

Лабораторная работа №2

Задача о погоне. Вариант 53

Чванова Ангелина Дмитриевна, НПИбд-02-21

Содержание

Цель работы	4
Задачи	5
Теоретическое введение	6
Julia	6
Выполнение лабораторной работы	7
Произведение расчетов	7
Моделирование	11
Результаты работы	12
Выводы	14

Список иллюстраций

1	Вычисление варианта	7
2	Вычисления x_1	8
3	Вычисления x_2	9
4	Вычисления	10
5	Установка Julia	11
6	Julia	11
7	Результат запуска программы - график №1	13
8	Результат запуска программы - график №2	13

Цель работы

Решение задачи о погоне, а также изучение основ языка программирования Julia.

Задачи

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

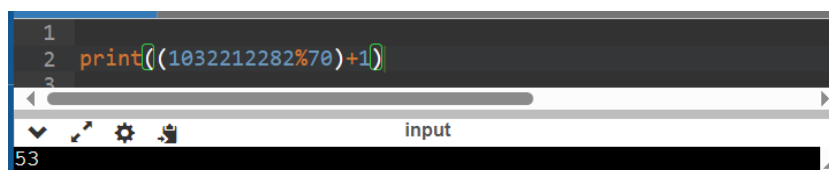
Теоретическое введение

Julia

Julia – это открытый свободный высокопроизводительный динамический язык высокого уровня, созданный специально для технических (математических) вычислений. Его синтаксис близок к синтаксису других сред технических вычислений, таких как Matlab и Octave. Он имеет в своем составе сложный компилятор, обеспечивает распределенное параллельное выполнение инструкций, вычислительную точность и обширную библиотеку математических функций. Имеет встроенную поддержку многопоточности и распределённых вычислений, реализованные в том числе в стандартных конструкциях.

Выполнение лабораторной работы

Выбор варианта вычислялся остатком от деления студенческого билета на количество вариантов, плюс один. Таким образом Получили 53 вариант (Рис.1).



```
1  
2 print((1032212282%70)+1)  
3  
input  
53
```

Рис. 1: Вычисление варианта

Произведение расчетов

1. За момент отсчета времени примем момент первого рассеивания тумана. Введем полярные координаты с центром в точке нахождения браконьеров и осью, проходящей через катер береговой охраны. Тогда начальные координаты катера (17,6; 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 во-

круг 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 $17,6 + x$ (или $17,6 - x$, в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как x/v или $(17,6 - x)/4,7v$, $((17,6 + x)/4,7v)$. 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):

$$\begin{cases} x/v = (17,6 - x)/4,7v \\ x/v = (17,6 + x)/4,7v \end{cases}$$

Из данных уравнений можно найти расстояние, после которого катер начнёт раскручиваться по спирали. Для данных уравнений решения будут следующими (Рис.2-3): $x_1 = 176/57$, $x_2 = 176/37$.

←
Калькулятор

$$\frac{x}{v} = \frac{17,6 - x}{4,7v}$$

$$x = \frac{176}{57}, v \neq 0$$

Альтернативная форма

$$x \approx 3,08772, v \neq 0$$

Рис. 2: Вычисления x_1

←
Калькулятор

$$\frac{x}{v} = \frac{17,6 + x}{4,7v}$$

$$x = \frac{176}{37}, v \neq 0$$

Альтернативная форма

$$x \approx 4,75676, v \neq 0$$

Рис. 3: Вычисления x_2

Задачу решаем для 2 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: $v_r = dr/dt = v$ - радиальная скорость и $v_\tau = r d\theta/dt$ - тангенциальная скорость (Рис.4).

