

---

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Introducere in Matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% (1) Notiuni de baza

% Matlab este un pachet de programe dedicat calculului numeric si reprezen-
% tarilor grafice in domeniul stiintei si ingineriei. El integreaza analiza
% numerica, calculul matriceal, procesarea semnalelor si reprezentarile
% grafice, intr-un mediu usor de invatat si de folosit.

% Matlab include aplicatii specifice, numite TOOLBOX-uri. Acestea sunt
% colectii extinse de functii Matlab (fisiere M) care dezvoltă mediul de
% programare de la o versiune la alta, pentru a rezolva probleme din dome-
% nii variate.

% Matlab lucreaza fie in modul linie de comanda, situatie in care fiecare
% linie este prelucrata imediat si rezultatele sunt afisate, fie cu progra-
% me continute in fisiere. Fisierele sunt discutate la punctul (6).

% In cele ce urmeaza ne vom concentra pe lucrul in linia de comanda. Scrieti
% comenzile Matlab in fereastra de comanda (Command Window). Spatiul de lu-
% cru (Workspace) indica informatii despre variabile.

% Simbolul "%" este folosit pentru a indica inceputul unui comentariu.

% In cazul in care instructiunea Matlab pe care vreti sa o scrieti este prea
% lunga pentru a incapa intr-o singura linie de cod, folositi "..." la
% sfarsitul liniei si continuati pe linia urmatoare. Spre exemplu:

A = [1, 2; ...
     3, 4];

% Folosirea simbolului ";" (punct si virgula) la sfarsitul unei linii de cod
% duce la neafisarea rezultatului evaluarii liniei de cod. Daca ";" e omis
% atunci Matlab va afisa rezultatul. Se poate printa astfel valoarea unei
% variabile, spre exemplu:

A

% Folositi tastele sus/jos pentru a naviga in istoria de comenzi.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% (2) Tipuri de baza in Matlab

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (A) Tipul de baza in Matlab este matricea (elementele matricei sunt de re-
% gula reprezentate in virgula mobila cu dubla precizie). Scalarii sunt
% exemple de matrice 1x1 (o linie, o coloana), iar vectorii sunt matrice 1xn
% (vectori coloana cu o linie si n coloane) sau nx1 (vectori linie cu n
% linii si o coloana). Spre exemplu:

```

---

---

```

A = [1 2; 3 4]; % A este o matrice 2x2
B = [1,2; 3,4]; % cel mai simplu mod de a crea o matrice este de a-i lista
                  % elementele intre paranteze patrate.
                  % simbolul ";" separa coloanele.

N = 5             % un scalar
v = [1 0 0]       % un vector linie
v = [1; 2; 3]     % un vector coloana
v = v'            % transpusul unui vector (linie -> coloana sau invers)
v = 1:.5:3         % un vector cu elemente dintr-un interval specificat:
v = pi*[-4:4]/4    % [primul element:pas:ultimul element]
v = []            % vector vid

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (B) Crearea de matrice speciale: primul parametru - nr de linii, al doilea
% parametru - nr de coloane
%
m = zeros(2, 3) % creeaza o matrice 2x3 de 0-uri
v = ones(1, 3) % creeaza o matrice 1x3 (vector linie) de 1-uri
m = eye(3)     % creeaza matricea identica (3x3)
v = rand(3, 1) % creeaza o matrice 3x1 (vector coloana) cu elemente alea-
                  % toare din intervalul [0 1]
                  % uitati-va si la comanda randn => "help randn"

m = zeros(3)    % creeaza o matrice 3x3 (!) de 0-uri

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (C) Indexarea vectorilor si a matricelor
% Atentie: indicii intotdeauna incep de la 1 si *NU* de la 0!

