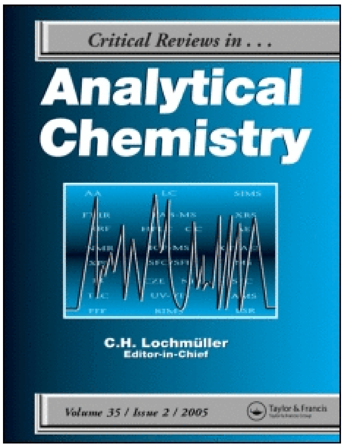

Critical Reviews in Analytical Chemistry
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713400837

Spreadsheet Simulation of Chemical Kinetics

Robert de Levie

To cite this Article de Levie, Robert(2002) 'Spreadsheet Simulation of Chemical Kinetics', Critical Reviews in Analytical Chemistry, 32: 1, 97 — 107

To link to this Article: DOI: 10.1080/10408340290765461

URL: http://dx.doi.org/10.1080/10408340290765461

Critical Reviews in Analytical Chemistry, 32(1):97–107 (2002)

Spreadsheet Simulation of Chemical Kinetics

Robert de Levie

Department of Chemistry, Bowdoin College, Brunswick, ME 04011

ABSTRACT: The use of a spreadsheet for the numerical simulation of differential equations is illustrated. The method is transparent and versatile. Strategies, including user-defined spreadsheet functions, are described to optimize either ease of use, or accuracy of the simulation.

I. INTRODUCTION

The mathematical description of the rates of chemical reactions involves differential equations. In simple cases those differential equations can be solved in closed form, but in most cases the equations can be written down but have no known solutions. In that case one has several options: either one simplifies the model until it allows a closed-form solution (e.g., by deleting some supposedly minor reaction partners, or by making assumptions such as a steady-state approximation), or one keeps the model intact but uses a numerical solution. Both approaches have their disadvantages. Simplifying the model may lead to the inadvertent omission of a crucial ingredient, and making simplifying assumptions may not be justified. On the other hand, it may be difficult or impossible to extract general information from a numerical solution. In general, before choosing an approach, one should be aware of the tradeoffs involved.

There are numerous specialized programs that will perform numerical simulations of chemical kinetics. There are also several general-purpose programs, such as Mathematica, Maple, and Mathcad, that can do so, easily and competently. In any environment, the choice of software often depends both on the availability of programs, and on the familiarity of the user with these. Here we will use spreadsheets, because they are widely available, and readily programmed, while their operation remains fully transparent to the user. Moreover, modern spreadsheets such as Excel are eminently suitable for numerical simulation.[1,2]

II. FIRST-ORDER KINETICS

As our first example we consider the unidirectional first-order reaction A → products, with a rate described by the differential equation

$$da/dt = -\,k\,a \tag{1}$$

where a denotes the concentration of the chemical species A. The mathematical solution of equation (1) consistent with the initial condition $a_{t=0} = a_0$ is

$$a = a_0 \exp[-kt] \tag{2}$$

where $t = 0$ specifies the time at the beginning of the experiment. Below we will first pretend not to know that Eq. 1 has the mathematical solution Eq. 2, before using this knowledge to check the simulation.

In order to simulate the resulting kinetics (without solving the differential equation) we recall that the *differential* quotient da/dt is the limit of a *difference* quotient,

$$\frac{da}{dt} = \lim_{\Delta t \to 0} \frac{\Delta a}{\Delta t} \tag{3}$$

where $\Delta a = a_{t+\Delta t} - a_t$. We can therefore replace the term da/dt in Eq. 1 by the corresponding quotient $\Delta a/\Delta t$,

$$da/dt \approx \Delta a/\Delta t \tag{4}$$

as long as the increment Δt is sufficiently small.

We now rewrite Eq. 1 as

1040-8347/02/$.50

$$\Delta a/\Delta t = -\,k\,a \qquad (5)$$

so that

$$\Delta a = -\,k\,a\,\Delta t \qquad (6)$$

$$a_1 = a_0 + \Delta a = a_0 - k\,a_0\,\Delta t = a_0\,(1 - k\,\Delta t) \quad \text{at } t = \Delta t \qquad (7)$$

$$a_2 = a_1 - k\,a_1\,\Delta t = a_1\,(1 - k\,\Delta t) = a_0\,(1 - k\,\Delta t)^n \quad \text{at } t = 2\,\Delta t \qquad (8)$$

$$a_n = a_{n-1} - k\,a_{n-1}\,\Delta t = a_{n-1}\,(1 - k\,\Delta t) = a_0\,(1 - k\,\Delta t)^n \quad \text{at } t = n\,\Delta t \qquad (9)$$

Thus, we start with the initial concentration a_0 at $t = 0$, from which we calculate $a_1 = a_0 + \Delta a = a_0 - k\,a_0\,\Delta t = a_0\,(1 - k\,\Delta t)$ at $t = \Delta t$. Next we compute $a_2 = a_1 + \Delta a = a_1\,(1 - k\,\Delta t)$ at $t = 2\,\Delta t$, and so on. Given enough intervals Δt, this process will simulate the entire time course of the concentration a, one interval Δt at a time, simply by multiplying the preceding concentration each time by $(1 - k\,\Delta t)$. Use of Eq. 4 with a sufficiently small interval Δt is the so-called explicit or Euler method. Because the problem posed here contains only one time constant, k, with the dimension of a reciprocal time, the condition of 'sufficient' smallness must be that $\Delta t \ll 1/k$ or $k\,\Delta t \ll 1$.

We now compare Eq. 9 with the exact result of Eq. 2, which we rewrite as

$$a = a_0\,\exp[-kt] = a_0\,\exp[-k\,n\,\Delta t] = a_0\,\{\exp[-k\,\Delta t]\}^n \approx a_0\,(1 - k\,\Delta t)^n \qquad (10)$$

which indeed corresponds to Eq. 9 as long as $k\,\Delta t \ll 1$.

