Array and Matrix Functions

EE 201

Array Functions

I=find(V)

returns the linear indices(**Elements Position Number**) corresponding to the nonzero entries of the array V.

Example:

Command Window

```
>> V=[3 -2 0 4 7 0];
>> I=find(V)

I =

     1     2     4     5
```

Example

Command Window

```
>> A=[6 0 3;0 4 0;2 7 0];
>> s=find(A)

s =

     1
     3
     5
     6
     7
```

Array Functions

length(A)

Computes either the number of elements of A if A is a vector or the largest value of m or n if A is an $m \times n$ matrix.

- A=[2 4 6 8 10 12];-----Vector

length(A) = 6

Matrix

- A = 2 4 6

  8 10 12

  14 16 18

>> length(A)= 3 Because: m×n=3×3

Size of the Matrix: size(A)

- a =

1	2	1	2	2
2	1	1	1	2
1	2	1	1	2

m×n=3×5, length(a) = 5

- a =

1	2	2
1	2	2
1	2	1
1	1	2

m×n=4×3 ,length(a) = 4

Example

Command Window

```
>>  u=[0:0.1:10];
>>  w=5*sin(u);
>>  length(u)

ans  =

    101

>>  length(w)

ans  =

    101
```

Array Functions

max(A)
Vector:

A=[9 10 40 8];

Matrix:

A =

41	32	48	48
46	5	49	25
7	14	8	41
46	28	49	8

Returns the algebraically largest element in A if **A** is a vector.

max(A)

Ans=40

Returns a row vector containing the largest elements in each column if **A** is a matrix.

>> max(A)

ans = 46 32 49 48

>>max(A,[],2) –largest elements in row-wise.

Array Functions

min(A)

Vector:

```
A=[9 10 40  8 ];
```

Matrix:

```
A =
    22    48    43
    46    33    47
    40     2    34
```

Returns the algebraically smallest element in A if **A** is a vector.

```
min(a)
ans=8
```

Returns a row vector containing the smallest elements in each column if **A** is a matrix.

```
min(A)
ans =22     2    34
```

A =

5	**2**	**4**
3	**1**	**6**
7	**9**	**8**

size(A)

Returns a row vector [m n] containing the sizes of the *m* x *n* array A.

```
        R    C
ans =   3    3
```

sort(A)

Sorts each column of the array A in ascending order and returns an array the same size as A.

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

sum(A)

Sums the elements in each column of the array A and returns a row vector containing the sums.

```
ans= 15   12   18
```

If a is matrix

A =5 2 4

3 1 6

7 9 8

- max(max(a))- Display max value of its elements.

Ans=9

- min(min(a))- Display min value of its elements

Ans=1

- sort(a,2)- Sort the elements row-wise

ans =

2 4 5

1 3 6

7 8 9

- sum(a(1,:))- Sum of all the elements in 1st row. Ans=11
- sum(a(:,1))- Sum of all the elements in 1st column. Ans=15
- sum(sum(A))- Sum of all the elements in matrix Ans=45

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

- sort(A,'descend')- Column-wise Big to Small

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

- sort(A,2,'descend')-Row-wise Big to Small

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

- For example, if $\mathbf{A}=\begin{bmatrix} 6 & 2 \\ -10 & -5 \\ 3 & 0 \end{bmatrix}$

-max(A)  [6,2]

-min(A)  [-10,-5]

-size(A)  [3,2]

-length(A)  3

-sort (A)  -10 -5
3 0
6 2

-sum(A)  -1 -3

- **zeros:** to create a matrix with all elements zero.
  - zeros(n): it creates n by n matrix with all elements zero.

```
d=zeros(3)
d =
 0     0     0
 0     0     0
 0     0     0
```

  - zeros(m,n); create the m by n matrix with all elements zero

```
d=zeros(2,3)
d=
     0     0     0
     0     0     0
```

  - zeros(size(A)): create the zero matrix for the size of A Matrix

```
if A =
     2     3
     6     8          size of A is 2 by 2

d= zeros(size(A))          size of d also 2 by 2
d =
     0     0
     0     0
```

- **ones:** It is used to create the matrix with all elements one. Syntax of ones command is similar as zeros.

