Question1:

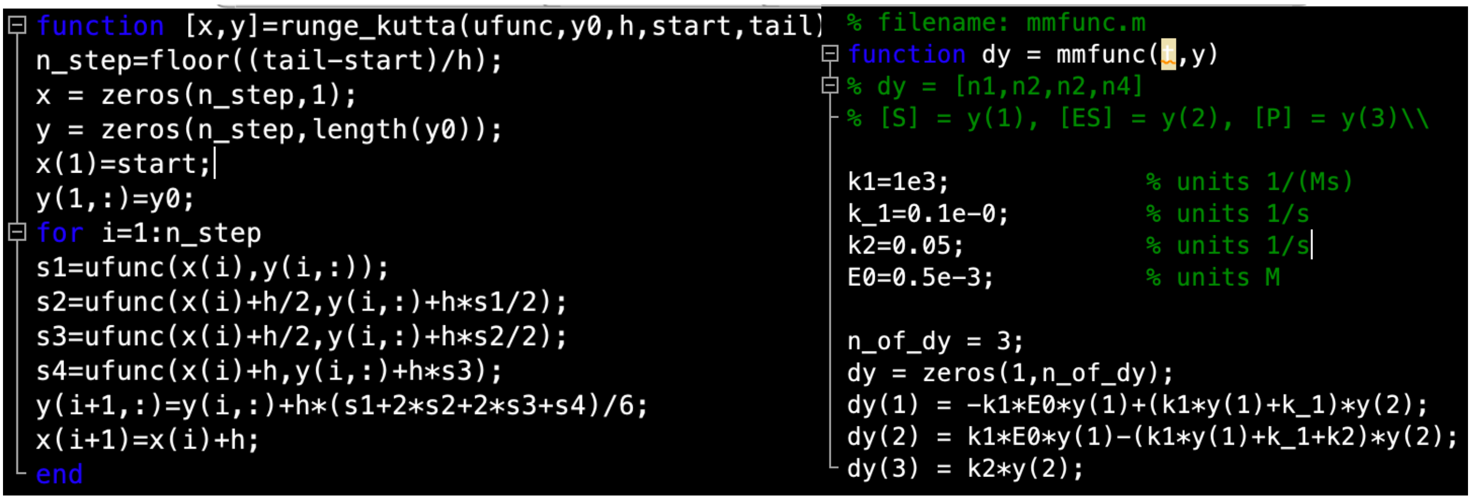
Try to write the matlab code to solve the enzymatic reaction kinetics without ODE solver (ode45, or any other odesolver), but do the integration yourself. You can modify the week2-3 matlab code, mm.m, mmfunc.m. The time step for the difference equation needs to be chosen carefully. You should choose two time step sizes, and compare the results.

Solutions:

The classic fourth-order Runge-Kutta method is used in programing. And the functions should be written in a function like the code below:

The fourth-order Runge-Kutta method:

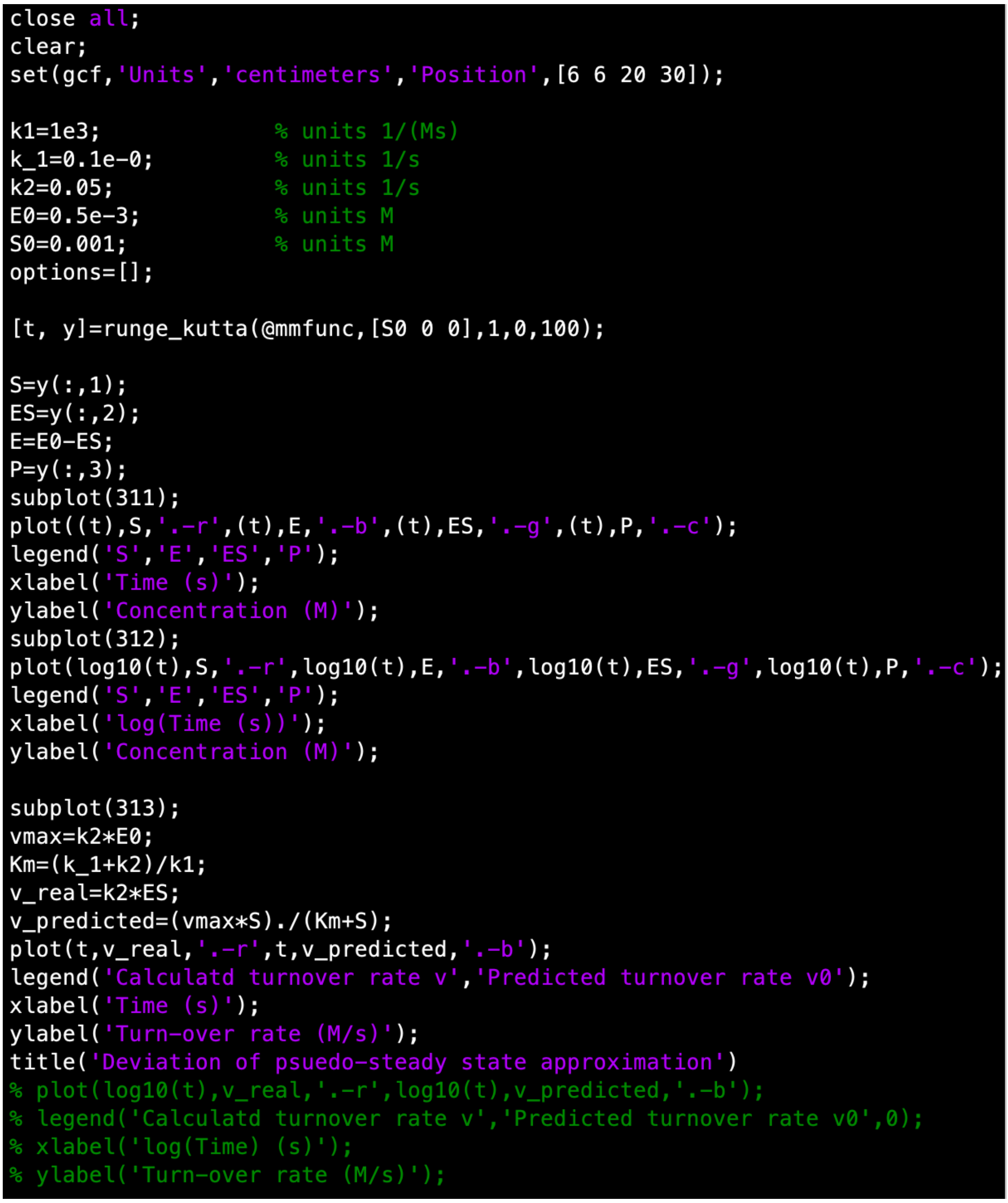
```
input $t_0$ and $y_0$
for k = 1 to N
    $s_1 = f(t_{k-1},\ y_{k-1})$
    $s_2 = f\left(t_{k-1} + \frac{h}{2}, y_{k-1} + \frac{h}{2}s_1\right)$
    $s_3 = f\left(t_{k-1} + \frac{h}{2}, y_{k-1} + \frac{h}{2}s_2\right)$
    $s_4 = f(t_{k-1} + h, y_{k-1} + hs_3)$
    $y_k = y_{k-1} + h\frac{s_1 + 2s_2 + 2s_3 + s_4}{6}$
    $t_k = t_{k-1} + h$
```



```
function [x,y]=runge_kutta(ufunc,y0,h,start,tail)
n_step=floor((tail-start)/h);
x = zeros(n_step,1);
y = zeros(n_step,length(y0));
x(1)=start;
y(1,:)=y0;
for i=1:n_step
s1=ufunc(x(i),y(i,:));
s2=ufunc(x(i)+h/2,y(i,:)+h*s1/2);
s3=ufunc(x(i)+h/2,y(i,:)+h*s2/2);
s4=ufunc(x(i)+h,y(i,:)+h*s3);
y(i+1,:)=y(i,:)+h*(s1+2*s2+2*s3+s4)/6;
x(i+1)=x(i)+h;
end
```

```
% filename: mmfunc.m
function dy = mmfunc(t,y)
% dy = [n1,n2,n2,n4]
% [S] = y(1), [ES] = y(2), [P] = y(3)\\

k1=1e3;             % units 1/(Ms)
k_1=0.1e-0;         % units 1/s
k2=0.05;            % units 1/s
E0=0.5e-3;          % units M

n_of_dy = 3;
dy = zeros(1,n_of_dy);
dy(1) = -k1*E0*y(1)+(k1*y(1)+k_1)*y(2);
dy(2) = k1*E0*y(1)-(k1*y(1)+k_1+k2)*y(2);
dy(3) = k2*y(2);
```

