ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

New Orleans JAZZED about Engineering Education

Paper ID #14440

Integrated Development of Programming Skills Using MATLAB within an Undergraduate Dynamics Course

Prof. Matthew Rhudy, Pennsylvania State University - Berks
Dr. Rungun Nathan, Pennsylvania State University - Berks

Dr. Rungun Nathan is an associate professor and program coordinator for the mechanical engineering in the division of engineering at Penn State Berks. He got his BS from University of Mysore, DIISc from Indian Institute of Science, MS from Louisiana State University and PhD from Drexel University. He has worked in the area of Electronic Packaging in C-DOT (India) and then as a Scientific Assistant in the Robotics laboratory at Indian Institute of Science, Bangalore, India. He worked as a post-doc at University of Pennsylvania in the area of Haptics and Virtual Reality. His research interests are in the areas of unmanned vehicles particularly flapping flight, mechatronics, robotics, MEMS, virtual reality and haptics, and teaching with technology. He has ongoing research in flapping flight, Frisbee flight dynamics, lift in porous material and brain injury He is an active member of ASEE and ASME and reviewer for several ASME, IEEE and ASEE, FIE conferences and journals.

Integrated Development of Programming Skills using MATLAB within an Undergraduate Dynamics Course

Abstract

This project introduces a method of teaching undergraduate students MATLAB programming within a typical engineering lecture course, dynamics. This particular course was chosen because its content offers natural connections between engineering and programming concepts. The MATLAB programming instruction and assignments were approached through the context of the dynamics course material. This allowed students to make connections between their typical dynamics course content and the more abstract programming concepts which can sometimes be difficult for them to understand. This idea was indicated to be effective through anonymous student comments from student evaluations.

Introduction

Although engineers are not always required to perform programming on a daily basis, it is important for engineering students to be exposed to programming concepts and problem solving techniques. Previously, practicing engineers had to perform coding in a high-level programming language such as FORTRAN or C, but currently many engineers do not need to write their own software[1]. MATLAB, however, is extremely prevalent in engineering industry and research due to its relative ease of use for students when compared to other programming languages[1]. A general trend in programming instruction for engineers has evolved from FORTRAN, to C/C++, and now to MATLAB[3,4]. The connections between MATLAB and the engineering community are plentiful. MATLAB is a useful tool for demonstrating and teaching engineering concepts, e.g. to teach dynamic systems and control[5] or acoustics[6]. On the other hand, engineering concepts tend to be more concrete for students to understand, thus providing an effective means to develop an understanding of more abstract programming concepts.

The methods for teaching MATLAB programming vary greatly among different institutions and instructors. One instructor has considered the use of an automated grading system to streamline grading of MATLAB assignments as well as to provide students with a means to check their programs online[7]. Other digital approaches have been considered in order to provide an automated interactive learning experience for students[8,9]. Some instructors consider the use of hardware such as microcontrollers[10] or robotics[11] in order to produce a more “hands-on” programming experience. Other instructors consider approaches designed to engage student interest through the use of media and technology[12], such as game controllers[13]. MATLAB has also been used as a teaching tool to assist in supplementing teaching and learning, e.g. in engineering mechanics[14].

Some engineering curricula do not have a dedicated programming course to teach students MATLAB. In order to remedy this lack of programming instruction, a different approach was considered in this work which incorporates MATLAB programming instruction into a dynamics course which already exists in many engineering curricula. Teaching MATLAB within a course such as dynamics not only helps to reduce the number of courses that a student needs to take, but also provides some engineering context to the programming techniques. I.e., certain concepts such as decision statements (“if” statements) in programming can be related to a particular application where this would be used, such as determining the direction of frictional forces. This helps students to not only understand how something is used, but also why it is important and where it can be useful. MATLAB has of course been included within engineering courses in the past such as circuits[15] and structural analysis[16], however typically the utility of this inclusion is directed at its usefulness for the specific course content, rather than to develop programming skills in general. This context, whether in dynamics or other subject areas, has a lot of potential for developing students' problem solving skills using MATLAB programming. The purpose of this work is to illustrate this potential for enhancing programming instruction for a specific example of a dynamics course, however these ideas could be applied to other courses as well.

