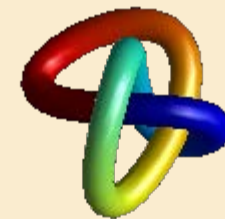
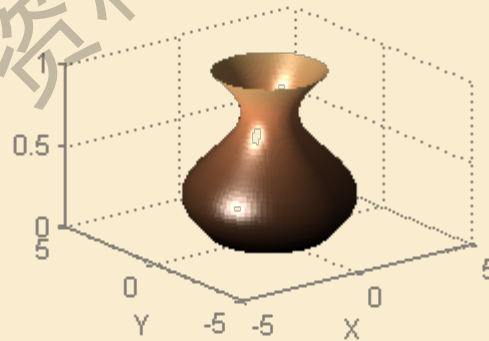
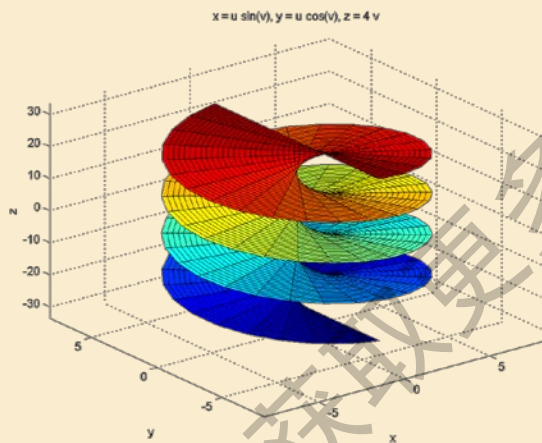
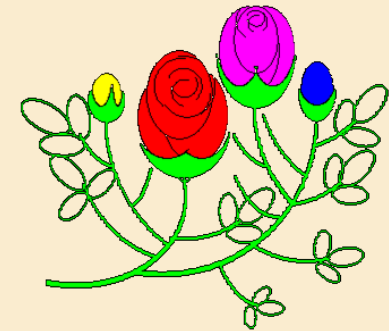
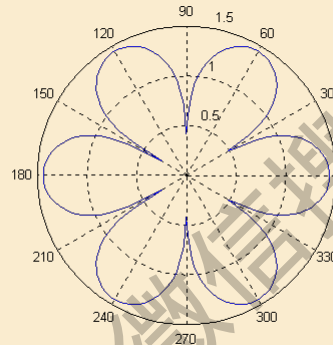
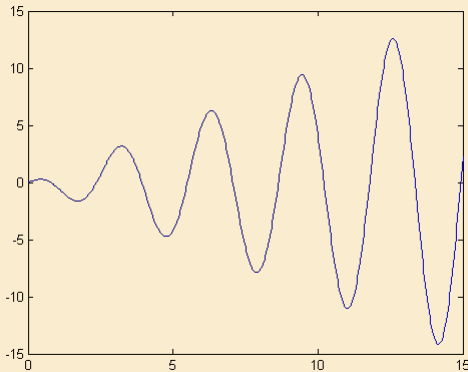


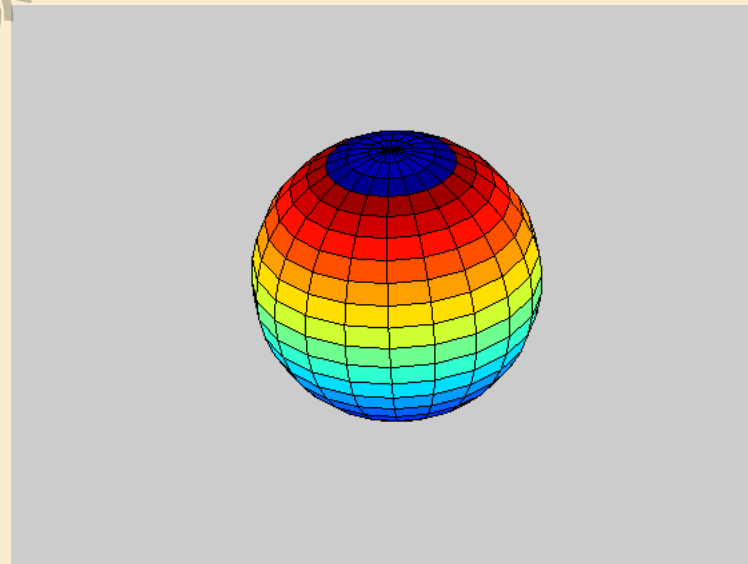
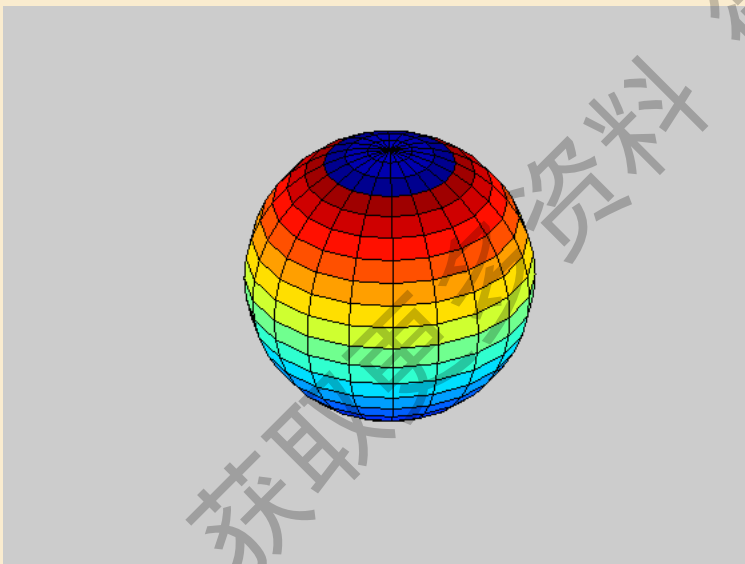
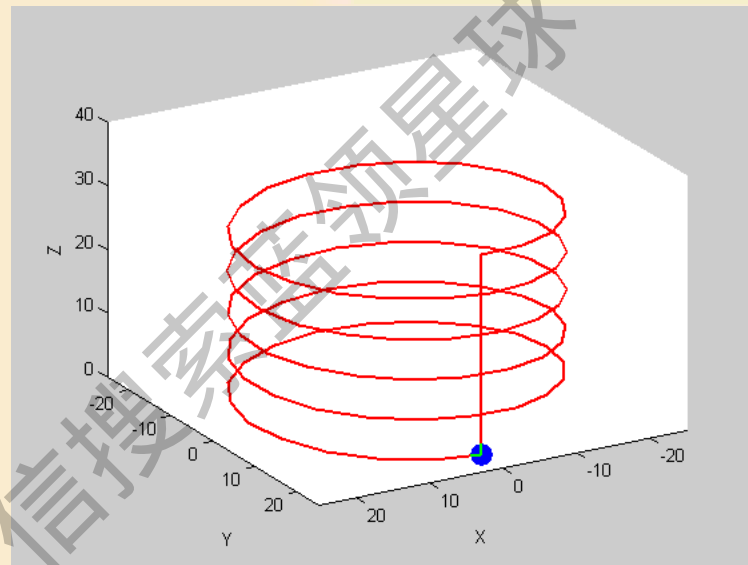
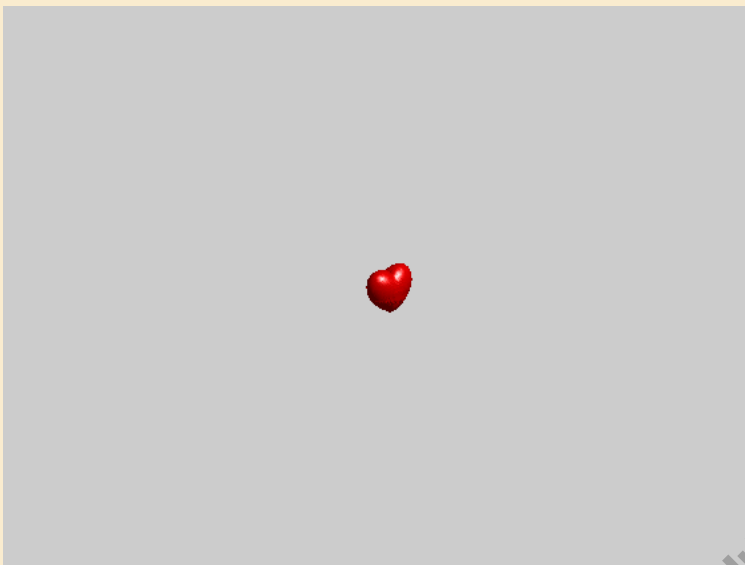
MATLAB绘图与可视化



获取更多资料 微信搜索蓝墨星球

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
- 动画制作

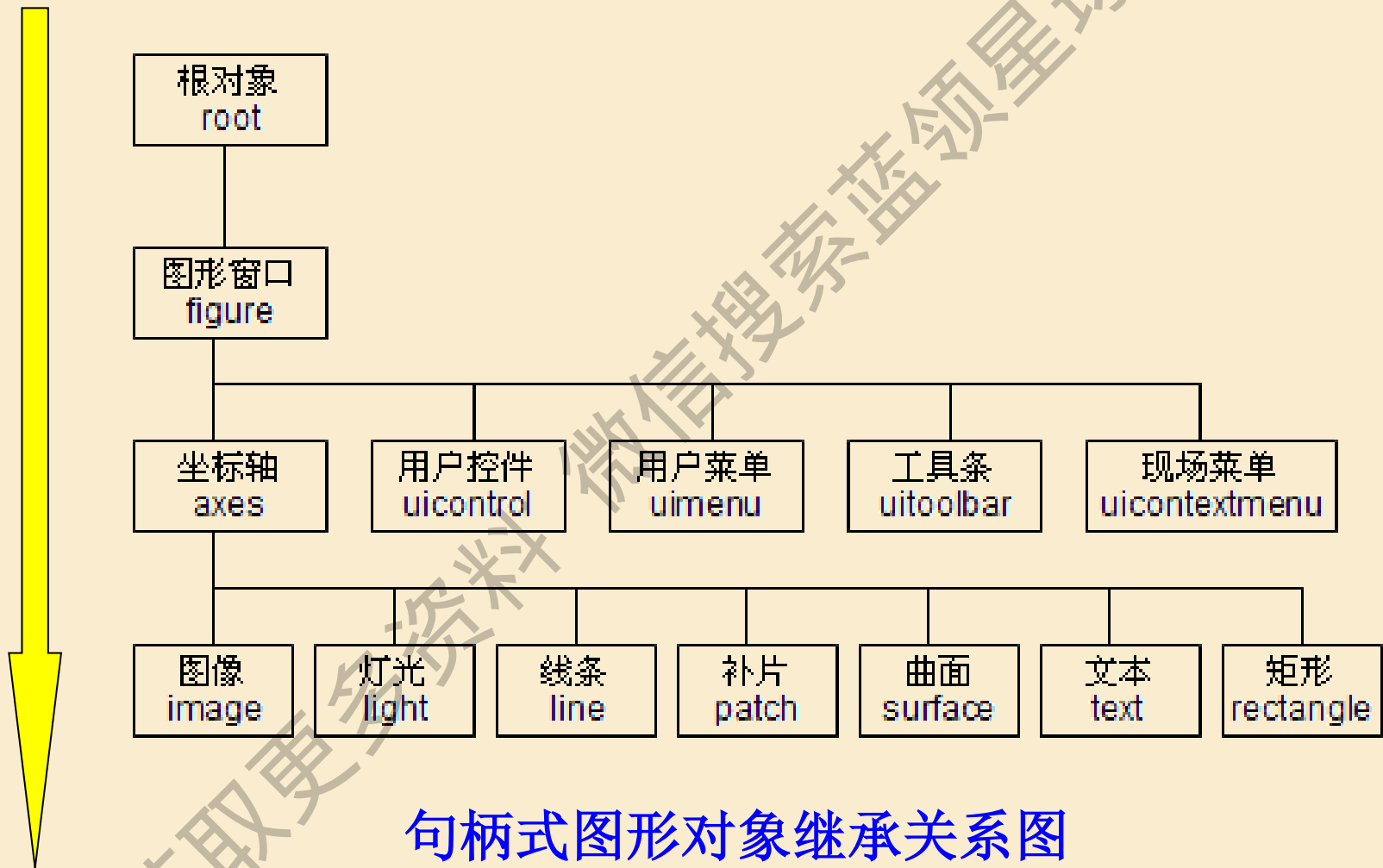
获取更多资料，微信搜索：蓝领星球

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，所有这些图形对象都可以通过一个被称为“句柄值”的东西加以控制，例如可以通过一个线条的句柄值来修改线条的颜色、宽度和线型等属性。这里所谓的“句柄值”其实就是一个数值，每个图形对象都对应一个唯一的句柄值，它就像一个指针，与图形对象一一对应。例如可以通过命令 `h = figure` 返回一个图形窗口的句柄值。

Parent (父)



Children (子)

句柄式图形对象继承关系图

二、获取图形对象属性名称和属性值

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

```
h =
```

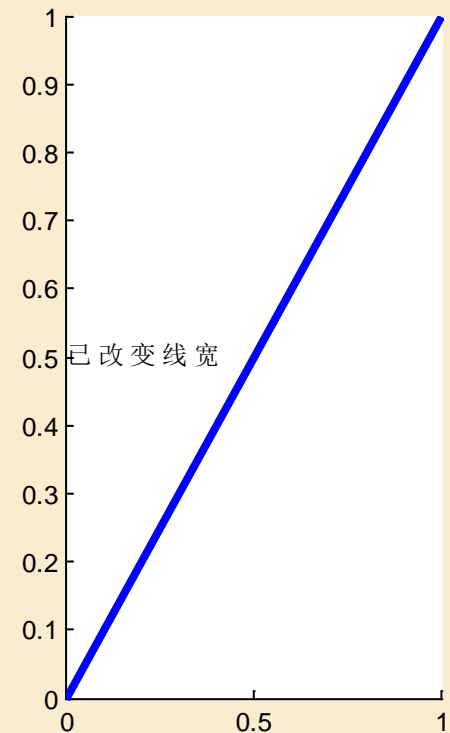
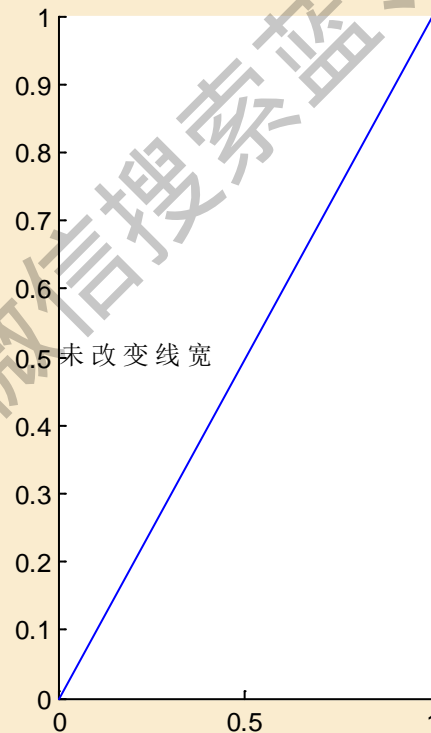
```
0.0149
```

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

获取更多资料 微信搜索蓝领星球

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

获取更多资料

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,...)

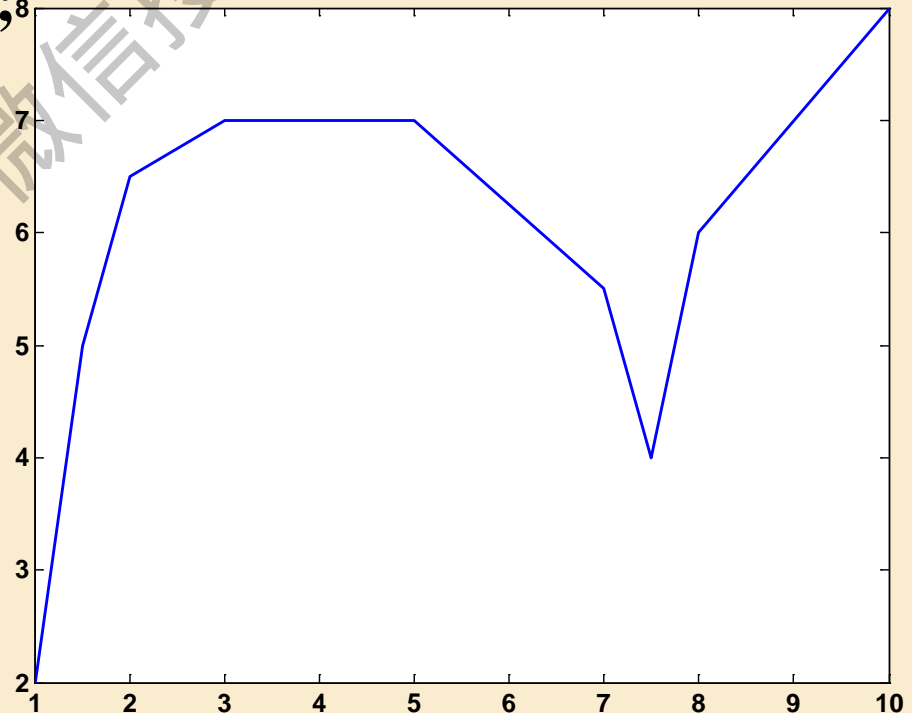
h = plot(...)

