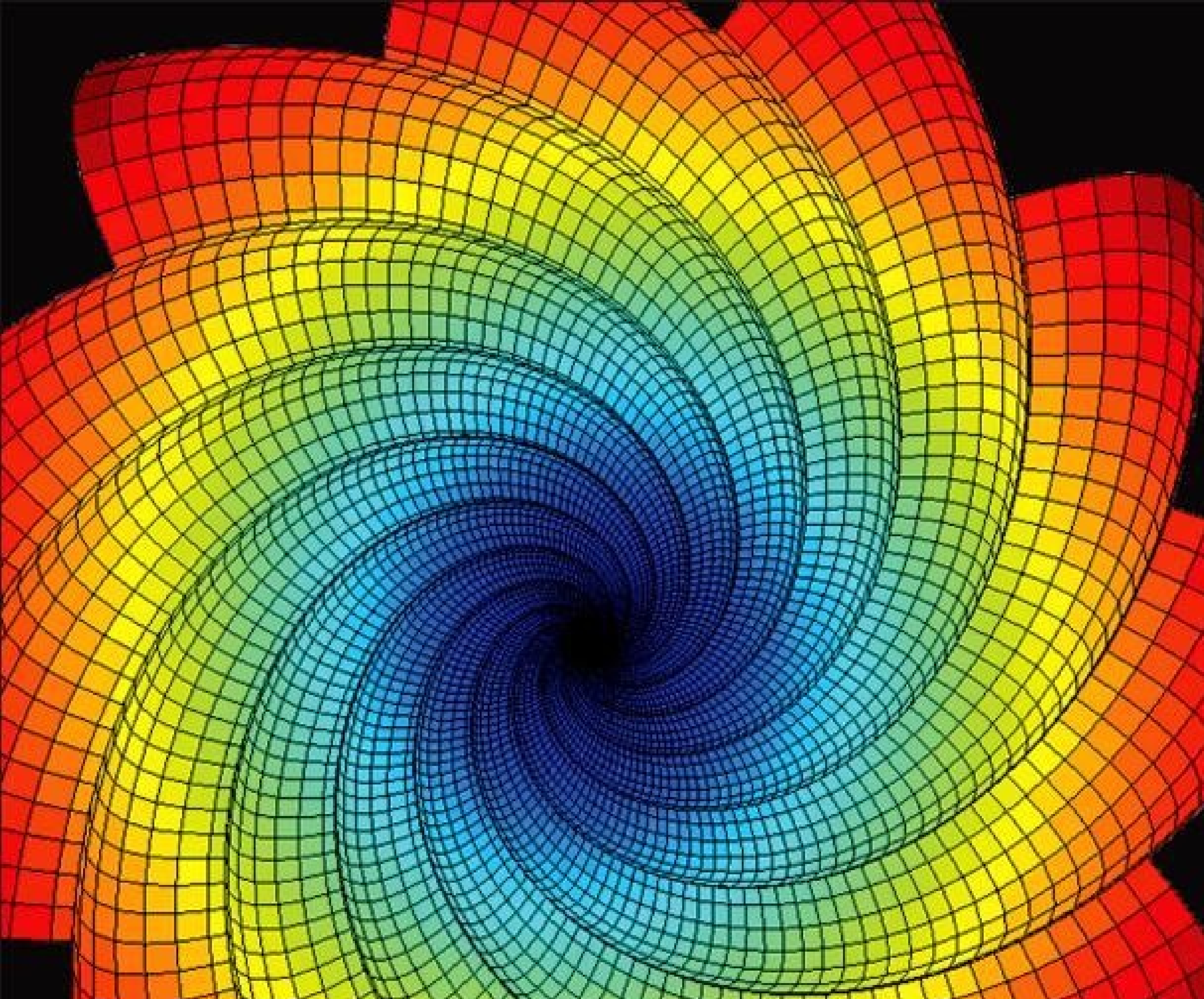

MATLAB®

A PRACTICAL INTRODUCTION TO PROGRAMMING AND PROBLEM SOLVING

SECOND EDITION

MATLAB® examples

STORMY ATTAWAY

B H

MATLAB®
A Practical Introduction to Programming and Problem Solving

Second Edition

Stormy Attaway

Department of Mechanical Engineering, Boston University

ELSEVIER

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

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.

Library of Congress Cataloging-in-Publication Data

Attaway, Stormy.

