

Preface

MOTIVATION

The purpose of this book is to teach basic programming concepts and skills needed for basic problem solving, all using MATLAB[®] as the vehicle. MATLAB is a powerful software package that has built-in functions to accomplish a diverse range of tasks, from mathematical operations to three-dimensional imaging. Additionally, MATLAB has a complete set of programming constructs that allows users to customize programs to their own specifications.

There are many books that introduce MATLAB. There are two basic flavors of these books: those that demonstrate the use of the built-in functions in MATLAB, with a chapter or two on some programming concepts, and those that cover only the programming constructs without mentioning many of the built-in functions that make MATLAB efficient to use. Someone who learns just the built-in functions will be well-prepared to use MATLAB, but would not understand basic programming concepts. That person would not be able to then learn a language such as C++ or Java without taking another introductory course, or reading another book, on the programming concepts. Conversely, anyone who learns only programming concepts first (using any language) would tend to write highly inefficient code using control statements to solve problems, not realizing that in many cases these are not necessary in MATLAB.

Instead, this book takes a hybrid approach, introducing both the programming and the efficient uses. The challenge for students is that it is nearly impossible to predict whether they will in fact need to know programming concepts later on or whether a software package such as MATLAB will suffice for their careers. Therefore, the best approach for beginners is to give them both: the programming concepts and the efficient built-in functions. Since MATLAB is very easy to use, it is a perfect platform for this approach of teaching programming and problem solving.

As programming concepts are critically important to this book, emphasis is not placed on the time-saving features that evolve with every new MATLAB release.

For example, in most versions of MATLAB, statistics on variables are available readily in the Workspace Window. This is not shown with any detail in the book, as whether this feature is available depends on the version of the software and because of the desire to explain the concepts in the book.

MODIFICATIONS IN FIFTH EDITION

The changes in the Fifth Edition of this book include the following.

- Use of MATLAB Version R2018a
- A revised Text Manipulation chapter, which includes manipulating character vectors as well as the new **string** type
- Introduction to alternate MATLAB platforms, including MATLAB Mobile
- Local functions within scripts
- The new output format for most expression types
- Introduction to the RESTFUL web functions which import data from websites
- Increased coverage of App Designer, which may eventually replace GUIs and uses object-oriented programming
- Introduction to recording audio from a built-in device such as a microphone
- Modified and new end-of-chapter exercises
- More coverage of data structures including categorical arrays and tables
- Increased coverage of built-in functions in MATLAB
- Modifications to symbolic math expressions

KEY FEATURES

Side-By-Side Programming Concepts and Built-In Functions

The most important and unique feature of this book is that it teaches programming concepts and the use of the built-in functions in MATLAB, side-by-side. It starts with basic programming concepts such as variables, assignments, input/output, selection, and loop statements. Then, throughout the rest of the book, many times a problem will be introduced and then solved using the “programming concept” and also using the “efficient method.” This will not be done in every case to the point that it becomes tedious, but just enough to get the ideas across.

Systematic Approach

Another key feature is that the book takes a very systematic, step-by-step approach, building on concepts throughout the book. It is very tempting in a MATLAB text to show built-in functions or features early on with a note that

says “we’ll do this later”. This book does not do that; functions are covered before they are used in examples. Additionally, basic programming concepts will be explained carefully and systematically. Very basic concepts such as looping to calculate a sum, counting in a conditional loop, and error-checking are not found in many texts, but are covered here.

Data Transfer

Many applications in engineering and the sciences involve manipulating large data sets that are stored in external files. Most MATLAB texts at least mention the **save** and **load** functions, and in some cases, also some of the lower-level file input/output functions. As file input and output is so fundamental to so many applications, this book will cover several low-level file input/output functions, as well as reading from and writing to spreadsheet files. Later chapters will also deal with audio and image files. These file input/output concepts are introduced gradually: first **load** and **save** in [Chapter 3](#), then lower-level functions in [Chapter 9](#), and finally sound and images in [Chapter 13](#). A brief introduction to RESTFUL web functions, which import data from websites is given in [Chapter 9](#).

