

3. MEDIUL DE PROGRAMARE MATLAB

3.1 Prezentare generală

Mediul de dezvoltare a aplicațiilor Matlab este organizat pe o structură clasică de directoare, așa cum se arată în figura 3.1. Fișierul *executabil*, care lansează Matlab-ul în execuție se găsește în subdirectorul *bin*. Accesul la el se face de regulă prin pictograma specifică aflată pe desktop:



Matlab-ul este constituit din pachete de programe de înaltă performanță destinate rezolvării unei game largi de probleme. Cea mai importantă componentă a sa este colecția de instrumente de modelare specializate pe diverse domenii tehnice și nu numai, organizate în așa numitele *toolbox-uri*. Alături de aceasta există și alte pachete de programe destinate modelării și simulării sistemelor cum sunt: *simulink*, *stateflow*, *rtw* și bineînțeles, un puternic suport demonstrativ *demos*.

Name ▲	Size	Type
bin		File Folder
demos		File Folder
extern		File Folder
help		File Folder
ja		File Folder
java		File Folder
notebook		File Folder
rtw		File Folder
simulink		File Folder
stateflow		File Folder
sys		File Folder
toolbox		File Folder
uninstall		File Folder
webserver		File Folder
work		File Folder
license	47 KB	Text Document
MATLAB 6.5	1 KB	Shortcut

Fig. 3.1. Organizarea directoarelor mediului Matlab

- Lansarea în execuție a mediului de dezvoltare a programelor Matlab deschide pe ecran fereastra de comandă (Command Window) , așa cum se arată în figura 3.2. Este de remarcat *prompt*-ul specific `>>` după care Matlab-ul așteaptă în general introducerea comenzilor.

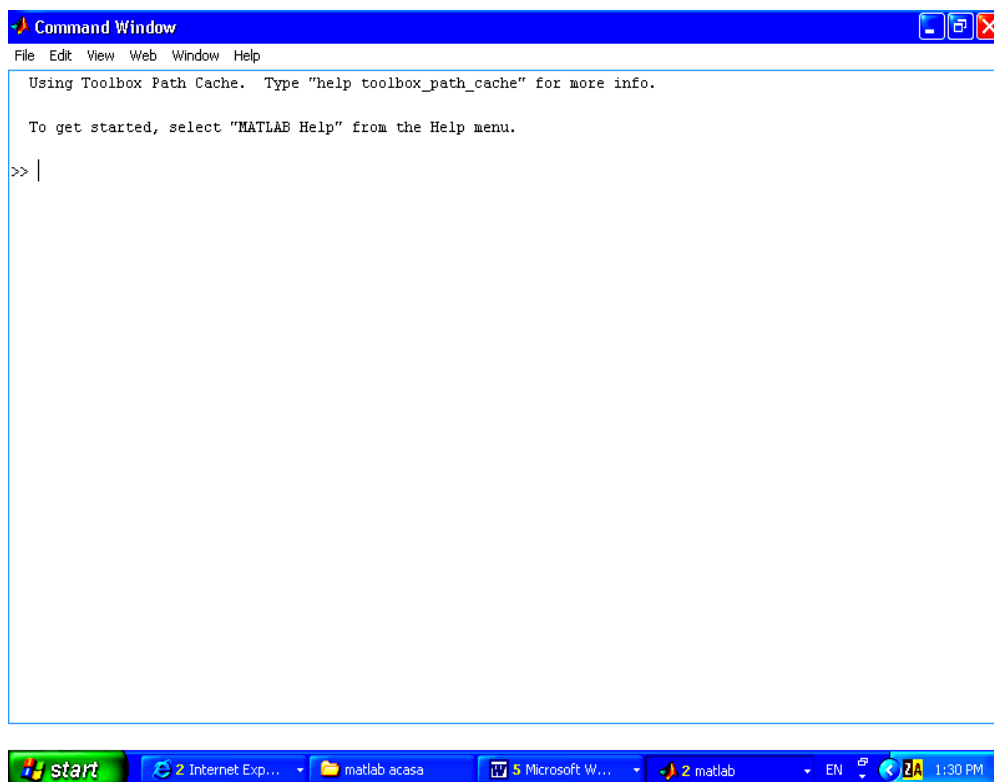


Fig. 3.2. Ecran cu fereastra principală în Matlab

Fereastra de comandă poate fi însoțită opțional așa cum se poate vedea de pildă în Fig. 3.3, de ferestre suplimentare, ce pot fi selectate din submeniul comenzii View în funcție de necesitățile utilizatorului, astfel:

- Command History pentru eșalonarea, vizualizarea, reapelarea funcțiilor (instrucțiunilor) lansate pe parcursul întregii sesiuni de lucru.

Notă. Instrucțiunile anterioare introduse pe parcursul unei sesiuni de lucru pot fi readuse succesiv în linia curentă de editare cu ajutorul săgeților de navigare sus-jos.

- **Current Directory** pentru afișarea conținutului și informațiilor despre fișierele din directorul de lucru curent.
- **Workspace** permite vizualizarea memoriei de lucru și a caracteristicilor variabilelor din sesiunea de lucru curentă.
- **Launch Pad** permite accesul facil la componentele mediului Matlab, inclusiv la Help, pentru selectarea și lansarea acestora (Simulink, Toolboxuri, etc.).
- **Profiler** permite urmărirea execuției programelor Matlab și prezentarea unor rapoarte de analiză statistică a execuției acestora cu privire la timpul consumat pentru execuția funcțiilor și instrucțiunilor componente.

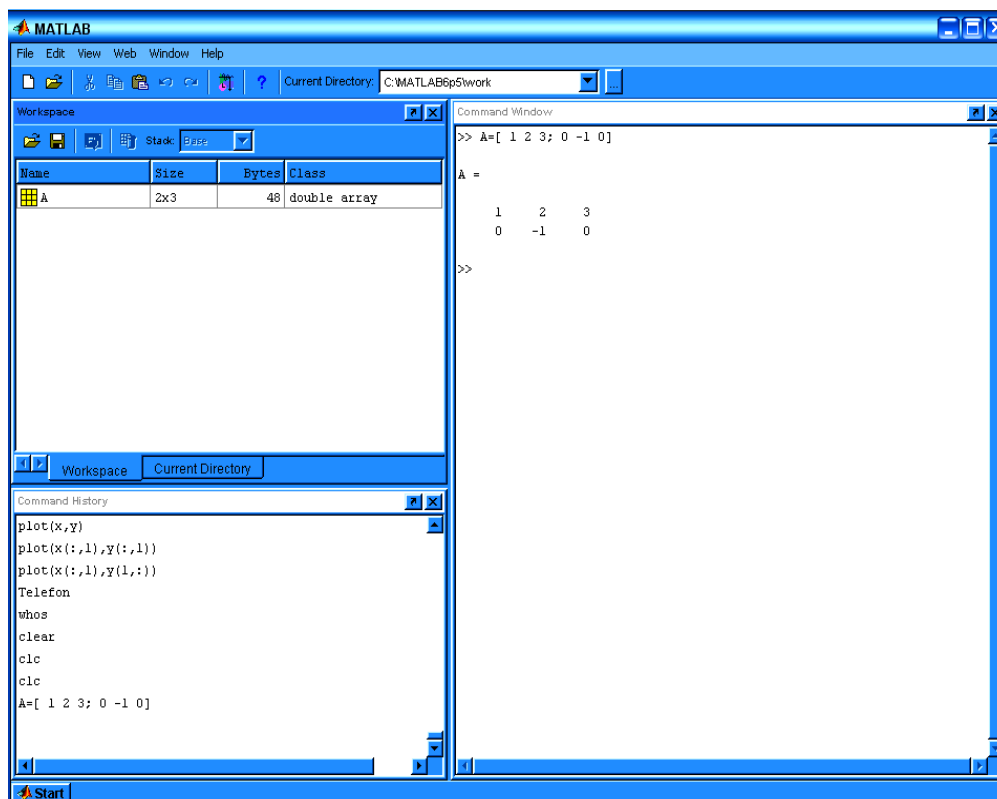


Fig. 3.3. Configurația cu ferestre Matlab opționale

Pe bara de comenzi a ferestrei principale se găsesc meniurile de apel pentru diferite grupe de comenzi, organizate în maniera standard: File, Edit, View, Web, Windows, Help. Conținutul acestor meniuri este detaliat în figura 3.4.

În meniul asociat comenzii File explicităm câteva dintre comenzi după cum urmează:

New – deschide o fereastră de editare goală (untitled) pentru următoarele tipuri de fișiere, la alegere:

- programe Matlab (.m),
- figură (.fig),
- model (.mdl),
- GUI (.fig).

Open – deschide fereastra de editare cu un fișier ales și selectat de utilizator.

Import Data – Încarcă date din fișiere recunoscute de Matlab în memoria de lucru curentă (Workspace). Se pot încărca date dintr-o multitudine de tipuri de fișiere (cum ar fi din Excel, fișiere text, fișiere imagine, audio, etc.), pe care Matlab-ul le asimilează ca matrici.

Save Workspace As – Salvează conținutul memoriei de lucru curente ca fișier de date cu extensia .mat.

Set Path – gestionează căile de directoare și subdirectoare de căutare ale Matlab-ului.

Preferences – configurează diverse proprietăți de lucru și de afișare ale mediului Matlab.

În general, comenzile și meniurile Matlab-ului sunt tipice standardului de interfață Windows. Fereastra principală poate conține bară de unelte cu butoane la care sunt asociate comenzile curente de lucru sub forma pictogramelor sugestive. Nu vom insista cu prezentarea acestora în detaliu, însușirea lor fiind la îndemâna oricărui utilizator familiarizat cu utilizarea calculatorului.

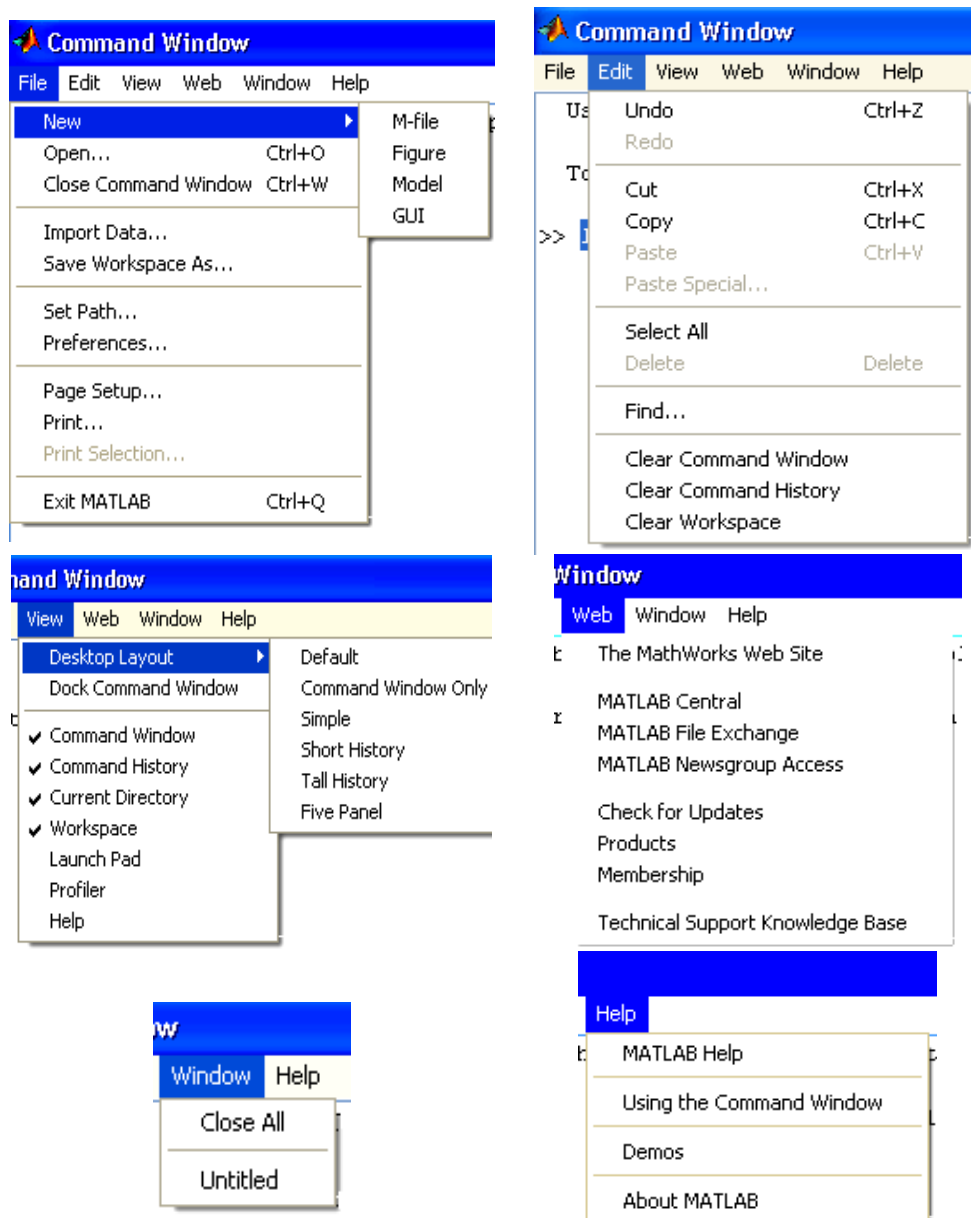


Fig. 3.4. Meniurile principale de comenzi în Matlab

3.2. Moduri de lucru în Matlab

Mediul Matlab utilizează un limbaj de programare de nivel înalt interpretat. Spre deosebire de alte medii de programare, în care dezvoltarea de programe presupune editarea unitară a programului și trecerea codului prin fazele de *compilare* și *execuție*, în Matlab se poate lucra mult mai flexibil și mai facil. În figura 3.5 este descrisă schematic maniera în care poate fi utilizat Matlab-ul pentru rezolvarea de probleme și dezvoltarea programelor de aplicație.

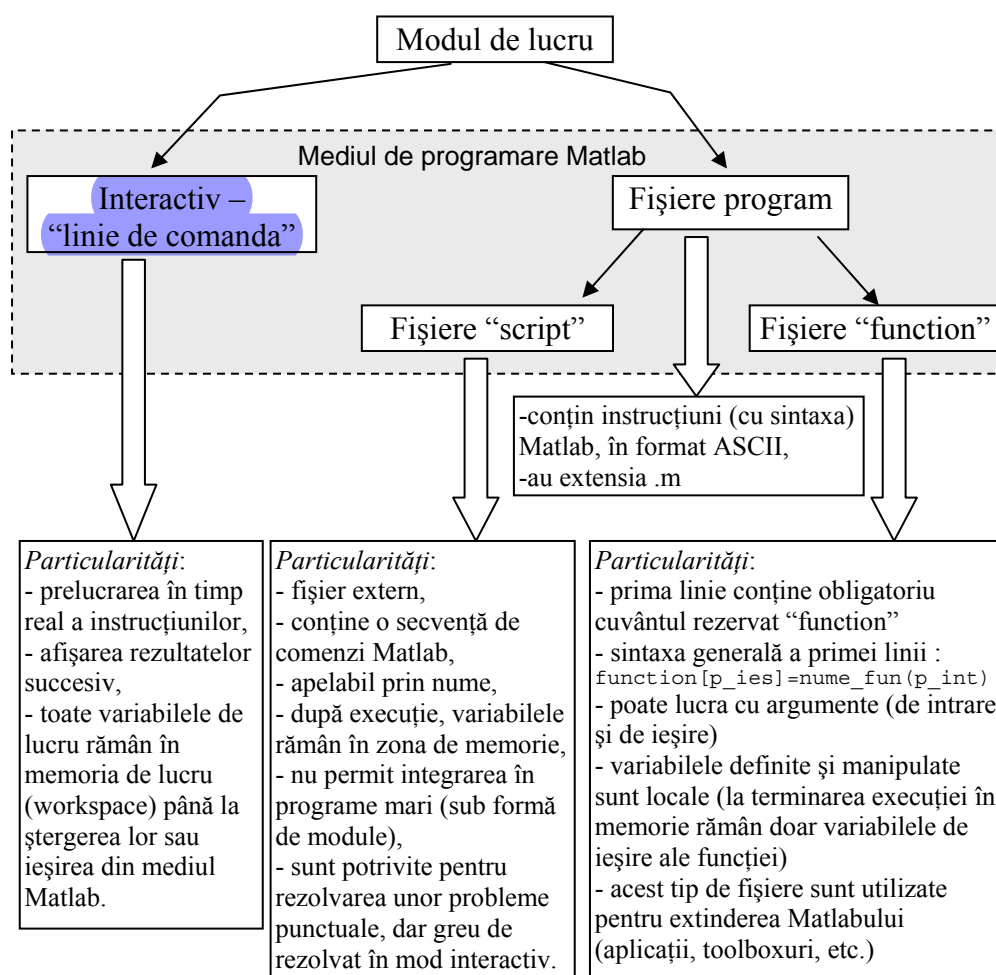


Fig. 3.5. Schema modurilor de programare în Matlab


a) Modul de programare "linie de comandă"

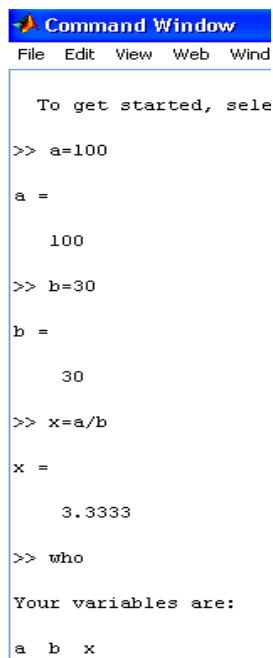
Pentru a lucra interactiv în modul "linie de comandă" este suficient să se înceapă scrierea instrucțiunilor în fereastra principală (fereastra de comenzi) sub prompt-ul acesteia `>>`. După fiecare instrucțiune/comandă se apasă tasta **Enter**, având ca efect interpretarea și execuția respectivei instrucțiuni sau comenzi, dacă s-a respectat sintaxa limbajului. În caz contrar apare un mesaj de eroare sau de avertisment. Rezultatul instrucțiunii/comenzii executate se manifestă imediat și după caz, acesta este afișat pe ecran unde rămâne până la o eventuală comandă de ștergere. Valorile numerice sau nenumerice ale variabilelor de lucru rămân de asemenea în memoria de lucru în sesiunea curentă (Workspace) până la o eventuală comandă de ștergere. De exemplu, în figura 3.6 este prezentat rezultatul efectuării unei operații de împărțire, urmat de confirmarea (verificarea) faptului (cu comanda `who`) că la final toate variabilele implicate au rămas în memoria de lucru.

b) Modul de lucru cu fișiere program

Pentru a lucra cu programe de tip fișier este necesar să se deschidă o fereastră de editare a codului sursă al programului. Această operație se face prin succesiunea de comenzi: *File*→*New*→*M-file*, care determină deschiderea unei ferestre de editare corespunzătoare unui nou fișier Matlab (cu extensia `.m`), având numele implicit **Untitled**, așa cum se arată în figura 3.7. Fereastra furnizează toate instrumentele pentru manipularea fișierelor: editare, vizualizare, salvare, depanare, help, etc.. Această fereastră permite editarea programelor (codului sursă) în mod text, fiind din acest punct de vedere perfect compatibilă cu fereastra Notepad.

Execuția programelor Matlab se poate face în câteva moduri :

- din fereastra de editare prin comanda *Debug*→*Run*,
- din fereastra de editare cu butonul pentru *Run* : ,
- din fereastra de editare cu tasta F5,
- din linia de comandă tastând numele fișierului program fără extensie.



The screenshot shows the MATLAB Command Window. The title bar is blue with the MATLAB logo and the text 'Command Window'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Web', and 'Wind'. The main area is white and contains the following text: 'To get started, sele', '>> a=100', 'a =', '100', '>> b=30', 'b =', '30', '>> x=a/b', 'x =', '3.3333', '>> who', 'Your variables are:', and 'a b x'.

```
Command Window
File Edit View Web Wind

To get started, sele

>> a=100

a =

    100

>> b=30

b =

    30

>> x=a/b

x =

    3.3333

>> who

Your variables are:

a b x
```

Fig. 3.6. Detaliu de lucru în modul linie de comandă

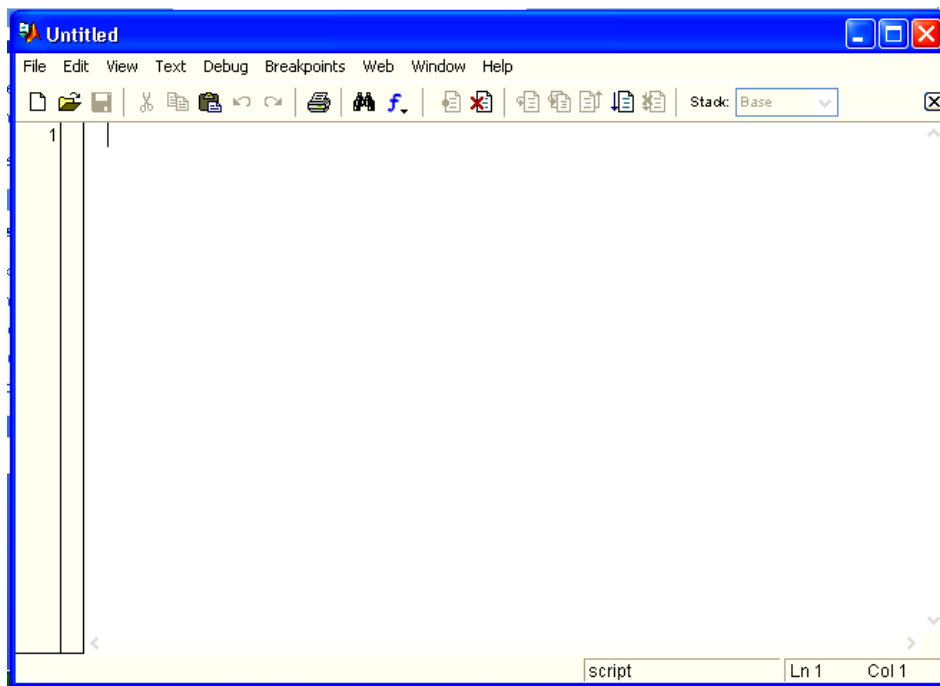


Fig. 3.7. Fereastra de editare a fișierelor program

Mediul de programare Matlab recunoaște diferite tipuri de fișiere:

- fișiere "script" sau function (programe Matlab) cu extensia *m*;
- fișiere externe mediului Matlab provenite din fișiere script compilate pentru alt mediu de programare, care se recunosc cu extensia *mex* ;
- fișiere *model* care conțin aplicații în Simulink și au extensia *mdl* ;
- fișiere de date cu extensia *mat* ;
- fișiere de tip figură (inclusiv fișierele GUI) specifice mediului Matlab, cu extensia *fig* ;
- fișiere cu aplicații dezvoltate cu diferite toolboxuri ale Matlab-ului (de exemplu sisteme de inferență fuzzy create cu toolboxul Fuzzy Logic cu extensia *fis*).

c) Modul de lucru simbolic

Modul de programare simbolică este un mod special de reprezentare a datelor și expresiilor ce apropie rezolvarea problemelor de modul clasic în care se folosește creionul și hârtia. Această facilitate de programare se întâlnește în medii de calcul ca: MathCad, Mathematica, Derive. În Matlab există toolbox-ul symbolic care permite lucrul cu expresii interpretate ca structuri de date simbolice. Bazele acestui sistem de calcul se regăsesc în mediul de calcul Maple. Maniera de efectuare a calculelor presupune recunoașterea unui tip special de dată numit *obiect simbolic*, care intern se reprezintă sub formă de șiruri de simboluri. Astfel, mediul Matlab permite rezolvarea de probleme ingineresti prin calcul simbolic pe baza interpretării tuturor variabilelor, expresiilor și matricelor ca obiecte simbolice.

Toolbox-ul symbolic pune la dispoziție o colecție de funcții pentru conversia diferitelor structuri de date în și din obiecte simbolice. De asemenea există o serie de funcții pentru manipularea obiectelor simbolice și pentru operarea cu expresii simbolice. Funcția `sym` este o funcție de bază pentru crearea obiectelor simbolice. În esență, pentru a înțelege diferența de interpretare a datelor în modul de lucru clasic (de pildă cu numere tip `double`) și în modul de lucru simbolic, vom exemplifica astfel :

<pre>>> sqrt(3) ans = 1.7321</pre>	<pre>>> a=sqrt(sym(3)) a = 3^(1/2)</pre>
--	--

Problematika calculului simbolic în Matlab va fi tratată într-o lucrare separată.

3.3. Structura programelor Matlab

Scrierea programelor Matlab este mai simplă comparativ cu majoritatea limbajelor de programare. *Mnemotehnica* și *simbolistica* utilizate pentru editarea comenzilor și instrucțiunilor în Matlab sunt apropiate de modul simbolic de scriere din matematică și nu abundă în particularități. De asemenea, *sintaxa* limbajului de programare este ușor de înțeles, intuitivă și nu face apel la reguli speciale. De exemplu, atribuirea valorii numerice 5 unei variabile A se exprimă prin instrucțiunea:

```
A= 5
```

iar o condiție se declară în Matlab prin următoarea instrucțiune:

```
if a < 5  
    k=k+1
```

Un program de tip fișier "script" poate să înceapă direct cu prima instrucțiune, fără alte declarații suplimentare, cu excepția programelor de tip funcție, care încep obligatoriu cu cuvântul cheie *function*. În Matlab declarațiile de variabile nu sunt necesare, deoarece acestea se *autodeclară* în momentul primei utilizări. Programele Matlab de tip "script" pot să înceapă cu linii de comentariu. Orice program de tip funcție este în esență un fișier (program) "script", care începe obligatoriu cu declarația (instrucțiunea):

```
function[param_ies]=nume_fun(param_int)
```

În programele funcție liniile de comentariu sunt permise după declarația *function* de mai sus.

În continuare se dau două exemple relevante atât pentru modul cum arată un program Matlab, cât și pentru a pune în evidență diferențele de structură dintre programele *script* și cele de tip *function*. Menționăm că orice program script poate fi definit ca program funcție și recunoscut ca atare prin efectuarea unor modificări minime așa cum se poate vedea în exemplele următoare.

- (a) Exemplu de transformare a unui program de tip fișier "script" (vezi caseta interioară) într-un fișier funcție, prin adăugarea liniei speciale de declarație *function* și eventual a unor comentarii. Se remarcă parametrii de ieșire separați prin virgulă în cadrul listei variabilelor de ieșire demarcată de paranteze pătrate.

```
function[m,d]=pstatistici
%calculeaza parametrii statistici medie si dispersie
%pentru un sir de 100 numere aleatoare
```

```
%fisier script
n=100;
sir_aleat=rand(1,n);
m=mean(sir_aleat);
d=std(sir_aleat);
disp('sir_aleat m d')
```

- (b) Exemplu de program *function* cu parametru de intrare în lista de argumente de intrare. Se remarcă parametrul de intrare *n* precizat ca argument al funcției *pstatistici* prin care se comunică acesteia pentru câte numere generate aleator se vor calcula parametrii medie și dispersie.

```
function[m,d]=pstatistici(n)
%calculeaza parametrii statistici medie si dispersie
%pentru un sir de n numere aleatoare
```

```
%fisier script
sir_aleat=rand(1,n);
m=mean(sir_aleat);
d=std(sir_aleat);
disp('sir_aleat m d')
```

3.4. Funcții de control general

Funcțiile de control general sunt destinate pentru controlul mediului de programare Matlab, precum și pentru obținerea de informații diverse cu privire la mediul de programare și la sistemul de calcul. În tabelul 3.1 sunt descrise o serie de astfel de funcții.

Tab. 3.1 Funcții Matlab de uz general

Funcția	Exemplu de aplicare	Returnează
<i>help</i>	<code>help nume_funcție</code>	Informații despre funcția nume_funcție
<i>help help</i>	<code>help help</code>	Informații despre însăși modul de utilizare a help-ului.
<i>lookfor</i>	<code>Lookfor cuvânt_căutat</code>	Toate numele de fișiere (din toți directorii), care au în prima linie de help cuvinte căutate.
<i>who</i>	<code>who</code>	Afișează toate variabilele din memoria de lucru curentă (workspace), de exemplu: Your variables are: D X1 ans Y1
<i>whos</i>	<code>whos</code>	Afișează inclusiv proprietăți ale variabilelor. Name Size Elements D 1 by 8 8 Bytes Density Complex 64 Full Ivo
<i>which</i>	<code>which demo</code>	Returnează calea în care se găsește un anumit fișier. C:\matlab\toolbox\matlab\demos\demo.m
<i>what</i>	<code>what general</code>	m.files in directory c:\matlab\toolbox\matlab\general cd diary getmax length memory readme unix whos cedit dir help load more
<i>type</i>	<code>type <<nume_fisier>></code>	Afișează lista instrucțiunilor din fișier
<i>path</i>	<code>path</code> <code>p= path</code> <code>path(p)</code> <code>path(p1,p2)</code> <code>path(path,c:\MATLAB\TEST)</code>	Afișează lista de căi, sau adaugă o nouă cale de căutare respectiv, C:\MATLAB\TEST la cele existente deja (în funcția path).
<i>exist</i>	<code>exist(nume_variabila)</code> <code>exist(startup)</code>	0 -dacă startup nu există; 1 -dacă startup este variabilă; 2 -dacă startup este un fișier .m într-una din căile Matlab; 3 -dacă startup este un fișier .mex într-una din căile Matlab; 4 -dacă startup este o funcție SIMULINK; 5 -dacă startup este o funcție internă a Matlab-ului.
<i>cd</i>	<code>cd</code> <code>cd<<cale/nume_director>></code>	→ afișează numele directorului curent; → schimbă directorul.

<i>dir</i>	<code>dir</code> <code>dir<<nume_director>></code>	→ afișează numele tuturor fișierelor din directorul curent; → afișează numele tuturor fișierelor din <i>nume director</i> .
<i>delete</i>	<code>delete <nume_fisier></code>	→ șterge fișierului <i>nume_fisier</i> .
<i>diary</i>	<code>diary <nume_fisier></code> <code>diary off</code> <code>diary on</code>	Efectuează o copie pe disc a sesiunii de lucru (comenzi de intrare și răspunsuri returnate) în <i>nume_fisier</i> , (nume implicit: <i>diary</i>) -suspendarea înregistrării cu argumentul <i>off</i> ; -reluarea înregistrării cu argumentul <i>on</i> ;
<i>!</i>	<code>!tp</code> <code>!format A:</code>	Lansează mediul TP. Formatează discheta din unitatea A (comandă DOS).
<i>clock</i>	<code>clock</code>	DATA=[1999 03 12 19 5 10]
<i>cputime</i>	<code>cputime</code>	Funcție pentru determinarea timpului de calcul al CPU
<i>etime</i>	<code>etime(t_{final},t₀)</code> <code>etime(clock,t₀)</code>	Răspunsul în secunde (durata=...)
<i>date</i>	<code>D=date</code>	Citeste data din SETUP D=[03-March-2003]
<i>tic,toc</i>	Succesiunea : <code>tic</code> <code>toc</code>	Funcții pentru pornirea și oprirea unui cronometru <code>elapsed_time</code> = durata în secunde dintre cele două comenzi
<i>clc</i>	<code>clc</code>	Șterge fereastra de comenzi a Matlab-ului
<i>home</i>	<code>home</code>	Poziționează cursorul în locația corespunzătoare primei linii și primei coloane a ferestrei de comenzi. Concomitent șterge și conținutul ferestrei
<i>format</i>	<code>format <optiune></code> <code>optiune = short, long, longe, shorte, hex, etc.</code> <code>format compact</code> <code>format loose</code>	Stabilește afișarea informației în diferite formate. <code>compact</code> - suprimă liniile libere; <code>loose</code> –afișare degajată (cu linii libere)
<i>echo</i>	<code>echo</code>	Afișează liniile de program și rezultatele în timpul execuției acestora
<i>more</i>	<code>more (n)</code>	Afișează succesiv câte n linii de program (sau matrice de date)
<i>disp</i>	<code>disp(X)</code> <code>disp('Nr. Var1 Var2')</code>	Afișează matricea X sau afișează textul dintre ghilimele: 'sir caractere'.
<i>clear</i>	<code>clear x</code> <code>clear x y z</code> <code>clear</code>	Șterge din spațiul (memoria) de lucru al aplicației: numai variabila x, variabilele x, y, z sau toate variabilele existente.
<i>pack</i>	<code>pack</code>	Compactează spațiul de memorie rămas discontinuu în urma creării/ștergerii variabilelor de lucru.
<i>size</i>	<code>[m,n]=size(X)</code>	Returnează dimensiunile matricei X: m-numărul de linii, n-numărul de coloane.
<i>length</i>	<code>a=length(X)</code>	Returnează lungimea (dimensiunea) vectorului X, sau numărul de coloane.

<i>save</i>	<code>save</code>	Salvarea <i>în format binar</i> a tuturor variabilelor din spațiul de lucru, în fișierul <code>matlab.mat</code>
	<code>save nume_fisier</code>	Salvarea în format binar a tuturor variabilelor din spațiul de lucru, în fișierul <code>nume_fisier.mat</code>
	<code>save nume_fisier A B C</code>	Salvarea în format binar a variabilelor A, B, C din spațiul de lucru, în fișierul <code>nume_fisier.mat</code>
	<code>save nume_fisier.extensie A B C -ascii</code>	Salvarea în format binar a variabilelor A, B, C din spațiul de lucru, în fișierul <code>nume_fisier.extensie</code>
<i>load</i>	<code>load</code>	Încarcă datele din fișierul <code>matlab.mat</code> .
	<code>load nume_fisier</code>	Încarcă datele din fișiere <i>în format binar</i> cu extensie <code>mat</code> .
	<code>load nume_fisier.extensie -mat</code>	Încarcă datele din fișiere <i>în format binar</i> cu extensie.
	<code>load nume_fisier-ascii</code>	Încarcă datele din fișiere <i>în format ASCII</i> , fara extensie.
	<code>load nume_fisier.extensie-ascii</code>	Încarcă datele din fișiere <i>în format ASCII</i> , cu extensie.