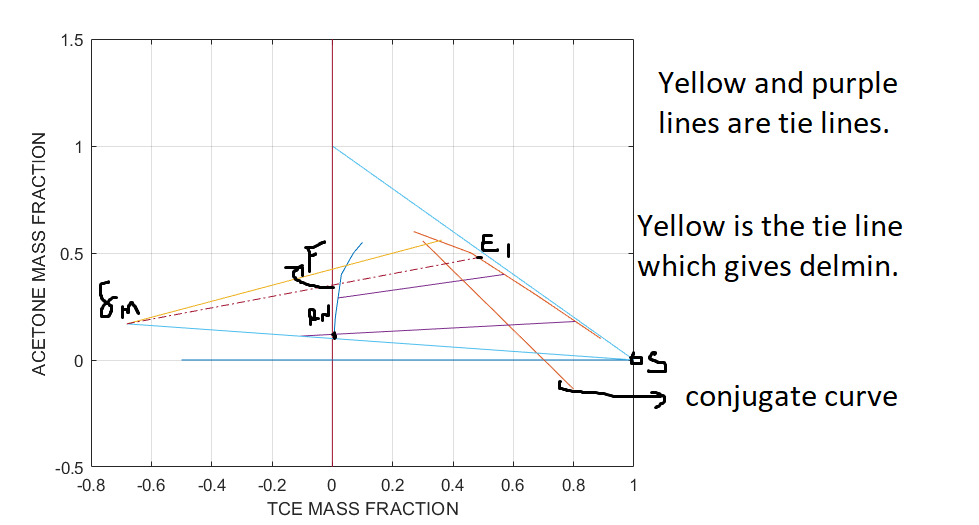
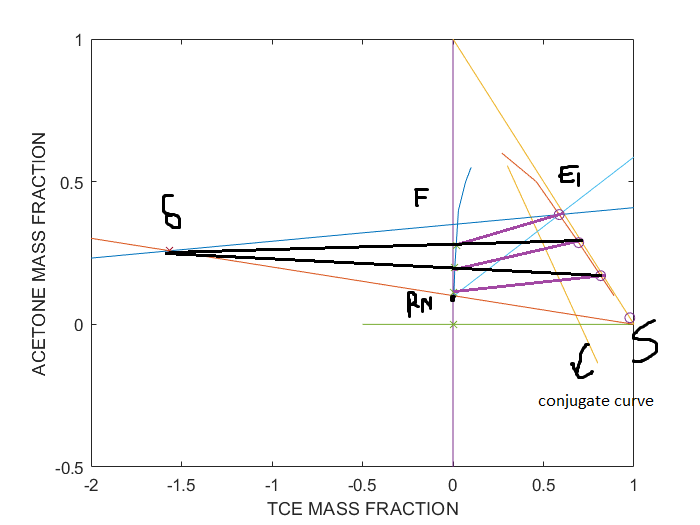
# Question 1 – RIGHT TRIANGLE METHOD

1. Delta point for minimum solvent flow rate



1. And c) Number of stages and composition of each stream at the exit of each stage



## Code for evaluating the points and plotting:

close all; clear;

%% Equilibrium Data

r\_tce = [0.1,0.07,0.03,0.02,0.01,0.005];

r\_a = [0.55,0.5,0.4,0.3,0.2,0.1];

r\_p = spline(r\_a,r\_tce);

e\_a = [0.6,0.5,0.4,0.3,0.2,0.1];

e\_tce = [0.27,0.46,0.57,0.68,0.785,0.89];

e\_p = spline(e\_a,e\_tce);

%% Given Feed Data

yF =0.35;

yRN = 0.1;

F = 1300;

%% Tie Line data

ac\_raff = [0.44,0.29,0.12];

ac\_ext = [0.56,0.4,0.18];

tce\_r = ppval(r\_p,ac\_raff);

tce\_e = ppval(e\_p,ac\_ext);

xRN = ppval(r\_p,yRN);

%% Plotting the data

plot(r\_tce,r\_a,e\_tce,e\_a);

hold on; grid on; grid minor;

for i = 1:3

plot([tce\_r(i),tce\_e(i)],[ac\_raff(i),ac\_ext(i)]);

end

%Completing the triangle

plot(linspace(0,1,5),1-linspace(0,1,5));

%Plotting the axes

plot(zeros(1,2),linspace(-0.5,1.5,2));

plot(linspace(-0.5,1,2),zeros(1,2));

%% Conjugate curve

curve = spline(ac\_raff,tce\_e);

plot(linspace(0.3,0.8,5),ppval(curve,linspace(0.3,0.8,5)));

%% Tie line intersection with RS

% From the plot we can infer topmost tie line gives farthest S

line1 = polyfit([tce\_r(1),tce\_e(1)],[ac\_raff(1),ac\_ext(1)],1);

line3 = polyfit([tce\_r(3),tce\_e(3)],[ac\_raff(3),ac\_ext(3)],1);

RS = @(x)(yRN + (x-xRN)\*(-yRN/(1-xRN)));

int1 = @(x)(RS(x)-polyval(line1,x));

xint1 = fsolve(int1,0);

int2 = @(x)(RS(x)-polyval(line3,x));

xint2 = fsolve(int2,0);

plot([xint1,tce\_r(1),tce\_e(1)],[polyval(line1,xint1),ac\_raff(1),ac\_ext(1)]);

plot([xint2,tce\_r(3),tce\_e(3)],[polyval(line3,xint2),ac\_raff(3),ac\_ext(3)]);

plot([xint1,1],[polyval(line1,xint1),0]);

%% Connecting F and delmin

xint = xint1;

yint = polyval(line1,xint1);

Fdel = polyfit([xint,0],[yint,yF],1);

int = @(x)(polyval(Fdel,x)-ppval(e\_p,x));

xE1 = fsolve(int,0);

yE1 = ppval(e\_p,xE1);

%hold off;

%figure();

hold on; grid on; grid minor;

plot([xint1,1],[polyval(line1,xint1),0]);

plot([xint,xE1],polyval(Fdel,[xint,xE1]),'-.');

xlabel('TCE MASS FRACTION');

ylabel('ACETONE MASS FRACTION');

hold off;

%% Getting Smin

%Find intersection of EF and RS

mER = (yE1-yRN)/(xE1-xRN);

xM = (yF-yRN+mER\*xRN)/(mER+yF);

yM = -yF\*xM+yF;

Smin = F\*(yF-yM)/(yM);

%% Stages

figure();

plot(r\_tce,r\_a,e\_tce,e\_a);

hold on;

%Conjugate Curve

plot(linspace(0.3,0.8,5),ppval(curve,linspace(0.3,0.8,5)));

%Plotting the axes

plot(zeros(1,2),linspace(-0.5,1,2));

plot(linspace(-0.5,1,2),zeros(1,2));

S = 1.5\*Smin;

yMnew = F\*yF/(F+S);

RM = polyfit([xRN,S/(F+S)],[yRN,yMnew],1);

fun1 = @(x)(polyval(RM,x)-spline(e\_tce,e\_a,x));

xE1 = fsolve(fun1,1);

yE1 = polyval(RM,xE1);

plot([0,1],polyval(RM,[0,1]));

FE = polyfit([xE1,0],[yE1,yF],1);

RS = polyfit([xRN,1],[yRN,0],1);

fun = @(x)(polyval(FE,x)-polyval(RS,x));

delx = fsolve(fun,-1.48);

dely = polyval(FE,delx);

plot(delx,dely,'x');

x = [-2,1];

plot(x,polyval(FE,x),x,polyval(RS,x));

% Stepping process

% Conjugate curve

cc = spline(tce\_e,ac\_raff);

i = 1;

yp = yE1;

xp = xE1;

EC = spline(e\_tce,e\_a);

ycoords= zeros(1,4);

xcoords= zeros(1,4);

ycoords(1)= yp;

xcoords(1) = xp;

ycoords2= zeros(1,4);

xcoords2= zeros(1,4);

while yp >= yRN

yp = ppval(cc,xp);

xp = ppval(r\_p,yp);

ycoords2(i) = yp;

xcoords2(i) = xp;

OL = polyfit([xp,delx],[yp,dely],1);

f = @(x)(ppval(EC,x)-polyval(OL,x));

xp = fsolve(f,0.5);

yp = polyval(OL,xp);

ycoords(i+1)= yp;

xcoords(i+1) = xp;

i = i + 1;

if i > 25

break;

end

end

plot(linspace(0,1,5),1-linspace(0,1,5));

plot(xcoords,ycoords,'o',xcoords2,ycoords2,'x');

xlabel('TCE MASS FRACTION');

ylabel('ACETONE MASS FRACTION');

%% Mass

E1 = (F\*(yF-yRN)-S\*yRN)/(yE1-yRN);

