**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** Малышев И.А.

**Группа** ИУ7-41Б

**Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Преподаватель** Градов В.М.

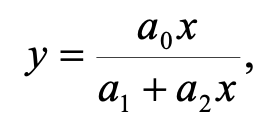
Москва.

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:



параметры функции неизвестны.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |  |  |  |  |  |

**Задание:**

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 - вторая разностная производная.

1. **Код программы**

def left(x, y, h, i):

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

def right(x, y, h, i):

return (y[i + 1] - y[i]) / h if i < len(y) - 1 else None

def center(x, y, h, i):

return (y[i + 1] - y[i - 1]) / 2 / h if i < len(y) - 1 and i > 0 else None

def left\_double(x, y, h, i):

return (y[i] - y[i - 2]) / 2 / h if i > 1 else None

def runge\_left(x, y, h, i):

if i < 2:

return None

f1 = left(x, y, h, i)

f2 = left\_double(x, y, h, i)

return f1 + (f1 - f2)

def align\_vars\_right(x, y, h, i):

if i > len(y) - 2:

return None

der = (1 / y[i + 1] - 1 / y[i]) / (1 / x[i + 1] - 1 / x[i])

return der \* y[i] \* y[i] / x[i] / x[i]

def second\_der(x, y, h, i):

return (y[i - 1] - 2 \* y[i] + y[i + 1]) / (h \* h) if i < len(y) - 1 and i > 0 else None

def main():

x = [1, 2, 3, 4, 5, 6]

y = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]

methods = [left, center, runge\_left, align\_vars\_right, second\_der]

print("┌─────────┬─────────┬─────────┬─────────┬─────────┬─────────┬─────────┐")

print("│ x │ y │ left │ center │ runge │ align │ second │")

print("├─────────┼─────────┼─────────┼─────────┼─────────┼─────────┼─────────┤")

for i in range(len(x)):

print("│{:9.3f}│".format(x[i]), end = "")

print("{:9.3f}│".format(y[i]), end = "")

for func in methods:

res = func(x, y, 1, i)

if res is None:

print(" - │", end = "")

else:

print("{:9.4f}│".format(res), end = "")

print()

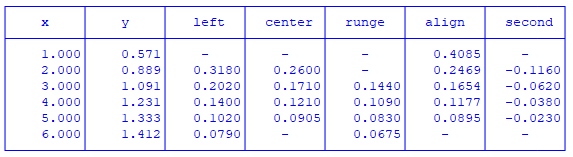
print("└─────────┴─────────┴─────────┴─────────┴─────────┴─────────┴─────────┘")

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. **Результат работы программы**

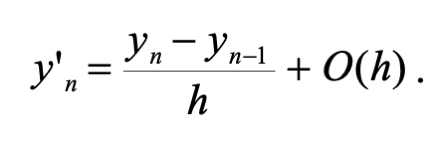
*Результат работы программы:*



Комментарии по поводу использованных формул и их точности:

1. *Левая разностная производная (столбец left)*

Формула:

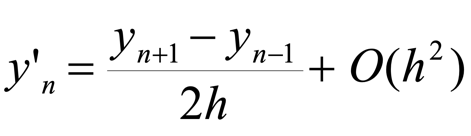


Точность: первый порядок точности относительно шага h.

Комментарий: получается из разложения функции в ряд Тейлора

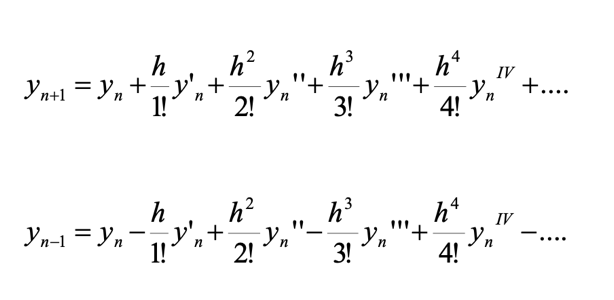
1. *Центральная разностная производная (столбец center)*

Формула:



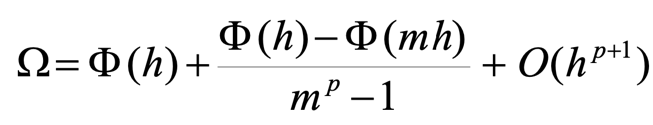
Точность: второй порядок точности относительно шага

