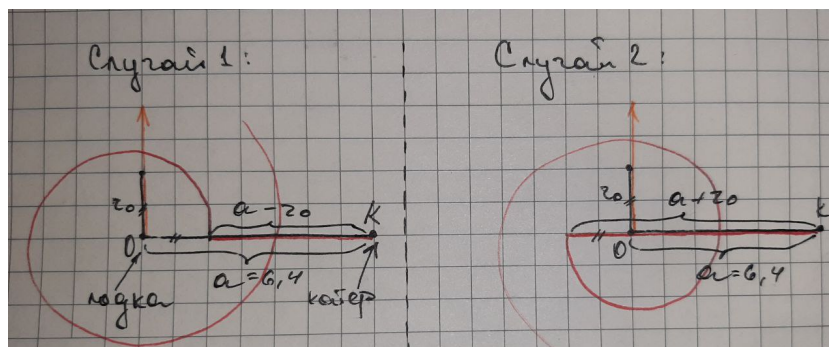


Рис. 4.4: Разница случаев

Браконьеры и охотники преодолевают расстояния равные r_0 и $a + r_0$ соответственно за равное время (формула [4.16]):

$$r_0/V_B = (a + r_0)/2.4V_B \quad (4.16)$$

Тогда, методом преобразований получим начальную точку, с которой стартует катер охотников, начиная движение по спирали для второго случая (см. формулу [4.17]).

$$r_0 = a/1.4 \quad (4.17)$$

По аналогии с первым случаем имеем скорость тангенсальную V_t и радиальную V_r . В таком случае $V_r = V_B$, чтобы катер охотников всегда находился на одном расстоянии от O , что и лодка браконьеров.

$$r(\theta) = Ce^{\frac{\theta}{\sqrt{4.76}}} \quad (4.18)$$

Исходя из формул [4.17]-[4.18] по аналогии с первым случаем:

$$Ce^{\frac{\theta}{\sqrt{4.76}}} = \frac{a}{1.4} \quad (4.19)$$

А значит для второго случая (см. формулу [4.20]):

$$C = \frac{a}{1.4e^{\frac{\theta}{\sqrt{4.76}}}} \quad (4.20)$$

5. Теперь напишем программу на Julia, которая будет отрисовывать оба случая в графиках полярных координат, выделяя точку пересечения:

```
using Plots
```

```
"Условие задачи:"
```

```
# Задаем расстояние, на котором изначально обнаруживается лодка браконьеров
```

```
const a = 6.4
```

```
# Во сколько раз катер быстрее лодки
```

```
const speed_difference = 2.4
```

```
"Случай 1:"
```

```

const r_01 = a/3.4 # Начальная точка, с которой стартует катер охотников,
# начиная движение по спирали из формулы 4.3 отчета
const C_01 = a/3.4 # Из формулы отчета 4.13

theta_1 = range(0, 2pi, 1000) # массив углов отклонения - целый оборот

function r1(theta_1)
    return C_01*exp(theta_1/sqrt(4.76))
end

R_1 = r1.(theta_1)

"Случай 2"
const r_02 = -a/1.4 # минус потому что в обратную сторону
const C_02 = a/(1.4*exp(-pi/sqrt(4.76)))

theta_2 = range(-pi, pi, 1000)

function r2(theta_2)
    return C_02*exp(theta_2 / sqrt(4.76))
end

R_2 = r2.(theta_2)

plt_1 = plot(
    proj = :polar,
    aspect_ratio=:equal,
    dpi=300,
    title="задача о погоне: случай 1",

```

```

        legend=true
    )

    plot!(
        plt_1,
        [0.0,0.0],
        [r_01,6.4],
        color=:red,
        label="Катер охотников"
    )

    plot!(
        plt_1,
        theta_1,
        R_1,
        xlabel="Theta",
        ylabel="r(t)",
        color=:red,
        label=""
    )

    plot!(
        plt_1,
        [0.0,theta_1[300]],
        [0,20],
        xlabel="Theta",
        ylabel="r(t)",
        color=:blue,
        label="Лодка браконьеров"
    )

