

**Universitatea Politehnica București**

## **Laborator 1 Teoria Sistemelor**

### **Introducere în MATLAB**

shiva.pub.ro

# Laboratorul 1

## Introducere în Matlab

MATLAB reprezintă una din cele mai răspândite platforme de calcul științific și numeric din lume. Această platformă este utilizată atât în diverse domenii ale industriei (industria auto, industria aerospațială), dar și în domenii de cercetare și dezvoltare. Denumirea MATLAB provine din *Matrix Laboratory*, evidențiind astfel operandul principal al platformei: **matricea**.

În *Teoria Sistemelor*, MATLAB reprezintă un instrument deosebit de util în etapele de proiectare, modelare și simulare a sistemelor dinamice.

### 1.1 Primii pași în MATLAB

**Pasul 1** - Pentru a putea lucra cu programul MATLAB, este esențial ca utilizatorul să își stabilească un **folder de lucru**. În cadrul acestei locații, se vor stoca **toate fișierele** asociate unei aplicații dezvoltate în MATLAB.

**Pasul 2** - Odată cu deschiderea soft-ului MATLAB, se poate putea vizualiza **fereastra principală** (Fig. 1.1). Această fereastră este împărțită în **trei secțiuni**:

- *Current Folder* - reprezintă secțiunea care afișează conținutul **folder-ului curent**.
- *Command Window* - **linia de comandă** asociată programului MATLAB. În cadrul acestei secțiuni, se pot introduce comenzi, efectua operații și apela funcții și scripturi MATLAB.
- *Workspace* - **spațiul de lucru** unde sunt expuse toate **variabilele** definite de utilizator.

**Pasul 3** - Se va configura programul în așa fel încât **folderul curent** (*Current Folder*) să fie același cu **folderul de lucru** ales anterior. Această configurare se poate realiza din bara de deasupra secțiunii *Command Window*.

## 1.2. DECLARAREA VARIABILELOR

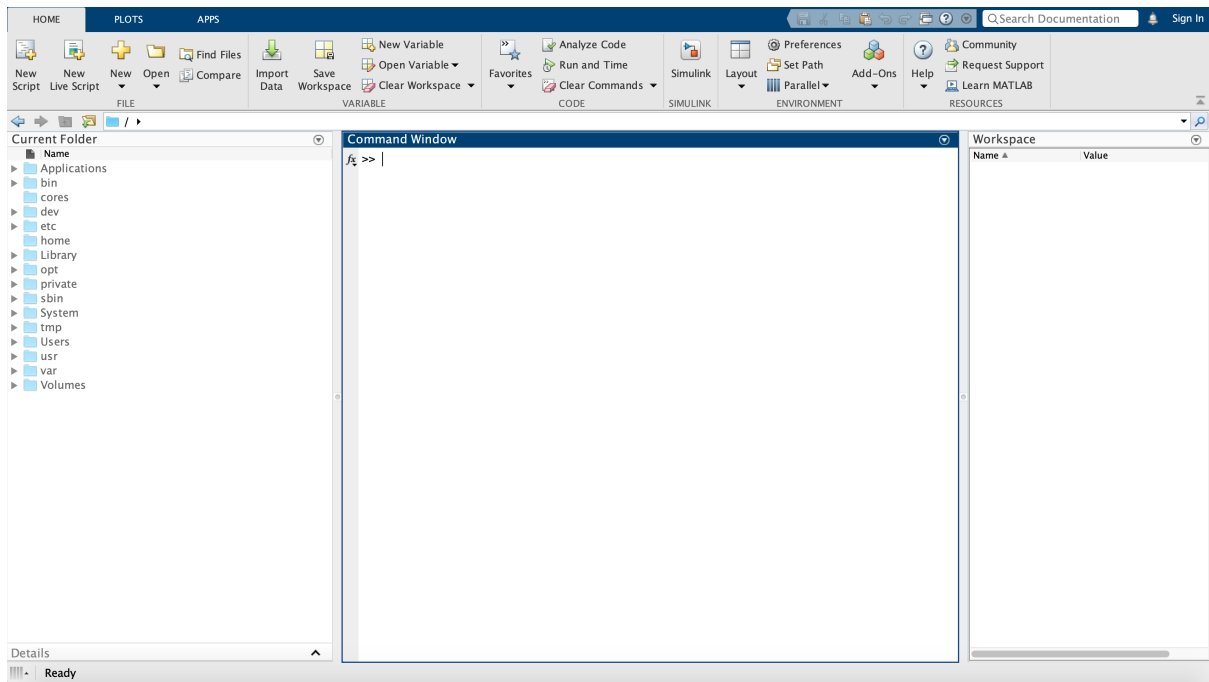


Figura 1.1: Fereastra principală a programului MATLAB

## 1.2 Declararea variabilelor

Din linia de comandă, se pot defini variabile simple de diverse tipuri (*int*, *string*, *char*). Definirea unei variabile presupune stabilirea unui nume și utilizarea operatorului de atribuire =, urmat de o valoare, o expresie sau o funcție.

### Exemple:

```
a = 2 + 3 %se declară variabila a și i se atribuie suma 2+3
b = a / 5
c = 't' %declararea unei variabile de tip char
d = "salut!" %declararea unei variabile de tip string
e = [1,2,3] %declararea unui vector linie
f = [1;2;3] %declararea unui vector coloana
g = [1,2;3,4] %declararea unei matrici
i = [1,2,3]' %transpunerea unui vector linie într-un vector
```

Pentru generarea vectorilor și matricilor, se pot utiliza o serie de funcții MATLAB specializate:

- Funcția *ones(m,n)* - generează o matrice de *m* linii și *n* coloane formată din **elemente de valoare 1**.
- Funcția *zeros(m,n)* - generează o matrice de *m* linii și *n* coloane formată din **elemente de valoare 0**.
- Funcția *linspace(a,b,n)* - generează un vector de *n* elemente egal distanțate în intervalul

## LABORATORUL 1. INTRODUCERE ÎN MATLAB

închis  $[a,b]$ .

- Dacă se dorește crearea unui vector cu un increment fix, atunci se poate utiliza următoarea procedură:

```
nume_vector = [primul_element:increment:ultimul_element];  
h = 1:3:20 %declaratia unui vector cu valori între  
%1 și 20, având incrementul 3
```

După definirea variabilelor, acestea pot fi **editate** sau **accesate** fie din linia de comandă, fie direct din *Workspace* dând dublu click pe numele variabilei.

O particularitate a programului MATLAB este dată de posibilitatea utilizatorilor de a defini **variabile simbolice** ce pot fi ulterior utilizate în ecuații și expresii matematice.

Declaratia variabilelor simbolice se face prin intermediul comenzii *syms*.

**Exemple:**

```
syms x  
syms y  
syms z  
z = sin(x)/y
```

### 1.3 Scripturi

MATLAB permite utilizatorilor să creeze fișiere specializate denumite **script-uri**. Script-urile reprezintă fișiere cu extensia *.m* care grupează mai multe instrucțiuni de tip MATLAB. La **executarea** unui script, interpretorul programului MATLAB execută secvențial toate funcțiile și instrucțiunile din scriptul respectiv.

Un script poate fi creat din meniul ferestrei principale, prin apăsarea butonului *New Script* din secțiunea *Home*.

De asemenea, se pot defini variabile prin intermediul unui script. Acele variabile sunt vizibile global și pot fi accesate sau modificate fie din **linia de comandă**, fie din *Workspace*.

**Executarea** unui script se poate face fie din secțiunea *Editor* - comanda **Run**, fie prin apelarea script-ului din linia de comandă prin intermediul denumirii.

### 1.4 Funcții

O funcție în MATLAB reprezintă un tip de fișier particular (cu extensia *.m*), având **propriul lor spațiu de lucru**, **argumente de intrare** și **argumente de ieșire**.

O funcție are obligatoriu sintaxa următoarea sintaxa:

## 1.5. COMANDA HELP

```
function [argumente_iesire] = nume_functie(argumente_intrare)
% Instructiuni
% din corpul functiei
%
end
```

Variabilele declarate în interiorul funcției au **caracter local** (sunt accesibile numai în spațiul de lucru al funcției). Transmiterea parametrilor în și din interiorul funcției se realizează prin intermediul argumentelor de intrare și de ieșire.

**Notă:** Argumentele au caracter opțional (există funcții care nu primesc argumente de intrare/ieșire) și în situația în care există mai multe argumente, acestea sunt separate prin virgulă.

**Observație:** Denumirea unui fișier care conține o funcție trebuie să fie *identică* cu denumirea funcției respective.

Programul MATLAB conține la instalare o serie de funcții deja definite intern ce pot fi apelate de utilizator (de exemplu funcțiile enumerate anterior: *ones*, *zeros*, *linspace*).

## 1.5 Comanda *help*

În situația în care utilizatorul dorește să afle mai multe detalii (mod de apelare, număr de argumente, exemple de utilizare) despre o funcție predefinită MATLAB, acesta poate apela funcția respectivă precedată de comanda *help*.

**Exemple:**

```
help sin
help syms
help linspace
```

## 1.6 Reprezentarea grafică în MATLAB

Programul MATLAB permite reprezentarea grafică a funcțiilor și a variabilelor, atât în format 2D, cât și în format 3D.

Pentru reprezentarea grafică a unui parametru  $y$  în funcție de un parametru  $x$  cu **parametrizarea stilului**  $z$ , se poate apela funcția *plot* astfel:

```
plot(x, y, z)
```

,unde  $x$  și  $y$  pot fi vectori sau matrici iar  $z$  reprezintă un șir de caractere format din unul sau mai multe elemente de stil din coloanele Tabelului 1.1

Șirul de caractere  $z$  reprezintă, astfel, **șirul de parametri de stil ai reprezentării grafice**. Elementele evidențiate mai sus reprezintă o selecție sumară a stilurilor definite în MATLAB,

## LABORATORUL 1. INTRODUCERE ÎN MATLAB

<b>b</b> <i>blue</i>	<b>.</b> <i>point</i>	<b>-</b> <i>solid</i>
<b>r</b> <i>red</i>	<b>o</b> <i>circle</i>	<b>:</b> <i>dotted</i>
<b>g</b> <i>green</i>	<b>x</b> <i>x-mark</i>	<b>-.</b> <i>dashdot</i>
<b>c</b> <i>cyan</i>	<b>+</b> <i>plus</i>	<b>--</b> <i>dashed</i>
<b>m</b> <i>magenta</i>	<b>*</b> <i>star</i>	
<b>y</b> <i>yellow</i>	<b>s</b> <i>square</i>	
<b>k</b> <i>black</i>	<b>d</b> <i>diamond</i>	

**Tabelul 1.1:** Elemente de stil pentru personalizarea graficelor MATLAB

se recomandă consultarea informațiilor oferite de instrucțiunea *help plot* pentru a descoperi întreaga colecție.

O reprezentare grafică în MATLAB este expusă într-o fereastră separată denumită **figure** (figură). Programul MATLAB permite combinarea mai multor reprezentări grafice pe aceeași figură. Acest lucru se poate realiza prin apelarea comenzii *plot*, având ca argumente de intrare **triplete de forma (X, Y, Z)** pentru fiecare reprezentare grafică în parte.

Apelul funcției *plot* pentru mai multe reprezentări grafice are următoarea formă:

```
plot(x1,y1,z1,x2,y2,z2,x3,y3,z3,...)
```

**Observație:** În apelarea funcției *plot* pentru un triplet (X, Y, Z), este obligatoriu ca numărul de coloane al parametrului X să fie identic cu numărul de coloane al parametrului Y.

### Exemplu

Pentru un vector de timp *t* și patru semnale *y1,y2,y3,y* deja definite în *Workspace*, secvența de cod pentru reprezentarea grafică a tuturor semnalelor pe aceeași figură poate fi:

```
plot(t,y1,'b+',t,y2,'r',t,y3,'g*',t,y,'m','LineWidth',1.5)
legend('y1 K=0.1','y2 K=0.5','y3 K=0.9','y treapta')
grid ON
```

Rezultatul poate fi vizualizat în fi vizualizat în Fig. 1.2.

**Notă:** pentru tripletul asociat semnalului *y* s-a utilizat proprietatea de modificare a grosimii graficului **LineWidth**.

De asemenea, a fost atașată graficului o **legendă** sugestivă prin intermediul comenzii **legend**. Apelul tipic al aceste comenzi este:

```
legend(label1, label2, ..., labelN)
```

Comanda **grid ON** activează **rasterul de fundal** al figurii.

**Observație:** Parametrii unei figuri pot fi modificați fie din linia de comandă (cum este cazul aici), fie din meniul figuri respective.

## 1.6. REPREZENTAREA GRAFICĂ ÎN MATLAB

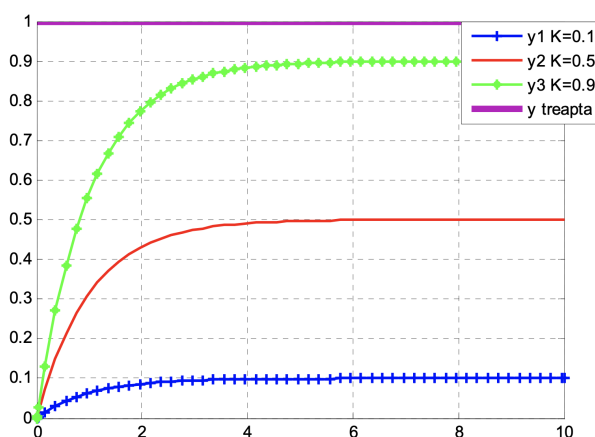


Figura 1.2: Exemplu - Reprezentarea grafică a patru semnale în aceeași figură.

### Chestiuni de studiat

Pentru semnalul sinusoidal definit de relația 1.1, să se realizeze cerințele de mai jos:

$$y(t) = 220\sqrt{2}\sin(50 \cdot 2\pi t) \quad (1.1)$$

1. Să se configureze programul MATLAB pentru implementarea aplicației, cu:
  - Folder-ul de lucru intitulat *Lab\_TS\_1*.
  - Să se creeze un script nou intitulat *main.m* în folder-ul de lucru.
2. Să se definească următoarele date de intrare ale problemei în script-ul creat la punctul precedent:
  - un vector *t* de 10 elemente egal distanțate între 0 și 0.02.
  - un vector *y* format din zerouri, având același număr de elemente ca vectorul *t*.
3. Să se definească o funcție MATLAB denumită *signalGen* care să implementeze ec. 1.1. Apelați corespunzător funcția în script-ul *main* și returnați rezultatul în vectorul *y*.
4. Să se genereze o reprezentare grafică a semnalului *y* în funcție de vectorul *t*, respectând următoarele cerințe:
  - Graficul generat să fie de culoare albastră, cu dimensiune 1.5 și linie continuă.
  - Rasterul de fundal al figurii este activat.
  - Conține o legendă corespunzătoare.
  - Prin intermediul comenzii *title* îi este atribuit un titlu corespunzător.
5. Repetați procedura de generare a graficului (a se vedea punctele 2,3,4) pentru intervalele  $[0, 0.5]$  și  $[0, 1]$ . În ambele cazuri, modificați numărul de elemente al vectorilor *t* și *y* la 100, apoi la 1000. Ce observați?