



# MATLAB<sup>®</sup>

A PRACTICAL INTRODUCTION TO PROGRAMMING  
AND PROBLEM SOLVING

SECOND EDITION

MATLAB<sup>®</sup>  
*examples*

STORMY ATTAWAY



MATLAB<sup>®</sup>

# A Practical Introduction to Programming and Problem Solving

Second Edition

Stormy Attaway

*Department of Mechanical Engineering, Boston University*



AMSTERDAM • BOSTON • HEIDELBERG • LONDON  
NEW YORK • OXFORD • PARIS • SAN DIEGO  
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO  
Butterworth-Heinemann is an imprint of Elsevier



# Copyright

---

Butterworth-Heinemann is an imprint of Elsevier

225 Wyman Street, Waltham, MA 02451, USA

The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, UK

© 2012 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the Publisher. Details on how to seek permission, further information about the Publisher's permissions policies, and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our web site: [www.elsevier.com/permissions](http://www.elsevier.com/permissions).

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

## Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary. Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility. To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

MATLAB<sup>®</sup> is a trademark of TheMathWorks, Inc., and is used with permission. TheMathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB<sup>®</sup> software or related products does not constitute endorsement or sponsorship by TheMathWorks of a particular pedagogical approach or particular use of the MATLAB<sup>®</sup> software.

MATLAB<sup>®</sup> and Handle Graphics<sup>®</sup> are registered trademarks of TheMathWorks, Inc.

### Library of Congress Cataloging-in-Publication Data

Attaway, Stormy.

MATLAB<sup>®</sup>: a practical introduction to programming and problem solving / Stormy

Attaway. — 2nd ed.

p. cm.

Includes index.

---

ISBN 978-0-12-385081-2

1. Numerical analysis—Data processing. 2. MATLAB. 3. Computer programming.

I. Title.

QA297.A87 2011

518.028553—dc22 2011015032

### **British Library Cataloguing-in-Publication Data**

A catalogue record for this book is available from the British Library.

For information on all Butterworth–Heinemann publications

visit our Web site at [www.elsevierdirect.com](http://www.elsevierdirect.com)

Printed in the United States

11 12 13 14 15 10 9 8 7 6 5 4 3 2 1

Working together to grow  
libraries in developing countries

[www.elsevier.com](http://www.elsevier.com) | [www.bookaid.org](http://www.bookaid.org) | [www.sabre.org](http://www.sabre.org)

**ELSEVIER** **BOOK AID** **Sabre Foundation**  
International

---

## Dedication

*This book is dedicated to my husband, Ted de Winter.*

---

# Preface

## Motivation

The purpose of this book is to teach fundamentals of programming concepts and skills needed for basic problem solving, all using MATLAB<sup>®</sup> 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.

The many books that introduce MATLAB come in two basic flavors: 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.

This book instead takes a hybrid approach, introducing both the programming and 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 or whether a software package such as MATLAB will suffice for their careers. Therefore, the best approach for beginning students 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 to teaching programming and problem solving.

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

## Modifications in Second Edition

---

Changes in the second edition of this book include:

Vectorized code has been made into a separate chapter to emphasize the importance of using MATLAB efficiently.

There are expanded examples on:

- Low-level file input functions
- Plots
- Graphical user interfaces
- Vectorized code, including functions **diff**, **meshgrid**, **tic**, and **toc**

Use of MATLAB version R2011a

Concepts used in image processing, such as three-dimensional matrices and unsigned integers, are now introduced early, in [Chapter 1](#).

Modified and new end-of-chapter exercises.

The introduction to Handle Graphics was moved to [Chapter 11](#), Advanced Plotting Techniques.

Discussion of symbolic mathematics was moved to [Chapter 15](#), Advanced Mathematics.

Improved labeling of plots.

Improved standards for variable names and documentation.

Added **end** to the end of all functions.

## 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. 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-

functions or features early on with a note that says “we’ll do this later.” This does not happen in this edition; all 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 will be covered here.

## **File Input/Output**

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 selected lower-level file input/output functions as well. Since 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 2](#), then lower-level functions in [Chapter 9](#), and finally sound and images in [Chapter 14](#).

