

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ

Звіт

з лабораторної роботи № 2 з дисципліни
«Методи та технології штучного інтелекту»

„Моделювання функції з двох змінних засобами нечіткої математики”

Виконав(ла)

ІІ-14 Шляхтун Денис Михайлович
(шифр, прізвище, ім'я, по батькові)

Перевірив

Шимкович Володимир Миколайович
(прізвище, ім'я, по батькові)

Київ 2023

Мета: Промодельовати засобами нечіткої логіки функцію з двох змінних. Провести дослідження форми функції приналежності на якість моделювання.

Постановка задачі.

1. Побудувати нечітку модель функції двох змінних згідно з варіантом з додатку А, що містить 6 функцій приналежності для вхідних змінних і не менше 9 для вихідної. Варіант з додатку А обрати за номер в списку журналу – 84, отже варіант 24.

| | | | | |
|-----|------------------------------------|-----|----|-----|
| 24. | $y = x \cdot \cos(2x) + \sin(x/2)$ | 24. | 7. | 18. |
| | $z = \sin(y) + \cos(x/2)$ | | | |

2. Дослідити вплив форми функції приналежності (трикутник, трапеція, Гауса) на якість моделювання (порівняти відносні помилки моделювання).

3. Дослідити можливість зменшення числа правил за рахунок виключення деяких (перевірити достатність використання правил, що представляють тільки діагональ таблиці).

4. Зробити висновки.

Виконання завдання.

Для виконання завдання була обрана мова програмування високого рівня Python.

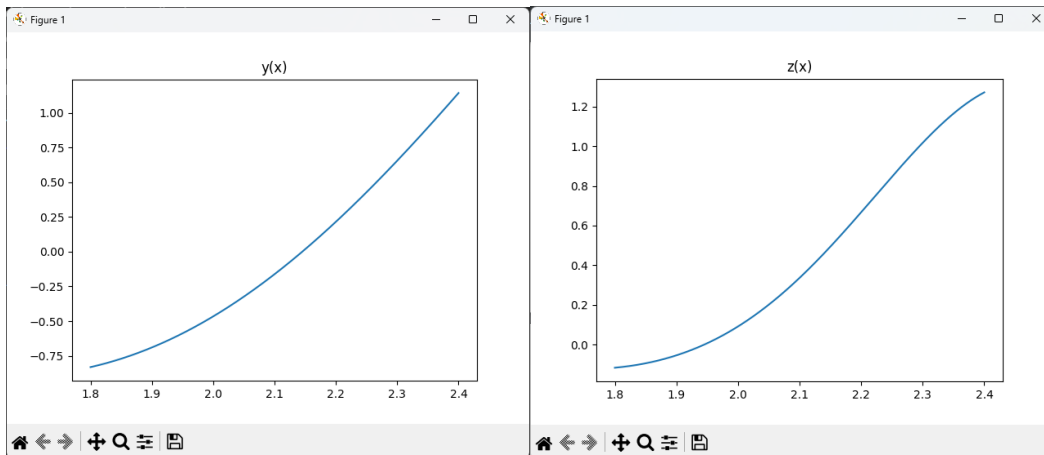
Візьмемо такі значення x , на яких функція z буде монотонною. Нехай змінна x набуватиме значень з проміжку $[1.8; 2.4]$. Значення $y(x)$ та $z(y, x)$ обчислимо за формулами відповідно до нашого варіанту, також задамо значення кількості вхідних та вихідних функцій

```

x_min = 1.8
x_max = 2.4
step = 0.001
X = np.linspace(x_min, x_max, round((x_max-x_min)/step+1))
Y = X*np.cos(2*X) + np.sin(X/2)
Z = np.sin(Y)+ np.cos(X/2)
input_mf_number = 6
output_mf_number = 12

plot_func(X, [Y], [], 'y(x)')
plot_func(X, [Z], [], 'z(x)')

