

Contents

Chapter 1 Introduction to MATLAB

1

- 1.1 The Advantages of MATLAB 2**
- 1.2 Disadvantages of MATLAB 3**
- 1.3 The MATLAB Environment 4**
 - 1.3.1 The MATLAB Desktop 4
 - 1.3.2 The Command Window 6
 - 1.3.3 The Toolstrip 7
 - 1.3.4 The Command History Window 8
 - 1.3.5 The Document Window 8
 - 1.3.6 Figure Windows 11
 - 1.3.7 Docking and Undocking Windows 12
 - 1.3.8 The MATLAB Workspace 12
 - 1.3.9 The Workspace Browser 14
 - 1.3.10 The Current Folder Browser 14
 - 1.3.11 Getting Help 15
 - 1.3.12 A Few Important Commands 18
 - 1.3.13 The MATLAB Search Path 19
- 1.4 Using MATLAB as a Calculator 21**
- 1.5 MATLAB Script Files 23**
 - 1.5.1 Setting Up a Problem to Solve 24
 - 1.5.2 Creating a Simple MATLAB Script File 24
- 1.6 Summary 28**
 - 1.6.1 MATLAB Summary 28
- 1.7 Exercises 29**

Chapter 2 MATLAB Basics

33

- 2.1 Variables and Arrays 33**

- 2.2 Creating and Initializing Variables in MATLAB 37**
 - 2.2.1 Initializing Variables in Assignment Statements 37
 - 2.2.2 Initializing with Shortcut Expressions 40
 - 2.2.3 Initializing with Built-In Functions 41
 - 2.2.4 Initializing Variables with Keyboard Input 41
- 2.3 Multidimensional Arrays 43**
 - 2.3.1 Storing Multidimensional Arrays in Memory 45
 - 2.3.2 Accessing Multidimensional Arrays with One Dimension 46
- 2.4 Subarrays 46**
 - 2.4.1 The end Function 47
 - 2.4.2 Using Subarrays on the Left-Hand Side of an Assignment Statement 47
 - 2.4.3 Assigning a Scalar to a Subarray 49
- 2.5 Special Values 49**
- 2.6 Displaying Output Data 51**
 - 2.6.1 Changing the Default Format 52
 - 2.6.2 The disp Function 53
 - 2.6.3 Formatted Output with the fprintf Function 54
- 2.7 Data Files 55**
- 2.8 Scalar and Array Operations 58**
 - 2.8.1 Scalar Operations 58
 - 2.8.2 Array and Matrix Operations 58
- 2.9 Hierarchy of Operations 62**
- 2.10 Built-in MATLAB Functions 65**
 - 2.10.1 Optional Results 65
 - 2.10.2 Using MATLAB Functions with Array Inputs 65
 - 2.10.3 Common MATLAB Functions 66
- 2.11 Introduction to Plotting 67**
 - 2.11.1 Using Simple xy Plots 68
 - 2.11.2 Printing a Plot 69
 - 2.11.3 Multiple Plots 70
 - 2.11.4 Line Color, Line Style, Marker Style, and Legends 71
- 2.12 Examples 75**
- 2.13 MATLAB Applications: Vector Mathematics 82**
 - 2.13.1 Vector Addition and Subtraction 84
 - 2.13.2 Vector Multiplication 85
- 2.14 MATLAB Applications: Matrix Operations and Simultaneous Equations 90**
 - 2.14.1 The Matrix Inverse 91
- 2.15 Debugging MATLAB Programs 92**
- 2.16 Summary 94**
 - 2.16.1 Summary of Good Programming Practice 95
 - 2.16.2 MATLAB Summary 96
- 2.17 Exercises 99**