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

【例5.2-1】 给定横纵坐标数据，绘制线图

x	1	1.5	2	3	5	7	7.5	8	10
y	2	5	6.5	7	7	5.5	4	6	8

```
>> x = [1 1.5 2 3 5 7 7.5 8 10];  
>> y = [2 5 6.5 7 7 5.5 4 6 8];  
>> plot(x,y)
```



➤ 线条属性的修改

线型、描点类型、颜色参数表

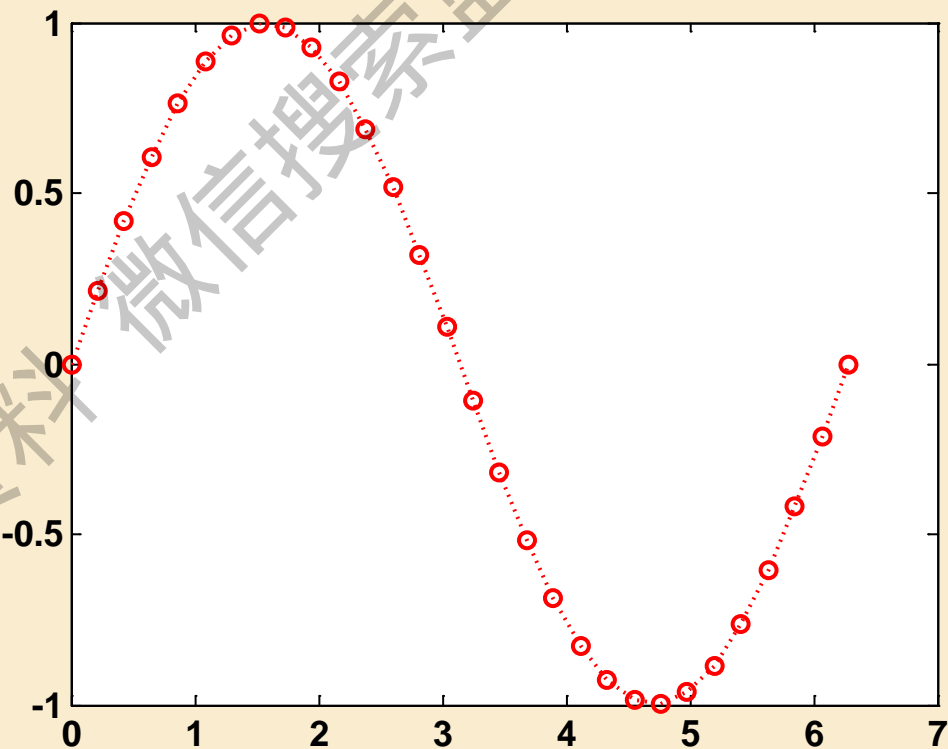
色彩字符	说明	线型字符	说明	描点类型	说明	描点类型	说明
r	红	-	实线（默认）	.	点	<	左三角形
g	绿	--	虚线	o	圆	s	方形
b	蓝	:	点线	x	叉号	d	菱形
c	青	-.	点划线	+	加号	P	五角星
m	品红			*	星号	h	六角星
y	黄			v	下三角形		
k	黑			^	上三角形		
w	白			>	右三角形		

【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',...
```

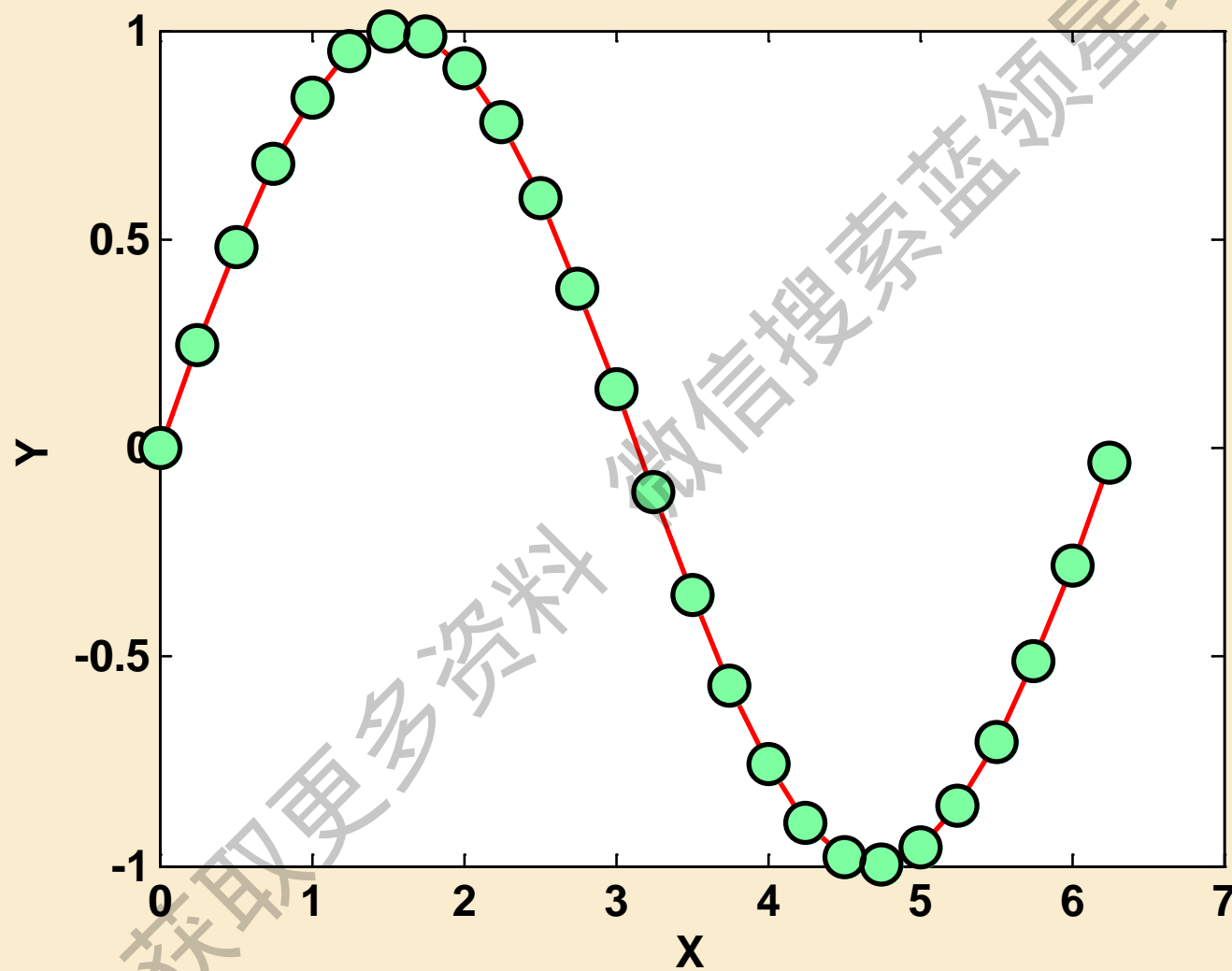
```
    'LineWidth',2,...
```

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

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

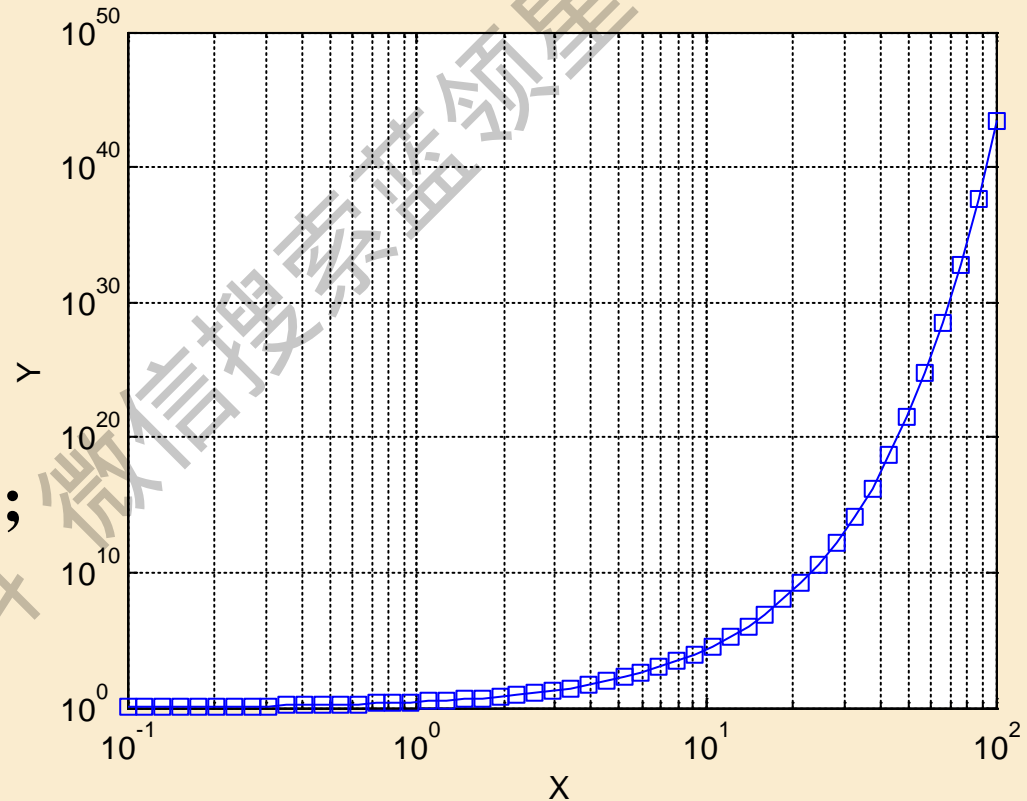
```
    'MarkerSize',12)
```

```
>> 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  
% 为X轴， Y轴加标签  
>> xlabel('X'); ylabel('Y');
```



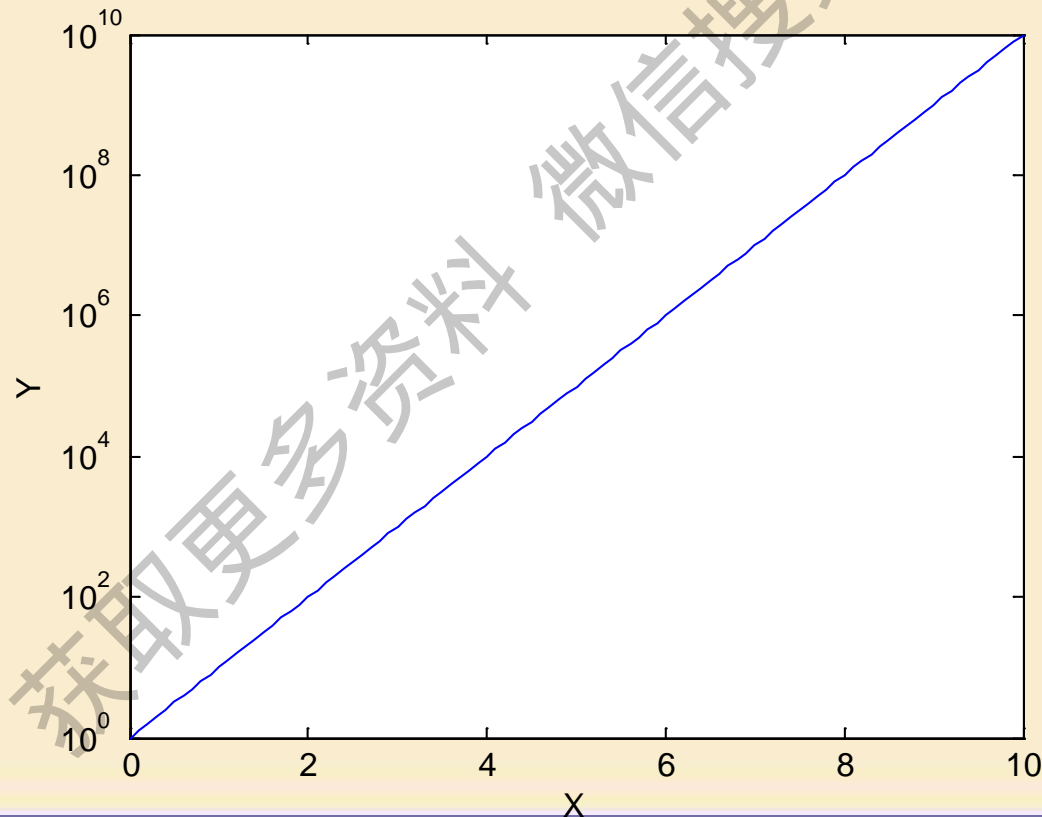
获取更多资料

3. semilogx, semilogy函数：半对数坐标绘图

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

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

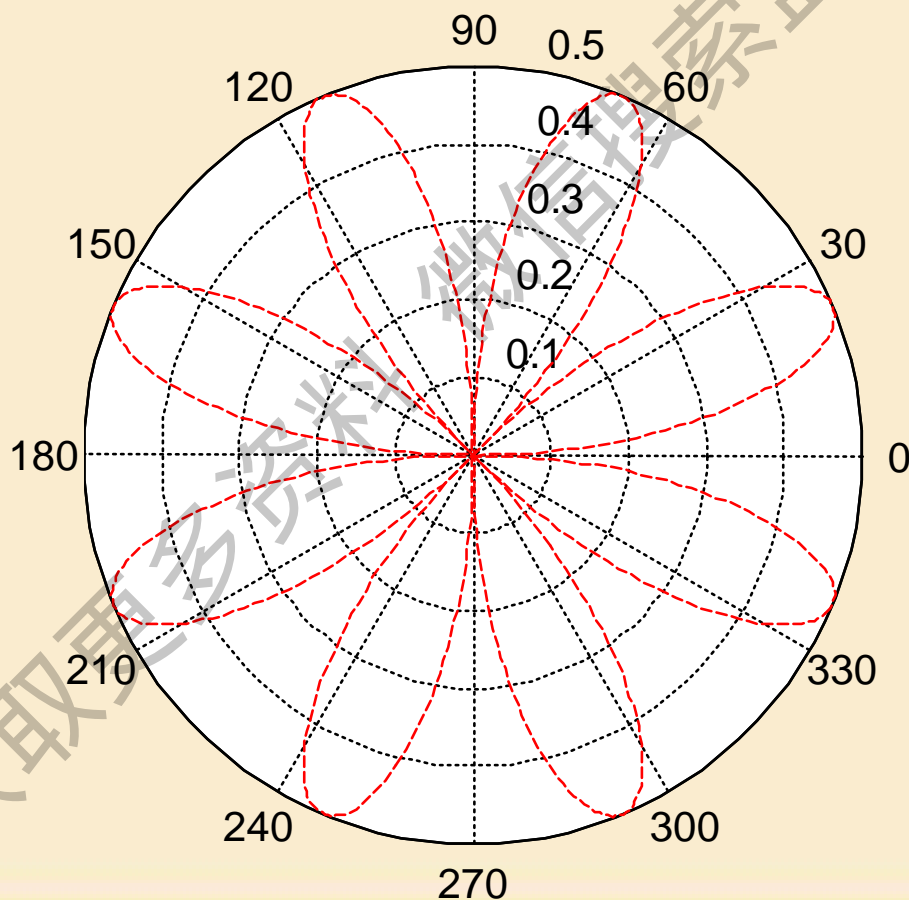
```
>> xlabel('X'); ylabel('Y');
```



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);
```

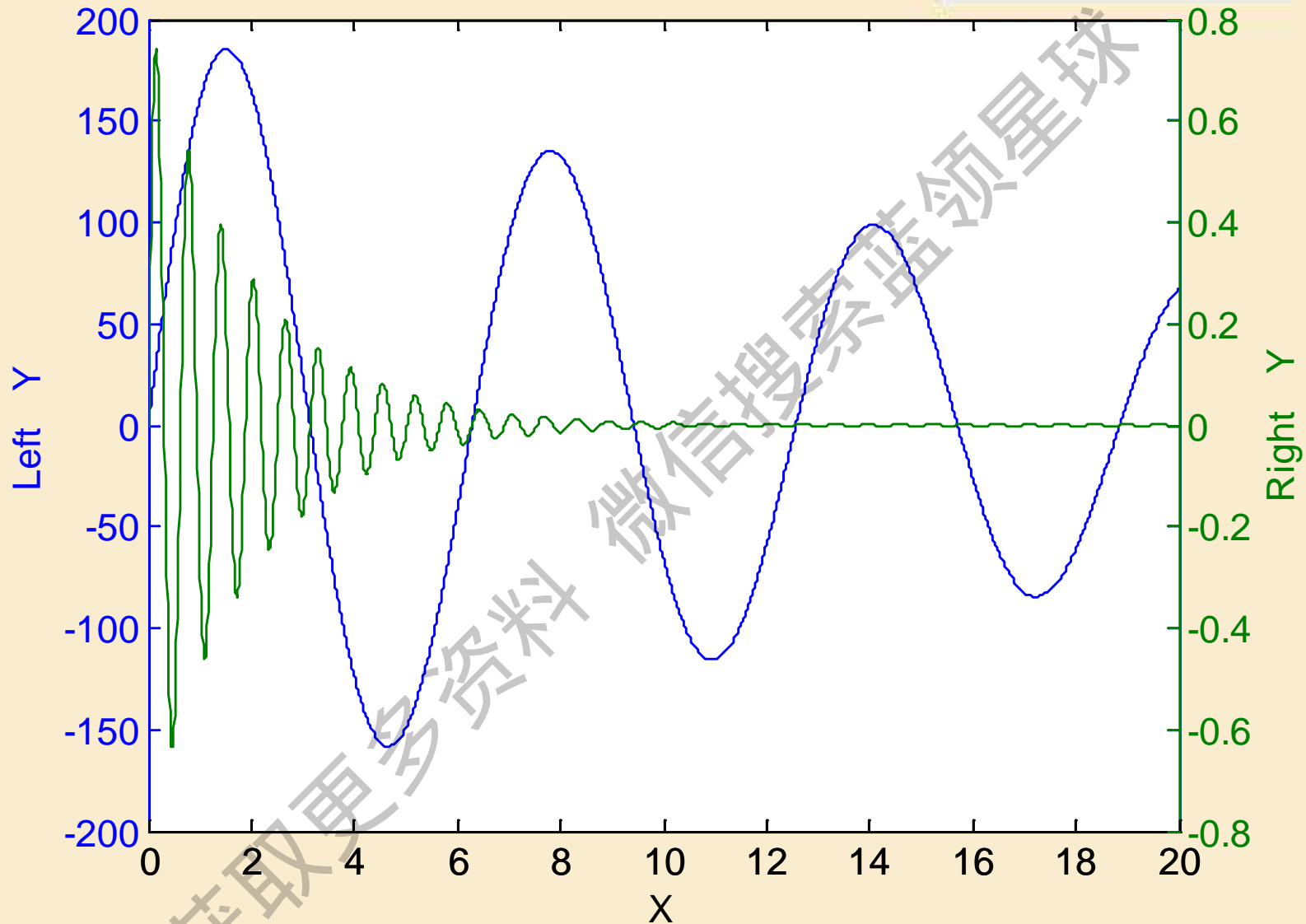
```
>> y2 = 0.8*exp(-0.5*x).*sin(10*x);
```

```
>> 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');
```

获取更多资料 微信搜索 蓝领星球



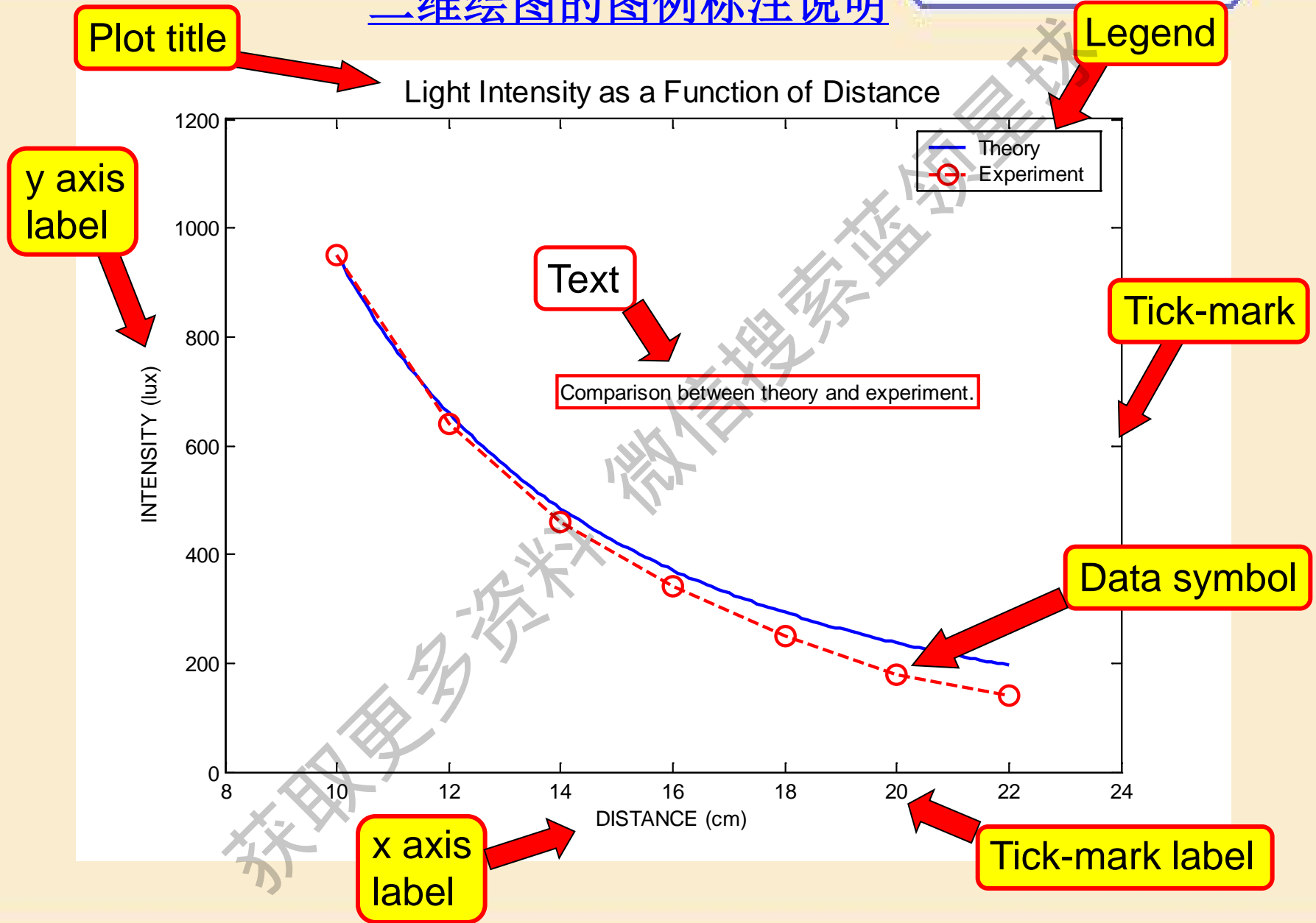
双纵坐标绘图

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

获取更多资料 微信搜索 谢中华MATLAB应用培训

二维绘图的图例标注说明



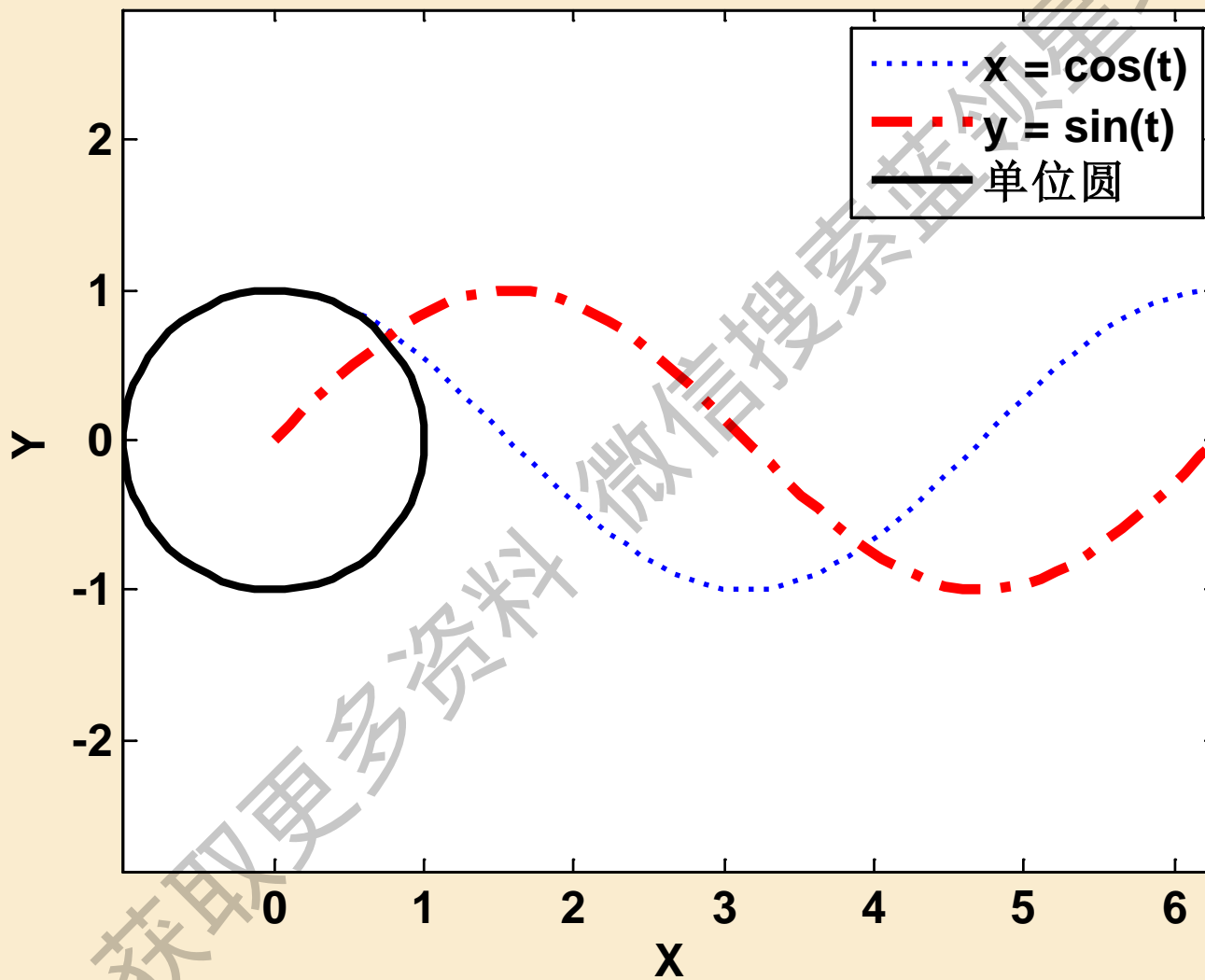
- **hold** 函数：开启和关闭图形窗口的图形保持功能
- **axis** 函数：设置坐标系的刻度和显示方式
- **box** 函数：显示或隐藏坐标边框
- **grid** 函数：为当前坐标系添加网格
- **title** 函数：为当前坐标系添加标题
- **xlabel**和**ylabel** 函数：为当前坐标轴添加标签
- **text**函数：在当前坐标系中添加文本对象（**text**对象）
- **gtext**函数：在当前坐标系中交互式添加文本对象
- **legend**函数：在当前坐标系中添加**line**对象和**patch**对象的图形标注框

- **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

获取更多资料 微信搜索 蓝星

【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。

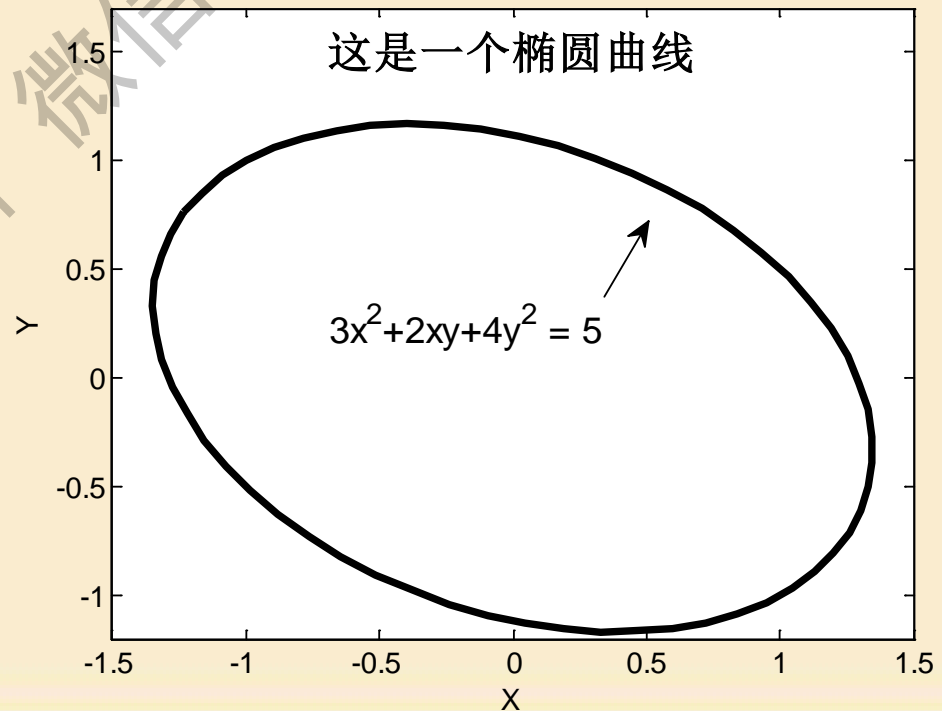
```
>> t = linspace(0,2*pi,60);  
>> x = cos(t);  
>> y = sin(t);  
>> plot(t,x,':','LineWidth',2);  
>> hold on;  
>> plot(t,y,'r-.','LineWidth',3);  
>> plot(x,y,'k','LineWidth',2.5);  
>> axis equal;  
>> xlabel('X');  
>> ylabel('Y');  
>> legend('x = cos(t)', 'y = sin(t)', '单位圆', 'Location', 'NorthEast');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



【例5.2-4】 根据椭圆方程 $(x \ y) \begin{pmatrix} 3 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 5$ 绘制椭圆曲线，并修饰图形。

```
>> P = [3 1; 1 4];  
>> r = 5;  
>> [V, D] = eig(P);  
>> a = sqrt(r/D(1));  
>> b = sqrt(r/D(4));  
>> t = linspace(0, 2*pi, 60);  
>> xy = V*[a*cos(t); b*sin(t)];  
>> plot(xy(1,:),xy(2,:), 'k', 'linewidth', 3);  
>> h = annotation('textarrow',[0.606 0.65],[0.55 0.65]);
```

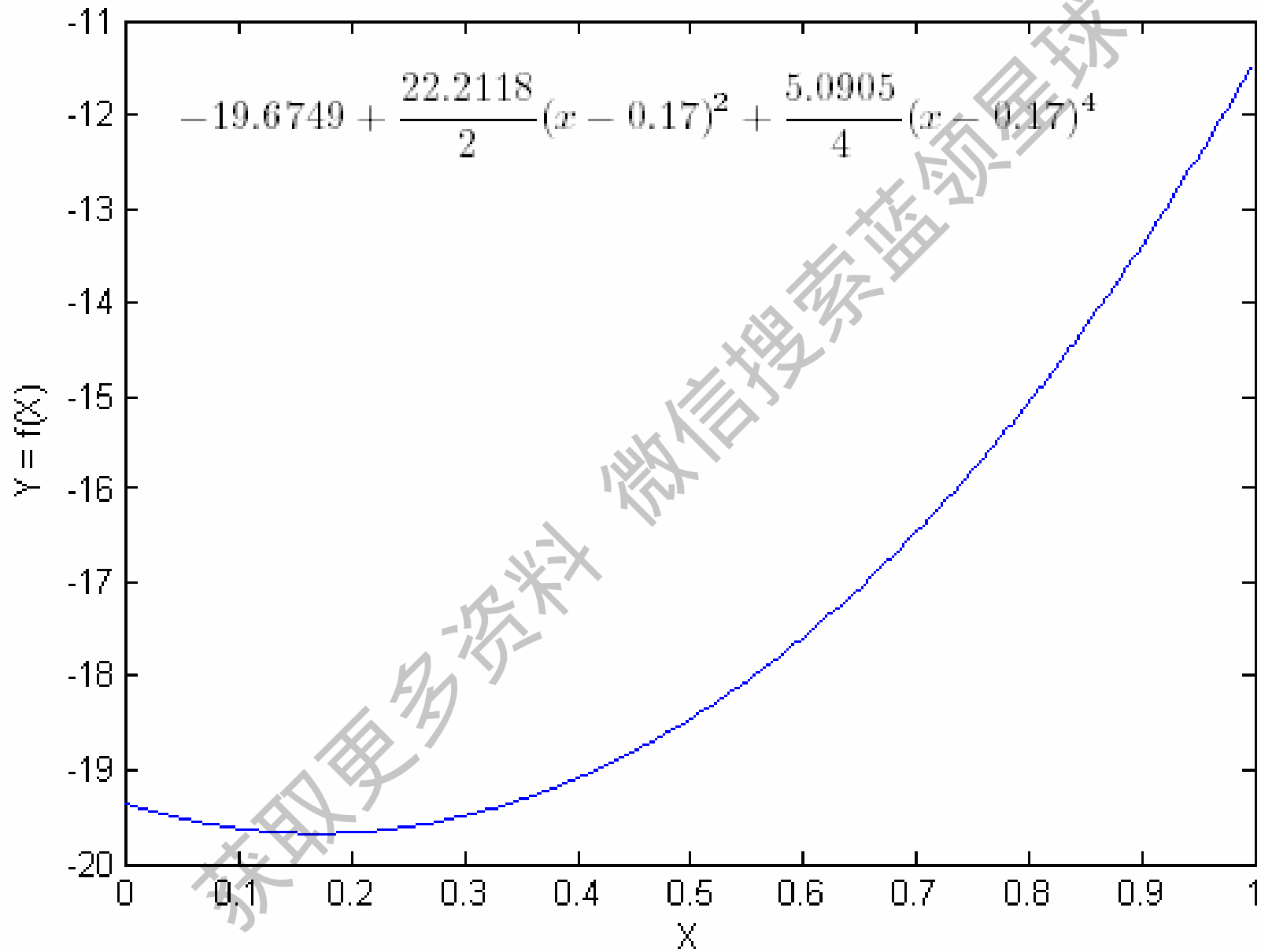
```
>> set(h, 'string', '3x^2+2xy+4y^2 = 5', 'fontsize', 15);  
>> h = title('这是一个椭圆曲线', 'fontsize', 18, 'fontweight', 'bold');  
>> set(h, 'position', [-0.00345622 1.35769 1.00011]);  
>> axis([-1.5 1.5 -1.2 1.7]);  
>> xlabel('X');  
>> ylabel('Y');
```



【例5.2-5】 绘制曲线 $-19.6749 + \frac{22.2118}{2}(x-0.17)^2 + \frac{5.0905}{4}(x-0.17)^4$

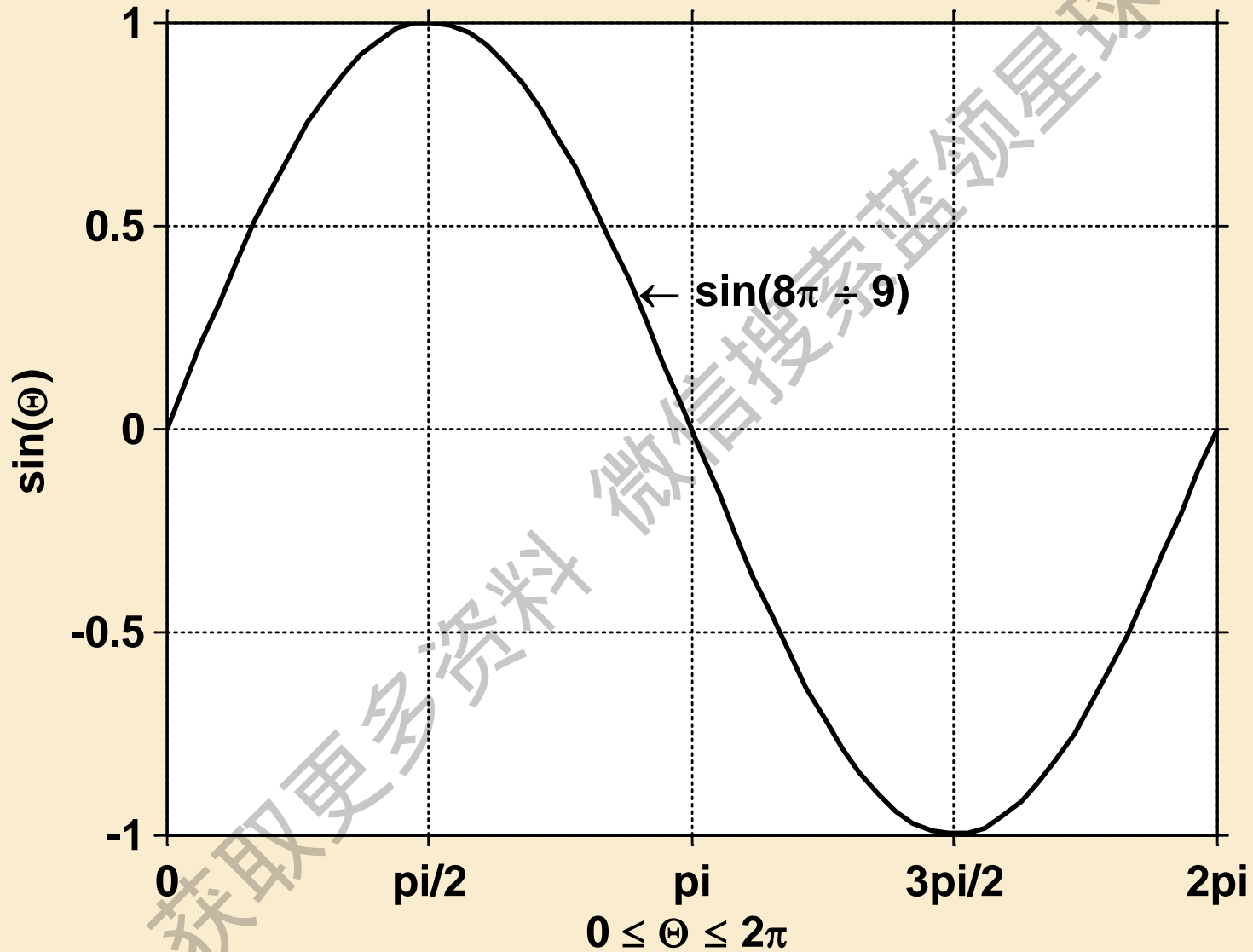
并添加曲线方程。

```
>> 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5.2-6】 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

常用统计绘图函数列表

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5.2-7】 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);
```

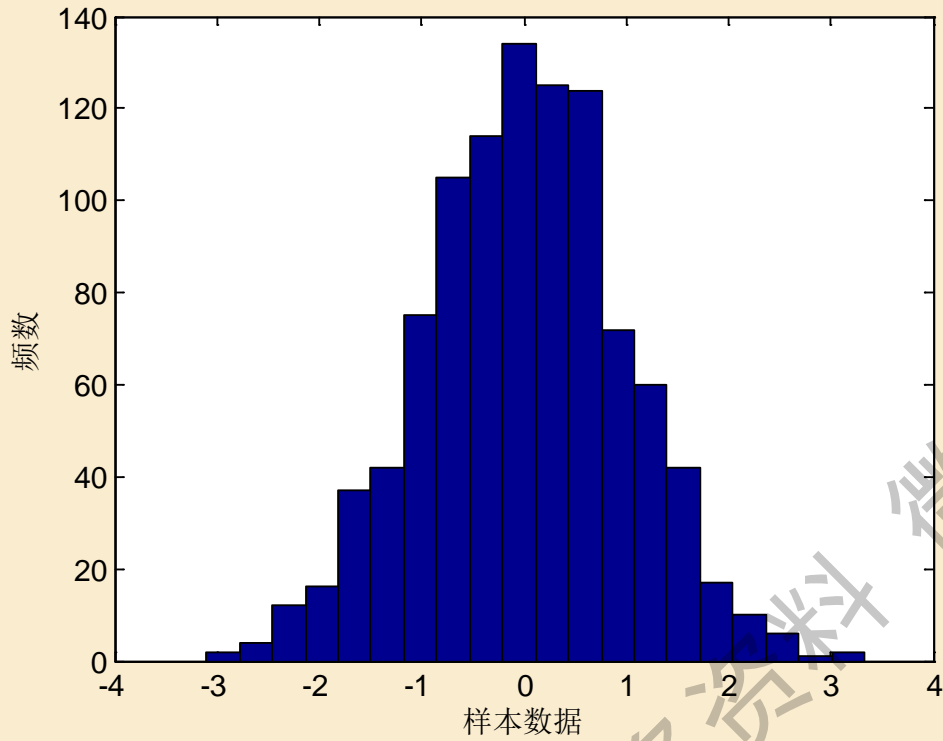
```
>> xlabel('样本数据');
```

```
>> ylabel('频数');
```

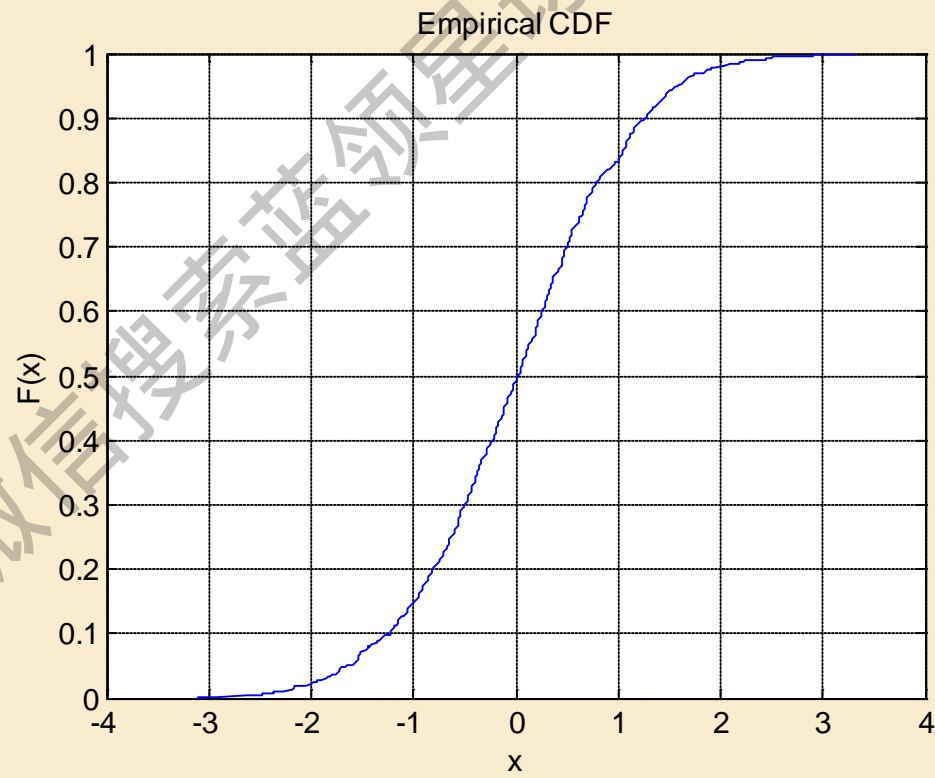
```
>> figure;
```

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

获取更多资料 微信搜索蓝领星球



频数直方图



经验分布函数图

四、特殊二维绘图函数

特殊二维绘图函数列表

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

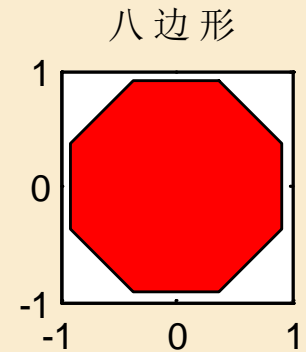
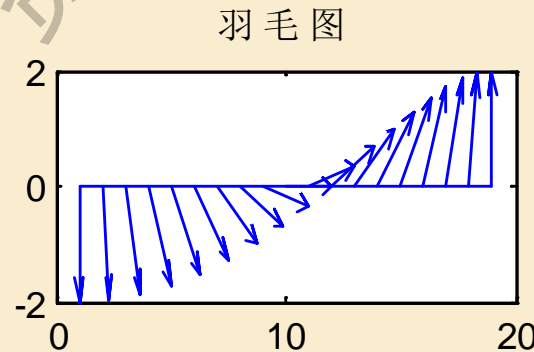
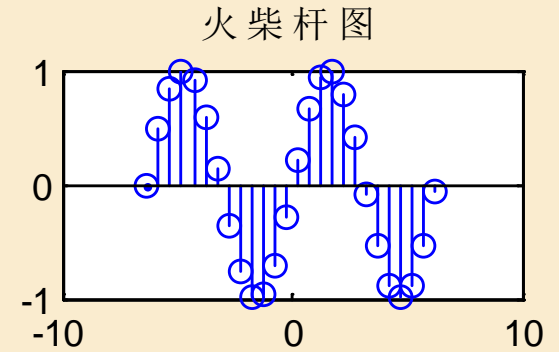
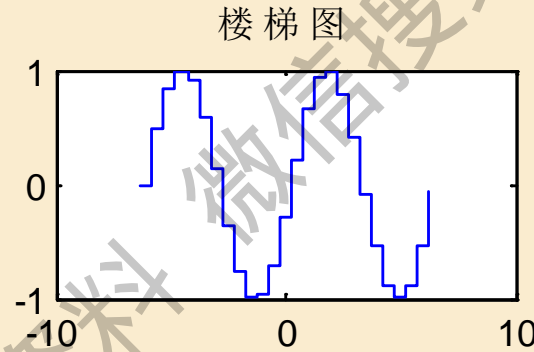
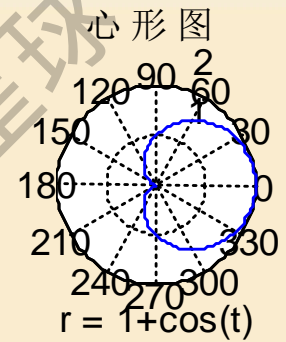
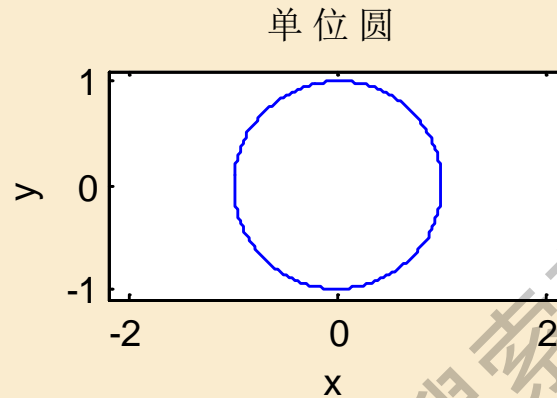
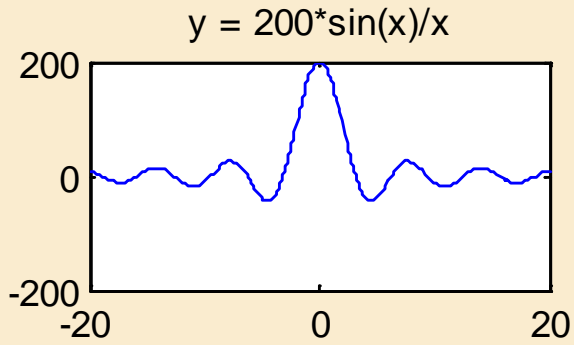
【例5.2-8】 特殊二维图形。

```
>> 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');
```

获取更多资料 微信搜索蓝领星球



第三节 三维图形绘制

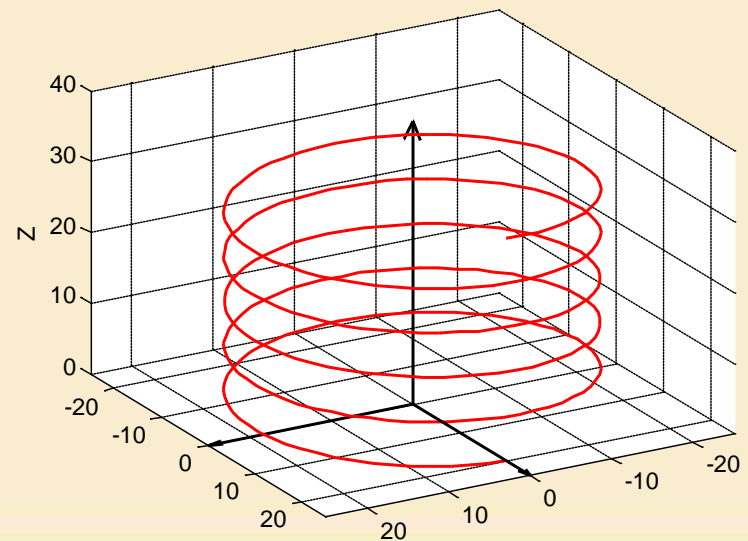
获取更多资料 微信搜索 领星球

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
surf1	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

【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);
```



【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
```

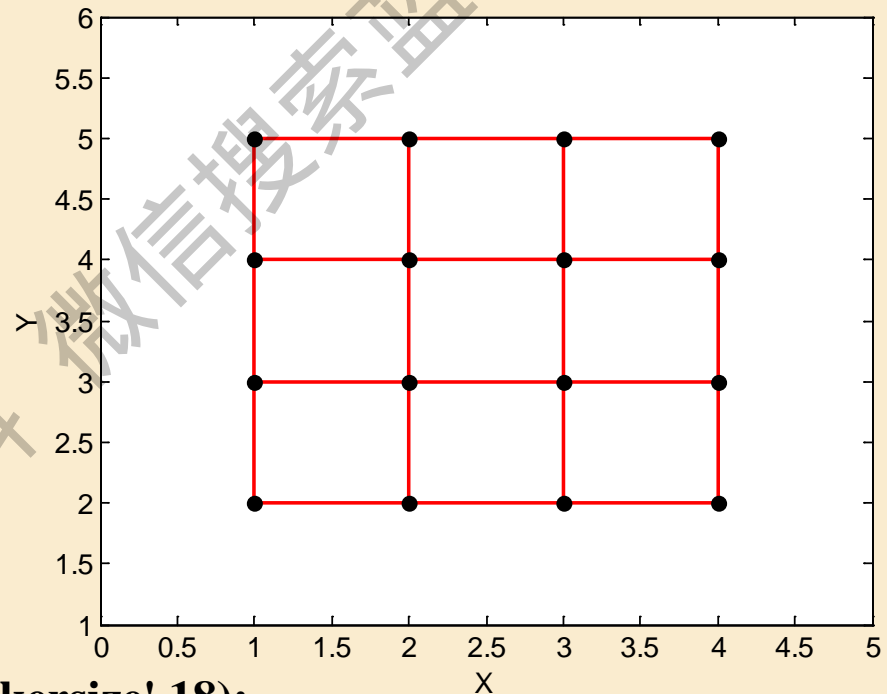
```
 4  4  4  4
```

```
 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');
```

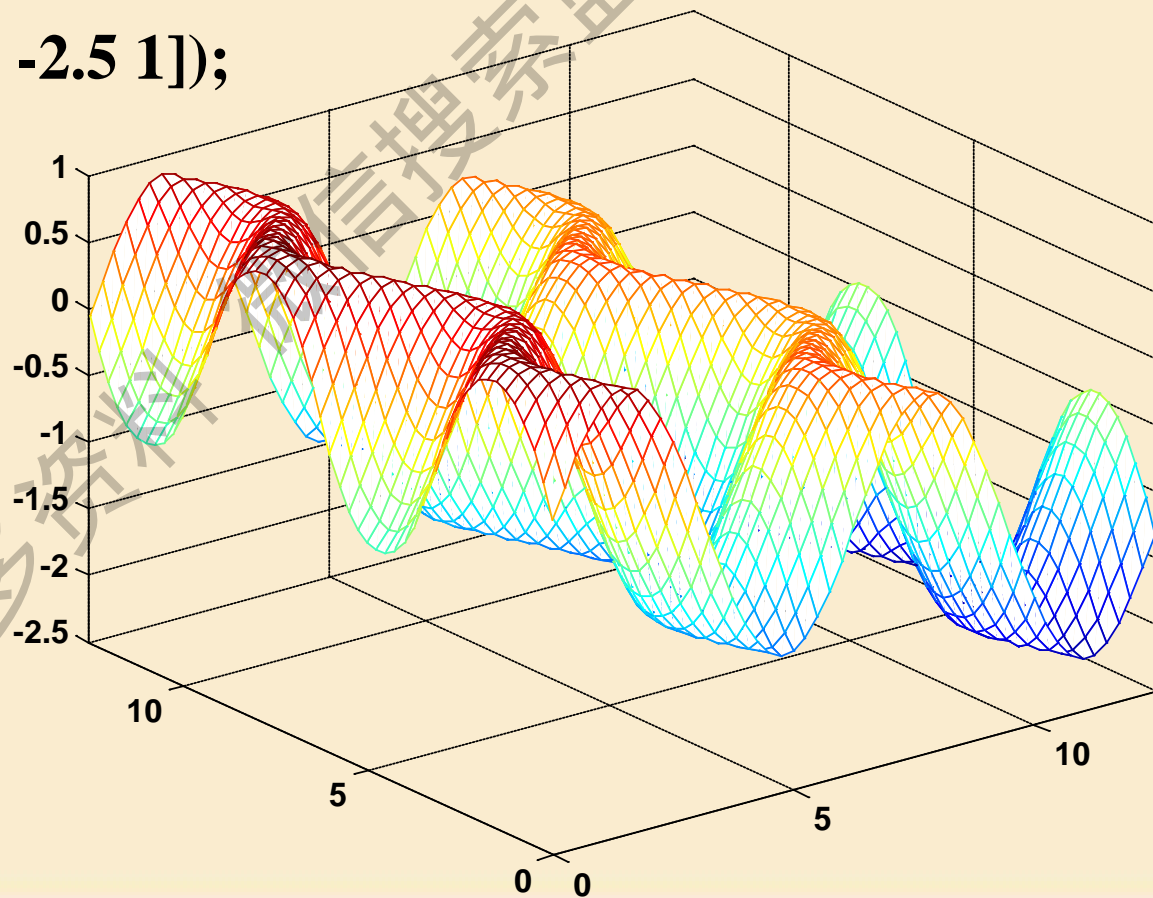


【例5.3-3】 绘制三维曲面图 $z=\sin(x+\sin(y))-x/10$ 。

```
>> [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]);
```

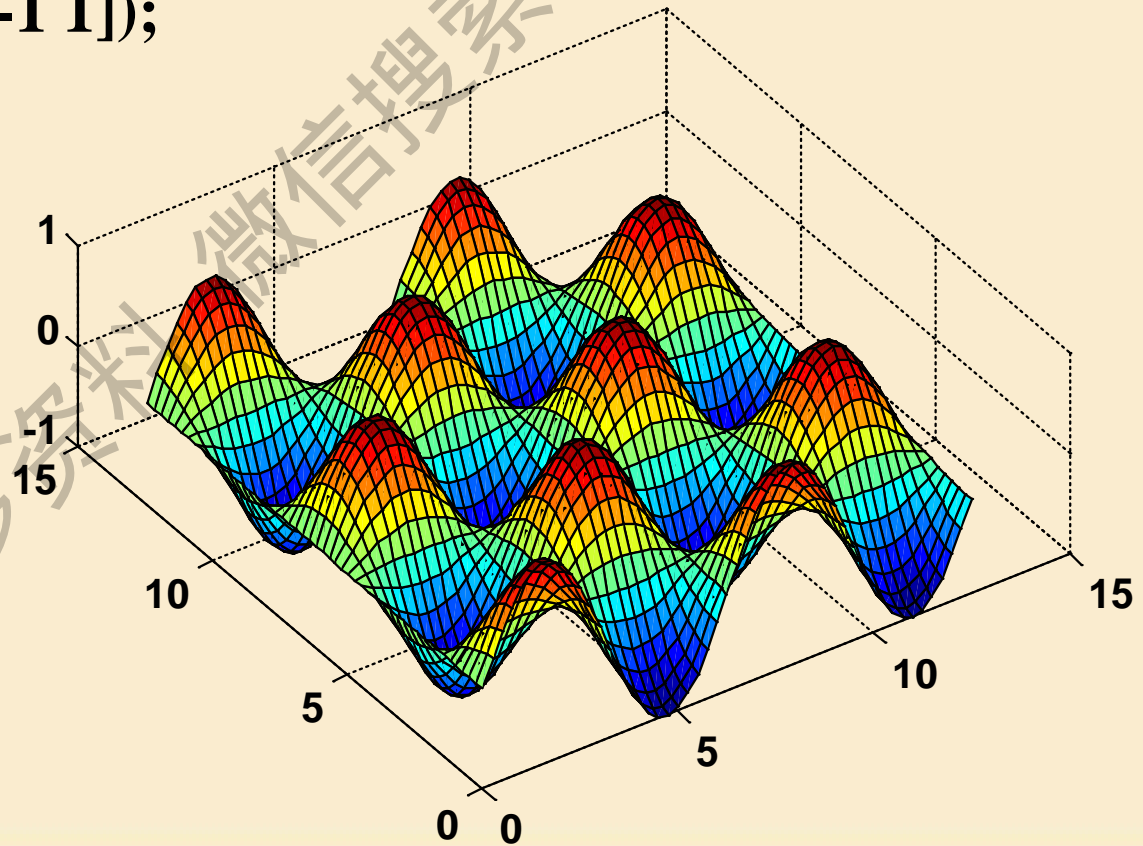


【例5.3-4】 绘制三维曲面图 $z = \sin(x) \cdot \cos(y)$ 。

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

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

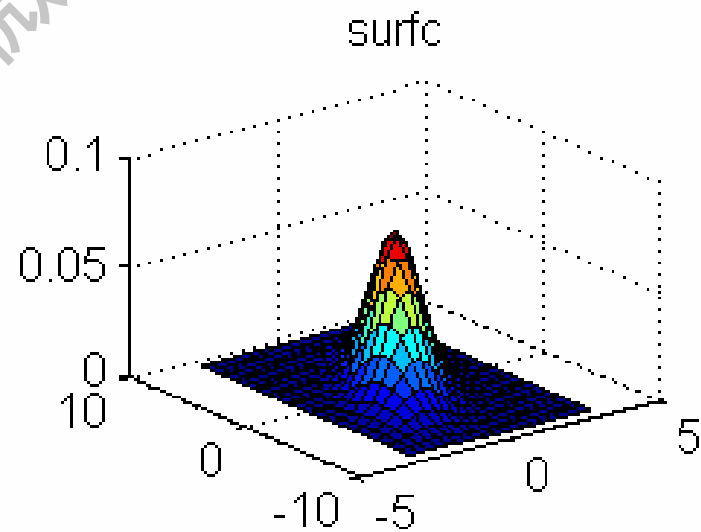
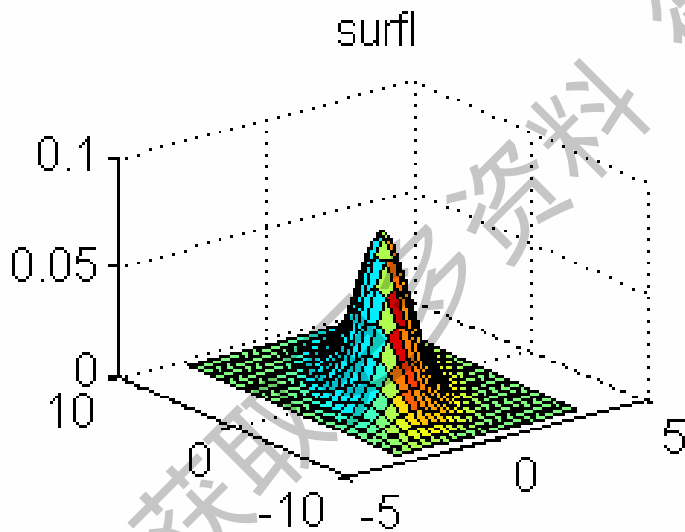
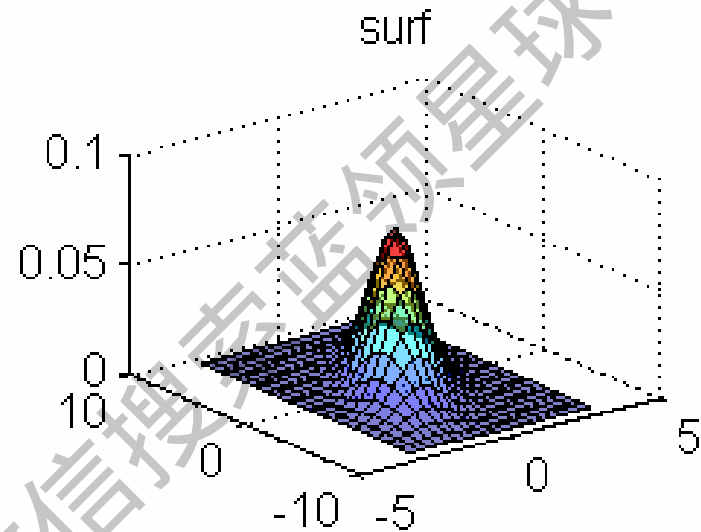
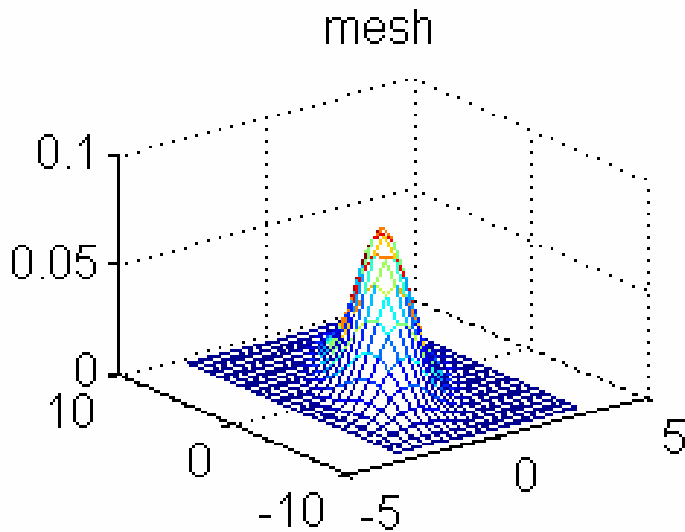
```
>> axis([0 4*pi 0 4*pi -1 1]);
```



【例5.3-5】用mesh, surf, surfl, surfc函数绘制二元正态分布的密度函数图。

```
>> x = linspace(-3,3,20);  
>> y = linspace(-9,9,20);  
>> [X, Y] = meshgrid(x,y);  
>> Z = mvnpdf([X(:) Y(:)], [0 0], [1 2;2 9]);  
>> Z = reshape(Z, size(X));  
>> subplot(2, 2, 1);  
>> mesh(X, Y, Z);  
>> title('mesh');
```

```
>> subplot(2, 2, 2);  
>> surf(X, Y, Z);  
>> alpha(0.5);  
>> title('surf');  
>> subplot(2, 2, 3);  
>> surf(X, Y, Z);  
>> title('surf');  
>> subplot(2, 2, 4);  
>> surf(X, Y, Z);  
>> title('surf');
```

【例5.3-6】调用函数绘制柱面，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);
```

```
>> t = 0:pi/10:2*pi;
```

```
>> [X,Y,Z] = cylinder(2+cos(t));
```

```
>> 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

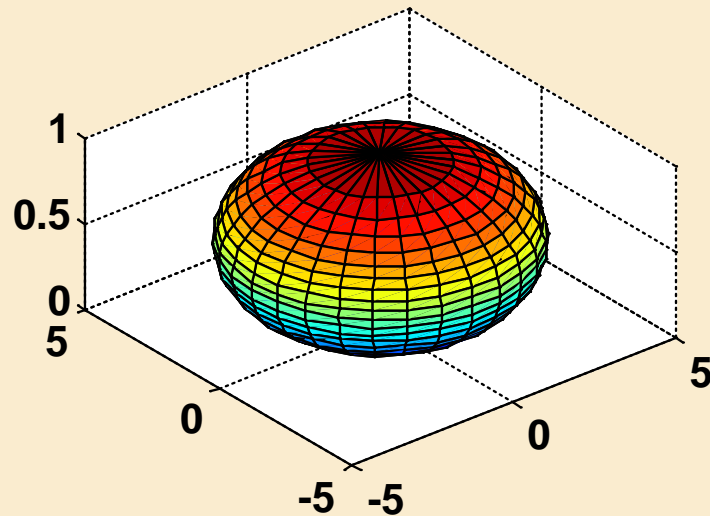
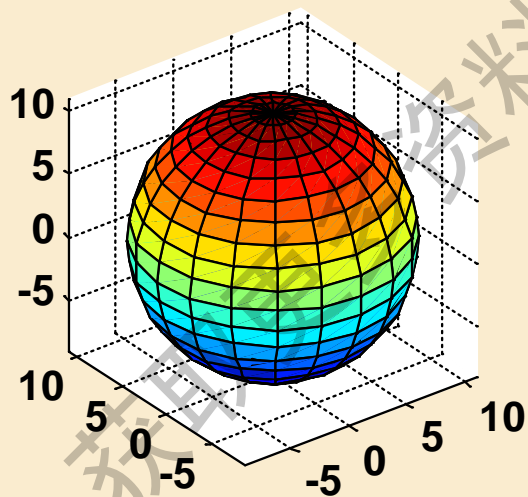
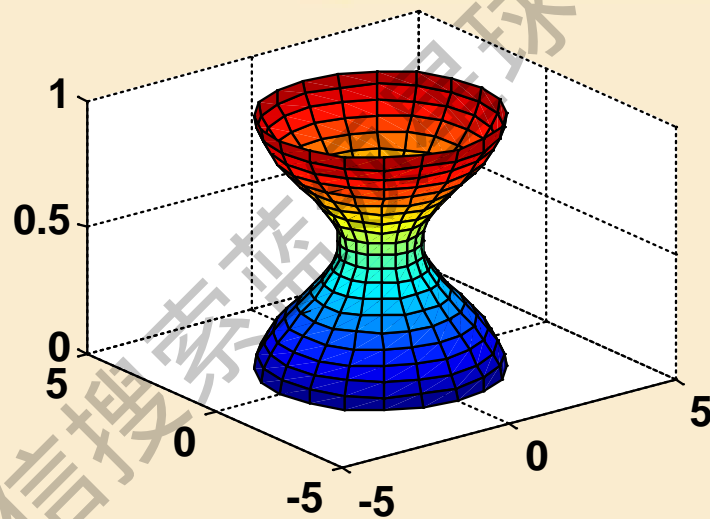
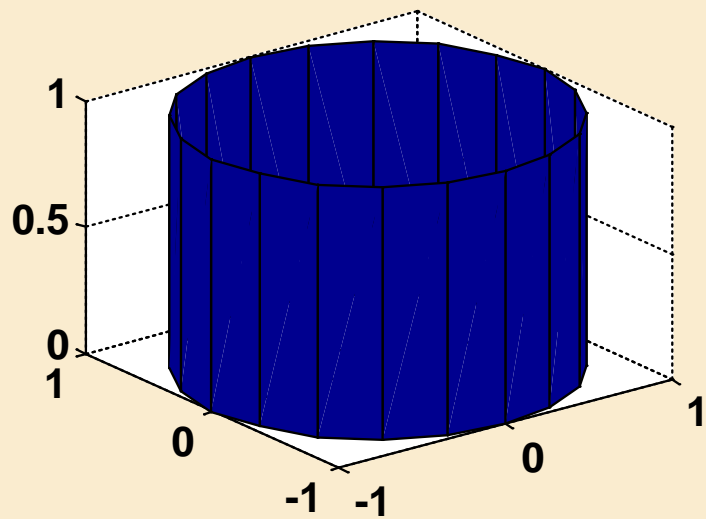
```
>> subplot(2,2,4);
```

```
>> 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');
```

获取更多资料 微信搜索 蓝领星球

% 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]);
```

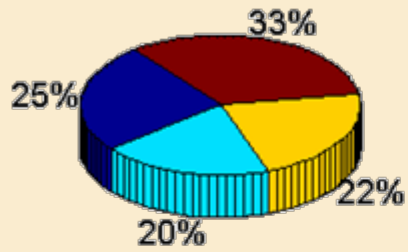
获取更多资料 微信搜索蓝领星球

% 立体切片图（四维图）

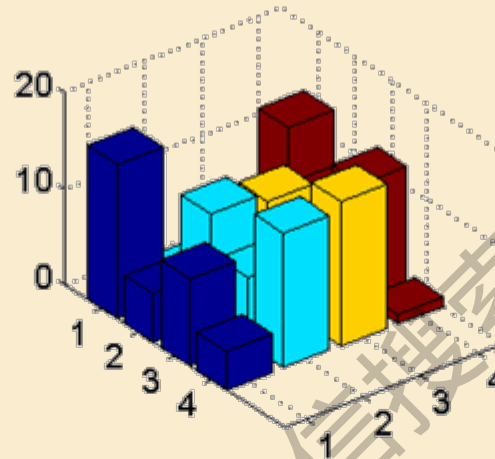
```
>> subplot(2,3,6);  
>> t = linspace(-2,2,20);  
>> [X,Y,Z] = meshgrid(t,t,t);  
>> V = X.*exp(-X.^2-Y.^2-Z.^2);  
>> xslice = [-1.2,.8,2];  
>> yslice = 2;  
>> zslice = [-2,0];  
>> slice(X,Y,Z,V,xslice,yslice,zslice);  
>> title('立体切片图（四维图）');
```

获取更多资料 微信搜索 蓝领星球

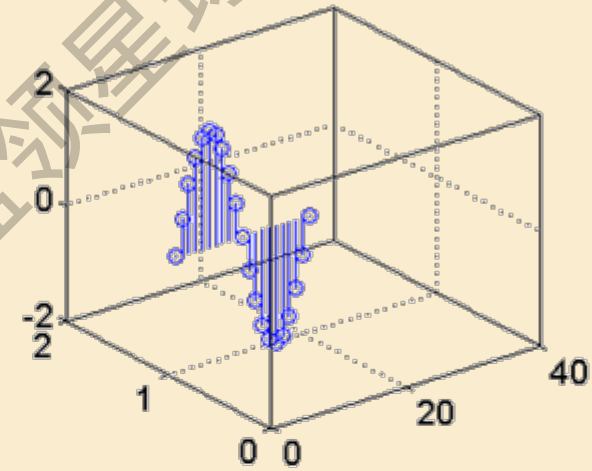
三维饼图



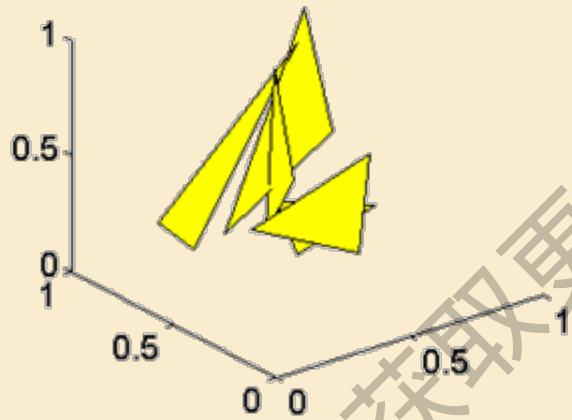
三维柱状图



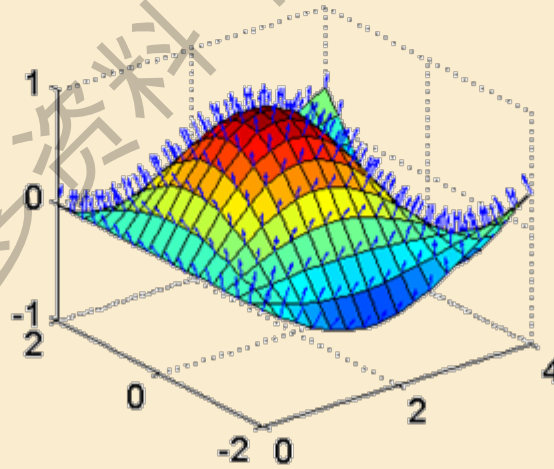
三维火柴杆图



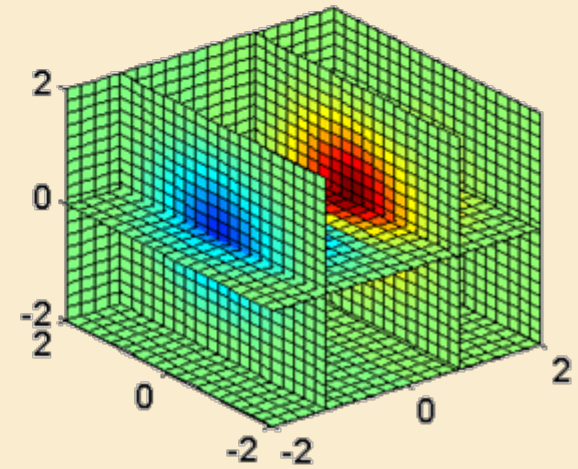
三维填充图



三维向量场图

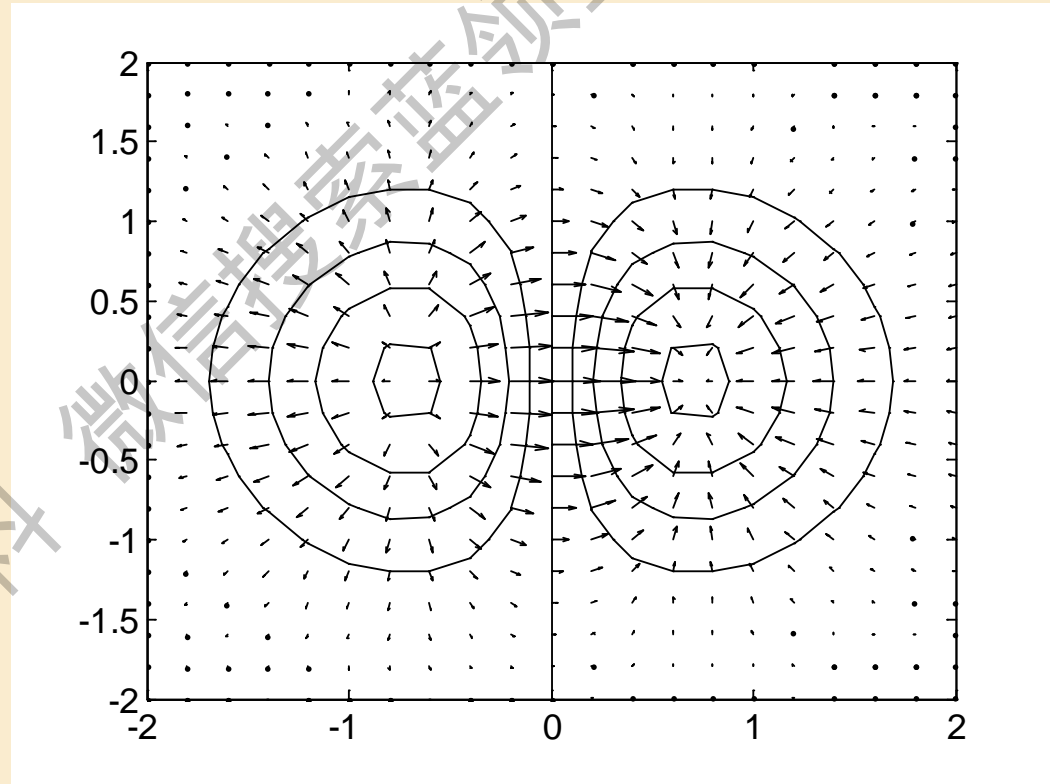


立体切片图（四维图）



【例5.3-8】 绘制三维曲面 $z = xe^{-(x^2+y^2)}$ 的等高线图和梯度场。

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

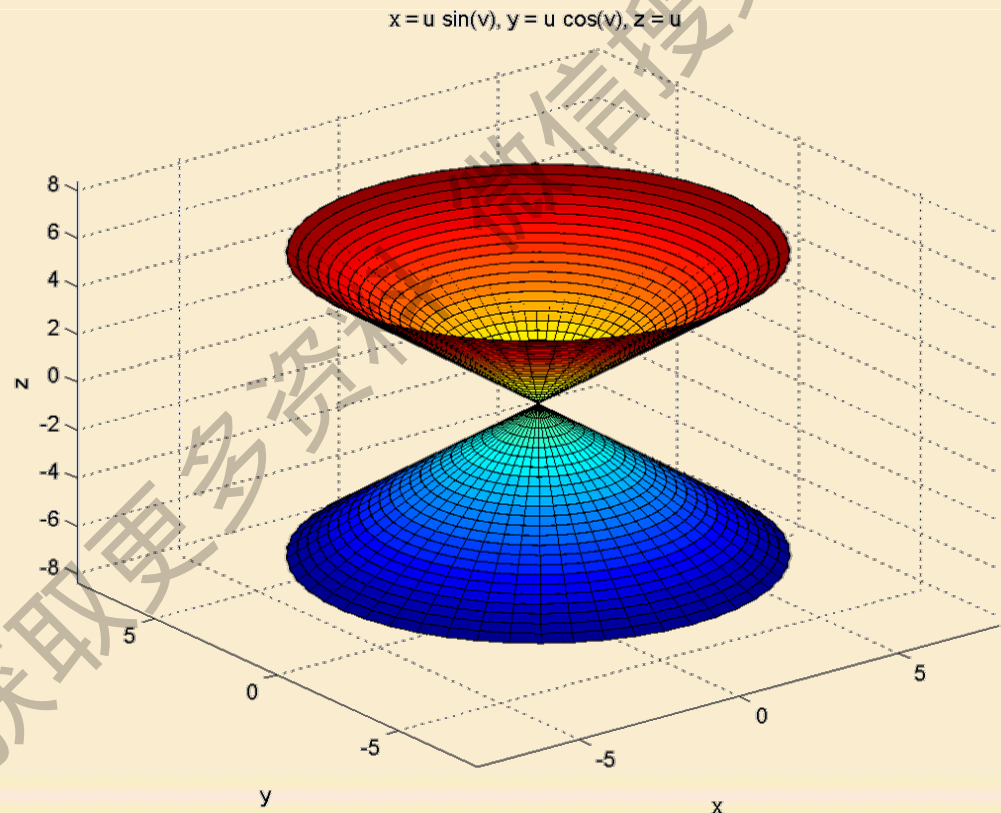


获取更多资料

【例5.3-9】调用ezsurf函数绘制圆锥面

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

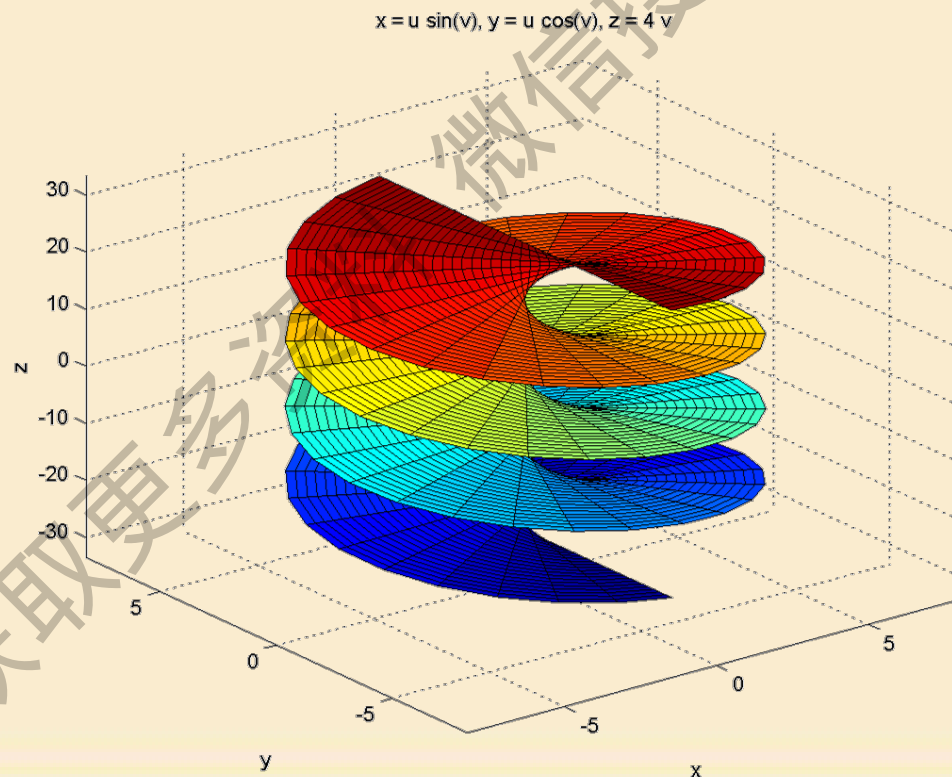
```
>> ezsurf('u*sin(v)', 'u*cos(v)', 'u', [-2*pi, 2*pi, -2*pi, 2*pi])
```



【例5.3-10】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])
```



二、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的设置有两种，可以人为指定一个元素值均在 $[0, 1]$ 之间的的矩阵，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

`colormap` 函数

- `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

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

- 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

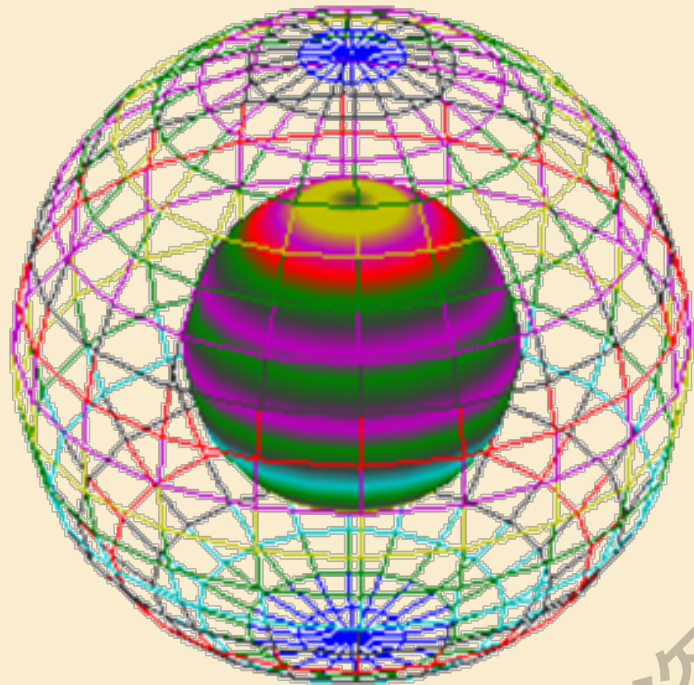
- 通过图形对象的“FaceAlpha”属性调整透明度

调用：**set(h,'FaceAlpha',0.25);**

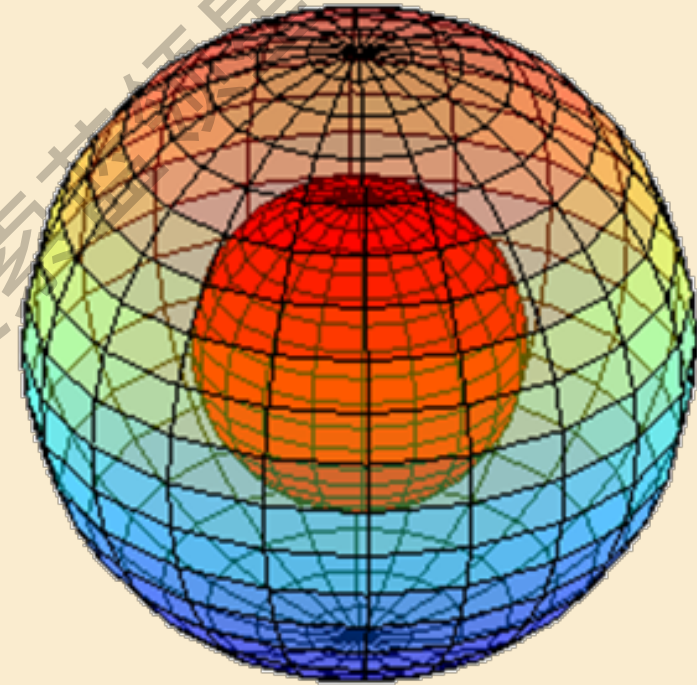
说明：**h**为图形对象句柄，**0.25**是“FaceAlpha”属性的属性值，由用户随意指定。

【例5.3-11】 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）'Position'是位置属性，设置光源位置，其属性值为三个元素的向量[x, y, z]，即光源的三维坐标；'Color'是颜色属性，设置光源颜色，其属性值可以是代表颜色的字符（如表5.2-1所列），也可以是由红、绿、蓝三元色的灰度值组成的向量；'Style'是光源类型属性，设置光源类型，其取值为字符串'infinite'或'local'，分别表示平行光源和点光源。

➤ **handle = light(...)**

建立一个光源，并获取其句柄值**handle**，之后可以通过**get(handle)**查看光源的所有属性，也可以通过**set(handle, 'PropertyName',propertyvalue,...)**设置光源的属性值。

获取更多资料 微信搜索 谢中华

5. 调整光照模式

lighting函数

- **lighting flat**, 产生均匀光照, 选择此方法, 以查看面对象, 是光照模式的默认设置。
- **lighting gouraud**, 计算顶点法线并作线性插值修改表面颜色, 选择此方法, 以查看曲面对象。
- **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, 选择此方法, 以查看曲面对象。此方法比 **lighting gouraud** 的效果好, 但是用于渲染的时间较长。
- **lighting none**, 关掉照明。

6. 图形表面对光照反射属性设置

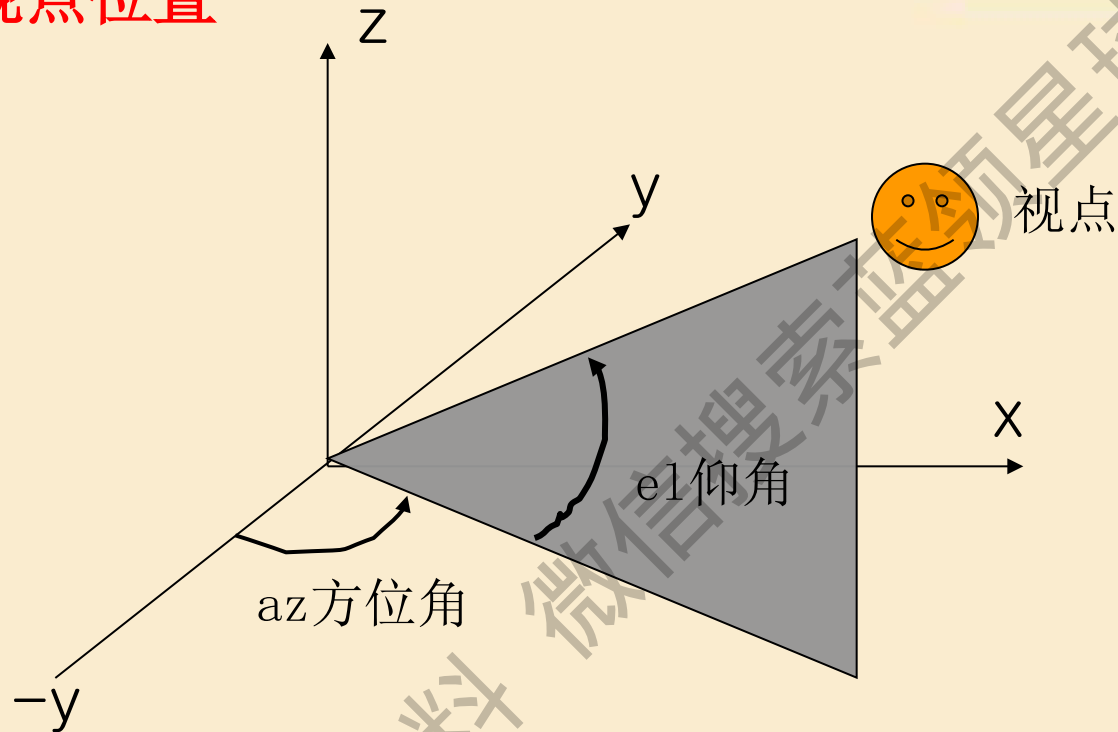
material函数

- **material shiny**, 使图形对象有相对较高的镜面反射, 镜面光的颜色仅取决于光源颜色。
- **material dull**, 使图形对象有更多的漫反射, 反射光的颜色仅取决于光源颜色。
- **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, 反射光的颜色取决于光源颜色和图形表面的颜色。

- `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。ka、kd、ks、n和sc均为标量, sc的取值介于0和1之间。
- `material default`, 恢复ka、kd、ks、n和sc的默认值。

获取更多资料

7. 调整视点位置



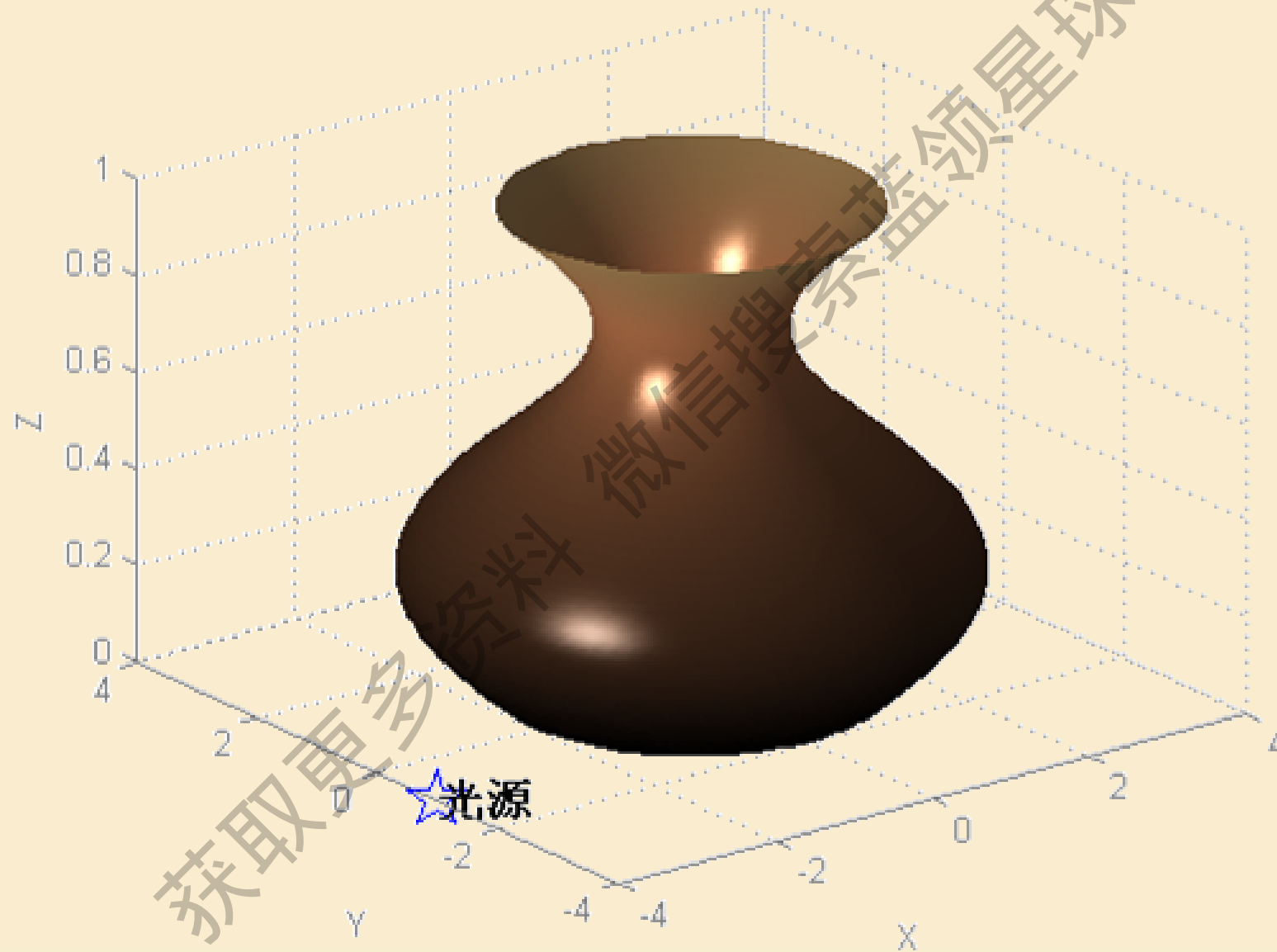
view函数

- **view(az,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)`, 设置默认的二维视角, $az = 0$, $el = 90$.
- `view(3)`, 设置默认的三维视角, $az = -37.5$, $el = 30$.
- `view(ax,...)`, 设置句柄值为`ax`的坐标系的视角。
- `view(T)`, 用`viewmtx`函数生成的 4×4 的视角转换矩阵作为输入来设置视角。
- `[az,el] = view`, 返回当前方向角和仰角。
- `T = view`, 返回一个 4×4 的视角转换矩阵。

【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');
```

