

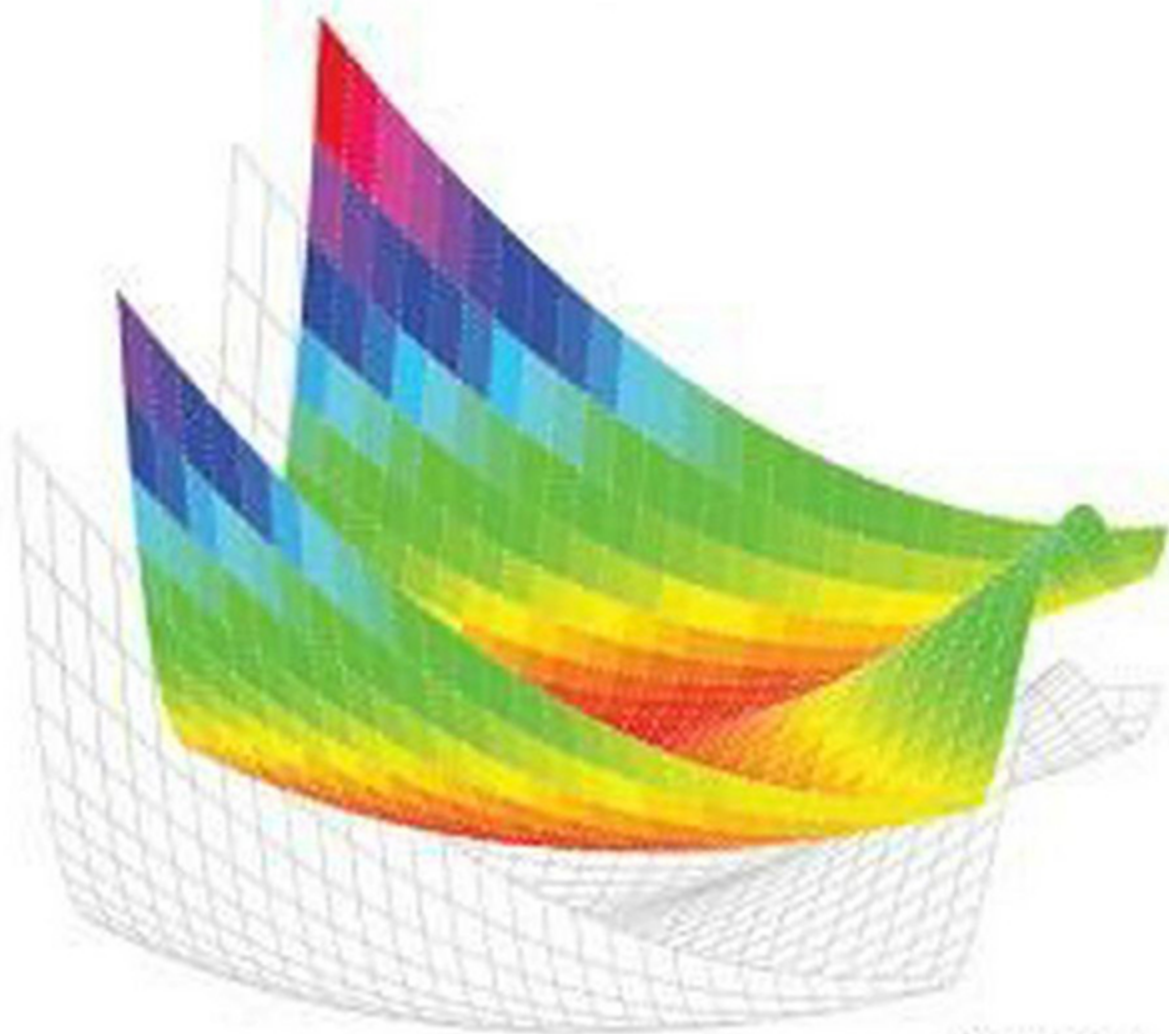
Wiley Student Choice

MATLAB

AN INTRODUCTION WITH APPLICATIONS

5TH EDITION

AMDS
SILAT



WILEY

WILEY

MATLAB

An Introduction with Applications

Sixth Edition

Amos Gilat

Department of Mechanical and Aerospace Engineering
The Ohio State University

WILEY

Copyright © 2017, 2014, 2011 John Wiley & Sons, Inc.

ISBN: 978-1-119-25683-0 (PBK)
ISBN 978-1-119-29931-8 (EVAL)

Library of Congress Cataloging-in-Publication Data:

Names: Gilat, Amos, author.

Title: MATLAB : an introduction with applications / Amos Gilat, Department of
Mechanical and Aerospace Engineering, The Ohio State University.

Description: Sixth edition. | Hoboken, NJ : John Wiley & Sons, Inc., [2017] |
Includes index.

Identifiers: LCCN 2016029050 (print) | LCCN 2016030206 (ebook) | ISBN
9781119256830 (paper) | ISBN 9781119299547 (pdf) | ISBN 9781119299257
(epub)

Subjects: LCSH: MATLAB.

