Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 6

Тема: построение и программная реализация алгоритмов численного дифференцирования.

Студент Тартыков Л.Е.

Группа ИУ7-44Б

Оценка (баллы) _______________

Преподаватель Градов В.М.

2021 г.

Содержание

Цель работы: получение навыков построения алгоритма вычисления производных от сеточных функций .

Задание

Задана табличная (сеточная) функция. Имеется информация, что закономерность, представленная этой таблицей, может быть описана формулой

$$y=\frac{a_0 x}{a_1+a_2 x}$$

параметры функции неизвестны и определять их не нужно.

x	y	1	2	3	4	5
1	0.571					
2	0.889					
3	1.091					
4	1.231					
5	1.333					
6	1.412					

Вычислить первые разностные производные от функции и занести их в столбцы (1)-(4) таблицы:

1 - односторонняя разностная производная ,

2 - центральная разностная производная,

3- 2-я формула Рунге с использованием односторонней производной,

4 - введены выравнивающие переменные.

В столбец 5 занести вторую разностную производную.

Код программы

Замечание: данная программа написана на языке Python 3.8

Листинг 1; main.py

```
from math import pow

class Differentiation:
    def __init__(self, x, y, h):
        """
        Конструктор
        """
        self.h = h
        self.x = x
        self.y = y
        self._len_x = len(x)
        self._len_y = len(y)
        self.diff_y_1 = [0 for i in range (self._len_x)]
        self.diff_y_2 = [0 for i in range (self._len_x)]
        self.diff_y_3 = [0 for i in range (self._len_x)]
        self.diff_y_4 = [0 for i in range (self._len_x)]
        self.diff_y_5 = [0 for i in range (self._len_x)]

    def calculate_diffenrece(self):
        self.left_differnece()
        self.center_diffenerce()
        self.second_runge_diffenrence(1)
        self.aligned_variable()
        self.second_left_diffenrece()

    def left_differnece(self):
        """
        Левая разностная производная
        """
        self.diff_y_1[0] = 0
        for i in range(1, self._len_y):
            self.diff_y_1[i] = (self.y[i] - self.y[i - 1]) / self.h

    def center_diffenerce(self):
        """
        Центральная разностная производная
        """
        self.diff_y_2[0] = self.diff_y_2[-1] = 0
        for i in range(1, self._len_y - 1):
            self.diff_y_2[i] = (self.y[i + 1] - self.y[i - 1]) / 2 * self.h

    def second_runge_diffenrence(self, p):
        """
        Вторая формула Рунге с использованием односторонней производной
        """
        y_temp = [0, 0]
```

```
        for i in range(2, self._len_y):
            y_temp.append((self.y[i] - self.y[i - 2]) / (2 / self.h))

        self.diff_y_3[0] = self.diff_y_3[1] = 0
        for i in range(2, len(self.diff_y_1)):
            self.diff_y_3[i] = self.diff_y_1[i] + (self.diff_y_1[i] - y_temp[i]) / (pow(2, p) - 1)

    def aligned_variable(self):
        """
        Выравнивающие переменные
        """
        for i in range(self._len_y - 1):
            k = pow(self.y[i], 2) / pow(self.x[i], 2)
            self.diff_y_4[i] = k * (((-1 / self.y[i + 1]) - (-1 / self.y[i])) / ((-1 / self.x[i + 1]) - (-1 / self.x[i])))
        self.diff_y_4[-1] = 0

    def second_left_diffenrece(self):
        """
        Вторая одностороняя производная (левая)
        """
        self.diff_y_5[0] = self.diff_y_5[-1] = 0
        for i in range(1, self._len_y - 1):
            self.diff_y_5[i] = (self.y[i - 1] - 2 * self.y[i] + self.y[i + 1]) / pow(self.h, 2)

    def output(self):
        """
        Вывод таблицы производных на экран
        """
        print("  x   |   y  |      1     |      2     |      3     |      4     |      5")
        print("-" * 70)
        for i in range(self._len_x):
            print("   {0}  |{1:7.3f}|{2:10.3f}|{3:10.3f}|{4:10.3f}|{5:10.3f}|{6:10.3f}". format(self.x[i], self.y[i], \
                    self.diff_y_1[i],  self.diff_y_2[i], self.diff_y_3[i], self.diff_y_4[i], self.diff_y_5[i]))


if __name__ == "__main__":
    h = 1
    x = [1, 2, 3, 4, 5, 6]
    y = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]
    object = Differentiation(x, y, h)
    object.calculate_diffenrece()
    object.output()
```

Результаты работы

Заполненная таблица с краткими комментариями по поводу использованных формул их точности

x	y	1	2	3	4	5
1	0.571	0	0	0	0.408	0
2	0.889	0.318	0.260	0	0.247	-0.116
3	1.091	0.202	0.171	0.144	0.165	-0.062
4	1.231	0.140	0.121	0.109	0.118	-0.038
5	1.333	0.102	0.090	0.083	0.089	-0.023
6	1.412	0.079	0	0.068	0	0

где 1 - односторонняя разностная производная (не можем вычислить первое значение сетки)

$$y'_n=\frac{y_n-y_{(n-1)}}{h}+O(h) \quad ,$$

2 - центральная разностная производная (не можем вычислить первое и последнее значения сетки)

$$y'_n=\frac{y_{(n+1)}-y_{(n-1)}}{2h}+O(h^2) \quad ,$$

3- 2-я формула Рунге с использованием односторонней производной (не можем вычислить первое и второе значения сетки)

$$y'_0=\frac{-3y_0+4y_1-y_2}{2h}+O(h^2) \quad ,$$

4 - введены выравнивающие переменные (не можем вычислить последнее значение сетки).

$$yx'=\frac{\eta_\xi'\xi_x'}{\eta_y'}=\frac{\eta_\xi' y^2}{x^2}$$

5 - вторая разностная производная (не можем вычислить первое и последнее значения сетки); обладает достаточно большой погрешностью

$$y''_n=\frac{y_{(n+1)}-2y_n+y_{(n-1)}}{h^2}+O(h^2)$$

Ответы на вопросы при защите лабораторной работы

1. Получить формулу порядка точности O(h^2) для первой разностной производной y'$_N$ в крайнем правом узле x$_N$.

$$\begin{cases} y_{(n-1)} = y_n - \frac{h}{1!} y'_n + \frac{h^2}{2!} y''_n + O(h^3) \\ y_{(n-2)} = y_n - \frac{2h}{1!} y'_n + \frac{(2h)^2}{2!} y''_n + O(h^3) \end{cases}$$

Первую строку системы домножаем на 4 и вычитаем вторую

$$4 y_{(n-1)} - y_{(n-2)} = 3 y_n - 2 h y'_n + O(h^3) ¿$$

$$y'_n = \frac{3 y_n - 4_{(n-1)} + y_{(n-2)}}{2h} + O(h^2) ¿$$

2. Получить формулу порядка точности O(h^2) для второй разностной производной y''$_0$ в крайнем левом узле x$_0$.

Используем разложение в ряд Тейлора

$$\begin{cases} y_1 = y_0 - \frac{h}{1!} y'_0 + \frac{h^2}{2!} y''_0 + \frac{h^3}{3!} y'''_0 + O(h^4) \\ y_2 = y_0 - \frac{2h}{1!} y'_0 - \frac{(2h)^2}{2!} y''_0 + \frac{(2h)^3}{3!} y'''_0 + O(h^4) \\ y_3 = y_0 - \frac{3h}{1!} y'_0 - \frac{(3h)^2}{2!} y''_0 + \frac{(3h)^3}{3!} y'''_0 + O(h^4) \end{cases}$$

Получим выражение для y''_0:

$$y''_0 = \frac{-y_3 + 4 y_2 - 5 y_1 + 2 y_0}{h^2} + O(h^2)$$

3. Используя 2-ую формулу Рунге, дать вывод выражения (9) из Лекции №7 для первой производной y'$_0$ в левом крайнем узле

$$y'_0 = \frac{-3 y_0 + 4 y_1 - y_2}{2h} + O(h^2)$$

Используем вторую формулу Рунге :

$$\Omega=\Phi(h)+\frac{\Phi(h)-\Phi(mh)}{m^p-1}+O(h^{(p+1)})$$

Левая крайняя производная :

$$y_0{}'=\frac{y_1-y_2}{h}+O(h)\Rightarrow p=1$$

Тогда имеем :

$$\Phi(h)=\frac{y_1-y_0}{h}$$

$$\Phi(mh)=\frac{y_m-y_0}{mh}\Rightarrow\Phi(2h)=\frac{y_2-y_0}{2\lambda}$$

$$y'_0=\Phi(h)+\Phi(h)-\Phi(2h)+O(h^2)=\frac{-y_2+4y_1-3y_0}{2h}+O(h^2)$$

$$y_0{}'=\frac{-3y_0+4y_1-y_2}{2h}+O(h^2)$$

4. Любым способом из Лекций №7, 8 получить формулу порядка точности O(h^3) для первой разностной производной y'$_0$ в крайнем левом узле x$_0$.

Используем вторую формулу Рунге :

$$\Omega=\Phi(h)+\frac{\Phi(h)-\Phi(mh)}{m^p-1}$$

$$y_0{}'=\frac{-3y_0+4y_1-y_2}{2h}+O(h^2)\Rightarrow p=2$$

Тогда имеем :

$$\Phi(h)=\frac{-3y_0+4y_1-y_2}{h}$$

$$\Phi(2h)=\frac{-3y_0+4y_2-y_4}{4h}$$

$$y'_0=\Phi(h)+\frac{\Phi(h)-\Phi(2h)}{3}+O(h^3)=\frac{y_4-12y_2+32y_1-21y_0}{12h}+O(h^3)$$

$$y_0{}'=\frac{-21y_0+32y_1-12y_2+y_4}{12h}+O(h^3)$$