

第14章 高级图形

MATLAB的图形系统是面向对象的，也就是说图形的输出，如曲线，是建立图形对象。通常用户不必去关心这些高级 MATLAB 命令包含的对象。然而有时为了调整对象也要用一些低级的MATLAB命令。

MATLAB中介绍了图形用户界面 (GUI) 的应用，如单选按钮、滑标和菜单。利用这些用户能够很容易地进行图形控制。

在MATLAB中加入一系列的图片就可以创建出动画来，利用这些动画可以做一些有趣的演示。

14.1 图的结构

一个图形是由许多的图形对象组成的，这些对象是以层次顺序保存的。举一个例子来说明它们之间的关系。

例14.1

用不同线型绘制图形的例程。

```
clear;
x=0.1:0.1:4*pi;           % 生成向量x。
y1=sin(x);                % 生成y1值。
y2=sin(x) ./x;            % 生成y2值。
figure;                   % 创建一个新窗口。
subplot(1, 2, 1);         % 定义第一个子图区域。
plot(x, y1);              % 用实线画曲线。
subplot(1, 2, 2);         % 定义第二个子图区域。
plot(x, y2, ' * ');       % 用 '*' 号画曲线。
```

上述命令在图形窗口的两个子域内画出了关于 x 的两个函数 y_1 和 y_2 的图形。结果如图 14-1 所示。

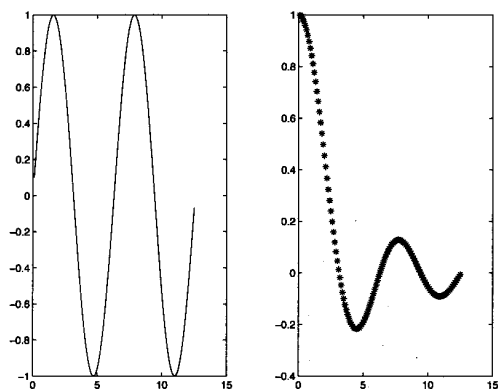


图14-1 用不同线型绘制的图形

一个图形由安排在一个层次结构上的五个图形对象组成。首先有一个窗口，这是一个图形对象。然后由两个轴对象来定义相应的坐标轴。这些都可以用 `subplot` 命令来完成。最后，用 `plot` 命令来创建两个线条对象。例14.1中的对象层次结构在图14-2中给出。

对于每一个对象都可以修改它的一些属性。例如，可以改变图形窗口的位置和图形对象的背景色。对于一个轴对象可以改变它在图形区域内的刻度大小和位置。线条对象可以变得更细，改变成另一种颜色，另一种线型等等。

由于是层次结构，所以某个对象改变时，会影响到这个结构中它以下的所有对象。如果使用鼠标改变图形对象的屏幕位置，线条和轴对象也会跟着变。但是如果改变右边轴对象的轴刻度，那么只影响这个轴上的线条。

图14-2中还少一个对象：根对象。它是所有图形对象的根，也就是整个层次结构的根。

在创建对象的同时可以直接修改一些对象的属性。在画 `y2` 图形时，`*` 号表示用星号来画曲线而不是用实线。调用 `subplot` 可以规定图形区域在窗口的位置。然而还有许多属性只能用在后面描述的 `set` 命令来改变。下面的例子修改了上例中图形的部分属性。

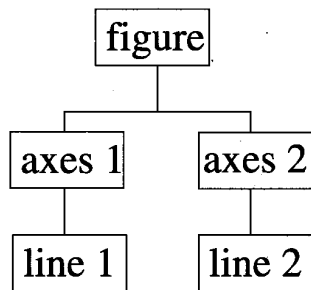


图14-2 对象层次结构

例14.2

% 在上例中，改变左边子图区域的位置和第2个子图的x轴坐标

```
clear;
```

```
x = 0.1:0.1:4*pi;
```

```
y1 = sin(x);
```

```
y2 = sin(x)./x;
```

```
fg = figure;
```

% 创建窗口和图句柄

```
r1 = subplot(1,2,1);
```

% 创建子图和轴句柄

```
l1 = plot(x,y1);
```

% 创建线条和线条句柄

```
r2 = subplot(1,2,2);
```

```
l2 = plot(x,y2,'*');
```

```
disp('The previous example');
```

```
pause;
```

```
set(r1,'Position',[0.1 0.1 0.3 0.3]); % 改变位置
```

```
set(l1,'LineWidth',5); % 加粗线条
```

```
set(r2,'XTick',[1 4 11]);
```

% 改变x轴

```
set(l2,'LineStyle','*');
```

% 改变线型

```
pause;
```

```
delete(fg);
```

% 删除窗口

在使用命令figure、subplot和plot时定义一个变量名来创建对象的句柄或者标识。通过使用句柄可以用set命令来修改对象属性。在本例中修改对象的一个属性，改变了第1个子图的位置和第2个子图中的x轴刻度值。现在得到的窗口如图14-3所示。

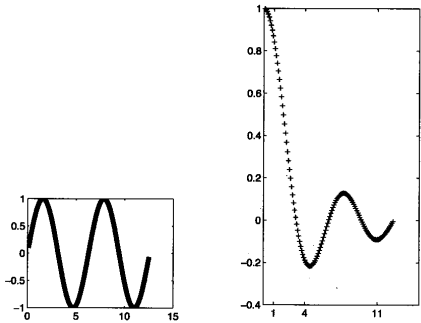


图14-3 对图14-1修改一些属性后的图形

14.2 图形对象

MATLAB 5.2中的各种对象都列在表14-1中。

在图14-4中给出了对象的层次结构图。

在图14-5中给出了不同图形对象的例子。

父对象影响它所有的子对象，这些子对象又影响它们的子对象，以此类推。结果是轴对象会影响像对象，但不影响用户界面控制。

根据表14-1可知，根对象的句柄值是零，而图形对象的句柄值是整数，其他对象则用浮点值作为句柄值。

画一个对象，可以使用和对象名字的不同低级命令。如画一线条，可以用命令 line。

对象的属性有两类：

- 属性，用来决定对象的显示和保存的数据。
- 方法，用来决定在对对象操作时调用什么样的函数，如当创建或者删除对象时，或当用户点击它们时。

一些属性有缺省值，如果没有特殊说明，就是用这些缺省值。

有一些属性是用来规定对象色彩的，它们以RGB三元组的形式给出，也就是说，用一个有三个元素的向量[r g b](0 ≤ r, g, b ≤ 1)来表示颜色中的红、绿和蓝色，例如，用[1, 0, 0]表示红色。也可以用预定义在MATLAB中表示颜色的字符串来代替RGB三元组，如'black'和'blue'。

在helpdesk中的句柄图形对象给出各种不同类型对象的详细说明。 MATLAB手册《使用MATLAB图形和用MATLAB建立GUI》也可得到相关的信息。

表14-1 图形对象

对 象	父 代	描 述
root	—	屏幕是一个根对象。所有其他的图形对象都是根的子对象。根对象的句柄值是零

(续)

对 象	父 代	描 述
figure	root	屏幕上的窗口是一个图形对象，对象的句柄值是整数，在窗口的标题中给出
axes	figure	轴对象在窗口中定义一个图形区域。可以用来描述子对象的位置和方向
uicontrol	figure	用户界面控制。当用户用鼠标在控制对象上点击时，MATLAB会完成一个相应规定的任务
uimenu	figure	创建一个窗口菜单，用户用这些菜单能够控制程序
uicontext-menu	figure	创建一个图形对象的快捷菜单。也就是当用户点击图形对象时会显示出菜单来
image	axes	用当前的色图矩阵定义一个图像。图像可以有自己的色图
line	axes	用plot、plot3、contour和contour3创建一些简单的图形
patch	axes	创建补片对象
surface	axes	输入定义一个有四个角的曲面，可以用实线或内插颜色来绘制，或者作为一个网格
text	axes	字符串，它的位置由它的父对象——轴对象来指定
light	axes	定义多边形或者曲面的光照

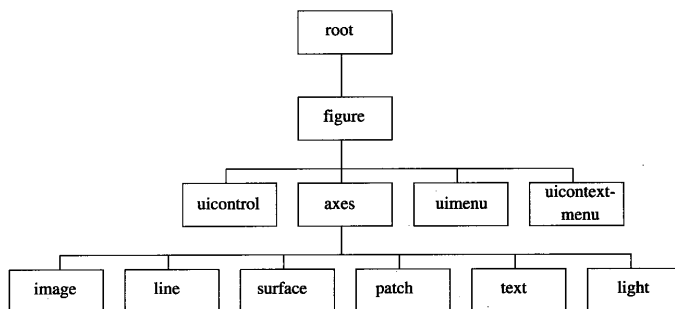


图14-4 图形对象层次结构图

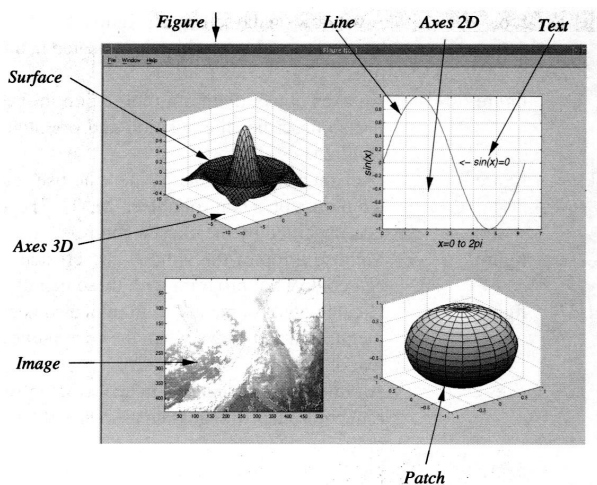


图14-5 不同类型图形对象

14.2.1 通用函数

MATLAB有两个基本命令 `get` 和 `set` 用来处理图形对象。通过使用它们，可以给出和修改所有对象的属性值。

例14.3

(a) 创建一个图形窗口，它的句柄是 `gfp`：

```
gfp=figure;
```

先查看窗口使用的单位类型：

```
get(gfp,'units')
```

```
ans =
```

```
pixels
```

现在来改变窗口的大小和位置：

```
set(gfp,'Position',[100 500 400 400])
```

窗口的左下角移动到点 (100, 500) 处，大小设置为 400 × 400 像素点。

(b) 有些高级命令，如 `plot` 可以直接改变这些属性值。例如，例14.1中的曲面可以这样来画：

```
plot(x,y1,'LineWidth',5);
```

```
plot(x,y2,'LineStyle','+'); or plot(x,y2,'+');
```

(c) 如果想知道在 `set` 和 `gcf` 之间有哪些指针可以挑选使用；见命令集 155。

```
set(gcf,'Pointer')
```

```
[ crosshair | fullcrosshair | {arrow} | ibeam | watch | topl |  
topr | botl | botr | left | top | right | bottom | circle |  
cross | fleur | custom ]
```

命令集154 通用对象函数

```
set(h,prstr,alt,  
...)
```

对向量 `h` 指向的对象设置其属性 `prstr` 为值 `alt`，在它们之后还可以有几个属性。

```
set(h,a)
```

对向量 `h` 指向的对象设置其属性值。参数 `a` 是一个结构，结构中的域名是对象中可改变的属性名，域的值是赋予相应属性的新值。

```
set(h,pn,pv,  
...)
```

对向量 `h` 指向的对象设置其属性值。参数 `pn` 是一个包含可改变的属性名的细胞向量，`pv` 是一个包含设置新值的细胞向量。在它们之后还可以有几个属性的细胞向量。

```
set(h,pn,pv)
```

对 `h(i)` 指向的对象设置名为 `pn(j)` 属性的值为 `pv(i, j)`。参数 `h` 是一个包含对象句柄的向量，`pn` 是一个包含属性名的细胞向量，`pv` 是一个包含有 $m \times n$ 个值的细胞矩阵， m 是 `h` 的长度， n 是 `pn` 的长度。这个变量可以单独地设置对象属性。

```
set(h)
```

将句柄 `h` 的对象所有可改变的属性返回在一个结构中，结构的域名是这些属性的名字，域值是相应的属性值。

```
set(0,'Default')
```

检查对象属性的缺省值。返回一个结构，域名为

	'ObjectNamePropertyName',域值是相应的属性缺省值。 注意：只返回值为字符串的属性。
set(0,'DefaultObjectNamePropertyName')	返回一个细胞向量，它包含有对象名为ObjectName、属性名为PropertyName的缺省值。注意：只返回值为字符串的属性。
set(h,prstr)	返回一个细胞向量，包含句柄h的对象中属性名为prstr的所有可能值。
get(h)	列出句柄h的对象所有属性和相应的值。所有这些都返回在一个结构中，域名为属性的名字，域值为相应的属性值。
get(h,prstr)	返回句柄h的对象且属性名为prstr的当前值。
get(h,pn)	返回一个细胞矩阵，元素(i,j)的值是句柄h(i)的对象且属性名为pn(j)的属性值。参数pn是一个细胞向量，h是一个包含句柄的向量。
get(0,'Factory')	返回所有对象类型的所有属性的厂家设置值，用户可以改变它们的缺省值。厂家设置值返回在一个结构中，域名为'ObjectNamePropertyName'，域值为相应属性的厂家设置值。
get(0,'FactoryObjectNamePropertyName')	返回对象ObjectName中属性PropertyName的厂家设置值。
get(0,'Default')	检查有缺省值的对象。结果返回在一个结构中，域名为'ObjectNamePropertyName'，域值为相应属性的缺省值。
get(0,'DefaultObjectNamePropertyName')	返回对象ObjectName的属性PropertyName的缺省值。
allchild(h)	返回句柄h(i)的对象中所有子对象的句柄。隐含句柄也返回，见表14-2中'HandleVisibility'。如果h是一个标量句柄，则返回得到一个向量。如果h是一个向量，则返回一个细胞矩阵。
findobj	返回一个根对象和它的所有子对象的句柄向量。
findobj(h,prstr,alt,...)	如果给出参数prstr和alt，则返回一个向量，包含将属性prstr设置为alt的所有对象的句柄。在它们之后还可以写上几个属性。如果给出参数h，就限制在h中的对象和它们的子对象中搜索。如果没有给出prstr和alt，则返回一个包含向量h中的对象的子对象句柄向量。
findobj(h,'flat',prstr,alt,...)	同上，但是不检查对象的子对象。
findall(h)	和findobj(h)相似，而且还返回子对象的隐含句柄，见表14-2中'HandleVisibility'。
copyobj(h,p)	复制对象和它们的子对象。向量h中包含要复制的对象句柄，向量p中包含的它们父对象的句柄。这些父对象的类型必须和原来的父对象相同。函数返回一个包含新对象的句柄向量。
ishandle(h)	检查向量h中的元素是否是对象的句柄。函数返回元素是0

	和1的向量。如果h中的元素是一个对象的句柄，则在向量的相应位置为1；否则为0
<code>setupprop(h,prstr,val)</code>	用户可以对图和轴类型的对象定义新的属性。在字符串 prstr 中给出句柄为 <i>h</i> 的对象新属性，属性值设置为值 val ， prstr 和 val 可以为任何值。除非属性 prstr 已经存在，否则就新建。
<code>getupprop(h,prstr)</code>	获取句柄为 <i>h</i> 的图形对象或者轴对象中用户自定义的属性 prstr 的属性值。
<code>clrupprop(h,prstr)</code>	删除句柄为 <i>h</i> 的图形对象或者轴对象中用户自定义的属性 prstr 。
<code>handle2struct(h)</code>	将向量 h 中的句柄对象层次变换成一个结构，这个结构有如下的域： .type 对象的类型。 .properties 包含属性值的一个结构。 .children 一个结构矩阵，元素是句柄向量 h 中的对象的每个子对象。 .handle 对象的句柄。 .special 非文件式成员。
<code>struct2handle(s)</code>	命令 <code>handle2struct</code> 的逆操作：将结构 s 变换成对象的层次。

MATLAB 5有一个名为 `propedit` 的工具，是属性编辑器。这个编辑器比低级命令 `set` 和 `get` 用起来更方便。在建立图形用户界面时，这个工具显得更加有用，见命令集 170。

而且，还有三个函数可以获取当前图形对象的句柄。

命令集155 当前对象

<code>gcf</code>	获取当前图形对象的句柄。
<code>gca</code>	获取当前轴对象的句柄。
<code>gco(h)</code>	获取当前对象的句柄，也就是鼠标最后点击的对象。如果给出了 <i>h</i> ，则获取句柄为 <i>h</i> 的窗口中当前对象的句柄。

例14.4

下面的命令用来获取当前窗口和轴的属性：

```
get(gcf);           % 获取当前图形对象的属性。
get(gca);           % 获取当前轴对象的属性。
```

例14.5

也可用 `subplot` 命令移动对象。假设有两个子图，一个是牙刷，另一个是牙膏，如图4-6所示。要获取下图的位置，可以输入：

```
subplot(2,1,2);
get(gca,'Position')

ans =
    0.1300    0.1100    0.7750    0.3400
移动对象，可以输入：
set(gca,'Position',[0.15 0.49 0.775 0.34])
```

得到图14-7。这样轴对象就放在另一个图的上面。

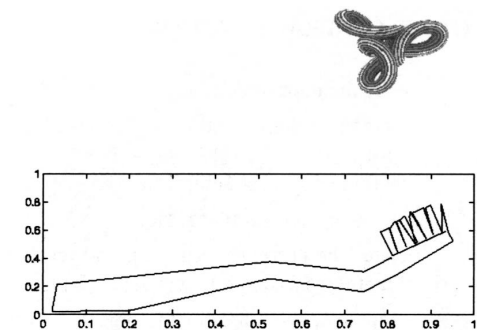


图14-6 subplot创建的两个轴对象

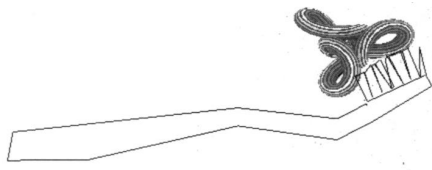


图14-7 将牙膏放在牙刷上的操作

还有一些函数用来删除对象、恢复缺省值和保存对象到文件中。

命令集156 其他通用函数

<code>clf</code>	清除当前窗口。
<code>clf reset</code>	清除当前窗口，除Position、Units、PaperPosition和PaperUnits外，重新设置所有图形对象的属性，见表14-6。
<code>cla</code>	删除当前坐标系。
<code>cla reset</code>	删除当前坐标系。除Position和Units外，重新设置所有轴对象的属性，见表14-10。
<code>rotate(h,ax,a,o)</code>	计算对象h中角a绕轴ax旋转的角度，可选参数o给出旋转的起始位置，缺省值为[0 0 0]。
<code>remapfig(pos, newpos,fig,h)</code>	将子对象移动到具有句柄fig的图形对象中。由pos规定内部区域大小的每个子对象移动到 newpos规定的新位置。参数pos和newpos都是这样的形式[left bottom width height]。一个轴对象的整个区域为[0 0 1 1]。如果没有指定fig，就使用缺省值。如果给出h，则只移动向量h中的子对象。
<code>reset(h)</code>	恢复具有句柄h的窗口(图形对象)或者坐标系(轴对象)的缺省值。
<code>delete(h)</code>	删除具有句柄h的对象。
<code>close(h)</code>	关闭当前窗口。如果给出h，则关闭h中的窗口。参数h可以是一个标量，也可以是包含有轴对象句柄的向量或者矩阵。
<code>close name</code>	关闭名为name的窗口。

