Motivating Civil Engineering Students to Learn Computer Programming With a Structural Design Project

James D. Bowen
University of North Carolina at Charlotte

Abstract

Like many other schools, at UNC Charlotte a student's first Civil Engineering course includes instruction in a high-level computer programming language. For many years the language taught was Fortran, but for the last few years students have learned programming as part of instruction in the MATLAB modeling language. Inclusion of computer programming early in the curricula has been seen by the Civil Engineering faculty as a way of improving the students' skills in logical reasoning, application of technical knowledge, and quantitative problem solving. But teaching programming in the "Introduction to Civil Engineering" course has always posed a dilemma for the faculty. A primary objective of this course is to retain existing Civil students and attract other students who have yet to choose an engineering major, yet many of these students find traditional programming instruction to be dry, boring, and irrelevant to Civil Engineering. In an effort to make programming instruction more relevant, interesting, and engaging, students now write MATLAB programs as an integral part of a structural design project where groups of students compete against one another to produce a truss-style balsa wood bridge having the highest profit. Throughout the semester a series of homework assignments require students to write MATLAB programs that calculate separate bridge characteristics that determine the cost and benefit of their design, such as amount of wood used, number of bridge nodes, bridge mass, and estimated strength. Frequent computer laboratory help sessions and an automated grading system are used to give students abundant and immediate feedback on their programming assignments. The automated grading system is designed in such a way so that the correct answers to questions can vary from group to group based on the characteristics of the particular bridge. While students still grumble about having to learn programming, as they frequently did when Fortran was taught, introduction of the bridge design project has injected an element of enthusiasm and energy into the class, which are certainly desirable outcomes for an introductory class in Civil Engineering.

Introduction

At the University of North Carolina at Charlotte there are three engineering departments (Civil, Mechanical, Electrical and Computer) and an Engineering Technology Department. In the freshman year, all engineering majors take a common first semester course (ENGR 1201). This course, whose content is relatively new, serves as an introduction to the engineering profession and training in some of the skills needed for professional success[1]. The course has a number of learning objectives including developing team and computer skills, creative problem solving, and effective project planning that are addressed through a team-based conceptual design experience. The content and delivery method of ENGR 1201 are quite similar to other schools where the initial course in engineering is common to all majors[2,3,4]. The class is a mixture of lectures that give "engineering survival skills" (computer, library, professional) combined with a semester-long team project that requires planning, conceptual design, and oral and written presentations.

The common Freshman Engineering curricula originally included a second course[1], but this course has more recently reverted to the individual departments. Only the common course name (ENGR 1202) remains. This article discusses the development of the Civil Engineering version of the second semester Freshman Engineering course. Like the other two versions of the course, the course taught by the Civil Engineering Department has not had a fixed course content. Each Civil Engineering faculty member teaching the course had his or her own vision for the course, but there was one common thread; the course has always included instruction in computer programming.

The fact that the Civil Engineering version of ENGR 1202 has always included computer programming is a reflection of the value attributed to the instruction by the faculty. Even though the faculty recognize that few Civil Engineers go on to careers in programming, they nonetheless believe that teaching programming provides tangible benefits in improving the students' logical reasoning and problem solving skills. Up until recently the programming language taught was Fortran, but since the beginning of the 2001-2002 academic year we have taught Civil Engineers programming as part of instruction in the MATLAB[5] modeling language. Adoption of MATLAB as a high-level language for teaching programming has occurred at a number of schools lately[6,7]. In each case MATLAB is valued for its ease of use, its powerful set of analysis and visualization functions, its complete set of high-level language constructs, and its usefulness across the engineering curricula.

Teaching computer programming to Freshman Civil Engineers has always presented some practical challenges. In part because of the common curricula, many freshmen in their second semester have yet to declare an engineering major, and those who have declared face only minor obstacles in changing their major. For this reason, an additional objective of ENGR 1202 is to retain existing Civil Engineering majors, and to attract additional undecided students to Civil Engineering. A shortcoming of the previous versions of the course was that students did not like the Fortran instruction. In ratings of the course, students described the Fortran module to be dry, boring, and not useful. A second challenge of teaching Fortran was that it was difficult to devise a Fortran application that could be used part of the conceptual design project that was the focus of the class. Again, the students perceived Fortran as being only marginally useful to the project. Teaching computer programming was clearly not attracting students to Civil Engineering.

