Freshman Engineering Teaching Experience Using Computers - Problems and Solutions

Samuel O. Osisanya
School of Petroleum and Geological Engineering
The University of Oklahoma

Abstract: The objectives of a freshman or introductory engineering course are to teach the students fundamentals of engineering as a profession and to motivate them to develop an understanding of engineering problem solving skills using computers. Problems associated with teaching computer applications to freshman engineering students have been identified based on experience. These are time limitation, class size (number of students), students' academic preparation and computer experience (skills), institutional needs and commitments (accessibility), instructor's preparation, commitments, and attitude and some essential human factors.

At the first year level, certain modes of computing are useful. Computer software applications are more useful than the traditional programming languages. Thus, some of the solutions to these problems are readily accessible computers (hardware), high-quality software, and dedicated support staff. Three important user-friendly software packages have been identified and used successfully at the University of Oklahoma (OU) College of Engineering. Above all, students must be informed about the benefits of computer applications to freshman engineering. The benefits include quick engineering analysis, better communication and creativity, competitive advantage in job search and value to the students' future employers.

This paper will discuss the problems, the solutions and the essential human elements needed for the success of engineering teaching using computers.

Introduction

An introductory engineering course is the first core course in engineering that is taken by all engineering majors. Students in this group are generally admitted either directly from high schools or those that just made up their mind to pursue engineering after one or two semesters in the college, The University of Oklahoma, College of Engineering decided to integrate computer instruction into freshman introduction to engineering in early 90' s[1]. This was in response to a perceived need within the college to increase its computer instruction and the express advice of visiting engineering council members most of them from industry. The objective was to simulate the environment that the engineer will work in when he or she goes out into the job market. It was important that the students upon entering the industry, be computer literate and capable of producing a computer-generated engineering report or design.

Highlights of the 1993 National Science Foundation Summer Workshop on Introductory Engineering Experience, held at Colorado State University's Pingere Park Campus from July 12-16, 1993

indicated that certain modes of computing are useful at the freshman level[2]. Computer software applications are recognized as useful tools for freshman engineering courses, but not programming languages. The success of introduction of computing to freshman engineering was identified to be dependent on several variables. In particular Fitzhorn[3] enumerated some of these variables as time limitation, number of students (class size), and their computer literacy
and skill, academic preparation, institutional needs and commitments, resources level (dedicated computer room and laboratories for freshmen), and instructor's attributes (dedication, level of commitment desire to motivate, etc.).

At the same conference several moderators debated on the goals of teaching computer applications and computer programming at the freshman level. The consensus was that computer applications are taught for the following reasons:

. To familiarize the students with the capability and use of the computer to solve engineering problems.
. To build student confidence in using computers.
• To develop an understanding of the nature of computing and its limitations.
• Good for analysis, communication, creativity and visualization.
. Valuable to employers.

Thus, the advantages of computer usage in the first year introductory engineering course were identified as follow:

. Early introduction of computers to the students is beneficial since computers are the new tools for engineers.
. It provides a systematic way of solving engineering problems.
. It is good for data storage and retrieval.
• It enables simulation of engineering problems (i.e. to play "what if").
• Students are comfortable in front of screens.
• It is a marketable skill for the students.
• Two other advantages were identified by this author while teaching. One is that more time can be devoted to more complex problems and thus increase the amount of learning that goes on. Also, students will be able to create their own mathematical models of reality and observe the effects of various actions.

On the other hand, some disadvantages of computer usage to first year engineering students were enumerated.

- The engineering course may look like computer science.
- Students may misconstrue what engineering is.
- The computer hardware/language may become obsolete.
- Computer bogs down the students with details and is time consuming in the initial learning stages.
- There is tendency for students to put too much trust in the answer.
- A burden is placed on resources due to high cost.

One other disadvantage identified by this author is inadequate student preparation.

Literature Review

There has been a computer revolution in engineering education, but to date it has been much less far--reaching than many proponents predicted. Computers and calculators have greatly increased the ability of students (and practicing engineers) to perform calculations. Since computers and calculators allow professors and students to do a much better job at calculation, they have been widely adopted in engineering education. As a result, professors have changed the nature of the problems presented, and they have changed many of the mathematical techniques taught. This has been an important change in the way engineering is taught (and practiced). However, some authors believe that we have not seen significant adoption of computers for the delivery of instructional.

The commonly used generic computer tools are spreadsheets, equation solvers, and symbolic algebra programs. Simulation programs tend to be much less generic but will be discussed with the other tools. Before any computer application is adopted, the professor needs to determine whether five prerequisites for instructional use of computers have been met. The first three prerequisites listed by Trollip[5] are accessibility, high-quality software, and faculty interest. The other two prerequisites listed by Wankat and Oreovicz[4] are the advantage of computers and student computer background.

Computer tools

Engineering professors have discovered the use of generic software such as word processors, spreadsheets and equation solvers for the solution of engineering problems. These available software packages have become more powerful, robust, and user-friendly. They represent an extremely useful middle ground between hand solutions and computer programming. Furthermore, because some students will do almost anything to avoid programming, the generic packages are user-friendly enough that, with a little training, almost all students can be induced to use them. Thus, in many applications computer tools are a significant advance over both hand calculation and programming. Because of this advantage, computer tools, particularly spreadsheets, have been widely adopted.

Students who have not learned a particular software tool before it is introduced in class are generally antagonistic about computer and they will not use it unless they receive help. Keedy[6] suggests the development of core manuals for software using the "20-80 rule". He recommended that instructors should identify approximately twenty concepts and associated keystrokes which represent 80 percent of the power of the package- and everything the students need to do. By so doing, when students first learn the package, they don't need to know the most efficient way to do something; instead they need to know the easiest way to learn and remember. Once the 20-80 items have been identified, a short core manual which explains how to use these selected features should be written. Furthermore, interested students will learn other operations on their own or from other students once they know how to use the basics of the software.

Several authors[7-10] have enumerated the advantages of the spreadsheets. As long as a spreadsheet has appropriate graphing and scientific function features and is fast enough, the choice of spreadsheet is almost immaterial'. In addition, students who learn how to use one type of spreadsheet can easily learn to use a different spreadsheet on their own. Thus, there is no need to worry about them seeing a different spreadsheet when they graduate. Spreadsheets are easy to learn. On the average, one two-hour laboratory is sufficient to learn the basics. Spreadsheets remove much of the tedium from doing calculations and allow the professor to assign more meaningful problems.

In many engineering classes spreadsheets allow students to get to real engineering problems faster, and permit them to focus on thinking since the program does the routine calculations. Spreadsheets are easier to use than programming from the start and hence tend to be more efficient. The students must be encouraged to use the spreadsheet regularly in some engineering problems because with practice the students become more proficient. The first generation of spreadsheets were slow, large-scale branching was difficult as was the use of variable names[10]. However, present generations of spreadsheets are fast and versatile. Thus, this author is strongly in favor of the integration of spreadsheets into the engineering curriculum at all levels.

Spreadsheets are not without problems. If students are unfamiliar with spreadsheets or do not use them for a significant period of time, their introduction along with engineering material may decrease the learning of material[11]. This could well be due to over saturation with new material. If spreadsheets are introduced early in a course and used throughout the course, this should not be a problem.

From above introduction and literature review the problems and solutions to freshman engineering using computers can be identified.

Associated Problems and Solutions to Freshman Engineering Computing

Time Limitation/Class Size: In general, the teaching of computer applications to solve engineering problems and working with computers are time consuming. Most engineering curriculum allocates two to three hours per week of instruction and computer application to freshman introductory engineering course. The three hours per week may be adequate, but definitely the two hours per week is not. This is especially true if class meets twice a week, during which too much time is wasted in opening, closing the class with regards to networks, soft ware and files computing. A small class is generally recommended for any computer class. This facilitates interaction between the students and the professor. There is no way one instructor can successfully teach more than fifty students on computer. For example, the Engineering Computing Network (ECN) at OU offers seminars for twenty people per one staff. Institutions must strive to balance the students to instructor ratio for effective teaching that enhances retention.

Institutional Needs and Commitments

These two factors include computer accessibility, instructor's attitude, commitment, and preparation and the dean's commitment.

Readily Accessible Hardware (Computer): Both the hardware and the software must be readily accessible to both students and faculty. The software must do something that the students want it to do, it must have clear and unambiguous screen displays, the interaction between user and machine must be easy, the software must be easy to use, the software must be relatively fast, and above all, the software must be robust. Access to computer is an administrative, technical, and teaching issue. Students learn best and fast when they each have a computer and are able to do hands-on work during class. In this respect the computer facility for freshman introduction engineering at OU comprises of 60486 PC's, two teaching podiums, dot matrix printers, 1 color printer, and 1 laser printer. The PCs are at least 8 meg of memory. All the PCs were upgraded to 16 meg recently. Again, this is the part of the commitment of an institution.

