

## Introduction to MATLAB

### Objectives:

1. To introduce the use of the MATLAB software package
2. To learn elementary mathematics in MATLAB

**Getting Started:** Log onto your machine using your ITS login (the same as for your @ucsc email). Note that unless otherwise specified, mouse clicks are left-clicks.

## 1 Starting MATLAB

Matlab is a commercial software package written by The Mathworks. Matlab is an interactive, numerical computation, visualization and technical graphics programming language for applications in engineering and science. The MATLAB environment that enables you to perform computationally intensive tasks such as: data visualization, data analysis, and numeric computation. MATLAB acts like a sophisticated, programmable calculator, allowing you to perform matrix algebra easily. It can also solve ODEs numerically (i.e. approximately, using some scheme to solve the differential equations accurately, if not exactly). Questions about the name "MATLAB" often arise. MATLAB stands for "MATrix LABoratory". See <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html> for an overview of MATLAB.

To start MATLAB, click on the "Start" button in the lower left corner of the screen, go to the "Math and Statistics and Graphing" menu item, go to the right to MATLAB > R2008b > MATLAB R2008b, or double-click the shortcut icon for MATLAB R2008b on your Windows desktop. This may bring up two windows. The top one is the tip of the day, which you may or may not find useful. For today, it is probably too advanced a tip, and you can just click on the close button. This should leave you with the MATLAB Starter window. This has a prompt, ">>", and a cursor awaiting instruction.

Useful references are available online from the Mathworks site:  
<http://www.mathworks.com/access/helpdesk/help/techdoc/>

## 2 Getting Help!

Matlab has an extensive help facility and a number of demonstration programs. The command "help *function – name*" is the syntax to ask for help regarding the function "*function – name*". For example,

```
>> help real <Enter>
```

provides a description of the function "real(x)". The help facility is also divided into categories, and one can access entire lists of functions. For example,

```
>> help elfun <Enter>
```

produces a list of the elementary functions that are available in Matlab. By itself,

```
>> help <Enter>
```

generates a list of possible categories.

Type `demod` for the demonstrations. These are often more complicated than you bargained for. (*e.g.* check out the cool PEAKS and MODES, which give some idea of the graphics that Matlab is capable of).

Type `demod` for the demonstrations. In the left pane, click the +-sign next to the "Getting Started" item to expand the list of demos.

**Exercise #1.** Run "Working in The Development Environment" video, by selecting the item and clicking on 'Run this demo' in the center pane. Don't worry about catching all the information, simply get an idea of how to navigate the MATLAB environment.

### 3 Stopping Execution

You can stop execution of whatever is currently running by pressing Ctrl+C or Ctrl+Break at any time. On Apple Macintosh platforms, you can also use Command+. (the Command key and the period key) to stop the program.

### 4 Setting up the working directory

On the lab PC workstations users can write to the local disk, however, user files are not saved at logout, and the disk is wiped clean when the machine reboots. Data files created by users can be saved to the local hard disk's Desktop temporary us. When you login on a lab workstation, the user's UCSC Home Directory is mapped as "X:" on the PCs. Files can be copied (dragged) to and from the Home Directory. The Home Directory can also be accessed via SFTP at 'unix.ic.ucsc.edu'. Alternatively, you can save your Matlab files using a USB Flash Drive, or CD-R/RW. Create a folder in your home directory or other device to house the course labs and name it `ams27`.

### 5 Recording Your Work

It is frequently useful to be able to record what happens in a MATLAB session. For example, to prepare a homework assignment or log a lab session, it is not necessary to copy all of the output from the computer screen. The MATLAB `diary` command makes this possible.

For example, suppose you are doing our first homework assignment and you want to record what you are doing in MATLAB. To do this, choose a name, perhaps `hw1` for the file in which you wish to record the output. Then enter `diary hw1` at the Matlab prompt. From this point on, everything that appears in the Command Window will also be recorded in the file `hw1`. When you want to stop recording enter `diary off`. If you want to start recording again, enter `diary on`.

## 6 Numbers and operations

In Matlab, numbers are what you expect: 12.4 entered after the prompt, followed by <Enter> means exactly that to Matlab:

```
>> 12.4
ans = 12.4
```

with answer, being the number 12.4 abbreviated as “ans” by Matlab. Matlab won’t do anything other than interpret the command as a number unless you tell it to do more. Note that <Enter> tells Matlab to execute the command contained on that line. The up arrows on the keyboard also work in recalling earlier commands and moving along each line. Simple arithmetic operations are also straightforward:

```
>> 12.4 * (48.5 + 342/39)
```

means  $12.4(48.5 + \frac{342}{39})$ , and produces the result:

```
ans = 710.1385
```

The arithmetic operations recognized by Matlab are:

+	addition
-	subtraction
/	division
^	raise to power

Multiple commands can be placed on the same line if separated by a “,” or a “;”:

**Question 1:** What type of number does the following arithmetic operation produce?

```
>> theta=pi; exp(i * theta), cos(theta) + i * sin(theta)
```

**Question 2:** Write down the answers supplied from Matlab.

**Question 3:** What is the output from the command: `real(ans)`.

## 7 Output

If you get sick of Matlab always repeating everything, you can stop it producing the output of a particular command by ending the line with a ";" so

```
>> x=6;
```

assigns the number 6 to  $x$  without the usual output. Matlab is also clever enough to recognize integers sometimes:

```
>> cos(pi)
ans = -1
```

**But it does not always get it right:**

```
>> acos(-1) * exp(i * pi)/pi
ans = -1.0000 + 0.0000i
```

Matlab also uses customary scientific notation:

```
>> cos(1.57)
ans = 7.9633e-04
```

To obtain more significant digits, execute the command:

```
>> format long
>> cos(1.57)
ans = 7.963267107332633e-04
```

The command

```
>> format short
```

returns the output format to the default. Matlab lets you know if you screwed up:

```
>> 1/0
Warning: Divide by zero.
ans = Inf
```

NOTE: If the warning does not display then enter the command: `warning('on','all')` to turn the display of warnings ON.

---

## 8 Variables

You can create variables in Matlab by assigning numerical values to variable names:

```
>> x=3
```

assigns the value of 3 to  $x$ , and Matlab remembers this until  $x$ 's value is re-assigned by another command or erased. For example,

```
>> 25*x
ans = 7.5
```

Also, simply typing the variable name produces the value Matlab thinks it has:

```
>> x
x = 3
```

To clear the values assigned to a variable, you can use the command `clear`. For example, “`clear x`” erases the assignment  $x = 3$ .

Variables can be assigned to the results of computations. For example,

```
>> y=(2+x) ^ 3
y = 125
```

assigns the value  $(s + x)^3 = 5^3 = 125$  to  $y$ .

Some variables are already assigned in Matlab: `pi` stands for  $\pi = 3.14159\dots$  and `i` is  $i = \sqrt{-1}$ . So

```
>> z=2 * pi + 6 * sqrt(-1)
```

**Question 4:** What is the answer to the above assignment?

## 9 Mathematical Functions

Matlab has a large number of built-in mathematical functions. Some of these are:

<code>abs(x)</code>	absolute value of $x$ , $ x $
<code>sqrt(x)</code>	square root of $x$ . $\sqrt{x}$
<code>sign(x)</code>	sign of $x$
<code>exp(x)</code>	$e^x$
<code>log(x)</code>	$\ln(x) = \log_e(x)$
<code>log10(x)</code>	$\log_{10}(x)$
<code>sin(x)</code>	$\sin(x)$
<code>cos(x)</code>	$\cos(x)$
<code>tan(x)</code>	$\tan(x)$
<code>asin(x)</code>	$\sin^{-1}(x)$
<code>acos(x)</code>	$\cos^{-1}(x)$
<code>atan(x)</code>	$\tan^{-1}(x)$
<code>sinh(x)</code>	$\sinh(x)$
<code>cosh(x)</code>	$\cosh(x)$
<code>tanh(x)</code>	$\tanh(x)$
<code>asinh(x)</code>	$\sinh^{-1}(x)$
<code>acosh(x)</code>	$\cosh^{-1}(x)$
<code>atanh(x)</code>	$\tanh^{-1}(x)$
<code>real(z)</code>	Real part of $z$
<code>imag(z)</code>	Imaginary part of $z$

Matlab assumes that the argument is expressed in radians for the trigonometric functions. *e.g.*

```
>> x=6
x=6
>> z=sin(x)-log(cos(x))
z= -0.2388
```

computes  $\sin(x) - \log_e(\cos(x)) = \sin(x) - \ln(\cos(x))$  for  $x = 6$ .

**Question #5:** Find the absolute value and argument of the complex number,  $x = z \sin(z^z)$ , where  $z = 2 + i$ . (*Note:* The argument is the angle formed by the graph of a complex number on the *(Real x Imaginary)* axes.)

**Question #6** Compute the value of the function  $e^{(\sinh\sqrt{y \sin(x^2)} - \cosh\sqrt{y \cos(x^2)})} - 1$  at  $x = 14$  and  $y = \ln(3)$ .

**Question #7:** Compute:  $y = e^{-\sqrt{x}} \left[ \sin(x) + \frac{28}{3x} \right]$  for  $x = 12$ .

(a) Check your answer by computing,  $\sin^{-1}(y \exp(\sqrt{x}) - 28/(3x)) + 4\pi$

(b) Why did we put in the  $4\pi$ ?

---

---

**Quit** MATLAB by clicking on the **File** menu in the upper left corner and choosing **Exit**. Please remember to **Log Off** (from the “Start” menu in the lower left of the screen).