BY: S. VISHAL
CH18B020

AMT- TUTORIAL 9

(1) → The given equilibrium data is plot on a right angled triangle diagram.

x- axis : concentration of solvent - ~~water~~ chlorobenzene.

y- axis : concentration of solute - pyridine

* point $S \equiv (1, 0)$

point $F \equiv (0.5, 0)$

~~the~~ R_N is the point on extract curve with y- coordinate as 0.02

$\Rightarrow R_N$ is approximately at $(0.\cancel{0}1, 0.02)$

S and F are marked on graph.

* $M = F + S$ —— ①

$x_M M = x_F F + x_S S$ —— ②

$$(x_A)_M = \frac{F(x_A)_F}{F+S} = \frac{0.5 \times 0.5}{}$$

$$\Rightarrow (x_A)_{M1} = 0.25$$

$$(x_S)_{M1} = \frac{\cancel{F}S}{F+S} = 0.5$$ (will remain constant for all midpoints, since at every stage mass of solvent = mass of feed entering)

(∵ S = F = 2000 kg)

* Tie line passing through M1 is drawn with the help of conjugate curve in MATLAB.

(only the tie line is shown in the picture attached and not the procedure used for the sake of clarity)

↳ The conjugate curve is fit using $((x_S)_{extract}, (x_A)_{raffinate})$ points

* Using the ~~tie~~ tie line, we determine that

$x_{R_1} = 0.004$, $y_{R_1} = 0.2153$; $x_{E_1} = 0.7157$, $y_{E_1} = 0.2651$

x denotes Solvent concentration (mole fraction)

y denotes acetone concentration

* Now the Raffinate obtained will be the feed for next stage.

Mass balances: $S + R_1 = F + S = 2F$ — ①

$E_1 y_E + R_1 = F y_F$ — ②

$$\Rightarrow E_1 = \frac{F(2y_{R_1} - y_F)}{y_{R_1} - y_{E1}}$$

$$R_1 = \frac{F(2y_{E_1} - y_F)}{y_{E_1} - y_{R1}} = 1212.785 \text{ kg}$$

(∵ we need only Raffinate mass).

* Solvent entering will have the mass same as R_1 (given)

$\Rightarrow S_2 = 1212.785$ kg

* $y_{M_2} = \frac{R_1 y_{R1}}{R_1 + S_2} = \frac{y_{R1}}{2} = 0.108$

$x_M = 0.5$ (same as previous stage)

* Once again we repeat the process of drawing a tie line & obtain

$x_{R_2} = 0.0018$, $y_{R_2} = 0.031$

$x_{E_2} = 0.8526$, $y_{E_2} = 0.139$.

The tie line was again plot ~~using~~ in MATLAB using Conjugate curve method.

* Writing a similar mass balance we obtain,

$$R_2 = \frac{R_1(2y_{E_2} - y_{R_1})}{y_{E_2} - y_{R_2}} = 1005.195 \text{ kg}$$

* We repeat this procedure for the next stage.

We obtain

$S_3 = R_2 = 1005.195$ kg.

$y_{M_3} = 0.0316$

$x_{R_3} = 0.0013$, $y_{R_3} = 0.0258$

$x_{E_3} = 0.96$, $y_{E_3} = 0.037$

* We see that y_{R_3} = also 2.58 %.

Close to a 2% which is the required raffinate composition.

If we use another stage, the concentration goes will below 2%, so that is not done.

