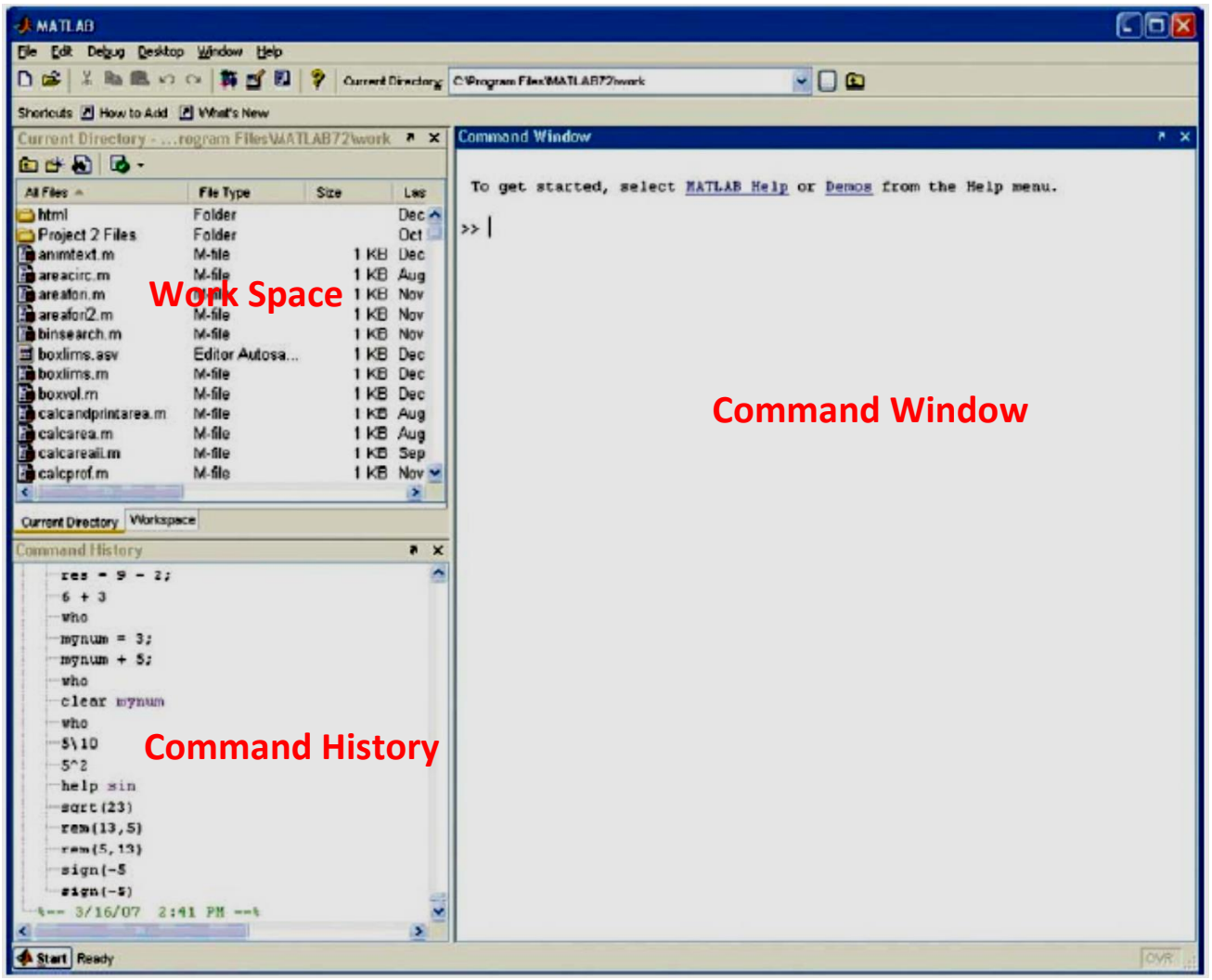
MATLAB System Environment



Arithmetic Operations in MATLAB

1. **+ :** is used for **addition** of two numbers or two matrices.
2. **- :** is used for **subtraction** of two numbers or two matrices.
3. *** :** is used for **multiplication** of two numbers or two matrices.
4. **/ or \ :** is used for **division** of two numbers or two matrices. Ex. **70/7=7, 70\7=0.1. (why?)**
5. **^ :** is used to raise a value to a certain exponent. Ex.: **10^2=100.**
6. **sqrt :** is used for finding the square root for any number. EX.: **sqrt(100)=10.**
7. **% :** for writing comments on command window or in m-file.
8. **.* :** for element by element matrix multiplication.
9. **./ or .\ :** for element by element matrix division.
10. **.^ :** for raising the elements of a matrix to a certain exponent.

