

Cuprins

Grafica 2D.....	1
1. Grafice de bază	1
2. Axe și adnotarea.....	14

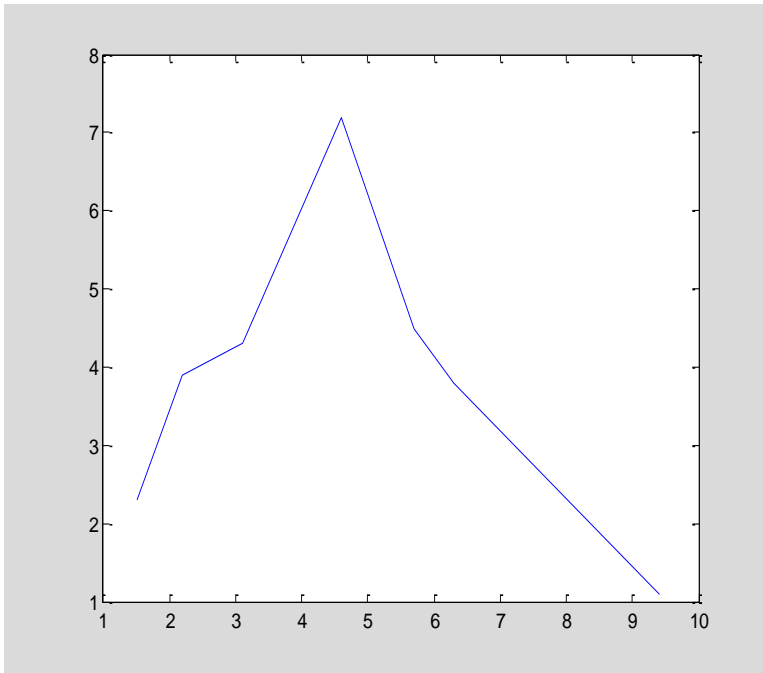
Grafica 2D

MATLAB are facilități grafice puternice și versatile. Se pot genera grafice și figuri relativ ușor, iar atributele lor se pot modifica cu ușurință. Nu ne propunem să fim exhaustivi, ci dorim doar să introducem cititorul în facilitățile grafice MATLAB care vor fi necesare în continuare. Figurile existente pot fi modificate ușor cu utilitarul Plot Editor. Pentru utilizarea sa a se vedea `help plottedit` și meniul Tools sau bara din ferestrele figurilor.

1. Grafice de bază

Funcția MATLAB `plot` realizează grafice bidimensionale simple unind punctele vecine.

```
x=[1.5, 2.2, 3.1, 4.6, 5.7, 6.3, 9.4];  
y=[2.3, 3.9, 4.3, 7.2, 4.5, 3.8, 1.1];  
plot(x,y)
```



MATLAB deschide o fereastră pentru figură (dacă una nu a fost deja deschisă ca rezultat al unei comenzi precedente) în care desenează imaginea. În acest exemplu se utilizează valori implicite ale unor facilități cum ar fi domeniul pentru axele x și y , spațiile dintre diviziunile de pe axe, culoarea și tipul liniei.

Mai general, în loc de `plot(x, y)`, putem utiliza `plot(x, y, șir)`, unde `șir` este un șir de caractere ce controlează culoarea, marcajul și stilul de linie. De exemplu, `plot(x, y, 'r*--')` ne spune că în fiecare punct $x(i)$, $y(i)$ se va plasa un asterisc roșu, punctele fiind unite cu o linie

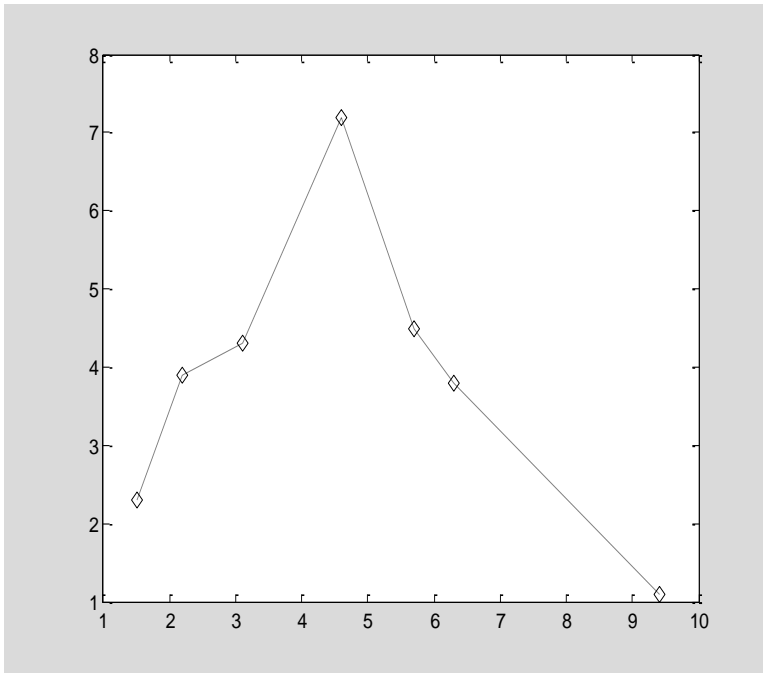
roșie întreruptă. Cele trei elemente din șir pot apare în orice ordine; de exemplu, `plot(x,y,'ms--')` și `plot(x,y,'s--m')` sunt echivalente.

`plot(x,y,'+y')` marchează punctele cu un plus galben, fără a le uni cu nici o linie. Tabela de mai jos dă toate opțiunile disponibile.

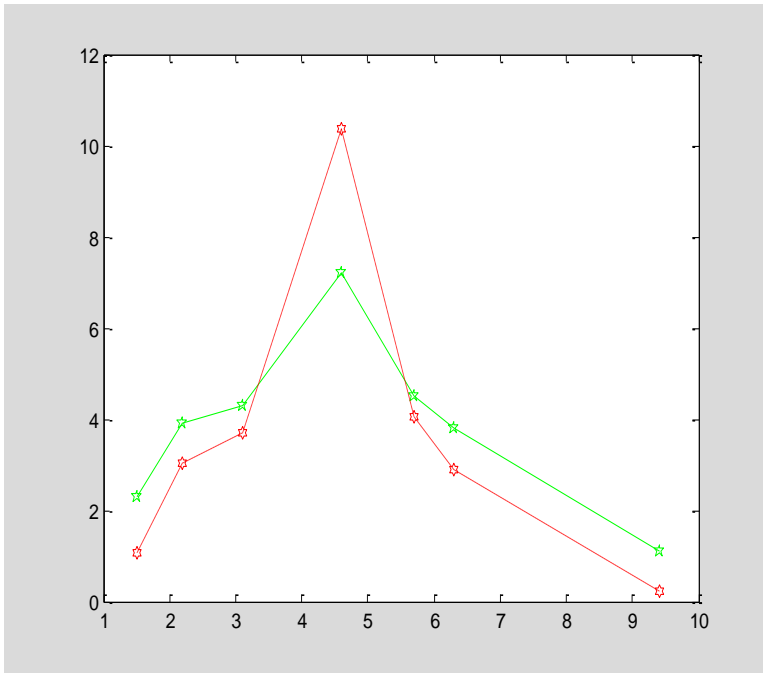
Culoare		Marcaj		Stil linie	
r	roșu	o	Cerc	-	continuă
g	verde	*	asterisc	--	întreruptă
b	albastru	.	punct	:	punctată
c	cyan	+	Plus	-.	linie-punct
m	magenta	x	Ori		
y	galben	s	pătrat		
k	negru	d	Romb		
w	alb	^	triunghi în sus		
		v	triunghi în jos		
		>	triunghi dreapta		
		<	triunghi stânga		
		p	pentagramă (stea cu 5 colțuri)		
		h	hexagramă (stea cu 6 colțuri)		

Exemplul de mai jos desenează o linie punctată neagră marcată cu romburi.

`plot(x,y,'kd:')`



De notat că plot acceptă mai multe seturi de date. De exemplu,
`b=x; c=y.^2/5;`
`plot(x,y,'g-p',b,c,'r--h')`

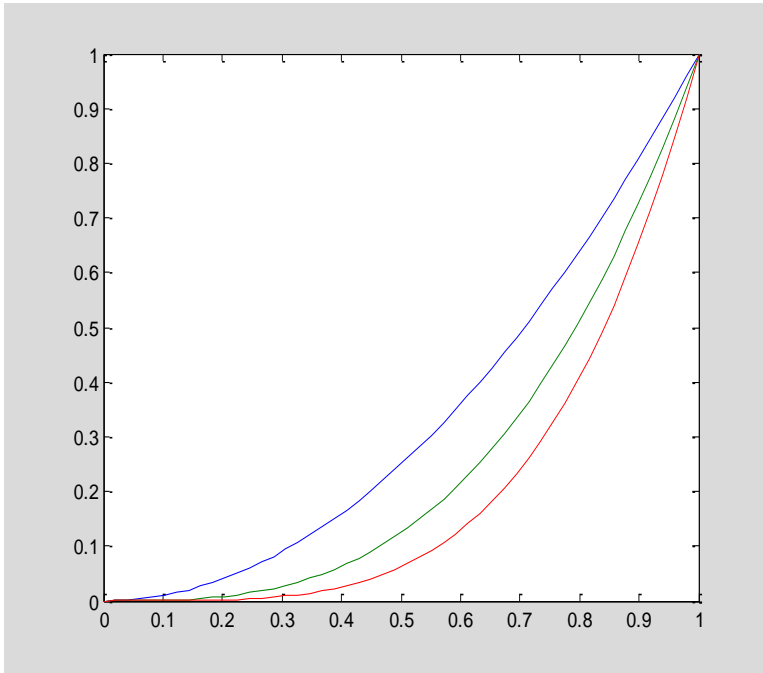


desenează în aceeași figură graficele pentru $x(i)$, $y(i)$ și $b(i)$, $c(i)$ cu linie continuă verde și respectiv cu linie întreruptă roșie, marcate cu pentagramă și respectiv hexagramă.

Comanda `plot` acceptă și argumente matriciale. Dacă x este un vector de dimensiune m și Y este o matrice $m \times n$, `plot(x, Y)` suprapune graficele obținute din x și fiecare coloană a lui Y . Similar, dacă X și Y sunt matrice de aceeași dimensiune, `plot(X, Y)` suprapune graficele obținute din coloanele corespunzătoare ale lui X și Y .

Exemplu: `plot` cu argument matrice

```
x=linspace(0,1,50)';  
plot(x,[x.^2,x.^3,x.^4])
```

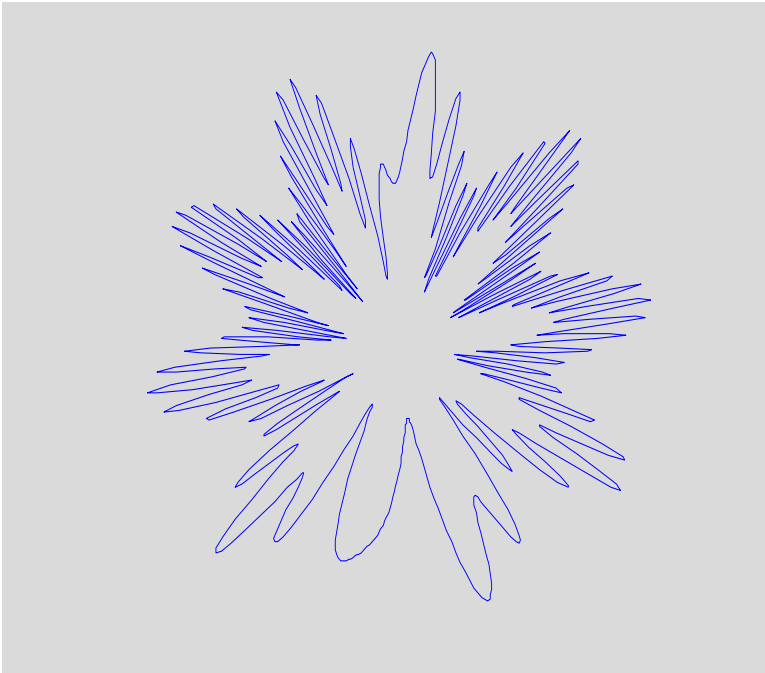


Dacă argumentele lui `plot` nu sunt reale, atunci părțile imaginare sunt în general ignorate. Singura excepție este atunci când `plot` este apelat cu un singur argument. Dacă Y este complex, `plot(Y)` este echivalent cu `plot(real(Y), imag(Y))`. În cazul când Y este real, `plot(Y)` desenează graficul obținut luând pe abscisă indicii punctelor și pe ordonată Y .

Exemplu de grafic în complex: dorim să reprezentăm funcția complexă de argument real $f: [-1,1] \rightarrow \mathbb{C}$,

$$f(x) = (3 + \sin 10\pi x + \sin 61e^{0.8 \sin \pi x + 0.7})e^{\pi i x}$$

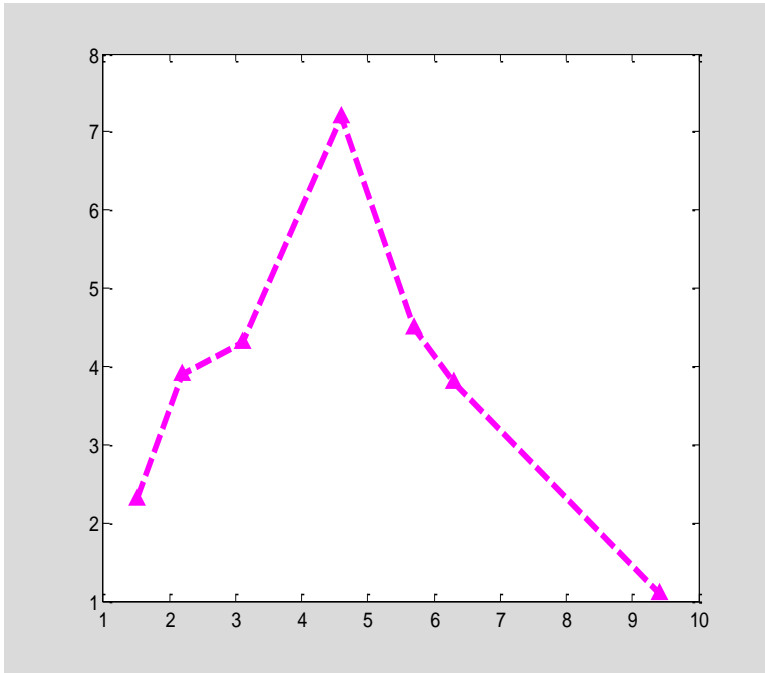
```
x=linspace(-1,1,650);  
f=(3+sin(10*pi*x)+sin(61*exp(0.8*sin(pi*x)+0.7))).*exp(pi*i*x);  
plot(f); axis equal; axis off
```



Atributele se pot controla furnizând argumente suplimentare lui plot. Proprietățile `Linewidth` (implicit 0.5 puncte) și `MarkerSize` (implicit 6 puncte) pot fi specificate în puncte, unde un punct este 1/72 inch.

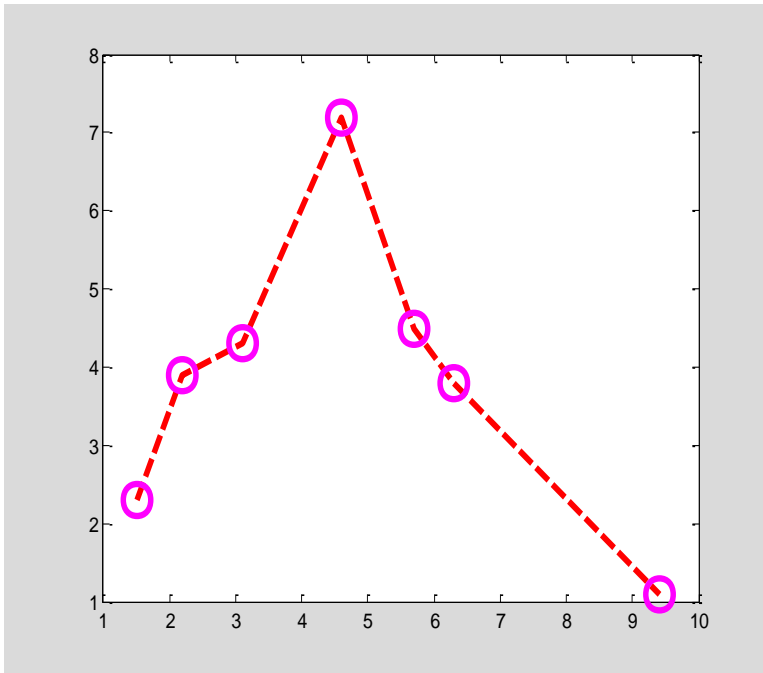
```
x=[1.5, 2.2, 3.1, 4.6, 5.7, 6.3, 9.4];
```

```
y=[2.3, 3.9, 4.3, 7.2, 4.5, 3.8, 1.1];  
plot(x,y,'m--^','LineWidth',3,'MarkerSize',5)
```



Culoarea laturilor marcajului și a interiorului marcajului se poate seta pe una din culorile din tabela 2.1 cu proprietățile `MarkerEdgeColor` și `MarkerFaceColor`.

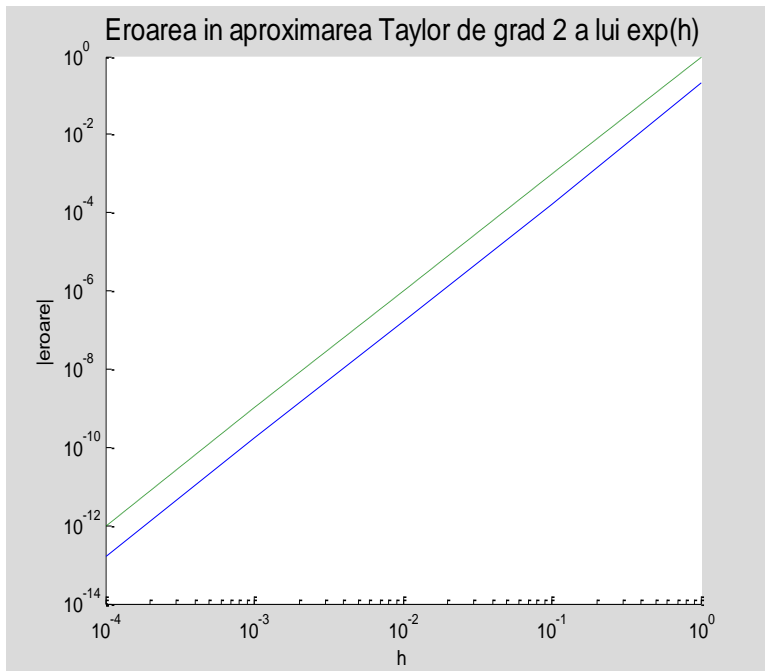
```
plot(x,y,'r--  
o','MarkerEdgeColor','m','LineWidth',3,'MarkerSize',15)
```

Funcția `loglog`, spre deosebire de `plot`, scalează axele logaritmice. Această facilitate este utilă pentru a reprezenta relații de tip putere sub forma unei drepte. În continuare vom reprezenta graficul restului Taylor de ordinul al doilea $|1 + h + \frac{h^2}{2} - e^h|$ pentru $h = 10^{-i}$, $i=0,1,\dots,4$. Când h este mic, această cantitate se comportă ca un multiplu al lui h^3 și deci pe o scară log-log valorile vor fi situate pe o dreaptă cu panta 3. Vom verifica aceasta reprezentând restul și dreapta de referință cu panta prevăzută cu linie punctată.

```
h=10.^[0:-1:-4];
taylerr=abs( (1+h+h.^2/2)-exp(h) );
loglog(h,taylerr,'-',h,h.^3,'-.'
```

```
xlabel('h'), ylabel('|eroare|')  
title('Eroarea in aproximarea Taylor de grad 2 a lui  
exp(h)', 'FontSize', 14)  
box off
```



În acest exemplu s-au utilizat comenzile `title`, `xlabel` și `ylabel`. Aceste funcții afișează șirul parametru de intrare deasupra imaginii, axei x și respectiv axei y . Comanda `box off` elimină caseta de pe marginea graficului curent, lăsând doar axele de coordonate. Dacă `loglog` primește și valori

nepozitive, MATLAB va da un avertisment și va afișa doar datele pozitive. Funcțiile înrudite `semilogx` și `semilogy`, scalează doar una din axe.

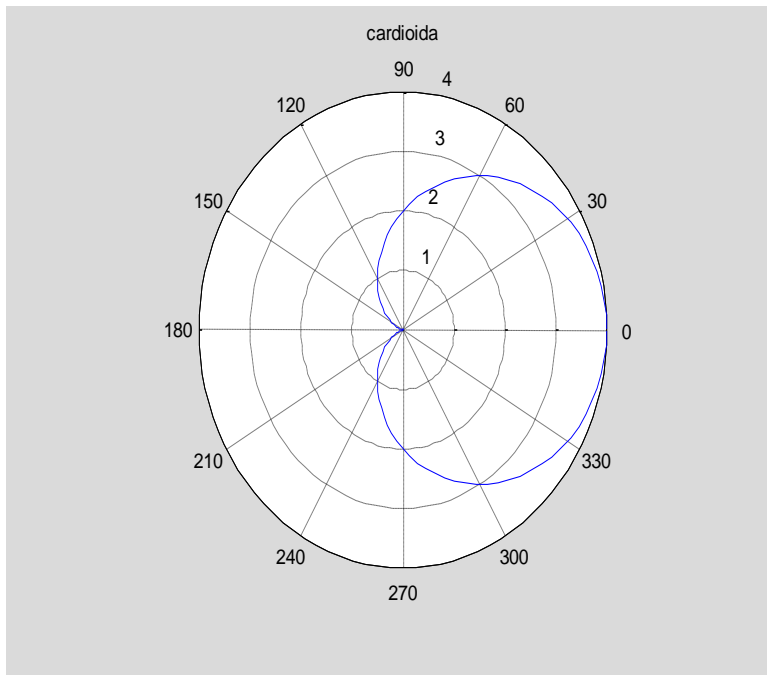
Dacă o comandă de afișare este urmată de alta, atunci noua imagine o va înlocui pe cea veche sau se va suprapune peste ea, depinzând de starea `hold` curentă. Comanda `hold on` face ca toate imaginile care urmează să se suprapună peste cea curentă, în timp ce `hold off` ne spune că fiecare imagine nouă o va înlocui pe cea precedentă. Starea implicită corespunde lui `hold off`.

Se pot reprezenta curbe în coordonate polare cu ajutorul comenzii `polar(t, r)`, unde t este unghiul polar, iar r este raza polară. Se poate folosi și un parametru suplimentar s , cu aceeași semnificație ca la `plot`. Graficul unei curbe în coordonate polare, numită cardioidă, și care are ecuația

$$r = a(1 + \cos t), \quad t \in [0, 2\pi]$$

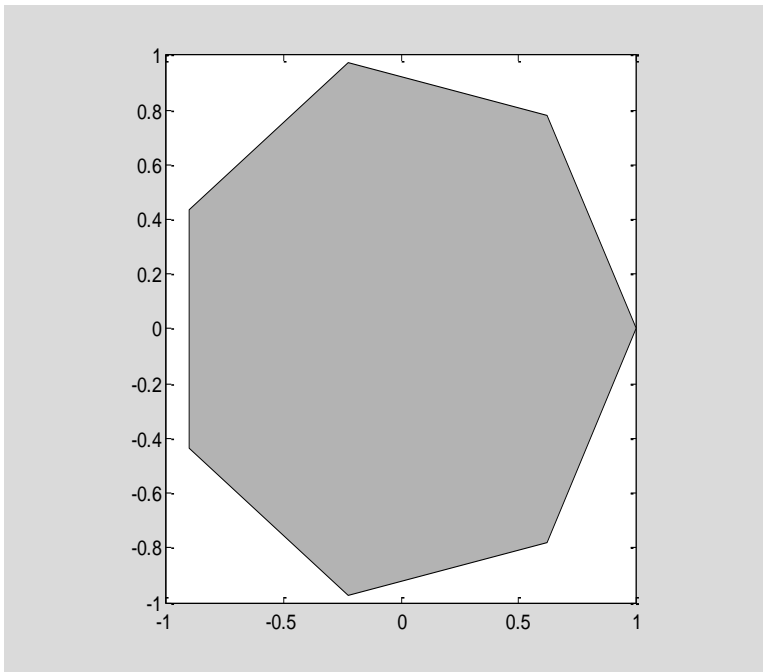
unde a este o constantă reală dată, se obține cu secvența:

```
%cardioida  
t=0:pi/50:2*pi;a=2;  
r=a*(1+cos(t));  
polar(t,r)  
title('cardioida')
```



Funcția `fill` lucrează la fel ca `plot`. Comanda `fill(x, y, c)` reprezintă poligonul cu vârfurile $x(i)$, $y(i)$ în culoarea c . Punctele se iau în ordine și ultimul se unește cu primul. Culoarea c se poate da și sub forma unui triplet RGB, $[r \ g \ b]$. Elementele r , g și b , care trebuie să fie scalari din $[0,1]$, determină nivelul de roșu, verde și albastru din culoare. Astfel, `fill(x, y, [0 1 0])` umple poligonul cu culoarea verde, iar `fill(x, y, [1 0 1])` cu magenta. Dând proporții egale de roșu, verde și albastru se obțin nuanțe de gri care variază de la negru ($[0 \ 0 \ 0]$) la alb ($[1 \ 1 \ 1]$). Exemplul următor desenează un heptagon regulat în gri:

```
n=7;  
t=2*(0:n-1)*pi/n;  
fill(cos(t),sin(t),[0.7,0.7,0.7])  
axis square
```



Comanda `clf` șterge figura curentă, iar `close` o închide. Este posibil să avem mai multe ferestre figuri pe ecran. Cel mai simplu mod de a crea o nouă figură este comanda `figure`. A n-a fereastră figură (unde n apare în bara de titlu) poate fi făcută figură curentă cu comanda `figure(n)`.

Comanda `close all` va închide toate ferestrele figuri. De notat că multe atribute ale unei figuri pot fi modificate interactiv, după afișarea figurii, utilizând meniul Tool al ferestrei sau bara de instrumente (toolbar). În particular, este posibil să se facă zoom pe o regiune particulară cu ajutorul mouse-ului (vezi `help zoom`).

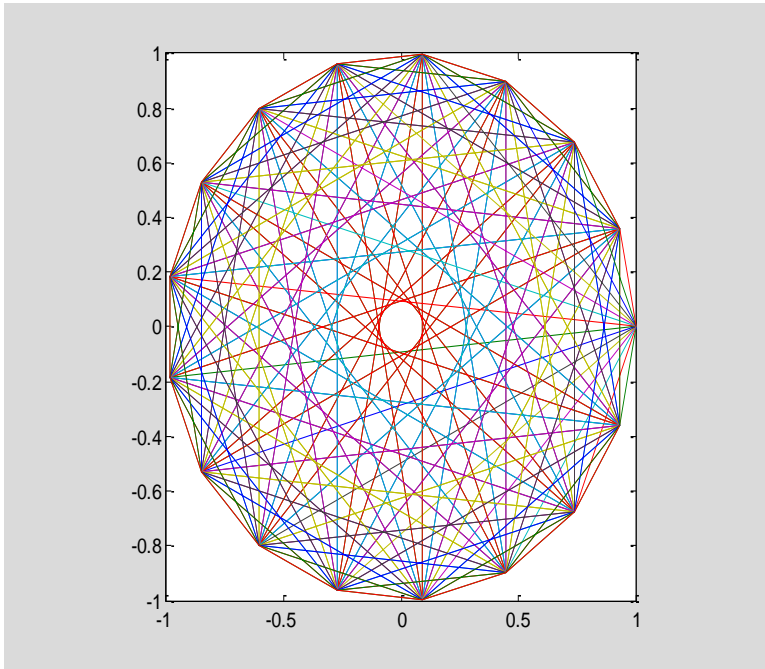
2. Axe și adnotarea

Diversele aspecte ale unui grafic pot fi controlate cu comanda `axis`. Unele opțiuni se dau în tabela de mai jos.

<code>axis([xmin xmax ymin ymax])</code>	Setează limitele axelor x și y
<code>axis auto</code>	Returnează limitele implicite
<code>axis equal</code>	Egalează unitățile pe axele de coordonate
<code>axis off</code>	Elimină axele
<code>axis square</code>	Face caseta axelor pătrată (cubică)
<code>axis tight</code>	Setează limitele axelor egale cu limitele datelor
<code>xlim([xmin xmax])</code>	Setează limitele pe axa x
<code>ylim([ymin, ymax])</code>	Setează limitele pe axa y

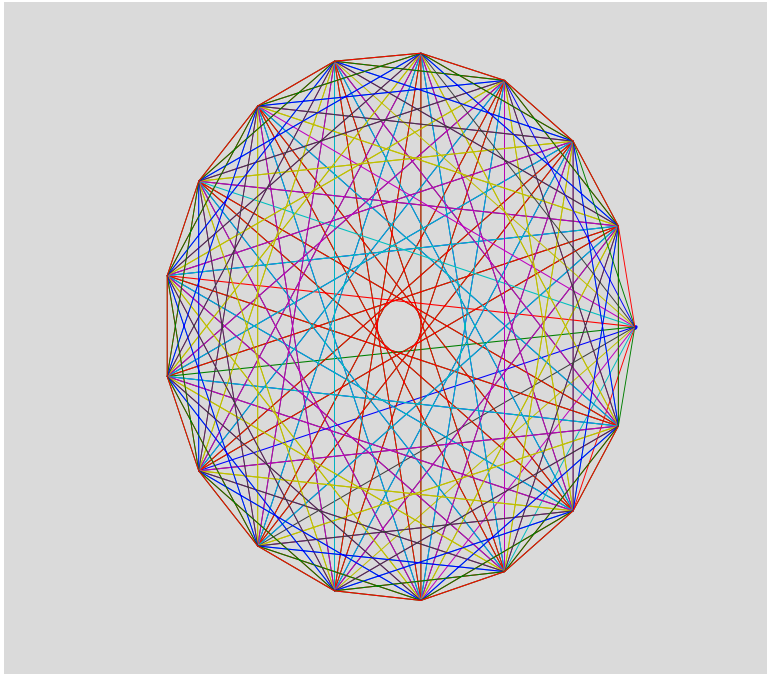
Axele pot fi eliminate cu `axis off`. Raportul dintre unitatea pe x și cea pe y (aspect ratio) poate fi făcut egal cu unu, astfel ca cercurile să nu pară elipse, cu `axis equal`. Comanda `axis square` face caseta axelor pătrată.

```
plot(fft(eye(17))), axis equal, axis square
```



Deoarece figura este situată în interiorul cercului unitate, axele sunt foarte necesare.

```
plot(fft(eye(17))), axis equal, axis off
```



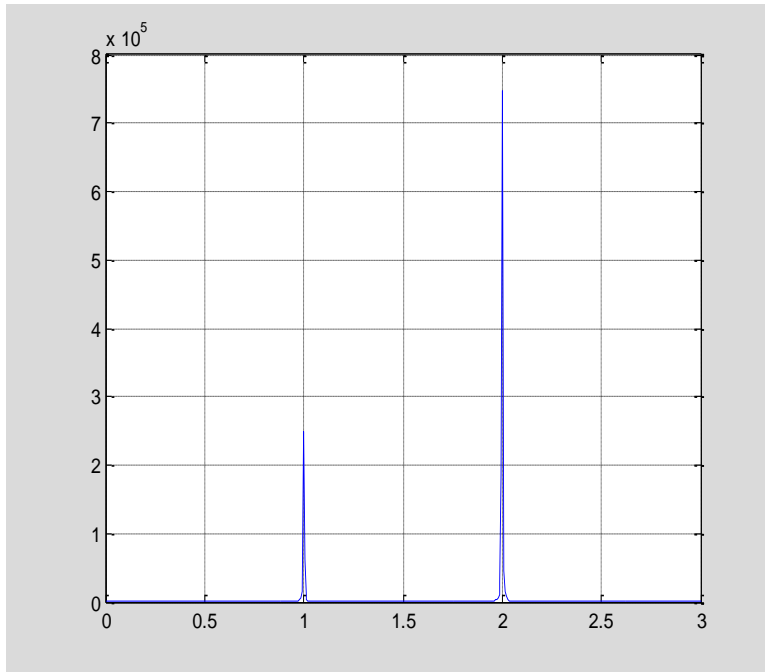
Comanda `axis([xmin xmax ymin ymax])` setează limitele pentru axa x și respectiv y . Pentru a reveni la setările implicite, pe care MATLAB le alege automat în funcție de datele care urmează a fi reprezentate, se utilizează `axis auto`. Dacă se dorește ca una dintre limite să fie aleasă automat de către MATLAB, ea se ia `-inf` sau `inf`; de exemplu, `axis([-1, 1, -inf, 0])`. Limitele pe axa x sau y se pot seta individual cu `xlim([xmin xmax])` și `ylim([ymin, ymax])`.

Exemplul următor reprezintă funcția $\frac{1}{(x-1)^2} + \frac{3}{(x-2)^2}$ pe intervalul $[0,3]$:

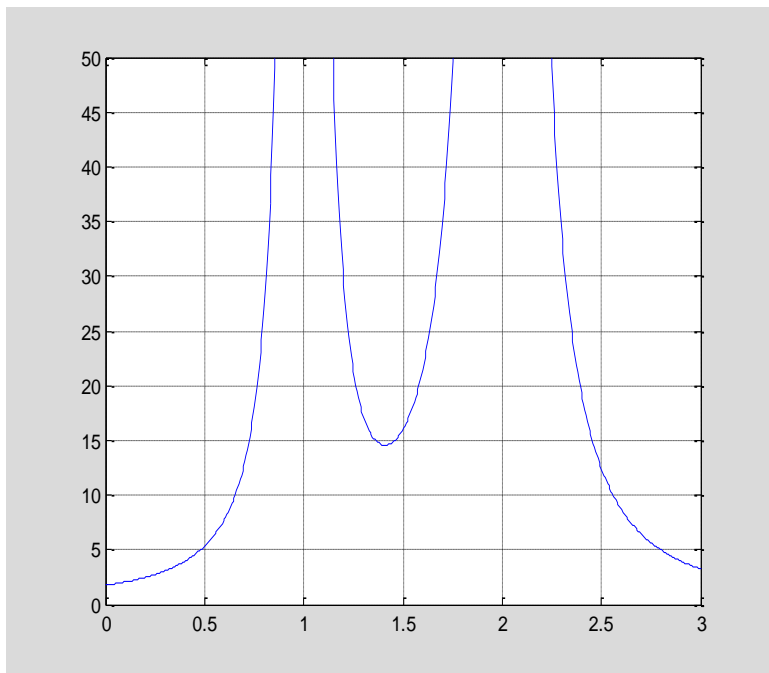
```
x=linspace(0,3,500);
```



```
plot(x,1./ (x-1) .^2+3./ (x-2) ^2;  
grid on
```



Comanda `grid on` produce o grilă de linii orizontale și verticale care pornesc de la diviziunile axelor. Rezultatul se poate vedea în figura de mai sus. Datorită singularităților din $x = 1; 2$ graficul nu dă prea multă informație. Totuși, executând comanda `ylim([0,50])`



se observă părțile interesante ale graficului.

Exemplul următor reprezintă epiciclopedia

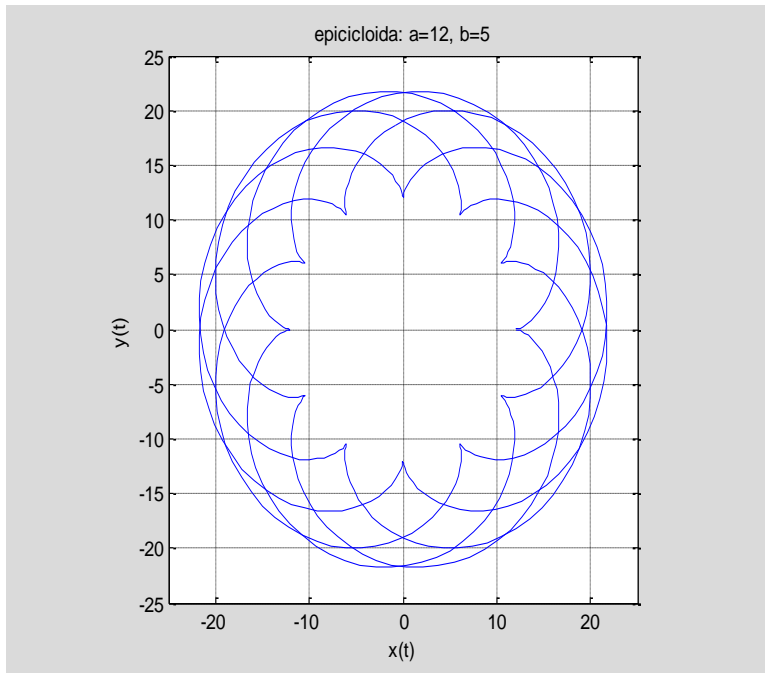
$$\left. \begin{aligned} x(t) &= (a + b) \cos t - b \cos \left(\frac{a}{b} + 1 \right) t \\ y(t) &= (a + b) \sin t - b \sin \left(\frac{a}{b} + 1 \right) t \end{aligned} \right\} 0 \leq t \leq 10\pi$$

pentru $a = 12$ și $b = 5$.

a = 12; b=5;

t=0:0.05:10*pi;

```
x = (a+b)*cos(t)-b*cos((a/b+1)*t);  
y = (a+b)*sin(t)-b*sin((a/b+1)*t);  
plot(x,y)  
axis equal  
axis([-25 25 -25 25])  
grid on  
title('epicicloida: a=12, b=5')  
xlabel('x(t)'), ylabel('y(t)')
```

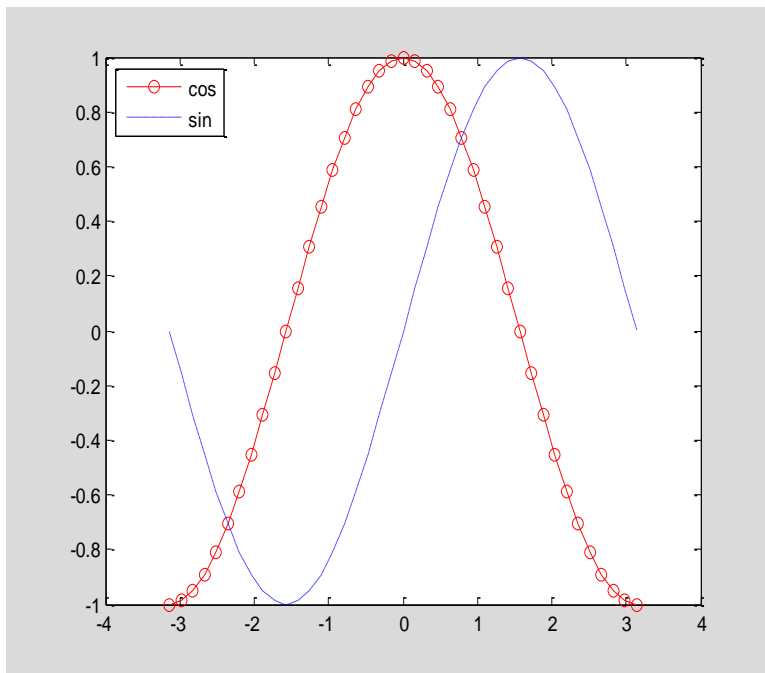


Limitele din `axis` au fost alese astfel ca să rămână un oarecare spațiu în jurul epicycloidei.

Comanda `legend('string1','string2',...,'stringn',pp)` va atașa unui grafic o legendă care pune 'stringi' după informația culoare/ marcaj/stil pentru graficul corespunzător.

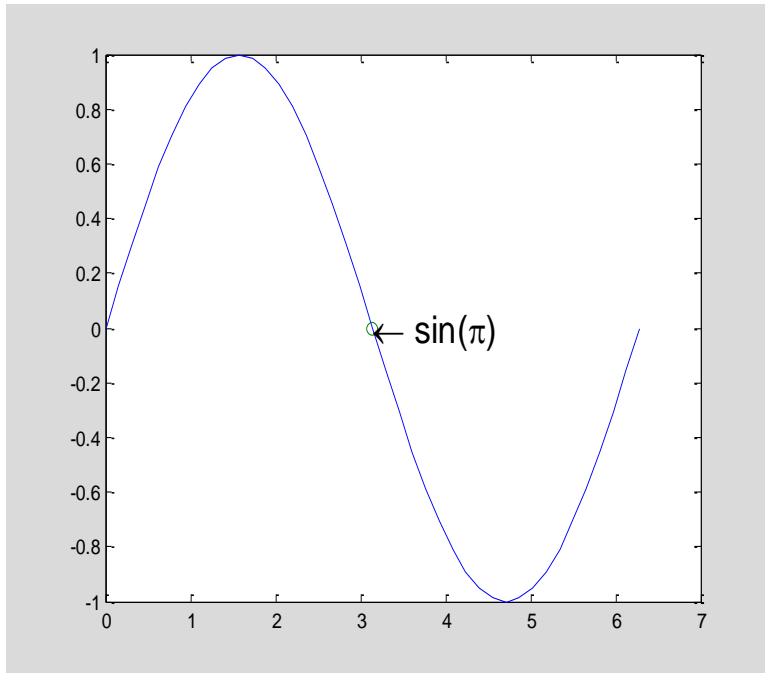
Parametrul opțional `pp` indică poziția legendei (vezi `help legend`). Exemplul care urmează adaugă o legendă unui grafic al sinusului și cosinusului

```
x = -pi:pi/20:pi;  
plot(x,cos(x),'-ro',x,sin(x),'-.b')  
h = legend('cos','sin',2);
```



Textele se pot include în grafice cu ajutorul comenzii `text(x, y, s)`, unde x și y sunt coordonatele textului, iar s este un șir de caractere sau o variabilă de tip șir. Începând cu versiunea 5, MATLAB permite introducerea în interiorul parametrului s a unor construcții TeX, de exemplu `_` pentru indice, `^` pentru exponent, sau litere grecești (`\alpha`, `\beta`, `\gamma`, etc.). De asemenea, anumite attribute ale textului, cum ar fi tipul font-ului, dimensiunea și altele sunt selectabile începând cu versiunea 4. Comenzile

```
plot(0:pi/20:2*pi, sin(0:pi/20:2*pi), pi, 0, 'o')  
text(pi, 0, '\leftarrow sin(\pi)', 'FontSize', 18)
```



adnotează punctul de coordonate $(\pi; 0)$ cu șirul $\sin(\pi)$. Aceste facilități se pot utiliza și în titluri, legende sau etichete ale axelor, care sunt obiecte de tip text. Începând cu MATLAB 7 primitivele text suportă un subset puternic LATEX. Proprietatea corespunzătoare se numește `Interpreter` și poate avea valorile `TeX`, `LaTeX` sau `none`. Pentru un exemplu de utilizare a macrourilor `LaTeX` a se vedea script-ul `graphLegendre.m`

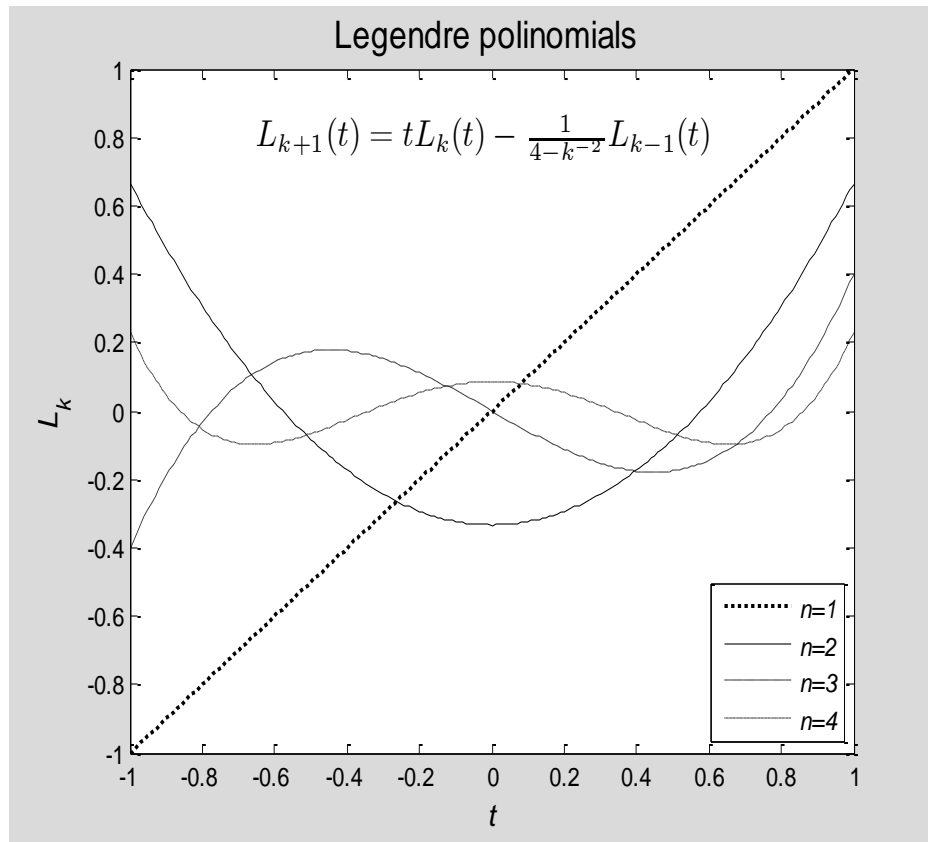
```
%graphs for Legendre polynomials  
n=4; clf
```

```

t=(-1:0.01:1)';
s=[];
ls={':', '-', '--', '-.'};
lw=[1.5, 0.5, 0.5, 0.5];
for k=1:n
    y=vLegendre(t, k);
    s=[s; strcat('\itn=', int2str(k))];
    plot(t, y, 'LineStyle', ls{k}, 'Linewidth', lw(k), 'Color', 'k');
    hold on
end
legend(s, 4)
xlabel('t', 'FontSize', 12, 'FontAngle', 'italic')
ylabel('L_k', 'FontSize', 12, 'FontAngle', 'italic')
title('Legendre polynomials ', 'FontSize', 14);
text(-0.65, 0.8, '$L_{k+1}(t)=tL_k(t)-\frac{1}{4-k^{-2}}L_{k-1}(t)$', ...
    'FontSize', 14, 'FontAngle', 'italic', 'Interpreter', 'LaTeX')

```

graphLegendre



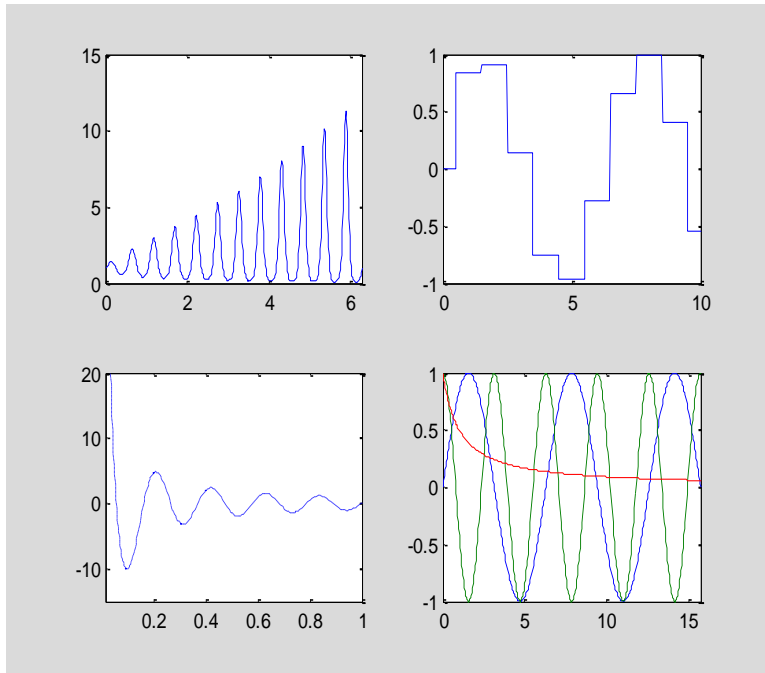
3. Mai multe grafice pe aceeași figură

Funcția `subplot` permite plasarea mai multor imagini pe o grilă în aceeași figură.

Format: `subplot(mnp)`, sau echivalent, `subplot(m,n,p)`.

Efect: fereastra figurii se împarte într-un tablou $m \times n$ de regiuni, fiecare având propriile ei axe. Comanda de desenare curentă se va aplica celei de-a p -a dintre aceste regiuni, unde contorul variază de-a lungul primei linii, apoi de-a lungul celei de-a doua ș.a.m.d. De exemplu, `subplot(425)` împarte fereastra figurii într-o matrice 4×2 de regiuni și ne spune că toate comenzile de desenare se vor aplica celei de-a cincea regiuni, adică primei regiuni din al treilea rând. Dacă se execută mai târziu `subplot(427)`, atunci poziția (4,1) devine activă. Vom da în continuare mai multe exemple.

```
subplot(221), fplot('exp(sqrt(x)*sin(12*x))',[0 2*pi])
subplot(222), fplot('sin(round(x))',[0,10],'-')
subplot(223), fplot('cos(30*x)/x',[0.01 1 -15 20],'-.')
subplot(224)
fplot('[sin(x),cos(2*x),1/(1+x)]',[0 5*pi -1.5 -1.5 1.5])
```



Sintaxa generală a lui `fplot` este

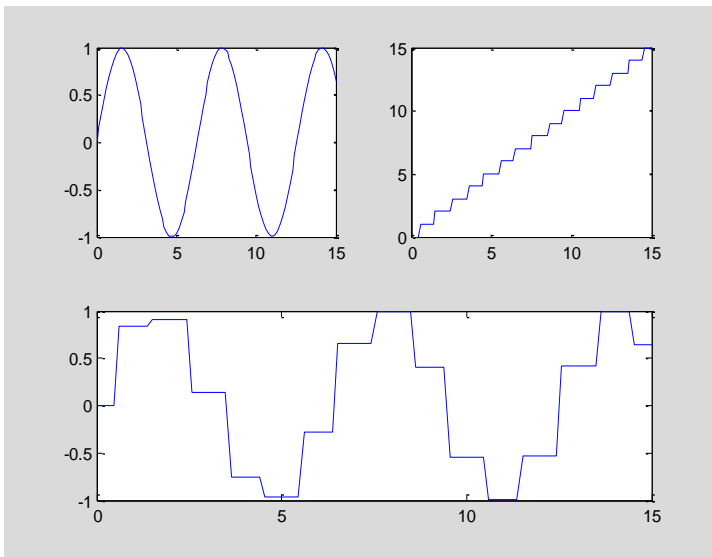
`fplot(fun,lims,tol,N,'LineStyle',p1,p2,...)`

- `fun` este funcția de reprezentat (function handle, obiect inline sau expresie);
- `lims` dă limitele pe axele x și/sau y ;
- `tol` este eroarea relativă (implicit 2×10^{-3});
- se folosesc cel puțin `N+1` puncte la generarea graficului;

- `LineStyle` determină tipul de linie;
- `p1, p2, ...` sunt parametrii adiționali transmiși lui `fun`, care trebuie să aibă parametrii de intrare `x, p1, p2, ...`

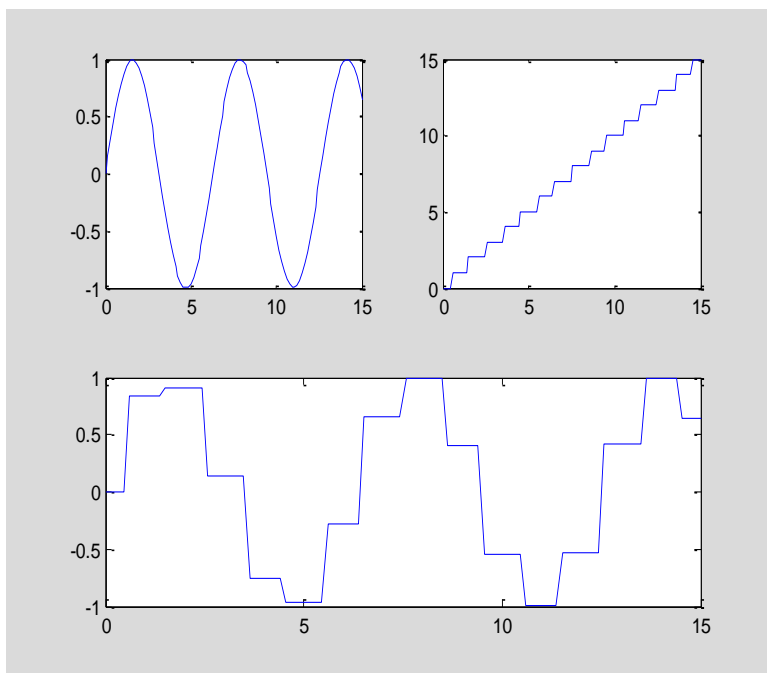
Este posibil să se obțină grile neregulate de imagini apelând `subplot` cu șabloane de grile diferite. Exemplu:

```
x = linspace(0,15,100);  
subplot(2,2,1), plot(x,sin(x))  
subplot(2,2,2), plot(x,round(x))  
subplot(2,1,2), plot(x,sin(round(x)))
```



Al treilea argument al lui `subplot` poate fi un vector ce specifică mai multe regiuni; ultima linie se poate înlocui cu

```
subplot(2,2,3:4) , plot(x,sin(round(x)))
```



Dăm o tabelă rezumativă a funcțiilor grafice bidimensionale

plot	grafic x - y simplu
loglog	grafic cu scară logaritmică pe ambele axe
semilogx	grafic cu scară logaritmică pe axa x
semilogy	grafic cu scară logaritmică pe axa y
plotyy	grafic x - y cu axe y și la stânga și la dreapta
polar	grafic polar
fplot	reprezentare grafică automată a unei funcții
ezplot	versiune ușor de utilizat (easy-to-use) a lui plot
ezpolar	versiune ușor de utilizat (easy-to-use) a lui polar
fill	umplere poligon
area	grafic de tip arie plină
bar	grafic de tip bară
barh	grafic de tip bară orizontală
hist	histogramă
pie	grafic cu sectoare de cerc
comet	grafic x - y animat
errorbar	grafic cu bare de eroare
quiver	câmp de vectori bidimensional
scatter	Grafic dispersat (nor de puncte)