

Tab. 6.5. Reprezentari grafice 3D elementare și speciale		
Categoria	Funcția și sintaxa de bază	Detalii de utilizare
Reprezentarea punctelor, liniilor și a curbelor în spațiu	<code>plot3(x,y,z)</code>	x, y, z - vectorii coordonatelor (au aceeași dimensiune)
	<code>plot3(x,y,z,s)</code>	s - variabilă șir cu maxim 3 caractere, pentru opțiuni
	<code>plot3(x1,y1,z1,s1,x2,y2,z2,s2)</code>	Trasează traiectorii multiple în același sistem de axe.
	<code>comet3(z)</code> <code>comet3(x,y,z)</code>	Reprezentare 3D (traiectorie) dinamică ; x, y, z - vectori
	<code>comet3(x,y,z,p)</code>	Lungimea (urma) cometei este dată de $p \cdot \text{length}(z)$, unde implicit $p = 0, 1$
Reprezentarea suprafețelor	<code>[Xi,Yi]=meshgrid(x,y)</code>	X_i, Y_i – matrici ce definesc punctele domeniului de evaluare a suprafeței
	<code>mesh(Z)</code> <code>mesh(Xi,Yi,Z)</code>	Z - matricea punctelor pe care se construiește plasa (mesh)
	<code>meshc(Z)</code>	Combinație mesh cu contour în planul de proiecție de dedesubt
	<code>meshz(Z)</code>	Combinație mesh cu cortină pe planul de proiecție de dedesubt
	<code>surf(Z)</code>	Construiește suprafețe pline pe punctele matricei Z
	<code>surfc(Z)</code>	Combinație surf cu contour în planul de proiecție de dedesubt
	<code>waterfall(Z)</code>	Reprezentare dinamică mesh
Reprezentarea fețelor poligonale ale obiectelor 3D	<code>fill3(x,y,z,c)</code> <code>patch(X,Y,Z,c)</code>	Crează suprafețe poligonale în 3D, respectiv pete de culoare c definind coordonatele vârfurilor prin vectorii de aceeași lungime x, y, z .

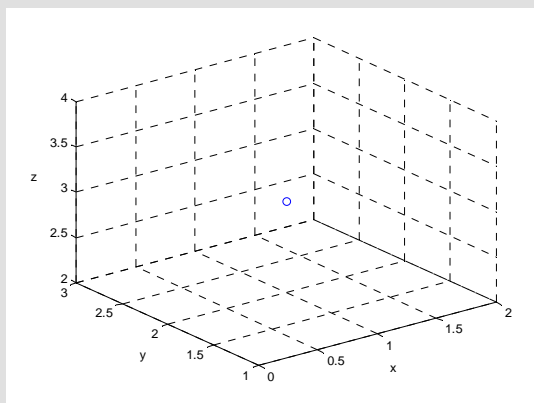
Tab. 6.5. Reprezentari grafice 3D elementare și speciale (continuare)

Reprezentare de obiecte spațiale	<code>cylinder(R,n)</code> sau secvența <code>[x,y,z]=cylinder(R,n)</code> <code>surface(x,y,z)</code>	Obiectul cilindru: $R=[R1 \ R2]$ vector cu razele bazelor, n –numărul punctelor de discretizare a circumferinței directe
	<code>sphere(n)</code> sau secvența <code>[x,y,z]=sphere(n)</code> <code>surf(x,y,z)</code>	Obiectul sferă construit din $n \times n$ suprafețe elementare; x, y, z –matrice egale ca dimensiune.
	<code>mesh(x,y,z)</code>	Suprafață sferică de tip plasă
	<code>ellipsoid(xc,yc,zc,xr,yr,zr,n)</code> sau secvența <code>[x,y,z]=ellipsoid(xc,yc,zc,xr,yr,zr,n)</code> <code>surf(x,y,z)</code>	Generează matricele x, y și z cu dimensiuni $(n+1) \times (n+1)$ ca bază de reprezentare pentru un elipsoid de centru (xc,yc,zc) și axe (xr,yr,zr) date, formată din $n \times n$ suprafețe elementare.

Exemple de utilizare a funcției `plot3`

(1) Reprezentarea punctului de coordonate (1,2,3) cu simbolul cerculeț.

```
>>plot3(1,2,3,'o')
>> grid
```



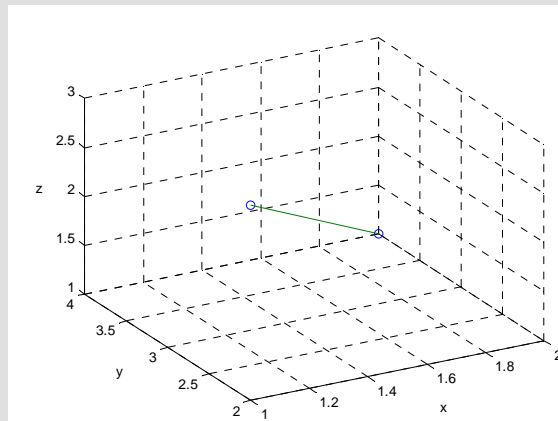
(2) Reprezentarea unui segment de dreapta în spațiu delimitat de punctele de coordonate P1(1,2,3) și P2(2,4,1) cu marcarea acestora cu simbol cerculeț.

```
>> x=[1 2]
x =
     1     2

>> y=[2 4]
y =
     2     4

>> z=[3 1]
z =
     3     1

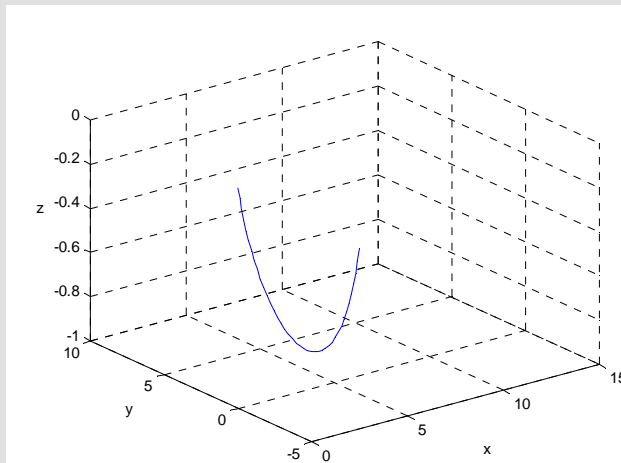
>> plot3(x,y,z,'o',x,y,z)
>> grid
```



(3) Reprezentarea unei curbe în spațiu definită parametric prin expresiile :

