НИУ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Вычислительная математика»

**Отчет**

По лабораторной работе №5 «Интерполяция функции»

Вариант 27

Выполнил:

*студент группы P32131*

*Овсянников Роман Дмитриевич*

Преподаватель:

*Малышева Татьяна Алексеевна*

Санкт-Петербург,

2023 г.

**Цель работы:**

Решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

**Порядок выполнения работы:**

Вычислительная реализация задачи:

1. Выбрать из табл. 1 заданную по варианту таблицу y = f(x) (таблица 1.1 –

таблица 1.5);

2. Построить таблицу конечных разностей для заданной таблицы. Таблицу отразить в отчете;

3. Вычислить значения функции для аргумента X1(см. табл.1), используя

первую или вторую интерполяционную формулу Ньютона. Обратить внимание какой конкретно формулой необходимо воспользоваться;

4. Вычислить значения функции для аргумента X2 (см. табл. 1), используя

первую или вторую интерполяционную формулу Гаусса. Обратить внимание

какой конкретно формулой необходимо воспользоваться;

5. Подробные вычисления привести в отчете.

Программная реализация задачи:

1. Исходные данные задаются тремя способами:

a) в виде набора данных (таблицы x,y), пользователь вводит значения с клавиатуры;

b) в виде сформированных в файле данных (подготовить не менее трех тестовых

вариантов);

c) на основе выбранной функции, из тех, которые предлагает программа, например, sinx. Пользователь выбирает уравнение, исследуемый интервал и количество точек на интервале (не менее двух функций).

2. Сформировать и вывести таблицу конечных разностей;

3. Вычислить приближенное значение функции для заданного значения аргумента, введенного с клавиатуры, указанными методами (см. табл. 5.2). Сравнить

полученные значения;

4. Построить графики заданной функции с отмеченными узлами интерполяции и

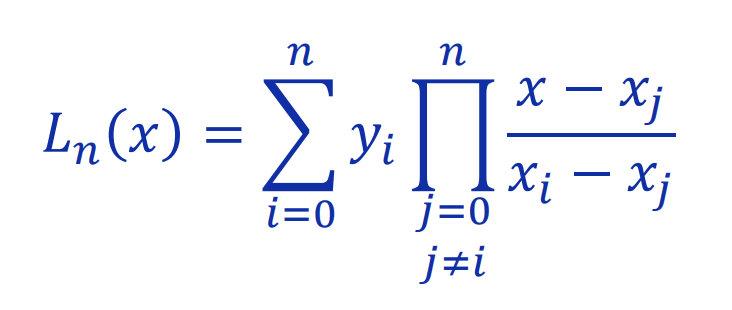
интерполяционного многочлена Ньютона/Гаусса (разными цветами);

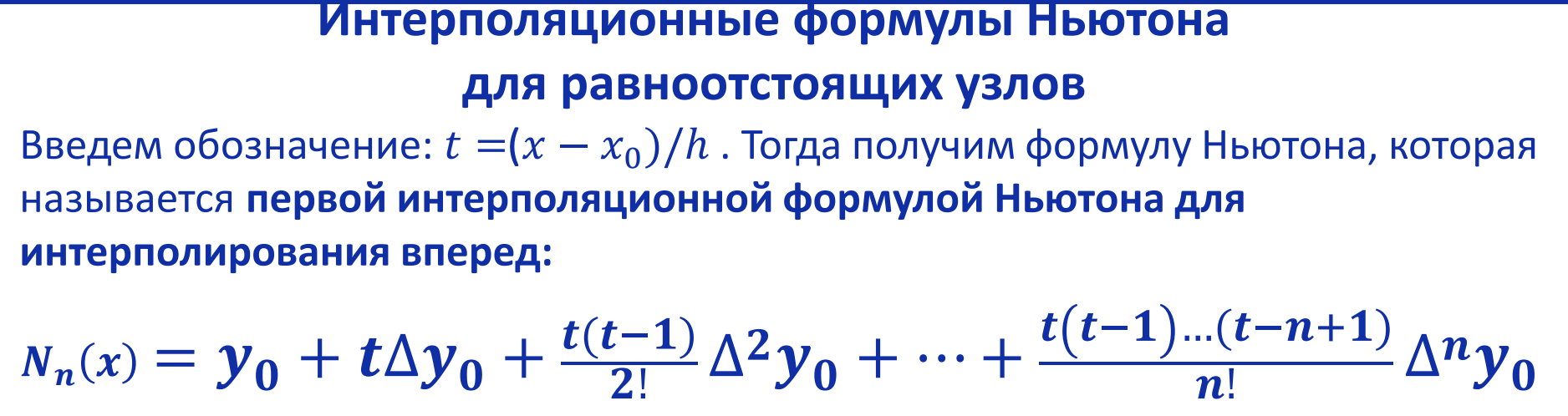
5. Программа должна быть протестирована на различных наборах данных, в том

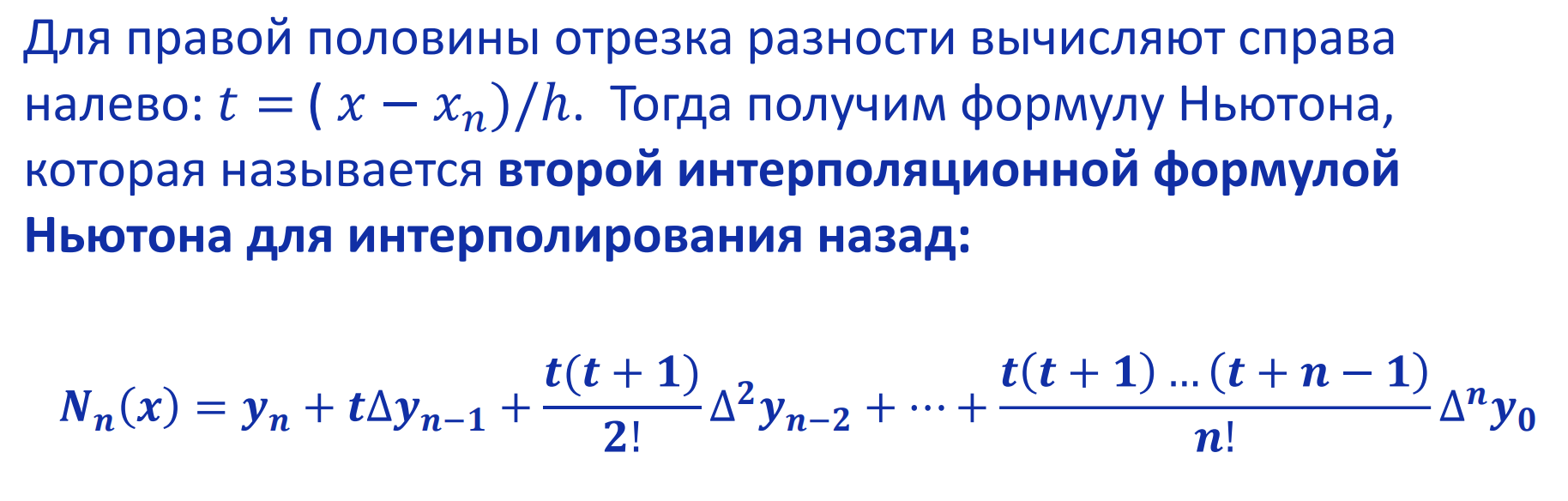
числе и некорректных.

