# 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<sub>0</sub>, E<sub>0</sub>, ES<sub>0</sub>, P<sub>0</sub>

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

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

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

$$\frac{S_{i+1} - S_{i}}{\Delta t} = -(k_{1}E_{i}S_{i} - k_{-1}ES_{i})$$

$$\frac{E_{i+1} - E_{i}}{\Delta t} = -(k_{1}E_{i}S_{i} - k_{-1}ES_{i} - k_{2}ES_{i})$$

$$\frac{ES_{i+1} - ES_{i}}{\Delta t} = k_{1}E_{i}S_{i} - k_{-1}ES_{i} - k_{2}ES_{i}$$

$$\frac{P_{i+1} - P_{i}}{\Delta t} = k_{2}ES_{i}$$

## Or, in terms of i+1:

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

$$S_{i+1} = (-k_1E_iS_i + k_{-1}ES_i)\Delta t + S_i$$
 $E_{i+1} = (-k_1E_iS_i + k_{-1}ES_i + k_2ES_i)\Delta t + E_i$ 
 $*ES_{i+1} = (k_1E_iS_i - k_{-1}ES_i - k_2ES_i)\Delta t + ES_i$ 
 $P_{i+1} = k_2ES_i\Delta t + P_i$ 

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

$$S_{1} = (-k_{1}E_{0}S_{0} + k_{-1}ES_{0})\Delta t + S_{0}$$

$$S_{2} = (-k_{1}E_{1}S_{1} + k_{-1}ES_{1})\Delta t + S_{1}$$

$$S_{3} = (-k_{1}E_{2}S_{2} + k_{-1}ES_{2})\Delta t + S_{2}$$

$$S_{4} = (-k_{1}E_{3}S_{3} + k_{-1}ES_{3})\Delta t + S_{3}$$

# Solve the computer problem (see Example 3.2, pp. 73-74)

$$k_1 = 30 \text{ L/g} \cdot \text{min}$$
 $k_{-1} = 160/\text{min}$ 
 $k_2 = 110/\text{min}$ 
 $S_0 = 10.0 \text{ g/L}$ 
 $P_0 = 0 \text{ g/L}$ 
 $E_0 = 0.00875 \text{ g enzyme/L}$ 
 $ES_0 = 0 \text{ g/L}$ 

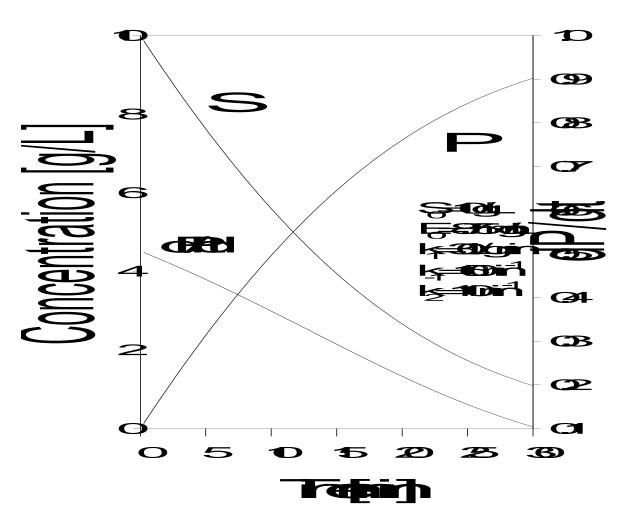
```
Program 1
Simple Enzyme Kinetics
*/
#include<stdio.h>
#include<conio.h>
main()
 FILE *fout;
 double kp1;
                                     // units = L/g min
 double km1,k2;
                                     // units = /minutes
 double Snew, Pnew, Enew, ESnew; // units = g/L
 double Sold, Pold, 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.

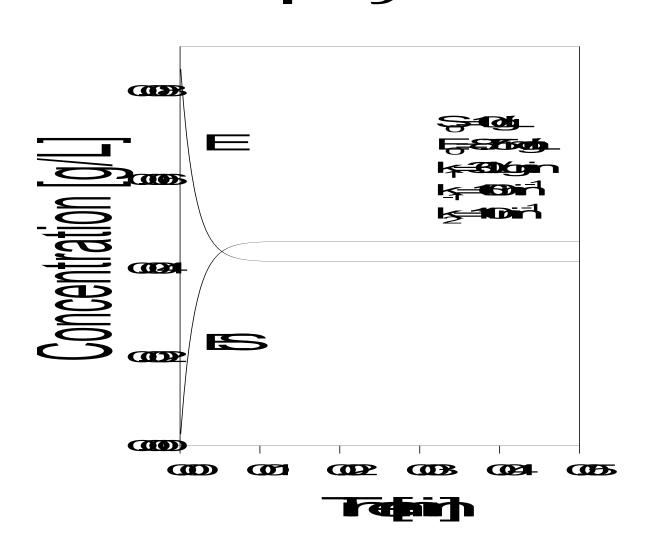
Must use a small value for  $\Delta t$ . The smaller, the more accurate (careful of computational precision!). However, the smaller this value, the longer the program will take to run. Note: with this  $\Delta t$  and end time, the program will run through 3 million iterations (30/0.00001).

```
// Calculations
                                                                  This is the meat of the program.
                                                                  Calculate the "new" values for each of
 fout=fopen("a:ex3a1.dat","w");
                                                                  the four components (e.g., S_1), given the
 currenttime = 0.;
                                                                  "old" values for the components (e.g., S_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;
                                                                          Send every 25000th time point
   Pnew = k2*ESold*deltatime+Pold;
   count++;
                                                                          (0.25 \text{ min}) to a file.
   if (count==25000)
              fprintf(fout, "%7.2f %7.4f %7.4f %6.4f\n", currenttime, Snew, Pnew, (Sold-Snew)/deltatime);
              count=0;
   Sold=Snew:
                                                           Must update the values for each of the
   Eold=Enew:
                                                           four components, so that next time through the
   ESold=ESnew:
   Pold=Pnew:
                                                           loop, they will be the "old" values.
   currenttime+=deltatime;
 while(currenttime<endtime);
 fclose(fout);
```

Results.....

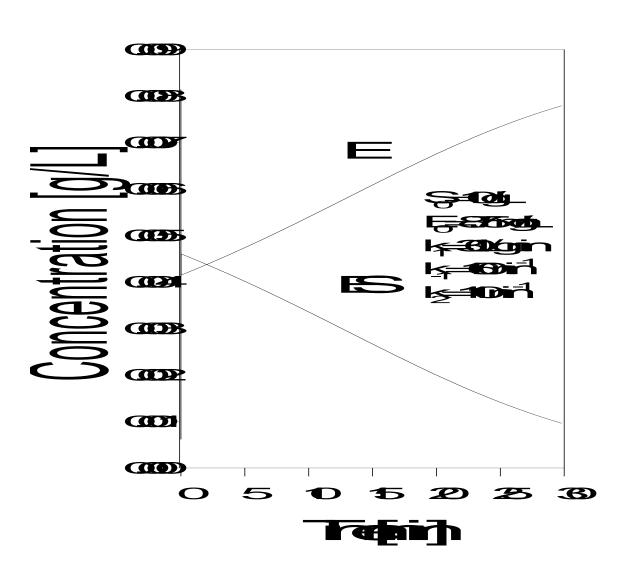


# 



#### FUEL

## SHOWINGS



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<sub>M</sub>. If the substrate concentration is very high initially (compared with K<sub>M</sub>), 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!

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.

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).

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.

$$K_{D} = \frac{k_{-1}}{k_{1}} = K_{M}' = \frac{[E][S]}{[ES]}$$

$$\frac{K_{D}}{[S]} = \frac{[E]^{\dagger}}{[ES]}$$

4) Fortunately, we obtain the same parameters either from Michaelis-Menten or Briggs-Haldane (K<sub>M</sub> and V<sub>MAX</sub>), 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)

Slope = 
$$\frac{1}{V_{MAX}}$$
 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.