



Comparative Analysis Between Manual and Computer Aided Design of Minimum and Optimum Reflux Ratio at Minimum Reflux Operating Condition of a Distillation Column

Okafor Blessing^{1,*}, Effiong Edward²

¹Department of Chemical Engineering, Chukwuemeka Odumegwu Ojukwu University, Anambra, Nigeria

²Department of Chemical Engineering, Federal University of Technology, Owerri, Nigeria

Email address:

blessingoramalu@yahoo.com (O. Blessing), eeffiong2008@yahoo.com (E. Edward)

*Corresponding author

To cite this article:

Okafor Blessing, Effiong Edward. Comparative Analysis Between Manual and Computer Aided Design of Minimum and Optimum Reflux Ratio at Minimum Reflux Operating Condition of a Distillation Column. *American Journal of Science, Engineering and Technology*. Vol. 1, No. 2, 2016, pp. 13-19. doi: 10.11648/j.ajset.20160102.11

Received: October 30, 2016; **Accepted:** November 30, 2016; **Published:** December 21, 2016

Abstract: This work presents the comparative analysis between manual and computer aided design of minimum and optimum reflux ratio at minimum reflux condition of a distillation column. The Underwood's shortcut design model for determining a column's minimum reflux ratio is applied, and a multiplier of the minimum reflux is used to obtain the optimum reflux ratio of the column. Matlab R2008b is the computer software used to achieve the computer aided design of the reflux due to the need for fast convergence solution to iterative computations. The manual design is a pen and paper approach. The results obtained from both methods were compared for better analysis of the column's separation efficiency at the reflux operating conditions.

Keywords: Reflux Ratio, Underwood Model, Design, MATLAB

1. Introduction

Minimum reflux is an extreme operating condition for a distillation column, and can be approximated by Underwood's method to obtain the minimum and optimum reflux ratio of the column for its optimization.

1.1. Objectives

This work has the following objectives:

- i. To obtain initial reflux ratio manual design estimate for a plate distillation column, using Underwood's shortcut model.
- ii. To obtain a computer based design and optimization of the Underwood's shortcut model.
- iii. To analyze the results obtained from the manual and computer shortcut design methods.

1.2. Benefits

Many chemical engineers do not want to specialize in

designing columns because of how hectic, boring and time wasting their calculations can be. This work will help to show proof that developing a good computer design skill is very efficient, time-saving and is now an essential 21st-generation skill to acquire to fully specialize as a process design expert, as compared to depending on the cumbersome manual design approach as still practiced in many Nigerian universities.

1.3. Matrix Laboratory (MATLAB)

Matlab is a high performance language for technical computing [1]. It integrates modern computation, visualization, and programming environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming making it an excellent tool for teaching and research [2]. Matlab is a fourth generation computer programming tool with highly interactive features making it user-friendly to its human users and other computer programming languages such as Java[®], FORTRAN, C/C++ and giving faster solutions

to iterative and design problems.

1.4. Benefits of the Model

The Underwood’s model serves as a good initializing design model for determining the minimum reflux for more complex distillation columns. It gives fast convergence of design solutions.

1.5. Limitations of the Model

It is impractical for useful operation, but they are valuable guidelines within which the practical distillation must be designed. It assumes a constant relative volatility for each component throughout the column.

2. Shortcut Underwood’s Design Model

Under the minimum reflux condition, Underwood [3] developed sets of equations to calculate the minimum vapour flow rate. In this section, only some important features of the Underwood method are addressed. The Underwood method makes the following assumptions [4]:

- i. Constant vapour and liquid molar overflow in the rectifying section and in the stripping section of a column.
- ii. Constant relative volatility throughout the whole column.

The Underwood method first solves an equation which relates to feed composition x_f , thermal condition of the feed q , and relative volatility α in order to determine a factor Θ , which lies numerically between the relative volatilities of the two key components. The value of the thermal condition of the feed q , according to [5] depends on the nature of the feed as follows:

- i. Cold feed as liquor, $q > 1$
- ii. Feed at boiling point, $q = 1$
- iii. Feed partly vapour, $0 < q < 1$
- iv. Feed saturated vapour, $q = 0$
- v. Feed superheated vapour, $q < 0$

For a binary distillation, the Underwood equations can be easily solved using the two steps listed above, since the two components in the mixture are already taken as the light and heavy key components and there is only one factor lying between the volatilities of such components. For multi-component distillations with only two adjacent distributing components, the Underwood equations can still be solved as binary component calculations and accurate results can be achieved [6]. For multi-component distillations with non-adjacent key components, the Underwood equations are difficult to solve, as multiple reflux factors exist but only one of the factor is required [7].

