

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра інформатики та програмної інженерії

Звіт до лабораторної роботи №2

з курсу

«Методи ІІІ»

студента 2 курсу
групи ІТ-02
Макарова Іллі Сергійовича

Викладач:

Київ – 2022

ЛАБОРАТОРНА РОБОТА №2

«Моделювання функції з двох змінних засобами нечіткої математики»

Мета роботи: Проаналізувати засоби нечіткої логіки для моделювання функції з двох змінних. Провести дослідження форми функції приналежності на якість моделювання..

ЗАВДАННЯ

Варіант 10:

$$z = \sin(x) + \cos(y/2)$$

ВИКОНАННЯ

В нас 6 вхідних та 9 вихідних значень, тобто 36 правил, таким чином:

Ви написали, що треба більше коментувати роботу, тож тут я просто буду підставляти значення x та y у функцію мого варіанта, та отримані значення записувати в таблицю, нічого складного.

```
x_values = np.arange(0, 1.2, 0.2)
y_values = np.arange(0, 1.2, 0.2)

def my_function(a, b):
    return np.sin(a) + np.cos(b/2)

for x in x_values:
    for y in y_values:
        print(round(my_function(x, y), 5), end=', ')
    print()
```

1.0, 0.995, 0.98007, 0.95534, 0.92106, 0.87758,
1.19867, 1.19367, 1.17874, 1.15401, 1.11973, 1.07625,
1.38942, 1.38442, 1.36948, 1.34475, 1.31048, 1.267,
1.56464, 1.55965, 1.54471, 1.51998, 1.4857, 1.44223,
1.71736, 1.71236, 1.69742, 1.67269, 1.63842, 1.59494,
1.84147, 1.83648, 1.82154, 1.79681, 1.76253, 1.71905,

X / Y	0	0.2	0.4	0.6	0.8	1
0	1	0.995	0.98007	0.95534	0.92106	0.87758
0.2	1.19867	1.19367	1.17874	1.15401	1.11973	1.07625
0.4	1.38942	1.38442	1.36948	1.34475	1.31048	1.267
0.6	1.56464	1.55965	1.54471	1.51998	1.4857	1.44223
0.8	1.71736	1.71236	1.69742	1.67269	1.63842	1.59494
1	1.84147	1.83648	1.82154	1.79681	1.76253	1.71905

Відповідно рахуємо виходи.

X/Y	my1	my2	my3	my4	my5	my6
mx1	mf1	mf1	mf1	mf1	mf1	mf1
mx2	mf3	mf2	mf2	mf2	mf2	mf1
mx3	mf4	mf4	mf4	mf4	mf3	mf2
mx4	mf5	mf5	mf5	mf5	mf5	mf4

mx5	mf7	mf7	mf7	mf6	mf6	mf6
mx6	mf8	mf8	mf8	mf7	mf7	mf7

Ось тепер це все у коді:

```

r_1 = ctrl.Rule(
    antecedent=(
        mx["mx1"] & my["my1"] |
        mx["mx1"] & my["my2"] |
        mx["mx1"] & my["my3"] |
        mx["mx1"] & my["my4"] |
        mx["mx1"] & my["my5"] |
        mx["mx1"] & my["my6"] |
        mx["mx2"] & my["my6"]
    ),
    consequent=mf["mf1"],
    label='mf1'
)
r_2 = ctrl.Rule(
    antecedent=(
        mx["mx2"] & my["my2"] |
        mx["mx2"] & my["my3"] |
        mx["mx2"] & my["my4"] |
        mx["mx2"] & my["my5"] |
        mx["mx3"] & my["my6"]
    ),
    consequent=mf["mf2"],
    label='mf2'
)
r_3 = ctrl.Rule(
    antecedent=(
        mx["mx2"] & my["my1"] |
        mx["mx3"] & my["my5"]
    ),
    consequent=mf["mf3"],
    label="mf3"
)
r_4 = ctrl.Rule(
    antecedent=(
        mx["mx3"] & my["my1"] |
        mx["mx3"] & my["my2"] |
        mx["mx3"] & my["my3"] |
        mx["mx3"] & my["my4"] |
        mx["mx4"] & my["my6"]
    ),
    consequent=mf["mf4"],
    label="mf4"
)
r_5 = ctrl.Rule(
    antecedent=(

