B.M.S. COLLEGE OF ENGINEERING, BENGALURU
Autonomous Institute, Affiliated to VTU
Department of Chemical Engineering

Lab Manual

Elements of chemical engineering (25CH2PSCHE)

Prepared by

Dr. Hari Prasad Uppara and Dr. Shiva Kumar R

Table of contents:

Practical#1

Fit the following experimental data for Arrhenius equation and obtain the activation energy and frequency factor.

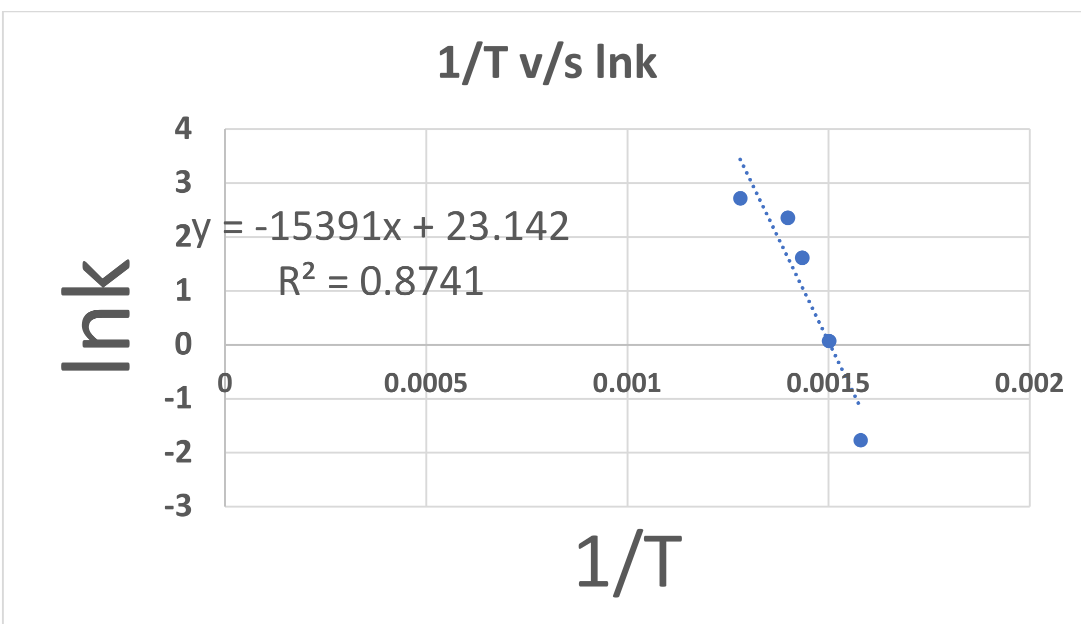
S. No.	Temperature, T (K)	rate constant, k (L/mol.s)
1	633	0.17
2	666	1.07
3	697	5.01
4	715	10.5
5	781	15.1

Arrhenius equation:

$$k = k_o e^{\frac{-E}{RT}}$$

$$lnk = lnk_o - \frac{E}{RT}$$

Plot lnk v/s 1/T, and obtain the values for k_o and E.



Practical#2

Spreadsheet for unit conversion and dimensional analysis

Activity#1:---

Syntax for conversion factor:

=convert(number, “from_unit”, “to_unit”)

Example:

Convert 1 inch to meter

=convert(1, “inch”, “m”)

Answer:

0.0254

Activity#2---

Determine the conversion factors for the following items and fill it in the Table.

S.No.	Quantity	from	to	Final value
1	1	kg	g	
2	1	lbf	N	
3	1	kgf	N	
4	1	cm	m	
5	1	ft	m	
6	1	min	sec	
7	1	min	hr	
8	1	hr	min	
9	1	cal	J	
10	1	atm	bar	
11	1	atm	torr	
12	1	atm	mm Hg	
13	1	deg C	deg K	
14	1	deg C	deg F	
15	1	BTU	J	
16	1	W	J/s	
17	1	W	hp	

Activity#3---

Convert 100 kg/m^3 to g/cm^3

Convert

Convert 10 kgf/cm^2 to N/m^2

Convert 0.25 lbf/cm^2 to N/m^2

Convert 15 W/m.K to kCal/hr. m. °C

Convert 14.25 Btu/hr. ft. F to W/m.K

Activity#4---

Convert the following

- 700 kg/m^3 to lb/ft^3
- 7 BTU to J
- 70 °F to K
- 7000 cP to Pa s
- 0.7 kg/L to ppm