Chapter 3 Two-Dimensional Plots

111

- 3.1 Additional Plotting Features for Two-Dimensional Plots 111**
 - 3.1.1 Logarithmic Scales 111
 - 3.1.2 Controlling *x*- and *y*-axis Plotting Limits 116
 - 3.1.3 Plotting Multiple Plots on the Same Axes 117
 - 3.1.4 Creating Multiple Figures 117
 - 3.1.5 Subplots 121
 - 3.1.6 Controlling the Spacing between Points on a Plot 122
 - 3.1.7 Enhanced Control of Plotted Lines 126
 - 3.1.8 Enhanced Control of Text Strings 127
- 3.2 Polar Plots 130**
- 3.3 Annotating and Saving Plots 132**
- 3.4 Additional Types of Two-Dimensional Plots 135**
- 3.5 Using the `plot` Function with Two-Dimensional Arrays 140**
- 3.6 Plots with Two *y* Axes 142**
- 3.7 Summary 149**
 - 3.7.1 Summary of Good Programming Practice 150
 - 3.7.2 MATLAB Summary 151
- 3.8 Exercises 151**

Chapter 4 Branching Statements and Program Design

157

- 4.1 Introduction to Top-Down Design Techniques 157**
- 4.2 Use of Pseudocode 161**
- 4.3 The `logical` Data Type 162**
 - 4.3.1 Relational and Logic Operators 162
 - 4.3.2 Relational Operators 163
 - 4.3.3 A Caution About the `==` and `~=` Operators 164
 - 4.3.4 Logic Operators 165
 - 4.3.5 Logical Functions 169
- 4.4 Branches 171**
 - 4.4.1 The `if` Construct 171
 - 4.4.2 Examples Using `if` Constructs 173
 - 4.4.3 Notes Concerning the Use of `if` Constructs 179
 - 4.4.4 The `switch` Construct 182
 - 4.4.5 The `try/catch` Construct 183
- 4.5 More on Debugging MATLAB Programs 189**
- 4.6 Code Sections 196**
- 4.7 MATLAB Applications: Roots of Polynomials 198**
- 4.8 Summary 201**
 - 4.8.1 Summary of Good Programming Practice 201
 - 4.8.2 MATLAB Summary 202
- 4.9 Exercises 203**

Chapter 5 Loops and Vectorization 207

- 5.1 The while Loop 207**
- 5.2 The for Loop 213**
 - 5.2.1 Details of Operation 220
 - 5.2.2 Vectorization: A Faster Alternative to Loops 222
 - 5.2.3 The MATLAB Just-In-Time (JIT) Compiler 223
 - 5.2.4 The break and continue Statements 227
 - 5.2.5 Nesting Loops 228
- 5.3 Logical Arrays and Vectorization 229**
 - 5.3.1 Creating the Equivalent of if/else Constructs with Logical Arrays 230
- 5.4 The MATLAB Profiler 232**
- 5.5 Additional Examples 235**
- 5.6 The textread Function 250**
- 5.7 MATLAB Applications: Statistical Functions 252**
- 5.8 MATLAB Applications: Curve Fitting and Interpolation 255**
 - 5.8.1 General Least-Squares Fits 255
 - 5.8.2 Cubic Spline Interpolation 262
 - 5.8.3 Interactive Curve-Fitting Tools 267
- 5.9 Summary 271**
 - 5.9.1 Summary of Good Programming Practice 271
 - 5.9.2 MATLAB Summary 272
- 5.10 Exercises 272**

Chapter 6 Basic User-Defined Functions 283

- 6.1 Introduction to MATLAB Functions 284**
- 6.2 Variable Passing in MATLAB: The Pass-by-Value Scheme 290**
- 6.3 Optional Arguments 300**
- 6.4 Sharing Data Using Global Memory 305**
- 6.5 Preserving Data between Calls to a Function 313**
- 6.6 Built-In MATLAB Functions: Sorting Functions 318**
- 6.7 Built-In MATLAB Functions: Random Number Functions 320**
- 6.8 Summary 320**
 - 6.8.1 Summary of Good Programming Practice 321
 - 6.8.2 MATLAB Summary 321
- 6.9 Exercises 322**