```

```

        mx["mx4"] & my["my1"] |
        mx["mx4"] & my["my2"] |
        mx["mx4"] & my["my3"] |
        mx["mx4"] & my["my4"] |
        mx["mx4"] & my["my5"]
    ),
    consequent=mf["mf5"],
    label="mf5"
)

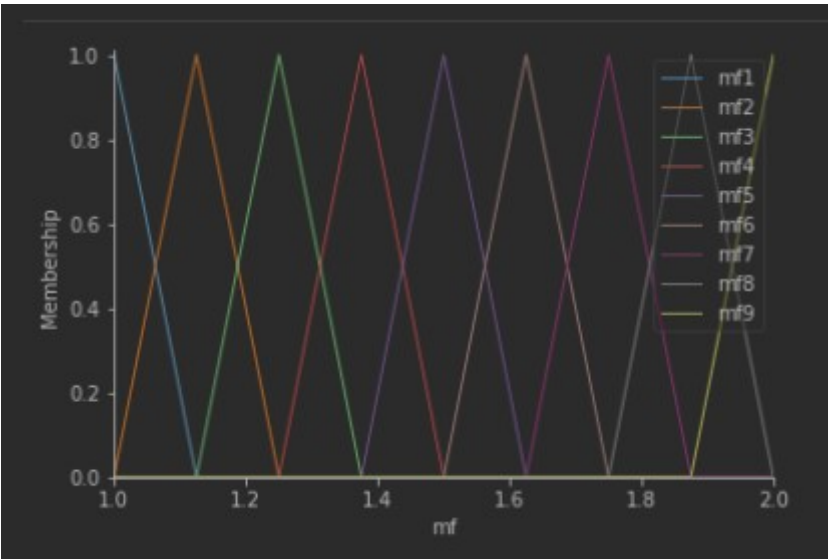
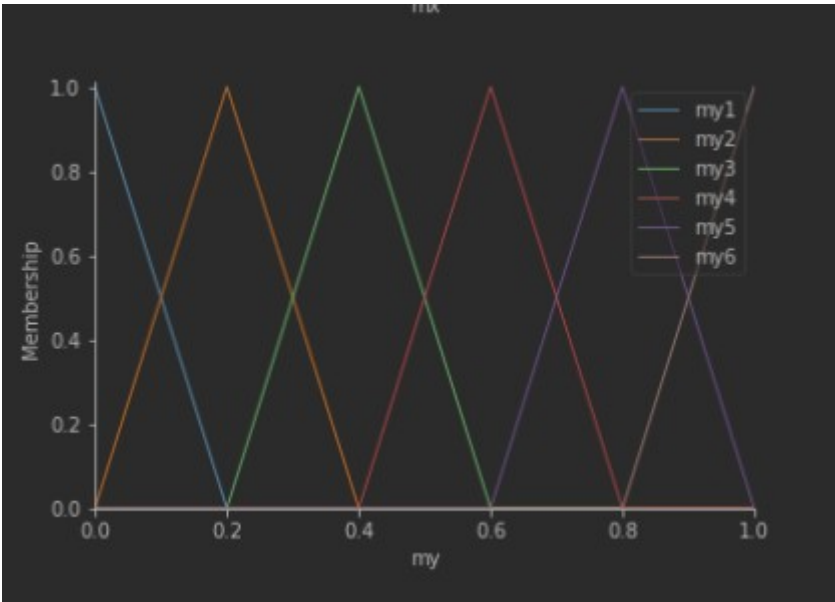
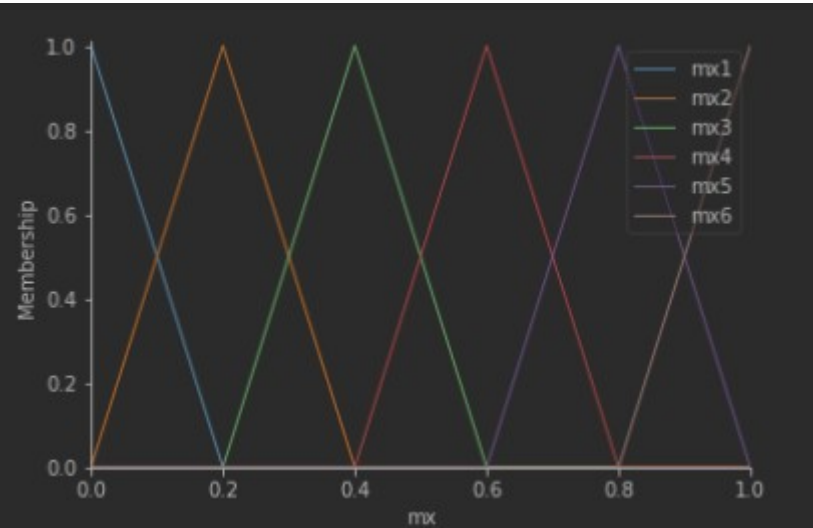
r_6 = ctrl.Rule(
    antecedent=(
        mx["mx5"] & my["my4"] |
        mx["mx5"] & my["my5"] |
        mx["mx5"] & my["my6"]
    ),
    consequent=mf["mf6"],
    label="mf6"
)

r_7 = ctrl.Rule(
    antecedent=(
        mx["mx5"] & my["my1"] |
        mx["mx5"] & my["my2"] |
        mx["mx5"] & my["my3"] |
        mx["mx6"] & my["my4"] |
        mx["mx6"] & my["my5"] |
        mx["mx6"] & my["my6"]
    ),
    consequent=mf["mf7"],
    label="mf7"
)

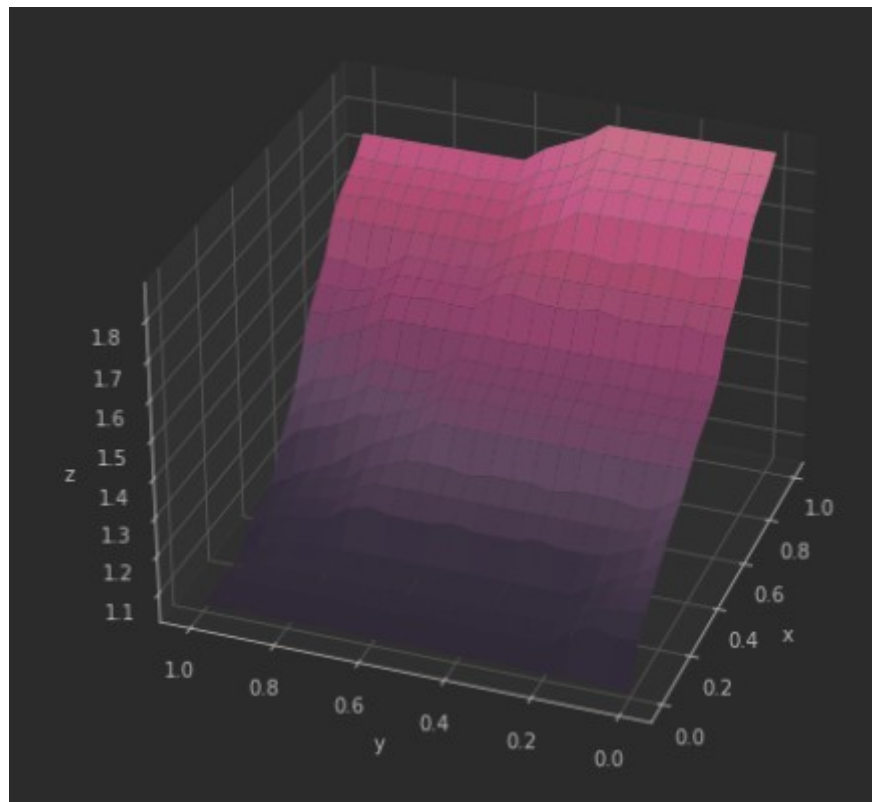
r_8 = ctrl.Rule(
    antecedent=(
        mx["mx6"] & my["my1"] |
        mx["mx6"] & my["my2"] |
        mx["mx6"] & my["my3"]
    ),
    consequent=mf["mf8"],
    label="mf8"
)

```

Моделювання Triangle:



Таким чином, наша модель:



Формула для вирахування похибки:

Похибка:

Для розрахунку похибки застосовують формулу:

$$\varepsilon = (|X_e - X_d| / X_e) * 100\% ,$$

де X_e – еталонне значення величини; X_d – дійсне значення величини.

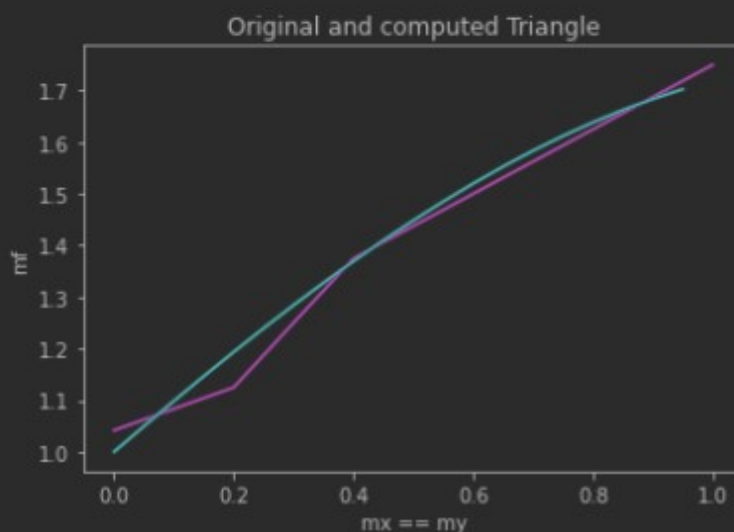
```

2 z = np.zeros_like(x)
3 x_for_graphic = np.arange(0,1, 0.05)
4
5 z_r = np.sin(x) + np.cos(x / 2)
6 z_g = np.sin(x_for_graphic) + np.cos(x_for_graphic / 2)
7
8 for i in range(len(x)):
9     simulation.input['mx'] = x[i]
