Practical 4

Name: Sharvari Kishor More
Class: D20B
Roll No: 35

Aim: To implement Fuzzy Membership Functions.

Theory:

Fuzzy membership functions define how each point in the input space (universe of discourse) is mapped to a membership value between 0 and 1. They are essential in fuzzy logic systems for representing uncertainty or imprecise information.
In this experiment, we apply these functions to classify the ripeness of fruits based on a **ripeness index** ranging from 0 (completely unripe) to 10 (overripe).

Different Types of Membership Functions:

1. **Singleton Membership Function:**
   - Assigns a membership value of 1 to a single point and 0 elsewhere.
   - Useful when a specific ripeness level has full membership in a fuzzy set.

   Equation:

$$\mu(x) = \begin{cases} 1, & \text{if } x = x_0 \\ 0, & \text{if } x \neq x_0 \end{cases}$$

2. **Triangular Membership Function:**
   - Defined by a triangular shape with three points: left endpoint, peak, right endpoint.
   - In ripeness, it models a gradual change from unripe to ripe.

   Equation:

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ or } x \geq c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases}$$

3. **Trapezoidal Membership Function:**
   - Similar to triangular but with a flat top representing full membership across a range.
   - Suitable for modeling stages like *semi-ripe* where multiple ripeness values are equally valid.

   Equation:

$$\mu(x) = \begin{cases} 0, & x \le a \text{ or } x \ge d \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \end{cases}$$

4. **Gaussian Membership Function:**
    - Bell-shaped curve that smoothly transitions between 0 and 1.
    - Ideal for modeling *ripe* stage centered at peak sweetness.

    Equation:

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

Example – Fruit Ripeness Classification:

- **Universe of Discourse:** Ripeness index (0 = unripe, 10 = overripe)
- **Fuzzy Sets:**
    - *Singleton:* Exact ripeness at index 5.
    - *Triangular:* Unripe (0–3).
    - *Trapezoidal:* Semi-ripe (2.5–6.5).
    - *Gaussian:* Ripe centered at index 7.5.

These functions allow smooth classification of fruit ripeness, enabling decisions like “ready to eat,” “needs more days,” or “overripe.”

I have attached the Colab file below.

Conclusion:

We have successfully implemented Fuzzy Membership Functions (Singleton, Triangular, Trapezoidal, Gaussian) for the *Fruit Ripeness Classification* problem. This helped us understand how fuzzy logic can model gradual transitions in ripeness stages, allowing more human-like decision-making in agricultural and food quality control systems.

Experiment 4

Name: Sharvari Kishor More

Class: D20B

Roll no.: 35

Aim: To implement Fuzzy Membership Functions.

Following are the steps for implementing Membership functions.

Importing libraires

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

Membership function definitions

```
def singleton_mf(x, x0):
    return np.where(x == x0, 1, 0)

def triangular_mf(x, a, b, c):
    return np.maximum(np.minimum((x-a)/(b-a), (c-x)/(c-b)), 0)

def trapezoidal_mf(x, a, b, c, d):
    return np.maximum(np.minimum(np.minimum((x-a)/(b-a), 1), (d-x)/(d-c)), 0)

def gaussian_mf(x, c, sigma):
    return np.exp(-((x-c)**2) / (2*sigma**2))
```

Universe of Discourse

```
x = np.linspace(-10, 10, 500)
```

Example fuzzy sets

```
set_A = singleton_mf(x, 2)            # Singleton
set_B = triangular_mf(x, -5, 0, 5)    # Triangular
set_C = trapezoidal_mf(x, -7, -4, 4, 7)  # Trapezoidal
set_D = gaussian_mf(x, 0, 2)          # Gaussian
```

Fuzzification

```
input_value = 1.0  # Example crisp input

deg_A = singleton_mf(np.array([input_value]), 2)[0]
deg_B = triangular_mf(np.array([input_value]), -5, 0, 5)[0]
deg_C = trapezoidal_mf(np.array([input_value]), -7, -4, 4, 7)[0]
deg_D = gaussian_mf(np.array([input_value]), 0, 2)[0]

print("Fuzzification results:")
print(f"Singleton Set A: {deg_A:.3f}")
print(f"Triangular Set B: {deg_B:.3f}")
print(f"Trapezoidal Set C: {deg_C:.3f}")
print(f"Gaussian Set D: {deg_D:.3f}")
```

```
Fuzzification results:
Singleton Set A: 0.000
Triangular Set B: 0.800
Trapezoidal Set C: 1.000
Gaussian Set D: 0.882
```

