第 7 章　绘图与图形处理

人们很难从一大堆原始的数据中发现它们的含义，而数据图形恰能使视觉感官直接感受到数据的许多内在本质，发现数据的内在联系。MATLAB 可以表达出数据的二维，三维，甚至四维的图形。通过图形的线型，立面，色彩，光线，视角等属性的控制，可把数据的内在特征表现得淋漓尽致。下面我们分别介绍图形的命令。

7.1　二维图形

7.1.1　基本平面图形命令

命令 1　**plot**

功能　线性二维图。在线条多于一条时，若用户没有指定使用颜色，则 plot 循环使用由当前坐标轴颜色顺序属性（current axes ColorOrder property）定义的颜色，以区别不同的线条。在用完上述属性值后，plot 又循环使用由坐标轴线型顺序属性（axes LineStyleOrder property）定义的线型，以区别不同的线条。

用法　plot(X,Y) 当 X,Y 均为实数向量，且为同维向量（可以不是同型向量），X=[x(i)]，Y=[y(i)]，则 plot(X,Y)先描出点(x(i)，y(i))，然后用直线依次相连；若 X，Y 为复数向量，则不考虑虚数部分。若 X，Y 均为同维同型实数矩阵，X = [X(i)]，Y = [Y(i)]，其中 X(i),Y(i)为列向量，则 plot(X,Y)依次画出 plot(X(i),Y(i))，矩阵有几列就有几条线；若 X，Y 中一个为向量，另一个为矩阵，且向量的维数等于矩阵的行数或者列数，则矩阵按向量的方向分解成几个向量，再与向量配对分别画出，矩阵可分解成几个向量就有几条线；在上述的几种使用形式中，若有复数出现，则复数的虚数部分将不被考虑。

plot(Y) 若 Y 为实数向量，Y 的维数为 m，则 plot(Y)等价于 plot(X,Y)，其中 x=1：m；若 y 为实数矩阵，则把 y 按列的方向分解成几个列向量，而 y 的行数为 n，则 plot(Y)等价于 plot(X,Y)其中 x=[1;2;…;n]；在上述的几种使用形式中，若有复数出现，则复数的虚数部分将不被考虑。

plot(X1,Y1,X2,Y2,…)，其中 Xi 与 Yi 成对出现，plot(X1,Y1,X2,Y2,…)将分别按顺序取两数据 Xi 与 Yi 进行画图。若其中仅仅有 Xi 或 Yi 是矩阵，其余的为向量，向量维数与矩阵的维数匹配，则按匹配的方向来分解矩阵，再分别将配对的向量画出。

plot(X1,Y1,LineSpec1,X2,Y2,LineSpec2…) 将按顺序分别画出由三参数定义 Xi,Yi,LineSpeci 的线条。其中参数 LineSpeci 指明了线条的类型，标记符号，和画线用的颜色。在 plot 命令中我们可以混合使用三参数和二参数的形式：

plot(X1,Y1,LineSpec1,X2,Y2,X3,Y3,LineSpec3)

plot(…,'PropertyName',PropertyValue,…) 对所有的用 plot 生成的 line 图形对象中指定的属性进行恰当的设置。

h = plot(…) 返回 line 图形对象句柄的一列向量，一线条对应一句柄值。

说明　参数 LineSpec

功能　定义线的属性。Maltab 允许用户对线条定义如下的特性：

1．线型

表 7-1

定义符	-	--	：	-.
线型	实线（缺省值）	划线	点线	点划线

2．线条宽度

指定线条的宽度，取值为整数（单位为像素点）

3．颜色

表 7-2

定义符	R（red）	G(green)	b(blue)	c(cyan)
颜色	红色	绿色	兰色	青色
定义符	M(magenta)	y(yellow)	k(black)	w(white)
颜色	品红	黄色	黑色	白色

4．标记类型

表 7-3

定义符	+	o(字母)	*	.	x
标记类型	加号	小圆圈	星号	实点	交叉号
定义符	d	^	v	>	<
标记类型	棱形	向上三角形	向下三角形	向右三角形	向左三角形
定义符	s	h	P		
标记类型	正方形	正六角星	正五角星		

5．标记大小

指定标记符号的大小尺寸，取值为整数（单位为像素）

6．标记面填充颜色

指定用于填充标记符面的颜色。取值在上表。

7．标记周边颜色

指定标记符颜色或者是标记符（小圆圈、正方形、棱形、正五角星、正六角星和四个方向的三角形）周边线条的颜色。取值在上表。

在所有的能产生线条的命令中，参数 LineSepc 可以定义线条的下面三个属性：线型、标记符号、颜色进行设置。对线条的上述属性的定义可用字符串来定义，如：plot(x,y,'-.or')

结合 x 和 y，画出点划线（-.），在数据点（x，y）处画出小圆圈（o），线和标记都用红色画出。其中定义符（即字符串）中的字母、符号可任意组合。若没有定义符，则画图命令 plot 自动用缺省值进行画图。若仅仅指定了标记符，而非线型，则 plot 只在数据点画出标记符。如：plot(x,y,'d')

例 7-1

```
>>t = 0:pi/20:2*pi;
>>plot(t,t.*cos(t),'-.r*')
>>hold on
>>plot(exp(t/100).*sin(t-pi/2),'--mo')
>>plot(sin(t-pi),':bs')
>>hold off
```

图形结果为图 7-1。

例 7-2

```
>>plot(t,sin(2*t),'-mo', 'LineWidth',2,'MarkerEdgeColor','k',…
    'MarkerFaceColor',[.49 1 .63],'MarkerSize',12)
```

图形结果为图 7-2。

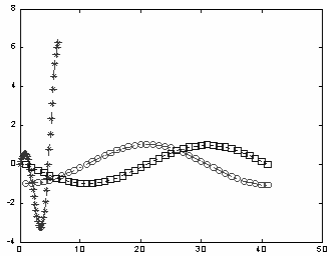

图 7-1　二维曲线图

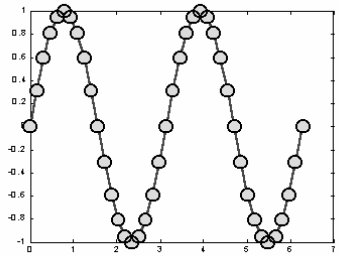

图 7-2　二维图形的绘制

命令 2　**fplot**

功能　在指定的范围 limits 内画出一元函数 y=f（x）的图形。其中向量 x 的分量分布在指定的范围内，y 是与 x 同型的向量，对应的分量有函数关系：y(i)=f(x(i))。若对应于 x 的值，y 返回多个值，则 y 是一个矩阵，其中每列对应一个 f（x）。例如，f（x）返回向量[f1(x),f2(x),f3(x)]，输入参量 x=[x1;x2;x3]，则函数 f（x）返回矩阵

f1(x1)　f2(x1)　f3(x1)
f1(x2)　f2(x2)　f3(x2)
f1(x3)　f2(x3)　f3(x3)

注意一点的是，函数 function 必须是一个 m-文件函数或者是一个包含变量 x，且能用函数 eval 计算的字符串。例如：'sin(x)*exp(2*x)'，'[sin(x),cos(x)]'，'hump(x)'。

用法　fplot('function',limits) 在指定的范围 limits 内画出函数名为 function 的一元函数图形。其中 limits 是一个指定 x-轴范围的向量[xmin xmax]或者是 x 轴和 y 轴的范围的向量[xmin xmax ymin ymax]。

fplot('function',limits,LineSpec) 用指定的线型 LineSpec 画出函数 function。

fplot('function',limits,tol) 用相对误差值为 tol 画出函数 function。相对误差的缺省值为 2e-3。

fplot('function',limits,tol,LineSpec) 用指定的相对误差值 tol 和指定的线型 LineSpec 画出函数 function 的图形。

fplot('function',limits,n) 当 n>=1，则至少画出 n+1 个点（即至少把范围 limits 分成 n 个小区间），最大步长不超过(xmax-xmin)/n。

fplot(‘function’,lims,…) 允许可选参数 tol，n 和 LineSpec 以任意组合方式输入。

[X,Y] = fplot('function',limits,⋯) 返回横坐标与纵坐标的值给变量 X 和 Y，此时 fplot 不画出图形。若想画出，可用命令 plot(X,Y)。

[⋯] = plot('function',limits,tol,n,LineSpec,P1,P2,⋯) 允许用户直接给函数 function 输入参数 P1，P2 等，其中函数 functiond 的定义形式为

y = function(x,P1,P2,⋯)

若想用缺省的 tol，n 或 LineSpec 值，只需将空矩阵（[ ]）传递给函数即可。

注意：fplot 采用自适应步长控制来画出函数 function 的示意图，在函数的变化激烈的区间，采用小的步长，否则采用大的步长。总之，使计算量与时间最小，图形尽可能精确。

例 7-3

```
>>fplot('tanh',[-2 2])
```

图形结果为图 7-3。

```
>>subplot(2,2,1);fplot('humps',[0 1])
>>subplot(2,2,2);fplot('abs(exp(-j*x*(0:9))*ones(10,1))',[0 2*pi])
>>subplot(2,1,2);fplot('[tan(x),sin(x),cos(x)]',2*pi*[-1 1 -1 1])
```

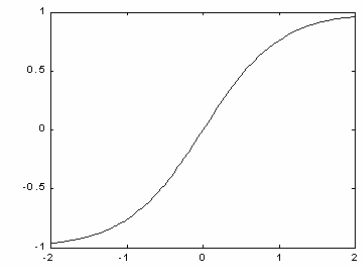

图 7-3　函数画图

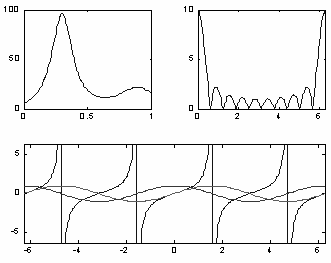

图 7-4

命令 3　**loglog**

功能　双对数图形。

用法　loglog(Y) 若 y 为实数向量或矩阵，则结合 y 列向量的下标与 y 的列向量画出。若 y 为复数向量或矩阵，则 loglog(Y)等价于 loglog(real(Y),imag(Y))，在 loglog 的其他使用形式中将忽略 Y 的虚数部分。

loglog(X1,Y1,X2,Y2⋯) 结合 Xn 与 Yn 画出图形。若只有 Xn 或 Yn 为矩阵，另一个为向量，行向量维数等于矩阵的列数，列向量的维数等于矩阵的行数，则 loglog 把矩阵按向量的方向分解成向量，再与向量结合分别画出图形。

loglog(X1,Y1,LineSpec1,X2,Y2,LineSpeec2 ⋯) 按顺序取三个参数 Xn,Yn,LineSpecn 画出线条，其中 LineSpecn 指定线条的线型，标记符号和颜色。用户可以混合使用二参数和三参数形式，如：

loglog(X1,Y1,X2,Y2,LineSpec2,X3,Y3)

loglog(⋯,'PropertyName',PropertyValue,⋯) 对所有由 loglog 命令生成的图形对象句柄的属性进行设置。

h = loglog(…) 返回 line 图形句柄向量，每条线对应一个句柄。

例 7-4

```
>>x = logspace(-1,2);
>>loglog(x,10*exp(x),'-s')
>>grid on
```

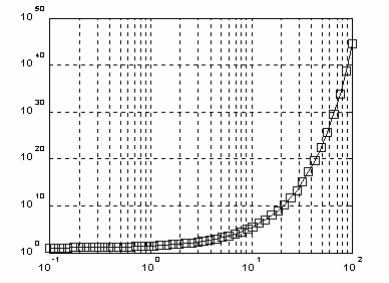

图 7-5

命令 4 **semilogx**

功能 x 轴对数图形。若没有指定使用的颜色，当所画线条较多时，semilogx 将自动使用由当前轴的 ColorOrder 和 LineStyleOrder 属性指定的颜色顺序和线型顺序来画线。

用法 semilogx(Y) %对 x 轴的刻度求常用对数（以 10 为底），而 y 轴为线性刻度。若 y 为实数向量或矩阵，则结合 y 列向量的下标与 y 的列向量画出线条；若 y 为复数向量或矩阵，则 semilogx(Y)等价于 semilogx(real(Y),imag(Y))。在 semilogx 的其他使用形式中，Y 的虚数部分将被忽略。

semilogx(X1,Y1,X2,Y2…) %结合 Xn 和 Yn 画出线条，若其中只有 xn 或 yn 为矩阵，另外一个为向量，行向量的维数等于矩阵的列数，列向量的维数等于矩阵的行数，则按向量的方向分解矩阵，再与向量结合，分别画出线条。

semilogx(X1,Y1,LineSpec1X2,Y2,LineSpec2, …) % 按顺序取三参数 Xn,Yn,LineSpecn 画线，参数 LineSpecn 指定使用的线型，标记符号和颜色。用户可以混合使用二参数和三参数形式，如：

semilogx(X1,Y1,X2,Y2,LineSpec2,X3,Y3)

semilogx(…,'PropertyName',PropertyValue,…) %对所有由 semilogx 命令生成的图形对象句柄的属性进行设置

h = semilogx(…) %返回 line 图形句柄向量，每条线对应一个句柄。

例 7-5

```
>>x = 0:.1:10;
  >>semilogx(x,cos(10.^x))
```

图形结果为图 7-6。

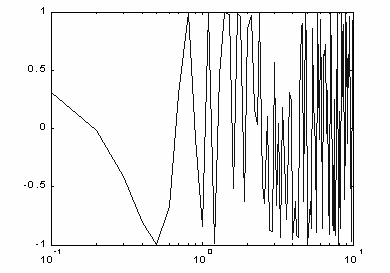

图 7-6

命令 5 **semilogy**

用法：参见 semilogx 命令。

命令 6 **fill**

功能 用颜色填充二维多边形。

用法 fill(X,Y,C) 用 x 和 y 中的数据生成多边形，

用 c 指定的颜色填充它。其中 c 为色图向量或矩阵。若 c 是行向量，则要求 c 的维数等于 x 和 y 的列数，若 c 为列向量，则要求 c 的维数等于 x 和 y 的行数。

fill(X,Y,ColorSpec) 用 ColorSpec 指定的颜色填充由 x 和 y 定义的多边形

fill(X1,Y1,C1,X2,Y2,C2,…) 指定多个要填充的二维区域

fill(…,'PropertyName',PropertyValue) 允许用户对一个 patch 图形对象的某个属性设定属性值。

h = fill(…) 返回 patch 图形对象句柄的向量，每一个 patch 对象对应一个句柄。

注意：

1. 若 x 或 y 是一矩阵，另一个是向量，向量应是维数与矩阵的行数相等的列向量或是维数等于矩阵列数的行向量时，函数 fill 将向量复制成与矩阵同型的矩阵。函数 fill 将矩阵 x 与 y 中列向量中的数据生成多边形的顶点。

2. 颜色阴影类型决定于用户在参数中列出的颜色，若用户用 ColorSpec 指定颜色，命令 fill 生成平坦阴影模式（flat-shaded）多边形，同时设置补片对象（patch）的 FaceColor 属性为相应的 RGB 颜色矩阵。

3. 若用户用参量 c 指定所用颜色，命令 fill 按坐标轴属性 Clim 的比例缩小 c 中的元素，之后，c 成为引用当前色图的下标矩阵。

4. 若 c 为行向量，命令 fill 生成平面阴影的多边形，c 的每一元素决定由矩阵 x，y 的每一列定义的多边形内的颜色，每一补片对象的 FaceColor 属性被设置为'flat'，x，y 的每一行元素变成第 n 块补片对象的 Cdata 属性值，其中 n 为矩阵 x 或 y 中的相应的列。

5. 若 c 为一列向量或一矩阵，命令 fill 运用一线性插值法计算每一节点的颜色，以便用插值颜色填充多边形的内部。它设置补片对象的 FaceColor 属性为 ‘ interp ’，且在一列中的元素变成每一补片的 Cdata 属性值。若 c 为一列向量，命令 fill 用该向量复制成需要大小的尺寸。

例 7-6

```
>>t = (1/16:1/8:1)'*2*pi;
>>x = exp(t).*sin(t);
>>y = t.*cos(t);
>>fill(x,y,'k')
>>grid on
```

图形结果为图 7-7。

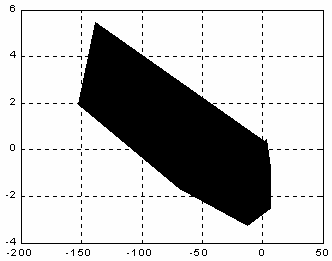

图 7-7

命令 7　**zoom**

功能　对二维图形进行放大或缩小。放大或缩小会改变坐标轴范围。

用法　zoom on 打开交互式的放大功能。当一个图形处于交互式的放大状态时，有两种方法来放大图形：

对于一键鼠标或二键，三键鼠标，单击坐标轴内的任意一点，可使图形放大一倍，这一操作可进行多次，直到 matlab 的最大显示为止；对于二键或三键的鼠标，在坐标轴内单击右键，可使图形缩小一倍，这一操作可进行多次，直到还原图形为止。对于一键鼠标，要想缩小图形，需要按住键盘上的 Shift 键，再单击鼠标键。

用鼠标拖出要放大的部分，系统将放大选定的区域。

zoom off 关闭交互式放大功能。

zoom out 将系统转回非放大状态，并将图形恢复原状。

zoom reset 系统将记住当前图形的放大状态，作为放大状态的设置值。以后使用 zoom out 或者是双击鼠标时，交互式放大状态打开，且图形并不是返回到原状，而是返回 reset 时的放大状态。

zoom 用于切换放大的状态：on 和 off。

zoom xon 只对 x 轴进行放大。

zoom yon 只对 y 轴进行放大。

zoom(factor) 用放大系数 factor 进行放大或缩小，而不影响交互式放大的状态。若 factor>1，系统将图形放大 factor 倍，若 0<factor≤1，系统将图形放大 1/factor 倍。

zoom(fig, option) 指定对窗口 fig 中（不一定为当前窗口）的二维图形进行放大，其中参数 option 为：on、off、xon、yon、reset、factor 等。

命令 8 **meshgrid**

功能 生成二元函数 z = f(x,y)中 x-y 平面上的矩形定义域中数据点矩阵 X 和 Y，或者是三元函数 u = f(x,y,z)中立方体定义域中的数据点矩阵 X,Y 和 Z。

用法 a：[X,Y] = meshgrid(x,y)

b：[X,Y] = meshgrid(x)

c：[X,Y,Z] = meshgrid(x,y,z)

说明 对于形式 a，输入向量 x 为 x-y 平面上矩形定义域的矩形分割线在 x 轴的值，向量 y 为 x-y 平面上矩形定义域的矩形分割线在 y 轴的值。输出向量 X 为 x-y 平面上矩形定义域的矩形分割点的横坐标值矩阵，输出向量 Y 为 x-y 平面上矩形定义域的矩形分割点的纵坐标值矩阵。

