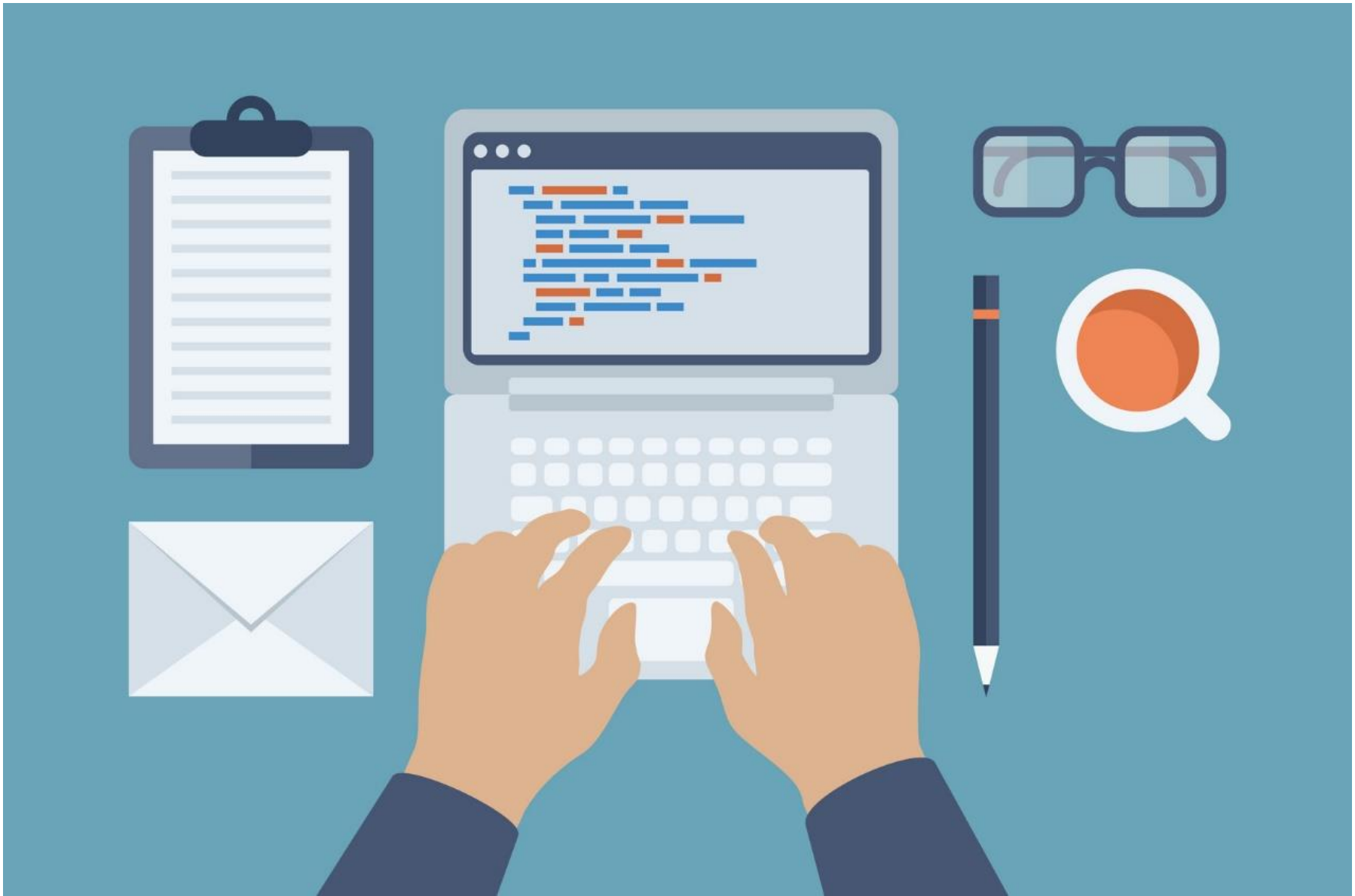
Computer Programming



INTRODUCTION

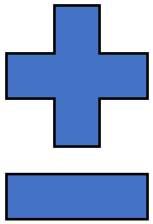
MATLAB ARRAYS:

uses to store and manipulate data

This Chapter Presents The Basic, Most Common Mathematical Operations That MATLAB Performs Using Arrays.

knowledge of ***MATRIX OPERATIONS*** *and* ***LINEAR ALGEBRA***

ADDITION AND SUBTRACTION

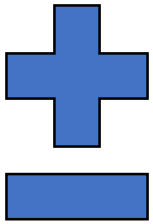


$$\mathbf{A} = \begin{bmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} & \mathbf{A_{13}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} & \mathbf{A_{23}} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B_{11}} & \mathbf{B_{12}} & \mathbf{B_{13}} \\ \mathbf{B_{21}} & \mathbf{B_{22}} & \mathbf{B_{23}} \end{bmatrix}$$

$$\mathbf{C} = \mathbf{A} \pm \boldsymbol{B} = \begin{bmatrix} (\mathbf{A_{11}} \pm \mathbf{B_{11}}) & (\mathbf{A_{12}} \pm \mathbf{B_{12}}) & (\mathbf{A_{13}} \pm \mathbf{B_{13}}) \\ (\mathbf{A_{21}} \pm \mathbf{B_{21}}) & (\mathbf{A_{22}} \pm \mathbf{B_{22}}) & (\mathbf{A_{23}} \pm \mathbf{B_{23}}) \end{bmatrix}$$

the same numbers of rows and columns



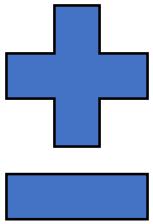
ADDITION AND SUBTRACTION

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} & \mathbf{A_{13}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} & \mathbf{A_{23}} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B_{11}} & \mathbf{B_{12}} & \mathbf{B_{13}} \end{bmatrix}$$

$$\mathbf{C} = \mathbf{A} \pm \mathbf{B}$$

??? Error using ==> plus
Matrix dimensions must agree.



ADDITION AND SUBTRACTION

```
>> VectA=[8 5 4]; VectB=[10 2 7];
>> VectC=VectA+VectB
VectC =
    18     7    11
>> A=[5 -3 8; 9 2 10]
A =
     5    -3     8
     9     2    10
>> B=[10 7 4; -11 15 1]
B =
    10     7     4
   -11    15     1
>> A-B
ans =
    -5   -10     4
    20   -13     9
>> C=A+B
C =
    15     4    12
    -2    17    11
>> VectA+A
??? Error using ==> plus
Matrix dimensions must agree.

>>
```

Define two vectors.