When the switch was made from Fortran to MATLAB, the opportunity was made to revamp the course to provide a more engaging and interesting means of teaching computer programming, while still providing the students with an introduction to the Civil Engineering profession. The revamped course is now organized around a hands-on, project-based experience where groups of students compete to produce a balsa wood bridge that has the highest profit. The overall design of the class is the subject of a previous article[8], but is also discussed briefly in the following section. Coupling the structural design project to homeworks that require MATLAB programming serves as the primary means of motivating students to learn programming. This coupling and several other key features of the course that serve to further motivate students to learn programming are then discussed in a later section. The article concludes with assessment of the course that has come from the experience of teaching it for the last four semesters.

Course Description

The course as it is presently conceived has been taught every semester since Spring 2002. The focus of the semester-long course is now a group-based structural design competition. Groups of one to three students compete against one another to produce the most "profitable" truss-style balsa wood bridge. The bridge benefit is determined using an "S" shaped function of bridge strength (Figure 1). Bridge costs are calculated by summing a fixed cost, a material cost for the wood used, a simulated construction cost based on the number of bridge nodes and member cross-section sizes, and a simulated foundation cost based on the mass of the bridge. An example calculation of costs, benefit, and profit is provided in Figure 2. During the semester the students develop a series of MATLAB programs that they can use to estimate the strength, costs and benefits of various candidate bridge designs. One of the computer assignments requires them to test five candidate designs and make x-y plots that compare the strength, cost, and profit of each one. The MATLAB instruction is supplemented with a two-week course unit that gives

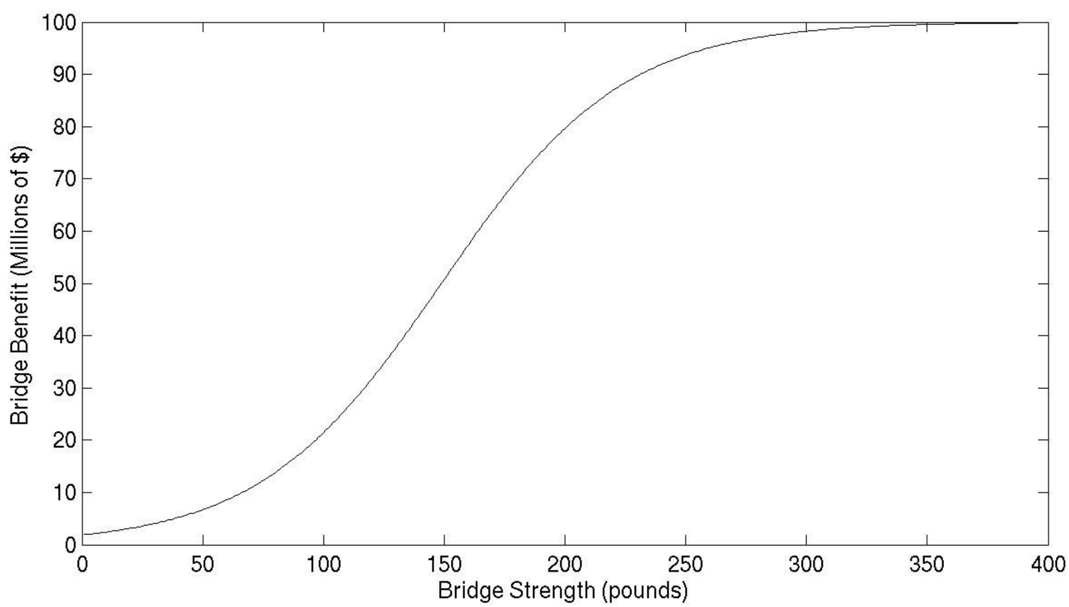
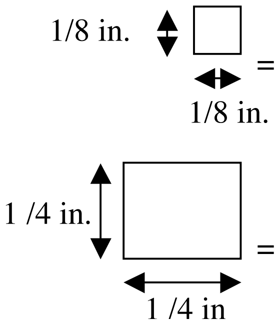