v = [1 2 3];
v(3)             % acceseaza al treilea element al vectorului v

m = [1 2 3 4; 5 7 8 8; 9 10 11 12; 13 14 15 16]
m(1, 3)          % acceseaza un element al matricei
                  % matrice(#linie,#coloana); #linie = numar linie

m(2, :)          % acceseaza o intreaga linie (a doua) a matricei
m(:, 1)          % acceseaza o intreaga coloana (prima) a matricei

m(1, 1:3)        % acceseaza primele 3 elemente din linia 1
m(2:3, 2)        % acceseaza elementele 2 si 3 ale coloanei a doua.

m(2:end, 3)      % cuvantul cheie "end" acceseaza sfarsitul unei linii sau
                  % coloane

m = [1 2 3; 4 5 6]
size(m)          % returneaza dimensiunile matricei
size(m, 1)       % #linii
size(m, 2)       % #coloane
m1 = zeros(size(m)) % creeaza o matrice noua de dimensiunile lui m cu 0-uri.

whos             % listeaza variabilele si detalii despre ele din Workspace

```

---

---

```

who                % la fel ca "who" dar fara detalii

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% (3) Operatii simple cu vectori si matrice

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (A) Operatii "element-cu-element":
% Daca doua matrice se aduna, scad, inmultesc, impart "element-cu-element",
% ele trebuie sa aiba aceleasi dimensiuni.

a = [1 2 3 4]'; % un vector coloana
2 * a           % multiplicare cu scalar
a / 4           % impartire cu scalar
b = [5 6 7 8]'; % alt vector coloana
a + b           % adunare de vectori
a - b           % scadere de vectori
a .^ 2          % ridicare la patrat "element-cu-element" (observati simbo-
                % lul ".")
a .* b          % multiplicare "element-cu-element" (observati simbolul ".")
a ./ b          % impartire "element-cu-element" (observati simbolul ".")
log([1 2 3 4]) % logaritmare "element-cu-element"
round([1.5 2; 2.2 3.1]) % rotunjire "element-cu-element" la cel mai apropiat
                        % intreg

% Alte operatii aritmetice "element-cu-element" includ:
% floor (parte intreaga inferioara), ceil (parte intreaga superioara)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (B) Operatii vectoriale
% Functii Built-in in Matlab care opereaza pe vectori
a = [1 4 6 3]    % vector linie
sum(a)           % suma elementelor vectorului
mean(a)          % media elementelor vectorului
var(a)           % varianta elementelor
std(a)           % deviatia standard
max(a)           % maximul
min(a)           % minimul

% Pentru o matrice, functiile de mai sus opereaza pe fiecare coloana a
% matricei si vor returna un vector linie

a = [1 2 3; 4 5 6] % o matrice
mean(a)            % media fiecărei coloane
max(a)             % maximul fiecărei coloane
max(max(a))        % maximumul tuturor elementelor matricei
mean(a, 2)         % media fiecărei linii (al doilea argument specifica
                  % dimensiunea de-a lungul careia se efectueaza operatia)

[1 2 3] * [4 5 6]' % vector linie 1x3 inmultit cu un vector coloana 3x1
                  % rezultatul este un scalar (aceasta inmultire se mai
                  % numeste "produs scalar"). Observati absenta simbolului
                  % "."

```

---

---

```

[1 2 3]' * [4 5 6] % vector coloana 3x1 inmultit cu un vector linie 3x1
                  % rezultatul este o matrice. Observati absenta simbolului
                  % "."

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (C) Operatii matriceale

a = rand(3,2) % matrice 3x2
b = rand(2,4) % matrice 2x4
c = a * b      % produs de matrice -> matrice 3x4
a = [1 2; 3 4; 5 6]; % matrice 3x2
b = [5 6 7];      % vector linie 1x3
b * a              % produs vector-matrice => vector linie 1x2
c = [8; 9];        % vector coloana 2x1
a * c              % produs matrice-vector => vector coloana 3x1
A = [1 3 2; 6 5 4; 7 8 9]; % matrice 3x3
inv(A)             % inversa matricei
eig(A)             % vector cu valori proprii ale lui A
[V, D] = eig(A)    % D - matrice cu valori proprii pe diagonala
                  % V - matrice cu vectori proprii pe coloana
                  % A*V = V*D
                  % exemplu de functie care intoarce mai multe valori

% Alte operatii matriceale: det, norm, rank, ...

