Class_Work_11

Contents

3-D Graphics

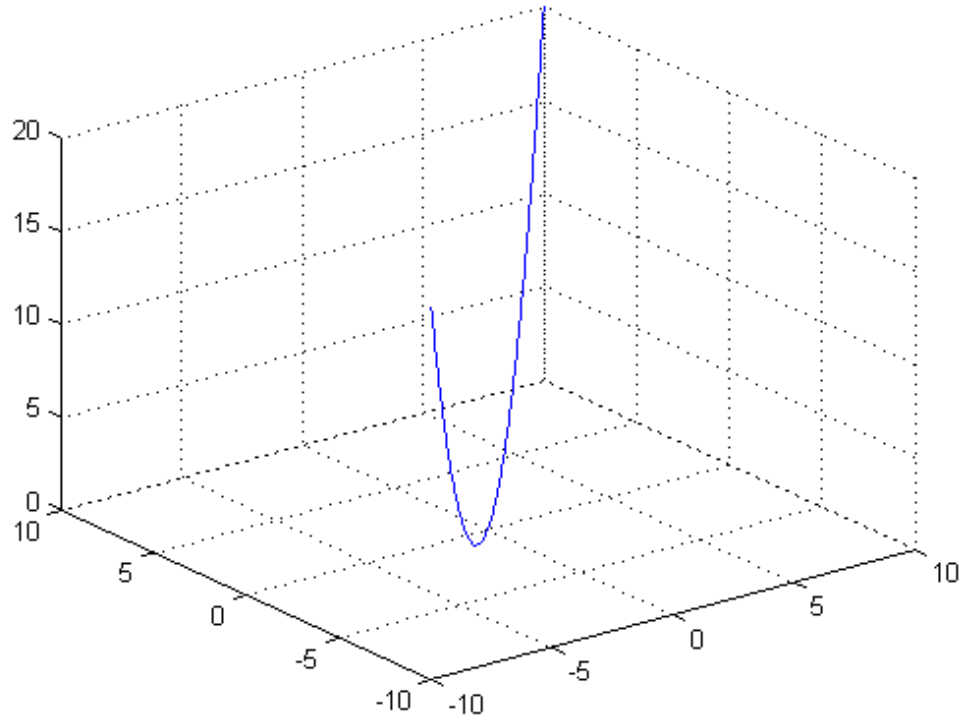
General

```
clc;clear all;clc;clf
```

Plot3 - 3D line plot

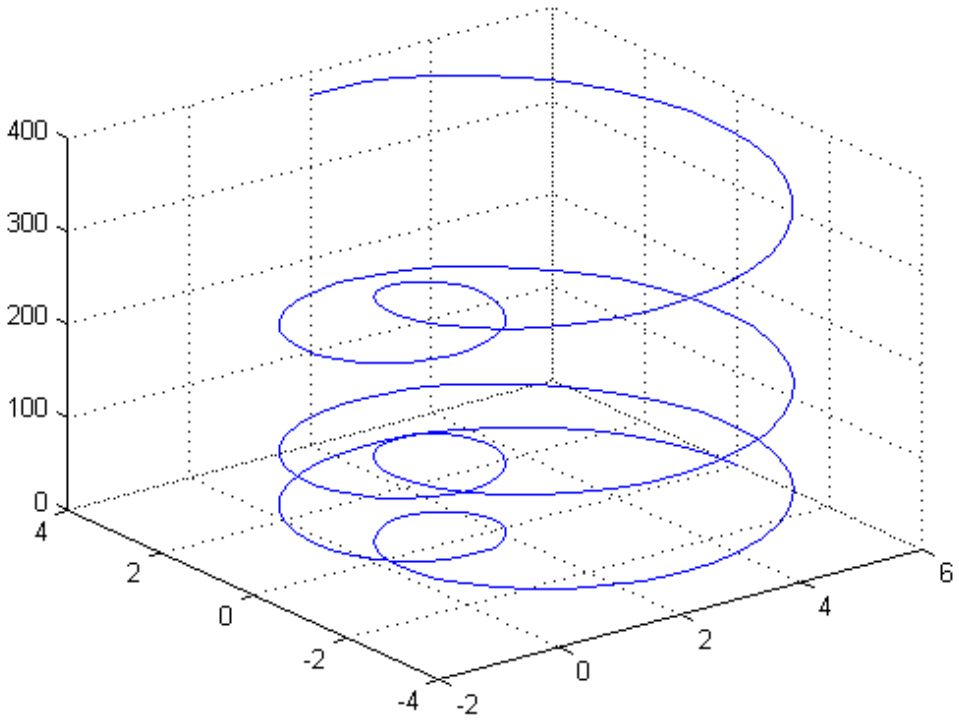
syntax: plot3(X1,Y1,Z1,S1,...) where X1, Y1, Z1 can be scalars, vectors or matrices, plots one or more lines in 3D space. S1 is a string determines line style, marker symbol, and color of the plotted lines

```
x=-10:0.1:10;
y=x;
z=0.1*(x.^2+y.^2);
plot3(x,y,z)
grid on
```



