

# “FEApps”: Boosting Students’ Enthusiasm for Coding and App Designing, With a Deeper Learning Experience in Engineering Fundamentals

PETER L. BISHAY

*Department of Mechanical Engineering, College of Engineering and Computer Science, California State University, Northridge, California*

Received 1 October 2015; accepted 26 January 2016

**ABSTRACT:** In this article, a model for a multi-goal course project is reported and discussed. Teams of students are asked to create attractive Matlab Apps for topics related to “Fundamentals of Engineering” (FE) exam (hence called “FEApps”). The main goals of this project, discussed in details in the article, are: (1) allowing students practice coding in a new innovative environment, (2) encouraging students to think as designers who create user-friendly and robust products or services and their user manuals, (3) giving students the chance to work in teams with more useful interactions among teams, (4) showing students the benefit of writing multiple drafts for their projects and getting many feedback before the final presentations, and (5) refreshing students’ knowledge about topics related to FE exam. The article provides some examples of “FEApps” created by the participating students. This course project guarantees the highest student retention rates based on the well-known “learning pyramid” and focuses on one of the relatively difficult mechanical engineering program outcomes, specified by the ABET, which is outcome i: “a recognition of the need for, and an ability to engage in life-long learning.” The project assessment survey results proved that all project goals had been successfully met. © 2016 Wiley Periodicals, Inc. *Comput Appl Eng Educ*; View this article online at [wileyonlinelibrary.com/journal/cae](http://wileyonlinelibrary.com/journal/cae); DOI 10.1002/cae.21723

**Keywords:** Matlab; graphical user interface (GUI); Apps

## INTRODUCTION

Matlab is a programming environment for algorithm development, data analysis, visualization and numeric computation. The software is developed by MathWorks and was founded in 1984 [1]. Currently, Matlab has more than one million users worldwide, and more than 5000 colleges and universities around the world use Matlab for teaching and research in a broad range of technical disciplines. More than 1500 Matlab based books in 27 languages were published [1].

Majid et al. [2] in 2013 presented a literature review on using Matlab as a teaching and learning tool in Mathematics

summarizing the different ways in which the software has been used for visualization, computation, problem-solving, fostering positive attitudes and math confidence, increasing motivation and math performance, and projects. Matlab has been incorporated in many engineering courses in different colleges and universities. Some institutions even dedicated a course to teach students how to use the software to solve different types of problems (such as linear algebra, differential equations, curve fitting and interpolation, etc.). Some instructors created their own graphical user interfaces (GUIs) and provided them to students as tools closely related to their course materials. For example, GUI for differential equations was used in [3], GUI for computational fluid dynamics (CFD) was created in [4], GUI for control systems was presented in [5], among many others. However, teaching students to create their own GUIs or applications has not been addressed extensively. Lent et al. [6] in 2012 presented an approach to teaching Matlab focusing on student-written

---

Correspondence to: P. L. Bishay ([peter.bishay@csun.edu](mailto:peter.bishay@csun.edu)).

computational models with a graphical user interface (GUI) in a first-year engineering class. They listed the reasons why building the GUI is valuable for student learning as: (1) helps students develop an intuitive insight into the model behavior, (2) allows students to construct more realistic models than the idealized text-book problems, (3) make students more design-oriented since they are building a model that someone else could use it, (4) makes students' first experience with programming more enjoyable and appealing.

Since many undergraduate and graduate courses in ME and AE departments rely on Matlab in solving problems and working on projects, Matlab is taught in universities either as a part of a general course on computer applications with other software such as word processing, spreadsheet, presentation, web-design, and programming software, or as a separate course that focuses on solving engineering problems using Matlab alone or Matlab with other similar software such as Mathematica or Maple. Many textbooks have proposed curricula for a course on Matlab such as [7–10]. These curricula focus on the essential tools for solving different types of engineering problems such as linear algebra, calculus, differential equations, interpolations and curve-fitting, in addition to graphing and programming. Creating graphical user interfaces (GUIs) or Matlab Apps is not commonly offered. Even when GUIs are offered, course projects, when assigned, usually do not require a GUI part.

This article presents an approach to engage students with programming and App designing in an “Engineering Computer Applications” course dedicated for teaching students Matlab as a tool for solving different types of engineering problems. A focus was made on engineering fundamentals covered in other engineering courses such as mechanics of materials, thermodynamics, heat transfer, statics, etc. that form the basic components of the Fundamentals of Engineering (FE) exam. Students were able to create very useful GUIs/Apps as will be illustrated in Section Examples of “FEApps.”

The rest of the article is organized as follows: Section Engineering Computer Applications course and “FEApps” gives an overview on the “FEApps” project, Section Examples of “FEApps” presents some examples of GUIs or “FEApps” created by participating students, Section Project Assessment presents the assessment survey results, Section Discussion and Possible Extensions provide some possible extensions, and finally conclusions are provided in Section Conclusions and Summary.