Spreadsheet implementation of the above digital simulation is straightforward, because Eq. 9 relates the value of the concentration a_n to the previous concentration a_{n-1}. On a spreadsheet, we make two parallel columns, one for time t, the other for the concentration $a(t)$. In the top row of these columns we deposit the value 0 for t_0, and a numerical value of a_0 for $a(0)$. The next row should then contain instructions to compute $t_1 = t_0 + \Delta t$ and $a(1) = a_0\,(1 + k\Delta t)$, the following row $t_2 = t_1 + \Delta t$ and $a(2) = a_1\,(1 + \mathrm{k}\Delta t)$, and so on. In practice, we only need to place instructions in the cells of the first and second row, then copy the instructions in the second row down as far as we wish. The calculation is virtually instantaneous, and the spreadsheet makes it easy to plot a as a function of t, as illustrated in Figure 1.

Because we know the corresponding, exact solution Eq. 2, it is useful to compute and plot a_{exact} as well as the difference $a_{exact} - a_{simul}$. This is also illustrated in Figure 1. For $k\,\Delta t \ll 1$ the curves for a_{exact} and a_{simul} visually overlap, but a graph of their difference $a_{exact} - a_{simul}$ will show the error inherent in this simulation. In Figure 1 we have used an intentionally rather large value for Δt in order to accentuate the difference between a_{simul} and a_{exact}.

We can perform the simulation with different values of k and Δt. However, because k and Δt only occur in Eq. (9) as the product $k\Delta t$, we merely need to vary one or the other. Here we will

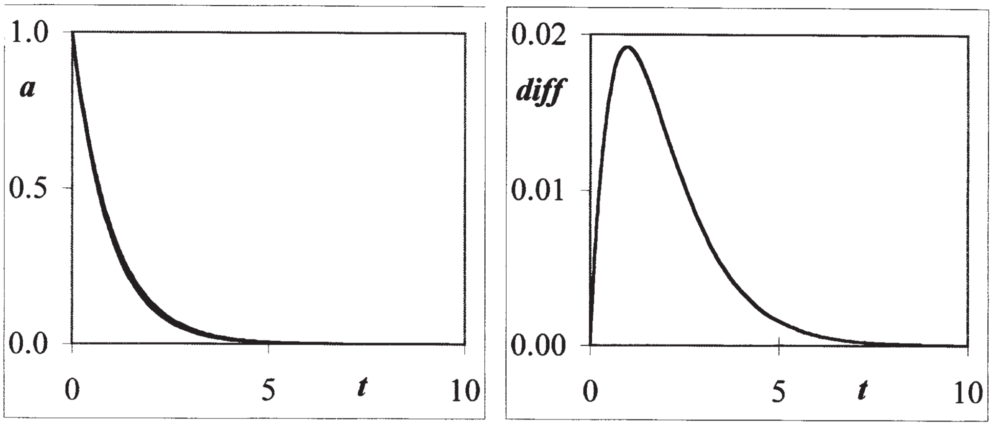

FIGURE 1. Left panel: The exact solution to the unidirectional first-order rate expression (1) and its numerical solution (2) for $a_0 = 1$, $k = 1$, and $\Delta t = 0.1$. Right panel: the corresponding difference $a_{exact} - a_{simul}$.

fix k, and vary Δt. In doing so we will find that the maximum simulation error $a_{exact} - a_{simul}$ is approximately 0.2 $a_0 \Delta t$ as long as $k \Delta t \ll 1$. In principle, then, we can make the simulation as accurate as we wish merely by making the interval Δt sufficiently small.

III. IMPROVING THE ACCURACY OF THE SIMULATION

Because the total time required for the actual reaction to run its course depends on k, and is independent of the computational intervals Δt used in the simulation, using shorter simulation intervals Δt implies that we must increase the total number of steps (and hence the number of rows in the spreadsheet) in order to obtain higher computational accuracy.

In the above example we used only 101 rows, with t-values ranging from 0 to 10 in increments of 0.1. On a modern spreadsheet we can easily extend the column length to 1001 or 10001 rows. Still, there is a practical limit on the length of a column we can use, not only because spreadsheets have a finite number of rows, but more importantly because calculations on long columns slow down the computer, and the data in long columns sometimes cannot even be plotted.

Fortunately, in the present case, there is no need for a large number of cells, because we can compress the effect of many steps into a single cell. Instead of computing every term, we can obtain a higher resolution in the calculation for a given column length, by computing only the n^{th} terms. For example, by replacing the instruction for $a_1 = a_0 (1 - k \Delta t)$ by, say, $a_{10} = a_0 (1 - k \Delta t/10)^{10}$, that for $a_2 = a_0 (1 - k \Delta t)^2$ by $a_{20} = a_0 (1 - k \Delta t/10)^{20}$, etc., we achieve the same computational result as we would have obtained by using a ten times longer column. In that case we compute a at each step $\Delta t = 0.1$ as if we were using ten steps of size 0.01.

Similarly, replacing a_1 by $a_{1000} = a_0 (1 - k\Delta t/1000)^{1000}$, a_2 by $a_{2000} = a_0 (1 - k\Delta t/1000)^{2000} = a_{1000} (1 - k\Delta t/1000)^{1000}$, and so on, will simulate the effect of a step size Δt / 1000 but without increasing the actual column length, or the number of points plotted. The top panels in Figure 2 illustrate the resulting differences $a_{exact} - a_{simul}$. Note that this procedure does not perceptibly slow the calculation, because the computer uses logarithms to calculate the results.