Classification: LCC QA297 .G48 2017 (print) | LCC QA297 (ebook) | DDC
518.0285/53--dc23

LC record available at <https://lcn.loc.gov/2016029050>

Printed in the United States of America

Preface

MATLAB[®] is a very popular language for technical computing used by students, engineers, and scientists in universities, research institutes, and industries all over the world. The software is popular because it is powerful and easy to use. For university freshmen it can be thought of as the next tool to use after the graphic calculator in high school.

This book was written following several years of teaching the software to freshmen in an introductory engineering course. The objective was to write a book that teaches the software in a friendly, non-intimidating fashion. Therefore, the book is written in simple and direct language. In many places bullets, rather than lengthy text, are used to list facts and details that are related to a specific topic. The book includes numerous sample problems in mathematics, science, and engineering that are similar to problems encountered by new users of MATLAB.

This sixth edition of the book is updated to MATLAB Release 2016a. In addition, the end of chapter problems have been revised. In Chapters 1 through 8 close to 70% of the problems are new or different than in previous editions.

I would like to thank several of my colleagues at The Ohio State University. Professor Richard Freuler for his comments, and Dr. Mike Parke for reviewing sections of the book and suggested modifications. I also appreciate the involvement and support of Professors Robert Gustafson, John Demel and Dr. John Merrill from the Engineering Education Innovation Center at The Ohio State University. Special thanks go to Professor Mike Lichtensteiger (OSU), and my daughter Tal Gilat (Marquette University), who carefully reviewed the first edition of the book and provided valuable comments and criticisms. Professor Brian Harper (OSU) has made a significant contribution to the new end of chapter problems in the present edition.

I would like to express my appreciation to all those who have reviewed earlier editions of the text at its various stages of development, including Betty Barr, University of Houston; Andrei G. Chakhovskoi, University of California, Davis; Roger King, University of Toledo; Richard Kwor, University of Colorado at Colorado Springs; Larry Lagerstrom, University of California, Davis; Yueh-Jaw Lin, University of Akron; H. David Sheets, Canisius College; Geb Thomas, University of Iowa; Brian Vick, Virginia Polytechnic Institute and State University; Jay Weitzen, University of Massachusetts, Lowell; and Jane Patterson Fife, The Ohio State University. In addition, I would like to acknowledge Chris Nelson who supported the production of the sixth edition.

I hope that the book will be useful and will help the users of MATLAB to enjoy the software.

Amos Gilat
Columbus, Ohio
May, 2016
gilat.1@osu.edu

To my parents Schoschana and Haim Gelbwacks

Contents

Preface v

Introduction 1

Chapter 1 Starting with MATLAB 5

- 1.1 STARTING MATLAB, MATLAB WINDOWS 5
- 1.2 WORKING IN THE COMMAND WINDOW 9
- 1.3 ARITHMETIC OPERATIONS WITH SCALARS 11
 - 1.3.1 Order of Precedence 11
 - 1.3.2 Using MATLAB as a Calculator 12
- 1.4 DISPLAY FORMATS 12
- 1.5 ELEMENTARY MATH BUILT-IN FUNCTIONS 14
- 1.6 DEFINING SCALAR VARIABLES 16
 - 1.6.1 The Assignment Operator 16
 - 1.6.2 Rules About Variable Names 18
 - 1.6.3 Predefined Variables and Keywords 19
- 1.7 USEFUL COMMANDS FOR MANAGING VARIABLES 19
- 1.8 SCRIPT FILES 20
 - 1.8.1 Notes About Script Files 20
 - 1.8.2 Creating and Saving a Script File 21
 - 1.8.3 Running (Executing) a Script File 22
 - 1.8.4 Current Folder 22
- 1.9 EXAMPLES OF MATLAB APPLICATIONS 24
- 1.10 PROBLEMS 27

Chapter 2 Creating Arrays 35

- 2.1 CREATING A ONE-DIMENSIONAL ARRAY (VECTOR) 35
- 2.2 CREATING A TWO-DIMENSIONAL ARRAY (MATRIX) 39
 - 2.2.1 The zeros, ones and, eye Commands 40
- 2.3 NOTES ABOUT VARIABLES IN MATLAB 41
- 2.4 THE TRANSPOSE OPERATOR 41
- 2.5 ARRAY ADDRESSING 42
 - 2.5.1 Vector 42
 - 2.5.2 Matrix 43
- 2.6 USING A COLON : IN ADDRESSING ARRAYS 44
- 2.7 ADDING ELEMENTS TO EXISTING VARIABLES 46
- 2.8 DELETING ELEMENTS 48
- 2.9 BUILT-IN FUNCTIONS FOR HANDLING ARRAYS 49
- 2.10 STRINGS AND STRINGS AS VARIABLES 53
- 2.11 PROBLEMS 55

Chapter 3 Mathematical Operations with Arrays 63

- 3.1 ADDITION AND SUBTRACTION 64
- 3.2 ARRAY MULTIPLICATION 65
- 3.3 ARRAY DIVISION 68

3.4	ELEMENT-BY-ELEMENT OPERATIONS	72
3.5	USING ARRAYS IN MATLAB BUILT-IN MATH FUNCTIONS	75
3.6	BUILT-IN FUNCTIONS FOR ANALYZING ARRAYS	75
3.7	GENERATION OF RANDOM NUMBERS	77
3.8	EXAMPLES OF MATLAB APPLICATIONS	80
3.9	PROBLEMS	86

Chapter 4 Using Script Files and Managing Data 95

4.1	THE MATLAB WORKSPACE AND THE WORKSPACE WINDOW	96
4.2	INPUT TO A SCRIPT FILE	97
4.3	OUTPUT COMMANDS	100
4.3.1	The disp Command	101
4.3.2	The fprintf Command	103
4.4	THE save AND load COMMANDS	111
4.4.1	The save Command	111
4.4.2	The load Command	112
4.5	IMPORTING AND EXPORTING DATA	114
4.5.1	Commands for Importing and Exporting Data	114
4.5.2	Using the Import Wizard	116
4.6	EXAMPLES OF MATLAB APPLICATIONS	118
4.7	PROBLEMS	123

Chapter 5 Two-Dimensional Plots 133

5.1	THE plot COMMAND	134
5.1.1	Plot of Given Data	138
5.1.2	Plot of a Function	139
5.2	THE fplot COMMAND	140
5.3	PLOTTING MULTIPLE GRAPHS IN THE SAME PLOT	141
5.3.1	Using the plot Command	141
5.3.2	Using the hold on and hold off Commands	142
5.3.3	Using the line Command	143
5.4	FORMATTING A PLOT	144
5.4.1	Formatting a Plot Using Commands	144
5.4.2	Formatting a Plot Using the Plot Editor	148
5.5	PLOTS WITH LOGARITHMIC AXES	149
5.6	PLOTS WITH ERROR BARS	150
5.7	PLOTS WITH SPECIAL GRAPHICS	152
5.8	HISTOGRAMS	153
5.9	POLAR PLOTS	156
5.10	PUTTING MULTIPLE PLOTS ON THE SAME PAGE	157
5.11	MULTIPLE FIGURE WINDOWS	157
5.12	PLOTTING USING THE PLOTS TOOLSTRIP	159
5.13	EXAMPLES OF MATLAB APPLICATIONS	160
5.14	PROBLEMS	165

Chapter 6	Programming in MATLAB	175
6.1	RELATIONAL AND LOGICAL OPERATORS	176
6.2	CONDITIONAL STATEMENTS	184
6.2.1	The if-end Structure	184
6.2.2	The if-else-end Structure	186
6.2.3	The if-elseif-else-end Structure	187
6.3	THE switch-case STATEMENT	189
6.4	LOOPS	192
6.4.1	for-end Loops	192
6.4.2	while-end Loops	197
6.5	NESTED LOOPS AND NESTED CONDITIONAL STATEMENTS	200
6.6	THE break AND continue COMMANDS	202
6.7	EXAMPLES OF MATLAB APPLICATIONS	203
6.8	PROBLEMS	211
Chapter 7	User-Defined Functions and Function Files	221
7.1	CREATING A FUNCTION FILE	222
7.2	STRUCTURE OF A FUNCTION FILE	223
7.2.1	Function Definition Line	224
7.2.2	Input and Output Arguments	224
7.2.3	The H1 Line and Help Text Lines	226
7.2.4	Function Body	226
7.3	LOCAL AND GLOBAL VARIABLES	226
7.4	SAVING A FUNCTION FILE	227
7.5	USING A USER-DEFINED FUNCTION	228
7.6	EXAMPLES OF SIMPLE USER-DEFINED FUNCTIONS	229
7.7	COMPARISON BETWEEN SCRIPT FILES AND FUNCTION FILES	231
7.8	ANONYMOUS FUNCTIONS	231
7.9	FUNCTION FUNCTIONS	234
7.9.1	Using Function Handles for Passing a Function into a Function Function	235
7.9.2	Using a Function Name for Passing a Function into a Function Function	238
7.10	SUBFUNCTIONS	240
7.11	NESTED FUNCTIONS	242
7.12	EXAMPLES OF MATLAB APPLICATIONS	245
7.13	PROBLEMS	248
Chapter 8	Polynomials, Curve Fitting, and Interpolation	261
8.1	POLYNOMIALS	261
8.1.1	Value of a Polynomial	262
8.1.2	Roots of a Polynomial	263
8.1.3	Addition, Multiplication, and Division of Polynomials	264
8.1.4	Derivatives of Polynomials	266
8.2	CURVE FITTING	267
8.2.1	Curve Fitting with Polynomials; The polyfit Function	267
8.2.2	Curve Fitting with Functions Other than Polynomials	271

