

Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 1

з дисципліни «Методи та технології штучного інтелекту»

На тему:

«Дослідження способів формування нечітких множин і операцій над ними»

Виконала:
студентка групи ІС-12.
Мельникова К.О.

Перевірив:
Шимкович В. М.

Мета: Побудувати нечіткі множини з використанням різних типів функцій приналежності. Виконати найбільш поширені логічні операції над нечіткими множинами.

Завдання:

За допомогою пакетів моделювання або мови програмування високого рівня:

1. Побудувати трикутну і трапецієподібну функцію приналежності.
2. Побудувати просту і двосторонню функцію приналежності Гаусса, утворену за допомогою різних функцій розподілу.
3. Побудувати функцію приналежності "узагальнений дзвін", яка дозволяє представляти нечіткі суб'єктивні переваги.
4. Побудувати набір сігмоїдних функцій: основну односторонню, яка відкрита зліва чи справа; додаткову двосторонню; додаткову несиметричну.
5. Побудувати набір поліноміальних функцій приналежності (Z-, PI- і S функцій).
6. Побудувати мінімаксну інтерпретацію логічних операторів з використанням операцій пошуку мінімуму і максимуму.
7. Побудувати вірогідну інтерпретацію кон'юнктивну і диз'юнктивних операторів.
8. Побудувати доповнення нечіткої множини, яке описує деяке розмите судження і представляє собою математичний опис вербального вираження, який заперечує це нечітка множина.

При виконанні пунктів 1 - 8 індивідуального завдання, значення змінних a , b , c , d і т.д. необхідно вибирати довільним чином.

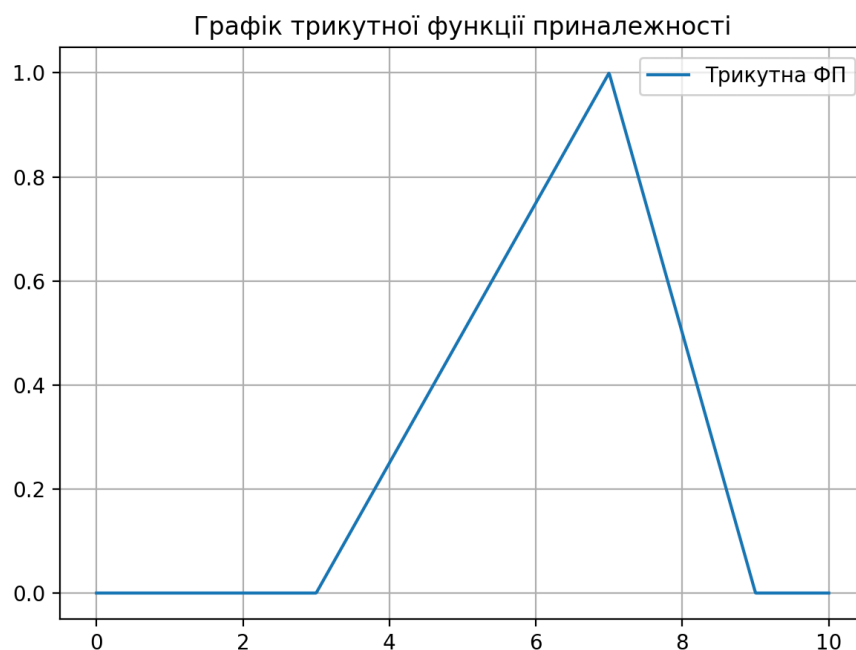
Виконання роботи:

Спочатку імпортуємо бібліотеки та створимо змінну, що містить діапазон значень X:

```
import matplotlib.pyplot as plt
import skfuzzy as fuzz
import numpy as np

# creates a range of x values
x = np.linspace(0, 10, 1000)
```

1. Трикутна функція приналежності:



```
# triangle
triangle = fuzz.trimf(x, [3, 7, 9])
plt.figure(figsize=(7, 5))
plt.plot(x, triangle, label='Трикутна ФП')
plt.legend()
plt.grid(True)
plt.title('Графік трикутної функції приналежності')
plt.show()
```

2. Графік трапецієподібної функції приналежності:



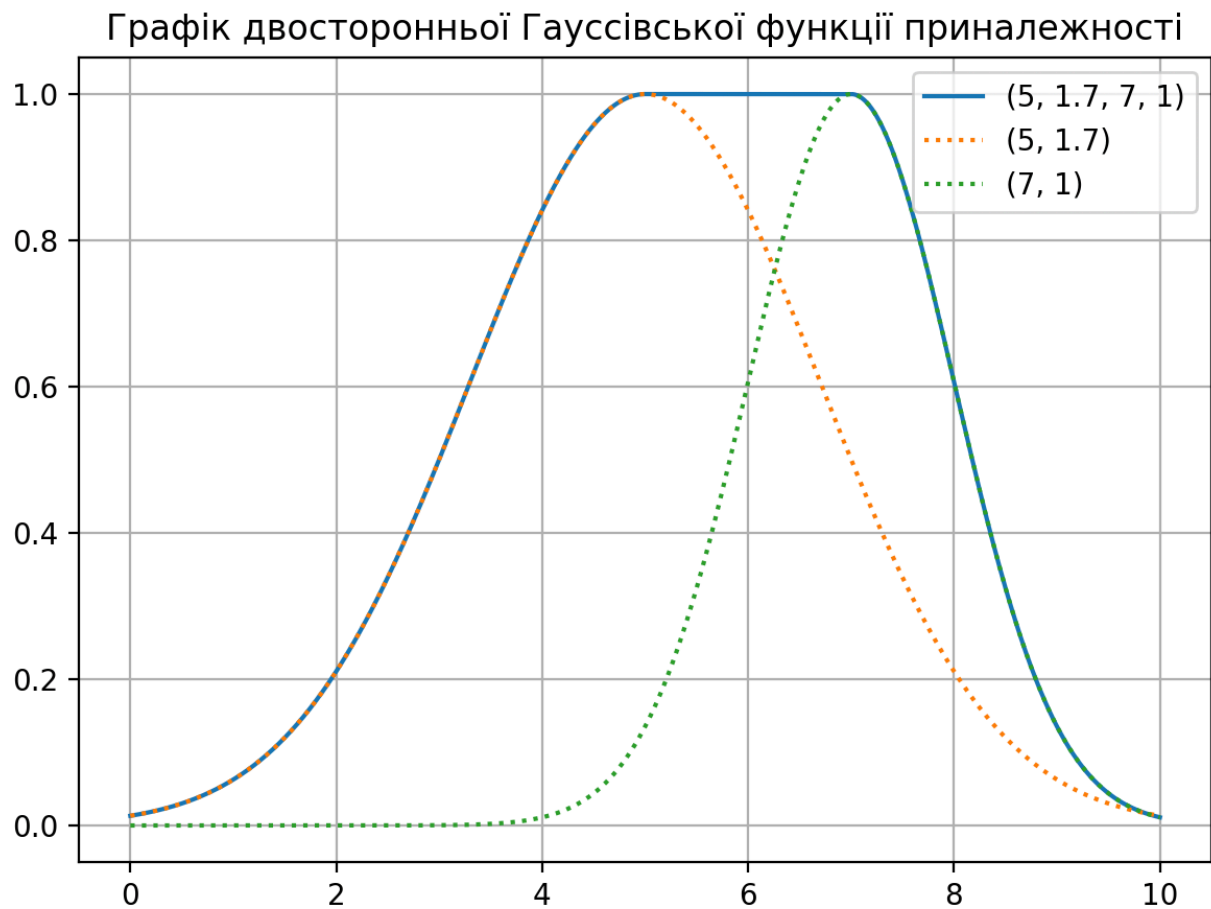
```
# trapeze
trapeze = fuzz.trapmf(x, [3, 5, 7, 9])
plt.figure(figsize=(7, 5))
plt.plot(x, trapeze, label='Трапецієподібна ФП')
plt.legend()
plt.grid(True)
plt.title('Графік трапецієподібної функції приналежності')
plt.show()
```