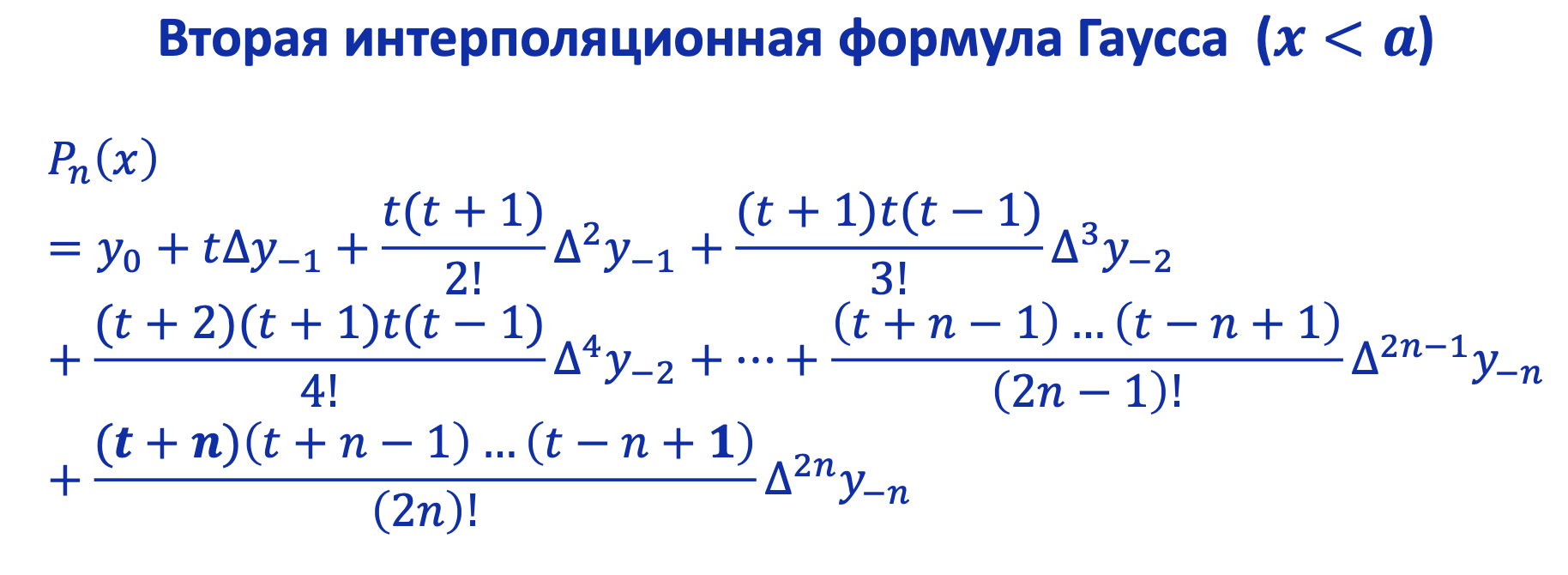
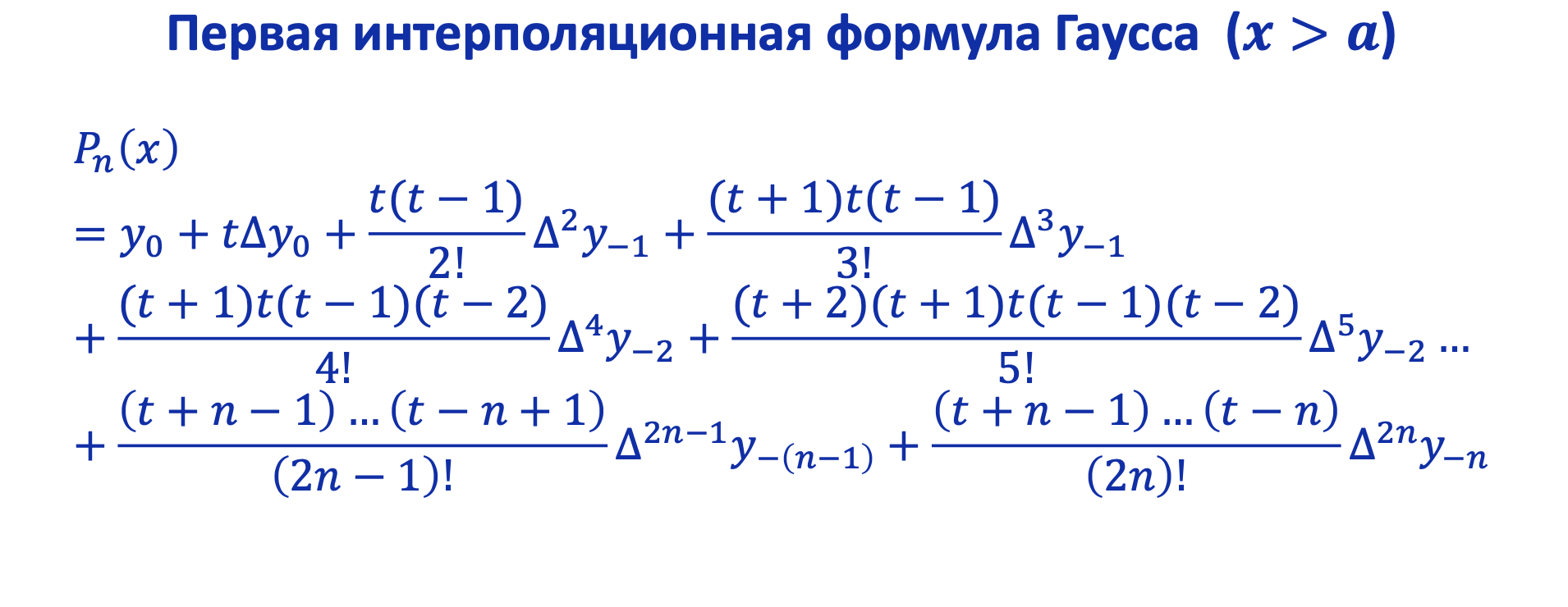
6. Проанализировать результаты работы программы.

