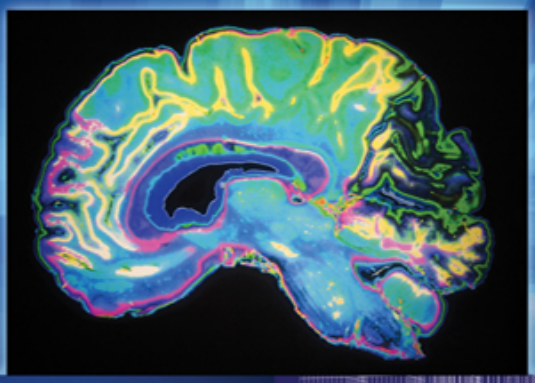
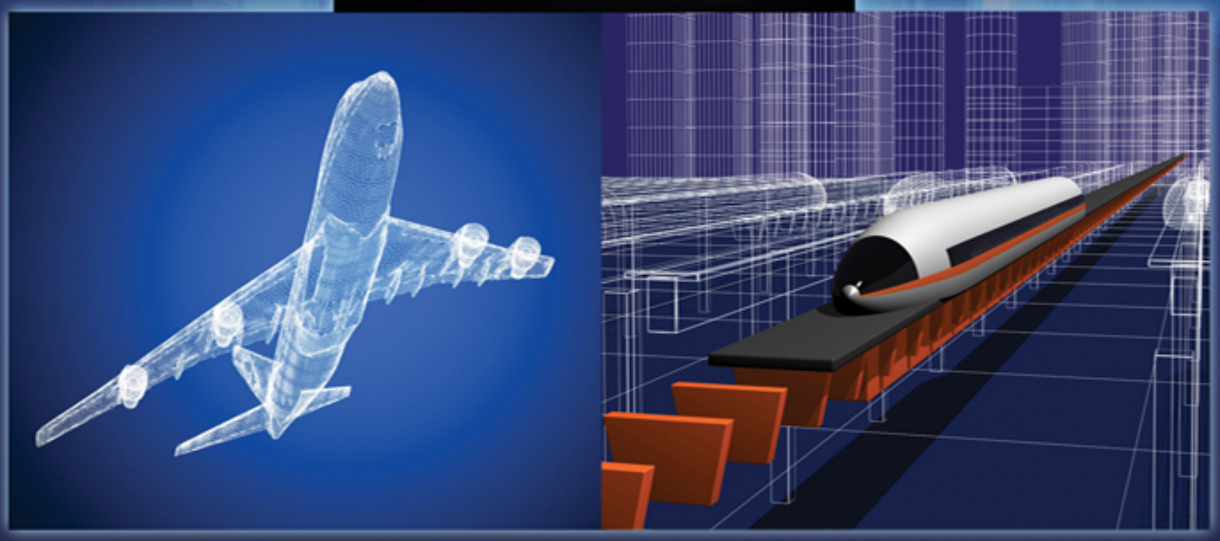
MATLAB®

PROGRAMMING

with Applications for Engineers





STEPHEN J. CHAPMAN

MATLAB® Programming with Applications for Engineers

MATLAB® Programming with Applications for Engineers

First Edition

Stephen J. Chapman
BAE Systems Australia

CENGAGE Learning™

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

MATLAB® Programming with Applications for Engineers
Stephen J. Chapman

Publisher, Global Engineering: Christopher M. Shortt
Senior Acquisitions Editor: Randall Adams
Senior Developmental Editor: Hilda Gowans
Editorial Assistant: Tanya Altieri
Team Assistant: Carly Rizzo
Marketing Manager: Lauren Betsos
Media Editor: Chris Valentine
Content Project Manager: D. Jean Buttrom
Production Service: RPK Editorial Services, Inc.
Copyeditor: Shelly Gerger-Knechtl
Proofreader: Harlan James
Indexer: Shelly Gerger-Knechtl
Compositor: Integra Software Solutions
Senior Art Director: Michelle Kunkler
Internal Designer: Carmela Periera
Cover Designer: Andrew Adams/4065042 Canada Inc.
Cover Image: © ivn3da/Shutterstock; © Martin Trajkovski/Shutterstock; © Daisy Daisy/Shutterstock; © PhotoStocker/Shutterstock
Rights Acquisitions Specialist: John Hill
Text and Image Permissions Researcher: Kristiina Paul
Senior First Print Buyer: Doug Wilke

Library of Congress Control Number: 2011934123

ISBN-13: 978-0-495-66807-7

ISBN-10: 0-495-66807-9

Cengage Learning
200 First Stamford Place, Suite 400
Stamford, CT 06902
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at: **international.cengage.com/region**.

