WORCESTER POLYTECHNIC INSTITUTE
BME 2211

Class Notes from March 20, 2015

Katie Brochu and Jack Blanchard

March 26, 2015

1 VECTORS AND MATRICES

Creating Rows and Vectors

x = 1:4 creates the following vector

$$[1,2,3,4] \tag{1.1}$$

You can also specify the amount you want the elements of the vector to increase by with the following format: start: step size: end
Example:
x = 2:2:8 creates the following vector

$$[2,4,6,8] \tag{1.2}$$

The first 2 represents the starting value, the second 2 is the step size, and the 8 is the end value. If you want the vector to decrease you can just make the step size negative.

1.1 LINSPACE AND CONCATENATION

The function linspace (x,y,n) creates a vector with n values in the inclusive range from x to y.
Example:

y=linspace(5, 50, 10) creates the vector

$$[5, 10, 15, 20, 25, 30, 35, 40, 45, 50] \tag{1.3}$$

Concatenation is a way to combine two vectors. If you say z = [x,y] for instance,then z will be the combination of x and y. If x='cat' and y='dog', then the resulting z would be 'catdog'.

1.2 Referring to Elements

MATLAB is a 1-based indexing software.
So if you wanted to refer to the first element of a vector such as x = [5,2,7] then you would use x(1) to return 5.
A way to access the last element when you don't know how long a vector is, is by typing x(end) into MATLAB.

You can also create an index vector. For example, if you have y=[1,3] and the same x as previously stated, typing x(y) would return the first and third elements of x in the form of a vector.

Another method of referring to an element is when you want to know which elements are of a certain value. For example, if you have the vector x= 1:10, and you want to know which elements are greater than 3, you can type in x>3 and a vector of logical values will be returned to you. You can then use this logical vector as an index by using x(x>3), which would return the vector [4, 5, 6, 7. 8, 9, 10].

1.3 Matrix Dimension

length(vector) = returns the number of elements in a vector
length(matrix) = returns the larger dimension (either row or column) for the matrix
size (vector or matrix) = returns the number of rows or columns for both vectors and matrices
numel(vector or matrix) = returns the number of elements in both vectors and matrices

1.4 Logical Built-in Functions

For these commands the outputs are all logical.
any = returns true if anything in the input argument is true
all = returns true only if everything in the input argument is true
find = finds location and returns indices

Example: with any(z([1:2:5]) you are asking if any of the first, third, and fifth elements of z are true. You are examining the first, third, and fifth elements because as mentioned above: 1 is the start value, 2 is the step value, and 5 is the end value.

1.5 Column Vectors and Matrices

There are two ways to create a column vector: You can transpose a row vector into a column vector by putting an ' symbol after the variable name.

You can also use semi-colons inside of the vector bracket to indicate a new row.
For example: x=[1;3;4;6] would produce a column with the values 1, 3, 4, and 6.

You can also use this method to create a matrix. An example of this is x = [1,2;3,4]. In this case the first row of the matrix would be 1 and 2, and the second row would be 3 and 4.

Similarly to referring to elements in a vector, the last element in a matrix can be found using the command x(end, end).
To refer to a specific element you just type the variable name(row, column).

If you want to convert a matrix to a vector you would use the : symbol.
For example: if you have a matrix x and wish to make it a vector you would say x(:).