**Рабочие формулы:**

****

****

****

****

**Вычислительная часть лабораторной работы:**

| x | y | Δy | Δ^2y | Δ^3y | Δ^4y | Δ^5y | Δ^6y |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0,5 | 1,532 | 1,0036 | 0,0014 | -0,0008 | -0,0012 | 0,0059 | -0,0166 |
| 0,55 | 2,5356 | 1,005 | 0,0006 | -0,002 | 0,0047 | -0,0107 |  |
| 0,6 | 3,5406 | 1,0056 | -0,0014 | 0,0027 | -0,006 |  |  |
| 0,65 | 4,5462 | 1,0042 | 0,0013 | -0,0033 |  |  |  |
| 0,7 | 5,5504 | 1,0055 | -0,002 |  |  |  |  |
| 0,75 | 6,5559 | 1,0035 |  |  |  |  |  |
| 0,8 | 7,5594 |  |  |  |  |  |  |

X1 = 0,783

Так как больше 0.65, то используем 2 формулу Ньютона:  
h = 0.05  
t = (0,783 - 0,8) / 0,05 = -0.34

N6(0,783) = 7.5594 + -0.34 \* 1.0035 + -0.34 \* (-0.34 + 1) \* -0.002 / 2 + -0.34 \* (-0.34 + 1) \* (-0.34 + 2) \* -0.0033 / 6 + -0.34 \* (-0.34 + 1) \* (-0.34 + 2) \* (-0.34 + 3) \* -0.006 / 24 + -0.34 \* (-0.34 + 1) \* (-0.34 + 2) \* (-0.34 + 3) \* (-0.34 + 4) \* -0.0107 / 120 + -0.34 \* (-0.34 + 1) \* (-0.34 + 2) \* (-0.34 + 3) \* (-0.34 + 4) \* (-0.34 + 5) \* -0.0166 / 720 = **7.22**

X2 = 0,683

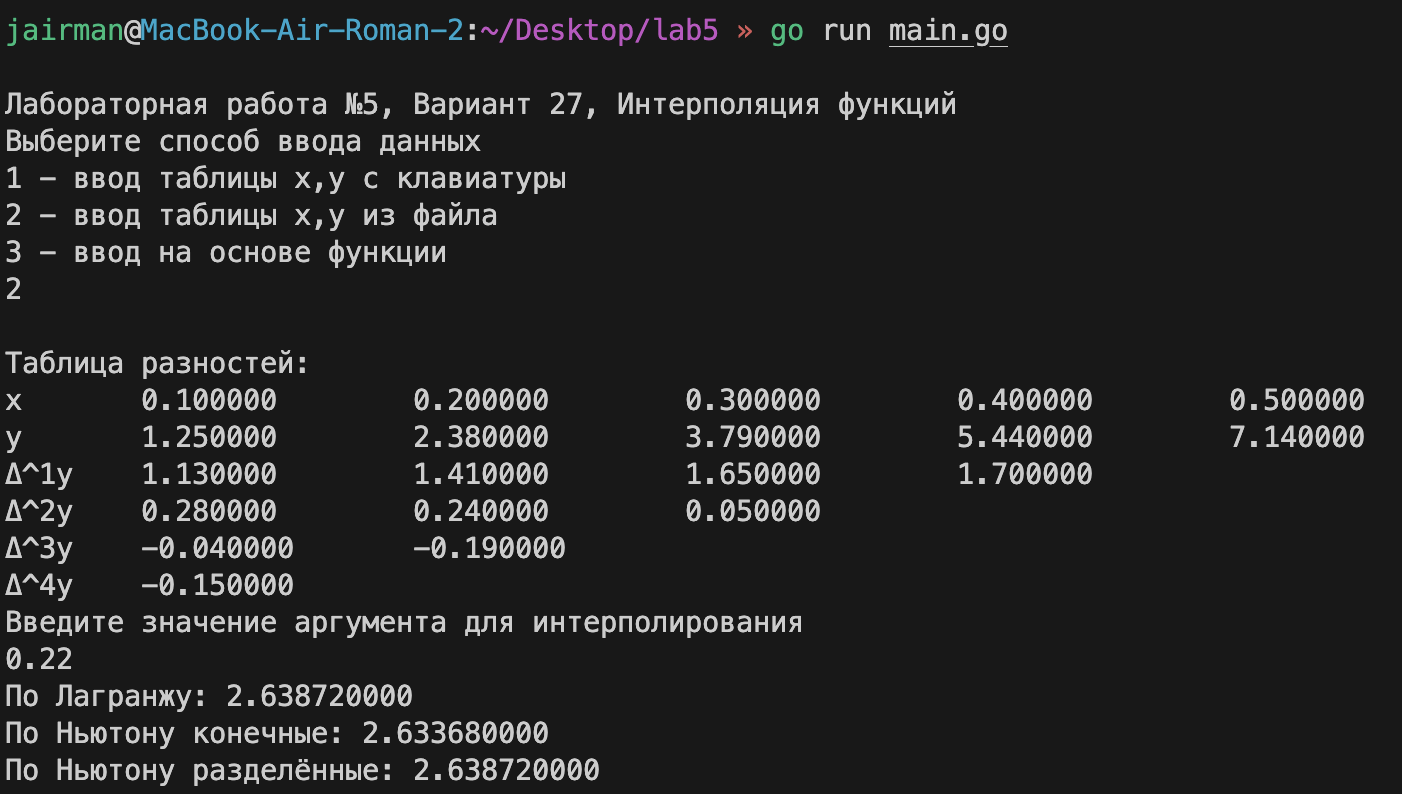
Так как больше 0.65, то используем 1 формулу Гаусса:  
h = 0.05  
t = (0.683 - 0.65) / 0.05 = 0.66