User-Defined Functions

User-defined functions are a very important programming concept, and yet, many times the nuances and differences between concepts such as types of functions and function calls versus function headers can be very confusing to beginning programmers. Therefore, these concepts are introduced gradually. First, arguably the easiest types of functions to understand, those that calculate and return one single value, are demonstrated in [Chapter 3](#). Later, functions that return no values and functions that return multiple values are introduced in [Chapter 6](#). Finally, advanced function features are shown in [Chapter 10](#).

Advanced Programming Concepts

In addition to the basics, some advanced programming concepts such as text manipulation, data structures (e.g., structures and cell arrays), recursion, anonymous functions, and variable number of arguments to functions are covered. Sorting and indexing are also addressed. All of these are again approached systematically; for example, cell arrays are covered before they are used in file input functions and as labels on pie charts.

Problem-Solving Tools

In addition to the programming concepts, some basic mathematics necessary for solving many problems will be introduced. These will include statistical functions, solving sets of linear algebraic equations, and fitting curves to data.

The use of complex numbers and some calculus (integration and differentiation) will also be introduced. The built-in functions in MATLAB to perform these tasks will be described.

Plots, Imaging, and Graphical User Interfaces

Simple two-dimensional plots are introduced very early in the book ([Chapter 3](#)) so that plot examples can be used throughout. A separate chapter, [Chapter 12](#), shows more plot types and demonstrates customizing plots and how the graphics properties are handled in MATLAB. This chapter makes use of text and cell arrays to customize labels. Also, there is an introduction to image processing and the basics necessary to understand programming Graphical User Interfaces (GUIs) in [Chapter 13](#). App Designer, which creates GUIs using object-oriented code, is also introduced in [Chapter 13](#).

Vectorized Code

Efficient uses of the capabilities of the built-in operators and functions in MATLAB are demonstrated throughout the book. In order to emphasize the importance of using MATLAB efficiently, the concepts and built-in functions necessary for writing vectorized code are treated very early in [Chapter 2](#). Techniques such as preallocating vectors and using logical vectors are then covered in [Chapter 5](#) as alternatives to selection statements and looping through vectors and matrices. Methods of determining how efficient the code is are also covered.

Object-Oriented Programming

Creating objects and classes in MATLAB has been an option for some time, but as of R2014b, all Graphics objects are truly objects. Thus, object-oriented programming (OOP) is now a very important part of MATLAB programming. Applications using App Designer reinforce the concepts.

LAYOUT OF TEXT

This text is divided into two parts: the first part covers programming constructs and demonstrates the programming method versus efficient use of built-in functions to solve problems. The second part covers tools that are used for basic problem solving, including plotting, image processing, and techniques to solve systems of linear algebraic equations, fit curves to data, and perform basic statistical analyses. The first six chapters cover the very basics in MATLAB and in programming and are all prerequisites for the rest of the book. After that, many chapters in the problem-solving section can be introduced when desired, to

produce a customized flow of topics in the book. This is true to an extent, although the order of the chapters has been chosen carefully to ensure that the coverage is systematic.

The individual chapters are described here, as well as which topics are required for each chapter.

PART 1: INTRODUCTION TO PROGRAMMING USING MATLAB

Chapter 1: Introduction to MATLAB begins by covering the MATLAB Desktop Environment. Variables, assignment statements, and types are introduced. Mathematical and relational expressions and the operators used in them are covered, as are characters, random numbers, and the use of built-in functions and the Help browser.

Chapter 2: Vectors and Matrices introduces creating and manipulating vectors and matrices. Array operations and matrix operations (such as matrix multiplication) are explained. The use of vectors and matrices as function arguments and functions that are written specifically for vectors and matrices are covered. Logical vectors and other concepts useful in vectorizing code are emphasized in this chapter.

Chapter 3: Introduction to MATLAB Programming introduces the idea of algorithms and scripts. This includes simple input and output, and commenting. Scripts are then used to create and customize simple plots and to do file input and output. Finally, the concept of a user-defined function is introduced with only the type of function that calculates and returns a single value.