<code>close all</code>	关闭所有没有隐含句柄的窗口，隐含句柄可通过属性 <code>HandleVisibility</code> 来设置，见表14-2。
<code>close all hidden</code>	关闭所有窗口，包括有隐含句柄的窗口。
<code>status=close(...)</code>	如果相应的窗口已关闭，则令 <code>status</code> 为1；否则为0。
<code>hgsave(h,filename)</code>	将具有句柄 <code>h</code> 的对象及其他的子对象保存到文件 <code>filename</code> 中。如果 <code>filename</code> 缺少后缀名，则加上 <code>.fig</code> 。
<code>hgload(filename)</code>	从文件 <code>filename</code> 中读取对象和它的子对象(如果有的话)。如果 <code>filename</code> 缺少后缀名，则加上 <code>.fig</code> 。最后返回读取对象的句柄。

14.2.2 共有属性和方法

由于继承，对于所有类型的对象有一些相同的属性和方法。然而这些当中有一部分对有些对象来说是没有意义的；见属性表和每种类型对象的方法。

表14-2 共有属性和方法

<code>ButtonDownFcn</code>	当对象被鼠标选择时，返回 MATLAB 回调字符串
<code>Children</code>	对象的所有子对象句柄的向量
<code>Clipping</code>	数据限幅模式，‘on’（缺省值）：只显示在坐标轴界限内的部分图形对象；‘off’：没有这个限制，也显示坐标轴外的部分
<code>CreateFcn</code>	决定用什么样的 M 文件或者 MATLAB 命令来创建对象。这必须用缺省值，例如创建一个图形对象： <code>set(0,'DefaultFigureCreateFcn',function)</code> ，其中字符串 <code>function</code> 是 M 文件名或者 MATLAB 命令
<code>DeleteFcn</code>	决定删除对象时运行的 M 文件或者 MATLAB 命令
<code>BusyAction</code>	MATLAB 处理对象的回调函数中断方式。如果将 <code>Interruptible</code> （见下面命令），设置为‘off’， <code>BusyAction</code> 可以有下面几种情况： ‘queue’（缺省值）将回调函数的中断请求放入一个挂起队列中直到对象的回调函数完成 ‘cancel’ 忽略其他回调函数所有可能的中断
<code>HandleVisibility</code>	对象的子对象列表中的对象句柄是否可访问；见 <code>Children</code> ‘on’（缺省值）总是可访问 ‘callback’ 只有回调函数或者调用回调函数的函数可以访问，这样防止用户从命令行中对对象进行修改 ‘off’ 不可访问
<code>HitTest</code>	对象是否被鼠标选中，也就是这个对象是否为当前对象。 <code>HitTest</code> 可以设置为‘on’（缺省值）或者‘off’。也可参见命令集 155 中的命令 <code>gco</code> 和表 14-7 中的轴对象属性 <code>CurrentObject</code> 。
<code>Interruptible</code>	指定对象回调字符串是否可中断。如果 <code>Interruptible</code> 是‘on’（缺省值），则该对象回调函数可以被其他回调中断。如果 <code>Interruptible</code> 是‘off’，则该对象回调函数不能被其他回调中断
<code>Parent</code>	对象的父对象句柄

(续)

Selected	对象是否被选中, 值可以为 'on' 或者 'off'
SelectionHighlight	在屏幕上选中的对象是否有四个边句柄和四个点句柄。值可以为 'on' (缺省值) 或者 'off'
Tag	用户用来标识对象的字符串, 在建立图形接口时这很有用
Type	只读对象类型的字符串
UserData	是一个矩阵, 包含有用户要在对象中保存的数据。矩阵不被对象本身使用
UIContextMenu	和对象相联的快捷菜单句柄。当在对象上按下鼠标右键时, MATLAB 显示出快捷菜单。见 14.2.8 节
Visible	控制对象在屏幕是否可见。值可以为 'on' (缺省值) 或者 'off'

14.2.3 根对象

上面提到屏幕是一个根对象, 它是所有其他对象的父对象。MATLAB 中可以用 set 和 get 设置和获取它的一些属性。根对象有如下的属性和方法。

表14-3 根对象的常用属性和方法

Automatic-FileUpdates	值可为 'on' 或者 'off', 它的使用是非文件式的
CallbackObject	获取正在执行的回调函数的对象句柄。如果没有回调函数在执行, 则返回 []。也可参见命令集 171 中的命令 gcob
Language	系统环境变量, 它的使用是非文件式的
CurrentFigure	获取当前图形对象的句柄。如果不存在图形对象, 则返回 []。也可参见命令集 155 中的命令 gcf
Diary	一个决定是否将命令窗口中的输入输出存入文件的字符串。值可为 'on' 或者 'off' (缺省值)
DiaryFile	一个表示保存输入输出文件名的字符串, 见 Diary
Echo	在文件执行时, 是否显示 M 文件的每一行。值可为 'on' 或者 'off' (缺省值)
ErrorMessage	包含 MATLAB 最近一次错误信息的字符串
Format	控制 MATLAB 数字显示格式的字符串。可选择的值有: 'short'、'shortE' (缺省值)、'shortG'、'long'、'longE'、'longG'、'bank'、'hex'、'+' 和 'rational'
FormatSpacing	控制 MATLAB 在命令窗口中输出间隔行的字符串。值可以为 'loose' (缺省值) 和 'compact'
PointerLocation	相对屏幕左下角指针的当前位置向量。
PointerWindow	含有鼠标指针的图形只读句柄。如果鼠标不在任何 MATLAB 图形窗口中, 则值为 0
Profile	决定是否使用 profile 工具的字符串。当程序执行时, 这个工具可以测出程序的不同部分运行所需的时间。值可以为 'on' 或者 'off' (缺省值)
ProfileFile	控制 profile 数据保存到什么样的文件中 (见上)
ProfileCount	是一个包含 profile 数据的 $n \times 1$ 向量。向量中的元素 k 是 M 文件在运行时在 k 行找到 profile 工具的次数
ProfileInterVal	在一定时间间隔内, profile 工具检查正在运行的脚本文件的行数
RecursionLimit	限定嵌套递归的次数。如果超过这个次数, MATLAB 会给出一个错误信息
ScreenDepth	整数, 指定以比特为单位的屏幕颜色深度

(续)

ScreenSize	有四个元素的只读位置向量 [left bottom width height]，指定屏幕大小尺寸
ShowHidden-	控制所有对象的所有句柄的可访问性，使各自的HandleVisibilty属性失效；见表14-2
Handles	属性值可以为 'on' 或者 'off' (缺省值)
Units	大小和位置的度量单位，可选下列单位： 'pixels' (标准) 屏幕像素 'normalized' 归一化坐标 'inches' 英寸 'centimeters' 厘米 'points' 排字机的点，等于0.353毫米 'characters' 字符
Parent	总是空矩阵[]，根对象没有父对象

MATLAB在X Windows环境下使用时，根对象还有一些另外的属性和方法。

表14-4 在X Windows环境下根对象的另外属性和方法

TerminalHideGraphCommand	文本串，从图形转换到命令模式时，隐含图形的命令序列
TerminalShowGraphCommand	文本串，从命令转换到图形模式时，显示图形的命令序列
TerminalOneWindow	终端有一个窗口，值为 'on' (缺省值)；终端有多个窗口，值为 'off'
TerminalDimensions	终端尺寸大小，以像素来衡量
TerminalProtocol	终端类型模式设置，可选的值为： 'none' 非终端模式，不连到X服务器 'x' 找到X服务器，X Windows模式 'tek401x' Tektronix 4010/4014仿真模式 'tek410x' Tektronix 4100/4105仿真模式

下列的常用属性和方法对根对象是没有意义的。

表14-5 根对象是没有意义的属性和方法

BusyAction	ButtonDownFcn	Clipping	CreateFcn
DeleteFcn	HandleVisibility	HitTest	Interruptible
Selected	SelectionHighlight	UIContextMenu	Visible

例14.6

根对象的句柄是0，可以象这样来调用：

```
scrsz = get(0,'ScreenSize')
```

```
Scrsz =  
      0      0  1152   900
```