Cengage Learning products are represented in Canada by Nelson Education Ltd.

For your course and learning solutions, visit **www.cengage.com/engineering**.

Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**.

Printed in the United States of America
1 2 3 4 5 6 7 13 12 11

This book is dedicated with love to my daughter Sarah Rivkah Chapman. As a student at Swinburne University in Melbourne, she may actually wind up using it!

About the Author

Stephen J. Chapman received a BS in Electrical Engineering from Louisiana State University (1975), an MSE in Electrical Engineering from the University of Central Florida (1979), and pursued further graduate studies at Rice University.

From 1975 to 1980, he served as an officer in the U.S. Navy, assigned to teach Electrical Engineering at the U.S. Naval Nuclear Power School in Orlando, Florida. From 1980 to 1982, he was affiliated with the University of Houston, where he ran the power systems program in the College of Technology.

From 1982 to 1988 and from 1991 to 1995, he served as a Member of the Technical Staff of the Massachusetts Institute of Technology's Lincoln Laboratory, both at the main facility in Lexington, Massachusetts, and at the field site on Kwajalein Atoll in the Republic of the Marshall Islands. While there, he did research in radar signal processing systems. He ultimately became the leader of four large operational range instrumentation radars at the Kwajalein field site (TRADEX, ALTAIR, ALCOR, and MMW).

From 1988 to 1991, Mr. Chapman was a research engineer in Shell Development Company in Houston, Texas, where he did seismic signal processing research. He was also affiliated with the University of Houston, where he continued to teach on a part-time basis.

Mr. Chapman is currently Manager of Systems Modeling and Operational Analysis for BAE Systems Australia, in Melbourne, Australia. He is the leader of a team that has developed a model of how naval ships defend themselves against antiship missile attacks. This model contains more than 400,000 lines of MATLAB™ code written over more than a decade, so he has extensive practical experience applying MATLAB to real-world problems.

Mr. Chapman is a Senior Member of the Institute of Electrical and Electronic Engineers (and several of its component societies). He is also a member of the Association for Computing Machinery and the Institution of Engineers (Australia).

Contents

Chapter 2 MATLAB Basics 25

Preface

MATLAB® (short for MATrix LABoratory) is a special-purpose computer program optimized to perform engineering and scientific calculations. It started life as a program designed to perform matrix mathematics, but over the years it has grown into a flexible computing system capable of solving essentially any technical problem.

The MATLAB program implements the MATLAB language and provides a very extensive library of pre-defined functions to make technical programming tasks easier and more efficient. This extremely wide variety of functions makes it much easier to solve technical problems in MATLAB than in other languages such as Java, Fortran, or C++. This book introduces the MATLAB language, and shows how to use it to solve typical technical problems.

This book seeks to simultaneously teach MATLAB as a technical programming language and also to introduce the student to many of the practical functions that make solving problems in MATLAB so much easier than in other languages. The book provides a complete introduction to the fundamentals of good procedural programming, developing good design habits that will serve a student well in any other language that he or she may pick up later. There is a very strong emphasis on proper program design and structure. A standard program design process is introduced at the beginning of Chapter 4 and then followed regularly throughout the remainder of the text.

In addition, the book uses the programming topics and examples as a jumping off point for exploring the rich set of highly optimized application functions that are built directly into MATLAB. For example, in Chapter 4 we present a programming example that finds the roots of a quadratic equation. This serves as a jumping off point for exploring the MATLAB function `roots`, which can efficiently find the

roots of polynomials of any order. In Chapter 5, we present a programming example that calculates the mean and standard deviation of a data set. This serves as a jumping off point for exploring the MATLAB functions `mean`, `median`, and `std`. There is also a programming example showing how to do a least-squares fit to a straight-line. This serves as a jumping off point for exploring MATLAB curve fitting functions such as `polyfit`, `polyval`, `spline`, and `ppval`. There are similar ties to MATLAB applications in many other chapters as well. In all cases, there are end of chapter exercises to reinforce the applications lessons learned in that chapter.

In addition, Chapter 11 is devoted totally to practical MATLAB applications, including solving systems of simultaneous equations, numerical differentiation, numerical integration (quadrature), and solving ordinary differential equations.