```



Обчислимо координати вершин функцій:

```

X_means = np.linspace(min(X), max(X), input_mf_number)
Y_means = np.linspace(min(Y), max(Y), input_mf_number)
Z_means = np.linspace(min(Z), max(Z), output_mf_number)

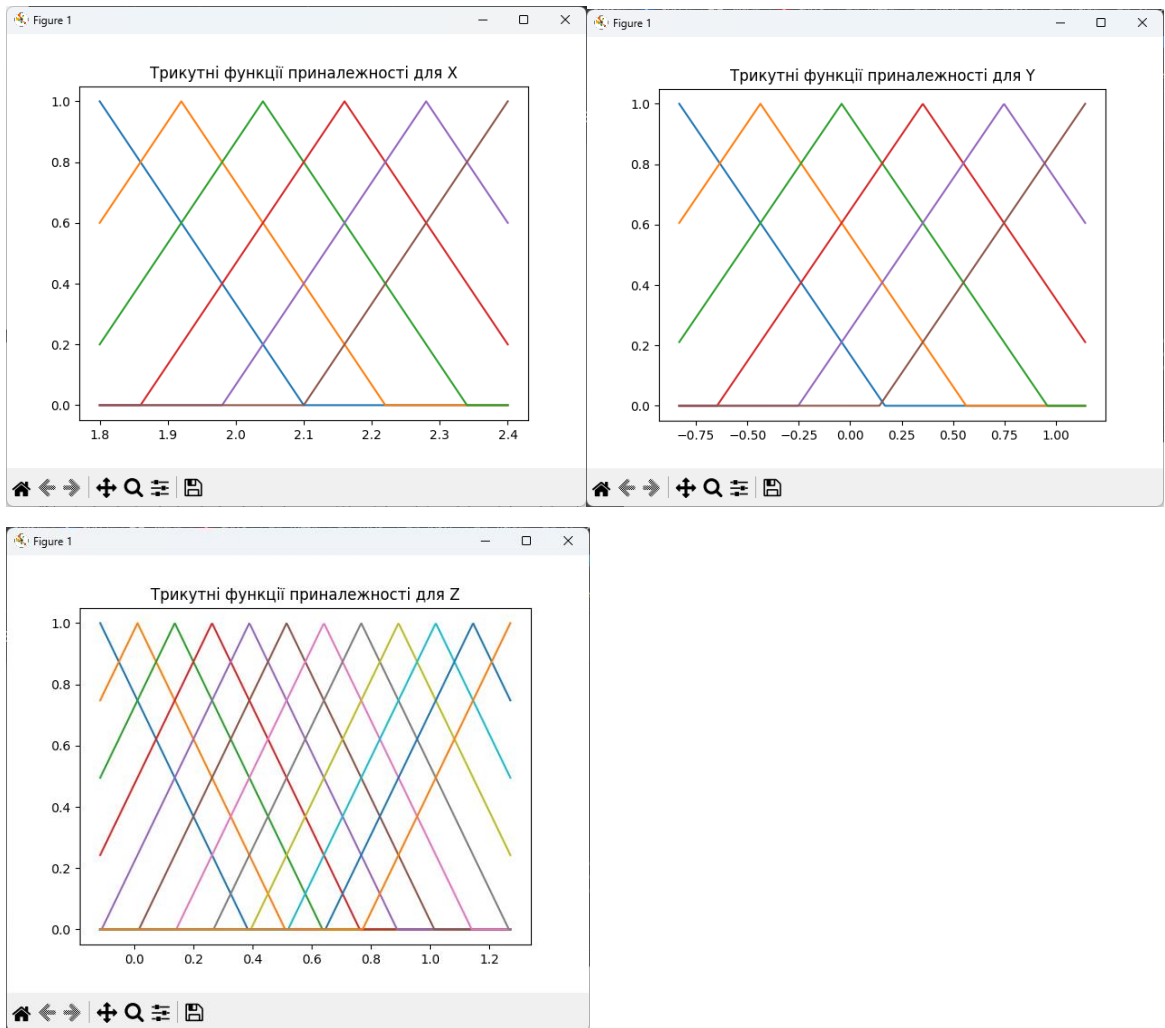
```

Побудуємо трикутні функції приналежності та візуалізуємо їх.

```

triangle_X_mf = [fuzz.trimf(X, [mean-0.3, mean, mean+0.3]) for mean in X_means]
triangle_Y_mf = [fuzz.trimf(Y, [mean-1, mean, mean+1]) for mean in Y_means]
triangle_Z_mf = [fuzz.trimf(Z, [mean-0.5, mean, mean+0.5]) for mean in Z_means]
plot_func(X, triangle_X_mf, [], 'Трикутні функції приналежності для X')
plot_func(Y, triangle_Y_mf, [], 'Трикутні функції приналежності для Y')
plot_func(Z, triangle_Z_mf, [], 'Трикутні функції приналежності для Z')

