

Отчёт по лабораторной работе №2

Математическое моделирование

Задача о погоне. Вариант №30

Выполнила: Данзанова Саяна Зоригтоевна, НПИбд-01-21

Содержание

Цель работы	5
Задание	6
Задача о погоне (вариант 30):	6
Задачи:	7
Выполнение лабораторной работы	8
Математическая модель	8
Решение с помощью программ	10
OpenModelica	10
Julia	10
Результаты работы кода	13
Анализ полученных результатов	16
Вывод	17
Список литературы. Библиография	18

Список иллюстраций

1	“Установка Julia”	10
2	“Установка библиотек”	11
3	“Компиляция программы lab02.jl”	13
4	“График для первого случая lab02.jl”	14
5	“График для второго случая lab02.jl”	15

Список таблиц

Цель работы

Изучить основы языков программирования Julia и OpenModelica. Освоить библиотеки этих языков, которые используются для построения графиков и решения дифференциальных уравнений. Решить задачу о погоне.

Задание

Задача о погоне (вариант 30):

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 12,2 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 4,1 раза больше скорости браконьерской лодки.

Задачи:

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

Выполнение лабораторной работы

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 $(12, 2; 0)$. 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 $12, 2 + x$ (или $12, 2 - x$, в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как $\frac{x}{v}$ или $\frac{12, 2 - x}{4, 1v}$ ($\frac{12, 2 + x}{4, 1v}$). 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):

$$\begin{cases} \frac{x}{v} = \frac{12,2-x}{4,1v} \\ \frac{x}{v} = \frac{12,2+x}{4,1v} \end{cases}$$

Из данных уравнений можно найти расстояние, после которого катер начнёт раскручиваться по спирали. Для данных уравнений решения будут следующими: $x_1 = \frac{122}{51}$, $x_2 = \frac{122}{31}$. 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.

$$v_\tau = \frac{\sqrt{1581}v}{10}$$

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

$$\begin{cases} \frac{dr}{dt} = v \\ r \frac{d\theta}{dt} = \frac{\sqrt{1581}v}{10} \end{cases}$$

с начальными условиями

$$\begin{cases} \theta_0 = 0 \\ r_0 = x_1 = \frac{122}{51} \end{cases}$$

или

$$\begin{cases} \theta_0 = -\pi \\ r_0 = x_2 = \frac{122}{31} \end{cases}$$

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

$$\frac{dr}{d\theta} = \frac{10r}{\sqrt{1581}}$$

Решением этого уравнения с заданными начальными условиями и будет являться траектория движения катера в полярных координатах. [3]

Решение с помощью программ

OpenModelica

К сожалению, OpenModelica не адаптирована к использованию полярных координат, поэтому адекватное отображение результатов данной задачи там невозможно. [2]

Julia

Решить дифференциальное уравнение, расписанное в постановке задачи лабораторной работы, поможет библиотека DifferentialEquations. Итоговые изображения в полярных координатах будут строиться через библиотеку Plots. [1]

Установим Julia:

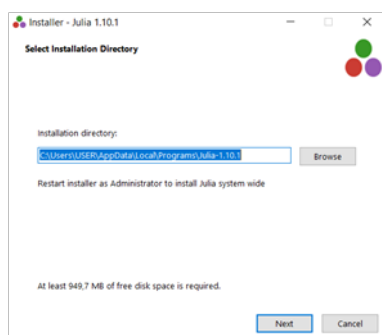
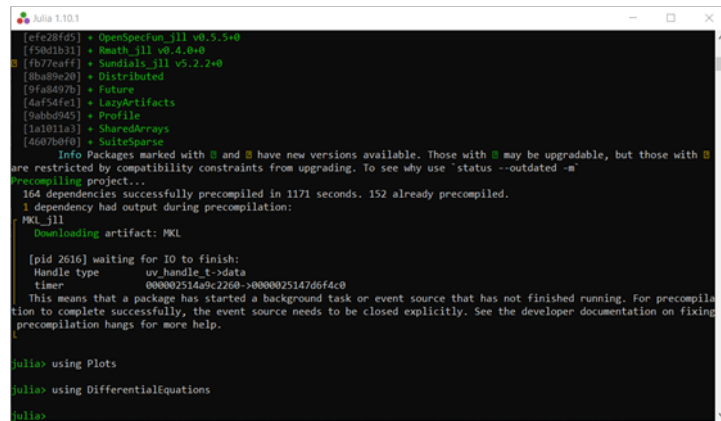


Рис. 1: “Установка Julia”

Установим нужные для работы библиотеки, проверим их установку:



```
Julia 1.10.1
[efe28fd5] + OpenSpectrum_jll v0.5.3+0
[f50d1b31] + Rmath_jll v0.4.0+0
[f677eaff] + Sundials_jll v5.2.2+0
[8ba89e20] + Distributed
[9fa492b5] + Future
[4a54fe1] + LazyArtifacts
[9abb0945] + Profile
[1a1011a3] + SharedArrays
[4607b0f0] + SuiteSparse
Info: Packages marked with ⚠ and ⚠ have new versions available. Those with ⚠ may be upgradable, but those with ⚠
are restricted by compatibility constraints from upgrading. To see why use `status --outdated -m`
Precompiling project...
164 dependencies successfully precompiled in 1171 seconds. 152 already precompiled.
1 dependency had output during precompilation:
MKL_jll
Downloading artifact: MKL
[pid 2616] waiting for IO to finish:
Handle type uv_handle_t->data
timer 000002514a9c2260->0000025147d6f4c0
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
julia> using DifferentialEquations
julia>
```

