1.11.2 Instrucţiunile `tic` şi `toc`

Pentru obţinerea duratei execuţiei unei secvenţe `secv` de instrucţiuni Matlab avem următoarea succesiune de instrucţiuni:

```
tic
  secv
toc
```

unde `tic` marchează începutul contorizării timpului, iar `toc` anunţă încheierea contorizării, variabila `toc` conţinând timpul măsurat în secunde.

1.12 Grafică tridimensională

Grafica tridimensională sau altfel numită `3D` este axată în sistemul Matlab pe două direcţii: reprezentarea curbelor în spaţiu şi reprezentarea suprafeţelor.

1.12.1 Instrucţiunea `plot3`

Instrucţiunea `plot3` este utilizată pentru reprezentarea curbelor în spaţiu şi în principiu nu se deosebeşte mult de instrucţiunea `plot` din grafica bidimensională.

Formele întâlnite pentru această instrucţiune sunt:

```
plot3(x,y,z)
plot3(x,y,z,s)
plot3(x1,y1,z1,x2,y2,z2,...)
plot3(x1,y1,z1,s1,x2,y2,z2,s2,...)
```

Dacă `x`, `y`, `z` sunt vectori de aceeaşi lungime `m`, atunci se unesc punctele de coordonate $(\mathtt{x}_i, \mathtt{y}_i, \mathtt{z}_i)$, $i = \overline{1, \mathtt{m}}$, printr–o linie poligonală de tipul precizat prin parametrul `s`, de la instrucţiunea `plot`, iar în lipsa acestui parametru, sistemul Matlab efectuează o alegere implicită a tipului curbei. Desigur că netezimea curbei este cu atât mai bună cu cât numărul punctelor este mai mare.

Dacă `x`, `y`, `z` sunt matrice de acelaşi tip `(m,n)`, atunci pentru fiecare coloană se obţine câte o linie poligonală, adică `n` linii poligonale. Acestea sunt generate respectiv de punctele de coordonate $\left(\mathtt{x}_{ij}, \mathtt{y}_{ij}, \mathtt{z}_{ij}\right)_{i=1}^{\mathtt{m}}$, pentru fiecare $j = \overline{1, \mathtt{n}}$, cu aceeaşi observaţie privind parametrul `s`, care precizează tipul curbei. În acest caz pe aceeaşi figură vor fi obţinute mai multe grafice.

Ultimele două forme sunt extinderi ale primelor două, pentru a putea controla mai bine în special tipul curbelor cu ajutorul parametrilor de tip `s`.

Remarcăm de asemenea extinderea acţiunilor comenzilor ce controlează un grafic din cazul bidimensional pentru cazul tridimensional.

Programul 1.12.1. Prezentăm un program care ilustrează mişcarea unei particule din punctul `A` pe o spirală până în punctul `B`.

```
pas=input('pas='); h=input('h=');
t=0:pas:h; r=exp(-0.2*t); th=pi*t*0.5;
x=r.*cos(th); y=r.*sin(th); z=t;
plot3(x,y,z), hold on
plot3([1,1],[-0.5,0],[0,0])
text(1,-0.7,0,'A'), n=length(t);
text(x(n),y(n),h+2,'B')
xlabel('x'), ylabel('x'), zlabel('z')
```

Să explicăm pe scurt o parte a instrucţiunilor acestui program.

Figura este obţinută în două etape, din cauza aceasta a fost introdusă instrucţiunea `hold on`. Prima instrucţiune `plot3` trasează efectiv graficul spiralei, care se mai completează prin segmentul din planul `xoy` delimitat de punctele de coordonate `(1,-0.5,0)` şi `(1,0,0)`, trasat cu a doua instrucţiune `plot3`. Cele două instrucţiuni au acelaşi rezultat ca şi instrucţiunea

```
plot3(x,y,z,[1,1],[-0.5,0],[0,0])
```

doar că spirala va fi trasată cu o anumită culoare, iar segmentul din planul `xoy` cu alta. Dacă vrem ca întreaga curbă să fie trasată cu aceeaşi culoare se poate apela la parametrul de tip `s`.

Se mai observă că poziţia iniţială `A` a punctului şi poziţia finală `B` sunt afişate cu instrucţiunile `text`, iar etichetele celor trei axe de coordoante sunt date prin instrucţiunile de tip `label`.

