

**Keywords**Fractionation,
Column Diameter Calculation

Received: August 25, 2014

Revised: September 20, 2014

Accepted: September 21, 2014

Matlab programme for the calculation of fractionation column diameter

Hassan Al-Haj Ibrahim, Mohamad Zaher Sabagh

Faculty of chemical and petroleum engineering, Al-Baath University, Homs, Syria

Email address

Hasahi123@hotmail.com (H. Al-Haj I.), Sabagh-mh@hotmail.com (M. Z. Sabagh)

CitationHassan Al-Haj Ibrahim, Mohamad Zaher Sabagh. Matlab Programme for the Calculation of Fractionation Column Diameter. *American Journal of Science and Technology*. Vol. 1, No. 5, 2014, pp. 221-228.**Abstract**

Different charts and calculation methods are available in the literature for the calculation of fractionation column diameter. A computer programme written in Matlab is given in order to facilitate such calculation. The results obtained on applying the programme on two fractionation columns used at an oil refinery are comparable and fall well below the accuracy limits of the methods of calculation.

1. Introduction

Fractionation or fractional distillation is one of the most important unit operations and the most common separation technique. It is a specialized technology that is widely used in the petroleum, chemical, petrochemical, beverage and pharmaceutical industries.

Fractionation is a physical process used to separate a mixture of miscible liquids with different boiling points. The process utilizes the difference of the vapour pressures to produce the separation.

Fractionation columns are designed to achieve this separation efficiently. The correct design of the fractionation column is not always a simple task. It is normally made in two steps; a process design, followed by a mechanical design. The purpose of the mechanical design is to select the tower internals, column diameter and height.

2. Calculation of Column Diameter

In most cases, the accurate calculation of column diameter is not straightforward. It depends on the type of tray used in the column, plate spacing, surface tension and other factors of column design. The principal factor that determines the column diameter, however, is the vapour flow rate. The vapour velocity must be below that which would cause excessive liquid entrainment or a high pressure drop.

The equation given below, which is based on the well-known Souders and Brown equation, can be used to estimate the maximum allowable superficial vapour velocity, and hence the column area and diameter.

$$U_v = \frac{K}{3600} \left(\frac{\rho_L - \rho_V}{\rho_V} \right)^{0.5}$$

Where:

 U_v = Maximum allowable superficial velocity (ft/sec or m/sec) ρ_L = Liquid density (kg/m³ or lb/m³)

ρ_v = Vapour density (kg/m^3 or lb/m^3)

K = A factor that depends on plate spacing. It can be calculated using the following equation:

$$K = 3600 (-0.171 T^2 + 0.27 T - 0.047)$$

Where:

T = plate spacing (m).

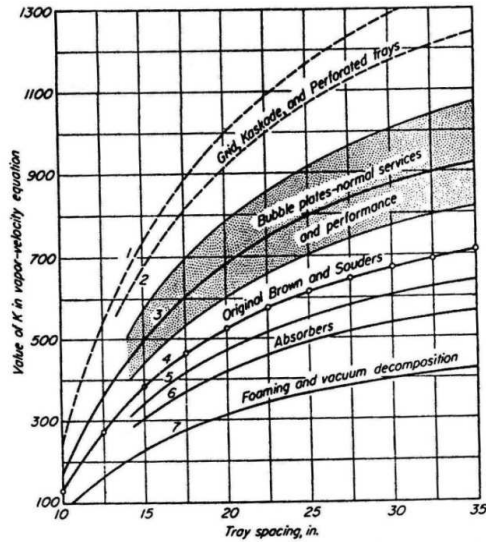


Fig. 1. Effect of tray spacing and type of service on allowable vapour load in fractionators [1].

This factor may also be estimated from charts available in the literature such as shown in Figure 1 (curve 3, for normal performance of bubble plates through normal range of liquid loads at atmospheric and higher pressures, or curve 2 for other tray types), and Figures 2 and 3.