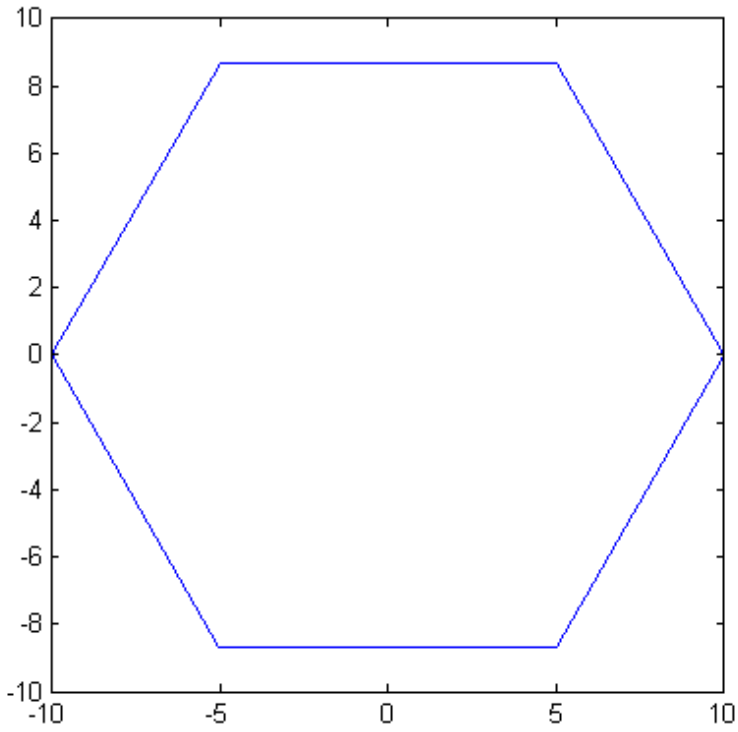
class_assign_11,1

```
t=0:0.1:20;
x=(2+4*cos(t)).*cos(t);
y=(2+4*cos(t)).*sin(t);
z=t.^2;
plot3(x,y,z), grid on
```



Create one Polygon

```
clc;clear all;clf;
R=10;n=6;%R is the radius, n is the number of the polygon's side
%teta divide the circle arc into n equal segments
teta=linspace(0,2*pi,n+1);
 % x and y are the coordinates of the polygon's vertices
x=R*cos(teta);
y=R*sin(teta);
plot(x,y)
axis square
axis on
```

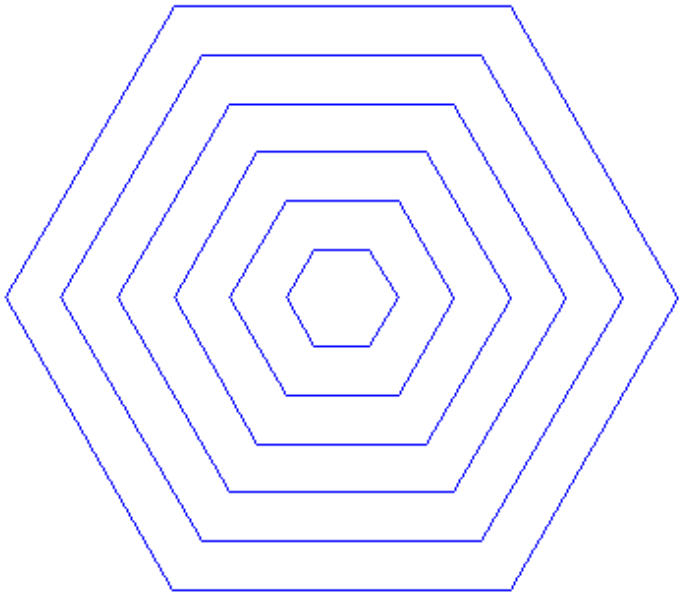


class_assign_11,2

see also polygon.m

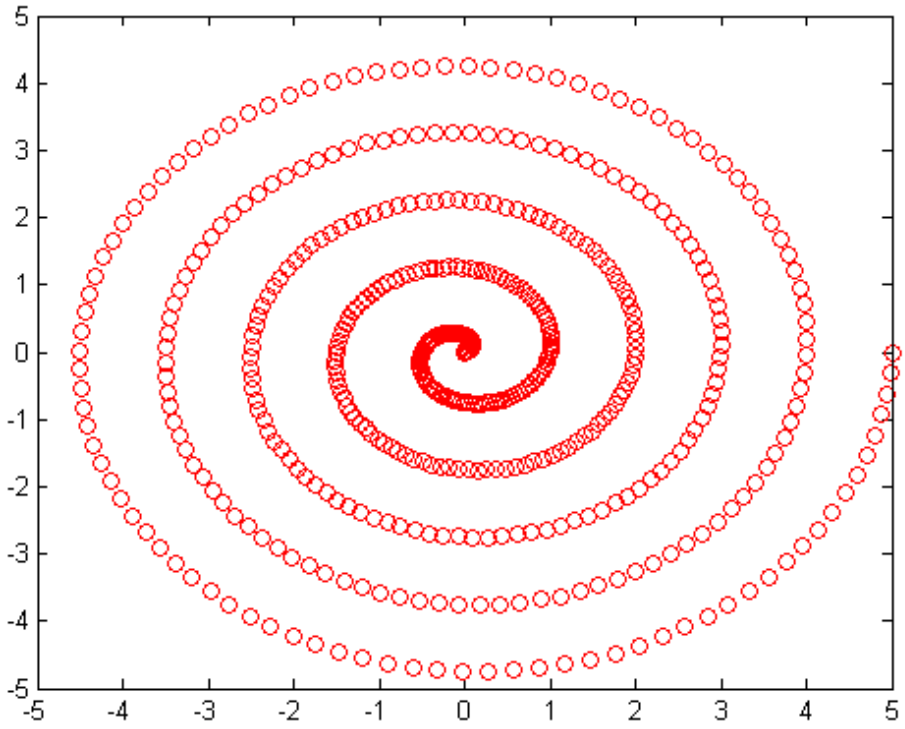
```
%n=input('the number of the polygon's sides? ');
%k=input('the number of the polygons?');
% for example:
clc;clear all;clf;
n=6,k=6
teta=linspace(0,2*pi,n+1);
for i=1:k
   x=i*cos(teta);
   y=i*sin(teta);
   plot(x,y);
   hold on
end
axis square
hold off
axis off
```

```
n =
     6
k =
     6
```



Spiral

```
r=linspace(0,5,500);
teta=linspace(0,10*pi,500);
y=r.*sin(teta);
x=r.*cos(teta);
plot(x,y,'ro')
```



Meshgrid-Generate two arrays containing the x- and y-coordinates at each position in a rectilinear grid

Syntax: [X,Y]=meshgrid (x,y)

```
clc;clear all;clc;
x=1:3;
y=10:14;
% [x1, y1]=meshgrid(x,y)
 %returns two 5X3 matrices - the x1 matrix defines the x-coordinates
 %and the y1 matrix the y-coordinates at each position in an 5 × 3 grid. T
[x1, y1]=meshgrid(x,y)
 [x2, y2]=meshgrid(y,x)
 [x3, y3]=meshgrid(x)
```

```
x1 =
     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3
y1 =
    10    10    10
    11    11    11
    12    12    12
    13    13    13
    14    14    14
x2 =
    10    11    12    13    14
    10    11    12    13    14
    10    11    12    13    14
y2 =
     1     1     1     1     1
     2     2     2     2     2
     3     3     3     3     3
```

```
x3 =
     1     2     3
     1     2     3
     1     2     3
y3 =
     1     1     1
     2     2     2
     3     3     3
```

Surface plot- Example

```
clc;clear all;clc;clf
%1. Define the vectors in the x y plane
%(resolution and range)
x=-pi:0.1:pi;
y=x;
%2. Generate the grid
[x1, y1]=meshgrid(x);
% 3. Evaluate the function
z=sin(x1).*y1.^3;
%plot3(x1,y1,z) % not recommended for sulface, plot 3D lines not a surface
```

mesh(X,Y,Z) – creates a 3D mesh of the function

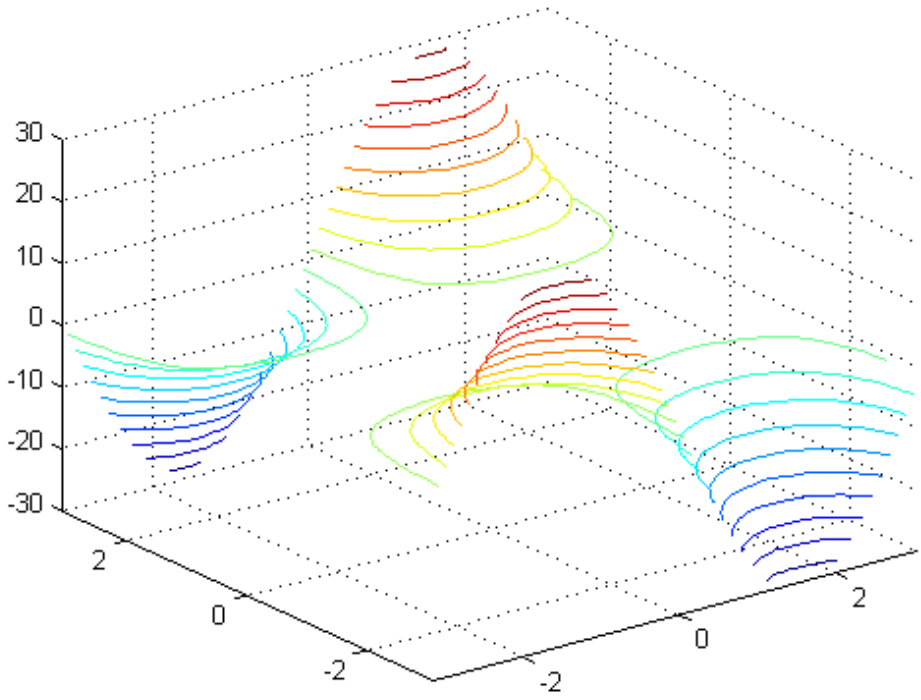
```
mesh(x1,y1,z)
grid on
```

surf(X,Y,Z) – creates a 3D surface of the function

```
surf(x1,y1,z)
```

contour3(X,Y,Z,n_contours) – creates a 3D contour of the function

```
contour3(x1,y1,z,20)
```



surfc(X,Y,Z)- draws a contour plot beneath the surface

```
surfc(x1,y1,z)
```

Surface plot: z=sin(xy/50)^2+2exp(-(x^2+y^2)/1500)

```
clc;clear all;clf;close all
x1=-90:5:90;
y1=x1;
[x y]=meshgrid(x1);
z=sind(x.*y/50).^2+2*exp(-(x.^2+y.^2)/1500);
subplot(2,2,1)
mesh(x,y,z)
subplot(2,2,2)
surf(x,y,z)
subplot(2,2,3)
contour3(x,y,z,50)
subplot(2,2,4)
surfc(x,y,z)
```

View: determines the orientation of the axes

syntax: view(az,el) for azimuth and elevation

view ([x,y,z]) for viewpoint location

```
clc;clear all;clf;close all
x1=-10:0.5:10;
[x y]=meshgrid(x1);
z=0.1*(x.^2+y.^2);
mesh(x,y,z)
view(-130,35)
```

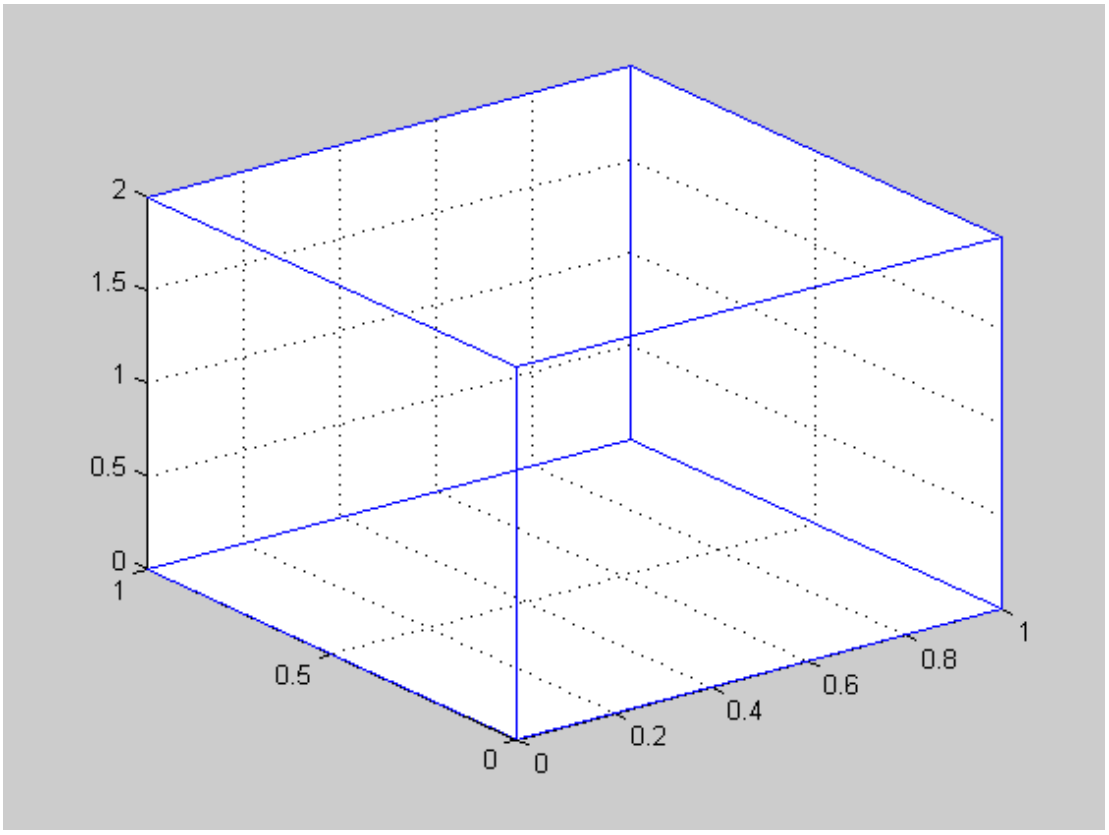
%% colorbar

Add a colorbar to a plot with the colorbar tool on the figure toolbar

```
colorbar %Colorbar showing color scale
```

3D box

```
clc;clear all;clf;close all
x=[0 1 1 0 0];
y=[0 0 1 1 0];
z=zeros(1,5);
plot3(x,y,z)
grid on
hold on
z1=z+2; %z and z1 are the basis of the box
plot3(x,y,z1)
% plotting the sides
for i=1:4
plot3([x(i) x(i)],[y(i) y(i)],[z(i) z1(i)])
end
```

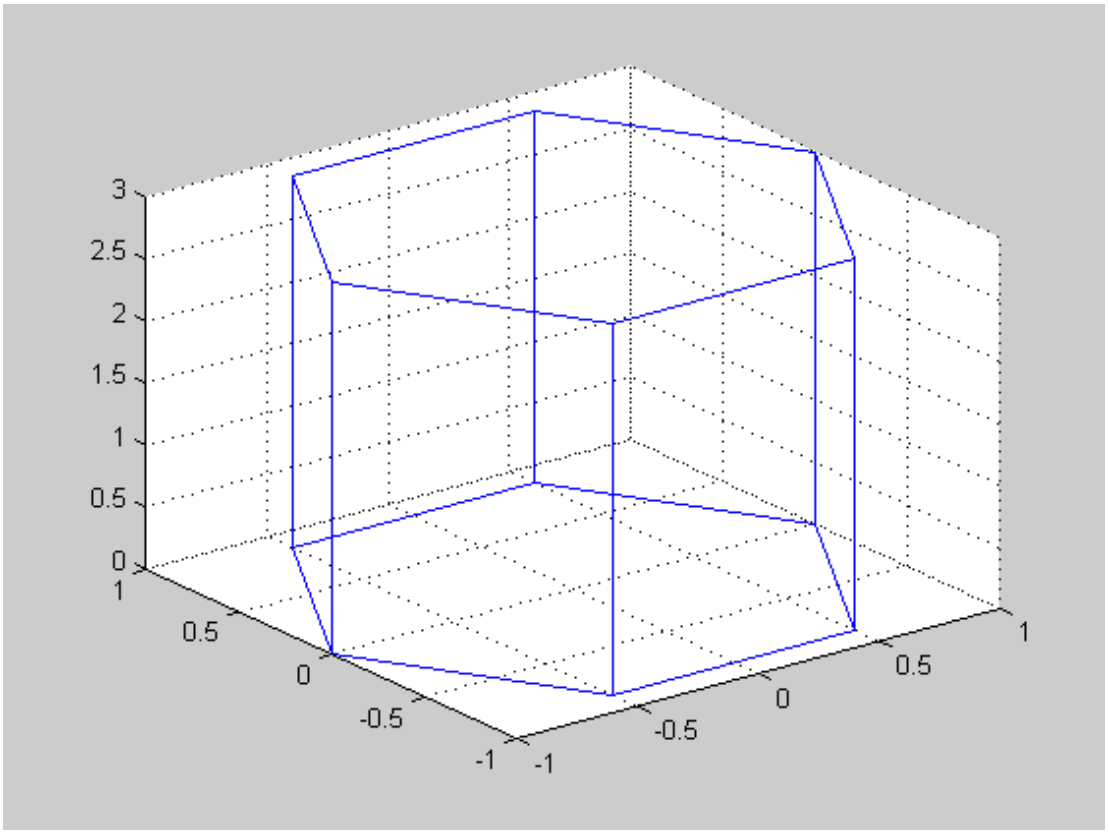


class_assign_11,3

see minsara.m

```
clc;clear all;clf;
minsara(6,3)
```

```
teta =
         0    1.0472    2.0944    3.1416    4.1888    5.2360    6.2832
y =
         0    0.8660    0.8660    0.0000   -0.8660   -0.8660   -0.0000
s =
krm
```