* Number of stages = 3

* Mass of solvent used = 2000 + 1212.785 + 1005.195

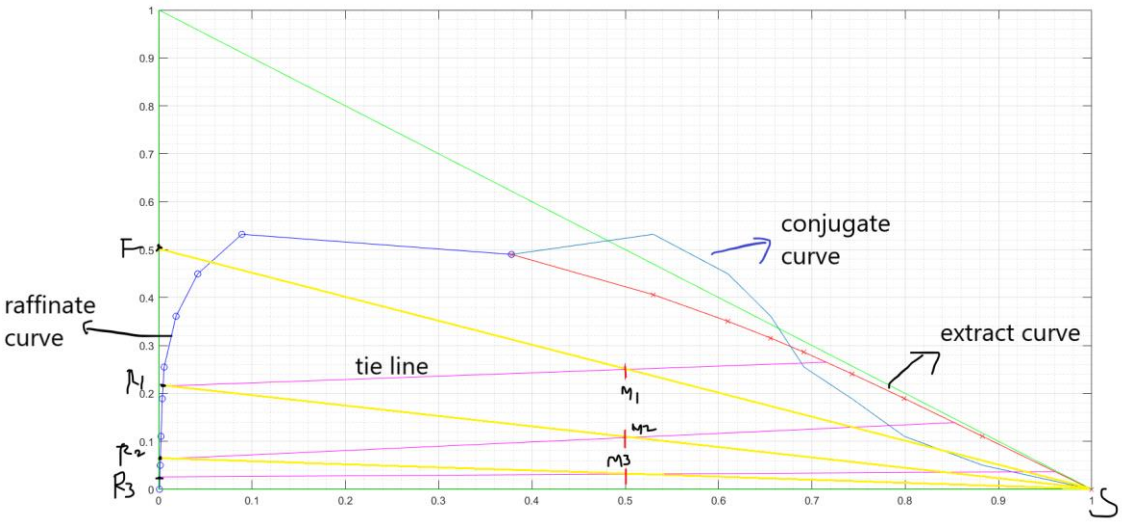
= 4217.98 kg

Question 1

Stepping process

X coordinate- composition of **chlorobenzene**

Y coordinate- composition of **Pyridine**



Code

```
clear; close all;
%% Equilibrium Data
a_e = [0,11.05,18.95,24.1,28.6,31.55,35.05,40.6,49]*0.01;
c_e = 0.01*[0.05 0.67 1.15 1.62 2.25 2.87 3.95 6.4 13.2];
s_e = 0.01*[99.95 88.28 79.9 74.28 69.15 65.58 61 53 37.8];
a_r = 0.01*[0 5.02 11.05 18.9 25.5 36.1 44.95 53.2 49];
s_r = 0.01*[0.08 0.16 0.24 0.38 0.58 1.85 4.18 8.9 37.8];
c_r = 0.01*[99.92 94.82 88.71 80.72 73.92 62.05 50.87 37.9 13.2];
plot(s_e,a_e,'red',s_r,a_r,'blue',s_e,a_e,'rx',s_r,a_r,'bo');
hold on;
plot([0,1],[0,0],'green',[0,0],[0,1],'green',[1,0],[0,1],'green');
% %Plotting the tie lines
% for i = 1:9
%     plot([s_r(i),s_e(i)],[a_r(i),a_e(i)],'m');
% end
grid on;grid minor;
rx = spline(a_r(1:8),s_r(1:8));
ey = spline(s_e(1:8),a_e(1:8));
%% conjugate curve
cc_y = spline(s_e,a_r);
plot(s_e,a_r);
cc_x = spline(a_r(1:8),s_e(1:8));
%% Feed data
mF1 = 2000;
yF1 = 0.5;
yS = 0;
```

```
S1 = mF1;
%% Mixing point 1
yM = yF1/2;
xM = 0.5;%Always
plot(xM,yM,'+');
sol = fsolve(@(x) cctie(x,xM,yM,rx,ey,cc_y),0.6);
[xR1,yR1,xE1,yE1] = endpts(sol,rx,ey,cc_y);
plot([xR1,xE1],[yR1,yE1],'m',0,yF1,'x');
%% Stage 2
yM2  = yR1/2;
plot(xM,yM2,'r+');
sol2 = fsolve(@(x) cctie(x,xM,yM2,rx,ey,cc_y),0.7);
[xR2,yR2,xE2,yE2] = endpts(sol2,rx,ey,cc_y);
plot([xR2,xE2],[yR2,yE2],'m');
%% Stage 3
yM3  = yR2/2;
plot(xM,yM3,'r+');
sol2 = fsolve(@(x) cctie(x,xM,yM3,rx,ey,cc_y),0.7);
[xR3,yR3,xE3,yE3] = endpts(sol2,rx,ey,cc_y);
plot([xR3,xE3],[yR3,yE3],'m');
% %% Stage 4
% yM3  = yR3/2;
% plot(xM,yM3,'r+');
% sol2 = fsolve(@(x) cctie(x,xM,yM3,rx,ey,cc_y),0.7);
% [xR3,yR3,xE3,yE3] = endpts(sol2,rx,ey,cc_y);
% plot([xR3,xE3],[yR3,yE3],'m');
%% mass balance
raff_m = zeros(1,3)+mF1;
R = mF1;
y_E = [yE1 yE2 yE3];
y_R = [yF1 yR1 yR2 yR3];
for i = 1:2
    R = R*(2*y_E(i)-y_R(i))/(y_E(i)-y_R(i+1));
    raff_m(i+1) = R;
end
%% Function to get end pts of tie line given pt on cc
function [xR,yR,xE,yE] = endpts(x,rx,ey,cc_y)
    xE = x;
    yR = ppval(cc_y,x);
    xR = ppval(rx,yR);
    yE = ppval(ey,xE);
end
%% function to return slope difference given point on cc and mid pt
function val = cctie(x,xM,yM,rx,ey,cc_y)
    xE = x;
    yR = ppval(cc_y,x);
    xR = ppval(rx,yR);
    yE = ppval(ey,xE);
    val = (yE-yR)*(xE-xM) - (yE-yM)*(xE-xR);
end
```

