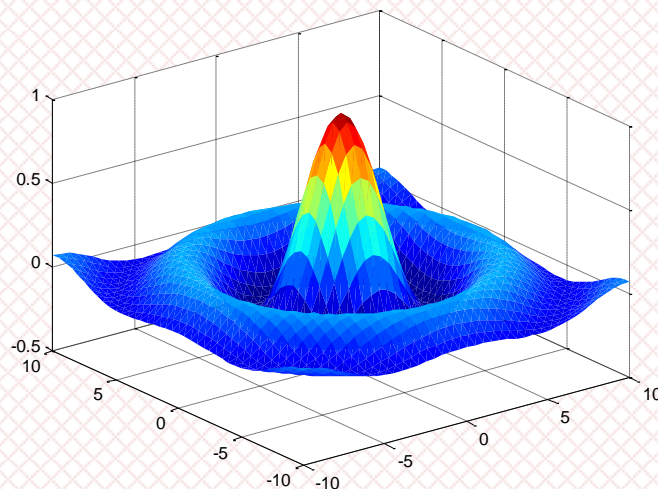


# 第二章 Matlab图形绘制

## (2) 三维图形





## 2.2 三维图形

2.2.1、三维曲线: `plot3()`

2.2.2、二维数据网格: `meshgrid()`

2.2.3、三维网格图: `mesh()` / `meshc()` / `meshz()`

2.2.4、三维表面图: `surf()` / `surfc()` / `surfl()`

2.2.5、图形修饰

2.2.6、特殊的三维图形函数

2.2.7、应用

## 2.2.1、三维曲线: **plot3( )**

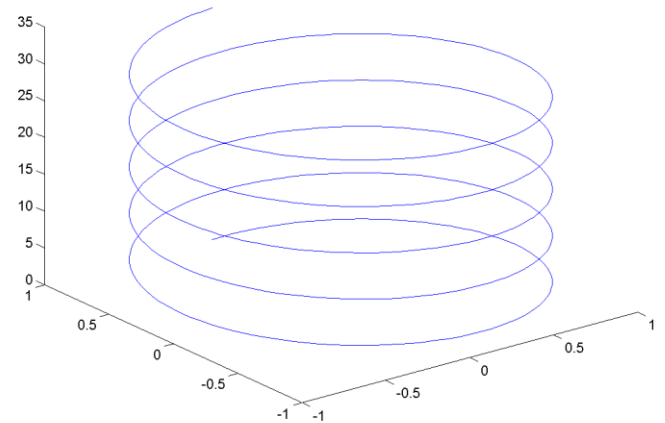
### 绘制三维曲线的最基本函数

**plot3**函数与**plot**函数用法十分相似，其调用格式为：

**plot3(x1,y1,z1,选项1,x2,y2,z2,选项2,...,xn,yn,zn,选项n)**

其中**xi,yi,zi**为3 个相同长度的向量。

例：作螺旋线  $x=\sin t, y=\cos t, z=t$



## plot3( )

```
plot3(X1,Y1,Z1,...)
```

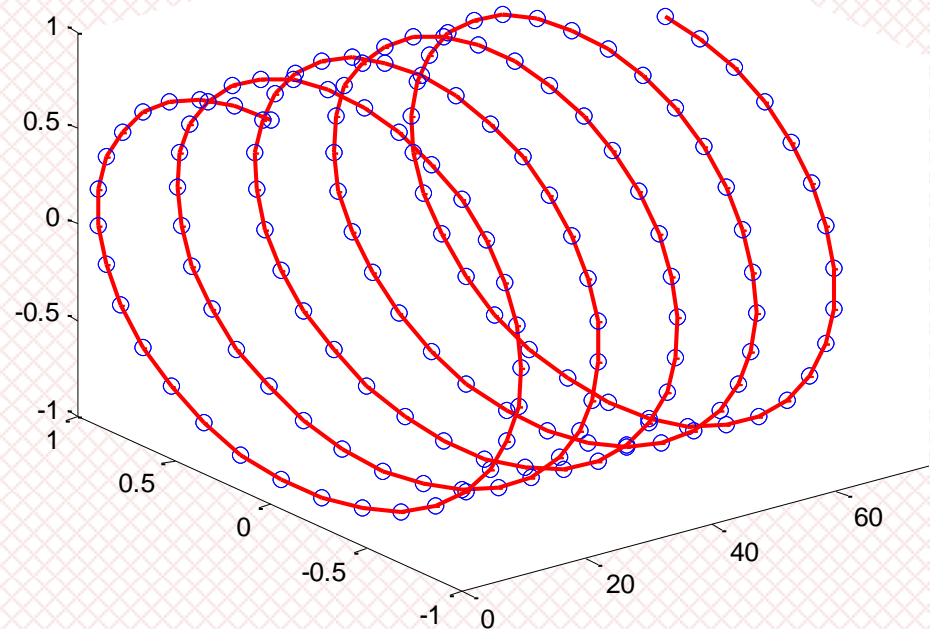
```
plot3(X1,Y1,Z1,LineSpec,...)
```

```
plot3(...,'PropertyName',PropertyValue,...)
```

```
h = plot3(...)
```

plot3( )的用法与plot( )类似，只是多了一个 Z 数组。

```
t=[0:0.2:10*pi];  
x=2*t;  
y=sin(t);  
z=cos(t);  
plot3(x,y,z,'bo');  
hold on  
plot3(x,y,z,'r-','LineWidth',2)
```



例： 绘制空间曲线。

$$X=8\cos t, y=4\sqrt{2}\sin t, z=-4\sqrt{2}\sin t$$

程序如下：

```
t=0:pi/50:2*pi;
```

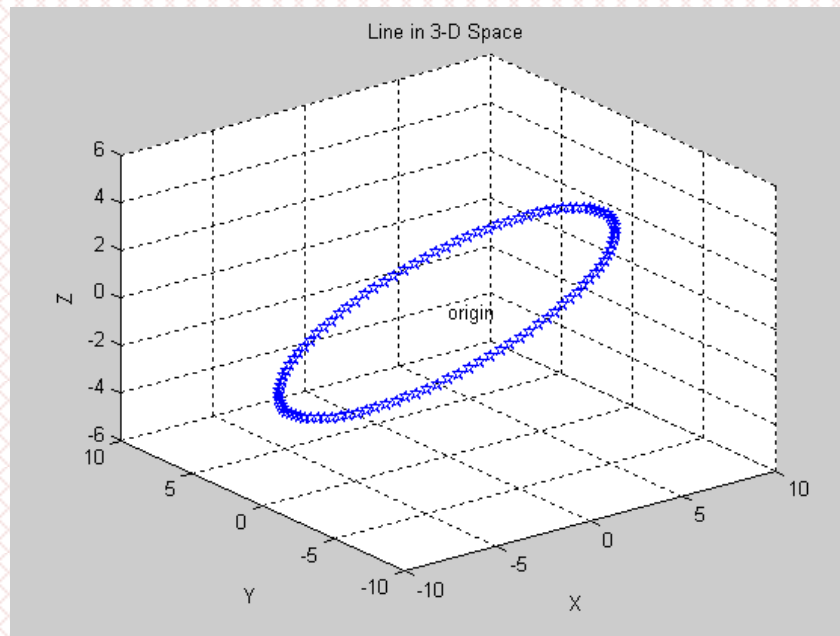
```
x=8*cos(t); y=4*sqrt(2)*sin(t);
```

```
z=-4*sqrt(2)*sin(t);
```

```
plot3(x,y,z,'p');
```

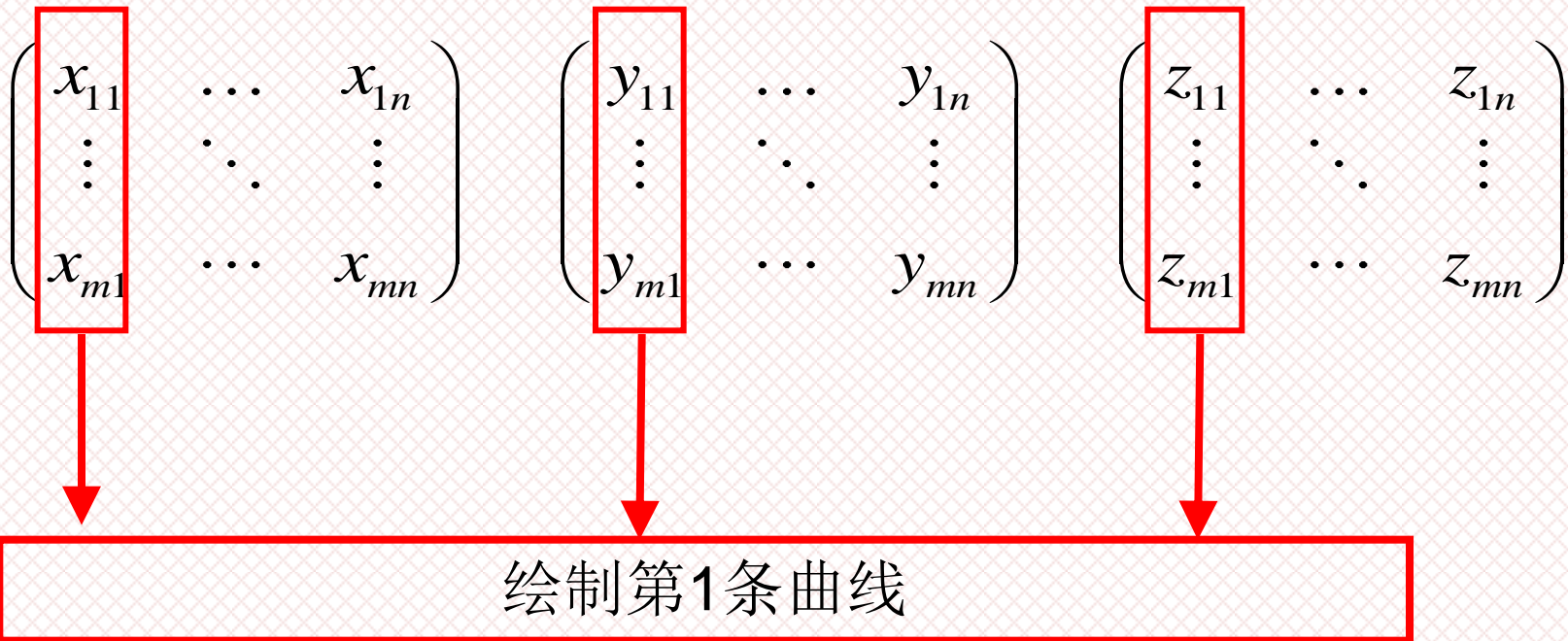
```
title('Line in 3-D Space'); text(0,0,0,'origin');
```

```
xlabel('X'),ylabel('Y'),zlabel('Z'); grid on;
```



# 用plot3( )同时绘制多条3d曲线

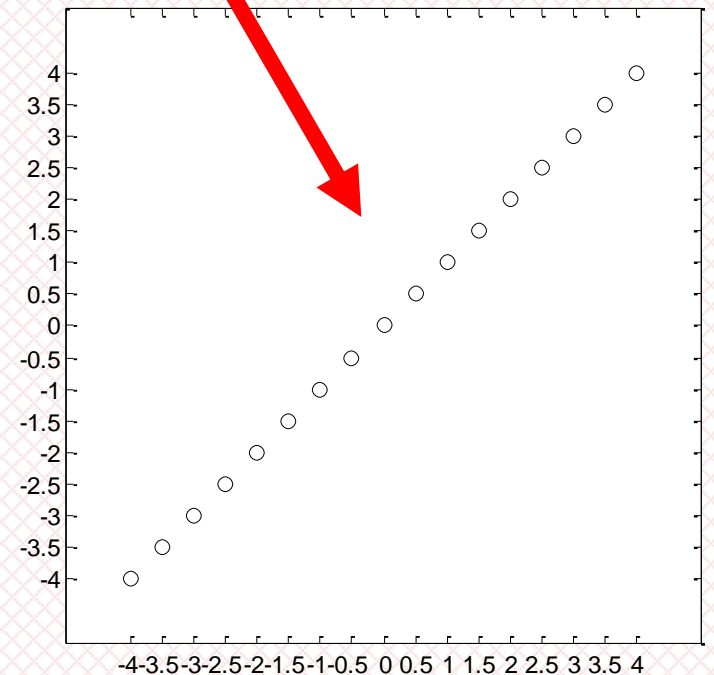
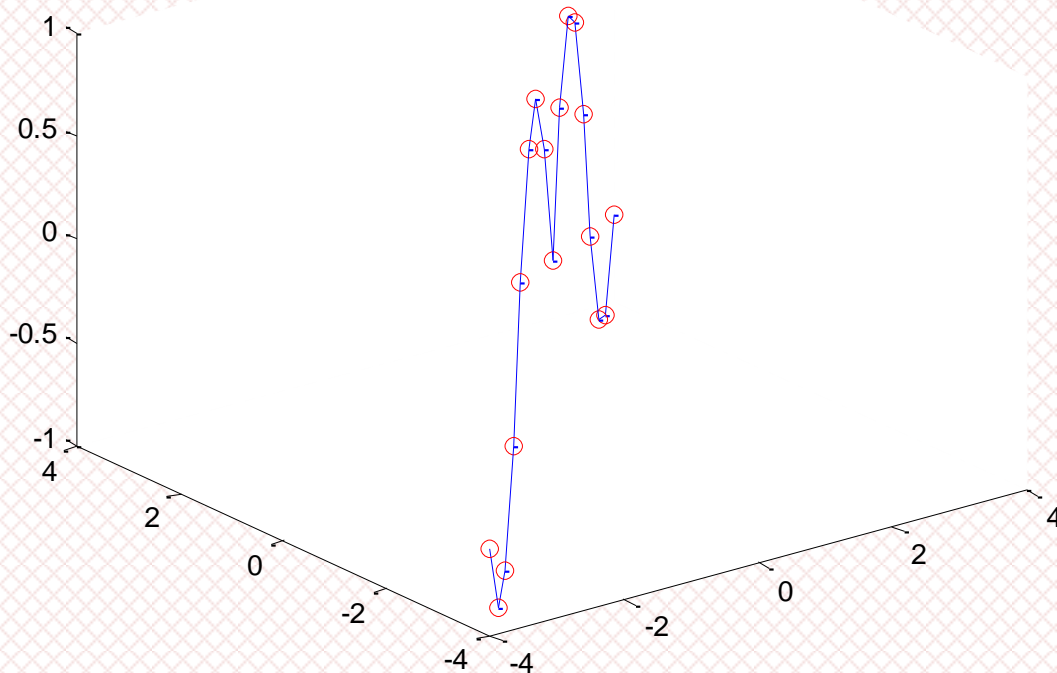
当 $X$ ， $Y$ ， $Z$ 为同维的二维数组， $\text{plot3}()$ 将  $X$ 、 $Y$ 、 $Z$  相应的列相组合，绘制多条3d曲线。