```



Опишемо функцію знаходження найбільшої функції приналежності в точці $z = \text{value}$:

```
def max_mf_num(value, mf_funcs_lists, real_values_list):
    index_in_list = min(range(len(real_values_list)), key=lambda i: abs(real_values_list[i] - value))
    values = [mf[index_in_list] for mf in mf_funcs_lists]
    return values.index(max(values))
```

Виведемо всі правила у таблиці імен функцій та списком:

```

rules = []
row = ['\033[91m' + "y\x " + '\033[0m']
for i in range(1, input_mf_number + 1):
    row.append('\033[93m' + "mx" + str(i).ljust(4))
print(''.join(row))
for i in range(input_mf_number):
    row = ['\033[92m' + "my" + str(i + 1) + '\033[0m' + ' ']
    for j in range(input_mf_number):
        z = np.sin(Y_means[i]) + np.cos(X_means[j]/2)
        best_func = max_mf_num(z, triangle_Z_mf, Z)
        row.append("mf" + str(best_func + 1))
        rules.append([j, i, best_func])
    print(''.join(["{:<6}".format(elem) for elem in row]))

print("\nВсі правила:")
for rule in rules:
    print(f"if (x is mx{rule[0] + 1}) and (y is my{rule[1] + 1}) then (z is mf{rule[2] + 1})")

```

| y\x | mx1 | mx2 | mx3 | mx4 | mx5 | mx6 |
|-----|------|------|------|------|------|------|
| my1 | mf1 | mf1 | mf1 | mf1 | mf1 | mf1 |
| my2 | mf3 | mf3 | mf3 | mf2 | mf2 | mf1 |
| my3 | mf7 | mf6 | mf6 | mf5 | mf5 | mf4 |
| my4 | mf10 | mf9 | mf9 | mf8 | mf8 | mf8 |
| my5 | mf12 | mf12 | mf11 | mf11 | mf11 | mf10 |
| my6 | mf12 | mf12 | mf12 | mf12 | mf12 | mf12 |

Всі правила:

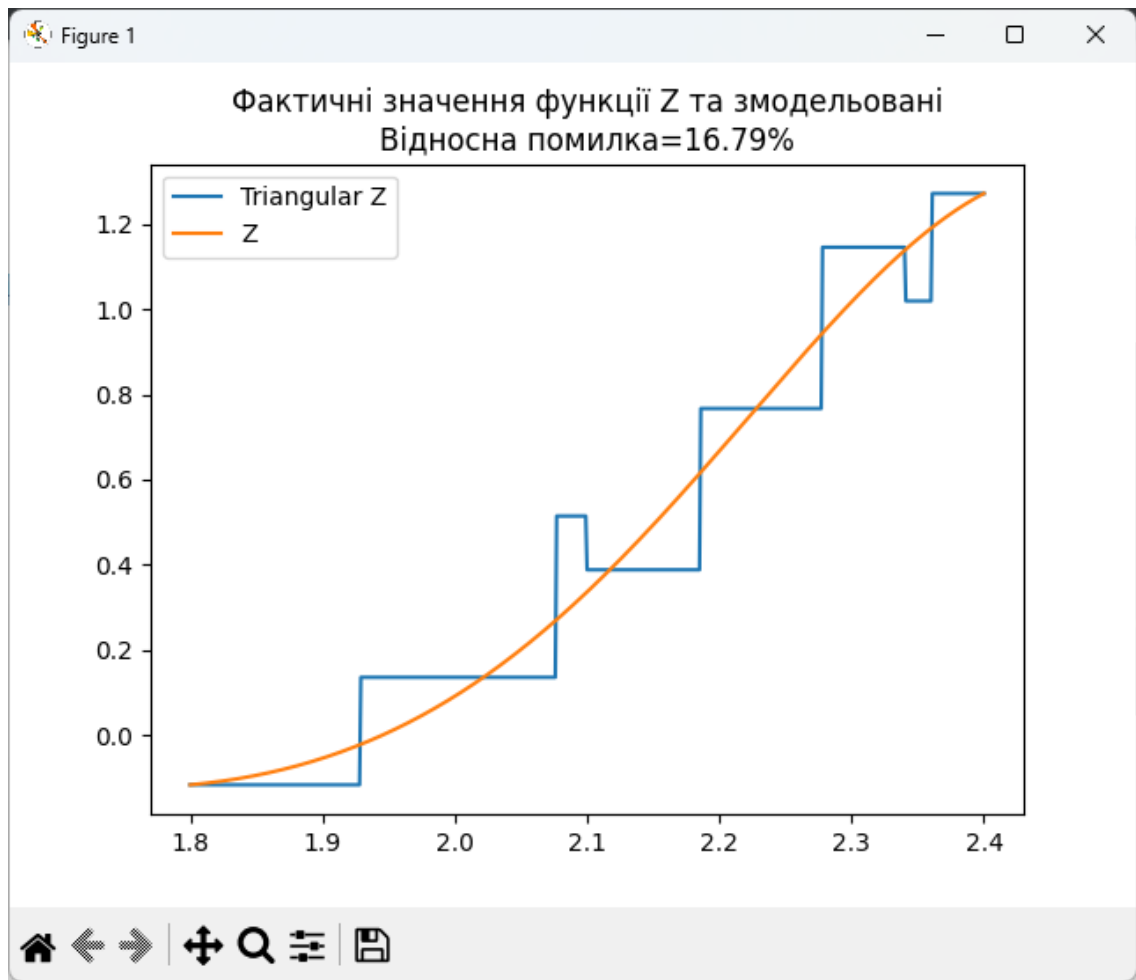
if (x is mx1) and (y is my1) then (z is mf1)
 if (x is mx2) and (y is my1) then (z is mf1)
 if (x is mx3) and (y is my1) then (z is mf1)
 if (x is mx4) and (y is my1) then (z is mf1)
 if (x is mx5) and (y is my1) then (z is mf1)
 if (x is mx6) and (y is my1) then (z is mf1)
 if (x is mx1) and (y is my2) then (z is mf3)
 if (x is mx2) and (y is my2) then (z is mf3)
 if (x is mx3) and (y is my2) then (z is mf3)
 if (x is mx4) and (y is my2) then (z is mf2)
 if (x is mx5) and (y is my2) then (z is mf2)
 if (x is mx6) and (y is my2) then (z is mf1)
 if (x is mx1) and (y is my3) then (z is mf7)
 if (x is mx2) and (y is my3) then (z is mf6)

if (x is mx3) and (y is my3) then (z is mf6)
 if (x is mx4) and (y is my3) then (z is mf5)
 if (x is mx5) and (y is my3) then (z is mf5)
 if (x is mx6) and (y is my3) then (z is mf4)
 if (x is mx1) and (y is my4) then (z is mf10)
 if (x is mx2) and (y is my4) then (z is mf9)
 if (x is mx3) and (y is my4) then (z is mf9)
 if (x is mx4) and (y is my4) then (z is mf8)
 if (x is mx5) and (y is my4) then (z is mf8)
 if (x is mx6) and (y is my4) then (z is mf8)
 if (x is mx1) and (y is my5) then (z is mf12)
 if (x is mx2) and (y is my5) then (z is mf12)
 if (x is mx3) and (y is my5) then (z is mf11)
 if (x is mx4) and (y is my5) then (z is mf11)
 if (x is mx5) and (y is my5) then (z is mf11)
 if (x is mx6) and (y is my5) then (z is mf10)
 if (x is mx1) and (y is my6) then (z is mf12)
 if (x is mx2) and (y is my6) then (z is mf12)
 if (x is mx3) and (y is my6) then (z is mf12)
 if (x is mx4) and (y is my6) then (z is mf12)
 if (x is mx5) and (y is my6) then (z is mf12)
 if (x is mx6) and (y is my6) then (z is mf12)

Візуалізуємо отриману модель функції z та власне функцію z, оцінимо відносну похибку моделювання:

```

Z_modelled = []
for i in range(len(X)):
    best_X = max_mf_num(X[i], triangle_X_mf, X)
    best_Y = max_mf_num(Y[i], triangle_Y_mf, Y)
    best_Z = None
    for rule in rules:
        if rule[0] == best_X and rule[1] == best_Y:
            best_Z = rule[2]
    Z_modelled.append(Z_means[best_Z])
relative_error = round(mean_absolute_error(Z, Z_modelled) * len(Z) / sum(abs(Z)) * 100, 2)
plot_func(X, [Z_modelled, Z], ["Triangular Z", "Z"], f"Фактичні значення функції Z та змодельовані\nВідносна помилка={relative_error}%")
  