Chapter 7 Advanced Features of User-Defined Functions 331

- 7.1 Function Functions 331**
- 7.2 Function Handles 336**

7.3	Functions <code>eval</code> and <code>feval</code>	341
7.4	Local Functions, Private Functions, and Nested Functions	342
7.4.1	Local Functions	342
7.4.2	Private Functions	344
7.4.3	Nested Functions	345
7.4.4	Order of Function Evaluation	348
7.4.5	Function Handles and Nested Functions	348
7.4.6	The Significance of Function Handles	350
7.5	An Example Application: Solving Ordinary Differential Equations	351
7.6	Anonymous Functions	358
7.7	Recursive Functions	359
7.8	Plotting Functions	360
7.9	Histograms	362
7.10	An Example Application: Numerical Integration	368
7.11	Summary	374
7.11.1	Summary of Good Programming Practice	374
7.11.2	MATLAB Summary	375
7.12	Exercises	375

Chapter 8 Complex Numbers and Additional Plots

385

8.1	Complex Data	385
8.1.1	Complex Variables	387
8.1.2	Using Complex Numbers with Relational Operators	387
8.1.3	Complex Functions	388
8.1.4	Plotting Complex Data	394
8.2	Multidimensional Arrays	397
8.3	Gallery of MATLAB Plots	399
8.4	Line Plots	410
8.4.1	The <code>plot3</code> Function	410
8.4.2	Changing the Viewpoint of Three-dimensional Plots	414
8.4.3	The <code>fplot3</code> Function	414
8.4.4	The <code>fimplicit</code> Function	415
8.5	Discrete Data Plots	417
8.5.1	The <code>stem3</code> Function	419
8.5.2	The <code>scatter</code> Function	420
8.5.3	The <code>scatter3</code> Function	424
8.6	Polar Plots	426
8.6.1	The <code>compass</code> Function	429
8.6.2	The <code>ezpolar</code> Function	429
8.7	Contour Plots	431
8.7.1	Function <code>contour</code>	431
8.7.2	Function <code>contourf</code>	433

- 8.7.3 Function `contour3` 435
- 8.7.4 Function `fcontour` 435
- 8.8 Surface and Mesh Plots 436**
 - 8.8.1 Creating Surface and Mesh Plots 437
 - 8.8.2 Creating Three-Dimensional Objects using Surface and Mesh Plots 442
 - 8.8.3 Ribbon Plots 444
 - 8.8.4 Function `pcolor` 445
 - 8.8.5 Functions `fsurf` and `fmesh` 447
 - 8.8.6 Function `fimplicit3` 448
- 8.9 Pie Charts, Bar Plots, and Histograms 450**
 - 8.9.1 The `area` Function 451
 - 8.9.2 Bar Plots 452
 - 8.9.3 Two-Dimensional Histograms 456
- 8.10 Color Order, Color Maps, and Color Bars 457**
 - 8.10.1 Plot Color Order 457
 - 8.10.2 Color Maps 459
 - 8.10.3 Color Bars 459
- 8.11 Summary 463**
 - 8.11.1 Summary of Good Programming Practice 463
 - 8.11.2 MATLAB Summary 463
- 8.12 Exercises 464**

Chapter 9 Additional Data Types

471

- 9.1 Character Arrays versus Strings 472**
 - 9.1.1 Character Arrays 472
 - 9.1.2 Strings 473
- 9.2 Character Arrays and Character Functions 473**
 - 9.2.1 Character Array Conversion Functions 474
 - 9.2.2 Creating Two-Dimensional Character Arrays 475
 - 9.2.3 Concatenating Character Arrays 476
 - 9.2.4 Comparing Character Arrays 476
 - 9.2.5 Searching/Replacing Characters within a Character Array 480
 - 9.2.6 Uppercase and Lowercase Conversion 481
 - 9.2.7 Trimming Whitespace from Strings 482
 - 9.2.8 Numerical-to-Character Array Conversions 482
 - 9.2.9 String-to-Numerical Conversions 484
- 9.3 The `string` Data Type 490**
 - 9.3.1 Creating Strings 491
 - 9.3.2 Converting Data into Strings 491
 - 9.3.3 Converting Strings to Other Data Types 493
 - 9.3.4 Concatenating Strings 494
 - 9.3.5 Comparing Strings 494
 - 9.3.6 Searching for Substrings within a String 495

