

Laborator #5 (din 7)

Prof.dr.ing. Iulian Lupea

DECIZII MULTIPLE cu **if-elseif**

1. Contorizați numărul de apariții în șirul **x** a valorilor 1, 2, nr. valori într-un interval dat și restul de valori.

<pre> x=input('Tastati un sir de valori x=[...] '); v1=0; v2=0; v3=0; v=0; for i=1:length(x) if x(i)==1 v1=v1+1; elseif x(i)== 2 v2=v2+1; elseif 3<=x(i) & x(i) <5 v3=v3+1; else v=v+1; end end fprintf('v1=%d v2=%d v3=%d v=%d\n', v1,v2,v3,v); </pre>	<p>Observați asemănarea dintre decizia multiplă realizată cu if-elseif și cea realizată cu switch-case</p> <p>1.T1. Rescrieți exemplul folosind și instrucțiunea switch-case</p> <p>1.T2. Rescrieți exemplul numai cu if.</p>
---	---

DEFINIȚII ȘI APELURI DE FUNCȚII

2. Un parametru de intrare și unul de ieșire

<pre> function y = average(x) % AVERAGE media elementelor vectorului y = sum(x) / length(x); end </pre>	<p>x = parametrul de intrare este vector y = parametrul de ieșire este scalar Salvați funcția average în fișierul average.m</p>
---	--

APELURI valide ale funcției:

<pre> v=[1 2 3 -3 5]; ave=average(v) </pre>	<pre> v=[1 2 3 -3 5]; average(v) ans = 1.6000 </pre>	<pre> average([1 2 3 -3 5]) ans = 1.6000 </pre>
---	---	---

3. Doi parametri de intrare și un parametru de ieșire

<pre> function z=xplusy(x,y); z=x+y; end </pre>	<p>x, y doi parametri de intrare z un parametru de ieșire Se salvează definiția funcției într-un fișier cu numele <i>xplusy.m</i></p>
---	--

APELURI valide ale funcției:

<pre> a=1 b=2 scal1=xplusy(a,b) scal2=xplusy(1,2) </pre>	<pre> a=[1 2 3; 3 4 5] b=[7 8 9; 2 0 4] matr1=xplusy(a,b) matr1=xplusy([1 2 3; 3 4 5] , b) </pre>	<pre> a=7 b=[7 8 9; 2 0 4] %apel în expresie matr2=xplusy(a,b) * b' matr2=xplusy(7, [7 8 9; 2 0 4]) </pre>
--	---	---

4) **Parametri de intrare** (2,3 sau 4), mai puțini sau egali cu numărul **param. formali** (4)

<pre> function z = xplusy2(x, y, a, b) fprintf('Nr par. actuali=%d\n',nargin) if nargin < 4 b=1; end if nargin < 3 a=1; end z=(x+y)*a*b; end </pre>	<p>% se poate apela funcția cu 2, 3 sau 4 parametri</p> <p>nargin conține nr. de param. actuali folosiți la apel</p> <p>Dacă nargin<4 se atribuie param. neglijat val. b=1 Dacă nargin<3 se atribuie param. neglijat val. a=1</p> <p>Pentru param. >4 sau <2 => eroare</p>	
<p>Apeluri valide:</p> <pre> >> xplusy2(3,5) ans = 8 </pre>	<pre> >> xplusy2(3,5,2) ans = 16 </pre>	<pre> >> xplusy2(3,5,2,4) ans = 64 </pre>

5. Un parametru intrare, 2 parametri de ieșire

```
function [s, sp]=sumapoz(x)
%returneaza suma si suma elem>0
s=0; sp=0;
for i=1:length(x)
    s=s+x(i);
    if x(i)>0
        %insumare x(i)>0
        sp=sp+x(i);
    end
end
```

Apel:
[suma, sp]=sumapoz([1 5 -3])

```
function [mean2, stdev] = stat2(x)
n = length(x);
mean2 = sum(x)/n;
stdev = sqrt( sum( (x-mean2).^2 ) /n );
end
```

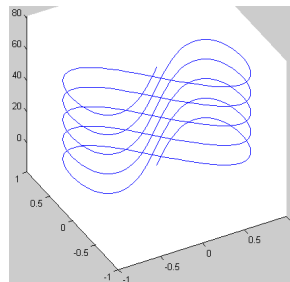
Apel: [media, dev_stand]=stat2([1 2 3])

$$stdev = \sqrt{\frac{1}{n} \sum_{i=1}^n (X[i] - mean2)^2}$$

REPREZENTĂRI GRAFICE SPATIALE

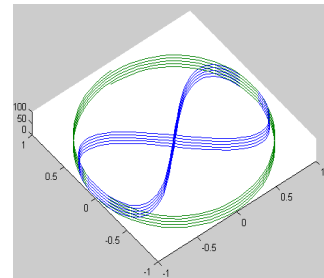
6. Curbă în spațiu 3D:

```
t=0 : pi/40 : 10*pi;
plot3(sin(t), sin(t).*cos(t), 2*t);
grid on;
```



6.2. Două curbe spațiale:

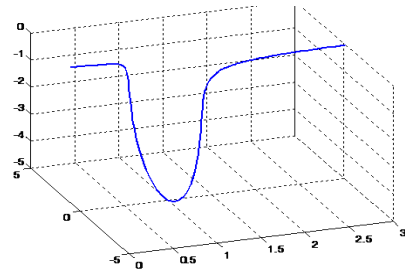
```
plot3(sin(t), sin(t).*cos(t), 2*t, sin(t), cos(t), 2*t); sau:
plot3(sin(t), sin(t).*cos(t), 2*t, '+', sin(t), cos(t), 2*t, '-');
%prima curba trasata din '+', a doua din '-'
```



7. Curbă 3D: funcție complexă, variabilă reală

```
ze=0.1; r=0:.01:3;
Ac=1./(1-r.^2+2*j*ze*r);
% factor complex amplificare deplasare a0
plot3(r, real(Ac), imag(Ac), 'linewidth', 2.5); grid on;
```

$$Ac = \frac{1}{1 - r^2 + 2j \cdot ze \cdot r}$$



8. Grafic suprafață în 3D

Graficul funcției z în domeniul dreptunghiular →

$$z = \sin\left(\sqrt{x^2 + y^2}\right) / \sqrt{x^2 + y^2}$$

a) Se creează rețeaua de puncte din planul Oxy:

```
[X,Y]=meshgrid(-12:4:12, -5:4:10);
% par1, par2=vector de valori după axele Ox respectiv Oy
% X,Y=matrice cu coord. x respectiv y ale punctelor din rețeaua de pte.
```

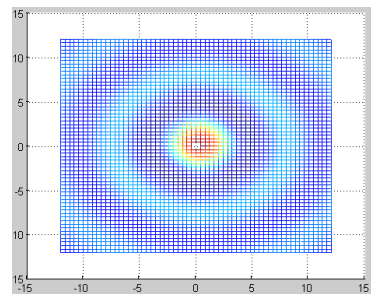
```
ra=sqrt(X.^2+Y.^2); %ra este matrice de valori intermediare
%matricele X și Y sunt ridicate la pătrat element cu element,
%adunate, apoi radicalii din fiecare element salvați în matricea ra
```

b) calcul valori funcție, păstrate în matricea Z

```
Z=sin(ra)./ra;
%sau direct: Z2=sin(sqrt(X.^2+Y.^2))./sqrt(X.^2+Y.^2);)
```

c) generare suprafață 1, h este handle la grafic

```
subplot(211);
```



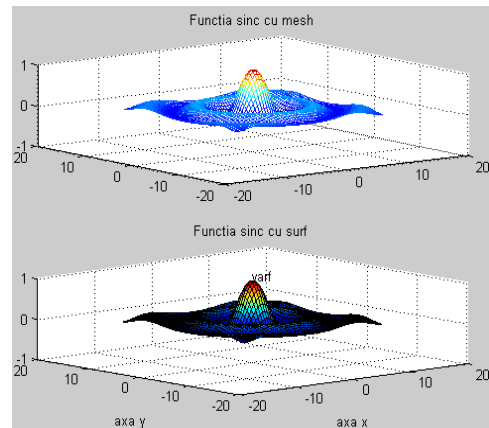
h=mesh(X,Y,Z); grid on;

d) generare suprafață2 (vezi figura)

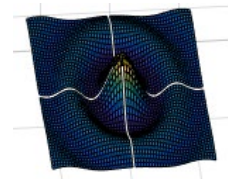
subplot(212);

s2=surf(X,Y,Z);

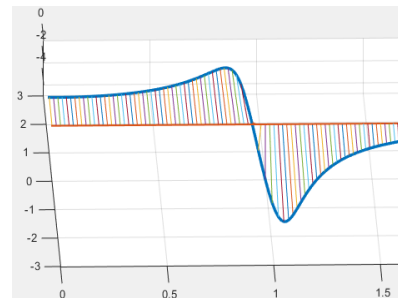
% matricea X conține coordonatele x ale tuturor punctelor din grafic; în mod similar matricele Y și Z conțin coordonatele y resp. z ale tuturor punctelor din grafic.



9.T. Să se traseze suprafața de mai sus din patru porțiuni ale suprafeței întreaga p1: $x < 0, y > 0$; p2: $x < 0, y < 0$; p3: $x > 0, y < 0$ și p4. Porțiunile p1, p2, p3, p4 se trasează/adaugă succesiv preluând zone din matricele X, Y, Z.



10.T. Considerăm graficul de la pct. 7. Se vor adăuga linii care leagă puncte echidistante de pe axa **Or** cu puncte de pe curba trasată (puncte de aceeași abscisă). Obs. se poate folosi line sau plot3 într-un ciclu.



11. Animație sistem masă-arc

clear;clf;

t=linspace(0, 5, 150); % timp de simulare t=5s in 150 pasi

axis([-10 15 -30 160]);

line([-3,8],[0 0],'linewidth',2); % linie capat fix arc

line([-8, 15],[100,100],'linewidth',.5); % linie echilibru

%CALCUL DEPLASARE MASA

A=40; Fi=pi/2; f=1; %frecventa din w=2*pi*f

x_de_t=A*cos(2*pi*f*t-Fi); %deplasarea masei

for k=1:length(t)

% trasare arc

[abs, ord]=arcEli(0,0,8,100+ x_de_t(k)); % pune in [abs ord] coordonate arc

hArc=line(abs, ord, 'linewidth',1.5);

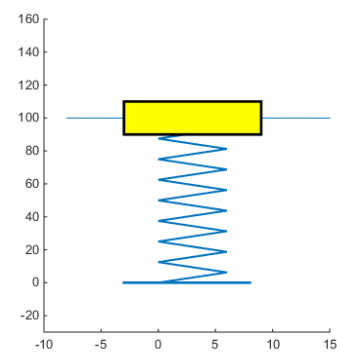
% trasare masa

hRmasa=rectangle('Position',[abs(end)-3, ord(end)-10,12,20], 'LineWidth',2, 'FaceColor','y');

pause(.01); set([hArc hRmasa], 'visible','off');

end

set([hArc hRmasa], 'visible','on');



function [abs,ord]= arcEli(origx,origy,spire, lun)

abs(1)=origx; ord(1)=origy;

...

for

.....

end