CHEMICAL ENGINEERING WITH MAPLE[1]

ROSS TAYLOR, KATHERINE ATHERLEY[2]
Clarkson University
Potsdam, NY 13699-5705

Chemical engineering students are required (by accreditation agencies) to make appropriate use of computers throughout their program. Appropriate use is defined as including most of the following: programming in a high-level language; use of software packages for analysis and design; use of appropriate utilities; and simulation of engineering problems.

Maple is a powerful and flexible computing tool that has the potential of becoming the software package of choice for much scientific and engineering work, perhaps replacing, at least in part, other computer-based methods such as traditional programming languages and special purpose analysis and design programs. In this paper we provide a brief description of Maple and discuss some of the ways it can be used in the chemical engineering curriculum.

MAPLE

Maple[2] is a computer algebra system (CAS). Computer algebra is defined as follows:[3]

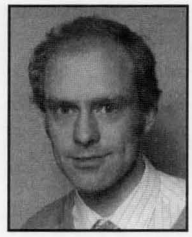
> *Computer algebra (sometimes called algebraic manipulation, or symbolic computation) can be defined to be computation with variables and constants according to the rules of algebra, analysis and other branches of mathematics, or formula manipulation involving symbols, unknowns, and formal operations rather than with conventional computer data of numbers and character strings.*

There are several computer algebra systems in use today. Macsyma, Reduce, Derive, Mathematica, Maple, and Scratchpad (now known as Axiom) are some of the better-known ones (Gonnet and Gruntz[3] give brief histories). This article focuses on Maple because it happens to be the CAS we use, although much of what follows would also be true of other systems.

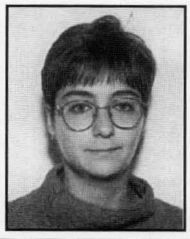
The emphasis in the above definition on symbolic manipulation should not be taken to imply that Maple is unsuitable for the numerical calculations that dominate engineering computing today. Maple combines symbolic mathematical capabilities (integration and differentiation, for example) with numerical capabilities (integration of ODE systems and sparse linear equation solving) and sophisticated graphics (including three-dimensional plots of surfaces and spacecurves, sparse matrix plots, and much more) that allow new approaches to the teaching of traditional materials.

Maple provides a command-line-in-a-window style of interface. Its commands can be entered at the prompt and executed immediately. This allows more immediate and easier experimentation and exploration in what are called worksheets. Worksheets can be scrolled forward and backward in order to review prior results; in fact, it is possible to go back, change and re-execute just one or a few Maple instructions without having to re-execute the entire worksheet.

Maple runs on a wide variety of computer platforms. The worksheets created by one version of Maple are saved as text files and can be imported with little difficulty. The interfaces to different versions of Maple are not the same, however. In some cases this is more or less inevitable (as, for example, between the MS DOS and MS Windows versions). In other cases (such as between the AIX, MS Windows, and Macintosh versions), it is less easy to understand the differences. The interfaces to all of the above mentioned versions possess interesting features that should be available in all versions.



Ross Taylor *is a professor of Chemical Engineering at Clarkson University. His interests are in the areas of mass transfer and separation processes, and he is the coauthor (with Professor R. Krishna, University of Amsterdam) of* Multicomponent Mass Transfer *(Wiley, 1993) and (with Harry Kooijman) of* ChemSep, *a software package for separation process simulation used in universities in several countries.*



Katherine Atherley *has a degree in pure mathematics from the University of Waterloo where she first encountered Maple. She worked with the Maple research group before joining WMS, and has been involved with many aspects of Maple, from programming and writing documentation to providing user support and teaching Maple courses. She is currently Product Manager at InContext Corporation.*

[1] *This paper is based on a presentation given at the 1994 Maple Summer Workshop. An abbreviated version appears in the proceedings of that event*[1]

[2] *Address: InContext Corp., Toronto, Ontario, Canada*

Maple: A Personal Perspective

My first exposure to Maple came some years ago when I considered using it for solving the examples in a textbook I was writing. At that time Maple was not able to do many of the things we needed. More recently I took another look at Maple when the engineering school at Clarkson was considering adopting it (see box *Maple at Clarkson*). To teach myself Maple, I did not go through the tutorials; instead, I picked problems from the literature in chemical engineering or which were somehow related to my research interests and tried to see if I could get anywhere with Maple. Finding the critical constants to a generic cubic equation of state with the help of Mike Monagan was the first problem I looked at.[6]

I have done (and continue to do) a great deal of programming in Fortran, and at first I found it difficult to get used to Maple. I believe that part of my difficulties with Maple had to do with the fact that I had to learn a new way of thinking when writing Maple. Despite (or perhaps because of) these difficulties, I became addicted to Maple in a very short time, and now it is the tool I turn to first.

