

Laborator 1: Introducere în MATLAB

Cuprins

Link-uri utile.....	1
Considerații generale.....	1
Noțiuni de bază.....	1
Scalari, vectori și matrici.....	3
Variabilele din spațiul de lucru (workspace).....	7
Programare și scripturi.....	8
Scripts.....	8
Live Scripts.....	9
Instrucțiuni de control logic.....	9
Funcții.....	10
Help și documentații.....	10
Desfășurarea lucrării.....	12
Exemplul 1. Formate pentru afișare.....	12
Exemplul 2. Operații cu matrici.....	13
Exemplul 3. Vector cu pas liniar.....	14
Exemplul 4. Vector cu pas logaritm.....	15
Exemplul 5. Vector cu pas liniar.....	15
Exemplul 6. Vector cu pas logaritm.....	16
Exemplul 7. Lungime vector.....	16
Exemplul 8. Matrice cu valori '1' sau '0'.....	16
Exemplul 9. Afișare grafică.....	17
Exemplul 10. Funcție.....	18
Exerciții.....	19

Link-uri utile

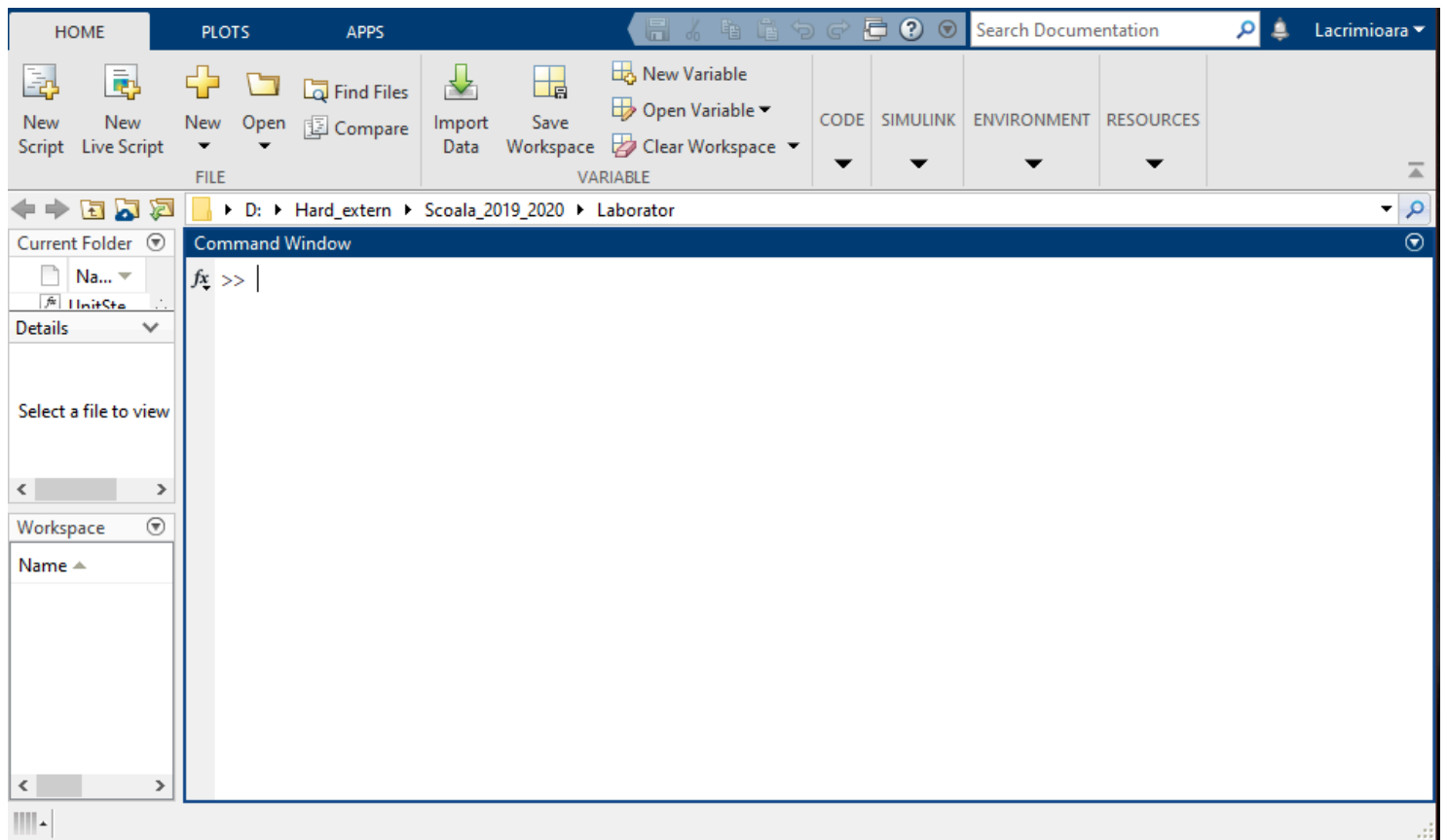
1. MATLAB Primer https://www.mathworks.com/help/pdf_doc/matlab/learn_matlab.pdf
2. MATLAB help and User's guide <https://www.mathworks.com/help/matlab/index.html>
3. MATLAB Central (script, toolbox, blog, newsgroup) <https://www.mathworks.com/matlabcentral/>
4. MATLAB Newsletters <https://www.mathworks.com/company/newsletters.html>

