

## CUPRINS - CURS #2

# INTRODUCERE ÎN SOFTWARE MATEMATIC CURS #2

Liviu Marin

Facultatea de Matematică și Informatică, Universitatea din București, România

E-mails: marin.liviu@gmail.com; liviu.marin@fmi.unibuc.ro

- 1 **Tablouri (Arrays): definire, apelare**
  - Crearea unui tablou unidimensional (vector)
  - Crearea unui tablou bidimensional (matrice)
  - Apelarea elementelor unui tablou
  - Adăugarea unor elemente la un tablou
  - Ștergerea unor elemente ale unui tablou
  - Funcții predefinite pentru manipularea tablourilor
  - Șiruri de caractere (Strings)
- 2 **Operații matematice cu tablouri**
  - Adunarea și scăderea
  - Înmulțirea tablourilor
  - Împărțirea tablourilor
  - Operații element cu element cu tablouri
  - Utilizarea tablourilor în funcții predefinite

Liviu Marin

Introducere în software matematic: Curs #2

Tablouri (Arrays): definire, apelare  
Operații matematice cu tablouri

Liviu Marin

Introducere în software matematic: Curs #2

Tablouri (Arrays): definire, apelare  
Operații matematice cu tablouri

Crearea unui tablou unidimensional (vector)  
Crearea unui tablou bidimensional (matrice)  
Apelarea elementelor unui tablou  
Adăugarea unor elemente la un tablou  
Ștergerea unor elemente ale unui tablou  
Funcții predefinite pentru manipularea tablourilor  
Șiruri de caractere (Strings)

## CUPRINS - CURS #2 (cont.)

- Funcții predefinite pentru analiza tablourilor
- Generarea numerelor aleatoare

## Tablouri (Arrays): definire, apelare

- **Tablou** este forma fundamentală folosită de MATLAB pentru stocarea și manipularea datelor.
- Un tablou este o listă/însiruire de numere aranjate pe linii și/sau coloane.
- Cel mai simplu tablou: o linie sau o coloană de numere, i.e. **vector**.
- Al doilea tip de tablou d.p.d.v. al complexității: o listă de numere aranjate pe linii și pe coloane, i.e. **matrice**.
- Tablourile pot fi și o listă/însiruire de caractere, i.e. **șiruri caracter**.

## Crearea unui tablou unidimensional (vector)

### ► Crearea directă a unui vector:

- o se face prin introducerea explicită a elementelor, delimitate de paranteze drepte [ ]
- o elementele unui **vector linie** se separă fie printr-un spațiu, fie prin caracterul virgulă (,)
- o elementele unui **vector coloană** se separă fie prin caracterul punct și virgulă (;), fie prin apăsarea tastei **Enter**

### EXEMPLUL#1: Crearea directă a unui vector linie sau coloană

```
>> a = [1 5 7];   Vector linie, i.e. matrice de tip (1,4)
>> b = [1, 5, 7]; Vector linie, i.e. matrice de tip (1,3)
>> c = b';        Transpusul unui vector linie, i.e. vector coloană,
                  matrice de tip (3,1)
>> d = [2; 4; 6; 8]; Vector coloană, i.e. matrice de tip (4,1)
```

### ► Crearea unui vector linie cu elemente echidistante și pasul dat:

**Vector = xi : step : xf**

- o **Vector** este numele variabilei vectoriale;
- o **xi** este valoarea primului element al vectorului;
- o **step** este valoarea pasului dintre două elemente consecutive;
- o **xf** este valoarea maximă admisibilă a ultimului element al vectorului.

### EXEMPLUL#2: Crearea unui vector linie cu elemente echidistante și pasul dat

```
>> x = 1 : 2 : 8   Creează vectorul linie x
x =               MATLAB afișează variabila x
    1 3 5 7        și valoarea atribuită variabilei x
```

### ► Crearea unui vector linie cu elemente echidistante și număr de elemente dat:

**Vector = linspace(xi, xf, N)**

- o **Vector** este numele variabilei vectoriale;
- o **xi** este valoarea primului element al vectorului;
- o **xf** este valoarea ultimului element al vectorului;
- o **N** este numărul de elemente al vectorului.