Chapter 4: Selection Statements introduces the use of logical expressions in if statements, with else and elseif clauses. The switch statement is also demonstrated, as is the concept of choosing from a menu. Also, functions that return **logical true** or **false** are covered.

Chapter 5: Loop Statements and Vectorizing Code introduces the concepts of counted (**for**) and conditional (**while**) loops. Many common uses such as summing and counting are covered. Nested loops are also introduced. Some more sophisticated uses of loops such as error-checking and combining loops and selection statements are also covered. Finally, vectorizing code, by using built-in functions and operators on vectors and matrices instead of looping through them, is demonstrated. Tips for writing efficient code are emphasized, and tools for analyzing code are introduced.

The concepts in the first five chapters are assumed throughout the rest of the book.

Chapter 6: MATLAB Programs covers more on scripts and user-defined functions. User-defined functions that return more than one value and also that do not return anything are introduced. The concept of a program in MATLAB, which normally consists of a script that calls user-defined functions, is demonstrated with examples. A longer menu-driven program is shown as a reference, but could be omitted. Local functions and scope of variables are also introduced, as are some debugging techniques. The Live Editor is introduced.

The concept of a program is used throughout the rest of the book.

Chapter 7: Text Manipulation covers many built-in text manipulation functions as well as converting between string and number types. Several examples include using custom strings in plot labels and input prompts. The new string type is introduced.

Chapter 8: Data Structures: Cell Arrays and Structures introduces two main data structures: cell arrays and structures. Once structures are covered, more complicated data structures such as nested structures and vectors of structures are also introduced. Cell arrays are used in several applications in later chapters, such as file input in **Chapter 9**, variable number of function arguments in **Chapter 10**, and plot labels in **Chapter 12**, and are therefore considered important and are covered first. The section on structures can be omitted, although the use of structure variables to store object properties is shown in **Chapter 11**. Other data structures such as categorical arrays and tables are also introduced. Methods of sorting are described. Finally, the concept of indexing into a vector is introduced. Sorting a vector of structures and indexing into a vector of structures are described, but these sections can be omitted.

PART II: ADVANCED TOPICS FOR PROBLEM SOLVING WITH MATLAB

Chapter 9: Data Transfer covers lower-level file input/output statements that require opening and closing the file. Functions that can read the entire file at once as well as those that require reading one line at a time are introduced and examples that demonstrate the differences in their use are shown. Additionally, techniques for reading from and writing to spreadsheet files and also .mat files that store MATLAB variables are introduced. Cell arrays and text functions are used extensively in this chapter. Reading data from websites is also introduced.

Chapter 10: Advanced Functions covers more advanced features of and types of functions, such as anonymous functions, nested functions, and recursive functions. Function handles and their use both with anonymous functions and function functions are introduced. The concept of having a variable number of input and/or output arguments to a function is introduced; this is implemented using

cell arrays. String functions are also used in several examples in this chapter. The section on recursive functions is at the end and may be omitted.

Chapter 11: Introduction to Object-Oriented Programming and Graphics As of version R2014b, all plot objects are actual objects. This chapter introduces Object-Oriented Programming (OOP) concepts and terminology using plot objects and then expands to how to write your own class definitions and create your own objects.

Chapter 12: Advanced Plotting Techniques continues with more on the plot functions introduced in [Chapter 3](#). Different two-dimensional plot types, such as logarithmic scale plots, pie charts, and histograms are introduced, as is customizing plots using cell arrays and string functions. Three-dimensional plot functions as well as some functions that create the coordinates for specified objects are demonstrated. The notion of Graphics is covered, and some graphics properties such as line width and color are introduced. Core graphics objects and their use by higher-level plotting functions are demonstrated. Applications that involve reading data from files and then plotting use both cell arrays and string functions.

Chapter 13: Sights and Sounds briefly discusses sound files and introduces image processing. An introduction to programming Graphical User Interfaces (GUIs) is also given, including the creation of a button group and embedding images in a GUI. Nested functions are used in the GUI examples. The App Designer is introduced; it creates OOP code and builds on the concepts from [Chapter 11](#).

