Aum Amriteswaryai Namah

Name:…………………………
Roll No:.………………………

Amrita Viswavidyapeetham Amritapuri Campus

21AIE315 AI in Speech Processing

## LABSHEET 1: Introduction to MATLAB

### MATRIX, the foundation for MATLAB

The basic element of data storage in MATLAB is a matrix. All data are stored in form of matrices. A matrix with one row and one column is called a **scalar** eg: A=[4]. When a matrix has only one row or one column we refer it as a **'row vector'** or a '**column vector**' respectively. A matrix with equal number of rows and column is called a **square matrix**.

Matrices can be **initialized** in different ways as follows:

**Method 1**: Seperating rows using semicolon

C = [-1, 0, 0; 1,1,0; 1,-1,0; 0,0,2] ; *Ques: What is the use of the semicolon at the end ?Is it compulsory to have a semicolon at the end of every statement like 'C' and 'C++'.*

**Method 2:** Listing each row on a separate line

```
C =     [-1, 0, 0
          1, 1, 0
          1, -1, 0
          0, 0,  2]
```

If there are too many numbers in a row of the matrix to fit on one line, you may continue the statement on the next line but a comma and 3 periods (an **ellipsis**) are needed at the end of the line to indicate that the row is to be continued eg: a row vector F with 10 values can be represented as

F = [ 1, 2, 3, 4, 5,6, 7, 8, 9, 10]      or

```
F = [ 1, 2, 3, 4, 5, …
      6, 7, 8, 9, 10]
```

**Method 3:** using Another Matrix

B = [3,5,6]

S = [7, 8, **B**]   *This is equivalent to saying S = [7,8,3,5,6]*

The matrix can be extended/modified by defining new elements eg:

S(2) = 2   *(change value of 2nd index)*,   S(3:4) = [6,3]   (*change range of values*)

S(6) = 9  (*add an element at end*)

S(9) = 13 (*add a 9th element*; Since **index 7 and 8 don't exist, they automatically get initialized to 0**).

I = [1,2,4] ;   S(I) = 42  => Change 1st, 2nd and 4th element to 42 ie: *changing an arbitrary subset of a matrix.*

S(I+1) = 45 => change 2nd, 3rd and 5th elements to 45

**Method 4**: Loading/storing from/To Files
2 types of file formats are supported; the MAT- files (used only by matlab programs) and ASCII files (data shared between matlab and outside world eg: notepad, wordpad).

**The Colon Operator:**

**The colon operator is a very powerful operator for creating new matrices. The different uses are given below.**

- It can be used to **create vectors from matrices**. When a colon is used in place of a specific subscript, the colon represents the entire row or column. Eg: **X = data (: , 1)** stores the entire 1st column. *This is used to extract rows and columns from a given matrix.*

- When a colon is used to separate 2 integers, all the integers between the 2 specified integers are generated. Eg **H = 1:8** generates a vector H containing integers from 1 to 8.

- When colons are used to separate 3 three numbers, values between the first and third numbers are generated using the second number as increment *(This is very important and will be used very often)* eg: **Time=0.0:0.5:5.0** generates a row vector named '**Time**' containing numbers from 0 to 5 in increments of 0.5
  **Values = 10 : -1 : 0** (increment can be negative also)
  **odd = 0: 0.7 :2** (can be done even if the end value is not a multiple)

.

**MATRIX Operators:**

Empty matrix can also be created eg: A = []; B = 4 : -1 : 5; A and B are created as empty matrices.
Transpose of a matrix : **A'**
Inverse of a matrix : inv(A)
Dot product : A.*B
Add,sub,mul,div, exp : + - * / ^
Matrix multiplication : A*B
Element by element multiplication : **.*** (if we don't use 'dot' operator, then matrix multiplication is done)
*(element by element operation is also called as 'Array Operation'. For scalars, array operation and matrix operation produce the same result)*
Element by element div, exp: **./ .^**

**Special Matrices:**

**Matrix of Zeros:** The **zeros** function generates a matrix containing all zeros. If argument is a scalar, generates a square matrix of that size. If argument contains 2 scalars, generates matrix of that size
Eg: zeros (6) : 6 by 6 matrix of zeros
Zeros (3,2) : 3 by 2 matrix of zeros

**Matrix of Ones**: Exactly similar to zero matrix. Instead of zeros, ones matrix is produced.
Eg: ones (6) : 6 by 6 matrix of ones
ones (3,2) : 3 by 2 matrix of ones

**Identity matrix** : An identity matrix is one with ones in the main diagonal and zeros elsewhere . Identity matrix is produced using **'eye'** function.
Eg: eye(6) produces a 6x6 identity matrix
eye(3,2) produces a 3x2 identity matrix

**Diagonal matrix**: **'diag'** function can be used to create a diagonal matrix or extract one of the diagonals of the matrix. Eg: C = [1, 2, 3 ; 4,5,6; 7,8,9]

**diag (C)** gives ans= [1,5,9]

**Other diagonal elements** may be extracted by **passing a second parameter 'k'**. K denotes the position of the diagonal from the main diagonal.
**diag (C,1)** gives ans=[2,6]

**Diagonal matrices can also be created** if an input vector (one row vector is provided)
**V = [1, 3, 4, 5]**
**diag(V,0) :** Creates a 4 x 4 matrix with diagonal elements as 1, 3, 4, 5

**Important MATLAB commands:**

clc (clears workspace)
Who (prints all the current variables in memory eg: The matrixes that have been defined till that point)
Whos (prints all the current variables in memory along with their sizes)
Clear (removes all variables from memory)

**Important MATLAB functions:**

**The following lists the most commonly used matlab functions.**
Typing *'help function_name' will provide all the information that you require to use the function.*
*Eg: help sqrt*

*NOTE: The argument 'x' can be either a scalar or a vector depending upon the function*

sqrt (x)
size (x) ; returns 2 scalar arguments representing number of rows and columns
abs(x) ; absolute value of x
sign(x)
exp(x)
log(x) ; computes natural logarithmof x to base e
log10(x) ; computer norma logarithm of x to base 10
sin(x) ; for trigonometric function angle 'x' should be given in radians and not in degrees
cos(x) ; 180 degrees = Π radians
tan(x) ;

max(x)
max(x,y) ; return an matrix the same size as x and y with each element as the max value from the corresponding values from x and y.
min(x)
min(x,y)
sum(x)
conj(x) ; **Complex numbers are represented as eg: x = [2+3j 4+5j 7+6j]**
conj(x) produces the complex conjugate of x
**abs(x)** gives magnitude, **phase(x)** gives phase/angle of the number in radians.

rand(n) ; returns a nxn matrix filled with random numbers uniformly distributed b/w **0 and 1**
rand(m,n) ; returns a m x n matrix filled with random numbers uniformly distributed b/w **0 and 1**

<u>**Hints/ points to note**</u>**:**

- Matlab is **case sensitive**.

- Use a **semicolon** at the end of a statement **to suppress the output from being printed** on to the screen.
- All variable names must start with a letter and may contain letters, digits and the underscore character.
- When an expression is entered without specifying a variable to store the result, the result or answer is automatically stored in a **default variable called 'ans'**. Each time a new value is stored in 'ans', the previous value is lost. Eg typing [1 3 5] will store this matrix in a variable called 'ans'.
- Matrix index starts with 1. **'0' index doesn't have meaning in matrix**. Eg: X[0] is not defined in MATLAB. Be careful with this since C/C++ programmers use 0 index often.
- Symbol for adding comments is '%'

Name and Class No.: . . . . . . . . . . . . . . . . . . . . . . . .

# LABSHEET 1
# Introduction to MATLAB

## Lab Exercises

1. Type the following commands and observe closely what happens after each instruction.

   a. help sqrt
   b. x = sqrt(4)
   c. y = sqrt(9) ;   % what do you see ? Or do you see anything?
   d. who
   e. whos
   f. z = sin(3.14)
   g. whos
   h. clc
   i. who
   j. clear
   k. who
   l. size(x)
   m. a = [1 2 3]
      b = [3 4 5]
      a*b                                        Think!!!!!!!!
      a.*b      % Good day friends. My name is Manokanth! I left my brain in the football field yesterday and can't think now. Can you please tell me what is the difference b/w these 2 expressions?  Will any of them give errors?
   n. C = [a b]                                  % concatenation operation
   o. zeros(4)
   p. zeros(1, 3)
   q. ones(1, 2)
   r. P = [zeros(1,3) 1 1 ones(1,2)]
   s. P = [zeros(3,1) 1 1 ones(1,2)]     % gives error why? Hint: can you concatenate any 2 arrays. What is the condition that needs to be satisfied before 2 matrices can be concatenated.
   t. Y = 1 : 0.5 : 5
   u. M = [1 2 3
               4 5 6
               7  8 9]
      fprintf('%f\n', M);   % hmm…how will this print?
      X = M(1,:)              % what is the matrix X ?  (using colon operator)

2. A matrix is given as below

$$A = \begin{pmatrix} 3 & 12 & 6 & 8 \\ 5 & 3 & 9 & 11 \\ 1 & 2 & 14 & 7 \\ 10 & 5 & 3 & 6 \end{pmatrix}$$

a. Generate the matrix in MATLAB using the 4 different methods mentioned in this document.

b. Find the transpose and inverse of A.

c. Use size command to find out the size of the matrix

d. Extract the second row of A into a vector X and third column of A into a vector Y.

e. Multiply the matrices X and Y

f. Multiply the matrices X and Y element by element.

g. Replace the last row by the vector [1 5 9 0]

h. Print the elements of the matrix in one row, in one column.

i. Print the elements of the matrix in the reverse order.

j. Extract the diagonal elements of A in to a vector D.

k. Create a diagonal matrix V with the extracted diagonal elements.

l. Create an identity matrix of the same size as V

m. Concatenate the diagonal matrix with the identity matrix and create a new matrix of double the size.

3. Create a vector with the first element zero and last element one, with each element being separated by 0.01. Find the length of the vector that you created.

4. Create vectors x = [1 6 9 2] and y = [2 0 3 8]. Find the element wise sum and product of x and y.

5. Use the command who and whos. What is the difference between these two?

6. Find the sum, mean and median of all elements of the Matrix U=[4 5 6 7]