Комментарий: получается из разности двух разложениий функции в ряд Тейлора:



1. *2-я формула Рунге с использованием односторонней производной (столбец runge)*

Формула:



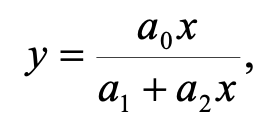
Точность: формула Рунге позволяет за счет расчета на двух сетках с отличающимися шагами получить решение с более высокой точностью, чем заявленная теоретическая точность используемой формулы, соотвественно, если точность Ф = p, то точность формулы будет p + 1.

Комментарий: так как была использована формула Рунге для левой разностной производной, то в формуле m = 2 (удвоенный шаг), а p = 1, поэтому:

res = left(h) + left(h) – left(2\*h)

1. *Введение выравнивающих переменных (столбец align)*

Так как в условии сказано, что сеточная функция может быть описана следующей зависимостью:



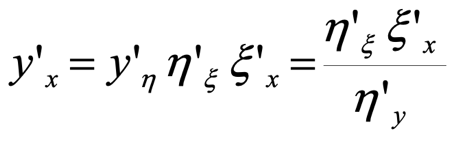
то целесообразно ввести такие выравнивающие переменные:

(y) = 1 / y (x) = 1 / x

Действительно, тогда указанная зависимость принимает вид:

А смысл выравнивающих переменных как раз и состоит в том, чтобы исходная кривая была преобразована в прямую линию, производная от которой вычисляется точно по самым простым формулам.

Для возврата к исходным переменным используется формула:



Окончательно, формула принимает вид (используется правая разностная производная):

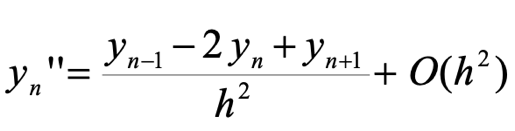
res = right(1 / x[i] - 1 / x[i - 1]) \* y[i]2 / x[i]2, где

right(1 / x[i] - 1 / x[i - 1]) = (1 / y[i + 1] - 1 / y[i]) / (1 / x[i + 1] - 1 / x[i])

Точность: формула абсолютно точная

1. *Вторая разностная производная (столбец second)*

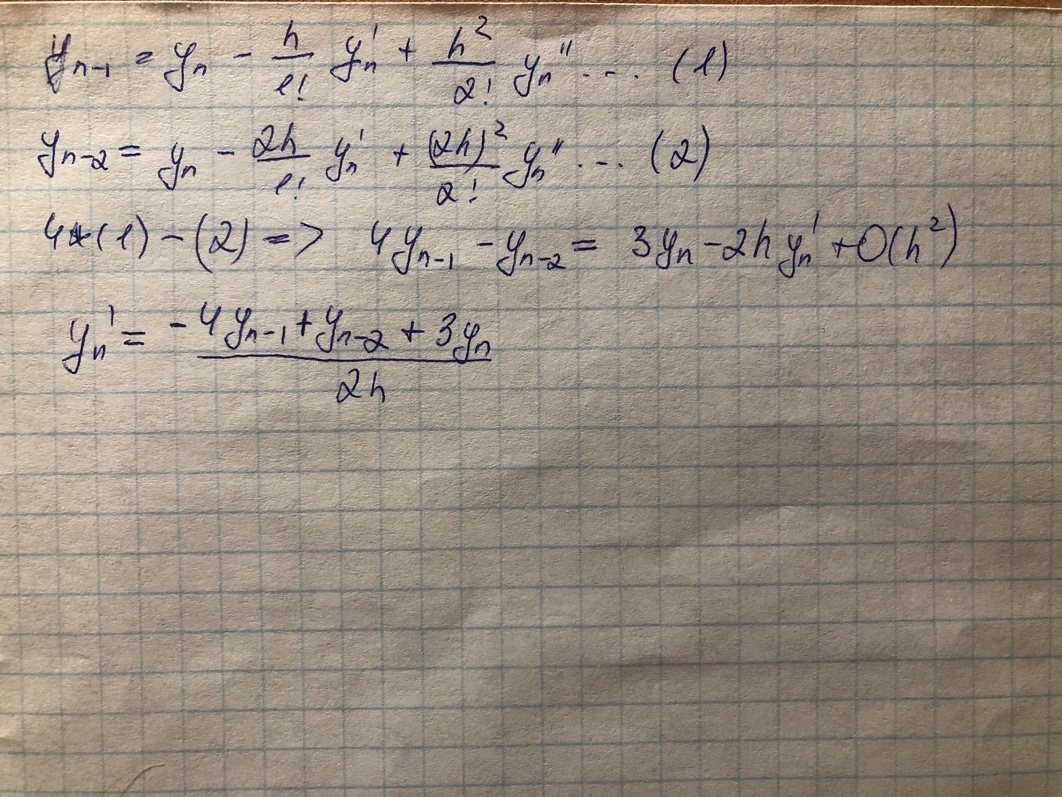
Формула:



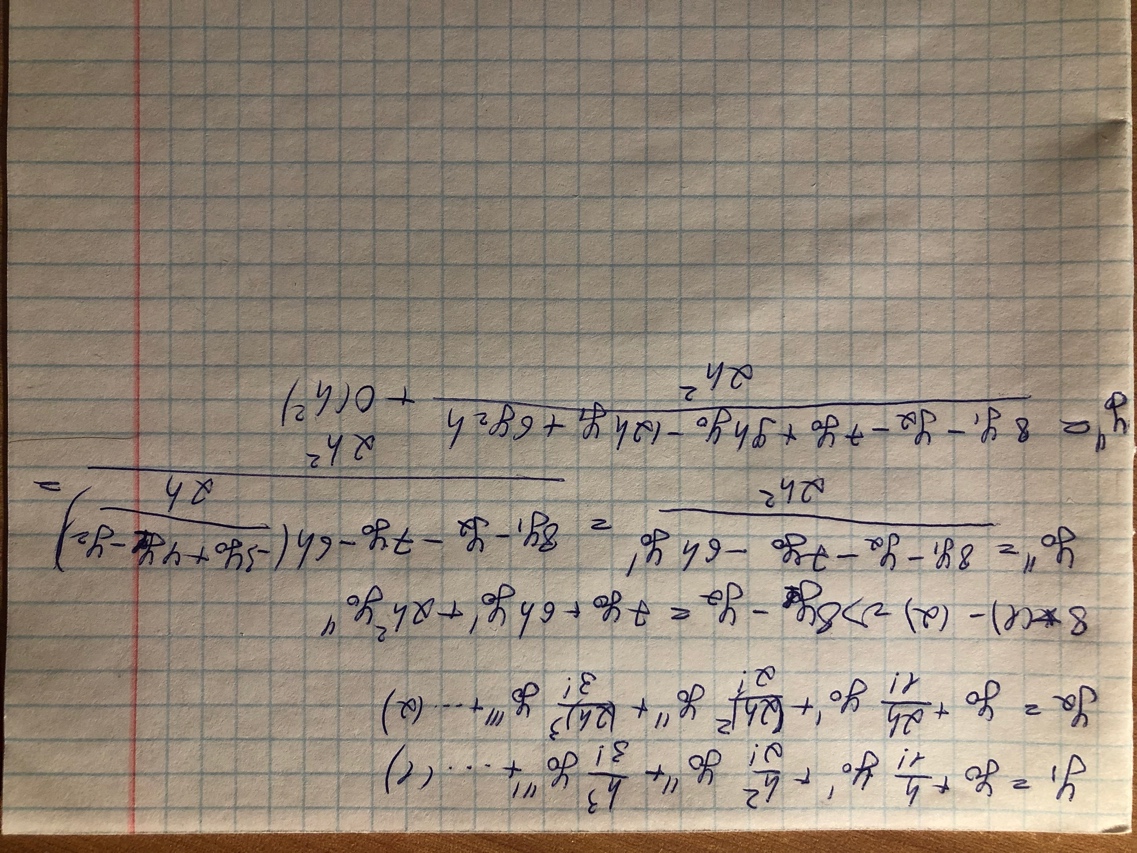
Точность: второй порядок точности относительно шага