```



Зробимо аналогічні дії для функції приналежності Гауса.

```

# параметр sigma для побудови функцій Гауса
X_sigma = 0.1
Y_sigma = 0.3
Z_sigma = 0.1
# функції приналежності Гауса
gaussian_X_mf = [fuzz.gaussmf(X, mean, X_sigma) for mean in X_means]
gaussian_Y_mf = [fuzz.gaussmf(Y, mean, Y_sigma) for mean in Y_means]
gaussian_Z_mf = [fuzz.gaussmf(Z, mean, Z_sigma) for mean in Z_means]
plot_func(X, gaussian_X_mf, [], 'Гаусівські функції приналежності для X')
plot_func(Y, gaussian_Y_mf, [], 'Гаусівські функції приналежності для Y')
plot_func(Z, gaussian_Z_mf, [], 'Гаусівські функції приналежності для Z')

rules = []
row = ['\033[91m' + "y\\x " + '\033[0m']
for i in range(1, input_mf_number + 1):
    row.append('\033[93m' + "mx" + str(i).ljust(4))
print(''.join(row))
for i in range(input_mf_number):
    row = ['\033[92m' + "my" + str(i + 1) + '\033[0m' + ' ']
    for j in range(input_mf_number):
        z = np.sin(Y_means[i]) + np.cos(X_means[j]/2)
        best_func = max_mf_num(z, gaussian_Z_mf, Z)
        row.append("mf" + str(best_func + 1))
        rules.append([j, i, best_func])
    print(''.join(["{:<6}".format(elem) for elem in row]))

print("\nВсі правила:")
for rule in rules:
    print(f"if (x is mx{rule[0] + 1}) and (y is my{rule[1] + 1}) then (z is mf{rule[2] + 1})")

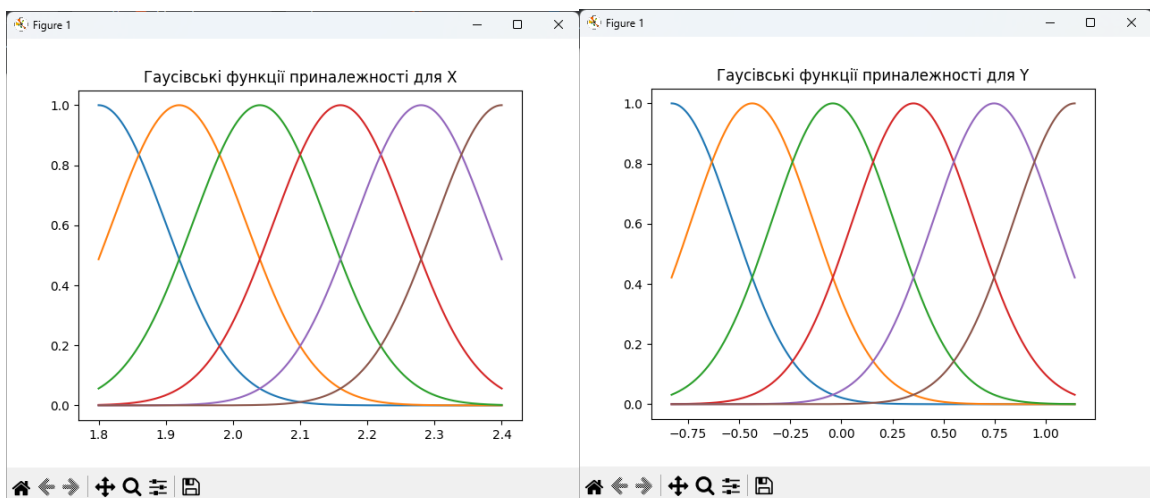
```

