

Studying Nonlinear Chemical Dynamics with Numerical Experiments ©

John A. Pojman
Department of Chemistry & Biochemistry
University of Southern Mississippi
Hattiesburg, MS 39406-5043
john.pojman@usm.edu
www-chem.st.usm.edu/jap.html

© Copyright 1999 by the Division of Chemical Education, Inc., American Chemical Society. All rights reserved. For classroom use by teachers, one copy per student in the class may be made free of charge. Write to JCE Online, jceonline@chem.wisc.edu, for permission to place a document, free of charge, on a class Intranet.

Introduction:

Oscillations of chemical origin have been present as long as life itself. Every living system contains scores, perhaps hundreds, of chemical oscillators. The systematic study of oscillating chemical reactions and of the broader field of nonlinear chemical dynamics is, however, of considerably more recent origin. In this module we introduce some of the models used to describe oscillatory systems and methods to numerically analyze them.

For more information on the field of **nonlinear chemical dynamics**, check out the references at the end of this document and especially the March 1989 (vol. 66 no.3) of the *Journal of Chemical Education*, devoted to self-organization in chemistry. The recent textbook, *An Introduction to Nonlinear Chemical Dynamics* (I. R. Epstein & J. A. Pojman, 1998, Oxford University Press, New York) may also prove useful.

Goals

To provide students with opportunities to:

- 1) Learn about numerical integration of ordinary differential equations and how this can be used in chemical kinetics
- 2) Understand the application of phase plane analysis to dynamical systems
- 3) Explore models of oscillatory systems.

Objectives:

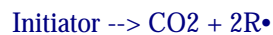
After completing the exercises in this document students should be able to:

- 1) identify potential limitations of numerical integration and be able to determine if a numerical result is valid
- 2) identify the attractor of a system using the time-delay method
- 3) test the pitfalls of numerical integration
- 4) use Mathcad to solve a set of differential equations derived from the corresponding chemical rate equations
- 5) explain the origins of oscillations in the Lotka-Volterra model and explain why this model can not be applied to chemical systems
- 6) numerically solve the Brusselator model of chemical oscillations and determine if its attractor is independent of the initial conditions
- 7) compare the predicted oscillation frequency dependence on initial concentrations in the Belousov-Zhabotinsky Oscillating Reaction to that of the Oregonator model. Compare both to experimental results

Solving differential equations numerically and using phase plane analysis.

Numerical Integration vs. Analytical Integration

Differential equations are extremely important in science. Very often we can write down a relationship that describes how some physical quantity changes with time. Lets consider a first-order irreversible reaction:



This is a general form for the thermal decomposition of a free-radical polymerization initiator, such as an acyl peroxide.

We write the rate equation as a differential equation in which I represents [Initiator]:

$$dI/dt = -k I$$

This differential equation says that the rate of disappearance of I is proportional to I. This is a typical first order reaction differential equation.

What we really want to know is the concentration of I at some future time. How do we go from an expression for the derivative with respect to time, dI/dt , to an explicit expression as a function of time, $I(t)$?

To go from a derivative expression to an explicit expression as a function of time must we integrate the differential equation. If we are lucky, there is an analytical expression, which means we can get it by using a pencil and paper and our knowledge of calculus or we can look it up in a book!. Either way we end up with

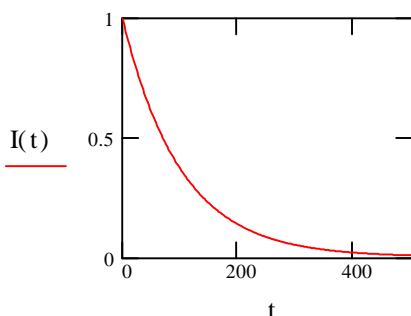
$$I(t) = I_0 \exp(-kt)$$

Our next question is, what does this equation look like?

First we set up some parameters for displaying a graph of the equation. We assign a value to the initial concentration and the rate constant in the equation and write the equation in a form for Mathcad to process.

$$I_0 := 1 \quad k := 0.01 \quad k \text{ has units of } 1/\text{time} \quad t := 0, 2.. 500$$

$$I(t) := I_0 \cdot \exp(-k \cdot t)$$



Vary k and examine the behavior of the curve. Record your observations in your notebook. Relate your observations to the concept of rate of a chemical reaction.

What if we didn't know the answer or recall how to integrate it analytically?

In this case we can use Numerical Integration. **Numerical integration** is essentially an exercise in curve fitting. One uses information about the function to be integrated-- its value and possibly the values of its derivatives -- at certain points to approximate the function over some interval by functions that we know how to integrate, usually polynomials. In this document we will use Euler's Method.

In Euler's method, we utilize only the value of the function and its derivative at a single point and make the simplest approximation, i.e. that the function of interest, $I(t)$, can be represented by a straight line through the point with slope equal to the derivative of the function at that point. If we are given the initial value $I = I_0$ at $t = 0$, the value of I at $t = h$ is

$$I(h) = I_0 + hf(I_0)$$

where $f(I_0)$ is the derivative of the function $I(t)$ evaluated at I_0 . This amounts to approximating the instantaneous derivative as the slope of a line (in this case, the tangent to the curve $I(t)$ at $t=0$) over a finite interval. More details of the Euler method can be found in standard differential equations books.

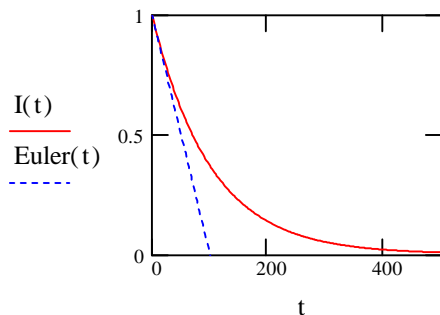
On your own. Start with $f(I) = \Delta I/\Delta t$ to show that $I(h) = I_0 + hf(I_0)$.

If we divide the entire interval over which we seek to solve the above equation into equal subintervals of length h , we can "march along," using the value of I obtained at one step to give us an approximation to I at the next step.

$$I(nh) = I((n - 1)h) + hf(I((n - 1)h))$$

As the figure below suggests, this approach is less than perfect. It gives the correct answer only if $f(I)$ is a constant, i.e., only if $I(t)$ is a straight line, but this, of course, is just what does not happen in nonlinear chemical dynamics. For a complex, non-monotonic function this problem can be quite serious. Even the qualitative features of the solution may not be approximated correctly.

$$\text{Euler}(t) := I_0 - k \cdot t$$



As we can see from this illustration, overshooting becomes a very serious problem for complex functions unless a very small Δt , i.e. h , is used.

