

Отчёт по лабораторной работе №2

Математическое моделирование

Задача о погоне. Вариант №27

Выполнила: Самсонова Мария Ильинична,
НФИбд-02-21, 1032216526

Содержание

Цель лабораторной работы №2	4
Теоретические данные	5
Задание лабораторной работы №2	7
Выполнение лабораторной работы №2	8
Информация о математической модели	8
Практическая часть. Решение задачи с помощью программ	11
Openmodelica	11
Julia	11
Результаты работы кода на Julia	14
Анализ итоговых результатов и вывод	16
Список литературы. Библиография	17

Список иллюстраций

1	(рис. 1. Формула вычисления варианта и её вывод)	7
1	“Установщик Julia.exe”	12
2	“Проверка установки библиотек”	12
3	“Проверка установки библиотек”	13
4	“Код программы. Часть 1”	13
5	“Код программы. Часть 2”	13
6	“Компляция программы lab02.jl”	14
7	“Полученные результаты. Первый случай”	14
8	“Полученные результаты. Второй случай”	15

Цель лабораторной работы №2

Изучение основ языков программирования Julia и OpenModelica и освоение библиотек данных языков, которые необходимы для построения графиков и решение дифференциальных уравнений.

Теоретические данные

Информация о языках программирования:

- 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.
- 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. По своим возможностям приближается к таким вычислительным средам как Matlab Simulink, Scilab xCos, имея при этом значительно более удобное представление системы уравнений исследуемого блока.

Математические термины:

- Дифференциальное уравнение — уравнение, которое помимо функции содержит её производные. Порядок входящих в уравнение производных может быть различен (формально он ничем не ограничен). Производные, функции, независимые

переменные и параметры могут входить в уравнение в различных комбинациях или отсутствовать вовсе, кроме хотя бы одной производной. Не любое уравнение, содержащее производные неизвестной функции, является дифференциальным.

В отличие от алгебраических уравнений, в результате решения которых ищется число (несколько чисел), при решении дифференциальных уравнений ищется функция (семейство функций).

Дифференциальное уравнение порядка выше первого можно преобразовать в систему уравнений первого порядка, в которой число уравнений равно порядку исходного дифференциального уравнения.

- Тангенциальная скорость - составляющая вектора скорости, перпендикулярная линии, соединяющей источник и наблюдателя. Измеряется собственному движению - угловому перемещению источника.
- Радиальная скорость — проекция скорости точки на прямую, соединяющую её с выбранным началом координат.
- Полярная система координат — двумерная система координат, в которой каждая точка на плоскости определяется двумя числами — полярным углом и полярным радиусом.

Задание лабораторной работы №2

1. Вычислите собственный вариант: (исходя из формулы $N_{student} \bmod K_{of variants} + 1$).

Этот же вариант будет использоваться для всех последующих лабораторных работ.

```
>>> print(1032216526%70+1)
27
```

Рис. 1: (рис. 1. Формула вычисления варианта и её вывод)

Условия задачи о погоне, 27 вариант:

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 11,7 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 3,7 раза больше скорости браконьерской лодки.

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

Выполнение лабораторной работы

№2

Информация о математической модели

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 $(11,7; 0)$. 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 $11,7 + x$ (или $11,7 - x$, в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как $\frac{x}{v}$ или $\frac{11,7-x}{3,7v}$ ($\frac{11,7+x}{3,7v}$). 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):

$$\begin{cases} \frac{x}{v} = \frac{11,7-x}{3,7v} \\ \frac{x}{v} = \frac{11,7+x}{3,7v} \end{cases}$$

Из данных уравнений можно найти расстояние, после которого катер начнёт раскручиваться по спирали. Для данных уравнений решения будут следующими: $x_1 = \frac{117}{47}$, $x_2 = \frac{13}{3}$. 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.

$$v_\tau = \frac{\sqrt{141} * 3v}{10}$$

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

$$\begin{cases} \frac{dr}{dt} = v \\ r \frac{d\theta}{dt} = \frac{\sqrt{141} * 3v}{10} \end{cases}$$

с начальными условиями

$$\begin{cases} \theta_0 = 0 \\ r_0 = x_1 = \frac{117}{47} \end{cases}$$

или

$$\begin{cases} \theta_0 = -\pi \\ r_0 = x_2 = \frac{13}{3} \end{cases}$$

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

$$\frac{dr}{d\theta} = \frac{10r}{\sqrt{141}}$$

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

Практическая часть. Решение задачи с помощью программ

Openmodelica

В данной программе не используются полярные координаты, поэтому мы не можем решить данную задачу с помощью данной программы.

