

Basic Matlab Arithmetic and Plotting

February 1, 2017

1 Introduction

This lab is intended as a basic introduction to Matlab. Matlab is a powerful computational tool with many built in features. It is one of the industry and academic standards in the applied sciences and engineering. Matlab is short for : Matrix Laboratory. In all of the labs, the text that is placed after the numbers will be the Matlab commands. For example: “1. 5²” means to place “5²” in the Matlab prompt which is viewed as “>> 5²” in Matlab. The “>>” denotes the Matlab prompt. Text to the right of the percent symbol % are comments. In fact, the % symbol is the actual comment symbol in Matlab.

2 Getting Started

Go to start, then programs, and then Matlab. The program should open and you should see the Matlab prompt >> in the window.

3 Basic Scalar Arithmetic

Matlab can be used for doing calculations on either scalars, vectors, or arrays. Here we will do some simple scalar arithmetic.

1. $3*2^2$
2. $(3*2)^4$ % parenthesis have biggest priority
3. $3-2^4$
4. 3^4-3
5. $8/2^4$
6. $2^4\backslash 8$ % same as preceding. 2 different divisions, \ and /
7. $8^4/2$

Order of arithmetic operations are:

1. parentheses
2. exponentiation
3. multiplication and division
4. addition and subtraction

Calculations proceed from left to right on the line and equal precedence in this way.

4 Creating Vectors

1. `x = [3 4 7 11]` % create row vector. note the spaces
2. `x = 3:8` % colon generates list. default stride 1
3. `x = 8:-1:0` % <start> : <stride> : <stop>
4. `xx = [8 7 6 5 4 3 2 1 0];` % semicolon suppresses output
5. `x = linspace(0,1,11)` % generate 11 linearly spaced points between 0 and 1
6. `x = 0:0.1:1` % same as what linspace does but you need to specify the spacing
7. `y = linspace(0,1);` % note semicolon, try it without
8. `length(x)` % self explanatory
9. `length(y)`
10. `size(x)` % self explanatory
11. `size(y)`
12. `y(3)` % access single element of vector
13. `y(1:14)` % access first 14 elements of y
14. `y([3 6 9 12])` % access values specified in a single vector
15. `x'` % transpose. changes a row vector x into a column vector and vice versa
16. `z = [1+2*i 4-3*i]`
17. `z'`
18. `z.'` % note differences in transpose for a complex vector!
19. `3*[1 2 5]` % factor multiplies each element

Important note, Matlab indexes arrays (vectors and matrices etc) beginning with 1.

5 Help with Vectors and More

1. `help help` % Matlabs extensive help
2. `help length`
3. `help size`
4. `help linspace`
5. `help logspace`
6. `help clc`
7. `help clear`
8. `help who`
9. `help whos`

6 Functions and Plotting

1. `help elfun` % Matlabs extensive built in elementary functions
2. use the following commands to make a plot of the cosine function. The evaluation of $v = \cos(u)$ in Matlab creates a vector whose elements are $v(k) = \cos(u(k))$ where $k = 1, 2, \dots, n$.
 - `n = 111;`
 - `u = linspace(0,2*pi,n);`
 - `v =cos(u);` % all function evaluations done in one fell swoop!
 - `plot(u,v)`
 - `xlabel('u'); ylabel('v'); title('v = cos(u)')` % Matlabs extensive help

7 Image Manipulation

1. Type the command: `>> f = imread('pout.tif');`
now type: `>> whos`
what do you notice? Can you find the max and min values of the image by using the commands `max(max(f))` and `min(min(f))`?
Set `f` to be a matrix of doubles, i.e. high precision alterable entries, by typing `>> f = double(f)`
Use the following code to view the image: `>> figure(1); image(f); colormap(gray(256))`
You can find the statistics of the image in the following manner:
`>> figure(2); histogram(f,256);`
The histogram showing the distribution of the pixel intensities should show up. Following in the manner from our first class, re-map the max and min values of `f` to 0 and 255 respectively.

Plot both images and show there is some improvement with the contrast. View the histogram for this adjusted image as well. You can view both using the subplot command. We notice that the processed image, albeit better in contrast, still has the white shirt looking a bit dark. Can you think of a method to adjust it? i.e. bring those values up? Implement it!