An obvious way to deal with this problem is to make the time increment h smaller. If one looks at a small enough interval, any reasonably behaved function is well approximated by a straight line. One can, in fact, show that as h approaches zero the error due to approximating the function by a straight line also approaches zero and that this error is proportional to h . This fact and its simplicity make Euler's method seem attractive, and it is, if the functions to be integrated change relatively slowly. If we look at realistic examples in nonlinear chemical dynamics, however, we soon discover that choosing h small enough to give good accuracy means that the amount of computer time consumed becomes very large. This problem is exacerbated by the fact that we are typically integrating not a single equation but a set of many coupled equations.

Multi-step Methods

What can we do to improve the efficiency and accuracy of our numerical integration technique?

Perhaps the most important advance we can make is to use more of the information available to us about the function we want to integrate. If we knew its value and maybe its derivative not just at one point but at several, we could fit something more flexible than a line, perhaps a quadratic or a cubic, that would pass through the known points with the correct slopes and give us a better shot at getting the ups and downs of the function right. Methods that use information from several previous points or steps in the integration are known as multi-step methods. They play a key role in the numerical solution of nonlinear ordinary differential equations [Gear, 1971].

Sources of Error

Any numerical integration method has errors associated with it. These arise from several sources. For a particular differential equation, one kind of error may be more significant than the others, but a satisfactory method will ensure that none of these gets too large. We list here the sources of error in numerical integration. The reader may wish to consult more specialized texts, e.g., [Gear, 1971; Celia, 1992] for more details.

1. Discretization error arises from treating the problem on a grid of points ($t = 0, h, 2h, \dots$) rather than as a continuous problem on an interval. In effect, we have replaced the original differential equation by a difference equation whose exact solution at the grid points may not coincide with that of the differential equation, even if we are able to solve the difference equation to perfect accuracy. In most methods, this error is proportional to the step size h to some positive integer power. Thus, the discretization error can be reduced by decreasing the step size.
2. Truncation error results from using a function like a polynomial to approximate the true solution. To reduce truncation error, we could use more sophisticated functions, e.g., higher order polynomials, that require more information. Multi-step methods do this, but the additional accuracy gained by increasing the order of the method drops off very quickly beyond about four steps.

3. Roundoff error occurs because all computers have a finite precision. At each step of the computation, a small error is introduced. While these errors would tend to average out if the solution were linear and one could make statistical estimates of their magnitude; in general these errors accumulate in an unpredictable fashion. Decreasing h , the step size, to lower the discretization error means that there are more steps and hence a larger roundoff error. There is no way around this problem except to use a computer with very high precision. For most computers and nonlinear problems, double precision is the minimum requirement to obtain satisfactory results.

The above factors contribute a local error at each step of the calculation. As we move from point to point, the local errors combine to produce a global error. Good integration packages estimate the amount of error as the calculation moves along. The user specifies a tolerance that defines the magnitude of local error that is acceptable. If this tolerance cannot be met, h is automatically decreased; if satisfactory results can still not be obtained, another method, e.g., one with more steps, may be tried.

It is also important that a numerical method be stable. The concept here is very similar to the notion of stability for the steady state and other asymptotic solutions of differential equations. If we introduce a small change (error) into the numerical solution at a grid point, we would like that error to decay rather than to grow. Since this behavior will depend on the equation being solved, stability is typically defined with respect to a standard equation like $dA/dt = -kA$. Unstable methods are generally unsuitable for numerical computation, because even small roundoff errors can cause the calculated solution to "explode", i.e. to get so large that a floating point error occurs.

Users of this document should be sure they understand all terms used in the explanations up to this point. Things like double precision and floating point error are terms used by computer programmers that should be understood by all software users.

So let's solve our first-order differential equation numerically. We'll explain all the parameters in detail in the next example -- here we just want to show that we obtain the same result as from the analytical integration.

$t_{\max} := 500$ This is the time for the integration.

$npts := 5000$ This is the number of steps the integrator will take.

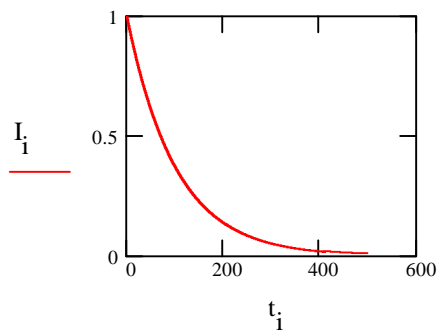
$y_0 := I0$ the initial condition with $y_0 = I0$

$D(t, y) := -k \cdot y_0$ the differential equation for rkfixed solver
we have only one differential equation

$s := rkfixed(y, 0, t_{\max}, npts, D)$ Here we are solving the differential equation.
Notice how we are using y instead of I as the variable name.

$t := s^{<0>}$ $I := s^{<1>}$ assigning columns in the matrix s to vectors characteristic of our problem. The matrix s is the solution for our problem.

$i := 0..npts - 1$ set up an index variable for displaying the i th point.



Compare this result with the graph for the function prepared at the beginning of this document. How does the numerical method compare to the analytic method?

A Simple Dynamical System

In this section we will examine a model for a weight attached to a spring that is attached to a wall. This is not a chemical system but it will allow us to observe some important concepts of dynamics before we examine nonlinear chemical systems. Here we assume the spring follows Hooke's law. Our system is a Simple Harmonic Oscillator, SHO.

The SHO has two dynamical variables that are functions of time, the position of the mass, x , and its velocity, v . To keep things simple we assume the mass is 1. Next we write some equations for the SHO system.

$$\frac{d}{dt}v = -k \cdot x$$

This says the velocity decreases because of the spring. The farther the spring is pulled, the stronger the restoring force. Recall that acceleration = F/m , and acceleration is the change in velocity with respect to time. This is one of the differential equations for our SHO. It is the second derivative of x with respect to t .

$$\frac{d}{dt}x = v$$

The change of the position with time is just the velocity. This is the second differential equation for our SHO. It is the first derivative of x with respect to time.

$$k := 90$$

This is the force constant for the spring.

Next we define some parameters for the numerical integration.

$$t_{\max} := 5$$

This is the time for the integration.

$$npts := 5000$$

This is the number of steps the integrator will take.

$$\frac{t_{\max}}{npts} = 1 \cdot 10^{-3}$$

We need to keep this ratio small or else the integrator will give spurious results. This is the step size for our analytic differential equation solution method.

$$v := 0$$

$$x := 1$$

These are the initial conditions, which mean the spring is pulled to position 1 but the mass is not moving.

$$y := \begin{bmatrix} v \\ x \end{bmatrix}$$

This is the vector of initial conditions with $y_0 = v$ and $y_1 = x$. Notice again how the differential equation solver is using y instead of x and v .

$$D(t, y) := \begin{bmatrix} -k \cdot y_1 \\ y_0 \end{bmatrix}$$

This is the vector of equations in which $dv/dt = -kx$ and $dx/dt = v$. Notice the use of y_1 for x and y_0 for v . The vector D contains the expressions for each differential equation that defines our SHO.

$$s := \text{rkfixed}(y, 0, t_{\max}, \text{npts}, D)$$

Mathcad has several different routines for integrating differential equations. We use the simplest, `rkfixed`, in which the value of the function is determined at fixed time steps. Recall our discussion above -- here $h = t_{\max}/\text{npts}$.

