ENHANCING THE UNDERGRADUATE COMPUTING EXPERIENCE

THOMAS F. EDGAR
corresponding author, CACHE Corporation

Many chemical engineering departments are wrestling with the following questions:

- *When should computing be introduced to the chemical engineering student?*
- *How should computer programming in chemical engineering be taught, and how much formal programming instruction on languages such as C should be provided (vs. usage of computing tools such as MATLAB, Mathcad, spreadsheets, etc.)?*
- *Is a numerical methods course required and where is it in the course sequence? How many credit hours are needed? Which department teaches it?*
- *Should every chemical engineering course include some computing?*

Since the mid-'80s two approaches have been taken toward introductory computing for engineers: "CS 101" and the engineering tools approach. The "CS 101" approach was catalyzed by the growth of computer science programs, which provided instruction in computer languages. Over the years the "CS 101" courses have migrated through several programming languages: Pascal, C, C++, and Java. In the engineering branch, software vehicles such as spreadsheets (first Lotus 123, then Quattro Pro, and now Excel), TK Solver, Mathcad, and MATLAB have gradually pushed out programming languages (primarily Fortran). Programming languages are becoming endangered species in these courses.

The "CS 101" branch would claim a number of reasons for existence[1]:

- *engineers should learn fundamental concepts of programming and computer science*
- *computing should be taught by computer scientists, not engineers*
- *engineering faculty are not interested in teaching computing languages to their students*
- *these courses provide a significant number of student credit hours (SCH) and budgetary resources*

There are concrete benefits to an engineering education that incorporates the ability to write computer programs: Students learn what assumptions go into the program, *i.e.*,

- *what the right answer should be*
- *what is the input, what is the output*
- *clear organization of thought, logic, and calculations is required*
- *that errors can exist in a program*
- *that programming is unforgiving for ambiguities and errors*

Thomas F. Edgar *is the Abell chair in the Department of Chemical Engineering at the University of Texas, where he has been a faculty member for 35 years. He is also executive officer of the CACHE Corporation, a nonprofit educational organization that promotes development and distribution of technology-based educational aids for the chemical engineering profession.*

When it comes to assessing computing needs, faculty often confuse what is important for their students with what is important for themselves.

The "Engineering Tools Approach" branch believes:

- *engineering students need a solid grounding in problem solving with modern computing tools*
- *engineering students need the knowledge and tools required in their professions*
- *engineering computing and problem solving are best taught by engineers*
- *there is no room in the curriculum for a separate three- or four-SCH course in programming*

While the two branches are complementary and most engineering students could benefit from both courses, most chemical engineering program curricula are too congested to make room for both.

Many departments no longer require a course in a computer programming language such as Fortran, C, or C++. It has been suggested that teaching computer programming is analogous to teaching plane geometry. It is a way of thinking but you may not have to use it. On the other hand, without some programming ability, engineers are limited by the built-in capability of commercial software without any way to extend it. This has led a number of departments to switch from teaching C++ to using MATLAB as the programming tool. MATLAB is a structured programming language that incorporates many elements of Fortran, C, and C++. It allows for modularity, flow control, and input/output control and has the following programming features.

1. *Loops: like* DO *and* WHILE *in Fortran, MATLAB has* for *and* while
2. *Conditional statements: like* IF *in Fortran, C, and C++; MATLAB has* if *for testing relational operations*
3. *Relational operations: like C and C++, MATLAB has the expected suite of* <, >, <=, >=, = =, ~=. *And like C and C++, the result of the operation is* 1 *or* 0, *and can be used outside of a conditional statement*
4. *Logical operations: like Fortran, C, and C++, relational operations can be strung together with* AND (&), OR (|), *and* NOT (~)
5. *Matching: like C, MATLAB has a switch/case syntax for matching string variables, integers, or logicals*
6. *I/O: not only can a user be prompted for input and then have results output to screen in formatted form (using* fprintf, *as in C), but MATLAB can* read (load) *and* write (save) *to files, in binary or ascii format*
7. *Modularity: like* SUBROUTINE *in Fortran or* function *in C and C++, MATLAB allows the user to create user-defined functions to be called by a main program; any number of inputs and outputs can be associated with a user-defined function*
8. *Error processing: using the* try/catch *syntax a user can attempt calculation(s) and then gracefully continue execution if an error occurs*
9. *Array math: like Fortran90, MATLAB transparently accommodates scalar and array math (i.e., implied* FOR *loops)*