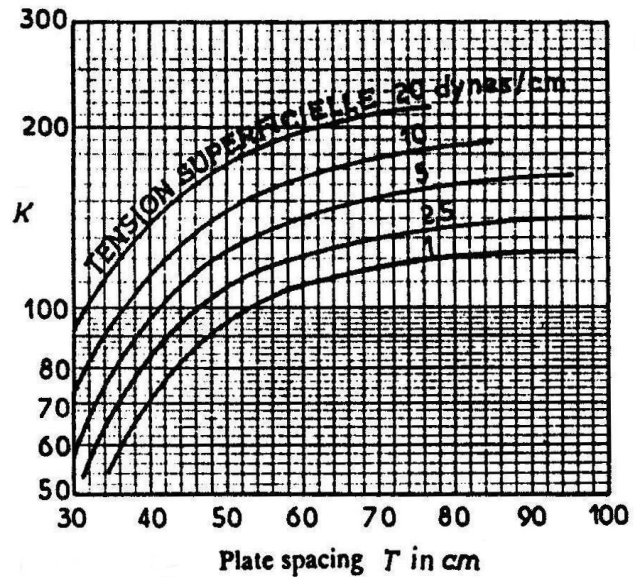


Fig. 2. Coefficient of entrainment as a function of plate spacing and surface tension [2].

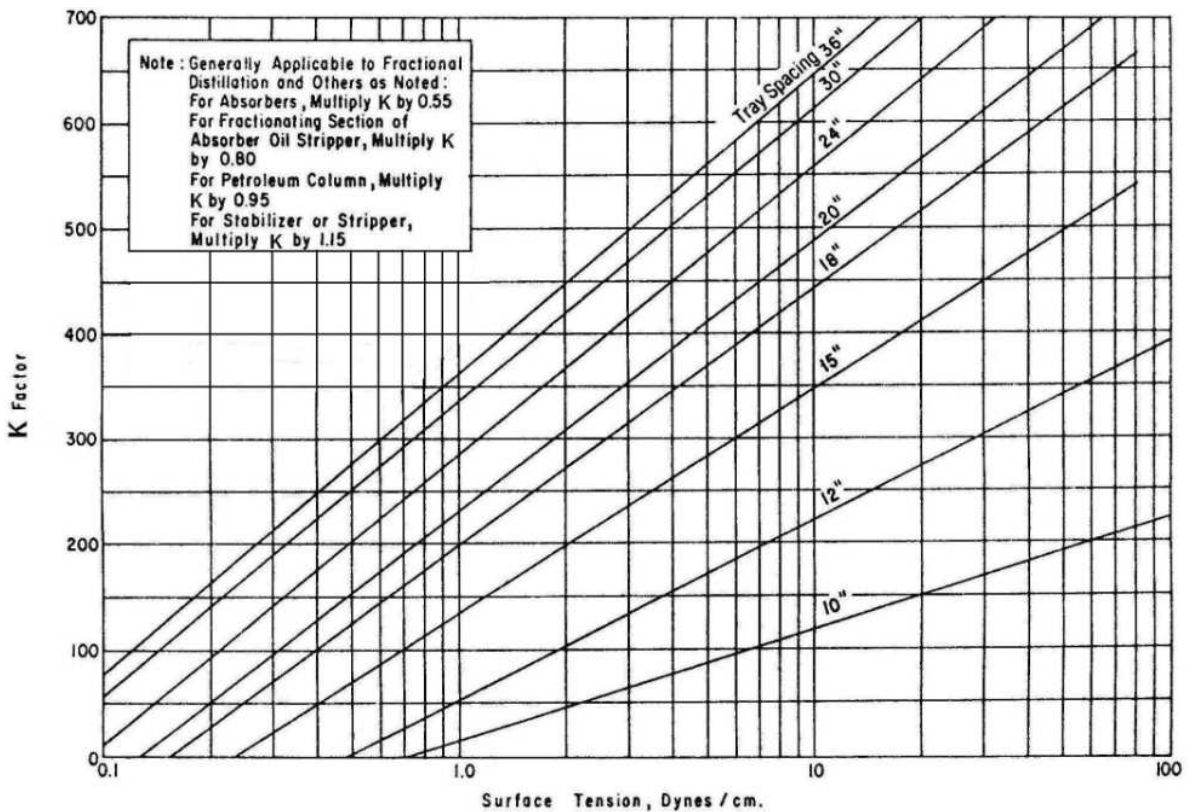


Fig. 3. K factors for column diameter using bubble cap trays.