对于形式 b，等价于形式 a：[X,Y] = meshgrid(x) = meshgrid(x,x)。

对于形式 c，输入向量 x 为立方体定义域的立方体分割平面在 x 轴上的值，输入向量 y 为立方体定义域的立方体分割平面在 y 轴上的值，输入向量 z 为立方体定义域的立方体分割平面在 z 轴上的值。输出向量 X 为立方体定义域中分割点的 x 轴坐标值,Y 为立方体定义域中分割点的 y 轴坐标值,Z 为立方体定义域中分割点的 z 轴坐标值。

例 7-7

```
>>x = [0.7 1.1 ]; y = [-2 3 1]; z = [2 5 3];   %分量不一定从小到大
>>[X_2d,Y_2d] = meshgrid(x,y)
>>[X_3d,Y_3d,Z_3d] = meshgrid(x,y,z)
```

计算结果为：

```
X_2d =
    0.7000    1.1000
    0.7000    1.1000
    0.7000    1.1000
Y_2d =
    -2    -2
     3     3
     1     1
X_3d(:,:,1) =
    0.7000    1.1000
```

```
    0.7000    1.1000
    0.7000    1.1000
X_3d(:,:,2) =
    0.7000    1.1000
    0.7000    1.1000
    0.7000    1.1000
X_3d(:,:,3) =
    0.7000    1.1000
    0.7000    1.1000
    0.7000    1.1000
Y_3d(:,:,1) =
    -2    -2
     3     3
     1     1
Y_3d(:,:,2) =
    -2    -2
     3     3
     1     1
Y_3d(:,:,3) =
    -2    -2
     3     3
     1     1
Z_3d(:,:,1) =
     2     2
     2     2
     2     2
Z_3d(:,:,2) =
     5     5
     5     5
     5     5
Z_3d(:,:,3) =
     3     3
     3     3
     3     3
```

7.1.2　特殊平面图形命令

命令 1　**polar**

功能　画极坐标图。该命令接受极坐标形式的函数 rho=f(θ)，在笛卡儿坐标系平面上画出该函数，且在平面上画出极坐标形式的格栅。

用法　polar(theta,rho)　用极角 theta 和极径 rho 画出极坐标图形。极角 theta 为从 x 轴到半径的单位为弧度的向量，极径 rho 为各数据点到极点的半径向量。

polar(theta,rho,LineSpec)　参量 LineSpec 指定极坐标图中线条的线型、标记符号和颜色等。

例 7-8

```
>>t = 0:.01:2*pi;
>>polar(t,sin(3*t).*cos(2*t),'--r')
```

图形结果为图 7-8。

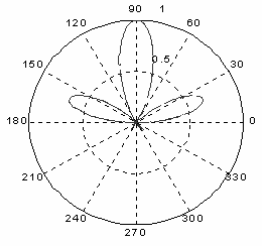

图 7-8

命令 2　**bar**

功能　二维垂直条形图。用垂直条形显示向量或矩阵中的值。

用法　bar(Y)　若 y 为向量，则分别显示每个分量的高度，横坐标为 1 到 length（y）；若 y 为矩阵，则 bar 把 y 分解成行向量，再分别画出，横坐标为 1 到 size（y，1），即矩阵的行数。

bar(x,Y) 在指定的横坐标 x 上画出 y，其中 x 为严格单增的向量。若 y 为矩阵，则 bar 把矩阵分解成几个行向量，在指定的横坐标处分别画出。

bar(…,width) 设置条形的相对宽度和控制在一组内条形的间距。缺省值为 0.8，所以，如果用户没有指定 x，则同一组内的条形有很小的间距，若设置 width 为 1，则同一组内的条形相互接触。

bar(…,'style') 指定条形的排列类型。类型有“ group ”和“ stack ”，其中“ group ”为缺省的显示模式。

“ group ”：若 y 为 n*m 阶的矩阵，则 bar 显示 n 组，每组有 m 个垂直条形的条形图。

“ stack ”：对矩阵 y 的每一个行向量显示在一个条形中，条形的高度为该行向量中的分量和。其中同一条形中的每个分量用不同的颜色显示出来，从而可以显示每个分量在向量中的分布。

bar(…,LineSpec) 用指定的颜色 LineSpec 显示所有的条形。

[xb,yb] = bar(…) 返回用户可用命令 plot 或命令 patch 画出条形图的参量 xb，yb。这对用户控制一个图形的显示是有用的，例如要在一个 plot 语句中加入装饰性的条形图等。

h = bar(…) 返回一个 patch 图形对象句柄的向量。每一条形对应一个句柄。

例 7-9

```
x = -2.9:0.2:2.9;
bar(x,exp(x.*sin(x)))
colormap gray
```

图形结果为图 7-9。

例 7-10

```
subplot(2,2,4)
bar(Y,1.5)
title 'Width = 1.5'
```

图形结果为图 7-10。

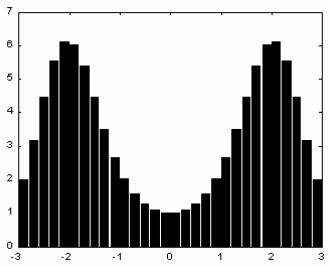

图 7-9

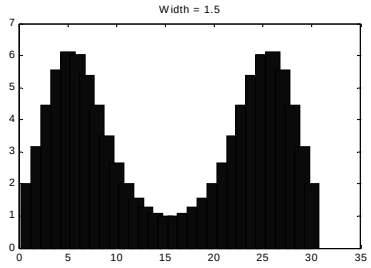

图 7-10

命令 3 **barh**

功能 二维水平条形图。用水平条形显示向量或矩阵中的值。

用法 barh(Y) 若 y 为向量，则分别显示每个分量的高度，纵坐标为 1 到 length(y)；若 y 为矩阵，则 bar 把 y 分解成行向量，再分别画出，纵坐标为 1 到 size(y,1)，即矩阵的行数。

barh(x,Y) 在指定的纵坐标 x 上以水平方向画出 y，其中 x 为严格单增的向量。若 y 为矩阵，则 barh 把矩阵分解成几个行向量，在指定的纵坐标处分别画出。

barh(…,width) 设置条形的相对宽度和控制在一组内条形的间距。缺省值为 0.8，所以，如果用户没有指定 x，则同一组内的条形有很小的间距，若设置 width 为 1，则同一组内的条形相互接触。

barh(…,'style') 指定条形的排列类型。类型有“ group ”和“ stack ”，其中“ group ”为缺省的显示模式。

“ group ”：若 y 为 n*m 阶的矩阵，则 bar 显示 n 组，每组有 m 个水平条形的条形图。

“ stack ”：对矩阵 y 的每一个行向量显示在一个条形中，条形的高度为

该行向量中的分量和。其中同一条形中的每个分量用不同的颜色显示出来，从而可以显示每个分量在向量中的分布。

barh(…,LineSpec)　用指定的颜色 LineSpec 显示所有的条形。

[xb,yb] = barh(…)　返回用户可用命令 plot 或命令 patch 画出条形图的参量 xb，yb。这给用户控制一个图形的显示是有用的，例如要在一个 plot 语句中加入装饰性的条形图等。

h = barh(…)　返回一个 patch 图形对象句柄的向量。每一条形对应一个句柄。

例 7-11

```
>>X = 1:.5:5;
>>Y = exp(X).*sin(X);
>>barh(Y,'stack')
```

图形结果为图 7-11。

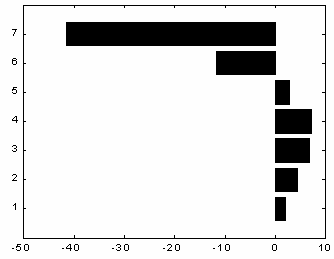

图 7-11

命令 4　**compass**

功能　从原点画箭头图。箭头图为一显示起点为笛卡儿坐标系中的原点的二维或三维方向或向量的图形，同时在坐标系中显示圆形的分隔线。

用法　compass(X,Y)　参量 x 与 y 为同型的 n 维向量，则命令显示 n 个箭头，箭头的起点为原点，箭头的位置为[X(i),Y(i)]。

compass(Z)　参量 z 为 n 维复数向量，则命令显示 n 个箭头，箭头起点为原点，箭头的位置为[real(Z),imag(Z)].

compass(…,LineSpec)　用参量 LineSpec 指定箭头图的线型、标记符号、颜色等属性。

h = compass(…)　返回 line 对象的句柄给 h。

例 7-12

```
Z = magic(20).*randn(20);
compass(Z)
```

图形结果为图 7-12。

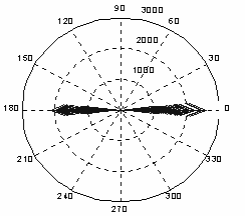

图 7-12

命令 5　**comet**

功能　二维彗星图。彗星图为彗星头（一个小圆圈）沿着数据点前进的动画，彗星体为跟在彗星头后面的痕迹，轨道为沿着整个函数的实线。我们要指出的是，由命令 comet 生成的轨迹是使用擦除模式（EraseMode）属性的值为 none，该属性使用户不能打印该图形（只能得到彗星头），且当用户改变窗口的大小时，动画将消失。

用法　comet(y)　彗星图动画显示向量 y 确定的路线。

comet(x,y)　彗星图动画显示向量 x 与 y 确定的路线。

comet(x,y,p)　指定彗星体的长度 p*length(y)，缺省的 p 值为 0.1。

例 7-13

```
>>t = 0:.01:2*pi;
>>x = exp(sin(2*t)).*(cos(t).^2/3);
>>y = t.*(sin(t).^2);
>>comet(x,y);
```

图形结果为图 7-13。

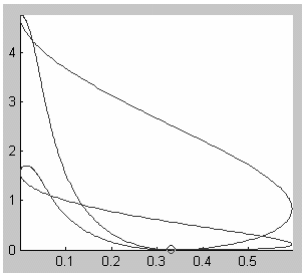

图 7-13

附：擦除模式(EraseMode)属性及属性值：{normal}|none|xor|background

该属性控制系统用于显示与擦除线条对象的技术。不同的擦除模式对于生成动画系列，即控制个别对象的重新显示方式，对于改进外在显示和获得理想的效果是很必要的。

表 7-4

属性值	含义
Normal (缺省值)	重新显示受影响的区域，在必要的时候，进行三维分析计算，以保证所有的对象的显示都是正确的。该模式下的图形显示是最精确的，不过也是最缓慢的，以下其他三种模式显示速度较快，不过没有执行一个完全的重显过程，因而，图形显示也不是很精确的。
none	当线条移动或改动时，该模式没有擦除线条，而是仍然显示于屏幕上。该模式下不能打印图形，因为系统没有存储前一图形的任何信息。
xor	使用异或运算(xor)计算线条颜色与当前位置下的颜色，用所得结果显示与擦除线条。该模式对于线条下面对象的颜色没有任何破坏，只是影响到线条的当前显示颜色而已。
Background	用当前坐标轴颜色重新显示线条的方式来擦除线条，若当前坐标轴颜色设置为 none，则用图形的背景色来代替坐标轴颜色。该模式对于处于擦除线条后面的对象来说是有损害的，不过当前线条的颜色总是最合适的。

命令 6　**errorbar**

功能　沿着一曲线画误差棒形图。误差棒为数据的置信水平或者为沿着曲线的偏差。在下列参数中，若为矩阵，则按列画出误差棒。

用法　errorbar(Y,E)　画出向量 y，同时显示在向量 y 的每一元素之上的误差棒。误差棒为 E(i)在曲线 y 上面与下面的距离，所以误差棒的长度为 2*E(i)。

errorbar(X,Y,E)　X,Y,E 必须为同型参量。若同为向量，则画出带长度为 2*E(i)、对称误差棒于曲线点(X(i),Y(i))之处；若同为矩阵，则画出带长度为 E(i，j)、对称误差棒于曲面点(X(i,j),Y(i,j))之处，

errorbar(X,Y,L,U)　X，Y，L，U 必须为同型参量。若同为向量，则在点(X(i),Y(i))处画出向下长为 L(i)，向上长为 U(i)的误差棒；若同为矩阵，则在点(X(i,j),Y(i.j))处画出向下长为 L(i,j),向上长为 U(i,j)的误差棒。

errorbar(…,LineSpec)　用 LineSpec 指定的线型、标记符、颜色等画出误差棒。

h = errorbar(…)　返回线图形对象的句柄向量给 h。

例 7-14

```
>>X = 0:pi/10:pi;
>>Y = exp(X).*sin(X);
>>E = std(Y)*ones(size(X));
>>errorbar(X,Y,E)
```

图形结果为图 7-14。

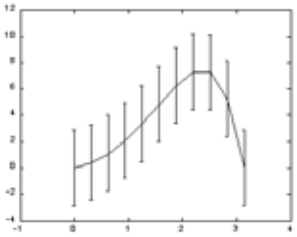

图 7-14

命令 7　**feather**

功能　画出速度向量图。一羽毛图在横坐标上等距地显示向量。用户要表示各个向量的、相对于原点的向量分量。

用法　feather(U,V)　显示由参量向量 u 与 v 确定的向量，其中 u 包含作为相对坐标系中的 x 成分，v 包含作为相对坐标系中的 y 成分。

feather(Z)　显示复数参量向量 z 确定的向量，等价于 feather(real(Z),imag(Z))。

feather(…,LineSpec) 用参量 LineSpec 指定的线型、标记符号、颜色等属性画出羽毛图。

例 7-15

```
>>th = (-90:10:90)*pi/180;
>>r = 4*ones(size(th));
>>[u,v] = pol2cart(th,r);
>>feather(u,v);
```

图形结果为图 7-15。

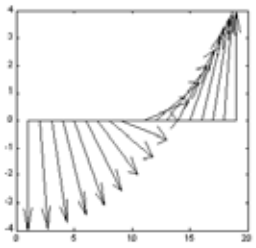

图 7-15

命令 8　**hist**

功能　二维条形直方图，可以显示出数据的分配情形。所有向量 y 中的元素或者是矩阵 y 中的列向量中的元素是根据它们的数值范围来分组的，每一组作为一个条形进行显示。条形直方图中的 x 轴反映了数据 y 中元素数值的范围，直方图的 y 轴显示出参量 y 中的元素落入该组的数目。所以 y 轴的范围从 0 到任一条形中包含元素最多的数字。直方图为一 patch 图形对象，若想改变图形的颜色，可以对 patch 对象的属性进行设置。缺省时，图形颜色是由当前色图进行控制，当前色图的第一个颜色为直方图的颜色。

用法　n = hist(Y) 把向量 y 中的元素放入等距的 10 个条形中，且返回每一个条形中的元素个数。若 y 为矩阵，则该命令按列对 y 进行处理。

n = hist(Y,x) 参量 x 为向量，把 y 中元素放到 m（m=length(x)）个由 x 中元素指定的位置为中心的条形中。

n = hist(Y,nbins) 参量 nbins 为标量，用于指定条形的数目。

[n,xout] = hist(…) 返回向量 n 与包含频率计数与条形的位置向量 xout，用户可以用命令 bar(xout,n)画出条形直方图。

例 7-16

```
>>x = -5:0.1:5;
>>y = randn(1000,1);
>>hist(y,x)
```

图形结果为图 7-16。

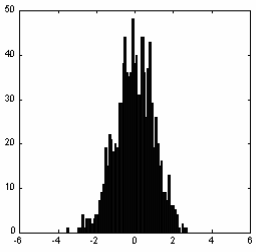

图 7-16

命令 9　**histc**

功能　直方图记数

用法　n = histc(x,edges) 统计向量 x 中、落入向量 edges（元素必须为单调的非减的）各个元素之间的元素个数。输出参量 n 为一与向量 edges 同维的向量。其中若有 edges(k)>=x(i)>=edges(k+1)，则 n(k)增加 1。X 中超出向量 edges 规定的范围的元素将不被统计。参量 edges 中可使用-inf 与 inf，用于包括向量 x 中非 NaN 的元素。若 x 为一矩阵，则对 x 的每一列进行上述操作。

n = histc(x,edges,dim) 对多维矩阵的第 dim 维进行统计。

[n,bin] = histc(…) n 结果同上，同时返回矩阵下标 bin。若 x 为向量，n(k) = sum(bin == k)。对于超出范围的数值，bin 为零值。

