

Отчёт по лабораторной работе

Лабораторная работа №2 (вариант 10)

Сергеев Тимофей Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	20
	Список литературы	21

Список иллюстраций

4.1	Обновляем Julia и добавляем библиотеку Plots	9
4.2	Добавляем библиотеку DifferentialEquations	10
4.3	Вычисления для дальнейшего использования их в программе, часть 1	11
4.4	Вычисления для дальнейшего использования их в программе, часть 2	11
4.5	В соответствии с теорией, рассказанной на семинаре, создаём функцию, задаём начальные значения	12
4.6	Затем переходим к непосредственному добавлению элементов на основе полученных расчётов с помощью библиотеки Plots (создаём поле, задаём траектории катера, лодки а так же доходим до начала движения охраны их их первоначального положения; задаём точность, чтобы взять большее количество точек, тем самым получим более точный и плавный график	13
4.7	Обозначим точку, в которой охранники догонят браконьеров и сохраним полученный результат в формате png	14
4.8	В соответствии с теорией, рассказанной на семинаре, создаём функцию, задаём начальные значения	15
4.9	Затем переходим к непосредственному добавлению элементов на основе полученных расчётов с помощью библиотеки Plots (создаём поле, задаём траектории катера, лодки а так же доходим до начала движения охраны их их первоначального положения	16
4.10	Обозначим точку, в которой охранники догонят браконьеров и сохраним полученный результат в формате png	17
4.11	Запускаем и выполняем программы	17
4.12	Результат первого случая	18
4.13	Результат второго случая	18

Список таблиц

1 Цель работы

Построить математическую модель для выбора правильной стратегии при решении задач поиска на примере задачи преследования браконьеров береговой охраной.

2 Задание

- Подготовить инструменты для выполнения лабораторной работы;
- Записать уравнение, описывающее движение катера, с начальными условиями для двух случаев;
- Построить траекторию движения катера и лодки для двух случаев;
- 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. [1]

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. [2]

4 Выполнение лабораторной работы

1. Настраиваем рабочую среду для выполнения лабораторной работы, а именно обновляем Julia, добавляем библиотеки Plots и DifferentialEquations (рис. 4.1, 4.2).


```

┌───┐
│                                     | Documentation: https://docs.julialang.org
│                                     |
│  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ | Type "?" for help, "]" for Pkg help.
│  | | | | | | | | | | | | | | | | |
│  | | | | | | | | | | | | | | | | |
│  _/ | \ _ _ ' _ _ | | \ _ _ ' _ _ | Version 1.8.3 (2022-11-14)
│  | _ _ /           | |               | Official https://julialang.org/ release
│  | _ _ /           | |               |
└───┘

(@v1.8) pkg>

(@v1.8) pkg> update
Installing known registries into `~/var/tmp/tssergeev`
Updating registry at `~/var/tmp/tssergeev/registries/General.toml`
No Changes to `~/var/tmp/tssergeev/environments/v1.8/Project.toml`
No Changes to `~/var/tmp/tssergeev/environments/v1.8/Manifest.toml`
[ Info: We haven't cleaned this depot up for a bit, running Pkg.gc()...
Active manifest files: 1 found
Active artifact files: 0 found
Active scratchspaces: 0 found
Deleted no artifacts, repos, packages or scratchspaces

(@v1.8) pkg> add Plots
Resolving package versions...
Installed Xorg_xcb_util_renderutil_jll - v0.3.9+1
Installed Wayland_protocols_jll - v1.25.0+0
Installed IrrationalConstants - v0.1.1
Installed x265_jll - v3.5.0+0
Installed JpegTurbo_jll - v2.1.2+0
Installed FFMPEG - v0.4.1
Installed Scratch - v1.1.1
Installed ColorTypes - v0.11.4
Installed Showoff - v1.0.3
Installed Graphite2_jll - v1.3.14+0
Installed libfdk_aac_jll - v2.0.2+0
Installed libass_jll - v0.15.1+0
Installed Libmount_jll - v2.35.0+0
Installed Plots - v1.38.5
Installed Pixman_jll - v0.40.1+0
Installed XML2_jll - v2.10.3+0
Installed GR_jll - v0.71.7+0
Installed Xorg_libXext_jll - v1.3.4+4
Installed JSON - v0.21.3
Installed LERC_jll - v3.0.0+1
Installed Gettext_jll - v0.21.0+0
Installed Wayland_jll - v1.21.0+0
Installed TensorCore - v0.1.1
Installed Preferences - v1.3.0
Installed Latexify - v0.15.18
Installed Zstd_jll - v1.5.4+0
Installed Qt5Base_jll - v5.15.3+2