3. Графік простої Гауссівської функції приналежності:



```
# gaussian simple function
gauss1 = fuzz.gaussmf(x, 6, 1)
plt.figure(figsize=(7, 5))
plt.plot(x, gauss1, label='Проста Гауссівська ФП')
plt.legend()
plt.grid(True)
plt.title('Графік простої Гауссівської функції приналежності')
plt.show()
```

4. Графік двосторонньої Гауссівської функції приналежності



```
# two-sided Gaussian membership functions
gauss21 = fuzz.gauss2mf(x, 5, 1.7, 7, 1)
gauss22 = fuzz.gaussmf(x, 5, 1.7)
gauss23 = fuzz.gaussmf(x, 7, 1)
plt.figure(figsize=(7, 5))
plt.plot(x, gauss21, label='(5, 1.7, 7, 1)')
plt.plot(x, gauss22, linestyle='dotted', label='(5, 1.7)')
plt.plot(x, gauss23, linestyle='dotted', label='(7, 1)')
plt.legend()
plt.grid(True)
plt.title('Графік двосторонньої Гауссівської функції приналежності')
plt.show()
```

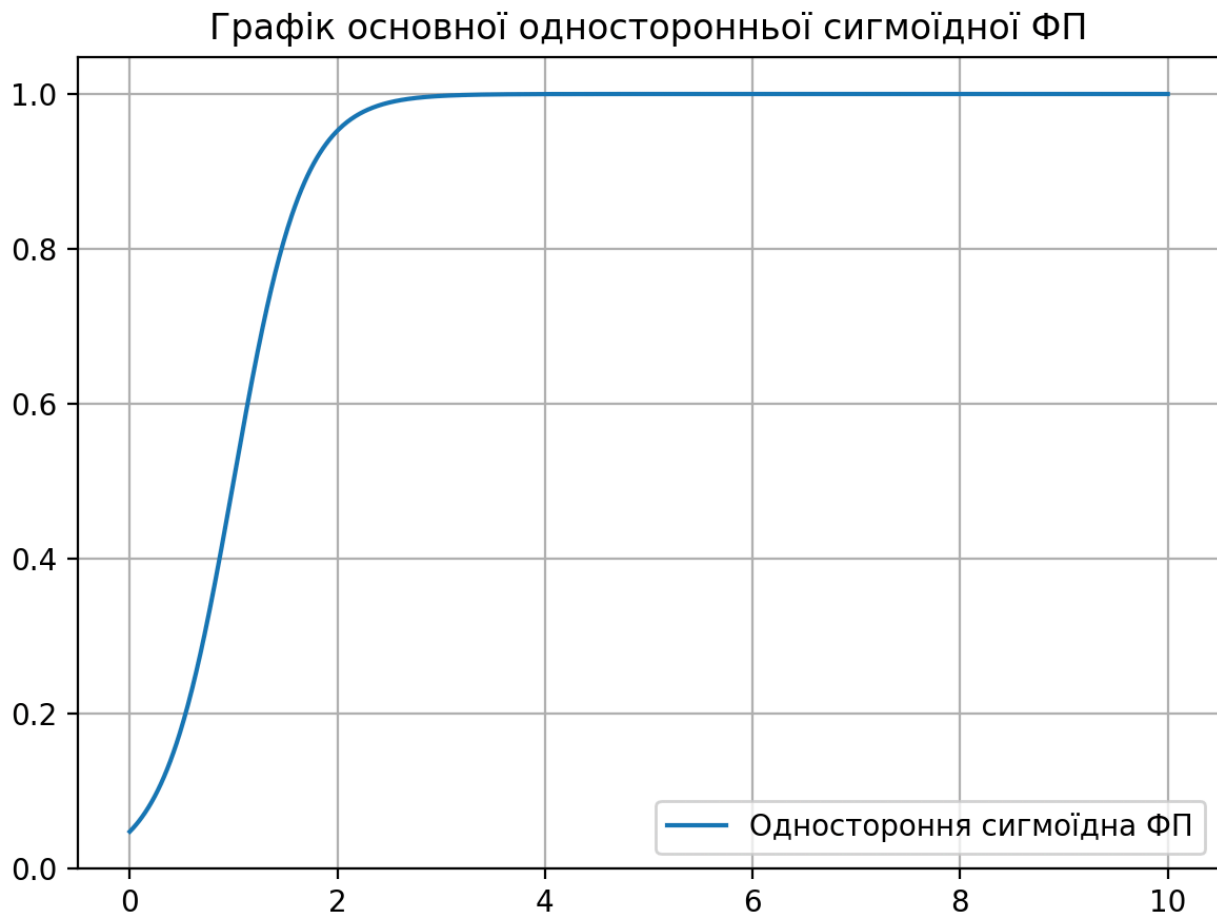
5. Графік функції приналежності "Узагальнений дзвін"



```
# Generalized Bell membership function
gbellmf_values = fuzz.gbellmf(x, 2, 3, 5)
plt.figure(figsize=(7, 5))
plt.plot(x, gbellmf_values, label='ФП "Узагальнений дзвін"')
plt.legend()
plt.grid(True)
plt.title('Графік функції приналежності "Узагальнений дзвін"')
plt.show()
```

