From Numerical Problem Solving to Model-Based Experimentation— INCORPORATING COMPUTER-BASED TOOLS OF VARIOUS SCALES Into the ChE Curriculum

Mordechai Shacham
Ben-Gurion University of the Negev • Beer-Sheva 84105, Israel
Michael B. Cutlip
University of Connecticut • Storrs, CT 06269
Neima Brauner
Tel-Aviv University • Tel-Aviv 69978, Israel

1. INTRODUCTION

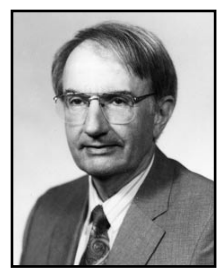
Many of the challenges facing chemical engineering departments regarding the use of computers in undergraduate education were recently reviewed by Edgar.[1] These challenges come about because of the substantial growth in the number of multiple-purpose and dedicated software packages used in the chemical industry, education, and research. At the same time, there remain substantial practical and educational benefits to teaching computer programming using languages such as FORTRAN, Visual Basic, C, or C++. There is obviously not enough room in the undergraduate curriculum to include all the courses required to teach computer programming and all the state-of-the-art software packages currently used in chemical engineering. Thus, it is desirable to construct a general framework that enables sufficient coverage of computer programming as well as the use of multiple purpose and dedicated software packages.

This paper is organized as follows. In sections 2 and 3, the current computing needs in academia and the chemical industry are reviewed. In section 4, the content of a suggested introductory course for modeling and computation for chemical engineers is outlined. It is demonstrated that with a proper choice of software packages several computing-related subjects can be combined in a time-efficient manner, enabling the study of the most important skills in a single course. Section 5 discusses a suggested numerical methods course. Finally, section 6 presents a proposed framework for incorporating computational tools of various scales into the ChE curriculum.

2. COMPUTING IN ACADEMIA

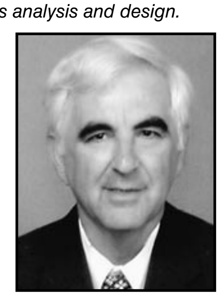
Seader[2] reviewed the education and training needs of chemical engineers related to the use of computers almost 30

__Michael B. Cutlip__ is professor emeritus of the Chemical, Materials and Biomolecular Engineering Department at the University of Connecticut and has served as department head and director of the university's Honors Program. He has B.Ch.E. and M.S. degrees from Ohio State and a Ph.D. from the University of Colorado. His current interests include the development of general software for numerical problem solving and application to chemical and biochemical engineering.



__Neima Brauner__ is a professor in the School of Mechanical Engineering and Heat Transfer at the Tel-Aviv University, Tel Aviv, Israel. She received her B.Sc. and M.Sc. in chemical engineering from the Technion Institute of Technology, Haifa, Israel, and her Ph.D. in mechanical engineering from Tel-Aviv University. Her research interests include hydrodynamics and transport phenomena in two-phase flow systems, and development of interactive statistical and numerical methods for data analysis in process analysis and design.

__Mordechai Shacham__ is the Benjamin H. Swig professor and Head of the Department of Chemical Engineering at the Ben-Gurion University of the Negev in Israel. He also serves as the chairman of the Israeli Inter-University Center for e-Learning (IUCEL). He received his B.Sc. and D.Sc. degrees from the Technion, Israel Institute of Technology. His research interest includes analysis, modeling and regression of data, applied numerical methods and prediction and consistency analysis of physical properties.



The study of computer languages such as Fortran has been included in the ChE curriculum since the 1960s.

years ago. Since then, this field has expanded considerably. Now it includes the study of computer languages, problem solving using numerical and statistical methods, process simulators, computational fluid dynamics (CFD), virtual laboratory experiments, process and product design, and molecular modeling.[1] A more detailed description of some of these issues follows.

