

Marian Nicolae VELEA

MATLAB

- Instrument de calcul în inginerie -



EDITURA
UNIVERSITĂȚII
TRANSILVANIA
DIN BRAȘOV

2022

CUPRINS

Prefață	9
1 Introducere despre MATLAB	11
1.1 Generalități	11
1.2 Interfața	11
1.3 Introducerea comenzilor și citirea rezultatelor	12
1.4 Formate numerice de afișare	13
1.5 Caractere speciale în Matlab	14
1.6 Tipuri de date	15
1.6.1 Denumirea variabilelor	15
1.6.2 Variabile booleene	16
1.6.3 Variabile text	16
1.6.4 Variabile numerice	17
1.6.5 Variabile celule – tablouri de celule	18
1.6.6 Variabile tabel	19
1.6.7 Variabile structura	20
1.6.8 Variabile predefinite	21
1.7 Tipuri de operatori	22
1.7.1 Operatori aritmetici	22
1.7.2 Operatori logici (Booleeni)	22
1.7.3 Operatori Relaționali	23
1.7.4 Prioritatea operatorilor	23
1.8 Funcții matematice elementare predefinite	24
1.9 Alte comenzi utile	26
1.10 Aplicații	27
2 Tablouri	29
2.1 Tablouri numerice	29
2.1.1 Unidimensionale – Vectori	29
2.1.2 Bidimensionale – Matrice	34
2.1.3 Funcții speciale pentru tablourile numerice	42
2.2 Tipuri de operații aritmetice între tablouri numerice	44
2.2.1 Operații matriceale	46
2.2.2 Operații element cu element	53

2.2.3	Utilizarea tablourilor numerice ca argumente ale funcțiilor predefinite	55
2.3	Comenzi utile pentru analiza tablourilor numerice	56
2.4	Generarea numerelor aleatorii	57
2.5	Aplicații	59
3	Afișarea datelor sub formă de grafice	65
3.1	Grafice bidimensionale (2D)	65
3.1.1	Funcția <code>plot</code>	65
3.1.2	Funcția <code>fplot</code>	66
3.1.3	Funcții și comenzi de bază pentru editarea graficelor	66
3.1.4	Afișarea grafică a unor seturi de date multiple	68
3.1.5	Afișarea legendei	70
3.1.6	Denumirea și dimensionarea axelor	70
3.1.7	Afișarea titlului	70
3.1.8	Salvarea graficelor în fișiere de tip imagine	70
3.1.9	Afișarea de text în grafice	72
3.1.10	Axe cu scală dublă pe ordonată	76
3.1.11	Axe logaritmice	77
3.1.12	Bare de eroare	77
3.1.13	Grafice 2D speciale	78
3.1.14	Sub-grafice	84
3.1.15	Animatii 2D	86
3.1.16	Afișarea graficelor în ferestre multiple	86
3.1.17	Alte comenzi utile	86
3.2	Grafice tridimensionale	87
3.2.1	Funcțiile <code>surf</code> și <code>surfc</code>	87
3.2.2	Funcția <code>mesh</code> și <code>meshc</code>	88
3.2.3	Funcția <code>fplot3</code>	89
3.3	Aplicații	91
4	Funcții	93
4.1	Funcții predefinite	93
4.2	Funcții definite de către utilizator	93
4.3	Funcții anonime	96
4.4	Aplicații	98
5	Moduri de definire a variabilelor	101
5.1	Interacțiunea utilizatorului cu Matlab prin dialog în fereastra de comandă	101

5.2	Salvarea și încărcarea variabilelor folosind fișiere specifice Matlab.....	105
5.3	Salvarea și încărcarea datelor folosind fișiere externe.....	105
5.4	Aplicație	111
6	Polinoame	113
6.1	Valoarea unui polinom	113
6.2	Rădăcinile unui polinom	114
6.3	Operații cu polinoame	115
6.3.1	Adunarea și scăderea polinoamelor	115
6.3.2	Înmulțirea polinoamelor	116
6.3.3	Împărțirea polinoamelor	116
6.4	Derivarea polinoamelor.....	117
6.5	Aplicații.....	118
7	Analiza regresiei.....	121
7.1	Funcții polinomiale	121
7.2	Funcții putere	127
7.3	Funcții exponențiale	129
7.4	Funcții logaritmice	130
7.5	Aplicații.....	132
8	Interpolare.....	133
8.1	Noțiuni introductive	133
8.2	Interpolare unidimensională.....	133
8.3	Interpolare bidimensională.....	136
8.4	Aplicații.....	139
9	Transformata Fourier	141
9.1	Noțiuni introductive	141
9.2	Comenzi Matlab	142
9.3	Aplicații.....	145
10	Rezolvarea numerică a ecuațiilor integrale și diferențiale.....	147
10.1	Integrarea numerică.....	147
10.2	Derivarea numerică	151

10.3	Ecuatii diferențiale de ordinul I.....	153
10.4	Aplicații.....	155
11	Calcul simbolic	157
11.1	Crearea obiectelor simbolice.....	157
11.2	Simplificarea expresiilor	158
11.2.1	Comanda <code>collect</code>	158
11.2.2	Comanda <code>expand</code>	160
11.2.3	Comanda <code>factor</code>	160
11.2.4	Comanda <code>simplify</code>	161
11.2.5	Comanda <code>pretty</code>	161
11.3	Rezolvarea ecuațiilor algebrice	162
11.4	Rezolvarea sistemelor de ecuații algebrice	163
11.5	Integrarea expresiilor simbolice	165
11.6	Derivarea expresiilor simbolice	167
11.7	Rezolvarea ecuațiilor diferențiale ordinare	168
11.8	Aplicații.....	173
12	Programarea cu Matlab	175
12.1	Structuri condiționale	175
12.1.1	<code>if-end</code>	175
12.1.2	<code>if-else-end</code>	176
12.1.3	<code>if-elseif-else-end</code>	177
12.1.4	<code>switch-case</code>	179
12.2	Structuri repetitive.....	181
12.2.1	<code>for</code>	181
12.2.2	<code>while-end</code>	182
12.2.3	Structuri imbricate	184
12.3	Comanda <code>break</code>	185
12.4	Comanda <code>continue</code>	186
12.5	Aplicații.....	188
13	Crearea de interfețe grafice	191
13.1	Crearea panoului de comandă	192
13.2	Scrierea liniilor de cod	194
13.3	Executarea și testarea aplicației.....	197
13.4	Crearea aplicațiilor de tip <i>stand-alone</i>	198

13.5 Aplicații.....	202
Lista figuri.....	203
Lista tabele	205
Lista exemple	207
BIBLIOGRAFIE	213

1 Introducere despre MATLAB

1.1 Generalități

Matlab (**Matrix Laboratory**) reprezintă un mediu de programare cu ajutorul căruia se pot efectua, calcule matematice numerice sau simbolice, se pot analiza, prelucra și vizualiza grafic seturi de date, se pot dezvolta algoritmi de calcul, pot fi compilate aplicații cu interfețe grafice care pot rula independent de platforma Matlab. Este folosit în domenii ce țin de inginerie, medicină, economie, etc.

Capabilitățile Matlab discutate în această carte precum și aplicațiile prezentate au fost lucrate în versiunea 2021b - licența academică. Documentarea funcțiilor specifice MATLAB a fost realizată folosind ca resursă principală *Matlab R2021 Documentation* [1]. Alte resurse internaționale de documentare recomandate celor interesați de mediul de programare MATLAB sunt referințele [2]–[5].

1.2 Interfața

Interfața implicit setată care se deschise la prima pornire a programului Matlab este prezentată în Figura 1.

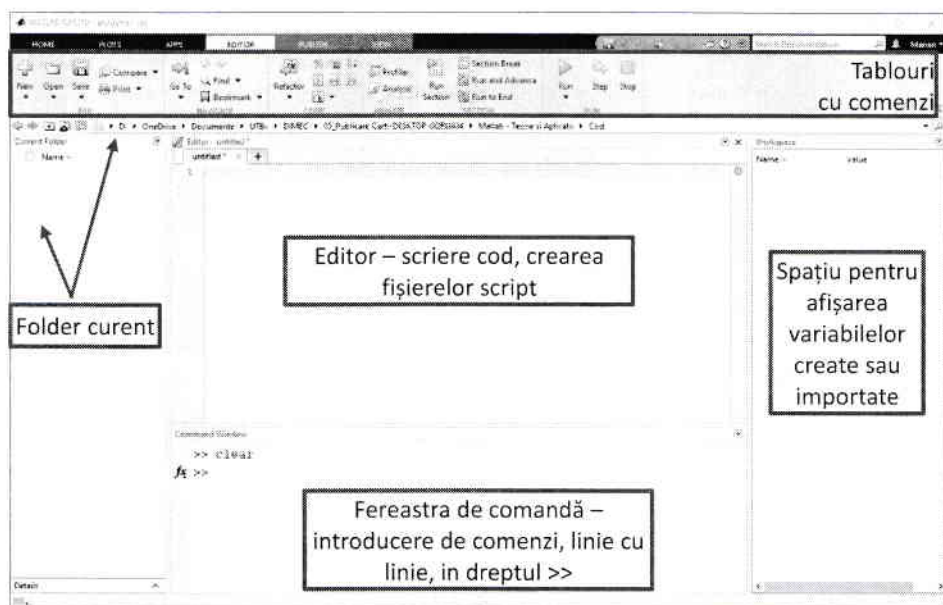


Figura 1 Interfața grafică cu utilizatorul – aranjament implicit

Aceasta poate conține:

Command window – fereastra de comandă

Current folder – directorul curent de lucru

Workspace – spațiu pentru afișarea variabilelor stocate sau importate

Editor – zona de scriere a fișierelor script (m-files) sau a fișierelor tip funcție

Meniul principal, organizat grupat pe tablouri cu comenzi în funcție de tipurile de acțiuni:

- *HOME* – comenzi generale pentru crearea sau deschiderea de fișiere, încărcare date, crearea, salvarea sau ștergerea variabilelor aflate în spațiul de stocare, setări preferențiale;
- *PLOTS* – comenzi legate de generarea graficelor;
- *APPS* – aplicații conexe Matlab (de exemplu, Simulink);
- *EDITOR* – editare, executare script;
- *PUBLISH* – editarea codului în vederea publicării acestuia în format html;
- *VIEW* – comenzi pentru organizarea interfeței.

1.3 Introducerea comenzilor și citirea rezultatelor

Utilizatorul poate introduce comenzi prin fereastra de comandă (*Command window*) sau în cadrul fișierelor tip *script*. În primul caz, comenzile se introduc linie cu linie, în dreptul simbolului `>>`, după cum este arătat în Figura 2, iar executarea acestora se realizează prin apăsarea tastei *Enter*, după fiecare linie; astfel, Matlab poate fi folosit precum un calculator obișnuit. Afișarea rezultatelor se face tot în fereastra de comandă (Figura 2) sau în spațiul de stocare a variabilelor (*workspace*) (Figura 3).

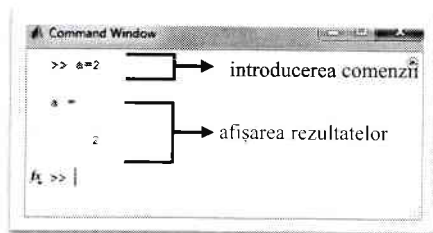


Figura 2 Scrierea comenzilor și citirea rezultatelor folosind fereastra de comandă

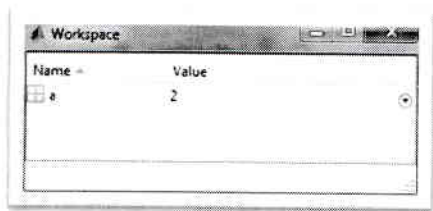


Figura 3 Spațiul stocare a variabilelor

În unele cazuri, când se efectuează calcule complexe, este benefică inhibarea afișării rezultatului în fereastra de comandă, pentru reducerea timpului de calcul; acest lucru se poate realiza prin adăugarea semnului ortografic „;” la sfârșitul comenzii.

În cel de-al doilea caz, comenzile se scriu în cadrul unor fișiere (fișiere tip *script* sau fișiere tip *funcție*) care conțin instrucțiunile de program iar executarea acestora se realizează prin apăsarea tastei *F5* de pe tastatură sau prin lansarea comenzii *Run* din tabloul cu comenzi, tab-ul *EDITOR*. În Exemplul 1-1 este prezentat modul de scriere a comenzii într-un fișier script. În acest exemplu, după rularea fișierului script, în *Workspace* va fi stocată variabila *a* care are valoarea 2.

Exemplul 1-1 Definirea unei variabile într-un fișier script

```
a=2;
```

→ Apăsare tasta *Run* sau *F5* → atribuirea valorii 2 variabilei *a*, cu stocare în *workspace*

În cazul în care, în urma rulării unor comenzi, se intră într-o buclă infinită sau timpul de calcul este mai mare decât se aștepta, se poate întrerupe execuția comenzilor folosind combinația de taste *Ctrl + c*.

1.4 Formate numerice de afișare

Schimbarea formatului de afișare a numerelor în fereastra de comandă este posibilă prin executarea comenzii `format` urmată de specificarea tipului de format, din cele prezentate în Tabel 1.

Tabel 1 Formate numerice de afișare

Format afișare	Descriere Rezultat	Rezultat
<code>short</code>	Numerele sunt afișate cu 4 zecimale. Format implicit	3.1416
<code>long</code>	Numerele sunt afișate cu 14 zecimale	3.141592653589793
<code>short E</code>	Numerele sunt afișate în notație științifică, cu 4 zecimale	3.1416e+00
<code>long E</code>	Numerele sunt afișate în notație științifică, cu 15 zecimale	3.141592653589793e+00
<code>bank</code>	Numerele sunt afișate cu 2 zecimale	3.14
<code>rat</code>	Rezultatul este afișat sub forma unui număr rațional	355/113
<code>+</code>	Rezultatul afișat este +, - sau 0	+
<code>hex</code>	Reprezentare hexazecimală a numerelor binare	400921fb54442d18

Spațiile dintre rândurile de text afișate în fereastra de comandă se poate regla prin selectarea uneia din opțiunile prezentate în Tabel 2.

Tabel 2 Formate ale spațiilor de afișare

Format afișare	Descriere Rezultat	Rezultat
compact	Elimină liniile goale pentru o afișare mai compactă a informației	>> alpha=pi/2 alpha = 1.5708
loose	Adaugă câte o linie goală pentru a facilita citirea. Format implicit.	>> alpha=pi/2 alpha = 1.5708

1.5 Caractere speciale în Matlab

Sunt acele caractere cu rol funcțional în sintaxa Matlab. În Tabel 3 sunt descrise mai multe astfel de caractere speciale.

Tabel 3 Caractere speciale

()	- Paranteze în expresiile matematice - Definirea argumentului unei funcții - Indexare
[]	- Definirea de vectori / matrice - Definirea de matrice goale sau ștergerea elementelor unui tablou/matrice - Definirea de rezultate cu valori multiple
[,]	- Separator între elementele aceleiași linii dintr-o matrice. - Spațiul liber dintre elementele unei matrice este echivalent cu utilizarea semnului „, „
[;]	Separator între liniile unei matrice.
{}	- Definirea unui tablou de celule - Accesează conținutul unei celule dintr-un tablou de celule
%	Tot ce este scris după acest caracter este considerat comentariu și nu este luat în calcul la rularea programului
%%	Delimitarea unui script în mai multe celule ce pot fi executate separat
%{ comentariu %}	Definirea unui bloc de comentarii
:	- Generarea diviziunilor – crearea de vectori - Definire bucle de tip FOR - Indexare
!	- Comenzi ce se aplică din Matlab sistemului de operare al calculatorului (creare / ștergere de directoare, mutare/copiere/ștergere fișiere, etc) - Accesare de medii de programare externe Matlab (C, Fortran)
;	Separator între instrucțiuni pe aceeași linie de comandă, fără afișare în fereastra de comandă

	• Semnalează sfârșitul de linie la definirea unei matrice
,	• Separator între instrucțiuni pe aceeași linie de comandă, cu afișare în fereastra de comandă • Separator între elementele unei linii dintr-o matrice / tablou • Separator între argumentele de intrare sau valorile de ieșire ale unei funcții
'	• Transpusa unei matrice
' '	• Creare de variabile vectori de caractere
" "	• Creare de variabile șiruri de caractere
=	• Atribuirea de valori
==	• Operator de verificarea a egalității
...	• Continuarea pe linia următoare a unei comenzi

1.6 Tipuri de date

În Matlab pot fi prelucrate mai multe tipuri de date sub forma de variabile numerice, logice, text, tabele, celule sau structuri (Figura 4).

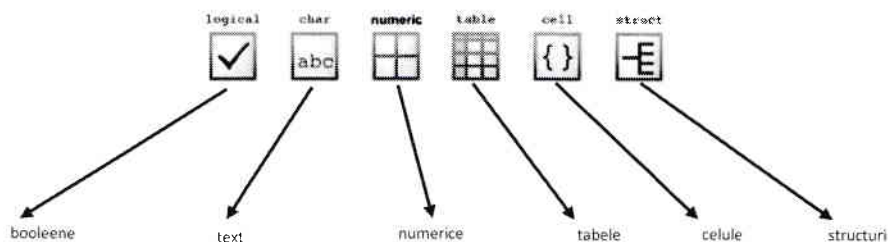


Figura 4 Tipuri de variabile utilizabile în Matlab

1.6.1 Denumirea variabilelor

Numele variabilelor trebuie să înceapă cu o literă și să nu conțină semne de punctuație sau spații, Tabel 4. De asemenea, un nume scris cu litere mici va fi înțeles ca fiind diferit față de același

nume dar scris cu litere mari. Trebuie evitate denumiri lungi întrucât cresc șansele unor greșeli la scriere.

Tabel 4 Denumirea variabilelor în Matlab

<i>forta_1</i>	da
<i>1_forta</i>	nu
<i>forta 1</i>	nu
<i>forta-1</i>	nu
<i>forta</i>	da
<i>Forta</i>	da, dar diferit de <i>forta</i>

1.6.2 Variabile booleene

Acestea sunt utilizate pentru stocarea stărilor cu valoarea de *adevăr* (1), Exemplul 1-2, sau cu valoarea de *fals* (0), Exemplul 1-3.

Exemplul 1-2 Atribuirea valorii logice 1

```
>> A=true
```

```
A=1
```

→ Definirea variabilei *A*

→ Afișarea rezultatului - atribuirea valorii logice 1 variabilei *A*

Exemplul 1-3 Atribuirea valorii logice 0

```
>> B=false
```

```
B=0
```

→ Definirea variabilei *B*

→ Afișarea rezultatului - atribuirea valorii logice 0 variabilei *B*

1.6.3 Variabile text

Există două moduri de stocare a textelor în Matlab:

- Vectori de caractere (''),
- Șiruri de caractere (" ")

Exemplul 1-4 Vector de caractere

```
A_char= 'Nume prenume'
```

→ definirea variabilei *A_char*

```
A_char =
```

```
'Nume prenume'
```

→ afișarea rezultatului - atribuirea textului *Nume prenume* variabilei *A* sub forma unui vector cu 12 elemente de tip caracter (toate caracterele, inclusiv spațiul). Rezultă o variabilă de tip vector de caractere cu dimensiunea 1 x 12

Exemplul 1-5 Șir de caractere

```
>> A_sir="Nume prenume"
```

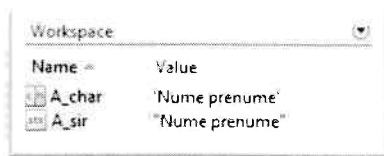
→ definirea variabilei *A_sir*

```
A_sir =
```

```
"Nume prenume"
```

→ afișarea rezultatului - atribuirea textului *Nume prenume* variabilei *A* sub forma unui singur șir de caractere. Rezultă o variabilă de tip tablou de caractere cu dimensiunea 1 x 1

Diferența dintre cele două tipuri de variabile text este vizibilă și în spațiul de stocare (Figura 5).



Name	Value
A_char	'Nume prenume'
A_sir	"Nume prenume"

Figura 5 Același text stocat ca un vector de caractere (*A_char*) și ca un șir de caractere (*A_sir*)

1.6.4 Variabile numerice

Există mai multe tipuri de variabile numerice folosite în Matlab. În mod implicit, Matlab lucrează cu variabile numerice memorate în format virgulă flotantă de tip *double* – acestea stochează valori numerice cuprinse între limitele $-1.7977e+308$ și $1.7977e+308$ și sunt reprezentate în memorie pe 8 octeți (Bytes), adică 64 bits pentru memorarea unui număr.

Alte tipuri de variabile numerice ce pot fi utilizate în Matlab sunt:

- *single* - Stochează valori numerice cuprinse între -3.4×10^{38} și 3.4×10^{38} . Reprezentarea internă a unui număr se face pe 4 octeți (32 bits).
- *uint8* - Stochează valori numerice cuprinse între 0 și $2^8 - 1$. Reprezentarea internă a unui număr se face pe 1 octet.
- *uint16* - Stochează valori numerice cuprinse între 0 și $2^{16} - 1$. Reprezentarea internă a unui număr se face pe 2 octeți.
- *uint32* - Stochează valori numerice cuprinse între 0 și $2^{32} - 1$. Reprezentarea internă a unui număr se face pe 4 octeți.
- *uint64* - Stochează valori numerice cuprinse între 0 și $2^{64} - 1$. Reprezentarea internă a unui număr se face pe 8 octeți.
- *int8* - Stochează valori numerice cuprinse între -2^7 și $2^7 - 1$. Reprezentarea internă a unui număr se face pe 1 octet.
- *int16* - Stochează valori numerice cuprinse între -2^{15} și $2^{15} - 1$. Reprezentarea internă a unui număr se face pe 2 octeți.
- *int32* - Stochează valori numerice cuprinse între -2^{31} și $2^{31} - 1$. Reprezentarea internă a unui număr se face pe 4 octeți.
- *int64* - Stochează valori numerice cuprinse între -2^{63} și $2^{63} - 1$. Reprezentarea internă a unui număr se face pe 8 octeți.

Variabilele numerice pot stoca atât numere reale cât și numere complexe. Numerele complexe sunt definite prin specificarea părții reale și a părții imaginare, după cum este arătat în Exemplul 1-6 și Figura 6.

Exemplul 1-6 Definirea numerelor complexe

```
>> a=5+6*i
a =
5.0000 + 6.0000i
```

→ definirea numărului complex unde partea reală are valoare 5 iar partea imaginară are valoarea 6

→ afișarea rezultatului în fereastra de comandă

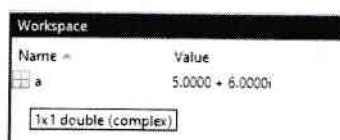


Figura 6 Stocarea numerelor complexe în Workspace

Numerele complexe pot fi descompuse în parte imaginară și parte reală folosind funcțiile matlab `real` și `imag`. De exemplu, descompunerea numărului complex $a=5+6i$ (definit în Exemplul 1-6), se face după cum este arătat în :

Exemplul 1-7 Descompunerea numerelor complexe

```
>> a_real=real(a)
a_real =
5

>> a_imaginar=imag(a)
a_imaginar =
6
```

→ extragerea părții reale a numărului complex $a=5+6i$

→ afișarea rezultatului în fereastra de comandă

→ extragerea părții imaginare a numărului complex $a=5+6i$

→ afișarea rezultatului în fereastra de comandă

1.6.5 Variabile celule – tablouri de celule

Un tablou de celule reprezintă un tip de date format din mai multe celule indexate (rând și coloană). Fiecare celulă poate să conțină variabile numerice, text, combinații de numere și text, tablouri numerice (matrice) de diferite dimensiuni.

Crearea unui tablou de celule se face prin utilizarea acoladelor (Exemplul 1-8). Conținutul celulelor este accesat prin utilizarea numelui celulei și a unui index (rând, coloană) scris în acolade (Exemplul 1-8). Apelarea unei celule din tabloul de celule se realizează prin utilizarea numelui celulei și a unui index (rând, coloană) scris în paranteze rotunde (Exemplul 1-8).