Комментарий: получается из разложения функции в ряд Тейлора

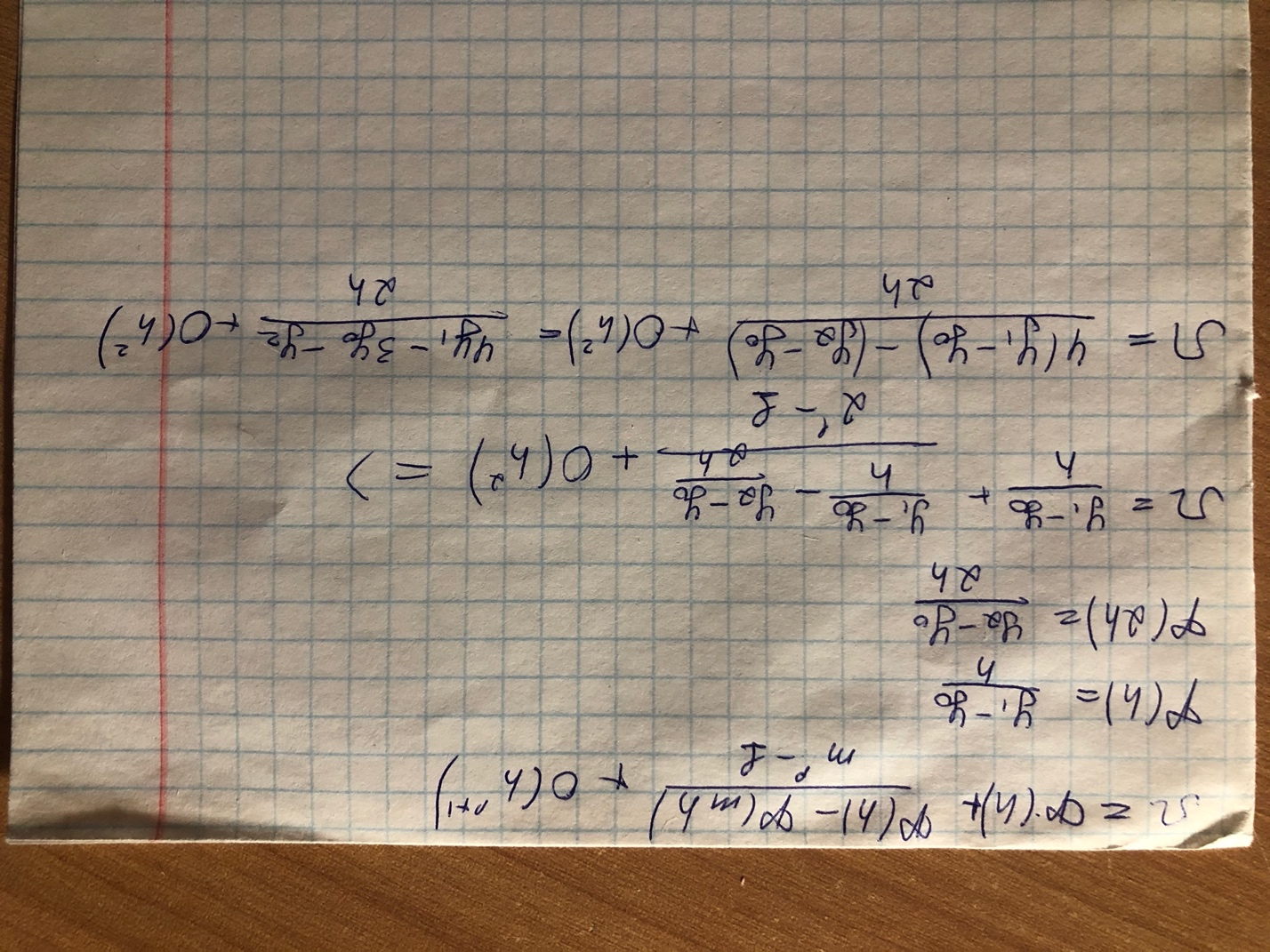
1. **Ответы на вопросы**
2. Получить формулу порядка точности O(h2) для первой разностной производной y'N в крайнем правом узле xN.

****

1. Получить формулу порядка точности O(h2) для второй разностной производной y''0 в крайнем левом узле x0.

****

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

****

1. Любым способом из Лекций №7, 8 получить формулу порядка точности O(h3) для первой разностной производной y'0 в крайнем левом узле x0.

