MATLAB: LECTURE NOTES

A. Hakan Argeşo

Received:

The Date

Abstract. *Replace this text with your own abstract.*

1980 AMS Subject Classification: the subject classification

Keywords and Phrases: the kew words and phrases

1. Introduction

1.1. What is MATLAB (Matrix Laboratory)?

MATLAB is an interactive computer language used for technical and scientific computing. It is available for many computer systems including MS windows, Linux and Macintosh. MATLAB integrates computation, visualization and programming from a user friendly (easy to use) environment. The user can solve problems and obtain their results through interactive sessions in a familiar mathematical notation.

MATLAB was first developed in the 1970's for the applications involving matrices, linear algebra and numerical analysis. Throughout, the years the program's capabilities have been greatly extended and today some of the typical uses of MATLAB include

- Maths and computation
- Scientific and Engineering graphics
- Algorithm development
- Modelling and simulation
- Data analysis and visualization
- Application development

Among many others, there are two very important features for using MATLAB. First, the basic data element of MATLAB is an array element that does not require dimensioning. This allows us to solve problems especially those involving matrix and vector formulations in a very short period of time, compared to developing a program to solve the same problem in traditional programming languages such as C, Fortran and Pascal. Second, there is no distinction among real, complex and integer numbers. All numbers are in double precision, which means, MATLAB connects all kinds of numbers continuously and any variable can take any type of number without special declaration in programming. This feature makes programming development faster.

These notes cover MATLAB version 6 for MS windows which was released in the fall of 2000.

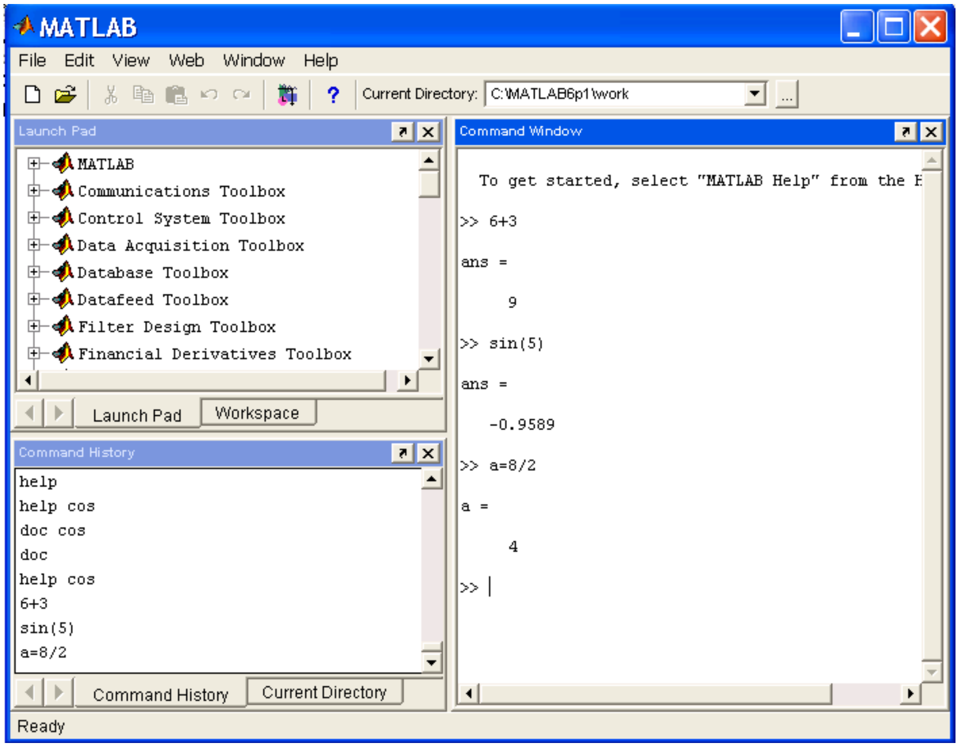
1.2. Starting MATLAB and Quitting MATLAB

Starting MATLAB: On MS windows double click on MATLAB icon

Quitting MATLAB: Select **Exit Matlab** from the **File Menu** in the desktop, or type `quit` in the Command Window

2. Matlab Desktop

When you open Matlab program the MATLAB desktop will be displayed as shown in the figure below. The desktop contains seperate tools that are used for file operations, variables, applications, visualizations associated with MATLAB.Note that, Figure 1 is the default appearence of MATLAB desktop, which contains three seperate windows(Command Window, Launch Pad, Command History). The appearence of the desktop can be changed according to your needs. We can resize, open, move, close the windows of desktop. The windows appearing in the desktop can also be moved outside the desktop (undocking).

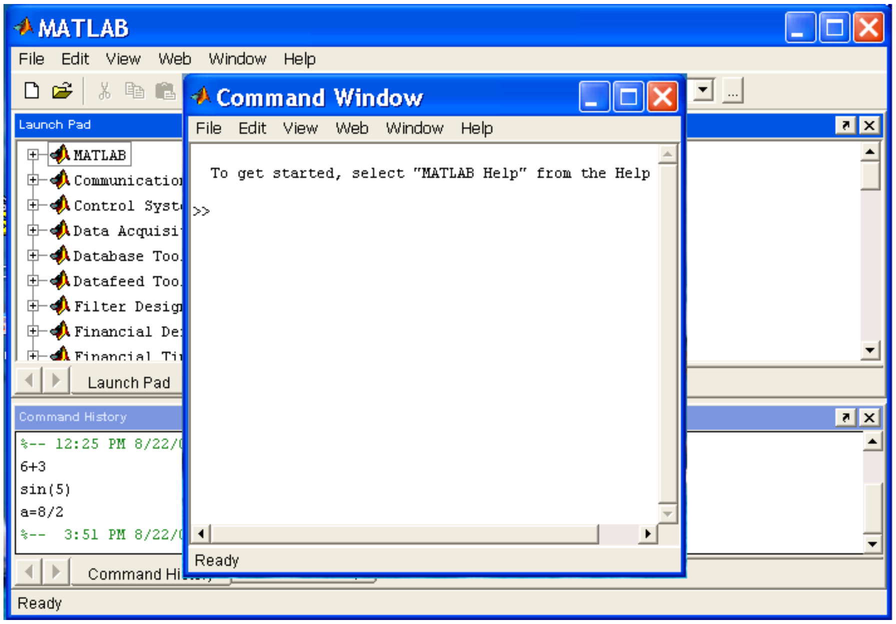


```
help
help cos
doc cos
doc
help cos
6+3
sin(5)
a=8/2
```

```
  To get started, select "MATLAB Help" from the H
>> 6+3

ans =

     9

>> sin(5)

ans =

   -0.9589

>> a=8/2

a =

     4

>> |
```

Figure 1: The default Matlab desktop

Example 1. *To undock (move window outside MATLAB desktop) the Command Window click on the button containing arrow on the upper right corner of Command Window (See Figure 2).*



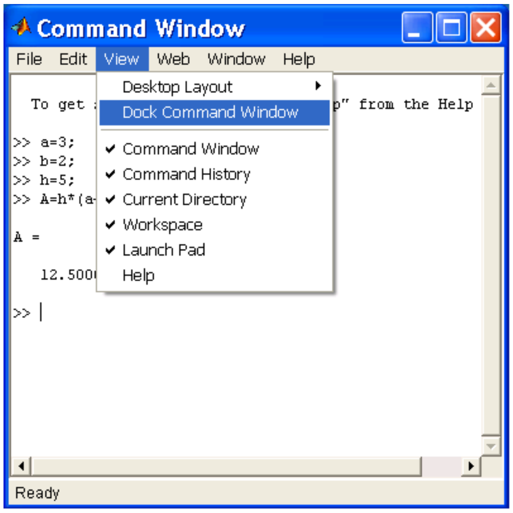
```
%-- 12:25 PM 8/22/(
6+3
sin(5)
a=8/2
%--  3:51 PM 8/22/(
```

```
  To get started, select "MATLAB Help" from the Help
>>
```

Figure 2: Undocking Command window

If you want to move an undocked window inside the desktop (docking), choose **Dock *name* Window** from the **View** menu where ***name*** is the name of Window that you are docking.

Example 2. *To dock the Command Window, click the* ***Dock Command Window*** *from the view menu.*



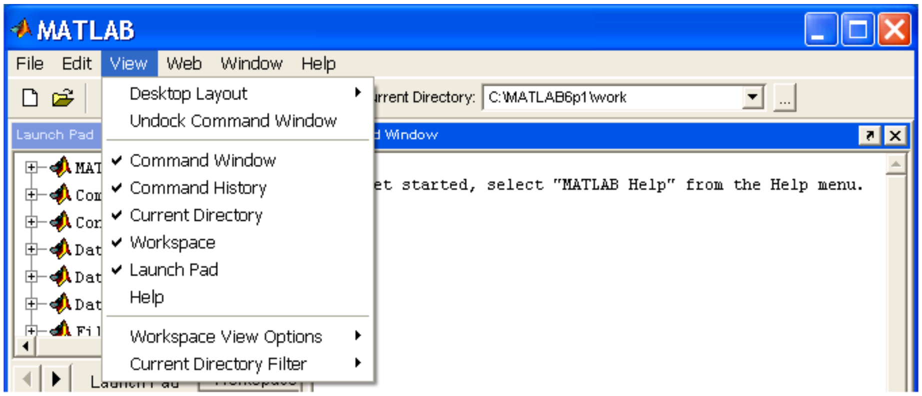
```
  To get                         p" from the Help

>> a=3;
>> b=2;
>> h=5;
>> A=h*(a-

A =

    12.500

>> |
```

Figure 3: Docking Command Window

The MATLAB desktop tools are listed below

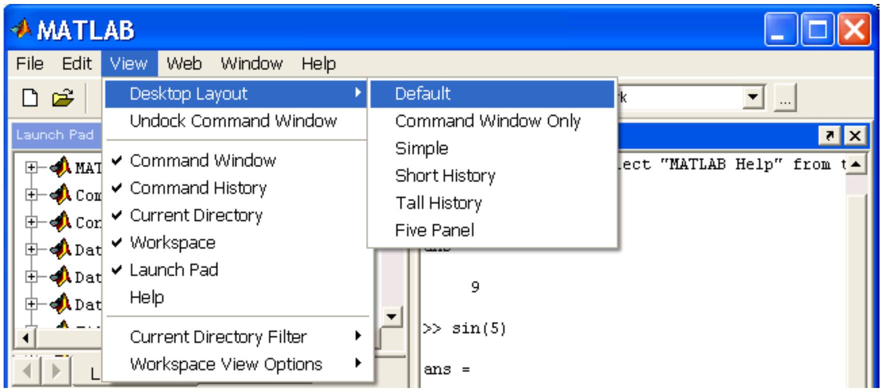
- Command Window
- Command History
- Current Directory Browser
- Workspace Browser
- Launch Pad
- Help Browser
- Array Editor
- Editor/Debugger

The first 6 items can be displayed from the **View** menu as shown in Figure 4.When an item is selected a check appears at the left hand side, which means the item is currently in use. You can also select predefined appearences of MATLAB desktop from the **Desktop Layout** in the **View** menu. Choosing the **Default** Layout from Desktop returns the apparence shown in Figure 1.



```
et started, select "MATLAB Help" from the Help menu.
```

Figure 4: Displaying Desktop tools



```
ect "MATLAB Help" from t

       9

>> sin(5)

ans =
```

Figure 5: Selecting Desktop appearences

2.1. Command Window

Command Window is used to communicate with MATLAB by typing commands, entering variables, running functions. MATLAB displays the prompt >> to indicate that it is ready to take input. To multiply 2 and 6, type 2*6 and press Enter (be sure the cursor is at the prompt in the Command window).

```
>>2*6
ans =
     12
```

The answer is stored into a variable called `ans`, which is the variable name for MATLAB to hold the answer to latest calculation. We can use `ans` in furher calculations such as

```
>>3+ans
ans =
     15
```

Note that the value stored in `ans` is changed to `15`. You can also assign mathematical expressions to variable names that you choose. For example, type x=5+2*7 (`=` is the assignment operator)

```
>>x=5+2*7
ans =
     19
```

MATLAB stores the scalar value `19` to a variable name `x`.You can use `x` in your further calculation such as

```
>>x=38/x
ans =
     2
```

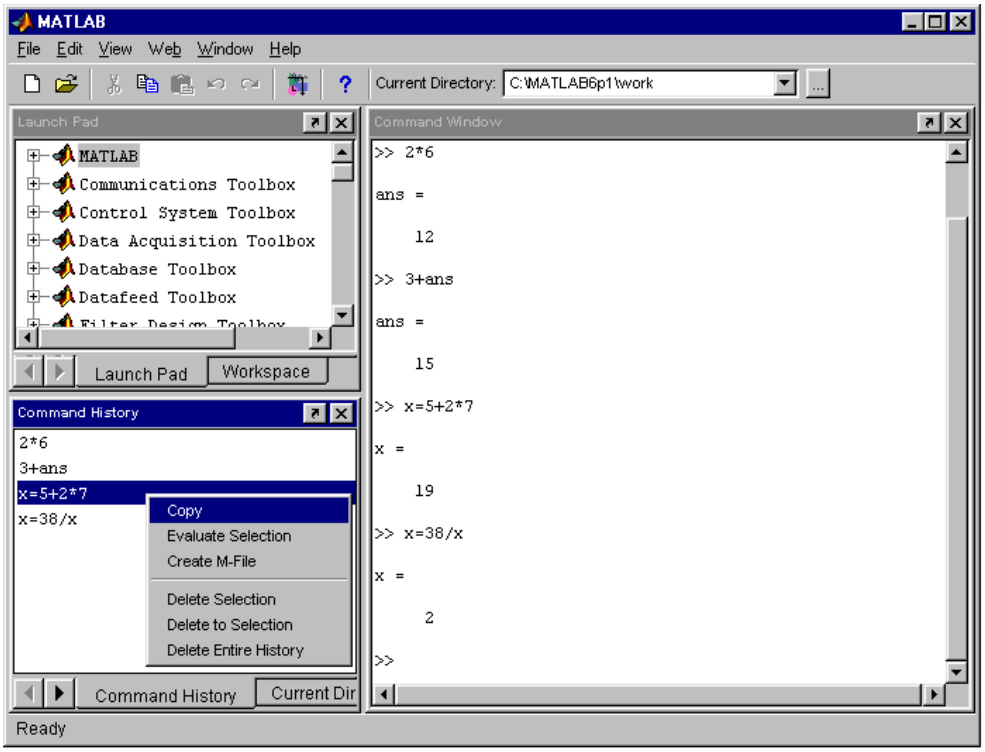
In MATLAB, variable names are case sensitive as in C. The names `xsum` and `Xsum` correspond to different variables. Variable names have to start with a letter and can contain up to 31 characters. If the name contains more than 31 charaters, the characters beyond the 31 st are ignored.

If you make a typing mistake when you are entering commands, you can correct the command by using Backspace, Delete keys and by using arrow keys ←− and −→. The previous keystores can be retained with the arrow keys ↑ and ↓.

2.2. Command History

In a session each line you typed is stored in Command History window. You can view the previously executed lines from the command history and copy, execute, these lines.

Example 3. *To copy a previously executed line, first, select the line from the Command History and right click on it (see Figure 3). A pop up menu opens. From that menu select copy. Next, click on the Command window and paste it into the desired location.*



```
2*6
3+ans
x=5+2*7
x=38/x
```

```
>> 2*6

ans =

    12

>> 3+ans

ans =

    15

>> x=5+2*7

x =

    19

>> x=38/x

x =

     2

>>
```

Figure 6: Copying a previously executed line from Command History

If you double click on a line in the Command History, it will automatically be executed in the Command Window.

Scalar aritmetic operations. A scalar is a single number. A scalar variable is variable that contains a single number. The aritmetic operations in mathematical expressions that MATLAB performs on scalars, are given in the table below. Mathematical expressions are always evaluated from left to right. You can use paranthesis in the expressions.

Mathematical Operation	MATLAB Symbol	MATLAB Form
Exponentiation a^b	^	a^b
Multiplication ab	*	a*b
Right division $a/b = \frac{a}{b}$	/	a/b
Left division $a\backslash b = \frac{b}{a}$	\	a\b
Addition $a + b$	+	a+b
Subtraction $a - b$	-	a-b

The next table shows the order of precedence of mathematical expressions from highest to lowest

Paranthesis beginning from the innermost pair	
^	:Exponentiation
*, /	:Multiplication, division
+,-	:Addition, subtraction

Example 4. *Evaluate* $\frac{5\times6}{2}+17$

```
>>5*6/2 + 17
ans =
     32
```

Example 5. *Evaluate* $8^2\times7-\frac{3\times11+7}{6-1}$

```
>>8^2*7 - (3*11 + 7)/(6 - 1)
ans =
     440
```

Example 6. *Evaluate* $27^{1/3}\times2-81^{0.5}+3\left(17+18+19\right)$

```
>>27^(1/3)*2 - 81^0.5 + 3*(17+18+19)
ans =
     159
```

Example 7. *Evaluate* $\frac{0.5\left(\left(2^3+1\right)^2+15\right)^2}{8}$

```
>>0.5*( (2^3+1)^2 + 15)^2/8
ans =
     576
```

Note that, in writing the expressions for the above examples, we use spaces to improve readability. MATLAB ignores the spaces before and after the assignment operator '=' when making its calculations. The expressions

```
>>x=8/2-3*(7+1)
>>x = 8/2 - 3*(7 + 1)
```

give the same result, however reading the second expression is much more easier than the first.

Assignment operator. In MATLAB, sign = is used as an assignment operator. Typing `x = 12` assigns the value 12 to the variable name x. MATLAB assigns the value which is calculated on the right hand side of operator = to a variable name on the left hand side. The opposite is not valid, that is, writing `12 = x` produces an error in MATLAB. We can use expressions like `x = x + 12` provided that x has been assigned a value before writing this expression. There must be only one variable on the left side of the assignment operator. Writing `x + 12 = x` is invalid in MATLAB.

Example 8. *Calculate the area of a trapezoid which is given by the formula* $A = h\frac{(a+b)}{2}$*, where* a *and* b *are the side lengths and* h *is the base length of the trapezoid. In our problem we choose* $a = 3$*,* $b = 2$ *and* $h = 5$*.*

```
>> a = 3;
>> b = 2;
>> h = 5;
>> A = h*(a+b)/2
A =
   12.5000
```

Notice that, we use semicolon ';' in the end of the first three expressions. In MATLAB, semicolon is used to suppress printing on the screen. In Example 8, the values $3, 2$ and 5 are assigned to the variables a, b and h respectively, but the results are not printed on the screen. We can write several commands on the same line if we seperate them with comma ',' and semicolon ';'. The previous example can also be entered in the form

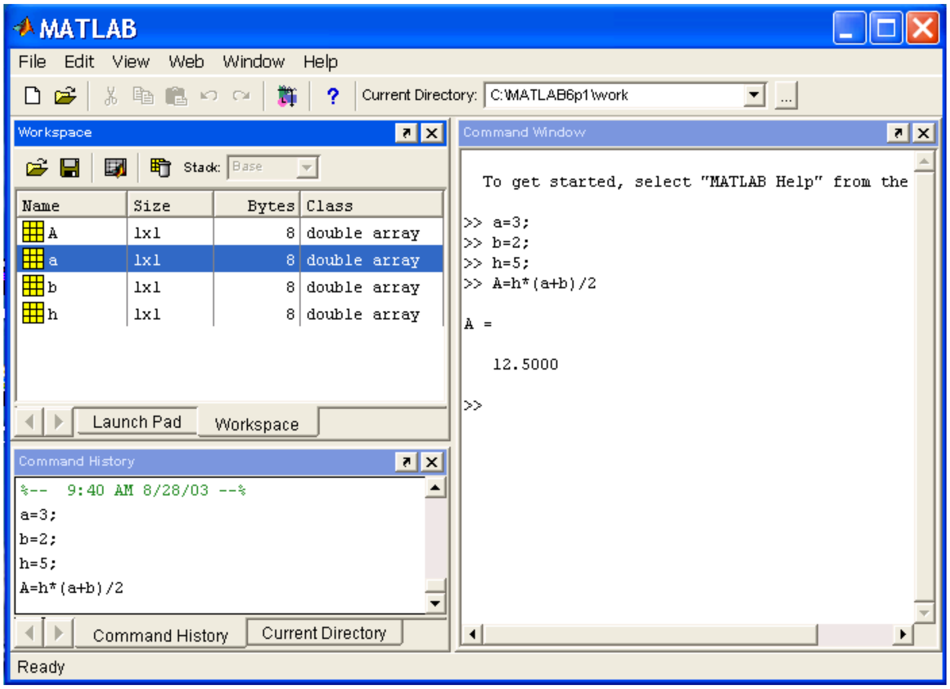
```
>> a = 3; b = 2; h = 5; A = h*(a+b)/2
A =
   12.5000
```

If we use commas instead of semicolons the output will be

```
>> a=3, b=2, h=5, A=h*(a+b)/2
a =
   3
b =
   2
h =
   5
A =
   12.5000
```

2.3. Workspace browser

In MATLAB, the term workspace refers to the names and values of variables that are stored in memory during the current working session. The information about each variable in the workspace can be obtained from the Workspace Browser. To open the Workspace Browser, select **View** menu in the desktop and choose **Workspace**. When the Workspace Browser is opened, you see the names, sizes and classes of each variable that MATLAB uses in the current session. The next figure shows the appearence of Workspace Browser, assuming that the workspace contains the variables for the previous example.



Name	Size	Bytes	Class
A	1x1	8	double array
a	1x1	8	double array
b	1x1	8	double array
h	1x1	8	double array

```
%-- 9:40 AM 8/28/03 --%
a=3;
b=2;
h=5;
A=h*(a+b)/2
```

```
  To get started, select "MATLAB Help" from the

>> a=3;
>> b=2;
>> h=5;
>> A=h*(a+b)/2

A =

    12.5000

>>
```

Figure 7: Workspace browser

You can change the values of the variables or delete them from the memory using Workspace Browser. We will cover the use of Workspace browser in more detail in the following chapters.

Managing the working Session in MATLAB. MATLAB uses some commands for managing the working session. In the following table, we introduce some of these commands and give examples on how to use them.

Command	Description
`who`	:Lists variables that are currently stored in the memory
`whos`	:Lists variables that are currently stored in the memory
	by giving their sizes and classes
`clear`	:Removes all the variables that are stored in the memory
`clear var1 var2 var3`	:Removes the variables `var1 var2` and `var3` from the memory
`clc`	:Clears the command window (the variables are still in the memory)

Example 9. *List the current variables for Example 8 using* `who` *command*

```
>> who
Your variables are:
A a b h
```

Example 10. *List the current variables and get their details for Example 8 using* `whos` *command*

```
>> whos
  Name      Size           Bytes  Class

  A         1x1                8  double array
  a         1x1                8  double array
  b         1x1                8  double array
  h         1x1                8  double array

Grand total is 4 elements using 32 bytes
```

Example 11. *Remove variable a from the memory and then list the current variables*

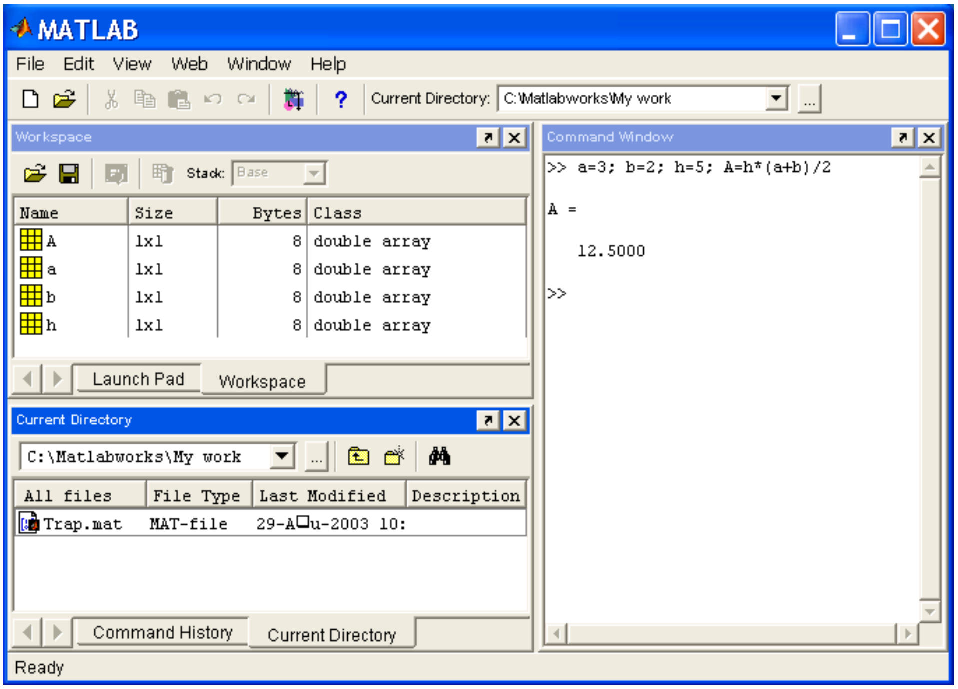
```
>> clear a
>> who
Your variables are:
A  b  h
```

2.4. Current Directory browser

While working with MATLAB, we need to save, open, search and view files. Therefore, it is important to know the files you use with MATLAB. MATLAB file operations use *current directory* and *search path* as reference points. Any file you must run must be either in the *current directory* or on the *search path. Current directory* can be seen from Current Directory field on top of the desktop or on top of Current Directory Browser as shown in Figure 8. MATLAB uses Current Directory Browser for file operations. When you open Current directory browser, you see a list of files that exist in the current directory. Suppose you want to view files in the directory named **Mywork**. First, you have to change the current directory to **Mywork**. This can be done either typing `cd C:\Mywork` in the Command Window or by using browser on the right hand side of Current Directory field. You can also display the current directory by typing `pwd` from the command window. The table below summarizes some directory commands.

`pwd`	Displayes the current direntory
`dir`	Lists all files in the current directory
`dir dirname`	Lists all files in the directory `dirname`
`cd dirname`	Changes the current directory to `dirname`

The search path is a set of directories that MATLAB uses to find MATLAB related files. The use of the search path and its related commands will be covered in later chapters.



Current Directory: C:\Matlabworks\My work

```
>> a=3; b=2; h=5; A=h*(a+b)/2

A =

   12.5000

>>
```

Name	Size	Bytes	Class
A	1x1	8	double array
a	1x1	8	double array
b	1x1	8	double array
h	1x1	8	double array

C:\Matlabworks\My work

All files	File Type	Last Modified	Description
Trap.mat	MAT-file	29-A□u-2003 10:	

Figure 8: Current Directory Browser

2.5. Getting help from MATLAB

One of the most unique features of MATLAB is its help system. There are two ways to get help from MATLAB. The first one is to use help commands from the Command Window and the other one is to use Help Browser. If you want to get a concise explanation of command, type `help commandname` in the Command Window.

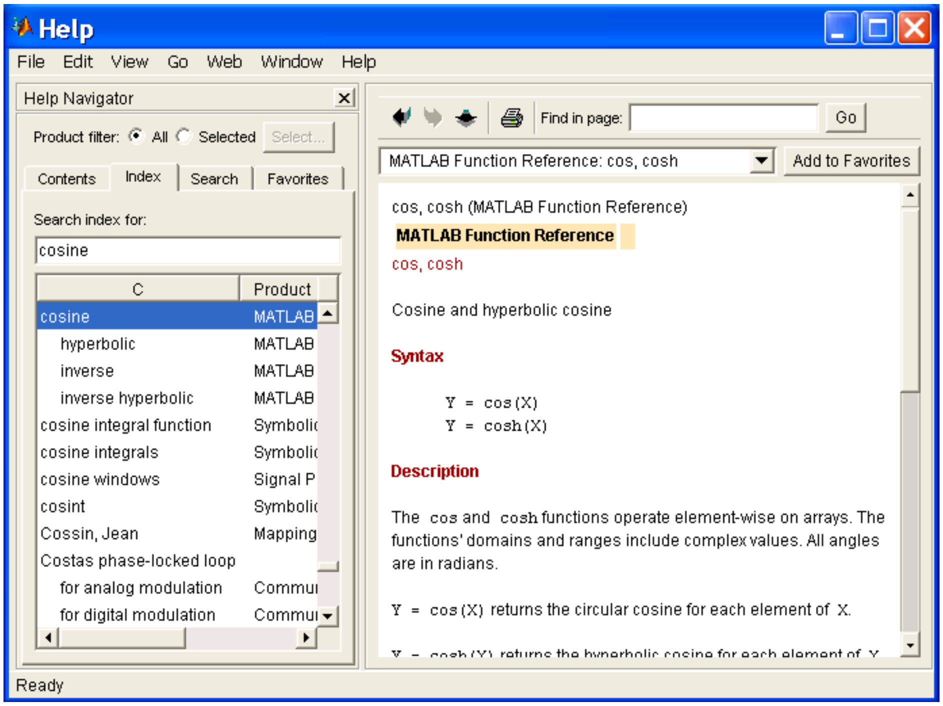
Example 12. *Getting help about the cos (cosines) function from MATLAB using* `help` *command*

```
>> help cos

 COS    Cosine.
    COS(X) is the cosine of the elements of X.

 Overloaded methods
    help sym/cos.m
```

More detailed information about a command or subject can be obtained from the Help Browser. To open Help Browser, select **Matlab Help** from the **Help** menu on the desktop. The help browser contains 2 windows: the Display Pane and Help Navigator. There are 4 Tabs on the Help Navigator:

- Contents Tab is used to list the titles and table of contents for all MATLAB documentation
- Index Tab is used to find keywords from all MATLAB documentation
- Search Tab is used to find all MATLAB documents containing a specified phrase
- Favorites Tab is used to view a list of documents you previously designated as favorites

Example 13. *To get help about cosinus function from MATLAB using help menu, first open the* ***Help Browser****, and click* ***Index*** *tab in the Help Navigator window. Next, type cosine into the search index box. As you type, the index displayes the maching items. Select* ***cosine*** *by clicking on it. As the item is selected, a detailed documentation about that item is displayed in the display pane. (See the figure below)*


Help
File Edit View Go Web Window Help
Help Navigator
Product filter: ● All ○ Selected Select...
Contents Index Search Favorites
Find in page:
Go
Add to Favorites
Ready

Search index for: cosine

MATLAB Function Reference: cos, cosh

C	Product
cosine	MATLAB
hyperbolic	MATLAB
inverse	MATLAB
inverse hyperbolic	MATLAB
cosine integral function	Symbolic
cosine integrals	Symbolic
cosine windows	Signal P
cosint	Symbolic
Cossin, Jean	Mapping
Costas phase-locked loop	
for analog modulation	Commun
for digital modulation	Commun

cos, cosh (MATLAB Function Reference)

MATLAB Function Reference

cos, cosh

Cosine and hyperbolic cosine

Syntax

```
Y = cos(X)
Y = cosh(X)
```

Description

The cos and cosh functions operate element-wise on arrays. The functions' domains and ranges include complex values. All angles are in radians.

Y = cos(X) returns the circular cosine for each element of X.

Y = cosh(X) returns the hyperbolic cosine for each element of X.

Figure 9: Help Browser

The documentation about a command can also be displayed by typing `doc commandname` in the command window. If you type

```
>> doc cosine
```

the Help Browser opens and displays the same documentation as in the case of the previous example. Notice that, in order to use `help` command you have to know the exact spelling of the command. If you type

```
>> help cosine
```

MATLAB gives you a warning message as below

```
>> help cosine

cosine.m not found.
```

because there is no function definition spelled as cosine in MATLAB.

Another important function is `lookfor` function. This function allows us to search functions based on a keyword. When you type `lookfor keyword` in the command window, MATLAB displays all functions that contain the `keyword`.

Example 14. *Search all functions that contain the keyword "tangent" using lookfor function*

```
>> lookfor tangent
ACOT   Inverse cotangent.
ACOTH  Inverse hyperbolic cotangent.
ATAN   Inverse tangent.
ATAN2  Four quadrant inverse tangent.
ATANH  Inverse hyperbolic tangent.
COT    Cotangent.
COTH   Hyperbolic cotangent.
TAN    Tangent.
TANH   Hyperbolic tangent.
DTANSIG Hyperbolic tangent sigmoid transfer derivative function.
TANSIG Hyperbolic tangent sigmoid transfer function.
ACOT   Symbolic inverse cotangent.
ACOTH  Symbolic inverse hyperbolic cotangent.
ATAN   Symbolic inverse tangent.
ATANH  Symbolic inverse hyperbolic tangent.
COT    Symbolic cotangent.
COTH   Symbolic hyperbolic cotangent.
TAN    Symbolic tangent function.
TANH   Symbolic hyperbolic tangent.
BLKATAN This block defines an output angle that is the arctangent of two inputs.
BLKCOSATAN This block defines the cosine of an angle whose tangent is u1/u2.
BLKSINATAN This block defines the sine of an angle whose tangent is u1/u2.
```

An efficient way, for getting help from MATLAB can be performed in two steps

1. Use `lookfor` function to list all functions that contain a certain keyword.
2. From that list, select the function name that you are looking for and then use `help` or `doc` command to get detailed information about that function or command name.

2.6. Menus and Toolbar

As we discussed before MATLAB has different type of windows such as *Command window*, *Command History*, *Launch Pad*, *Workspace Browser* and *Current Directory browser.* On the top of Desktop there is a menu bar and a toolbar which contains icons and current directory field. There are also other windows in MATLAB that appear when you plot graphs or when you write your MATLAB programs. Each window type has a menu bar and the contents of these menus can be changed depending on which window you are using.

Below, we summarize the use of desktop menus when the *Command Window* is active. The desktop contains 6 different menus if you activate the Command Window. These are

- **File**
- **Edit**

- **View**
- **Web**
- **Window**
- **Help**

menus. You can activate one of these menus by clicking on it. After clicking, MATLAB brings a list of items. You can select one of the items by clicking on it and MATLAB performs the operation. When you examine the items that appear when a menu is opened, you see that, some of the item names are followed by three dots "**...**". This means that, if you select that item a *submenu* or another window will be opened.

File menu in MATLAB 6. The file menu is used to perform file operations in MATLAB. Each item and its use are listed in the table below.

New	Opens a dialog box that allows you to create a new program file, a new Figure or Model File.
Open	Opens a dialog box that allows you to select a file for editing.
Close Command Window	Closes the command window.
Import Data	Starts the Import Wizard that allows you to import data easily.
Save Workspace As...	Opens a dialog box that enables you to save a file.
Set Path...	Opens a dialog box that enables you to set MATLAB search path.
Preferences...	Opens a dialog box that enables you to set preferences for such items as fonts, colors, tab spacing....
Print...	Opens a dialog box that enables you to print all of the Command Window.
Print Selection...	Opens a dialog box that enables you to print selected portions of the Command Window.
File List	Contains a list of previously used files, in the order of most recently used.
Exit MATLAB	Closes MATLAB.

Edit menu in MATLAB 6. The items and their use in the edit menu are given in the table below

Undo	Reverses the previous editing action.
Redo	Reverses the previous Undo operation.
Cut	Removes the selected text and stores it for pasting later.
Copy	Copies the selected text for pasting later, without removing it.
Paste	Inserts any text on the clipboard at the current location of the cursor.
Paste Special...	Inserts the contents of the clipboard into the workspace as one or more variables.
Select All	Highlights all text in the Command Window.
Delete	Clears the highligted text from the Command Window.
Clear Command Window	Removes all text from the Command Window.
Clear Command History	Removes all text from the Command History window.
Clear Workspace	Removes the values of all variables from the workspace.

View menu in MATLAB 6. View menu is used to control the appearence of MATLAB. The table below list the items and summarizes their use.

Desktop Layout	used to select the predefined appearences of MATLAB, where there are six options: Default, Command Window Only, Simple, Short History, Tall History and Five Panel.
Undock Command Window	Undocks Command Window (See Example 1).
Command Window	Opens or closes Command Window
Command History	Opens or closes Command History
Current Directory	Opens or closes Current Directory Browser
Workspace	Opens or closes Workspace Browser
Launch Pad	Opens or closes Launch Pad
Help	Opens or closes Help Browser
Workspace View Options	Arranges the appearence of Workspace Browser

Note that when you select a window name from the **View** menu, a check appears in front of the name of window.

Web menu in MATLAB 6. Web menu is used to access the MathWorks (the owner of MATLAB) support site. You can get information about the other products of MathWorks and also you can find technical support and updates about MATLAB.

Window menu in MATLAB 6. The Window menu has one or more items depending on what you have done in your session. Click the name of the window that appears on the menu to open it.

Help menu in MATLAB 6. The Help menu is explained in the previous sections.

The use of these menus will be explained in a more detailed form by giving some example in the following sections.

2.7. Saving and Loading your Workspace variables

Recall that, the term workspace refers to the names and values of variables that are stored in memory during the current working session. You can save your workspace variables in files called *mat.* files (The extension of these files are `.mat`) . *Mat* files are binary files that can only be read by MATLAB and they contain the names, types, sizes and values of your workspace variables. Saving workspace variables allows you to continue your working session at a later time. There are two ways for saving workspace variables

1. Using `save` command from the Command Window

   ∗ If you type `save` from the Command Window, MATLAB saves your workspace variables in a file called `matlab.mat`. You can also specify the name of file by typing `save filename` where `filename` is the name of the file. It should be recalled that MATLAB saves these files in *current directory* which can be seen from the *Current Directory Field.*

   Example 15. *Save your workspace variables into a file named "Mywork"*

```
>> save Mywork
```

2. Selecting the item **Save Workspace As...** from the **File** menu

   ∗ When you select **Save Workspace As...** item, a window appears from which you can specify the name and the location of the file that is to be saved.

When needed, an existing *.mat* file can be opened and the workspace variables of that *.mat* file can be loaded into your session. Before loading your workspace variables, be sure that the location of currect directory is correct. There are three ways of loading workspace variables.

1. Using `load` command from the Command Window

   ∗ If you type `load` from the Command Window, MATLAB loads your workspace variables from the file `matlab.mat`. Recall that, `matlab.mat` is the default filename that MATLAB gives when you use `save` command without specifying filename. Typing `load filename` loads the workspace variables in the file `filename.mat`.

   Example 16. *Load the workspace variables in the file named "Mywork"*

```
>> load Mywork
```

2. Selecting the **Open** item from the file menu

   ∗ Click **Open**, and go to the location of the *mat* file you want to load. Select the *mat* file you want to open and press the Open button or double click on the file.

3. Using Current Directory Browser

   ∗ From the list in the Current Directory Browser, if you double click on the a mat file, MATLAB loads the workspace variables of that file to your working session.

2.8. Recording your session

In your working session, you can record all your activities in a file. Typing `diary on` from the Command Window, starts recording your keystrokes and most of the screen output to a file named `diary`. It should be noted that, this record will not include your graphs. Typing `diary off` terminates recording your activities.

In MATLAB, this kind of recording is called keeping *diary.* The status of the *diary* can be `on` or `off` depending on whether MATLAB is recording your activities or not. When we type `diary on`, the status of *diary* becomes `on` and recording starts. On the other hand, when we type `diary off`, the status of the *diary* becomes `off` and MATLAB terminates recording. If you type `diary off` to stop recording and later type `diary on`, MATLAB appends the remainder of the session to the end of file.

You can change the status of the *diary* by typing `diary` itself. This means, if the status of the *diary* is `on`, typing `diary` makes the status of the *diary* `off`, or if the status of the *diary* is `off`, typing `diary` makes the status of *diary* `on`.

You can also keep your diary in a file with a filename you choose by typing `diary filename`.

To see the status of the *diary,* type `get(0,'diary')` and MATLAB will indicate the diary status by returning either `on` or `off`.

An Important Note

You can not view your current recordings while the diary status is `on`. In order to view these recordings first you must change the diary status to `off` state.

Example 17. *Getting the diary status*

```
>> get(0,'diary')
ans =
off
```

To see the diary filename type `get(0,'diaryfile')`, and MATLAB returns the name off the diary filename

Example 18. *Getting the diary filename*

```
>> get(0,'diaryfile')
ans =
diary
```

Note that, the *diary* files are ASCII text files which means that, with any type of word processor, you can read, edit and print these files. At this point, we introduce MATLAB's text editor which is called *Editor/Debugger. Editor/Debugger* is a text editor which is used to write, edit and debug MATLAB programs. It can also be used to write and edit text files such as *diary* files. If you want to open a *diary* file from MATLAB, you have two options

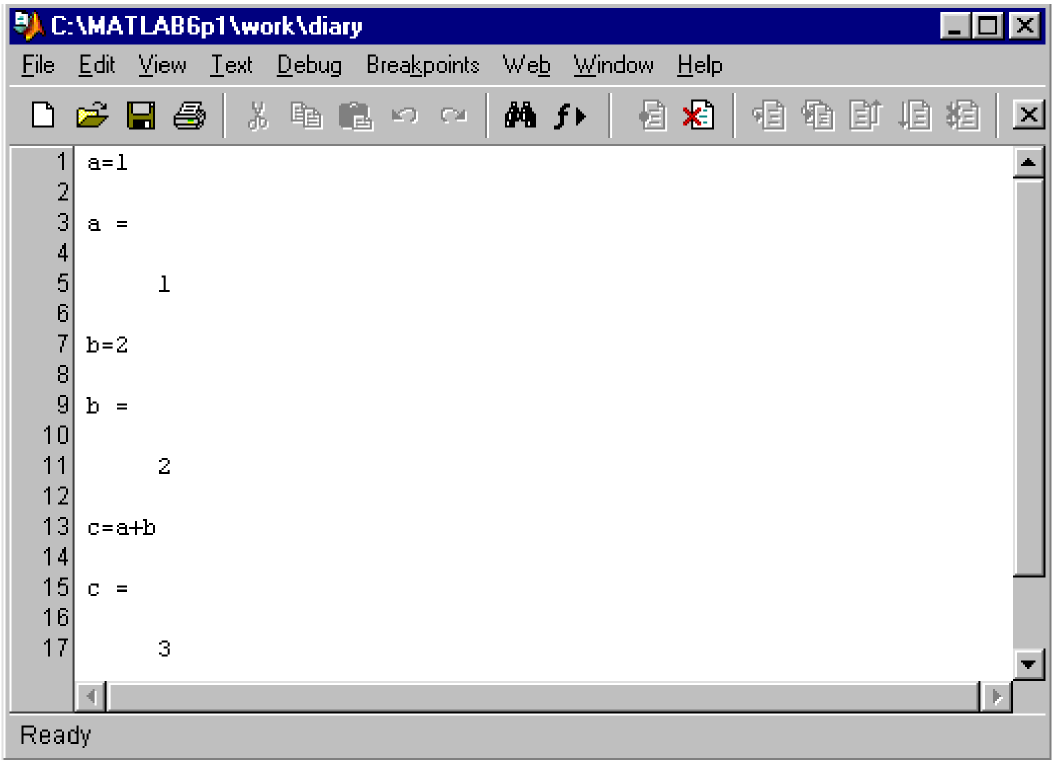
- Select **Open** from the **File** menu and go to the location of the *diary* file. Change **All MATLAB files** to **All Files (*.*)** in the **File Type** field. Now you can see all the files that are stored in the directory. Select the *diary* file and press **Open**.

- Double click the name of the *diary* file from the list in the Current Directory Browser.

Example 19. *Lets assume we record the activities shown below in a file named* `diary` *and we want to view this file using MATLAB Editor/Debugger*

```
>> diary
>> a=1

a =

     1

>> b=2

b =

     2

>> c=a+b

c =

     3

>> diary off
>>
```

When you use one of the produres explained above, the *Editor/Debugger Window* will be opened as shown belowBy using this window, you can edit the diary file and perform file operations such as printing.



```
a=1

a =

     1

b=2

b =

     2

c=a+b

c =

     3
```

Figure 10: