Batch: C2 **Roll No.: 16010122323**

16010122320

Experiment No. : 07

Grade: AA / AB / BB / BC / CC / CD /DD

Title: Implementation of de-fuzzification methods (Center of Sum, Centre of Gravity, Mean of Maximum).

Objective: To understand de-fuzzification methods

Expected Outcome of Experiment:

CO4 : Apply basics of Fuzzy logic and neural networks

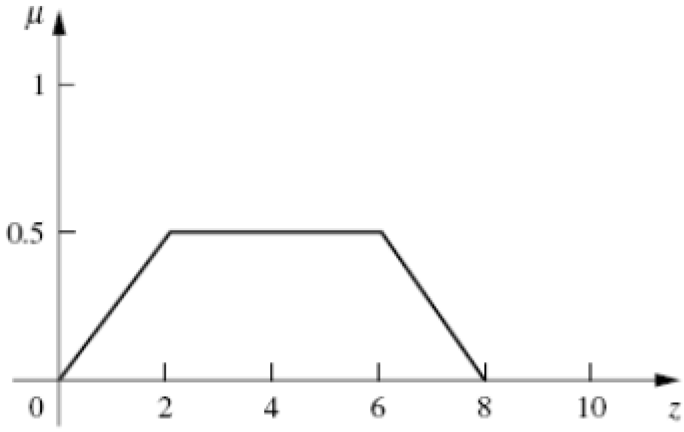
Books/ Journals/ Websites referred:

Pre Lab/ Prior Concepts:

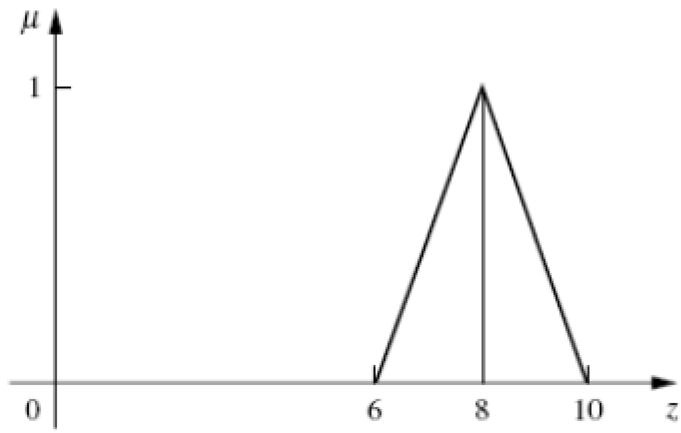
Defuzzification :

Defuzzification is the process of producing a quantifiable result in Crisp logic, given fuzzy sets and corresponding membership degrees. It is the process that maps a fuzzy set to a crisp set. It is typically needed in fuzzy control systems. These will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets. Defuzzification is the conversion of a fuzzy quantity to a precise quantity, just as fuzzification is the conversion of a precise quantity to a fuzzy quantity. μ

For example, **Fig (a)** shows the first part of the Fuzzy output and **Fig (b)** shows the second part of the Fuzzy output.

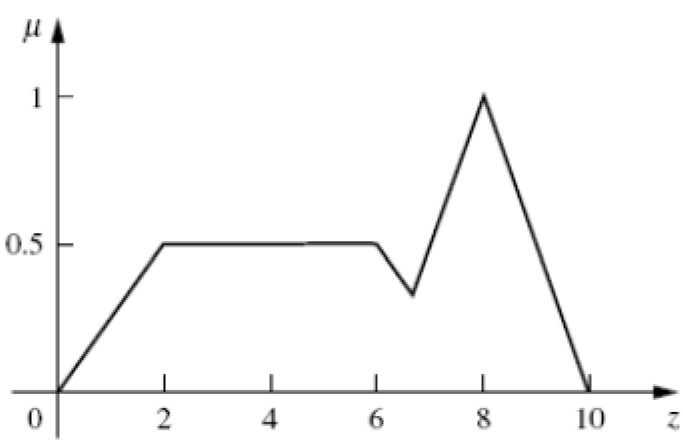


(a)



(b)

Then **Fig (c)** shows the union of the two parts (a) and (b).