Reasons given by faculty for switching from C++ to MATLAB include ease of use and widespread availability due to an inexpensive student version. Because MATLAB is an interpreted rather than compiled language, the user can create (write), debug, and run code in the same environment. The built-in editor can pass code directly to the MATLAB application for execution. Also, MATLAB has a solid graphical interface for creating 2-D and 3-D plots; and plots can be created using appropriate MATLAB code from within a user's program.

Based on informal surveys many chemical engineering departments now introduce programming and engineering problem solving in the freshman year. The view is that

these subjects are best taught by an engineering department in the context of an application. A typical introductory course has the following outline.

- *problem-solving: engineering method, units, precision in calculations*
- *symbolic computing: algebra, calculus*
- *spreadsheet techniques: solutions to engineering problems, Visual Basic for Application (VBA) in Excel*
- *programming fundamentals: data types, program flow, modularity, object-oriented features*
- *elementary numerical methods: linear and nonlinear equation solving, linear regression*
- *software tools: Mathcad, MATLAB, Excel*

When it comes to assessing computing needs, faculty often confuse what is important for their students with what is important for themselves. Faculty needs, more often than not, align with their research interests and activities, and these may be disconnected from the needs of their undergraduate students. Also, faculty may have an incorrect impression of the computing needs of professionals by either being out of date or out of touch. Discussions on computing needs do not always proceed on the basis of evidence from alumni and employer surveys. Finally, computing is not part of the daily professional existence of most faculty and is not expected to be. Their computing skills can be oxidized, and most of their computing is carried out by their students.

In the area of computing software, there is a noticeable disconnect between industry and academia. The appendix summarizes a survey of computing practices of recent graduates in chemical engineering, most of whom now work in industry. Typically chemical engineering departments teach the use of MATLAB, Mathcad, Mathematica, or Maple but not the use of spreadsheets. Yet in industry, spreadsheet software (*e.g.*, Excel) is the dominant computer package in use. Of course this may reflect the nature of many calculations that need to be performed by chemical engineers in industry, rather than a need to de-emphasize the teaching of sound numerical approaches in universities. Some faculty resist teaching spreadsheets, for example, because it is difficult to analyze the logic in the code, but this appears to be changing with the availability of VBA. Another objection is that a spreadsheet approach can encourage the use of inaccurate or inefficient numerical calculations (no error control, etc.) For complex calculations, it may be better to program spreadsheets using VBA, where programming logic is more transparent.

The survey of industrial usage of computing in the appendix also indicates that less than 50% of recent graduates in chemical engineering actually perform programming on the job (although there is no clear definition of what constitutes "programming" in industry). The use of spreadsheets in chemical engineering practice appears to be increasing. The application of spreadsheets in university courses may be attractive because of student-driven usage. David Clough (U. Colorado) and Brice Carnahan (U. Michigan) have developed many examples of spreadsheet applications, as presented at the 2002 ASEE Chemical Engineering Summer School.

