

Laborator #10, Dec 2009

1. Funcții pentru reprezentări grafice 2D în MATLAB

subplot(abc) divizează graficul a=număr linii, b=număr coloane, c=poziția *curentă* în matrice numărând pe linii: a11, a12... a21...

plot(y) realizează reprezentări grafice în **coordonate liniare**

a) y este vector => reprezintă valorile lui y în funcție de numărul de ordine al fiecărui element din vector

b) y este matrice m x n => plot(y) reprezintă n curbe câte o curbă pentru fiecare coloană a matricei

```
x=0 : pi/100 : 4*pi %vector real de 400
                     %elemente
```

```
y=sin(x);
```

```
plot(y)
```

```
    Exemplu
```

```
x=0 : pi/100 : 3*pi
```

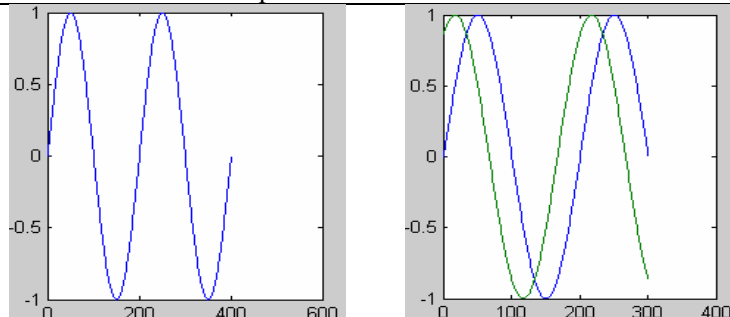
```
y=sin(x);
```

```
y1=sin(x+pi/3);
```

```
ma(:,1)=y; %generare coloana 1
```

```
ma(:,2)=y1; %generare coloana 2
```

```
plot(ma);
```



c) dacă y este un vector de valori complexe va reprezenta plot(real(y), imag(y))

```
x=0 : pi/100 : 3*pi %vector real
```

```
y=sin(x);
```

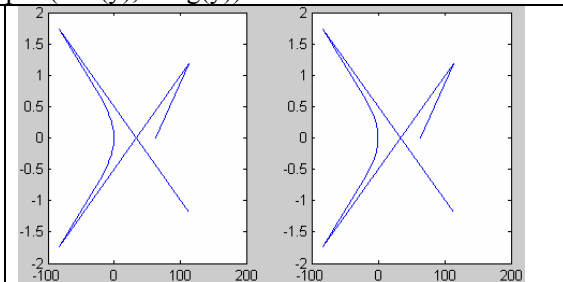
```
z=fft(y); % transformata Fourier calculează vectorul z
           % având length(y) valori complexe(simetrice)
```

```
subplot(121) % linia 1, coloana 2, pozitia 1 devine curenta
```

```
plot(z)
```

```
subplot(122) % linia 1, coloana 2, pozitia 2 devine curenta
```

```
plot(real(z), imag(z)) % grafic identic cu plot(z)
```



plot(x,y)

a) reprezintă grafic vectorul y (ordonata) în funcție de vectorul x (abscisa)

b) x vector y matrice => coloanele matricei y vor fi reprezentate în funcție de vectorul x

plot(x1,y1, x2,y2) => în același grafic sunt reprezentate curbele (x1, y1) și (x2, y2)

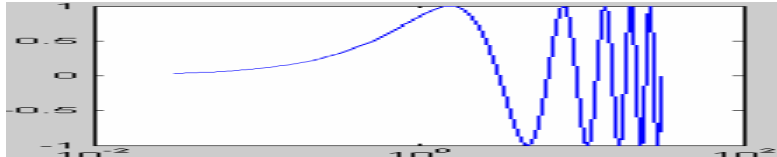
Reprezentare în coordonate logaritmice și semilogaritmice

loglog(x,y) scalează ambele axe folosind logaritmul în baza 10 (...10⁻¹, 10⁰, 10¹, 10², ...sunt echidistante)

semilogx(x,y) scalează logaritm axa x (...10⁻¹, 10⁰, 10¹, 10², ...sunt echidistante pe x)

semilogy(x,y) scalează logaritm axa y (...10⁻¹, 10⁰, 10¹, 10², ...sunt echidistante pe y)

```
x=0 : pi/100 : 10*pi
    %vector real de 1000 elemente
y=sin(x);
semilogx(x, y)
```