命令 10　**rose**

功能　画角度直方图。该直方图是一个显示所给数据的变化范围内数据的分布情形的极坐标图，所给数据分成不同的组。每一组作为一小扇形进行显示。

用法　rose(theta)　画一角度直方图，显示参数 theta 的数据在 20 个区间或更少的区间内的分布。向量 theta 中的角度单位为弧度，用于确定每一区间与原点的角度。每一区间的长度反映出输入参量的元素落入一区间的个数。

rose(theta,x) 用参量x指定每一区间内的元素与区间的位置，length(x)等于每一区间内元素的个数与每一区间位置角度的中间角度。例如，若x为一 5 维向量，rose命令分配参量theta中的元素为 5 部分，每一部分的角度中线由x指定。

rose(theta,nbins) 于区间 [0,2*pi] 内画出 nbins 个等距的小扇形。缺省值为 20。

[tout,rout] = rose(…) 返回向量 tout 与 rout，可以用 polar(tout,rout)画出图形。该命令没有画任何的图形。

例 7-17

```
>>theta = 3*pi*randn(1,30);
>>rose(theta)
```

图形结果为图 7-17。

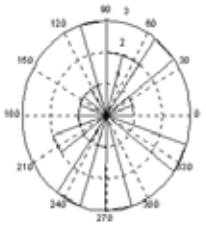

图 7-17

命令 11　**stairs**

功能　画二维阶梯图，这种图对与时间有关的数字样本系统的作图很有用处。

用法　stairs(Y) 用参量 y 的元素画一阶梯图。若 y 为向量，则横坐标 x 的范围从 1 到 m=length(y)，若 y 为矩阵，则对 y 的每一行画一阶梯图，其中 x 的范围从 1 到 y 的列数 m。

stairs(X,Y) 结合 x 与 y 画阶梯图。其中要求 x 与 y 为同型的向量或矩阵。此外，x 可以为行向量或为列向量，且 y 为有 m=length（x）行的矩阵。

stairs(…,LineSpec) 用参数 LineSpec 指定的线型、标记符号和颜色画阶梯图。

[xb,yb] = stairs(Y) 该命令没有画图，而是返回可以用命令 plot 画出参量 y 的阶梯图的向量 xb 与 yb。

[xb,yb] = stairs(X,Y) 该命令没有画图，而是返回可以用命令 plot 画出参量 x，y 的阶梯图的向量 xb 与 yb。

例 7-18

```
>>x = 0:.25:10;
>>stairs(x,exp(sin(x.^2)))
```

图形结果为图 7-18。

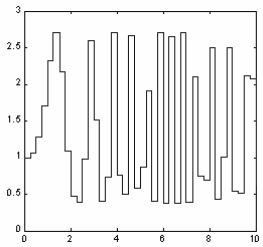

图 7-18

命令 12　**stem**

功能　画二维离散数据的柄形图。该图用线条显示数据点与 x 轴的距离，一小圆圈（缺省标记）或用指定的其他标记符号与线条相连，在 y 轴上标记数据点的值。

用法　stem(Y) 按 y 元素的顺序画出柄形图，在 x 轴上，柄与柄之间的距离相等；若 y 为矩阵，则把 y 分成几个行向量，在同一横坐标的位置上画出一个行向量的柄图。

stem(X,Y) 在横坐标 x 上画出列向量 y 的柄形图 。其中 x 与 y 为同型的向量或矩阵，此外，x 可以为行向量或列向量，而 y 为有 m=length（x）行的矩阵。

stem(…,'fill') 指定是否对柄形图末端的小圆圈填充颜色。

stem(…,LineSpec) 用参数 LineSpec 指定线型，标记符号和柄图末端的小圆圈的颜色画柄图。

h = stem(…) 返回柄形图的 line 图形对象句柄向量。

例 7-19

```
>>x = linspace(0,2,10);
>>stem(exp(-x.^2),'fill','-.')
```

图形结果为图 7-19。

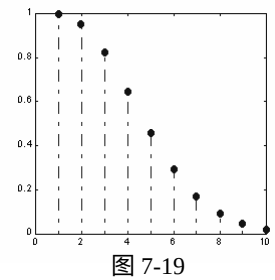

图 7-19

命令 13　**stem3**

功能　画三维离散数据的柄形图。该图用一线段显示数据离开 xy 平面的高度，在线段的末端用一小圆圈（缺省记号）或其他的标记符号表示数据的高度。

格式　stem3(Z) 用柄形图显示 z 中数据与 xy 平面的高度。若 z 为一行向量，则 x 与 y 将自动生成，stem3 将在与 x 轴平行的方向上等距的位置上画出 z 的元素；若 y 为列向量，stem3 将在与 y 轴平行的方向上等距的位置上画出 z 的元素。

stem3(X,Y,Z) 在参数 x 与 y 指定的位置上画出 z 的元素，其中 x，y，z 必须为同型的向量或矩阵。

stem3(…,'fill') 指定是否要填充柄形图末端小圆圈。

stem3(…,LineSpec) 指定线型，标记符号和末端小圆圈的颜色。

h = stem3(…) 返回柄形图的 line 图形对象句柄。

例 7-20

```
[X,Y,Z] = peaks(20);
stem3(X,Y,Z,'r*')
```

图形结果为图 7-20。

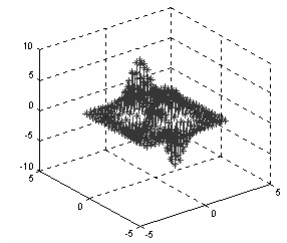

图 7-20

命令 14　**pie**

功能　饼形图

格式　pie(X)　用 x 中的数据画一饼形图，x 中的每一元素代表饼形图中的一部分。X 中元素 X(i)所代表的扇形大小通过 X(i)/sum(X)的大小来决定。若有 sum(X)=1，则 x 中元素就直接指定了所在部分的大小；若 sum(X)<1，则画出一不完整的饼形图。

pie(X,explode) 从饼形图中分离出一部分，explode 为元素为零或非零的、与 x 相对应的向量或矩阵。与 explode 的非零值对应的部分将从饼形图中心分离出来。explode 必须与 x 同型。

h = pie(…) 返回一 patch 与 text 的图形对象句柄向量 h。

例 7-21

```
>>x = [1 3 0.5 2.5 2];
>>explode = [0 1 0 0 0];
>>pie(x,explode)
```

图形结果为图 7-21。

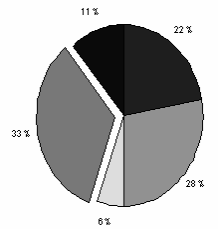

图 7-21

7.1.3　二维图形注释命令

命令 1　**grid**

功能　给二维或三维图形的坐标面增加分隔线。该命令会对当前坐标轴的 Xgrid，Ygrid，

Zgrid 的属性有影响。

用法　grid on 给当前的坐标轴增加分隔线。

grid off 从当前的坐标轴中去掉分隔线。

grid 转换分隔线的显示与否的状态。

grid(axes_handle,on|off) 对指定的坐标轴 axes_handle 是否显示分隔线。

命令 2　**gtext**

功能　在当前二维图形中用鼠标放置文字。当光标进入图形窗口时，会变成一个大十字，表明系统正等待用户的动作。

用法　gtext('string') 当光标位于一个图形窗口内时，等待用户单击鼠标或键盘。若按下鼠标或键盘，则在光标的位置放置给定的文字“ string ”

h = gtext('string') 当用户在鼠标指定的位置放置文字“ string ”后，返回一个 text 图形对象句柄给 h。

命令 3　**legend**

功能　在图形上添加图例。该命令对有多种图形对象类型（线条图，条形图，饼形图等）的窗口中显示一个图例。对于每一线条，图例会在用户给定的文字标签旁显示线条的线型，标记符号和颜色等。当所画的是区域（patch 或 surface 对象）时，图例会在文字旁显示表面颜色。Matlab 在一个坐标轴中仅仅显示一个图例。图例的位置有几个因素决定，像遮挡的对象等，用户可以用鼠标拖动图例到恰当的位置，双击标签可以进入标签编辑状态。

用法　legend('string1','string2',…) 用指定的文字 string 在当前坐标轴中对所给数据的每一部分显示一个图例。

legend(h,'string1','string2',…) 用指定的文字 string 在一个包含于句柄向量 h 中的图形显示图例。用给定的数据对相应的图形对象加上图例。

legend(string_matrix) 用字符矩阵参量 string_matrix 的每一行字符串作为标签。

legend(h,string_matrix) 用字符矩阵参量 string_matrix 的每一行字符串作为标签给包含于句柄向量 h 中的相应的图形对象加标签。

legend(axes_handle,…) 给由句柄 axes_handle 指定的坐标轴显示图例。

legend('off') 从当前的坐标轴，或是由 axes-handle 指定的坐标轴中除掉图例。

legend(axes_handle,'off') 从由 axes_handle 指定的坐标轴中除掉图例。

legend_handle = legend 返回当前坐标轴中的图例句柄，若坐标轴中没有图例存在，则返回空向量。

legend 对当前图形中所有的图例进行刷新。

legend(legend_handle) 对由句柄 legend_handle 指定的图例进行刷新。

legend(…,pos) 在指定的位置 pos 放置图例见表 7-5。

表 7-5

Pos 取值	pos=-1	pos=0	Pos=1
图例位置	坐标轴之外的右边	坐标轴之内，有可能遮挡部分图形	坐标轴的右上角（缺省位置）
Pos 取值	pos=2	pos=3	pos=4
图例位置	坐标轴的左上角	在坐标轴的左下角	坐标轴的右下角

h = legend(…) 返回图例的句柄向量。

[legend_handle,object_handles] = legend(…) 返回图例句柄，该句柄为坐标轴定义于图例中的图形对象、line 对象、text 对象的句柄。这些句柄允许用户对每个对象进行详细的操作。

例 7-22

```
>>x = -pi:pi/20:pi;
>>plot(x,(cos(x)).^2,'rd',x,asin(x),'-.b')
>>h = legend('cos2x','asin',2);
```

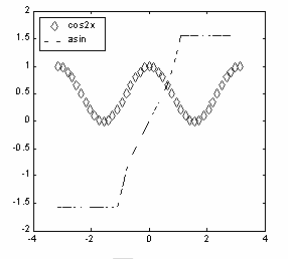

图 7-22

图形结果为图 7-22。

命令 4　**title**

功能　给当前轴加上标题。每个 axes 图形对象可以有一个标题。标题定位于 axes 的上方正中央。

用法　title('string') 在当前坐标轴上方正中央放置字符串 string 作为标题

title(fname) 先执行能返回字符串的函数 fname，然后在当前轴上方正中央放置返回的字符串作为标题

title(…,'PropertyName',PropertyValue,…) 对由命令 title 生成的 text 图形对象的属性进行设置

h = title(…) 返回作为标题的 text 对象句柄。

命令 5　**text**

功能　在当前轴中创建 text 对象。函数 text 是创建 text 图形句柄的低级函数。可用该函数在图形中指定的位置上显示字符串。

用法　text(x,y,'string')在图形中指定的位置(x,y)上显示字符串 string

text(x,y,z,'string') 在三维图形空间中的指定位置(x,y,z)上显示字符串 string

text(x,y,z,'string'.'PropertyName',PropertyValue…) 对引号中的文字 string 定位于用坐标轴指定的位置，且对指定的属性进行设置。表 7-6 给出文字属性名、含义及属性值。

表 7-6

属性名	属性说明	属性值
定义字符串		
Editing	能否对文字进行编辑	有效值：on、off 缺省值：off
Interpretation	TeX 字符是否可用	有效值：tex、none 缺省值：tex
String	字符串(包括 TeX 字符串)	有效值：可见字符串
放置字符串		
Extent	text 对象的范围（位置与大小）	有效值：[left, bottom, width, height]
HorizontalAlignment	文字水平方向的对齐方式	有效值：left(文本外框左边对齐，缺省对齐方式)、center(文本外框中间对齐)、right(文本外框右边对齐) 缺省值：left
Position	文字范围的位置	有效值：[x,y,z]直角坐标系 缺省值：[]（空矩阵）
Rotation	文字对象的方位角度	有效值：标量（单位为度） 缺省值：0

Units	文字范围与位置的单位	有效值：pixels（屏幕上的像素点）、normalized (把屏幕看成一个长、宽为 1 的矩形)、inches(英寸)、centimeters(厘米)、points (图象点)、data 缺省值：data
VerticalAlignment	文字垂直方向的对齐方式	有效值：top (文本外框顶上对齐)、cap(文本字符顶上对齐)、middle(文本外框中间对齐)、baseline(文本字符底线齐)、bottom(文本外框底线对齐) 缺省值：middle
指定文字字体		
FontAngle	设置斜体文字模式	有效值：normal(正常字体)、italic(斜体字)、oblique(斜角字) 缺省值：normal
FontName	设置文字字体名称	有效值：用户系统支持的字体名或者字符串 FixedWidth。 缺省值为 Helvetica
FontSize	文字字体大小	有效值：结合字体单位的数值 缺省值为：10 points
FontUnits	设置属性 FontSize 的单位	有效值：points（1 点=1/72 英寸）、normalized(把父对象坐标轴作为一单位长的一个整体；当改变坐标轴的尺寸时，系统会自动改变字体的大小)、inches（英寸）、Centimeters(厘米)、Pixels(像素) 缺省值：points
FontWeight	设置文字字体的粗细	有效值：light(细字体)、normal(正常字体)、demi(黑体字)、Bold(黑体字) 缺省值：normal
控制文字外观		
Clipping	设置坐标轴中矩形的剪辑模式	有效值：on、off on：当文本超出坐标轴的矩形时，超出的部分不显示； off：当文本超出坐标轴的矩形时，超出的部分显示。 缺省值：off
EraseMode	设置显示与擦除文字的模式。这些模式对生成动画系列与改进文字的显示效果很有好处。	有效值：normal、none、 xor、 background 缺省值：normal
SelectionHighlight	设置选中文字是否突出显示	有效值：on、off 缺省值：on
Visible	设置文字是否可见	有效值：on、off 缺省值：on
Color	设置文字颜色	有效的颜色值：ColorSpec
控制对文字对象的访问		
HandleVisibility	设置文字对象句柄对其他函数是否可见	有效值：on、callback、off 缺省值：on
HitTest	设置文字对象能否成为当前对象（见图形 CurrentObject 属性）	有效值：on、off 缺省值：on
文字对象的一般信息		
Children	文字对象的子对象（文字对象没有子对象）	有效值：[]（即空矩阵）
Parent	文字对象的父对象（通常为 axes 对象）	有效值：axes 的句柄
Seleted	设置文字是否显示出“选中”状态	有效值：on、off 缺省值：off

Tag	设置用户指定的标签	有效值：任何字符串 缺省值：”（即空字符串）
Type	设置图形对象的类型（只读类型）	有效值：字符串’text’
UserData	设置用户指定数据	有效值：任何矩阵 缺省值：[]（即空矩阵）
控制回调例行执行程序		
BusyAction	设置如何处理对文字回调过程中断的句柄	有效值：cancel、queue 缺省值：queue
ButtonDownFcn	设置当鼠标在文字上单击时，程序做出的反应（即执行回调程序）	有效值：字符串 缺省值：' '（空字符串）
CreateFcn	设置当文字被创建时，程序做出的反应（即执行的回调程序）	有效值：字符串 缺省值：' '（空字符串）
DeleteFcn	设置当文字被删除（通过关闭或删除操作）时，程序做出的反应（即执行的回调程序）	有效值：字符串 缺省值：' '（空字符串）
Interruptible	设置回调过程是否可中断	有效值：on、off 缺省值：on（能中断）
UIContextMenu	设置与文字相关的菜单项	有效值：用户相关菜单句柄

h = text(…) 返回文字对象句柄的列向量，每一对象对应一句柄。该命令的其他使用形式中，将随意地返回这个输出参量。

例 7-23

```
>>plot(0:pi/20:2*pi,sin(0:pi/20:2*pi))
>>text(pi,0,’Zeros Point’)
>>grid on
```

图形结果为图 7-23。

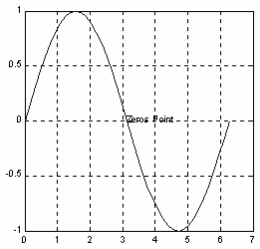

图 7-23

命令 6　**xlabel、ylabel**

功能　给 x、y 轴贴上标签

用法　xlabel('string')、 ylabel(‘string’)　给当前轴对象中的 x、y 轴贴标签；注意：若再次执行 xlabel 或 ylabel 命令，则新的标签会覆盖旧的标签。

xlabel(fname)、ylabel(fname) 先执行函数 fname，其返回一个字符串，然后在 x、y 轴旁边显示出来；

xlabel(… ,'PropertyName',PropertyValue, …) 、 ylabel(… ,’PropertyName’, PropertyValue)指定轴对象中的要控制的属性名和要改变的属性值，这些都是由 xlabel 或 ylabel 创建的 text 图形对象的成对值；

h = xlabel(…)、h = ylabel(…)返回作为标签的 text 对象的句柄。

7.2　三维图形

7.2.1　三维曲线、面填色命令

命令 1　**comet3**

功能　三维空间中的彗星图。彗星图为一个三维的动画图像，彗星头（一个小圆圈）沿

着数据指定的轨道前进，彗星体为跟在彗星头后面的一段痕迹，彗星轨道为整个函数所画的实曲线。注意一点的是，该彗星轨迹的显示模式 EraseMode 为 none，所以用户不能打印出彗星轨迹（只能得到一个小圆圈），且若用户调整窗口大小，则彗星会消失。

用法　comet3(z) 用向量 z 中的数据显示一个三维彗星

comet3(x,y,z) 显示一个彗星通过数据 x，y，z 确定的三维曲线。

comet3(x,y,z,p) 指定彗星体的长度为：p*length(y)。

例 7-24

```
>>t = -20*pi:pi/50:20*pi;
>>comet3((cos(2*t).^2).*sin(t),(sin(2*t).^2).*cos(t),t);
```

图形的结果为图 7-24。

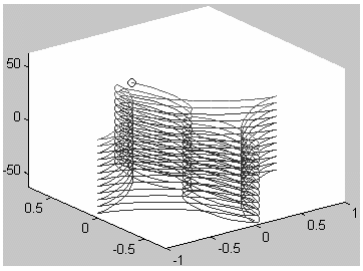

图 7-24

命令 2　**fill3**

功能　用指定的颜色填充三维多边形。阴影类型为平面型和 Gouraud 型。

用法　fill3(X,Y,Z,C) 填充由参数 x，y 和 z 确定多边形。若 x，y 或 z 为矩阵，fill3 生成 n 个多边形，其中 n 为矩阵的列数。在必要的时候，fill3 会自动连接最后一个节点和第一个节点。以便能形成封闭的多边形。参数 c 指定颜色，这儿 c 为引用当前色图的下标向量或矩阵。若 c 为行向量，则 c 的维数必须等于 x 的列数和 y 的列数，若 c 为列向量，则 c 的维数必须等于矩阵 x 的行数和 y 的行数。

fill3(X,Y,Z,ColorSpec) 用指定的颜色 ColorSpec 填充由 x，y 和 z 确定的多边形。

fill3(X1,Y1,Z1,C1,X2,Y2,Z2,C2,…) 对多边形的不同区域用不同的颜色进行填充。

fill3(…,'PropertyName',PropertyValue) 允许用户对特定的 patch 属性进行设置。

h = fill3(…) 返回 patch 图形对象的句柄向量，每一块（patch）对应一个句柄。

运算规则：

1．若 X，Y，Z 为同型的矩阵，fill3 生成 X，Y，Z 中相同位置的元素确定的顶点，每一列生成一个多边形。

2．若只有 X，Y 或 Z 为矩阵，则 fill3 由列向量参数生成可用的同型矩阵。

3．若用户对填充的颜色指定为 ColorSpec，则 fill3 生成阴影类型为 flat-shaded 的多边形，且设置块（patch）的属性 FaceColor 为 RGB 颜色形式的矩阵。

4．若用户用矩阵 C 指定颜色，命令 fill3 通过坐标轴属性 Clim 来调整 C 中的元素，在引用当前色图之前，用于指定颜色坐标轴的参数比例。

5．若参数 C 为一行向量，命令 fill3 生成带平面阴影（flat-shaded）的多边形，同时设置补片对象的面颜色（FaceColor）属性为 flat。向量 c 中的每一元素成为每一补片对象的颜色数据（CData）属性的值。

6．若参数 C 为一矩阵，命令 fill3 生成带内插颜色的多边形，同时设置多边形补片对象的 FaceColor 属性为 interp。命令 fill3 采用对多边形顶点色图的下标指定的颜色采用线性内插算法，同时多边形的颜色采用对顶点颜色用内插算法得到的颜色。矩阵 C 的每一列元素变

成对应补片对象的 Cdata 属性值。

7．若参数 C 为一列向量，命令 fill3 先复制 C 的元素，使之成为所需维数的矩阵，再按上面的方法 6 进行计算。

例 7-25

```
>>X = 10*rand(4);Y=10*rand(4);Z=10*rand(4);
>>C = rand(4);
>>fill3(X,Y,Z,C)
```