Chapter 14: Advanced Mathematics covers seven basic topics: it starts with some of the built-in statistical and set operations in MATLAB, then curve fitting, complex numbers, solving systems of linear algebraic equations, and integration and differentiation in calculus. Finally, some of the symbolic math toolbox functions are shown, including those that solve equations. This method returns a structure as a result.

PATH THROUGH THE BOOK

It has come to my attention that not all courses that use this text use all sections. In particular, not everyone gets to images and apps, which are the cool applications! I have reorganized some of the chapters and sections to make it easier to get to the fun, motivating applications including images and App Designer. What follows is a path through the book to get there, including which sections can be skipped.

Chapter 1: the last three [Sections 1.7, 1.8, and 1.9](#) can be skipped (although 1.9 is recommended)

Chapter 2: Section 2.5 on matrix multiplication can be skipped

Chapters 3 and 4: are both fundamental

Chapter 5: the last section on Timing can be skipped

Chapter 6: the last two sections can be skipped

Chapter 7: the last section can be skipped

Chapter 8: cell arrays and structures are important, but the last 3 sections can be skipped

Chapter 9: this can be skipped entirely, although there are very cool applications in 9.4

Chapter 10: Variable number of arguments, nested functions, and anonymous functions are all used in App Designer, but the last two sections can be skipped

Chapter 11: the first two sections are fundamental, but the last can be skipped

Chapter 12: this can be skipped entirely

Chapter 13: most sections are independent, although the concept of callback functions is explained in the GUI section and then used in the App Designer section

Chapter 14: all sections can be skipped

PEDAGOGICAL FEATURES

There are several pedagogical tools that are used throughout this book that are intended to make it easier to learn the material.

First, the book takes a conversational tone with sections called “*Quick Question!*”. These are designed to stimulate thought about the material that has just been covered. The question is posed, and then the answer is given. It will be most beneficial to the reader to try to think about the question before reading the answer! In any case, they should not be skipped over, as the answers often contain very useful information.

“*Practice*”- problems are given throughout the chapters. These are very simple problems that drill the material just covered.

“*Explore Other Interesting Features*” This book is not intended to be a complete reference book and cannot possibly cover all of the built-in functions and tools available in MATLAB; however, in every chapter, there will be a list of functions and/or commands that are related to the chapter topics, which readers may wish to investigate.

When some problems are introduced, they are solved both using “*The Programming Concept*” and also “*The Efficient Method*.” This facilitates understanding the built-in functions and operators in MATLAB as well as the underlying programming concepts. “*The Efficient Method*” highlights methods that will save time for the programmer, and in many cases, are also faster to execute in MATLAB.

Additionally, to aid the reader:

- Identifier names are shown in *italic*
- MATLAB function names are shown in **bold**
- Reserved words are shown in **bold and underlined**
- Key important terms are shown in ***bold and italic***

The end of chapter “**Summary**” contains, where applicable, several sections:

- **Common Pitfalls:** a list of common mistakes that are made and how to avoid them
- **Programming Style Guidelines:** In order to encourage “good” programs that others can actually understand, the programming chapters will have guidelines that will make programs easier to read and understand and therefore easier to work with and modify.
- **Key Terms:** a list of the key terms covered in the chapter, in sequence.
- **MATLAB Reserved Words:** a list of the reserved keywords in MATLAB. Throughout the text, these are shown in bold, underlined type.
- **MATLAB Functions and Commands:** a list of the MATLAB built-in functions and commands covered in the chapter, in the order covered. Throughout the text, these are shown in bold type.
- **MATLAB Operators:** a list of the MATLAB operators covered in the chapter, in the order covered.
- **Exercises:** a comprehensive set of exercises, ranging from the rote to more engaging applications.

ADDITIONAL BOOK RESOURCES

A companion website with additional teaching resources is available for faculty using this book as a text for their course(s). Please visit <https://www.elsevier.com/books/matlab/attaway/978-0-12-815479-3> to register for access to:

- Instructor solutions manual for end of chapter problems
- Instructor solutions manual for “Practice” problems
- Electronic figures from the text for creation of lecture slides
- Downloadable code files for all examples in the text

Other book-related resources will also be posted there from time to time.