- 9.3.7 Extracting Substrings from a String 496
- 9.3.8 Inserting Strings into a String 497
- 9.3.9 Replacing Characters within a String 497
- 9.3.10 Erasing Characters in a String 498
- 9.3.11 Uppercase and Lowercase Conversion 499
- 9.3.12 Trimming Whitespace from Strings 499
- 9.4 Summary of Character Array and String Functions 499**
- 9.5 The `single` Data Type 503**
- 9.6 Integer Data Types 504**
- 9.7 Limitations of the `single` and Integer Data Types 505**
- 9.8 The `datetime` and `duration` Data Types 507**
 - 9.8.1 The `datetime` Data Type 507
 - 9.8.2 The `duration` Data Type 508
 - 9.8.3 `calendarDuration` Arrays 508
 - 9.8.4 Time Calculations 509
 - 9.8.5 Using Time Data in MATLAB 511
- 9.9 Summary 513**
 - 9.9.1 Summary of Good Programming Practice 513
 - 9.9.2 MATLAB Summary 513
- 9.10 Exercises 514**

Chapter 10 Sparse Arrays, Cell Arrays, Structures, and Tables

517

- 10.1 Sparse Arrays 517**
 - 10.1.1 The `sparse` Attribute 519
- 10.2 Cell Arrays 525**
 - 10.2.1 Creating Cell Arrays 527
 - 10.2.2 Using Braces { } as Cell Constructors 528
 - 10.2.3 Viewing the Contents of Cell Arrays 528
 - 10.2.4 Extending Cell Arrays 529
 - 10.2.5 Deleting Cells in Arrays 531
 - 10.2.6 Using Data in Cell Arrays 532
 - 10.2.7 Cell Arrays of Strings 532
 - 10.2.8 The Significance of Cell Arrays 534
 - 10.2.9 Summary of `cell` Functions 538
- 10.3 Structure Arrays 539**
 - 10.3.1 Creating Structure Arrays 539
 - 10.3.2 Adding Fields to Structures 542
 - 10.3.3 Removing Fields from Structures 542
 - 10.3.4 Using Data in Structure Arrays 543
 - 10.3.5 The `getfield` and `setfield` Functions 544
 - 10.3.6 Dynamic Field Names 545
 - 10.3.7 Using the `size` Function with Structure Arrays 546

- 10.3.8 Nesting Structure Arrays 547
- 10.3.9 Summary of `structure` Functions 548
- 10.4 Table Arrays 548**
 - 10.4.1 Creating Table Arrays 548
 - 10.4.2 Accessing Data in a Table 551
 - 10.4.3 Table Metadata (Properties) 552
 - 10.4.4 Examining the Contents and Properties of a Table 553
 - 10.4.5 Table Summary 554
- 10.5 Summary 560**
 - 10.5.1 Summary of Good Programming Practice 560
 - 10.5.2 MATLAB Summary 561
- 10.6 Exercises 561**

