Teaching "Computer Tools" in a Workshop Format

Surendra K. Gupta
Rochester Institute of Technology

Abstract

Mechanical engineering freshmen at RIT take **342**-*Problem Solving with Computers* as the introductory computer course. **342** replaces the traditional course in Fortran Programming. Students now develop function subprograms in Visual Basic for Applications (VBA) within the Microsoft Excel environment. **342** emphasizes software tools such as a spreadsheet, word processor and symbolic computational system. Skilled in such software tools, students are submitting improved home assignments and lab reports in their subsequent science and engineering courses.

Many 2-year and 4-year colleges have not updated their introductory computer courses to include such software tools. To accommodate students transferring into the third year from such colleges, the department developed a 1-credit **441**-*Computer Tools* course. In the last two years, the course was offered in a five-week hands-on lab tutorial format. Two-hour tutorials were held twice a week in the department's PC lab to acclimatize students to PC-based software tools. Student evaluations indicated that this format was not providing timely support to **440**-*Numerical Methods* and **413**-*Thermodynamics* students took concurrently.

This Fall (19961), **441** was offered in a weekend workshop format (the first two Saturdays of the Fall quarter). The development of the new format was financially supported by a 1996-97 Provost's Productivity & Teaching Innovation Grant. The course content was divided into eight largely independent instructional modules. A detailed Users' Guide was prepared. Each module consists of an instructor-led hands-on tutorial, an engineering or science application to provide the context or relevance, and an in-lab exercise to reinforce the lesson. Students evaluated the workshop in regard to its content, instruction, teaching aids, lab facilities, and relevance to other courses. The evaluations indicate that the new format has been successful.

Historical Perspective

By the early 1980s, almost all mechanical engineering programs required their freshmen or sophomores to take a course in higher-level programming language (primarily Fortran, C or Basic). Most programs required students to take an additional course in Numerical Methods[1,2]. With skills students developed in these two courses, they were able to write programs for engineering design and analysis projects, and to develop data acquisition and analysis software for experimental projects in their upper division courses.

By the late 1980s, with the personal computer (PC) revolution underway, many upper-division technical courses relied almost exclusively on specialized software packages that were inexpensive, easy-to-use-and-learn and available on almost all platforms. For example, our students use CADKEY for computer-aided design and drafting, Algor Supersap for finite element analysis, Fluent in fluid mechanics courses, and Tudor for elective courses in the aerospace option. No upper-level technical course currently requires development of programs in a higher-level language.

Recent Developments

By the early 1990s, new challenges surfaced. Most of the lab instructors began asking students to submit word-processed lab reports with computer-produced graphs and data analysis. Even in traditional courses such as Thermodynamics, Statics and Dynamics, instructors began assigning homework problems involving parametric studies that required the use of general-purpose software such as a spreadsheet (Excel, Lotus 1-2-3 etc.) or a symbolic computation program (Maple, Mathematica etc.). To respond to these challenges, we revised our freshmen-level **342**-*Fortran Programming* course. It was renamed **342**-*Problem Solving with Computers* (visit http://www.rit.edu/~skgeme/emem342), and introduced students to general-purpose computing and problem solving tools[3,4]. Students developed skills with spreadsheet (Excel), word processor (Word) and computer algebra system (Maple). The follow-on courses (see Table I below) **440**-*Numerical Methods* (visit http://www.rit.edu/~skgeme/emem440)and **518**-*Advanced Computational Techniques* (visit http://www.rit.edu/~skgeme/emem518) were revised accordingly to take advantage of new skills students possessed[5,6,7]. These innovations have been selected to receive an Honorable Mention in the 1996 ASME Curriculum Innovation Award Program[8].

Table I: "Computational Methods" Course Sequence

Code	Course Title	Credits	Year	Topics
342	Problem Solving with Computers	3	1st	Spreadsheet, Computer Algebra System, Word Processing
440	Numerical Methods	4	3rd	Numerical Techniques with applications
518	Advanced Computational Techniques	4	3rd	Finite Element Techniques, Finite Difference Techniques

Transfer Students

Many 2-year and 4-year colleges have not updated their introductory computer courses to include PC-based software tools. To accommodate students transferring into the third year from such colleges, the department developed a 1-credit **441**-*Computer Tools* course. In the last two years, the course was offered in a five-week hands-on lab tutorial format. Two-hour tutorials were held twice a week in the department's PC lab to acclimatize students to PC-based software tools. Student evaluations indicated that this format was not providing timely support to **440** (and other third year courses) students took concurrently.

This Fall, **441** was offered in a weekend workshop format (the first two Saturdays of the Fall quarter). The development of the new format was financially supported by a 1996-97 Provost's Productivity & Teaching Innovation Grant. The course content was divided into eight largely independent instructional modules. A detailed Users' Guide was prepared[9]. Each module consists of an instructor-led hands-on tutorial, an engineering or science application to provide the context or relevance, and an in-lab exercise to reinforce the lesson. Students evaluated the workshop in regard to its content, instruction, teaching aids, lab facilities, and relevance to other courses. The evaluations indicate that the new format has been successful.