MATLAB®: 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*

Printed in the United States

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

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 neede for basic problem solving, all using MATLAB® as the vehicle. MATLAB is a powerful softwa package that has built-in functions to accomplish a diverse range of tasks, from mathematic operations to three-dimensional imaging. Additionally, MATLAB has a complete set 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 demonstra the use of the built-in functions in MATLAB, with a chapter or two on some programmin concepts; and those that cover only the programming constructs without mentioning many the built-in functions that make MATLAB efficient to use. Someone who learns just the buil in functions will be well-prepared to use MATLAB, but would not understand bas 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 th programming concepts. Conversely, anyone who learns only programming concepts fir (using any language) would tend to write highly inefficient code using control statements 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 efficie uses. The challenge for students is that it is nearly impossible to predict whether they will fact need to know programming concepts later or whether a software package such MATLAB will suffice for their careers. Therefore, the best approach for beginning students to give them both: the programming concepts and the efficient built-in functions. Sinc MATLAB is very easy to use, it is a perfect platform for this approach to teachin programming and problem solving.

Since programming concepts are critically important to this book, emphasis is not placed the time-saving features that evolve with every new MATLAB release. For example, current versions of MATLAB, statistics on variables are available readily in the Workspa Window. This is not shown in any detail in the book, since whether this feature is availab depends on the software version, and because of the desire to explain the concepts in th 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 usi 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 unsigne 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 Plottin 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 concep and the use of the built-in functions in MATLAB side by side. It starts with basic programmi concepts such as variables, assignments, input/output, selection, and loop statements. The throughout the rest of the book, many times a problem will be introduced and then solve using the “programming concept” and also using the “efficient method.” This will not be don 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, buildi on concepts throughout the book. It is very tempting in a MATLAB text to show builtfunctions or features early on with a note that says “we'll do this later.” This does not happe in this edition; all functions are covered before they are used in examples. Additionally, bas programming concepts will be explained carefully and systematically. Very basic concept such as looping to calculate a sum, counting in a conditional loop, and error-checking, are n found in many texts but will be covered here.

File Input/Output

Many applications in engineering and the sciences involve manipulating large data sets th are stored in external files. Most MATLAB texts at least mention the **save** and **load** function and in some cases selected lower-level file input/output functions as well. Since file input ar output is so fundamental to so many applications, this book will cover several low-level fi input/output functions, as well as reading from and writing to spreadsheet files. Late chapters will also deal with audio and image files. These file input/output concepts a introduced gradually: first **load** and **save** in Chapter 2, then lower-level functions in Chapt 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 th nuances and differences among concepts such as types of functions and function calls versu function headers can be very confusing to beginning programmers. Therefore, these concep are introduced gradually. First, functions that calculate and return one single value—arguab the easiest type of functions to understand—are demonstrated in Chapter 2. Later, functio that return no values and functions that return multiple values are introduced in Chapter Finally, advanced function features are shown in Chapter 10.

Advanced Programming Concepts

In addition to the basics, some advanced programming concepts, such as string manipulatio data structures (e.g., structures and cell arrays), recursion, anonymous functions, and variab number of arguments to functions, are covered. Sorting, searching, and indexing are als addressed. All of these are again approached systematically; for example, cell arrays a 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 man problems will be introduced. These will include statistical functions, solving sets of line algebraic equations, and fitting curves to data. The use of complex numbers and som calculus (integration and differentiation) will also be introduced. The basic math will b 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 pl examples can be used throughout. Chapter 11 then shows more plot types, and demonstrat customizing plots and how the graphics properties are handled in MATLAB. This chapt makes use of strings and cell arrays to customize labels. Also, there is an introduction image processing and the basics necessary to understand programming graphical us interfaces (GUIs) in Chapter 14.

Vectorized Code

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

Layout of Text

The book consists of two parts. The first part covers programming constructs ar demonstrates the programming method versus efficient use of built-in functions to solv problems. The second part covers tools that are used for basic problem solving, includin plotting, image processing, and mathematical techniques to solve systems of linear algebra equations, fit curves to data, and perform basic statistical analyses. The first six chapte cover the very basics in MATLAB and in programming, and are all prerequisites for the rest the book. After that, many chapters in the problem-solving section can be introduced whe desired, to produce a customized flow of topics in the book. This is true to an exten although the order of the chapters has been chosen carefully to ensure that the coverage systematic.

The individual chapters are described here, as well as which topics are required for eac chapter. Part I, Introduction to Programming Using MATLAB, includes the following chapter

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

Chapter 2: Introduction to MATLAB Programming introduces the idea of algorithms ar scripts. This includes simple input and output, and commenting. Scripts are then used create and customize simple plots, and to do file input and output. Finally, the concept of user-defined function is introduced with only the type of function that calculates ar returns a single value.