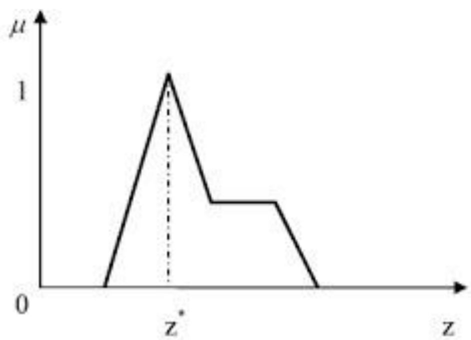
(c)

Different Defuzzification methods

1. **Max membership method**

This method is also known as height method and is limited to peak output functions. This method is given by the algebraic expression:

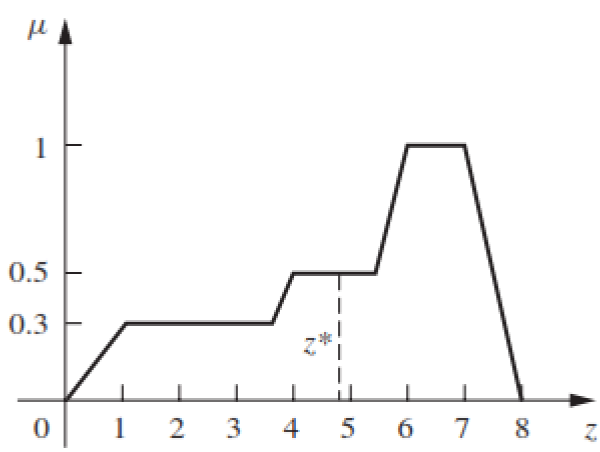
$\boldsymbol{\mu}(z^*) >= \boldsymbol{\mu}(z)$ for all $z \in Z$.



2. **Center of gravity or centroid**

This method is also known as the centre of mass, centre of area or centre of gravity. It is the most commonly used defuzzification method. The defuzzified output z* is given by:

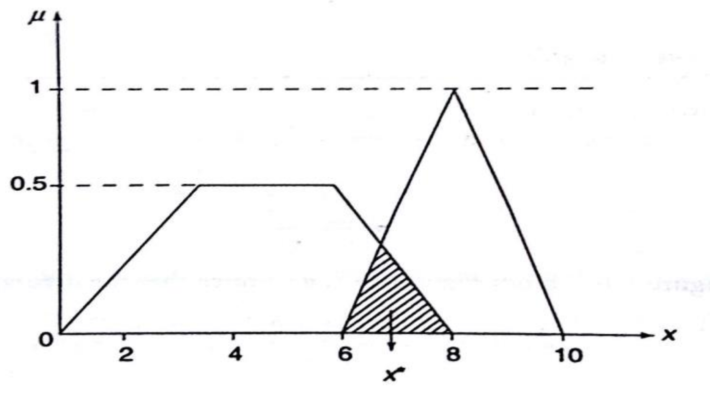
$\mathbf{z^* = \int \mu(z).z dz / \int \mu(z) dz}$



3. Centre of sums

This method employs the algebraic sum of the individual fuzzy subsets instead of their union. The calculations here are very fast, but the main drawback is that the intersecting areas are added twice. The defuzzified value z* is given by

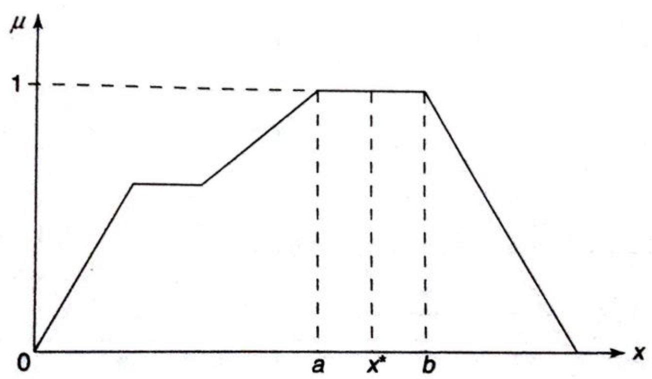
$$z^* = \int z^* \sum \mu(z).z dz / \int \sum \mu(z) dz$$



