

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 6**

|  |  |
| --- | --- |
| **Тема:** Построение и программная реализация алгоритмов численного дифференцирования.  **Студент:** Елгин И. Ю.  **Группа:** ИУ7-44Б  **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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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** |  |  |  |  |  |

1. **Код на Python.**
2. *# односторонняя разностная производная*
3. **def** LeftDiff(y, h):
4. result = []
5. **for** i **in** range(len(y)):
6. **if not i:**
7. result.append("-")
8. **else**:
9. result.append((y[i] - y[i - 1]) / h)
10. **return** result
12. *#центральная разностная производная*
13. **def** CenterDiff(y, h):
14. result = []
15. **for** i **in** range(len(y)):
16. **if not i or i == (len(y) - 1):**
17. result.append("-")
18. **else**:
19. result.append((y[i + 1] - y[i - 1]) / (2 \* h))
20. **return** result
22. *# 2-я формула Рунге с использованием односторонней производной*
23. **def** RungeLeft(y, h):
24. result = []
25. **for** i **in** range(len(y)):
26. **if i <= 1:**
27. result.append("-")
28. **else**:
29. result.append(2 \* ((y[i] - y[i - 1]) / h) -
30. ((y[i] - y[i - 2]) / (2 \* h)))
32. **return** result
34. *# c выравнивающими переменными*
35. **def** Alignment(x, y, h):
36. **result = []**
37. **for** i **in** range(len(y)):
38. **if** i == (len(y) - 1):
39. result.append("-")
40. **else**:
41. **result.append((1 / y[i + 1] - 1 / y[i]) / (1 /**
42. x[i + 1] - 1 / x[i]) \* y[i]\*\*2 / x[i]\*\*2)
44. **return** result
46. ***#вторая разностная производная***
47. **def** SecondDiff(y, h):
48. result = []
49. **for** i **in** range(len(y)):
50. **if** **not** i **or** i == (len(y) - 1):
51. **result.append("-")**
52. **else**:
53. result.append((y[i - 1] - 2 \* y[i] +
54. y[i + 1]) / h \*\* 2)
56. **return result**
58. *#вывод результата*
59. **def** Result(lst):
60. **for** i **in** range(len(lst)):
61. **if lst[i] != "-":**
62. **print**("{:5.3f}".format(lst[i]), end=" ")
63. **else**:
64. **print**("{:<5}".format("-"), end=" ")
65. **print**()

68. *#таблица данных*
69. h = 1
70. x = [i **for** i **in** range(1, 7)]
71. **y = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]**
73. *#результаты диффиренцирования*
74. res\_left\_diff = LeftDiff(y, h)
75. res\_center = CenterDiff(y, h)
76. **res\_runge = RungeLeft(y, h)**
77. res\_alignment = Alignment(x, y, h)
78. res\_second\_diff = SecondDiff(y, h)
80. *#вывод результата*
81. **for i in [x, y, res\_left\_diff, res\_center, res\_runge, res\_alignment, res\_second\_diff]:**
82. Result(i)

**Результаты работы.**

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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **x** | **y** | **1** | **2** | **3** | **4** | **5** |
| **1** | **0.571** | **-** | **-** | **-** | **0.408** | **-** |
| **2** | **0.889** | **0.318** | **0.26** | **-** | **0.247** | **-0.116** |
| **3** | **1.091** | **0.202** | **0.171** | **0.144** | **0.165** | **-0.062** |
| **4** | **1.231** | **0.14** | **0.121** | **0.109** | **0.118** | **-0.038** |
| **5** | **1.333** | **0.102** | **0.09** | **0.083** | **0.089** | **-0.023** |
| **6** | **1.412** | **0.79** | **-** | **0.068** | **-** | **-** |

1. Левосторонняя разностная производная:

Выполним разложение функции в ряд Тейлора

И получаем разностную формулу для вычисления первых производных



Порядок точности функции O(h).

1. Центральная разностная производная:

Выполним вычитание разложенных функций в ряд Тейлора, результат:

Порядок точности формулы O(h^2).

1. Формула Рунге на основе левосторонней разностной производной: Имеем некоторую приближенную формулу:



Для шага mh:



Из этого получаем формулу:



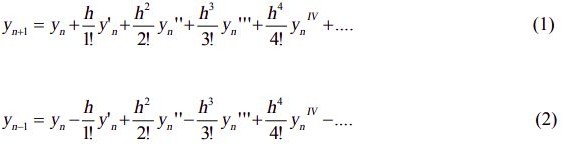
, точность которой выше.

1. Метод выравнивающих переменных:

При удачном выборе выравнивающих переменных исходная

кривая может быть преобразована в прямую линию, производная от которой вычисляется точно по самым простым формулам.

1. Вторая разностная производная:



Сложив (1) и (2) получим разностный аналог второй производной:

**4 Вопросы при защите лабораторной работы.**

1. Получить формулу порядка точности для первой разностной производной в крайнем правом узле .

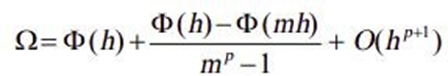
2. Получить формулу порядка точности для второй разностной производной в крайнем левом узле .

*Получим:*

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



4. Любым способом из Лекций №7, 8 получить формулу порядка точности для первой разностной производной в крайнем левом узле .

****