Activity#5---

In the SI system, thermal conductivity has the unit $\frac{\boldsymbol{W}}{\boldsymbol{m.\ K}}$. The thermal conductivity of a solid material can be calculated as $\boldsymbol{k} = \frac{\boldsymbol{XQ}}{\boldsymbol{A\Delta T}}$, where Q is the rate of heat, 'X' is the thickness of solid, 'A' is the area of heat transfer and ΔT is the temperature difference across the solid. The following values were obtained experimentally: Q = 10 000 kJ/h, A = 1 m^2, x = 100 mm and ΔT = 800 K.

a) Calculate the thermal conductivity of the solid in $\frac{\boldsymbol{W}}{\boldsymbol{m.\ K}}$.

b) Express the thermal conductivity in $\frac{kCal}{hr.\ m.\ °C}$

Activity#6---

The equation for the heat transfer to or form a stream of gas flowing in turbulent motion is as follows.

$$h = \frac{\alpha C_p G^{0.8}}{D^{0.2}} = 16.6 \frac{C_p G^{0.8}}{D^{0.2}}$$

Where,

C_p *is heat capacity,* $\frac{Btu}{lb\ °F}$

D *is the diameter of the pipe, inches*

G *is mass velocity of the pipe,* $\frac{lb}{s\ ft^2}$

h *is heat transfer coeffeient,* $\frac{Btu}{hft^{2}°F}$

Convert the equation to

$$h' = \frac{\alpha' C_p' (G')^{0.8}}{(D')^{0.2}}$$

Where,

C'_p *is heat capacity,* $\frac{kcal}{kg\ °C}$

D' *is the diameter of the pipe, cm*

G' *is mass velocity of the pipe,* $\frac{kg}{s\ m^2}$

h' *is heat transfer coeffeient,* $\frac{kcal}{s\ m^{2}°C}$

Conversion Factor Table

Table 2.7 Conversion factors

To convert from	*To*	*Multiply by*
atmosphere, standard (atm)	pascal (Pa)	$1.013\ 25 \times 10^5$
bar (bar)	pascal (Pa)	1.0×10^5
British thermal unit (Btu)	joule (J)	$1.055\ 056 \times 10^3$
British thermal unit per pound degree Fahrenheit (Btu lb^{-1} $°F^{-1}$)	joule per kilogram kelvin (J kg^{-1} K^{-1})	4.1868×10^3
calorie (cal)	joule (J)	4.1868
centimetre (cm)	metre (m)	1.0×10^{-2}
centipoises (cP)	pascal second (Pa s)	1.0×10^{-3}
centistokes (cSt)	metre square per second (m^2 s^{-1})	1.0×10^{-6}
cubic foot (ft^3)	cubic metre (m^3)	$2.831\ 685 \times 10^{-2}$
cubic foot per minute (ft^3 min^{-1})	cubic metre per second (m^3 s^{-1})	$4.719\ 474 \times 10^{-4}$
dyne (dyn)	newton (N)	1.0×10^{-5}
erg (erg)	joule (J)	1.0×10^{-7}
foot (ft)	metre (m)	3.048×10^{-1}
foot per hour (ft h^{-1})	metre per second (m s^{-1})	$8.466\ 667 \times 10^{-5}$
gram per cubic centimetre (g cm^{-3})	kilogram per cubic metre (kg m^{-3})	1.0×10^3
horsepower (550 ft lbf s^{-1}), (hp)	watt (W)	$7.456\ 999 \times 10^2$
horsepower (metric)	watt (W)	$7.354\ 988 \times 10^2$
inch (in)	metre (m)	2.54×10^{-2}
kilocalorie (kcal)	joule (J)	4.1868×10^3
kilogram-force (kgf)	newton (N)	9.806 65
kilogram-force per square centimetre (kgf cm^{-2})	pascal (Pa)	$9.806\ 65 \times 10^4$
kilowatt hour (kW h)	joule (J)	3.6×10^6
mile (based on U.S. survey foot), (mi)	metre (m)	$1.609\ 347 \times 10^3$
mile per hour (mi h^{-1})	metre per second (m s^{-1})	4.4704×10^{-1}
poise (P)	pascal second (Pa s)	1.0×10^{-1}
pound (avoirdupois) (lb)	kilogram (kg)	$4.535\ 924 \times 10^{-1}$
poundal	newton (N)	$1.382\ 550 \times 10^{-1}$
pound-force (lbf)	newton (N)	4.448 222
pound-force per square inch, psi, (lbf in^{-2})	pascal (Pa)	$6.894\ 757 \times 10^3$
pound per cubic foot (lb ft^{-3})	kilogram per cubic metre (kg m^{-3})	$1.601\ 846 \times 10^1$
square foot (ft^2)	square metre (m^2)	$9.290\ 304 \times 10^{-2}$
stokes (St)	metre square per second (m^2 s^{-1})	1.0×10^{-4}
ton, metric (t)	kilogram (kg)	1.0×10^3
torr (Torr)	pascal (Pa)	$1.333\ 224 \times 10^2$
watt second (W s)	joule (J)	1.0
degree Celsius (temperature) (°C)	kelvin (K)	$T/K = t/°C + 273.15$
degree Fahrenheit (temperature) (°F)	degree Celsius (°C)	$t/°C = (t/°F - 32)/1.8$
degree Fahrenheit (temperature) (°F)	kelvin (K)	$T/K = (t/°F + 459.67)/1.8$
degree Rankine (°R)	kelvin (K)	$T/K = (T/°R)/1.8$