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(D) Remodelarea (reshaping) matricelor

a = [1 2; 3 4; 5 6]; % matrice 3x2
b = a(:)              % obtinem un vector coloana 6x1 prin concatenarea
                  % coloanelor

sum(a(:))             % mod eficient de a afla suma elementelor lui a

a = reshape(b, 2, 3) % obtinem o matrice 2x3 din vectorul coloana b

a = [1 2]; b = [3 4]; % doi vectori linie
c = [a b]             % concatenare orizontala (help "horzcat")

a = [1; 2; 3];        % vector coloana
c = [a; 4]            % concatenare verticala (help "vertcat")

a = [eye(3) rand(3)] % concatenare pentru matrice
b = [eye(3); ones(1, 3)]

b = repmat(5, 3, 2) % creeaza o matrice 3x2 cu toate elementele = 5
b = repmat([1 2; 3 4], 3, 2) % copiaza matricea 2x2 de 3 ori pe linie, de 2
                  % ori pe coloana; obtine o matrice 6x4

b = diag([1 2 3])     % creeaza o matrice diagonala 3x3 cu elementele pe
                  % diagonala

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

---

---

```

% (4) Instructiuni de control & vectorizare

% Sintaxa pentru instructiunile de control:
%
% for VARIABILA = VECTOR
%     instructiune
%     ...
%     instructiune
% end
%
% VECTOR este un vector, spre exemplu 1:10 sau -1:0.5:1 sau [1 4 7]
%
%
% while EXPRESIE
%     instructiuni
% end
%
% if EXPRESIE
%     instructiuni
% elseif EXPRESIE
%     instructiuni
% else
%     instructiuni
% end
%
% (clauzele "elseif" si "else" sunt optionale, "end" este obligatoriu)
%
% Exemple pt EXPRESIE: a < b, a<=b, a==b, a~= b
%
% Observatie:
% Instructiunile repetitive ruleaza foarte incet in Matlab, interpretorul
% propriu este optimizat pentru operatii matriceale. Incercati pe cat posi-
% bil sa "vectorizati" calculele, scriind codul orientat pe operatii matri-
% ceale. Exemple de optimizare sunt prezentate mai jos.

%Exemplul 1
for i=1:2:7 % repeta pentru 1 pana la 7 din 2 in 2
    i % afiseaza i
end

%Exemplul 2
for i=[5 13 -1] % repeta pentru valori ale unui vector
    if (i > 10) % exemplu de instructiune if
        disp('Larger than 10') % afiseaza un string
    elseif i < 0 % parentezele sunt optionale
        disp('Negative value')
    else
        disp('Something else')
    end
end

% Exemplul 3
% Pentru o matrice A de dimensiuni mxn si un vector v linie 1xn, vrem sa
% scadem v din fiecare linie din A.

```

---

---

```

m = 50; n = 10; A = ones(m, n); v = 2 * rand(1, n);
%
% Solutie folosind instructiuni repetitive:
for i=1:m
    A(i,:) = A(i,:) - v;
end
% Solutie folosind numai operatii matriceale
A = ones(m, n) - repmat(v, m, 1); % mult mai rapid

% Exemplul 4
% Data fiind o matrice A de dimensiuni mxn, creati o matrice B de aceleasi
% dimensiuni cu A cu toate elementele = 0, apoi copiat in matricea B toate
% elementele matricei A > 0.
%
% Solutie folosind instructiuni repetitive:
B = zeros(m,n);
for i=1:m
    for j=1:n
        if A(i,j)>0
            B(i,j) = A(i,j);
        end
    end
end
% Solutie eficienta, fara instructiuni repetitive
B = zeros(m,n);
ind = find(A > 0); % gaseste indicii elementelor > 0 din A
                  % pentru mai multe informatii tastati "help find"
B(ind) = A(ind);  % copiaza in B numai elementele > 0 din A

%Exemplul 5
% Data fiind o matrice A de dimensiuni 10000x10000, calculati suma tuturor
% elementelor din A
%
%Solutie folosind instructiuni repetitive:
A = rand(10000,10000);
tic          % initializare ceas intern
suma = 0;
for i = 1:size(A,1)
    for j = 1:size(A,2)
        suma = suma + A(i,j);
    end
end
toc          % stop ceas intern
%Solutie eficienta
tic, suma = sum(sum(A(:)));toc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%(5) Salvarea variabilelor

save myfile % salveaza toate variabilele din workspace in myfile.mat

save myfile a b % salveaza numai variabilele a si b in fisierul myfile.mat