2.1 Study of Computer Programming Languages

The study of computer languages such as Fortran has been included in the ChE curriculum since the 1960s. This was enabled by the publication of the book by Lapidus[3] on "Digital Computation for Chemical Engineers," and textbooks containing Fortran programs, such as *Material and Energy Balances* by Henley and Rosen[4] and *Applied Numerical Methods* by Carnahan, *et al*.[5] Programming languages have been studied in "Computer Science" courses. These have varied over the years and have included Fortran, PL/1, Pascal, C, C++, MATLAB (a registered trademark of The Math Works, Inc., <http://www.mathworks.com>), and Visual Basic for Applications (VBA, a registered trademark of Microsoft Corporation, <http://www.microsoft.com>). In the early days of computing, the study of computer languages was essential to enable numerical solution of engineering problems. Upon the introduction of the mathematical software packages, however, (*e.g.*, spreadsheets, Mathcad, a registered trademark of Mathsoft, Inc., <http://www.mathsoft.com>, and POLYMATH, a trademark of Polymath Software <http://www.polymath-software.com > in the mid '80s, the practical importance of computer languages has somewhat diminished. This trend is also reflected in the small percentage of practicing engineers who use programming languages and numerical libraries in their work as was found in a recent survey.[1] Consequently, there is an ongoing debate whether it is still justified to teach programming languages and how many student credit hours should be allocated to this subject area.

Programming languages are often taught by computer scientists not engineers, and this is usually before the students encounter any engineering problems that are complex enough to require programming. This may lead to low motivation among engineering students to study programming. As "programming is unforgiving for ambiguities and errors" (Edgar[11]), many students may forgo their capability to master programming, and some may rely on cheating to get their homework assignments and projects done.[6] Whatever the reasons, there is a very low level of source code programming conducted by practicing engineers.

2.2 Numerical Problem Solving and Visualization

With the introduction of user-friendly mathematical software packages, numerical solution techniques have gradually replaced analytical and graphical solution techniques in engineering education and practice. Fogler[7] in the second edition of his widely used textbook *The Elements of Chemical Reaction Engineering*, replaced many of the analytical and graphical solutions that were included in the first edition[8] by numerical solution obtained via the POLYMATH software package. In 1998 a group of educators presented a set of 10 representative chemical engineering problems[9] and demonstrated that all the problems could be solved by various software packages, including Excel (a registered trademark of Microsoft Corporation, <http://www.microsoft.com>), Maple (a trademark of Waterloo Maple, Inc., <http://maplesoft.com>), MathCAD, MATLAB, Mathematica (a registered trademark of Wolfram Research, Inc., <http://www.wolfram.com>) and POLYMATH. A comparison of the performance of the various packages in solving the set of 10 problems was reported by Shacham and Cutlip.[10] A textbook demonstrating the use of POLYMATH for numerical solution of problems in various required chemical engineering courses was published by Cutlip and Shacham.[11] Currently there are many textbooks that rely on one or more mathematical software packages to numerically solve the presented problems. (See Edgar[1] for a list of such textbooks).

Most of the problems that are included in the textbooks and the publications mentioned in the previous section can be characterized as Single-Model, Single-Algorithm (SMSA). Typical examples of SMSA type problems include the following:

1. *Steady state operation of a tubular reactor, where the model consists of a system of ordinary differential equations and explicit algebraic equations. One numerical integration algorithm (such as the 4th order Runge-Kutta) can be used to solve this model.*
2. *Calculation of the 3-phase bubble-point temperature for a nonideal liquid mixture, where the model includes a system of implicit and explicit algebraic equations. A nonlinear equation solver algorithm (such as the Newton-Raphson technique) can be used to solve this problem.*
3. *Fitting the Wagner equation to vapor pressure data using a linear regression algorithm.*

These types of problems can be solved efficiently by several software packages, as was shown by Cutlip, *et al*.[9] Even in undergraduate education, however, there is often a need to solve more complex problems that can be characterized as: Multiple-Model, Single-Algorithm (MMSA), Single-Model, Multiple-Algorithm (SMMA), and Multiple-Model, Multiple Algorithm (MMMA) problems.

A typical example of a MMSA problem is the cyclic operation of a semi-batch bioreactor.[12] The three modes of operation of the bioreactor (initialization, processing, and harvesting) are represented by different models comprised of simultaneous ordinary differential equations and explicit algebraic equations. All models can be solved by one numerical integration algorithm (such as the 4th order Runge-Kutta).

An example of an SMMA problem is the problem of parameter estimation in dynamic systems. In this case there is a model comprised of ordinary differential equations and explicit algebraic equations, with parameters that can be fitted to experimental data using nonlinear regression techniques. One solution option is to solve this system by integrating the differential equations with specified parameter values in an internal loop, and then minimizing the sum of squares of the difference between the calculated and the experimental values using an optimization algorithm in an outer loop. Additional examples for SMMA problems include a) the solution of two-point boundary value problems, where the integration of the model is carried out in the inside loop and a nonlinear equation solver algorithm adjusts the boundary values in an outer loop, and b) the solution of differential-algebraic systems of equations where the same algorithms are used, but in an opposite hierarchy.

