

# **MATLAB for beginners**

KiJung Yoon,<sup>1</sup>

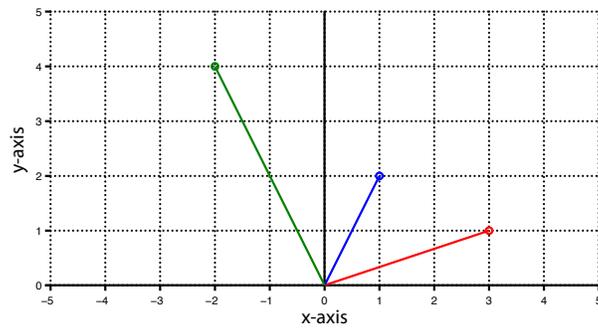
<sup>1</sup>*Center for Learning and Memory, University of Texas at Austin, Austin, TX 78712, USA*

# MATLAB Tutorial I

## • What is a matrix?

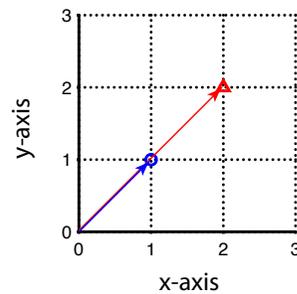
1) A way of representation for data (# of rows: dimensionality, # of columns: the number of data).

$$D = \left[ \begin{array}{ccc} \textcircled{1} & \textcircled{3} & \textcircled{-2} \\ \textcircled{2} & \textcircled{1} & \textcircled{4} \end{array} \right] \left. \vphantom{\begin{array}{ccc} \textcircled{1} & \textcircled{3} & \textcircled{-2} \\ \textcircled{2} & \textcircled{1} & \textcircled{4} \end{array}} \right\} \begin{array}{l} \text{\# of data} \\ \text{dim} \end{array}$$



2) Linear transformation (e.g. scaling, rotation, projection, reflection etc.).

$$T = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$



3) A system of linear equations.

$$\begin{cases} x - 3y = 5 \\ 2x + 5y = 6 \end{cases} \Rightarrow \begin{bmatrix} 1 & -3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

- **Why MATLAB?**

MATLAB (**matrix laboratory**) is an interactive, matrix-based numerical computing system, which works with essentially only one kind of object, which is a **matrix**.

Easy to learn.  
Many useful features ('toolboxes').  
Standard: used all over by engineers and scientists, etc.

Not good for large software projects.  
Proprietary.

- **How to open it?**

Click the matlab icon generated after installation, or type "matlab" or "matlab -nodisplay" in a terminal window (especially for opening many matlab sessions).

- **How to create a matrix?**

» A = [1 2 3; 4 5 6; 7 8 9]

» A = [

1 2 3

4 5 6

7 8 9]

- **Scalars or vectors are a special case of matrix:**

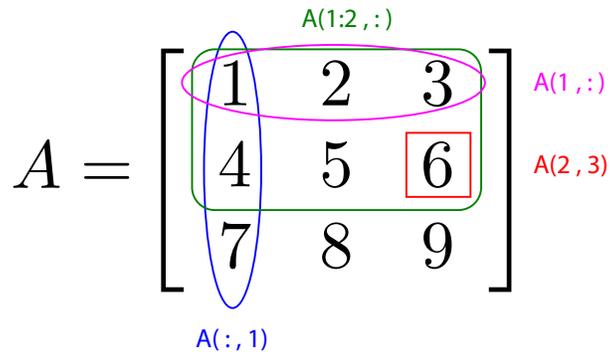
» B = [1]

» B = 1

» B = [1 2 3]

» B = [1;2;3]

• How to reference an element, a row, or a column vector:



»  $A(2,3)$

»  $A(1,:)$

»  $A(1:2,:)$

»  $A(:,3)$

»  $A(:)$

• Matrix arithmetic:

+ addition:  $(A + B)_{ij} = A_{ij} + B_{ij}$

- subtraction:  $(A - B)_{ij} = A_{ij} - B_{ij}$

\* multiplication:  $(A * B)_{ij} = \sum_{k=1} A_{ik} B_{kj}$

.\* element-by-element multiplication:  $(A .* B)_{ij} = A_{ij} B_{ij}$

/ division:  $A/B = A * B^{-1}, A \setminus B = A^{-1} * B$

./ element-by-element division:  $(A ./ B)_{ij} = A_{ij} / B_{ij}$

^ power:  $A^2 = A * A$

.^ element-by-element power:  $(A.^2)_{ij} = A_{ij} A_{ij}$

' transpose:  $(A')_{ij} = A_{ji}$

• **Exercise for matrix arithmetic:**

» % try addition and subtraction between the 1st and 2nd columns of A.

»  $A(:,1) + A(:,2)$

»  $A(:,1) - A(:,2)$

» % try element-wise square and transpose of the 1st row of A.

»  $A(1,:).^2$

»  $A(1,:)'$

» % try inner product of the 1st and 2nd columns of A.

»  $A(:,1)' * A(:,2)$

» % try inner product of the 1st and 2nd rows of A.

»  $A(1,:) * A(2,:)'$

» % try element-wise product of the 1st and 2nd columns of A.

»  $A(:,1) .* A(:,2)$

• **Usage of colon and semicolon**

1) Matrix row separator: When used within square brackets to create a new matrix or concatenate existing arrays, the semicolon creates a new row in the array.

2) If the last character of a statement is a semicolon, the printing is suppressed, but the assignment is carried out. This is essential in suppressing unwanted printing and intermediate results.

»  $x = \sin(A);$

»  $x = \sin(A)$

3) The colon notation can be used to access sub-matrices or sub-vectors of a matrix.

4) Range specifier: A colon allows ones to generate a series of numbers without the use of loops, which will be studied in the next class.

»  $x = 1:10$

»  $x = 10:-1:1$

»  $x = 0:0.5:3$

### • What is M-files?

MATLAB can execute a sequence of statements stored in a file. Such files are called M-files because they must have the file type of ".m" as the last part of their filename. Much of your work with MATLAB will be in creating and refining M-files. M-files are usually created using your local editor. There are two types of M-files: *script files* and *function files*.

1) *Script files* consist of a sequence of normal MATLAB statements. If the file has the filename, say, **test.m**, then the MATLAB command **test** will cause the statements in the file to be executed.

» % write **test.m** with the following pseudo-code:

» % type cmd+n to create a new file.

» % make a column vector originating from (0,0) to (2,1), and assign it to **x**.

»  $x = [2; 1];$

» % make a scaling matrix (scale factor : 2), and assign it to **A**.

»  $A = [2 \ 0; 0 \ 2];$

» % scale up **x** twice, assign it to **y**, and print it out.

»  $y = A * x$

2) *Function files* provide extensibility to MATLAB. You can create new functions specific to your problem which will then have the same status as other MATLAB functions. (This was not studied in the class.)

» % write **scaling.m** that scales up a vector by an arbitrary amount  $\beta$ .

» function [vScaled] = scaling(vInput, beta)

» scaleMat = [beta 0; 0 beta];

» vScaled = scaleMat \* vInput;

3) Modify **test.m** by incorporating the **scaling** function. (This was not studied in the class.)

- **How to make a 2D plot?**

- » `% "doc functionname"` displays documentation for the functionality specified by name.

- » `doc plot`

- » `% plot the input vector x and the scaled vector y`

- » `% set axis limit, marker(size, color) and show axis label, title, legend, etc.`

- » `% refer to the attached file test.m`