Aim: To implement Fuzzy Membership Functions.

Theory:

Fuzzy logic is a form of reasoning that deals with imprecise or uncertain data.

Instead of saying something is completely true (1) or false (0) like in classical logic, fuzzy logic allows for partial truth values between 0 and 1.

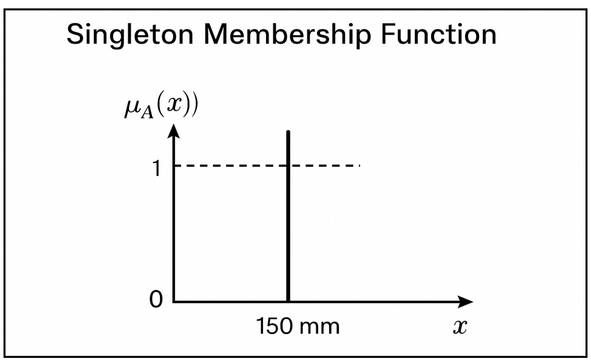
A fuzzy membership function defines how each element in the input space (called the universe of discourse) is assigned a degree of membership between 0 and 1 for a given fuzzy set.

These functions are essential in fuzzy systems because they help in modeling vague or subjective concepts like “high temperature,” “moderate rainfall,” or “low speed.”

Types of Fuzzy Membership Functions

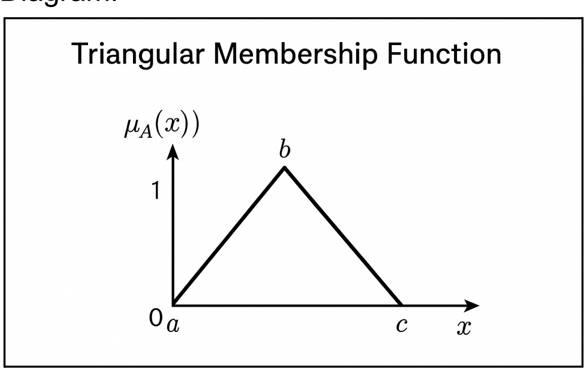
1. Singleton Membership Function
    - Assigns a membership value of 1 to one specific value and 0 to all others.
    - Used when only one exact point has full membership.
    - Example: Rainfall exactly at 150 mm is considered "ideal" for crops.

    Diagram:



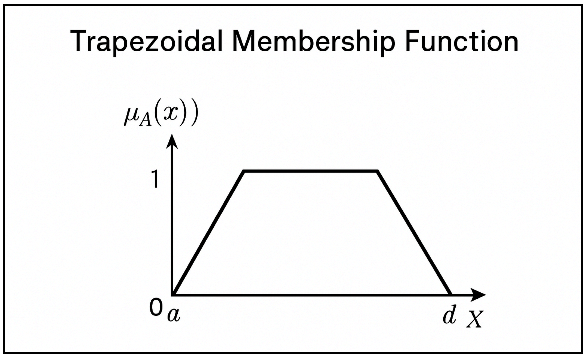
2. Triangular Membership Function
    - Defined by three points: the start (a), the peak (b), and the end (c).
    - The membership increases linearly from a to b, then decreases linearly from b to c.
    - Example: Light rainfall between 0 mm and 100 mm.

    Diagram:



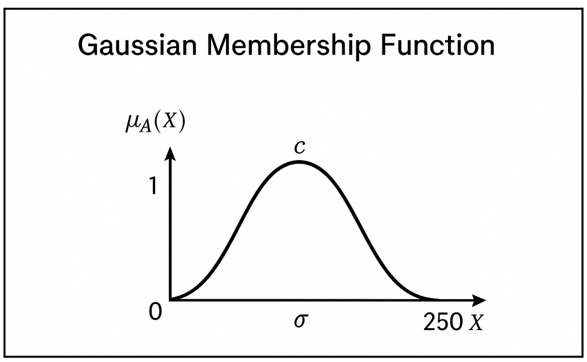
3. Trapezoidal Membership Function
    - Similar to triangular but with a flat top where membership remains 1 between two points (b and c).
    - Defined by four points: start (a), start of top (b), end of top (c), and end (d).
    - Example: Moderate rainfall between 80 mm and 200 mm.