A typical example of an MMMA problem is the optimization of the semi-batch bioreactor, described earlier, with respect to some of its operational parameters. Another example is the modeling of an exothermic batch reactor, where the two stages of operation (heating and cooling) require different models and different integration algorithms (stiff and non-stiff).

The use of visualization, based on graphical solution techniques (such as the McCabe-Thiele diagram) for pedagogical purposes, has seen renewed application recently.[13, 14] The creation of the diagrams needed for visualization can also be characterized as a complex problem that cannot be easily solved with many of the mathematical software packages.

2.3 Large-Scale Simulation

The most commonly used large-scale simulation software packages in undergraduate education include process simulators,[15-17] computational fluid dynamic (CFD) packages, virtual laboratory experiments,[18] and molecular modeling related programs.[1]

When the large-scale simulation tools are employed, the student is usually not required to model the physical phenomena. Therefore, pedagogical drawbacks are often associated with their use in the undergraduate curriculum. Typical claims for large-scale simulators include "they enable black-box modeling" and "it is possible for the students to successfully construct and use models without really understanding the physical phenomena," as noted by Dahm, *et al*.[15] Those drawbacks are only relevant, however, when the use of large-scale simulation programs completely replaces numerical problem solving in the curriculum. There are many potential applications for large-scale simulation programs that cannot be carried out by the general-purpose mathematical software packages. Such applications include visualization of flow fields using CFD software, investigation of cause-effect relationships among operational parameters of a particular process, and the simulation of virtual laboratory experiments. Thus large-scale simulation tools enable students to experience complex systems that may be difficult to attain through direct contact with the equipment itself.[17]

3. COMPUTING IN INDUSTRY

Surveys concerning the use of computer-based tools in the industry were carried out recently by the CACHE Corporation (Edgar[1]) and by Cameron and Ingram.[19] These surveys suggest that engineers and scientists in industry can be separated into two groups: those whose main task is modeling (modelers) and those whose tasks do not include modeling. In the CACHE survey, there was no differentiation between the two groups, while the second survey[19] included only the modelers group. The CACHE survey found that practically all the engineers and scientist in the industry (98%) use spreadsheet programs (the most popular being Excel). Spreadsheet programs are used mainly for data analysis (88%), numerical analysis (47%), material balances (25%), and economic studies (24%). The survey indicated a considerable level of use of database management systems (65%). The level of use of other software tools among the general population of industrial practitioners is much lower, and most of it probably represents their use by the modelers group.

Cameron and Ingram[19] list the tools used by the modelers group according to the extent of their use, as follows: Excel, flow sheeting packages, MATLAB, direct coding, CFD, hybrid modeling, and simulation with optimization programs. Additional tools such as molecular simulation, expert systems, and programs for risk analysis are used to a lesser extent.

4. AN INTRODUCTORY COURSE FOR MODELING AND COMPUTATION FOR CHEMICAL ENGINEERS

A review of the state of the art of computing in academia and in industry has demonstrated that incorporation of the most necessary computing tools into the undergraduate curriculum represents a major challenge. To meet this challenge, it is necessary to provide the students the ability to solve problems of various levels of complexity in a single course.

One possible approach uses the software packages POLYMATH, Excel, and MATLAB in such a course.[20, 21] POLYMATH is an easy-to-learn and user-friendly problem-solving tool, which can be employed in most undergraduate and graduate courses for solving SMSA problems and carrying out various types of regressions with statistical analysis. Excel is included in the introductory computing course because of its widespread use in the industry, suitability for carrying out parametric studies, and connection with programming through VBA (Visual Basic for Applications). MATLAB can be used as a means to learn and apply programming, carry out symbolic manipulations, solve various types of MMMA problems, and provide 2D and 3D visualizations. MATLAB should be recognized as a programming language in terms of the logic skills it requires. There is little difference between MATLAB and the older programming languages, such as FORTRAN, in this respect, as demonstrated for example by Shacham, *et al*.[22]

A convenient new feature included in the POLYMATH package enables the automated export of POLYMATH models to Excel and MATLAB. This capability can significantly shorten the learning curve for these programs. After defining and checking a particular model with POLYMATH, it can be exported to Excel as a complete worksheet, or to MATLAB as a function m-file. The exported models facilitate the introduction and use of the other software tools and help to remove the "unforgiveness" barrier, which prevents many students from attaining programming proficiency. Advanced programming capabilities are provided with MATLAB.