The result of the numerical solution to the differential equation is returned as the vector s . This vector has the values of t in column 0, the values of v in column 1, and the values of x in column 2. Remember Mathcad starts numbering rows and columns of matrices at 0.

Let's rename the columns variables we recognize.
NB: The s to a superscript, say $s^{<0>}$, means column zero.

$$t := s^{<0>}$$

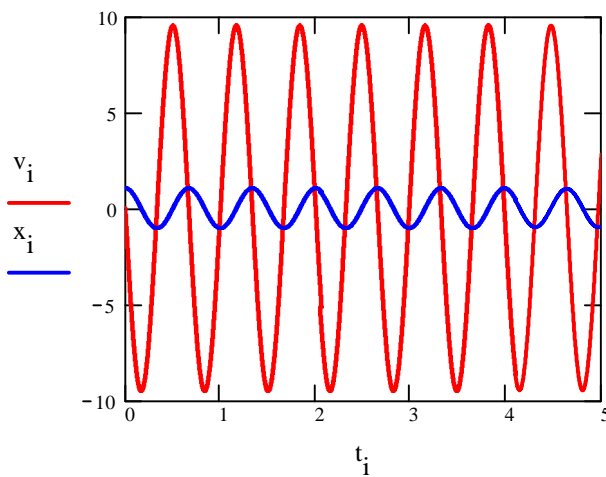
$$v := s^{<1>}$$

$$x := s^{<2>}$$

We set up an index variable that we can use to display the i th point.

$$i := 0.. \text{npts} - 1$$

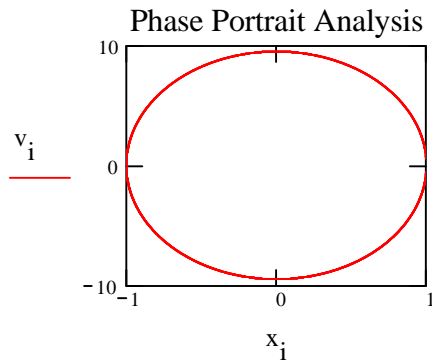
	0	1	2
0	0	0	1
1	$1 \cdot 10^{-3}$	-0.09	1
2	$2 \cdot 10^{-3}$	-0.18	1
3	$3 \cdot 10^{-3}$	-0.27	1
4	$4 \cdot 10^{-3}$	-0.36	0.999
5	$5 \cdot 10^{-3}$	-0.45	0.999
6	$6 \cdot 10^{-3}$	-0.54	0.998
7	$7 \cdot 10^{-3}$	-0.63	0.998
8	$8 \cdot 10^{-3}$	-0.719	0.997
9	$9 \cdot 10^{-3}$	-0.809	0.996
10	0.01	-0.899	0.996
11	0.011	-0.988	0.995
12	0.012	-1.078	0.994
13	0.013	-1.167	0.992
14	0.014	-1.256	0.991
15	0.015	-1.345	0.99
16	0.016	-1.434	0.989
17	0.017	-1.523	0.987



You may be able to see in the plot that when the velocity is at a maximum when the position is zero. Confirm this using the matrix shown above. Click on the matrix to scroll through the data.

Phase Plane Analysis

In "Phase Plane" or "Phase Portrait" analysis we plot one dynamical variable against another one. In this way we can see the "phase" relationship between that pair of dynamical variables, such as when one is high is the other low?



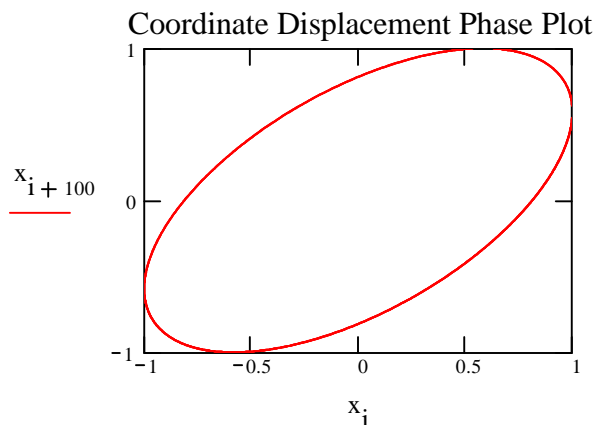
It should be clear from this diagram that the object reaches maximum (or minimum) velocity as it passes through the zero position. As the object moves on, the spring slows it down.

Now for a trick question and quick reply. What if we can't measure the velocity but can measure the position. Can we still construct a phase plot?

Yep! We do this by plotting the position at a later time versus the position now.

$i := 0..npts - 2010$

We need to drop off some points to keep the plot in range. Here we are ignoring 2010 points.

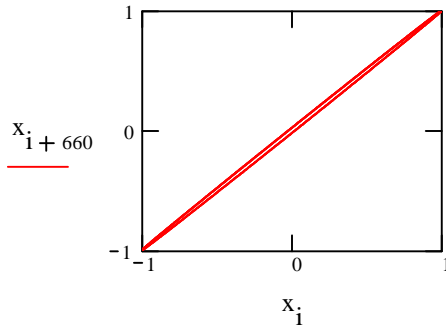


We plotted the $i + 100^{\text{th}}$ point vs. the i^{th} point. The method works! The shape is not exactly the same but we capture the essential idea that periodic motion occurs. This will prove useful for doing experiments when we can only measure one variable but "want to reconstruct the attractor".

The system exhibits a **limit cycle**, which is a type of an attractor. (An **attractor** is an object in phase space that the system tends to in the long term.) A limit cycle indicates that the system exhibits periodic behavior. Notice that the plot in the phase plane closes back on itself. This indicates that the same behavior occurs over and over again.

If you compare the v-x plot to the time delay plot above, they don't look exactly alike but they have the same topology, i.e. they have the same property, namely, being a closed curve.

Can we use just any delay? Not exactly



Choosing a time delay of 660 steps, where each step is 0.001 time units, gives us a delay almost exactly equal to the period of oscillation. Note the results at the left.

Moral: Don't chose a time delay = period of oscillation (or a multiple of it).

