```
>> A=[2 1 4; 4 1 8; 2 -1 3]
A =
     2     1     4
     4     1     8
     2    -1     3
>> B=inv(A)
B =
    5.5000   -3.5000    2.0000
    2.0000   -1.0000         0
   -3.0000    2.0000   -1.0000
>> A*B
ans =
     1     0     0
     0     1     0
     0     0     1
```

Creating the matrix A.

Use the inv function to find the inverse of A and assign it to B.

Multiplication of A and B gives the identity matrix.

MATLAB has two types of array division, right division and left division.

Left division, \ :

Left division is used to solve the matrix equation AX=B . In this equation X and B are column vectors. This equation can be solved by multiplying, on the left, both sides by the inverse of A:

the solution of $AX = B$ is:

$$X = A^{-1}B$$

In MATLAB the last equation can be written by using the **left division** character:

X = A\B

Right division, / :

The right division is used to solve the matrix equation XR=D . In this equation X and D are row vectors. This equation can be solved by multiplying, on the right, both sides by the inverse of R:

XR=D

$X=D.R^{-1}$

In MATLAB the last equation can be written using the right division character:

X=D/R

The following example demonstrates the use of the left and right division, and the inv function to solve a set of linear equations.

The other concept is using in the matrix operation is ***Determinants*** ,the determinant is a function that associates with each square matrix A a number, called the determinant of the matrix. The determinant is typically denoted by **det(A)** or |A|. The determinant is calculated according to specific rules. For a second-order matrix the rule is:

$$|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}, \text{ for example, } \begin{vmatrix} 6 & 5 \\ 3 & 9 \end{vmatrix} = 6 \cdot 9 - 5 \cdot 3 = 39$$

Sample Problem: Solving three linear equations (array division) Use matrix operations to solve the following system of linear equations.

$$\begin{aligned} 4x - 2y + 6z &= 8 \\ 2x + 8y + 2z &= 4 \\ 6x + 10y + 3z &= 0 \end{aligned}$$

Solution:

Using the rules of linear algebra demonstrated earlier, the above system of equations can be written in the matrix form AX=B or **XR=D** in the form:

$$\begin{bmatrix} 4 & -2 & 6 \\ 2 & 8 & 2 \\ 6 & 10 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} 4 & 2 & 6 \\ -2 & 8 & 10 \\ 6 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 8 & 4 & 0 \end{bmatrix}$$

Solutions for both forms are shown below:

```
>> A=[4 -2 6; 2 8 2; 6 10 3];          Solving the form AX = B.
>> B=[8; 4; 0];
>> X=A\B                               Solving by using left division: X = A \ B.
X =
   -1.8049
    0.2927
    2.6341
>> Xb=inv(A)*B                         Solving by using the inverse of A: X = A^-1 B.
Xb =
   -1.8049
    0.2927
    2.6341
```

```
>> C=[4 2 6; -2 8 10; 6 2 3];
>> D=[8 4 0];
>> Xc=D/C
Xc =
   -1.8049    0.2927    2.6341
>> Xd=D*inv(C)
Xd =
   -1.8049    0.2927    2.6341
```

Solving the form $XR = D$.

Solving by using right division: $X = D/R$

Solving by using the inverse of C: $X = D \cdot C^{-1}$.

2.5 Strings and Strings as Variables

•A string is an array of characters. It is created by typing the characters within single quotes.
• Strings can include letters, digits, other symbols, and spaces.
• Examples of strings: 'ad ef ', '3%fr2', '{edcba:21!', 'MATLAB'.
• A string that contains a single quote is created by typing two single quotes within the string.
• When a string is being typed in, the color of the text on the screen changes to maroon when the first single quote is typed. When the single quote at the end of the string is typed, the color of the string changes to purple.

Strings can also be assigned to variables by simply typing the string on the right side of the assignment operator, as shown in the examples below:

```
>> a='FRty 8'
a =
FRty 8
>> B='My name is John Smith'
B =
My name is John Smith
>>
```

When a variable is defined as a string, the characters of the string are stored in an array just as numbers are. Each character, including a space, is an element in the array. This means that a one-line string is a row vector in which the number of elements is equal to the number of characters. The elements of the vectors are addressed by position. For example, in the vector B that was defined above the 4th Element is the letter n, the 12th element is J, and so on.

```
>> B(4)
ans =
n
>> B(12)
ans =
J
```

As with a vector that contains numbers, it is also possible to change specific elements by addressing them directly. For example, in the vector B above the name John can be changed to Bill by:

```
>> B(12:15)='Bill'
B =
My name is Bill Smith
>>
```

Using a colon to assign new characters to elements 12 through 15 in the vector B.

MATLAB has a built-in function named char that creates an array with rows having the same number of characters from an input of rows not all of the same length. MATLAB makes the length of all the rows equal to that of the longest row by adding spaces at the end of the short lines. In the char function, the rows are entered as strings separated by a comma according to the following format:

```
variable_name = char('string 1','string 2','string 3')
```

For example:

```
>> Info=char('Student Name:','John Smith','Grade:','A+')
Info =
Student Name:
John Smith
Grade:
A+
>>
```

A variable named Info is assigned four rows of strings, each with different length.

The function char creates an array with four rows with the same length as the longest row by adding empty spaces to the shorter lines.

2.6 Built-in Functions for Analyzing Arrays

MATLAB has many built-in functions for analyzing arrays. Table below lists some of these functions.

Function	Description	Example
mean(A)	If A is a vector, returns the mean value of the elements of the vector.	>> A=[5 9 2 4]; >> mean(A) ans = 5
C=max(A)	If A is a vector, C is the largest element in A. If A is a matrix, C is a row vector containing the largest element of each column of A.	>> A=[5 9 2 4 11 6 11 1]; >> C=max(A) C = 11
[d,n]=max(A)	If A is a vector, d is the largest element in A, and n is the position of the element (the first if several have the max value).	>> [d,n]=max(A) d = 11 n = 5