The K factor is determined at the top and bottom (or intermediate) positions of the column in order to evaluate the point of maximum required diameter.

The column diameter can then be calculated using the following equation:

$$D = \left[\frac{4V}{3600\pi\rho_V U_V} \right]^{0.5}$$

Where:

D = Column diameter (m or ft)

V = Maximum allowable vapour flow rate (kg/h or lb/h)

The diameter obtained in this solution, corresponding to the maximum allowable flow rate, is the minimum acceptable diameter for operation with essentially no entrainment carryover from plate to plate. If irregularities in capacity, system pressure or other significant variables are anticipated, the maximum allowable flow rate can then be divided by a safety factor of 1.10 to 1.25. Alternatively, the maximum flow rate may be multiplied by 1.05 to 1.15 for satisfactory operation at conditions tolerating some entrainment with no noticeable loss in fractionation efficiency.

In another procedure that may also be used for the estimation of column diameter, the chart of Figure 4 may be used to determine the maximum vapour velocity relative to the net area for vapour flow (U_{VN}) for a given value of flow parameter and plate spacing. This may be used for both bubble-cap plates and perforated plates with the following restrictions:

1. The distilling system is non-foaming or low-foaming type.
2. The weir height is less than 15% of the tray spacing.
3. The perforation diameters are equal to or less than 0.25 in.
4. The plate spacings of 6 and 9 in. do not apply to bubble-cap plates.

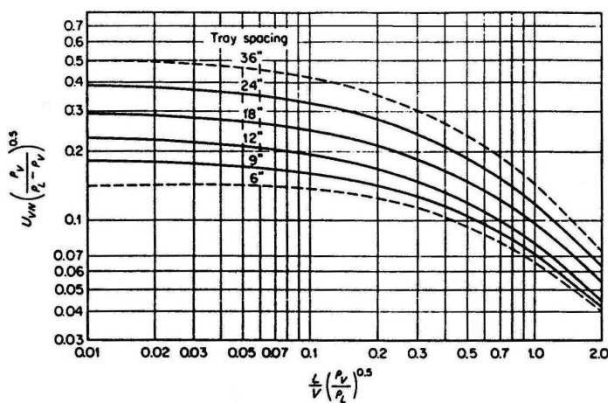


Fig. 4. Flooding limits for bubble cap and perforated plates [3].

Furthermore, the ratio of slot area, or hole area, to active plate area must be equal to or greater than 0.1, and the system surface tension 20 dynes/cm.

For hole-to-active area ratios of less than 0.1, it is recommended by Fair that the vapour velocity from the chart be modified by the following factors:

A_s/A_A	Multiply U_{VN} by
0.10	1.00
0.08	0.90
0.06	0.80

For surface tensions different from 20 dynes/cm, the velocity from the chart is modified by:

$$\text{Corrected } U_{VN} = U_{VN} \left(\frac{\sigma}{20} \right)^{0.2}$$

The flooding vapour velocity can be estimated from the corrected maximum vapour velocity. It is usually about 50-60% of the maximum value, and can be as high as 85% or even 95% if sufficient information is available about the system foaming characteristics. The net area for vapour flow (A_N) can then be calculated and hence the column diameter.

Assuming that the downflow area (A_d) is 10% of the total cross-sectional area of the column (A),

$$A_N = 0.90 A = 0.90 \times 0.785 D^2 = \frac{Q_V}{U_{VN}}$$

$$D^2 = 1.4 \frac{Q_V}{U_{VN}}$$

Where:

Q_V = Vapour volume flow rate (m^3 or ft^3).

3. Matlab Programming

In this paper, a computer programme is written in Matlab for the calculation of fractionator diameter using the various charts and calculation methods available in the literature, Matlab being a programming language that is widely used in all scientific fields and in engineering sciences in particular. Two examples are given below to illustrate the use of the programme.

Example 1

Fractionation column data

T = plate spacing = 22 in. = 56 cm

Q_V = Vapour volume flow rate = $29 \times 10^2 m^3/h = 10 \times 10^4 ft^3/h$

L = Liquid flow rate = $65.3 \times 10^2 kg/h = 14.4 \times 10^3 lb/h$

V = Maximum allowable vapour flow rate = $83.3 \times 10^2 kg/h = 18.3 \times 10^3 lb/h$

ρ_L = Liquid density = $684 kg/m^3 = 42.7 lb/ft^3$

ρ_V = Vapour density = $2.9 kg/m^3 = 0.18 lb/ft^3$

t = Temperature = $141^\circ C$

M = Average molecular weight = 110

The calculated tower diameter varied between 1.0 and 1.2 m depending on the method or chart used for calculation. The results are as follows:

$D_1 = 1.0$ m
 $D_2 = 1.2$ m
 $D_3 = 1.2$ m
 $D_4 = 1.2$ m
 $D_5 = 1.1$ m

Example 2

Fractionation column data

$T =$ plate spacing = 18 in. = 46 cm

$Q_V =$ Vapour volume flow rate = 96×10^2 m³/h = 33.1×10^4 ft³/h

$L =$ Liquid flow rate = 25.6×10^3 kg/h = 56.5×10^3 lb/h

$V =$ Maximum allowable vapour flow rate = 11.5×10^3 kg/h = 25.4×10^3 lb/h

$\rho_L =$ Liquid density = 7.2×10^2 kg/m³ = 45 lb/ft³

$\rho_V =$ Vapour density = 1.2 kg/m³ = 0.075 lb/ft³

$t =$ Temperature = 149°C

$M =$ Average molecular weight = 98

The calculated diameter varied between 1.6 and 2.0 m, and the results are as follows:

$D_1 = 1.6$ m
 $D_2 = 2.0$ m
 $D_3 = 2.0$ m
 $D_4 = 2.0$ m
 $D_5 = 1.8$ m

4. Conclusions

There are several available methods which are normally used by design engineers for the calculation of fractionator diameter. Such methods are based for the most part on the use of charts, tables and empirical equations. Using Matlab computer programming can be of great help and assistance in facilitating the design calculations and improving their accuracy and dependability. The use of charts, tables and empirical equations can then be dispensed with and the different available calculation methods can be directly compared and evaluated in order to achieve better design results. Using Matlab computer programming is, furthermore, time saving which is often an important consideration from a designer's perspective.

In this paper, a computer programme is written in Matlab for the calculation of fractionator column. The results obtained on applying the programme on two fractionation columns used at an oil refinery were found to be comparable to the actual diameters of the columns. Furthermore, the results obtained fall well below the accuracy limits of the methods of calculation.

Appendix

disp('Matlab programme for the calculation of fractionation column diameter')

$N =$ input('N = ')

switch N

Case 1

disp('Fig - 1')

$t =$ input('T = ')
 $x = t$;
 $y = ((0.0254 * x^3 - 2.793 * x^2 + 110.6 * x - 616.6) + 0.5)$;

$k = y$
 disp(['K=' int2str(k)])

$EL =$ input ('EL');

$EV =$ input ('EV');

$V =$ input ('V');

$Uv = k / 3600 * ((EL - EV) / EV)^{0.5}$;

$W = 3600 * EV * Uv$;

$D = (4 * V / (3.14 * W))^{0.5}$;

disp(['D=' num2str(D * 0.3048)])

Case 2

disp(' Fig - 2')

$t =$ input ('t= ');

$M =$ input ('M = ');

if $t == -20$ Then

$x = M$;

$y = 0.000006446 * x^3 - 0.0033908 * x^2 + 0.61513 * x - 9.3259$;

$s = y$;

Else if $t == 40$

$x = M$;

$y = -0.000000018 * x^4 + 0.000014 * x^3 - 0.0044 * x^2 + 0.675 * x - 17$;

$s = y$;

Else if $t > -20$ & $t < 40$

$x = M$;

$y = 0.000006446 * x^3 - 0.0033908 * x^2 + 0.61513 * x - 9.3259$;

$S1 = y$;

$x = M$;

$y = -0.000000018 * x^4 + 0.000014 * x^3 - 0.0044 * x^2 + 0.675 * x - 17$;

$S2 = y$;

$s = S1 - (S1 - S2) / (-20 - 40) * (-20 - t)$;

Else if $t == 90$

$x = M$;

$y = -0.000000001 * x^4 + 0.00001 * x^3 - 0.0036 * x^2 + 0.601 * x - 20$;

$s = y$;

Else if $(t > 40)$ & $(t < 90)$

$x = M$;