On your own:

What happens to the period of the oscillation if you make the spring stiffer, i.e., increase the value of k? (You can do this by returning to the top of the document where we defined "k".) Does the phase relationship between the velocity and position change? Record your observations in your notebook.

Another Example

What other type of attractors are there? What can we see if we modify our system to be more realistic? A real oscillator can't go on forever (unless we continuously input energy) because frictional forces will damp it out. So let's add frictional drag to the system.

$$\frac{d}{dt}v = -k \cdot v - \text{drag} \cdot x$$

This says the velocity decreases because of the spring and frictional drag.

$$\frac{d}{dt}x = v$$

This defines velocity as we did above.

$$k := 90 \quad \text{drag} := 5$$

This sets the force constant and the coefficient that accounts for friction.

$$t_{\text{max}} := 5$$

This is the time for the integration.

$$\text{npts} := 5000$$

This is the number of steps the integrator will take

$$\frac{t_{\max}}{npts} = 1 \cdot 10^{-3}$$

Here is the step size again.

$v := 0$ $x := 1$ The initial conditions for the oscillation.

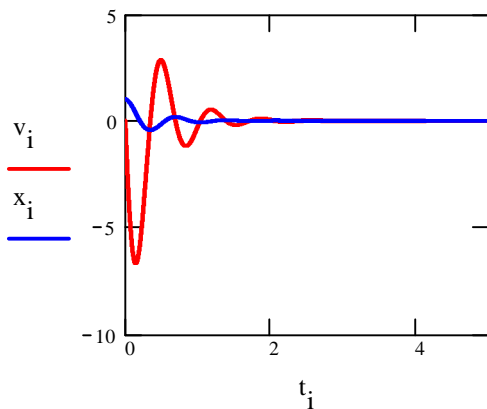
$y := \begin{bmatrix} v \\ x \end{bmatrix}$ The vector of initial conditions for the oscillation.

$D(t, y) := \begin{bmatrix} -k \cdot y_1 - \text{drag} \cdot y_0 \\ y_0 \end{bmatrix}$ The vector of differential equations. Compare this to the previous vector of this type. Notice how differential equations are arranged in this vector. Be sure you understand the relationship between the elements in the D vector and the differential equations for our SHO with drag.

$s := \text{rkfixed}(y, 0, t_{\max}, npts, D)$ Using the differential equation solver.

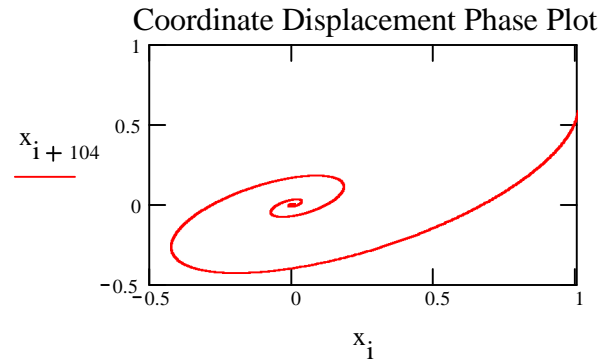
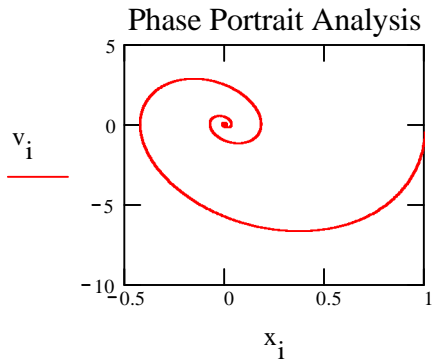
$t := s^{<0>}$ $v := s^{<1>}$ $x := s^{<2>}$ Set matrix s columns to vectors for time, velocity, and position.

$i := 0.. npts - 1$ Here we set the counter for the elements of each vector



You can see that the oscillations damp out. The asymptotic (long term) behavior is the mass sitting at $x = 0$ as we expect.

$i := 0.. npts - 110$



These phase plots show that the attractor is now a **fixed point**. This indicates that the system that does not change with time.

On your own:

What sort of attractor you would expect for an oscillating reaction in a beaker? Explain.

Models of Oscillatory systems

Now we move on to consider an ecological model, a predator-prey system, described by the Lotka-Volterra model [D'Ancona (1954), Lotka (1920) and Volterra (1926)]. In the system, rabbits are the prey, and lynxes (a type of small bobcat) are the predator. We need to define the system first.

The model is:

grass + rabbits \rightarrow 2rabbits k_1

rabbits + lynxes \rightarrow 2lynxes k_2

lynxes \rightarrow dead lynxes k_3

k_1 and k_2 are characteristic of the reproductive rates of rabbits and lynxes, respectively. k_3 reflects the death rate of the lynxes. We assume rabbits only die when they are eaten.

NB: "rabbits" and "lynxes" means "number of rabbits", etc.

The equations indicate that the rabbit population grows faster when there is plenty of grass and depends on the current population of rabbits. The rabbit population grows **autocatalytically** (is 'self-catalyzed'). This is a sufficient but not necessary characteristic of all the oscillatory systems we will examine. This means the rabbit population grows faster and faster and would continue to infinity (assuming enough grass to eat) were it not for the wolves who eat them.

The lynx population also grows autocatalytically and depends on the number of rabbits available to eat.

First we need to write down the differential equations that describe this system. The model consists of three irreversible steps. X is the population of rabbits, which reproduce autocatalytically. G is the amount of grass, which we assume to be constant, or at least in great excess compared to its consumption by the rabbits. Y is the population of lynxes (bobcats), and P represents dead lynxes.

$G + X \rightarrow 2X$ k_1

$X + Y \rightarrow 2Y$ k_2

$Y \rightarrow P$ k_3

As indicated above, each step is irreversible. Rabbits will never turn back into grass, nor dead lynxes into live ones. We can write down a system of differential equations to describe the behavior of the predator and prey species, and we assume the amount of grass, g , remains constant.

$$dx/dt = k_1 g x - k_2 xy$$

$$dy/dt = k_2 xy - k_3 y$$

where k_1 is a rate constant reflecting how fast rabbits reproduce, k_2 specifies how fast lynxes reproduce given a number, x , of rabbits to eat, and k_3 indicates the mortality rate of lynxes. For any set of these constants, the numbers of rabbits and lynxes will oscillate with a period that depends upon k_1 , k_2 , k_3 and g . The "net reaction," which proceeds monotonically, is the conversion of grass, which is assumed to grow as fast as it is consumed, into dead lynxes.

