LAUTECH Journal of Engineering and Technology 18 (2) 2024: 138-147
10.36108/laujet/4202.81.0231

Construction of ternary diagram for liquid-liquid extraction process using computer-based program

***Sulayman A. A., Araromi D. O., Ayodeji O. S., Giwa S. B., Akinwumi O. D. and Salam K. K.**

Chemical Engineering Department, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria

Article Info

Article history:

Received: May 25, 2024
Revised: June 21, 2024
Accepted: June 27, 2024

Keywords:

Liquid-Liquid Extraction,
Ternary,
simulation,
MATLAB,
ANFIS.

Corresponding Author:
aasulayman@lautech.edu.ng

ABSTRACT

Liquid-liquid extraction (LLE) is a liquid-liquid phase mass transfer process in which solute chemicals are transferred between two immiscible or partially miscible liquids due to the difference in solubility of the solutes in the two liquids. The phase equilibrium relationships are generally inconvenient to handle algebraically, therefore, LLE computational analyses are usually made graphically. Manual construction of a ternary diagram becomes cumbersome, tedious, and time-consuming when used for a process that requires a large number of stages. Therefore, this work aimed to develop a customized computer package for the ternary diagram. The customized computer package for the ternary diagram construction was achieved with the use of a fuzzy inference system toolbox and primitive line function in MATLAB. The computer program was test-run using data from the literature to determine the theoretical number of stages for a given LLE separation process. The result compared well with the existing literature results.

INTRODUCTION

Liquid-liquid extraction (LLE) is an important unit operation process in chemical process industries for the separation of solute chemicals from one solvent to another solvent based on the difference in the solubility of the solute chemicals in two solvents (Zhang & Hu, 2013). The two solvents must be either immiscible or partially miscible with each other (Schmidt and Strube, 2018). Separation is achieved because the solute chemicals are preferentially more soluble in one of the solvents. In LLE processes, the feed -stream that consists of solute chemicals is separated and the original solvent known as diluent is contacted with a stream consisting of a new solvent. The stream that comes out from the contacting unit consists of two phases – one is a solvent-rich phase called the extract phase and the other one is a diluent-rich phase called raffinate. The two phases are at thermodynamic equilibrium (Lamm and Jarboe, 2021). The desired separation may not be achieved in a single contacting stage, therefore multiple contacting stages may be needed to produce a significant separation. The phase equilibrium relationships are generally inconvenient to handle algebraically, therefore LLE computational analyses are usually made graphically. The two common graphical methods used are McCabe Thiele and ternary phase diagram methods. The McCabe Thiele method is best suited for conditions of total immiscibility while the ternary diagram method is most suitable for partial immiscibility solutions.

Ternary diagram is a phase diagram used to describe the equilibrium behaviour of a three-component system. It is used to describe all three components in a liquid-liquid phase mixture of one plot (Magdah and Nory, 2017). Triangle (equilateral) phase diagrams are commonly used and is based on the

property of an equilateral triangle in which the sum of the perpendicular distances from any point within the triangle to the three sides equals the altitude of the triangle (Magdah and Nory, 2017). The altitude represents 100% composition and the distances to the three sides represent the percentages or fractions of the three components. Each apex of the triangle represents a pure component, and each side represents the binary system.

Considering a ternary LLE system comprising a binary mixture and an extracting solvent, the ternary phase diagram helps to show the phase composition and calculate equilibrium data for the system. A ternary phase diagram has the shape of a triangular prism with a triangle as a foundation which is also called a composition triangle as shown in Figure 1. The point present on the edge opposite a corner of the triangle known as a vertex represents a two-component system or binary system and a point present within the internal area of the triangle represents a three-component system or ternary system.

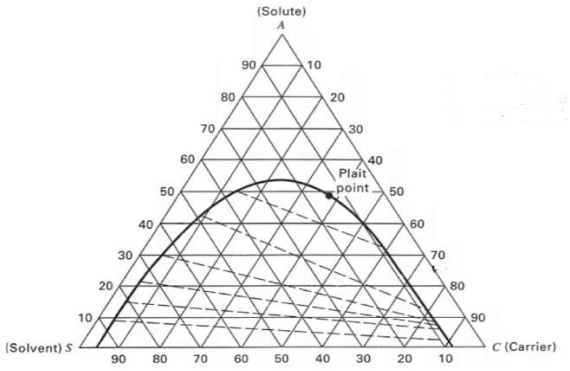

Figure 1: A Ternary phase diagram showing LLE process data joined together by a binodal curve.

For a ternary system consisting of solute (A), solvent (S), and carrier (C) the ternary phase diagram displays a lot of information on the phase behavior of the mixture of the three components A, C, S. However, it also contains considerable information on the phase behaviour of the three binary mixtures, A–S, S–C and A–C that is implied by examining the edges of the diagram and the intersecting regions of phase equilibria (Smith, Peters, and Inomata, 2013).

To design and optimize the LLE, liquid-liquid equilibrium data provided by ternary phase diagrams involving a solubility curve, and tie-lines as well as knowledge of phase equilibrium is essential in modern extraction process design and control methods. The computer-aided method using writing computer programs (codes) is fast becoming a very important method considered in the construction of ternary diagrams due to the less time expended to visually represent and analyze LLE process data. Specific computer programs or codes for short are written to communicate with an automated system to take in experimental data as input, process this given input and then return a visual representation of this data on a ternary graph as an output.

Among the computer-aided software that can be used for writing program codes for a ternary diagram are JAVA, MATLAB, and Python simulation software. Other methods used towards the achievement of time efficient means of constructing the ternary diagrams through computer visualization are using a few various programming languages from the numerous ones available some of which include MOVIE.BYU as done by (Kong, 1997), ProSim, and Open Inventor written in the C++ language. However, some of these programs are flawed in the readability of their source codes, the friendly user interface of these programs as well as codes not easy to understand by others who are not the writers which causes setbacks when minor debugging needs to be carried out. Harper (2015) constructed a ternary phase diagram using a python simulation package for the representation of geological data such as heat maps, grid maps and scatter plots. This work developed a computer

package for the ternary diagram construction with the use of a fuzzy inference system toolbox and primitive line function in MATLAB.

METHODOLOGY

Liquid-liquid extractor Design

The solvent extraction process employed in this work is the counter-current method because it allows the complete removal of the solute and re-usage of solvent, hence, less solvent is needed. The schematic representation of a multistage extraction with counter-current flow is shown in Figure 2.

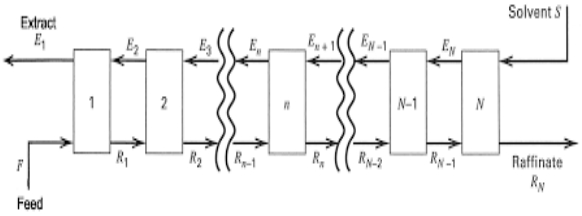

Figure 2: Multistage extraction with countercurrent flow.

The material balance for the LLE process is as follows:

Total material of each stage around the plant is:

Stage 1:

$$F + E_2 = E_1 + R_1 \quad (1)$$

Stage 2:

$$E_2 + R_2 = E_3 + R_1 \quad (2)$$

Nth stage:

$$E_n + R_n = E_{n+1} + R_1 \quad (3)$$

where F= Feed; E_1 = Final Extract; R_n = Final raffinate, n= number of stages.

The equations derived from the total material balance equations in Equations (1) – (3) is re-written as Equations (4) – (6)

$$F - E_2 = R_1 - E_2 \quad (4)$$

$$R_n - E_n = R_n - E_{n+1} \quad (5)$$

$$R_n - S \quad (6)$$

where R_n = Final raffinate, S = solvent; P = The difference between flows and also, E_n and R_n are in equilibrium leaving each stage (Dunn, 2013).

Principle of Operation

The operation of the algorithm designed is based on the plotting of the ternary phase diagram as illustrated in Figure 1 for an LLE process given the composition data of the components. The equilateral triangle is drawn with each of its three sides (vertex) divided into ten equal parts. Each apex is used to represent a 100 per cent composition of each component of the ternary system and each base opposite an apex known as a vertex represents a binary mixture of two components, i.e. at any point on the baseline of the triangle only two components are present. On the inside of the triangle, each point represents a mixture of the three components and the composition at any point sums up to a mole fraction of 1.0 or 100 per cent composition depending on the choice of representation.

The composition data is plotted on the triangle and the binodal curve dividing the system into two regions that is the homogeneous phase region and the heterogeneous phase region further separating the two-liquid phase region into the raffinate section and the extract section.

Development of customized computer package using MATLAB

The construction of a customized computer package for the ternary diagram construction requires the design of algorithms to achieve the graphical representation of the LLE process. The procedure used in the construction of customized ternary diagram package using MATLAB simulation software is shown in Figure 3.

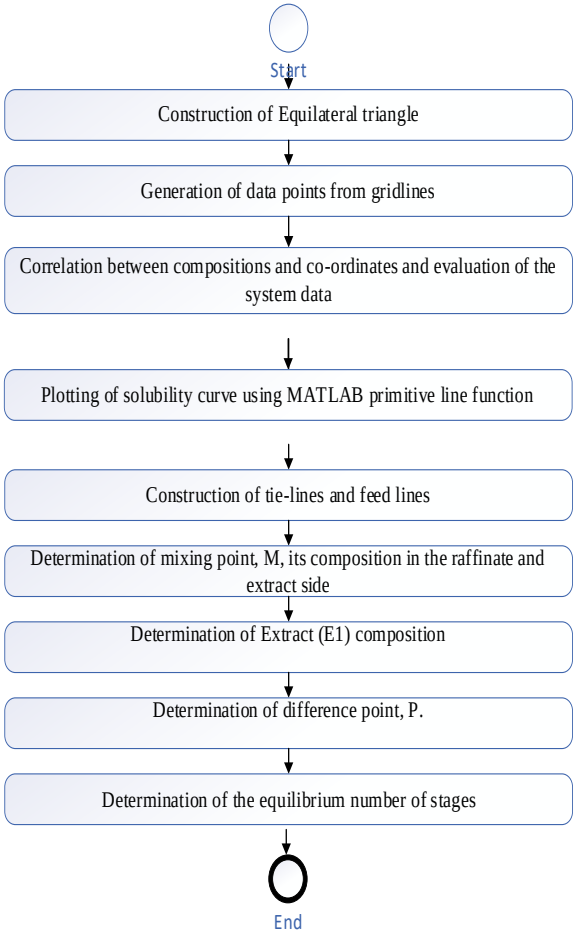

Figure 3: Procedure for the construction of computer program for a ternary diagram for LLE

A triangle with three equal sides and angles was constructed using MATLAB primitive line function to represent the ternary system, it serves as the graph upon which the solubility curve of the ternary system and other information were represented. In this work, the left side of the triangle represents the acetic acid and water binary system, the apex represents pure water composition, the right side represents the acetic acid and isopropyl-ether binary system while the bottom represents Isopropyl-ether and acetic acid binary system. Each side of the triangle was divided into ten equal parts using gridlines to represent the one per cent composition of each component.

The constructed gridlines were used to derive the relationship between the x-coordinate, y-coordinates, and composition of each component. Correlation was determined for each gridline data point. The correlation derived for the data points was used to train the system to make predictions of x and y coordinates for given component compositions using the Artificial Neural Fuzzy Logic Inference System (ANFIS). The LLE data considered for this work was the Equilibrium data of a ternary system consisting of Water, Acetic acid and Isopropyl ether (ISE) at 293 K and 1 atm obtained from Abdelbasit and Salih (2017) as shown in Table 1.

Table 1: Equilibrium data of the ternary system

Water-rich Phase			**Solvent rich phase**		
Water	**Acetic Acid**	**Isopropyl-ether**	**Water**	**Acetic Acid**	**Isopropyl-ether**
98.1	0.07	0.12	0.05	0.02	9.93
97.1	1.4	1.5	0.7	0.4	98.9
95.5	2.9	1.6	0.8	0.8	98.4
91.7	6.4	1.9	1	1.9	97.1
84.4	13.3	2.3	1.9	4.8	93.3
71.1	25.5	3.4	3.9	11.4	84.7
58.9	36.7	4.4	6.9	21.6	71.5
45.1	44.3	10.6	10.8	31.1	58.1
37.1	46.4	16.5	15.4	36.2	48.7

Tie-lines drawn were used to define the equilibrium curve and to divide the solubility curve into two regions known as the raffinate region and the extract region. The generated points in the triangle are joined from the raffinate side and extract side using the primitive line function. The feed line was generated by drawing a line to connect the two feed streams in the extraction process i.e. pure solvent

composition (Isopropyl-ether=100) and the binary mixture composition point (Acetic-Acid=30% and Water=70%).