## ENGINEERING COMPUTER APPLICATIONS COURSE AND “FEAPPS”

Because of the extensive use of Matlab software in many courses in the mechanical engineering curriculum, such as mechatronics, system dynamics, control theory, finite element analysis, heat transfer, among others, a special course named “Engineering Computer Applications” was introduced to the curriculum of the fall semester of the junior year at Saint Martin's University (SMU), Lacey, WA, in order to well prepare junior students to use this programming and problem solving software in these courses. The course requires only vector calculus as a prerequisite. The author used Palm's textbook [7] as the course reference, and scheduled the course as in Table 1.

The author had four goals in mind when he planned for and taught the course: (1) to expose students to most of the important

**Table 1** Schedule of “Engineering Computer Applications” Course at SMU (Fall 2014)

Week	Topic	Notes
1	Overview of Matlab	
2	Matrices, arrays and cells	
3	Functions and files	
4	Programming	
5	Programming	
6	Programming	Intro. to “FEApps”
7	Advanced plotting and graphics	
8	Midterm exam	Proposals submission
9	Graphical user interface (GUI)	
10	Graphical user interface (GUI)	
11	Numerical methods and differential equations	
12	Numerical methods and differential equations	First draft submission
13	MuPAD (symbolic math)	
14	MuPAD (symbolic math)	Second draft submission
15	Simulink	
Final	Final exam	Final submission and presentations of “FEApps”

tools used in solving engineering problems, (2) to introduce students to programming in a new and interesting environment, (3) to encourage students think as designers and become innovative, (4) to support the efforts of the ME department to raise the pass rate in the Fundamentals of Engineering (FE) exam. Accordingly, the author introduced the “FEApps” (or Fundamentals of Engineering Apps) project to the curriculum of the course. Students were introduced to the idea of the “FEApps” project in the sixth week as they learn about programming and were asked to form teams of two or three and propose an idea for an App or program that solves a specific class of problems related to any course from the FE exam reference handbook [11]. Topics include Mathematics, Statics, Dynamics, Mechanics of Materials, Thermodynamics, Heat Transfer, Fluid Mechanics, and Control. Proposals are to be submitted in the eighth week after the midterm exam. Once approved, students were asked to start working on their projects by writing a code that solves the selected problem. In the ninth and tenth weeks, students learn how to create a GUI in Matlab and how to package their GUIs to create Apps. Hence teams are asked to design the layout of their Apps and link them to their codes. First drafts of the Apps should be submitted in the twelfth week, and teams are asked to share their Apps with other teams in order to get feedback beside the feedback they get from the instructor. Second drafts should be submitted in the fourteenth week after considering all points raised in the feedback each team got. The instructor may recommend any “final touches” before the final submission and presentations of the projects/Apps in the last lecture before the final exam. Teams are encouraged to share their Apps with other teams in order to get feedback and refresh their knowledge about topics covered in other Apps, which are essential topics in the FE exam reference handbook. This is considered a deep learning experience because students not only studied the topic, but also created a code and App for solving this class of problems related to this topic, and critically reviewed other Apps and codes written by their colleagues about other topics and problems. Although interaction between teams and the exchange of feedback were encouraged, innovation is emphasized as one of

the most important criteria in grading this project. The schedule mentioned here and in Table 1 can be cast in a more compact way if applied in a quarter academic system, rather than the semester system.

Students are encouraged to consider the following in designing their Apps: (1) attractive program interface, (2) ease of use and user-friendliness, (3) well documentation in a professional user manual, (4) robust and well studied program performance. The fourth point means that if the program is requiring an integer as an input, the program should give the user a guiding message if the user inputs rational, irrational (fraction or decimal), complex number, or any letters. The user should be able to reset the program, and should be able to use a help button to get some guidance about how to use the program. Teams should create their own design layouts and use the different objects (Push Button, Slider, Check box, Edit Text, Static Text, Pop-up Menu, Listbox, Toggle Button, Tables, Axes, Panel, Button Group [see Fig. 1]) as they see fit for their selected applications. Students were asked to imagine that their Apps will be commercially sold and accordingly the most attractive and robust Apps will be more successful and rewarding in the software market. Teams are encouraged to think about who may be interested in this App, and how can the App design be useful, attractive, easy and professional. These Apps will be useful for students preparing for taking their FE exam, as well as engineers and designers in several engineering industries.

The 15 apps created by the 15 participating teams are listed in Table 2 with the corresponding topics from FE exam reference handbook [8]. One team requested to create a more advanced industry-related App from the students' experience working in different companies as part time employees. This request was approved by the instructor.