4. Mean of maximum method

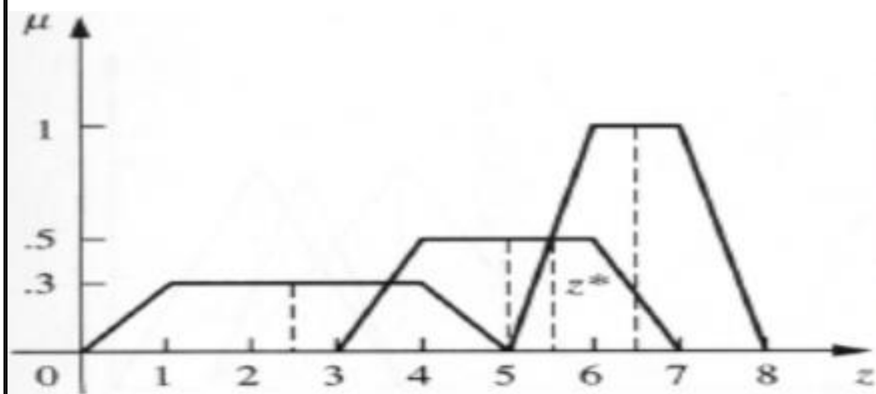
This method is also known as the middle of the maxima. This is closely related to the max-membership method, except that the locations of the maximum membership can be nonunique. The output here is given by:

$z^* = \sum z' / n$; where z' is the maximum value of the membership function.



5. **Weighted average method**

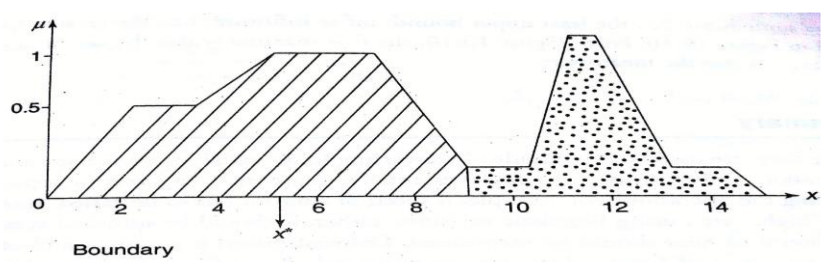
This method is valid for symmetrical output membership functions only. Each membership function is weighted by its maximum membership value. The output in the case is given by $z^* = \sum\mu(z').z' / \sum\mu(z')$; where z' is the maximum value of the membership function.



6. **Centre of Largest Area**

This method can be adopted when the output of at least two convex fuzzy subsets which are not overlapping. The output, in this case, is biased towards a side of one membership function. When output fuzzy st has at least two convex regions, then the centre of gravity of the convex fuzzy subregion having the largest are is used to obtain the defuzzified value z*. The value is given by

$z^* = \int \mu c(z).zdz / \int \sum\mu c(z)dz$



Implementation Details:

Implement the above 6 defuzzification methods (draw graph)