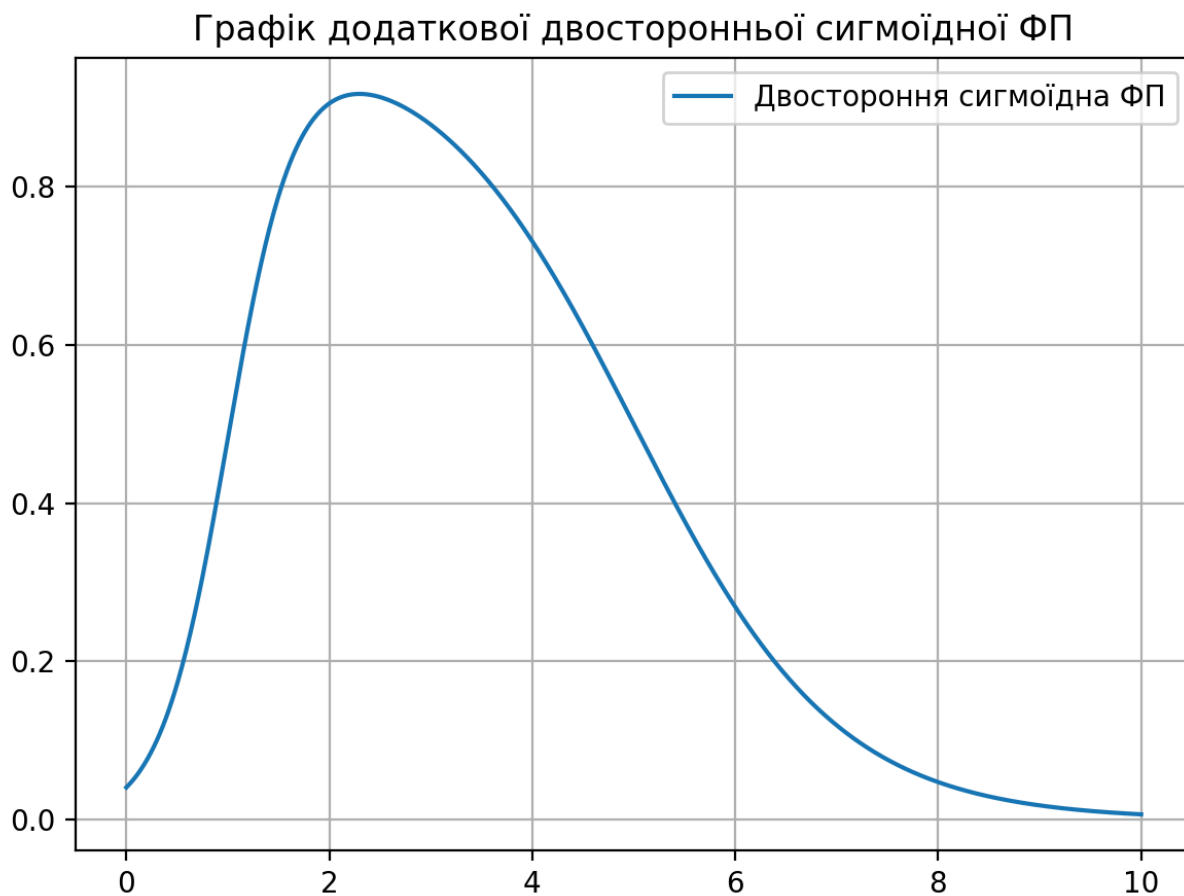
6. Сигмоїдні функції приналежності:

6.1. Основна сигмоїдна одностороння ФП:



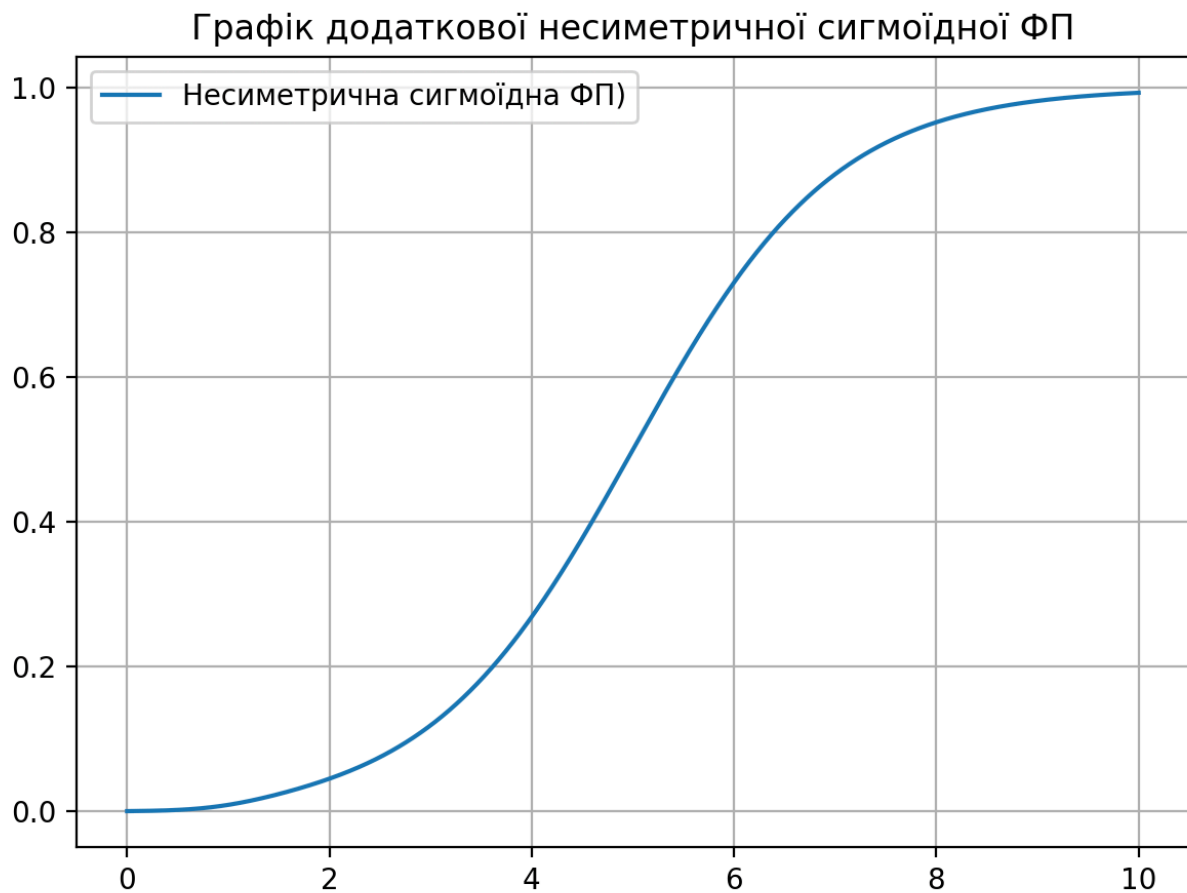
```
# Основна сигмоїдна одностороння ФП
sigm = fuzz.sigmf(x, 1, 3)
plt.figure(figsize=(7, 5))
plt.plot(x, sigm, label='Одностороння сигмоїдна ФП')
plt.legend()
plt.grid(True)
plt.title('Графік основної односторонньої сигмоїдної ФП')
plt.show()
```


