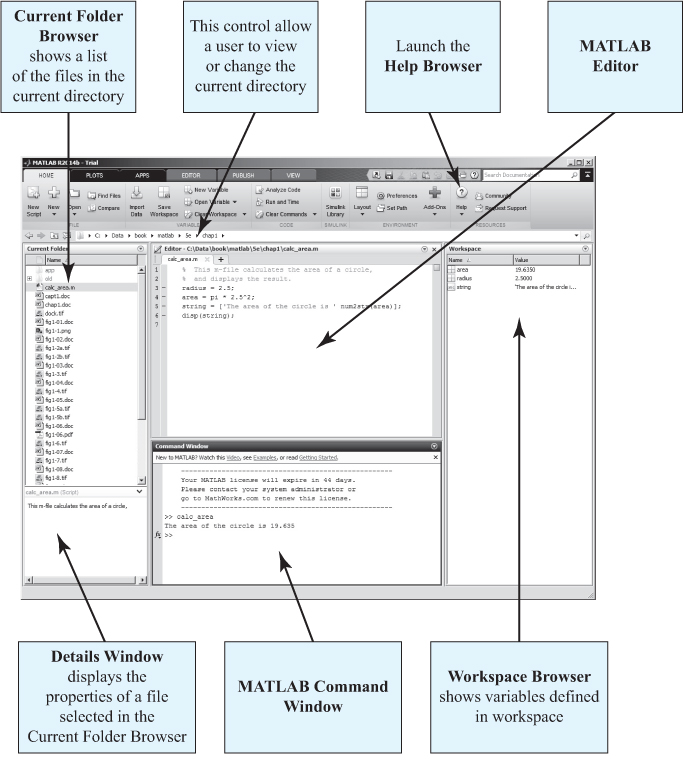
**[BIEN 203: BIOMED PRINCIPLES II](https://grader.mathworks.com/courses/5861-bien-203-biomed-principles-ii)**

**Class Notes 1**

Here is an overview of the MATLAB user interface:



**Command window:**  commands can be entered here and are immediately executed.

**The Command History Window:** Displays previously executed commands.

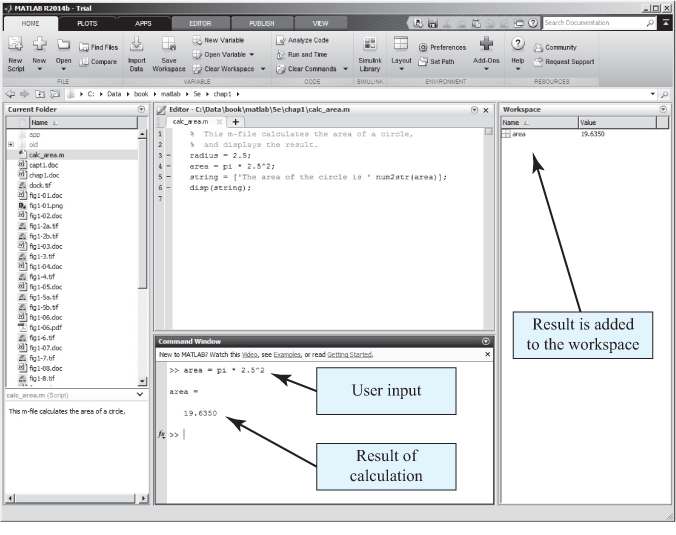
**Workspace window:** Area where the stored variables are displayed along with the variable’s dimension and type. Double clicking on any variable will open the Array editor which allows the user to edit the information stored in the variable.

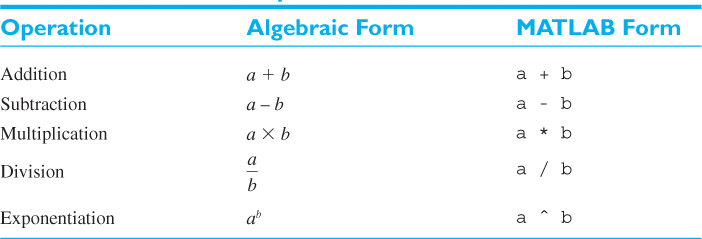
**Path Browser:** allows the user to examine and modify the search path used by MATLAB to find M-files.

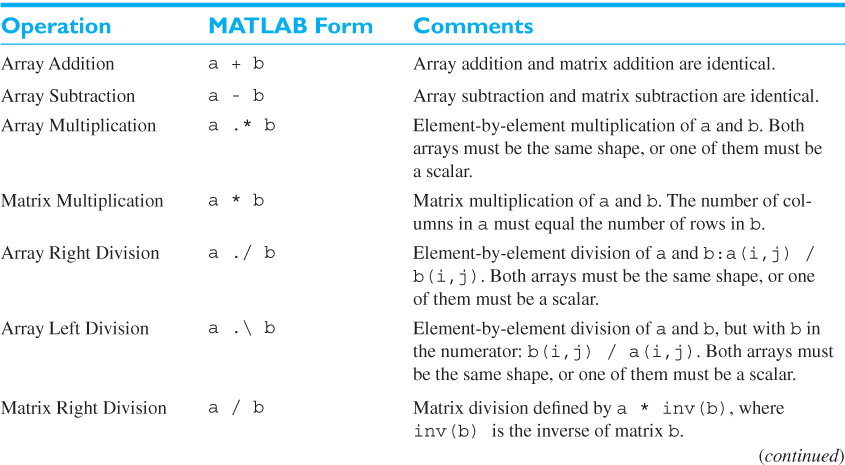
**Editor window:** This is where programs are created or modified. You can make functions or scripts here.

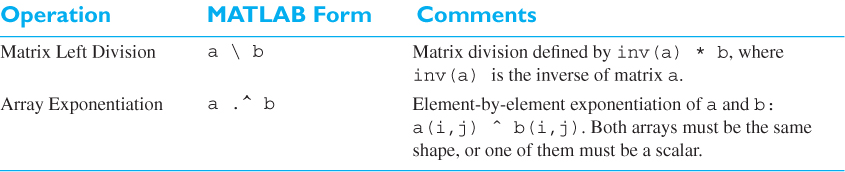
**Figure window:** Plots, graphs, images, or other figures can be displayed in these windows. When creating graphical user interfaces (GUI) controls such as buttons or slider bars can be placed on these.

**The Command window can be treated like a calculator.**









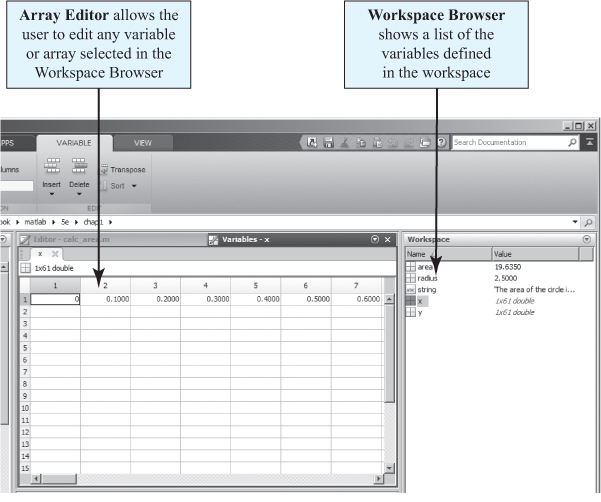
The built-in MATLAB functions below can be used to performed cross and dot product:

cross

dot

Detailed descriptions of these commands can be found at mathworks.com. Simply search these commands within the search bar for more information.

**Double clicking the variables in the workspace will open up the array editor where the variables can be edited in a spreadsheet (this was not mentioned in class, but can be useful).**



A **Variable** isregion of memory containing an array which is known by a user-specified name

* Name must begin with a letter followed by any combination of letters, numbers, or underscores.
* Spaces are not allowed in variable names (use underscores instead).
* Variable names are case sensitive, e.g., “flow” is not the same as “Flow”.

**Numeric** classes in MATLAB® include signed and unsigned integers, and single-precision and double-precision floating-point numbers.

By default, MATLAB stores all numeric values as double-precision floating point.

**Double:** scalars and arrays of 64-bit double-precision floating-point numbers.

* Positive or negative numbers between10-308 and 10+308 with 15 to 16 significant decimal digits of accuracy
* Numerical values assigned to double variables can be
  + Real, e.g. var = 10.5
  + Imaginary. e.g., var = 4i
  + Complex, e.g., var = 10 +10i

All numeric types support basic array operations, such as subscripting, reshaping, and mathematical operations.

Types of Numeric Arrays:

|  |  |
| --- | --- |
| [double](https://www.mathworks.com/help/matlab/ref/double.html) | Double-precision arrays |
| [single](https://www.mathworks.com/help/matlab/ref/single.html) | Single-precision arrays |
| [int8](https://www.mathworks.com/help/matlab/ref/int8.html) | 8-bit signed integer arrays |
| [int16](https://www.mathworks.com/help/matlab/ref/int16.html) | 16-bit signed integer arrays |
| [int32](https://www.mathworks.com/help/matlab/ref/int32.html) | 32-bit signed integer arrays |
| [int64](https://www.mathworks.com/help/matlab/ref/int64.html) | 64-bit signed integer arrays |
| [uint8](https://www.mathworks.com/help/matlab/ref/uint8.html) | 8-bit unsigned integer arrays |
| [uint16](https://www.mathworks.com/help/matlab/ref/uint16.html) | 16-bit unsigned integer arrays |
| [uint32](https://www.mathworks.com/help/matlab/ref/uint32.html) | 32-bit unsigned integer arrays |
| [uint64](https://www.mathworks.com/help/matlab/ref/uint64.html) | 64-bit unsigned integer arrays |

**Character Arrays** and **String Arrays** provide storage for text data in MATLAB®.

* **Character Array:** a sequence of characters, just as a numeric array is a sequence of numbers. A typical use is to store short pieces of text as character vectors, such as c = 'Hello World'.
* **String Array:** a container for pieces of text. String arrays provide a set of functions for working with text as data. Starting in R2017a, you can create strings using double quotes, such as str = "Greetings friend".

**Structural Array:** A data type that groups related data using data containers called fields. Each field can contain any type of data. Access data in a field using dot notation of the form structName.fieldName.

**Cell Array:** a data type with indexed data containers called cells, where each cell can contain any type of data. Cell arrays commonly contain either lists of text strings, combinations of text and numbers, or numeric arrays of different sizes. Refer to sets of cells by enclosing indices in smooth parentheses, (). Access the contents of cells by indexing with curly braces, {}.

**Generating a vector array**

test = [1 2 3] provides an array of numbers listed in a row

test = [1;2;3] provides an array of numbers listed in a column

**Transpose using ‘**

test = [1 2 3]’ provides an array of numbers listed in a column

test = [1; 2; 3]’ provides an array of numbers listed in a row

**Generating a Matrix**

matrix = [1 2 3; 1 2 3; 1 2 3] generates a 3 x 3 matrix as shown below:

1 2 3

1 2 3

1 2 3

**Scripts**: A program containing a series of command lines that can be executed consecutively.

Scripts are the simplest type of program file. They are useful for automating a series of MATLAB® commands, such as computations that you have to perform repeatedly from the command line or series of commands you have to reference.

**Functions:** A program similar to a script, but has user defined inputs.

Setting path for m-files was discussed with mentions of the **addpath** and **rmpath** functions.

The command below identifies where the m-file is located.

which m-filename -all

e.g.

which rmpath -all

C:\Program Files\MATLAB\R2018a\toolbox\matlab\general\rmpath.m

**Plotting**

Typing “figure” in the command window will generator a figure window. All figure windows are numbered.

Typing “figure (1337)” will generate a figure window with the number 1337.

We discussed commands such as:

plot

subplot

xlabel

ylabel

title

gcf (get current figure)

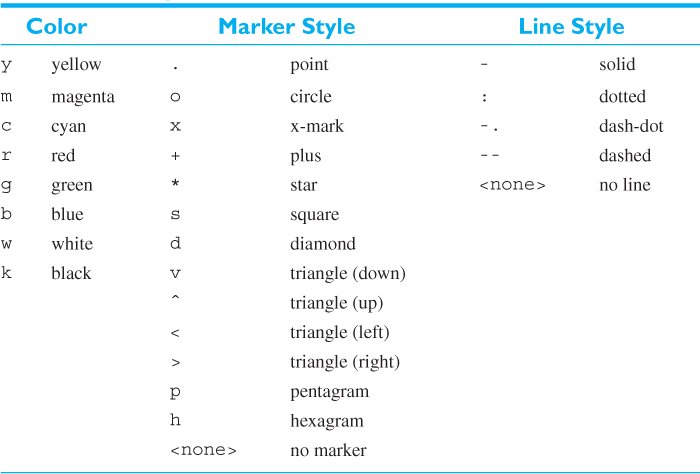
gca (get current axis)

Detailed descriptions of these commands can be found at mathworks.com. Simply search these commands within the search bar for more information.

**Figure handles** (The structural array of the figure)

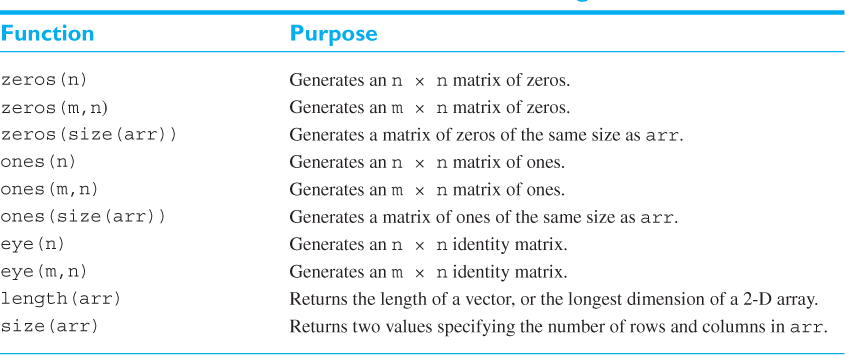
We covered how to access the Figure handles to change properties such as the figure’s background color and the figure’s window position.

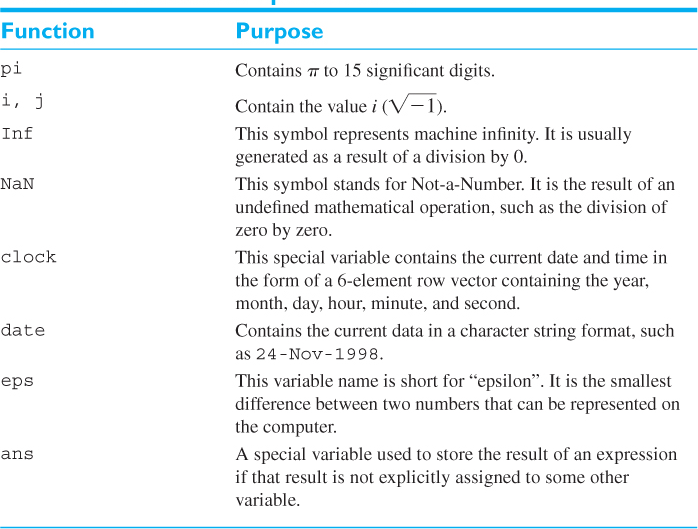
See the table below for options for different plot parameters. We will cover more of these options in the proceeding classes.

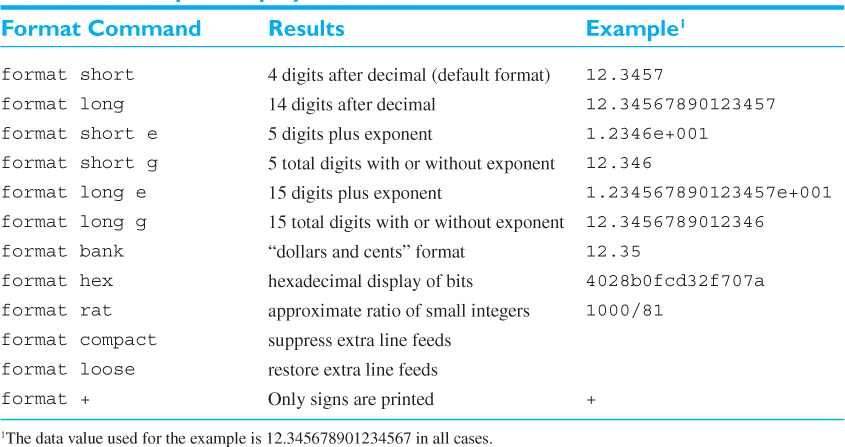
****

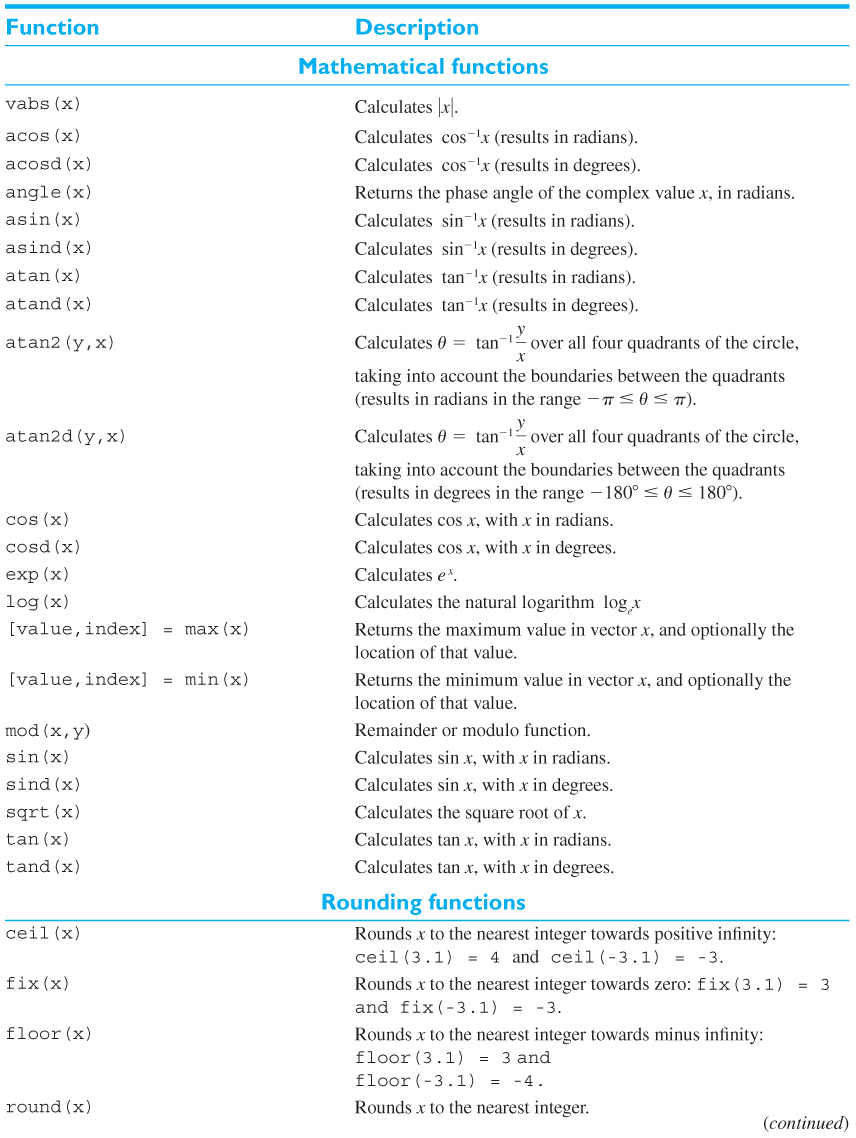
**Not covered in class**

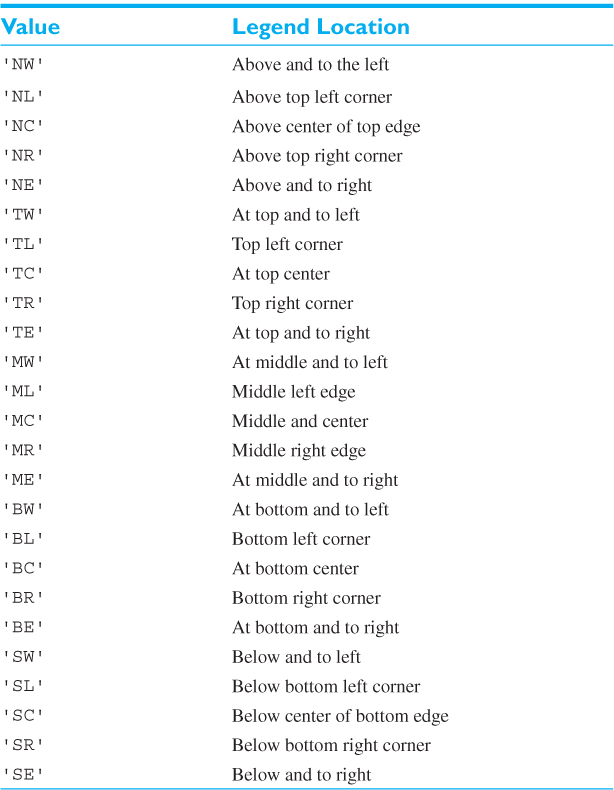
**Useful list of commands:**









****

What we talked about:

* To delete use delete function
* To clear workspace use clear function
* To clear command window use clc function
* To stop a running program press Ctrl+C
* To run a program file just type the name of the file
  + Make sure your matlab is in the right folder