By manipulating equations with Maple I have gained new insights (at least for me) into methods of solving certain problems. Now that I am past the initial hurdles, I am able to solve real engineering problems with much less Maple code (and in much less time) than would have been the case had I used Fortran. Moreover, I can do things with Maple that are not even possible with Fortran. For example, after I had solved a binary distillation problem with Maple, it took me only fifteen minutes to program Maple to construct the McCabe-Thiele diagram shown in Figure 4.

I expect (hope) that students who do not have any prior experience programming in Fortran (or other traditional language) will not face the same degree of difficulty that I experienced.

I use the MS Windows and AIX versions of Maple in my own work and have shared files with others using Maple on Sun and Next workstations and on Macintoshes (and possibly others for all I know).

Ross Taylor

In the next few sections we focus on a few ways in which Maple can be used in selected chemical engineering courses.

Computer Programming

Undergraduate engineering students are required to do some programming in a high-level language. Often the programming course is the first course to expose students to elements of engineering problem solving. Most students learn FORTRAN, and C or Basic are included in some curricula. With more and more students learning to use Maple in the calculus classes (and fewer and fewer of them using any kind of traditional programming language after they graduate), it makes sense to consider adopting Maple as the programming language for use in their engineering courses (see the box *Maple at Clarkson*).

Maple is built around a programming language that is custom-designed for symbolic mathematical calculations and manipulations. Unlike FORTRAN, the Maple language supports standard mathematical structures as data-types and can work with them in sensible mathematical ways. We feel that the language is as suitable as any of the traditional languages for emphasizing those problem-solving skills that are acquired by learning a programming language. One might even make the case that Maple is better suited; Maple code is more natural for mathematical work and produces much shorter, easier to understand, programs than FORTRAN.

Material and Energy Balances

Maple allows one to solve elementary material balance problems in a systematic way that makes it almost impossible to get the problem formulation wrong. Just a few lines of Maple code can set up the material balances and mole fraction summation equations for any process unit regardless of the number of components and input and output streams. Systems with chemical reactions also can be modeled.

Maple's ability to handle symbolic indices makes it possible to identify components with a number, name, chemical formula, or any other convenient label. The number of unknown variables and independent equations can be quickly counted and, hence, the number of degrees of freedom determined. Specification equations can be added to the model equations and Maple asked to solve the entire set of equations in one go.

Problems often can be solved symbolically in terms of an unspecified parameter (reactor conversion, say). This is useful if it is desirable to evaluate the solution at several parameter values. It is unnecessary even to choose a basis as the actual specification of interest can be included among the set of equations.

Thermodynamics

Calculation of the critical constants for a cubic equation of state is a classic problem in thermodynamics, one that is covered in most thermodynamics textbooks.[4,5] Textbook examples usually include finding the critical constants for the simplest cubic equation of state, that of Van der Waals or those of the Redlich-Kwong family. With Maple, however, it is possible to obtain explicit expressions for the critical constants for a generic cubic equation of state.[6] Constants for particular equations of state can be obtained as special cases of the general result.

Maple is a useful tool for the visualization of thermodynamic functions. Figure 1 shows the roots of the compressibility polynomial for the Soave-Redlich-Kwong equation as a function of reduced temperature at a reduced pressure of 0.75. The well-known fact that a cubic equation can have complex roots at certain parameter values is illustrated here. This figure was obtained with about thirty

lines of Maple code, which included the calculation and ordering of the roots themselves. Figure 2 provides a three-dimensional view of a compressibility diagram; only one line of code (over and above that used for Figure 1) was needed to create Figure 2. Three-dimensional diagrams in Maple can also be rotated and viewed from other angles.

A Maple procedure to obtain *expressions* for the activity coefficients from any model of the excess Gibbs energy and for a specified number of components can be written in about ten lines of Maple code. Students can investigate different models of the Gibbs energy function without running the risk of getting the derivation incorrect.

Simple phase equilibrium calculations and the creation of phase diagrams for binary systems using Maple have been discussed by Taylor.[7]

Separation Processes

Multicomponent distillation simulations require the numerical solution of a large set of equations: material balances, energy balances, and equilibrium (thermodynamics) equations. These equations are sparse, nonlinear, and can easily number in the hundreds and sometimes in the thousands. The literature on distillation contains scores of papers discussing methods of solving these equations (see Seader[8]). Issues such as what form the equations should take, what variables should be used, in what way should the equations and variables be ordered, what variables should be computed from what equations, and what numerical methods should be used to solve (each subset of) the equations have been thoroughly explored. Figure 3 shows an incidence matrix (sparsity pattern) for a column distilling a nonideal binary mixture in ten stages. It is pertinent to point out that although with Maple it is simple to explore different computational strategies, Maple also makes topics like this largely irrelevant (even if they are interesting).

A Maple session to obtain the flows inside a distillation column under the assumptions of constant molar overflow is shown in the Appendix. It is a simple matter to investigate alternative operating strategies such as what happens to the flows when the feed is a saturated vapor or partially vaporized liquid.

Four pages of Maple code is all that is required to obtain numerical solutions to many multicomponent distillation problems (including the *derivation* of *all* the equations). This compares to the many hundreds (or even thousands) of lines that would be needed to solve the same problem using FORTRAN. Interlinked columns and nonstandard specifications also are simple to deal with (and only slightly more difficult to solve). Figure 4 shows the McCabe-Thiele diagram plotted from the results obtained by numerically solving the equations that gave us Figure 3.

Chemical Reaction Engineering

Chemical reaction engineering problems often require solution of systems of coupled differential equations. Textbook problems sometimes are specially simplified so that the equations can be solved analytically. While such solutions can also be obtained with Maple, it is no longer necessary to simplify problems in this way. Software packages that possess numerical methods for solving ODEs can be used to solve more realistic problems. This point of view has already been expressed by Fogler,[9] who uses Mathematica and Polymath for solving reaction engineering problems. The advantages of a CAS over a purely numerical method of

Maple at Clarkson

In common with many other schools, undergraduate engineering students at Clarkson University are required to take a programming course. Until this year, the language taught in this course was Fortran (although some students received instruction in Basic). No longer will Clarkson students take courses in Fortran (unless it be by choice); the introductory computing course has been completely revised, and Clarkson students will now be programming using the computer algebra system Maple.

The decision to abandon Fortran in favor of Maple was not reached lightly or without considerable debate within the engineering school at Clarkson. Factors in favor of Maple included the fact that our students have been using Maple in their calculus classes for the past few years. It was deemed to be unwise to introduce first-year students to a second major software package such as MATLAB. In addition, our site license permits us to install Maple on nearly all machines owned by Clarkson, including the machines that all of our students (regardless of discipline) are issued when they arrive on campus. Clarkson's PC program has been in place for over a decade, and in the fall 1993 semester new students were issued a computer with Maple already installed on the hard drive.

The fact that Maple can exchange files across platforms is a great advantage for a computationally diverse campus like Clarkson; in addition to the huge number of PCs (over three thousand), there are nearly two hundred IBM RS/6000 workstations (most of which are grouped in instructional laboratories) and more than a few Suns and DECs.

The 1994 graduating class was the last one to have had no formal instruction in Maple. Nevertheless, several chemical engineering students had found opportunities to use Maple on occasions and more than a few of them used Maple to solve problems in our Design II course where, as it happened, the plant could not be simulated using a more conventional process flowsheeting program.

The reaction of the first-year students to the first course on using Maple as a programming language was more positive than any time in the past several years when Fortran was introduced to them. Faculty members in the engineering departments also are learning Maple with a view to incorporating aspects of problem solving with Maple into selected engineering courses. It will be interesting to follow the progress of both students and faculty as Maple finds increased use in the engineering school.