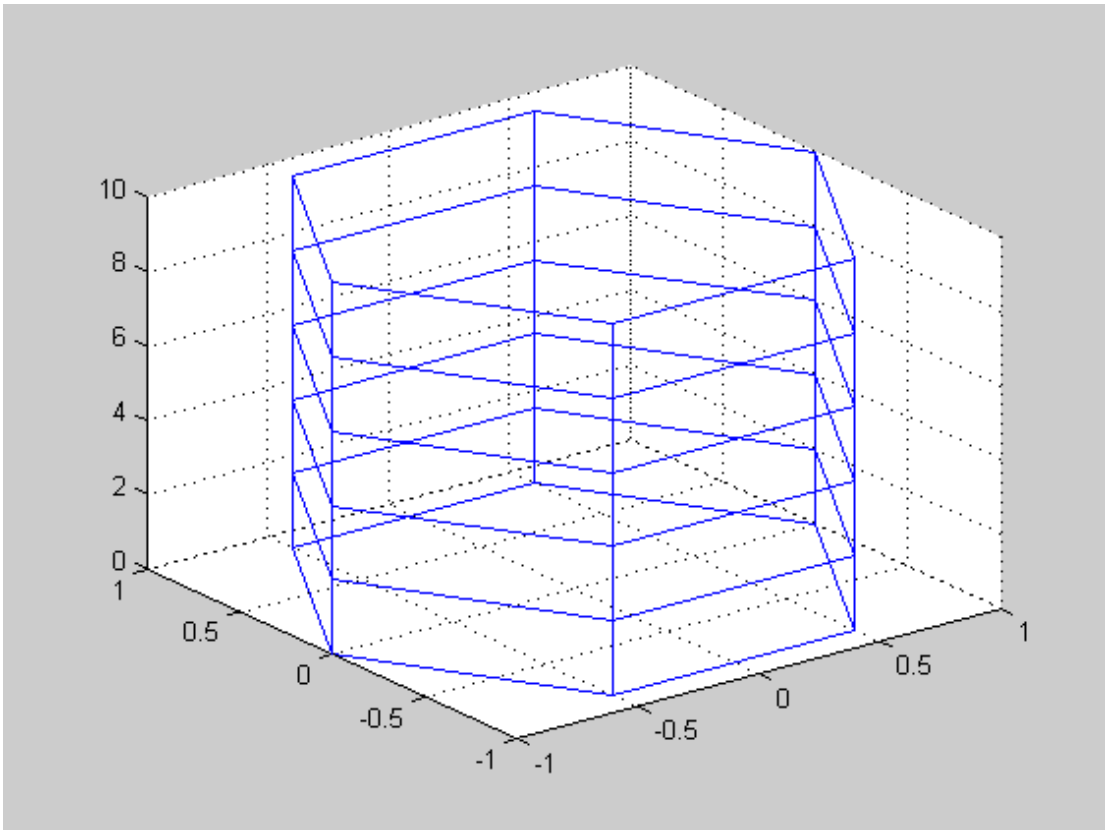
class_assign_11,4

A tower of prisms

Use the user-defined function minsara2 to build a tower of several prisms

```
clc;clear all;clf;close all
n=6; b=0;m=5;h=2;r=1
for i=0:m-1
    minsara2(n,b+i*h,h,r)
    hold on
   end
```

