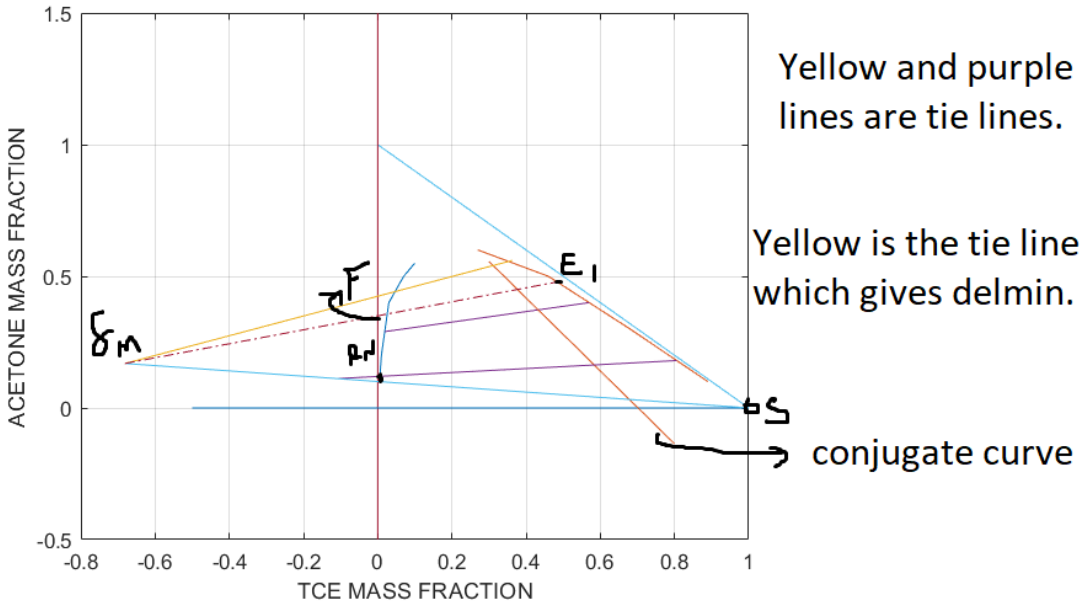
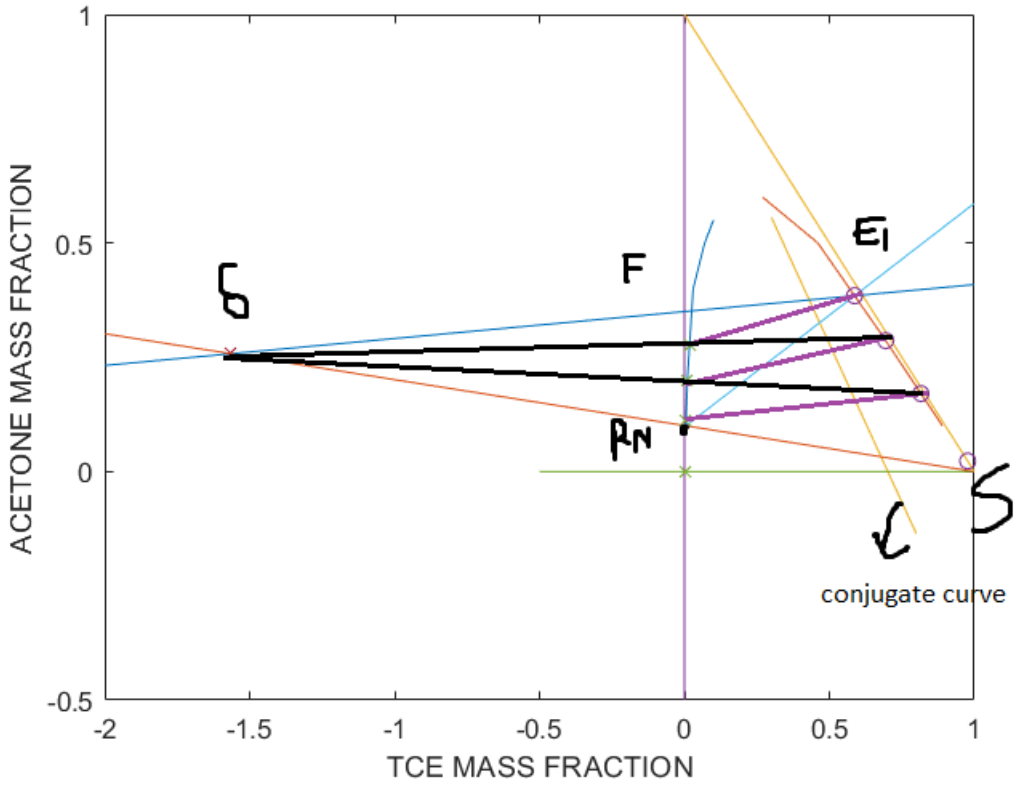
Question 1 – RIGHT TRIANGLE METHOD

a) Delta point for minimum solvent flow rate



b) 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);
```

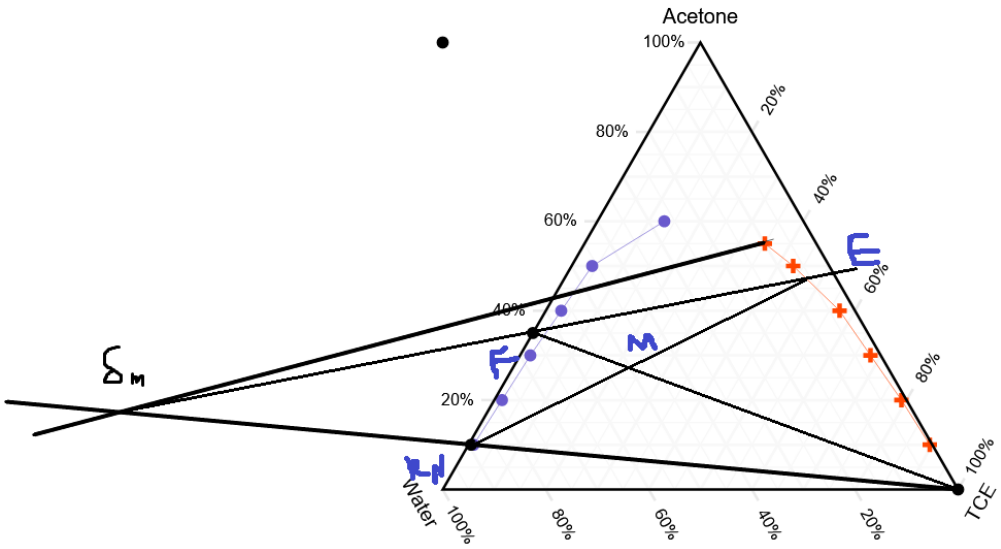
```
yE1 = ppval(e_p,xE1);
%hold off;
%figure();
hold on; grid on; grid minor;
plot([xint1,1],[polyval(line1,xint1),0]);
plot([xint,xE1],polyval(Fdel,[xint,xE1]),'-.');