Julia

Используя библиотеку DifferentialEquations, мы в данной программе решаем дифференциальное уравнение, расписанное в постановке задачи лабораторной работы. Благодаря библиотеке Plots мы получаем Итоговые изображения в полярных координатах. [1]

Установим Julia:

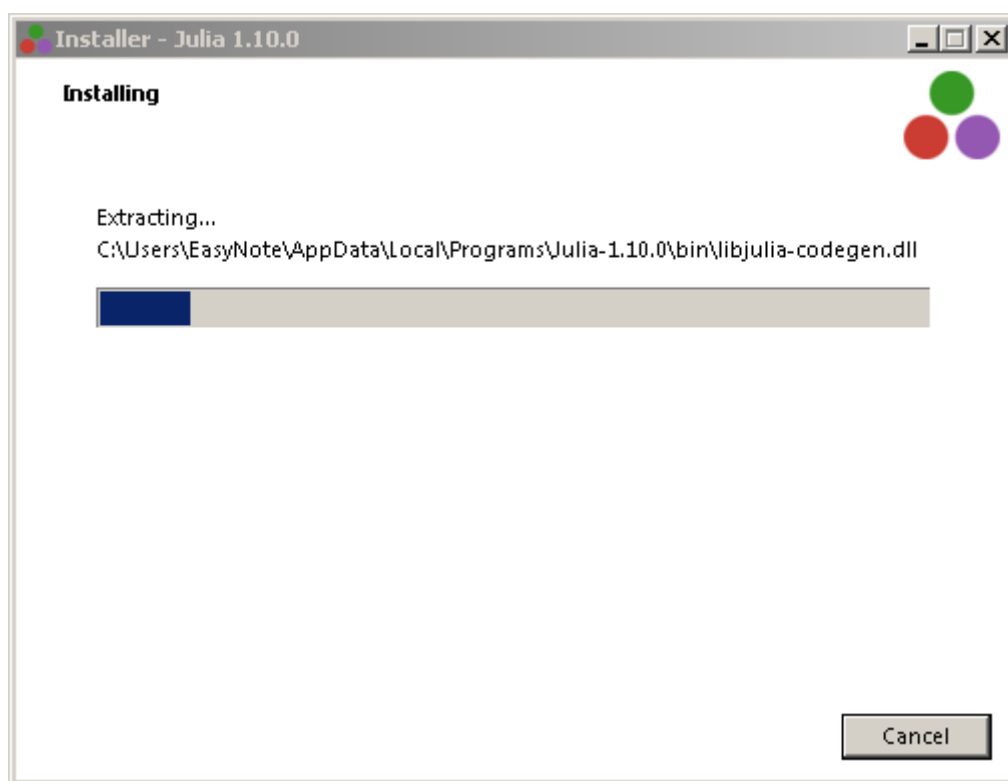


Рис. 1: “Установщик Julia.exe”

Установим нужные библиотеки, проверим их установку:

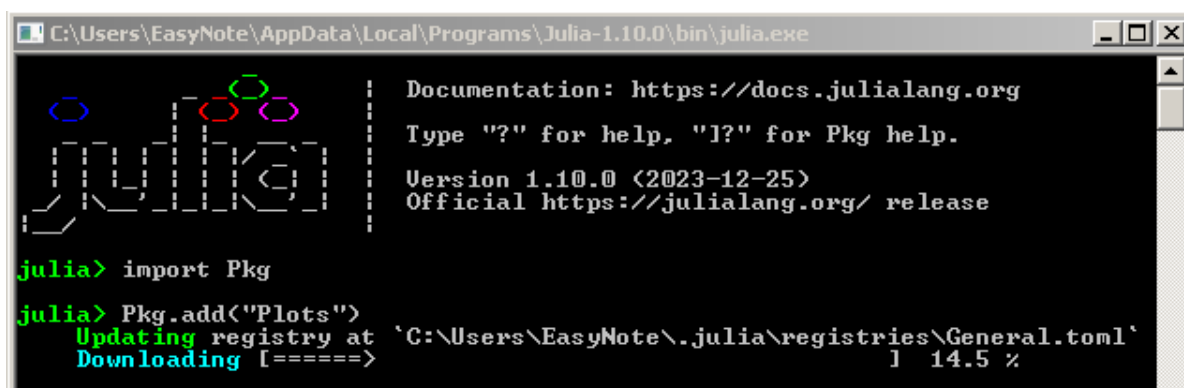


Рис. 2: “Проверка установки библиотек”

```
julia> Pkg.add("DifferentialEquations")
Installed Calculus v0.5.1
Progress [=> 1 1/126
```

Рис. 3: “Проверка установки библиотек”

Код программы:

```
1 using Plots
2 using DifferentialEquations
3
4 # расстояние от лодки до катера
5 const a = 11.7
6 const n = 3.7
7
8 # расстояние начала спирали
9 const r0 = a/(n + 1)
10 const r0_2 = a/(n - 1)
11 # интервал
12 const T = (0, 2*pi)
13 const T_2 = (-pi, pi)
14
15 function F(u, p, t)
16 | return u / sqrt(n*n - 1)
17 end
18
19 # задача ОДУ
20 problem = ODEProblem(F, r0, T)
21
22 # решение
23 result = solve(problem, abstol=1e-8, reltol=1e-8)
24 @show result.u
25 @show result.t
26
27 dxR = rand(1:size(result.t)[1])
28 rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]
29
30 #хорд1
31 plt = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)
32
33 #параметры для холста
34 plot!(plt, xlabel="theta", ylabel="r(t)", title="Задача 0 погоне - случай 1", legend=:outerbottom)
35 plot!(plt, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь лодки", color=:blue, lw=1)
36 scatter!(plt, rAngles, result.u, label="", mc=:blue, ms=0.0005)
```

Рис. 4: “Код программы. Часть 1”

```
33 #параметры для холста
34 plot!(plt, xlabel="theta", ylabel="r(t)", title="Задача 0 погоне - случай 1", legend=:outerbottom)
35 plot!(plt, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь лодки", color=:blue, lw=1)
36 scatter!(plt, rAngles, result.u, label="", mc=:blue, ms=0.0005)
37 plot!(plt, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь катера", color=:green, lw=1)
38 scatter!(plt, result.t, result.u, label="", mc=:green, ms=0.0005)
39
40 savefig(plt, "lab02_01.png")
41
42 problem = ODEProblem(F, r0_2, T_2)
43 result = solve(problem, abstol=1e-8, reltol=1e-8)
44 dxR = rand(1:size(result.t)[1])
45 rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]
46
47 #хорд2
48 plt1 = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)
49
50 #параметры для холста
51 plot!(plt1, xlabel="theta", ylabel="r(t)", title="Задача 0 погоне - случай 2", legend=:outerbottom)
52 plot!(plt1, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь лодки", color=:blue, lw=1)
53 scatter!(plt1, rAngles, result.u, label="", mc=:blue, ms=0.0005)
54 plot!(plt1, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь катера", color=:green, lw=1)
55 scatter!(plt1, result.t, result.u, label="", mc=:green, ms=0.0005)
56
57 savefig(plt1, "lab02_02.png")
```

Рис. 5: “Код программы. Часть 2”

Скомпилируем файл командной в Командной строке:

```
Выбрать Командная строка
Microsoft Windows [Version 10.0.19045.3803]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Оля>cd Documents
C:\Users\Оля\Documents>cd Samsonov
C:\Users\Оля\Documents\Samsonov>julia lab02.jl
result.u = [2.4893617021276593, 2.522354396224884, 2.73738368566027, 3.0969406153140655, 3.560258549193575, 4.1727113827
83983, 4.961371376878123, 5.981214248254182, 7.29006326809691, 8.970340533823622, 11.122689783435387, 13.879438644913527
, 14.524330257650655]
result.t = [0.0, 0.04690276000321975, 0.3383342227923363, 0.7779655452294407, 1.2746162290903136, 1.840070740934981, 2.4
567628694601207, 3.1227047420341996, 3.827643913525675, 4.566506139200148, 5.332627437045586, 6.1213973443435155, 6.2831
85307179586]
```

Рис. 6: “Компиляция программы lab02.jl”

Результаты работы кода на Julia

На рис. @fig:008 и @fig:009 изображены итоговые графики траектории движения катера и лодки для случая обоих случаев.

Задача о погоне - случай 1

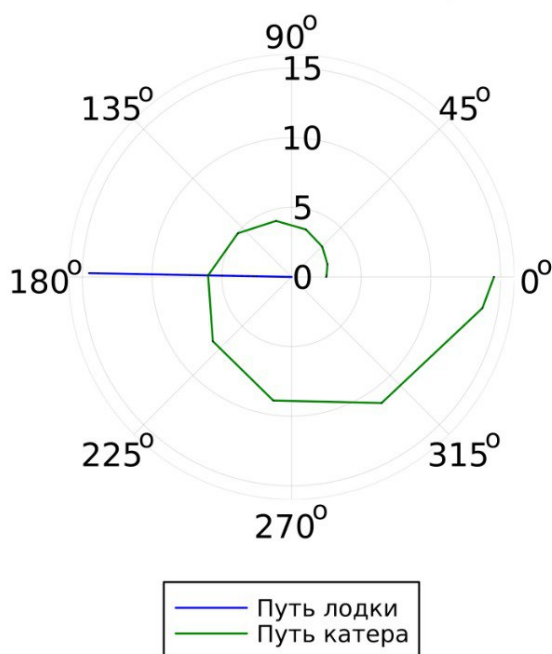


Рис. 7: “Полученные результаты. Первый случай”

Задача о погоне - случай 2

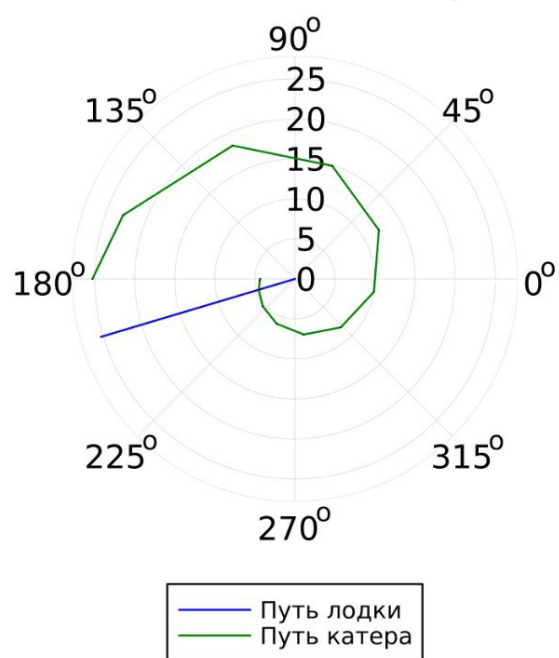


Рис. 8: “Полученные результаты. Второй случай”

Анализ итоговых результатов и вывод

В данной лабораторной работе №2 мы потроили графики для обоих случаев, где отрисованы траектории лодки и катера и наглядно показали пересечения. Таким образом, успешно решили поставленную задачу.

Также изучили основы языков программирования Julia и освоили библиотеки данного языка, которые необходимы для построения графиков и решение дифференциальных уравнений. Однако из-за отсутствия возможности работы языка OpenModeica с полярными координатами мы не смогли приобрести опыт.

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/>