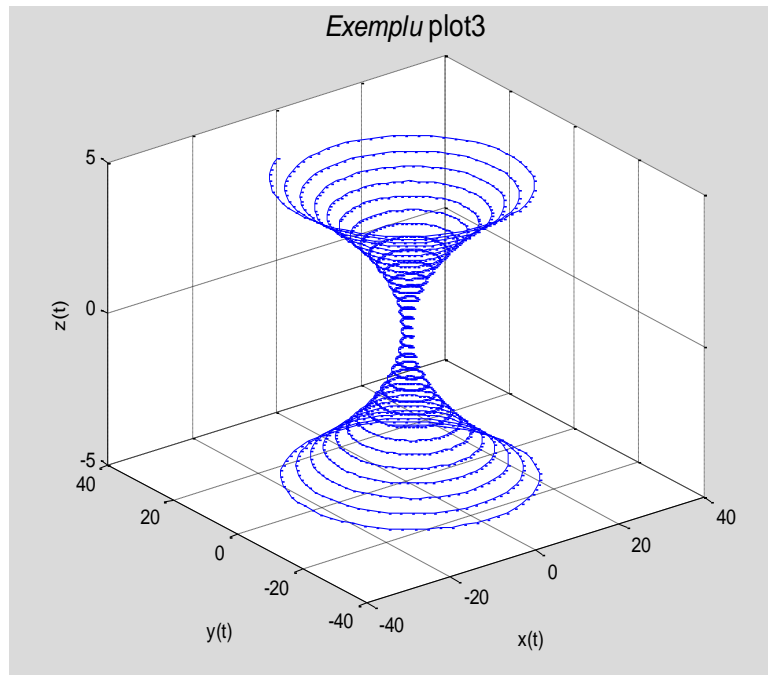


# Grafice 3D

## Curbe 3D

Funcția `plot3` este un analog tridimensional al lui `plot`. Exemplu:

```
clf
t = -5:0.005:5;
x = (1+t.^2).*sin(20*t);
y = (1+t.^2).*cos(20*t);
z=t;
plot3(x,y,z)
grid on
xlabel('x(t)'), ylabel('y(t)'), zlabel('z(t)')
title('{\itExemplu} plot3','FontSize',14)
```



Limitele de axe în spațiul tridimensional se determină automat, dar ele pot fi schimbate cu

```
axis([xmin, xmax, ymin, ymax, zmin, zmax])
```

În afară de `xlim` și `ylim`, există și `zlim`, prin care se pot schimba limitele pe axa  $z$ .

## Suprafețe

O funcție de două variabile  $z=f(x,y)$  se reprezintă cu ajutorul valorilor ei pe o mulțime discretă  $z_{i,j} = z(x_i; y_j)$ , unde  $x_i; i = 1, \dots, m$  și  $y_j; j = 1, \dots, n$  sunt puncte de pe axele  $x$  și  $y$  luate în ordine crescătoare. Produsul lor cartezian ne dă o grilă rectangulară carteziană.

Pentru ilustrare, să considerăm grila definită prin

$$x_{i,j} = x_i = -2 + 0.2(i-1); \quad 1 \leq i \leq 21$$

$$y_{i,j} = y_j = -2 + 0.2(j-1); \quad 1 \leq j \leq 21$$

și valorile funcției date de

$$z_{i,j} = x_i \exp(-x_i^2 - y_j^2)$$

Funcția `meshgrid` este utilă la obținerea grilelor. Ea are forma

$$[X, Y] = \text{meshgrid}(x, y)$$

$$[X, Y] = \text{meshgrid}(x)$$

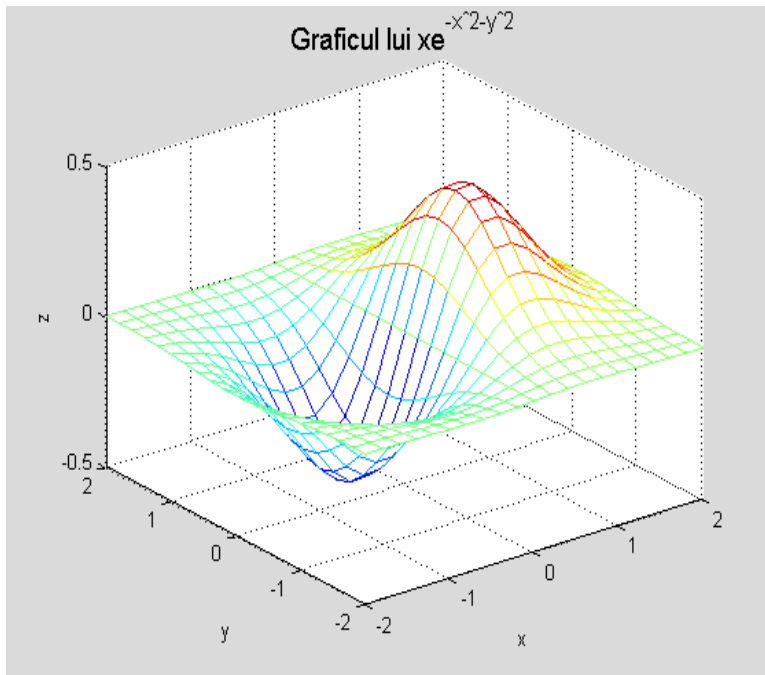
$$[X, Y, Z] = \text{meshgrid}(x, y, z)$$

Prima formă transformă domeniul specificat de vectorii  $x$  și  $y$  în tablourile bidimensionale  $X$  și  $Y$ , care pot fi utilizate la evaluarea funcțiilor de două variabile și la obținerea graficelor tridimensionale cu ajutorul funcțiilor `mesh` și `surf` și a variantelor lor. Liniile tabloului  $X$  sunt copii ale lui  $x$ , iar coloanele lui  $Y$  sunt copii ale lui  $y$ . Această regulă trebuie respectată și dacă elementele lui  $z$  se obțin prin cicluri `for/end`. Forma a doua are același efect ca  $[X, Y] = \text{meshgrid}(x, x)$ , iar cea

de-a treia este utilizată pentru producerea unor tablouri tridimensionale, folosite la calculul valorilor unor funcții tridimensionale sau la obținerea unor grafice volumetrice.

Funcția `mesh(x, y, z)`, unde `x`, `y` și `z` sunt tablouri bidimensionale de aceeași dimensiune, generează o reprezentare de tip cadru de sârmă a suprafeței parametrice specificate de `x`, `y` și `z`. Exemplu:

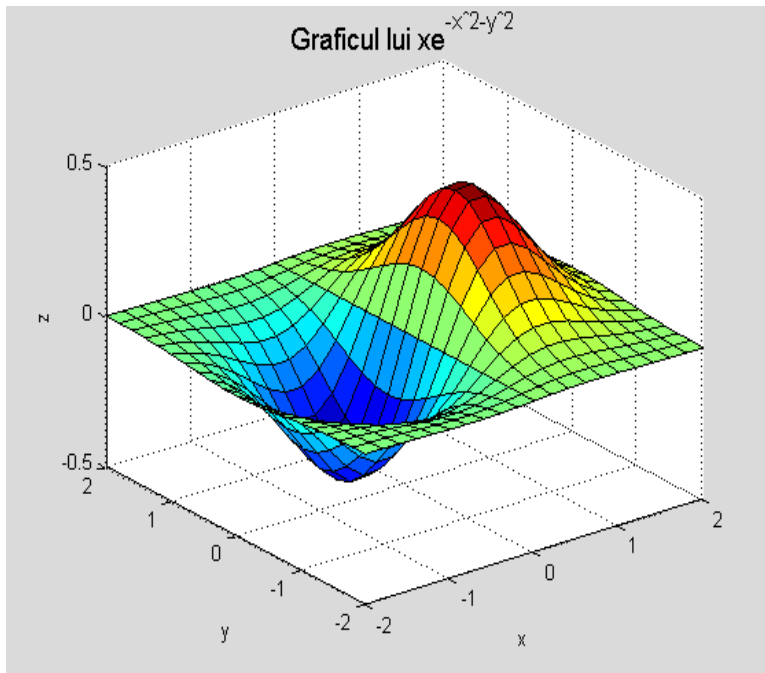
```
clf
xa=-2:0.2:2;
ya=-2:0.2:2;
[x,y]=meshgrid(xa,ya);
z=x.*exp(-x.^2-y.^2);
mesh(x,y,z)
title('Graficul lui  $e^{-x^2-y^2}$ ','FontSize',14)
xlabel('x'),ylabel('y'),zlabel('z')
```