(2) Again a Right-angled triangle diagram is used.

Ether - carrier, Alcohol - Solute, Water - Solvent
(y-axis) (x-axis)

a) * The given equilibrium data and the tie lines are plot on a right angled triangle ternary diagram.

$F \equiv (0.05, 0.5)$; $S \equiv (1, 0)$

$R_N \equiv$ point on raffinate curve with y-coordinate 0.05

$E_1 \equiv$ point on extract curve with y-coordinate 0.2.

* $R_N S$ and FE_1 are joined respectively. The intersection of $R_N S$ and FE_1 gives δ (the operating point)

* Any line through δ becomes an operating line. The y coordinate of intersection of such a line with the extract curve and the y coordinate of its intersection with raffinate curve gives y and x coordinates of a point on the operating line in the x-y diagram.

[x - represents raffinate composition
y - " extract " in such a diagram]

* multiple lines (10, with slopes equally spaced b/w slopes of FE_1 & $R_N S$) are drawn and the corresponding xy points are obtained. The xy points are numerically solved for in MATLAB.

* The equilibrium curve is also done by projecting it on xy diagram.

* Y coordinates of point at the end of a tie line on raffinate curve & Y coordinate of the point at the end of the same tie line on extract curve form an (X, Y) point on the equilibrium curve.

* After the equilibrium and operating curves are plot, the stepping process is done.

* We start from the raffinate end (x coordinate is 5% on the operating line) and step till the extract end

* Number of stages obtained = 2.

b) Obtaining extract limit

* Conjugate curve is plot

* Utilising the conjugate curve a tie line passing through F is plot.

* This gives the extract limit; since tie line coincides with operating line it is theoritically requires ∞ stages.

* Using the tie line (coinciding with OL), E, point is obtained on the extract curve

~~* y coordinate~~

* y coordinate of E_1 gives the maximum limit of alcohol content in breath
* From the graph we read the value as 0.25 (approx)