Chapter 11 Input-Output Functions

565

- 11.1 The `textread` Function 565**
- 11.2 More about the `load` and `save` Commands 567**
- 11.3 An Introduction to MATLAB File Processing 570**
- 11.4 File Opening and Closing 571**
 - 11.4.1 The `fopen` Function 571
 - 11.4.2 The `fclose` Function 574
- 11.5 Binary I/O Functions 575**
 - 11.5.1 The `fwrite` Function 575
 - 11.5.2 The `fread` Function 575
- 11.6 Formatted I/O Functions 580**
 - 11.6.1 The `fprintf` Function 580
 - 11.6.2 Understanding Format Conversion Specifiers 581
 - 11.6.3 How Format Strings Are Used 583
 - 11.6.4 The `sprintf` Function 585
 - 11.6.5 The `fscanf` Function 587
 - 11.6.6 The `fgetl` Function 588
 - 11.6.7 The `fgets` Function 589
- 11.7 Comparing Formatted and Binary I/O Functions 589**
- 11.8 File Positioning and Status Functions 594**
 - 11.8.1 The `exist` Function 595
 - 11.8.2 The `ferror` Function 597
 - 11.8.3 The `feof` Function 598
 - 11.8.4 The `ftell` Function 598
 - 11.8.5 The `frewind` Function 598
 - 11.8.6 The `fseek` Function 598
- 11.9 The `textscan` Function 604**
- 11.10 Function `uiimport` 606**
- 11.11 Summary 609**
 - 11.11.1 Summary of Good Programming Practice 610
 - 11.11.2 MATLAB Summary 610
- 11.12 Exercises 611**

Chapter 12 User-Defined Classes and Object-Oriented Programming

615

- 12.1 An Introduction to Object-Oriented Programming 615**
 - 12.1.1 Objects 616
 - 12.1.2 Messages 617
 - 12.1.3 Classes 617
 - 12.1.4 Static Methods 618
 - 12.1.5 Class Hierarchy and Inheritance 620
 - 12.1.6 Object-Oriented Programming 620
- 12.2 The Structure of a MATLAB Class 621**
 - 12.2.1 Creating a Class 622
 - 12.2.2 Adding Methods to a Class 624
 - 12.2.3 Listing Class Types, Properties, and Methods 628
 - 12.2.4 Attributes 629
- 12.3 Value Classes versus Handle Classes 633**
 - 12.3.1 Value Classes 634
 - 12.3.2 Handle Classes 635
- 12.4 Destructors: The delete Method 638**
- 12.5 Access Methods and Access Controls 640**
 - 12.5.1 Access Methods 640
 - 12.5.2 Access Controls 642
 - 12.5.3 Example: Creating a Timer Class 642
 - 12.5.4 Notes on the MyTimer Class 647
- 12.6 Static Methods 648**
- 12.7 Defining Class Methods in Separate Files 649**
- 12.8 Overriding Operators 650**
- 12.9 Events and Listeners 655**
 - 12.9.1 Property Events and Listeners 658
- 12.10 Exceptions 659**
 - 12.10.1 Creating Exceptions in Your Own Programs 660
 - 12.10.2 Catching and Fixing Exceptions 661
- 12.11 Superclasses and Subclasses 662**
 - 12.11.1 Defining Superclasses and Subclasses 663
 - 12.11.2 Example Using Superclasses and Subclasses 668
- 12.12 Summary 678**
 - 12.12.1 Summary of Good Programming Practice 679
 - 12.12.2 MATLAB Summary 679
- 12.13 Exercises 680**

Chapter 13 Handle Graphics and Animation

685

- 13.1 Handle Graphics 685**
- 13.2 The MATLAB Graphics System 686**
- 13.3 Object Handles 688**

- 13.4 Examining and Changing Object Properties 689**
 - 13.4.1 Changing Object Properties at Creation Time 689
 - 13.4.2 Changing Object Properties after Creation Time 689
 - 13.4.3 Examining and Changing Properties Using Object Notation 690
 - 13.4.4 Examining and Changing Properties Using get/set Functions 692
 - 13.4.5 Examining and Changing Properties Using the Property Editor 694
- 13.5 Using set to List Possible Property Values 698**
- 13.6 User-Defined Data 700**
- 13.7 Finding Objects 701**
- 13.8 Selecting Objects with the Mouse 703**
- 13.9 Position and Units 706**
 - 13.9.1 Positions of figure Objects 706
 - 13.9.2 Positions of axes and polaraxes Objects 707
 - 13.9.3 Positions of text Objects 707
- 13.10 Printer Positions 710**
- 13.11 Default and Factory Properties 711**
- 13.12 Restoring Default Properties 713**
- 13.13 Graphics Object Properties 713**
- 13.14 Animations and Movies 714**
 - 13.14.1 Erasing and Redrawing 714
 - 13.14.2 Creating a Movie 719
- 13.15 Summary 721**
 - 13.15.1 Summary of Good Programming Practice 721
 - 13.15.2 MATLAB Summary 721
- 13.16 Exercises 722**

