3) Computer solution

Many computer software packages are available which solve sets of differential equations to yield concentrations as functions of time. Also, one may write his own simple program.

Recall the four differential equations:

$$-\frac{dS}{dt} = k_1[E][S] - k_{-1}[ES]$$

$$-\frac{dE}{dt} = k_1[E][S] - k_{-1}[ES] - k_2[ES]$$

$$\frac{dES}{dt} = k_1[E][S] - k_{-1}[ES] - k_2[ES]$$

$$\frac{dP}{dt} = k_2[ES]$$

Need initial conditions to solve: S_0, E_0, ES_0, P_0

A simple (!) method of solving is to approximate each differential equation with a difference equation:

$$\frac{dS}{dt} \approx \frac{\Delta S}{\Delta t} = \frac{S_{i+1} - S_i}{\Delta t}$$

$$\frac{dE}{dt} \approx \frac{\Delta E}{\Delta t} = \frac{E_{i+1} - E_i}{\Delta t}$$

$$\frac{dES}{dt} \approx \frac{\Delta ES}{\Delta t} = \frac{ES_{i+1} - ES_i}{\Delta t}$$

$$\frac{dP}{dt} \approx \frac{\Delta P}{\Delta t} = \frac{P_{i+1} - P_i}{\Delta t}$$

where i+1 refers to the value at a future time, and i refers to the value at the present time.

Thus, a new concentration (i+1) may be calculated from an old concentration (i):

$$\frac{S_{i+1} - S_i}{\Delta t} = -(k_1 E_i S_i - k_{-1} E S_i)$$

$$\frac{E_{i+1} - E_i}{\Delta t} = -(k_1 E_i S_i - k_{-1} E S_i - k_2 E S_i)$$

$$\frac{E S_{i+1} - E S_i}{\Delta t} = k_1 E_i S_i - k_{-1} E S_i - k_2 E S_i$$

$$\frac{P_{i+1} - P_i}{\Delta t} = k_2 E S_i$$

Or, in terms of i+1:

$$S_{i+1} = (-k_1 E_i S_i + k_{-1} ES_i)\Delta t + S_i$$

$$E_{i+1} = (-k_1 E_i S_i + k_{-1} ES_i + k_2 ES_i)\Delta t + E_i$$

$$*ES_{i+1} = (k_1 E_i S_i - k_{-1} ES_i - k_2 ES_i)\Delta t + ES_i$$

$$P_{i+1} = k_2 ES_i \Delta t + P_i$$

* or: $ES_{i+1} = E_0 - E_{i+1}$

For example, to calculate S, must step off time:

$$S_1 = (-k_1 E_0 S_0 + k_{-1} E S_0)\Delta t + S_0$$

$$S_2 = (-k_1 E_1 S_1 + k_{-1} E S_1)\Delta t + S_1$$

$$S_3 = (-k_1 E_2 S_2 + k_{-1} E S_2)\Delta t + S_2$$

$$S_4 = (-k_1 E_3 S_3 + k_{-1} E S_3)\Delta t + S_3$$

Solve the computer problem

(see Example 3.2, pp. 73-74)

$k_1 = 30$ L/g•min

$k_{-1} = 160$/min

$k_2 = 110$/min

$S_0 = 10.0$ g/L

$P_0 = 0$ g/L

$E_0 = 0.00875$ g enzyme/L

$ES_0 = 0$ g/L

```
/*

Program 1

Simple Enzyme Kinetics

*/

#include<stdio.h>
#include<conio.h>
main()
{
   FILE *fout;
   double kp1;                          //  units = L/g min
   double km1,k2;                       //  units = /minutes
   double Snew,Pnew,Enew,ESnew;         //  units = g/L
   double Sold,Pold,Eold,ESold;         //  units = g/L
   double deltatime;                    //  units = minutes
   double currenttime;                  //  units = minutes
   double endtime;                      //  units = minutes
   int count=0;

// Define Some Constants

   kp1 = 30;
   km1 = 160;
   k2 = 110.;
   endtime = 30;
   deltatime=0.00001;

// Set Initial Conditions

   Sold = 10.;
   Pold = 0.;
   Eold = 0.00875;
   ESold = 0.;
```

Must terminate program at some point. We will end at a time of 30 minutes.

