AMT TUTORIAL-7

BY: S.VISHAL
CH18B020

① Freundlich isotherm:

$$q = KC^{1/n}$$ ; parameters: q and n.

Model is fit using nlinfit function in MATLAB.

$K = \underline{5.655}$ $n = \underline{2.176}$

$$\Rightarrow q = 5.655\, C^{0.4596}.$$

Mean Squared error = $\underline{30.467}$

Langmuir Isotherm:

$$q = \frac{q_m KC}{1+KC}$$ ; parameters: $q_M$ and K.

Model is fit using nlinfit function in MATLAB.

$\underline{K = 0.0433}$, $\underline{q_M = 52.002}$.

$$\Rightarrow q = \frac{2.252\, C}{1+0.0433C}$$

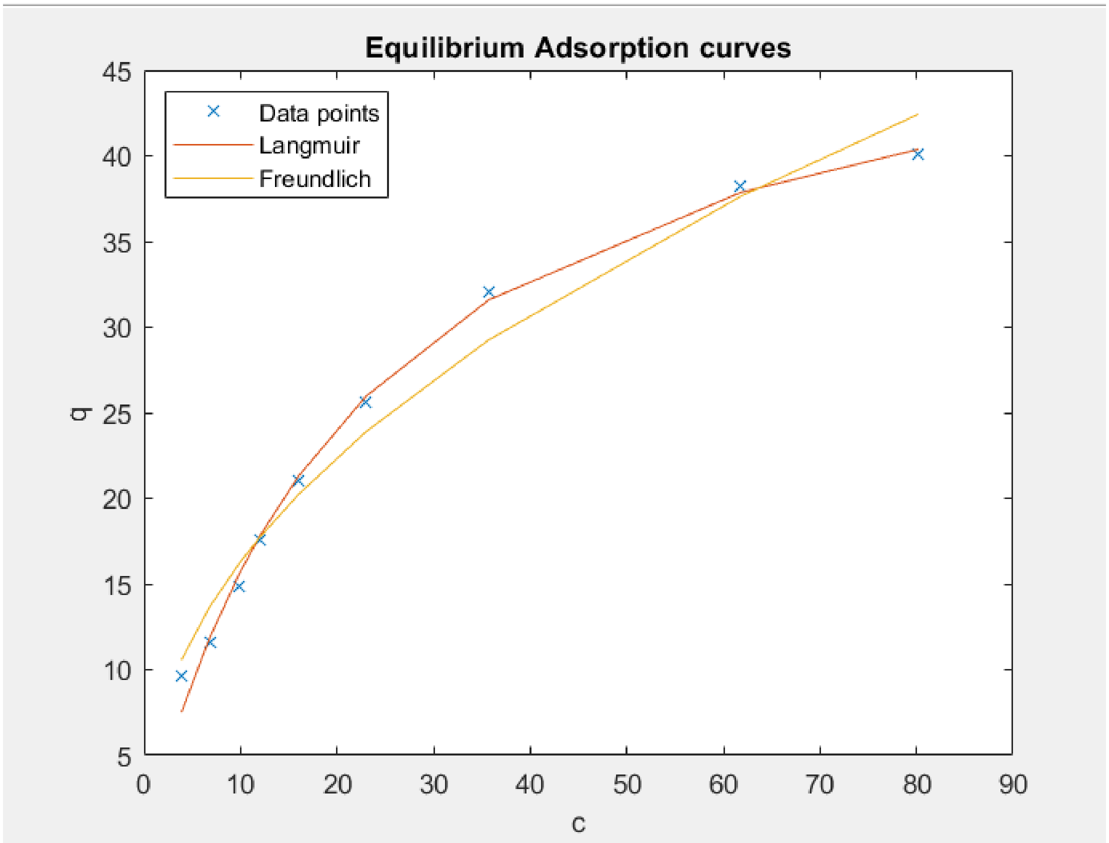
Mean squared error = $\underline{0.801}$.

By comparing the mean squared errors as well as the visual examination of the ~~atta~~ adsorption isotherm plots, we find that Langmuir isotherm best fits the data.