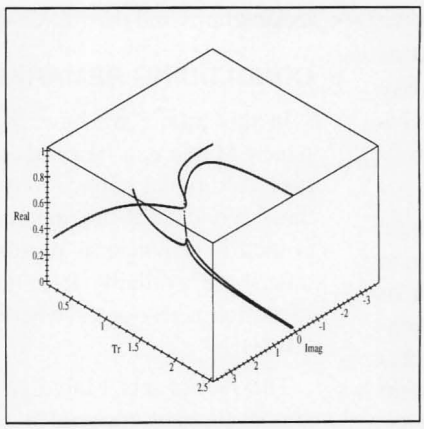


Figure 1. *Compressibility as a function of reduced temperature (T_r) at a reduced pressure of 0.75 computed using the Soave-Redlich-Kwong cubic equation of state. The vertical axis is the real part of the compressibility; the horizontal axis at the back of the figure is the imaginary part of the compressibility. All axes are dimensionless. The region of three real roots is shown between a region possessing a real liquid-like root and complex vapor-like roots and another region with a real vapor-like root and two complex liquid-like roots.*

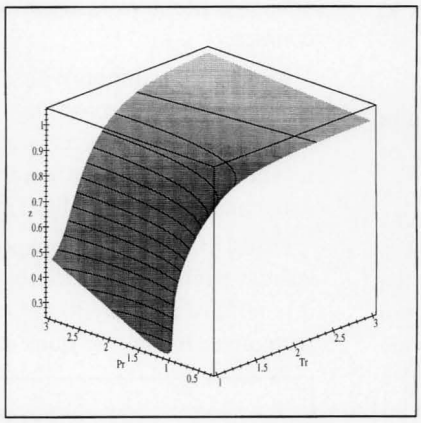


Figure 2. *Compressibility surface as a function of reduced temperature and pressure. The vapor-like root from Figure 1 is used in the construction of this figure.*

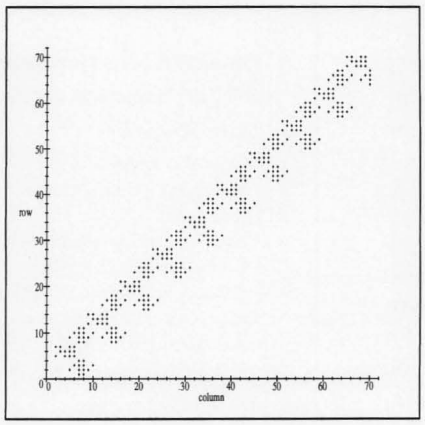


Figure 3. *Incidence matrix for a small distillation problem (2 components, 10 stages). The equations and variables are grouped by stage leading to the familiar block tridiagonal pattern shown in the figure; however, it only takes two lines of Maple to reorder the equations so they are grouped by type rather than by stage. The pattern is "upside-down" because Maple plots cannot (yet) go from high to low on any axis.*

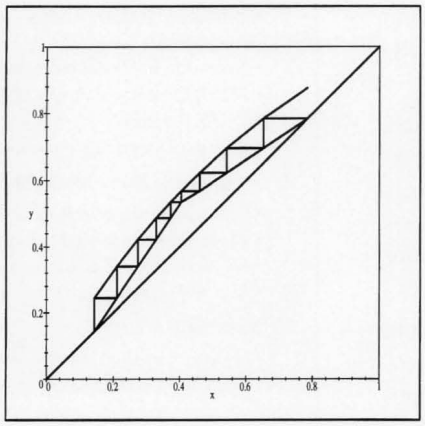


Figure 4. *McCabe-Thiele diagram constructed from the solution obtained with Maple to a binary distillation problem for a column with 10 stages (including a total condenser). About 10 lines of Maple code were used to create this figure. The various parts of the diagram are created separately (in different colors if so desired) and then combined into a single figure using a Maple command designed for this purpose.*

solution include the fact that the reaction system can be analyzed symbolically and the material and energy balance equations also may be derived.

Figure 5 shows multiple steady states in a nonisothermal continuous stirred tank reactor (CSTR). This plot was created using the parameters given by Shacham, *et al.*[10] who also considered the CSTR dynamics and noted that the stability of each steady state could be determined by computing the eigenvalues of the state matrix using Polymath. All of these things are possible with Maple; however, Maple is capable of evaluating the eigenvalues of the state matrix symbolically as well!

Process Design

We do not think it requires too great a leap of imagination to expect students to set up and solve entire process flowsheeting problems using Maple. The techniques that are useful for material and energy balances around simple units are readily applied to process flowsheets with any number of units and their interconnections. Flowsheets containing recycle streams are easy to handle.