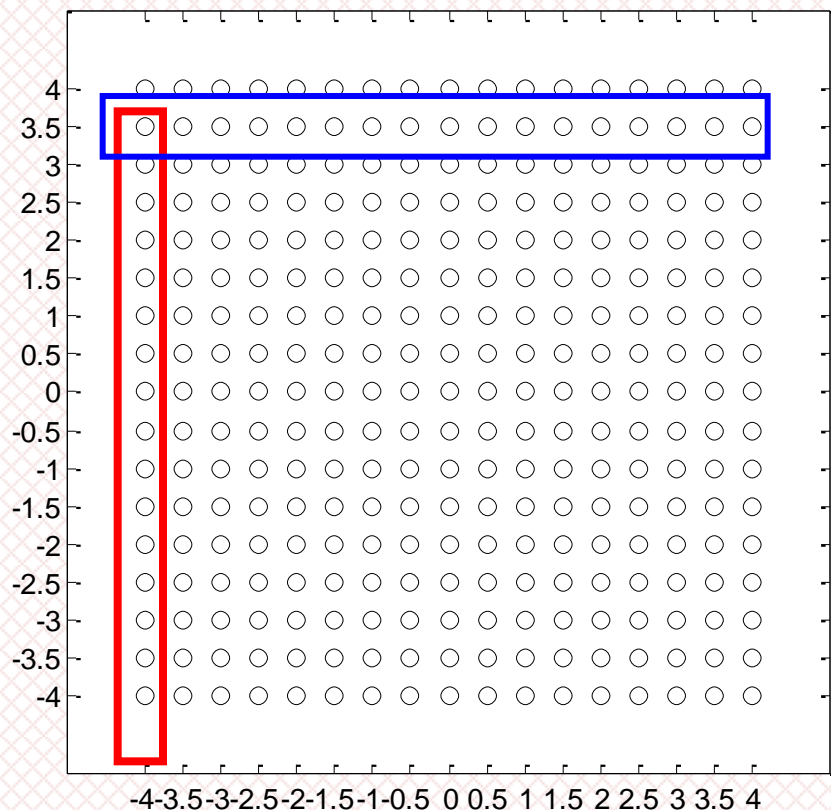


如何用plot3( )绘制 $z = \sin(\sqrt{x^2 + y^2})$ ?

```
x = ( -4:0.5:4 )';  
y = ( -4:0.5:4 )';  
z = sin(sqrt(x.^2+y.^2));  
h = plot3(x,y,z,'-bo');  
set(h,'MarkerEdgeColor','r');
```

x	y	z
-4.0000	-4.0000	-0.5862
-3.5000	-3.5000	-0.9720
-3.0000	-3.0000	-0.8917
-2.5000	-2.5000	-0.3838
.....		
3.0000	3.0000	-0.8917
3.5000	3.5000	-0.9720





```
x = -4:0.5:4;
y = ( 4:-0.5:-4 ) `;
```

```
X = repmat(x,length(y),1)
Y = repmat(y,1,length(x))
```

**repmat**全称是Replicate Matrix  
**repmat(A,m,n)**, 将矩阵 **A** 复制  $m \times n$  块

**X** =

-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000
...	...	...	...	...
-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000

**Y** =

4.0000	4.0000	...	4.0000	4.0000
3.5000	3.5000	...	3.5000	3.5000
...	...	...	...	...
-3.5000	-3.5000	...	-3.5000	-3.5000
-4.0000	-4.0000	...	-4.0000	-4.0000

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





## 2.2.2、二维数据网格 **meshgrid( )**

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

由向量  $x$  和  $y$  生成二维数组  $X$  和  $Y$ ，用来计算二元函数  $f(x,y)$  的值  $Z = f(X,Y)$ 。二维数组  $X$ ， $Y$ ， $Z$  可用来绘制三维曲线、三维网格图、三维曲面图等。

输出数组  $X$  中的行向量相当于向量  $x$ ，输出数组  $Y$  中的列向量相当于向量  $y$ 。

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

等价于  $[X,Y] = \text{meshgrid}(x,x)$ 。



# 1. 平面网格坐标矩阵的生成

(1)利用矩阵运算生成。

```
x=a:dx:b; y=(c:dy:d)';
```

```
X=ones(size(y))*x;
```

```
Y=y*ones(size(x));
```

(2)利用meshgrid函数生成。

```
x=a:dx:b; y=c:dy:d;
```

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

%表示生成一个横坐标起始于a,步长为dx,终止于b; 纵坐标起始于c,步长为dy,终止于d的网格分割。 $(x_i, y_j)$ 代表网格的坐标。



产生一个小网格:

**[X,Y]=meshgrid(0:3,0:3)**

**X =**

**0   1   2   3**

**0   1   2   3**

**0   1   2   3**

**0   1   2   3**

**Y =**

**0   0   0   0**

**1   1   1   1**

**2   2   2   2**

**3   3   3   3**

```
x = -4:0.5:4;  
y = -4:0.5:4;  
[X,Y] = meshgrid(x,y);  
Z = sin(sqrt(X.^2+Y.^2));
```

```
subplot(2,2,1),  
plot3(X,Y,Z,'bo');
```

```
subplot(2,2,2),  
plot3(X,Y,Z,'b')
```

```
subplot(2,2,3),  
plot3(X',Y',Z','b')
```

```
subplot(2,2,4),  
plot3(X,Y,Z,'b',X',Y',Z','b')
```

X =

-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000
...	...	...	...	...
-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000

Y =

-4.0000	-4.0000	...	-4.0000	-4.0000
-3.5000	-3.5000	...	-3.5000	-3.5000
...	...	...	...	...
3.5000	3.5000	...	3.5000	3.5000
4.0000	4.0000	...	4.0000	4.0000

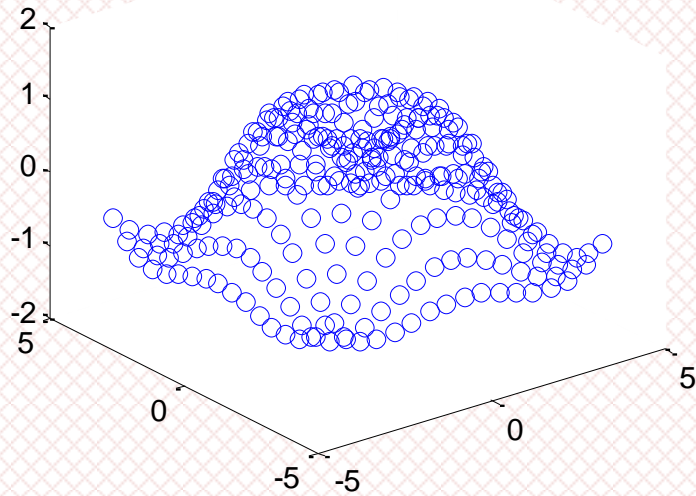
Z =

-0.5862	-0.8238	...	-0.8238	-0.5862
-0.8238	-0.9720	...	-0.9720	-0.8238
...	...	...	...	...
-0.8238	-0.9720	...	-0.9720	-0.8238
-0.5862	-0.8238	...	-0.8238	-0.5862

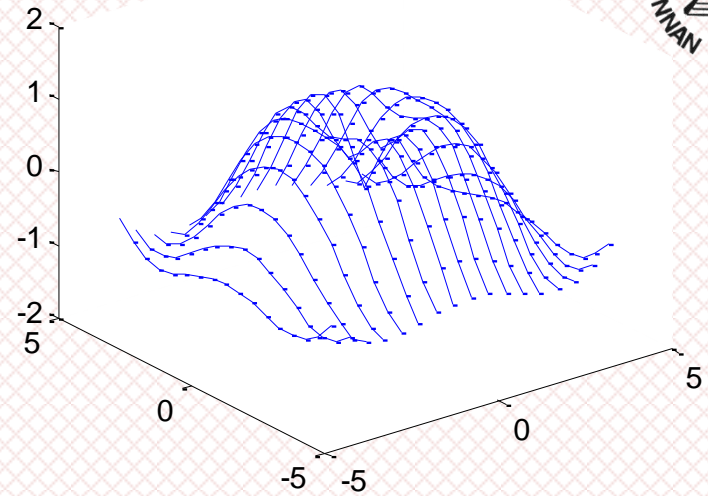
绘制第1条  
3d曲线

绘制第2条  
3d曲线

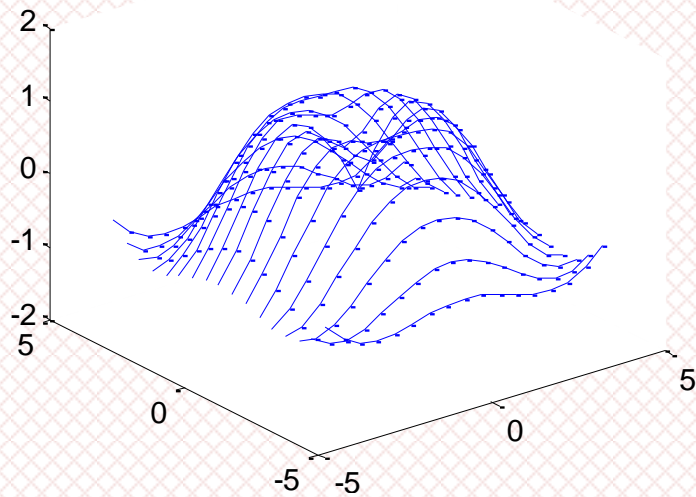
plot3(X,Y,Z,'bo')



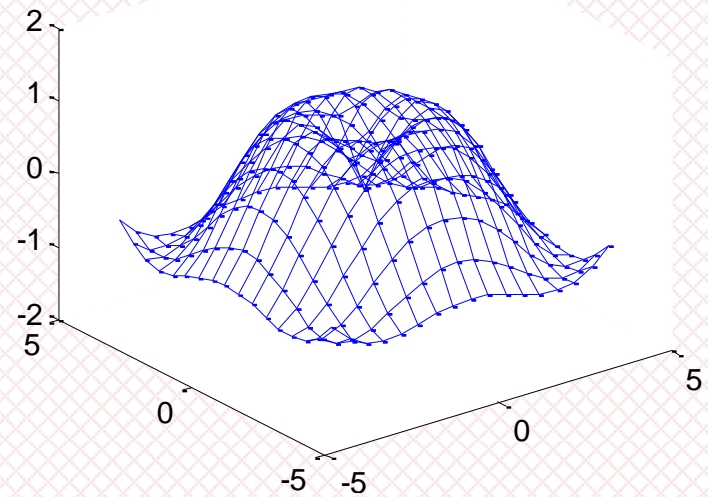
plot3(X,Y,Z,'b')



plot3(X',Y',Z','b')



plot3(X,Y,Z,'b',X',Y',Z','b')



## 2.2.3、三维网格图 `mesh()` / `meshc()` / `meshz()`

**`mesh(X,Y,Z)`**: 绘制由数组 `X,Y,Z` 所确定的曲面的网格图

`X, Y, Z` 都为二维数组时，要求它们的维数相同。`X, Y` 也可以是向量，但 `Z` 必须为二维数组，`[m,n] = size(Z)`，此时必须满足：`length(X) = n` 且 `length(Y) = m`。

**`mesh(Z)`**: 相当于 `X = 1:n`，`Y = 1:m`，其中 `[m,n] = size(Z)`

**`mesh(...,C)`**: 二维数组 `C` 确定网格颜色，省略 `C` 时相当于 `C=Z`

**`mesh(...,'PropertyName',PropertyValue,...)`**: 设置属性值

**`mesh(axes_handles,...)`**: 在指定的坐标轴绘图

**`h = mesh(...)`**: 返回句柄



$$z = \sin\left(\sqrt{x^2 + y^2}\right)$$

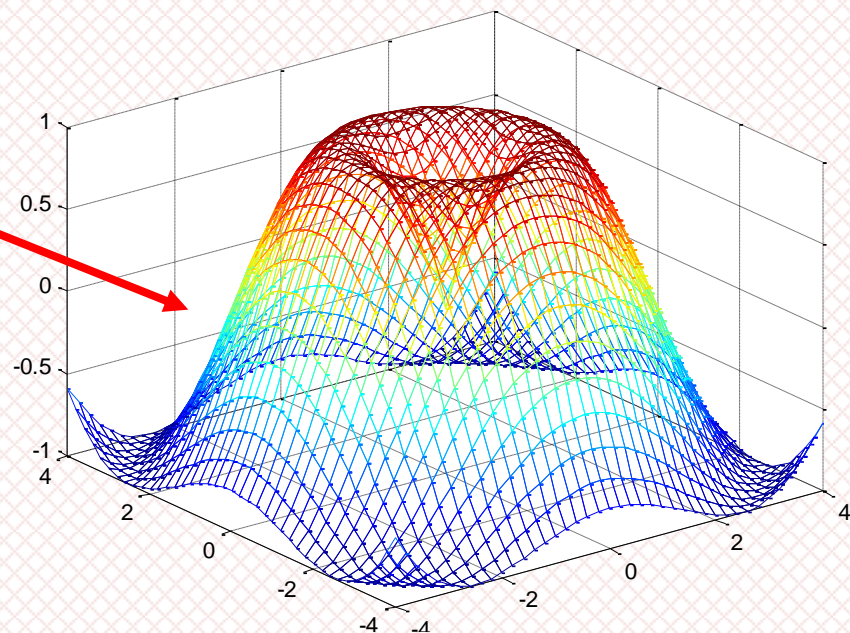
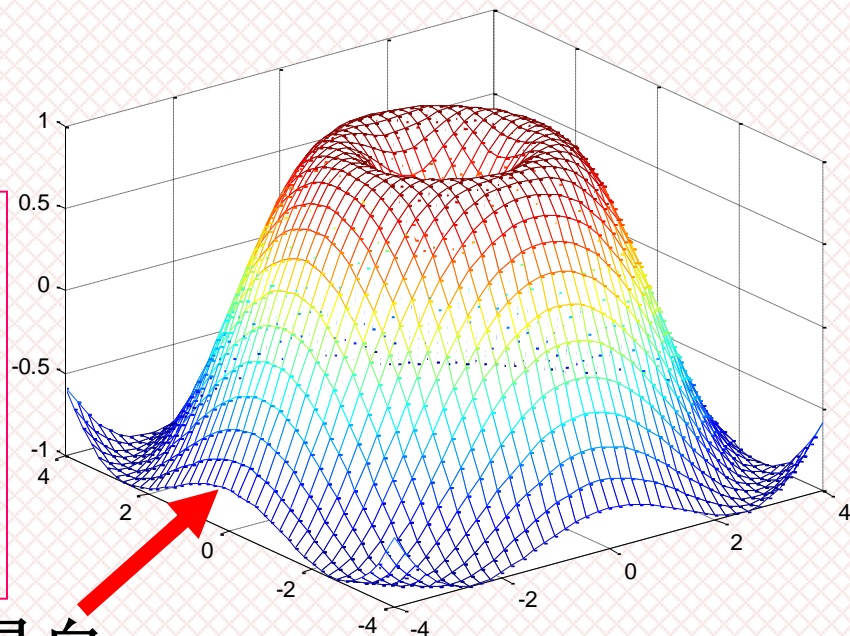
```
x = -4:0.2:4;
[X,Y] = meshgrid(x);
Z = sin(sqrt(X.^2+Y.^2));
h = mesh(X,Y,Z);
c1 = get(h,'FaceColor');
```

默认情况下每个四边形区域填充的是白色，因此c1的值 [1,1,1]

`hidden off`

```
c2 = get(h,'FaceColor');
```

`hidden off`命令是使每个四边形区域不填充任何颜色，是空的，可以看到后面的图线。因此c2的值为 **none**



## **meshc(X,Y,Z)**

调用方式与 **mesh** 相同，在 **mesh** 基础上增加等高线

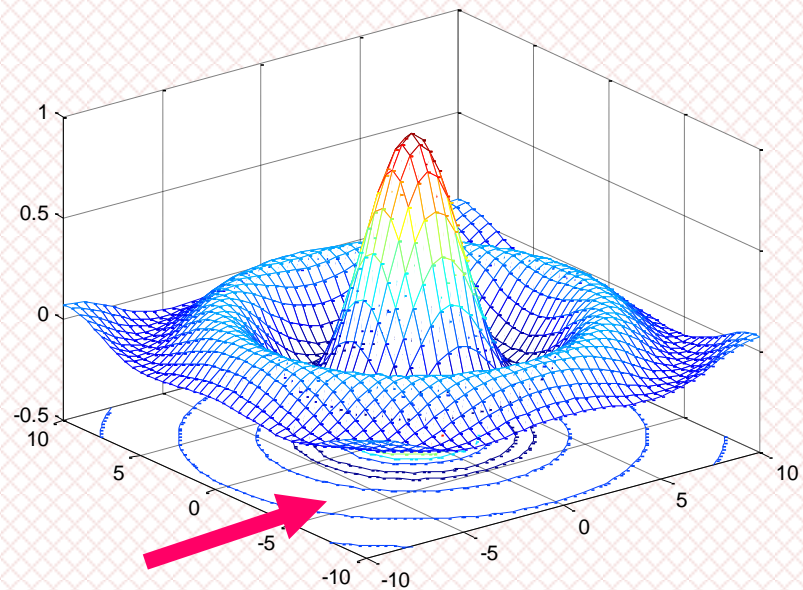
$$Z = \frac{\sin\left(\sqrt{x^2 + y^2}\right)}{\sqrt{x^2 + y^2}}$$

墨西哥帽子

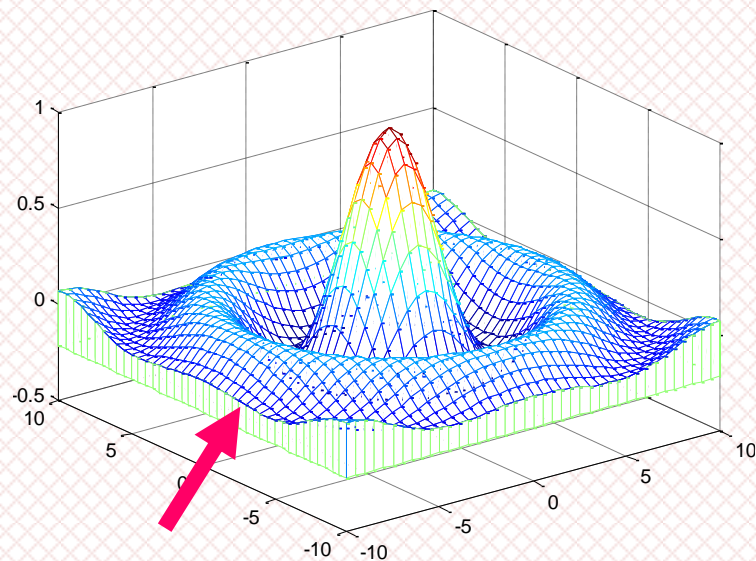
```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x) ;  
r = sqrt(X.^2+Y.^2)+eps ;  
Z = sin(r)./r ;  
meshc(X,Y,Z)
```

## **meshz(X,Y,Z)**

调用方式与 **mesh** 相同，在 **mesh** 基础上屏蔽边界面



等高线



屏蔽边界面



## 2.2.4、三维表面图 **surf()** / **surfc()**



绘制由矩阵 **X,Y,Z** 所确定的表面图，参数含义同 **mesh**

**surf(Z)** : 相当于  $X = 1:n$  ,  $Y = 1:m$  , 其中  $[m,n] = \text{size}(Z)$

**surf(Z,C)** : 二维数组 **C** 确定网格颜色，省略 **C** 时相当于  $C=Z$

**surf(X,Y,Z)** : 绘制由数组 **X,Y,Z** 所确定的曲面图

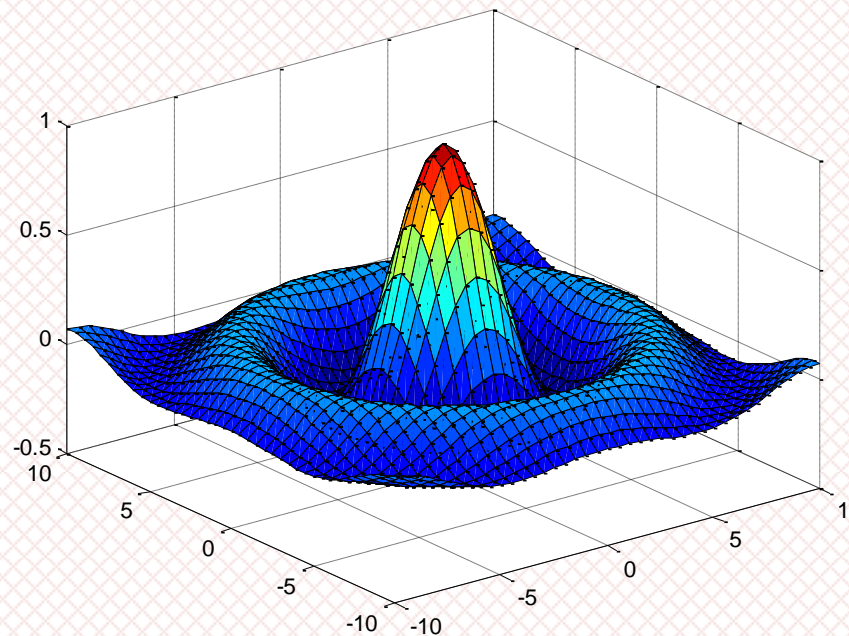
**surf(X,Y,Z,C)** :

**surf(...,'PropertyName',PropertyValue)** : 设置属性值

**surf(axes\_handles,...)**: 在指定的坐标轴绘图

**h = surf(...)** : 返回句柄

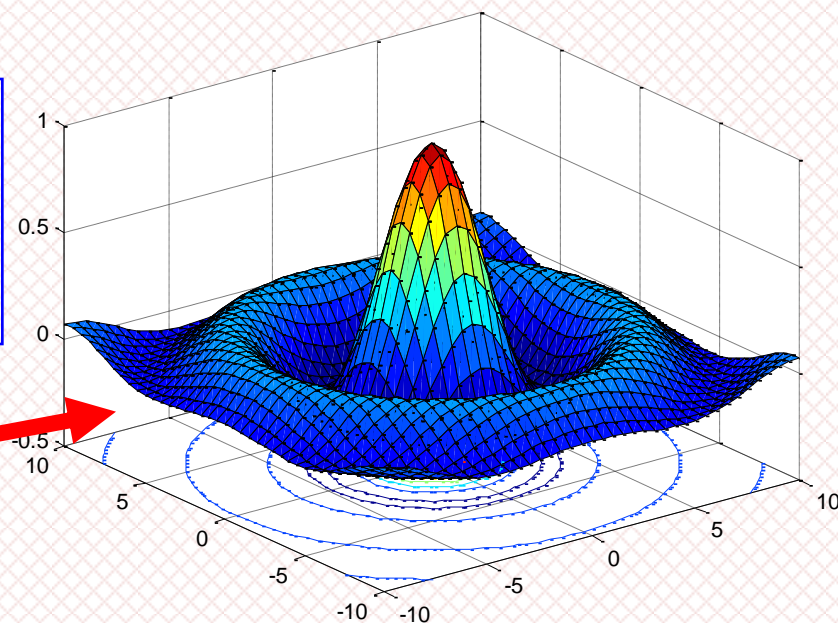
```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x);  
r = sqrt(X.^2+Y.^2)+eps;  
Z = sin(r)./r;  
surf(X,Y,Z)
```



**surf**(X,Y,Z)

调用方式与 surf 相同，在 surf  
基础上增加等高线

等高线



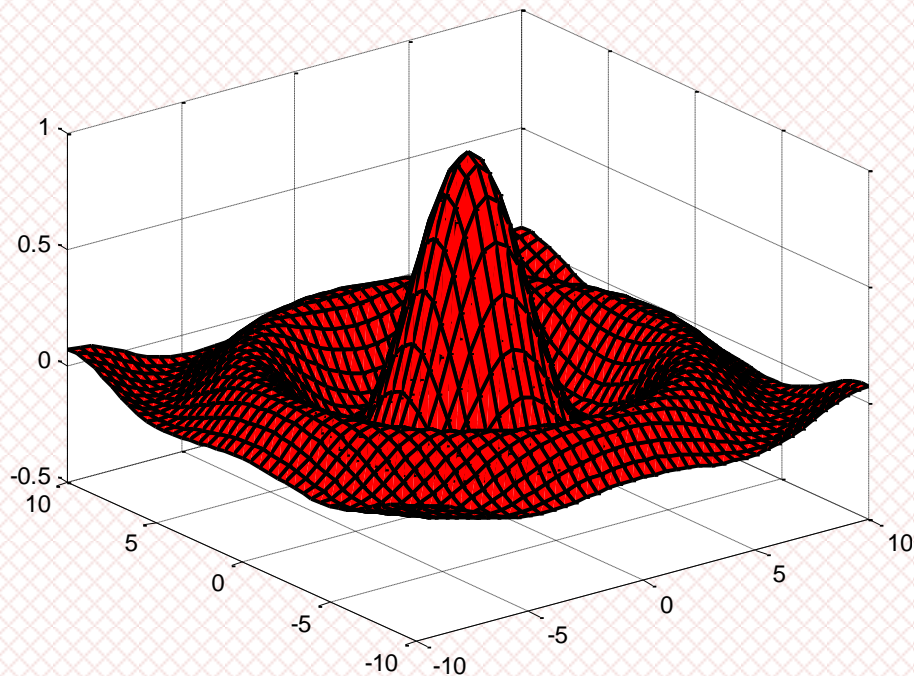
# mesh( ) / surf( )的一些常用属性



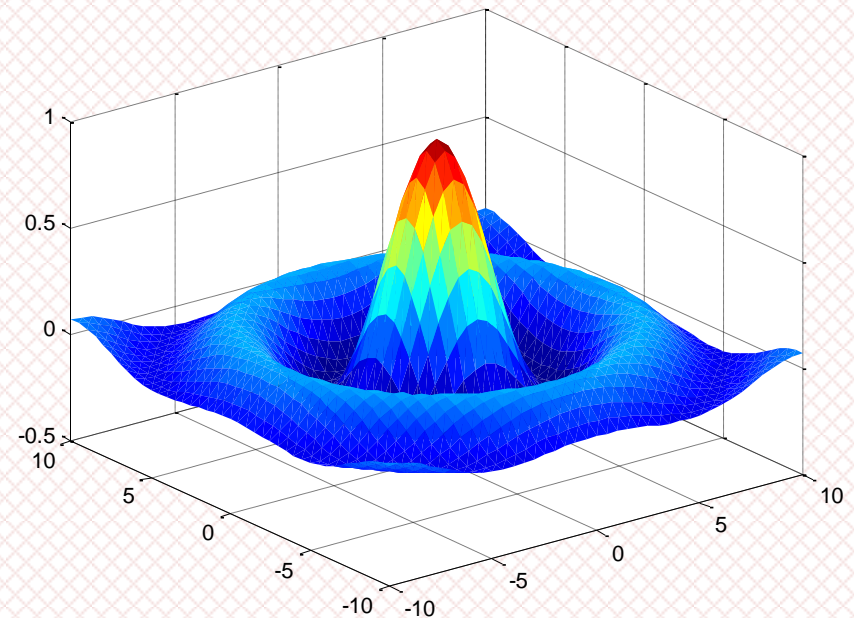
属性名称	意义	取值
<b>EdgeColor</b>	网格线颜色	<b>{ColorSpec}   none   flat   interp</b>
<b>FaceColor</b>	四边形网格的填充颜色	<b>ColorSpec   none   {flat}   interp   texturemap</b>
<b>LineStyle</b>	网格线线型	<b>{-}   --   :   -.   none</b>
<b>LineWidth</b>	网格线线宽	
<b>Marker</b>	标记点形状	<b>none   +   o   *   .   x   s   d   p   h .....</b>
<b>MarkerEdgeColor</b>	标记点边界颜色	<b>none   {auto}   flat   ColorSpec</b>
<b>MarkerFaceColor</b>	闭合的标记点填充颜色	<b>{none}   auto   flat   ColorSpec</b>
<b>MarkerSize</b>	标记点大小	<b>size in points</b>
<b>MeshStyle</b>	网格类型	<b>{both}   row   column</b>

# 例：mesh() / surf()属性设置

```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x);  
r = sqrt(X.^2+Y.^2)+eps;  
Z = sin(r)./r;  
h = mesh(X,Y,Z,'EdgeColor','black');  
set(h,'FaceColor','r');  
set(h,'LineWidth',2);
```



```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x);  
r = sqrt(X.^2+Y.^2)+eps;  
Z = sin(r)./r;  
surf(X,Y,Z,'EdgeColor','none');
```



例：用曲面图表现函数 $z=x.^2+y.^2$ 。

Clf

```
x=-4:4; y=x;[
```

```
X,Y]=meshgrid(x,y); %生成 x-y 坐标“网格点”矩阵
```

```
Z=X.^2+Y.^2;
```

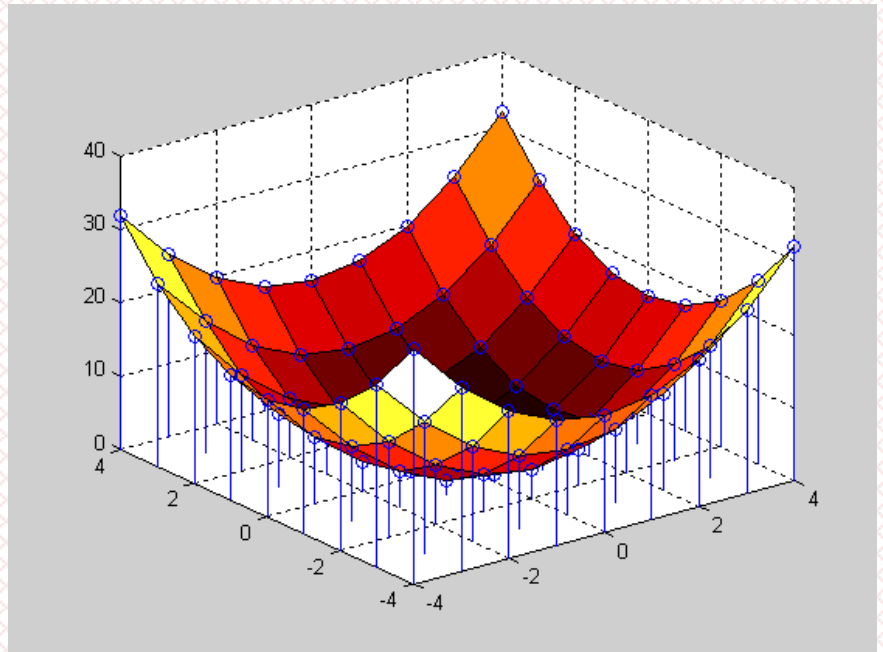
```
surf(X,Y,Z);
```

```
hold on
```

```
colormap(hot)
```

```
stem3(X,Y,Z,'bo')
```

```
%用来表现在格点上计算函数值
```

