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 Варламова Е.А.

Группа ИУ7-41Б

Оценка (баллы) _______________

Преподаватель Градов В.М.

Москва.
2021 г

Цель работы. Получение навыков построения алгоритма вычисления производных от сеточных функций.

1. **Исходные данные**

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.

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					

Задание:

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.

5 - вторая разностная производная.

2. Код программы

```
def left(y, h, i):
    return (y[i] - y[i - 1]) / h if i > 0 else "-"

def right(y, h, i):
    return (y[i + 1] - y[i]) / h if i < len(y) - 1 else "-"

def center(y, h, i):
    return (y[i + 1] - y[i - 1]) / 2 / h if i < len(y) - 1 and i > 0 else "-"

def left_double(y, h, i):
    return (y[i] - y[i - 2]) / 2 / h if i > 1 else "-"
def runge_left(y, h, i):
    if i < 2:
        return "-"
    f1 = left(y, h, i)
    f2 = left_double(y, h, i)
    return f1 + (f1 - f2)

def align_vars_right(y, h, i):
    if i > len(y) - 2:
        return "-"
    der = (1 / y[i + 1] - 1 / y[i]) / (1 / x[i + 1] - 1 / x[i])
    return der * y[i] * y[i] / x[i] / x[i]

def second_der(y, h, i):
    return (y[i - 1] - 2 * y[i] + y[i + 1]) / (h * h) if i < len(y) - 1 and i >
0 else "-"

x = [1, 2, 3, 4, 5, 6]
y = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]

methods = [left, center, runge_left, align_vars_right, second_der]
print("x---------x---------x---------x---------x---------x---------x--------x")
print("|    x    |    y    |   left  | center  |  runge  |  align  | second |")
print("x---------x---------x---------x---------x---------x---------x--------x")
for i in range(len(x)):
    print("|{:9.3f}|".format(x[i]), end = "")
    print("{:9.3f}|".format(y[i]), end = "")
    for func in methods:
        res = func(y, 1, i)
        if res == '-':
            print("    -    |", end = "")
        else:
            print("{:9.4f}|".format(res), end = "")
    print()
print("x---------x---------x---------x---------x---------x---------x--------x")
```

3. Результат работы программы

Результат работы программы:

```
======= RESTART: /Users/kate/Desktop/comp_algs/labs/lab_06/src/lab_06.py
x---------x---------x---------x---------x---------x---------x---------x
|    x    |    y    |  left   | center  |  runge  |  align  | second  |
x---------x---------x---------x---------x---------x---------x---------x
|    1.000|    0.571|    -    |    -    |    -    |   0.4085|    -    |
|    2.000|    0.889|   0.3180|   0.2600|    -    |   0.2469|  -0.1160|
|    3.000|    1.091|   0.2020|   0.1710|   0.1440|   0.1654|  -0.0620|
|    4.000|    1.231|   0.1400|   0.1210|   0.1090|   0.1177|  -0.0380|
|    5.000|    1.333|   0.1020|   0.0905|   0.0830|   0.0895|  -0.0230|
|    6.000|    1.412|   0.0790|    -    |   0.0675|    -    |    -    |
x---------x---------x---------x---------x---------x---------x---------x
```

Комментарии по поводу использованных формул и их точности:

1. *Левая разностная производная (столбец left)*

Формула:

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

Точность: первый порядок точности относительно шага h.

Комментарий: получается из разложения функции в ряд Тейлора

2. *Центральная разностная производная (столбец center)*

Формула:

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

Точность: второй порядок точности относительно шага

Комментарий: получается из разности двух разложениий функции в ряд Тейлора:

$$y_{n+1} = y_n + \frac{h}{1!}y'_n + \frac{h^2}{2!}y_n'' + \frac{h^3}{3!}y_n''' + \frac{h^4}{4!}y_n^{IV} + ...$$

$$y_{n-1} = y_n - \frac{h}{1!}y'_n + \frac{h^2}{2!}y_n'' - \frac{h^3}{3!}y_n''' + \frac{h^4}{4!}y_n^{IV} - ....$$

3. *2-я формула Рунге с использованием односторонней производной (столбец runge)*

Формула:

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

Точность: формула Рунге позволяет за счет расчета на двух сетках с отличающимися шагами получить решение с более высокой точностью, чем заявленная теоретическая точность используемой формулы, соотвественно, если точность Ф = p, то точность формулы будет p + 1.

