Teaching Mathematics to Students of Chemistry with Symbolic Computation

J. F. Ogilvie* and M. B. Monagan

Simon Fraser University, *Centre for Experimental and Constructive Mathematics*, *Department of Mathematics, 8888 University Drive*, *Burnaby*, *British Columbia V5A 1S6 Canada*
**also Universidad de Costa Rica*, *Escuela de Quimica*, *Ciudad Universitaria Rodrigo Facio*, *San Pedro de Montes de Oca*, *San Jose 2060*, *Costa Rica*
electronic mail ogilvie @ cecm.sfu.ca *and* mmonagan @ cecm.sfu.ca

abstract

We explain how the use of mathematical software improves the teaching of mathematics to, and its understanding by, students of chemistry, while greatly expanding their capabilities to solve realistic chemical problems. After an explanation of the need to improve this teaching and the opportunity with symbolic computation for this purpose, we outline the content of curriculum and its implementation, and provide examples of pertinent applications from thermodynamics and chemical kinetics.

Why should we teach mathematics with computers?

Students of chemistry find mathematics difficult: some students entering a post-secondary institution even select chemistry rather than physics because they think that they might thereby avoid much mathematics. Even while chemistry has become more mathematical during the past half century, largely because of an increasing prominence of statistics in analytical chemistry and chemometrics and of quantum mechanics in physical chemistry that diffuses into inorganic and organic chemistry, there has been a tendency for the number of courses in mathematics required of a student with chemistry as major subject to decrease significantly. For instance, at Simon Fraser University, in 1997 the requirements for chemistry as a major subject included five courses in mathematics – two first-year courses in differential and integral calculus, two second-year courses in multivariate calculus and linear algebra and a third-year course in differential equations; in 1998 the course on differential equations became no longer required.

During the same period, the methods of undertaking calculations have likewise altered, in a progression from use of tables of logarithms and of slide rules, through pocket calculators with basic arithmetical operations, to powerful and large digital computers with software possessing ever increasing capabilities, eventually to ubiquitous graphic calculators for the pocket and computers on most desks and on many shoulders. Whereas before 1970 children in primary school learned how to extract square roots manually, since that era the topic has practically vanished from curricula: the standard method to calculate a square root now involves depressing an appropriate button on a calculator. Likewise, during the latter decades computers have evolved from being rare, huge and expensive machines devoted to mainly scientific and technological applications to become compact and inexpensive devices for which, at least in a common domestic or commercial environment, technical applications are typically peripheral, even while their computational power and other properties have enormously increased.

Within the same past half century there has been some evolution in the teaching of mathematics, from a formal and abstract approach based largely on theorems to a more pragmatic and less systematic development, and to *service* courses, with decreased numbers of courses or hours of classes notwithstanding their intent to cover material over an increased range. The use of computers in the present conditions is non-uniform: in some institutions courses are taught with greater or lesser invocation of computer algebra; in North America, the programs *Maple* (*1*) and *Mathematica* (*2*) predominate. Even within a particular university this practice might vary from one instructor to another; the result is that students progressing from one course to the next are subject to conflicting philosophies of pedagogy and disparate expected standards of competence related to manual or machine execution. In many cases, when computers have become involved, the content and delivery of standard courses have simply been developed in an isolated context, retaining a traditional sequence and scope of topics.

Taking into account both the learning capabilities of students of chemistry and the heuristic applicability of computer software, we contend that a radical reorganization of the teaching of mathematics to these students is both timely and feasible (*3*). Our concern here is with the mathematical material typically taught by mathematicians, rather than the mathematics of chemistry, such as solutions of Schrodinger's equation for prototypical systems and ‘group theory’ or symmetry that are generally taught within particular chemistry courses. Programs for computer algebra or symbolic computation that operate readily on all current computers, even some devices small enough to fit in a pocket, not only possess embedded mathematical knowledge accumulated over thousands of years during the development of civilization but also might include material primarily directed toward the teaching of that knowledge. Moreover, new features are being continually added to some products specifically for instruction; instructors who were disappointed with software available for teaching purposes a decade or more ago should reexamine the current programs.

We assert also that an holistic approach to the teaching of these mathematics at a post-secondary level is obligatory, so as to optimize the progress of a student through not only the newly encountered mathematical topics but also their implementation with the selected software: instead of merely trying to convert existing courses within a traditional pattern, we must consider the total extent of mathematical knowledge and capability reasonably expected to be acquired by chemistry students, and chart a course through that material in association with chosen software. The scope of applications is not only their immediate chemical courses but even their entire technical career to follow, for which undergraduate studies are a direct or indirect preparation; we must organize the content of mathematical courses accordingly.

The teaching of mathematics that is strongly based on symbolic computation allows an instructor to explore a topic or principle according to four points of view:

- a formal statement is devised in words, just as according to tradition, but with increased emphasis on explanations of both pertinent terms and their inter-relations according to an accessible dictionary or encyclopaedia of mathematics;

- an algebraic or symbolic treatment can expand to take advantage of the speed and scope of software for algebraic operations, instead of leaving a student bemused with "it can be shown that ...";
- numerical illustrations, with test cases over a large range, are readily generated through simple repetition constructs, and numerical techniques, such as construction of splines and their applications, are effortlessly applied to complement the algebraic aspects;
- not only striking graphics are readily produced, in two and three dimensions, taking full advantage of colouring and contouring, but also dynamic animations of mathematical processes to portray geometrical interpretations; such a capability is consistent with a pertinent adage "a picture is worth a thousand words", and that picture can remain branded into the memory of a student long after algebraic details are abjectly forgotten.