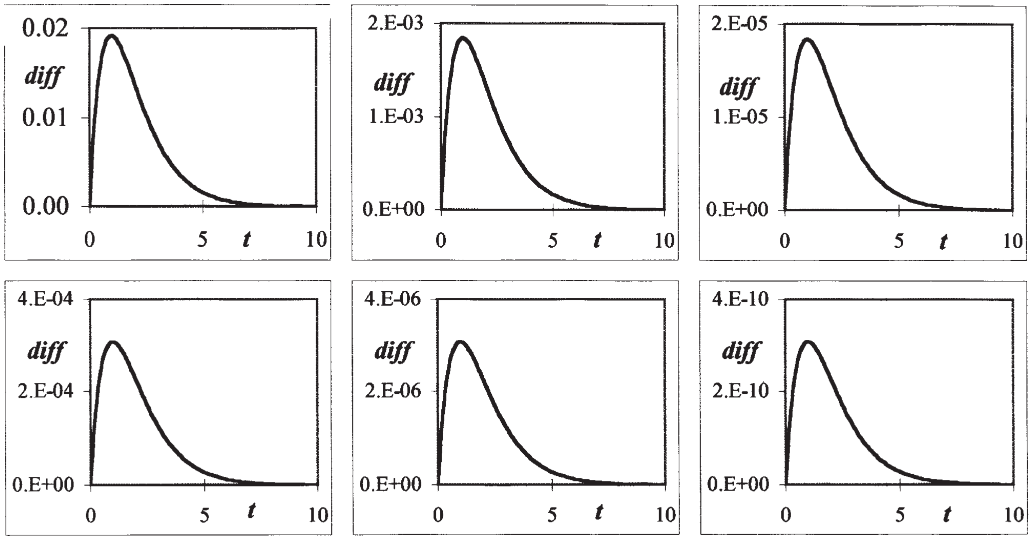

FIGURE 2. The differences $a_{exact} - a_{simul}$ for the approach to numerical solution of the unidirectional first-order rate expression for $a_0 = 1$, $k = 1$, and $\Delta t = 0.1$. Top row: for the explicit method; bottom row: for the 1st order semiimplicit method. Left panels: direct simulation; middle and right panels: leapfrogging the computation by calculating results for n successive increments of size $\Delta t/n$, where $n = 10$ for the middle panels, $n = 1000$ for the right panels. In this figure the primary message is in the vertical scales used rather than in the curves themselves.

IV. THE SEMI-IMPLICIT METHOD

While the explicit method we have illustrated so far is very easy to implement, it tends to introduce a *systematic bias* since it uses, at each step Δt, the concentration(s) computed for the end of the previous time increment. Although the resulting errors can in principle be made vanishingly small, by sufficiently reducing the time interval Δt used, it is more effective to use a more efficient algorithm.

When the concentration of a reagent is a continuously decreasing function of time, and the steps Δt are not sufficiently small, the systematic bias will lead to a small but consistent overestimate of that concentration. Over many steps, even a rather small overestimate may accumulate to generate a substantial composite error. Similarly, when the concentration of a product increases, this concentration increase in every interval Δt will be computed on the basis of its previous concentration, which again will lead to a systematic error.

An improved simulation might therefore be obtained by using an estimate of the *average* (rather than the *previous*) concentration during the period Δt. This is done in the semi-implicit method, which considers not only the previous value of the concentration, but also its next, yet to be determined value. The name *implicit* indicates that the new value is not simply calculated from the previous one, as in the above examples, but through an equation in which the new value occurs on both sides of the equal sign. That equation must then be solved in order to obtain an *explicit* expression for the new value.

In fact, there are many different semi-implicit and implicit methods, of various levels of sophistication and complexity. Here we only illustrate the simplest of these, by assuming that all concentrations change linearly over a sufficiently small interval Δt. Higher-order methods instead assume higher-order (e.g., quadratic, exponential, etc.) dependencies of concentration on time.

Let us return to the unidirectional first-order reaction, and assume that a varies in a linear fashion with t over the small interval Δt. Let the concentration a at $t = 0$ have the value a_0, and let its value at $t = \Delta t$ be $a_1 = a_0 + \Delta a$. We then use the average value of a, viz. $(a_0 + a_1) / 2 = (2a_0 + \Delta a) / 2 = a_0 + {}^1/_2\, \Delta a$, as an improved estimate of a during the interval Δt, and therefore substitute it into the right-hand side of the difference equation (5), which now will read

$$\Delta a/\Delta t = k\,(a + {}^1/_2\,\Delta a) \tag{5'}$$

Consequently, Eqs. 6 through 9 must now be replaced by

$$\Delta a = -\,a\,k\,\Delta t \,/\, (1 + {}^1/_2\,k\,\Delta t) \tag{6'}$$

$$a_1 = a_0 + \Delta a = a_0\,(1 - {}^1/_2\,k\,\Delta t) \,/\, (1 + {}^1/_2\,k\,\Delta t) \quad \text{at } t = \Delta t \tag{7'}$$

$$a_2 = a_1 + \Delta a = a_1\,(1 - {}^1/_2\,k\,\Delta t) \,/\, (1 + {}^1/_2\,k\,\Delta t) = a_0\,\{(1 - {}^1/_2\,k\,\Delta t) \,/\, (1 + {}^1/_2\,k\,\Delta t)\}^2 \quad \text{at } t = 2\,\Delta t \tag{8'}$$

$$a_n = a_0\,\{(1 - {}^1/_2\,k\,\Delta t) \,/\, (1 + {}^1/_2\,k\,\Delta t)\}^n \quad \text{at } t = n\,\Delta t \tag{9'}$$

Equation 9′ leads to a much more accurate simulation, as can be seen in the bottom panels of Figure 2. Furthermore, by replacing the computation of the effect of one time interval Δt per cell by that of n intervals of size $\Delta t/n$, the simulation error now decreases with n^2, rather than with n as in the explicit method. Higher-order Runge-Kutta methods converge even faster: with n^{p+1} for a p^{th} order Runge-Kutta method. In semi-implicit or implicit methods for chemical kinetics, it may be even more efficient to use exponential rather than higher-order polynomial interpolation.[3]

It may be illuminating to compare Eqs. 2 and 9′. Expansion of the exponent in Eq. 2 for $k\,\Delta t \ll 1$ yields

$$\begin{aligned} a_n &= a_0 \exp[-\,k\,t] = a_0 \exp[-\,k\,n\,\Delta t] \\ &= a_0\,\{\exp[-\,k\,\Delta t]\}^n \\ &\approx a_0\,\{1 - k\,\Delta t + (k\,\Delta t)^2/2 - (k\,\Delta t)^3/6 + \dots\}^n \end{aligned}$$

whereas series expansion of the term $1 / (1 + {}^1/_2\,k\,\Delta t)$ in Eq. 9′ leads to

$$\begin{aligned} a_n &= a_0\,\{(1 - {}^1/_2\,k\,\Delta t) \,/\, (1 + {}^1/_2\,k\,\Delta t)\}^n \\ &\approx a_0\,\{1 - k\,\Delta t + (k\,\Delta t)^2/2 - (k\,\Delta t)^3/4 \\ &\quad + \dots\}^n \end{aligned}$$

so that the simulation is now accurate to within the second-order term.