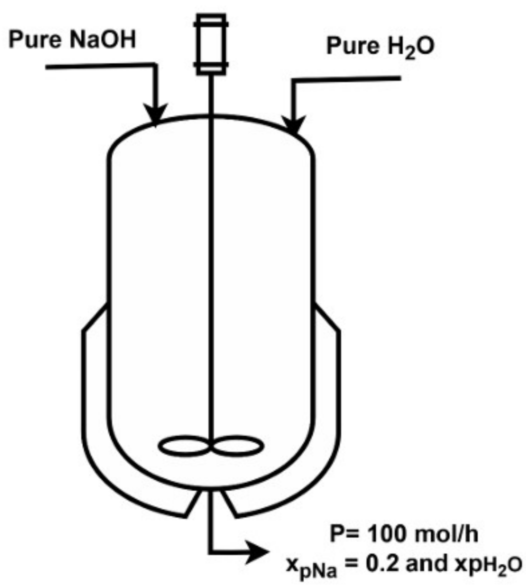
Practical#3

Material Balance for a Mixing Process

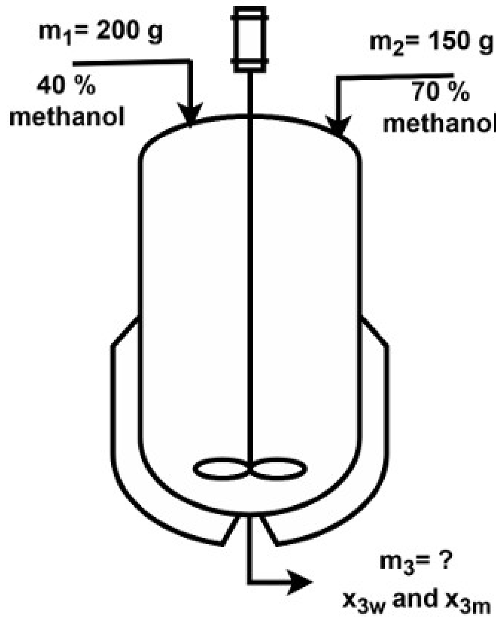
Key Concepts

- System Boundary: Draw a box around the mixer; inputs cross the boundary inward, outputs cross outward.
- Total Balance: Sum of input mass flow rates equals total output mass flow rate.
- Component Balance: Balance for specific components (e.g., salt, acid) equals mass in of component to mass out of component.
- Steady State: Assumes no accumulation inside the mixer, so input equals output.

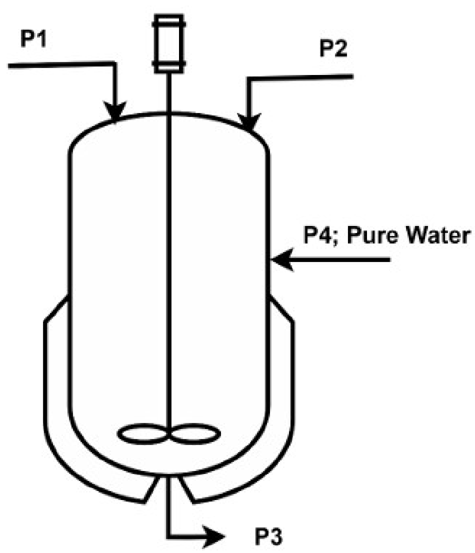
Activity#1: An Aqueous solution at 20 mol% NaOH is prepared by mixing pure NaOH an Pure water. Find the amount of NaOH and Water added.



Activity#2: Two Methanol-water mixtures are sent into the mixer unit. The 200 g first Mixture contains 40.0 Wt % methanol, and the 150 g of second Mixture contains 70.0 Wt % methanol. Find the mass and composition of the product.



