

# Introducere în MATLAB

MATLAB este unul dintre cele mai răspândite programe, în special în teoria reglării automate, pentru calculul științific și numeric. Pe lângă calculul efectiv, MATLAB oferă și posibilități de reprezentare grafică și programarea propriilor rutine. Este vorba de un sistem interactiv cu matricea ca element fundamental. Denumirea MATLAB provine de la *Matrix Laboratory*. SIMULINK este un mediu de operare grafic, bazat pe MATLAB prin intermediul căruia pot fi modelate și simulate sisteme complexe. Calculele matriceale necesare sunt realizate automat de MATLAB. Dezvoltatorul ambelor produse este firma The Math Works Inc., SUA. Scopul acestei lucrări este o familiarizare cu mediul de lucru MATLAB. Instrucțiunile de lucru sunt redactate pe baza versiunii MATLAB 7.11.0 (R2010b). În cazul în care lucrați cu o versiune diferită pot apărea modificări în structura meniurilor și a căsuțelor de dialog.

## 1 Primii pași

### Fereastra de comandă

După lansarea MATLAB apare spațiul de lucru MATLAB (vezi 1). Acesta este compus din mai multe ferestre, dintre care fereastra de comenzi (**Matlab Command Window**) cu linia de comandă MATLAB `>>`, ocupă cea mai mare suprafață. Fereastra de comenzi este utilizată la introducerea de variabile și rularea comenzilor, funcțiilor și rutinelor. Fiecare rând de comandă este terminat cu un Return  $\leftrightarrow$  iar expresia este evaluată imediat. Exemplele prezentate în acest document încep după simbolul liniei de comandă. Comentariile, cele care au în față simbolul procent `%`, nu trebuie introduse. Încercați să rulați următoarele exemple:

```
>> 5 * 7           % Atribuirea variabilei standard ans
>> a = 3 * pi      % Atribuirea variabilei a
>> b = a/5
>> c = a * b
>> b = 1 : 3 : 20
>> plot(b)
>> help cos
>> b = 2*cos(b)
```

Ultimul rând produce unul dintre cele mai des întâlnite mesaje de eroare: **missing operator, comma, or semi-colon**. În acest caz lipsește operatorul `*` pentru înmulțirea dintre 2 și cosinus. Comanda corectă are forma: `>> b = 2*cos(b)`

Afișarea rezultatului pe ecran poate fi suprimată print adăugarea unui `;` la sfârșitul rândului. Testați această funcționalitate prin repetarea comenzii cu `;`.

În locul reintroducerii comenzii de la tastatură, poate fi folosită și săgeata sus  $\uparrow$ , prin care se pot accesa rapid comenzile anterioare. Aceste comenzi pot fi editate înainte de apăsarea tastei Return  $\leftrightarrow$ . Se poate opera și mai rapid atunci când este cunoscută prima literă a comenzii căutate. La introducerea acesteia în linia de comandă, urmată de apăsarea tastei  $\leftrightarrow$ , vor fi afișate numai comenzile care încep cu această literă.

Ce comandă este afișată la introducerea **p**  $\uparrow$ ?

### Paleta de simboluri a ferestrei de comenzi

Paleta de simboluri (vezi fig. 1) permite accesul rapid la funcționalități des utilizate ale mediului de dezvoltare. Semnificația simbolurilor individuale este descrisă de la stânga la dreapta:

- Pornirea editorului MATLAB într-o fereastră nouă

Această operație este similară cu selectarea din meniu a **File:New:M.File**. Editorul precum și realizarea respectiv modificarea unor rutine proprii, vor fi prezentate ulterior.

- Deschiderea unui document MATLAB existent în editorul MATLAB

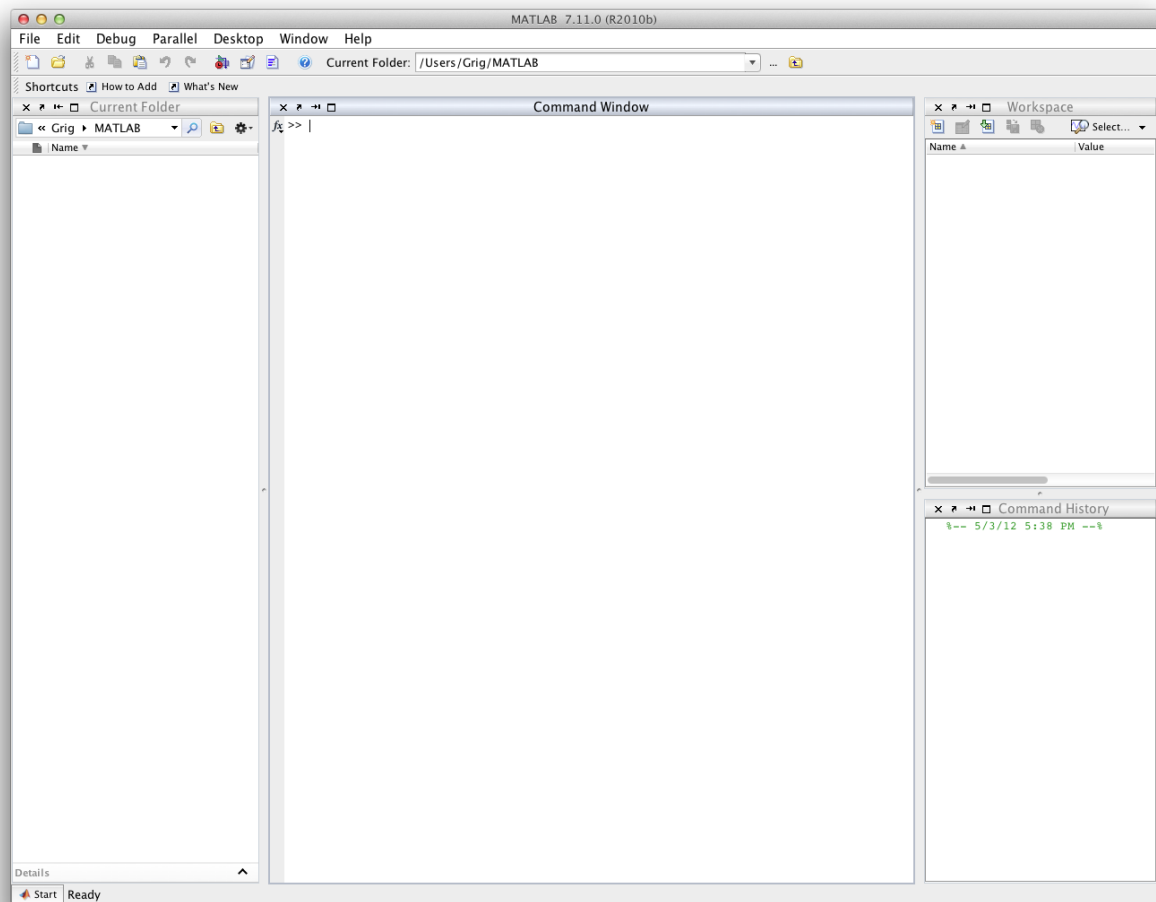


Figure 1: Spațiul de lucru MATLAB

Este afișată o selecție de fișiere cu extensiile: **.m** pentru rutine MATLAB, **.mdl** pentru modele SIMULINK sau **.fig** pentru figuri MATLAB. Editorul permite însă și modificarea oricăror fișiere ASCII.

- Decupare, copiere și lipire

- *Undo* și *Redo*

- Lansare SIMULINK

- Lansare GUIDE

Un editor de interfețe grafice cu utilizatorul

- Lansare Profiler

pentru optimizarea fișierelor M

- Deschiderea ferestrei de ajutor

Fereastra de ajutor conține același text referitor la o funcție arbitrară *fname* precum cel care apare la introducerea **help fname** în linia de comandă. Textul nu apare însă în linia de comandă ci într-o fereastră dedicată, ceea ce permite navigarea ușoară între diferite comenzi.

- Calea actuală

Aici sunt căutate fișierele necesare și sunt salvate rezultatele operării. Dacă o căutare nu are succes în calea actuală este continuată la celelalte adrese din calea de căutare. Lista de căi poate fi afișată și modificată din meniul **File - Set Path**. În acest context, sunt date o serie de comenzi utile pentru operarea cu directoare:

```
>> cd      %Afișează calea actuală
>> cd ..   %Trecere în directorul superior
>> cd work  %Trecere în directorul inferior work
>> dir      %Listează toate fișierele din directorul actual
>> what     %Listează toate fișierele MATLAB/SIMULINK din directorul actual
```

### Ferestre suplimentare

Toate variabilele utilizate sunt salvate automat în fereastra **Workspace**. Aici sunt afișate și dimensiunile și tipul acestor variabile. Printr-un dublu clic pe numele variabilei este afișat și poate fi modificat conținutul acesteia.

Fereastra **Current Directory** prezintă conținutul directorului de lucru actual într-o structură arborescentă. Sunt posibile operațiile uzuale de: deschidere, ștergere, redenumire și copiere.

În fereastra **Command History** este afișată o listă a ultimelor comenzi introduse. Printr-un dublu clic pe o comandă aceasta este rulată din nou.

### Ajutor

Se întâlnesc adesea situații în care este necesară descrierea și, în special, sintaxa unei funcții. Pentru aceasta este folosită comanda **help** (sau **doc** pentru detalii suplimentare) urmată de numele funcției:

```
>> help cos
>> help linspace
>> help plot
>> doc cos
>> lookfor plot
```

Comanda **lookfor** realizează o căutare după cuvântul introdus în primele rânduri ale tuturor fișierelor din calea de căutare.

Există mai multe modalități care pot fi utilizate pentru a obține ajutor referitor la introducerea comenzilor:

- Help Window

O fereastră care prezintă textul de ajutor aferent unei funcții (vezi paleta de simboluri). Prin dublu clic pe unul din elementele listei **Help Topics** poate fi parcurs un domeniu de interes. Fereastra de ajutor apare și la introducerea comenzii **helpwin** în linia de comandă.

- Help Desk

Fereastra Help Desk conține fișiere în format .html sau .pdf cu informații complete despre fiecare funcție. Aceasta apare și la introducerea comenzii **helpdesk** în linia de comandă.

- Exemple și demonstrații

Fereastra de demonstrații oferă o listă de exemple detaliate incluse cu mediul de dezvoltare MATLAB. Aceasta apare și la introducerea comenzii **demo** în linia de comandă.

### Exerciții

- 1.1. Modificați calea de căutare prin adăugarea unui director de lucru personalizat de forma **Nume\_Grupa**.
- 1.2. Descrieți pe scurt funcționalitatea și sintaxa comenzilor: **exp**, **grid**, **figure** și **hold**.
- 1.3. Prezentați funcționalitatea comenzilor: **which**, **pause** și **disp**.
- 1.4. Afișați cu ajutorul comenzii **linspace** o serie de 22 de numere de la 1.5 la 9.
- 1.5. Deschideți un fișier nou în editor și introduceți comenzile:

```
% Aceasta este prima incercare!  
cd  
disp('Salut!')  
a=5*7  
b=5*7;
```

Salvați fișierul în directorul de lucru cu numele **exemplu1.m**. Închideți editorul. Introduceți în linia de comandă comanda **exemplu1** și **help exemplu1.m**. Care este funcționalitatea noii comenzi?

Accesați **Current Directory**. Deschideți fișierul **exemplu1.m** prin dublu clic și apăsați tasta **F5**. (Astfel fișierul este salvat și rulat imediat.)

## 2 Constante și Variabile

### Constante

MATLAB a fost dezvoltat special pentru calculul matricial. O matrice reprezintă pentru MATLAB un câmp cu valori numerice. Cuvintele (*strings*) sunt câmpuri numerice, pentru care elementele conțin codurile ASCII ale caracterelor (valori întregi, pozitive). MATLAB cunoaște și alte metode pentru salvarea datelor alfanumerice, în această etapă vom rămâne însă la reprezentarea matricilor numerice.

Constantele (scalarii) sunt reprezentate intern ca matrice 1x1. Vectorii sunt matrice 1xn (vectori linie) sau nx1 (vectori coloană). În cazul calculelor matriciale (e.g. înmulțirea) trebuie ținut întotdeauna cont de ce se dorește a fi calculat și ce tip de vectori sunt folosiți. Altfel vor fi obținute mesaje de eroare sau rezultate care nu corespund cu intenția inițială.

Matricile cu mai mult de un element sunt introduse pe linii, încadrate în paranteze pătrate []. În interiorul unei matrici se folosesc virgule sau spații goale pentru separarea coloanelor și ; pentru separarea liniilor:

```
>> 5  
>> 3.14159  
>> 2.0e-10  
>> [1,2,3]      % [1 2 3] Vector linie
```

```
>> [4;5;6]      %  $\begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$  Vector coloană
```

```
>> A=[1,2,3;4,5,6;7,8,9]      %  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  Matrice
```

```
>> b=[1,2,3]
```

```
>> c='test'
```

Vectorii pot fi introduși și, mai rapid, în următoarea formă:

**vector** = [primul element:increment:ultimul element];

Vectorul astfel generat va conține elemente echidistante. Incrementul poate fi și negativ sau, în cazul în care este 1, poate fi omis.

```
>> [1:10]
```

```
>> [1:-1.5:-8]
```

Pentru a genera vectori, pot fi folosite și funcțiile **linspace** și **logspace**. **linspace** este cunoscută din secțiunea precedentă. Cum se diferențiază **logspace** de **linspace**?

### Variabile

Precum în orice alt limbaj de programare, MATLAB permite definirea de variabile. Atribuirea se face prin semnul egal =. Fiecare șir de caractere, ce începe cu o literă și nu include simboluri speciale, poate fi folosit ca nume de variabilă. Este făcută diferența între litere mari și litere mici. MATLAB nu necesită declarări inițiale; o variabilă este creată automat atunci când este plasată în partea din dreapta a unei alocări. În partea din dreapta pot fi utilizate numai variabile sau constante create anterior. Există posibilitatea alocărilor recursive.

Variabilele MATLAB se bazează pe matrici. Acestea pot conține atât cifre cât și litere. Pe lângă cifre și litere, mai există **celule** și **structuri**.

Analog ferestrei **Workspace**, prin comanda **who** + numele variabilei utilizate și **whos** + numele și dimensiunea variabilei utilizate, este afișat conținutul acestora în fereastra de comenzi.

```
>> b*A
```

```
>> d=2*A
```

```
>> d
```

```
>> who
```

```
>> whos
```

Suplimentar, există un număr de variabile de sistem rezervate, a căror modificare nu este permisă:

- **ans** conține rezultatul ultimei comenzi dacă nu a fost făcută nicio alocare către o altă variabilă.
- **pi** conține numărul  $\pi$ .
- **eps** conține numărul  $2^{-52} = 2.22e - 16$ .
- **realmin**, **realmax** sunt cele mai mici, respectiv cele mai mari valori reale disponibile în MATLAB.
- **Inf** infinit, rezultatul operației  $1/0$
- **NaN** (Not a Number) rezultatul operației  $0/0$
- **i,j** conțin ambele unitatea imaginară  $\sqrt{-1}$  pentru introducerea de numere complexe.

În cazul în care o variabilă de sistem *vname* este suprascrisă cu o altă valoare, aceasta poate fi resetată la valoarea originală prin comanda **clear vname**, care șterge variabila redefinită din spațiul de lucru.

```
>> pi=2.3
```

```
>> clear pi
```

```
>> pi
```

```
>> complex=3 + 2 * i
```

Dacă se dorește accesul la valori individuale ale elementelor unei matrici, numărul liniei și al coloanei corespunzătoare trebuie să fie puse între paranteze rotunde. Numărul liniei și al coloanei pot fi și vectori, pentru a accesa simultan mai multe linii sau coloane.

```
>> A(1,3)
>> A(1,:)    % ':' reprezintă linii complete
>> A(:,2)    % ':' reprezintă coloane complete
>> A([1 3],1)
```

Matricile logice ajută la extragerea de valori care îndeplinesc o serie de condiții, dintr-o matrice A. Suplimentar, matricea logică trebuie să aibă aceleași dimensiuni și constrângeri, e.g. valori între 2 și 5. Ulterior, aceste valori pot fi extrase. Exemplul următor ilustrează modul de lucru:

```
>> X=(A>=2&A<=5)
>> A(X)
```

Cu ajutorul unei matrici vide [] se pot șterge linii sau coloane dintr-o altă matrice:

```
>> A(:,2)=[]    Ștergerea celei de-a doua coloane a matricii A.
```

### Încărcare, salvare și ștergere

Următoarele comenzi servesc la administrarea variabilelor din spațiu de lucru și punerea lor la dispoziția utilizatorului pentru sesiuni ulterioare. Acestea sunt:

- **clear**: șterge toate variabilele din spațiul de lucru
- **clear *variable***: șterge *variable* din spațiul de lucru
- **save**: salvează toate variabilele în fișierul 'matlab.mat'
- **save *fname***: salvează toate variabilele în fișierul '*fname*.mat'
- **save *fname* x y z**: salvează variabilele x, y și z în fișierul '*fname*.mat'
- **load**: încarcă toate variabilele din fișierul 'matlab.mat' în spațiul de lucru
- **load *fname***: încarcă toate variabilele din fișierul '*fname*.mat' în spațiul de lucru
- **load *fname* x y z**: încarcă variabilele x, y și z din fișierul '*fname*.mat' în spațiul de lucru

### Exerciții

- 2.1. Creați două variabile  $\mathbf{t}=[2\ 4\ 6\ \dots\ 14]$ ,  $\mathbf{y}=\begin{bmatrix} 1 & 5 & 7 \\ 2 & 5 & \pi \end{bmatrix}$  și modificați-le conținutul printr-un dublu clic pe numele variabilei din fereastra corespunzătoare din spațiul de lucru. Verificați modificarea prin introducerea numelui variabilei în linia de comandă.
- 2.2. Modificați numele lui **y** în **ymat**.
- 2.3. Creați un vector linie **t1**, cu valori de la 0 la 1 și incrementul 0.01.
- 2.4. Creați un vector coloană **t2** cu o distribuție logaritmică și valori de la  $10^{-3}$  la  $10^0$ .
- 2.5. Extrageți prima linie **L1** și cea de-a treia coloană **C3** din **ymat**.
- 2.6. Extrageți matricea **ymic**, care conține prima și cea de-a treia coloană din **ymat**.
- 2.7. Extrageți vectorul **interm** cu valorile mai mari de 3 din matricea **ymat**.

2.8. Ștergeți cea de-a doua linie din **ymat**.

2.9. Salvați toate variabilele într-un fișier și ștergeți spațiul de lucru prin comanda **clear all**. Verificați conținutul spațiului de lucru (e.g. **who**). Încărcați variabilele din fișier înapoi în spațiul de lucru.

### 3 Operatori și funcții

#### Operatori

Partea din dreapta a unei expresii MATLAB poate conține o formulă algebrică complicată. Constantele și variabilele cunoscute trebuie conectate logic prin intermediul operatorilor. În cazul operatorilor aritmetici, pe lângă cei comuni (+, -, \*, /), mai sunt folosiți și ridicarea la putere (^) și transpunerea ('). Suplimentar, există și operatorii logici: negare ~ și &, sau |, sau exclusiv xor() și operatorii relativi: egalitate ==, inegalitate ~=, <, ≤, >, ≥. Prin următoarea comandă este afișată o imagine de ansamblu a operatorilor:

```
>> help ops
```

**Atenție:** Operatorii fac referire la calcule matriciale. Pentru a utiliza operatorii pe elementele individuale ale unei matrici, trebuie plasat un punct (.) în fața operatorului!

```
>> A=[1,2,3;4,5,6;7,8,0]
>> b      % Vector linie
>> b=b'    % Vector coloană
>> b+b     % Sumă de vectori
>> b'*b    % Produs scalar
>> b*b     % Eroare datorată dimensiunilor incompatibile
>> b.*b    % Multiplicare la nivel de element
>> 2*A     % Înmulțire cu un scalar
>> A*A     % A ori A și
>> A^2     % A la pătrat, sunt identice
>> A.^2    % dar nu și în acest caz!
```

#### Funcții

După cum s-a putut vedea din secțiunile anterioare, MATLAB include un număr mare de funcții din domenii variate. Apelarea unei funcții utilizează numele acesteia și un număr de parametri. Rezultatul poate fi stocat într-o variabilă. O imagine de ansamblu asupra funcțiilor matematice standard, oferă comanda **help elfun**.

```
>> help elfun
>> cos(0)
>> 4* atan(1)    % pi, atan(x) = tan-1(x)
>> bexp=exp(b)   % Funcția exponențială , aplicată fiecărui element din b
```

Două funcții foarte utile sunt **size** și **length**. **size** returnează numărul de linii și coloane al unei matrici și permite astfel diferențierea vectorilor linie de vectorii coloană. **length** returnează lungimea unui vector și, în cazul matricilor, valoarea mai mare dintre numărul de linii și cel al coloanelor.

```
>> A=[1,2,3;4,5,6]
>> size(A)
>> length(A)
>> size(b)
>> length(b)
```

Comanda **help elmat** afișează o imagine de ansamblu asupra matricilor standard și a funcțiilor de manipulare ale matricilor precum și asupra funcțiilor din domeniul algebrei liniare **help matfun**. Funcțiile **ones** și **zeros**, de exemplu, generează matrici numai cu elemente 1 resp. 0 și funcția **eye** generează matricea unitate.

```
>> help elmat  
>> D=ones(3,4)  
>> E=zeros(5)  
>> F=eye(4)  
>> G=eye(4,3)  
>> help matfun
```

### Exerciții

3.1. Determinați numărul de linii și de coloane ale matricilor D, E, F și G.