Implementation

This idea was implemented within an undergraduate dynamics course that met 4 hours per week. The structure of this course was approximately 75% (~3 hours per week) dedicated to conventional dynamics lecture instruction, with the remaining 25% (~1 hour per week) dedicated to MATLAB programming within the dynamics context. This course is a requirement for all Mechanical Engineering students, and is typically taken in their sophomore or junior years in the program. Most of these students at this point in their academic career have only had minor exposure to MATLAB within their introduction to engineering course, which is a common approach taken by many institutions[3,4,17]. The goal of the MATLAB portion of the class was to teach students about computer programming using MATLAB as well as to leverage and reinforce engineering concepts from dynamics in the process. Each week typically contained 25 minutes of MATLAB demonstration from the instructor, 25 minutes of in-class time to work on exercises facilitated by the instructor, and one MATLAB programming assignment which was tied into the dynamics course content for the week. A list summarizing the considered connections between the engineering concepts studied in dynamics and various programming concepts are offered in Table 1.

To be more specific about how MATLAB programming was integrated into the dynamics course material, a few example programming assignments are described. Each of these examples were problems that were part of a weekly homework assignment. Each assignment consisted of 6 problems, 1 of which was programming focused. There were a total of 8 homework assignments throughout the semester, and four of the programming problems from these assignments are presented here:

Example 1: This programming assignment teaches students about the use of MATLAB's symbolic toolbox, including functions such as `syms`, `ezplot`, `diff`, and `subs`. The use of symbolic math has been linked to developing students' critical thinking and problem solving skills[18,19]. The problem statement provides the student with a given function for the shape of a hill, over which a car is driving at a given speed and acceleration. The students are asked to determine the normal and frictional forces the car is exerting on the road, as well as to plot the profile of the hill and the radius of curvature, e.g. as shown in Figure 1. This assignment is introduced early in the course to help demonstrate to students the applicability of MATLAB to engineering situations. After performing some relatively tedious radius of curvature calculations by hand, students are offered a modern alternative to pen-and-paper algebra and calculus.

Table 1. Connections between Engineering and Programming

Engineering Concepts	Programming Concepts	MATLAB Functions
Calculation and visualization of particle trajectory	Vectors, vector calculations, plotting	Constructing vectors using :, [], and linspace, plot, xlabel, ylabel, title, legend, grid, etc.
Solving a linear system of equations	Matrices, matrix multiplication, matrix inverse	Constructing matrices using [], "\" operator, inv
Designing the parameters of an engineering system	User defined functions, iterating through a loop	function, for loops
Find a solution to a nonlinear equation	Finding a numerical root using iteration to within a desired tolerance	while loops
Calculation of angular momentum for a few different scenarios	Calculation of vector cross product, user defined functions, selection statements	cross, function, menu, switch-case
Clipping on measurement signals	Decision statements, logical statements, Boolean variables	if statements, use of > and < for Boolean vector indices
Reading/writing a table of moment of inertia values	File input and output	save, load, xlsread, xlswrite, fopen, fprintf, fclose
Interpretation and analysis of experimentally collected measurements	Interpolation and curve fitting	interp1, polyfit, polyval
Solving the equation of motion of a dynamic system	Numerical solution to ordinary differential equations	ode45
Design and analysis of a vibrating system	Many of the above concepts	Many of the above functions

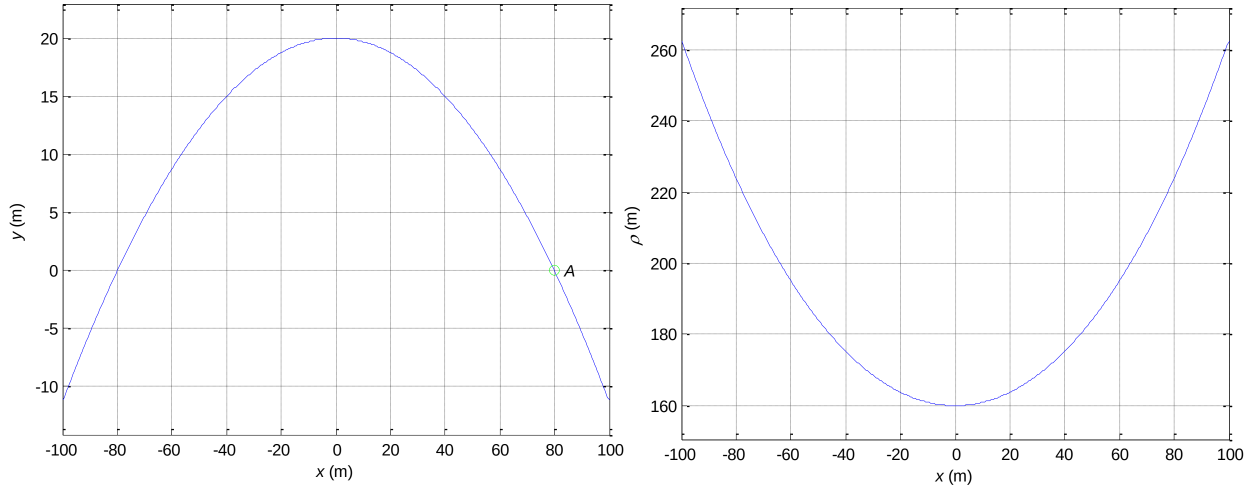

Figure 1. Plots generated by students for assignment example 1

Additionally, students are exposed to plotting and visualization techniques to help break away from many dynamics problems which only consider what is happening at a particular point in space and time. Note that students are still asked to capture information for a specific point by leveraging the `subs` command.

Example 2: The idea behind this programming assignment is to take an existing textbook dynamics problem involving instantaneous calculations for a roller coaster, and transforming it into a more detailed analysis considering various possible design parameters and situations. This assignment leverages the idea of user defined functions in MATLAB, combined with iterating through different possible parameters using `for` loops. Students are then able to not only solve the textbook version of the problem with a simple function call, they can then use the loops to generate vectors of possible solutions. These solutions can then be plotted and analyzed for the various considered situations, e.g. as shown in Figure 2.

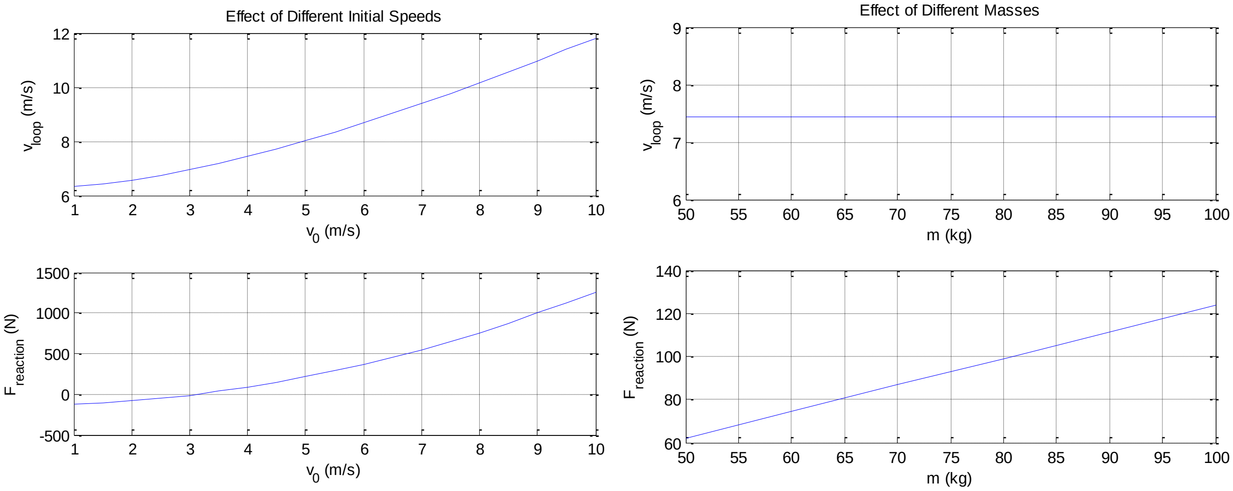

Figure 2. Solutions for various parameters for assignment example 2

Example 3: Later in the semester, the programming assignments build in difficulty to become more complex and integrate multiple programming concepts. This assignment provides students with a data set containing the position of a particle with respect to time in a spreadsheet file. Students are asked to import this data using `xlsread` and then perform some analysis on the data set. In particular, students are tasked with finding the time when the particle passes a certain position. This could be applicable in various situations, such as finding a time for crossing a finish line, or time of impact with an obstacle. Since this position is not contained exactly in the data, students utilize interpolation using `interp1` to more accurately determine the time. Then students are further asked to use interpolation to resample the signal to a different time interval and plot the resulting signal with the original data points as shown in Figure 3. Additionally, students perform curve fitting on the data and provide comparison between interpolation and polynomial fitting using `polyfit` and `polyval`. Note that the considered data set was quadratic with some added noise, however this information is not provided to the students.

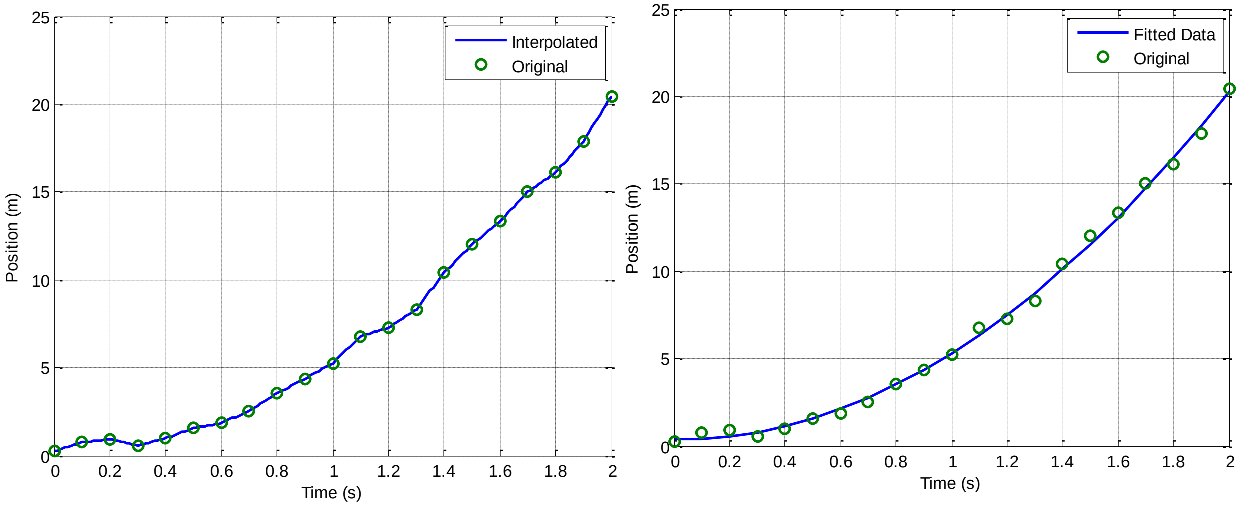

Figure 3. Interpolated and fitted data for assignment example 3

Example 4: This final example is simple but important. Students are given an acceleration equation as a function of time and asked to solve the differential equation using numerical techniques. Students create a `function` and use `ode45` to numerically integrate the equation. Students then plot their solution, e.g. as shown in Figure 4. This numerical solving technique is useful for various future engineering courses for students such as vibrations, instrumentation, and controls.

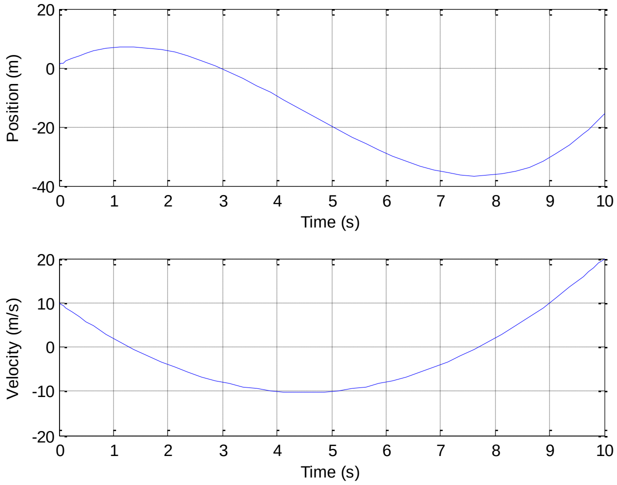

Figure 4. Numerical integration solution for assignment example 4

The culmination of the MATLAB portion of the dynamics course is an individual student programming project. This project was designed to integrate many of the various programming concepts taught throughout the semester into a single dynamics programming problem. The problem description for the project is presented as follows:

Consider the spring mass system shown in Figure 5.