The capacity of contemporary software for symbolic computation to produce outstanding plots is astonishing; teaching mathematics without use of such displays is incontestably inferior. Rather than being distracted from the significance of a particular topic by tedious details of mathematical operations, teaching mathematics directly with computer algebra enables an instructor to convey, and his or her students to acquire, profound mathematical insight and understanding of the concepts and principles through the above four approaches in combination. In a course in mathematics, emphasis on concepts and reasoning can hence replace drill on technical details of manipulation required to solve routine exercises, and plots of geometric constructions can underpin those concepts to enliven the reasoning.

We illustrate with an example that had a profound impact on us personally, increasing our own understanding of eigenvectors. Suppose that we seek to introduce the concept of the eigenvectors of a matrix. In a typical textbook on linear algebra one would begin with a definition.

> Let $\mathbf{A}$ be a $n \times n$ matrix over the set of real numbers. A non-zero vector $\underline{v}$ is called an eigenvector of $\mathbf{A}$ if there exist a scalar λ called an eigenvalue of $\mathbf{A}$, such that $\mathbf{A}\,\underline{v} = \lambda\,\underline{v}$.

From this point, how do we proceed? A typical instructor might show an example and proceed to develop the method of calculating the eigenvalues and eigenvectors of $\mathbf{A}$ via the characteristic polynomial of the matrix $\lambda\, I - \mathbf{A}$. Some properties of eigenvectors are presented, some applications are presented. Three years later, what will the student remember about eigenvalues and eigenvectors? Will the student know what an eigenvector is? Will he or she remember even the definition?

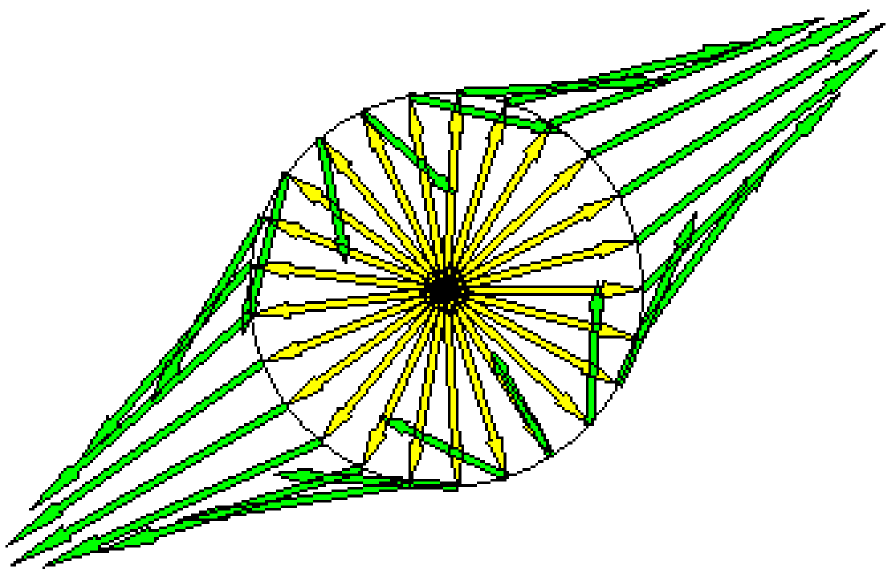
We contend that the following plot provides a much deeper understanding of what the eigenvectors are, and it fixes the definition in the mind of the student for years to come. This particular 2×2 matrix is called Fibonacci's matrix. In what follows, *Maple* input lines begin with the character > ; these are *Maple* commands that we have typed. If required, the output from *Maple* appears after such input.

```
> with(Student:-LinearAlgebra):
> A := Matrix([[1,1],[1,0]]);
```

$$A := \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

```
> EigenPlot(A,numvectors=25,showeigenvectors=false,
                 showunitvectors=true);
```

The Images of Unit Vectors and Eigenvectors



Shown in the plot are 25 unit vectors. For each unit vector $\underline{u}$, we have computed $\underline{v} = \mathbf{A}\,\underline{u}$ and displayed the vector $\underline{v}$ with its tail placed at the head of $\underline{u}$. Thus the graphic provides an enlightening picture of what happens on multiplication by **A**. We have deliberately suppressed the display of the eigenvectors. Can you estimate what the eigenvectors and the eigenvalues are? From the definition, it follows that, if $\underline{v}$ were an eigenvector of **A**, it should lie in the same direction as, or opposite direction to, the corresponding $\underline{u}$. We can see one such vector that almost points in the direction approximately [3, 2]; this vector $\underline{v}$ is about 1.5 times as long as the corresponding vector $\underline{u}$. Thus we have one eigenvector $\underline{v}_1 \approx [3, 2]$ with eigenvalue $\lambda_1 \approx 1.5$. We see another eigenvector $\underline{v}_2$ in the opposite direction of a unit vector $\underline{u}_2$ in the direction $[-2, 3]$; the length of $\underline{v}_2$ is just under half the length of $\underline{u}_2$. Hence we have a second eigenvector $\underline{v}_2 \approx [-2, 3]$ with eigenvalue $\lambda_2 \approx -0.5$. We can see much more from this plot: for instance, repeated multiplication of the vector $\underline{w} = [1, 1]$ by the matrix **A** causes the magnitude of $\underline{w}$ to increase and its direction to approach the first eigenvector. This plot offers us a first understanding of what a stable eigenvector is, and, indeed, a method to compute it. For further visual examples, including examples of 3×3 matrices visualized in three dimensions, we refer the reader to (*4*).