命令返回得到屏幕大小的像素值。

14.2.4 图形对象

图形对象就是一个图形窗口，它的父对象是屏幕，即根对象。因此图形对象继承了根对

象的许多属性。属性可以在图形对象创建时修改，也可以用 `set` 命令来修改。

创建图形对象有下列几种方式。

命令集157 图形对象

<code>figure(prstr,alt, ...)</code>	设置属性 <code>prstr</code> 的值为 <code>alt</code> ，并返回一个图形对象的句柄。还可以再带几个属性及其相应的值。
<code>figure(f p)</code>	设置句柄为 <code>f p</code> 的窗口为当前图形对象。所有的图形命令都可用在当前图形中。
<code>refresh(f p)</code>	重新画句柄为 <code>f p</code> 的图形。如果只是给出 <code>refresh</code> ，则重画当前图形。
<code>drawnow</code>	强制 MATLAB 画一个对象。例如，在循环语句中用 <code>plot</code> 命令画对象时。如果不在每个 <code>plot</code> 命令后都给出 <code>drawnow</code> 命令，就只有在循环结束才画出对象。
<code>newplot</code>	打开一个新图形或者坐标系；是否清除当前图形或者轴取决于属性 <code>NextPlot</code> 的属性值。用 <code>help newplot</code> 可以得到更多信息。
<code>figname(str)</code>	得到以字符串 <code>str</code> 开始的下一个空闲图形的名字。这样就可以用来设置下面的 <code>Name</code> 属性。
<code>setp(h,cursor)</code>	设置句柄 <code>h</code> 的图形指针形状。指针形状由 <code>cursor</code> 给出，用 <code>help setp</code> 可以知道有哪些可选的形状。返回一个指针形状的小区矩阵，也可参见下面的各种 <code>Pointer</code> 属性。
<code>p=getp(h)</code>	返回得到一个包含句柄 <code>h</code> 的图形对象指针形状的小区矩阵 <code>p</code> 。这个小区矩阵可以用来设置指针形状： <code>setp(h,p{:})</code> 。也可参见下面的各种 <code>Pointer</code> 属性。

图形对象有下列一些常用属性和方法。

表14-6 图形对象的常用属性和方法

<code>BackingStore</code>	当图形窗口移动到屏幕前时，为了快速刷新屏幕，决定是否存储它的拷贝。它的值可以为 'on' (缺省值) 或者 'off'
<code>IntegerHandle</code>	控制句柄是可用的整数值还是不可用的浮点数值。如果是整数， <code>IntegerHandle</code> 设为 'on' (缺省值)，否则设为 'off'
<code>MenuBar</code>	控制 MATLAB 菜单在图形窗口的顶部显示。值可以为 'figure' (显示菜单，缺省值) 或者 'none' (隐藏菜单)
<code>Name</code>	图形窗口的标题，缺省值为空字符串
<code>NextPlot</code>	决定在图形窗口中新图的绘制方式： 'add' (缺省值) 在当前图形窗口中加上新的对象 'replace' 在画图前，将除位置属性外的所有图形对象属性重新设置为缺省值，并删除所有子对象。等同于命令 <code>clf reset</code> 'replace children' 删除所有子对象，但是不重新设置属性值。等同于命令 <code>clf</code>
<code>NumberTitle</code>	在图形标题中加上图形编号 如果 <code>NumberTitle</code> 设为 'on' (缺省值)，窗口标题是 "FigureNo.N: Name"，其中 Name (见上) 是一个字符串。如果 <code>NumberTitle</code> 设为 'off'，窗口标

(续)

	题仅仅为“Name”
Pointer	鼠标指针形状, 可选的形状有下列几种: ‘crosshair’, ‘arrow’ (缺省值), ‘watch’, ‘topl’, ‘topr’, ‘botl’, ‘botr’, ‘circle’, ‘cross’, ‘fleur’, ‘left’, ‘right’, ‘top’, ‘bottom’, ‘fullcrosshair’, ‘ibeam’ 和 ‘custom’
PointerShapeCData	16×16的矩阵表示用户自定义的鼠标指针。只有属性Pointer设为‘custom’
PointerShapeHot-Spot	时PointerShapeCData才可用。矩阵中1代表黑色, 2代表白色, NaN代表透明色 是一个有两个元素的向量, 表示 PointerShapeCData中的点 (行和列)。 这个点规定了鼠标指针的位置, 缺省值为 [1, 1]
Position	位置向量 [left bottom width height], 表示图形对象在屏幕上的位置和大小
Renderer	决定绘图的方式。有下列值可以选择: ‘painters’ 适合于画简单的图形 ‘zbuffer’ 当图形复杂时, 比用 ‘painters’ 的速度快而且精确, 但是需要更多的内存 ‘OpenGL’ 比 ‘painters’ 和 ‘zbuffer’ 更高级, 但是它需要软件或者硬件的支持。
RendererMode	作图使用的方法。如果 RendererMode设为 ‘auto’ (缺省值), MATLAB就会根据图形的复杂程度选择一种方法来作图。如果设为 ‘manual’, MATLAB就不改变作图方法。
Units	MATLAB中位置和大小的度量单位, 有下列值可供选择: ‘pixels’ (标准) 屏幕像素 ‘normalized’ 归一化坐标 ‘inches’ 英寸 ‘centimeters’ 厘米 ‘points’ 排字机的点, 等于0.353毫米 ‘characters’ 字符 单位的选择会影响属性CurrentPoint和Position
WindowStyle	图形窗口模式, 值可以为 ‘normal’ (缺省值) 或者 ‘modal’。后一种情况下所有的输入 (鼠标的点击和通过键盘输入的内容) 都被限制在窗口内, 直到将它设为 ‘normal’ 或者将Visible设为 ‘off’。模式窗口不显示任何菜单或者用户子菜单。MATLAB中可以用Control-C将所有的模式窗口转换成普通窗口
Clipping	不起作用
Parent	图形父对象的句柄, 总是 0
SelectionHighLight	不起作用

和图形对象有关的事件用下列的属性和方法来处理。

表14-7 图形对象的属性和方法驱动的事件

CloseRequestFcn	图形窗口关闭时执行的函数 (缺省是 ‘closereq’)
CurrentCharacter	键盘上最新按下的字符键
CurrentAxes	当前坐标轴的句柄, 也可参见命令集 155中命令gca
CurrentObject	当前对象的句柄, 也就是最近被选择的对象, 见下面的 CurrentPoint。也可参见命令集 155中命令gco

(续)

CurrentPoint	鼠标最近一次按下时所在的位置向量
KeyPressFcn	当鼠标指针在图形内，按下键时执行的回调函数，也就是脚本文件或者MATLAB命令
Resize	是否允许用户使用鼠标重新设定图形窗口的尺寸。它的值可以为‘ on ’ (缺省值)或者‘ off ’
ResizeFcn	当图形窗口重新设定尺寸时运行的函数
SelectionType	鼠标最新的选择类型。UNIX下有如下的选择：‘ normal ’ (左键，缺省值)，‘ extended ’ (shift—左键或者中间键)，‘ alternate ’ (control—左键或者右键)和‘ open ’ (双击)
WindowButtonDownFcn	当在图形窗口内鼠标按下一个键时运行的函数
WindowButtonDown— MotionFcn	当鼠标指针移动到图形窗口时运行的函数
WindowButtonUpFcn	当在图形窗口内释放开鼠标一个键时运行的函数

下面的属性和方法是用来控制图形对象及其子对象颜色的。

表14-8 图形对象及其子对象的颜色属性和方法

Color	图形窗口的背景色，一个RGB三元组或者MATLAB预定义的颜色名
Colormap	$m \times 3$ 的矩阵。包含 m 个不同的RGB三元组(颜色)来制成供多边形、曲面和图象对象使用的颜色表。矩阵缺省包含64种颜色，也可参见13.6节
Dithermap	$m \times 3$ 的矩阵(颜色表)。当在‘伪彩’(8位或者更低)屏幕上查看‘真彩’数据时使用的颜色表。缺省的64色颜色表也可用
DithermapMode	当在‘伪彩’(8位或者更低)屏幕上查看‘真彩’数据时，决定是人工还是自动转换颜色。如果设置为‘manual’(缺省值)，则颜色表Dithermap可用(见上)。如果设置为‘auto’，MATLAB会基于当前的颜色生成一个颜色表。‘auto’模式产生的效果要比‘manual’模式好，但是花费的时间要多一些
FixedColors	$m \times 3$ 的只读矩阵(颜色表)，Colormap没有给出颜色，如线条、文本和用户控制对象的颜色
MinColorMap	MATLAB存储到颜色表Colormap中最少的系统颜色数。应该等于Colormap中的行数，缺省值为64
ShareColors	MATLAB保存Colormap的方式，值可以为‘ on ’(缺省值)或者‘ off ’。想要Colormap能快速改变，最好用‘ off ’。见helpdesk可得更多信息

下列属性用来控制图形对象的输出打印。

表14-9 图形对象的输出属性

InvertHardCopy	是否互换图形对象的打印颜色。缺省值是‘ on ’，即图形的背景色为白色，而对象的颜色为黑色。‘ off ’正好和它相反
PaperUnits	MATLAB输出图形时使用的度量单位。有下列单位可用：‘ normalized ’，‘ inches ’(缺省值)，‘ centimeters ’和‘ points ’。度量单位的选择会影响属性PaperSize和PaperPosition
PaperOrientation	打印图形时纸张的方向。值可以为‘ portrait ’(垂直方向，缺省值)或者‘ landscape ’(水平方向)
PaperPosition	位置向量[<i>left bottom width height</i>]，代表页面上图形打印的位置