6.2. Додаткова сигмоїдна двостороння ФП:



```
# Додаткова сигмоїдна двостороння ФП
sigm_dif = fuzz.dsigmf(x, 1, 3, 5, 1)
plt.figure(figsize=(7, 5))
plt.plot(x, sigm_dif, label='Двостороння сигмоїдна ФП')
plt.legend()
plt.grid(True)
plt.title('Графік додаткової двосторонньої сигмоїдної ФП')
plt.show()
```

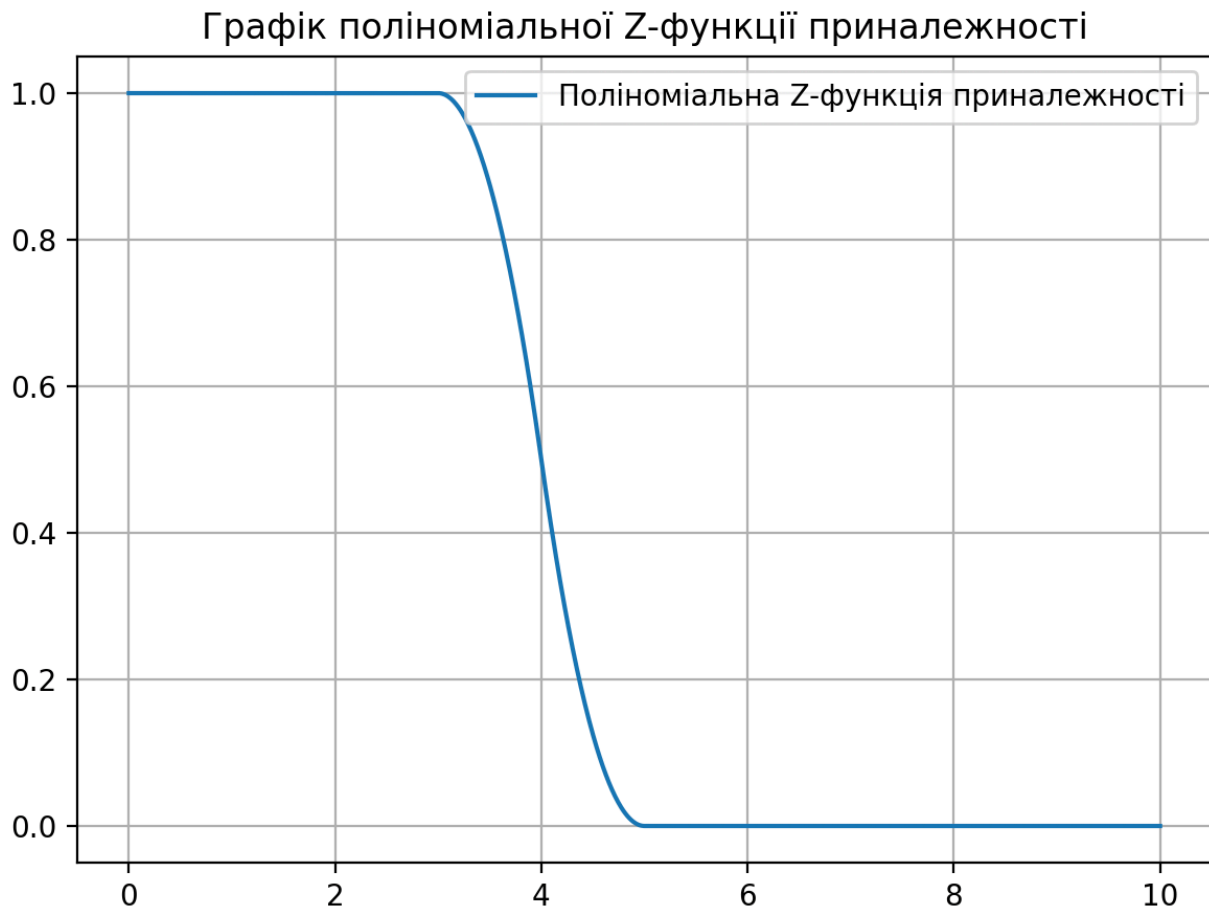
6.3. Додаткова сигмоїдна несиметрична ФП:



```
# Додаткова сигмоїдна несиметрична ФП
psigm = fuzz.psigmf(x, 1, 3, 5, 1)
plt.figure(figsize=(7, 5))
plt.plot(x, psigm, label='Несиметрична сигмоїдна ФП')
plt.legend()
plt.grid(True)
plt.title('Графік додаткової несиметричної сигмоїдної ФП')
plt.show()
```