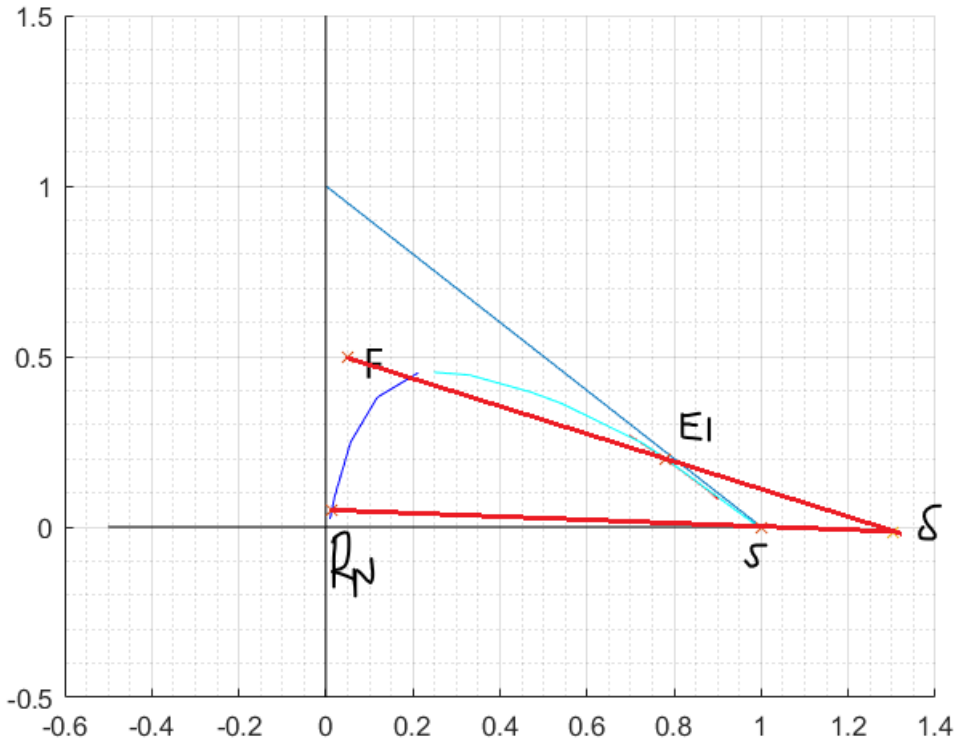
∴ Breath limit ≤ 0.25

Question 2

Determining Delta point.

X coordinate- composition of **water**

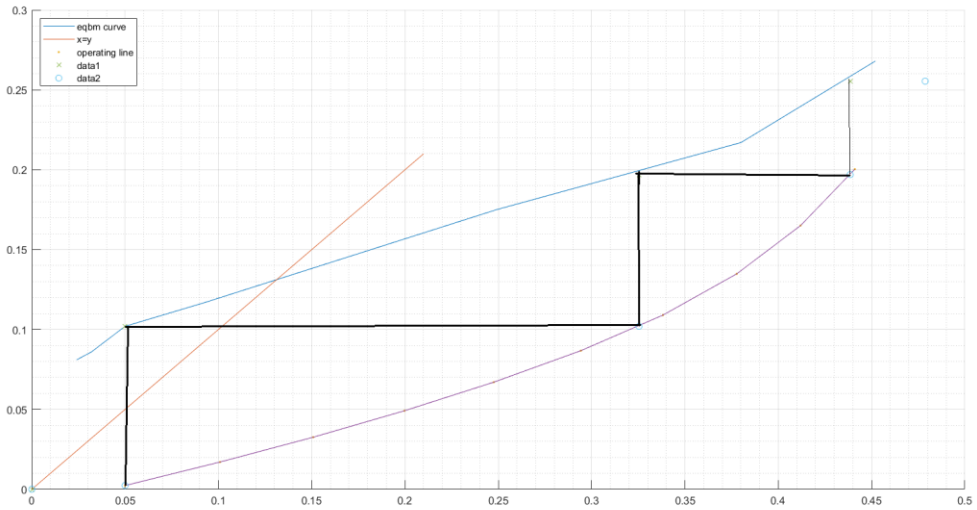
Y coordinate- composition of **alcohol**



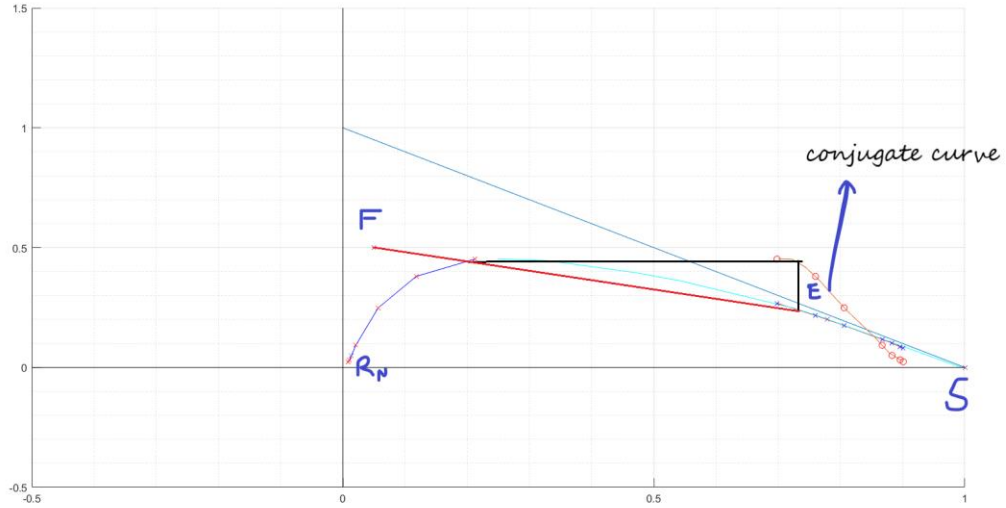
Drawing stages after projecting the right angled diagram to xy.

