

## mesh函数：绘制出在某一区间内完整的网格图

**mesh(X,Y,Z)**的用法，其中**X**是**n**维向量,Y是**m**维向量，Z是**m\*n**维的矩阵

```
X = [1,2,4]
```

```
X =  
     1     2     4  
•
```

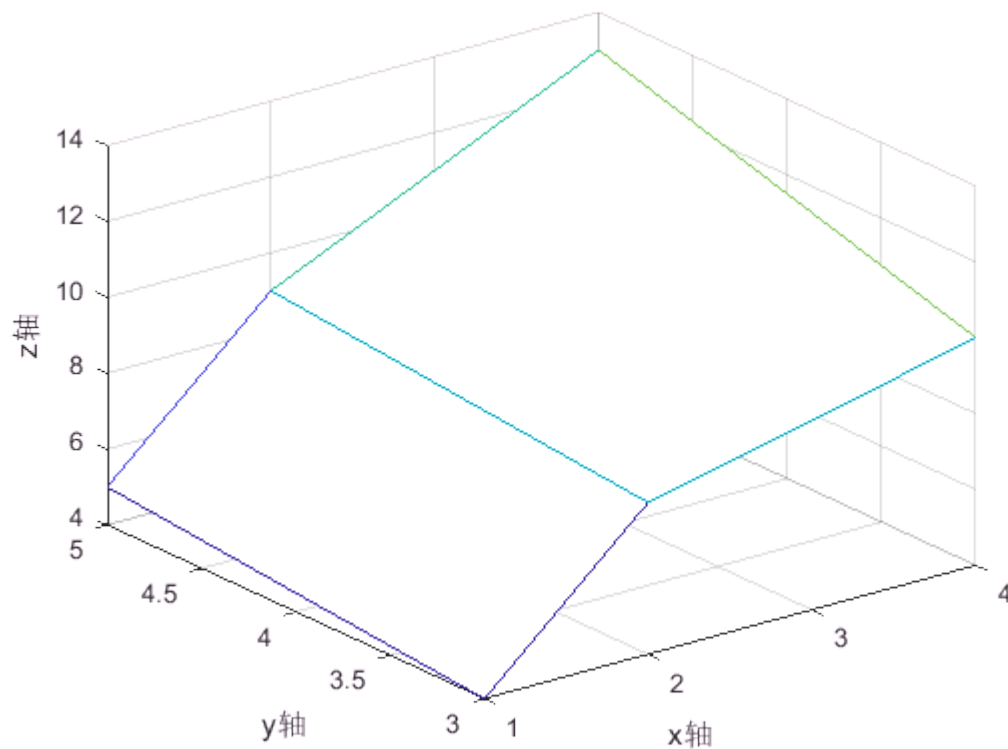
```
Y = [3,5]
```

```
Y =  
     3     5  
•
```

```
Z = [4,8,10;5,9,13]
```

```
Z =  
     4     8    10  
     5     9    13  
•
```

```
mesh(X,Y,Z) % (X(j), Y(i), Z(i,j))是线框网格线的交点的坐标  
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
```



```
% 三维旋转和数据游标的使用，以及X-Y视图的切换（在三维旋转状态下点击鼠标右键）  
% 数据游标默认只能添加一个，按住Alt键不动，可以添加多个数据游标
```

```
% 插入颜色栏，可显示色阶
```

**mesh(Z)**的用法，其中**Z**是**m\*n**维的矩阵

```
Z = [4,8,10;5,9,13]
```

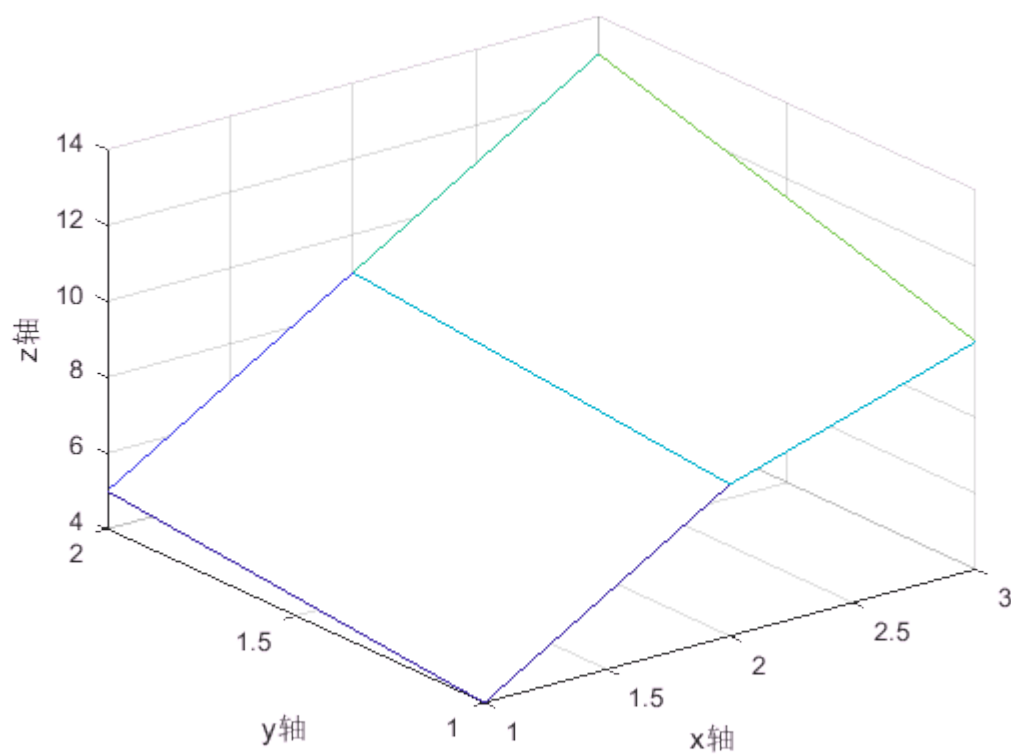
```
Z =
```

```
     4     8    10  
     5     9    13
```

•

```
mesh(Z)
```

```
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
```



```
% 等价于
```

```
X = 1:3
```

```
X =
```

```
     1     2     3
```

```
Y = 1:2
```

```
Y =
```

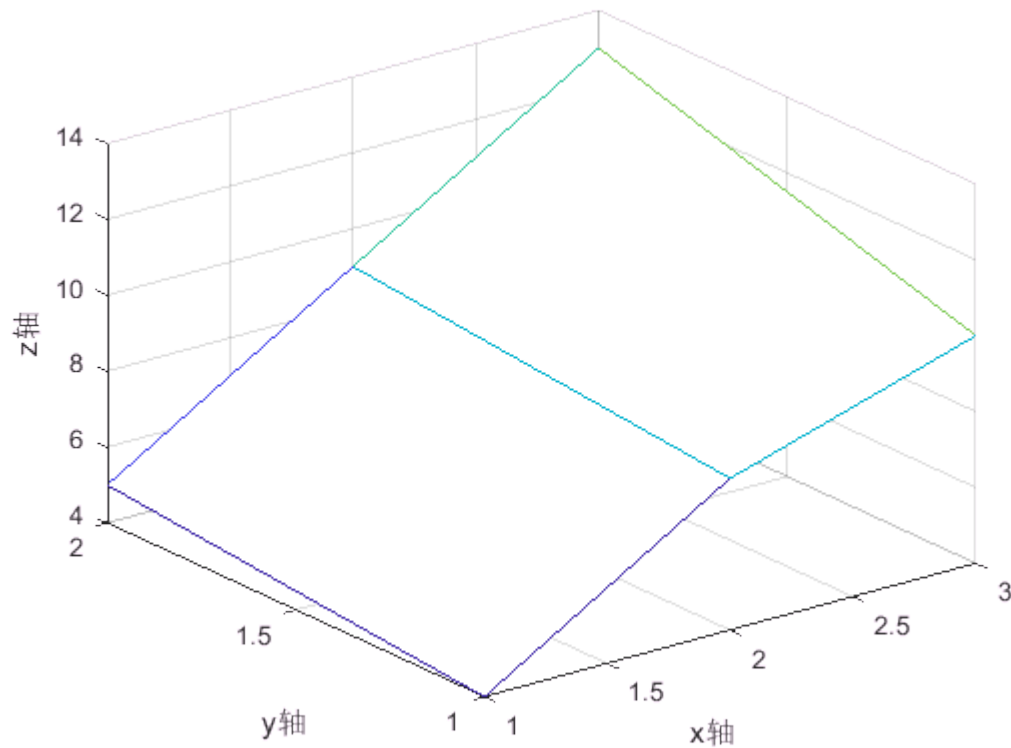
```
     1     2
```

```
Z = [4,8,10;5,9,13]
```

```
Z =
```

4	8	10
5	9	13

```
mesh(X,Y,Z)
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
```



思考：如果**X**中元素不是按照从小到大排序的，图像会是什么样子？

```
X = [1,10,4]
```

```
X =
     1    10     4
```

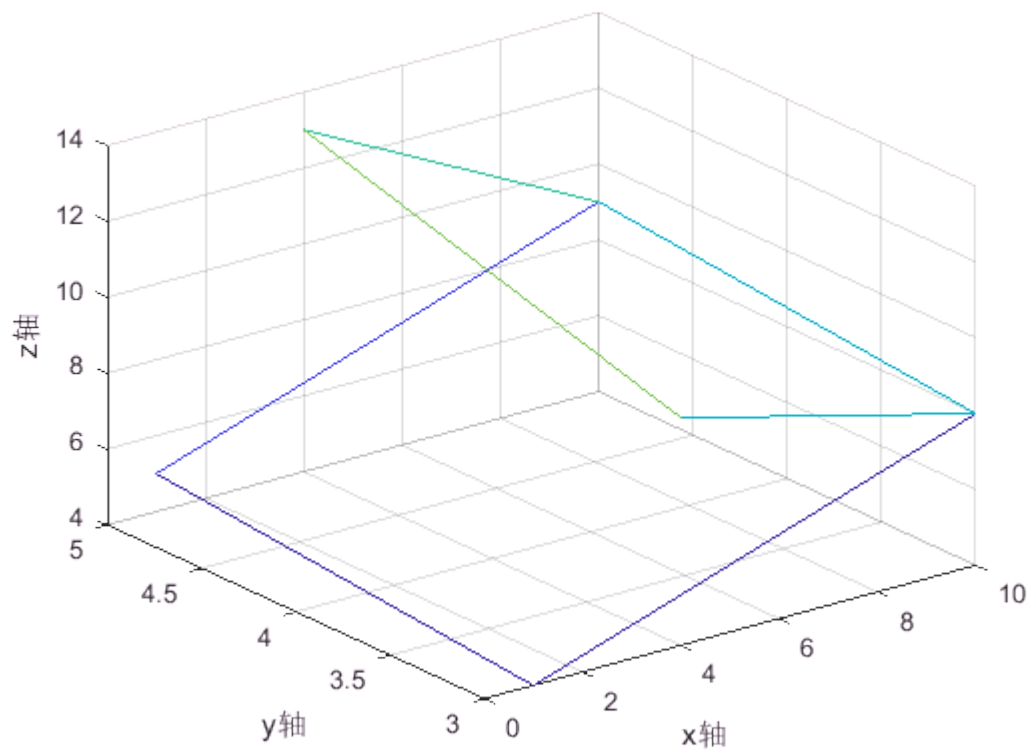
```
Y = [3,5]
```

```
Y =
     3     5
```

```
Z = [4,8,10;5,9,13]
```

```
Z =
     4     8    10
     5     9    13
```

```
mesh(X,Y,Z)
hidden off % 可以看到背部的图像，不会遮挡（默认是看不到的）
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
```

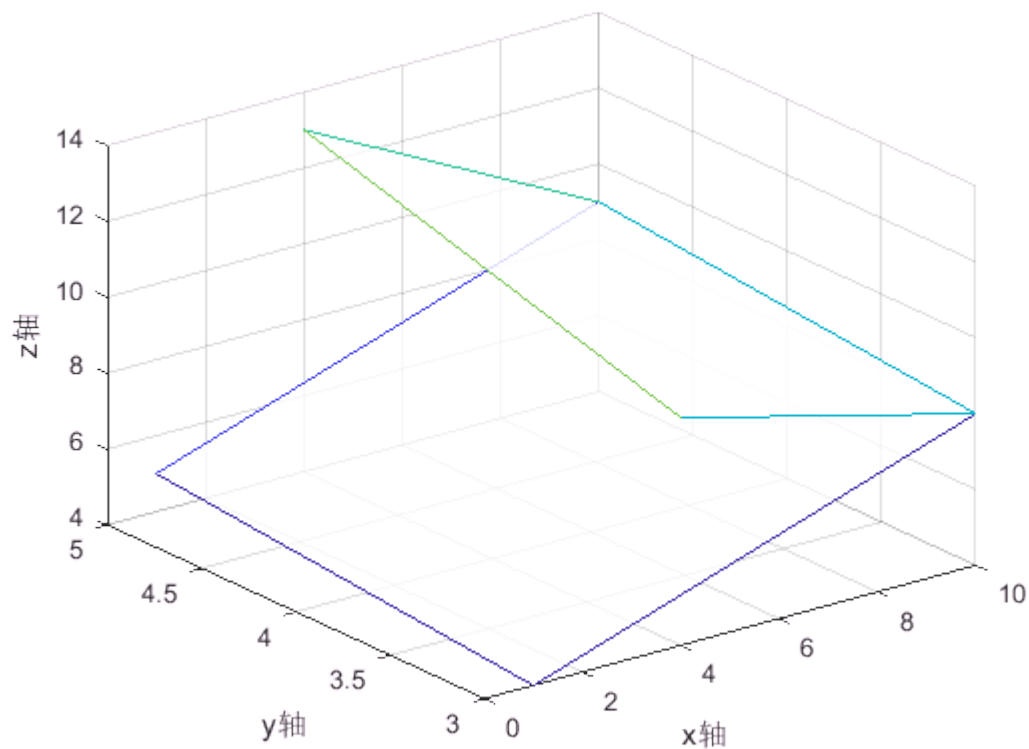


% 如果觉得背部的图像显示的颜色太深了，可以更改透明度

mesh(X,Y,Z)

alpha(0.8) % 设置透明度为0.8，这时候隐隐约约能看到（透明度介于0-1之间，越大表示越透明）

xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签



**mesh(X,Y,Z)**的用法，其中X、Y和**Z**都是**m\*n**维的矩阵

```
X = [1,2,4;1,2,4]
```

```
X =  
    1    2    4  
    1    2    4
```

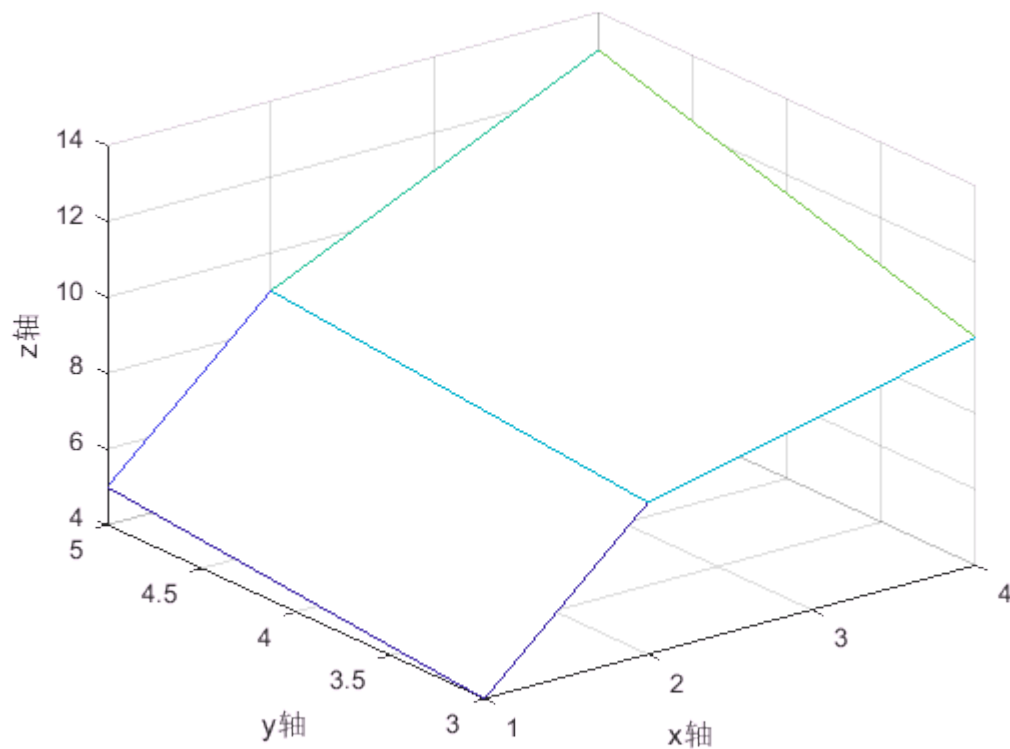
```
% X = [1,2,4;1,2,5]  
Y = [3,3,3;5,5,5]
```

```
Y =  
    3    3    3  
    5    5    5
```

```
Z = [4,8,10;5,9,13]
```

```
Z =  
    4    8   10  
    5    9   13
```

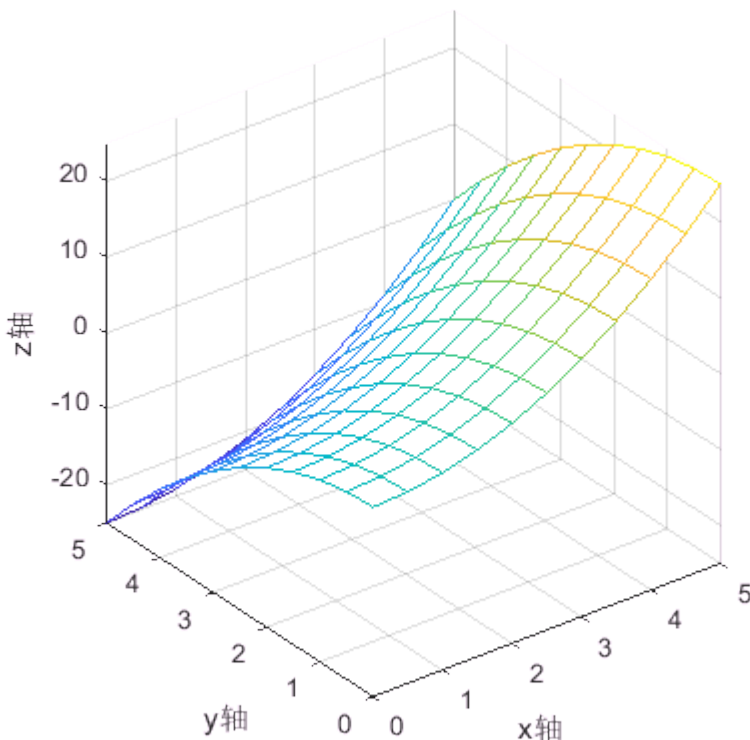
```
mesh(X,Y,Z) % (X(i,j), Y(i,j), Z(i,j))是线框网格线的交点的坐标  
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
```



例题1：绘制  $z = x^2 - y^2$  的图像，其中  $x$  和  $y$  都位于  $[0,5]$  之间

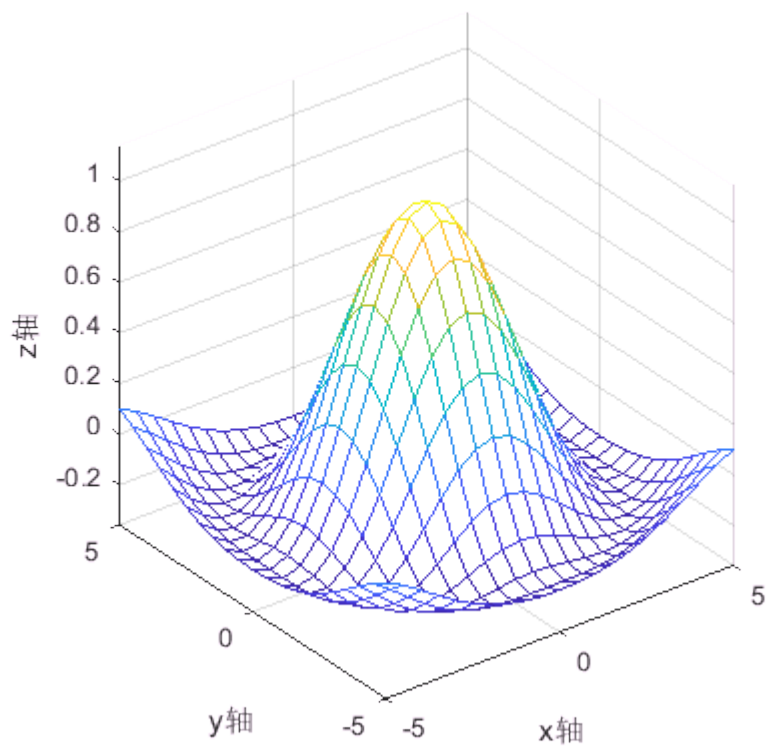
`linspace` 是 Matlab 中的一个函数，用于产生给定范围内指定数量的点数，相邻数据跨度相同，并返回一个行向量。  
调用方法：`linspace(x1,x2,N)` 功能：用于产生  $x_1$ ,  $x_2$  之间的  $N$  点行向量，相邻数据跨度相同。其中  $x_1$ 、 $x_2$ 、 $N$  分别为起始值、终止值、元素个数。

```
n = 11;
tem = linspace(0,5,n); % 将[0,5]这个区间等分为n个点（等差数列的形式）
x = repmat(tem,n,1);
y = repmat(tem',1,n);
z = x.^2 - y.^2; % 要使用点运算符哦~
mesh(x,y,z)
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
axis vis3d % 冻结屏幕高宽比，使得一个三维对象的旋转不会改变坐标轴的刻度显示
```



例题2：绘制  $z = \sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$  的图形，其中 $x$ 和 $y$ 都位于 $[-5,5]$ 之间

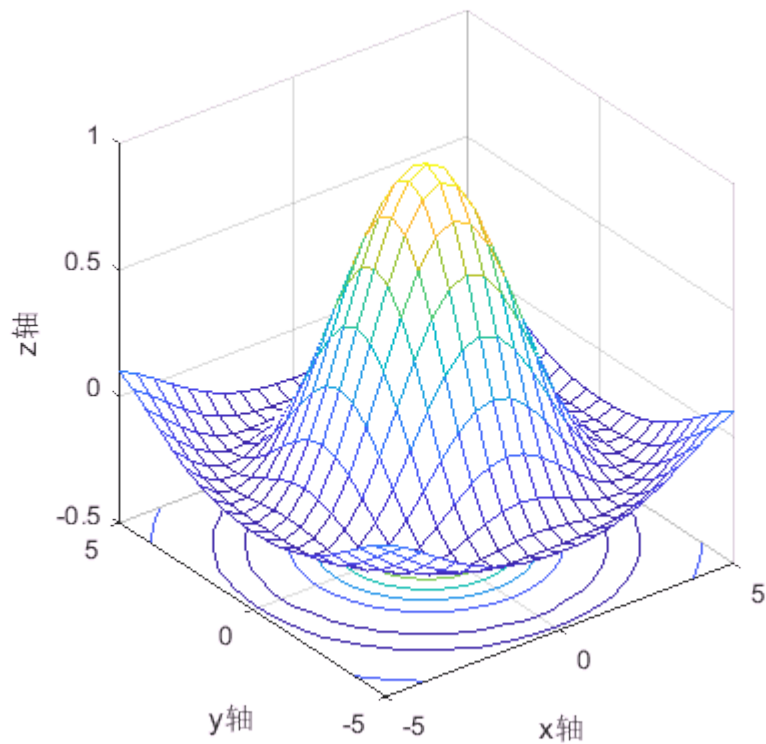
```
[x,y] = meshgrid(-5:0.5:5); % 快速生成网格所需的数据
tem = sqrt(x.^2+y.^2)+1e-12; % tem=sqrt(x.^2+y.^2); % 在后面加上一个非常非常小的数字: 1e-12 =
z = sin(tem)./tem; % 如果不对tem处理, 那么z的最中间的一个值 0/0 = NaN
mesh(x,y,z)
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
axis vis3d % 冻结屏幕高宽比, 使得一个三维对象的旋转不会改变坐标轴的刻度显示
```



**meshc**函数：除了**mesh**函数图形外，还在**xy**平面上绘制曲面的等高线

```
meshc(x,y,z)
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
axis vis3d % 冻结屏幕高宽比，使得一个三维对象的旋转不会改变坐标轴的刻度显示
```





**meshz**函数：除了**mesh**函数图形外，还在**xy**平面上绘制曲面的底座

```
meshz(x,y,z)
xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签
axis vis3d % 冻结屏幕高宽比，使得一个三维对象的旋转不会改变坐标轴的刻度显示
```