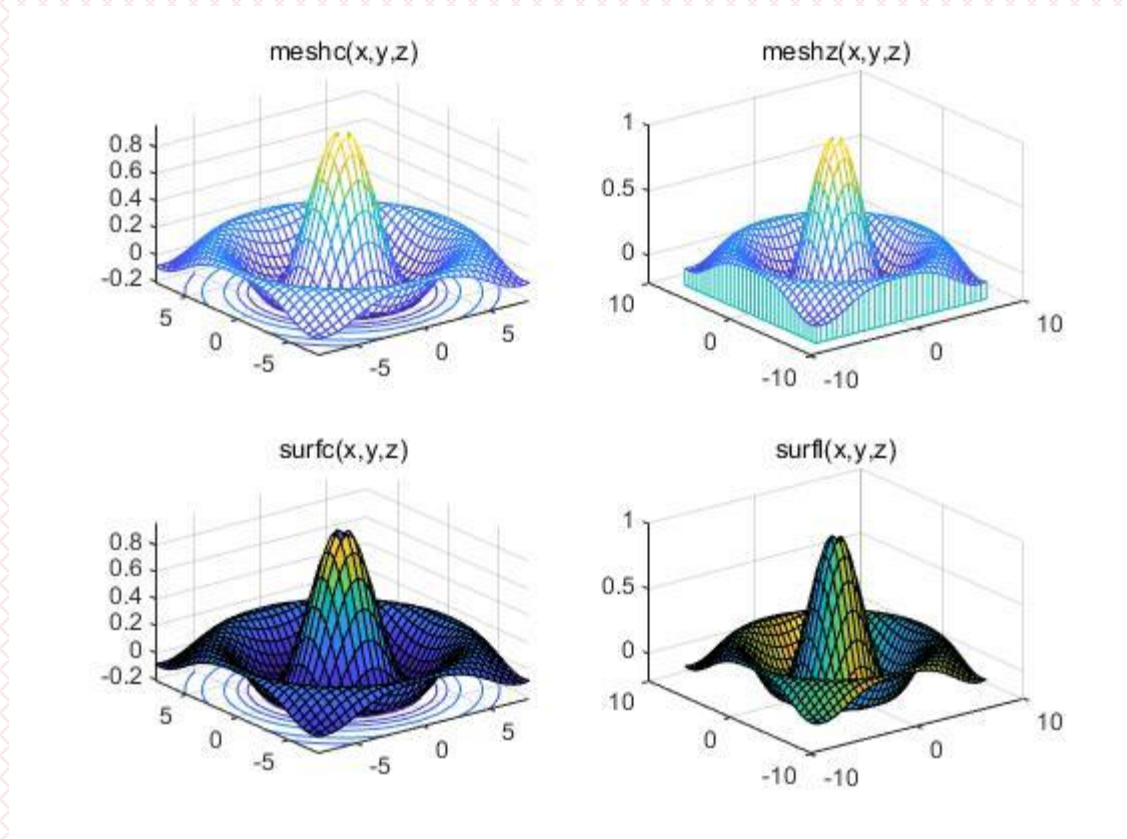




例：在xy平面内选择区域 $[-8,8] \times [-8,8]$ ，绘制函数的4种三维曲面图。

程序如下：

```
[x,y]=meshgrid(-8:0.5:8);  
z=sin(sqrt(x.^2+y.^2))./sqrt(x.^2+y.^2+eps);  
subplot(2,2,1);  
meshc(x,y,z);  
title('meshc(x,y,z)')  
subplot(2,2,2);  
meshz(x,y,z);  
title('meshz(x,y,z)')  
subplot(2,2,3);  
surfc(x,y,z);  
title('surfc(x,y,z)')  
subplot(2,2,4);  
surfl(x,y,z);  
title('surfl(x,y,z)')
```



surfl画的三维曲面有光照效果



## 2.2.5 图形修饰方法

### 图形颜色的修饰

- **matlab** 有极好的颜色表现功能，其颜色数据又构成了一维新的数据集合，也可称为四维图形
- **colormap(MAP)** —— 色图设定函数，**MAP**为  $m \times 3$  维色图矩阵
- 图形颜色可根据需要任意生成，也可用**matlab** 配备的色图函数



matlab 的色图函数:

hsv —— 饱和度色图

gray —— 线性灰度色图

hot —— 暖色色图

cool —— 冷色色图

bone —— 兰色调灰色图

copper —— 铜色色图

pink —— 粉红色色图

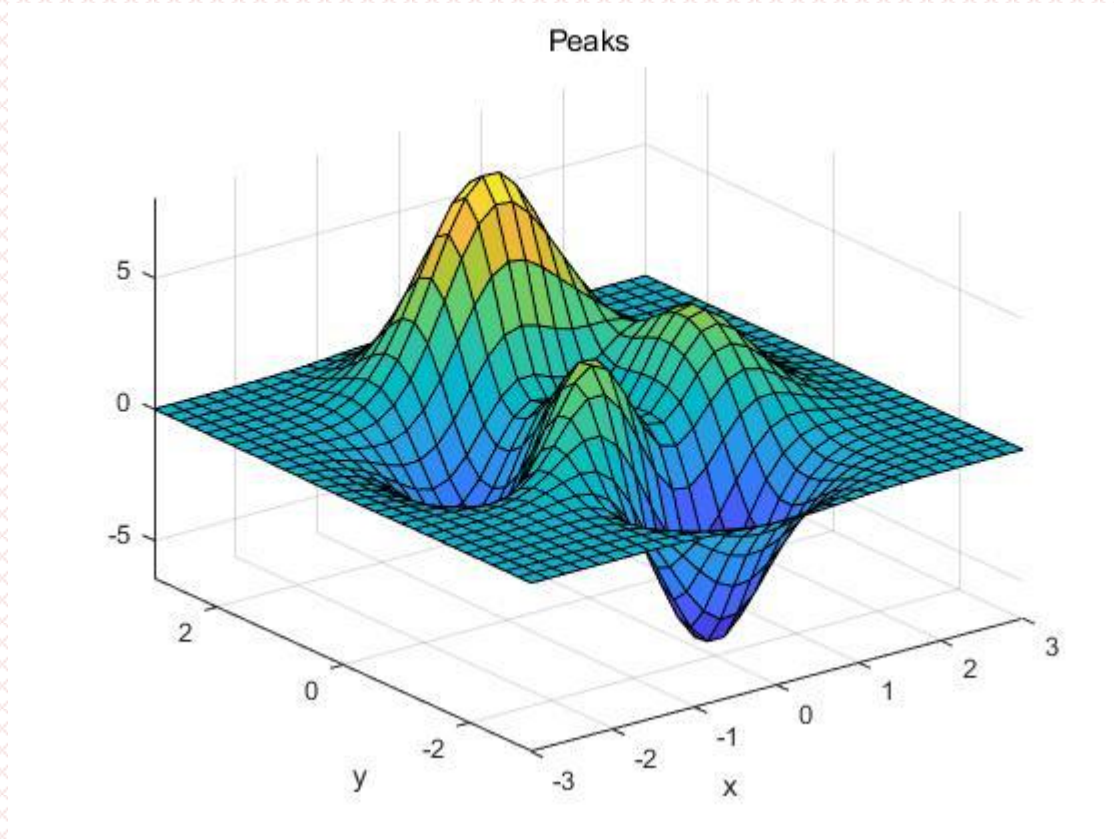
prism —— 光谱色图

jet —— 饱和度色图II

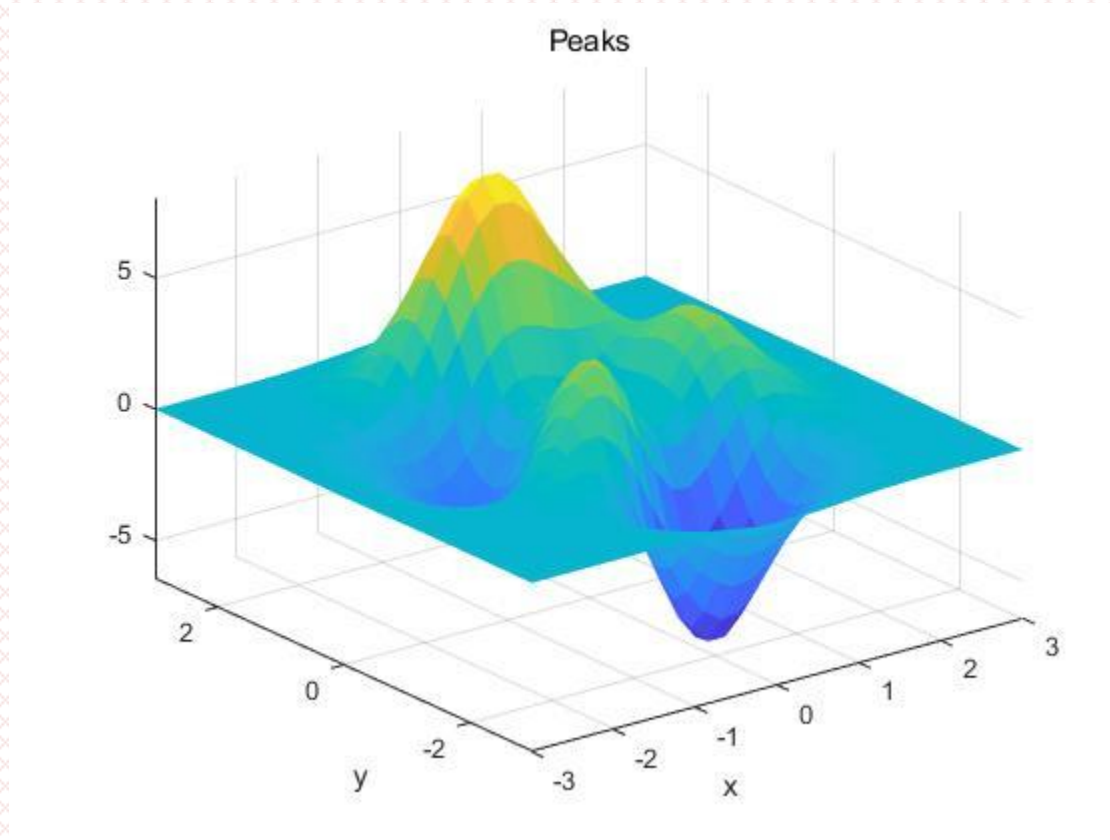
flag —— 红、白、蓝交替色图



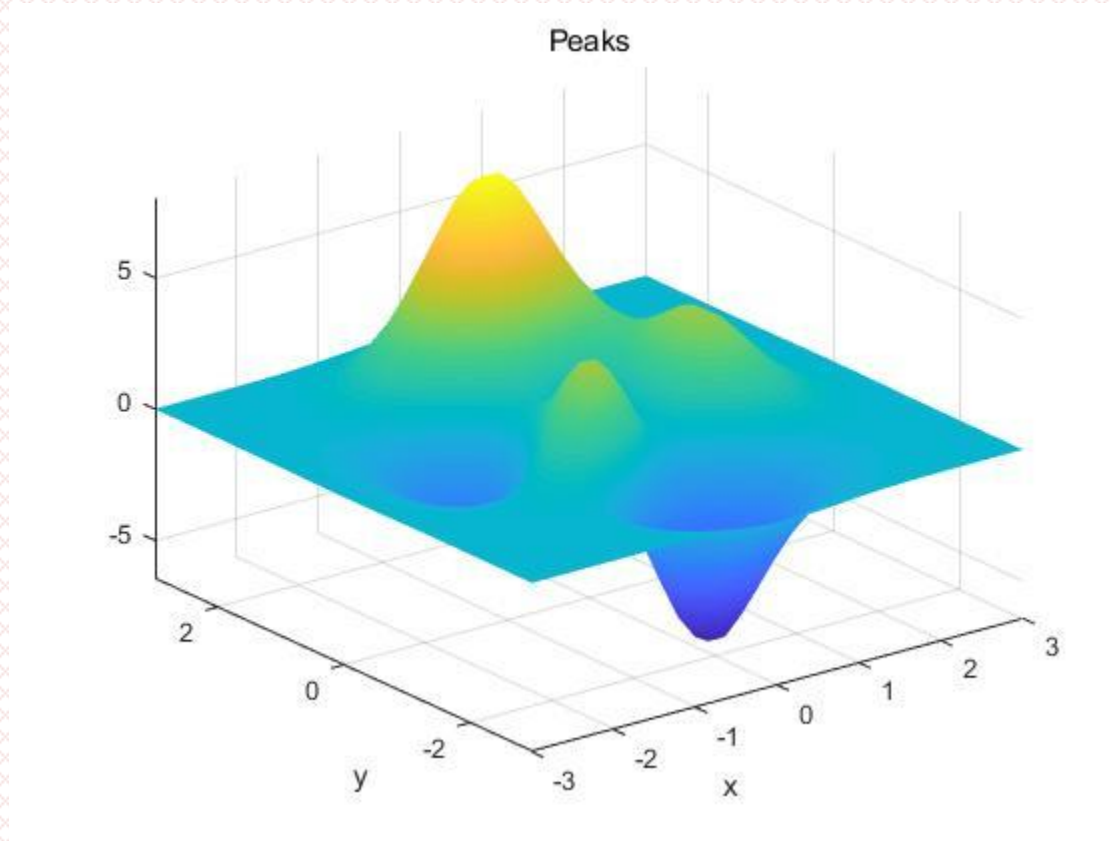
- **shading faceted** — 网格修饰，缺省方式



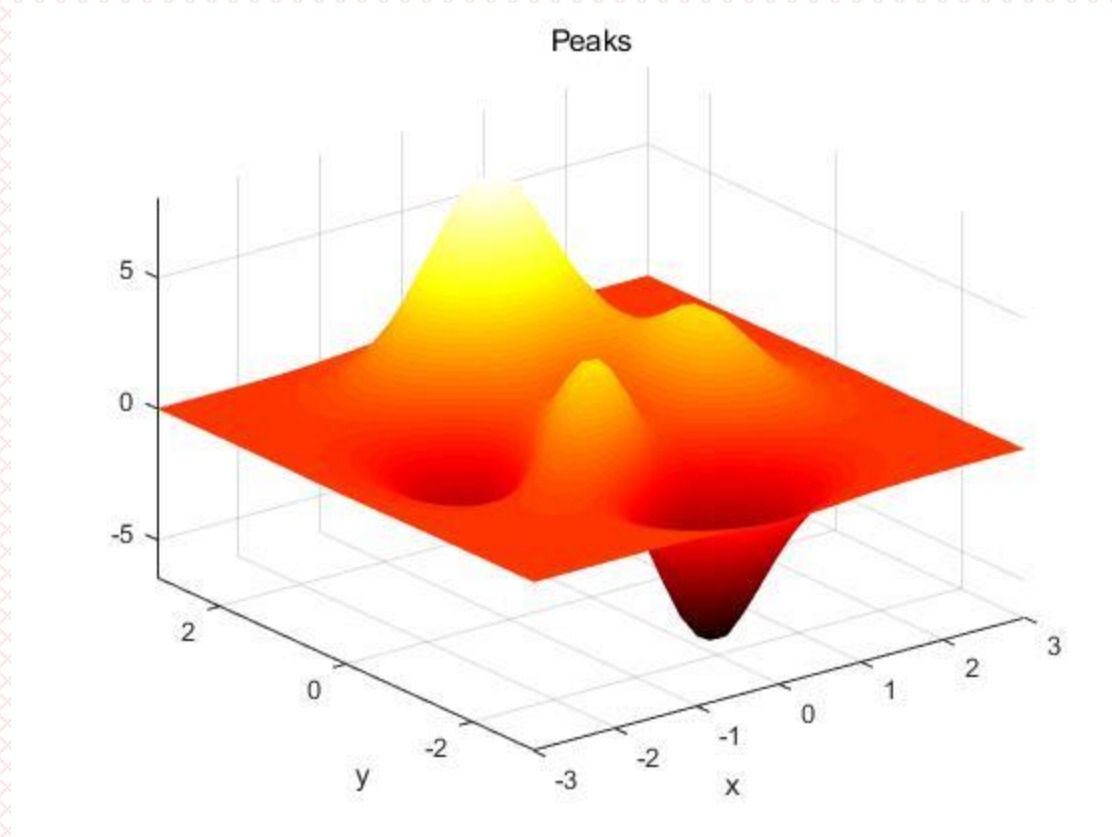
- **shading flat** —— 去掉黑色线条，根据小方块的值确定颜色



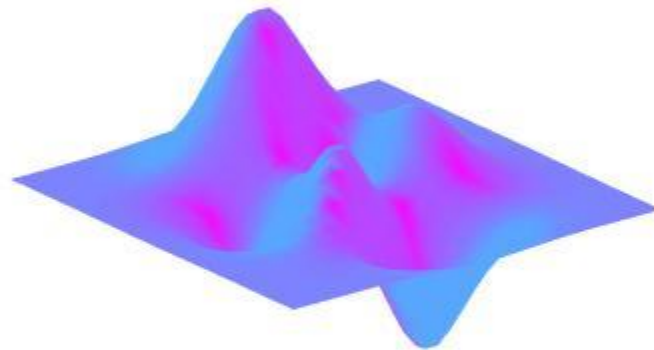
- **shading interp** —— 颜色整体改变，根据小方块四角的值差补过度点的值确定颜色



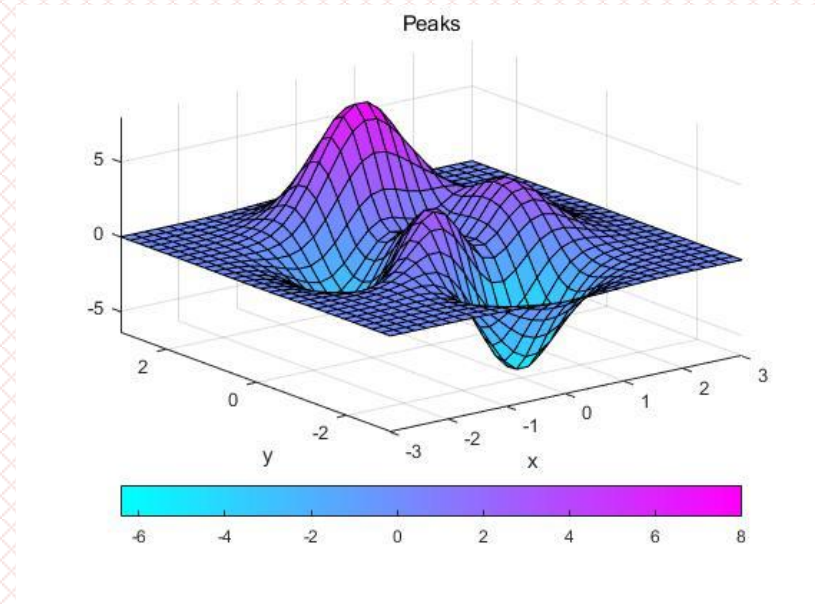
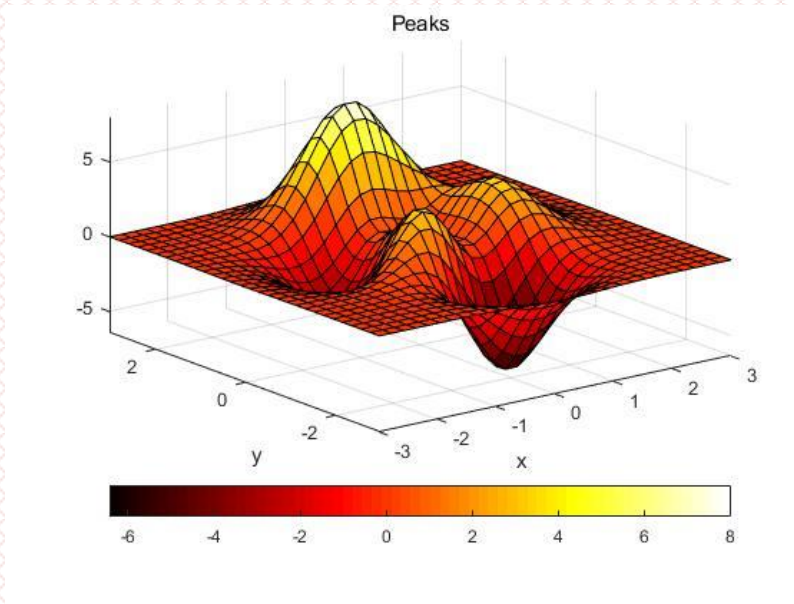
```
peaks(30);shading interp;colormap(hot)
```



```
[X,Y,Z]=peaks(30); surf(X,Y,Z)  
shading interp; colormap(cool); axis off
```



```
peaks(30); colormap(hot); colorbar('horiz')  
figure(2); peaks(30); colormap(cool); colorbar('horiz')
```



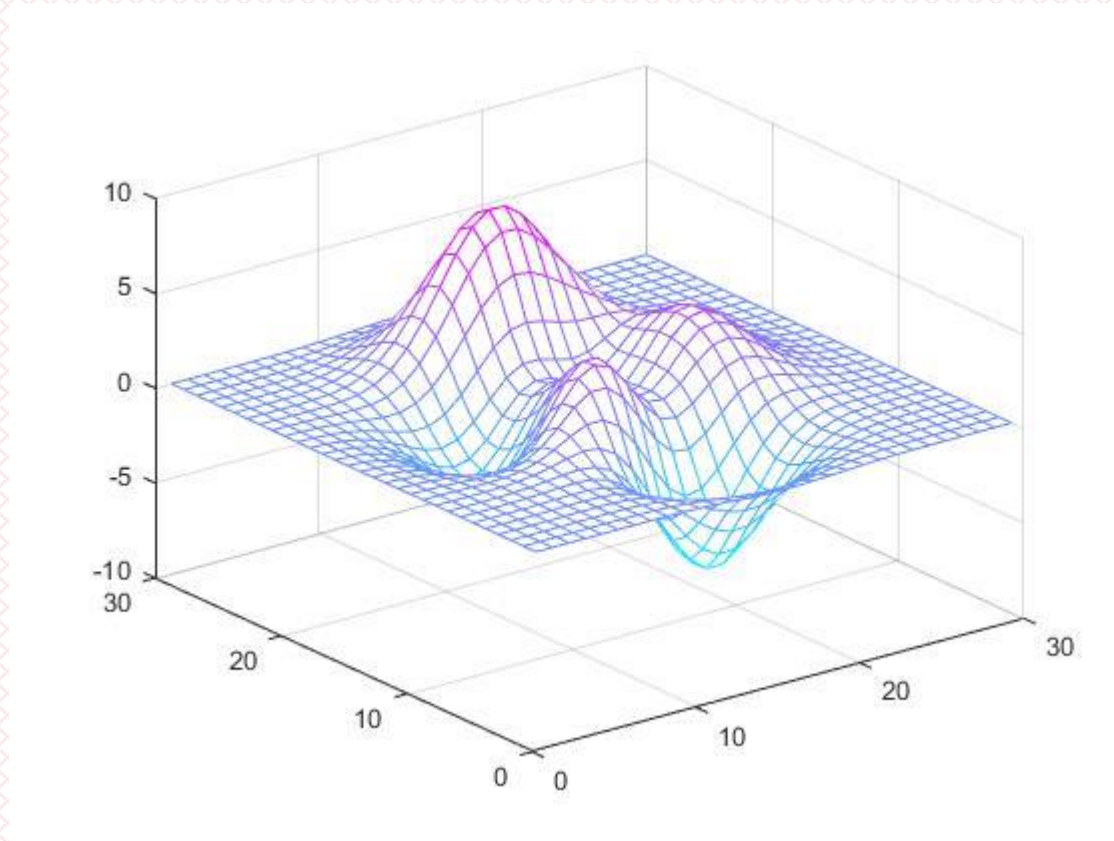
## ② 图形效果修饰

- 透视与消隐——用于网线图
- 裁剪修饰——用于网线图、曲面图
- 视角修饰——观察不同角度的三维视图
- 其它修饰：
  - a. 水线修饰
  - b. 等高线修饰



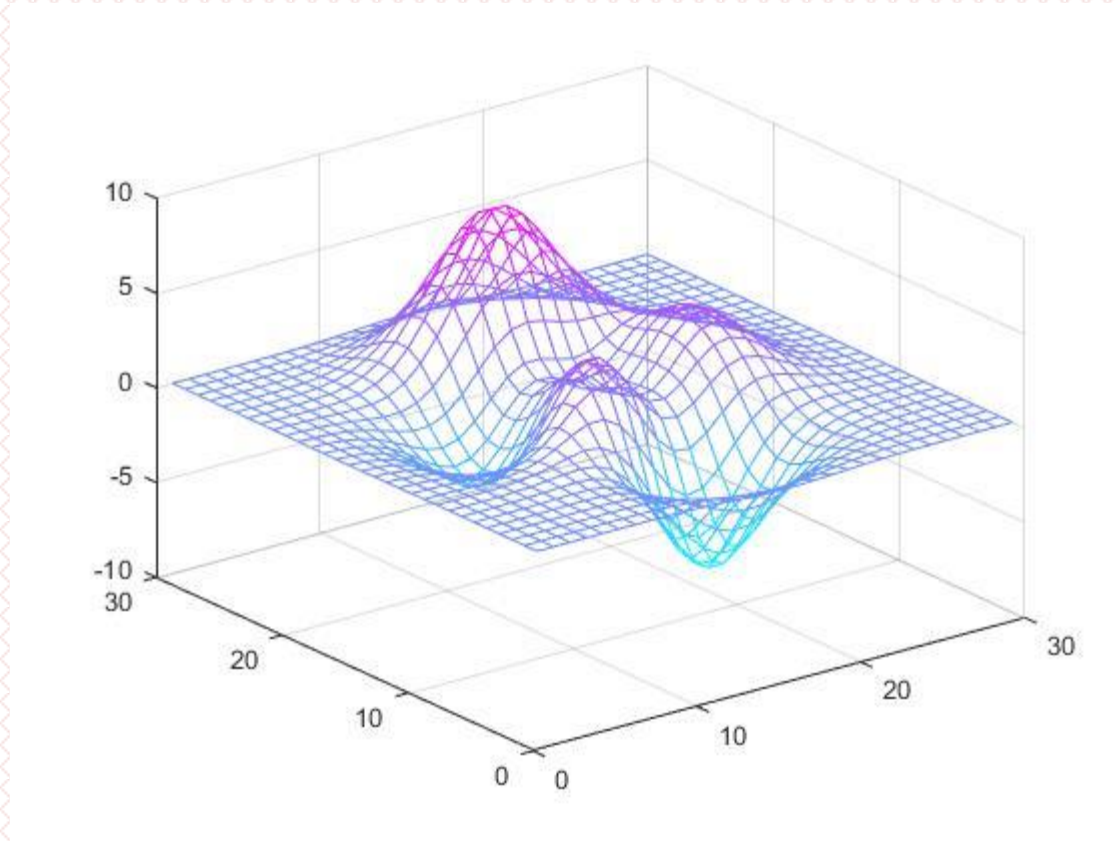
- 透视与消隐

`p=peaks(30); mesh(p);` **hidden on**





`p=peaks(30);mesh(p);` **hidden off**

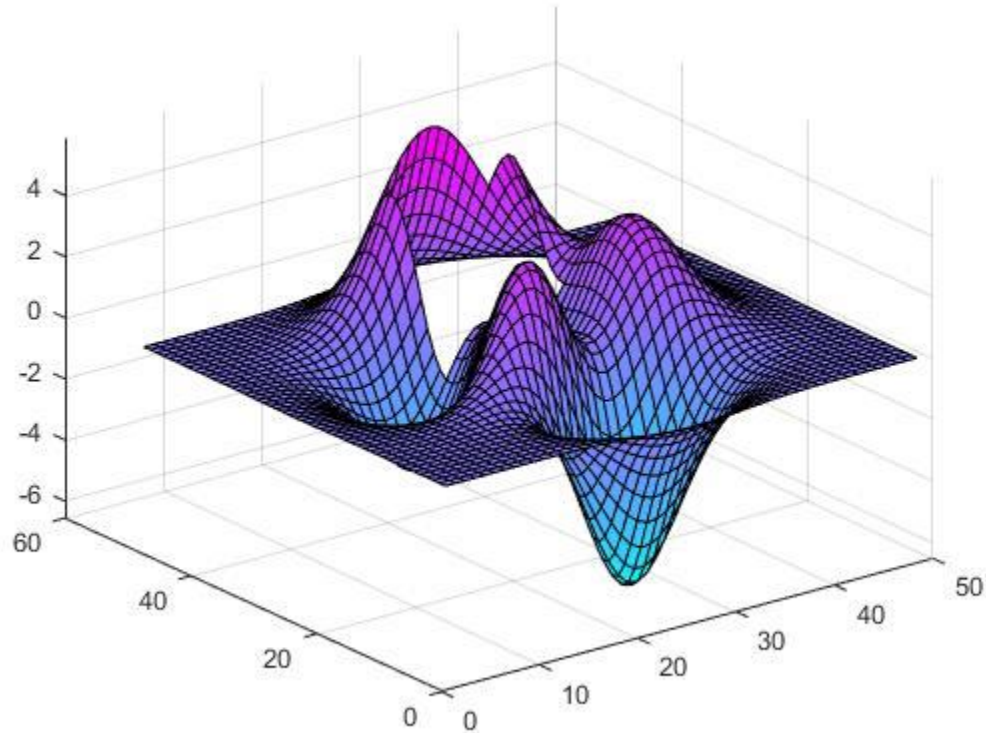


- 裁减修饰

**p=peaks;**

**p(30:40,20:30)=nan\*p(30:40,20:30);**

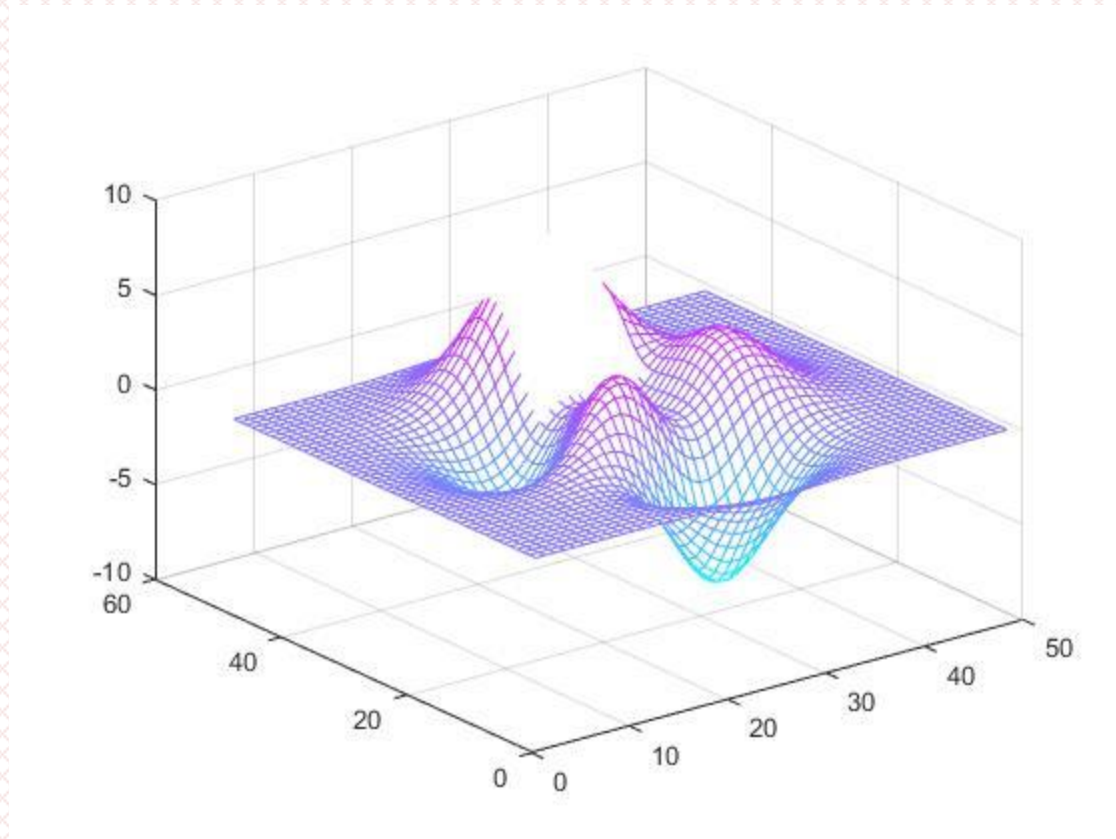
**surf(p)**



```
p=peaks;
```

```
p(30:40,20:30)=nan*p(30:40,20:30);
```

```
mesh(peaks,p)
```





- 视角修饰 (函数view(az,el))

az ----方位角; el ---- 俯视角

缺省值为: az=-37.5; el=30

例: 观察不同视角的波峰图形

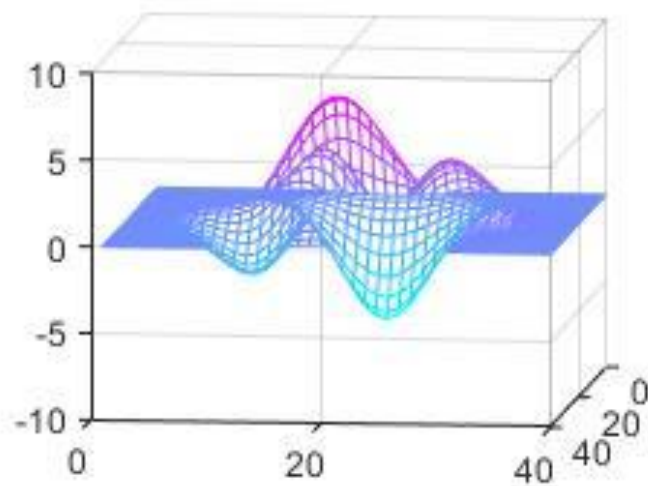
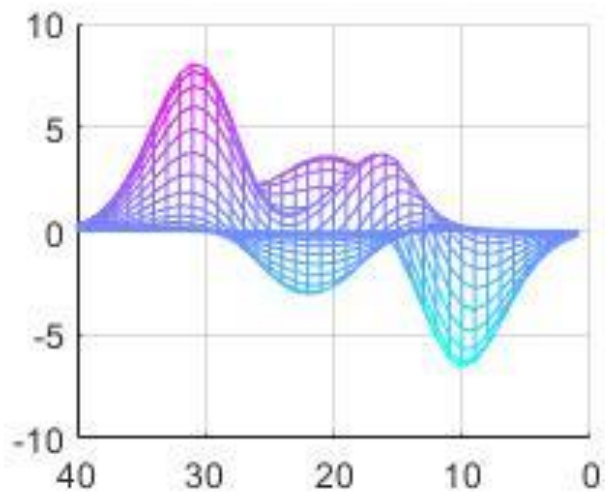
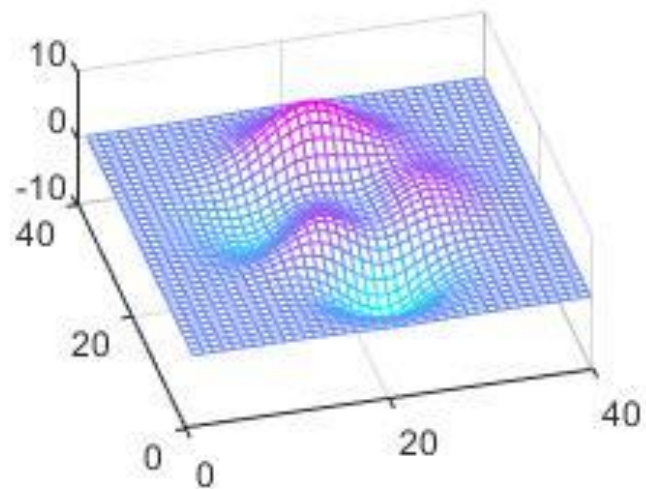
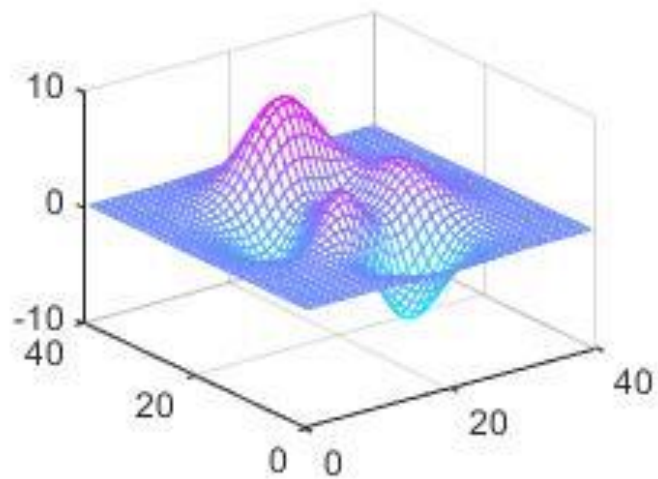
```
z=peaks(40);
```

```
subplot(2,2,1);mesh(z);
```

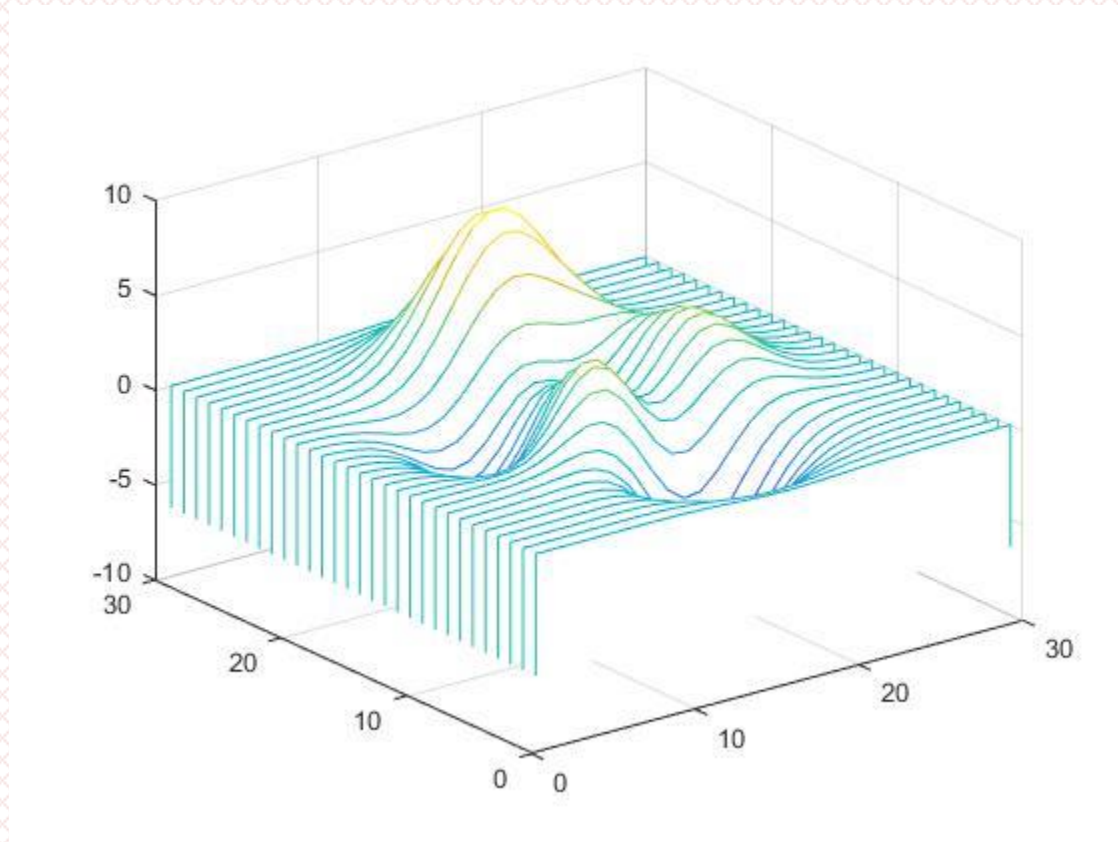
```
subplot(2,2,2);mesh(z);view(-15,60);
```

```
subplot(2,2,3);mesh(z);view(-90,0);
```

```
subplot(2,2,4);mesh(z);view(-7,-10);
```



