	Министерство образования и науки 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 (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

Лабораторная работа №6

Тема: Численное дифференцирование

Студент: Артемьев И.О.

Группа: ИУ7-43Б

Оценка (баллы): _________

Преподаватель: Градов В.М.

Москва.
2021 г.

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

Входные данные

Таблица из задания.

Выходные данные

Заполненная таблица.

Описание алгоритма

Используя разложение в ряд Тейлора можно получить левую

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

и правую разностную формулы

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

Данные формулы имеют самый низкий - первый - порядок точности. Из данных формул можно получить центральную разностную формулу

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

Центральная разностная формула имеет второй порядок точности.

Приведенные выше формулы имеют погрешность вида $R = \psi(x)h^p$. С помощью преобразований в рядах Тейлора можно получить первую формулу Рунге

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

Отсюда можно получить вторую формулу Рунге

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

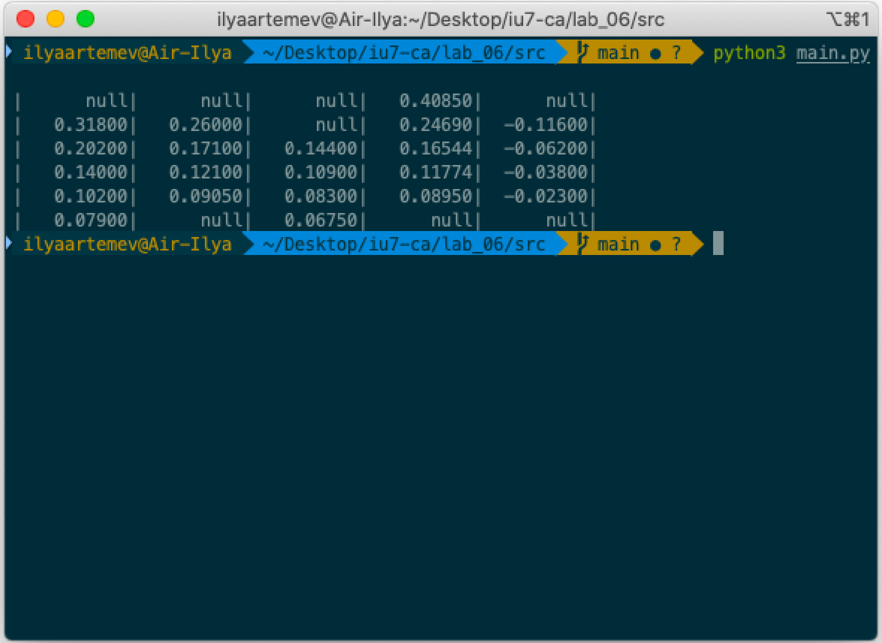
Формулы Рунге справедливы не только для операций дифференцирования, но и для других приближенных вычислений (при условии, что погрешность формул имеет вышеприведенный вид).

Помимо приведенных выше методов стоит отметить метод, заключающийся в применении выравнивающих переменных. При правильном подборе исходная кривая может быть преобразована в прямую, производная от которой вычисляется точно даже по простым формулам.

Пусть задана функция $y(x)$ с введенными переменными $\xi = \xi(x)$ и $\eta = \eta(y)$. Тогда возврат к заданным переменным будет осуществлен по формуле

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

Результаты работы программы



Первый столбец - левосторонняя формула (точность $O(h)$).
Второй столбец - центральная формула (точность $O(h^2)$).
Третий столбец - вторая формула Рунге (с использованием левосторонней формулы).
Четвертый столбец - применение выравнивающих переменных (оценка точность сложна,
так как неизвестны параметры). Использовано соотношение

$$y'_x = \frac{\eta'_\xi y^2}{x^2}$$

Пятый столбец - вторая разностная производная.