Chapter 14 MATLAB Apps and Graphical User Interfaces 725

- 14.1 How a Graphical User Interface Works 726**
- 14.2 Creating and Displaying a Graphical User Interface 732**
 - 14.2.1 The Structure of a Callback Function (Method) 738
 - 14.2.2 Adding Application Data to a Figure 739
- 14.3 Object Properties 739**
 - 14.3.1 Key Properties of Numerical Components 741
 - 14.3.2 Key Properties of Text Components 743
- 14.4 Additional Containers: Panels, Tab Groups, and Button Groups 749**
 - 14.4.1 Panels 749
 - 14.4.2 Tab Groups 752
 - 14.4.3 Button Groups 752
- 14.5 Dialog Boxes 754**
 - 14.5.1 Alert Dialog Boxes 755

- 14.5.2 Confirmation Dialog Boxes 755
- 14.5.3 Input Dialog Boxes 757
- 14.5.4 The `uigetfile`, `uisetfile`, and `uigetdir` Dialog Boxes 757
- 14.5.5 The `uisetcolor` and `uisetfont` Dialog Boxes 759
- 14.6 Menus 760**
 - 14.6.1 Creating Your Own Menus 763
 - 14.6.2 Accelerator Keys and Keyboard Mnemonics 763
- 14.7 Summary 774**
 - 14.7.1 Summary of Good Programming Practice 775
 - 14.7.2 MATLAB Summary 775
- 14.8 Exercises 777**
- A UTF-8 Character Set 779**
- B Answers to Quizzes 781**
- Index 807**

Chapter 15 Guide-Based Graphical User Interfaces (On-line Only)

- 15.1 How a Graphical User Interface Works**
- 15.2 Creating and Displaying a Graphical User Interface**
 - 15.2.1 A Look Under the Hood
 - 15.2.2 The Structure of a Callback Subfunction
 - 15.2.3 Adding Application Data to a Figure
 - 15.2.4 A Few Useful Functions
- 15.3 Object Properties**
- 15.4 Graphical User Interface Components**
 - 15.4.1 Static Text Fields
 - 15.4.2 Edit Boxes
 - 15.4.3 Pushbuttons
 - 15.4.4 Toggle Buttons
 - 15.4.5 Checkboxes and Radio Buttons
 - 15.4.6 Popup Menus
 - 15.4.7 List Boxes
 - 15.4.8 Sliders
 - 15.4.9 Tables
- 15.5 Additional Containers: Panels and Button Groups**
 - 15.5.1 Panels
 - 15.5.2 Button Groups
- 15.6 Dialog Boxes**
 - 15.6.1 Error and Warning Dialog Boxes
 - 15.6.2 Input Dialog Boxes

15.6.3 The `uigetfile`, `uisetfile`, and `uigetdir` Dialog Boxes

15.6.4 The `uisetcolor` and `uisetfont` Dialog Boxes

15.7 Menus

15.7.1 Suppressing the Default Menu

15.7.2 Creating Your Own Menus

15.7.3 Accelerator Keys and Keyboard Mnemonics

15.7.4 Creating Context Menus

15.8 Tips for Creating Efficient GUIs

15.8.1 Tool Tips

15.8.2 Toolbars

15.8.3 Additional Enhancements

15.9 Summary

15.9.1 Summary of Good Programming Practice

15.9.2 MATLAB Summary

15.10 Exercises