The mixing point M is a point in the ternary system which represents a combined composition of the feed F (binary mixture of acetic acid and water) and the entering solvent S (Isopropyl-ether). The average method was used to determine the composition of the raffinate and extract side for the mixing point. This was done by finding the average between the lower and upper tie-line for which the mixing point falls as well as the corresponding slope on both sides. A simultaneous equation was then developed and solved to determine the points and a line was drawn to connect these points. The composition of the mixing point was determined by a series of material balances i.e. Total material balance and component material balance for each of the three components.

Taking a total material balance of the system:

F = feed consisting of Acetic acid and water = 8000 kg/h

S = Solvent Isopropyl-Ether (ISE) = 20000 kg/h = Final extract

M = Amount of mixing point

Total material balance:

$$M = F + S \quad (7)$$

Substituting the values of F and S in equation 7,

M = 8000 + 20,000 = 28,000kg/h

The mixing point M is located on the line joining feed and solvent compositions.

Component material balance:

For component A:

$$M \times X_{AM} = \left(\mathrm{F} \times (X_{AF})\right) + \left(\mathrm{S} \times (X_{AS})\right) \quad (8)$$

$$X_{AM} = \left(\mathrm{F} \times (X_{AF})\right) + \left(\mathrm{S} \times (X_{AS})\right)/M$$

Where M = Mixing point

F = Feed composition

S = Solvent composition

X_{AF} = weight fraction of A in feed = 0.3

X_{AS} = weight fraction of A in solvent

X_{AM} = weight fraction of A at mixing point

For component B:

$$M \times X_{\mathrm{B}F} = \left(\mathrm{F} \times (X_{\mathrm{B}F})\right) + \left(\mathrm{S} \times (X_{\mathrm{B}S})\right) \quad (9)$$

$$X_{\mathrm{B}M} = \left(\mathrm{F} \times (X_{\mathrm{B}F})\right) + \left(\mathrm{S} \times (X_{\mathrm{B}S})\right)/M$$

$X_{\mathrm{B}F}$ = weight fraction of B in feed = 0.7

$X_{\mathrm{B}S}$ = weight fraction of B in solvent

$X_{\mathrm{B}M}$ = weight fraction of B at mixing point

For component C:

$$M \times X_{\mathrm{C}F} = \left(\mathrm{F} \times (X_{\mathrm{C}F})\right) + \left(\mathrm{S} \times (X_{\mathrm{C}S})\right) \quad (10)$$

$$X_{\mathrm{C}M} = \frac{\left(\mathrm{F} \times (X_{\mathrm{C}F})\right) + \left(\mathrm{S} \times (X_{\mathrm{C}S})\right)}{M}$$

$X_{\mathrm{C}F}$ = weight fraction of B in feed

$X_{\mathrm{C}S}$ = weight fraction of B in solvent

$X_{\mathrm{C}M}$ = weight fraction of B at mixing point

where x_{af}, x_{bf}, x_{cf} are the mole fractions of components a, b, c in the feed F

The extract point, E1, is determined on the solubility curve by joining a line from point Rn through point M to the solubility curve. The point where this line cuts the solubility curve is the composition of the first extract stage. The difference point is a point found on the intersection of two lines which include a line connecting the feed ad extract (E1) point and another line connecting the raffinate (Rn) and solvent S point. Using the equation of both lines, the

point at which both lines intersect is determined by solving simultaneously the equations of both lines.

The equilibrium number of stages implies the amount of time the extraction process needs to be carried out for optimal and complete separation of solvent from the mixture. After the location of the feed point (point F), solvent composition (point S), final raffinate stage (R_n), mixing point (M), first extract stage (E_1) and difference point (P) the equilibrium number of stages are determined by firstly determining the closest tie-line to the first extract stage (E_1) this tie-line is then projected from point E_1 to the intercept the solubility curve on the raffinate side to determine the equivalent raffinate composition (R_1) since extract and raffinate from the first stage are in equilibrium.

After point R_1 is determined a line is drawn from point R_1 to point P, the point where this line intersects the solubility curve on the extract side becomes the second extract stage (E_2) and the same process as stated above is carried out continuously until the last raffinate stage Rn is achieved.

RESULTS AND DISCUSSIONS

The equilateral triangle constructed used to represent the ternary diagram in the LLE process is shown Figure in 4. It is shown from the figure that the triangle contains gridlines, with each gridline representing the percentage composition of each component.