Define a vector `VectC` that is equal to `VectA+VectB`.

Define two 2×3 matrices `A` and `B`.

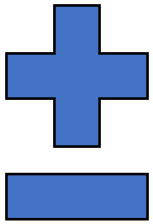
Subtracting matrix `B` from matrix `A`.

Define a matrix `C` that is equal to `A+B`.

Trying to add arrays of different size.

An error message is displayed.

ADDITION AND SUBTRACTION



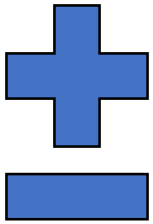
scalar (number)

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} & \mathbf{A_{13}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} & \mathbf{A_{23}} \end{bmatrix}$$

$$\mathbf{B} = \boldsymbol{n}$$

$$\mathbf{C} = \mathbf{A} \pm \mathbf{B} = \begin{bmatrix} (\mathbf{A_{11}} \pm \boldsymbol{n}) & (\mathbf{A_{12}} \pm \boldsymbol{n}) & (\mathbf{A_{13}} \pm \boldsymbol{n}) \\ (\mathbf{A_{21}} \pm \boldsymbol{n}) & (\mathbf{A_{22}} \pm \boldsymbol{n}) & (\mathbf{A_{23}} \pm \boldsymbol{n}) \end{bmatrix}$$

ADDITION AND SUBTRACTION



scalar (number)

```
>> VectA=[1 5 8 -10 2]
VectA =
     1     5     8   -10     2

>> VectA+4
ans =
     5     9    12    -6

>> A=[6 21 -15; 0 -4 8]
A =
     6    21   -15
     0    -4     8

>> A-5
ans =
     1    16   -20
    -5    -9     3
```

Define a vector named `VectA`.

Add the scalar 4 to `VectA`.

4 is added to each element of `VectA`.

Define a 2×3 matrix `A`.

Subtract the scalar 5 from `A`.

5 is subtracted from each element of `A`.

ARRAY MULTIPLICATION

multiplication

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$A_{[4\times3]} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix} \qquad B_{[3\times2]} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{bmatrix}$$

$$C_{[4\times2]} = A_{[4\times3]} \times B_{[3\times2]}$$

$$C_{[4\times2]} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31} & A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32} \\ A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31} & A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32} \\ A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31} & A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32} \\ A_{41}B_{11} + A_{42}B_{21} + A_{43}B_{31} & A_{41}B_{12} + A_{42}B_{22} + A_{43}B_{32} \end{bmatrix}$$

ARRAY MULTIPLICATION

multiplication

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$\begin{bmatrix} 1 & 4 & 3 \\ 2 & 6 & 1 \\ 5 & 2 & 8 \end{bmatrix} \begin{bmatrix} 5 & 4 \\ 1 & 3 \\ 2 & 6 \end{bmatrix} = \begin{bmatrix} (1\cdot 5+4\cdot 1+3\cdot 2) & (1\cdot 4+4\cdot 3+3\cdot 6) \\ (2\cdot 5+6\cdot 1+1\cdot 2) & (2\cdot 4+6\cdot 3+1\cdot 6) \\ (5\cdot 5+2\cdot 1+8\cdot 2) & (5\cdot 4+2\cdot 3+8\cdot 6) \end{bmatrix} = \begin{bmatrix} 15 & 34 \\ 18 & 32 \\ 43 & 74 \end{bmatrix}$$

ARRAY MULTIPLICATION

multiplication *square matrices*

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$A_{[n\times n]} = \begin{bmatrix} A_{11} & \dots & A_{1n} \\ \dots & \dots & \dots \\ A_{n1} & \dots & A_{nn} \end{bmatrix}$$

$$B_{[n\times n]} = \begin{bmatrix} B_{11} & \dots & B_{1n} \\ \dots & \dots & \dots \\ B_{n1} & \dots & B_{nn} \end{bmatrix}$$

$$\mathbf{A \times B \neq B \times A}$$

ARRAY MULTIPLICATION

multiplication vector

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$A_{[1\times n]} = [A_{11} \quad \dots \quad A_{1n}] \qquad B_{[n\times 1]} = \begin{bmatrix} B_{11} \\ \dots \\ B_{n1} \end{bmatrix}$$

$$C_{[1\times 1]} = \mathrm{A} \times \mathrm{B} = A_{11}B_{11} + \cdots + A_{1n}B_{n1}$$

$$C_{[n\times n]} = \mathrm{B} \times \mathrm{A} = \begin{bmatrix} B_{11}A_{11} & \dots & B_{11}A_{1n} \\ \dots & \dots & \dots \\ B_{n1}A_{11} & \dots & B_{n1}A_{1n} \end{bmatrix}$$

ARRAY MULTIPLICATION

multiplication *vector*

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$A_{[1\times n]} = [A_{11} \quad \dots \quad A_{1n}] \qquad B_{[n\times 1]} = \begin{bmatrix} B_{11} \\ \dots \\ B_{n1} \end{bmatrix}$$

$$C_{[1\times 1]} = \mathrm{A} \times \mathrm{B} = A_{11}B_{11} + \cdots + A_{1n}B_{n1}$$

$$C_{[n\times n]} = \mathrm{B} \times \mathrm{A} = \begin{bmatrix} B_{11}A_{11} & \dots & B_{11}A_{1n} \\ \dots & \dots & \dots \\ B_{n1}A_{11} & \dots & B_{n1}A_{1n} \end{bmatrix}$$

ARRAY MULTIPLICATION

multiplication *vector* *dot*

ACCORDING TO THE RULES OF LINEAR ALGEBRA

dot

R2018b

Dot product

collapse all in page

Syntax

```
C = dot(A,B)
C = dot(A,B,dim)
```

Description

example

`C = dot(A,B)` returns the scalar dot product of A and B.

- If A and B are vectors, then they must have the same length.
- If A and B are matrices or multidimensional arrays, then they must have the same size. In this case, the `dot` function treats A and B as collections of vectors. The function calculates the dot product of corresponding vectors along the first array dimension whose size does not equal 1.

example

`C = dot(A,B,dim)` evaluates the dot product of A and B along dimension, `dim`. The `dim` input is a positive integer scalar.

