Experiment No:- 04

Aim:- To implement Fuzzy Membership Functions.

Theory:- Explain different member functions with plotting & Implement Singleton, triangular, Trapezoidal and gaussian membership function.

→ Fuzzy membership functions define how each point in the input space (often referred to as the universe of discourse) is mapped to a membership value between 0 and 1. These functions are crucial in fuzzy logic systems for handling uncertain or imprecise information.

Different Types of Membership Functions:

1. **Singleton Membership Function**:
    - The simplest membership function.
    - It assigns a membership value of 1 to a single point and 0 to all other points.
    - Useful when a specific value has full membership in a fuzzy set.

    Equation μ(x) =
    {
    1 if x = x ,
    0 if x ≠ x_0
    }

2. **Triangular Membership Function**:
    - Defined by a triangular shape with three points: the left endpoint, the peak, and the right endpoint.
    - It's a simple and widely used function due to its ease of calculation.

    Equation: μ(x) =
    {
    0 if x ≤ a or x ≥ c,
    (x - a) / (b - a) if a ≤ x ≤ b,
    (c - x) / (c - b) if b ≤ x ≤ c
    }

3. **Trapezoidal Membership Function**:
    - Similar to the triangular function but with a flat top, indicating that all values in the top range have full membership.

    Equation: μ(x) =
    {
    0 if x ≤ a or x ≥ c,
    (x - a) / (b - a) if a ≤ x ≤ b,
    (c - x) / (c - b) if b ≤ x ≤ c

}

where a, b, c, and d are the left endpoint, start of the top, end of the top, and right endpoint, respectively

4. **Gaussian Membership Function**:

   Characterized by a bell-shaped curve.
   It smoothly transitions from 0 to 1 and back to 0, which is useful in systems where smooth transitions are needed.

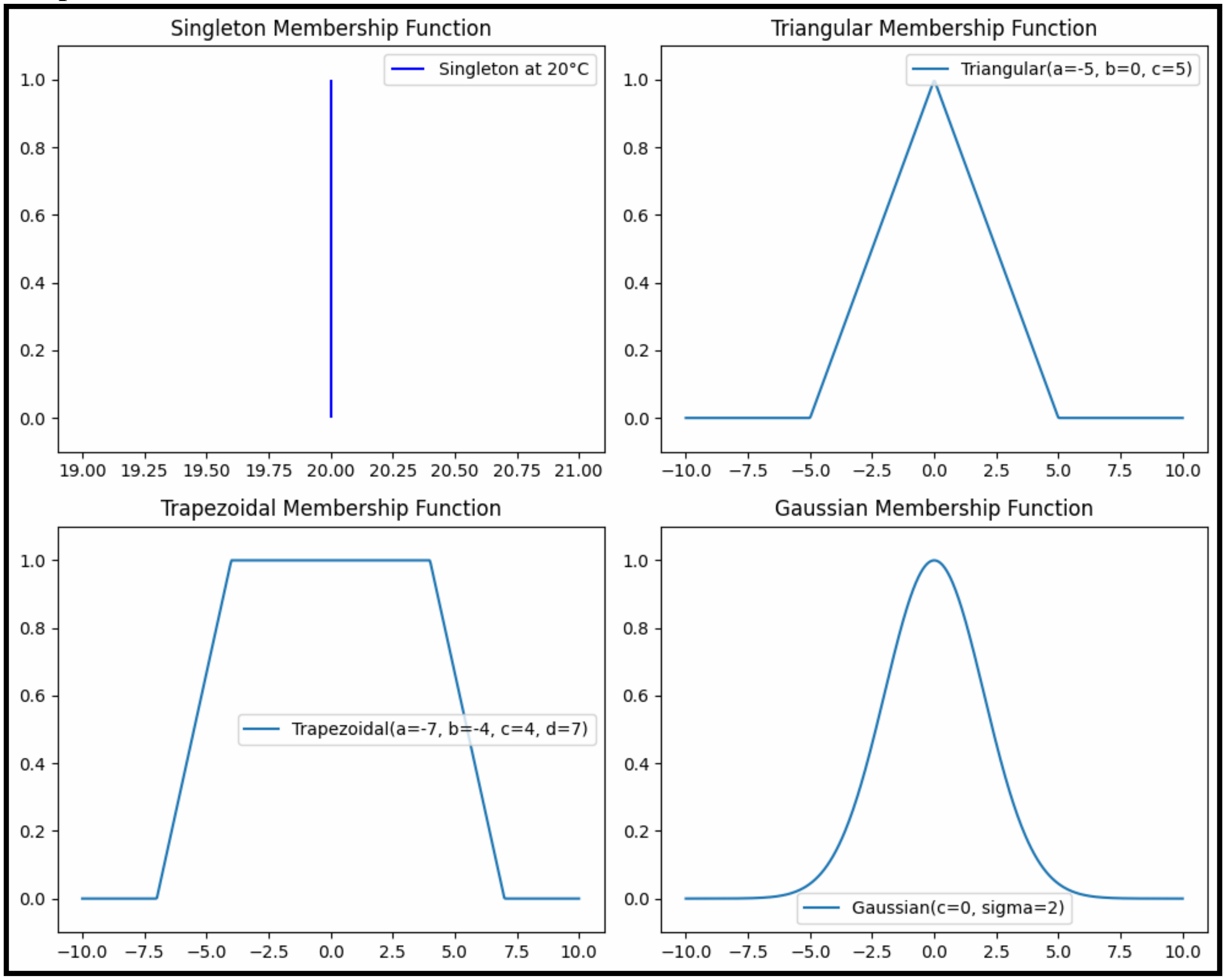
   Equation: $\mu(x) = e^{(-(x - c)^2 / (2\sigma^2))}$