(K = ~~8.655~~ 0.0433, ~~n = 2.176~~) $q_M = 52.002$)

# Question 1 MATLAB CODE AND PLOT

```
clear;close all; clc;
c = [3.9;6.8;9.9;12;16;23;35.7;61.8;80.2];
q= [9.6;11.6;14.9;17.6;21.0;25.6;32.1;38.2;40.1];
options = statset('MaxIter',500);
[beta,R,J,COVB,MSE] = nlinfit(c,q,@Freund,[1,1]);
[beta1,R1,J1,COVB1,MSE1] =
nlinfit(c,q,@Lang,[0.00001,2*10000],options);
qlang = Lang(beta1,c);
qfreund = Freund(beta,c);
plot(c,q,'x',c,qlang,c,qfreund);
title('Equilibrium Adsorption curves');
legend('Data points','Langmuir', 'Freundlich');
xlabel('c');
ylabel('q');
function y = Freund(parms,c)
    k = parms(1);
    n = parms(2);
    y = k.*(c.^(1./n));
end
function y= Lang(parms,c)
    k = parms(1);
    qm = parms(2);
    y = qm.*(k.*c./(1+k.*c));
end
```



(2) a) At minimum Adsorbent condition,

$$\frac{W_{min}}{L} = \frac{Y_0 - Y_1}{X_1 - X_0}$$

$Y_0 = 3.5$ mg/L  $Y_1 = 0.1$ mg/L

$X_0 = 0$ and $X_1$ is in eqbm with $Y_1$

$$\Rightarrow X_1 = 50(0.1)^{0.32}$$

$= 23.932$ mg compound/g charcoal

$L = 10$ Litres

$$\Rightarrow W_{min} = 10 \times \frac{(3.5 - 0.1)}{23.932}$$

$$\Rightarrow \boxed{W_{min} = 1.4207 \text{ g charcoal}}$$

b) $$Sh = \frac{k_L \times (\text{diameter of particle})}{D_{AB}}$$

$D_{AB} = 1.5 \times 10^{-5}$ cm²/s $= 1.5 \times 10^{-9}$ m²/s

$Sh = 30$ ; particle diameter $= 2 \times 10^{-3}$ m.

$$\therefore k_L = \frac{1.5 \times 10^{-9} \times 30}{2 \times 10^{-3}} = 2.25 \times 10^{-5} \text{ m/s.}$$

$W = 2\, W_{min} = 2.841$ g

Adsorbate Balance :

$$-\frac{\partial c}{\partial t} = k_L a (c - c^*)$$

c is the concentration of adsorbate in the solution at any time t.

a is the surface area of charcoal per unit volume of solution.

$\Rightarrow a = 5\ m^2/\text{kg charcoal} \times \dfrac{(2.841 \times 10^{-3}\ \text{kg charcoal})}{10 \times 10^{-3}\ m^3\ \text{solution}}$

$= 1.421\ m^2/m^3$ solution.

$c^*$ is the eqbm concentration of adsorbate in solution

$$\Rightarrow c^* = \left(\frac{(c_i - c)L}{wK}\right)^n \quad mg/L$$

From the given isotherm,

$n = \dfrac{1}{0.32} = 3.125$ , $K = 50$

Substituting all the values in the eqn,

$$-\frac{\partial c}{\partial t} = k_L a\left(c - \left(\frac{(c_i - c)L}{wK}\right)^n\right)$$

$$\Rightarrow \int_0^t dt = \int_{3.5}^{0.1} \frac{-dc}{\left(k_L a\left(c - \left(\frac{(c_i - c)L}{wK}\right)^n\right)\right)}$$

$$= \int_{0.1}^{3.5} \frac{dc}{k_L a\left(c - \left(\frac{(c_i - c)L}{wK}\right)^n\right)}$$

