A Self Assessment of Computer Science Education in a Chemical Engineering Curriculum

William Josephson, K.C. Kwon & Nader Vahdat
Chemical Engineering Department / CEAPS
Tuskegee University
Tuskegee, Alabama 36088

Abstract
The Department of Chemical Engineering at Tuskegee University (T.U.) regularly reviews its undergraduate curriculum to ensure that it fulfills the department's objective of providing graduates with the skills necessary to begin a career in chemical engineering. Department faculty recently assessed the status of computer science education within the curriculum. Currently, T.U. chemical engineering undergraduates are required to take one introductory programming course (either C++ or FORTRAN) offered by the university's Department of Computer Science. The proper use of commercial software packages such as Excel is taught as a part of required chemical engineering courses. The use of chemical engineering specific software is also required beginning with the gateway Materials and Energy Balances course. The departmental review team examined the relevance of the programming course in addition to how well the other software packages were being learned. The team also studied the computer programming requirements within the chemical engineering curricula at other institutions. This latter study was done via a combination of web based curricula searches and personal contacts. This paper describes the findings of these examinations and the results of the self-assessment process.

Background
Computers have been used to perform chemical engineering-related tasks for more than half a century (e.g., Seader[1] used 1951 as the transition from "Before Computers" to "After Digital"). The early uses of computing devices were characterized by the writing of code specific to a given situation. High level languages such as FORTRAN were used. Since the usual mode of processing was batch, an in-depth knowledge of mathematical methods was needed in order to maximize efficiency of computer resources. Chemical engineering instruction still used (and relied upon) traditional problem-solving techniques such as graphical integration, manual iteration and, in the case of stagewise operations, the plotting of equilibrium curves and operating lines. Electronic calculators were not common in the classroom until the 1970s.

At the dawn of the PC revolution (say 1980) a typical chemical engineering curriculum would include the following technical courses:

Freshman Year – basic college mathematics (Calculus I & II), basic chemistry, basic physics, an "introduction to engineering" course with instruction in engineering graphics, an "introduction to chemical engineering" course with small-scale case studies of what chemical engineers actually do for a living, a computer programming course using FORTRAN and taught by the school's computer science department

Sophomore & Junior Years – more mathematics (Calculus III, Differential Equations), a gateway Materials & Energy Balances (M.E.B.) course; chemical engineering courses in fluid mechanics, heat transfer, thermodynamics, reactors and mass transfer operations; chemistry courses including physical chemistry and organic chemistry; entry-level courses in other technical disciplines such as electrical engineering and materials science

Senior Year – process control, senior design project, senior laboratory, technical chemical engineering electives such as petroleum operations

Students would learn programming skills during their freshman year and make active use of their skills in isolated instances throughout the remainder of their undergraduate academic career. The senior design project would be the course most likely requiring computer programming expertise.

During the 1980s and 1990s the computing resources available to chemical engineering undergraduates expanded dramatically. Universities began requiring that all students have their own personal computer (Drexel started this practice in 1983[2]). Even those universities that didn't have this requirement provided more and better access to user-friendly, local PC applications such as word-processing software (as opposed to mainframe-based applications accessible only by visiting a centrally located computer lab). Some of the expansion in available computing resources may be termed revolutionary in that students gained access to chemical engineering simulation packages which, for the first time, allowed students to forgo the nuts and bolts of code-writing. Using these programs chemical engineering unit operations could be simulated and (importantly) linked together to provide meaningful data. Graphical representations of the linkages became commonplace as Microsoft Windows® based software evolved.

Throughout the years the state of the art has evolved to our current situation: virtually all chemical engineering students have access to personal computers connected to a university network. They're also likely to have a programmable calculator that is capable of many of the simple computing tasks of 25 years ago. Among the computing resources potentially available to them are 1) high level programming languages such as C++, 2) mathematical toolkits/problem solvers such as MATLAB®, TK Solver and Mathematica® and 3) chemical engineering simulation packages such as Aspen Plus® and CHEMCAD. The simulation packages may be full-scale facility simulators such as Aspen Plus® or be more narrow in scope as with Control Station® (used for process control analysis).

