

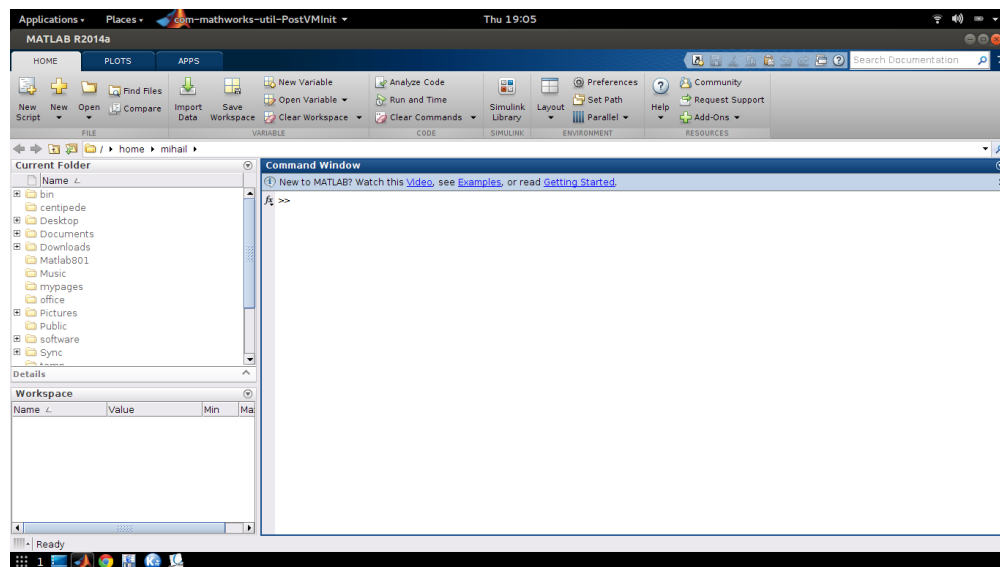
# Computing for Scientists - Lab 1

CS 1340 — Dr. Mihail  
Department of Computer Science  
Valdosta State University

August 23, 2018

## 1 Introduction

In this first lab assignment, you will get acquainted with the MATLAB environment. When you first start MATLAB, you will see a window that looks (more or less) like this:



You will first interact with the “Command Window”. This is an environment similar to a notepad, awaiting commands with the prompt easily recognizable by the two > signs: >>. The values (or variables) that result from your commands are saved in the “Workspace” window. You can save the variables you create (although it won’t be necessary for this lab) using the command *save* followed by the file name with the extension *.m*. For example:

```
>> save mywork.mat
```

will save the variables you created in the file *mywork.mat*. To restore the variables at a later time, you may load them using the *load* command followed by the file name:

```
>> load mywork.mat
```

In the following exercises, you will enter commands in the command window and report the results by selecting and copying the text from the command window, and pasting it into a blank Notepad text document. You will then save the text file in the following format: **LastName\_FirstName\_LabX\_ProblemX.txt**, where X is the problem number you are solving. At the beginning of every file you submit, please include the course, name, lab number and problem number. For example:

```
% CS 1340
% John Doe
% Lab 1, Problem 1
```

The percentage sign is used in MATLAB scripts to indicate comments, i.e. text that will be ignored by MATLAB, but used to document your code. Comments are for people, put in code not only to help you organize and document your work, but also to help others working on the same project better understand your logic. Comments explain details about a block of code, or even a line of code.

The = sign is called the **assignment operator**. The syntax for simple assignment statements in MATLAB is the following:

$$\textit{variable\_name} = \textit{expression} \tag{1}$$

The above expression stores the result of *expression* into the variable *variable\_name* if it exists; if it doesn't yet exist, MATLAB first creates the variable, then stores (assigns) the result of the *expression* to it. The following assignment statement

```
>> x = 10
```

```
x =
```

```
    10
```

```
>>
```

creates a variable named *x* and stores the value 10 in it (10 considered a simple expression). Notice MATLAB reports the value of the variable right after the command is entered. This “verbose” behavior can be suppressed by including a semicolon “;” at the end of the assignment statement, as follows:

```
>> x = 10;
>>
```

## Problem 1, in-class individual work

Type the following *statements*, in order, in MATLAB's command window:

1. `x = 2`
2. `x = x + 2`
3. `y = x * 2 - 1`
4. `z = y / 3`
5. `a = (x + y + z)^2`
6. `b = 9^(1/2)`
7. `factorial(10)`
8. `x = ans / 10`
9. `ls = linspace(0, 1, 100)`

Notice how #7 is not an assignment statement (there is no = sign). The result is stored in a variable called `ans` (answer). This variable contains the result of the previously entered expression. It can be used as a variable, as you have done in #10.

Select and copy the contents of the Command Window and paste into a new text document (named as directed above). At the end of the file, write the final values of all the variables you created. Also, answer the following questions:

- Using the command `numel(ls)`, how many elements does `ls` have?
- What does the command `size(ls)` return? What can you say about the size of `ls` from the output of `size`?

Submit this file using the electronic submission link on the course web page:

<http://mypages.valdosta.edu/rpmihail/teaching/esubmit.html>

After submitting the text file, enter the command: `clc` in the command window. This will clear the command window. You can then start working on Problem 2.

## Problem 2, in-class individual work

In Problem 1, all the variables you created were scalars. In this exercise, you will create vectors and matrices. Row vectors are matrices with dimensionality 1 by  $c$ , where  $c$  is the number of columns. Column vectors are matrices with dimensionality  $r$  by 1, where  $r$  is the number of rows. Make sure the command window is cleared, then enter the following statements, in order:

1. `a = [1 2 3 4 5]`

2. `size(a)`
3. `b = [1; 2; 3; 4; 5]`
4. `size(b)`
5. `c = [1 2; 3 4; 5 6]`
6. `size(c)`
7. `d = [1 2 3; 4 5 6]`
8. `e = [1 2 3; 4 5 6 7]`
9. `whos`

The `size` function takes a variable as input and returns the size of matrices (and scalars and vectors as special cases of matrices) as a column vector, where the first element of the returned vector is the number of rows, and the second is the number of columns. Based on the results seen in the command window, answer the following questions:

1. What is `a`? What is the size of `a`?
2. What is `b`? What is the size of `b`?
3. What is `c`? What is the size of `c`?
4. What is `d`? What is the size of `d`?
5. Given what you observed, what does the space do in statements that create vectors and/or matrices?
6. What does the semicolon do inside `[ ]`?
7. Why was there an error after entering statement `#8`?
8. What did the command `whos` do?

Copy/paste the contents of the command window for Problem 2 into a text file using Notepad, just like you did for Problem 1. Name the file similarly, the only thing that's different in the file name is the problem number. Submit the text file using the course electronic submission page: <http://mypages.valdosta.edu/rpmihail/teaching/submit.html>

## Due dates

1. Lab Problems 1 and 2 are due at the end of the day 8/26/2018, before midnight.