TEXTBOOKS AND AFFILIATED SOFTWARE

The fragmented nature of software tied to leading undergraduate textbooks makes integration of computing through the curriculum difficult, *e.g.*,

(a) *material and energy balances: Felder and Rousseau[2] – EZ Solve; Himmelblau and Riggs[3] – POLYMATH*

(b) *thermodynamics: Sandler[4] – MathCAD; Kyle[5] – POLYMATH; Elliott and Lira[6] – various programs*

(c) *separations: Wankat[7] – Aspen*

(d) *process control: Seborg, Edgar, Mellichamp[8] – MATLAB; Bequette[9] – MATLAB; Riggs[10]– MATLAB/Excel*

(e) *chemical reaction engineering: Fogler[11] – POLYMATH; Ekerdt and Rawlings[12] – Octave/MATLAB*

(f) *product and process design: Seider, Seader, Lewin[13] – Aspen, HYSYS, CHEMCAD, PROII*

In addition to these courses, many departments are teaching a statistics course, which involves still one more software package such as JMP, SAS, or Minitab. Clearly, using a subset of these textbooks sequentially through the sophomore, junior, and senior years will require a student to learn up to five or more different software packages. Adding software packages from outside of chemical engineering can push the total number of packages beyond 10, which becomes problematic for the typical student. It would be desirable to keep the number of software packages below three or four if possible [note that Excel is only mentioned once in (a)-(f) above although it is used with many of the textbooks listed]. But usually textbooks are not chosen because of the bundled software. In addition, departments must address issues of software availability, licensing, cost, and providing software in computer labs vs. student-owned computers. A textbook that is closely coupled to a software package, a CD-ROM, or a Web site is clearly an attractive option.

TEACHING PROCESS SIMULATORS THROUGH THE CURRICULUM

Several departments have found that the difficulty of integration of computing tools mentioned above can be avoided by more extensive use of process simulators. It is quite common to expose students to a commercial simulator in a thermodynamics or separations course. At Virginia Tech, ChE undergraduates have been using Aspen Plus and Aspen Dynamics to solve problems in all subjects, starting in the sophomore year. It is fairly straightforward to convert a steady-state model in Aspen Plus into a dynamic model (with PID control schemes) in Aspen Dynamics. The applicability

of Aspen Plus to mass and energy balances, thermodynamics (physical and thermodynamic property analysis, estimation and regression), multicomponent separations, reactor design, and process flowsheet simulation is well known. In process control Aspen Dynamics enables students to evaluate controller tuning, process dynamics, startup and shutdown, etc. HYSYS has similar features, and has been used at Rowan University for analysis in freshman-senior years.

Recently, Version 2.0 of a CD-ROM, *Using Process Simulators in Chemical Engineering: A Multimedia Guide for the Core Curriculum*,[14] has become available. Modules and tutorials are provided for self-paced instruction in the use of the process simulators to solve open-ended problems in courses on material and energy balances, thermodynamics, heat transfer, reactor design, separations, and product and process design. A 110-page document has been prepared for instructors suggesting the best instruction sequence and providing exercises and solutions, for each of the core courses (first introduced at the 2002 ASEE Chemical Engineering Summer School).

NUMERICAL AND ANALYTICAL APPROACHES IN MODELING OF PHYSICAL BEHAVIOR

Historically many engineering courses have been taught from an analytical viewpoint, but a transition is starting to occur in which numerical experiments are being gradually added in fluid flow or heat transfer courses. Problems and experiments should not be so simplified that they are not realistically formulated. Students are normally exposed to idealized fluid flow cases in the curriculum, for which application of theoretical concepts results in a solution of a one-dimensional ordinary differential equation or an algebraic equation. Therefore it is very easy for them to come away with the notion that theory is useless for most real-life situations.

Students should be able to select either analytical or numerical techniques to solve a problem, hence they should learn the advantages and disadvantages of either approach. Use of more sophisticated numerical tools such as CFD (computational fluid dynamics) will reduce the need to make many simplifying assumptions because you do not need as many assumptions to solve the problem numerically. Chemical engineering students should understand that there are both numerical experiments and physical experiments. In some cases we can make observations from numerical experiments that you cannot see in physical data, but the converse is also true. This does not suggest that all derivations should be replaced by numerical simulation, neither should every experiment be replaced with a simulator. There should, however, be a balance of experimental fluid dynamics (EFD), analytical fluid dynamics (AFD), and CFD.

