

MATLAB统计分析与应用

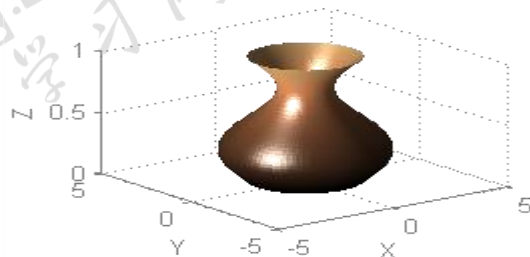
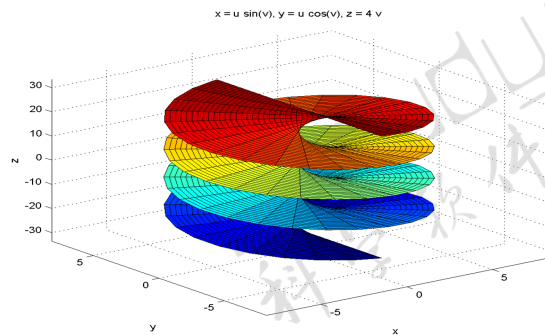
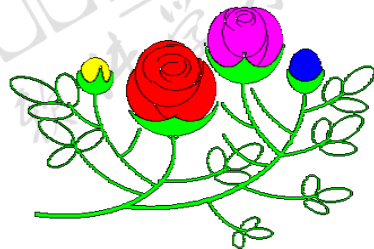
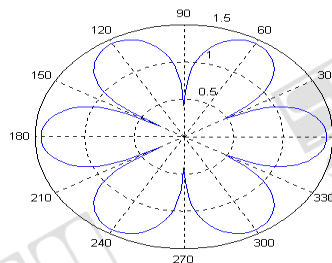
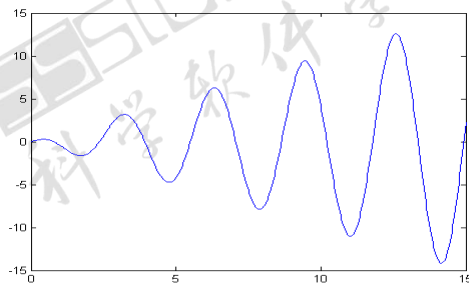
——MATLAB绘图与可视化

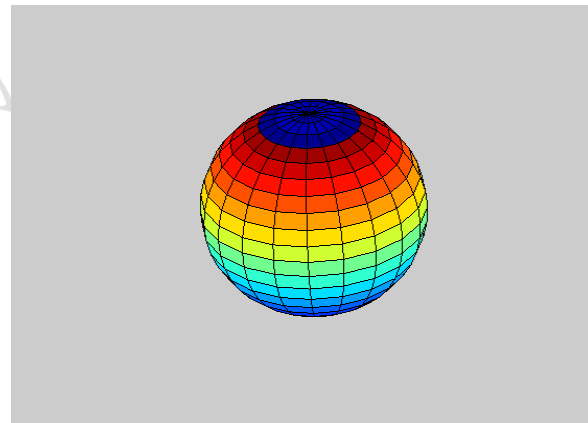
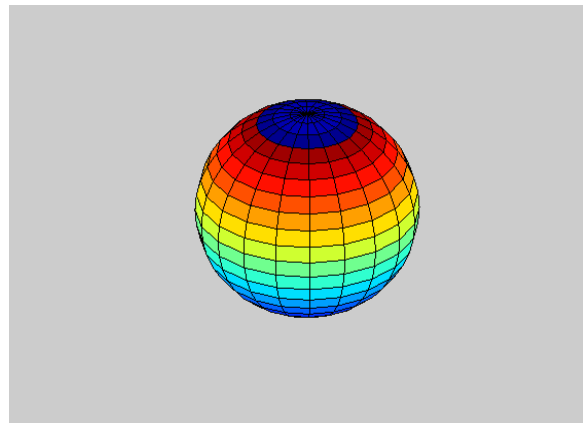
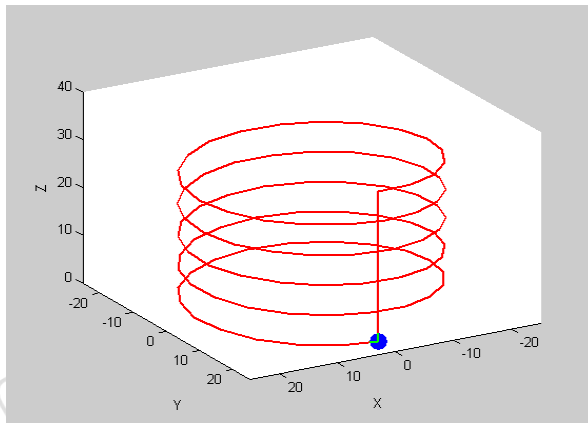
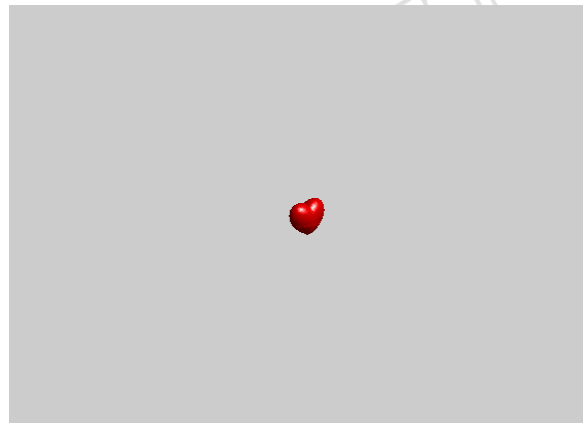
主讲人：谢中华

SSstudy.com

科学软件学习网

MATLAB设有大量函数和命令来绘制出各种各样的图形，具有强大的绘图功能





主要内容

- 图形对象与图形句柄
- 二维图形绘制
- 三维图形绘制
- 图形的打印和输出

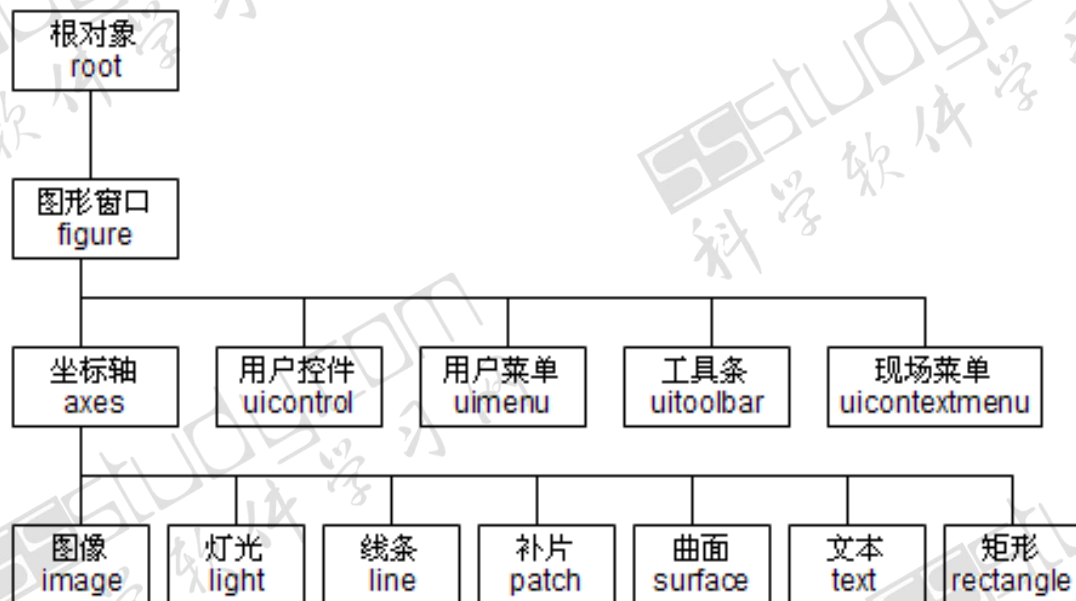
第一节 图形对象与图形对象句柄

一、什么是图形对象与图形句柄

图形窗口、线条、曲面和注释等都被看作是MATLAB中的图形对象，每个图形对象都对应一个唯一的句柄值。例如命令 `h = figure` 返回一个图形窗口的句柄值，可通过句柄控制图形对象的各种属性。

句柄式图形对象继承关系图

Parent (父)



Children (子)

二、获取图形对象属性名称和属性值

```
>> h = line([0 1],[0 1])
```

```
h =
```

```
0.0149
```

```
>> get(h)
```

三、设置图形对象属性值

```
>> subplot(1, 2, 1);
```

```
>> h1 = line([0 1],[0 1]) ;
```

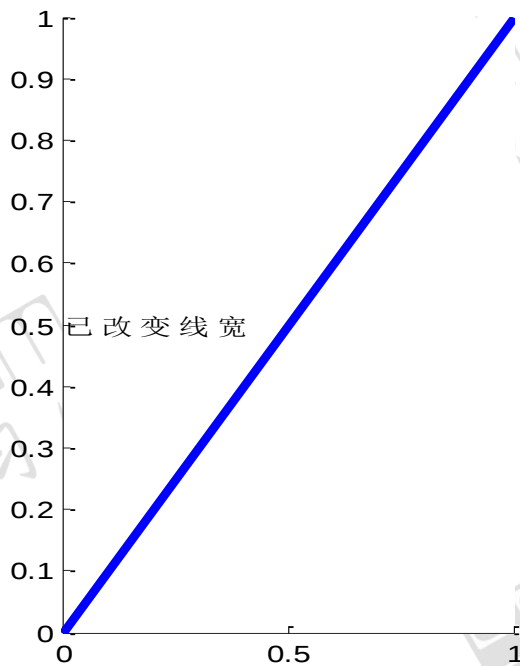
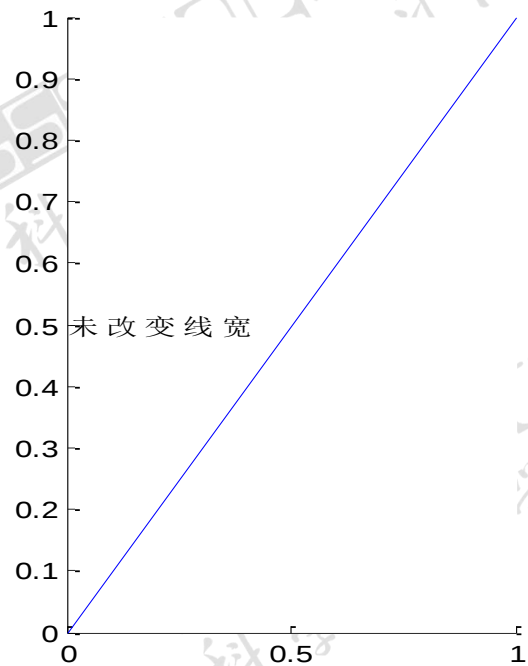
```
>> text(0, 0.5, '未改变线宽') ;
```

```
>> subplot(1, 2, 2);
```

```
>> h2 = line([0 1],[0 1]) ;
```

```
>> set(h2, 'LineWidth', 3)
```

```
>> text(0, 0.5, '已改变线宽') ;
```



第二节 二维图形绘制

一、基本二维绘图函数

MATLAB 中提供了 `plot` , `loglog` , `semilogx` , `semilogy` , `polar` , `plotyy` 等6个非常实用的基本二维绘图函数，下面重点介绍 `plot` 函数的用法。

1. plot函数

➤ 调用格式：

`plot(Y)`

`plot(X,Y)`

`plot(X1,Y1, X2,Y2,...)`

`plot(X1,Y1,LineStyle,...)`

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

`plot(axes_handle,...)`

`h = plot(...)`

`hlines = plot('v6',...)`

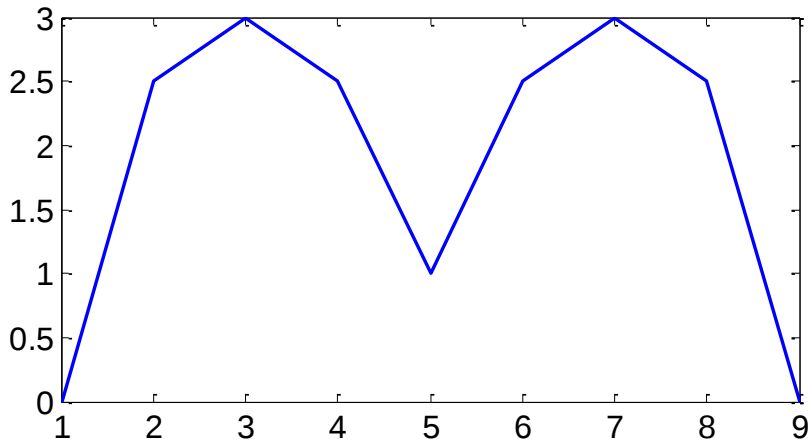
【例5.2-1】给定横纵坐标数据，绘制线图

x	1	2	3	4	5	6	7	8	9
y	0	2.5	3	2.5	1	2.5	3	2.5	0

```
>> x = 1:9;
```

```
>> y = [0 2.5 3 2.5 1 2.5 3 2.5 0];
```

```
>> plot(x,y)
```



➤ 线条属性的修改

线型、描点类型、颜色参数表

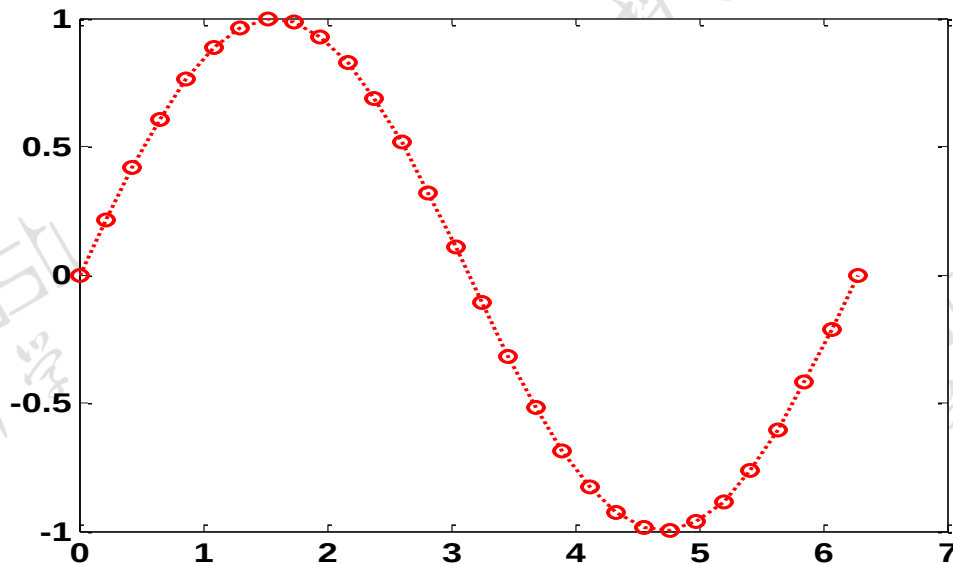
色彩 字符	说明	线型 字符	说明	描点 类型	说明	描点 类型	说明
r	红	-	实线（默认）	.	点	<	左三角形
g	绿	--	虚线	o	圆	s	方形
b	蓝	:	点线	x	叉号	d	菱形
c	青	-.	点划线	+	加号	P	五角星
m	品红			*	星号	h	六角星
y	黄			v	下三角形		
k	黑			^	上三角形		
w	白			>	右三角形		

【例5.2-2】绘制正弦函数图像。

```
>> x = linspace(0,2*pi,30);
```

```
>> y = sin(x);
```

```
>> plot(x,y,'r:o');
```



【例5.2-2续】绘制正弦函数图像。

```
>> x = 0 : 0.25 : 2*pi;
```

```
>> y = sin(x);
```

```
>> plot(x, y, '-ro',...
```

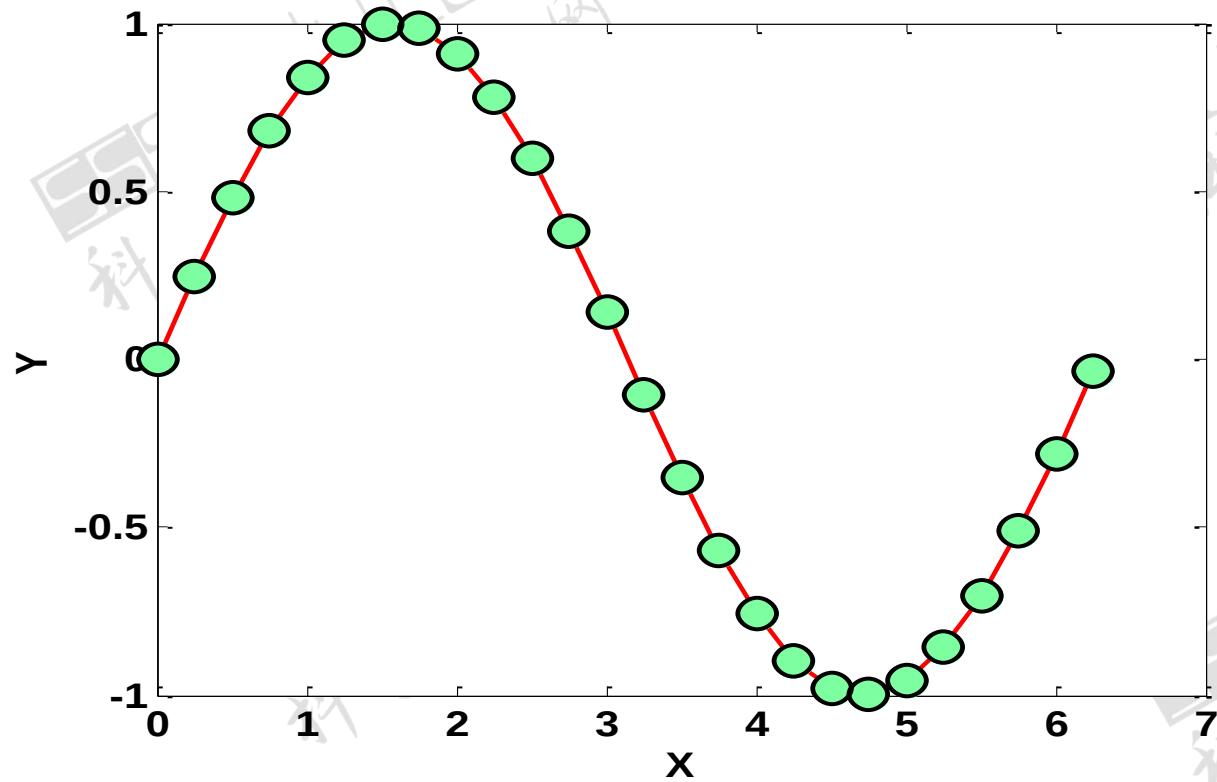
```
    'LineWidth',2,...
```

```
    'MarkerEdgeColor','k',...
```

```
    'MarkerFaceColor',[0.49, 1, 0.63],...
```

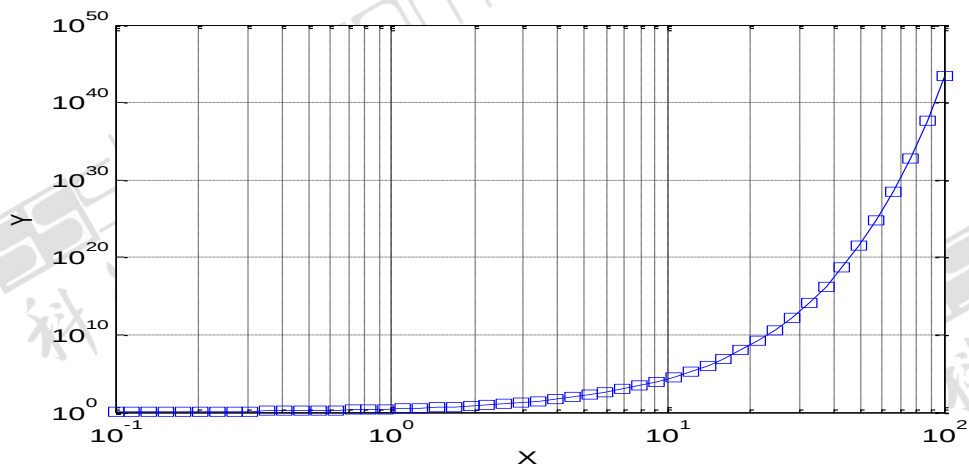
```
    'MarkerSize',12)
```

```
>> xlabel('X'); ylabel('Y');
```



2. loglog函数：双对数坐标绘图

```
>> x = logspace(-1,2);  
>> loglog(x,exp(x),'-s')  
>> grid on  
>> xlabel('X'); ylabel('Y');
```

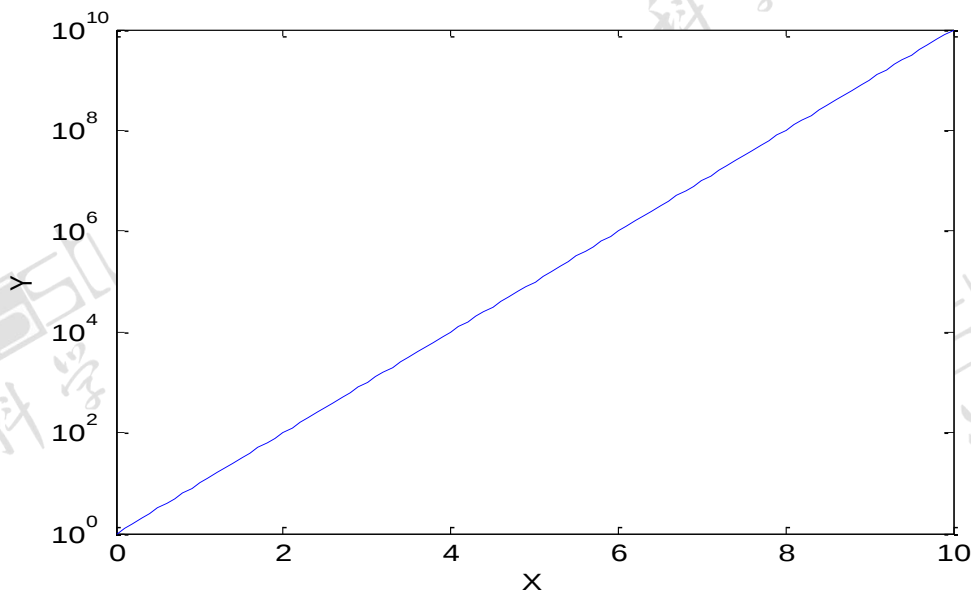


3. semilogx , semilogy函数：半对数坐标绘图

```
>> x = 0 : 0.1 : 10;
```

```
>> semilogy(x, 10.^x)
```

