Engineering Computation with MATLAB®

Third Edition

DAVID M. SMITH
Georgia Institute of Technology

PEARSON

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director: Marcia Horton
Editor in Chief: Michael Hirsch
Editorial Assistant: Emma Snider
Director of Marketing: Patrice Jones
Marketing Manager: Yez Alayan
Marketing Assistant: Kathryn Ferranti
Vice President of Production: Vince O'Brien
Managing Editor: Jeff Holcomb
Production Project Manager: Kayla Smith-Tarbox
Production Editor: Pat Brown
Senior Operations Supervisor: Alan Fischer
Manufacturing Buyer: Pat Brown
Art Director: Anthony Gemmellaro
Creative Director: Jayne Conte
Designer: Kathy Foot
Manager, Visual Research: Karen Sanatar
Manager, Rights and Permissions: Michael Joyce
Media Director: Daniel Sandin
Full-Service Project Management: Kailash Jadli/ Aptara®, Inc.
Composition: Aptara®, Inc.
Printer/Binder: Courier/Westford
Cover Printer: Lehigh-Phoenix Color
Text Font: Palatino Lt Std

Photo Credits: Cover image: The cover image is of a three-dimensional model of HMS Victory developed entirely in Matlab by the Author. Page 4: Fig. 1.1: David Smith; Fig 1.2: Courtesy of the UK Government. Page 25: Fig 2.1: Screenshots copyright © 2011 by MathWorks. Reprinted with permission. Page 28: Fig 2.2, Fig 2.3: Screenshots copyright © 2011 by MathWorks. Reprinted with permission. Page 29: Fig 2.4, Fig 2.5: Screenshots copyright © 2011 by MathWorks. Reprinted with permission. Page 30: Fig 2.6: Screenshots copyright © 2011 by MathWorks. Reprinted with permission. Page 31: Fig 2.7: Screenshots copyright © 2011 by MathWorks. Reprinted with permission. Page 32: Fig 2.8: Screenshots copyright © 2011 by MathWorks. Reprinted with permission. Page 37: Fig. 2.9: AP Photo/Scaled Composites. Page 200: Fig 9.3: © PhotoDisc. Page 302: Fig 13.10: David Smith. Page 397: Fig 17.4: jorgen mcleman/Shutterstock.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. Screen shots and icons reprinted with permission from the Microsoft Corporation. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

Library of Congress Cataloging-in-Publication Data

Smith, David M.
Engineering and computation with MATLAB/David M. Smith.—3rd ed.
p. cm.
ISBN 978-0-13-256870-8
1. Engineering mathematics—Data processing. 2. MATLAB. I. Title.
TA345.S585 2013
620.00285'53—dc23

