EE2 2004
University of Iceland

Hannes Jonsson
Kristján Friðrik Alexandersson

C. Numerical Solution of Rate Equations

C1 Solution using ode45

Matlab has on offer many ordinary differential equation (ode) solvers, each using a different numerical method. ode45 is the one that should be used first and is usually sufficient for most problems. For most differential equations an analytical solution cannot be obtained but a numerical solution can almost always be obtained. The arguments for ode45 are the not the same as those one would use for dsolve. The initial values are defined separately and the range for which the differential equations are solved is also defined within the function. Open the help file for ode45 to learn more. Returning to the sequential first order reactions

$$A \rightleftharpoons B \rightleftharpoons C$$

which was solved analytically in the previous exercise, the numerical solution can be obtained in the following way. Note that a numerical value to all constants need to be given first before calling ode45.

First we define a function that calculates dA/dt, dB/dt and dC/dt from the concentration of each species. This is the system of differential equations, y' = f(t,y), that is mentioned in the help file. These are the same equations that we used in the last exercise.

```
------------------------------------------------------------
function dy=hvarfabc(t,y)

k1=1;
k2=0.1;
km1=0.05;
km2=0.1;

dy = zeros(3,1);    % a column vector

dy(1)=-k1*y(1)+km1*y(2);
dy(2)=k1*y(1)-km1*y(2)-k2*y(2)+km2*y(3);
dy(3)=k2*y(2)-km2*y(3);
------------------------------------------------------------
```

Now we can use the ode45 function to see what happens between t=0 and t=30, with only species A present at t=0.

```
EDU>> [t,y]=ode45(@hvarfabc,[0,30],[1,0,0])
```

The @ symbol creates a function handle. A function handle basically captures all the information about a function that Matlab needs to execute that function. The output is time column vector, followed by a matrix containing 3 concentration column vectors. We can now plot these results in the normal way.

```
EDU>> plot(t,y(:,1),t,y(:,2),t,y(:,3))
```

Q1: Do a calculation of the concentrations as a function of time for the reaction

$$A \rightleftharpoons B \rightleftharpoons C \rightleftharpoons D \rightleftharpoons E$$

assuming that only chemical A is present initially at a concentration of $[A]_0=1$. It would not be practical to solve this problem analytically. Remember to define and give numerical value to all rate constants. Choose some reasonable values for the rate constants (on the order of 0.1 to 10).

C2 Solution using Euler's Method; a simple first order

reaction

It is instructive to see how numerical methods for solving differential equations work. This way you will be able to write a program yourself for solving differential equations. Here you will learn how to use a simple finite difference algorithm for solving some of the kinetics equations discussed previously.

By using a finite difference approximation for the first derivative

$$\frac{dA}{dt} \approx \frac{A(l+h) - A(l)}{h}$$

where h is a small and yet not too small time interval (the so called 'time step'), the first order rate law for the reaction A -> products

$$\frac{dA}{dt} = -kA$$

becomes a recursion relation

$$A_{n+1} = A_n - khA_n$$

which can be used to generate a numerical value for the concentration A at points in time t = nh. The calculation can be carried out in Matlab in many different ways, but as usual it is useful to employ the "for-loop" here because then the Matlab code can easily be extended to more complex reactions. Note that it is essential to choose a numerical value for all constants before the calculation, since this is strictly a numerical solution of the differential equation. The total number of steps needed to cover the requested total time interval is evaluated by dividing the time with the time step and rounding off to nearest integer, then adding 1 because the first entry into the arrays corresponds to t=0.

```
---------------------------------------
clear k h nsteps a n t
close

k=0.05;
totaltime=100;
h=1;
a(1)=1;
t(1)=0;
nsteps=round(totaltime/h)+1;

for n=2:nsteps
   a(n)=a(n-1)-h*k*a(n-1);
   t(n)=t(n-1)+h;
end

plot(t,a)
xlabel('t')
ylabel('[A]')
---------------------------------------
```

Note that the data consists of discrete points, not a continuous curve.

Q2: It is essential to check how small the step size, h, needs to be in order to get accurate enough results. Repeat the calculation above for a smaller step size of h=0.5 and a larger step size of h=2. Plot the results in the same graph. For example you could do this by creating a function of h that creates vectors a and t by modifying the mfile above.
Note what value of the concentration of A is obtained at time t = 100 in each case. Let's assume that you are willing to accept an error of 0.001 in the concentration. Try using a very small step size, such as h = 0.02 and check what value of the concentration is obtained at t = 100. How small a step size do you need to use? Evaluate what concentration the analytical solution predicts at t = 100 and compare with your numerical

results.

C3 Sequential first order reactions with no back reactions

The numerical procedure above can rather easily be generalized to more complex reactions. For example, for the sequential first order reaction

$$A \rightarrow B \rightarrow C$$

the finite difference algorithm can be executed using the for loop in the following way

```
---------------------------------------------------------
clear k k1 k2 h nsteps a b c n t
close

k1 = 0.01;
k2 = 0.2;
totaltime=100;
h=0.5;
a(1)=1;
b(1)=0;
c(1)=0;
t(1)=0;
nsteps=round(totaltime/h)+1;

for n=2:nsteps
   a(n)=a(n-1)-h*k1*a(n-1);
   b(n)=b(n-1)+h*(k1*a(n-1)-k2*b(n-1));
   c(n)=c(n-1)+h*k2*b(n-1)
   t(n)=t(n-1)+h;
end

plot(t,a,t,b,t,c)
xlabel('t')
ylabel('M')
---------------------------------------------------------
```

C4 Solution using Euler's Method; sequential reversible reactions

Q3: Extend the Matlab code in section C3 to include back reactions. Calculate and plot the solution for a few different values of the back reaction rate constants.