МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Кафедра дискретного аналізу

**Нечітка логіка**

**ЛАБОРАТОРНА РОБОТА №1**

Виконав:

Студент групи ПМі-43

Тимчишин Ярема

Викладач:

ас. Прядко О.

2025

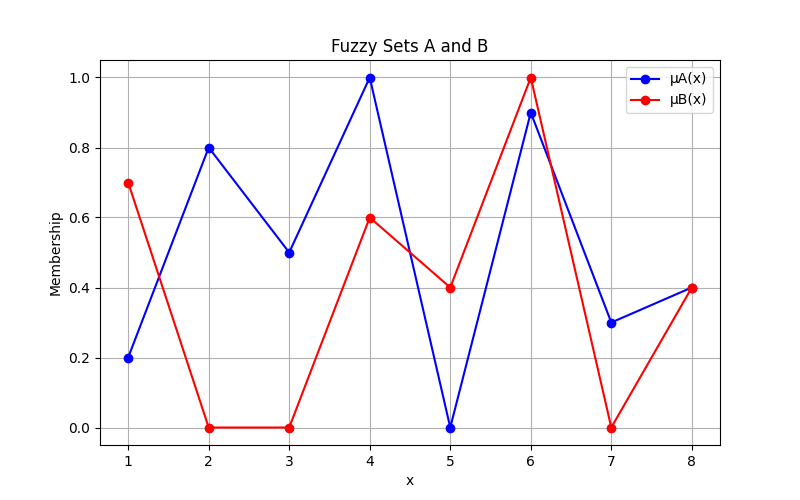
**Варіант 21**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| μA(x) | 0.2 | 0.8 | 0.5 | 1 | 0 | 0.9 | 0.3 | 0.4 |
| μB(x) | 0.7 | 0 | 0 | 0.6 | 0.4 | 1 | 0 | 0.4 |

**Хід роботи**

Для виконання даної лабораторної роботи використав мову програмування Python та середовище PyCharm, де реалізував програму.

1. **Представлення множин А і В геометрично.**



В програмі реалізував це наступним чином.

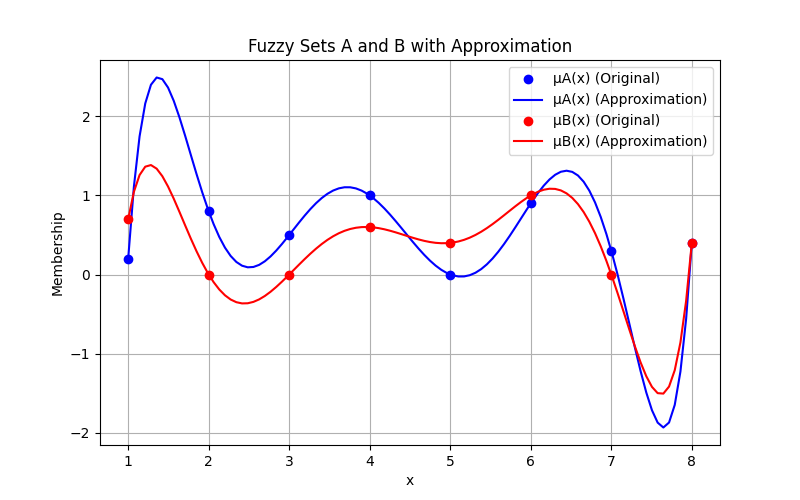
Спочатку під’єднав необхідні бібліотеки: *numpy (np)* – для роботи з масивами та векторними обчисленнями, *matplotlib.pyplot (plt)* – для побудови графіків, та *scipy.interpolate (lagrange)* – для знаходження полінома Лагранжа для апроксимації нечітких множин.

Далі реалізував функцію *plot\_fuzzy\_sets* для побудови графіків нечітких множин A, B і C, де *x* – вхідні значення (від 1 до 8), *A, B, C* – нечіткі множини, *title* – назва графіка, *colors* – кольори ліній для кожної множини (bo- – синій, ro- – червоний, mo- – малиновий), *labels* – підписи для графіків. Потім створив нову фігуру розміром 8x5 дюймів. Оголосив змінні *used\_colors* і *used\_labels*, щоб уникнути помилок при роботі з необов’язковими аргументами. І якщо множина A передана, програма будує її графік. Аналогічно для B та C. В кінці додаю легенду, сітку та виводжу графік.

A computer screen shot of a program

AI-generated content may be incorrect.

1. **Представлення їх апроксимацій графічно та аналітично**



A black background with white text

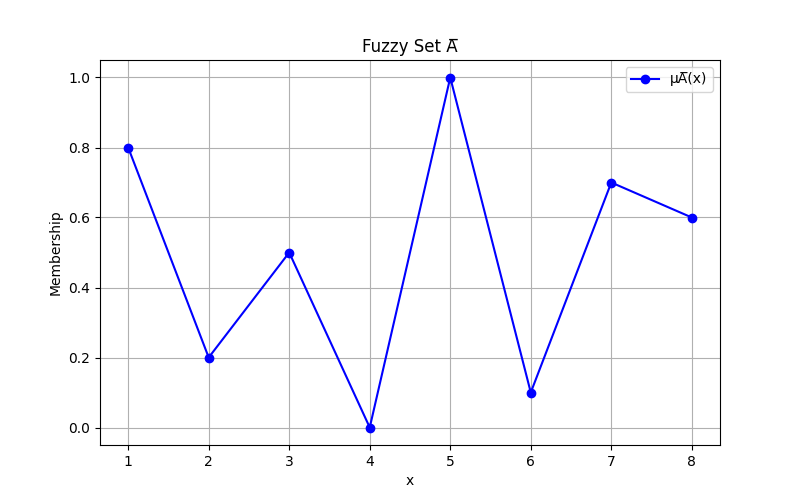
AI-generated content may be incorrect.