Present day ABET requirements are non-specific with regards to the role of computers in the chemical engineering curriculum. ABET merely requires that chemical engineering graduates have working knowledge of "…appropriate modern experimental and computing techniques."[3]

Current Practices at T.U.

Tuskegee University, located in Tuskegee, Alabama, is perhaps the best known of the U.S.'s Historically Black Colleges and Universities (HBCUs). Home to nationally and internationally recognized educators such as Booker T. Washington and George Washington Carver, T.U. has since its founding in 1881 prepared its graduates to play effective professional and leadership roles in society. The Chemical Engineering department at Tuskegee University is an EAC/ABET accredited program started in 1977 and presently offers the Bachelor of Science degree. The department's curriculum is similar to the one outlined above; an environmental option is available that requires additional courses in environmental engineering, environmental science and chemistry.

The department regularly reviews its undergraduate curriculum to ensure that it fulfills the department's objective of providing graduates with the skills necessary to begin a career in chemical engineering. Some of the review processes have resulted in additions or other revisions to the curriculum (e.g., an introductory chemical engineering course that will be taught immediately prior to the gateway materials and energy balances course).

During a recent review, department faculty assessed the status of computer science education within the curriculum. Currently, T.U. chemical engineering undergraduates are required to take one introductory programming course (either CSCI 205, "FORTRAN Programming", or CSCI 229, "C++ for Engineers") offered by the university's Department of Computer Science. Our sample curriculum suggests that the student take the course during the freshman year. The CSCI 205/229 course is the only chemical engineering required course that is explicitly focused on computer usage. However, chemical engineering students are required to use computers in many other instances throughout their stay at Tuskegee. Aspen Plus®, a chemical engineering simulator, is a critical part of the senior design courses and is also used for individual problems within sophomore and junior level courses. Microsoft® Excel is available to the students and is used by most for data presentation purposes in the lab courses. In some courses problems are assigned that require iteration or numerical integration; such problems may be done via Excel, via the high level programming language that the student has learned or by hand with a calculator. A mathematical equation-solving and graphing program (E-Z Solve) is provided with new copies of the textbook used in the Materials and Energy Balances course. However, no instruction by the department is given for E-Z Solve and its use is not required for any M.E.B. problems.

The impetus for the review was an observation and a resulting question. The observation was "our upper-class students do not seem to be utilizing the programming skills they learned as freshmen". And of course the resulting question was: "how relevant is the introductory programming course to the needs of our students?" This question led to a host of others. Both CSCI 205 and CSCI 229 are supposed to provide an introduction to structured programming and problem solving by stressing engineering and scientific applications via the assignment of programming projects. However, do the chemical engineering undergraduate students make active use of these skills during the rest of their academic career and beyond? Some recent[4] and no-so-recent[5] observations have been made that there is little need for a practicing engineer to be able to write computer programs. But if there actually exists such a need, are the specific language skills the most important aspect of the courses or is the ability to structure a program the vital aspect? If the former, then should one language be emphasized at the expense of the other (e.g., change the requirement to only C++ because "no one uses FORTRAN anymore"). If the latter, then why bother teaching a specific language at all? Mathematical toolkits that have programming capabilities built in would serve as an acceptable substitute. Also, if the language is not important, then is it possible that the students actually are using the skills learned in the programming course even though they may not be actively programming in C++ or FORTRAN? On a related note, are the department's other computer-related requirements (i.e., Aspen Plus® and Excel) useful to our students? Could their presentation be improved?

These questions (and others!) led to our department's self-assessment of how computers are used in the program of study for the typical undergraduate chemical engineer at Tuskegee University.

Survey of Other Institutions

One of the first items that was done in performing our self-assessment was to survey other institutions and determine what their computer-related requirements are. We hoped to learn if the traditional model of teaching a computer programming course early in the chemical engineering curriculum was still prevalent and, in cases where it was not required, what alternatives existed. Information on which programming languages are required would be useful in our review as well as what other software is utilized (e.g., math toolkit software such as MATLAB®). We studied the computer programming requirements within the chemical engineering curricula at 17 other institutions. This survey was done via a combination of web based curricula searches and personal contacts. The results are shown in the following table. The "Suggested Term" item reflects the semester that the course is recommended to be taken as shown in the school's sample course schedule. The course descriptions are excerpts from the official course description as shown on the school's web site. The "Taught in" columns refer to the likely course home; e.g., is it taught from the computer science department, the chemical engineering department or is it a general engineering course? A point to consider in analyzing the table's data is that the listed courses are those in which the course description explicitly states that computer skills will be taught. These skills may be needed in other courses in the school's chemical engineering curriculum but are not the focus of the courses. Also bear in mind that, while the departments at some of schools might be considered cross-disciplinary (e.g., Cornell has a department of Chemical and Biomolecular Engineering), the core values of all may be considered to those of chemical engineering.

