

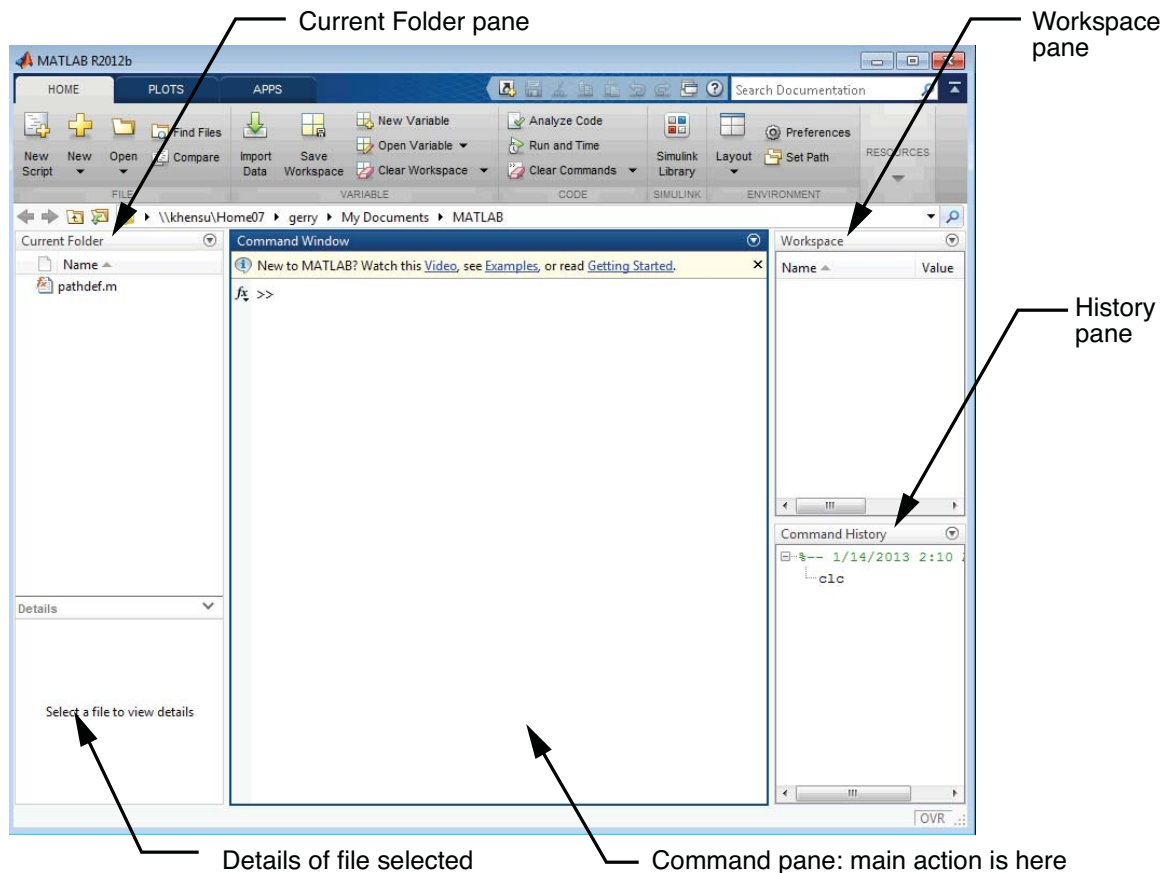
Orientation to MATLAB

Review of screencast

ME 121

1. MATLAB Command Window

Most user interaction with MATLAB involves typing commands into the Command Window.



Commands are entered after the command prompt

```
>>
```

2. Assign values to variables

The expression

```
x = 3
```

creates the variable `x`, assigns a value to it, and echoes that value to the command window. Ending the expression with a semicolon

```
x = 3;
```

has the same effect except that the value of `x` is not echoed to the command window.

3. Use semicolons at the end of lines (or not).

Entering a command in the command window will often yield a response, for example,

```
>> x = pi/4
x =
    0.7854
```

To avoid the echo of the result, end the line with a semicolon

```
>> x = pi/4;
```

When entering commands that produce vectors or matrices, it is usually necessary to suppress output by terminating commands with a semicolon. Otherwise the screen quickly becomes cluttered with displays of large numbers of numerical values.

Semicolons are also useful for entering multiple commands on a single line. For example, when running commands that produce plots, it is often helpful to precede the command that produces the plot with a “close all;” command.

```
>> close all; plot(x,y,'o')
```

The “close all” trick causes any new plots to be created in front of the command window. Of course, the “close all” command also closes any open plot windows, which may not be the desired outcome.

4. Use ‘format compact’ to reduce the use of vertical space

The command

```
>> format compact
```

sets the command window to display results with a minimal amount of blank vertical space, i.e. fewer blank lines. This is an advantage when entering complex sequences of commands since more of those commands (and the output of those commands) remain visible in the command window.

Note that the “format compact” command only remains in effect for the current MATLAB session. You can set the default behavior of MATLAB sessions to use the compact display format by setting your personal preferences as indicated by the partial screen shots in Figure 1.

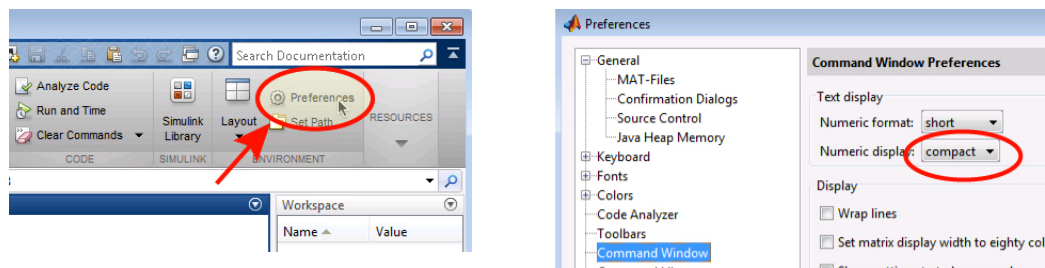


Figure 1 Selecting Preferences from the MATLAB ribbon (left) and selecting the compact text formatting in the Preferences dialog box (right).

4. linspace command

MATLAB has many, many built-in commands. We demonstrated the `linspace` command, which creates a vector of linearly spaced values. For example,

```
x = linspace(0,2*pi);
```

creates 100 equally spaced values between 0 and 2π , where π is the irrational number constant 3.1414159... When creating vectors with the `linspace` command (and other vector-creating commands) it is a good idea to end the statement with a semicolon to suppress printing of the values assigned to the new variable.

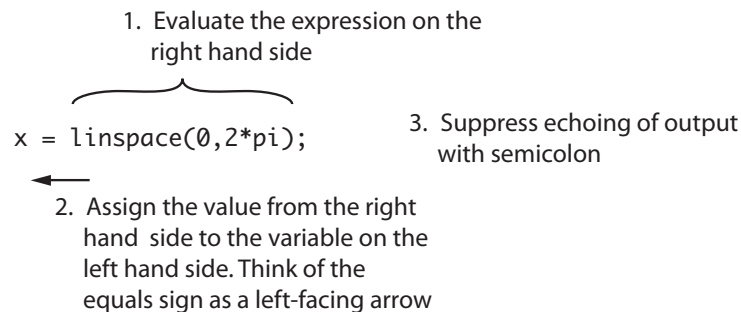
5. Evaluating vector expressions

Vectors matrices are mathematical objects that obey the rules of linear algebra. MATLAB was created to do linear algebra. We don't need to know much about linear algebra just yet. However, it is helpful to know that MATLAB allows any variable to hold a vector or matrix of values.

Consider the following statements

```
x = linspace(0,2*pi);  
y = sin(x);
```

The first statement creates a vector of 100 values and stores them in `x`. The second statement evaluates the sine function for each of the values in `x` and stores them in `y`. The following diagram illustrates the steps that occur during the first assignment.



6. Simple plots of $y=f(x)$

Given the variables `x` and `y` defined as vectors with the same number of elements, the statement

```
plot(x,y)
```

creates a plot of $y = f(x)$ in a new figure window. No semicolon is required to suppress output because there is no assignment (no equals sign) in the expression. The symbol and line type of the plot are controlled by an optional third argument to the plot command. For example

```
plot(x,y,'o')
```

creates a plot of $y = f(x)$ with round circles at each of the (x,y) pairs.

7. MATLAB documentation with help and doc

The `help` command returns MATLAB documentation as plain text in the command window. For example

```
help plot
```

provides information on the `plot` command. The `help` command is useful for a quick review on how to use a command. The `help` command is convenient for experienced MATLAB users, but can frustrate beginners.

The `doc` command returns MATLAB documentation in a separate window called the help browser. The documentation in the help browser includes graphics that cannot be displayed in the command window.