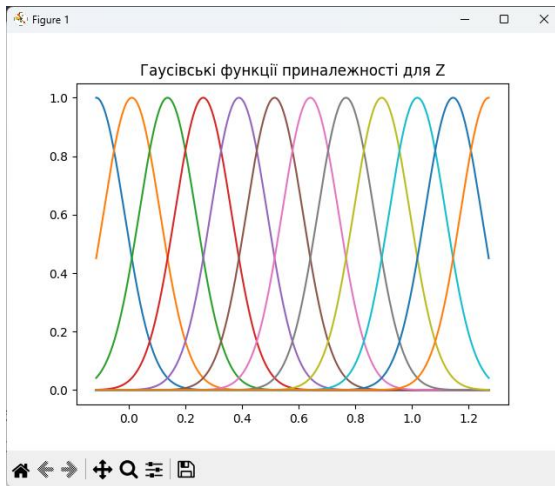
```

Z_modelled = []
for i in range(len(X)):
    best_X = max_mf_num(X[i], gaussian_X_mf, X)
    best_Y = max_mf_num(Y[i], gaussian_Y_mf, Y)
    best_Z = None
    for rule in rules:
        if rule[0] == best_X and rule[1] == best_Y:
            best_Z = rule[2]
    Z_modelled.append(Z_means[best_Z])

relative_error = round(mean_absolute_error(Z, Z_modelled) * len(Z) / sum(abs(Z)) * 100, 2)
plot_func(X, [Z_modelled, Z], ["Gaussian Z", "Z"], f"Фактичні значення функції Z та змодельовані\nВідносна помилка={relative_error}%")

```



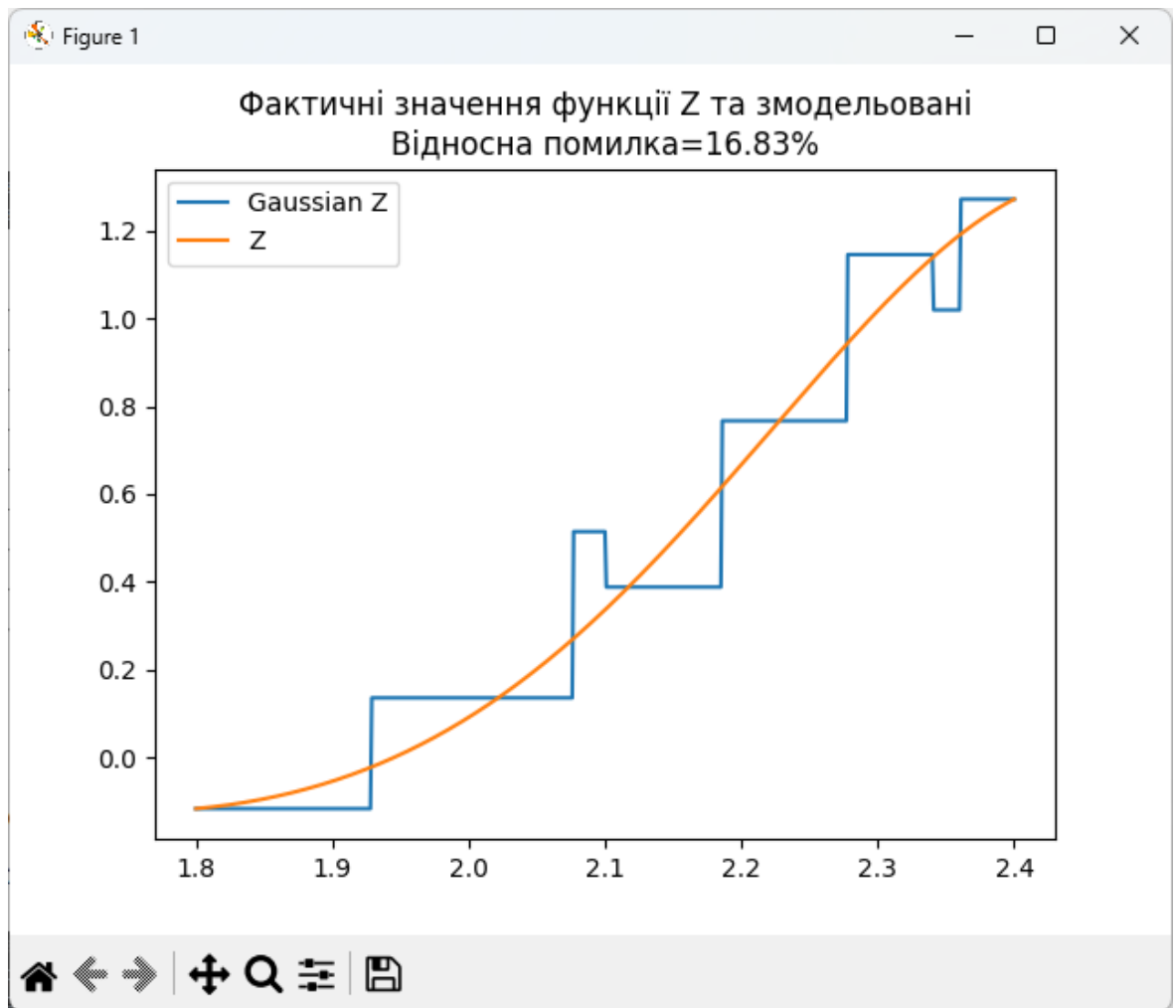


| y \ x | mx1 | mx2 | mx3 | mx4 | mx5 | mx6 |
|-------|------|------|------|------|------|------|
| my1 | mf1 | mf1 | mf1 | mf1 | mf1 | mf1 |
| my2 | mf3 | mf3 | mf3 | mf2 | mf2 | mf1 |
| my3 | mf7 | mf6 | mf6 | mf5 | mf5 | mf4 |
| my4 | mf10 | mf9 | mf9 | mf8 | mf8 | mf8 |
| my5 | mf12 | mf12 | mf11 | mf11 | mf11 | mf10 |
| my6 | mf12 | mf12 | mf12 | mf12 | mf12 | mf12 |

Всі правила:

- if (x is mx1) and (y is my1) then (z is mf1)
- if (x is mx2) and (y is my1) then (z is mf1)
- if (x is mx3) and (y is my1) then (z is mf1)
- if (x is mx4) and (y is my1) then (z is mf1)
- if (x is mx5) and (y is my1) then (z is mf1)
- if (x is mx6) and (y is my1) then (z is mf1)
- if (x is mx1) and (y is my2) then (z is mf3)
- if (x is mx2) and (y is my2) then (z is mf3)
- if (x is mx3) and (y is my2) then (z is mf3)
- if (x is mx4) and (y is my2) then (z is mf2)
- if (x is mx5) and (y is my2) then (z is mf2)
- if (x is mx6) and (y is my2) then (z is mf1)
- if (x is mx1) and (y is my3) then (z is mf7)
- if (x is mx2) and (y is my3) then (z is mf6)
- if (x is mx3) and (y is my3) then (z is mf6)

if (x is mx4) and (y is my3) then (z is mf5)
if (x is mx5) and (y is my3) then (z is mf5)
if (x is mx6) and (y is my3) then (z is mf4)
if (x is mx1) and (y is my4) then (z is mf10)
if (x is mx2) and (y is my4) then (z is mf9)
if (x is mx3) and (y is my4) then (z is mf9)
if (x is mx4) and (y is my4) then (z is mf8)
if (x is mx5) and (y is my4) then (z is mf8)
if (x is mx6) and (y is my4) then (z is mf8)
if (x is mx1) and (y is my5) then (z is mf12)
if (x is mx2) and (y is my5) then (z is mf12)
if (x is mx3) and (y is my5) then (z is mf11)
if (x is mx4) and (y is my5) then (z is mf11)
if (x is mx5) and (y is my5) then (z is mf11)
if (x is mx6) and (y is my5) then (z is mf10)
if (x is mx1) and (y is my6) then (z is mf12)
if (x is mx2) and (y is my6) then (z is mf12)
if (x is mx3) and (y is my6) then (z is mf12)
if (x is mx4) and (y is my6) then (z is mf12)
if (x is mx5) and (y is my6) then (z is mf12)
if (x is mx6) and (y is my6) then (z is mf12)



Висновок.

При виконанні лабораторної роботи було побудовано нечітку модель функції двох змінних згідно з варіантом, що містить 6 функцій приналежності для вхідних змінних і 12 вихідної. Для дослідження впливу форми функції приналежності на якість моделювання було обрано трикутну функцію та функцію Гауса. Відповідно до відносної помилки, трикутна функція приналежності незначно краща за Гауса, 16,79% проти 16,83%. Це пояснюється тим, що обидві функції приналежності мають однакові точки перетину і, як наслідок, утворюються однакові правила утворення результуючої функції z.