图形结果可能为图 7-25。

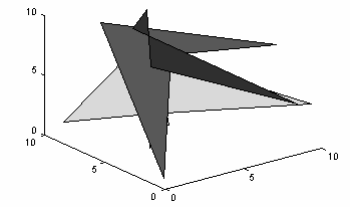

图 7-25

7.2.2　三维图形等高线

命令 1　**contour**

功能　曲面的等高线图

用法　contour(z)　把矩阵 z 中的值作为一个二维函数的值，等高曲线是一个平面的曲线，平面的高度 v 是 Matlab 自动取的；

contour(x,y,z)　(x,y)是平面 z=0 上点的坐标矩阵，z 为相应点的高度值矩阵。效果同上；

contour(z,n)　画出 n 条等高线；

contour(x,y,z,n)　画出 n 条等高线；

contour(z,v)　在指定的高度 v 上画出等高线；

contour(x,y,z,v)　同上；

[c,h] = contour(…)　返回如同 contourc 命令描述的等高矩阵 c 和线句柄或块句柄列向量 h，这些可作为 clabel 命令的输入参量，每条线对应一个句柄，句柄中的 userdata 属性包含每条等高线的高度值；

contour(…,'linespec')　因为等高线是以当前的色图中的颜色画的，且是作为块对象处理的，即等高线是一般的线条，我们可象画普通线条一样，可以指定等高线的颜色或者线形。

例 7-26

```
>>contour(peaks(40))
```

图形结果为图 7-26。

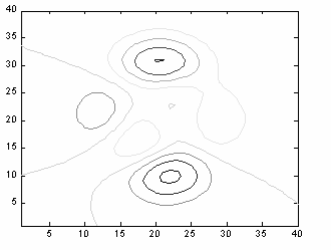

图 7-26

命令 2　**clabel**

功能　在二维等高线图中添加高度标签。在下列形式中，若有 h 出现，则会对标签进行恰当的旋转，否则标签会竖直放置，且在恰当的位置显示个一个“ + ”号。

用法　clabel(C,h)　把标签旋转到恰当的角度，再插入到等高线中。只有等高线之间有足够的空间时才加入，当然这决定于等高线的尺度。

clabel(C,h,v)　在指定的高度 v 上显示标签 h，当然要对标签做恰当的处理。

clabel(C,h,'manual')　手动设置标签。用户用鼠标左键或空格键在最接近指定的位置上放置标签，用键盘上的回车键结束该操作。当然会对标签做恰当的处理。

clabel(C)　在从命令 contour 生成的等高线结构 c 的位置上添加标签。此时标签的放置的位置是随机的。

clabel(C,v) 在给定的位置 v 上显示标签

clabel(C,'manual') 允许用户通过鼠标来给等高线贴标签

例 7-27

```
>>[x,y] = meshgrid(-2:.2:2);
>>z = x.*y.*exp(-x.^2-y.^2);
>>[C,h] = contour(x,y,z);
>>clabel(C,h);
```

图形结果为图 7-27。

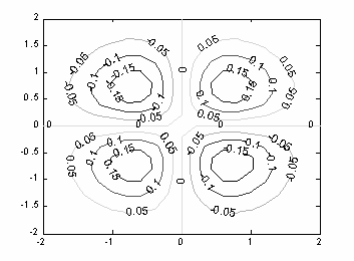

图 7-27

命令 3 **contourc**

功能 低级等高线图形计算命令。该命令计算等高线矩阵 c，该矩阵可用于命令 contour，contour3 和 contourf 等。矩阵 z 中的数值确定平面上的等高线高度值，等高线的计算结果用由矩阵 z 维数决定的间隔的宽度。

用法 C = contourc(Z) 从矩阵 z 中计算等高矩阵，其中 z 的维数至少为 2*2 阶，等高线为矩阵 z 中数值相等的单元。等高线的数目和相应的高度值是自动选择的。

C = contourc(Z,n) 在矩阵 z 中计算出 n 个高度的等高线。

C = contourc(Z,v) 在矩阵 z 中计算出给定高度向量 v 上计算等高线，当然向量 v 的维数决定了等高线的数目。若只要计算一条高度为 a 的等高线，输入：contourc(Z,[a,a])；

C = contourc(x,y,Z) 在矩阵 z 中，参量 x，y 确定的坐标轴范围内计算等高线；

C = contourc(x,y,Z,n) 从矩阵 Z 中，参量 x 与 y 确定的坐标范围内画出 n 条等高线；

C = contourc(x,y,Z,v) 从矩阵 Z 中，参量 x 与 y 确定的坐标范围内，画在 v 指定的高度上指定的等高线。

命令 4 **contour3**

功能 三维空间等高线图。该命令生成一个定义在矩形格栅上曲面的三维等高线图。

用法 contour3(Z) 画出三维空间角度观看矩阵 z 的等高线图，其中 z 的元素被认为是距离 xy 平面的高度，矩阵 z 至少为 2*2 阶的。等高线的条数与高度是自动选择的。若[m，n]=size（z），则 x 轴的范围为[1：n]，y 轴的范围为[1：m]。

contour3(Z,n) 画出由矩阵 z 确定的 n 条等高线的三维图。

contour3(Z,v) 在参量 v 指定的高度上画出三维等高线，当然等高线条数与向量 v 的维数相同；若想只画一条高度为 h 的等高线，输入：contour3(Z,[h,h])

contour3(X,Y,Z)、contour3(X,Y,Z,n)、contour3(X,Y,Z,v) 用 X 与 Y 定义 x-轴与 y-轴的范围。若 X 为矩阵，则 X(1,:)定义 x-轴的范围；若 Y 为矩阵，则 Y(:,1)定义 y-轴的范围；若 X 与 Y 同时为矩阵，则它们必须同型。不论为哪种使用形式，所起的作用与命令 surf 相同。若 X 或 Y 有不规则的间距，contour3 还是使用规则的间距计算等高线，然后将数据转变给 X 或 Y。

contour3(…,LineSpec) 用参量 LineSpec 指定的线型与颜色画等高线。

[C,h] = contour3(…) 画出图形，同时返回与命令contourc中相同的等高线矩阵C，包含所有图形对象的句柄向量h；除非没有指定LineSpec参数，contour3 将生成patch图形对象，且当前的colormap属性与caxis属性将控制颜色的显示。不

论使用何种形式，该命令都生成line图形对象。

例 7-28

```
>>[X,Y] = meshgrid([-2:.25:2]);
>>Z = X.*exp(-X.^2-Y.^2);
>>contour3(X,Y,Z,30)
```

图形结果为图 7-28。

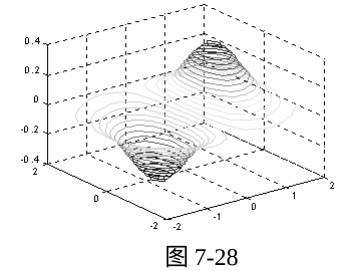

图 7-28

命令 5　**contourf**

功能　填充二维等高线图。即先画出不同等高线，然后相邻的等高线之间用同一颜色进行填充。填充用的颜色决定于当前的色图颜色。

用法　contourf(Z)　矩阵 z 的等高线图，其中 z 理解成距平面的高度。Z 至少为 2*2 阶的。等高线的条数与高度是自动选择的。

contourf(Z,n)　画出矩阵 z 的 n 条高度不同的等高线。

contourf(Z,v)　画出矩阵 z 的、由 v 指定的高度的等高线图。

contourf(X,Y,Z)、contourf(X,Y,Z,n)、contourf(X,Y,Z,v)　画出矩阵 z 的等高线图，其中 X 与 Y 用于指定 x-轴与 y-轴的范围。若 X 与 Y 为矩阵，则必须与 Z 同型。若 X 或 Y 有不规则的间距，contour3 还是使用规则的间距计算等高线，然后将数据转变给 X 或 Y。

[C,h,CF] = contourf(…)画出图形，同时返回与命令 contourc 中相同的等高线矩阵 C，C 也可被命令 clabel 使用；返回包含 patch 图形对象的句柄向量 h；返回一用于填充用的矩阵 CF。

例 7-29

```
>>contourf(peaks(30),20);
>>colormap gray
```

图形结果为图 7-29。

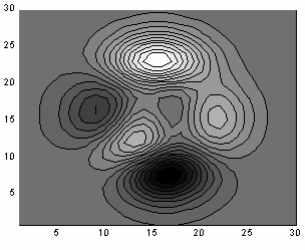

图 7-29

命令 6　**pie3**

功能　三维饼形图

用法　pie3(X)　用 x 中的数据画一个三维饼形图。X 中的每一个元素代表三维饼形图中的一部分。

pie3(X,explode) x 中的某一部分可以从三维饼形图中分离出来。explode 是一个与 x 同型的向量或矩阵，explode 中非零的元素对应 x 中从饼形图中分离出来的分量。

h = pie3(…) 返回一个分量为 patch，surface 和 text 图形句柄对象的向量。即每一块对应一个句柄。

注意：命令 pie3 将 x 的每一个元素在所有元素的总和中所占的比例表达出来。若 x 中的分量和小于 1（则所有元素小于 1），则认为 x 中的值指明三维饼形图的每一部分的大小。

例 7-30

```
>>x = [1 3 0.5 2.5 2]
>>ex = [0 1 0 0 0]
>>pie3(x,ex)
```

图形结果为图 7-30。

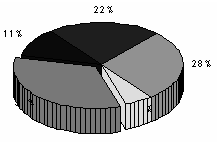

图 7-30

7.2.3 曲面与网格图命令

命令 1 **mesh**

功能 生成由 X，Y 和 Z 指定的网线面，由 C 指定的颜色的三维网格图。网格图是作为视点由 view（3）设定的 surface 图形对象。曲面的颜色与背景颜色相同（当要动画显示不透明曲面时，这时可用命令 hidden 控制），或者当画一个标准的可透视的网线图时，曲面的颜色就没有（命令 shading 控制渲染模式）。当前的色图决定线的颜色。

用法 mesh(X,Y,Z) 画出颜色由 c 指定的三维网格图，所以和曲面的高度相匹配，

1．若 X 与 Y 均为向量，length（X）=n，length（Y）=m，而[m，n]=size（Z），空间中的点 (X(j),Y(I),Z(I,j)) 为所画曲面网线的交点，分别地，X 对应于 z 的列，Y 对应于 z 的行。

2．若 X 与 Y 均为矩阵，则空间中的点 (X(I,j),Y(I,j),Z(I,j))为所画曲面的网线的交点。

mesh(Z) 由[n，m] = size（Z）得，X =1：n 与 Y=1：m，其中 z 为定义在矩形划分区域上的单值函数。

mesh(…,C) 用由矩阵 c 指定的颜色画网线网格图。Matlab 对矩阵 c 中的数据进行线性处理，以便从当前色图中获得有用的颜色。

mesh(…,PropertyName’,PropertyValue, …) 对指定的属性 PropertyName 设置属性值 PropertyValue,可以在同一语句中对多个属性进行设置。

h = mesh(…) 返回 surface 图形对象句柄。

运算规则：

1．数据 X，Y 和 z 的范围，或者是对当前轴的 XLimMode，YLimMode 和 ZLimMode 属性的设置决定坐标轴的范围。命令 aXis 可对这些属性进行设置。

2．参量 c 的范围，或者是对当前轴的 Clim 和 ClimMode 属性的设置（可用命令 caxis 进行设置），决定颜色的刻度化程度。刻度化颜色值作为引用当前色图的下标。

3．网格图显示命令生成由于把 z 的数据值用当前色图表现出来的颜色值。Matlab 会自动用最大值与最小值计算颜色的范围（可用命令 caxis auto 进行设置），最小值用色图中的第一个颜色表现，最大值用色图中的最后一个颜色表现。Matlab 会对数据的中间值执行一个线性变换，使数据能在当前的范围内显示出来。

例 7-31

```
>>[X,Y] = meshgrid(-3:.125:3);
>>Z = peaks(X,Y);
>>mesh(X,Y,Z);
```

图形结果为图 7-31。

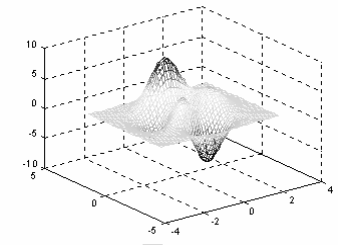

图 7-31

命令 2 **surf**

功能 在矩形区域内显示三维带阴影曲面图。

用法 surf(Z) 生成一个由矩阵 z 确定的三维带阴影的曲面图，其中 [m，n] = size（Z），而 X = 1：n，Y = 1：m。高度 z 为定义在一个几何矩形区域内的单值函数，z 同时指定曲面高度数据的颜色，所以颜色对于曲面高度是恰当的。

surf(X,Y,Z) 数据 z 同时为曲面高度，也是颜色数据。X 和 Y 为定义 X 坐标轴和 Y 坐标轴的曲面数据。若 X 与 Y 均为向量，length（X）=n，length（Y）=m，而[m,n]=size（Z），在这种情况下，空间曲面上的节点为（X(I),Y(j),Z(I,j)）。

surf(X,Y,Z,C) 用指定的颜色 c 画出三维网格图。Matlab 会自动对矩阵 c 中的数据进行线性变换，以获得当前色图中可用的颜色。

surf(…,'PropertyName',PropertyValue) 对指定的属性 PropertyName 设置为属性值 PropertyValue

h = surf(…) 返回一个 surface 图形对象句柄给变量 h。

运算规则：

1．严格地讲，一个参数曲面是由两个独立的变量 I、j 来定义的，它们在一个矩形区域上连续变化。例如，a<=I<=b,c<=j<=d，三个变量 X，Y，Z 确定了曲面。曲面颜色由第四参数矩阵 C 确定。

2．矩形定义域上的点有如下关系：

```
          A(I-1,j)
             |
B(I,j-1) ---- C(I,j) ---- D(I,j+1)
             |
          E(I+1,j)
```

这个矩形坐标方格对应于曲面上的有四条边的块，在空间的点的坐标为[X(☺,Y(☺,Z)，每个矩形内部的点根据矩形的下标和相邻的四个点连接；曲面上的点只有相邻的三个点，曲面上四个角上的点只有两个相邻点，上面这些定义了一个四边形的网格图。

3．曲面颜色可以有两种方法来指定：指定每个节点的颜色或者是每一块的中心点颜色。在这种一般的设置中，曲面不一定为变量 X 和 Y 的单值函数，进一步而言，有四边的曲面块不一定为平面的，而可以用极坐标，柱面坐标和球面坐标定义曲面。

4．命令 shading 设置阴影模式。若模式为 interp，C 必须与 X，Y，Z 同型；它指定了每个节点的颜色，曲面块内的颜色由附近几个点的颜色用双线性函数计算出来的。若模式为 facted（缺省模式）或 flat，c(I,j)指定曲面块中的颜色：

```
A(I,j)---------- B (I,j+1)
   |      C(I,j)       |
C(I+1,j) --------- D(I+1,j)
```

在这种情形下，C 可以与 X，Y，和 Z 同型，且它的最后一行和最后一列将被忽略，换句话说，就是 C 的行数和列数可以比 X，Y，Z 少 1。

5．命令 surf 将指定图形视角为 view（3）。

6．数据 X，Y，Z 的范围或者通过对坐标轴的属性 XlimMode，YlimMode 和 ZlimMode 的当前设置（可以通过命令 axis 来设置），将决定坐标轴的标签。

7．参数 C 的范围或者通过对坐标轴的属性 Clim 和 ClimMode 的设置（可以通过命令 caxis 来设置），将决定颜色刻度化。刻度化的颜色值将作为引用当前色图的下标。

例 7-32

```
>>[X,Y,Z] = peaks(30);
>>surf(X,Y,Z)
>>colormap hsv
```

结果图形为图 7-32。

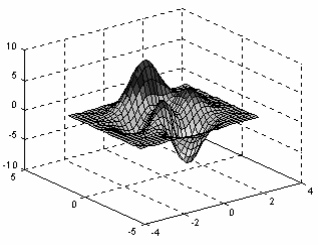

图 7-32

命令 3　**surfc**

功能　在矩形区域内显示三维带阴影曲面图，且在曲面下面画出等高线。

用法　surfc(Z)、surfc(X,Y,Z)、 surfc(X,Y,Z,C)、
surfc(…,'PropertyName',PropertyValue)、
surfc(…)、 h = surfc(…)

上面各个使用形式的曲面效果与命令 surf 的相同，只不过是在曲面下面增加了曲面的等高线而已。

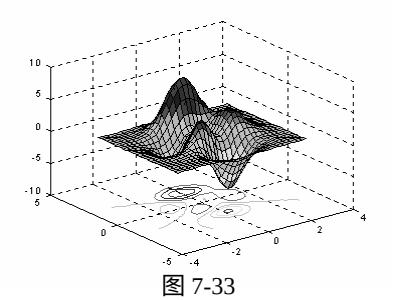
图 7-33

例 7-33

```
>>[X,Y,Z] = peaks(30);
>>surfc(X,Y,Z)
>>colormap hsv
```

图形结果为图 7-33。

命令 4　**surfl**

功能　画带光照模式的三维曲面图。该命令显示一个带阴影的曲面，结合了周围的，散射的和镜面反射的光照模式。想获得较平滑的颜色过度，要使用有线性强度变化的色图（如：gray，copper，bone，pink 等）。参数 X，Y，Z 确定的点定义了参数曲面的“里面”和“外面”，若用户想曲面的“里面”有光照模式，只要使用：

surfl(X',Y',Z')

用法　surfl(Z) 以向量 z 的元素生成一个三维的带阴影的曲面，其中阴影模式中的光源的方位、光照系数为缺省值（见下面）。

surfl(X,Y,Z) 以矩阵 X，Y，Z 生成的一个三维的带阴影的曲面，其中阴影模式中的光源的方位、光照系数为缺省值（见下面）。

surfl(…,'light') 用一个 matlab 光照对象（light object）生成一个带颜色、带光照的曲面，这与用缺省光照模式产生的效果不同。

surfl(…,'cdata') 改变曲面颜色数据（color data），使曲面成为可反光的曲面。

surfl(…,s) 指定光源与曲面之间的方位 s，其中 s 为一个二维向量[azimuth，elevation]，或者三维向量[sx，sy，sz]。缺省光源方位为从当前视角开始，逆时针 45°（度）。

surfl(X,Y,Z,s,k) 指定反射常系数 k，其中 k 为一个定义环境光（ambient light）系数（0<=ka<=1）、漫反射(diffuse reflection)系数（0〈=kb〈=1〉、镜面反射(specular reflection)系数（0〈=ks〈=1〉与镜面反射亮度（以相素为单位）等的四维向量[ka，kd，ks，shine]，缺省值为 k=[0.55 0.6 0.4 10]。

h = surfl(…) 返回一个曲面图形句柄向量 h。

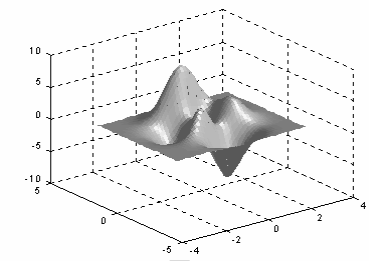
图 7-34

例 7-34

```
>>[X,Y] = meshgrid(-3:1/8:3);
>>Z = peaks(X,Y);
>>surfl(X,Y,Z);
>>shading interp
>>colormap(gray);
```