Considerații generale

Obiectiv: prezentarea unor comenzi MATLAB utile în prelucrarea digitală a semnalelor.

Noțiuni de bază

După lansarea în execuție, programul MATLAB intră în modul de comandă, afișând promptul '>>', și așteaptă introducerea unei comenzi de la utilizator.



De exemplu, comanda '>>var = 0:5' generează variabila 'var', afișând în fereastra de comandă cele 6 elemente ale variabilei 'var', de la 'var(1) = 0' la 'var(6) = 5'.

```
Command Window
>> var = 0:5

var =

    0     1     2     3     4     5
```

Dacă nu se specifică o variabilă de ieșire, MATLAB-ul folosește variabila 'ans' (de la *answer*) pentru a afișa rezultatul calculului.

```
Command Window
>> cos(0.5)

ans =

    0.8776
```

Dacă la finalul liniei se pune punct și virgulă ';', MATLAB-ul realizează calculul, dar nu afișează nimic în fereastra de comandă.

```
Command Window
>> a = var(2)+5;
```

Comenzile introduse anterior pot fi corectate sau reutilizate; prin utilizarea săgeților de pe tastatură apar în linia de comandă comenzile introduse anterior; se pot realiza corecțiile necesare iar, apoi, prin apăsarea tastei [Enter] se rulează comanda corectată.

Scalar, vectori și matrici

Scalarii sunt asimilați matricilor cu o linie și o coloană, iar vectorii sunt asimilați matricilor cu o linie sau o coloană.

- Pentru a genera un vector linie, se separă elementele fie prin virgulă ',', fie prin spațiu

```
Command Window
>> b = [1 3 5 7]

b =

     1     3     5     7
```

- Pentru a genera o matrice cu mai multe linii, se separă liniile prin punct și virgulă ';'.

```
Command Window
>> A = [1 3 5; 7 9 11]

A =

     1     3     5
     7     9    11
```

- Se pot crea matrici folosind funcții de genul 'ones', 'zeros', 'rand' sau 'randn'

```
Command Window

>> c = ones(1, 3)

c =

    1    1    1

>> d = zeros(3, 1)

d =

    0
    0
    0

>> e = rand(2, 3)

e =

    0.0975    0.5469    0.9649
    0.2785    0.9575    0.1576
```

Există două modalități de referire la un anumit element dintr-o matrice:

- Cel mai uzual mod este de a specifica linia și coloana dorită, de exemplu

```
Command Window

>> A

A =

     1     3     5
     7     9    11

>> A(2, 3)

ans =

    11
```

- Mai puțin obișnuit, dar uneori util, este utilizarea unui singur index care parcurge fiecare coloană, în ordine (utilizarea unui singur index pentru a face referire la un element dintr-o matrice se numește *indexare liniară*):

```
Command Window

>> A(5)

ans =

     5
```

În partea stângă a unei instrucțiuni de atribuire, se pot specifica elemente în afara dimensiunilor actuale ale matricii. Dimensiunea matricii crește, pentru a se acomoda cu noile valori.

```
Command Window

>> A(4, 5) = 20

A =

     1     3     5     0     0
     7     9    11     0     0
     0     0     0     0     0
     0     0     0     0    20
```

Pentru a face referire la mai multe elemente ale unei matrici, se utilizează operatorul două puncte ':', care permite specificarea unui domeniu de forma 'start : end'. De exemplu, dacă se doresc elementele din primele trei rânduri și coloana a doua a matricii A:

```
Command Window

>> A(1:3, 2)

ans =

     3
     9
     0
```

Operatorul două puncte ':', fără valori pentru început și final, specifică toate elementele din acea dimensiune. De exemplu, dacă se dorește selectarea celei de-a doua linii din A:

```
Command Window

>> A(2, :)

ans =

     7     9    11     0     0
```

Operatorul două puncte ':' permite, de asemenea, crearea unui vector cu valori egal distanțate, folosind o sintaxă de forma 'start : step : end'

```
Command Window

>> v = 0:5:20

v =

    0     5    10    15    20
```