```

Рис. 4.1: Обновляем Julia и добавляем библиотеку Plots

```

[cf7118a7] + UUIDs
[4ec0a83e] + Unicode
[e66e0078] + CompilerSupportLibraries_jll v0.5.2+0
[deac9b47] + LibCURL_jll v7.84.0+0
[29816b5a] + LibSSH2_jll v1.10.2+0
[c8ffd9c3] + MbedTLS_jll v2.28.0+0
[14a3606d] + MozillaCACerts_jll v2022.2.1
[4536629a] + OpenBLAS_jll v0.3.20+0
[05823500] + OpenLibm_jll v0.8.1+0
[efcefd7] + PCRE2_jll v10.40.0+0
[83775a58] + Zlib_jll v1.2.12+3
[8e850b90] + libblastrampoline_jll v5.1.1+0
[8e850ede] + nghttp2_jll v1.48.0+0
[3f19e933] + p7zip_jll v17.4.0+0
Info Packages marked with ⚠ have new versions available but com
Precompiling project...
134 dependencies successfully precompiled in 74 seconds

julia>

(@v1.8) pkg> add DifferentialEquations
Resolving package versions...
Installed TreeViews _____ v0.3.0
Installed Calculus _____ v0.5.1
Installed OffsetArrays _____ v1.12.9
Installed HypergeometricFunctions _____ v0.3.11
Installed PDMats _____ v0.11.16
Installed StatsFuns _____ v1.1.1
Installed NonlinearSolve _____ v1.3.0
Installed DifferentialEquations _____ v7.6.0
Installed StaticArrays _____ v1.5.14
Installed EnumX _____ v1.0.4
Installed Polyester _____ v0.7.2
Installed Sundials_jll _____ v5.2.1+0
Installed CEnum _____ v0.4.2
Installed FunctionWrappers _____ v1.1.3
Installed RecursiveArrayTools _____ v2.37.0
Installed TriangularSolve _____ v0.1.19
Installed ArrayInterfaceOffsetArrays _____ v0.1.7
Installed ZygoteRules _____ v0.2.2
Installed RandomNumbers _____ v1.5.3
Installed Static _____ v0.8.3
Installed SLEEFpirates _____ v0.6.38
Installed Distances _____ v0.10.7
Installed BoundaryValueDiffEq _____ v2.11.0
Installed Tricks _____ v0.1.6
Installed KrylovKit _____ v0.6.0
Installed SteadyStateDiffEq _____ v1.12.0
Installed FiniteDiff _____ v2.17.0
Installed MuladdMacro _____ v0.2.4
Installed SIMDTypes _____ v0.1.0
Installed ArnoldiMethod _____ v0.2.0

```

Рис. 4.2: Добавляем библиотеку DifferentialEquations

2. Далее необходимо произвести математические расчёты в соответствии с материалами к лабораторной работе, но использовав свои данные, после чего дорешать получившееся уравнение. После выполнения расчётов ниже, мы получим траекторию движения катера в полярных координатах (рис.

4.3, 4.4).

$t_0 = 0$ $x_0 = 0$ $\dot{x}_0 = 0$ $\Theta = x_0 = 0$	$\dot{x} = \frac{x - x_1}{2.5V}$ $2.5Vx_1 = (x - x_1)V/1$ $2.5x_1 = x - x_1$ $x = 3.5x_1$ $x_1 = \frac{x}{3.5} = \frac{6.8}{3.5}$	$\frac{x_2}{V} = \frac{x_1 + K}{2.5V}$ $2.5Vx_2 = (x_1 + K)V/1$ $2.5x_2 = x_1 + K$ $x_2 = \frac{x_1 + K}{2.5} = \frac{6.8}{2.5}$	$V \frac{dr}{dt} = r \frac{d\theta}{dt}$
---	---	---	--

$V_f = \sqrt{7.24^2 + 1.2^2} = \sqrt{6.84} \rightarrow r \frac{d\theta}{dt} = \sqrt{6.84}$
 $\frac{d\theta}{dt} = \frac{\sqrt{6.84}}{r}$
 $\frac{dr}{dt} = \frac{r}{\sqrt{6.84}}$
 $\frac{dr}{r} = \frac{dt}{\sqrt{6.84}}$

Рис. 4.3: Вычисления для дальнейшего использования их в программе, часть 1

$\frac{dr}{d\theta} = \frac{r}{\sqrt{6.84}}$	$dr = \frac{r d\theta}{\sqrt{6.84}}$ $\frac{dr}{r} = \frac{d\theta}{\sqrt{6.84}}$	$\int \frac{dr}{r} = \int \frac{d\theta}{\sqrt{6.84}}$ $\ln r = \frac{\theta}{\sqrt{6.84}} + C$ $e^{\ln r} = e^{\frac{\theta}{\sqrt{6.84}} + C} = e^{\frac{\theta}{\sqrt{6.84}}} \cdot e^C = C e^{\frac{\theta}{\sqrt{6.84}}}$ $r = C e^{\frac{\theta}{\sqrt{6.84}}}$ $r(0) = r_0$ $1) r_0 = C e^{\frac{0}{\sqrt{6.84}}} = C = x_1$ $2) r_0 = C e^{\frac{0}{\sqrt{6.84}}} \rightarrow C = \frac{r_0}{e^{\frac{0}{\sqrt{6.84}}}}$
--	--	--