10    simulation.input['my'] = x[i]
11    simulation.compute()
12
13    z[i] = simulation.output['mf']
14
15 e = abs(z_r - z) / z_r
16 print(f"ERROR: {round(np.mean(e) * 100, 3)} %")
17 plt.plot(x, z, color = 'm')
18 plt.plot(x_for_graphic, z_g, color = 'c')
19 plt.title("Original and computed Triangle")
20 plt.xlabel("mx == my")
21 plt.ylabel("mf")
22 plt.show()

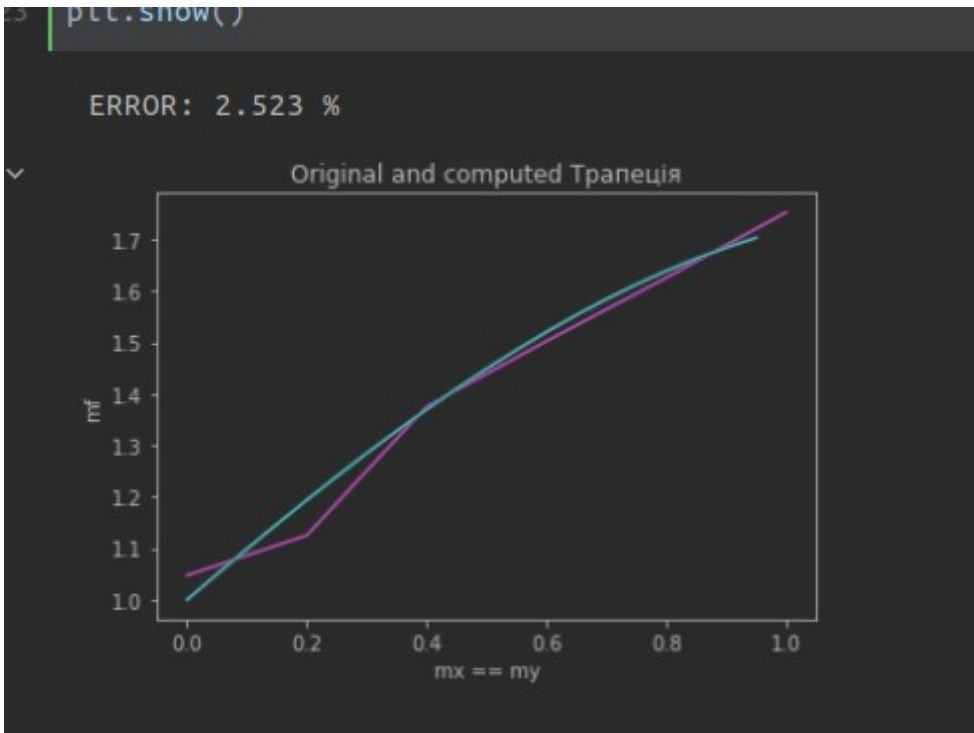
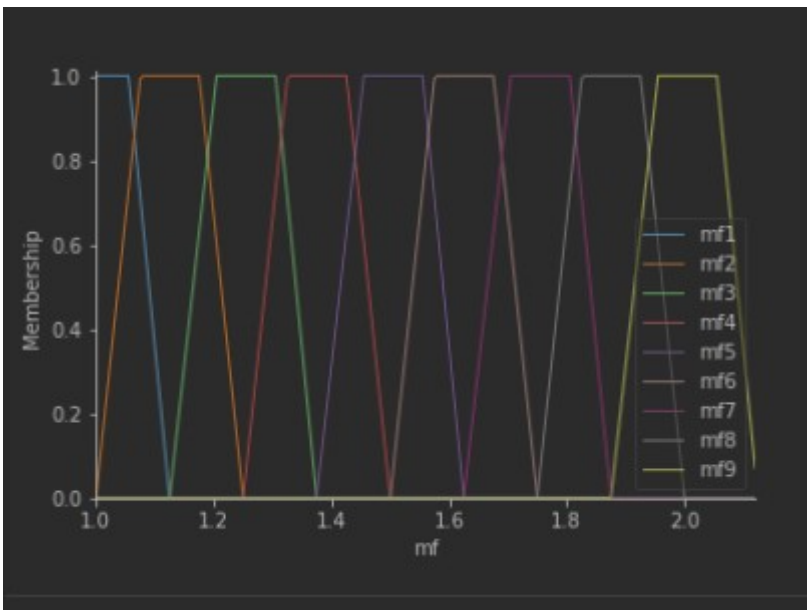
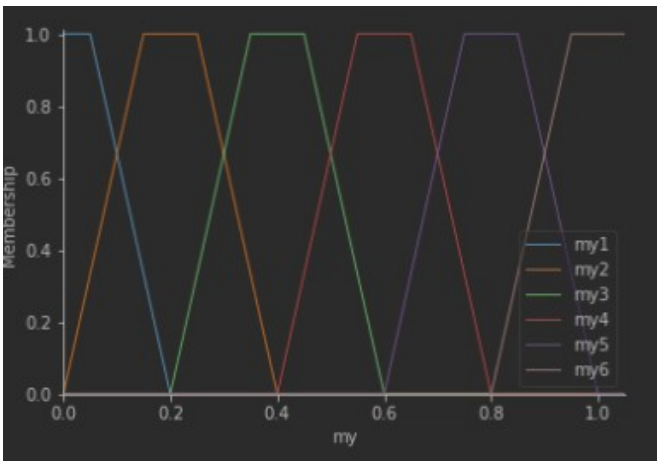
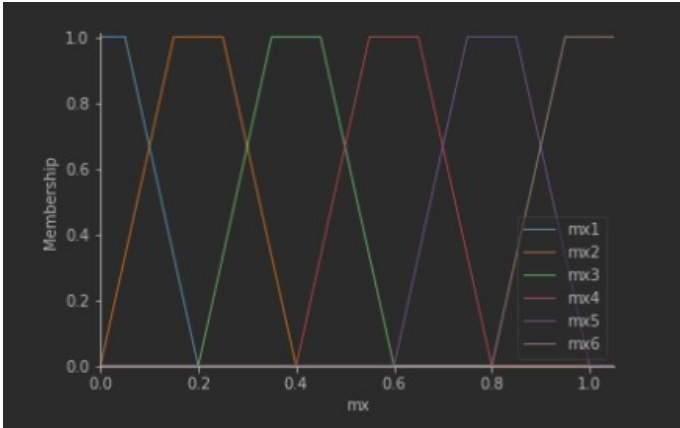
```