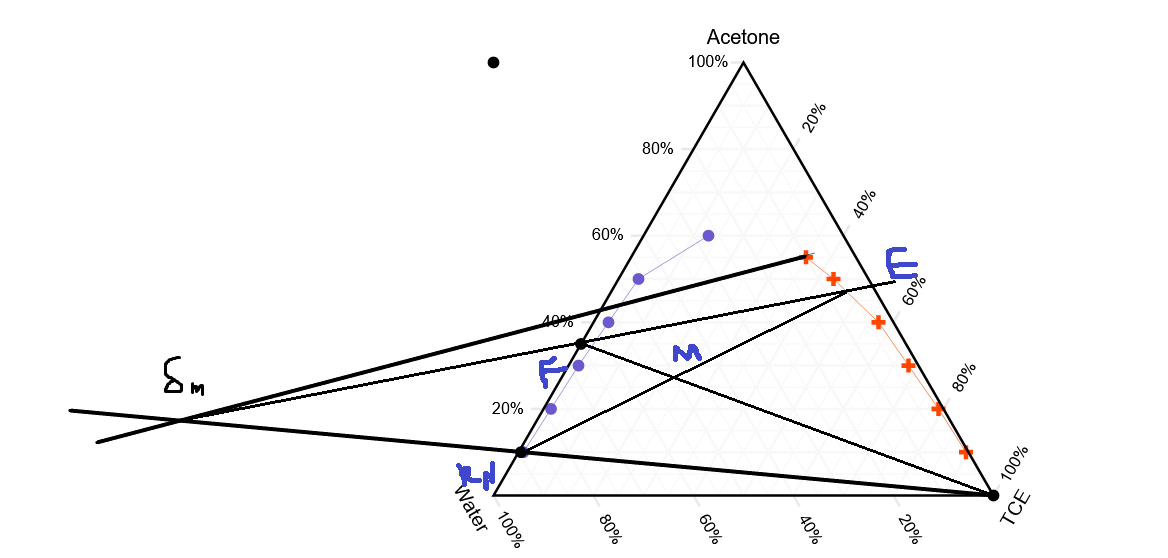
E = ((yE1-ycoords2(1:2))\*E1-(yF-ycoords2(1:2))\*F)./(ycoords(2:3)-ycoords2(1:2));

R = ((yE1-ycoords(2:3))\*E1-(yF-ycoords(2:3))\*F)./(ycoords(2:3)-ycoords2(1:2));

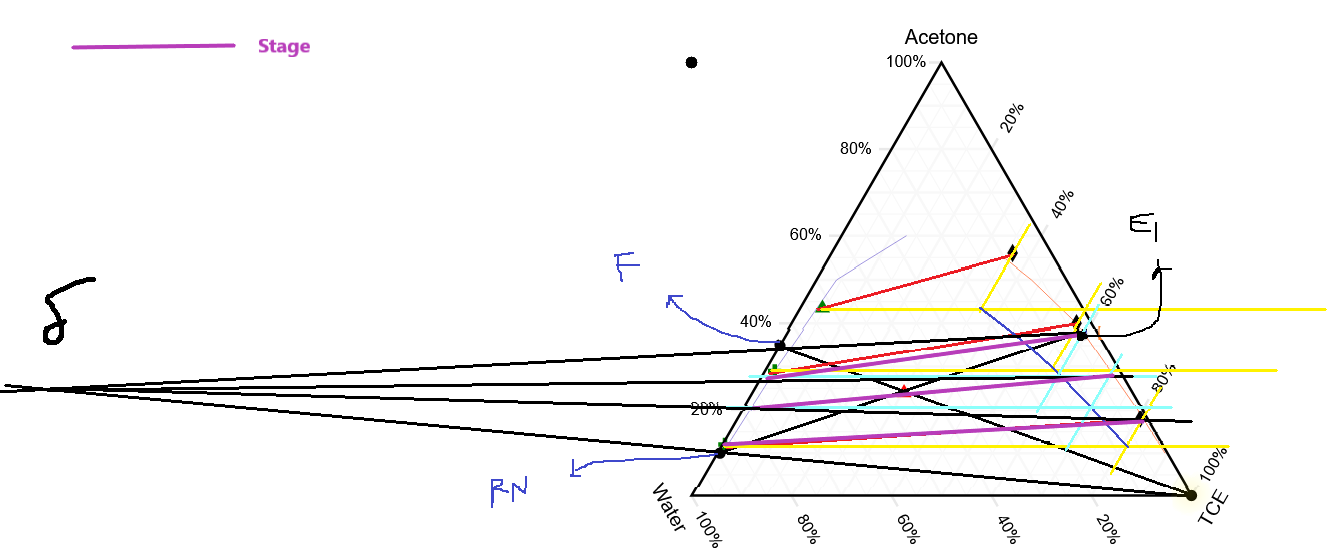
# Question 1- EQUILATERAL TRIANGLE METHOD

Diagram source: www.ternaryplot.com

1. Smin determination **(Graph Lines not visible when I upload it in word)**



1. And c) Stages and composition determination**(Graph Lines not visible when I upload it in word)**

**Legend for each of these graphs is in the handwritten part**

## MATLAB Code for solving the Mass balance equations

xcoords = [0.60,0.70,0.825];

ycoords = [0.375 0.29 0.175];

ycoords2 = [0.28,0.20,0.115];

xcoords2 = 1-ycoords2-[0.70 0.80 0.875];

Snew = 1.5\*354.5455;

E1 = (F\*(yF-yRN)-Snew\*yRN)/(yE1-yRN);

E = ((yE1-ycoords2(1:2))\*E1-(yF-ycoords2(1:2))\*F)./(ycoords(2:3)-ycoords2(1:2));

R = ((yE1-ycoords(2:3))\*E1-(yF-ycoords(2:3))\*F)./(ycoords(2:3)-ycoords2(1:2));

RNew = F + Snew- E1;