To prepare students for industrial practice, there should be a department-level re-examination of the role of detailed analytical solutions. Is the purpose of some of these exercises the preparation of undergraduates for graduate school or industry? Today practicing engineers are not expected to carry out complex derivations in project work. Once a fluid flow situation is analyzed theoretically or the governing principles are discussed, that same situation can be visualized using the computer. This visualization of the flow phenomena can significantly facilitate and enhance the learning process, especially for the visual learner. CFD software makes flow visualization easy. Students can simulate flow processes in a transient or steady-state mode. Flow patterns can be displayed via velocity contours, velocity vector plots, or graphs of velocity profiles. A key element in flow visualization exercises is exploring the effects of different parameters. Using CFD, students can quickly change the size of the pipe, viscosity of the fluid, size of the particles, velocity of the feed, etc., and see the resulting changes in the flow behavior. This type of parametric analysis also ties in nicely with a discussion of dimensionless groups and geometric and dynamic similarity.

While computing and visualization can increase understanding, educators do not want students to view such simulators as black boxes. In the fluid mechanics course, simulations can become a mathematical exercise with little intuition, unless the instructor has the students solve a simple problem by hand first. More work on the software tools is needed, and it is critical to match the software tool to the student's knowledge base.

Two specific recent packages that have been developed for educational usage are FlowLab (a finite volume-based code) by Fluent, Inc., and FEMLAB (a finite element-based code) by Comsol, Inc. Based on a survey by Professor Jennifer Curtis at the University of Florida, about 20 departments of chemical engineering in the United States expose their undergraduate students to CFD software. FlowLab allows students to solve fluid dynamics problems without requiring a long training period. Using carefully constructed examples, FlowLab allows students to get started immediately without having to spend the large time commitment to learn geometry and mesh-creation skills required by traditional CFD software. Current exercises that have been developed include sudden expansion in a pipe, flow and heat transfer in a pipe, flow around a cylinder, and flow over a heated plate, among others. In addition, professors can create their own examples or customize the predefined ones.

FEMLAB provides ready-to-use application modes, where the user can build his or her own model by defining the relevant physical quantities rather than the equations directly. The software also allows for equation-based modeling, which gives the user the freedom to create equations. FEMLAB's programming language is an extension of the MATLAB language; this feature gives much flexibility to the user. FEMLAB's graphical interface includes functions for automatic mesh generation of a user-defined geometry. Recently a k-ε turbulence model has been added to its menu of options.

Seider, *et al.*,[13] present the design of configured consumer products, which usually involves 2-D or 3-D simulations. In Chapter 19 (Product Design), momentum and species balances in a 2-D plasma CVD reactor are employed to produce thin Si films using CFD packages such as FEMLAB. This illustrates where it is very effective for students to use CFD packages to optimize designs—even without understanding all of the physical and chemical interactions in the transport-reaction processes.

Even with these recent advances in educational CFD software, this computing technology has been slow to penetrate undergraduate transport and reactor engineering courses. A 2002 CACHE survey of all chemical engineering departments in the United States on barriers to implementing CFD identified a lack of knowledge concerning available CFD resources, a lack of professor training in CFD, the relative difficulty of use and the long learning curve associated with using CFD software in a given course, and cost of CFD software.

VIRTUAL LABORATORY EXPERIMENTS

Laboratory courses are evolving, and new directions are being examined at specific universities, combining elements of simulation and also distance learning. In the chemical process industries, the high cost of pilot-scale equipment and operating manpower has led to more reliance on computer-based simulations rather than traditional pilot-scale experiments. During a typical day, the plant engineer works from a control room, or at least behind a computer screen. An engineer rarely is in the field adjusting valve positions, flow rates, and temperatures, because that is normally done using the computer interfaces of distributed control systems.

