



RAMAIAH
Institute of Technology

COMPUTATIONAL METHODS LABORATORY MANUAL

M S RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
(Approved by AICTE, New Delhi & Govt. of Karnataka)
Accredited by NBA & NAAC with 'A+' Grade

INDEX

Program No	Title	Page No
1	Solution of non-linear algebraic equation – Regula-Falsi, Secant and Newton-Raphson	1
1.a	Program to evaluate molar volume of a gas using van der Waal equation of state.	5
1.b	Program to evaluate molar volume of a mixture of gases using Redlich-Kwong EOS	7
2	Solution of simultaneous linear equations – Jacobi and Gauss-Siedel method	7
2.a	Mixing of polymer streams	10
3	Solution of definite integral equation – Trapezoidal and Simpson's 1/3 and 3/8 method	12
3.a	Average specific heat	12
3.b	Rayleigh's equation	13
4	Solution of ordinary differential equations – Runge-Kutta 4 th order method	14
4.a	Concentration profile in a plug flow reactor	14
4.b	Temperature gradient across furnace wall	15
5	Curve fitting – Method of Least Squares	16
5.a	f vs Nre for laminar and turbulent flow	16
5.b	Arrhenius equation – Relation between k and T	17
5.c	Settling velocity – Relation between U and D_p	18
6	MS Excel – Root of on-linear equation using <i>goalseek</i>	20
7	MS-Excel – Material balance for a flow sheet and optimization using <i>solver</i>	22
8	MATLAB - Bubble and Dew calculations	30
8.a	MATLAB - Bubble pressure calculation for an ideal binary mixture	31
8.b	MATLAB- Dew pressure calculation for an ideal binary mixture	32
9.a	MATLAB- Bubble temperature calculation for an ideal binary mixture	33
9.b	MATLAB-Dew temperature for an ideal binary mixture	35
10	MATLAB-Flash vaporization calculations	37
11	MATLAB-Residence time of a batch reactor	40
12	MATLAB-Volume of a PFR	41
13	MATLAB- Volume of CSTR	44
14	MATLAB- Double pipe heat exchanger design	45
	Questions for viva	47

Program 1

Solution of Non-linear algebraic equation

Fixed point iteration method: The transcendental equation $f(x) = 0$ can be converted algebraically into the form $x = g(x)$ and then using the iterative scheme with the recursive relation

$$x_{i+1} = g(x_i), \quad i = 0, 1, 2, \dots,$$

with some initial guess x_0 is called the fixed point iterative scheme.

Algorithm:

Given an equation $f(x) = 0$

Convert $f(x) = 0$ into the form $x = g(x)$

Let the initial guess be x_0

Do

$$x_{i+1} = g(x_i)$$

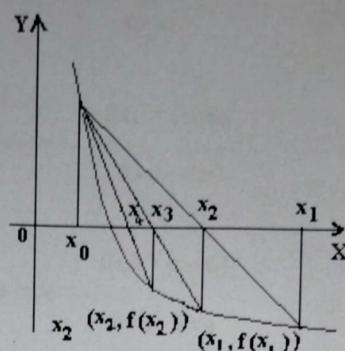
while any one of the convergence criterion is met

1. Fixing apriori the total number of iterations N . or
2. By testing the condition $|x_{i+1} - g(x_i)|$ (where i is the iteration number) less than some tolerance limit, say epsilon, fixed apriori.

Regula Falsi method: The convergence process in the bisection method is very slow. It depends only on the choice of end points of the interval $[a,b]$. The function $f(x)$ does not have any role in finding the point c (which is just the mid-point of a and b). It is used only to decide the next smaller interval $[a,c]$ or $[c,b]$. A better approximation to c can be obtained by taking the straight line L joining the points $(a,f(a))$ and $(b,f(b))$ intersecting the x -axis. To obtain the value of c we can equate the two expressions of the slope m of the line L .

$$m = \frac{f(b) - f(a)}{(b-a)} = \frac{0 - f(b)}{(c-b)}$$

$$\Rightarrow (c-b) * (f(b)-f(a)) = -(b-a) * f(b)$$



$$c = b - \frac{f(b) * (b-a)}{f(b) - f(a)}$$

Now the next smaller interval which brackets the root can be obtained by checking

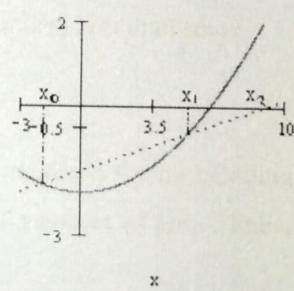
$f(a) * f(b) < 0$ then $b = c$

> 0 then $a = c$

$= 0$ then c is the root.

Selecting c by the above expression is called Regula-Falsi method or False position method.

Secant method: The Newton-Raphson algorithm requires the evaluation of two functions (the function and its derivative) per each iteration. If they are complicated expressions it will take considerable amount of effort to do hand calculations or large amount of CPU time for machine calculations. Hence it is desirable to have a method that converges (please see the section order of the numerical methods for theoretical details) as fast as Newton's method yet involves only the evaluation of the function. Let x_0 and x_1 are two initial approximations for the root 's' of $f(x) = 0$ and $f(x_0)$ & $f(x_1)$ respectively, are their function values. If x_2 is the point of intersection of x-axis and the line joining the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ then x_2 is closer to 's' than x_0 and x_1 . The equation relating x_0 , x_1 and x_2 is found by considering the slope 'm'



$$\begin{array}{ccc}
 f(x_1) - f(x_0) & f(x_2) - f(x_1) & 0 - f(x_1) \\
 m = & = & = \\
 x_1 - x_0 & x_2 - x_1 & x_2 - x_1 \\
 \\
 \frac{-f(x_1) * (x_1 - x_0)}{x_2 - x_1} = & & \\
 & f(x_1) - f(x_0) & \\
 \\
 \frac{f(x_1) * (x_1 - x_0)}{x_2 - x_1} & & \\
 & f(x_1) - f(x_0) &
 \end{array}$$

or in general the iterative process can be written as

$$\begin{array}{c}
 f(x_i) * (x_i - x_{i-1}) \\
 x_{i+1} = x_i - \frac{f(x_i) - f(x_{i-1})}{f(x_i) * (x_i - x_{i-1})} \quad i = 1, 2, 3, \dots
 \end{array}$$

Algorithm:

Given an equation $f(x) = 0$

Let the initial guesses be x_0 and x_1

Do

$$\begin{array}{c}
 f(x_i) * (x_i - x_{i-1}) \\
 x_{i+1} = x_i - \frac{f(x_i) - f(x_{i-1})}{f(x_i) * (x_i - x_{i-1})} \quad i = 1, 2, 3, \dots
 \end{array}$$

While any one of the convergence criteria mentioned below are met

Fixing apriori the total number of iterations N.

By testing the condition $|x_{i+1} - x_i|$ (where i is the iteration number) less than some tolerance limit, say epsilon, fixed apriori.

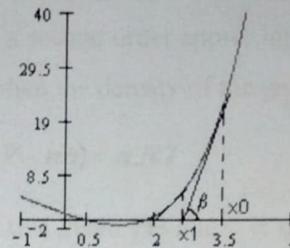
The Newton-Raphson method is one of the most widely used methods for root finding. It can be easily generalized to the problem of finding solutions of a system of non-linear equations, which is referred to as Newton's technique.

Let the given equation be $f(x) = 0$ and the initial approximation for the root is x_0 . Draw a tangent to the curve $y = f(x)$ at x_0 and extend the tangent until x-axis. Then the point of intersection of the tangent and the x-axis is the next approximation for the root of $f(x) = 0$. Repeat the procedure with $x_0 = x_1$ until it converges. If m is the slope of the Tangent at the point x_0 and b is the angle between the tangent and x-axis then

$$m = \tan b = f'(x_0) = \frac{f(x_0)}{x_0 - x_1}$$

$$(x_0 - x_1) * f'(x_0) = f(x_0)$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$



This can be generalized to the iterative process as

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad i = 0, 1, 2, \dots$$

The same also can be obtained from the Taylor series. Let $x_1 = x_0 + h$ be the root of $f(x) = 0$

$$\begin{aligned} f(x_1) &= f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \dots \\ 0 &= f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \dots \end{aligned}$$

$$h = -\frac{f(x_0)}{f'(x_0)}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

or in general

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad i = 0, 1, 2, \dots$$

- a. Write a program to determine the molar volume of a gas using van der Waal's equation of state.

Theory:

The Ideal Gas Law, $PV = nRT$, can be derived by assuming that the molecules that make up the gas have negligible sizes, that their collision with themselves and the wall are perfectly elastic, and that the molecules have no interactions with each other. The van der Waal's equation is a second order approximation of the equation of state of a gas that will work even when the density of the gas is not low.

$$\left(P + a \left(\frac{n}{V} \right)^2 \right) (V - nb) = nRT$$

Here **a** and **b** are constants particular to a given gas and **n** is the number of moles. This equation was derived in 1873 by Johannes Diderik van der Waals who received Nobel prize in 1910 for his work on the equation of state for gases and liquids.

Some van der Waals Constants:

Component	a (J m ³ /mole ²)	b (m ³ /mole)	P _c (MPa)	T _c (K)
Air	.1358	3.64x10 ⁻⁵	3.77	133
Carbon Dioxide (CO ₂)	.3643	4.27x10 ⁻⁵	7.39	304.2
Nitrogen (N ₂)	.1361	3.85x10 ⁻⁵	3.39	126.2
Hydrogen (H ₂)	.0247	2.65x10 ⁻⁵	1.30	33.2
Water (H ₂ O)	.5507	3.04x10 ⁻⁵	22.09	647.3
Ammonia (NH ₃)	.4233	3.73x10 ⁻⁵	11.28	406
Helium (He)	.00341	2.34x10 ⁻⁵	0.23	5.2
Freon (CCl ₂ F ₂)	1.078	9.98x10 ⁻⁵	4.12	385

The parameter **b** is related to the size of each molecule. The volume that the molecules have to move around in is not just the volume of the container **V**, but is reduced to (**V** - **nb**). The parameter **a** is related to intermolecular attractive force between the molecules, and **n/V** is the density of molecules. The net effect of the intermolecular attractive force is to reduce the pressure for a given volume and temperature.

When the density of the gas is low (i.e., when n/V is small and nb is small compared to V) the van der Waals equation reduces to that of the ideal gas law. One region where the van der Waals equation works well is for temperatures that are slightly above the critical temperature T_c of a substance. Observe that inert gases like Helium have a low value of a as one would expect since such gases do not interact very strongly, and that large molecules like Freon have large values of b .

The molar volume is calculated using vanderwaal's equation which is non linear and is solved using Newton-Raphson Method .

Notation:

P	Pressure
T	Temperature
P_c	Critical Pressure
T_c	Critical Temperature
R	Gas Constant
V	Molar volume

Equations:

$$(P + a/V^2)(V-b) = RT \text{ where } a = (27R^2T_c^2)/(64P_c) \text{ and } b = RT_c/(8P_c)$$

$$f(v) = (P + a/V^2)(V-b) - RT$$

Algorithm :

1. Start
2. Input name of the component
3. Input pressure, temperature, critical temperature critical pressure.
4. Input accuracy and number of iterations 'N'
5. Calculate a and b
6. Calculate the guess value of Vo using perfect gas law
7. Initialize n $\leftarrow 0$
8. Calculate F(Vo) & F'(Vo)
9. Find the new estimate for v using N-R method

$$V_1 = V_o - f(V_o)/f'(V_o) \text{ where } f'(V_o) \text{ is not zero}$$

10. Find error $\text{eps} = (\text{V}_1 - \text{V}_0)/\text{V}_0$
11. Increment $N=N+1$
12. Check if $N < \text{number of iterations}$
13. If $\text{eps} < \text{accuracy}$, $\text{V}_0 \leftarrow \text{V}_1$ go to step 8 , else
14. Print V and N.

b. Write a program to evaluate molar volume of a mixture of gases for the following data using Redlich - Kwong equation of state (EOS).

Theory :

The very first noteworthy successful modification to the attraction parameter came with the publication of the equation of state of Redlich-Kwong in 1949. Redlich and Kwong revised van der Waals EOS and proposed the following expression:

$$\left(P + \frac{a}{T^{0.5}\tilde{v}(\tilde{v} + b)} \right) (\tilde{v} - b) = RT$$

Notice that the fundamental change they introduced was to the functional form of $\delta P_{\text{attraction}}$. Additionally, they introduced the co-volume “b” into the denominator of this functional form.

The important concept here is that the attraction parameter “a” of van der Waals needed to be made a function of temperature before any cubic EOS was able to do a better job of quantitatively matching experimental data. This was a realization that van der Waals himself had suggested, but no actual functional dependency had been introduced until the Redlich-Kwong EOS.

$$a = 0.427480 \frac{R^2 T_c^{2.5}}{P_c}$$

$$b = 0.086640 \frac{RT_c}{P_c}$$

This EOS radically improved, in a quantitative sense, the predictions of van der Waals EOS. van der Waals -type equations are cubic because they are cubic polynomials in molar volume and compressibility factor. This equation has 3 volume roots of which two may be complex. When $T > T_c$, solution of V at any positive value of P yields only real positive root. $T = T_c$, except at critical pressure where there are 3 roots all equal to V_c , this is also true. $T < T_c$, only one real positive root exists at high pressures, but for a range of

lower pressures there are three real positive roots. The smallest root is a liquid and the largest root is a vapour. It comes as no surprise then, that we can transform equation Redlich-Kwong EOS into:

$$\tilde{v}^3 - \left(\frac{RT}{P} \right) \tilde{v}^2 + \frac{1}{P} \left(\frac{a}{T^{0.5}} - bRT - pb^2 \right) \tilde{v} - \frac{ab}{PT^{0.5}} = 0$$

and, by defining the following parameters,

$$A = \frac{aP}{R^2 T^{2.5}}$$

$$B = \frac{bP}{RT}$$

and introducing the compressibility factor definition ($Z = \frac{P\tilde{v}}{RT}$), we get:

$$Z^3 - Z^2 + (A - B - B^2)Z - AB = 0$$

We may also verify the two-parameter corresponding state theory by introducing equations as

$$Z^3 - Z^2 + \frac{P_r}{T_r} \left(\frac{0.42748}{T_r^{1.5}} - 0.08664 - 0.007506 \frac{P_r}{T_r} \right) Z - 0.03704 \frac{P_r^2}{T_r^{3.5}} = 0$$

For mixtures, however, we apply *the same equation*, but we impose certain mixing rules to obtain "a" and "b", which are functions of the properties of the pure components. Strictly speaking, we create a new "pseudo" pure substance that has the average properties of the mixture. Redlich-Kwong preserved the same mixing rules that van der Waals proposed for his EOS:

$$a_m = \sum_i \sum_j y_i y_j a_{ij}$$

$$a_{ij} = \sqrt{a_i a_j}$$

$$b_m = \sum_i y_i b_i$$

Naturally, Redlich and Kwong did not have the last word on possible improvements to the vdW EOS. The Redlich-Kwong EOS, as shown here, is no longer used in practical applications. Research continued and brought with it new attempts to improve the RK EOS. After more than two decades, a modified RK EOS with very good potential developed was Soave-RK EOS.

Notation :

P	=	Pressure
T	=	Temperature
R	=	Gas constant
a,b,	=	constants

For multicomponent system,

Algorithm:

1. Input the Number of components
 2. Input the compound name, critical temperature, critical pressure, composition, pressure for all components and gas constant.
 3. Accept temperature, pressure.
 4. Calculate $P_{cm} = \sum x_j P_{cj}$; $T_{cm} = \sum x_j T_{cj}$
 $a = \sum [(aj)^{1/2} x_j]^2$ $b = \sum b_j x_j$
 5. Calculate T_r and P_r
- $$T_r = \frac{T}{T_{cm}} \quad P_r = \frac{P}{P_{cm}}$$
6. Calculate $V_o = \frac{RT}{P}$, initial value of molar volume
 7. Calculate f_v , fdv , V_n .

$$f_v = \left[P + \frac{a}{T^{0.5} V(v+b)} \right] [V - b] - RT$$

$$fdv = \frac{RT}{(V-b)^2} - \frac{a}{T^{0.5}} \left[\frac{2V+b}{V^2(v+b)^2} \right]$$

$$V_n = V_o - \frac{f_v}{fdv}$$

Where, fdv is not zero

8. Check for accuracy $\frac{V_n - V_o}{V_o} < \text{acc.}$
9. If yes, print molar volume and no of iterations
 If no, increment count by 1 and $V_o = V_n$ and repeat steps 7 and 8.
10. Stop.

c. The rate of heat loss from a steam pipe is given by

$$q = 1.54 \times 10^{-9} T^4 + 0.38(T - 540)^{1.25}$$

where T is the temperature of the outer surface of pipe. If the rate of heat loss is 250 in consistent units, write a program to determine the equilibrium temperature using fixed point iteration, regula falsi, secant and Newton-Raphson, method to compare the results.

Program 2**Solution of simultaneous linear equations**

- a. A polymer blend is to be formed from three feed streams A, B and C having the following compositions in weight percent. Write a program to determine A,B and C for 500 kg of blend production. Use Jacobi and Gauss-siedel methods and compare the results.

Component	A	B	C	Blend
1	55	35	5	30
2	35	50	40	30
3	10	15	55	40

The first iterative technique is called the Jacobi method, after Carl Gustav Jacob Jacobi (1804–1851). This method makes two assumptions: (1) that the system given by has a unique solution and (2) that the coefficient matrix A has no zeros on its main diagonal. If any of the diagonal entries are zero, then rows or columns must be interchanged to obtain a coefficient matrix that has nonzero entries on the main diagonal.

To begin the Jacobi method, solve the first equation for the second equation for and so on, as follows. Then make an initial approximation of the solution, Initial approximation and substitute these values of into the right-hand.

A modification of the Jacobi method is called the Gauss-Seidel method, named after Carl Friedrich Gauss (1777–1855) and Philipp L. Seidel (1821–1896). This modification is no more difficult to use than the Jacobi method, and it often requires fewer iterations to produce the same degree of accuracy. With the Jacobi method, the values of obtained in the nth approximation remain unchanged until the entire nth approximation has been calculated. With the Gauss-Seidel method, on the other hand, you use the new values of each as soon as they are known. That is, once you have determined from the first equation, its value is then used in the second equation to obtain the new Similarly, the new and are used in the third equation to obtain the new and so on.

Algorithm

1. Start.
2. Declare the variables and initialize with required values.

3. Ask the user to enter the accuracy.
4. Input the accuracy.
5. Initialize flag variable i=0 and initial values xo,yo,zo to 0.
6. Calculate values of x, y, z using Gauss Seidel formula:

$$x = (a4 - (a2 * yo) - (a3 * zo)) / a1$$

$$y = (b4 - (b1 * xo) - (b3 * zo)) / a2$$

$$z = (c4 - (c1 * xo) - (c2 * yo)) / a3$$

7. Check if the input accuracy is obtained for x,y,z.

If yes,

Print values of x, y, z.

Update flag variable ,i=1

If no,

Update initial values as:

$$Xo=x$$

$$Yo=y$$

$$Zo=z$$

8. Check the value of flag variable, i :

If i=0,

Go to step 6.

If i=1,

Exit

9. Stop.

Program 3**Solution of definite integral equation**

- a) Write a program to find average specific heat of methane using trapezoidal, simpson's 1/3rd and 3/8th rule. T₁=300K T₂ = 400K

Theory:

Simpson's procedure consists of subdividing the interval a < x < b into h/2 subintervals, each of length 2h, where n is an even integer. It is given as

$$\int_{x_{n-2}}^{x_n} f(x) dx = \frac{h}{3} (y_{n-2} + 4y_{n-1} + y_n)$$

Step length h is calculated as

$$h = \frac{(T_2 - T_1)}{n}$$

Algorithm:

1. Start
2. Input compound name
3. Input initial and final temperature
4. Input the constants of compound a,b,c
5. Assign the number of segments.
6. calculate the step size – h = (T₂ – T₁)/n
7. If h is even call function

$$f2=f(x+h,a,b,c)$$

$$f3=f(x+2h,a,b,c)$$

$$\text{Sum} = \text{Sum} + 4f2 + f1 + f3$$

$$f1=f3$$

8. Print average specific heat.

9. If h is not even, go to step 6

/* function */

1. Initialize variable that are required
2. calculate h=a*(b*t) + (cxt²).

Return h.

b) 500 moles (F) of a mixture of two components A and B placed in a simple batch distillation still. The initial composition is 20 mole% A and 80 mole% B. After the mixture is distilled for some time, the residue left in the still contains 3 mole% A. Write program to evaluate total moles (W) left in the still, given the equation

$$\ln(F/W) = \int_{x_0}^{x_f} \frac{dx}{y^* - x} \quad \text{where } y^* = \frac{ax}{1 + (a-1)x^2}$$

where 'x' is the mole fraction of A in liquid phase y^* is the equilibrium vapour phase composition and 'a' is a constant equal to 2.5. Use Simpson's 1/3rd method. Compare the result with trapezoidal method.

1. Start.
2. Declare the variables and initialize with required values.
3. Ask the user to input the number of divisions, n, to be used in Trapezoidal method.
4. Calculate step length, h :

$$h = (b-a)/2$$

5. Update value of Sum to the sum of first and last terms of the trapezoidal formula.
6. Initialize loop variable, i=1

Calculate sum, Sum, of the terms of integral, I by calling the user-defined function for each term. , for $i < n$

7. Find integral, I, using:

$$I = h * \text{Sum}$$

8. Calculate residue, W :

$$W = F/e^I$$

9. Print value of residue.

10. Stop.

Program 4

Solution of Ordinary Differential equations

- a. Write a program to calculate concentration for a range $dc/dx = ac^{1.5}(1+x^{0.5})$
 $0 < x < 1$ where $a = \text{constant} = 0.5$ X is distance. Using fourth order range
 Kutta method in steps of 0.1

Theory :

Consider $dy/dx = f(x, y)$

$$Y(x_0) = y_0 \text{ and } x_1 = x_0 + h$$

According to Euler's Method $y_1 = y_0 + hf(x_0, y_0)$

$$K_1 = h * f(x_0, y_0)$$

$$K_2 = h * f(x_0 + h/2, y_0 + k_1/2)$$

$$K_3 = h * f(x_0 + h/2, y_0 + k_2/2)$$

$$K_4 = h * f(x_0 + h, y_0 + k_3)$$

$$y_1 = y_0 + (1/6) \cdot (K_1 + 2K_2 + 2K_3 + K_4)$$

In Runge Kutta method of order 4 the truncation error is h^4 , hence it requires 4 evolutions per step. This is method of averaging the slopes at different points in the given interval. R.K. method suggests average suggests average slopes at different number of points between x_0 and $x_0 + h$.

Algorithm :

1. Start
2. Assign the initial value of x,y.
3. Input the value of x a at which y is required
4. Decide the step size h.
5. Compute the no. of steps required $n = (x - x_0)/h$
6. Find the values of K_1, K_2, K_3, K_4 for first step $n=0$.

$$K_1 = h * f(x_0, y_0)$$

$$K_2 = h * f\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$$

$$K_3 = h * f\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right)$$

$$K_4 = h * f(x_0 + h, y_0 + k_3)$$

7. Compute $Y_{n+1} = y_{n+1} = \frac{1}{6}(K_1 + K_2 + 2K_3 + K_4)$
8. Increase the value of X_n by 1 step
9. Repeat steps till $X_n \leq X$
10. Display values of X and y .

- b. The temperature gradient in a furnace wall, heated from one side at a particular instant of time is given by the equation

$$dT / dx = -40x^3 + 72x^2 + 24x - 70$$

where 'x' is in metres. Write a C program to calculate the temperature profile (T as a function of x) for the range $0 \leq x \leq 0.5$ in steps of 0.05m. Given $T = 1000K$ at $x = 0$. Use Runge-Kutta fourth order method.

Algorithm

1. Start.
2. Declare the variables and initialize with required values.
3. Print the table and column headings.
4. Print the initial known value of distance, x_0 and corresponding temperature, y_0 .
5. Calculate the temperature for the next value of distance using Runge-Kutta method using the following formulas:
 $y_1 = y_0 + h$ where,
 $k = (k_1 + 2*(k_2 + k_3) + k_4)/6$
 $k_1 = f(x_0, y_0)$
 $k_2 = f((x_0 + h/2), (y_0 + k_1/2))$
 $k_3 = f((x_0 + h/2), (y_0 + k_2/2))$
 $k_4 = f((x_0 + h), (y_0 + k_3))$
6. Vary the loop variable, i as the range of distances, incrementing by the given step length, h .
7. Print the distance and corresponding calculated value of temperature.
8. Repeat steps 5,6 and 7 for the entire range of distances.
9. Define a function so as to calculate and return the value of the given differential equation when called during calculation.
10. Exit.

Program 5**Curve fitting: Method of least square**

- a. Write a program f vs Nre for laminar, turbulent regions by least square method. Calculate pressure drop in laminar flow using Hagen Poiseulle's equation.

Theory:

At low velocities fluid tend to flow without lateral mixing and adjacent layers slide past one another. These are neither cross currents nor eddies and is called laminar flow.

If the flow consists of cross currents and eddies and there is lateral mixing, it is called turbulent flow.

Reynold's number $N_{Re} = DV\rho/\mu$. It is a dimensionless number.

If $N_{Re} < 2100$. Laminar

$N_{Re} > 4000$ turbulent

'f' is Fannings Friction factor defined as the ratio of the wall shear stress to the product of density and velocity head $V^2/2$.

The relation between f and N_{Re} , $f = 16/N_{Re}$

Hagen Poiseuille's equation is $\Delta P_s = 32LV\mu/D^2$

f Friction Factor

N_{Re} Reynold's Number

ρ Density of Liquid

D Diameter of tube

V Average velocity of liquid

μ Viscosity of liquid

Algorithm:

1. Start
2. Input the number of valves of N_{Re} .
3. Input the valves of S and N_{Re} .
4. Set a loop to read the valves of f and N_{Re} .
5. Set a loop to check whether f and N_{Re} Valve he is Laminar / turbulent region.
6. Output the valves of f and N_{Re} .
7. Find the relation between f and N_{Re} . Using the function. /* function */

1. Start
2. Initialize fum x, sumy, sumxx, sumxy, to zero
3. Calculate the equation for f and N_{Re} .
4. Return the equation
5. Stop.

b. With the help of least square method, find value of A and E is Arrhenius Equation

$$K = Ae^{-E/RT}$$

Theory :

Let x be an independent variable and y a dependent variable $y=f(x)$. Suppose we are given n values y_1, y_2, \dots, y_n of variable y. Then the pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ give us n points in the xy plane. It is not possible to find the actual curve $y=f(x)$ that passes through these points. Hence we try to find a curve that serves as best approximation. This is called curve fitting.

The experimental data always contains errors. If the experimental points are plotted on a graph, instrumental errors causes the points to scatter about the true function. Suppose the function passes through $y=mx+c$, and if there were no mistakes then one could solve m and c easily. The method of least square takes into account all the experimental points by selection m and c so as to minimize the sum of squares of deviations.

$$m = (n \sum x_i y_i - \sum x_i \sum y_i) / (n \sum x_i^2 - (\sum x_i)^2)$$

N = Number of ordered pairs

$$C = \text{intercept} = \sum y_i - m \sum x_i / n.$$

Algorithm :

1. Start
2. Input the number of values
3. Input the values of $\ln k$ and $1/t$
4. Initialize
Sum x = sumy = sumxx = sumxy = 0
5. Calculate for n Number of turns
Sum $x + T(i)$
Sum $y += K[I]$
Sumxy += $T[I] * K[I]$
Sumxx ++ $T[I] = T(I)$
6. $M = (n * \text{sumxy} - \text{sumx} * \text{sumy}) / (n * \text{sumxx} - \text{sumx} * \text{sumx})$
7. Print slope
8. Print $K = A e^{-(E/RT)}$ Equation

- c) The free settling velocities of particles of different sizes are given in the following table. The data refers to settling in air at 1 atm and 20°C at a density difference of 5000 kg/m³.

Particle diameter (μm)	Velocity (mm/s)	Particle diameter (μm)	Velocity (mm/s)
5	3.871	55	378
10	14.87	60	439
15	33.47	65	500
20	58.22	70	560
25	59.31	75	625
30	125	80	695
35	168.3	85	759
40	215.3	90	823
45	266.4	95	890
50	320	100	957

**Write a program to fit an equation of the type $U_t = A(D_p)^B$ to the data.
Tabulate the values calculated from the equation along with the given data.
Plot the data using MS-Excel and determine regression coefficient.**

Algorithm

1. Start.
2. Declare the variables and initialize with required values.
3. Ask the user to enter the corresponding velocities in mm/s for the displayed diameter in micrometer.
4. Convert the given relation, $V_f = a(D_p)^b$ into $y=mx+c$ form by taking log on both sides:

$$\log V_f = b \log D_p + \log a$$

5. Vary loop variable, i for the given diameter range and increment.

Calculate $\log V_f$ and $\log D_p$ values.

Calculate sum of x terms, x^*y terms, y terms and x^2 terms using the following formulas:

$$\sum x = \sum \log D_p$$

$$\sum y = \sum \log V_f$$

$$\sum xy = \sum (\log V_f * \log D_p)$$

$$\sum x^2 = \sum \log D_p^2$$

6. Calculate regression coefficient, b and constant a, using least square method:

$$b = (n * \sum xy - \sum x * \sum y) / (n * \sum x^2 - (\sum x)^2)$$

$$a = e^{(\sum y - b \sum x) / n}$$

7. Print the entered old velocity and the new velocity found using:

$$V_f = a(D_p)^b$$

8. Stop.

Program 6

Using goalseek in MS Excel to approximate the roots of a non-linear algebraic equation

Determine the molar volume of n-butane at 500K and 18 atm. Data $T_c = 425.2\text{K}$, $P_c = 37.5$, $R=0.08206$.

The Equation of state to be used is Redlich Kwong equation

$$v^3 \cdot p - v^2 \cdot RT + v(a - pb^2 - RTb) - ab = 0$$

$$\text{where } a = 0.42748(R^2 T_c^2)/P_c \text{ and } b = 0.08664(R T_c/P_c)$$

Use MS-Excel. Plot compressibility factor for pressure range 50(50)500 atm.

Use goal seek command in MS Excel to estimate the molar volume of the stated cubic equation as follows:

REDLICH-KWONG EQUATION
N-BUTANE

PARAMETERS - n-butane		
TC	425.2	K
PC	37.5	ATM
R	0.08206	L.ATM/GMOL.K

DATA		
T	500	K
P	18	ATM

VAR	FORMULA		RESULT
TR	1.175917215		
A	12.79810579		
B	0.080614273		
F(V)	6.53945E-06		2.037689
IDEAL GAS VOL	2.279444444		
Z			0.893941

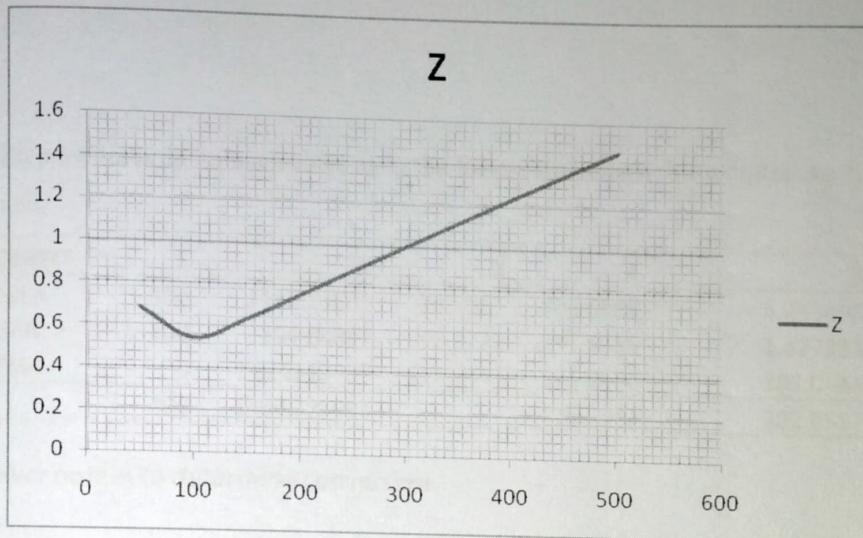
PARAMETERS - Ammonia		
TC	405.5	K
PC	111.3	ATM
R	0.08206	L.ATM/GMOL.K

DATA		
T	450	K
P	56	ATM

VAR	FORMULA		RESULT
TR	1.10974106		
A	4.036960975		
B	0.025902737		
F(V)	1.02601E-05		0.569801
IDEAL GAS VOL	0.659410714		
Z			0.864106

Generate compressibility factor data table and plot z for stated pressure range as follows:

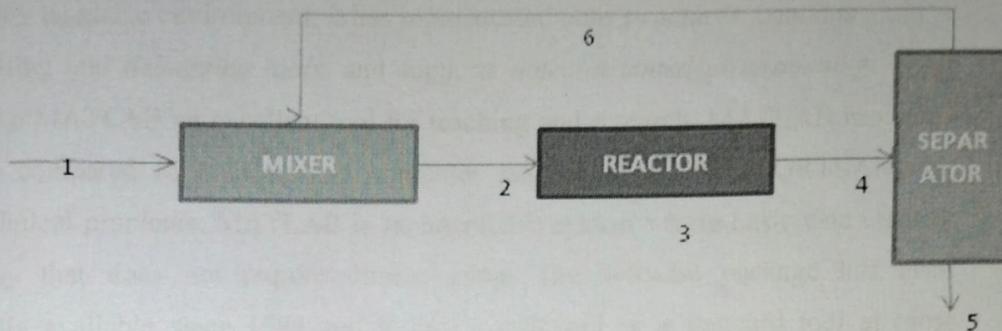
COMPRESSIBILITY FACTOR FOR A RANGE						
P	TR	A	B		f(v)	Z
50	1.175917	12.79811	0.080614	-2.4E-05	0.558545	0.680655
100	1.175917	12.79811	0.080614	-0.00046	0.221378	0.539552
150	1.175917	12.79811	0.080614	-0.00079	0.174154	0.636682
200	1.175917	12.79811	0.080614	-0.0005	0.155264	0.756832
250	1.175917	12.79811	0.080614	5.49E-05	0.144291	0.879182
300	1.175917	12.79811	0.080614	-0.0008	0.136734	0.99976
350	1.175917	12.79811	0.080614	1.06E-05	0.131256	1.119658
400	1.175917	12.79811	0.080614	0.000116	0.126942	1.237554
450	1.175917	12.79811	0.080614	-9.3E-06	0.123438	1.353815
500	1.175917	12.79811	0.080614	-0.0001	0.120526	1.468752



Program 7

Using solver in MS Excel to optimize

For the production of ammonia using nitrogen and hydrogen as the following flowsheet, determine the amount of all components in all streams using MS Excel. Conversion in reactor is 30%. 98% ammonia and 0.5% N₂ and H₂ are removed as a bottom product of separator.



Prepare material balance chart as in the following table writing equations for each cell.

COMPONENT	1	2	3	4	5	6
HYDROGEN	300	1182.26601	-295.567	886.6995	4.433498	882.266
NITROGEN	100	394.08867	-98.5222	295.5665	1.477833	294.0887
AMMONIA	0	4.021312959	197.0443	201.0656	197.0443	4.021313
TOTAL	400	1580.375993		1383.332	202.9557	

Use solver option to determine conversion

Part 2 Equilibrium conversion $K_p = p\text{NH}_3/(p\text{N}_2)^{1/2} \cdot (p\text{H}_2)^{3/2}$

Conversion not specified, $y\text{NH}_3/(y\text{N}_2)^{1/2} \cdot (y\text{H}_2)^{3/2}$

p 220 0.05 kp

CONV 0.740306

COMPONENT	1	2	3	4	5	6	MOL FR.
HYDROGEN	300	404.5284	-299.475	105.0536	0.525268	104.5284	0.30557
NITROGEN	100	134.8428	-99.8249	35.01788	0.175089	34.84279	0.101857
AMMONIA	0	4.074486	199.6498	203.7243	199.6498	4.074486	0.592574
TOTAL	400	543.4456		343.7958	200.3502		1

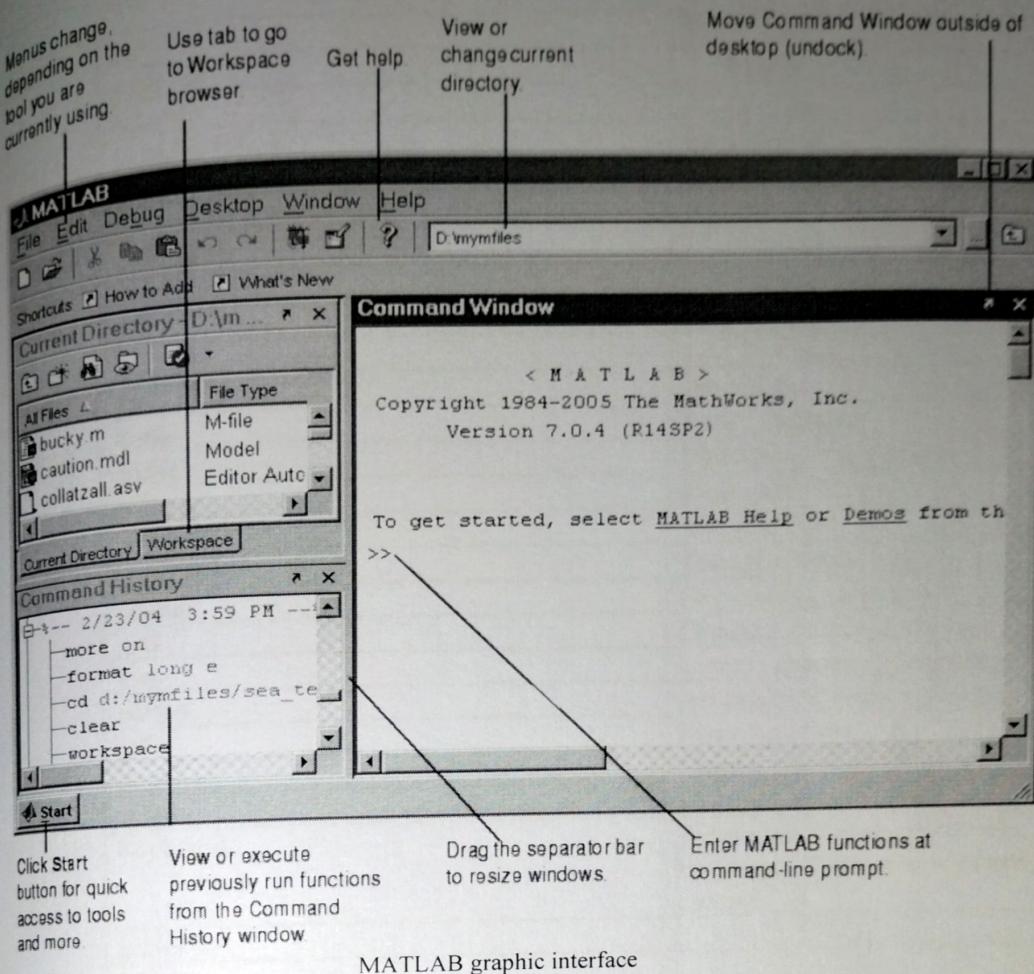
K 3.58E-05

MATLAB

Introduction: Then expansion of name MATLAB is MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects. It is a high-performance language for technical computing. It integrates *computation*, *visualization*, and *programming* environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated *data structures*, contains built-in editing and *debugging tools*, and supports *object-oriented programming*. These factors make MATLAB an excellent tool for teaching and research. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an *array* that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide. It has powerful *built-in* routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as *toolbox*. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

Starting MATLAB: After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut *icon* (MATLAB) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains *other* windows. The major tools within or accessible from the desktop are:

- The Command Window
- The Command History
- The Workspace
- The Current Directory
- The Help Browser
- The Start button



All MATLAB functions and evaluations are entered at the command prompt ‘>>’ in command window. History of all inputs at command prompt are presented in command history window. To write M-files, open a new file, type the statements and save the file with extension “.M”. Execute them in the command window.

Common MATLAB inbuilt functions

General

eps	2.22e-16	
realmax	1.79e308	
realmin	2.22e-308	
rem(23,4) = 3	Remainder	
acos	Inverse cosine	
cos, sin, cot, cosec, sec		
sinh, cosh		
exp		
log	Natural log	
log10	Log base 10	
abs	Absolute	
imag	Imaginary part	
angle	Phase angle in radians	
real	Real part	
isreal	True for real	
isprime	True for prime nos	
primes	Generates prime nos	
gcd	Greatest common divisor	
lcm	Lowest common multiple	
perms	All possible combinations	
erf	Error function	
disp(x)	Displays variable x	
input	Prompts user to input	
keyboard	Temporarily gives control to keyboard. Type return to return control	
pause		
input		
waitforbuttonpress		
%	Comment single line	
%{ }%	Comment multiple lines in braces	
islogical	Returns 1 if true	
isnumeric	Returns 1 if true	
class(x)	Returns logical, double, etc depending on variable type	
Arrays	x=linspace(0,pi,11)	0 to pi divided in 11 points
	(1:10)', x'	Transpose
	g.*h	Every element is multiplied in the two arrays
	ones(2)	1 1 1 1
	zeros(2,3)	0 0 0 0 0 0
	eye(2)	1 0 0 1
	rand(1,3)	Random nos – 1 row 3 columns
	randn(2)	Random nos from normal distribution (0 mean, unit variance)

	a=1:4 diag(4) repmat(p,2,3) reshape(d,2,3) a(:,4)	2x2 random nos in matrix 1,2,3,4 in diagonal elements of matrix Replicates matrix p, in 2x3 matrix elements Reshape vector d in to 2x3 matrix All rows, fourth column
	randperm(4) sort(x)	Random 4 nos Sorting in ascending order by default
	xs=sort(x, 'ascend') [xs, idx]=sort(x)	Ascending order Sorts in ascending and stored in xs Idx stores the index values of the numbers
	k=find(a>4) find(x>4,1) find(x>4, 'last')	Finds elements greater than 4 in matrix A Finds first value greater than 4 Finds last value greater than 4
	max(a) min(a)	Maximum value in A Minimum value in A
	triu(a) tril(a)	Upper triangular matrix Lower triangular matrix
	numel(a)	Total nos of elements in array
	x=length(a)	No of elements along largest dimension
	size(a)	No of row no of column
Multi-D arrays	ndims(a)	No of dimensions
Cell array	a(1,1)={[1 2 3; 4 5 6]} a(1,2)={2+3i} a(2,1)={"name"} a(2,2)={12:-2:0}	Each location is called a cell. To display use >>celldisp(A) >>cell(1,2)
Structures	circle.radius=2 circle.diameter=4 circle.linestyle='—' circle.color='red'	Structure creation >>circle – will give all info about circle
Strings	fprintf sprintf	Both are same. Sprint creates a character array that can be displayed. Similar to 'C'
	sscanf(v,'%f') sscanf(x,'%d') sscanf(x,'%s') char(c)	Get floating point Get integer Get string character
Logical	& && ~=	Element by element AND in arrays Scalar And Element by element OR Scalar OR for short circuiting Not equal to
Control	for, if elseif else, while,	Like 'C'

flow	try –catch (error), switch	Function ends with 'end', switch as 'otherwise'
Functions	error	Used to abort function If (length(x)<4) Error('x is smaller') end
	inline	$Y=Inline('1/(1+x)', 'x')$ $P=feval(Y, [-1 0 1])$ Not used much
	anonymous	$Y=@(x) 1/(1+x)$
Matrix	rank(a), rank([a,y]) x=a\y or y/a	A.x=y, $X=inv(A)*y$
	det(a)	Determinant of A
	eig(a)	Vector of eigen values
	rref(a)	Reduced row echelon form
Data analysis	mean(x)	average
	sdev=std(x)	Std deviation
	median(x)	median
	cov(x)	covariance
	corrcoef(x)	Correlation coefficient
	var(x)	Variance
	prod(x)	Product
Data interpolation	interp1(hz,db,2e3) interp1(hz,db,2e3, 'linear')	Interpolation of 1D array. At 2kHz what will be the dB level will be obtained by Linear interpolation.
	interp1(hz,db,2e3, 'cubic')	Cubic interpolation
	interp1(hz,db,2e3, 'spline')	Spline – most accurate but takes time
	interp1(hz,db,2e3, 'nearest')	Nearest neighbor – crudest, returns data point nearest the given value.
	zi=interp2(x,y,z,2.2,3.3)	At x=2.2 and y= 3.3, what is z value, linear interp.. Ex; ocean depth values when data at different points are available. X=0:0.5:4 z=[100 99 100..... Y=0:0.5:6]
Polynomial s	p=[1 2 3 4]	$P=x^3+2x^2+3x+4$
	r=roots(p)	Finds roots of polynomial 'p' Polynomials are row vectors, roots are column vectors
	p=poly(r)	Constructs polynomial 'p' using roots 'r'
	c=conv(a,b)	Product of two polynomials
	[q,r]=deconv(a,b)	Deconvolution or division, q-quotient, r-remainder
	d=polyder(g)	Differentiation of 'g' polynomial
	polyint(g)	Integrates polynomial

	<code>polyval(p,x)</code>	Evaluate polynomial 'p' at x
	<code>polyfit(x,y,n)</code>	Fitting a curve to polynomial for x and y data arrays. 'n' is order, if n=2, quadratic curve is fit
Optimization	<code>[x,value]=fzero(humps,[-2 0])</code>	X is the root, value gives the function value at x in the range -2 to 0.
	<code>options=optimset('display','iter','tolx,0.1)</code>	<code>[x,value]=fzero(H-humps,[-2 0], options)</code>
	<code>[xmin, value]=fminbnd(humps, 0.5,0.8)</code>	Minimization
Integration	<code>area=trapz(x,y)</code>	Trapezoidal method
	<code>cumtrapz(x,y)</code>	Cumulative area using trapezoidal
	<code>quad(func,-1,2), quadl(func,0,2)</code>	Quadrature function, quadl is more rigorous
	<code>area=dblquad(fn,0,pi,-pi,pi)</code>	Double integral
	<code>triplequad(xmin,xmax,ymi,ymax,zmin,zmax)</code>	Triple integral – volume integration
Differentiation	<code>p=polyfit(x,y,n)</code>	For a given data, fit a curve and find derivative using polyder
	<code>pd=polyder(p)</code>	
Differential equation	<code>dy=diff(y)./diff(x)</code>	Method of finite differences to find derivative
	<code>ode45(func,x,y0)</code>	4-5 order R-K method
	<code>ode23</code>	Euler method
	<code>ode113</code>	Adam-Bashforth-Moulton PECEsolver
		First create a M-file for the function as <code>ydot=odefun(t,y)</code> and then <code>[t,y]=ode45(@func,t,y0)</code>
	<code>pdepe</code>	Solves IVP for parabolic-elliptic PDEs in one -D
	<code>pdeval</code>	Evaluates/interpolates the solution found using pdepe
Graphics-2D	<code>bvp4c,bvpget,bvpinit,bvpset,deval</code>	For boundary value problems
	<code>plot(x,y)</code>	
	<code>plot(x,y,'style')</code>	Style – color : b,g,r,c,m,y,k,w Marker:o,x,+,*s,d,v, ^{^<>} ,p,h Linestyle: -, : , -. , --
	<code>grid on</code>	Grids
	<code>axis(xmin,xmax,ymin,ymax)</code>	Sets axes limits
	<code>axis on, axis off</code>	
	<code>subplot(2,2,i)</code>	
	<code>title('figure')</code>	
	<code>legend('sin(x)', 'cos(x)')</code>	
	<code>xlabel, ylabel</code>	
	<code>zoom, zoomout, zoom off</code>	
	<code>semilogx, semilogy, loglog</code>	
	<code>fill(x,y,'r')</code>	
	<code>text(0,0,'stop','color'..fontsize..fontweight..)</code>	

	horizontalalignment..) pie bar(x,y), bar3(x,y,'r'), stairs(x,y), barh(x,y) hist(y,x), stem(z,'—') errorbar(x,y,e) compass(z), feather(x), rose(v) scatter(x,y,area) \alpha, \beta, \gamma, \theta, \mu, \pi, \rho, \sigma, \tau \int, \omega, \approx, \leftarrow, \rightarrow, \uparrow, \partial, \pm, \neq	
Graphics 3-D	plot3(x,y,z) mesh(x,y,z) meshc(x,y,z) meshz(x,y,z) waterfall(x,y,z) surf(x,y,z) shading flat, shading interp surfl(x,y,z) surfnorm(x,y,z) rotate3d, rotate3d off contour(x,y,z,20) contour3(x,y,z,20) contourf(x,y,z12) clabel,pcolor(x,y,z) ribbon(z) quiver(x,y,dx,dy)	Prints symbol Mesh with contour plot Mesh with zero plane Mesh line appear in x direction Surface plot like mesh but filled inbetween lines Surface plot with lighting With normal 20 contour lines Filled contour Velocity vectors(dx,dy) at points x,y [x,y,z]=peaks(16) [dx,dy]=gradient(z,.5,.5) Contour Quiver
	stem3(z,'ro','filled')	Stem plot
	slice(x,y,z,v,[0,3],[5,15],[-3,5])	For volume visualization slices

Programs 8 and 9

Bubble and Dew calculations using MATLAB

Raoult's law: It states that the equilibrium partial pressure of a component in gas/vapor phase is equal to the product of its mole fraction in liquid phase and pure component vapor pressure.

$$P y_i = P_i^{\text{sat}} x_i$$

Calculations Using Raoult's Law

Bubble-point pressure problem -- T, x given -- P, y unknown.

This is a straight-forward calculation. The vapor pressures are found at the given temperature, which allows direct calculation of the pressure and vapor mole fractions:

$$\begin{aligned} \sum P y_i &= P = \sum P_i^{\text{sat}} x_i \\ y_i &= \frac{P_i^{\text{sat}} x_i}{P} \end{aligned}$$

Bubble-point temperature problem -- P, x given -- T, y unknown.

A trial and error procedure must be followed, where T is assumed, the vapor pressures calculated, and then see if the correct total pressure is obtained (or the vapor mole fractions sum to unity).

$$\sum y_i = 1 = \sum \frac{P_i^{\text{sat}} x_i}{P}$$

Dew-point pressure problem -- T, y given -- P, x unknown.

No trial and error is needed, as P can be directly calculated.

$$\begin{aligned} x_i &= \frac{P y_i}{P_i^{\text{sat}}} \quad \sum x_i = 1 \\ P &= \frac{1}{\sum (y_i / P_i^{\text{sat}})} \end{aligned}$$

Dew-point temperature problem -- P, y given -- T, x unknown.

Guess a T , find the vapor pressures, and see if the liquid mole fractions sum to unity.

$$\sum x_i = 1 = \sum \frac{P y_i}{P_i^{\text{sat}}}$$

Discuss calculation of a $y-x$ diagram for this system at fixed P .

K-values, flash calculations, and relative volatilities

The T - x - y diagram It represents all possible dew and bubble points for this system at the pressure given. The P - x - y diagram It represents all possible bubble and dew points for this system at the T given. A vapor-liquid equilibrium ratio (K -value) is defined for a component as the ratio of mole fraction in the vapor to mole fraction in the liquid for that component. The criteria for flash calculations can be written in various forms. A useful one is given below.

$$K_i = \frac{y_i}{x_i} = \frac{P_i^{sat}}{P} = f(P, T)$$

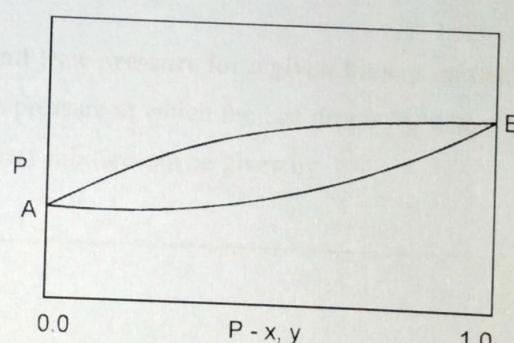
$$\sum \frac{z_i(K_i - 1)}{1 + V(K_i - 1)} = 0$$

T , P , and feed compositions (z_i 's) are known. A trial and error for V must be done (fraction to vapor stream). According to Raoult's law, the relative volatility (defined below) is just the ratio of component vapor pressures.

$$\text{Relative volatility: } \alpha_{ij} = \frac{y_i / x_i}{y_j / x_j} = \frac{K_i}{K_j} = \frac{P_i^{sat}}{P_j^{sat}}$$

(a) Write a program to find Bubble Pressure for a given binary mixture.

Theory : Bubble pressure is the pressure at which the first drop of a liquid vapourises. The P-x,y diagram for a given binary mixture can be given by



Notation:

P	-	Bubble Pressure
P ₁	-	Partial Pressure
P _{1S}	-	Saturated Vapour Pressure
X	-	Mole fraction in liquid phase
Y	-	Mole fraction in gaseous phase
T	-	Temperature of the system.

Equations and formulae

$$P_1 = P_1^S \cdot X_1$$

$$P_1^S = \exp \left[A_1 - \frac{B_1}{C_1 + T} \right]$$

$$P = \sum_i P_i$$

$$\gamma_1 = \frac{P_1}{P} = \frac{X_1 P_1^S}{P}$$

Algorithm :

1. Start
2. Accept the values of X_i, T
3. Accept the Antoine's constants
4. Calculate

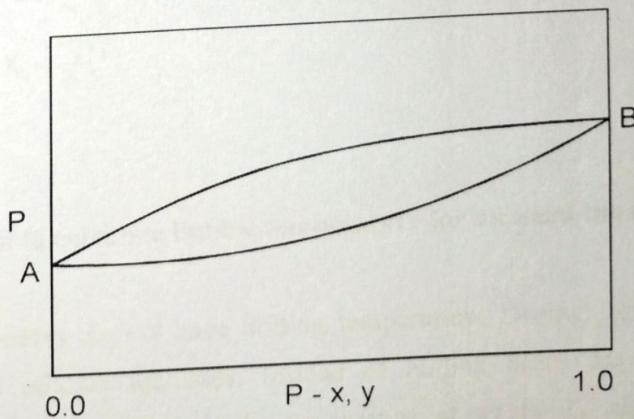
$$P_1^S ; \quad P_1^S = \exp \left[A_1 - \frac{B_1}{T + C_1} \right]$$

$$P_2^S ; \quad P_2^S = \exp \left[A_2 - \frac{B_2}{T + C_2} \right]$$

5. Calculate $P_1 = X_1 P_1^S$ and
 $P_2 = (1-X_2) P_2^S$
6. Calculate $P = P_1 + P_2$
7. Calculate P_1 and P_2
8. End.

(b) Write a program to find Dew pressure for a given binary mixture.

Theory: Dew pressure is the pressure at which the last drop of the liquid vapourises. The $P-x,y$ diagram for a given binary mixture can be given by



Notation:

P_D	-	Dew Pressure
P_1	-	Partial Pressure
P_1^S	-	Saturated vapour pressure
X	-	Mole fraction in liquid phase
Y	-	Mole fraction in gaseous phase
T	-	Temperature of the system.

Equation and Formulae

$$P_1^S = \exp \left[A_1 - \frac{B_1}{T+C_1} \right]; \quad P_2^S = \exp \left[A_2 - \frac{B_2}{T+C_2} \right]$$

$$P = \frac{1}{\frac{Y_1}{P_1^S} + \frac{Y_2}{P_2^S}}$$

$$X_1 = \frac{P_d Y_1}{P^S}$$

Algorithm :

1. Start
2. Accept the values of y and T
3. Accept the Antoine's Constant
4. Calculate $P_1^S = \exp \left[A_1 - \frac{B_1}{T+C_1} \right]$

$$P_2^S = \exp \left[A_2 - \frac{B_2}{T+C_2} \right]$$
5. Calculate $P = \frac{1}{\frac{Y_1}{P_1^S} + \frac{Y_2}{P_2^S}}$
6. Calculate $X_1 = \frac{P_d Y_1}{P^S}$

$$X_2 = \frac{P_d Y_2}{P^S}$$
7. End.

(c) Write a program to calculate bubble temperature for an ideal binary mixture.

Theory : Liquids mixtures do not have boiling temperature. During vapourisation the temperature of liquid mixture increases. Instead of boiling point they have bubble temperature. During vapourisation. It is the temperature at which the first drop of the

mixture vapourises. Similarly during condensation of a gaseous mixture at dew temperature it condenses to form first drop of liquid.

Formulae :

Raoult's Law :

$$Y_i = \frac{x_i p_i^s}{P}$$

Antoine's Equation :

$$\ln P_{1s} = A_i - \frac{B_i}{C_i + T}$$

$$T_g = \sum x_i T_i$$

$$P_s = \frac{P}{x_i x_{ij} + (1-x_i)}$$

$$x_{ij} = \exp \left[A_i - \frac{B_i}{C_i + T_g} - A_j + \frac{B_j}{C_j + T_g} \right]$$

$$\text{relative error} = \left[\frac{T_g - T_m}{T_m} \right]$$

Algorithm :

1. Assume $X_1 = 0$ and $X_2 = 1$
2. Evaluate T_1, T_2 from Antoine's equation.

$$T_1 = \frac{B_1}{A_1 - \ln P_1^s} - C_1 \quad ; \quad T_2 = \frac{B_2}{A_2 - \ln P_2^s} - C_2$$

by substituting $P_1^s = P$ and $P_2^s = P_1$ as substance boils at vapours pressure equals total pressure.

3. Calculate T_g ; $T_g = \sum x_1 T_1$
4. Using T_g calculate X_{12} I

$$X_{12} = \exp \left[A_1 - \frac{B_1}{C_1 + T_g} - A_2 + \frac{B_2}{C_2 + T_g} \right]$$

5. Using X_{12} calculate P_2^s

$$P_2^s = \frac{P}{X_1 X_{12} + (x - x_1)}$$

- Calculate T_{mix} from above equation
6. Evaluate relative error $\leq acc$; Calculate Point A. (P_1^S) using Antoine's equation and
 7. T_m
 8. Calculate y_i from P_1^S .
 9. Repeat steps for different X_1 values

(d) Write a program to calculate dew temperature for an ideal binary mixture.

Theory: Liquids mixtures do not have boiling temperature. During vapourisation the temperatures of liquid increases. Instead of boiling temperature, they have bubble temperature and dew temperature. During vapourisation bubble temperature is the temperature at which the first drop of the liquid mixture vapourises. Similarly during condensation dew temperature is the temperature at which the first drop of condensate forms.

Formulae :

Raoult's Law :

$$X_i = \frac{y_i P}{P_1^S}$$

Antoine's Equation :

$$\ln P_1^S = A_1 - \frac{B_1}{C_1 + T}$$

$$T_g = \sum y_i T_g$$

$$P_1^S \frac{P}{Y_i \alpha_{12} + (1 - y_1)}$$

$$X_{12} = \exp \left[A_1 - \frac{B_1}{C_1 + T_g} - A_2 + \frac{B_2}{C_2 + T_g} \right]$$

$$\text{Relative error} = \left[\frac{T_g - T_m}{T_m} \right]$$

Algorithm :

1. Assume $Y_1=0$ and $Y_2=1$
2. From Antoine's Equation calculate T_1 and T_2

- $$T_1 = \frac{B_1}{A_1 - \ln P_1^S} - C_1 \quad ; \quad T_2 = \frac{B_2}{A_2 - \ln P_2^S} - C_2$$
3. Calculate T_g , $T_g = \sum y_i T_i$
 4. Using T_g calculate X_{12}
 5. Using X_{12} , Calculate P_2^S
 6. $P_2^S = \frac{P}{Y_1 X_{12} + (1-y_1)}$
 7. Calculate relative error, $\frac{T_g - T_m}{T_m} \leq \text{acc.}$ Calculate P_2^S using Antoine's equation and T_m
 8. Calculate X_2 from P_2^S
 9. Calculate the above for different values of Y_1 and Y_2 .

(e) For acetone(1)/ acetonitrile(2)/ nitromethane(3) system, the Antoine's equations are

$$\ln P_1^S = 14.5463 - 2940.46/(t+237.22); \quad \ln P_2^S = 14.2724 - 2945.467(t+224.0); \quad \ln P_3^S = 14.2043 - 2972.64/(t+209.00)$$

where t is in $^{\circ}\text{C}$, P is in kPa. Assuming Raoult's is valid Write a MATLAB function to calculate

- (i) P and y_i , given $t=80^{\circ}\text{C}$, $x_1 = 0.25$, $x_2=0.35$ and $x_3=0.4$
- (ii) t and y_i , given $P=80\text{kPa}$, $x_1=0.3$, $x_2=0.45$, $x_3=0.25$

(f) For acetone(1)/ acetonitrile(2)/ nitromethane(3) system, the Antoine's equations are

$$\ln P_1^S = 14.5463 - 2940.46/(t+237.22); \quad \ln P_2^S = 14.2724 - 2945.467(t+224.0);$$

$$\ln P_3^S = 14.2043 - 2972.64/(t+209.00)$$

where t is in $^{\circ}\text{C}$, P is in kPa. Assuming Raoult's is valid write a MATLAB function to calculate

- (i) P and x_i , given $t=70^{\circ}\text{C}$, $y_1=0.5$, $y_2=0.3$ and $y_3=0.2$
- (ii) t and x_i , given $P=90\text{kPa}$, $y_1=0.6$, $y_2=0.2$ and $y_3=0.2$

Program 10**Flash vaporization calculations**

Flash vapourisation or equilibrium distillation is a single stage operation where in a liquid mixture is partially vapourised the vapour allowed to come equilibrium with the residual liquid and the resulting vapour and liquid phases are separated and removed. It may be a batch wise or continuous. In flash vapourisation an entering stream preferably liquid at high pressure flashes as it comes out of a flash drum forming two phases.

Notation:

F	Flow rate of feed
L	Flow rate of liquor
V	Flow rate of vapour
X_i	Mole fraction of Component I in liquor
Y_i	Mole fraction of component I in liquor
Z_i	Mole fraction of component I in feed

Material Balance:

$$F = L + V$$

$$FZ_i = Lx_i + Vy_i$$

$$Z_i = L \frac{x_i}{F} + \frac{V y_i}{F}$$

$$Z_i = L \frac{y_i}{k_i} + Vy_i \quad [\text{For } F = 1 \text{ mol}]$$

$$Z_i = Y_i \left[\frac{1}{K_i} + V \right]$$

$$Y_i = \frac{Z_i K_i}{1 + V(1 - K_i)}$$

$$\sum Y_i = \sum \left[\frac{Z_i K_i}{1 + V(K_i - 1)} \right]$$

$$f(V) = \sum \left[\frac{Z_i K_i}{1 + V(K_i - 1)} \right] - 1$$

Formulae :

$$1. K_i = \frac{Y_i}{X_i} = \frac{P_1^S}{P}$$

$$2. \ln P_1^S = A - \frac{B}{T + C}$$

$$3. P_B = \sum X_i P_1^S$$

$$P_D = \frac{Y_i}{\sum \left[\frac{Y_i}{P_1^S} \right]}$$

$$4. Y_i = \frac{X_i P_1^S}{P_B}$$

$$X_i = \frac{P_D Y_i}{P_1^S}$$

$$5. V_G = \frac{P_B - P}{P_B - P_D}$$

$$6. F(V) = \sum \left[\frac{Z_i K_i}{1 + V(K_i - 1)} \right] - 1$$

Algorithm :

1. Input the constants $a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3, \dots$
2. Input Z_i and T

3. Calculate P_1^S , P_2^S , P_3^S , from Antoine's equation
4. Calculate bubble pressure taking $Z_i = X_i$, P_B
5. Calculate dew pressure taking $Z_i = Y_i$, P_D
6. Calculate total pressure $P = \frac{P_B + P_D}{2}$
7. Check for conditions $P_B > P > P_D$
 - a. If conditions satisfies, proceed
 - b. If condition does not satisfy, stop
8. If condition is satisfied
Calculate equilibrium constant K_i at different composition
- $K_1 = \frac{P_1^S}{P}$, $K_2 = \frac{P_2^S}{P}$, $K_3 = \frac{P_3^S}{P}$,
9. Calculate V_1 , Using initial guess
$$V_o = \frac{P_B - P}{P_B - P_D}$$
10. Calculate V from above equation using Newton Raphson method.
11. Calculate $L = 1 - V$
12. Calculate vapour composition y_i and liquid x_i
13. Check $\sum x_i = 1$, $\sum y_i = 1$.

A system acetone(1)- acetonitrile(2) – nitromethane (3) at 80°C and 110 kPa has the overall composition $z_1 = 0.45$, $z_2 = 0.35$, $z_3 = 0.2$. Write a MATLAB code to determine L, V, x_i and y_i if this mixture is flash vaporized.

Component	A	B	C
1	14.5463	2940.46	237.22
2	14.2724	2945.47	224.0
3	14.2043	2972.64	209.0

Program 11

Write a M-file to calculate residence time of a batch reactor by using simpson's 1/3 method.

The saponification reaction $\text{NaOH} + \text{CH}_3\text{COOC}_2\text{H}_5 \rightarrow \text{CH}_3\text{COONa} + \text{C}_2\text{H}_5\text{OH}$ is a second order irreversible reaction. A stirred batch reactor is charged with an aqueous solution containing NaOH and ethyl acetate both at initial concentration of 0.1 M. After 15 minutes the conversion of ethyl acetate is 18%. For initial charge concentration of 0.2 M of both reactants, write a program to determine the time required for 30% conversion using a suitable numerical technique?

Theory:

In a batch reactor, the reactants are initially charged into a container, are well mixed, and are left to react for a certain period. The resultant mixture is then discharged.

The performance equation for an ideal batch reactor can be given by

$$t = C_{AO} \int_0^{X_A} \frac{dX_A}{(-r_A)} = \int_{C_{AO}}^{C_A} \frac{dC_A}{(-r_A)} = N_{AO} \int_0^{X_A} \frac{dX_A}{(-r_A)V}$$

Notations:

T	-	residence time
C_{AO}	-	Initial concentration
C_A	-	Final concentration
$(-r_A)$	-	rate of reaction
X_A	-	Conversion
V	-	Volume of reactor.

Algorithm:

1. Accept initial and final conversions (x_0, x_f)
 2. Accept initial concentration and reaction constant
 3. Accept No. of segments n
 4. for $n \% 2 = 0$,
- $h = (x_f - x_0) / n$

- $f_1 = F(x)$, $f_2 = F(x+h)$, $f_3 = F(x+2*h)$
5. Calculate sum, $\text{sum} = \text{sum} + f_1 + 4 * f_2 + f_3$.
 6. Accept $f_1 = f_3$ and $x = x + 2 * h$ and keep iterating
 7. Calculate ics , $ics = (h/3) * \text{sum}$.
 8. Calculate t , $t = C_{AO} * ics / K$
 9. If $n \% 2 \neq 0$, repeat from step 3.
 10. Stop.

Program 12

Write a program to calculate the volume for a given plug flow reactor.

Theory:

In a plug flow reactor, the reactants are fed and it is characterized by the fact that the flow of the fluid through the reactor is orderly with no element of fluid overtaking or mixing with any other element ahead or behind. There may be lateral mixing of fluid in a plug flow reactor, and there must be no missing or diffusion along the flow path. In this kind of reactor the residence time in the reactor to be the same for all elements of fluid. This kind of reactor is also called as slug flow, tubular flow, piston flow and unmixed flow reactor.

The performance equation for plug flow reactor can be given by

$$V = F_{AO} \int_0^{X_A} \frac{dX_A}{(-r_A)} = \frac{F_{AO}}{C_{AO}} \int_{C_{AO}}^{C_A} \frac{dC_A}{(-r_A)}$$

Notation:

V	-	Volume of reactor
F_{AO}	-	Flow rate of feed
C_{AO}	-	Initial concentration of feed
X_A	-	Conversion
$(-r_A)$	-	Reaction rate.

Algorithm:

1. Accept the feed rate and conversion
2. Accept the reference temperature and initial temperature
3. Accept the initial pressure
4. accept the enthalpies of a,b,c
5. accept the values of alpha, beta and gamma for a,b,c
6. Calculate $C_{AO}, C_{AO} = \frac{Px1000.0}{8.314 * T}$
7. Calculate $V_o, V_o = \frac{F_{AO}}{C_{AO}}$
8. Calculate $\Delta H_r, \Delta H_r = \Sigma (H_r \text{ of products}) - \Sigma (H_r \text{ of reac})$

$$\Delta \alpha = \Sigma (\text{alpha of prod}) - \Sigma (\text{alpha of reac})$$

$$\Delta \beta, \text{del beta} = \Sigma (\text{beta of prod}) - \Sigma (\text{beta of reac})$$

$$\Delta \gamma = \Sigma (\text{gamma of prod}) - \Sigma (\text{gamma of reac})$$

9. Calculate a, $a = \text{alpha} * +\text{del alpha}(\text{to}-\text{t}) + \frac{\text{del beta}}{2.0} * (\text{to}^3 - \text{t}^3)$

10. Calculate b, $b = \text{del H}_r * + \text{del alpha}(\text{to}-\text{t}) + \frac{\text{del beta}}{2.0} * (\text{to}^2 - \text{t}^2)$

11. Calculate x, k, f

$$X=a/b, K=\exp \left(34.34 - \frac{34222}{t} \right), f = \left(\frac{to}{t} \right) * \left(\frac{1.0+x}{K*(1.0-x)} \right)$$

12. Calculate sum, sum=sum+f

13. Calculate h, $h = \frac{5.0}{\text{to} - \text{t}}$

14. Calculate V, $V=V_0 * \left(\frac{h}{3} \right) * \text{sum}$

15. Stop.

Program 13

Write a program to calculate the volume required for a constant stirred tank reactor system.

Theory: In a constant stirred tank reactor, the reactants are fed continuously and are stirred well so that the composition is uniform throughout. This type of system is also called as mixed reactor, backmix reactor or stirred tank reactor.

The performance equation for this kind of reactor can be given by

$$V = F_{AO} \frac{\Delta X_A}{(-r_A)}$$

Here let the reaction taking place be $A + B \rightarrow C$

Formulae:

$$(-r_A) = K C_{AO} (1 - X_A)$$

$$K = K_{exp} \frac{E \left[\frac{1}{T} - \frac{1}{T_0} \right]}{R}$$

$$T = \frac{T_0 C_p^{-1} - \Delta H_R (T_R) X_A + \Delta C_p T_R X_A}{C_p^{-1} + \Delta C_p X_A}$$

$$C_p^{-1} = \sum C_{PA}$$

$$\Delta C_p = \sum (C_p \text{ of products}) - \sum (C_p \text{ of reactants})$$

Algorithm:

1. Accept the feed rate and conversion ($F_{AO} - X_A$)
2. Accept the initial and reference temperature (T_0, T_R)
3. Accept the values of $K_0, E, \text{ gas constant } (R)$
4. Accept the specific heat of a,b,c. (C_{PA}, C_{PB}, C_{PC})
5. Accept the enthalpy of a,b,c. (h_a, h_b, h_c)
6. Calculate $dH_R(T_R), dH_r(T_r) = hc - ha - hb$
 $dC_p, dC_p = C_{PC} - C_{PA} - C_{PB}$
 $C_{PI}, C_{PI} = C_{PA} = C_{PB} + C_{PC}$
7. Calculate $T, T = \frac{T_0 C_{PI} - dH_r(T_r) * X_A + dC_p * T_R * X_A}{C_{PI} + dC_p * X_A}$
8. Calculate $K, K = K_0 * \exp(-E/R) * (1/T - 1/T_0)$
9. Calculate $V, V = (F_{AO} * X_A) / (K * (1 - X_A))$
10. Stop.

Program 14

Write a program to find heat transfer area and no. of hairpins required for a double pipe heat exchanger.

2900 kg/hr of toluene is to be cooled from 71°C to 37°C by heating amylo acetate from 32°C to 37°C using 6000mm hairpins. The exchanger are 2" x 1 ¼" diameter allowing 0.7 kgf/cm² and providing a maximum dirt factor of 0.0082 hr.m².°C/kcal. Write a MATLAB code to determine the number of hairpins required.

Theory: A double pipe heat exchanger is a simple heat transfer equipment which is assembled of a standard metal pipe and standardized return bends and return heads, the latter equipped with stuffing boxes. One fluid flows through the inside pipe and the other fluid through the annular space between the inner and outer pipe. the function of a heat exchanger is to transfer the heat from hot fluid to cold fluid. Double pipe heat exchanger is mainly useful when not more than 14 m² of heat transfer surface is required.

Formulae:

$$Q = mC_p \Delta T$$

$$M = \delta V A$$

$$(\Delta T_{LMTD}) = \frac{\Delta T_1 - \Delta T_2}{\ln \left[\frac{\Delta T_1}{\Delta T_2} \right]}$$

$$N_{Re} = \frac{\rho DV}{\mu}, N_{Pr} = \frac{C_p \mu}{K}$$

$$N_{Nu} = 0.023 \times (N_{Re})^{0.8} \times (N_{Pr})^{0.33}$$

$$Di = \frac{d_i^2 - d_o^2}{d_o}$$

$$\frac{1}{U_i} = \frac{1}{h_i} + \frac{1}{h_o} \times \frac{d_o}{d_i} + \frac{X_w}{K} \ln \left(\frac{d_o}{d_i} \right)$$

$$Q = U_i A \Delta T_{LMTD}$$

Algorithm:

1. Find the average temperature of cold and hot fluid.
2. At these average temperature find C_p , μ , δ , K for the two liquids.
3. Calculate heat load $Q, Q = MC_p \Delta T$

4. Calculate LMTD, $(\Delta T)_{LMTD} = \frac{\Delta T_1 - \Delta T_2}{\ln\left(\frac{\Delta T_1}{\Delta T_2}\right)}$
5. Find NRC, N_{pr} , N_{nu} for both liquids
6. Find h_i for inner fluid and h_o for outer fluid
7. Calculate U_i
8. Calculate the area required A , $A = \frac{Q}{U_i \Delta T_{LMTD}}$
9. Calculate total length of DPHE and then find No. of tubes required by assuming standard pipe length.
10. Calculate number of bends and number of hairpins.

Questions for Viva

1. What is Numerical Computing?
2. What are traditional numerical computing methods?
3. What are characteristics of numerical methods?
4. What is convergence?
5. What are numerical errors?
6. How equations are classified?
7. What is Transcendental equation?
8. Explain different methods for solving non-linear equation?
9. What are different iterative methods?
10. Define LMTD . Can it be zero?
11. Define flash vaporization? Can water be flashed?
12. Write performance equation for PFR?
13. Can you separate kerosene and water by flash vaporization?
14. What are equations of state? Mention some equation of state other than ideal gas law .
15. Draw P-x, y and T-x, y diagram.
16. Draw the concentration profile for PFR.
17. Write performance equation of CSTR, PFR and Batch reactor.
18. Write the temperature profile for both cocurrent and counter current form in a DPHE.
19. Where does the binary mixture of Liquid-vapor lies in P-x, y ant T-x, y diagram.
20. How does the temperature influence the CSTR and PFR?
21. What is bubble temperature and dew temperature?
22. Draw a block diagram of distillation column with reboiler and condenser?
23. Write Fenske equation?
24. What is flash vaporization?
25. Mention the condition for flash vaporization.
26. State the criteria for phase equilibria.
27. State the criteria for chemical equilibria.
28. Difference between boiling point and bubble point.
29. Write Redlich Kwong equation?
30. What is two parameter and three-parameter model?
31. What is the difference between distillation, evaporation and flash vaporization?
32. Define non-ideal binary mixture.
33. Define Relative volatility?
34. What are azeotropes?
35. Is adiabatic CSTR is steady state or unsteady state reactor.
36. How would the temperature profile look if one fluid is vaporizing and other fluid is condensing? Since ΔT is constant.
37. What are the options in MS Excel for optimization ?
38. What are the features of MATLAB?
39. What are advantages of MATLAB over C ?
40. What is the difference between M-file and script?
41. Define equivalent diameter.
42. How do you select fluids in pipe side and annulus side in DPHE?