Institution	Computer Courses	(Suggested Term) Software Used Descriptions	Taught in		
			Comp Sci	Gen Eng.	Chem Eng.
Cornell University	COM S 100 (all engineers)	(SPR FRESH) Java & MATLAB "The course assumes basic high school mathematics (no calculus) but no programming experience."	X		
	ENGRD 241 (most common)	(SPR SOPH) MATLAB & spreadsheets "Introduction to numerical methods, numerical mathematics, and probability and statistics. Development of programming and graphics proficiency with MATLAB and spreadsheets."		X	

Georgia Institute of Technology	CS 1321	(SPR FRESH) N/A Foundations of computing with an emphasis on the design, construction, and analysis of algorithms. Laboratory-based introduction to computers and software tools."	X		
	CHE 2120	(SPR SOPH) MATLAB, HYSYS "Introduction to Numerical Methods (MATLAB and Programming)… Outcomes 5. Perform basic flowsheet calculations using commercial computer aided design (CAD) packages (i.e., HYSYS)."			X
Hampton University	EGR 102	(SPR FRESH) N/A "Intro to Structured Programming"		X	
Howard University	(Systems & Computer Sci.) 306-165	N/A N/A "Introduces programming and use of digital computers. Programming includes general problem-solving and the systematic development of algorithms; use includes the coding of programs and practical experience in operating computers and peripheral equipment."	X		
	CHEG 306	N/A FLOWTRAN & CHESS "Treatment of experimental data, with steady state simulation of chemical processes using FLOWTRAN and CHESS; solution of ordinary differential equations by LaPlace transforms and numerical methods"			X
Kansas State University	CHE 316	(FALL SOPH) N/A "Application of computational methods including programming to chemical engineering problems"			X
	CHE 516	(FALL SR) N/A "Application of computational methods with emphasis on simulation to chemical engineering problems"			X

Louisiana State University	ChE 2160	(FALL SOPH) N/A “Introduction to operating systems, programming techniques, and software packages used in the solution of chemical engineering problems.”			X
	ChE 2176	(SPR SOPH) MAPLE “Basic concepts and techniques in analysis of engineering processes; mathematical descriptions of physical systems and application of modern computers to solution of resulting equations.”			X
North Carolina A&T University	CHEN 102	(SPR FRESH) N/A “This course is an introduction to the solution of chemical engineering problems…. Topics to be covered include problem solving methods, pseudocode, flow charts, variable declaration, input and output, formatting, sequence control structures, selection control structures, repetition control structures, array manipulations, functions, subprograms, plotting, round-off and truncation error, program documentation and program debugging.”			X
	CHEN 220	(SPR SOPH) ASPEN PLUS “The use of a chemical process simulator is introduced.”			X
Pennsylvania State University	CMPSC 201C/F	(FALL SOPH) C or FORTRAN “Development and implementation of algorithms in a procedure-oriented language, with emphasis on numerical methods for engineering problems.”	X		
Purdue University	ENGR 106	(Fall FRESH) MATLAB “Introduction to engineering problem solving and the use of computer software, UNIX, computer communications, spreadsheets, and MATLAB.”		X	
	CS 156	(SPR FRESH) C (preferred) or FORTRAN “Introduction to programming in C to solve engineering problems: integer and floating-point data, standard mathematics library, control structures, user-defined functions, arrays, character data, strings, input and output.”	X		