***Software** availability and selection:* The instructor must research the software available, and if they are going to use the software in conjunction with a textbook, it is very advantageous to have a textbook that uses the same software in its examples in the text. This may not be possible in all cases. It is frustrating if one has to teach the computer content from a computer manual and the substantive areas from a textbook that doesn't have either that same computer program or has none at all.

In general, a spreadsheet is recommended for freshman introductory engineering. Spreadsheets are in many ways easier than programming. They are structured and encourage students to structure their calculations, even for hand calculations. A spreadsheet can easily show tabular solutions. It is easy to debug since syntax errors are shown immediately, and the instant display of numerical results makes it easier to spot obvious mistakes. Input and output are easy since any cell can be displayed or changed at any time. The inclusion of graphics capabilities means that students can easily prepare presentation-quality graphics and search for trends visually instead of looking at a mass of numbers. In addition, spreadsheets are easily documented since each cell can be labeled. Spreadsheets have been used in all engineering courses and engineering disciplines. To mention a few, this author used spreadsheet in petroleum engineering to simulate the analytical solution to a flow pressure differential equation-developed in his dissertation work[12]. The

spreadsheet allowed for rapid analyses and for sensitivity studies of several variables. Miscovich and Biasca[10] have used it in chemical engineering to solve mass and energy balance problems.

New Development at OU

By this Fall (1996) at OU a pilot test (of 25 students) will get underway at College of Engineering students to purchase their own PCs[13]. There are a number of fundamental reasons for requiring all engineering students to purchase their own computers. The first is that the students will be required to use computers in the College of Engineering for their work. Results of committee on student computer usage survey indicated that over 65% of freshman indicated that they use computers more than 2 hr./week with 30% reporting greater than 10 hr./week. These percentages jump at the senior level. The second reason is that computer usage will continue to expand at a great rate and more space will be needed. It may not be possible for OU Engineering Computing Network (ECN) to handle this load. The third reason is that most students already own computers. The fourth reason is the practicality. If students are required to have a computer with standard software applications, then all faculty could be assured that the students have the necessary computing power to tackle their assignments. This would allow the faculty to increase computer usage as a classroom tool.

Instructor's Preparation, Commitments and Attitude

The faculty must have sufficient interest, motivation and energy to follow through with the project. The amount of interest and energy required depends on the project. For adopting generic tools such as spreadsheets, the amount is modest, but for writing computer-aided instruction packages it can be staggering. Students learn best when acting and reflecting not by watching and listening. Interactive computer software packages that supplement traditional classroom presentations can provide opportunities for both. Thus, a well designed package can supplement the presentation of the basic course concepts, test their understanding, provide feedback for their efforts, and generally increase both the quantity and the depth of their learning.

One of the key factors to the successful development and use of interactive computing is the identification of activities that cannot be accomplished by other means (e.g. pencil and paper, calculator). That is, a computer must be able to do something better than the student can do it working without the computer. If there is no perceived advantage, then students will not use the computer and the faculty will drop the experiment. These activities are those that are used to learn new material, practice certain skills, or test comprehension of previously learned material. Activities that must be included in educational software can be based on the Bloom et al[14]. Taxonomy of Educational objectives which has identified six skill levels: knowledge, comprehension, application, analysis, synthesis, and evaluation. The first three levels are low-level thinking skills which are generally used for many undergraduate courses. The other three levels are high-thinking skills and the computer-based materials are one way to allow students to exercise their higher level thinking skills.

Clearly, instructors and students must prepare differently for computer-oriented courses. With the ease and speed of learning increased, instructors need to prepare more material. Much educational software is currently being developed. When evaluating software for possible course use, it is important to focus on the pedagogical roles, educational objectives, and learning styles addressed by the software. Instructors must be very careful about on-line exams. On-line examination involves using the PCs to do tests. This process is a function of the size of the class. Also, what happens if the server goes down 45 minutes into the test? In general, this idea is not encouraged for freshman engineering for the following reason. It is difficult for all the students to perform the same function at the same time. However, if this is a must then it is better to divide the exam into independent parts. The number of parts will depend on the total number of students in

the class. Even with this division there are still problems. Another advice is to let the teaching assistant try the test from a student PC. For the peace of mind, one will know that it does work on the system that all the students will use.