### EXEMPLUL#3: Crearea unui vector linie cu elemente echidistante și număr de elemente dat

```
>> x = linspace(1,3,5)   Creează vectorul linie x
x =                       MATLAB afișează variabila x
    1.0000 1.5000 2.0000 2.5000 3.0000   și valoarea atribuită variabilei x
```

## Crearea unui tablou bidimensional (matrice)

### ► Crearea directă a unei matrice:

- o se face prin introducerea explicită a elementelor matricei, delimitate de paranteze drepte [ ]
- o **elementele unei linii a unei matrice** se separă între ele fie printr-un spațiu, fie prin caracterul virgulă (,)
- o **elementele unei linii a unei matrice** pot fi definite în același mod în care se definesc vectorii linie
- o **liniile unei matrice** se separă între ele fie prin caracterul punct și virgulă (;), fie prin apăsarea tastei **Enter**

- Valorile atribuite elementelor unei matrice pot fi numere sau expresii matematice care pot conține numere, variabile predefinite și funcții
- Toate liniile unei matrice trebuie să aibă același număr de elemente!

#### EXEMPLUL#4: Crearea directă a unei matrice

```
>> A = [5 35 43; 4 76 81; ...      Matrice de tip (4,3)
21 32 40; 2 3 4];
>> B = [1 5 7 9                    Matrice de tip (3,4)
2 4 6 8
1 4 6 9];
>> C = B';                          Transpusa unui matrice de tip (3,4)
>> D = [1 : 2 : 7; linspace(2,8,4); ... Matrice de tip (3,4)
1 4 6 9];
```

Funcție	Descriere
<code>zeros(m,n)</code>	$\mathbf{O}_{m,n} \in \mathcal{M}_{m,n}(\mathbb{R})$
<code>zeros(n)</code>	$\mathbf{O}_n \in \mathcal{M}_n(\mathbb{R})$
<code>ones(m,n)</code>	$\mathbf{A} = (a_{ij})_{i=1, \dots, m; j=1, \dots, n} \in \mathcal{M}_{m,n}(\mathbb{R})$ , cu $a_{ij} = 1$ , $i = \overline{1, m}$ , $j = \overline{1, n}$
<code>ones(n)</code>	$\mathbf{A} = (a_{ij})_{i,j=1, \dots, n} \in \mathcal{M}_n(\mathbb{R})$ , cu $a_{ij} = 1$ , $i, j = \overline{1, n}$
<code>eye(m,n)</code>	$\mathbf{I}_{m,n} = (\delta_{ij})_{i=1, \dots, m; j=1, \dots, n} \in \mathcal{M}_{m,n}(\mathbb{R})$
<code>eye(n)</code>	$\mathbf{I}_n = (\delta_{ij})_{i,j=1, \dots, n} \in \mathcal{M}_n(\mathbb{R})$

Table: Funcțiile zeros, ones și eye

#### Observații (asupra variabilelor din MATLAB):

##### 1) Toate variabilele din MATLAB sunt tablouri:

- o un **scalar** este un tablou cu un element;
- o un **vector** este un tablou cu o linie sau o coloană de elemente;
- o o **matrice** este un tablou cu elementele situate pe linii și pe coloane.

##### 2) Variabilele (scalare, vectoriale sau matriciale) sunt definite de inputuri atunci când elementelor variabilei respective le sunt atribuite valori, i.e. nu trebuie definită în prealabil dimensiunea variabilei.

##### 3) Odată ce o variabilă a fost definită, dimensiunile și tipul acesteia pot fi schimbate. Modificările dimensiunii și/sau ale tipului unei variabile pot fi făcute prin adăugarea sau ștergerea elementelor variabilei respective.

#### Apelarea elementelor unui tablou

- Se face prin **menționarea poziției** acestora.
- Se poate face **individual** sau **în grup**.

Apelare	Descriere
<code>v(k)</code>	Apelarea elementului vectorului <b>v</b> de pe poziția <b>k</b>
<code>v(:)</code>	Apelarea tuturor elementelor vectorului <b>v</b>
<code>v(m : n)</code>	Apelarea elementelor vectorului <b>v</b> de pe pozițiile <b>m, m + 1, ..., n</b>
<code>v([k, m, n])</code>	Apelarea elementelor vectorului <b>v</b> de pe pozițiile <b>k, m și n</b>
<code>v([k, m : n])</code>	Apelarea elementelor vectorului <b>v</b> de pe pozițiile <b>k și m, m + 1, ..., n</b>

