

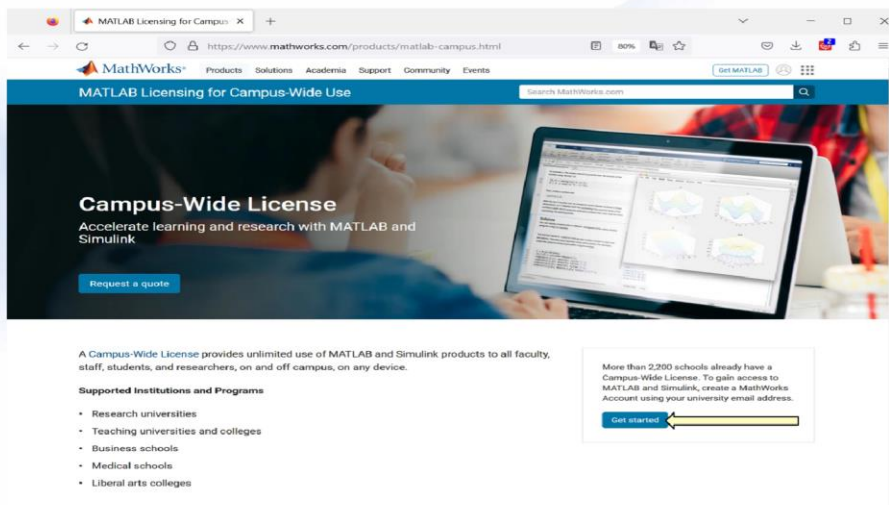
Semnale și sisteme

Lucrare de laborator nr. 1

Introducere in Matlab

Instalare Matlab

<https://www.mathworks.com/products/matlab-campus.html>



Instalare Matlab



Introducere in Matlab (MATrix LABoratory)

- ❑ Functii Matlab de interes general: *doc, help, whos, what, lookfor, demo, clock, date, cputime, tic/toc*
- ❑ Formate de afisare a numerelor: *format short / long /short e /long e / short g / long g / hex / + / bank / rat*
- ❑ Comenzi pentru spatiere: *format compact / loose*
- ❑ Functii pentru controlul formatului de citire/scriere: *sscanf, fscanf / sprint, fprintf*
- ❑ Functii speciale si constante: *ans, pi, i, j, Inf, NaN*

Variabile in Matlab

- ❑ Matlab este *case sensitive* (distinge între litere mari și litere mici).
- ❑ nu sunt necesare declarații de tip sau de dimensiune pentru variabile
- ❑ toate variabilele sunt *array*-uri multidimensionale; o *matrice* este un array 2D
- ❑ un scalar este reprezentat sub forma unei matrice de dimensiune 1×1 ; un vector linie sub forma unei matrice $1 \times N$; un vector coloană sub forma unei matrice de dimensiune $M \times 1$
- ❑ elementele unei matrice pot fi numere reale sau complexe
- ❑ dimensiunea unei variabile se poate obține utilizând funcțiile *size* și *length*

Exemple de variabile:

```
Scalar real    >> x = 5
Scalar complex >> x = 5 + 10j (sau x = 5 + 10i)
Vector linie   >> x = [1 2 3] (sau x = [1, 2, 3])
Vector coloana >> x = [1; 2; 3]
Matrice 3 x 3  >> x = [1 2 3; 4 5 6; 7 8 9]
```

Generarea matricelor

- ❑ funcții pentru generarea matricelor elementare: *eye*, *zeros*, *ones*, *rand*, *randn*
- ❑ vectorii pot fi generați utilizând operatorul `:`
 - `x = [0 : 0.5 : 10]` % generează un vector x cu valori între 0 și 10 cu pasul 0.5
- ❑ comanda *linspace* poate fi de asemenea utilizată pentru generarea vectorilor
 - `x = linspace(0, 10, 21)` % generează 21 de valori egal distribuite în intervalul [0, 10]

Accesarea elementelor unei matrice

- ❑ elementele unei matrice pot fi accesate specificând numărul liniei și al coloanei
 - `A = [1 2 3; 4 5 6; 7 8 9]` % generarea unei matrice de dimensiune 3x3
 - `x = A(1, 3)` % accesarea elementului din prima linie și a treia coloană din matricea A
 - `y = A(2, :)` % accesarea celei de a doua linii a matricei A
 - `z = A(1 : 2, 1 : 3)` % accesarea unei sub-matrice a lui A care constă din liniile 1 și 2 și toate cele trei coloane

Operatii aritmetice asupra matricelor

- operatii aritmetice de baza asupra matricelor:
 - + adunare, - scadere, * inmultire, ^ ridicare la putere, / impartire la dreapta, \ impartire la stanga, ' transpusa
- adunarea si scaderea a doua matrice implica operatii aritmetice element-cu-element; inmultirea si impartirea nu
- operatiile aritmetice element-cu-element sunt indicate prin utilizarea simbolului '.' inainte de operator:
 - .* inmultire, .^ ridicare la putere, ./ impartire la dreapta, .\ impartire la stanga

Operatori relationali

- < mai mic, <= mai mic sau egal, > mai mare, >= mai mare sau egal, == egal, ~= diferit
- rezultatul aplicarii unui operator relational este de tip logic (adevarat (1) sau fals (0)).
- daca operatorii se aplica unui scalar rezultatul va fi un scalar, aplicand operatorii asupra matricelor, rezultatul va consta intr-o matrice cu valori zero si unu

Operatori logici

- &, && si
- |, || sau
- ~ nu
- operatorii && si || pot fi aplicati doar asupra valorilor scalare
- operatorii & si | pot fi aplicati atat asupra valorilor scalare cat si asupra matricelor

Functii matematice

- **sin** - sinus, **cos** - cosinus, **tan** - tangenta, **exp** - exponential, **log** - logaritm natural, **log10** - logaritm in baza 10, **sqrt** - radacina patrata, **abs** - valoare absoluta, **sign** - semn

Instructiuni de control

- instructiuni iterative: *for*, *while*
- instructiuni de control logic: *if / elseif / else; switch / case / otherwise*
- fiecare dintre aceste instructiuni necesita cuvantul cheie **end** pentru incheierea secventei de control

Numere complexe

- exemple pentru generarea matricelor cu valori complexe

```
B = [ 1 2 ; 3 4 ] + i * [ 5 6 ; 7 8 ]  
B = [1+5i , 2+6i ; 3+7i , 4+8i ]  
B = complex ( [ 1 2 ; 3 4 ] , [ 5 6 ; 7 8 ] )
```

- functii pentru numere complexe

Partea reala a lui x	>> real(x)
Partea imaginara a lui x	>> imag(x)
Amplitudinea (modulul) lui x	>> abs(x)
Argumentul lui x	>> angle(x)
Conjugatul complex	>> conj(x)

Fisiere MATLAB

- *fisierele m* (fisiere cu extensia .m) sunt utilizate atat pentru scripturi cat si pentru functii
 - *fisierele script* sunt secvente de comenzi scrise cu un editor si salvate intr-un *fisier m*; nu accepta parametri de intrare, nu returneaza parametri; toate variabilele utilizate sunt salvate in spatiul de lucru al MATLAB-ului
 - *fisierele functie* sunt utilizate pentru stocarea functiilor; functiile pot primi parametri de intrare si pot returna parametri; variabilele interne sunt locale functiei

```
function [out, out2] = nume_functie(in1, in2)  
%  
%Comentarii pentru a explica ce face functia  
%  
comanda MATLAB 1;  
...
```

- *fisierele mat* (fisiere cu extensia .mat) sunt fisiere binare utilizate pentru a stoca rezultate numerice
 - save file_name.mat var1 var2 % salveaza variabilele var1, var2 in fisierul file_name.mat
 - save file_name.mat % salveaza toate variabilele din spatiul de lucru in fisierul file_name.mat
 - load file_name % incarca fisierul mat file_name

Reprezentari grafice

- ❑ functii pentru reprezentari grafice: *plot*, *subplot*, *stem*
- ❑ comenzi pentru personalizarea graficelor: *xlabel*, *ylabel*, *title*, *grid*, *text*, *axis*
- ❑ alte functii utile: *figure*, *hold on*

Exemplu (bucle *for* vs operatii cu vectori)

```
function x = semnal_sinusoidal
%generarea unui semnal sinusoidal utilizand bucle for
for i = 0:9999
    x(i+1) = sin(2*pi*i/100);
end
```

```
tic
sine_ signal;
toc
```

```
function x = semnal_sinusoidal2
%generarea unui semnal sinusoidal utilizand vectori
x = sin(2*pi*[0:9999]/100);
```

```
tic
sine_ signal2;
toc
```

Exercitii

1. Scrieti functii pentru afisarea urmatoarelor semnale:

a. $y(t) = \begin{cases} -t/2, & t \in [-2, 0) \\ t/2, & t \in [0, 2] \end{cases}$

b. $y(t) = \begin{cases} \sin(t), & \{-10 \leq t < -4\} \cup \{4 \leq t \leq 10\} \\ |\cos(2t)|, & -4 \leq t < 4 \end{cases}$

c. $y(t) = 0.5^t, \quad t \in [0, 10]$

d. $x[n] = \left(1 + \frac{1}{n}\right)^n, n \in [1 \dots 100]$

2. Scrieti o functie pentru calculul valorii factorialului unui numar intreg pozitiv n ($n!$).
3. Scrieti o functie pentru calculul sumei $x(n) = \sum_{k=0}^n \frac{1}{k!}$.
4. Scrieti functii pentru generarea semnalelor $f(x) = 3x^3 + 2x^2 + x$, $g(x) = e^x - 2x + 3$ si un fisier script pentru calculul rezultatului $f(100)/g(10)$.
5. Determinati valorile lui x, y, z pentru urmatorul sistem de ecuatii liniare:

$$\begin{cases} -x + 2y - 3z = -6 \\ 5x - y + 2z = 9 \\ -4x - y + 2z = 0 \end{cases}$$