După executarea programului, cu `h=20` şi `pas=0.01`, s–a obţinut graficul din Figura 1.17.

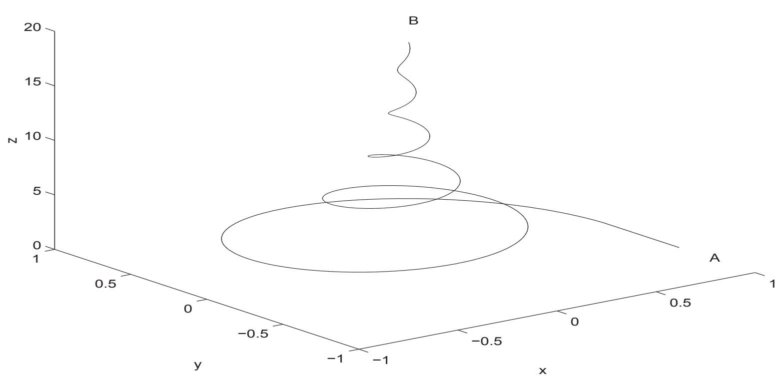

Figura 1.17: Mişcarea pe spirală

1.12.2 Instrucţiunea `ezplot3`

Pentru reprezentarea grafică a curbelor în spaţiu, poate fi folosită instrucţiunea `ezplot3`, care se lansează prin una din comenzile:

```
ezplot3('x','y','z')
ezplot3('x','y','z',[a,b])
ezplot3('x','y','z','animate')
ezplot3('x','y','z',[a,b],'animate')
```

Parametrii `x`, `y` şi `z`, conţin expresii algebrice ale reprezentărilor curbei în funcţie de un parametru `t`, din `[a,b]`. Valoarea implicită a acestui interval este $[0,2\pi]$.

Prezenţa parametrului `animate` produce pe figură un buton cu numele `repeat`, care prin activare prezintă modul de deplasare a unui punct pe curba reprezentată grafic.

De exemplu, dacă se consideră comanda

```
ezplot3('sin(t)','cos(t)','t',[0,10*pi],'animate')
```

se va obţine graficul din Figura 1.18, în care se observă şi butonul `repeat`.

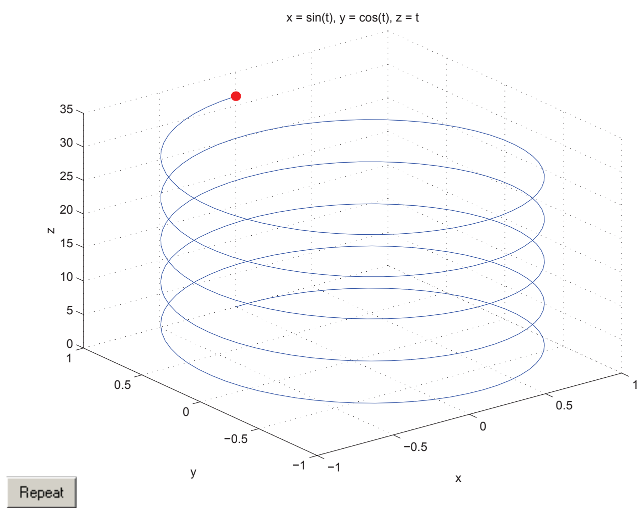

Figura 1.18: $x\,(t) = \sin\,(t)$, $y\,(t) = \cos\,(t)$, $z(t) = t$, $t \in [0, 10\pi]$

1.12.3 Instrucţiunile `meshgrid`, `mesh` şi `surf`

Reprezentarea grafică a unei suprafeţe, dată prin funcţia $z = f\,(x, y)$, pe domeniul $\boldsymbol{D} \subset \mathbb{R}^2$, cuprinde două etape.

Prima dată se realizează o reţea rectangulară de puncte, cu ajutorul instrucţiunii `meshgrid`, care are sintaxa

```
[X,Y]=meshgrid(x,y)
```

unde `x` şi `y` sunt vectori, în general de lungimi diferite, `m` şi `n`, care conţin puncte de pe axele `ox` şi `oy` şi care sunt folosite la generarea reţelei rectangulare $\left(\mathrm{x}_i, \mathrm{y}_j\right)$ $i = \overline{1,\mathrm{m}}$, $j = \overline{1,\mathrm{n}}$.

