**Problem 2**

x = [0:1:100];

mf = trapmf(x,[10 30 50 90]);

hold on

grid on

axis([ 0 100 0 1.25])

plot(x,mf)

xlabel('X = Universe of Discourse');

ylabel('Membership Grade');

title('Trapazoid Membership Function and Defuzzification');

% Center of Area (CoA) Defuzzification

% z\_coa = defuzz(x,mf,'centroid') % using Fuzzy Logic Toolbox

z\_coa = trapz(mf.\*x)/trapz(mf); % trapz() 'estimates' integral of a trapazoid

stem(z\_coa,1,'b')

% Mean of Max (MoM) Defuzzification

% z\_mom = defuzz(x,mf,'mom') % using Fuzzy Logic Toolbox

index = find(mf==max(mf)); % all index(s) of mf that equal the max value

mean\_index = mean(index); % assuming the max area is contiguous...

z\_mom = x(mean\_index); % ...can use index with input vector to find z

stem(z\_mom,1,'r')

% Bisector of Area (BoA) Defuzzification

% z\_boa = defuzz(x,mf,'bisector') % using Fuzzy Logic Toolbox

index=1;

while (logical(trapz(mf(1:index))~=trapz(mf(index:length(mf)))))

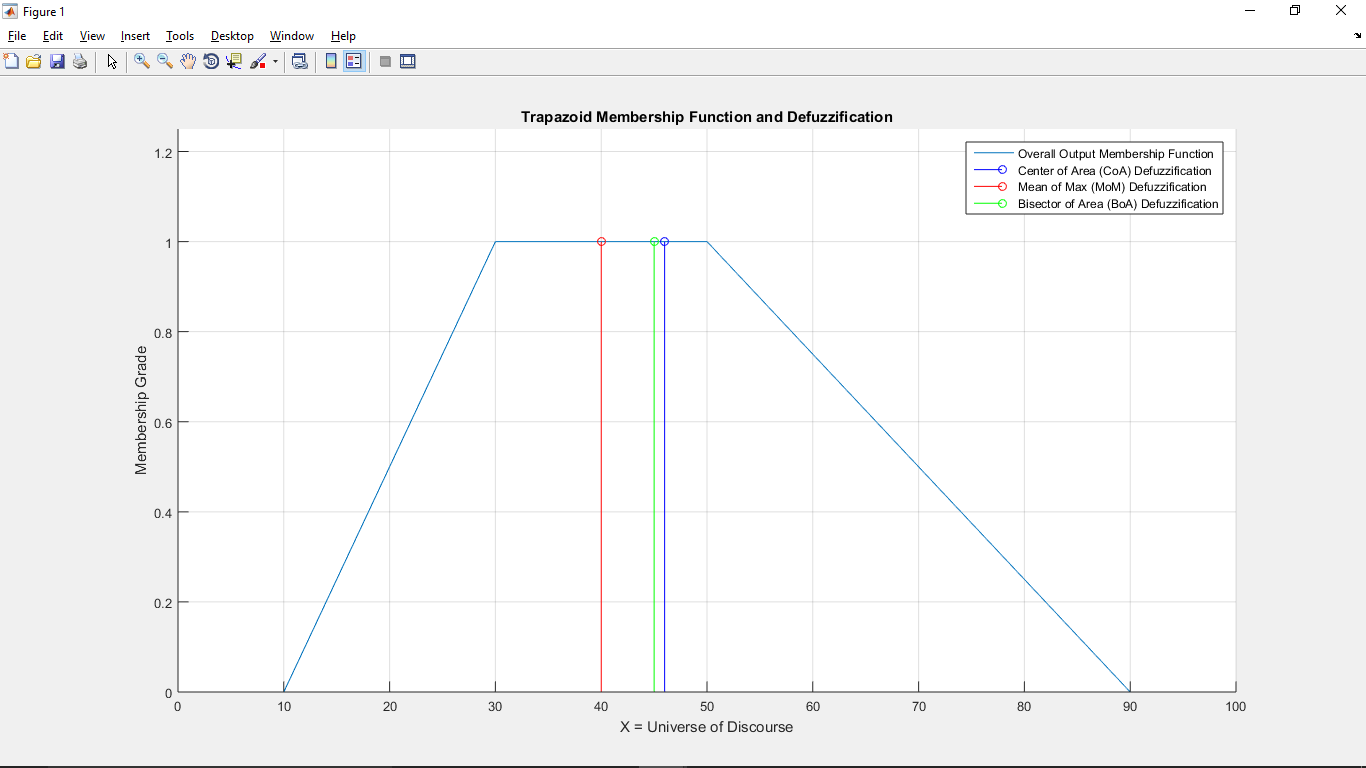
index=index+1;

end

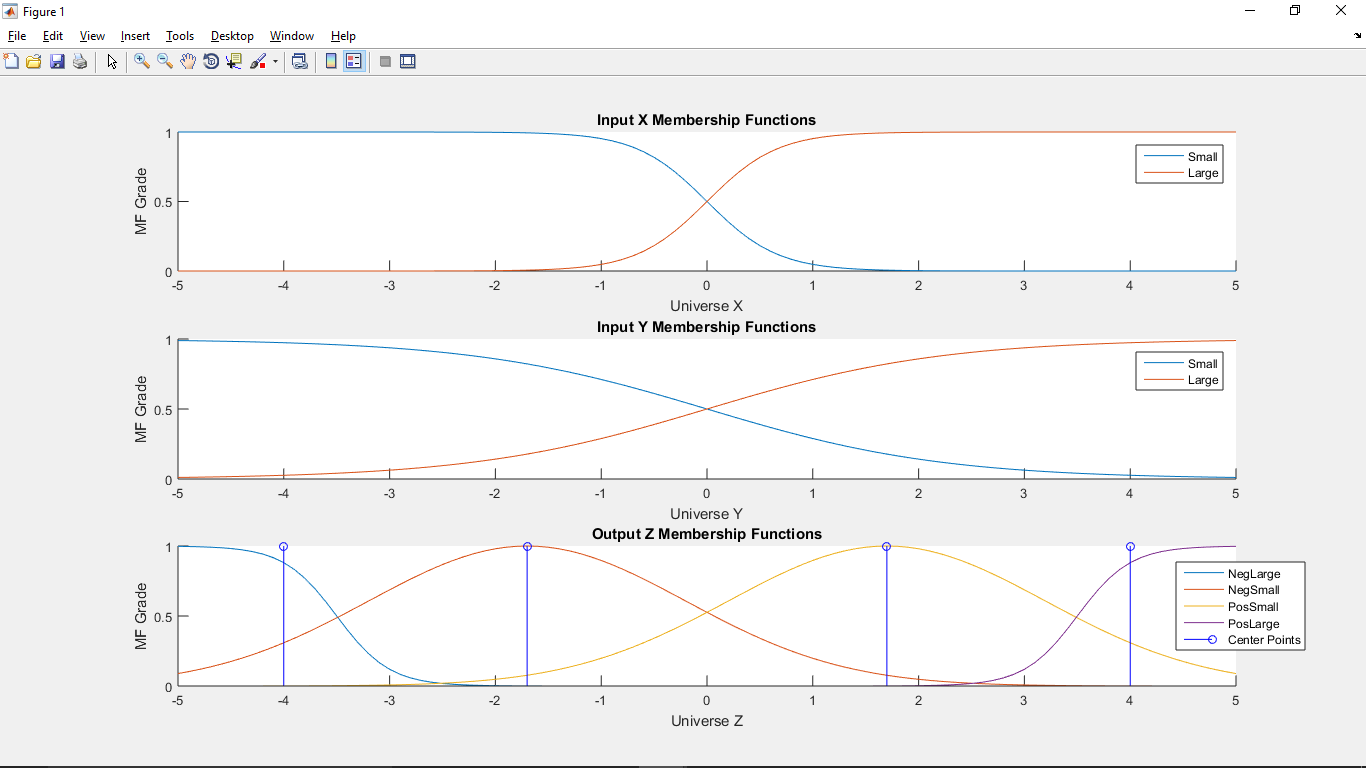
z\_boa = x(index);