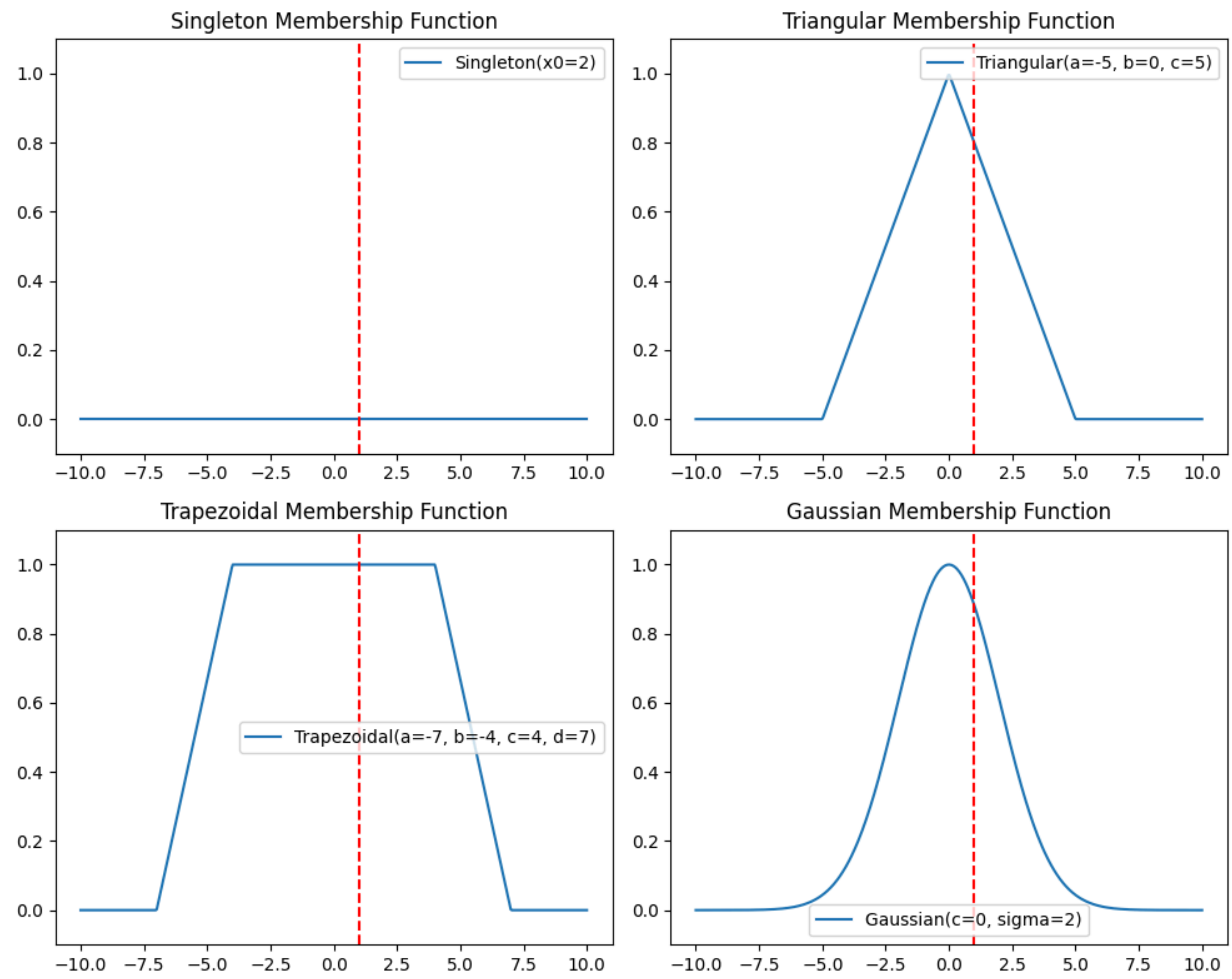
Defuzzification

```
val_A, val_B, val_C, val_D = 2, 0, 0, 0  # Example: using set centers
numerator = (deg_A * val_A +
             deg_B * 0 +   # center of triangle
             deg_C * 0 +   # midpoint of trapezoid
             deg_D * 0)    # center of Gaussian
denominator = deg_A + deg_B + deg_C + deg_D

if denominator != 0:
    crisp_output = numerator / denominator
else:
    crisp_output = 0

print(f"Defuzzified Output: {crisp_output:.2f}")
```

```
Defuzzified Output: 0.00
```