The fourth-year unit operations laboratory at Texas Tech University is emulating industrial practice, by providing computer-generated simulations based upon mathematical models for laboratory equipment.[15] The unit operations laboratory can familiarize students with safety concerns and operational issues regarding each piece of equipment. Major pieces of equipment include a double-pipe heat exchanger, an ammonia gas-absorber packed column, and a cooling tower. The Virtual Unit Operations laboratory (VUOL) complements the existing laboratory to give students a realistic experience with industrial operations. LabVIEW computer interfaces of the VUOL permit students to control the equipment in addition to physically turning valves and checking temperatures.

In the Texas Tech course each student operates two physical and two virtual experiments. Based on preliminary assessment data, students reported that this type of laboratory class contributed either a great deal or considerably in all areas of ABET criteria a-k. Virtual and physical experiments complement each other and enhance student learning. In addition, there appears to be no significant difference in the student perception to their learning in using virtual vs. actual unit operations experiments, in 18 out of 20 ABET-related skill areas. While students believe both types of experiments are valuable, a total virtual unit operations laboratory would apparently not be well-received by the students. With the physical portion of the lab, students get a feel for the equipment and how it operates. With the virtual portion, the students become familiar with the computer interfaces that are similar to industrial control rooms, and learn to manipulate the equipment via those controls instead of manually turning valves and knobs. They can also explore operating scenarios which are not easily or economically investigated with physical equipment.

Web-access of laboratory experiments enables real chemical engineering laboratory equipment to be controlled and monitored interactively by computers that are connected to the Internet, *i.e.*, under the command of users over the Web. This capability is now available in the labs at University of Tennessee-Chattanooga as well as other schools

Today practicing engineers are not expected to carry out complex derivations in project work. Once a fluid flow situation is analyzed theoretically or the governing principles are discussed, that same situation can be visualized using the computer.

such as University of Texas-Austin, Columbia University, University of Toledo, and MIT. Such labs permit faculty and students from any university to run Web-connected experiments at any time of the day or night, any day of the week. The laboratory station computer operates the equipment (pumps, valves, heaters, relays, etc.), collects the data (pressure, temperature, position, speed, concentration, etc.) and sends it to the Web user. The University of Tennessee site is accessed through the Web address <http://chem.engr.utc.edu/>, and even includes audio and video of the operating equipment.

All established chemical engineering programs are facing increased financial pressure to keep existing laboratory experiments up to date and in satisfactory operating condition. Major operating costs of unit operations laboratories include maintenance and teaching-assistant support. Using highly automated experiments for remote operations will allow a drastic reduction in TA time requirements for those particular experiments. In addition, by sharing the operation of the experiments among several universities, there can be a pro rata reduction in maintenance costs. There is also the opportunity to add experimental assignments to a lecture class using this technology. In a lecture class, it may be desirable to have students individually or in small groups carry out an experiment, much like a homework assignment; in contrast, a traditional experiment would require continuous supervision by teaching assistants (*e.g.*, one week of TA time for an entire class). Therefore, using an Internet-based experiment can greatly reduce the time commitment by the TA. It is clear that traditional experiments should remain in the curriculum to give students "hands-on" exposure, but they can be augmented with Internet labs.

While computing and visualization can increase understanding, educators do not want students to view such simulators as black boxes.

PROCESS AND PRODUCT DESIGN

Historically there has been a process design emphasis in the curriculum that is now transitioning to a dual product and process design emphasis. This means that a framework is needed to make process decisions to make structured products. This has added a performance layer, *i.e.*, not just purity of the product. Given a structure, we can often predict at some level what the properties of the material are likely to be. The accuracy of the results and the methods used to treat them depend critically on the complexity of the structure as well as the availability of information on similar structures. For example, various quantitative structure property relationship (QSPR) models are available for the prediction of polymer properties. The inverse engineering design problem, however—designing structures given a set of desired properties—is far more difficult. The market may demand or need a new material with a specific set of properties, yet given the properties it is extremely difficult to know which monomers to put together to make a polymer and what molecular weight the polymer should have. Today the inverse design problem is attacked empirically by the synthetic chemist with his/her wealth of knowledge based on intuition and experience. A significant amount of work is already under way to develop the "Holy Grail" of materials design, namely, effective and powerful reverse-engineering software to solve the problem of going backwards from a set of desired properties to the realistic chemical structures and material morphologies that may have these properties. After this is completed, a subsequent step would involve how to manufacture the desired new product.