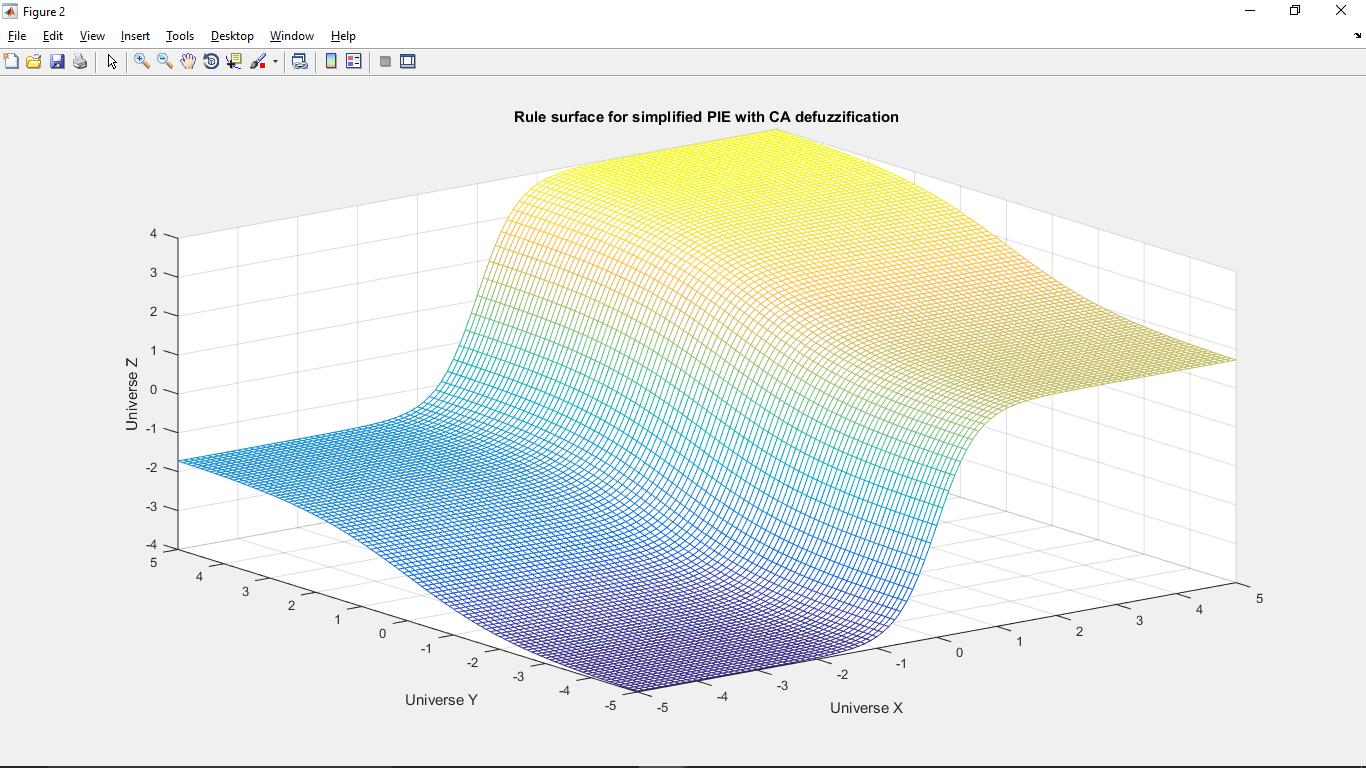
stem(z\_boa,1,'g')

legend('Overall Output Membership Function', 'Center of Area (CoA) Defuzzification', 'Mean of Max (MoM) Defuzzification', 'Bisector of Area (BoA) Defuzzification')



**Problem 3**





% Create simplified PIE with CA defuzzification using the following inputs, sets, rules

x = [-5:.1:5];

y = x;