The correlation between coordinate x, y and each component (A, B, C) determined all points of the equilateral triangle where lines intersect. 66 data points were observed and a relationship for all 66 points was computed and stored. The result of coordinates of x and y values for each side (raffinate and extract) determined using the fuzzy logic inference system function in MATLAB simulation software indicated that for x-coordinate ANFIS training completed at epoch 10, 21 nodes, 12 linear parameters, 4 fuzzy rules and Minimal training RMSE obtained is 0.000059 while the y-coordinate indicated that ANFIS training was completed at epoch 10, 21 nodes, 12 linear parameters, 4 fuzzy rules and Minimal training RMSE obtained is 0.00000.

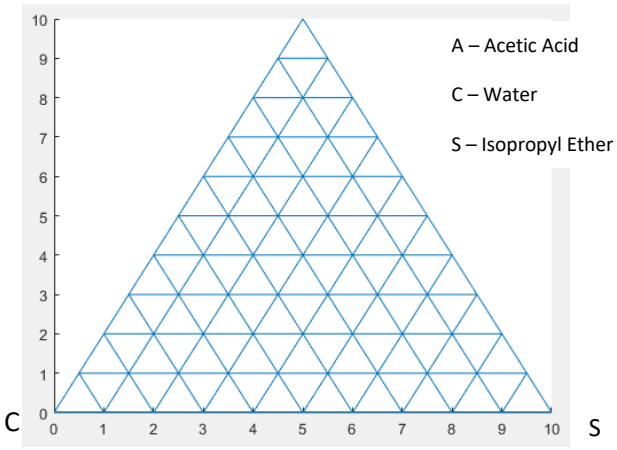

Figure 4: Equilateral triangle showing gridlines representing mole fraction.

The x and y coordinates were used to obtain the solubility curve for the LLE process shown in Figure 5 while the tie-line plot is shown in Figure 6. It is shown in Figure 6 that the Tie-lines join the raffinate and extract sides of the solubility curve. It can be shown that there are nine (9) tie-lines across the solubility curve.

The feed line and mixing point plot are shown in Figure 7. The feed line is the straight black line that connects the two feed streams in the extraction process i.e. pure solvent composition (Isopropyl-ether=100) and the binary mixture composition point (Acetic-Acid=30% and Water=70%). As indicated in Figure 7, Point M represents the mixing point at which there is a combined composition of the feed and solvent entering the system mixed with the feed line.