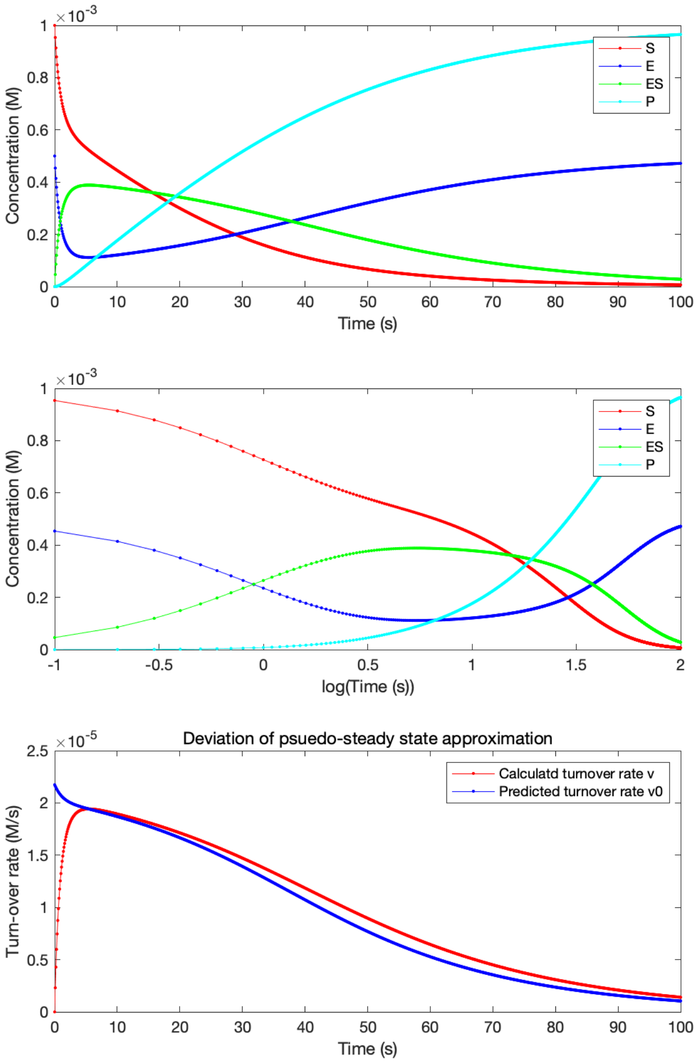
Changed mm.m which named Use_mm_to_test.m is used to run the runge_kutta.m with mmfunc.m.