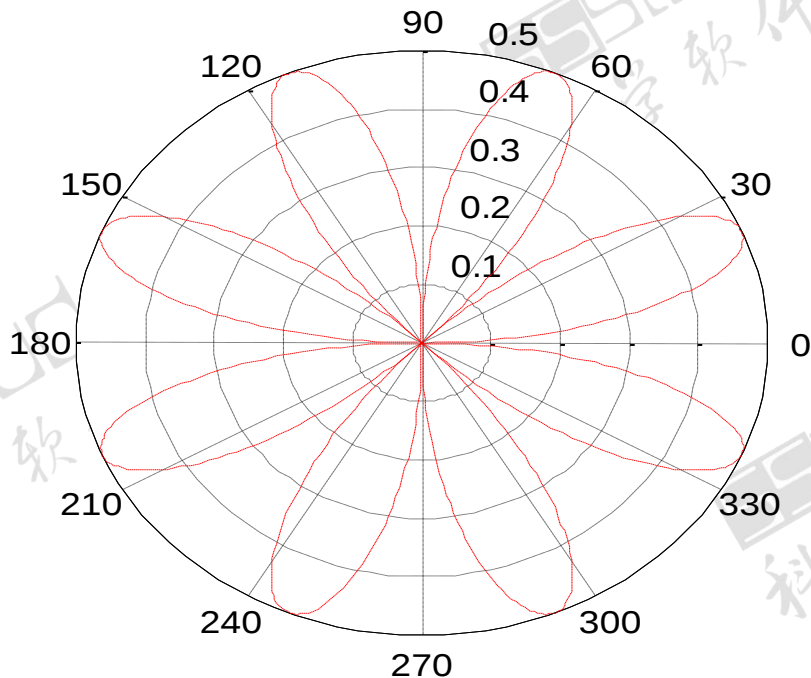
```
>> xlabel('X'); ylabel('Y');
```



4. polar函数：极坐标绘图

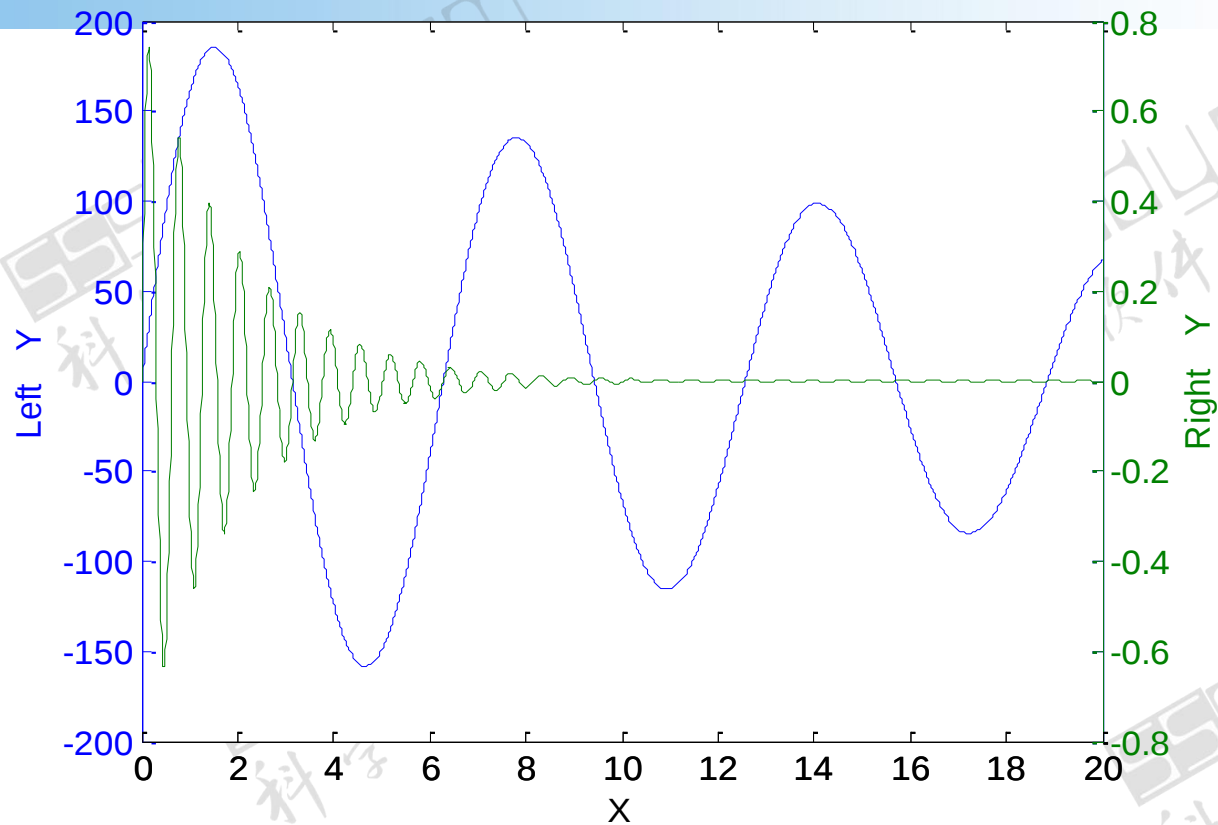
```
>> t = 0 : 0.01 : 2*pi;
```

```
>> polar(t, sin(2*t).*cos(2*t),'--r')
```



5. plotyy函数：双纵坐标绘图

```
>> x = 0:0.01:20;  
>> y1 = 200*exp(-0.05*x).*sin(x);  
>> y2 = 0.8*exp(-0.5*x).*sin(10*x);  
>> ax = plotyy(x,y1,x,y2,'plot'); xlabel('X');  
>> set(get(ax(1),'Ylabel'),'string','Left Y');  
>> set(get(ax(2),'Ylabel'),'string','Right Y');
```



双纵坐标绘图

二、二维图形修饰和添加注释

可以通过MATLAB命令对图形进行修饰和添加注释，也可以通过图形窗口的菜单项和工具栏完成这些工作。后者通过鼠标点击操作，相对比较简单，下面仅对相关命令进行介绍。

二维绘图的图例标注说明

Plot title

Light Intensity as a Function of Distance

Legend

y axis
label

INTENSITY (lux)

Text

Comparison between theory and experiment.

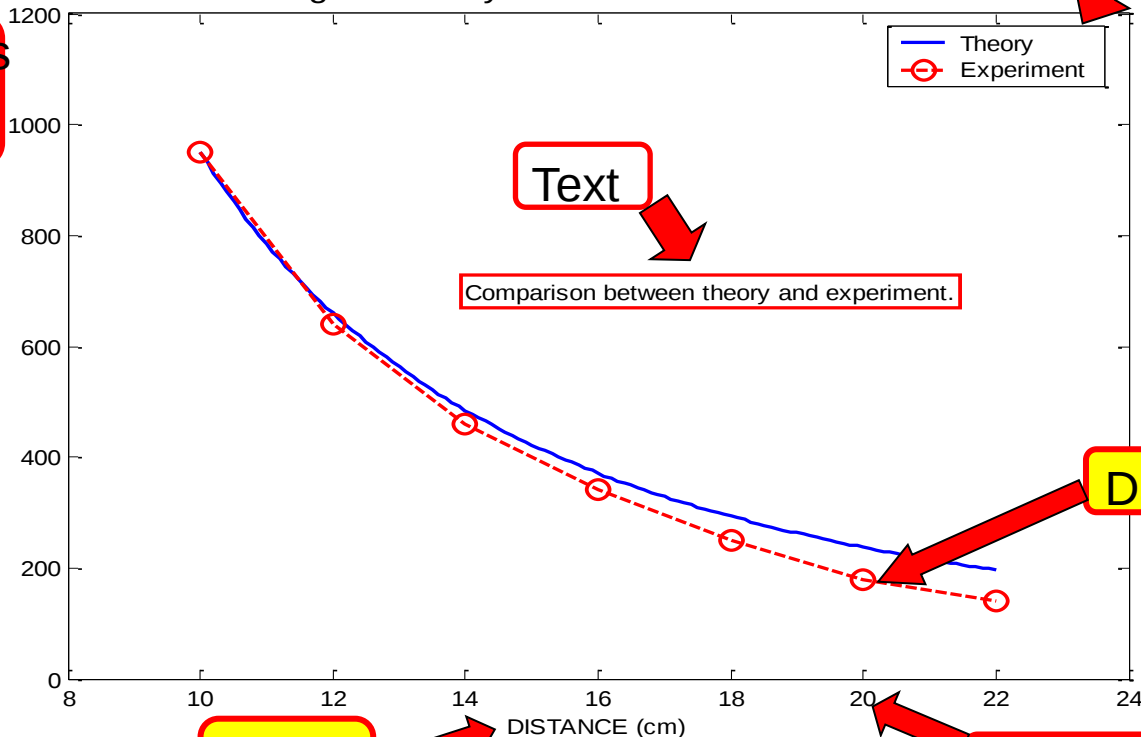
Tick-mark

Data symbol

x axis
label

DISTANCE (cm)

Tick-mark label



➤ **hold** 函数：开启和关闭图形窗口的图形保持功能

➤ **axis** 函数：设置坐标系的刻度和显示方式

➤ **box** 函数：显示或隐藏坐标边框

➤ **grid** 函数：为当前坐标系添加网格

➤ **title** 函数：为当前坐标系添加标题

➤ **xlabel**和**ylabel** 函数：为当前坐标轴添加标签

➤ **text**函数：在当前坐标系中添加文本对象（text对象）

➤ **gtext**函数：在当前坐标系中交互式添加文本对象

➤ **legend**函数：在当前坐标系中添加line对象和
patch对象的图形标注框

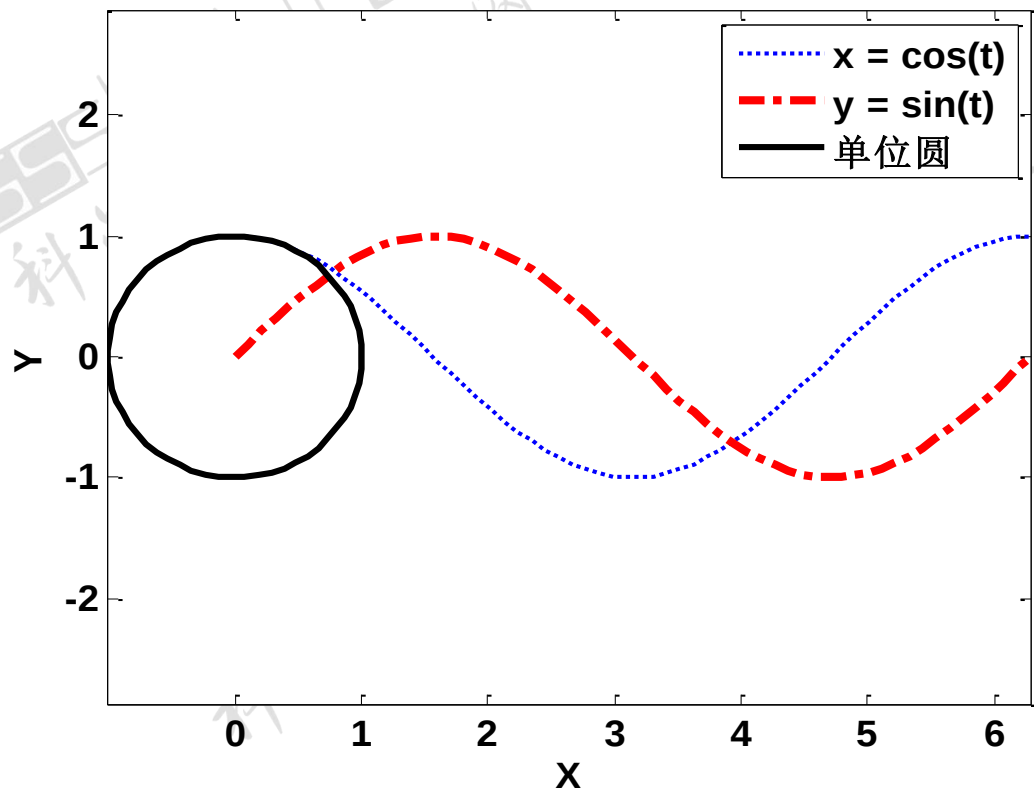
➤ **annotation**函数：在当前图形窗口建立注释对象
(annotation对象)