    Diagram:



4. Gaussian Membership Function
    - Has a bell-shaped curve, defined by a center (c) and width (σ).
    - Smooth and continuous, suitable for gradual transitions.
    - Example: Heavy rainfall centered around 250 mm.

    Diagram:



Importance in Fuzzy Logic Systems

- Models human-like reasoning where boundaries are not strict.
- Allows systems to handle uncertainty, vagueness, and approximate data.
- Commonly used in control systems, decision-making, pattern recognition, and AI.

```
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)
```

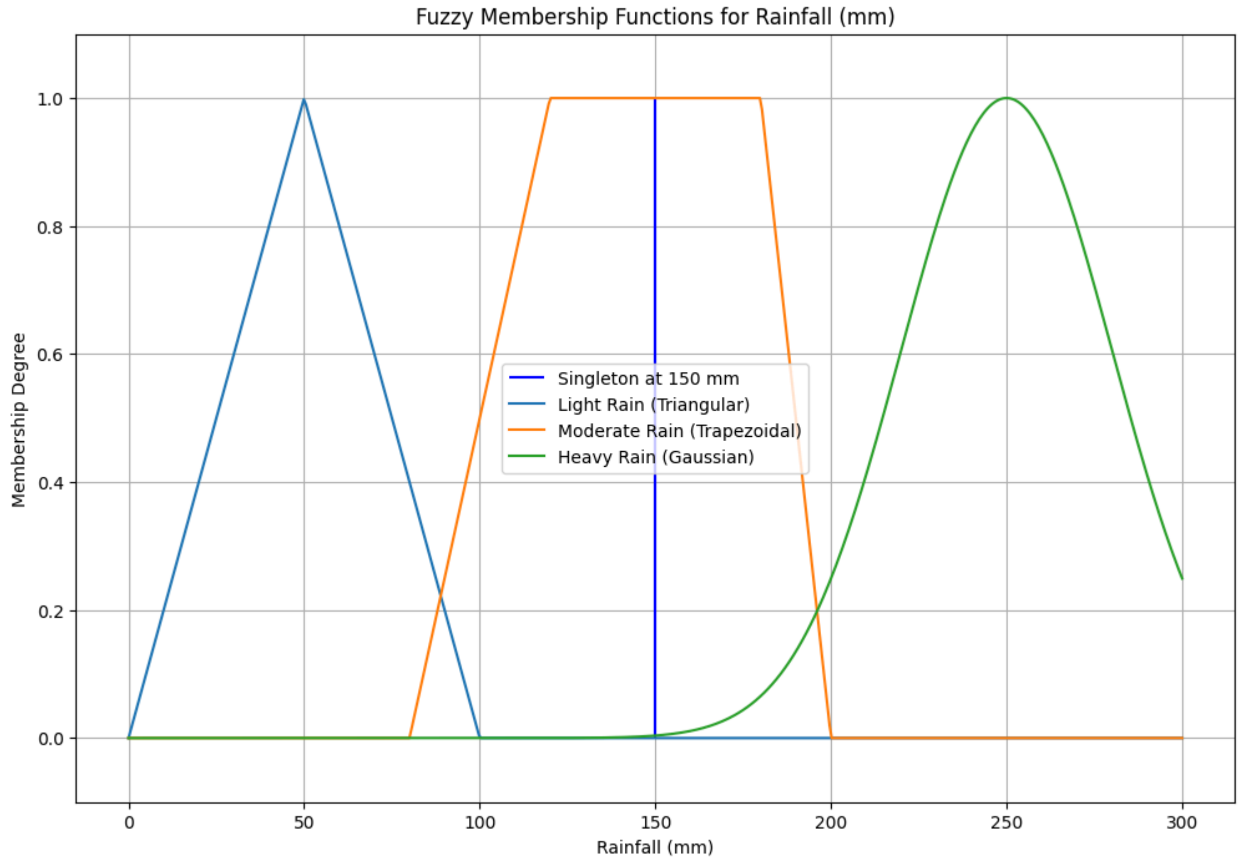
```
# Gaussian Membership Function
def gaussian_mf(x, c, sigma):
    return np.exp(-((x-c)**2) / (2*sigma**2))
```

```
# Rainfall range (in mm) for plotting
x = np.linspace(0, 300, 500)  # 0 mm to 300 mm
```

```
# Define different fuzzy sets for Rainfall
singleton_rain = singleton_mf(x, 150)  # Singleton at exactly 150 mm
light_rain = triangular_mf(x, 0, 50, 100)  # Light rainfall
moderate_rain = trapezoidal_mf(x, 80, 120, 180, 200)  # Moderate rainfall
heavy_rain = gaussian_mf(x, 250, 30)  # Heavy rainfall, smooth bell curve
```