- Dacă se omite valoarea din mijloc 'step', pasul va fi considerat 1

```
Command Window

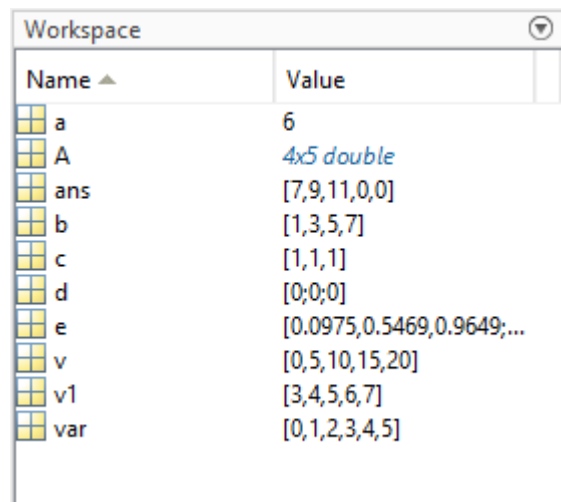
>> v1 = 3:7

v1 =

    3     4     5     6     7
```

Variabilele din spațiul de lucru (*workspace*)

Spațiul de lucru conține toate variabilele create sau importate în MATLAB. De exemplu, toate variabilele create anterior pot fi vizualizate



Name ▲	Value
a	6
A	4x5 double
ans	[7,9,11,0,0]
b	[1,3,5,7]
c	[1,1,1]
d	[0;0;0]
e	[0.0975,0.5469,0.9649;...
v	[0,5,10,15,20]
v1	[3,4,5,6,7]
var	[0,1,2,3,4,5]

- Variabilele din workspace pot fi vizualitate folosind comanda 'whos'

Command Window					
>> whos					
Name	Size	Bytes	Class	Attributes	
A	4x5	160	double		
a	1x1	8	double		
ans	1x5	40	double		
b	1x4	32	double		
c	1x3	24	double		
d	3x1	24	double		
e	2x3	48	double		
v	1x5	40	double		
v1	1x5	40	double		
var	1x6	48	double		

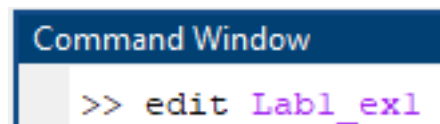
Programare și scripturi

Cel mai simplu tip de program MATLAB se numește script. Un script este un fișier care conține mai multe linii secvențiale de comenzi MATLAB și apelări de funcții. Un script se poate rula tastând numele acestuia în linia de comandă. Scripturile nu acceptă argumente de intrare sau de ieșire. Ele operează cu datele din workspace.

Scripts

Pentru a crea un script

- Se poate folosi comanda 'edit'. Această comandă deschide un fișier gol cu numele 'Lab1_ex1.m'.

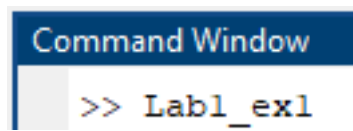


- Sau 'New -> Script'. Se deschide un fișier gol care trebuie salvat 'Ctrl + S' și se introduce numele dorit (extensia este '.m')

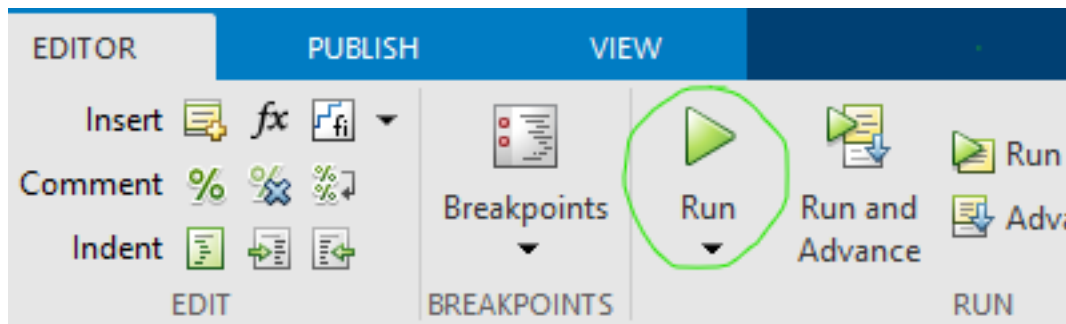
Pe urmă se scriu liniile de cod necesare. Comentariile sunt precedate de simbolul procent '%'.