Table: Apelarea elementelor unui vector

Apelare	Descriere
$A(m, n)$	Apelarea elementului matricei $A$ din linia $m$ și coloana $n$
$A(m, :)$	Apelarea tuturor elementelor matricei $A$ din linia $m$
$A(:, n)$	Apelarea tuturor elementelor matricei $A$ din coloana $n$
$A(m : n, :)$	Apelarea tuturor elementelor matricei $A$ din liniile $m, m + 1, \dots, n$
$A(:, m : n)$	Apelarea tuturor elementelor matricei $A$ din coloanele $m, m + 1, \dots, n$
$A([m, n], :)$	Apelarea tuturor elementelor matricei $A$ din liniile $m$ și $n$
$A([m, n], :)$	Apelarea tuturor elementelor matricei $A$ din coloanele $m$ și $n$
$A([m : n], [p : q])$	Apelarea tuturor elementelor matricei $A$ din liniile $m, m + 1, \dots, n$ și coloanele $p, p + 1, \dots, q$
$A([m, n], [p, q])$	Apelarea tuturor elementelor matricei $A$ din liniile $m$ și $n$ și coloanele $p$ și $q$

Table: Apelarea elementelor unei matrice

### Adăugarea unor elemente la un tablou

- Se pot adăuga noi elemente unui **vector** deja definit prin alocarea de valori acestor noi elemente ale vectorului.

```
>> v = 1 : 3      Definește vectorul v cu 3 elemente
>> v =
      1 2 3
>> v(4 : 6) = 1 : 2 : 5  Adaugă 3 noi elemente
>> v =
      1 2 3 1 3 5
>> w = 2 : 4      Definește vectorul w cu 3 elemente
>> w =
      2 3 4
>> w(6) = 5      Alocă valoarea 5 celui de-al șaselea element
>> w =
      2 3 4 0 0 5
```

- Se pot adăuga noi elemente unui **vector** deja definit prin alipirea unui alt vector definit.

```
>> v = 1 : 3;      Definește vectorul v cu 3 elemente
>> w = 2 : 4;      Definește vectorul w cu 3 elemente
>> u = [v w]      Alipește w la v
>> u =
      1 2 3 2 3 4
>> U = [v'; w']   Alipește w' la v'
>> U =
      1
      2
      3
      2
      3
      4
```

- Se pot adăuga noi linii și/sau coloane unei **matrice** deja definite prin alocarea de valori acestor noi linii și/sau coloane ale matricei.

```
>> A = [1 2 3; 4 5 6]  Definește matricea A de tip (2, 3)
>> A =
      1 2 3
      4 5 6
>> A(3, :) = 7 : 9      Definește noua linie 3 a lui A
>> A =
      1 2 3
      4 5 6
      7 8 9
>> B = ones(3, 2)      Definește matricea B de tip (3, 2)
>> C = [A B]           Definește matricea C prin alipirea lui B la A
>> C =
      1 2 3 1 1
      4 5 6 1 1
      7 8 9 1 1
```

- Se pot adăuga noi elemente unei **matrice** deja definite prin alocarea de valori acestor noi elemente ale matricei.

```
>> D = [1 2 3; 4 5 6]  Definește matricea D de tip (2,3)
>> D =
    1 2 3
    4 5 6
>> D(3,6) = 1          Definește noul element D(3,6) al lui D
>> D =
    1 2 3 0 0 0
    4 5 6 0 0 0
    0 0 0 0 1
>> E(2,3) = 1          Alocă o valoare elementului (2,3) al unei
                        noi matrice E
>> E =
    0 0 0
    0 0 1
```

## Ștergerea unor elemente ale unui tablou

```
>> v = [1 2 3 4 5 6]  Definește vectorul v cu 6 elemente
>> v =
    1 2 3 4 5 6
>> v(4) = []          Elimină elementul 4 al lui v
>> v =
    1 2 3 5 6
>> v(1:3) = []        Elimină elementele 1, 2, 3 ale lui v
>> v =
    5 6
>> A = [1 2 3 4; 5 6 7 8] Definește matricea A de tip (2,4)
>> A =
    1 2 3 4
    5 6 7 8
>> A(:,2:3) = []      Elimină elementele din coloanele 2 și 3 ale lui A
>> A =
    1 4
    5 8
```