图形结果为图 7-34。

命令 5　**waterfall**

功能　瀑布图

用法　waterfall(X,Y,Z) 用所给参数 X、Y 与 Z 的数据画一“瀑布”效果图。若 X 与 Y 都是向量，则 X 与 Z 的列相对应，Y 与 Z 的行相对应，即 length(X)=Z 的列数，length(Y)=Z 的行数。参数 X 与 Y 定义了 x-轴与 y-轴，Z 定义了 z-轴的高度，Z 同时确定了颜色，所以颜色能恰当地反映曲面的高度。若想研究数据的列，可以输入：waterfall(Z')或 waterfall(X',Y',Z')

waterfall(Z) 画出一瀑布图，其中缺省地有：X=1:Z 的行数，Y=1:Z 的行数，且 Z 同时确定颜色，所以颜色能恰当地反映曲面高度。

waterfall(…,C) 用比例化的颜色值从当前色图中获得颜色，参量 C 决定颜色的比例，为此，必须与 Z 同型。系统使用一线性变换，从当前色图中获得颜色。

h = waterfall(…) 返回 patch 图形对象的句柄 h，可用于画出图形。

例 7-35

```
>>[X,Y,Z] = peaks(30);
>>waterfall(X,Y,Z)
```

图形结果为图 7-35。

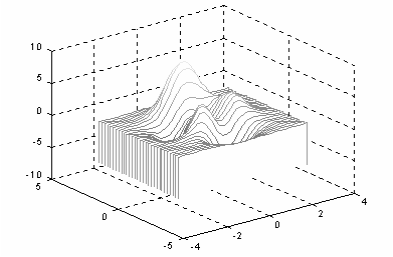

图 7-35

命令 6　**cylinder**

功能　生成圆柱图形。该命令生成一单位圆柱体的 x-，y-，z-轴的坐标值。用户可以用命令 surf 或命令 mesh 画出圆柱形对象，或者用没有输出参量的形式而立即画出图形。

用法　[X,Y,Z] = cylinder 返回一半径为 1、高度为 1 的圆柱体的 x-，y-，z-轴的坐标值，圆柱体的圆周有 20 个距离相同的点。

[X,Y,Z] = cylinder® 返回一半径为 r、高度为 1 的圆柱体的 x-，y-，z-轴的坐标值，圆柱体的圆周有 20 个距离相同的点。

[X,Y,Z] = cylinder(r,n) 返回一半径为 r、高度为 1 的圆柱体的 x-，y-，z-轴的坐标值，圆柱体的圆周有指定的 n 个距离相同的点。

cylinder(…) 没有任何的输出参量，直接画出圆柱体。

例 7-36

```
>>t = 0:pi/10:2*pi;
>>[X,Y,Z] = cylinder(2+(cos(t)).^2);
>>surf(X,Y,Z); axis square
```

图形结果为图 7-36。

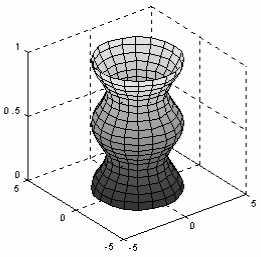

图 7-36

命令 7　**sphere**

功能　生成球体

用法　sphere 生成三维直角坐标系中的单位球体。该单位球体由 20*20 个面。

sphere(n) 在当前坐标系中画出有 n*n 个面的球体

[X,Y,Z] = sphere(n) 返回三个阶数为(n+1)*(n+1)的，直角坐标系中的坐标矩阵。该命令没有画图，只是返回矩阵。用户可以用命令 surf（X，Y，Z）或 mesh

（X，Y，Z）画出球体。

例 7-37

```
>>[X,Y,Z]=sphere;
>>mesh(X,Y,Z)
>>hidden off
```

图形结果为图 7-37。

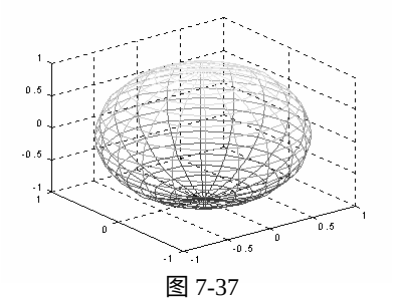

图 7-37

7.2.4 三维数据的其他表现形式命令

命令 1 **pcolor**

功能 伪彩色图。该图为一矩形单元的、由参数 c 定义了颜色的阵列，系统通过 c 中的每相邻的四点定义的曲面补片而生成一伪彩色图。是从上面向下观看的“平面”曲面图。若用户使用命令 shading faceted 或 shading flat，则每一单元的固定颜色是与之相连的角的颜色有关的。所以，C(i,j)定义了单元的地 i 行与地 j 列的颜色。C 中的最后一行与最后一列都没有用上。若用户使用命令 shading interp，则每一单元的颜色是对它的四个顶点的颜色进行一双线性插值后的颜色，这时 c 的所有元素都参加了运算。

用法 pcolor(C) 画一伪彩色图。C 中的元素都线性地映射于当前色图下标。从 C 映射到当前的色图是由命令 colormap 和 caxis 定义的。

pcolor(X,Y,C) 在参数 x 和 y 指定的位置上画一由 C 确定的为彩色图。该图为一逻辑上为矩形、带二维格栅的、顶点在[X(i,j),Y(i,j)]的图形（若 X 和 Y 为矩阵时）。参量 X 与 Y 为指定格栅线的向量或矩阵。若 X 与 Y 为向量，则 X 对应于 C 的列，而 y 对应于 C 的行；若 X 与 Y 同为矩阵，则必须为同型矩阵。该命令等价于命令：surf(X,Y,0,C)，观察角度为：view([0,90])。

h = pcolor(…) 返回一 surface 图形对象句柄于 h

例 7-38

```
>>pcolor(magic(20))
>>colormap(gray(2))
>>axis ij;axis square
```

图形结果为图 7-38。

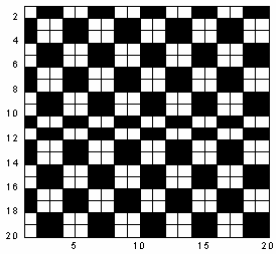

图 7-38

命令 2 **quiver**

功能 矢量图或速度图

用法 quiver(U,V) 在范围为 x =1:n 和 y =1:m 的坐标系中显示由 U 和 V 定义的向量，而[m,n]=size(U)=size(V)，这种形式是在一个几何矩形中画出 U 和 V 的，quiver 命令本身会自动地画出这些向量，使之不会重叠。

quiver(X,Y,U,V) 由向量 X 和 Y 中的分量的任意组合而成的向量与。若 X 与 Y 都是向量 length（X）=n，而 length（Y）=m，而[m，n]=size（U）=size（V），向量 X 对应于矩阵 U、V 的列向量，而向量 Y 对应于矩阵 U、V 的行向量。

quiver(…,scale) 自动对向量的长度进行处理。使之不会重叠，当然可以对 scale 进行取值，若 scale=2，则向量长度伸长 2 倍，若 scale=0，则如实画出向量图。

quiver(…,LineSpec) 可以指定画矢量图用的线型，符号，颜色，quiver 命令会在

原来的向量图上画出记号。

quiver(…,LineSpec,'filled') 对用 LineSpec 指定的记号进行填充

h = quiver(…) 返回每个向量图的句柄

例 7-39

```
>>[z,x,y]=peaks(30);
>>[Dx,Dy]=gradient(z,0.1,0.1);
>>quiver(x,y,Dx,Dy)
```

图形结果为图 7-39。

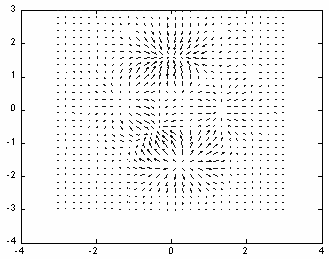

图 7-39

命令 3　**slice**

功能　立体切片图。该命令显示通过立体图形的矩形切片图。

用法　slice(X,Y,Z,V,sx,sy,sz) 显示三元函数 V=V(X,Y,Z)确定的超立体形在 x-轴、y-轴与 z-轴方向上的若干点（对应若干平面。即若函数 V=V(X,Y,Z)中有一变量如 X 取一定值 X0，则函数 V=V(X0,Y,Z)变成一立体曲面（只不过是将该曲面通过颜色表示高度 V,从而显示于一平面而已。）的切片图，各点的坐标由参量向量 sx、sy 与 sz 指定。参量 X、参量 Y 与参量 Z 为三维数组，用于指定立方体 V 的坐标。参量 X、Y 与 Z 必须有单调的、正交的间隔（如同用命令 meshgrid 生成的一样）。在每一点上的颜色由对超立体 V 的三维内插值确定。

slice(V,sx,sy,sz) 显示三元函数 V=V(X,Y,Z)确定的超立体形在 x-轴、y-轴与 z-轴方向上的若干点（对应若干平面）的切片图，各点的坐标由数量向量 sx、sy 与 sz 指定。其中 V 为三维数组（阶数为 m*n*p），缺省地有：X = 1:m、Y = 1:n、Z = 1:p。

slice(V,XI,YI,ZI) 显示参量矩阵 XI、YI 与 ZI 确定的、超立体图形的切面图。参量 XI、YI 与 ZI 定义了一曲面，同时会在曲面的点上计算超立体 V 的值。参量 XI、YI 与 ZI 必须为同型矩阵。

slice(X,Y,Z,V,XI,YI,ZI) 沿着由矩阵 XI、YI 与 ZI 定义的曲面画穿过超立体图形 V 的切片。

slice(…,'method') 指定内插值的方法。‘ method ’为如下方法之一：‘ linear ’、‘ cubic ’、‘ nearest ’：

‘ linear ’——指定使用三次线性内插值法（该状态为缺省的）；

‘ cubic ’——　指定使用三次立方内插值法；

‘ nearest ’——指定使用最近点内插值法。

h = slice(…) 返回一曲面图形对象的句柄向量 h。

命令 4　**axis**

功能　坐标轴的刻度与外在显示

用法　axis([xmin xmax ymin ymax]) 设置当前坐标轴的 x-轴与 y-轴的范围。

axis([xmin xmax ymin ymax zmin zmax cmin cmax]) 设置当前坐标轴的 x-轴、y-轴与 z-轴的范围，当前颜色刻度范围。该命令也同时设置当前坐标轴的属性 Xlim、Ylim 与 Zlim 为所给参数列表中的最大值和最小值。另外，坐标轴属

性 XlimMode、YlimMode 与 ZlimMode 设置为‘ manual ’。

v = axis 返回一包含 x-轴、y-轴与 z-轴的刻度因子的行向量，其中 v 为一四维或六维向量，这取决于当前坐标为二维还是三维的。返回的值包含当前坐标轴的 XLim、Ylim 与 Zlim 属性值。

axis auto 设置系统到它的缺省动作——自动计算当前轴的范围，这取决于输入参量 x，y 与 z 的数据中的最大值与最小值。同时将当前坐标轴的属性 XlimMode、YlimMode 与 ZlimMode 设置为‘ auto ’用户可以指定对某一坐标轴进行自动操作。例如：

axis ’auto x’ 将自动计算 x-轴的范围；

axis ’auto yz’ 将自动计算 y-轴与 z-轴的范围。

axis manual 、axis(axis) 把坐标固定在当前的范围，这样，若保持状态(hold)为 on，后面的图形仍用相同界限。该命令设置了属性 XLimMode、属性 YLimMode 与属性 ZlimMode 为 manual。

axis tight 把坐标轴的范围定为数据的范围，即坐标轴中没有多余的部分。

axis fill 该命令用于将坐标轴的取值范围分别设置为绘图所用数据在相应方向上的最大、最小值。

axis ij 使用矩阵坐标系：坐标原点在左上角、横坐标（j-轴）的值从左到右增加，纵坐标（i-轴）的值从上到下增加。

axis xy 使用笛卡儿坐标系（缺省）：坐标原点在左下角、横坐标（x-轴）的值从左到右增加，纵坐标（y-轴）的值从下到上增加。

axis equal 设置坐标轴的纵横比，使在每个方向的数据单位都相同。其中 x-轴、y-轴与 z-轴将根据所给数据在各个方向的数据单位自动调整其纵横比。

axis image 效果与命令 axis equal 相同，只是图形区域刚好紧紧包围图象数据。

axis square 设置当前图形为正方形（或立方体形），系统将调整 x-轴、y-轴与 z-轴，使它们有相同的长度，同时相应地自动调整数据单位之间的增加量。

axis normal 自动调整坐标轴的纵横比，还有用于填充图形区域的、显示于坐标轴上的数据单位的纵横比。

表 7-7 显示由上面三个命令设置的坐标轴属性。

表 7-7

命令 / 坐标轴属性	axis equal	axis normal	axis square	axis tightequal
DataAspectRatioMode	[1 1 1]	没有设置	没有设置	[1 1 1]
PlotBoxAspectRatio	manual	auto	auto	Manual
PlotBoxAspectRatioMode	[3 4 4]	没有设置	[1 1 1]	Auto
Stretch-to-fill	禁止	可行	禁止	禁止

axis vis3d 该命令将冻结坐标系此时的状态，以便进行旋转。

axis off 关闭所用坐标轴上的标记、格栅和单位标记。但保留由 text 和 gtext 设置的对象。

axis on 显示坐标轴上的标记、单位和格栅。

[mode,visibility,direction] = axis('state') 返回表明当前坐标轴的设置属性的三个字

符串，见表 7-8。

表 7-8

输出参量	返回字符串	说明
Mode	‘’auto’或’manual’	若 XLimMode、YlimMode 与 ZlimMode 都设置为 auto，则 mode 为 auto；若 XLimMode、YlimMode 或者 ZlimMode 都设置为 manual，则 mode 为 manual
Visibility	‘’on’或’off’	
Direction	‘’xy’或’ij’	

例 7-40

```
>>x = 0:.025:pi/2;
>>plot(x,exp(x).*sin(2*x),'-m<')
>>axis([0    pi/2    0    5])
```

图形结果为图 7-40。

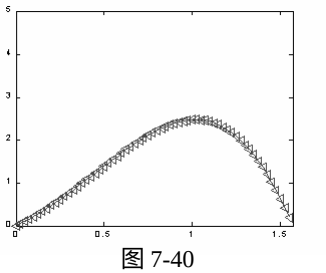

图 7-40

命令 5　**hidden**

功能　在一网格图中显示隐含线条。隐含线条的显示，实际上是显示那些从观察角度观看没有被其他物体遮住的线条。

用法　hidden on 对当前图形打开隐含线条的显示状态，使网格图后面的线条被前面的线条遮住。设置曲面图形对象的属性 FaceColor 为坐标轴背景颜色。这是系统的缺省操作。

hidden off 对当前图形关闭隐含线条的显示

hidden 在两种状态 on 与 off 之间切换

例 7-41

```
>>mesh(peaks)
>>hidden off
```

图形结果为图 7-41。

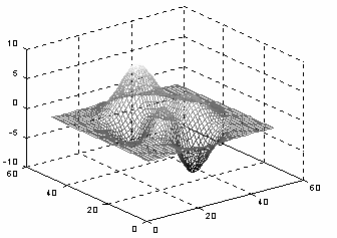

图 7-41

命令 6　**shading**

功能　设置颜色色调属性。该命令控制曲面与补片等的图形对象的颜色色调。同时设置当前坐标轴中的所有曲面与补片图形对象的属性 EdgeColor 与 FaceColor。命令 shading 设置恰当的属性值，这取决于曲面或补片对象是表现网格图或实曲面。

用法　shading flat 使网格图上的每一线段与每一小面有一相同颜色，该颜色由线段的末端的端点颜色确定；或由小面的、有小型的下标或索引的四个角的颜色确定。

shading faceted 带重叠的黑色网格线的平面色调模式。这是缺省的色调模式。

shading interp 在每一线段与曲面上显示不同的颜色，该颜色为通过在每一线段两边的、或者为不同小曲面之间的色图的索引或真颜色进行内插值得到的颜色。

例 7-42

```
>>sphere(16)
>>axis square
>>shading flat
>>title('Flat Shading')
```

图形结果为图 7-42。

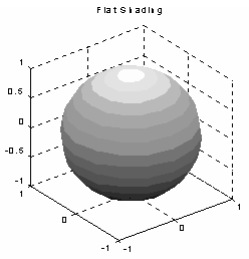

图 7-42

命令 7　**caxis**

功能　颜色坐标轴刻度。命令 caxis 控制着对应色图的数据值的

映射图。它影响下面对象之一的、用带索引的颜色数据（CData）与颜色数据映射（CDataMapping）控制的刻度的图形对象 surface、patches 与 images；它没有影响带用颜色数据（CData）或颜色数据映射（CDataMapping）直接设置的颜色的图形对象 surface、images 或 patches。该命令还改变坐标轴图形对象的属性 Clim 与 ClimMode。

用法　caxis([cmin cmax]) 用指定的最大值与最小值设置颜色范围。数据值中小于 cmin 或大于 cmax 的，将分别地映射于 cmin 与 cmax；处于 cmin 与 cmax 之间的数据将线性地映射于当前色图。

caxis auto 让系统自动地计算数据的最大值与最小值对应的颜色范围。这是系统的缺省动作。数据中的正无穷大（Inf）对应于最大颜色值；负无穷大（-Inf）对应于最小颜色值；带颜色值设置为 NaN 的面或者边界将不显示。

caxis manual、caxis(caxis) 冻结当前颜色坐标轴的刻度范围。这样，当 hold 设置为 on 时，可使后面的图形命令使用相同的颜色范围。

v = caxis 返回一包含当前正在使用的颜色范围的二维向量 v=[cmin cmax]。

caxis(axes_handle,…) 使由参量 axis_handle 指定的坐标轴，而非当前坐标轴。

颜色坐标轴刻度工作原理：

使用带索引的颜色数据(Cdata)与颜色数据映射(CdataMapping)的图形对象 surface、patch 与 image 将设置成刻度化的，在每次图形渲染时，将映射颜色数据值为当前图形的颜色。当颜色数据值等于或小于 cmin 时，将它映射为当前色图中的第一个颜色；当颜色数据值等于或大于 cmax 时，将它映射为当前色图中的最后一个颜色；对于处于 cmin 与 cmax 之间的颜色数据(例如 c)，系统将执行下列线性转换，以获得对应当前色图（它的长度为 m）中的颜色的索引（当前色图的行指标 index）：

index = fix((C-min)/(cmax-cmin)*m)+1

例 7-43

```
>>[X,Y,Z] = sphere;
>>C = Z;surf(X,Y,Z,C)
>>caxis([-1 3])
```

图形结果为图 7-43。

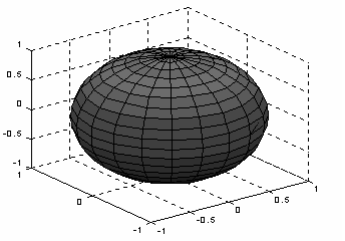

图 7-43

命令 8　**view**

功能　指定立体图形的观察点。观察者（观察点）的位置决定了坐标轴的方向。用户可以用方位角（azimuth）和仰角（elevation）一起，或者用空间中的一点来确定观察点的位置。