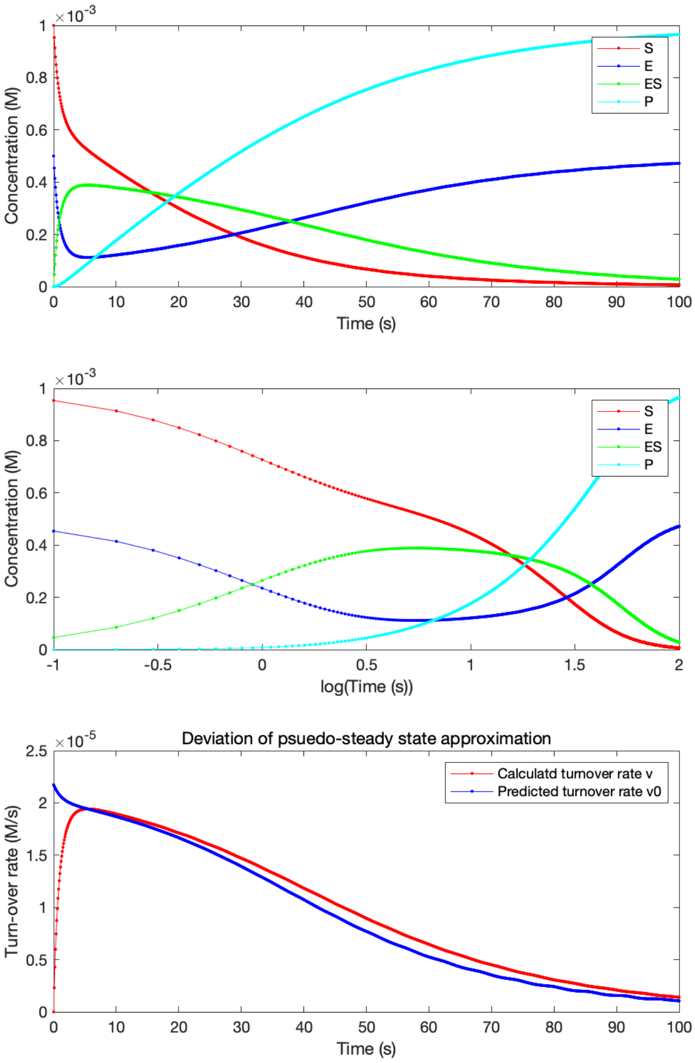
```
close all;
clear;
set(gcf,'Units','centimeters','Position',[6 6 20 30]);

k1=1e3;               % units 1/(Ms)
k_1=0.1e-0;           % units 1/s
k2=0.05;              % units 1/s
E0=0.5e-3;            % units M
S0=0.001;             % units M
options=[];

[t, y]=runge_kutta(@mmfunc,[S0 0 0],1,0,100);

S=y(:,1);
ES=y(:,2);
E=E0-ES;
P=y(:,3);
subplot(311);
plot((t),S,'.-r',(t),E,'.-b',(t),ES,'.-g',(t),P,'.-c');
legend('S','E','ES','P');
xlabel('Time (s)');
ylabel('Concentration (M)');
subplot(312);
plot(log10(t),S,'.-r',log10(t),E,'.-b',log10(t),ES,'.-g',log10(t),P,'.-c');
legend('S','E','ES','P');
xlabel('log(Time (s))');
ylabel('Concentration (M)');

subplot(313);
vmax=k2*E0;
Km=(k_1+k2)/k1;
v_real=k2*ES;
v_predicted=(vmax*S)./(Km+S);
plot(t,v_real,'.-r',t,v_predicted,'.-b');
legend('Calculatd turnover rate v','Predicted turnover rate v0');
xlabel('Time (s)');
ylabel('Turn-over rate (M/s)');
title('Deviation of psuedo-steady state approximation')
% plot(log10(t),v_real,'.-r',log10(t),v_predicted,'.-b');
% legend('Calculatd turnover rate v','Predicted turnover rate v0',0);
% xlabel('log(Time) (s)');
% ylabel('Turn-over rate (M/s)');
```

Different time step sizes are supposed to be used and relevant results would be shown and compared in the below.