## Funcții predefinite pentru manipularea tablourilor

Funcție	Descriere	Exemplu
length(v)	Numărul de elemente ale vectorului v	<pre>&gt;&gt; v = [1 3 5 7]; length(v) ans =     4</pre>
size(A)	Dimensiunile matricei A, (m,n), sub forma unui vector linie [m n]	<pre>&gt;&gt; A = [1 2 3; 4 5 6]; &gt;&gt; size(A) ans =     2 3</pre>
reshape(A,m,n)	Creează o matrice de tip (m,n) din elementele lui A luate coloană cu coloană	<pre>&gt;&gt; A = [1 2 3; 4 5 6]; &gt;&gt; B = reshape(A,3,2) B =     1 5     4 3     2 6</pre>

Funcție	Descriere	Exemplu
diag(v)	Creează o matrice pătratică a cărei diagonală este formată din elementele vectorului v	<pre>&gt;&gt; v = [1 3 5]; A = diag(v) A =     1 0 0     0 3 0     0 0 5</pre>
diag(A)	Creează un vector ale cărui componente sunt elementele de pe diagonală matricei A	<pre>&gt;&gt; A = [1 2 3; 4 5 6; 7 8 9]; &gt;&gt; v = diag(A) v =     1     5     9</pre>

- Informații despre funcții predefinite pentru manipularea tablourilor:  
[Help Window](#) → [Functions by Category](#) → [Mathematics](#) → [Arrays and Matrices](#)

## Șiruri de caractere (Strings)

- Un **șir de caractere** este un tablou alcătuit din caractere.
- Șirurile de caractere pot conține litere, cifre, alte simboluri și spații.
- Un șir de caractere se creează tipărind caractere între ghilimele simple, i.e. `'ad ef'`, `'3%fr2'`, `'{edcba:2!}'`, `'MATLAB'` etc.
- Șirurile de caractere au diverse utilizări:
  - în comenzi de afișare a unui mesaj text;
  - la formatarea comenzilor de reprezentare grafică;
  - ca argumente de intrare pentru funcții.
- Când șirurile de caractere sunt folosite la formatarea comenzilor de reprezentare grafică (etichete ale axelor, titlul figurii, note pe grafic), caracterele din cadrul acestor șiruri pot fi formate pentru a avea specificate fontul, mărimea, poziția, culoarea etc. acestora.

## Șiruri de caractere (Strings)

- Șirurile de caractere pot fi alocate variabilelor prin simpla tipărire a lor la dreapta operatorului de alocare (`=`).
- Când o variabilă este definită ca un șir de caractere, caracterele acestui șir sunt stocate într-un tablou, similar cu numerele.
  - Fiecare caracter este un element al unui tablou, i.e. un șir de caractere este un tablou unidimensional (vector linie) al cărui număr de elemente este egal cu numărul de caractere al șirului respectiv.
  - Apelarea unui anumit caracter dintr-un șir de caractere se face similar cu apelarea unui element al unui vector, i.e. prin invocarea poziției caracterului/elementului.
  - Ca și elementele unui tablou, și caracterele dintr-un șir de caractere se pot modifica prin adresarea lor directă.

## EXEMPLUL#5: Șiruri de caractere

```
>> A = 'Numele meu este Liviu Marin'  Alocarea unui șir de caractere
                                     variabilei A

>> A =
Numele meu este Liviu Marin

>> A(17:21)
                                     Apelarea caracterelor de pe
                                     pozițiile 17 – 21 ale variabilei A

'Liviu'

>> A(17:21) = 'Ionel'
                                     Înlocuirea caracterelor de pe
                                     pozițiile 17 – 21 ale variabilei A

'Ionel'

>> A
                                     Apelarea variabilei A

>> A =
Numele meu este Ionel Marin
```

## Șiruri de caractere (Strings)

- Șirurile de caractere pot fi, de asemenea, inserate într-un tablou bidimensional (matrice) prin tipărirea caracterului punct și virgulă (`:`) sau prin apăsarea tastei **Enter** la sfârșitul fiecărei linii.
- Fiecare linie trebuie tipărită ca un șir de caractere, i.e. trebuie delimitată de ghilimele simple (`'`). Acesta este un **inconvenient** pentru că toate liniile unei matrice trebuie să aibă același număr de caractere, i.e. trebuie adăugate spații pentru completare.
- Există o funcție predefinită în MATLAB, **char**, care creează un tablou bidimensional având același număr de caractere pe fiecare linie prin adăugarea unui număr corespunzător de spații acolo unde este necesar.