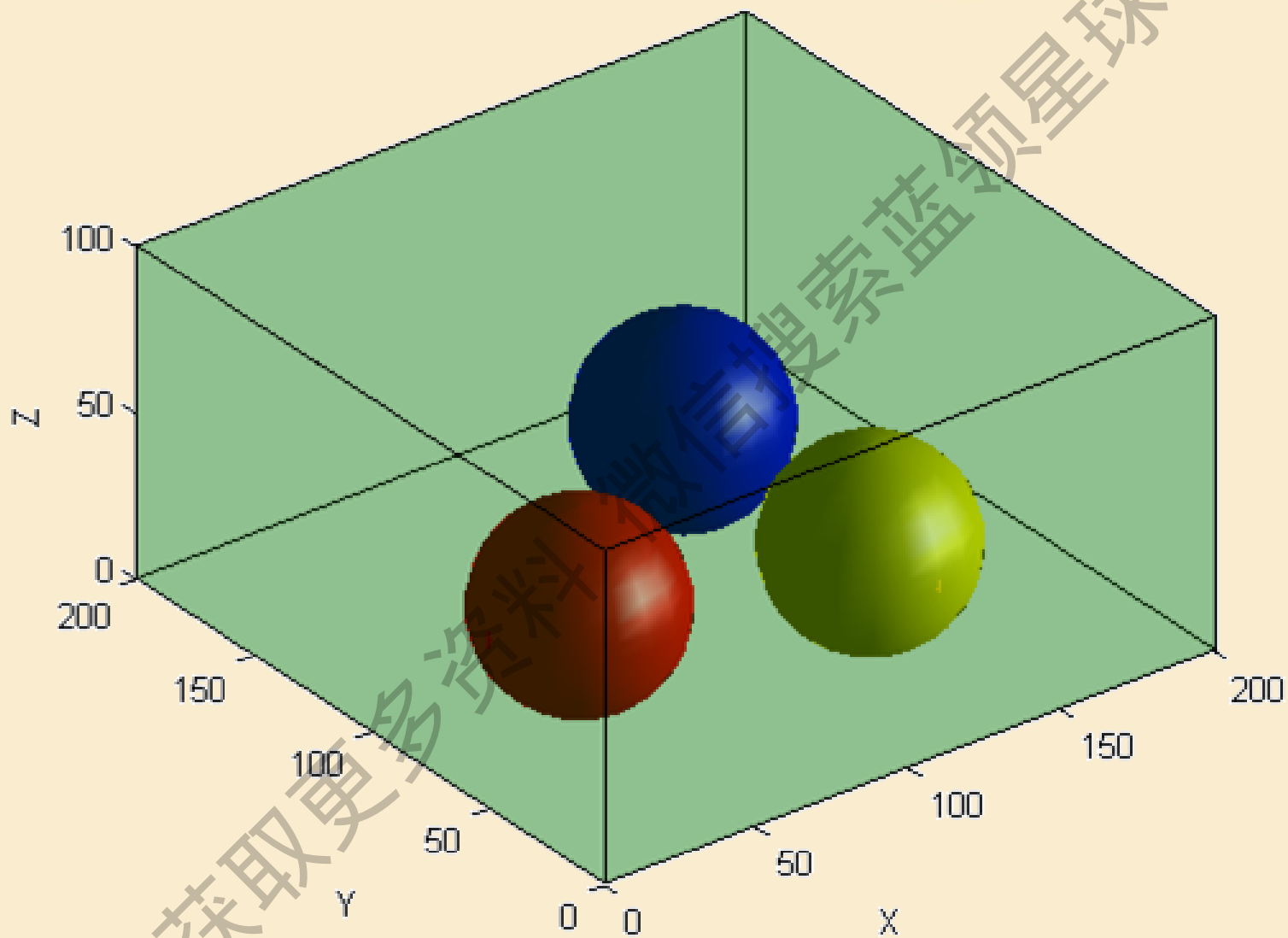


【例5.3-13】 绘制一个透明的立方体盒子，里面放红色、蓝色和黄色三个球。

```
>> vert = [0 0 0;0 200 0;200 200 0;200 0 0;0 0 100;0 200 100;200 200 100;200 0 100];  
>> fac = [1 2 3 4;2 6 7 3;4 3 7 8;1 5 8 4;1 2 6 5;5 6 7 8];  
>> view(3);  
>> h = patch('faces',fac,'vertices',vert,'FaceColor','g');  
>> set(h,'FaceAlpha',0.25);  
>> hold on;  
>> [x0,y0,z0] = sphere;  
>> x = 30 + 30*x0; y = 50 + 30*y0; z = 50 + 30*z0;
```

```
>> h1 = surf(x,y,z,'linestyle','none','FaceColor','r','EdgeColor','none');  
>> x = 110 + 30*x0; y = 110 + 30*y0; z = 50 + 30*z0;  
>> h2 = surf(x,y,z,'linestyle','none','FaceColor','b','EdgeColor','none');  
>> x = 110 + 30*x0; y = 30 + 30*y0; z = 50 + 30*z0;  
>> h3 = surf(x,y,z,'linestyle','none','FaceColor','y','EdgeColor','none');  
>> lightangle(45,30);  
>> lighting phong;  
>> axis equal;  
>> xlabel('X'); ylabel('Y'); zlabel('Z');
```

获取更多资料
微信搜索蓝领星球

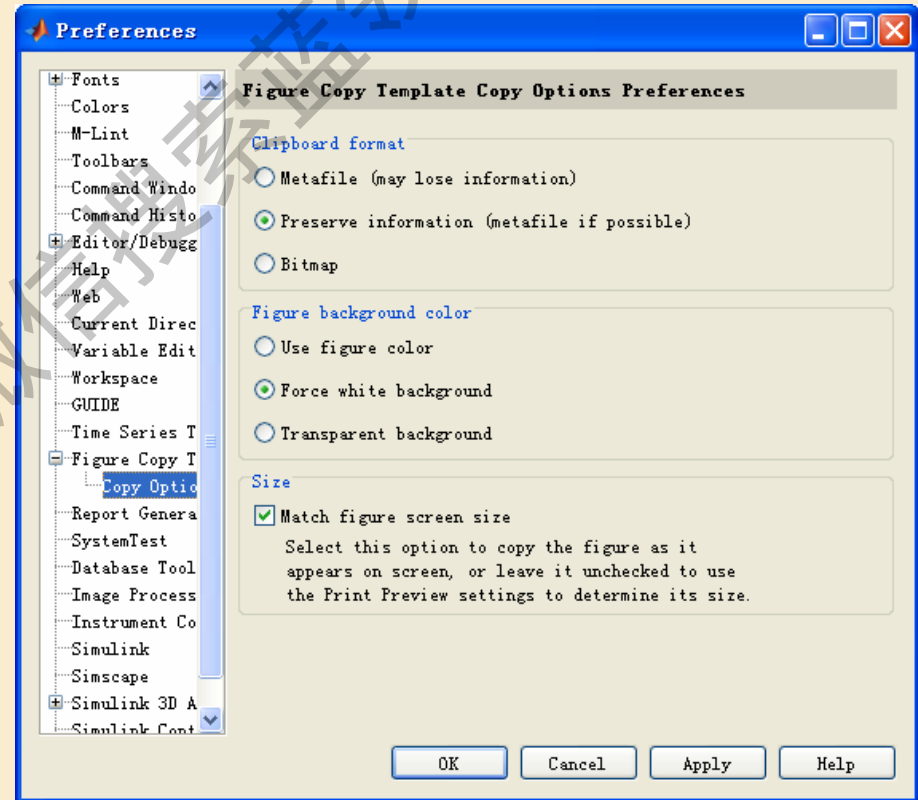
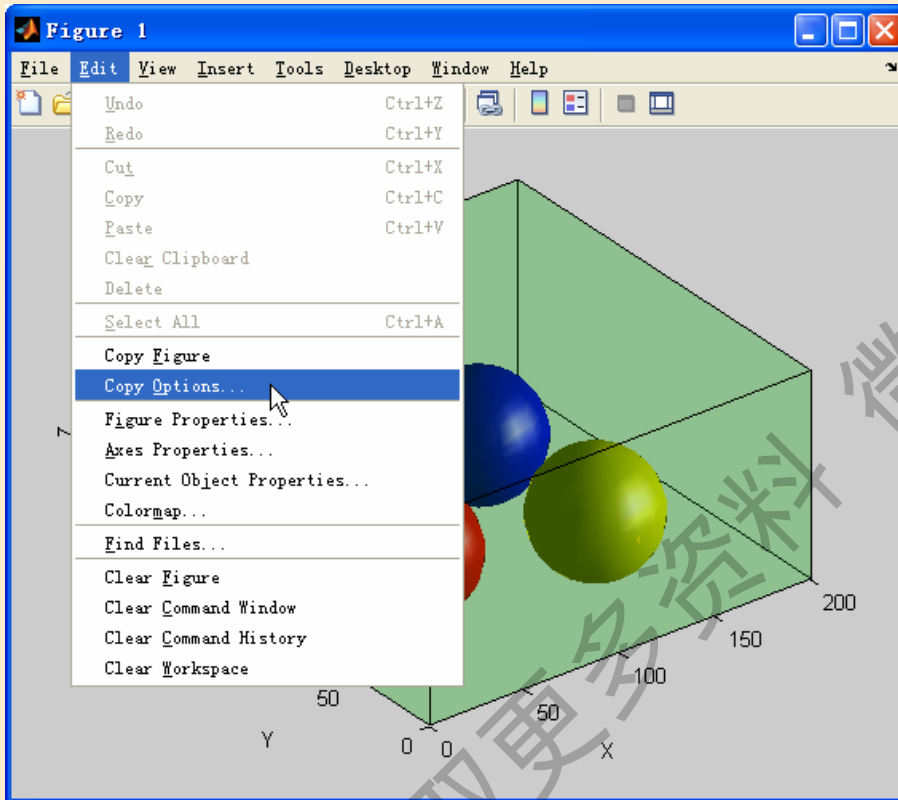


第四节 图形的打印和输出

获取更多资料 微信搜索 精英星球

一、把图形复制到剪贴板

1. 界面操作



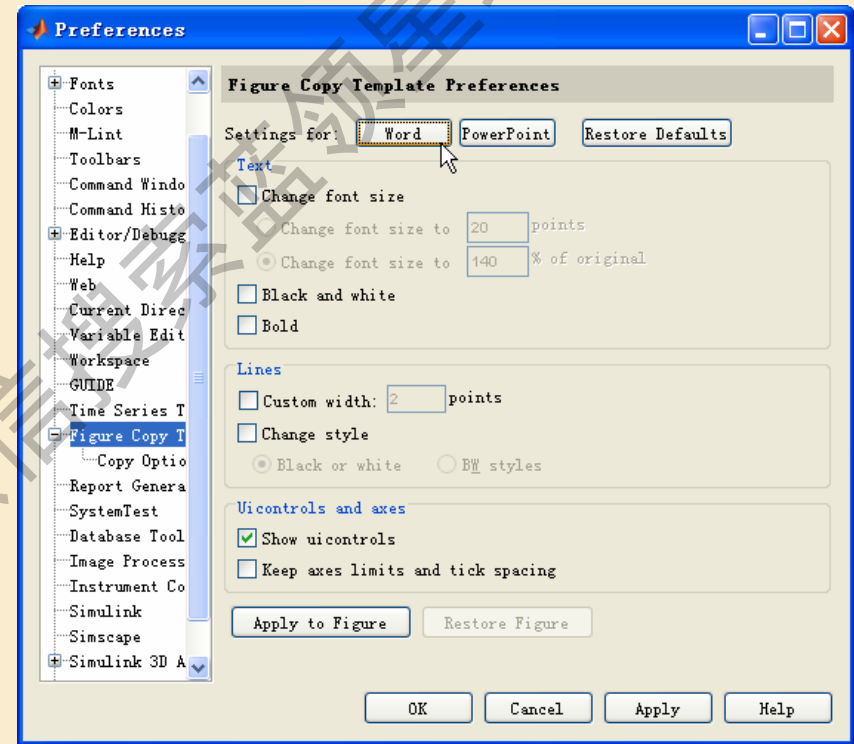
➤ 复制选项设置

实际上在将图形窗口中的图形复制到剪贴板之前，还可以通过界面操作对复制选项进行设置。选择“**Edit**”菜单下的“**Copy Options**”选项，弹出复制选项界面，本界面用来设置图形复制到剪贴板的格式（**Clipboard format**）、背景颜色（**Figure background color**）和图形尺寸（**Size**）。

获取更多资料

➤ 复制模板设置

在复制选项界面的左方浏览树中选中**Figure Copy Template**节点，将弹出复制模板界面。复制模板界面中给出了3个模板选项：**Word**、**PowerPoint**和**Restore Defaults**，当需要把MATLAB图形窗口中的图形复制粘贴到Microsoft Word和Microsoft PowerPoint应用程序时，分别可以用前两个模板，最后一个选项用来恢复默认设置。



2. 利用MATLAB命令进行复制操作

➤ print函数

调用格式:

print

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

print(handle,'filename')

print argument1 argument2 ... argumentn

[pcmd,dev] = printopt

【例5.4-1】调用print函数复制图形到剪贴板。

```
>> h = figure;
```

```
>> ezplot('sin(x)', [0, 2*pi]);
```

```
>> print(h, '-dmeta')
```

获取更多资料 微信搜索蓝领星球

➤ **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。**filename**为字符串，用来指明文件名和保存路径，如果不指明保存路径，图形默认保存到MATLAB当前文件夹。

➤ **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续】调用hgexport函数复制图形到剪贴板。

```
>> h = figure;
```

```
>> ezplot('sin(x)', [0, 2*pi]);
```

```
>> hgexport(h, '-clipboard');
```

获取更多资料 微信搜索 蓝领星球

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。

获取更多资料

2. 利用MATLAB命令把图形导出到文件

➤ print函数

调用格式:

print

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

print(handle,'filename')

print argument1 argument2 ... argumentn

[pcmd,dev] = printopt

【例5.4-2】调用print函数保存图形。

```
>> h = figure;
```

```
>> ezplot('sin(x) ',[0,2*pi]);
```

```
>> print(h, '-djpeg', '示例1.jpg')
```

获取更多资料 微信搜索蓝领星球

➤ **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。**filename**为字符串，用来指明文件名和保存路径，如果不指明保存路径，图形默认保存到MATLAB当前文件夹。

➤ **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5.4-2续】调用hgexport函数保存图形。

```
>> h = figure;
```

```
>> ezplot('sin(x)', [0, 2*pi]);
```

```
>> hgexport(h, '示例.eps')
```

获取更多资料 微信搜索 蓝领星球

➤ saveas函数

调用格式:

➤ **saveas(h,'filename.ext')**

把句柄值为h的图形或句柄值为h的Simulink模块图保存为文件filename.ext。文件格式取决于文件的扩展名ext。

➤ **saveas(h,'filename','format')**

把句柄值为h的图形或句柄值为h的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续】调用saveas函数保存图形。

```
>> h = figure;
```

```
>> ezplot('sin(x)', [0, 2*pi]);
```

```
>> saveas(h, '示例2.jpg')
```

获取更多资料 微信搜索蓝领星球

三、打印图形

1. 界面操作

在MATLAB中绘制好图形之后，单击图形窗口File菜单下的Print Preview选项，打开打印预览界面，本界面用来设置打印相关属性。设置好打印属性之后，单击界面左上方的Save As按钮可将用户设置保存下来以备以后使用。单击界面右上方的Print按钮弹出打印机属性设置界面，在打印机属性设置界面中选择合适的打印机，设置好打印机属性和打印份数之后单击“确定”按钮即完成打印。如果在打印机属性设置界面中勾选了“打印到文件”选项，则可将图形导出到文件。

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;
```

```
>> print -dmeta
```

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

获取更多资料 微信搜索蓝领星球

【注】当MATLAB中同时打开多个图形窗口（Figure对象）时，还可利用如下命令对第 i 个图形窗口中的图形进行打印操作：

```
>> print -dmeta -fi
```

```
>> print -djpeg -fi filename
```

```
>> print -fi
```

获取更多资料 微信搜索蓝领精英



第五节 动画制作

获取更多资料 微信搜索 蔡志领星球

一、质点运动轨迹动画

1. comet函数

调用格式:

➤ **comet(y)**

显示质点绕着向量y的二维动画轨迹

➤ **comet(x,y)**

显示质点绕着向量y与x的二维动画轨迹

➤ **comet(x,y,p)**

用输入参数p定义轨迹尾巴线的长度

➤ **comet(axes_handle,...)**

在句柄值为axes_handle的坐标系中显示动画

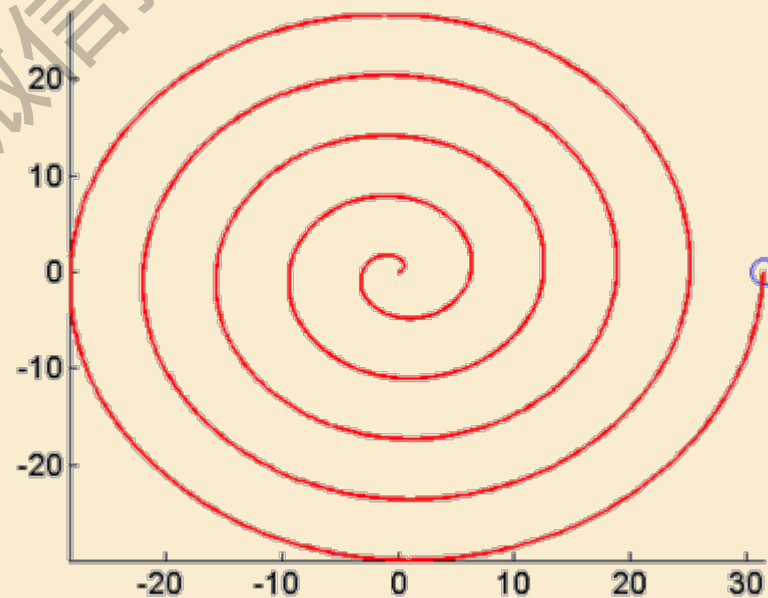
【例5.5-1】质点绕阿基米德螺线（极坐标方程为 $\rho = \theta$ ）运动的二维彗星运行轨迹动画。

