
Integrating Differential Equations Using Excel 7.0

Edward M. Rosen, EMR Technology Group

The need to solve sets of ordinary differential equations arises in a number of chemical engineering applications (1). The ability to carry out these calculations within a spreadsheet environment (2, 3, 4) has a number of advantages:

1. The spreadsheet is a well known and commonly used environment for carrying out chemical engineering calculations.
2. The input and output capabilities of the spreadsheet can be utilized.
3. No additional programs or systems need to be learned.

What is required, however, is the use of Visual Basic for Applications (VBA) which is supplied as an integral part of Excel 7.0 (5).

Ordinary Differential Equations- Initial Value Problems

There are a number of algorithms that may be used to solve ordinary differential equations with specified initial conditions. For problems which are not stiff the 4th order Runge Kutta method is very popular. A step size (h) must be specified.

For Each Equation(j)

$$Y_{i+1,j} = y_{ij} + [1/6(k_{1j} + 2k_{2j} + 2k_{3j} + k_{4j})]*h$$

Where:

$$k_{1j} = f_j(x_i, y_{i1}, y_{i2}, y_{i3}, y_{i4}, y_{i5})$$

$$k_{2j} = f_j(x_i + h/2, y_{i1} + 1/2*h*k_{1,1}, y_{i2} + 1/2*h*k_{1,2}, \dots)$$

$$k_{3j} = f_j(x_i + h/2, y_{i1} + 1/2*h*k_{2,1}, y_{i2} + 1/2*h*k_{2,2}, \dots)$$

$$k_{4j} = f_j(x_i + h, y_{i1} + h*k_{3,1}, y_{i2} + h*k_{3,2} + \dots)$$

Figure 1 is a VBA implementation of the 4th Order Runge Kutta algorithm for up to five simultaneous differential equations. The *array function(2) Rk4x* has the following inputs:

1. The stepsize, h
2. The current value of the independent variable, x
3. The current values of the dependent variables.
4. A parameter vector, prm.

The output of the function is the array DDD(1) to DDD(7). It consists of

1. The new stepsize (can be the same value as the current stepsize).
2. The new value of the independent variable.
3. The new values of the dependent variables.

The **Rk4x** function is implemented on the spreadsheet by entering

$$=Rk4x(h, x, y1, y2, y3, y4, y5, prm)$$

where

h = step size

x = independent variable

y1..y5 = dependent variables

prm = a parameter vector of any length

A separate function (**Rhs**) must be written by the user to specify the right hand sides of the equations.

An Example Problem

Three tanks (6) are set up so that the flow out of the top tank into tank 1 is a constant. The flow through a valve from tank 1 into tank 2 depends on the height of the liquid and the flow from tank 2 through a valve also depends on the height of the liquid in that tank. (see Figure 2).

The following equations apply:

$$q_1 = C_1 \sqrt{h_1}$$

$$q_2 = C_2 \sqrt{h_2}$$

$$A * dh_1/dt = -C_1 * \sqrt{h_1} + q_{in}$$

$$A * dh_2/dt = C_1 * \sqrt{h_1} - C_2 * \sqrt{h_2}$$

If a volume of liquid equal to M is added to the second tank, the problem is to show the response of the height vs time in the two tanks. Use M = 400 cm³.

Steady state measurements show:

$$A = 48.65 \text{ cm}^2$$

$$h_{1s} = 24.5 \text{ cm}$$

$$h_{2s} = 16.6 \text{ cm}$$

$$q_{ins} = 94.47$$

By setting the derivatives equal to 0 (steady state) the values of C_1 and C_2 may be determined. ($C_1 = 19.085$, $C_2 = 23.1867$)

Figure 3 is a listing of the function **Rhs** which defines the right hand sides of the two differential equations. **Rhs** is called by **Rk4x**.

Figure 4 is a listing of the spreadsheet. Row 15 contains the labels for each of the first four columns. Row 16 specifies the integration step size, the total time elapsed and the initial heights of the tanks. The following actions are then taken:

1. Columns A17 to D17 must first be selected (highlighted) with the mouse. The following is entered into the formula bar:

=Rk4x(\$A16,\$B16,\$C16,\$D16,0,0,0,\$B\$4)

2. **Cntrl+Shift+Enter** is then entered.

The new values (at the new time) of the step size, time and dependent variables will then appear in row 17. Row 17 may be copied as many times as desired out to a final time.

Figure 5 is a plot of the results shown in the spreadsheet of Figure 4.

Summary

The integration of sets of differential equations may be performed rapidly and easily using Excel 7.0 and Visual Basic for Applications. The **Rk4x** integration routine is part of the EMR Technology Group Library and may be ordered from the CACHE Corporation.

References

1. Rosen, E., M. and R. N. Adams, A Review of Spreadsheet Usage in Chemical Engineering Calculations, *Computers and Chemical Engineering*, **11**, 723 (1987).
2. Rosen, E.M, Visual Basic for Applications, Add-Ins and Excel 7.0. *CACHE News*, No. 45, Fall 1997.
3. Rosen, E.M., The Case for Excel and Visual Basic for Applications. *CACHE News*, No. 46, Spring 1998.
4. Rosen, E.M, Executing FORTRAN Programs From Excel: Use of the Shell Function. *CACHE News*, No. 47, Fall 1998.
5. Harris, M., Teach Yourself Visual Basic for Applications in 21 days. *Sams Publications*, Indianapolis (1994).
6. Palanki, S. and V. Sampath, A Simple Process Dynamics Experiment. *Chemical Engineering Education*, **31**, No 1 Winter 1997.

Public Function Rk4x(h,x,y1, y2, y3, y4, y5, prm)

'Written by EMRosen 10/30/97
'Copyright © EMR Technology Group

'h = step size
'x = independent variable
'y1, y2, y3, y4, y5 = dependent variables

'prm a parameter vector of unspecified length
'prm(1) = Number of Equations Being Integrated

'kij : i is the k value, j is the equation number of dependent variable

'Output is the Seven Element Vector
'Actual Output must be pre-selected

Dim DDD(1 To 7)

Dim fff(1 To 5)

Neq = prm(1)

Rtn1 = Rhs(x,y1,y2,y3,y4,y5,prm,fff)

k11 = fff(1)

k12 = fff(2)

k13 = fff(3)

k14 = fff(4)

k15 = fff(5)

Rtrn2 = Rhs(x + 0.5 * h,y1 + 0.5*h*k11,y2 + 0.5*h*k12,y3 + 0.5* h*k13,y4 + 0.5*h*k14,y5 + 0.5* h* k15,prm,fff)

k21 = fff(1)

k22 = fff(2)

k23 = fff(3)

k24 = fff(4)

k25 = fff(5)

Rtrn3 = Rhs(x + 0.5 * h,y1 + 0.5*h*k21,y2 + 0.5*h*k22,y3 + 0.5* h*k23,y4 + 0.5*h*k24,y5 + 0.5* h *k25,prm,fff)

k31 = fff(1)

k32 = fff(2)

k33 = fff(3)

k34 = fff(4)

k35 = fff(5)

Rtrn4 = Rhs(x + h,y1 + h*k31,y2 + h*k32,y3 + h*k33,y4 + h*k34,y5 + h*k35,prm,fff)

k41 = fff(1)

k42 = fff(2)

k43 = fff(3)

k44 = fff(4)

k45 = fff(5)

Values of new h and new x

$$\begin{aligned} \text{DDD}(1) &= h \\ \text{DDD}(2) &= x + h \end{aligned}$$

First Independent Variable

$$\text{DDD}(3) = y1 + (h / 6) * (k11 + 2 * k21 + 2 * k31 + k41)$$

Second Independent Variable

$$\text{DDD}(4) = y2 + (h / 6) * (k12 + 2 * k22 + 2 * k32 + k42)$$

Third Independent Variable

$$\text{DDD}(5) = y3 + (h / 6) * (k13 + 2 * k23 + 2 * k33 + k43)$$

Fourth Independent variable

$$\text{DDD}(6) = y4 + (h / 6) * (k14 + 2 * k24 + 2 * k34 + k44)$$

Fifth Independent variable

$$\text{DDD}(7) = y5 + (h / 6) * (k15 + 2 * k25 + 2 * k35 + k45)$$

Rk4x = DDD

End Function

Figure 1. Listing of Rk4x Function

Tank Emptying Model – Chem Eng Education Vol 31 No 1
Winter 1997

Page 64

prm array

- prm(1) = 2
- prm(2) – A = 48.65
- prm(3) – qins = 94.47
- prm(4) – h1s = 24.5
- prm(5) – h2s = 16.6

- prm(6) – C1 = qins/Sqr(h1s)
- prm(7) – C2 = C1 * Sqr(h1s/h2s)

Public Function Rhs(x,y1,y2,y3,y4,y5, prm, fff)

$$\begin{aligned} \text{fff}(1) &= \text{prm}(3) / \text{prm}(2) - \text{prm}(6) * \text{Sqr}(y1) / \text{prm}(2) \\ \text{fff}(2) &= \text{prm}(6) / \text{prm}(2) * \text{Sqr}(y1) - \text{prm}(7) / \text{prm}(2) * \text{Sqr}(y2) \end{aligned}$$

$$\begin{aligned} \text{fff}(3) &= 0 \\ \text{fff}(4) &= 0 \\ \text{fff}(5) &= 0 \end{aligned}$$

