Tema Lab. 9

NICOLETA RADU

# Exercitiul 1

Reprezentaţi grafic, prin 2 metode, în 2 ferestre diferite, curba dată de ecuaţiile: sin cos , t [0,10 ] x t y t z t π ⎧ = ⎪ ⎨ = ∈ ⎪ ⎩ = Indicație:Met1) plot3; Met 2) ezplot3

t = 0:0.1:10\*pi;

x = sin(t);

y = cos(t);

z = t;

hold on

subplot(1,2,1); plot3(x,y,z);

subplot(1,2,2); ezplot3('sin(t)','cos(t)','t');

hold off

//

t = 0:0.1:10\*pi;

x = sin(t);

y = cos(t);

z = t;

hold on

figure(1)

plot3(x,y,z);

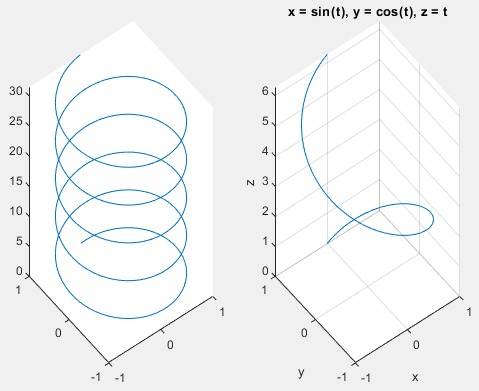
view(3)

figure(2)

ezplot3('sin(t)','cos(t)','t');

view(3);

hold off



# Exercitiul 2

Reprezentaţi grafic curba dată de ecuaţiile: \*cos \*sin , r [0,1], t [0,2 ] x r t y r t z t π ⎧ = ⎪ ⎨ = ∈ ∈ ⎪ ⎩ = Vectorii generați r, t au aceeași dimensiune. Trasați un grid și etichetați toate axele. Adăugați un titlu graficului. Folosind butonul Rotate 3D rotiți axele pentru a observa reprezentarea curbei pe axele XYZ.

r = linspace(0,1,20);

t = linspace(0,2\*pi,20);

x = r .\* cos(t);

y = r .\* sin(t);

z = t;

plot3(r .\* cos(t),r .\* sin(t),z); title('titlu :)');

grid on;

// sau

t = linspace(0,2\*pi,100);

r = linspace(0,1,100);

x = r .\* cos(t);

y = r .\* sin(t);

z = t;

hold on

plot3(x,y,z); title('Al meu titlu!');

view(3)

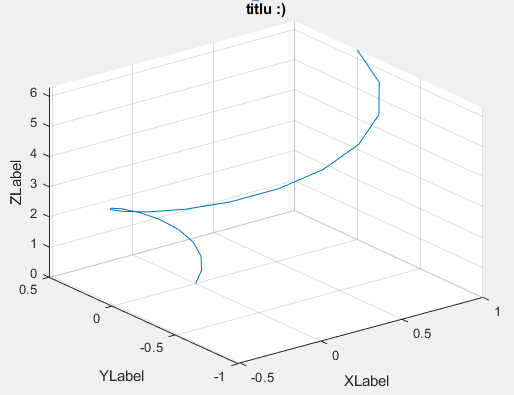
xlabel('xlabel');

ylabel('ylabel');

zlabel('zlabel');

hold off

grid on



# Exericitiul 3

Scrieţi un fișier script pentru reprezentarea graficului funcţiei 1 2 2 ( ) 2 1 ( , ) 2 x y f x y e π − + = , pe domeniul D = − × − [ 1,1] [ 1,1] , în două subgrafice alăturate, folosind funcţiile mesh şi surf. În fiecare domeniu avem 30 de puncte. Se cere construirea unui fișier funcție separat pentru definirea lui f(x,y). Indicație: Domeniul D = − × − [ 1,1] [ 1,1] specifică domeniul de definire a lui x și y, adică -1 ≤ x ≤ 1 și -1 ≤ y ≤ 1. De exemplu, x=linspace(-1,1,30)

[x y] = meshgrid(linspace(-1,1,30))

z=(1/(2\*pi))\*exp((-1/2)\*(x^2+y^2))

subplot(1,2,1)

mesh(x,y,z)

subplot(1,2,2)

surf(x,y,z)

//

y = linspace(-1,1,30);

x = linspace(-1,1,30);

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

z = f3(X,Y);

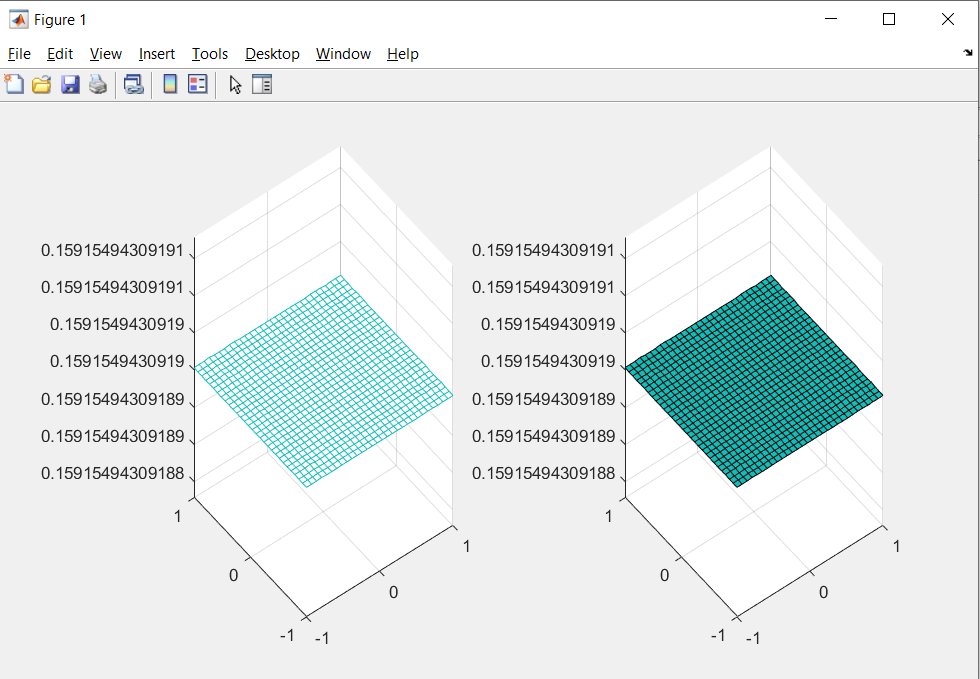
hold on

subplot(1,2,1); mesh(X,Y,z);

subplot(1,2,2); surf(X,Y,z);

view(3)

hold off



# Exercitiul 4

Scrieţi un fișier script care efectuează umbrirea suprafeţei de la problema 3, folosind toate cele 3 tipuri de umbrire. Fiecare grafic va fi desenat în ferestre diferite. Se cere închiderea pe rând a ferestrelor grafice la un interval de 5 secunde. (shading, close)

y = linspace(-1,1,30);

x = linspace(-1,1,30);

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

z = f3(X,Y);

figure(1)

surf(X,Y,z)

shading faceted

pause(5)

close

figure(2)

surf(X,Y,z)

shading flat

pause(5)

close

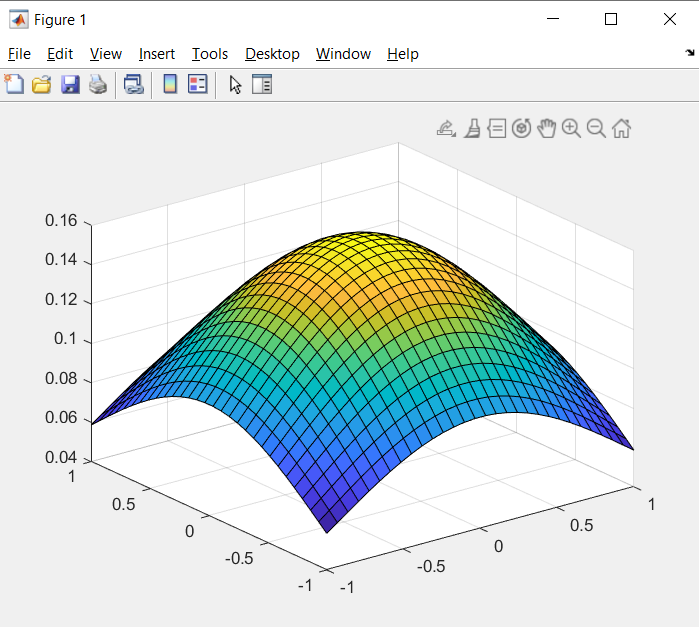
figure(3)

