Lab 1: Introduction

What is Matlab?

- A computer programming language and software environment to manage data interactively.
- Originally developed in 1970s for applications involving matrices, linear algebra and numerical analysis.
- Maintained and sold by the MathWorks, Inc.
- Functionality: manages variables, import/export data, calculations, generate plots, and more …
- Toolboxes such as image processing, statistics, control, financial analysis, and more …

Matlab as an Interactive Calculator:

```
>> 8/10

ans =

    0.8000

>> 5*ans

ans =

     4

>> r=8/10

r =

    0.8000

>> r

r =

    0.8000

>> s=20*r

s =

    16
```

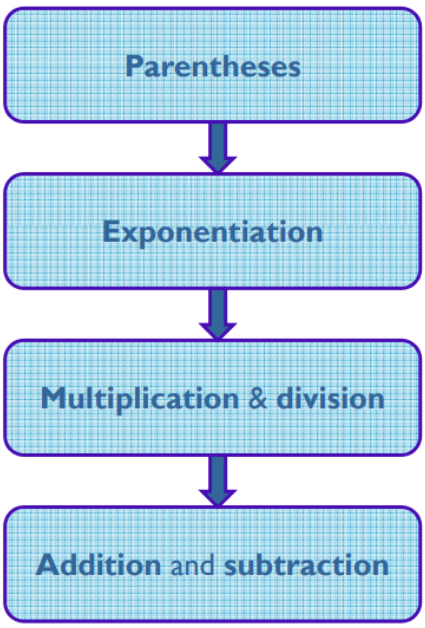
- ***ans*** *is a built-in special variable to store the result of the last computation. It can be reused by typing* ***ans.***
- *Variables defined by the assignment operator '=' are stored in memory and can be used again using their names.*

- Matlab retains your previous keystrokes.
- Use the up-arrow key to scroll back through the commands.
- Press the key once to see the previous entry, and so on.
- Use the down-arrow key to scroll forward. Edit a line using the left and right arrow keys, the backspace key, and the delete key.
- Press Enter to execute the command.

Scalar Arithmetic Operations

Symbol	Operation	Mathematical Syntax	Matlab Syntax
^	Exponentiation	a^b	a^b
*	Multiplication	ab	a*b
/	Forward Division	a/b	a/b
\	Backward Division	a\b	a\b
+	Addition	a+b	a+b
-	Subtraction	a-b	a-b

Order of Precedence



Note: if precedence is equal, evaluation is performed from left to right.

Examples on Precedence

```
 >> 8+3*5

ans =

    23

>> 8+(3*5)

ans =

    23

>> (8+3)*5

ans =

    55

>> 4^2-12-8/(4*2)

ans =

     3

>> 3*4^2+5

ans =

    53

>> (3*4)^2+5

ans =

   149

>> 27^(1/3)+32^(0.2)

ans =

     5

>> 27^1/3+32^(0.2)

ans =

    11
```

The Assignment Operator '='

- Typing x = 3 assigns the value 3 to the variable x.
- We can type x = x+2. This assigns the value 3+2=5 to x.
- In algebra we can write x+2=20, but in Matlab we cannot.
- In Matlab the left side of the assignment operator '=' must be a single variable.
- The right side must be a computable value.

Variables in Matlab

- No declaration or dimension statements are required to define variables in Matlab.
- To define a new variable, simply write the variable name followed by the assignment operator and the value to be stored in the variable.
- If the variable already exists, Matlab changes its contents and, if necessary, allocates new storage.
- To view the value of the variable, type its name on the command prompt and hit Enter.
- Variable names can consist of letters, digits, and underscores, but they should start with a letter. Variable names are case sensitive.

Special Variables and Constants

Variable	Description
ans	Temporary variable containing the most recent answer
i and j	The imaginary unit $\sqrt{-1}$
inf	Infinity
NaN	Indicates an undefined numerical result
pi	The number π

Commonly used Mathematical Functions

Function	Matlab Syntax
e^x	exp(x)
$\sqrt{x}$	sqrt(x)
ln x	log(x)
$\log_{10} x$	log10(x)
cos x	cos(x)
sin x	sin(x)
tan x	tan(x)
$\cos^{-1} x$	acos(x)
$\sin^{-1} x$	asin(x)
$\tan^{-1} x$	atan(x)
|x|	abs(x)

NOTES

- *Trigonometric functions in Matlab use radian measure*
- *$\cos^2(x)$ is written (cos(x))^2 in Matlab*

Complex Number Operations

- The number c1 = 1-2i is entered without the asterisk sign,
- The asterisk sign is not needed between i or j and a number, however it is required with a variable, such as: c2 = 5-i*c1.
- Be careful: y = (7/2)i = 3.5i while x = 7/(2i) = -3.5i.

Command for managing the work session

Command	Operation
clc	Clears the command window
clear	Clears all the variables from the workspace
clear v1 v2	Clears variables v1 and v2 only
exist('var')	Check if the variable or function exist
quit	Exits Matlab session
who	List the names of the variables defined in the work space
whos	List the names of the variables defined in the work space and their details
Semicolon (;)	Suppresses the display of results in the command window
Ellipsis (…)	Line continuation

Numeric Display Formats

Format	Operation	Example
short	Four decimal digits (the default format);	13.6745
long	16 decimal digits;	17.27484029463547
short e	Five digits (four decimals) plus exponent	6.3792e+03
long e	16 digits (15 decimals) plus exponent;	6.379243784781294e–04

Command Resolving in Matlab

What happens when you type Problem1 in the Command Window?

1. Matlab first checks to see if Problem1 is a **variable** and if so, displays its value.
2. If not, then Matlab checks to see if Problem1 is on of its **own commands,** and executes it.
3. If not, then Matlab looks in the current directory for a **script file** named Problem1.m and executes it.
4. If not, then Matlab displays an error.

Help Functions

- `help funcname`: Displays in the command window a description of the specified function funcname.
- `Lookfor topic`: Displays in the command window a brief description for all functions whose description includes the specified keyword topic.

Introduction to Matlab Arrays

- Matlab has the capability of handling arrays of numbers and many data types.
- Array manipulation in Matlab is so much simpler when compared to other programming languages (C=A+B is done without writing loops)
- This makes Matlab the choice for many engineering applications that require processing of data sets.

Frequently used Arrays in Matlab

- Numeric:
  - Single and Double Precision
  - Signed Integers

- Unsigned Integers
- Character: Array of strings
- Logical: contains '1' or '0' that corresponds to True or False.

Vectors in Matlab

- Vectors are a special case of arrays, with one of its dimensions being 1: it is either a row vector or a column vector.
- Note that scalars in Matlab are treated as vectors of size 1x1.

Creating Vectors in Matlab

- To create a row vector, separate the elements by commas or spaces:

```
>> p = [3,7,9]

p =

     3     7     9

>> q = [5 9 6]

q =

     5     9     6
```

- You can create a column vector by using the transpose notation (‘):

```
>> p = [3,7,9]'

p =

     3
     7
     9
```

- You can also create a column vector by separating elements by semicolons:

```
>> g = [3;7;9]

g =

     3
     7
     9
```

- You can create larger vectors by appending one vector to another. For example, to create a row vector u whose first three columns contain the values of r=[2 4 20] and whose fourth, fifth, and sixth columns contain the values of w=[9 -6 3], you type u=[r w]. The result is vector u=[2 4 20 9 -6 3].
- Note that in order to append vectors they should be either row vectors or column vectors. Otherwise the transpose notation is required before appending.
- The colon operator (:) easily generates a large vector of regularly spaced elements.

Syntax: x = [m:q:n], where m is the first element, n is the last element, and q is the increment. If m-n is an integer multiple of q, then the last value of the vector is n, if not, the last value of the vector is less than n. If the increment q is not specified, the default value is 1.
Number of Elements = ((n-m)/q)+1

```
>> x = [0:2:8]

x =

     0     2     4     6     8

>> y = [0:2:7]

y =

     0     2     4     6

>> z = [-3:2]

z =

    -3    -2    -1     0     1     2
```

- The **linspace** command also creates a linearly spaced row vector, but instead you specify the number of elements rather than the increment. If n is not specified, the default number of elements is 100.
  Syntax: linspace(x1,x2,n), where x1 and x2 are the lower and upper limits and n is the number of elements.
  Increment = (x2-x1)/(n-1)
- Note: linspace(5,8,31) is equivalent to [5:0.1:8].

Magnitude and Length of a Vector

- The **length(v)** command gives the number of elements in a vector.
- The magnitude of a vector should be calculated manually. There is no function to do it.

Matrices

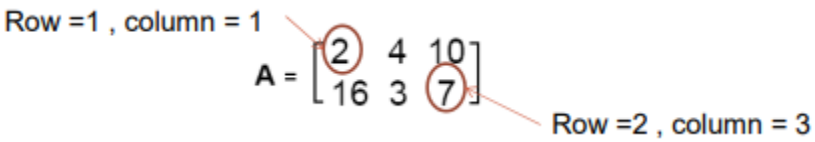
- A matrix has multiple rows and columns.
- Vectors are special cases of matrices having one row or one column.

Creating Matrices in Matlab

- For small matrices, type the elements such that elements in the same row are separated by commas or spaces, and rows are separated by semicolons.

```
>> A = [2 4 10; 16 3 7]

A =

     2     4    10
    16     3     7



>> a = [1 3 5]

a =

     1     3     5

>> b = [7 9 11]

b =

     7     9    11

>> c = [a b]

c =

     1     3     5     7     9    11

>> d = [a;b]

d =

     1     3     5
     7     9    11
```

Addressing Array and Vector Elements



- To access a single element in a matrix: A(row number, column number)
- To access a single element in a vector: v(element position)
- To access a group of elements, rows, columns, or subarrays of arrays use the colon symbol (:)
  - v(:) represents all the row or column elements in a vector

- v(2:5) represents the second to fifth row or column elements in a vector
- A(:,3) denotes all the elements in the third column of matrix A
- A(:,2:5) denotes all the elements in the second to the fifth columns of matrix A
- A(2:3,1:3) denotes all the elements in the second to the third rows that are also in the first to the third columns of matrix A

```
>> B  = [2 4 10 13;16 3 7 18; 8 4 9 25; 3 12 15 17]

B =

     2     4    10    13
    16     3     7    18
     8     4     9    25
     3    12    15    17

>> C = B(2:3,1:3)

C =

    16     3     7
     8     4     9
```

Additional Array Functions

Function	Description
size(A)	Returns a row vector [m n] containing the size of the mxn array A
sort(A)	Sorts each column of the array A in ascending order and returns an array of same size as A
sum(A)	Sums the elements in each column of the array A and returns a row vector containing the sums
inv(A)	Computes the inverse of array A
diag(A)	Returns the elements along the main diagonal of A
fliplr(A)	Flips array A about it central column
flipud(A)	Flips array A about it central row

Exercise: open the Matlab help browser and read the documentation of the following array-related functions: find(A), max(A) , min(A), cat(n,A,B,C)

```
>> A = [3 0 6 1 ; 0 2 4 5 ; 7 2 0 4]

A =

     3     0     6     1
     0     2     4     5
     7     2     0     4

>> size(A)

ans =

     3     4

>> sort(A)

ans =

     0     0     0     1
     3     2     4     4
     7     2     6     5

>> sum(A)

ans =

    10     4    10    10
```