2.1. Manual-Based Design Steps

- i. Determine the required design specifications
- ii. Calculate the dew point temperature:

$$\sum x_d = \sum (x_d / K) = 1 \quad (1)$$

The temperature is obtained by solving the above equation iteratively. The K-values were obtained from the vapour-liquid equilibrium data for hydrocarbons (Depriester chart) in [5].

- i. Calculate the bubble-point temperature, solving iteratively:

$$\sum x_b = \sum K x_b = 1 \quad (2)$$

- ii. Obtain the relative volatilities of the components at the dew-point and bubble-point temperatures and determine the average:

$$\alpha_i = K_i / K_{HK} \quad (3)$$

$$\alpha_{av} = (\alpha_{i,Td} + \alpha_{i,Tb}) / 2 \quad (4)$$

- iii. Test for α_{av} accuracy in estimation using Douglas’s inequality [8] below:

$$(\alpha_{top} - \alpha_{bot}) / (\alpha_{top} + \alpha_{bot}) \leq 0.1 (\ln \alpha_{top} + \alpha_{bot}) / 2 \quad (5)$$

- iv. Design the minimum reflux ratio at condition of minimum reflux, using the two Underwood’s model equations:

- a) Obtain the reflux factor using equation below:

$${}^n \sum_{i=1} [\alpha_i x_{i,f} / \alpha_i - \Theta] = 1 - q \quad (6)$$

Θ will lie between α_{LK} and α_{HK} , that is $\alpha_{LK} > \Theta > \alpha_{HK}$ [9]

- b) Obtain the minimum reflux ratio using equation (7) below:

$${}^n \sum_{i=1} [\alpha_i x_{i,d} / \alpha_i - \Theta] = R_{min} + 1 \quad (7)$$

2.2. Computer-Based Design Steps

For the column’s variables obtained, the following steps were followed:

- i. State the problem statement of the design
- ii. Define the input and output variables
- iii. Design the program algorithm, convert to MATLAB statements
- iv. Test the program for errors

3. Results and Discussion

Table 1. Results for the product compositions.

| Components | F | x_f | D | x_D | B | x_B |
|---------------|-----|-------------------|----|-------------------|----|-------------------|
| (LK)n-butane | 35 | 0.35 | 34 | 0.971 | 1 | 0.016 |
| (HK)n-pentane | 25 | 0.25 | 1 | 0.029 | 24 | 0.369 |
| n-hexane | 18 | 0.18 | - | - | 18 | 0.277 |
| n-heptane | 22 | 0.22 | - | - | 22 | 0.338 |
| Total: | 100 | $\sum x_f = 1.00$ | 35 | $\sum x_D = 1.00$ | 65 | $\sum x_B = 1.00$ |

Basis: 100kmol of feed

Overall material balance:

Input = output (steady state without chemical reaction). Table 1 above shows the specified product specification used.

$$F = D + B$$

$$100 = 65 + 35$$

$$100\text{kmol} = 100\text{kmol}$$

Table 2. Results for the dew point temperature.

| Component | x _d | T _d = 340k | | T _d = 350k | | T _d = 345k | | T _d = 344k | |
|------------------|----------------|-----------------------|-------------------|-----------------------|-------------------|-----------------------|-------------------|-----------------------|-------------------|
| | | K | x _d /K | K | x _d /K | K | x _d /K | K | x _d /K |
| n-C ₄ | 0.971 | 0.88 | 1.103 | 1.20 | 0.809 | 1.05 | 0.925 | 1.00 | 0.971 |
| n-C ₅ | 0.029 | 0.35 | 0.083 | 0.48 | 0.060 | 0.42 | 0.069 | 0.41 | 0.070 |
| Total: | 1.000 | | 1.186 | | 0.869 | | 0.994 | | 1.041 |