Rhs = 0

End Function

Figure 3. Listing of RHS Function

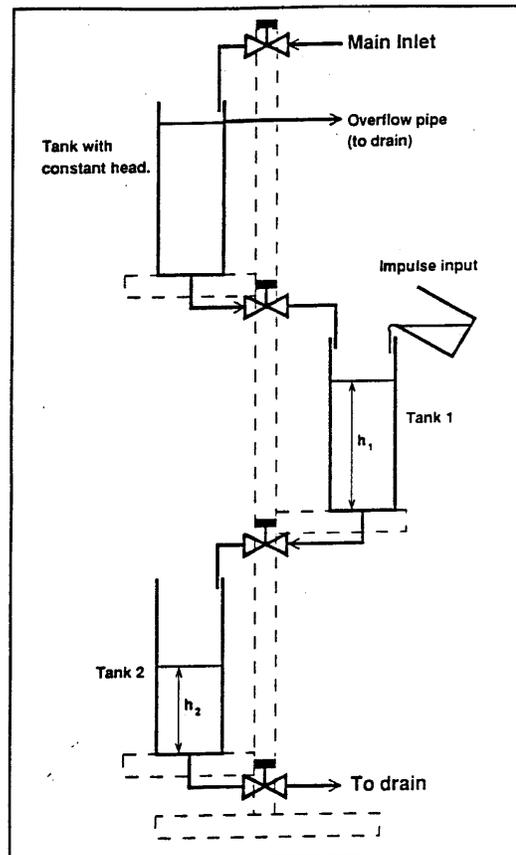


Figure 2. Configuration of Two Tanks

Parameters	
No of Equations	2
A	48.65
qins	94.47
h1s	24.5
h2s	16.6
C1	19.08582218
C2	23.18674957
M	400

Step Size	Time(Sec)	Tank 1 (cm)	Tank 2 (cm)	Row 15
4	0	32.72199383	16.6	Row 16
4	4	31.59251323	17.60552549	Row 17
4	8	30.61014698	18.27507339	
4	12	29.75763201	18.69452998	
4	16	29.01929699	18.92870841	
4	20	28.38101284	19.02716033	
4	24	27.8301214	19.0279769	
4	28	27.35534762	18.96043795	
4	32	26.94670087	18.84695034	
4	36	26.59537025	18.70451896	
4	40	26.29361793	18.54589244	
4	44	26.03467378	18.38047084	
4	48	25.81263394	18.21503234	
4	52	25.62236487	18.05431734	
4	56	25.45941422	17.9014983	
4	60	25.31992901	17.75855627	
4	64	25.20058137	17.62658102	
4	68	25.09850182	17.50600839	
4	72	25.01121971	17.39680619	
4	76	24.93661045	17.29861832	
4	80	24.87284903	17.21087492	
4	84	24.81836923	17.13287547	
4	88	24.77182801	17.06385032	
4	92	24.73207449	17.00300536	
4	96	24.69812306	16.94955348	
4	100	24.66913004	16.90273607	
4	104	24.6443736	16.86183687	
4	108	24.62323635	16.82619015	
4	112	24.60519042	16.79518478	
4	116	24.58978461	16.76826528	
4	120	24.57663331	16.74493091	
4	124	24.56540709	16.72473333	
4	128	24.55582449	16.7-727335	
4	132	24.54764512	16.69219736	
4	136	24.5406637	16.67919333	
4	140	24.5347049	16.66798699	
4	144	24.52961903	16.65833794	
4	148	24.52527828	16.6500361	
4	152	24.52157354	16.64289828	
4	156	24.51841166	16.63676511	
4	160	24.51571312	16.63149816	

Figure 4. Spreadsheet

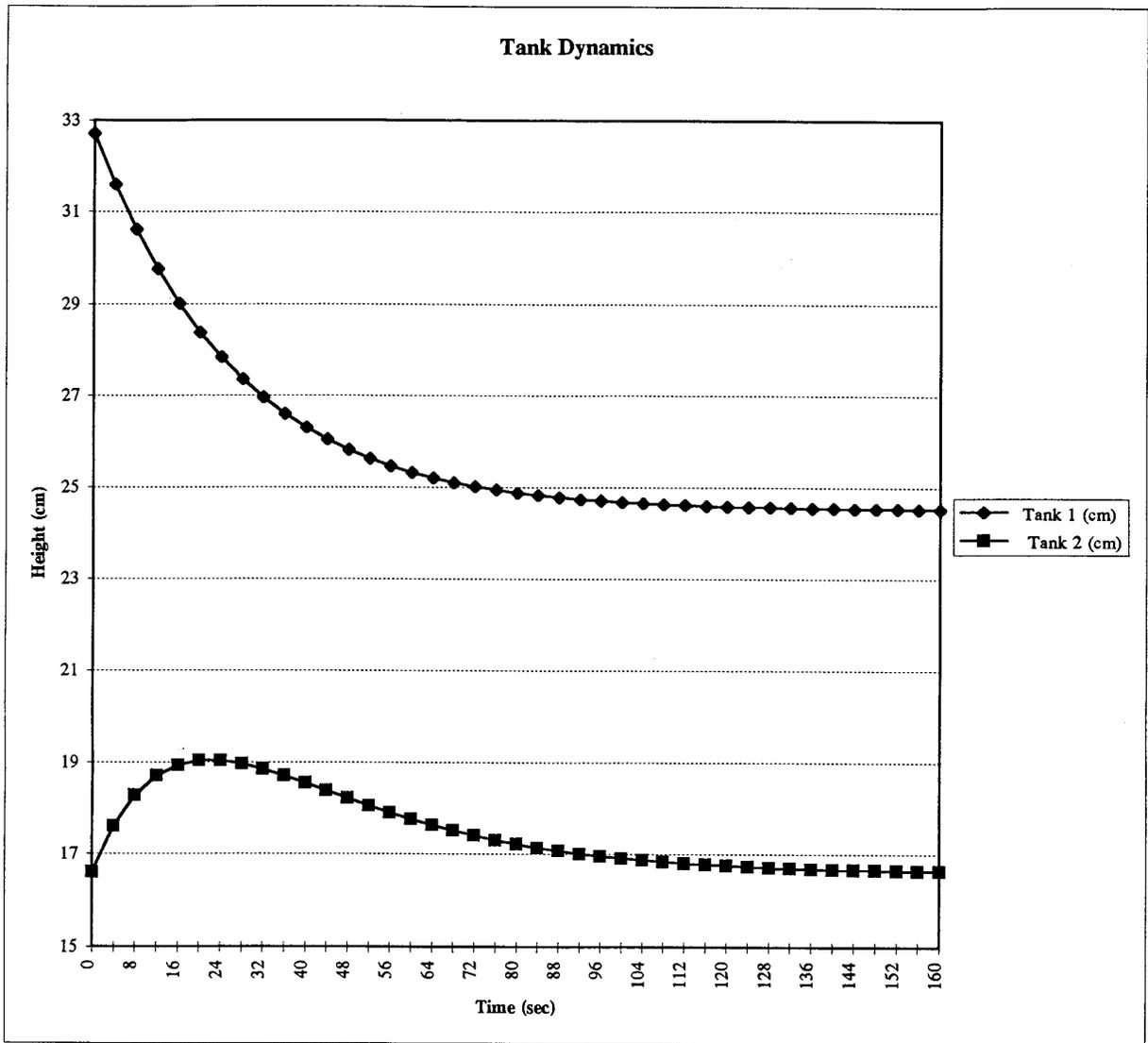


Figure 5. Graph of Tank Dynamics

Educational Materials Available Through the CACHE WWW Site

A Collection of ten numerical problems in chemical engineering solved by
various mathematical software packages

Michael B. Cutlip, University of Connecticut

Session 12 of the Chemical Engineering Summer School at Snowbird, Utah on August 13, 1997 was concerned with "The Use of Mathematical Software in Chemical Engineering." This session provided a major overview of three major mathematical software packages (MathCAD, Mathematica, and POLYMATH), and a set of ten problems was distributed that utilizes the basic numerical methods in problems that are appropriate to a variety of chemical engineering subject areas. The problems are titled according to the chemical engineering principles that are used, and the numerical methods required by the mathematical modeling effort are identified. This problem set is summarized in Table 1.

After the ASEE Summer School, three more sets of solutions were provided by authors who had considerable experience with additional mathematical software packages. The current total is now six packages, and the packages (listed alphabetically) and authors are given below.