Code and Snap shot of graph

Untitled7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

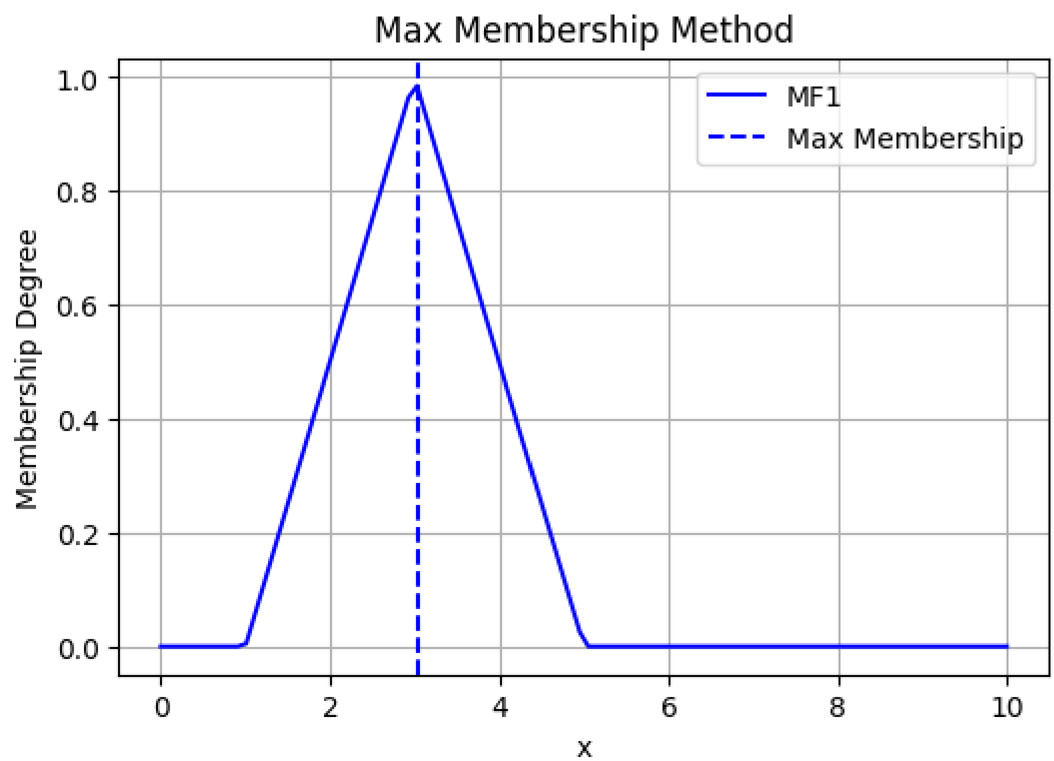
+ Code + Text

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

# Define the triangular membership function
def triangular_mf(x, a, b, c):
    """Triangular membership function with points a, b, and c"""
    return np.maximum(np.minimum((x - a) / (b - a), (c - x) / (c - b)), 0)

# Generate input values
x = np.linspace(0, 10, 100)

# Define fuzzy membership functions
mf1 = triangular_mf(x, 1, 3, 5)  # Membership function 1
mf2 = triangular_mf(x, 4, 6, 8)  # Membership function 2
mf3 = triangular_mf(x, 7, 8, 10)  # Membership function 3

# Defuzzification Methods

# 1. Max Membership Method (Height Method)
def max_membership(mf, x):
    return x[np.argmax(mf)]

max_mf_value = max_membership(mf1, x)
```

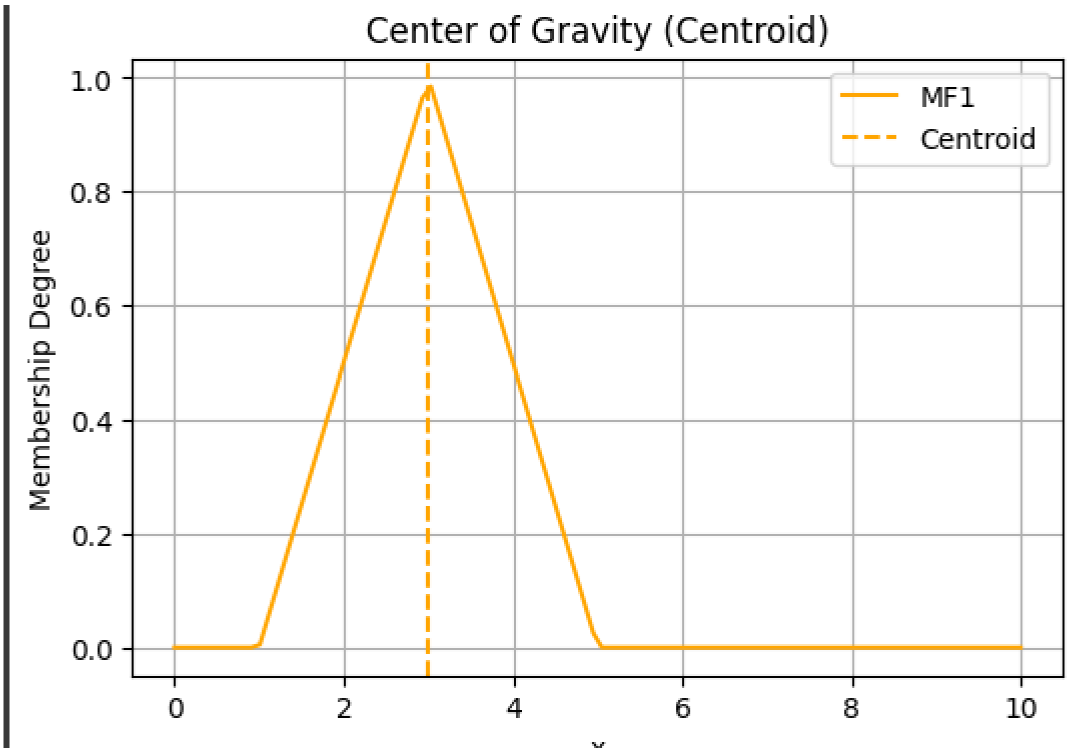
```python
# 2. Center of Gravity (Centroid)
def centroid(mf, x):
    return np.sum(x * mf) / np.sum(mf)

centroid_value = centroid(mf1, x)

# 3. Center of Sums
def center_of_sums(mf1, mf2, x):
    summed_mf = mf1 + mf2
    return np.sum(x * summed_mf) / np.sum(summed_mf)

center_sums_value = center_of_sums(mf1, mf2, x)

# 4. Mean of Maximum (Middle of Maxima)
def mean_of_maximum(mf, x):
    max_indices = np.where(mf == np.max(mf))
    return np.mean(x[max_indices])

mean_max_value = mean_of_maximum(mf1, x)
```

```python
# 5. Weighted Average Method
def weighted_average(mf, x):
    return np.sum(mf * x) / np.sum(mf)

weighted_avg_value = weighted_average(mf1, x)

# 6. Center of Largest Area
center_largest_area_value = centroid(mf1, x)
```

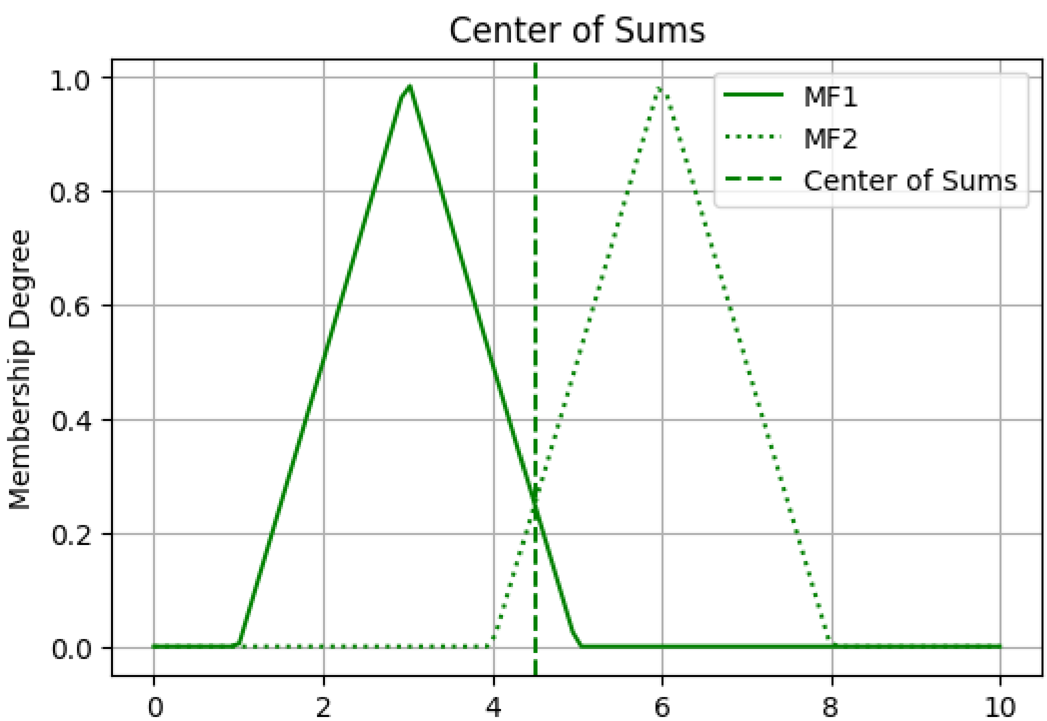
```
# Plotting each defuzzification method