From table 2 above, Σx_d = 0.994 is closest to 1.000
 Therefore (dew point temperature) T_d = 345k (72°C)

Table 3. Results for the bubble point temperature.

| Component | x _b | T _b = 420k | | T _b = 419k | | T _b = 422k | | T _b = 421k | |
|------------------|----------------|-----------------------|-----------------|-----------------------|-----------------|-----------------------|-----------------|-----------------------|-----------------|
| | | K | Kx _b | K | Kx _b | K | Kx _b | K | Kx _b |
| n-C ₄ | 0.016 | 2.90 | 0.046 | 2.40 | 0.038 | 3.05 | 0.049 | 2.98 | 0.048 |
| n-C ₅ | 0.369 | 1.25 | 0.461 | 1.40 | 0.517 | 1.60 | 0.590 | 1.43 | 0.528 |
| n-C ₆ | 0.277 | 0.90 | 0.249 | 0.80 | 0.222 | 0.95 | 0.263 | 0.93 | 0.258 |
| n-C ₇ | 0.338 | 0.50 | 0.169 | 0.36 | 0.122 | 0.55 | 0.186 | 0.52 | 0.176 |
| Total: | 1.000 | | 0.925 | | 0.899 | | 1.088 | | 1.010 |

From table 3, since of all the iterations, ΣKx_b = 1.010 is closest to 1.000
 The bubble-point temperature (T_b) = 421k (148°C)

Table 4. Results for the average relative volatility of components.

| Component | T _d = 345k | | T _b = 421k | | α _{av} = (α _{i,Td} + α _{i,Tb})/2 |
|-----------------------|-----------------------|--|-----------------------|--|--|
| | K _i | α _i = K _i /K _{HK} | K _i | α _i = K _i /K _{HK} | |
| (LK) n-C ₄ | 1.050 | 2.500 | 10.50 | 2.234 | 2.367 |
| (HK) n-C ₅ | 0.420 | 1.000 | 4.70 | 1.000 | 1.000 |
| n-C ₆ | 0.185 | 0.440 | 2.50 | 0.532 | 0.486 |
| n-C ₇ | 0.080 | 0.190 | 1.30 | 0.277 | 0.234 |

Testing for α_{av} accuracy in estimation using Douglas’s inequality below:

$$(\alpha_{top} - \alpha_{bot}) / (\alpha_{top} + \alpha_{bot}) \leq 0.1 (\ln \alpha_{top} + \alpha_{bot}) / 2$$

That is,

$$(2.50 - 2.234) / (2.50 + 2.234) \leq 0.1 (\ln 2.50 + 2.234) / 2$$

$$0.266 / 4.734 \leq 0.1 \ln 2.367$$

$$0.0562 \leq 0.0862$$

Therefore the estimated α_{av} for each component in table 4 are accurate since the left hand side of the inequality is lesser than its right hand side.

Table 5. Result for the reflux factor.

| Cpts | x _{i,r} | α _i | α _i x _{i,r} | Θ = 1.35 | | Θ = 1.36 | | Θ = 1.37 | |
|------------------|------------------|----------------|---------------------------------|--------------------|--------|--------------------|--------|--------------------|--------|
| | | | | α _i - Θ | Div | α _i - Θ | Div | α _i - Θ | Div |
| n-C ₄ | 0.35 | 2.367 | 0.828 | 1.017 | 0.814 | 1.007 | 0.822 | 0.997 | 0.830 |
| n-C ₅ | 0.25 | 1.000 | 0.250 | -0.350 | -0.714 | -0.360 | -0.694 | -0.370 | -0.676 |
| n-C ₆ | 0.18 | 0.486 | 0.087 | -0.864 | -0.101 | -0.874 | -0.099 | -0.884 | -0.098 |
| n-C ₇ | 0.22 | 0.234 | 0.051 | -1.116 | -0.046 | -1.126 | -0.045 | -1.136 | -0.045 |
| TOTAL: | 1.00 | | | | -0.047 | | -0.016 | | 0.011 |

From table 5, Σ(Div) = Σ[(α_ix_{i,r}) / (α_i - Θ)] = 0.011 is closest to 0.00 of all the iterations.