ERROR: 2.376%

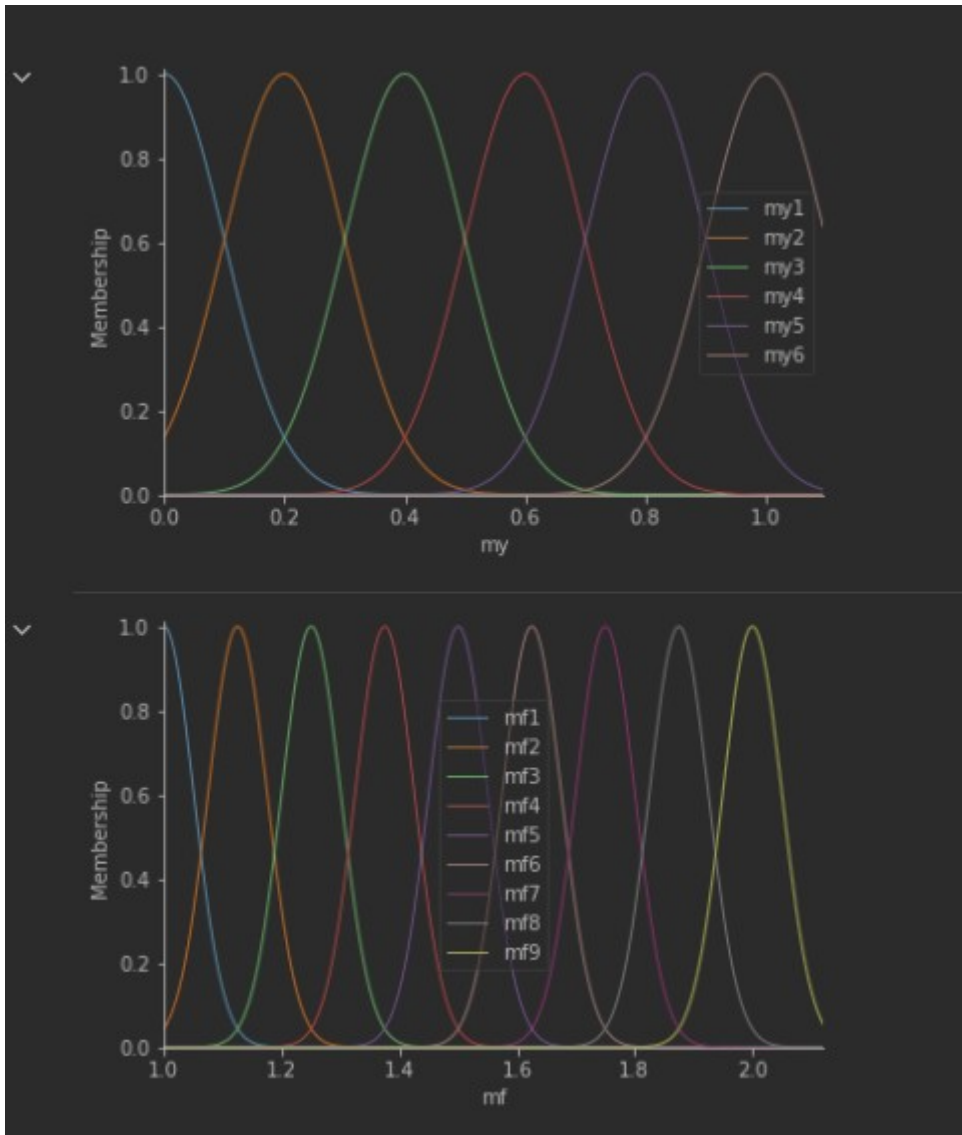
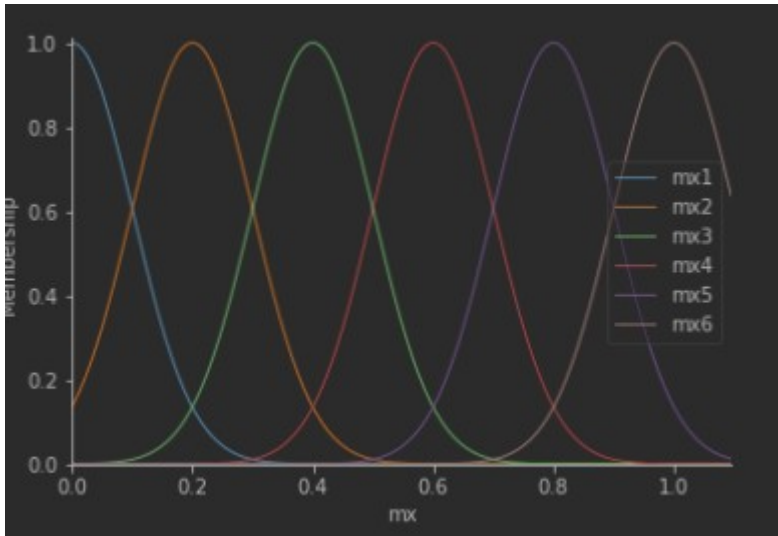
▼

