Re-Designing the Primers on MATLAB Marina

Thomas Murphy[1], Christopher Williams[2], Priya T. Goeser[3]
[1]Associate Professor, Department of Electrical and Computer Engineering, Georgia Southern University – Armstrong Campus, Savannah GA, [2]Lecturer, Department of Computer Science, Georgia Southern University – Armstrong Campus, Savannah GA, [3]Professor, Department of Mechanical Engineering, Georgia Southern University – Armstrong Campus, Savannah GA

Abstract

MATLAB Marina is a Virtual Learning Environment (VLE) designed to be used by students to learn programming using MATLAB. The VLE serves as the primary resource in a freshmen programming course: Computing Applications in Mechanical Engineering at Georgia Southern University. The VLE consists of modules that address the topics for learning MATLAB programming. Prior to fall 2020, each module had a primer in the form of a static pdf document, a set of video tutorials for most topics, and a set of suggested exercises in the form of another static pdf document. This paper presents the re-design of the primers into shorter documents. Each document now addresses a smaller number of key concepts and is designed to be read in under 10 minutes.

Keywords

virtual learning environment, open educational resource, online textbook

Introduction

MATLAB Marina[1] is a virtual learning environment (VLE) initially developed by faculty at Armstrong State University (now Georgia Southern University). The VLE is a free resource available to anyone and was designed to be used by students to learn programming using MATLAB. MATLAB Marina consists of modules separated into five categories: Introduction (four modules), Fundamental Programming Concepts (seven modules), Advanced Programming Topics (eight modules), Applications (five modules), and Numerical Methods (six modules). The modules consisted of a primer and exercises in the form of static pdf documents and a set of video tutorials housed on YouTube. Previous work[2-5] describes MATLAB Marina's development, use by students, and assessment.

MATLAB Marina is currently the primary resource in one section of a freshmen programming course, Computing Applications in Mechanical Engineering, at Georgia Southern University. Starting in Spring 2021, it will be the primary resource in all sections of this course. It is also used as a secondary resource for other Georgia Southern Engineering courses. The materials in MATLAB Marina were primarily developed prior to 2013 and many had not been updated since fall 2013. MATLAB, the Integrated Development Environment (IDE) and the programming language used in this course, has two scheduled updates per year. Generally, the MATLAB updates are minor, but there have been some significant changes since most the of the primers were created. Two areas significantly affected are Microsoft Excel file input/output and strings.

The MATLAB Marina primers serve a role similar to a textbook chapter. They introduce fundamental concepts and MATLAB syntax for each module. It was intended that students first read these primers when beginning work on any module and possibly return to them as a valuable reference. However, it was observed by the instructors that students did not read the primers and only used them for the examples they included.

This paper focuses on the re-design of the primers into shorter documents. Each document now addresses a smaller number of key concepts and is designed to be read in under 10 minutes. The more involved algorithm and programming examples have been moved to an additional document intended to be read after the short primers. The static documents are all formatted using a standard template designed to meet accessibility requirements. The authors were part of a team that received a Textbook Transformation Grant in January 2020 from Affordable Learning Georgia[6] making the re-design possible.

Primer Format

The original primers had a prerequisites list; learning objectives; terms; and useful functions, keywords, operators. This took approximately half a page. The majority of the primers consisted of programming fundamentals, MATLAB syntax, code examples, and programming examples including program output. The primers ranged in length from three pages (variables, logic expressions) to 16 pages (structures) with most primers falling between five and 10 pages in length. Conditional structures (if, switch) were five and six pages respectively and iterative structures (for loops, while loops) were nine and ten pages respectively. The primers were on the long side for keeping student attention, and students often did not read the entire primer or only skimmed the primer missing key details in the material. Some students would only watch the VLE videos, also missing key details in the material.

The primers were created using a Microsoft Word template and converted to pdf documents for posting in the VLE. The template was created by one of the authors and was thought to meet accessibility requirements but had a few problems. Screen readers would not recognize the titles of the documents. Formulas were created using MathType[7] which Microsoft Word treats as graphics objects and are ignored by screen readers. Images and code segments were placed in text boxes which made for good formatting but are also ignored by screen readers.

Primer Requirements

The re-designed primers needed to: reflect the current version of MATLAB, have clear examples that increased in difficulty, have a more consistent code style, be accessible, and be shorter in length. As MATLAB Marina would be the primary resource and partially funded through Affordable Learning Georgia[6], all materials needed to be accessible and open educational resources (OER).

