Санкт-Петербургский политехнический университет
Высшая школа прикладной математики
и вычислительной физики, ФизМех

Направление подготовки
«Прикладная математика и информатика»

Отчет по лабораторной работе №5
«Численнные методы решения задачи Коши для обыкновенных дифференциальных уравнений»

Выполнил студент гр. 5030102/00003 Петрошенко А.В.

Преподаватель: Курц В.В.

Санкт-Петербург
2021

Формулировка задачи и ее формализация

Постановка задачи:
Дифференциальное уравнениеnго порядка

$$F(x, y, y', ..., y^{(n)}) = 0 \tag{1}$$

где $y(x)$ - неизвестная функция
Уравнение (1), разрешенное относительно старшей производной

$$y^{(n)} = f(x, y, y', ..., y^{(n-1)}) \tag{2}$$

Общее решение: $y(x) = y(c_1, ..., c_n)$ - n-параметрическое семейство
Пусть через любую точку проходит единственная интегральная кривая. Чтобы выделить единственное решение, нужно задать n условий.

В данной лабораторной работе будут решаться уравнения 1-го и 2-го порядков
Пусть задача решается на конечном отрезке $[a, b]$ и все условия заданы в одной точке $[a, b] \rightarrow$ задача Коши, или задача с начальными условиями.

Алгоритм метода и условия его применимости

Метод Рунге-Кутты 3-го порядка с коэффициэнтом 1/2
Задача Коши

$$\begin{cases} y' = f(x, y) \\ y(a) = y_0 \end{cases} \tag{3}$$

$x, x + h \in [a, b]$

$$y(x + h) = y(x) + \underbrace{\frac{h}{1!}y'(x) + \frac{h^2}{2!}y''(x) + ... + \frac{h^s}{s!}y^{(s)}(x)}_{\Delta_s y(x)} + O(h^{s+1}) \tag{4}$$

Заменим $\Delta_s y(x)$ некоторой функцией $\delta_s y(x, h)$, которая удовлетворяет условию

$$\Delta_s y(x) = \delta_s y(x, h) + O(h^{s+1}) \tag{5}$$

Будем искать $\delta_s y(x, h)$ в виде линейной комбинации значений f

$$\delta_s y(x, h) = h \sum_{i=1}^{l} \rho_i f(x + \delta x_i, y + \delta y_i) \tag{6}$$

где s - порядок метода, l - шаговость или стадийность метода Подберем l, ρ_i, δx_i и δy_i так, чтобы (5) выполнялось. Выбираем l. Будем искать $\delta_s y(x, h)$ в виде:

$$\delta_s^l y(x, h) = h \sum_{i=1}^{l} \rho_i K_i \tag{7}$$

где

$$\begin{cases} K_1 = f(x, y) \\ K_2 = f(x + \alpha_2 h, y + h\beta_{21}K_1) \\ K_3 = f(x + \alpha_3 h, y + h\beta_{31}K_1 + h\beta_{32}K_2) \\ ... \\ K_l = f(x + \alpha_l h, y + h\sum_{j=1}^{l-1} \beta_{lj}K_j) \end{cases} \tag{8}$$

Коэффициенты ρ_i, α_i, β_{ij} выбираются из условия (5): обеспечение порядка апроксимации s. Тогда в (4) отбросим $O(h^{s+1})$, $x \to x_k$, $x + h \to x_k + 1$

$$y_{k+1} = y_k + h \sum_{i=1}^{l} \rho_i K_i \tag{9}$$

<u>Алгоритм метода</u>

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6}(K_1 + 4K_2 + K_3) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}K_1) \\ K_3 = f(x_i + h, y_i - hK_1 + 2hK_2) \end{cases} \tag{10}$$

Таким образом в данной работе реализуется 3-х стадийный метод 3-го порядка.
<u>Условия применимости</u>

1. Функция f должна быть достаточное число раз непрерывно дифференцируемой.
2. $x + h \in [a, b]$

Предварительный анализ задачи

Мы строим равномерную сетку для $[a, b]$, поэтому 2-ой пункт выполнен по построению

В данной работе будет рассматриваться ОДУ $xy' + (x+1)y = 3x^2e^{-x}$, значит $f = \dfrac{3x^2e^{-x} - (x+1)y}{x}$, которая бесконечно дифференцируема.

Тестовый пример для задач малой размерности

$f(x, y) = \dfrac{3x^2e^{-x} - (x+1)y}{x}$, точное решение $y^*(x) = x^2e^{-x}$
Начальное условие на $[a, b]$ $y(a) = 1/e$, где $a = 1$, $b = 5$
Возьмем $h = \dfrac{b-a}{2} = 2$ и сделаем 2 шага по h, 1 шаг $2h$ и оценим погрешность с помощью правила Рунге.

$$\epsilon_{k+2}^{(h)} \approx \frac{y_{k+2}^{(2h)} - y_{k+2}^{(h)}}{2^s - 1}$$

<u>1 Шаг</u>

$$\begin{cases} y_b = y_a + \frac{2}{3}(K_1 + 4K_2 + K_3) \\ K_1 = f(x_a, y_a) \\ K_2 = f(x_a + 2, y_a + 2K_1) \\ K_3 = f(x_a + 4, y_a - 4K_1 + 8K_2) \end{cases} \tag{11}$$

$$K_1 = 0.368$$

$$K_2 = -1.024$$

$$K_3 = 11.256$$

$$y_b = 5.386$$

Фактическая погрешность: $|y^*(b) - y_b| = 5.386 - 0.168 = 5.218$
<u>2 Шага</u>

$$\begin{cases} y_{(b-a)/2} = y_a + \frac{2}{3}(K_1 + 4K_2 + K_3) \\ K_1 = f(x_a, y_a) \\ K_2 = f(x_a + 2, y_a + 2K_1) \\ K_3 = f(x_a + 4, y_a - 4K_1 + 8K_2) \end{cases} \tag{12}$$

$$K_1 = 0.368$$

$$K_2 = -0.292$$

$$K_3 = 2.494$$

$$y_{(b-a)/2} = 0.933$$

$$\begin{cases} y_b = y_{(b-a)/2} + \frac{2}{3}(K_1 + 4K_2 + K_3) \\ K_1 = f(x_{(b-a)/2}, y_{(b-a)/2}) \\ K_2 = f(x_{(b-a)/2} + 2, y_{(b-a)/2} + 2K_1) \\ K_3 = f(x_{(b-a)/2} + 4, y_{(b-a)/2} - 4K_1 + 8K_2) \end{cases} \tag{13}$$

$$K_1 = -0.796$$

$$K_2 = 0.048$$

$$K_3 = -3.161$$

$$y_b = -0.321$$

Фактическая погрешность: $|y^*(b) - y_b| = 0.168 + 0.321 = 0.489$

Правило Рунге: $\epsilon_{k+2}^{(h)} \approx \dfrac{y_{k+2}^{(2h)} - y_{k+2}^{(h)}}{2^s - 1} = \dfrac{5.386 + 0.321}{7} = 0.815$

Фактическая погрешность и погрешнсть по правилу Рунге одного порядка, но погрешность по Рунге больше.

Контрольные тесты

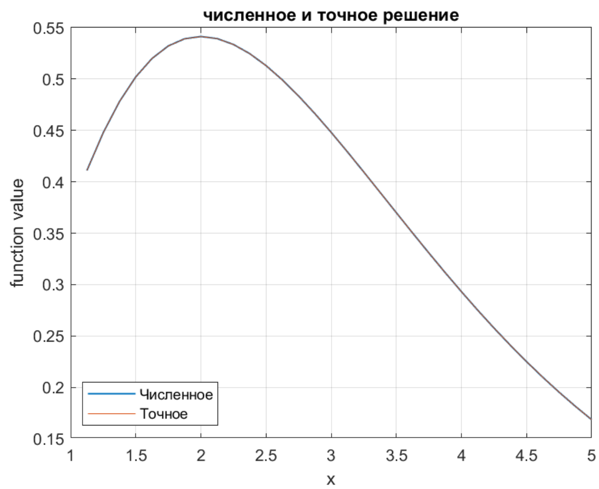
1. Разобъем отрезок $[a, b]$ на 2^5 частей и решим ур-ние нашим методом
2. Будем разбивать наш отрезок на части(от 1 до 2^{15}) и посмотрим на зависимость локальной и глобальной погрешностей
3. Зададим возмущение(порядка от 10^{-10} до 10^0) в начальное условие и посмотрим, как меняется погрешность при шаге $1/2^{10}$

Модульная структура программы

`double f(double x, double y)`
`double y(double x)`
-Функция f и функция точного решения $y = e^{-x}$
`double Calculate(double h, double x_j, double y_j)`
-Функция для вычисления значения y на следующем шаге
`void Error(double a, double f_a, double b, ofstream* F)`
-Функция для вычисления локальной и глобальной погреностей
`vector<double> Runge_Kutta(double a, double f_a, double b, int n)`
-Функция решения ОДУ
`double Pertubation(double a, double f_a, double df_a, double b, int n)`
-Функция решения ОДУ с возмущением