Like,

- ones(n)
- ones(m,n)
- ones(size(A))

Generating Vectors from functions

- zeros(m,n)

mxn matrix of zeros

```
x = zeros(1,3)
x =
     0     0     0
```

- ones(m,n)

mxn matrix of ones

```
x = ones(1,3)
x =
     1     1     1
```

Example:

```
Command Window
>>  zeros(3)

ans  =

     0     0     0
     0     0     0
     0     0     0

>>  zeros(3,2)

ans  =

     0     0
     0     0
     0     0

>>  ones(3,3)

ans  =

     1     1     1
     1     1     1
     1     1     1
```

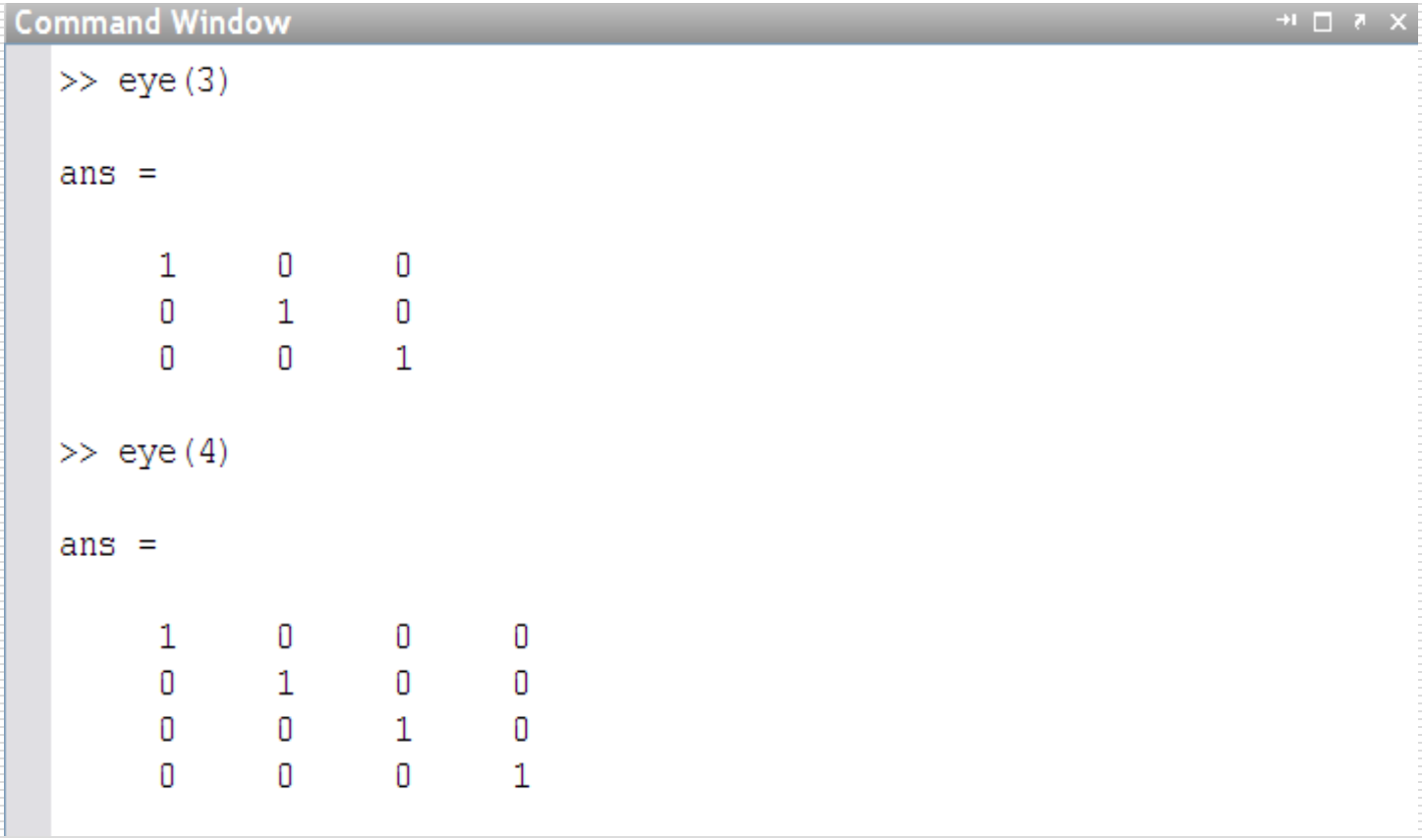
The identity matrix is a square matrix whose diagonal elements are all equal to one, with the remaining elements equal to zero.

***For example*, the 4 × 4 identity matrix is**

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The functions `eye(n)` and `eye(size(A))` create an *n* × *n* identity matrix and an identity matrix the same size as the matrix `A`.

Example:


Command Window
>> eye(3)
ans =
1 0 0
0 1 0
0 0 1
>> eye(4)
ans =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1