Code:-

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

# Singleton Membership Function
def singleton_mf(x, x0):
   return np.where(x == x0, 1, 0)

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

# Trapezoidal Membership Function
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)

# Gaussian Membership Function
def gaussian_mf(x, c, sigma):
   return np.exp(-((x-c)**2) / (2*sigma**2))

# Plotting the membership functions
x = np.linspace(-10, 10, 500)

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

# Singleton
plt.subplot(2, 2, 1)
plt.plot(x, singleton_mf(x, 2), label="Singleton(x0=2)")
plt.title('Singleton Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()

# Triangular
plt.subplot(2, 2, 2)
plt.plot(x, triangular_mf(x, -5, 0, 5), label="Triangular(a=-5, b=0, c=5)")
plt.title('Triangular Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()

# Trapezoidal
plt.subplot(2, 2, 3)
plt.plot(x, trapezoidal_mf(x, -7, -4, 4, 7), label="Trapezoidal(a=-7, b=-4, c=4, d=7)")
plt.title('Trapezoidal Membership Function')
```

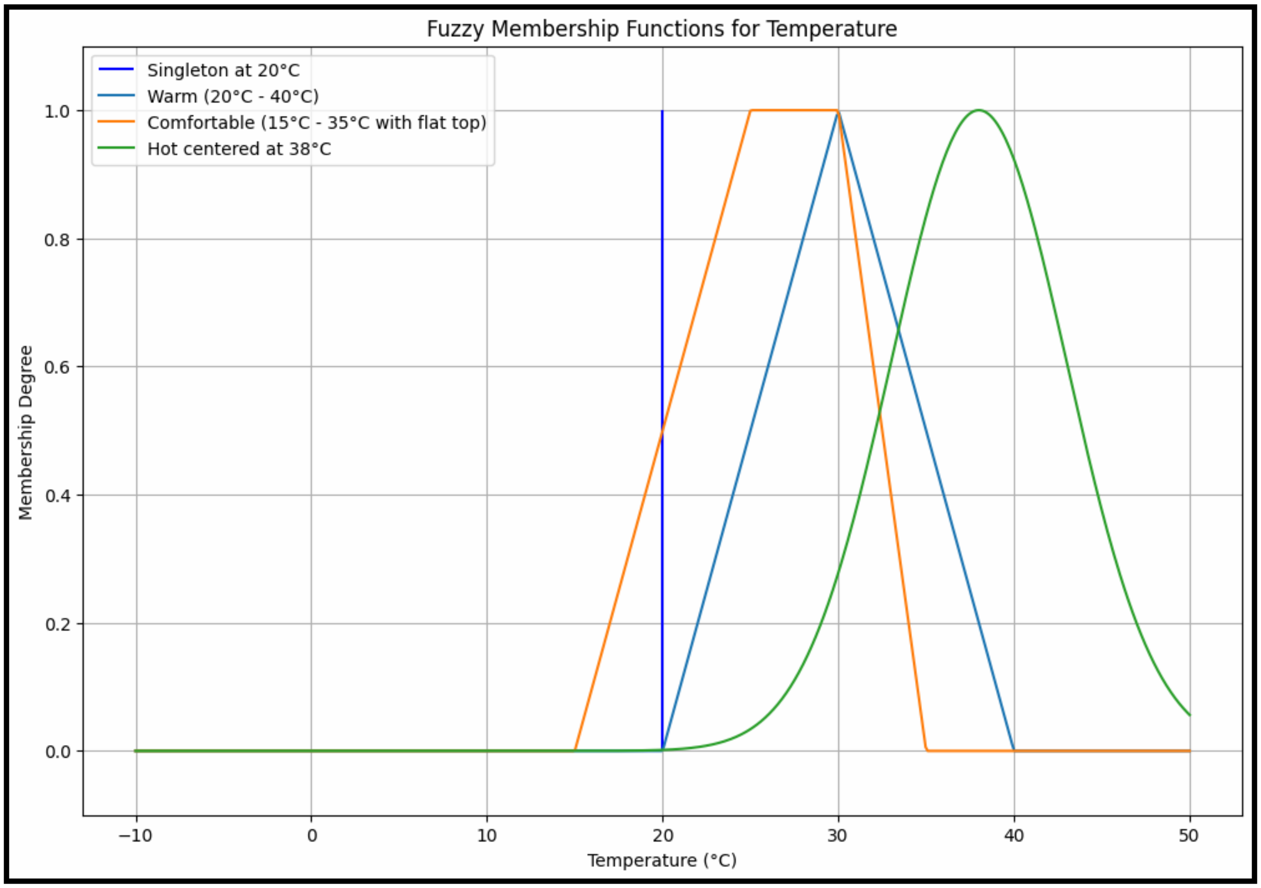
```
plt.ylim(-0.1, 1.1)
plt.legend()

# Gaussian
plt.subplot(2, 2, 4)
plt.plot(x, gaussian_mf(x, 0, 2), label="Gaussian(c=0, sigma=2)")
plt.title('Gaussian Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()
plt.tight_layout()
plt.show()
```

Output:-



Example - Temperature