Activity#3: 45 kg of beet molasses containing 50 % sucrose, 1 % invert sugars, 18 % water, and the remainder made up solids is mixed with corn-steep liquor in a mixing tank. The 125 kg of corn-steep liquor contains 2.5 % invert sugars, 50 % water, and rest solids. Find the concentration of water, solids, and sucrose.

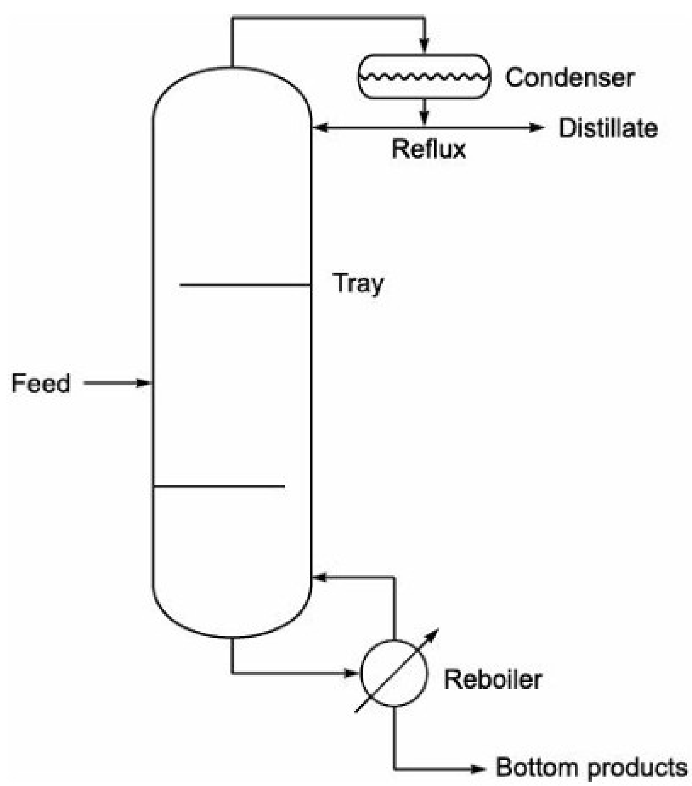


Practical#4

Material balances for a separation process

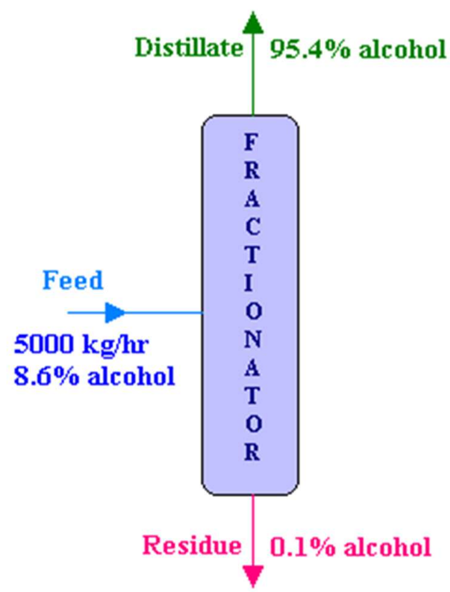
Key Concepts

- Material balance for a distillation column ensures that input mass equals output mass, expressed as (Feed = Distillate + Bottoms) for continuous, steady-state operation.
- Component balances is used to determine component distribution, crucial for designing or analyzing separation efficiency and calculating flow rates of top/bottom streams.



Activity#1: Calculate the mass flowrate of bottom stream and mass fraction of methanol in bottoms stream.

Activity#2: A solution of ethyl alcohol containing 8.6% alcohol by weight is fed at the rate of 5000 kg/hr to a continuous fractionating column operating at atmospheric pressure. The distillate which is the desired product contains 95.4% alcohol by weight and the residue from the bottom of the column contains 0.1% alcohol by weight. Calculate the mass flow rates of the distillate and residue in kg/hr.



Activity#3: A mixture containing 45% benzene (B) and 55% toluene (T) by mass is fed to a distillation column at a rate of 2,000 kg/h. An overhead stream of 95 wt% B is produced. If 8% of the benzene fed to the column leaves in the bottom stream, determine the overhead flow.

Activity#4: A feed stream containing 1,000 kg/hour of a mixture with 50% by mass methanol and 50% water is distilled. The overhead stream (distillate) contains 96% by weight methanol. The bottom stream has a flow rate of 673 kg/hour. Calculate the distillate flow rate and the concentration of methanol in the bottom stream.

Activity#5: 10,000 kg/h of solution containing 20% methanol is continuously fed to a distillation column. Distillate is found to contain 98% methanol and waste solution from the column carries 1% methanol. All percentage are by weight. Calculate: (i) The mass flow rates of distillate and bottom product (ii) The percentage loss of methyl alcohol.