Out of 15 Apps created, 4 teams selected Mechanics of Materials topics, 3 selected thermodynamics, 2 selected Electricity, and 2 selected Fluid Mechanics topics as can be seen in Table 2. The other four topics selected are: Statics, Dynamics, Probability and Statistics, and Control. The selection is clearly

**Table 2** Apps Created in the "FEApps" Course Project at SMU in Fall 2014 With the Corresponding Topics From the FE Exam Reference Handbook[8]

App	Topic
Mohr's circle App	Mechanics of materials
Torsion analysis of shafts	Mechanics of materials
Shear force and bending moment diagrams	Mechanics of materials
Centroid and moment of inertia calculator	Mechanics of materials
Basic chemistry equation calculator	Thermodynamics
Superheated water steam tables calculator	Thermodynamics
Saturated water calculator	Thermodynamics
Resistor circuit analyzer	Electricity and magnetism
RLC circuit analyzer	Electricity and magnetism
Water pump power calculator	Fluid mechanics
Ductulator*	Fluid mechanics
Analyzing trusses using joints method	Statics
Dynamic system of particles analysis	Dynamics
Statistical data analyzer	Probability and statistics
Feedback control system	Controls

\*The only topic not related to FE exam.

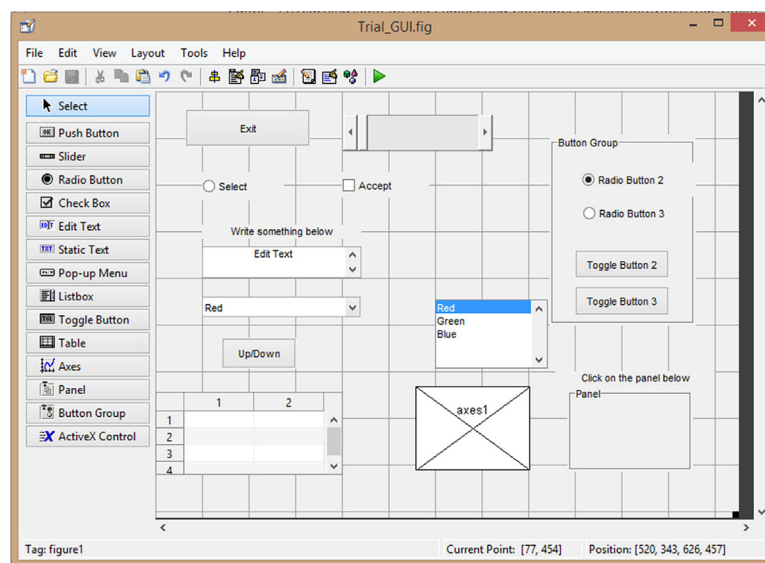
influenced by the topics/courses the students finished by the time they started working on their Apps.

In the next section, some examples of FEApps created by SMU students in "ME 305: Engineering Computer Applications" course during fall 2014 are presented. Acknowledgment of the students who created these Apps under the author's guidance is included in the Acknowledgment section.

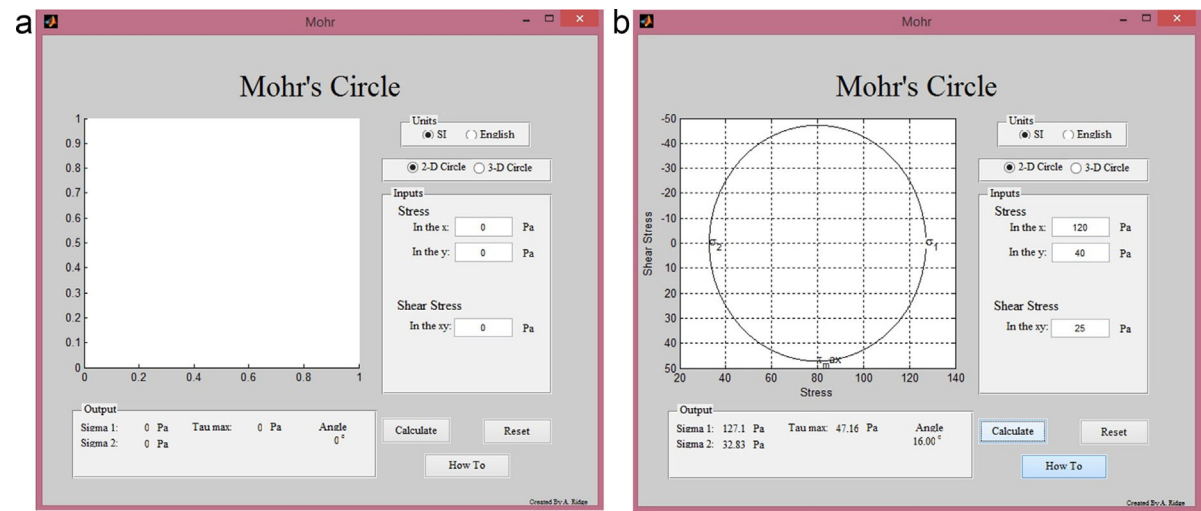
## EXAMPLES OF "FEAPPS"

### Mohr's Circle App

The layout of the App is shown in Figure 2 (left) where the user is expected to select the units he/she is using (SI or English), and to

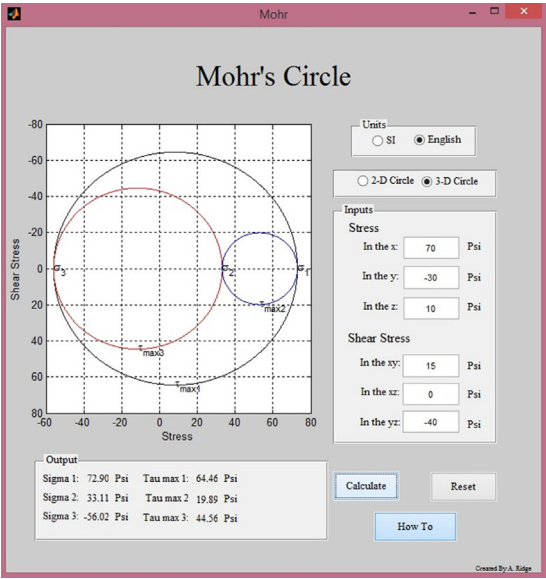


**Figure 1** Available objects in creating a Matlab GUI or app. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]



**Figure 2** Mohr's circle app layout (left), Example of creating a 2D Mohr's circle in SI units (right). [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

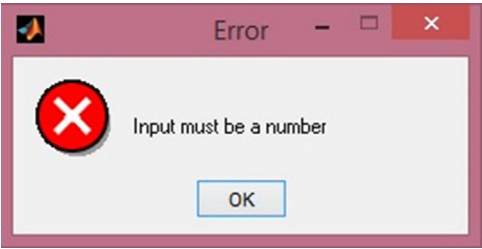
select whether he/she is dealing with a 2D or 3D case: for a 2D case, the user should specify the values of the stresses  $\sigma_x$ ,  $\sigma_y$ , and  $\tau_{xy}$ , while for a 3D case, the user should specify the values of  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$ ,  $\tau_{xy}$ ,  $\tau_{xz}$ , and  $\tau_{yz}$  as shown in Figure 2 (right) and Figure 3, respectively. Once the user clicks "Calculate," Mohr's circles are plotted in the graphics window and the values of the principal stresses appear in the panel below the plot as shown in the figures. The user can reset the App, by clicking on the "Reset" button, and can get more information about how to use the App and the equations used to construct Mohr's circle by clicking on "How To" button. Finally, a pop-up error message appears if the user's input is non-numeric (see Fig. 4).



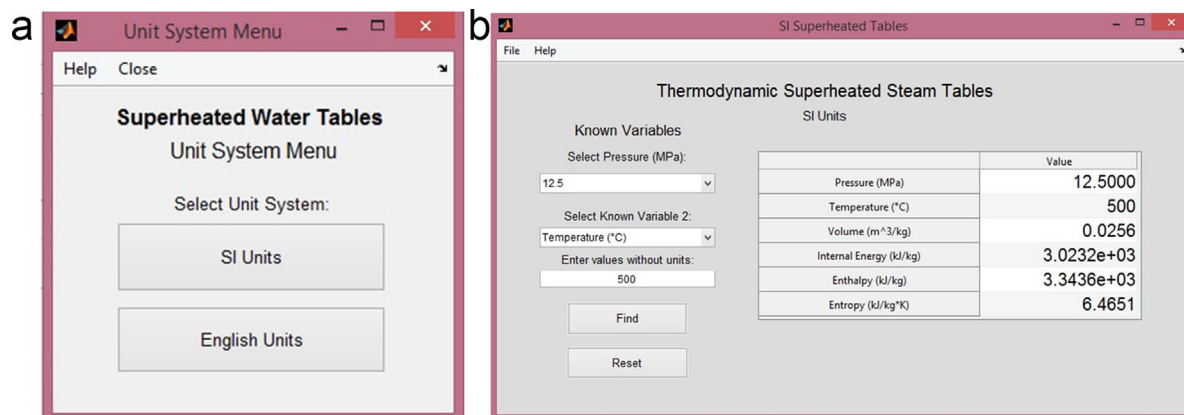
**Figure 3** Example of creating a 3D Mohr's circle in English units. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

### Thermodynamic Superheated Water Steam Tables App

Launching this App displays the Unit System Menu (see Fig. 5 [left]) so that the user can select the unit system (SI or English). The user can also click "Close" or "Help" to open a PDF file that describes how the App works and all equations used in coding. Once the user selects a unit system, the Superheated Water Steam Table calculator appears where the user can select a pressure level (ranges from 0.01 to 60 MPa in SI units), and also selects one other variable (Temperature, Volume, Internal Energy, Enthalpy, or Entropy) as shown in Figure 6 (left), and writes a value for this variable as can be seen in Figure 5 (right). When the user clicks "Find," the properties of the superheated water appears in the right panel. Three error messages, similar to the one in Figure 4 may appear: "Input must be a number" if the user tries a non-numeric input, "Input is below table range" and "Input is above table range" if the user tries values out of the table's range. The user can click "Reset" any time to reset the App, "Help" to open the help file, or "File" which provides three options (see Fig. 6 [right]): "Close", "Return to Unit System Menu", and "Reference Tables" which opens a PDF file for the reference superheated water steam tables used.