Pentru a rula scriptul

- Se poate scrie numele fișierului în linia de comandă



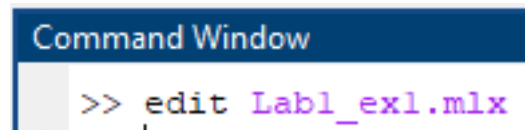
- Sau se poate folosi butonul "Run" din Editor



Live Scripts

Scripturile live permit vizualizarea și interacțiunea atât cu codul, cât și cu ieșirea. Acestea pot include pe lângă cod, text formatat, ecuații și imagini. Pentru a crea un script live

- Se poate folosi comanda 'edit'. Această comandă deschide un fișier gol cu numele 'Lab1_ex1.mlx'



- 'New Live Script'. Se deschide un fișier gol care trebuie salvat 'Ctrl + S' și se introduce numele dorit (extensia este '.mlx')

Instrucțiuni de control logic

În cadrul oricărui script, se pot defini secțiuni de cod care se repetă într-o buclă sau se execută condiționat. Buclele folosesc un cuvânt cheie 'for' sau 'while', iar instrucțiunile condiționale folosesc 'if' sau 'switch'.

Buclele sunt utile pentru crearea secvențelor. De exemplu, pentru a calcula primele 50 de numere ale secvenței lui Fibonacci, se folosește o buclă 'for'. Primele două numerele sunt 1 și fiecare număr ulterior este suma celor două numere anterioare.

```
N = 50;  
f(1) = 1; f(2) = 1;  
  
for i = 3:N  
    f(i) = f(i-1) + f(i-2);  
end
```

```
f(1:15)
```

```
ans = 1x15
```

```
1      1      2      3      5      8     13     21     34     55     89    144    233 ...
```

- La rularea scriptului, instrucțiunea 'for' definește un contor 'i' care începe de la 3. Apoi, bucla se alocă în mod repetat lui 'f(i)', incrementându-l pe 'i' la fiecare execuție, până când ajunge la 50. Ultima comandă 'f(1:15)' afișează primele 15 elemente ale lui 'f'

Instrucțiunile condiționale se execută doar atunci când expresiile date sunt adevărate. De exemplu, putem asigura unei variabile anumite "nume" în funcție de valoarea unui număr: 'mic', 'mediu', 'mare'. În acest caz numărul aleator este un număr întreg între 1 și 300.

```
numar = randi(300)
```

```
numar =  
288
```

```
if numar<100  
    out = 'mic'  
elseif numar<200  
    out = 'mediu'  
else  
    out = 'mare'  
end
```

```
out =  
'mare'
```

Funcții

Funcțiile pot accepta argumente de intrare și returnează argumente de ieșire. Variabilele interne sunt variabile locale pentru funcție. Numele fișierului și al funcției trebuie să fie aceleași.

Prima linie a unei funcții începe cu cuvântul cheie 'function'. Aceasta oferă numele funcției și ordinea argumentelor

```
function [out1, out2, ...] = TestLab1(in1, in2, ...)
```

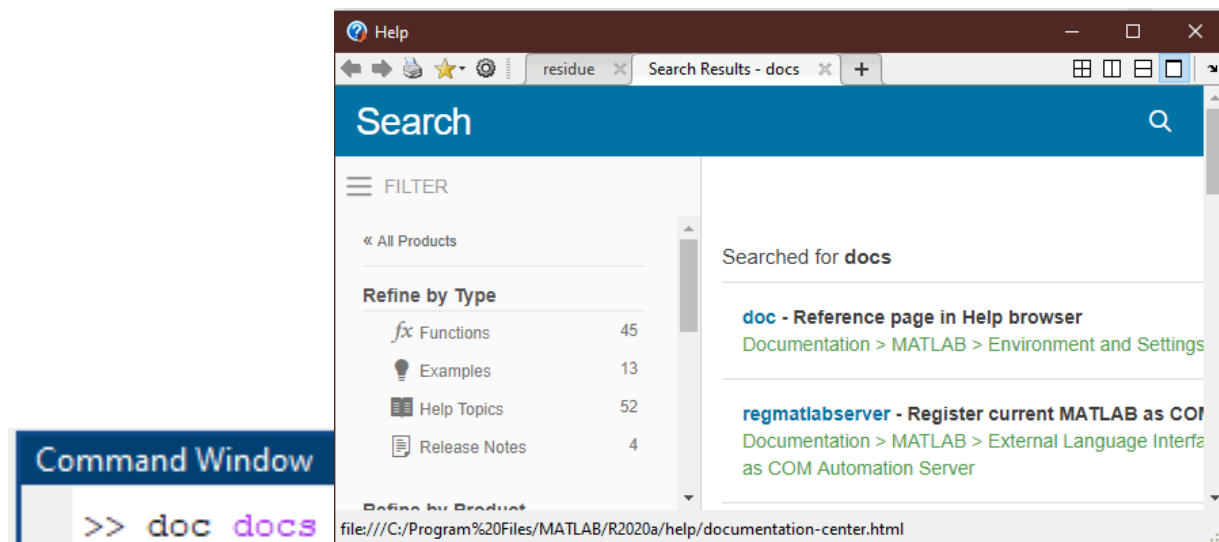
Pentru a crea o funcție

- 'New → Function'
- Sau 'New → Live Function'

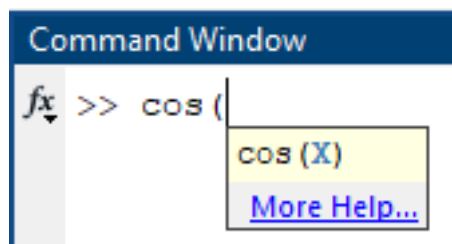
Help și documentații

Toate funcțiile MATLAB au documentații suport ce includ exemple și descriu intrările, ieșirile și modul de apelare al funcțiilor. Sunt mai multe modalități pentru a accesa aceste informații din linia de comandă:

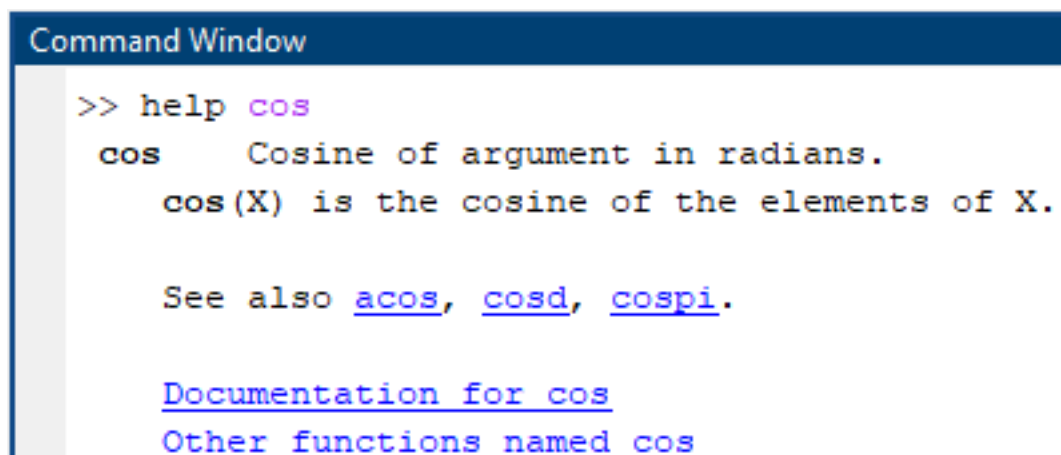
- Deschiderea documentației funcției într-o fereastră separată folosind comanda 'doc'



- Afișarea anumitor indicații despre funcție (sintaxa funcției) în fereastra de comandă scriind numele funcției urmată de paranteză rotundă deschisă '('



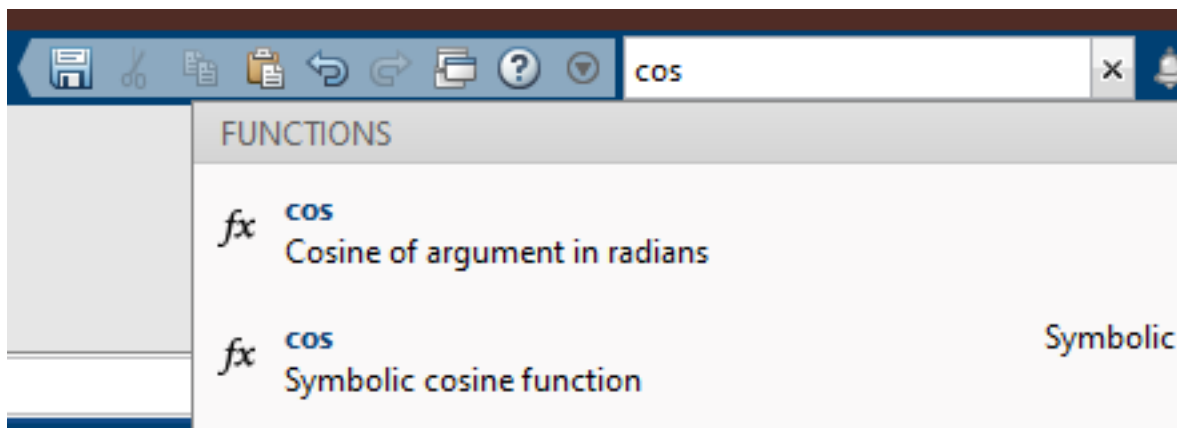
- Vizualizarea unui sumar a documentației funcției în fereastra de comandă folosind comanda 'help'



Se poate vizualiza documentația completă a produsului făcând clic pe pictograma '?'.
?



