



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

«Построение и программная реализация алгоритмов
численного дифференцирования»

Студент Маслова Марина Дмитриевна
фамилия, имя, отчество

Группа ИУ7-43Б

Оценка (баллы) _____

Преподаватель Градov Владимир Михайлович
фамилия, имя, отчество

Оглавление

Задание	3
Описание алгоритма	4
Код программы	6
Результат работы	10
Контрольные вопросы	11

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.

Описание алгоритма

Для вычисления значений в *первом столбце* используется *левосторонняя формула*, полученная путем разложения функции в предыдущем узле в ряд Тейлора с центром разложения в текущем узле и выражения из него первой производной (так как используется предыдущий угол значение при $x = 1$ будет отсутствовать):

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

Формула имеет *первый* порядок точности.

Для вычисления значений во *втором столбце* используется *центральная формула*, полученная из разложений функции в ряд Тейлора в предыдущем и следующем узле (из-за чего будут отсутствовать значения в крайних узлах):

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

Формула имеет *второй* порядок точности.

Для вычисления значений в *третьем столбце* используется *2-ая формула Рунге на основе левой разностной производной*:

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

где $\Phi(h)$, $\Phi(mh)$ — формулы левой разностной производной с соответствующим шагом сетки (в общем случае — некоторые приближенные формулы), (в программе используется $m = 2$),

p — точность используемой формулы.

У данной формулы точность *на порядок выше*, чем у левой разностной производной.

Для вычисления значений в *четвертом столбце* используется прием введения *выравнивающих переменных*. Для формулы из задания вычисления производятся следующим образом:

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

$$\frac{1}{y} = \frac{a_1}{a_0} \frac{1}{x} + \frac{a_2}{a_0},$$

$$\xi(x) = \frac{1}{x}, \quad \eta(y) = \frac{1}{y},$$

$$y'_x(x_i) = \frac{y_i^2}{x_i^2} \cdot \frac{\eta_{i+1} - \eta_i}{\xi_{i+1} - \xi_i} = \frac{y_i^2}{x_i^2} \cdot \frac{\frac{1}{y_{i+1}} - \frac{1}{y_i}}{\frac{1}{x_{i+1}} - \frac{1}{x_i}},$$

где второй множитель является правой разностной производной.

Так как при введении выравнивающих переменных исходная кривая преобразуется в прямую, производная которой вычисляется простейшими формулами, то найденное значение будет наиболее близким к истинному значению по сравнению с другими формулами.

Для вычисления значений в **пятом столбце** используется **формула второй разностной производной**, которая аналогично первой разностной производной получается из разложений функции в ряд Тейлора в соответствующих узлах:

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

Формула имеет **второй** порядок точности.

Код программы

Код программы представлен на листингах 1-3.

Листинг 1. differentiation.py

```
"""
    Модуль численного дифференцирования
"""

def getLeftHandDer(cur, prev):
    """
        Первая левосторонняя разностная производная
    """
    return (cur[1] - prev[1]) / (cur[0] - prev[0])

def getCentralDer(prevP, nextP):
    """
        Первая центральная разностная
        производная в каждой точке
    """
    return (nextP[1] - prevP[1]) / (nextP[0] - prevP[0])

def getRungeDer(prev2P, prevP, curP):
    """
        2-ая формула Рунге с использованием
        односторонней производной
    """
    step1Der = getLeftHandDer(curP, prevP)
    step2Der = getLeftHandDer(curP, prev2P)

    return 2 * step1Der - step2Der

def getAlignVarDer(cur, nextP):
    """
        Первая производная при
        введенных выравнивающих переменных
    """
    alignVarCoef = (1 / nextP[1] - 1 / cur[1]) / (1 / nextP[0] - 1 /
cur[0])
    return cur[1] * cur[1] / cur[0] / cur[0] * alignVarCoef

def getSecondDer(prevP, curP, nextP):
    """
        Вторая разностная производная
    """
    return ((prevP[1] - 2 * curP[1] + nextP[1])
            / (nextP[0] - curP[0])
            / (curP[0] - prevP[0]))
```

Листинг 2. table_func.py

```
"""
    Модуль ввода-вывода табличных функций
"""

def read_table(file_name):
    """
        Чтение табличной функции из файла
    """

    func_table = []

    with open(file_name, "r") as file:
        for i, rec in enumerate(file):
            func_table.append(list(map(float, rec.split())))

            if len(func_table[i]) != 2:
                raise TypeError

    if not func_table:
        raise EOFError

    return func_table

def print_table(table):
    """
        Вывод табличной функции
    """

    if table:
        print("Загруженная таблица:")
        print("      x      y")
    else:
        print("Пустой файл!")

    for rec in table:
        print("      {:.2f}   {:.3f}".format(rec[0], rec[1]))
```