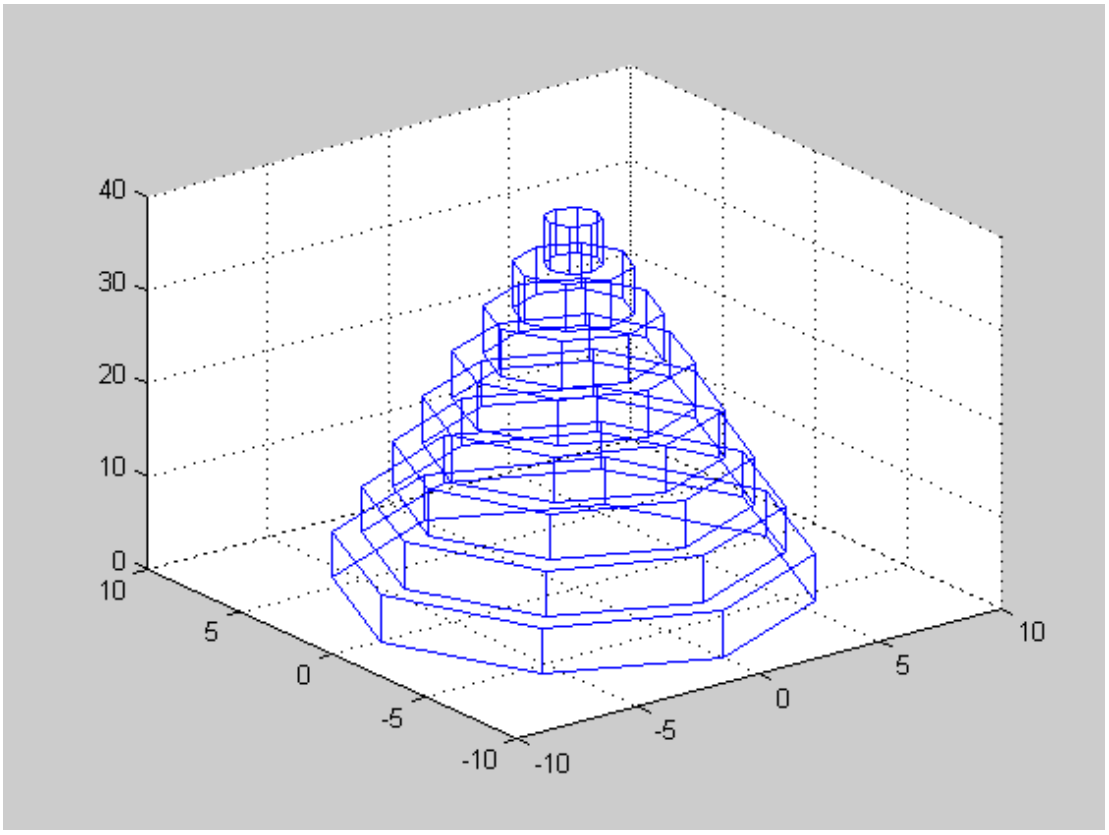
```
r =
     1
```



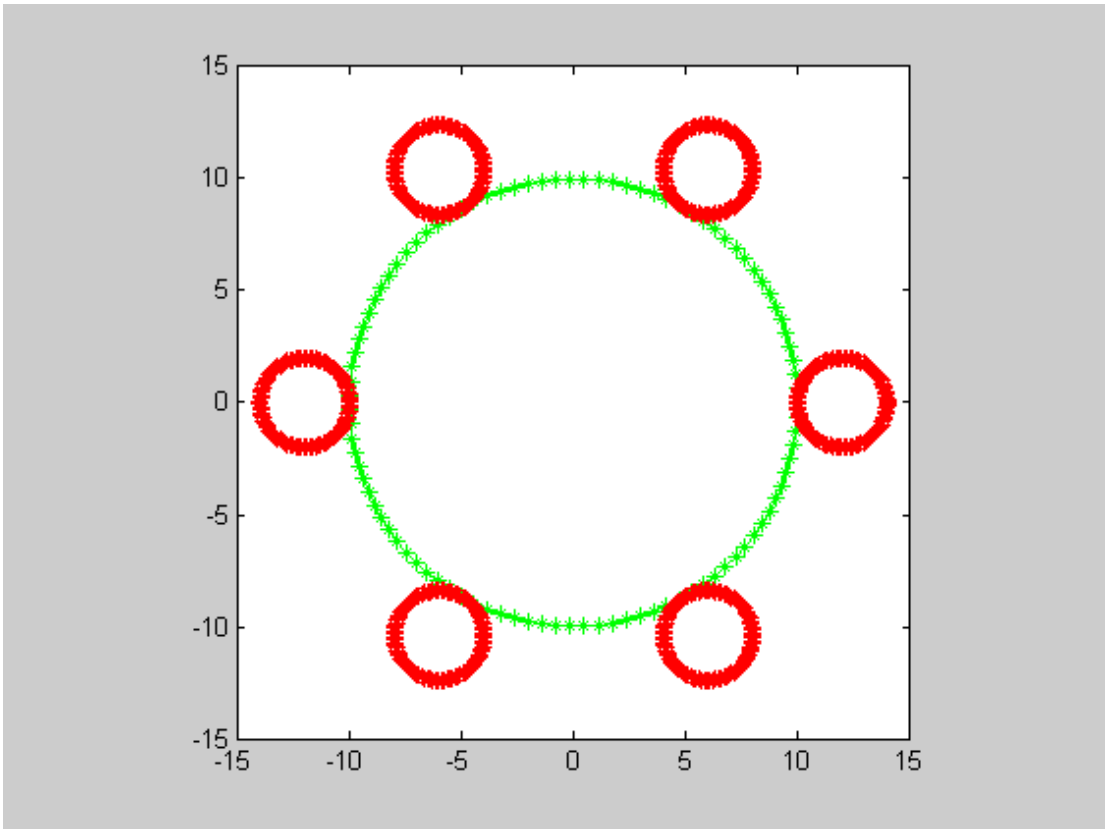
class_assign_11,5

Pyramid of prisms

```
   clc;clear all;clf;
clf, hold off
h=5; b=0;n=8;
for i=0:n-1
minsara2(n,b+i*h,h,n-i)
hold on
end
```




```
clc;clear all;clf
flower(6,10,2)
```



Published with MATLAB® 7.6