While Maple will not replace (nor should it) specialized programs designed for large-scale plant simulation, it is a useful tool for teaching students how flowsheeting simulations work. The Maple programming language encourages problems to be formulated in a way that is reminiscent of equation-oriented flowsheeting,[11] but it is not too difficult to instruct Maple to solve flowsheet problems using tearing or simultaneous modular strategies. Maple's open interface and powerful language make it possible for engineers to create their own unit models in the Maple programming language; others would then be able to use them as plug-in modules in their own problems if they were made available.

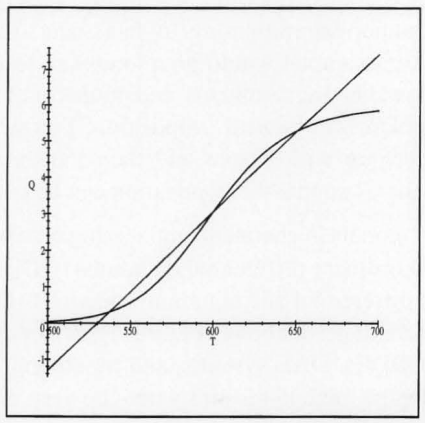


Figure 5. *Multiple steady states in a nonisothermal continuous stirred tank reactor (CSTR). The vertical axis is the heat duty (Btu/hr x 10^{-5}), the x-axis is the reactor temperature (K). The straight line represents the heat lost due to cooling the reactor. The S-curve is the heat generated by the reaction.*

More Maple

There are, of course, a number of ways that Maple needs to be improved. It needs the ability to read and write binary direct-access files (as is possible in FORTRAN and C). This would make it possible to access databanks of physical property data for direct use in engineering calculations.

Maple also needs improved symbolic capabilities. There are many engineering formulas where it is necessary to differentiate arbitrary sums and products (*e.g.,* a sum or product of indexed variables where the index range is nonnumeric, such as i = 1..c). The fact that Maple cannot do this is a serious impediment to using it for certain important problems (such as the derivation of thermodynamic properties of mixtures). We also need to be able to (elegantly) exclude selected elements from sums and products of indexed variables.

A great many problems in chemical engineering require finding numerical solutions to large (or small) systems of (sparse) nonlinear equations. This is not one of Maple's strengths. The floating point solver built into Maple lacks some of the features that would be useful. In particular, it is not possible to provide the initial estimates or to control the iteration history. On the other hand, it is possible to program Newton's method in Maple so that all the user must provide is a set of equations, a list of unknown variables, and a starting point; Maple can compute the Jacobian symbolically, thereby removing one of the major chores that must be faced when using the method as a part of a FORTRAN program. In fact, Maple is used for precisely this purpose in some companies that write software for process engineering simulations. Differential arc-length homotopy continuation methods, recommended by Seader[8] for solving difficult nonlinear problems, may also be easily (and elegantly) programmed in the Maple programming language.

Unfortunately, Maple is currently many times slower than compiled FORTRAN when carrying out large-scale numerical computations. It needs better (*i.e.,* faster) routines for purely numerical computing. A fast sparse linear equation solver, for example, would go a long way to making large-scale flowsheeting problems and multicomponent distillation problems a practical proposition. For now, a problem can be set up with Maple and then translated into FORTRAN (or C) so that the application can be compiled.

Many models in chemical engineering consist of large sets of (stiff) ordinary differential equations (ODEs), mixed systems of differential and algebraic equations (DAEs), or partial differential equations (PDEs). Fast numerical methods for stiff ODEs, DAE systems, and for solving PDEs by, for example, the method of lines would be very welcome.

Maple's graphics capabilities, although quite good, could be improved by adding more basic plot types that are encountered often in (chemical) engineering (such as triangular diagrams and their three-dimensional counterparts).

CONCLUDING REMARKS

In this article we have highlighted only a few ways in which Maple can be used in chemical engineering education. Additional applications are listed in Table 1. Some of these worksheets are included in the Maple share library (which is provided as part of Release 3 of Maple V), and others are available from the first author. We have also identified a few areas where Maple needs improved capabilities.

The fact is that Maple can have a significant impact in almost all areas of chemical engineering education, but there are some problems associated with using Maple in existing courses.

Maple (in common with all other computer algebra systems) contains far more bugs than does even the worst FORTRAN compiler. This is largely a reflection of the relatively recent development of computer algebra and the fact that some mathematics are hard to do with a computer.

