

3-D graphics

Matlab provides extensive facilities for visualization of three-dimensional data. The most common are plots of curves in a three-dimensional space, mesh plots and surface plots. The command **plot3(x,y,z,'style option')** produces a curve in the three-dimensional space. The **title**, **xlabel**, **ylabel**, etc., may be used for three-dimensional plots. The **mesh** and **surf** commands have several optional arguments and are used for plotting meshes and surfaces.

Following is a list of some specialized 3-D graphs.

plot3 Plot lines and points in 3-D space

mesh 3-D mesh surface

surf 3-D colored surface

ezmesh Easy to use 3-D mesh plotter

ezplot3 Easy to use 3-D parametric curve plotter

ezsurf Easy to use 3-D colored surface plotter

cylinder Generate cylinder

sphere Generate sphere

plot3: Plot lines and points in 3-D space

Three-dimensional plots completely analogous to plot in two dimensions, the command **plot3** produces curves in three-dimensional space. If x , y , and z are three vectors of the same size, then the command **plot3(x,y,z)** will produce a perspective plot of the piecewise linear curve in 3-space passing through the points whose coordinates are the respective elements of x , y , and z . These vectors are usually defined parametrically.

For example, the following equations specify a curve in 3 dimensions:

$$x = e^{-0.05t} \sin(t)$$

$$y = e^{-0.05t} \cos(t)$$

$$z = t$$

If t ranges from 0 to 10 to plot type:

```
t = [0:pi/50:10*pi];
```

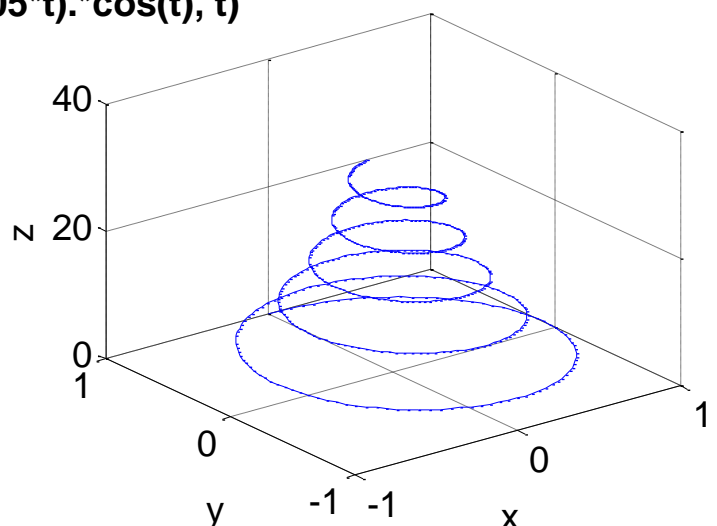
```
plot3(exp(-0.05*t).*sin(t), exp(-0.05*t).*cos(t), t)
```

```
xlabel('x')
```

```
ylabel('y')
```

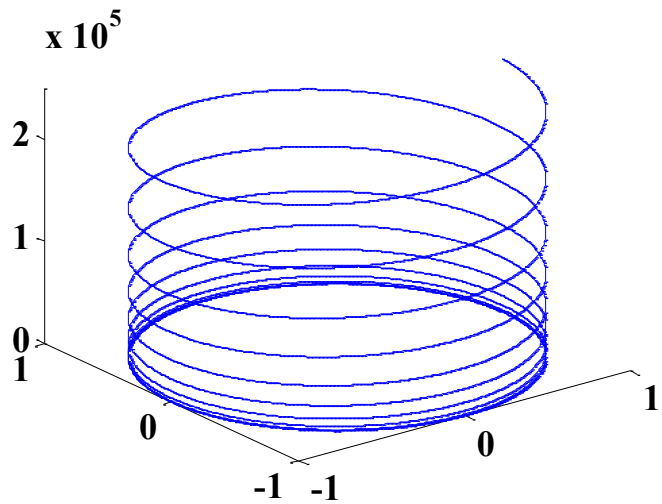
```
zlabel('z')
```

```
grid on
```



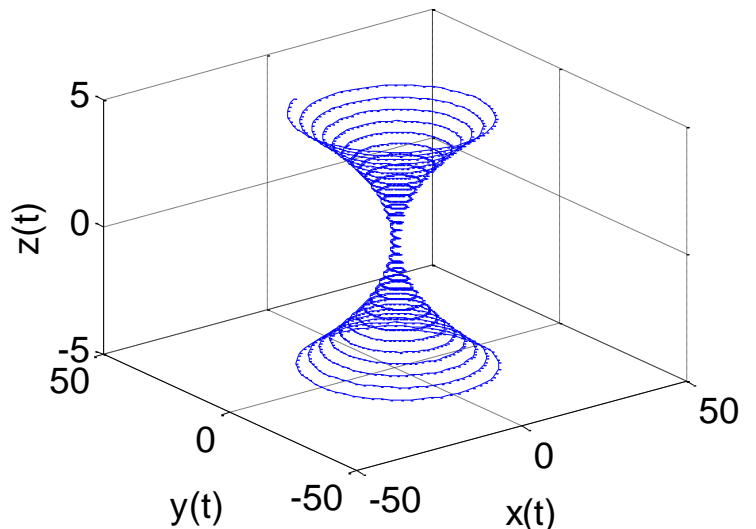
Example:

```
t=.01:.01:20*pi;  
x=cos(t);  
y=sin(t);  
z=t.^3;  
plot3(x,y,z)
```



Example:

```
t=-5:.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)')
```



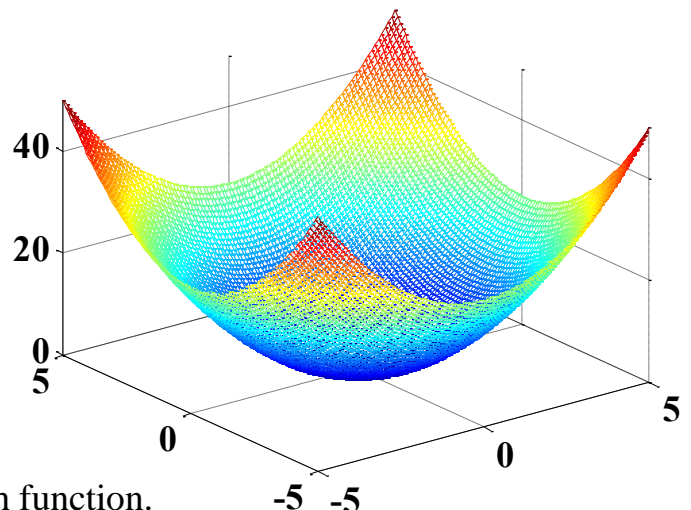
Creating 3-D Graphs

MATLAB defines a surface by the z -coordinates of points above a rectangular grid in the x - y plane. The plot is formed by joining adjacent points with straight lines. Surface plots are useful for visualizing matrices that are too large to display in numerical form and for graphing functions of two variables.

mesh: A mesh surface is defined by the z coordinates of points above a rectangular grid in the x - y plane. The plot is formed by joining adjacent points with straight lines.

Example: Generate the matrix table of the function $f=x.^2+y.^2$; where $0 \leq x \leq 5$, $0 \leq y \leq 5$

```
xv=[-5:.1:5];  
yv=[-5:.1:5];  
nx=length(xv);  
ny=length(yv);  
for i=1:nx  
for j=1:ny  
zmat(i,j)=(xv(j).^2+yv(i).^2);  
end  
end
```



To display the function, we can use mesh function.

```
mesh(xv,yv,zmat);
```

meshgrid

To generate the matrix table, we needed to perform a double nested loop which is time consuming. MATLAB has it's own method of generating matrix tables using the meshgrid function.

To see what meshgrid does

```
x=[6:1:10];  
y=[1:1:5];  
[X,Y]=meshgrid(x,y);  
X =
```

```
 6  7  8  9 10  
 6  7  8  9 10  
 6  7  8  9 10  
 6  7  8  9 10  
 6  7  8  9 10
```

```
Y =
```

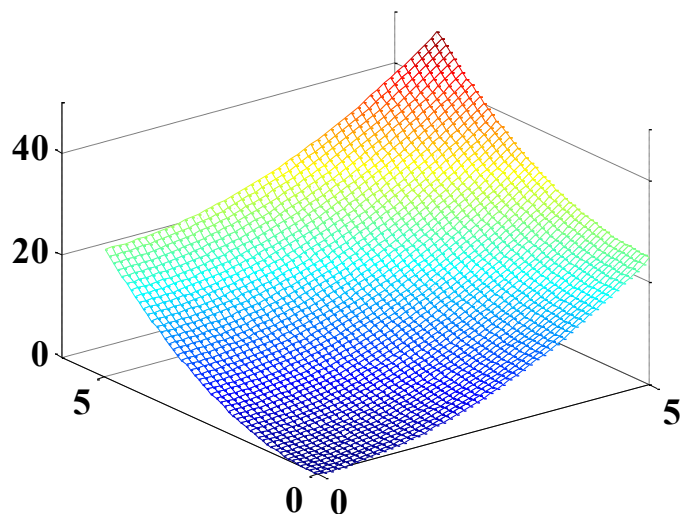
```
 1  1  1  1  1  
 2  2  2  2  2  
 3  3  3  3  3  
 4  4  4  4  4  
 5  5  5  5  5
```

X is a matrix in which the x domain [6 7 8 9 10] is copied in every row and Y is a matrix in which the y domain [1 2 3 4 5] is copied in every column.

Now since $Z(i,j)=Z(x(j),y(i))$ by definition and since $X(i,j)=x(j)$ for any i and $Y(i,j)=y(i)$ for any j, $Z(i,j)=Z(X(i,j),Y(i,j))$ which is the definition of a matrix array operation.

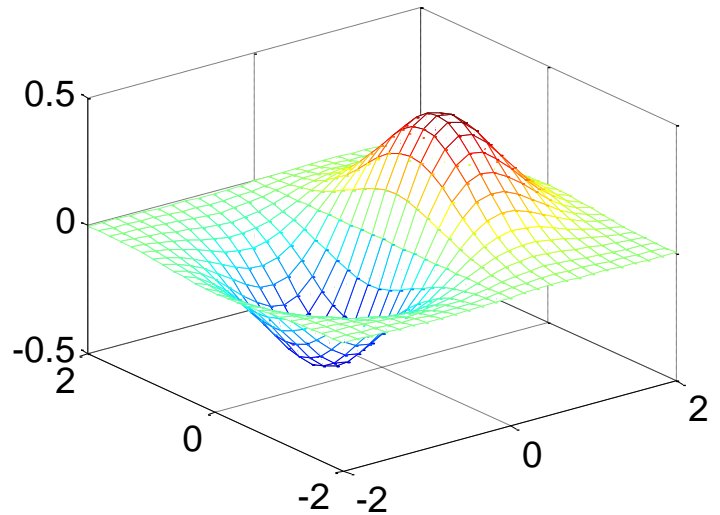
For example, **meshgrid** can be used to generate X- and Y-grid matrices for use when computing the displayed Z matrix.

```
x=[0:.1:5];  
y=[0:.1:5];  
[X,Y]=meshgrid(x,y);  
Z=(X.^2)+(Y.^2);  
mesh(x,y,Z)
```



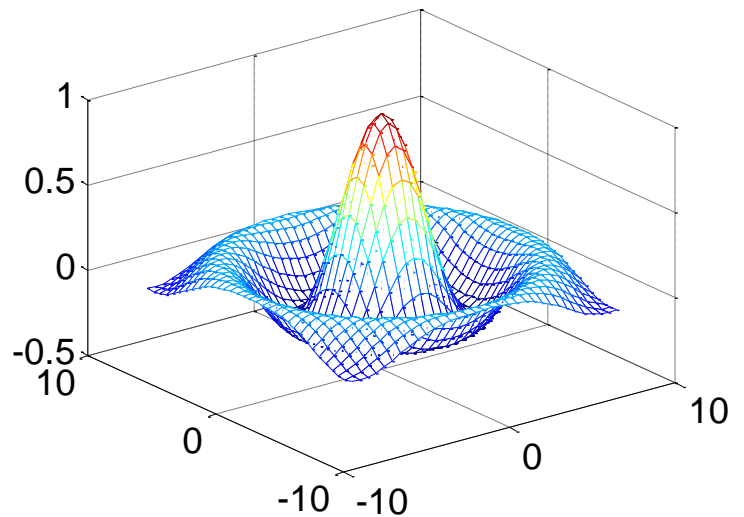
Example:

```
x=linspace(-2, 2, 25);  
y=linspace(-2, 2, 25);  
[xx,yy]=meshgrid(x, y);  
zz=xx.*exp(-xx.^2-yy.^2);  
mesh(xx, yy, zz);
```



Example: Graphing the sinc function.

```
[X,Y] = meshgrid(-8:.5:8);  
R = sqrt(X.^2 + Y.^2) + eps;  
Z = sin(R)./R;  
mesh(X,Y,Z)
```

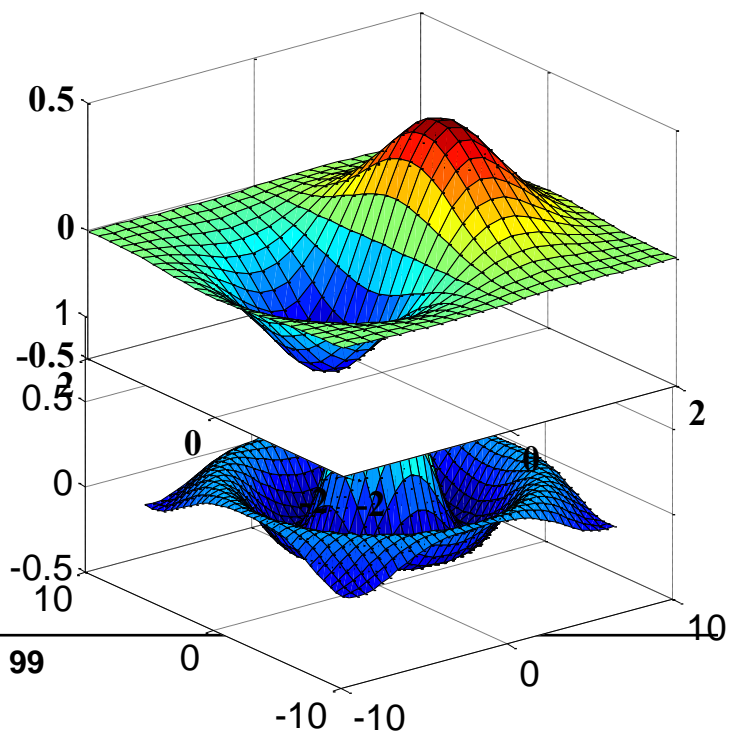


surf: produces a solid filled surface plot.

A surface plot is similar to a mesh plot except the rectangular faces of the surface are colored.

Example:

```
x=linspace(-2, 2, 25);  
y=linspace(-2, 2, 25);  
[xx,yy]=meshgrid(x, y);  
zz=xx.*exp(-xx.^2-yy.^2);  
surf(xx, yy, zz);
```

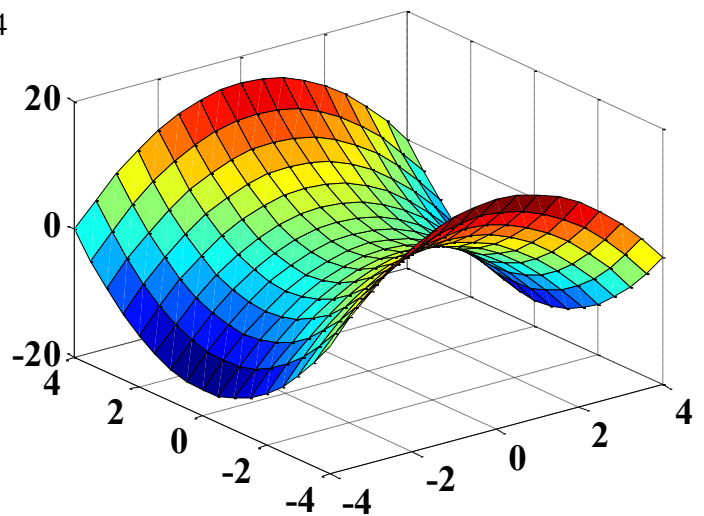


Example:

```
[X,Y] = meshgrid(-8:.5:8);  
R = sqrt(X.^2 + Y.^2) + eps;  
Z = sin(R)./R;  
surf(X,Y,Z)
```

Example: $z = y^2 - x^2, -4 \leq x \leq 4, -4 \leq y \leq 4$

```
[xx yy]=meshgrid(-4:0.5:4,-4:0.5:4);  
z=yy.^2-xx.^2;  
surf(xx,yy,z)
```

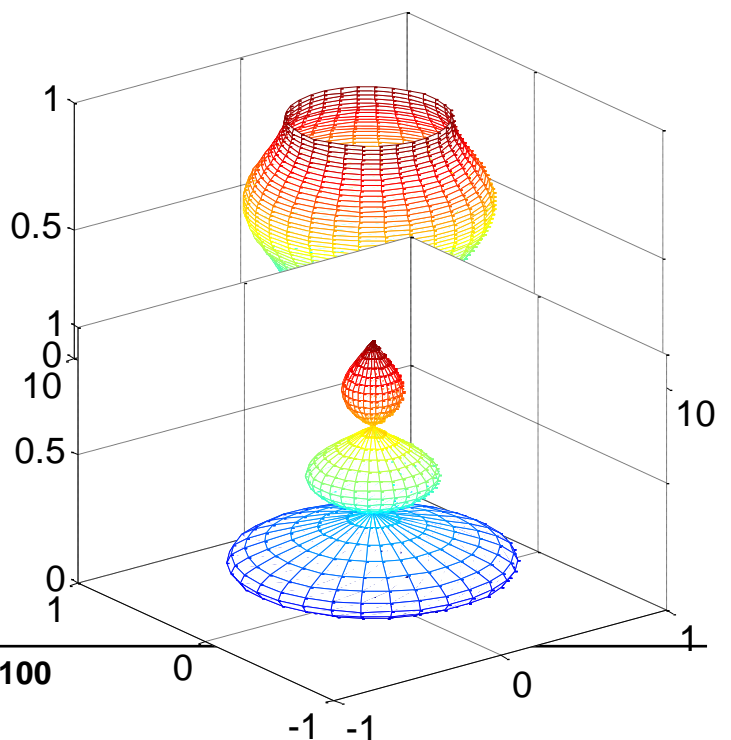


cylinder Generate cylinder.

[X,Y,Z] = cylinder (R,N) forms the unit cylinder based on the generator curve in the vector R. Vector R contains the radius at equally spaced points along the unit height of the cylinder. The cylinder has N points around the circumference.

Example:

```
x=0:pi/20:pi*3;  
r=5+cos(x);  
[a,b,c]=cylinder(r,30);  
mesh(a,b,c)
```

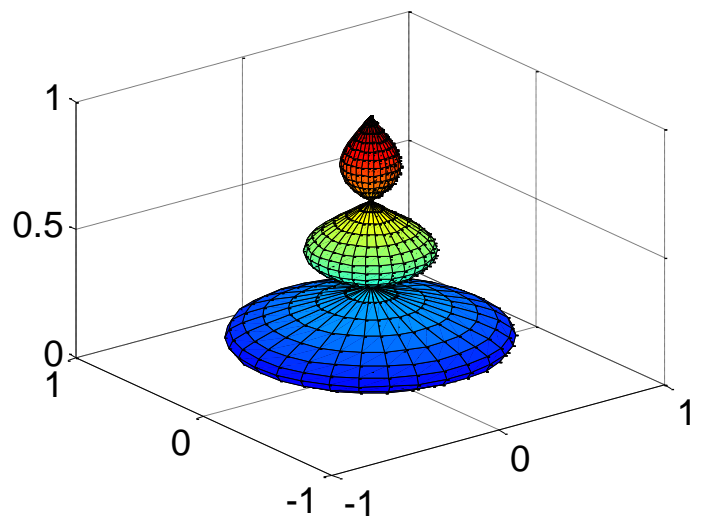


Example:

```
t=0:pi/12:3*pi;  
r=abs(exp(-0.25*t).*sin(t));  
[X,Y,Z]=cylinder(r,30);  
mesh(X,Y,Z)
```

Example:

```
t=0:pi/12:3*pi;  
r=abs(exp(-0.25*t).*sin(t));  
[X,Y,Z]=cylinder(r,30);  
surf(X,Y,Z)
```

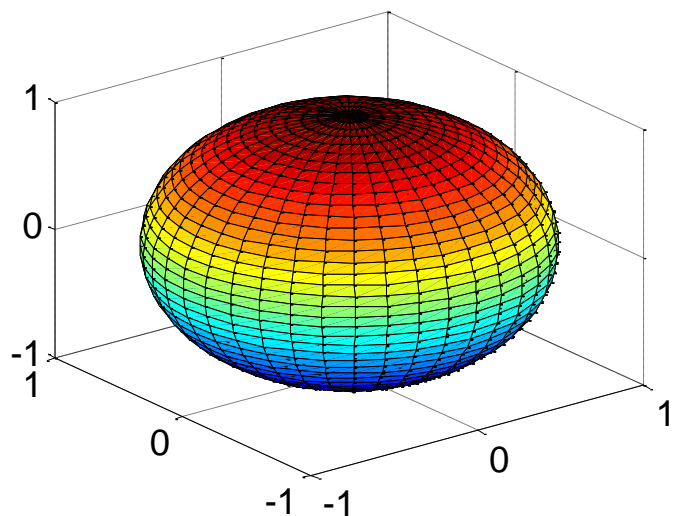


sphere Generate sphere.

$[X,Y,Z] = \text{sphere}(N)$ generates three $(N+1)$ by $(N+1)$ matrices so that $\text{surf}(X,Y,Z)$ produces a unit sphere.

Example:

```
[a,b,c]=sphere(40);  
surf(a,b,c);
```



ezplot3 Easy to use 3-d parametric curve plotter.

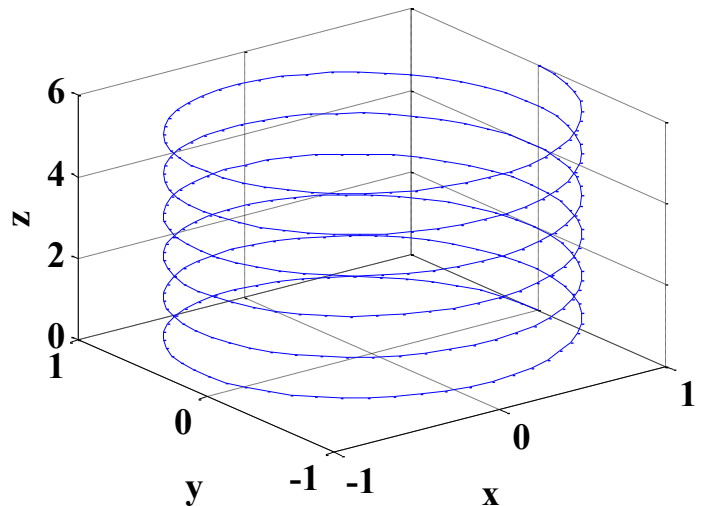
ezplot3 (FunX, FunY, FunZ) plots the spatial curve FunX(T), FunY(T), and FunZ(T) over the default domain $0 < T < 2\pi$.

ezplot3 (FunX, FunY, FunZ, [Tmin, Tmax]) plots the curve FunX(T), FunY(T), and FunZ(T) over $T_{min} < T < T_{max}$.

Example:

ezplot3('cos(2*pi*t)', 'sin(2*pi*t)', 't', [0, 6])

$$x = \cos(2\pi t), y = \sin(2\pi t), z = t$$



ezsurf Easy to use 3-D colored surface plotter.

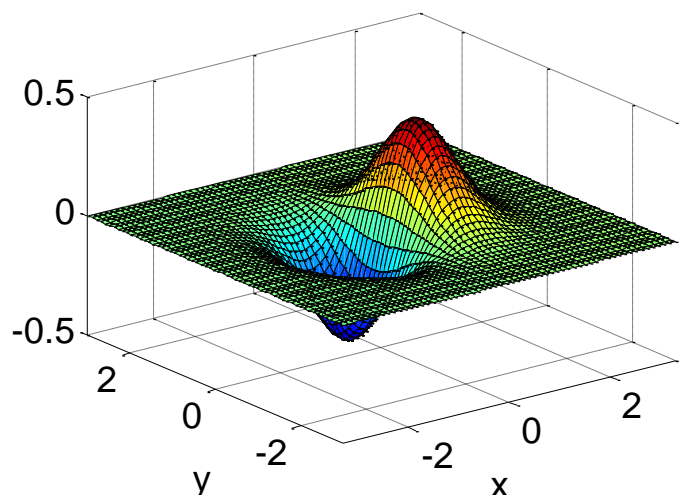
ezsurf(Fun) plots a graph of the function Fun(X,Y) using surf. Fun is plotted over the default domain $-2\pi < X < 2\pi$, $-2\pi < Y < 2\pi$.

ezsurf(Fun, Domain) plots Fun over the specified Domain instead of the default domain. Domain can be the vector [Xmin, Xmax, Ymin, Ymax] or the vector [A, B] (to plot over $A < X < B$, $A < Y < B$).

Example: $f(x, y) = xe^{-x^2-y^2}$

ezsurf('x*exp(-x^2-y^2)')

$$x \exp(-x^2 - y^2)$$



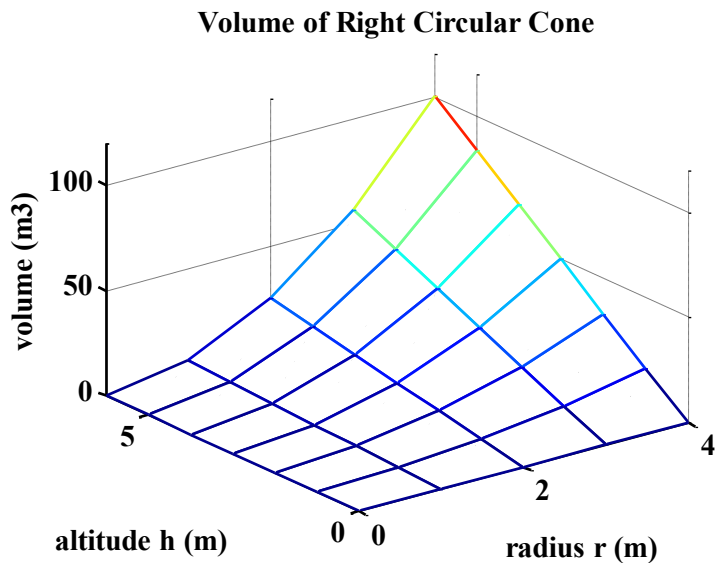
Exercise 1:

The volume of a right circular cone of radius and height is given by $V = \frac{1}{3}\pi r^2 h$

Plot the volume of the cone as and vary on the radius and height.

Solution:

```
[R,H]=meshgrid(0: 4, 0: 6);  
V=(pi .* R .^ 2 .* H) ./ 3;  
mesh(R, H, V)  
xlabel('radius r (m)');  
ylabel('altitude h (m)');  
zlabel('volume (m³)');  
title('Volume of Right  
Circular Cone');
```



The three-dimensional plot of Figure above shows how the volume of the cone increases as the radius and height are increased.

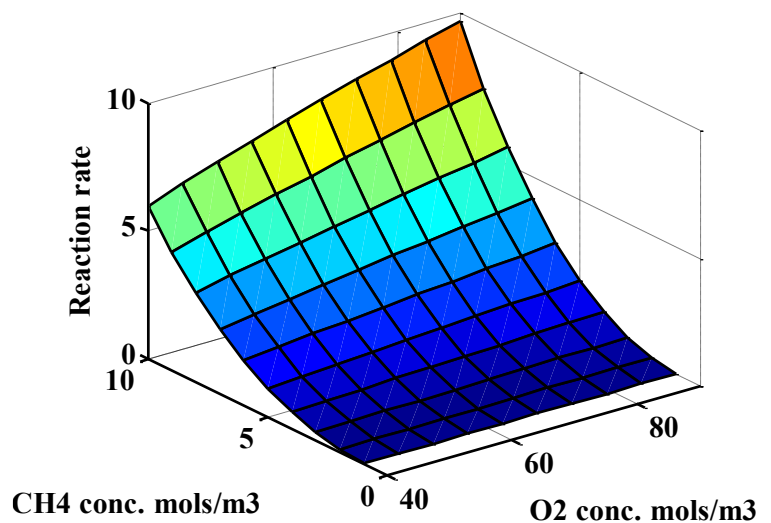
Exercise 2:

We would like to plot a surface for the rate of reaction of methane oxidation. The reaction rate is represented power law $R = k_r c_{\text{CH}_4}^{0.6} c_{\text{O}_2}^{2.4}$, where k is the reaction rate constant $k_r = 2.6 \times 10^{-3}$

The concentration of O_2 in the range between 1 to 10 mols/ m^3 , where the Methane concentration in the range 40 to 90 mols/ m^3 . Taken 10 point of each component plot a surface for the reaction rate.

Solution

```
kr=2.6e-3;  
X1=linspace(40,90,10);  
X2=linspace(1,10,10);  
[Xch4,Xo2]=meshgrid(X1,X2);  
R=kr*(Xch4.^0.6).*(Xo2.^2.4);  
Surf(Xch4,Xo2,R);  
xlabel('O2 conc. mols/m3')  
ylabel('CH4 conc. mols/m3')  
zlabel('Reaction rate')
```



Practice Problems

1) Create surface plots for the functions

$$z = (x - 2)^2 + 2xy + y^2$$

$$z = x^2 - 2xy + 4y^2$$

$$z = -x^2 + 2xy + 3y^2$$

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

2) For gases, the pressure is given by the van der Waals equation of state,

$$p = \frac{8}{3} \frac{t}{v - \frac{1}{3}} - \frac{3}{v^2}$$

Plot this equation in three dimensional graph over the range $0.8 \leq t \leq 1.2$

and $v = 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$.