

Отчет по лабораторной работе №2

Задача о погоне

Махорин Иван Сергеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Моделирование с помощью Julia	10
5	Выводы	14
6	Список литературы	15

Список иллюстраций

3.1	Получение нужного номера варианта	7
4.1	Скачивание Julia	10
4.2	Запуск Julia	10
4.3	Процесс запуска	11
4.4	Скачивание необходимых для работы пакетов	11
4.5	Скачивание необходимых для работы пакетов	11
4.6	Запуск кода	13
4.7	Случай 1	13
4.8	Случай 2	13

1 Цель работы

Решить задачу о погоне и изучить основы языка программирования Julia.

2 Задание

1. Запишите уравнение, описывающее движение катера, с начальными условиями для двух случаев (в зависимости от расположения катера относительно лодки в начальный момент времени).
2. Постройте траекторию движения катера и лодки для двух случаев.
3. Найдите точку пересечения траектории катера и лодки

3 Выполнение лабораторной работы

Расчитаем свой вариант по формуле и получаем наш вариант №59 (рис. 3.1).

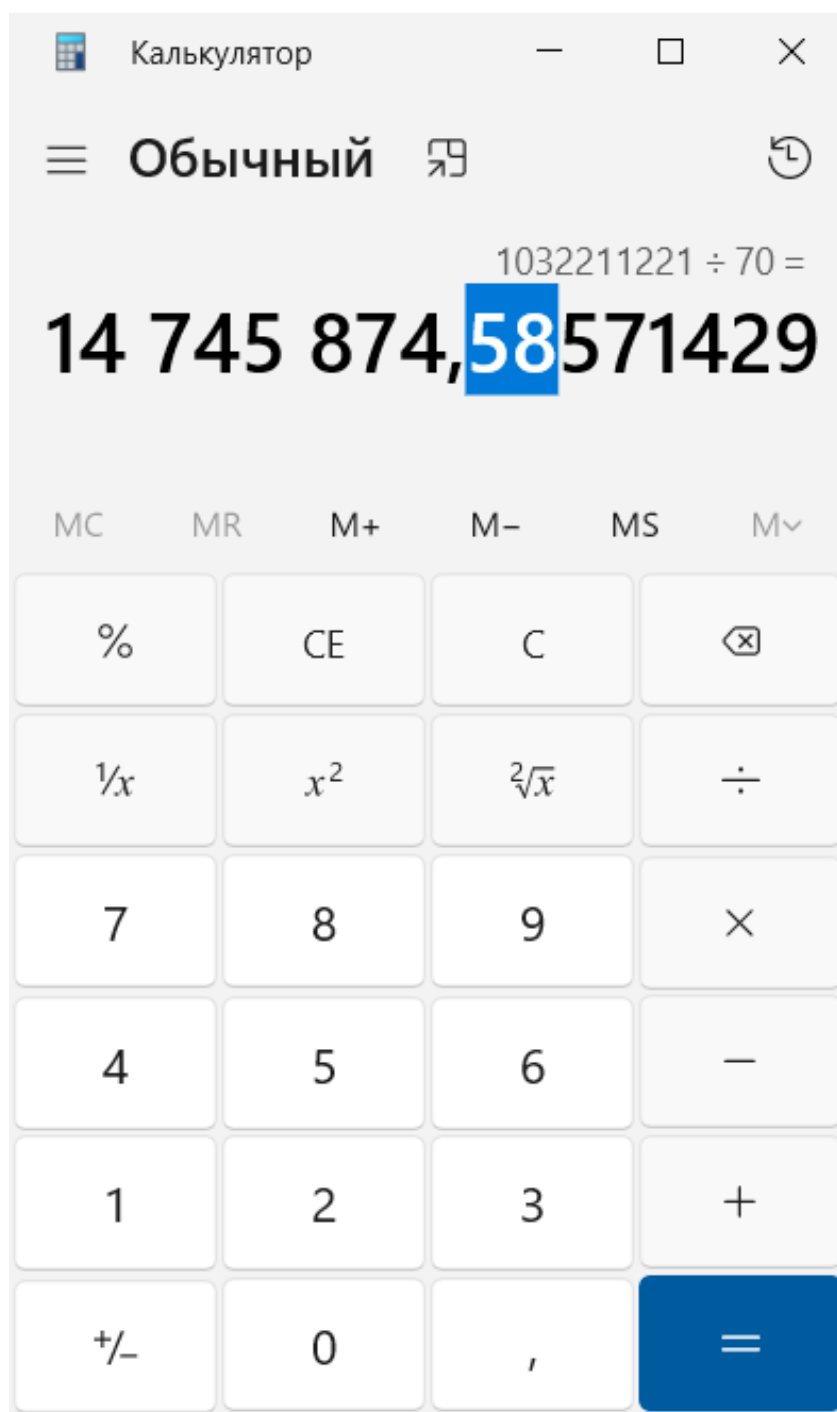


Рис. 3.1: Получение нужного номера варианта

1. На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 20,3 км от катера. Затем лодка снова скрыва-