Dacă se dorește documentația doar pentru o anumită funcție, se poate tasta numele funcției sau un cuvânt cheie în câmpul "Search Documentation", iar apoi se apasă tasta [Enter].



Desfășurarea lucrării

Exemplul 1. Formate pentru afișare

Să se verifice diferite formate de afișare pentru variabila

$$x = \pi$$

```
% Ex1_1  
clear variables;  
x = pi;  
format short e  
x
```

```
x =  
3.1416e+00
```

```
format short  
x
```

```
x = 3.1416
```

```
format long e  
x
```

```
x =  
3.141592653589793e+00
```

```
format long  
x
```

```
x =  
3.141592653589793
```

```
format rat  
x
```

```
x =  
355/113
```

Exemplul 2. Operații cu matrici

Se consideră matricea

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

- Transpusa matricii A

```
% Ex1_2  
clear variables;  
format short;  
A = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
A = 3x4  
     1     2     3     4  
     5     6     7     8  
     9    10    11    12
```

```
B1 = A.'
```

```
B1 = 4x3  
     1     5     9  
     2     6    10  
     3     7    11  
     4     8    12
```

```
B2 = transpose(A)
```

```
B2 = 4x3  
     1     5     9  
     2     6    10  
     3     7    11  
     4     8    12
```

- Dimensiunea matricii A

```
[row, col] = size(A)
```

```
row = 3  
col = 4
```

```
size(A)
```

```
ans = 1x2
```

3 4

- Elementul de pe linia 3, coloana 2 din matricea A

```
A(3, 2)
```

```
ans = 10
```

- Elementele de pe linia 3, coloana 1 si 3 din matricea A

```
A(3, [1, 3])
```

```
ans = 1x2
     9    11
```

- A^2

```
C1 = A.^2
```

```
C1 = 3x4
     1     4     9    16
    25    36    49    64
    81   100   121   144
```

```
C2 = A.*A
```

```
C2 = 3x4
     1     4     9    16
    25    36    49    64
    81   100   121   144
```

```
C3 = power(A, 2)
```

```
C3 = 3x4
     1     4     9    16
    25    36    49    64
    81   100   121   144
```

- $3.5 \cdot A$

```
D = 3.5*A
```

```
D = 3x4
   3.5000   7.0000  10.5000  14.0000
  17.5000  21.0000  24.5000  28.0000
  31.5000  35.0000  38.5000  42.0000
```

Exemplul 3. Vector cu pas liniar

Să se genereze un vector cu pas liniar, între -8 și 8, cu pas 2.

```
% Ex1_3
clear variables;
```

```
v1 = -8:2:8
```

```
v1 = 1×9  
    -8    -6    -4    -2     0     2     4     6     8
```

- $$\text{nr. term.} = \frac{\text{val. fin.} - \text{val. init.}}{\text{pas}} + 1$$

```
v2 = linspace(-8, 8, (8-(-8))/2+1)
```

```
v2 = 1×9  
    -8    -6    -4    -2     0     2     4     6     8
```

Exemplul 4. Vector cu pas logaritmic

Să se genereze un vector cu 5 elemente distribuite logaritmic între

10^{-4} și 10^4

```
% Ex1_4  
clear variables;  
format long e;  
y = logspace(-4, 4, 5)
```

```
y = 1×5  
    1.000000000000000e-04    1.000000000000000e-02    1.000000000000000e+00 ...
```

Exemplul 5. Vector cu pas liniar

Să se genereze un vector cu 5 elemente distribuite liniar între

$-\pi$ și π

```
% Ex1_5  
clear variables;  
p1 = linspace(-pi, pi, 5)
```

```
p1 = 1×5  
   -3.141592653589793e+00   -1.570796326794897e+00           0 ...
```

- $$\text{pas} = \frac{\text{val. fin.} - \text{val. init.}}{\text{nr. term.} - 1}$$

```
p2 = -pi:(pi-(-pi))/(5-1):pi
```

```
p2 = 1×5  
   -3.141592653589793e+00   -1.570796326794897e+00           0 ...
```

Să se genereze un vector cu 5 elemente distribuite liogaritmnic între

```
% Ex1_6
clear variables;
x = logspace(-pi, pi, 5)
```

Să se genereze un vector coloană de lungime 50, cu toate elementele egale cu 2.

```
% Ex1_7
clear variables;
b = 2*ones(50, 1)
```

- Lungimea vectorului

```
L = length(b)
```

$$L = 50$$

Să se genereze o matrice cu 5 linii și 7 coloane, cu toate elementele egale cu 0, și o matrice cu 4 linii și 2 coloane cu toate elementele egale cu 1.

```
% Ex1_8
clear variables;
Z = zeros(5, 7)
```

$$Z = \begin{bmatrix} 5 & 7 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

```
U = ones(4, 2)
```

```
U = 4x2
```

