Al-Mustaqbal University
College of Engineering and Technology
Department of Medical Instrumentation Techniques Engineering
Class: First Class
Subject: Computer Programming and application I
Lecturer: Dr.Myasar Mundher Adnan
Lecture Address: Specialized Matrix Functions Sum , fliplr and other function
2024 - 2025

Specialized Matrix Functions Sum , fliplr and other function

To enter matrix, simply type in the Command Window A

= [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered:

```
A =
   16    3    2   13
    5   10   11    8
    9    6    7   12
    4   15   14    1
sum(A)
```

MATLAB replies with ans

```
=
   34   34   34   34
```

When you do not specify an output variable, MATLAB uses the variable ans, short for *answer*, to store the results of a calculation. You have computed a row vector containing the sums of the columns of A. Each of the columns has the same sum, the *magic* sum, 34.

How about the row sums? MATLAB has a preference for working with the columns of a matrix, so one way to get the row sums is to transpose the matrix, compute the column sums of the transpose, and then transpose the result. For an additional way that avoids the double transpose use the dimension argument for the sum function.

MATLAB has two transpose operators. The apostrophe operator (e.g., A') performs a complex conjugate transposition. It flips a matrix about its main diagonal, and also changes the sign of the imaginary component of any complex elements of the matrix. The dot-apostrophe operator (e.g., A.'), transposes without affecting the sign of complex elements. For matrices containing all real elements, the two operators return the same result.

A' produces

```
ans =
16     5     9     4
3      10    6     15
2      11    7     14
13     8     12    1
```

and sum(A')'

produces a column vector containing the row sums

```
ans =
   34
   34
   34
   34
```

The sum of the elements on the main diagonal is obtained with the sum and the diag functions:

diag(A) produes

```
ans =
  16
  10
   7
   1
```

and sum(diag(A)) produces

ans =34

The other diagonal, the so-called *antidiagonal*, is not so important mathematically, so MATLAB does not have a ready-made function for it. But a function originally intended for use in graphics, fliplr, flips a matrix from left to right:

sum(diag(fliplr(A))) ans =34

Table 2.2: MATLAB functions for matrices

Command	Return
`a'`	Transpose of a
`find(a)`	Indices of all non-zero elements in a.
`fliplr(a)`	Matrix a, flipped horizontally
`flipud(a)`	Matrix a, flipped vertically.
`inv(a)`	Inverse of a
`min(a)`	Minimum-valued element of a. †
`max(a)`	Maximum-valued element of a. †
`numel(a)`	The number of elements of a.
`repmat(a,m,n)`	A matrix where matrix a is repeated in m rows and n columns
`reshape(a,m,n)`	Matrix a reshaped into m rows and n columns.
`size(a)`	The size of a (#rows, #columns, ...)
`sort(a)`	Vector a sorted into ascending order. †
`sum(a)`	Sum of elements of a. †
`unique(a)`	The list of unique elements of a in ascending order.

† For a matrix, the operation will be carried out separately on each column. For a vector (row or column), the operation will be carried out on the vector.