МГТУ им. Н. Э. Баумана

Отчет по лабораторной работе №6 по предмету «Вычислительные алгоритмы»
«Построение и программная реализация алгоритмов численного дифференцирования»

Выполнила Параскун София,
ИУ7-44Б
Проверил
Градов В. М.

Москва, 2021 г.

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

1. Задание

Задана табличная (сеточная) функция. Имеется информация, что закономерность, представленная этой таблицей, может быть описана формулой $y = a_0x / (a_1 + a_2x)$; параметры функций неизвестны и определять их ненужно.

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,
5 — 2-я разностная производная.

2. Код программы

Программа состоит из файла main.py.

```
from math import pow

class Differentiate:
  def __init__(self):
    self.h = 1.0
    self.x = [1, 2, 3, 4, 5, 6]
    self.y = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]
    self.y1 = [0 for i in range(6)]
    self.y2 = [0 for i in range(6)]
    self.y3 = [0 for i in range(6)]
    self.y4 = [0 for i in range(6)]
    self.y5 = [0 for i in range(6)]

  def output(self):
    print("  x  |   y   |     1    |     2    |     3    |     4    |     5")
```

```
    print("-" * 70)
    for i in range(6):
      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.y1[i], self.y2[i], self.y3[i], self.y4[i], se
lf.y5[i]))

  def calc(self):
    self.leftDif()
    self.centerDif()
    self.secondRungeDif(1)
    self.alignedCoeffsDif()
    self.secondLeftDif()

  def leftDif(self):
    for i in range(len(self.y)):
      if i == 0:
        self.y1[i] = 0.0
      else:
        self.y1[i] = (self.y[i] - self.y[i - 1]) / self.h

  def centerDif(self):
    for i in range(len(self.y)):
      if i == 0 or i == len(self.y) - 1:
        self.y2[i] = 0.0
      else:
        self.y2[i] = (self.y[i + 1] - self.y[i - 1]) / 2 * self.h

  def secondRungeDif(self, p):
    yTmp = []
    for i in range(len(self.y)):
      if i < 2:
        yTmp.append(0)
      else:
        yTmp.append((self.y[i] - self.y[i - 2]) / (2 / self.h))

    for i in range(len(self.y1)):
      if i < 2:
        self.y3[i] = 0.0
      else:
        self.y3[i] = self.y1[i] + (self.y1[i] - yTmp[i]) / (pow(2, p) - 1)

  def alignedCoeffsDif(self):
    for i in range(len(self.y)):
      if i == len(self.y) - 1:
        self.y4[i] = 0.0
      else:
        k = pow(self.y[i], 2) / pow(self.x[i], 2)
        self.y4[i] = k * (((-1.0 / self.y[i + 1]) - (-1.0 / self.y[i])) / ((-1.0 / self.x[i
+ 1]) - (-1.0 / self.x[i])))

  def secondLeftDif(self):
    for i in range(len(self.y)):
      if i == 0 or i == len(self.y) - 1:
        self.y5[i] = 0.0
      else:
        self.y5[i] = (self.y[i - 1] - 2 * self.y[i] + self.y[i + 1]) / pow(self.h, 2)
```

```
if __name__ == "__main__":
  print("Lets start!")
  calc = Differentiate()
  calc.__init__()
  calc.calc()
  calc.output()
```

3. Результаты работы

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

1 – левосторонняя формула точностью O(h);
2 – центральная формула точностью O(h^2);
3 – вторая формула Рунге (с использованием левосторонней формулы);
4 — применение выравнивающих переменных (из-за того, что параметры неизвестны, оценка точности затруднительна). Было использовано соотношение

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

5 — вторая разностная производная.

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.

$$y_{N-1} = y_N - hy'_n + \frac{h^2}{2!}y''_N - \frac{h^3}{3!}y'''_N \dots$$
$$y_{N-2} = y_N - 2hy'_n + \frac{4h^2}{2!}y''_N - \frac{8h^3}{3!}y'''_N \dots$$

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

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

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

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

Проведем преобразования — домножим y_1 на 4 и вычтем y_2

$$4y_1 - y_2 = 4y_0 - y_0 + 2hy'_0 + O(h^2)$$
$$y'_0 = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2)$$
$$y''_0 = \frac{-y_3 + 4y_2 - 5y_1 + 2y_0}{h^2} + O(h^2)$$

3. Используя 2-ю формулу Рунге, дать вывод выражения (9) из лекции №7 для первой производной y'0 в левом крайнем узле

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

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

Для левого узла n = 0, n + 1 = 1, n + 2 = 2, откуда $y_0' = \frac{-3y_0+4y_1-y_2}{2h} + O(h^2)$

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.

$$y_1 = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' \ldots$$
$$y_2 = y_0 + 2hy_0' + \frac{4h^2}{2!}y_0'' + \frac{8h^3}{3!}y_0''' \ldots$$
$$y_3 = y_0 + 3hy_0' + \frac{9h^2}{2!}y_0'' + \frac{27h^3}{3!}y_0''' \ldots$$

$$y' = \frac{y_3 + 27y_1 - 28y_0}{30h} + O(h^3)$$