In summary, then, if we want an easy, approximate idea of the course of the reaction, the explicit method is our best choice: it is simple and straightforward to implement, and usually (as long as the intervals Δt are not too large) gives the over-all shape of the time course of the concentration(s) of the reacting species with a minimum of effort. On the other hand, if we need higher accuracy, the semi-implicit approach is the method of choice. If that is not enough, we can use the multi-step method to obtain even higher simulation accuracy, as seen in the middle and right panels of the bottom row of Figure 2. Few experimental kinetic data will be of such quality to justify this combined approach.

V. MONOMER-DIMER KINETICS

As our second example we consider a somewhat more complicated case, in which two monomers A react to form a dimer B, while the reverse reaction can also take place:

$$2\ \mathrm{A} \underset{k'}{\overset{k}{\rightleftarrows}} \mathrm{B} \tag{10}$$

with the associated differential equations

$$da/dt = -\,k\,a^2 + 2\,k'\,b \tag{11}$$

$$db/dt = +\,{}^1\!/_2\,k\,a^2 - k'\,b \tag{12}$$

Here a and b represent the concentrations of the monomer A and the dimer B, respectively. The coefficients 2 and $^1/_2$ have been introduced to satisfy the mass balance requirement $a + 2b = a_0 + 2b_0$, so that $d(a{+}2b)/dt = 0$. This mass balance can be used to simplify the mathematics, since it allows us to eliminate b from Eq. 11 with $2b = a_0 + 2b_0 - a$, in which case we only need to solve

$$\begin{aligned} da/dt &= -\,k\,a^2 - k'\,a + k'\,(a_0 + 2\,b_0) \\ &= -\,k\,a^2 - k'\,a + k'' \end{aligned} \tag{13}$$

where $k'' = k'\,(a_0 + 2\,b_0)$. Conversion of Eq. 13 into the corresponding explicit difference equation leads to

$$\Delta a/\Delta t = -\,k\,a^2 - k'\,a + k'' \tag{14}$$

so that we obtain

$$a_1 = a_0 + \Delta a = a_0 + (-\,k\,a_0{}^2 - k'\,a_0 + k'')\,\Delta t \tag{15}$$

$$a_2 = a_1 + \Delta a = a_1 + (-\,k\,a_1{}^2 - k'\,a_1 + k'')\,\Delta t \tag{16}$$

$$a_n = a_{n-1} + \Delta a = a_{n-1} + (-\,k\,a_{n-1}{}^2 - k'a_{n-1} + k'')\,\Delta t \tag{17}$$

As in our first example, a closed-form mathematical solution to Eq. 13 exists. It reads

$$a(t) = \frac{-(k'+q)\,(2ka_0 + k' - q) + (k' - q)\,(2ka_0 + k' + q)\,e^{\,qt}}{2\,k\,[(2ka_0 + k' - q) - (2ka_0 + k' + q)\,e^{\,qt}]} \tag{18}$$

$$q = \sqrt{(k')^2 + 4\,k\,k''} \tag{19}$$

which we can use to calibrate the accuracy of the simulation. Figure 3 illustrates the results for a_{exact} and a_{simul}, and also shows the consequent differences $a_{exact} - a_{simul}$. Again, for the purpose of the illustration, we have used a rather large value of Δt.

VI. USING A SPREADSHEET FUNCTION

In this case we do not have an explicit expression for a_n in terms of a_0, as in Eq. 9. Instead, Eq. 17 is a recursion formula. Therefore, we cannot use the trick of condensing a number of computational steps into a single cell instruction, in order to improve the numerical accuracy of the simulation within a given column length. If higher accuracy is needed, we have the following options: (1) use smaller time increments Δt and correspondingly extend the column length; (2)

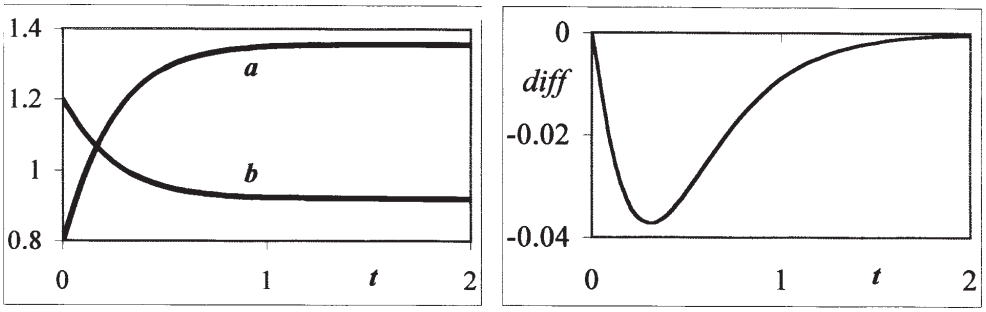

FIGURE 3. Left panel: The exact solution (1) for the approach to equilibrium of a monomer-dimer system, and its numerical simulation (2), for $a_0 = 0.8$, $b_0 = 1.2$, $k = 1$, $k' = 1$, and $\Delta t = 0.1$. Right panel: the corresponding difference $a_{exact} - a_{simul}$. Because $b = b_0 + {}^1/_2\,(a_0 - a)$, the corresponding difference $b_{exact} - b_{simul}$ is half the size and of opposite sign.

perform the computation off-screen; or (3) look for a more efficient algorithm. Here we will consider the second option; the third option will be discussed in Section VII.