A chapter on Molecular Structure Design in Seider, *et al.*, [13] contains simple optimization procedures using GAMS to determine polymer repeat units, refrigerants, and solvents that have desired properties using group-contribution methods. Eventually, these will be replaced (and augmented) by molecular models.

Another subject related to product design is the scheduling of batch processes, which can be done using simple simulation techniques, as in BATCH PLUS and SUPERPRO DESIGNER. Hence design of optimal processing can be viewed as "product design" for specialty chemicals. Clearly, spreadsheets and optimization packages can also be used for many of these computations. Finally, the use of large databases and software systems, such as ASPEN IPE, for equipment sizing and purchase and installation cost estimation, is becoming common throughout the chemical industries for product and process design.

MOLECULAR MODELING

A molecular-level understanding of chemical manufacturing processes would greatly aid the development of steady-state and dynamic models of these processes. Process modeling is extensively practiced by the chemical industry to optimize chemical processes. One needs, however, to be able to develop a model of the process and then predict not only thermochemical and thermophysical properties but also accurate rate constants as input data for the process simulation. Another critical set of data needed for the models is thermophysical properties. These properties include such simple quantities as boiling points and also more complex phenomena such as vapor/liquid equilibria phase diagrams, diffusion coefficients, liquid densities, and the prediction of critical points. A key role of computational chemistry is to provide input parameters of increasing accuracy and reliability to the process simulations.

Under the NSF grant, "World Wide Web-Based Modules for Introduction of Molecular Simulation into the Chemical Engineering Curriculum," seven university experts in molecular simulations have developed Web-based modules to facilitate introduction of molecular simulation into the chemical engineering undergraduate curriculum. These teaching modules can be integrated directly into chemical engineering core undergraduate courses, supplying for the instructor and the student the appropriate linkage material between macroscopic concepts currently taught in these courses and molecular simulations designed to aid student understanding of the molecular underpinnings of the phenomena. Modules are centered around Java Applets that run the molecular simulations and provide an "experimental" simulation platform for students to explore concepts. In addition, modules contain instructor materials, fundamental tutorials, student problems, and assessment materials.

A consistent Web-based interface has been designed that organizes all of the material in each module and develops scripts using perl; this eases the job of putting the written material into this common format. The developer of a module must construct simple text files, perhaps with HTML markup that permits inclusion of figures and tables. Then he or she runs the files through the perl script, which adds HTML formatting and links to put the set of files into the common configuration. The files are uploaded to the module site for anyone to access. This site is perhaps best accessed through the Etomica site. Etomica is a Java-based support environment developed for the modules project, which has now been expanded for other applications (<http://www.ccr.buffalo.edu/etomica>, contact is Professor David Kofke).

Following is a list of phenomena and concepts for which modules are completed or planned:

- *Chemical reaction equilibrium*
- *Osmosis*
- *Diffusion*
- *Molecular dynamics*
- *Normal modes of a solid*
- *Chemical reaction kinetics*
- *Dissipative particle dynamics*
- *Surface tension*
- *Crystal viewer*
- *Joule-Thomson expansion*
- *Self-assembly*
- *Chemical potential*
- *Multicomponent phase equilibrium*
- *Heat transfer*
- *Atomic billiards*
- *Viscosity*

CONCLUSIONS AND RECOMMENDATIONS

One way to foster renewal of the curriculum is to identify departments where curriculum revision is being carried out and to evaluate best computing practices and current trends. There may not be one answer because of different constraints under which various universities operate, such as number of faculty in the department and whether computing courses are taught outside the department. Contributions to this article came from nearly 20 universities, so we are aware of local issues.