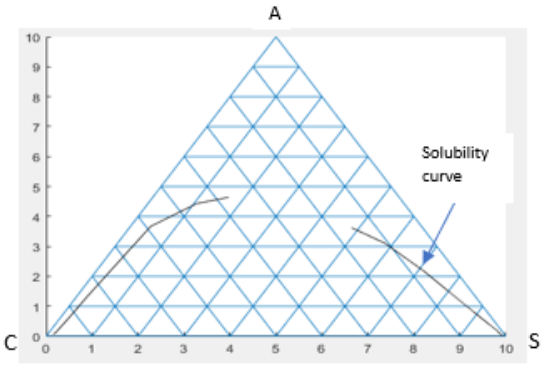

Figure 5: Solubility curve plot

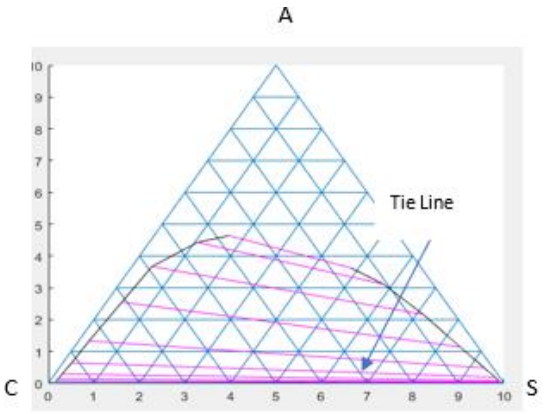

Figure 6: Plotting tie-line

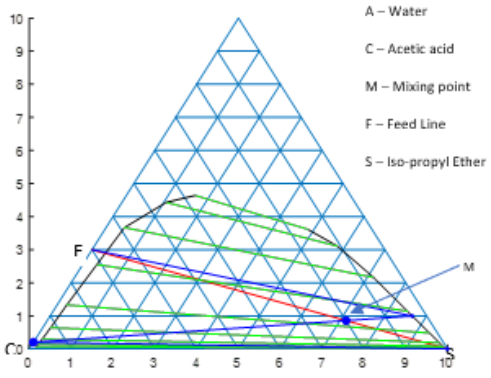

Figure 7: Raffinate and Extract Composition for Mixing Point M

Figure 8 shows the location of Point E_1 and Point P. Point E_1 is indicated by the line drawn from R_N through mixing point M to intersect the solubility curve on the extract side a line was then drawn from this point to feed point F on the raffinate side, the point where this line intersects the solubility curve is the first raffinate point R_1. It is shown in Figure 8 that the operating point P is the point that lies outside the triangle along an extrapolation of the line through R_N and S, F and E_1. It is the point where the straight line drawn from R_N through S and another line drawn from F through E_1 intersects.

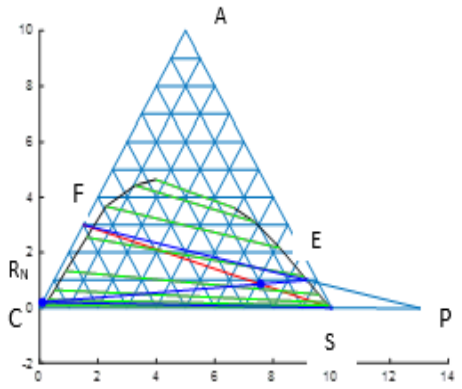

Figure 8: Locating operating point P

The number of stages for the LLE process is shown in Figure 9. Stepping-off stages were determined by the alternative use of tie-lines and operating lines, starting from R_1, a line was drawn from point R_1 to P the point where it is seen to intersect the solubility curve on the extract side is the next extract phase E_2. Using the closest tie-line a line was drawn from point E_2 towards the raffinate side the intersection point on the curve becomes R_2.

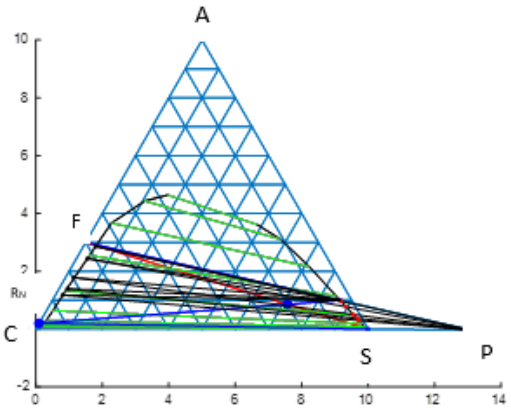

Figure 9: Ternary diagram showing the equilibrium number of stages.

This approach was then repeated until R_N was reached. It can be shown from the result that the number of stages obtained for the work is seven (7). The result obtained was compared with Magdah and Nory (2017) work where the data used was the same, the number of stages obtained when McThiele method and ASPEN software method showed 7 stages.

CONCLUSIONS

The construction of a ternary-phase diagram is not only important to the complete understanding of the liquid-liquid extraction operations but also useful in the analysis of binary and ternary mixtures where chemical analysis may be difficult or impossible. The equilateral triangular diagrams are most useful as they can depict all concentration characteristics of the system. Industrially, it is common for liquid-liquid separation processes to require many stages, however, this computer package reduces the time and extra effort expended for the manual construction of the ternary phase diagram by supplying raffinate, extract and feed compositions of the information necessary to successfully carry out the separation process can be determined. From the development of this computer package, it is seen that tie-lines are of utmost importance as they show the relationship between the raffinate and extract compositions and help in the determination of other extraction properties like the number of required stages which are all very useful in the understanding of the extraction process as a whole.

REFERENCES

Abdelbasit, M., & Salih, N. (2017). Liquid-Liquid Equilibrium for the Design of Extraction. *European Journal of Engineering and Technology*, *5*(4), 18–36.