```
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))

# Example: Temperature Classification
x = np.linspace(-10, 50, 500)  # Temperature range from -10°C to 50°C

# Membership Functions for different fuzzy sets
singleton_temp = singleton_mf(x, 20)  # Singleton at 20°C
warm_temp = triangular_mf(x, 20, 30, 40)  # Warm (20°C - 40°C)
comfortable_temp = trapezoidal_mf(x, 15, 25, 30, 35)  # Comfortable (15°C - 35°C)
hot_temp = gaussian_mf(x, 38, 5)  # Hot centered at 38°C

# Plot the membership functions
plt.figure(figsize=(12, 8))

# Use vlines for Singleton to show it as a vertical line
plt.vlines(20, 0, 1, colors='blue', label="Singleton at 20°C", linestyles='solid')

plt.plot(x, warm_temp, label="Warm (20°C - 40°C)")
plt.plot(x, comfortable_temp, label="Comfortable (15°C - 35°C with flat top)")
plt.plot(x, hot_temp, label="Hot centered at 38°C")

plt.title('Fuzzy Membership Functions for Temperature')
plt.xlabel('Temperature (°C)')
plt.ylabel('Membership Degree')
plt.ylim(-0.1, 1.1)
plt.legend()
plt.grid(True)
plt.show()
```



Conclusion:- Hence, We've implemented and understood Fuzzy Membership Functions, using Singleton, Triangular, Trapezoidal, and Gaussian types to model and process uncertainty. This has enhanced our ability to represent imprecise data and make flexible, human-like decisions in fuzzy logic systems.

AIDS_2-EXP-4.ipynb