

## I. INTRODUCERE ÎN MATLAB

1. Prezentare Matlab.
2. Vectori și matrice. Definirea lor. Extragere element. Operatii. Tablouri. Generare vectori.
3. Funcții. Reprezentare grafică în sistemul cartezian.
4. Câteva elemente de programare
  - if...elseif...else...end
  - bucla for
  - bucla while
5. Calcul simbolic

### 1. Prezentare Matlab

Matlab este un limbaj puternic de calcul tehnic. Numele provine de la LABoratorul de MATrice, pentru că, elementele de bază cu care lucrează sunt matricele. Pe lângă funcțiile de bază pe care le are Matlab cu care se pot rezolva probleme uzuale de matematică și altele Matlab conține niște pachete de programe numite toolboxes, specializate pe anumite domenii. Fereastra de comenzi (Command window) este fereastra principală unde se pot introduce variabile, evalua expresii, etc. și rula programe.

Fereastra de editare (Editor window) este fereastra unde se pot edita și verifica de erori fișierele script și function.

Fereastra Help conține help-ul Matlab

Dosarul curent (Current folder) conține toate fișierele din dosarul curent.

Command History (Istoricul comenzilor) înmagazinează comenzile utilizate în fereastra de comenzi.

Fereastra workspace furnizează date despre variabilele curente

Editor de variabile (Variable editor) fereastra în care se pot edita, modifica valoarea variabilelor.

Fereastra de editare figuri (Figures window) formatează figurile

Câteva comenzi utile:

Rularea și salvarea fișierului script se va efectua cu ajutorul tastei **F5** sau din meniul **Debug/Save and Run** sau apăsând butonul săgeată verde.

Dacă lipsesc anumite ferestre ele se pot deschide din meniul Desktop. Cele mai utilizate se pot afișa astfel: **Desktop/Desktop Layout/Default.**

## 2. Vectori și matrice. Definirea lor. Extragere element. Operatii. Tablouri.

### Generare vectori

Pentru început, toate calculele se vor face în fereastra de comenzi.

- Definirea matricelor

```
>> A=[1 2; 3 4]
```

```
>> B=[2 4; 3 5]
```

- Definirea vectorilor

```
>> C=[1; 2; 3]
```

```
>> D=[2;4;6]
```

- Extragerea unui element dintr-o matrice

```
>> A(1,2) %Elementul de pe linia 1 și coloana 2
```

- Extragerea unui element dintr-un vector

```
>> C(2) %Elementul 2
```

- Afșarea unei coloane dintr-o matrice

```
>> A(:,1) %Prima coloana
```

- Afișarea unei linii dintr-o matrice

```
>> a=A(2,:) % Linia a doua memorată ca variabila a
```

- Operații cu matrice

- Adunarea

```
>> M=A+B
```

- Înmulțirea

```
>> N=A*B
```

- Transpusa

```
>> A'
```

- Ridicarea la putere

```
>> A^2
```

➤ Inversa unei matrice

```
>> inv(A)
```

```
>> A^(-1)
```

➤ Determinantul

```
>> det(A)
```

- Operații cu vectori

➤ Produs scalar

```
>> dot(C,D)
```

- Operații cu tablouri

Operațiile cu tablouri sunt operații aritmetice (înmulțire, împărțire, ridicare la putere, etc.) între elementele situate pe aceeași poziție a tablourilor, cunoscute sub numele de operații element cu element.

➤ Înmulțirea tablourilor

```
>> A.*B
```

➤ Împărțirea tablourilor

```
>> A./B
```

➤ Ridicarea la putere

```
>> A.^4
```

- Generarea vectorilor cu pasul dat

Instrucțiunea de generare: **a=amin:pas:amax**

Generarea unui vector de la 1 la 10 cu pasul 2

```
>> a=1:2:10
```

- Generarea vectorilor dacă se dă numărul N de elemente

Instrucțiunea de generare: **a=linspace(amin,amax,N)** .

Pasul dintre două elemente se calculează automat conform formulei:  $\text{pas} = (\text{amax} - \text{amin}) / (N - 1)$ .

Generarea unui vector cu 4 elemente de la 1 la 10.

```
>> a=linspace(1,10,4)
```

**Sarcină:** Copiați toate liniile de comenzi din fereastra Comand History într-un fișier script și rulați fișierul.

### 3. Funcții. Reprezentare grafică în sistemul cartezian

- Reprezentarea grafică a funcției  $y = f(x), x \in [a, b]$ . Comanda plot.

**Exemplu 1:** Fie  $y = \sin(x)$ . Să se construiască graficul acestei funcții pe intervalul  $[a, b] = [0, 2\pi]$ .

Scrieți următoarele comenzi într-un fișier script și rulați-l

```
x=0 :pi/100 :2*pi ;  
y=sin(x);  
plot(x,y);  
xlabel('x=0 :2*pi') ; %etichetă pe x  
ylabel('functia sin(x)'); %etichetă pe y  
title('Graficul functiei sin(x)'); %titlul graficului
```

Obs: Aceeași funcție poate fi reprezentată cu ajutorul unui fișier de tip function.

- Fișiere de tip function

Dacă prima linie a fișierului conține cuvântul “**function**” atunci fișierul respectiv este declarat ca fișier funcție. O funcție diferă de un “script” prin faptul că poate lucra cu argumente. Variabilele definite și manipulate în interiorul fișierului funcție sunt localizate la nivelul acestuia. Prin urmare, la terminarea execuției unei funcții, în memoria calculatorului nu rămân decât variabile de ieșire ale acesteia.

Forma generală a primei linii unui fișier funcție este:

**function [param\_iesire]=nume\_funcție(param\_intrare)**

**Exemplu 2:** Să se scrie un fișier function care are ca parametru de intrare argumentul  $x$  și returnează valoarea funcției  $f(x) = \sin x$ . Să se calculeze  $f\left(\frac{\pi}{3}\right)$ . Să se construiască graficul acestei funcții pe intervalul  $[0, 2\pi]$ .

Într-un fișier de tip function scrieți comenzile de mai jos :

```
function y=fun(x)  
y=sin(x);
```

Fișierul se va salva cu numele fun.m și va fi apelat în fereastra de comenzi astfel :

```
>>fun(pi/3)
```

Pentru a construi graficul acestei funcții se va discretiza domeniul  $[0, 2\pi]$  și se vor calcula valorile ordonate ale punctelor apelând funcția  $f$ .

Scrieți într-un fișier script și rulați următoarea secvență de comenzi:

```
x=0 :pi/100 :2*pi ;  
y=fun(x);  
plot(x,y);
```

#### 4. Câteva elemente de programare

- Sintaxa generală a structurii if...elseif...else...end este:

```
if expresie  
    corpul 1 instrucțiuni  
elseif expresie  
    corpul 2 instrucțiuni  
else  
    corpul 3 instrucțiuni  
end
```

##### Ex. 1:

Să se definească următoarea funcție într-un fișier de tip function:

$$f(x) = \begin{cases} x^2, & x \in (-2, 0) \\ x^3, & x \in [0, 3) \\ x^2 + 18, & x \in [3, 4) \\ 0, & \text{in rest} \end{cases}$$

Să se construiască graficul acestei funcții pe intervalul  $[-5, 5]$ .

- Sintaxa buclei for este:

```
for index=expresie  
    corp instrucțiuni  
end
```

##### Ex. 2:

Calculați suma și produsul numerelor de la 1 la 10.

- Sintaxa while

```
while expresie  
    corp instrucțiuni  
end
```

**Ex. 3:**

Să se calculeze suma primelor numere astfel încât valoarea ei să nu depășească 1000.

**5. Calcul simbolic**

- Evaluarea expresiilor simbolice

Introduceți următoarele secvențe în fereastra Comand Window.

`syms a b` %declararea simbolică a variabilelor a, b

`c=a+b+2*b` %evaluarea simbolică a expresiilor algebrice

`A=[a a; b 2*a]`

`d=det(A)` %calculează simbolic determinantul matricei

`Ainv=inv(A)` %calculează simbolic inversa unei matrice

`A0=subs(A,{a,b},{2,3})` %substituie în matricea A variabilele a,b cu

%valorile corespunzătoare, 2 și respectiv 3.

- Derivata simbolică

`syms x y`

`f=x^2+y^2;`

`dfdx=diff(f,x)` %calculează derivata parțială în raport cu variabila x

`dfdy=diff(f,y)` %calculează derivata parțială în raport cu variabila y

`d2fdx2=diff(f,x,2)` %derivata de ordinul doi în raport cu x

- Integrala simbolică

`I1=int(f,x)`

`I2=int(f,y)` %calculează integrala nedefinită, făcând abstracție de

%constantă de integrare

`I3=int(x^3,1,2)` %calculează integrala definită  $\int_1^2 x^3 dx$

- Grafice de funcții folosind expresii simbolice. Comanda **ezplot**

Următorul exemplu construiește graficul funcției  $y = x^2 - 3x + 2, x \in [-1, 4]$

`syms x`

`f=x^2-3*x+2`

`ezplot(f,[-1,4])`