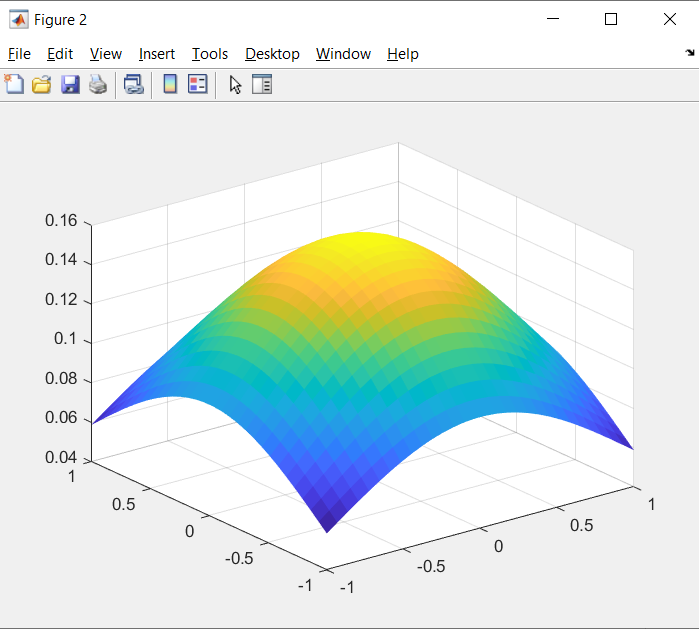
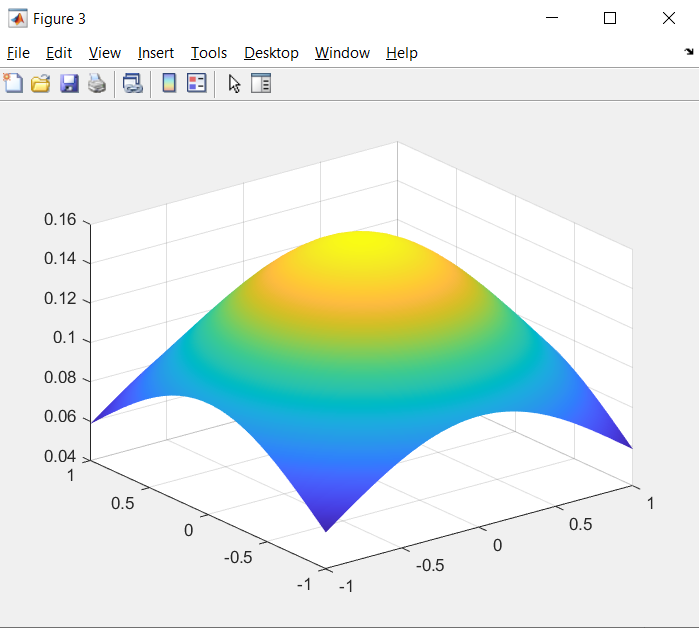
surf(X,Y,z)

shading interp

pause(5)

close

pause(5) 

close() 

# Exercitiul 5

Scrieţi un fișier script pentru reprezentarea graficului funcţiei z f x y = ( , ) 2 2 z x y = − − − ( 3) ( 3) , pe domeniul D = × [2,4] [1,3] , în două subgrafie diferite și alăturate, folosind funcţiile mesh şi surf. În fiecare domeniu avem 40 de puncte.

[x y] = meshgrid(linspace(2,4,40),linspace(1,3,40))

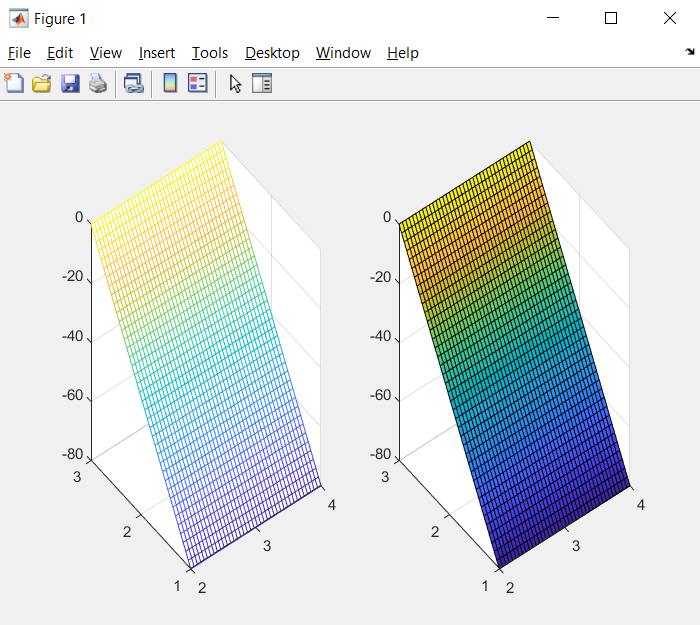
z = (x-3)^2-(y-3)^2

subplot(1,2,1)

mesh(x,y,z)

subplot(1,2,2)

surf(x,y,z)



# Exercitiul 6

6. Aceleași cerințe ca la pb. 5 doar că se vor folosi funcțiile ezmeshc şi ezsurfc.

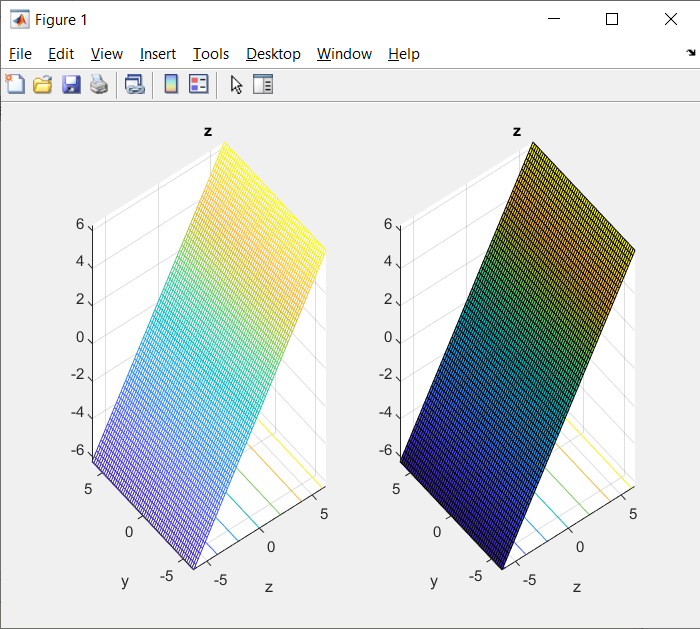
[x y] = meshgrid(linspace(2,4,40),linspace(1,3,40))

z = (x-3)^2-(y-3)^2

subplot(1,2,1)

ezmeshc('z')

subplot(1,2,2)

ezsurfc('z') 

# Exercitiul 7

r=input('Dati o valoare pentru r: ')

[u v] = meshgrid(0:2\*pi)

x=(r+cos(u/2)\*sin(v)-sin(u/2)\*sin(2\*v))\*cos(u)

y=(r+cos(u/2)\*sin(v)-sin(u/2)\*sin(2\*v))\*sin(u)

z=sin(u/2)\*sin(v)+cos(u/2)\*sin(2\*v)

surf(x,y,z)

//

r = input('dati r: ')

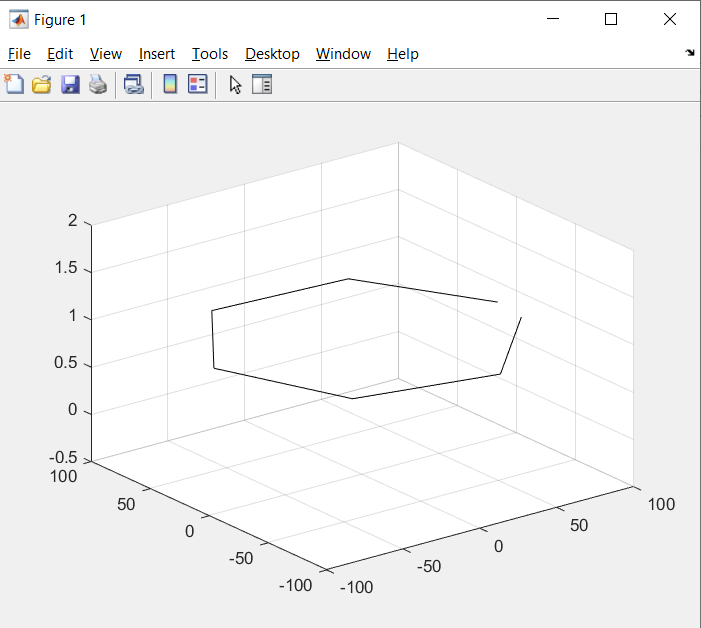
[u,v]= meshgrid(linspace(0,2\*pi,20));

x = (r + cos(u./2) .\* sin(v) - sin(u./2) .\* sin(2.\*v)) .\* cos(u);

y = (r + cos(u./2) .\* sin(v) - sin(u./2) .\* sin(2.\*v)) .\* sin(u);

z = sin(u./2) .\* sin(v) + cos(u./2) .\* sin(2.\*v);

surf(x,y,z);



# Exercitiul 8

[x y]=meshgrid(linspace(-2,2,30))

z=-x\*y\*exp(-2\*(x^2+y^2))

surf(x,y,z)

figure

k\_max = find(z==max(max(z)));

plot3(x(k\_max),y(k\_max),z(k\_max),'\*r');

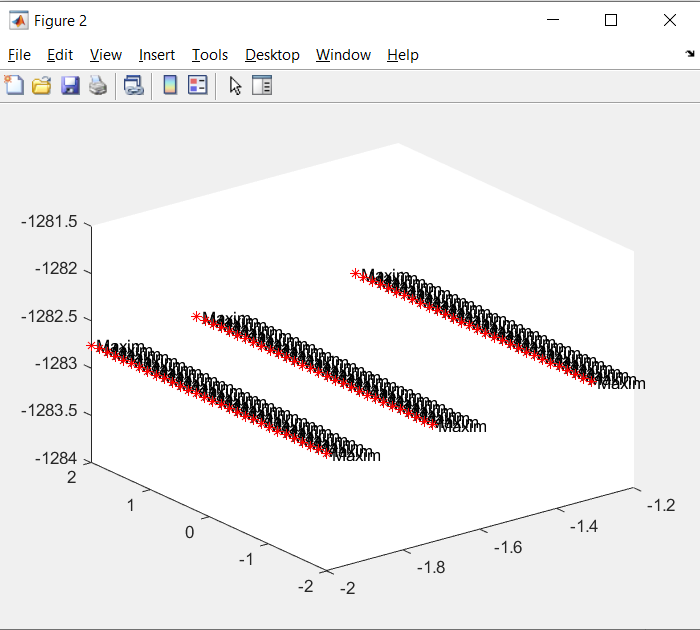
text(x(k\_max),y(k\_max),z(k\_max),' Maxim');

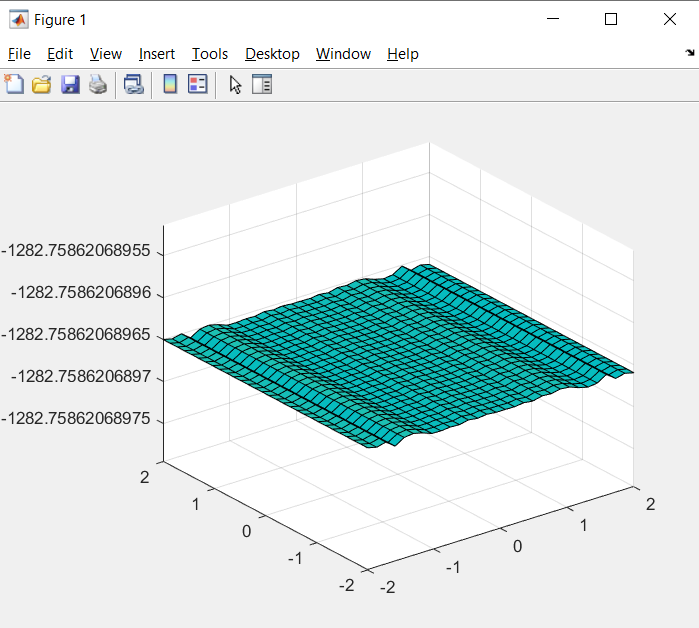
figure

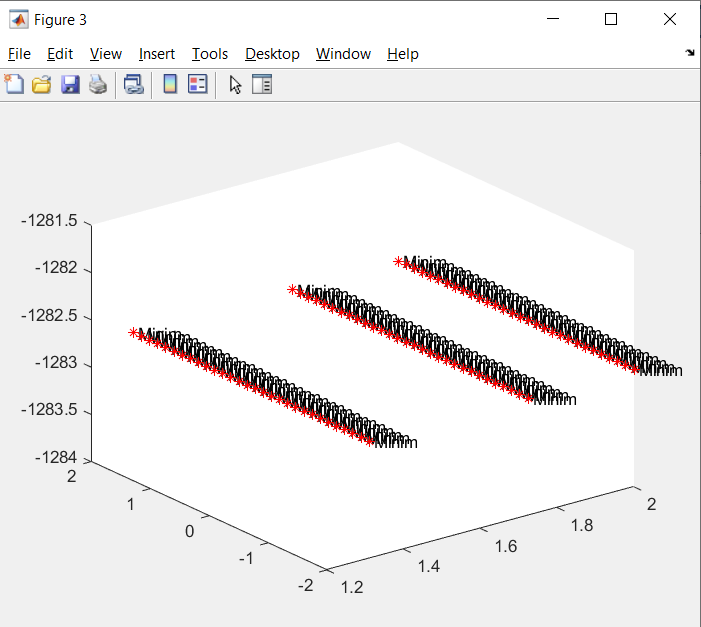
k\_min = find(z==min(min(z)));

plot3(x(k\_min),y(k\_min),z(k\_min),'\*r');

text(x(k\_min),y(k\_min),z(k\_min),' Minim');







# Exercitiul 9

[x y]=meshgrid(linspace(-pi,pi,20))

z=sin(x)\*sin(y)

subplot(2,1,1)

ezmeshc('z')

subplot(2,1,2)

contourf(x,y,z)

//

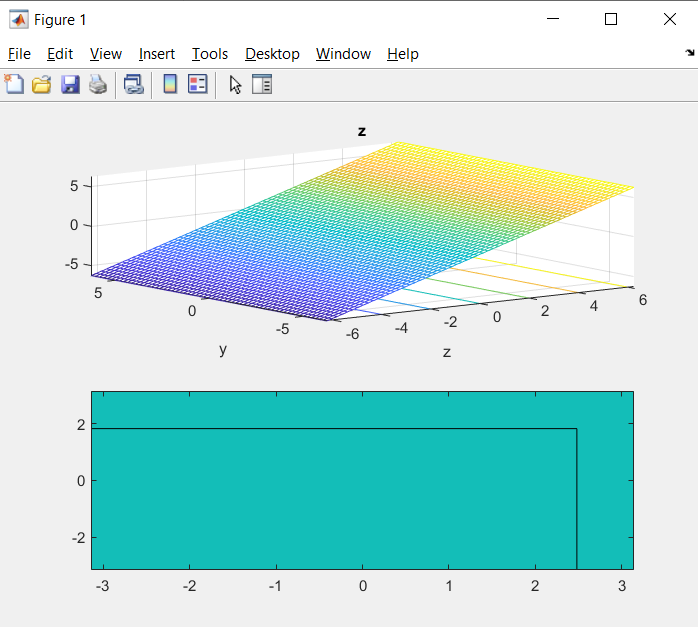
[x,y] = meshgrid(linspace(-pi,pi,30));

z = sin(x) .\* sin(y);

subplot(2,1,1); ezsurfc('z');

subplot(2,1,2); ezsurf('z');

view(3)



# Exercitiul 10

R=[10 3];

N=20;

z = 14;

[x,y,z]=cylinder(R,N);

surf(x,y,z,'FaceColor','white');

# Exercitiul 11

R=[10 3];

N=4;

z = 3;

[x,y,z]=cylinder(R,N);

surf(x,y,z,'FaceColor','green');

# Exercitiul 12

R = [7 6 5 4 3 2 1];

N = 4;

z = 3;

[x,y,z]=cylinder(R,N);

colormap("spring");

surf(x,y,z,'FaceColor','texturemap');

# Exercitiul 13

function [] = functia\_13(raza\_baza,raza\_varf,nr\_nivele,nr\_laturi)

%UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

if raza\_baza < raza\_varf

aux = raza\_baza;

raza\_baza = raza\_varf;

raza\_varf = aux;

end

corrective = raza\_baza / nr\_nivele;

R = [ raza\_baza, linspace(raza\_baza - 1, raza\_varf + corrective, nr\_nivele), raza\_varf];

N = nr\_laturi;

[x,y,z] = cylinder(R,N);

surf(x,y,z);

end

# Exercitiul 14

% punctul a

x1 = linspace(1,2,20);

y1 = linspace(-6,6,20);

[X1,Y1] = meshgrid(x1,y1);

Z1 = 20 ./ ((X1.^2 + Y1.^2).^3);

% punctul b

x2 = linspace(-10,10,20);

y2 = linspace(-6,6,20);

[X2,Y2] = meshgrid(x2,y2);

Z2 = X2 .\* Y2 ./ (sqrt((1 + X2.^2 + Y2.^ 2).^3));

% punctul c

[X3,Y3] = meshgrid(linspace(0,pi,20));

Z3 = Y3 .\* sin(3) .\* X3;

% punctul d

[X4,Y4] = meshgrid(linspace(1,4,20));

Z4 = (1 - X4) .\* Y4.^2;

hold on

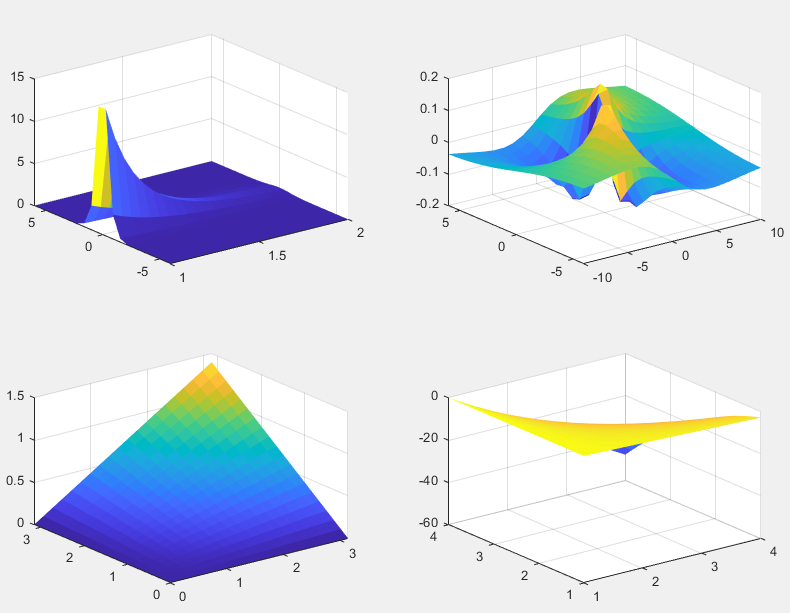
subplot(2,2,1); surf(X1,Y1,Z1); shading flat

subplot(2,2,2); surf(X2,Y2,Z2); shading flat

subplot(2,2,3); surf(X3,Y3,Z3); shading flat

subplot(2,2,4); surf(X4,Y4,Z4); shading flat

hold off



# Exercitiul 15

% t = linspace(0,12\*pi,nrPuncte);

% a = 10;

% b = 3;

% omega = 5;

% nrPuncte = 20;

% plot3(a \* cos(omega \* t),a \* sin(omega \* t),b \* t)

// si de la tastatura

omega = input('dati omegaLUL: ')

nrPuncte = input('dati numarul de puncte: ')

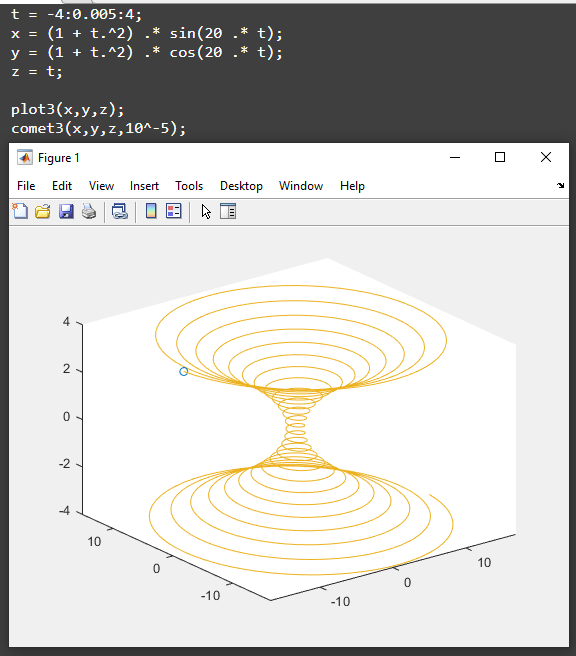
t = linspace(0,12\*pi,nrPuncte);

a = linspace(0,20,nrPuncte);

b = linspace(0,10,nrPuncte);

plot3(a .\* cos(omega .\* t),a .\* sin(omega .\* t),b .\* t)

# Exercitiul 16



t = -4:0.005:4;

x = (1 + t.^2) .\* sin(20 .\* t);

y = (1 + t.^2) .\* cos(20 .\* t);

z = t;

plot3(x,y,z);

comet3(x,y,z,10^-5);

# Exercitiul 17

[x,y] = meshgrid(linspace(0,pi,80));

z = sin(y.^2 + x) - cos(y - x.^2);

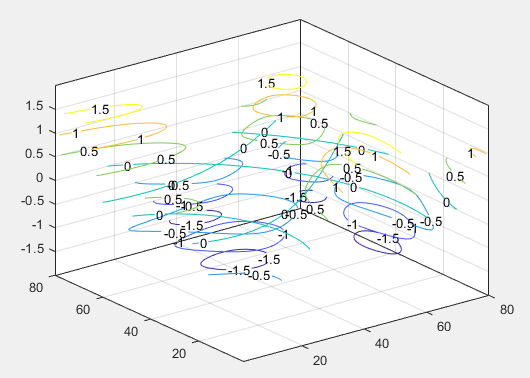
figure(1)

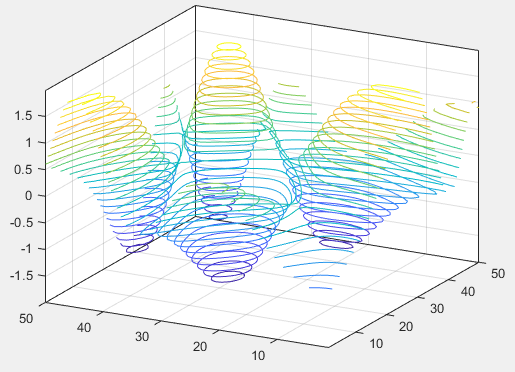
mesh(z);

cs = contour3(z,50); clabel(cs);

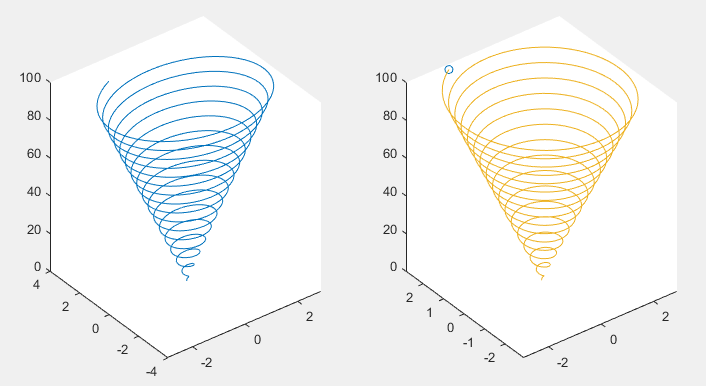
[cs,h] = contour3(z);

clabel(cs,h);





# Exercitiul 18



t = linspace(0,100,1000);

c = 0.03;

r = c .\* t;

x = r .\* sin(t);

y = r .\* cos(t);

z = t;

subplot(1,2,1); plot3(x,y,z);

subplot(1,2,2); comet3(x,y,z,0.002);

# Exercitiul 19

function [outputArg1,outputArg2] = createSpheres(numOfSpheres,numOfVertices)

%UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

dimSubplot = ceil(sqrt(numOfSpheres))

for ind = 1:numOfSpheres

[x,y,z] = sphere(numOfVertices);

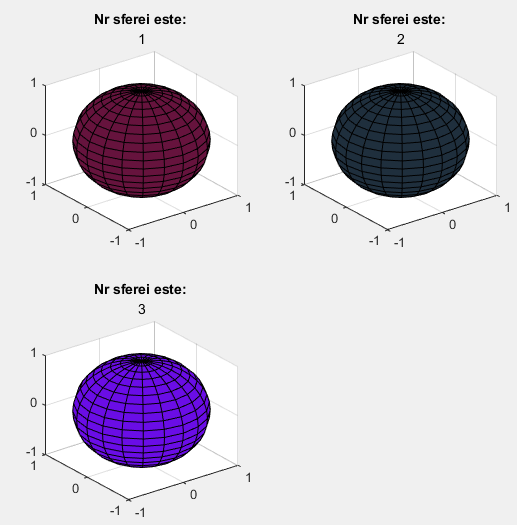
subplot(dimSubplot,dimSubplot,ind);

surf(x,y,z,'FaceColor',[rand(1),rand(1),rand(1)]);

title('Nr sferei este: ',num2str(ind));

end

createSpheres(3,20)



# Exercitiul 20

r=input('Dati o valoare pentru r: ')

R=input('Dati o valoare pentru R: ')

u=linspace(0,2\*pi)

v=linspace(-2,2)

[u v] = meshgrid(u,v)

x1=cos(u)

y1=sin(u)

z1=v

s=linspace(0,2\*pi)

t=linspace(0,pi)

[s t] = meshgrid(s,t)

x2=R.\*sin(t).\*cos(s)

y2=r.\*sin(t).\*sin(s)

z2=cos(t)

hold on

surf(x2,y2,z2)

mesh(x1,y1,z1)

hold off