Activity#6: A distillation column separates 20% C_6H_6, 50 % Toluene, 30 % Xylene into 95 % C_6H_6, 4% Toluene and 1% Xylene and waste product containing 2 % C_6H_6. Calculate the quantities of distillate and residue if 1000 kgmol/h of feed is fed.

Practical#5

MATLAB Introduction part -1

SCALAR OPERATORS {+, -, *, /, ^, log (base e), exp, sqrt, power(2,3), sin, cos, tan, asin, acos, atan}

```
A=2; B=3;
sum=A+B %{2+3=5}
product=A*B %{2*3=6}
quotient=A/B %{2 /3=0.66}
A^B %{2^3=8}
log(2.7183) % Returns the natural logarithm (i.e logarithm to base
 e)of 2.718
log10(10)   % Returns the common logarithm (i.e logarithm to base 10)
 of 10}
exp(2)      % Returns the exponential e^2 =7.3891
sqrt(16)    % Returns the square root of 16 =4
power(2,3)  % Returns the 2^3=8
sin(pi/2)   % sin(x) returns the sine of angle x. x should be in
 RADIANS.
asin(1)     % asin(x) is Inverse sin (x). It returns the angle in
 RADIANS.
```

Creating VECTORS

Generating a ROW VECTOR. Elements separated by spaces or commas (,).

```
A=[1,1,5]
B=[1 1 5]
% Generating a COLUMN VECTOR. Elements separated by semicolons (;)
C=[1;1;5]
```

MATRIX OPERATIONS -1

Creating a 3x3 matrix P.

```
P=[1,2,3;4,5,6;7,8,9]
% VERTICAL CONCATENATION
Q=[10,11,12]
R=[P;Q]
% HORIZONTAL CONCATENATION
S=[13;14;15;16]
T=[R,S]
% ADDITION of two matrices (The matirces should be of the same
 dimensions)
```

```
U=P+P
% MULTIPLICATION of two matrices (The numbers of columns of the 1st
 matrix
% should be the same as the number of rows of the 2nd matrix){1*1+2*3,
 1*2+2*4; 3*1+4*3, 3*2+4*4}
V=[1,2;3,4]
W=V*V
```

MATRIX OPERATIONS -2

To get the SIZE of the matrix P. {It is a 3 x 3 matrix}

```
size(P)
% To get the SUM each column of the matrix P.
sum(P)
% To get the DIAGONAL ELEMENTS {1,5,9} of the matrix P.
diag(P)
% To get the TRANSPOSE of the matrix P.
P'
% To get the MAXIMUM values for each coloumn of the matrix P.
max(P)
% To get the MINIMUM values for each column of the matrix P.
min(P)
% To produce the AVERAGE of each column of the matrix P.
mean(P)
% To produce the AVERAGE  of all the numbers in the matrix P.
mean2(P)
% To get one long column, which is a concatenation of all the
 individual columns of the matrix P.
P(:)
% To produce the SUM of all the numbers in the matrix P.
sum(P(:))
% To produce the AVERAGE of each ROW of the matrix P.
mean(P,2) % The first input is the data, the 2nd inout is the
 dimension to operate on. Therefore mean(P,2)
% is the mean over the 2nd dimension of matrix P, which returns the
 means of the rows.
(mean(P'))' % P' converts all the rows to columns. mean(P') gets the
 mean of the columns of P' in the form
% of a row array. Converting row array to column array is done by
 transpose of mean(P').
% To raise matrix P to the POWER B. P^3 = P * P * P
P^B
% To raise each element of P to the POWER B.
P.^B % power(A,B) is an alternate way to execute A.^B
```

EXTRACTING ELEMENTS of a matrix

To Extract 2nd row's 2nd element {i.e 5} of matrix P

```
P(2,2)
% To Extract the ROW {4,5,6} of the matrix P
P(2,1:3)
```

```
% Extracting the COLUMN {1,4,7} of the matrix P
P(1:3,1)
% % Extracting SUB-MATRIX {2,3; 5,6}
P(1:2,2:3)
```

INBUILT MATRIX FUNCTIONS

eye(n,m) returns an n-by-m IDENTITY matrix with ones on the main diagonal and zeros elsewhere.

```
eye(3,4)
% eye(n) returns an n-by-n IDENTITY matrix with ones on the main
 diagonal and zeros elsewhere.
eye(4)
% ones(n,m) returns an n-by-m matrix with ones as its elements.
ones(3,4)
% zeros(n,m) returns an n-by-m matrix with zero's as its elements.
zeros(2,3)
% rand(n,m) returns an n-by-m matrix of uniformly distributed random
 numbers between 0 and 1 as its elements.
rand(2,3)
% randn(n,m) returns an n-by-m matrix of normally distributed random
 numbers with mean 0 and standard deviation 1
randn(2,3)
```

SCRIPTS and FUNCTIONS

Scripts are m-files containing MATLAB statements. MATLAB `functions' are another type of m-file. The biggest difference between scripts and functions is that functions have input and output parameters. Script files can only operate on the variables that are hard-coded into their m-file. Example : Create a SCRIPT file named triarea.m that computes the area of a triangle:

```
base = 5;
height = 3;
area = 0.5*(base*height)
% Call the script 'triarea' at the command line OR run the script.
area
% MATLAB returns area = 7.5000 .
% Each time you run it, the script stores the result in a variable
 named 'area' that is in the base workspace.
% To calculate the area of another triangle using the same script, you
 could update the values of 'base' and 'height' in the
% script and rerun it.
% However, instead of manually updating the script each time, you can
 make your program more flexible by
% converting it to a FUNCTION by name 'triarea' as shown below:
function area = triarea(base,height)
area = 0.5*(base*height);
end
% We can call this function with different 'base' and 'height' values
 from the command line without modifying
% the script as shown below:
area1 = triarea(1,5)
area2 = triarea(2,10)
```

Practical#6

MATLAB Introduction part -2

Calculate density

The density of a fluid is:

[
\rho=\frac{m}{V}
]

Suppose the mass is 50 kg and volume is 0.04 m³.

```
clc;

clear;

mass = 50;      % kg

volume = 0.04;   % m^3

density = mass/volume;

fprintf('Density = %.2f kg/m^3\n',density);
```

Output

Density = 1250.00 kg/m^3

MATLAB can be used to visualize mathematical functions.

Consider:

[
y=x^2
]

```
clc;

clear;

close all;

x = -5:0.1:5;

y = x.^2;


plot(x,y,'LineWidth',2);
```

```
xlabel('x');
ylabel('y');
title('Plot of y = x^2');
grid on;
```

The resulting graph is a parabola.

Consider:

[
y_1=x
]

and

[
y_2=x^2
]

```
clc;
clear;
close all;
x = 0:0.1:5;
y1 = x;
y2 = x.^2;
plot(x,y1,'LineWidth',2);
hold on;
plot(x,y2,'LineWidth',2);
hold off;
xlabel('x');
ylabel('y');
title('Comparison of y = x and y = x^2');
legend('y = x','y = x^2');
grid on;
```

Practical#7

MATLAB code for determining the regression coefficient

Activity#1

Consider the following steps to write and execute the MATLAB code. You are supposed to used MATLAB help command to write this code. Don't use internet.

1. The data of two variables are considered as follows:

X1	1	2	3	4	5	6
Y1	1	20	50	65	100	115

2. Plot the data for x1 and y1, and try to do changes in the plot with symbols, lines, color, markersize – USE help command
3. Use the syntax **polyfit** to fit the data [% consider first order and second order polynomial equation]

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

4. Use the syntax **polyval** to determine the predicted y1 values
5. Plot x1 v/s y1 predicted
6. Plot the both graphs in one figure [x1 v/s y1 and x1 v/s y1_pred]
7. Label the graph x-axis, y-axis, title, legend
8. Find error = y1_pred – y1
9. Find root mean square error (RMS)
10. Find R^2

Matlab code:

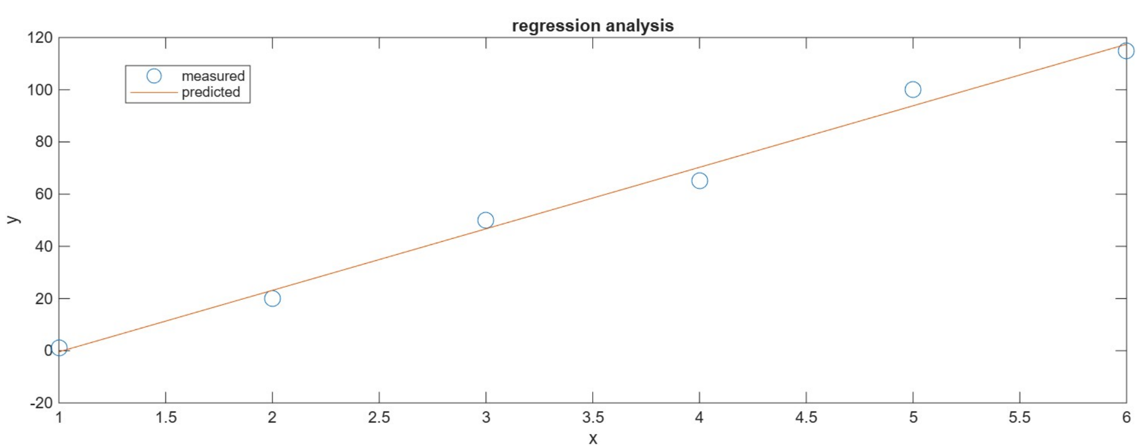
```
clc; clear all; close all;

x = 1:6;

y = [1 20 50 65 100 115];


plot(x,y,'o','MarkerSize',10)
```

```
a = polyfit(x,y,1)
%the polynomial equation becomes y = 22.342x-21.2
y_hat = polyval(a,x)
hold on
plot(x,y_hat)
xlabel('x')
ylabel('y')
title('regression analysis')
legend('measured','predicted')

%error
error = y_hat-y;
RMS = sqrt(mean(error.^2))
```