Integrating this expression numerically in MATLAB,

$t = 1.1428 \times 10^5\ s$

$= \boxed{31.743 \text{ hours}}$

## Question 2 MATLAB CODE

```
clear; close all;
kl = 2.25*10^-5;%m/s
L = 10;%Litres
wmin = (3.5-0.1)/(50*0.1^(0.32))*L;%g
w = 2*wmin;%g
ci = 3.5;%mg/L
K = 50;%q should be in mg/g, c in mg/L
a = 5;%m^2/kg
time = integral(@dtbydc,3.5,0.1);
t_inhours = time/3600;
function val = dtbydc(c)
    kl = 2.25*10^-5;%m/s
    L = 10;%Litres
    wmin = (3.5-0.1)/(50*0.1^(0.32))*L;%g
    w = 2*wmin;%g
    ci = 3.5;%mg/L
    K = 50;%q should be in mg/g, c in mg/L
    a = 5;%m^2/kg
    %units of val
    %c is the conc in solution, second term is the ideal
c in soln
    val = -1./(kl*(a*w/L).*(c-((ci-
c)*L/(w*K)).^(1/0.32)));
end
```

③ a) $L_{expt} = 0.2\ m.$

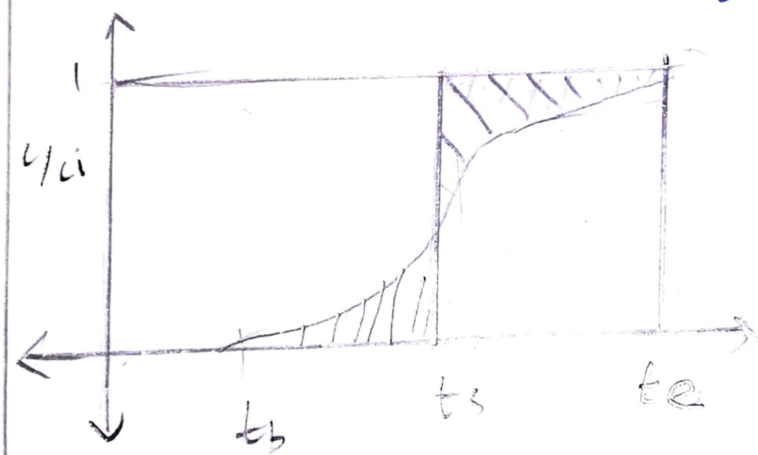
$u_{superficial} = 0.29\ m/s$

$c_i = 0.11\ gmol/m^3.$

Breakthrough time : time at which $\frac{c}{c_i} = 0.03.$

$$\Rightarrow \boxed{t_b = 20.626 \text{ minutes}}$$

(Obtained using Spline interpolation in MATLAB)




At stoichiometric time Area above curve = Area below the curve

$$\Rightarrow \int_{t_b}^{t_s} \frac{c}{c_i}\, dt = 1 \times (t_e - t_s) - \int_{t_s}^{t_e} \frac{c}{c_i}\, dt$$

$$\Rightarrow \int_{t_b}^{t_s} \frac{c}{c_i}\, dt + \int_{t_s}^{t_e} \frac{c}{c_i}\, dt = (t_e - t_s)$$

$$\Rightarrow (t_e - t_s) = \int_{t_b}^{t_e} \frac{c}{c_i}\, dt.$$

$t_e$ is the saturation time - the bed is fully used up. ($c/c_i = 1$)

$\Rightarrow$ from the data $t_e = 108.6$ minute

Performing the integration numerically,

$t_s = 45.18$ minutes

$$u_s = \frac{L}{t_s} = \frac{0.2}{45.18} = 4.427 \times 10^{-3} \text{ m/min}$$

$$\Rightarrow u_s = 0.2656 \text{ m/h.}$$

$$LUB = L \times \left(1 - \frac{t_b}{t_s}\right).$$

$$= 0.2 \times \left(1 - \frac{20.63}{45.18}\right)$$

$$\Rightarrow \boxed{LUB = 0.1087 \text{ m}} = 0.109 \text{ m.}$$

Since the curve is not symmetric ($\frac{c}{c_i}$ vs $t$ curve)

we can't use $MTZ = 2 \times LUB$.

