Chapter 1

Introduction

MATLAB (**Mat**rix **lab**oratory) is an interactive software system for numerical computations and graphics. As the name suggests, MATLAB is especially designed for matrix computations: manipulating arraays, solving systems of linear equations, computing eigenvalues and eigenvectors, factoring matrices, and so forth. In addition, it has a variety of graphical capabilities: plotting data, creating animations, creating user interfaces, and can be extended through programs written in its own programming language.

MATLAB is designed to solve problems numerically, that is, in finite-precision arithmetic. Therefore it produces approximate rather than exact solutions, and should not be confused with a symbolic computation system (SCS) such as Mathematica or Maple. It should be understood that this does not make MATLAB better or worse than an SCS; it is a tool designed for different tasks and is therefore not directly comparable.

In the following sections, we give an introduction to some of the most useful features of MATLAB. We include plenty of examples; the best way to learn to use MATLAB is to read this while running MATLAB, trying the examples and experimenting.

1.1 Starting MATLAB

Microsoft Windows: Double -click on the MATLAB shortcut icon on the Windows desktop, or find it under Start - Programs - Math & Numerical Metehods - Matlab 6.5 - Matlab 6.5
Linux: Type `MATLAB` at the operating system prompt.

1.2 Quitting MATLAB

To end your MATLAB session, select **`Exit MATLAB`** from the **`File`** menu in the desktop, or type `quit` in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a `finish.m` script.

1.3 MATLAB Environment

When you start MATLAB, a special window called the MATLAB desktop appears, Figure 1.1. It integrates many tools for managing files, variables, applications, and debugging within the MATLAB environment. This is what is known as an integrated development environment (IDE).

1.4 The Desktop

The major tools within or accessible from the desktop are:

Window	Purpose
Command Window	Main window: for running functions and entering variables
Command History	Logs previously run commands in the Command Window
Launch Pad	Provides access to tools, demos, and documentation
Help Browser	Provides help for accessing documentation
Current Directory	Browser for accessing files
Workspace	Provides information about the variables that are used
Editor/Debugger	Helpful in creating, modifying, and debugging MATLAB script and function files (M-files)
Figure Window	Contains output from graphic commands
Array Editor	Browser for editing arrays

1.4.1 Command Window

Use the command window to enter variables and run functions and M-files. Type the functions and variables at the MATLAB prompt (>>) and they will be executed on the spot

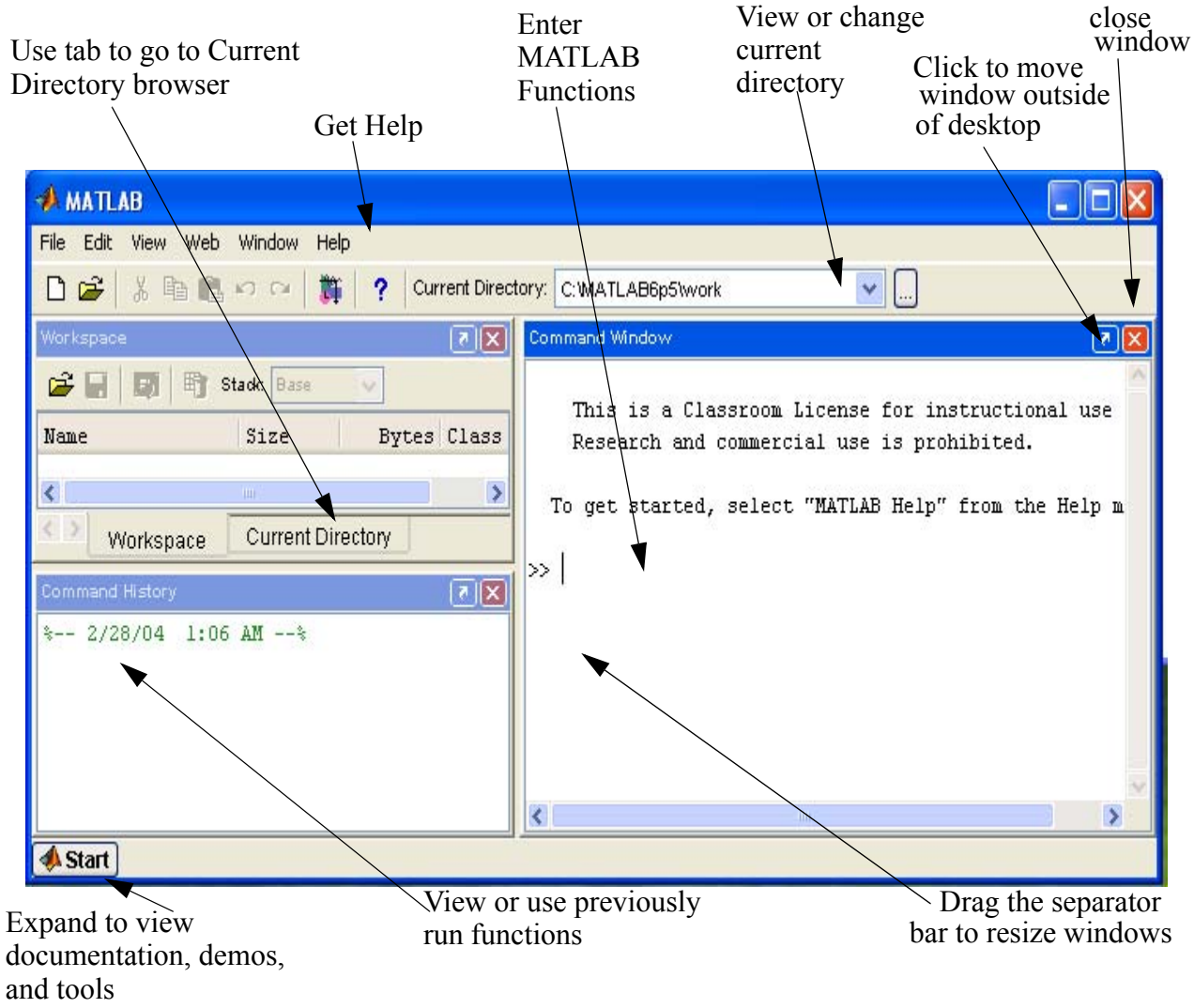


Figure 1.1

1.4.2 Notes on working with the Command Window

- To type a command, the cursor MUST be placed next to the command prompt ">>"
- The end of a command is when the **`Enter`** key is pressed. Once a command is entered, it is executed. However, only the last command is executed. Previous commands are unaltered and unexecuted.
- Several command may be typed on the same line by typing a comma "," between the commands. Again, when the Enter key is pressed, the commands are executed in order from left to right.
- A previous command may be "recalled" to be the current command by using the up arrow. When the command is displayed at the command prompt, it can be modified is needed and then executed. The down-arrow may be used to move down the previously typed commands.

- If a command is too long to fit on one line, it can be split between lines by typing three periods "...", pressing the Enter key, and continuing the command on the next line. A command may extended over several lines up to a total of 4096 characters.

1.4.3 Continuation and Suppress Echo:

MATLAB	C++
Statement: Terminated by an end-of-line <carriage return> unless you put a ... at the end of the line to continue. Can also terminate with a semicolon ; which suppresses the echo of the statement.	Statement: Terminated with a semicolon
x1 = 1 + 1/2 + 1/3 + 1/4 ... + 1/5 + 1/6;	x1 = 1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6;
Continuation: ...	NA
Suppress echo: ; >> x = 3 %evaluation printed x = 3 >> x = 3; %evaluation suppressed >>	NA

1.4.4 Comments:

MATLAB	C++
% a single line comment v = 1584; %Initial velocity (km/h)	// single line comment /* a multi-line comment */

1.4.5 The clc command:

The **`clc`** command clears the `Command Window`. After a period of time working in the `Command Window`, the display may become long. The **`clc`** command puts it back to a blank window.

1.4.6 Running External Programs

You can run external programs from the MATLAB `Command Window`, i.e., an editor of your choice. The exclamation point character "!" is a shell escape and indicates that the rest of the

input line is a command to the operating system. This is useful for running other program without quitting MATLAB.

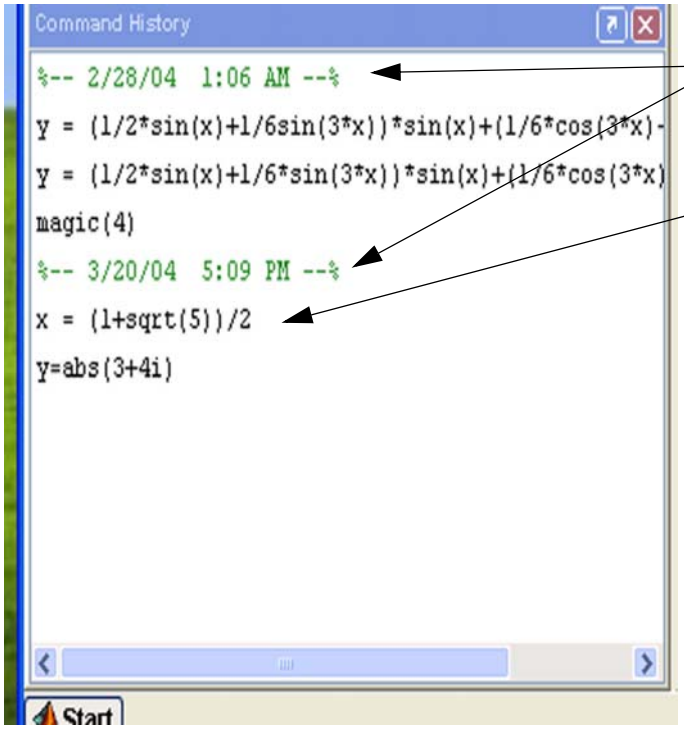
Example:

```
!emacs proj7.m
```

will invoke the `emacs` editor for the file called `proj7.m`. When you exit the external program, the operating system returns control to MATLAB.

1.4.7 Command History Window

The commands you enter in the `Command Window` are logged in the `Command History` window. In `Command History`, you can view previously used functions, and copy and paste to the `Command Windows` to execute your selected lines. You can erase your command history by selecting `Clear Command History` from the `Edit` menu.

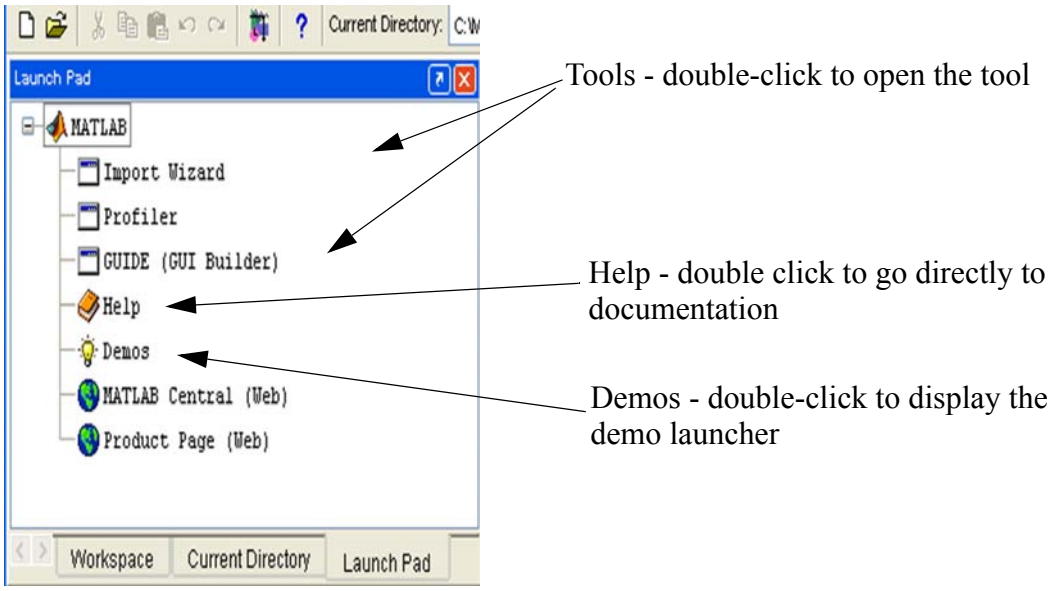


Timestamp marks the start of each session

Select one or more lines and right-click to copy, evaluate, or create an m-file from the selection

1.4.8 Launch Pad Window

The `Launch Pad` gives easy access to the different tools within MATLAB. You may need to go through the `Start Button` (lower left-hand corner) to get to the `Launch Pad`.



1.4.9 Help

- Use the `Help Browser` to search for and view documentation. It is a web browser that displays HTML documents. There are many ways to get to the `Help Browser`
- click the help button **?** in the toolbar
- type `helpbrowser` in the `Command Window`
- pick **Help** from the `Launch Pad` choices

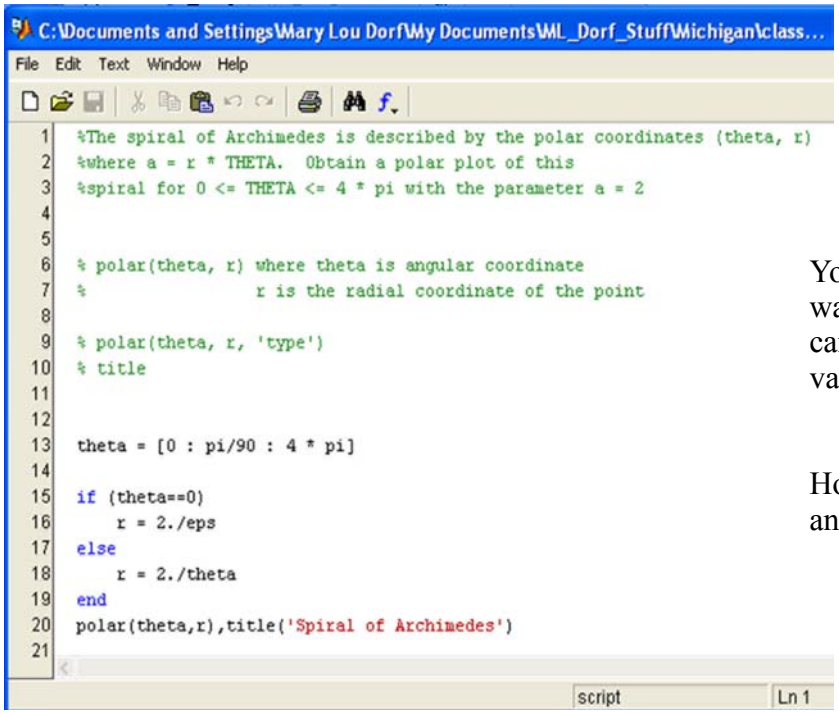
1.4.10 Current Directory

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path. A quick way to view or to change the current directory is by using the `Current Directory` field on the desktop toolbar.

To search for, view, open, or make changes to MATLAB M-files related directories/files, use the `Current Directory` browser.

1.4.11 Editor/Debugger

The `Editor Window` is used for writing and editing programs and functions. This window is opened from the `File` option of the `Main Menu Bar`. The `Editor/Debugger` provides a graphical user interface for basic text editing along with M-file debugging. An example of an `Editor Window` is shown below.



You can set breakpoints where you want execution to pause so you can examine the values of the variables.

Hold the cursor over a variable and its current value will appear

You can display the contents of an M-file in the `Command Window` by using the **`type`** command.

```
>> type input_Example.m

%MatLab - Example of reading input from the keyboard
%check to make sure that the values are within range
%loop back if they are not

% all digits will be from 1 - 9

ones = [1 1 1 1];
nines = [9 9 9 9];


board = [ 0 0 0 0];

indices = [];


while length(indices) < 4
   guess = input('\nPlease enter your guesses--format of [a b c d] :  ');

   %check if guesses are within range
   %if not within range, get another guess
   done = (guess==0);
```

```
   if length(find(done)) == 4
       fprintf('\nThanks for playing.  I win - you gave up ')

       return
   else
      inrange = (ones <= guess & guess <= nines);
      indices = find(inrange);
      if length(indices) < 4
         fprintf(' You need to enter values from 1 - 9 ');
      else
         fprintf(' Your code goes here\n\n')

      end
   end
end
fprintf ('Congratulations  You WIN !!!!!');
```