**Figure 4** Error message the user gets if any input is non-numeric. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]



**Figure 5** Superheated Water Steam Tables App: Unit system menu (left), SI Superheated Water Steam Tables layout (right). [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

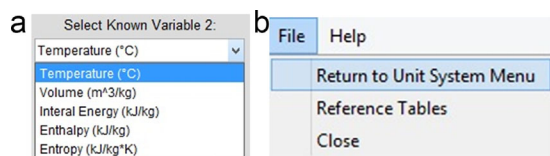
### Other FEApps

In this section we just present some figures that show other FEApps created by SMU students in ME 305 course, fall 2014. It is easy for the readers to understand the function and behavior of these Apps just by looking on these figures. Area and Mass Moment of Inertia Calculator App is presented in Figure 7, Resistor Circuit Analyzer App in Figure 8, Basic Chemistry Equation Calculator App in Figure 9 (left), and Shaft Analysis App in Figure 9 (right), and finally Saturated Water Calculator App in Figure 10.

### PROJECT ASSESSMENT

A survey was done to assess the project goals. All participating teams were asked to rate their benefit in the items listed below (A–M) using a scale of 1–5 (where 1 means “very low benefit,” 2 means “slight benefit,” 3 means “average benefit,” 4 means “very good benefit,” and 5 means “excellent benefit”). Table 3 shows the statistics of the survey findings.

- A. Translating an engineering problem into a code or an App.
- B. General Matlab Coding.
- C. GUI/App designing.
- D. Thinking as designer by creating a “user friendly” and a “robust” App.
- E. Thinking as designer by writing a user-manual for your App.
- F. Working in teams.
- G. Writing multiple drafts to improve your design.
- H. Getting feedback from others (instructor and colleagues) to improve your design.



**Figure 6** (left) selection of second known variable, (right) File menu in the Superheated Water App. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

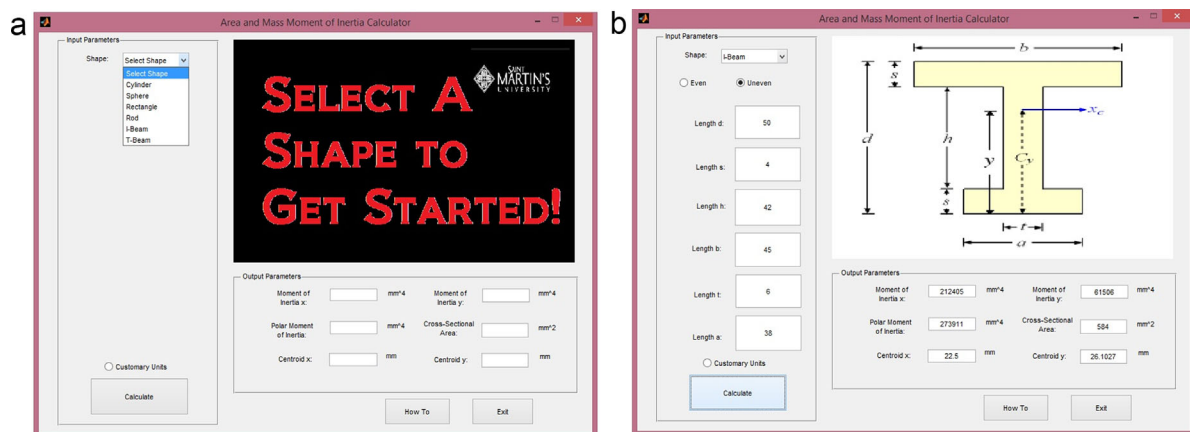
- I. Reviewing previous fundamental engineering problems (Mechanics of materials, thermodynamics, etc.).
- J. Among the goals of this project was to increase “the ability of the students to engage in life-long learning” by showing students how they can convert a real world problem into a matlab code, and learn on their own the required commands to build the code and the GUI. Rate your experience in this aspect.
- K. Among the goals of this project was “to motivate students to become innovative by selecting their own projects, designing their GUI layouts, and designing the behavior of their Apps.” Rate your experience in this aspect.
- L. Among the goals of this project was “to make students’ first experience with programming more enjoyable and appealing.” How enjoyable was the project?
- M. In general, rate your overall experience in this project.
- N. Please write any additional comments (if any) about your experience with this “FEApps” project specifically or ME305 course generally.

Sample student responses to survey question “N” are presented in Appendix A.

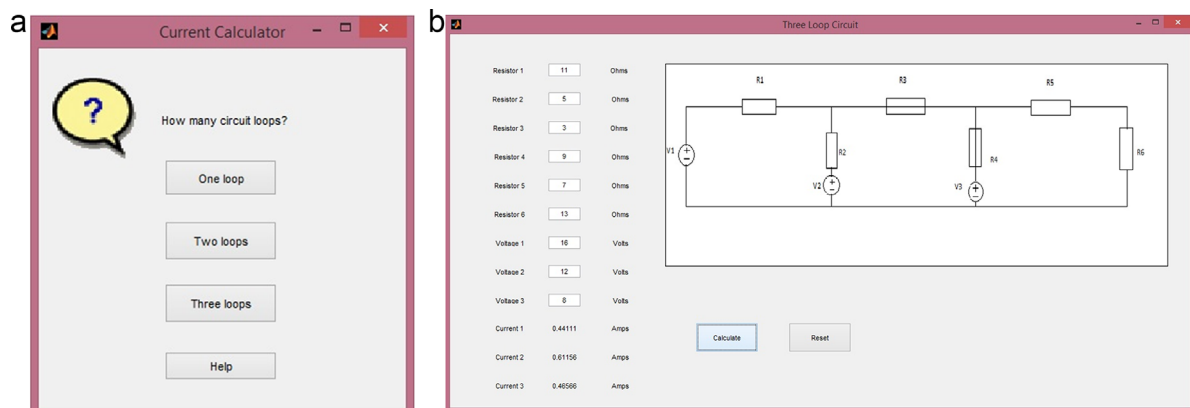
### DISCUSSION AND POSSIBLE EXTENSIONS

Some ideas can be incorporated in “FEApps” project model or any similar projects, such as:

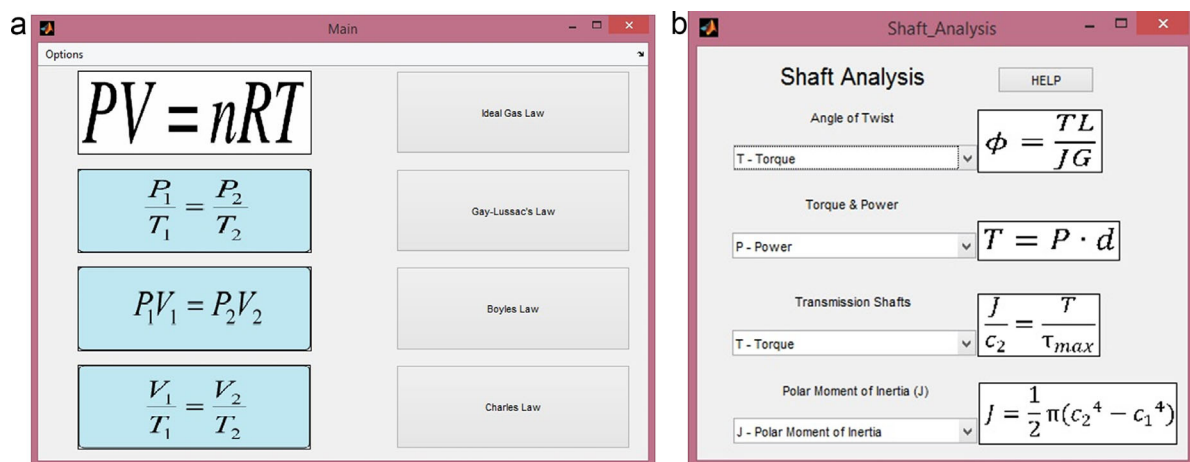
- (1) A website can be created for the project and all Apps can be posted with the names of the students who created them. A students’ club for “FEApps and Matlab programming” can be formed and students are encouraged to submit more Apps they create after finishing the course. The submitted Apps could be related to their courses or any other industrial or engineering applications. The website will list the Apps based on the discipline (Mechanics of Materials, Thermodynamics, Fluid Mechanics, Mathematics, etc.) and also provides a list of Apps related to FE exam and Apps related to other engineering or industrial applications. This engages students in life-long learning process as one of the desired Mechanical Engineering program outcomes based on ABET accreditation (outcome i [12]).



**Figure 7** Area and Mass Moment of Inertia Calculator App. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

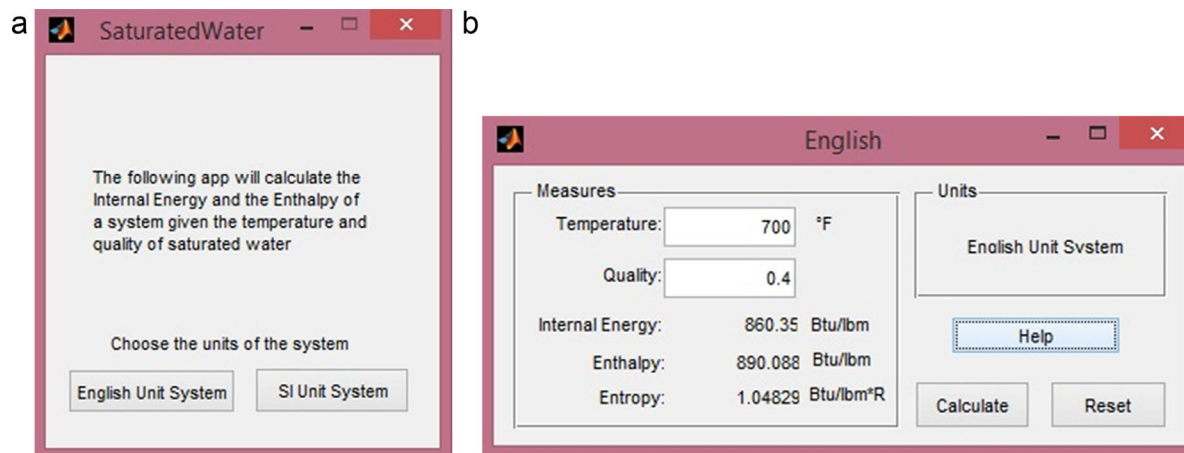


**Figure 8** Resistor Circuit Analyzer App. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]



**Figure 9** Basic Chemistry Equation Calculator App (left), Shaft Analysis App (right). [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]





**Figure 10** Saturated Water Calculator App. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

- (2) Teams that created the best three Apps could receive certificates signed by the instructor, the chair of the department and the dean of engineering. The Apps can be evaluated by two or three invited faculty members who give scores to each App out of 10 based on a given rubric, and teams with the highest cumulative scores will be the winners. The rubric will consider: (a) creativeness and innovation in the Apps, (b) difficulty level of the problem being coded, (c) user friendliness, (d) robust performance, (e) quality of the documentation and help files.
- (3) Participating students have the freedom to share their Apps with Matlab community by uploading them to Matlab File Exchange website.
- (4) Instructors adopting this model may also require each team to review at least three other Apps created by other teams and provide feedback about the important aspects in each App in some prepared forms. This will guarantee better interaction among teams.

## CONCLUSIONS AND SUMMARY

The well-known “learning pyramid” [13] clarifies that the average student retention rate increases in the following order: listening to a lecture, reading, audiovisual, demonstration, then discussion,

**Table 3** Statistics of the Students’ Responses to the Assessment Survey Questions (Number of Responding Teams: 15)

Item	Mean	Median	Mode	Max	Min	Variance
A	4.75	5	5	5	4	0.2
B	4.9375	5	5	5	4	0.0625
C	4.6875	5	5	5	3	0.3625
D	4.75	5	5	5	3	0.3333
E	4.5	5	5	5	2	0.6667
F	4.625	5	5	5	2	0.7833
G	4.75	5	5	5	3	0.3333
H	4.75	5	5	5	2	0.6
I	4.625	5	5	5	3	0.5167
J	4.75	5	5	5	3	0.3333
K	4.875	5	5	5	4	0.1167
L	4.5625	5	5	5	3	0.5292
M	4.75	5	5	5	4	0.2

followed by practicing, and finally teaching others. The presented “FEApps” project model is allowing students to select specific class of engineering problems related to one of many topics in the FE exam reference handbook, write a Matlab code and design a Matlab GUI that will facilitate for the users (students or engineers) solving problems or designing systems. In this activity, students practice coding and designing, as well as refresh their knowledge about materials covered in previous courses. Teams of students are also required to write user manuals for their Apps and give final presentations about their Apps. In this activity, students teach others. Accordingly the learning experience reaches the bottom of the learning pyramid with the highest possible retention rates guaranteed. Students also practice working in teams, submitting multiple drafts for their work before the final submission and presentation, get feedback not only from the course’s instructor but also from their colleagues, review other projects and provide their feedback, think as designers and show their innovations. Hence it is believed that such a model provides a deep learning experience with inspiring environment and life-long engagement with learning. The article presented the project with its goals and constraints, and provided some “FEApps” examples created by the participating students. The results of the assessment survey prove the effectiveness of proposed model. The model can be implemented in any engineering programming course (if the used software provides App-creation capability as Matlab does), or any engineering courses that follow an Engineering Computer Applications or Matlab course, where instructors ask teams of students to create Matlab Apps for specific classes of problems in the curriculum.