**Var = char('string1', 'string2', ..., 'stringN')**

- **Var** este numele variabilei;
- **string1, string2, ..., stringN** sunt șiruri de caractere.

## EXEMPLUL#6: Șiruri de caractere atribuite unei variabile tablou bidimensional

```
>> s1 = 'Nume profesor : '
>> s2 = 'Liviu MARIN'
>> s3 = 'Nume curs : '
>> s4 = 'Introducere in software matematic'
>> Info = char(s1,s2,s3,s4)

Info =
Nume profesor :
Liviu MARIN
Nume curs :
Introducere in software matematic
```

*Șir de caractere s1*  
*Șir de caractere s2*  
*Șir de caractere s3*  
*Șir de caractere s4*  
*Variabilei Info îi sunt atribuite patru linii de caractere, de lungimi diferite*

## Operații matematice cu tablouri

### Adunarea și scăderea

- Adunarea și scăderea tablourilor de **aceleași dimensiuni**
- Adunarea și scăderea **unui scalar cu un tablou**

### EXEMPLUL#1:

```
>> a = [13 57];
>> b = [24 68];
>> c = a + b
>> c =
    37    11    15
>> d = 1;
>> e = b + d
>> e =
    35    69
```

*Definește vectorul a*  
*Definește vectorul b*  
*Suma vectorilor a și b*  
*Definește scalarul d*  
*Suma vectorului b cu scalarul d*

Liviu Marin

Introducere în software matematic: Curs #2

Tablouri (Arrays): definire, apelare  
Operații matematice cu tablouri

Adunarea și scăderea  
Înmulțirea tablourilor  
Împărțirea tablourilor  
Operații element cu element cu tablouri  
Utilizarea tablourilor în funcții predefinite  
Funcții predefinite pentru analiza tablourilor  
Generarea numerelor aleatoare

Liviu Marin

Introducere în software matematic: Curs #2

Tablouri (Arrays): definire, apelare  
Operații matematice cu tablouri

Adunarea și scăderea  
Înmulțirea tablourilor  
Împărțirea tablourilor  
Operații element cu element cu tablouri  
Utilizarea tablourilor în funcții predefinite  
Funcții predefinite pentru analiza tablourilor  
Generarea numerelor aleatoare

## Înmulțirea tablourilor

- Înmulțirea tablourilor de **dimensiuni compatibile**
- Înmulțirea **unui scalar cu un tablou**

### EXEMPLUL#2:

```
>> A = [1 5; 9 7];
>> B = [1 2 3; 4 5 6];
>> C = A * B
>> C =
    21    27    33
    37    53    69
>> a = 2;
>> D = a * A
>> D =
     2     10
    18    14
```

*Definește matricea A*  
*Definește matricea B*  
*Înmulțirea matricelor A și B*  
*Definește scalarul a*  
*Înmulțirea matricei A cu scalarul a*

Liviu Marin

Introducere în software matematic: Curs #2

## EXEMPLUL#3: Funcția predefinită **dot(x, y)**, i.e. produsul scalar a doi vectori (fie linie, fie coloană) de aceeași dimensiune

```
>> a = [1 3 5];
>> b = [2; 4; 6];
>> c = a * b
>> c =
    44
>> d = dot(a,b')
>> d =
    44
>> e = dot(a',b)
>> e =
    44
```

*Definește vectorul linie a*  
*Definește vectorul coloană b*  
*Înmulțirea vectorilor a și b*  
*Funcția predefinită dot (vectori linie)*  
*Funcția predefinită dot (vectori coloană)*

Liviu Marin

Introducere în software matematic: Curs #2

## Împărțirea tablourilor

### ► Inversa unei matrice:

- o prin ridicarea matricei la puterea  $-1$
- o prin invocarea funcției predefinite `inv`

### EXEMPLUL #4: Inversa unei matrice

```
>> A = [2 14; 4 18; 2 -13]      Crearea matricei A
>> A =
    2.0000    1.0000    4.0000
    4.0000    1.0000    8.0000
    2.0000   -1.0000    3.0000

>> B = inv(A)                  Folosește funcția inv
                                pentru crearea lui A-1

    5.5000   -3.5000    2.0000
    2.0000   -1.0000    0.0000
   -3.0000    2.0000   -1.0000
```

### ► Determinantul unei matrice unei matrice:

- o prin invocarea funcției predefinite `det`, i.e. `det(A)`

### ► Împărțirea la stânga ( \ ):

- o se folosește pentru a rezolva ecuația matricială  $AX = B$ , i.e. se obține  $X = A^{-1}B$  folosind comanda MATLAB®  $X = A \backslash B$