A new introductory computing course replaced the FORTRAN programming course at the Ben-Gurion University in 2003.[20] The course is either given to freshman chemical engineering students who have already taken an introductory Material and Energy Balance course, or the two courses are given concurrently. Realistic problems, which are simple enough to be understood at the early stage of the ChE studies, are extensively used. A typical first problem, for example, involves the solution of the Redlich-Kwong equation of state for the compressibility factor that requires determination of all the real roots of a quadratic equation (see problems 4.1 and 5.1, in Cutlip and Shacham[21]). The model of the problem is prepared using POLYMATH and solved for a few temperature values. The results are compared with calculated values obtained from other sources to verify the correctness of the model. The model is then exported to Excel, where the "two input data table" is used for calculating and plotting the compressibility factor and molar volume for a large number of pressure and temperature combinations. The same assignment is carried out using MATLAB, starting by export of the model from POLYMATH to MATLAB. The parametric study and tabular and graphical display of the results require derivation of the algorithm needed for carrying out the parametric study and learning the MATLAB commands associated with the definition of scalars and arrays, flow control, command window control, and math, logical, and graphic functions. Additional examples that have been used in the introductory course are presented in Chapters 2, 4, and 5 of the Cutlip and Shacham[21] textbook.

Further enhancement of the knowledge and capabilities acquired in the introductory course can be achieved through the use of computational tools of various scales in more advanced courses. A framework to achieve this objective is described in section 6.

5. A NUMERICAL METHODS COURSE FOR CHEMICAL ENGINEERS

A numerical methods course, which is taught in most, but not all, chemical engineering departments, can considerably enhance the programming and the numerical problem-solving capabilities of the students. The effectiveness of the course can be increased by introducing a set of interesting problems that keep the students engaged. In this section a brief review is presented of the sources of problems and case studies, applicable to chemical, biochemical, and environmental engineering as well as to process safety analysis, which require numerical solution.

A library of SMSA problems involving solution of nonlinear algebraic equation of various levels of difficulty was presented by Shacham, *et al*.[23] References 9, 12, and 24 through 32 present examples where the mathematical model includes ordinary differential equations (ODE). Most of the problems are of SMSA type, however References 12 and 31 present MMMA examples and Reference 32 presents a two-point boundary value problem, which can be categorized as an SMMA problem. Inadequate error tolerances, use of inappropriate integration algorithms, and careless rounding of model parameter values can lead to erroneous solutions. Examples regarding such situations are presented in References 26 and 30. References 12 and 31 present examples applicable to the biochemical engineering field. Process safety related examples are presented in References 27 and 29. An environmental engineering related example is presented in Reference 28.

An example associated with solution of differential-algebraic equations (DAE) is presented in Reference 33. Various aspects associated with data analysis and regression are demonstrated in References 32 and 34-36. The particular problems demonstrated include examples of collinearity between independent variable, use of inappropriate statistics and plots to assess the quality of the regression model, and the use of insufficient number or redundant regression parameters.

Determination of the number of significant digits used in computations, when rounding the model parameters or in presenting the results, represents a special challenge. Examples associated with these issues are presented in References 23, 26, 30, and 36.

Shacham[37] presents a typical midterm exam that was recently given at the Ben-Gurion University of the Negev in a Mathematical Modeling and Numerical Methods course that involves MATLAB programming. This course is given to third-year ChE students and the duration of the midterm exam is two hours. The exam questions are based on problem 12.3 in the book of Cutlip and Shacham[21] and can be characterized as an SMSA problem. There are two questions in the exam. The first one involves the calculation of the Wilson equation coefficients for a binary system, which includes ethyl alcohol and another randomly assigned organic compound. The Wilson equation represents activity coefficients for nonideal systems and in this question the students should use azeotropic point data to calculate the coefficients. This requires the solution of a system of two nonlinear algebraic equations. The students should specify the mathematical model of the problem, use MATLAB's symbolic manipulation capabilities to derive the partial derivatives of the functions, and solve the problem iteratively using the Newton-Raphson method. All the steps of the solution are implemented in MATLAB programs. The second question involves the calculation of the dew point temperature for the same nonideal binary system that was used in question 1. The method of solution is similar to the solution of question 1, except that in this case there are three simultaneous nonlinear algebraic equations and the partial derivatives (for the Jacobian matrix) are calculated using finite differences.

