Cambridge University Press
978-1-107-66222-3 - A Guide to MATLAB®: for Beginners and Experienced Users: Third Edition
Brian R. Hunt, Ronald L. Lipsman and Jonathan M. Rosenberg
Excerpt
More information

Chapter 1

Getting Started

In this chapter, we will introduce you to the tools you need in order to begin using MATLAB effectively. These include the following: some relevant information on computer platforms and software; installation protocols; how to launch MATLAB, enter commands and use online help; a roster of MATLAB's various windows; and, finally, how to exit the program. We know that you are anxious to get started using MATLAB, so we will keep this chapter brief. After you complete it, you can go immediately to Chapter 2 to find concrete and simple instructions for using MATLAB to do mathematics. We describe the MATLAB interface more elaborately in Chapter 3.

Platforms and Versions

The latest version of MATLAB, MATLAB 8 (Release 2012b or later) runs on three operating systems: various versions of Microsoft Windows (both 32-Bit and 64-Bit), Mac OS X, and many distributions of Linux (including Ubuntu, Debian, SUSE, and Red Hat Enterprise). For exact system requirements, you can consult:

```
http://www.mathworks.com/support/sysreq/current_release/
```

For definitiveness, we shall assume that the reader is using a Windows computer. In those very few instances where our instructions must be tailored differently for Linux or Mac users, we shall point it out clearly.

✓ We use the word Windows to refer to all flavors of the Windows operating system. Windows 7 or 8, especially the 64-Bit version, is preferred, though you can also use some Windows XP or Windows Vista installations.

This book is written to be compatible with the current version of MATLAB, namely MATLAB 8 (Release 2012b or later). The vast majority of the MATLAB *commands* we describe apply also to MATLAB 7 (Release 2012a or earlier), but in those earlier versions of the software, many features of the MATLAB interface are different. When important differences between different versions arise, we will point them out. We also note that the differences between the MATLAB Professional Version and the MATLAB Student Version are rather minor, and virtually unnoticeable to a beginner, or even a mid-level user. Again, in the few instances where we describe a MATLAB feature that is available only in the Professional Version, we highlight that fact clearly.

Installation

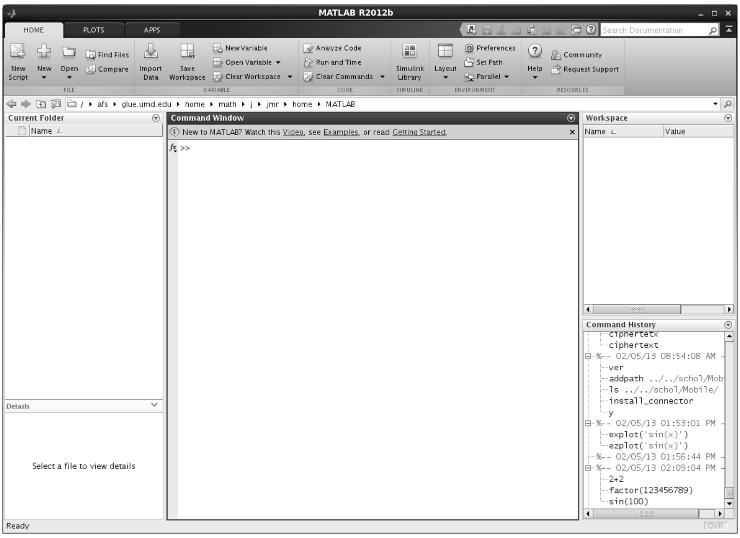
If you intend to run MATLAB, especially the Student Version, on your own computer, it is quite possible that you will have to install it yourself. You can easily accomplish this using the product CDs or via web download, as appropriate. Follow the installation instructions as you would with the installation of any new software. At some point in the installation, you may be asked which *toolboxes* you wish to install. Unless you have severe space limitations, we suggest that you install any included in your license that seem of interest to you or that you think you might use at some point in the future. However, for the purposes of this book, you should be sure to include the *Symbolic Math Toolbox*. We also strongly encourage you to install Simulink, which is described in Chapter 9.