Funcția `surf` produce grafice asemănătoare cu cele obținute cu ajutorul funcției `mesh`, dar cu deosebirea că celulele suprafeței sunt colorate. Dacă în exemplul precedent înlocuim `mesh` cu `surf` se obține graficul:

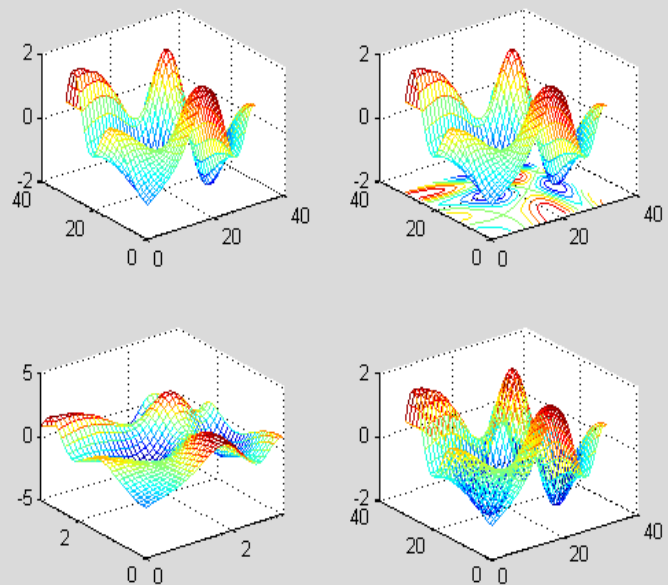
```
clear,clf
xa=-2:0.2:2;
ya=-2:0.2:2;
[x,y]=meshgrid(xa,ya);
```

```
z=x.*exp(-x.^2-y.^2);  
surf(x,y,z)  
title('Graficul lui  $xe^{-x^2-y^2}$ ','FontSize',14)  
xlabel('x'),ylabel('y'),zlabel('z')
```



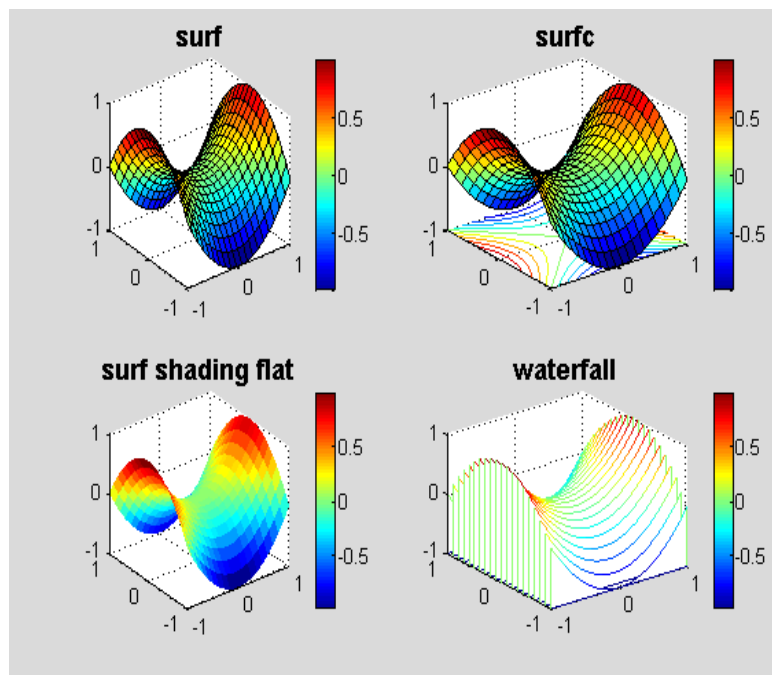
Alte exemple:

```
x = 0:0.1:pi; y=0:0.1:pi;  
[X,Y]=meshgrid(x,y);  
Z=sin(Y.^2+X)-cos(Y-X.^2);  
subplot(221)  
mesh(Z)  
subplot(222)  
meshc(Z)  
subplot(223)  
mesh(x,y,Z)  
axis([0 pi 0 pi -5 5])  
subplot(2,2,4)  
mesh(Z)  
hidden off
```





