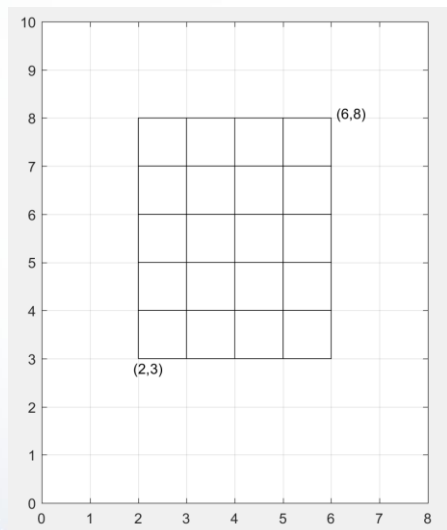


• 4.5 三维曲面

- 平面网格数据的生成
- 绘制三维曲面的mesh函数和surf函数
- fmesh函数和fsurf函数

1. 平面网格数据的生成

用矩阵X、Y分别存储每一个小矩形顶点的x坐标与y坐标，
矩阵X、Y就是该矩形区域的xy平面网格坐标矩阵。



X

6x5 double

	1	2	3	4	5	
1	2	3	4	5	6	^
2	2	3	4	5	6	
3	2	3	4	5	6	
4	2	3	4	5	6	
5	2	3	4	5	6	
6	2	3	4	5	6	v

Y

6x5 double

	1	2	3	4	5	
1	3	3	3	3	3	^
2	4	4	4	4	4	
3	5	5	5	5	5	
4	6	6	6	6	6	
5	7	7	7	7	7	
6	8	8	8	8	8	v

1. 平面网格数据的生成

在MATLAB中，产生平面区域内的网格坐标矩阵有两种方法。

(1) 利用矩阵运算生成。

```
>> x = 2:6;  
>> y = (3:8)';  
>> X = ones(size(y))*x;  
>> Y = y*ones(size(x));
```

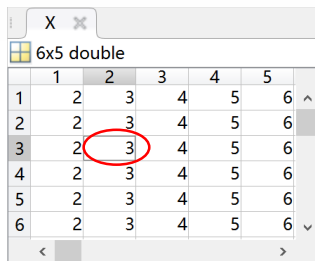


Figure 1: MATLAB display of the X matrix. It is a 6x5 double matrix. The values are 2, 3, 4, 5, 6 repeated for each of the 6 rows. The element at row 3, column 2 (3, 5) is circled in red.

	1	2	3	4	5	6
1	2	3	4	5	6	
2	2	3	4	5	6	
3	2	3	4	5	6	
4	2	3	4	5	6	
5	2	3	4	5	6	
6	2	3	4	5	6	

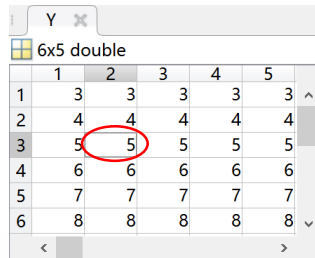
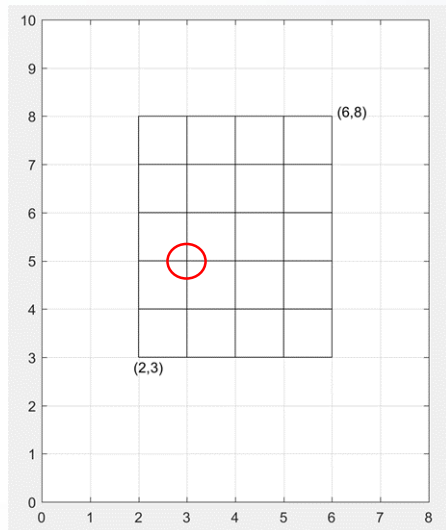


Figure 2: MATLAB display of the Y matrix. It is a 6x5 double matrix. The values are 3, 4, 5, 6, 7, 8 repeated for each of the 5 columns. The element at row 3, column 2 (3, 5) is circled in red.

	1	2	3	4	5	6
1	3	4	5	6	7	8
2	3	4	5	6	7	8
3	3	4	5	6	7	8
4	3	4	5	6	7	8
5	3	4	5	6	7	8
6	3	4	5	6	7	8



X和Y相同位置上的元素，如 X_{32} 、 Y_{32} 是区域的第3行第2列网格点的坐标(3, 5)。

1. 平面网格数据的生成

(2) 利用meshgrid函数生成。

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

其中，参数x、y为向量，存储网格点坐标的X、Y为矩阵。

```
>> x = 2:1:6;
```

```
>> y = (3:1:8)';
```

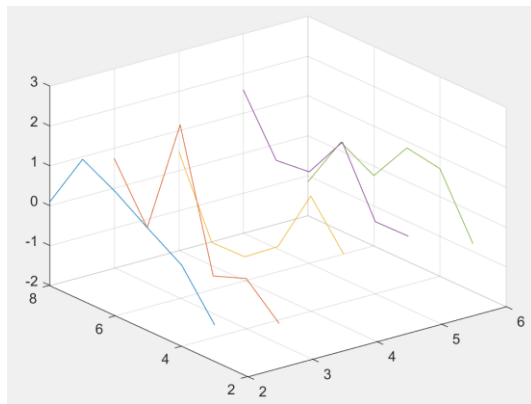
```
>> [X, Y] = meshgrid(x, y);
```

第3行命令生成的网格坐标矩阵X、Y与方法(1)得到的相同。

若根据每一个网格点上的 x 、 y 坐标求函数值 z ，则得到函数值矩阵 Z 。矩阵 X 、 Y 、 Z 中的各个列向量，对应于一条曲线数据点的坐标。

例1 绘制空间曲线。

```
x = 2:6;  
y = (3:8)';  
[X, Y] = meshgrid(x, y);  
Z = randn(size(X));  
plot3(X, Y, Z)  
grid on;
```



2. 绘制三维曲面的函数

□ mesh函数和surf函数的调用格式:

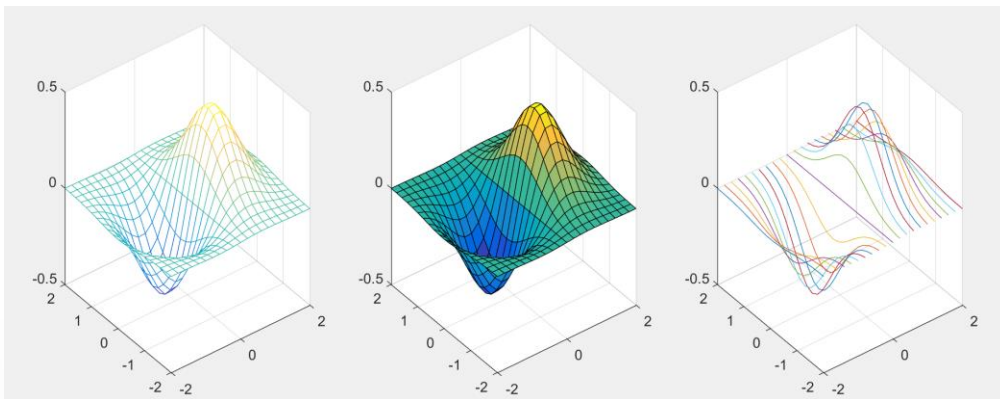
`mesh(x, y, z, c)`

`surf(x, y, z, c)`

其中, x 、 y 是网格坐标矩阵, z 是网格点上的高度矩阵, c 用于指定在不同高度下的曲面颜色。 c 省略时, 颜色的设定正比于图形的高度。

例2 绘制三维曲面图 $z = xe^{-x^2 - y^2}$ 。

```
t = -2:0.2:2;  
[X, Y] = meshgrid(t);  
Z = X .* exp(-X.^2 - Y.^2);  
subplot(1, 3, 1)  
mesh(X, Y, Z);  
subplot(1, 3, 2)  
surf(X, Y, Z);  
subplot(1, 3, 3)  
plot3(X, Y, Z);  
grid on
```



2. 绘制三维曲面的函数

□ mesh函数和surf函数的其他调用格式:

`mesh(z, c)`

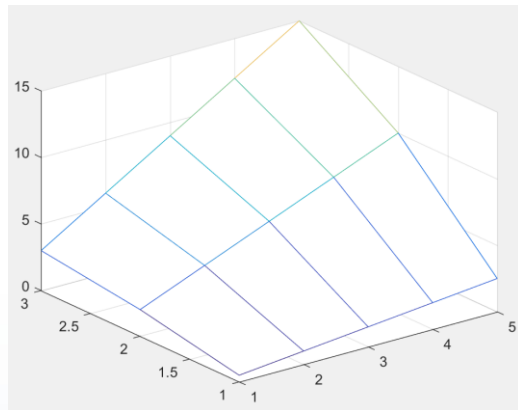
`surf(z, c)`

当x、y省略时，z矩阵的第2维下标当作x轴坐标，z矩阵的第1维下标当作y轴坐标。

```
>> t = 1:5;
```

```
>> z = [0.5*t; 2*t; 3*t];
```

```
>> mesh(z);
```

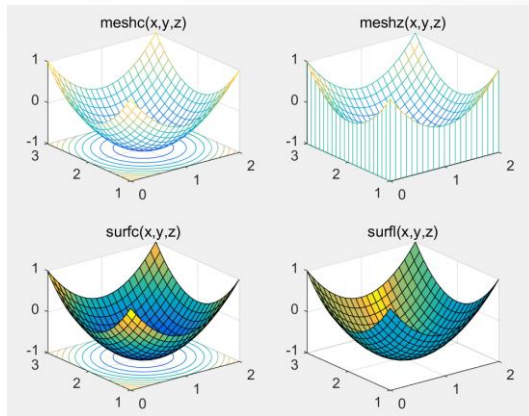


2. 绘制三维曲面的函数

- ❑ 带等高线的三维网格曲面函数meshc
- ❑ 带底座的三维网格曲面函数meshz
- ❑ 具有等高线的曲面函数surfc
- ❑ 具有光照效果的曲面函数surf1

例3 用4种方式绘制函数 $z = (x - 1)^2 + (y - 2)^2 - 1$ 的曲面图。
其中, $x \in [0, 2]$, $y \in [1, 3]$ 。

```
[x,y]=meshgrid(0:0.1:2,1:0.1:3);  
z=(x-1).^2+(y-2).^2-1;  
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);  
surf1(x,y,z); title('surf1(x,y,z)')
```



3. 标准三维曲面

(1) sphere函数

$$[x, y, z] = \text{sphere}(n)$$

产生3个 $(n+1)$ 阶的方阵，采用这3个矩阵可以绘制出圆心位于原点、半径为1的单位球体。

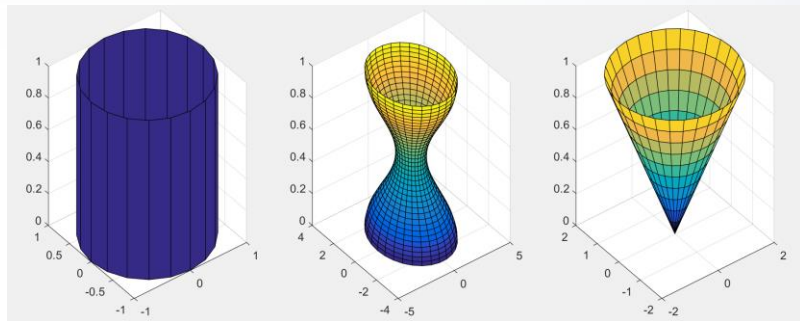
(2) cylinder函数

$$[x, y, z] = \text{cylinder}(R, n)$$

其中，参数 R 是一个向量，存放柱面各个等间隔高度上的半径， n 表示在圆柱圆周上有 n 个间隔点，默认有20个间隔点。

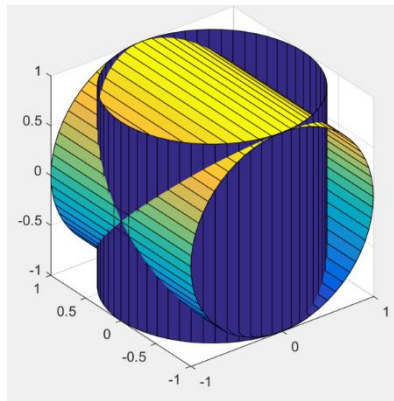
例4 用cylinder函数分别绘制柱面、花瓶和圆锥面。

```
subplot(1, 3, 1);  
[x, y, z]=cylinder;  
surf(x, y, z);  
subplot(1, 3, 2);  
t=linspace(0, 2*pi, 40);  
[x, y, z]= cylinder(2+cos(t), 30);  
surf(x, y, z);  
subplot(1, 3, 3);  
[x, y, z]= cylinder(0:0.2:2, 30);  
surf(x, y, z);
```



例5 用cylinder函数绘制两个相互垂直且直径相等的圆柱面的相交图形。

```
[x, y, z]= cylinder(1, 60);  
z=[-1*z(2, :); z(2, :)];  
surf(x, y, z)  
hold on  
surf(y, z, x)  
axis equal
```



3. 标准三维曲面

(3) peaks 函数

peaks函数的调用格式:

peaks(n)

```
>> p1=peaks(10);
```

peaks(V)

```
>> p2=peaks;
```

peaks(x, y)

```
>> p3=peaks(-3:0.2:3);
```

peaks

```
>> [x, y]=meshgrid(-2:0.1:2, 0:0.1:5);
```

```
>> p4=peaks(x, y);
```

4. fmesh函数和fsurf函数

用于绘制参数方程定义的曲面

```
fsurf(funx, funy, funz, uvlims)
```

```
fmesh(funx, funy, funz, uvlims)
```

其中，funx、funy、funz代表定义曲面x、y、z坐标的函数，通常采用函数句柄的形式。uvlims为funx、funy和funz的自变量的取值范围，用4元向量[umin, umax, vmin, vmax]描述，默认为[-5, 5, -5, 5]。

例6 绘制螺旋曲面。

$$\begin{cases} x = u \sin v \\ y = -u \cos v \\ z = v \end{cases} \quad -5 < u < 5, \quad -5 < v < 2$$

```
funx = @(u,v) u.*sin(v);  
funy = @(u,v) -u.*cos(v);  
funz = @(u,v) v;  
fsurf(funx,funy,funz, [-5 5 -5 -2])  
hold on  
fmesh(funx,funy,funz, [-5 5 -2 2])  
hold off
```