- 其它修饰:
    - a. 水线修饰 (waterfall)
- waterfall(peaks(30))**







## **b. 等高线修饰**

- **二维**

**contour(Z,n)-----绘制n条等高线**

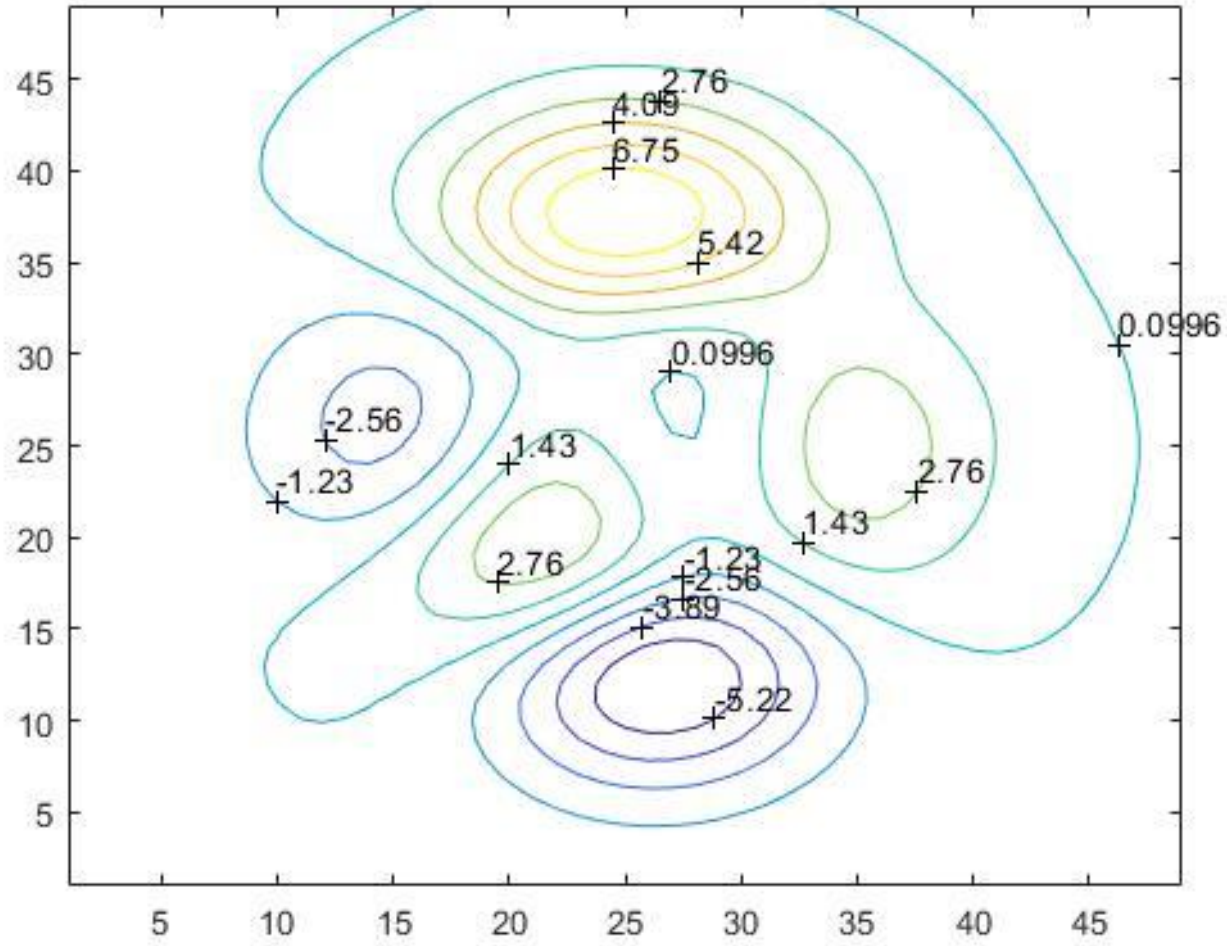
**C= contourc(Z,n)-----计算n条等高线的坐标**

**Clable(c)-----给等高线加标注**

**例、在二维平面上绘制peaks函数的10条等高线**

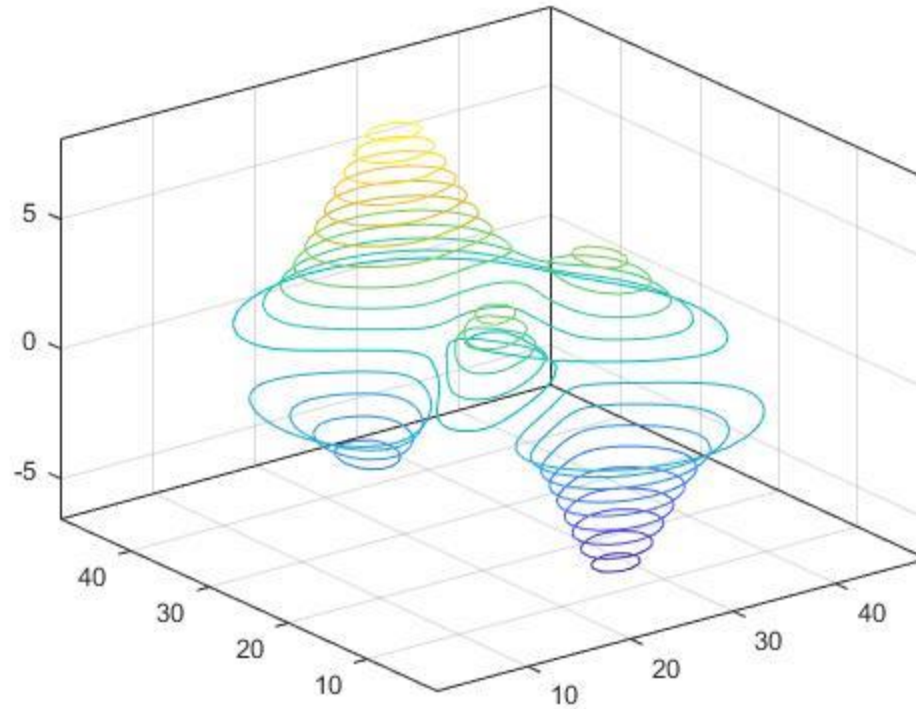
**contour(peaks,10);**

**C=contourc(peaks,10); clabel(C)**





`contour3(peaks,20)`





## 动画

**Moviein(),getframe,movie()**

函数**Moviein()**产生一个帧矩阵来存放动画中的帧；函数**getframe**对当前的图像进行快照；函数**movie()**按顺序回放各帧。

例 将曲面**peaks**做成动画。

解 **[x,y,z]=peaks(30);**

**surf(x,y,z)**

**axis([-3 3 -3 3 -10 10])**

**m=moviein(15);**

**for i=1:15**

**view(-37.5+24\*(i-1),30)**

**m(:,i)=getframe;**

**end**

**movie(m)**



## 2.2.6 特殊的三维图形函数

### 1、空间等值线图: **contour3(x,y,z,n)**

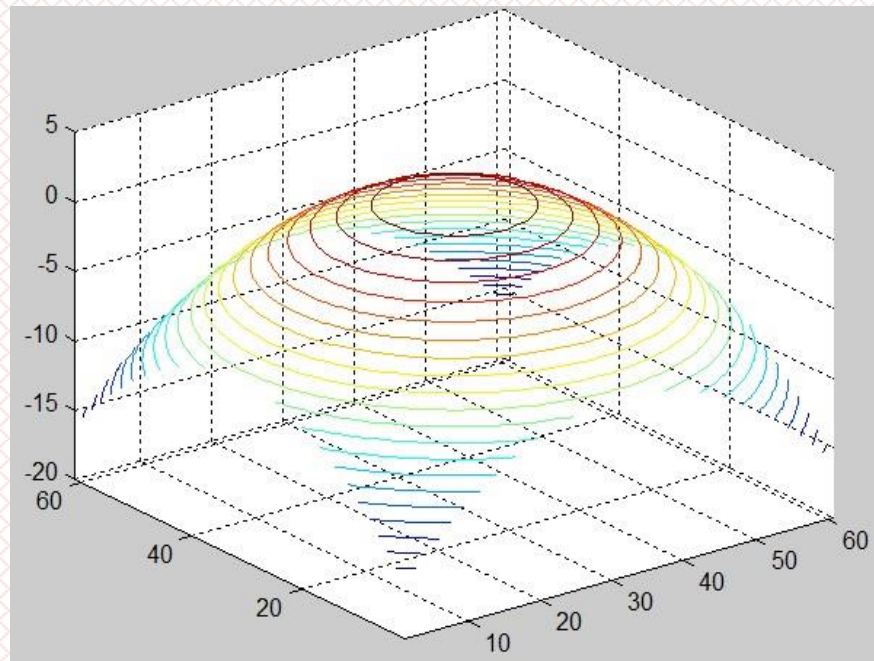
其中n表示等值线数。

例： 山峰的三维和二维等值线图。

```
解: [x,y,z]=peaks;  
subplot(1,2,1)  
contour3(x,y,z,16,'s')  
grid, xlabel('x-axis'),ylabel('y-axis')  
zlabel('z-axis')  
title('contour3 of peaks');  
subplot(1,2,2)  
contour(x,y,z,16,'s')  
grid, xlabel('x-axis'), ylabel('y-axis')  
title('contour of peaks');
```

# 1、空间等值线图: **contour 3(x,y,z,n)**

```
>> [x,y]=meshgrid(-3:0.1:3);  
>> z=2-x.^2-y.^2;  
>> contour3(z,20)
```





**2、三维散点图 `scatter3 (x,y,z,s,c)`**  
在向量**x**,**y**和**z**指定的位置上显示彩色圆圈。  
向量**x**,**y**和**z**的大小必须相同。

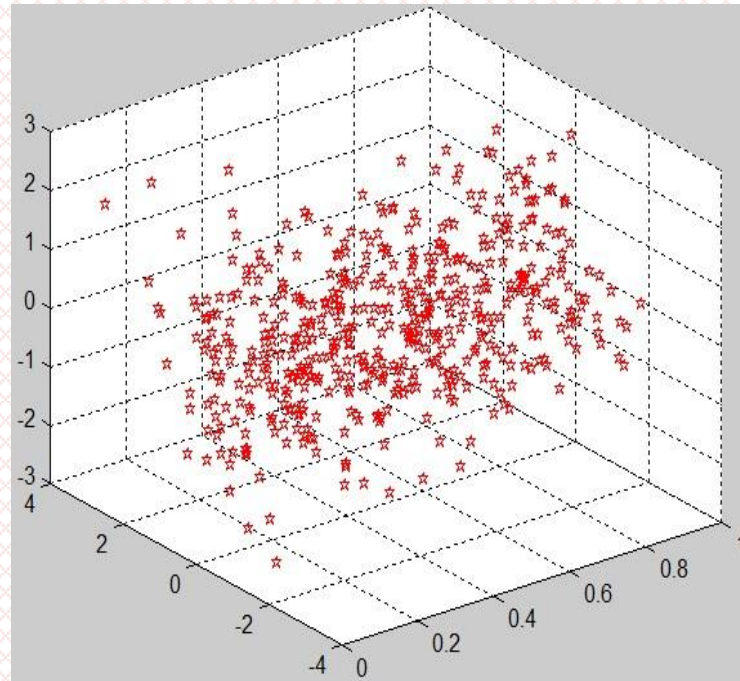
**例** 绘制三维散点图。

**解** 输入命令：

```
[x,y,z]=sphere(16);  
X=[x(:)*.5 x(:)*.75 x(:)];  
Y=[y(:)*.5 y(:)*.75 y(:)];  
Z=[z(:)*.5 z(:)*.75 z(:)];  
S= repmat([1 .75 .5]*10,prod(size(x)),1);  
C= repmat([1 2 3],prod(size(x)),1);  
scatter3(X(:),Y(:),Z(:),S(:),C(:),'filled'),view(-60,60)
```

2、三维散点图 `scatter3 (x,y,z,s,c)`  
在向量 $x$ ,  $y$ 和 $z$ 指定的位置上显示彩色圆圈。  
向量 $x$ ,  $y$ 和 $z$ 的大小必须相同。

```
>> x=rand(500,1);  
>> y=randn(500,1);  
>> z=randn(500,1);  
>> scatter3(x,y,z,'p','r')
```

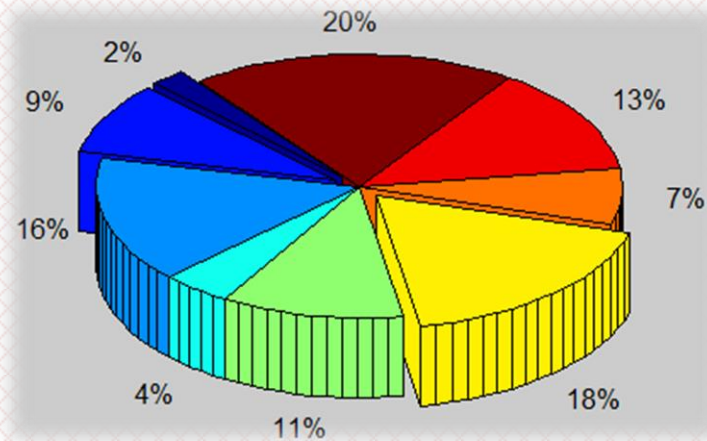




### 3、三维饼图： **pie3( )**

例、绘制三维饼图。

```
>> A=[1 2 3;4 5 6;7 8 9];  
>> ex=[1 0 0;4 0 0;0 8 0];  
>> pie3(A,ex)
```

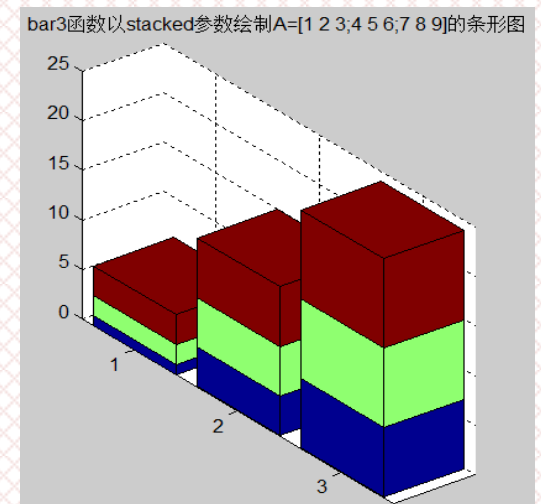
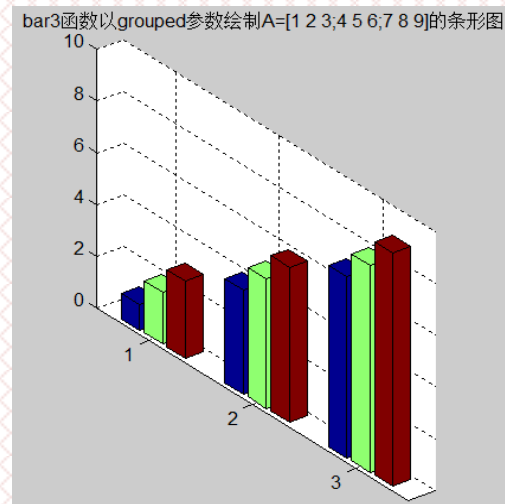
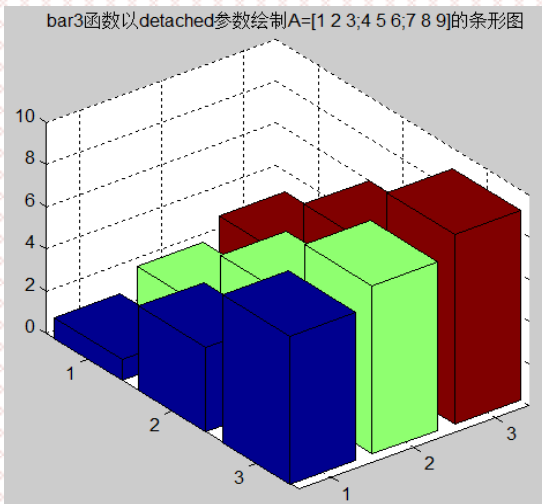




## 4、三维条形图: **bar3()**

在各种**style**参数的条件下绘制矩阵的三维条形图。

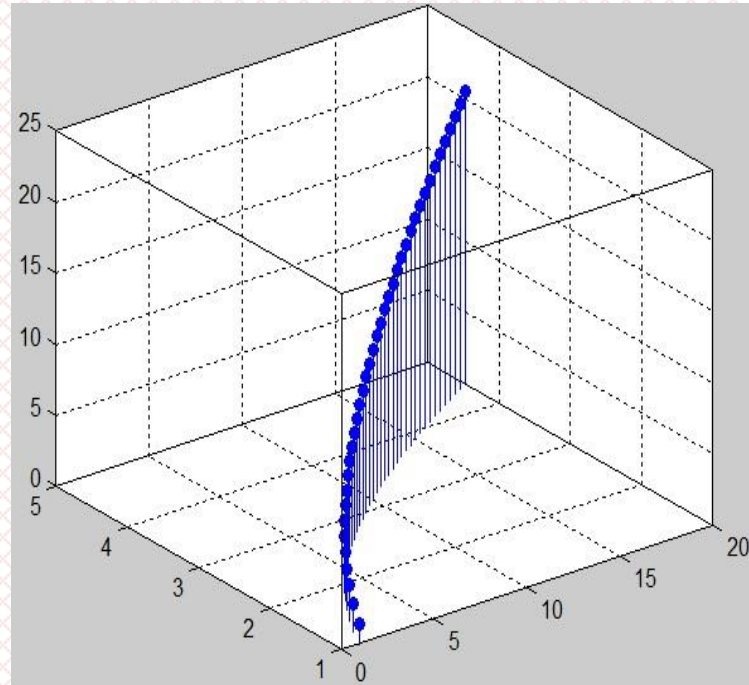
```
z=[1 2 3;4 5 6;7 8 9];  
bar3(z,'detached')  
title('bar3函数以detached参数绘制A=[1 2 3;4 5 6;7 8 9]的条形图')  
bar3(z,'grouped')  
title('bar3函数以grouped参数绘制A=[1 2 3;4 5 6;7 8 9]的条形图')  
bar3(z,'stacked')  
title('bar3函数以stacked参数绘制A=[1 2 3;4 5 6;7 8 9]的条形图')
```



## 5、三维火柴杆图: **stem3()**

绘制三维火柴杆图。

```
>> x=[1:0.5:20];  
>> y=sqrt(x);  
>> z=sqrt(x.^2+y.^2);  
>> stem3(x,y,z,'filled')
```