```

---

---

```

clear a b % inlatura variabilele a si b din workspace

clear % inlatura toate variabilele din workspace

load myfile % incarca variabilele din fisierul myfile.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%(6) Crearea script-urilor si a functiilor folosind fisiere .m

% In Matlab, script-urile sunt fisiere cu extensie .m ce contin comenzi Mat-
% lab. Variabilele declarate intr-un script sunt globale. Eventualele modi-
% ficari ale valorilor acestor variabile globale actioneaza si asupra vari-
% abilelor cu acelasi nume din sesiunea Matlab curenta. Un script cu numele
% "script1.m" poate fi invocat din linia de comanda tastand "script1".

% Functiile sunt tot fisiere .m. Prima linie a unei functii trebuie sa fie
% de forma:
%
% function [outarg_1, ..., outarg_m] = myfunction(inarg_1, ..., inarg_n)
% unde
% inarg_1, ..., inarg_n - argumente de intrare
% outarg_1, ..., outarg_n - argumente de iesire
%
% Numele functiei e bine sa fie identic cu numele fisierului (i.e. functia
% "myfunction" ar trebui salvata in fisierul "myfunction.m").
% Mai jos gasiti doua exemple de functii: myfunction.m si myotherfunction.m
%
% Exemplul 1 de functie
% Deschideti un fisier nou (File->New->Function)
function y = myfunction(x)
% Functie cu un singur argument de intrare si o singura valoare returnata
a = [-2 -1 0 1]; % variabila a este locala functiei
y = a + x;
% Salvati functia in fisierul cu numele myfunction.m
%
% Exemplul 2 de functie
% Deschideti un fisier nou (File->New->Function)
function [y, z] = myotherfunction(a, b)
% Functie cu 2 argumente de intrare si doua valori returnate
y = a + b;
z = a - b;
% Salvati functia in fisierul cu numele myfunction.m
%
% Functiile sunt executate folosind un workspace local, nu exista riscul de
% conflicte cu variabilele din workspace-ul principal. La sfarsitul executa-
% rii unei functii numai argumentele returnate sunt vizibile in workspace-ul
% principal.
a = [1 2 3 4]; % a este o variabila globala
b = myfunction(2 * a) % call myfunction care are variabila locala a
a % variabila globala a este neschimbata
[c, d] = ...
myotherfunction(a, b)% call myotherfunction