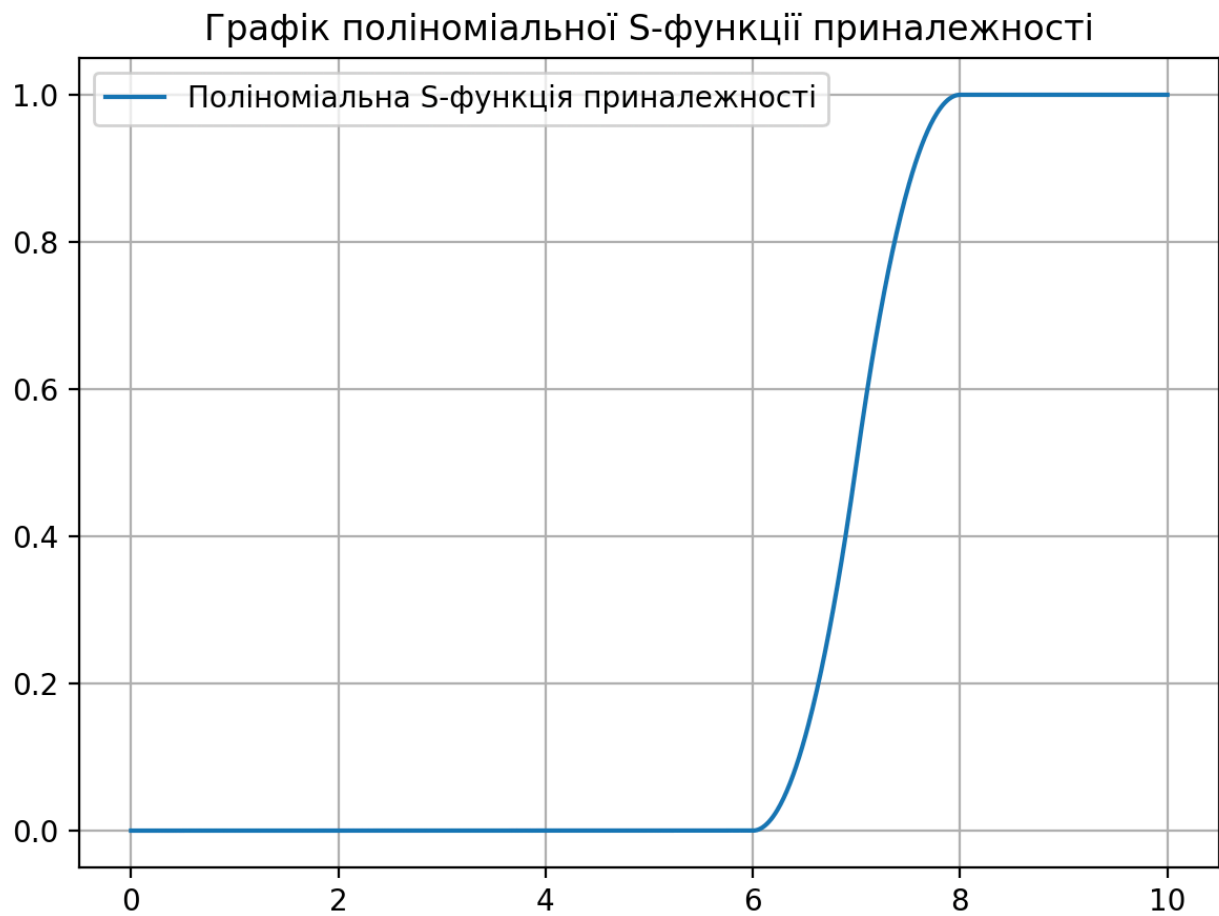
7. Поліноміальні функції приналежності:

7.1. Z-функція:



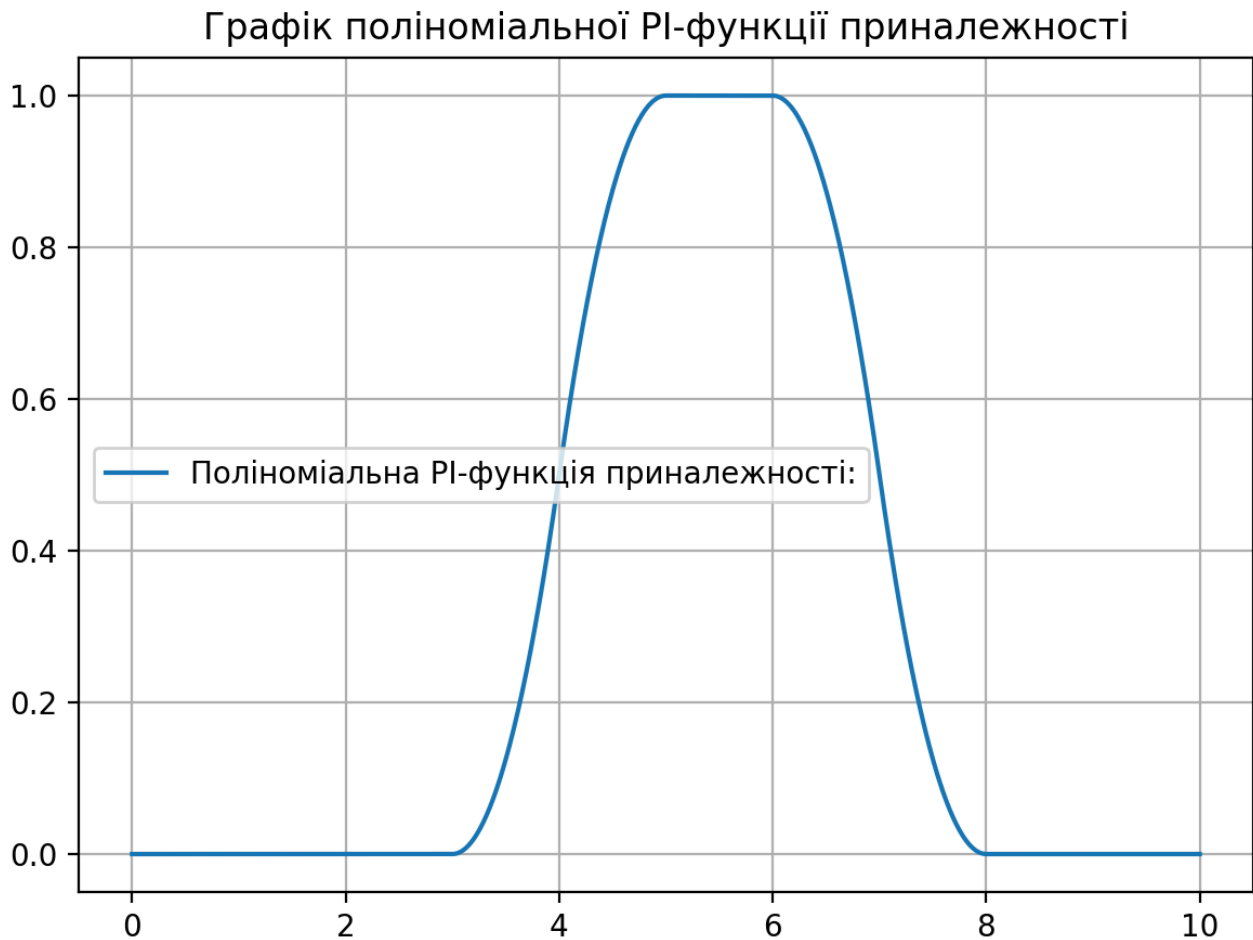
```
# Поліноміальна Z-функція приналежності
z = fuzz.zmf(x, 3, 5) # Графік
plt.figure(figsize=(7, 5))
plt.plot(x, z, label='Поліноміальна Z-функція приналежності')
plt.legend()
plt.grid(True)
plt.title('Графік поліноміальної Z-функції приналежності')
plt.show()
```

7.2. S-функція:



```
# Поліноміальна S-функція приналежності
s = fuzz.smf(x, 6, 8)
plt.figure(figsize=(7, 5))
plt.plot(x, s, label='Поліноміальна S-функція приналежності')
plt.legend()
plt.grid(True)
plt.title('Графік поліноміальної S-функції приналежності')
plt.show()
```