Lamm, M. H. ., & Jarboe, L. R. (2021). *Chemical Engineering Separations: A Handbook for Students*. https://iastate.pressbooks.pub/chemicalengineeringseparations/

Palmer, J. A., Kaiser, B. A., Polinsky, M. S., Dunn, S. P., Braas, C., Waltz, R., & Baluarte, H. J. (1994). Peritoneal dialysis catheter infections in children after renal transplantation: choosing the time of removal. *Pediatric Nephrology*, *8*(6), 715–718. https://doi.org/10.1007/BF00869099

Schmidt, A., & Strube, J. (2018). Application and Fundamentals of Liquid-Liquid Extraction Processes: Purification of Biologicals, Botanicals, and Strategic Metals. In *Kirk-Othmer Encyclopedia of Chemical Technology*. https://doi.org/10.1002/0471238961.koe00041

Zhang, J., & Hu, B. (2013). Liquid-Liquid Extraction (LLE). In S. Ramaswamy, H.-J. Huang, & B. V. Ramarao2 (Eds.), *Separation and Purification Technologies in Biorefineries* (1st ed., pp. 61–78). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118493441.ch3

APPENDIX

MATLAB CODE FOR THE CONSTRUCTION OF TERNARY DIAGRAM

```
clear
k=1;
for i=1:11
    k1=10-(i-1);
    k2=5-(i-1)*0.5;
    for j=1:i
        x(k)=k2+j-1;
        y(k)=k1;

        A(k)=k1*10;
        B(k)=100-(10*k1)-(j-1)*10;
        C(k)=100-(A(k)+B(k));
        k=k+1;
    end
end

xyABC=[x' y' A' B' C'];
```

```
for i=1:10
    line([(i-1)*0.5 (10-(i-1)*0.5)],
[(i-1) (i-1)]) %horizontal gridlines
    line([10-(i-1) (5-(i-1)*0.5)], [0
10-(i-1)]) %right diagonal gridlines
    line([(i-1) (5+(i-1)*0.5)], [0
10-(i-1)]) %left diagonal gridlines
end

% Determine values for x and y give A
and B
fisX=anfis([A' B',x']); % X colum
relatioship
fisY=anfis([A' B',y']); % Y colum
relatioship


% To evaluate
input=[A' B']
X=evalfis([A' B'], fisX)
Y=evalfis([A' B'], fisY)


% LLE Data
data=  xlsread('Data.xlsx');

% For raffinate side
fis = readfis('tipper');
input=[data(:,2) data(:,1)]
Xr= evalfis(input, fisX);
Yr= evalfis(input, fisY);


% For extract side
fis = readfis('tipper');
input=[data(:,5) data(:,4)];
Xe=evalfis(input, fisX);
Ye=evalfis(input, fisY);

% draw curve using primitive line
line(Xe, Ye, 'color','k');
line(Xr, Yr, 'color','k');


% To draw tie-lines
for i=1:9
    line([Xr(i,1),Xe(i,1)], [Yr(i,1)
Ye(i,1)], 'color','m');
end

points=[Xe' Ye' Xr' Yr'];
pointss=reshape(points,9,4)

% To determine feedline
in= [30 70]
Xu=evalfis(in, fisX);
Yu=evalfis(in, fisY);
E1=[10,Xu];
F1=[0,Yu];
line(E1, F1, 'color','k')

% To determine mixing point
% Overall material balance
F=8000; S=20000; xaf=30; xas=0;
xbf=70; xbs=0; xcf=0; xcs=100;
M=F+S;

% Component material balance
% for component A
Mxam=(F*(xaf))+(S*(xas));
xam=(Mxam/M);

%for component B
Mxbm=(F*(xbf))+(S*(xbs));
xbm=(Mxbm/M);

%for component C
Mxcm=(F*(xcf))+(S*(xcs))
xcm=(Mxcm/M)
inp= [xam xbm]
Xun=evalfis(inp, fisX)
Yun=evalfis(inp, fisY)
line(Xun, Yun, 'Marker','.',
'Markersize', 10)


for i=1:length(Yr)
    if Yr(i)>Yun
        lower=i
        upper=(i+1)
        break
    end
end
% To obtain composition on raffinate
and extract side for the mixing point
for i=1:9