After finishing the exam the students turn in the exam form, where their individual data are specified, and all the MATLAB files that were used for the solution. The MATLAB programs provide clear and precise documentation of all the solution steps. Thus, the programs are the best means to assess the knowledge level of the student and to grade the exam.

Problems, such as this exam problem, were assigned to students in the past as homework assignments for solution with programming languages such as FORTRAN or PASCAL. Typically two or three weeks were allocated to complete the assignment. The same problems can be solved today in two hours in the tense atmosphere of an exam. This demonstrates the advantages of the new software tools and programming languages and the new approaches presented here for numerical problem solving.

6. A FRAMEWORK FOR INCORPORATING COMPUTATION TOOLS OF VARIOUS SCALES IN THE UNDERGRADUATE CURRICULUM

A proposed framework for integrating computation tools of various scales into the curriculum is shown in Table 1. The

TABLE 1
A Framework for Incorporating Computation Tools of Various Scales in the Undergraduate Curriculum

No.	Course Name	Recommended software and/or database	Purpose	References
1	Introduction to Modeling and Computation	POLYMATH	Solution of SMSA problems, Regression and statistical analysis	Shacham,[20] Cutlip and Shacham[21]
		Excel	Parametric studies, Tabular and graphical presentation of results	
		MATLAB	Study of programming, Parametric studies, Tabular and graphical presentation of results	
2	Material and Energy Balances	Process Simulator	Simple design project	Dahm, *et al*.[15]
		DIPPR and NIST	Reliable physical property data, Units and experimental errors	
3	Thermodynamics	Process Simulator	Selecting the right thermodynamic package for the system, Multiphase equilibrium	Dahm, *et al*.[15]
4	Equilibrium Stage operations	Instructor-prepared MATLAB and Mathematica programs	Visualization of graphical solution techniques for pedagogical purposes	Joo and Choundary[13] Rasteiro, *et al*.[14]
5	Fluid Dynamics & Heat transfer	CFD software	Numerical experimentation, Visualization of the flow phenomena	Edgar[1]
6	Unit Operations Laboratory	Pre-prepared simulation programs	Virtual laboratory experiments complementing "hands on" experiments	Wiesner and Lan[18]
7	Process control, and process control laboratory	MATLAB toolbox & SIMULINK, Dynamic simulation programs	Control theory related exercises. Virtual laboratory experiments complementing "hands on" experiments.	Edgar[1]
8	Molecular Modelling	Molecular simulation programs	Virtual experiments	Edgar[1]
9	Process and Product Design	Commercial simulation, design, and optimization programs	Interactive process and product design and optimization, Validation of the design through the simulation program	Seider, *et al*.[38] Rockstraw[16]

main feature of the new framework is a basic computational course that replaces the traditional computer programming course. This course has been described in section 4.

Further enhancement of the knowledge acquired in the introductory course can be achieved by using the packages in other modeling and computation-oriented courses. These include courses in numerical methods, optimization, process simulation, dynamics and control, and advanced math. The software packages POLYMATH, Excel, and MATLAB can be used throughout the curriculum for solving problems of various complexities (SMSA, MMSA, etc) and for correlation of data via multiple linear, polynomial, and nonlinear regressions. A detailed demonstration is available of the use of various software packages for multi-scale modeling in a problem involving a biokinetic modeling of imperfect mixing in a Chemostat and the optimization of its operation.[31]

In the first chemical engineering course (Material and Energy Balances), it is desirable to introduce physical property databases (such as NIST and DIPPR) to encourage the use of reliable data sources, considerations of the units associated with the various properties, and awareness of the experimental errors associated with their values. Process simulation programs (such as HYSIS or Aspen) can be used for mini-projects as recommended by Dahm, *et al*.[15]

Additional software packages (such as commercial dynamic and steady-state process simulation, optimization, design, CFD, and molecular simulation), as well as instructor-prepared demonstration programs, can be introduced into the various courses of the ChE curriculum as shown in Table 1. In these courses, the objectives are to use the programs for numerical, model-based, and virtual experimentation, analysis of cause-effect relationships in complex systems, and visualization of challenging concepts. The packages can be introduced to the students in a time-efficient and effective way while simultaneously enabling a better understanding of the specific course material.