ется в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 5,2 раза больше скорости браконьерской лодки.

2. 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 (20,3; 0). Обозначим скорость лодки v .
3. 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.
4. Пусть время t - время, через которое катер и лодка окажутся на одном расстоянии от начальной точки.

$$t = \frac{x}{v}$$

$$t = \frac{20,3 - x}{5,2v}$$

$$t = \frac{20,3 + x}{5,2v}$$

Из этих уравнений получаем объединение двух уравнений:

$$\begin{cases} \frac{x}{v} = \frac{20,3-x}{5,2v} \\ \frac{x}{v} = \frac{20,3+x}{5,2v} \end{cases}$$

Решая это, получаем два значения для x :

$$x_1 = 3,27419355$$

$$x_2 = 4,83333333$$

$$v_\tau$$

– тангенциальная скорость

$$v$$

– радиальная скорость

$$v = \frac{dr}{dt}$$

$$v_\tau = \sqrt{((5,2 * v)^2 - v^2)} = \frac{\sqrt{651} * v}{5}$$

$$\begin{cases} \frac{dr}{dt} = v \\ r \frac{d\theta}{dt} = \frac{\sqrt{651} * v}{5} \end{cases}$$

$$\begin{cases} \theta_0 = 0 \\ r_0 = x_1 = 3,27419355 \end{cases}$$

или

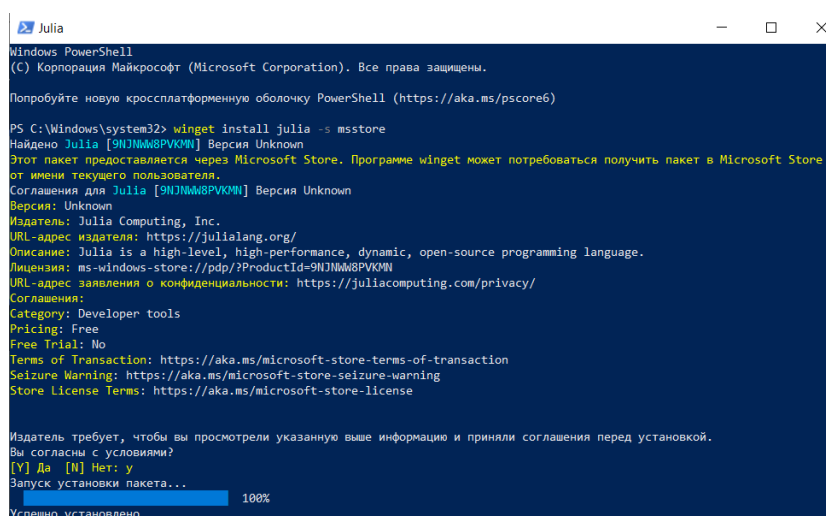
$$\begin{cases} \theta_0 = -\pi \\ r_0 = x_2 = 4,83333333 \end{cases}$$