While we cannot leapfrog the computation by using a single formula for n time increments Δt, Excel allows us to perform n successive steps *off-screen* with a user-defined function. Such a function is stored in a module, just like a macro. The function STEPA (or whatever name you may want to give it) will do our bidding. It uses as input the values of k, k', $k'' = k'(a_0 + 2\,b_0)$, Δt, and n, as well as the immediately preceding value of a, and places the new value of a in the cell. Assuming that k, k', k'', Δt, and n have been given the names k, kk, kkk, dt, and n respectively, STEPA can be defined (in the Visual BASIC used in Excel) as

```
Function STEPA(k, kk, kkk, dt, n, a)
For i = 1 To n
    a = a + (–k*a*a –kk*a +kkk)*dt/n
Next i
STEPA = a
End Function
```

Since the For … Next loop runs n times, this function performs the computation n times before writing its result to the cell. If you wonder what happened in this function with the labels n and n–1 on a in Eq. 17, remember that the equal sign in BASIC denotes an assignment, in which the computer first evaluates the value of the right-hand side of the expression (using the old value for a, i.e., a_{n-1}), then assigns that value to the variable a on the left-hand side of the equal sign (which thereby becomes a_n).

Where do you store this function? In Excel, functions are written and kept in the same place as macros are, that is, in so-called modules. In Excel 5 and Excel 95, click on Insert, then on Macro > Module, and type the function in the blank module sheet. Afterwards, click on the worksheet or module tab to get from one to the other. Starting with Excel 97, the modules are more hidden from view, and it is easiest to use Alt+F11 to generate the module, and subsequently to go back and forth between spreadsheet and module.

Once you have defined a function by typing it in the module, you can use it like any other function, such as SIN(x) or LOG(y). For instance, if B is the column for a, and it starts on row 14 with the initial values a_0, then enter in cell B15 the instruction =STEPA(k, kk, kkk, dt, n, B14). Copy this instruction down the column as far as you need.

This method does compress n steps per row, but it must be used with caution, because the computer now must still make n computations for each cell. Thus, while it will keep the column length manageable, it will not reduce the number of computations. The latter is, in fact, considerably larger, because of all the bookkeeping involved in the For … Next loops. For example, if you have a column of some 100 rows, and you set $n = 1000$, the computer must perform at least 100,000 calculations (100 cells times 1000 calculations per cell) every time you change any one of the parameters k, k', Δt, n, a_0, or b_0. On a spreadsheet, 100,000 recalculations may perceptibly slow down even a fast processor.

VII. THE SEMI-IMPLICIT METHOD

Application of the semi-implicit method to the monomer/dimer reaction scheme (10) and its associated differential equation, Eq. 13, yields the difference equation

$$\begin{aligned}\Delta a/\Delta t &= -k\,(a+{}^1/_2\,\Delta a)^2 - k'\,(a+{}^1/_2\,\Delta a) + k'' \\ &\approx -k\,a\,(a+\Delta a) - k'\,(a+{}^1/_2\,\Delta a) + k'' \\ &= -k\,a^2 - k'a + k'' - (ka + {}^1/_2\,k')\,\Delta a\end{aligned} \tag{14'}$$

where we have linearized $(a+{}^1/_2\,\Delta a)^2 = a\,\{a+\Delta a+{}^1/_4\,(\Delta a)^2\} \approx a\,(a+\Delta a)$ by neglecting the higher-order term in $(\Delta a)^2$. Consequently

$$a_1 = a_0 + \Delta a = a_0 + (-k\,a_0^{\,2} - k'a_0 + k'')\,/\,(1/\Delta t + ka_0 + {}^1/_2\,k') \tag{15'}$$

$$a_2 = a_1 + \Delta a = a_1 + (-k\,a_1^{\,2} - k'a_1 + k'')\,/\,(1/\Delta t + ka_1 + {}^1/_2\,k') \tag{16'}$$

$$a_n = a_{n-1} + \Delta a = a_{n-1} + (-k\,a_{n-1}^{\,2} - k'a_{n-1} + k'')\,/\,(1/\Delta t + ka_{n-1} + {}^1/_2\,k') \tag{17'}$$

which results in a simulation error that is more than fifteen times smaller than that shown in Figure 3b for otherwise identical conditions. Moreover, a reduction in Δt by a factor f results in a reduction in the inaccuracy by a factor of f^2, compared with a factor of f for the explicit method. For more accurate simulations, we should therefore use 17$'$ instead of 17.

If that is not sufficient, a function such as STEPA, modified to represent 17$'$ rather than (17), can be enlisted as well. Such a function is, of course, more efficient when used in conjunction with the semi-implicit method, which will further reduce the inaccuracy of the simulation by a factor n^2, rather than by a factor of n when used with the explicit method. Higher-order semi-implicit or fully implicit methods exhibit an even higher power dependence on n.

VII. THE LOTKA OSCILLATOR

As an application of the above, we will use as our last example the Lotka model, which is based on two coupled autocatalytic reactions, and can give rise to *oscillatory* behavior. In a catalytic reaction, the catalyst speeds up (or retards) a reaction without being consumed itself. Formally, such a reaction can be depicted as $\mathrm{A}\xrightarrow[\mathrm{C}]{}\mathrm{B}$.

Of course, in order to affect the reaction rate, C cannot be a mere spectator, but must be involved in the reaction, for example, as in $\mathrm{A + C \rightarrow B + C}$. The efficiency of many important chemical processes, such as the production of gasoline and of nitrogen-based fertilizer, depend on catalytic processes. Similarly, many biochemical processes depend on nature’s catalysts, which are called enzymes.

In an autocatalytic reaction, the reaction product serves as a catalyst. The simplest example of such a process is the reaction $\mathrm{A + B \rightarrow 2B}$, which can be written alternatively as $\mathrm{A}\xrightarrow[\mathrm{B}]{}\mathrm{B}$.

There are many known examples of autocatalytic reactions, for example, the Landolt clock reaction[4], or the $\mathrm{MnO_2}$-catalyzed reduction of permanganate.