Therefore reflux factor (Θ) = 1.37

Table 6. Result for the minimum reflux ratio.

| Components | x _{i,d} | α _i | α _i x _{i,d} | (α _i - 1.37) | (α _i x _{i,d}) / (α _i - Θ) |
|------------------|------------------|----------------|---------------------------------|-------------------------|---|
| n-C ₄ | 0.971 | 2.367 | 2.298 | 0.997 | 2.305 |
| n-C ₅ | 0.029 | 1.000 | 0.029 | -0.370 | -0.078 |
| Total | 1.000 | | | | 2.227 |

$$2.227 = R_{min} + 1$$

$$R_{min} = 2.227 - 1$$

$$R_{min} = 1.227$$

Approximately from table 6, R_{min} = 1.23

Optimal reflux ratio (R) = 1.15R_{min}

$$R = 1.15 * 1.23$$

$$R = 1.41$$

Table 7. Results of applied design approaches.

| Column variables | Manual results | Program results |
|-------------------|----------------------------|----------------------------|
| T _d | 345k | 345k |
| T _b | 421k | 414k |
| Q _{av,i} | 2.367, 1.000, 0.486, 0.234 | 2.497, 1.000, 0.421, 0.186 |
| Θ | 1.37 | 2.01 |
| R _{min} | 1.23 | 3.95 |
| R | 1.41 | 4.54 |

In table 7, the result generated by the program is same as that of the manual in the obtained dew point temperature, but differs in the other obtained variables.

4. Conclusion

The computer-based approach gave a better analysis for the minimum reflux ratio of the distillation column at fast convergence of design solution which is time-saving for the column’s designer. The manual method is so tedious and time-consuming and contains obvious data approximation errors along the computational approach.

Appendix

For an n-butane and n-pentane splitter, the following specifications were considered:

- Column pressure was taken to be 800kN/m².
- The feed is assumed to be at boiling point, therefore q=1.
- Feed flowrate: 100kmol/s
- Feed compositions: 0.35mole n-butane; 0.25mole n-pentane; 0.18mole n-hexane; 0.22mole n-heptane.
- Ideal multicomponent mixture of n-butane (light key), n-pentane (heavy key), n-hexane and n-heptane is used as the case-study.
- Based on Antoine constants data obtained from [10]:
 For n-butane (C₄): A =15.6782; B=2154.90; C = -34.42
 For n-pentane (C₅): A = 15.8333; B=2477.07; C = -39.94
 For n-hexane (C₆): A = 15.8366; B=2697.55; C = -48.78
 For n-heptane (C₇): A = 15.8737; B=2911.32; C = -56.51
- The column is assumed to utilize a partial condenser.
- Product recovery specification: 97% of n-butane is to be recovered at the distillate.

MATLAB M-file design algorithms:

```
(1)
% file name: dew_point.m
% purpose: this program calculates the dew point
temperature of top
% hydrocarbon distillate.
% Define variables:
% i --Counter for components(1,2,3,...,n)
% Ni --Names of distillate components
% j --Counter for temperatures(1,2,3,...,50)
% Tj --Assumed dew point temperature at j (Kelvin)
% xi --Composition of distillate component
% Ki --Distribution coefficient of component
% Ai,Bi,Ci --Constants in antoine equation for components
% Pi --Pure component vapour pressure for i(N/m^2)
```

```
% P --Total column pressure (N/m^2)
% Prompt the user for the input data:
x1=input('Enter distillate composition of component
1,x1:');
x2=input('Enter distillate composition of component
2,x2:');
A1=input('Enter Antoine constant for component 1,A1:');
A2=input('Enter Antoine constant for component 2,A2:');
B1=input('Enter Antoine constant for component 1,B1:');
B2=input('Enter Antoine constant for component 2,B2:');
C1=input('Enter Antoine constant for component 1,C1:');
C2=input('Enter Antoine constant for component 2,C2:');
P=input('Enter the total column pressure in N/m^2:');
% Calculate pure components vapour pressures
for Tj = 340:1:350;
V1 = A1-(B1/(Tj+C1));
V2 = A2-(B2/(Tj+C2));
P1= exp(V1)*133.32;
fprintf('P1=%f/n',P1);
P2= exp(V2)*133.32;
fprintf('P2=%f/n',P2);
K1= P1/P;
fprintf('K1=%f/n',K1);
K2= P2/P;
fprintf('K2=%f/n',K2);
sum=(x1/K1)+(x2/K2);
if sum<=0.9998;
continue
elseif 0.9999<=sum<=1.0000;
disp('The dew point temperature is obtained');
fprintf('Tj=%6.2f/n',Tj);
end
end
(2)
disp('file name: bubble_point.m')
%This program calculates the bubble point temperature of
a bottom
%distillation column hydrocarbon mixture.
%Definition of variables:
% i --Counter for bottom distillation column
components(1,2,3,...,n)
% Ni --Names of bottom components (n-butane,n-
pentane,n-hexane,n-heptane)
% b --Counter for temperatures(1,2,3,...,50)
% Tb --Assumed bubble point temperature at b (Kelvin)
% xi --Composition of bottom components
% Ki --Distribution coefficient of components
% Ai,Bi,Ci --Constants in antoine equation for components
% Pi --Pure component vapour pressure for i(N/m^2)
```

```

% P --Total column pressure (N/m^2)
% Prompt the user to enter the input data:
x1=input('Enter bottom composition of component 1,x1:');
x2=input('Enter bottom composition of component 2,x2:');
x3=input('Enter bottom composition of component 3,x3:');
x4=input('Enter bottom composition of component 4,x4:');
A1=input('Enter Antoine constant for component 1,A1:');
A2=input('Enter Antoine constant for component 2,A2:');
A3=input('Enter Antoine constant for component 3,A3:');
A4=input('Enter Antoine constant for component 4,A4:');
B1=input('Enter Antoine constant for component 1,B1:');
B2=input('Enter Antoine constant for component 2,B2:');
B3=input('Enter Antoine constant for component 3,B3:');
B4=input('Enter Antoine constant for component 4,B4:');
C1=input('Enter Antoine constant for component 1,C1:');
C2=input('Enter Antoine constant for component 2,C2:');
C3=input('Enter Antoine constant for component 3,C3:');
C4=input('Enter Antoine constant for component 4,C4:');
P=input('Enter the total column pressure in N/m^2:');
% Calculate pure components vapour pressures
for Tb=400:500;
V1 = A1-(B1/(Tb+C1));
V2 = A2-(B2/(Tb+C2));
V3 = A3-(B3/(Tb+C3));
V4 = A4-(B4/(Tb+C4));
P1= exp(V1)*133.32;
fprintf('P1=%f/n',P1);
P2= exp(V2)*133.32;
fprintf('P2=%f/n',P2);
P3= exp(V3)*133.32;
fprintf('P3=%f/n',P3);
P4= exp(V4)*133.32;
fprintf('P4=%f/n',P4);
K1= P1/P;
fprintf('K1=%f/n',K1);
K2= P2/P;
fprintf('K2=%f/n',K2);
K3= P3/P;
fprintf('K3=%f/n',K3);
K4= P4/P;
fprintf('K4=%f/n',K4);
sum=(K1*x1)+(K2*x2)+(K3*x3)+(K4*x4);
if sum<0.999998
continue
elseif 0.999998<=sum<=1.000000;
disp('The bubble point temperature is obtained');
fprintf('Tb=%6.2f/n',Tb);
end
end
3)
disp('file name: avrelative_volatilty.m');
%this program calculates the average relative volatility of
distillation
%components at the dew and bubble point temperatures
% Definition of variables:
%1,5= n-butane(LK)
%2,6= n-pentane(HK)