$1) r(0) = x_1 e^{\frac{0}{\sqrt{6.84}}}$
 $2) r(0) = \frac{x_1}{e^{\frac{0}{\sqrt{6.84}}}}$

Рис. 4.4: Вычисления для дальнейшего использования их в программе, часть 2

3. Теперь приступим к написанию программы на Julia (рис. 4.5, 4.6, 4.7).

```

1  # Решаем ОДУ
2
3  using DifferentialEquations
4  using Plots
5
6  """Правая часть ОДУ
7  u --- переменная (скаляр или массив)
8  p --- параметры (кортеж, tuple)
9  t --- аргумент (скаляр, время)
10 """
11 function F(u,p,t)
12 # p и t не используются
13 # но нужны для библиотеки
14 |   return u / sqrt(6.84)
15 end
16
17
18 "Начальное значение"
19 const u_0 = 6.8 / 1.8
20
21
22 "Интервал (кортеж, tuple)"
23 const T = (0, 2*pi)
24
25 # Задача
26 prob = ODEProblem(F, u_0, T)

```

Рис. 4.5: В соответствии с теорией, рассказанной на семинаре, создаём функцию, задаём начальные значения

```

27
28 # Решение задачи
29 sol = solve(
30     prob,
31     dtmax=0.05
32 )
33
34 plt = plot(
35     proj = :polar,
36     aspect_ratio=:equal,
37     dpi=300,
38     legend=true)
39
40 plot!(
41     plt,
42     sol.t,
43     sol.u,
44     xlabel="θ",
45     ylabel="r(t)",
46     label="Траектория катера",
47     color=:red)
48
49 plot!(
50     plt,
51     [0.0, sol.t[45]],
52     [0.0, 15],
53     xlabel="x",
54     ylabel="t",
55     label="Траектория лодки",
56     color=:green)
57 )
58
59 plot!(
60     plt,
61     [0.0, 0.0],
62     [6.8/1.8, 6.8],
63     xlabel="x",
64     ylabel="t",
65     label="",
66     color=:red)
67 )

```

Рис. 4.6: Затем переходим к непосредственному добавлению элементов на основе полученных расчётов с помощью библиотеки Plots (создаём поле, задаём траектории катера, лодки а так же доходим до начала движения охраны их их первоначального положения; задаём точность, чтобы взять большее количество точек, тем самым получим более точный и плавный график

```

plot!(
    plt,
    [sol.t[45]],
    [sol.u[45]],
    seriestype=:scatter,
    xlabel="t",
    ylabel="r(t)",
    label="Точка встречи",
    color=:black,
    title="Погоня")

savefig(plt, "lab2_first.png")

```

Рис. 4.7: Обозначим точку, в которой охранники догонят браконьеров и сохраним полученный результат в формате png

4. В зависимости от начального положения катера относительно полюса появляется второй случай (рис. 4.8, 4.9, 4.10).

```

1  # Решаем ОДУ
2
3  using DifferentialEquations
4  using Plots
5
6  """Правая часть ОДУ
7  u --- переменная (скаляр или массив)
8  p --- параметры (кортеж, tuple)
9  t --- аргумент (скаляр, время)
10 """
11 function F(u,p,t)
12 # p и t не используются
13 # но нужны для библиотеки
14     return u / sqrt(6.84)
15 end
16
17
18 "Начальное значение"
19 const u_0 = 6.8 / 3.8
20
21 "Интервал (кортеж, tuple)"
22 const T = (0, 2*pi)
23
24 # Задача
25 prob = ODEProblem(F, u_0, T)
26

```

Рис. 4.8: В соответствии с теорией, рассказанной на семинаре, создаём функцию, задаём начальные значения

```

# Решение задачи
sol = solve(
    prob,
    dtmax=0.05
)

plt = plot(
    proj = :polar,
    aspect_ratio=:equal,
    dpi=300,
    legend=true)

plot!(
    plt,
    sol.t,
    sol.u,
    xlabel="θ",
    ylabel="r(t)",
    label="Траектория катера",
    color=:red)

plot!(
    plt,
    [0.0, sol.t[45]],
    [0.0, 10],
    xlabel="x",
    ylabel="t",
    label="Траектория лодки",
    color=:green
)

plot!(
    plt,
    [0.0, 0.0],
    [-6.8, 6.8/3.8],
    xlabel="x",
    ylabel="t",
    label="",
    color=:red
)