This book makes no pretense at being a complete description of all of MATLAB's hundreds of functions. Instead, it teaches the student how to use MATLAB as a language to solve problems, and how to locate any desired function with MATLAB's extensive on-line help facilities. It highlights quite a few of the key engineering applications, but there are far more good ones built into the language than can be covered in any course of reasonable length. With the skills developed here, students will be able to continue discovering features on their own.

The Advantages of MATLAB for Problem Solving

MATLAB has many advantages compared to conventional computer languages for technical problem solving. Among them are:

1. **Ease of Use.** MATLAB is very easy to use. The program can be used as a scratch pad to evaluate expressions typed at the command line, or it can be used to execute large pre-written programs. Programs may be easily written and modified with the built-in integrated development environment, and debugged with the MATLAB debugger. Because the language is so easy to use, it is ideal for educational use, and for the rapid prototyping of new programs.

   Many program development tools are provided to make the program easy to use. They include an integrated editor / debugger, on-line documentation and manuals, a workspace browser, and extensive demos.

2. **Platform Independence.** MATLAB is supported on many different computer systems, providing a large measure of platform independence. At the time of this writing, the language is supported on Windows XP/Vista/7, Linux, Unix, and the Macintosh. Programs written on any platform will run on all of the other platforms, and data files written on any platform may be read transparently on any other platform. As a result, programs written in MATLAB can migrate to new platforms when the needs of the user change.

3. **Pre-defined Functions.** MATLAB comes complete with an extensive library of pre-defined functions that provide tested and pre-packaged solutions to many basic technical tasks. For example, suppose that you are writing a program that must calculate the statistics associated with an input data set. In most languages, you would need to write your own subroutines or functions to implement calculations such as the arithmetic mean, standard deviation, median, etc. These and hundreds of other functions are built right into the MATLAB language, making your job much easier.

   The built-in functions can solve an astonishing range of problems, such as solving systems of simultaneous equations, sorting, plotting, finding roots of equations, numerical integration, curve fitting, solving ordinary and partial differential equations, and much, much more.

   In addition to the large library of functions built into the basic MATLAB language, there are many special-purpose toolboxes available to help solve complex problems in specific areas. For example, a user can buy standard toolboxes to solve problems in Signal Processing, Control Systems, Communications, Image Processing, and Neural Networks, among many others.

4. **Device-Independent Plotting.** Unlike other computer languages, MATLAB has many integral plotting and imaging commands. The plots and images can be displayed on any graphical output device supported by the computer on which MATLAB is running. This capability makes MATLAB an outstanding tool for visualizing technical data. Plotting is introduced in Chapter 2, and covered extensively in Chapters 3 and 8. Advanced features such as animations and movies are covered in Chapter 10.

5. **Graphical User Interface.** MATLAB includes tools that allow a program to interactively construct a Graphical User Interface (GUI) for his or her program. With this capability, the programmer can design sophisticated data analysis programs that can be operated by relatively-inexperienced users.

Features of this Book

Many features of this book are designed to emphasize the proper way to write reliable MATLAB programs. These features should serve a student well as he or she is first learning MATLAB, and should also be useful to the practitioner on the job. They include:

1. **Emphasis on Top-Down Design Methodology.** The book introduces a top-down design methodology in Chapter 4, and then uses it consistently throughout the rest of the book. This methodology encourages a student

to think about the proper design of a program *before* beginning to code. It emphasizes the importance of clearly defining the problem to be solved and the required inputs and outputs before any other work is begun. Once the problem is properly defined, it teaches the student to employ stepwise refinement to break the task down into successively smaller sub-tasks, and to implement the subtasks as separate subroutines or functions. Finally, it teaches the importance of testing at all stages of the process, both unit testing of the component routines and exhaustive testing of the final product.

The formal design process taught by the book may be summarized as follows:

1. *Clearly state the problem that you are trying to solve.*
2. *Define the inputs required by the program and the outputs to be produced by the program.*
3. *Describe the algorithm that you intend to implement in the program.* This step involves top-down design and stepwise decomposition, using pseudocode or flow charts.
4. *Turn the algorithm into MATLAB statements.*
5. *Test the MATLAB program.* This step includes unit testing of specific functions, and also exhaustive testing of the final program with many different data sets.

2. **Emphasis on Functions.** The book emphasizes the use of functions to logically decompose tasks into smaller subtasks. It teaches the advantages of functions for data hiding. It also emphasizes the importance of unit testing functions before they are combined into the final program. In addition, the book teaches about the common mistakes made with functions, and how to avoid them.