    line([Xr(i,1),Xe(i,1)], [Yr(i,1)
Ye(i,1)], 'color','m')
 % using average method
    avgY= (Yr(lower,:)+Yr(upper,:))/2
    avgX= (Xr(lower,:)+Xr(upper,:))/2
%     line(avgX,avgY, 'Marker','.',
'Markersize', 10)
%     line([avgX,Xun],[avgY,Yun],
'color','k')

    slope1= (Yun-avgY)/(Xun-avgX)
    slope2= (Ye(upper,:)-
Ye(lower,:))/(Xe(upper,:)-
Xe(lower,:))

  % Solving simultaneously using two
slopes
    invA=inv([1 -slope1; 1 -slope2])
    y2=((slope1*(-Xun))+Yun)
    x2=((slope2*(-
Xe(lower,:)))+Ye(lower,:))
    bb=[y2 x2]'
    soln=invA*bb;
end
```

```
inps= [2,98]
Xuv=evalfis(inps, fisX)
Yuv=evalfis(inps, fisY)
line(Xuv,Yuv, 'color','k', 'marker',
'.', 'Markersize',10) %Point Rn
line([Xuv,10],[Yuv, 0], 'color','k',
'marker', '.', 'Markersize',10) %Rn
to S
line([Xuv Xun], [Yuv Yun],
'color','k') %Rn through M

% To determie poit E1 ad joi F to E1
slope1p=(Yun-Yuv)/(Xun-Xuv); %slope
of lie that joi M ad RN
slope2p=(Ye(upper,:)-
Ye(lower,:))/(Xe(upper,:)-
Xe(lower,:)) %slope of line where
point E1 should lie
invAE=inv([1 -slope1p; 1 -slope2p]);
    ye1=((slope1p*(-Xun))+Yun);
    xe1=((slope2p*(-
Xe(lower,:)))+Ye(lower,:));
    be1=[ye1 xe1]';
    E11=invAE*be1;
line(E11(2,:),E11(1,:), 'color','k',
'marker', '.', 'Markersize',10)%
marker "." is used to denote point on
the curve markersize determines its
thickness
line([Xun, E11(2,:)],[Yun, E11(1,:)],
'color','k') %line from point M to
point E1

% line from F to E1
line([Xu, E11(2,:)], [Yu, E11(1,:)],
'color', 'b')


%To determie poit P
slopeE1=(E11(1,:)-Yu)/(E11(2,:)-Xu)%
slope of line through F and E1
slopeRN=(0-Yuv)/(10-Xuv) % slope of
line from RN through S

% Solving simultaneously
invP=inv([-slopeE1 1; -slopeRN 2])
    xp=(E11(1,:))-
(slopeE1*(E11(2,:)))
    yp=(0-slopeRN*10)
    pp=[xp yp]'
    PP=invP*pp
line([E11(2,:),PP(1,:)], [E11(1,:),
PP(2,:)] ) %% to join line E1 to
point P
line([10,PP(1,:)], [0,PP(2,:)]) %% to
join line RN to P


% %Determinig Number of stages
for i=6:-1:3
% To determine R1
slopei=(Yr(i,:)-Ye(i,:))/(Xr(i,:)-
Xe(i,:)) % slope of closest tie-line
to E1
slopeii=(Yr(i,:)-Yr(i-1,:))/(Xr(i,:)-
Xr(i-1,:)) % slope of line on
raffinate side where R1 will lie

% Solve simultaneously
invR1=inv([-slopei 1; -slopeii 1])
    xR1=(-slopei*(E11(2,:)))+E11(1,:)
    yR1=(Yr(i,:)-slopeii*Xr(i,:))
    R1i=[xR1 yR1]'
    R1p=invR1*R1i
line(R1p(1),R1p(2), 'color','k',
'marker', '.', 'Markersize',10) %To
denote point R1
line([E11(2,:),R1p(1,:)], [E11(1,:),
R1p(2,:)], 'color', 'k')   %line from
E1 to R1

% line from R1 to P to get E2
line([R1p(1,:) PP(1,:)], [R1p(2,:)
PP(2,:)], 'color', 'k')
% to determie value of E2

slopeR1=(R1p(2,:)-PP(2,:))/(R1p(1,:)-
PP(1,:)) %slope of lie R1 to P
slopeR1i=(Ye(upper,:)-
Ye(lower,:))/(Xe(upper,:)-
Xe(lower,:))%slope of line where E2
occurs on the curve

% Solve simultaneously
invE2=inv([-slopeR1 1; -slopeR1i 1])
    xE2=(-slopeR1*PP(1,:))+PP(2,:)
    yE2=(-
slopeR1i*Xe(lower,:)+Ye(lower,:))
    E2=[xE2 yE2]'
    E2p=invE2*E2
line(E2p(1,:),E2p(2,:), 'color','k',
'marker', '.', 'Markersize',10)% to
denote E2
end
```