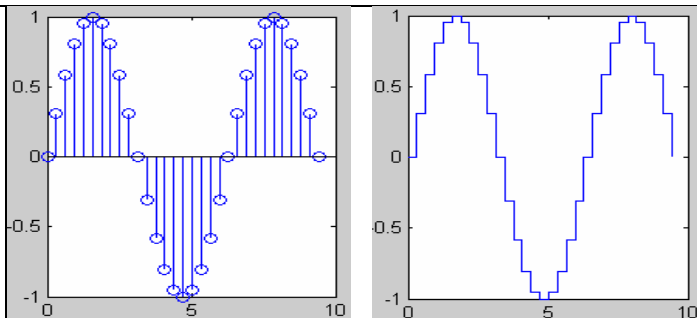


Reprezentarea **discretă** a datelor

```
x=0 : pi/10 : 3*pi %vector real de elemente
y=sin(x);
stem(x, y)
```

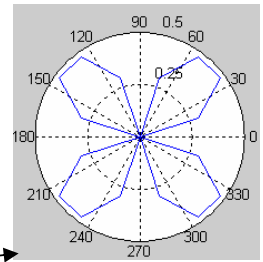
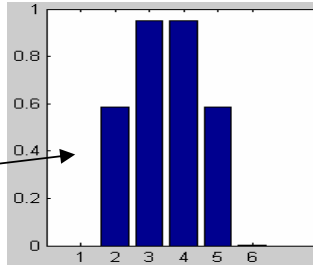
Reprezentare în **trepte**

```
x=0 : pi/10 : 3*pi %vector real
y=sin(x);
stairs(x, y)
```



Grafic de bare

```
x=0 : pi/5 : pi %vector real
y=sin(x);
subplot(1,2,1)
bar(x,y)
subplot(1,2,2)
bar(y)
```



Reprezentare în coordonate **polare** (u,R)

```
u=0 : pi/10 : 2*pi;
R=sin(u) .* cos(u);
polar(u,R);
```

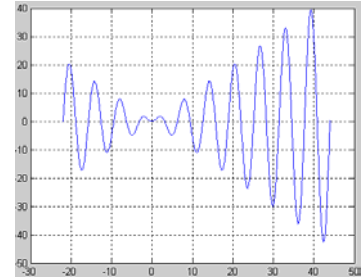
fplot('nume_fun', limite, n) reprezintă grafic o funcție cu limite specificate

```
function y=f2(x) % x poate fi vector
y=sin(x).*x; % salvam functia in fisierul f2
```

```
>>[x,y]=fplot('f2', [-22 44], 1000); grid;
```

Funcția **fplot** returnează 1000 abscise și 1000 ordonate în vectorii **x** și **y**. Putem apela **plot(x,y)**.

comet(y) și **comet(x,y)** realizează reprezentare dinamică a unei curbe 2D printr-un punct care trasează curba.



2. Depanarea programelor (debug)

1. se deschide (open) fișierul .m sau funcția, 2. se setează minim un punct de oprire a rulării **breakpoint** (bulină roșie) 3. Se rulează (run) și se continuă (step) pas cu pas prin program până la terminarea rulării (folosind butoanele step, step in, step out); o săgeată (verde) indică instrucțiunea curentă 4. în timpul rulării pas cu pas urmărim valoarea variabilelor: în fereastra *workspace* pentru fișiere .m sau ținem cursorul peste o variabilă în fereastra edit afișându-se într-un chenar valoarea acelei variabile și tipul de dată asociat.

3. Tipărire cu format, funcția **fprintf**

```
x=input('vector x= ');
for i=1:length(x)
    fprintf('element x(%3d)=%12.3f\n',i , x(i) );
end
```

Pentru scriere în fișier se folosește secvența de mai jos.

```
x=input('vector x= ');
fid=fopen('fisdate.dat', 'wt');
for i=1:length(x)
    fprintf(fid,'element x(%3d)=%12.3f\n',i , x(i) )
end
```

Tipărire în formă tabelară:

```
x=input('medii x= ');
s=0; m=0; ma=0;
nivel={'slab', 'mediu', 'bun'}; %cell array
fprintf('      Note student\n');
fprintf('      ----- \n');
for i=1:length(x)
    if x(i)<=6
        s=s+1;
        fprintf('student #%d nivel %s =%12.3f s=%d\n',i,nivel{1}, x(i), s)
    elseif x(i)<=8
        m=m+1;
        fprintf('student #%d nivel %s =%12.3f m=%d\n',i,nivel{2}, x(i), m)
    else
        ma=ma+1;
        fprintf('student #%d nivel %s =%12.3f m=%d\n',i,nivel{3}, x(i), ma)
    end
end
```