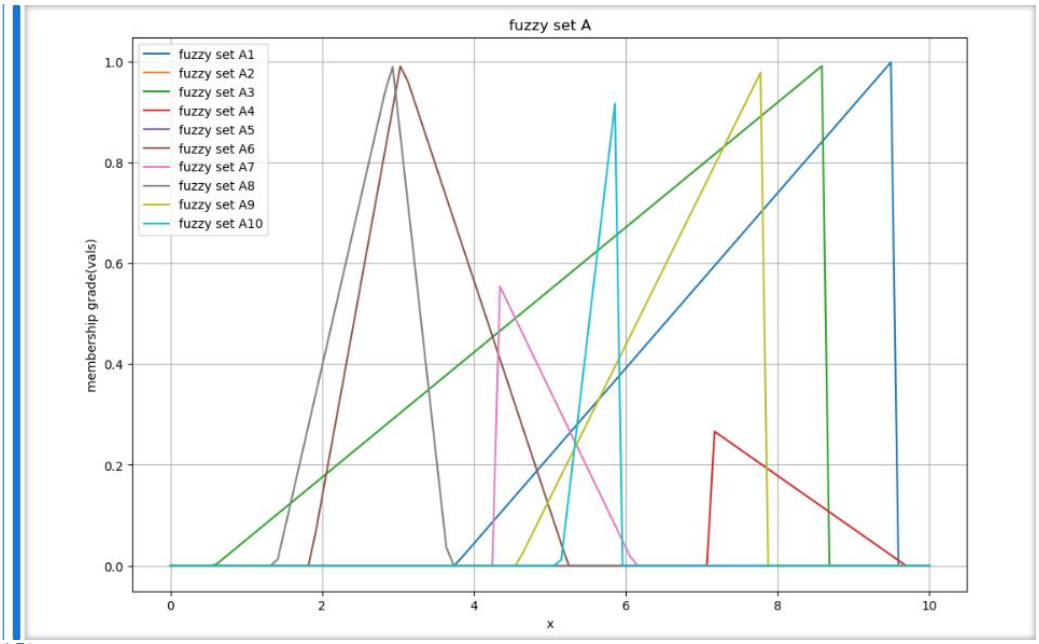
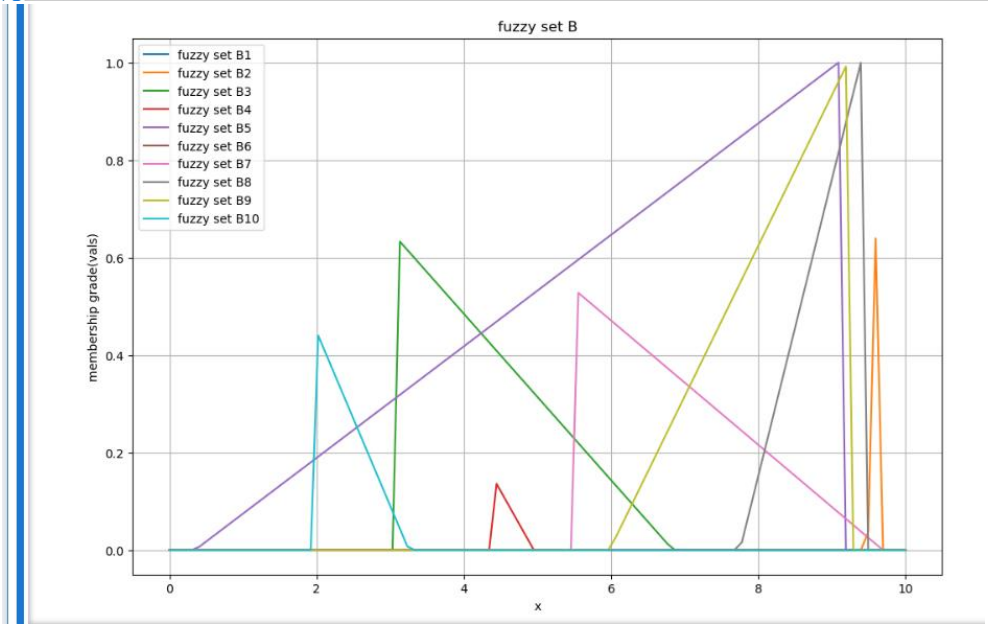
fuzzy combine & plot

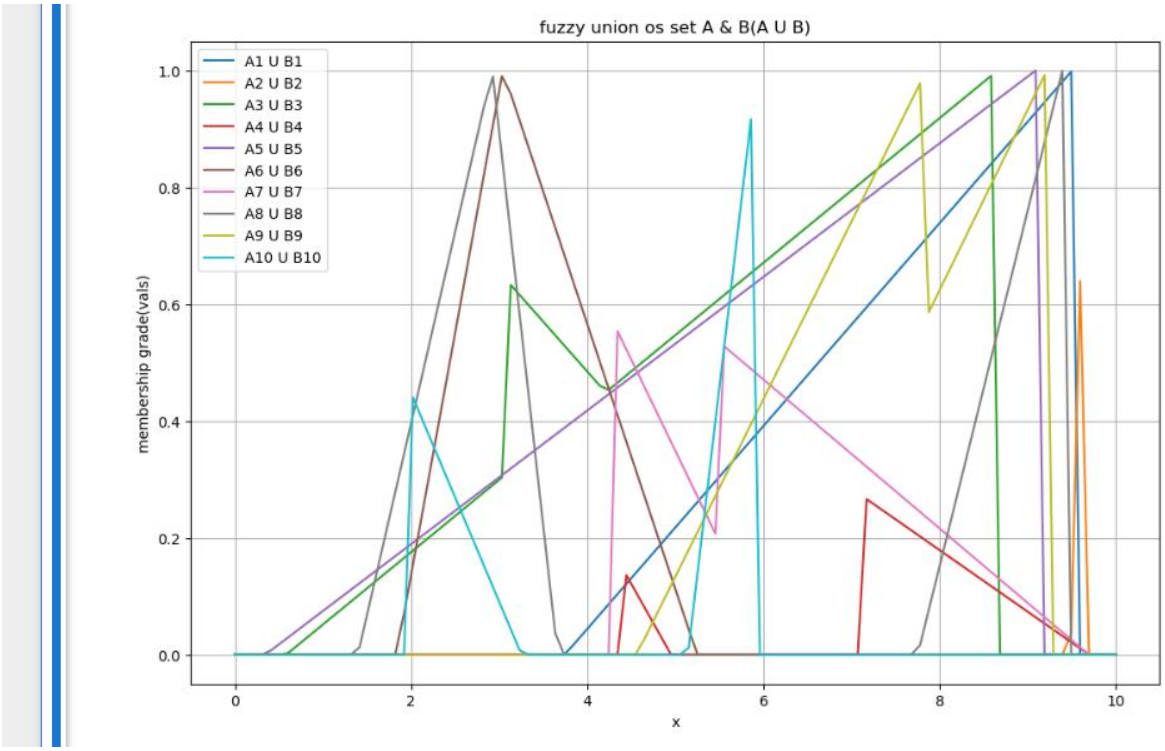
```
import numpy as np
import matplotlib.pyplot as plt

#define fuzzy set A & B(randomly)
np.random.seed(42)
num_eles=10 #no. of eles in each fuzzy set
param_A=np.random.uniform(low=0,high=10,size=(num_eles,3))  #10 rows & 3 cols in each fuzzy set
param_B=np.random.uniform(low=0,high=10,size=(num_eles,3))  #10 rows & 3 cols in each fuzzy set

#define triangular MF
def tri(x,a,b,c):
    return np.where(x <= a, 0,
            np.where(x < b, (x - a) / (b - a),
                np.where(x < c, (c - x) / (c - b), 0)))

#define x(an array of 100 evenly spaced random no.s started from 0 & ended at 10)
x=np.linspace(0,10,100)

#initialize list that wil contain membership vals
mv_A_list=[]
mv_B_list=[]
mv_union_list=[]

#cal. memebership vals
for (a1,b1,c1),(a2,b2,c2) in zip(param_A,param_B):   #zip() function pairs the first tuple from param_A with the first tuple from param_B, the second tuple from each, and so on.
    mv_A=tri(x,a1,b1,c1)
    mv_B=tri(x,a2,b2,c2)
    mv_union=np.maximum(mv_A,mv_B)
```

```
    mv_A_list.append(mv_A)
    mv_B_list.append(mv_B)
    mv_union_list.append(mv_union)



#plotting (MV of each eles of a set)
#plot fuzzy set A
plt.figure(figsize=(12,8))
for i,mv_A in enumerate(mv_A_list):  #numerate() is used to loop over mv_A_list while simultaneously providing the index i and the value mv_A in each iteration.
    plt.plot(x,mv_A,label=f'fuzzy set A{i+1}')
plt.title("fuzzy set A")
plt.xlabel("x")
plt.ylabel("membership grade(vals)")
plt.legend()
plt.grid(True)
plt.show()

#plot fuzzy set B
plt.figure(figsize=(12,8))
for i,mv_B in enumerate(mv_B_list):
    plt.plot(x,mv_B,label=f'fuzzy set B{i+1}')
plt.title("fuzzy set B")
plt.xlabel("x")
plt.ylabel("membership grade(vals)")
plt.legend()
plt.grid(True)
plt.show()
```

```
#plot fuzzy set A
plt.figure(figsize=(12,8))
for i,mv_union in enumerate(mv_union_list):
    plt.plot(x,mv_union,label=f'A{i+1} U B{i+1}')
plt.title("fuzzy union os set A & B(A U B)")
plt.xlabel("x")
plt.ylabel("membership grade(vals)")
plt.legend()
plt.grid(True)
plt.show()
```






fuzzy union os set A & B(A U B)
A1 U B1
A2 U B2
A3 U B3
A4 U B4
A5 U B5
A6 U B6
A7 U B7
A8 U B8
A9 U B9
A10 U B10
membership grade(vals)
x
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10