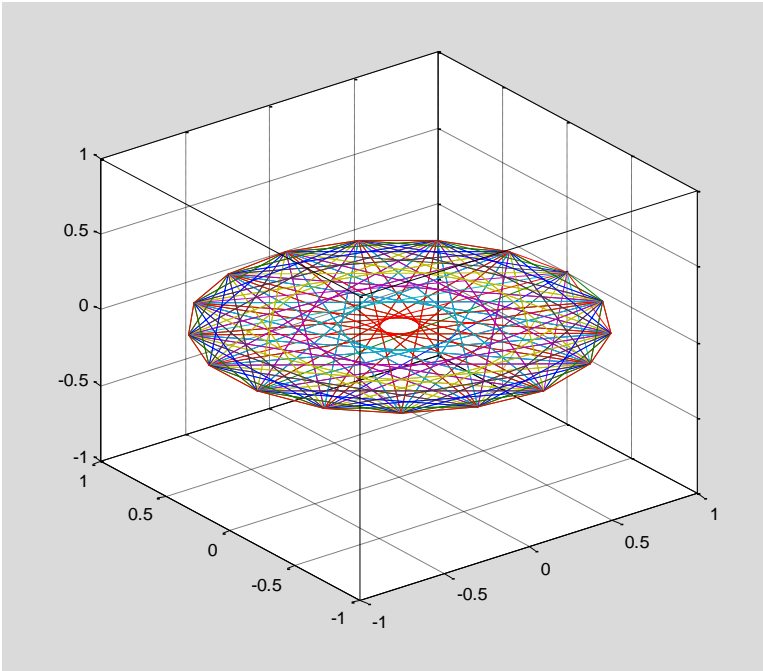
```
clf
[X,Y]=meshgrid(linspace(-1,1,20)); Z=X.^2-Y.^2;
FS = 'FontSize';
subplot(2,2,1), surf(X,Y,Z),
title('\bf{surf}',FS,14), colorbar
subplot(2,2,2), surfc(X,Y,Z),
title('\bf{surfc}',FS,14), colorbar
subplot(2,2,3), surf(X,Y,Z), shading flat
title('\bf{surf} shading flat',FS,14), colorbar
subplot(2,2,4), waterfall(X,Y,Z)
title('\bf{waterfall}',FS,14), colorbar
```



Graficele tridimensionale exemplificate utilizează unghiurile de vizualizare implicite ale MATLAB. Acestea pot fi modificate cu `view`. Apelul `view(a,b)` alege unghiul de rotație în sens invers acelor de ceasornic în jurul axei  $z$  (azimutul) de  $a$  grade și unghiul față de planul  $xOy$  (elevația) de  $b$  grade. Implicit este `view(-37.5, 30)`. Instrumentul `rotate 3D` de pe bara de instrumente a ferestrei figurii permite utilizarea mouse-ului pentru schimbarea unghiurilor de vedere.

Este posibil să vedem un grafic 2D ca pe unul 3D, utilizând comanda `view` pentru a da unghiurile de vedere, sau mai simplu utilizând `view(3)`. Exemplu

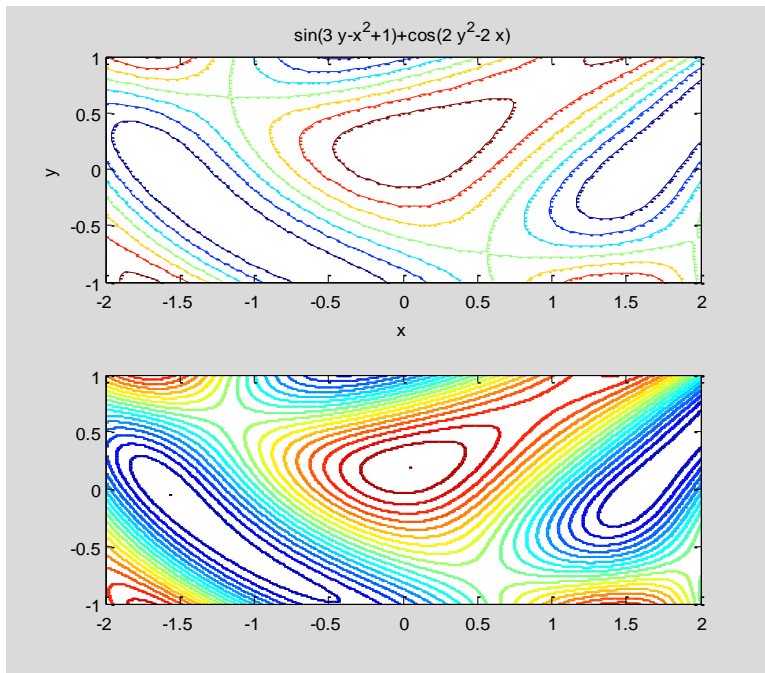
```
clf, plot(fft(eye(17))); view(3); grid
```