Starting MATLAB

You start MATLAB as you would any other software application. In Windows, you access it via the **Start** menu, under a name like **MATLAB R2013b** or **Student MATLAB**. Alternatively, you may have a desktop icon that enables you to start MATLAB with a simple double-click. In Linux, generally you need only type **`matlab`** in a terminal window, though you may first have to find the `bin` subdirectory of the MATLAB installation directory and add it to your path. Or you may have an icon on your desktop that achieves the task.

However you start MATLAB, you will briefly see a window that displays the MATLAB logo as well as some product information, and then a *MATLAB Desktop* window will launch. That window will contain a *Toolstrip* and several tabs at the top, with five embedded windows just below the Toolstrip. The largest and most important window is the *Command Window* in the center. We will go into more detail in Chapter 3 on the use and manipulation of three other windows: the *Command History Window*, the *Current Folder Browser*, and the *Workspace Browser*. For now we concentrate on the Command Window in order to get you started running MATLAB commands as quickly as possible. At the top of the Command Window, you may see some general information about MATLAB, perhaps some special instructions for getting started or accessing help, but most important of all, you will see a *command prompt* (**`>>`** or **`EDU>>`**). If the Command Window is "active," its title bar will be dark, and the prompt will be followed by a cursor (a blinking vertical line). That is the place where you will enter your MATLAB commands (see Chapter 2). If the Command Window is not active, just click in it anywhere. Figure 1.1 contains an example of a newly launched MATLAB Desktop.

✓ MATLAB 7 had a Desktop with similar windows, but with fewer menu options and a different look. If you need to do a fast calculation and do not want to wait for the Desktop to load, there is an option to run MATLAB without the Desktop. (In Linux or from a Windows Command Prompt, the command is **`matlab -nodesktop`**.) This gives you a Command Window in a simple terminal window with very few "frills," which often runs faster.



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>>
```

Select a file to view details

Ready

```
ciphertetx
ciphertext
%-- 02/05/13 08:54:08 AM -
ver
addpath ../../schol/Mob
ls ../../schol/Mobile/
install_connector
y
%-- 02/05/13 01:53:01 PM -
explot('sin(x)')
ezplot('sin(x)')
%-- 02/05/13 01:56:44 PM -
%-- 02/05/13 02:09:04 PM -
2+2
factor(123456789)
sin(100)
```

Figure 1.1. A MATLAB Desktop.

Typing in the Command Window

Click in the Command Window to make it active. When a window becomes active, its title bar darkens, and a blinking cursor appears after the prompt. Now you can begin entering commands. Try typing **`2+2`**, then press the ENTER or RETURN key. Next try **`factor(123456789)`**, and finally **`sin(100)`**. Your MATLAB Desktop should look like Figure 1.2.

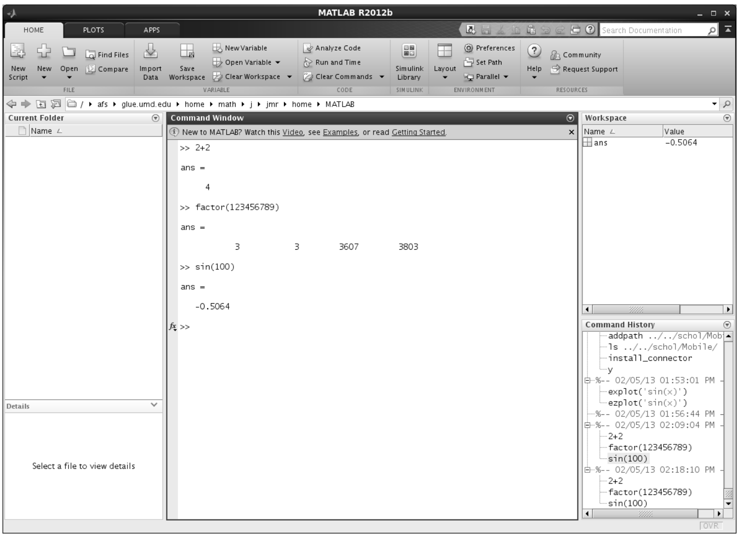
Online Help

MATLAB has extensive online help. In fact, you should be able to become quite proficient with MATLAB using only this book and the online help.

The easiest way to get help on any aspect of MATLAB or its associated toolboxes is to bring up the *Help Browser* by clicking on one of the two “question mark” icons at the top right of the Desktop or by typing into the search box (also located at the top right of the Toolstrip). You can also type **`doc`** in the Command Window, followed if you like by the name of the command on which you want documentation. Figure 1.3 depicts the MATLAB Help Browser.

The Help Browser is a powerful and easy-to-use aid in finding the information you need on various features of MATLAB. Like any such tool, the more you use it, the more adept you become at its use.

If you enter a word into the **Search Documentation** box of the Help Browser, or if you enter a word into the search box at the top right of the Desktop Toolstrip, you will see a list of possible matches. On the left of the Help Browser, three regions will appear, enabling you to **Refine by Product**, **Refine by Category**, or **Refine by**



```
>> 2+2

ans =

     4

>> factor(123456789)

ans =

           3           3        3607        3803

>> sin(100)

ans =

   -0.5064

>>
```

```
addpath ../../schol/Mob
ls ../../schol/Mobile/
install_connector
y
%-- 02/05/13 01:53:01 PM --
explot('sin(x)')
ezplot('sin(x)')
%-- 02/05/13 01:56:44 PM --
%-- 02/05/13 02:09:04 PM --
2+2
factor(123456789)
sin(100)
%-- 02/05/13 02:18:10 PM --
2+2
factor(123456789)
sin(100)
```

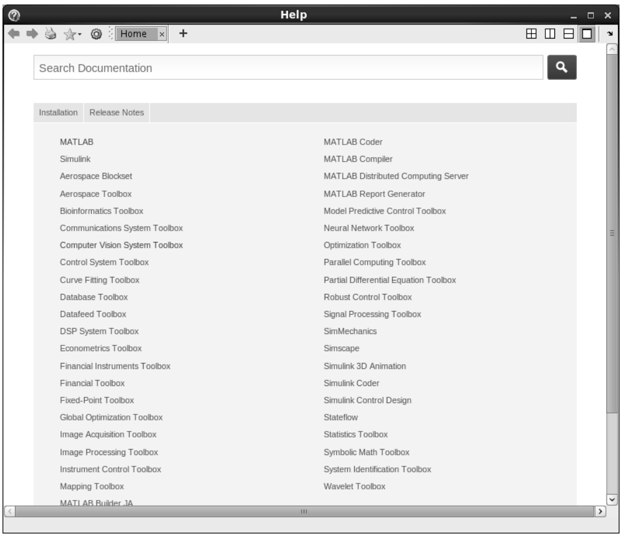
Figure 1.2. The MATLAB Desktop with Several Commands Evaluated.

Type. Refining by product may be important if you have installed a large number of toolboxes, as by default all of them will be searched, but it may be that you are only interested in a single toolbox. For example, a certain command may have a numerical version in MATLAB and a symbolic version in the Symbolic Math Toolbox, and you may want only one of these. Refining by category may be important if the same word is used in several contexts, even within the same toolbox. For example, even within MATLAB, the word "scale" could be used in connection with rescaling graphics, or it could be used in the context of rescaling numbers. Refining by the categories "Graphics" or "Mathematics" would help you find the appropriate help page for your purposes. Refining by type helps you limit your search to MATLAB functions or to examples or to concepts.

✓ In MATLAB Release 2012a or earlier, the Help Browser had a slightly different appearance, with more than one pane. It could still be used in most of the ways we have described, either by searching down the documentation tree or searching for a particular command or key word.

Sometimes the Help Browser produces more detail than you need and it is convenient to have something simpler. If you just want to know the name or syntax of a particular MATLAB command, it is useful to bring up the *Function Browser* by right-clicking in the Command Window and selecting the **Function Browser** menu item.

Finally, for the quickest and simplest version of product help, you can type **`help`** in the Command Window, followed if you wish by the name of a command or a category of commands. For example, typing **`help`** at the command prompt will reveal a long list of topics for which help is available. Typing **`help general`** will give a long list of "general purpose" MATLAB commands. Typing **`help factor`** will



MATLAB
Simulink
Aerospace Blockset
Aerospace Toolbox
Bioinformatics Toolbox
Communications System Toolbox
Computer Vision System Toolbox
Control System Toolbox
Curve Fitting Toolbox
Database Toolbox
Datafeed Toolbox
DSP System Toolbox
Econometrics Toolbox
Financial Instruments Toolbox
Financial Toolbox
Fixed-Point Toolbox
Global Optimization Toolbox
Image Acquisition Toolbox
Image Processing Toolbox
Instrument Control Toolbox
Mapping Toolbox
MATLAB Builder JA

MATLAB Coder
MATLAB Compiler
MATLAB Distributed Computing Server
MATLAB Report Generator
Model Predictive Control Toolbox
Neural Network Toolbox
Optimization Toolbox
Parallel Computing Toolbox
Partial Differential Equation Toolbox
Robust Control Toolbox
Signal Processing Toolbox
SimMechanics
Simscape
Simulink 3D Animation
Simulink Coder
Simulink Control Design
Stateflow
Statistics Toolbox
Symbolic Math Toolbox
System Identification Toolbox
Wavelet Toolbox

Figure 1.3. The MATLAB Help Browser.

display brief information about the **factor** command. In every instance above, more information than your screen can hold will scroll by. You can use the scroll bar on the right of the Command Window to scroll back up. Alternatively, you can force MATLAB to display information one screenful at a time by typing **more on**. You press the space bar to display the next screenful; type **help more** for details. Typing **more on** affects all subsequent commands, until you type **more off**.

The **lookfor** command searches the first line of every MATLAB help file for a specified string (use **lookfor -all** to search all lines). For example, if you wanted to see a list of all MATLAB commands that contain the word "factor" as part of the command name or brief description, then you would type **lookfor factor**. If the command you are looking for appears in the list, then you can use **help** or the Function Browser on that command to learn more about it.

If you are patient, and not overly anxious to get to Chapter 2, you can type **demo** in the Command Window to bring up a list of videos and scripts that illustrate various aspects of MATLAB and your installed toolboxes.

MATLAB Windows

We have already described the MATLAB Command Window and the Help Browser, and have mentioned in passing the Command History Window, the Current Folder Browser, and the Workspace Browser. These, and several other windows you will encounter as you work with MATLAB, will allow you to do the following: control files and folders that you and MATLAB will need to access; write and edit the small MATLAB programs (M-files) that you will use to run MATLAB effectively; keep track of the variables and functions that you define as you use MATLAB; display and manipulate graphics; and design models to solve problems and simulate processes. Some of these windows launch separately, and some are embedded in the Desktop. You can dock into the Desktop many of those that launch separately by clicking on the downward curved arrow (if there is one) in their tool bars, or by clicking on a button that looks like ▼ (if there is one) in the upper right corner of the window and selecting **Dock**. You can separate windows inside your MATLAB Desktop out to your computer desktop by clicking on the ▼ button and selecting **Undock**. These features are described more thoroughly in later chapters. For now, we want to call your attention to the other main type of window you will encounter, namely graphics windows. Many of the commands you issue will generate graphics or pictures. These will appear in a separate window, called a *figure window*. In Chapter 5, we will discuss how to generate and manipulate MATLAB figure windows most effectively.

See Figure 2.1 in Chapter 2 for a simple example of a figure window.

Ending a Session

The simplest way to conclude a MATLAB session is to type **`quit`** or **`exit`** (they are indistinguishable) at the prompt. You can also click on the button that generally closes your windows (usually an × in the upper right-hand corner of the Desktop window). *Before* you exit MATLAB, you should be sure to save your work and print any graphics or other files you need. Some strategies for doing so are discussed in Chapter 3.

Mobile Access

Once you've become adept at using MATLAB, you can carry it with you by using the MATLAB Mobile app on a smartphone or tablet device. The app is free to download and use if you have a valid MATLAB license. You can configure it either to connect to your computer, running either the Professional Version or Student Version (R2012a or later) of MATLAB, or to "connect to the cloud" using a MathWorks Account (which you can get with the Professional Version of MATLAB). If you connect to the cloud, then MATLAB Mobile will work even if your computer is turned off. We have tested MATLAB Mobile in both configurations and it does enable you to do many calculations wherever you may be, though you do not have access to the MATLAB Editor or the Simulink graphical environment.

Chapter 2

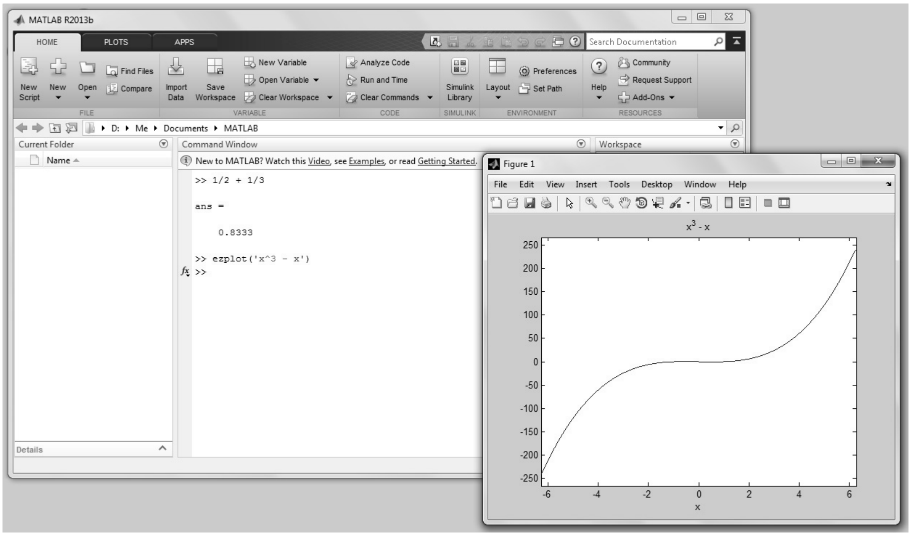
MATLAB Basics

In this chapter, you will start learning how to use MATLAB to do mathematics. We recommend that you read this chapter while running MATLAB. Try the commands as you go along, and please experiment with variants of the examples we present – if you're not sure whether a command will work, try it!

☞ *For further practice, you can work the problems in* Practice Set A.

Input and Output

The simplest way to input commands to MATLAB is by typing them in the Command Window. MATLAB returns output in two ways: typically, text or numerical output is returned in the same Command Window, but graphical output appears in a separate figure window. Figure 2.1 shows a sample screen, with both a MATLAB Desktop and a figure window. To generate this screen on your computer, first type **1/2 + 1/3**. Then type **ezplot('x^3 - x')**.



```
>> 1/2 + 1/3

ans =

    0.8333

>> ezplot('x^3 - x')
>>
```

Figure 2.1. MATLAB Windows.

Cambridge University Press
978-1-107-66222-3 - A Guide to MATLAB®: for Beginners and Experienced Users: Third Edition
Brian R. Hunt, Ronald L. Lipsman and Jonathan M. Rosenberg
Excerpt
More information

✓ After typing the latter command, you may see MATLAB briefly display the word "Busy" in the lower left corner of its Desktop. When MATLAB is finished running the command, a new command prompt appears.

Arithmetic

As we have just seen, you can use MATLAB to do arithmetic as you would a calculator. You can add with **+**, subtract with **-**, multiply with *****, divide with **/**, and exponentiate with **^**. For example:

```
>> 3^2 - (5 + 4)/2 + 6*3

ans =
   22.5000
```

MATLAB prints the answer and assigns the value to a *variable* called **ans**. If you want to perform further calculations with the answer, you can use the variable **ans** rather than retype the answer. For example, you can compute the sum of the square and the square root of the previous answer as follows:

```
>> ans^2 + sqrt(ans)

ans =
   510.9934
```

Observe that MATLAB assigns a new value to **ans** with each calculation. To do more complex calculations, you can assign each computed value to a variable of your choosing. The assigned value is substituted when you later use the variable. For example:

```
>> u = cos(10)

u =
   -0.8391

>> v = sin(10)

v =
   -0.5440

>> u^2 + v^2

ans =
   1
```

⇨ **Trigonometric functions in MATLAB use radians, not degrees.**

MATLAB uses double-precision floating-point arithmetic, which is accurate to approximately 15 digits; however, MATLAB displays only five digits by default. To display more digits, type **format long**. Then all subsequent numerical output will have 15 digits displayed. Type **format short** to return to five-digit display.

MATLAB can also do *exact* arithmetic. For example, it can add the fractions $1/2$ and $1/3$ symbolically to obtain the correct fraction $5/6$. We discuss how to do this in the section *Symbolic Expressions, Variable Precision, and Exact Arithmetic* later in this chapter.

Recovering from Problems

Inevitably, when using any mathematical software system, you are bound to encounter minor glitches. Even while entering simple arithmetic commands, you may accidentally mistype an entry or inadvertently violate a MATLAB rule. In this brief section, we mention three methods for coping with these kinds of problems.

Errors in Input

If you make an error in an input line, MATLAB will normally print an error message. For example, here's what happens when you try to evaluate **3u^2**:

```
>> 3u^2

 3u^2
   |
Error: Unexpected MATLAB expression.
```

The error is a missing multiplication operator *****. The correct input would be **3*u^2**. Note that MATLAB places a marker (a vertical line segment) at the place where it thinks the error might be; however, the actual error may have occurred earlier or later in the expression.

✓ Missing multiplication operators and parentheses are among the most common input errors made by beginning users.

☞ *Recall our discussion of online help in Chapter 1. If you can't decipher an error message caused by a MATLAB command, refer to the online help for that command, or use the search feature of the Help Browser.*

You can edit a previously executed input line as follows. First, redisplay the input line by dragging it with the mouse from the Command History to the Command Window, or by simply typing the up-arrow key. Then edit the command and press Enter or Return. You can also use the Command History or the up- and down-arrow keys to scroll back and forth through all the commands you've typed in your current and previous MATLAB sessions.

Stopping Execution

If MATLAB gets hung up in a calculation, or seems to be taking too long to perform an operation, you can usually abort it by typing Ctrl+C; that is, hold down the key labeled Ctrl, or Control, and press C. Make sure the Command Window is highlighted, or else click in the Command Window first. You can also type Ctrl+Break in Windows or Command+. on a Macintosh. While not foolproof, this is the method of choice when MATLAB is not responding.

Algebraic or Symbolic Computation

Using MATLAB's Symbolic Math Toolbox, you can carry out algebraic or symbolic calculations such as factoring polynomials or solving algebraic equations. Type **help symbolic** to make sure that the Symbolic Math Toolbox is installed on your system.

To perform symbolic computations, you must use **syms** to declare the variables you plan to use to be symbolic variables. Consider the following series of commands:

```
>> syms x y
>> (x - y)*(x - y)^2

ans =
(x - y)^3

>> expand(ans)

ans =
x^3 - 3*x^2*y + 3*x*y^2 - y^3

>> factor(ans)

ans =
(x - y)^3
```

✓ Notice that symbolic output is left-justified, while numerical output is indented. This feature is often useful in distinguishing symbolic output from numerical output.

Although MATLAB makes minor simplifications to the expressions you type, it does not make major changes unless you tell it to. Typing **expand** told MATLAB to multiply out the expression, and **factor** forced MATLAB to restore it to factored form.

MATLAB has a command called **simplify**, which you can sometimes use to express a formula as simply as possible. For example:

```
>> simplify((x^3 - y^3)/(x - y))

ans =
x^2 + x*y + y^2
```

✓ MATLAB has a more elaborate command, called **simple**, that tries more methods than **simplify** does. Try both commands on the trigonometric expression **(sin(x) + cos(x))^2** to compare – you'll have to read the online help for **simple** to completely understand the answer.

Substituting in Symbolic Expressions

When you work with symbolic expressions, you often need to substitute a numerical value, or even another symbolic expression, for one (or more) of the original variables in the expression. This can be done with the command **subs**. For example, if the