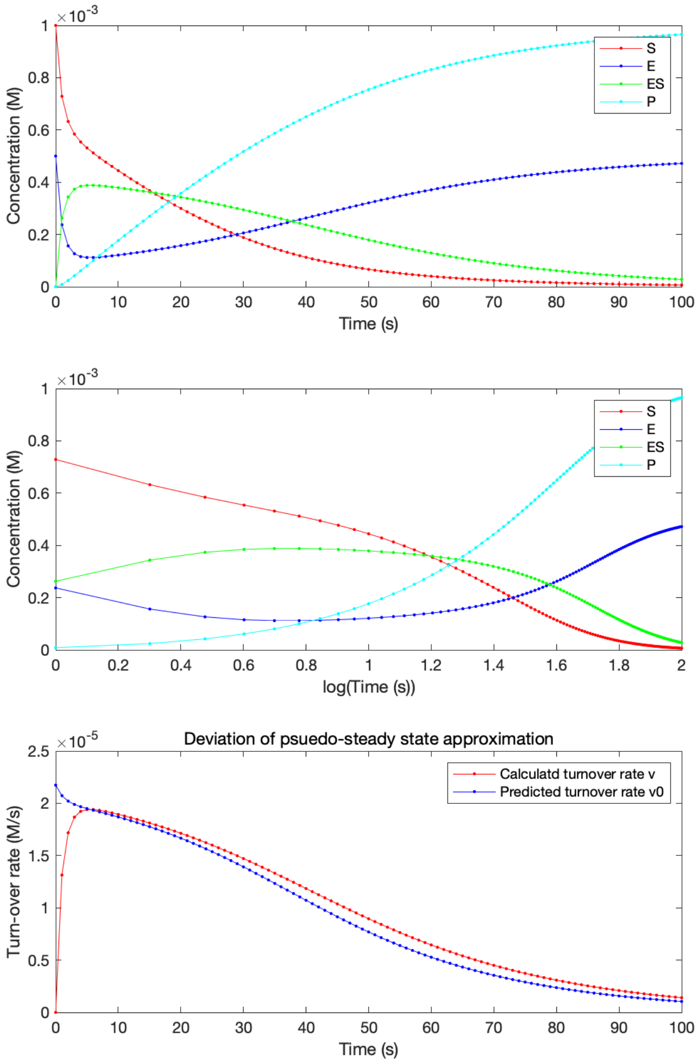
time step = 0.1 s my solver



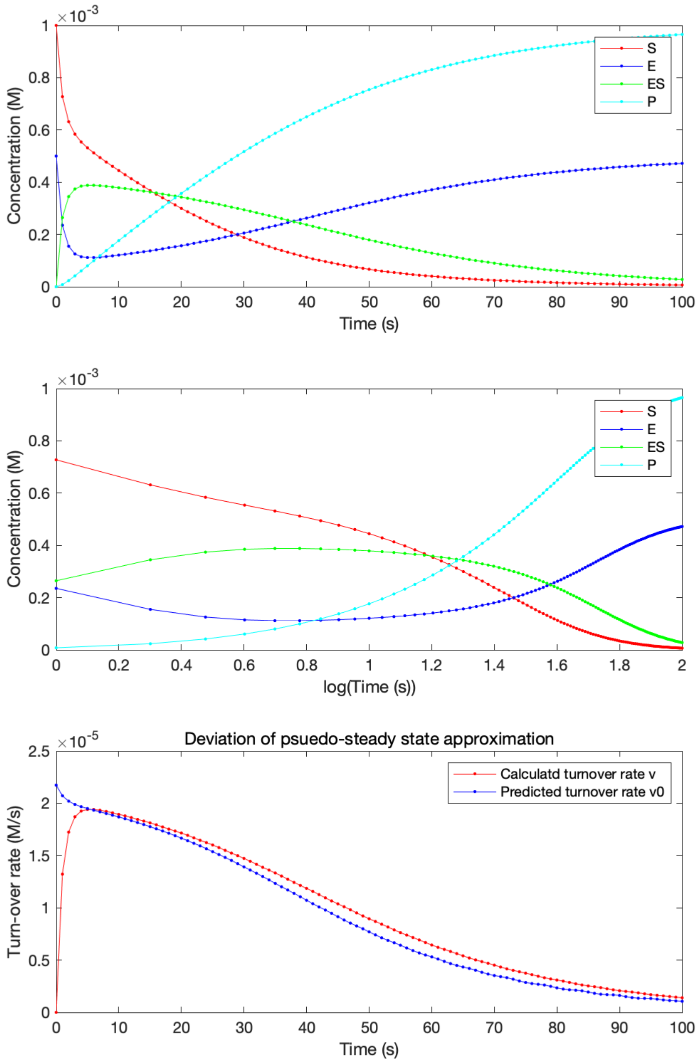
time step = 0.1 s ode45



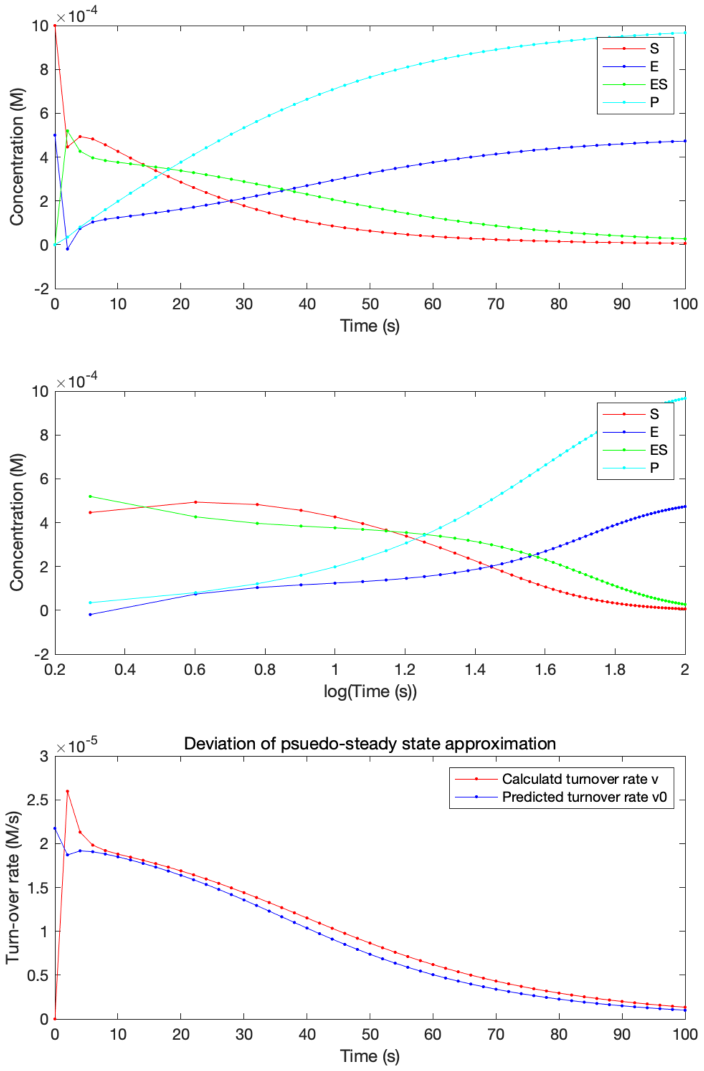
time step = 1.0 s my solver



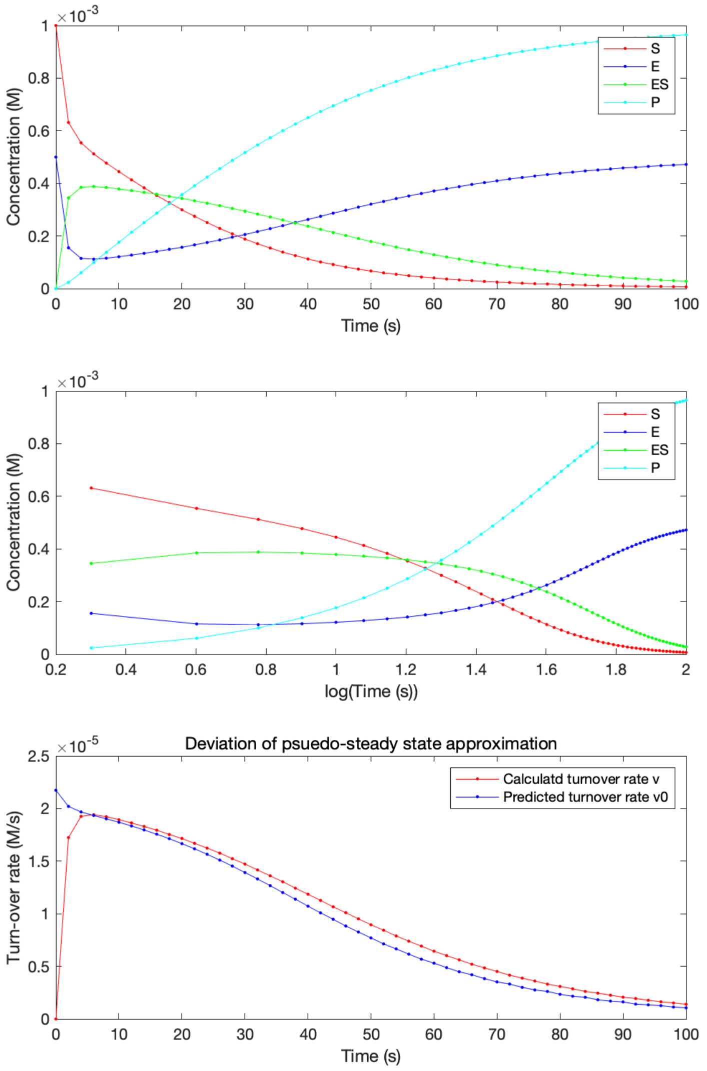
time step = 1.0 s ode45



time step = 2.0 s my solver



time step = 2.0 s ode45



As the results show us, if we choose smaller step length, the figures which are plotted recording to my own ode algorithm would be much more similar to the figures plotted through ode45. Because in the principle of calculation, the smaller step size means we divide the same time interval into more tiny time intervals. That is, let the error become smaller and smaller, making the results closer to exact value. Only when the step size is almost equal to zero, my own ode code can be meaningful.

Question2:

For equilibrium binding assumption n identical and independent binding sites, please proof (do verification if you cannot proof) the conclusion (in the slide no. 11 of week01-02 ppt) $r = \dfrac{nK[S]}{1+K[S]}$ based on Adair equation and other related information on the week01-02 ppt.

Proof :

Previously it was assumed that one substrate molecule binds to one enzyme.