Output:

a= 23.5714 -24.0000

RMS= 3.9641



Activity#2

Follow the above steps to fit the following experimental data.

Fit the following experimental data for Arrhenius equation and obtain the activation energy and frequency factor.

S. No.	Temperature, T (K)	rate constant, k (L/mol.s)
1	633	0.17
2	666	1.07
3	697	5.01
4	715	10.5
5	781	15.1

- Arrhenius equation:

$$k = k_o e^{\frac{-E}{RT}}$$

$$lnk = lnk_o - \frac{E}{RT}$$

- Plot lnk v/s 1/T, and obtain the values for k_o and E.
- Estimate error, RMS, R^2 for the above experimental data.

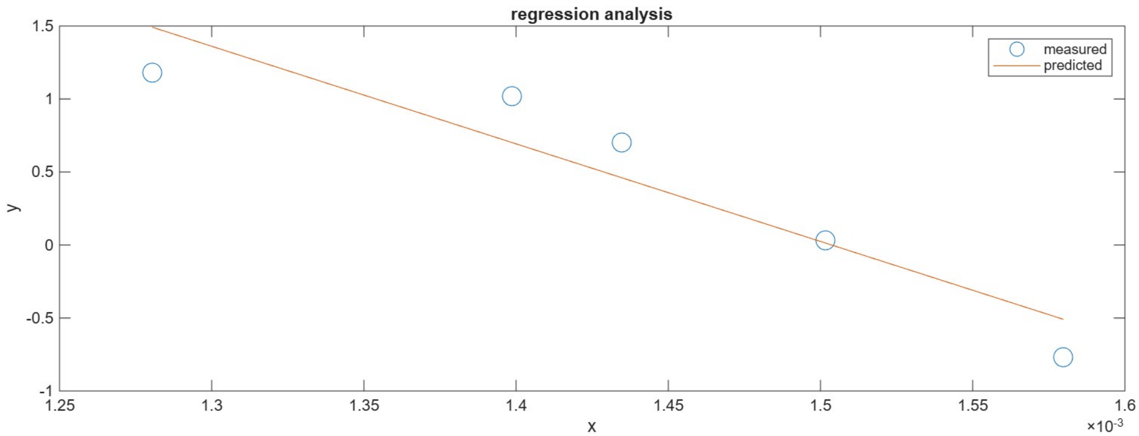
Matlab code:

```
clc; clear all; close all;


x1 = [633 666 697 715 781];

x = 1./x1;

y1 = [0.17 1.07 5.01 10.5 15.1];

y = log10(y1);

plot(x,y,'o','MarkerSize',12)

a = polyfit(x,y,1)

%the polynomial equation becomes y = 22.342x-21.2
```

```
y_hat = polyval(a,x)
hold on
plot(x,y_hat)
xlabel('x')
ylabel('y')
title('regression analysis')
legend('measured','predicted')
%error
error = y_hat-y;
RMS = sqrt(mean(error.^2));
SSresid = sum(error.^2);
SStotal = (length(y)-1*var(y));
Rsq = 1 - SSresid/SStotal
% Display the results
fprintf('R-squared: %.4f\n', Rsq);
fprintf('Root Mean Square Error: %.4f\n', RMS);
```

Output:

```
R-squared:0.9253
Root Mean Square Error: 0.2550
```


regression analysis
measured
predicted
1.5
1
0.5
0
-0.5
-1
y
1.25
1.3
1.35
1.4
1.45
1.5
1.55
1.6
x
×10-3

Practical#8 and 9

MATLAB code for solving ordinary differential equations – application in chemical engineering

Theory:

ode45 is MATLAB's medium-order explicit Runge–Kutta solver (Dormand–Prince) for nonstiff initial-value problems. Use it when your ODE system is reasonably smooth and nonstiff; rewrite higher-order equations as a first-order system and supply a function f(t,y) that returns dydt.