(续)

PaperPositionMode	图形输出在纸张上的位置是手动还是自动给出。如果PaperPositionMode设为 ' manual ' (缺省值)则MATLAB会使用PaperPosition(见上), 如果PaperPositionMode设为 ' auto ', 输出的位置就和在屏幕上看到的一样。
PaperSize	有两个元素的只读向量 [width height], 规定打印纸张的大小。
PaperType	打印图形纸张的类型, 有下列值可供选择: ' usletter ' (缺省)、' uslegal ', ' A0 ', ' A1 ', ' A2 ', ' A3 ', ' A4 ', ' A5 ', ' B0 ', ' B1 ', ' B2 ', ' B3 ', ' B4 ', ' B5 ', ' arch—A ', ' arch—B ', ' arch—C ', ' arch—D ', ' arc—E ', ' A ', ' B ', ' C ', ' D ', ' E ' 和 ' tabloid '。

例14.7

(a) 用给定的名字取代窗口的数字标题, 可以用如下命令:

```
fp = figure;
set(fp, 'NumberTitle', 'off');
set(fp, 'Name', 'ExampleWindow');
```

(b) 改变窗口的位置和大小:

```
set(fp, 'Position', [100 100 400 400]);
```

14.2.5 轴对象

轴对象可以在图形窗口中定义画图区域, 所以它的父对象是图形对象, 它的子对象是线条、图象、补片、曲面和文本对象。

轴对象的属性不仅可以定义位置, 还可以规定画图区域内的图形方向和图形大小。

命令集158 轴对象

axes(prstr, alt, ...)	创建轴对象, 根据缺省的属性值创建覆盖全部窗口的坐标轴。可选属性 prstr 的值设为 alt, 在它们之后还可以设置几个属性及其相应的值。返回得到坐标轴的句柄。
axes(h)	设置当前的轴对象为 h。

因为可以定义轴对象的大小, 所以在同一个图形对象中可以有多个轴对象; 和 subplot 命令相比。图形命令只作用在当前轴对象上。

轴对象有下列属性和方法。

表14-10 轴对象的常用属性和方法

Box	坐标轴是否有边框, 值可以为 ' on ' 或者 ' off ' (缺省值)。
CurrentPoint	包含最新按下鼠标位置的点的坐标轴对象上的 2×3 的矩阵 $\begin{pmatrix} x_b & y_b & z_b \\ x_f & y_f & z_f \end{pmatrix}$, 它定义了从坐标空间前面延伸到后面的一条三维直线。
DataAspectRatio	向量 [dx dy dz] 定义图形数据在 x、y 和 z 方向上各自的分量。
DataAspectRatioMode	MATLAB 是否自动计算数据在 x、y 和 z 方向上各自的分量。如果 DataAspectRatioMode 设为 ' auto ' (缺省值), MATLAB 会产生 DataAspectRatio; 如果设为 ' manual ', 就直接使用 DataAspectRatio。

(续)

	如果DataAspectRatio设定,那么DataAspectRatioMode就会自动地设为'manual'
DrawMode	画图象使用的模式,值可以为'normal'(缺省值)或者'fast'。后者更快,但会产生不正确的图。当父对象(图形对象)用Renderer命令设置为'zbuffer'时,该属性无意义
LineStyleOrder	指定线条和点的类型的字符串,用于在坐标轴上画多条线。例如:'-* : o'将通过点划线、点线和实线循环。LineStyleOrder缺省值为 '-',即所有的线是实线
LineWidth	X、Y和Z坐标轴的宽度,缺省值为0.5点
NextPlot	指定在坐标轴内画新图形的方式: 'add' (缺省值)使用当前的坐标轴 'replace' 删除窗口及其子对象,并重新设置所有属性值(除位置属性外)。等同于命令cla reset 'replace children'删除所有子对象,但是不重新设置属性值。等同于命令cla
PlotBoxAspectRatio	向量[px py pz]分别规定了由*Lim(见表14-13)定义的坐标框架在x,y和z方向的大小
PlotBoxAspect-RatioMode	MATLAB是否自动计算坐标边框在x,y和z方向的分量。如果设为缺省值'auto',MATLAB会生成PlotBoxAspectRatio;如果设为'manual',就直接使用PlotBoxAspectRatio。如果设置了PlotBoxAspectRatio,那么PlotBoxAspectRatioMode就自动地被设为'manual'
Projection	观察模式的设定。Projection设为缺省值'orthographic',则将数据集合内的平行线平行地画在屏幕上。这种模式适合于数值数据。另一种模式是'perspective',从远处看对象在屏幕上变得更小,所以这种模式适合于'real'对象,例如实心模型
Position	位置向量[left bottom width height]代表坐标轴在图形窗口中的位置和大小
Title	坐标轴标题文本对象的句柄
Units	大小和位置属性值的度量单位。有下列单位可供选择:'inches','centimeters','normalized'(缺省值),'points','pixels'和'characters'。单位的选择会影响Position属性,见上
View	这是一个旧属性,用CameraPosition、CameraUpVector和Camera-ViewAngle代替,见表14-12,也可参见命令view

例14.8

假设要建一个用带有限参数的subplot命令不能显示出的图像。用axes命令创建五个坐标轴。

```
[X,Y] = meshgrid(-2:0.3:2,-2:0.3:2);
ZS    = cos(X).*cos(Y);

                                     % 坐标轴位置
a(1) = axes('Position',[0.1 0.1 0.2 0.2]); % 左下
a(2) = axes('Position',[0.8 0.1 0.2 0.2]); % 右下
a(3) = axes('Position',[0.8 0.8 0.2 0.2]); % 右上
a(4) = axes('Position',[0.1 0.8 0.2 0.2]); % 左上
a(5) = axes('Position',[0.3 0.3 0.5 0.5]); % 中间

for i = 1:5,
    axes(a(i));                    % 在不同坐标轴内画图
```



```
surf(ZS);

if i == 1
    view(37.5,30);
elseif i == 2
    view(-37.5,70);
elseif i == 3
    view(10,30);
elseif i == 4
    view(0,-20);
end;
end;
```

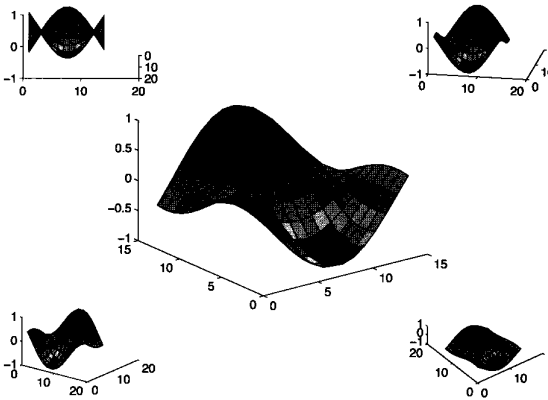


图14-8 用axes 画的不同坐标轴位置

注意，前两个数定义坐标轴的左下角位置，后两个数定义坐标轴的宽度和高度。在当前的窗口内将位置归一化到 [0, 1] 之间，结果如图 14-8 所示。这是从不同的角度看同一个曲面在坐标轴内的图形。

下列属性用来控制轴对象和它子对象的颜色。

表14-11 控制轴对象和它子对象颜色的属性和方法

AmbientLightColor	是一个RGB三元组或者是一种MATLAB中预定义的颜色，用来指定背景光的颜色。它用相同的数量影响所有对象。仅在轴对象中有一个光源对象时，它才有效
CLim	颜色界限向量[<i>cmin cmax</i>]，它确定将多边形和曲面对象的颜色数据 (CData) 映射到图形对象的颜色表 Colormap。 <i>cmin</i> 是映射到颜色表中的第一种颜色， <i>cmax</i> 映射到最后一种颜色。 CData 中的值线性映射到在 <i>cmin</i> 和 <i>cmax</i> 之间的值，小于 <i>cmin</i> 的值和大于 <i>cmax</i> 的值分别映射到第一种或者最后一种颜色上
CLimMode	MATLAB 是否自动计算 CLim。如果 CLimMode 设为缺省值 'auto'，则 MATLAB 在所有可见的图形对象中搜索出 Data 的最小和最大值来。如果 CLimMode 设为缺省值 'manual'，就直接使用 CLim。设置 CLim 就把 CLimMode 设为 'manual'
Color	坐标轴背景颜色；缺省值 'none' 表示坐标轴是透明的。颜色可以是一个 RGB 三元组或者一个 MATLAB 中预定义的颜色
ColorOrder	一个 <i>m</i> × 3 RGB 值的矩阵，用在 plot 和 plot3 中。如果用户没有指定颜色，这些函数就从这个矩阵中循环地使用颜色