Рис. 2: “Установка библиотек”

Код программы:

```
using Plots
using DifferentialEquations

# расстояние от лодки до катера
const a = 12.2
const n = 4.1

# расстояние начала спирали
const r0 = a/(n + 1)
const r0_2 = a/(n - 1)
# интервал
const T = (0, 2*pi)
const T_2 = (-pi, pi)

function F(u, p, t)
    return u / sqrt(n*n - 1)
end
```

```

# задача ОДУ
problem = ODEProblem(F, r0, T)

#решение
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]]

#холст1
plt = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)

#параметры для холста
plot!(plt, xlabel="theta", ylabel="r(t)", title="Задача о погоне - случай 1", legend=:none)
plot!(plt, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь катера")
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")
scatter!(plt, result.t, result.u, label="", mc=:green, ms=0.0005)

savefig(plt, "lab02_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]]

#холст2

```

```
plt1 = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)

#параметры для холста
plot!(plt1, xlabel="theta", ylabel="r(t)", title="Задача о погоне - случай 2", lc=:black)
plot!(plt1, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь катера", lw=2, mc=:blue)
scatter!(plt1, rAngles, result.u, label="", mc=:blue, ms=0.0005)
plot!(plt1, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь лодки", lw=2, mc=:green)
scatter!(plt1, result.t, result.u, label="", mc=:green, ms=0.0005)

savefig(plt1, "lab02_02.png")
```

Скомпилируем файл командой в PowerShell:

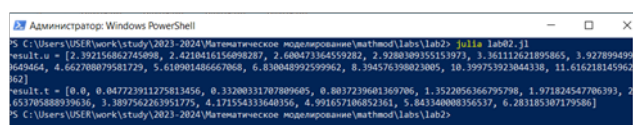


Рис. 3: “Компиляция программы lab02.jl”

Результаты работы кода

На рис. @fig:004 и @fig:005 изображены итоговые графики траектории движения катера и лодки для обоих случаев.

Задача о погоне - случай 1

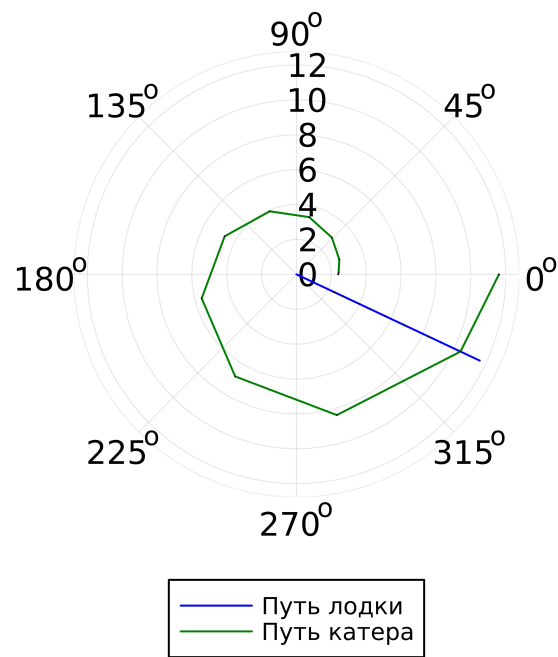


Рис. 4: “График для первого случая lab02.jl”

Задача о погоне - случай 2

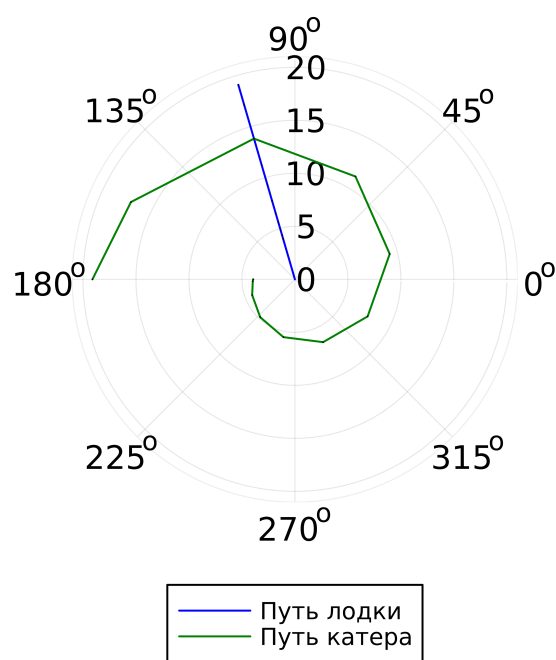


Рис. 5: “График для второго случая lab02.jl”

Анализ полученных результатов

Мною были построены графики для обоих случаев. На них получилось отрисовать траекторию катера, траекторию лодки и получилось наглядно найти их точки пересечения. Мы успешно решили задачу о погоне.

Вывод

Были изучены основы языков программирования Julia и OpenModelica. Освоены библиотеки этих языков, которые используются для построения графиков и решения дифференциальных уравнений. Поскольку OpenModelica не работает с полярными координатами, она пока что не была использована в данной лабораторной работе.

Список литературы. Библиография

- [1] Документация по Julia: <https://docs.julialang.org/en/v1/>
- [2] Документация по OpenModelica: <https://openmodelica.org/>
- [3] Решение дифференциальных уравнений: <https://www.wolframalpha.com/>