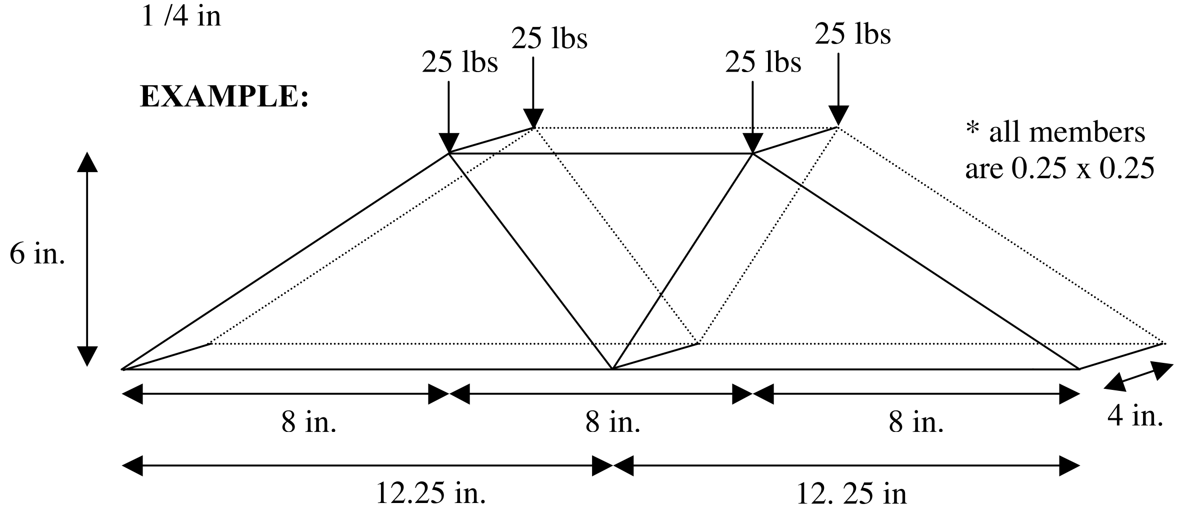
Figure 1. Bridge Benefit as a Function of Bridge Strength (Spring '04 competition)

students the equations needed to calculate compressive and tensile member strengths. These equations are introduced as part of a course unit on dimensional analysis. A slightly longer



Member Types = $100,000/member type
Decking, Pavement, Other Fixed Costs = $5,000,000
Engineering Fee = $500,000
Nodes = $100,000/Node
Weight Dependent Costs = $4e5*(model weight (g) – 40)

EXAMPLE:



From Matlab -> Sum of member length's = 66.42 inches/truss
Number of Nodes = 5 nodes/truss
Model Weight=67.2 grams (estimated, vol.=8.30 in^3 , 200 g/m^3 density)
Minimum Safety Factor = 0.46

Cost:

2 (trusses) * [66.42 (inches/truss) * $80,000/inch] = $10,627,200
2 (trusses) * [5 (nodes/truss) * $100,000/node] = $1,000,000
1 member type * $100,000/member type = $100,000
Weight Dependent Costs = $400,000 /gram * (67-40) grams = $10,800,000
Engineering Fee + Decking, Pavement, Other Fixed = $5,500,000
TOTAL COST = $28,027,200

Benefit: (see Figure *):**

*Strength = Minimum Safety Factor * Load on Truss * 2 Trusses*
Strength = 0.46 * 50 lb load on truss * 2 trusses = **46.0 lb** → **Benefit = $6,070,500**

Profit = Benefit – Cost = $6,070,500 - $28,027,200 = **$-21,956,700**

Figure 2. An Example Calculation of Bridge Costs and Benefit

course unit introduces students to the method of joints truss analysis. This concept is introduced through the property of static equilibrium of truss nodes. Together these two units enable students to perform hand calculations to calculate the loads on and strengths of members in simple triangular or trapezoidal trusses.

The end-of-semester bridge competition serves as a fitting climax to the course. Each group gives a short oral presentation describing their bridge, the division of labor within the group, and their design process. The presentation is followed by destructive testing of the group's bridge, using a small load frame and a digital readout load measurement device. Costs, benefit, and profit of each group's bridge are calculated in real-time so that the winning entry can be announced immediately after the last bridge is broken. Entries are ranked by using a weighted comparison of bridge profits (82%) and the group's accuracy in predicting the strength of their bridge (18%). Students use their MATLAB scripts to estimate bridge strength; some groups also perform preliminary destructive tests of their bridge to refine the strength estimate.

Motivating Students to Learn Programming – Key Course Elements

Integrating computer programming into a semester-long structural design project is the primary mechanism for motivating students to learn computer programming. Unlike traditional programming courses, students have an immediate, tangible reason for attaining programming skills. After three homework assignments that cover the most basic elements of MATLAB and its high-level programming language, the remaining four homework assignments provide students with information useful to their bridge design, as follows:

1. (Homework 4) Students use a custom MATLAB function to calculate the length and safety factor for each member of a truss that they design, given a specified load (50 lb.), and span (12 in.). A preliminary competition awards extra credit points for the three students whose trusses have the highest minimum safety factor (i.e., highest strength). The students are also required to build a truss according to their design using 1/8 in. by 1/8 in. balsa wood sticks that are provided to them.

2. (Homework 6) This assignment, and the remaining two are group assignments. Not all group members get the identical assignment, although each student must calculate in their MATLAB script all the variables needed to calculate bridge cost (amount of wood used, number of nodes, number of member cross-section sizes, estimated bridge mass). In this assignment students are free to use various member sizes from 1/8 in. by 1/8 in. up to 1/2 in. by 1/2 in.

3. (Homework 7) In this assignment, the MATLAB script written by each student must analyze a bridge that meets all contest rules. The bridge span must be at least 24 in. The script must simulate a downward vertical load from a 5 in. by 15 in. loading plate. A complete set of the contest rules can be found on the class web site[9]. In this script students must calculate bridge costs as in the previous assignment, and bridge benefit and profit.

4. (Homework 8) In this assignment students use a program loop to analyze five different designs that differ in member cross-section size. They must display bridge profit, strength, and mass for each candidate design using the x-y plotting features of MATLAB.

While integration of programming into the design project helps motivate students to learn programming, this course element in itself was not considered sufficient to produce a course that students and faculty would consider successful. Two particular concerns were that students might not attain sufficient programming skill so that they could use MATLAB in subsequent coursework and that the course would be incredibly time consuming for the instructor to manage. Concern about the learning outcome arose from faculty judgment regarding the background and interests of Civil Engineering freshman. Most of the students have little or no background in computer programming. In addition, they do not seem particularly interested in computing, and seem to feel that it is not a skill that is needed in the Civil Engineering profession. Many of them also seem to lack skill in diagnosing and testing faulty computer programs. Concern about the time required to administer the course stems from the sheer number of assignments to grade and the amount of time needed to help students with their assignments. Based on expected enrollments (approximately 100) and the number of homework assignments, approximately 800 MATLAB scripts would need to be graded each semester, with many of these assignments requiring some assistance of the student by the professor or the teaching assistant. In addition to meeting these challenges, it was desired to add elements of the course that would further motivate the students to learn programming. The following are course elements that are intended to provide additional student motivation for learning program, to help maximize student achievement of programming skills, and to minimize the time required to administer the course:

- Students are able to get immediate feedback on their programming assignments, and faculty can quickly grade assignments through the use of an automated grading system;

- Students have access to a set of custom MATLAB functions that assist in the analysis of their bridge design;

- Special attention is paid during the class to MATLAB features that students may find useful in future engineering coursework;

- Students work together on the bridge design project and the accompanying programming assignments, at the same time measures are taken to detect sharing or copying of programming assignments;

- Ranking of bridge entries from all previous semesters before the competition using the current set of rules allows students to gather information on good bridge designs and fosters healthy competition between groups of students.

Each of these key course elements is described separately in the following sections.

Use of an Automated Grading System

Anyone who has done the least bit of programming understands what a frustrating experience it can be. Learning programming is like writing in a new and foreign language where every sentence must be absolutely perfect to avoid catastrophic syntax errors. Finding and fixing these syntax errors is time consuming and frustrating for students. In addition, even once the program runs, there is no guarantee that it solves the assigned problem. Early on in the design of the course it was recognized that a system was needed to provide students with immediate feedback on their programming assignments whenever and wherever they were working on them. To provide this feedback, an automated grading system was devised[10] that allows students to check and grade their script while they work. To use the system, students enter the script name "check_script" at the MATLAB command line at which point a dialog box pops up and prompts them for their login ID and assignment number. The login ID is needed to determine which version of the assignment the student is working on. The student then browses through their directories to select the script to be checked, and then clicks on the "run" button. The script checking system then checks for syntax errors, and gives the line number if there is an error. If not, then the script is checked against the correct answers, and then reports to the student the number of correct answers and a numerical grade from 0 to 100. In this way students who use the system get immediate feedback and know exactly the grade they would receive for the script as written. The system is also used as part of an automated grading system used by the instructor to quickly grade all submitted scripts for each programming assignment, which greatly reduces the time spent grading. By providing abundant and immediate feedback, students find programming less frustrating and seemingly arbitrary, which helps motivate them to continue working on their script until an acceptable grade is produced.

Providing Students with a Set of Custom MATLAB Functions

While MATLAB has a large number of useful built-in functions, and is quickly learned, nonetheless, asking students to complete all of the programming needed to analyze the costs and benefits of a limitless variety of truss designs was considered to be unreasonably onerous. Luckily, user-defined custom functions can easily be programmed in MATLAB. With the proper programming, it is possible to write functions that appear to the user to be functionally equivalent to MATLAB's built-in functions. Several of these functions have been written and have been placed in one of the directories in the student's standard path at the MATLAB login whenever they use any of the computers in the College of Engineerning's local area network of PC's and UNIX machines. These functions are also available to the students for download if they choose to work on a home computer. Some of the functions provided to the students are as follows:

draw_truss — creates a scaled drawing of the truss showing all the members and nodes of the truss with the accompanying node and member numbers,

jbtrussf2 — performs the method of joints structural analysis and calculates the loads, strength, and length of each member in the truss, get_ben calculates the bridge benefit given an estimated bridge strength, and

find_cost calculates bridge cost given the number of nodes, amount of wood used, and size of each member in the bridge.

With these functions, students are able by the third month of the semester to perform all of the cost and benefit analyses needed to evaluate a candidate bridge design. Providing these functions allows students to produce useful MATLAB scripts early enough in the semester so that their programs can be used to evaluate candidate bridge designs, which helps motivate them to use their programs as an integral part of the structural design project.

Emphasizing in Class Lectures MATLAB Features Useful to Future Coursework

One of the most persuasive arguments for learning MATLAB is the wide variety of tasks that can be easily performed with simple MATLAB scripts. MATLAB is a complete modeling environment and has an incredible number of built-in functions for tasks such as statistical analysis, optimization, array manipulation, signal processing, and data visualization. During the semester, every opportunity is taken to point out these useful features, and to explain how they might be used within the engineering curricula. For instance, an early homework assignment uses the linear algebra capabilities of MATLAB to calculate vector dot products and solve systems of equations. The usefulness of these features in later math and engineering classes is mentioned during the classroom lecture. Later in the semester students learn about the x-y plotting features, and use these features to plot their bridge's expected profit for five member size cases (Figure 3). Again, mention is made during lecture of how these features could be used in later coursework. By giving students concrete examples of the advantages of learning MATLAB, their motivation to become proficient programmers is increased.

Allowing Group Work While Checking for Cheating

Most programmers realize that it often takes a second set of eyes to spot even simple problems in a computer program. Even experienced programmers often get frustrated looking for errors that seem basic and obvious to others. It seems that new programmers have a particularly tough time in finding and diagnosing problems, as they lack the experience to know what are the common programming pitfalls. Allowing students to collaborate on programs can help alleviate this problem, yet allowing collaboration also increases the possibility that students will cheat by copying assignments. The course and the automated grading system that assists with course administration have been designed with several specific measures to discourage cheating. Four versions of each assignment are created, and are randomly assigned to students based on their login ID. These assignments are similar enough to allow students to collaborate, yet differ enough in detail so that simple copying will result in a failing grade for differing versions of the assignment. In addition, each student works in their own distinct file space, which makes copying somewhat more complicated. The assignments are also written to require the student to personalize the script by setting a script variable to their personal login ID. During grading, the automated grading system checks to see if the proper ID has been set, otherwise the student receives a zero for the assignment. While these measures are relatively easy to defeat, they seem