# 1. Max Membership Method
plt.figure(figsize=(6, 4))
plt.plot(x, mf1, label='MF1', color='blue')
plt.axvline(max_mf_value, color='blue', linestyle='--', label='Max Membership')
plt.title("Max Membership Method")
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.legend()
plt.grid(True)
plt.show()
```



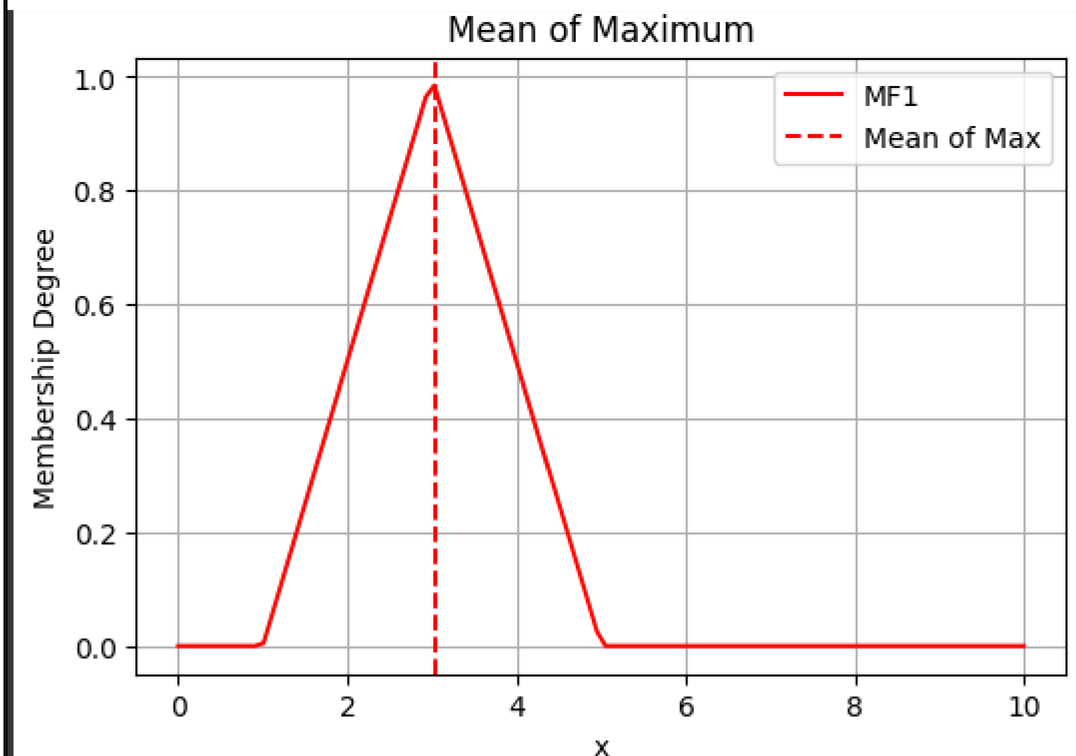
```
# 2. Center of Gravity (Centroid)
plt.figure(figsize=(6, 4))
plt.plot(x, mf1, label='MF1', color='orange')
plt.axvline(centroid_value, color='orange', linestyle='--', label='Centroid')
plt.title("Center of Gravity (Centroid)")
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.legend()
plt.grid(True)
plt.show()
```



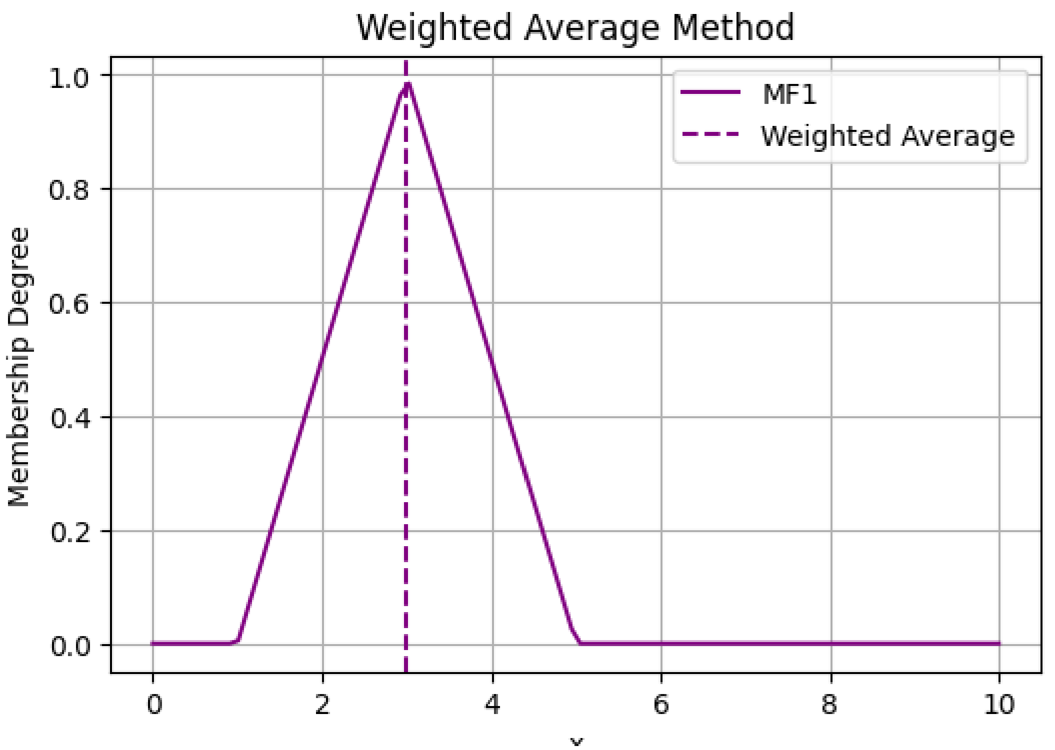
```python
# 3. Center of Sums
plt.figure(figsize=(6, 4))
plt.plot(x, mf1, label='MF1', color='green')
plt.plot(x, mf2, label='MF2', color='green', linestyle='dotted')
plt.axvline(center_sums_value, color='green', linestyle='--', label='Center of Sums')
plt.title("Center of Sums")
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.legend()
plt.grid(True)
plt.show()
```



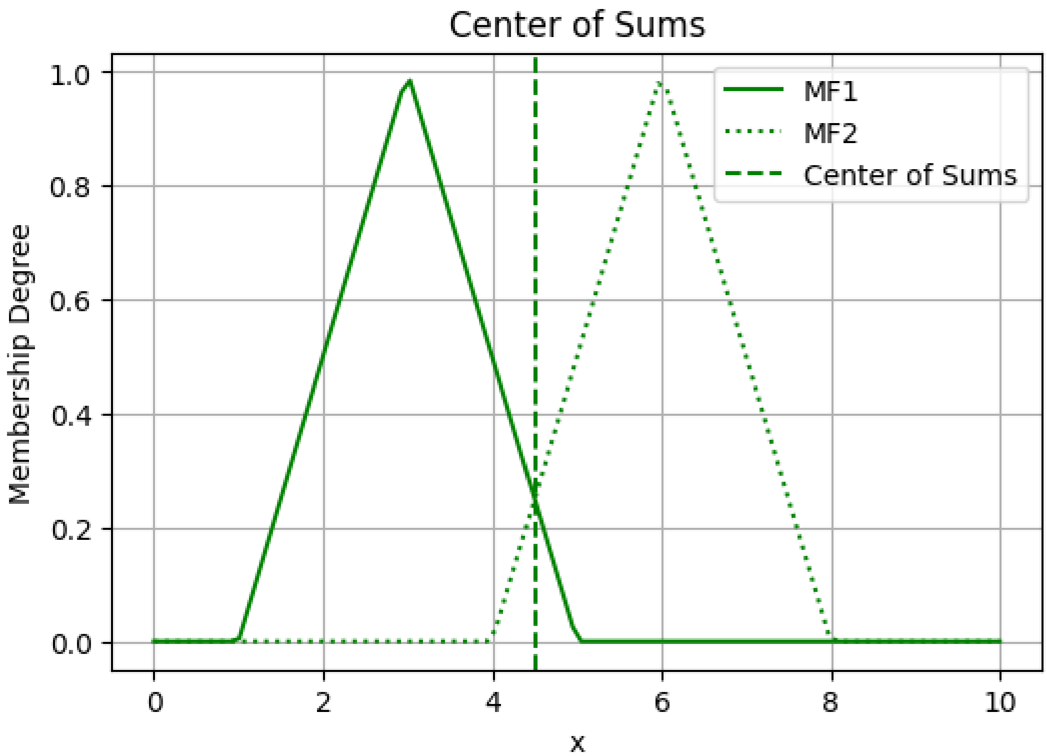
```python
# 4. Mean of Maximum
plt.figure(figsize=(6, 4))
plt.plot(x, mf1, label='MF1', color='red')
plt.axvline(mean_max_value, color='red', linestyle='--', label='Mean of Max')
plt.title("Mean of Maximum")
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.legend()
plt.grid(True)
plt.show()
```




```python
# 5. Weighted Average Method
plt.figure(figsize=(6, 4))
plt.plot(x, mf1, label='MF1', color='purple')
plt.axvline(weighted_avg_value, color='purple', linestyle='--', label='Weighted Average')
plt.title("Weighted Average Method")
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.legend()
plt.grid(True)
plt.show()
```



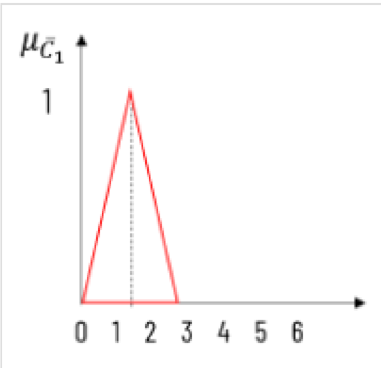

```python
# 6. Center of Largest Area
plt.figure(figsize=(6, 4))
plt.plot(x, mf1, label='MF1', color='brown')
plt.axvline(center_largest_area_value, color='brown', linestyle='--', label='Center of Largest Area')
plt.title("Center of Largest Area")
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.legend()
plt.grid(True)
plt.show()
```



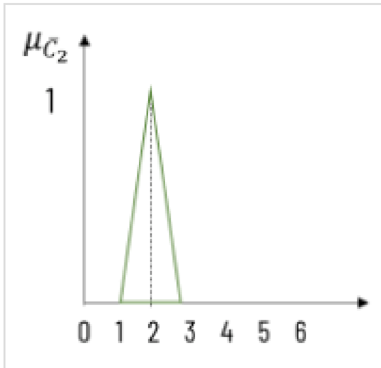
Conclusion: Implementation of defuzzification methods was done successfully.

Post Lab Descriptive Questions :

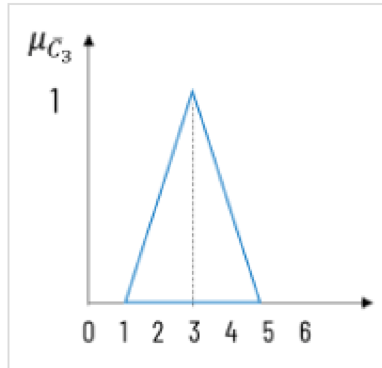
Find crisp value corresponding to the following fuzzy output sets using various de-fuzzification methods.



Output fuzzy set 1



Output fuzzy set 2



Output fuzzy set 3

To find the crisp values corresponding to the given fuzzy output sets using various defuzzification methods, we can apply a few common techniques such as the **Centroid method**, **Mean of Maximum (MoM)**, and **Bisector method**. Let's break it down for each output fuzzy set:

1. **Centroid Method** (also known as the Center of Gravity): This method calculates the center of gravity of the fuzzy set. It is done by finding the weighted average of all possible values within the fuzzy set.

   Formula:

   Formula:

   $$\text{Crisp Value} = \frac{\int x \cdot \mu(x)\,dx}{\int \mu(x)\,dx}$$

   where $\mu(x)$ is the membership function.

   Mean of Maximum (MoM): This method selects the average of the maximum membership values. If multiple values have the same maximum membership, we take the mean of those values.

2. **Bisector Method**: This method finds the value where the fuzzy set is divided into two regions with equal area.

Given the triangular shapes of the fuzzy sets in the image:

- **Output fuzzy set 1**: A triangular membership function with the peak at 3 and the base ranging from 1 to 5.
- **Output fuzzy set 2**: A triangular membership function with the peak at 3 and the base ranging from 2 to 4.
- **Output fuzzy set 3**: A triangular membership function with the peak at 4 and the base ranging from 2 to 6.

Approximate Defuzzification Results:

1. **Centroid Method**:
   - **Set 1**: The centroid would be near 3.
   - **Set 2**: The centroid would also be near 3, as the triangular shape is symmetric around 3.
   - **Set 3**: The centroid would be around 4.
2. **Mean of Maximum (MoM)**:
   - **Set 1**: The maximum membership is at 3, so the crisp value is 3.
   - **Set 2**: The maximum membership is at 3, so the crisp value is 3.
   - **Set 3**: The maximum membership is at 4, so the crisp value is 4.
3. **Bisector Method**:
   - **Set 1**: The bisector divides the area equally, which would again result in a value near 3.
   - **Set 2**: The bisector would yield a value near 3.
   - **Set 3**: The bisector would yield a value around 4.

In conclusion:

- For **Set 1**, the crisp value is approximately 3 for all methods.
- For **Set 2**, the crisp value is approximately 3 for all methods.
- For **Set 3**, the crisp value is approximately 4 for all methods.

These methods can be applied depending on the desired characteristics of the defuzzification process.