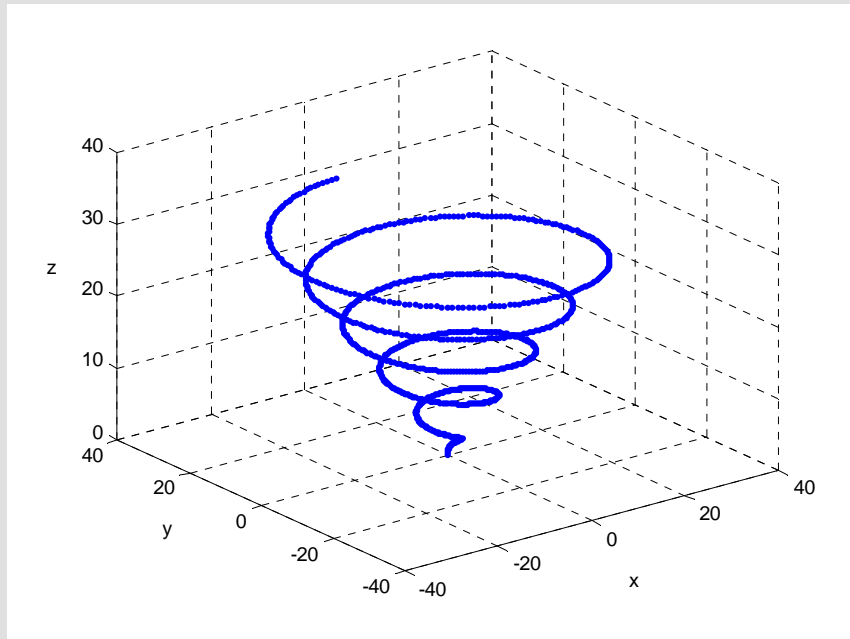
$$\begin{cases} x = 3t^2 + t \\ y = t^3 + t^2 - t \\ z = -\sin(t) \end{cases}, \text{ pentru valori ale parametrului } t \in [0,2], \text{ cu pasul de } 0,5.$$

```
clear
t=0:0.05:2
x=3*t.^2+t
y=t.^3+t.^2-t
z=-sin(t)
plot3(x,y,z)
end
```



(4) Reprezentarea unei spirale conice

```
t=0:pi/100:5*pi
plot3(sin(t).*t,cos(t).*t,t,'.')
grid
```



Reprezentarea suprafețelor

Ca și curbele, suprafețele sunt obiecte grafice generate punct cu punct în conformitate cu o anumită expresie analitică descrisă matematic sub cele trei forme: *explicit*, *implicit* și *parametric*. Astfel, teoretic, orice suprafață este constituită dintr-o infinitate de puncte (de coordonate x, y, z), care îndeplinesc (verifică) funcția *explicită* de două variabile: $z = f(x, y)$, respectiv varianta *implicită* $F(x, y, z) = 0$. În descrierea *parametrică*, un punct situat pe o suprafață este reprezentat prin vectorul $\bar{R}_s[x(u, v) \ y(u, v) \ z(u, v)]$, unde $u \in [0, 1]$ și $v \in [0, 1]$ sunt parametrii.

Reprezentarea grafică a suprafețelor prin puncte în 3D constituie o problemă de interpolare pe un număr finit de puncte. Așa cum se cunoaște, interpolarea datelor este un caz special al problemei generale de aproximare a funcțiilor. Astfel, se pune problema găsirii unei forme analitice care să descrie cât mai precis o mulțime de puncte date în spațiu. Această

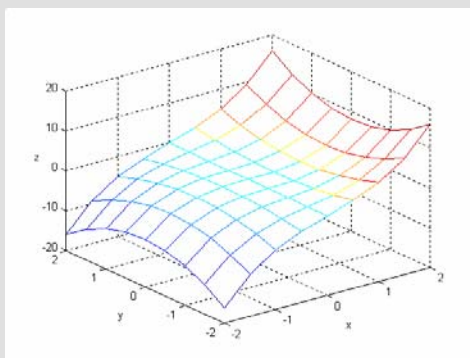
problematică este tratată în Cap. 5.7.5 în cadrul punctului B -*Interpolarea funcțiilor de două variabile* unde se exemplifică reprezentarea grafică a suprafețelor descrise explicit.

Funcțiile Matlab `mesh` și `surf` se folosesc pentru trasarea grafică a suprafețelor descrise explicit, prin expresii de forma $z = f(x, y)$. Pentru a putea face acest lucru variabilele (unidimensionale) x și y trebuie convertite în matrici bidimensionale. Aceasta operație se poate face cu funcția `meshgrid` apelată cu sintaxa `[X,Y]=meshgrid(x,y)`, care transformă datele specificate de vectorii x și y în tablourile X și Y necesare pentru evaluarea funcțiilor de două variabile și pentru reprezentări 3D de tipul `mesh` sau `surface` care admit argumente matrice.

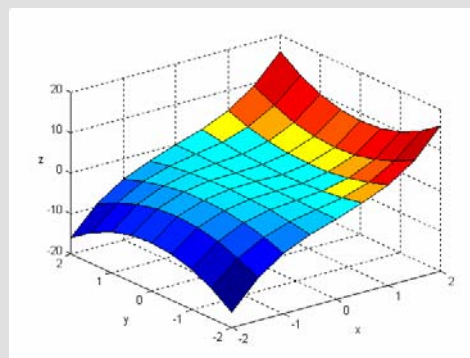
Exemplu. Să se evalueze funcția $z = x(x^2 + y^2)$ pe domeniul $[-2, 2] \times [-2, 2]$ și să se reprezinte grafic ca plasă de puncte (`mesh`) și ca suprafață (`surf`).

```
%program suprafata explicita
%definire a domeniului de reprezentare
x=-2:0.5:2
y=-2:0.5:2
%transpunere vector y
y=y'
%reteaua de puncte asociata domeniului
[X,Y]=meshgrid(x,y)
%ecuatia suprafetei
Z=X.*(X.^2+Y.^2)
%reprezentarea grafica
mesh(X,Y,Z)
```

mesh(X,Y,Z)



surf(X,Y,Z)



Există anumite funcții predefinite, care definesc suprafețe tipice cum este cea descrisă de funcția Matlab `peaks` sau suprafețe generate de curbe (polinoame cubice) consacrate în grafica pe calculator (Ferguson, Bezier, Coons).

Funcția `peaks`

Aceasta este o funcție specială de două variabile, care pune la dispoziție o formă analitică dată de expresia :

```
>> peaks
```

```
z = 3*(1-x).^2.*exp(-(x.^2)-(y+1).^2) ...
    - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...
    - 1/3*exp(-(x+1).^2 - y.^2)
```

Forma analitică de mai sus descrie o suprafață cu formă Gaussiană utilă în diferite demonstrații pentru funcții grafice cum sunt: `mesh`, `surf`, `contour`, `pcolor`, etc.

Se poate apela în mai multe moduri, conform sintaxelor următoare :

<code>z = peaks;</code>	<code>peaks;</code>	<code>[x,y,z] = peaks;</code>
<code>z = peaks(n);</code>	<code>peaks(n);</code>	<code>[x,y,z] = peaks(n);</code>
<code>z = peaks(v);</code>	<code>peaks(v);</code>	<code>[x,y,z] = peaks(v);</code>
<code>z = peaks(x,y);</code>	<code>peaks(x,y);</code>	

În primul grup :

- Prima variantă generează matrici de puncte cu dimensiunea implicită 49×49.
- A doua variantă generează matrici de puncte cu dimensiune specificată $n \times n$.
- Varianta a treia generează matrici de puncte cu dimensiune specificată $n \times n$, unde n este lungime unui vector dat v , adică $n = \text{length}(v)$.
- Varianta a patra evaluează funcția `peaks` în punctele tablourilor de date x și y , care trebuie să aibă aceeași dimensiune.