CACHE makes the following recommendations to enhance computing through the curriculum:

(1) There is increasing pressure on the total number of hours in the curriculum, especially with the addition of life science courses. Departments should continue to re-examine whether a formal three- or four-credit-hour computer programming course is required for the chemical engineering degree (vs. teaching how to use software or write m-files in MATLAB, for example). The chemical engineering computing course also provides students with a valuable experience in quantitative problem solving.

(2) The number of software tools that implement numerical methods used by students should be minimized; departmental agreement on software used in each course should be reached within the faculty. Faculty need to reach consensus on how student computing skills can grow systematically through evaluating each course in the curriculum.

(3) Courses such as transport phenomena and thermodynamics offer new possibilities for introduction of computing physical and chemical behavior, such as with computational fluid dynamics or molecular modeling. Process design can add a product design emphasis by using such tools as well.

(4) Internet-based and virtual laboratories offer a new means of strengthening the student simulation experience in order to reinforce theoretical concepts.

(5) To prepare students to optimize process designs, it helps to expose students to process simulators for solution of a problem(s) in the core courses of the chemical engineering curriculum. Also, as software develops and product design is added at the senior level, instructors must select from among optimization packages (such as GAMS), batch process simulators (such as BATCH PLUS and SUPERPRO DESIGNER), and packages for estimating equipment sizes and installation costs (such as Aspen IPE). The use of comprehensive software packages and databases is common in industrial design and needs to be introduced in design courses and used for solution of design projects.

ACKNOWLEDGMENTS

Contributors to this paper include T.F. Edgar, W.D. Seider, D.E. Clough, J. Curtis, D.S. Dandy, B.L. Knutson, P.R. Westmoreland, J.J. Siirola, Chau-Chyun Chen, G.V. Reklaitis, R. LaRoche, J.B. Rawlings, E.M. Rosen, D. Kofke, M. Cutlip, M.J. Savelski, and M. Shacham.

REFERENCES

1. Clough, D.E., "ChE's Teaching Introductory Computing to ChE Students—A Modern Computing Course with Emphasis on Problem Solving and Programming," ASEE Annual Meeting, Montreal (2002)
2. Felder, R.M., and R.W. Rousseau, *Elementary Principles of Chemical Processes*, 3rd Ed., Wiley, New York (1999)
3. Himmeblau, D.M., and J.B. Riggs, *Basic Principles and Calculations in Chemical Engineering*, 7th Ed., Prentice-Hall, Upper Saddle River, NJ (2003)
4. Sandler, S.I., *Chemical, Biochemical, and Engineering Thermodynamics*, 4th Ed., Wiley, New York (2006)
5. Kyle, B.G., *Chemical and Process Thermodynamics*, 3rd Ed., Prentice-Hall, Upper Saddle River, NJ (1999)
6. Elliott, J.R., and C.T. Lira, *Introductory Chemical Engineering Thermodynamics*, Prentice-Hall, Upper Saddle, NJ (1999)
7. Wankat, P.C., *Equilibrium-Staged Separations*, 2nd Ed., Prentice-Hall, Upper Saddle River, NJ (2006)
8. Seborg, D.E., T.F. Edgar, and D.A. Mellichamp, *Process Dynamics and Control*, 2nd Ed., Wiley, New York (2004)
9. Bequette, B.W., *Process Control*, Prentice-Hall, Upper Saddle River, NJ (2003)
10. Riggs, J.B., *Chemical Process Control*, Fernet Publishing, Lubbock, TX (2001)
11. Fogler, H.S., *Elements of Chemical Reaction Engineering*, 4th Ed., Prentice-Hall, Upper Saddle River, NJ (2006)
12. Ekerdt, J.G., and J.B. Rawlings, *Chemical Reactor Analysis and Design Fundamentals*, Nob Hill, Madison, WI (2002)
13. Seider, W.D., J.D. Seader, and D.R. Lewin, *Product and Process Design Principles*, 2nd Ed., Wiley, New York (2004)
14. Lewin, D.R., *et al.*, "Using Process Simulators in Chemical Engineering," CD-Rom for Seider, *et al.*, textbook (2004)
15. Wiesner, T.F., and W. Lan, "Comparison of Student Learning in Physical and Simulated Unit Operations Experiments," *J. Engr. Educ.*, 195-204, (2003); see also <www.che.ttu.edu/vuol>

APPENDIX

2003 Computing Survey of Recent Graduates

In 1997 the CACHE Corporation carried out a survey of recent graduates in chemical engineering from three universities to determine how that group used (or did not use) computing in performing their jobs. Since that time, there have been considerable changes in the field of information technology. Four universities volunteered to participate for the 2003 survey: Carnegie Mellon University, Clarkson University, McMaster University, and the University of Texas. A Web-based form was used to tabulate the responses using database software.

The four universities used different approaches for contacting their recent graduates, defined as students who graduated during the previous five years (1998-2003). Printed mail, e-mail, and/or Web forms were used depending on the specific school. The response rate for the four universities was estimated to be between 20% and 30%, which actually is quite good given the complexity and length of the survey (which took less than one hour to complete). The results of the survey are available in PowerPoint form on the CACHE Web site, <www.che.utexas.edu/cache/survey>.

The questionnaire asked for the nature of the work carried out and the degree level of the respondents. No attempt was made to remove current graduate students from the sample even though they are technically not in the workplace. The overwhelming majority of engineers value computing skills as critical to industrial problem solving. About 75% of recent graduates in the survey characterize their work as "technical."

Compared to 1997, there was a gradual increase in the use of the computer as a general productivity tool. The personal computer is ubiquitous in all business and engineering work, including standard office tasks, with 70% of respondents using a computer actively at least half the day. For the range of using the computer 3/4 to all day, the percentages doubled from 19% to 44% between 1997 and 2003.

Of the respondents, 99% report they use spreadsheets on a daily basis. Faculty have observed that spreadsheets are used by most if not all undergraduates, often with minimal formal instruction in the department. Industry clearly values the use of spreadsheets for a variety of applications based on the percentage of respondents who use them: data analysis (88%), numerical analysis (47%), material balances (25%), economic studies (23%), and other tasks such as financial modeling or emission calculations (17%).

Similar to spreadsheets, database software (70%) has the same level of penetration in daily work usage. It is noteworthy that even with continued improvement of packages such as MATLAB and MathCAD, they are used much more heavily in academia than in industry (26%). Numerical methods libraries are only infrequently used (6%), which illustrates their general decline in popularity since the 1970s.

Less than half of the survey respondents use a process simulator in their work, probably because a growing percentage of students are working in nontraditional industries (outside the CPI). Even in the CPI, not all chemical engineers are actively using simulators in the performance of their jobs.

In 2003 there was more emphasis on and time devoted to training new engineers to use computing in their jobs (compared to 1997). There is a continued reliance by recent graduates on learning new computing skills on their own or with the help of colleagues. This supports the notion that universities should prepare their graduates to "learn how to learn." The amount of formal training to use computing tools continues to be fairly small.

A majority of the respondents (78%) replied that undergraduates should be exposed to some form of programming. This is not surprising even though a minority of engineers write programs in the workplace. Most people agree that use of programming logic is an important skill, whether it is C++, VBA, or MATLAB m-files. Of the respondents, 38% indicate they write computer programs at work (compared to 20% in 1997), but it is not clear what actually constitutes programming in the workplace today (is running simulations considered to be programming?). Use of VBA along with spreadsheets is a dominant practice. The growth of usage of VBA to 34% of the respondents is an important development. C++ leads the rest of the programming options (24%). ❐