

Efficient Integration of Numerical Problem Solving Throughout the Chemical Engineering Curriculum

Michael B. Cutlip

Department of Chemical, Materials and Biomolecular Engineering, University of Connecticut,
Storrs, CT 06269, USA

Mordechai Shacham

Department of Chemical Engineering, Ben-Gurion University of the Negev,
Beer-Sheva 84105, Israel

INTRODUCTION

The Chemical Engineering curriculum presents formidable challenges to departmental faculty that require considerable efficiency in the organization of course content and careful integration of problem solving tools that are made available to the student. This paper focuses upon the introduction and integration of numerical problem solving throughout the curriculum so that the graduating student will have experience with a combination of widely-used calculational tools in the solution of representative numerical problems within the core courses.

Students need to gain experience and expertise in the mathematical modeling of physical phenomena as this is a very important component of general problem solving, scientific research, and engineering design. The models may take forms of systems of nonlinear algebraic equations (NLEs), systems ordinary (ODEs) or partial differential equations (PDEs) that require analytical and/or numerical solution techniques for simulating the behavior of the physical phenomena. It is also very important to provide students with the ability to correlate data by linear, polynomial, multiple linear, and nonlinear regressions with appropriate statistical interpretation of the results. In recent years the importance of numerical solution techniques has increased considerably because of the widespread availability of the user friendly mathematical software packages.

DEPARTMENTAL OPTIONS FOR TEACHING COMPUTER USE

Many of the challenges to academic chemical engineering departments have been summarized by Edgar⁽¹⁾. Several departmental approaches are available through the CACHE Corporation*, CACHE⁽²⁻⁵⁾. Some philosophical dilemmas that departments may consider are given in Table 1.

Table 1 – Philosophical Challenges

1. To teach or not to teach a programming language. If to teach a programming language, then what to teach?
2. To provide mathematical software or not to provide mathematical software to students for their personal computers?
3. To require or provide an elective course in numerical analysis?
4. To integrate numerical problem solving throughout the curriculum?

* The CACHE Corporation, which stands for Computer Aids for Chemical Engineering, is a not-for-profit organization whose purpose is to promote cooperation among universities, industry and government in the development and distribution of computer-related and/or technology-based educational aids for the chemical engineering profession. <http://www.cache.org>

There has been substantial growth in the number of general purpose and also dedicated software packages that can be used by academic departments and by practicing professional chemical engineering. Faculty members have their own backgrounds and preferences that influence their particular computing preferences as indicated in Table 2.

Table 2 – Typical Faculty Computing Preferences

Prof. A – Likes MATLAB because he/she has used it since grad school and applies advance use to his/her research program in optimal control.

Prof. B – Doesn't like programming languages since he/she used FORTRAN on a CDC 6600 in 1978. Desires something that is easy-to-use.

Prof. C – Wants to use EXCEL since he/she has always made his course grade calculation with his spreadsheet.

Prof. D – Has learned to prefer POLYMATH because the FOGLER textbook in Chemical Reaction Engineering uses it, and it is found on a CD-ROM with the book.

Prof. E – Likes C++ since he/she learned this language when teaching the general engineering computing course to first-year students.

This choice of mathematical software and/or programming language(s) can be difficult for academic departments. A CACHE Corporation survey⁽⁶⁾ in 2003 of computer use by 293 professional chemical engineers with various years of experience in a wide variety of positions can provide some guidance in this area. Some 90% of the engineers surveyed spent more than 25% of their days at a computer with 70% using their computer for at least one-half of their day. These are substantial increases from a similar survey in 1997. Interesting survey results are summarized in Table 3 which indicated extensive use of Excel and modest use of MATLAB.

Table 3 – CACHE Survey Summary

98.3% Use Spreadsheets with Microsoft Excel being Most Popular

Data Analysis	88.2% (253)
Numerical Analysis	47.4% (136)
Material Balances	25.1% (72)
Economic Studies	23.7% (68)
Other	16.7% (48)

90.3% Use Presentation Graphics Software with PowerPoint being the Most Popular

26.9% Used Dedicated Statistical Software with JMP being the Most Popular

25.5% Used Numerical Analysis Software with MATLAB being the Most Popular

65.6% Used Database Management Systems with Access being the Most Popular

10.1% Used Symbolic and Mathematical Manipulation Software with Mathematica being the Most Popular

41.8% Used Process Simulation Programs with Aspen+ being the Most Popular

38.2% Used Computer Programming with Visual Basic being the Most Popular

A growing sense among academics is that a programming language is probably gone as a key tool for all students. A common student's software toolbox might consist of the following packages:

1. POLYMATH
2. Excel
3. MATLAB
4. Aspen Plus or HYSIS or?

There are programming aspects of the first three packages. Excel also provides access to Visual Basic which is widely used in industry. POLYMATH is commonly used in academic departments as about 110 departments provide this software under site licenses in their computer labs and allow all faculty, staff, and students to install the software on personal computers. MATLAB provides advanced computing and programming capabilities, and thus is used in both academia and industry. Aspen Plus or other process simulation software sees educational and industrial use in the general areas of process design and process optimization.

Integrating the Use of POLYMATH, Excel, and MATLAB

The latest educational site-license versions of POLYMATH, provided by the CACHE Corporation at very low costs, enable unique connection with Excel and MATLAB software. This is shown in Figure 1 where the mathematical problem is entered into Polymath which has user-friendly software to perform the basic numerical capabilities. POLYMATH capabilities are summarized in Table 4. Problems are easily entered with a full-screen editor, and templates are provided for entry of nonlinear equations and differential equations. Solutions are normally obtained with proven and robust algorithms. A variety of outputs are selectable. For many problems, such as stiff differential equations, a selection of algorithms is provided.

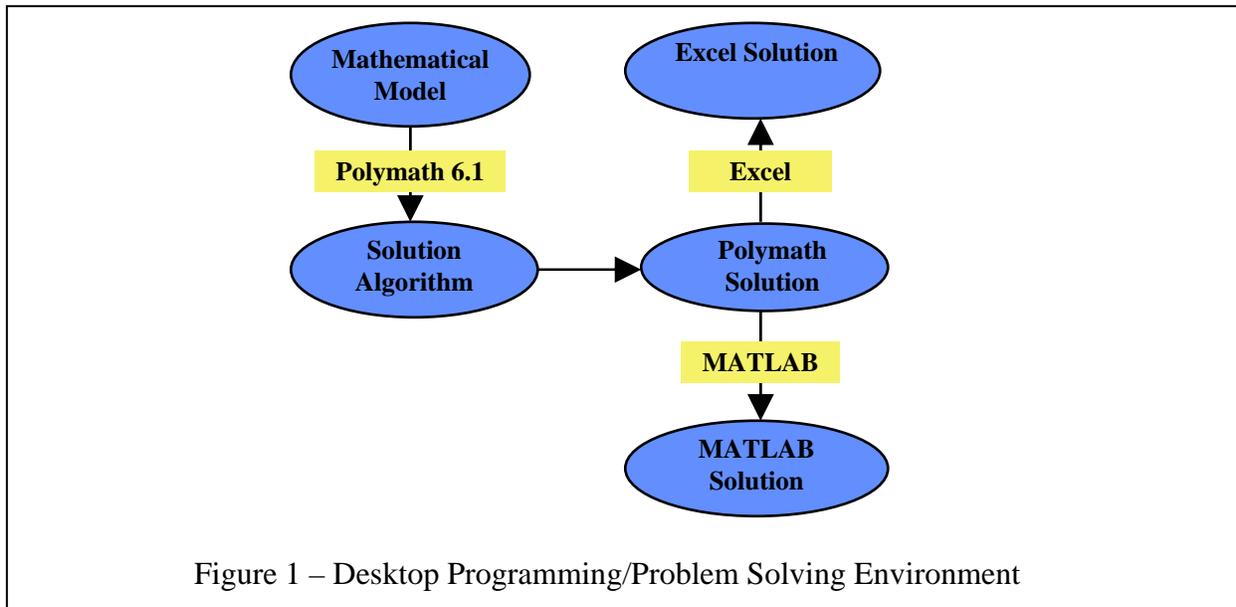
Table 4 – POLYMATH Educational Version

- Linear Equations - up to 100 simultaneous equations.
- Nonlinear Equations - up to 30 simultaneous nonlinear and 40 explicit algebraic equations
- Differential Equations - up to 30 simultaneous ordinary differential and 40 explicit algebraic equations
- Data analysis and Regression - up to 300 data points with capabilities for linear, multiple linear, and nonlinear regressions
- Automatic Creation of Excel Spreadsheet Solution for Problem (include unique ODE_Solver Add-In for solving simultaneous first-order differential equations)
- Generation of Ordered MATLAB Code for Problem