Variantele din al doilea grup, fără argumente de ieșire efectuează direct o trasare `surf`.

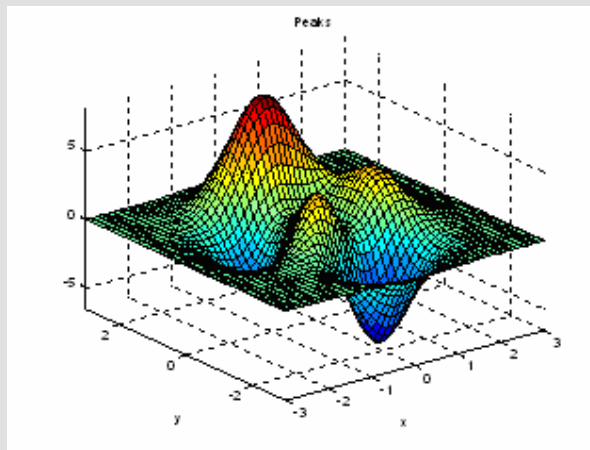
Ultimele trei variante produc matricele de puncte pentru reprezentări de tip `contour(x,y,z)` și `pcolor(x,y,z)`.

Notă. Dacă variabilele de intrare x și y nu sunt matrice atunci ele trebuie obținute ca atare cu funcția `meshgrid`.

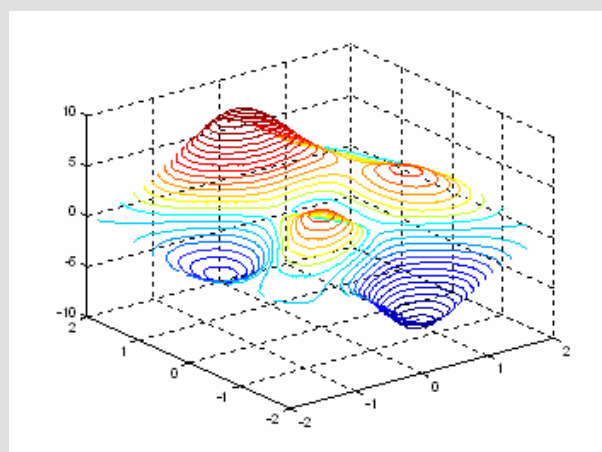
Exemplu.

```
%reprezentare peaks contour si pcolor
clear
x=-2:0.1:2;
y=x;
[X,Y]=meshgrid(x,y);
Z=peaks(X,Y);
figure(1)
peaks
figure(2)
contour3(X,Y,Z,30)
figure(3)
pcolor(X,Y,Z)
```

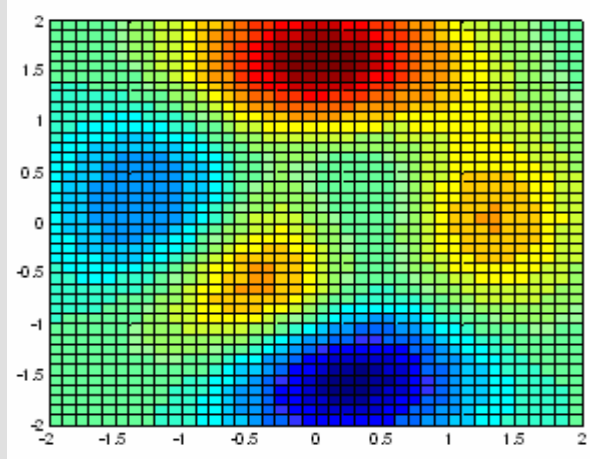
figure(1)



figure(2) %reprezentare contururi dispuse spatial



figure(3) %reprezentare cu pete de culoare (pcolor)



Suprafețe descrise parametric

Se va exemplifica reprezentarea grafică a unei suprafețe consacrate-cunoscută sub denumirea de *suprafața Ferguson* descrisă prin matricea "amprentă" generată din valori aleatoare, pentru trei rulări succesive.

Reamintim că, în descrierea *parametrică*, un punct situat pe o suprafață este reprezentat prin vectorul $\bar{R}_S[x(u, v) \ y(u, v) \ z(u, v)]$, unde $u \in [0, 1]$ și $v \in [0, 1]$ sunt parametrii.

Algoritmul de rezolvare se bazează pe ecuația matriceală vectorială [17]:

$$\bar{R}_S(u, v) = [1, u, u^2, u^3] \cdot \underbrace{\begin{bmatrix} \bar{a}_{00} & \bar{a}_{01} & \bar{a}_{02} & \bar{a}_{03} \\ \bar{a}_{10} & \bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\ \bar{a}_{20} & \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} \\ \bar{a}_{30} & \bar{a}_{31} & \bar{a}_{32} & \bar{a}_{33} \end{bmatrix}}_{\text{Matricea "amprentă" a suprafeței având elementele vectori tridimensionali}} \cdot \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

Ecuția matriceală vectorială echivalează cu trei ecuații parametrice ale suprafeței:

$$x(u,v) = \begin{bmatrix} 1, u, u^2, u^3 \end{bmatrix} \cdot \begin{bmatrix} a_{00}^x & a_{01}^x & a_{02}^x & a_{03}^x \\ a_{10}^x & a_{11}^x & a_{12}^x & a_{13}^x \\ a_{20}^x & a_{21}^x & a_{22}^x & a_{23}^x \\ a_{30}^x & a_{31}^x & a_{32}^x & a_{33}^x \end{bmatrix} \cdot \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