University of Alabama – Tuscaloosa	GES 126	(SPR FRESH) FORTRAN & spreadsheets "Introduction to computing problem solving and algorithm development."		X	
University of Colorado	GEEN 1300	(FALL FRESH) EXCEL/VBA, MathCAD & MATLAB "Learn programming fundamentals, including data and algorithm structure, and modular programming. Numerical methods learned include solving single, nonlinear equations, fixed-poing iteration, Gaussian elimination, and linear regression."		X	
University of Florida	CGS2425 & CGS2425L	(FALL SOPH, 5 yr program) no specific language required, C, C++, FORTRAN or Visual Basic suggested "The chemical engineering undergraduate degree program requires that you be able to program a computer to solve engineering problems"	X		
University of Michigan	ENG 101	(SPR FRESH) C++ & MATLAB "Algorithms and programming in C++ and MATLAB, computing as a tool in engineering, introduction to the organization of digital computers."		X	
University of Mississippi	CSCI 251	(FALL SOPH) N/A "Algorithm development and structured programming techniques"	X		
University of South Alabama	CHE 203(?)	(FALL JR) MATHCAD, MATLAB, EXCEL & ASPEN is a new course (Chemical Engineering Calculations III) which uses the above software to solve chem eng problems			X
University of Toledo	CHEE1000 (Orientation for Chem. & Env.	(FALL FRESH) N/A "Includes an introduction to engineering computing and programming and basic chemical			X

	Engineers)	engineering calculations." Elsewhere on site are help sections for MATLAB & ChemCad			
University of Wisconsin	CS 310	(SPR SOPH) FORTRAN & C "Discusses several methods of using computers to solve problems, including elementary Fortran and C programming techniques, the use of spreadsheets, symbolic manipulation languages, and software packages."	X		

The table shows the following:

- Of the 16 institutions for which we were able to determine recommended course timing, 9 of them have a computer programming oriented course as part of their recommended freshman year curriculum and 6 of them have such a course as part of the sophomore year. One (U. of S. Alabama) doesn't formally require programming skills until the junior year but the university has a university wide requirement of student PC access/ownership and competency.
- 12 of the 17 schools introduce computer skills in either a course offered by the university's computer science department or in a general engineering course. The other 5 schools have the introductory course within the university's chemical engineering department. We surmise that in the latter case the course is not a formal programming course but is more applied and links chemical engineering topics to the programming concepts being taught.
- C, C++ and/or FORTRAN are popular languages at the introductory level (mentioned 6 times) but toolkit languages such as MATLAB® are about as popular at that level (5 mentions). When programming is taught to more experienced students then the toolkit languages are used exclusively. The use of (and formal instruction in) a toolkit language at some point in a chemical engineering student's career appears to occur in approximately two-thirds of the institutions (8 mentions out of the 12 institutions for which we were able to determine the languages used).

Self Assessment

Our analysis of the survey data showed that the computer programming policies required by chemical engineering department at T.U. fit the norm of the surveyed institutions. Our requirement of a C++ or FORTRAN course early in the suggested curriculum is fairly standard. The use of a computer based process simulator is also standard in a discipline that has a senior-level, facility design component to the curriculum.

The wide assortment of languages confirmed our initial feelings that the choice of language is not as important as the structured programming concepts contained within the course. MATLAB® being mentioned equally as often as the traditional languages of FORTRAN and C++. Although the form may differ, every surveyed institution requires its students to master basic programming skills.

If we judge solely on the basis of teaching students programming skills then our current requirements are acceptable. Could we improve our curriculum by switching to a toolkit approach as other institutions have done? If, for example, we required MATHCAD® instead of C++ or FORTRAN then our students would 1) still learn structured programming concepts, 2) have an increased chance of utilizing a software package that they may come across in industry and 3) be able to take advantage of all the other aspects of toolkit software such as easy graphing capabilities. While these are advantages, the T.U. review team also realized that cost-effective adoption of toolkit software would necessitate utilizing it throughout the entire chemical engineering curriculum. In addition, using math toolkit software would most likely have to be done on a college-wide basis as opposed to on a departmental basis. The other three engineering disciplines within Tuskegee's College of Engineering, Architecture and Physical Sciences (i.e., Aerospace, Electrical and Mechanical) all have a computer programming requirement similar to that of the chemical engineering department. It would be infeasible for the chemical engineering department to unilaterally drop the computer science requirement. Therefore the review team decided not to recommend the switch to a mathematical toolkit approach. Such an approach remains an option, however, and the chemical engineering department will continue to review the curriculum as it relates to computer programming skills.