Students of chemistry traditionally receive instruction in mathematics either through *service courses* offered in departments of mathematics for other than their own major students, or, less commonly, directly by instructors in chemistry. In agreement with Simons (*5*), symbolic computation can serve to make trivial the traditional service courses in mathematics; although Simons wrote in a context of teaching mathematics to students of engineering, exactly the same logic and argument are applicable to a chemical context. The content of these traditional service courses in mathematics has been developed in the light of needs of users of mathematics before the era of accessible computers; a revised course must emphasize concepts and their implementation with software, rather than manual techniques. For that reason a service course in mathematics in which the calculations are performed with appropriate software must have a different content, and emphasis on different skills, from those of a traditional course. Likewise, the use of computer algebra increases the level of what a student can achieve without much understanding of technical details; for instance, a few general commands in relation to exact or numerical solution of differential equations can replace instruction in a long sequence of particular methods applicable to individual cases. Service courses in mathematics that concentrate on solution of exercises, in linear algebra or differential equations for instance, can be transformed into courses on use of software for computer algebra. Although understanding essentially all major concepts of mathematics traditionally taught in service courses for undergraduates is not difficult for most students of chemistry, applicable manual techniques to implement those mathematical principles might be tedious; in that sense many students of chemistry find mathematics difficult and repulsive, because not everybody has the capacity to be successful in such techniques. When the tedious details are executed with a computer, the material becomes tractable and attractive. In contrast, superior students who find such manual techniques boring can benefit from learning mathematics at a higher level of concepts and applications; software for computer algebra is again a valuable tool for enriching the curriculum.

What should courses of mathematics taught with symbolic computation contain?

First, we must appreciate and accept that teaching with, or use of, computers for mathematical operations incurs an overhead in the form of learning to use the particular software. Although some programs have a steeper learning curve than others, any software imposes on a user the constraint that he or she must comply with conventions of that particular software. The design of curriculum must hence include an explicit initial component of instilling acquaintance with common commands and conventions of chosen software. After a student gains familiarity with one program, switching to another program is not particularly challenging, because almost all programs for computer algebra operate in similar manners. It is important for a prospective instructor to recognize that, if a student learns and practises mathematics with computer software, that student must be expected to use and to apply that software generally; for instance, an examination written without access to a computer is counter-productive: a student will question why he or she should bother to learn to use the software if that skill is irrelevant for the eventual assessment in the course.

The plan of curriculum that we proffer begins at an almost zero formal level of mathematical knowledge: arithmetic with integers, rational fractions, real numbers, random numbers and complex numbers, including relevant aspects of the *International System of Units, Symbols and Notation*, is followed by simple algebraic operations and solution of equations of various types. This beginning permits a student to become acquainted with the *language* of a symbolic processor whilst making no significant demands on the assimilation of difficult mathematical concepts; an instructor need not, however, hesitate to include aspects of number theory, for instance, and of sets and other formal mathematical structures that were likely absent from courses in preceding school years. The next large component of the total program of mathematical study begins with elementary functions – exponential and logarithmic, and forming and working with plots of various kinds; structures of simple molecules and unit cells of prototypical inorganic compounds are an immediate application of plotting in three dimensions. This basis provides an excellent platform for a review of descriptive geometry: triangles in some detail, quadrangles and other polygons, but also polyhedra from a tetrahedron that might represent the structure of P_4, to a truncated icosahedron to model the shape of C_{60}, each with its geometric properties. Trigonometry follows, including circular and hyperbolic functions, and their inverses, that are readily inter-related in a way that is never envisaged within a school environment. Although complex numbers in cartesian form were previously introduced as numbers of a particular kind, with trigonometry and plots other properties can be explored through their polar forms; conformal plots provide enlightenment about significant aspects of complex analysis. Properties of series, polynomials and rational functions, and their applications in exact fits of data through polynomials and splines of varied degree, are standard. Hence, although much of this material might be considered to be a precursor to study of mathematics at university level, one can avail of the opportunity to include, without undue strain on students, additional related topics still at a rather simple level; with astounding graphics to aid assimilation of concepts, such new topics actually serve to stimulate the interest of students at the same time that much emphasis is placed on learning the commands of the symbolic processor, with chemical illustrations and applications where appropriate, rather than to present fresh material in a concerted manner.

Even though some introduction to calculus might have been attempted in schools, the study of mathematics in traditional courses at tertiary educational level by students of chemistry begins typically with calculus, first differentiation then integration, progressing to multivariate calculus. When comparable topics are taught with a symbolic processor, a student might be amazed to discover the power of a few commands or operators to implement all calculus, relative to the many commands and operators associated with the preceding topics. Traditional textbooks on calculus typically omit or neglect numerical differentiation and integration, but such topics are strongly relevant to the processing of numerical data collected in a chemical laboratory; their inclusion within a course taught with computer algebra poses no difficulties in either comprehension or implementation: the computer does the tedious calculations. Fourier series constitute a topic readily explored as an application of integration. Plots of rotatable surfaces in three dimensions are readily generated with computer graphics to illuminate aspects of directional or partial derivatives, whereas comparable renderings made free-hand by an instructor on a blackboard must be much less inspiring. Operations with thermodynamic state functions and use of Lagrange multipliers in constrained optimization illustrate important applications of derivatives of functions of multiple variables. When a student has grasped the significance of a derivative through geometric constructs in first two and then three dimensions through appropriate plots, including animations, extension to variables of increased number typical of a chemical problem, which are handled with the software just as readily as for a single independent variable, is straightforward.

