

BE.320 Problem Set #3

1. $\Delta G = -10.49 \frac{\text{kcal}}{\text{mol}}$ @ 25°C

a. $k_{on} = 10^5 \frac{L}{\text{mol}\cdot\text{s}}$

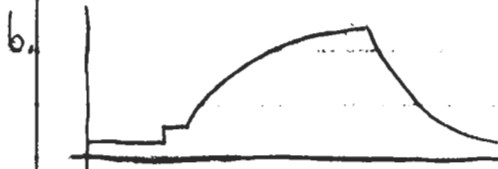
$\Delta G^\circ = RT \ln K_d$

$K_d = \frac{k_{off}}{k_{on}} = e^{\frac{\Delta G^\circ}{RT}}$

$\therefore k_{off} = k_{on} e^{\frac{\Delta G^\circ}{RT}}$

$k_{off} = 10^5 \frac{L}{\text{mol}\cdot\text{s}} e^{-10.490 / (1.987 \cdot 298)}$

$k_{off} = 2.02 \times 10^{-3} \frac{L}{\text{s}}$



Assume $[P]_0 \gg [L]_0$

$[P]_0 = 100 \text{ nM}$

$\tau_{1/2} = \frac{\ln 2}{k_{on}[L]_0 + k_{off}} = 57.67 \text{ seconds}$

$[P]_0 = 10 \text{ nM}$

$\tau_{1/2} = \frac{\ln 2}{k_{on}[L]_0 + k_{off}} = 229.52 \text{ seconds}$

$\tau_{1/2}$ increases by 171.85 seconds

c. $y = \frac{[C]}{[L]_0}$

Assume pseudo first-order reaction:

$\therefore y = \frac{[P]_0}{[P]_0 + K_d}$

$[P]_0 = 100 \text{ nM}$

$y = \frac{100 \times 10^{-9}}{100 \times 10^{-9} + 2.02 \times 10^{-3}} = 83.2\%$

$[P]_0 = 10 \text{ nM}$

$y = \frac{10 \times 10^{-9}}{10 \times 10^{-9} + 2.02 \times 10^{-3}} = 33.1\%$

2. No, you cannot use $\tau_{1/2}$ to estimate peak width in an ITC experiment because the first order approximation does not hold true because $[L]_0 \gg [P]_0$.

3. $k_{obs} = k_{on}[L]_0 + k_{off}$

Let $k_{on} = 10^5 \frac{L}{\text{mol}\cdot\text{s}}$ and $[L]_0 = 1 \text{ nM}$

Let $k_{off} = 10^{-6} \frac{1}{\text{s}}, 10^{-5} \frac{1}{\text{s}}, 10^{-4} \frac{1}{\text{s}}$

1) $k_{off} = 10^{-6} \frac{1}{\text{s}}$

$k_{obs} = 10^5 \cdot 10^{-9} + 10^{-6}$
 $= 1.01 \times 10^{-4} \frac{1}{\text{s}}$

2) $k_{off} = 10^{-5} \frac{1}{\text{s}}$

$k_{obs} = 10^5 \cdot 10^{-9} + 10^{-5}$
 $= 1.1 \times 10^{-4} \frac{1}{\text{s}}$

3) $k_{off} = 10^{-4} \frac{1}{\text{s}}$

$k_{obs} = 10^5 \cdot 10^{-9} + 10^{-4}$
 $= 2 \times 10^{-4} \frac{1}{\text{s}}$

Increasing k_{off} has the effect of a faster approach to equilibrium because the larger the k_{off} , the quicker the complex dissociates and so there will not be as much complex at equilibrium, so equilibrium will be reached quicker.

BE.320 Problem Set #3 (Continued)

4. SPR @ 25°C and 5°C

6 a. Code and Plot below

k decreases 50% when the temperature is decreased from 25°C to 5°C

$$k = Ae^{-E_a/RT}$$

$$\frac{k_1}{k_2} = \frac{Ae^{-E_a/RT_1}}{Ae^{-E_a/RT_2}}$$

$$\frac{k_1}{k_2} = \frac{e^{-E_a/RT_1}}{e^{-E_a/RT_2}}, \text{ assuming } A \text{ constant}$$

$$\ln \frac{k_1}{k_2} = \frac{-E_a}{RT_1} + \frac{E_a}{RT_2}$$

$$\ln \frac{k_1}{k_2} = E_a \left(\frac{1}{RT_2} - \frac{1}{RT_1} \right)$$

$$E_a = \frac{\ln(k_1/k_2)}{(1/RT_2 - 1/RT_1)}$$

$$k_2 = \frac{1}{2}k_1 \text{ where}$$

subscript 1 is 25°C and

subscript 2 is 5°C

$$\therefore E_a = \frac{\ln 2}{(1/(8.31447 \cdot 278)) - 1/(8.31447 \cdot 298)}$$

$$E_a = 23.872 \frac{\text{kJ}}{\text{mol}}$$

$$= 5.705 \frac{\text{kcal}}{\text{mol}}$$

b. No, we cannot come up with a set of k_{2on} and k_{2off} rate constants that causes $[C_2] > [C_1]$ by the end of the simulation. We should modify the amount of the initial concentration of our drug $[L_2]$. The initial concentration of $[L_2]$ must be greater than 2.42M for any chance that $[C_2] > [C_1]$.

5. Code and Plot Below

```

%hw3 nlinfit example
%this code creates "experimental data" with error and then fits a model
%with 2 parameters to this data

global L0; %global parameter, passed into functions with "global L0"

error = 1/100; %amount of experimental noise

L0 = 100.0; %initial concentration of L, in M

t = 0:.01:.11; %time vector

%we will solve for C(t) using the pseudo first-order assumption
%and our initial concentration of receptor (R) is 1M so does not enter the equation
%first, make the true solution (Ctrue) using the true parameters
%then make Cdata, our experimental data (Ctrue+noise)
kon = 1/2; koff = 4; %true parameters
Kd = koff/kon;
Kobs = kon*L0+koff;
for i=1:length(t)
    Ctrue(i) = L0/(L0+Kd).*(1-exp(-Kobs.*t(i))); %R=1
    Cdata(i) = Ctrue(i)+error*(rand-1/2);
end

%then, starting with the known initial condition for L
%and guesses for kon and koff,
%fit the model to the data and get fitted values for kon and koff
k = [0.1, 1.0]; %guesses for kon and koff

%fit 2 parameters: kon and koff
%the missing line should go below this line
fit_k = nlinfit(t,Cdata,@C_function,k);

%and finally, solve for C using the fitted parameters so you can compare
%the fitted model and the data
kon = fit_k(1)
koff = fit_k(2)
Kd_fit = koff/kon;
kobs_fit = kon*L0+koff;
for i=1:length(t)
    Cfitted(i) = L0./(L0+Kd_fit).*(1-exp(-kobs_fit.*t(i))); %R=1
end

figure(1);
plot(t,Ctrue,'b-');
hold on
plot(t,Cdata,'r*');
plot(t,Cfitted,'g-');
legend('true C','C data','fitted solution for C','Location','BEST');
xlabel('t');
ylabel('C');
title('solution, using a double fit for kon and koff');

```

```
%ODE example
clear all;
close all;

%define constant parameters
k1on = 10e-4; %s/M
k1off = 5e-4; %1/s
k2on = 1e-2; %s/M
k2off = 1e-5; %1/M

%define initial conditions
L1 = 10; %initial concentration of L, M
R = 5; %initial concentration of R, M
C1 = 0; %initial concentration of C, M
L2 = 3; %initial concentration of L2, M
C2 = 0; %initial concentration of C2, M

%create initial condition and parameter matrices
Y0 = [L1 R C1 L2 C2];
param = [k1on k1off k2on k2off];

%set simulation parameters
timeRange = [0 1000]; %s
options = [];

%call the numerical ODE solver
[t,Y] = ode45(@ODEexample_equations, timeRange, Y0, options, param);

%plot the results
plot(t,Y);
legend('L1', 'R', 'C1', 'L2', 'C2', 'Location', 'BEST');
xlabel('time');
ylabel('[Concentration]');
title('R + L <--> C');
```