Let's define the initial conditions:

$$\text{rabbits}_{\text{init}} := 1.5 \quad \text{lynxes}_{\text{init}} := 1.0 \quad k_1 g := 2.4 \quad k_2 := 4.2 \quad k_3 := 5.1$$

These define the differential equations in terms of a column matrix, D , and y is column matrix of x and y .

$$D(t, y) := \begin{bmatrix} k_1 g \cdot y_0 - k_2 \cdot y_0 \cdot y_1 \\ k_2 \cdot y_0 \cdot y_1 - k_3 \cdot y_1 \end{bmatrix} \quad y := \begin{bmatrix} \text{rabbits}_{\text{init}} \\ \text{lynxes}_{\text{init}} \end{bmatrix} \quad \text{This you should be able to do based on what was done twice before above in the document.}$$

First we test out the Runge-Kutta integration (the rk in the $rkfixed$ equation solver) call that keeps a fixed integration time step but with relatively few steps.

$$\text{npts} := 100 \quad \text{tmax} := 10 \quad \text{setting some initial parameters}$$

$$\frac{\text{tmax}}{\text{npts}} = 0.1 \quad \text{Once again we compute the step size.}$$

s := rkfixed(y, 0, tmax, npts, D)

use the differential equation solver as before

Matrix s has 3 columns with time in the first column, rabbits at each time step in the second, and lynxes in the third.

t := s^{<0>} rabbits := s^{<1>}

lynxes := s^{<2>}

s =

	0	1	2
0	0	1.5	1
1	0.1	1.231	1.065
2	0.2	1.007	1.021
3	0.3	0.853	0.904
4	0.4	0.764	0.76
5	0.5	0.727	0.624
6	0.6	0.729	0.508
7	0.7	0.764	0.417
8	0.8	0.828	0.349
9	0.9	0.918	0.302
10	1	1.035	0.273
11	1.1	1.177	0.261
12	1.2	1.34	0.266
13	1.3	1.517	0.291
14	1.4	1.69	0.342
15	1.5	1.829	0.431
16	1.6	1.888	0.568
17	1.7	1.824	0.747
18	1.8	1.629	0.929
19	1.9	1.363	1.047

rabbits =

	0
0	1.5
1	1.231
2	1.007
3	0.853
4	0.764
5	0.727
6	0.729
7	0.764
8	0.828
9	0.918
10	1.035
11	1.177
12	1.34
13	1.517
14	1.69
15	1.829
16	1.888
17	1.824
18	1.629
19	1.363

lynxes =

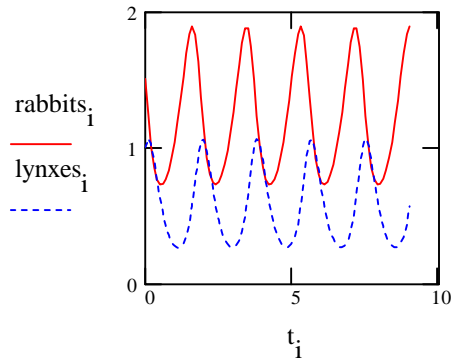
	0
0	1
1	1.065
2	1.021
3	0.904
4	0.76
5	0.624
6	0.508
7	0.417
8	0.349
9	0.302
10	0.273
11	0.261
12	0.266
13	0.291
14	0.342
15	0.431
16	0.568
17	0.747
18	0.929
19	1.047

t =

	0
0	0
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
6	0.6
7	0.7
8	0.8
9	0.9
10	1
11	1.1
12	1.2
13	1.3
14	1.4
15	1.5
16	1.6
17	1.7
18	1.8
19	1.9

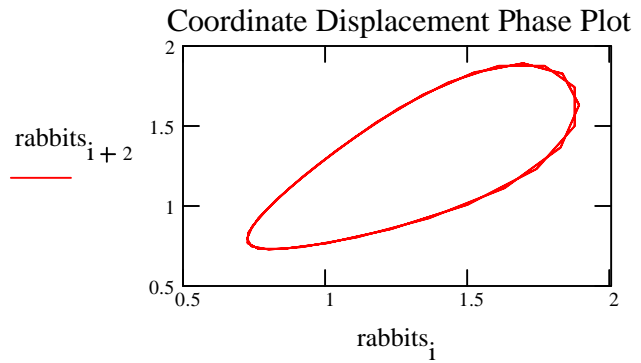
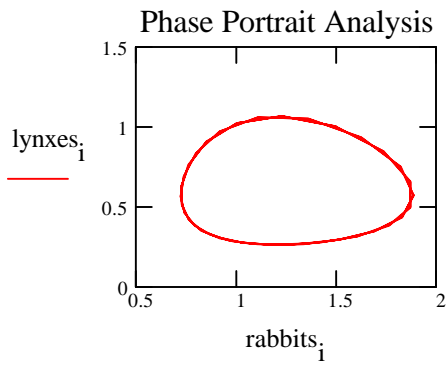
We plot the data versus time:

$i := 0..npts - 10$



Now we do our phase plots.

If we can only measure the rabbit population, we can use the time delay technique to reconstruct the attractor.



Recall in our discussion of sources of error that the time step was important. Let's examine the effect of reducing the number of points and thus increase the time step, $tmax/npts$:

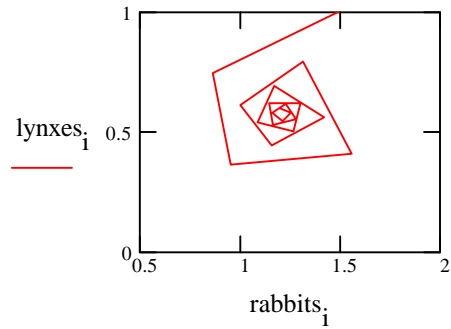
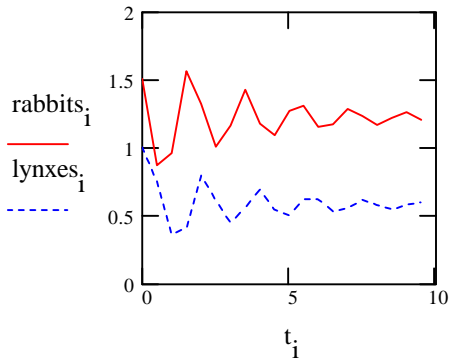
$$npts := 20 \quad tmax := 10 \quad \frac{tmax}{npts} = 0.5$$

$s := rkfixed(y, 0, tmax, npts, D)$

$t := s^{<0>}$ $\text{rabbits} := s^{<1>}$ $\text{lynxes} := s^{<2>}$

Plot the data as a function of time

$i := 0.. \text{npts} - 1$



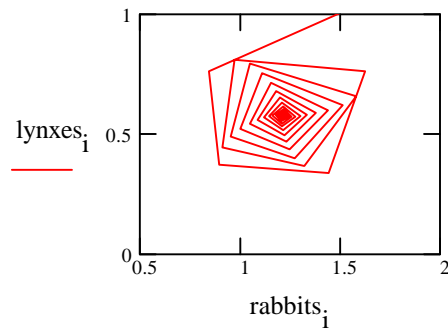
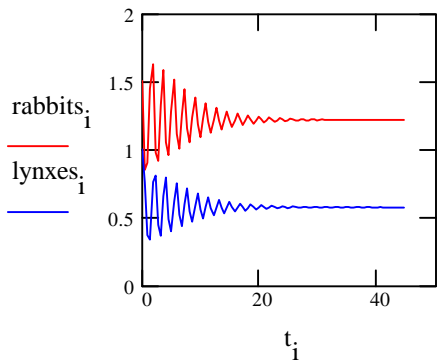
On your own: Predict what would happen after a long time. Record your prediction in your notebook.