```

```

%3,7= n-hexane
%4,8= n-heptane
%i -components counter(1,2,3,...)
%Tb --bubble point temperature(Kelvin)
%Td --dew point temperature(Kelvin)
%Ki --Distribution coefficient of components
%Ai,Bi,Ci --Constants in antoine equation for components
%Pi --Pure component vapour pressure for i(N/m^2)
%P --Total column pressure (N/m^2)
%Ai --relative volatility for component i
% prompt the user to enter the input data
Td=input('Enter the dew point temperature,Td:');
Tb=input('Enter the bubble point temperature,Tb:');
A1=input('Enter Antoine constant for component 1,A1:');
A2=input('Enter Antoine constant for component 2,A2:');
A3=input('Enter Antoine constant for component 3,A3:');
A4=input('Enter Antoine constant for component 4,A4:');
B1=input('Enter Antoine constant for component 1,B1:');
B2=input('Enter Antoine constant for component 2,B2:');
B3=input('Enter Antoine constant for component 3,B3:');
B4=input('Enter Antoine constant for component 4,B4:');
C1=input('Enter Antoine constant for component 1,C1:');
C2=input('Enter Antoine constant for component 2,C2:');
C3=input('Enter Antoine constant for component 3,C3:');
C4=input('Enter Antoine constant for component 4,C4:');
P=input('Enter the total column pressure in N/m^2:');
%To calculate the relative volatility at the dew point
temperature:
V1= A1-(B1/(Td+C1));
V2= A2-(B2/(Td+C2));
V3= A3-(B3/(Td+C3));
V4= A4-(B4/(Td+C4));
V5= A1-(B1/(Tb+C1));
V6= A2-(B2/(Tb+C2));
V7= A3-(B3/(Tb+C3));
V8= A4-(B4/(Tb+C4));
P1= exp(V1)*133.32;
P2= exp(V2)*133.32;
P3= exp(V3)*133.32;
P4= exp(V4)*133.32;
P5= exp(V5)*133.32;
P6= exp(V6)*133.32;
P7= exp(V7)*133.32;
P8= exp(V8)*133.32;
K1= P1/P;
fprintf('K1=%f/n',K1);
K2= P2/P;
fprintf('K2=%f/n',K2);
K3= P3/P;
fprintf('K3=%f/n',K3);
K4= P4/P;
fprintf('K4=%f/n',K4);
K5= P5/P;
fprintf('K5=%f/n',K5);
K6= P6/P;
fprintf('K6=%f/n',K6);
K7= P7/P;

```

```
fprintf('K7=%f/n',K7);
K8= P8/P;
fprintf('K8=%f/n',K8);
KHK=K2;
A1=K1/KHK;
disp('the relative volatility of cpt 1 is A1');
fprintf('A1=%6.3f/n',A1);
A2=K2/KHK;
disp('the relative volatility of cpt 2 is A2');
fprintf('A2=%6.3f/n',A2);
A3=K3/KHK;
disp('the relative volatility of cpt 3 is A3');
fprintf('A3=%6.3f/n',A3);
A4=K4/KHK;
disp('the relative volatility of cpt 4 is A4');
fprintf('A=%6.3f/n',A4);
KHK1=K6;
A5=K5/KHK1;
disp('the relative volatility of cpt 1 is A5');
fprintf('A5=%6.3f/n',A5);
A6=K6/KHK1;
disp('the relative volatility of cpt 2 is A6');
fprintf('A6=%6.3f/n',A6);
A7=K7/KHK1;
disp('the relative volatility of cpt 3 is A7');
fprintf('A7=%6.3f/n',A7);
A8=K8/KHK1;
disp('the relative volatility of cpt 4 is A8');
fprintf('A8=%6.3f/n',A8);
Aav1=(A1+A5)/(2);
disp('the av.relative volatility of cpt 1 is A1av');
fprintf('Aav1=%6.3f/n',Aav1);
Aav2=(A2+A6)/(2);
disp('the av.relative volatility of cpt 2 is A2av');
fprintf('Aav2=%6.3f/n',Aav2);
Aav3=(A3+A7)/(2);
disp('the av.relative volatility of cpt 3 is A3av');
fprintf('Aav3=%6.3f/n',Aav3);
Aav4=(A4+A8)/(2);
disp('the av.relative volatility of cpt 4 is A4av');
fprintf('Aav4=%6.3f/n',Aav4);
4)
disp('file name: minimum_refluxratio.m');
%this program obtains the minimum reflux ratio of a
distillation column
%at condition of minimum reflux using Underwood
equations
%Definition of variables used:
% i --Counter for distillation column
components(1,2,3,...,n)
% Ni --Names of components (n-butane,n-pentane,n-
hexane,n-heptane)
% xfi --Composition of feed components
% Ai --average relative volatility of each components
% q --thermal condition of feed
% Q -- reflux factor
% xdi --composition of distillate component
```

```
%prompt the user for the following inputs:
xf1=input('Enter feed composition of component 1,xf1:');
xf2=input('Enter feed composition of component 2,xf2:');
xf3=input('Enter feed composition of component 3,xf3:');
xf4=input('Enter feed composition of component 4,xf4:');
xd1=input('Enter distillate composition of component
1,xd1:');
xd2=input('Enter distillate composition of component
2,xd2:');
A1=input('Enter average relative vilatility for component
1,A1:');
A2=input('Enter average relative vilatility for component
2,A2:');
A3=input('Enter average relative vilatility for component
3,A3:');
A4=input('Enter average relative vilatility for component
4,A4:');
q=input('Enter q:');
s=1-q;
M1=(A1*xf1);
M2=(A2*xf2);
M3=(A3*xf3);
M4=(A4*xf4);
for Q = 1.01:2.36;
N1=A1-Q;
N2=A2-Q;
N3=A3-Q;
N4=A4-Q;
sum=(M1/N1)+(M2/N2)+(M3/N3)+(M4/N);
if sum<0.00;
continue
elseif 0.01>=sum>=0.00
disp('the reflux factor Q is:');
fprintf('Q=%6.2f/n',Q);
end
end
D1=A1*xd1;
fprintf('D1=%6.3f/n',D1);
D2=A2*xd2;
fprintf('D2=%6.3f/n',D2);
E1=A1-Q;
fprintf('E1=%6.3f/n',E1);
E2=A2-Q;
fprintf('E2=%6.3f/n',E2);
summ=(D1/E1)+(D2/E2);
fprintf('summ=%6.3f/n',summ);
Rmin=summ-1;
disp('the minimum reflux ratio is:');
fprintf('Rmin=%6.2f/n',Rmin)
```