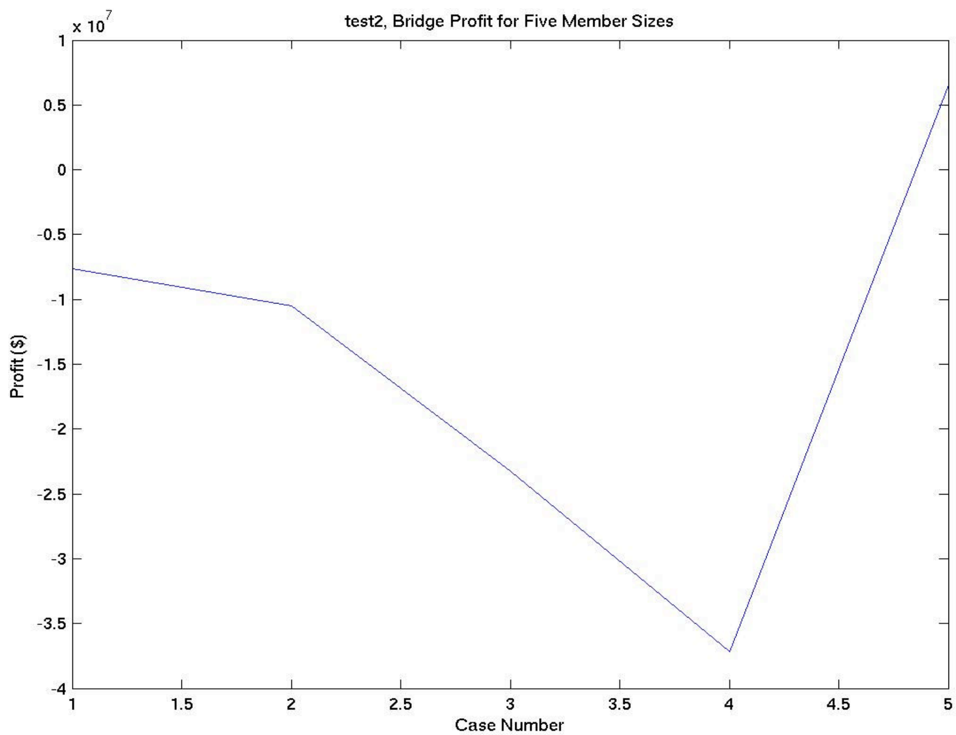


Figure 3. A Sample x-y Plot Produced as Part of a MATLAB Programming Assignment

to work, judging by student attendance and conduct during computer laboratory help sessions. Students do collaborate, and therefore benefit through the debugging assistance of their coworkers, but produce individually written scripts for their particular version of each assignment.

Ranking All Previous Entries in the Competition Using the Current Rules

A total of 85 bridges have been entered in the contest over the past four semesters. As a way of spurring interest in the contest in current and former students, all of these entries are ranked on the class web site (Table 1). Each semester the rules are changed so that a particularly good design cannot be reused from semester to semester. Each semester some of the parameters affecting the cost and/or the benefit calculation are modified. For instance, as compared to the previous semester, the Fall 2003 rules used a higher unit cost for the larger wood (1/4 in. by 1/4 in.), and a higher bridge cost per gram, while lowering the strength needed to achieve the maximum benefit of 100 million dollars. In the Spring 2004 semester, the strength needed to achieve maximum benefit was increased, while holding the unit costs constant. These changes have the effect of shuffling the entries from their original ranking. Even with the shuffling, however, the collective performance of the various entries provides valuable information and a performance benchmark for the current set of students. By having some idea of what it takes to make a winning bridge, students are motivated to look for the optimal design for the current rules. The search for the optimal design is done using the students' MATLAB scripts, thus ranking and reporting the performance of previous entries helps motivate students work on their own MATLAB scripts.

Table 1. Top Ten Bridge Contest Entries After Four Semesters

Term & Group	Length Large (in.)	Measured Mass (g)	Cost (10^6 $)	Measured Strength (lb)	Profit (10^6 $)	Profit Rank	Prediction Error (%)	Weighted Score	Overall Rank
Sp 02, A	125.0	79.2	36.481	323.0	62.57	1	26.32%	5.05	1
Sp 03, A	201.8	58.7	32.824	197.1	45.57	4	17.96%	5.80	2
Fa 03, A	104.0	50.0	20.002	175.6	47.16	2	43.05%	7.55	3
Sp 02, B	257.0	99.5	53.159	363.6	46.52	3	50.50%	8.70	4
Fa 02, A	223.0	97.0	49.140	225.0	39.28	10	11.11%	8.95	5
Sp 03, B	173.2	83.1	38.696	206.0	43.45	5	49.03%	9.80	6
Fa 02, B	59.5	46.5	15.619	162.0	43.12	7	38.89%	10.45	7
Fa 02, C	166.0	50.0	25.121	178.0	43.44	6	49.44%	10.65	8
Sp 03, C	218.0	84.6	44.480	199.8	35.11	12	16.62%	11.10	9
Sp 02, B	236.9	96.4	49.510	211.6	34.72	13	14.93%	11.65	10