$y = -0.000000018 * x^4 + 0.000014 * x^3 - 0.0044 * x^2 + 0.675 * x - 17$;

$S1 = y$;

$x = M$;

$y = -0.000000001 * x^4 + 0.00001 * x^3 - 0.0036 * x^2 + 0.601 * x - 20$;

$S2 = y$;

$s = S1 - (S1 - S2) / (90 - 40) * (90 - t)$;

Else if $t == 150$

$x = M$;

$y = 0.00000235 * x^3 - 0.00154 * x^2 + 0.387 * x - 17.71$;

$s = y$;

Else if $(t > 90)$ & $(t < 150)$

$x = M$;

$y = -0.000000001 * x^4 + 0.00001 * x^3 - 0.0036 * x^2 + 0.601 * x - 20$;

$S1 = y$;

$x = M$;

```

y = 0.00000235 * x ^ 3 - 0.00154 * x ^ 2 + 0.387 * x - 17.71;
S2 = y;
s = S1 - (S1 - S2) / (150 - 90) * (150 - t);
end
disp(['s=' num2str( s)])

t = input('T = ');
if s == 20
x = t;
y = -0.049 * x ^ 2 + 7.941 * x - 97.82;
k = y;
Else if s == 10
x = t;
y = -0.045 * x ^ 2 + 7.359 * x - 107.2;
k = y;
Else if s > 10 & s < 20
x = t;
y = -0.049 * x ^ 2 + 7.941 * x - 97.82;
k1 = y;
x = t;
y = -0.045 * x ^ 2 + 7.359 * x - 107.2;
K2 = y;
k = k1 - (k1 - K2) / (20 - 10) * (20 - s);
Else if s == 5
x = t;
y = -0.034 * x ^ 2 + 5.834 * x - 81.98;
k = y;
Else if (s > 5) & (s < 10)
x = t;
y = -0.045 * x ^ 2 + 7.359 * x - 107.2;
k1 = y;
x = t;
y = -0.034 * x ^ 2 + 5.834 * x - 81.98;
K2 = y;
k = k1 - (k1 - K2) / (10 - 5) * (10 - s);
Else if s == 2.5
x = t;
y = -0.032 * x ^ 2 + 5.4 * x - 83.02;
k = y;
Else if (s > 2.5) & (s < 5)
x = t;
y = -0.034 * x ^ 2 + 5.834 * x - 81.98;
k1 = y;
x = t;
y = -0.032 * x ^ 2 + 5.4 * x - 83.02;
K2 = y;
k = k1 - (k1 - K2) / (5 - 2.5) * (5 - s);
Else if t == 1
x = t;
y = -0.029 * x ^ 2 + 4.918 * x - 77.79;
k = y;
Else if s > 1 & s < 2.5
x = t;
y = -0.032 * x ^ 2 + 5.4 * x - 83.02;
k1 = y;
x = t;
y = -0.029 * x ^ 2 + 4.918 * x - 77.79;

```

```

K2 = y;
k = k1 - (k1 - K2) / (2.5 - 1) * (2.5 - s);
end
disp(['K=' int2str(k)])
EL = input('EL');
EV = input('EV');
V = input('V');
Uv = k / 3600 * ((EL - EV) / EV) ^ 0.5;
W = 3600 * EV * Uv;
D = (4 * V / (3.14 * W)) ^ 0.5;
disp(['D=' num2str(D)])

Case 3
disp(' Fig - 3')
t = input('t= ');
M = input('M = ');
if t == -20
x = M;
y = 0.000006446 * x ^ 3 - 0.0033908 * x ^ 2 + 0.61513 * x - 9.3259;
s = y;
Else if t == 40
x = M;
y = -0.000000018 * x ^ 4 + 0.000014 * x ^ 3 - 0.0044 * x ^ 2 + 0.675 * x - 17;
s = y;
Else if t > -20 & t < 40
x = M;
y = 0.000006446 * x ^ 3 - 0.0033908 * x ^ 2 + 0.61513 * x - 9.3259;
S1 = y;
x = M;
y = -0.000000018 * x ^ 4 + 0.000014 * x ^ 3 - 0.0044 * x ^ 2 + 0.675 * x - 17;
S2 = y;
s = S1 - (S1 - S2) / (-20 - 40) * (-20 - t);
Else if t == 90
x = M;
y = -0.000000001 * x ^ 4 + 0.00001 * x ^ 3 - 0.0036 * x ^ 2 + 0.601 * x - 20;
s = y;
Else if (t > 40) & (t < 90)
x = M;
y = -0.000000018 * x ^ 4 + 0.000014 * x ^ 3 - 0.0044 * x ^ 2 + 0.675 * x - 17;
S1 = y;
x = M;
y = -0.000000001 * x ^ 4 + 0.00001 * x ^ 3 - 0.0036 * x ^ 2 + 0.601 * x - 20;
S2 = y;
s = S1 - (S1 - S2) / (90 - 40) * (90 - t);
Else if t == 150
x = M;
y = 0.00000235 * x ^ 3 - 0.00154 * x ^ 2 + 0.387 * x - 17.71;
s = y;
Else if (t > 90) & (t < 150)
x = M;
y = -0.000000001 * x ^ 4 + 0.00001 * x ^ 3 - 0.0036 * x ^ 2 + 0.601 * x - 20;
S1 = y;
x = M;

```

```

y = 0.00000235 * x ^ 3 - 0.00154 * x ^ 2 + 0.387 * x - 17.71;
S2 = y;
s = S1 - (S1 - S2) / (150 - 90) * (150 - t);
end
disp(['s=', num2str(s)])

t = input('T = ')
if t == 36
x = s;
y = 123.2 * Log(x) + 363.9;
k = y;
Else if t == 30
x = s;
y = 120.7 * Log(x) + 337.3;
k = y;
Else if t < 36 And t > 30
x = s;
y = 123.2 * Log(x) + 363.9;
k1 = y;
x = s;
y = 120.7 * Log(x) + 337.3;
K2 = y;
k = k1 - (k1 - K2) / (36 - 30) * (36 - t);
Else if t == 24
x = s;
y = 118.6 * Log(x) + 287.4;
k = y;
Else if t < 30 And t > 24
x = s;
y = 120.7 * Log(x) + 337.3;
k1 = y;
x = s;
y = 118.6 * Log(x) + 287.4;
K2 = y;
k = k1 - (k1 - K2) / (30 - 24) * (30 - t);
Else if t == 20
x = s;
y = 113.2 * Log(x) + 231.1;
k = y;
Else if t < 24 And t > 20
x = s;
y = 118.6 * Log(x) + 287.4;
k1 = y;
x = s;
y = 113.2 * Log(x) + 231.1;
K2 = y;
k = k1 - (k1 - K2) / (24 - 20) * (24 - t);
Else if t == 18
x = s;
y = 107 * Log(x) + 197.9;
k = y;
Else if t < 20 And t > 18
x = s;
y = 113.2 * Log(x) + 231.1;
k1 = y;
x = s;
y = 107 * Log(x) + 197.9;

```

```

K2 = y;
k = k1 - (k1 - K2) / (20 - 18) * (20 - t);
Else if t == 15
x = s;
y = 92.91 * Log(x) + 136.55;
k = y;
Else if t < 18 And t > 15
x = s;
y = 107 * Log(x) + 197.9;
k1 = y;
x = s;
y = 92.91 * Log(x) + 136.55;
K2 = y;
k = k1 - (k1 - K2) / (18 - 15) * (18 - t);
Else if t == 12
x = s;
y = 74.62 * Log(x) + 53.18;
k = y;
Else if t < 15 And t > 12
x = s;
y = 92.91 * Log(x) + 136.55;
k1 = y;
x = s;
y = 74.62 * Log(x) + 53.18;
K2 = y;
k = k1 - (k1 - K2) / (15 - 12) * (15 - t);
Else if t == 10
x = s;
y = 46.68 * Log(x) + 15.74;
k = y;
Else if t < 12 And t > 10
x = s;
y = 74.62 * Log(x) + 53.18;
k1 = y;
x = s;
y = 46.68 * Log(x) + 15.74;
K2 = y;
k = k1 - (k1 - K2) / (12 - 10) * (12 - t);
end
disp(['k=' int2str(k)])
EL = input('EL');
EV = input('EV');
V = input('V');
Uv = k / 3600 * ((EL - EV) / EV) ^ 0.5;
W = 3600 * EV * Uv;
D = (4 * V / (3.14 * W)) ^ 0.5;
disp(['D=' num2str(D * 0.3048)])

Case 4
disp('Fig - 4')
t = input('t= ');
M = input('M = ');
if t == -20 Then
x = M;
y = 0.000006446 * x ^ 3 - 0.0033908 * x ^ 2 + 0.61513 * x - 9.3259;
s = y;
Else if t == 40

```

```

x = M;
y = -0.000000018 * x ^ 4 + 0.000014 * x ^ 3 - 0.0044 * x ^ 2 + 0.675 * x -
17;
s = y;
Else if t > -20 & t < 40
x = M;
y = 0.000006446 * x ^ 3 - 0.0033908 * x ^ 2 + 0.61513 * x - 9.3259;
S1 = y;
x = M;
y = -0.000000018 * x ^ 4 + 0.000014 * x ^ 3 - 0.0044 * x ^ 2 + 0.675 * x -
17;
S2 = y;
s = S1 - (S1 - S2) / (-20 - 40) * (-20 - t);
Else if t == 90
x = M;
y = -0.00000001 * x ^ 4 + 0.00001 * x ^ 3 - 0.0036 * x ^ 2 + 0.601 * x - 20;
s = y;
Else if (t > 40) & (t < 90)
x = M;
y = -0.000000018 * x ^ 4 + 0.000014 * x ^ 3 - 0.0044 * x ^ 2 + 0.675 * x -
17;
S1 = y;
x = M;
y = -0.00000001 * x ^ 4 + 0.00001 * x ^ 3 - 0.0036 * x ^ 2 + 0.601 * x - 20;
S2 = y;
s = S1 - (S1 - S2) / (90 - 40) * (90 - t);
Else if t == 150
x = M;
y = 0.00000235 * x ^ 3 - 0.00154 * x ^ 2 + 0.387 * x - 17.71;
s = y;
Else if (t > 90) & (t < 150)
x = M;
y = -0.00000001 * x ^ 4 + 0.00001 * x ^ 3 - 0.0036 * x ^ 2 + 0.601 * x - 20;
S1 = y;
x = M;
y = 0.00000235 * x ^ 3 - 0.00154 * x ^ 2 + 0.387 * x - 17.71;
S2 = y;
s = S1 - (S1 - S2) / (150 - 90) * (150 - t);
end
disp(['s=' num2str( s)])
L = input('L = ');
V = input('V = ');
EV = input('EV = ');
DL = input('DL = ');
t = input('T = ');
Qv = input('Qv = ');
pF = (L / V) * (EV / dL) ^ 0.5;
if t == 36
x = pF;
y = 0.104 * x ^ 4 - 0.52 * x ^ 3 + 0.974 * x ^ 2 - 0.916 * x + 0.508;
pc = y;
Else if t == 24
x = pF;
y = 0.111 * x ^ 4 - 0.524 * x ^ 3 + 0.903 * x ^ 2 - 0.765 * x + 0.399;
pc = y;
Else if t > 24 And t < 36
x = pF;
y = 0.104 * x ^ 4 - 0.52 * x ^ 3 + 0.974 * x ^ 2 - 0.916 * x + 0.508;
PC1 = y;
x = pF;
y = 0.111 * x ^ 4 - 0.524 * x ^ 3 + 0.903 * x ^ 2 - 0.765 * x + 0.399;
PC2 = y;
pc = PC1 - (PC1 - PC2) / (36 - 24) * (36 - t);
Else if t == 18
x = pF;
y = 0.058 * x ^ 4 - 0.295 * x ^ 3 + 0.555 * x ^ 2 - 0.518 * x + 0.298;
pc = y;
Else if (t > 18) And (t < 24)
x = pF;
y = 0.111 * x ^ 4 - 0.524 * x ^ 3 + 0.903 * x ^ 2 - 0.765 * x + 0.399;
PC1 = y;
x = pF;
y = 0.058 * x ^ 4 - 0.295 * x ^ 3 + 0.555 * x ^ 2 - 0.518 * x + 0.298;
PC2 = y;
pc = PC1 - (PC1 - PC2) / (24 - 18) * (24 - t);
Else if t == 12
x = pF;
y = -0.053 * x ^ 5 + 0.292 * x ^ 4 - 0.618 * x ^ 3 + 0.663 * x ^ 2 - 0.438 * x
+ 0.234;
pc = y;
Else if (t > 12) And (t < 18)
x = pF;
y = 0.058 * x ^ 4 - 0.295 * x ^ 3 + 0.555 * x ^ 2 - 0.518 * x + 0.298;
PC1 = y;
x = pF;
y = -0.053 * x ^ 5 + 0.292 * x ^ 4 - 0.618 * x ^ 3 + 0.663 * x ^ 2 - 0.438 * x
+ 0.234;
PC2 = y;
pc = PC1 - (PC1 - PC2) / (18 - 12) * (18 - t);
Else if t == 9
x = pF;
y = 0.034 * x ^ 4 - 0.161 * x ^ 3 + 0.284 * x ^ 2 - 0.269 * x + 0.186;
pc = y;
Else if t > 9 And t < 12
x = pF;
y = -0.053 * x ^ 5 + 0.292 * x ^ 4 - 0.618 * x ^ 3 + 0.663 * x ^ 2 - 0.438 * x
+ 0.234;
PC1 = y;
x = pF;
y = 0.034 * x ^ 4 - 0.161 * x ^ 3 + 0.284 * x ^ 2 - 0.269 * x + 0.186;
PC2 = y;
pc = PC1 - (PC1 - PC2) / (12 - 9) * (12 - t);
Else if t == 6
x = pF;
y = -0.013 * x ^ 4 + 0.042 * x ^ 3 - 0.002 * x ^ 2 - 0.11 * x + 0.147;
pc = y;
Else if t > 6 And t < 9
x = pF;
y = 0.034 * x ^ 4 - 0.161 * x ^ 3 + 0.284 * x ^ 2 - 0.269 * x + 0.186;
PC1 = y;
x = pF;
y = -0.013 * x ^ 4 + 0.042 * x ^ 3 - 0.002 * x ^ 2 - 0.11 * x + 0.147;
PC2 = y;
pc = PC1 - (PC1 - PC2) / (9 - 6) * (9 - t);

```

```

end
disp('pc')
disp('pF=')
disp('pF')
disp('pc=')
disp('pc')
corpc = pc * (s / 20) ^ 0.2;
Uvn = corpc / (EV / (DL - EV)) ^ 0.5;
D = Sqrt((4 * Qv / (0.54 * 3.14 * Uvn)) / 60;
disp(['D=' num2str(D * 0.3048)])

Case 5
disp('Eq')
T = input('T');
EL = input('EL');
EV = input('EV');
V = input('V');
k = 3600 * (-0.171 * T ^ 2 + 0.27 * T - 0.047);
Uv = k / 3600 * ((EL - EV) / EV) ^ 0.5;
W = 3600 * EV * Uv;
D = ((4 * V) / (3.14 * W)) ^ 0.5;
disp(['k=' num2str(k)])
disp(['D=' num2str(D)])

End

```

Nomenclature

A_A	Active area (m ² or ft ²)
A_s	Slot area (or hole area) (m ² or ft ²)
D	Column diameter (m or ft)
K	A factor that depends on plate spacing.
L	Liquid flow rate (kg/h or lb/h)
M	Average molecular weight
Q_V	Vapour volume flow rate (m ³ or ft ³).
T	plate spacing (in., cm or m)
t	Temperature (°C)
U_V	Maximum allowable superficial vapour velocity (m/sec or ft/sec).
U_{VN}	Vapour velocity relative to the net area for vapour flow (m/sec or ft/sec).
V	Maximum allowable vapour flow rate (kg/h or lb/h)
ρ_L	Liquid density (kg/m ³ or lb/ft ³)
ρ_V	Vapour density (kg/m ³ or lb/ft ³)
σ	Surface tension (dynes/cm)

References

- [1] Nelson WL, Petroleum refinery engineering, McGraw-Hill Book Co., New York, (1958).
- [2] Wuithier P (Ed.), Le pétrole : Raffinage et génie chimique, L'institut français du pétrole, Paris, (1972), Vol. 2.
- [3] Van Winkle M, Distillation, McGraw-Hill Book Co., New York, (1967).