In biological reactions proteins often bind multiple substrates.

$$r := \frac{[S]_{bound}}{[P]_0} = \frac{[P_1]+2[P_2]+3[P_3]+\ldots+n[P_n]}{[P_0]+[P_1]+[P_2]+[P_3]+\ldots+[P_n]}$$

Assume protein P has n binding sites, Pj denotes protein bound to j substrates S and in the steady state we can get :

$$\frac{d[P_{j-1}]}{dt} = 0 = -k_{+j}[P_{j-1}][S] + k_{-j}[P_j]$$

So :

$$K_j = \frac{k_{+j}}{k_{-j}} = \frac{P_j}{[P_{j-1}][S]}$$

Correspondingly:

$$P_j = K_j[P_{j-1}][S] = K_jK_{j-1}[P_{j-2}][S] = K_jK_{j-1}\ldots K_2K_1[P_0][S]^j$$

Then :

$$r = \frac{K_1[P_0][S]+2K_2K_1[P_0][S]^2+\ldots+nK_nK_{n-1}\ldots K_2K_1[P_0][S]^n}{[P_0]+(K_1[P_0][S]+K_2K_1[P_0][S]^2+\ldots+K_nK_{n-1}\ldots K_2K_1[P_0][S]^n)} = \frac{\sum_{i=1}^{n} i[S]^i(\prod_{j=1}^{i}K_j)}{1+\sum_{i=1}^{n}[S]^i(\prod_{j=1}^{i}K_j)}$$

If intrinsic association constant K is defined as: $K := \dfrac{k_+}{k_-}$

There are $(n-(j-1))$ possible unbinding sites on P_{j-1}, and if one side is occupied by a substrate, P_{j-1} become P_j . Also, there are j possible sites in the enzyme could loose a substrate.

So:

$$jk_-[P_j] = (n-(j-1))k_+[P_{j-1}][S]$$

Thus evolves:

$$K_j = \frac{(n-(j-1))k_+[P_{j-1}][S]}{jk_-[P_{j-1}][S]} = \frac{(n-(j-1))K[P_{j-1}][S]}{j[P_{j-1}][S]} = \frac{(n-(j-1))K}{j}$$

Then :

$$r = \frac{\sum_{i=1}^{n} i[S]^i K^i (\prod_{j=1}^{i} \frac{n+1-j}{j})}{1+\sum_{i=1}^{n} [S]^i K^i (\prod_{j=1}^{i} \frac{n+1-j}{j})}$$

Since $\binom{n}{i} = \frac{n!}{i!(n-i)!}$, r may be written in the form :

$$r = \frac{\sum_{i=1}^{n} i\binom{n}{i}[S]^i K^i}{1+\sum_{i=1}^{n} \binom{n}{i}[S]^i K^i}$$

Applying the binomial rule, the denominator can be converted as$(1+K[S])^n$. For the numerator the derived binomial rule applies :

$$\sum_{i=1}^{n} i\binom{n}{i}[S]^i K^i = \sum_{i=1}^{n} i\frac{n!}{i!(n-i)!}[S]^i K^i = \sum_{i=1}^{n} n\frac{(n-1)!}{(i-1)!(n-i)!}[S]^i K^i$$

let $q = i-1$:

$$\sum_{i=1}^{n} i\binom{n}{i}[S]^i K^i = n\sum_{q=0}^{n-1} \frac{(n-1)!}{q!(n-1-q)!}[S]^{q+1}K^{q+1} = n[S]K\sum_{q=0}^{n-1} \frac{(n-1)!}{q!(n-1-q)!}[S]^q K^q$$

So :

$$r = \frac{n([S]K)(1+[S]K)^{n-1}}{(1+[S]K)^n} = \frac{nK[S]}{1+K[S]}$$

Question3:

Dilution of proteins due to cell growth

a. A single bacterial cell at time t=0 has volume V_0 . After a time interval Td, the doubling time, the cell grows and divides into two cells, each of volume V_0; after another interval Td, there are four cells and so on.

b. Show that the combined volume of cells at time t may be written as $V(t) = V_0 e^{\gamma t}$.

Find γ in the terms of Td.

The protein X is created at some rate k(t), so that the total number of molecules of X satisfies $\dfrac{dn}{dt} = k(t)$. Show the concentration $[X] = \dfrac{n}{V}$ satisfies $\dfrac{d[X]}{dt} = \dfrac{k(t)}{V} - \gamma[X]$. Discuss the origin of the decay term.