Итоговое уравнение после того, как убрали производную по t:

$$\frac{dr}{d\theta} = \frac{5r}{\sqrt{651}}$$

4 Моделирование с помощью Julia

1. Скачиваем Julia (рис. 4.1).



```
Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

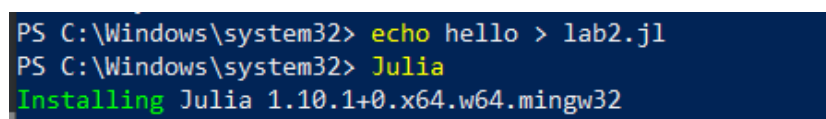
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Windows\system32> winget install julia -s msstore
Найдено Julia [9NJNMW8PVKMN] Версия Unknown
Этот пакет предоставляется через Microsoft Store. Программе winget может потребоваться получить пакет в Microsoft Store
от имени текущего пользователя.
Соглашения для Julia [9NJNMW8PVKMN] Версия Unknown
Версия: Unknown
Издатель: Julia Computing, Inc.
URL-адрес издателя: https://julialang.org/
Описание: Julia is a high-level, high-performance, dynamic, open-source programming language.
Лицензия: ms-windows-store://pdp/?ProductId=9NJNMW8PVKMN
URL-адрес заявления о конфиденциальности: https://juliacomputing.com/privacy/
Соглашения:
Category: Developer tools
Pricing: Free
Free Trial: No
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
Seizure Warning: https://aka.ms/microsoft-store-seizure-warning
Store License Terms: https://aka.ms/microsoft-store-license

Издатель требует, чтобы вы просмотрели указанную выше информацию и приняли соглашения перед установкой.
Вы согласны с условиями?
[Y] Да [N] Нет: y
Запуск установки пакета...
100%
Успешно установлено
```

Рис. 4.1: Скачивание Julia

2. Запускаем Julia (рис. 4.2).



```
PS C:\Windows\system32> echo hello > lab2.jl
PS C:\Windows\system32> Julia
Installing Julia 1.10.1+0.x64.w64.mingw32
```

Рис. 4.2: Запуск Julia

3. Процесс запуска Julia (рис. 4.3).

```
Installing Julia 1.10.1+0.x64.w64.mingw32

Documentation: https://docs.julialang.org

Type "?" for help, "]"? for Pkg help.

Version 1.10.1 (2024-02-13)
Official https://julialang.org/ release

julia>
```

Рис. 4.3: Процесс запуска

4. Скачаем необходимые для работы пакеты (рис. 4.4-4.5).

```
Julia

Installed Unzip v0.2.0
Installed UnitfulLatexify v1.6.3
Installed SortingAlgorithms v1.2.1
Downloaded artifact: jpegTurbo
Downloaded artifact: x265
Downloaded artifact: libfdk_aac
Downloaded artifact: GR
Downloaded artifact: LERC
Downloaded artifact: Opus
Downloaded artifact: Cairo
Downloaded artifact: Fontconfig
Downloaded artifact: Libpng_error
Downloaded artifact: Bzip2
Downloaded artifact: XZ
Downloaded artifact: HarfBuzz
Downloaded artifact: LZ0
Downloaded artifact: fzf
Downloaded artifact: FriBidi
Downloaded artifact: GLFW
Downloaded artifact: x264
Downloaded artifact: FreeType2
Downloaded artifact: libpng
Downloaded artifact: libaom
Downloaded artifact: gperf
Downloaded artifact: Zstd
Downloaded artifact: Expat
Downloaded artifact: Libtiff
Downloaded artifact: Libffi
Downloaded artifact: Ogg
Downloaded artifact: Vulkan_loader
Downloaded artifact: XSLT
Downloaded artifact: Wayland_protocols
Downloaded artifact: Graphite2
Downloaded artifact: libass
Downloaded artifact: Pixman
```

Рис. 4.4: Скачивание необходимых для работы пакетов

```
Precompiling project...
163 dependencies successfully precompiled in 444 seconds. 152 already precompiled.
1 dependency had output during precompilation:
MKL_jll

Downloading artifact: MKL

[pid 17400] waiting for IO to finish:
Handle type uv_handle_t->data
timer 0000023524874a80->00000235224cf610
This means that a package has started a background task or event source that has not finished running. For precompila
tion to complete successfully, the event source needs to be closed explicitly. See the developer documentation on fixing
precompilation hangs for more help.

[pid 17400] waiting for IO to finish:
Handle type uv_handle_t->data
timer 0000023524874a80->00000235224cf610
This means that a package has started a background task or event source that has not finished running. For precompila
tion to complete successfully, the event source needs to be closed explicitly. See the developer documentation on fixing
precompilation hangs for more help.

julia> using Plots
```