2012000377

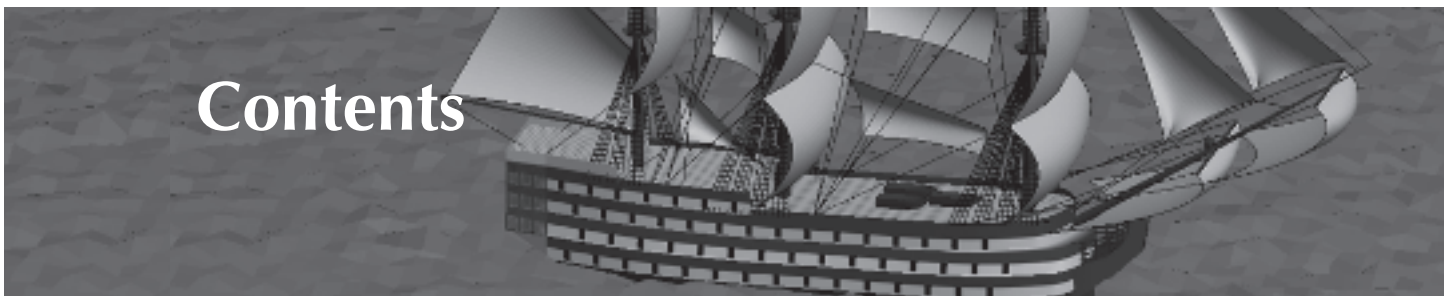
10 9 8 7 6 5 4 3 2 1

PEARSON

ISBN 10: 0-13-256870-5
ISBN 13: 978-0-13-256870-8

This book is dedicated to the glory of Almighty God

~David M. Smith

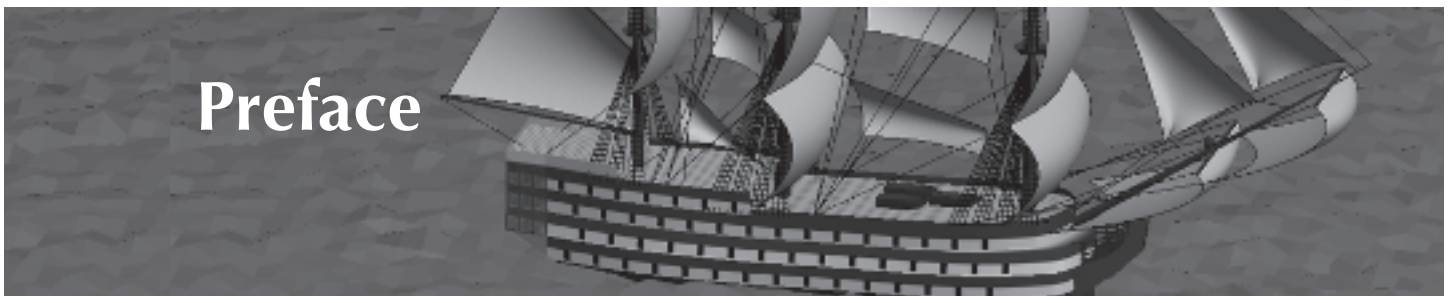


Contents

About the Author

David Smith has been teaching introductory computer science classes for engineers at the Georgia Institute of Technology since 1997 when he retired from industry. Previously, he worked 31 years for Lockheed-Martin at its Marietta, Georgia, facility as a systems and software specialist with a focus on intelligent systems. He was active in designing and developing software for the C-130J, C-27J, F-22, and C-5 aircraft and was the technical leader of the Pilot's Associate program, a $64 million research project sponsored by the Defense Advanced Research Projects Agency.

Mr. Smith has a bachelor's degree in aeronautical engineering from Southampton University and a master's degree in control systems from Imperial College, London.



Preface

"That of all the several ways of beginning a book which are now in practice throughout the known world, I am confident my own way of doing it is the best—I'm sure it is the most religious—for I begin with writing the first sentence, and trusting to Almighty God for the second."

Laurence Sterne (1713–1768), British author, clergyman

This book introduces the power, satisfaction, and joy of computing to beginning engineering students who have little or no previous computing experience. It began as a snapshot of the content of a Georgia Tech course that introduces engineers to computing. However, it has been extensively enhanced to meet the needs of a wider audience of students and educators who want to understand programming for other reasons. In this book, to understand computing, we use the basic syntax and capabilities of MATLAB, a user-friendly language that is emerging as one of the most popular computing languages in engineering.

New to the Third Edition

Many engineering disciplines use the concept of graphs to represent specific ideas. We have added a chapter that deals with the fundamentals of graph manipulation from an engineering standpoint—specifically, how to find a minimum spanning tree, and both exact and approximate methods for finding the best path from one point to another. We also try to note those new features of MATLAB that are relevant to students in an introductory programming class. For examples, features were added recently allowing a user to manipulate plotted data by adjusting and saving values. Although interesting, one can achieve the same result with more traceability and repeatability by editing to the source data and repeating the plots.

One interesting observation emerged when refreshing the analysis of sorting algorithms in Chapter 16. In older versions of MATLAB (prior to R2008), our crude recursive implementations of Merge Sort and Quick Sort did not achieve the expected performance. The reason we deduced was that when data are passed into and out of a function, they must be copied between the workspaces of the calling and called functions. With R2011, however, the same code works splendidly, suggesting that the earlier inefficient parameter passing mechanisms have been significantly improved.

Pedagogical Style

Computing is not a spectator sport. Students learn computing by using a computing system to solve problems. This text not only presents computing concepts and their MATLAB implementation, but also offers students extensive hands-on exercises. The text illustrates the ideas with examples from the world of engineering, provides style points, and presents sample problems that students might encounter. Each chapter includes topics that go a step beyond the basic content of an introductory class. This gives professors the choice to progress slowly, and more thoroughly, through the material in two semesters. It also offers advanced students enrichment materials for their personal study.

The overall philosophy of this text approaches programming tools in the following manner:

1. Explain a computing concept in general
2. Discuss its implementation in MATLAB
3. Provide exercises to master the concept

To help facilitate students' understanding of the concept and its implementation, the text uses two features: general templates and MATLAB listings. The general templates provide a foundation for students to understand concepts in general and can be applied to any language. The MATLAB listings show students how to implement concepts in MATLAB and are followed by detailed explanations of the code.

Features of the Text

- **Exercises:** Allow students a "Do It Yourself" approach to master concepts by trying what they just learned. Exercises follow each new topic.
- **Style Points:** Advise students about writing quality code that is easy to understand, debug, and reuse.
- **Hints:** Enrich students' understanding of a topic. Hints are interspersed through the book at points where students may benefit from a little extra "aside."
- **Engineering Examples:** Provide robust models and apply to real-world issues that will motivate students. Examples from different engineering disciplines are presented at the end of each chapter.
- **Special Characters, Reserved Words, and Functions:** Provides a quick reference for the key MATLAB principles discussed in each chapter.
- **Self Test:** Helps students to check their understanding of the material in each chapter.
- **Programming Projects:** Offer a variety of large-scale projects that students can work on to solidify their skills.

Chapter Overview

Chapter 1: *Introduction to Computers and Programming* discusses the history of computer architectures as they apply to computing systems today. The chapter provides an overview of computer hardware and software and how programs execute.

Chapter 2: *Getting Started* discusses some basic concepts of computing and then introduces the basic operation of the MATLAB user interface. The chapter also describes how to capture simple MATLAB programs in the form of scripts.

Chapter 3: *Vectors and Arrays* introduces the fundamental machinery that sets MATLAB apart from other languages—its ability to perform mathematical and logical operations on homogeneous collections of numbers.

Chapter 4: *Execution Control* describes the common techniques used to control the execution of code blocks—conditional operation and iteration.

Chapter 5: *Functions* describes how to implement procedural abstraction by defining reusable code blocks.

Chapter 6: *Character Strings* discusses how MATLAB operates on variables containing text.

Chapter 7: *Cell Arrays and Structures* discusses two kinds of heterogeneous data collections accessed by index and by name.

Chapter 8: *File Input and Output* describes three levels of ability provided in MATLAB for transferring data to and from data files—saving workspaces, specific tools that read and write specific data files, and general-purpose tools for processing any kind of file.

Chapter 9: *Recursion* discusses and illustrates a widely used alternative approach to repetitive code execution.

Chapter 10: *Principles of Problem Solving* introduces ideas that help students design solutions to new problems and avoid the "blank sheet of paper" syndrome—how to start a program.

Chapter 11: *Plotting* takes the student from basic plotting in two dimensions to the advanced tools that draw representations of three-dimensional objects with smooth shading and even multiple light effects.

Chapter 12: *Matrices* describes specific MATLAB capabilities that implement matrix algebra.

Chapter 13: *Images* discusses how to use vector and array algebra to manipulate color pictures.

Chapter 14: *Processing Sound* shows how to analyze, synthesize, and operate on sound files.

Chapter 15: *Numerical Methods* introduces numerical techniques that commonly occur in engineering: interpolation, curve fitting, integration, and differentiation.

Chapter 16: *Sorting* presents five algorithms for ordering data, each of which has applicability under certain circumstances—Insertion Sort, Bubble Sort, Quick Sort, Merge Sort, and Radix Sort—and then compares their performance on large quantities of data.

Chapter 17: *Processing Graphs* discusses how to represent graphs in general and then how to solve two important engineering problems—finding a minimal spanning tree and finding an optimal path between two nodes of the graph.

Appendices provide a summary of the MATLAB special characters, reserved words, and functions used throughout the text, the ASCII character set, the internal number representation inside the computer, and answers to the True or False and Fill in the Blank questions.