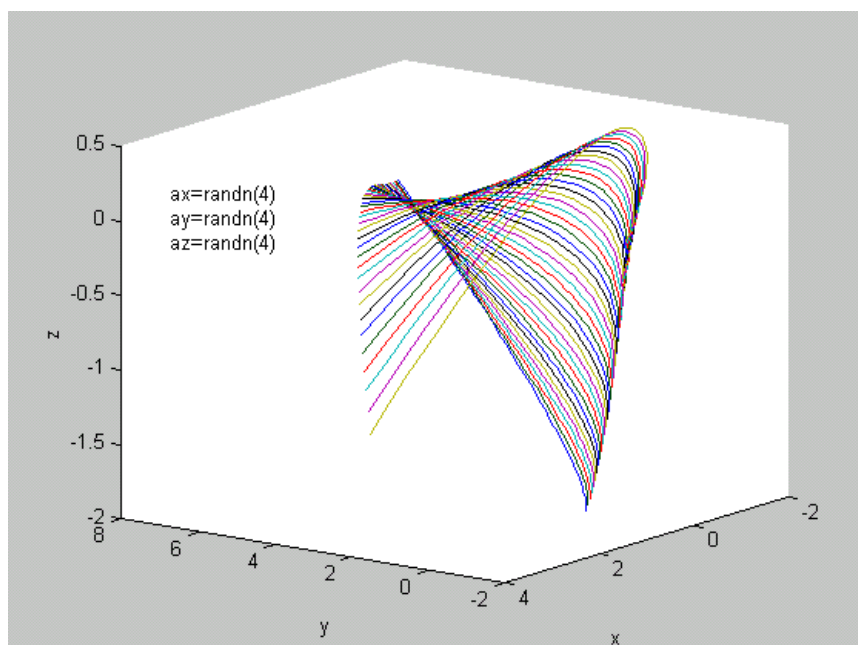
$$y(u,v) = \begin{bmatrix} 1, u, u^2, u^3 \end{bmatrix} \cdot \begin{bmatrix} a_{00}^y & a_{01}^y & a_{02}^y & a_{03}^y \\ a_{10}^y & a_{11}^y & a_{12}^y & a_{13}^y \\ a_{20}^y & a_{21}^y & a_{22}^y & a_{23}^y \\ a_{30}^y & a_{31}^y & a_{32}^y & a_{33}^y \end{bmatrix} \cdot \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

$$z(u,v) = \begin{bmatrix} 1, u, u^2, u^3 \end{bmatrix} \cdot \begin{bmatrix} a_{00}^z & a_{01}^z & a_{02}^z & a_{03}^z \\ a_{10}^z & a_{11}^z & a_{12}^z & a_{13}^z \\ a_{20}^z & a_{21}^z & a_{22}^z & a_{23}^z \\ a_{30}^z & a_{31}^z & a_{32}^z & a_{33}^z \end{bmatrix} \cdot \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

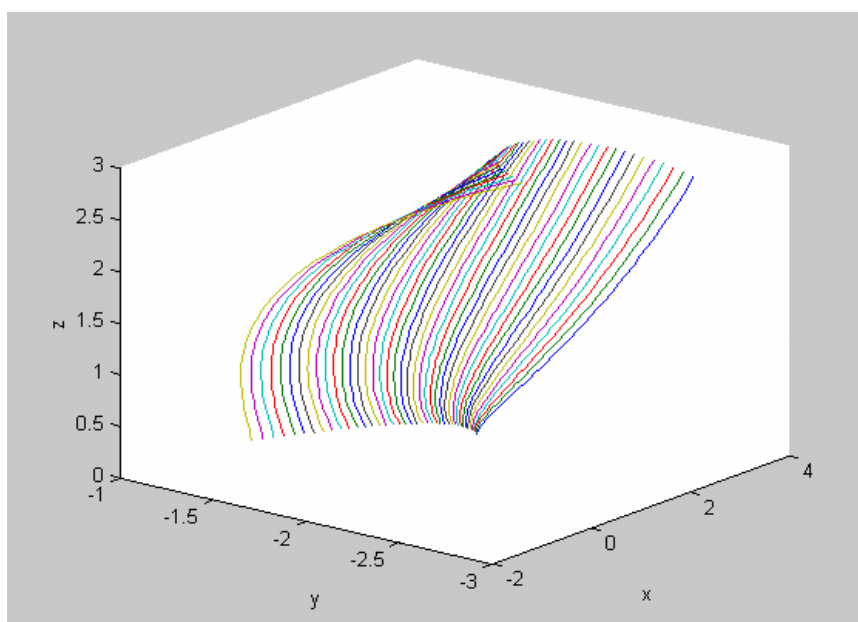
Programul care implementează modelul de mai sus este următorul:

```
%Program pentru generarea libera a suprafetelor bazate pe cubice
%Ferguson
%suprafata Ferguson
clear
%vectorii parametrilor
u=0:0.025:1
v=0:0.025:1
%matricea amprenta
ax=randn(4)
ay=randn(4)
az=randn(4)
%matricile parametrilor
for i=1:length(u)
    for j=1:length(v)
        up=[1 u(i) u(i).^2 u(i).^3];
        vp=[1 v(j) v(j).^2 v(j).^3];
        x(i,j)=up*ax*vp';
        y(i,j)=up*ay*vp';
        z(i,j)=up*az*vp';
    end
end
%trasare grafica
plot3(x,y,z)
xlabel('x')
ylabel('y')
zlabel('z')
end
```

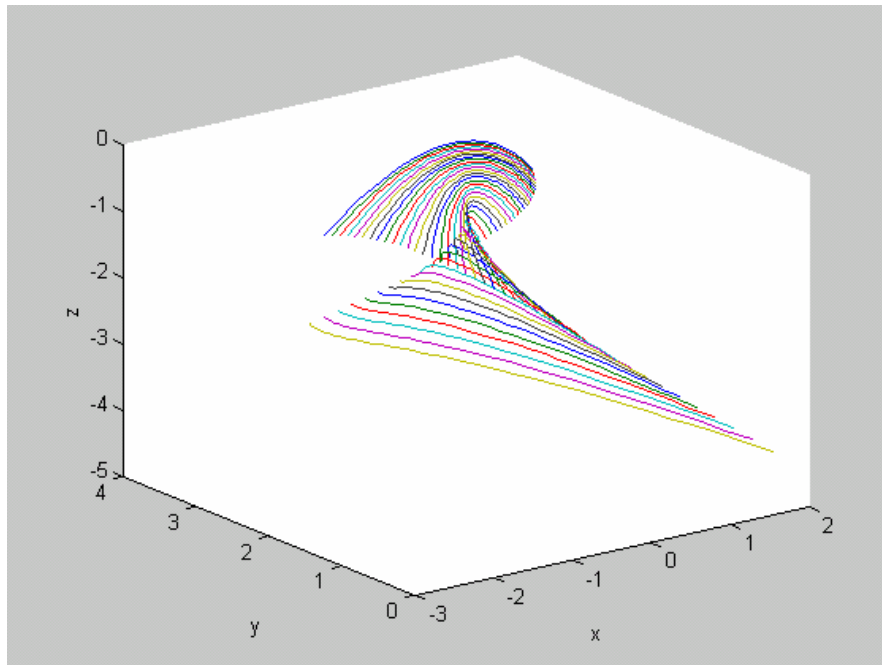
Suprafață Ferguson având ca amprentă o matrice aleatoare cu distribuție normală (rulare 1).



Suprafață Ferguson (rulare 2)



Suprafață Ferguson (rulare 3)



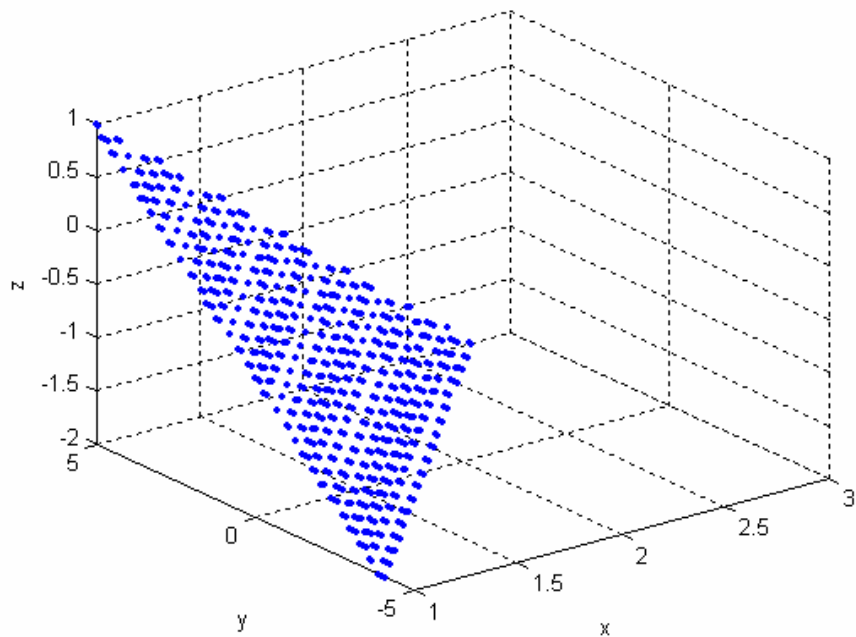
Generare de suprafețe pe baza funcției implicite

În situațiile practice în care avem la dispoziție ecuații canonice de forma $F(x, y, z) = 0$ ce descriu entități grafice de tip suprafață se folosesc algoritmi de găsire a soluțiilor *prin încercări* pe anumite domenii date.

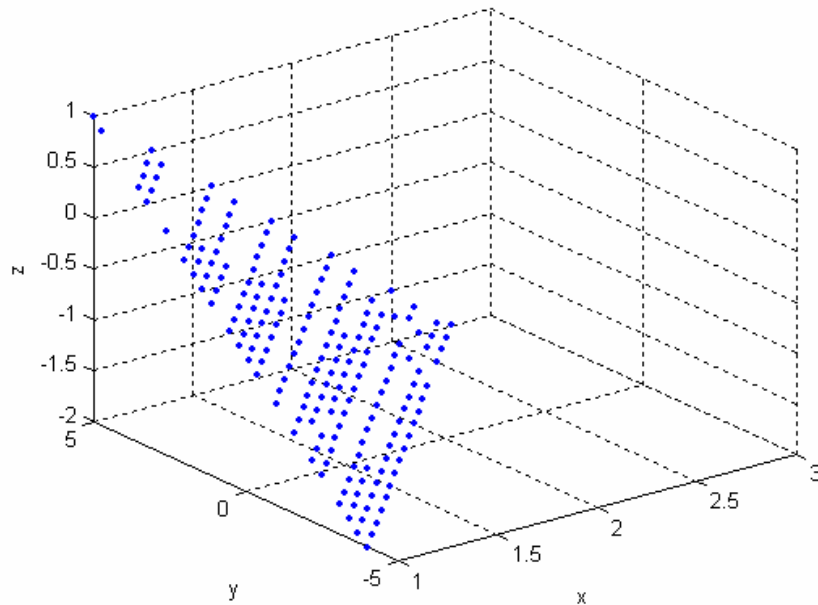
De exemplu, generarea unei suprafețe plane având la dispoziție ecuația planului în forma canonică $A \cdot x + B \cdot y + C \cdot z + D = 0$ se face prin rezolvarea iterativă a acestei ecuații. Condiția de apartenență a punctelor la plan poate fi pusă strict (ca identitate) sau ca o dublă inegalitate cu o anumită marjă de eroare. În cazul al doilea, un număr mai mare de puncte vor avea șansa să verifice condiția într-un anumit domeniu de căutare.

Dacă domeniul de căutare nu este ales corespunzător în sensul că planul nu intersectează spațiul de căutare, condiția nu va fi îndeplinită pentru nici un punct. Pentru a evita astfel de situații în locul ecuației canonice a planului se poate folosi ecuația planului prin *tăieturi*, ceea ce permite alegerea domeniului de căutare în funcție de punctele în care planul taie axele sistemului de coordonate cartezian.

```
%plan
clear
%definirea planului in forma canonica
a=-5
b=1
c=-3
d=3
%generarea punctelor de verificare a ecuatiei pe un domeniu dat
for x=1:0.1:5
    for y=-5:0.1:5
        for z=-2:0.1:2
            p=a*x+b*y+c*z+d
%conditia de apartenenta la plan si reprezentarea grafica
            if p>=-0.1 & p<=0.1
                plot3(x,y,z, '.')
                hold on
            end
        end
    end
end
end
grid
xlabel('x')
ylabel('y')
zlabel('z')
```



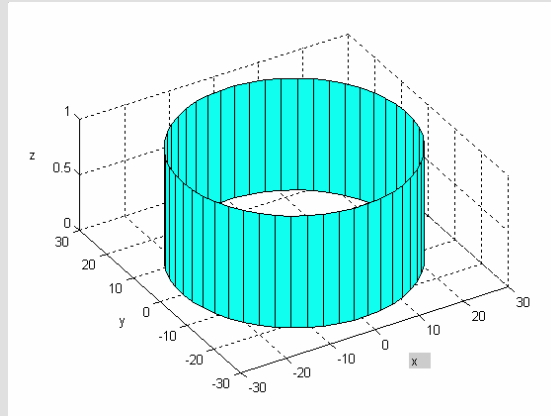
Observatie. Efectul condiției stricte de apartenență a punctelor la plan $if\ p==0$ conduce la mai puține puncte generate în domeniul de căutare dat.



Reprezentarea obiectelor în spațiu

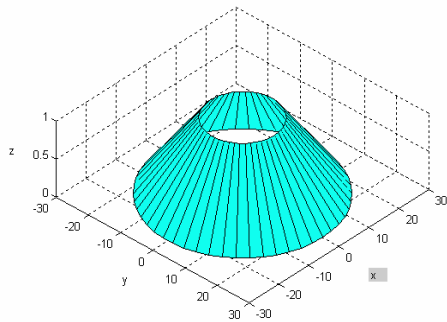
În Matlab pot fi reprezentate forme geometrice predefinite cu funcții specializate `cylinder`, `sphere`, `ellipsoid` și obiecte poliedrice cu ajutorul funcțiilor de reprezentare a suprafețelor poligonale în 3D: `fill3` și `patch`.

```
>> [x,y,z]=cylinder(25,50)
%Optional se poate preciza harta de culoare folosita
>> colormap([0.0625 1.0000 0.9375])
>> surface(x,y,z,z)
```