Plotting membership functions

```
plt.figure(figsize=(10, 8))

plt.subplot(2, 2, 1)
plt.plot(x, set_A, label="Singleton(x0=2)")
plt.axvline(input_value, color='red', linestyle='--')
plt.title('Singleton Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()

plt.subplot(2, 2, 2)
plt.plot(x, set_B, label="Triangular(a=-5, b=0, c=5)")
plt.axvline(input_value, color='red', linestyle='--')
plt.title('Triangular Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()

plt.subplot(2, 2, 3)
plt.plot(x, set_C, label="Trapezoidal(a=-7, b=-4, c=4, d=7)")
plt.axvline(input_value, color='red', linestyle='--')
plt.title('Trapezoidal Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()

plt.subplot(2, 2, 4)
plt.plot(x, set_D, label="Gaussian(c=0, sigma=2)")
plt.axvline(input_value, color='red', linestyle='--')
plt.title('Gaussian Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()

plt.tight_layout()
plt.show()
```



Example - Fruit Ripeness with fuzzification and defuzzification.

Following are the steps.

Importing Libraries

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

Membership Function Definitions

```
def singleton_mf(x, x0):
    return np.where(x == x0, 1, 0)

def triangular_mf(x, a, b, c):
    return np.maximum(np.minimum((x-a)/(b-a), (c-x)/(c-b)), 0)

def trapezoidal_mf(x, a, b, c, d):
    return np.maximum(np.minimum(np.minimum((x-a)/(b-a), 1), (d-x)/(d-c)), 0)

def gaussian_mf(x, c, sigma):
    return np.exp(-((x-c)**2) / (2*sigma**2))
```

Universe of Discourse: Ripeness index (0 to 10)

```
x = np.linspace(0, 10, 500)
```

Membership functions for ripeness stages

```
unripe = triangular_mf(x, 0, 1.5, 3)
semi_ripe = trapezoidal_mf(x, 2.5, 4, 5, 6.5)
ripe = gaussian_mf(x, 7.5, 1)
```

Fuzzification

```
input_value = 5.8  # Example fruit ripeness index

unripe_degree = triangular_mf(np.array([input_value]), 0, 1.5, 3)[0]
semi_ripe_degree = trapezoidal_mf(np.array([input_value]), 2.5, 4, 5, 6.5)[0]
ripe_degree = gaussian_mf(np.array([input_value]), 7.5, 1)[0]

print("Fuzzification results:")
print(f"Unripe: {unripe_degree:.3f}")
print(f"Semi-ripe: {semi_ripe_degree:.3f}")
print(f"Ripe: {ripe_degree:.3f}")
```

```
Fuzzification results:
Unripe: 0.000
Semi-ripe: 0.467
Ripe: 0.236
```

