



## 7. PROGRAMUL MATLAB -un mediu ideal de calcul și simulare



### 7.1. Introducere

MATLAB (MATrix LABoratory) reprezintă un pachet de programe de calcul numeric și de reprezentare grafică, de înaltă performanță și acuratețe, cu o interfață sugestivă și ușor de folosit. Elementul de bază cu care operează acest program este *matricea*. MATLAB a suferit o dezvoltare permanentă de-a lungul anilor -concretizată în apariția de noi versiuni- devenind un *standard* în mediile universitare și în cercetarea științifică. Față de un *germene* de prelucrare matematică matriceală, acest program are secțiuni dedicate diverselor domenii științifice și tehnice (denumite *toolbox* -uri), care permit rezolvarea comodă a problematicii specifice.

O trecere în revistă a resurselor programului MATLAB evidențiază:

#### 1. Operații matematice fundamentale:

- ▶ calcule cu matrice și tablouri;
- ▶ operatori relaționali și logici;
- ▶ funcțiile trigonometrice, hiperbolice, exponențiale, logaritmice, etc.;
- ▶ funcții speciale (Gamma, Euler, Bessel, etc.);
- ▶ calcule cu polinoame;
- ▶ aritmetică în virgulă mobilă IEEE (15 cifre zecimale în rezultat).

#### 2. Algebră liniară și funcții matriceale:

- ▶ ecuații liniare;
- ▶ analiză matriceală;
- ▶ factorizare QR;
- ▶ descompunerea valorilor proprii;
- ▶ descompunerea valorilor singulare;
- ▶ generarea matricelor particulare;
- ▶ algoritmi pentru matrice structurate sau rare.

**3. Analiza datelor și transformate Fourier:**

- ▶ corelație, covarianță, diferențe finite;
- ▶ media statistică, mediană, deviație standard;
- ▶ transformata Fourier rapidă (FFT);
- ▶ interpolarea datelor.

**4. Analiză numerică neliniară:**

- ▶ ecuații diferențiale;
- ▶ integrare numerică;
- ▶ calculul rădăcinilor ecuațiilor neliniare;
- ▶ minimizare.

**5. Programare:**

- ▶ structuri de control;
- ▶ citirea/scrierea din/în fișiere cu format ASCII sau binar;
- ▶ depanarea fișierelor program;
- ▶ crearea de interfețe grafice interactive.

**6. Reprezentarea graficelor 2D și 3D:**

- ▶ folosirea pixelilor, liniilor, poligoanelor pline, liniilor de contur;
- ▶ folosirea diagramelor polare, cu bare sau a histogramelor;
- ▶ reprezentare în coordonate liniare, semilogaritmice sau logaritmice;
- ▶ reprezentarea suprafețelor.

**7. Vizualizarea graficelor:**

- ▶ controlul culorilor și a poziției surselor de lumină;
- ▶ vizualizări volumetrice;
- ▶ afișarea imaginilor;
- ▶ animație.

**8. Proprietăți și resurse grafice:**

- ▶ crearea și manipularea figurilor;
- ▶ desenarea, scalarea, etichetarea și colorarea axelor;
- ▶ desenarea și manipularea liniilor și textului.

**9. Utilizarea interfeței grafice:**

- ▶ crearea de meniuri pe bara superioară (pull-down) și iconițe (pop-up);
- ▶ crearea de butoane selectabile prin apăsare (push buttons), butoane radio (radio buttons), cutii de dialog (check boxes) și cursoare liniare (sliders);
- ▶ cutii de dialog cu texte editabile sau fixe;
- ▶ utilizarea mouse-ului pentru detectarea unui eveniment sau a unei condiții impuse, sau pentru lansarea unei comenzi.

**10. Schimbul de documente:**

- ▶ importul și exportul de documente în/din format ASCII, binar sau hexa;
- ▶ transmitere de grafice în formatul standard META;
- ▶ compatibilitate cu formatul Clipboard.

Este de asemenea foarte important că, orice utilizator poate să insereze în programul MATLAB propriile fișiere (cu extensia “.m”), creindu-se astfel posibilitatea de realizare de proprii aplicații.

Principalele toolbox -uri prezente în cadrul acestui program sunt:

**1. SIGNAL PROCESSING** este destinat procesării semnalelor și a seriilor temporale și conține: proiectarea și implementarea filtrelor analogice și numerice; analiza și estimarea spectrului; simularea răspunsului filtrelor; transformate FFT, DCT, etc.;

modelare parametrică; modulare și demodulare.

**2. IMAGE PROCESSING** prelucrează imagini și semnale bidimensionale folosind: proiectarea filtrelor 2D și filtrarea semnalelor bidimensionale; reconstituirea și extragerea imaginilor; operații de colorare, geometrie și morfologie; transformări bidimensionale; analiza și statistica imaginilor.

**3. SYMBOLIC MATH** folosește pachetul software MAPLE V și conține proceduri pentru: calcul simbolic pentru analiza problematicii algebrei liniare; simplificarea simbolică a expresiilor; evaluarea simbolică a funcțiilor matematice speciale;

**4. NEURAL NETWORK** este destinat proiectării și simulării rețelelor neurale și conține: funcție de transfer de tip sigmoidă, liniară, limită și concurențială; rețele asociative, cu punct de oprire, cu transformare de caracteristici, cu autoorganizare, Hopfield, Kohonen, Widrow -Hoff; arhitecturi cu estimări recurente, succesive; funcții pentru analiza grafică a rezultatelor; straturi, elemente și conexiuni nelimitate.

**5. STATISTICS** conține funcții pentru analiza, modelarea și simularea datelor: analiză grafică interactivă; distribuțiile beta, binomială, hi-pătrat, Poisson, etc.; generarea numerelor aleatoare; calcule de regresie, polinomiale și de varianță; descrieri statistice; analiza interactivă a liniilor de contur.

**6. SPINE** realizează interpolări ale datelor prin: interpolări polinomiale; funcții spine de interpolare; aproximarea curbilor și suprafețelor prin netezire; calculul derivatei, integralei și evaluarea funcțiilor.

**7. CONTROL SYSTEM DESIGN** folosește la analiza și sinteza sistemelor automate: sisteme cu timp continuu și discret; exprimări pe stare sau pe funcții de transfer și conversii între modele; conexiuni ale sistemelor liniare; răspunsul în frecvență și reprezentări Bode, Nyquist, Nichols, SVD; locul geometric al rădăcinilor pentru analiza și sinteza SRA.

**8. ROBUST CONTROL** conține programe pentru analiza și sinteza sistemelor automate la care procesele conduse sunt tarate de incertitudini, îndeosebi cele ce afectează însăși structura acestora (incertitudini *nestructurate*); astfel sunt vizate: sinteza optimală LQG/LTR; sinteza  $H_2$  sau  $H_\infty$ ; reducerea dimensională pe baza

factorizării Hankel.

**9. SYSTEM IDENTIFICATION** grupează proceduri pentru tratarea semnalelor și a seriilor de timp pentru a furniza modele ale proceselor ce fac obiectul conducerii automate. În acest sens avem: modelarea MA, AR, ARMA și Box-Jenkins; modele de stare sau pe funcții de transfer; selectarea automată sau manuală a ordinului modelului; analiză spectrală.

**10.  $\mu$ -ANALYSIS AND SYNTHESIS** se constituie din programe particulare de proiectare a sistemelor cu incertitudini:  $\mu$ -analiză și sinteză; sinteză  $H_2$  sau  $H_\infty$ .

**11. OPTIMISATION** abordează sinteza optimală a proceselor statice sau dinamice folosind: programare liniară și programare pătratică; determinarea minimumului (sau maximumului) pentru funcții obiectiv generale; metoda celor mai mici pătrate pentru rezolvarea ecuațiilor neliniare; optimizare multiobiectiv.

În această lucrare se prezintă programul MATLAB4 (adică versiunea 4) sub WINDOWS; trebuie înțeles că versiunile anterioare sunt acceptate - practic în totalitate - de această versiune, pe când programele create cu versiunile ulterioare au puține șanse de a fi recunoscute de programul prezentat aici.

## 7.2. Funcții de control general

Se pot folosi următoarele funcții pentru *controlul acestui mediu de programare*:

<i>help</i>	Furnizează informații "on-line" despre MATLAB și funcțiile acestuia;
<i>what</i>	Listează fișierele ".m", ".mat" și ".mex" din directorul curent;
<i>type</i>	Listează fișierul ".m" menționat;
<i>lookfor</i>	Returnează numele fișierelor ce au în prima linie cuvintele date ca argument;
<i>which</i>	Returnează calea în care este fișierul sau funcția MATLAB;
<i>path</i>	Returnează căile cu care lucrează MATLAB-ul;
<i>who</i>	Listează variabilele curente din memorie;
<i>whos</i>	Listează variabilele curente, dimensiunile lor, precum și tipul acestora (reale, complexe);
<i>exist</i>	Verifică existența variabilei menționate în mediul MATLAB.

Pentru *controlul directoarelor, al fișierelor și al sistemului de operare* se folosesc:

<i>cd</i>	Returnează sau schimbă directorul curent;
<i>dir</i>	Listează conținutul directorului curent;
<i>delete</i>	Șterge de pe disc fișierul precizat;
<i>!</i>	Execută comenzi din sistemul de operare DOS, fără a părăsi MATLAB-ul;
<i>diary</i>	Salvează sesiunea de lucru într-un fișier text (ASCII).

Pentru controlul *mărimilor de timp* se dispune de:

<i>clock</i>	Funcție pentru citirea ceasului calculatorului;
<i>cputime</i>	Idem pentru determinarea timpului de calcul al "CPU";
<i>date</i>	Idem pentru citirea datei din calculator;
<i>etime</i>	Cronometrează timpul dintre două evenimente;
<i>tic, toc</i>	Funcții pentru pornirea și respectiv oprirea cronometrului.

Funcțiile destinate *controlului ferestrei de comenzi* sunt:

<i>clc</i>	Șterge fereastra de comenzi;
<i>home</i>	Mută cursorul în poziția inițială;
<i>format</i>	Setează formatul de afișare a datelor;
<i>echo</i>	Permite afișarea liniilor de program în timpul execuției acestora;
<i>more</i>	Funcție pentru controlul numărului de linii afișate pe monitor.

Se menționează că formatul de afișare a datelor este

Opțiune	Rezultat	Exemplu pentru 1024/13
short	4 zecimale	78.7692
long	14 zecimale	78.76923076923077
short e	5 cifre + exponent	7.8769E+001
long e	16 cifre + exponent	7.876923076923077E+001
hex	hexazecimal	4053b13b13b13b14
plus	semnul (plus sau minus)	+
bank	două zecimale (bancar)	78.77
rat	exprimare rațională	1024/13

Pentru *controlul variabilelor din memorie* se folosesc următoarele funcții:

<i>disp</i>	Afișează o matrice sau un text;
<i>clear</i>	Șterge variabile și funcții;
<i>pack</i>	Compactează datele din memorie;
<i>size</i>	Returnează dimensiunile unei matrice;
<i>length</i>	Returnează dimensiunea unui vector.

Programul MATLAB folosește următoarele *variabile speciale* (care nu pot fi declarate de utilizator)

<i>ans</i>	Variabilă creată automat pentru un rezultat de calcul, când nu i s-a asignat un alt nume;
<i>eps</i>	Variabilă permanentă în care este memorată eroarea relativă pentru calculele efectuate în virgulă mobilă -valoarea implicită este $\text{eps}=2.2204\text{e}-016$ ;
<i>pi</i>	Variabilă permanentă asignată cu valoarea 3.14159265358;

$i = \sqrt{-1}$	Variabilă permanentă asociată unității imaginare și folosită la exprimarea numerelor complexe în forma $z=2+5*i$ ; ea poate fi exprimată și sub forma unei variabile oarecare, ca de exemplu $p=\text{sqrt}(-1)$ , numărul anterior primind scrierea $z=2+5*p$ ;
$j$	Idem ca și $i$ ;
$inf$	Variabilă pentru reprezentarea lui <i>plus infinit</i> în aritmetica IEEE, ca rezultat al împărțirii unui număr nenul prin zero (de exemplu $5/0.0=inf$ );
$NaN$	Variabilă folosită în aritmetica IEEE pentru rezultatul împărțirii nedefinite $0.0/0.0=NaN$ (Not -a- Number);
$nargin$	Variabilă permanentă pentru testarea numărului argumentelor de intrare la apelarea unei funcții;
$nargout$	Variabilă permanentă pentru testarea numărului argumentelor de ieșire ale unei funcții;
$flops$	Returnează numărul de operații în virgulă mobilă efectuate într-o procedură folosită de utilizator, aducerea la zero a contorului făcându-se cu $flops(0)$ ;
$computer$	Variabilă folosită pentru obținerea informațiilor referitoare la tipul calculatorului utilizat și la numărul maxim de elemente pe care le poate gestiona versiunea de MATLAB folosită de utilizator;
$realmax$	Reprezintă cea mai mare valoare pozitivă în virgulă mobilă ce poate fi folosită în calcule, respectiv $1.7977e+308$ ;
$realmin$	Idem pentru cea mai mică valoare $2.2251e-308$ ;
$isieee$	Funcție ce returnează 1 dacă calculatorul folosește aritmetica IEEE și 0 în caz contrar;
$version$	Funcție pentru determinarea versiunii de MATLAB și a toolbox -urile instalate pe calculator;
$ver$	Idem.

Pentru *importul și exportul fișierelor de date* se folosesc următoarele funcții:

$load$	Încarcă variabilele dintr-un fișier de date de pe disc;
$save$	Salvează variabilele într-un fișier de date pe disc.

### 7.3. Elemente fundamentale de programare în MATLAB

**1. Matrice.** Programul MATLAB utilizează numai elementul “*matrice*”; în acest sens, scalarii sunt asimilați cu o matrice având o linie și o coloană. Introducerea unei matrice se face introducând elementele pe linii, componentele liniilor fiind separate prin “,” sau “Blanc” (spațiu), distincția dintre linii făcându-se cu “;” sau “Enter”; toate acestea se localizează între paranteze drepte “[ ]” (prin excepție, scalarii se introduc fără paranteze). Elementele unei matrice pot fi numere (reale sau complexe) sau expresii aritmetice introduse explicit de utilizator; în egală măsură matricele pot să rezulte în urma efectuării calculelor sau din încărcarea unor fișiere de pe disc. Localizarea unui element al unei matrice se face specificând linia și coloana pe care se

situează acesta. Extragerea unei submatrice se face specificând liniile și coloanele ce o localizează; de exemplu pentru matricea a cu 6 linii și 8 coloane, cu  $b=a(2:5,4:7)$  se obține submatricea localizată între liniile 2 și 5 (inclusiv), respectiv între coloanele 4 și 7 (inclusiv).

**2. Declarații și variabile.** MATLAB-ul este un program de expresii, introduse pe linii de program, care sunt interpretate și executate în mod succesiv (dacă prin instrucțiuni nu se constituie alte căi). Uzual instrucțiunile sunt de forma

variabilă=expresie

sau pur și simplu

expresie

pentru ultimul caz, MATLAB-ul îi atașează automat variabila “ans”; În scrierea *expresiilor* se folosesc operatori, caractere speciale, funcții matematice (cu sintaxa proprie a programului MATLAB) sau nume de variabile. Dacă o expresie nu încapă pe un rând de program, prin simbolul “...” (trei puncte) se comunică programului continuarea acesteia pe linia următoare. Numele de variabile sau funcții trebuie să înceapă obligatoriu cu o literă, urmată, eventual de alte litere, cifre sau linie de subliniere “\_”. De reținut că acest program face distincție între caracterele mari și cele mici (mod *senzitiv*); de exemplu “A” și “a” sunt *variabile diferite*. Folosind instrucțiunea

casesen off

se trece în modul *nesenzitiv* (nu se mai face distincție între caractere mari și mici); revenirea se face cu “casesen on”.

**3. Numere și expresii numerice.** Programul MATLAB folosește *notația zecimală convențională*, cu *punct zecimal*; puterea este un factor de scală inclus ca sufix (de exemplu 9587.5462 se poate scrie 9.5875462e+03). Se folosesc următorii operatori aritmetici (valabili și pentru matrice)

+	Adunarea;
-	Scăderea;
*	Înmulțirea;
/	Împărțirea la dreapta ( $4/5=0.8$ );
\	Împărțirea la stânga ( $5\backslash 4=0.8$ );
^	Ridicarea la putere.

Parantezele rotunde precizează ordinea operațiilor din cadrul expresiilor.

**4. Structura programelor MATLAB.** MATLAB-ul lucrează fie în modul “linie de comandă” (liniile de program se prelucrează succesiv și rezultatele se afișează imediat), fie cu “programe conținute în fișiere”, toate cu extensia “.m”. Se folosesc următoarele tipuri de fișiere:

a) *Fișierele script* sunt fișiere externe care conțin secvențe de comenzi MATLAB; prin apelarea numelui fișierului se execută secvența conținută de acesta și rezultatele rămân în zona de memorie a aplicației. De reținut că fișierele “script” nu

lucrează cu argumente.

b) *Fișiere funcție* sunt acele fișiere M la care prima linie conține cuvântul “*function*”; e important ca aceste subprograme pot lucra cu argumente precizate în titlu, permițând scrierea de programe structurate. La terminarea rulării în memoria calculatorului rămân însă numai variabilele de ieșire. Forma generală pentru prima linie a unui astfel de fișier este

```
function[param_ieșire]=nume_funcție(param_intrare)
```

în care argumentele reprezintă “parametri de ieșire”, respectiv “parametri de intrare”.

**5. Comentarii și help-ul în programele MATLAB.** Un comentariu se poate introduce după simbolul “%” pe o linie întreagă de program sau după o instrucțiune și, se subînțelege că, ceea ce urmează după acest simbol este omis de compilator. Comentariile pot fi puse pe orice linie a programului dar, important, liniile cu comentarii aflate imediat după linia de declarare a funcției (așa numitele *linii H* ce țin până la prima linie liberă) vor fi afișate explicit atunci când se cer informații despre programul respectiv cu instrucțiunea

```
help nume_funcție
```

Liniile de program încheiate cu “;” nu vor afișa rezultatele de calcul. Funcția *help* vă permite să aflați toate informațiile despre orice funcție a MATLAB-ului (inclusiv modul de apelare și funcțiile înrudite), *dacă se cunoaște numele funcției*. De aceea, în finalul acestui capitol, se dă -în ordine alfabetică- cvasitotalitatea funcțiilor de care dispune acest program.

#### 7.4. Instrucțiuni și funcții de control

Programul MATLAB folosește următoarele *instrucțiuni de control logic*:

<i>if</i>	Instrucțiune pentru execuția condiționată;
<i>else</i>	Instrucțiune asociată cu “ <i>if</i> ”;
<i>elseif</i>	Instrucțiune asociată cu “ <i>if</i> ”;
<i>for</i>	Instrucțiune pentru crearea de cicluri cu număr specificat de pași;
<i>while</i>	Instrucțiune pentru crearea de cicluri cu condiție logică;
<i>break</i>	Instrucțiune pentru terminarea forțată într-un ciclu;
<i>return</i>	Returnează execuția la funcția precizată;
<i>error</i>	Instrucțiune pentru afișarea unui mesaj de eroare;
<i>end</i>	Instrucțiune pentru încheierea ciclurilor “ <i>for</i> ”, “ <i>while</i> ” și “ <i>if</i> ”.

Instrucțiunile condiționale apelează la *operatori relaționali* și *operatori logici* prezentați mai jos. Operatorii relaționali compară două matrice sau două expresii matriceale element cu element și furnizează o matrice de aceeași dimensiune având elemente 1 (pentru *adevărat*), respectiv 0 (pentru *fals*). Operatorii logici compară matrici cu elemente 1 sau 0 obținute cu operatorii relaționali și furnizează un rezultat de tipul “*da*” sau “*nu*”.

Op. relaționali	Semnificație	Op. logici	Simbol MATLAB	Prioritate
<	mai mic	NU	~	1
<=	mai mic sau egal	ȘI	&	2
>	mai mare	SAU		3
>=	mai mare sau egal			
==	identic			
~=	diferit			

O formă generală a instrucțiunilor prezentate este dată în continuare (și evident, ele pot fi combinate între ele)

<i>if</i> expresie_logică_1 grupul de instrucțiuni_A <i>elseif</i> expresie_logică_2 grupul de instrucțiuni_B <i>elseif</i> expresie_logică_3 grupul de instrucțiuni_C <i>else</i> grupul de instrucțiuni_D <i>end</i>	<i>for</i> index=expresie grup_de_instrucțiuni <i>end</i>
	<i>while</i> expresie grup_de_instrucțiuni <i>end</i>

Programul MATLAB dispune de următoarele *funcții de control logic*:

<i>exist</i>	Verifică dacă variabilele sau funcțiile argument sunt definite;
<i>any</i>	Testează dacă cel puțin un element al matricei verifică o condiție logică;
<i>all</i>	Testează dacă toate elementele matricei verifică o condiție logică;
<i>find</i>	Returnează indicii elementelor diferite de zero;
<i>isnan</i>	Testează dacă elementele matricei sunt NaN;
<i>isinf</i>	Testează dacă elementele matricei sunt infinite;
<i>finite</i>	Testează dacă elementele matricei sunt finite.

Este de bun augur pentru utilizator faptul că programul MATLAB permite *vectorizarea tuturor calculelor* ceea ce simplifică structura programelor și crește spectaculos viteza de execuție a acestora. Sensul termenului de *vectorizare* este acela că, în mediul de programare MATLAB, toate funcțiile se pot aplica matricelor; de exemplu, dacă  $x \in \mathbb{R}^{m \times n}$ ,  $x_s = \cos(x)$  returnează matricea rezultată în urma aplicării funcției trigonometrice menționate fiecărui element al tabloului  $x$  și astfel se evită o scriere de program cu două cicluri “*for*”.

### 7.5. Funcții matematice uzuale

Pentru *aproximarea numerelor reale* se folosesc:

<i>ceil</i>	Returnează un număr întreg, rotunjit la cel mai apropiat întreg spre $+\infty$ ;
<i>fix</i>	Returnează un număr întreg, rotunjit la cel mai apropiat întreg spre zero;
<i>floor</i>	Returnează un număr întreg, rotunjit la cel mai apropiat întreg spre $-\infty$ ;
<i>round</i>	Returnează un număr întreg, rotunjit la cel mai apropiat întreg;
<i>rem</i>	Returnează restul împărțirii argumentelor;
<i>rat</i>	Returnează aproximarea unui număr cu fracții raționale continue;
<i>rats</i>	Returnează aproximarea unui număr cu numere raționale;
<i>sign</i>	Returnează semnul argumentului.

La operațiile cu *numere complexe* se folosesc *funcțiile*:

<i>abs</i>	Calculează modulul numărului complex dat ca argument al funcției;
<i>angle</i>	Calculează faza numărului complex dat ca argument al funcției;
<i>unwrap</i>	Calculează parte reală și imaginară a numărului complex dat în forma polară;
<i>conj</i>	Calculează conjugatul complex al numărului dat ca argument al funcției;
<i>imag</i>	Extrage partea imaginară a argumentului funcției;
<i>real</i>	Idem partea reală;.

*Funcțiile algebrice* uzuale se regăsesc în MATLAB în forma:

$\wedge$	Funcția putere ( $a^n$ este scrierea pentru $a^n$ );
<i>exp</i>	Funcția exponențială ( $\exp(x)$ exprimă operația $e^x$ );
<i>log</i>	Calculează logaritmul natural al argumentului ( $\ln$ );
<i>log2</i>	Calculează logaritmul în bază 2 ( $\log_2$ );
<i>log10</i>	Calculează logaritmul în bază 10 ( $\log_{10}$ );
<i>nextpow2</i>	Pentru argumentul p determină cel mai mic n ce asigură $ p  \leq 2^n$ ;
<i>pow2</i>	Calculează puterile lui 2 ( $2^n$ );
<i>sqrt</i>	Funcția radical de ordinul 2.

Funcțiile *trigonometrice directe și inverse* (argumentele se exprimă în radiani!), precum și cele *hiperbolice* se exprimă natural prin:

Funcții trigonometrice				Funcții hiperbolice			
directe		inverse		directe		inverse	
<i>sin</i>	<i>cot</i>	<i>asin</i>	<i>acot</i>	<i>sinh</i>	<i>coth</i>	<i>asinh</i>	<i>acoth</i>
<i>cos</i>	<i>sec</i>	<i>acos</i>	<i>asec</i>	<i>cosh</i>	<i>sech</i>	<i>acosh</i>	<i>asech</i>
<i>tan</i>	<i>csc</i>	<i>atan</i>	<i>acsc</i>	<i>tanh</i>	<i>csch</i>	<i>atanh</i>	<i>acsch</i>

Ca un element de particularitate, funcția *atan2(y,x)* calculează argumentul numărului complex  $x+iy$  furnizând valori din intervalul centrat  $[-\pi, +\pi]$ .

Tabelul 7.1. Indexul funcțiilor MATLAB

abs	clf	disp	gammainc	isletter	
acos	clock	drawnow	gammaIn	isnan	
acosh	close		gca	isstr	
acot	colormap	echo	gcd		
acoth	ColorSpec	eig	gcf	j	
acsc	comet	else	get	jet	
acsch	comet3	elseif	getframe		
all	compan	end	ginput	kron	
and	compass	eps	gradient		
angle	computer	erf	gray	lcm	
ans	cond	erfc	grid	length	
any	condest	erfinv	griddata	line	
asec	conj	error	gtext	linspace	
asech	contour	errorbar		load	
asin	contourc	etime	hadamard	log	
asinh	contrast	eval	hankel	log10	
atan	conv	exist	help	log2	
ata2	cool	exit	hex2dec	loglog	
atanh	copper	exp	hidden	logspace	
axes	corrcoef	eye	hilb	lookfor	
axis	cos		hist	lower	
	cosh	feather	hold	lu	
bar	cot	feval	home	magic	
bessel	coth	figure	hot	matlabrc	
besseli	cov	fill	hsv	max	
besselj	cputime	fill3	hsv2rgb	mean	
besselk	csc	find		median	
bessely	csch	findstr	i	mesh	
beta	cumprod	finite	if	meshc	
betainc	cumsum	fix	imag	meshgrid	
betaln	cylinder	flag	image	meshz	
blanks		fliplr	Inf	min	
bone	date	flipud	int2str	more	
break	deblank	floor	interp1	movie	
brighten		dec2hex	flops	interp2	moviein
	deconv	fmin		interp3	
caxis	del2	fmins		interp4	NaN
cd	delete	for		interp5	nargin
ceil	det	format		interpft	nargout
chol	diag	fplot		inv	newplot
cla	diary	function		invhilb	nextpow2
clabel	diff	fzero		ishold	norm

clc	diffuse		isieee	not	
clear	dir	gamma	isinf	num2str	
ode23	quad		strcmp	uigetfile	
ode45	quad8	save	strings	uimenu	
ones	quit	script	subplot	uiputfile	
or	quiver	sec	sum	uisetcolor	
orient	qz	sech	surf	uisetfont	
		semilogx	surface	unwrap	
pack	rand	semilogy	surfc	upper	
pascal	randn	set	surfl		
path	rank	setstr	surfnorm	vander	
pcolor	rat	shading	svd	ver	
pi	rats	sign		version	
pink	rcond	sin	table1	view	
pinv	real	sinh	table2		
plot	realmax	size	tan	what	
plot3	realmin	sort	tanh	which	
polar	rem	specular		text	while
poly	reset	sphere	tic	who	
polyder	reshape	spinmap	title	whos	
polyfit	residue	spline	toc	wilkinson	
polyval	return	sprintf	toeplitz		
pow2	rgb2hsv	sqrt	trace	xlabel	
print	rgbplot	stairs	trapz		
printopt	roots	startup	tril	ylabel	
prism	rose	std	triu		
prod	rot90	steam	type	zeros	
	round	str2mat		zlabel	
qr	ref	str2num		uicontrol	