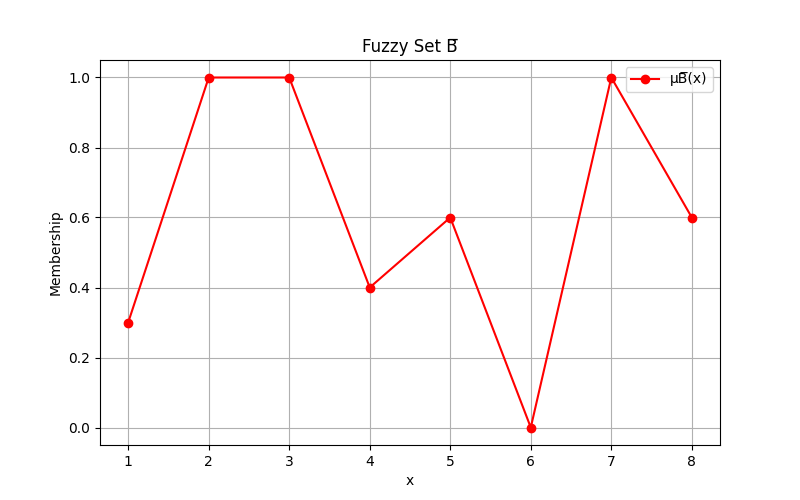
Для цього реалізував функцію *fuzzy\_approximation\_and\_plot*, яка апроксимує нечіткі множини A та B за допомогою полінома Лагранжа і будує їх графіки. Викликаю *lagrange(x, y)*, яка знаходить поліном Лагранжа. Далі використовую *np.poly1d(poly)*, яка гарно форматує поліном для друку. Потім обчислюю поліноми для A і B. Далі оголошую *x\_dense* – 100 точок між 1 і 8 для побудови гладкого графіка апроксимації. *A\_poly(x\_dense)*, *B\_poly(x\_dense)* – обчислює значення поліномів у цих точках. Після чого використовую *plt.plot* для побудови графіків апроксимацій.

A computer screen shot of a program

AI-generated content may be incorrect.

1. **Знаходження функцій приналежності і геометричне представлення множин А̅ , В̅ , А ∪ В, А ∩ В. Операції алгебраїчного перетину, алгебраїчного об’єднання, різниці, симетричної різниці, концентрування, розведення, граничного перетину та граничного об’єднання.**





A graph with lines and dots

AI-generated content may be incorrect.

A graph with lines and dots

AI-generated content may be incorrect.

A graph with lines and dots

AI-generated content may be incorrect.

A graph with lines and dots

AI-generated content may be incorrect.

A graph with lines and dots

AI-generated content may be incorrect.

A graph with lines and dots

AI-generated content may be incorrect.

A graph with blue lines

AI-generated content may be incorrect.

A graph with red lines

AI-generated content may be incorrect.

A graph with blue lines

AI-generated content may be incorrect.

A graph with red lines

AI-generated content may be incorrect.

A graph with red and blue lines and dots

AI-generated content may be incorrect.

A graph with lines and dots

AI-generated content may be incorrect.

В програмі реалізував це наступним чином.

Оголосив функцію *fuzzy\_operations*, яка міститиме всі потрібні операції: *A\_complement* (А̅), *B\_complement* (В̅), *union* (А ∩ В), *intersection* (А ∪ В), *difference* (різниця A та B), *symmetric\_difference* (симетрична різниця A та B), *algebraic\_intersection* (алгебраїчний перетин), *algebraic\_union* (алгебраїчне об’єднання), *concentration\_A* (концентрування множини A), *concentration\_B* (концентрування множини B), *dilution\_A* (розведення множини A), *dilution\_B* (розведення множини B), *boundary\_intersection* (граничний перетин A та B), *boundary\_union* (граничне об’єднання A та B).

Функція повертає словник, який містить результати операцій, відповідні кольори для графіків та заголовки.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen with many white and green text

AI-generated content may be incorrect.

1. **Виведення характеристик для множин A та B: носія нечіткої множини, ядра, альфа рівня (який задає користувач) та висоти.**

A screenshot of a computer program

AI-generated content may be incorrect.

Для цього в програмі оголосив функцію *characteristics*, яка обчислює:

* **Носій (support)** – всі елементи, де μA(x) > 0.
* **Ядро (core)** – всі елементи, де μA(x) = 1.
* **Висоту (height)** – максимальне значення функції приналежності.
* **α-рівень** – всі елементи, де μA(x) ≥ α.

Після чого перетворює множини у списки чисел, та повертає їх.

1. **Виклик функцій.**

Спершу визначив нечіткі множини. Далі викликав функції *plot\_fuzzy\_sets*, *fuzzy\_approximation\_and\_plot* та *operations*, для побудови графіків, їх апроксимації та операцій з ними. Далі за допомогою *alpha\_A* та *alpha\_B* отримуємо рівні *α* від користувача, обчислюємо характеристики множин та виводимо їх на консоль.

A computer screen shot of a program

AI-generated content may be incorrect.