Computer Science II

Environmental Engineering
Second Level
2024-2025
1st Course

COLLEGE OF ENGINEERING - كلية الهندسة
Tikrit University - جامعة تكريت

Lecturer Assist. Aalaa Ahmed

طريقك إلى النجاح
YOUR WAY TO SUCCESS

Lecture #5

Programming with MATLAB (Cont.)

COLLEGE OF ENGINEERING - كلية الهندسة
Tikrit University - جامعة تكريت

Lecturer Assist. Aalaa Ahmed

13. MATRICES (Cont.)

Entering a matrix:

A matrix is an array of numbers. To type a matrix into MATLAB you must

- begin with a square bracket, [
- separate elements in a row with spaces or commas (,)
- use a semicolon (;) to separate rows
- end the matrix with another square bracket,].

13. MATRICES (Cont.)

Entering a matrix (Cont.):

Ex: This technique work for two-dimensional matrices as well

```
>> k=[ 8  9  10, 1  2  20, 4  3  30]
k= 8   9  10
   1   2  20
   4   3  30
```

Note that the use of semicolons (;) here is different from their use mentioned earlier to suppress output or to write multiple commands in a single line.

13. MATRICES (Cont.)

Entering a matrix (Cont.):

Once we have entered the matrix, it is automatically stored and remembered in the Workspace. We can refer to it simply as matrix k. We can then view a particular element in a matrix by specifying its location. We write:

>> k (end, end) → ans = 30

>> k (2, end-1:end) → ans = 2 20

>> k(2,1) → ans = 1

k(2,1) is an element located in the second row and first column. Its value is 1.

13. MATRICES (Cont.)

Matrix indexing:

We select elements in a matrix just as we did for vectors, but now we need two indices. The element of row i and column j of the matrix k is denoted by k(i,j). Thus, k(i,j) in MATLAB refers to the element kij of matrix k. The first index is the row number and the second index is the column number. For example, k(1,3) is an element of first row and third column. Here, k(1,3)=10.

13. MATRICES (Cont.)

Matrix indexing (Cont.):

Correcting any entry is easy through indexing. Here we substitute k(3, 3) = 30 by k(3, 3) = 0. The result is:

```
>> k(3, 3) = 0
k= 8   9   10
   1   2   20
   4   3    0
```

13. MATRICES (Cont.)

Deleting rows or columns:

To delete a row or column of a matrix, use the empty vector operator, [ ].

Ex: delete the second column

```
>> a =[1  2  3, 4  5  6, 7  8  9]
a = 1   2   3
    4   5   6
    7   8   9
>> a (:, 2) = [ ]   →   a = 1   3
                           4   6
                           7   9
```

13. MATRICES (Cont.)

Deleting rows or columns (Cont.):

Ex: delete the second row

```
>> a (2, :) = [ ]
a =  1   2   3
     7   8   9
```

Second row of matrix A is now deleted. To restore the third row, we use a technique for creating a matrix:

```
>> a = [ a (1, : ) ; [4  5  6] ; a (3,: ) ]
```

13. MATRICES (Cont.)

Dimension:

To determine the dimensions of a matrix or vector, use the command **size**. For example,

```
>> size(A)

ans =

     3   3
```

means 3 rows and 3 columns.

Or more explicitly with,

```
>> [m, n] = size(A)
```

13. MATRICES (Cont.)

Continuation:

If it is not possible to type the entire input on the same line, use consecutive periods, called an ellipsis ..., to signal continuation, then continue the input on the next line.

```
B = [ 4/5      7.23*tan(x)     sqrt(6); ...
      1/x^2          0         3/(x*log(x)); ...
      x-7        sqrt(3)       x*sin(x) ];
```

Note that blank spaces around +, −, = signs are optional, but they improve readability.

13. MATRICES (Cont.)

Transposing a matrix:

The transpose operation is denoted by an apostrophe or a single quote ('). It flips a matrix about its main diagonal and it turns a row vector into a column vector. Thus, >> A' → ans = 1 4 7

2 5 8

3 6 0

By using linear algebra notation, the transpose of m × n real matrix A is the n × m matrix that results from interchanging the rows and columns of A. The transpose matrix is denoted A^T.

