Submit the answers by 11. 05.

9.4 ***Crosscurrent liquid–solid contact)***[3] Experimental 'equilibrium data' on extraction of oil from a meal by using benzene are given below:

Mass fraction of oil (C) in solution	*Mass fractions in the underflow*		
y_C	x_C	x_A	x_B
0	0	0.67	0.33
0.1	0.0336	0.664	0.302
0.2	0.0682	0.660	0.272
0.3	0.1039	0.6541	0.242
0.4	0.1419	0.6451	0.213
0.5	0.1817	0.6366	0.1817
0.6	0.224	0.6268	0.1492
0.7	0.268	0.6172	0.1148

Two thousand kilograms per hour of the meal containing 26 mass% oil is extracted with 2100 kg/h of benzene. The underflow leaving the countercurrent cascade must not contain more than 0.015 mass fraction of oil. The feed benzene has 0.005 mass fraction oil in it. The overflow is essentially solid-free. Calculate the mass fraction in the rich extract leaving the cascade and the number of ideal stages required.

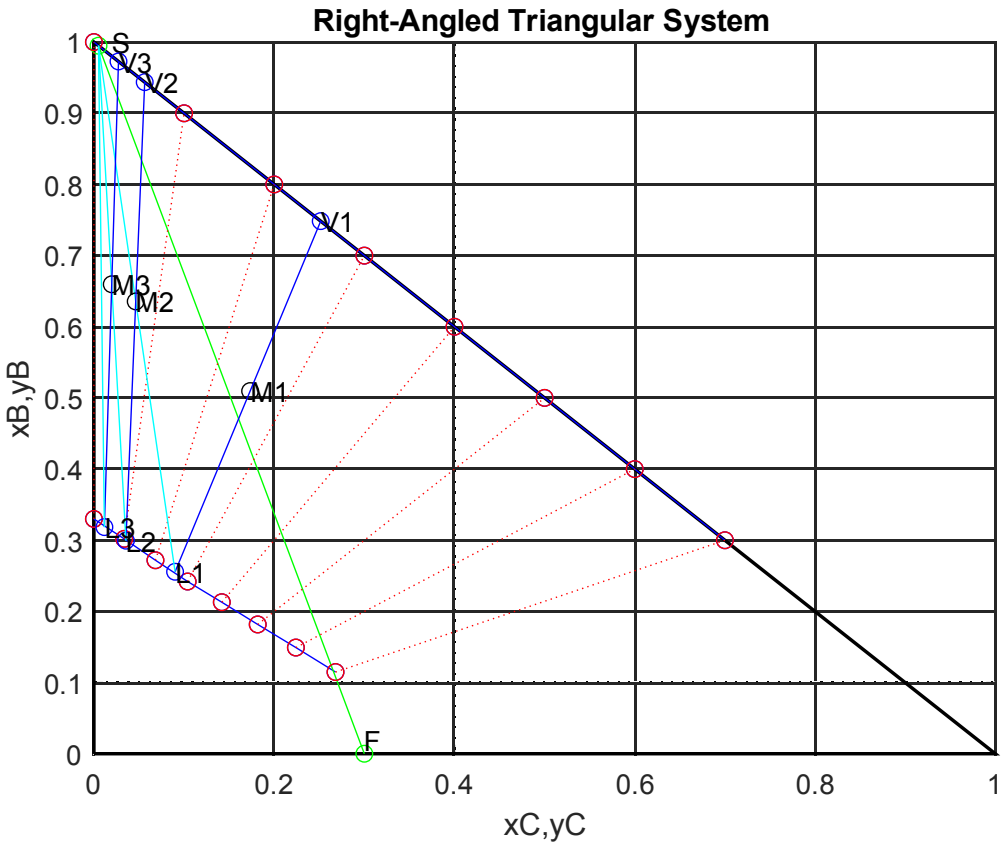
Cross-current operation: [right angled triangle co-ordinate System]

1. Plot the solid liquid eqbm data in using right angled triangle diagram (plot xc vs xb and yc vs yb). [10]
2. Calculate the concentration of the solute in mixture x_m from the material balance and locate the M point. [10]

B. Draw the FM line and the tie lines. Calculate the slopes of the tie_line [15], take xf=0.3. [Will be evaluated in class]

4. Calculate the number of stages for 0.015 mass fraction oil in underflow. Show the stages and also join the underflow point to the solvent point for every stage. Calculate the amount of overflow and underflow and fraction of oil in overflow and underflow in all the stages. [25]

Homework: Propose an algorithm that works well for interpolation of tie line in the region where the feed level is more than 0.4/ algorithm that works well for the tie line going from vertical to horizontal.



Code Hints:

```
xb = [0.33 0.302 0.272 0.242 0.213 0.1817 0.1492 0.1148];
xc = [0 0.0336 0.0682 0.1039 0.1419 0.1817 0.224 0.268];
xa = 1-xb-xc;

yc = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7];
for i = 1: length(yc);

 yb(i) = -yc(i) + 1;

end
ya = 1-yb-yc;


%plotting in right triangular system

figure(1)

plot([0 1 0 0], [0 0 1 0], 'k-', 'linewidth', 1.25); grid on; hold on;

plot(xc, xb, 'bo-'); grid on; hold on;
```

```
plot(yc, yb, 'bo-'); grid on; hold on;
xlabel('xC,yC'); ylabel('xB,yB');
title('Right-Angled Triangular System');
```

```
%plotting FS line
F1 = [0.3 -----];
S1 = [--- ----];
plot(F1, S1, 'go-');grid on; hold on;
text(0.3, 0.02, 'F');
text(0.02, 1, 'S');
```

```
tie_xb = xb;
tie_xc = xc;
tie_yb = yb;
tie_yc = yc;

%plotting the tie lines
for i = 1:length(tie_xb)
    plot([      ],[tie_xb(i) tie_yb(i)], 'ro:');grid on; hold on;
end
tie_slope = ones(1, length(tie_xc));
tie_slope(1)=60;
for i = 2:length(tie_xc)
   tie_slope(i) = ……..
end
```

```
 %fitting to polynomials
p1 = polyfit(xc, xb, 1);
f1 = polyval(p1, xc);
```

```
p2 = polyfit(yc, yb, 1);
f2 = polyval(p2, yc);


%given data

S =   ;

F =.    ;

xcf = 0.3;

xbf = ;

ycs = ;

ybs = ;

xcr_n = 0.015;
```

```
%cross current process


i=1;

xcr=1;

Sum_solvent = 0;

solute_leaving = 0;

while (xcr > xcr_n)

   %plotting the M(i) points
   M =
   Mx =
   My =
   plot(Mx, My, 'ko')
   text(Mx, My, ['M' num2str(i) ''])

   % tie line interpolation
    for k =2:length(tie_xc)
      if Mx<=((tie_yc(k)+tie_xc(k))/2) && Mx>=((tie_yc(k-1)+tie_xc(k-1))/2)
         break;
       end
    end

   if (k==2)
```

```
      slope = tie_slope(k-1) + ((Mx-tie_yc(k-1))/(tie_yc(k)-tie_yc(k-1))) * (tie_slope(k) -
tie_slope(k-1));
    else
     slope = …….
    end



    % calculating the concentrations after each stage
    syms x y

    assume(x > 0 & x < 1)
    assume(y > 0 & y < 1)

    [xcr, xbr] = vpasolve([y ==……., y == ……..], [x,y], [0 1;0 1]);
    [yce, ybe] = vpasolve([y == ……., y ==……..], [x,y], [0 1;0 1]);

    plot([…….], […….],'bo-');grid on; hold on;

    plot([…….], [……..], 'c-');grid on; hold on;

    %calculating the amount of underflow and overflow after each stage
    [r, e] = vpasolve([y == ……, y ==…….], [x,y], [0, Inf; 0, Inf]);

    %printing the results
     fprintf('Amount of underflow after stage_%d =', i);
     disp(r);
     fprintf('Amount of overflow after stage_%d =', i);
     disp(e);
     fprintf('Underflow concentration after stage_%d = ', i);
     disp(xcr);
     fprintf('Overflow concentration after stage_%d = ', i);
     disp(yce);

    %calculating fractional recovery after each stage
    fprintf('Fraction of solute removed in stage_%d =', i);
    disp((yce*e-S*ycs)/(F*xcf));

    Sum_solvent = ……..;
    solute_leaving = ……;

    xbf = …..;
    xcf = …..;
    F= …..;

    text( xcf, xbf,['L' num2str(i) '']);
```

```
    text( yce, ybe,['V' num2str(i) '']);
    i=i+1;


 end

number_of_stages = i-1

%calculating overall fractional recovery
disp('Overall fraction of solute separated from the feed =');
disp((……) / (2000*0.26));
```