

***Engineering Computation with MATLAB Review***  
**David Smith**

**REVIEWER INFORMATION**

Department: Mechanical Engineering Technology School: Purdue University

**COURSE INFORMATION**

Course Title: Computing in Engineering Technology Grade Level: Freshman

Annual Enrolment: 30

Text In Use: Engineering Computation with MATLAB 3<sup>rd</sup> Edition

**Engineering Computation with MATLAB Review**  
**David Smith**

1. What factors made you chose your current book?
  - a. The coding examples.
  - b. The details provided in a specific chapter allows an instructor to jump or skip chapters without having to follow the chapters in order.
2. How does this book compete with your current text?
  - a. I am using this current text.
3. Please examine the table of contents of *Engineering Computation with MATLAB* and answer the following:
  - a. How does the Table of Contents match with your current course syllabus? Please elaborate.
    - i. I use chapters 1-8, 11, and 12. I do not use the chapters of the text in that order. I teach the chapters in the following order: 1-3, 12, 8, 11, 4-7. I teach Matlab as a tool for analysing data, so I want the input/output and plotting chapters reviewed earlier in the course. I also include chapter 12 early to introduce students to matrix manipulation. That chapter assists students in intro mechanics courses to solve systems of equations.
  - b. Would you change the order of chapters? If so, how?
    - i. Yes, I would move plotting (chapter 11) and input/output (chapter 8) earlier in the text. I know that it works well with the current organization because of the variable types discussed in previous chapters, but I think having students plot and pull in data from other sources is beneficial and allows an instructor to use real world examples in the course with real data to learn about execution control, functions, strings, and cell arrays.
  - c. Do you, or have you, used this text for more than one course? Please explain.
    - i. I have only used it for a single course. We do not have any follow-up programming classes.
2. Please comment on the quality of the book's pedagogy. Are there pedagogical elements in other books you have used that you feel should be incorporated into this text? What are they?
  - a. I think the book's pedagogy is fine. I like how the author includes some history in the first chapter. I think that information allows the students to put what they are learning in perspective.
  - b. I think the self-test are beneficial to the students to make sure that they understand the big concepts in a chapter, but I do not know how much they are used.
  - c. I think the chapter summaries are great and I like how there is a review of new commands at the end of each chapter. Students do not need to flip through the entire chapter looking for that one command to use in their code.
  - d. I like how exercise examples and pitfalls are separated from the text in grey boxes. Students are able to quickly find examples in the chapter because of the box graphics.
  - e. I cannot think of other specific pedagogical elements that I feel should be incorporated into this text.

3. How well does the author present the material in terms of readability, clarity, etc?
  - a. I think the author does a great job in presenting the material. I feel that the author does a good job of explaining why certain things are done, especially for students that are new to programming.
4. Which topics in your course present you with the greatest teaching challenge? How does this text handle these topics?
  - a. I think one of the greatest teaching challenges is incorporating flow charts and teaching how to use flow charts to assist with programming. I think additional discussion of flow charts need to be included in the text. The author shows graphics for flow charts in execution control (chapter 4), but does not do a great job of showing how to create flow charts and how flow charts can be used to assist in programming.
  - b. Another challenge is code organization. The author does a great job of instructing students about comments and showing examples of using comments, but code organization can be improved to allow others to review codes and understand what is going on in the code.
5. Are there other topics that you would suggest deleting or shortening?
  - a. Chapters 13, 14, 16, and 17 appear to be very applied topics. The other chapters of the text provide details that are applicable to introductory programming in engineering. Those later chapters appear to be very specific in the content provided. I think these topics could be placed in a second volume with numerous other examples of specific engineering programming tools. GUIs, controls, and Simulink are examples that initially come to mind where the specific advanced applied topics could be included in a second volume.
6. Are there topics that you would suggest expanding?
  - a. Flow charts: I think the author really needs to include flow chart development and how that can assist you in writing programs.
  - b. Debugging: There is little instruction showing students the debugging tools available in Matlab. I think expanding the debugging discussion would be beneficial.
7. How do you find the quality of the examples? Are there too many or too few?
  - a. I think the quality is fine. I feel there are sufficient details that students can use to improve their own programs. I think most topics have a sufficient number of examples, but I would encourage the author to show one or two larger examples in each chapter of how specific commands can be used to solve a real world programming problem.
8. Do you find that the problems are sufficient? Do they relate well to the material?
  - a. I think the problems are sufficient and relate well to the material. Unfortunately, as with most texts, the answers to the problems are easily found using Google. Therefore, I use the problems as a model to create my own programming problems.
9. How do you find the quality of the illustrations and photos? Are there too many or too few?
  - a. I find the quality of photos and illustrations fine.

10. When considering the adoption of a new edition, what kinds of revisions are most important to you? For example, new problems, new examples, up-to-date references, etc.
  - a. New problems: as stated, old problems have solutions found with Google. Solutions are not initially found in new editions. Once the solution to new editions are found, the older solutions are typically harder to find and those problems are useful again.
11. What supplements do you use in teaching your course? Which would you like to see provided with this text and why? Specifically, are there any particular digital supplements that you would like to see included?
  - a. I use the learning management system Blackboard to teach this course. I provide assignments, notes, exams, etc. from Blackboard. There are no digital supplements that I am interested to see included. I feel that these supplements add extra cost to the students without providing a lot of benefit.
12. Do you utilize online homework in your course? In your opinion, do you think online grading of programming homework would be a useful tool to accompany this text?
  - a. I do use online homework in this course for the concepts with Blackboard tools. Online grading of the programming could be useful, but difficult to implement as an instructor that likes to create custom programming problems.
13. In your opinion what are the best features of this text?
  - a. The end of the chapter section that reviews new characters, words, and functions. This information allows students to look in one place to review new items to incorporate in their code.

**Please provide any additional comments you may have below.**

None.