The power of a symbolic processor is most prominent in application to linear algebra, differential equations and statistics. No longer must an instructor restrict examples of matrices to those of second or third order, but he or she can work effortlessly with matrices of, for instance, sixth order that might be applicable to an important chemical system such as benzene. Operations with vectors, arrays, eigenvalues and eigenvectors, vector calculus and even tensors of second order, amply illuminated with striking plots, are simple to understand and readily executed with symbolic software. Plots of a slope, or direction, field impress upon a student the meaning of a differential equation of first order in a manner that is impracticable without such a plot, and again laborious efforts by an instructor on a blackboard are largely ineffectual compared with the impact of a plot generated instantly with a symbolic processor. Solution of differential equations, single or within sets, pertaining to prototypical cases in chemical kinetics poses no problem for a symbolic processor, but has immediate relevance to obvious chemical applications. Numerical solution of ordinary or partial differential equations is extremely tedious by hand, but just a few commands to contemporary symbolic processors readily yield accurate results. Statistics, ranging from probability through distributions, linear and nonlinear regression to optimization, provide another instance of tedious human manipulation because of the extent of the data that must be treated in a realistic case. Although some calculators provide multiple statistical functions, the additional and profound capability of advanced software on a standard computer is a valuable asset in teaching statistics and its embodiment in chemometrics.

Implementation of courses and examples of applications of symbolic software

Will any professor of chemistry be unhappy if his or her undergraduate students both understood the concepts of all topics mentioned above and are able to execute the corresponding mathematical operations to solve chemical problems? What is even more enticing about this approach is that the total duration of formal courses, comprising lecture demonstrations and supervised practice sessions at a weekly rate of two or three hours each for lectures and practice, might require as little as one year (or equivalent), although courses through three semesters would likely place an optimal pressure on students to meet applicable standards of proficiency.

The most obvious characteristic of a textbook to support such courses would be that it should have an electronic nature, operating on a computer with particular software in a truly interactive manner; it should provide both explanatory text, including mathematical definitions and explaining concepts and principles, and description of commands and elucidation of their results. A reader interactively executes commands that readily lend themselves to experimenting with values of parameters; sequences of commands illustrate the concepts and implement pertinent operations in a concerted sequence through the entire progression of topics. Although traditional contemporary textbooks of mathematics typically contain dozens or hundreds of problems per chapter, few purely mathematical exercises suffice in a context of computer algebra because generally the same command can effect operations on diverse functions that might be explored manually; instead, problems involving truly chemical applications can be assigned. Such a textbook is eminently suitable even for self study, but students likely expect, and can in most cases profit from, lecture classes in almost a traditional format, in which an instructor relies mostly on a computer display to accompany explanation, rather than presenting all traditional material with the aid of a blackboard.

Despite the availability of software for pedagogical purposes to operate in association with some symbolic processors, one factor that has hampered the widespread application of symbolic computation for the teaching of mathematics has been the lack of a specially designed textbook for this purpose. Even though software for symbolic computation has become remarkably enhanced and extended since 1997 when Simons wrote his provocative essay (*5*), great emphasis in the development of software has been placed, in some notable cases, on *packages* designed specifically for pedagogical purposes. Such a lack of an appropriate textbook has been remedied with the preparation of at least one interactive electronic textbook of mathematics for chemistry (*6*) of a type and content described above. For use in composite classes of students from multiple science departments, it would be highly desirable for each student to have a textbook with examples and exercises designed for each particular subject; a mathematician as instructor could then concentrate on mathematical concepts and their implementation in purely mathematical contexts with which he or she might feel most comfortable, but a student could apply the particular examples and exercises pertaining to his subject to gain an improved knowledge of his or her field. At this time of writing, no comparable textbook is, however, available for biology, geology or physics for instance, although there are generally several worthy printed books that might supplement conventional textbooks by supporting the use of a symbolic processor for applications of mathematics to each other subject. Experience has shown, however, that an interested professor of chemistry without an exceptionally profound knowledge of mathematics can employ that textbook (*6*) to teach mathematics to chemistry students.

At this point, a reader might be curious – or even expect – to see displayed here some examples of advantageous application of symbolic computation to teach mathematics to chemistry students. Although one can readily invoke many such examples, naturally their presentation on a printed page falls far short of their impact on a live computer screen; such examples include animation of a plot of a Riemann sum approaching a limit of a definite integral, illustration of a distinction between image vectors and eigenvectors when a matrix of third order acts on vectors on a unit sphere, visual illustrations of partial derivatives in three dimensions and so forth.

As three explicit illustrative examples, we present one from linear algebra and two from differential equations, for which we have employed commands in *Maple* (*1*). In our first example, not particularly chemical, we form a 3×3 matrix **A** with numeric elements,

```
> with(Student:-LinearAlgebra):
> A := Matrix([[1,1,1],[2,-1,0],[1,-2,-1]]);
```

$$A := \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & 0 \\ 1 & -2 & -1 \end{bmatrix}$$

and a column vector $\underline{b}$ with these numeric components,

```
> b := Vector([2,3,1]);
```

$$b := \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

With this intuitively obvious command we construct the augmented matrix,

```
> Ab := <A|b>;
```

$$Ab := \begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & -1 & 0 & 3 \\ 1 & -2 & -1 & 1 \end{bmatrix}$$

and convert it to reduced row-echelon form, which yields this result:

```
> ReducedRowEchelonForm(Ab);
```

$$\begin{bmatrix} 1 & 0 & \frac{1}{3} & \frac{5}{3} \\ 0 & 1 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

For this purpose, *Maple* has evidently applied exact arithmetic, not decimal numbers, just as an instructor would use: the closer that the output from the software is to what one would do in class, the more confident the student becomes in proceeding to the next step. With a *Maple* command,

```
> LinearSolve(A, b, free=t);
```

$$\begin{bmatrix} t_1 \\ 2\,t_1 - 3 \\ -3\,t_1 + 5 \end{bmatrix}$$

we solve this linear system. Instead of an error message being presented because the system of linear equations exhibits a linear dependence, the solution is presented in terms of a parameter *t* in the same way that we would present the solution set by hand on the blackboard. A command to form a plot for which one specifies the matrix **A** and the vector $\underline{b}$ yields a three-dimensional plot containing three solid circles, coloured red, blue and green, that all intersect along a line, consistent with the solution above. This graphic facility can not be underestimated: an instructor might easily sketch points, lines and vectors in a plane, but on a blackboard the latter diagram in three dimensions is difficult to draw quickly, and is not rotatable, unlike the computer display. For the same reason, to show this diagram on a printed page is difficult; if *Maple* be available to the reader, this *Maple* command generates the plot.

```
> LinearSystemPlot(A,b);
```

A typical traditional course on differential equations comprises a sequence of recipes for solving differential equations of several selected types; although it is valuable and necessary for a student to be able to solve some equations by hand, it is poor use of a student's time to devote an entire course to this activity. For students of chemistry the course should emphasize modeling and applications, teaching students how to construct differential equations, or systems thereof, from a chemical or physical model. For instance, as a problem in thermodynamics, consider two bodies, A and B, at different temperatures, that are placed in contact. Heat can then flow from one body to another as well as from each body to the surroundings; the temperatures of the bodies vary with time, as A(t) and B(t). We can then write these two coupled differential equations.

```
> des := diff(A(t), t) = -k[1]*(A(t)-T[m]) + k[2]*(B(t)-A(t)),
     diff(B(t), t) = -k[1]*(B(t)-T[m]) - k[2]*(B(t)-A(t)) + F;
```

$$des := \frac{d}{dt}\mathrm{A}(t) = -k_1\,(\mathrm{A}(t) - T_m) + k_2\,(\mathrm{B}(t) - \mathrm{A}(t)),$$
$$\frac{d}{dt}\mathrm{B}(t) = -k_1\,(\mathrm{B}(t) - T_m) - k_2\,(\mathrm{B}(t) - \mathrm{A}(t)) + F$$

Here T_m is the temperature of the surroundings that can accept all heat isothermally, and F denotes a constant positive flux of heat being supplied directly to object B; positive coefficients

k_1 and k_2 pertain to the rates of transfer of heat from either body to the surroundings and from one body to the other, respectively. Perhaps the most important feature of a processor for symbolic computation such as *Maple* is that the display of a mathematical formula is typeset, and resembles closely the way that it would appear on the blackboard or in a textbook.

We seek to compute the temperatures of the two bodies in a steady state in terms of parameters k_1, k_2 and F. Invoking a command to solve two linear simultaneous equations in this set for the temperatures of A and B in a steady state, i. e. when both derivatives in the left sides are equal to zero, and simplifying the resulting expressions yields the following results.

```
> sys := {0 = -k[1]*(A - T[m]) + k[2]*(B - A),
          0 = -k[1]*(B - T[m]) - k[2]*(B - A) + F};
```

$$sys := \{0 = -k_1 (A - T_m) + k_2 (B - A), 0 = -k_1 (B - T_m) - k_2 (B - A) + F\}$$

Solving the system, we write the solutions as polynomials in F, and simplify their coefficients.

```
> TempSteadyState := collect(solve(sys, {A,B}), T[m], simplify);
```

$$TempSteadyState := \left\{A = T_m + \frac{k_2 F}{k_1 (k_1 + 2 k_2)}, B = T_m + \frac{F (k_1 + k_2)}{k_1 (k_1 + 2 k_2)}\right\}$$

Expressing the solutions in this way using symbolic computation provides physical insight; according to these results one can directly understand that the temperature in the steady state is directly proportional to F, that, if $F = 0$, the temperature in the steady state is that T_m of the surroundings, and that otherwise object B is hotter than A – this is a proof! With the available graphical capability and inserting appropriate values of the parameters, we can show a phase-portrait plot with an initial-value solution, and make an animation as a function of the ratio of k_1 and k_2. Such plots exhibit the physical or chemical principles and activities, and confirm the algebraic results. We can show no animation here on the printed page, but, for $F = 5$, $k_1 = 0.1$, $k_2 = 0.3$ and $T_m = 0$ in appropriate units, we show one frame, with four solution curves; these curves show the extent, at time $t = 10$, of the approach of the temperatures of both bodies A and B to the steady state from four initial conditions, in which one or other, or both, bodies begin at 0 or 35 degrees.