用法　view(az,el)、view([az,el]) 给三维空间图形设置观察点的方位角。方位角 az 与仰角 el 为这两个旋转角度：做一通过视点与 z-轴的平面，与 xy 平面有一交线，该交线与 y-轴的反方向的、按逆时针方向（从 z-轴的方向观察）计算的、单位为度的夹角，就是观察点的方位角 az。若角度为负值，则按顺时针方向计算；在通过视点与 z-轴的平面上，用一直线连接视点与坐标原点，该直线与 xy 平面的夹角就是观察点的仰角 el。若仰角为负值，则观察点转移到曲面下面。

view([x,y,z]) 在笛卡儿坐标系中于点（x,y,z）设置视点。注意：输入参量只能是方括号的向量形式，而非数学中的点的形式。

view(2)　设置缺省的二维形式视点。其中 az=0，el=90，即从 z-轴上方观看。

view(3)　设置缺省的三维形式视点。其中 az=-37.5，el=30。

view(T)　根据转换矩阵 T 设置视点。其中 T 为 4*4 阶的矩阵，如同用命令 viewmtx 生成的透视转换矩阵一样。

[az,el] = view　返回当前的方位角 az 与仰角 el。

T = view　返回当前的 4*4 阶的转换矩阵 T。

例 7-44

```
>>peaks;
>>az = 0;el = 90;
>>view(az, el)
```

图形结果为图 7-44。

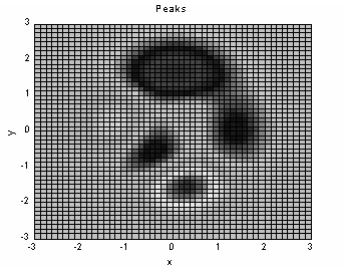

图 7-44

命令 9　**viewmtx**

功能　视点转换矩阵。计算一个 4*4 阶的正交的或透视的转换矩阵，该矩阵将一四维的、齐次的向量转换到一个二维的视平面上（如计算机平面上）。

用法　T = viewmtx(az,el)　返回一与视点的方位角 az 与仰角 el（单位都为度）对应的正交矩阵，并没有改变当前视点。

T = viewmtx(az,el,phi)　返回一透视的转换矩阵，其中参量 phi 是单位为度的透视角度，为标准化立方体（单位为度）的对像视角角度与透视扭曲程度。

表 7-9

Phi 的值	说明
0 度	正交投影
10 度	类似以远距离投影
25 度	类似以普通投影
60 度	类似以广角投影

用户可以通过使用返回的矩阵，用命令 view(T)改变视点的位置。该 4*4 阶的矩阵将变换四维的、同次的向量成形式为(x,y,z,w)的非标准化的向量，其中 w 不等于 1。正交化的 x-元素与 y-元素组成的向量(x/w,y/w,z/w,1)为我们所需的二维向量。（注：一四维同次向量为在对应的三维向量后面增加一个 1。例如：[x,y,z,1]为对应于三维空间中的点[x,y,z]的四维向量。）

T = viewmtx(az,el,phi,xc)　返回以在标准化的图形立方体中的点 xc 为目标点的透视矩阵（就像相机正对着点 xc 一样），目标点 xc 为视角的中心点。用户可以用一三维向量 xc=[xc,yc,zc]指定该中心点，每一分量都在区间[0，1]上。缺省值为 xc=[0 0 0]。

命令 10　**surfnorm**

功能　计算与显示三维曲面的法线。该命令计算用户命令 surf 中的曲面法线。

用法　surfnorm(Z)、surfnorm(X,Y,Z)　画出一曲面与它的法线图。其中矩阵 Z 用于指定曲面的高度值；X 与 Y 为向量或矩阵，用于定义曲面的 x 与 y 部分。

[Nx,Ny,Nz] = surform(…)　返回组成曲面的法线在三个坐标轴上的投影分量 Nx,Ny 与 Nz。

例 7-45

```
>>[x,y,z] = cylinder(1:10);
>>surfnorm(y,x,z)
```

```
>>axis([-12 12 -12 12 -0.1 1])
```

图形结果为图 7-45。

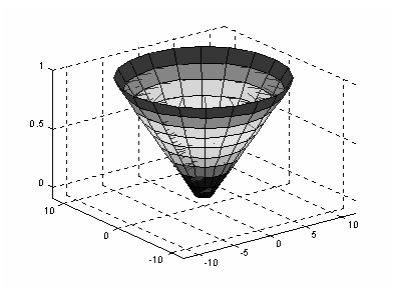

图 7-45　曲面法线图

7.3　通用图形函数命令

7.3.1　图形对象句柄命令

命令 1　**figure**

功能　创建一个新的图形对象。图形对象为在屏幕上单独的窗口，在窗口中可以输出图形。

用法　figure 用缺省的属性值创建一个新的图形对象。

figure('PropertyName',PropertyValue,…) 对指定的属性 PropertyName 用指定的属性值 PropertyValue（属性名与属性值成对出现）创建一个新的图形窗口，对于那些没有指定的属性，则用缺省值。属性名与有效的属性值见下表。

figure(h) 1. 若 h 为一个已经存在的图形的句柄，则 figure(h)使由 h 标记的图形成为当前图形，使它可见，且在屏幕上把它显示到所有图形之前。当前图形为图像输出的地方。

2. 若 h 不是已经存在图形的句柄，但是为一整数，则该命令生成一图形窗口，同时把该窗口的句柄赋值为 h；若 h 不是一图形窗口的句柄，也不是一整数，则返回一错误信息。

h = figure(…) 返回图形窗口对象的句柄给 h。

表 7-10

属性名	属性说明	有效属性值
窗口位置		
Position	图形窗口的位置与大小	有效值：四维向量[left,bottom, width,height] 缺省值：决定于显示
Units	用于解释属性 Position 的单位	有效值：inches(英寸) centimeters(厘米) normalized(标准化单位，认为窗口为一长宽都是 1) points(点) pixels(像素)

		characters(字符) 缺省值：pixels
指定类型与外在显示		
Color	窗口的背景颜色	有效值：ColorSpec（有效的颜色 参数） 缺省值：取决于颜色表(参见命令 colordef)
Menubar	转换图形窗口菜单条的“开”与“关”	有效值：none、figure 缺省值：figure
Name	显示图形窗口的标题	有效值：任意字符串 缺省值：‘ ’（空字符串）
NumberTitle	标题栏中是否显示’Figure No. n’,其中 n 为图形窗口的编号	有效值：on、off 缺省值：on
Resize	指定图形窗口是否可以通过鼠标改变大小	有效值：on、off 缺省值：on
SelectionHighlight	当图形窗口被选中时，是否突出显示	有效值：on、off 缺省值：on
Visible	确定图形窗口是否可见	有效值：on、off 缺省值：on
WindowStyle	指定窗口为标准窗口还是典型窗口	有效值：normal（标准窗口）、 modal（典型窗口） 缺省值：normal
控制色图		
Colormap	图形窗口的色图	有效值：m*3 阶的 RGB 颜色矩阵 缺省值：jet 色图
Dithermap	用于真颜色数据以伪颜色显示的色图	有效值：m*3 阶的 RGB 颜色矩阵 缺省值：有所有颜色的色图
DithermapMode	是否使用系统生成的抖动色图	有效值：auto、manual 缺省值：manual
FixedColors	不是从色图中获得的颜色	有效值：m*3 阶的 RGB 颜色矩阵 缺省值：无（只读模式）
MinColormap	系统颜色表中能使用的最少颜色数	有效值：任一标量 缺省值：64
ShareColors	允许 MATLAB 共享系统颜色表中的颜色	有效值：on、off 缺省值：on
指定透明度		
Alphamap	图形窗口的α色图，用于设定透明度。	有效值：m*1 维向量，每一分量在[0 1]之间 缺省值：64*1 维向量
指定渲染模式		
BackingStore	打开或关闭屏幕像素缓冲区	有效值：on、off 缺省值：on
DoubleBuffer	对于简单的动画渲染是否使用快速缓冲	有效值：on、off 缺省值：off
Renderer	用于屏幕和图片的渲染模式	有效值：painters、zbuffer、OpenGL 缺省值：系统自动选择
关于图形窗口的一般信息		
Children	显示于图形窗口中的任意对象句柄	有效值：句柄向量
FileName	命令 guide 使用的文件名	有效值：字符串
Parent	图形窗口的父对象：根屏幕	有效值：总是 0（即根屏幕）
Selected	是否显示窗口的“选中”状态	有效值：on、off 缺省值：on
Tag	用户指定的图形窗口	有效值：任意字符串

	标签	缺省值：' '（空字符串）
Type	图形对象的类型（只读类型）	有效值：'figure'
UserData	用户指定的数据	有效值：任一矩阵 缺省值：[]（空矩阵）
RendererMode	缺省的或用户指定的渲染程序	有效值：auto、manual 缺省值：auto
关于当前状态的信息		
CurrentAxes	在图形窗口中的当前坐标轴的句柄	有效值：坐标轴句柄
CurrentCharacter	在图形窗口中最后一个输入的字符	有效值：单个字符
CurrentOject	图形窗口中的当前对象的句柄	有效值：图形对象句柄
CurrentPoint	图形窗口中最后单击的按钮的位置	有效值：二维向量[x-coord， y-coord]
SelectionType	鼠标选取类型	有效值：normal、extended、alt、open
回调程序的执行		
BusyAction	指定如何处理中断调用程序	有效值：cancel、queue 缺省值：queue
ButtonDownFcn	当在窗口中空闲点按下鼠标按钮时，执行的回调程序	有效值：字符串 缺省值：' '（空字符串）
CloseRequestFcn	当执行命令关闭时，定义一回调程序	有效值：字符串 缺省值：closereq
CreateFcn	当打开一图形窗口时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
DeleteFcn	当删除一图形窗口时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
Interruptible	定义一回调程序是否可中断	有效值：on、off 缺省值：on（可以中断）
KeyPressFcn	当在图形窗口中按下一键时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
ResizeFcn	当图形窗口改变大小时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
UIContextMenu	定义与图形窗口相关的菜单	有效值：属性 UIContrextmenu 的句柄
WindowButtonDownFcn	当在图形窗口中按下鼠标时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
WindowButtonMotionFcn	当将鼠标移进图形窗口中时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
WindowButtonUpFcn	当在图形窗口中松开按钮时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
访问对象的控制		
IntegerHandle	指定使用整数或非整数图形句柄	有效值：on、off 缺省值：on（整数句柄）
HandleVisiblity	指定图形窗口句柄是否可见	有效值：on、callback、off 缺省值：on
HitTest	定义图形窗口是否能变成当前对象(参见图	有效值：on、off 缺省值：on

	形窗口属性 CurrentObject)	
NextPlot	在图形窗口中定义如何显示另外的图形	有效值：replacechildren、add、replace 缺省值：add
定义鼠标指针		
Pointer	选取鼠标记号	有效值：crosshair、arrow、topr、watch、topl、botl、botr、circle、cross、fleur、left、right、top、fullcrosshair、bottom 、ibeam、custom 缺省值：arrow
PointerShapeCData	定义鼠标外形的数据	有效值：16*16 阶矩阵 缺省值：将鼠标设置为'custom'且可见
PointerShapeHotSpot	设置鼠标活跃的点	有效值：二维向量[row，column] 缺省值：[1 1]

例 7-46

```
>>scrsz = get(0,'ScreenSize');
>>figure('Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2])
```

执行上面的语句，会在屏幕的左上角生成一没有任何符号的窗口。

命令 2　**line**

功能　生成线（line）对象。命令 line 在当前坐标轴中生成一个线对象。用户可以指定线的颜色，宽度，类型和标记符号等其他特性。

命令 line 有两种形式：

1．自动循环使用颜色和类型。当用户用非正式语法来指定矩阵坐标数据：line(X,Y,Z)，Matlab 将循环使用由坐标轴 ColorOrder 和 LineStyle 指定的颜色顺序和类型顺序。

2．纯粹低级操作。当用户用属性名和属性值调用命令 line：

line(‘XData’,x,’YData’,y,’ZData’,z)

Matlab 将在当前用缺省的颜色（参见命令 colordef 的使用）画出线对象。注意一点的是，用户不能在命令 line 的低级形式中使用矩阵数据。

用法　line(X,Y) 在当前的坐标轴中画出由向量 x 和 y 定义的线条。若 x 与 y 为同型的矩阵，则对于 x，y 的每一列画出一线条。

line(X,Y,Z) 在三维空间中画出由 x，y，z 定义的线条。

line(X,Y,Z,'PropertyName',PropertyValue,…) 画出由参数 x，y，z 确定的线条，其中对指定属性 PropertyName 设置为 PropertyValue，其他没有指定属性用缺省值。属性 LineStyle 和 Marker 参见命令 plot。

line('PropertyName',PropertyValue,…) 对属性用相应的输入参数来设置而画出线条。这是命令 line 的低级使用形式，此时不接受矩阵参数。除了该情形，其他形式都接受矩阵参数。

h = line(…) 返回每一条线的线对象对应的句柄向量。

表 7-11

属性名	说明	有效属性值
定义对象的数据		
Xdata	定义线条的 x-轴坐标参量	有效值：向量或矩阵 缺省值：[0 1]
Ydata	定义线条的 y-轴坐标参量	有效值：向量或矩阵 缺省值：[0 1]

Zdata	定义线条的 z-轴坐标参量	有效值：向量或矩阵 缺省值：[0 1]
定义线型与数据点标记符		
LineStyle	定义线条的类型	有效值：-、--、:、-.、 none 缺省值：-（实线）
LineWidth	定义线条的宽度（以磅为单位）	有效值：一标量 缺省值：0.5 磅
Marker	定义标记数据点的标记符号	有效值：13 种类型之一 缺省值：none
MarkerEdgeColor	定义标记颜色或可填充标记的边界颜色	有效值： auto、none、 ColorSpec 缺省值：auto
MarkerFaceColor	定义封闭形标记的填充颜色	
MarkerSize	定义标记大小	有效值：标量（磅） 缺省值：６（磅）
控制线条的显示		
Clipping	坐标轴矩形区域是否可剪辑	有效值：on、off 缺省值：on
EraseMode	定义显示与擦除线条的方法(对于动画显示)	有效值：normal、none、 xor、background 缺省值：normal
SelectionHighlight	当线条被选中时，是否突出显示	有效值：on、off 缺省值：on
Visible	定义线条是否可见	有效值：on、off 缺省值：on
Color	定义线条颜色	有效值：ColorSpec
对象访问的控制		
HandleVisibility	定义线条句柄对其他函数是否可见	有效值：on、off、callback 缺省值：on
HitTest	定义线条能否成为当前对象	有效值：on、off 缺省值：on
关于线条的一般信息		
Children	线条没有子对象	有效值：[]（空矩阵）
Parent	线条对象的父对象为坐标轴对象	有效值：坐标轴句柄
Selected	是否显示线条的“选中”状态	有效值：on、off 缺省值：on
Tag	用户定义的标签	有效值：任一字符串 缺省值：”（空字符串）
Type	图形对象的类型（只读类型）	有效值：'line'
UserData	用户定义的数据	有效值：任一矩阵 缺省值：[]（空矩阵）
与回调程序执行有关的属性		
BusyAction	定义如何处理回调中断程序	有效值：cancel、queue 缺省值：queue
ButtonDownFcn	当在线条上按下鼠标时,定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
CreateFcn	当生成线条时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
DeleteFcn	当删除线条时，定义一回调程序	有效值：字符串 缺省值：' '（空字符串）
Interruptible	定义回调程序是否可中断	有效值：on、off 缺省值：on（可中断）
UIContextMenu	定义与线条相关的菜单	有效值：UIContextMenu 的句柄

例 7-47

```
>>t = 0:pi/20:2*pi;
>>hline1 = plot(t,exp(t).*sin(t),'k');
>>hline2 = line(t+.06,exp(t).*sin(t),'LineWidth',4,'Color',[.8 .8 .8]);
>>set(gca,'Children',[hline1 hline2])
```

生成图形为图 7-46。

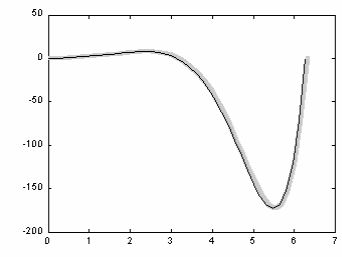

图 7-46　命令 line 画的函数图

例 7-48

生成随机直线图：

```
>>line(rand(4,2),rand(4,2),rand(4,1))
>>line(rand(1,4),rand(1,4),rand(1,4))
>>line(rand(4,1),rand(4,1),rand(4,1))
>>line(rand(2,4),rand(2,4),rand(1,4))
>>line(rand(4,2),rand(4,2),rand(4,1))
```

