Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій



**ЗВІТ**

**Про виконання лабораторної роботи № 8**

«Наближення дискретних (таблично заданих) функцій»

**з дисципліни «Чисельні методи»**

**Лектор:**

доцент кафедри ПЗ

Мельник Н.Б.

**Виконав:**

студ. групи ПЗ-15

Марущак А. С.

**Прийняв:**

асистент кафедри ПЗ

Гарматій Г.Ю

«\_\_\_» \_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2022

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

**Мета роботи:** ознайомитися з методом інтерполяції таблично заданих функцій.

**Теоретичні відомості**

Найпростіша задача інтерполяції полягає в тому, що на відрізку [a,b] задано ( n +1 ) точок , які називають вузлами інтерполяції, і значення деякої функції у цих точках



Необхідно побудувати інтерполяційну функцію F(x) , яка приймає у вузлах інтерполяції ті самі значення, що й функція f (x) . Тобто треба знайти таку функцію F(x) , щоб



Геометрично це означає, що треба знайти криву y = F(x) певного типу, яка проходить через задану систему точок.

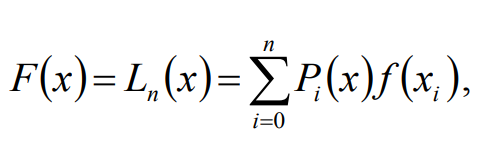
Зауважимо, що через задану множину точок можна провести безліч гладких кривих. Тому задача інтерполяції є неоднозначною. Вона стає однозначною тоді, коли за інтерполяційну функцію вибрати поліном n-го степеня, де степінь полінома на одиницю менший від к-ті вузлів інтерполяції, і такий, що виконуються умови



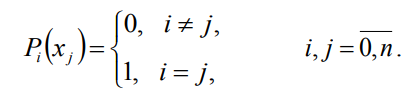
**Інтерполяційний поліном Лагранжа**

Один з методів знаходження інтерполяційного полінома запропонував Лагранж. Основна ідея цього методу полягає в пошуку полінома, який в одному довільному вузлі інтерполяції приймає значення один, а в усіх інших вузлах – нуль.

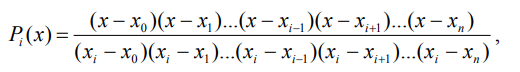
Наближену функцію y = F(x) розглянемо у вигляді



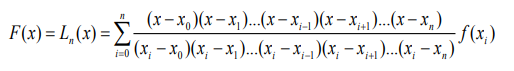
де - такий многочлен, що



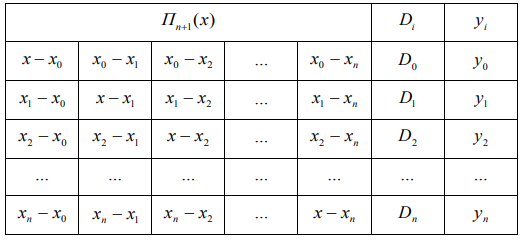
Оскільки точки є коренями полінома, то його можна записати у такому вигляді



а наближена функція F(x) , яку називають інтерполяційним многочленом Лагранжа, матиме вигляд



Для запису інтерполяційного полінома Лагранжа зручно використовувати таблицю:

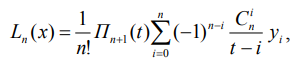


Тут – добуток елементів i –го рядка, – добуток елементів головної діагоналі.

Тоді поліном Лагранжа можна записати у вигляді



У випадку рівновіддалених вузлів вираз набуде форми



де t – крок інтерполяції.

**Інтерполяційний поліном Ньютона**

Інший спосіб розв’язування задачі інтерполяції запропонував Ньютон. Цей спосіб полягає в тому, що поліном для загального випадку нерівновіддалених вузлів записують у вигляді



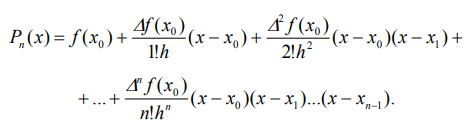
де

– розділена різниця 1-го порядку,

– розділена різниця 2-го порядку,

– розділена різниця n-го порядку.

Для випадку рівновіддалених вузлів маємо вираз



де – скінченна різниця першого порядку і обчислюється наступним чином



Для обчислення скінченної різниці другого порядку використовуємо скінченні різниці першого порядку



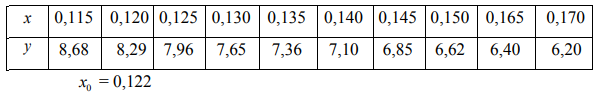
Аналогічно запишемо скінченну різницю n -го порядку



**Індивідуальне завдання**

1. Ознайомитися з теоретичним матеріалом.
2. Використовуючи інтерполяційні поліноми Лагранжа та Ньютона, обчислити значення табличної заданої функції у точці .

**Варіант завдання**



**Хід роботи**

Проведемо деякі обчилення вручну для кращого розуміння вивченого матеріалу.

По-перше, як бачимо, вузли не є рівновіддаленими, тому знаходити поліном будемо як для загального випадку.

По-друге, як бачимо, ми маємо 10 точок, тому інтерполяційні поліноми Лагранжа та Ньютона будуть 9-го степеня.

Поліном Лагранжа представлятиме собою суму 10 доданків, кожен з яких – добуток коефіцієнта Лагранжа на значення функції в певній точці. Наприклад, для першої точки цей доданок матиме вигляд

Очевидно, що проводити таку к-ть обчислень вручну не є доцільним, тому решту обчислень проведемо програмно.