11. … : used for continuing writing the expression located in the current line in the next line.

Some predefined variables in MATLAB

Predefined Variable	Stands For
pi	$\pi = 3.1416$
Inf	$\infty \equiv$ Infinity
NaN	Not a Number
i	The complex variable $\sqrt{-1}$
j	The complex variable $\sqrt{-1}$

Examples:

```
>> pi

ans =

   3.1416

>> k=pi

k =

   3.1416

>> d=i

d =

      0 + 1.0000i

>> a=j

a =

      0 + 1.0000i

>> s=0/0

s =

  NaN

>> b=90/0

b =

  Inf
```

We noticed that all the names of the variables exist in the ***work space*** window and all the commands previously implemented are found in the ***command history*** window. The command ***who*** gives the names of variables and the command ***whos*** gives the names of variables, their types and their storage space

```
>> who

Your variables are:

a   ans  b   c   d   k   s

>> whos
```

Name	Size	Bytes	Class	Attributes
a	1x1	16	double	complex
ans	1x1	8	double	
b	1x1	8	double	
c	1x1	8	double	
d	1x1	16	double	complex
k	1x1	8	double	
s	1x1	8	double	

The command ***clear*** is used to delete variables from the computer storage and from the work space.

```
>> clear b c

>> clear

>> who

>>
```

Notice that there are no results when who command is used after clear command.

Arithmetic Expression

The arithmetic expression consists of set of constants and variables related by arithmetic operations. The arithmetic symbols like +, -, /, *, ^ etc. Some examples of arithmetic expressions are indicated below:

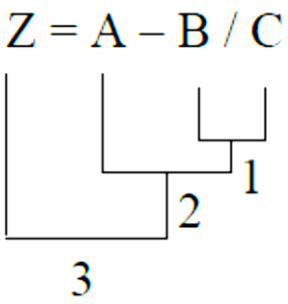
Expression in MATLAB	Arithmetic Expression
a-3*b	a-3b
c^2-10	$c^2 - 10$
(a^2+b^2)/12	$\frac{a^2 + b^2}{12}$
m*(7*d-3*t)	m(7d-3t)
1/(2+3^2)+4/5*6/7	$\frac{1}{2 + 3^2} + \frac{4}{5} \times \frac{6}{7}$

Rules of Precedence

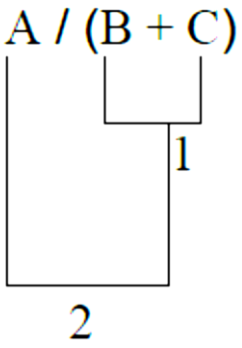
The following table summarizes the precedence rules for implementing arithmetic expressions

Precedence	Arithmetic Operation
1. Brackets	The expression between the brackets has the higher precedence than other expressions where all brackets are implemented firstly from the inner to the outer brackets.
2. Power	After brackets all exponents are implemented.
3. Multiplication and division	All multiplications and divisions are implemented from left to right.
4. Addition and subtraction	All additions and subtractions are implemented from left to right.
5. Assignment operation	The last step.

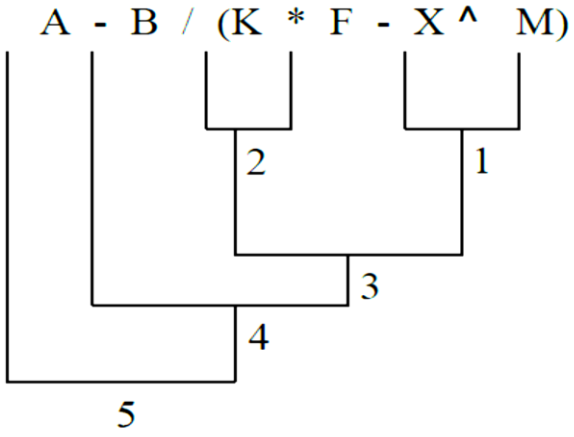
Example: if we have the expression $\boldsymbol{Z = A - \frac{B}{C}}$ the sequence of operations will be as follows:



Example: while the sequence of operations for the expression $\frac{A}{B+C}$ would be as follows



Example:



H.W.: for each of the following, convert the algebraic expressions to the MATLAB expressions formats and determine the precedence map:

1. $\boldsymbol{Z = aX + \frac{bXY}{Y^2} - cX^3}$
2. $\boldsymbol{F = \frac{-b+\sqrt{b^2-4ac}}{2a}}$
2. $\boldsymbol{T = \frac{\pi}{2}[4F_1 - 3F_2 + \frac{6F_1F_2}{F_1-F_2}]}$
4. $\boldsymbol{X = \frac{X_1Y_2-X_2Y_1}{Y_2-Y_1}}$

Numerical Systems

If x is a real positive number where:

$$x = (a_n a_{n-1} \dots a_0 . b_1 b_2 \dots)_{10}$$

Then the value of x is the summation of the two sequences:

$$\sum_{k=0}^{n} a_k \times 10^k + \sum_{k=1}^{\infty} b_k 10^{-k}$$

If is a real positive binary number:

$$x = (a_n a_{n-1} \dots a_0 . b_1 b_2 \dots)_2$$

Then the value of x is the summation of the two sequences:

$$\sum_{k=0}^{n} a_k \times 2^k + \sum_{k=1}^{\infty} b_k 2^{-k}$$

If is a real positive binary number:

$$x = (a_n a_{n-1} \dots a_0 . b_1 b_2 \dots)_{16}$$

Then the value of x is the summation of the two sequences:

$$\sum_{k=0}^{n} a_k \times 16^k + \sum_{k=1}^{\infty} b_k 16^{-k}$$

Example: convert the decimal number (39.125) to binary hexadecimal and octal.

Solution:

We firstly convert the integer part of the real number as follows:

number	Base	Residual
39	2	1
19	2	1
9	2	1
4	2	0
2	2	0
1	2	1
0		

First place in the number

0.125 x 2 = 0.250 0

0.25 x 2 = 0.5 0

0.5 x 2 = 1.0 1

$(39.125)_{10}=(111001.001)_2$

العدد	الأساس	
39	8	7 ← First place in the number
4	8	4
0		

0.125 x 8 = 1.0 1

$(39.125)_{10}=(47.1)_8$

Example: convert the binary number $(111.001)_2$ to decimal.

$$(111.001)_2 = 1\times2^0 + 1\times2^1 + 1\times2^2 + 0\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$$

$= 1 + 2 + 4 + 0 + 0 + 0.125$

$= (7.125)_{10}$

Example: convert the octal number $(212.726)_8$ to decimal.

$$(212.726)_8 = 2\times8^0 + 1\times8^1 + 2\times8^2 + 7\times8^{-1} + 2\times8^{-2} + 6\times8^{-3}$$
$$= 2 + 1 + 128 + 0.875 + 0.03125 + 0.01171875 = (131.91795)_{10}$$

Numerical Systems in MATLAB

1. **dec2bin**: convert the decimal number d to binary. Either of the following formats can be used:

   str = dec2bin(d)

   str = dec2bin(d,n)

where:

str: contains the result of conversion, d: is the number to be converted and n: is the number of binary places on the resulted number.

>> dec2bin(100)

ans =

1100100

>> dec2bin(100,4)

ans =

1100100

>> dec2bin(100,8)

ans =

01100100

>> dec2bin(100,10)

ans =

0001100100

2. **dec2hex**: converts the decimal number to hexadecimal number. One of the following formats can be used:

   str=dec2hex(d)
   str=dec2hex(d, n)

>> dec2hex(1023)

ans =

3FF

>> dec2hex(1023,6)

ans =

0003FF

3. **dec2base**: converts the decimal number d to a number to the base *'base'*.
   str = dec2base(d, base)
   str = dec2base(d, base, n)

   >> dec2base(100,2)
   ans =
   1100100

```
>> dec2base(1023,16)
ans =
3FF
>> dec2base(80,8)
ans =
120
>> dec2base(120,4)
ans =
1320
>> dec2base(120,4,7)
ans =
0001320
```

4. **bin2dec**: converts the binary number to decimal.

   bin2dec(*binarystr*)

5. **hex2dec:** converts the hexadecimal number to decimal.

   d = hex2dec('*hex_value*')

6. **base2dec:** converts the number to the base *'base'* to decimal.

   d = base2dec('*strn*', base)

H.W.: convert each of the following decimal numbers to binary, octal and hexadecimal

(500) .5 (100) .4 (101.776) .3 (22.892) .2 (10.675) .1

(1000) .7 (750) .6

H.W.: convert each of the following numbers to decimal:

$(765.67)_8$ .3 $(6E.FA)_{16}$ .2 $(111101.011)_2$ .1

$(4571)_8$ .6 $(12B)_{16}$ .5 $(111000001)_2$ .4