1	1
1	1
1	1
1	1

Exemplul 9. Afișare grafică

Să se reprezinte grafic funcția

$$f(t) = \sin(2\pi 0.2t)$$

cu linie continuă, și funcția

$$g(t) = 1 - f(t)$$

cu linie întreruptă. Să se eticheteze axa-x 'Timp', iar axa-y 'Amplitudine'; titlul graficului trebuie să fie

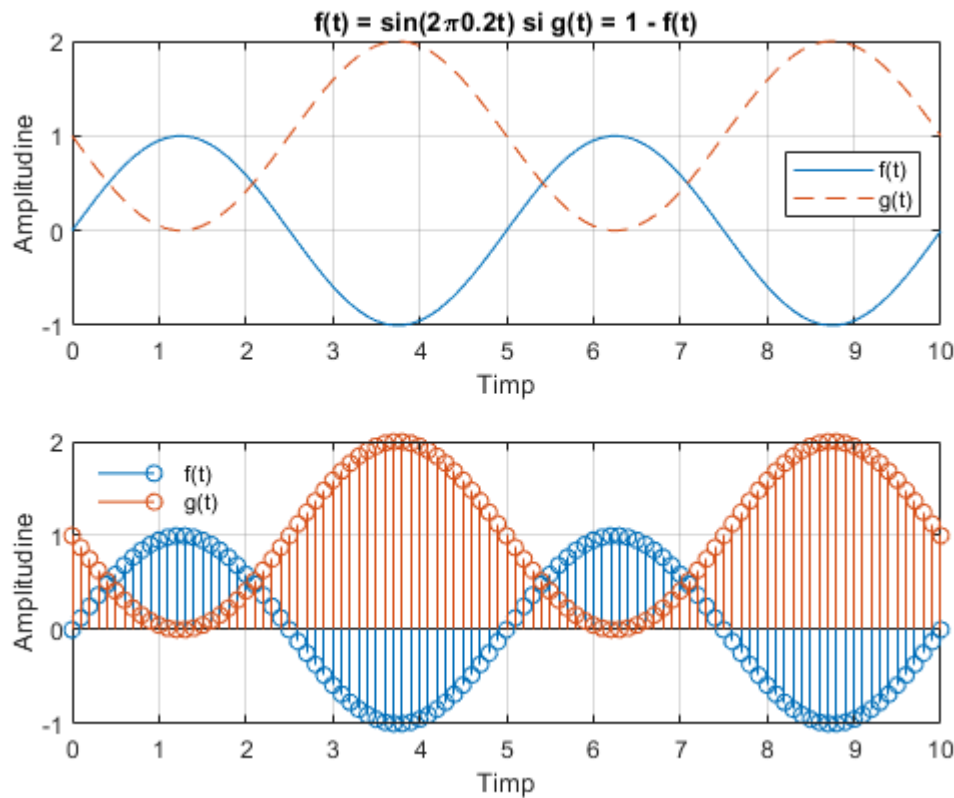
$$'f(t) = \sin(2\pi 0.2t) \text{ si } g(t) = 1 - f(t)'$$

Pentru reprezentarea grafică să se utilizeze comanda 'plot' în subfigura 1 și comanda 'stem' în subfigura 2. Fiecare subfereastră trebuie să conțină o legendă a graficelor reprezentate.

```
% Ex1_9
clear variables;
% generare functii
t = 0:0.1:10;
f = sin(2*pi*0.2*t);
g = 1 - f;

% afisare grafica
figure,
subplot(211),
plot(t, f), hold all,
plot(t, g, '--'), hold off; grid,
xlabel('Timp'), ylabel('Amplitudine'), title('f(t) = sin(2\pi 0.2t) si g(t) = 1 - f(t)'),
legend('f(t)', 'g(t)', 'Location', 'best')

subplot(212),
stem(t, f), hold all,
stem(t, g), hold off; grid,
xlabel('Timp'), ylabel('Amplitudine'),
legend('f(t)', 'g(t)', 'Location', 'best'), legend boxoff
```



Exemplul 10. Funcție

Să se scrie o funcție MATLAB 'medieAritm' care să realizeze media aritmetică a elementelor unui vector. Funcția trebuie să returneze valoarea mediei aritmetice și numărul de elemente ale vectorului.

- Se va crea o funcție de tip live 'medieAritm.mlx' în folderul curent, de forma

```
function [ma, Lx] = medieAritm(x)
    Lx = length(x);
    ma = sum(x)/Lx;
    disp(['Media aritmetica a elementelor vectorului x =' num2str(x) ' este ' num2str(ma)]);
    disp(['Vectorul x =' num2str(x) ' are ' num2str(Lx) ' elemente']);
end
```

- Funcția se apelează astfel

```
% Ex1_10
clear variables;
v = 2:2:10;
medieAritm(v);
```

```
Media aritmetica a elementelor vectorului x = [2  4  6  8 10] este 6
Vectorul x = [2  4  6  8 10] are 5 elemente
```

Exerciții

1) Să se genereze un vector cu pas liniar, între 3 și 9, cu increment 2, și să se afle numărul de elemente.

```
%% Ex. 1  
clear variables;
```

2) Să se genereze un vector cu pas liniar, între 3 și 9, cu 13 elemente.

```
%% Ex. 2  
clear variables;
```

3) Să se genereze un vector cu 9 elemente distribuite logaritmic între

$$10^{-3} \text{ și } 10^3$$

```
%% Ex. 3  
clear variables;
```

4) Se consideră vectorul $\mathbf{y} = 3 : 0.9 : 123$. Să se genereze un nou vector cu toate elementele 1, cu lungimea egală cu cea a vectorului \mathbf{y} .

```
%% Ex. 4  
clear variables;
```

5) Se consideră scalarul $m = 5$ și matricile

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 1 \\ 8 & 4 & 5 \\ 0 & 2 & 0 \end{bmatrix} \text{ și } \mathbf{B} = \begin{bmatrix} 2 & 3 & 4 \\ 1 & 1 & 1 \\ 2 & 3 & 2 \end{bmatrix}$$

Să se evalueze:

$$\begin{aligned} \mathbf{C} &= \mathbf{A} + \mathbf{B}; & \mathbf{D} &= \mathbf{A} - \mathbf{B}; & \mathbf{E} &= m + \mathbf{C}; & \mathbf{F} &= \mathbf{A} \cdot \mathbf{B}; & \mathbf{G} &= m \cdot \mathbf{B}; \\ \mathbf{H} &= \mathbf{A}^t; & \mathbf{I} &= \mathbf{B}^t; & \mathbf{J} &= \mathbf{C}^m; & \mathbf{K} &= \frac{\mathbf{A}}{\mathbf{B}}; & \mathbf{L} &= \mathbf{A}^{-1} \cdot \mathbf{B} \end{aligned}$$

Să se verifice dacă \mathbf{K} și \mathbf{L} sunt egale sau nu.

```
%% Ex. 5  
clear variables;
```

6) Să se evalueze produsul scalar al vectorilor $\mathbf{a} = [1 \ 2 \ 3]$ și $\mathbf{b} = [10 \ 20 \ 30]$.

- Indicație: rezultatul trebuie să aibă o singură valoare (puteți folosi, de exemplu, 'dot(a, b)')

```
%% Ex. 6  
clear variables;
```

7) Folosind comanda 'stem', să se reprezinte grafic secvența

$$x(n) = \cos\left(2\pi \frac{1}{10}n + \frac{\pi}{4}\right), n = \overline{0, 49}$$

```
%% Ex. 7  
clear variables;
```

8) Să se scrie o funcție MATLAB care să realizeze suma a doi vectori și apoi să se apeleze funcția creată.

```
%% Ex. 8  
clear variables;
```

9) Să se scrie o funcție MATLAB care să realizeze media geometrică a elementelor unui vector și apoi să se apeleze funcția creată.

```
%% Ex. 9  
clear variables;
```

10) Să se genereze o secvență aleatoare, cu distribuție uniformă între 0 și 10. Să se reprezinte secvența pentru variabila timp între 0 și 49 (cu pas unitar).

- Indicație: Pentru a se genera o secvență aleatoare cu distribuție uniformă într-un interval specificat $[a; b]$, ieșirea funcției 'rand' trebuie multiplicată cu $(b - a)$, și apoi se adună a . În cazul de față $a = 0$ și $b = 10$.

```
%% Ex. 10  
clear variables;
```

11) Să se genereze o secvență aleatoare, cu distribuție normală (Gaussiană) între 0 și 10. pentru variabila timp între 0 și 49 (cu pas unitar).

- Indicație: Această secvență are media 5 și varianța tot 5. Pentru a se genera o secvență cu distribuție normală, cu acești parametri, se multiplică ieșirea funcției 'randn' cu valoarea deviației standard (radical din 5), iar apoi se adaugă valoarea medie dorite (5).

```
%% Ex. 11  
clear variables;
```

COPYRIGHT NOTICE: This tutorial is intended for the use of students at Faculty of Electronics, Telecommunications and Information Technology from Technical University of Cluj-Napoca. You are welcome to use the tutorial for your own self-study, but please seek the author's permission before using it for other purposes.