- 8.3 INTERPOLATION 274
- 8.4 THE BASIC FITTING INTERFACE 278
- 8.5 EXAMPLES OF MATLAB APPLICATIONS 281
- 8.6 PROBLEMS 286

Chapter 9 Applications in Numerical Analysis 295

- 9.1 SOLVING AN EQUATION WITH ONE VARIABLE 295
- 9.2 FINDING A MINIMUM OR A MAXIMUM OF A FUNCTION 298
- 9.3 NUMERICAL INTEGRATION 300
- 9.4 ORDINARY DIFFERENTIAL EQUATIONS 303
- 9.5 EXAMPLES OF MATLAB APPLICATIONS 307
- 9.6 PROBLEMS 313

Chapter 10 Three-Dimensional Plots 323

- 10.1 LINE PLOTS 323
- 10.2 MESH AND SURFACE PLOTS 324
- 10.3 PLOTS WITH SPECIAL GRAPHICS 331
- 10.4 THE `view` COMMAND 333
- 10.5 EXAMPLES OF MATLAB APPLICATIONS 336
- 10.6 PROBLEMS 341

Chapter 11 Symbolic Math 347

- 11.1 SYMBOLIC OBJECTS AND SYMBOLIC EXPRESSIONS 348
 - 11.1.1 Creating Symbolic Objects 348
 - 11.1.2 Creating Symbolic Expressions 350
 - 11.1.3 The `findsym` Command and the Default Symbolic Variable 353
- 11.2 CHANGING THE FORM OF AN EXISTING SYMBOLIC EXPRESSION 354
 - 11.2.1 The `collect`, `expand`, and `factor` Commands 354
 - 11.2.2 The `simplify` Command 356
 - 11.2.3 The `pretty` Command 356
- 11.3 SOLVING ALGEBRAIC EQUATIONS 357
- 11.4 DIFFERENTIATION 362
- 11.5 INTEGRATION 363
- 11.6 SOLVING AN ORDINARY DIFFERENTIAL EQUATION 365
- 11.7 PLOTTING SYMBOLIC EXPRESSIONS 368
- 11.8 NUMERICAL CALCULATIONS WITH SYMBOLIC EXPRESSIONS 371
- 11.9 EXAMPLES OF MATLAB APPLICATIONS 375
- 11.10 PROBLEMS 382

Appendix: Summary of Characters, Commands, and Functions 391

Answers to Selected Problems www.wiley.com/college/gilat

Index 399

Introduction

MATLAB is a powerful language for technical computing. The name MATLAB stands for MATrix LABoratory, because its basic data element is a matrix (array). MATLAB can be used for math computations, modeling and simulations, data analysis and processing, visualization and graphics, and algorithm development.

MATLAB is widely used in universities and colleges in introductory and advanced courses in mathematics, science, and especially engineering. In industry the software is used in research, development, and design. The standard MATLAB program has tools (functions) that can be used to solve common problems. In addition, MATLAB has optional toolboxes that are collections of specialized programs designed to solve specific types of problems. Examples include toolboxes for signal processing, symbolic calculations, and control systems.

Until recently, most of the users of MATLAB have been people with previous knowledge of programming languages such as FORTRAN and C who switched to MATLAB as the software became popular. Consequently, the majority of the literature that has been written about MATLAB assumes that the reader has knowledge of computer programming. Books about MATLAB often address advanced topics or applications that are specialized to a particular field. Today, however, MATLAB is being introduced to college students as the first (and often the only) computer program they will learn. For these students there is a need for a book that teaches MATLAB assuming no prior experience in computer programming.

The Purpose of This Book

MATLAB: An Introduction with Applications is intended for students who are using MATLAB for the first time and have little or no experience in computer programming. It can be used as a textbook in freshmen engineering courses or in workshops where MATLAB is being taught. The book can also serve as a reference in more advanced science and engineering courses where MATLAB is used as a tool for solving problems. It also can be used for self-study of MATLAB by students and practicing engineers. In addition, the book can be a supplement or a secondary book in courses where MATLAB is used but the instructor does not have the time to cover it extensively.

Topics Covered

MATLAB is a huge program, and therefore it is impossible to cover all of it in

one book. This book focuses primarily on the foundations of MATLAB. The assumption is that once these foundations are well understood, the student will be able to learn advanced topics easily by using the information in the Help menu.

The order in which the topics are presented in this book was chosen carefully, based on several years of experience in teaching MATLAB in an introductory engineering course. The topics are presented in an order that allows the student to follow the book chapter after chapter. Every topic is presented completely in one place and then used in the following chapters.

The first chapter describes the basic structure and features of MATLAB and how to use the program for simple arithmetic operations with scalars as with a calculator. Script files are introduced at the end of the chapter. They allow the student to write, save, and execute simple MATLAB programs. The next two chapters are devoted to the topic of arrays. MATLAB's basic data element is an array that does not require dimensioning. This concept, which makes MATLAB a very powerful program, can be a little difficult to grasp for students who have only limited knowledge of and experience with linear algebra and vector analysis. The concept of arrays is introduced gradually and then explained in extensive detail. Chapter 2 describes how to create arrays, and Chapter 3 covers mathematical operations with arrays.