➤ **subplot** 函数：绘制子图，即在当前图形窗口以
平铺的方式创建多个坐标系

【例5.2-3】 在同一个图形窗口内绘制多条曲线，

设置不同的属性，并添加标注。

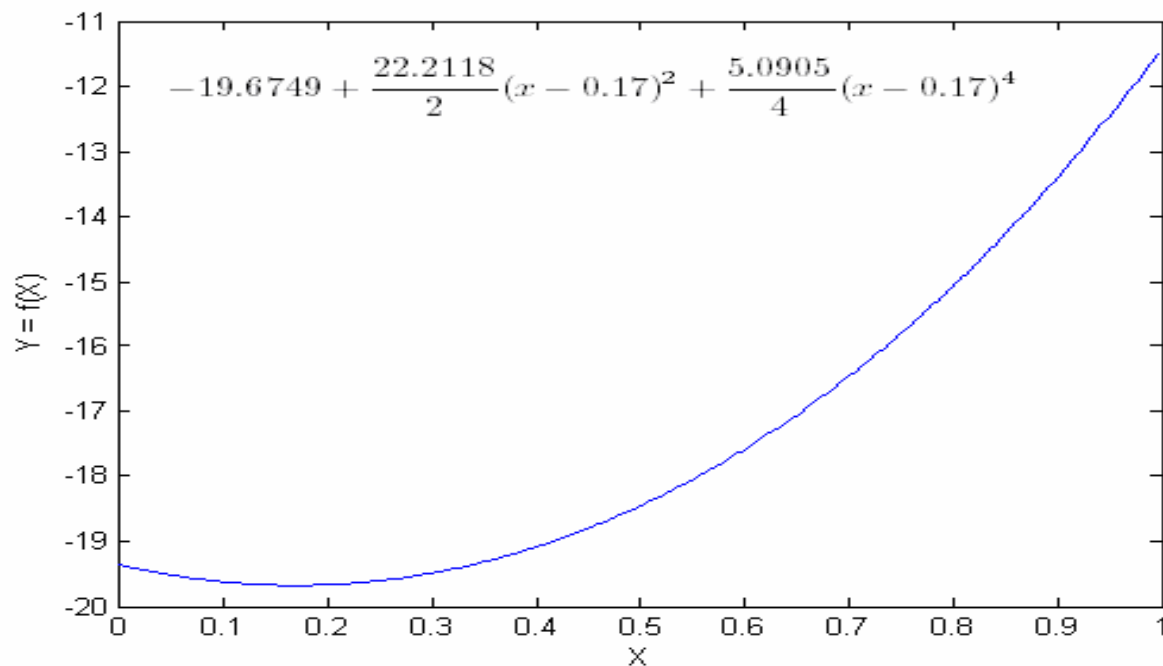
```
>> t = linspace(0,2*pi,60);  
>> x = cos(t);  
>> y = sin(t);  
>> plot(t,x,'.', 'LineWidth',2);  
>> hold on;  
>> plot(t,y,'r-.', 'LineWidth',3);  
>> plot(x,y,'k', 'LineWidth',2.5);  
>> axis equal;  
>> xlabel('X');  
>> ylabel('Y');  
>> legend('x = cos(t)', 'y = sin(t)', '单位圆', 'Location', 'NorthEast');
```



【例5.2-4】绘制曲线

$$-19.6749 + \frac{22.2118}{2}(x - 0.17)^2 + \frac{5.0905}{4}(x - 0.17)^4$$

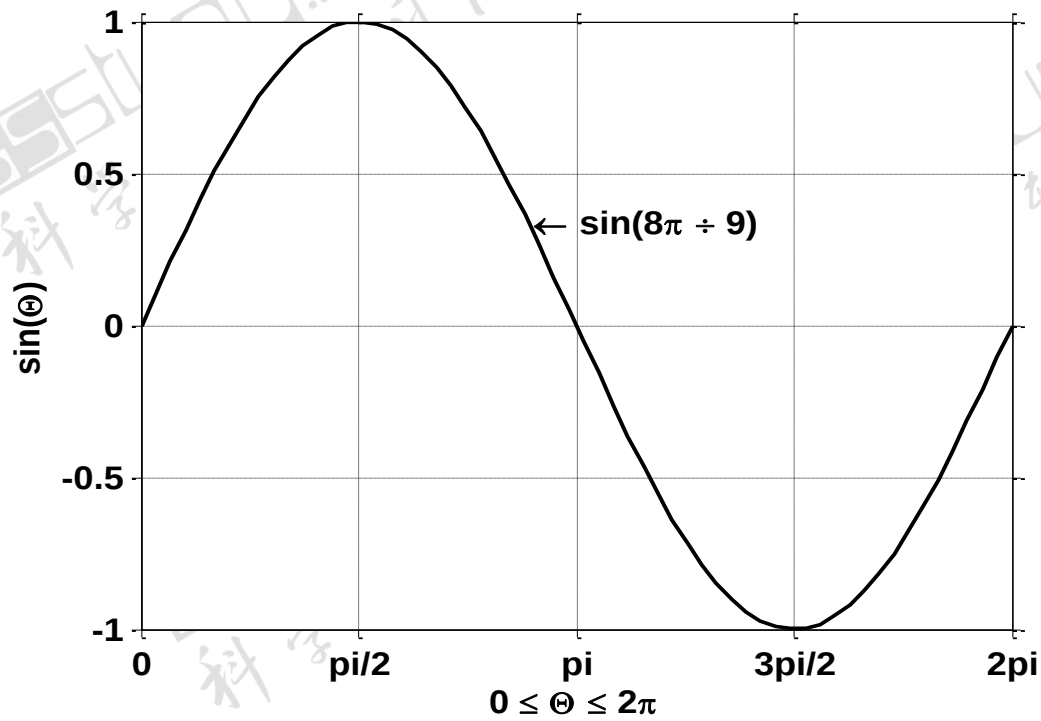
并添加曲线方程。



```
>> a = [-19.6749 22.2118 5.0905];  
>> fun = @(a,x)[a(1)+a(2)/2*(x-0.17).^2+a(3)/4*(x-0.17).^4];  
>> x = 0:0.01:1;  
>> y = fun(a, x);  
>> plot(x,y);  
>> xlabel('X'); ylabel('Y = f(X)');  
>> text('Interpreter','latex',...  
        'String', ['$-19.6749+\frac{22.2118}{2}(x-0.17)^2$...  
        '+\frac{5.0905}{4}(x-0.17)^4$'], 'Position', [0.05, -12],...  
        'FontSize', 12);
```

【例5.2-5】利用图形对象属性修饰图形。

```
>> x = linspace(0,2*pi,60);  
>> y = sin(x);  
>> h = plot(x,y);  
>> grid on;  
>> set(h,'Color','k','LineWidth',2);  
>> XTickLabel = {'0','pi/2','pi','3pi/2','2pi'};  
>> set(gca,'XTick',[0:pi/2:2*pi],...  
    'XTickLabel',XTickLabel,...  
    'TickDir','out');  
>> xlabel('0 \leq \Theta \leq 2\pi');  
>> ylabel('sin(\Theta)');  
>> text(8*pi/9,sin(8*pi/9),'\leftarrow sin(8\pi \div 9)',...  
    'HorizontalAlignment','left')  
>> axis([0 2*pi -1 1]);
```



三、常用统计绘图函数

函数名	功能说明	函数名	功能说明
hist / hist3	二维/三维频数直方图	cdfplot	经验累积分布图
histfit	直方图的正态拟合	ecdfhist	经验分布直方图
boxplot	箱线图	lsline	为散点图添加最小二乘线
probplot	概率图	refline	添加参考直线
qqplot	q-q 图（分位数图）	refcurve	添加参考多项式曲线
normplot	正态概率图	gline	交互式添加一条直线
ksdensity	核密度图	scatterhist	绘制边缘直方图

【例5.2-6】 用normrnd函数产生1000个标准正态分布随机数，并做出频数直方图和经验分布函数图。

```
>> x = normrnd(0, 1, 1000, 1);
```

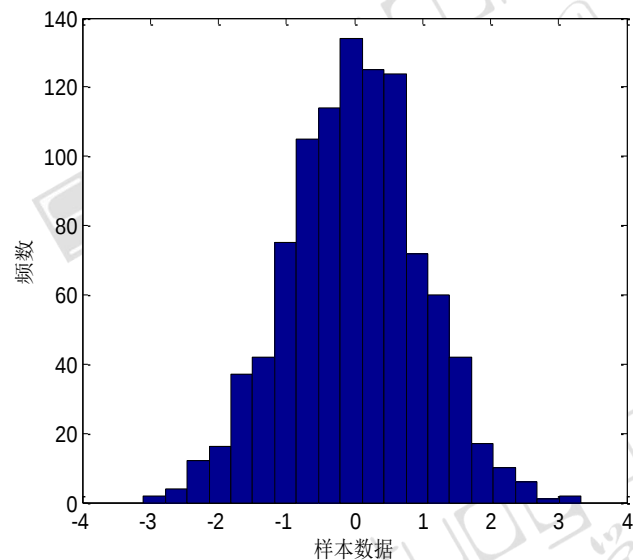
```
>> hist(x, 20);
```

```
>> xlabel('样本数据');
```

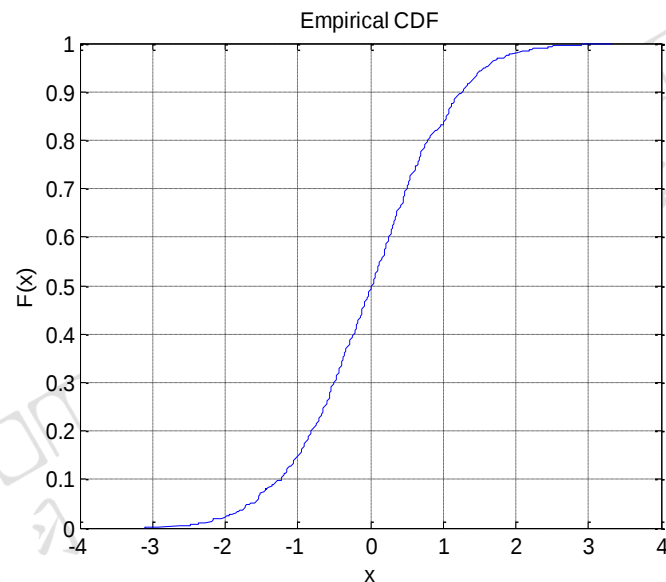
```
>> ylabel('频数');
```

```
>> figure;
```

```
>> cdfplot(x);
```



频数直方图



经验分布函数图

四、特殊二维绘图函数

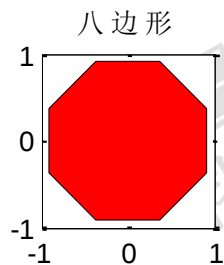
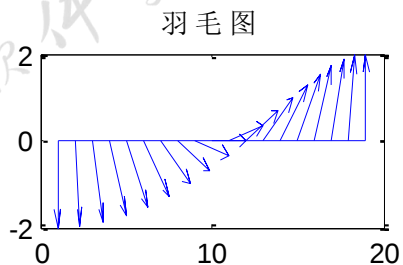
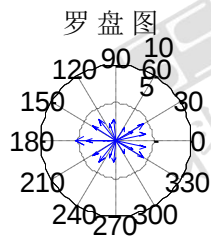
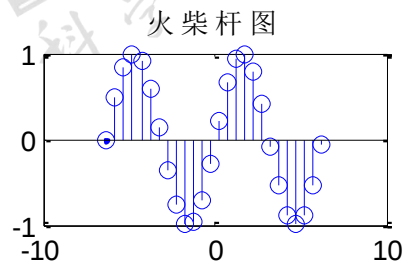
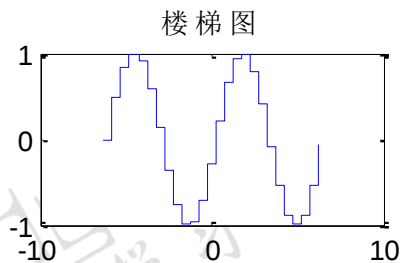
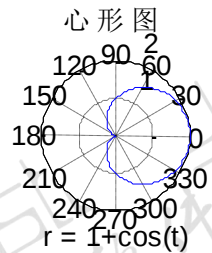
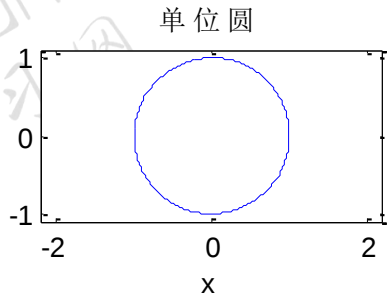
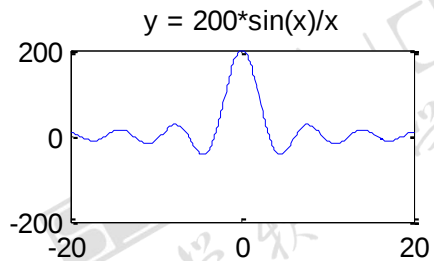
函数名	功能说明	函数名	功能说明
fplot	绘制函数图	comet	彗星图
ezplot	隐函数直角坐标绘图	compass	罗盘图
ezpolar	隐函数极坐标绘图	feather	羽毛图
pie	饼图	rose	玫瑰图
stairs	楼梯图	errorbar	误差柱图
stem	火柴杆图	pareto	Pareto（帕累托）图
bar	柱状图	fill	多边形填充图
barh	水平柱状图	patch	生成 patch 图形对象

【例5.2-7】特殊二维图形。

```
>> subplot(3, 3, 1);  
>> f = @(x)200*sin(x)./x;  
>> fplot(f, [-20 20]);  
>> title('y = 200*sin(x)/x');  
>> subplot(3, 3, 2);  
>> ezplot('x^2 + y^2 = 1', [-1.1 1.1]);  
>> axis equal;  
>> title('单位圆');  
>> subplot(3, 3, 3);  
>> ezpolar('1+cos(t)');  
>> title('心形图');  
>> subplot(3, 3, 4);  
>> x = [10 10 20 25 35];
```

```
>> name = {'赵', '钱', '孙', '李', '谢'};
>> explode = [0 0 0 0 1];
>> pie(x, explode, name)
>> title('饼图');
>> subplot(3, 3, 5);
>> stairs(-2*pi:0.5:2*pi, sin(-2*pi:0.5:2*pi));
>> title('楼梯图');
>> subplot(3, 3, 6);
>> stem(-2*pi:0.5:2*pi, sin(-2*pi:0.5:2*pi));
>> title('火柴杆图');
>> subplot(3, 3, 7);
>> Z = eig(randn(20,20));
>> compass(Z);
>> title('罗盘图');
```

```
>> subplot(3, 3, 8);  
>> theta = (-90:10:90)*pi/180;  
>> r = 2*ones(size(theta));  
>> [u,v] = pol2cart(theta,r);  
>> feather(u,v);  
>> title('羽毛图');  
>> subplot(3, 3, 9);  
>> t = (1/16:1/8:1)*2*pi;  
>> fill(sin(t), cos(t),'r');  
>> axis square; title('八边形');
```



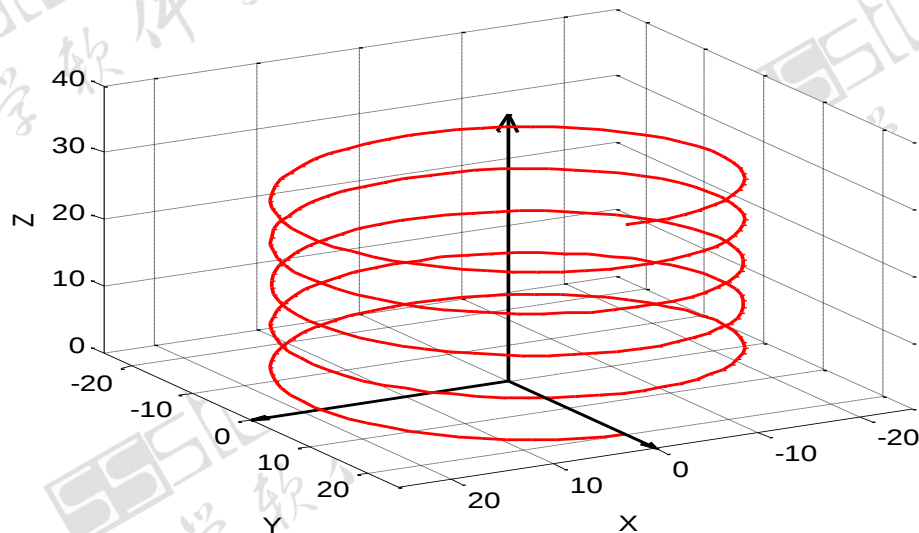
第三节 三维图形绘制

一、常用三维绘图函数

函数名	功能说明	函数名	功能说明
plot3	三维线图	sphere	单位球面
mesh	三维网格图	ellipsoid	椭球面
surf	三维表面图	quiver3	三维箭头
fill3	三维填充图	pie3	三维饼图
trimesh	三角网格图	bar3	竖直三维柱状图
trisurf	三角表面图	bar3h	水平三维柱状图
ezmesh	易用的三维网格绘图	stem3	三维火柴杆图
ezsurf	易用的三维彩色面绘图	contour	矩阵等高线图
meshc	带等高线的网格图	contour3	三维等高线图
surfc	带等高线的面图	contourf	填充二维等高线图
surfl	具有亮度的三维表面图	waterfall	瀑布图
hist3	三维直方图	pcolor	伪色彩图
slice	立体切片图	hidden	设置网格图的透明度
cylinder	圆柱面	alpha	设置图形对象的透明度

二、三维线图

【例5.3-1】用plot3函数绘制三维螺旋线。



```
>> t = linspace(0, 10*pi, 300);  
>> plot3(20*sin(t), 20*cos(t), t, 'r', 'linewidth', 2);  
>> hold on %图形保持  
>> quiver3(0,0,0,1,0,0,25,'k','filled','LineWidth',2);  
>> quiver3(0,0,0,0,1,0,25,'k','filled','LineWidth',2);  
>> quiver3(0,0,0,0,0,1,40,'k','filled','LineWidth',2);  
>> grid on  
>> xlabel('X'); ylabel('Y'); zlabel('Z');  
>> axis([-25 25 -25 25 0 40]);  
>> view(-210,30);
```

三、三维面图

1. 调用meshgrid函数划分网格

调用格式：

$[X,Y] = \text{meshgrid}(xgv,ygv)$

$[X,Y,Z] = \text{meshgrid}(xgv,ygv,zgv)$

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

$[X,Y,Z] = \text{meshgrid}(gv)$

【例5.3-2】用meshgrid函数生成网格矩阵，并

用plot函数画出平面网格图形。

```
>> [x,y] = meshgrid(1:4, 2:5)
```

```
x =
```

```
1 2 3 4
```

```
1 2 3 4
```

```
1 2 3 4
```

```
1 2 3 4
```

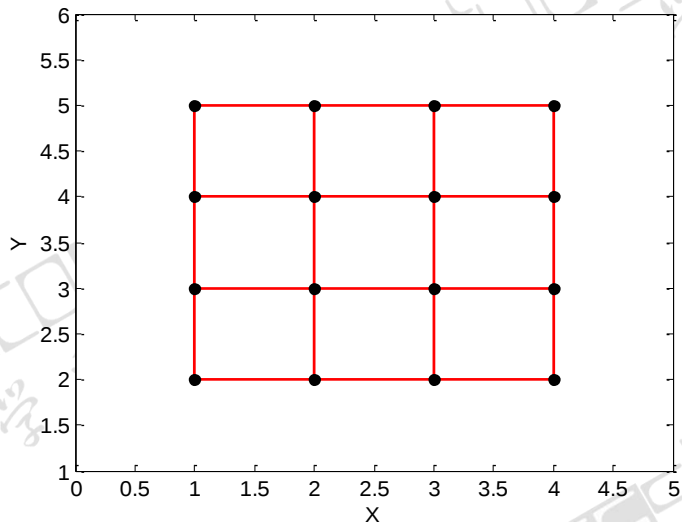
```
y =
```

```
2 2 2 2
```

```
3 3 3 3
```

```
4 4 4 4
```

```
5 5 5 5
```



```
>> plot(x, y, 'r','x', y, 'r', x, y, 'k.','markersize',18);
```

```
>> axis([0 5 1 6]);
```

```
>> xlabel('X'); ylabel('Y');
```

2. 调用mesh函数绘制网目图

调用格式：

`mesh(X,Y,Z)`

`mesh(Z)`

`mesh(...,C)`

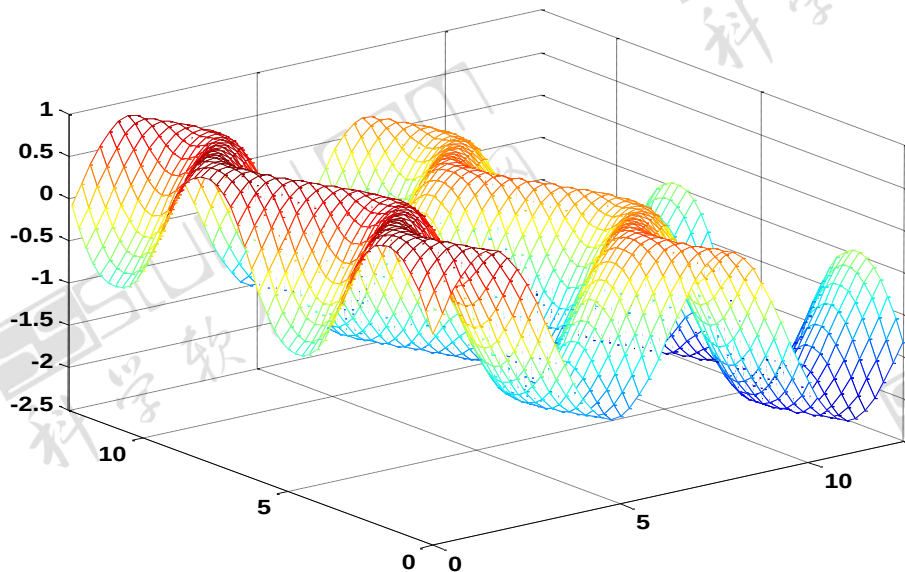
`mesh(...,'PropertyName',PropertyValue,...)`