Excel - Edward M. Rosen, EMR Technology Group

Maple - Ross Taylor, Clarkson University

MathCAD - John J. Hwalek, University of Maine

Mathematica - H. Eric Nuttall, University of New Mexico

MATLAB - Joseph Brule, John Widmann, Tae Han, and Bruce Finlayson, University of Washington

POLYMATH - Michael B. Cutlip, University of Connecticut and Mordechai Shacham, Ben-Gurion University of the Negev

The complete problem set has now been solved with the following mathematical software packages: Excel, Maple, MathCAD, Mathematica, MATLAB, and Polymath. As a service to the academic community, the CACHE Corporation provides this problem set as well as the individual package writeups and problem solution files for downloading on the WWW at <http://www.che.utexas.edu/cache/>. The problem set and details of the various solutions (about 300 pages) are given in separate documents as Adobe PDF files. The problem solution files can be executed with the particular mathematical software package. Alternately, all of

these materials can also be obtained from an FTP site at the University of Connecticut: <ftp.engr.uconn.edu/pub/ASEE/>

Bruce Finlayson is also planning to distribute a CD-ROM containing these materials to Chemical Engineering Departments in the near future on behalf of the ASEE Summer School. These CD-ROMs will also be available from CACHE at a nominal price.

A recent article on this problem set has appeared in Computer Applications in Engineering Education, Vol. 6, No. 3, pp 169-180 (1998) from which this material is adapted.

Use of the Problem Set

The complete problem writeups from the various packages allow potential users to examine the detailed treatment of a variety of typical problems. This method of presentation should indicate the convenience and strengths/weaknesses of each of the mathematical software packages. The problem files can be executed with the corresponding software package to obtain a sense of the package operation. Parameters can be changed, and the problems can be resolved. These activities should be very helpful in the evaluation and selection of appropriate software packages for personal or educational use.

Additionally attractive for chemical engineering faculty is that individual problems from the problem set can be easily integrated into existing coursework. Problem variations or even open-ended problems can quickly be created. This problem set and the various writeups should be helpful to chemical engineering faculty who are continually faced with the selection of a mathematical problem solving package for use in conjunction with their courses.

Excel is a trademark of Microsoft Corporation (<http://www.microsoft.com>)

Maple is a trademark of Waterloo Maple, Inc. (<http://www.maplesoft.com>)

MathCAD is a trademark of Mathsoft, Inc. (<http://www.mathsoft.com>)

Mathematica is a trademark of Wolfram Research, Inc.

(<http://www.wolfram.com>)

MATLAB is a trademark of The Math Works, Inc.

(<http://www.mathworks.com>)

POLYMATH is copyrighted by M. Shacham and M. B. Cutlip

(<http://www.polymath-software.com>)

Table 1. Problem Set for Use with Mathematical Software Packages^a

SUBJECT AREA	PROBLEM TITLE	MATHEMATICAL MODEL	PROBLEM
Introduction to Ch.E.	Molar Volume and Compressibility Factor from Van Der Waals Equation	Single Nonlinear Equation	1
Introduction to Ch.E.	Steady State Material Balances on a Separation Train*	Simultaneous Linear Equations	2
Mathematical Methods	Vapor Pressure Data Representation by Polynomials and Equations	Polynomial Fitting, Linear and Nonlinear Regression	3
Thermodynamics	Reaction Equilibrium for Multiple Gas Phase Reactions*	Simultaneous Nonlinear Equations	4
Fluid Dynamics	Terminal Velocity of Falling Particles	Single Nonlinear Equation	5
Heat Transfer	Unsteady State Heat Exchange in a Series of Agitated Tanks*	Simultaneous ODE's with known initial conditions	6
Mass Transfer	Diffusion with Chemical Reaction in a One Dimensional Slab	Simultaneous ODE's with split boundary conditions	7
Separation Processes	Binary Batch Distillation**	Simultaneous Differential and Non-linear Algebraic Equations	8
Reaction Engineering	Reversible, Exothermic, Gas Phase Reaction in a Catalytic Reactor"	Simultaneous ODE's and Algebraic Equations	9
Process Dynamics and Control	Dynamics of a Heated Tank with PI Temperature Control**	Simultaneous Stiff ODE's	10

^aThese problems are taken in part from a new book entitled "Problem Solving in Chemical Engineering with Numerical Methods" by Mordechai Shacham and Michael B. Cutlip. See the Announcement on page 22 of this newsletter.

* Problem originally suggested by H.S. Fogler of the University of Michigan

** Problem preparation assistance by N. Brauner of Tel-Aviv Univeristy

Executing FORTRAN Programs From Excel: Use of the Shell Function

Edward M. Rosen, EMR Technology Group

Excel 95 and its associated Visual Basic For Applications (VBA) language provide a very substantial computational capability (1, 2). Nevertheless the ability to transfer out of a VBA subroutine procedure, execute a module in the DOS environment and transfer a result back to the spreadsheet adds substantially to this capability. The shell function of VBA is a means of carrying this out (3).

FORTRAN Programs

FORTRAN has been broadly used to carry out chemical engineering calculations in both academia and industry for a number of years. There is, naturally, a great deal of reluctance to recode FORTRAN programs into the VBA language just so the calculations can be integrated into the spreadsheet.

There are additional reasons to utilize a FORTRAN program. The program may be very long and complex. It may be very much more efficient than the comparable VBA program in terms of execution time. Transferring data to the FORTRAN program from the spreadsheet, doing the FORTRAN based calculations and transferring results back to the spreadsheet may therefore be a very attractive procedure.

The Shell Function

The shell function will start another application from within a VBA procedure. Once the application is started, the next command in the VBA procedure is executed. There is no delay to allow the application to run to completion. If the application takes a relatively long time a delay must be added in the VBA procedure so that the application can be completed and any result can be available to be transferred back to the spreadsheet.

Use of the Shell Function

Figure 1 is a listing of a FORTRAN program that reads in three numbers from an input file (Tsin1), performs some simple arithmetic calculations and then writes the results (five numbers) to an output file (Tsout). The exec module, TESTX.EXE is located in the directory C:\TMP

Figure 2 is a spreadsheet that has three numbers in A3, A4, and A5. It is desired to transmit these numbers to the FORTRAN program of Figure 1 and return the results back into the spreadsheet in locations D3 to D7.

Figure 3 gives a listing of the subroutine procedures that are written in VBA to accomplish these tasks. The procedures are invoked by clicking the CallFile macro in the window resulting from clicking Tools → Macro.

The subroutine procedure then proceeds to:

1. Clear the contents of D3:D7 on the spreadsheet (procedure moda).
2. Calls subroutine WriteDataToTsin1 to create the input file Tsin1.
3. Invokes the execution of TESTX.
4. Causes a delay of 1 sec so that TESTX can complete its calculations.
5. Calls subroutine ReadBackFromTsout to place the results of the calculation in spreadsheet locations D1...D7.

To the user of the macro, the calculation appears as if it took place entirely in the spreadsheet.

Conclusion

Use of Excel 95 and VBA allows for the possible integration of FORTRAN written programs into spreadsheet calculations. At most, some modifications of these FORTRAN programs may be needed to allow for file input and output.

References

1. Rosen, E. M., Visual Basic for Applications, Add-Ins and Excel 7.0, *CACHE News*, No. 45, Fall, 1997.
2. Rosen, E. M., The Case for Excel and Visual Basic for Applications, *CACHE News*, No.46, Spring 1998.
3. Harris, M., Teach Yourself Visual Basic for Applications in 21 Days. *Sams Publications*, Indianapolis (1994).

```

C      PROGRAM TO READ A FILE, DO CALCULATIONS AND OUTPUT THE RESULTS
C
C      DIMENSION X(3), Y(5)
C
C      OPEN (9, FILE='C:\TMP\TSIN1')
C      OPEN (8, FILE='C:\TMP\TSOUT')
C
C      READ (9, *, END = 100) (X(I), I=1,3)
C
C      MISCELLANEOUS CALCULATIONS
C
100    Y(1) = X(1) /X(2)
        Y(2) = X(1) *X(2)
        Y(3) = X(1) +X(2) +X(3)
        Y(4) = X(1) *X(2) *X(3)
        Y(5) = X(1) **2 + X(2) **2 + X(3) **2
C
C      WRITE (8, 1000) (Y(K), K= 1, 5)
1000  FORMAT (4(F9.2,','),1X), F9.2)
C
C      STOP 777
C      END

```

Figure 1. FORTRAN Program

Numbers to Tsin1	Results from TestX
6	0.86 1st/2nd
7	42.00 1st * 2nd
8	21.00 Sum of three
	336.00 Product of Three
	149.00 Sum Sqs of Three

Figure 2. Spreadsheet Showing Input and Output

```

Sub CallFile0
Reset

Call moda

FileIn = "C:\Tmp\Tsin1"
Call WriteDataToTsin1(FileIn)

RetVal = Shell("C:\TMP\TESTX.exe",1)

AddTimeSec = 1
newHour = Hour(Now())
newMinute = Minute(Now())
newSecond = Second(Now0) + AddTimeSec

WaitTime = TimeSerial(newHour, newMinute,
newSecond)

Application Wait WaitTime

FileOut = "C:\Tmp\Tsout"
Call ReadBackFromTsout(FileOut)

End Sub
*****

Sub WriteDataToTsin1(FileIn)

Dim A3 As Single
Dim A4 As Single
Dim A5 As Single

A3 = Cells(3, 1)
A4 = Cells(4, 1)
A5 = Cells(5, 1)

Open FileIn For Output As #1
Write #1, A3, A4, A5

Close #1

End Sub
*****