Instead we use the general expression,

MTZ Length of mass transfer zone $= L \times \left(1 - \frac{t_b}{t_e}\right)$

$\Rightarrow$ Length of mass transfer zone $= \boxed{0.162 \text{ m}}$

(Note: If ~~assum~~ the curve was symmetric,

$MTZ = 2 \times LUB$)

b) $\dot{V} = 3000\ m^3/h.$

$= \frac{3000}{3600}\ m^3/s$

$\Rightarrow \dot{V} = 0.833\ m^3/s$ ; $u_{superficial} = 0.29\ m/s$

Area of Bed $= \frac{\dot{V}}{u_{superficial}}.$

$\Rightarrow$ Diameter of bed $= \sqrt{\frac{4\dot{V}}{\pi u_{superficial}}}$

$= 1.913\ m.$

Adsorption cycle, $t = 8$ hours

$\Rightarrow$ length of used bed $= u_s \times t$

$= 2.125\ m$

LUB and length of mass transfer zone will remain the same before and after scale up because the wavefront is the same.

$\Rightarrow$ Length of bed = Length used + Length unused $\underset{(LUB)}{\downarrow}$

$\Rightarrow L = 2.125 + 0.1087$

$\Rightarrow L_{total} = 2.233\ m.$

Weight of adsorbate adsorbed till break through, $w = \dot{V} \times c_i \times t_b$.

$(\because c_{exit} \approx 0)$

$$\Rightarrow W = \underline{2640\ \text{gmol}} = \underline{190.37\ \text{kg}}$$

($MW = 72.11\ \text{g mol}^{-1}$)

$$\text{Average loading} = \frac{\text{Mass of adsorbate}}{\text{Mass of bed}}$$

$$= \frac{\text{Mass of adsorbate}}{\rho_{bed} \frac{\pi D^2}{4} L_{total}}$$

$$= \underline{0.5876\ \text{gmol/kg bed}}$$

$$= \underline{0.0424}\ \text{kg adsorbate / kg bed.}$$

$$\text{Maximum loading} = \frac{\text{Mass of adsorbate}}{\text{mass of } \underline{\text{used part of bed}}}$$

$$= \frac{W}{\rho_{bed} \frac{\pi D^2}{4} L_{used}}$$

$$\therefore \text{Maximum loading} = \underline{0.6177}\ \text{gmol MEK / kg bed}$$

$$= \underline{0.0445}\ \text{kg MEK/kg Bed.}$$

## Question 3 MATLAB CODE AND PLOT

```
clear; close all;
%% Given data
u_superficial = 0.29;
t =
[9.5,19,21,25.7,34.3,39,42,46.7,51.4,56.2,64.7,68.6,72.4,
77.1,84.8,97.1,104.7,108.6];
te = 108.6;
c =
[0,0.018,0.037,0.083,0.287,0.435,0.491,0.62,0.713,0.768,0
.852,0.935,0.952,0.963,0.97,0.987,0.991,1];
L = 0.2;
ci = 0.11;
plot(t,c);
title('c/ci vs t');
ylabel('c/ci');
xlabel('time');

%% Part a
t_break = spline(c,t,0.03);
pp = spline(t,c);
n = 40;
time = linspace(9.5,108.6,n*(10));
ts = te - trapz(time,ppval(pp,time));
% Following code evaluates both the integrals numerically
and minimises it.
% No need to run this part because we have already
simplified it into a simple linear eqn in ts.
% min = trapz(time,ppval(pp,time));
% ts = 230;
% ar_diff = zeros(n*10-1,1);
% l = length(time);
% for i = 2:l-1
%     ar_diff(i) = trapz(time(1:l),ppval(pp,time(1:l)))-
(108.6-time(i));
%     if min > abs(ar_diff(i))
%         min = abs(ar_diff(i));
%         ts = time(i);
%     end
% end
us = L/ts*60; %m/h
LUB = L*(1-t_break/ts);
MTZ = (te-t_break)/te*L;%Since curve is NOT symmetric

%% Part b
V = 3000/60/60;%m^3/s
D = sqrt(4*V/(u_superficial*pi));
```

```
t_cycle = 8;
bed_density = 700;
L_used = 8*us;
L_scaleup = LUB + L_used;
sol_adsorbed = V*ci*t_cycle*3600;
avg_loading =
sol_adsorbed/(L_scaleup*bed_density*pi*D^2/4);
max_loading = sol_adsorbed/(L_used*bed_density*pi*D^2/4);
```