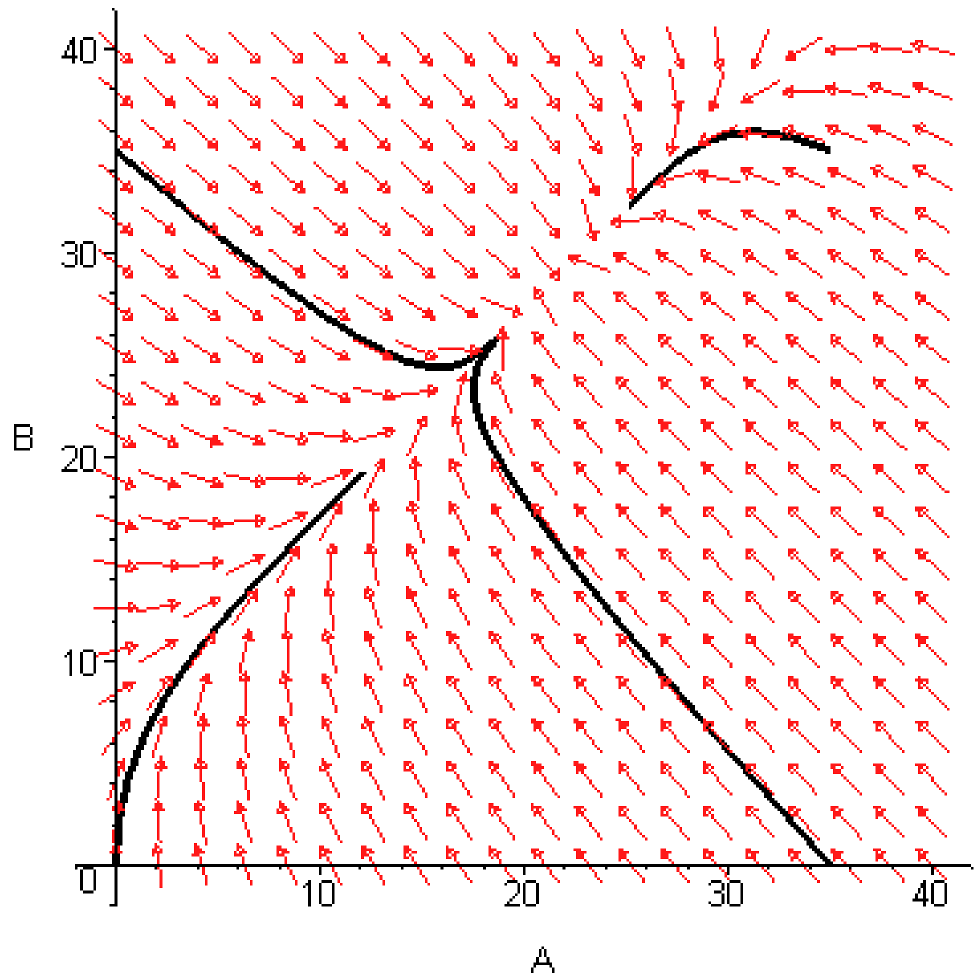
```
> (k[1], k[2], F, T[m]) := (0.1, 0.3, 5, 0);
```

$$k_1, k_2, F, T_m := 0.1, 0.3, 5, 0$$

```
> ivs := {[A(0)=0, B(0)=0], [A(0)=0, B(0)=35], [A(0)=35,
            B(0)=0], [A(0)=35, B(0)=35]};
```

$ivs := \{[A(0)=0, B(0)=0], [A(0)=0, B(0)=35], [A(0)=35, B(0)=0],$
$[A(0)=35, B(0)=35]\}$

```
> DEtools[DEplot]( [des], [A(t),B(t)], t=0..10,
  A=0..40, B=0..40, ivs, linecolour=black, arrows=medium);
```



We compute to three digits the temperature at the steady state.

```
> evalf[3](TempSteadyState);
```

$$\{B = 28.6, A = 21.4\}$$

As a final example genuinely appropriate to chemical kinetics, we consider a chemical system of two reactants, A and B, that combine to form product C,

$$A + B \xrightarrow{k_f} C$$

but C also dissociates to reform A and B;

$$C \xrightarrow{k_r} A + B$$

the coefficients of rates of forward, k_f, and reverse, k_r, reactions have similar magnitudes. We use here A, B and C as both the names of reactants and their respective concentrations. With initial concentrations *a*, *b* and *c*, the differential equation for the loss of reactant A in this kinetic system,

$$-\,dA(t)/dt = k_f\,A(t)\,B(t) - k_r\,C(t)$$

is expressed in this *Maple* statement with x(*t*) as the extent of depletion of A at duration *t* after the onset of reaction.

```
> eqA := - Diff((a - x(t)),t) = k[f]*(a - x(t))*(b - x(t))
                                        - k[r]*(c + x(t));
```

$$eqA := -\left(\frac{\partial}{\partial t}(a - x(t))\right) = k_f\,(a - x(t))\,(b - x(t)) - k_r\,(c + x(t))$$

Although *Maple* can produce an algebraic expression as the solution to this equation with *a*, *b*, *c*, k_f and k_r in symbolic form, to avoid complicated formulae it is preferable here to apply numerical values for these parameters, as follows.

```
> a := 1:  b := 2:  c := 0:  k[f] := 1:  k[r] := 1/5:
```