MATLAB中使用‘camera’来指定观察坐标轴中的对象的角度。下列属性用来控制‘camera’。

表14-12 轴对象的观察点属性

CameraPosition	位置向量 $[x\ y\ z]$ ，规定在坐标轴内观察点的位置
CameraPositionMode	决定MATLAB是否自动计算观察点位置。如果CameraPositionMode设为缺省值‘auto’，MATLAB会生成CameraPositionMode。如果设为‘manual’，就直接使用CameraPositionMode。设置CameraPosition，就自动把CameraPositionMode设为‘manual’
CameraTarget	位置向量 $[x\ y\ z]$ ，规定在坐标轴内目标观察点的位置
CameraTargetMode	决定MATLAB是否会自动计算目标观察点的位置。如果CameraTargetMode设为缺省值‘auto’，MATLAB就将CameraTarget值设为轴对象的中心点；如果设为‘manual’，就直接使用CameraTarget。如设置CameraTarget，就自动把CameraTargetMode设为‘manual’
CameraUpVector	坐标轴内的位置向量 $[x\ y\ z]$ ，规定了观察点的旋转角度。在三维视图中缺省值为 $[0\ 0\ 1]$ ，这表示 z 正半轴上的点是向上的
CameraUpVectorMode	决定MATLAB是否会自动计算CameraUpVector。如果CameraUpVectorMode设为缺省值‘auto’，那么在三维视图中将CameraUpVector设为 $[0\ 0\ 1]$ 。在二维视图中将CameraUpVector设为 $[0\ 1\ 0]$ 。如果CameraUpVectorMode设为‘manual’，就直接使用CameraUpVector。设置CameraUpVector，就自动把CameraUpVectorMode设为‘manual’
CameraViewAngle	观察点的角度（0~180度），观察角度越大，对象看起来就越小
CameraViewAngleMode	决定MATLAB是否会自动计算观察点角度。如果CameraViewAngleMode设为缺省值‘auto’，MATLAB就会将CameraViewAngleMode设为能观察到所有对象的可能最小角度；如果设为‘manual’，就不改变设置。设置CameraViewAngle，就自动把CameraViewAngleMode设为‘manual’

轴对象和它子对象的轴、刻度、图注和格栅由下列属性控制，其中字符*代表X、Y和Z之一。

表14-13 控制轴的轴对象属性和方法

GridLineStyle	轴对象中格栅使用的线型，可选值有‘—’，‘——’，‘：’（缺省值），‘—.’和‘none’。和命令集132中的命令grid比较
Layer	坐标轴放置在轴的子对象上的位置。子对象必须是二维视图，或者view值为 $[0\ 90]$ 和DrawMode值为‘fast’的三维视图，见表14-10。Layer的值可以为‘bottom’（缺省值）或者‘top’，也就是分别将坐标轴放在画图区域的下面或上面
TickLength	向量 $[2Dlen\ 3Dlen]$ ，分别代表了在二维和三维视图中的坐标轴刻度标记的长度。TickLength的值是坐标轴的最长长度分之一
TickDir	坐标轴刻度标记的指向。二维视图的缺省值是‘on’，表示刻度标记从坐标轴线向内；三维视图的缺省值是‘out’，表示刻度标记从坐标轴线向外
TickDirMode	MATLAB是否自动设置TickDir。如果TickDirMode设为缺省值‘auto’，就设置TickDir；如果设为‘manual’，就直接使用TickDir。设置TickDir，就自动把TickDirMode设为‘manual’
XAxisLocation	刻度和数字文本放在 x 轴上的位置，值可以为‘bottom’（缺省值）或者‘top’，表示将刻度和文本放在图的下面或者图的上面
YAxisLocation	刻度和文本（数字）放在 y 轴上的位置，值可以为‘left’（缺省值）或者‘right’，表示将刻度和文本放在图的左边或者图的右边

(续)

*Color	一个RGB三元组或者MATLAB中一种预定义的颜色,表示*轴的颜色。在*方向的刻度标记、数字文本和格栅线都是这种颜色,缺省为'white'(白色)
*Dir	表示*轴的方向,值可以为'normal'(缺省值)或者颠倒方向的'reverse'
*Grid	是否在*方向画格栅线,值可以为'on'或者'off'(缺省值)。使用命令grid来画或清除在各方向上的格栅线,见命令集132
*Label	*轴标志文本对象的句柄
*Lim	向量[min max],表示*轴的最小和最大值
*LimMode	MATLAB是否自动计算*Lim。如果*LimMode设为缺省值'auto',就自动计算*Lim;如果设为'manual',就直接使用*Lim。设置*Lim,就自动把*LimMode设为'manual'
*Scale	表示*轴的线性换算或者对数换算,值可以为'linear'或者'log'
*Tick	是一个元素值单调递增的向量,表示刻度标记画在*轴上的位置
*TickLabel	数字文本字符串矩阵,用在*轴上标出刻度标记。如果矩阵中字符串不够,就重复使用
*TickLabelMode	MATLAB是否自动计算*TickLabel。如果*TickLabelMode设为缺省值'auto',就设置*TickLabel;如果设为'manual',就直接使用*TickLabel。设置*TickLabel,就自动把*TickLabelMode设为'manual'
*TickMode	MATLAB是否自动计算*Tick。如果*TickMode设为缺省值'auto',就设置*Tick;如果设为'manual',就直接使用*Tick。设置*Tick,就自动把*TickMode设为'manual'

轴对象中坐标轴上的文本和数字显示由下列属性控制。

表14-14 控制坐标轴文本的属性

FontAngle	字符串表示坐标轴文本字体的角度,设定值可以为:'normal'(缺省值)、'italic'和'oblique'
FontName	字符串表示使用的字体
FontSize	字体的大小,使用FontUnits中的单位,见下面。
FontUnits	FontSize中使用的字体大小单位,可以用下列选择:'inches','centimeters','normalized','points'(缺省值)和'pixels'
FontWeight	坐标轴文本加黑,设置值可以为:'light','normal'(缺省值)、'demi'和'bold'

坐标轴的区间可以用给定数据的最大值和最小值,每个刻度之间单位可以用缺省值1、2或者5。可以通过属性'XTick'和'YTick'对刻度和格栅线进行操作。

例14.9

下面的程序给出如何来改变坐标轴刻度标记和格栅线。

```
x = [1 3 7];           % 生成x向量
y = [6 9 2];           % 和y向量

s1 = subplot(2,2,1);    % 左上角建一个坐标系
plot(x,y);
grid;                   % 使用缺省的格栅线
title('Default');
```

```

s2 = subplot(2,2,2);           % 右上角建一个坐标系
plot(x,y);
set(s2,'XTick',x);             % 改变x轴标记
set(s2,'XGrid','on');          % 画x轴的格栅线
title('X scale manipulated');

s3 = subplot(2,2,3);           % 左下角建一个坐标系
plot(x,y);
set(s3,'YTick',[2 6 9]);       % 改变y轴标记
set(s3,'YGrid','on');          % 画y轴的格栅线
set(s3,'GridLineStyle','-');   % 使用虚线格栅
title('Y scale manipulated');

s4 = subplot(2,2,4);           % 右下角建一个坐标系
plot(x,y);
set(s4,'XTick',x);             % 改变xy轴标记
set(s4,'YTick',[2 6 9]);       % 画xy轴的格栅线
grid;                           % 画xy轴的格栅线
title('Both scales manipulated');

```

得到的结果如图 14-9 所示。

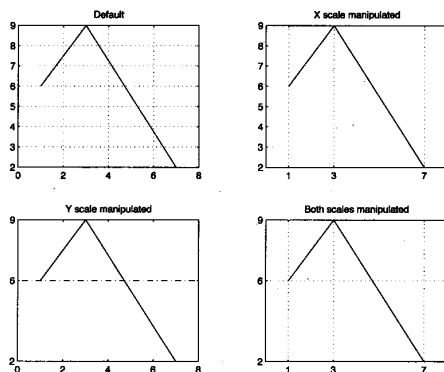


图14-9 不同的刻度标记和格栅的坐标轴

14.2.6 用户控制对象

用户控制对象用来建立图形用户界面，即 GUI。它们可以用来创建在 GUI 中的所有控制对象，比如，按钮、滑标和无线按钮等。正是因为此，在程序中使用用户控制对象使得程序处理起来更容易。

有关 GUI 的更多信息可参见 14.3 节。

命令集 159 用户控制对象

```

uicontrol(fp,
prstr,val,...)

```

在句柄为 *fp* 的图形窗口中创建控制窗口，返回得到这个控制窗口的句柄。如果没有给出 *fp*，则在当前图形窗口中创建。用 *prstr* 和 *alt* 来设置属性及其相应的值：类型(按钮、滑标等)、大小、位置等等。可以同时多个属

```
[outstr,pos]=
textwrap(h,instr)

popupstr(h)
```

性进行设置。属性 ' Callback ' 控制用户选择结果。将文本粘贴在 `instr` 句柄 `h` 的用户控制对象上。参数 `instr` 是一个细胞向量，每个细胞是一行文本内容，`outstr` 是粘贴的文本。向量 `pos` 包含的是用户控制对象的缺省位置和大小。句柄 `h` 的弹出式菜单中当前被选中的项文本)。弹出式菜单是将属性 `Style` 设为 ' `popupmenu` ' 的一个用户控制对象，见下面。