```

```

Sub ReadBackFromTsout(FileOut)

Open FileOut For Input As #1

Do Until EOF(1)
    Input #1, D1, D2, D3, D4, D5
Loop

Close #1

Cells(3, 4) = D1
Cells(4, 4) = D2
Cells(5, 4) = D3
Cells(6, 4) = D4
Cells(7, 4) = D5

End Sub
*****

Sub moda0
    Sheets("Sheet1").Select
    Range("D3:D7").Select
    Selection.ClearContents
End Sub
*****

```

Figure 3. Listing of Macro Sheet

Engineering Computing with Maple

Ross Taylor, Clarkson University

Introduction

The future of general purpose scientific and engineering computing belongs to software systems that provide exceptional integration (pun intended) of several key functions: symbolic mathematics, graphics, numerical computations, document creation, and an ability to link to other software tools (e.g. spreadsheets, programming languages, numerical libraries, databanks, etc.). While we do not yet have tools that perform all of these tasks with the required level of excellence, the class of software that at present comes closest to this ideal are computer algebra systems (CASs). There are several computer algebra systems in use today: *Macsyma*, *Reduce*, *Derive*, *Mathematica*, *Maple* and *Scratchpad* are some of the better known ones. Many modern CASs integrate (if we may be permitted to use the word) symbolic mathematical capabilities (integration and differentiation, for example) with numerical capabilities (e.g. integration of ODE systems and linear and nonlinear equation solving) and sophisticated graphics (including 3-dimensional plots of (parametric) surfaces, spacecurves, and much more). Other tools are useful for the numerical and graphical calculations but only a CAS can help us with the symbolic computation.

For the past few years the author has made extensive use of *Maple* to tackle a wide variety of problems in chemical engineering (primarily thermodynamics) [1-9], numerical computing [10,11] and cartography [12]. This article provides an overview of some of our work with *Maple* (for complete details readers should consult the references).

Thermodynamics with Maple

Thermodynamics deals with relationships between eight state variables normally assigned the letters P , v , T , U , H , S , A , and G (these letters stand for Pressure, Volume, Temperature, Internal Energy, Enthalpy, Entropy, Helmholtz Energy, and Gibbs Energy respectively). In order to be useful in the exploration of thermodynamics, a computer algebra system needs:

1. The ability to differentiate undefined functions of several variables including undefined functions of an undefined number of variables.
2. To be able to work with the subscripted (indexed) partial derivatives of thermodynamic functions found in many thermodynamic formulae.
3. The ability to differentiate arbitrary sums (a summation where the upper limit is a symbol rather than a number) with respect to an indexed variable.

4. The ability to display the specialized graphical images required in thermodynamics.

To the best of our knowledge none of the major commercial computer algebra systems can do all (or, in some cases, any) of the above right out of the box. However, most CASs can be taught new tricks and we have done this in a package for Maple called TDtools that can "do" all of the above, as illustrated below.

Derivatives of Undefined Functions

Consider an arbitrary thermodynamic property, X , and assume it to be a function of two other variables, Y and Z say.

```
> xdef:=X=X(Y,Z) : xdef;
```

$$X = X(Y, Z)$$

In order to explore the world of thermodynamic property relations we need to be able to obtain differentials of undefined functions like this one. Maple's built in differential operator, D , is unable to perform this operation. We have, therefore, created a new operator called TD that is able to take the total differential of such functions as shown below. In all other respects, TD is supposed to behave like D .

```
> alias(d=D):  
> t0:=TD(xdef): t0;
```

$$d(X) = \left(\frac{\partial}{\partial Y} X \right)_Z d(Y) + \left(\frac{\partial}{\partial Z} X \right)_Y d(Z)$$

Partial Derivatives in Thermodynamics

Note the indexed partial derivatives in the above expression. Such derivatives are important in thermodynamics and are returned by the TDiff procedure.

```
> TDiff(X,Y);
```

$$\frac{\partial}{\partial Y} X$$

If TDiff is called with more than two arguments, the additional arguments are assumed to be constant variables and become (or are added to) the index.

> TDiff(X,Y,Z);

$$\left(\frac{\partial}{\partial Y} X\right)_L$$

Second (and higher) derivatives can be obtained by repeated application of TDiff.

The functions TD and TDiff (as well as a few others) allow us easily to derive many other relations between partial derivatives: the triple product rule,

$$\left(\frac{\partial}{\partial Y} X\right)_Z \left(\frac{\partial}{\partial Z} Y\right)_X \left(\frac{\partial}{\partial X} Z\right)_Y = -1$$

the chain rule,

$$\left(\frac{\partial}{\partial Y} X\right)_K \left(\frac{\partial}{\partial Z} Y\right)_K \left(\frac{\partial}{\partial X} Z\right)_K = 1$$

the insertion rule,

$$\left(\frac{\partial}{\partial K} X\right)_L = \left(\frac{\partial}{\partial Y} X\right)_Z \left(\frac{\partial}{\partial K} Y\right)_L + \left(\frac{\partial}{\partial Z} X\right)_Y \left(\frac{\partial}{\partial K} Z\right)_L$$

and the following expression (see [8] for details).

$$\left(\frac{\partial}{\partial Y} X\right)_Z = \frac{\left(\frac{\partial}{\partial K} X\right)_L \left(\frac{\partial}{\partial L} Z\right)_K - \left(\frac{\partial}{\partial L} X\right)_K \left(\frac{\partial}{\partial K} Z\right)_L}{\left(\frac{\partial}{\partial K} Y\right)_L \left(\frac{\partial}{\partial L} Z\right)_K - \left(\frac{\partial}{\partial L} Y\right)_K \left(\frac{\partial}{\partial K} Z\right)_L}$$

This seemingly complicated result is of immense utility for deriving relationships between thermodynamic properties. We may encapsulate Maple results in procedures for later use.

```
> DXYZ := proc(X,Y,Z,K,L)
> Diff(X,Y) [Z] = (Diff(X,L) [K] *
> Diff(Z,K) [L] - Diff(X,K)[L] *
> Diff(Z,L)[K]) /
> (Diff(Y,L)[K] * Diff(Z,K)[L]-
> Diff(Y,K)[L] * Diff(Z,L)[K]);
> end;
```

Some of the partial derivatives in the above expression can be either 0 or 1 and these simplifications are encoded into `simplify/TDiff` to which we include a call inside the DXYZ procedure shown below.

```
> DXYZ := proc (X,Y,Z)
> local expr, L, K;
> expr:=Diff(X,Y)[Z] =
> (Diff(X,T)[P]
> Diff(Z,P)[T] - Diff(X,P)[T] *
> Diff(Z,T)[P]) /
```

```
> (Diff(Y,T)[P] *
> Diff(Z,P)[T] -
> Diff(Y,P)[T] *
> Diff(Z,T)[P]);
> expr := simplify(expr,TDiff);
> expr := simplify(Right(
> simplify,expr,Thermo));
> end;
```

We have cut down the number of arguments to three since the last two are T and P. We have also added a call to `simplify/Thermo` that knows the Maxwell relations, the definition of heat capacity and a few other thermodynamic property relations. The code is not shown here since it is lengthy; however, we should note that all of the simplifications contained within that procedure can be derived using Maple. We can use this procedure to express any thermodynamic partial derivative in terms of measurable properties (PvT data, heat capacities) and entropy. Some examples follow.

```
> DXYZ(G,A,U);
```

$$\left(\frac{\partial}{\partial A} G\right)_U = \frac{ST\left(\frac{\partial}{\partial T} v\right)_P + SP\left(\frac{\partial}{\partial P} v\right)_T - vC_p + vP\left(\frac{\partial}{\partial T} v\right)_P}{P\left(\frac{\partial}{\partial T} v\right)_P^2 T + ST\left(\frac{\partial}{\partial T} v\right)_T + SP\left(\frac{\partial}{\partial P} v\right)_T + \left(\frac{\partial}{\partial P} v\right)_T C_p}$$

```
> DXYZ(S,v,A);
```

$$\left(\frac{\partial}{\partial v} S\right)_A = -\frac{P\left(\frac{\partial}{\partial P} v\right)_T C_p + P\left(\frac{\partial}{\partial T} v\right)_P^2 T + ST\left(\frac{\partial}{\partial T} v\right)_P}{TS\left(\frac{\partial}{\partial P} v\right)_T}$$

All of the more than 300 different partial derivatives can now be presented error-free and completely automatically in just a few seconds. Alternative forms (perhaps involving compressibility) can be obtained by adding to the set of relations known to `simplify/Thermo`.

Differentiation of Arbitrary Sums

The NRTL model for the (dimensionless) Gibbs free energy of a mixture is:

```
> Qdef:=Q=sum(x[i] * (sum(x[j] *
> tau[j,i]*G[j,i],j=1..c) /sum(x[j] *
> G[j,i],j=1..c)),i=1..c): Qdef;
```

$$Q = \sum_{i=1}^c \frac{x_j \left(\sum_{j=1}^c x_j \tau_{j,i} G_{j,i} \right)}{\sum_{j=1}^c x_j G_{j,i}}$$

This expression is the starting point for working out expressions for the activity coefficients of each species in the mixture defined by:

> ln(gamma[i])=Diff(n[t]*Q,n[i]);

$$\ln(\gamma_i) = \frac{\partial}{\partial n_i} n_i Q$$

The differentiation required by the two prior expressions involves differentiation of an undefined or arbitrary sum, where the upper limit is a symbol that represents some presently unknown number of species. Maple's own differentiation functions do not know how to differentiate such objects with respect to one of the indexed variables; TDiff does.

