I. Conformation change kinetics

I.1 This will display <u>first-order</u> kinetics. This is because only one molecule of the protein is involved in each reaction, and the concentration of the closed-form protein will affect how much of it is converting to the open state in a given amount of time.

I.2 $d[closed]/dt = - k_{open} * [closed] + k_{close} * [open]$
$d[open]/dt = - k_{close} * [open] + k_{open} * [closed]$

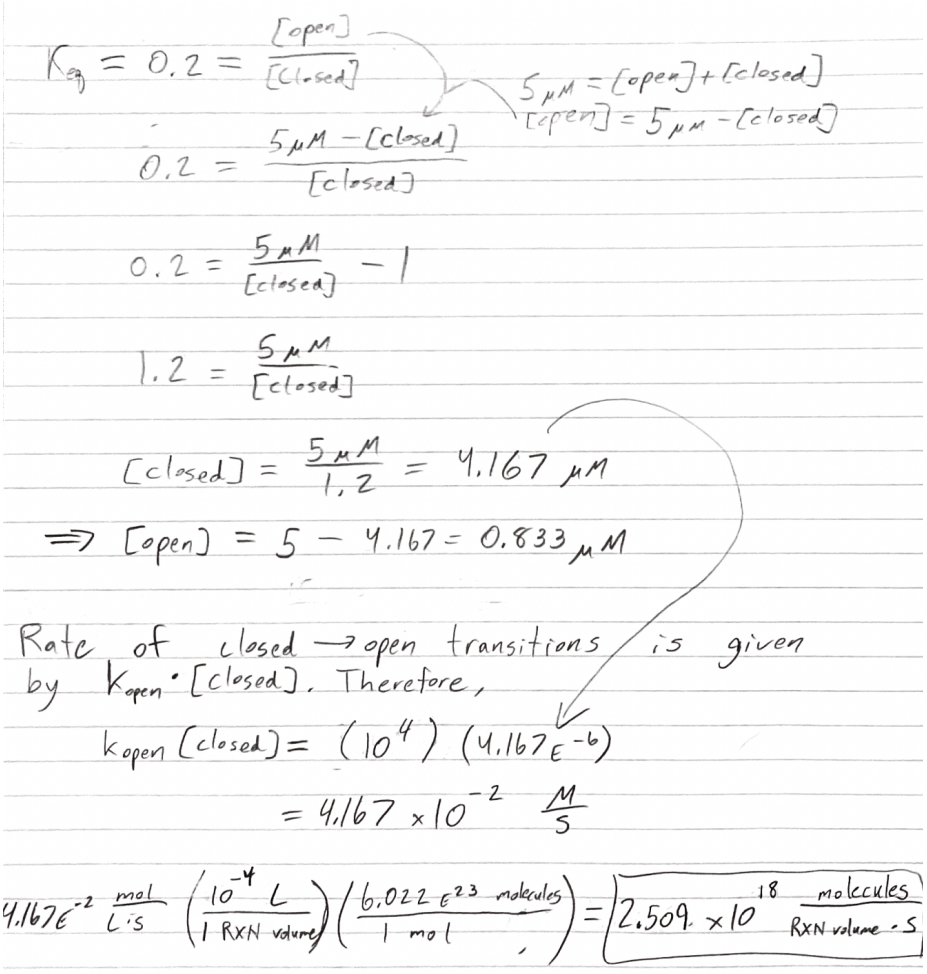
I.3 We know that $\Delta[closed] = \Delta[closed]_0 * \exp(- (k_{open} + k_{close}) * t)$. With a little rearrangement we get:

$$[closed] = ([closed]_o - [closed]_{eq}) * \exp(- (k_{open} + k_{close}) * t) + [closed]_{eq}$$

This means that [closed] is dependent on time in an *exponential* manner, and [closed] will exponentially decay to $[closed]_{eq}$ as time progresses.

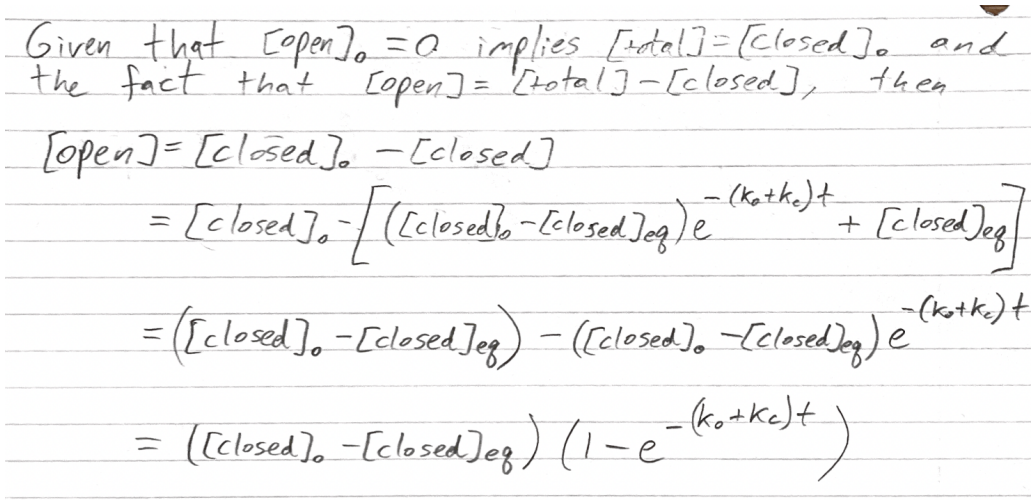
I.4 See image below for work. This is my final equation:

$$[open] = ([closed]_0 - [closed]_{eq}) * (1 - \exp(- (k_{open} + k_{close}) * t))$$



I.5 $K_{eq,22} / K_{eq,25} = (k_{open,22} / k_{close}) / (k_{open,25} / k_{close})$
$K_{eq,22} / K_{eq,25} = k_{open,22} / k_{open,25}$
$K_{eq,22} = K_{eq,25} * (k_{open,22} / k_{open,25}) = 0.2 * (8.4E3 / 10^4) =$ **0.168**

I.6 The system is at equilibrium, so the number of open → closed transitions is the same as the number of closed → open transitions. I calculate the number of closed → open transitions in the image below. My final answer is **2.509x10^{18} transitions/second** in one reaction volume.



I.7 They do not change. The rate constants and equilibrium constant are independent of concentration. Typically, temperature is the only main parameter that, when changed, can result in a change of rate constants.

I.8 At 22°, K_{eq} = 0.168, while at 25°, K_{eq} = 0.2. This means that the product (ie. the open state), becomes more favored when shifting from 22° to 25°. Since the open state has lower fluorescence than the closed state, this shift will result in an overall **decrease** in solution fluorescence.

I.9 I would set up a relaxation experiment in which I use a known concentration of total protein at a given temperature at which I know the K_D, and then instantaneously change to another temperature at which I also know the K_D. Given that we know the fluorescent intensities of both the open and closed state and we know the total protein concentration, we can determine the equilibrium concentrations at both the initial and final temperatures. For example, since [total] = [closed] + [open] at any given point, we can say:

$$Fluor_{eq} = f_{open} * [open]_{eq} + f_{closed} * [closed]_{eq}$$
$$Fluor_{eq} = f_{open} * ([total] - [closed]_{eq}) + f_{closed} * [closed]_{eq}$$

where Fluor is the total fluorescence measured and f_{open}, f_{closed} are the proportionality constants relating concentration of the open or closed state to their fluorescence. Since the above equation has one unknown ($[closed]_{eq}$) it can be solved for $[closed]_{eq}$ and this can be used to also find $[open]_{eq}$. The same strategy can be applied to find $[closed]_0$ and $[open]_0$.

Then, based on the curve of changing fluorescence immediately after the shift, we can fit a curve in the form of the equation in I.3—plugging in our calculated initial and final concentrations—to get a fitted value C of the exponential constant ($k_{open} + k_{close}$).
Since, $k_{close} = k_{open} / K_D$, then we can say:

$$C = k_{open} + k_{close} = k_{open} + (k_{open} / K_D)$$

This equation has only one unknown (k_{open}), so it is solvable, and from this we can calculate k_{close}. Hence, one experiment in which the temperature is shifted and fluorescence is measured is sufficient to determine the rate constants at the final temperature.

II. Enzyme kinetics

II.1 K_m is the substrate concentration (M) at which the initial velocity v_0 of the enzyme will be one-half of the v_{max}. K_m is quantitatively defined as $(k_{-1} + k_2) / k_1$.

II.2 Most likely the problems are that (c) the substrate concentration is too high, and/or (e) the time resolution of the measurement is too low. If $[S]_0 >> K_m$, then the reaction will occur at a velocity near v_{max}. Thus, if all the tested [S] satisfy this criteria, then a similar v_0 will be measured for all conditions, approximately equal to v_{max}. To fix this, I would drastically decrease [S] to see if this gives a smaller v_0 than what has previously been found, and this would tell us if there is a better range of [S] to test.