The primer length was decided to be 3 - 4 pages maximum. Primers with a large proportion of images were allowed to exceed this. This was based on video lecture length research[8] and what student workers indicated they would be likely to read in one sitting. Reading a document of this length would approximate watching a 5 - 10 minute video. Material would be organized by first providing content introduction, then short illustrative examples, detailed examples, and then any optional or advanced material. For primers that would be broken up: the first primer would have introductory material, syntax, and straight forward examples; the additional primers would have more involved examples and advanced or optional topics.

The design team decided that the prerequisite list was not needed as course syllabi/schedules would indicate the order the modules should be used. The VLE modules are organized in a way that encourages the use in the order specified in the previous prerequisite list. Students find the terms and functions/operators/keywords useful, so they were kept. There was discussion on removing the learning objectives since they would also be in other course materials, but they were ultimately kept although the students indicated they usually skipped reading them. For consistency, program code and commenting standards as well as variable naming conventions were decided.

Copyright needed to less restrictive than the previously used license to meet the OER requirements of the Textbook Transformation Grant. A Creative Commons[9] Attribution-NonCommercial-ShareAlike was chosen to meet the grant OER requirements and several design team members felt very strongly about not letting the work be used for commercial use.

A new Microsoft Word template incorporating all this was created. The template was designed using guidelines for accessibility[10-12]. Screen readers can determine formatting from the styles. The template used a small school logo at the upper right to meet marketing guidelines but be small and unobtrusive. The new template has 18 styles: four types of headings (heading 1 is for titles), three for table titles and row/column headings, seven for bulleted/numbered/lettered lists, two for program code, one for normal text, and one for copyright. All formatting in the re-designed primers is done via the styles. This is necessary for screen readers to detect headings, tables, etc. The program code style uses a non-proportional font (Courier New) and code segments have additional styling of a solid line at the start and end of the code block. This was to clearly separate program code without using text boxes. The inline program code style does not have this. Text boxes were eliminated and any existing formulas or figures would have alt-text added.

Re-Designed Primers

Georgia Southern University courses were delivered remotely after March 13, 2020 due to Covid-19. Minimal work re-designing the primers was accomplished until Summer 2020 due to the work of altering course delivery for the remainder of the Spring semester. Two student workers were hired for most of Summer 2020 through the Textbook Transformation Grant funds. The students were a sophomore computer science major and sophomore computer engineering major each who had taken at least one programming course from one of the authors. The students worked up to 20 hours a week for 10 weeks on the project.

The process for re-designing the primers included:

- Initial edit to identify typos, inaccuracies, and areas that needed clarification and improvement. If the primer length was long; identify how to break up the document.

- Create draft of new primer(s) using template. Add alt-text for any formulas and figures.
- Develop and incorporate additional program examples for topics that did not have enough examples illustrating the concepts. Many of these were created by the students. Update program code and ensure all code is consistent with new coding/commenting style.
- Final edit of new primer(s) by both students and one of faculty team members.

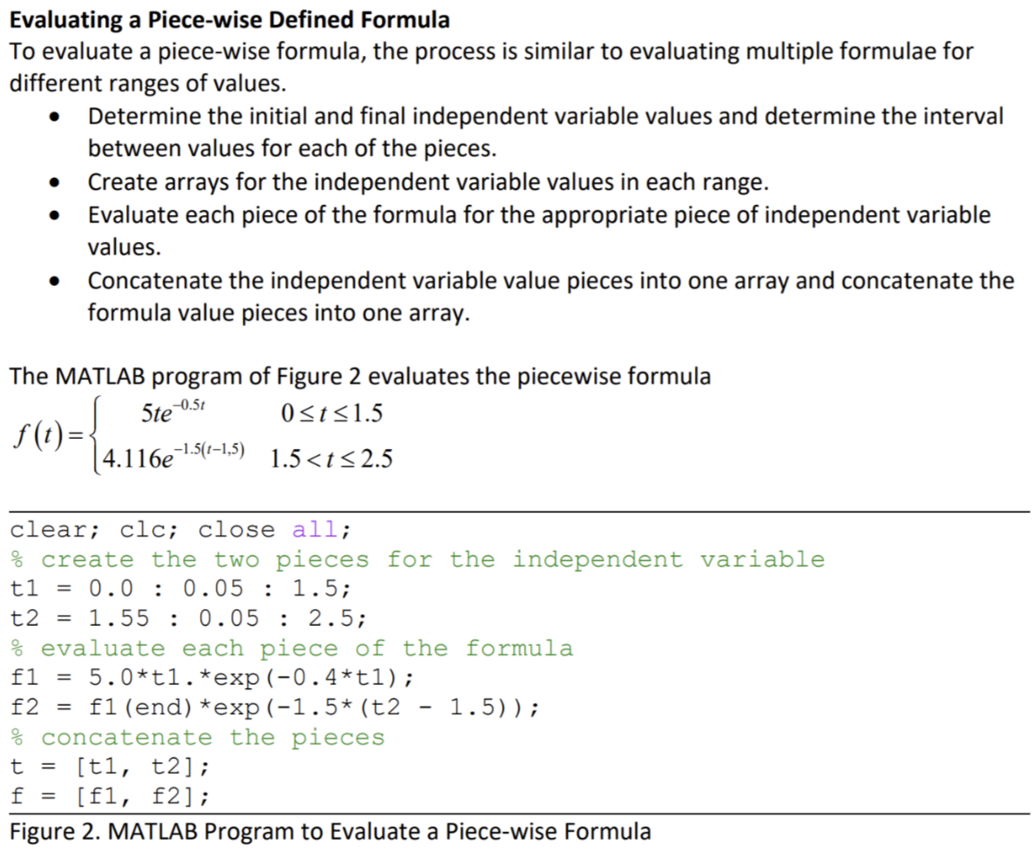
Following are two descriptions of the original and re-designed primers. The other primers have similar modifications.