Instructional Modules

The eight instructional modules introduce students to spreadsheet (Excel) operations, word processing (Word), and computer algebra system (Maple). A brief description of each module is given below:

Module #1 covers spreadsheet basics. Students are introduced to elements of an Excel window and mouse operations. They create a table, learn to format cells, perform simple arithmetic, save the worksheet in a file, and print the worksheet. A good example for this module is to build a gradebook for the course.

Module #2 teaches students to create and edit graphs of functions and data. Students are able to modify plots for visual appeal and plot several curves on the same graph with two abscissa values. They also learn to create user-defined functions in Visual Basic for Applications (VBA) within the Excel environment.

Module #3 covers several methods to solve a non-linear equation. The objective of the module is to instruct students in creating conditional formulas (If-Then-Else) in cells, naming cells and copying formulas from a group of cells to another group of cells. Students develop formulas that incorporate user-defined functions, built-in functions and named parameters. With this module, students begin to appreciate the convenience of a spreadsheet when compared to developing a program for the same task in a higher-level language.

Module #4 introduces students to curve fitting. Students learn to fit a straight line to a set of data points by two techniques: TRENDLINE and the LINEST function. The latter is more flexible. It allows them to fit any linearized fitting function to the data. They also learn about the concept of mean square deviation to evaluate a curve fit. Stress-strain data may be very appropriate to test a variety of curve fits.

Module #5 teaches students about numerical integration and differentiation. Students find area under a curve described by data points that may or may not be equi-spaced. This module also emphasizes the relationship between the area and the average of abscissa values. We also introduce students to forward, backward and central difference formulas to find first and second order derivatives. Our intent is to teach students to work with tabular data in a variety of ways.

Module #6 presents elements of Microsoft Word. Students are introduced to formatting text, inserting symbols and equations, creating tables, and importing Excel objects. A useful exercise at the end of this module is to have students write a short lab report using the stress-strain data of module #4.

Module #7 introduces students to algebra with Maple V Release 4. The Maple worksheet interface is very similar to Excel. Students learn to define functions, plot functions, solve an equation algebraically and numerically, and perform simple algebraic operations. The intent in this module is to show students how quickly they can explore a variety of functions with Maple.

In module #8, students do calculus with Maple. Coverage is limited to indefinite and definite integration, and differentiation of functions. We select examples to demonstrate to students that not all functions can be integrated analytically. However, proper integrals can be approximated numerically.

Concluding Remarks

The workshop was offered for the first time this Fall. We have asked students to evaluate the workshop in regards to its content, instruction and teaching aids. An informal survey indicates that the workshop was successful. Students liked the users' guide, and have offered suggestions to improve it. In the last two years, students returning from their cooperative employment after having taken **441** and **440** inform us that they used their computer skills on co-op. Furthermore, many students report that they trained their co-workers in the use of computer tools and techniques while on co-op assignment. Many engineering instructors inform us that students are submitting improved home assignments and lab reports. Thus, it appears that our initiatives are in the right direction.

Acknowledgments

This project was supported by a 1996-97 Provost's Productivity and Teaching Innovation Grant. The author gratefully acknowledges the assistance of Professor Robert J. Hefner in developing the workshop format and the Users' Guide.

References

1. S. K. Gupta et. al., *Integrating numerical computation into BSME curriculum at RIT,* ASEE St. Lawrence Section Meeting, Buffalo, NY, 1985.

2. S. K. Gupta, *Gaussian elimination with maximal row pivoting and scaling*, Proceedings of the 1988 ASEE Annual Conference, v5(1988)2205-2207.

3. S. K. Gupta and R. J. Hefner, *Creative problem solving strategies and tools*, ASEE St. Lawrence Section Meeting, Seneca College, ON, 1992.

4. R. J. Hefner and S. K. Gupta, *Problem Solving Tools in Engineering Education,* ASEE St. Lawrence Section Meeting, Rochester, NY, 1993.

5. S. K. Gupta, *Innovative techniques in teaching Numerical Computations sequence,* TBEEC Fourth Annual Meeting, Cocoa Beach, FL, 1992.

6. S. K. Gupta, *Revitalizing the Numerical Computations course sequence*, Proceedings of the 1993 ASEE Annual Conference, v2(1993)2000-2004.

7. S. K. Gupta, *Finite Difference techniques with Excel 5.0*, Simulation Series, v27#2(1995)135-138.

8. S. K. Gupta, *Innovations in the "Computational Methods" course sequence*, Innovations in Mechanical Engineering Curricula for the 1990's, ASME 1996 Curriculum Innovation Awards, The American Society of Mechanical Engineers, (1996)20-22.

9. S. K. Gupta and R. J. Hefner, *EMEM441 - Computer Tools, Users' Guide*, Rochester Institute of Technology, Rochester, NY, 1996.

SURENDRA K. GUPTA

Professor Vinnie Gupta has a BTech degree in Metallurgical Engineering, MS degrees in Computer Science, Mechanical Engineering, Metallurgical Engineering & Materials Science, and a PhD in Materials Science. In the Department of Mechanical Engineering at RIT, he teaches undergraduate and graduate courses in Applied Mechanics, Computational Techniques, and Materials Science.