生成图形为图 7-47。

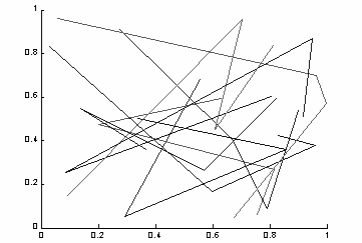

图 7-47　随机直线图

命令 3　**patch**

功能　生成补片图形对象。该命令为生成补片图形对象的低级图形函数。补片为一个或多个多边形，多边形的顶点为坐标中的点。用户可以指定补片的颜色与光照模式。

用法　patch(X,Y,C)　在当前坐标轴中增加二维带填充模式的补片。参量 X,Y 确定顶点的位置。若 X,Y 为矩阵（同型或不同型），系统按列生成多个多边形。若 x，y 没有定义一封闭的多边形，则命令自动地将多边形封闭。参量 x 与 y 可以定义凹的或自身交叉的多边形。可是，一个不可分隔的补片的边界自身交叉，则不能完整填充。在这种情况下，最好是将多边形分解成几个小的、自身没有交叉的多边形。参量 c 指定每一补片的颜色，它可以为简单的 ColorSpec，每面一颜色或每一顶点一颜色。若 c 为三维列向量，它将被认为是一直接指定的 RGB 颜色。

patch(X,Y,Z,C)　生成三维的补片对象。

patch(FV)　用结构 FV 生成一补片。结构 FV 包含这些域名 vertices，faces 和可选的 facevertecdata，这些域名对应于补片的 Vertices 属性、Faces 属性、FaceVertexCData 属性。

patch(…,C,'PropertyName',PropertyValue…)　在二维(X,Y)或三维(X,Y,Z)空间中对补片指定的属性 PropertyName 设置为 PropertyValue。

patch('PropertyName',PropertyValue…)　对所有指定的多个属性 PropertyName 设置

为相应的值 PropertyValue。该命令形式可以使用户免除颜色的指定，因为系统将使用缺省的面颜色和边界颜色，除非用户准确地对属性 FaceColor 与 EdgeColor 进行设置。该命令形式也允许用户通过对属性 Faces 与 Vertices 的设置来代替 x-，y-与 z-轴的输入。

handle = patch(…) 返回命令 patch 生成的补片对象句柄。

说明 函数 patch 不象其他的高级的区域生成函数，例如函数 fill 或 area，它没有检测图形窗口与坐标轴的属性 NextPlot 的设置情形。它只是简单地在当前坐标轴中添加补片对象而已。

有两种指定颜色的补片属性名：

（1）Cdata——当指定 x-，y-与 z-轴坐标(XData,YData,ZData)时使用；

（2）FaceVertexCData——当指定多边形的顶点与连接矩阵时使用。

以上两个属性接受颜色数据作为索引颜色或者是真颜色(RGB)。其中索引颜色数据 能代表当前色图的直接索引或者代表映射到整个色图的线性数据的比例数值。

命令 4 **surface**

功能 生成面对象。该命令是生成面图形对象的低级函数。面对象为由矩阵元素的 A（I，j）所在的行下标 I 为 x-坐标，所在的列下标 j 为 y-坐标，元素值为 z-坐标确定的点生成的空间多边形。

用法 surface(Z) 画出由矩阵 z 确定的曲面，其中 z 为定义在一几何矩形区域上的单值函数。

surface(Z,C) 画出颜色由 c 指定的、面由 z 指定的空间曲面。

surface(X,Y,Z) 曲面由参数 x，y，z 确定，颜色参数 c=z，因此颜色能恰当地反映曲面的高度。

surface(X,Y,Z,C) 曲面由参数 x，y，z 确定，颜色由参数 c 确定。

Surface(x,y,Z) 参数 x 与 y 为向量，若[m,n]=size(z)，则要求 length(x)=n，length(y)=m，面上的点由(x(j),y(i),z(I,j))确定。

Surface(x,y,Z,C) 曲面确定如上情形，颜色由参数 c 确定。

surface(…'PropertyName',PropertyValue,…) 对指定的曲面属性 PropertyName 指定为 PropertyValue，对曲面进行细微控制。

h = surface(…) 返回生成面对象的句柄。

命令 5 **image**

功能 显示图片对象。该命令通过对矩阵 c 中每一个元素（每一元素作为引用图形色图下标或直接给出 RGB 值）的解释而生成一个图片对象。Image 命令有两种使用格式：

1．一个调用命令 newplot 的高级函数，可以确定在何处放置图片与坐标轴的范围为刚好围住图片；使刚生成的图片放置在坐标轴的刻度线与格栅线之上；属性 Ydir 设置为 rervse；属性 View 为[0 90]。

2．一个增加图片到当前坐标轴的低级命令，而没有调用命令 newplot，在低级使用形式中，只能对指定属性进行设置操作。

用户在命令的输入参量中可以输入属性名/属性值，结构数组，细胞数组等。

用法 image(C) 把 C 作为一图片进行显示。C 中的每一个元素指定了一个“图片”矩

形中的相应部分的颜色。

image(x,y,C) 在(x,y)确定的位置上画 C 的元素。其中 x，y 都为 2 维矩阵，分别指定 x 轴与 y 轴的范围，其效果与 image(C)相同，只不过是进行了恰当的比例缩放。

image(x,y,C,'PropertyName',PropertyValue,…) 该形式为指定属性名/属性值的高级使用形式，在执行该命令之前，先执行命令 newplot。

image('PropertyName',PropertyValue,…) 该形式为低级使用形式，它只接受属性名/属性值的输入。

handle = image(…) 返回刚生成的图片对象的句柄。用户可以从上面的任何形式的调用后获得图片句柄。

例 7-49

```
>>load clown
>>image(X,'CDataMapping','scaled')
>>colormap(map)
```

图形结果为图 7-48。

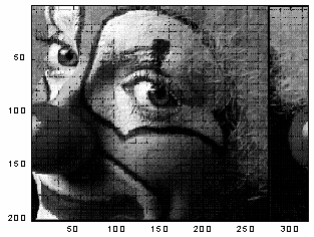

图 7-48

命令 6　**uicontrol**

功能　生成用户控制图形对象（用户界面控制）。也通过该命令运行图形用户界面。当对象被选中时，一般会执行相应的操作。系统支持多种控件，每一种都有不同的作用：

- 校验框——当单击检验框时，会执行一操作。该组件对于提供用户多个独立的选择是很有用的。要激活一校验框，只需用鼠标单击该组件即可，且选中的状态在组件上显示出来。
- 可编辑文本框——允许用户输入与修改文本文字的区域。当用户想把文字作为输入时，可使用该组件。若一可编辑文本框有焦点，则单击文本框的菜单栏不会执行任何操作。因此，在单击菜单条后，语句 get(edit_handle,'String')并没有返回当前编辑框中的内容。因为系统必须执行回调函数来改变属性 string 的值，即使屏幕上显示的文字已经改变。
- 框架——该组件为一封闭的、可见的、图形窗口区域。框架能使一用户图形界面中相关的控制组件能容易理解。框架没有相关的回调程序。只有控制组件能在框架中显示。框架不是透明的，因此用户定义的组件先后顺序决定了组件是否被框架遮住或可见。属性 Stacking order 决定了控制组件的显示顺序：第一个定义的组件最先显示，后面定义的控制组件则覆盖已经存在的组件。若用户要用一框架包围一些组件，则必须第一个定义框架。
- 列表框——显示一些项目的列表（用命令 string 设置），且允许用户选择一个或多个项目。属性 Min 与 Max 控制着选择的模式。属性 Value 显示可选择的项目与包含着字符串列表中项目的索引；对于选择了多个项目则用向量表示。在任何的能改变属性 Value 值的、鼠标松开的操作之后，系统 MATLAB 将马上执行列表框的回调函数。因此，用户有必要增加一“ Done ”按钮，用于推迟当要多次选择项目时的操作。在执行列表框回调函数 Callback 属性之前，列表框中项目的选择有单击或双击之分，对应于将图形窗口属性 SelectionType 设置为 normal 或 open。

- 弹出菜单——当组件被按下时，打开且显示一选择列表（用命令 string 设置）。当没有打开时，该组件显示当前的选择项。该组件对于用户想给其他用户提供一系列的互斥的选择项，又不想占用太多的区域。
- 普通按钮——当该组件被按下时，将执行一操作。要激活一按钮，只需在按钮上按下鼠标按钮。
- 单选按钮——该组件与校验框相类似，但它包含几个互斥的、而且相关的选项（例如在任意时刻，只能选择一个状态）。要激活某一单选按钮，只需在该组件上按下鼠标即可。被选中的组件同时显示出来。
- 滑块——该组件允许用户通过移动某一范围之内的滑块来输入一指定的数值。用户要移动一滑块，只需在滑块上按下鼠标不放，且在滑块方向上移动；或者是在滑槽内单击鼠标；或者是单击滑块条上的箭头。当松开鼠标后，滑块所在位置将与一数值对应。用户可以设置滑块的最大值、最小值与当前值等。
- 静态文本框——显示文本行。静态文本经常作为其他控制对象标签，以提供其他用户相关信息，或者是显示一滑块的数值。其他用户不能交互地改变静态文本，因此对于静态文本，没有相关的回调函数。
- 触发按钮——当该组件被单击且显示出它们的状态（on 或者 off）时，控制是否执行回调函数。

用法　handle = uicontrol(parent) 在父对象 parent 上生成一用户图形控制界面。用户界面控制对象都是图形窗口的子对象，所以当窗口中没有坐标轴时，同样可以放置控制组件于图形窗口中。

handle = uicontrol(…,'PropertyName',PropertyValue,…) 参量 PropertyName 为属性名，参量 PropertyValue 可为结构数组或者为单元数组，同时随意地返回刚生成的对象的句柄。当然用户可以通过命令 set 与 get 来设置与询问生成对象的属性值。

附：表 7-12 列出所有的用于命令 uicontrol 对象的属性名。每一属性名当作一描述该属性的链接。

表 7-12

属性名	属性名含义	属性值
控制控件类型与显示		
BackgroundColor	对象的背景颜色	有效值：ColorSpec 缺省值：与系统有关
Cdata	显示于对象之上的真颜色图片	有效值：矩阵
ForegroundColor	文本字体的颜色	有效值：ColorSpec 缺省值：[0 0 0]（黑色）
SelectionHighlight	当对象被选中时突出显示	有效值：on、off 缺省值：on
String	用户控制界面的标签，也是列表框与弹出菜单中的项目	有效值：任意有效的字符串
Visible	用户界面控制是否可见	有效值：on、off 缺省值：on
关于控件对象的一般信息		
Children	用户界面控制界面没有子对象	

Enable	用户界面控制是否可用	有效值：on、inactive、off 缺省值：on
Parent	用户界面控制对象的父对象	有效值：图形窗口标量句柄
Selected	对象是否为选中状态	有效值：on、off 缺省值：off
SliderStep	滑块步长尺度	有效值：二维向量 缺省值：[0.001 0.1]
Style	用户界面控制对象的类型	有效值：pushbutton、edit、togglebutton、slider、text、radiobutton、popupmenu 、listbox、frame 缺省值：pushbutton
Tag	由用户指定的对象的标记符	有效值：任意有效字符串
TooltipString	对象的工具提示	有效值：任意有效字符串
Type	图形对象的类型	有效值：字符串（只读） 缺省值：uicontrol
UserData	用户指定的数据	有效值：矩阵
控制控件对象的位置		
Position	用户界面控制对象的大小与位置	有效值：位置矩形 缺省值：[20 20 60 20]
Units	解释属性 position 向量的单位	有效值：pixels、inches、character、normalized、points、centimeters 缺省值：pixels
控制字体与标签		
FontAngle	字符的倾斜度	有效值：normal、italic、oblique 缺省值：normal
FontName	字体系列名称	有效值：字符串 缺省值：与系统有关
FontSize	字体大小	有效值：一标量 缺省值：与系统有关
FontUnits	字体大小单位	有效值：pixels、normalized 、inches、centimeters、points 缺省值：points
FontWeight	文本字体的磅值	有效值：light、normal、demi、bold 缺省值：normal
HorizontalAlignment	标签字符串的对齐方式	有效值：left、center、left 缺省值：决定于用户界面控制的对象
String	用户控制界面的标签，也是列表框与弹出菜单中的项目	有效值：字符串
控制回调函数的执行		
BusyAction	回调函数中断方式	有效值：cancel、queue 缺省值：queue
ButtonDownFcn	当按钮按下时执行的回调函数	有效值：字符串
Callback	控制操作	有效值：字符串
CreateFcn	在对象生成过程中执行的回调函数	有效值：字符串
DeleteFcn	在对象删除过程中执行的回调函数	有效值：字符串
Interruptible	回调函数中断的模式	有效值：on、off 缺省值：on
UIContextMenu	与界面控制中的对象相关的菜单（如按下鼠标右键）	有效值：句柄
关于当前状态的一般信息		
ListboxTop	第一个显示于列表框中的项目的索引	有效值：标量 缺省值：[1]
Max	最大值（与用户界面控制对象有关）	有效值：标量

		缺省值：与系统有关
Min	最小值（与用户界面控制对象有关）	有效值：标量 缺省值：与系统有关
Value	用户界面控制对象的当前值	有效值：标量或向量 缺省值：与系统有关
控制组件的访问		
HandleVisibility	句柄是否可从命令窗口中与 GUIs 中访问	有效值：on、callback、off 缺省值：on
HitTest	组件是否可由鼠标单击选中	有效值：on、off 缺省值：on

命令 7 **uimenu**

功能 生成图形窗口的菜单中的层次的菜单与下一级子菜单。即增加新的菜单于已经存在的菜单后面，当一菜单项被选中时，该菜单项与它的下一级菜单也将显示。也可用该命令生成与组件相关的菜单。

用法 handle = uimenu('PropertyName',PropertyValue,…) 在当前图形窗口菜单条上用指定的属性 PropertyName 与相应的属性值 PropertyValue 创建一菜单，同时将该菜单的句柄赋给 handle。其中两个输入参量可以是结构数组或者是单元数组。用户界面菜单的回调函数属性定义了当用户激活菜单项时，进行的响应操作。

uimenu('PropertyName',PropertyValue,…) 效果同上，但不返回句柄值。

handle = uimenu(parent,'PropertyName',PropertyValue,…) 生成一父菜单的子菜单，或者是生成由 parent 指定的相关菜单中的菜单项目。若 parent 不是另外的用户界面菜单对象或用户界面相关菜单对象，而是一图形窗口，则系统将生成该图形窗口菜单条上的新的菜单。同时将生成的菜单赋值给句柄 handle。

uimenu(parent,'PropertyName',PropertyValue,…) 效果同上，但不返回菜单的句柄。

附：表 7-13 列出了所有对 uimenu 对象有用的属性，分别按功能进行了分类。每一属性名作为该属性描述的索引。

表 7-13

属性名	属性名描述	属性值
控制控件类型与显示		
Checked	菜单检查记号	有效值：on、off 缺省值：off
ForegroundColor	文本的颜色	有效值：ColorSpec 缺省值：黑色[0 0 0]
Label	菜单标签	有效值：任何字符串
Separator	分隔线模式	有效值：on、off 缺省值：off
SelectionHighlight	对象选中时是否突出显示	有效值：on、off 缺省值：on
Visible	用户界面菜单是否可见	有效值：on、off 缺省值：on
关于对象的一般信息		
Acceleratro	键盘等价字符	有效值：任何的字符
Children	子菜单的句柄	有效值：句柄向量
Enable	用户界面菜单是否可用	有效值：on、off 缺省值：on
Parent	用户界面菜单的父对象	有效值：句柄

Tag	用户指定的对象标记符	有效值：任何字符串
Type	图形对象类型	有效值：字符串 read-only 缺省值：uimenu
UserData	用户指定数据	有效值：任何矩阵
控制对象的位置		
Position	用户界面菜单的相对位置	有效值：标量 缺省值：[1]
控制回调程序的执行		
BusyAction	回调程序的中断	有效值：cancel、queue 缺省值：queue
ButtoDownFcn	按钮按下回调程序	有效值：字符串
Callback	控制操作	有效值：字符串
CreateFcn	在对象生成期间执行的回调程序	有效值：字符串
DeleteFcn	在对象删除期间执行的回调程序	有效值：字符串
Interruptible	回调程序中断模式	有效值：on、off 缺省值：on
控制对象的访问		
HandleVisibility	是否可从命令行上访问图形用户界面	有效值：on、callback、off 缺省值：on
HitTest	是否可用鼠标选择	有效值：on、off 缺省值：on

7.3.2　轴的产生和控制命令

命令 1　**axes**

功能　创建坐标轴图形对象。该命令是创建坐标轴图形对象的低级函数命令。

用法　axes　在当前图形窗口中用缺省的属性值创建一坐标轴图形对象。

axes('PropertyName',PropertyValue,…)　用参量'PropertyName'指定的属性名与用参量 PropertyValue 指定的属性值创建一坐标轴。对于没有指定的属性名，系统则使用缺省的属性值。

axes(h)　使已经存在的坐标轴 h 成为当前的坐标轴。同时使坐标轴 h 为图形窗口中的所有子对象属性（Children property）的第一坐标轴，也使图形窗口的 CurrentAxes 属性为 h。当前坐标轴是图形函数 image、line、patch、surface 与 text 等命令输出图形对象的目的地。

h = axes(…)　返回已经创建的坐标轴对象的句柄。

命令 2　**cla**

功能　清除当前坐标轴。该命令在命令窗口中执行与在回调程序中执行效果是一样的，即它不能区别由 callback 设置的属性 HandleVisibility，也就是说，当它从一回调程序中执行时，命令 cla 仅仅删除属性 HandleVisibility 为 on 的图形对象。

用法　cla　清除当前坐标轴中所有句柄为不隐藏（例如，图形对象属性 HandleVisibility 设置为 on）的图形对象。

cla reset　无条件地清除当前坐标轴中所有图形对象，且重新设置坐标轴的属性，（除了属性 Position 和 Units）。

命令 3　**gca**

功能　获取当前坐标轴句柄。

用法　h=gca　返回当前图形窗口中的坐标轴句柄。若坐标轴不存在，系统则生成一坐标

轴同时返回它的句柄。用户不想得到上面的结果，可以输入 get(gcf,’CurrentAxes’)。

当前坐标轴为用户创建坐标轴以下子对象的目的地。有许多图形命令可以在当前坐标轴中画出图形对象，如：plot，text，surf 等。改变了当前窗口，相应地改变了当前坐标轴。

7.3.3 图形句柄操作命令

命令 1 **gco**

功能 返回当前对象的句柄。“ 当前对象 ”为最后用鼠标单击的对象，除了命令 uimenus 之外。若鼠标没有单击到一图形对象之下的子对象，则该图形对象为 “ 当前对象 ”。系统会把当前图形对象的句柄存放于图形的属性 CurrentObject 之中。当前图形窗口中的当前对象并非总是那些它们的回调函数，而是正在执行的对象。其他函数的回调中断函数可以改变当前对象或者甚至是当前图形窗口。一些回调函数，如生成命令 CreateFcn、删除命令 DeleteFcn 与用户界面菜单命令 Callback 等就没有改变当前图形窗口或者当前对象。