ARRAY MULTIPLICATION

multiplication vector dot

ACCORDING TO THE RULES OF LINEAR ALGEBRA

```
A = [4 -1 2];
B = [2 -2 -1];
```

Calculate the dot product of A and B.

```
C = dot(A,B)
```

```
C = 8
```

The result is 8 since

```
C = A(1)*B(1) + A(2)*B(2) + A(3)*B(3)
```

ARRAY MULTIPLICATION

multiplication vector dot

ACCORDING TO THE RULES OF LINEAR ALGEBRA

```
A = [1 2 3;4 5 6;7 8 9];
B = [9 8 7;6 5 4;3 2 1];
```

Find the dot product of A and B.

```
C = dot(A,B)
```

```
C = 1×3

    54    57    54
```

ARRAY MULTIPLICATION

multiplication *vector* *dot*

ACCORDING TO THE RULES OF LINEAR ALGEBRA

The result, `C`, contains three separate dot products. `dot` treats the columns of `A` and `B` as vectors and calculates the dot product of corresponding columns. So, for example, `C(1) = 54` is the dot product of `A(:,1)` with `B(:,1)`.

ARRAY MULTIPLICATION

multiplication vector dot

ACCORDING TO THE RULES OF LINEAR ALGEBRA

Find the dot product of A and B, treating the *rows* as vectors.

```
D = dot(A,B,2)
```

```
D = 3×1

    46
    73
    46
```

In this case, `D(1) = 46` is the dot product of `A(1,:)` with `B(1,:)`.

ARRAY MULTIPLICATION


```
>> A=[1 4 2; 5 7 3; 9 1 6; 4 2 8]
A =
     1     4     2
     5     7     3
     9     1     6
     4     2     8
>> B=[6 1; 2 5; 7 3]
B =
     6     1
     2     5
     7     3
>> C=A*B
C =
    28    27
    65    49
    98    32
    84    38
>> D=B*A
??? Error using ==> *
Inner matrix dimensions must agree.

>> F=[1 3; 5 7]
F =
     1     3
     5     7
>> G=[4 2; 1 6]
```

Define a 4×3 matrix A.

Define a 3×2 matrix B.

Multiply matrix A by matrix B and assign the result to variable C.

Trying to multiply B by A, B*A, gives an error since the number of columns in B is 2 and the number of rows in A is 4.

Define two 2×2 matrices F and G.

ARRAY MULTIPLICATION

multiplication

```
G =
     4     2
     1     6
>> F*G
ans =
     7    20
    27    52
>> G*F
ans =
    14    26
    31    45
>> AV=[2 5 1]
AV =
     2     5     1
>> BV=[3; 1; 4]
BV =
     3
     1
     4
>> AV*BV
ans =
    15
>> BV*AV
ans =
     6    15     3
     2     5     1
     8    20     4
>>
```

Multiply F*G

Multiply G*F

Note that the answer for G*F is not the same as the answer for F*G.

Define a three-element row vector AV.

Define a three-element column vector BV.

Multiply AV by BV. The answer is a scalar. (Dot product of two vectors.)

Multiply BV by AV. The answer is a 3×3 matrix.

ARRAY MULTIPLICATION

multiplication *number*

```
>> A=[2 5 7 0; 10 1 3 4; 6 2 11 5]
A =
     2     5     7     0
    10     1     3     4
     6     2    11     5
```

Define a 3×4 matrix `A`.

```
>> b=3
b =
     3
```

Assign the number 3 to the variable `b`.

```
>> b*A
ans =
     6    15    21     0
    30     3     9    12
    18     6    33    15
```

Multiply the matrix `A` by `b`. This can be done by either typing `b*A` or `A*b`.

```
>> C=A*5
C =
    10    25    35     0
    50     5    15    20
    30    10    55    25
```

Multiply the matrix `A` by 5 and assign the result to a new variable `C`. (Typing `C=5*A` gives the same result.)

ARRAY MULTIPLICATION

multiplication linear equations

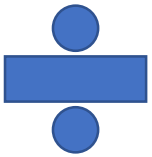
$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 = B_1$$

$$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 = B_2$$

$$A_{31}x_1 + A_{32}x_2 + A_{33}x_3 = B_3$$

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

$$AX = B \text{ where } A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \text{ and } B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$



division

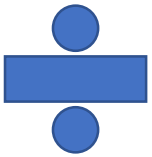
ARRAY DIVISION

ACCORDING TO THE RULES OF LINEAR ALGEBRA

Identity matrix

inverse operation

determinants



division Identity matrix

ARRAY DIVISION

ACCORDING TO THE RULES OF LINEAR ALGEBRA

eye R2018b

Identity matrix

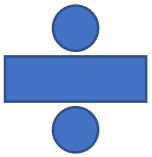
collapse all in page

Syntax

```
I = eye
I = eye(n)
I = eye(n,m)
I = eye(sz)
```

```
I = eye(4)
```

```
I = 4×4

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```



division Identity matrix

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$\begin{bmatrix}7 & 3 & 8\\ 4 & 11 & 5\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}7 & 3 & 8\\ 4 & 11 & 5\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}8\\ 2\\ 15\end{bmatrix}=\begin{bmatrix}8\\ 2\\ 15\end{bmatrix} \quad \text{or}$$

$$\begin{bmatrix}6 & 2 & 9\\ 1 & 8 & 3\\ 7 & 5 & 4\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}6 & 2 & 9\\ 1 & 8 & 3\\ 7 & 5 & 4\end{bmatrix}$$

If a matrix A is square, it can be multiplied by the identity matrix, I, from the left or from the right:

$$AI = IA = A$$

ARRAY DIVISION

division

inverse operation

ACCORDING TO THE RULES OF LINEAR ALGEBRA

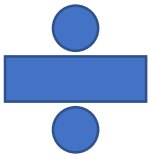
$$AB = BA = I$$

$$\begin{bmatrix} 2 & 1 & 4 \\ 4 & 1 & 8 \\ 2 & -1 & 3 \end{bmatrix} \begin{bmatrix} 5.5 & -3.5 & 2 \\ 2 & -1 & 0 \\ -3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 5.5 & -3.5 & 2 \\ 2 & -1 & 0 \\ -3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 4 \\ 4 & 1 & 8 \\ 2 & -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A\text{^} - 1$$

$$inv(A)$$

ARRAY DIVISION



division *inverse operation* *inv*

ACCORDING TO THE RULES OF LINEAR ALGEBRA

inv

R2018b

Matrix inverse

collapse all in page

Syntax

```
Y = inv(X)
```

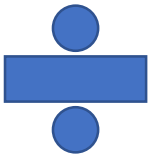
Description

example

`Y = inv(X)` computes the inverse of square matrix X.

- `X^(-1)` is equivalent to `inv(X)`.
- `x = A\b` is computed differently than `x = inv(A)*b` and is recommended for solving systems of linear equations.

ARRAY DIVISION



division | inverse operation | inv

ACCORDING TO THE RULES OF LINEAR ALGEBRA