If-Else Statements

The original If-else primer was six pages. It included MATLAB syntax for the different versions of the if-else statement but no flowcharts and minimal examples. There are two new If-else primers: If-Else Statements (4.5 pages) and If-Else Statements Examples (2.5 pages). The If-Else Statements primer has the learning objectives, terms, etc. and now includes flowcharts showing the different if-elseif-else structures with corresponding MATLAB syntax and MATLAB program code examples. The If-Else Statements Examples primer has two additional detailed examples: determining if a part meets a set of specifications and inline error handling.

1D Arrays

The original 1D Arrays primer was eight and one half pages with a brief description of arrays and a series of short examples including creating arrays, indexing arrays, modifying arrays, operating on arrays, applying built in functions on arrays, comparisons on arrays, and some array examples. It read like a laundry list and did not clearly indicate to students which topics were related. There are three new 1D Arrays primers: Creating and modifying arrays (3.5 pages), Operations on arrays (3.5 pages), and Applying arrays (4 pages). The Creating and modifying arrays includes most of the learning objectives, terms, and functions/keywords/operators. It also includes material and examples on creating arrays, indexing arrays, and modifying arrays. The Operations on arrays primers includes material and examples on element by element operations, applying functions, comparisons on arrays, and naming conventions for array results. The Applying arrays primer has examples illustrating common uses of arrays for engineering: evaluating a formula, evaluating a piece wise formula, analyzing a set of data, extracting a subrange of data, and series summation. Figure 1 shows an example of some of the content and formatting for one of the 1D arrays primers. The updated primers can be found on MATLAB Marina[1].



Evaluating a Piece-wise Defined Formula

To evaluate a piece-wise formula, the process is similar to evaluating multiple formulae for different ranges of values.

- Determine the initial and final independent variable values and determine the interval between values for each of the pieces.
- Create arrays for the independent variable values in each range.
- Evaluate each piece of the formula for the appropriate piece of independent variable values.
- Concatenate the independent variable value pieces into one array and concatenate the formula value pieces into one array.

The MATLAB program of Figure 2 evaluates the piecewise formula

$$f(t) = \begin{cases} 5te^{-0.5t} & 0 \le t \le 1.5 \\ 4.116e^{-1.5(t-1,5)} & 1.5 < t \le 2.5 \end{cases}$$

```
clear; clc; close all;
% create the two pieces for the independent variable
t1 = 0.0 : 0.05 : 1.5;
t2 = 1.55 : 0.05 : 2.5;
% evaluate each piece of the formula
f1 = 5.0*t1.*exp(-0.4*t1);
f2 = f1(end)*exp(-1.5*(t2 - 1.5));
% concatenate the pieces
t = [t1, t2];
f = [f1, f2];
```

Figure 2. MATLAB Program to Evaluate a Piece-wise Formula

Figure 1. Sample 1D Arrays Primer Content

Conclusions and Future Work

MATLAB Marina was used as the primary resource in one section of the Computing Applications in Mechanical Engineering course in Fall 2020. Initial impressions from students who have used some of the re-designed primers are positive. Students like the shorter primers and find the new format clearer and easier to navigate. In Spring 2021, MATLAB Marina with the re-designed primers will be used in all sections of Computing Applications in Mechanical Engineering at Georgia Southern University as the primary course resource for MATLAB. The updated VLE with re-designed primers will be assessed over the course of the spring and changes made as needed.