Alternatively, if the time resolution of the measurement is too low, then most of the substrate may be converted to product before obtaining the data necessary to calculate v_0. In this case, the v_0 for all the conditions will be the same since you see roughly the same amount of substrate converted to product in the too-large measurement window. To fix this, I would decrease the amount of time between initiating the reaction and measuring the substrate disappearance in order to more closely approximate the initial velocity.

II.3 We can obtain v_{max} by approximating the v_0 value that the graph asymptotically approaches as [S] increases. Thus,
$v_{max} \approx$ **1.4×10^{-7} M/s**

K_m is the [S] that results in v_0 that is one-half v_{max}. In this case, this would be [S] when $v_0 \approx 0.5*1.4*10^{-7} = 0.7 \times 10^{-7}$. This occurs when [S] ≈ 0.0015 M. Thus,
$K_m \approx$ **0.0015 M**

Given that $v_{max} = [E]_0 * k_{cat}$, then
$k_{cat} = v_{max} / [E]_0 \approx (1.4 \times 10^{-7}) / (10^{-8}) =$ **$14\ s^{-1}$**

Determine by the definition: catalytic efficiency $= k_{cat} / K_m \approx 14 / 0.0015 =$ **$9.333 \times 10^3\ (Ms)^{-1}$**

II.4 For this question, let L be the analog and P be the enzyme.
a. $k_{obs} = [L]_0 * k_1 + k_{-1}$

b. Given that $[P•L] = ([P]_0 - [P]_{eq}) * (1 - \exp(- k_{obs} * t))$ (similar derivation as in I.4), then if we know α, a constant such that $\alpha * [P•L]$ = fluorescent signal of [P•L], then we can say

$$\text{fluorescent signal} = \alpha * ([P]_0 - [P]_{eq}) * (1 - \exp(- k_{obs} * t))$$

c. We must assume [L] >> [P], and this means that [L] is approximately constant for the entirety of the reaction. This is reasonable in this case because [L] is in the range of mM while the protein concentration is 10 nM, so [L] is 4-6 magnitudes greater than [P].

d. The data given falls exactly on the line $k_{obs} = 10^4 * [analog] + 1$, where [analog] is in units of M. Because we are expecting an equation in the form $k_{obs} = k_1 * [analog] + k_{-1}$, this means: **$k_1 = 10^4\ (Ms)^{-1}$** and **$k_{-1} = 1\ s^{-1}$**

II.5 a. acts as a competitive inhibitor

I expect this to be a competitive inhibitor because the analog binds with the same characteristics as the real substrate, meaning it likely binds at the active site, blocking the substrate from binding. This is characteristic of competitive inhibitors, which compete with the substrate for access to the active site.

II.6 c. increase k_2 by a factor of 2

Increasing k_2 by a factor of 2 has the greatest impact because it increases the velocity of the enzyme the most at the cellular concentrations, as we can see from the calculated v_0 values in the below python script I wrote. The reason this has the most dramatic effect is because it doubles the value of v_{max}, and thus it can nearly double the value of v_0 (this effect is partially offset by the near doubling of K_m, so it does not completely double v_{max}). Meanwhile, the proposed changes to k_1 and k_{-1} only change K_m, and they change K_m by at most a factor of 2. Because [S] is large compared to K_m, this does not appreciably change the value of K_m/[S] and thus does not greatly change $v_0 = v_{max} / (1 + K_m / [S])$.

Hence, increasing k_2 by a factor of 2 results in the largest increase to v_0, and because this will increase the overall rate of substrate-to-product conversion, it will be the most beneficial for cellular fitness.

Question II.6 on Exam 04

```
def multiply(a, b):
    product = []
    for x,y in zip(a,b):
        product.append(x*y)
    return product

E = 10**-8
S = 5 * 10**-3
part = ['original', 'a', 'b', 'c']
k = [3*(10**4), 1.5, 18]
change = [[1, 1, 1],
          [2, 1, 1],
          [1, 0.5, 1],
          [1, 1, 2]]
for i in range(4):
    [k1, kr, k2] = multiply(k, change[i])
    Km = (kr + k2) / k1
    vmax = k2 * E

    v0 = vmax / (1 + Km / S)

    print(part[i]+':', str(v0))
```

```
original: 1.592920353982301e-07
a: 1.6901408450704225e-07
b: 1.6e-07
c: 2.88e-07
```