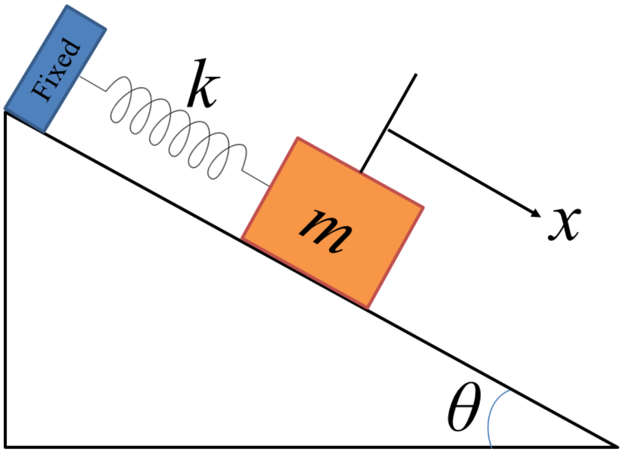

Figure 5. MATLAB Project Diagram

The block has a mass, *m* = 50 kg. The coefficient of friction between the block and the inclined plane is μ_k = 0.3. Four different cases of the angle of the inclined plane, *θ*, must be considered (0º, 30º, 45º, and 60º). For each case of *θ*, determine the smallest value to within 100 N/m of spring constant, *k*, that can be used to keep the maximum displacement, *x*, within 0.2 m of the equilibrium position. The block experiences an initial velocity of 2 m/s down the slope from the equilibrium position. Consider times spanning from 0 to 2.5 seconds.

Some additional guidance is offered to the students to help with the project, such as requirements for formatting, deliverables, etc. The idea behind this project is to have students implement a user defined function for the differential equation and then solve using `ode45`. Within their differential equation function, students need to consider an `if` statement in order to appropriately model the direction of the frictional force in the system. The different scenarios are captured through `for` loops for the different angles, and `while` loops to try different spring constants. Students are evaluated not only on the proper solution to the problem, but also on the quality of their programming practice. Due to the small sample size for the class (12 students) and ease of the grading rubric, the project grades do not carry much meaning to evaluate this work. However, each student was able to successfully complete the project accurately and with good programming practice. An example figure for the solution curves for the project is offered in Figure 6.

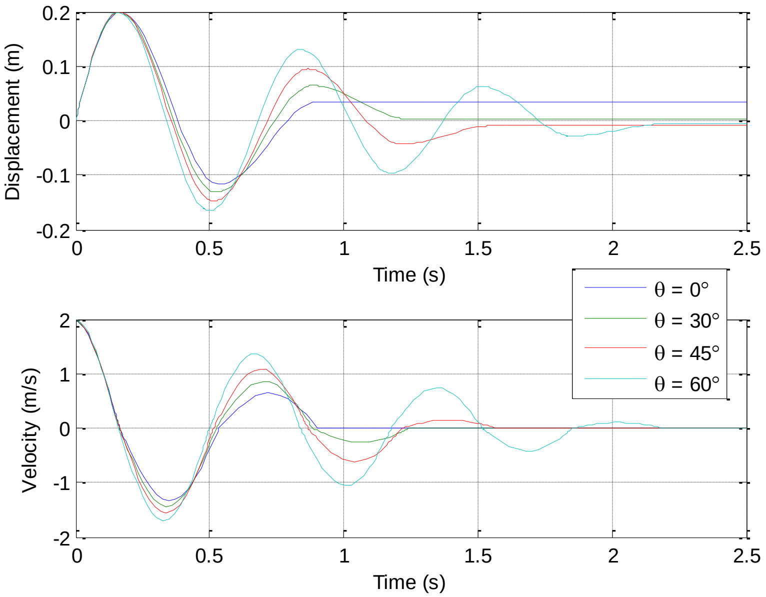

Figure 6. Various Solution Curves for the Final MATLAB Project

Results

No formal assessment was used specifically for this project, however students were encouraged to offer their opinions on this approach anonymously in their student evaluations of teaching. All of the comments pertaining to MATLAB from the student evaluations are listed below:

- MATLAB section was my favorite
- MATLAB sessions were very helpful
- I really enjoyed the MATLAB lectures. I feel more comfortable using it
- The instructor included MATLAB in practical applications which really helped me understand it better
- I learned a lot about MATLAB
- The project really helped us learn MATLAB
- The instructor did a very good job at including MATLAB to make us understand the material
- The instructor put a lot of effort into teaching us MATLAB, which we all greatly appreciate
- Significant exposure to MATLAB
- I also learned how to use basic MATLAB functions
- Learned a lot of MATLAB
- I was a big fan of how the instructor did the MATLAB sections of the course as I learned a great deal from it
- The MATLAB lectures were very helpful to my development of MATLAB skills
- I learned a great deal about MATLAB, which was especially helpful in other courses

These comments indicate that the students overall reflected positively on this experience. It is especially encouraging that students noted the benefit of using "practical applications" and recognized that this programming knowledge was "helpful in other courses."

Conclusions

This paper presented the idea of teaching MATLAB programming to mechanical engineering students through the context of their existing dynamics course. The connections between the dynamics course content and programming concepts were exploited as much as possible in order to provide an integrated learning experience for the students. Overall, this idea seems to offer a reasonable approach for teaching engineers programming skills, and is worth considering further.

Acknowledgments

The author would like to thank the fall 2014 dynamics students for their participation, feedback, and support of this project.

References

[1] Bjedov, G., and Andersen, P. K., "Should Freshman Engineering Students Be Taught a Programming Language?" *IEEE Frontiers in Education Conference*, Vol. 1, 1996, pp. 90-92.

[2] Fangohr, H., "A comparison of C, MATLAB, and Python as teaching languages in engineering," *Computational Science-ICCS*, 2004, pp. 1210-1217.

[3] Azemi, A., and Pauley, L. L., "Teaching the Introductory Computer Programming Course for Engineers Using Matlab," *ASEE/IEEE Frontiers in Education Conference*, Saratoga Springs, NY, 2008.

[4] Azemi, A., and Pauley, L. L., "Using Matlab to Teach the Introdutory Computer-Programming Course for Engineers," *ASEE Annual Conference and Exposition*, 2004.

[5] Rodriguez, Armando, DeHerrera, M. F., and Metzger Jr., R. P., "An interactive MATLAB-based tool for teaching classical systems and controls," *IEEE Frontiers in Education Conference*, Vol. 2, 1996, pp. 624-627.

[6] Campbell, D., Palomaki, K., and Brown, G., "A Matlab simulation of "shoebox" room acoustics for use in research and teaching," *Computing and Information Systems*," Vol. 9, No. 3, 48, 2005, pp. 48-51.

[7] Bowen, J. D., and Price, C. E., "An automated grading system for teaching MATLAB to freshman engineers," *Proceedings of the 2003 American Society for Engineering Education*, 2003.

[8] Daku B., L., F., and Jeffrey K., D., "An Interactive Computer Based Tutorial for MATLAB," *IEEE Frontiers in Education*, Kansas City, MO, Nov. 2000.

[9] Daku B., L., F., and Jeffrey K., D., "Development of an interactive CD-ROM-based tutorial for teaching MATLAB," *IEEE Transactions on Education*, Vol. 44, No. 2, 2001.

[10] Canfield, S., Ghafoor, G., and Abdelrahman, M., “Enhancing the programming experience for first-year engineering students through hands-on integrated computer experiences,” *Journal of STEM Education*, Vol. 13, No. 4, 2012, pp. 43–54.

[11] Behrens, A., Atorf, L., Schwann, R., Neumann, B., Schnitzler, R., Balle, J., Herold, T., Telle, A., Noll, T. G., Hameyer, K., and Aach, T., “MATLAB meets LEGO Mindstorms—A freshman introduction course into practical engineering,” IEEE Transactions on Education, Vol. 53, No. 2, 2010, pp. 306-317.

[12] Guzdial, M., and Soloway, E., “Teaching the Nintendo generation to program,” *Communications of the ACM*, Vol. 45, No. 4, 2002, pp. 17-21.

[13] Brindza, J., Szweda, J., Liao, Q., Jiang, Y., & Striegel, A., “WiiLab: bringing together the Nintendo Wiimote and MATLAB,” *IEEE Frontiers in Education Conference*, 2009, pp. 1-6.

[14] Al-Masoud, N., “Development of MATLAB Graphical User Interface for Teaching Statics,” ASME International Mechanical Engineering Congress and Exposition, Orlando, FL, 2005.

[15] Azemi, A., and Stook, C., “Utilizing MATLAB in Undergraduate Electric Circuits Courses,” *IEEE Frontiers in Education Conference*, Vol. 2, 1996.

[16] Sathyamoorthy, M., “Integrating MATLAB in Mechanics and Structural Analysis Courses,” *ASEE Annual Conference and Exposition*, 2003.

[17] Clough, D. E., Chapra, S. C., & Huvard, G. S., “A change in approach to engineering computing for freshmen–similar directions at three dissimilar institutions,” *ASEE Annual Conference and Exposition*, St. Louis, MO, June 2001, pp. 773-784.

[18] Gao, J., Dave, S., Patton, R., and Duane, J., “Teaching Critical, Creative, and Practical Thinking with MATLAB Symbolic Math,” *ASEE North-Central Section Conference*, 2013.

[19] Cai, L.-W., “Using MATLAB to Enhance Engineering Students’ Analytical Problem-Solving Skills,” ASME International Mechanical Engineering Congress and Exposition, Seattle, WA, 2007.