



Programarea si utilizarea calculatoarelor II

(1) Elemente de baza ale limbajului MATLAB

Autori: Calin Vaida, Bogdan Gherman, Doina Pisla

Introducere

Un **limbaj de programare** este un limbaj formal (care folosește anumite cuvinte cheie) folosit pentru a furniza **instrucțiuni** unui **echipament**. Primele limbaje de programare au fost inventate înaintea calculatorului, cu utilizări în industria textilă și cea muzicală. Astfel, trebuie menționat războiul de țesut inventat de Joseph Marie Jacquard, un echipament mecanic, care pe baza unor cartele perforate (figura 1 – Imagine din Muzeul Tehnic German din Berlin) permitea realizarea, în mod automat, a unor modele complexe de țesături. Cu această tehnică, a fost creată și cea mai celebră imagine asociată cu tehnica de calcul automat, și anume un portretul lui Jacquard, realizat pe unul din războaiele lui de țesut (figura 2).

Introducere



Război de țesut cu cartele perforate



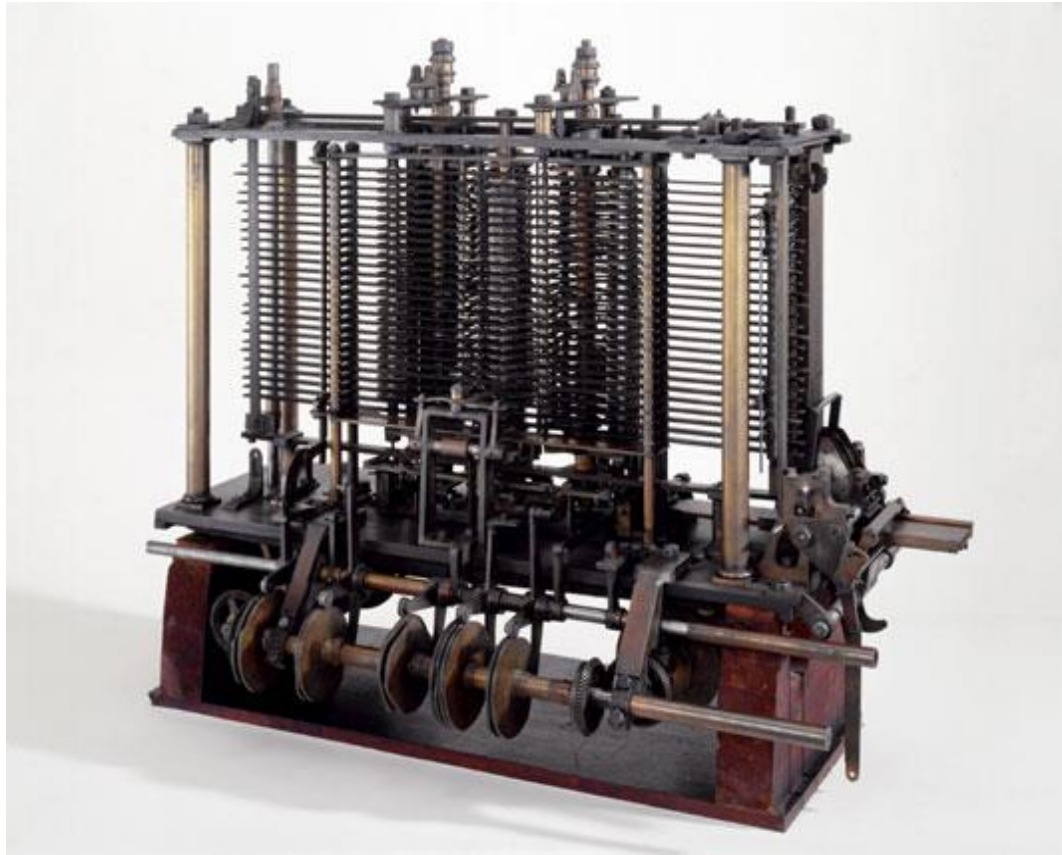
Portretul lui J.M. Jacquard,
brodat din mătase



Introducere

Unul din portretele lui Jacquard a aparținut lui Charles Babbage, de a cărui nume este legat conceptul de calculator programabil. Considerat părintele calculatorului, Babbage a dezvoltat primul calculator mecanic, cunoscut sub numele „Difference Engine”, urmat de un al doilea model, „Analytical Engine” conceput ca un calculator mecanic de uz general. Acest al doilea model este prezentat în figura 3, fiind expus în prezent în Muzeul de Știință din Londra (Marea Britanie).

Introducere



Calculatorul „Analytical Engine” dezvoltat
de C. Babbage



Generatii de limbaje de programare

O dată cu revoluția industrială și dezvoltarea componentelor electronice, calculatoarele au avut o evoluție explozivă care continuă și în zilele noastre. Considerate **cele mai complexe echipamente programabile**, pentru a comunica cu calculatorul, un utilizator (programator) poate folosi **limbaje de programare de diferite generații** pentru a rezolva diverși algoritmi. Astfel, o posibilă clasificare a limbajelor de programare le grupează în **5 generații evolutive**:

Generatii de limbaje de programare

Limbaje de programare de generația 1: sunt limbaje de programare scrise direct în **cod mașină**. Fără existența unui compilator, instrucțiunile se introduc direct în cod binar (secvențe de 0 și 1). Principalul lor avantaj este dat de viteza mare de execuție și eficiența lor, însă ele necesită un nivel foarte ridicat de cunoștințe. În prezent sunt folosite pentru programarea funcțiilor de bază ale sistemelor de operare: drivere, interfețe de comunicare.



Generatii de limbaje de programare

Limbaje de programare de generația 2: cuprind în general **limbajele de asamblare**, fiind considerate limbaje de bază. Codul acestora poate fi scris și citit de un programator, însă pentru a fi interpretate de calculator acestea trebuie convertite, printr-un asamblor. Și aceste limbaje necesită cunoștințe foarte avansate și s-au impus, în sistemele din zilele noastre în procesele care necesită putere foarte mare de calcul, cum ar fi jocurile, editarea video și manipularea imaginilor.



Generatii de limbaje de programare

Limbaje de programare de generația 3: reprezintă o treaptă evolutivă față de generația a doua prin crearea unui mediu de lucru mai ușor de utilizat. Limbajele din această generație cuprind multe elemente care ajută programatorul, cum ar fi structurarea datelor. Majoritatea limbajelor de programare de uz general (C, C++, JAVA, BASIC, PASCAL) sunt considerate limbaje de generația a treia, de un nivel mediu.



Generatii de limbaje de programare

Limbaje de programare de generația 4: cuprind **limbaje specializate pentru rezolvarea algoritmilor și ingineria sistemelor** reducând mult efortul de programare și timpul necesar rezolvării unui algoritm. Într-o prezentare sinoptică, multe funcționalități ale acestor limbaje sunt o „**cutie neagră**” prin care datele sunt procesate, fără ca utilizatorul să fie nevoit să cunoască algoritmul respectiv. Deși permit rezolvarea problemelor într-o formă simplă, accesibilă unui spectru mai larg de utilizatori, utilizarea lor este limitată de viteza mai redusă de rulare a codului de execuție.



Generatii de limbaje de programare

Limbaje de programare de generația 5: reprezintă limbajele care propun rezolvarea problemelor pe baza unor constrângeri impuse programului, înlocuind astfel algoritmul scris de programator. Aceste limbaje de programare se folosesc în prezent în cercetări legate de inteligența artificială.

A decorative teal circular graphic is located in the top-left corner of the slide, partially overlapping the title area.

MATLAB

MATLAB (**MAT**rix **LAB**oratory) este un limbaj de programare de nivel înalt, ce oferă un mediu de lucru interactiv pentru calcul numeric, vizualizări și programare. Cu ajutorul MATLAB se pot realiza diverse analize de date, dezvolta algoritmi și realiza modele și aplicații. Prin modul de structurare al limbajului, a modulelor disponibile și a funcțiilor predefinite, MATLAB permite dezvoltarea unor soluții multiple pentru o problemă dată, ceea ce permite optimizarea algoritmului final, generând în general soluții mai rapide decât limbajele clasice de programare.

MATLAB

Limbajul de programare MATLAB include un modul de bază utilizat pentru calcul numeric și analiză a datelor, însă conține și un număr extrem de mare de module suplimentare dezvoltate pentru operații specifice:

- »Calcul paralel;
- »Statistică și optimizări;
- »Proiectarea și evaluarea sistemelor de comandă;
- »Procesarea semnalelor și comunicații;
- »Procesarea imaginilor și realitate virtuală;
- »Teste și măsurători;
- »Calcul financiar;
- »Biologie;
- »Verificarea și generarea de cod;
- »Dezvoltarea de aplicații independente;
- »Procesarea bazelor de date și generarea de rapoarte.



Ce este MATLAB?

Un sistem interactiv pentru calcule numerice

Are implementati algoritmi care asigura obtinerea unor rezultate certe

Permite efectuarea unor operatii complexe folosind doar una, doua comenzi

Permite definirea unor functii utilizator

Are implementate functii grafice care permit reprezentari de complexitate ridicata

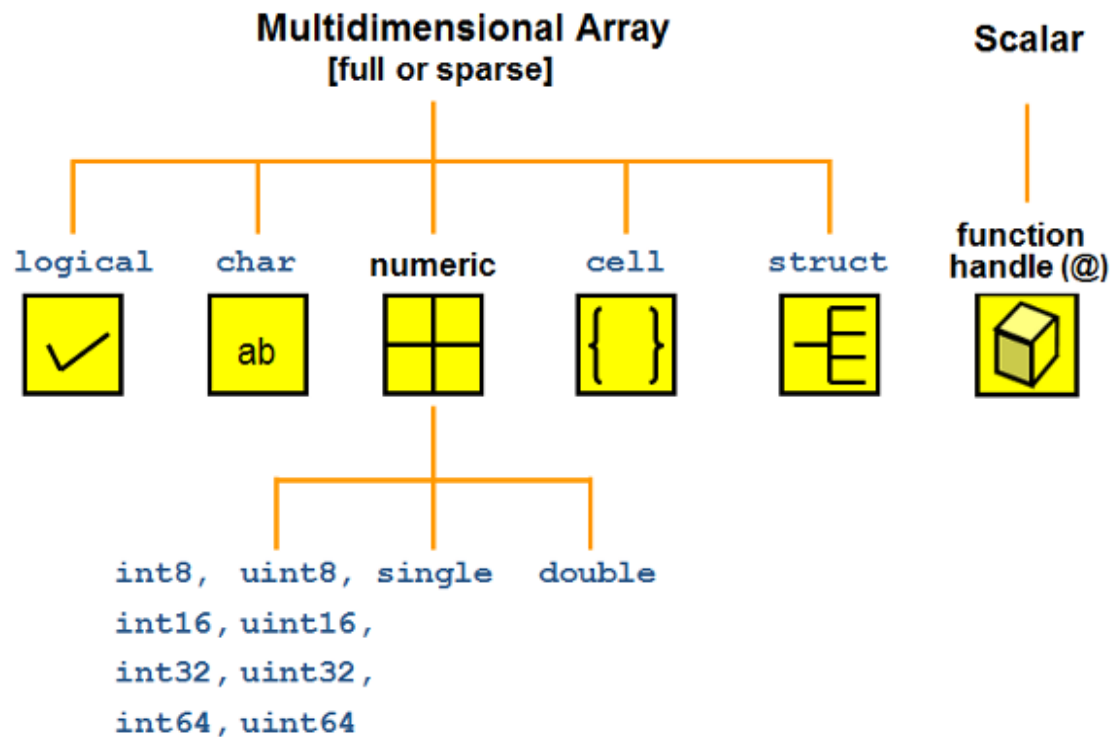
Tipuri de date in MATLAB

Indiferent de modul de reprezentare al datelor, limbajul MATLAB efectuează toate calculele numerice în dublă precizie, cu o reprezentare a datelor pe 8 bytes (64 de biți). Astfel tipul implicit de date al tuturor variabilelor este **double**. Spre deosebire de alte limbaje de programare, în MATLAB o variabilă nu trebuie declarată, ea fiind automat creată de program atunci când i se face prima atribuire.

În MATLAB există 15 tipuri fundamentale de date, și toate sunt văzute sub formă matriceală, cu excepția adresei unei funcții care este întotdeauna un scalar.

Figura urmatoare prezintă succint tipurile de date existente în MATLAB. Fiecare tip principal de date are asociată și o pictogramă pentru o identificare facilă în **workspace**.

Tipuri de date in MATLAB



Tipuri de date in MATLAB

Variabile de tip logic (logical): sunt variabile de formă matriceală a căror termeni pot lua doar două valori – logic adevărat (True) și logic fals (False) codificate numeric prin 1 (pentru valoarea adevărat) și 0 (pentru valoarea fals). Acest tip de variabile se folosesc în evaluări comparative sau în generarea unor șiruri de valori logice (e.g. valorile rezultate în urma testelor unui lot de 100 de becuri).

Variabile de tip caracter (char): sunt variabile în care se salvează secvențe de caractere, numite și string-uri. Aceste variabile suportă conversia în variabile numerice, unde pentru fiecare caracter se va aloca valoarea corespunzătoare codului ASCII al acestuia. În cazul în care se dorește alocarea mai multor string-uri se recomandă folosirea unei structuri de tip **cell array** (grup de șiruri).

Tipuri de date in MATLAB

Variabile de tip numeric (numeric): este cea mai răspândită clasă de variabile din MATLAB și cuprinde mai multe tipuri de date, deosebite prin modul de reprezentare, incluzând:

Numere întregi cu și fără semn;

Numere în virgulă mobilă în simplă și dublă precizie.

Trebuie subliniat faptul că în mod normal în MATLAB toate variabilele se salvează ca numere în virgulă mobilă în dublă precizie (8 bytes) acest lucru fiind un parametru care nu poate fi modificat. Însă, dacă se dorește, o variabilă numerică se poate salva cu un alt tip, prin specificarea acestuia la alocarea de valori. Prin utilizarea variabilelor ca numere întregi sau în simplă precizie se obține o gestionare mai eficientă a memoriei și acest lucru se recomandă acolo unde se cunoaște formatul datelor și o precizie mai mică în calcule nu conduce spre rezultate incorecte. Orice fel de variabilă numerică poate fi salvată ca un scalar, un vector (linie sau coloană) sau o matrice multidimensională

Tipuri de date in MATLAB

Tip de dată (clasă)	Domeniu de valori	Funcția de conversie
Tipuri de date întregi		
Întreg cu semn pe 8 biți	$-2^7 \div 2^7 - 1$	int8
Întreg cu semn pe 16 biți	$-2^{15} \div 2^{15} - 1$	int16
Întreg cu semn pe 32 biți	$-2^{31} \div 2^{31} - 1$	int32
Întreg cu semn pe 64 biți	$-2^{63} \div 2^{63} - 1$	int64
Întreg fără semn pe 8 biți	$0 \div 2^8 - 1$	uint8
Întreg fără semn pe 16 biți	$0 \div 2^{16} - 1$	uint16
Întreg fără semn pe 32 biți	$0 \div 2^{32} - 1$	uint32
Întreg fără semn pe 64 biți	$0 \div 2^{64} - 1$	uint64

Tipuri de date in MATLAB

Tip de dată (clasă)	Domeniu de valori	Funcția de conversie
Tipuri de date în virgulă mobilă		
Simplă precizie	Cel mai mare număr negativ: -3.40282e+38 Cel mai mic număr negativ: -1.17549e-38 Cel mai mic număr pozitiv: 1.17549e-38 Cel mai mare număr pozitiv: 3.40282e+38	single
Dublă precizie	Cel mai mare număr negativ: -1.79769e+308 Cel mai mic număr negativ: -2.22507e-308 Cel mai mic număr pozitiv: 2.22507e-308 Cel mai mare număr pozitiv: 1.79769e+308	<u>double</u>

Observație: notația X.XXXe+Y înseamnă $X.XXX \cdot 10^Y$.

Tipuri de date in MATLAB

Variabile de tip celulă (cell): aceste variabile stochează șiruri cu diferite tipuri de date și dimensiuni. Astfel, datele pot fi grupate după preferințele utilizatorului, accesarea acestora făcându-se exact ca și în cazul șirurilor.

Variabile de tip structură (struct): acest tip de variabile, foarte asemănător cu cel anterior, cell, permite stocarea datelor de diferite tipuri și dimensiuni într-o formă structurată, aleasă de utilizator. Accesul este foarte facil, permițând accesarea unei singure variabile sau a unui întreg grup printr-o singură operație.

Variabile de tip adresă de funcție (function handle): sunt variabile care permit adresarea (utilizarea) unei funcții în mod indirect. Acestea reprezintă pointer-i spre funcție și permit transferul unei funcții spre altă funcție.

Tipuri de date in MATLAB

Exemplu de definire a unor variabile pentru fiecare tip de date din MATLAB

```
>> var1=logical(1);  
>> var2=char('Exemplu text');  
>> var3=5/3;  
>> var4=cell(3,4,6);  
>> var5.nume='Nume test';, var5.valoare=10;, var5.sir=[1 2 3 4 5];  
>> var6=@sin;  
>> var5
```

```
var5 =
```

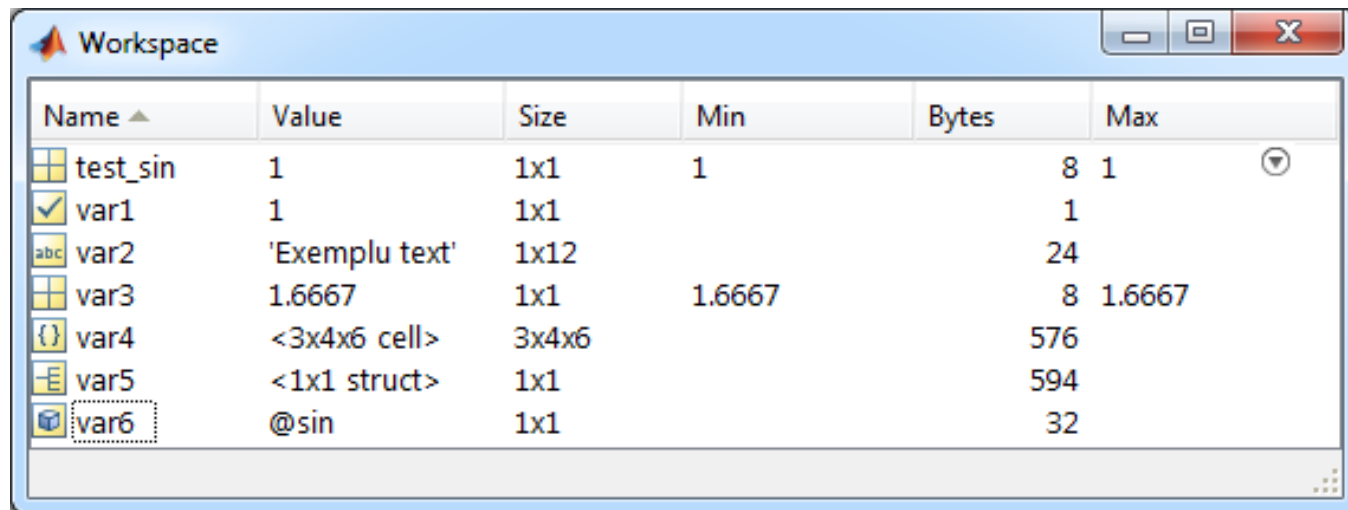
```
      nume: 'Nume test'  
      valoare: 10  
      sir: [1 2 3 4 5]  
>> test_sin=var6(pi/2)
```

```
test_sin =
```

Tipuri de date in MATLAB

```
>> whos var1 var2 var3 var4 var5 var6
```

<u>Name</u>	<u>Size</u>	<u>Bytes</u>	<u>Class</u>	<u>Attributes</u>
var1	1x1	1	<u>logical</u>	
var2	1x12	24	<u>char</u>	
var3	1x1	8	<u>double</u>	
var4	3x4x6	576	<u>cell</u>	
var5	1x1	594	<u>struct</u>	
var6	1x1	32	<u>function_handle</u>	



Interfata Matlab

MATLAB R2012b

HOME PLOTS APPS

New Script New Open Find Files Import Data Save Workspace Open Variable Clear Workspace Analyze Code Run and Time Clear Commands Simulink Library Layout Set Path Preferences Help Community Request Support

FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

Current Folder: C:\Users\Calin\Documents\MATLAB

Name	Date Modified
Apps	6/20/2013 11:5...

Select a file to view details

Command Window

New to MATLAB? Watch this [Video](#), see [Examples](#), or read [Getting Started](#).

```
f> >>
```

Workspace

Name	Value	Min	Max
------	-------	-----	-----

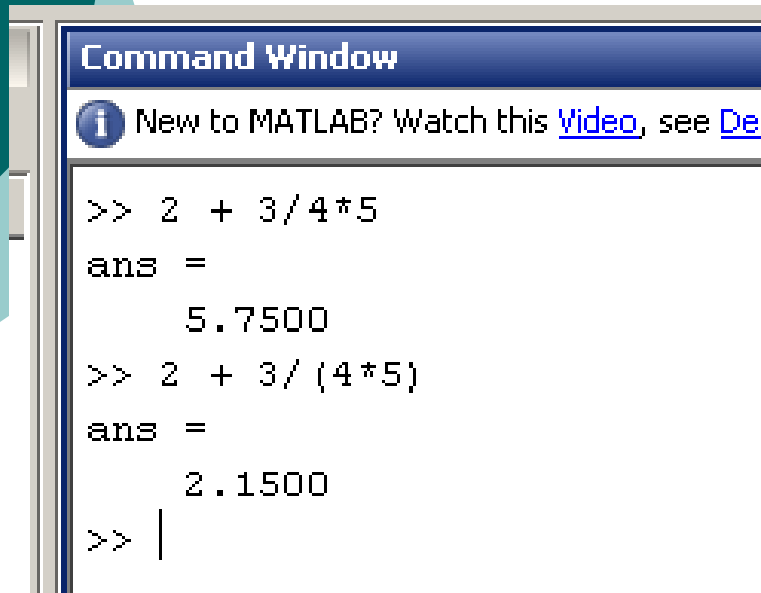
Command History

```
%-- 12/15/2013 5:40 PM --%  
%-- 12/15/2013 5:46 PM --%
```

Ready

OVR

MATLAB ca si calculator

A screenshot of the MATLAB Command Window. The title bar is blue with the text 'Command Window'. Below the title bar, there is a message: 'i New to MATLAB? Watch this [Video](#), see [De](#)'. The command prompt shows two calculations: first, '>> 2 + 3/4*5' followed by 'ans = 5.7500', and second, '>> 2 + 3/(4*5)' followed by 'ans = 2.1500'. The cursor is at the end of the second command line, '>> |'.

Ordinea de efectuare a operatiilor

Operanzi: +, -, *, /, ^

- Termenii din paranteze
- Ridicarea la putere
- *, / - de la stanga la dreapta
- +, - de la stanga la dreapta

Toate calculele se efectueaza in dubla precizie, care inseamna aproximativ 15 cifre semnificative dupa virgula

Numere si formatari

Type	Examples
Integer	1362, -217897
Real	1.234, -10.76
Complex	$3.21 - 4.3i$ ($i = \sqrt{-1}$)
Inf	Infinity (result of dividing by 0)
NaN	Not a Number, 0/0

Notatia "e" se foloseste de obicei pentru numere foarte mari sau mici si inlocuieste ridicarea numarului la puterea 10

Ex:

$$-1.3412e+03 = -1.3412 \times 10^3 = -1341.2$$

$$-1.3412e-01 = -1.3412 \times 10^{-1} = 0.13412$$

Numere si formatare

Command	Example of Output
>>format short	31.4162(4-decimal places)
>>format short e	3.1416e+01
>>format long e	3.141592653589793e+01
>>format short	31.4162(4-decimal places)
>>format bank	31.42(2-decimal places)

Variabile

Command Window

 New to MATLAB? Watch this [Video](#), see [Der](#)

```
>> 3-2^2
ans =
    -1
>> ans*3
ans =
    -3
>> ans*-10
ans =
    30
>> |
```

Command Window

 New to MATLAB? Watch this [Video](#), see [Der](#)

```
>> x=3-2^3
x =
    -5
>> y=3*x
y =
   -15
>> x+y
ans =
   -20
>> |
```

Ans – variabila generica



Variabile

Notarea variabilelor se poate face utilizand orice combinatie de litere si cifre, cu conditia ca primul caracter sa fie litera:

NetCost, 4ever, Left2Play, z23es234x3

Exista variabile predefinite care nu ar trebui modificate:

Eps, pi etc

Variabile predefinite

Variabilă predefinită	Valoare predefinită
i	0 + 1.0000i
j	0 + 1.0000i
eps	2.2204e-16
pi	3.141592653589793
Inf	Inf (infinit)
NaN	NaN (Not a Number)
clock	returnează o valoare sub forma unui șir cu șase elemente de forma: [an lună zi oră minute secundă]
date	returnează un șir de caractere care reprezintă data curentă în format: dd-mmm-yyyy e.g. 16-Dec-2013

Variable

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> eps
ans =
    2.2204e-016
>> pi
ans =
    3.1416
>> format long
>> pi
ans =
    3.141592653589793
>>
```

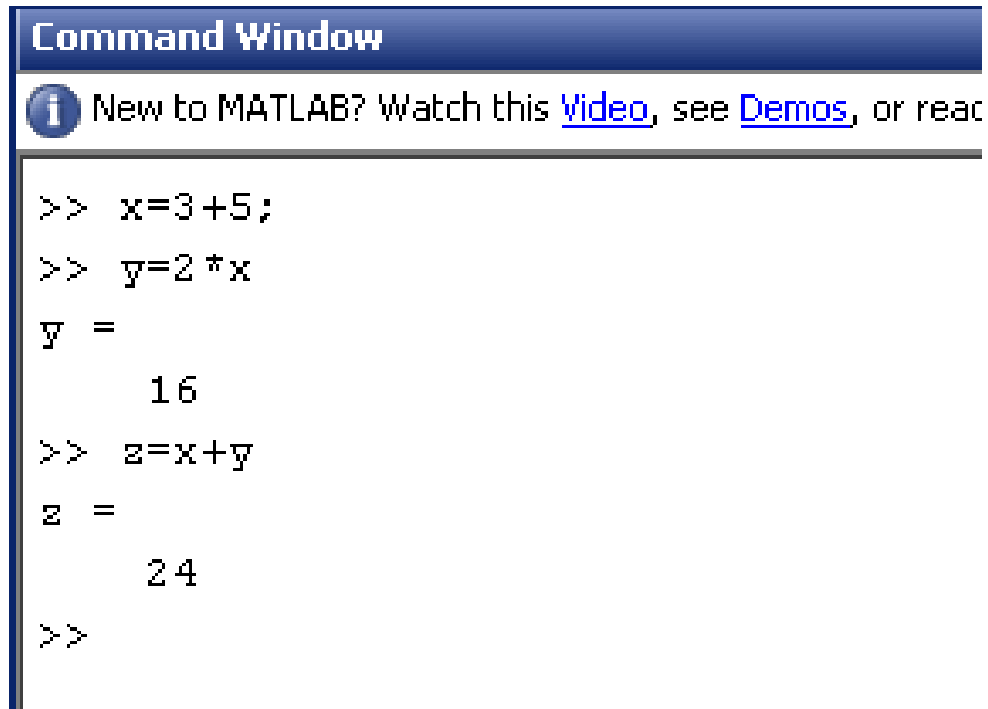
Operatii cu numere complexe

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)

```
>> i,j
ans =
      0 + 1.0000i
ans =
      0 + 1.0000i
>> i=3
i =
      3
>> clear
>> i
ans =
      0 + 1.0000i
>>
```


Suprimarea afisarii rezultatelor

A screenshot of the MATLAB Command Window. The title bar is dark blue with the text 'Command Window' in white. Below the title bar is a light gray area with an information icon (a lowercase 'i' in a circle) and the text 'New to MATLAB? Watch this [Video](#), see [Demos](#), or reac'. The main area of the window is white and contains the following text: '>> x=3+5;', '>> y=2*x', 'y =', '16', '>> z=x+y', 'z =', '24', and '>>'.

```
Command Window
i New to MATLAB? Watch this Video, see Demos, or reac
>> x=3+5;
>> y=2*x
y =
    16
>> z=x+y
z =
    24
>>
```

Daca o operatie este urmata de ";" atunci rezultatul acesteia nu se va afisa

Funcția de help

Help + “cuvant cheie” sau doc + “cuvant cheie”

```
Command Window
i New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> help if
IF Conditionally execute statements.
    The general form of the IF statement is

        IF expression
            statements
        ELSEIF expression
            statements
        ELSE
            statements
        END

    The statements are executed if the real part of the expression
    has all non-zero elements. The ELSE and ELSEIF parts are optional.
    Zero or more ELSEIF parts can be used as well as nested IF's.
    The expression is usually of the form expr rop expr where
    rop is ==, <, >, <=, >=, or ~=.

    Example
        if I == J
            A(I,J) = 2;
        elseif abs(I-J) == 1
            A(I,J) = -1;
```

Functii integrate in meniul MATLAB (built-in)

Functii trigonometrice

sin, cos, tan cu argumentul in radiani

```
Command Window
i New to MATLAB? Watch this Video, see Demos.

>> x=pi/4
x =
    0.7854
>> cos(x)
ans =
    0.7071
>> x=pi
x =
    3.1416
>> cos(x)
ans =
    -1
>>
```

Alte functii elementare

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)

```
>> x=9;  
>> sqrt(x),exp(x),log(sqrt(x)),log10(x^2+6)  
ans =  
      3  
ans =  
  8.1031e+003  
ans =  
  1.0986  
ans =  
  1.9395  
>>
```

$\exp(x)=e^x$

Vectori – vectorii linie

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> v=[3 1 sqrt(5) 2^2]
v =
    3.0000    1.0000    2.2361    4.0000
>> v2=[1+ 2 3]
v2 =
     3     3
>> v2=[1 +2 3]
v2 =
     1     2     3
>> v3=[1:1:4]
v3 =
     1     2     3     4
>> v4=v+v3
v4 =
    4.0000    3.0000    5.2361    8.0000
>> v+v2
??? Error using ==> plus
Matrix dimensions must agree.

>>
```

Operatii cu elementele unui vector

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> v=[-2 5 -4 -7 4 8 11]
v =
    -2     5    -4    -7     4     8    11
>> sum(v)
ans =
    15
>> length(v)
ans =
     7
>> 15/7
ans =
    2.1429
>> mean(v)
ans =
    2.1429
>> max(v)
ans =
    11
>> vv=sort(v)
vv =
    -7    -4    -2     4     5     8    11
>> |
```

Crearea unui vector cu elemente consecutive cu un pas oarecare

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> 1:3
```

```
ans =
```

```
     1     2     3
```

```
>> 1:-1:-9
```

```
ans =
```

```
     1     0    -1    -2    -3    -4    -5    -6    -7    -8    -9
```

```
>> 1:0.1:2
```

```
ans =
```

```
Columns 1 through 7
```

```
    1.0000    1.1000    1.2000    1.3000    1.4000    1.5000    1.6000
```

```
Columns 8 through 11
```

```
    1.7000    1.8000    1.9000    2.0000
```

```
>> |
```

Vectori – Vectorii coloana

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> c=[1; 2; 3]
```

```
c =
```

```
    1
```

```
    2
```

```
    3
```

```
>> length(c)
```

```
ans =
```

```
    3
```

```
>> sum(c)
```

```
ans =
```

```
    6
```

```
>>
```

Liniile unui vector sunt separate de caracterul **;**

Transpunerea unui vector

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting](#)

```
>> c=[1 5 sqrt(3)]  
c =  
    1.0000    5.0000    1.7321  
>> c_transpus=c'  
c_transpus =  
    1.0000  
    5.0000  
    1.7321  
>> c_transpus'  
ans =  
    1.0000    5.0000    1.7321  
>> |
```

Transpunerea unui vector se face cu caracterul ' ▸

Plotarea functiilor elementare

Sa se ploteze graficul functiei $y = \sin(3\pi x)$ pentru $x \in (0,1)$

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> N = 10; h = 1/N; x = 0:h:1
```

```
x =
```

```
Columns 1 through 7
```

```
0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
```

```
Columns 8 through 11
```

```
0.7000    0.8000    0.9000    1.0000
```

```
>> y=sin(3*pi*x)
```

```
y =
```

```
Columns 1 through 7
```

```
0    0.8090    0.9511    0.3090   -0.5878   -1.0000   -0.5878
```

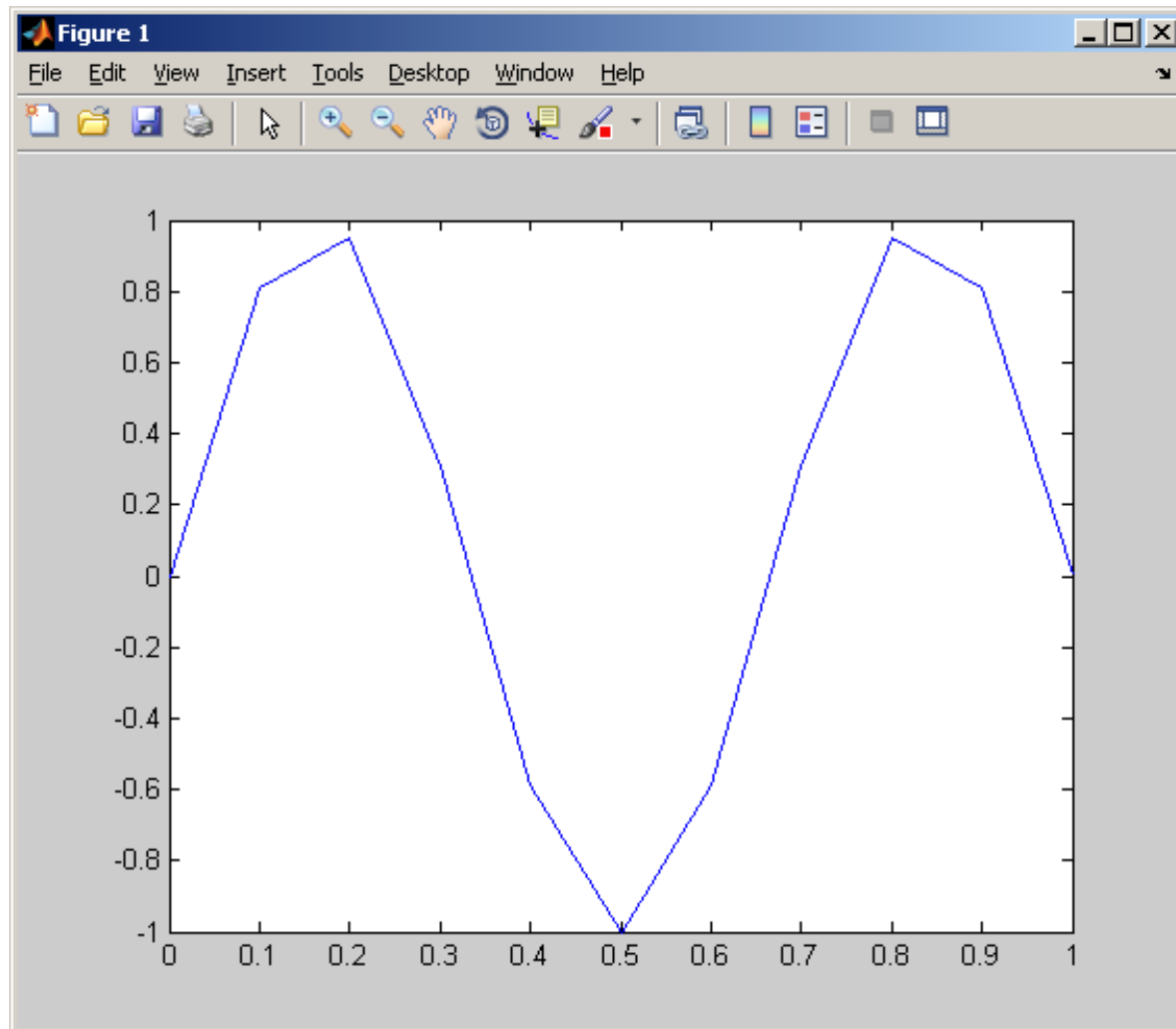
```
Columns 8 through 11
```

```
0.3090    0.9511    0.8090    0.0000
```

```
>> plot(x,y)
```

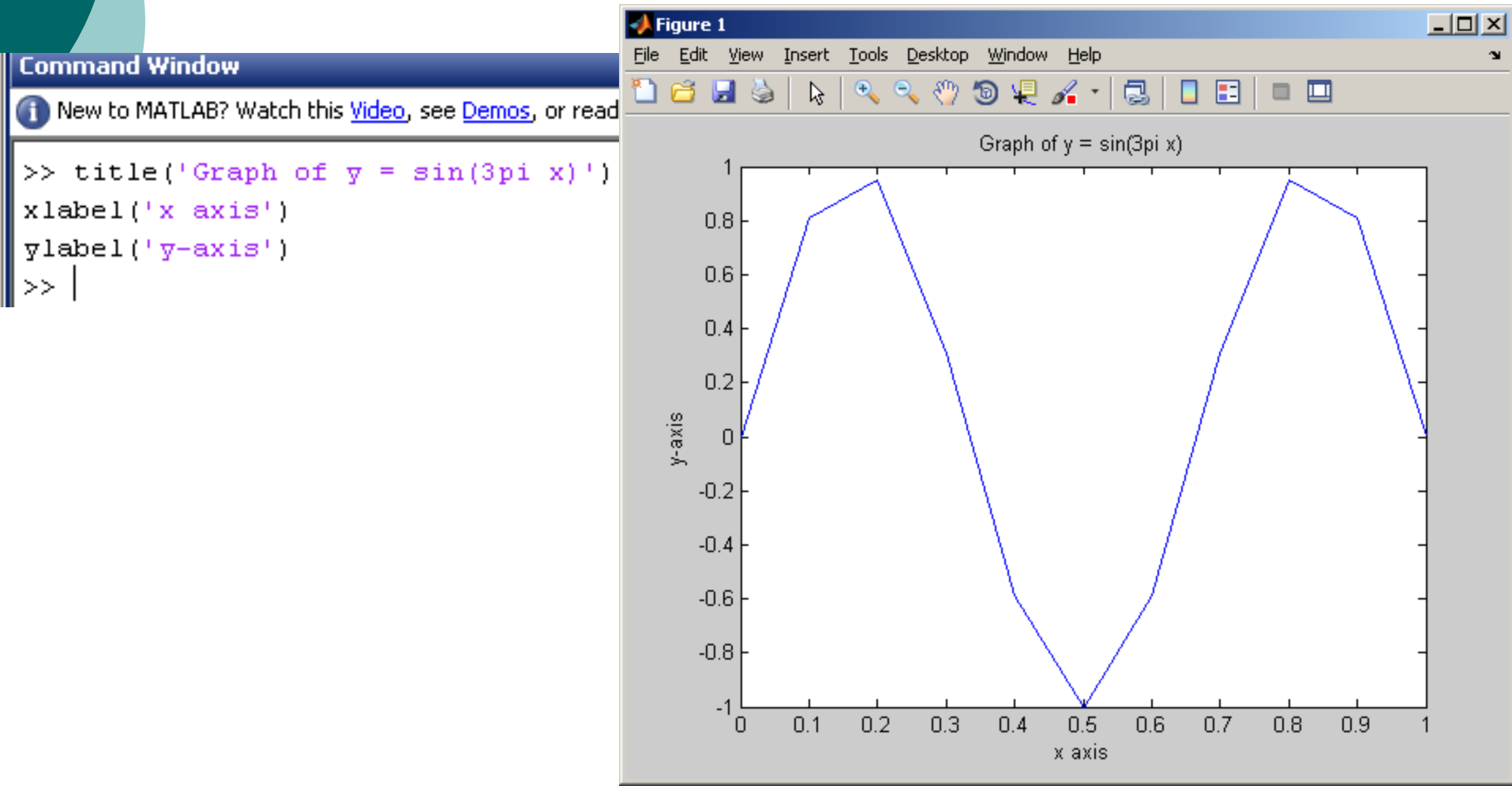
```
>> |
```

Plotarea functiilor elementare



Plotarea functiilor elementare

Notatii pe grafice

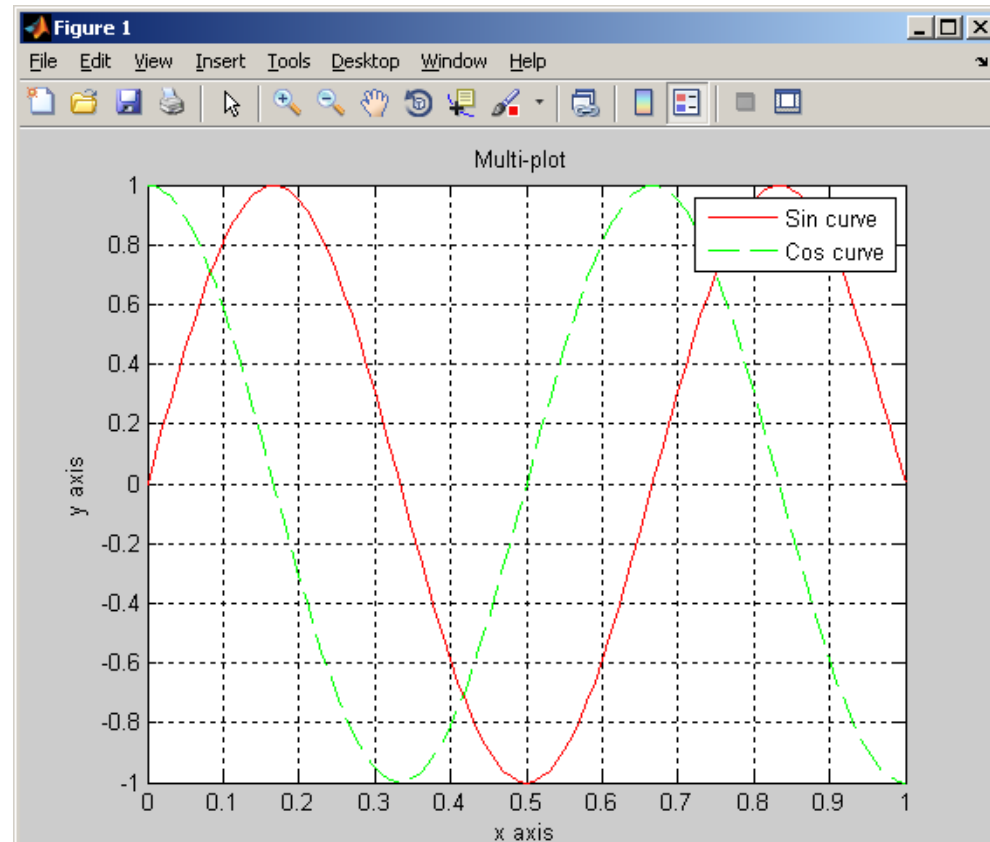


Plotarea functiilor elementare

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)

```
>> N = 100; h = 1/N; x = 0:h:1;  
>> y = sin(3*pi*x); plot(x,y)  
>> grid  
>> plot(x,y,'r-',x,cos(3*pi*x),'g--')  
>> legend('Sin curve','Cos curve')  
title('Multi-plot ')  
xlabel('x axis'), ylabel('y axis')  
grid  
>> |
```





Plotarea functiilor elementare

La fiecare comanda plot, fereastra grafica este initializata si se incepe un nou grafic. Daca se doreste reprezentarea mai multor elemente pe acelasi grafic, din comenzi diferite, dupa realizarea primului grafic, se foloseste functia:

Hold

Iar pentru anulara efectului acesteia:

Hold off

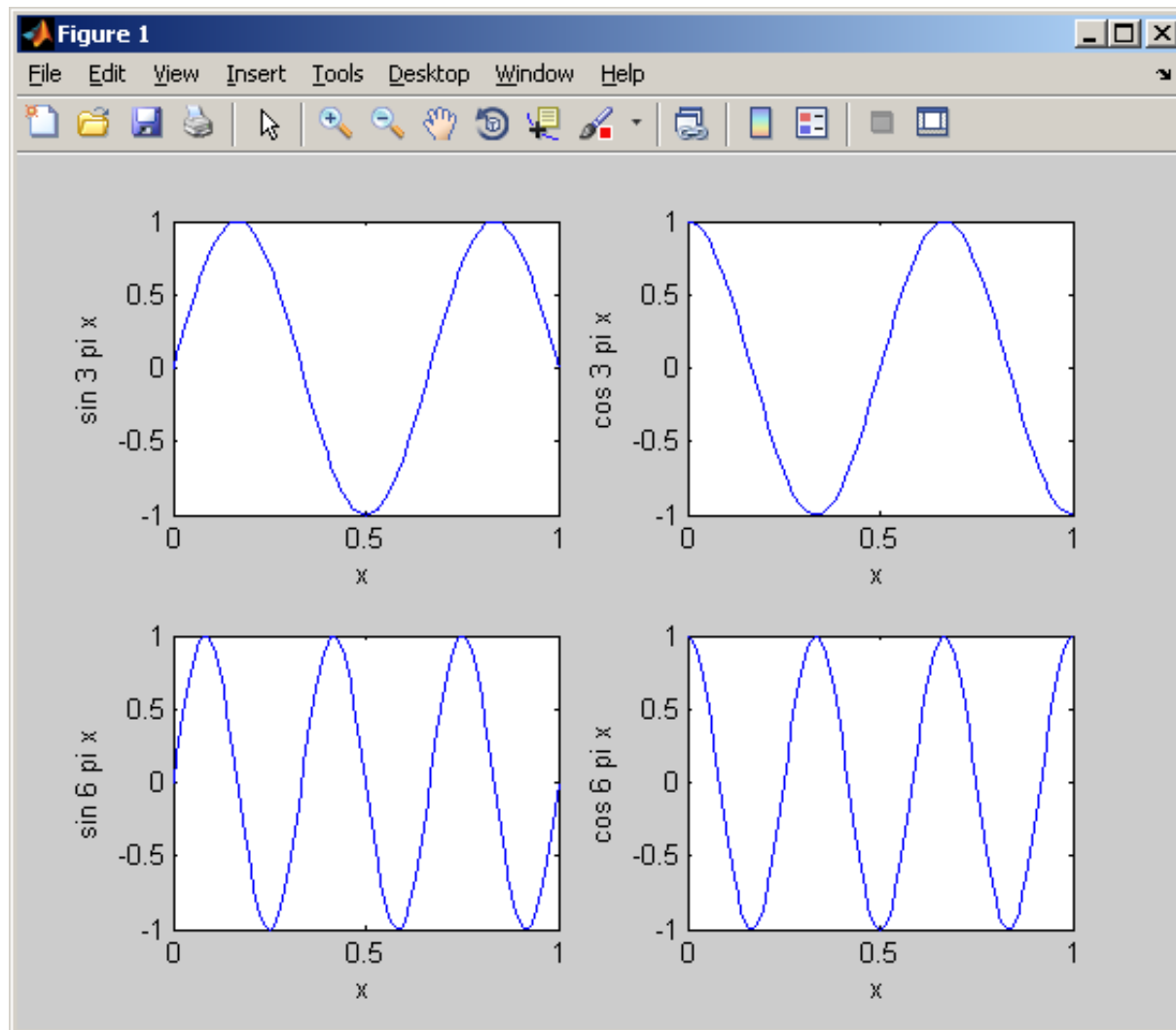
Plotare - subploturi

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)

```
>> N = 100; h = 1/N; x = 0:h:1;
>> y = sin(3*pi*x);
>> subplot(221), plot(x,y)
xlabel('x'), ylabel('sin 3 pi x')
subplot(222), plot(x,cos(3*pi*x))
xlabel('x'), ylabel('cos 3 pi x')
subplot(223), plot(x,sin(6*pi*x))
xlabel('x'), ylabel('sin 6 pi x')
subplot(224), plot(x,cos(6*pi*x))
xlabel('x'), ylabel('cos 6 pi x')
>> |
```

Plotare - subploturi



Operatii cu vectori

Produs scalar

*

```
>> u = [ 10, -11, 12], v = [20; -21; -22]
prod = u*v
u =
    10    -11     12
v =
    20
   -21
   -22
prod =
    167
>> w = [2, 1, 3], z = [7; 6; 5]
w =
     2     1     3
z =
     7
     6
     5
>> w*z
ans =
    35
>> prod*w
ans =
    334    167    501
>> prod*z
ans =
    1169
    1002
     835
>> v*z
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

Operatii cu vectori

Produs element cu element

“dot product”

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)

```
>> u
u =
    10    -11    12

>> w
w =
     2     1     3

>> u.*w
ans =
    20    -11    36

>> u.*w'
??? Error using ==> times
Matrix dimensions must agree.

>> z=1:4
z =
     1     2     3     4

>> z.*z
ans =
     1     4     9    16

>> |
```

Operatii cu vectori

Calculati valoarea functiei $y = x \sin \pi x$ pentru

$x = 0, 0.25, \dots, 1$

$$x \times \sin \pi x = x \sin \pi x$$

$$0 \times 0 = 0$$

$$0.2500 \times 0.7071 = 0.1768$$

$$0.5000 \times 1.0000 = 0.5000$$

$$0.7500 \times 0.7071 = 0.5303$$

$$1.0000 \times 0.0000 = 0.0000$$

```
>> x=(0:0.25:1)'
```

```
x =
```

```
0
```

```
0.2500
```

```
0.5000
```

```
0.7500
```

```
1.0000
```

```
>> y=x.*sin(pi*x)
```

```
y =
```

```
0
```

```
0.1768
```

```
0.5000
```

```
0.5303
```

```
0.0000
```

```
>>
```

Manual

Matlab

Operatii cu vectori

**Impartirea element
cu element
"dot division"**

./

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> a=1:5, b=6:10, a./b
```

```
a =
```

```
     1     2     3     4     5
```

```
b =
```

```
     6     7     8     9    10
```

```
ans =
```

```
    0.1667    0.2857    0.3750    0.4444    0.5000
```

```
>> b.*ans
```

```
ans =
```

```
     1     2     3     4     5
```

```
>> c=-2:2
```

```
c =
```

```
    -2    -1     0     1     2
```

```
>> a./c
```

```
ans =
```

```
   -0.5000   -2.0000         Inf    4.0000    2.5000
```

Operatii cu vectori

Estimati valoarea limitei: $\lim_{x \rightarrow 0} \frac{\sin \pi x}{x}$

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> x=[0.1, 0.01, 0.001, 0.0001, 0.00001]
x =
    0.1000    0.0100    0.0010    0.0001    0.0000
>> format long
>> y=sin(pi*x)./x
y =
Columns 1 through 3
    3.090169943749474    3.141075907812829    3.141587485879564
Columns 4 through 5
    3.141592601912665    3.141592653073022
>> y-pi
ans =
Columns 1 through 3
   -0.051422709840319   -0.000516745776964   -0.000005167710230
Columns 4 through 5
   -0.000000051677128   -0.000000000516771
>> |
```

Operatii cu vectori

**Ridicarea la
putere element
cu element
"dot power"**

.^

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> u=[10 11 12]
```

```
u =  
    10    11    12
```

```
>> u=u.^2
```

```
u =  
   100   121   144
```

```
>> a=1:8
```

```
a =  
     1     2     3     4     5     6     7     8
```

```
>> a.^2
```

```
ans =  
     1     4     9    16    25    36    49    64
```

```
>>
```

Matrice

Introducerea unei
matrice 2x3

$$A = \begin{bmatrix} 5 & 7 & 9 \\ 1 & -3 & -7 \end{bmatrix}$$

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read

```
>> a=[5 7 9  
1 -3 -7]  
a =  
     5     7     9  
     1    -3    -7  
>> b=[5 7 9; 1 -3 -7]  
b =  
     5     7     9  
     1    -3    -7  
>> c=[5,7,9;1,-3,-7]  
c =  
     5     7     9  
     1    -3    -7  
>> |
```

Matrice - dimensioni

Size(a,b)

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting](#)

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

```
A =
```

```
     1     2     3
     1     3     5
     3     2     1
```

```
>> size(A)
```

```
ans =
```

```
     3     3
```

```
>> size(A,1)
```

```
ans =
```

```
     3
```


Matrice – configuratii speciale

ones(m,n)

zeros(m,n)

eye(n)

```
Command Window
i New to MATLAB? Watch this Video, see Der

>> ones(3,4)

ans =

     1     1     1     1
     1     1     1     1
     1     1     1     1

>> zeros(1,2)

ans =

     0     0

>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1

>>
```

Matricea diagonală

Diag(vector)

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or r

```
>> d=[1; -4; -5; 7]
```

```
d =
```

```
    1  
   -4  
   -5  
    7
```

```
>> D=diag(d)
```

```
D =
```

```
    1    0    0    0  
    0   -4    0    0  
    0    0   -5    0  
    0    0    0    7
```

```
>>
```

Matricea diagonala

Diag(vector)

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read the [Help](#) documentation.

```
>> ddd=[1:4;2:5;]
```

```
ddd =
```

```
    1    2    3    4
    2    3    4    5
```

```
>> d=diag(ddd)
```

```
d =
```

```
    1
    3
```

```
>>
```

Operatii cu matrice

Suma si produsul
a doua matrice

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#)

```
>> a=[1 2 3;4 5 6;7 8 9]
```

```
a =
```

1	2	3
4	5	6
7	8	9

```
>> a+a
```

```
ans =
```

2	4	6
8	10	12
14	16	18

```
>> a*a
```

```
ans =
```

30	36	42
66	81	96
102	126	150

```
>>
```

Produs
intre o
matrice si
un vector

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Gettir](#)

```
>> a
```

```
a =
```

```
     1     2     3  
     4     5     6  
     7     8     9
```

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

```
b =
```

```
     1     2     3
```

```
>> a*b
```

```
??? Error using ==> mtimes  
Inner matrix dimensions must agree.
```

```
>> a*b'
```

```
ans =
```

```
    14  
    32  
    50
```

Produsul a doua matrice

Fie $A[m][n]$ și $B[n][l]$. Matricea produs dintre A și B va fi de forma $P[m][l]$. Formula de calcul este:

$$P_{[i][j]} = \sum_{k=0}^{n-1} A_{[i][k]} \cdot B_{[k][j]}$$

Produs intre doua matrice

```
Command Window
i New to MATLAB? Watch this Video, see Demos, or read (

>> a

a =

     1     2     3
     4     5     6
     7     8     9

>> b=[3 3 3]

b =

     3     3     3

>> a*b
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> a*b'

ans =

    18
    45
    72

>> |
```

Matrice –
produsul a doua
matrice element
cu element
(dot product)

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or

```
>> A=[1,2;3,4]
```

```
A =
```

```
     1     2  
     3     4
```

```
>> B=[2,2;2,2]
```

```
B =
```

```
     2     2  
     2     2
```

```
>> A.*B
```

```
ans =
```

```
     2     4  
     6     8
```

```
>> |
```


Operatori aritmetici in MATLAB

Limbajul MATLAB folosește operatori matematici clasici la care se adaugă câțiva specifici. Fiind un limbaj matriceal operatorii se folosesc în mod identic între orice fel de variabile, dar respectă toate regulile matematice cunoscute. Ca și operatori clasici sunt operatorii:

Adunare: +; Scădere: -; Înmulțire: *; Împărțire: /; Ridicare la putere: ^; Parantezele: ();

La care se adaugă:

Împărțire la stânga: \;

Transpusa: ';

Operatorul element cu element: „.”.

Operatori aritmetici in MATLAB

De subliniat faptul că și dacă se folosesc mai multe rânduri de paranteze se vor folosi tot cele rotunde: $3*(2*x-5*(y^2+1))$.

O posibilă clasificare a operatorilor matematici din MATLAB, care îi grupează în două categorii distincte, se poate face după modul de aplicare al acestora asupra variabilelor:

1. Operatori de calcul matriceal, care respectă toate regulile matematice de algebră liniară, putând fi aplicați în mod identic asupra oricăror variabile;
2. Operatori de calcul vectorial (element cu element), în care operatorul clasic este precedat de semnul "." (punct), care pot fi aplicați variabilelor de tip vector sau matrice.

Operatori relationali in MATLAB

Operatorii relaționali pot fi aplicați între orice fel de variabile de aceeași dimensiune, returnând o variabilă de aceeași dimensiune care va conține rezultatele aplicării operatorului relațional între elementele corespunzătoare din cele două variabile. Rezultatele posibile sunt:

1, dacă rezultatul comparației este adevărat din punct de vedere logic, și

0, dacă rezultatul comparației este fals.

În acest moment se poate defini și valoarea de adevăr a unei expresii, care din punct de vedere logic poate fi:

True, adică adevărată pentru orice **valoare diferită de zero**, și

False, adică falsă pentru valoare nulă.

Operatori relationali in MATLAB

Operator	Descriere
<	Operatorul mai mic. a<b va returna 1 logic (true) dacă valoarea lui a este mai mică decât valoarea lui b. În caz contrar rezultatul va fi 0 logic (fals).
>	Operatorul mai mare. a>b va returna 1 logic (true) dacă valoarea lui a este mai mare decât valoarea lui b. În caz contrar rezultatul va fi 0 logic (fals).
<=	Operatorul mai mic sau egal. a<=b va returna 1 logic (true) dacă valoarea lui a este mai mică sau egală față de valoarea lui b. În caz contrar rezultatul va fi 0 logic (fals).

Operatori relationali in MATLAB

Operator	Descriere
<code>>=</code>	Operatorul mai mare sau egal. <code>a>=b</code> va returna 1 logic (true) dacă valoarea lui a este mai mare sau egală față de valoarea lui b. În caz contrar rezultatul va fi 0 logic (fals).
<code>==</code>	Operatorul egal. <code>a==b</code> va returna 1 logic (true) dacă valoarea lui a este egală cu valoarea lui b. În caz contrar rezultatul va fi 0 logic (fals). A nu se confunda cu operatorul „=” care înseamnă atribuire.
<code>~=</code>	Operatorul diferit. <code>a~=b</code> va returna 1 logic (true) dacă valoarea lui a este diferită față de valoarea lui b. În caz contrar rezultatul va fi 0 logic (fals).

Operatori logici in MATLAB

Operatorii logici sunt folosiți pentru a face o evaluare, pe baza unui tabel de adevăr între două variabile. Acești operatori se folosesc în general în expresii condiționale pentru a decide sau evita execuția unui anumit bloc de instrucțiuni. Orice operație logică va returna un vector logic cu elemente de 1 logic (true) și 0 logic (false), în funcție de rezultatul estimării. Se definesc în principal trei operatori logici elementari:

~ - operatorul unar negație (NOT);

& - operatorul binar și (AND);

| - operatorul binar sau (OR).

Operatori logici in MATLAB

Acești operatori pot fi aplicați în mod similar cu cei relaționali, oricăror tipuri de variabile, cu condiția ca ambele variabile matriceale să aibă aceeași dimensiune sau cel puțin una dintre ele să fie scalară.

Fiecare din acești operatori are câte o funcție corespunzătoare, rezultatul aplicării acestora fiind identic. Astfel:

Funcție logică	Operator logic	Descriere
not(x)	$\sim x$	x negat
or(x,y)	$x y$	x sau y
and(x,y)	$x\&y$	x și y
xor(x,y)	-	x sau exclusiv y

Operatori logici in MATLAB

Rezultatul aplicării acestor funcții este ilustrat în următorul tabel de adevăr în care se vor compara toate variantele posibile pentru fiecare funcție.

x	y	$\sim x$	$x y$	$x\&y$	$xor(x,y)$
1	1	0	1	1	0
1	0	0	1	0	1
0	1	1	1	0	1
0	0	1	0	0	0

Operatori logici in MATLAB

```
>> a_scalar=5;
>> b_scalar=3;
>> a_sir=[1; 2; 3;];
>> a_matrice=[4 5 2; 1 2 4];
>> b_sir=[1; 0; 2;];
>> b_matrice=[0 0 1; 0 0 0];
>> c_scalar=0;
>> a_scalar | b_scalar
ans =
    1
>> a_scalar | b_scalar & ~c_scalar
ans =
    1
```

```
>> a_scalar | b_scalar & c_scalar
ans =
    1
>> a_matrice & b_matrice
ans =
    0    0    1
    0    0    0
>> a_matrice | b_matrice
ans =
    1    1    1
    1    1    1
>> xor(a_matrice, b_matrice)
ans =
    1    1    0
    1    1    1
>> c_scalar & a_matrice
ans =
    0    0    0
    0    0    0
```

Ordinea de efectuare a operatiilor

Nivel prioritate	Operator(i)
1.	Parantezele - ()
2.	Transpusa vectorială (.'), ridicarea la putere vectorială (.^), transpusa conjugată complexă (') și ridicarea la putere (^)
3.	Operatorii unari: plus (+), minus(-) și negație (~)
4.	Operatorii vectoriali de înmulțire (.*), împărțire la dreapta (./), împărțire la stânga (.\), și operatorii clasici de înmulțire (*), împărțire la dreapta (/) și împărțire la stânga (\)
5.	Adunarea (+) și scăderea (-) adică operatorii binari plus și minus
6.	Operatorul două puncte (:)
7.	Operatorii relaționali: mai mic (<), mai mic sau egal (<=), mai mare (>), mai mare sau egal (>=), egal (==) și diferit (~=)
8.	Operatorul logic AND (&)
9.	Operatorul logic OR ()
10.	Operatorul logic AND cu scurtcircuitare (&&)
11.	Operatorul logic OR cu scurtcircuitare ()

Verificarea sintaxei în momentul introducerii codului

Atât în cazul scrierii codului în fereastra de comandă cât și în cazul scrierii de algoritmi în fișiere specifice, limbajul MATLAB folosește un cod de culori care permite identificarea facilă a tipului de text introdus. Ca și culori predefinite MATLAB-ul folosește:

- Albastru pentru cuvintele cheie;
- Mov pentru string-uri (secvențe de text);
- Maro pentru string-uri neterminate;
- Verde pentru comentarii;
- Negru pentru restul textului.



Coduri de eroare

La introducerea unei linii de comandă greșite ca sintaxă sau ca și operație matematică, limbajul MATLAB va furniza întotdeauna un mesaj de eroare contextual, intuitiv, în care va încerca, pe lângă semnalarea erorii și o motivare a naturii acesteia care de cele mai multe ori indică și soluția de corecție.

Exemple de erori în MATLAB

```
>> a=[1 1; 1 1]
```

```
a =
```

```
    1    1  
    1    1
```

```
>> b=[2 2]
```

```
b =
```

```
    2    2
```

```
>> a*b
```

```
Error using *
```

```
Inner matrix dimensions must agree.
```

```
>> b*a
```

```
ans =
```

```
    4    4
```

```
>> sing(pi/2)
```

```
Undefined function 'sing' for input arguments of type 'double'.
```

```
Did you mean:
```

```
>> sin(pi/2)
```

Caractere speciale în MATLAB

Parantezele rotunde: ()

Paranteze rotunde.

1. Indică ordinea de efectuare a operațiilor în expresiile aritmetice:

$(2+3) * 2$

2. Se folosesc pentru a încadra argumentele unei funcții: $\sin(x)$

3. Se folosesc ca metodă de referire a elementelor din vectori sau matrice, pe baza indicilor:

$a(2)$ – al doilea element din șirul a

$M(1, 5)$ – al cincilea element de pe prima linie a matricei M

Caractere speciale în MATLAB

Parantezele drepte: []

Sunt folosite pentru definirea (formarea) vectorilor și matricelor.

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

A =

1.00	2.00	3.00
------	------	------

```
>> B=[1; 2; 3]
```

B =

1.00
2.00
3.00

```
>> C=[1 2 3; 3 2 1]
```

C =

1.00	2.00	3.00
3.00	2.00	1.00

```
>> whos A B C
```

Name	Size	Bytes	Class	Attributes
A	1x3	24	double	
B	3x1	24	double	
C	2x3	48	double	

Caractere speciale în MATLAB

Acoladele: {}

Acoladele.

Se folosesc pentru a forma vectori de celule. De asemenea, pot fi utilizate pentru a referi elemente din aceste tipuri de date.

```
>> AA={'Acesta' 'este' 'un'; 'exemplu' 'de' 'celula'}
```

```
AA =
```

```
    'Acesta'    'este'    'un'  
    'exemplu'    'de'    'celula'
```

```
>> AA{2,3}
```

```
ans =
```

```
celula
```


Caractere speciale în MATLAB

Semnul egal: =

Este folosit în operațiile de atribuire, unde elementului din stânga i se atribuie rezultatul final al evaluării termenului din dreapta. Trebuie subliniat faptul că utilizarea lui pentru a compara două elemente este greșită, în acea situație folosindu-se un alt operator.

```
>> a=3;  
>> b=5;  
>> c=2*(a+b)  
c =  
    16.00  
  
>> c=c-2  
c =  
    14.00
```

Caractere speciale în MATLAB

Semnul apostrof: `

1. Se folosește pentru a calcula transpusa unei matrice.
2. Se folosește ca și caracter care marchează începutul și sfârșitul unui text.

```
>> A=[1 2 3 4]
```

```
A =
```

```
    1.00    2.00    3.00    4.00
```

```
>> B=[1 2 4 6; A]
```

```
B =
```

```
    1.00    2.00    4.00    6.00
```

```
    1.00    2.00    3.00    4.00
```

```
>> A'
```

```
ans =
```

```
    1.00
```

```
    2.00
```

```
    3.00
```

```
    4.00
```

Caractere speciale în MATLAB

Semnul apostrof: `

```
>> B'
```

```
ans =
```

```
1.00
```

```
1.00
```

```
2.00
```

```
2.00
```

```
4.00
```

```
3.00
```

```
6.00
```

```
4.00
```

```
>> C='exemplu de text'
```

```
C =
```

```
exemplu de text
```

Caractere speciale în MATLAB

Semnul punct: .

1. Se folosește ca și separator al părții întregi și a celei zecimale a unui număr real.
2. În combinație cu un operator aritmetic clasic, definește operațiile element cu element pentru șiruri și matrice.
3. Accesează un câmp dintr-o structură.

```
>> a=1.2
```

```
a =
```

```
1.20
```

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

```
A =
```

```
1.00
```

```
2.00
```

```
3.00
```

```
>> B=[4 5 6]
```

```
B =
```

```
4.00
```

```
5.00
```

```
6.00
```

```
>> A.*B
```

```
ans =
```

```
4.00
```

```
10.00
```

```
18.00
```

Caractere speciale în MATLAB

Semnul punct: .

```
>> Exemplu.nume='Test';  
>> Exemplu.valoare=100;  
>> Exemplu(2).nume='Alt Test';  
>> Exemplu(2).valoare=200;  
>> Exemplu.nume
```

```
ans =
```

```
Test
```

```
ans =
```

```
Alt Test
```

```
>> whos Exemplu
```

Name	Size	Bytes	Class	Attributes
Exemplu	1x2	616	struct	



Caractere speciale în MATLAB

Semnul doua puncte: ..

Folosit pentru a referi directorul părinte al celui curent.

Caractere speciale în MATLAB

Semnul trei puncte: ...

Trei puncte.

Permite întreruperea codului și continuarea lui pe un rând nou.

```
>> A=[1 2; 3 4];
```

```
>> B=[1 2; ...  
3 4];
```

```
>> A
```

A =

1.00	2.00
3.00	4.00

```
>> B
```

B =

1.00	2.00
3.00	4.00

Caractere speciale în MATLAB

Semnul virgula: ,

Virgula.

Se folosește ca și element separator pentru o enumerare de mai multe elemente.

```
>> a=[1, 2, 3, 4]
```

```
a =
```

```
1.00
```

```
2.00
```

```
3.00
```

```
4.00
```

```
>> b=4, c=5
```

```
b =
```

```
4.00
```

```
c =
```

```
5.00
```


Caractere speciale în MATLAB

Semnul punct si virgula: ;

Punct și virgula.

1. Folosit la finalul unei expresii sau a unei operații va suprima afișarea acesteia în fereastra de comandă.
2. La definirea vectorilor și matricelor, se folosește pentru a face trecerea la o nouă linie.

```
>> a=[1; 2; 3]
```

```
a =
```

```
1.00
```

```
2.00
```

```
3.00
```

```
>> b=[4 5 6; 7 8 9];
```

```
>> b
```

```
b =
```

```
4.00
```

```
5.00
```

```
6.00
```

```
7.00
```

```
8.00
```

```
9.00
```

Caractere speciale în MATLAB

Semnul doua puncte: :

1. Se folosește pentru a defini un șir, o linie sau coloană a unei matrice.
2. Se folosește pentru genera un șir de valori care respectă o regulă comună, sub forma:

`a:p:b` – generează un vector de valori de la `a` la `b` cu pasul `p`

`a:b` – generează un vector de valori de la `a` la `b` cu pasul 1

```
>> a=[1:4; 10:-1:7]
```

```
a =
```

1.00	2.00	3.00	4.00
10.00	9.00	8.00	7.00

```
>> a(:,1)
```

```
ans =
```

1.00
10.00

```
>> a(2,:)
```

```
ans =
```

10.00	9.00	8.00	7.00
-------	------	------	------

Caractere speciale în MATLAB

Semnul procent: %

Procent.

Marchează sfârșitul logic la unei linii. Orice text introdus pe acea linie după caracterul % nu este luat în considerare din punct de vedere al algoritmului.

```
>> a=[1 2 3; 4 5 6] %b=[1 2 3]
```

```
a =
```

1.00	2.00	3.00
4.00	5.00	6.00

```
>> a,b
```

```
a =
```

1.00	2.00	3.00
4.00	5.00	6.00

```
Undefined function or variable 'b'.
```

Caractere speciale în MATLAB

Semnul procent si acolade: `%{` `%}`

Procent și acolade.

Cele două elemente (`%{`) și (`%}`) marchează începutul și respectiv sfârșitul unei zonei de comentariu. Orice text cuprins între cele două elemente va fi considerat comentariu și nu va fi evaluat din punct de vedere al algoritmului.

Caractere speciale în MATLAB

Semnul exclamării: !

Semnul exclamării.

Indică faptul că restul liniei reprezintă o comandă direct spre sistemul de operare.

```
>> !dir
```

```
Volume in drive C has no label.
```

```
Volume Serial Number is 38DB-F2EA
```

```
Directory of C:\Users\Calin\Documents\MATLAB
```

```
12/28/2013  11:29 AM    <DIR>          .
12/28/2013  11:29 AM    <DIR>          ..
06/20/2013  11:51 AM    <DIR>          Apps
12/26/2013  01:32 PM                2,007 Untitled1.prj
12/26/2013  01:33 PM                2,007 Untitled2.prj
                2 File(s)                4,014 bytes
                3 Dir(s)  17,165,955,072 bytes free
```

Caractere speciale în MATLAB

Semnul arond: @

Arond.

Este folosit ca și pointer spre o funcție. Astfel, funcția poate fi utilizată în mod indirect.

```
>> a=@sin;, b=@cos;
```

```
>> a(pi/6)^2+b(pi/6)^2
```

```
ans =
```

```
1.00
```