Must use a small value for Δt. The smaller, the more accurate (careful of computational precision!). However, the smaller this value, the longer the program will take to run. Note: with this Δt and end time, the program will run through 3 million iterations (30/0.00001).

```
// Calculations

   fout=fopen("a:ex3a1.dat","w");
   currenttime = 0.;
   do
     {
     Snew = (-kp1*Eold*Sold+km1*ESold)*deltatime+Sold;
     Enew = (-kp1*Eold*Sold+km1*ESold+k2*ESold)*deltatime+Eold;
     ESnew = (kp1*Eold*Sold-km1*ESold-k2*ESold)*deltatime+ESold;
     Pnew = k2*ESold*deltatime+Pold;
     count++;
     if (count==25000)
       {
                fprintf(fout,"%7.2f %7.4f %7.4f %6.4f\n",currenttime,Snew,Pnew,(Sold-Snew)/deltatime);
                count=0;
       }
     Sold=Snew;
     Eold=Enew;
     ESold=ESnew;
     Pold=Pnew;
     currenttime+=deltatime;
     }
   while(currenttime<endtime);
   fclose(fout);
}
```

This is the meat of the program. Calculate the "new" values for each of the four components (e.g., S_1), given the "old" values for the components (e.g., S_0).

Send every 25000th time point (0.25 min) to a file.

Must update the values for each of the four components, so that next time through the loop, they will be the "old" values.

Results……..

Concentration [g/L]
Time [h]
S
P
0
2
4
6
8
10
0
5
10
15
20
25
30

Figure 1

Substrate and Enzyme Kinetics

Figure 1

Substrate and Enzyme Kinetics

Notes:

1) dP/dt is not a constant (see Figure 1a). The reaction rate (i.e., the "reaction velocity") decreases with time. The rate of this decrease depends on the substrate concentration and the value of K_M. If the substrate concentration is very high initially (compared with K_M), then the rate of decline in dP/dt will be low.

   When we conduct our experiment (i.e., in a beaker), we must try our best to calculate the initial value of dP/dt, because in general the value of dP/dt will decrease the moment the experiment begins!

Notes:

2) Equilibrium cannot occur immediately (since ES=0 initially). The speed to equilibrium depends on the enzyme but is often quicker than can be measured.

Notes:

3) dES/dt is not equal to zero (in very short time scale – Figure 1b – and in very large time scale – Figure 1c). Thus, strictly speaking, the quasi-steady state assumption of Briggs and Haldane is not valid.

It is invalid in a short time scale (up to 0.01 min in this particular case) because the reaction started with enzyme alone (ES=0).

Notes:

It is invalid at a long time scale because as the reaction proceeds, the concentration of S decreases. To maintain equilibrium, then, ES must decrease and/or E must increase. Since E+ES is a constant (E_0), both must occur.

$$K_D = \frac{k_{-1}}{k_1} = K_M' = \frac{[E][S]}{[ES]}$$

$$\frac{K_D}{[S]} \uparrow = \frac{[E] \uparrow}{[ES] \downarrow}$$

Notes:

4) Fortunately, we obtain the same parameters either from Michaelis-Menten or Briggs-Haldane (K_M and V_{MAX}), and they still very closely match the results from the differential equations. The time to reach equilibrium for most enzymatic reactions will be much shorter than we are capable of experimentally measuring.

Although the Michaelis-Menten and Briggs-Haldane equations may be developed from questionable premises, they work. Specifically, the two parameters (K_M and V_{MAX}) are readily calculable from simple experiments.

3. Determination of Michaelis-Menten Parameters

As described in Section 3.D.1, a series of batch experiments is conducted with different initial substrate concentrations, but with identical enzyme concentrations. Measure v_0 for each S_0.

The goal is to find the values for K_M and V_{MAX} which best fit the v_0 and S_0 data.

$$v_0 = \frac{V_{MAX} S_0}{K_M + S_0}$$

a. Hanes-Woolf Plot (also, Langmuir Plot)

Linearize the data in the following form:

$$\frac{S_0}{v_0} = \frac{K_M}{V_{MAX}} + \frac{S_0}{V_{MAX}}$$

Plot $\frac{S_0}{v_0}$ versus S_0 (y vs. x)

$$\text{Slope} = \frac{1}{V_{MAX}} \qquad \text{Intercept} = \frac{K_M}{V_{MAX}}$$

b. Lineweaver-Burke Plot

Linearize the data in the following form:

$$\frac{1}{v_0} = \frac{1}{V_{MAX}} + \frac{K_M}{V_{MAX}} \frac{1}{S_0}$$

Plot $\frac{1}{v_0}$ versus $\frac{1}{S_0}$

Slope = $\frac{K_M}{V_{MAX}}$ Intercept = $\frac{1}{V_{MAX}}$

c. Eadie-Hofstee Plot

Linearize the data in the following form:

$$v_0 = V_{MAX} - K_M \frac{v_0}{S_0}$$

Plot v_0 versus $\frac{v_0}{S_0}$

Slope = $-K_M$ Intercept = V_{MAX}

Each of these three linearizations would provide precisely “correct” results if there were no random error associated with the measurement of v_0 and S_0. However, since such errors always occur, each of the linearizations will inappropriately weight one region of the data while essentially excluding another region. The best method of determining K_M and V_{MAX} is by a non-linear curve fit.