Note that the latest POLYMATH software provides options to work with Excel by automatically providing the export of the problem as a well-formulated spreadsheet. During this export, the intrinsic functions and logical statements are automatically translated to run in Excel. Often the Excel spreadsheet that is created automatically solves the problem after the export. In some

cases, the Excel “solver” Add-In must be used. POLYMATH provides a separate Add-In for Excel that allows simultaneous differential equations to be easily solved within Excel.

The proposed programming and problem solving environment utilizing POLYMATH, Excel, and MATLAB is shown in Figure 1. This environment can be provided in a computer laboratory or to students and faculty for use on their own personal computers. The CACHE site license for POLYMATH provides this software at very modest costs. The current pricing for academic departments is \$200 for the first year and \$150 for subsequent years. This is less for CACHE supporting departments. This site license allows distribution of POLYMATH to all students, faculty, and staff plus use in all computer labs. EXCEL is widely available as a component of the Microsoft Office software. MATLAB provides inexpensive site licenses to most colleges and universities and also markets student versions.



POLYMATH, Excel, and MATLAB (PEM) Problem Solving Environment

The PEM environment involves problem solving that begins with POLYMATH. This approach thus starts with an easy-to-use software package that student enjoy using. The same problem can be solved with Excel and MATLAB. This keeps the students from getting discouraged as they can easily create working problems. A set of five examples, summarized in Table 6, is available from the following web site:

<http://www.polymath-software.com/PEM>

These problems and their associated files in POLYMATH, Excel, and MATLAB are available so that the solutions can be demonstrated via the internet on personal computers provided that all three software packages are available for use on a personal computer. The pdf file will execute directly from the website to expedite the solution of these sample problems.

Consider Example Problem 4. This represents a simple model of a well-mixed batch reactor that is described by a system of ordinary differential equations. The batch reactor initially has only

Table 6 – Representative Examples of the PEM Equation Solving System

1. Linear Equations – Material Balances for Distillation Columns – Polymath
2. Explicit Calculations – Equation of State – Polymath and Excel
3. Nonlinear Equations – Pressure Drop for Pipe Flow – Polymath and Excel
4. Differential Equations – Series Reactions in a Batch Reactor - Polymath, Excel, and MATLAB
5. Regressions (Linear and Nonlinear) – Vapor Pressure Data - Polymath, Excel

reactant A. The reactions are first order and irreversible. This example will be solved with all three software packages.



The differential mass balances on a batch reactor yield

$$\begin{aligned} \frac{dC_A}{dt} &= -k_1 C_A & \text{I. C. } C_A|_{t=0} &= 1 \\ \frac{dC_B}{dt} &= k_1 C_A - k_2 C_B & \text{I. C. } C_B|_{t=0} &= 0 \\ \frac{dC_C}{dt} &= k_2 C_B & \text{I. C. } C_C|_{t=0} &= 0 \\ k_1 &= 2 & k_2 &= 3 \end{aligned}$$

POLYMATH Solution – A Brief Look

Entry of this sample problem is accomplished within the POLYMATH full screen editor as shown in Figure 2. Note that entry is facilitated for new users by use of templates for the input of differential equations or explicit algebraic equations.

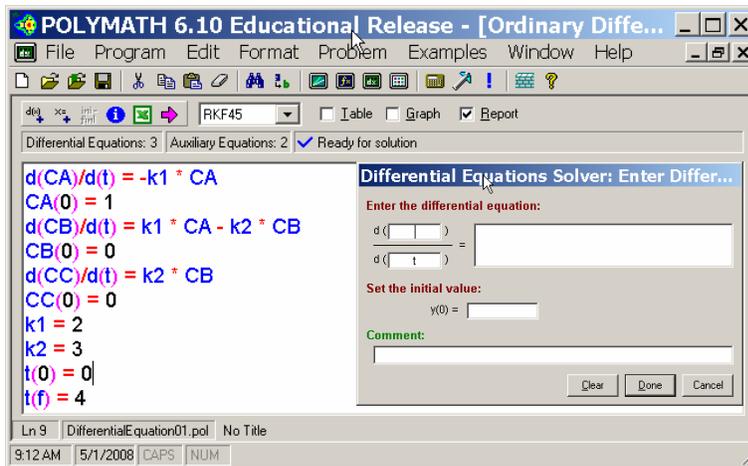


Figure 2 – Entry of Differential Equations into POLYMATH

Selection of the desired integration algorithm and the desired output followed by a mouse click on the highlighted arrow yields the numerical solution. The graph output option gives the expected result shown in Figure 3. Additional options for a Table of results and for a Report of the problem solution are not shown.

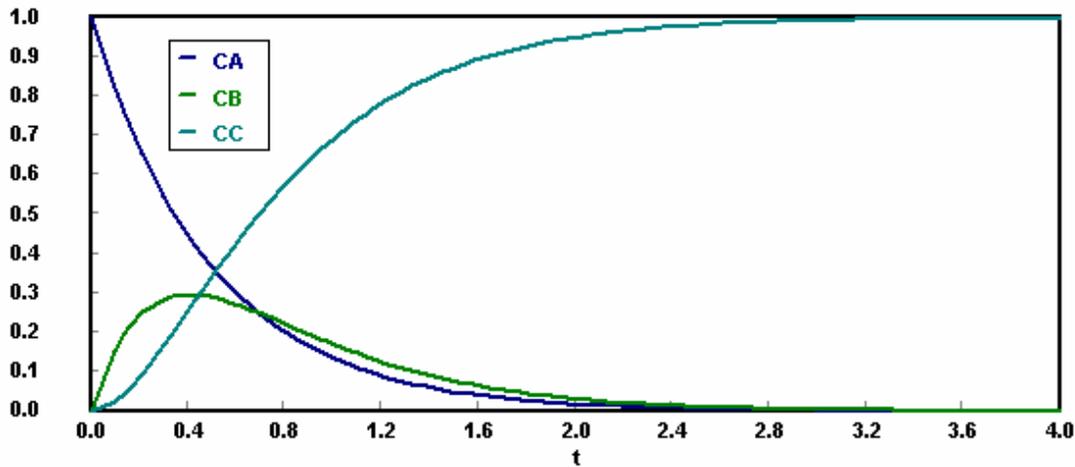


Figure 3 – Graph Output of the Differential Variables from POLYMATH

EXCEL Solution – A Brief Look

The Excel solution for this sample problem is achieved by clicking the Excel icon from the POLYMATH editor. This then automatically exports the problem to an open Excel spreadsheet as shown in Figure 4. Note that all intrinsic functions and logical variables are automatically converted and placed in the Excel spreadsheet. The POLYMATH variable names and equations are shown for the convenience of the user. Since Excel does not have a differential equation solver, an Add-In, ODE_Solver.xla, is provided along with POLYMATH that will solve simultaneous ordinary differential equations. This additional software runs completely independently of POLYMATH.

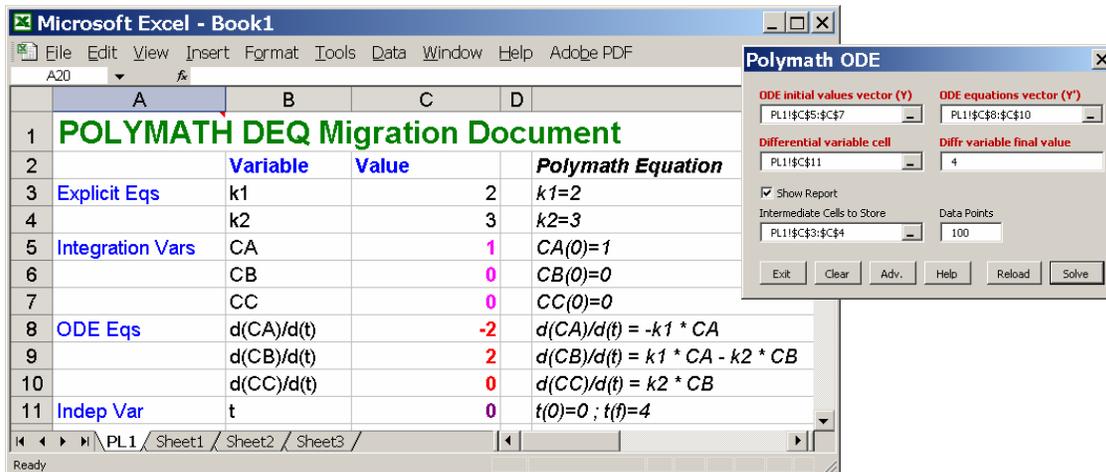


Figure 4 – Excel Spreadsheet with POLYMATH ODE_Solver Add-In

The Excel solution is automatically generated in a new worksheet as shown in Figure 5. Note that a Table is given that summarizes the solution over the integration interval. Also output of the desired variables by tabulated column give the user options for use or plotting within Excel.

POLYMATH Report DEQ						
Ordinary Differential Equations (RK45).						
Calculated values of DEQ variables						
	Variable	Initial	Minimal	Maximal	Final	
1	t	0	0	4	4	
2	CA	1	0.000335	1	0.000335	
3	CB	0	0	0.296062	0.000659	
4	CC	0	0	0.999006	0.999006	
5	k1	2	2	2	2	
6	k2	3	3	3	3	
Intermediate data points						
	t	CA	CB	CC	k1	k2
1		0	1	0	0	2
2	0.082463	0.847957	0.134239	0.017804	2	3
3	0.133428	0.765783	0.191308	0.04291	2	3

Figure 5 – Partial Solution Worksheet Generated During Sample Problem Solution

MATLAB Solution – A Brief Look

The MATLAB solution for this sample problem is obtained by setting an option within POLYMATH that will output the problem code for MATLAB to utilize. This option leads to code which contains the translation of the intrinsic function and logical from POLYMATH into MATLAB code automatically. The generated code for this sample problem is shown in Figure 6 as is provided at the end of the POLYMATH program report.

```

Matlab formatted problem
tspan = [0 4]; % Range for the independent variable
y0 = [1.; 0; 0]; % Initial values for the dependent variables
function dYfuncvecdt = ODEfun(t,Yfuncvec);
CA = Yfuncvec(1);
CB = Yfuncvec(2);
CC = Yfuncvec(3);
k1 = 2;
k2 = 3;
dCAdt = 0 - (k1 * CA);
dCBdt = k1 * CA - (k2 * CB);
dCCdt = k2 * CB;
dYfuncvecdt = [dCAdt; dCBdt; dCCdt];
  
```

Figure 6 – POLYMATH Generated MATLAB Code

Incorporation of the generated code into the MATLAB Template for simultaneous ordinary differential equation gives the m-file given in Figure 7 where the yellow bars indicate the placement of the code generated by POLYMATH.

```

1 function MATLAB01
2 clear, clc, format short g, format compact
3 tspan = [0 4.]; % Range for the independent variable
4 y0 = [1.; 0; 0]; % Initial values for the dependent variables
5 disp(' Variable values at the initial point ');
6 disp([' t      = ' num2str(tspan(1))]);
7 disp('          y          dy/dt          ');
8 disp([y0 ODEfun(tspan(1),y0)]);
9 [t,y]=ode45(@ODEfun,tspan,y0);
10 for i=1:size(y,2)
11     disp([' Solution for dependent variable y' int2str(i)]);
12     disp(['          t          y' int2str(i)]);
13     disp([t y(:,i)]);
14     plot(t,y(:,i));
15     title([' Plot of dependent variable y' int2str(i)]);
16     xlabel(' Independent variable (t)');
17     ylabel([' Dependent variable y' int2str(i)]);
18     pause
19 end
20 %-----
21 function dYfuncvecdt = ODEfun(t,Yfuncvec);
22 CA = Yfuncvec(1);
23 CB = Yfuncvec(2);
24 CC = Yfuncvec(3);
25 k1 = 2;
26 k2 = 3;
27 dCAdt = 0 - (k1 * CA);
28 dCBdt = k1 * CA - (k2 * CB);
29 dCCdt = k2 * CB;
30 dYfuncvecdt = [dCAdt; dCBdt; dCCdt];

```

Figure 7 – Code for MATLAB m-file

The working m-file gives both tabular results (not shown) and graphical results (partly shown in Figure 8).