用户控制对象有下列属性和方法。

表14-15 用户控制对象的属性和方法

BackgroundColor	是一个RGB三元组或MATLAB一个预定义的颜色，来设置对象的背景色
Callback	当用户激活用户控制对象如点击对象时运行的回调函数。' frame ' 和 ' text ' 类型的用户控制对象的回调函数不可交互选择
CData	是一个 RGB 值的 $m \times n \times 3$ 矩阵，表示在 ' pushbutton ' 或 ' togglebutton ' 上显示的真彩图像
Enable	用鼠标点击用户控制对象时用户对象的动作，有下列值可以选择： ' on ' (缺省值)表示执行 ' Callback ' ' inactive ' 表示执行 ' ButtonDownFcn ' ' off ' 表示执行 ' ButtonDownFcn '，同时对象的标题 (见 String) 变得模糊不清
Extent	只读向量 [0 0 width height] 表示文本 String 的大小
ForegroundColor	一个RGB三元组或MATLAB一个预定义的颜色，设置文本 String 的颜色，见下面。缺省的文本颜色为黑色
Horizontal Alignment	标志文本 String 的水平排列。值可以为：' left '、' center ' (缺省值) 和 ' right '。
ListboxTop	用在 ' listbox ' 对象中的一个数值，表示将细胞向量 String 中以这个数值为下标的字符串放在列表框的最上面
Max	表示属性 Value (见下面) 的最大值，取决于控制对象的类型： <ul style="list-style-type: none"> 当 ' radiobutton '、' checkbox ' 和 ' togglebutton ' 处于 ' on ' 状态时，将它们的 Value 设定为 Max 对于 ' slider '，它就是可选的最大值 对于 ' edit ' Max，当 Min>1 时，可编辑文本框是多行文本 对于 ' listbox ' Max，当 Min>1 时，表示列表框中有多个选项可以选择 对于对象 ' frame '、' popupmenu ' 和 ' text '，它是无效属性
Min	表示属性 Value (见下) 的最小值，取决于控制对象的类型： <ul style="list-style-type: none"> 当 ' radiobutton '、' checkbox ' 和 ' togglebutton ' 处于 ' off ' 状态时，将它们的 Value 设定为 Min 对于 ' slider ' 它就是可选的最小值 对于 ' edit ' Max，当 Min>1 时，可编辑文本框是多行文本 对于 ' listbox ' Max，当 Min>1 时，表示列表框中有多个选项可以选择 对于对象 ' frame '、' popupmenu ' 和 ' text '，它是无效属性。
Position	位置向量 [left bottom width height] 表示对象在屏幕上的位置

(续)

String	用来设置对象文本的字符串。对于 'popupmenu' 来说, 如果String是一个细胞向量, 字符串矩阵或者用垂直条 ' ' 分隔用括号括起来的字符串如 ('popupmenu', 'listbox') 或 '\n' ('edit', 'text')。可以用这个字符串来设置 'listbox'、'edit' 和 'text' 的几个参数和文本的线条
Style	定义对象类型的字符串。有下列值可以选择: 'pushbutton' (缺省值)、'radiobutton'、'checkbox'、'edit' (可编辑文本)、'text'、'slider'、'frame'、'listbox'、'popupmenu' 和 'togglebutton' (开/关按钮)
SliderStep	向量 [minstep maxstep] 分别表示每次单击滑标箭头移动的最小范围和单击滑标移动的最大范围值。缺省值为 [0.01 0.1]
TooltipString	用户将鼠标移动到控制对象上显示的提示字符串
Units	位置属性值的单位, 有下列单位可以选用: 'inches'、'centimeters'、'normalized'、'points'、'pixels' (缺省值) 和 'characters'。单位的选择会影响属性Extent和Position, 见上面。
Value	不同类型对象的值是不同的 <ul style="list-style-type: none"> 对于 'radiobutton'、'checkbox' 和 'togglebutton' 分别参见上面的Max和Min属性 对于 'slider', 它就是当前值 对于 'popupmen', 它是当前可选项的个数 对于 'listbox', 它是包含当前可选项的向量 对于其他的用户控制对象, 这个属性是无效的
Children	用户对象没有子对象, 总是空矩阵 []
Clipping	无效属性
HitTest	无效属性

用户控制对象中的文本显示由下列属性控制。

表14-16 用户控制对象中的文本控制属性

FontAngle	FontName	FontSize	FontUnits	FontWeight
-----------	----------	----------	-----------	------------

在表14-13中对它们进行了详细描述。

14.2.7 用户菜单对象

用户菜单对象和用户控制对象一样也是用来建立图形用户界面的。可以用它来建立下拉式菜单, 将它放置在图形窗口的顶部。

命令集160 用户菜单对象

`uimenu(p,prstr,alt)` 在句柄 p 的图形窗口顶部建立一个下拉式菜单, 返回这个对象的句柄。如果没有指定 p , 就使用当前窗口句柄。如果句柄 p 已经和一个菜单相关联, 则建立子菜单, 所以可以建立多级菜单。菜单的属性和它的名字等可以由参数 **prstr** 和 **alt** 给出, 将属性 **prstr** 设置为值 **alt**。可以对多个属性及其属性值进行设置。

<code>makemenu(h,mencho, calls, tags)</code>	在句柄 h 的图形对象中创建一个菜单结构。参数 mencho 是包含菜单选项的字符串矩阵， calls 是一个字符串矩阵，包含的是当用户在菜单中选择菜单选项时 MATLAB 要执行的命令。如果给出字符串矩阵 tags ，就设置作为菜单一部分的菜单对象的属性 Tag(见表 14-2)。命令 <code>makemenu</code> 返回创建对象的句柄向量，用 <code>help makemen</code> 可以得到更多信息。也可以用工具 <code>menuedit</code> 来建立菜单，参见 14.3.5 节。
<code>umtoggle(h)</code>	句柄 h 的用户菜单对象的选择状态。如果选中用户菜单对象，则返回 1；否则返回 0。也可参见下面的用户菜单属性 <code>Checked</code> 。
<code>winmenu(h)</code>	创建句柄 h 的图形窗口中的菜单 Window 的子菜单。如果没有给出 h ，则用当前图形窗口的句柄。菜单 Window 的 Tag 属性必须设置为 'winmenu'，见表 14-2。用 <code>help winmenu</code> 可以得到更多信息。

用户菜单对象有下列属性和方法。

表 14-17 用户菜单对象的属性和方法

Accelerator	指定菜单项的快捷键，用户可以用 Control—Accelerator 来选择菜单项；Macintosh 系统，用 Command—Accelerator 来选择。这对有回调函数但是没有子菜单的选项才有效
Callback	表示当用户选中菜单项时运行的回调函数
Checked	菜单项的选中状态，值可以为 'on' 或者 'off' (缺省值)
Enable	菜单使能状态，值可以为 'on' (缺省值) 或者 'off'。当设为 'off' 时，菜单项将变成灰色
ForegroundColor	用户菜单的前景(文本)色，是一个 RGB 三元组或者 MATLAB 预定义的颜色。 注意：这个属性只能使用在 X Windows 系统中
Label	含有菜单项名字的文本串，用 ' & ' 放在名字前会将名字中的一个字符加上下划线。这样就可以用键盘上相对应于带下划线的字符的键来激活菜单项
Position	表示菜单行中的相对位置，值 1 表示最左边，在菜单中表示最上面
Separator	分割符，表示在菜单项上是否画一个分割线。值为 'on' 或者 'off' (缺省值)
ButtonDownFcn	无效属性
Children	包含子菜单的句柄向量
Clipping	无效属性
Selected	无效属性
SelectionHightLight	无效属性
UIContextMenu	无效属性

14.2.8 用户快捷菜单对象

Uicontextmenu 对象可以用建立快捷菜单，这是 MATLAB 5.2 中介绍的一种新型菜单。当

鼠标指针停在对象上时，如用户按下鼠标右键（在Macintosh系统按下Control—单击）就可以显示出这些菜单来。从这个菜单中用户可以进行选择，快捷菜单总是和其他的对象联系在一起，参见表14-2中的属性UIContextMenu。

命令集161 用户快捷菜单对象

`uicontextmenu` 建立快捷菜单，将属性`prstr`设置为`alt`，返回菜单的句柄。
(`prstr,alt, ...`) 可以对多个属性进行设置。

Uicontextmenu对象有下列属性和方法。

表14-18 Uicontextmenu对象的属性和方法

Callback	当在对象上按下鼠标右键（在Macintosh系统中按下Control—单击）时运行的回调函数
Children	包含用户菜单对象句柄的向量，也就是快捷菜单中的菜单项
ButtonDownFcn	无效属性
Clipping	无效属性
HitTest	无效属性
Selected	无效属性
SelectionHighlight	无效属性
UIContextMenu	无效属性

例14.10

在这个例子中将建立曲面对象的快捷菜单，用户可以在这个菜单中改变曲面的线型。

```
% 画一个球体曲面
figure(1)
sphere;
sp = findobj(1,'Type','surface'); % 获取曲面指针

% 定义菜单选项，即回调函数。
cb1 = ['set(sp, 'LineStyle', 'none')'];
cb2 = ['set(sp, 'LineStyle', '--')'];
cb3 = ['set(sp, 'LineStyle', ':')'];
cb4 = ['set(sp, 'LineStyle', '-')'];

% 定义快捷菜单
cmenu = uicontextmenu;
set(sp, 'UIContextMenu', cmenu)
menp = uimenu(cmenu, 'Label', 'Linetypes');
item1 = uimenu(menp, 'Label', 'none', 'Callback', cb1);
item2 = uimenu(menp, 'Label', 'dashed', 'Callback', cb2);
item3 = uimenu(menp, 'Label', 'dotted', 'Callback', cb3);
item4 = uimenu(menp, 'Label', 'solid', 'Callback', cb4);
```