Рис. 4.5: Скачивание необходимых для работы пакетов

5. Код для файла lab2.jl:

```
using Plots using DifferentialEquations
const a = 20.3 const n = 5.2
const r0 = a/(n + 1) const r0_2 = a/(n - 1)
const T = (0, 2*pi) const T_2 = (-pi, pi)
function F(u, p, t) return u / sqrt(n*n - 1) end
problem = ODEProblem(F, r0, T)
result = solve(problem, abstol=1e-8, reltol=1e-8) [show?] result.u [show?] result.t
dxR = rand(1:size(result.t)[1]) rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]
plt = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)
plot!(plt, xlabel="theta", ylabel="r(t)", title="Случай номер 1", legend=:outerbottom)
plot!(plt, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь лод-
ки", color=:blue, lw=1) scatter!(plt, rAngles, result.u, label="", mc=:blue, ms=0.0005)
plot!(plt, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь катера",
color=:green, lw=1) scatter!(plt, result.t, result.u, label="", mc=:green, ms=0.0005)
savefig(plt, "lab2_01.png")
problem = ODEProblem(F, r0_2, T_2) result = solve(problem, abstol=1e-8, reltol=1e-
8) dxR = rand(1:size(result.t)[1]) rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]
plt1 = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)
plot!(plt1, xlabel="theta", ylabel="r(t)", title="Случай номер 2", legend=:outerbottom)
plot!(plt1, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь
лодки", color=:blue, lw=1) scatter!(plt1, rAngles, result.u, label="", mc=:blue,
ms=0.0005) plot!(plt1, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь
катера", color=:green, lw=1) scatter!(plt1, result.t, result.u, label="", mc=:green,
ms=0.0005)
savefig(plt1, "lab2_02.png")
```

6. Запуск кода (рис. 4.6).

```

Администратор: Windows PowerShell
657175, 6.161025134819635, 7.383001074062188, 8.956702729564856, 10.973790852959699, 11.216196309581527]
result.t = [0.0, 0.04887440408060568, 0.3507536629041826, 0.9181946440187447, 1.6037607827662956, 2.370919548874501, 3.2
259338896021097, 4.149244725666782, 5.135244831089191, 6.171690894826499, 6.283185307179586]
PS C:\Users\Ivan\work\study\2023-2024\Математическое моделирование\mathmod\labs\lab2>

```

Рис. 4.6: Запуск кода

7. Просмотр результата работы (рис. 4.7-4.8).

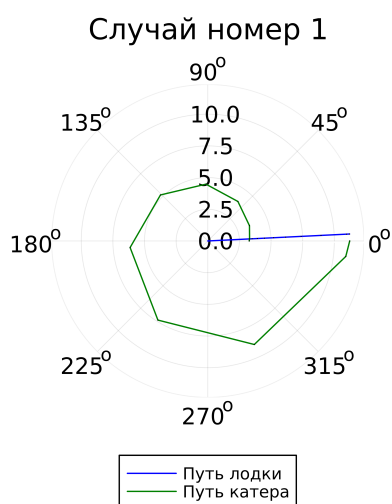


Рис. 4.7: Случай 1

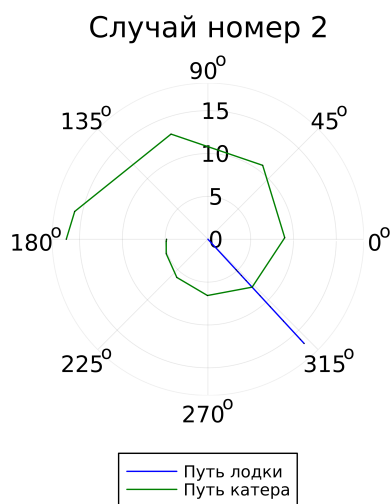


Рис. 4.8: Случай 2

5 Выводы

Были изучены основы языка программирования Julia и его библиотеки, которые используются для построения графиков и решения дифференциальных уравнений. А также решили задачу о погоне.

6 Список литературы

[1] Документация по Julia: <https://docs.julialang.org/en/v1/>

[2] Решение дифференциальных уравнений: <https://www.wolframalpha.com/>