Also, instructors should be very careful introducing equation solvers to freshman engineering even though they are similar to spreadsheets. Equation solvers perform the input and output routine for the user, including graphics routines, and they choose the algorithm, although the user may be able to override this choice. The programs require that the students know how to write the equations. However, they do not need to know how to solve the equations, and in the worst case the program can be a black box. Equation solvers are in general more powerful than spreadsheets, they are normally recommended for upper division engineering course. In fact spreadsheet should be taught first since they are more generic, are more visual, are easier to learn, and are applicable to the problems taught in lower-division courses. Additionally, students who learn the power of the spreadsheets are more likely to believe that the time invested in learning to use an equation solver will be well spent.

Students Academic Preparation, Computer Skills and Attitude

Students must be taught how to use both the hardware and in particular software, If this has not been done in a prerequisite course, then they must be taught in the current course. Probably the ideal arrangement is to teach engineering courses. Learning about unfamiliar hardware and software in a discipline-oriented course can lead to cognitive overload and a poorer performance for many students [15]. This is especially true for weaker students. Students who have had only minimal exposure to computers sometimes have difficulty keeping up. It is recommended that such students gain preparatory training before enrolling in the course. This may mean studying on their own, taking a class independently or seeking tutoring. The instructor must be able to identify these students and give them his or her honest opinion on what to do. The ECN at OU has been working to lower this learning curve by creating teaching modules which are available to students on the network, short courses, and technotes (quick tips sheets).

Students are generally frustrated when working with computers. This occurs frequently especially when one does not know what to do next to make the computer obey certain instruction. The instructor must teach students in this situation by telling them to be aware and patient. The students must be warned not to rely completely on computer models. Computer programs have made too many people including students believe that all technical problems are readily solvable. They should be encouraged to perform the same problems with the old-fashioned, direct hands-on calculations. In other words, students must be informed that the physical meaning of the subjects is necessary. Successful engineering design still requires expert tacit knowledge and intuitive “feel” based on experience. Computer simulations break reality into chunks, as many as possible but always too few. A computer model is just a set of arbitrary rules; chosen by the programmers. The students must be warned of the danger of the number of significant figures in their calculations, because the computer’s apparent precision to six or more significant figures can give students an unwarranted confidence in the validity of the resulting numbers [16]. Students should beware of a bridge designed on the first pentiums!

The students should be warned of several time wasting activities on the computer in freshman engineering classes. Some of these activities are programming, setting up of files, computer games, and e-mailing. Computer programming is a very time-consuming process. Students should be asked to write fewer simple programs. Setting up files causes the files to become an end unto themselves and they are seldom used for a productive purpose. Students often use games for relaxation and they are generally non-educational. Some students engage in e-mailing during class period. Definitely this is a nonproductive use of a computer. In regards to the last three cases, the Engineering computing center at OU always delete non-

system files on local hard drives daily. No games software on network. E-mail is available and roving TA generally monitors "off subject" use of computing during class.

Some Essential Human Factors

Any computer oriented engineering course needs the support of dedicated and knowledgeable staff just as a technician is needed in the laboratory. In other words the students must be provided with dedicated and caring staff, terminal operators, teaching assistants. The College of Engineering at OU makes sure that these essential human factors are provided. Assistance is offered in terms of a staff member (assigned to a particular class), student assistants to assist the students on how to log in, how to get in or out of software, how to print, and provide suggestions on debugging. Also student terminal operators are available 24 hours a day; and are responsible for the hardware such as maintaining printers, plotters. The instructor must treat these people and their work with respect. By so doing they will be helpful. The ECN provides several means to accessing information to assist the students in interfacing with the computer, Internet WEB pages, interactive tutorials (Explore ECN and SANHELP) and the traditional written technotes free of charge to students. Explore ECN is a PC based interactive teaching module which covers all the "user" basics in text, animation and sound to address the different learning styles of the students. The computing center also maintains an extensive library of software manuals to be used as reference books.