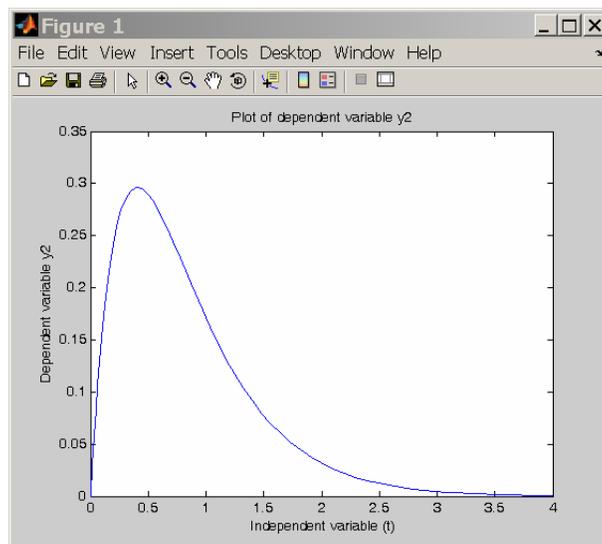


Figure 8 – Partial Graphical Results as Generated by MATLAB for Concentration of B

PROBLEM SOLVING IN CHEMICAL ENGINEERING

Departments of Chemical Engineering can introduce computer use in a variety of ways. A listing of typical courses core courses in chemical engineering is summarized in Table 7.

Table 7 – Typical Courses in Chemical Engineering

Introduction to Computing
Introduction to Chemical Engineering
Thermodynamics
Fluid Mechanics
Heat Transfer
Mass Transfer
Reaction Engineering
Separation Processes
Process Dynamics and Control
Biochemical Engineering
Numerical Methods
Others - Design, Safety, Green Engineering, etc.

One approach is to have the computer-literate faculty introduce various software packages in the core courses that extensively utilize the computer. This might include the Introductory Computing course, the Introduction to Chemical Engineering course, and the Numerical Methods course. This may be a useful approach, but it does not assure that the computing skills that a student acquires will be used throughout the curriculum. It also requires a separate Numerical Methods course that is typically required of all students. This may be difficult with the current needs of the curriculum.

An alternate approach and preferable approach would be to integrate problem solving using the POLYMATH-Excel-MATLAB (PEM) approach throughout the curriculum by introducing elementary concepts of this approach in the Introduction to Computing and Introduction to Chemical Engineering courses. Some typical problems that can be used in the Introduction to Chemical Engineering or first Thermodynamics course are listed in Table 8. This selection is from Cutlip and Shacham⁽⁷⁾ which also encourages the use of the PEM problem solving/computer programming approach.

Careful integration of the various types of problems that involve numerical methods can be introduced into the core courses as is indicated in Table 9. This approach has the desirable qualities that the students are learning to apply computer problem solving at the same time that they are learning the core subject matter within Chemical Engineering. Only a few problems in each course are needed to provide the students with practical experience in numerical problem solving, and they can utilize the variety of software that they will probably rely upon after their graduation from Chemical Engineering. Faculty teaching these courses can rely on their own problems or can select problems from textbooks such as Cutlip and Shacham⁽⁷⁾.

Table 8 – Introductory Computational Problems in Chemical Engineering

1	MOLAR VOLUME AND COMPRESSIBILITY FACTOR FROM VAN DER WAALS EQUATION
2	MOLAR VOLUME AND COMPRESSIBILITY FACTOR FROM REDLICH-KWONG EQUATION
3	STOICHIOMETRIC CALCULATIONS FOR BIOLOGICAL REACTIONS
4	STEADY-STATE MATERIAL BALANCES ON A SEPARATION TRAIN
5	FITTING POLYNOMIALS AND CORRELATION EQUATIONS TO VAPOR PRESSURE DATA
6	VAPOR PRESSURE CORRELATIONS FOR SULFUR COMPOUNDS IN PETROLEUM
7	MEAN HEAT CAPACITY OF n-PROPANE
8	VAPOR PRESSURE CORRELATION BY CLAPEYRON AND ANTOINE EQUATIONS
9	GAS VOLUME CALCULATIONS USING VARIOUS EQUATIONS OF STATE
10	BUBBLE POINT CALCULATION FOR AN IDEAL BINARY MIXTURE
11	DEW POINT CALCULATION FOR AN IDEAL BINARY MIXTURE
12	BUBBLE POINT AND DEW POINT FOR AN IDEAL MULTICOMPONENT MIXTURE
13	ADIABATIC FLAME TEMPERATURE IN COMBUSTION
14	UNSTEADY-STATE MIXING IN A TANK
15	UNSTEADY-STATE MIXING IN A SERIES OF TANKS
16	HEAT EXCHANGE IN A SERIES OF TANKS