xlabel('TCE MASS FRACTION');
ylabel('ACETONE MASS FRACTION');
hold off;
%% Getting Smin
%Find intersection of EF and RS
mER = (yE1-yRN)/(xE1-xRN);
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));
```

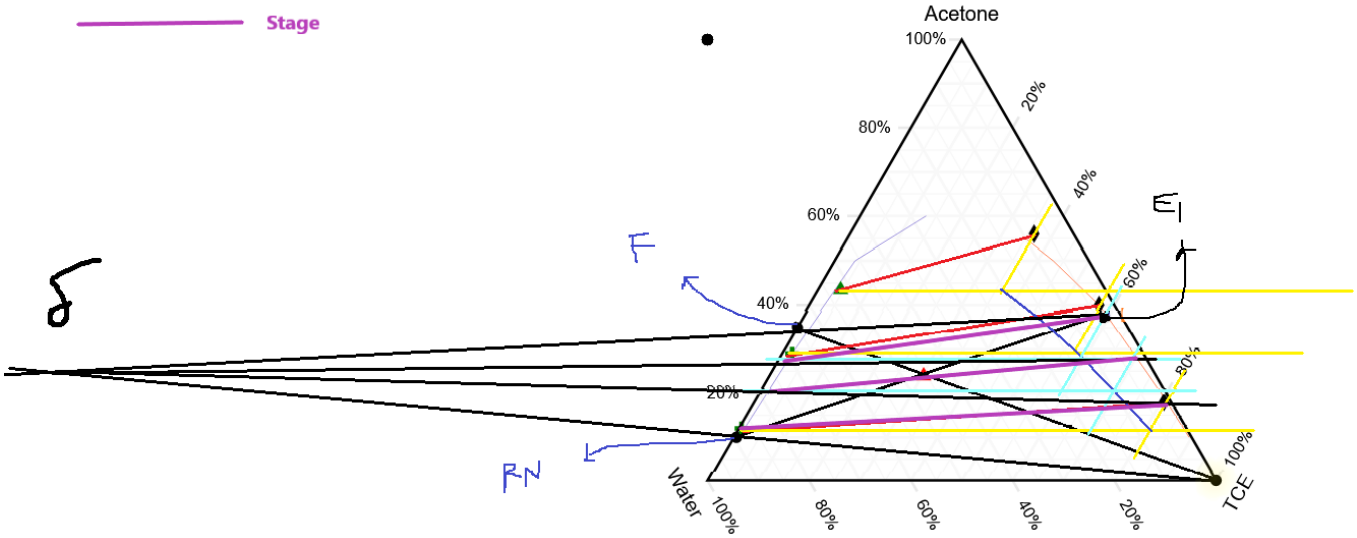
Question 1- EQUILATERAL TRIANGLE METHOD

Diagram source: www.ternaryplot.com

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



b) 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;
```