Find γ in the terms of Td :

We known that : $V(0) = V_0$, $V(Td) = 2V_0$, $V(2Td) = 4V_0$ and $V(nTd) = 2 * V((n-1)Td)$

Get : $V(nTd) = 2^n V_0$

So : $V(t) = V_0 2^{\frac{t}{Td}} = V_0 e^{ln2\frac{t}{Td}} = V_0 e^{\frac{ln2}{Td}t} = V_0 e^{\gamma t}$

We get : $\gamma = \dfrac{ln2}{Td}$

Proof of the concentration$[X] = \dfrac{n}{V}$ **satisfies** $\dfrac{d[X]}{dt} = \dfrac{k(t)}{V} - \gamma[X]$ **:**

$$\frac{d[X]}{dt} = \frac{d(\frac{n(t)}{V(t)})}{dt} = \frac{\frac{dn(t)}{dt}V(t) - \frac{dV(t)}{dt}n(t)}{V(t)\ *\ V(t)} = \frac{k(t) - \gamma n(t)}{V(t)} = \frac{k(t)}{V(t)} - \gamma\frac{n(t)}{V(t)} = \frac{k(t)}{V(t)} - \gamma[X]$$

Discuss the origin of the decay term:

The origin of the decay term is when the volume of them increases, which means they grow larger. Because when decay term occurs, the volume of cells increases while the protein synthesis is constant, thus the concentration of proteins is diluted with relative reduction.

c. In addition to the term derived in (b), there should be an extra term that takes into account the degradation of protein by proteinase, which we can model by the effective reaction $X \overset{\delta}{\to} \phi$.

Modify the equation in part (b) in include protein degradation.

Solution 1 :

If the ϕ is stable in the system :

From $X \overset{\delta}{\to} \phi$ and (b), we can get the equation: $\frac{d[\phi]}{dt} = \delta[X]$

For this question: $[X] = \frac{n}{V} - [\phi]$ & $\frac{dn}{dt} = k(t)$

So : $\frac{d[X]}{dt} = \frac{d(\frac{n(t)}{V(t)} - [\phi](t))}{dt} = \frac{\frac{dn(t)}{dt}V(t) - \frac{dV(t)}{dt}n(t)}{V(t)\ *\ V(t)} - \frac{d[\phi]}{dt} = \frac{k(t) - \gamma n(t)}{V(t)} - \delta[X]$

Finally we get :

$$\frac{d[X]}{dt} = \frac{k(t)}{V(t)} - \gamma\frac{n(t)}{V(t)} - \delta[X] = \frac{k(t)}{V(t)} - \gamma([X] + [\phi]) - \delta[X]$$

$$= \frac{k(t)}{V(t)} - (\gamma + \delta)[X] - \gamma[\phi]$$

Solution 2 :

If the ϕ is stable in the system :

If we set $[X\ \&\ \phi] = [X] + [\phi] = \frac{n}{V}$:

We can get from (b) that : $\frac{d[X\ \&\ \phi]}{dt} = \frac{k(t) - \gamma n(t)}{V(t)} = \frac{d[X]}{dt} + \frac{d[\phi]}{dt} = \frac{d[X]}{dt} + \delta[X]$

Thus :$\frac{d[X]}{dt} + \delta[X] = \frac{k(t) - \gamma n(t)}{V(t)}$

Finally we get :

$$\frac{d[X]}{dt} = \frac{k(t)}{V(t)} - (\gamma + \delta)[X] - \gamma[\phi]$$

Solution 3 :

If the ϕ is not stable in the system :

From $X \overset{\delta}{\rightarrow} \phi$ and (b), we can get the equation: $\frac{dn}{dt} = k(t) - \delta[X]V$

And

So : $$\frac{d[X]}{dt} = \frac{d(\frac{n(t)}{V(t)})}{dt} = \frac{\frac{dn(t)}{dt}V(t) - \frac{dV(t)}{dt}n(t)}{V(t)\ *\ V(t)} = \frac{k(t) - \delta[X]V - \gamma n(t)}{V(t)}$$

Finally we get :

$$\frac{d[X]}{dt} = \frac{k(t)}{V(t)} - \gamma[X] - \delta[X]$$

While $[\phi] = 0$ **:**

Solution 1 = Solution 2 = Solution 3

While $[\phi] \neq 0$ **:**

Solution 1 = Solution 2 ≠ Solution 3