The Lotka oscillator[5,6] is based on the reaction scheme

$$\mathrm{A+B}\xrightarrow[k_1]{}\mathrm{2B} \tag{20}$$

$$\mathrm{B+C}\xrightarrow[k_2]{}\mathrm{2C} \tag{21}$$

$$\mathrm{C}\xrightarrow[k_3]{}\text{products} \tag{22}$$

where we will assume that the concentration a of A is kept constant in order to obtain stationary oscillations, so that $da/dt = 0$. The corresponding rate expressions for b and c are

$$\frac{db}{dt} = k_1\,a\,b - k_2\,b\,c \tag{23}$$

$$\frac{dc}{dt} = k_2\,b\,c - k_3\,c \tag{24}$$

For the explicit method we have

$$\frac{\Delta b}{\Delta t} = k_1\,a\,b - k_2\,b\,c \tag{25}$$

$$\frac{\Delta c}{\Delta t} = k_2\, b\, c - k_3\; c \tag{26}$$

$$b_1 = b_0 + \Delta b = b_0 + (k_1\, a\, b_0 - k_2\, b_0\, c_0)\,\Delta t \tag{27}$$

$$c_1 = c_0 + \Delta c = c_0 + (k_2\, b_0\, c_0 - k_3\, c_0)\,\Delta t \tag{28}$$

$$b_2 = b_1 + \Delta b = b_1 + (k_1\, a\, b_1 - k_2\, b_1\, c_1)\,\Delta t \tag{29}$$

$$c_2 = c_1 + \Delta c = c_1 + (k_2\, b_1\, c_1 - k_3\, c_1)\,\Delta t \tag{30}$$

$$b_n = b_{n-1} + \Delta b = b_{n-1} + (k_1\, a\, b_{n-1} - k_2\, b_{n-1}\, c_{n-1})\,\Delta t \tag{31}$$

$$c_n = c_{n-1} + \Delta c = c_{n-1} + (k_2\, b_{n-1}\, c_{n-1} - k_3\, c_{n-1})\,\Delta t \tag{32}$$

which really does not take much effort to write down.

Such a simulation is shown in Figure 4 for a column of 1001 rows, with $\Delta t = 0.01$. Unfortunately, even though the value of Δt is considerably smaller than $1/k$ for even the largest k-value used, the result is unsatisfactory, because it does not lead to steady-state oscillations (as you should expect when the concentration a is kept constant). Figure 4 illustrates how, in such a cyclic process, a small but systematic error can accumulate in successive cycles, thereby quickly leading to quite significant deviations.

We can reduce Δt to verify that it was indeed too large, and thereby caused the run-away behavior shown in Figure 4. Because the column is already fairly long, we use functions instead. Note that the concentrations b and c are mutually dependent, so that both must be computed inside the For … Next loop. And because we deal here with an open system (since we must continually supply A to keep its concentration a constant) there is no convenient mass balance equation to eliminate either b or c. (Note that names assigned to spreadsheet constants in Excel apply to the entire workbook, and that different names must be used to refer to different spreadsheet cells.)

```
Function STEPB(a, b, c, k1, k2, k3, dt, n)
For i = 1 To n
    b = b + (k1*a*b – k2*b*c)*dt/n
    c = c + (k2*b*c – k3*c)*dt/n
Next i
STEPB = b
End Function

Function STEPC(a, b, c, k1, k2, k3, dt, n)
For i = 1 To n
    b = b + (k1*a*b – k2*b*c)*dt/n
    c = c + (k2*b*c – k3*c)*dt/n
Next i
STEPC = c
End Function
```

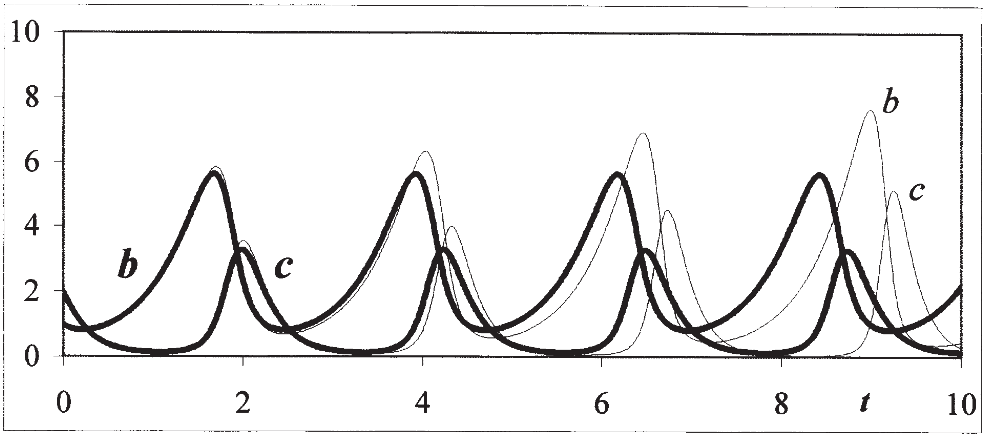

FIGURE 4. Numerical simulations for the concentrations b and c of the Lotka oscillator with $a = 1$, $b_0 = 1$, $c_0 = 2$, $k_1 = 2$, $k_2 = 2$, $k_3 = 5$, $\Delta t = 0.01$. Thin curves: semi-implicit explicit method. Thick curves: visually identical results for the method, and for the explicit method supplemented with functions using 10 computations per cell (making the effective Δt equal to 0.001).

For n = 10 this results in a stationary oscillation, as illustrated in Figure 4. How will you know whether this *n*-value is sufficient? Copy the numerical values (with Copy, Edit, Paste Special, Values) to another spot on the spreadsheet, then run the simulation with, say, $n = 20$, and compare the results. Keep increasing *n* by, say, factors of 2 or $2^{1}/_{2}$ until you are satisfied with the consistency of the result.