Course Assessment

The course as described in this article has been taught four times. A number of conclusions can be made about the course at this point. Numerical ratings of the overall course have improved markedly from the few years before the course redesign. In written course ratings, students often comment on how much they enjoyed the course and the balsa wood bridge contest in particular. There is an obvious enthusiasm in the students as they design, test, and build their bridges. Most students seem to genuinely enjoy the semester-ending contest. No student has ever commented, however, on how much they enjoyed the programming instruction part of the course. They do seem convinced by the end of the semester that it could be useful in other courses, and many of the students seem to be capable programmers by the end of the semester. It is not known how many of these students have had previous programming experience. Quite a few former students have come by asking for assistance with MATLAB programming that they were working on as part of a subsequent math or engineering course. This indicates that students are using MATLAB, even though no formal steps have been taken to implement it across the curricula, as has been done at other schools[11]. Results from the assessment of learning outcomes related to MATLAB programming have been somewhat disappointing so far. These assessments indicate the just over half of the students attain a minimum capability to write MATLAB programs to solve simple quantitative problems. With these assessments in mind, a number of changes have been made in the portion of the course that provides MATLAB instruction. We have also experimented with different MATLAB texts to use for the course. It is hoped that these measures will eventually improve student learning so that most students attain the capability to use MATLAB programming in subsequent coursework.

Bibliography

1. Shelnutt, W., Allen, M., Badami, V., Buch, K., et al. 1998. A Multidisciplinary Course Sequence Stressing Team Skills, Conceptual Design, Creative Problem Solving, Professional Practice, and Computing Skills for Students Entering the William States Lee College of Engineering. *Proceedings of the 1998 American Society for Engineering Education.* Washington, D.C.: American Society for Engineering Education.

2. Craven, K., Wayne, S., and Stiller, A. 1999. The Freshman Engineering Experience at West Virginia University. *Proceedings of the 1999 American Society for Engineering Education*. Washington, D.C.: American Society for Engineering Education

3. Gardner, J.F., Ackler, H.D., Paris, A.J, and Moll, A.J. 2002. Rethinking First Year Engineering at Boise State: Assessment and Improvement. *Proceedings of the 2002 American Society for Engineering Education*. Washington, D.C.: American Society for Engineering Education.

4. Sorby, S.A., Monte, A.E., and Hein, G.L. 2001. Implementing a Common First Year Engineering Program at Michigan Tech. *Proceedings of the 2001 American Society for Engineering Education*. Washington, D.C.: American Society for Engineering Education.

5. MATLAB, MATrix LABoratory is a trademark of The Mathworks, Inc., Natick Massachusetts.

6. Devens, P.E. 2000. MATLAB & Freshman Engineering. *Proceedings of the 2000 American Society for Engineering Education*. Washington, D.C.: American Society for Engineering Education.

7. Herniter, M.E., and Scott, D.R. 2001. Teaching Programming Skills with MATLAB. *Proceedings of the 2001 American Society for Engineering Education.* Washington, D.C.: American Society for Engineering Education

8. Bowen, J.D. 2003. Using a Hands-On, Project-Based Approach to Introduce Civil Engineering to Freshmen. *Proceedings of the 2003 American Society for Engineering Education.* Washington, D.C.: American Society for Engineering Education

9. URL: http://www.coe.uncc.edu/~jdbowen/1202/2004/spring/index.html

10. Bowen, J.D. 2003. An Automated Grading System for Teaching MATLAB to Freshman Engineers. *Proceedings of the 2003 American Society for Engineering Education.* Washington, D.C.: American Society for Engineering Education.

11. Perrry, W.B., Barocas, V.H., and Clough, D.E. 1999. Implementing Computational Methods into Classees throughout Undergraduate Chemical Engineering Curriculum. *Proceedings of the 1999 American Society for Engineering Education*. Washington, D.C.: American Society for Engineering Education.

JAMES D. BOWEN

James D. Bowen is an Associate Professor in the Civil Engineering Department at UNC Charlotte. He received his Ph.D. degree from the Massachusetts Institute of Technology. Dr. Bowen teaches MATLAB programming, hydraulics, aquatic chemistry, and water quality modeling. His research interests include water quality and eutrophication modeling, model uncertainty analysis, and the microscale fluid motions around phytoplankton cells.