Код программы

```
"""
   main.py
"""

from class_numerical_differentiator import NumericalDifferentiator
```

```python
def main():
    x = [1, 2, 3, 4, 5, 6]
    y = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]

    exe = NumericalDifferentiator(x, y)
    exe.printCompletedTable()


if __name__ == "__main__":
    main()


"""
    class_numerical_differentiator.py
"""


class NumericalDifferentiator:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    @staticmethod
    def __left(y, step, ind):
        return (y[ind] - y[ind - 1]) / step if ind > 0 else "%10s" % "null"

    @staticmethod
    def __center(y, step, ind):
        return (
            (y[ind + 1] - y[ind - 1]) / 2 / step
            if ind > 0 and ind < len(y) - 1
            else "%10s" % "null"
        )

    @staticmethod
    def __secondDiff(y, step, ind):
        return (
            (y[ind - 1] - 2 * y[ind] + y[ind + 1]) / step ** 2
            if ind > 0 and ind < len(y) - 1
            else "%10s" % "null"
```

```
    )

@staticmethod
def __alignCoeff(y, x, step, ind):
    if ind > len(y) - 2:
        return "%10s" % "null"

    res = (
        (1 / y[ind + 1] - 1 / y[ind])
        / (1 / x[ind + 1] - 1 / x[ind])
        * y[ind]
        * y[ind]
        / x[ind]
        / x[ind]
    )

    return res

def __rungeLeft(self, y, step, ind):
    if ind < 2:
        return "%10s" % "null"

    f1 = self.__left(y, step, ind)
    f2 = (y[ind] - y[ind - 2]) / 2 / step

    return 2 * f1 - f2

def __getMethodsLinks(self):
    return [
        self.__left,
        self.__center,
        self.__rungeLeft,
        self.__alignCoeff,
        self.__secondDiff,
    ]

def printCompletedTable(self):
    methods_links = self.__getMethodsLinks()
```

```python
for i in range(len(self.x)):
    print("|", end="")

    for j in range(len(methods_links) - 2):
        res = methods_links[j](self.y, self.x[1] - self.x[0], i)
        print(
            f"{res:10.5f}" if res != "%10s" % "null" else res,
            "|",
            sep="",
            end="",
        )

    res = self.__alignCoeff(self.y, self.x, self.x[1] - self.x[0], i)
    print(
        f"{res:10.5f}" if res != "%10s" % "null" else res, "|", sep="", end=""
    )

    res = self.__secondDiff(self.y, self.x[1] - self.x[0], i)
    print(f"{res:10.5f}" if res != "%10s" % "null" else res, "|", sep="")
```

Контрольные вопросы

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

$$y_{N-1} = y_N - h y'_N + \frac{h^2}{2!} y''_N - \frac{h^3}{3!} y'''_N \ldots$$

$$y_{N-2} = y_N - 2h y'_N + \frac{4h^2}{2!} y''_N - \frac{8h^3}{3!} y'''_N \ldots$$

$$y'_N = \frac{3y_N - 4y_{N-1} + y_{N-2}}{2h} + \frac{h^2}{3} y'''_N$$

$$y'_N = \frac{3y_N - 4y_{N-1} + y_{N-2}}{2h} + O(h^2)$$

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

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

$$y_2 = y_0 + 2h y_0' + \frac{4h^2}{2!} y_0'' - \frac{8h^3}{3!} y_0''' \ldots$$

$$4y_1 - y_2 = 4y_0 - y_0 + 2h y_0' + O(h^2)$$

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

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

3. Используя 2-ую формулу Рунге, дать вывод выражения (9) из Лекции No7 для первой производной y'_0 в левом крайнем узле

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

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

$$\left.\begin{aligned} \varphi(h) &= \frac{y_1 - y_0}{h} \\ \varphi(2h) &= \frac{y_2 - y_0}{2h} \end{aligned}\right\} \Omega = \frac{y_1 - y_0}{h} + \frac{y_1 - y_0}{h} - \frac{y_2 - y_0}{2h} = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2)$$

4. Любым способом из Лекций No7, 8 получить формулу порядка точности O(h^3) для первой разностной производной y'_0 в крайнем левом узле x_0 .

$$y_1 = y_0 + \frac{h}{1!}y_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \frac{h^4}{4!}y_0^{IV} \ldots$$

$$y_2 = y_0 + \frac{(2h)}{1!}y_0' + \frac{(2h)^2}{2!}y_0'' + \frac{(2h)^3}{3!}y_0''' + \frac{(2h)^4}{4!}y_0^{IV} \ldots$$

$$y_3 = y_0 + \frac{(3h)}{1!}y_0' + \frac{(3h)^2}{2!}y_0'' + \frac{(3h)^3}{3!}y_0''' + \frac{(3h)^4}{4!}y_0^{IV} \ldots$$

$72(1) - 9(2) + 8(3)$:

$$72y_1 - 9y_2 + 8y_3 = 71y_0 + 78hy_0' + O(h^3)$$

$$y_0' = \frac{-71y_0 + 72y_1 - 9y_2 + 8y_3}{78h} + O(h^3)$$