Using Maple to derive expressions that are standard fare in current engineering textbooks will rapidly demonstrate that it is not always easy to get from Maple an answer that you recognize. It may be quite trivial for Maple to solve your

TABLE 1
Maple Worksheets in Chemical Engineering

1. ***Chemical Process Calculations***
   - Material balances on single and multiple process units
2. ***Thermodynamics***
   - Critical constants for cubic equations of state[6]
   - Phase equilibrium calculations and phase diagrams for ideal systems[7]
   - Activity coefficients in binary and multicomponent systems
   - Gibbs free energy surfaces
   - Phase equilibrium calculations for nonideal systems
   - Flash calculations for ideal systems
   - Advanced flash calculations
   - Thermodynamic property relations and the Maxwell equations[14]
3. ***Reactor Engineering***
   - Material balances in tubular reactors
   - Isothermal tubular reactor (multiple reactions, numerical integration)
   - Nonisothermal tubular reactor
   - Multiple steady states in a CSTR
   - CSTR dynamics
   - Fitting reaction rate coefficients to rate data
4. ***Equilibrium Stage Separations***
   - Constant molar overflow in distillation
   - Multicomponent distillation—stage-to-stage calculations
   - Multicomponent distillation—simultaneous solution
   - McCabe-Thiele diagrams
5. ***Numerical Methods***
   - Newton's method for systems of equations
   - Homotopy-continuation for systems of equations

problem correctly, but it can require considerable skill in expression manipulation in order to get a familiar result. While this may not be important in solving original problems, it can make matching the results in established textbooks a frustrating experience. Perhaps we will have to get used to new ways of looking at old results. The problem of simplifying the chore of obtaining recognizable results remains as a challenge for the computer-algebra community.

The exercises and examples in many standard textbooks, (*e.g.,* Felder and Rousseau[12] and Reklaitis[13]) were designed to be solved by hand. Many (if not most) of these problems are far too simple if Maple is on hand to assist with the problem solving; when you have solved one, you have solved them all.

This brings us to some important questions: Do we want students to use Maple for solving engineering problems? Can the use of a CAS prevent students from mastering essential skills that are better assimilated when solving problems by hand? Computer algebra is finding increasing use in the teaching of calculus at many schools. It is impossible to turn back the clock and abandon the use of computer algebra in mathematics courses, thereby making its use in engineering a nonissue. It will not be possible to prevent students from using tools they have learned once it has become clear that they are useful. It will be up to us as educators to find the proper time and place in our courses to introduce students to engineering problem solving with Maple.

REFERENCES

1. Taylor, R., and K. Atherley, "Chemical Engineering with Maple," in *Maple V: Mathematics and Its Application,* R. Lopez (Ed.), Birkhauser, Boston, MA (1994)
2. Char, B.W., K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, and S.M. Watt, *Maple V Language Reference Manual,* Springer-Verlag (1991)
3. Gonnet, G.H., and D.W. Gruntz, Algebraic Manipulation Systems, in *Encyclopedia of Computer Science and Engineering,* 3rd Ed., Van Nostrand Reinhold (1991)
4. Walas, S.M., *Phase Equilibria in Chemical Engineering,* Butterworths, Stoneham, MA (1985)
5. Sandler, S.I., *Chemical and Engineering Thermodynamics,* 2nd ed., Wiley, New York, NY (1989)
6. Taylor, R., and M.B. Monagan, "Thermodynamics with Maple. I. Equations of State," *Maple Tech,* **10,** 50 (1993)
7. Taylor, R., "Thermodynamics with Maple V. II. Phase Equilibria in Binary Systems," *Maple Tech,* **1**(1), 83 (1994)
8. Seader, J.D., "The BC and AD of Equilibrium-Stage Operations," *Chem. Eng. Ed.,* **19**(2), 88 (1985)
9. Fogler, H.S., *Elements of Chemical Reaction Engineering,* 2nd ed., Prentice-Hall, Englewood Cliffs, NJ (1993)
10. Shacham, M., N. Brauner, and M.B. Cutlip, "Exothermic CSTRs: Just How Stable are Those Steady States," *Chem. Eng. Ed.,* **28**(1), 30 (1994)
11. Westerberg, A.W., H.P. Hutchison, R.L. Motard, and P. Winter, *Process Flowsheeting,* Cambridge University Press, Cambridge, UK (1979)
12. Felder, R.M., and R.W. Rousseau, *Elementary Principles of Chemical Processes,* 2nd ed., Wiley, New York, NY (1986)
13. Reklaitis, G.V., *Introduction to Material and Energy Balances,* McGraw-Hill, New York, NY (1983)
14. Adams, S., and R. Taylor, "Thermodynamics with Maple. III. Thermodynamic Property Relations and the Maxwell Equations," *Maple Tech,* In press (1995)