## **User-Defined Functions**

User-defined functions are a very important programming concept, and yet many times the nuances and differences among 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, functions that calculate and return one single value—arguably the easiest type of functions to understand—are demonstrated in [Chapter 2](#). Later, functions that return no values and functions that return multiple values are introduced in [Chapter 9](#). Finally, advanced function features are shown in [Chapter 10](#).

## **Advanced Programming Concepts**

In addition to the basics, some advanced programming concepts, such as string manipulation, data structures (e.g., structures and cell arrays), recursion, anonymous functions, and variable number of arguments to functions, are covered. Sorting, searching, 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 basic math will be explained and 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 in [Chapter 2](#) so that plot examples can be used throughout. [Chapter 11](#) then shows more plot types, and demonstrates customizing plots and how the graphics properties are handled in MATLAB. This chapter makes use of strings 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 14](#).

## **Vectorized Code**

Efficient uses of the capabilities of the built-in operators and functions in MATLAB are demonstrated throughout the book. However, to emphasize the importance of using MATLAB efficiently, vectorized code is treated in a separate chapter. Techniques, such as preallocating vectors and using logical vectors, are featured, as well as methods of determining how efficient the code is.

## **Layout of Text**

The book consists of 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 mathematical 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 I, Introduction to Programming Using MATLAB, includes the following chapters:

**Chapter 1: Introduction to MATLAB** covers expressions, operators, characters, variables, and assignment statements. Scalars, vectors, and matrices are all introduced as are many built-in functions that manipulate them.

**Chapter 2: 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 an answer and returns a single value.

**Chapter 3: Selection Statements** introduces relational expressions and their use in selection 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 introduced.

**Chapter 4: Loop Statements** introduces the concepts of counted (**for**) and conditional loops (**while**). 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.

**Chapter 5: Vectorized Code** introduces the idea of “vectorizing” code, which essentially means rewriting code that uses loops to more efficiently make use of built-in functions, and the fact that operations can be done on vectors and matrices in MATLAB. Functions that are useful in vectorizing code are emphasized in this chapter. Functions that time the speed of code are also introduced.

Knowledge of the concepts presented in the first five chapters is 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 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. Subfunctions and scope of variables are also introduced, as are some debugging techniques.

This program concept is used throughout the rest of the book.

**Chapter 7: String Manipulation** covers many built-in string manipulation functions as well

as converting between string and number types. Several examples include using custom strings in plot labels and input prompts.

**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](#), a variable number of function arguments in [Chapter 10](#), and plot labels in [Chapter 11](#), and are therefore considered important and are covered first. The rest of the chapter on structures can be omitted.

**Chapter 9: Advanced File Input and Output** 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, reading from and writing to spreadsheet files and also *.mat* files that store MATLAB variables are introduced. Cell arrays and string functions are used extensively in this chapter.

**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.

Part II, Advanced Topics for Problem Solving with MATLAB, contains the following chapters:

**Chapter 11: Advanced Plotting Techniques** continues with more on the plot functions introduced in [Chapter 2](#). Various two-dimensional plot types, such as pie charts and histograms, are introduced, as is customizing plots using cell arrays and string functions. Three-dimensional plot functions as well as selected functions that create the coordinates for specified objects are demonstrated. The notion of Handle Graphics<sup>®</sup> is covered, and selected graphics properties, such as line width and color, are introduced. Applications that involve reading data from files and then plotting use both cell arrays and string functions.

**Chapter 12: Matrix Representation of Linear Algebraic Equations** introduces a basic method that can be used in MATLAB to solve systems of equations using a matrix representation. First, matrix and vector operations and matrix definitions are described. This section can be covered at any point after [Chapter 5](#). Then, matrix solutions using the

Gauss-Jordan and Gauss-Jordan elimination methods are described. This section includes mathematical techniques and also the MATLAB functions that implement them.

**Chapter 13: Basic Statistics, Sets, Sorting, and Indexing** starts with some of the built-in statistical and set operations in MATLAB. Since some of these require a sorted data set, methods of sorting are described. Finally, the concepts of indexing into a vector and searching a vector are introduced. Sorting a vector of structures and indexing into a vector of structures are described, but these sections can be omitted. A recursive binary search function is in the end and may be omitted.

**Chapter 14: Sights and Sounds** briefly discusses sound files and introduces image processing. An introduction to programming graphical user interfaces is also given, including the creation of a button group. Nested functions are used in the GUI examples. The `patch` function example uses a structure.

**Chapter 15: Advanced Mathematics** covers three basic topics: curve fitting, complex numbers, 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.

## Pedagogical Features

There are several pedagogical tools that are used throughout this book that are intended to make it easier to learn the material. A list of **Key Terms** covered in each chapter, in sequence, is on the first page.

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. A 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, these sections 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 serve as drills of the material just covered.

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

Additionally, to aid the reader:

Identifier names (variables and user-defined functions) are shown in *italics* (as are filenames and file extensions).

MATLAB function names are shown in **bold**.

Reserved words are shown in **bold and underline**.

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:** To encourage the creation of “good” programs that other people can actually understand, the programming chapters have guidelines that make programs easier to read and understand, and therefore easier to work with and to modify.

**MATLAB Reserved Words:** A list of the reserved key words in MATLAB. Throughout the text, these are shown in bold, underlined type.

**MATLAB Functions and Commands:** A boxed 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 boxed 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 web site is available with downloadable *.m* files for all examples in the text, [www.elsevierdirect.com/9780123850812](http://www.elsevierdirect.com/9780123850812). Other book-related resources will also be posted on the web site from time to time.

Additional teaching resources are available for faculty using this book as a text for their course(s). Please visit [www.textbooks.elsevier.com](http://www.textbooks.elsevier.com) to register for access to:

Instructor solutions manual for end-of-chapter problems

Electronic figures from the text for creation of lecture slides

Downloadable M-files for all examples in the text

---

## Acknowledgments

I am indebted to many, many family members, colleagues, mentors, and to numerous students.

Throughout the last 24 years of coordinating and teaching the basic computation courses for the College of Engineering at Boston University, I have been blessed with many fabulous students as well as graduate teaching fellows and undergraduate teaching assistants. There have been hundreds of teaching assistants over the years, too many to name individually, but I thank them all for their support.

In particular, the following teaching assistants were very helpful in reviewing drafts of the original manuscript and suggesting examples: Edy Tan, Megan Smith, Brandon Phillips, Carl Sherwood, Ashmita Randhawa, Mike Green, Kevin Ryan, and Brian Hsu. For this Second Edition, Brian Hsu and Paul Vermilion suggested several revisions. Brian Hsu, Jakob Herrmann, and Ben Duong contributed exercises. Kevin Ryan created the script to produce the cover illustrations.

A number of colleagues have been very encouraging through the years. I would especially like to thank my former and current department chairmen, Tom Bifano and Ron Roy, for their support and motivation, and Tom for his GUI example suggestions. I am also indebted to my mentors at Boston University, Bill Henneman of the Computer Science Department, and Merrill Ebner of the Department of Manufacturing Engineering, as well as to Bob Cannon from the University of South Carolina.

I would like to thank all the reviewers of the proposal and drafts of this book. Their comments have been extremely helpful and I hope I have incorporated their suggestions to their satisfaction. In addition to several anonymous reviewers, the reviewers for this edition include:

Peter Bernard, University of Maryland  
Sanjukta Bhowmick, Pennsylvania State University  
Chris Brown, University of Rochester  
Steven Brown, University of Delaware  
Anthony Muscat, University of Arizona  
Charles Riedesel, University of Nebraska, Lincoln

Jeff Ringenberg, The University of Michigan

---

Richard Ulrich, University of Arkansas

Also, I thank those at Elsevier who helped to make this book possible, including Joseph Hayton, Publisher; Fiona Geraghty, Editorial Project Manager; Marilyn Rash, Project Manager; Eric DeCicco, Cover Designer/Illustrator; and Tim Pitts, a Publisher at Elsevier in the United Kingdom.

Finally, thanks go to all members of my family, especially my parents Roy Attaway and Jan Conklin, both of whom encouraged me at an early age to read and to write. Thanks also to my husband Ted de Winter for his encouragement and good-natured taking care of the weekend chores while I worked on this project!

The photo of Ted fishing in the image-processing section was taken by Wes Karger.