%ODE example equations

function dYdt = ODEexample_equations(t,Y,params)

%this function defines the ODEs describing the bimolecular

%reaction $L + R \leftrightarrow C$ with association and dissociation

%rates kon and koff

%initial conditions

L1 = Y(1);

R = Y(2);

C1 = Y(3);

L2 = Y(4);

C2 = Y(5);

%parameters

k1on = params(1);

k1off = params(2);

k2on = params(3);

k2off = params(4);

dL1dt = -k1on*L1*R + k1off*C1;

dRdt = -k1on*L1*R + k1off*C1 - k2on*L2*R + k2off*C2;

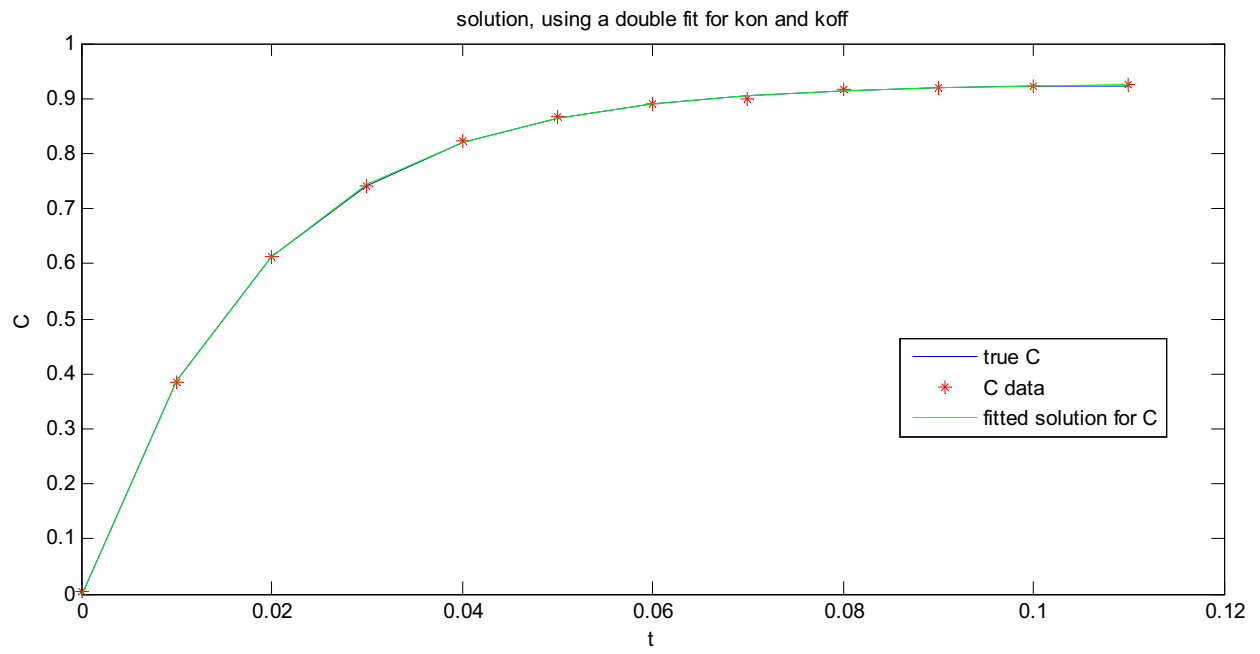
dC1dt = k1on*L1*R - k1off*C1;

dL2dt = -k2on*L2*R + k2off*C2;

dC2dt = k2on*L2*R - k2off*C2;

dYdt = [dL1dt; dRdt; dC1dt; dL2dt; dC2dt];

Problem 5:



Problem 6:

