Laborator #2 (din 7)

martie 2022 **- Matlab -** Prof. dr. ing. Iulian Lupea

1. Matlab: operaţii cu şiruri şi matrice (continuare)

1.1. Operatorul *două puncte* referă toate elementele de pe o linie sau pe coloana unei matrice iar *cuvântul end* referă ultima linie sau coloană din matrice.

E1=**sum(A(: , end))** → calculează suma elementelor de pe ultima coloană a matricei A (oarecare)

Reţineţi următoarele expresii:

j:k exprimă un vector de valori similar cu [j,j+1,...,k] % pas 1 implicit

j:**pa**:k este similar cu expresia: [j, j+pa, j+2*pa, ...,k] % pa este pasul

j:k nu selectează nici un element dacă j > k

j:p:k este mulţimea vidă dacă p == 0, dacă p > 0 şi j> k, sau dacă p < 0 şi j < k

(p, j, şi k sunt scalari).

1.2. Exemple cu şiruri de valori numerice:

Expresia **2: 8** generează valorile: 2 3 4 5 6 7 8 **100 : -7 : 50** generează valorile: 100 93 86 79 72 65 58 51	**0 : pi/4 : pi** generează valorile: 0 0.7854 1.5708 2.3562 3.1416

1.3. Exemple de folosire a operatorului două puncte (:) pentru selectare de linii, coloane, elemente ale vectorilor, matricelor şi tablourilor multidimensionale.

1.3.T: răspundeţi la îndemnurile de forma → *Calculaţi ....*

A(: , j) este coloana j a lui A

→ *Calculaţi suma elementelor de pe coloana indice 2 dintr-o matrice*

A(i , :) este linia i a lui A

→ *Calculaţi suma elementelor de pe linia indice 3 dintr-o matrice*

A(: , :) este referire la întregul tablou 2D (identic cu A dacă A este matrice).

A(j : k) este A(j), A(j+1),...,A(k) lista elementelor din matrice luate pe coloane (col. 1, col.2, ...) formează un şir.

→Calculaţi suma pătratelor elementelor de pe primele 2 coloane dintr-o matrice **a** pătratică 3x3.

→Calculaţi suma pătratelor de pe primele două linii dintr-o matrice **a** pătratică 3x3.

A(: , j:k) selectează coloanele de la j la k: A(:, j), A(:, j+1),...,A(:, k)

Elementele dintr-o matrice **a** (3x3) luate pe linii se pot copia într-un şir b → **b(1: 9)=a'**

A(:, :, k) este pagina întreagă (2D) indice k a matricei A cu trei dimensiuni.

<table>
<tr><td rowspan="3">Extrageţi valori din matricea a

» a=[1 2 3 ; ...
3 4 5 ; ...
6 7 8]</td><td>a(: , 1 : 2)
1 2
3 4
6 7</td><td>a(1:2,:)
1 2 3
3 4 5</td><td>a(1 : 2, 3)
3
5</td></tr>
<tr><td colspan="2">a(1:7)
1 3 6 2 4 7 3</td><td>a(:)'
1 3 6 2 4 7 3 5 8
(' transpus)</td></tr>
<tr><td colspan="2">a(2,5)=9; % se adaugă 2 coloane</td><td>a(2, [1:3,5])
ans → 3 4 5 9</td></tr>
</table>

1.4. Referirea unor porţiuni de matrice

A(1: k, j) sunt primele k elemente ale coloanei j din A.

sum(A(1:4, 4)) calculează suma elementelor selectate de pe coloana 4-a

1.4.T: Comanda A(:,:,2) = pascal(4) generează un **tablou 3D** cu prima pagină conţinând numai elemente nule.

A(: , : , 1) =	A(: , : , 2) =
0 0 0 0	1 1 1 1
0 0 0 0	1 2 3 4
0 0 0 0	1 3 6 10
0 0 0 0	1 4 10 20

Adăugaţi pagina a 3-a cu o matrice **hadamard** 4x4 iar în pagina a 4-a puneţi o matrice 4x4 creată cu funcţia **magic** (sumele pe linii si sumele pe coloane au aceeaşi unică valoare).

Se umple *prima pagină* astfel:

prima linie va fi prima linie din matricea **pascal**,
a doua linie va conţine coloana a doua din matricea Hadamard,
a treia linie va conţine suma pe linii din matricea creată cu **magic**, iar
a patra linie din A va conţine suma pe coloane din inversa matricei Hadamard
(h=hadamard(4)^-1; sum(h(:,:))).

>> A(:,:,1) →	1 1 1 1 1 -1 1 -1 34 34 34 34 1 0 0 0

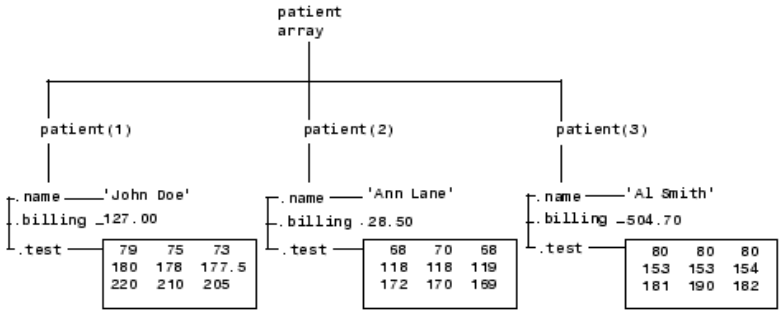
1.5. Stergerea unor linii şi coloane
Iniţializaţi matricea (2D), X=pascal(3). Dacă încercăm să ştergem un singur element dintr-o matrice rezultă eroare fiindcă nu se obţine o matrice. Se va şterge linia a doua a lui X:

<table>
<tr><td rowspan="3">X(1,2) = [ ] % => eroare

X(2,:)=[ ] % sterge linia #2
X =
1 1 1
1 3 6</td><td colspan="3">a=[1 2 3; % se defineste matricea a
3 4 5;
6 7 8]</td></tr>
<tr><td rowspan="2">a(:,1)=[ ]
2 3
4 5
7 8</td><td>%vizualizare
a(1:2,:) =[ ]
6 7 8</td><td>%vizualizare zona
a(1:2, 2:3)
2 3
4 5</td></tr>
<tr><td>a(1,1)=[ ] % eroare</td><td>a(1:2, 2:3)=[ ]
% eroare</td></tr>
</table>

Răspunsuri: sum(a(1:6).^2); sum([a(1,:) a(2,:)].^2)

2. Date structurate: tablou de Structuri de date şi **cell arrays**
2.1. Să se genereze datele structurate din figura alăturată folosind *student* în loc de *patient* şi nume studenţi din grupă în locul numelor din figură.
student.nume=’prenume nume’
student.billing= -123.0
student.test=[1 1 1;2 2 2;3 3 3]
student(2).nume=’prenume2 nume2’
etc.

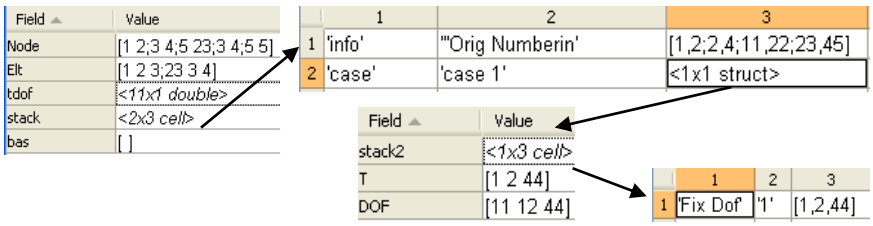


2.1.T: Creaţi poziţia student(4) cu **nume** Abcd. Câmpul **billing** este suma câmpurilor billing ale elementelor precedente din tabloul de structuri >> sum([stud(:).val]); în mod similar creaţi câmpul **test:** stud(4).test=stud(1).test+stud(2).test+stud(3).test; Extrageţi porţiuni din matricea .test.
Obs.: Stergerea unui câmp al structurii: **numestr=rmfield**(numestr, 'numefield')
numestr=rmfield(numestr, {'numefield1','numefield2')

2.2. Tabloul de celule este folosit **la stocarea indexată** a datelor de tipuri diferite. Putem spune că celulele sunt câmpuri fără nume (dar indexate pentru a putea fi referite).
Exemplu: **cell_a** va avea **2 linii**, **3 coloane**:
»cell_a={2:7 'abc'};
»cell_a{2,1}=sind(30);
»cell_a(2,2)={exp(1)}; % cell_a contine acum 2 lin. şi 2 col.
Referiri: »cell_a{1,2}(1:2) [Enter], »cell_a{1} [Enter],
»cell_a{1,3}=cell_a{1}(2:4) * cell_a{2,1} [Enter]

2.3. Să se construiască prin comenzi lansate în fereastra de comenzi următoarea structură de date WF. **Stuctura WF** are 5 câmpuri:

NumeCâmp	*date*
.Node	5x2 double
.Elt	2x3 double
.tdof	11x1 double
.stack	2x3 cell
.bas	[ ]



Câmpul unei structuri se referă în forma: WF.**Elt** şi se iniţializează ca în exemplul precedent (2.1).
Câmpul ***.bas*** va fi iniţializat cu şirul vid.
Câmpul **.stack** este un ***cell array* (structură indexată) cu 2 linii şi 3 coloane**. Elementele sunt referite ca pentru matrice cu două dimensiuni, **indicii fiind între *acolade***.
Cele **şase celule** conţin următoarele date:
'info' 'Orig Numbering' <4 x 2 int32>
'case' 'case 1' <**1 x 1 struct** >
Celula {2,1} este iniţializată astfel:
>> WF.**stack**{2,2}='case 1'
Celula {2,3} este o structură (singulară 1x1) având câmpurile: →

Câmpurile structurii <1x1 struct>	
NumeCâmp	*date*
.stack2	<1x3> cell
.T	[ ]
DOF	[ ]

Câmpul .T poate fi iniţializat cu tabloul vid: >> WF.stack{2,3}.T=[ ];
Câmpul stack2 conţine un tablou (1x3) de celule cu conţinutul: 'Fix Dof' , '1', <3x1 double>
Iniţializăm prima celulă a structurii astfel: WF.stack{2,3}.stack2{1,1}='Fix Dof'
2.3.T: Continuaţi definirea întregii structuri de date. Verificaţi conţinutul în fereastra Workspace.

3. Exemple de funcţii

Se deschide fereastra edit; folosiţi acelaşi nume pentru fisier şi funcţia salvată.

```
function rep_sin  %functie fara parametri
t=-0.01:.0005:.03; fr=100; om=2*pi*fr;
x=sin(om*t)
subplot(2,1,1)
stem(t,x); title('stem plot');
subplot(2,1,2)
plot(t,x); title('grafic linie continua')
```

Un parametru de intrare şi unul de ieşire

```
function y = variance(x)
       %completati corp functie cu calcul VAR
mu = sum(x)/length(x);
     . . . ;
y = . . .;
```

$$VAR = \sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(X[i]-\mu)^2$$

```
function y=pol(x)
y=2*x.^3+7*x.^2-3*x+1
```

```
function y = movile2(x)
y = 1./((x-.7).^2 + .01) + 1./((x-1.5).^2 + .04) - 6;
```

3.1.Funcţia rep_sin nu returnează valoare. Observaţi în paralel funcţiile stem şi plot.
3.2.Apelaţi pol(5), pol([1 2 3]), pol([1:12]) sau pol([1,2;3,4])
3.3.Funcţia movile2 poate fi evaluată pentru abscise în intervalul $0 \le x \le 2.5$ cu comenzile:

```
x = 0 : .01 : 2.5;
y = movile2(x);
```

şi apoi se poate reprezenta grafic funcţia cu comanda: plot(x,y)
3.4.Completaţi corpul funcţiei *variance* (folosiţi 1/(n-1) în loc de 1/n în expresia VAR). Pentru verificarea corectitudinii valorii returnate de funcţia *variance (dispersia valorilor în jurul mediei)* apelăm funcţia MATLAB *var*. Se exemplifică pentru » x = [1 2 3 4];

3.4.T. Incărcaţi în Workspace tabloul date2Ch de valori reale 25000x2 din fişierul AccTah.mat (folosiţi load('AccTah')). Reprezentaţi grafic cele două coloane considerând timpul de achiziţie a semnalului de 5 secunde (vezi linspace) şi calculaţi varianţa pentru fiecare coloană din tablou.

3.5.T. Apelaţi acele funcţii de mai sus care returnează valoare/valori în cadrul unor expresii (astfel încât valorile returnate să participe în continuare la calcule în cadrul expresiei); plot(x**.***movile2([0:.01:2.5]))