```

Рис. 4.9: Затем переходим к непосредственному добавлению элементов на основе полученных расчётов с помощью библиотеки Plots (создаём поле, задаём траектории катера, лодки а так же доходим до начала движения охраны их их первоначального положения


```

plot!(
    plt,
    [sol.t[45]],
    [sol.u[45]],
    seriestype=:scatter,
    xlabel="t",
    ylabel="r(t)",
    label="Точка встречи",
    color=:black,
    title="Погоня")

savefig(plt, "lab2_second.png")

```

Рис. 4.10: Обозначим точку, в которой охранники догонят браконьеров и сохраним полученный результат в формате png

5. Последним действием необходимо запустить и выполнить программы следующим образом (рис. 4.11).

```

tssergeev@dk4n68 ~/work/study/2022-2023/Математическое моделирование/mathmod/labs/lab2/task $ julia lab2_first.jl
tssergeev@dk4n68 ~/work/study/2022-2023/Математическое моделирование/mathmod/labs/lab2/task $ julia lab2_second.jl

```

Рис. 4.11: Запускаем и выполняем программы

6. Получаем следующий результат (рис. 4.12, 4.13).

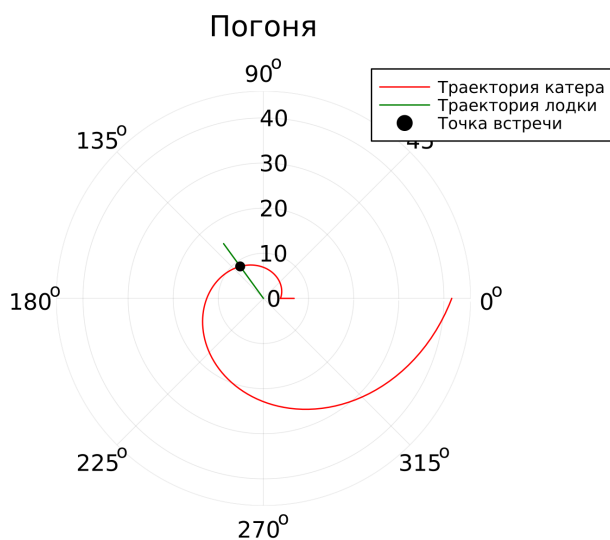


Рис. 4.12: Результат первого случая

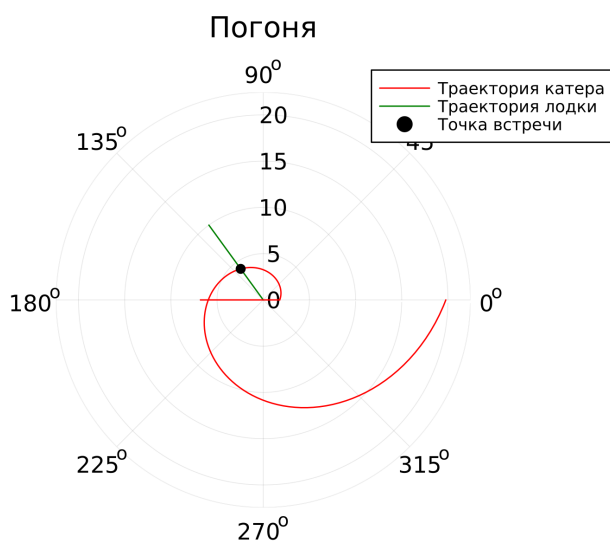


Рис. 4.13: Результат второго случая

7. Результаты соответствуют поставленной задаче: сперва береговая охрана доходит до точки начала движения в зависимости от положения относительно полюса, после чего начинает движение и доходит до точки пересечения с браконьерами, которая так же обозначена на рисунке.

8. С помощью стандартных решений OpenModelica не получится представить решение данной программы, поэтому приступим к решению лабораторных работ на этом языке при выполнении следующей лабораторной работы.

5 Выводы

Выполнив данную лабораторную работу, мы познакомились с языком программирования Julia и с некоторыми его основами, которые понадобятся нам при выполнении следующих лабораторных работ. В дальнейших работах будем осваивать язык OpenModelica и сравнивать реализации программ на этих языках.

Список литературы

1. Julia [Электронный ресурс]. Free Software Foundation, 2023. URL: [https://ru.wikipedia.org/wiki/Julia_\(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\)](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)).
2. OpenModelica [Электронный ресурс]. Free Software Foundation, 2023. URL: <https://ru.wikipedia.org/wiki/OpenModelica>.