$$v_\tau = \sqrt{2109}v/10$$

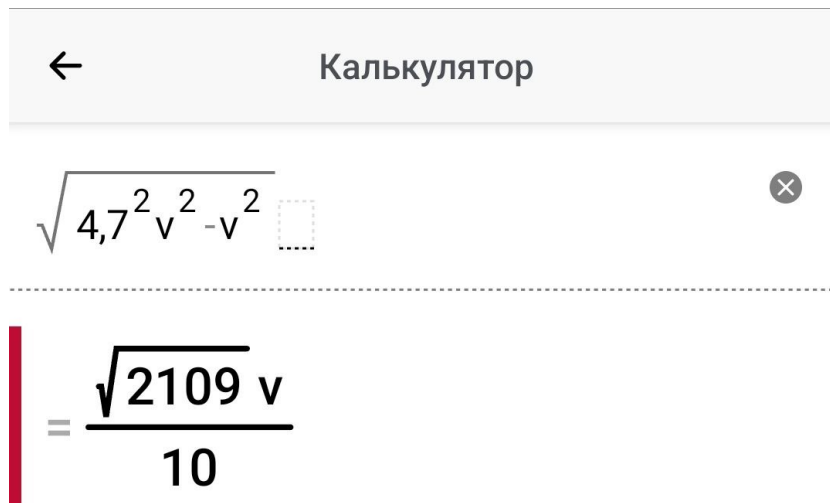


Рис. 4: Вычисления

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

$$\begin{cases} dr/dt = v \\ rd\theta/dt = \sqrt{2109}v/10 \end{cases}$$

с начальными условиями

$$\begin{cases} \theta_0 = 0 \\ r_0 = x_1 = 176/57 \end{cases}$$

или

$$\begin{cases} \theta_0 = -\pi \\ r_0 = x_2 = 176/37 \end{cases}$$

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

$$dr/d\theta = 10r/\sqrt{2109}$$

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

Моделирование

OpenModelica не может быть использована для этой задачи, так как здесь используются полярные координаты.

Установка Julia и необходимых для нее пакетов (Рис.5-6).

```
PS C:\Windows\system32> winget install julia -s msstore
Перед использованием источника "msstore" необходимо просмотреть следующие соглашения.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
Для правильной работы источника требуется отправить во внутреннюю службу двухбуквенный код текущего региона компьютера (
например, "RU").

Вы согласны со всеми условиями исходных соглашений?
[Y] Да [N] Нет: y
Найдено Julia [9NJNM8PVKMM] Версия Unknown
Этот пакет предоставляется через Microsoft Store. Программе winget может потребоваться получить пакет в Microsoft Store
от имени текущего пользователя.
Соглашения для Julia [9NJNM8PVKMM] 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=9NJNM8PVKMM
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. 5: Установка Julia

```
PS C:\Windows\system32> echo hello > lab_2.jl
PS C:\Windows\system32> Julia
Installing Julia 1.10.1+0.x64.w64.mingw32
```

Рис. 6: Julia

Исходный код программы:

using Plots using DifferentialEquations

#расстояние от лодки до катера const a = 17.6 const n = 4.7

#расстояние начала закругления-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="Чванова. Вар
53. Задача о погоне. Случай 1", legend=:outerbottom) plot!(plt, [rAngles[1], rAngles[2]],
[0.0, result.u[size(result.u)[1]]], label="Путь лодки", 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]]
#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="Чванова. Вар
53. Задача о погоне. Случай 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")

```

Результаты работы

Запуск программы и получение результатов (Рис.7-9):

```

PS C:\Users\adchv\work\study\2023-2024\Математическое моделирование\mathmod\labs\lab2> julia
result.u = [3.0877192982456143, 3.120325047211804, 3.345121267212414, 3.759980962632483, 4.3
73697, 5.959934361414241, 7.160864002360708, 8.704456026934224, 10.684211794760897, 12.12909
result.t = [0.0, 0.048240483172708334, 0.36771327276985527, 0.904613357121681, 1.53162129445
.0200751800977828, 3.86310469809919, 4.759558627347584, 5.7006854405979, 6.283185307179586]
PS C:\Users\adchv\work\study\2023-2024\Математическое моделирование\mathmod\labs\lab2>

```

Чванова. Вар 53. Задача о погоне. Случай 1

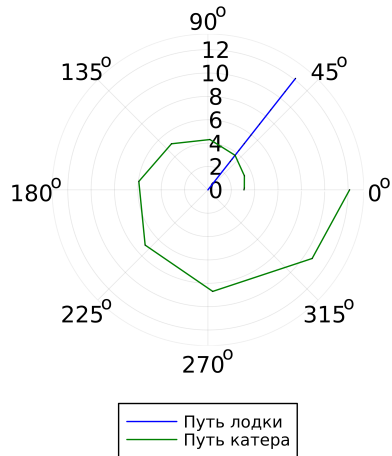


Рис. 7: Результат запуска программы - график №1

Чванова. Вар 53. Задача о погоне. Случай 2

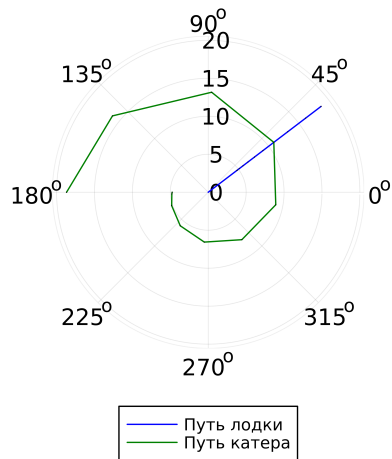


Рис. 8: Результат запуска программы - график №2

Выводы

Нами была решена задача о погоне, а также изучены основы языка программирования Julia, были выполнены все поставленные задачи: построение графиков для обоих случаев, где получилось отрисовать траекторию катера, траекторию лодки и получилось наглядно найти их точки пересечения.