Rezultatul executării instrucţiunii `meshgrid`, are ca efect generarea matricelor `X` şi `Y` având acelaşi tip `(n,m)`. Fiecare linie a matricei `X` este formată din componentele vectorului `x`, iar fiecare coloană a matricei `Y` este formată din componentele vectorului `y`. În acest fel punctul de coordonate $\left(\mathrm{x}_i, \mathrm{y}_j\right)$ al reţelei rectangulare, se poate exprima, cu ajutorul matricelor `X` şi `Y`, prin perechea `(X(j,i),Y(j,i))`.

Înainte de a trece la a doua etapă, să facem observaţia că dacă `x` coincide cu `y`, atunci se poate utiliza

```
[X,Y]=meshgrid(x)
```

Reprezentarea grafică a funcţiei $z = f(x, y)$, se face cu ajutorul instrucţiunii `mesh`, care are una din formele:

```
mesh(Z)
mesh(Z,C)
mesh(x,y,Z)
mesh(x,y,Z,C)
mesh(X,Y,Z)
mesh(X,Y,Z,C)
```

De la început, să remarcăm faptul că parametrul `C` precizează scara culorilor. Când lipseşte acest parametru, scara culorilor este dată de parametrul `Z`, care reprezintă valorile funcţiei $z = f(x, y)$ pe o reţeaua rectangulară. În acest ultim caz culoarea va fi proporţională cu înălţimea z din punctul (x, y).

De asemenea, trebuie să remarcăm faptul că reprezentarea grafică a suprafeţei este de forma unei reţele curbilinii sprijinită pe înălţimile date prin matricea `Z`.

Parametrii `X`, `Y` şi `Z` sunt matrice de aceleaşi dimensiuni, dacă avem în vedere cum au fost obţinute matricele `X` şi `Y`, atunci toate aceste trei matrice sunt de tipul `(n,m)`.

Parametrii `x` şi `y` sunt vectori de tipul celor ce generează matricele `X` şi `Y` prin instrucţiunea `meshgrid`, adică dacă aceştia au lungimile `m` şi respectiv `n`, matricea `Z` va fi de tipul `(n,m)`, iar punctele pe care se sprijină reţeaua curbilinie, care reprezintă graficul funcţiei, au coordonatele `(x(j),y(i),Z(i,j))`.

Dacă primii doi parametri lipsesc, adică suntem în cazul primelor două forme ale instrucţiunii `mesh`, acestea sunt echivalente respectiv cu următoarele două, prin considerarea valorilor implicite `x=1:m` şi `y=1:n`.

Instrucţiunea `surf` are aceeaşi sintaxă ca instrucţiunea `mesh`, adică

```
surf(Z)
surf(Z,C)
```

```
surf(x,y,Z)
surf(x,y,Z,C)
surf(X,Y,Z)
surf(X,Y,Z,C)
```

unde parametrii au aceleaşi caracteristici. Deosebirea în reprezentarea grafică este că prin instrucţiunea `surf` se produce o colorare a patrulaterelor obţinute prin curbele care generează suprafaţa corespunzătoare funcţiei $z = f(x, y)$.

Programul 1.12.2. Vom scrie un program care să reprezinte grafic funcţia

$$z = f(x, y) = \frac{1}{2\pi} \mathrm{e}^{-\frac{1}{2}\left(x^2+y^2\right)},$$

pe domeniul $\boldsymbol{D} = [-1, 1] \times [-1, 1]$, în două figuri alăturate, odată cu instrucţiunea `mesh` şi apoi cu instrucţiunea `surf`.

```
clear, clf
m=input('m=');
h=1/m; x=-1:h:1;
[X,Y]=meshgrid(x);
Z=1/(2*pi)*exp(-0.5*(X.^2+Y.^2));
subplot(1,2,1), mesh(X,Y,Z)
title('Grafica  prin MESH')
xlabel('x'), ylabel('y'), zlabel('z')
subplot(1,2,2), surf(X,Y,Z)
title('Grafica prin SURF')
xlabel('x'), ylabel('y'), zlabel('z')
```

Executarea programului, cu $m = 6$, conduce la reprezentările grafice din Figura 1.19.

În încheierea acestui paragraf, amintim că există posibilitatea de umbrire a unei suprafeţe cu ajutorul comenzii

```
shading tip
```

unde `tip` poate fi:

- `faceted` (implicit) – umbreşte fiecare patrulater de pe suprafaţa trasată cu o intensitate fixă şi trasează laturile patrulaterelor;
- `flat` – umbreşte fiecare patrulater de pe suprafaţa trasată cu o intensitate fixă, fără trasarea laturilor patrulaterelor;
- `interp` – umbrirea fiecărui patrulater de pe suprafaţa trasată se face în mod gradat, printr–un procedeu de interpolare, care asigură o trecere netedă de la un patrulater la altul, fără trasarea laturilor patrulaterelor.

Programul 1.12.3. Să scriem un program care efectuează umbrirea suprafeţei reprezentată grafic prin Programul 1.12.2:

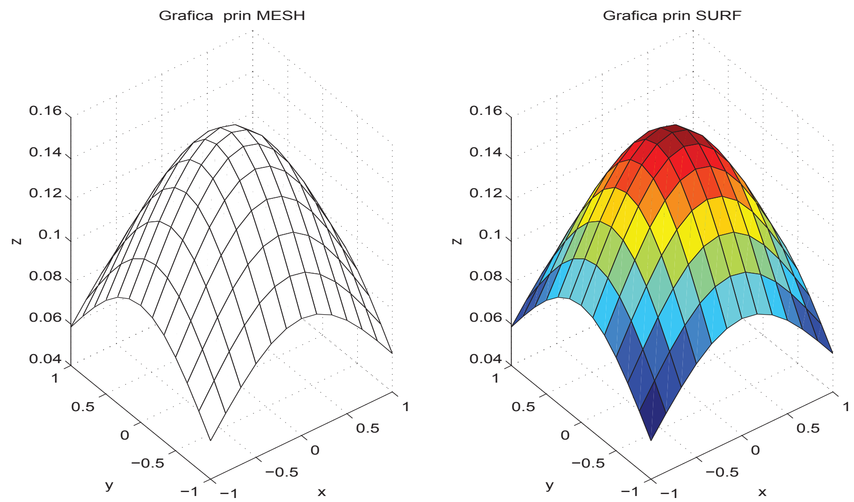

Figura 1.19: $z = \frac{1}{2\pi} \mathrm{e}^{-\frac{1}{2}\left(x^2+y^2\right)}$

```
m=input('m=');
h=1/m; x=-1:h:1;
[X,Y]=meshgrid(x);
Z=1/(2*pi)*exp(-0.5*(X.^2+Y.^2))
subplot(1,3,1), surf(X,Y,Z),
shading faceted
title('Umbrire - faceted')
subplot(1,3,2), surf(X,Y,Z),
shading flat
title('Umbrire - flat')
xlabel('x'), ylabel('y'), zlabel('z')
subplot(1,3,3), surf(X,Y,Z),
shading interp
title('Umbrire - interp')
```

Rezultatul executării programului este în Figura 1.20.

1.12.4 Instrucţiunile `contour` şi `contourf`

Reprezentarea curbelor de nivel ale suprafeţei, dată prin funcţia $z = f(x, y)$, se poate face cu una din următoarele forme ale instrucţiunii `contour`:

```
contour(Z)
contour(Z,n)
contour(Z,v)
contour(X,Y,Z)
```

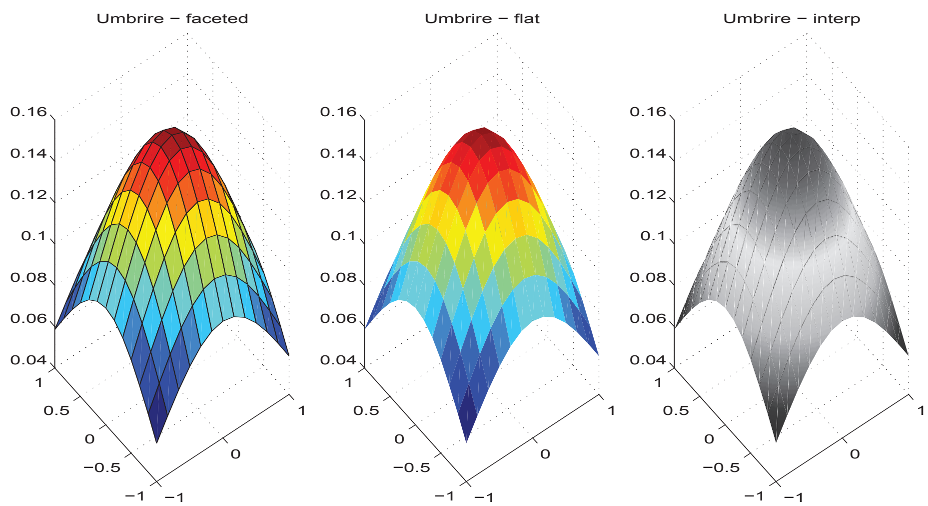

Figura 1.20: Umbrirea unei suprafeţe

```
contour(X,Y,Z,v)
contour(Z,[w w])
contour(X,Y,Z,[w w])
```

unde parametrii `X`, `Y` şi `Z` au aceleaşi interpretări ca şi în cazul instrucţiunilor `mesh` şi `surf`.

Parametrul `v` este un vector prin care se specifică nivelurile curbelor de nivel ce urmează a fi reprezentate grafic, iar în absenţa sa, valorile acestui parametru sunt automat calculate. Numărul curbelor de nivel se poate preciza prin parametrul `n`.

Formele instrucţiunii `contour` cu parametrul de tipul `[w w]` are ca efect trasarea curbei de nivel precizată prin `w`. Aceste forme permit reprezentarea grafică a funcţiilor date sub formă implicită. Astfel, dacă vrem să reprezentăm grafic funcţia $y = y(x)$ dată prin $f(x, y) = 0$, atunci graficul funcţiei $y = y(x)$ va fi dat de curba de nivel pentru funcţia $z = f(x, y)$, considerând `w=0`.

Instrucţiunea `contourf` diferă de `contour` doar prin faptul că ariile delimitate de curbele de nivel sunt umbrite.

Programul 1.12.4. Să reprezentăm `n` curbe de nivel pentru funcţia considerată în Programul 1.12.2. Reprezentările acestor curbe de nivel să fie făcute atât cu instrucţiunea `contour`, cât şi cu `contourf`.

```
clear, clf
m=input('m='); n=input('n=');
h=1/m; x=-1:h:1;
```

```
[X,Y]=meshgrid(x);
Z=1/(2*pi)*exp(-0.5*(X.^2+Y.^2));
subplot(1,2,1), contour(Z,n)
title(['n=',num2str(n),' curbe de nivel'])
subplot(1,2,2), contourf(Z,n)
title(['n=',num2str(n),' curbe de nivel umbrite'])
```

Executarea programului, cu $m = 6$ şi `n=5`, conduce la reprezentările grafice din Figura 1.21.

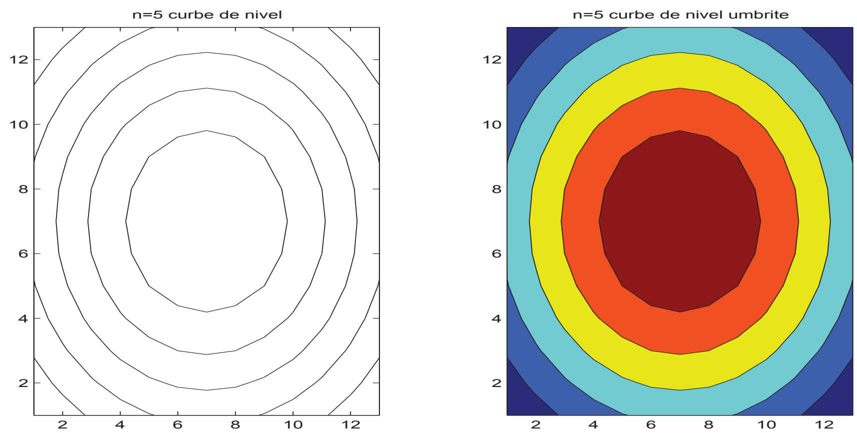

Figura 1.21: Curbe de nivel

1.12.5 Instrucţiunile `ezcontour` şi `ezcontourf`

O formă simplificată, pentru reprezentarea curbelor de nivel, pentru o funcţie dată prin $z = f(x, y)$, se obţine cu ajutorul instrucţiunii `ezcontour`, având una din formele:

```
ezcontour(f)
ezcontour(f,lim)
ezcontour(f,n)
```

unde `f` este expresia matematică a unei funcţii de două variabile, privită ca un şir de caractere.

Parametrul `lim` este fie un vector de forma `[xmin,xmax]`, fie de forma `[xmin,xmax,ymin,ymax]`. Dacă lipseşte, se ia implicit `[-2`π`,2`π`,-2`π`,2`π`]`, iar dacă este de prima formă se ia `[xmin,xmax,xmin,xmax]`.

Parametrul `n` specifică faptul că în reprezentarea grafică se foloseşte o reţea de `n`$\times$`n` puncte (implicit se consideră `n=60`).

De exemplu,

```
f='sqrt(x^2+y^2)'
ezcontour(f,[-2,2])
```

are acelaşi efect cu

```
ezcontour('sqrt(x^2+y^2)',[-2,2])
```

şi produc curbele de nivel pentru funcţia $z = f(x, y) = \sqrt{x^2 + y^2}$, pe domeniul $\boldsymbol{D} = [-2, 2] \times [-2, 2]$.

Instrucţiunea `ezcontourf`, faţă de `ezcontour` execută şi umbrirea ariilor generate de curbele de nivel.

1.12.6 Instrucţiunile `ezmesh`, `ezsurf`, `ezmeshc` şi `ezsurfc`

Sintaxa apelurilor funcţiilor `ezmesh`, `ezsurf`, `ezmeshc` şi `ezsurfc` este aceeaşi, iar efectul executării lor este reprezentarea grafică a unei suprafeţe, respectiv a unei suprafeţe, împreună cu sau fără curbele de nivel corespunzătoare. De aceea prezentăm modurile de apelare, numai pentru una dintre ele.

```
ezmesh('f')
ezmesh('f',lim)
ezmesh('x','y','z')
ezmesh('x','y','z',lim)
```

Parametrul `f` reprezintă o expresie algebrică, ce defineşte suprafaţa, care urmează să fie reprezentată grafic, pe domeniul definit prin `lim`. Implicit, în absenţa parametrului `lim`, domeniul este `[-2`π`,2`π`]`$\times$`[-2`π`,2`π`]`. Dacă `lim=[a,b]`, atunci domeniul va fi `[a,b]`$\times$`[a,b]`, iar dacă `lim=[a,b,c,d]`, atunci domeniul va fi `[a,b]`$\times$`[c,d]`.

Parametrii `x`, `y`, `z`, reprezintă expresii algebrice ce definesc suprafaţa în modul parametric. În acest caz, corespunzător, parametrul `lim`, precizează domeniul în care iau valori parametrii cu ajutorul cărora se defineşte în mod parametric suprafaţa ce urmează a fi reprezentată.

Remarcăm faptul că formele mai sus prezentate, mai pot avea doi parametri. Unul, `n`, de tip întreg pozitiv, care precizează numărul `n`$\times$`n` al punctelor reţelei. Implicit se consideră `n=60`. Un altul este `'circ'`, care prin prezenţa sa atrage reprezentarea lui `f` pe un disc centrat în domeniul mai sus precizat.

Pentru exemplificare, dacă se execută programul Matlab format din următoarele instrucţiuni:

```
subplot(1,2,1)
ezmesh('sin(x)*sin(y)',[-pi,pi],'circ')
subplot(1,2,2)
ezmeshc('sin(x)*sin(y)',[-pi,pi],'circ')
colormap spring
```

se obţin graficele din Figura 1.22.

Mai remarcăm faptul că `f`, `x`, `y`, `z` pot fi numele unor funcţii predefinite sau scrise de utilizator.

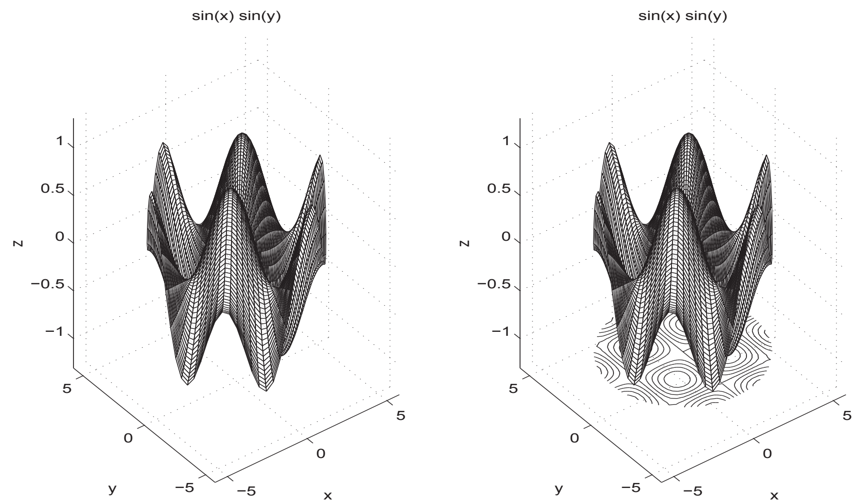

Figura 1.22: Suprafaţă fără şi cu curbe de nivel

Pentru a ilustra că cele două instrucţiuni de tipul `ezmesh` şi `ezsurf` au de regulă acelaşi efect, programul următor reprezintă grafic aceeaşi suprafaţă, respectiv cu `ezmesh` şi `ezsurf`.

```
clf
subplot(1,2,1),
ezsurf('x*exp(-x^2 - y^2)',[-2,2],20)
title('Reprezentare cu ezsurf')
subplot(1,2,2)
ezsurf('x*exp(-x^2 - y^2)',[-2,2],20)
colormap autumn
title('Reprezentare cu ezmesh')
```

Graficele sunt prezentate în Figura 1.23.

1.12.7 Instrucţiunile `bar3` şi `bar3h`

Instrucţiunile `bar3` şi `bar3h` corespund instrucţiunilor `bar` şi respectiv `barh` din grafica bidimensională.

Forme ale acestor instrucţiuni sunt:

```
bar3(z,w,'tip','color')
bar3(y,z,w,'tip','color')
bar3h(y,w,'tip','color')
bar3h(y,z,w,'tip','color')
```

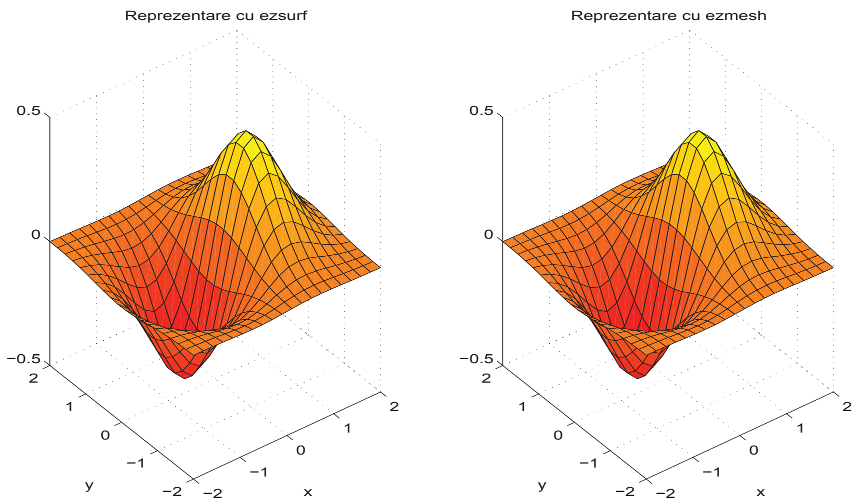

Figura 1.23: Suprafaţă reprezentată $z = f(x, y) = x\mathrm{e}^{-\left(x^2+y^2\right)}$

argumentele `w`, `'tip'` şi `'color'` sunt opţionale, dar când sunt specificate, trebuie să păstreze această ordine.

Parametrul `y` este un vector de lungime `m`, având componentele ordonate crescător sau descrescător. Dacă `y` lipseşte, atunci se consideră implicit `y=1:m`.

Dacă `z` este un vector de aceeaşi lungime cu `y`, atunci vor fi reprezentate grafic `m` bare de lungimi $\mathtt{z}_i$ în dreptul punctelor $\mathtt{y}_i$, $i = \overline{1, \mathtt{m}}$, de pe axa `oy`. Dacă `z` este o matrice de tipul `(m,n)`, reprezentările sunt efectuate pentru fiecare din cele `n` coloane ale matricei, adică în fiecare punct $\mathtt{y}_i$, $i = \overline{1, \mathtt{m}}$, de pe axa `oy`, vor fi reprezentate câte `n` bare (batoane).

Parametrul `w` specifică grosimea barelor, implicit fiind $\mathtt{w} = 0.8$, iar pentru $\mathtt{w} > 1$ barele se suprapun.

Dacă parametrul `tip=detached`, care este valoarea implicită, atunci barele din punctele $\mathtt{y}_i$ de pe axa `oy` vor fi detaşate în adâncime după axa `ox`. Dacă `tip=grouped`, barele (batoanele) vor fi grupate în punctele $\mathtt{y}_i$, de pe axa `oy`, iar dacă `tip=stacked`, atunci barele vor fi stivuite în aceste puncte.

Parametrul `color` specifică culoarea barelor (batoanelor) şi are una din următoarele valori: `r` (roşu), `g` (verde), `b` (albastru), `y` (galben), `m` (violet), `c` (ciclamen), `k` (negru), `w` (alb).

Deosebirea între `bar3` şi `bar3h` este că prima instrucţiune acţionează pe verticală, iar a doua pe orizontală.

Programul 1.12.5. Să scriem un program care generează o matrice magică de ordin $m > 2$. Folosind primele trei coloane ale matricei magice, să reprezentăm prin bare verticale valorile acestora cu fiecare din tipurile `detached`, `grouped` şi `stacked`. Folosind ultimele trei coloane să se facă astfel de reprezentări cu ajutorul barelor orizontale.

```
m=input('m:'); A=magic(m);
subplot(2,3,1), bar3(A(:,1:3),.6,'detached')
title('Bar3 - detached')
subplot(2,3,2), bar3(A(:,1:3),'grouped')
title('Bar3 - grouped')
subplot(2,3,3), bar3(A(:,m-2:m),.6,'stacked')
title('Bar3 - stacked')
subplot(2,3,4), bar3h(A(:,m-2:m),.6,'detached')
title('Bar3h - detached')
subplot(2,3,5), bar3h(A(:,m-2:m),'grouped')
title('Bar3h - grouped')
subplot(2,3,6), bar3h(A(:,1:3),.6,'stacked')
title('Bar3h - stacked')
colormap spring
```

Executarea acestui program, pentru `m=4`, produce graficele din Figura 1.24.

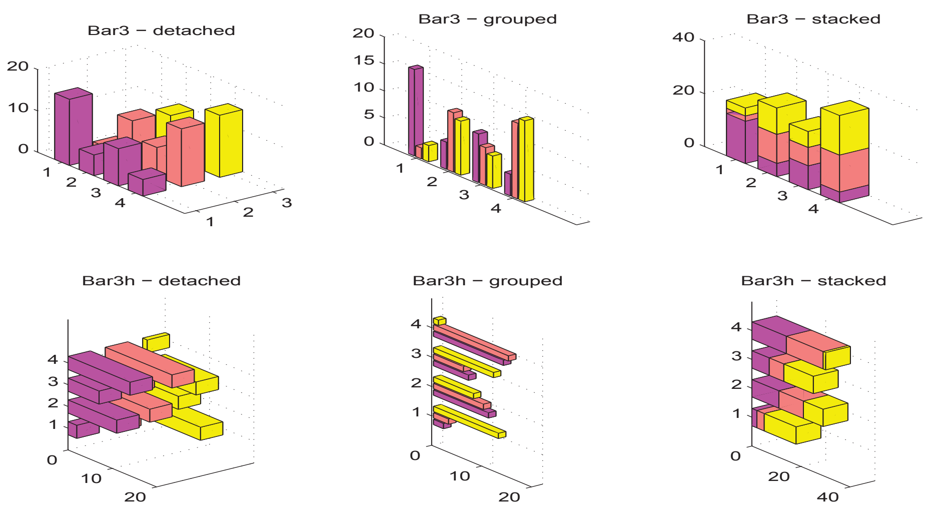

Figura 1.24: Bare verticale şi bare orizontale în 3D