% X input MF's

xsmall = 1./(1+exp(-(-3)\*(x-0)));

xlarge = 1./(1+exp(-(3)\*(x-0)));

subplot(3,1,1);

hold on

plot(x,xsmall);

plot(x,xlarge);

xlabel('Universe X');

ylabel('MF Grade');

title('Input X Membership Functions');

legend('Small', 'Large')

% Y input MF's

ysmall = 1./(1+exp(-(-0.9)\*(y-0)));

ylarge = 1./(1+exp(-(0.9)\*(y-0)));

subplot(3,1,2);

hold on

plot(y,ysmall);

plot(y,ylarge);

xlabel('Universe Y');

ylabel('MF Grade');

title('Input Y Membership Functions');

legend('Small', 'Large')

% Z output MF's

zneglarge = 1./(1+exp(-(-4)\*(x-(-3.5))));

znegsmall = exp(-0.5\*(((x-(-1.7))/1.5).^2));

zpossmall = exp(-0.5\*(((x-(1.7))/1.5).^2));

zposlarge = 1./(1+exp(-(4)\*(x-(3.5))));

subplot(3,1,3);

hold on

plot(x,zneglarge);

plot(x,znegsmall);

plot(x,zpossmall);

plot(x,zposlarge);

stem(-4,1,'b'); % center points, in same order

stem(-1.7,1,'b');

stem(1.7,1,'b');

stem(4,1,'b');

xlabel('Universe Z');

ylabel('MF Grade');

title('Output Z Membership Functions');

legend('NegLarge', 'NegSmall', 'PosSmall', 'PosLarge', 'Center Points')

%% Product Inference Engine (PIE) with CA Defuzzification

% Determine Z fuzzy set center points for defuzzification

zneglarge\_center = -4; % sigmoids are arbitrary, based on class slides

zposlarge\_center = 4;

znegsmall\_center = -1.7; % gaussian center = crossover point z2 - z1 / 2

zpossmall\_center = 1.7;

% Crisp inputs are used as fuzzy singleton facts

xfacts = [-5:.1:5];

yfacts = [-5:.1:5];

% Evaluate X and Y membership functions at these fact values

xfacts\_xsmall\_MFgrade = 1./(1+exp(-(-3)\*(xfacts-0)));

xfacts\_xlarge\_MFgrade = 1./(1+exp(-(3)\*(xfacts-0)));

yfacts\_ysmall\_MFgrade = 1./(1+exp(-(-0.9)\*(yfacts-0)));

yfacts\_ylarge\_MFgrade = 1./(1+exp(-(0.9)\*(yfacts-0)));

z\_ca = [];

for i=1:length(yfacts),

% Rule 1, if x is small and y is small then z is neglarge

wR1 = xfacts\_xsmall\_MFgrade \* yfacts\_ysmall\_MFgrade(i);

wR1\_center = wR1 \* zneglarge\_center;

% Rule 2, if x is small and y is large then z is negsmall

wR2 = xfacts\_xsmall\_MFgrade \* yfacts\_ylarge\_MFgrade(i);

wR2\_center = wR2 \* znegsmall\_center;

% Rule 3, if x is large and y is small then z is possmall

wR3 = xfacts\_xlarge\_MFgrade \* yfacts\_ysmall\_MFgrade(i);

wR3\_center = wR3 \* zpossmall\_center;

% Rule 4, if x is large and y is large then z is poslarge

wR4 = xfacts\_xlarge\_MFgrade \* yfacts\_ylarge\_MFgrade(i);

wR4\_center = wR4 \* zposlarge\_center;

sum\_wR\_center = wR1\_center + wR2\_center + wR3\_center + wR4\_center;

sum\_wR = wR1 + wR2 + wR3 + wR4; % in this case = 1

% ouput center average vector for 1 value of y

z\_ca = [z\_ca; (sum\_wR\_center ./ sum\_wR)];

end

figure()

mesh(x,y,z\_ca) % rule surface

xlabel('Universe X');

ylabel('Universe Y');

zlabel('Universe Z');

title('Rule surface for simplified PIE with CA defuzzification');