Chapter 2

Getting Started

Outline

2.1 Programming Language Background

2.2 Basic Data Manipulation

2.3 the Matlab User Interface

2.4 Scripts

2.5 Engineering Example— Spacecraft Launch

Introduction

MATLAB can perform the same functions as your calculator, but it has expanded far beyond its original capabilities.

Fundamental components of MATLAB:

- Accepts one instruction at a time
- A large library of modules that provide high-level capabilities for processing data
- pressing a quality apart from a particular implementation." The specifics are unimportant
- Programs are interpreted rather than compiled
- Excels at matrix calculations and has built-in graphics capabilities
- DOES NOT work well for large computing projects
- Professional GUIs and windowing applications are best written in a compiled language

2.1 Programming Language Background

Abstraction: "expressing a quality apart from a particular implementation." The specifics are unimportant.

Data Abstraction: "To convert from degrees Celsius to Kelvin, you add 273 to *the temperature ." "The temperature"* could mean a single or a table of temperature readings.

Procedural Abstraction: "He *drove home from the office*." Three possible subjects with different levels of abstraction required:

- A businessperson taking the same route home every night
- A competent driver unfamiliar with the route
- A future robotic commuter vehicle

2.1 Programming Language Background

Algorithms: are solutions to a problem or a group of sub-problems, which are a sequence of instructions for solving sub-problems **.**

The level of abstraction needed to describe an algorithm varies greatly with the mechanism available.

Three main practices used in programming are:

- **Functional programming**: every programming operation is actually implemented as a function call.
- **Procedural programming**: is typical of languages like MATLAB, where the basic programs are sequences of operations on data items.
- **Object-oriented programming (OOP**): typical of languages like C, and is characterized by the concept of packaging data items together with the methods and functions that manipulate that data items.

2.2 Basic Data Manipulation

Variable names: In general, variable names may contain any combination of uppercase and lowercase alphabetic letters, numbers, and the special characters '_' and '$' beginning with an alphabetic letter.

Assigning Values to Variables:

e.g. z = x + y

This simply means that you want to sum the values given to the variables *x* and *y* , and store the result in a variable called *z* . If either *x* or *y* is unknown at the time of executing this statement, an error ensues.

2.2 Basic Data Manipulation

Data Typing: When a value is assigned to a variable, Matlab determines the type of data storage depending on the type of data being stored. The following code is legal:

```
a = 42;    % a is type double
a = true;   % a is now type logical
a = 'abc';   % a is now type char
```

MATLAB is an **untyped** language, where each statement is presumed to be correct. If a variable exists, both its type and value are reassigned; otherwise the variable is created.

2.2 Basic Data Manipulation

Typed languages require that programmers declare both the name and type of a variable before a value can be assigned to it.

Typed languages fall into two categories:

- weak typing where the programmer might assign variables of different types.
- strong typing where the programmer (and the compiler) requires to only recognize assignments of similar types.

2.2 MATLAB User Interface

MATLAB uses several displays. The default view includes:

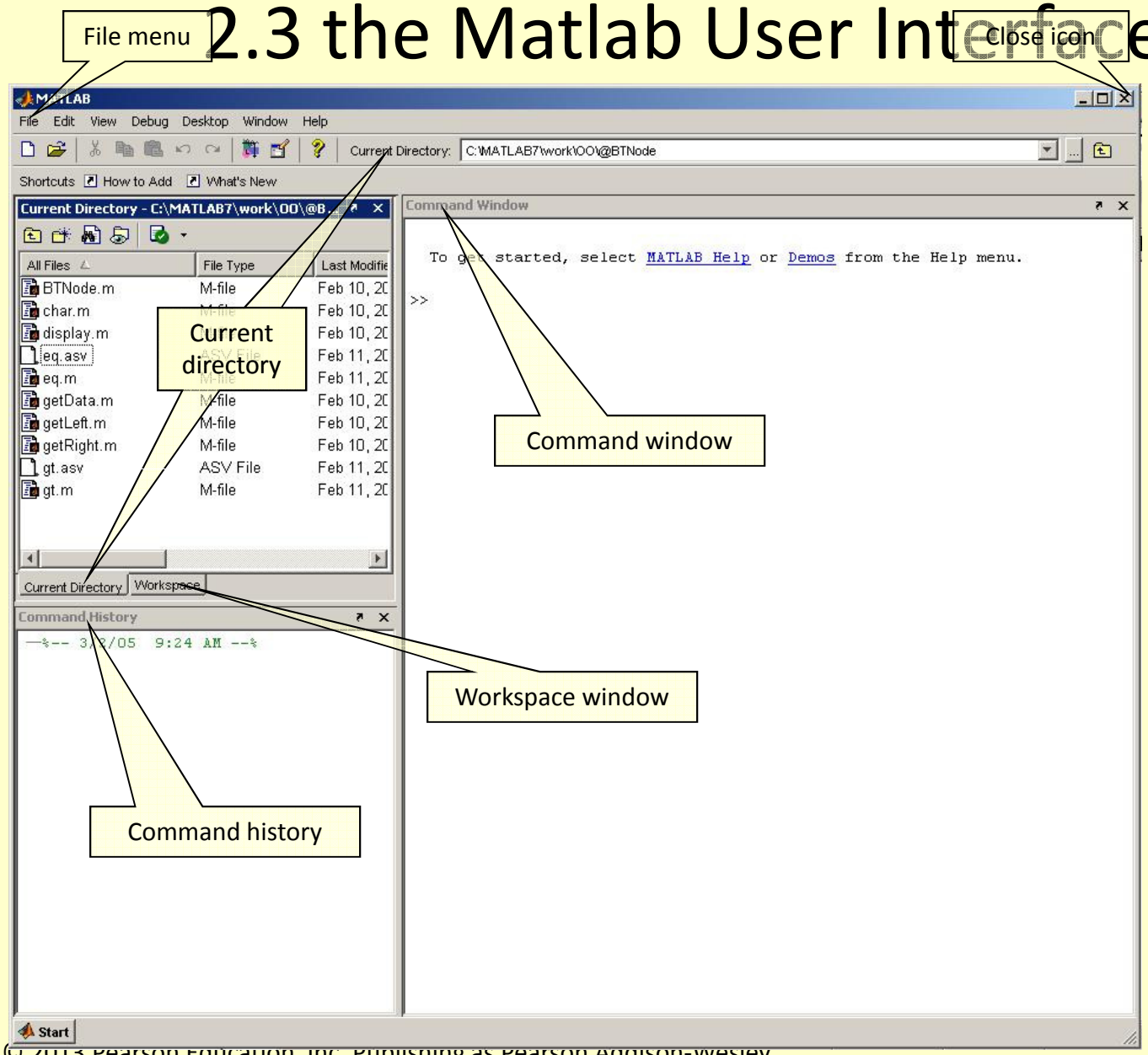
- Command Window
- Files, Workspace, and Command History

Command Window

Useful if you need instant responses to specific MATLAB commands. This window lets you save any values you calculate, but you CANNOT permanently save the commands.

You can use the command window much like doing calculations on a scientific calculator.

2.3 the Matlab User Interface



2.2 MATLAB User Interface

Command History

Records the commands you issued in the Command Window in chronological sequence. When you exit MATLAB or when you issue a clc (Clear Commands) instruction, the commands listed in the Command Window are cleared.

You will find the Command History useful as you perform more and more complicated calculations in the Command Window.

Workspace Window

The workspace keeps track of the variables you have defined as you execute commands in the Command Window.

2.4 Scripts

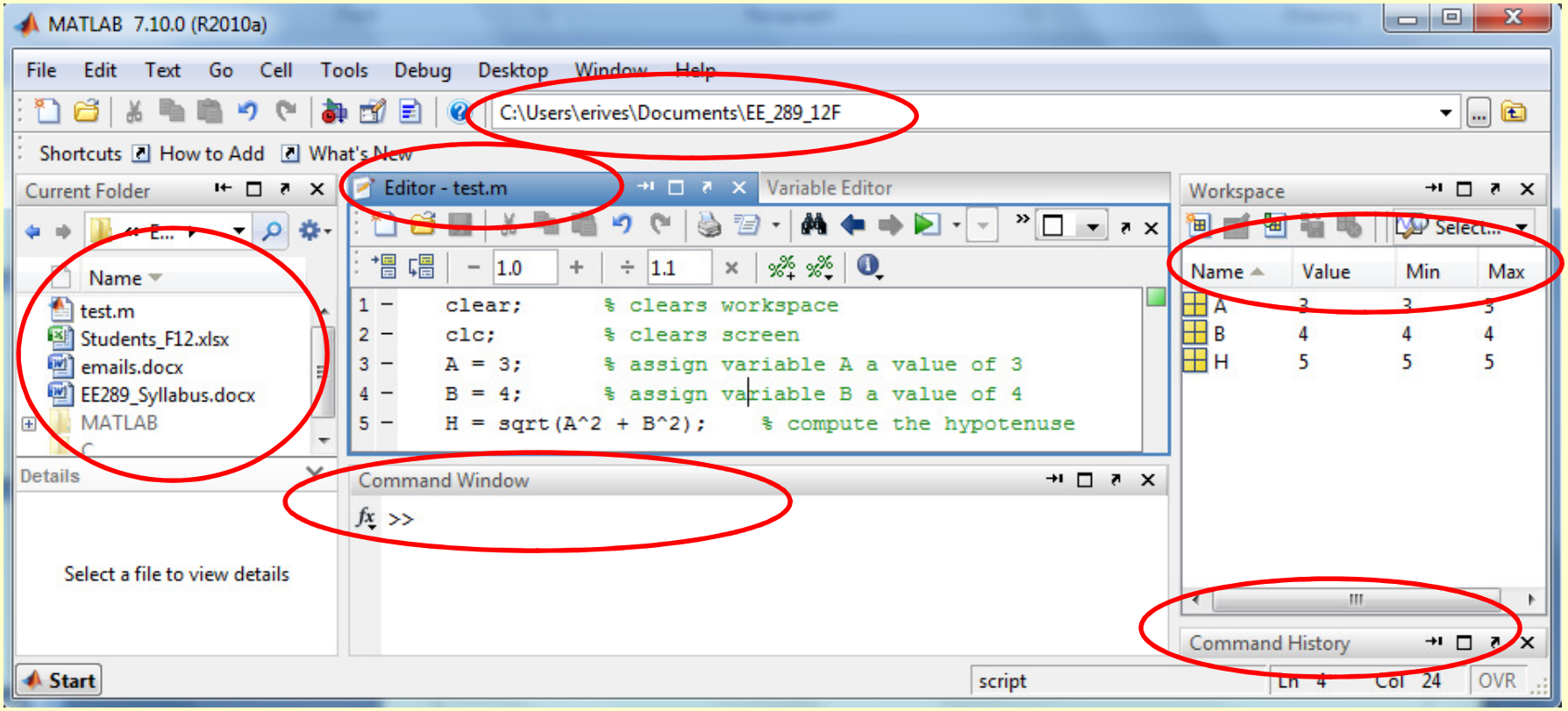
MATLAB uses text files (scripts) as a permanent means of saving sets of instructions.

Text files are streams of characters stored sequentially with "markers" that indicate the end of each line of text.

- You **create comments** by putting a percent sign (%) in the text file. MATLAB will ignore all text from that mark to the end of the current line.
- MATLAB uses the extension .m, and the script files are often referred to as m-files .
- Because MATLAB treats the names of .m files like variable names, the names of your m-files must follow the variable naming convention.

2.4 Scripts

Let us create the simple script to compute the hypotenuse of a right angle



Let’s Write some Code …

2.5 Engineering Example— Spacecraft Launch

In 1996, the Ansari X Prize was offered for the first time for a private venture: a reusable spacecraft. The requirements were for the same vehicle to carry 3 people into the outer space twice in a two-week time period.

The concept was to have a mother ship take off carrying the Space Ship One spacecraft. The spacecraft would be launched at 25,000 feet altitude and would reach outer space (an altitude of 100 Km), then glide back and land.

2.5 Engineering Example— Spacecraft Launch

2.5 Engineering Example— Spacecraft Launch

Problem:

Assuming that the spacecraft uses all its fuel to achieve a vertical velocity u at 25,000 feet, what is the value of u for the spacecraft to reach outer space?

Solution:

There are two parts to this problem: converting units to the metric system, and choosing and solving an equation for motion under constant acceleration (the rocket motor is no longer burning)

2.5 Engineering Example— Spacecraft Launch

1. Convert the launch altitude from feet to meters.

We will write a MATLAB script to find the conversion from feet to meters

$$\text{meters} = (\text{feet}) \times \text{meters/cm} \times \text{cm/inch} \times \text{inch/feet}$$

```
cm_in = 2.54;    % cm/inch
in_ft = 12;      % inches/ft
mt_cm = 1/100;   % meters/cm
mt_ft = mt_cm*cm_in*in_ft;  % meters/ft
h0_ft = 25000;  % Initial height altitude (feet)
h0_mt = h0_ft*mt_ft;  % Initial height altitude (meters)
```

2.5 Engineering Example— Spacecraft Launch

2. Find and solve the equation. Given the following:

- Initial and final altitudes from which you can compute the distance traveled: d
- The motion is under constant acceleration, the force of gravity: g
- To just reach outer space, the final velocity, vf, is 0
- The initial velocity, v0, is needed

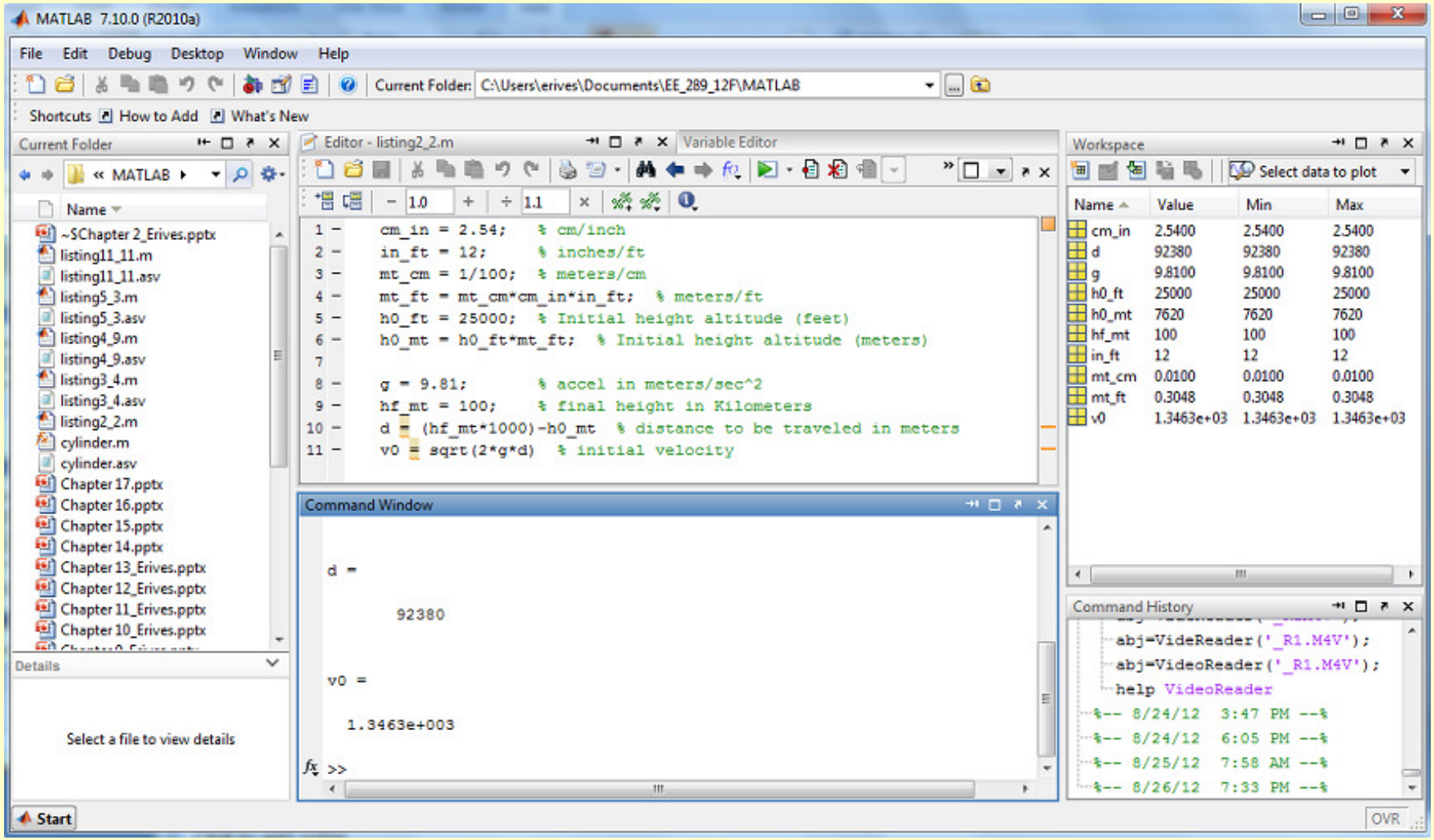
We know that the equation of motion under constant acceleration is

$$vf^2 = v0^2 + 2ad$$

Where vf=0, and since the we assume that positive velocity is upwards, then the acceleration of gravity is negative , so a=-g in our equation:

$$v0 = \sqrt{2gd}$$

2.5 Engineering Example— Spacecraft Launch



Homework on Chapter 2 is posted on the website:

http://www.ee.nmt.edu/~erives/289_F12/EE289.html

Homework is due in a week