

Отчёт по лабораторной работе №2

дисциплина: Математическое моделирование

Абрамян Артём Арменович

Содержание

1	Цель работы	3
2	Теория	4
3	Задание	6
4	Выполнение лабораторной работы	7
5	Выводы	9
6	Библиографический список	10

1 Цель работы

В данной лабораторной работе мне было необходимо решить математическую задачу с помощью языка программирования julia.

2 Теория

- julia Язык программирования Julia разрабатывается с 2009 года в Массачусетском технологическом институте (MIT). Распространяется бесплатно по лицензии MIT. Официальный сайт проекта: <https://julialang.org/>. Все исходные тексты размещены в интернете на GitHub: <https://github.com/JuliaLang/julia>. Ключевые идеи языка изложены его авторами в статьях [1, 2]. Достоинства. Наличие компилятора позволяет создавать программы, быстродействие которых сопоставимо с быстродействием программ, написанных на C, Fortran. Исходный текст общедоступен и распространяется бесплатно. Язык кросс-платформенный. Большая часть Julia написана на Julia. Язык очень гибкий, что облегчает реализацию алгоритмов. Синтаксис Julia похож на синтаксис Matlab и Python, что облегчает перенос программ с одного языка на другой. Обеспечивается поддержка параллельных вычислений. В языке присутствуют широкие возможности метапрограммирования, благодаря чему можно написать программу, которая сгенерирует программу, которая будет выполняться в среде Julia. Язык очень удобен для реализации численных методов с использованием готовых библиотек (линейная алгебра, линейная и нелинейная оптимизация, с ограничениями и без них). Обеспечивается возможность использования прикладных библиотек, созданных для Python. Наконец, язык Julia достаточно прост для изучения. Недостатки. Язык относительно молодой, поэтому возможны изменения, число прикладных библиотек не так велико, как для Python, учебников

немного, на русском языке почти нет вообще. Время компиляции может быть ощутимым. Сторонние прикладные библиотеки не всегда до конца отлажены. Прикладные библиотеки время от времени изменяются, и тексты программ, которые используют эти библиотеки иногда перестают работать, поскольку, например, изменилось название одной из функций. Ситуация с быстродействием программ на языке Julia не столь однозначна. На сайте <https://docs.julialang.org/en/v1/manual/performance-tips/index.html> приводится довольно большой текст, посвященный тому, как повысить эффективность работы программы. Из текста можно понять, что быстродействие — это возможность, которой нужно еще суметь воспользоваться.

3 Задание

1. Провести аналогичные рассуждения и вывод дифференциальных уравнений, если скорость катера больше скорости лодки в n раз (значение n задайте самостоятельно)
2. Построить траекторию движения катера и лодки для двух случаев. (Задайте самостоятельно начальные значения) Определить по графику точку пересечения катера и лодки.

4 Выполнение лабораторной работы

1. Вывели необходимые значения для решения задачи. (рис. 4.1)

```
function F(u, p, t)
    # p и t не используются в нашем примере
    # но нужны в любом случае для библиотеки
    return u / √(23.01)
end

"Начальное значение"
const u_0 = 17.4
```

Рис. 4.1: Необходимые значения

2. Написали код на ЯП julia для расчёта траектории и построения графика.
(рис. 4.2)

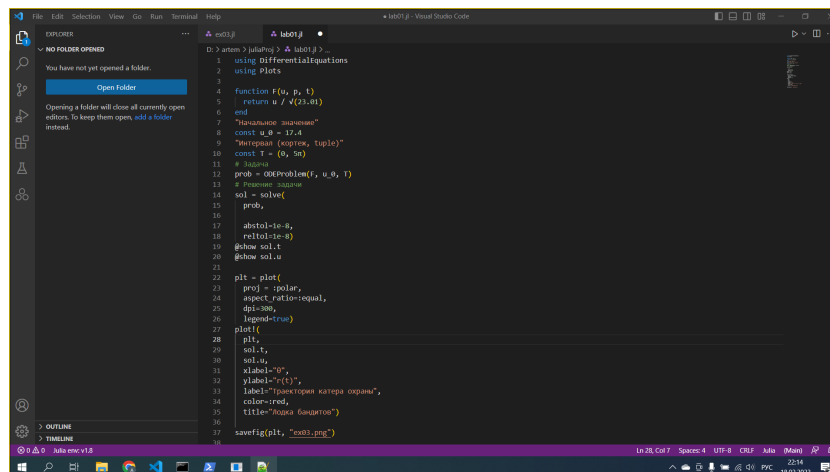


Рис. 4.2: Код программы

3. Получили траекторию движения катера охраны. (рис. 4.3)

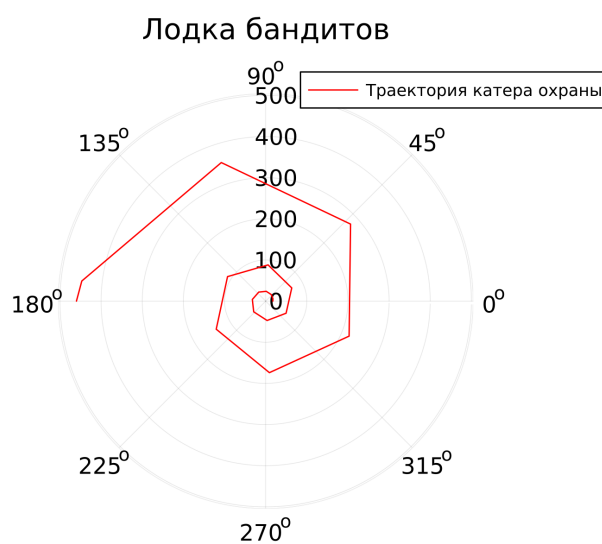


Рис. 4.3: Траектория катера охраны

5 Выводы

В данной лабораторной работе мне успешно удалось решить математическую задачу с помощью языка программирования julia.

6 Библиографический список

1. Документация julia (<https://julialang.org/>)