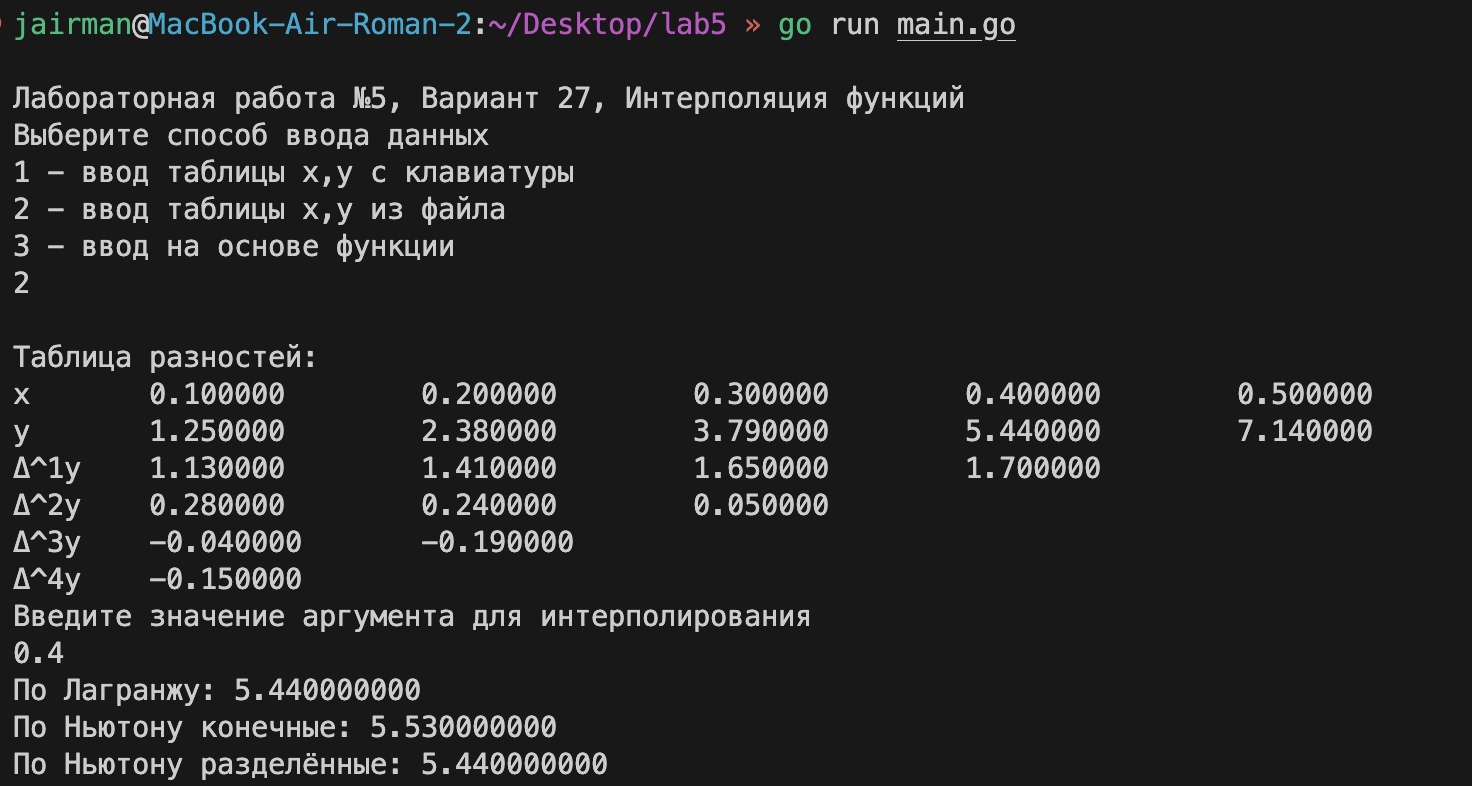
P6(0.683) = 4.5462 + 0.66\*1.0042 + 0.66\*(0.66-1)/2 \* (-0.0014) + (0.66+1)\*0.66\*(0.66-1)/(2\*3) \* (0.0027) + (0.66+1)\*0.66\*(0.66-1)\*(0.66-2)/(2\*3\*4) \* (0.0047) + (0.66+2)\*(0.66+1)\*0.66\*(0.66-1)\*(0.66-2)/(2\*3\*4\*5) \* (-0.0107) + (0.66+2)\*(0.66+1)\*0.66\*(0.66-1)\*(0.66-2)\*(0.66-3)/(2\*3\*4\*5\*6) \* (-0.0166) = **5.21**

**Листинг программы:**

Код программы: <https://github.com/Ja1rman/Computational-Mathematics/blob/main/lab5/main.go>

**Результаты выполнения программы при различных исходных данных**:





**Вывод:**

В результате выполнения лабораторной работы я познакомился с интерполяции функции и реализовал метод с использованием многочлена Лагранжа и метод с использованием многочлена Ньютона с конечными разностями, а также дополнительно реализовал метод с использованием многочлена Ньютона с разделёнными разностями на языке программирования Go.

Использование многочлена Лагранжа целесообразно для нескольких точек на одном и том же отрезке, так как этот метод медленнее и нестабильнее метода с использованием многочлена Ньютона.