## ACKNOWLEDGMENTS

The generous supports of Saint Martin’s University and California State University, Northridge to the author are thankfully acknowledged. Students who participated in this project are also acknowledged for their creative ideas and sincere efforts (Teams created the Apps mentioned in this article are as follows: “Mohr’s circle App” was done by Andrew Ridge; “Superheated Water Steam Table App” was done by Matthew Dunford, Max Flukey and Kevin Krueger; “Area and Mass Moment of Inertia Calculator App” by Conor McDaid-O’Neill and James Malloy;

“Resistor Circuit Analyzer App” by Ian Nisbet, Jacob Ambrose and Hiroshi Kanno; “Basic Chemistry equation calculator App” by Chris Sparks and Carl Svanevik; “Shaft Analysis App” by Ethan Nelson and Jack Freeman; “Saturated Water Calculator App” by Seog Choi and Shengran Tian).

#### APPENDIX A: SAMPLES FROM STUDENT'S ADDITIONAL COMMENTS (SURVEY QUESTION N)

##### Mohr's circle App

“This project and course were very helpful in teaching students how to interpret problems that we face in different classes and translate them into Matlab codes to be solved. This became a very useful to verify the answers to the more complex problems in other classes.”

##### Truss Analysis using Joints method App

“Fantastic project. Basically every detail was accounted for, non-MATLAB engineering principles were reviewed (physics for some, statics for others, etc.) to demonstrate how useful MATLAB can be for engineers, and instructor feedback was extremely useful. The project took a problem (necessity for some sort of calculator rather than working by hand for repetitive calculations) and addressed it from start to finish, by determining the necessary formula(s), coding them into MATLAB, developing a GUI, writing a user manual, etc. It was about as perfect and helpful toward learning as a school project could possibly be.”

##### Ductulator App

“This project met the intended goals in terms of my learning and experience with Matlab. My app dealt with a problem that was not learned in conventional engineering fundamentals. However, my app dealt with something that is used every day in my workplace so it was very applicable to my industry. The focus of this project

was spot on. The key components that made this project successful were the real life application of the end product to fundamental engineering and the guidance by the instructor. Without these two aspects, this project would not have been so helpful to me in my programming.”

#### REFERENCES

- [1] [www.mathworks.com/company/aboutus/](http://www.mathworks.com/company/aboutus/)
- [2] M. A. Majid, Z. A. Huneiti, W. Balachandran, and Y. Balarabe, Matlab as a teaching and learning tool for mathematics: A literature review, *Int J Arts Sci* 6 (2013), 23–44.
- [3] F. Carneiro, C. P. Leão, and S. F. C. F. Teixeira, Teaching differential equations in different environments: A first approach, *Comput Appl Eng Educ* 18 (2010), 555–562.
- [4] C. Depcik and D. N. Assanis, Graphical user interfaces in an engineering educational environment, *Comput Appl Eng Educ* 13 (2005), 48–59.
- [5] A. Andreatos and A. Zagorianos, MATLAB GUI application for teaching control systems. 6th WSEAS International Conference on Engineering Education 2009, pp. 208–211.
- [6] C. S. Lent, J. Brockman, V. Goodrich, and K. Meyers, Teaching MATLAB in first-year engineering: A GUI tool directed approach. 4th First Year Engineering Experience (FYEE) Conference, August 9–10, 2012, Pittsburgh, PA.
- [7] W. J. Palm, III, *Introduction to Matlab for engineers*, 3rd ed., McGraw Hill, New York, NY, 2011.
- [8] S. Attaway, *Matlab, a practical introduction to programming and problem solving*, 3rd ed., Butterworth-Heinemann, Waltham, MA, 2013.
- [9] B. Hahn and D. Valentine, *Essential MATLAB for engineers and scientists*, 5th ed., Academic Press, Waltham, MA, 2013.
- [10] A. Gilat, *MATLAB: An introduction with applications*, 5th ed., Wiley, Hoboken, NJ, 2014.
- [11] NCEES (2013). FE reference handbook, 9.2.
- [12] Accreditation board for engineering and technology (ABET) mechanical engineering program outcomes. [www.abet.org](http://www.abet.org)
- [13] National training laboratories (NTL) for applied behavioral science. [www.ntl.org](http://www.ntl.org)

#### BIOGRAPHY



**Dr. Peter L. Bishay** is an assistant professor of Mechanical Engineering at California State University, Northridge. He earned his PhD from the Mechanical and Aerospace Engineering Department at University of California, Irvine (UCI). His doctoral dissertation was focused on formulating novel numerical methods for accurate and efficient micromechanical modeling of multifunctional materials and composites.

Dr. Bishay's major areas of research are Computational Solid Mechanics, Smart Materials and Structures, Modeling and Simulation, Aerospace Structures, as well as Engineering Education. He taught several graduate and undergraduate mechanical and aerospace engineering courses at University of California, Irvine (UCI), Rutgers, the state University of New Jersey, Cairo University Egypt, as well as Saint Martin's University.