A good teaching assistant (TA) can extremely be helpful to professors, particularly in an engineering class that is computer oriented. The teaching assistant must be made a partner in teaching the course. Before the semester starts the TA must be informed of the following: the professor's expectations, attendance, note taking in class, grading, computer laboratory hours and making sure equipment is in good condition. The TA must prepare ahead of time for computer laboratory assignments.

Summary and Conclusions

Engineering students require analytical tools of science and mathematics, but more importantly they need sound judgment and an intuitive sense of judgment and intuitive sense of fitness and adequacy. Computer application is an excellent time saver for both the professor and the students especially if used for activities that require a great deal of time. Some of these activities are writing or preparing manuscripts, graphics, and problems solving using spreadsheets. The activities will be time saver if the software applications are used always. While trying to get started teaching freshman engineering computing, the professor must not overshoot or try to do everything. The instructor must start slowly, rather than trying to use computer to do everything. The instructor must be sure that the software applications are ones that the students will be able to understand and use in their future courses.

Students must be warned of the black box syndrome. As the program becomes more complex, it becomes increasingly likely that the student will not understand or perhaps even care what the program is doing. It is recommended that students should do simple hand calculations and then repeat the same problem with a computer. This helps them understand what the computer is doing, gives them confidence, and shows them how the computer can save time.

Acknowledgments

The author gratefully acknowledged Dr. Badiru, Pat Wyatt, Kelly Deaver, and Cathy Pierson for their advice and suggestions on this paper. Financial support from NSF through the NSF Faculty Career Award Grant made possible the research and preparation time for this work.

References:

1. Spotlight on Teaching, A University of Oklahoma Newsletter for faculty and Teaching Assistants form the Instructional Development Program, Vol. 14, No.2, April 1993.

2. Osisanya, S.0., Highlights of the 1993 National Science Foundation Summer Workshop on Introductory Engineering Experience, held at Colorado State University's Pingere Park Campus from July 12-16, 1993, a report prepared for the Dean of Engineering, The University of Oklahoma, August, 1993.

3. Fitzhorn, P. A., "Computing is Design and Design is Computing," One of the keynote addresses at the 1993 NSF Foundation Summer Workshop on Introductory Engineering Experience, held at Colorado State University's Pingere Park Campus from July 12-16, 1993.

4. Wankat, P.C. and F. S. Oreovicz, *Teaching Engineering,* McGraw-Hill, New York, 1993.

5. Trollip, S. R., "Wrestling with instructional computing," Acad. Comput., 2 (Dec. 1987), 26 (Jan. 1988).

6. Keedy, H. F., "If they don't need it, don't give it to them," Proceeding ASEE Annual Conference, ASEE, Washington, DC, 1916, 1988a.

7. Lofy, F. J., "Design of electronics circuits using spreadsheet techniques," Proceedings ASEE Annual Conference, ASEE, Washington, DC, 1567, 1988.

8. Mortimer, K., "Spreadsheets ain't just for bean counters," Proceedings ASEE Annual Conference, ASEE, Washington, DC, 1529, 1987.

9. Cress, C., "Introducing spreadsheet modeling across the engineering science curriculum," Proceeding ASEE Annual Conference, ASEE, Washington, DC, 909, 1989.

10. Misovich, M. J., and K.L. Biasca, "The power of spreadsheets in a mass and energy balance courses," Chem. Eng. Educ., 46 (Winter 1970).

11. Merino, D.N., "Effectiveness of computer based problem solving in teaching engineering economics," Proceeding ASEE Annual Conference, ASEE, Washington, DC, 293, 1989.

12. Osisanya, S.0., and M.E. Chenevert, "Physico-Chemical Modeling of Wellbore Stability in Shale Formations," JCPT, February 1996, Vol. 35, #2; pp. 53-63.

13. Committee Report on Students-Owned PC submitted to Dean Billy Crynes, College of Engineering The University of Oklahoma; May 30, 1995.

14. Bloom, B. S., M.D. Engelhart, E.J. Furst, W.H. Hill, and D.R. Krathwohl, Taxonomy of Educational Objectives: The Classification of Educational Objectives. Handbook I: Cognitive Domain, David McKay, New York, 1956.

15. Whitney, R. E., and N. S. Urquhart, "Microcomputers in the mathematical science: Effect on courses, students, and instructors," Acad. Comput., 4 (6), 14 (1990).

16. Ferguson, E. S., "How Engineers Lose Touch," Invention and technology, Winter 1993; pp 16-24.