3. **Emphasis on MATLAB Tools.** The book teaches the proper use of MATLAB's built-in tools to make programming and debugging easier. The tools covered include the Editor / Debugger, Workspace Browser, Help Browser, and GUI design tools.

4. **Emphasis on MATLAB applications.** The book teaches how to harness the power of MATLAB's rich set of functions to solve a wide variety of practical engineering problems. This introduction to MATLAB functions is spread throughout the book, and is generally tied to the topics and examples being discussed in a particular chapter.

5. **Good Programming Practice Boxes.** These boxes highlight good programming practices when they are introduced for the convenience of the student. In addition, the good programming practices introduced in a chapter are summarized at the end of the chapter. An example Good Programming Practice Box is shown below.

Good Programming Practice:

Always indent the body of an `if` construct by 2 or more spaces to improve the readability of the code.

6. **Programming Pitfalls Boxes**
   These boxes highlight common errors so that they can be avoided. An example Programming Pitfalls Box is shown below.

Programming Pitfalls:

Make sure that your variable names are unique in the first 63 characters. Otherwise, MATLAB will not be able to tell the difference between them.

Pedagogical Features

This book includes several features designed to aid student comprehension. A total of 13 quizzes appear scattered throughout the chapters, with answers to all questions included in Appendix D. These quizzes can serve as a useful self-test of comprehension. In addition, there are approximately 215 end-of-chapter exercises. Answers to all exercises are included in the Instructor's Manual. Good programming practices are highlighted in all chapters with special Good Programming Practice boxes, and common errors are highlighted in Programming Pitfalls boxes. End of chapter materials include Summaries of Good Programming Practice and Summaries of MATLAB Commands and Functions.

The book is accompanied by an Instructor's Manual, containing the solutions to all end-of-chapter exercises. The IM, PowerPoint slides of all figures and tables in the book and the source code for all examples in the book is available from the book's Web site, and the source code for all solutions in the Instructor's Manual is available separately to instructors.

To access additional course materials [including CourseMate], please visit www.cengagebrain.com. At the cengagebrain.com home page, search for the ISBN of your title (from the back cover of your book) using the search box at the top of the page. This will take you to the product page where these resources can be found.

A Thank You to the Reviewers

I would like to offer a special thank you to the book's reviewers. Their invaluable suggestions have made this a significantly better book, and they certainly deserve thanks for the time they devoted to reviewing drafts of the text. The reviewers who were willing to be named are:

Steven A. Peralta, University of New Mexico
Jeffrey Ringenberg, University of Michigan
Lizzie Santiago, West Virginia University
John R. White, University of Massachusetts, Lowell

A Final Note to the User

No matter how hard I try to proofread a document like this book, it is inevitable that some typographical errors will slip through and appear in print. If you should spot any such errors, please drop me a note via the publisher, and I will do my best to get them eliminated from subsequent printings and editions. Thank you very much for your help in this matter.

I will maintain a complete list of errata and corrections at the book's World Wide Web site, which is http://www.cengage.com/engineering. Please check that site for any updates and/or corrections.

STEPHEN J. CHAPMAN
Melbourne, Australia

CHAPTER 1

Introduction to MATLAB

MATLAB (short for MATrix LABoratory) is a special-purpose computer program optimized to perform engineering and scientific calculations. It started life as a program designed to perform matrix mathematics, but over the years, it has grown into a flexible computing system capable of solving essentially any technical problem.

The MATLAB program implements the MATLAB programming language and provides an extensive library of predefined functions to make technical programming tasks easier and more efficient. This book introduces the MATLAB language as it is implemented in MATLAB Version 7.9 and shows how to use it to solve typical technical problems.

MATLAB is a huge program, with an incredibly rich variety of functions. Even the basic version of MATLAB without any toolkits is much richer than other technical programming languages. There are more than 1000 functions in the basic MATLAB product alone, and the toolkits extend this capability with many more functions in various specialties. Furthermore, these functions often solve very complex problems (solving differential equations, inverting matrices, and so forth) in a *single step*, saving large amounts of time. Doing the same thing in another computer language usually involves writing complex programs yourself or buying a third-party software package (such as IMSL or the NAG software libraries) that contains the functions.

The built-in MATLAB functions are almost always better than anything that an individual engineer could write on his or her own, because many people have worked on them and they have been tested against many different data sets. These functions are also robust, producing sensible results for wide ranges of input data and gracefully handling error conditions.