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 Климов И.С.

Группа ИУ7-42Б

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 **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.

2. Код программы

Листинг 1. main.py

```python
import tkinter as tk


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


def left_side(y, h, i):
    if i > 0:
        return (y[i] - y[i - 1]) / h
    else:
        return None


def center_side(y, h, i):
    if 0 < i < len(y) - 1:
        return (y[i + 1] - y[i - 1]) / (2 * h)
    else:
        return None


def runge(y, h, i):
    if i > 1:
        return 2 * left_side(y, h, i) - (y[i] - y[i - 2]) / (2 * h)
    else:
        return None
```

```
def align_vars(x, y, i):
    if i < len(y) - 1:
        tmp = (1 / y[i + 1] - 1 / y[i]) / (1 / x[i + 1] - 1 / x[i])
        return tmp * y[i] ** 2 / x[i] ** 2
    else:
        return None


def second(y, h, i):
    if 0 < i < len(y) - 1:
        return (y[i - 1] - 2 * y[i] + y[i + 1]) / h ** 2
    else:
        return None


def get_result():
    result = [[x, y] for x, y in zip(x_init, y_init)]
    length = len(x_init)
    methods = [left_side, center_side, runge, align_vars, second]
    for i in range(length):
        for method in methods:
            if method != align_vars:
                value = method(y_init, x_init[1] - x_init[0], i)
            else:
                value = align_vars(x_init, y_init, i)

            result[i].append(value)

    return result


def show_table(table):
    root = tk.Tk()
    rows, columns = len(table), len(table[0])
    names = ['x', 'y', '1', '2', '3', '4', '5']
    for i in range(len(names)):
        tk.Label(root, text=names[i], width=10, font='Times 16',
                 borderwidth=2, relief='ridge').grid(row=0, column=i)

    for i in range(rows):
        tk.Label(root, text=f'{table[i][0]}', width=10, font='Times 16',
                 borderwidth=2, relief='ridge').grid(row=i + 1, column=0)
        for j in range(1, columns):
            tk.Label(root, text=f'{table[i][j]:.3f}' if table[i][j] else '-',
width=10, font='Times 16',
                     borderwidth=2, relief='ridge').grid(row=i + 1, column=j)

    root.mainloop()


def main():
    table = get_result()
    show_table(table)


if __name__ == '__main__':
    main()
```

3. Результаты работы

x	y	1	2	3	4	5
1	0.571	-	-	-	0.408	-
2	0.889	0.318	0.260	-	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	-	0.068	-	-

1) Первый столбец – левосторонняя формула (точность – O(h))

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

Получается путем разложения функции в ряд Тейлора для точки, в которой хотим найти производную. Затем выражаем значение через этот ряд для следующей точки и выражаем первую производную от него.

$$y_{n+1} = y_n - \frac{h}{1!} y'_n + \frac{h}{2} y''_n - \cdots$$

2) Второй столбец – центральная формула (точность – O(h^2)).

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

Получается при $f_{n+1} - f_{n-1}$, где f – ряд Тейлора, построенный для точки, в которой требуется найти производную.

3) Третий столбец – вторая формула Рунге (точность – O(h^2)).

Для вывода используется левосторонняя формула.

4) Четвертый столбец – применение выравнивающих переменных (невозможно точно оценить погрешность, так как параметры неизвестны).

Использовалось следующее соотношение:

$$y'_x = \frac{\eta'_\varepsilon \varepsilon'_x}{\eta'_y} = \frac{y^2}{x^2} \left(\frac{\frac{1}{y_{n+1}} - \frac{1}{y_n}}{\frac{1}{x_{n+1}} - \frac{1}{x_n}} \right)$$

5) Пятый столбец – вторая разностная производная (точность – O(h^2)).

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

Получается путем разложения функции в ряд Тейлора. При этом проводятся преобразования: из ряда Тейлора для точки, в которой надо найти производную, выражаем следующую и предыдущую и складываем получившиеся значения, выразив y``.

4. 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.

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

$$\Phi(h) = \frac{y_{n+1} - y_n}{h}$$

$$\Phi(mh) = \frac{y_{n+m} - y_n}{mh}$$

m = 2

$$\Phi(2h) = \frac{y_{n+2} - y_n}{2h}$$

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

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

2) Получить формулу порядка точности O(h^2) для второй разностной производной y\`\`$_0$ в крайнем правом узле x_0.

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

$$\Phi(h) = \frac{y_0 - 2y_1 + y_2}{h^2}$$

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

$$y_0'' = 2\Phi(h) - \Phi(2h) + O(h^2)$$

$$y_0'' = \frac{2y_0 - 4y_1 + 2y_2}{h^2} - \frac{y_0 - 2y_2 + y_4}{4h^2} + O(h^2) = \frac{7y_0 - 8y_1 + 10y_2 - y_4}{4h^2} + O(h^2)$$

3) Используя 2-ую формулу Рунге, дать вывод выражения для первой производной y\`$_0$ в левом крайнем узле.

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

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

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

$$y_0' = 2\Phi(h) - \Phi(2h) + O(h^2)$$

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

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$_0$.**

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

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

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

$$y_0' = \Phi(h) + \frac{\Phi(h) - \Phi(mh)}{m^p - 1} + O(h^{p+1})$$

$p = 2$

$$y_0' = \frac{4\Phi(h) - \Phi(2h)}{3} + O(h^3) = \frac{y_4 - 12y_2 + 16y_1 - 3y_0}{12h} + O(h^3)$$