

Российский университет дружбы народов им. П. Лумумбы
Факультет физико-математических и естественных наук

Лабораторная работа №4

Дисциплина: Вычислительные методы

Студент: Шуплецов Александр Андреевич

Группа: НФИбд-01-22

Москва

2024 г.

Оглавление

Задание.....	3
Теоретическая справка.....	4
Полный программный код, с подробным описанием функции, реализующей метод Эйлера.....	5
Определение функций.....	6
Основная часть программы.....	6
Численные расчеты	7
Вывод	8

Задача Коши

1. Реализовать в программе метод Эйлера для численного решения задачи Коши, приведенной ниже:

$$\begin{cases} y' = f(x, y), & x \in [a, b], \\ y(a) = y_0, \end{cases}$$

где функция $f(x, y)$, концы отрезка a, b и начальное значение y_0 заданы в индивидуальном варианте.

2. В программе вывести таблицу данных следующего вида:

$$\begin{array}{cccc} x_0 & \tilde{y}(x_0) & y(x_0) & \delta(x_0) \\ x_1 & \tilde{y}(x_1) & y(x_1) & \delta(x_1) \\ \vdots & \vdots & \vdots & \vdots \\ x_N & \tilde{y}(x_N) & y(x_N) & \delta(x_N) \end{array}$$

где целое число $N = 32$ определяет равномерное разбиение отрезка $[a, b]$, $\tilde{y}(x_j)$ – значение численного решения задачи Коши, полученного методом Эйлера, в узлах разбиения отрезка; $y(x_j)$ – значение аналитического решения задачи Коши в узлах разбиения отрезка; $\delta(x_j) = |\tilde{y}(x_j) - y(x_j)|$ – погрешность.

3. Определить значение N , при котором $\delta(x_j) < 10^{-2}$, $j = \overline{0, N}$.
4. Оформить результаты выполнения лабораторной работы в виде отчета, содержащего:
- титульный лист с указанием фамилии, имени и отчества (при наличии) студента, выполнившего лабораторную работу;
 - содержание, включающее все разделы лабораторной работы;
 - раздел с краткой теоретической справкой о решаемой задаче, включающий постановку задачи, краткое описание численного метода ее решения и все формулы, используемые при расчетах;
 - раздел с полным программным кодом, содержащим подробные комментарии всех написанных функций (либо методов классов);
 - раздел с численными расчетами (достаточно включить скриншот с выводом программы);
 - необходимо, чтобы результаты выполнения пп.2-3 вошли в отчет;
 - отчет загружается на ТУИС в формате pdf.

Я возьму вариант 20:

20.	$f(x, y) = 4x^3 y$	$a = 0, \quad b = 1, \quad y_0 = 1$
-----	--------------------	-------------------------------------

Теоретическая справка

Данная лабораторная работа представляет собой компьютерную реализацию метода Эйлера для численного решения задачи Коши.

Пусть нам нужно построить решение задачи Коши (26), (27) на отрезке $[x_0, x_0 + l]$ длины l .

Возьмем некоторое целое число n , введем шаг $h = l/n$ и образуем на отрезке сетку

$$x_i = x_0 + ih, \quad 0 \leq i \leq n. \quad (28)$$

Сопоставим задаче (26), (27) на отрезке разностную задачу

$$(y_{i+1} - y_i)/h = f(x_i, y_i), \quad 0 \leq i \leq n-1; \quad (29)$$

$$y_0 = u_0. \quad (30)$$

Здесь производная $u'(x)$ в уравнении (26) заменена правой разностной производной и сохранено неизменным начальное условие (27).

Уравнение (29)

$$(y_{i+1} - y_i)/h = f(x_i, y_i), \quad 0 \leq i \leq n-1;$$

является разностным уравнением первого порядка, которое принято называть **схемой Эйлера**.

Его можно переписать в виде рекуррентного соотношения:

$$y_{i+1} = y_i + hf(x_i, y_i), \quad 0 \leq i \leq n-1; \quad (31)$$

Полный программный код, с подробным описанием функции, реализующей метод Эйлера.

```
import math

def func(x,y):
    return 4*(x**3)*y

def splitter(a,b,N):
    step = (b-a)/N
    arr_x = [0]*(N+1)
    for i in range (0,N+1):
        arr_x[i] = a + step*i
    return arr_x

def metod_Eylera(u,h, N):
    arr_x = splitter(a,b,N)
    arr_y = [0]*(N+1)
    arr_y[0] = u
    for i in range (0,N):
        arr_y[i+1] = arr_y[i] + h*func(arr_x[i],arr_y[i])
    return arr_y

def anal_res(N):
    arr_x = splitter(a,b,N)
    arr_y = [0]*(N+1)
    for i in range (0,N+1):
        arr_y[i] = math.e**arr_x[i]**4
    return arr_y

global a, b
a = 0
b = 1
N = 32
h = (b-a)/N
max_delta = 0

print("{:<10}{:<10}{:<10}{:<10}".format("x", "Euler", "Analytic", "Error"))

for i in range (N+1):
    print("{:<10f}{:<10f}{:<10f}{:<10f}".format(splitter(a,b,N)[i],
    metod_Eylera(1,h,N)[i], anal_res(N)[i], abs(anal_res(N)[i] -
    metod_Eylera(1,h,N)[i])))

    if abs(anal_res(N)[i] - metod_Eylera(1,h,N)[i]) > max_delta:
        max_delta = abs(anal_res(N)[i] - metod_Eylera(1,h,N)[i])

print("Макс ошибка: ", max_delta)

for n in range(1, 10000):
    h = (b-a)/n
    max_delta = 0
    resh = anal_res(n)
    metod_Eylera_cur = metod_Eylera(1,h,n)
    for i in range (n+1):
        if abs(resh[i] - metod_Eylera_cur[i]) > max_delta:
            max_delta = abs(resh[i] - metod_Eylera_cur[i])
```

```

if max_delta < 0.01:
    print()
    print("N, при котором максимальная ошибка меньше 0.01")
    print(n, " ", max_delta)
    break