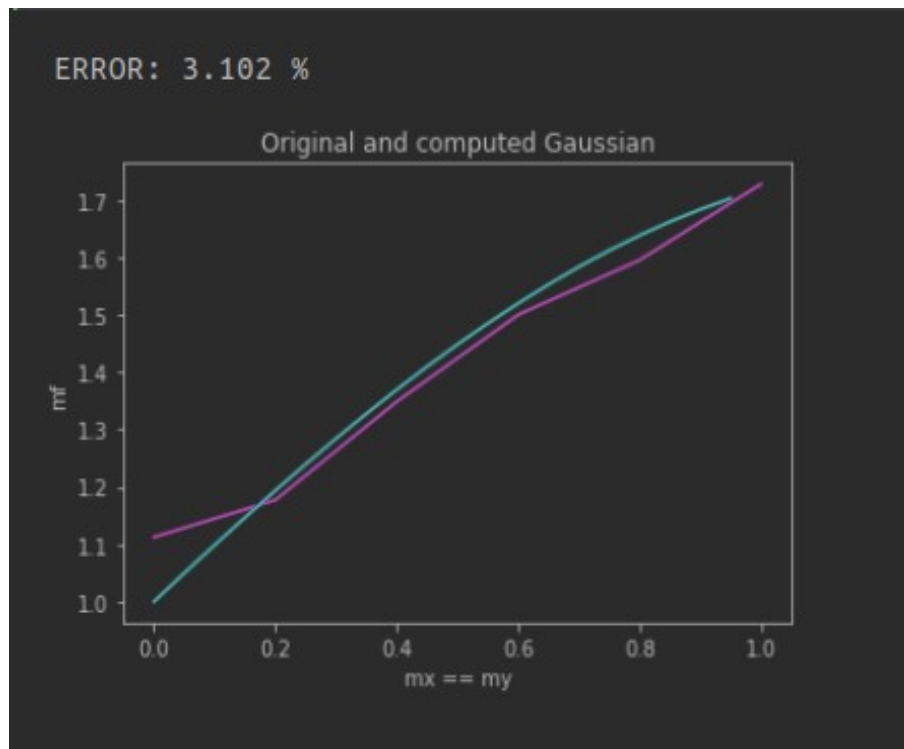


Моделювання за допомогою Трапеція



Моделювання за ФП Гауса





Спробуємо зменшити кількість правил, та подивимось як це повпливає на обчислення.

X/Y	my1	my2	my3	my4	my5	my6
mx1	mf1	mf1	mf1	mf1	mf1	mf1
mx2	mf3	mf2	mf2	mf2	mf2	mf1
mx3	mf4	mf4	mf4	mf4	mf3	mf2
mx4	mf5	mf5	mf5	mf5	mf5	mf4
mx5	mf7	mf7	mf7	mf6	mf6	mf6
mx6	mf8	mf8	mf8	mf7	mf7	mf7

Нові правила:

```
r_1 = ctrl.Rule(antecedent=(mx["mx1"] & my["my1"]), consequent=mf["mf1"],
label = 'rule mf1')
```

```
r_2 = ctrl.Rule(antecedent=(mx["mx2"] & my["my2"]), consequent=mf["mf2"],
label = 'rule mf2')
```

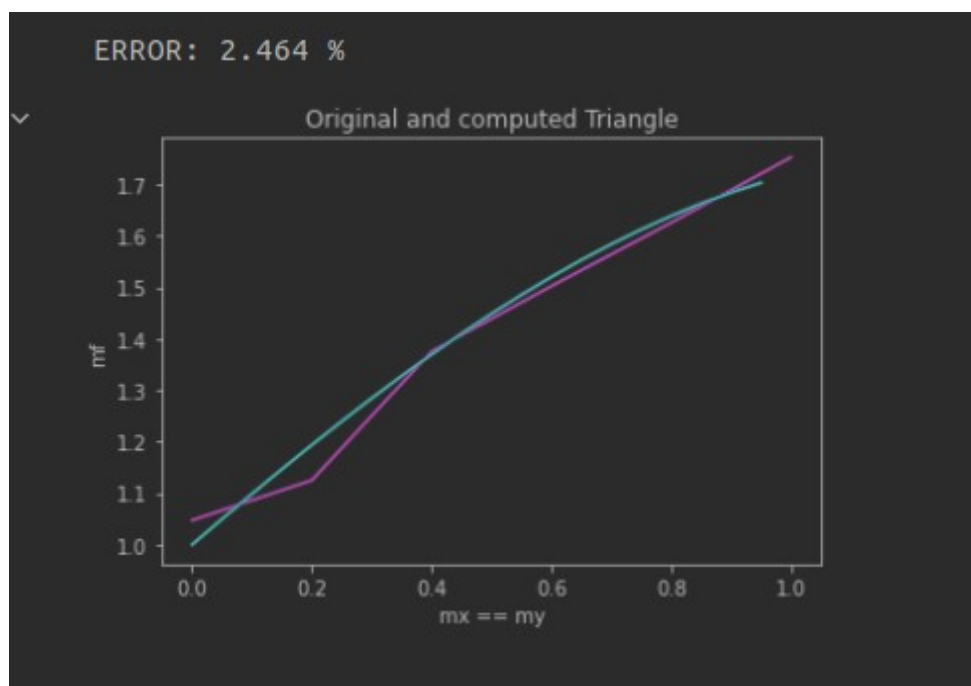
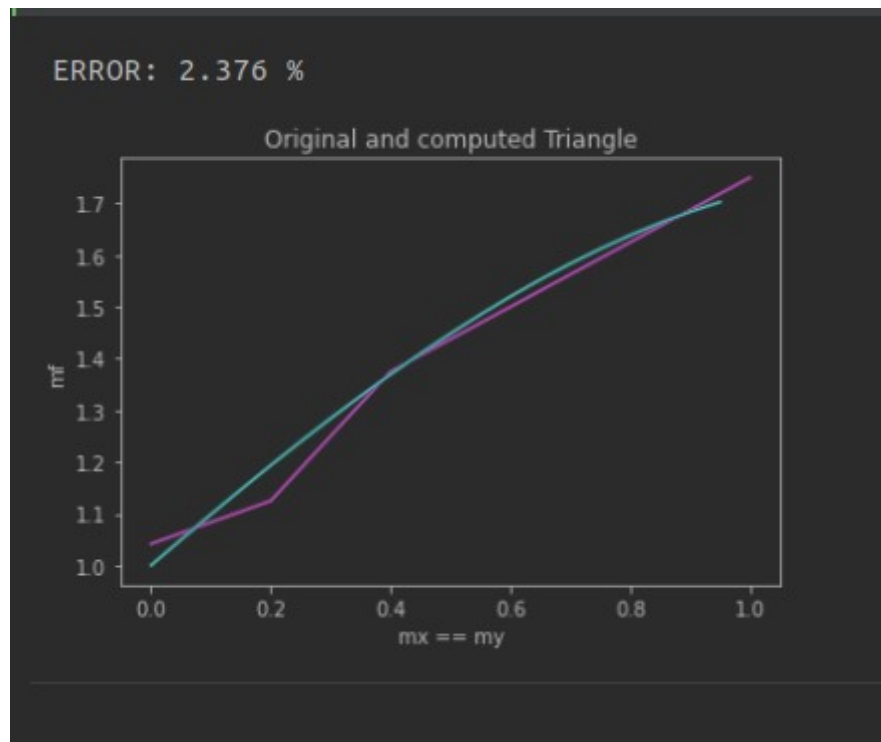
```
r_4 = ctrl.Rule(antecedent=(mx["mx3"] & my["my3"]), consequent =
mf["mf4"], label = "rule mf4")
```