Table 9 – Introductory Computational Problems in Chemical Engineering

Introduction to Chemical Engineering Course – EEs, LEs, NLEs, R (L)
Thermodynamics – EEs, LEs, NLEs, ODEs , Integration of Data, Differentiation of Data, R (NL)
Fluid Mechanics - NLE with Logical Variables, NLEs, ODEs with split boundary conditions, R (L, ML, NL), DAEs, PDEs
Heat Transfer - NLEs, ODEs, R (ML, NL), DAEs, PDEs
Mass Transfer - NLEs, ODEs, R (ML, NL), DAEs, PDEs
Reaction Engineering - All
Separation Processes - All
Process Dynamics and Control - All
Biochemical Engineering – All (Parameters in Systems of ODEs)
Numerical Methods - All

Explicit Equations – EEs

Nonlinear Equations – NLEs

Regression – R (Linear-L, Multiple Linear-ML, Nonlinear-NL)

Differential Algebraic Eqns. – DAEs

Linear Equations – LEs

Differential Equations – ODEs

Partial Differential Eqns. – PDEs

CONCLUSIONS

- I. The most recent CACHE survey indicates increasing daily use of personal computer by chemical engineering professions with extensive use of Excel and modest use of MATLAB.
- II. Appropriate use of mathematical software for equation-based problem solving can be accomplished by integration of content and assigned problems throughout the core curriculum.
- III. The latest POLYMATH effectively combines the use of POLYMATH, Excel, and MATLAB and can be used to introduce computer problem solving on personal computers within the Chemical Engineering curriculum.

REFERENCES

1. Edgar, T. F., "Enhancing the undergraduate computing experience," *Chemical Engineering Education*, 40, 231-238 (2006).
2. CACHE Corporation, "Computing Through the Curriculum: An Integrated Approach for Chemical Engineering," <http://www.cache.org/Edgar-computing%20through.html>
3. CACHE Corporation, "Ben Gurion University – Integration of Computing," <http://www.che.utexas.edu/cache/Appendix%20B-2%20Ben.pdf>
4. CACHE Corporation, "University of Colorado – Integration of Computing," <http://www.che.utexas.edu/cache/Appendix%20B-3%20Colorado.pdf>
5. CACHE Corporation, "University of Texas – Integration of Computing," <http://www.che.utexas.edu/cache/Appendix%20B-1%20UT.pdf>
6. CACHE Corporation, "How Recent ChE Graduates Use Computing," (2003). <http://www.che.utexas.edu/cache/Appendix%20A-survey.ppt>
7. Cutlip, M. B. and M. Shacham, *Problem Solving in Chemical and Biochemical Engineering with Polymath, Excel and MATLAB*, 2nd Ed, Prentice-Hall, Upper Saddle River, New-Jersey (2008).

AUTHORS

Michael B. Cutlip is an emeritus professor and former chair of the Department of Chemical, Materials, and Biomolecular Engineering at the University of Connecticut. He is a coauthor of POLYMATH. He has served as national programming chair and national chair of the Chemical Engineering Division of the American Society for Engineering Education and was the co-chair of the ASEE Summer School for Chemical Engineering in 2002. His research interests include chemical and electrochemical reaction engineering.

Mordechai Shacham is the Benjamin H. Swig professor of Chemical Processes and a former chair of the Department of Chemical Engineering at the Ben-Gurion University of the Negev in Israel. He is a coauthor of POLYMATH. His research interests include analysis, modeling, regression of data, applied numerical methods, and prediction and consistency analysis of physical properties.