Let's do it again with a long time period.

$\text{npts} := 100$ $\text{tmax} := 45$ $\frac{\text{tmax}}{\text{npts}} = 0.45$

$s := \text{rkfixed}(y, 0, \text{tmax}, \text{npts}, D)$

$t := s^{<0>}$ $\text{rabbits} := s^{<1>}$ $\text{lynxes} := s^{<2>}$ $i := 0.. \text{npts} - 1$



Wow, we don't get oscillations at all! But notice the key parameter is the ratio of the time/nsteps. So to make sure we get "real" answers we should increase that ratio until we no longer see a change in the results.

On your own:

Write a summary statement describing the effect of step size on the observed oscillator behavior in the rabbit/lynxes system. Include in your statement the result from using very small step sizes.

Vary the initial populations of rabbits and lynxes. What happens to the oscillations? Change the rate constants. What happens? Explain. (Watch your time step-- you may need a smaller one to obtain valid results. If you don't receive any results, suspect that the time step is too large and that a numerical overflow error occurred, i.e. in calculating the solution Mathcad encountered larger numbers that it could handle.)

What terms would you have to add to the model to include the death of the rabbits and the consumption of the grass?

Adaptive time step method: RKadapt

Another differential equation solver used by Mathcad is Rkadapt. Rkadapt is suited for situations where the differential equations have solutions that vary quickly at some points and slowly at others. Rkadapt will still return answers every $t_{max}/npts$ but will internally use whatever type step is necessary to maintain the error below a user-established tolerance. This is a value you control under the Math menu (options-built-in variables).

We use the same equations as above and set the integration parameters here.

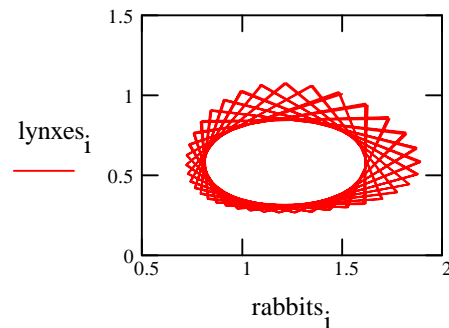
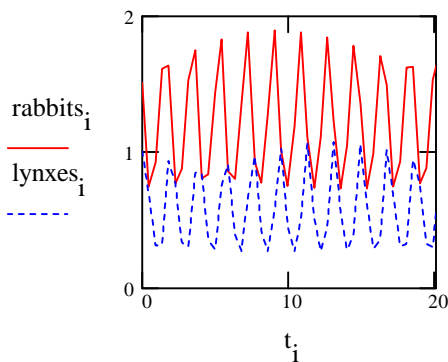
$$npts := 100 \qquad t_{max} := 45 \qquad \frac{npts}{t_{max}} = 2.222$$

$$i := 0..npts - 1$$

$$s := Rkadapt(y, 0, t_{max}, npts, D)$$

Notice how Rkadapt uses the same inputs as rkfixed. You don't need to learn any new steps to use it.

$$rabbits := s^{<1>} \qquad lynxes := s^{<2>} \qquad t := s^{<0>}$$



So the adaptive approach is more accurate in representing the true solution but it still introduces error because the time step is too large. Moral of the story: caveat integrator! **"Let the integrator beware!"** You should always reduce the time step until you obtain a result that is independent of the time step.

On your own: discover the optimal time step. How do you know when you have reached an optimal time step?

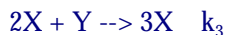
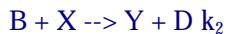
The Brusselator

Despite the fact that it generates sustained oscillatory behavior from simple "chemical reactions" with mass action kinetics, the Lotka-Volterra model is not an appropriate description of any actual chemical, as opposed to ecological, system. Its fatal flaw is that, in a sense, it is *too* successful in generating oscillatory behavior. It is possible to prove that the model has an oscillatory solution for any values of the rate constants, food-supply (a), and initial values of x and y in the equations, and that the amplitude and period of the oscillations obtained depend upon all of these quantities; in other words there is an infinite array of oscillatory solutions. If the system is perturbed, say by adding a bit more a , x , or y , it continues to oscillate, but now with a new period and amplitude until it is perturbed again. Furthermore, in the presence of any significant amount of noise, the behavior would hardly be recognizable as periodic, since it would constantly be jumping from one oscillatory behavior to another.

On your own: Change the initial conditions in this Lotka-Volterra model and see what happens? Describe your observations in your notebook.

Real chemical systems do not behave this way. They oscillate only within a finite range of parameters, and they have a single mode (amplitude and frequency) of oscillation, to which they return if the system is perturbed. Is it possible to construct a reasonable model that has these features? The first chemically respectable model was proposed by Prigogine and Lefever in 1968 and dubbed the Brusselator by Tyson (1973).

A and B are assumed to be species in large excess such that their concentrations do not change.



The rate equations are:

$$\frac{dx}{dt} = a k_1 - b k_2 x + k_3 x^2 y - k_4 x$$

We are interested in how x and y vary over time.

$$\frac{dy}{dt} = b k_2 x - k_3 x^2 y$$

Let's define the initial conditions for the Brusselator model:

$$\begin{aligned} X_0 &:= 1. & Y_0 &:= 1.5 & a &:= 1 & b &:= 3 \\ k_1 &:= 1 & k_2 &:= 1 & k_3 &:= 1 & k_4 &:= 1 \end{aligned}$$

Create the matrix D which contains the differential equations for our system. Note that y_0 is used for x and y_1 is used for y in the matrix for the differential equations written above.

$$D(t, y) := \begin{bmatrix} a \cdot k_1 - b \cdot y_0 \cdot k_2 + k_3 \cdot y_0 \cdot y_0 \cdot y_1 - k_4 \cdot y_0 \\ b \cdot k_2 \cdot y_0 - k_3 \cdot y_0 \cdot y_0 \cdot y_1 \end{bmatrix} \quad y := \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

We repeat the process we used above to solve the differential equations.

$$\text{npts} := 10000 \quad \text{tmax} := 100 \quad \frac{\text{npts}}{\text{tmax}} = 100$$

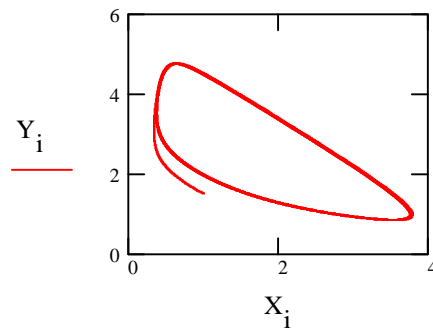
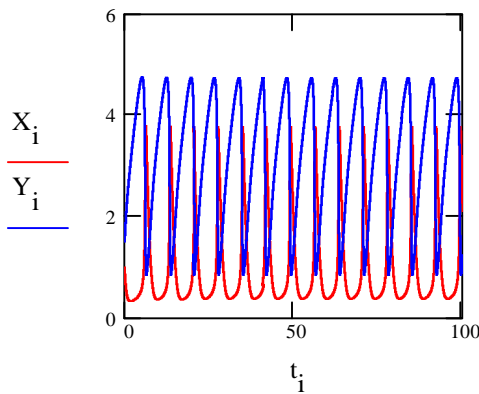
$$i := 0.. \text{npts} - 1$$

$$s := \text{Rkadapt}(y, 0, \text{tmax}, \text{npts}, D)$$

$$t := s^{<0>}$$

$$X := s^{<1>}$$

$$Y := s^{<2>}$$



On your own:

Change the initial conditions slightly. (For example, increase X or Y by 50%. Do you get the same attractor? How does this differ from you what you saw with the Lotka-Volterra Model?

The Oregonator Model of the Belousov-Zhabotinsky Oscillating Reaction

Nota Bene: This section is designed to go along with experiments described in Pojman (1994) If you are not doing the experiments, ignore references to comparison with lab data.

Field and Noyes (1972) published the first mechanism for an oscillating chemical reaction, the Belousov-Zhabotinsky reaction. For more information on the BZ reaction check out Degn (1972), Winfree (1984), Shakhshiri (1985), Noyes (1989), Geckle (1986), Field (1989), Pojman (1994, 1995), Benini (1996), Strizhak (1996).

They then developed a simple model called the Oregonator (they worked at the University of Oregon) in 1977 that captures most of the dynamical features. In order to make it easier to solve numerically and for other mathematical reasons, the variables are **scaled**, divided by combinations of rate constants to make all the variables have values about 1.

The variant we will use was developed by John Tyson, described in the excellent book edited by Field and Burger (1985).

First define the scaled variables.

$$x = [\text{HBrO}_2]/1.2 \times 10^{-7} \text{ M}$$

$$y = [\text{Br}^-]/6 \times 10^{-7} \text{ M}$$

$$z = [\text{Ce(IV)}]/10^{-3} \text{ M}$$

$$t = \text{time}/50 \text{ s}$$

$$f := 1 \quad \text{number of bromide ions released for each Ce(IV) that is reduced}$$

$$\varepsilon := 5 \cdot 10^2 \quad \text{proportional to both } [\text{H}^+] \text{ and the [bromate]}$$

$$\varepsilon_{\text{prime}} := 100 \quad \text{proportional to both } [\text{H}^+]^2 \text{ and the [bromate]}$$

$$q := 0.004 \quad \text{a combination of several rate constants}$$

Go to the original papers or a text for the chemical equations for the Belousov-Zhabotinsky reaction.

Here are the differential equations describing the chemical model:

$$dx/dt = [q \cdot y - xy + x(1 - x)]\epsilon$$

$$dy/dt = [-qy - xy + fz]\epsilon_{\text{prime}}$$

$$dz/dt = x - z$$

Note: the equations to the left are not active Mathcad equations. Refer to the references for the chemical reactions leading to these rate equations.

$$y_{\text{ initial}} := 24$$

$$x_{\text{ initial}} := 0.0008 \quad z_{\text{ initial}} := .2$$

Here we define the initial conditions.

Next we define the differential equations matrix and initial conditions vector.

$$y := \begin{bmatrix} y_{\text{ initial}} \\ x_{\text{ initial}} \\ z_{\text{ initial}} \end{bmatrix} \quad D(t, y) := \begin{bmatrix} \epsilon \cdot [q \cdot y_1 - y_0 \cdot y_1 + y_0 \cdot (1 - y_0)] \\ \epsilon_{\text{ prime}} \cdot (-q \cdot y_1 - y_0 \cdot y_1 + f \cdot y_2) \\ y_0 - y_2 \end{bmatrix}$$

We define the number of points, time range.

$$npts := 2000 \quad tmax := 50$$

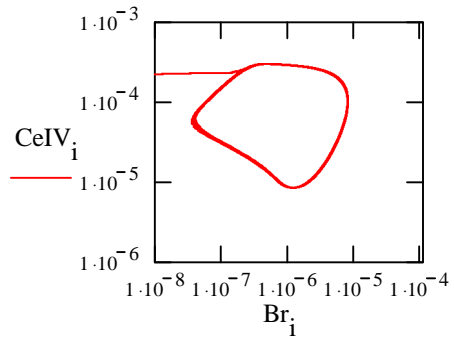
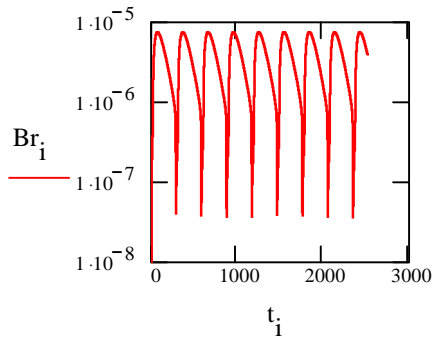
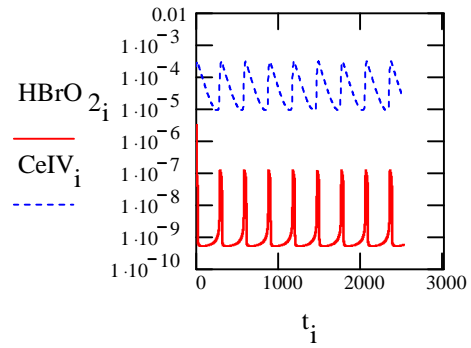
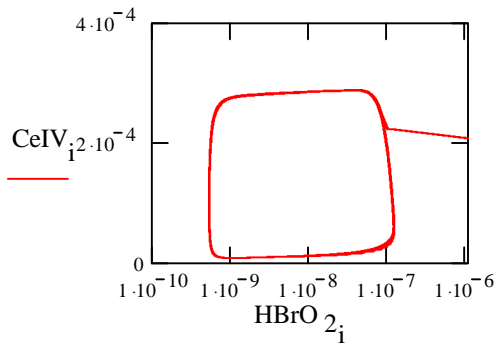
$$\frac{tmax}{npts} = 0.025$$

$$s := \text{Rkadapt}(y, 0, tmax, npts, D) \quad \text{Call library routine Rkadapt}$$

We convert back to "real" variables and plot them versus time.

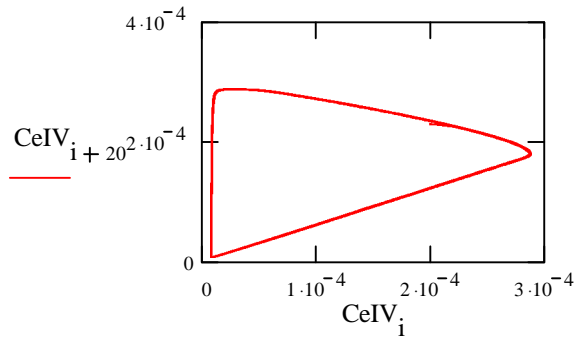
$$i := 0..npts \quad t := s^{<0>} \cdot 50$$

$$\text{HBrO}_2 := s^{<1>} \cdot 1.2 \cdot 10^{-7} \quad \text{Br} := s^{<2>} \cdot 6 \cdot 10^{-7} \quad \text{CeIV} := s^{<3>} \cdot .001$$



Now we do our time delay trick to get the attractor from only the Ce(IV) data.

$i := 0.. npts - 50$



On your own:

1. Does the attractor change with time? Compare this to what you saw in the BZ reaction. What does it mean about the consumption of reactants in the model?
2. Change the initial conditions. Do you get the same attractor? Explain.
3. Change ϵ and ϵ_{prime} . A linear change in both (they both must be changed) will correspond to a change in the bromate concentration. If you change ϵ linearly and ϵ_{prime} quadratically, you can represent changes in the acid concentration. For example, if you want to double the $[\text{H}^+]$, double ϵ and increase ϵ_{prime} four times. How does the period vary with changes in the bromate concentration? The amplitude? Compare this to your experimental results.

Acknowledgment: The author acknowledges the National Science Foundation for support of the 1997 NSF-UFE Workshop on "Numerical Methods in the Undergraduate Chemistry Curriculum Using the Mathcad Software" and the organizers (Jeff Madura, Andrzej Wierzbicki and Sidney Young, University of South Alabama) and Randy Washington for assistance in beta testing in the physical chemistry lab at USM.

References

Benini, O.; Cervellati, R.; Fetto, P. "The BZ Reaction: Experimental and Model Studies in the Physical Chemistry Laboratory," *J. Chem. Ed.* **1996**, *73*, 865-868.

Celia, M. A.; Gray, W. G. *Numerical Methods for Differential Equations. Fundamental Concepts for Scientific and Engineering Applications*; Ed.; Prentice Hall: Englewood Cliffs, 1992.

D'Ancona, U. *The Struggle for Existence*; E. J. Brill: Leiden, 1954.

Degn, H. "Oscillating Chemical Reactions in Homogeneous Phase," *J. Chem. Ed.* **1972**, *49*, 302-307.

Ferreira, M. M. C.; Ferreira Jr., W. C.; Lino, A. C. S.; Porto, M. E. G. "Uncovering Oscillations, Complexity, and Chaos in Chemical Kinetics Using *Mathematica*," *J. Chem. Ed.* **1999**, *76*, 861-866.

Field, R. J.; Burger, M. *Oscillations and Traveling Waves in Chemical Systems*; Wiley: New York, 1985.

Field, R. J.; Körös, E.; Noyes, R. "Oscillations in Chemical Systems. II. Thorough Analysis of Temporal Oscillation in the Bromate-Cerium-Malonic Acid System," *J. Am. Chem. Soc.* **1972**, *94*, 8649-8664.

Field, R. J.; Noyes, R. M. "Oscillations in Chemical Systems.18. Mechanisms of Chemical Oscillators: Conceptual Bases," *Acc. Chem. Res.* **1977**, *10*, 214-221.

Field, R. J.; Noyes, R. M. "Oscillations in Chemical Systems.19. Mechanism of Chemical Oscillations: Experimental Examples," *Acc. Chem. Res.* **1977**, *10*, 273-280.

Field, R. J.; Schneider, F. W. "Oscillating Chemical Reactions and Nonlinear Dynamics," *J. Chem. Ed.* **1989**, *66*, 195-204.

Gear, C. W. *Numerical Initial Value Problems in Ordinary Differential Equations*; Ed.; Prentice-Hall: Englewood Cliffs, NJ, 1971.

Geckle, D. S.; Salmon, J. F. "The Application of Simultaneous Electrochemical and Spectral Data in Analyzing Complex Oscillating Reactions," *J. Chem. Ed.* **1986**, *63*, 908-909.

- Lotka, A. J. "Undamped Oscillations Derived From the Law of Mass Action," *J. Am. Chem. Soc.* **1920**, *42*, 1595-1599.
- Noyes, R. M. "Some Models of Chemical Oscillators," *J. Chem. Ed.* **1989**, *66*, 190-191.
- Pojman, J. A.; Craven, R.; Leard, D. "Oscillating Reactions and Traveling Waves in the Physical Chemistry Lab," *J. Chem. Ed.* **1994**, *71*, 84-90.
- Pojman, J. A.; West, W. W. "A Unified Physical Chemistry Lab Based on Oscillating Reactions and Traveling Fronts," *J. Chem. Ed.* **1996**, *73*, 35.
- Prigogine, I.; Lefever, R. "Symmetry Breaking Instabilities in Dissipative Systems. II," *J. Chem. Phys.* **1968**, *48*, 1695-1700.
- Shakhashiri, B. Z. *Chemical Demonstrations: A Handbook for Teachers. Volume 2*; University of Wisconsin Press: Madison, 1985.
- Strizhak, P.; Menzinger, M. "Nonlinear Dynamics of the BZ Reaction: A Simple Experiment That Illustrates Limit Cycles, Chaos, Bifurcations and Noise," *J. Chem. Ed.* **1996**, *73*, 868-873.
- Tyson, J. "Some Further Studies of Nonlinear Oscillations in Chemical Systems," *J. Chem. Phys.* **1973**, *58*, 3919-3930
- Volterra, V. "Fluctuations in the Abundance of a Species Considered Mathematically," *Nature* **1926**, *118*, 558-560.
- Winfree, A. T. "The Prehistory of the Belousov-Zhabotinsky Oscillator," *J. Chem. Ed.* **1984**, *61*, 661-663.