```
X = [1 0 2; -1 5 0; 0 3 -9]
```

```
X = 3×3

     1     0     2
    -1     5     0
     0     3    -9
```

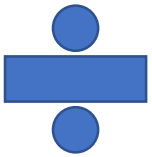
```
Y = inv(X)
```

```
Y = 3×3

    0.8824   -0.1176    0.1961
    0.1765    0.1765    0.0392
    0.0588    0.0588   -0.0980
```

```
Y*X
```

$X * X\text{^} - 1$

```
ans = 3×3

    1.0000    0.0000   -0.0000
         0    1.0000   -0.0000
         0   -0.0000    1.0000
```



division

inverse operation

ARRAY DIVISION

ACCORDING TO THE RULES OF LINEAR ALGEBRA

Tips

- It is seldom necessary to form the explicit inverse of a matrix. A frequent misuse of `inv` arises when solving the system of linear equations $Ax = b$. One way to solve the equation is with `x = inv(A)*b`. A better way, from the standpoint of both execution time and numerical accuracy, is to use the matrix backslash operator `x = A\b`. This produces the solution using Gaussian elimination, without explicitly forming the inverse. See `mldivide` for further information.

Not every matrix has an inverse. A matrix has an inverse only if it is square and its determinant is not equal to zero.

ARRAY DIVISION

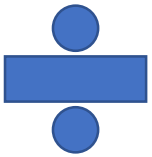
division determinants

ACCORDING TO THE RULES OF LINEAR ALGEBRA

square matrices

$det(A)$ or $|A|$

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



division determinants det

ARRAY DIVISION

ACCORDING TO THE RULES OF LINEAR ALGEBRA

det

R2018b

Matrix determinant

collapse all in page

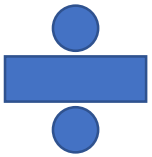
Syntax

```
d = det(A)
```

Description

example

`d = det(A)` returns the determinant of square matrix `A`.



division determinants det

ARRAY DIVISION

ACCORDING TO THE RULES OF LINEAR ALGEBRA

```
A = [1 -2 4; -5 2 0; 1 0 3]
```

```
A = 3×3

     1    -2     4
    -5     2     0
     1     0     3
```

Calculate the determinant of A.

```
d = det(A)
```

```
d = -32
```

array division *left division, *

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + A_{13}x_3 &= B_1 \\ A_{21}x_1 + A_{22}x_2 + A_{23}x_3 &= B_2 \\ A_{31}x_1 + A_{32}x_2 + A_{33}x_3 &= B_3 \end{aligned}$$

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

$$AX = B \text{ where } A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \; X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \text{ and } B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

ARRAY DIVISION

array division *left division, *

ACCORDING TO THE RULES OF LINEAR ALGEBRA

$$A^{-1}AX = IX = X$$

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

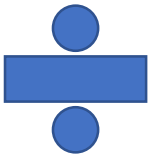
same result

$$X = A\backslash B$$

Gauss Elimination

recommended for solving a set of linear equations

calculation of the inverse may be less accurate than the Gauss Elimination method when large matrices are involved



array division *right division, /*

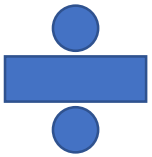
ACCORDING TO THE RULES OF LINEAR ALGEBRA

used to solve the matrix equation $XC = D$

$$XC = D$$
$$XCC^{-1} = DC^{-1}$$
$$X = DC^{-1}$$

X and D are row vectors

$$X = D/C$$



array division | sample problem 01

Solving Three Linear Equations (Array Division)

Use matrix operations to solve the following system of linear equations.

$$4x - 2y + 6z = 8$$
$$2x + 8y + 2z = 4$$
$$6x + 10y + 3z = 0$$

array division | sample problem 01

Solving Three Linear Equations (Array Division)

Solution

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

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

array division | sample problem 01

Solving Three Linear Equations (Array Division)

```
>> A=[4 -2 6; 2 8 2; 6 10 3];
>> B=[8; 4; 0];
>> X=A\B
X =
   -1.8049
    0.2927
    2.6341
>> Xb=inv(A)*B
Xb =
   -1.8049
    0.2927
    2.6341
```

Solving the form $AX = B$.

Solving by using left division: $X = A \setminus B$.

Solving by using the inverse of A: $X=A^{-1}B$.

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

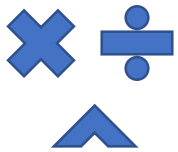
```
>> C=[4 2 6; -2 8 10; 6 2 3];
>> D=[8 4 0];
>> Xc=D/C
Xc =
   -1.8049    0.2927    2.6341
>> Xd=D*inv(C)
Xd =
   -1.8049    0.2927    2.6341
```

Solving the form $XC = D$.

Solving by using right division: $X = D/C$.

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

$$X = DC^{-1}$$
$$X = D/C$$



element-by-element

ELEMENT-BY-ELEMENT OPERATIONS

Symbol	Description	Symbol	Description
.*	Multiplication	./	Right division
.^	Exponentiation	.\	Left Division

$$a=[a_1\ a_2\ a_3\ a_4] \text{ and } b=[b_1\ b_2\ b_3\ b_4]$$

$$a.*b = [a_1*b_1 \quad a_2*b_2 \quad a_3*b_3 \quad a_4*b_4]$$

$$a./b = \left[a_1/b_1 \quad a_2/b_2 \quad a_3/b_3 \quad a_4/b_4\right]$$

$$a.\hat{}\,b = \left[(a_1)^{b_1} \quad (a_2)^{b_2} \quad (a_3)^{b_3} \quad (a_4)^{b_4}\right]$$

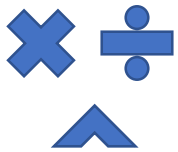
ELEMENT-BY-ELEMENT OPERATIONS

element-by-element

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

$$A \text{ .* } B = \begin{bmatrix} A_{11}B_{11} & A_{12}B_{12} & A_{13}B_{13} \\ A_{21}B_{21} & A_{22}B_{22} & A_{23}B_{23} \\ A_{31}B_{31} & A_{32}B_{32} & A_{33}B_{33} \end{bmatrix} \qquad A \text{ ./ } B = \begin{bmatrix} A_{11}/B_{11} & A_{12}/B_{12} & A_{13}/B_{13} \\ A_{21}/B_{21} & A_{22}/B_{22} & A_{23}/B_{23} \\ A_{31}/B_{31} & A_{32}/B_{32} & A_{33}/B_{33} \end{bmatrix}$$

$$\text{A .\^{} } n = \begin{bmatrix} (A_{11})^n & (A_{12})^n & (A_{13})^n \\ (A_{21})^n & (A_{22})^n & (A_{23})^n \\ (A_{31})^n & (A_{32})^n & (A_{33})^n \end{bmatrix}$$



element-by-element

ELEMENT-BY-ELEMENT OPERATIONS

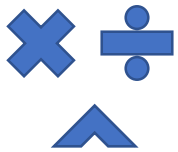
```
>> A=[2 6 3; 5 8 4]
A =
     2     6     3
     5     8     4
>> B=[1 4 10; 3 2 7]
B =
     1     4    10
     3     2     7
>> A.*B
ans =
     2    24    30
    15    16    28
>> C=A./B
C =
    2.0000    1.5000    0.3000
    1.6667    4.0000    0.5714
```

Define a 2×3 array A.

Define a 2×3 array B.

Element-by-element multiplication of array A by B.

Element-by-element division of array A by B. The result is assigned to variable C.



element-by-element