6. CONCLUSIONS

A review of the state of the art of chemical engineering computing in academia and in industry has demonstrated that incorporating efficient and widely used computing tools into the undergraduate curriculum remains a continuing major challenge to educators. We suggest that a combination of three popular packages can be integrated into a basic computational course that enables the solution of problems of increasing complexity in the educational setting. These same software packages are also widely used by chemical engineering professionals.

The suggested approach is valid for simple SMSA problems to rather complicated MMMA problems. Shacham[20] has shown that the same three software packages can be used for instruction in programming including modeling and parametric studies as well as regression and statistical data analysis. The described combination of these packages also fulfills most of the basic computational needs in the undergraduate chemical engineering curriculum and in engineering practice.

The presented framework also enables and encourages the inclusion of additional software tools and databases within the undergraduate curriculum as part of the regular courses. The proposed framework represents a proper balance between the educational computing necessary for the chemical engineering curriculum and the requisite professional computing capabilities expected of current graduates.

REFERENCES

1. Edgar, T.F., "Enhancing the Undergraduate Computing Experience," *Chem. Eng. Ed.*, **40**, 231 (2006)
2. Seader, J.D., "Education and Training in Chemical Engineering Related to the Use of Computers," *Comp. & Chem. Eng.*, **13**, 377 (1989)
3. Lapidus, L., *Digital Computation for Chemical Engineers*, McGraw-Hill, New York (1962)
4. Henley, E.J., and E.M. Rosen, *Material and Energy Balance Computation*, Wiley, New York (1969)
5. Carnahan, B., H.A. Luther, and J.O. Wilkes, *Applied Numerical Methods*, Wiley, New York (1969)
6. Grose, T.K., "Do I Hear $20?", *Prism* (American Society for Engineering Education), **16**, 19 (2006)
7. Fogler, H.S., *The Elements of Chemical Reaction Engineering*, 2nd Ed., Prentice-Hall, Englewood Cliffs, N.J. (1992)
8. Fogler, H.S., *Elements of Chemical Reaction Engineering*, Prentice-Hall, Englewood Cliffs, N.J. (1986)
9. Cutlip, M.B., J.J. Hwalek, H.E. Nuttall, M. Shacham, J. Brule, J. Widman, T. Han, B. Finlayson, E.M. Rosen, and R. Taylor, "A Collection of 10 Numerical Problems in Chemical Engineering Solved by Various Mathematical Software Packages," *Comp. Appl. Eng. Ed.*, **6**, 169 (1998)
10. Shacham, M., and M.B. Cutlip, "Selecting the Appropriate Numerical Software for a Chemical Engineering Course," *Comp. & Chem. Eng.*, **23**(suppl.), S645 (1999)
11. Cutlip, M.B., and M. Shacham, *Problem Solving in Chemical Engineering With Numerical Methods*, Prentice-Hall, Upper Saddle River, N.J. (1998)
12. Cutlip, M.B., and M. Shacham, "Modular and Multilayer Modeling—Application to Biological Processes," pp. 1019-1024 in V. Plesu and P. S. Agaci (Eds.), *Proceedings of the 17th European Symposium on Computer Aided Process Engineering*, Elsevier, Amsterdam (2007)
13. Joo, Y. L., and D. Choudhary, "Using Visualization and Computation in the Analysis of Separation Processes," *Chem. Eng. Ed.*, **40**, 313 (2006)
14. Rasteiro, M.G., F.P. Bernardo, and P.M. Saraiva, "Using Mathematica to Teach Process Units: A Distillation Case Study," *Chem. Eng. Ed.*, **39**, 116 (2005)
15. Dahm, K.D., R.P. Hesketh, and M.J. Savelski, "Is Process Simulation Used Effectively in ChE Courses?" *Chem. Eng. Ed.*, **36**, 192 (2002)
16. Rockstraw, D.A., "ASPEN Plus in the ChE Curriculum. Suitable Course Content and Teaching Methodology," *Chem. Eng. Ed.*, **39**, 68 (2005)
17. Streicher, S.J., K. West, D.M. Fraser, J.M. Case, and C. Linder, "Learning Through Simulation, Student Engagement," *Chem. Eng. Ed.*, **39**, 288 (2005)
18. Wiesner, T.F., and W. Lan, "Comparison of Student Learning in Physical and Simulated Unit Operations Experiments," *J. Eng. Ed.*, **93**, 195 (2004)
19. Cameron, I.T., and G.D. Ingham, "A Survey of Industrial Process Modeling Across the Product and Process Lifecycle," *Comp. & Chem. Eng.*, **32**, 420 (2008)