Листинг 3. main.py

```
"""
    Модуль для запуска программы
    ЛАБОРАТОРНАЯ РАБОТА №6
    ПОСТРОЕНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ
    АЛГОРИТМОВ ЧИСЛЕННОГО ДИФФЕРЕНЦИРОВАНИЯ
"""

import argparse

import table_func
from differentiation import getLeftHandDer, getCentralDer
from differentiation import getRungeDer, getAlignVarDer
from differentiation import getSecondDer

def create_args():
    """
        Добавление аргументов командной строки
    """
```

```

parser = argparse.ArgumentParser()
parser.add_argument('file_name', nargs='?', default='data/data.csv')
args = parser.parse_args()

return args

def getLeftHandString(i, func):
    """
        Получение строки для вывода
        левосторонней разностной производной
    """

    return ("{:7.3f}".format(getLeftHandDer(func[i], func[i - 1]))
           if i else "{:>7s}".format("--"))

def getCentralString(i, func):
    """
        Получение строки для вывода
        центральной разностной производной
    """

    return ("{:7.3f}".format(getCentralDer(func[i - 1], func[i + 1]))
           if i and i != len(func) - 1 else "{:>7s}".format("--"))

def getRungeString(i, func):
    """
        Получение строки для вывода
        разностной производной по формуле
        Рунге
    """

    return ("{:7.3f}".format(getRungeDer(func[i - 2], func[i - 1],
func[i]))
           if i > 1 else "{:>7s}".format("--"))

def getAlignVarString(i, func):
    """
        Получение строки для вывода
        строки при введенных выравнивающих
        переменных
    """

    return ("{:7.3f}".format(getAlignVarDer(func[i], func[i - 1]))
           if i != len(func) - 1 else "{:>7s}".format("--"))

def getSecondDerString(i, func):
    """
        Получение строки для вывода
        второй разностной производной
    """

    return ("{:7.3f}".format(getSecondDer(func[i - 1], func[i], func[i +
1]))
           if i and i != len(func) - 1 else "{:>7s}".format("--"))

```



```

if __name__ == "__main__":
    ARGS = create_args()

    try:
        func_table = table_func.read_table(ARGS.file_name)

        func_table.sort(key=lambda table: table[0])
        table_func.print_table(func_table)

    except FileNotFoundError:
        print("\nТакого файла не существует!")

    except ValueError:
        print("\nНечисловые данные недопустимы!")
        print("\nПроверьте содержимое файла!")

    except EOFError:
        print("\nПустой файл!")

    except TypeError:
        print("\nВ файле должно быть два столбца с данными!")

    else:
        print("\nТаблица разностных производных")
        print("    1        2        3        4        5    ")
        for i in range(len(func_table)):
            print(getLeftHandString(i, func_table), end=" ")
            print(getCentralString(i, func_table), end=" ")
            print(getRungeString(i, func_table), end=" ")
            print(getAlignVarString(i, func_table), end=" ")
            print(getSecondDerString(i, func_table), end="\n")

        print()
        print("1 -- левая разностная производная;")
        print("2 -- центральная разностная производная;")
        print("3 -- 2-ая формула Рунге с использованием\n"
              + "    односторонней (левой) производной;")
        print("4 -- введены выравнивающие переменные;")
        print("5 -- вторая разностная производная.")

```

Результат работы

Вывод программы:

```
Таблица разностных производных
  1      2      3      4      5
  --      --      --      --      --
0.318    0.260      --    0.247 -0.116
0.202    0.171    0.144    0.165 -0.062
0.140    0.121    0.109    0.118 -0.038
0.102    0.090    0.083    0.089 -0.023
0.079      --    0.068      --      --

1 -- левая разностная производная;
2 -- центральная разностная производная;
3 -- 2-ая формула Рунге с использованием
    односторонней (левой) производной;
4 -- введены выравнивающие переменные;
5 -- вторая разностная производная.
```

Таким образом, итоговая таблица:

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	—	—

Комментарии по поводу использованных формул и их точности приведены в разделе “Описание алгоритма”.

