**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний університет “Львівська політехніка”**

**Інститут ІКНІ**

**Кафедра ПЗ**



**ЗВІТ**

**До лабораторної роботи №8**

**На тему: “Наближення дискретних (таблично заданих) функцій”**

**З дисципліни : “Чисельні методи”**

**Лектор:**

доц. кафедри ПЗ

Мельник Н.Б.

**Виконав:**

студ. групи ПЗ-18

Кириченко М.К.

**Прийняв:**

проф. каф. ПЗ

Гавриш В.І.

«\_\_\_» \_\_\_\_\_\_ 2023 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2023

**Тема роботи:** Наближення дискретних (таблично заданих) функцій**.**

**Мета роботи:** Ознайомлення на практиці з методами інтерполяції функцій.

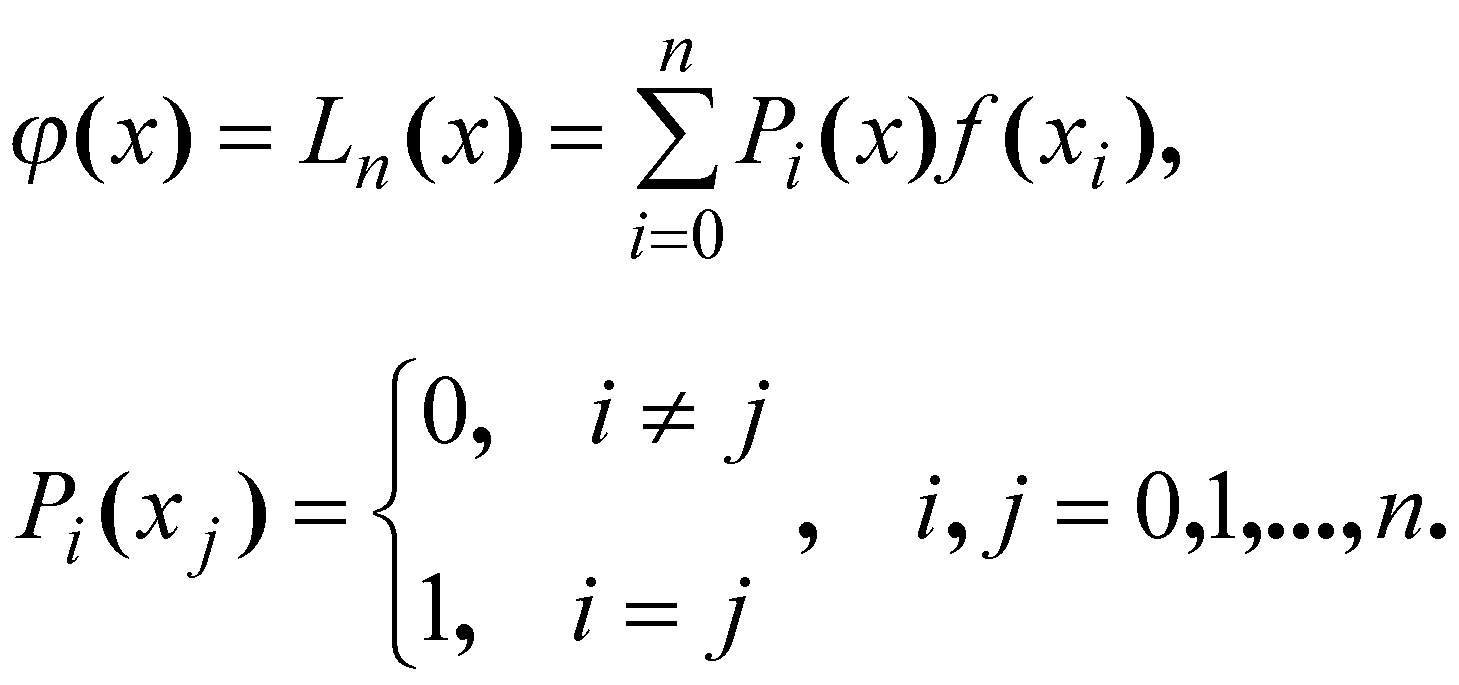
**Теоретичні відомості**

**Інтерполяційний поліном Лагранжа**

Один з підходів до задачі інтерполяції – метод Лагранжа. Основна ідея цього методу полягає в пошуку поліному, який приймає значення 1 в одному довільному вузлі інтерполяції і значення 0 у всіх інших вузлах.

Наближену функцію представимо у вигляді





Оскільки точки є коренями многочлена , то його можна записати наступним чином



,



а наближена функція , яку називають *інтерполяційним многочленом Лагранжа*, матиме вигляд



**Інтерполяційний поліном Ньютона**

Іншим підходом до задачі інтерполяції є метод Ньютона (метод розділених різниць). Нехай для функції задано її значення в точках Треба побудувати такий поліном степеня не вище від значення якого у вузлах інтерполювання збігаються із значенням функції , тобто



Поліном будемо шукати у вигляді



розділена різниця го порядку.



Нехай вузли інтерполяції утворюють арифметичну прогресію , крок інтерполяції.



Таким чином, скінченну різницю го порядку можна записати у вигляді



Розділену різницю го порядку можна виразити через скінченну різницю го порядку



Тоді наведений вище поліном можна записати у вигляді



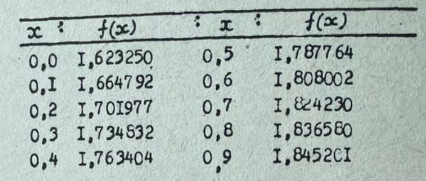
отримане представлення називають *інтерполяційним поліномом Ньютона для інтерполяціїї вперед* для рівновіддалених вузлів інтерполяції*.*

Формули Ньютона та Лагранжа характеризують один і той самий поліном, вони відрізняються лише алгоритмом його побудови.

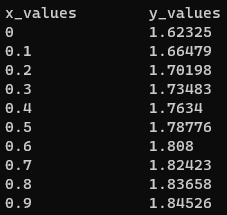
**Індивідуальне завдання**

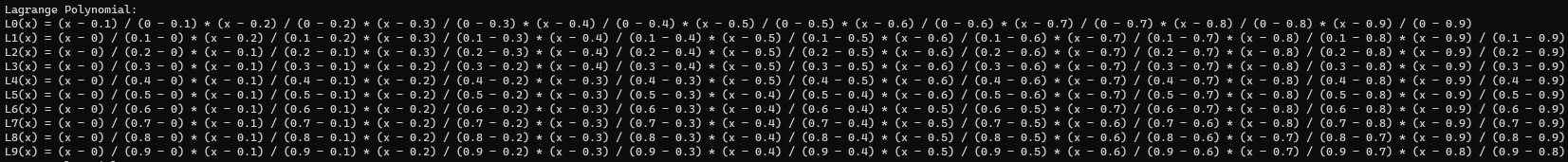
Використовуючи інтерполяційні поліноми Лагранжа та Ньютона, обчислити значення таблично заданої функції у точці x0.

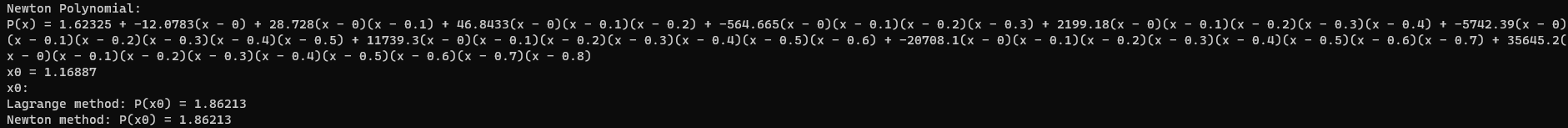
*Таблично задана функція:*



**Результат виконання програми**

****

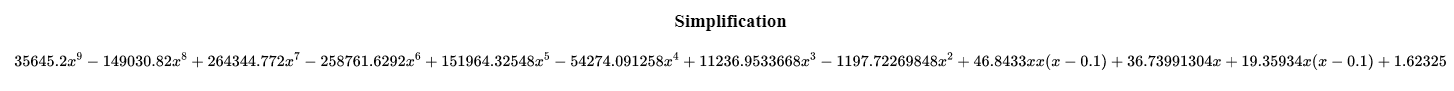


****

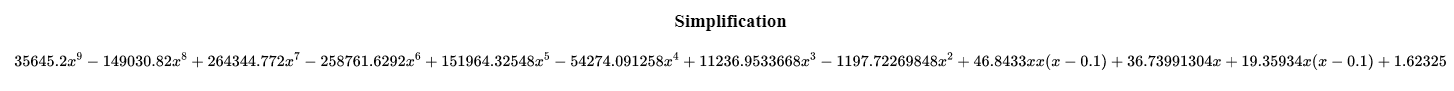
**Висновки**

У результаті виконання лабораторної роботи обчислено значення таблично заданої функції у точці x0 (x0 = 1.16887), використовуючи інтерполяційні поліноми Лагранжа та Ньютона. Методи Лагранжа та Ньютона націлені на знаходження інтерполяційного поліному. У випадку, якщо вони використовуються для знаходження інтерполяційного поліному однієї функції, двома методами виходить однаковий поліном.

Інтерполяційний поліном Лагранжа:



Інтерполяційний поліном Ньютона:



**Додаток**

#include <iostream>

#include <vector>

#include <cmath>

using namespace std;

// функція для знаходження полінома Лагранжа

double lagrange\_interpolation(double x, const vector<double>& x\_values, const vector<double>& y\_values) {

double result = 0;

for (int i = 0; i < x\_values.size(); i++) {

double term = y\_values[i];

for (int j = 0; j < x\_values.size(); j++) {

if (i != j) {

term \*= (x - x\_values[j]) / (x\_values[i] - x\_values[j]);

}

}

result += term;

}

return result;

}

// функція для знаходження полінома Ньютона

double newton\_interpolation(double x, const vector<double>& x\_values, vector<double>& y\_values) {

int n = x\_values.size();

vector<double> div\_diff(n);

div\_diff[0] = y\_values[0];

for (int i = 1; i < n; i++) {

for (int j = n - 1; j >= i; j--) {

y\_values[j] = (y\_values[j] - y\_values[j - 1]) / (x\_values[j] - x\_values[j - i]);

}

div\_diff[i] = y\_values[i];

}

double result = div\_diff[n - 1];

for (int i = n - 2; i >= 0; i--) {

result = div\_diff[i] + (x - x\_values[i]) \* result;

}

return result;

}

int main() {

vector<double> x\_values = { 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 };

vector<double> y\_values = { 1.623250, 1.664792, 1.701977, 1.734832, 1.763404, 1.787764, 1.808002, 1.824230, 1.836580, 1.845261 };

// введення даних у вигляді таблиці

cout << "x\_values\ty\_values" << endl;

for (int i = 0; i < x\_values.size(); i++) {

cout << x\_values[i] << "\t\t" << y\_values[i] << endl;

}

double x0 = 1.168869;

double lagrange\_result = lagrange\_interpolation(x0, x\_values, y\_values);

double newton\_result = newton\_interpolation(x0, x\_values, y\_values);

cout << "x0 = " << x0 << endl;

cout << "x0: " << endl;

cout << "Lagrange method: " << lagrange\_result << endl;

cout << "Newton method: " << newton\_result << endl;

return 0;

}