The next step in the derivation is to replace the mole fractions by their mole number ratios (and to multiply the result by the total number of moles):

> Q1:=n[t]*subs(seq(x[v]=n[v]/n[t],
v=[i,j,k,l,m,n]),Qdef);
> Q2:=expand(%): Q2;

$$n_i Q = \sum_{i=1}^c \frac{n_i \left(\sum_{j=1}^c n_j \tau_{j,i} G_{j,i} \right)}{\sum_{j=1}^c n_j G_{j,i}}$$

and invoke TDiff to obtain the derivative of Q with respect to the number of moles of the i -th species (the parameters are assumed constant).

> t1:=ln(gamma[i])=
TDiff(rhs(Q2),n[i],[tau,G]);

The result of the above operation is quite correct but is not shown for reasons of space. In what follows we use Maple to present the result in a somewhat more familiar form using mole fractions rather than mole numbers.

> subs(seq(n[v]=x[v]*n[t],
> v=[i,j,k,l,m,n]),t1);
> expand(expand(%));

$$\ln(\gamma_j) = \frac{\sum_{k=1}^c x_k \tau_{k,i} G_{k,i}}{\sum_{k=1}^c x_k G_{k,i}} + \left(\frac{\sum_{j=1}^c x_j \tau_{i,j} G_{i,j}}{\sum_{k=1}^c x_k G_{k,j}} \right) - \left(\frac{\sum_{j=1}^c x_j G_{i,j} \left(\sum_{k=1}^c x_k \tau_{k,j} G_{k,j} \right)}{\left(\sum_{k=1}^c x_k G_{k,j} \right)^2} \right)$$

All this can, of course, be automated and the same commands can be used for any activity coefficient model [6]. Similar procedures may also be developed for deriving expressions for fugacity coefficients in multicomponent mixtures [6,7].

Maple Graphics

Maple possesses an extensive set functions for creating a wide variety of graphical effects (the plots and plottools pack-

ages). There remain several plot types that are of interest in thermodynamics but which cannot be displayed by standard Maple (curves, surfaces, and vector fields in triangular or tetrahedral coordinates, for example). Fortunately, it is reasonably simple to create new plot types in Maple. In the accompanying figures we show some of the thermodynamic images we have created - often surprisingly easily - with Maple. Limitations of space prevent us from showing in detail exactly how these plots are created, interested readers should consult the cited references and our web site. This author has found the Programming Guide that is supplied with Maple and the book by Klimek and Klimek [13] to be particularly helpful in explaining how to dismantle and re-assemble Maple plots.

The fundamental relationships between state variables were explored in three famous papers by J.W. Gibbs [14]. In the second of these papers Gibbs explores the geometry of the $U(S,v)$ surface. It is interesting to note that there are very few illustrations in Gibbs' papers; it seems that Gibbs could "see" the relationships between state variables. There should be little doubt that visualization of thermodynamic relations can aid an understanding of what for many is a difficult subject. Not all of us are able to see thermodynamics in our heads as easily as did Gibbs, but few texts contain graphical visualizations of any of what we refer to here as Gibbs surfaces. An exception is the advanced text of Tester and Modell [15] which contains a few images created using computer software developed by Jolls and Coy [16-19]. Their 12,000 line Fortran program, initially created to run on a few specific platforms and later ported to others, provides some wonderful graphic images of fundamental thermodynamic functions.

Figures 1 to 3 show Gibbs' surfaces created using Maple. Given the parametric equations (in terms of temperature and volume) Maple can create these surfaces with a one-line command. The black and red lines are the binodal and spinodal curves, respectively. These lines require some additional computational work. Several other examples and complete details of how to construct these figures are given by Baur et al. [9]. It is very instructive to view these surfaces from different vantage points, a task that is made considerably easier in the latest version of Maple which can rotate three dimensional images in real time.

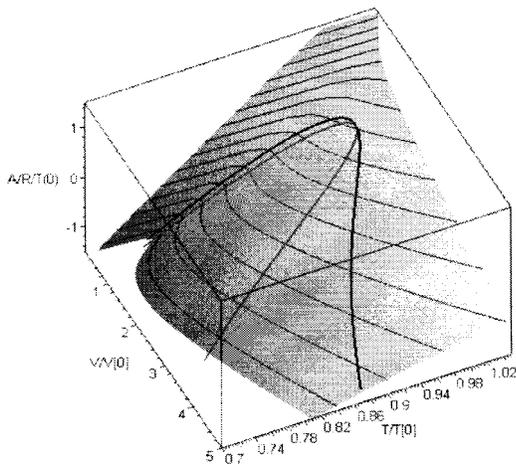


Figure 1. Helmholtz energy surface as a function of (dimensionless) temperature and volume

Additional examples, including some frames from an animation showing the tangent plane sliding over the $U(S,v)$ surface are given by Baur et al. (1998).

Figures 4 and 5 show residue curves and a vector field diagram for a mixture of acetone, methanol, and chloroform. These figures require extensive numerical computation prior to their construction; details are available in [8]. Maple can create vector field plots out of the box, but not for the triangular field needed here. Creating one's own field plots is very simple given the tools available in Maple.

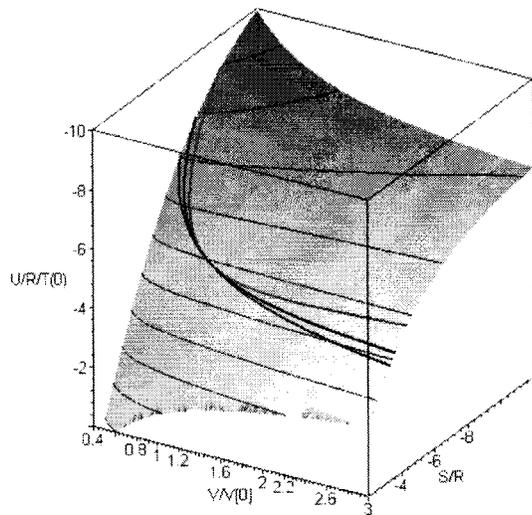


Figure 2. Internal energy surface as a function of (dimensionless) entropy and volume

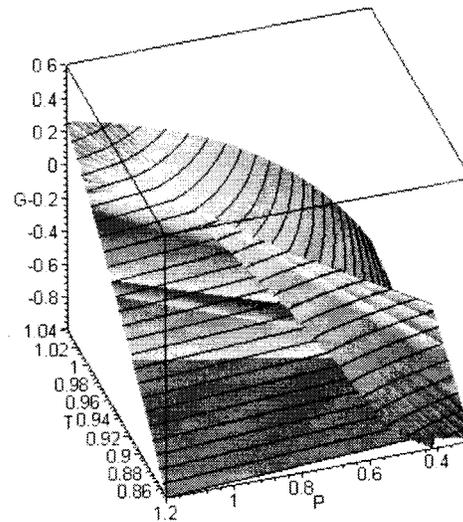


Figure 3. Gibbs energy as a function of temperature and pressure

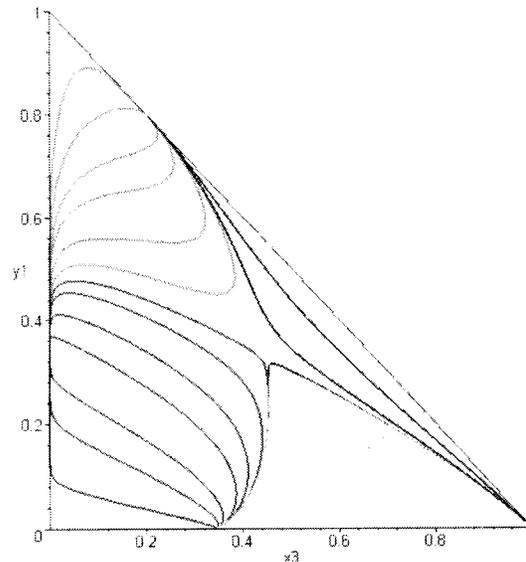


Figure 4. Residue curves for acetone, methanol, and chloroform

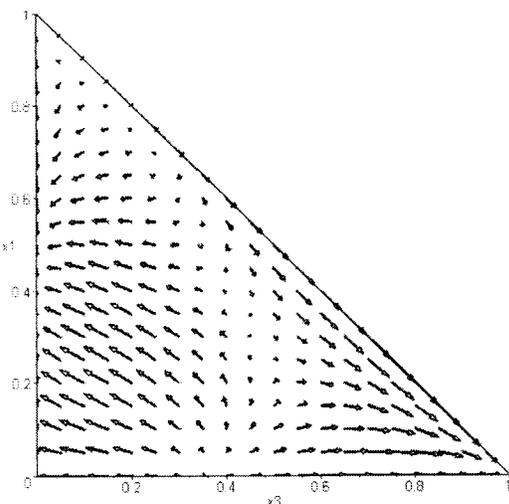


Figure 5. Vector field diagram for acetone, methanol, and chloroform

The location of the binary and ternary azeotropes is clearly visible in Figures 4 and 5. The computation of all azeotropes in a multicomponent mixture has been the subject of a number of recent papers and sophisticated algorithms for this purpose have been published (e.g. [20]). We have found it possible to compute all azeotropes using nothing more than a constrained Newton's method (at least, we find all the azeotropes listed in [20], the difference is that we cannot prove that we have them all). The result of our computations (see [8] for the code) is a Maple table, an example for this system is:

$$\text{Azeotable3} = \begin{bmatrix} \text{Acetone} & \text{Chloroform} & \text{Methanol} & \text{T} & \text{Type} \\ 1 & 0 & 0 & 56.07 & \text{S} \\ 0 & 1 & 0 & 61.79 & \text{S} \\ 0 & 0 & 1 & 64.54 & \text{SN} \\ .3437 & .6563 & 0 & 65.47 & \text{SN} \\ 0 & .6482 & .3518 & 53.91 & \text{UN} \\ .7944 & 0 & .2056 & 55.34 & \text{UN} \\ .3234 & .2236 & .4530 & 57.58 & \text{S} \end{bmatrix}$$

Note that our code has determined the composition and temperature of the azeotropes as well as whether it is a stable node (SN), unstable node (UN) or saddle point (S). We can pass this table to another procedure that works out the distillation boundaries and draws an approximate distillation boundary map as shown below. The procedure DBM plot is an implementation in Maple of the algorithm of Foucher et al. [21] with the exception that we have the advantage of knowing the stability of the azeotropic points.

> DBMplot(Azeotable3);

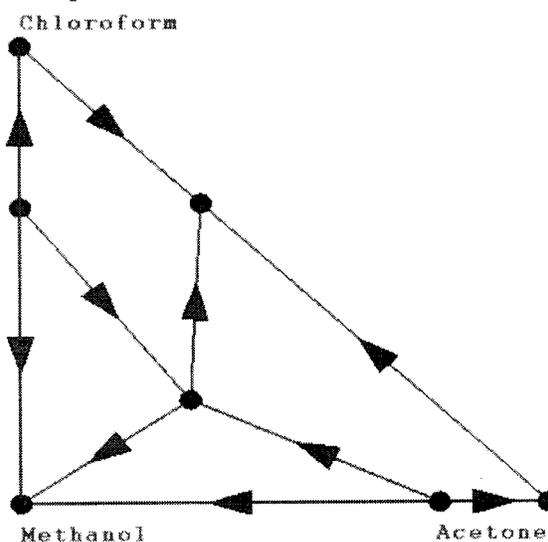


Figure 6. Approximate distillation boundary map for acetone, methanol, chloroform

Numerical Computing

The emphasis on symbolic manipulation in any definition of computer algebra should not be taken to imply that a CAS is unsuitable for the predominantly numerical calculations that are such an important part of modern engineering computing. It is true that numerical computing is not Maple's strongest suit, but the program does possess a number of functions designed specifically for such activities. The built in numerical equation solver, fsolve, is an excellent tool for solving single nonlinear equations; sadly it is much less successful on problems involving systems of nonlinear equations. In view of this weakness we have developed a fairly sophisticated implementation of Newton's method for Maple that allows the user considerable control over the computations. Early versions of Newton were supplied with Maple in the share library of user supplied functions. The latest and considerably improved version is available from the authors web site. We have also implemented homotopy continuation methods in Maple, but that code has not yet been released as part of the Newton package (although we do intend to do so in the near future).

Many engineering problems require the numerical solution of differential equations, some require the solution of mixed systems of differential and algebraic equations (DAEs). Maple comes with a number of built in numerical integration methods including Runge-Kutta, Gear's method and the LSODE algorithms. It is not, however, capable of solving DAEs. To deal with this shortcoming we have developed BESIRK [10], an implementation in Maple of Michelsen's 3rd order SIRK method combined with a Bulirsch-Stoer extrapolation technique. The BESIRK code can handle stiff

systems of ODEs significantly faster than Maple's own integration routines and was used in the computation of the residue curves shown in Figure 4. We have also found the BESIRK code to be of considerable value in solving PDEs by the Method of Lines[11].

Numerical problem solving using interval methods has been used to solve a number of chemical engineering problems that often are considered "difficult" [22]. Even out of the box, Maple is capable of some elementary interval computations. A more complete package for interval computing with Maple has been developed by Rob Corless and Amanda Connell of the University of Western Ontario and supplied with Maple as part of its share library. George Corliss from Marquette University and his students developed an interval Newton method that uses the Corless/Connell package. However, their code does not work in the more recent releases of Maple (3, 4 and 5). We have modified the Corliss code for Release 4; the code is available on our web site.

Those of us that still prefer to program in more traditional languages may like to know that Maple can also be used to turn expressions and procedures into Fortran or C code. Maple has its own built in functions for this purpose and there is a package called Transfor developed by Claude Gomez that can be used for this purpose. Gomez is also a developer of Scilab, a (free) scientific software package for numerical computing in a user friendly environment. Scilab (with which this author has no personal experience) has links to high powered numerical algorithms (e.g. DASSL) and can carry out symbolic computing via a link to Maple.

More Maple

Readers wishing to experiment for themselves with Maple may find sample worksheets to be of some value. Complete worksheets for all of the above (and many more) examples are available on the authors web site. Worksheets by the author that demonstrate the use of Maple for solving simple material balance problems are supplied with Maple as part of the share library (in /share/engineer/chemeng). The author has also contributed 10 Maple worksheets to the CD that accompanies a new book by Cutlip and Schacham [23] (and many more are under development).

Perhaps the single most useful additional capability of Release 4 of Maple V (the current version is Release 5) was its ability to read binary random access files of arbitrary format. Maple's new found abilities at handling binary data were, however, flawed since, other than conversion to and from text strings, Release 4 did not come with utilities to convert binary data into the integers and floating point numbers that could be used in other computations. Following a request by this author posted on the internet, electrical engineer Joe Riel provided the functions needed to make it very easy to manipulate binary data within Maple. We have, for example,

used his package to develop code for reading the databank that comes with ChemSep, thus providing Maple with access to physical property data for use in many engineering computations. Riel has also written a units conversion package that this author uses in a great many worksheets. Riel's packages (ieec and mks) are available on the web site of Alex Walz. Walz is also the author of some of the best mathematical and utility packages that significantly enhance the ease of use of Maple.

Acknowledgements

Several students, both graduate and undergraduate, contributed to the development of the Maple codes used in this work. They are: Arnoud Higler, Richard Baur, Dorothy Scheckler, James Reese, and Warren Hoffmaster.

References

1. Taylor, R. and K. Atherley, Chemical Engineering with Maple, *Chem. Eng. Ed.*, 56-61, Winter 1995.
2. Taylor, R., Thermodynamics with Maple. I – Equations of State, Maple Tech, Issue 10, 50-59 (1993).
3. Taylor, R., Thermodynamics with Maple. II – Phase Equilibria in Binary Systems, Maple Tech, **1**(1), 83-92 (1994)
4. Adams, S. and R. Taylor, Thermodynamics with Maple. III - Thermodynamic Property Relations and the Maxwell equations, Maple Tech, **1**(2), 68-81, 1994.
5. Taylor, R., Thermodynamics with Maple. IV – The Properties of Steam, Maple Tech, **3**(2), 61-68, 1996.
6. Taylor, R., Automatic Derivation of Thermodynamic Property Functions using Computer Algebra, *Fluid Phase Equilibria*, **129**, 37-47, 1997.
7. Taylor, R., Thermodynamics with Maple. I - Symbolic Computation, *Mathematics and Computation in Simulation*, **45**, 101-119, 1998.
8. Taylor, R., Thermodynamics with Maple. II - Numerical and Graphical Applications, *Mathematics and Computation in Simulation*, **45**, 121-146, 1998.
9. Baur, R. J. Bailey, B. Brol, A. Tatusko, and R. Taylor, Maple and the Art of Thermodynamics, *Computer Applications in Engineering Education*, in press 1998.
10. Schwalbe, D., H.A. Kooijman, and R. Taylor, Solving Stiff Differential Equations and Differential Algebraic Systems with Maple V Maple Tech, **3**(2), 47-53, 1996.
11. Kooijman, H.A. and R. Taylor, Numerical Computing with Maple - Solving PDEs by the Method of Lines, under review, 1998.
12. Taylor, R., R. Baur, and J. Oprea, Maple Maps, under review, 1998.
13. Klimek, G. and M. Klimek, *Discovering Curves and Surfaces with Maple*, Springer, New York, 1997.
14. Gibbs, J.W., The Scientific Papers of J.W. Gibbs, Ph.D., LLD., Volume 1 - Thermodynamics, Dover, New York, 1961.

-
15. Testor, J.W. and M. Modell, *Thermodynamics and its Applications*, 3rd edition, Prentice-Hall, Inc, 1996.
 16. Jolls, K.R. Gibbs and the Art of Thermodynamics, Proc. Gibbs Symposium (G.D. Mostow and D.G. Caldi, eds.), 293, American Mathematical Society, 1990.
 17. Coy, D.C., Visualizing Thermodynamic Stability and Phase Equilibrium through Computer Graphics. Ph.D. thesis, Iowa State University, 1993.
 18. Jolls, K.R. Understanding Thermodynamics through Interactive Computer Graphics, *Chem. Eng. Progress*, 64-69, Feb. 1989.
 19. Jolls, K.R., M.C. Schmitz, and D.C. Coy, Seeing is Believing: A New Look at an Old Subject. *The Chemical Engineer*, 42-46, May 30, 1991.
 20. Fidkowski, Z.T., M.F. Malone, and M.F. Doherty, Computing Azeotropes in Multicomponent Mixtures, *Comput. Chem. Eng.*, 17, 1141-1155, 1993.
 21. Foucher, E.R., M.F. Doherty, and M.F. Malone, Automatic Screening of Entrainers for Homogeneous Azeotropic Distillation, *Ind. Eng. Chem. Research*, **29**, 760-772, 1991;
 22. Hua, J.Z., J.F. Brenneke, and M.A. Stadtherr, *Ind. Eng. Chem. Research*, **37**, 1519-1527, 1998; *Fluid Phase Equilibria*, **116**, 52-59, 1996.
 23. Cutlip, M.B., and M. Schacham, *Problem Solving in Chemical Engineering with Numerical Methods*, Prentice-Hall, 1998.

Web Sites

Claude Gomez:

www-rocq.inria.fr/scilab/gomez/transfor/transfor.html

R. Taylor:

www.clarkson.edu/~chengweb/faculty/taylor/taylor.html

Alexander F. Walz:

www.SunSite.informatik.rwth-achen.de/maple/maplev.html

The Numerical Method of Lines for Partial Differential Equations

Michael B. Cutlip, University of Connecticut and
Mordechai Shacham, Ben Gurion University of the Negev

The method of lines is a general technique for solving partial differential equations (PDEs) by typically using finite difference relationships for the spatial derivatives and ordinary differential equations for the time derivative. William E. Schiesser at Lehigh University has been a major proponent of the numerical method of lines, NMOL.¹ This solution approach can be very useful with undergraduates when this technique is implemented in conjunction with a convenient ODE solver package such as POLYMATH.²

A Problem in Unsteady-State Heat Transfer³

This approach can be illustrated by considering a problem in unsteady-state heat conduction in a one-dimensional slab with one face insulated and constant thermal conductivity as discussed by Geankoplis.⁴

Unsteady-state heat transfer in a slab in the x direction is described by the partial differential equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (1)$$

where T is the temperature in K, t is the time in s, and α is the thermal diffusivity in m^2/s given by $k/\rho c_p$. In this treatment, the thermal conductivity k in $\text{W}/\text{m}\cdot\text{K}$, the density ρ in kg/m^3 and the heat capacity c_p in $\text{J}/\text{kg}\cdot\text{K}$ are all considered to be constant.

Consider that a slab of material with a thickness 1.00 m is supported on a nonconducting insulation. This slab is shown in Figure 1. For a numerical problem solution, the slab is divided into N sections with $N + 1$ node points. The slab is initially at a uniform temperature of 100°C . This gives the initial condition that all the internal node temperatures are known at time $t = 0$.

$$T_n = 100 \text{ for } n = 2 \dots (N + 1) \text{ at } t = 0 \quad (2)$$

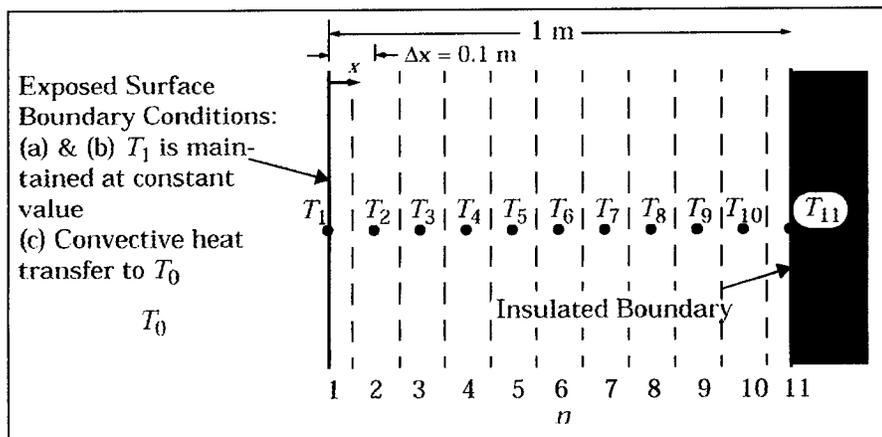


Figure 1. Unsteady-State Heat Conduction in a One-dimensional Slab

¹ Schiesser, W.E. *The Numerical Method of Lines*. San Diego, CA: Academic Press, 1991.

² POLYMATH is a numerical analysis package for IBM-compatible personal computers that is available through the CACHE Corporation. Information can be found at www.polymath-software.com.

³ This problem is adapted in part from Cutlip, M.B., and M. Shacham, *Problem Solving in Chemical Engineering with Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall, 1999.

⁴ Geankoplis, C.J. *Transport Processes and Unit Operations*. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1993.

If at time zero the exposed surface is suddenly held constant at a temperature of 0 °C, this gives the boundary condition at node 1:

$$T_1 = 0 \text{ for } t \geq 0 \quad (3)$$

The other boundary condition is that the insulated boundary at node $N + 1$ allows no heat conduction. Thus

$$\frac{\partial T_{N+1}}{\partial x} = 0 \text{ for } t \geq 0 \quad (4)$$

Note that this problem is equivalent to having a slab of twice the thickness exposed to the initial temperature on both faces.

Problem (a) - Numerically solve Equation (1) with the initial and boundary conditions of (2), (3), and (4) for the case where $\alpha = 2 \times 10^{-5} \text{ m}^2/\text{s}$ and the slab surface is held constant at $T_1 = 0 \text{ }^\circ\text{C}$. This solution should utilize the numerical method of lines with $N = 10$ sections. Plot the temperatures $T_2, T_3, T_4,$ and T_5 as functions of time to 6000 s.

For this problem with $N = 10$ sections of length $\Delta x = 0.1 \text{ m}$, Equation (1) can be rewritten using a central difference formula for the second derivative as

$$\frac{\partial T_n}{\partial t} = \frac{\alpha}{(\Delta x)^2} (T_{n+1} - 2T_n + T_{n-1}) \text{ for } (2 \leq n \leq 10) \quad (5)$$

The boundary condition represented by Equation (4) can be written using a second-order backward finite difference as

$$\frac{\partial T_{11}}{\partial t} = \frac{3T_{11} - 4T_{10} + T_9}{2\Delta x} = 0 \quad (6)$$

that can be solved for T_{11} to yield

$$T_{11} = \frac{4T_{10} - T_9}{3} \quad (7)$$

The problem then requires the solution of Equations (3), (5), and (7) which results in nine simultaneous ordinary differential equations and two explicit algebraic equation for the 11 temperatures at the various nodes. This set of equations can be entered into the POLYMATH Simultaneous Differential Equation Solver or some other ODE solver. The resulting equation set for POLYMATH is

Equations:

```

d(T2)/d(t)=alpha/deltax^2*(T3-2*T2+T1)
d(T3)/d(t)=alpha/deltax^2*(T4-2*T3+T2)
d(T4)/d(t)=alpha/deltax^2*(T5-2*T4+T3)
d(T5)/d(t)=alpha/deltax^2*(T6-2*T5+T4)
d(T6)/d(t)=alpha/deltax^2*(T7-2*T6+T5)
d(T7)/d(t)=alpha/deltax^2*(T8-2*T7+T6)
d(T8)/d(t)=alpha/deltax^2*(T9-2*T8+T7)
d(T9)/d(t)=alpha/deltax^2*(T10-2*T9+T8)
d(T10)/d(t)=alpha/deltax^2*(T11-2*T10+T9)
alpha=2.e-5
T1=0
T11=(4*T10-T9)/3
deltax=.10
    
```

The initial condition for each of the T 's is 100 and the independent variable t varies from 0 to 6000. The plots of the temperatures in the first four sections, node points 2 ... 5, are shown in Figure 2. The transients in temperatures show an approach to steady state. The numerical results are compared to the hand calculations of a finite difference solution