7.3. PI-функція:

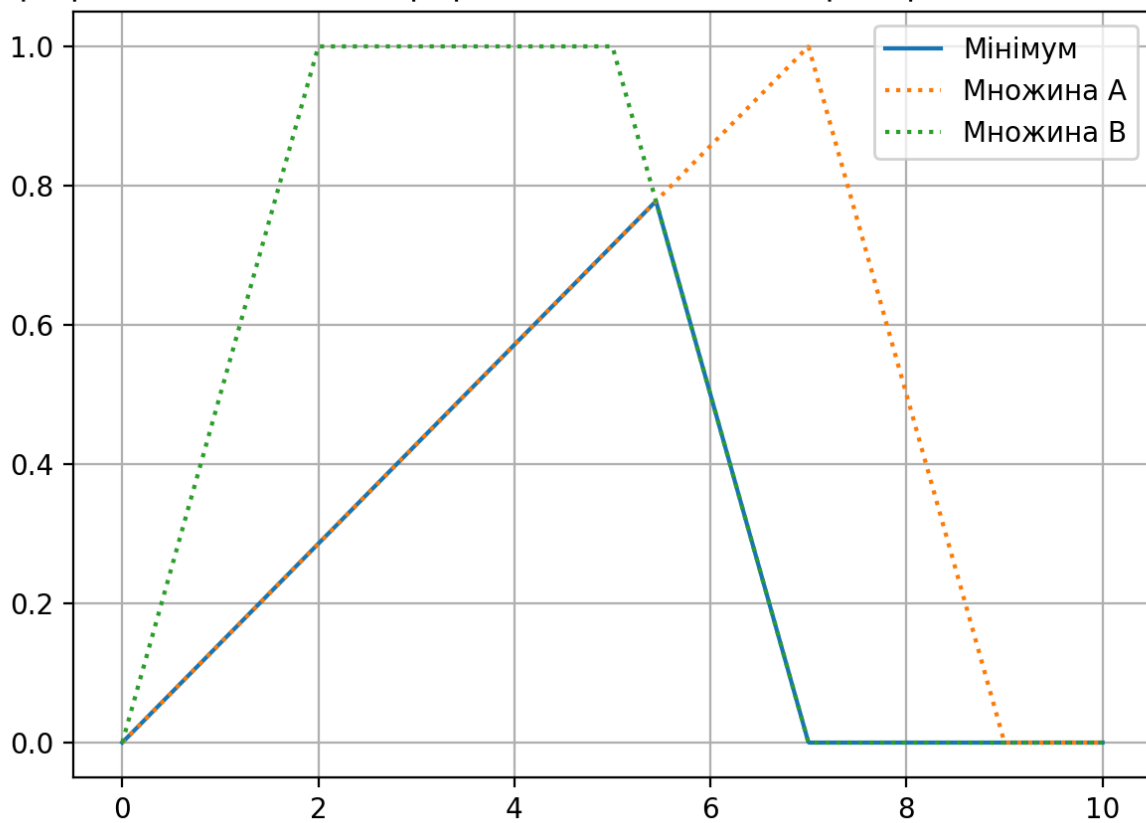


```
# Поліноміальна PI-функція приналежності
pi = fuzz.pimf(x, 3, 5, 6, 8)
# Графік
plt.figure(figsize=(7, 5))
plt.plot(x, pi, label='Поліноміальна PI-функція приналежності:')
plt.legend()
plt.grid(True)
plt.title('Графік поліноміальної PI-функції приналежності')
plt.show()
```

8. Мінімаксна інтерпретація:

8.1. Графік мінімаксної інтерпретації логічного оператора AND (minimum):

Графік мінімаксної інтерпретації логічного оператора AND (minimum)

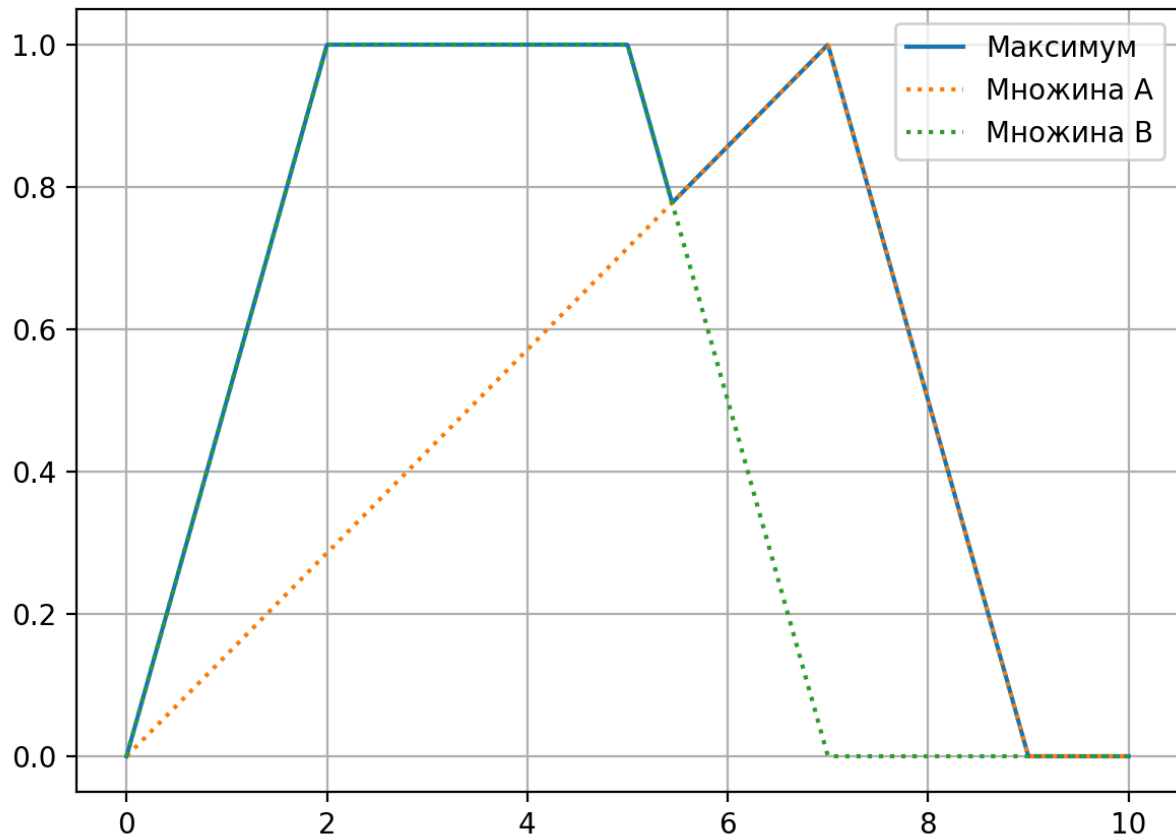


```
# мінімаксна інтерпретація
A = fuzz.trimf(x, [0, 7, 9]) # множина A
B = fuzz.trapmf(x, [0, 2, 5, 7]) # множина B

# AND (minimum)
min_values = np.fmin(A, B) # Графік
plt.figure(figsize=(7, 5))
plt.plot(x, min_values, label='Мінімум')
plt.plot(x, A, linestyle='dotted', label='Множина A')
plt.plot(x, B, linestyle='dotted', label='Множина B')
plt.legend()
plt.grid(True)
plt.title('Графік мінімаксної інтерпретації логічного оператора AND (minimum)')
plt.show()
```