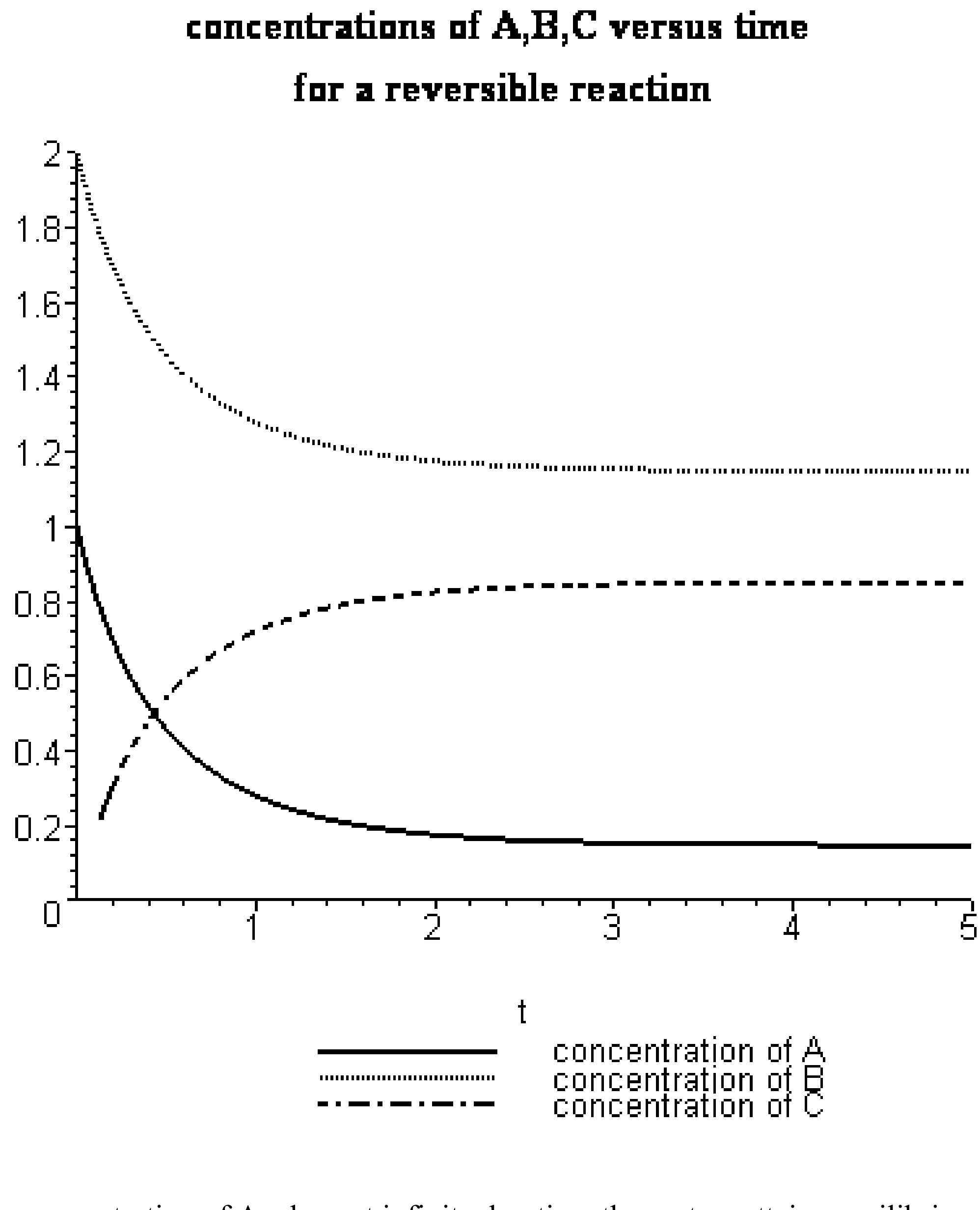
With the initial condition x(*t*) = 0 at *t* = 0, the differential equation is then readily solved with this command.

```
> sol := dsolve( {eqA, x(0)=0}, x(t) );
```

$$sol := x(t) = -\frac{1}{35}\left(-4\sqrt{14} + 7\tanh\left(\frac{1}{70}\left(14\,t + 5\sqrt{14}\,\operatorname{arctanh}\left(\frac{4\sqrt{14}}{7}\right)\right)\sqrt{14}\right)\right)\sqrt{14}$$

We plot the results, which show how the concentrations of reactants A and B decrease from their initial values, eventually becoming constant but neither zero for A nor unit concentration for B, whereas the concentration of C increases from its initially assigned zero value to a constant value.

```
> plot( [a-rhs(sol), b-rhs(sol), c+rhs(sol)], t=0..5, 0..2,
  colour=[red,blue,green], titlefont=[TIMES,BOLD,12], title=
  "concentrations of A,B,C versus time\n for a reversible
reaction");
```



We calculate the concentration of A when, at infinite duration, the system attains equilibrium.

```
> Aequil := limit( a-rhs(sol), t=infinity );
```

$$Aequil := -\frac{3}{5} + \frac{\sqrt{14}}{5}$$

```
> evalf(Aequil);
```

0.1484

According to analogous statements, the corresponding concentration of B is 1.148 and of C is 0.852, in the same units implied in the initial conditions. The rate of a chemical reaction is proportional to concentrations of pertinent species whereas equilibrium properties depend on their activities; neglecting a distinction between concentrations and activities, we find that the equilibrium quotient, $K_{eq} = C_{eq} / (A_{eq} B_{eq})$, for this system in this approximation is simply the ratio, k_f / k_r, of the coefficients of rates of forward and reverse reactions. By altering the numerical values of the five parameters we can discern how the final concentrations of the three compounds A, B and C and the equilibrium quotient depend on the rate coefficients.