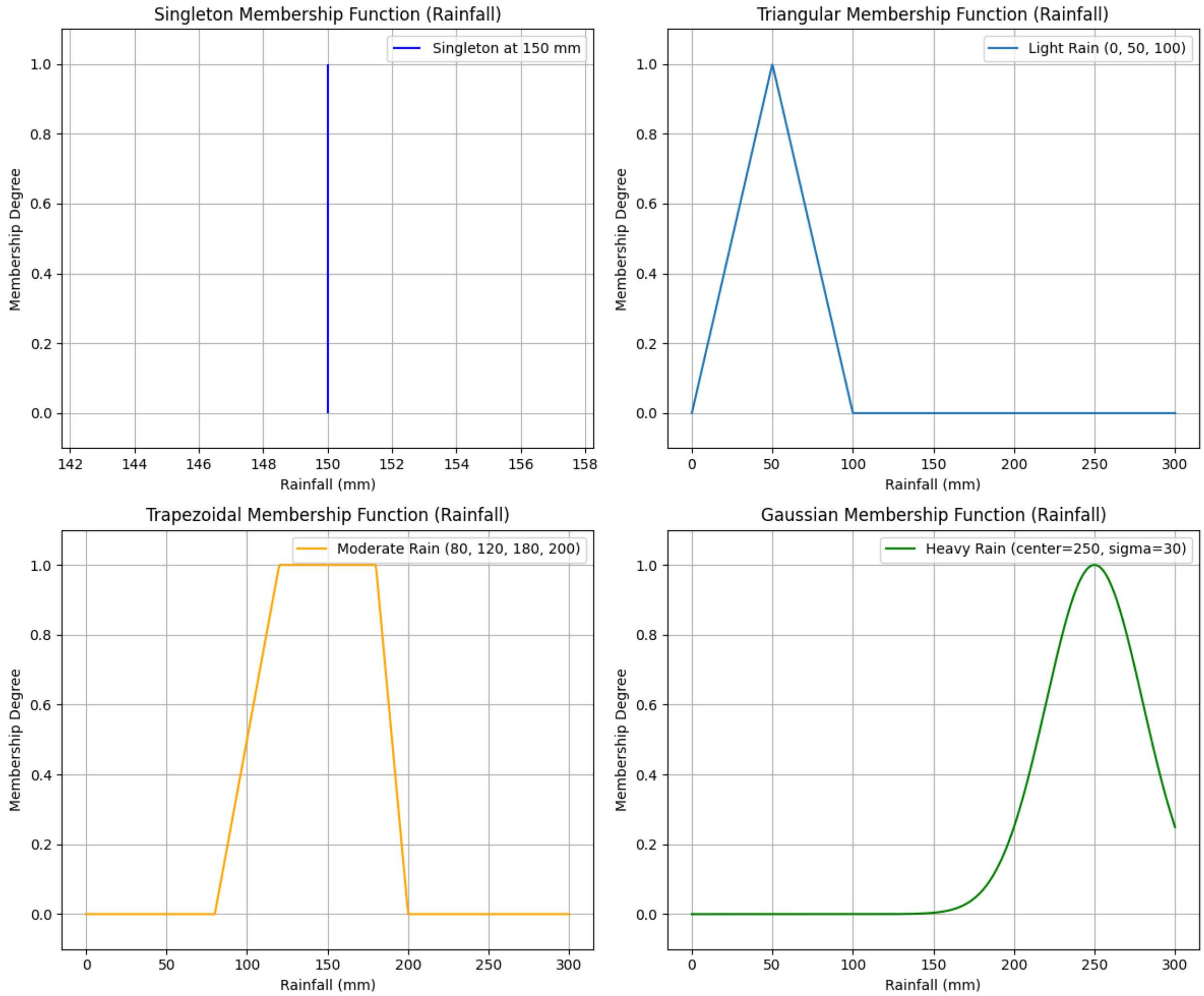
```
plt.figure(figsize=(12, 8))

# Singleton shown as vertical line
plt.vlines(150, 0, 1, colors='blue', label="Singleton at 150 mm",
linestyles='solid')

# Other membership functions
plt.plot(x, light_rain, label="Light Rain (Triangular)")
plt.plot(x, moderate_rain, label="Moderate Rain (Trapezoidal)")
plt.plot(x, heavy_rain, label="Heavy Rain (Gaussian)")

# Labels and Formatting
plt.title("Fuzzy Membership Functions for Rainfall (mm)")
plt.xlabel("Rainfall (mm)")
plt.ylabel("Membership Degree")
plt.ylim(-0.1, 1.1)
```

```
plt.legend()
plt.grid(True)
plt.show()
```




```
# Plot each membership function individually
plt.figure(figsize=(12, 10))

# 1. Singleton Membership Function
plt.subplot(2, 2, 1)
plt.vlines(150, 0, 1, colors='blue', linestyles='solid', label="Singleton
at 150 mm")
plt.title("Singleton Membership Function (Rainfall)")
plt.xlabel("Rainfall (mm)")
plt.ylabel("Membership Degree")
plt.ylim(-0.1, 1.1)
plt.legend()
plt.grid(True)

# 2. Triangular Membership Function
plt.subplot(2, 2, 2)
plt.plot(x, light_rain, label="Light Rain (0, 50, 100)")
plt.title("Triangular Membership Function (Rainfall)")