8.2. Графік мінімаксної інтерпретації логічного оператора OR (maximum):

Графік мінімаксної інтерпретації логічного оператора OR (maximum)

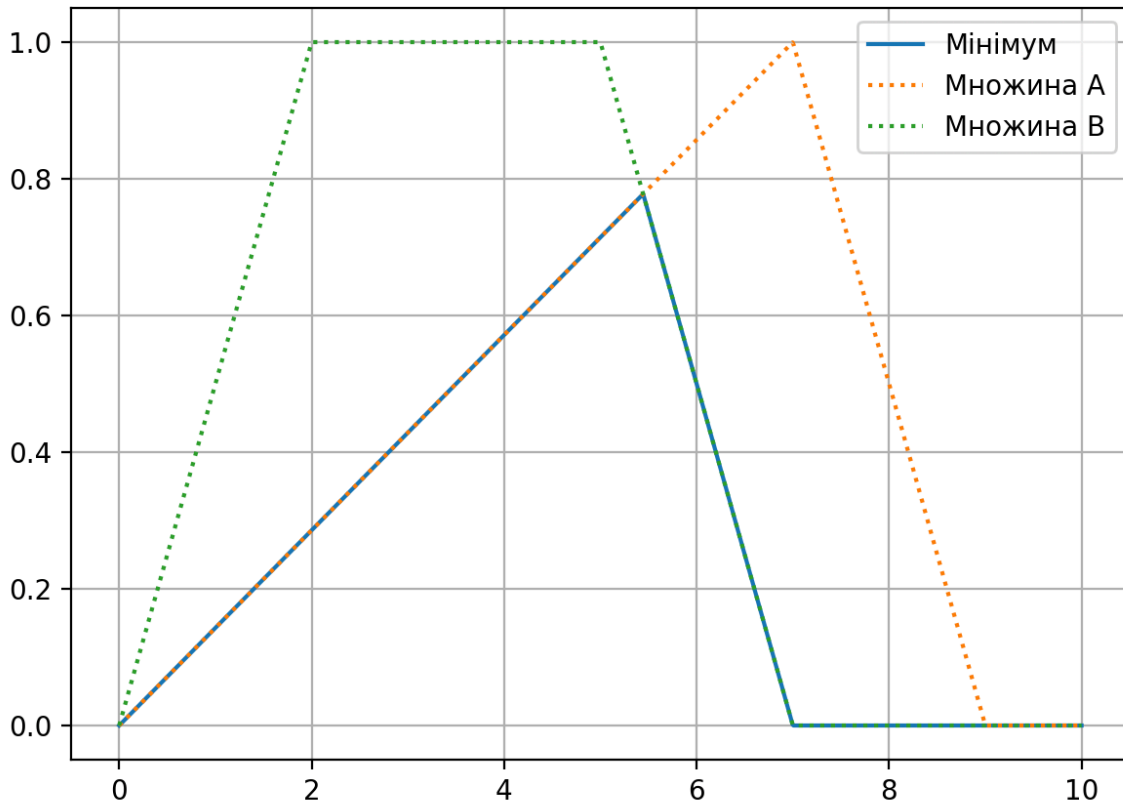


```
# OR (maximum))
max_values = np.fmax(A, B) # Графік
plt.figure(figsize=(7, 5))
plt.plot(x, max_values, label='Максимум')
plt.plot(x, A, linestyle='dotted', label='Множина A')
plt.plot(x, B, linestyle='dotted', label='Множина B')
plt.legend()
plt.grid(True)
plt.title('Графік мінімаксної інтерпретації логічного оператора OR (maximum)')
plt.show()
```

9. Ймовірнісна інтерпретація:

9.1. Кон'юнкція (AND):

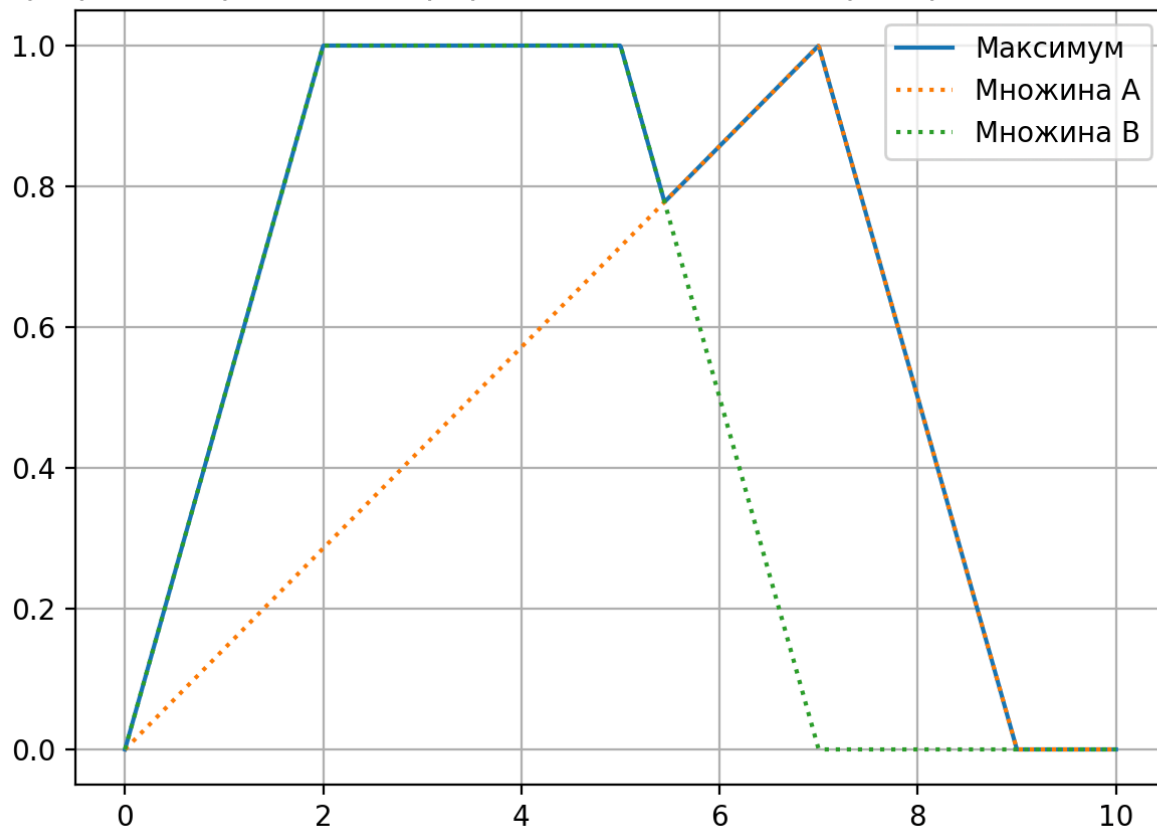
Графік імовірнісної інтерпретації логічного оператора AND (minimum)



```
# Імовірнісна інтерпретація
# AND (minimum)
values_and = fuzz.fuzzy_and(x, A, x, B)
plt.figure(figsize=(7, 5))
plt.plot(values_and[0], values_and[1], label='Мінімум')
plt.plot(x, A, linestyle='dotted', label='Множина A')
plt.plot(x, B, linestyle='dotted', label='Множина B')
plt.legend()
plt.grid(True)
plt.title('Графік імовірнісної інтерпретації логічного оператора AND (minimum)')
plt.show()
```

