Лабораторная работа №2
Метод стрельбы

Колобаев Дмитрий
Группа Б01-909

Задание

XI.9.3. Построить алгоритм метода пристрелки для вычисления решения следующих нелинейных задач:

а) $\begin{cases} y'' - x\sqrt{y} = 0, & 0 \le x \le 1, \\ y(0) = 0, \ \ y(1) = 2; \end{cases}$

б) $\begin{cases} y'' - x\sqrt{y} = 0, & 0 \le x \le 1, \\ y(0) = 0, \ \ \int\limits_0^1 y(x)dx = 1. \end{cases}$

Метод решения

Используем метод стрельбы. Для начала сведем уравнение к системе ОДУ первого прядка.

$$\begin{cases} p' = x\sqrt{y} \\ y' = p \end{cases}$$

Параметризуем начальные условия на $y(0) = 0, p = \alpha$. И решим полученную задачу Коши. Для решения используем метод Рунге-Кутты второго порядка - "метод Эйлера с пересчетом". Полученное решение подставим в граничные условия и проверим, что $y(1) - 2 = 0$. В итоге получим нелинейное уравнение $y(1, \alpha) = 0$ относительно параметра α, где $y(1, \alpha)$ это значение решения задачи Коши в точке 1 при данном параметре α.

Решим полученное нелинейное уравнение на α с помощью метода Ньютона. Значение производной в точке для метода Ньютона будем получать численно по формуле первого аорядка.

$$F'(\alpha) = \frac{F(\alpha + h) - F(\alpha)}{h}$$

Во варианте *б* для интегрирования будем использовать метод прямоугольников.

Результаты

Результаты вычисления приведены на графиках. Как видно в первом варианте в точке $x = 1$ выполняется граничное условие $y(1) = 2$. Во втором варианте интеграл $\int_0^1 y(x)dx$ можно оценить как площадь прямоугольного треугольника с катетами 1 и 2. Получается интеграл равен $\int_0^1 y(x)dx \approxeq 1$, как и требовалось в начальных условиях. Итоговые значения параметра α:

а) $\alpha = 1.84295$

б) $\alpha = 1.92686$

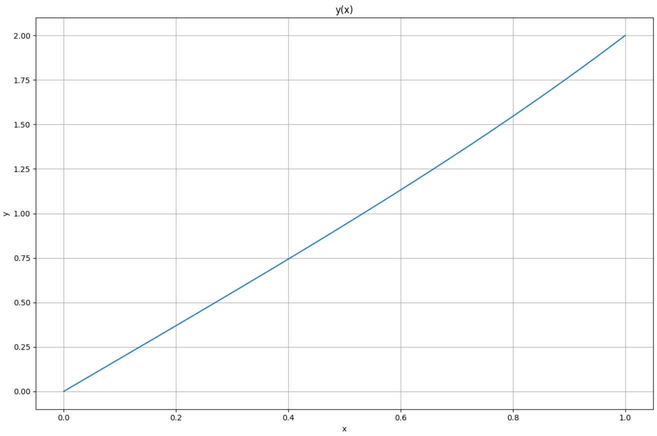

Рис. 1: Вариант А

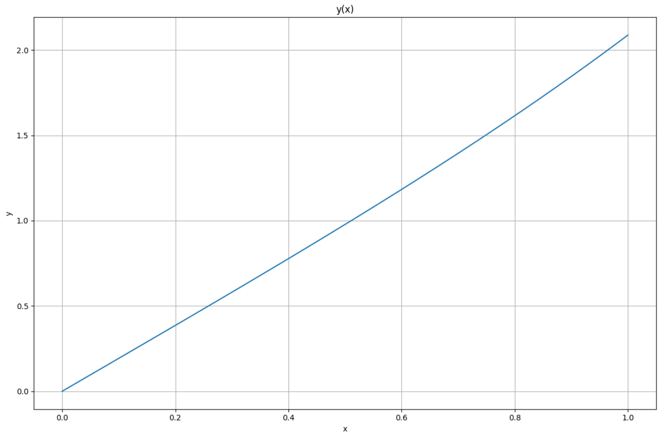

Рис. 2: Вариант Б

Программная реализация

Программа для метода стрельбы реализована в файле main.cpp папки lab3_shootout. Методы Рунге-Кутты и Ньютона взяты из предыдущих лабораторных работ и могу быть найдены в соответсвующих файлах библиотеки Computational-Math: RKS.hpp/cpp и Solvers.hpp/cpp