X coordinate- composition of alcohol in raffinate

Y coordinate- composition of alcohol in extract



Determining the maximum extract limit



Code

```
% Ether-Carrier, Alcohol-A, Water-Solvent
clear; close all;
%% Equilibrium data
ar = 0.01*[2.4 3.2 5 9.3 24.9 38 45.2];
sr = 0.01*[0.9 1.1 1.4 2.1 5.7 11.8 21.2];
raff = spline(ar,sr);
r = spline(sr,ar);
ae = 0.01*[8.1 8.6 10.2 11.7 17.5 21.7 26.8];
se = 0.01*[90.1 89.6 88.3 86.7 80.6 76 69.8];
```

```
ext = spline(ae,se);
e = spline(se,ae);
%additional points
a_add = 0.01*[45.37 44.55 39.57 36.23 24.74 21.33 0 0];
s_add = 0.01*[24.93 33 47.01 54.11 72.52 76.61 99.4 0.5];
hold on;
plot(sr,ar,'b',se,ae,'r',s_add,a_add,'c');
%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),'k');
plot(linspace(-0.5,1,2),zeros(1,2),'k');
plot(sr,ar,'rx',se,ae,'bx');
grid on; grid minor;
%% conjugate curve
curve = spline(se,ar);
plot(linspace(0.7,0.9,10),ppval(curve,linspace(0.7,0.9,10)),se,ar,'r
o');
%% Given Feed Data
yF =0.5;
xF = 0.05;
yRN = 0.05;
xRN = ppval(raff,yRN);
yE1 = 0.2;
xE1 = ppval(ext,yE1);
plot([xF,xE1,xRN,1],[yF,yE1,yRN,0],'x');
%% Finding Del point
RS = polyfit([xRN,1],[yRN,0],1);
FE = polyfit([xE1,xF],[yE1,yF],1);
fun = @(x)(polyval(RS,x)-polyval(FE,x));
delx = fsolve(fun,0);
dely = polyval(RS,delx);
%% Drawing lines from the del point
%no. of lines
n = 10;
m = linspace(FE(1),RS(1),n);
line = @(y)(delx + 1./m.*(y-dely));
raff_int = @(y)(line(y)-ppval(raff,y));
% ext_int = @(y)(line(y)-ppval(ext,y));
xA = fsolve(raff_int,zeros(1,n)+1);
xB = line(xA);
line1 = @(x)(m.*(x-delx)+dely);
% raff_int = @(x)(line(x)-ppval(r,x));
[p,s] = polyfit(se,ae,4);
ext_int = @(x)(line1(x)-polyval(p,x));
yB = fsolve(ext_int,zeros(1,n)+1);
yA = line1(yB);
plot([xF,xE1,xRN,1],[yF,yE1,yRN,0],'x');
% % Resizing since only 7 points were properly obtained
% yB = yB(1:7);
% yA = yA(1:7);
% xA = xA(1:7);
% xB = xB(1:7);
figure();
hold on;
plot(sr,ar,'b',se,ae,'r',s_add,a_add,'c');
```

```
plot(delx,dely,'x');
%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),'k');
plot(linspace(-0.5,1,2),zeros(1,2),'k');
grid on; grid minor;
plot(xB,xA,'rx',yB,yA,'bx');%,linspace(0.7,1,10),polyval(p,linspace(
0.7,1,10)),'k');
%% Equilibrium projection
xeqbm = ar;
yeqbm = ae;
figure();
hold on;
plot(xeqbm,yeqbm,[0.21,0],[0.21,0],xA,yA,'.',xA,yA);
grid on; grid minor;
legend('eqbm curve','x=y','operating line','Location','northwest');
%% Stepping
i = 1;
xy = spline(yeqbm,xeqbm);
OL = spline(xA,yA);
xcoords = zeros(1,7);
ycoords = zeros(1,7);
xcoords2 = zeros(1,8);
ycoords2 = zeros(1,8);
x0 = 0.05;
y0 = spline(xA,yA,x0);
yx = spline(xeqbm,yeqbm);
LO = spline(yA,xA);
xcoords2(1)= x0;
ycoords2(1) = y0;
while y0 < 0.2
    y0 = ppval(yx,x0);
    xcoords(i) = x0;
    ycoords(i)=y0;
    x0 = ppval(LO,y0);
    xcoords2(i+1) = x0;
    ycoords2(i+1) = y0;
    i = i + 1;
end
plot(xcoords,ycoords,'x',xcoords2,ycoords2,'o');
```