Following the basics, more advanced topics that are related to script files and input and output of data are presented in Chapter 4. This is followed by coverage of two-dimensional plotting in Chapter 5. Programming with MATLAB is introduced in Chapter 6. This includes flow control with conditional statements and loops. User-defined functions, anonymous functions, and function functions are covered next in Chapter 7. The coverage of function files (user-defined functions) is intentionally separated from the subject of script files. This has proven to be easier to understand by students who are not familiar with similar concepts from other computer programs.

The next three chapters cover more advanced topics. Chapter 8 describes how MATLAB can be used for carrying out calculations with polynomials, and how to use MATLAB for curve fitting and interpolation. Chapter 9 covers applications of MATLAB in numerical analysis. It includes solving nonlinear equations, finding minimum or a maximum of a function, numerical integration, and solution of first-order ordinary differential equations. Chapter 10 describes how to produce three-dimensional plots, an extension of the chapter on two-dimensional plots. Chapter 11 covers in great detail how to use MATLAB in symbolic operations.

The Framework of a Typical Chapter

In every chapter the topics are introduced gradually in an order that makes the concepts easy to understand. The use of MATLAB is demonstrated extensively within the text and by examples. Some of the longer examples in Chapters 1–3 are titled as tutorials. Every use of MATLAB is printed with a different font and with a gray background. Additional explanations appear in boxed text with a

white background. The idea is that the reader will execute these demonstrations and tutorials in order to gain experience in using MATLAB. In addition, every chapter includes formal sample problems that are examples of applications of MATLAB for solving problems in math, science, and engineering. Each example includes a problem statement and a detailed solution. Some sample problems are presented in the middle of the chapter. All of the chapters (except Chapter 2) have a section at the end with several sample problems of applications. It should be pointed out that problems with MATLAB can be solved in many different ways. The solutions of the sample problems are written such that they are easy to follow. This means that in many cases the problem can be solved by writing a shorter, or sometimes “trickier,” program. The students are encouraged to try to write their own solutions and compare the end results. At the end of each chapter there is a set of homework problems. They include general problems from math and science and problems from different disciplines of engineering.

Symbolic Calculations

MATLAB is essentially a software for numerical calculations. Symbolic math operations, however, can be executed if the Symbolic Math toolbox is installed. The Symbolic Math toolbox is included in the student version of the software and can be added to the standard program.

Software and Hardware

The MATLAB program, like most other software, is continually being developed and new versions are released frequently. This book covers MATLAB Version 9.0.0.341360, Release 2016a. It should be emphasized, however, that the book covers the basics of MATLAB, which do not change much from version to version. The book covers the use of MATLAB on computers that use the Windows operating system. Everything is essentially the same when MATLAB is used on other machines. The user is referred to the documentation of MATLAB for details on using MATLAB on other operating systems. It is assumed that the software is installed on the computer, and the user has basic knowledge of operating the computer.

The Order of Topics in the Book

It is probably impossible to write a textbook where all the subjects are presented in an order that is suitable for everyone. The order of topics in this book is such that the fundamentals of MATLAB are covered first (arrays and array operations), and, as mentioned before, every topic is covered completely in one location, which makes the book easy to use as a reference. The order of the topics in this sixth edition is the same as in the previous edition. Programming is introduced before user-defined functions. This allows using programming in user-defined functions. Also, applications of MATLAB in numerical analysis follow Chapter 8 which covers polynomials, curve fitting, and interpolation.

Chapter 1

Starting with

MATLAB

This chapter begins by describing the characteristics and purpose of the different windows in MATLAB. Next, the Command Window is introduced in detail. The chapter shows how to use MATLAB for arithmetic operations with scalars in much the way that a calculator is used. This includes the use of elementary math functions with scalars. The chapter then shows how to define scalar variables (the assignment operator) and how to use these variables in arithmetic calculations. The last section in the chapter introduces script files. It shows how to write, save, and execute simple MATLAB programs.

1.1 STARTING MATLAB, MATLAB WINDOWS

It is assumed that the software is installed on the computer, and that the user can start the program. Once the program starts, the MATLAB desktop window opens with the default layout, Figure 1-1. The layout has a Toolstrip at the top, the Current Folder Toolbar below it, and four windows underneath. At the top of the Toolstrip there are three tabs: HOME, PLOTS, and APPS. Clicking on the tabs changes the icons in the Toolstrip. Commonly, MATLAB is used with the HOME tab selected. The associated icons are used for executing various commands, as explained later in this chapter. The PLOTS tab can be used to create plots, as explained in Chapter 5 (Section 5.12), and the APPS tab can be used for opening additional applications and Toolboxes of MATLAB.