Для знаходження інтерполяційного поліному Ньютона нам необхідно знайти розділені різниці. Знову ж таки, знаходження всіх розділених різниць потребує довгих обчислень, тому покажемо обчислення лише деяких:

Тоді перші доданки інтерполяційного поліному Ньютона запишемо наступним чином

Наступна мета – реалізувати подані методи програмно, використовуючи отримані знання, для того щоб завершити розрахунки.

Для легшої роботи створимо структуру даних Polynom для керування поліномами, в ній реалізуємо операції множення та додавання поліномів наступним чином:

public static Polynom operator+(Polynom a, Polynom b)

{

Polynom result = new Polynom(Math.Max(a.Coefs.Length, b.Coefs.Length));

for(int i = 0; i < Math.Max(a.Coefs.Length, b.Coefs.Length); i++)

{

result.Coefs[i] = a.Coefs.ElementAtOrDefault(i) + b.Coefs.ElementAtOrDefault(i);

}

return result;

}

public static Polynom operator\*(Polynom a, Polynom b)

{

Polynom result = new Polynom(a.Coefs.Length + b.Coefs.Length - 1);

for(int i = 0; i < a.Coefs.Length; i++)

{

for(int j = 0; j < b.Coefs.Length; j++)

{

result.Coefs[i + j] += a.Coefs[i] \* b.Coefs[j];

}

}

return result;

}

І тоді отримання поліномів Лагранжа та Ньютона опишемо наступним чином:

public static Polynom GetLangrange(KeyValuePair<decimal, decimal>[] xypairs)

{

Polynom res = new Polynom(xypairs.Length);

for(int i = 0; i < xypairs.Length; i++)

{

Polynom L\_i = 1; //Поліном-доданок

for(int j = 0; j < xypairs.Length; j++)

{

if (i == j) continue; // Пропускаємо, якщо номер доданку дорівнює номеру точки

Polynom numerator = new Polynom(new decimal[] { -xypairs[j].Key, 1 });

decimal denominator = (xypairs[i].Key - xypairs[j].Key);

L\_i \*= numerator \* (1M / denominator);

}

res += L\_i \* xypairs[i].Value;

}

//Повертаємо результат

return res;

}

public static Polynom GetNewton(KeyValuePair<decimal, decimal>[] xypairs)

{

//Масив-таблиця для зберігання розділених різниць

decimal[,] diff = new decimal[xypairs.Length, xypairs.Length + 1];

for(int i = 0; i < xypairs.Length+1; i++)

{

for(int j = 0; j < xypairs.Length - i; j++)

{

if (i == 0) diff[j, i] = xypairs[j].Value;

else

{

//Для обчислення різниці i-го порядку використовуємо різницю (i-1)-го порядку

decimal num = (diff[j + 1, i - 1] - diff[j, i - 1]);

decimal denom = (xypairs[j + i].Key - xypairs[j].Key);

diff[j, i] = num / denom;

}

}

}

Polynom result = new Polynom();

for(int i = 0; i < xypairs.Length; i++)

{

Polynom P\_i = diff[0, i]; //Поліном-доданок

for (int j = 0; j < i; j++)

P\_i \*= new Polynom(new decimal[] { -xypairs[j].Key, 1 });

result += P\_i;

}

//Повертаємо результат

return result;

}

Подамо 10 точок з варіанту завдання на вхід програми і подивимось на результат

**Результат** виконання програми:

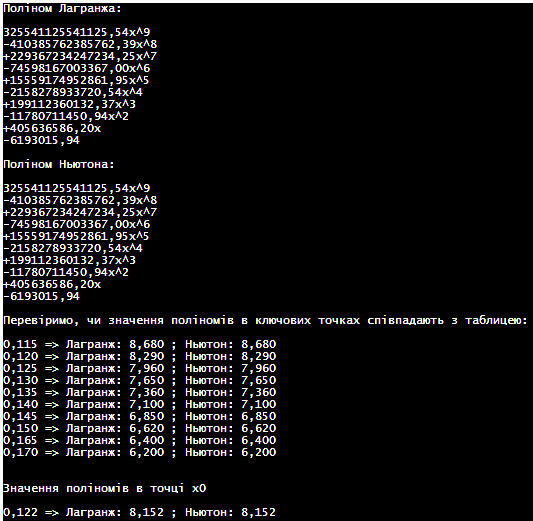


Рис 1.1 Результат виконання програми.

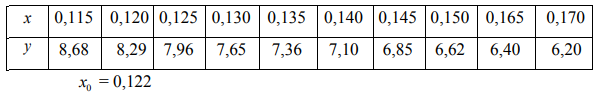
**Аналіз результатів:**

Як бачимо, поліноми, отримані обома методами, співпали. До того ж, значення поліномів в ключових точках збігаються з табличними. Це свідчить про те, що завдання з високою вірогідністю було виконано правильно. Також, можемо бачити, що лежить між 2-им та 3-ім значенням аргументів в таблиці. Враховуючи, що при функція монотонно спадає, те, що значення функції лежить в межах значень функції в 2-ій та 3-ій точках таблиці є очікуваним.

**Висновок:**

Виконуючи цю лабораторну роботу, ми ознайомилися з методом інтерполяції таблично заданих функцій.

За допомогою цих знань ми реалізували програму, що змогли представити таблично задану функцію



у вигляді поліному методами Ньютона та Лагранжа.

Також, ми знайшли значення цього поліному у точці і отримали, що .