

# 1 Matlab

MATLAB (MATrix LABoratory) este un limbaj de nivel înalt și un mediu interactiv dedicat calculului numeric, vizualizării și programării. MATLAB facilitează analiza datelor, dezvoltarea de algoritmi, crearea de modele și aplicații. Limbajul, uneltele și funcțiile matematice încorporate facilitează explorarea unor abordări diverse și găsirea unei soluții într-un timp mai scurt decât utilizând aplicații de calcul tabelar sau limbaje de programare tradiționale. Utilizările tipice includ: matematică și calcul numeric, dezvoltare de algoritmi, modelare și simulare, achiziție de date, analiză a datelor, explorare și vizualizare, dezvoltare de aplicații (inclusiv construirea de interfețe grafice cu utilizatorul). MATLAB-ul include o serie de soluții specifice anumitor domenii, numite TOOLBOX-uri. Acestea sunt colecții extinse de funcții MATLAB care extind mediul, pentru a rezolva probleme specifice din domenii variate.

## 1.1 Funcții Matlab de interes general

Lansarea în execuție a MATLAB-ului se poate face selectând Matlab din meniul *Start* al sistemului de operare (Windows). Procedura de lansare în execuție afișează utilizatorului fereastra de comandă unde linia de comandă este indicată prin `'>>'`.

Informații referitoare la comenzi Matlab pot fi obținute utilizând comanda *help* subiect din linia de comandă sau din fereastra Help a meniului Help. Alte comenzi ajutătoare sunt *whos* (afișează informații despre variabilele din spațiul de lucru), *what* (afișează calea pentru directorul curent și numele fișierelor și directoarelor relevante pentru Matlab aflate în directorul curent), și *which* (localizare funcții și fișiere). O altă comandă utilă este *lookfor cuvânt\_căutat* - pentru a căuta în fișierele help acest cuvânt. MATLAB conține de asemenea o varietate de exemple care pot fi afișate utilizând comanda *demo*.

### 1.1.1 Funcții de referitoare la timp/dată

<i>clock</i>	Timpul și data curentă sub forma unui vector
<i>date</i>	Data curentă sub formă de sir de caractere
<i>cputime</i>	Timpul CPU total (în secunde) utilizat de aplicația MATLAB din momentul pornirii
<i>tic, toc</i>	Timer pentru măsurarea timpului de execuție a unei secvențe de cod.

### 1.1.2 Formate de afișare a numerelor

Prin intermediul funcției *format*, MATLAB-ul pune la dispoziție utilizatorilor două moduri de control a afișării valorilor numerice pe ecran: formatul de afișare și spațierea.

Comandă	Rezultat
format short	reprezentare în virgulă fixă, 5 cifre
format long	reprezentare în virgulă fixă, 15 cifre
format short e	notație științifică, 5 cifre și exponent
format long e	notație științifică, 5 cifre și exponent
format short g	reprezentare în virgulă fixă sau flotantă, 5 cifre
format long g	reprezentare în virgulă fixă sau flotantă, 15 cifre
format hex	reprezentare hexazecimală
format +	+, -, spațiu
format bank	format bancar, două zecimale
format rat	exprimare rațională

Comenzi pentru spațiere:

format compact Suprimă liniile libere  
format loose

Sunt disponibile pentru controlul asupra formatului de ieșire și funcțiile *sprintf* și *fprintf*.

### 1.1.3 Funcții speciale și constante

	<i>ans</i>	Ultimul rezultat afișat
	<i>pi</i>	3.14159265...
Mai multe funcții speciale furnizează valori ale unor constante utile:	<i>i, j</i>	Unitatea imaginară
	<i>Inf</i>	Infinit
	<i>NaN</i>	Not a number

## 1.2 Matrice și operații cu matricele

### 1.2.1 Variabile în Matlab

În MATLAB nu sunt necesare declarații de tip sau de dimensiune. Când se întâlnește un nou nume de variabilă, variabila se creează automat și i se alocă spațiu în memorie. Dacă variabila există deja, MATLAB schimbă conținutul variabilei existente și, dacă este necesar, îi alocă mai mult spațiu în memorie.

Numele variabilelor constau dintr-o literă, urmată de un număr de litere, cifre, sau caracterul `'.'`. MATLAB este *case sensitive* (distinge între litere mari și litere mici).

MATLAB stochează variabilele sub forma matricelor de dimensiune  $M \times N$ , unde  $M$  este numărul de linii și  $N$  numărul de coloane. O matrice de dimensiune  $1 \times 1$  este un scalar; o matrice  $1 \times N$  este un vector linie, și o matrice  $M \times 1$  este un vector coloană. Elementele unei matrice pot fi numere reale sau complexe.

De exemplu:

```
Scalar real      >> x = 5
Scalar complex  >> x = 5 + 10j (sau x = 5 + 10i)
Vector linie    >> x = [1 2 3] (sau x = [1, 2, 3])
Vector coloană  >> x = [1; 2; 3]
Matrice 3 x 3   >> x = [1 2 3; 4 5 6; 7 8 9]
```

Dimensiunea unei variabile se poate obține utilizând funcțiile *size* și *length*.

### 1.2.2 Generarea matricelor

Exista o serie de funcții pentru generarea matricelor elementare:

<i>eye</i>	Matricea identitate
<i>zeros</i>	Toate elementele zero
<i>ones</i>	Toate elementele unu
<i>rand</i>	Elemente aleatoare cu distribuție uniformă
<i>randn</i>	Elemente aleatoare cu distribuție normală

Vectorii pot fi generați utilizând operatorul `':'`. De exemplu, pentru a genera un vector  $x$  de la 0 la 10 cu pasul 0.5: `>> x = [0 : 0.5 : 10]`. Comanda *linspace* poate fi de asemenea utilizată pentru generarea vectorilor: `>> x = linspace(0, 10, 21)`.

### 1.2.3 Accesarea elementelor unei matrice

Elementele unei matrice pot fi accesate specificând numărul liniei și al coloanei. De exemplu, în matricea specificată de  $A = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$ , elementul din prima linie și a treia coloană poate fi accesat utilizând comanda `>> x = A(1,3)`, rezultând valoarea 3. A doua linie poate fi accesată cu `>> y = A(2,:)` rezultând  $[4 \ 5 \ 6]$ , unde `':'` înseamnă "returnează toate elementele coloanei". O submatrice a lui  $A$  constând din liniile 1 și 2 și toate trei coloanele este specificată de `>> z = A(1 : 2, 1 : 3)` rezultând  $[1 \ 2 \ 3; 4 \ 5 \ 6]$ .

### 1.2.4 Operații aritmetice asupra matricelor

Operațiile aritmetice de bază asupra matricelor sunt:

+	adunare
-	scădere
*	înmulțire
^	ridicare la putere
/	împărțire la dreapta
\	împărțire la stânga
'	transpusa

În cazul în care dimensiunea matricelor este incompatibilă, se afișează un mesaj de eroare.

Împărțirea este definită astfel: soluția ecuației  $A * x = b$  este  $x = A \backslash b$  și soluția ecuației  $x * A = b$  este  $x = b / A$  dacă  $A$  este inversabilă și dimensiunile matricelor sunt compatibile. Operațiile  $x = A \backslash b$  și  $x = b / A$  sunt în principiu aceleași ca  $inv(A) * b$ , respectiv  $b * inv(A)$ , unde  $inv(A)$  calculează inversa matricei  $A$ , însă operatorii de împărțire la stânga și la dreapta sunt mai eficienți.

```
n = 1000;
A = rand(n);
b = rand(n,1);
tic; x = A\b; t = toc
tic; x = inv(A) * b; t = toc
```

Adunarea și scăderea a două matrice implică operații aritmetice element-cu-element; înmulțirea și împărțirea nu. MATLAB furnizează operații aritmetice element-cu-element, indicate prin utilizarea simbolului `'.'` înainte de operator:

```
.*   înmulțire
.^   ridicare la putere
./   împărțire la dreapta
.\   împărțire la stânga
.'   transpusa
```

Diferența între înmulțirea a două matrice și înmulțirea element-cu-element a două matrice se poate observa din următorul exemplu:

```
>> A = [1 2; 3 4]   >> B = A * A   >> C = A .* A
A =                B =                C =
 1  2                7 10                1  4
 3  4                15 22               9 16
```

### 1.2.5 Operatori relaționali

```
<    mai mic
<=   mai mic sau egal
>    mai mare
>=   mai mare sau egal
==   egal
~=   diferit
```

Rezultatul aplicării unui operator relațional este de tip logic (adevărat (1) sau fals(0)). Dacă operatorii se aplică unui scalar rezultatul va fi un scalar, aplicând operatorii asupra matricelor, rezultatul va consta într-o matrice cu valori zero și unu.

```
3 < 5
3 == 5
A = [12; 34]
A >= 2
B = [13; 42]
A < B
```

### 1.2.6 Operatori logici

```
&, &&   și
|, ||    sau
~        nu
```

Spre deosebire de operatorii `&&` și `||` care pot fi aplicați doar asupra valorilor scalare și evaluează mai întâi expresia din stânga, iar dacă rezultatul este fals nu se mai evaluează expresia din dreapta, operatorii `&`, `și` și `|` pot fi aplicați atât asupra valorilor scalare cât și asupra matricelor.

### 1.3 Funcții matematice

<i>sin</i>	sinus
<i>cos</i>	cosinus
<i>tan</i>	tangentă
<i>exp</i>	exponențial
<i>log</i>	logaritm natural
<i>log10</i>	logaritm
<i>sqrt</i>	rădăcina pătrată
<i>abs</i>	valoare absolută
<i>sign</i>	semn

### 1.4 Instrucțiuni de control

Instrucțiunile de control în Matlab includ: *for*, *while*, *if* și *switch*, fiecare dintre aceste instrucțiuni necesitând cuvântul cheie *end* pentru încheiere.

Instrucțiuni de control logic:

*if/elseif/else*

*switch/case/otherwise*.

Instrucțiuni iterative:

*for*

*while*

### 1.5 Numere complexe

Matlab permite lucrul cu numerele complexe. Variabilele *i* și *j* reprezintă unitatea imaginară  $\sqrt{-1}$ . Metodele de generare a matricelor complexe includ:

$B = [1\ 2; 3\ 4] + i * [5\ 6; 7\ 8]$

$B = [1 + 5i, 2 + 6i; 3 + 7i, 4 + 8i]$

$B = \text{complex}([1\ 2; 3\ 4], [5\ 6; 7\ 8])$ .

#### 1.5.1 Funcții pentru numere complexe

Partea reală a lui $x$	<code>&gt;&gt; real(x)</code>
Partea imaginară a lui $x$	<code>&gt;&gt; imag(x)</code>
Amplitudinea (modulul) lui $x$	<code>&gt;&gt; abs(x)</code>
Argumentul lui $x$	<code>&gt;&gt; angle(x)</code>
Conjugatul complex	<code>&gt;&gt; conj(x)</code>

### 1.6 Fișiere MATLAB

#### 1.6.1 Fișiere *M*

MATLAB-ul este un limbaj interpretativ. Pentru a evita scrierea repetată a unei secvențe de cod utilizată pentru realizarea unei funcționalități, capacitățile Matlab-ului pot fi extinse prin crearea de scripturi și funcții. Ambele utilizează

fişiere *m* (numite astfel deoarece au extensia *.m*). Avantajul fişierelor *m* este faptul că comenzile sunt salvate şi pot fi modificate uşor fără a fi necesară rescrierea întregii secvenţe de comenzi.

Fişierele MATLAB *script* sunt secvenţe de comenzi scrise cu un editor şi salvate într-un fişier *m*. Nu se acceptă parametri de intrare, nu se returnează parametri, toate variabilele utilizate sunt salvate în spaţiul de lucru al MATLAB-ului. Pentru rularea scriptului se poate scrie în linia de comandă numele fişierului, spre exemplu pentru fişierul *nume\_fişier.m* : `>> nume_fişier`.

Un al doilea tip de fişier *m* este fişierul *funcţie*. Funcţiile pot accepta parametri de intrare şi pot returna parametri. Variabilele interne sunt locale funcţiei. Forma generală a unei funcţii este:

```
function [out, out2] = nume_funcţie(in1, in2)
%
%Comentarii pentru a explica ce face funcţia
%
comanda MATLAB 1;
```

...

Numele fişierului *m* pentru această funcţie va fi *nume\_funcţie.m*. Prima linie a funcţiei începe cu cuvântul cheie *function*, şi specifică numele funcţiei şi ordinea parametrilor. În acest caz, sunt specificaţi doi parametri de intrare şi doi parametri de ieşire. Următoarele trei linii încep cu % şi sunt comentarii. Restul fişierului este codul MATLAB care defineşte funcţia. Funcţia poate fi apelată din linia de comandă sau dintr-un alt fişier *m*:

```
>> [out1, out2] = nume_funcţie(in1, in2)
```

Comentariile funcţiei pot fi afişate utilizând comanda:

```
>> help nume_funcţie
```

Afişarea pe ecran a rezultatelor poate fi suprimată terminând linia cu `;`. O linie poate fi extinsă scriind `'...'` la sfârşitul liniei şi continuând instrucţiunile pe linia următoare.

### 1.6.2 Fişiere *mat*

Fişierele *mat* (denumite astfel datorită extensiei *.mat*) sunt fişiere binare utilizate pentru a stoca rezultate numerice. De exemplu, salvarea valorii a două variabile, *var1* şi *var2* într-un fişier *nume\_fişier.mat* se poate face utilizând comanda:

```
>> save nume_fişier.mat var1 var2
```

Salvarea tuturor variabilelor existente în spaţiul de lucru al Matlab-ului într-un fişier se poate face utilizând comanda:

```
>> save nume_fişier.mat
```

Un fişier *mat* poate fi încărcat în MATLAB prin comanda: `>> load nume_fişier` (sau `load nume_fişier.mat`).

## 1.7 Reprezentări grafice

MATLAB conține numeroase comenzi pentru generarea graficelor 2D și 3D.

<i>plot, subplot</i>	Reprezentare funcții continue
<i>stem</i>	Reprezentare funcții discrete
<i>hist</i>	Histograma

Există numeroase comenzi pentru personalizarea graficelor prin titluri, etichete pe axe, etc. Dintre acestea amintim: *xlabel, ylabel, title, grid, text, axis, figure, hold on, close(n)*.

**Example:** Să se scrie două funcții pentru generarea unui semnal sinusoidal; una utilizând bucle *for* și alta utilizând vectori. Să se afișeze timpul de execuție pentru ambele funcții utilizând comenzile *tic, toc*.

*semnal\_sinusoidal.m*

```
function x = semnal_sinusoidal
```

```
%
```

```
%generarea unui semnal sinusoidal utilizând bucle for
```

```
%
```

```
for i = 0 : 9999
```

```
    x(i + 1) = sin(2 * pi * i / 20);
```

```
end
```

*semnal\_sinusoidal2.m*

```
function x = semnal_sinusoidal2
```

```
%
```

```
%generarea unui semnal sinusoidal utilizând vectori
```

```
%
```

```
x = sin(2 * pi * [0 : 9999] / 20);
```

*test.m*

```
help semnal_sinusoidal
```

```
tic
```

```
semnal_sinusoidal;
```

```
toc
```

```
help semnal_sinusoidal2
```

```
tic
```

```
semnal_sinusoidal2;
```

```
toc
```

**Exerciții:**

1. Scrieți funcții pentru afișarea următoarelor semnale:

$$y = \begin{cases} -t, & t \in [-4, 0) \\ t, & t \in [0, 4] \end{cases} \quad (1)$$

$$y = \begin{cases} \sin(t), & \{-10 \leq t < -4\} \cup \{4 \leq t \leq 10\} \\ \cos(5t), & -4 \leq t < 4 \end{cases} \quad (2)$$

$$y = 0.5^t, t \in [0, 10] \quad (3)$$

2. Scrieți o funcție pentru calculul valorii factorialului unui număr  $n$ ,  $(n!)$ .
3. Considerăm un model pentru un circuit electric care conține un rezistor, un inductor, o sursă de tensiune continuă și un întrerupător (Figura ??).

Dacă la momentul  $t = 0$  se închide întrerupătorul (anterior acestui moment curentul prin inductor este nul), avem ecuația 4,

$$L \frac{di}{dt} + Ri = V \quad (4)$$

- Arătați că soluția acestei ecuații este:

$$i = \frac{V}{R} [1 - \exp\{-Rt/L\}]$$

- Creați variabilele Matlab  $V, R, L$  cu valorile  $R = 100\Omega$ ,  $V = 3.3V$ ,  $L = 33mH$ .
- Scrieți o funcție Matlab pentru calculul valorii curentului  $i(t)$  având ca argumente momentul de timp ( $t$ ), valoarea tensiunii sursei ( $V$ ), valoarea inductanței ( $L$ ) și valoarea rezistenței ( $R$ )
- Reprezentați grafic această soluție pe intervalul de timp  $[0, 5ms]$