Комментарий: так как была использована формула Рунге для левой разностной производной, то в формуле m = 2 (удвоенный шаг), а p = 1, поэтому:

res = left(h) + left(h) – left(2*h)

4. *Введение выравнивающих переменных (столбец align)*

Так как в условии сказано, что сеточная функция может быть описана следующей зависимостью:

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

то целесообразно ввести такие выравнивающие переменные:

$$\eta(y) = 1 / y \qquad \xi(x) = 1 / x$$

Действительно, тогда указанная зависимость принимает вид:

$$\eta(\xi) = \xi \frac{a1}{a0} + \frac{a2}{a0}$$

А смысл выравнивающих переменных как раз и состоит в том, чтобы исходная кривая была преобразована в прямую линию, производная от которой вычисляется точно по самым простым формулам.

Для возврата к исходным переменным используется формула:

$$y'_x = y'_\eta \, \eta'_\xi \, \xi'_x = \frac{\eta'_\xi \, \xi'_x}{\eta'_y}$$

Окончательно, формула принимает вид (используется правая разностная производная):

res = right(1 / x[i] - 1 / x[i - 1]) * y[i]^2 / x[i]^2, где

right(1 / x[i] - 1 / x[i - 1]) = (1 / y[i + 1] - 1 / y[i]) / (1 / x[i + 1] - 1 / x[i])

Точность: формула абсолютно точная

5. *Вторая разностная производная (столбец second)*

Формула:

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

Точность: второй порядок точности относительно шага

Комментарий: получается из разложения функции в ряд Тейлора

4. Ответы на вопросы

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

$$y_{n-1} = y_n - \frac{h}{1!} y_n' + \frac{h^2}{2!} y_n'' \ldots (1)$$

$$y_{n-2} = y_n - \frac{2h}{1!} y_n' + \frac{(2h)^2}{2!} y_n'' \ldots (2)$$

$$4*(1) - (2) => 4y_{n-1} - y_{n-2} = 3y_n - 2h y_n' + O(h^2)$$

$$y_n' = \frac{-4y_{n-1} + y_{n-2} + 3y_n}{2h}$$

2. Получить формулу порядка точности O(h^2) для второй разностной производной y''_0 в крайнем левом узле x_0.

$$y_1 = y_0 + \frac{h}{1!} y_0' + \frac{h^2}{2!} y_0'' + \frac{h^3}{3!} y_0''' + \ldots (1)$$

$$y_2 = y_0 + \frac{2h}{1!} y_0' + \frac{(2h)^2}{2!} y_0'' + \frac{(2h)^3}{3!} y_0''' + \ldots (2)$$

$$8*(1) - (2) => 8y_1 - y_2 = 7y_0 + 6h y_0' + 2h^2 y_0''$$

$$y_0'' = \frac{8y_1 - y_2 - 7y_0 - 6h y_0'}{2h^2} = \frac{8y_1 - y_2 - 7y_0 - 6h\left(\frac{-3y_0 + 4y_1 - y_2}{2h}\right)}{2h^2} =$$

$$y_0'' = \frac{8y_1 - y_2 - 7y_0 + 9hy_0 - 12hy_1 + 6y_2h}{2h^2} + O(h^2)$$

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

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

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

$$\varphi(2h) = \frac{y_2 - y_0}{2h}$$

$$\Omega = \frac{y_1 - y_0}{h} + \frac{\frac{y_1 - y_0}{h} - \frac{y_2 - y_0}{2h}}{2^1 - 1} + O(h^2) \Rightarrow$$

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

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.

$$y_1 = y_0 + \frac{h}{1!} y_0' + \frac{h^2}{2!} y_0'' + \frac{h^3}{3!} y_0''' + \ldots \quad (1)$$

$$y_2 = y_0 + \frac{2h}{1!} y_0' + \frac{(2h)^2}{2!} y_0'' + \frac{(2h)^3}{3!} y_0''' + \ldots \quad (2)$$

$$y_3 = y_0 + \frac{3h}{1!} y_0' + \frac{(3h)^2}{2!} y_0'' + \frac{(3h)^3}{3!} y_0''' + \ldots \quad (3)$$

$$18 \cdot (1) - 9 \cdot (2) + 2(3) \Rightarrow$$

$$18y_1 - 9y_2 + 2y_3 = 11y_0 + 6h y_0'$$

$$y_0' = \frac{18y_1 - 9y_2 + 2y_3 - 11y_0}{6h}$$