

## Chapter 2

# MATRICES AND OPERATORS

### PREFACE

1. Introduction to Matrices and Operators
2. Accessing parts of a Matrix
3. Combining matrices
4. Operator Precedence

### 1. Introduction to Matrices and Operators

- **MATRICES:** Special kinds of an **ARRAY**.
- An **ARRAY** is a set of numbers arranged in a rectangular pattern; a collection of similar elements in simple terms.
- 1D array --> Vector, 2D array and further dimensions --> Matrix
- Ex: `>> x_1D = [1 2 3]`  
`x_1D = 1    2    3`  
`>> x_2D = [1 2 3; 4 5 6]`  
`x_2D = 1    2    3`  
`4    5    6`
- There are many inbuilt functions in MATLAB that help us find the properties of a matrix.
- Ex: `size(array_name)` --> prints the size of the matrix i.e, 2 2 if it is a 2\*2 matrix
- The hierarchy of variable formats is mentioned below from the smallest to largest:
  - Scalar**     --> A single element.
  - Vector**    --> Collection of elements.
  - Matrices** --> Row-Column representation of values; Matrix format.
  - Arrays**    --> Extends to multiple dimensions; 1D, 2D, 3D and so on..
- **COLON** operator helps us to store the values in incremental fashion to store it in an array.
- Ex: `>> x = 1:2:3 --> x = 1 3 ;`
- Start at 1, increase at steps of 2 and go no higher than 3 (go no below can be implemented using the minus operator for the second operand).
- `Vx = 1:100` stores first 100 numbers in the variable **Vx**.
- Other operators include: +, -, \*, / and so on.

### 2. Accessing parts of a Matrix

- To access an element in any matrix, specify the row and column of the element in parentheses next to the array name in the prompt window.
- Ex: `A(3,2)` --> accesses the element present in the 3rd row and 4th column of the array.
- You can also access multiple elements by following the specified format:  
`A(1,4) + A(2,4) + A(3,4) + A(4,4)`
- There are many other ways of accessing the elements or parts of an array. Few of the functions are : **end**, **logical**, **linear** and furthermore.

### 3. Combining Matrices

- Matrix combination row wise:
- A1, A2, A3 are the three 2\*2 matrices; [A1 A2 A3] will concatenate the three matrices in a row.

```
>> A1=[1 2 3;4 5 6]
```

```
A1 =
```

```
    1    2    3
    4    5    6
```

```
>> A2=[1 3 5;3 5 7]
```

```
A2 =
```

```
    1    3    5
    3    5    7
```

```
>> A3=[1 2 4;3 5 6]
```

```
A3 =
```

```
    1    2    4
    3    5    6
```

```
>> [A1 A2 A3]
```

```
ans =
```

```
Columns 1 through 7
```

```
    1    2    3    1    3    5    1
    4    5    6    3    5    7    3
```

```
Columns 8 through 9
```

```
    2    4
    5    6
```

- We can combine matrices column wise by adding a **semicolon(;** between the array names.

### 4. Operator Precedence

- The precedence rules for MATLAB operators are shown in this list, ordered from **highest** precedence level to **lowest** precedence level:
  1. Parentheses **()**
  2. Transpose **(.)'**, power **(.^)**, complex conjugate transpose **(')'**, matrix power **(^)**
  3. Unary plus **(+)**, unary minus **(-)**, logical negation **(~)**
  4. Multiplication **(.\*)**, right division **(./)**, left division **(.\)**, matrix multiplication **(\*)**, matrix right division **(/)**, matrix left division **(\)**
  5. Addition **(+)**, subtraction **(-)**
  6. Colon operator **(:)**
  7. Less than **(<)**, less than or equal to **(<=)**, greater than **(>)**, greater than or equal to **(>=)**, equal to **(==)**, not equal to **(~=)**
  8. Element-wise AND **(&)**
  9. Element-wise OR **(|)**
  10. Short-circuit AND **(&&)**
  11. Short-circuit OR **(||)**