The Role of MATLAB Fundamentals, Debugging, and The Variable Editor in Teaching Engineering Problem Solving

Kun Gao, Jason Conklin, David Siegal, John Slone, and JosAnn Duane
Engineering Education Innovation Center
The Ohio State University, Columbus, Ohio 43210
Email: duane.1@osu.edu

Background and Objective:

This poster paper[1]summarizes the methods developed for student success in learning MATLAB programming fundamentals. It explains the role that programming fundamentals play as part of a broader learning experience in engineering problem solving. A related paper[2], describes the content and development of this course on engineering problem solving, Engineering 1221, that is taught at The Ohio State University.

Students enter this class with some previous MATLAB programming experience, however, that experience may not be current, or completely aligned the expectations of this class. It is our goal to evaluate students' understanding of these fundamental topics and bring the class as a whole to a level of MATLAB competency commensurate with study of intermediate and advanced topics.

This poster paper summarizes some basic knowledge and concept for the MATLAB learning. It emphasizes the following contents: Introduction of the command window and editor; array building; flowcharting; *for-loops*; *if statements*; and, debugging. The objective for this part of the Engineering 1221learning experience offers students an entry into process of mastering MATLAB as an engineering problem solving tool. It teaches students elementary programming concepts that will help them continue their study of MATLAB in upper level engineering courses.

Method:

In this part of course design, the iterative, incremental design method[3]is widely used. Starting with an initial design, the interactive design method repeats the following cycle until the desired level of performance is achieved: Observe, test, and evaluate the design; define the problems; develop the solutions; and, implement the solutions. Midway through the first offering, we made a detailed evaluation of student perception of course content and delivery methods[2]. We are now reworking parts of the course content based on student feedback.

Throughout all the procedures that are mentioned above, the class contents for MATLAB fundamentals, debugging and the MATLAB Variable Editor are essential to student success in learning. We are in the process of reevaluating all the student labs to ensure that sufficient background material is either part of the lab, or references to that background material is part of the pre-class assignment.

We have upgraded the course content on MATLAB self-help features for programming and debugging MATLAB software. These features include:

1. The MATLAB Central community of users[4] which give MATLAB programmers access to other programmers in the MATLAB professional network;
2. The MATLAB library of video tutorials[5] which contains a collection of video tutorials on "how to use" various MATLAB products;
3. MATLAB Product Help[6] covers a larger range of MATLAB product with "how to use" text instructions;
4. The MATLAB function browser[7] lists all the MATLAB built-in functions with "how to use" instructions and,
5. The MATLAB debugger[8] has the capability to analyze MATLAB script files for multiple types of errors.

Our intention is to provide self-help resources to students entering the class with deficiencies in some aspects of MATLAB and to give students who elect to take on the "technical challenge" assignments the resources to complete these assignments.

Results:

The figures on this poster illustrate the learning objectives for this fundamental phase of study. Each figure is framed in the color associated with its primary learning objective as follows:

1. THINK (coral)**:** *Demonstrate ability in critical, creative and practical thinking through algorithm design, MATLAB software design and evaluation.* **For example**: Students learn logic and functioning of the following: fundamental array operations; *if* and *case* selection structures; *for* and *while* loops; application of user-defined functions; user interface communications, and, error trapping methods.

2. USE TOOLS (purple)**:** *Utilize MATLAB software tools to solve engineering problems.* **For example**: Using the MATLAB help, debugger, workspace, and command history to assist software design, troubleshooting, and debugging.

3. COMMUNICATE (orange)**:** *Demonstrate skill in technical communication related to engineering and software development.* **For example**: Code documentation form and function; Visualization methods including, charts, graph and diagrams applied to communicating results; Interactive analysis using MATLAB GUIDE.

These learning objectives are explained in more detail by the poster paper on goal directed course design[2].

The figures are grouped into the following three categories that depict three of the major content modules of the course. Figure 1 and 2 shows the editor window and command window respectively. Figure 3 offers an example of how to create an array. Figure 4 shows how to address the element of an array.

Figures 5 through 10 depict methods of array processing. Figure5 displays the Variable Editor showing the contents of the array. In figure 6, the method of formatting output through *fprintf*[9] is given. The *for-loop* and *if statement* are illustrated in figure 7 and figure 8. Figure 9 displays the flow charts for a *for-loop* and an *if statement*, which give student a clearer vision of repetition and selection in MATLAB programming. Figure 10 is an example of using 'input' command to read data from the user's data entry.

Figures 5 and 11 show two important tools for understanding processing and information flow in MATLAB programs. Figure 11 specifies the procedures of debugging error in MATLAB. This procedure covers errors at multiple levels including syntax errors and a number of runtime errors. In addition the MATLAB debugger gives suggestions for correcting errors and links to additional help on topics related to the errors.

Conclusions:

This poster paper describes methods for teaching fundamental MATLAB knowledge. This part of the course teaches basic skills such as creating and utilizing arrays, managing program flow with loops and selection structures, and debugging MATLAB programs. These skills are aligned with the following three fundamental learning objectives: *Demonstrate ability in critical, creative and practical thinking through algorithm design, MATLAB software design and evaluation; Utilize MATLAB software tools to solve engineering problems; and, Demonstrate skill in technical communication related to engineering and software development*. The skills learned in meeting these learning objectives lay the foundation for performing labs and completing the course term project.

We plan to continue to refine the procedures for incorporating material and learning activities that help students achieve the above learning objectives. We are now in the process of reworking students labs accordingly. We will be making a second solicitation and review of student feedback related to fundamentals. We plan a follow-on paper reporting the results of the second iteration of Engineering 1221 course design.

References:

1. Duane, JosAnn, Liu, Ye and Maynell, Laurie. Value Creation in Engineering Education through Goal Directed Course Design. EEIC ASEE North Central Section Conference, April 5 and 6, 2013, The Ohio State University, Columbus, OH. [Online] April 5, 2013. [Cited: April 5, 2013.]http://eeic.osu.edu/courses-services/asee-papers.

2. Gao, Kun, et al. MATLAB Fundamentals, Debugging, and the MATLAB Variable Editor. EEIC ASEE North Central Section Conference, April 5 and 6, 2013, The Ohio State University, Columbus, OH. [Online] April 5, 2013. [Cited: April 5, 2013.] http://eeic.osu.edu/courses-services/asee-papers.

3. Larman, Craig and Basil, Victor. Iterative and Incremental Development: A Brief History. Computer. [Online] June 2003. [Cited: February 1, 2013.]https://www.it.uu.se/edu/course/homepage/acsd/vt08/SE1.pdf.

4. MATLAB. Blogs.MATLAB Central. [Online] [Cited: February 16, 2013.]http://www.mathworks.com/matlabcentral/.

5. MATLAB. Videos and Examples.Products & Services. [Online] [Cited: February 16, 2013.]http://www.mathworks.com/products/matlab/examples.html.

6. MATLAB. Documentation Center.Products & Services. [Online] [Cited: February 16, 2013.]http://www.mathworks.com/help/matlab/ref/demo.html.

7. MATLAB. Find Functions Using the Function Browser. Documentation Center. [Online] [Cited: February 16, 2013.]http://www.mathworks.com/help/matlab/matlab_env/find-functions-using-the-function-browser.html.

8. MATLAB. Debugging Process and Features.Documentation Center. [Online] [Cited: February 16, 2013.] http://www.mathworks.com/help/matlab/matlab_prog/debugging-process-and-features.html.

9. Recktenwald, Gerald. "Using fprintf in MATLAB".People. [Online] October 13, 2008. [Cited: January 27, 2013.]http://people.bath.ac.uk/jchb20/xx10190/demofprintf.pdf.