---

# Table of Contents

[Instructions for online access](#)

[Cover Image](#)

[Title](#)

[Copyright](#)

[Dedication](#)

[Preface](#)

[Acknowledgments](#)

**[PART 1. Introduction to Programming Using MATLAB](#)**

**[Chapter 1. Introduction to MATLAB](#)**

**[Contents](#)**

**[1.1 Getting into MATLAB](#)**

**[1.2 The MATLAB Desktop Environment](#)**

**[1.3 Variables and Assignment Statements](#)**

**[1.4 Expressions](#)**

**[1.5 Characters and Encoding](#)**

**[1.6 Vectors and Matrices](#)**

**[Summary](#)**

**[Chapter 2. Introduction to MATLAB Programming](#)**

**[Contents](#)**

**[2.1 Algorithms](#)**

**[2.2 Matlab scripts](#)**

**[2.3 Input and Output](#)**

**[2.4 Scripts with Input and Output](#)**

**[2.5 Scripts to Produce and Customize Simple Plots](#)**



## **2.6 Introduction to File Input/Output (Load and Save)**

---

## **2.7 User-Defined Functions That Return a Single Value**

### **Summary**

## **Chapter 3. Selection Statements**

### **Contents**

#### **3.1 Relational Expressions**

#### **3.2 The if Statement**

#### **3.3 The If-else Statement**

#### **3.4 Nested if-Else Statements**

#### **3.5 The switch statement**

#### **3.6 The menu function**

#### **3.7 The “is” functions in MATLAB**

### **Summary**

## **Chapter 4. Loop Statements**

### **Contents**

#### **4.1 The for loop**

#### **4.2 Nested For Loops**

#### **4.3 While Loops**

### **Summary**

## **Chapter 5. Vectorized Code**

### **Contents**

#### **5.1 Loops with Vectors and Matrices**

#### **5.2 Operations on Vectors and Matrices**

#### **5.3 Vectors and Matrices as Function Arguments**

#### **5.4 Logical Vectors**

#### **5.5 Vectorizing Code**

#### **5.6 Timing**

## **Summary**

---

### **Chapter 6. MATLAB Programs**

#### **Contents**

#### **6.1 More Types of User-Defined Functions**

#### **6.2 MATLAB Program Organization**

#### **6.3 Application: Menu-Driven Modular Program**

#### **6.4 Variable Scope**

#### **6.5 Debugging Techniques**

#### **Summary**

### **Chapter 7. String Manipulation**

#### **Contents**

#### **7.1 Creating String Variables**

#### **7.2 Operations on Strings**

#### **7.3 The “is” functions for strings**

#### **7.4 Converting Between String and Number Types**

#### **Summary**

### **Chapter 8. Data Structures**

#### **Contents**

#### **8.1 Cell Arrays**

#### **8.2 Structures**

#### **Summary**

### **Chapter 9. Advanced File Input and Output**

#### **Contents**

#### **9.1 Lower-level File I/O Functions**

#### **9.2 Writing and reading spreadsheet files**

#### **9.3 Using MAT-files for Variables**

#### **Summary**

## **Chapter 10. Advanced Functions**

---

### **Contents**

#### **10.1 Anonymous Functions**

#### **10.2 Uses of Function Handles**

#### **10.3 Variable Numbers of Arguments**

#### **10.4 Nested Functions**

#### **10.5 Recursive Functions**

### **Summary**

## **PART 2. Advanced Topics for Problem Solving with MATLAB**

### **Chapter 11. Advanced Plotting Techniques**

#### **Contents**

#### **11.1 Plot Functions**

#### **11.2 Animation**

#### **11.3 Three-Dimensional Plots**

#### **11.4 Customizing Plots**

#### **11.5 Handle Graphics and Plot Properties**

#### **11.6 Plot Applications**

### **Summary**

### **Chapter 12. Matrix Representation of Linear Algebraic Equations**

#### **Contents**

#### **12.1 Matrix Definitions**

#### **12.2 Matrix Solutions to Systems of Linear Algebraic Equations**

### **Summary**

### **Chapter 13. Basic Statistics, Sets, Sorting, and Indexing**

#### **Contents**

#### **13.1 Statistical Functions**

#### **13.2 Set Operations**

## **13.3 Sorting**

---

### **13.4 Index Vectors**

### **13.5 Searching**

#### **Summary**

## **Chapter 14. Sights and Sounds**

### **Contents**

#### **14.1 Sound Files**

#### **14.2 Image Processing**

#### **14.3 Introduction to Graphical User Interfaces**

#### **Summary**

## **Chapter 15. Advanced Mathematics**

### **Contents**

#### **15.1 Fitting Curves to Data**

#### **15.2 Complex Numbers**

#### **15.3 Symbolic Mathematics**

#### **15.4 Calculus: Integration and Differentiation**

#### **Summary**

## **Index**

---