13. MATRICES (Cont.)

Concatenating matrices:

Matrices can be made up of sub-matrices. Here is an example. First, let's recall our previous matrix A.

A = 1 2 3
    4 5 6
    7 8 9

The new matrix B will be,

>> B = [A 10*A ; -A [1 0 0; 0 1 0; 0 0 1]]

13. MATRICES (Cont.)

Concatenating matrices (Cont.):

```
>> B = [A   10*A ;  -A  [1  0  0 ; 0  1  0; 0   0  1] ]

B = 1   2   3   10   20   30
     4   5   6   40   50   60
     7   8   9   70   80   90
    -1  -2  -3    1    0    0
    -4  -5  -6    0    1    0
    -7  -8  -9    0    0    1
```

13. MATRICES (Cont.)

Matrix generators:

MATLAB provides functions that generate elementary matrices.

Elementary matrices	
eye(m,n)	Returns an m-by-n matrix with 1 on the main diagonal
eye(n)	Returns an n-by-n square identity matrix
zeros(m,n)	Returns an m-by-n matrix of zeros
ones(m,n)	Returns an m-by-n matrix of ones
diag(A)	Extracts the diagonal of matrix A
rand(m,n)	Returns an m-by-n matrix of random numbers

For a complete list of elementary matrices and matrix manipulations, type **help elmat** or **doc elmat.**

13. MATRICES (Cont.)

Matrix generators (Cont.): Here are some examples:

1. >> b = ones (3, 1)

```
b = 1
    1
    1
    1
```

Equivalently, we can define b as >> b=[1;1;1]

2. >> eye (3)

```
ans = 1   0   0
      0   1   0
      0   0   1
```

13. MATRICES (Cont.)

Matrix generators (Cont.):

3. >> c = zeros (2, 3)

```
c = 0   0   0
    0   0   0
```

4. >> C = [1 2; 3 4];

>> D = [C zeros (2); ones (2) eye (2)]

```
D = 1   2   0   0
    3   4   0   0
    1   1   1   0
    1   1   0   1
```

13. MATRICES (Cont.)

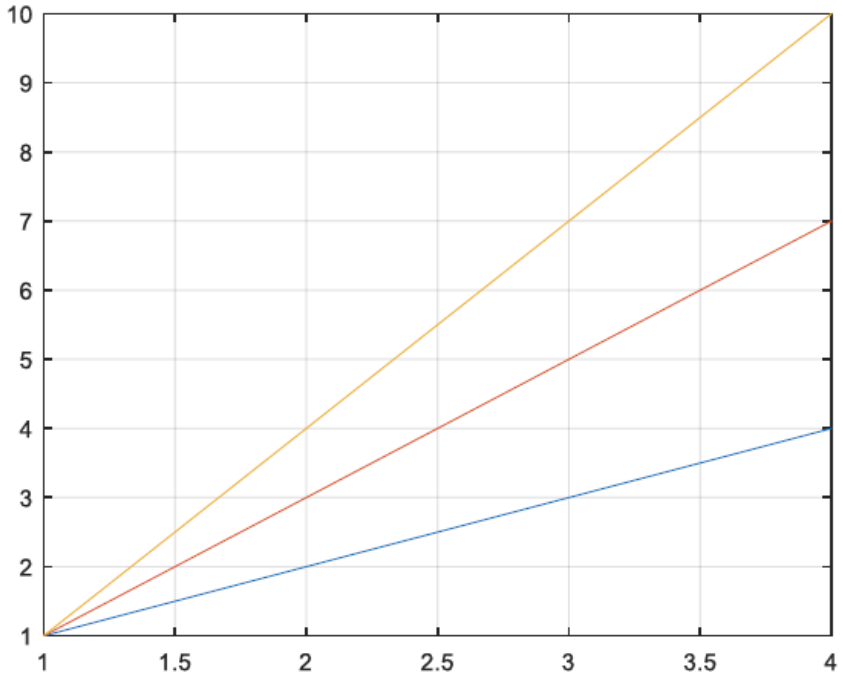
Plotting Matrices:

If one of the arguments of the plot command is a matrix, MATLAB will use the columns of the matrix to plot a set of lines, one line per column.

Ex:- >> q= [1 1 1; 2 3 4; 3 5 7; 4 7 10]

>> plot (q)

>> grid



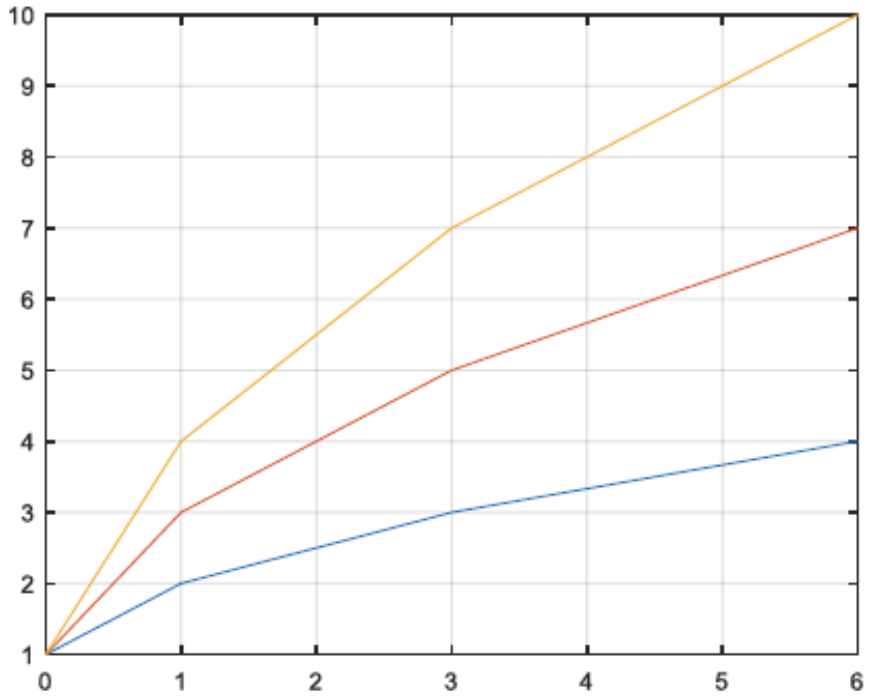
MATLAB plots the columns of the matrix q against the row index.

13. MATRICES (Cont.)

Plotting Matrices (Cont.):

You can also supply an x variable:

```
>> x = [ 0   1   3   6] ;
>> plot (x, q) ;
>> grid
```



13. MATRICES (Cont.)

Plotting Matrices (Cont.):

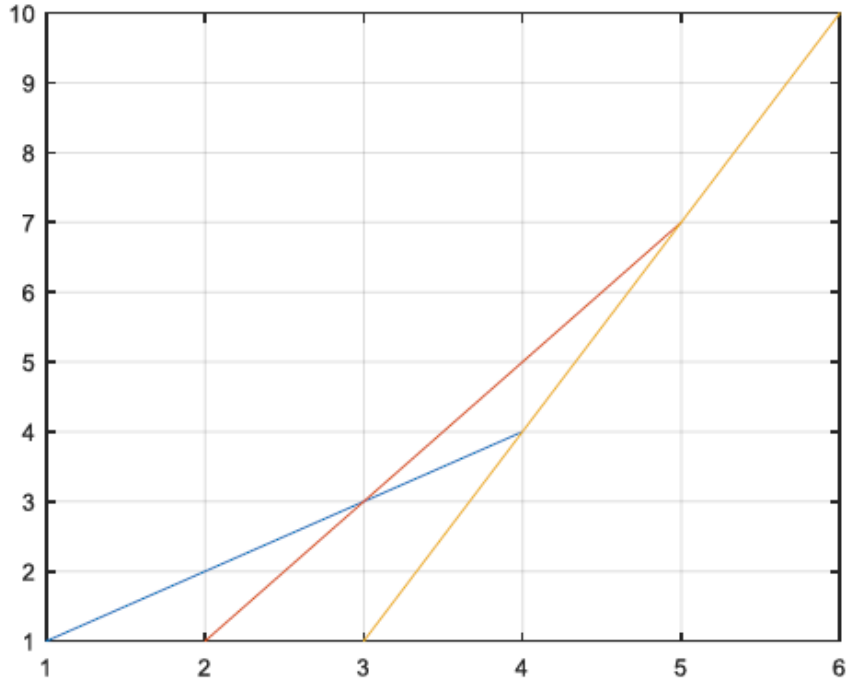
If both the x and y arguments are matrices, MATLAB will plot the successive columns on the same plot.

Ex:- >> q =[1 1 1; 2 3 4; 3 5 7; 4 7 10]

>> x =[1 2 3; 2 3 4; 3 4 5; 4 5 6]

>> plot (x, q)

>> grid



13. MATRICES (Cont.)

Subplots:

To plot more than one set of axes in the same window, use the subplot command. You can type subplot (m, n, p) ,to break up the plotting window into m plots in the vertical direction and n plots in the horizontal direction, choosing the pth plot for drawing into.

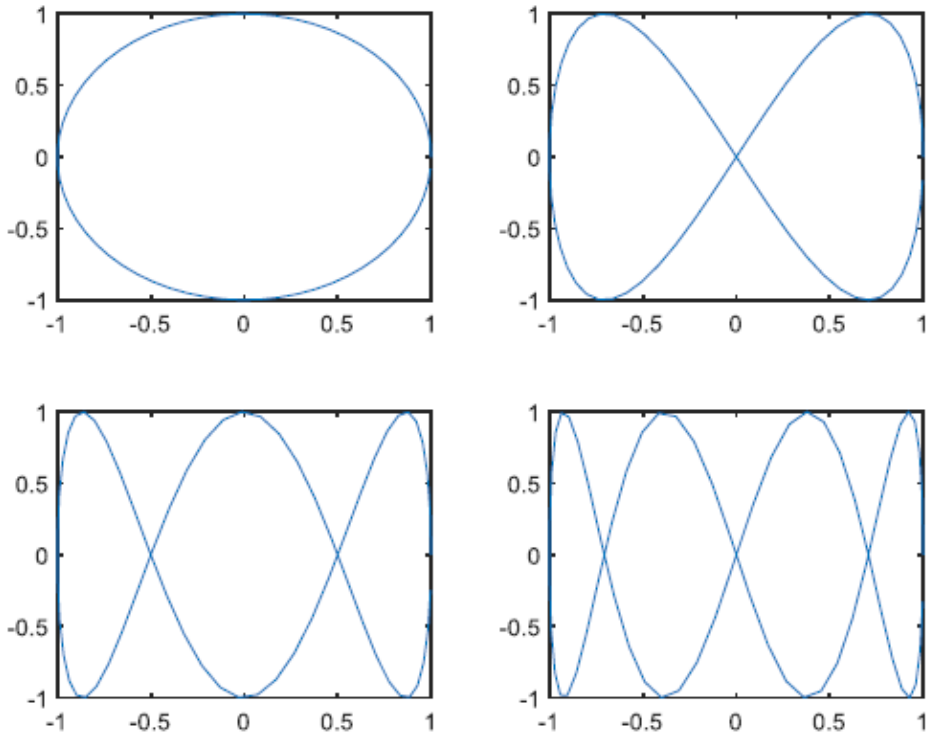
The subplots are counted as you read text: left to right along the top row, then left to right along the second row, and so on.

13. MATRICES (Cont.)

Subplots (Cont.):

Ex:- >> t= 0 : 0.1 : 2*pi;

```
>> subplot (2, 2, 1)
>> plot (cos(t), sin(t))
>> subplot (2, 2, 2)
>> plot (cos(t), sin(2*t))
>> subplot(2, 2, 3)
>> plot(cos(t), sin(3*t))
>> subplot (2, 2, 4)
>> plot (cos(t), sin (4*t))
```



13. MATRICES (Cont.)

Subplots (Cont.):

As long as your subplots are based on an array of 9×9 little plots or less, subplot (2 2 1) or subplot 2 2 1 are equivalent to subplot (2,2,1). You can mix different subplot arrays on the same figure, as long as the plots do not overlap.

Ex:- >> subplot 221

>> plot (1:10)

>> subplot 222

>> plot (0,'*')

>> subplot 212

>> plot ([1 0 1 0])