`mesh(axes_handles,...)`

`h = mesh(...)`

【例5.3-3】绘制三维曲面图 $z=\sin(x+\sin(y))-x/10$ 。

```
>> [x,y]=meshgrid(0:0.25:4*pi);  
>> z=sin(x+sin(y))-x/10; mesh(x,y,z);  
>> axis([0 4*pi 0 4*pi -2.5 1]);
```



3. 调用surf函数绘制面图

调用格式：

`surf(Z)`

`surf(Z,C)`

`surf(X,Y,Z)`

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

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

`surf(axes_handles,...)`

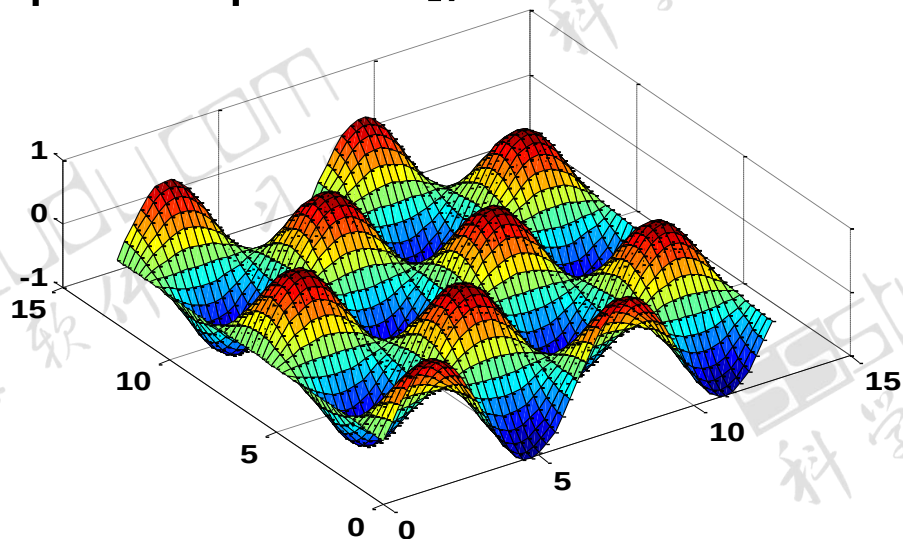
`h = surf(...)`

【例5.3-4】绘制三维曲面图 $z = \sin(x) \cdot \cos(y)$ 。

```
>> [x,y]=meshgrid(0:0.25:4*pi);
```

```
>> z=sin(x).*cos(y); surf(x,y,z);
```

```
>> axis([0 4*pi 0 4*pi -1 1]);
```



【例5.3-5】调用mesh , surf , surfl ,

surfc函数绘制peaks曲面。

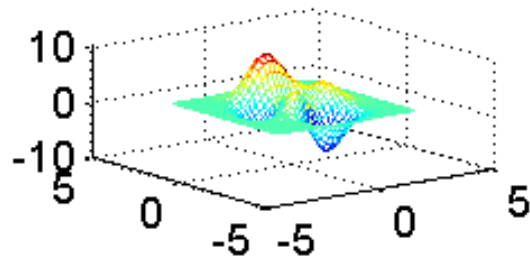
```
>> [X,Y,Z] = peaks(30);  
>> subplot(2, 2, 1);  
>> mesh(X, Y, Z);  
>> title('mesh');  
>> subplot(2, 2, 2);  
>> surf(X, Y, Z, 'FaceAlpha', 0.5);  
>> title('surf');  
>> subplot(2, 2, 3);  
>> surfl(X, Y, Z);  
>> title('surfl');
```

```
>> subplot(2, 2, 4);
```

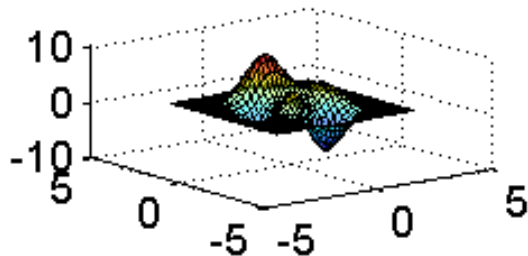
```
>> surfc(X, Y, Z);
```

```
>> title('surfc');
```

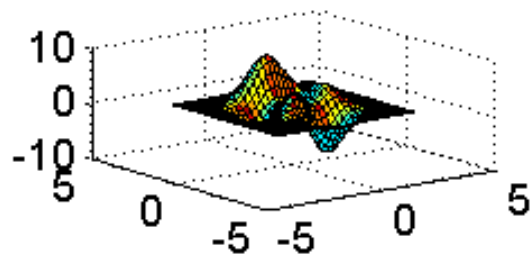
mesh



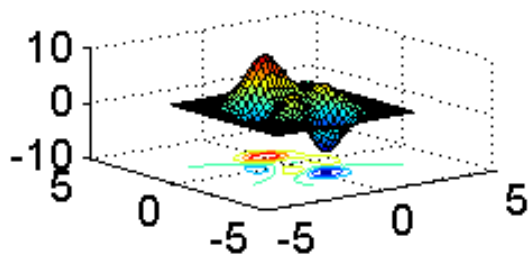
surf



surfl



surfc



【例5.3-6】绘制柱面，球面，椭球面。

% 绘制圆柱面

```
>> subplot(2,2,1);  
>> [x,y,z] = cylinder;  
>> surf(x,y,z);
```

% 绘制哑铃面

```
>> subplot(2,2,2);  
>> z = 0:pi/10:2*pi;  
>> y = 2+cos(z);  
>> [X,Y,Z] = cylinder(y);  
>> surf(X,Y,Z);
```

% 绘制球面，半径为10，球心 (1,1,1)

```
>> subplot(2,2,3);
```

```
>> [x,y,z] = sphere;
```

```
>> surf(10*x+1,10*y+1,10*z+1); axis equal;
```

% 绘制椭球面

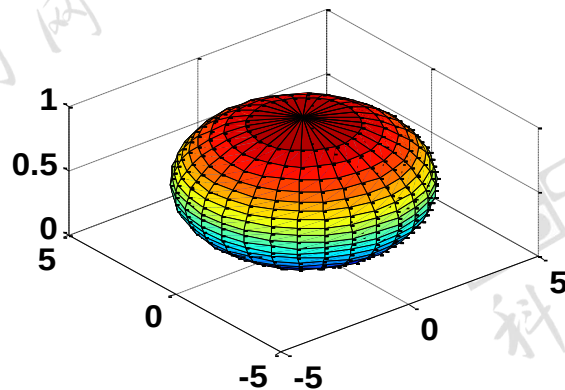
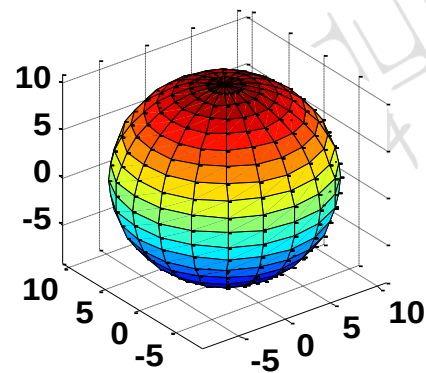
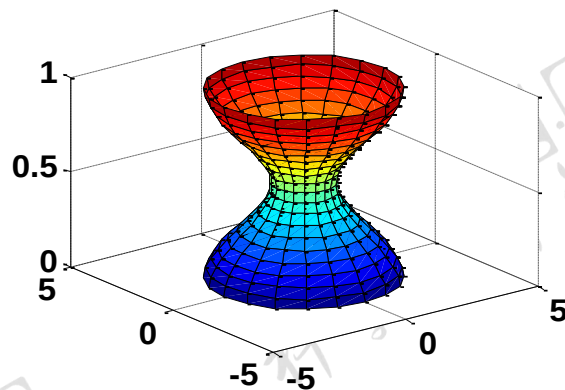
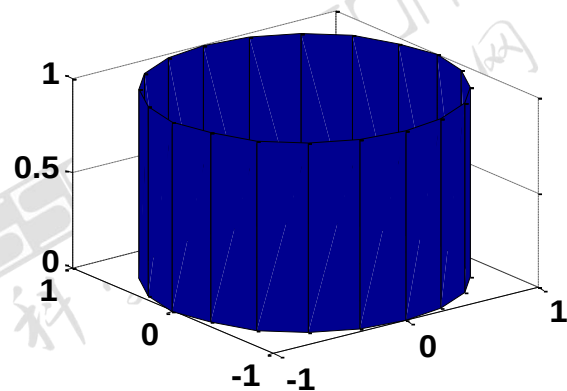
```
>> subplot(2,2,4);
```

```
>> a=4; b=3;
```

```
>> t = -b:b/10:b;
```

```
>> [x,y,z] = cylinder(a*sqrt(1-t.^2/b^2),30);
```

```
>> surf(x,y,z);
```



【例5.3-7】绘制三维饼图、三维柱状图、三维火柴杆图、三维填充图、三维向量场图和立体切片图（四维图）。

% 饼图

```
>> subplot(2,3,1);  
>> pie3([2347,1827,2043,3025]);  
>> title('三维饼图');
```

% 柱状图

```
>> subplot(2,3,2);  
>> bar3(magic(4));  
>> title('三维柱状图');
```

% 火柴杆图

```
>> subplot(2,3,3);  
>> y=2*sin(0:pi/10:2*pi);  
>> stem3(y);  
>> title('三维火柴杆图');
```

% 填充图

```
>> subplot(2,3,4);  
>> fill3(rand(3,5),rand(3,5),rand(3,5), 'y' );  
>> title('三维填充图');
```

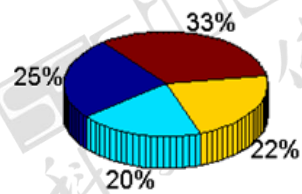
% 三维向量场图

```
>> subplot(2,3,5);  
>> [X,Y] = meshgrid(0:0.25:4,-2:0.25:2);  
>> Z = sin(X).*cos(Y);  
>> [Nx,Ny,Nz] = surfnorm(X,Y,Z);  
>> surf(X,Y,Z);  
>> hold on;  
>> quiver3(X,Y,Z,Nx,Ny,Nz,0.5);  
>> title('三维向量场图');  
>> axis([0 4 -2 2 -1 1]);
```

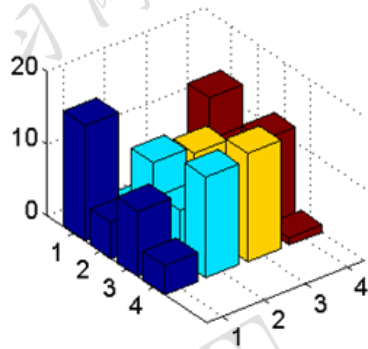
% 立体切片图（四维图）

```
>> subplot(2,3,6);  
>> t = linspace(-2,2,20);  
>> [X,Y,Z] = meshgrid(t,t,t);  
>> V = X.*exp(-X.^2-Y.^2-Z.^2);  
>> xslice = [-1.2,8,2];  
>> yslice = 2;  
>> zslice = [-2,0];  
>> slice(X,Y,Z,V,xslice,yslice,zslice);  
>> title('立体切片图（四维图）');
```

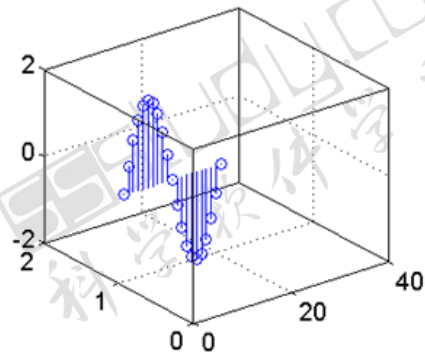
三维饼图



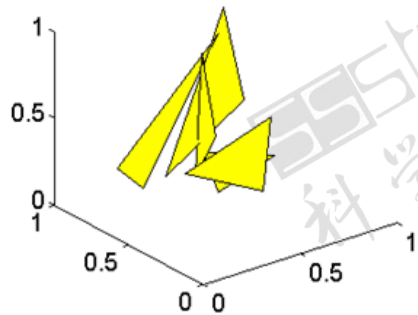
三维柱状图



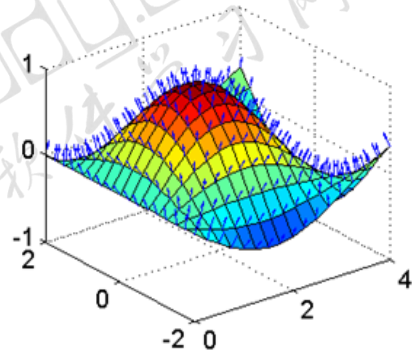
三维火柴杆图



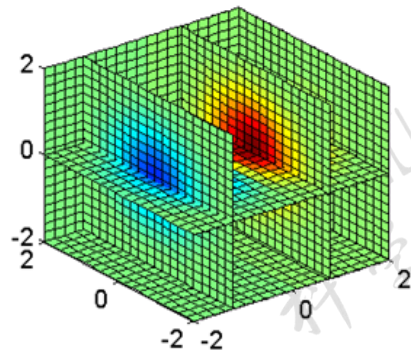
三维填充图



三维向量场图



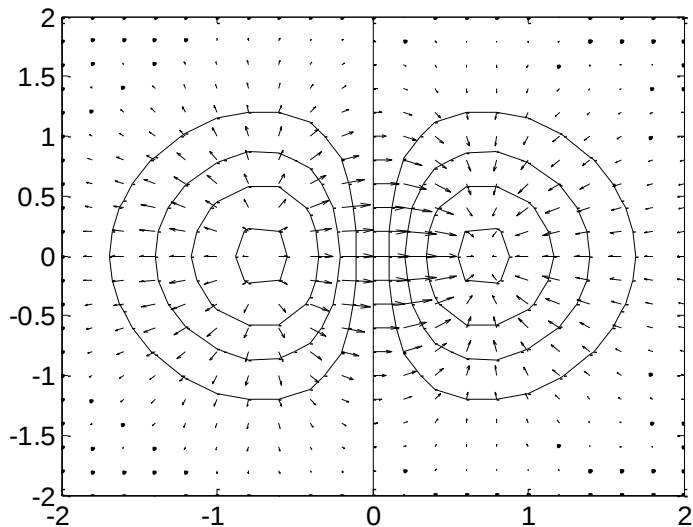
立体切片图（四维图）



【例5.3-8】绘制三维曲面 $z = xe^{-(x^2+y^2)}$

的等高线和梯度场。

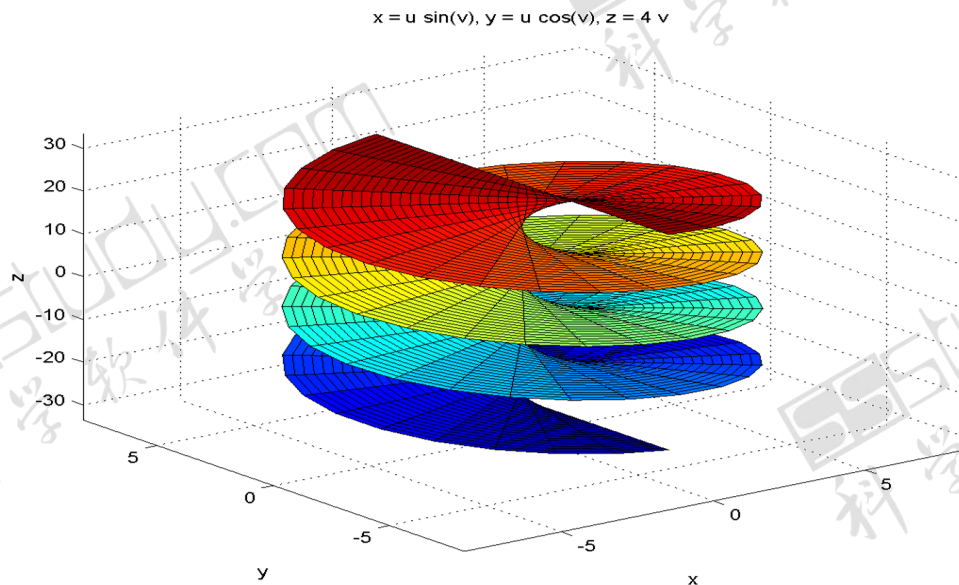
```
>> [X,Y] = meshgrid(-2:.2:2);  
>> Z = X.*exp(-X.^2 - Y.^2);  
>> [DX,DY] = gradient(Z,0.2,0  
>> contour(X,Y,Z) ;  
>> hold on ;  
>> quiver(X,Y,DX,DY) ;  
>> h = get(gca,'Children');  
>> set(h, 'Color','k');
```



【例5.3-9】调用ezsurf函数绘制螺面

$$\begin{cases} x = u \sin v \\ y = u \cos v \\ z = 4v \end{cases}$$

```
>> ezsurf('u*sin(v)', 'u*cos(v)', '4*v', [-2*pi, 2*pi, -2*pi, 2*pi])
```



四、三维图形的修饰和添加注释

1. 绘图色彩的调整

MATLAB中提供了colormap函数，可以根据颜色映像矩阵对图形对象的色彩进行调整。所谓的颜色映像矩阵就是一个 $k \times 3$ 的矩阵， k 行表示有 k 种颜色，每行3个元素分别代表红、绿、蓝三元色的灰度值，取值均在 $[0, 1]$ 之间。

colormap 函数调用格式

- colormap(map), 设置map为当前颜色映像矩阵
- colormap('default'), 恢复当前颜色映像矩阵为默认值
- cmap = colormap, 获取当前颜色映像矩阵
- colormap(ax,...), 设置当前axes对象的颜色映像矩阵

注：map的设置有两种，可以人为指定一个元素值均在[0, 1]之间的矩阵，也可以用MATLAB自带的17种颜色映像矩阵。在MATLAB命令窗口分别运行autumn、bone、colorcube、cool、copper、flag、gray、hot、hsv、jet、lines、pink、prism、spring、summer、white和winter函数，就可得到这17种颜色映像矩阵

2. 着色方式调整

shading 函数

- shading flat , 平面着色 , 同一个小网格面和相应的线段用同一种颜色着色。
- shading faceted , 类似于shading flat , 平面着色 , 只是网格线都用黑色 , 这是默认着色方式。
- shading interp , 通过颜色插值方式着色。
- shading(axes_handle,...) , 为句柄值为 axes_handle 的坐标系内的图形对象设置着色方式。

3. 透明度调整

➤ 通过图形对象的 “FaceAlpha” 属性调整透明度

调用：`set(h,'FaceAlpha',0.25);`

说明：h为图形对象句柄，0.25是
“FaceAlpha” 属性的属性值，由用户随意
指定。

3. 透明度调整

➤ 利用alpha函数调整透明度

调用：`alpha(alpha_data)`

说明：`alpha_data`是一个介于0和1之间的数，`alpha_data = 0`表示完全透明，`alpha_data = 1`表示完全不透明，`alpha_data` 的值越接近于0，透明度越高

3. 透明度调整

➤ 调用hidden函数调整网目图的遮挡关系

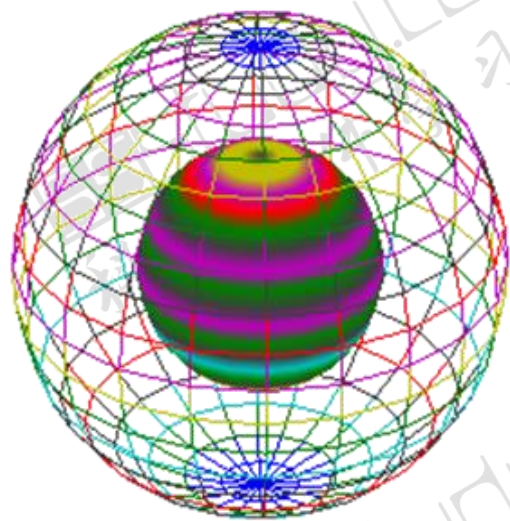
调用：`hidden on`

`hidden off`

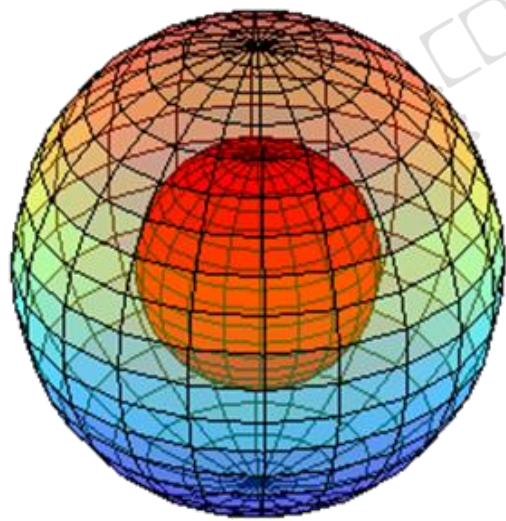
说明：`hidden off`用来取消遮挡，透明效果。

【例5.3-10】 三维图形的透视效果

```
>> figure;  
>> [X,Y,Z] = sphere;  
>> surf(X,Y,Z);  
>> colormap(lines);  
>> shading interp  
>> hold on;  
>> mesh(2*X,2*Y,2*Z)  
>> hidden off  
>> axis equal  
>> axis off  
>> figure;  
>> surf(X,Y,Z,'FaceColor','r');  
>> hold on;  
>> surf(2*X,2*Y,2*Z,'FaceAlpha',0.4);  
>> axis equal  
>> axis off
```



(a) 网格图的透视效果图



(b) 面图的透视效果图

网格图和面图的透视效果图

4. 光源设置与属性调整

light函数

➤ `light('PropertyName',propertyvalue,...)`

建立一个光源，并设置光源属性（Position、Color和Style）。

➤ `handle = light(...)`

建立一个光源，并获取其句柄值handle。

5. 调整光照模式

lighting函数

- lighting flat , 产生均匀光照（默认）。
- lighting gouraud , 计算顶点法线并作线性插值修改表面颜色。
- lighting phong , 做线性插值并计算每个像素的反射率来修改表面颜色。
- lighting none , 关掉照明。

6. 图形表面对光照反射属性设置

material函数

- material shiny , 相对较高的镜面反射。
- material dull , 更多的漫反射。
- material metal , 使图形对象有非常高的镜面反射和非常低的环境光及漫反射。

➤ material([ka kd ks])

➤ material([ka kd ks n])