Nomenclature

- B – Bottom flowrate in kmol/s
- D – Distillate flowrate in kmol/s
- F – Feed flowrate in kmol/s
- K – Distribution coefficient of components
- MATLAB – Matrix Laboratory

n – Number of components in mixture
 q – Thermal condition of the feed
 R – Actual or optimum reflux ratio
 R_{\min} – minimum reflux ratio
 T – Temperature in Kelvin
 T_b – bubble point temperature in Kelvin
 T_d – dew point temperature Kelvin
 x_b – bottom composition
 x_d – distillate composition
 x_f – feed composition
 x_i – mole fraction of component i in the liquid
 x_{LK} – mole fraction of the light key in the liquid
 x_{HK} – mole fraction of the heavy key in the vapour

Greek

α – relative volatility
 Θ – parameter in Underwood equation or reflux factor
 Σ – summation

Subscripts

av – average
 B, b, bot – bottom of column
 D, d – distillate
 f – feed
 HK – heavy key
 i – component no
 LK – light key
 min – minimum
 top – top of column

References

- [1] The Math Works Inc., "Learning MATLAB 7.0, fifth printing revised for MATLAB 7.1, 2005.
- [2] David Houcque, "Introduction to Matlab for Engineering Students", Version 1.2, 2005, pp. 1-10; 35-51, North-western University.
- [3] A. J. V. Underwood, "Ind. Eng. Chemistry", Vol. 41, No. 12, 1948, pp 2844-2847.
- [4] J. D. Seader, E. J. Henley, "Separation process principles", 2nd edition, 2001, John Wiley and sons Inc., New York, USA.
- [5] J. F. Richardson, J. H. Harker, and J. R. Backhurst, Coulson and Richardson's Chemical Engineering series: "Particle technology and separation processes", Vol. 2, fifth ed., Oxford: Butterworth-Heinemann, 2002, pp. 599-616.
- [6] R. N. Shiras, D. N. Hanson, and C. H. Gibson, Industrial and Engineering Chemistry: "Calculation of minimum reflux in distillation columns", 1950, pp 871-876.
- [7] R. E. Treybal, "Mass transfer operations", 3rd ed., 1980, McGraw-Hill, New York.
- [8] H. Z. Kister, "Distillation design analysis", 1992, pp. 2; 87-123, McGraw-Hill Inc., New York.
- [9] C. J. King, "Separation processes", 1980, McGraw-Hill Inc., New York.
- [10] R. Sinnott, G. Towler, Coulson & Richardson's Chemical Engineering series: "Chemical engineering design", vol. 6, fifth ed., Butterworth-Heinemann, Burlington, USA, 2009, pp 682-715; 1121-1133.