## Contururi

O facilitate ușor de utilizat de desenare a contururilor este oferită de `ezcontour`. Apelul lui `ezcontour` în exemplul următor produce contururi pentru funcția  $\sin(3y - x^2 + 1) + \cos(2y^2 - 2x)$  pe domeniul dat de  $-2 \leq x \leq 2$  și  $-1 \leq y \leq 1$ ; rezultatul se poate vedea în jumătatea de sus a figurii .

```
subplot(211)
ezcontour('sin(3*y-x^2+1)+cos(2*y^2-2*x)', [-2,2,-1,1]);
%
x=-2:.01:2; y=-1:0.01:1;
[X,Y] = meshgrid(x,y);
Z =sin(3*Y-X.^2+1)+cos(2*Y.^2-2*X);
subplot(212)
contour(x,y,Z,20)
```



De notat că nivelurile de contur au fost alese automat. Pentru jumătatea de jos a figurii s-a utilizat funcția `contour`. Întâi se fac inițializările  $x = -2:.01:2$  și  $y = -1:.01:1$  pentru a obține puncte mai apropiate în domeniul respectiv. Apoi se execută  $[X,Y] = \text{meshgrid}(x,y)$ , care obține matricele  $X$  și  $Y$  astfel încât fiecare linie a lui  $X$  să fie o copie a lui  $x$  și fiecare coloană a lui  $Y$  să fie o copie a vectorului  $y$ . Matricea  $Z$  este apoi generată prin operații de tip tablou din  $X$  și  $Y$ ;  $Z(i,j)$  memorează valoarea funcției corespunzând lui  $x(j)$  și  $y(i)$ . Aceasta este forma cerută de `contour`. Apelul `contour(x,y,Z,20)` spune MATLAB să privească  $Z$  ca fiind

formată din cote deasupra planului  $xOy$  cu spațierea dată de  $x$  și  $y$ . Ultimul argument de intrare spune că se vor utiliza 20 de niveluri de contur; dacă acest argument este omis, MATLAB va alege automat numărul de niveluri de contur.

Funcția `contour` se poate utiliza și la reprezentarea funcțiilor implicite cum ar fi

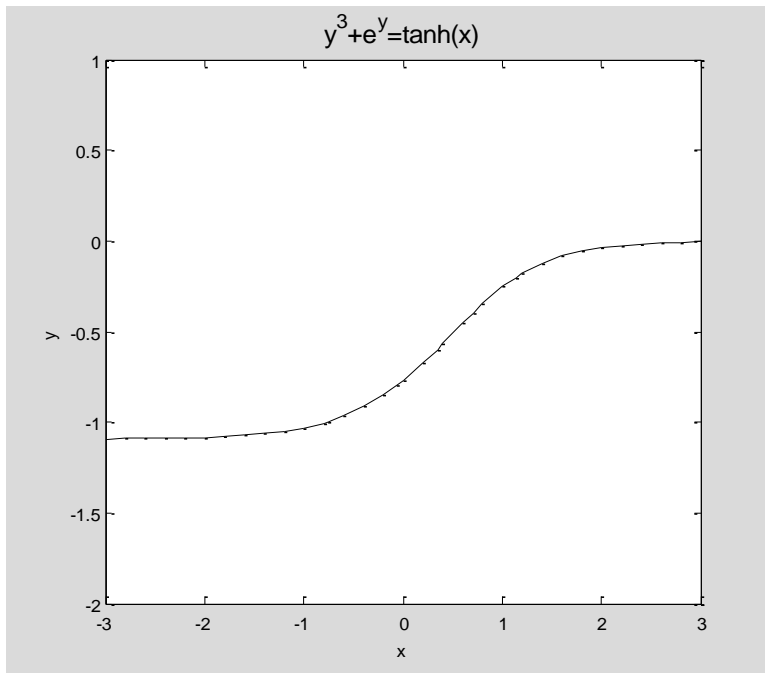
$$y^3 + e^y = \tanh x.$$

Pentru a o reprezenta grafic, rescriem ecuația sub forma

$$f(x, y) = y^3 + e^y - \tanh x$$

și desenăm conturul pentru  $f(x, y)$ .

```
clf; xm=-3:0.2:3; ym=-2:0.2:1;  
[x,y]=meshgrid(xm,ym);  
f=y.^3+exp(y)-tanh(x);  
contour(x,y,f,[0,0],'k-')  
xlabel('x'); ylabel('y');  
title('y^3+e^y=tanh(x)', 'FontSize',14)
```



plot3*	grafic simplu $x$ - $y$ - $z$
contour*	Contur
contourf*	contur plin
contour3	contur 3D
mesh	reprezentare wire-frame
meshc*	reprezentare wire-frame plus contururi
meshz	suprafață wire-frame cu cortină

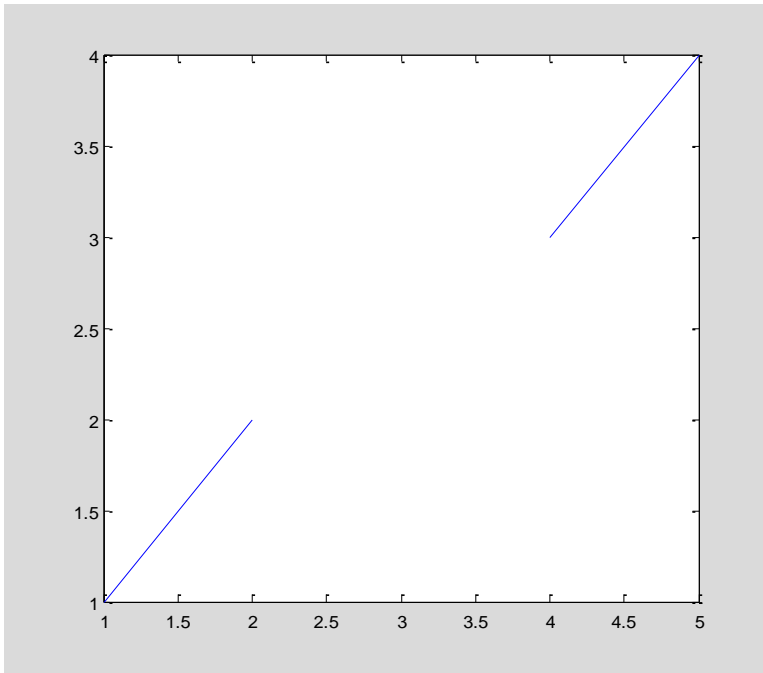
<code>surf*</code>	suprafață plină
<code>surfc*</code>	suprafață plină plus contururi
<code>waterfall</code>	wire-frame unidirecțional
<code>bar3</code>	bare 3D
<code>bar3h</code>	bare 3D orizontale
<code>pie3</code>	grafice sector 3D
<code>fill3</code>	poligon umplut tridimensional
<code>comet3</code>	curbă 3D animată
<code>scatter3</code>	nor de puncte 3D

Aceste funcții `fun` au correspondent `ezfun`

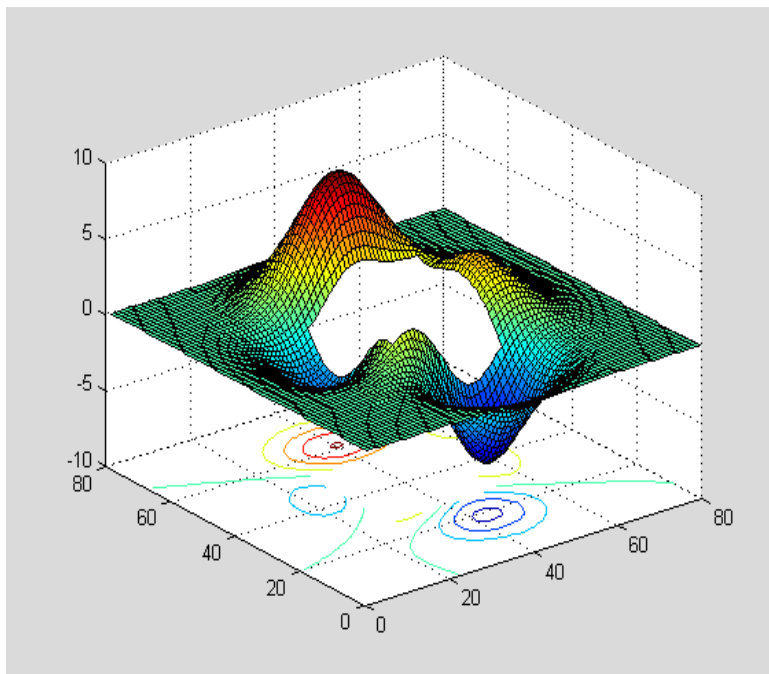
O trăsătură comună tuturor funcțiilor grafice este aceea că valorile NaN sunt interpretate ca „date lipsă” și nu sunt reprezentate. De exemplu,

```
plot([1 2 NaN 3 4])
```





desenează două linii disjuncte și nu unește punctele 2 și 3, în timp ce  
**`A=peaks(80) ; A(28:52,28:52)=NaN; surf(A)`**



produce graficul `surf` cu gaură din figură. (Funcția `peaks` din MATLAB are expresia

$$z = 3*(1-x).^2.*\exp(-(x.^2) - (y+1).^2) \dots$$

$$- 10*(x/5 - x.^3 - y.^5).*\exp(-x.^2-y.^2) \dots$$

$$- 1/3*\exp(-(x+1).^2 - y.^2)$$

și generează o matrice de cote utilă pentru a testa și demonstra facilitățile grafice 3D.)

## Salvarea și imprimarea graficelor

Comanda `print` permite listarea unui grafic la imprimantă sau salvarea lui pe disc într-un format grafic sau sub formă de fișier M. Formatul ei este:

```
print -dperiferic -optiuni numefisier
```

Ea are mai multe opțiuni, care pot fi vizualizate cu `help print`. Dintre tipurile de periferice admise amintim:

- `dps` - Postscript pentru imprimante alb-negru;
- `dpsc` - Postscript pentru imprimante color;
- `dps2` - Postscript nivelul 2 pentru imprimante alb-negru;
- `dpsc2` - PostScript nivelul 2 pentru imprimante color;
- `deps` - Encapsulated PostScript pentru imprimante alb-negru;
- `depssc` - Encapsulated PostScript pentru imprimante color;
- `deps2` - Encapsulated PostScript nivelul 2 pentru imprimante alb-negru;
- `depssc2` - Encapsulated PostScript nivelul 2 pentru imprimante color;
- `djpeg` - `<nn>` - imagine JPEG la nivelul de calitate `nn` (implicit `nn=75`).

Dacă imprimanta dumneavoastră este setată corespunzător, comanda `print` va trimite conținutul figurii curente spre ea. Comanda

```
print -deps2 myfig.eps
```

crează un fișier Postscript încapsulat alb și negru, nivelul 2, numit `myfig.eps`, care poate fi listat pe o imprimantă PostScript sau inclus într-un document. Acest fișier poate fi încorporat într-un document LATEX, așa cum se schițează mai jos:

```
\documentclass{article}
\usepackage[dvips]{graphics}
...
\begin{document}
...
\begin{figure}
\begin{center}
\includegraphics[width=8cm]{myfig.eps}
\end{center}
\caption{...}
\end{figure}
...
\end{document}
```

Comanda `print` se poate utiliza și în formă funcțională . Pentru a ilustra utilitatea formei funcționale, exemplul următor generează o secvență de cinci figuri și le salvează în fișierele

`fig1.eps`, ..., `fig5.eps`:

```
x = linspace(0,2*pi,50);
for i=1:5
plot(x,sin(i*x))
```

```
print(-deps2', ['fig', int2str(i), '.eps'])  
end
```

Al doilea argument al comenzii `print` este format prin concatenare, utilizând funcția `int2str`, care convertește un întreg în șir. Astfel, de exemplu, pentru `i=1`, instrucțiunea `print` este echivalentă cu `print('-deps2', 'fig1.eps')`.

Comanda `saveas` salvează o figură într-un fișier care apoi poate fi încărcat de către MATLAB. De exemplu,

```
saveas(gcf, 'myfig', 'fig')
```

salvează figura curentă în format binar FIG, care poate fi încărcat în MATLAB cu comanda `open('myfig.fig')`. Se pot salva și imprima figuri din meniul `File` al ferestrei figurii.