```
>> t = linspace(0,10*pi,2000);
```

```
>> x = t.*cos(t);
```

```
>> y = t.*sin(t);
```

```
>> comet(x,y);
```



2. comet3函数

调用格式:

➤ **comet3(z)**

显示质点绕着向量 z 的三维动画轨迹

➤ **comet3(x,y,z)**

显示质点绕着向量 x 、 y 与 z 的三维动画轨迹

➤ **comet(x,y,z,p)**

用输入参数 p 定义轨迹尾巴线的长度

➤ **comet(axes_handle,...)**

在句柄值为 $axes_handle$ 的坐标系中显示动画

【例5.5-2】质点绕螺旋线
轨迹动画。

$$\begin{cases} x = 20 \sin t \\ y = 20 \cos t \\ z = t \end{cases} \quad \text{运动的三维彗星运行}$$

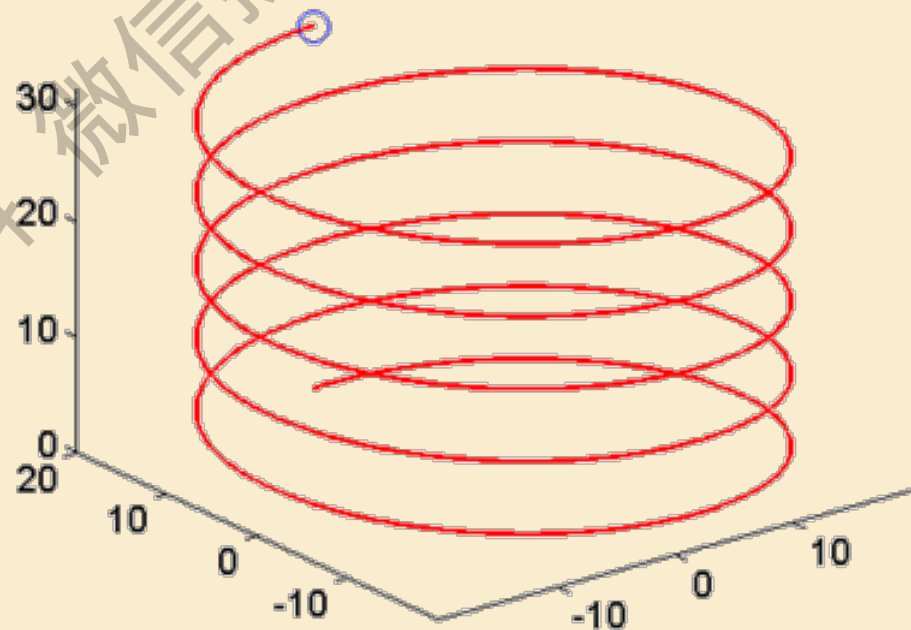
```
>> t = linspace(0, 10*pi, 1000);
```

```
>> x = 20*sin(t);
```

```
>> y = 20*cos(t);
```

```
>> z = t;
```

```
>> comet3(x,y,z)
```



二、霓虹闪烁动画

如今繁华大都市的夜色中，霓虹闪烁，煞是好看，调用MATLAB中的spinmap函数可以做出这种效果的动画，它是通过旋转颜色映像的方式来呈现这种霓虹闪烁的动画效果。

获取更多资料 微信搜公众号 星球

1. spinmap函数

调用格式:

- `spinmap`, 旋转颜色映像约5秒钟;
- `spinmap(t)`, 旋转颜色映像约t秒钟, 具体时间取决于硬件;
- `spinmap(t,inc)`, 旋转颜色映像约t秒钟, 并设置增量参数inc, 该参数用来调整闪烁频率;
- `spinmap('inf')`, 不限时旋转颜色映像, 若需终止, 请按Ctrl+C键。

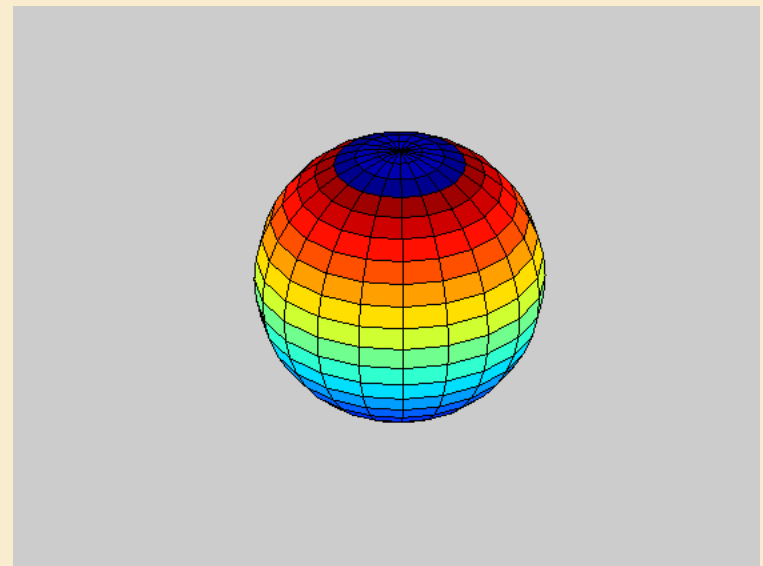
【例5.5-3】霓虹闪烁的球体。

```
>> sphere;
```

```
>> axis equal;
```

```
>> axis off;
```

```
>> spinmap(20,1);
```



三、电影动画

所谓的电影动画是指先把一帧帧图片保存起来，然后再像放电影一样把它们按次序播放出来。MATLAB中提供了**getframe**函数和**movie**函数，用来制作电影动画。其中**getframe**函数用来抓取图形对象作为电影的帧，**movie**函数用来播放电影动画。

获取更多资料 微信: 谢中华

1. getframe函数

调用格式:

➤ **getframe**

返回一个电影帧，它是当前坐标系或图形窗口的快照

➤ **F = getframe**

抓取当前坐标系作为一帧

➤ **F = getframe(h)**

抓取句柄值为h的图形对象作为一个电影帧

➤ **F = getframe(h,rect)**

抓取句柄值为h的图形对象的指定区域作为一个电影帧，rect是形如[左边距，下边距，宽度，高度]的向量，用来设定抓取区域

2. movie函数

调用格式:

➤ **movie(M)**

在当前坐标系中只播放一次由矩阵M所保存的电影

➤ **movie(M,n)**

播放n次，若n为负数，则倒着循环播放，若n为向量，第一个元素为播放次数，后续元素为帧序号

➤ **movie(M,n,fps)**

每秒播放fps帧，播放n次，默认每秒12帧

➤ **movie(h,...)**

在句柄值为h的图形窗口或坐标系中播放电影

➤ **movie(h,M,n,fps,loc)**

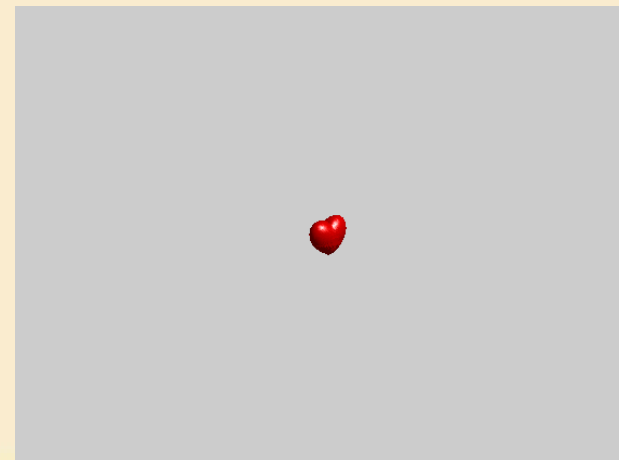
在句柄值为h的图形窗口或坐标系的指定位置播放电影。

M,n,fps的说明同上。loc是4个元素的向量[x y 0 0]，x和y用来设定帧的左下角在图形窗口或坐标系的位置，单位为像素

获取更多资料

【例5.5-4】 一颗跳动的红心。

```
>> x = linspace(-2,2,100);  
>> [X,Y,Z] = meshgrid(x,x,x);  
>> V = (X.^2+9/4*Y.^2+Z.^2-1).^3-X.^2.*Z.^3-9/80*Y.^2.*Z.^3;  
>> p = patch(isosurface(X,Y,Z,V,0));  
>> set(p, 'FaceColor', 'red', 'EdgeColor', 'none');  
>> view(3); axis equal ; axis off;  
>> light('Posi',[0 -2 3]);  
>> lighting phong  
>> set(gca,'nextplot','replacechildren');  
>> XX = get(p,'XData'); YY = get(p,'YData'); ZZ = get(p,'ZData');  
>> for j = 1:20  
    bili = sin(pi*j/20);  
    set(p,'XData',bili*XX,'YData',bili*YY,'ZData',bili*ZZ)  
    F(j) = getframe;  
end  
>> movie(F,10)
```



三、录制AVI格式视频动画

MATLAB中提供了**avifile**函数，用来把**getframe**函数抓取的电影帧录制为AVI格式视频动画。利用MATLAB录制AVI格式视频动画的步骤依次如下：

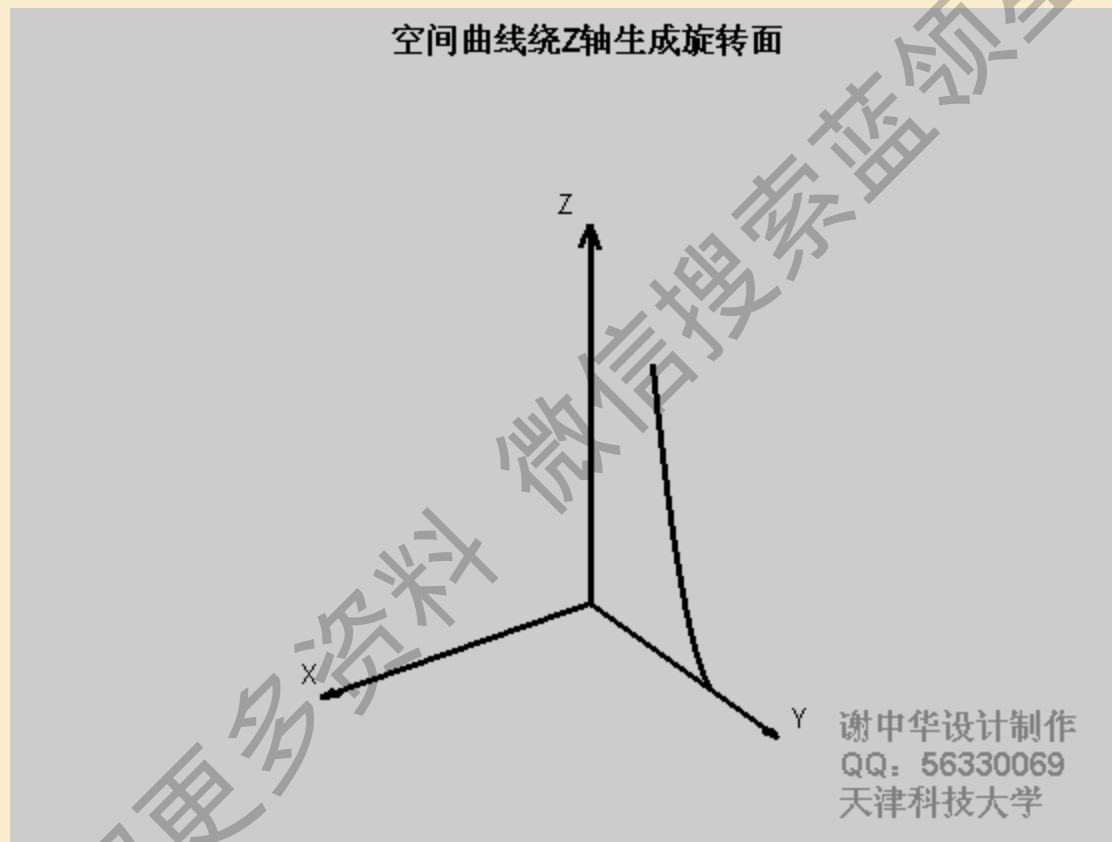
- 调用**avifile**函数先创建一个空白的avifile文件。
- 绘制视频的每一个帧所对应的图形。
- 调用**getframe**函数抓取当前图形窗口或坐标系中的图片。
- 调用**addframe**函数把抓取的图片添加到avifile文件中。
- 调用**close**函数关闭avifile文件。

【例5.5-5】录制空间曲线 $z = 3(y - 2)^2$, $1 \leq y \leq 2$ 绕Z轴旋转生成旋转面的AVI格式视频动画。

```
mov = avifile('旋转面.avi');
y = 1:0.01:2;
z = 3*(y-2).^2;
x = zeros(size(y));
theta = [pi/2,pi/2:pi/20:2.5*pi];
figure;
h = plot3(x,y,z,'k','LineWidth',2);
hold on
quiver3(0,0,0,1,0,0,3,'k','filled','LineWidth',2);
quiver3(0,0,0,1,0,3,'k','filled','LineWidth',2);
quiver3(0,0,0,0,1,4,'k','filled','LineWidth',2);
text(0,-0.5,4,'Z')
text(0,3,2,0,3,'Y')
text(3,2,0,0,3,'X')
axis equal
axis([-2,-2,-2,2,0,4])
axis off
view(145,30);
title('空间曲线绕Z轴生成旋转面','fontsize',12,'fontweight','bold')
for i = 1:length(theta)-1
    [r,alpha1] = meshgrid(y,linspace(theta(i),theta(i+1),10));
    zz = repmat(z,10,1);
    xx = r.*cos(alpha1);
    yy = r.*sin(alpha1);
    surf(xx,yy,zz);
    shading interp
    set(h,'XData',xx(end,:), 'YData',yy(end,:), 'ZData',zz(end,:))
    drawnow;
    pause(0.05)
    F = getframe(gcf);
    mov = addframe(mov,F);
end
hold off
mov = close(mov);
```



空间曲线绕Z轴生成旋转面



四、GIF 动画

在浏览网页的时候，我们会看到好多很炫的动画，它们大多都是GIF格式的图片。其实GIF格式动画是将多幅图像保存为一个图像文件，从而形成动画。MATLAB中制作GIF动画要用到`getframe`、`frame2im`、`rgb2ind`和`imwrite`函数，`getframe`函数用来抓取当前图形窗口或坐标系中的图像，`frame2im`函数和`rgb2ind`函数用来将抓取的图像转为索引图像，`imwrite`函数用来将索引图像写入GIF格式动画。

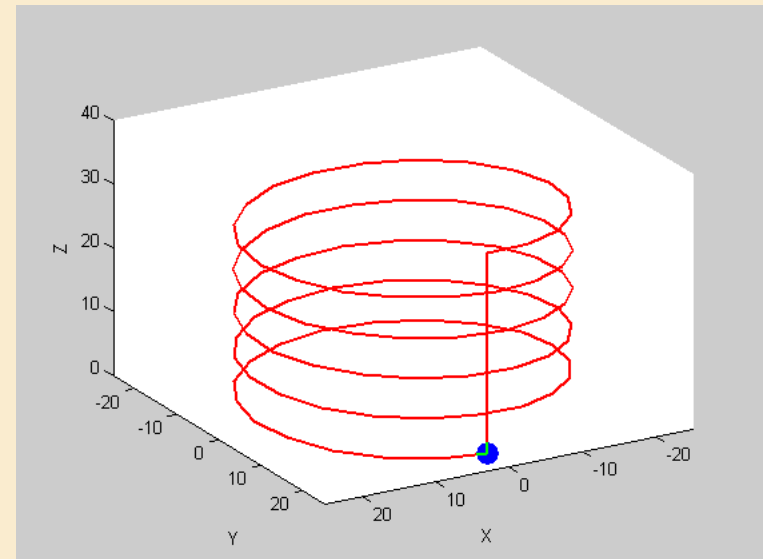
注意：`imwrite`函数不能将真彩图像（RGB图像）写入GIF格式动画，必须先将真彩图像转为索引图像或灰度图像，然后才能写入。

【例5.5-6】制作小球绕螺旋线运动的GIF格式动画。

```

>> filename = 'Ball.gif';
>> t = linspace(0, 10*pi, 100);
>> x = [20*sin(t),zeros(1,10)];
>> y = [20*cos(t),20*ones(1,10)];
>> z = [t,linspace(10*pi,0,10)];
>> plot3(x, y, z, 'r', 'linewidth', 2);
>> hold on
>> h = plot3(0,20,0,'.', 'MarkerSize',40, 'EraseMode', 'xor' );
>> xlabel('X'); ylabel('Y'); zlabel('Z');
>> axis([-25 25 -25 25 0 40]);
>> view(-210,30);
>> for i = 1:length(x)
    set(h, 'xdata',x(i), 'ydata',y(i), 'zdata',z(i));
    drawnow;
    pause(0.05)
    f = getframe(gcf);
    [IM,map] = frame2im(f);
    if isempty(map)
        [IM,map] = rgb2ind(IM,256);
    end
    if i == 1
        imwrite(IM,map,filename,'gif', 'Loopcount',inf,'DelayTime',0.1);
    else
        imwrite(IM,map,filename,'gif', 'WriteMode','append','DelayTime',0.1);
    end
end
end

```



【例5.5-7】 空间曲线绕空间曲线运动形成空间曲面。

>> 程序略