plt.xlabel("Rainfall (mm)")
```

```
plt.ylabel("Membership Degree")
plt.ylim(-0.1, 1.1)
plt.legend()
plt.grid(True)

# 3. Trapezoidal Membership Function
plt.subplot(2, 2, 3)
plt.plot(x, moderate_rain, label="Moderate Rain (80, 120, 180, 200)",
color='orange')
plt.title("Trapezoidal Membership Function (Rainfall)")
plt.xlabel("Rainfall (mm)")
plt.ylabel("Membership Degree")
plt.ylim(-0.1, 1.1)
plt.legend()
plt.grid(True)

# 4. Gaussian Membership Function
plt.subplot(2, 2, 4)
plt.plot(x, heavy_rain, label="Heavy Rain (center=250, sigma=30)",
color='green')
plt.title("Gaussian Membership Function (Rainfall)")
plt.xlabel("Rainfall (mm)")
plt.ylabel("Membership Degree")
plt.ylim(-0.1, 1.1)
plt.legend()
plt.grid(True)

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



Conclusion:

In this experiment, we learned how different fuzzy membership functions like Singleton, Triangular, Trapezoidal, and Gaussian work to represent uncertain data. By using the rainfall example, we saw how each function gives a degree of membership between 0 and 1, helping to model real-life situations where boundaries are not exact.

AI&DS2_Expt_06