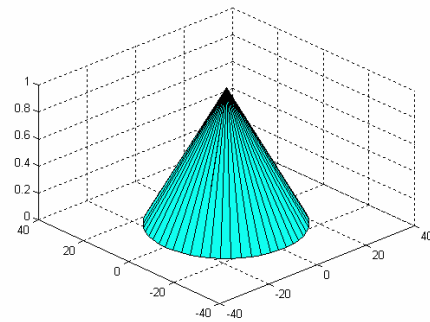


Reprezentarea suprafețelor conice

```
>> [x,y,z]=cylinder([25 10],50)
>> colormap([0.062 1.000 0.937])
>> surface(x,y,z,z)
>> grid
```

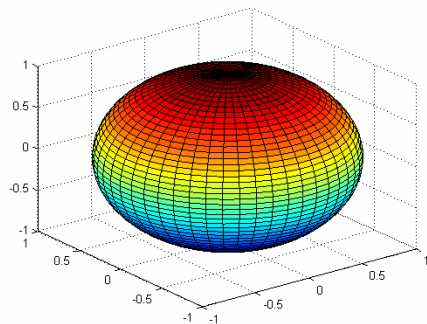


```
>> [x,y,z]=cylinder([25 0], 50)
>> colormap([0.062 1.000 0.937])
>> surface(x,y,z,z)
>> grid
```

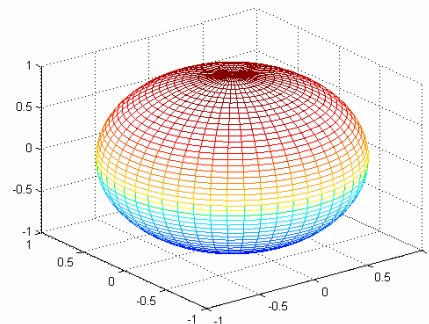


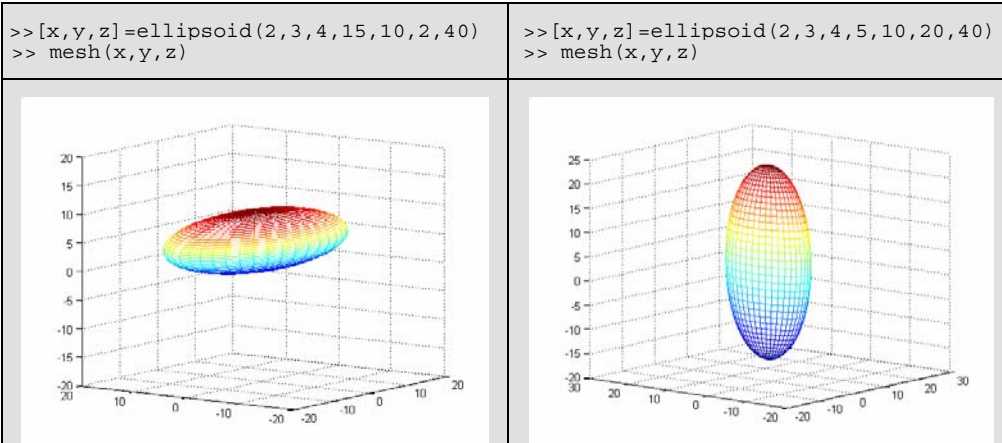
```
>> [x,y,z]=sphere(50)
```

```
>> surf(x,y,z)
```

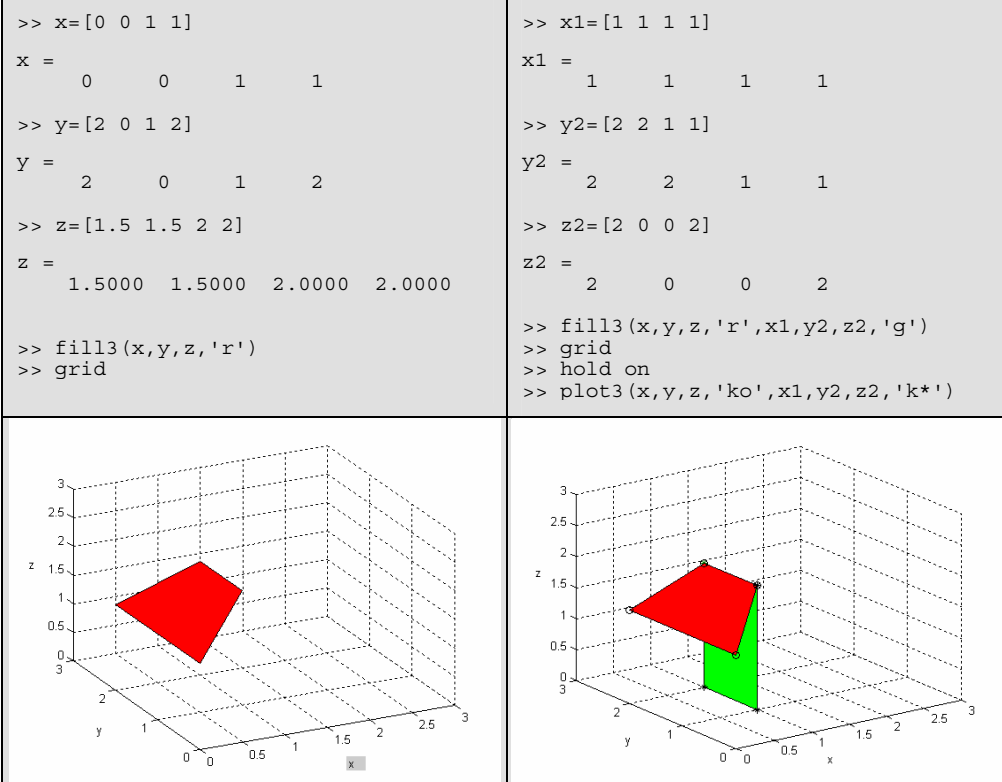


```
>> mesh(x,y,z)
```





Crearea și reprezentarea suprafețelor în 3D



6.3. Personalizarea graficelor

Personalizarea reprezentărilor grafice în Matlab include câteva categorii de funcții destinate adnotării (etichetarea) graficelor, alegerea (impunerea) domeniilor de reprezentare a funcțiilor, subdivizarea ferestrei grafice și în general controlul tuturor obiectelor grafice inclusiv controlul culorilor. Aceste funcții sunt destinate utilizării în cadrul programelor Matlab. Reamintim că utilizatorul poate personaliza reprezentările grafice în Matlab și după ce acestea au fost create în urma apelului unei funcții grafice sau a execuției unui program, acționând asupra meniului și comenzilor ce însoțesc obiectul `figure`.

Tab. 6.7. Funcții pentru personalizarea graficelor

Categoria	Sintaxa
Precizarea titlului graficului	<code>title('text')</code>
Precizarea datei curente pe grafic	<code>title(date)</code>
Precizarea etichetelor axelor	<code>xlabel('text')</code> <code>ylabel('text')</code> <code>zlabel('text')</code>
Afișarea unui text pe grafic 2D respectiv 3D în poziție specificată	<code>text(x,y,'text')</code> <code>text(x,y,z,'text')</code>
Afișarea unui text pe grafic în poziție aleasă cu mouse-ul	<code>gtext('text')</code>
Trasarea rețelelor de linii ajutătoare	<code>grid</code>
Trasarea graficului unei funcții (f) definită ca variabilă și cu restricții pe intervale și diferite opțiuni	<code>fplot(f,lims)</code> <code>fplot(f,lims,tol)</code> <code>fplot(f,lims,'linespec')</code> <code>[x,y] = fplot(f,lims,...)</code> <code>[...]= fplot(f,lims,tol,n,'linespec',p1,p2,...)</code>
	<code>ezplot(f)</code> <code>ezplot(f,[min,max])</code> <code>ezplot(f,[xmin,xmax,ymin,ymax])</code>
	<code>ezplot(x,y)</code> <code>ezplot(x,y,[tmin,tmax])</code>
	pentru curbe definite parametric $x(t), y(t)$
	<code>ezplot(...,figure)</code>
Trasare de curbe parametrice în 3D cu restricții pe intervale	<code>ezplot3(x,y,z)</code> <code>ezplot3(x,y,z,[tmin,tmax])</code> <code>ezplot3(...,'animate')</code>

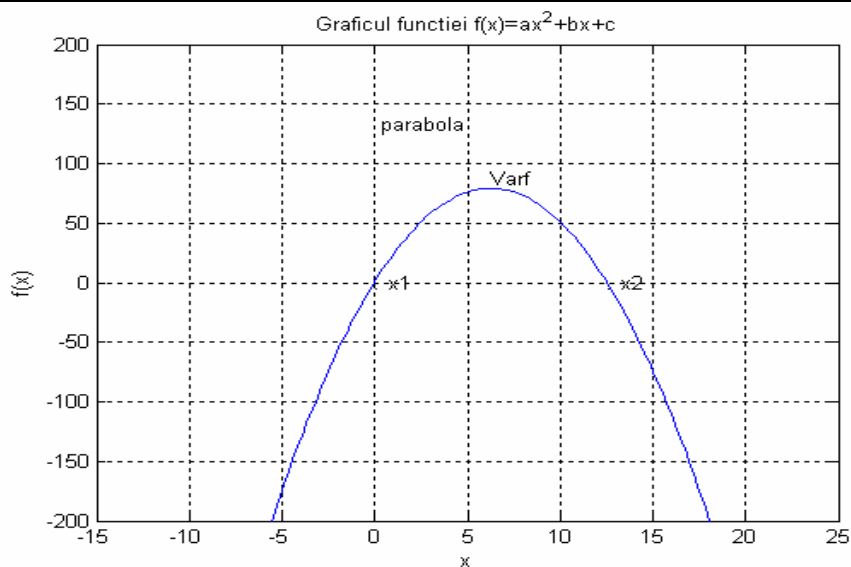
Notă. Pentru realizarea unor reprezentări grafice mai flexibile în ceea ce privește modul de personalizare se recomandă și familia de funcții grafice complementare: `ezcontour`, `ezcontourf`, `ezmesh`, `ezmeshc`, `ezsurf` și `ezsurf`. Particularitățile sintactice și detaliile de utilizare a

funcțiilor respective pot fi obținute cu comanda cunoscută `help` `nume_funcție`, din linia de comandă.

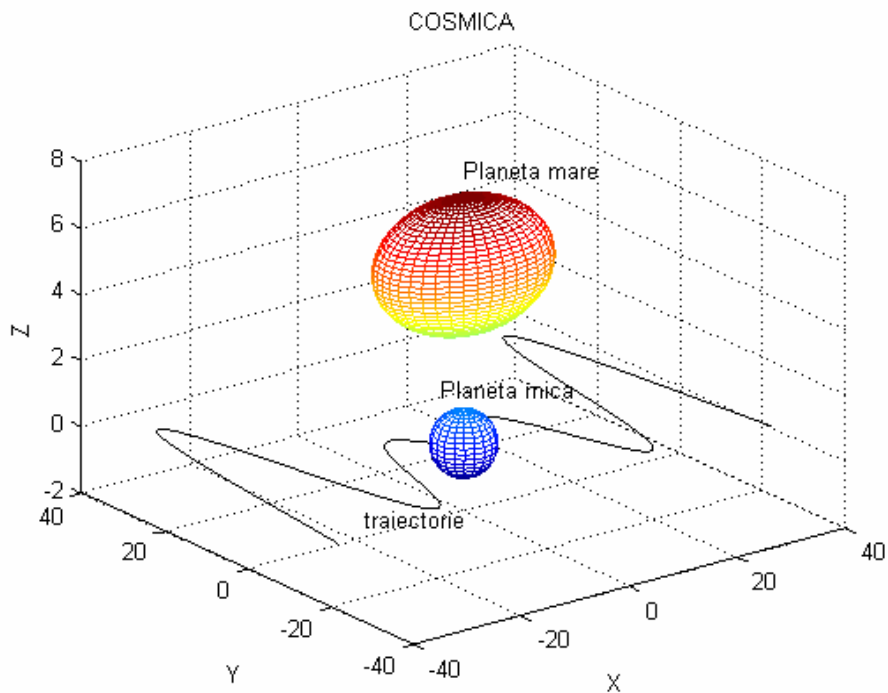
Observație. Personalizarea graficelor poate fi realizată și cu ajutorul funcțiilor pentru crearea și controlul obiectelor grafice (axe, figură, linie, text, etc.). De pildă, funcția `subplot` deja utilizată în exemple anterioare are rolul de a crea axe în mai multe poziții ale obiectului figură.

Programele prezentate în continuare exemplifică diferite moduri de personalizare a graficelor 2D și 3D utilizând funcții din Tab. 6.7 pentru adnotarea și reprezentarea graficelor cu parametrii impuși.

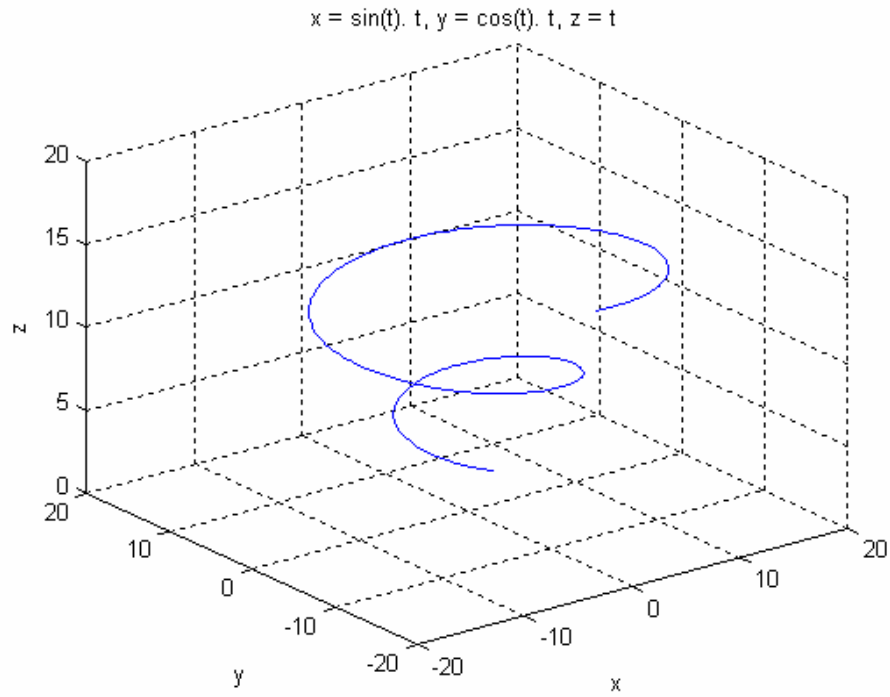
```
%grafic 2D personalizat
clear
x=-20:0.1:20
a=-2; b=25; c=1
g = inline('P1*x.^2+P2*x+P3',3)
fplot(g, [-15 25 -200 200], 1e-3, 10, '-b', a, b, c)
grid
xlabel('x')
ylabel('f(x)')
title('Graficul functiei f(x)=ax^2+bx+c')
delta=b^2-4*a*c
x12=[(-b+sqrt(delta))/(2*a) (-b-sqrt(delta))/(2*a)]
v=-b/(2*a)
fv=polyval([a b c],v)
text(x12(1,1),0,'. x1')
text(x12(1,2),0,'. x2')
text(v,fv+fv/10,'Varf')
gtext('Parabola')
```



```
%reprezentare grafica personalizata
clear
[x1,y1,z1]=ellipsoid(0,0,0,5,5,1,20)
mesh(x1,y1,z1)
hold on
[x2,y2,z2]=ellipsoid(3,4,5,15,10,2,40)
mesh(x2,y2,z2)
hold on
x=-0.5:0.01:2
fplot('sin(0.25.*x).*x', [-40 40 -40 40],'k')
xlabel('X')
ylabel('Y')
zlabel('Z')
title('COSMICA')
text(-30,-15,'traectorie')
text(-0.5,5,1.5,'Planeta mica')
text(3,4,8,'Planeta mare')
```

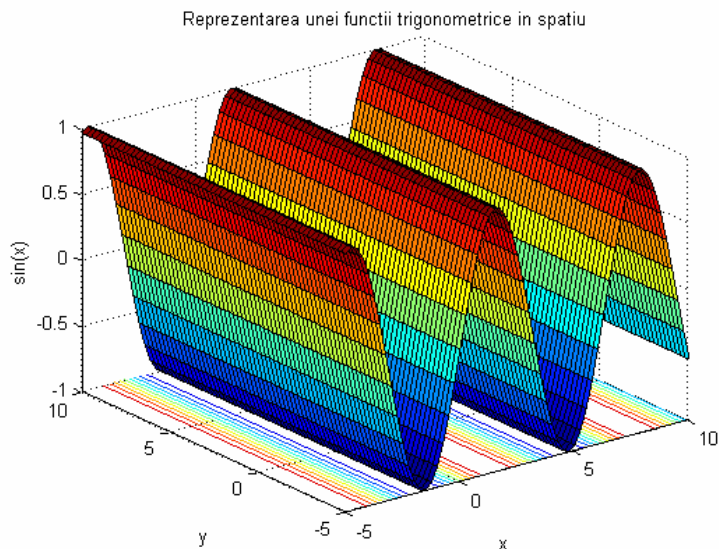


```
%reprezentare 3D parametrizata  
>> t=0:pi/100:5*pi  
>> ezplot3('sin(t).*t','cos(t).*t','t',[pi 5*pi])
```



$$\begin{cases} x(t) = \sin(t) \cdot t \\ y(t) = \cos(t) \cdot t \\ z(t) = t \end{cases}$$

```
clear
[x,y]=meshgrid(-5:.25:10);
z=sin(x);
surfc(x,y,z)
xlabel('x')
ylabel('y')
zlabel('sin(x)')
title('Reprezentarea unei functii trigonometrice in spatiu');
```



Controlul hărții de culoare

Reprezentarea formelor geometrice predefinite în 3D cu funcții specializate utilizează o culoare din paleta de culori implicită (în general albastru), dacă nu se specifică altfel. Controlul hărții de culoare se face cu funcția `colormap`, selectând componentele RGB (liniile sau linia) dorite din matricea de culoare dată în tabelul 6.6. Prima linie a matricii corespunde culorii *albastru*, iar ultima linie culorii *roșu*. Se recomandă asocierea matricii de culoare la o variabilă, de exemplu `C=colormap` și apoi efectuarea oricărei selecții dorite (vezi subcapitolul 4.2.3. Manipularea matricelor). De pildă, selectarea unei palete de culori din care se exclud nuanțele de albastru se poate face cu instrucțiunea `colormap(C(25:60,:))`. Reamintim că setarea proprietății `color` se poate face pentru obiectele grafice în Matlab, așa cum s-a arătat în subcapitolul 6.1.

Tab. 6.6. Paleta de culori implicită

```
>> colormap
ans =
    0         0    0.5625
    0         0    0.6250
    0         0    0.6875
    0         0    0.7500
    0         0    0.8125
    0         0    0.8750
    0         0    0.9375
    0         0    1.0000
    0    0.0625    1.0000
    0    0.1250    1.0000
    0    0.1875    1.0000
    0    0.2500    1.0000
    0    0.3125    1.0000
    0    0.3750    1.0000
    0    0.4375    1.0000
    0    0.5000    1.0000
    0    0.5625    1.0000
    0    0.6250    1.0000
    0    0.6875    1.0000
    0    0.7500    1.0000
    0    0.8125    1.0000
    0    0.8750    1.0000
    0    0.9375    1.0000
    0    1.0000    1.0000
    0.0625    1.0000    0.9375
    0.1250    1.0000    0.8750
    0.1875    1.0000    0.8125
    0.2500    1.0000    0.7500
    0.3125    1.0000    0.6875
    0.3750    1.0000    0.6250
    0.4375    1.0000    0.5625
    0.5000    1.0000    0.5000
    0.5625    1.0000    0.4375
    0.6250    1.0000    0.3750
    0.6875    1.0000    0.3125
    0.7500    1.0000    0.2500
    0.8125    1.0000    0.1875
    0.8750    1.0000    0.1250
    0.9375    1.0000    0.0625
    1.0000    1.0000         0
    1.0000    0.9375         0
    1.0000    0.8750         0
    1.0000    0.8125         0
    1.0000    0.7500         0
    1.0000    0.6875         0
    1.0000    0.6250         0
    1.0000    0.5625         0
    1.0000    0.5000         0
    1.0000    0.4375         0
    1.0000    0.3750         0
    1.0000    0.3125         0
    1.0000    0.2500         0
    1.0000    0.1875         0
    1.0000    0.1250         0
    1.0000    0.0625         0
    1.0000         0         0
    0.9375         0         0
    0.8750         0         0
    0.8125         0         0
    0.7500         0         0
    0.6875         0         0
    0.6250         0         0
    0.5625         0         0
    0.5000         0         0
```