Paths through the Book

Not all courses that cover programming and MATLAB follow the same syllabus. *Engineering Computation with MATLAB* is designed to facilitate teaching the material with different styles and at different speeds. For example, Chapters 3, 4, and 5 cover MATLAB array manipulation, iteration, and writing your own functions. There are three schools of thought about the appropriate way to introduce these concepts. One would introduce array constructs first and follow up with the more "traditional" concept of iteration; another would teach iteration first and deal with the MATLAB-specific array operations later; and the third would treat functions first. I chose to order the book according to the arrays-first approach, to suit a particular teaching style. However, should you prefer iteration or functions first, Chapters 3, 4, and 5 can be used in any order you wish. In practice, over the years, our course has shifted to a functions-first approach so that we can use function interfaces to isolate students' code for automated code grading. Chapters 6–9 should be taught in sequence—there are dependencies between chapters that would make it awkward change the order. Chapter 10 is an important chapter that is difficult to place on a class schedule. Where it stands in the book appears to be a logical position. However, at that point in the semester, beginning students are still not ready to think about larger problems. I have usually covered this material (if at all) at the end of each semester by way of review. Chapter 11 provides basic plotting capability and is necessary for the remaining chapters. After that, Chapters 12–17 are virtually independent and can be taught in any order, but should follow Chapters 2–9 and 11.

Supplements

Various supplemental materials for this text are available at the book's Companion Web site: www.pearsonhighered.com/smith. The following are accessible to all readers:

- Solutions to selected Programming Projects
- Selected full-color figures
- Source code for all MATLAB listings
- Bonus chapters including: Object-Oriented Programming, Linked Lists, N-ary Trees and Graphs, and the Cost of Computing

In addition, the following supplements are available to qualified instructors at Addison-Wesley's Instructor Resource Center. Please visit www.pearsonhighered.com/irc, or send an e-mail to computing@aw.com.

- Solutions to all of the Programming Projects
- PowerPoint lecture slides

Acknowledgments

The underlying philosophy of this book and the material that forms its skeleton originated in the work of Professor Russell Shackelford around 1996. Dr. Melody Moore, currently an Associate Professor in the Interactive Computing department of the College of Computing at Georgia Tech, was instrumental in creating many of the teaching materials (then as overhead transparencies) from which this class was first taught. I am deeply indebted to Professor James Craig from the Aerospace Engineering department at Georgia Tech, who joined me in co-teaching the first engineering version of CS1, taught me much about MATLAB, and pioneered this class from the original 35 students to its current size of over 1,000 engineering students per semester. This engineering class became a vessel for introducing the students to the MATLAB language.

I would like to thank the following reviewers for their insight and wisdom during the process of manuscript development:

Kenneth Rouse, *Auburn University*
Suparna Datta, *Northeastern University*
Gerardine G. Botte, *Ohio University*
Mica Grujicic, *Clemson University*
Kuldip S. Rattan, *Wright State University*
Y.J. Lin, *The University of Akron*
Mark Nagurka, *Marquette University*
Michael Peshkin, *Northwestern University*
Howard Silver, *Fairleigh Dickinson University*
Steve Swinnea, *The University of Texas at Austin*

The material has benefited from the efforts of every Georgia Tech teaching assistant (TA), graduate student, instructor, and professor who has taught CS1, a list too long to enumerate. In particular, those wonderfully creative TAs who developed the ideas for examples used in this text have enriched it immeasurably. I wish to credit Professor Aaron Bobick with an important contribution made in the course of one short conversation. That conversation was responsible for pulling the class back from the brink of being merely a MATLAB programming class to one with roots in CS concepts. Professor Bobick taught CS1 with me in the fall of 2004. Early in the semester he made a very simple request: he said it would be easier for him to teach the class if we explicitly expressed the computing concepts inherent in each lesson, rather than leaving him—and the students—to tease the concepts out of the teaching materials.

I cannot adequately express my appreciation for the team at Addison-Wesley that helped bring this book to fruition. Many of them have done, and I am sure continue to do, their work "behind the scenes": Michael Hirsch, Emma Snider, Yez Alayan, Kayla Smith-Tarbox, Pat Brown, and Jeff Holcomb. I also really appreciate the work of Kailash Jadli and his team at Aptara®, Inc. for the care with which they designed the third edition.

Most important, I would like to acknowledge the personal contributions of those people without whom this book would not exist. My wife and best friend, Julie, has been an unwavering source of strength and encouragement during the process of writing this text. Bill Leahy was a student in the first CS1 class I taught in the spring of 1998. In spite of this beginning, he continued to a master's in computer science from Tech and is now an instructor in the College of Computing. Beyond his uncountable technical contributions to the material in this book, I want to acknowledge his friendship, encouragement, and wise judgment, all of which have been an inspiration to me during the process of developing this text.