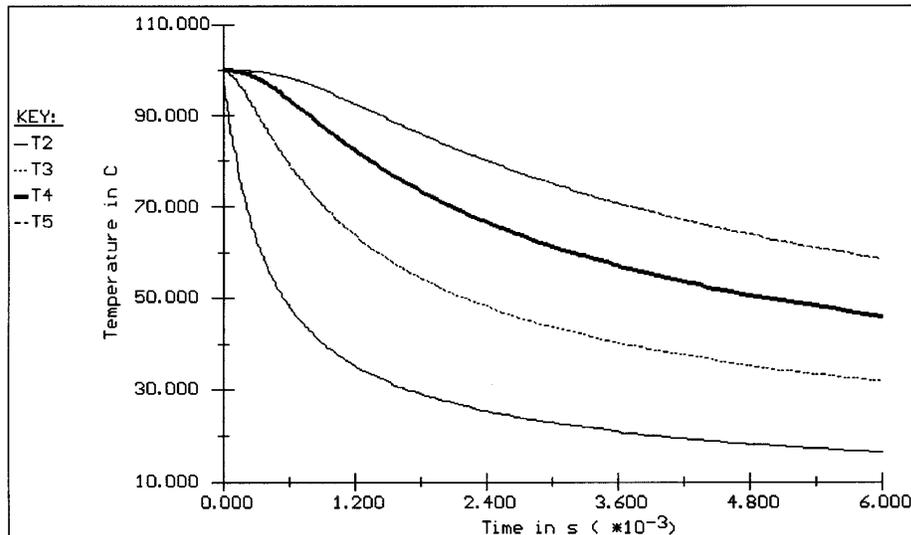


Figure 2. Temperature Profiles for Unsteady-state Heat Conduction in a One-dimensional Slab

by Geankoplis⁴ (pp. 471–3) at the time of 6000 s in Table 1. These results indicate that there is general agreement regarding the problem solution, but differences between the tem-

peratures at corresponding nodes increase as the insulated boundary of the slab is approached.

Table 1. Results for Unsteady-state Heat Transfer in a One-dimensional Slab at $t = 6000$ s

Distance from Slab Surface in m	Geankoplis ⁴		Numerical Method of Lines					
	$\Delta x = 0.2$ m N = 5		$\Delta x = 0.1$ m N = 10		$\Delta x = 0.05$ m N = 20		$\Delta x = 0.0333$ m N = 30	
	n	T	n	T	n	T	n	T
0	1	0.0	1	0.0	1	0.0	1	0.0
0.2	2	31.25	3	31.75	5	31.68	7	31.67
0.4	3	58.59	5	58.49	9	58.47	13	58.47
0.6	4	78.13	7	77.46	13	77.49	19	77.50
0.8	5	89.84	9	88.22	17	88.29	25	88.31
1.0	6	93.75	11	91.66	21	91.72	31	91.73

Problem (b) - Repeat Problem (a) with 20 sections and compare results with part (a).

The validity of the numerical solution can be investigated by doubling the number of sections for the NMOL solution. This involves adding an additional 10 equations given by the relationship in Equation (5), modifying Equation (7) to calculate T_{21} , and halving Δx . The results for these changes in the POLYMATH equation set are also summarized in Table 1 as are similar results for 30 sections. Here the numerical solutions are similar to the previous solution in part (a) as the temperature profiles are virtually unchanged as the number of section is increased.

Problem (c) - Repeat parts (a) and (b) for the case where heat convection is present at the slab surface. The heat transfer coefficient is $h = 25.0$ W/m²·K, and the thermal conductivity is $k = 10.0$ W/m·K.

When convection is considered as the only mode of heat transfer to the surface of the slab, an energy balance can be made at the interface that relates the energy input by convection to the energy output by conduction. Thus at any time for transport normal to the slab surface in the x direction

$$h(T_0 - T_1) = -k \left. \frac{\partial T}{\partial x} \right|_{x=0} \quad (8)$$

where h is the convective heat transfer coefficient in W/m²·K and T_0 is the ambient temperature.

The preceding energy balance at the slab surface can be used to determine a relationship between the slab surface temperature T_1 , the ambient temperature T_0 , and the temperatures at internal node points. In this case, the second-order

forward difference equation for the first derivative can be applied at the surface

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} = \frac{(-T_3 + 4T_2 - 3T_1)}{2\Delta x} \quad (9)$$

and can be substituted into Equation (8) to yield

$$h(T_0 - T_1) = -k \frac{(-T_3 + 4T_2 - 3T_1)}{2\Delta x} \quad (10)$$

The preceding equation can be solved for T_1 to give

$$T_1 = \frac{2hT_0\Delta x - kT_3 + 4kT_2}{3k + 2h\Delta x} \quad (11)$$

and the above equation can be used to calculate T_1 during the NMOL solution.

The resulting equation set for POLYMATH for $\Delta x = 0.10$ m for $N = 10$ is

Equations:

$$\begin{aligned} d(T2)/d(t) &= \alpha/\text{deltax}^2 * (T3 - 2*T2 + T1) \\ d(T3)/d(t) &= \alpha/\text{deltax}^2 * (T4 - 2*T3 + T2) \\ d(T4)/d(t) &= \alpha/\text{deltax}^2 * (T5 - 2*T4 + T3) \\ d(T5)/d(t) &= \alpha/\text{deltax}^2 * (T6 - 2*T5 + T4) \\ d(T6)/d(t) &= \alpha/\text{deltax}^2 * (T7 - 2*T6 + T5) \\ d(T7)/d(t) &= \alpha/\text{deltax}^2 * (T8 - 2*T7 + T6) \\ d(T8)/d(t) &= \alpha/\text{deltax}^2 * (T9 - 2*T8 + T7) \\ d(T9)/d(t) &= \alpha/\text{deltax}^2 * (T10 - 2*T9 + T8) \\ d(T10)/d(t) &= \alpha/\text{deltax}^2 * (T11 - 2*T10 + T9) \end{aligned}$$

$\alpha = 2 \cdot 10^{-5}$
 $\Delta x = 0.10$
 $T_{11} = (4 \cdot T_{10} - T_9) / 3$
 $h = 25$
 $T_0 = 0$
 $k = 10$
 $T_1 = (2 \cdot h \cdot T_0 \cdot \Delta x - k \cdot T_3 + 4 \cdot k \cdot T_2) / (3 \cdot k + 2 \cdot h \cdot \Delta x)$

The preceding equation set can be integrated to any time t with POLYMATH or another ODE solver. The results at $t = 1500$ s are summarized in Table 2.

Table 2. Results for Unsteady-state Heat Transfer with Convection in a One-dimensional Slab at $t = 1500$ s

Distance from Slab Surface in m	Geankoplis ⁴		Numerical Method of Lines					
	$\Delta x = 0.2$ m N = 5		$\Delta x = 0.1$ m N = 10		$\Delta x = 0.05$ m N = 20		$\Delta x = 0.0333$ m N = 30	
	n	T	n	T	n	T	n	T
0	1	64.07	1	64.40	1	64.99	1	65.10
0.2	2	89.07	3	88.13	5	88.77	7	88.90
0.4	3	98.44	5	97.38	9	97.73	13	97.80
0.6	4	100.00	7	99.61	13	99.72	19	99.74
0.8	5	100.00	9	99.96	17	99.98	25	99.98
1.0	6	100.00	11	100.00	21	100.00	31	100.00

There is reasonable agreement between the various NMOL results as the number of sections (smaller Δx) is increased. The slower response of the temperatures within the slab due to the additional convective resistance to heat transfer is evident when the temperatures are compared to those presented in Table 1. Selected temperatures are presented in Figure 3 for the same locations and at the same scale as Figure 2. The delays in the responses of the various temperatures are quite evident.

Problem Extensions

There are a number of extensions to this problem that can be solved with the Numerical Method of Lines. The thermal

conductivity of the solid could vary with the local temperature. There could be an initial temperature profile in the solid. Radiative heat transfer to the surface could be considered in addition to the convection. The convective heat transfer coefficient could be a function of the ΔT between the bulk gas and the slab surface. All these possibilities and more can be solved with the NMOL and an ODE solver such as POLYMATH. This type of problem can be used to effectively introduce undergraduate students to transient heat transfer and instruct them to the numerical solution of partial differential equations – a subject area that is not normally considered in the typical curriculum.

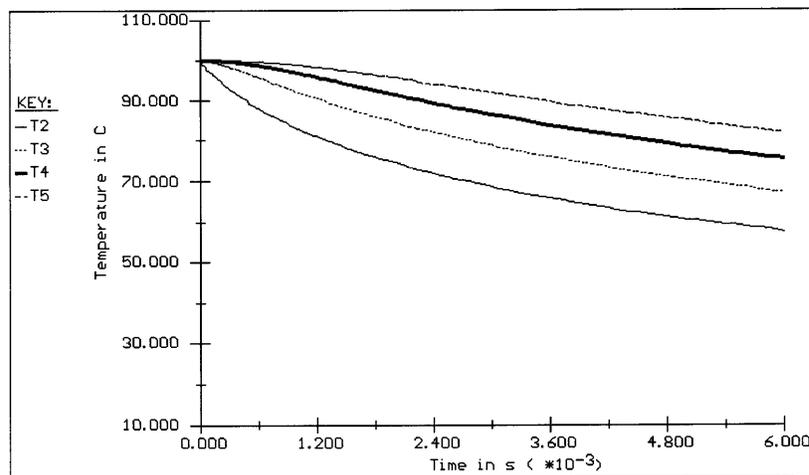


Figure 3. Temperature Profiles for Unsteady-state Heat Transfer with Convection in a One-dimensional Slab