

# **AC 2007-2107: ENABLING CURRICULAR INTEGRATION THROUGH MULTI-COURSE ASSESSMENT**

**Daina Briedis, Michigan State University**

**Mark Urban-Lurain, Michigan State University**

**Robert Ofoli, Michigan State University**

**Dennis Miller, Michigan State University**

**Jon Sticklen, Michigan State University**

# **Enabling Curricular Integration through Multi-Course Assessment**

## **Introduction**

In 1991, ABET was faced with a major challenge of transforming from a rigid set of accreditation criteria to evaluation criteria based on constituency focus, continuous program improvement, and outcomes in student learning.<sup>1</sup> To accomplish this change, ABET underwent a massive reformation process. In 1997, as a result of this process, ABET adopted Engineering Criteria 2000 (EC2000), which focused on program evaluation based on what is learned rather than what is taught. At the core was a continuous improvement process driven by the specific and unique missions and goals of individual institutions and programs. Questions remain in the minds of most engineering faculty and administrators as to whether the requirements of the “new” criteria are accomplishing their ultimate purpose. Early evidence<sup>2</sup> suggests that they are; students are now better prepared for engineering careers than they were ten years ago.

The ABET outcomes-based criteria were also instituted to give engineering programs the freedom to exercise innovation in curriculum design, rather than be confined by rigid criteria. This paper offers preliminary evidence that the regular assessment of the ABET-designated outcomes has opened the eyes of our faculty to issues in student learning that may not have been considered before. While initial assessment was conducted at the disciplinary course level, improvement actions have been more far-reaching including non-trivial course and program improvements, interdepartmental faculty collaboration, redesign of course content, and renewal of faculty interest in improved classroom pedagogy. This paper reports on the assessment-based approaches used to implement curricular change and the benefits that have resulted to date. In a broader sense, this paper proposes a model process by which faculty in service courses and program faculty may be encouraged to collaborate in establishing sound links between the content of service and disciplinary courses. In this paper, our emphasis is on the smooth integration of math and computational tools throughout our curriculum. The approach used may also be applied in integrating other “basic” skills (e.g., writing, statistics, biology, chemistry) downstream into disciplinary curricula. The theoretical basis for the suggested changes is also presented.

## **Assessment as the Foundation**

The engineering programs at our institution have been conducting outcomes-based assessment since before the first “EC2000” visit conducted here in 1998. Although the evaluation and assessment processes themselves have been redesigned and simplified, assessment has become a regular and fairly accepted role for the faculty members who teach our undergraduate courses. As part of the regular assessment cycle, it has been encouraging and mildly surprising that assessment of our chemical engineering program outcomes has resulted in groundwork for curricular improvements involving interdepartmental faculty in multiple courses.

Over the past several years, assessment data taken in our chemical engineering courses have identified specific weaknesses in students' mathematical and computer-based problem-solving skills. The weakness appeared to worsen as students progressed through the curriculum. These results were validated by student self-assessment (surveys) and by cooperative education employer.

All chemical engineering students are required to take the standard battery of mathematics courses and an introductory computing tools course (CSE 131). The computing course was originally designed to provide students with an understanding of problem solving approaches, ethics, and the use of basic computing tools for in technical problem solving. The early version of the course emphasized the "engineer's toolkit" and included coverage of Excel and MATLAB.

In the 2003-04 academic year, a fairly large number of students voiced concern about the need for the introductory computing course. That year, as part of the annual year-end survey of chemical engineering students at all levels, specific questions were included to address concerns over this course.

Survey results indicated that almost 85% of our students took CSE 131; other students transferred in with credit in other computing courses. When the students that had taken CSE 131 were asked about the utility of this course for chemical engineering, almost 70% of them reported that less than 50% of the course material was useful in chemical engineering courses. In another question, almost 70% of the students claimed that as much as 70% of the course material was already familiar to them.

Similar results were observed in a 2004-05 survey, in which students further elaborated on the frequency of their use of MATLAB beyond the CSE 131 course. On a scale from 1 (never used) to 5 (most often used), MATLAB use was ranked at 1.7. Clearly this indicated that students perceived that CSE 131 held no relevance to the chemical engineering program.

Additional information gathered from student focus groups indicated that sources of the weakness in skills and in negative student attitudes toward computing and mathematics courses might be two-fold:

1. Timing: A gap of at least one year existed between the basic math courses taken by first- and second-year students and the offering of our own applied mathematics courses. Students were unable to see the relevance of the mathematics courses to disciplinary problem-solving and had forgotten some of the common mathematical techniques due to lack of use. An even larger gap existed between the first-year computing course and any significant disciplinary applications of computational tools later in the curriculum.
2. Reinforcement and integration: Besides the timing gap described above, except for the extensive use of Excel, software in which most students were already proficient, most of our disciplinary courses did not incorporate the use of the tools

learned in this introductory courses into the problem-solving schemes offered in upper level courses (specifically MATLAB). Although other software packages were used in chemical engineering courses, including ASPEN, Control Station, Polymath, and Fluent, the relevance of MATLAB to chemical engineering had not been established. The reasons for student skepticism about the relevance of CSE 131 were clear.

The first problem, which is not the topic of this paper, was solved simply by moving a revised version of our transport phenomena course to an earlier point in our curriculum; we also incorporated significant use of MATLAB for the solution of ordinary and partial differential equations. Results of this change appear to be positive, but are not discussed in this paper.

The second problem, the topic of this paper, was a more serious problem in that one of the most important barriers for students deciding not to pursue engineering studies is their perception of “disconnectedness” in engineering curricula.<sup>3,4</sup> In this case, the disconnect was between the computational tools and their implementation in the student’s engineering discipline of choice. Our attempted solution to this disconnectedness has developed into a fruitful collaboration between the introductory computing course faculty and several engineering programs. We established a stronger link between the introductory computing course and its importance in upper level disciplinary courses, not only for students, but for faculty as well. Many faculty members have been stimulated to add MATLAB in their courses because of a “jump-start” that was provided by a refresher short course offered by the computer science faculty; this short course will be described later.

More disciplinary examples and specific content have been included in, or excluded from, CSE 131. Less emphasis has been placed on Excel; and instruction for advanced Excel features is provided in tutorials rather than during class time. However, for the long-term, chemical engineering faculty also felt that change was needed in our own disciplinary courses. Hence, our goal now is that our students will be proficient in using MATLAB to solve chemical engineering problems by the time they graduate. Most importantly, we are on a path of curricular reform that we believe will substantially improve student learning; repeated application of the same mathematical and computational tools will lead to student use of the tools in a natural and “second-nature” way, thus allowing faculty and students to focus more on chemical engineering course content.

The study of the results of this improvement process is still underway. This paper reports some early assessment results of student performance and confidence in the application of MATLAB in chemical engineering. We believe that our collaborative and outcomes-based approach will serve as a model for other programs undertaking curricular changes, needing to motivate faculty and administrators, and seeking improved student learning.

## **Rationale**

A distinctive attribute of engineers is their problem-solving ability. Indeed, ABET criteria<sup>5,6</sup> reflect this since at least five of the eleven outcomes relate directly to problem-solving skills. Many engineering curricula have focused strongly on providing students a through grounding in the basics of a given discipline as delivered through lecture. A steady slide ruled by “content tyranny” to increased reliance on “lecturing about” more and more technical material is a common symptom of the huge amounts of information in the engineering disciplines. Yet with so much content to master, we as faculty frequently forget that effective problem solving is predicated on *integrated understanding* of technical material.

Froyd and Ohland<sup>7</sup> emphasize the need for integrated engineering curricula to help students build connections between topics. This is particularly important for establishing relevance of basic math and science courses (first-year topics) in engineering, engineering practice, and engineering careers. While most engineering faculty recognize that the transfer of information from basic to applied courses is intended, it is not usually occurring to the degree it should.

The specific problem at hand is the integrated student understanding of computer-based computational tools that strongly support technical problem solving. Computational software such as MATLAB, Polymath, and Mathematica are all versatile and powerful. However, most faculty, because of their varied backgrounds, have learned (or developed!) one or another of these computational tools. It is not surprising that the choice of computational software for problem solving in typical undergraduate courses will reflect a professor’s preference. When a student moves from one course to another, the implicit assumption is that the student can simply “pick up” a different computational tool and apply it to assigned problems—that a student having learned the basics in one computational environment would be able to transfer that understanding to another computational setting. Students must have a threshold level of understanding to support the transfer from one environment to another. When knowledge or skills are taught in a variety of contexts, students are much more likely to transfer from one context-specific example to another.<sup>8,9</sup> Much as a pilot trained on a Boeing 777 aircraft would never be asked to take over an Airbus A380, this assumption of easy transfer of understanding is flawed.

Modern computational software packages have many features in common. However, students experience a steep learning curve jumping from one tool to another. This learning curve comes at the expense of course content, so frequently no course time is allowed for students to climb this learning curve. The versatility and complexity of modern computing tools coupled with the desire of faculty to select their “favorite tool” leave undergraduate students at a disadvantage.

The result is that students seek cookbook approaches to particular assigned problem or work in teams where only the one “expert” performs the computer calculations. Both approaches prevent the student from learning enough about the tool to understand how it

may be used in a different problem setting. In order to support any transfer of understanding from one context to another, a student must thoroughly understand and integrate knowledge at a conceptual level from the very first step.<sup>9</sup> For undergraduate students to become comfortably competent with MATLAB—or any other engineering tool—repeated use in different contexts is mandatory. They must “use it or lose it.”

About eight years ago as part of our response to negative feedback from students, this had already happened in our program with the application of the process modeling software, ASPEN. Seniors were spending a significant portion of class time in the capstone design course learning to use the software, thus compromising coverage of other important course topics. Students were spending extra time out of class playing catch-up with ASPEN proficiency. Therefore, ASPEN was implemented as a problem-solving tool in courses at all levels of the curriculum, most recently including our freshman course as well. As a result, ASPEN use has become comfortable and second-nature to all chemical engineering students allowing faculty and students to focus more on important course content. Our expectation is that this will happen with MATLAB as well.

### **Implementation**

The process for MATLAB integration began with approval of proposed curriculum changes in the Fall, 2004 semester. While not all faculty members accepted this challenge for their own courses, some faculty in several key courses became involved. The vertical integration was partially accomplished in a few pilot courses and extended to additional courses in the following year.

To jump-start and sharpen faculty MATLAB skills, the instructor of the introductory computing course (co-author JS) prepared and taught several “QuickStart” workshops in the Summer, 2005. They were conducted at a convenient time that allowed faculty several weeks for fine-tuning course preparation for MATLAB applications. Forty percent of the faculty teaching in the chemical engineering and materials science programs participated in the two half-day sessions. The QuickStart workshops included screen movies, personal mentoring, disciplinary examples, and college-funded teaching assistant support throughout the following two semesters. The teaching assistants helped faculty with troubleshooting and with the development of course-specific applications of MATLAB. It is this extended support that was the key to prompt implementation of the plan.

In the first year, MATLAB was used in the new sophomore transport phenomena course, ChE 210, for the solution of ordinary and partial differential equations of various types. These included first order differential equations (initial value problems) describing unsteady state macroscopic material and energy balances, second order ordinary differential equations representing steady state heat conduction and diffusion, and second order partial differential equations describing unsteady state heat conduction in solids. In several cases, solutions to these problems were generated by students using finite difference techniques such as Euler’s method as well. Students were then able to realize the advantages in computation and presentation of solutions offered by MATLAB.

MATLAB was also implemented in the junior level mass transfer and separations course in three problems: for phase equilibrium calculations for x-y and T-x-y phase diagrams, to determine optimal-state locations for a two-feed distillation column construction, and to design a two-column distillation process to produce anhydrous alcohol from fermented beer. In the senior year, MATLAB was incorporated into the process modeling and controls course (ChE 432). In this course MATLAB use was straightforward in such applications as symbolic solution of ordinary differential equations, plotting, inverting functions to and from the Laplace domain, finding roots of polynomials, creating and using transfer function models, generating dynamic system responses, and plotting root locus diagrams.

In the second year of the project, integration was expanded to include CHE 201, the material and energy balances course. Here students were assigned a project using ASPEN. The goal of the project was to maximize profits from a process that converted methanol and propionic acid to methyl propionate and water. The students designed their own process flow sheets and varied input flow rates to identify the conditions that would lead to maximal profits. Each group was asked to provide at least three different process simulations, each building from the previous in some manner. For one of these processes, the students were asked to write the steady-state, continuous material balance equations required to solve the process. They were then asked to use MATLAB to solve the linear system of material balance equations.

The intent of the assignment was for students to build the necessary matrices in MATLAB and use "back division" to solve the linear system. An example solution was posted on the course website prior to assigning the project. No in-class teaching on the solution was done, and, when the project was assigned, no specific rules for the use of MATLAB were described. While many of the students used the example as a model and solved the linear system as expected, other solutions were provided by the students that took advantage of MATLAB programming as well. In general, the students who had taken the introductory computing course that introduces MATLAB seemed to enjoy applying the tool in this context. It is important to note, however, that a number of students from other departments and colleges (approximately 1/3 of the class) are required to take CHE 201. This means that 24% (14 students) of the ChE 201 students had never been exposed to MATLAB prior to its introduction in this course, and another 15% were taking the computing course concurrently. Those students tended to rely on teammates who had MATLAB expertise and may not have really gained from the project. This is addressed further in the data analysis section.

## Assessment

The goal of the assessment of these students was to determine whether the vertical integration was helping students build confidence and skills in applications of MATLAB to problem solving. The necessary Institutional Review Board approvals and consent forms were obtained from all students. Pre-course surveys (called pre-surveys) were administered within the first two weeks of the beginning of the semester; post-course surveys (called post-surveys) were administered during the last week of the semester.

Surveys for all courses were identical even though some of the questions were renumbered and not all questions were relevant to all students. The same questions were analyzed both years. Four *general* (attitudinal) questions dealt with student confidence in and perceptions of the importance of MATLAB in their learning; answers were given on a Likert scale from 1=disagree to 7=agree. Fifteen additional questions asked about specific MATLAB skills with responses on a scale as follows:

- 1=I do not know what this means.
- 2=I know what this means, but do not have any idea how to do it.
- 3=I know what this means and have a vague idea of how to do it.
- 4=I know what this means and am pretty sure I can figure out how to do it.
- 5=I know what this means and am certain I can do it.

Examples of each category of questions are shown below.

General Questions (Scale: 1=disagree to 7=agree)	Task-Specific Questions regarding student's ability to use MATLAB to . . . (Scale 1 to 5 as illustrated above)
I believe that MATLAB will be an important tool for my upper division courses.	construct a FOR loop if I need to.
I believe that mastering MATLAB will help me in finding a job once I graduate.	solve an ODE initial value problem using ode45.
I feel confident that I can extend my knowledge of MATLAB if I need to.	invert functions to and from the Laplace domain.
Learning MATLAB has helped me to organize my understanding of other subjects (like physics).	solve a set of N simultaneous, linear equations in N unknowns.
	. . . plus eleven other questions

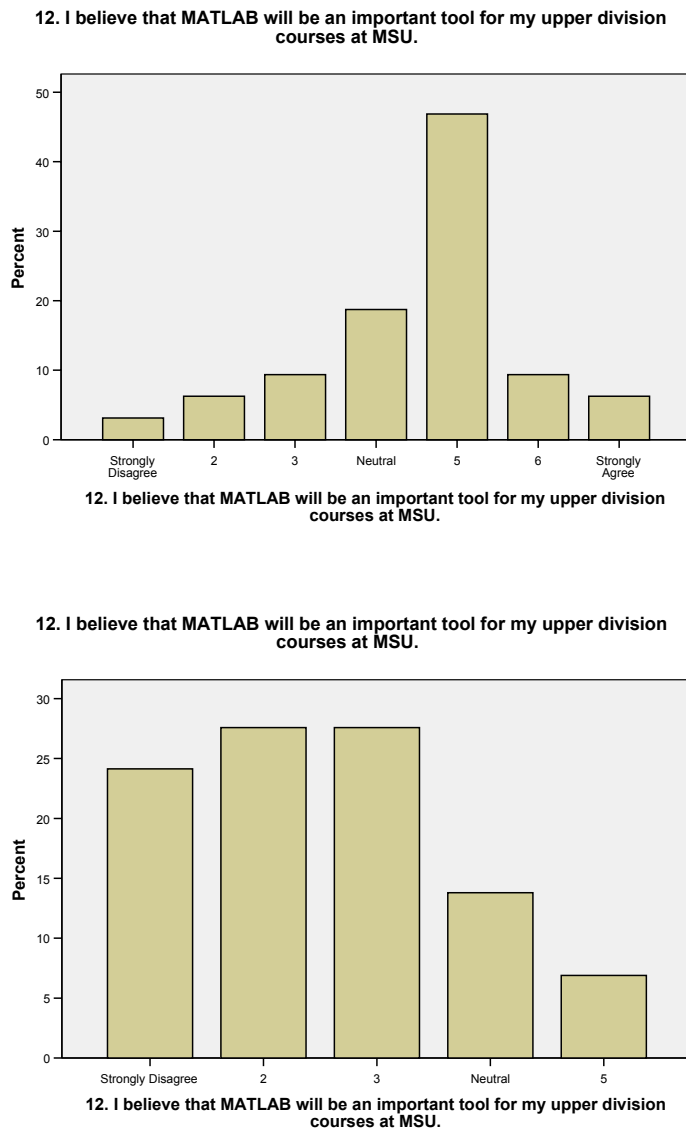
Pre- and post-surveys were analyzed using SPSS and correlations to student performance and personal data (gender, ACT scores, GPA, semesters since taking CSE 131, and other parameters). The SPSS analysis was based on strength of the correlation ( $r$ ) and a significance level of  $p < 0.05$ .



## Results

In the first year of assessment, pre-course surveys showed that ChE 210 (applied math course) students (n=33, mainly sophomores) showed a significantly greater confidence in MATLAB than ChE 432 (process control) students (n=38, seniors). A strong negative correlation was demonstrated between student confidence in MATLAB and the number of semesters since taking CSE 131—an average of 2.5 semesters for the sophomores and 8.4 semesters for the seniors. The clear implication here is that the proximity to the early exposure to MATLAB boosted student confidence in the use of this tool. These results are shown in Figure 1 below.

Figure 1. Pre-Course Confidence in MATLAB as an Important Tool, Spring 06: ChE 210 (2.5 semesters) and ChE 432 (8.4 semesters)



The post-course surveys showed little difference between the assessment of confidence in MATLAB as a useful tool for ChE 210 and ChE 432 students. This suggests that ChE 432 students had essentially “caught up” in their confidence level due their use of MATLAB in ChE 432. Their maturity as seniors may also have been a factor.

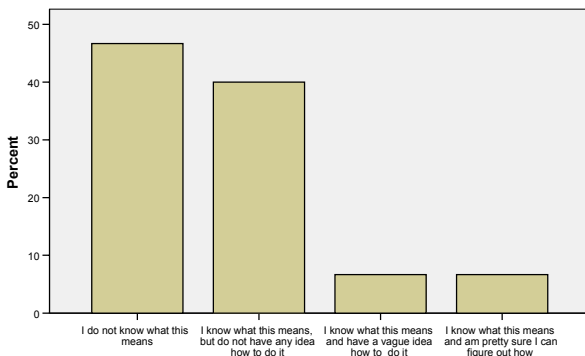
In knowledge and skills gained in ChE 210, positive correlations were observed between confidence in the following areas:

- Solution of ODEs using the function ode45
- Solution of initial value PDEs with pdepe
- Solution of two-point boundary value problems
- Solution of boundary/initial value problems

One example of these improvements is demonstrated in Figure 2 and is, of course, encouraging in that students improved in the desired areas. Results were also positively correlated to performance on homework.

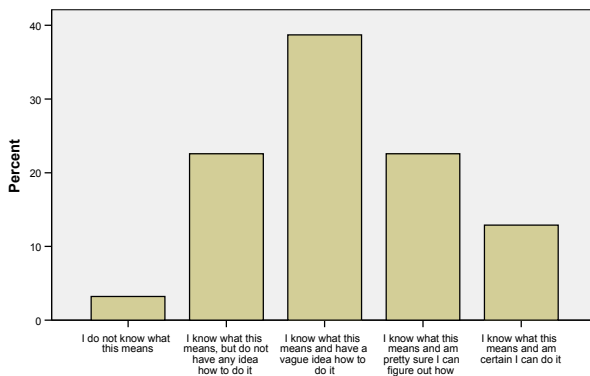
Figure 2. Pre- and Post-Survey Comparison of ODE Skills  
ChE 210, Spring 06

19. Using MATLAB, I can solve an ODE boundary value problem using bvp4c.



19. Using MATLAB, I can solve an ODE boundary value problem using bvp4c.

8. solve an ODE boundary value problem using bvp4c.

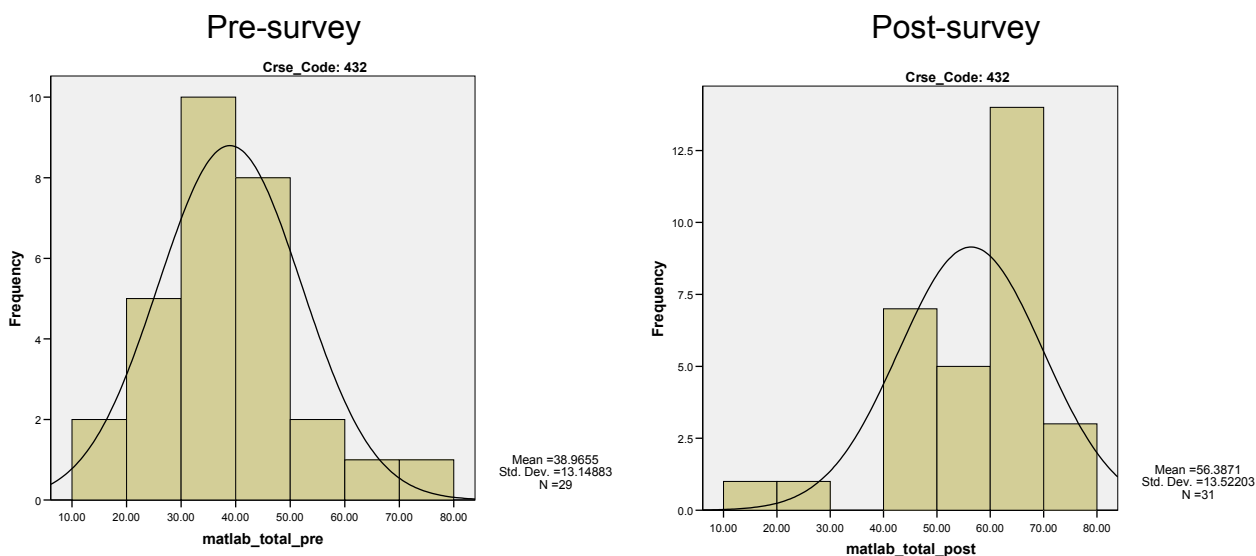


8. solve an ODE boundary value problem using bvp4c.

Similarly (not shown here), ChE 432 students showed gains in ability to invert functions to and from the Laplace domain, find roots of polynomials, and test the dynamic response of a system using MATLAB.

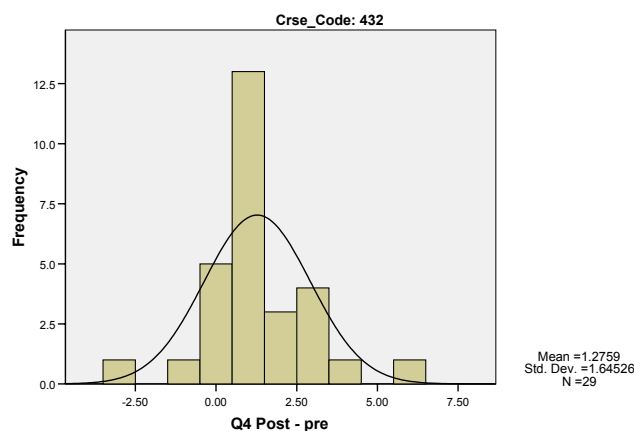
In the second year of assessment, only ChE 201 (material and energy balances) (n=59 pre, 51 post, mainly sophomores) and ChE 432 (n=29 pre, 31 post, seniors) were assessed. Results from ChE 210 had to be eliminated because of a surveying error. In this assessment cycle, more positive results were observed in ChE 432 than in ChE 201. In the summation of all responses to all questions for all ChE 432 students, they showed a dramatic increase (increase in positive responses on a Likert scale) in both confidence and self-assessment of their MATLAB skills between the pre- and post-surveys. The pre- to post-survey mean increased from 39 to 56. The means for ChE 201 students did not change significantly.

Figure 3.  $\Sigma$  of all responses, ChE 432, F 06



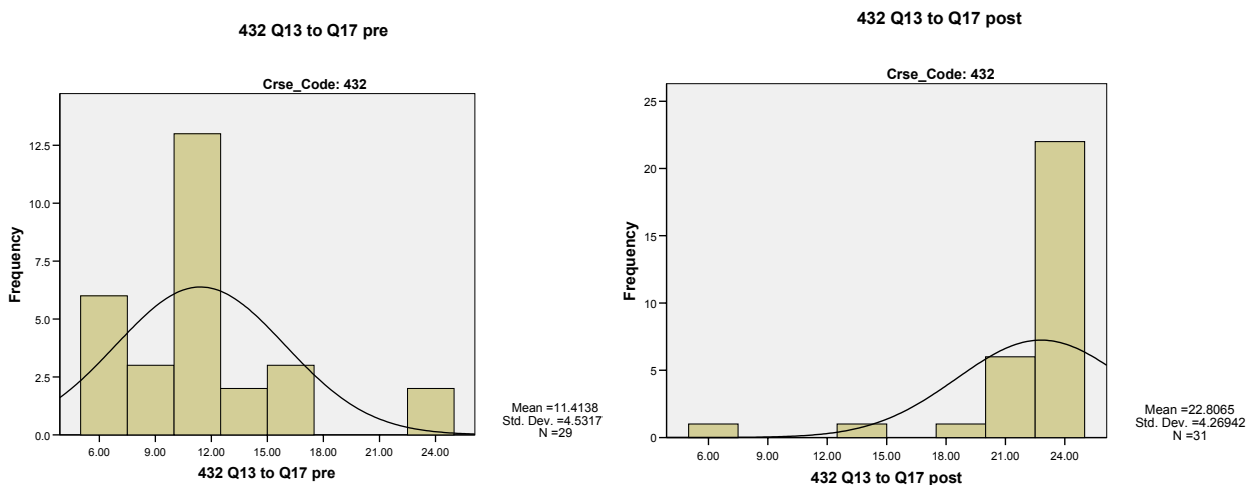
More specifically, ChE 432 students gained in confidence and attitude about MATLAB over the semester. Evidence is provided in a student-by-student comparison of the difference between post- and pre-survey results for confidence in their abilities to apply MATLAB and the effect of learning MATLAB in understanding other subjects (Figure 4). In these cases the mean of the response increased about one unit for each question (responses for only one question are shown in Figure 4). This supports the idea that use of a tool encourages transfer of the skills to other contexts. While the direct performance data are not yet available, confidence is certainly an important factor in willingness to apply tools to new situations.

Figure 4.  $\Delta$  in MATLAB as a tool in organizing understanding of other subjects;  $\Delta$  in mean = 1.3; ChE 432, F 06



Students also showed positive gains in the process controls course in understanding the application of MATLAB to course-specific skills. The mean of responses to skills-specific questions increased from 11.41 to 22.81 between the beginning and end of the course (Figure 5). A moderately positive correlation was demonstrated between responses to these questions and actual performance on MATLAB homework, yet there was no correlation between the course grade and responses to the skills-specific questions.

Figure 5. Pre- and Post-Survey Comparison of MATLAB Skills, ChE 432, Fall 06



The confidence boost gained in course-specific skills also rippled to other areas. ChE 432 students responded with increased post-survey confidence about skills that were not specifically covered in the course.

Additional interesting results and insight are provided by the regression analysis that was done on the total post-survey results for Fall, 2006 data (Figure 3). The ChE 201 post-survey results (not shown) were strongly correlated to the number of terms since the students had taken the introductory computing course, yet not much difference was observed between pre- and post-survey means. This is in contrast to the ChE 432 results that showed a dramatic change in responses between pre- and post-survey totals (Figure 3), but demonstrated no correlation to the number of semesters since taking CSE 131. For ChE 201, it is possible that student attitude and confidence in MATLAB did not change significantly during the course because the course did not include specific instruction on MATLAB. ChE 432 did include specific instruction on MATLAB, which may have been the cause of the significant increase in student MATLAB confidence.

As mentioned earlier in the paper, the student population in ChE 201 also included students who had not taken the introductory computation course. The statistical results showed clear differences between the students who took CSE 131 before ChE 201 and those who did not. On every pre-course survey question, the differences between these groups were significant; the students who took CSE 131 at anytime before ChE 201 scored higher. By the end of the class, the post-survey differences between the two populations largely disappeared. The pre/post-survey difference was significant; students who hadn't taken CSE 131 had more to gain, but they started with low confidence and self-assessment, which is to be expected. Even their relatively limited experience with MATLAB in ChE 201 did much to raise awareness and confidence.

### **Conclusions**

We have presented the first few chapters of a success story in which routine program assessment has resulted in a ripple effect in curricular integration and program improvement. The collaboration between the computing faculty and several engineering programs has established a strong link between our introductory computing and upper level disciplinary courses, which is paving the way for a major restructuring of our college's freshman engineering offering. Statistical analyses have shown clear support of the "use it or lose it" theory for both confidence and performance in MATLAB. Results also provide preliminary evidence on the benefits of integration to encourage transfer of understanding from one context to another. Additional implementation and analysis of the results continues, and Spring, 2007 results will be included in the presentation.

## Bibliography

1. "The Vision for Change: A Summary Report of the ABET/NSF/Industry Workshops," ABET, Inc., 1995.
2. Latucca, L.R., P.T. Terenzini, and J. F. Volkwein, (2006) "Engineering Change: A Study of the Impact of EC2000," ABET, Inc.
3. Foundation Coalition. (2001). "Curriculum integration - Students linking ideas across disciplines." <http://www.foundationcoalition.org/home/>
4. [keycomponents /CurriculumIntegration/intro.html](http://www.foundationcoalition.org/home/keycomponents/CurriculumIntegration/intro.html) (2003).
5. *Criteria for Accrediting Engineering Programs*, ABET, Inc., <http://www.abet.org>.
6. Shuman, L. J., C. J. Atman, et al., (2002), "The future of engineering education." *Frontiers in Education*, Boston, MA, IEEE.
7. Froyd, J.E. and M.W. Ohland, (2005) "Integrated Engineering Curricula," *Journal of Engineering Education*, 94(1).
8. Pennington, N., R. Nicolich, and J. Rahm, (1995). Transfer of training between cognitive subskills - is knowledge use specific. *Cognitive Psychology*, 28(2), 175-224.
9. Bransford, J.D., A.L. Brown, and R.R. Cocking, eds., (2000) How People Learn: Brain, Mind, Experience, and School, National Academy Press, Washington, D.C.