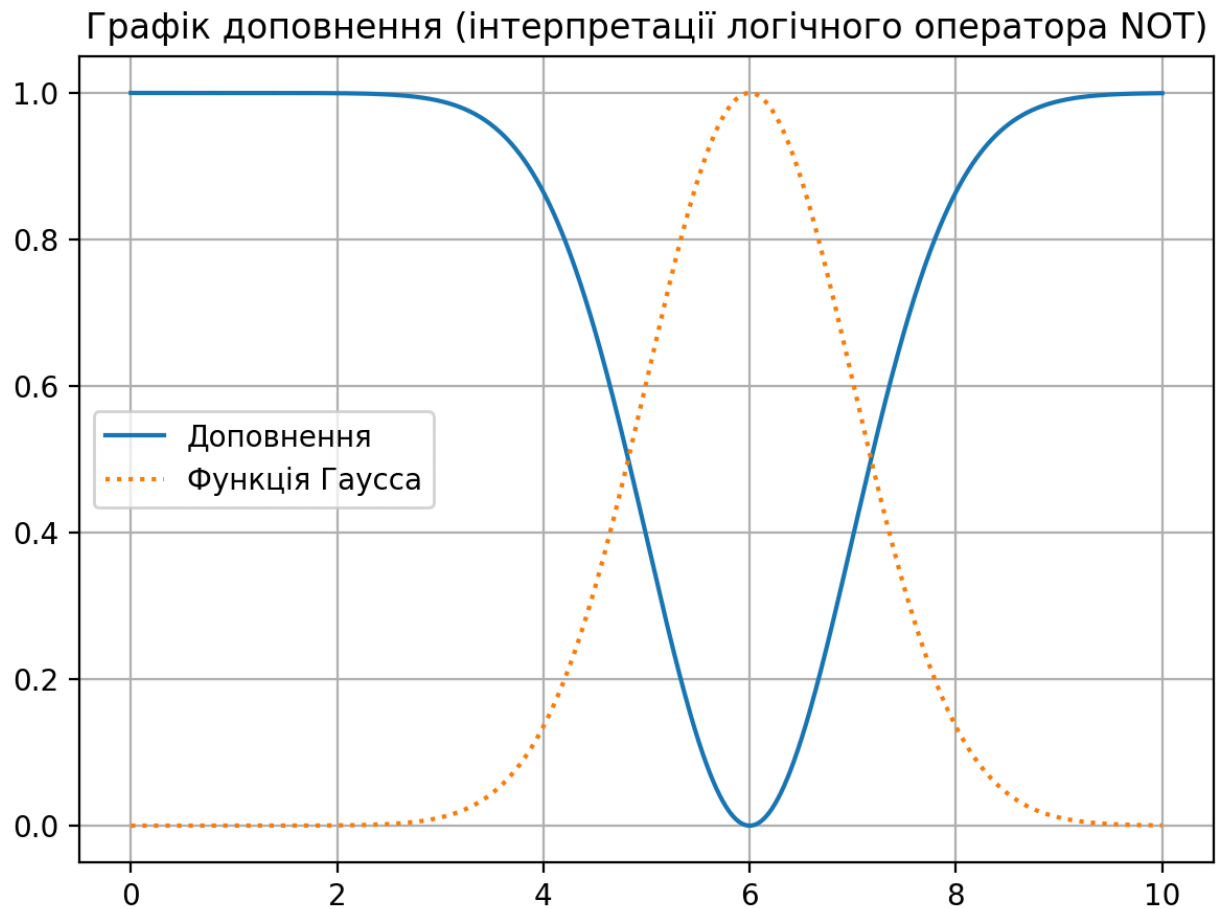

9.2. Диз'юнкція (OR):

Графік імовірнісної інтерпретації логічного оператора OR (maximum)



```
# OR (maximum)
values_or = fuzz.fuzzy_or(x, A, x, B)
plt.figure(figsize=(7, 5))
plt.plot(values_or[0], values_or[1], label='Максимум')
plt.plot(x, A, linestyle='dotted', label='Множина A')
plt.plot(x, B, linestyle='dotted', label='Множина B')
plt.legend()
plt.grid(True)
plt.title('Графік імовірнісної інтерпретації логічного оператора OR (maximum)')
plt.show()
```

10. Доповнення нечіткої множини (NOT):



```
# Доповнення нечіткої множини (NOT)
not_values = 1 - gauss1
plt.figure(figsize=(7, 5))
plt.plot(x, not_values, label='Доповнення')
plt.plot(x, gauss1, linestyle='dotted', label='Функція Гаусса')
plt.legend()
plt.grid(True)
plt.title('Графік доповнення (інтерпретації логічного оператора NOT)')
plt.show()
```

Висновок:

Під час виконання даної лабораторної роботи, я ознайомилася з методами побудови нечітких множин за допомогою різних типів функцій приналежності; розробила програму з використанням мови програмування python та бібліотек pyplot, scikit-fuzzy та numpy для побудови цих функцій (трикутної, трапецієвидної, Гауссівської, узагальненого дзвону, сигмоїдальних функцій та поліноміальних функцій приналежності). А також побудувала графіки операцій кон'юнкції (логічний оператор AND) та диз'юнкції (логічний оператор OR) в мінімаксній та імовірнісній інтерпретаціях та виконала операцію доповнення (логічний оператор NOT) для побудованих нечітких множин.