Chapter Four
Mathematical Operations

4.1 Priority of mathematical operations

Matlab performs mathematical operations in the following order

a) Any operation between parentheses.

b) Raising to a power (exponent).

c) Multiplication or division whichever is first from left to right.

d) Addition or subtraction whichever is first from left to right.

Note: the number of parentheses must always be even because they must be used as pairs.

Ex. (4.1): Write Matlab code to evaluate the following equation for any value of x:

$$y = \frac{x^2 - 1}{x^2 + 1} + \frac{x^3 - x + 2}{\sqrt{x^2 - 1}} - x^2 + x + 1$$

Sol:

```
clear , clc
x = input( ' Enter the value of x = ' ) ;
A = ( x ^ 2 – 1 ) / ( x ^ 2 + 1 ) ;
B = ( x ^ 3 – x + 2 ) / sqrt( x ^ 2 – 1 ) ;
C = – x ^ 2 + x + 1 ;
y = A + B + C ;
fprintf( ' \t  %6.3f  \n  ' , [ x , y ] )
```

Run:

```
Enter the value of x = 2
      2.000
      4.219
```

4.2 Mathematical operations between matrices

When a mathematical operation is carried out between two matrices the number of their elements must be identical. If the two matrices are two dimensional the number of rows and columns of the first matrix must equal their counterparts in the seconds. However, any operation can be performed between a single number (scalar) and any matrix whether it is one or two dimensional. To ensure element by element operation between identical matrices the dot operator (.) must be added before (*) (/) (^) signs:

Ex. (4.2): Write Matlab program to display a matrix of five rows and four columns where the rows represent the output values of the addition, subtraction, product, division and raising to power. All of the five operations are carried out between a matrix containing four identical numbers (2 for instance) and a matrix having four arbitrary numbers.

```
clear , clc
 x = 2 * ones( 1 , 4 ) ;
 y = [ 3 , 5 , 6 , 8 ] ;
 z = x + y ;
w = y – x ;
 a = x . * y ;
 b = x . / y ;
 c = y . ^ x ;
 R = [ z ; w ; a ; b ; c ] ;
 disp( R )
```

Run:

```
   5.0000    7.0000    8.0000   10.0000
   1.0000    3.0000    4.0000    6.0000
   6.0000   10.0000   12.0000   16.0000
   0.6667    0.4000    0.3333    0.2500
   9.0000   25.0000   36.0000   64.0000
```