Chapter 3: Selection Statements introduces relational expressions and their use in statements, with **else** and **elseif** clauses. The **switch** statement is also demonstrated, as the concept of choosing from a menu. Also, functions that return **logical true** or **false** a introduced.

Chapter 4: Loop Statements introduces the concepts of counted (**for**) and conditional loo (**while**). Many common uses, such as summing and counting, are covered. Nested loops a also introduced. Some more sophisticated uses of loops, such as error-checking ar combining loops and selection statements, are also covered.

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

Knowledge of the concepts presented in the first five chapters is assumed throughout the re of the book.

Chapter 6: MATLAB Programs covers more on scripts and user-defined functions. Use defined functions that return more than one value and also that do not return anything a introduced. The concept of a program in MATLAB, which consists of a script that calls use defined functions, is demonstrated with examples. A longer menu-driven program is show as a reference, but could be omitted. Subfunctions and scope of variables are als 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 we as converting between string and number types. Several examples include using custo strings in plot labels and input prompts.

Chapter 8: Data Structures: Cell Arrays and Structures introduces two main da structures: cell arrays and structures. Once structures are covered, more complicated da structures, such as nested structures and vectors of structures, are also introduced. Ce arrays are used in several applications in later chapters, such as file input in Chapter variable number of function arguments in Chapter 10, and plot labels in Chapter 11, ar are therefore considered important and are covered first. The rest of the chapter o structures can be omitted.

Chapter 9: Advanced File Input and Output covers lower-level file input/outp statements that require opening and closing the file. Functions that can read the entire fi at once as well as those that require reading one line at a time are introduced, ar examples that demonstrate the differences in their use are shown. Additionally, readin from and writing to spreadsheet files and also *.mat* files that store MATLAB variables a introduced. Cell arrays and string functions are used extensively in this chapter.

Chapter 10: Advanced Functions covers more advanced features of and types of functio such as anonymous functions, nested functions, and recursive functions. Function handl and their use both with anonymous functions and function functions are introduced. Th concept of having a variable number of input and/or output arguments to a function introduced; this is implemented using cell arrays. String functions are also used in sever examples in this chapter. The section on recursive functions is at the end and may b 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 functio introduced in Chapter 2. Various two-dimensional plot types, such as pie charts ar histograms, are introduced, as is customizing plots using cell arrays and string function Three-dimensional plot functions as well as selected functions that create the coordinat for specified objects are demonstrated. The notion of Handle Graphics® is covered, ar selected graphics properties, such as line width and color, are introduced. Applications th 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 bas method that can be used in MATLAB to solve systems of equations using a matr representation. First, matrix and vector operations and matrix definitions are describe This section can be covered at any point after Chapter 5. Then, matrix solutions using th

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

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

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

Chapter 15: Advanced Mathematics covers three basic topics: curve fitting, comple numbers, and integration and differentiation in calculus. Finally, some of the Symbol Math Toolbox functions are shown, including those that solve equations. This metho returns a structure as a result.

Pedagogical Features

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

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

Practice problems are given throughout the chapters. These are very simple problems th serve as drills of the material just covered.

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

Additionally, to aid the reader:

Identifier names (variables and user-defined functions) are shown in *italics* (as are filenam 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 othe can actually understand, the programming chapters have guidelines that make progran 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 tl text, these are shown in bold, underlined type.

MATLAB Functions and Commands: A boxed list of the MATLAB built-in functions ar commands covered in the chapter, in the order covered. Throughout the text, these a shown in bold type.

MATLAB Operators: A boxed list of the MATLAB operators covered in the chapter in tl order covered.

Exercises: A comprehensive set of exercises, ranging from the rote to more engagi applications.

Additional Book Resources

A companion web site is available with downloadable *.m* files for all examples in the text, *www.elsevierdirect.com/9780123850812* . Other book-related resources will also be posted c the web site from time to time.

Additional teaching resources are available for faculty using this book as a text for the course(s). Please visit *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 numerou students.

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

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

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

I would like to thank all the reviewers of the proposal and drafts of this book. The comments have been extremely helpful and I hope I have incorporated their suggestions their satisfaction. In addition to several anonymous reviewers, the reviewers for this editio 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 Josep Hayton, Publisher; Fiona Geraghty, Editorial Project Manager; Marilyn Rash, Proje Manager; Eric DeCicco, Cover Designer/Illustrator; and Tim Pitts, a Publisher at Elsevier 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 my husband Ted de Winter for his encouragement and good-natured taking care of th 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

PART 1

Introduction to Programming Using MATLAB

Chapter 1

Introduction to MATLAB

Contents

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

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

1.1 Getting into MATLAB

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

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