```
>> B.^3

ans =
     1     64   1000
    27      8    343

>> A*B

??? Error using ==> *
Inner matrix dimensions must agree.
```

Element-by-element exponentiation of array B. The result is an array in which each term is the corresponding term in B raised to the power of 3.

Trying to multiply A*B gives an error, since A and B cannot be multiplied according to linear algebra rules. (The number of columns in A is not equal to the number of rows in B.)

ELEMENT-BY-ELEMENT OPERATIONS

element-by-element

very useful

$$y = x^2 - 4x$$

```
>> x=[1:8]
x =
    1    2    3    4    5    6    7    8
>> y=x.^2-4*x
y =
   -3   -4   -3    0    5   12   21   32
>>
```

Create a vector x with eight elements.

Vector x is used in element-by-element calculations of the elements of vector y.

$$y = \frac{z^3+5z}{4z^2-10}$$

```
>> z=[1:2:11]
z =
     1     3     5     7     9    11
>> y=(z.^3 + 5*z)./(4*z.^2 - 10)

 y =
   -1.0000    1.6154    1.6667    2.0323    2.4650    2.9241
```

Create a vector z with six elements.

Vector z is used in element-by-element calculations of the elements of vector y.

USING ARRAYS IN MATLAB BUILT-IN MATH FUNCTIONS

```
>> x=[0:pi/6:pi]

x =
    0    0.5236    1.0472    1.5708    2.0944    2.6180    3.1416

>>y=cos(x)

y =
   1.0000    0.8660    0.5000    0.0000   -0.5000   -0.8660   -
1.0000
>>
```

```
>> d=[1 4 9; 16 25 36; 49 64 81]

d =
     1     4     9
    16    25    36
    49    64    81

>> h=sqrt(d)

h =
     1     2     3
     4     5     6
     7     8     9
```

Creating a 3×3 array.

h is a 3×3 array in which each element is the square root of the corresponding element in array d.

BUILT-IN FUNCTIONS FOR ANALYZING ARRAYS

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)`
 	
`[d,n]=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`.
	
	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).
		`>> A=[5 9 2 4 11 6 11 1];`
		`>> C=max(A)`
		`C =`
		`11`
		
		`>> [d,n]=max(A)`
		`d =`
		`11`
		`n =`
		`5`

`min(A)`
 	
`[d,n]=min(A)`
	The same as `max(A)`, but for the smallest element.
	
	The same as `[d,n]=max(A)`, but for the smallest element.
		`>> A=[5 9 2 4];`
		`>> min(A)`
		`ans =`
		`2`

`sum(A)`	If `A` is a vector, returns the sum of the elements of the vector.	`>> A=[5 9 2 4];`
		`>> sum(A)`
		`ans =`
		`20`

`sort(A)`	If `A` is a vector, arranges the elements of the vector in ascending order.	`>> A=[5 9 2 4];`
		`>> sort(A)`
		`ans =`
		`2 4 5 9`

`median(A)`	If `A` is a vector, returns the median value of the elements of the vector.	`>> A=[5 9 2 4];`
		`>> median(A)`
		`ans =`
		`4.5000`

BUILT-IN FUNCTIONS FOR ANALYZING ARRAYS

Function	Description	Example

`std(A)`	If `A` is a vector, returns the standard deviation of the elements of the vector.	`>> A=[5 9 2 4];`
		`>> std(A)`
		`ans =`
		`2.9439`

`det(A)`	Returns the determinant of a square matrix `A`.	`>> A=[2 4; 3 5];`
		`>> det(A)`
		`ans =`
		`-2`

`dot(a,b)`	Calculates the scalar (dot) product of two vectors `a` and `b`. The vectors can each be row or column vectors.	`>> a=[1 2 3];`
		`>> b=[3 4 5];`
		`>> dot(a,b)`
		`ans =`
		`26`

`cross(a,b)`	Calculates the cross product of two vectors `a` and `b`, (`axb`). The two vectors must have each three elements.	`>> a=[1 3 2];`
		`>> b=[2 4 1];`
		`>> cross(a,b)`
		`ans =`
		`-5 3 -2`

`inv(A)`	Returns the inverse of a square matrix `A`.	`>> A=[2 -2 1; 3 2 -1; 2 -3 2];`
		`>> inv(A)`
		`ans =`
		`0.2000 0.2000 0`
		`-1.6000 0.4000 1.0000`
		`-2.6000 0.4000 2.0000`

GENERATION OF RANDOM NUMBERS

rand

randn

randi

The rand command

GENERATION OF RANDOM NUMBERS

Command	Description	Example

`rand`	Generates a single random number between 0 and 1.	`>> rand`
		`ans =`
		`0.2311`

`rand(1,n)`	Generates an n-element row vector of random numbers between 0 and 1.	`>> a=rand(1,4)`
		`a =`
		`0.6068 0.4860 0.8913 0.7621`

`rand(n)`	Generates an n × n matrix with random numbers between 0 and 1.	`>> b=rand(3)`
		`b =`
		`0.4565 0.4447 0.9218`
		`0.0185 0.6154 0.7382`
		`0.8214 0.7919 0.1763`

`rand(m,n)`	Generates an m × n matrix with random numbers between 0 and 1.	`>> c=rand(2,4)`
		`c =`
		`0.4057 0.9169 0.8936 0.3529`
		`0.9355 0.4103 0.0579 0.8132`

`randperm(n)`	Generates a row vector with n elements that are random permutation of integers 1 through n.	`>> randperm(8)`
		`ans =`
		`8 2 7 4 3 6 5 1`

The rand command

GENERATION OF RANDOM NUMBERS

Sometimes there is a need for random numbers that are distributed in an interval other than (0,1), or for numbers that are integers only. This can be done using mathematical operations with the `rand` function. Random numbers that are distributed in a range (a,b) can be obtained by multiplying `rand` by $(b - a)$ and adding the product to a:

$$(b - a)*\texttt{rand} + a$$

For example, a vector of 10 elements with random values between –5 and 10 can be created by ($a = -5$, $b = 10$):

```
>> v=15*rand(1,10)-5

v =
   -1.8640    0.6973    6.7499    5.2127    1.9164    3.5174
6.9132   -4.1123    4.0430   -4.2460
```

The randi command

GENERATION OF RANDOM NUMBERS

Command	Description	Example

`randi(imax)`
 	
(imax is an integer)
	Generates a single random number between 1 and imax.	`>> a=randi(15)`
		`a =`
		`9`

`randi(imax,n)`	Generates an `n` × `n` matrix with random integers between 1 and imax.	`>> b=randi(15,3)`
		`b =`
		`4 8 11`
		`14 3 8`
		`1 15 8`

`randi(imax,m,n)`	Generates an `m` × `n` matrix with random integers between 1 and imax.	`>> c=randi(15,2,4)`
		`c =`
		`1 1 8 13`
		`11 2 2 13`

The randi command

The range of the random integers can be set to be between any two integers by typing `[imin imax]` instead of `imax`. For example, a 3×4 matrix with random integers between 50 and 90 is created by:

```
>> d=randi([50 90],3,4)
d =
    57    82    71    75
    66    52    67    61
    84    66    76    67
```

The randn command

The `randn` command generates normally distributed numbers with mean 0 and standard deviation of 1. The command can be used to generate a single number, a vector, or a matrix in the same way as the `rand` command. For example, a 3×4 matrix is created by:

```
>> d=randn(3,4)

d =
   -0.4326    0.2877    1.1892    0.1746
   -1.6656   -1.1465   -0.0376   -0.1867
    0.1253    1.1909    0.3273    0.7258
```

The mean and standard deviation of the numbers can be changed by mathematical operations to have any values. This is done by multiplying the number generated by the `randn` function by the desired standard deviation, and adding the desired mean. For example, a vector of six numbers with a mean of 50 and standard deviation of 6 is generated by:

The randn command

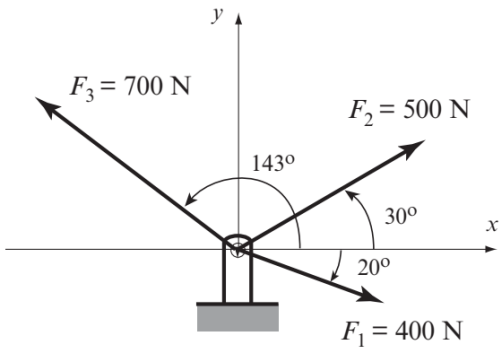

```
>> v=4*randn(1,6)+50

v =
   42.7785   57.4344   47.5819   50.4134   52.2527   50.4544
```

Integers of normally distributed numbers can be obtained by using the round function.

```
>> w=round(4*randn(1,6)+50)

w =
    51    49    46    49    50    44
```

Sample Problem 02

Three forces are applied to a bracket as shown. Determine the total (equivalent) force applied to the bracket.



Solution

A force is a vector (a physical quantity that has a magnitude and direction). In a Cartesian coordinate system a two-dimensional vector **F** can be written as:

$$\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} = F\cos\theta\mathbf{i} + F\sin\theta\mathbf{j} = F(\cos\theta\mathbf{i} + \sin\theta\mathbf{j})$$

where F is the magnitude of the force and θ is its angle relative to the x axis, F_x and F_y are the components of **F** in the directions of the x and y axes, respectively, and **i** and **j** are unit vectors in these directions. If F_x and F_y are known, then F and θ can be determined by:

$$F = \sqrt{F_x^2 + F_y^2} \quad \text{and} \quad \tan\theta = \frac{F_y}{F_x}$$

The total (equivalent) force applied on the bracket is obtained by adding the forces that are acting on the bracket. The MATLAB solution below follows three steps:

- Write each force as a vector with two elements, where the first element is the x component of the vector and the second element is the y component.
- Determine the vector form of the equivalent force by adding the vectors.
- Determine the magnitude and direction of the equivalent force.

The problem is solved in the following program, written in a script file.

Sample Problem 02


```
% Sample Problem 3-2 solution (script file)
clear
F1M=400; F2M=500; F3M=700;

Th1=-20; Th2=30; Th3=143;

F1=F1M*[cosd(Th1) sind(Th1)]
F2=F2M*[cosd(Th2) sind(Th2)]
F3=F3M*[cosd(Th3) sind(Th3)]
Ftot=F1+F2+F3
FtotM=sqrt(Ftot(1)^2+Ftot(2)^2)
Th=atand(Ftot(2)/Ftot(1))
```

Define variables with the magnitude of each vector.

Define variables with the angle of each vector.

Define the three vectors.

Calculate the total force vector.

Calculate the magnitude of the total force vector.

Calculate the angle of the total force vector.

When the program is executed, the following is displayed in the Command Window:

```
F1 =
  375.8770 -136.8081
F2 =
  433.0127  250.0000
F3 =
 -559.0449  421.2705
Ftot =
  249.8449  534.4625
FtotM =
  589.9768
Th =
   64.9453
```

The components of F_1.

The components of F_2.

The components of F_3.

The components of the total force.

The magnitude of the total force.

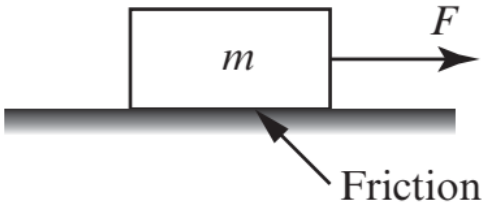
The direction of the total force in degrees.

The equivalent force has a magnitude of 589.98 N, and is directed 64.95° (ccw) relative to the x axis. In vector notation, the force is $\mathbf{F} = 249.84\mathbf{i} + 534.46\mathbf{j}$ N.

Sample Problem 03

Friction Experiment (Element-by-element Calculations)

The coefficient of friction, μ, can be determined in an experiment by measuring the force F required to move a mass m. When F is measured and m is known, the coefficient of friction can be calculated by:



$$\mu = F/(mg) \quad (g = 9.81 \text{ m/s}^2).$$

Results from measuring F in six tests are given in the table below. Determine the coefficient of friction in each test, and the average from all tests.

Test	1	2	3	4	5	6
Mass m (kg)	2	4	5	10	20	50
Force F (N)	12.5	23.5	30	61	117	294

Solution

A solution using MATLAB commands in the Command Window is shown below.

Sample Problem 03

Friction Experiment (Element-by-element Calculations)