Alternatively we can use an implicit simulation, in which case we use

$$\frac{\Delta b}{\Delta t} = k_1\, a\,(b + \Delta b/2) - k_2\,(b + \Delta b/2)\,(c + \Delta c/2) \tag{25′}$$

$$\frac{\Delta c}{\Delta t} = k_2\,(b + \Delta b/2)\,(c + \Delta c/2) - k_3\,(c + \Delta c/2) \tag{26′}$$

$$\left(\frac{1}{\Delta t} - k_1\, a/2 + k_2 c/2\right)\Delta b + \left(k_2\, b/2\right)\Delta c = k_1\, a\, b - k_2\, b\, c \tag{33}$$

$$\left(-k_2\, c/2\right)\Delta b + \left(\frac{1}{\Delta t} - k_2\, b/2 + k_3/2\right)\Delta c = k_2\, b\, c - k_3\, c \tag{34}$$

Thus we have two equations, 33 and 34, and two unknowns, Δb and Δc, and the latter are most readily obtained by matrix algebra as

$$\Delta b = \frac{\begin{vmatrix} k_1 ab - k_2 bc & k_2\, bc/2 \\ k_2 bc - k_3 c & 1/\Delta t - k_2\, b/2 + k_3/2 \end{vmatrix}}{\begin{vmatrix} 1/\Delta t - k_1\, a/2 + k_2\, c/2 & k_2\, b/2 \\ -k_2\, c/2 & 1/\Delta t - k_2\, b/2 + k_3/2 \end{vmatrix}}$$

$$= \frac{(k_1 ab - k_2 bc)\,(1/\Delta t - k_2\, b/2 + k_3/2) - (k_2\, b/2)\,(k_2 bc - k_3 c)}{(1/\Delta t - k_1\, a/2 + k_2\, c/2)\,(1/\Delta t - k_2\, b/2 + k_3/2) - (k_2\, b/2)\,(-k_2\, c/2)} \tag{35}$$

and

$$\Delta c = \frac{\begin{vmatrix} 1/\Delta t - k_1\, a/2 + k_2\, c/2 & k_1 ab - k_2 bc \\ -k_2\, c/2 & k_2 bc - k_3 c \end{vmatrix}}{\begin{vmatrix} 1/\Delta t - k_1\, a/2 + k_2\, c/2 & k_2\, b/2 \\ -k_2\, c/2 & 1/\Delta t - k_2\, b/2 + k_3/2 \end{vmatrix}}$$

$$= \frac{(1/\Delta t - k_1\, a/2 + k_2\, c/2)\,(k_2 bc - k_3 c) - (k_1 ab - k_2 bc)\,(-k_2\, c/2)}{(1/\Delta t - k_1\, a/2 + k_2\, c/2)\,(1/\Delta t - k_2\, b/2 + k_3/2) - (k_2\, b/2)\,(-k_2\, c/2)} \tag{36}$$

from which we can compute b_n as $b_{n-1}+\Delta b$ and c_n as $c_{n-1}+\Delta c$. Such results are also shown in Figure 4 where, for a column of 1001 rows, with $\Delta t = 0.01$, the semi-implicit method generates results that are visually indistinguishable from those obtained with the explicit method supplemented with functions to reduce their effective Δt to 0.001.

It is the interplay between the two autocatalytic reactions that causes the oscillatory behavior. The conversion of A into B in reaction 20 is catalyzed by the reaction product B, while the conversion of B into C is catalyzed by C. When the concentration of C increases, it will speed up the decomposition of B, so that the corresponding concentration *b* will decrease. This is clearly visible in Figure 4. However, the decrease in *b* will lead to a decrease in the rate of production both of B, through reaction 20, and of C, through reaction 21. Depending on the numerical values of the rate constants involved, *b* may recover while, initially, *c* remains low, in which case the process may become cyclic. In that case, the concentrations *b* and *c* will reach their maximum values at different times: while both are cyclic, they are out of phase with each other. This is clearly seen in Figure 4. Another way to display this behavior is to plot, for example, *c* versus *b*, as in the *phase diagram* of Figure 5, which shows a *limit cycle*. Figures 4 and 5 represent the same information in different formats, viz. in the time and frequency domains, respectively. Such alternative represen-

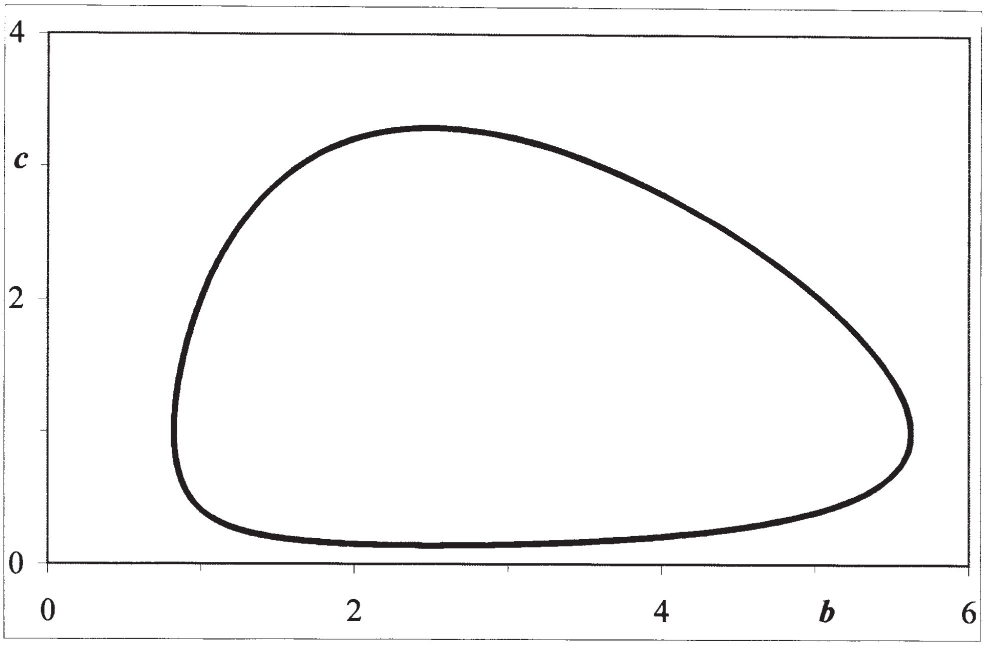

FIGURE 5. The phase diagram, that is, a plot of the concentration *c* of species C as a function of the concentration *b* of species B. The data plotted are the same as those used for the solid curves in Figure 4.

tations are of course readily made on the spreadsheet, with its convenient, built-in graphing capabilities.

IX. SUMMARY

The first example given dealt with unidirectional first-order kinetics and introduced the different approaches used in the explicit and semi-implicit methods. The second case considered, the establishment of a monomer-dimer equilibrium, was of intermediate complexity, and showed how numerical simulation treats higher-order processes that often make closed-form mathematical solutions unattainable. This case also served to introduce user-defined functions. The final example illustrated a fairly complicated case, the Lotka model of two coupled autocatalytic reactions. In all of these, the spreadsheet displays all intermediate steps, and clearly shows what tradeoffs are made. In all three cases we have used initial values of Δt that were rather large, so that we could demonstrate the effects of numerical errors.

