Introduction to MATLAB-2

For B.Sc. Sem-4 ELTA Sec-2

Basic arithmetic operations in MATLAB

MATLAB allows you to do operations on variables as single entities like in matrix-matrix multiplication and also operations on the individual elements of the matrix.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 4 & 2 \\ 3 & 5 \end{bmatrix} \text{ is equal to } \mathbf{AB} = \begin{bmatrix} 10 & 12 \\ 3 & 5 \end{bmatrix}.$$

Remember that the multiplication AB is possible because the number of columns in A is equal to the number of rows in B. Matrix multiplication is generally not commutative (commutative means AB = BA), so it is no surprise to find that

$$\mathbf{BA} = \begin{bmatrix} 4 & 10 \\ 3 & 11 \end{bmatrix} \text{ is a different matrix.}$$

Ready-made matrices and vectors

- MATLAB provides functions to create several basic matrices automatically without having to type or read in each of the elements. The most important functions are
- **zeros** **zeros(m,n)** creates an mxn matrix whose elements are equal to zero
- **Ones** **ones(m,n)** creates an mxn matrix whose elements are equal to one.
- **eye** **eye(m,n)** creates an mxn identity matrix
- **rand** **rand(m,n)** creates an mxn matrix whose elements are all random number between 0 and 1
- Create a 5x5 matrix with random numbers. Now add to it a diagonal matrix with the diagonal elements equal to 2. You can create such a diagonal matrix using the command **2*eye(5,5)** .

The colon operator as a way of generating row vectors

- The colon (:) is used a lot in MATLAB. One of its functions is to generate sequences of equally spaced numbers in a row vector.
- Type **h=10:2:20**. You should see that you have created a row vector h with elements starting at 10 and increasing in steps of 2 up to 20. We could also have created this vector with **h=[10 12 14 16 18 20]**. Note that we did NOT need brackets in **h=10:2:20**
- If the step size is negative the sequence decreases. So **i=9:-1:5** sets i to be [9 8 7 6 5].
- Type **r=50:55**. You should have the vector [50 51 52 53 54 55]. In other words, if the step size is omitted it is assumed to be +1.
- MATLAB suppresses the output of a command if you finish the command with a semi-colon **;**.

Building larger matrices from smaller ones

Create the following matrices

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} -8 & -7 & -6 \\ -5 & -4 & -3 \end{bmatrix}$$

Now type **C=[A B]** or **C=[A,B]** and you should see that MATLAB put A and B side-by-side and called the resulting matrix C.

Type **D=[A B;B A]** and you should see that MATLAB placed the matrix [B A] below the matrix [A B] and called it D.
Now type **D=[A;B]**

This gives an error message, which tells you that this operation cannot be done. The number of columns in A and B are not equal

Changing values of individual elements

- Individual elements in a variable can be identified using index numbers. For example the element 3 in the matrix A above can be referred to as A(2,1) because it is the first element of the second row of A. Note that the row and column indices are separated by a comma.
- Change this element to have the value of 1 by typing **A(2,1)=1**
- Note how MATLAB prints the whole matrix (unless you include a ; at the end), and that A(2,1) has changed. Similarly sections of matrices (sub-matrices) can be identified using the colon operator.
- Type **A2 =A(1:2,1)** to extract the first column of A and store it in A2.
- We could also have achieved this by typing **A2 = A(:,1)** . The colon operator identifies *all* of the first column (taking the first element in ALL of the rows). Similarly **A(2,:)** identifies the whole of the second row of A. Note that MATLAB will object if you use negative (or zero!) indices for matrices.

Transposing matrices and vectors

The operator ' (single quote) finds the transpose of a variable (note that the transpose of a scalar is just the scalar itself). Transposing a matrix or vector swaps rows and columns while retaining their order. For example if A is the matrix

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \text{ then } \mathbf{B} = \mathbf{A'} \text{ gives the matrix } \mathrm{B} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

A column vector is the transpose of a row vector.

Saving and retrieving your work

- All variables that you created in this MATLAB session are stored in MATLAB's workspace. Upon exiting MATLAB this workspace will be destroyed. So you must save your workspace if you want to use it in later MATLAB sessions.
- Select *Save Workspace As* under *File* on the menu. A save window will come up. Check that the correct folder is given. Type in the name you wish to use for the workspace file.
- **VERY IMPORTANT:**
- These .mat files are **BINARY** files <u>which means that you CAN NOT EDIT OR READ THEM like you can with ascii files. SO DO NOT ATTEMPT TO DO THAT</u>. Reading the saved information MUST be done using the MATLAB *Import Data* option discussed below.
- Type **clear** to delete your workspace. Type **who** to check that your workspace is empty (or check your Workspace window). Retrieve the saved workspace using *Import Data* under *File*.