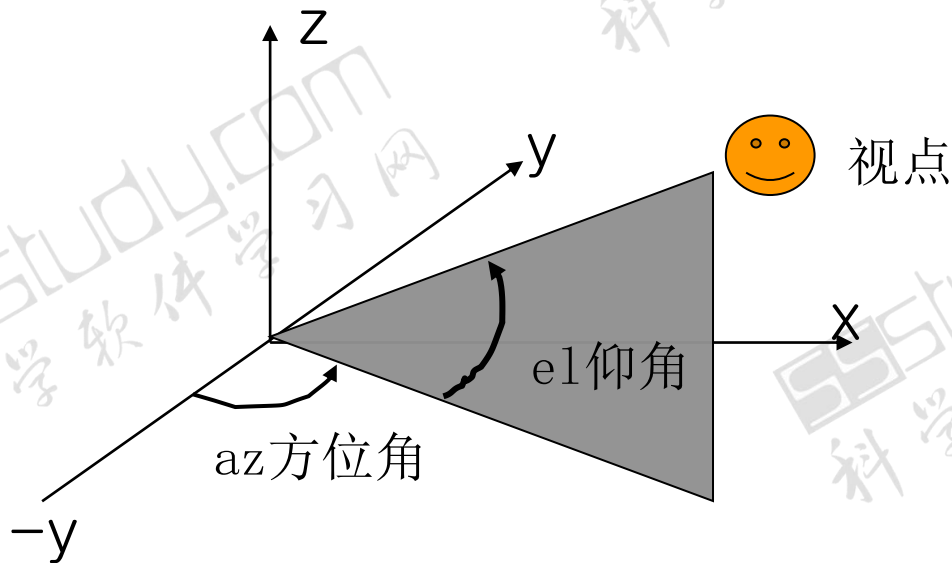
➤ material([ka kd ks n sc])，用ka、kd和ks分别设置图形对象的环境光、漫反射和镜面反射的强度，用镜面指数n控制镜面亮点的大小，用sc设置镜面颜色的反射系数。

➤ material default，恢复ka、kd、ks、n和sc的默认值。

7. 调整视点位置

view函数

➤ $\text{view}(\text{az}, \text{el})$ ，设置三维绘图的视角， az 是方位角， el 表示相对于 xoy 平面的仰角。

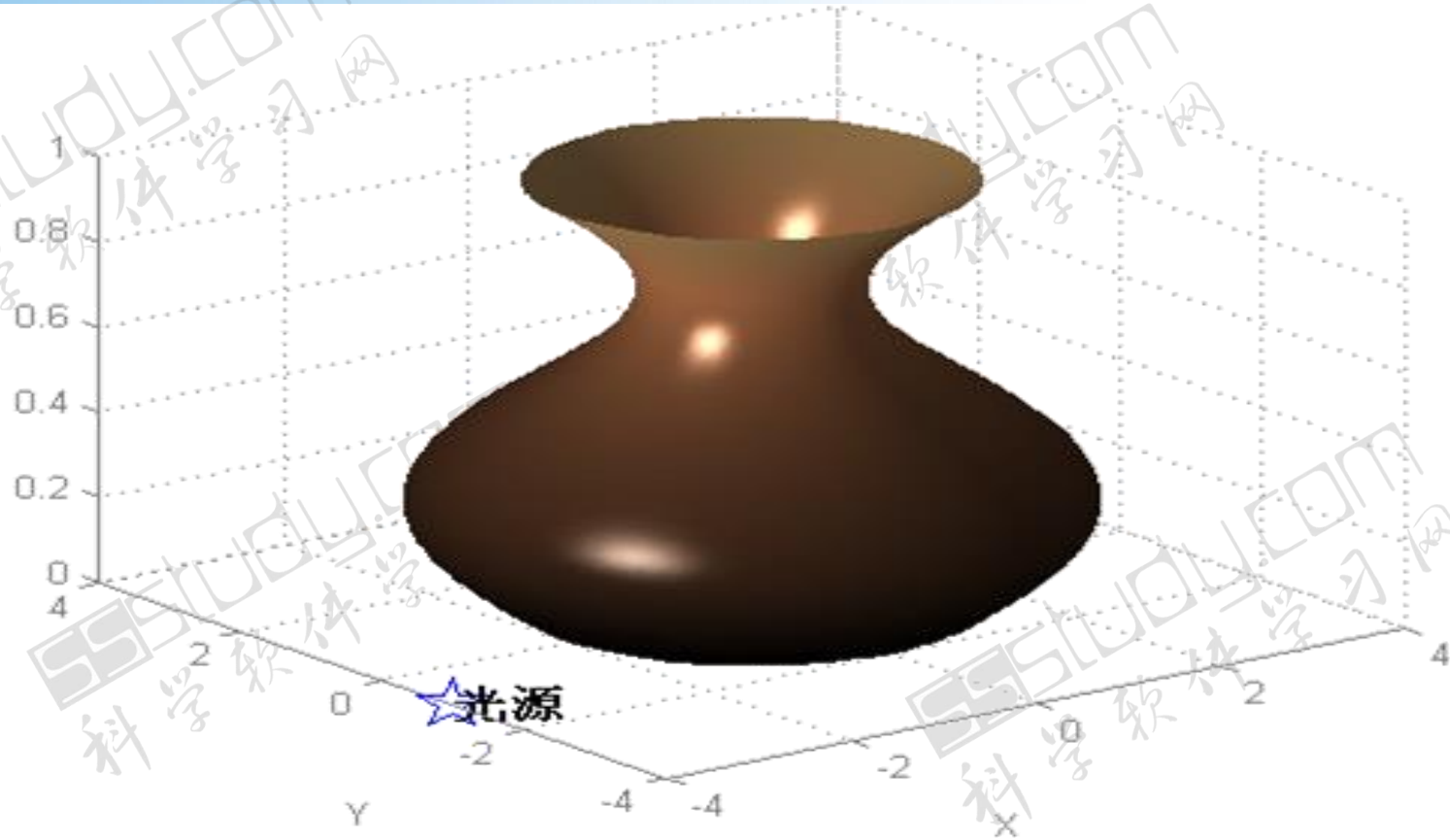


- `view([x,y,z])` , 设置视点的三维直角坐标 $[x, y, z]$.
- `view(2)` , 默认的二维视角 , $az = 0$, $el = 90$.
- `view(3)` , 默认的三维视角 , $az = -37.5$, $el = 30$.
- `view(ax,...)` , 设置句柄值为 ax 的坐标系的视角。
- `[az,el] = view` , 返回当前方向角和仰角。

【例5.3-12】 绘制带光照的花瓶。

```
>> t = 0:pi/20:2*pi;  
>> [x,y,z] = cylinder(2+sin(t), 100);  
>> surf(x,y,z);  
>> xlabel('X'); ylabel('Y'); zlabel('Z');  
>> set(gca,'color','none');  
>> shading interp;  
>> colormap(copper);  
>> light('Posi',[-4 -1 0]);  
>> lighting phong;  
>> material metal;  
>> hold on;  
>> plot3(-4,-1,0,'p','markersize', 18);  
>> text(-4,-1,0,'光源','fontsize',14,'fontweight','bold');
```

SSStudy.com
科学软件学习网



SSStudy.com
科学软件学习网

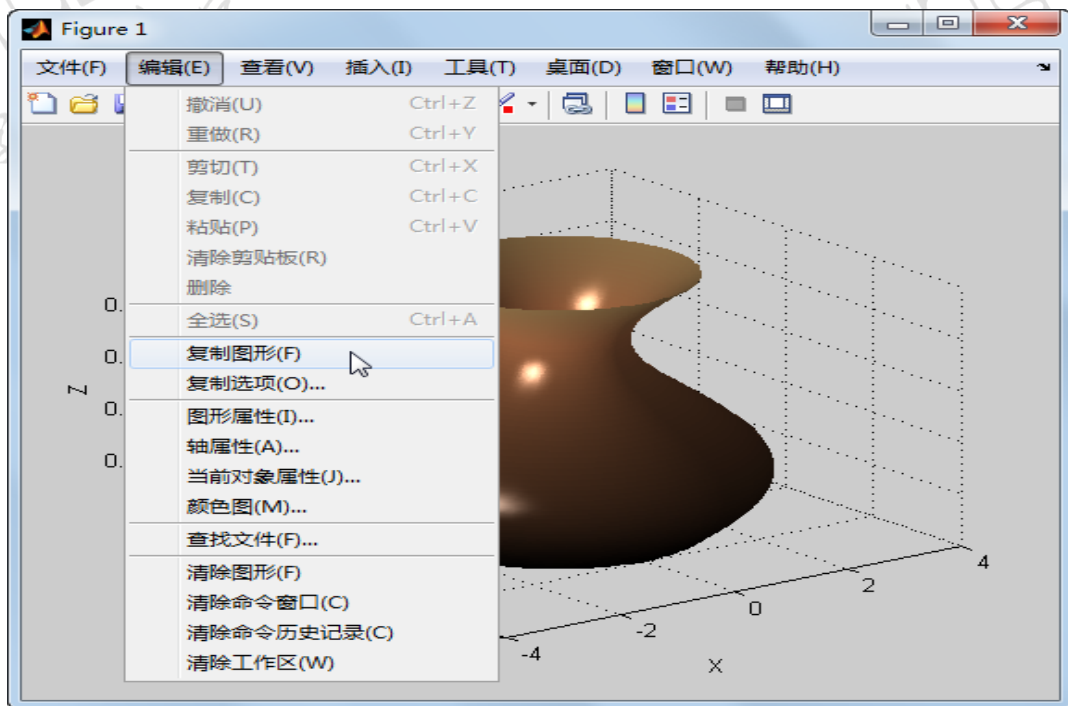
SSStudy.com
科学软件学习网

SSStudy.com
科学软件学习网

第四节 图形的打印和输出

一、把图形复制到剪贴板

1. 界面操作



2. 利用MATLAB命令进行复制操作

hgexport函数调用格式：

➤ hgexport(h,filename)

把句柄值为h的图形窗口中的图形写入默认的eps格式文件。

➤ hgexport(h,'-clipboard')

把句柄值为h的图形窗口中的图形复制到Windows剪贴板。

【例5.4-1】调用hgexport函数复制图形到剪贴板。

```
>> h = figure;  
>> ezplot('sin(x) ',[0,2*pi]);  
>> hgexport(h, '-clipboard');
```

二、把图形导出到文件

1. 界面操作

在MATLAB中通过界面操作可以很方便地把图形窗口中的图形保存为各种标准格式的图像文件。图形窗口的File菜单下有Save、Save As和Export Setup三个选项，均可用来将图形窗口中的图形导出到文件。

2. 利用MATLAB命令把图形导出到文件

`saveas`函数调用格式：

➤ `saveas(h,'filename.ext')`

把句柄值为h的图形或Simulink模块图保存为文件filename.ext。

➤ `saveas(h,'filename','format')`

把句柄值为h的图形或Simulink模块图按指定格式保存为文件filename。参数format是字符串，用来指明文件扩展名。

【例5.4-2】调用saveas函数保存图形。

```
>> h = figure;  
>> ezplot('sin(x) ',[0,2*pi]);  
>> saveas (h, '示例2.jpg')
```

三、打印图形

1. 界面操作

在MATLAB中绘制好图形之后，单击figure窗口File菜单下的Print Preview或Print选项进行打印设置或打印。

2. 利用MATLAB命令进行打印操作

➤ print函数

调用格式：

print

print('argument1','argument2',...)

print(handle,'filename')

print argument1 argument2 ... argumentn

[pcmd,dev] = printopt

【例5.4-3】调用print函数打印图形。

```
>> x = 0 : 0.25 : 2*pi;
```

```
>> y = sin(x);
```

```
>> h = plot(x, y);
```

```
>> print; % 打印图形
```

```
>> print -dmeta
```

```
>> print -djpeg heping.jpg
```

Thank You