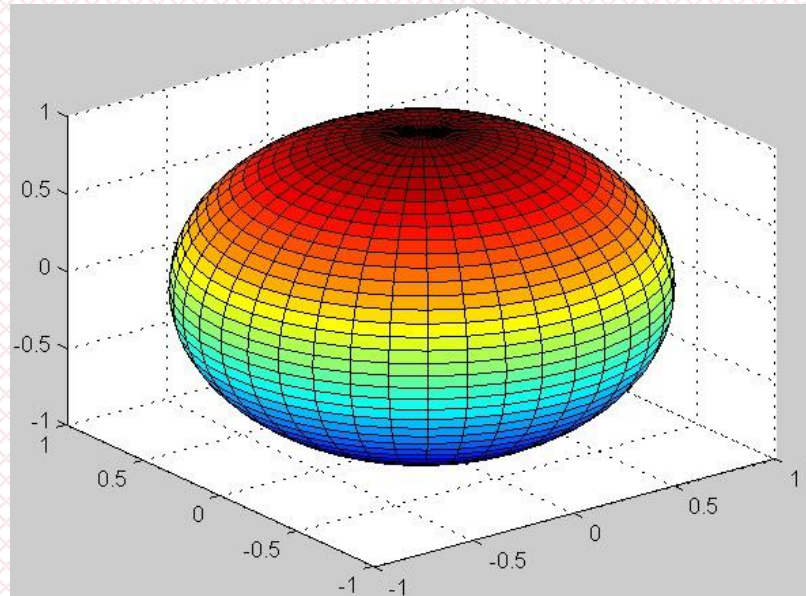
## 6、球面: **sphere()**

**sphere**函数的调用格式为:

**[x,y,z]=sphere(n)**

生成由 $n*n$ 个面组成的一个球面; **n** 省略时, 生成由**20\*20**个面组成的一个球面。

**[x,y,z]=sphere(40);**  
**surf(x,y,z)**





## 7、柱面: **cylinder()**

**[X,Y,Z]=cylinder** 返回一半径为1、高度为1的圆柱面的x-, y-, z-轴的坐标值, 圆柱面的圆周有20个距离相同的点。

**[X,Y,Z]=cylinder(r)** 返回一半径为r、高度为1的柱面的x-, y-, z-轴的坐标值, 柱面的圆周有20个距离相同的点。

**[X,Y,Z]=cylinder(r,n)** 返回一半径为r、高度为1的柱面的x-, y-, z-轴的坐标值, 圆柱面的圆周有指定的n个距离相同的点

**cylinder(...)** 画出柱面。

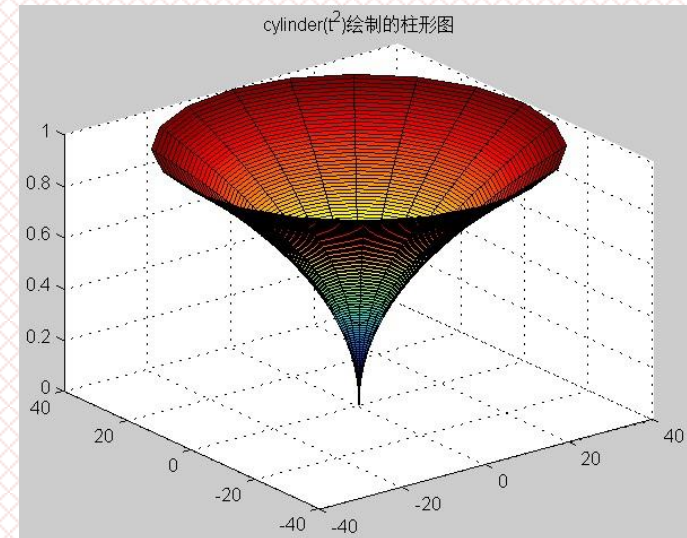
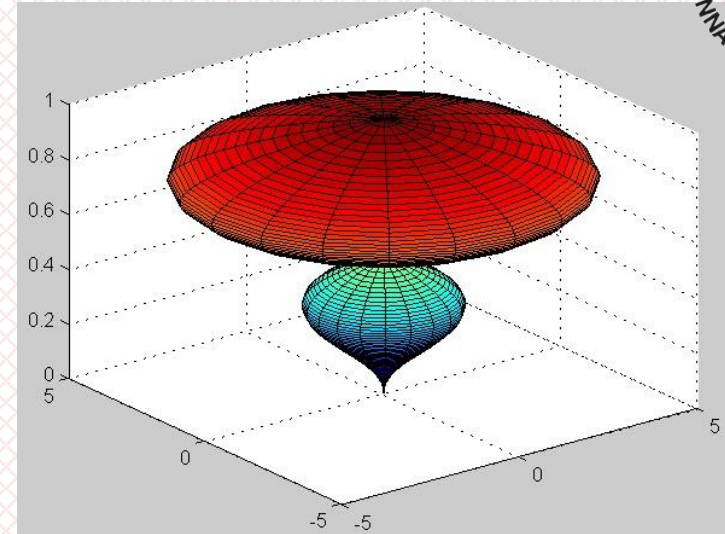
```
t = 0:pi/10:2*pi;  
[X,Y,Z] = cylinder(2+cos(t));  
surf(X,Y,Z)  
axis square
```

## 7、柱面: **cylinder()**

**cylinder**函数的调用格式为:  
**[x,y,z]= cylinder(R,n)**

绘制柱形图。

```
>> t=[0:pi/50:2*pi];  
>> [x,y,z]=cylinder(t.*sin(t));  
>> surf(x,y,z)  
>> cylinder(t.^2)  
>> title('cylinder(t^2)绘制的柱形图')
```







## 8、椭球面: **ellipsoid ( )**

**Matlab**提供的绘制椭球面的函数**ellipsoid**

**[x,y,z] = ellipsoid(xc,yc,zc, a,b,c,n)**

返回绘图数据，x，y，z均为 $(n+1) \times (n+1)$ 的二维数组

**[x,y,z] = ellipsoid(xc,yc,zc,a,b,c)**

返回绘图数据， $n = 20$

**ellipsoid(axes\_handle,...)**在指定坐标轴画出椭球面

**ellipsoid(...)** 画出椭球面

$$\frac{(x - xc)^2}{a^2} + \frac{(y - yc)^2}{b^2} + \frac{(z - zc)^2}{c^2} = 1$$

```
[x, y, z] = ellipsoid(0,0,0,3,2,1,30);  
surf(x, y, z)  
axis equal
```





## 2.2.7 应用

例1： 用三维曲面图表现函数 $z=\sin(y)\cos(x)$ 。

程序1：

```
x=0:0.1:2*pi;[x,y]=meshgrid(x);z=sin(y).*cos(x);  
mesh(x,y,z);xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis');  
title('mesh');
```

程序2：

```
x=0:0.1:2*pi;[x,y]=meshgrid(x);z=sin(y).*cos(x);  
surf(x,y,z);xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis');  
title('surf');
```

程序3：

```
x=0:0.1:2*pi;[x,y]=meshgrid(x);z=sin(y).*cos(x);  
plot3(x,y,z);xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis');  
title('plot3-1');grid;
```



## 例2： 绘制两个直径相等的圆管的相交图形。

程序如下：

```
%两个等直径圆管的交线
m=30;
z=1.2*(0:m)/m;
r=ones(size(z));
theta=(0:m)/m*2*pi;
x1=r'*cos(theta);y1=r'*sin(theta);    %生成第一个圆管的坐标矩阵
z1=z'*ones(1,m+1);
x=(-m:2:m)/m;
x2=x'*ones(1,m+1);y2=r'*cos(theta);    %生成第二个圆管的坐标矩阵
z2=r'*sin(theta);
surf(x1,y1,z1);                        %绘制竖立的圆管
axis equal,axis off
hold on
surf(x2,y2,z2);                        %绘制平放的圆管
axis equal,axis off
title('两个等直径圆管的交线');
hold off
```



**例3： 分析由函数 $z=x^2-2y^2$ 构成的曲面形状及与平面 $z=a$ 的交线。**

**程序如下：**

```
[x,y]=meshgrid(-10:0.2:10);  
z1=(x.^2-2*y.^2)+eps;           %第1个曲面  
a=input('a=?'); z2=a*ones(size(x)); %第2个曲面  
subplot(1,2,1);mesh(x,y,z1);hold on;mesh(x,y,z2); %分别画出两个曲面  
v=[-10,10,-10,10,-100,100];axis(v);grid;      %第1子图的坐标设置  
hold off;  
r0=abs(z1-z2)<=1;                %求两曲面z坐标差小于1的点  
xx=r0.*x; yy=r0.*y; zz=r0.*z2; %求这些点上的x,y,z坐标，即交线坐标  
subplot(1,2,2);  
plot3(xx(r0~=0),yy(r0~=0),zz(r0~=0),'*'); %在第2子图画出行交线  
axis(v);grid;                    %第2子图的坐标设置
```



# 例4：绘制山区地貌图

要在某山区方圆大约27平方公里范围内修建一条公路，从山脚出发经过一个居民区，再到达一个矿区。横向纵向分别每隔400米测量一次，得到一些地点的高程：（平面区域 $0 \leq x \leq 5600$ ,  $0 \leq y \leq 4800$ ），需作出该山区的地貌图和等高线图。

3600	1480	1500	1550	1510	1430	1300	1200	980
3200	1500	1550	1600	1550	1600	1600	1600	1550
2800	1500	1200	1100	1550	1600	1550	1380	1070
2400	1500	1200	1100	1350	1450	1200	1150	1010
2000	1390	1500	1500	1400	900	1100	1060	950
1600	1320	1450	1420	1400	1300	700	900	850
1200	1130	1250	1280	1230	1040	900	500	700
Y/x	1200	1600	2000	2400	2800	3200	3600	4000



```
x=1200:400:4000;  
y=1200:400:3600;  
z=[1130 1250 1280 1230 1040 900 500 700  
1320 1450 1420 1400 1300 700 900 850  
1390 1500 1500 1400 900 1100 1060 950  
1500 1200 1100 1350 1450 1200 1150 1010  
1500 1200 1100 1550 1600 1550 1380 1070  
1500 1550 1600 1550 1600 1600 1600 1550  
1480 1500 1550 1510 1430 1300 1200 980];  
[X,Y]=meshgrid(x,y);  
mesh(x,y,z)
```

例5. 绘制曲线

$$\begin{cases} z = \sqrt{1-x^2-y^2} \\ (x-\frac{1}{2})^2 + y^2 = (\frac{1}{2})^2 \end{cases}$$

首先我们把方程组变化成以下形式:

$$\begin{cases} x = t \\ y = \sqrt{t(1-t)} \\ z = \sqrt{1-x^2-y^2} \end{cases}$$

于是输入:

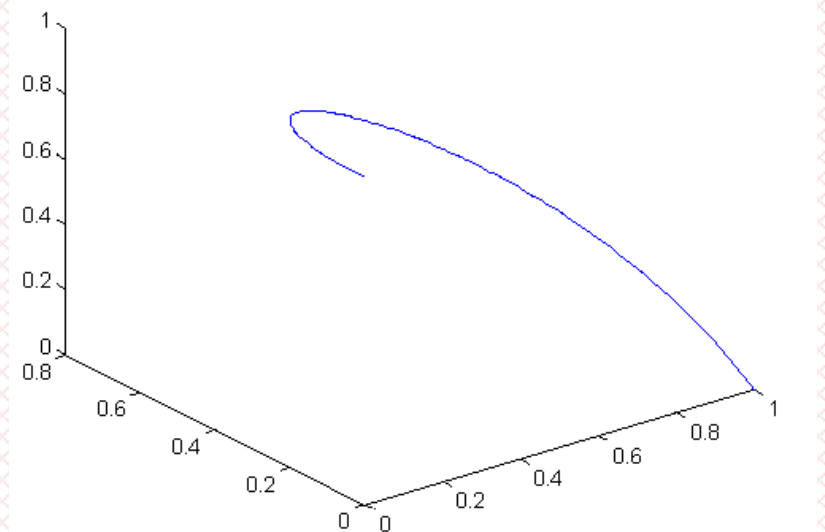
```
>>t=0:0.01:1
```

```
>>x=t;
```

```
>>y=sqrt(t.*(1-t));
```

```
>>z=sqrt(1-x.^2-y.^2);
```

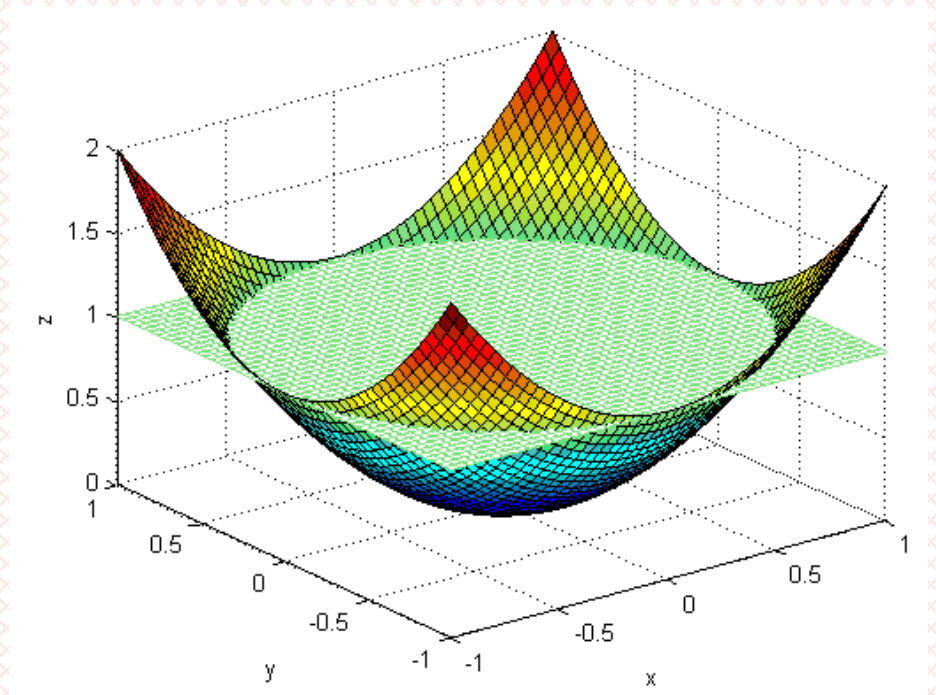
```
>>plot3(x,y,z)
```




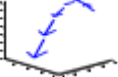
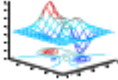


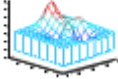
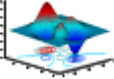
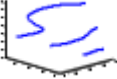


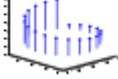
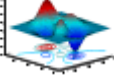



例6. 画出抛物面  $z = x^2 + y^2$  被平面  $z = 1$  所截图形。

```
>> [x,y]=meshgrid(-1:0.04:1);  
>> z1=x.^2+y.^2;z2=ones(size(z1));  
>> surf(x,y,z1)  
>> hold on  
>> mesh(x,y,z2)  
>> xlabel('x');ylabel('y');zlabel('z');
```



# 三维绘图函数汇总

Line Graphs	Mesh Graphs and Bar Graphs	Area Graphs and Constructive Objects	Surface Graphs	Direction Graphs	Volumetric Graphs
<a href="#">plot3</a> 	<a href="#">mesh</a> 	<a href="#">pie3</a> 	<a href="#">surf</a> 	<a href="#">quiver3</a> 	<a href="#">scatter3</a> 
<a href="#">contour3</a> 	<a href="#">meshc</a> 	<a href="#">fill3</a> 	<a href="#">surfl</a> 	<a href="#">comet3</a> 	<a href="#">coneplot</a> 
<a href="#">contourslice</a> 	<a href="#">meshz</a> 	<a href="#">patch</a> 	<a href="#">surfc</a> 	<a href="#">streamslice</a> 	<a href="#">streamline</a> 
<a href="#">ezplot3</a> 	<a href="#">ezmesh</a> 	<a href="#">cylinder</a> 	<a href="#">ezsurf</a> 	<a href="#">streamribbon</a> 	
<a href="#">waterfall</a> 	<a href="#">stem3</a> 	<a href="#">ellipsoid</a> 	<a href="#">ezsurfz</a> 	<a href="#">streamtube</a> 	
	<a href="#">bar3</a> 	<a href="#">sphere</a> 			
	<a href="#">bar3h</a> 				



# 练习题

- **1** 设  $z = xe^{-x^2-y^2}$ ，当 $x$ 和 $y$ 的取值范围均为-2到2时，用建立子窗口的方法在同一个图形窗口中绘制出三维线图、网线图、表面图和带渲染效果的表面图。
- **2.** 绘制三维曲线  $\begin{cases} x = t \\ y = 2t\cos t, (-2\pi \leq t \leq 2\pi) \\ z = 5t\sin t \end{cases}$  的图形。

## Q & A

- 有什么问题吗？