Контрольные вопросы

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

Выполним разложение функции в ряд Тейлора в точках x_{N-1} и x_{N-2} , приняв за центр разложения точку x_N :

$$y_{N-1} = y_N - hy'_N + \frac{h^2}{2!}y''_N - \frac{h^3}{3!}y'''_N + \dots \quad (1)$$

$$y_{N-2} = y_N - 2hy'_N + \frac{4h^2}{2!}y''_N - \frac{8h^3}{3!}y'''_N + \dots \quad (2)$$

4 · (1) – (2):

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

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

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

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

$$y_1 = y_0 + hy'_0 + \frac{h^2}{2!}y''_0 + \frac{h^3}{3!}y'''_0 + \frac{h^4}{4!}y^{IV}_0 + \dots \quad (1)$$

$$y_2 = y_0 + 2hy'_0 + \frac{4h^2}{2!}y''_0 + \frac{8h^3}{3!}y'''_0 + \frac{16h^4}{4!}y^{IV}_0 + \dots \quad (2)$$

$$y_3 = y_0 + 3hy'_0 + \frac{9h^2}{2!}y''_0 + \frac{27h^3}{3!}y'''_0 + \frac{81h^4}{4!}y^{IV}_0 + \dots \quad (3)$$

(2) – 2 · (1):

$$y_2 - 2y_1 = -y_0 + \frac{2h^2}{2!}y''_0 + \frac{6h^3}{3!}y'''_0 + \frac{14h^4}{4!}y^{IV}_0 + \dots \quad (4)$$

(3) – 3 · (1):

$$y_3 - 3y_1 = -2y_0 + \frac{6h^2}{2!}y''_0 + \frac{24h^3}{3!}y'''_0 + \frac{78h^4}{4!}y^{IV}_0 + \dots \quad (5)$$

4 · (1) – (5):

$$4y_2 - 8y_1 - y_3 + 3y_1 = -2y_0 + \frac{2h^2}{2!}y''_0 - \frac{22h^4}{4!}y^{IV}_0 + \dots$$

$$-5y_1 + 4y_2 - y_3 = -2y_0 + h^2y''_0 - \frac{11h^4}{12}y^{IV}_0 + \dots$$

$$y''_0 = \frac{2y_0 - 5y_1 + 4y_2 - y_3}{h^2} + \frac{11}{12}h^2y^{IV}_0$$

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

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

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

2-ая формула Рунге:

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

В нашем случае:

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

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

$$m = 2; p = 1$$

Тогда:

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

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

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

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 .

$$y_1 = y_0 + hy'_0 + \frac{h^2}{2!}y''_0 + \frac{h^3}{3!}y'''_0 + \frac{h^4}{4!}y^{IV}_0 + \dots \quad (1)$$

$$y_2 = y_0 + 2hy'_0 + \frac{4h^2}{2!}y''_0 + \frac{8h^3}{3!}y'''_0 + \frac{16h^4}{4!}y^{IV}_0 + \dots \quad (2)$$

$$y_3 = y_0 + 3hy'_0 + \frac{9h^2}{2!}y''_0 + \frac{27h^3}{3!}y'''_0 + \frac{81h^4}{4!}y^{IV}_0 + \dots \quad (3)$$

$$(2) - 4 \cdot (1):$$

$$y_2 - 4y_1 = -3y_0 - 2hy'_0 + \frac{4h^3}{3!}y'''_0 + \frac{12h^4}{4!}y^{IV}_0 + \dots \quad (4)$$

$$(3) - 9 \cdot (1):$$

$$y_3 - 9y_1 = -8y_0 - 6hy'_0 + \frac{18h^3}{3!}y'''_0 + \frac{72h^4}{4!}y^{IV}_0 + \dots \quad (5)$$

$$2 \cdot (5) - 9 \cdot (4):$$

$$2y_3 - 18y_1 - 9y_2 + 36y_0 = 11y_0 + 6hy'_0 + \frac{36h^4}{4!}y^{IV}_0 + \dots$$

$$18y_1 - 9y_2 + 2y_3 = 11y_0 + 6hy'_0 + \frac{3h^4}{2}y^{IV}_0 + \dots$$

$$y'_0 = \frac{-11y_0 + 18y_1 - 9y_2 + 2y_3}{6h} - \frac{1}{4}h^3 y^{IV}_0$$

$$y'_0 = \frac{-11y_0 + 18y_1 - 9y_2 + 2y_3}{6h} + O(h^3)$$