To assess the efficacy of the modules with the new primers and exercises, a pre-test and post-test will be given to test students' knowledge of MATLAB prior to and after taking the course. Student performance in the course will be measured by compiling aggregate grade distribution in all sections for in-class lab assignments, exams and projects (based on material in the VLE). Additional student surveys will be used through the semester to evaluate the use and readability of the primers. Google Analytics is used to track page usage.

There are MATLAB Marina modules that were not reworked as part of this effort. These modules are used as supplements in other engineering courses, for example there are modules for

Statics and Thermodynamics. These will be re-designed in the future using the same process as described here.

This work is supported by a Textbook Transformation Grant, Affordable Learning Georgia, University System of Georgia, Board of Regents.

References

1 MATLAB Marina, http://www.matlabmarina.com/, 2011 – 2020, MATLAB Marina Team, Allen E. Paulson College of Engineering and Computing, Georgia Southern University

2 Goeser, P.T., Flett, A., Kriske, J. and Panter, C. "MATLAB Marina: Web-Based Tutorials for Teaching Programming Concepts using MATLAB", *American Society for Engineering Education Southeastern Section Annual Conference*, Starkville, MS, April 2012.

3 Goeser, P., Murphy, T., Williams, C., and Johnson, W., "The Effective Use of Virtual Learning Environments like MATLAB Marina as the Primary Resource for an Introductory Programming Course", 2014 ASEE Southeast Section Annual Conference, Macon, GA, March 30 – April 1, 2014.

4 Goeser, P.T., Murphy, T., and Williams, C., "A Virtual Learning Environment as an Open Educational Resource for an Engineering Programming Course", University System of Georgia Teaching and Learning Conference, Athens, GA, April 5 – 7, 2017.

5 Murphy, T., Goeser, P.T., and Williams, C., "Analysis of Usage Statistics of MATLAB Marina - A Virtual Learning Environment", 2018 ASEE Southeast Section Annual Conference, Daytona Beach, FL, March 4 – 6, 2018.

6 Affordable Learning Georgia, https://affordablelearninggeorgia.org/

7 MathType, https://www.dessci.com/en/products/mathtype/, Design Science

8 Video Length in Online Courses: What the Research Says, https://www.qualitymatters.org/qa-resources/resource-center/articles-resources/research-video-length, 2020 Quality matters

9 Creative Commons, https://creativecommons.org/

10 GSA Section508.gov, Create Accessible Documents, https://www.section508.gov/create/documents, GSA U.S. General Services Administration

11 Microsoft Word Creating Accessible Documents, https://webaim.org/techniques/word/, 2020 WebAIM

12 Make your Word documents accessible to people with disabilities, https://support.microsoft.com/en-us/office/make-your-word-documents-accessible-to-people-with-disabilities-d9bf3683-87ac-47ea-b91a-78dcacb3c66d, Microsoft, 2020

Thomas Murphy

Thomas Murphy is an Associate Professor of Electrical and Computer Engineering at Georgia Southern University-Armstrong Campus. He received his PhD and M.E. in Electrical and Computer Engineering from the University of Florida and his B.S. in Electrical Engineering from the University of Notre Dame. His research interests are primarily in the areas of digital systems, control systems, signal processing, engineering education, and computer applications in these areas. Dr. Murphy is a member of The Institute of Electrical and Electronics Engineers (IEEE) and the IEEE Education Society.

Priya T. Goeser

Priya T. Goeser is a Professor of Mechanical Engineering at Georgia Southern University-Armstrong Campus, Savannah GA. She received her Ph.D. in Mechanical Engineering from the

University of Delaware and her B. Tech in Mechanical Engineering from the Indian Institute of Technology, Chennai, India. Her current research interests are structural health monitoring, functionally graded materials and innovative teaching methods in engineering education. She is the primary instructor for the Computing Applications in Mechanical Engineering Course at Armstrong and is the project director of MATLAB Marina.

Christopher Williams

Christopher Williams is a Lecturer at Georgia Southern University-Armstrong Campus in Savannah, GA. He received his B.S. and M.S. in Computer Science from Armstrong Atlantic State University. His research interests are primarily in the areas of mobile computing, usability, and education. Mr. Williams is a member of the Association for Computing Machinery (ACM).