### ► Împărțirea la dreapta ( / ):

- o se folosește pentru a rezolva ecuația matricială  $XC = D$ , i.e. se obține  $X = DC^{-1}$  folosind comanda MATLAB®  $X = D / C$

**PROBLEMA #1:** Să se rezolve, folosind funcția predefinită `inv`, respectiv operatorul de împărțire la stânga ( \ ), sistemul liniar, verificând în prealabil că sistemul are soluție:

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

## Operații element cu element cu tablouri

- Operațiile de înmulțire, împărțire și ridicare la putere **element cu element** au loc între două tablouri de aceleași dimensiuni și se realizează prin inserarea unui punct (.) în fața operatorului aritmetic corespunzător.
- Operațiile de adunare și scădere sunt implicit operații element cu element și, prin urmare, nu necesită semnalizarea de mai sus, i.e. inserarea punctului în fața operatorului aritmetic respectiv.

### EXEMPLUL #5: Înmulțirea element cu element a două tablouri

```
>> a = [1 3 5];   Vector linie, i.e. matrice de tip (1,3)
>> b = [4 2 1];   Vector linie, i.e. matrice de tip (1,3)
>> c = a .* b      Înmulțirea element cu element
                   a vectorilor a și b

>> c =
    [4 6 5]
```

### EXEMPLUL #6: Înmulțirea element cu element a două tablouri

```
>> A = [2 6 3; 5 8 4];      Matrice de tip (2,3)
>> B = [1 4 10; 3 2 7];     Matrice de tip (2,3)
>> C = A ./ B                Împărțirea element cu element
                              a matricelor A și B

>> C =
    2.0000    1.5000    0.3000
    1.6667    4.0000    0.5714
```

### EXEMPLUL #7: Ridicarea la putere element cu element a două tablouri

```
>> a = [1 3 5];   Vector linie, i.e. matrice de tip (1,3)
>> b = [4 2 2];   Vector linie, i.e. matrice de tip (1,3)
>> c = a.^b        Ridicarea la putere element cu element
                   a vectorilor a și b

>> c =
    [4 9 25]
```



**EXEMPLUL#8:** Calculul unei funcții de o variabilă în mai multe valori ale argumentului său

```
>> x = 1 : 6
>> x =
    1    2    3    4    5    6
```

Vector linie

```
>> y = x.^2 - 4 * x
```

Vectorul x este folosit  
pentru calcule element  
cu element

```
>> y =
    -3    -4    -3    0    5    12
```

```
>> z = (x.^2 - x)./(x.^2 + 1)
```

```
>> z =
    0.4000    0.6000    0.7059    0.7692    0.8108
```

**Utilizarea tablourilor în funcții predefinite**

- Funcțiile predefinite sunt scrise a.i. dacă un argument este de tip tablou, atunci operațiile care definesc funcția sunt executate asupra fiecărui element al tabloului.
- Rezultatul unei astfel de operații este un tablou având aceleași dimensiuni cu tabloul argument al funcției

**EXEMPLUL#9:** Funcții predefinite cu argumentul un tablou

```
>> x = 0 : pi/6 : pi
>> x =
    0.0000    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
```

**Funcții predefinite pentru analiza tablourilor**

Funcție	Descriere	Exemplu
mean(v)	v vector: <b>media</b> elementelor lui v; v matrice: un vector linie conținând <b>media</b> elementelor fiecărei coloane a lui v.	>> v = [1 4 6 7]; >> mean(v) >> ans = 4.5000
median(v)	v vector: <b>mediana</b> elementelor lui v; v matrice: un vector linie conținând <b>mediana</b> elementelor fiecărei coloane a lui v.	>> v = [1 4 6 7]; >> median(v) >> ans = 5
std(v)	v vector: <b>deviația standard</b> a elementelor lui v; v matrice: un vector linie conținând <b>deviația standard</b> a elementelor fiecărei coloane a lui v.	>> v = [1 4 6 7]; >> std(v) >> ans = 2.6458