APPENDIX
EXAMPLE SESSION USING MAPLE

Maple session to compute flows inside a distilation column
Note that Maple input can be terminated by a ; or by :. In the latter case the result is not echoed to the screen

```
>restart:
>ns:=10: # Define the number of stages (1 is the condenser, ns the reboiler)
```
Material balances
```
>for j from 2 to ns-1 do TMB[j]:=V[j]+L[j]-V[j+1]-L[j-1]=0; od:
>TMB[1]:=D+L[1]-V[2]=0; #Material balance for the condenser
```
$$TMB_1 := D + L_1 - V_2 = 0$$
```
>L[ns]:=B: # The liquid flow leaving the reboiler is given the symbol B
>TMB[ns]:=V[ns]+L[ns]-L[ns-1]=0; # Material balance for the reboiler
```
$$TMB_{10} := V_{10} + B - L_9 = 0$$
Energy balances
```
>for j from 2 to ns-1 do EB[j]:=V[j]*H[V,j]+L[j]*H[L,j]-V[j+1]*H[V,j+1]
 -L[j-1]*H[L,j-1]=0; od:
>FeedStage:=5: # Put a feed on stage 5
 j:=FeedStage: # Modify the balances for the feed stage
 TMB[j]:=V[j]+L[j]-V[j+1]-L[j-1]-F=0;
 EB[j]:=V[j]*H[V,j]+L[j]*H[L,j]-V[j+1]*H[V,j+1]-L[j-1]*H[L,j-1]-F*H[F]=0
```
$$TMB_5 := V_5 + L_5 - V_6 - L_4 - F = 0$$
$$EB_5 := V_5 H_{V,5} + L_5 H_{L,5} - V_6 H_{V,6} - L_4 H_{L,4} - F H_F = 0$$
```
>for j from 1 to ns do H[V,j]:=H[V]; H[L,j]:=H[L]; od: # Make enthalpies of
 each phase the same
>H[F]:=H[L]: # Assume feed is saturated liquid
```
The overall material balance for the column can be obtained by summing the material balances for all stages
```
>j:='j': sum(TMB[j],j=1..ns): TCB:=op(solve({''},{F}));
```
$$TCB := F = D + B$$
```
>RRdef:=R=L[1]/D: RRdef; #Define the reflux ratio
```
$$R = L_1/D$$
Now we solve the material and energy balances for the flows. We create a set of equations which includes the material balances for stages 2 to ns, the energy balances for stages 2 to ns-1 and the definition of the reflux ratio, R. Then we create a set of unknown variables that we wish to compute (the flows) and invoke Maple's solve command.
```
>Eqns:={seq(TMB[i],i=2..ns),seq(EB[i],i=2..ns-1),RRdef}:
>Vars:={seq(V[i],i=2..ns),seq(L[i],i=1..ns-1)}:
>result:=solve(Eqns,Vars): #output hidden to save space

>subs(TCB,result): # Eliminate F using the overall column balance
>collect('',D): # Tidy up by collecting terms in D
>assign('');
>Flows :=linalg[matrix](ns,2): # Create an array to hold the flows
>V[1]:='': #Hide V[1] since there is no vapor flow from the condenser
>for j from 1 to ns do Flows[j,1] := V[j]; Flows[j,2] := L[j] od:
>print(Flows);
```
$$\begin{pmatrix} & RD \\ (1+R)D & RD \\ (1+R)D & RD \\ (1+R)D & RD \\ (1+R)D & (1+R)D+B \\ (1+R)D & (1+R)D+B \\ (1+R)D & (1+R)D+B \\ (1+R)D & (1+R)D+B \\ (1+R)D & (1+R)D+B \\ (1+R)D & B \end{pmatrix}$$
The vapor flows are in the left hand column, the liquid flows in the right
```
>
```
❐