20. Shacham, M., "An Introductory Course of Modeling and Computation for Chemical Engineers," *Comp. Appl. Eng. Ed.*, **13**, 137 (2005)
21. 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, N.J. (2008)
22. Shacham, M., N. Brauner, and M.B. Cutlip, "An Exercise for Practicing Programming in the ChE Curriculum--Calculation of Thermodynamic Properties Using the Redlich-Kwong Equation of State," *Chem. Eng. Ed.*, **27**(2), 148 (2003)
23. Shacham, M., N. Brauner, and M.B. Cutlip, "A Web-Based Library for Testing Performance of Numerical Software for Solving Nonlinear Algebraic Equations," *Comp. & Chem. Eng.*, **26**(4-5), 547 (2002)
24. Shacham, M., N. Brauner, and M.B. Cutlip, "Exothermic CSTRs—Just How Stable Are the Multiple Steady States?", *Chem. Eng. Ed.*, **28**(1), 30 (1994)
25. Brauner, N., M. Shacham, and M.B. Cutlip, "Application of an Interactive ODE Simulation Program in Process Control Education," *Chem. Eng. Ed.*, **28**(2), 130 (1994)
26. Brauner, N., M. Shacham, and M.B. Cutlip, "Computational Results: How Reliable Are They? A Systematic Approach to Model Validation," *Chem. Eng. Ed.*, **30**(1), 20 (1996)
27. Shacham, M., N. Brauner, and M.B. Cutlip, "Prediction and Prevention of Chemical Reaction Hazards – Learning by Simulation," *Chem. Eng. Ed.*, **35**(4), 268 (2001)
28. Brenner, A., M. Shacham, and M.B. Cutlip, "Applications of Mathematical Software Packages for Modeling and Simulations in Environmental Engineering Education," *Environmental Modeling & Software*, **20**, 1307-1313 (2005)
29. Eisenberg, S., M. Shacham, and N. Brauner, "Combining HAZOP with Dynamic Simulation—Applications for Safety Education," *J. Loss Prevention in the Process Industries*, **19**, 754 (2006)
30. Shacham, M., N. Brauner, W.R. Ashurst and M.B. Cutlip, "Can I Trust This Software Package?—An Exercise in Validation of Computational Results," *Chem. Eng. Ed.*, **42**(1), 53-59 (2008)
31. Cutlip, M.B., N. Brauner, and M. Shacham, "Biokinetic Modeling of Imperfect Mixing in a Chemostat—an Example of Multiscale Modeling," *Chem. Eng. Ed.*, **43**, 243 (2009)
32. Shacham, M., M.B. Cutlip, and M. Elly, "Live Problem Solving via Computer in the Classroom to Avoid 'Death by PowerPoint,' " *Computer Applications in Eng. Ed.*, **17**, 285 (2009)
33. Shacham, M., and N. Brauner, "What To Do If Relative Volatilities Cannot Be Assumed To Be Constant?—Differential-Algebraic Equation Systems in Undergraduate Education," *Chem. Eng. Ed.*, **31**(2), 86 (1997)
34. Shacham, M., and N. Brauner, "Correlation and Over-correlation of Heterogeneous Reaction Rate Data," *Chem. Eng. Ed.*, **29**(1) 22-25, 45 (1995)
35. Shacham, M., N. Brauner, and M.B. Cutlip, "Replacing the Graph Paper by Interactive Software in Modeling and Analysis of Experimental Data," *Comput. Appl. Eng. Ed.*, **4**(3), 241 (1996)
36. Shacham, M., M.B. Cutlip, and M. Elly, "Common Errors in Numerical Problem Solving—How Can They Be Detected and Prevented," *Chem. Eng. Progress* (Accepted for publication, also available at <http://cee.che.ufl.edu/>)
37. Shacham, M., "Applying New Technologies to the Classroom—What Have We Learned from Past Experience?", Paper 133a, Presented at the AIChE100 Annual Meeting, Philadelphia, PA, Nov. 16-21, 2008 (also available at <http://cee.che.ufl.edu/>)
38. Seider, W.D., J.D. Seader, and D.R. Lewin, *Product and Process Design Principles*, 2nd Ed., Wiley, New York, (2004) ❐