```

---

---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### ``` %(7) Plotare ```

```
x = [0 1 2 3 4]; %  
plot(x);          % ploteaza x (pe axa Oy) versus indecsii lui x (pe axa Ox)  
pause             % pauza - se asteapta pana la apasarea unei taste  
plot(x, 2*x);     % ploteaza 2*x (pe axa Oy) versus x (pe axa Ox)  
axis([0 8 0 8]); % ajusteaza axele figurii
```

```
figure;           % deschide o figura noua  
x = pi*[-24:24]/24;  
plot(x, sin(x));  
xlabel('radiani'); % asigneaza un label (eticheta) pentru axa Ox  
ylabel('valoare sin'); % asigneaza un label pentru axa Oy  
title('incercare'); % titlu  
close all;        % inchide toate figurile deschise  
plot(x, sin(x), 'or'); % ploteaza puncte in forma de o de culoare 'r' - red  
plot(x, sin(x), 'or:'); % ploteaza puncte + linie  
% help 'plot'
```

```
figure;           % vom plota functii diferite in figuri separate  
subplot(1, 2, 1); % vezi ce face functia subplot -> "help subplot"  
plot(x, sin(x));  
axis square;      % axe patrate  
subplot(1, 2, 2);  
plot(x, 2*cos(x));  
axis square;
```

```
figure;           % vom plota functii diferite in aceeaasi figura  
plot(x, sin(x));  
hold on;          % pastreaza graficul din figura  
plot(x, 2*cos(x), '--'); % ploteaza o alta functie cu '--'  
legend('sin', 'cos'); % nume pentru fiecare plot  
hold off;
```

```
figure;           % matrice vs. imagini  
m = rand(64,64);  
imagesc(m)        % ploteaza matricea ca imagine  
colormap gray;    % seteaza colormap = gri, obtine astfel intensitati  
% "help colormap"  
axis image;       % arata coordatele pixelilor ca axe  
axis off;         % inlatura axele  
colormap jet;     % colormap implicit  
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### ``` %(8) Lucrul cu imagini si cu toolbox-ul de Image Processing ```

```
I = imread('football.jpg'); % citeste o imagine JPEG in I
```

```
figure  
imshow(I)          % arata I
```



---

```

impixelinfo                                % afiseaza valorile pixelilor

whos I                                     % I - imagine RGB (3 canale)
                                           % R - red, G - green, B -blue

I_red = I;
I_red(:,:,2) = 0;
I_red(:,:,3) = 0;
figure
imshow(I_red)                             % afiseaza canalul rosu din I
impixelinfo

I_green = I;
I_green(:,:,1) = 0;
I_green(:,:,3) = 0;
figure
imshow(I_green)                           % afiseaza canalul verde din I
impixelinfo

I_blue = I;
I_blue(:,:,1) = 0;
I_blue(:,:,2) = 0;
figure
imshow(I_blue)                             % afiseaza canalul albastru din I
impixelinfo

rgbI = I_red + I_green + I_blue;
isequal(rgbI,I)                            % rgbI si I sunt la fel?

figure(1)                                  % inapoi la figura 1
d = getrect;                               % selecteaza un dreptunghi in imaginea I
I2 = imcrop(I,d);                          % crop I dupa d si obtine I2
I2 = rgb2gray(I2);                         % converteste I2 din rgb in intensitate
imagesc(I2)                                % scaleaza datele pentru a utiliza intreg
                                           % colormap

colormap('gray')                           % colormap gray -> intensitate
colorbar                                   % activeaza bara de culori

true_size                                  % afiseaza imaginea la rezolutia de
                                           % 1 pixel pe ecran : 1 pixel de imagine

true_size(3*size(I2))                      % schimbam rezolutia, 9:1

close all;
I3 = imresize(I2,0.5,'bil'); % redimensionare la 50% folosind
figure(1),imshow(I3)           % interpolarea biliniara

I4 = imrotate(I2,45,'bil'); % roteste I2 cu 45 de grade
figure(2),imshow(I4)

I5 = imrotate(I2,45,'bil','crop'); % roteste I2 cu 45 de grade
figure(3),imshow(I5)           % crop I5 la dimensiunea lui I2

close all

```

---

---

```
I3 = double(I2);           % converteste din uint8 in double,
                           % pentru a putea efectua operatii

imagesc(I3.^2)
imagesc(log(I3))

imwrite(I4,'imagea4.jpg','jpg');%scrie I4 intr-un fisier, "help imwrite"
info1 = imfinfo('imagea4.jpg')  %"help imfinfo"
I6 = imread('imagea4.jpg');
isequal(I4,I6)
imwrite(I4,'imagea4_lossless.jpg','jpg','mode','lossless');
info2 = imfinfo('imagea4_lossless.jpg')
I6 = imread('imagea4_lossless.jpg');
isequal(I4,I6)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

*Published with MATLAB® 7.13*