121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014
Paper ID #10140

The Impact of Reducing Numerical Methods and Programming Courses on Undergraduate Performance

Dr. Stephen Michael Morse, Texas Tech University
Dr. Audra N. Morse, Texas Tech University

Dr. Audra Morse, P.E., is the Associate Dean for Undergraduate Studies in the Whitacre College of Engineering and a Professor in the Department of Civil and Environmental Engineering at Texas Tech University. She leads the Engineering Opportunities Center which provides retention, placement and academic support services to WCOE students.

Dr. Venkatesh Uddameri
Dr. Elma Annette Hernandez, Texas Tech University
Mr. David L Ernst, Texas Tech University - Construction Engineering

BS - Mechanical Engineering - Cornell University - 1973 ME - Nuclear Engineering - Cornell University - 1974 US Air Force - 8 years Texas Instruments - 1.5 years Texas Tech University - 30 years

The Impact of Reducing Numerical Methods and Programming Courses on Undergraduate Performance

abstract

Due to pressure from state legislatures reducing credit hour limits coupled with a required set of core courses, science and engineering discipline curricula have optimized and downsized the topics covered in degree programs[2]. Many secondary skills such as computer programming, numerical methods, finite elements, and stochastic analysis have been dropped completely from the curricula or minimally covered through the use of industry standard software. While these skills are not specific to a particular discipline the authors opine they constitute fundamental knowledge, similar to calculus, in which all engineers should have competence. This paper explores the effect of phasing out these secondary skills has on students perceived understanding and ability to apply them in upper level engineering courses and graduate courses. Assessment data regarding secondary skill competency from a freshman engineering analysis class and from a graduate numerical methods class is presented and discussed. The authors conclude with a series of strategies they intend to employ with assessments in future course offerings to help students learn these secondary skills without covering them in a formal course.

introduction

Many state legislatures have mandated 120 semester credit hours (SCH)[1,2] for all university undergraduate degrees, theoretically enabling students to complete any degree in four years. This requirement has resulted in schools choosing various methods to reduce civil and environmental degree program requirements from 135-140 SCH to 120-125 SCH averaging approximately 130 SCH[3]. Strategies for maximizing remaining credit hours include: a) requiring entering students to have completed Calculus I and in some cases also Calculus II; b) cutting, removing or combining some general engineering courses (such as combining Statics and Dynamics into one 3 to 4 SCH course); c) treating Physics II and Electrical Circuits as essentially equivalent and requiring only one; d) eliminating or turning Numerical Methods and Finite Difference/Element courses into electives; and e) eliminating numerical methods topics from curricula due to the inclusion of industry standard software (ISS) packages such as MODFLOW (groundwater modeling), ANSYS (for structural analysis) and HEC-HMS (for hydrologic routing)[3,4,5,6].

Due to the curriculum reduction approaches described, the potential impacts on the knowledge and skills students learn and develop during their college academic experience include the students' lack of understanding of general theoretical concepts of physics, a decreased knowledge on fundamental engineering principles, decreased math background and limited programming ability, as well as “soft skills” associated with integrating and managing[3,7]. In fact, Barlish and

Traylor[8] found there was a disconnect between the skills that students obtain from their degree program and the skills (as defined by the industry) necessary to be successful. Therefore, studies have looked at how to address this deficiency through proper class sequencing[9] emphasizing "life long learning" which calls on the individual to self teach[10], and to a larger extent changing teaching methods[4,11].

The aforementioned studies have not specifically addressed the impact of curriculum reduction on programming and numerical analysis skills. Therefore, the objective of this study is to inform the development of engineering curricula with regards to the skills associated with numerical analysis and programming. Students not exposed to numerical methods may find it difficult to comprehend the importance of model parameters and the output of the ISS software. To address these concerns, the authors reviewed the coursework at Texas Tech University to assess the coverage of numerical methods and programming topics in the civil engineering degree program and assessed student mastery of these topics. To mitigate the deficiency of skills and knowledge related to numerical methods and programming, the authors conclude with a series of strategies they intend to employ to improve students' numerical methods and programming skills.

numerical method and programming exposure

Recognizing incoming freshmen students often have minimum exposure to numerical methods and programming during their K12 education, the College of Engineering at Texas Tech University recently revised an existing course to introduce these concepts early across every engineering program. The objective of the course is to provide a basic introduction to engineering problem solving and programming and impart competence in entry level numerical methods including interpolation, regression, numerical integration and solving linear systems of equations. This course is not intended to cover all numerical methods students may need in subsequent courses in their degree plan but rather to provide a basic skill set they can build on throughout their education. Table 1 lists the topics covered in the Introduction to Engineering course. The first section of Table 1 lists the topics related to numerical methods and the later section lists the topics related to programming.

Table 1. Introduction to Engineering Course Topics

	Course Topics
Numerical Methods	Geometry and Trigonometry Review
	Fermi Problems
	Dimensional Analysis
	Unit Conversion
	Imaginary Numbers
	Statistics (Histograms, Normal Distribution, Boxplots)
	Linear Interpolation and Regression
	Numerical Integration
	Vectors
	Matrix Notation and Operations
	Solving Linear Systems of Equations
Programming	Algorithms and Flowcharts
	Variables
	Input and Output
	Arithmetic, Relational and Logical Operators
	Boolean Expressions
	Making Decisions (If Statements)
	Loops
	Functions

MATLAB is introduced early in the course so students may use the software to solve problems in addition to solving problems by hand. While the students are expected to apply numerical methods without programming on exams, employing MATLAB in parallel on homework assignments provides the students with the ability to check their MATLAB solutions against hand calculations as they become more proficient with MATLAB. Additionally, longer, more complex problems may be easily solved in MATLAB providing a quick way for students to explore the concepts more extensively than with hand calculations alone. Bearing in mind this is a freshman level course, many of the topics are introduced with basic methods. For example, only the 1^{st} order (linear) methods for interpolation, regression and numerical integration are covered. While these methods are relatively simple, many students find them challenging as they have not been exposed to these types of methods before this course.

When the programming topics are addressed in the course, the students are proficient with MATLAB allowing them to concentrate on the programming logic without struggling with using MATLAB. Approximately one third of the course is used to introduce basic programming logic to the student using the MATLAB scripting language. The authors opine computer programming logic, provides a small, discrete set of logic concepts that when mastered provide a skill set for reducing any engineering problem down to its base components (inputs): 1) indentifying the core problem to be solved, and 2) creating a solution (output). In short, any engineering problem can be represented by an algorithm and subsequently a computer program. Thus, if one is proficient at computer program logic, the same reductionist skill set can be applied to engineering design. The advantage of computer program logic is there are only few fundamental logic concepts one must master, where as the logic of engineering design is often obscured by the myriad design codes and standards required for a particular project.

To quantify the level of competency the students obtained while enrolled in the course, a series of assessments were selected for the course topics. The assessments were typically exam problems, more than half of which were from the final exam. Table 2 lists data from the past two semesters. This data is from a single course section with 50 students for each semester. As the course curriculum, including homework and exams, is standardized for all sections and the instructors worked closely together throughout the semester, this data is considered representative for all sections of the course. The numerical values shown in Table 2 represent the percent passing of the assessed problem.

Table 2. Mastery of Introduction to Engineering Course Topics

	Course Topics	Spring	Fall
Numerical Methods	Geometry and Trigonometry Review	46.9	84.9
	Fermi Problems	--	74.8
	Unit Conversion	68.0	93.0
	Imaginary Numbers	80.8	64.5
	Statistics (Histograms, Normal Distribution, Boxplots)	76.2	74.0
	Linear Interpolation and Regression	70.0	65.4
	Numerical Integration	--	80.2
	Vectors	64.7	69.4
	Matrix Notation and Operations	75.2	84.3
	Solving Linear Systems of Equations	67.6	60.2
Programming	Algorithms and Flowcharts	66.0	70.4
	Boolean Expressions	84.5	90.7
	Making Decisions (If Statements)	37.3	81.4
	Loops	44.2	42.6

Variations in performance between semesters is expected and likely due to differences in the enrolled student population and differences in the questions selected as assessments between semesters. Despite the variation, a clear overall 70% competency of the course material is shown for the two semesters. The data suggests the students consistently mastered matrix notation and Boolean expression and consistently struggled with algorithms, flowcharts and loops. Data from all sections will be collected in the future, providing a more accurate representation of the entire course enrollment and allow for identification of variations in performance, if any, between course sections and semesters.

The freshman Introduction to Engineering course is the only course in the civil and environmental engineering curricula at Texas Tech University that specifically addresses computer programming and the various numerical methods listed in Table 1. Higher level courses often rely on the use of numerical methods beyond those introduced in the freshman Engineering Analysis course (Table 4). Consequently, the instructors must either cover the numerical method as a course topic or require the students to teach themselves the method on their own. Each approach has negative impacts on the students' progress in the course, most notably the additional time spent to learn the new method rather than concentrating on the course topics. Alternatively, the instructor can employ the use of ISS packages reducing the need for explicit numerical analysis and programming topics to be covered. The replacement of numerical methods and computer program in civil engineering curricula is often justified with the logical argument that people drive cars without knowing the theory behind internal combustion engines. Furthermore, some argue students' time is better spent using ISS, as it allows them to explore significantly more complex problems in a short amount of time leading to a better understanding of underlying principles. Additionally, students with inadequate background, preparation or interest in programming and computational methods can more readily navigate through the civil engineering curricula resulting in improved retention and graduation rates. Student exit interview surveys at Texas Tech University have identified student readiness to use ISS is viewed positively by potential employers and satisfies the ABET student outcome criterion (3k) - an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice[11]. It is desirable to a least have an understanding of what numerical method and programming skills the undergraduate students have when they graduate.

numerical methods and programming skills of BSCE program graduates

The Department of Civil and Environmental Engineering at Texas Tech University offers a graduate Numerical Methods course. As many graduates continue their education by pursuing a Master degree, typically about 50% of the students enrolled in the course have graduated from Texas Tech University. On the first day of class the students complete a survey, where the students rate their knowledge of the course topics. Although this survey was originally designed to inform the instructor on the starting level of skills the students enrolled in the course possess, the survey data can also provide insight to the students perception of their numerical methods and programming skills upon graduation. The survey consisted of 16 topics, listed in Table 3, which the students indicate their level of knowledge for each base on a four point scale. Where, 1 denotes they have never heard of the topic; 2 denotes they have heard of the topic but they have not used it; 3 denotes the students have some idea of the topic but their not to clear about it; and 4 denotes they have a clear understanding of the topic and they can explain it.

Table 3. Programming Survey Questions

#	Survey Topic
1	MATLAB
2	Performing simple calculations using MATLAB
3	Performing calculations with matrices and vectors using MATLAB
4	Using m-Files to perform a multistep process or complex set of calculations in MATLAB
5	Creating programs in MATLAB with command line user prompts and output
6	Creating programs in MATLAB which read and write data to files

Table 4. Concept-based Survey Questions and Mapped Upper Level Dependency

#	Survey Topic	Upper Level Dependency
7	Matrix operations such as multiply, inverse	Groundwater Hydrology
8	Orthogonal vector space	Groundwater Hydrology, Structural Analysis
9	Solution to linear system of equations using Gauss Elimination	Water Systems, Structural Analysis
10	Solution to linear system of equations using banded matrix methods	Water Systems, Structural Analysis
11	Least-squares fitting of a curve to data	Hydrology, Water Systems, Groundwater Hydrology
12	Cubic Spline interpolation	
13	Numerical integration using Gaussian Quadrature	Hydrology, Groundwater Hydrology
14	Determination of Eigenvalues and Eigenvectors	Structural Analysis
15	Finite Difference models	Water Systems, Hydrology, Structural Analysis, Groundwater Hydrology
16	High level programming languages, e.g. C++, VB, Java, FORTRAN	Programming constructs (eg looping, for statements, etc) are used in all courses

The first six questions (Table 3) pertain to the use of MATLAB, which the graduates from Texas Tech University have at least used in the Introduction to Engineering course describe previously. The next 9 questions (Table 4) address numerical method topics, some of which are commonly used in undergraduate civil engineering courses. The last question provides an indication of students that know how to program but are not familiar with MATLAB specifically. Figure 1 shows the average response for each question for the past three years the course was taught. The number of students who completed the survey is shown in parenthesis next to each year in the legend.

The data for each topic across the three years is relatively consistent with a slight trend downward on several topics for the years presented indicating the students' perception of their skills has reduced over the last three years. Furthermore, the data for the first three MATLAB related topics indicate on average the students feel they understand the topics but are a little out of practice. The responses for topics 4 through 6 drop sharply suggesting not as many students have performed these features in MATLAB. The students marked Topic 7 as the topic they were most familiar with indicating they understand Matrix operations very well whereas Topic 12 was marked the lowest by the students indicating they were least familiar with Spline Interpolation. Figure 2 shows the combined average of the three years of survey data collection which follow similar trends as Figure 1. The error bars indicate +/- one standard deviation from the mean. The standard deviation appears to be relatively consistent for all responses with an average magnitude of 0.75.

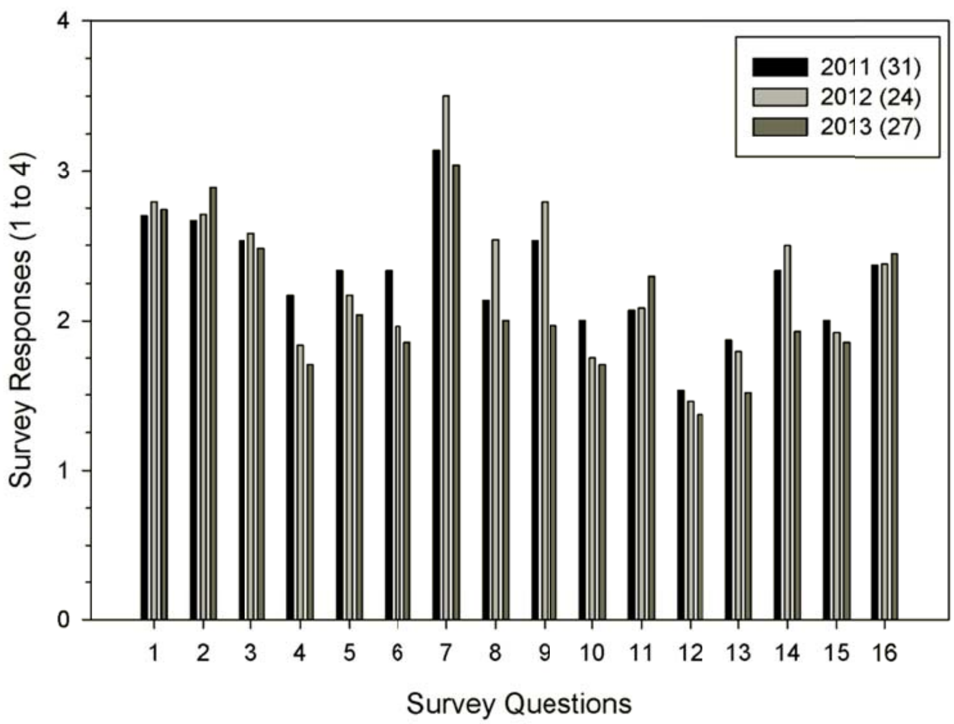

Figure 1. Average Survey Topic Responses for the Past Three Years in the Graduate Numerical Methods Course

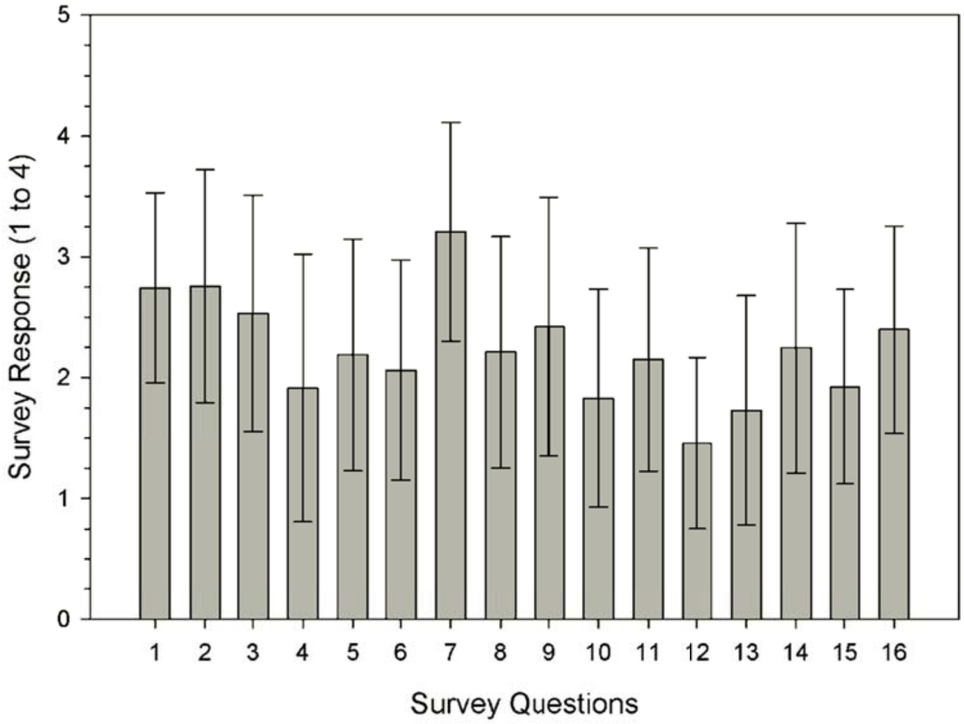

Figure 2. Overall Average of Survey Topic Responses for the Past Three Years in the Graduate Numerical Methods Course

A Mann-Whitney- Wilcoxon test[10] analysis was performed on the survey data over the assessment period. This test is a non-parametric test of the null hypothesis which states that two populations are the same. The results indicate there is, in fact a difference between students' perceived understanding of programming versus their perceived understanding of the fundamental concepts (numerical methods) driving the program (U = 3790, p value = 0.09). Overall the students rating of their familiarity with the course topics was consistent with the assessment data from the Introduction to Engineering course indicating the students have not advanced their programming or numerical methods skill much beyond what they learned in the freshman Introduction to Engineering course. Furthermore, the data suggests instructors are not introducing numerical methods concepts in higher level courses, or they may be relying more on ISS.

plan of action

The authors are currently developing a series of online interactive tutorials covering programming and advanced numerical methods commonly used in upper level courses and graduate programs. The typical format of the interactive tutorials consists of a video-based lesson introducing the concept. The tutorial will then have a series of walkthrough examples where the concept is applied to a problem and solved. The final component of the interactive tutorial will contain auto graded programs where the students can practice applying the concept to problem and get instant feedback when they submit their answers. A few secondary features of the tutorials include bookmark links to previous, subsequent and related topics, progress tracking, and star reward systems similar to many popular video game. The intention of these online interactive tutorials is to make them available to students in the College of Engineering at Texas Tech University, independent of particular courses and the faculty who teach them. In the event a particular method is required for a course, the faculty can simply refer to the topics required and the students can use the system to learn the required topics on their own. In the event the students are deficient in an area, the students can follow the dependent topic links and learn those concepts first. The plan is for topics to be added to the system over time with the goal of having an in-house content rich series of topics for the students to use. Furthermore, extra lessons for traditional courses can be added allowing the students to get lessons outside the scheduled course meeting times. This platform for online interactive tutorials is not intended to completely replace traditional course lesson meetings but rather provide supplemental content. The tutorial system will also track student usage for subsequent assessment of their efficacy in addition to assessments conducted of their use in courses which reference them.

references

1. Chambers, T.L., 2012. "Reduced credit hours and engineering licensure - A proposal to break the impasse." ASEE Annual Conference and Exposition, Conference Proceedings, 2012, 119th ASEE Annual Conference and Exposition.
2. ASCE, 2014. "Raise the Bar: Answers to Frequently Asked Questions", http://www.asce.org/PPLContentWide.aspx?id=2147485470.
3. Russell, J., and Stouffer, W. (2005). "Survey of the National Civil Engineering Curriculum." *Journal of Professional Issues in Engineering Education and Practice*, 131(2), 118-128.
4. Felder, R. M., Woods, D. R., Stice, J. E., and Rugarcia, A. (2000). "The future of engineering education II. Teaching methods that work." *Chemical Engineering Education*, 34(1), 26-39.
5. Stern, F., Xing, T., Muste, M., Yarbrough, D., Rothmayer, A., Rajagopalan, G., Caughey, D., Bhaskaran, R., Smith, S., and Hutchings, B. (2006). "Integration of simulation technology into undergraduate engineering courses and laboratories." *International Journal of Learning Technology*, 2(1), 28-48.
6. Busch-Vishniac, I., Kibler, T., Campbell, P. B., Patterson, E., Guillaume, D., Jarosz, J., Chassapis, C., Emery, A., Ellis, G., Whitworth, H., Metz, S., Brainard, S., and Ray, P. (2011). "Deconstructing Engineering Education Programmes: The DEEP Project to reform the mechanical engineering curriculum." *European Journal of Engineering Education*, 36(3), 269-283.
7. Cheah, C., Chen, P., and Ting, S. (2005). "Globalization Challenges, Legacies, and Civil Engineering Curriculum Reform." *Journal of Professional Issues in Engineering Education and Practice*, 131(2), 105-110.
8. Barlish, K., and Traylor, C. (2014). "Career Paths and Development: Actions and Examples from the Heavy Civil Sector." *Practice Periodical on Structural Design and Construction*, 19(1), 2-7.
9. Christodoulou, S. (2004). ”Educating Civil Engineering Professionals of Tomorrow.” J. Prof. Issues Eng. Educ. Pract., 130(2), 90–94. Yaşar, O., and Landau, R. H. (2003). "Elements of Computational Science and Engineering Education." *SIAM Review*, 45(4), 787-805.
10. Siller, T. J. (2001). "Sustainability and critical thinking in civil engineering curriculum." *Journal of Professional Issues in Engineering Education and Practice*, 127(3), 104-108.
11. ABET, 2014. "Criteria for Accrediting Engineering Programs 2014-2015", http://www.abet.org/eac-criteria-2014-2015/
12. Mann, Henry B.; Whitney, Donald R. (1947). "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other". Annals of Mathematical Statistics 18 (1): 50–60.