Defuzzification

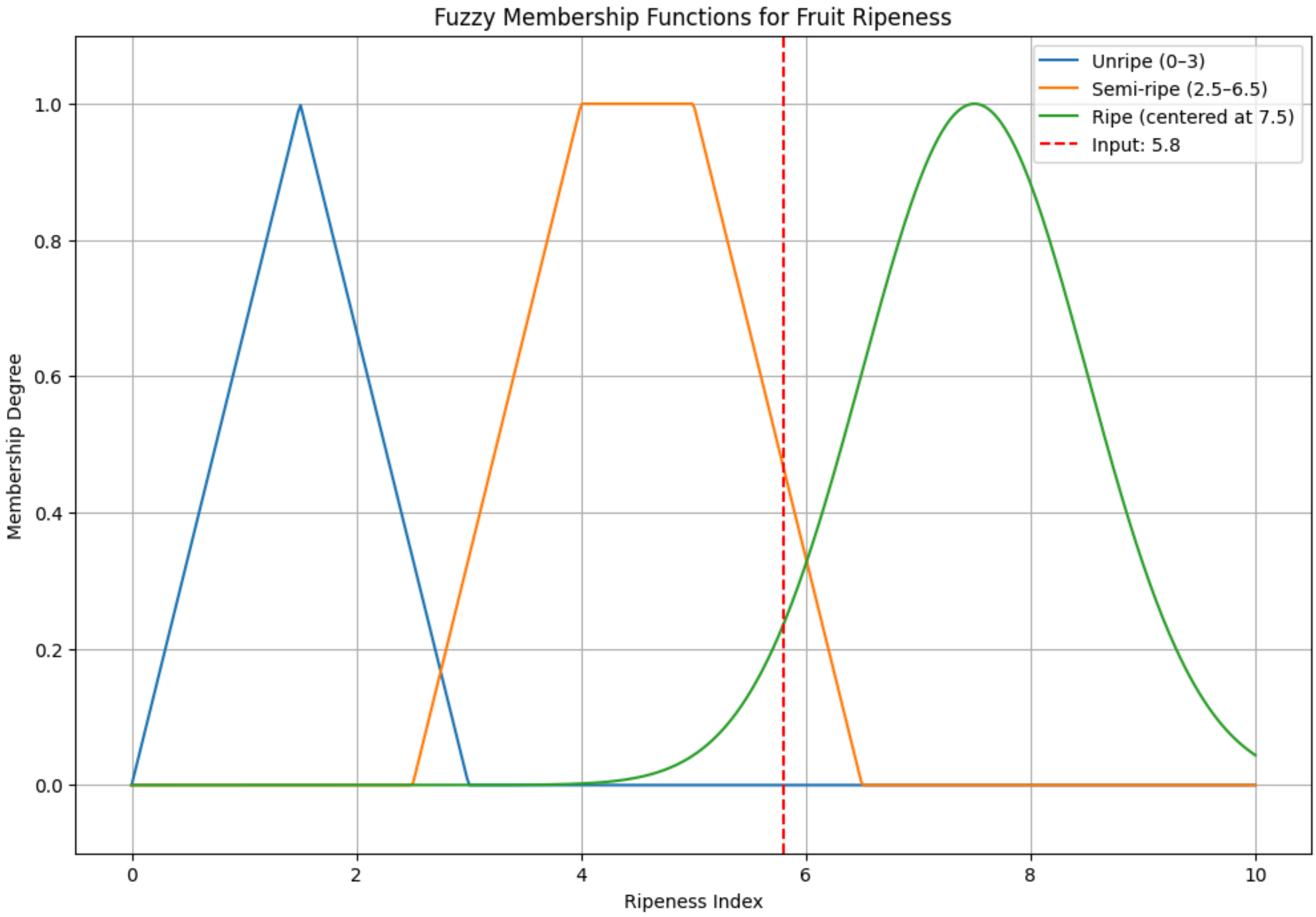
Assign representative crisp values for each category.

```
unripe_val = 1.5
semi_ripe_val = 4.5
ripe_val = 7.5

numerator = (unripe_degree * unripe_val +
             semi_ripe_degree * semi_ripe_val +
             ripe_degree * ripe_val)
denominator = (unripe_degree + semi_ripe_degree + ripe_degree)

if denominator != 0:
    crisp_output = numerator / denominator
else:
    crisp_output = 0

print(f"Defuzzified ripeness score: {crisp_output:.2f}")
```

```
Defuzzified ripeness score: 5.51
```

Plotting membership functions

```
plt.figure(figsize=(12, 8))
plt.plot(x, unripe, label="Unripe (0–3)")
plt.plot(x, semi_ripe, label="Semi-ripe (2.5–6.5)")
plt.plot(x, ripe, label="Ripe (centered at 7.5)")

# Mark the input value
plt.axvline(input_value, color='red', linestyle='--', label=f"Input: {input_value}")
plt.title('Fuzzy Membership Functions for Fruit Ripeness')
plt.xlabel('Ripeness Index')
plt.ylabel('Membership Degree')
plt.ylim(-0.1, 1.1)
plt.legend()
plt.grid(True)
plt.show()
```



Conclusion: We have successfully implemented Fuzzy Membership Functions (Singleton, Triangular, Trapezoidal, Gaussian) for the Fruit Ripeness Classification problem. This helped us understand how fuzzy logic can model gradual transitions in ripeness stages, allowing more human-like decision-making in agricultural and food quality control systems.

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit