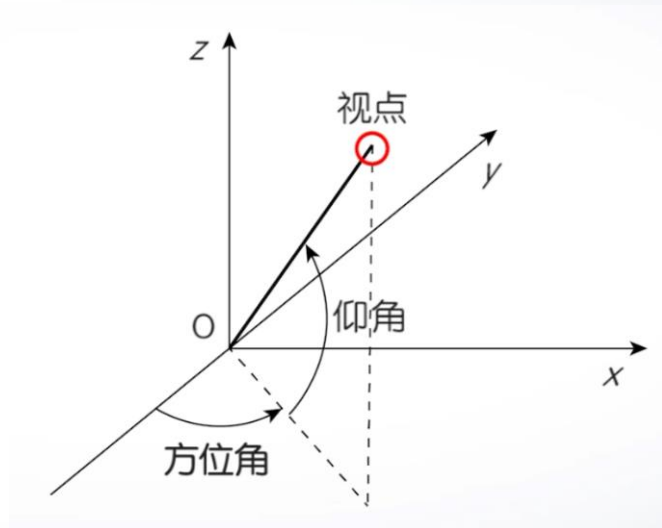


• 4.6 图形修饰处理

- 视点处理
- 色彩处理
- 图形的裁剪处理

1. 视点处理

- 方位角：视点与原点连线在 xy 平面上的投影与 y 轴负方向形成的角度，正值表示逆时针，负值表示顺时针。
- 仰角：视点与原点连线与 xy 平面的夹角，正值表示视点在 xy 平面上方，负值表示视点在 xy 平面下方。



1. 视点处理

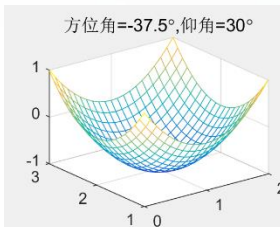
(1) view函数的基本用法

`view(az, el)`

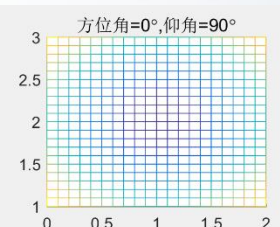
其中，az为方位角，el为仰角。系统默认的视点定义为方位角 -37.5° ，仰角 30° 。

例1 绘制函数 $z = (x - 1)^2 + (y - 2)^2 - 1$ 曲面，并从不同视点展示曲面。

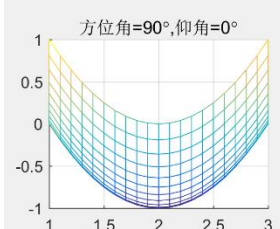
```
[x, y]=meshgrid(0:0.1:2, 1:0.1:3);  
z=(x-1).^2+(y-2).^2-1;  
subplot(2, 2, 1); mesh(x, y, z)  
title('方位角=-37.5{\circ}, 仰角=30{\circ}')
```



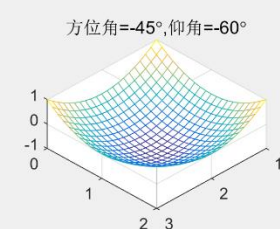
```
view(0, 90); title('方位角=0{\circ}, 仰角=90{\circ}')
```



```
subplot(2, 2, 2); mesh(x, y, z)  
view(90, 0); title('方位角=90{\circ}, 仰角=0{\circ}')
```



```
subplot(2, 2, 3); mesh(x, y, z)  
view(-45, -60); title('方位角=-45{\circ}, 仰角=-60{\circ}')
```





1. 视点处理

(2) view函数的其他用法

`view(x, y, z)`

`view(2)`

`view(3)`

2. 色彩处理

(1) 颜色的向量表示

向量元素在 $[0, 1]$ 范围内取值，3个元素依次表示红、绿、蓝3种颜色的相对亮度，称为RGB三元组。

[R	G	B]
↑	↑	↑
红	绿	蓝

[0	0	1]	: 蓝色
[1	0	0]	: 红色
[0	1	0]	: 绿色
[1	1	1]	: 白色
[0	0	0]	: 黑色

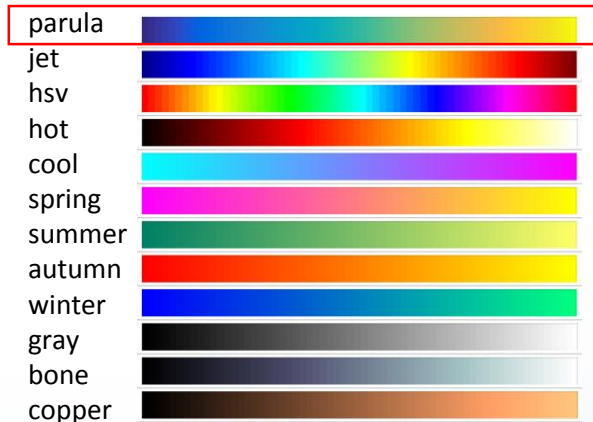
2. 色彩处理

(2) 色图 (Colormap)

□ 色图矩阵

```
>> cmap = colormap(parula(5))  
cmap =  
    0.2081    0.1663    0.5292  
    0.0795    0.5159    0.8328  
    0.1986    0.7214    0.6310  
    0.8266    0.7320    0.3464  
    0.9763    0.9831    0.0538
```

□ 内建色图



2. 色彩处理

(2) 色图 (Colormap)

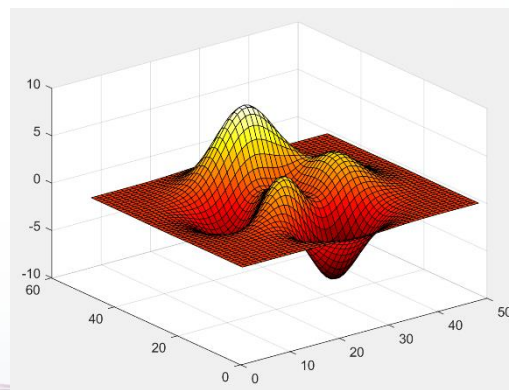
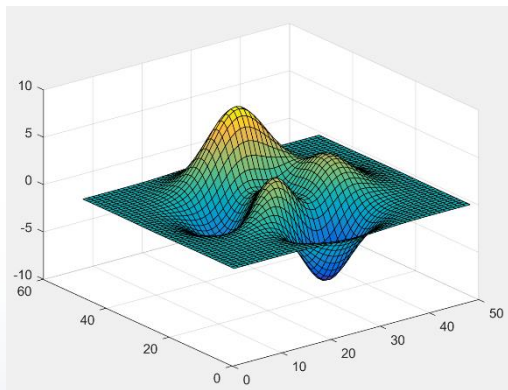
- 指定当前图形使用的色图

`colormap cmapname`

`colormap(cmap)`

```
>> surf(peaks)
```

```
>> colormap hot
```



2. 色彩处理

(2) 色图 (Colormap)

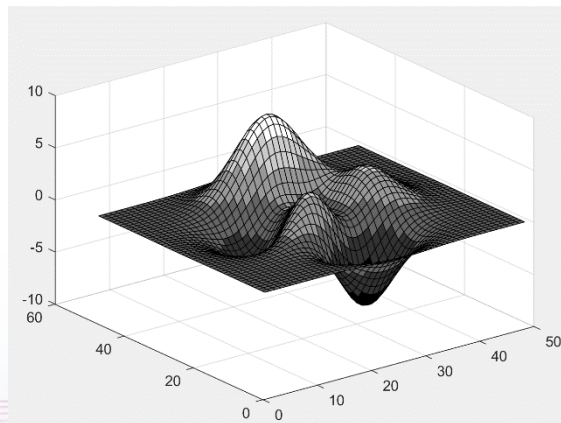
□ 创建色图矩阵

色图矩阵的每一行是RGB三元组。可以自定义色图矩阵，也可以调用MATLAB提供的函数来定义色图矩阵。

例2 创建一个灰色系列色图矩阵。

```
c = [0, 0.2, 0.4, 0.6, 0.8, 1.0]';  
cmap = [c, c, c];  
surf(peaks)  
colormap(cmap)
```

`cmap=gray(6);`



2. 色彩处理

(3) 三维图形表面的着色

可以用shading函数来改变着色方式。

`shading faceted`

每个网格片用其高度对应的颜色进行着色，网格线是黑色。这是默认着色方式。

`shading flat`

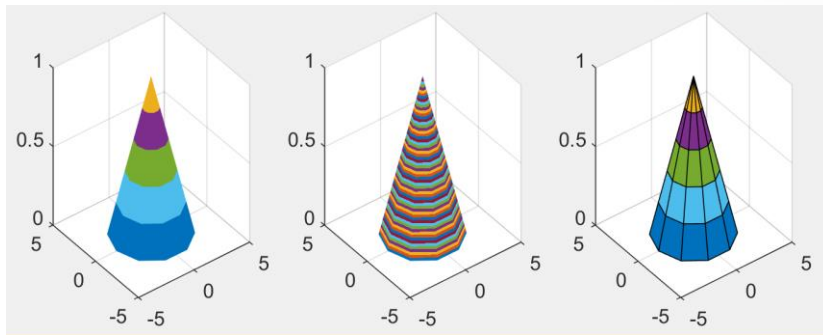
每个网格片用同一个颜色进行着色，且网格线也用相应的颜色。

`shading interp`

网格片内采用颜色插值处理。

例3 使用同一色图，以不同着色方式绘制圆锥体。

```
[x,y,z]= cylinder(pi:-pi/5:0,10);  
colormap(lines);  
subplot(1,3,1);  
surf(x,y,z); shading flat  
subplot(1,3,2);  
surf(x,y,z); shading interp  
subplot(1,3,3);  
surf(x,y,z);
```

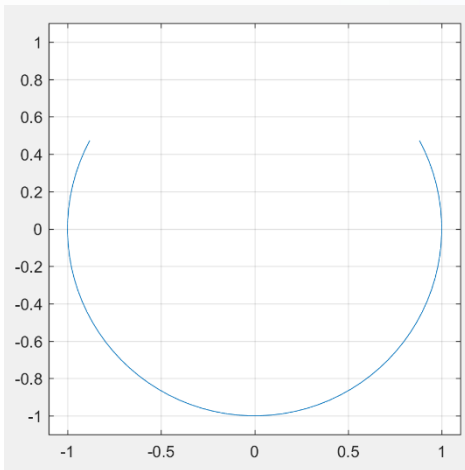


3. 图形的裁剪处理

将图形中需要裁剪部分对应的函数值设置成NaN，这样在绘制图形时，函数值为NaN的部分将不显示出来，从而达到对图形进行裁剪的目的。

例4 绘制3/4圆。

```
t = linspace(0, 2*pi, 100);  
x = sin(t);  
y = cos(t);  
p = y > 0.5;  
y(p) = NaN;  
plot(x, y)  
axis([-1.1, 1.1, -1.1, 1.1])  
axis square  
grid on
```



例5 绘制3/4球面。

```
[X, Y, Z] = sphere(60);  
p = Z>0.5;  
Z(p) = NaN;  
surf(X, Y, Z)  
axis([-1, 1, -1, 1, -1, 1])  
axis equal  
view(-45, 20)
```