```



```
)
```

```
scatter!(  
    plt_1,  
    [theta_1[300]],  
    [R_1[300]],  
    label="Точка встречи",  
    ms=1.5  
)
```

```
savefig(plt_1, "lab02case01.png")
```

```
plt_2 = plot(  
    proj = :polar,  
    aspect_ratio=:equal,  
    dpi=300,  
    title="Задача о преследовании: случай 2",  
    legend=true)
```

```
plot!(  
    plt_2,  
    theta_2,  
    R_2,  
    xlabel="ϕ",  
    ylabel="r(t)",  
    label="Катер охотников",  
    color=:red)
```

```
plot!(
```

```

plt_2,
[0.0,0.0],
[r_02,6.4],
color=:red,
label="")

plot!(
    plt_2,
    [0.0,theta_2[300]],
    [0,20],
    xlabel="ϕ",
    ylabel="r(t)",
    label="Лодка браконьеров",
    color=:blue)

scatter!(
    plt_2,
    [theta_2[300]],
    [R_2[300]],
    label="Точка встречи",
    ms=1.5
)

savefig(plt_2, "lab02case02.png")

```

6. В качестве результата имеем следующие картинки, на которых все предельно ясно подписано (рис. [4.6]).

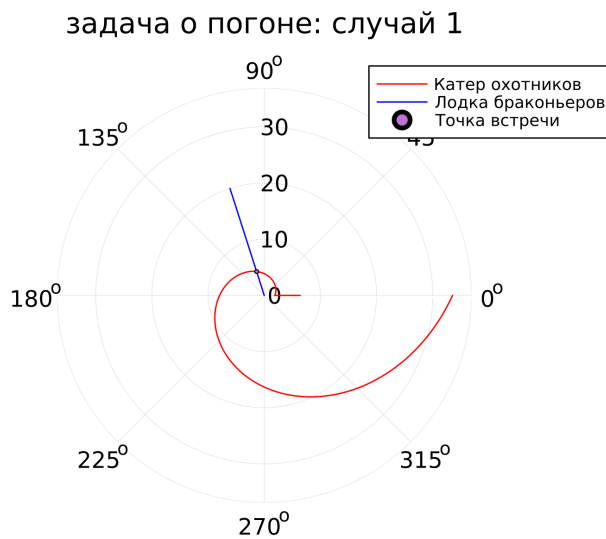


Рис. 4.5: График первого случая задачи



Рис. 4.6: График второго случая задачи

7. OpenModelica не предназначена для решения подобного рода задач, только если с использованием нестандартных методов, поэтому данная задача представлена только на одном языке программирования.

5 Выводы

Изучила языки Julia и OpenModelica, освоила моделирование реальных математических задач с помощью Julia.

Список литературы

1. Julia (язык программирования) [Электронный ресурс]. 2023. URL: [https://ru.wikipedia.org/wiki/Julia_\(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\)#%3A%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9%20%D0%B2%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9%20%D1%81%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9%20%D1%8F%D0%B7%D1%8B%D0%BA,%D0%B4%D0%BB%D1%8F%20%D0%BD%D0%B0%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%20%D0%BE%D0%B1%D1%89%D0%B5%D0%B3%D0%BE%20%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F](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)#%3A%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9%20%D0%B2%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9%20%D1%81%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9%20%D1%8F%D0%B7%D1%8B%D0%BA,%D0%B4%D0%BB%D1%8F%20%D0%BD%D0%B0%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%20%D0%BE%D0%B1%D1%89%D0%B5%D0%B3%D0%BE%20%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F).
2. Julia 1.8 Documentation [Электронный ресурс]. 2023. URL: <https://docs.julialang.org/en/v1/>.
3. OpenModelica [Электронный ресурс]. 2022. URL: <https://ru.wikipedia.org/wiki/OpenModelica>.