Feedback from current and past students as well as discussions within the department faculty did reveal areas for improvement. Although Excel is used by the students they are not using it to its full potential. As mentioned above, its most common use is for plotting data. This sometimes poses difficulties because Excel is, at heart, a business/accounting program. A common error by students when they are plotting data in Excel is to use Excel's default mode of data layout in which x axis data is treated as a label with no numerical significance. The resulting plot will then be skewed but not always obviously so.

The review team also found instances in which students were unaware of Excel's full potential. An example: during a heat transfer course students were required to solve a 2-dimensional, steady state heat conduction problem via the finite difference method. The students in the class were unaware of Excel's capability to solve this type of problem (by treating spreadsheet cells as nodes in the finite difference method and using Excel's iteration feature) and instructional time had to be devoted to this task.

It is important to realize that the difficulties our students have with Excel lie in the fact that there is no formal instruction in Excel or any spreadsheet program, students "pick it up as they go" through their academic career. The authors have noted similar difficulties with Excel at other institutions that do not formally train their students in the best use of the software. Accordingly, the review team determined that an Excel instruction component in the introductory chemical engineering course is needed. We will instruct the entry level students in the basic techniques of spreadsheet usage and also teach them the intermediate level skills that are highly useful in later courses.

The other area for improvement within our curriculum is similar in nature. Although the use of Aspen Plus® is required for most of the chemical engineering courses, instruction in the software is done on a case-by-case basis until the senior design courses are reached (e.g., the instructor of a heat transfer course may teach the specifics of Aspen's shell-and-tube heat exchanger models in order to assist the students in a design project). The result is that students have no sense of the overall scope of the software and may struggle when first presented with a project involving multiple unit operations. Our solution to this problem has been to establish a series of required Aspen Plus® workshops taught outside the normal curriculum. Students attend these workshops starting in their sophomore year and may attend as often as desired. The emphasis during the workshops is a holistic one with an overall approach to Aspen Plus® being stressed as opposed to a nuts-and-bolts analysis of individual unit operation models. We hope that our students' comprehension and ability to use Aspen Plus® will improve.

Conclusions

The review team used two approaches in assessing the status of computer usage within Tuskegee University's chemical engineering department: a survey of computer related instruction at other institutions and an examination of how our students are actually making use of the computer in their chemical engineering courses. The survey showed that our requirement of a computer programming course (C++ or FORTRAN) fits within the spectrum of practices at other institutions. However, we did find that many institutions have de-emphasized these traditional languages and have instead used an approach that teaches programming skills within the context of a mathematical toolkit environment. The review team decided not to recommend such an approach for T.U. at this time but the toolkit approach will be reconsidered in the future.

Our internal examination showed that our students were not making the best use possible of available software, in particular Excel. We have therefore made plans to include an Excel instructional computer in our introductory chemical engineering course. In a similar move, we have instituted dedicated sessions in which the use of Aspen Plus® is taught. It is hoped that our students ability to use the computer tools available to them will improve as a result of these efforts; our degree of success in this area of the curriculum will continue to be regularly evaluated.

---

[1] J.D. Seader, "The B.C. (Before Computers) and A.D. of Equilibrium-Stage Operations", Chemical Engineering Tutorial Lecture presented at the 92nd ASEE Annual Conference, Salt Lake City, Utah, June 22-28, 1984.
[2] R.G. Quinn, "Drexel's E[4] Program: A Different Professional Experience for Engineering Students and Faculty", *Journal of Engineering Education*, 82(4): 196-202 (1993).
[3] from ABET's 2003-2004 Criteria for Accrediting Engineering Programs - PROGRAM CRITERIA FOR CHEMICAL AND SIMILARLY NAMED ENGINEERING PROGRAMS
[4] K.A. Neeley et al., "Integration as a Means to Excellence in Engineering Education and Practice", report from a working group of the "Liberal Studies and the Integrated Engineering Education of ABET 2000" conference held in Charlottesville, Virginia, April 6-7, 2002.
[5] J.C. Kantor and T.F. Edgar, "Computing Skills In The Chemical Engineering Curriculum", in Computers in Chemical Engineering Education, Brice Carnahan, editor, CACHE Corp., Austin, 1996.