Funcție	Descriere	Exemplu
max(v)	v vector: <b>maximul</b> elementelor lui v; v matrice: un vector linie conținând <b>maximul</b> elementelor fiecărei coloane a lui v.	>> v = [1 7 6 7]; >> max(v) >> ans = 7
[d, n] = max(v)	v vector: d = <b>maximul</b> elem. lui v, n = <b>poziția maximului</b> elem. lui v v matrice: un vector linie conținând <b>maximul</b> elementelor fiecărei coloane a lui v.	>> v = [1 7 6 7]; >> [d, n] = max(v) >> d = 7 >> n = 2
min(v)	v vector: <b>minimul</b> elementelor lui v; v matrice: un vector linie conținând <b>minimul</b> elementelor fiecărei coloane a lui v.	>> v = [1 7 6 7]; >> min(v) >> ans = 1

Funcție	Descriere	Exemplu
sum(v)	v vector: <b>suma</b> elementelor lui v; v matrice: un vector linie conținând <b>suma</b> elementelor fiecărei coloane a lui v.	>> v = [1 7 6 7]; >> sum(v) >> ans = 21
sort(v)	v vector: elementele lui v aranjate în <b>ordine crescătoare</b> ; v matrice: fiecare coloană a lui v aranjată în <b>ordine crescătoare</b> .	>> v = [4 7 6 1]; >> min(v) >> ans = [1 4 6 7]
det(v)	<b>determinantul</b> matricei v.	>> v = [1 7; 6 7]; >> det(v) >> ans = -35
inv(v)	<b>inversa</b> matricei pătratică v.	
dot(v,w)	<b>produsul scalar</b> al vectorilor v și w.	
cross(v,w)	<b>produsul vectorial</b> al vectorilor v, w ∈ ℝ <sup>3</sup> .	

## Generarea numerelor aleatoare

- Comanda **rand** generează numere aleatoare uniforme distribuite în intervalul [0, 1].
- Comanda **rand** este folosită pentru a asocia aceste numere aleatoare unui scalar, unui vector sau unei matrice.

Comandă	Descriere	Exemplu
rand	Generează un singur număr aleator din [0, 1]	>> rand >> ans = 0.2311
rand(1,n)	Generează un vector linie cu elementele numere aleatoare din [0, 1]	>> rand(1,3) >> ans = 0.8147 0.9058 0.1270
rand(n)	Generează o matrice pătratică n × n cu elementele numere aleatoare din [0, 1]	>> rand(3) >> ans = 0.6324 0.5469 0.1576 0.0975 0.9575 0.9706 0.2785 0.9649 0.9572

Comandă	Descriere	Exemplu
rand(m,n)	Generează o matrice m × n cu elementele numere aleatoare din [0, 1]	>> rand(2,3) >> ans = 0.4854 0.1419 0.9157 0.8003 0.4218 0.7922
randperm(n)	Generează un vector linie cu elementele o permutare aleatoare a numerelor 1, 2, ..., n	>> randperm(4) >> ans = 3 2 1 4

- Comanda **rand** poate fi folosită pentru generarea de numere aleatoare uniforme distribuite într-un interval [a, b] ≠ [0, 1]:

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

- Comanda **randi** generează un număr întreg aleatoare uniform distribuit.
- Comanda **randi** poate fi folosită pentru a asocia aceste numere întregi aleatoare unui scalar, unui vector sau unei matrice.

Comandă	Descriere	Exemplu
randi(imax)	Generează un singur număr aleator din [1, imax] cu imax ∈ ℕ	>> randi(15) >> ans = 9
randi(imax,n)	Generează o matrice pătratică n × n cu elementele numere întregi aleatoare din [0, imax]	>> randi(15,3) >> ans = 4 8 11 14 3 8 1 15 8
rand(imax,m,n)	Generează o matrice m × n cu elementele numere întregi aleatoare din [1, imax]	>> randi(2,3) >> ans = 5 2 11 1 13 5

- ▶ Comanda `randn` generează un număr aleator distribuit normal cu media 0 și cu deviația standard 1.
- ▶ Comanda `randn` poate fi folosită pentru a asocia aceste numere aleatoare unui scalar, unui vector sau unei matrice.
- ▶ Comanda `randn` poate fi folosită pentru generarea de numere aleatoare distribuite normal cu media  $m$  și cu deviația standard  $s$ :

$$s * \text{rand}(2,3) + m$$

- ▶ Numere întregi aleatoare distribuite normal cu media  $m$  și cu deviația standard  $s$  pot fi generate folosind funcția predefinită `round`:

$$\text{round}(s * \text{rand}(1,3) + m)$$