

Mathematics 3

Lecture 10

MAT 132

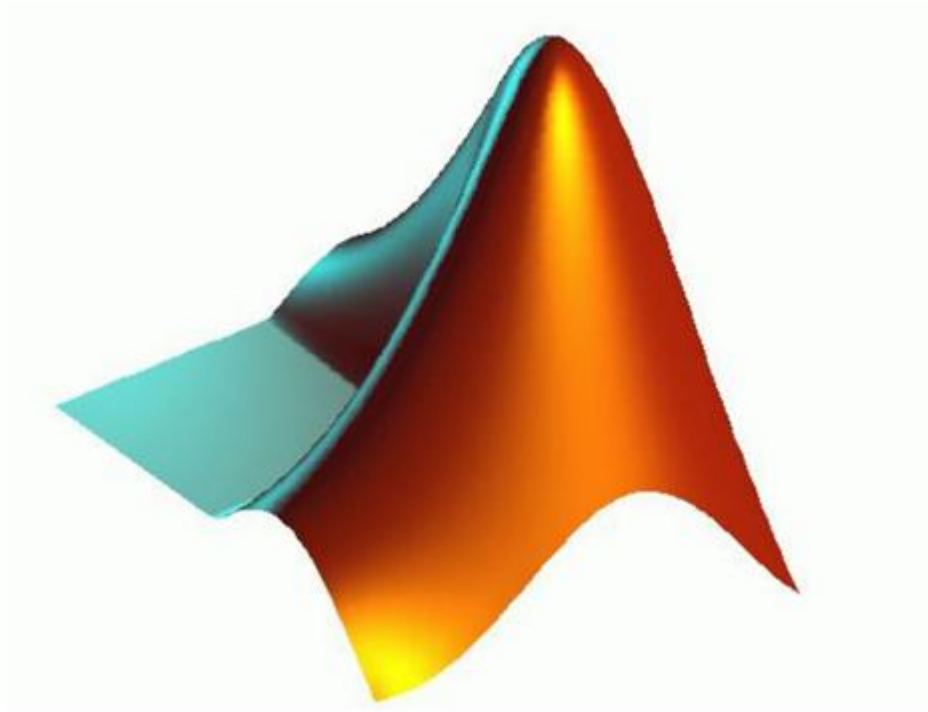
Mathematical Operations by MATLAB

Prepared by
Dr. Israa Lewaa

- 1- Learn the Importance of MATLAB.
- 2- Explain more about Elementary Functions.
- 3- Give a full illustration for Vectors and Matrices.
- 4- Understand how to conduct Matrix Multiplication.
- 5- Perform many Practical Examples on MATLAB

- 1- Importance of MATLAB.
- 2- Elementary Functions.
- 3- Vectors and Matrices.
- 4- Matrix Multiplication.
- 5- Practical Examples on MATLAB

Importance of MATLAB



- Its basic data element is the matrix. A simple integer is considered an matrix of one row and one column. Several mathematical operations that work on arrays or matrices are built-in to the Mat lab environment. For example, cross-products, dot-products, determinants, inverse matrices.
- Vectorized operations. Adding two arrays together needs only one command, instead of a for or while loop.

- The graphical output is optimized for interaction. You can plot your data very easily, and then change colors, sizes, scales, etc, by using the graphical interactive tools.
- Mat lab's functionality can be greatly expanded by the addition of toolboxes. These are sets of specific functions that provided more specialized functionality. Ex: Excel link allows data to be written in a format recognized by Excel, Statistics Toolbox allows more specialized statistical manipulation of data (Anova, Basic Fits, etc)

- Or you can create programs and functions that perform repetitive tasks, just as any other computer language.

**Try a simple operation now:
2 + 2 <enter>**

**Answer is '4' displayed immediately on
command window**

One of the most important features of the MATLAB interface is the help. It is very thorough and you can learn almost anything you need from it.

Help <enter>



The best way for you to get started with MATLAB is to learn how to handle matrices !!!!!

You can enter matrices into MATLAB in several different ways:

1. Enter an explicit list of elements.
2. Load matrices from external data files.
3. Generate matrices using built-in functions.
4. Create matrices with your own functions in M-files.

2- Elementary Functions

Some commonly used functions, where variables x and y can be numbers, vectors, or matrices are listed in the following Table:

Table : Elementary functions

<code>cos(x)</code>	Cosine	<code>abs(x)</code>	Absolute value
<code>sin(x)</code>	Sine	<code>sign(x)</code>	Signum function
<code>tan(x)</code>	Tangent	<code>max(x)</code>	Maximum value
<code>acos(x)</code>	Arc cosine	<code>min(x)</code>	Minimum value
<code>asin(x)</code>	Arc sine	<code>ceil(x)</code>	Round towards $+\infty$
<code>atan(x)</code>	Arc tangent	<code>floor(x)</code>	Round towards $-\infty$
<code>exp(x)</code>	Exponential	<code>round(x)</code>	Round to nearest integer
<code>sqrt(x)</code>	Square root	<code>rem(x)</code>	Remainder after division
<code>log(x)</code>	Natural logarithm	<code>angle(x)</code>	Phase angle
<code>log10(x)</code>	Common logarithm	<code>conj(x)</code>	Complex conjugate

Creating MATLAB variables

Operations	Operators	Examples
Addition	+	>> 5 + 3
Subtraction	-	>> 5 - 3
Multiplication	*	>> 5 * 3
Right Division	/	>> 5/3
Left Division	\	>> 5\3 = 3/5
Exponentiation	^	>> 5^3 (means $5^3 = 125$)

Precedence Order	Operators
1	Parentheses (). For nested parentheses, the innermost
2	are executed first.
3	Exponentiation, ^
4	Multiplication, *; Division, /, \
	Addition, +; Subtraction, -

In addition to the elementary functions, MATLAB includes a number of predefined *constant values*. A list of the most common values is given in the following Table :

Table : Predefined constant values

pi	The π number, $\pi = 3.14159 \dots$
i, j	The imaginary unit i , $\sqrt{-1}$
Inf	The infinity, ∞
NaN	Not a number

Matrices on MATLAB

Input matrix

$$A = [1 \ 2 \ 5; 3 \ 9 \ 0]$$

Use semicolon ";" to suppress output

Use ... for line continuation

- ❑ Vector is a special case of matrix
- ❑ Scalar is considered 1x1 matrix
 $g = 9.81$ (no $[\dots]$ needed)
- ❑ $X = []$ creates a null matrix

1. Separate the elements of a row with blanks or commas.
2. Use a semicolon, **;**, to indicate the end of each row.
3. Surround the entire list of elements with square brackets, **[]**

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

This is what MATLAB displays after you hit <enter>

A =

MAGIC SQUARE



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

Let's prove it is a magic square. **Let's get the sum of all columns by typing `'sum(A)'`**

The answer (ans) is:

ans =

34 34 34 34

This result is a row vector. Each column in A adds up to 34. **That's magic!**

How about the row sums? MATLAB has a preference for working with the columns of a matrix, so the easiest way to get the row sums is to **transpose the matrix**, compute the column sums of the transpose, and then transpose the result. The transpose operation is denoted by an **apostrophe or single quote, '**. It flips a matrix about its main diagonal and it turns a row vector into a column vector.

A'

ans =

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

Now: **sum(A)'** produces a column vector containing the row sums

ans =

34
34
34
34

❖ Example

As the value of the expression $y = e^{-a} \sin(x) + 10\sqrt{y}$, for $a = 5$, $x = 2$, and $y = 8$ is computed by

As the value of the expression $y = e^{-a} \sin(x) + 10\sqrt{y}$, for $a = 5$, $x = 2$, and $y = 8$ is computed by

```
>> a = 5; x = 2; y = 8;  
>> y = exp(-a)*sin(x)+10*sqrt(y)  
y =  
    28.2904
```

❖ Example

To calculate $\sin(\pi/4)$ and e^{10} , we enter the following commands in MATLAB,

To calculate $\sin(\pi/4)$ and e^{10} , we enter the following commands in MATLAB,

```
>> sin(pi/4)
```

```
ans =  
    0.7071
```

```
>> exp(10)
```

```
ans =  
2.2026e+004
```

OPERATION	MATRIX	ARRAY
Addition	+	+
Subtraction	−	−
Multiplication	*	.*
Division	/	./
Left division	\	.\
Exponentiation	^	.^

Table 1.1: Summary of matrix and array operations

produces another matrix **C** of the same size with elements $c_{ij} = a_{ij}b_{ij}$. For example, using the same 3×3 matrices,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

Extra Practice – Example 3 Solution

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

we have,

```
>> C = A.*B
C
    10    40    90
   160   250   360
   490   640   810
```

Extra Practice – Example 4

$$\begin{array}{c} \text{A} \\ \begin{bmatrix} 3 & 8 & 0 \\ 1 & 2 & 5 \end{bmatrix} \end{array} * \begin{array}{c} \text{B} \\ \begin{bmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 1 & 2 \\ 0 & 2 & 3 & 0 \end{bmatrix} \end{array} = \begin{array}{c} \text{C} \\ \begin{bmatrix} 35 & 46 & 17 & 19 \\ 9 & 22 & 20 & 5 \end{bmatrix} \end{array}$$

In MATLAB, the * operator will perform this matrix multiplication:

$$\begin{matrix} & A & & B & & C \\ \begin{bmatrix} 3 & 8 & 0 \\ 1 & 2 & 5 \end{bmatrix} & * & \begin{bmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 1 & 2 \\ 0 & 2 & 3 & 0 \end{bmatrix} & = & \begin{bmatrix} 35 & 46 & 17 & 19 \\ 9 & 22 & 20 & 5 \end{bmatrix} \end{matrix}$$

In MATLAB, the * operator will perform this matrix multiplication:

```
>> A = [3 8 0; 1 2 5];  
>> B = [1 2 3 1; 4 5 1 2; 0 2 3 0];  
>> C = A*B  
C =  
    35    46    17    19  
     9    22    20     5
```

For example, consider the following system of linear equations

$$\begin{cases} x + 2y + 3z &= 1 \\ 4x + 5y + 6z &= 1 \\ 7x + 8y &= 1 \end{cases}$$

Extra Practice – Example 5 Solution

$$\begin{cases} x + 2y + 3z &= 1 \\ 4x + 5y + 6z &= 1 \\ 7x + 8y &= 1 \end{cases}$$

The coefficient matrix A is

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{and the vector } b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

With matrix notation, a system of simultaneous linear equations is written

1. The first one is to use the matrix inverse, `inv`.

```
>> A = [1 2 3; 4 5 6; 7 8 0];  
>> b = [1; 1; 1];  
>> x = inv(A)*b  
x      =  
      -1.0000  
       1.0000  
      -0.0000
```

Extra Practice – Example 5 Solution

```
>> A = [1 2 3; 4 5 6; 7 8 0];  
>> inv(A)  
ans =  
    -1.7778    0.8889   -0.1111  
     1.5556   -0.7778    0.2222  
    -0.1111    0.2222   -0.1111
```

which is similar to:

$$A^{-1} = \frac{1}{9} \begin{bmatrix} -16 & 8 & -1 \\ 14 & -7 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

Vectors and Matrices

- Vectors (arrays) are defined as
- `>> v = [1, 2, 4, 5]`
- `>> w = [1; 2; 4; 5]`

- Matrices (2D arrays) defined similarly
- `>> A = [1,2,3;4,-5,6;5,-6,7]`

$$v = [1 \ 2 \ 4 \ 5]$$

$$w = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 5 \end{bmatrix}$$

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

❑ `t=1:10`

t =

1 2 3 4 5 6 7 8 9 10

❑ `k=2:-0.5:-1`

k =

2 1.5 1 0.5 0 -0.5 -1

❑ `B = [1:4; 5:8]`

x =

1 2 3 4
5 6 7 8

Extra Practice – Example 6

Given A and B:

>> A = [1 2 3;4 5 6;7 8 9]									>> B = [3 5 2; 5 2 8; 3 6 9]								
A =									B =								
1	2	3							3	5	2						
4	5	6							5	2	8						
7	8	9							3	6	9						

Extra Practice – Example 6 Solution

Addition

```
>> X = A + B
```

X =

4	7	5
9	7	14
10	14	18

Subtraction

```
>> Y = A - B
```

Y =

-2	-3	1
-1	3	-2
4	2	0

Product

```
>> Z = A * B
```

Z =

22	27	45
55	66	102
88	105	159

Transpose

```
>> T = A'
```

T =

1	4	7
2	5	8
3	6	9

Extra Practice – Example 7

$$-x_1 + x_2 + 2x_3 = 2$$

$$3x_1 - x_2 + x_3 = 6$$

$$-x_1 + 3x_2 + 4x_3 = 4$$

Extra Practice – Example 7 Solution

```
» A = [-1 1 2; 3 -1 1;-1 3 4];
```

```
» b = [2;6;4];
```

```
» x = inv(A)*b
```

```
x =
```

```
1.0000
```

```
-1.0000
```

```
2.0000
```

```
» x = A\b
```

```
x =
```

```
1.0000
```

```
-1.0000
```

```
2.0000
```

$$-x_1 + x_2 + 2x_3 = 2$$

$$3x_1 - x_2 + x_3 = 6$$

$$-x_1 + 3x_2 + 4x_3 = 4$$

- 1- Importance of MATLAB.
- 2- Elementary Functions.
- 3- Vectors and Matrices.
- 4- Matrix Multiplication.
- 5- Practical Examples on MATLAB

Success Tips for Math 3 Students

1 Math is not about memorizing ... Math is about understanding

Every rule in differentiation or integration has a geometric and practical meaning. Understand **why** we differentiate and **why** we integrate. Plot the function before solving — you will understand **50% of the problem** before you even start.

2 Solve problems every day, even just one

Consistency matters more than quantity. One problem a day is better than 30 problems the night before the exam. Remember:
Math = Practice × Practice

3 Use MATLAB to Explore, Not to Replace Thinking

MATLAB is a powerful tool to help you understand complex concepts like:

Functions and plotting

Linear algebra operations

Numerical methods

Solving differential equations

Matrix computations

Use MATLAB to **visualize**, **check your work**, and **experiment**, not as a shortcut.

Example:

```
f = @(x) x.^3 - 5*x;  
fplot(f, [-5 5])  
grid on
```

4 Create a One-Page Summary After Each Lecture

Keep it short and useful.

Include:

Key formulas

Important examples

Common mistakes

Two practice questions

Your summary sheet will save you **half the study time** at the end of the semester.

5 Never Hesitate to Ask Questions

Most students get stuck in Chapter 2 or 3 simply because they don't ask.

There is **no such thing as a silly question** — the real mistake is continuing without understanding.



Thank You