



ROMÂNIA  
MINISTERUL EDUCAȚIEI NAȚIONALE  
*Universitatea din Craiova*



**FACULTATEA DE INGINERIE ELECTRICĂ**  
B-dul. Decebal, nr. 107, Craiova 200440  
tel/fax: +40 251 436 447, e-mail : secretariat@ie.ucv.ro

---

**Prof.dr.ing. Mihaela POPESCU      As.dr.ing. Alexandra Pătrașcu**

# **GRAFICĂ ASISTATĂ DE CALCULATOR – MATLAB 7**

## **Îndrumar de laborator**



**EDITURA UNIVERSITARIA**  
**Craiova, 2013**

Referenți științifici:

Prof.univ.dr.ing. Alexandru BITOLEANU

Universitatea din Craiova

Conf.univ.dr.ing. Daniel CISMARU

Universitatea din Craiova

Copyright © 2013 Universitaria

Toate drepturile sunt rezervate Editurii Universitaria

---

**Descrierea CIP a Bibliotecii Naționale a României**

**POPESCU, MIHAELA**

**Grafică asistată de calculator - MATLAB 7 : îndrumar**

**de laborator** / Mihaela Popescu, Alexandra Pătrașcu. - Craiova : Universitaria, 2013

ISBN 978-606-14-0595-4

I. Pătrașcu, Alexandra

004.92(075.8)

Apărut: 2013

**TIPOGRAFIA UNIVERSITĂȚII DIN CRAIOVA**

Str. Brestei, nr. 156A, Craiova, Dolj, România

Tel.: +40 251 598054

Tipărit în România

## PREFAȚĂ

**MATLAB (MATrix LABoratory)** este limbajul de înaltă performanță și, mai ales, mediul de programare produs și dezvoltat de firma The MathWorks Inc., care integrează calculul matematic (numeric și simbolic), grafică avansată, analiză de date, explorare și vizualizare a acestora, aplicații de dezvoltare, inclusiv realizarea de interfețe grafice utilizator.

Sistemul interactiv cu care utilizatorul este în contact, puternic și ușor de utilizat, modul simplu de exprimare și rezolvare a problemelor de calcul tehnic cu formulări matriceale și vectoriale, dar și familia de aplicații specifice puse la dispoziție pentru rezolvarea unor clase particulare de probleme, fac din MATLAB un instrument de lucru foarte util oricărui inginer și cercetător.

Acest îndrumar de laborator se adresează, în special, studenților de la Facultatea de Inginerie Electrică a Universității din Craiova, programele de studii gestionate de Departamentul de Electromecanică, Mediu și Informatică Industrială, ca suport teoretic pentru aplicațiile aferente disciplinei Grafică asistată de calculator, programată în anul II al studiilor de licență. În egală măsură, este un material util tuturor celor care doresc să se familiarizeze cu facilitățile grafice oferite de mediul de programare MATLAB® 7.

După parcurgerea lucrărilor de laborator conținute în îndrumar, utilizatorul va fi familiarizat cu facilitățile grafice oferite de mediul de programare MATLAB 7 pentru reprezentarea grafică 2D și 3D a datelor conținute în vectori și matrice, a funcțiilor predefinite, interpolarea și ajustarea datelor, crearea și utilizarea interfețelor grafice utilizator.

Craiova, 2013

Autoarele

# L1. ÎNȚIERE ÎN UTILIZAREA MEDIULUI DE PROGRAMARE MATLAB 7

## 1.1. Scopul lucrării

Lucrarea are ca scop inițierea studenților în lucrul cu mediul de programare MATLAB 7 și cu facilitățile oferite de acesta.

## 1.2. Noțiuni teoretice

MATLAB este un sistem interactiv, al cărui element de bază este o matrice care nu are o dimensiune ce trebuie specificată. Acesta permite rezolvarea multor probleme de calcul tehnic, în special cu formulări matriceale și vectoriale.

### 1.2.1. Lansarea în execuție

Programul se lansează în execuție din mediul Windows prin selectarea pictogramei MATLAB.

### 1.2.2. Desktop-ul MATLAB 7

Desktop-ul MATLAB 7 constă în mai multe instrumente de tip interfață grafică utilizator (GUI - Graphical User Interfaces), cele mai multe dintre ele ferestre, care formează mediul de lucru și permit gestionarea variabilelor, fișierelor și a aplicațiilor asociate.

În mod implicit, aspectul desktop-ului este cel din fig. 1.1, dar acesta poate fi personalizat.

Instrumentele afișate în mod implicit sunt cele de mai jos.

1. **Command Window** (fereastra de comenzi) este fereastra principală prin care utilizatorul comunică cu interpretorul MATLAB.

Aceasta conține cursorul (prompt-ul) MATLAB de forma `>>` sub care se apelează comenzile MATLAB sau programe utilizator ce conțin instrucțiuni MATLAB numite fișiere `-M` (deoarece au extensia `.m`).

Afișarea prompt-ului indică faptul că interpretorul MATLAB este pregătit să accepte comenzi de la utilizator. Orice comandă este, în mod normal, terminată de apăsarea tastei Enter, în urma căreia MATLAB afișează în ecou rezultatul operației. Dacă însă înainte de Enter se folosește caracterul `" ; "` această afișare nu mai are loc.

Dacă în fereastra de comenzi este afișat prompt-ul `K>>`, înseamnă că MATLAB este în modul depanare (*debug mode*).

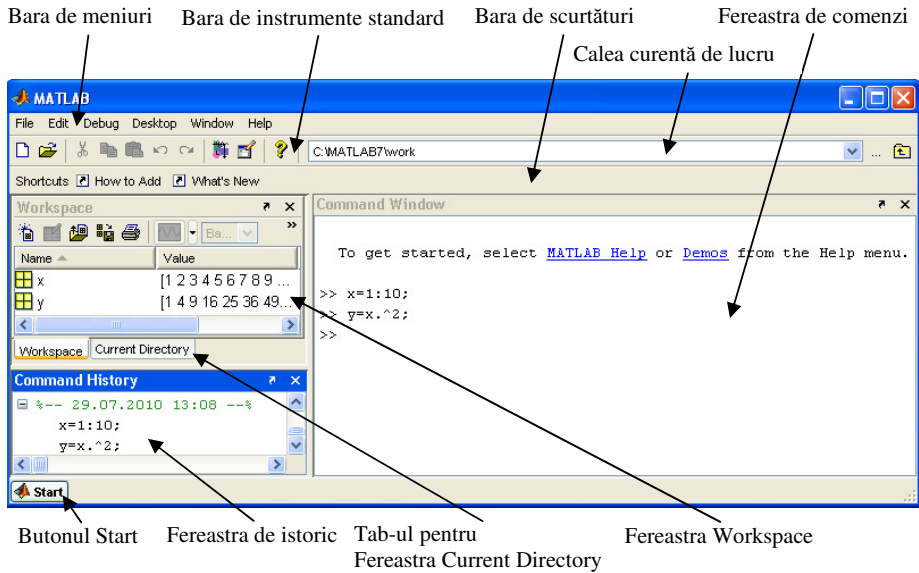


Fig. 1.1 Desktop-ul MATLAB implicit

2. ***Command History*** (fereastra cu istoricul) afișează o listă cu cele mai recent apelate comenzi, în ordinea cronologică a orelor de deschidere a sesiunilor MATLAB.

Operațiile asupra comenzii/comenzilor selectate din *Command History* pot fi efectuate, în modul cel mai simplu, prin selectarea acestora din meniul contextual, afișat cu un clic dreapta cu mouse-ul pe selecție (fig. 1.2).

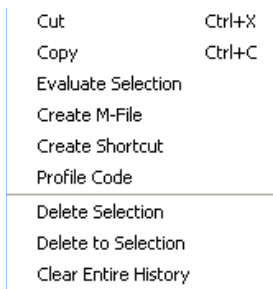


Fig. 1.2 Meniul contextual asociat ferestrei *Command History*

3. ***Browser-ul Workspace***, afișează variabilele existente în spațiul de lucru (*workspace*) MATLAB. Conținutul acestuia se actualizează automat pe măsură ce variabilele sunt create/șterse.

După selectarea uneia sau mai multor variabile, pot fi realizate o serie de acțiuni, precum vizualizarea, editarea sau reprezentarea grafică, prin utilizarea opțiunilor meniului contextual (afișat cu un clic dreapta al mouse-ului) sau a butoanelor de pe bara de instrumente a ferestrei (fig. 1.3).

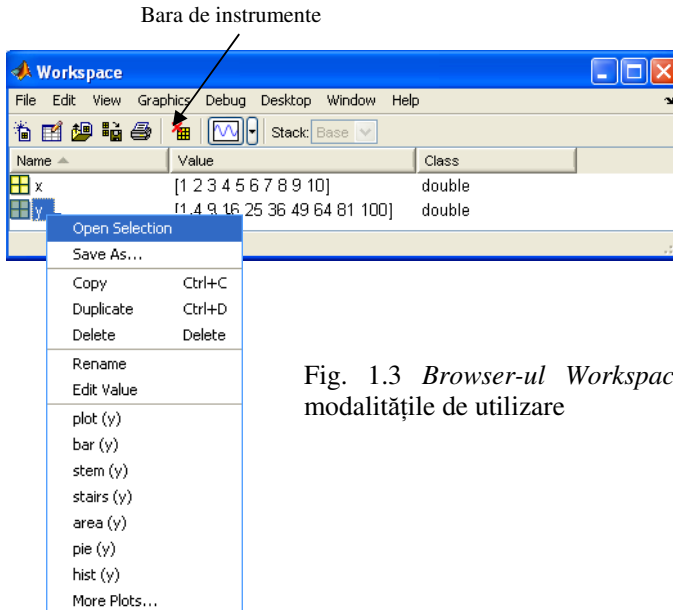


Fig. 1.3 *Browser-ul Workspace* și modalitățile de utilizare

4. ***Browser-ul Current Directory*** permite căutarea, vizualizarea, deschiderea și efectuarea de modificări referitoare la foldere și fișiere MATLAB (fig 1.4).

Permit schimbarea căii curente de lucru

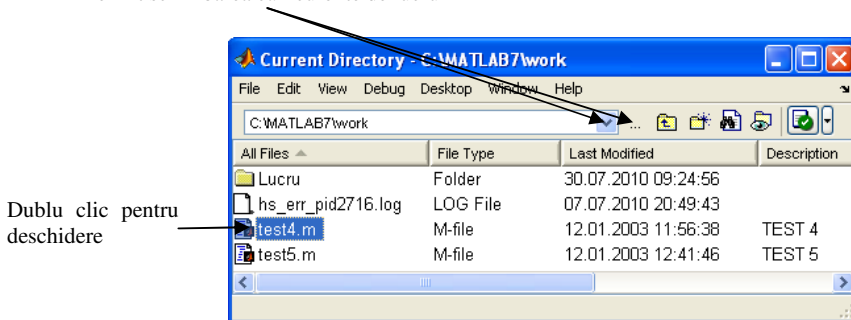



Fig. 1.4 *Browser-ul Current Directory*

5. ***Butonul Start*** , afișat în colțul din stânga jos, permite, prin opțiunile disponibile (fig. 1.5), accesul la diferite instrumente MATLAB, demonstrații, documentații, dar și crearea/rularea unor shortcut-uri (grupuri de instrucțiuni) MATLAB.

Controlul afișării/ascunderii instrumentelor desktop se realizează prin selectarea/deselectarea articolelor de meniu corespunzătoare din meniul *Desktop* (fig. 1.6).

Există configurații predefinite pentru modul de prezentare a desktop-ului, care pot fi selectate din meniul *Desktop*, articolul *Desktop Layout* ► (fig. 1.7).



Fig. 1.5 Opțiunile butonului *Start*

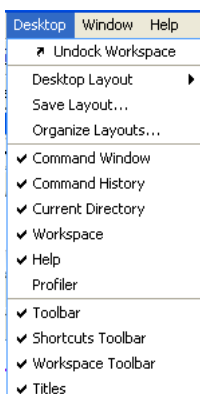


Fig. 1.6 Articolele meniului *Desktop*

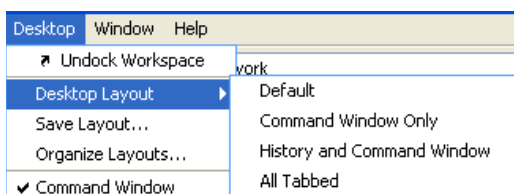




Fig. 1.7 Opțiunile articolului *Desktop Layout*

*Notă: Toate ferestrele desktop-ului suportă operații de tipul undock/dock (detașare/atașare) prin apăsarea butonului  /  din partea dreaptă a barei de titlu. În modul undocked, fereastra desprinsă este mărită, poate fi ușor redimensionată și, uneori, poate afișa elemente care nu apar în modul docked.*

### 1.2.3. Executarea comenzilor MATLAB

În modul cel mai simplu, instrucțiunile MATLAB se execută prin tastare, sub prompt-ul `>>` din *Command Window*, urmată de apăsarea tastei `Enter` (fig. 1.1).

Atunci când se folosesc instrucțiuni de programare (precum *if*, *for*, *while* etc.), deși se folosește tasta `Enter` pentru a trece la rândul următor, prompt-ul `>>` nu apare decât după terminarea setului de instrucțiuni.

Pentru a executa o funcție MATLAB, se tastează numele funcției (inclusiv argumentele) sub prompt-ul `>>` și se apasă tasta Enter, care determină execuția. După execuție, prompt-ul `>>` re apare.

Pentru a executa un fișier-M de tip script (o succesiune de instrucțiuni MATLAB salvate într-un fișier cu extensia `.m`), se tastează numele acestuia sub prompt-ul `>>` și se apasă tasta Enter.

Pentru a vizualiza conținutul unei variabile, se tastează numele acesteia sub prompt-ul `>>` și se apasă tasta Enter.

Înteruperea execuției unei comenzi sau rulării unui program se realizează apăsând combinația de taste Ctrl – c sau Ctrl – Break.

## 1.2.4. Lucrul cu matrice

Sistemul MATLAB este caracterizat prin lucrul cu matrice, referite ca *arrays* în interacționarea utilizator-software. În acest sens, scalarii sunt asimilați matricelor cu o linie și o coloană ( $1 \times 1$ ), iar vectorii sunt asociați matricelor cu o linie ( $1 \times n$ ) sau cu o coloană ( $n \times 1$ ).

MATLAB-ul nu conține instrucțiuni de declarații de tip și dimensiune, memoria fiind alocată automat, până la valoarea maxim disponibilă.

Elementele unei matrici pot fi de diferite tipuri, respectând tipul de date acceptate de MATLAB.

### 1.2.4.1. Crearea matricelor

Cea mai simplă metodă de creare a unei matrici cu elemente numerice sau caractere text este introducerea explicită a elementelor acesteia pe linii și coloane, respectând următoarele reguli generale:

- elementele matricii să fie cuprinse între paranteze drepte `, [ ] ,`,
- elementele unei linii să fie separate prin spații sau virgule
- liniile să fie separate prin caracterul `, ; ,`

*Exemplu:* Matricea A se creează în linia de comandă prin secvența MATLAB

```
>> A=[1 1 1; 1 2 3; 1 3 6]
```

care returnează rezultatul

```
A =
     1     1     1
     1     2     3
     1     3     6
```

Pentru definirea matricelor mari, la care datele de intrare nu încap pe o singură linie, se pot scrie liniile pe rânduri diferite, înlocuind caracterul `, ; ,` cu apăsarea tastei Enter.

*Exemplu:*

```
>> A = [1 1 1
        1 2 3
        1 3 6]
```

Matricele mai pot fi generate prin:

- instrucțiuni și funcții;
- executarea unui fișier -M;
- apelarea numelui unui fișier de date existent.



### 1.2.4.2. Tipuri de date în MATLAB

Principalele tipuri de date folosite în MATLAB sunt:

- *double* (numere reale dublă precizie – pe 64 biți);
- *single* (numere reale simplă precizie – pe 32 biți);
- *int8*, *int16*, *int32*, *int64* (numere întregi – pe 8 / 16 / 32 / 64 biți);
- *char* (șir de caractere);
- *struct* (date de diferite tipuri și dimensiuni plasate în câmpurile unei structuri);
- *cell* (date de diferite tipuri și dimensiuni plasate în celulele unei matrice);
- *logical* (pentru condiții de relaționare și testare, cu valorile 0 și 1).

Pentru a vedea tipul de date al variabilei A, se folosește funcția *class(A)*.

Pentru exemplele de mai sus, *class(A)* returnează valoarea *double*.

Tabelul următor oferă câteva exemple de creare a unei variabile cu diferite tipuri de date.

Secvența MATLAB	Tipul de date rezultat	Valoarea afișată ca rezultat
x=25	double	25
x=25.5	double	25.5000
x=single(25.5)	single	25.5000
x=int16(25)	int16	25
x='Atentie'	char	Atentie
x=['t1'; 't2']	char	t1 t2
x={'Linie de text'; 'Linie de text de alta lungime'}	cell	'Linie de text' 'Linie de text de alta lungime'
x.nume='Ana' x.an=1990	struct	nume: 'Ana' an: 1990
[1 5]>2	logical	0 1

### 1.2.4.3. Crearea vectorilor cu elemente valori numerice echidistante

Sintaxa clasică utilizată pentru a crea un vector linie cu elemente valori numerice echidistante este:

***x=xmin:pas:xmax***

unde *xmin* este valoarea de la care se pleacă, *pas* este diferența dintre două valori succesive (poate fi și un număr negativ) și *xmax* este valoarea finală care nu se dorește a fi depășită.

Cu sintaxa

***x=xmin: xmax***

se interpretează că pasul este implicit 1.

În tabelul de mai jos, sunt prezentate câteva exemple de secvențe MATLAB.

>> x=1:2:8 x = 1 3 5 7	>> x=1:-1:-3 x = 1 0 -1 -2 -3
>> x=1:0.2:2 x = 1.0000 1.2000 1.4000 1.6000 1.8000 2.0000	>> x=1:4 x = 1 2 3 4

În MATLAB 7, poate fi folosită și funcția *linspace*, cu sintaxa

$x=linspace(xmin, xmax, n)$

în care *n* este numărul de elemente ale vectorului.

Exemplu: >> x=linspace(1, 2,5)

x =  
1.0000 1.2500 1.5000 1.7500 2.0000

Cu sintaxa simplificată  $x=linspace(xmin, xmax)$  se consideră că, implicit, *n* = 100.

#### 1.2.4.4. Determinarea dimensiunilor matricilor și vectorilor

De foarte multe ori, chiar în programarea simplă, este nevoie să se determine și rețină dimensiunea unei matrice existente sau a unui vector. Pentru aceasta există funcțiile:

$[l,c]=size(x);$

unde *x* este matricea sau vectorul existent în spațiul de lucru, iar *l* și *c* sunt variabilele în care va fi reținut numărul de linii și coloane al matricei/vectorului.

$n=length(x);$

unde *x* este vectorul existent în spațiul de lucru, iar *n* variabila în care va fi reținut numărul de elemente al acestuia;

Dacă *x* este o matrice, funcția *length* furnizează maximul dintre numărul de linii și coloane al acesteia. De aceea, nu este indicat să se folosească funcția *length* pentru matrice.

#### 1.2.4.5. Asamblarea matricelor mari (metoda concatenării)

O matrice poate fi definită prin concatenarea unor matrice existente, de dimensiuni inferioare, cu condiția ca matricele folosite în asamblare să se poată integra compact în matricea rezultantă. În caz contrar (dimensiuni necorespunzătoare) se obțin mesaje de eroare. Tot metoda concatenării se folosește și pentru a adăuga linii/ coloane noi la o matrice/vector care există.

Exemple:

<pre>&gt;&gt;A=[1 1 1; 1 2 3]; B=[1 3 6; 5 7 9]; C=[A;B] C =  1  1  1  1  2  3  1  3  6  5  7  9</pre>	<pre>&gt;&gt; A=[1 1 1; 1 2 3]; &gt;&gt; D=[A; 4 6 8] D =  1  1  1  1  2  3  4  6  8</pre>
<pre>&gt;&gt; t={'Linie de text'; 'Linie de text de alta lungime'};t1=[t;{'linie'}] t1 =  'Linie de text'  'Linie de text de alta lungime'  'linie'</pre>	<pre>&gt;&gt; x=[1:2:5 8 10] x =  1  3  5  8  10</pre>

#### 1.2.4.6. Manipularea matricelor

##### Extragerea submatricelor și subvectorilor prin indici

Specificarea elementelor individuale ale unei matrice se realizează prin numele acesteia urmat de doi indici (al liniei și al coloanei) cuprinși între paranteze rotunde și separați de virgulă. Indicii pot fi scalari sau vectori.