The default layout

The default layout (Figure 1-1) consists of the following four windows that are displayed under the Toolstrip: the Command Window (the larger window), the Current Folder Window (on the top left), the Details Window and the Workspace Window (on the bottom left). A list of several MATLAB windows and their purposes is given in Table 1-1.

Four of the windows—the Command Window, the Figure Window, the Editor Window, and the Help Window—are used extensively throughout the book

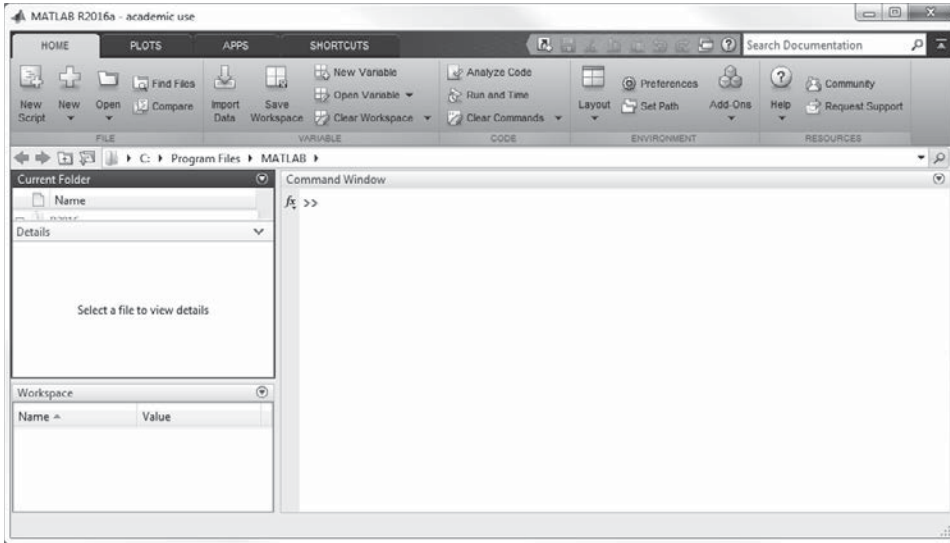


Figure 1-1: The default view of MATLAB desktop.

and are briefly described on the following pages. More detailed descriptions are included in the chapters where they are used. The Command History Window, Current Folder Window, and the Workspace Window are described in Sections 1.2, 1.8.4, and 4.1, respectively.

Command Window: The Command Window is MATLAB's main window and opens when MATLAB is started. It is convenient to have the Command Window as the only visible window. This can be done either by closing all the other windows, or by selecting **Command Window Only** in the menu that opens when the **Layout** icon on the Toolstrip is selected. To close a window, click on the pull-down menu at the top right-hand side of the window and then select Close. Working in the Command Window is described in detail in Section 1.2.

Table 1-1: MATLAB windows

Window	Purpose
Command Window	Main window, enters variables, runs programs.
Figure Window	Contains output from graphic commands.
Editor Window	Creates and debugs script and function files.
Help Window	Provides help information.
Command History Window	Logs commands entered in the Command Window.

Table 1-1: MATLAB windows

Window	Purpose
Workspace Window	Provides information about the variables that are stored.
Current Folder Window	Shows the files in the current folder.

Figure Window: The Figure Window opens automatically when graphics commands are executed, and contains graphs created by these commands. An example of a Figure Window is shown in Figure 1-2. A more detailed description of this window is given in Chapter 5.

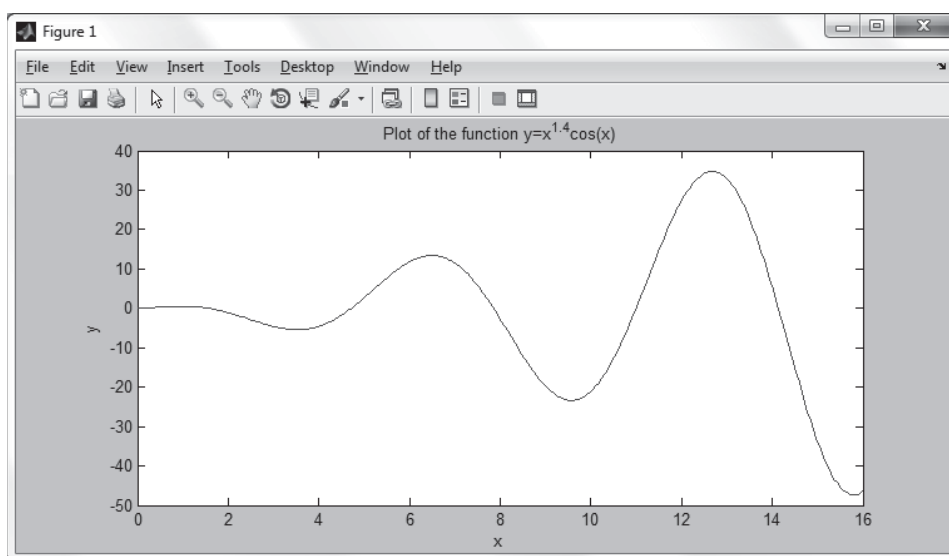


Figure 1-2: Example of a Figure Window.

Editor Window: The Editor Window is used for writing and editing programs. This window is opened by clicking on the **New Script** icon in the Toolstrip, or by clicking on the **New** icon and then selecting **Script** from the menu that opens. An example of an Editor Window is shown in Figure 1-3. More details on the Editor Window are given in Section 1.8.2, where it is used for writing script files, and in Chapter 7, where it is used to write function files.

Help Window: The Help Window contains help information. This window can be opened from the **Help** icon in the Toolstrip of the Command Window or the toolbar of any MATLAB window. The Help Window is interactive and can be used to obtain information on any feature of MATLAB. Figure 1-4 shows an open Help Window.

When MATLAB is started for the first time, the screen looks like that shown in Figure 1-1. For most beginners it is probably more convenient to close

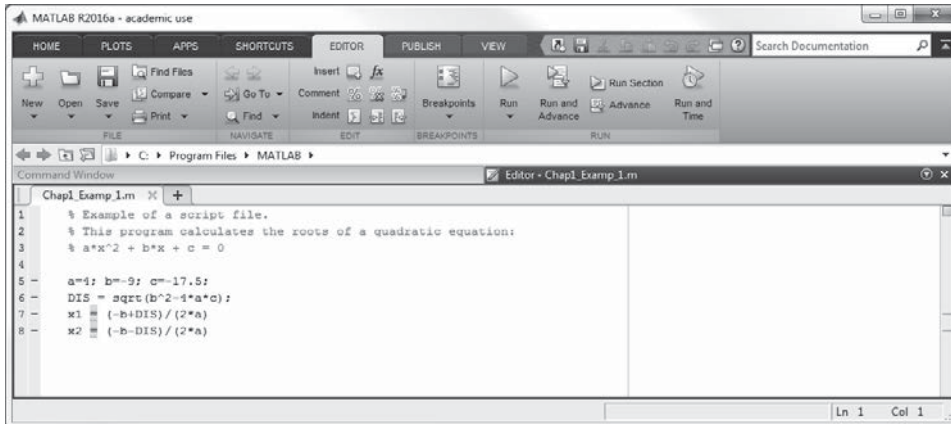


Figure 1-3: Example of an Editor Window.

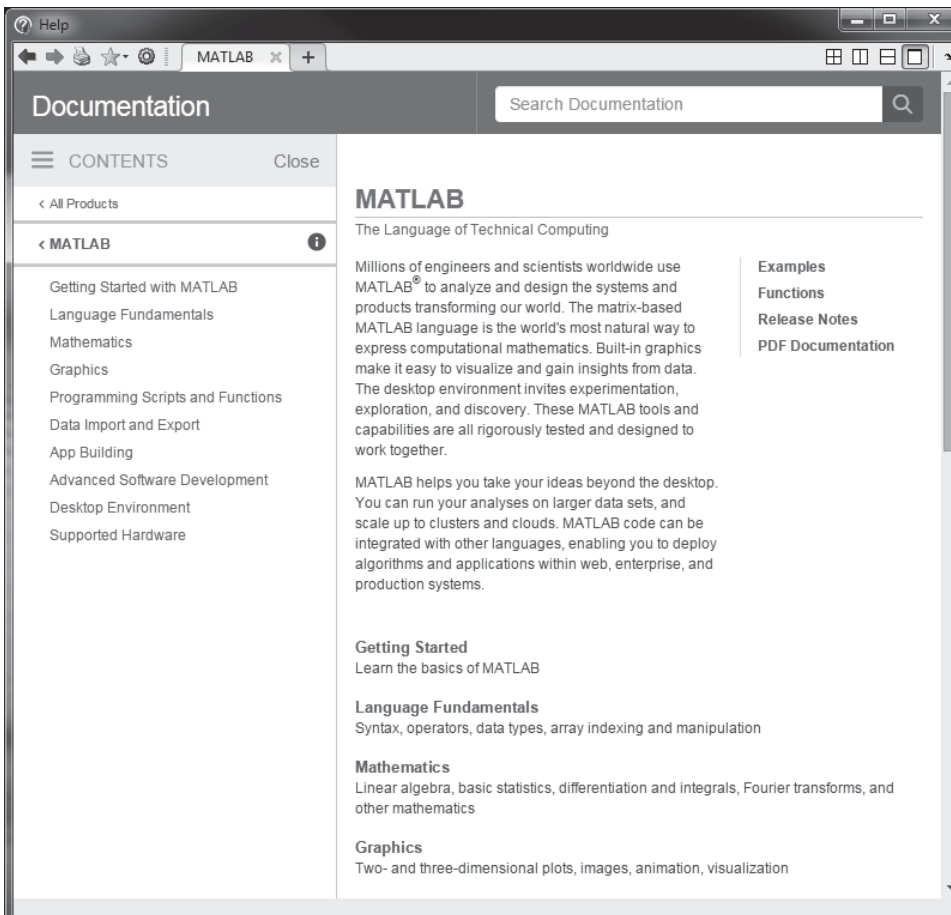


Figure 1-4: The Help Window.

all the windows except the Command Window. The closed windows can be reopened by selecting them from the **layout** icon in the Toolstrip. The windows shown in Figure 1-1 can be displayed by clicking on the **layout** icon and selecting **Default** in the menu that opens. The various windows in Figure 1-1 are docked to the desktop. A window can be undocked (become a separate, independent window) by dragging it out. An independent window can be redocked by clicking on the pull-down menu at the top right-hand side of the window and then selecting Dock.