用法 h = gco 返回当前对象的句柄给 h。

h = gco(figure_handle) 返回指定窗口 figure_handle 中的当前对象的值。

命令 2 **get**

功能 获取对象属性。

用法 get(h) 返回由句柄 h 指定的图形对象的所有属性与相应的当前属性值；

get(h,'PropertyName') 返回由句柄 h 指定的图形对象的指定属性 PropertyName 的属性值。

<m-by-n value cell array> = get(H,<property cell array>) 返回由 m 个图形对象的 n 个属性值组成的 m*n 阶的细胞数组，其中 m=length(H)，且 n 为指定的属性细胞数组<property cell of array>中包含的属性名个数。

a = get(h) 返回一结构，其中该结构的域名为该对象的属性名，结构的域名值为相应属性的当前值。H 必须为标量。若用户没有指定输出参量，则系统将信息显示于屏幕之上。

a = get(0,'Factory') 返回所有能由用户设置的属性的缺省定义值。输出参量 a 为一结构数组，该结构的域名为对象的属性名，域名值为相应属性的当前值。若用户没有指定输出参量，则系统将信息显示于屏幕之上。

a = get(0,'FactoryObjectTypePropertyName') 返回指定对象类型的指定的属性的缺省属性值。输入参量 FactoryObjectTypePropertyName 为一关键字，由字符 Factory 与对象类型（如：Figure）还有属性名(如：Color)组成：FactoryFigureColor

a = get(h,'Default') 返回由句柄 h 指定的对象的所有缺省属性值。输出参量 a 为一结构，该结构的域名为缺省值对应的属性名。若用户没有指定输出参量，则系统将该结构信息显示于屏幕。

a = get(h,'DefaultObjectTypePropertyName') 返回对象类型的指定属性的缺省属性值。输入参量 DefaultObjectTypePropertyName 为一关键字，该字由字符 Default 与对象类型名（例如：Figure）还有具体的属性名（例如：Color）组成：DefaultFigureColor

例 7-50

若想获得定义于屏幕之上的图形对象属性 LineWidth 的缺省属性值，输入：get(0,'DefaultLineLineWidth')

命令 3　**set**

功能　设置对象的属性。

用法　set(H,'PropertyName',PropertyValue,…) 用属性值'PropertyValue'设置关于用参量 H 标志的对象（一个或多个）的属性名'PropertyName'（一个或多个）。H 可以为一句柄的向量。在这种情形下，命令 set 可以设置所有对象的属性值。

set(H,a) 用指定的属性值设置由 H 标志的对象的属性。其中 a 为一结构数组，该结构数组的域名为对象的属性名，域名值为相应属性名的属性值。

set(H,pn,pv…) 对由 H 指定的所有对象中指定的细胞数组属性名 pn 设置为相应的细胞数组属性值 pv。

set(H,pn,<m-by-n cell array>) 对于每 m 个图形对象设置 n 个属性值，其中 m=length(H)，n 为包含属性名的细胞数组 pn 中包含的属性名个数。即允许用户对每一对象的指定的属性设置不同的属性值。

a= set(h) 返回句柄 h 中允许用户设置的属性名与可能的属性值。输出参量 a 为一结构数组，其域名为对象的属性名，域名值为相应的属性名对应的属性值。若没有指定输出参量 a，则系统自动将信息显示于屏幕，h 必须为标量。

a= set(0,'Factory') 返回那些用户可以设置缺省值的所有对象的属性名，同时显示可能的属性值，输出参量 a 为一结构数组，其域名为对象的属性名，域名值为相应的属性名对应的属性值，若没有指定输出参量 a，则系统自动将信息显示于屏幕。

a= set(0,'FactoryObjectTypePropertyName')返回指定根对象（0）类型中指定的属性名 ObjectTypePropertyName 的所有可能的属性值。输入参量是由固定的关键字 Factory、对象类型（如 axes）与属性名（如 position 等）组成。

a= set(h,'Default') 返回由 h 标记的对象上缺省设置的值，其中 h 必须是标量。

a= set(h,'DefaultObjectTypePropertyName') 返回指定对象 h 的类型中指定的属性名 ObjectTypePropertyName 的所有可能的属性值。输入参量是由固定的关键字 Factory、对象类型（如 axes）与属性名（如 position 等）组成。

命令 4　**reset**

功能　重新设置图形对象的属性为它们的缺省值。

用法　reset(h) 重新设置由句柄 h 指定的图形对象的属性为系统为它们设置的初始值。若 h 为一图形 figure，该命令不能重新设置属性 Position，Units，PaperPosition 和 PaperUnits；若 h 为一坐标轴 axes，该命令不能重新设置属性 Position 和 Units。

例 7-51

```
reset(gca)    %重新设置当前坐标轴的属性。
reset(gcf)    %重新设置当前图形的属性。
```

命令 5　**delete**

功能　删除文件或图形对象。作为一可供选择的函数，用户可从当前目录浏览器(Current

Directory browser)中删除文件。要打开该浏览器，从 MATLAB 桌面上的 View 菜单中选择 Current Directory 命令。

用法　delete filename 从磁盘上删除指定的文件 filename。参量 filename 可以是绝对路径或与当前路径相关的路径名。其中可以包括通配符（*）。

delete(h) 删除由句柄 h 指定的图形对象。该命令无条件地、直接地删除对象，甚至是图形窗口。

delete('filename') 这是第一种情形的函数形式。当文件名包含于字符串 filename 中时，使用函数形式。

例：

delete('D:\MATLABR12\work*.m') % 将删除指定目录上的所有.m 文件。

命令 6　**findobj**

功能　定位图形对象且返回它们的句柄。用户可用特定的属性值与沿着指定的层次分支来限定搜索条件。

用法　h = findobj 返回根对象与它的所有的子孙对象句柄。

h = findobj('PropertyName',PropertyValue,…) 返回属性名 PropertyName 具有属性值 PropertyValue 的所有图形对象。用户可指定一对或多对 PN 与 PV 值，对此，findobj 返回满足所有条件的那些对象。

h = findobj(objhandles,…) 限定搜索的对象为列表于 objhandles 中的对象与它们子孙对象。

h = findobj(objhandles,'flat','PropertyName',PropertyValue,…) 限定搜索对象为 objhandles 中列出的对象，而不包含它们的子孙对象。

7.3.4　图形窗口的控制命令

命令 1　**subplot**

功能　生成与控制多个坐标轴。把当前图形窗口分隔成几个矩形部分，不同的部分是按行方向以数字进行标号的。每一部分有一坐标轴，后面的图形输出于当前的部分中。

用法　subplot(m,n,p) 将一图形窗口分成 m*n 个小窗口，在第 p 个小窗口中创建一坐标轴。则新的坐标轴成为当前坐标轴。若 p 为一向量，则创建一坐标轴，包含所有罗列在 p 中的小窗口。

subplot(h) 使句柄 h 对应的坐标轴称为当前的，用于后面图形的输出显示。

subplot('Position',[left bottom width height]) 在由 4 个元素指定的位置上创建一坐标轴。位置元素的单位为归一化单位。

h = subplot(…) 返回一新坐标的句柄于 h。

命令 2　**hold**

功能　保持当前图形窗口中的图形。该命令是决定是否在当前坐标轴中只能增加新的图形对象还是覆盖原有图形对象。测试保持状态命令为 ishold。该命令可以设置当前坐标轴与当前图形的属性 NextPlot。若一图形窗口中有多个坐标轴，则每个坐标轴有自己的保持状态。

用法　hold on 保留当前图形与当前坐标轴的属性值，后面的图形命令只能在当前存在的坐标轴中增加图形，即设置当前坐标轴属性 NextPlot 为 add。当必要的时

候，坐标轴的一些属性在增加新图时还是要进行相应的改变。例如，当新图形的数据范围超出了当前坐标轴的范围，则命令会自动地改变坐标轴的范围，使能显示新图形。

hold off 在画新图形之前，重新设置坐标轴的属性为缺省值。off 是命令 hold 命令的缺省值。设置当前坐标轴的属性 NextPlot 为 replace。

hold 在 on 与 off 之间转换。即在增加图形与覆盖图形之间切换。当坐标轴不存在时，则生成一坐标轴。同时使当前坐标轴属性 NextPlot 在 add 与 replace 之间切换。

命令 3　**gcf**

功能　获得当前图形窗口的句柄。

用法　h = gcf 返回当前图形窗口的句柄。当前窗口为由命令 plot、title 与 surf 等得到的结果。若不存在图形窗口，则系统自动地生成一个，并返回它的句柄。若用户想当图形窗口不存在时，也不创建新的，则输入：get(0,'CurrentFigure')

命令 4　**clf**

功能　清除当前图形窗口。该命令在命令窗口中执行与在回调程序中执行效果是一样的，即它不能区别由 callback 设置的属性 HandleVisibility，也就是说，当它从一回调程序中执行时，命令 clf 仅仅删除属性 HandleVisibility 为 on 的图形对象。

用法　clf 清除所有当前图形窗口与窗口中的所有那些句柄为不隐藏（例如它们的属性 HandleVisibility 为 on）的图形对象。

clf reset 无条件地清除当前图形窗口中所有的图形对象，且重新设置所有图形窗口属性为缺省值，除了属性 Position，Units，PaperPosition，PaperUnits。

命令 5　**close**

功能　删除指定的图形窗口。

用法　close 删除当前的图形窗口。

close(h) 删除由句柄 h 指定的图形窗口。若 h 为一向量或矩阵，则 close 全部删除其中每一分量指定的图形句柄。

close name 删除指定名字 name 的窗口。

close all 删除所有没有隐藏的图形。

close all hidden 删除所有具有隐藏的图形。

status = close(…) 若成功地删除了指定的对象则返回 status=1，否则返回 0。

命令 6　**newplot**

功能　做好开始画新图形对象的准备。在高级图形 m-文件的开始使用该命令，用于确定在哪一个图形窗口与坐标轴中输出图形。调用命令 newplot 能改变当前窗口与坐标轴。基本上，当要在已经存在的窗口与坐标轴中画图，有三个选项可选：

1．没有改变任何属性与删除任何对象，直接在当前坐标轴中增加新的图形对象；

2．在画图形的对象之前，删除所有存在于当前坐标轴中的，句柄为非隐藏的对象；

3．在画图形的对象之前，无条件删除所有的存在于当前坐标轴中的对象（不管句柄是否为隐藏），同时设置大部分的属性为缺省值；

4．首先，newplot 读取当前图形的属性 NextPlot 的属性值（关于该属性的含义参见 figure 或 axes 的属性表），再执行相应的动作；

5．然后，newplot 确定在哪一个窗口中画图，它读取当前图形的属性 NextPlot 的属性值，执行相应的操作。

用法　newplot 画好图形窗口与坐标轴，后面的图形命令就可以在该坐标轴内画图。

h = newplot 效果如上，且返回当前坐标轴的句柄给 h。

7.4　颜色与光照模式命令

7.4.1　颜色控制命令

命令 1　**colormap**

功能　设置或获取当前色图。色图为一个 m*3 的、元素在 0 到 1 之间的实数的矩阵，每一行为定义一个颜色的 RGB 向量。色图矩阵的第 k 行定义了第 k 个颜色，其中 map(k,:)=[r(k) g(k) b(k)]指定了组成该颜色中红色、绿色、兰色的强度。

用法　colormap(map) 通过矩阵 map 设置色图。若矩阵 map 中的元素不在[0 1]区间之内，则返回一个错误。在目录 color 中的 m-文件能够生成许多色图，每一个 m-文件能够接受颜色数作为函数参数，例如命令 colormap(hsv(64))生成了有 64 种颜色的 hsv 色图。若用户没有指定颜色数，例如命令 colormap(hsv)，生成与当前色图中颜色数相同的 hsv 色图。MATLAB 支持的色图见表 7-14。

表 7-14

色图名称	包含的颜色范围
Cool	青蓝和洋红的色度
Bone	带一点蓝色的灰度
Flag	交替为红色、白色、蓝色和黑色
Jet	Hsv 的一种变形（以兰色开始和结束）
Copper	线性铜色度
Hsv	色彩饱和值（以红色开始和结束）
Hot	从黑色到黄色到白色
Gray	线性灰度
Pink	粉红的彩色度
Prim	三棱镜。交替为红色、橘黄色、黄色、绿色和天蓝色
Lines	线性色图
White	全白色图
Colorcube	增强立方色图
Autumn	红色黄色阴影色图
Spring	洋红黄色阴影色图
Summer	绿色黄色阴影色图
Winter	兰色绿色阴影色图

例 7-52

colormap('default') 设置当前色图为缺省色图。

cmap = colormap 获取当前色图矩阵。

命令 2　**bone**

功能　生成带淡兰色的灰度刻度化的色图。

用法　bone(m) 返回个一个阶数为 m*3 的包含“ bone ”的色图。

bone 返回一个与当前色图行数相同的色图。

命令 3　**cool**

功能　生成带阴影的青色和品红的色图。

用法　cool(m) 返回一个阶数为 m*3 的包含“ cool ”的色图。

cool 返回一个与当前色图行数相同的色图。

命令 4　**copper**

功能　生成线性铜色色图。

用法　copper（m）返回一个阶数为 m*3 的包含“ copper ”的色图。

copper 返回一个与当前色图行数相同的色图。

命令 5　**flag**

功能　生成一个颜色顺序为红、白、兰、黑的色图。

用法　flag（m）返回一个阶数为 m*3 的包含“ flag ”的色图。增加 m 的值，会增加色图的颗粒程度。

flag 返回一个与当前色图函数相同的色图。

命令 6　**gray**

功能　生成一个线性灰度化的色图。

用法　gray（m）返回个一个阶数为 m*3 的包含灰度化的的色图。

gray 返回一个与当前色图函数相同的色图。

命令 7　**hot**

功能　生成一个颜色顺序为黑、红、黄、白的色图。

用法　hot（m）返回个一个阶数为 m*3 的包含“ hot ”的色图。

hot 返回一个与当前色图函数相同的色图。

命令 8　**hsv**

功能　生成一个包含色度-饱和度值的色图。一个 hsv 色图包含各种饱和色度颜色的色度的成分。其颜色从红色到黄色、绿色、青色、蓝色、品红，最后返回红色。该色图对于显示周期函数很有用处。

用法　hsv（m）返回个一个阶数为 m*3 的包含 hsv 的色图。

hsv 返回一个与当前色图函数相同的色图。

命令 9　**jet**

功能　不同于 hsv 色图的另外一种色图。

用法　jet（m）返回个一个阶数为 m*3 的，与 hsv（m）不同的色图，用于显示 NCSA 流体激光图片。

jet 返回一个与当前色图函数相同的色图。

命令 10　**pink**

功能　生成一个带柔和阴影粉红色图。

用法　pink（m）返回一个阶数为 m*3 的包含“ pink ”的色图。

pink 返回一个与当前色图函数相同的色图。

命令 11　**prism**

功能：生成一个三棱镜色图。如同 hsv 色图一样，prism 色图中的颜色使用顺序是一样的，不同的是，命令 prism 重复使用它的六中颜色，而命令 hsv 是连续地变换它的颜色。

用法　prism（m）返回一个阶数为 m*3 的包含六种循环使用的颜色：红色、橙色、黄色、绿色、蓝色、紫色。

prism 这种没有任何输入输出参量的形式，改变当前坐标轴中的线对象的颜色为三棱镜中的颜色。

7.4.2　色图控制命令

命令 1　**brighten**

功能　增亮或变暗色图。

用法　brighten(beta) 增亮或变暗当前的色图。若 0<beta<1，则增亮色图；若-1<beta<0，则变暗色图。改变的色图将代替原来的色图，但本质上是相同的颜色。

brighten(h,beta) 对指定的句柄对象 h 中的子对象进行操作。

newmap = brighten(beta) 该命令没有改变当前图形的亮度，而是返回变化后的色图给 newmap。

newmap = brighten(cmap,beta) 该命令没有改变指定色图 cmap 的亮度，而是返回变化后的色图给 newmap。

命令 2　**colorbar**

功能　显示能指定颜色刻度的颜色条。且调整当前坐标轴，以适应当前的颜色条。

用法　colorbar 更新最近生成的颜色条。或若当前坐标轴没有一颜色条，则在右边显示一垂直的颜色条。

colorbar('vert') 增加一垂直的颜色条到当前的坐标轴。

colorbar('horiz') 增加一水平的颜色条到当前的坐标轴。

colorbar(h) 用坐标轴 h 来生成一颜色条。若坐标轴的宽度大于高度，则颜色条是水平放置的。

h = colorbar(…)返回一颜色条句柄 h，该句柄是一坐标轴对象。

colorbar(…,'peer',axes_handle) 生成一与坐标轴 axes-handle 有关的颜色条，代替当前的坐标轴。

命令 3　**contrast**

功能　提高灰度色图的对比度。该命令可以增强图像的对比度。

用法　cmap = contrast(X) 返回一灰度色图，该色图与当前色图有相同的维数。参量 cmap 为生成的灰度色图。

cmap = contrast(X,m) 返回维数为 m*3 的灰度色图 cmap。

例 7-53

```
>>load clown;
>>cmap = contrast(X);
>>image(X);
>>colormap(cmap);
```

命令 4　**rgbplot**

功能　画出色图。

用法　rgbplot(cmap) 画出维数为 m*3 的色图矩阵 cmap 的每一列，矩阵的第一列为红色强度，第二列为绿色强度，第三列为蓝色强度。

命令 5　**diffuse**

功能　漫反射率。

用法　R = diffuse(Nx,Ny,Nz,S) 返回曲面的漫反射率向量[Nx,Ny,Nz]，S 为一三维向量，用于定义光源的方向；S 也可以为球面坐标系中的二维向量[Theta,Phi]。

Lambert 定律：R = cos(PST)，其中 PST 为曲面法线与光源方向之间夹角。

命令 6　**specular**

功能　镜面反射率。

用法　R = specular(Nx,Ny,Nz,S,V,spread) 返回一曲面的镜面反射率向量[Nx,Ny,Nz]，向量参量 S 与 V 分别用于指定光源位置与观察点的位置。它们可以为三维直角坐标系向量[x,y,z]或者为二维球面向量[Theta,Phi]。当标准向量的方向为(S+V)/2，则镜面的高光效果最强。第六个参量 spread 为镜面反射扩散系数。

命令 7　**surfl**

功能　三维带光照模式的阴影图。图形的色泽取决于曲面的漫反射、镜面反射与环境光照模式。

用法　surfl(…)效果与命令 surf(…)基本上一样，除了它受光源影响的曲面之外。

surfl(Z)、surfl(X,Y,Z)、surfl(Z,S)、surfl(X,Y,Z,S)、surfl(X,Y,Z,S,K) 这些都是有效的使用形式。若参数中有 S，则为一三维向量[Sx,Sy,Sz]，用于指定光源的方向。S 也可视为点坐标系下的二维向量[AZ,EL]。S 的缺省值为从当前观察方向逆时针旋转 45 度。使用命令组 cla；hold on；view(AZ,EL)；surfl(…)；hold off 等可画出视角方向为(AZ,EL)的带光照模式的曲面图。第五参数 K=[ka,kd,ks,

spread]指定环境光、漫反射光、镜面反射光、扩散系数等的强弱。

surfl(…,'light') 用 LIGHT 对象生成一带颜色的、带光照模式的曲面。该命令可以生成与用缺省光照模式不同效果的曲面。

surfl(…,'cdata') 指定的曲面的反射光的颜色为 cdata。

H = surfl(…) 返回曲面与光源的句柄。