```
>> m=[2  4  5  10  20  50];
>> F=[12.5  23.5  30  61  117  294];
>> mu=F./(m*9.81)

mu =

    0.6371    0.5989    0.6116    0.6218    0.5963    0.5994

>> mu_ave=mean(mu)

mu_ave =
    0.6109
```

Enter the values of `m` in a vector.

Enter the values of `F` in a vector.

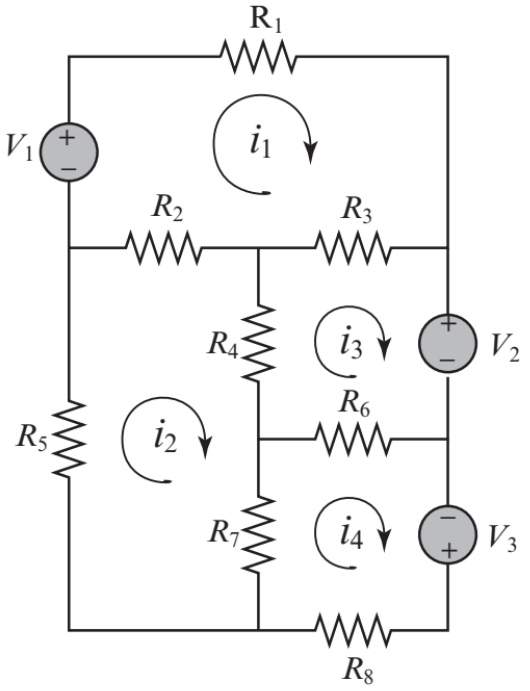
A value for `mu` is calculated for each test, using element-by-element calculations.

The average of the elements in the vector `mu` is determined by using the function `mean`.

Sample Problem 04

The electrical circuit shown consists of resistors and voltage sources. Determine the current in each resistor using the mesh current method, which is based on Kirchhoff's voltage law.

$V_1 = 20\,\text{V},\ V_2 = 12\,\text{V},\ V_3 = 40\,\text{V}$
$R_1 = 18\,\Omega,\ R_2 = 10\,\Omega,\ R_3 = 16\,\Omega$
$R_4 = 6\,\Omega,\ R_5 = 15\,\Omega,\ R_6 = 8\,\Omega$
$R_7 = 12\,\Omega,\ R_8 = 14\,\Omega$



Solution

Kirchhoff's voltage law states that the sum of the voltage around a closed circuit is zero. In the mesh current method a current is first assigned for each mesh (i_1, i_2, i_3, i_4 in the figure). Then Kirchhoff's voltage law is applied for each mesh. This results in a system of linear equations for the currents (in this case four equations). The solution gives the values of the mesh currents. The current in a resistor that belongs to two meshes is the sum of the currents in the corresponding meshes. It is convenient to assume that all the currents are in the same direction (clockwise in this case). In the equation for each mesh, the voltage source is positive if the current flows to the – pole, and the voltage of a resistor is negative for current in the direction of the mesh current.

Sample Problem 04

Electrical Resistive Network Analysis (Solving a System of Linear Equations)

The equations for the four meshes in the current problem are:

$$V_1 - R_1 i_1 - R_3(i_1 - i_3) - R_2(i_1 - i_2) = 0$$

$$-R_5 i_2 - R_2(i_2 - i_1) - R_4(i_2 - i_3) - R_7(i_2 - i_4) = 0$$

$$-V_2 - R_6(i_3 - i_4) - R_4(i_3 - i_2) - R_3(i_3 - i_1) = 0$$

$$V_3 - R_8 i_4 - R_7(i_4 - i_2) - R_6(i_4 - i_3) = 0$$

The four equations can be rewritten in matrix form $[A][x] = [B]$:

$$\begin{bmatrix} -(R_1+R_2+R_3) & R_2 & R_3 & 0 \\ R_2 & -(R_2+R_4+R_5+R_7) & R_4 & R_7 \\ R_3 & R_4 & -(R_3+R_4+R_6) & R_6 \\ 0 & R_7 & R_6 & -(R_6+R_7+R_8) \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} -V_1 \\ 0 \\ V_2 \\ -V_3 \end{bmatrix}$$

Sample Problem 04

Electrical Resistive Network Analysis (Solving a System of Linear Equations)

The problem is solved in the following program, written in a script file:

```
V1=20; V2=12; V3=40;
R1=18; R2=10; R3=16; R4=6;
R5=15; R6=8; R7=12; R8=14;
A=[-(R1+R2+R3) R2 R3 0
R2 -(R2+R4+R5+R7) R4 R7
R3 R4 -(R3+R4+R6) R6
0 R7 R6 -(R6+R7+R8)]
>> B=[-V1; 0; V2; -V3]
>> I=A\B
```

Define variables with the values of the V's and R's.

Create the matrix A.

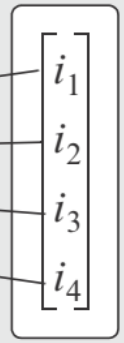
Create the vector B.

Solve for the currents by using left division.

Sample Problem 04

When the script file is executed, the following is displayed in the Command Window:

```
A =
   -44    10    16     0
    10   -43     6    12
    16     6   -30     8
     0    12     8   -34

B =
   -20
     0
    12
   -40

I =
    0.8411
    0.7206
    0.6127
    1.5750
>>
```



The numerical value of the matrix A.

The numerical value of the vector B.

The solution.

Sample Problem 04

The last column vector gives the current in each mesh. The currents in the resistors R_1, R_5, and R_8 are $i_1 = 0.8411\,\text{A}$, $i_2 = 0.7206\,\text{A}$, and $i_4 = 1.5750\,\text{A}$, respectively. The other resistors belong to two meshes and their current is the sum of the currents in the meshes.

The current in resistor R_2 is $i_1 - i_2 = 0.1205$ A.

The current in resistor R_3 is $i_1 - i_3 = 0.2284$ A.

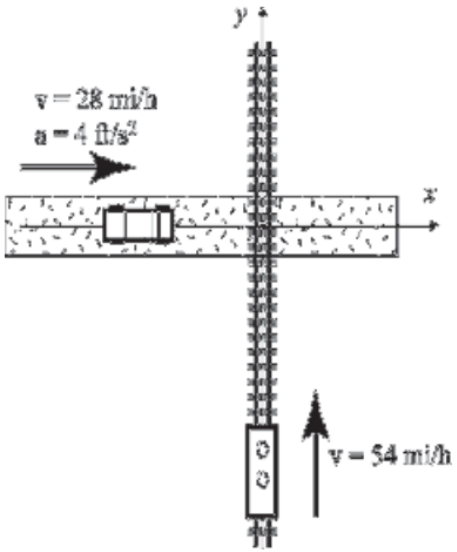
The current in resistor R_4 is $i_2 - i_3 = 0.1079$ A.

The current in resistor R_6 is $i_4 - i_3 = 0.9623$ A.

The current in resistor R_7 is $i_4 - i_2 = 0.8544$ A.

Sample Problem 05

A train and a car are approaching a road crossing. At time $t = 0$ the train is 400 ft south of the crossing traveling north at a constant speed of 54 mi/h. At the same time the car is 200 ft west of the crossing traveling east at a speed of 28 mi/h and accelerating at 4 ft/s^2. Determine the positions of the train and the car, the distance between them, and the speed of the train relative to the car every second for the next 10 seconds.



To show the results, create an 11×6 matrix in which each row has the time in the first column and the train position, car position, distance between the train and the car, car speed, and the speed of the train relative to the car in the next five columns, respectively.

Sample Problem 05

Solution

The position of an object that moves along a straight line at a constant acceleration is given by $s = s_0 + v_0 t + \frac{1}{2}at^2$ where s_0 and v_0 are the position and velocity at $t = 0$, and a is the acceleration. Applying this equation to the train and the car gives:

$$y = -400 + v_{0train}t \quad \text{(train)}$$

$$x = -200 + v_{0car}t + \frac{1}{2}a_{car}t^2 \quad \text{(car)}$$

The distance between the car and the train is: $d = \sqrt{x^2 + y^2}$. The velocity of the train is constant and in vector notation is given by $\mathbf{v}_{train} = v_{otrain}\mathbf{j}$. The car is accelerating and its velocity at time t is given by $\mathbf{v}_{car} = (v_{ocar} + a_{car}t)\mathbf{i}$. The velocity of the train relative to the car, $\mathbf{v}_{t/c}$, is given by $\mathbf{v}_{t/c} = \mathbf{v}_{train} - \mathbf{v}_{car} = -(v_{0\,car} + a_{car}t)\mathbf{i} + v_{0\,train}\mathbf{j}$. The magnitude (speed) of this velocity is the length of the vector.

The problem is solved in the following program, written in a script file. First a vector `t` with 11 elements for the time from 0 to 10 s is created, then the positions of the train and the car, the distance between them, and the speed of the train relative to the car at each time element are calculated.

Sample Problem 05

Motion of Two Particles

```
v0train=54*5280/3600; v0car=28*5280/3600; acar=4;
```

Create variables for the initial velocities (in ft/s) and the acceleration.

```
t=0:10;
```

Create the vector `t`.

```
y=-400+v0train*t;
x=-200+v0car*t+0.5*acar*t.^2;
```

Calculate the train and car positions.

```
d=sqrt(x.^2+y.^2);
```

Calculate the distance between the train and car.

```
vcar=v0car+acar*t;
```

Calculate the car's velocity.

```
speed_trainRcar=sqrt(vcar.^2+v0train^2);
```

Calculate the speed of the train relative to the car.

```
table=[t' y' x' d' vcar' speed_trainRcar']
```

Create a table (see note below).

Sample Problem 05

Note: In the commands above, `table` is the name of the variable that is a matrix containing the data to be displayed.

When the script file is executed, the following is displayed in the Command Window:

```
table =
         0 -400.0000 -200.0000  447.2136   41.0667   89.2139
    1.0000 -320.8000 -156.9333  357.1284   45.0667   91.1243
    2.0000 -241.6000 -109.8667  265.4077   49.0667   93.1675
    3.0000 -162.4000  -58.8000  172.7171   53.0667   95.3347
    4.0000  -83.2000   -3.7333   83.2837   57.0667   97.6178
    5.0000   -4.0000   55.3333   55.4777   61.0667  100.0089
    6.0000   75.2000  118.4000  140.2626   65.0667  102.5003
    7.0000  154.4000  185.4667  241.3239   69.0667  105.0849
    8.0000  233.6000  256.5333  346.9558   73.0667  107.7561
    9.0000  312.8000  331.6000  455.8535   77.0667  110.5075
   10.0000  392.0000  410.6667  567.7245   81.0667  113.3333
```

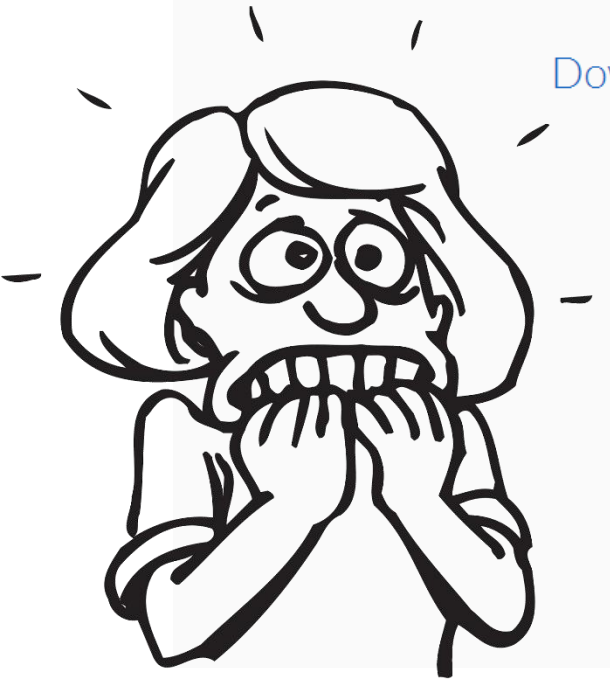
Time (s)	Train position (ft)	Car position (ft)	Car-train distance (ft)	Car speed (ft/s)	Train speed relative to the car (ft/s)

PROBLEMS

POOYA SHIRAZI Blog General Irrigation Computer Programming Upload Your Files

MATLAB Assignment - Session 06

This page contains your homework assignments. Please first download **"Download MATLAB Assignment – Session 06"**, then based on the number of your group complete your assignments.



Download MATLAB Assignment - Session 06

Group	Problem Number				
A	1	9	17	25	33
B	2	10	18	26	34
C	3	11	19	27	35
D	4	12	20	28	36
E	5	13	21	29	37
F	6	14	22	30	-
G	7	15	23	31	-
H	8	16	24	32	-