Key usage pattern:

- Basic call: Provide time span tspan = [t0 tf] and initial condition y0; call
- Passing extra parameters: Wrap with an anonymous function, for example
- [t,y] = ode45(@(t,y) odefcn(t,y,A,B),tspan,y0);

Difference between ode45 and ode23:

Both ode23 and ode45 solve nonstiff initial-value problems y' = f(t,y), but they differ in order, accuracy, and typical use:

- Method and order: ode45 uses the Dormand–Prince pair (explicit Runge–Kutta of order 4(5)) for medium-order accuracy. ode23 uses an explicit Runge–Kutta pair of order 2(3), a lower-order method.
- Accuracy and tolerances: ode45 provides higher accuracy per step and is generally the recommended first choice. ode23 provides lower accuracy and can be preferable when crude tolerances suffice.
- Efficiency tradeoffs: For smooth problems with moderate accuracy needs ode45 is usually efficient. For problems with loose error

tolerances or some moderate stiffness, ode23 can be cheaper because its steps are less costly even if more numerous.

- When to choose which: Start with ode45; use ode23 if you intentionally request coarse tolerances or observe ode45 being inefficient. For stiff problems, use stiff solvers like ode15s.

Activity#1

Batch Reactor Concentration Decay

A first-order irreversible reaction is carried out in a batch reactor:

$$A \rightarrow Products$$

Rate equation:

$$\frac{dC_A}{dt} = -kC_A$$

Given:

- $k = 0.25\ \mathrm{min}^{-1}$
- Initial concentration $C_{A0} = 2$ mol/L

Plot the concentration variation for 20 minutes. Use ODE45 syntax to solve this problem.

Activity#2

Solve

$$\frac{dy}{dt} = -2y + 3e^{-t}$$

with initial condition

$$y(0) = 1$$

Activity#3

Cooling of a Hot Fluid Tank

According to Newton's law of cooling:

$$\frac{dT}{dt} = -k(T - T_s)$$

Given:

- Surrounding temperature $T_s = 30°C$
- Initial temperature $T_0 = 120°C$
- Cooling constant $k = 0.12\ \text{min}^{-1}$

Find temperature variation and plot the graph.

Activity#4

Non-Isothermal Chemical Reactor

Temperature rise in an exothermic reactor:

$$\frac{dT}{dt} = Q - k(T - T_s)$$

Given:

- Heat generation $Q = 8$
- Cooling coefficient $k = 0.3$
- Surrounding temperature $T_s = 25°C$
- Initial temperature $T_0 = 40°C$

Practical#10

MATLAB Program for variation of height with respect to time in a tank (Solution of a differential equation).

Problem Statement

A cylindrical tank has a constant cross-sectional area of **2 m²**. Water enters the tank at a constant rate of **0.20 m³/min** and leaves the tank at a constant rate of **0.10 m³/min**.

The initial water level in the tank is **1.0 m**.

Determine the variation of water height with time for **0–20 min** using MATLAB and plot the height-versus-time curve.

MATLAB Program — Using ode45

```
clc;

clear;

close all;


% Parameters

A = 2;          % Tank cross-sectional area, m^2

Qin = 0.20;     % Inlet flow rate, m^3/min

Qout = 0.10;    % Outlet flow rate, m^3/min

h0 = 1.0;       % Initial liquid height, m


% Time interval

tspan = [0 20];


% Differential equation

dhdt = @(t,h) (Qin - Qout)/A;
```

```
% Solve the ODE using ode45
[t,h] = ode45(dhdt, tspan, h0);

% Display final height
fprintf('Final liquid height = %.2f m\n', h(end));

% Plot height versus time
figure;
plot(t,h,'LineWidth',2);
xlabel('Time (min)');
ylabel('Liquid Height (m)');
title('Variation of Liquid Height with Time');
grid on;
```

Sample MATLAB Output

The Command Window displays approximately:

Final liquid height = 2.00 m

The expected plot is a straight line because the difference between inlet and outlet flow rates is constant.

Expected graphical output

```
Height (m)
 2.0 |                         *
     |                      *
 1.8 |                   *
     |                *
 1.6 |             *
     |          *
 1.4 |       *
     |    *
 1.2 | *
     |
 1.0 |*
     +-----------------------------

       0   5   10   15   20
              Time (min)
```

The graph shows that the liquid height increases linearly with time.