```

Определение функций

1. **func(x, y):**
 - Это функция, определяющая производную для нашего дифференциального уравнения.
2. **splitter(a, b, N):**
 - Функция делит интервал $[a, b]$ на (N) равных частей.
 - $step = (b-a)/N$: вычисляет шаг, с которым будет изменяться (x) .
 - Возвращает массив arr_x из $(N+1)$ элементов, представляющих узлы сетки на интервале.
3. **metod_Eylera(u, h, N):**
 - Реализует метод Эйлера для численного решения дифференциального уравнения.
 - $arr_x = splitter(a, b, N)$: генерирует массив точек (x) .
 - $arr_y[0] = u$: устанавливает начальное значение для (y) , равное (u) .
 - Цикл вычисляет значения (y) для каждой точки, используя метод Эйлера: $(y_{i+1} = y_i + h \times f(x_i, y_i))$.
 - Возвращает массив значений (y) .
4. **anal_res(N):**
 - Вычисляет аналитическое решение уравнения для каждого (x) из сетки.
 - Использует экспоненциальную функцию: $(y = e^{x^4})$.
 - Возвращает массив значений (y) для аналитического решения.

Основная часть программы

1. Инициализация переменных:
 - a, b : это границы интервала, от 0 до 1.
 - N : Количество сегментов интервала для метода Эйлера.
 - $h = (b-a)/N$: Шаг сетки.
 - max_delta : Переменная для отслеживания максимальной ошибки.
2. Вывод заголовков таблицы:
 - `print("{<10}{<10}{<10}{<10}".format("x", "Euler", "Analytic", "Error"))`:
Форматированная строка, которая печатает заголовки с выравниванием для всех ее столбцов.
3. Цикл для вычисления и вывода значений:
 - Проходит по всем узлам сетки:
 - Вычисляет значения (x) , (y) методом Эйлера, (y) аналитическое и вычисляет их погрешность.
 - `print(...)` выводит значения в строку таблицы.
 - Вычисляется максимальная погрешность (разница).
4. Поиск необходимого n :
 - Перебирает разные значения (n) от 1 до 9999.
 - Проверяет точность численного решения методом Эйлера, пока максимальная ошибка не станет меньше 0.01.
 - Печатает ту величину (n) вместе с достигнутой погрешностью, при которой эта ошибка выполнена, и завершает цикл с помощью `break`.

Этот код помогает сравнить точность численного метода (метод Эйлера) с аналитическим решением и определить минимальное количество узлов сетки, требуемых для достижения заданной точности.

Численные расчеты

x	Euler	Analytic	Error
0.000000	1.000000	1.000000	0.000000
0.031250	1.000000	1.000001	0.000001
0.062500	1.000004	1.000015	0.000011
0.093750	1.000034	1.000077	0.000043
0.125000	1.000137	1.000244	0.000107
0.156250	1.000382	1.000596	0.000215
0.187500	1.000859	1.001237	0.000378
0.218750	1.001683	1.002292	0.000609
0.250000	1.002994	1.003914	0.000920
0.281250	1.004953	1.006277	0.001324
0.312500	1.007748	1.009582	0.001835
0.343750	1.011592	1.014061	0.002469
0.375000	1.016728	1.019972	0.003244
0.406250	1.023430	1.027612	0.004182
0.437500	1.032007	1.037316	0.005308
0.468750	1.042810	1.049464	0.006654
0.500000	1.056236	1.064494	0.008259
0.531250	1.072739	1.082910	0.010171
0.562500	1.092844	1.105296	0.012452
0.593750	1.117157	1.132337	0.015180
0.625000	1.146387	1.164845	0.018457
0.656250	1.181372	1.203786	0.022414
0.687500	1.223108	1.250326	0.027218
0.718750	1.272789	1.305880	0.033091
0.750000	1.331864	1.372188	0.040324
0.781250	1.402099	1.451401	0.049302
0.812500	1.485670	1.546209	0.060539
0.843750	1.585280	1.660007	0.074727
0.875000	1.704310	1.797113	0.092803
0.906250	1.847030	1.963082	0.116052
0.937500	2.018871	2.165121	0.146250
0.968750	2.226809	2.412680	0.185872
1.000000	2.479871	2.718282	0.238411
Макс ошибка: 0.2384106857158903			

N, при котором максимальная ошибка меньше 0.01
851 0.009993873982764523

Вывод

Я реализовал метод Эйлора с помощью компьютерной программы на языке Python.