Lecture Three

✓ Array operations

- Matrix arithmetic operations
- Array arithmetic operations
- Solving linear equations

Array operations

MATLAB has two different types of arithmetic operations: matrix arithmetic operations and array arithmetic operations.

- Matrix arithmetic operations

MATLAB allows arithmetic operations: +, -, *, and ^ to be carried out on matrices. Thus,

- A+B or B+A is valid if A and B are of the same size
- A*B is valid if A's number of column equals B's number of rows
- A^2 is valid if A is square and equals A*A
- α*A or A*α multiplies each element of A by α

Adding matrices

Add two matrices together is just the addition of each of their respective elements. If A and B are both matrices of the same dimensions (size),

then C = A + B produces C, where the i^{th} row and j^{th} column are just the addition of the elements (numbers) in the i^{th} row and j^{th} column of A and B.

Let`s say that :

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{bmatrix}, \text{ and } B = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{bmatrix}$$

so that the addition is :

$$C = A + B = \begin{bmatrix} 3 & 7 & 11 \\ 15 & 19 & 23 \end{bmatrix}$$

The MATLAB commands to perform these matrix assignments and the addition are:

A = [1 3 5 ; 7 9 11] B = [2 4 6 ; 8 10 12] C = A + B

Rule: A, B, and C must all have the same dimensions

Multiplication

Matrix multiplication, also known as matrix product and the multiplication of two matrices produces a single matrix. It is a type of binary operation. It is not an element-by-element multiplication. Rather, matrix multiplication is the result of the dot products of rows in one matrix with columns in another. Consider:

$$C = A * B$$

matrix multiplication gives the i^{th} row and k^{th} column spot in C as the scalar results of the dot product of the i^{th} row in A with the k^{th} column in B.

Example

Step 1: Dot Product (a 1-row matrix times a 1-column matrix) The Dot product is the scalar result of multiplying one row by one column

$$\underset{1x3}{\begin{bmatrix} 2 & 5 & 3 \end{bmatrix}} * \begin{bmatrix} 6 \\ 8 \\ 7 \end{bmatrix}_{3x1} = 2*6 + 5*8 + 3*7 = 73_{1x1} \qquad \text{DOT PRODUCT OF ROW AND COLUMN}$$

Rule:

1) # of elements in the row and column must be the same
2) 2) must be a row times a column, not a column times a row

Step 2: general matrix multiplication is taking a series of dot products each row in pre-matrix by each column in post-matrix.

$$\underset{2x3}{\begin{bmatrix} 1 & 4 & 2 \\ 9 & 3 & 7 \end{bmatrix}} * \begin{bmatrix} 5 & 6 \\ 8 & 12 \\ 10 & 11 \end{bmatrix}_{3x2} = \begin{bmatrix} 1*5+4*8+2*10 & 1+6*4*12+2*11 \\ 9*5+3*8+7*10 & 9*6+3*12+7*11 \end{bmatrix} = \begin{bmatrix} 57 & 76 \\ 139 & 167 \end{bmatrix}_{2x2}$$

- Array arithmetic operations

On the other hand, array arithmetic operations or array operations for short, are done element-by-element. The period character, . , distinguishes the array operations from the matrix operations. However, since the matrix and array operations are the same for addition (+) and subtraction (-), the character pairs (.+) and (.-) are not used. The list of array operators is shown below in Table.

Table 1: Array operators

.*	Element-by-element multiplication
./	Element-by-element division
.^	Element-by-element exponentiation

If A and B are two matrices of the same size with elements A = [a_{ij}] and B = [b_{ij}], then the command produces another matrix C of the same size with elements $c_{ij} = a_{ij}b_{ij}$. For example, using the same 3 x 3 matrices,

```
>> C = A.*B
```

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

we have,

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

To raise a scalar to a power, we use for example the command 10^2. If we want the operation to be applied to each element of a matrix, we use .^2. For example, if we want to produce a new matrix whose elements are the square of the elements of the matrix A, we enter

```
>> A.^2
      ans =
            1     4     9
            16    25    36
            49    64    81
```

The relations below summarize the above operations. To simplify, let's consider two vectors U and V with elements U = [u_i] and V = [v_j].

- U. ∗ V produces $[u_1v_1\ u_2v_2 \ldots u_nv_n]$
- U./V produces $[u_1/v_1\ u_2/v_2 \ldots u_n/v_n]$
- U.ˆV produces $[u1^{v1}\ u2^{v2} \ldots un^{vn}]$

Table 2: Summary of matrix and array operations

Operation	Matrix	Array
Addition	+	+
Subtraction	−	−
Multiplication	∗	.∗
Division	/	./
Left division	\	.\
Exponentiation	ˆ	.ˆ

Solving linear equations

One of the problems encountered most frequently in scientific computation is the solution of systems of simultaneous linear equations. With matrix notation, a system of simultaneous linear equations is written

$$Ax = b$$

where there are as many equations as unknown. A is a given square matrix of order n, b is a given column vector of n components, and x is an unknown column vector of n components. In linear algebra we learn that the solution to Ax = b can be written as x = A−1 b, where A−1 is the inverse of A. For example, consider the following system of linear equations

$$\mathbf{x + 2y + 3z = 1}$$
$$\mathbf{4x + 5y + 6z = 1}$$
$$\mathbf{7x + 8y = 1}$$

The coefficient matrix A is

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

There are typically two ways to solve for x in MATLAB:

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
```

2. The second one is to use the backslash (\ ) operator. The numerical algorithm behind this operator is computationally efficient. This is a numerically reliable way of solving system of linear equations by using a well-known process of Gaussian elimination.

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

Ex. *Solving a set of linear equations*

$-6x = 2y - 2z + 15$
$4y - 3z = 3x + 13$
$2x + 4y - 7z = -9$

First, rearrange the equations

$-6x - 2y + 2z = 15$

$-3x + 4y - 3z = 13$

$2x + 4y - 7z = -9$

Second, write the equations in a matrix form **Ax = b**

The coefficient matrix is

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

The constant column vector is

$$b = \begin{bmatrix} 15 \\ 13 \\ -9 \end{bmatrix}$$

Third, solve the simultaneous equations in Matlab

```
>> x = A\b
```

The Matlab answer is:

```
x =   -2.7273
       2.7727
       2.0909
```