In the rather demanding case of an oscillatory reaction, where we require a long time range in order to sample several cycles of the oscillation, we most clearly recognize the different options to obtain high simulation accuracy on a spreadsheet. These options are (1) to supplement the explicit method with functions, or (2) to use the semi-implicit method. While the latter method is much more efficient, it also requires considerably more effort to formulate. The choice between the two approaches may well depend on the available computer power: on a 486-type computer the use of functions may make the calculations unacceptably slow, and the more efficient implicit method might be preferable, whereas creature comfort might tip the balance the other way when a much faster cpu is available.

There are yet other methods than those described here, such as the 4th order Runge-Kutta method, that can also be incorporated in the spreadsheet as a function[7]. The material presented here is only intended as an introduction. Problems can arise when the rate constants are strongly dissimilar, especially in periodic systems such as oscillating reactions, which may exhibit fast features separated by long time periods during which there is little change. For example, in the Belousev-Zhabotinski reaction, there may be dramatic concentration changes occurring well within 0.3 s but repeating themselves only once every 300 s. In that case, the sharp concentration change requires that the simulation uses small time increments Δt of, say, 3 ms, but this would cause the simulation of a few repeat cycles to become excessively long. Fortunately, once the problem is recognized as such, its solution becomes obvious: the step size Δt must be reduced whenever one of the concentrations changes faster per step than a given criterion. Such *adaptive* algorithms[8] can be implemented on a spreadsheet by using IF statements in the column for time *t*.

Systems of differential equations with rate parameters of greatly differing magnitudes are called *stiff,* and were recently discussed by Bond et al.[9] Methods for solving stiff problems can be found in, for example, Gear's book[10] and in a collection of chapters edited by Aiken.[11]

Simulations can illustrate kinetic phenomena that cannot be described in terms of closed-form analytical solutions. This includes complex reaction mechanisms as well as simpler (e.g., first-order) reaction schemes run under nonisothermal conditions.[12,13] They can also be used to establish the ranges of applicability of various approximations.[14] Another use of simulations is to generate model curves with which experimental data can be compared, and possibly fitted. Again, this can be achieved readily on a spreadsheet, by using some fitting criterion, such as minimizing the sum of squares of the differences between the simulated and the experimental data points. Excel has a convenient function for this purpose, SUMXMY2(Pi:Pj, Qi:Qj), where P and Q represent the two columns of data to be compared, and i and j the first and last rows of these. Crude parameter adjustment should initially be done by hand (which is often faster than one might anticipate) in order to get approximate rate parameters. The result can then be refined efficiently with Solver, the multiparameter Levenberg-Marquardt routine provided with Lotus 1-2-3, QuattroPro, and Excel.

Apart from its widespread availability, an advantage of using a spreadsheet for these computations is primarily a didactic one: the treatment is completely transparent to the student user, nothing is hidden or magical, even in cases such as the oscillating reactions that, initially, may appear baffling. But the principles are equally valid in state-of-the art research, as in a fairly recent simulation[15] of electrochemical oscillations, from basic electrochemical principles and independently determined rate parameters, to well within the experimental reproducibility of such a system.

The treatment demonstrates that even quite complicated kinetic phenomena, such as those resulting from coupled autocatalytic reactions, can be understood in terms of the basic concepts of chemical reaction dynamics. Demystifying complex behavior by showing it to be the logical consequence of simple underlying principles is one of the main objectives of science education, and often requires a numerical approach.

REFERENCES

1. de Levie, R., How to use Excel in anayltical chemistry and in general scientific data analysis, Cambridge Univ. Press, Cambridge 2001.
2. Billo, E. J., *Excel for Chemists, a comprehensive guide*, Willey, New York 2001.
3. Liniger, W., Willoughby, R. A., Efficient integration methods for stiff systems of ordinary differential equations, *SIAM J. Numer. Anal*. **1970, *7,* 47.**
4. Landolt, H., Über die Zeitdauer der Reaktion zwischen Jodsäure und Schwefliger Säure, *Ber. Deut. Chem. Ges.* **1886**, *19*, 1317.
5. Lotka, A. J., Undamped oscillations derived from the laws of mass action, *J. Am. Chem. Soc.* **1920**, *42*, 1595.
6. Lotka, A. J., Analytical note on certain rhythmic relations in organic systems, *Proc. Natl. Acad. Sci. USA* **1920**, *6*, 410.
7. de Levie, R., *Advanced Excel for science and engineering*, Oxford Univ. Press **2002**.
8. Prokopakis, G. J., Seider, W. D., Adaptive semi-implicit Runge-Kutta method for solution of stiff ordinary differential equations, *Ind. Eng. Chem. Fundam*. **1981**, *20*, 255.
9. Bond, R. A. B., Martincigh, B. S., Mika, J. R., Simoyi, R. H., The quasi-steady-state approximation: numerical validation, *J. Chem. Educ*. **1998**, *75*, 1158.
10. Gear, C. W., *Numerical initial value problems in ordinary differential equations*, Prentice Hall, Englewood Cliffs N.J. **1971**.
11. Aiken, R. C., ed., *Stiff Computation*, Oxford University Press, Oxford **1985**.
12. Ricci, R. W., van Doren, J. M., Using dynamic simulation software in the physical chemistry laboratory, *J. Chem. Educ*. **1997**, *74*, 1372.
13. Hodgson, S. C., Ngeh, L. N., Orbell, J. D. , Bigger, S. W., A student experiment in non-isothermal chemical kinetics, *J. Chem. Educ*. **1998**, *75*, 1150.
14. Goodman, J. M. How do approximations affect the solutions to kinetic equations? *J. Chem. Educ*. **1999**, *76*, 275.
15. Rudolph, M., Hromadova, M., de Levie, R. Demystifying an electrochemical oscillator, *J. Phys. Chem.* **1998**, *102,* 4405.