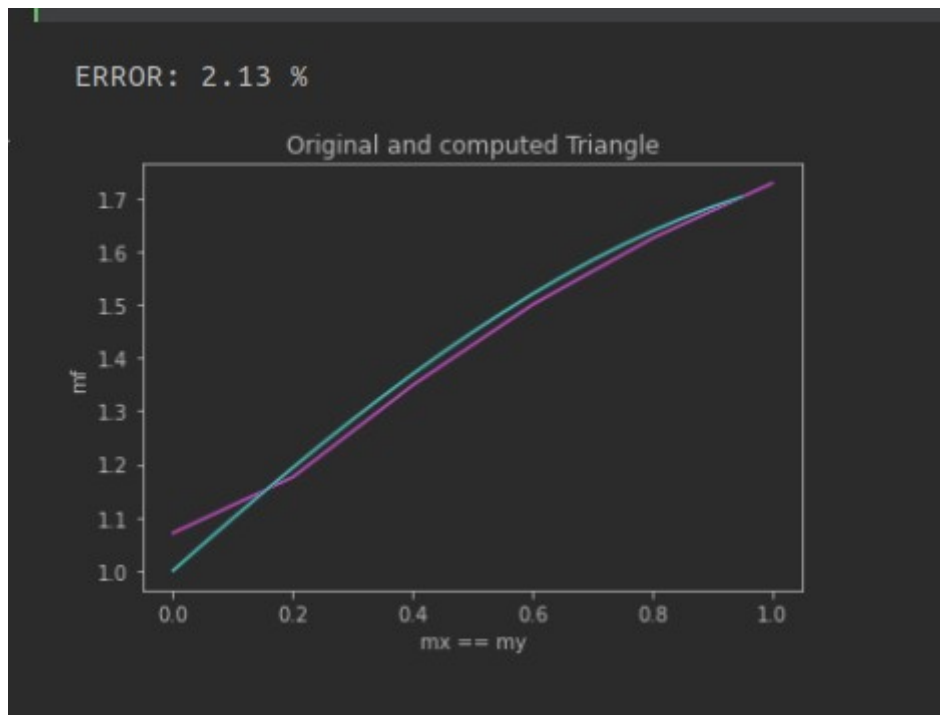
```
r_5 = ctrl.Rule(antecedent=(mx["mx4"] & my["my4"]), consequent =
mf["mf5"], label = "rule mf5")
```

```
r_6 = ctrl.Rule(antecedent=(mx["mx5"] & my["my5"]), consequent =  
mf["mf6"], label = "rule mf6")
```

```
r_7 = ctrl.Rule(antecedent=(mx["mx6"] & my["my6"]), consequent =  
mf["mf7"], label = "rule mf7")
```

НОБИ РОЗРАХУНКИ





Висновок:

Ну то щож, в цій лабі я навчився моделювати контроллери за допомогою елементів fuzzy logic. Я зробив об'єкт управління з двома входами і одним виходом. Провів дослідження як на якість моделі впливає ФП. Перевіряв, чи достатньо для моделі тільки діагональних правил. Таким чином можна зменшити кількість правил, шляхом видалення деяких з обчислень.