# PART 1

## Introduction to Programming Using MATLAB

Chapter 1 Introduction to MATLAB

Chapter 2 Introduction to MATLAB Programming

Chapter 3 Selection Statements

Chapter 4 Loop Statements

Chapter 5 Vectorized Code

Chapter 6 MATLAB Programs

Chapter 7 String Manipulation

Chapter 8 Data Structures

Chapter 9 Advanced File Input and Output

Chapter 10 Advanced Functions

# Chapter 1

---

# Introduction to MATLAB

## Contents

- 1.1 [Getting into MATLAB](#)
- 1.2 [The MATLAB Desktop Environment](#)
- 1.3 [Variables and Assignment Statements](#)
- 1.4 [Expressions](#)
- 1.5 [Characters and Encoding](#)
- 1.6 [Vectors and Matrices](#)

## Key Terms

prompt  
programs  
script files  
variables  
assignment statement  
assignment operator  
user  
initializing  
incrementing  
decrementing  
identifier names  
reserved words  
key words  
mnemonic  
default  
unary  
operand  
binary  
scientific notation  
exponential notation  
precedence

associativity

nesting

call a function

arguments

returning values

constants

types

classes

double precision

floating point

unsigned

characters

strings

type casting

saturation arithmetic

random numbers

seed

pseudorandom

character encoding

character set

vectors

matrices

row vector

column vector

scalar

elements

array

array operations

iterate

step value

concatenating

index

subscript

index vector

transposing

subscripted indexing

unwinding a matrix

linear indexing

vector of variables

empty vector

deleting elements

three-dimensional matrices

MATLAB<sup>®</sup> is a very powerful software package that has many built-in tools for solving problems and developing graphical illustrations. The simplest method for using the MATLAB product is interactively; an expression is entered by the user and MATLAB immediately responds with a result. It is also possible to write scripts and programs in MATLAB, which are essentially groups of commands that are executed sequentially.

This chapter will focus on the basics, including many operators and built-in functions that can be used in interactive expressions. Means of storing values, including vectors and matrices, will also be introduced.

## 1.1 Getting into MATLAB

MATLAB is a mathematical and graphical software package with numerical, graphical, and programming capabilities. It has built-in functions to perform many operations, and there are toolboxes that can be added to augment these functions (e.g., for signal processing). There are versions available for different hardware platforms, in both professional and student editions.

When the MATLAB software is started, a window opens in which the main part is the Command Window (see [Figure 1.1](#)). In the Command Window, you should see:



- [Handbook of Damage Mechanics: Nano to Macro Scale for Materials and Structures.pdf](#)
- [read BBC Sky At Night \[UK\] \(April 2014\).pdf, azw \(kindle\), epub, doc, mobi](#)
- [read The Paul Goodman Reader.pdf](#)
- [click Fallen Prince.pdf](#)
- **[read The Cardinal Divide \(Cole Blackwater, Book 1\) for free](#)**
- **[The Woman of Andros: & the Ides of March.pdf, azw \(kindle\)](#)**
  
- <http://toko-gumilar.com/books/A-Color-of-His-Own.pdf>
- <http://kamallubana.com/?library/Loin-de-Rueil.pdf>
- <http://interactmg.com/ebooks/Law-on-the-Screen--The-Amherst-Series-in-Law--Jurisprudence--and-Social-Thought-.pdf>
- <http://crackingscience.org/?library/It-Will-Be-Exhilarating.pdf>
- <http://www.gateaerospaceforum.com/?library/Market-Led-Strategic-Change--Transforming-the-Process-of-Going-to-Market--4th-Edition-.pdf>
- <http://www.uverp.it/library/The-Woman-of-Andros----the-Ides-of-March.pdf>