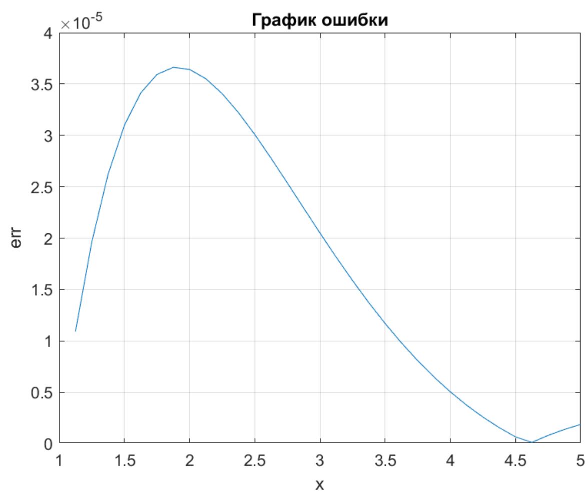
Численный анализ

▷ Точное и численное решение:



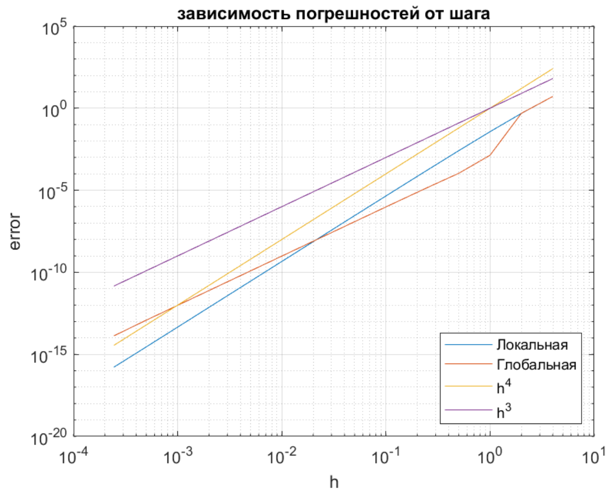
Оба графика накладываются друг на друга, что показывает, что решения практически совпадают.

▷ Ошибка численного решения:



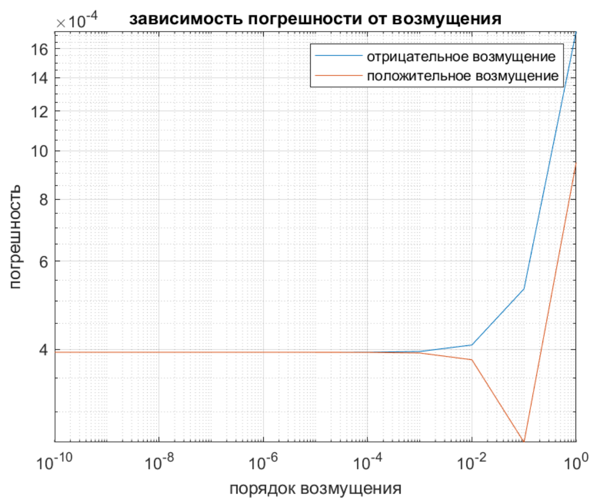
В начальной точке ошибка очевидно нулевая, т.к. мы просто подставили начальное условие. Ошибка ненулевая, порядка 10^-5. На конце происходит небольшой излом, что обуславливается видом интегральной кривой. В какой то момент мы просто перешагнули через нужную кривую и ошибка снова возросла.

▷ Погрешности:



Порядки погрешностей совпадают с установленными методом порядками, что означает, что метод реализован правильно. Погрешности пересекаются, что так же обуславливается видом интегральной кривой.

▷ Возмущение:



Мы рассмотрели 2 возмущения: отрицательное и положительное. При малых возмущениях погрешность не меняется, но начиная с 10^{-3} они отклоняются, что совпадает с порядком погрешности метода. При чем одна уменьшается, а другая увеличивается, для разных кривых это отклонение будет разным, но для достаточно больших возмущений обе погрешности увеличиваются.

Общие выводы

В данной лабораторной работе мы научились численно решать ОДУ 1-го и 2-го порядка на заданном промежутке с помощью метода Рунге-Кутты 3-го порядка с коэффициентом 1/2. Реализация метода очень простая, что видно из модульной структуры. Метод довольно эффективный, но он, как и все остальные, зависит от вида интегральной кривой.