For innumerable additional examples, we respectfully suggest that an interested reader sample a particular comprehensive interactive electronic textbook (*6*) that has been designed, and tested in practice, for the teaching of mathematics to students of chemistry. Computer files that constitute this electronic textbook require operation with specific software for computer algebra, but the cost to a student of both the book and that software that operates on all common operating systems is comparable with the cost of a traditional printed textbook; that printed book might be prescribed for a particular course during only one semester, whereas the software, with or without the electronic book, is useful for general mathematical applications until the user hungers for a newer version! Of course software other than *Maple* is available, in some cases even free, on the basis of which one might compose an alternative electronic textbook, but so far such a realization is lacking. Another factor hampering the widespread application of software for symbolic computation has been the price of the software, but the price of software is in some cases decreasing; for example, versions of *Maple* are being bundled with textbooks for a modest cost.

We add a cautionary note about the teaching of mathematics in schools. We in no way advocate the replacement, during primary and secondary education, of acquiring substantial mental and manual skills in arithmetic and algebra by direct use of calculators and computers with their associated software, although such devices might enrich the teaching of pertinent topics through illustrative plots or otherwise. Furthermore, when the total duration of primary and secondary education involves study of arithmetic and mathematics for twelve years or less, we have grave reservations about the inclusion of calculus as a significant component of that curriculum; instead, in addition to algebra and introductory statistical topics, geometry, with appropriate trigonometry, in both formal and descriptive aspects should be emphasized as a preparation of every adult to appreciate the concepts of space and form, as a basis of understanding architecture and art. For the illustration of geometric operations and concepts, there exists commendable heuristic software that is applicable to a school context.

Conclusion

We must all agree not only that computers are here forever but also that they affect strongly the teaching and practice of mathematics, for chemistry students or otherwise, just like every other aspect of knowledge activity and communication; hence our student of chemistry who is deprived of a significant acquaintance with mathematical – not merely arithmetical – software is not being prepared properly for a technical career. Currently available mathematical software, nominally for symbolic computation but with associated numerical and graphical capabilities highly developed, provides an invaluable tool for both teaching and doing mathematics, and should become an integral component of routine instructional presentation. Instruction should emphasize mathematical concepts and principles, with numerical and graphical interpretations and illustrations, and indicate how mathematical operations are implemented, although there is no necessity to restrict implementation to a single software product. Just as each student practises manipulation of chemicals and instruments in the chemical laboratory, he or she should learn how to adopt an experimental and constructive approach to mathematics, based on mathematical software, rather than a sterile formal description according to theorems, corollaries, lemmas et cetera, for the chemist will be a user of mathematics not a developer of mathematics. As mathematical software continues to evolve, both instructors and their students must expect to expand their mathematical horizons, and to progress in their own development stimulated through that software. The future development of internet communication and its impact on education are difficult to predict – even a few years into the future, but what is certain is that both content and process of mathematical and chemical education are evolving rapidly as a consequence of the existence and deployment of digital computers and symbolic computation. Each instructor of both chemistry and mathematics has a solemn duty and responsibility to adapt to, and to work with, computers to prepare optimally his or her students for future technical careers. The future might be unpredictable in detail, but the trends are clear: computers and symbolic computation in the teaching and practice of chemistry and mathematics are indisputably part of them.

We must likewise accept that, just like the demise of manual calculation of square roots of numbers, the future practice – already well in progress – of mathematics by chemists will involve not manipulation of numbers and formulae by hand but mostly invocation of symbolic computation for such purposes. At this time it is only sensible to revise the teaching of mathematics for students of chemistry to reflect the realities of available computer hardware and software; with an interactive electronic textbook, software in the form of sophisticated processors for computer algebra and cognate operations makes such study of mathematics for chemistry eminently practicable.

Literature cited

1. Monagan, M. B.; Geddes, K.; Heal, K.; Labahn, G.; Vorkoetter, S.; McCarron, J.; DeMarco, P. *Maple 10 Introductory Programming Guide*. Maplesoft: Waterloo, Canada, 2005. ISBN 1-894511-76.

2. Wolfram, S. *The Mathematica Book*, Cambridge University Press: Cambridge, UK, 1996. ISBN 0-5221-58889-8.

3. Ogilvie, J. F. Teaching and doing mathematics with symbolic computation, in *Proceedings, Eighth Asian Technology Conference in Mathematics* 2003, 1, 16 – 22; http://www.atcminc.com

4. Monagan, M. B. 2D and 3D Graphical Routines for Teaching Linear Algebra, *Proceedings, 2002 Maple Summer Workshop*, Waterloo Maple Inc., 2002; available from http://www.cecm.sfu.ca/CAG/products2002.html

5. Simons, F. Computer algebra in service courses, *CTI Math & Stats Newsletter*, 8 (3), 1997 August, CTI Centres for Mathematics and Statistics, University of Birmingham, UK http://aweb.bham.ac.uk/ctimath/reviews/aug97/algebra.pdf

6. Ogilvie, J. F. with Doggett, G.; Fee, G. J.; and Monagan, M. B. *Mathematics for Chemistry with Symbolic Computation*, volume 1, Maplesoft, Waterloo, Canada, 2005; http://www.maplesoft.com/products/thirdparty/math_for_chemistry/index.aspx