File: /General/MLAB-Text/Papers/oscchem/oscchem.tex

An Oscillating Chemical Reaction

Gary D. Knott, Ph.D.
Civilized Software Inc.
12109 Heritage Park Circle
Silver Spring, MD 20906 USA
Tel: (301)962-3711
Email: csi@civilized.com

The possibility of oscillations in chemical kinetics depends upon the existence of sources and sinks for catalytic intermediates with very different time scales, resulting in very wide swings in their concentrations. The Belusov-Zhabotinsky reaction is perhaps the best known of the oscillatory chemical reactions. The following example is a gross simplification of the kinetics of the BZ reaction, retaining, however, the main features and properties.

In this example, there are five chemical reactions involving the chemicals y_1, y_2, y_3, y_4, y_5, and y_6.

$$y_1{+}y_3 \xrightarrow{k_1} y_2 \quad y_2{+}y_3 \xrightarrow{k_2} y_5 \quad y_1{+}y_2 \xrightarrow{k_3} y_4{+}2y_2 \quad y_2{+}y_2 \xrightarrow{k_4} y_6 \quad y_4 \xrightarrow{k_5} y_3$$

In these reactions, the rate constants k_1, k_2, k_3, k_4, and k_5 take on the following values.

$$k_1 = 1.34, \quad k_2 = 1.6 \cdot 10^9, \quad k_3 = 8000, \quad k_4 = 8 \cdot 10^7, \quad \text{and} \quad k_5 = 1.$$

The disposition of the products y_5 and y_6 of the second and fourth reactions is ignored. The initial concentrations of the four remaining chemicals are:

$$y_1(0) = .06, \quad y_2(0) = 5.01 \cdot 10^{-11}, \quad y_3(0) = 3.3 \cdot 10^{-7}, \quad y_4(0) = 2.4 \cdot 10^{-8}.$$

These reactions imply the following system of differential equations, (ignoring the products y_5 and y_6).

$$\begin{aligned}
dy_1/dt &= -y_1(k_1y_3 + k_3y_2) \\
dy_2/dt &= -y_2(k_2y_3 - k_3y_1 + k_4y_2) + k_1y_1y_3 \\
dy_3/dt &= -y_3(k_1y_1 + k_2y_2) + k_5y_4 \\
dy_4/dt &= -k_5y_4 + k_3y_1y_2
\end{aligned}$$

These equations, initial conditions, and rate constants are defined by the MLAB statements below.

```
k1 = 1.34; k2 = 1.6e9; k3 = 8e3; k4 = 8e7; k5 = 1
init y1(0) = .06; init y2(0) = 5.01e-11; init y3(0) = 3.3e-7; init y4(0) = 2.4e-8
fct y1'*t(t) = -y1*(k1*y3+k3*y2)
fct y2't(t) = -y2*(k2*y3-k3*y1+k4*y2)+k1*y1*y3
fct y3't(t) = -y3*(k1*y1+k2*y2)+k5*y4
fct y4't(t) = -k5*y4+k3*y1*y2
```

Because of the wide variation in the rate constants, these equations are very "stiff" in some regions. They are also oscillatory.

There are two other properties of these equations that make them difficult to solve numerically. First, there are very sharp "spikes" in y_2 and y_3; if the stepsize is not sufficiently small, the solver can skip over these spikes without "seeing" them. Thus we must either set `errfac` to a sufficiently small value to cause the stepsize used to be appropriately small or we must explicitly choose where the integrator should "land" and specify these times to be so-called "mandatory" points by setting `mandsw` to 1, or both. Second, these equations are unstable with respect to the numerical methods employed to solve them in certain regions of their common period. (This means that the eigenvalues of the Jacobian matrix of the linearized system are positive at some points during the solution time course and this implies "explosive" error is injected in the solution at these points.) This episodic instability can only be combatted by using a method with the largest-possible region of stability, together with a small step-size. This means we should use the Gear-Shrager (gear2) method with a small step-size as produced by choosing a sufficiently-small value for errfac.

```
method = gear2; errfac = 1e-8
```

This system has an oscillatory period of about 50 time units, so the integration period specified below in the list t includes that interval. MLAB generates its integration output only at the time-points that are specified by the user. Such values are generated by interpolation at the user-specified time points, unless mandatory points are specified, in which case, values are computed exactly at the specified points. Thus, in order to "catch" the very steep spikes in the output matrix, we must specify output time points during the spikes; otherwise, the sharp spikes would not be present in the output matrix, even though the integrator generated them correctly. Appropriate points to catch the spikes for the parameter values used in the differential equations above are listed in the 1-column matrix t below.

```
t =list(0:.9:.1,1:1.8:.2,2:5,10:40:5,41:46,46.5:48:.5,48.1:50:.1,50:55,60:70:5)
```

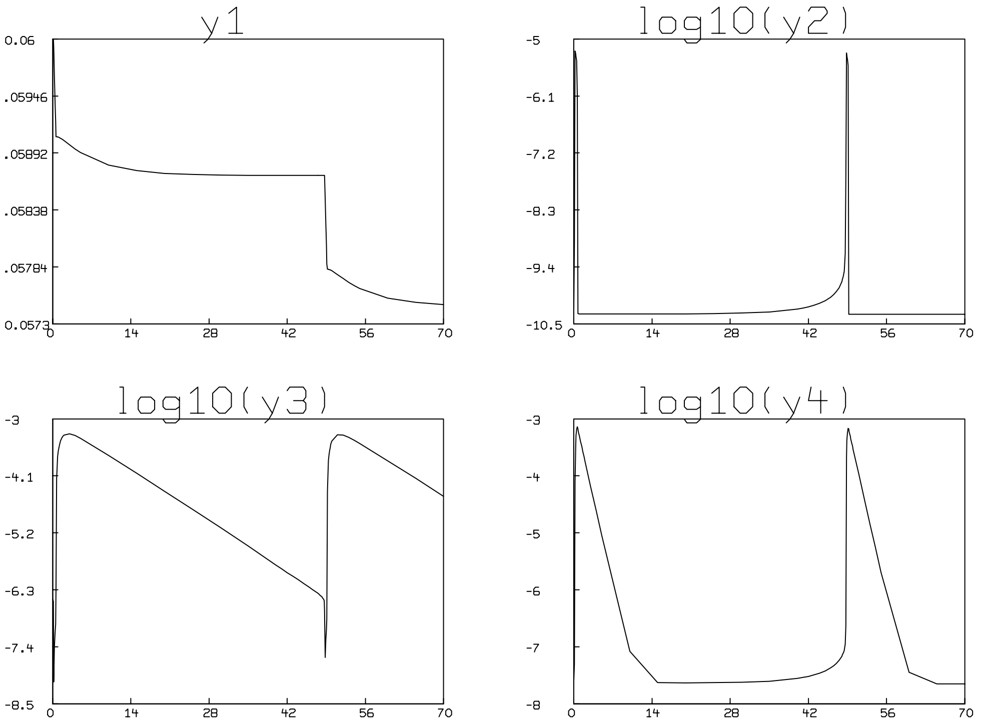
The solution is obtained using the integrate function.

```
m = integrate(y1't,y2't,y3't,y4't,t)
```

The solutions for y_1, y_2, y_3, and y_4 are in columns 2, 4, 6, and 8 of `m`, respectively. Because of the wide range of concentrations assumed in this problem, we take the base-10 logarithm of the y_2, y_3, and y_4 values before plotting.

```
l2 = log10 on m col 4; l3 = log10 on m col 6; l4 = log10 on m col 8

draw m col (1:2); top title "y1" size .05; frame 0 to .5, .5 to 1; w1=w
draw t&'l2; top title "log10(y2)" size .05; frame .5 to 1, .5 to 1; w2 = w
draw t&'l3; top title "log10(y3)" size .05; frame 0 to .5, 0 to .5; w3 = w
draw t&'l4; top title "log10(y4)" size .05; frame .5 to 1, 0 to .5; w4 = w
view
```


y1
log10(y2)
log10(y3)
log10(y4)