1.2 WORKING IN THE COMMAND WINDOW

The Command Window is MATLAB's main window and can be used for executing commands, opening other windows, running programs written by the user, and managing the software. An example of the Command Window, with several simple commands that will be explained later in this chapter, is shown in Figure 1-5.

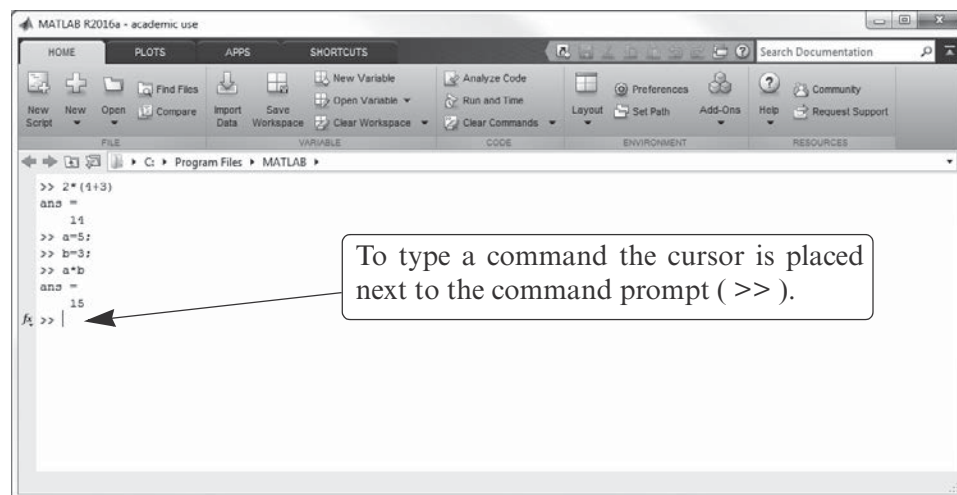


Figure 1-5: The Command Window.

Notes for working in the Command Window:

- To type a command, the cursor must be placed next to the command prompt (>>).
- Once a command is typed and the **Enter** key is pressed, the command is executed. However, only the last command is executed. Everything executed previously (that might be still displayed) is unchanged.
- Several commands can be typed in the same line. This is done by typing a comma between the commands. When the **Enter** key is pressed, the commands are executed in order from left to right.
- It is not possible to go back to a previous line that is displayed in the Command

Window, make a correction, and then re-execute the command.

- A previously typed command can be recalled to the command prompt with the up-arrow key (\uparrow). When the command is displayed at the command prompt, it can be modified if needed and then executed. The down-arrow key (\downarrow) can be used to move down the list of previously typed commands.
- If a command is too long to fit in one line, it can be continued to the next line by typing three periods ... (called an ellipsis) and pressing the **Enter** key. The continuation of the command is then typed in the new line. The command can continue line after line up to a total of 4,096 characters.

The semicolon (;):

When a command is typed in the Command Window and the **Enter** key is pressed, the command is executed. Any output that the command generates is displayed in the Command Window. If a semicolon (;) is typed at the end of a command, the output of the command is not displayed. Typing a semicolon is useful when the result is obvious or known, or when the output is very large.

If several commands are typed in the same line, the output from any of the commands will not be displayed if a semicolon instead of a comma is typed between the commands.

Typing %:

When the symbol % (percent) is typed at the beginning of a line, the line is designated as a comment. This means that when the **Enter** key is pressed the line is not executed. The % character followed by text (comment) can also be typed after a command (in the same line). This has no effect on the execution of the command.

Usually there is no need for comments in the Command Window. Comments, however, are frequently used in a program to add descriptions or to explain the program (see Chapters 4 and 6).

The `clc` command:

The `clc` command (type `clc` and press **Enter**) clears the Command Window. After typing in the Command Window for a while, the display may become very long. Once the `clc` command is executed, a clear window is displayed. The command does not change anything that was done before. For example, if some variables were defined previously (see Section 1.6), they still exist and can be used. The up-arrow key can also be used to recall commands that were typed before.

The Command History Window:

The Command History Window lists the commands that have been entered in the Command Window. This includes commands from previous sessions. A command in the Command History Window can be used again in the Command Window. By double-clicking on the command, the command is reentered in the Command Window and executed. It is also possible to drag the command to the Command Window, make changes if needed, and then execute it. The list