用户按下鼠标右键（Macintosh系统中为Control—单击），图14-10中的菜单就会显示出来。

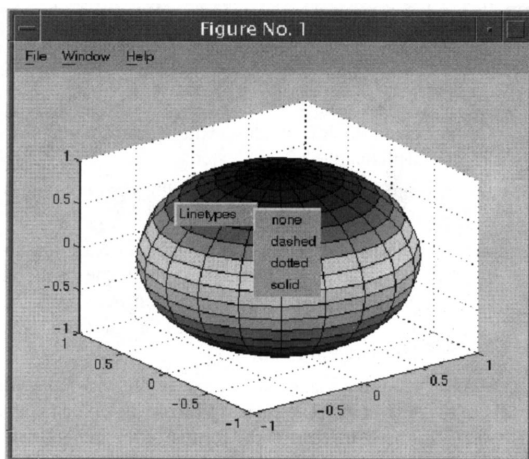


图14-10 有快捷菜单的曲面对象

14.2.9 图像对象

MATLAB中也可以显示图像。MATLAB中的图像是由矩阵来定义的，矩阵的元素相对应于图像中的点，元素的值相对应于点的颜色，所以 MATLAB中的图像是‘位图’类型。

命令 `image` 既是一个高级命令又是一个低级命令，可以用来在屏幕上画出由矩阵定义的图像。相应的捕获图形窗口图像的函数是命令 `capture`。然而要注意的是通常命令 `capture` 和 `image` 给出的结果要比 MATLAB 内部表示的要低级一些，而且需要较多的内存。

命令集162 图像对象

<code>image(C)</code>	把矩阵 <code>C</code> 作为一个图像画出， <code>C</code> 中的每个元素是图像的一个点，元素的值表示颜色。如果没有给出 <code>C</code> ，就用缺省值。最后返回得到图像的句柄。
<code>image(x,y,C)</code>	把矩阵 <code>C</code> 作为一个图像画出，向量 <code>x</code> 和 <code>y</code> 分别表示 <code>x</code> 、 <code>y</code> 轴的最大和最小值。
<code>image(prstr,alt, ...)</code>	用设定值 <code>prstr</code> 和 <code>alt</code> 来创建一个新图像，可以同时为多个属性进行设定。这些属性可以用 <code>set</code> 命令来修改。
<code>imagesc(...)</code>	和 <code>image</code> 相同，但是标定数据以使用全部的颜色表。
<code>capture(h)</code>	建立一个与句柄 <code>h</code> 窗口有相同内容的新窗口。
<code>[C,Cm]=capture</code>	返回用相应颜色表 <code>Cm</code> 的图像矩阵 <code>C</code> ，但是并不画出图像。
<code>imfinfo(filename, fmt)</code>	返回得到关于图像文件信息的结构。字符串 <code>filename</code> 是文件名， <code>fmt</code> 是文件的扩展名。扩展名可以为 ‘ <code>bmp</code> ’、‘ <code>hdf</code> ’、‘ <code>jpg</code> ’、‘ <code>jpeg</code> ’、‘ <code>pcx</code> ’、‘ <code>tif</code> ’、‘ <code>tiff</code> ’ 或者 ‘ <code>xwd</code> ’。见 <code>helpdesk</code> 可得更多信息。
<code>[A, Ftab]=imread</code> <code>(filename,fmt,idx)</code>	从图像文件 <code>filename</code> 中将读出图像到矩阵 <code>A</code> 中，矩阵 <code>Ftab</code> 是图像的颜色表。字符串 <code>fmt</code> 是图像文件的扩展名，见上

```
imwrite(A,Ftab,
filename,fmt,
prstr,alt)
```

面。如果没有给出扩展名，MATLAB就试着匹配。如果文件中有多个图像(只对扩展名为‘tiff’和‘hdf’的文件)，标量 *idx* 表示从文件读出的图像的个数。见 helpdesk 可得更多信息。

同上，不同的是将图像写入到文件中。如果没有文件格式 **fmt**，文件必须要有 **fmt** 作为扩展名。用扩展名‘hdf’、‘jpeg’和‘tiff’可以设定图像文件的某些属性；将属性 **prstr** 设为 **alt**。见 helpdesk 可得更多信息。

图P-7中显示的地球图像，就是用 **image** 命令绘制的。命令 **colormap(map)** 给出了正确的颜色。图像对象有下列属性和方法。

表14-19 图像对象的属性和方法

CData	指定图像矩阵中的元素颜色值。数据格式和补片中顶点的颜色格式相同
CDataMapping	见表14-21
EraseMode	见表14-20
Xdata	是一个大小为 $1 \times \text{size}(\text{CData}, 2)$ 的向量，表示图像 <i>x</i> 的坐标
YData	是一个大小为 $1 \times \text{size}(\text{CData}, 3)$ 的向量，表示图像 <i>y</i> 的坐标
Children	空矩阵[]，图像对象没有子对象

例14.11

不带参数的 **image** 命令生成倒立的图像，如图14-11所示。

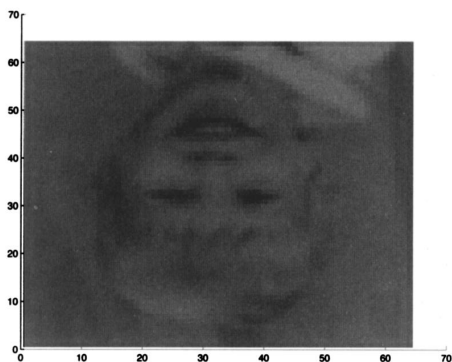


图14-11 不带参数的 **image** 命令生成的图像

14.2.10 线条对象

一般不用低级命令 **line** 来直接创建线条对象，而是用高级命令，如 **plot**。然而有时也用 **line** 命令来操作已有的线条和画新线条。和高级命令 **plot** 比较，**line** 命令只是画一条线，而 **plot** 命令除了画线条外，还可以做一些其他的事情，如替换坐标轴。

命令集163 线条对象

<code>line(x,y)</code>	在当前图形中画向量 <code>x</code> 和 <code>y</code> 的直线。如果 <code>x</code> 和 <code>y</code> 是同样大小的矩阵，则每列画一条直线。返回得到线条对象的句柄。
<code>line(x,y,z)</code>	类似上个命令在三维坐标系中画一条直线。
<code>line(prstr,alt, ...)</code>	画一条直线，将属性 <code>prstr</code> 的值设为 <code>alt</code> 。同时可以对多个属性值进行设定。

线条对象有下列属性和方法。

表14-20 线条对象的属性和方法

Color	线条颜色，一个RGB三元组或一个MATLAB预定义的颜色
EraseMode	消除和重画模式 'normal' (缺省值)重画影响显示的整个区域，这是最精确的、也是最慢的一种模式 'none' 当移动或删除线条时该线不会被消除 'xor' 用图形的背景颜色执行异或XOR运算，画出和消除线条。当消除线条时不影响背景色，但是线条颜色取决于下面的图形 'background' 通过在图形背景色中重画线来消除线条。这样会影响背景色，但是线条颜色取决于线下图形
LineStyle	线条类型，有下列可选值：'—' (缺省值)、'——'、': ':'、'-'和'none'。
LineWidth	以点为单位的线宽，缺省值是0.5点
Marker	标记数据的记号类型，有下列可选值：'+','o','*','.'、'x'、'square'、'diamond'、'^'、'v'、'>'、'<'、'pentagram'、'hexagram'和'none' (缺省值)
MarkerSize	以点为单位的记号大小，缺省值是点。注意，MATLAB在指定大小的1/3内标记记号
MarkerEdgeColor	没有填充的点颜色或填充点的边颜色，值可以为一个 RGB三元组、一个MATLAB预定义的颜色、'none' 或 'auto' (缺省值，给出颜色Color；见上面)
MarkerFaceColor	填充点的填充色，值可以为一个 RGB三元组、一个MATLAB预定义的颜色、'none' 或 'auto' (缺省值，给出颜色Color；见上面)
XData	x轴坐标的向量
YData	y轴坐标的向量
ZData	z轴坐标的向量
Children	空矩阵[]，线条对象没有子对象

14.2.11 补片对象

补片对象是一个填充的多边形定义的区域。根据维数，多边形的角由两个或三个向量 `x`、`y`和`z`定义，并且按向量元素的顺序来定义。多边形用 `patch`命令和指定的颜色进行填充。

命令集164 补片对象

<code>patch(x y c)</code>	在当前坐标系内画出向量和 <code>y</code> 定义的多边形。参数是多边形的颜色，可以是数值、向量或矩阵，见 <code>Data</code> 和下面的 <code>FaceVertexCData</code> 。
---------------------------	--

如果 x 和 y 是矩阵，则每列画一个多边形。命令返回补片对象的句柄。

`patch(x,y,z,c)` 在三维坐标系内画出多边形，和在二维坐标系内相似。
`patch(prstr,alt, ...)` 创建一个多边形，将它的属性`prstr`的值设为`alt`，可以同时为多个属性进行设定。

例14.12

下面是球体和球体补片对象的例子。

```
sphere(10);           % 画一个有121个角的球体
x=[ -2 -2 2 2];       % 定义多边形的x轴位置
y=[ -2 2 2 -2];       % 定义多边形的y轴位置
z=[ -2 -2 -2 2];      % 定义多边形的z轴位置
c=[ -2 -1 1 2];       % 定义多边形颜色
pl=patch(x, y, z, c); % 画多边形，并保存其句柄在pl中
```

结果如图14-12所示。

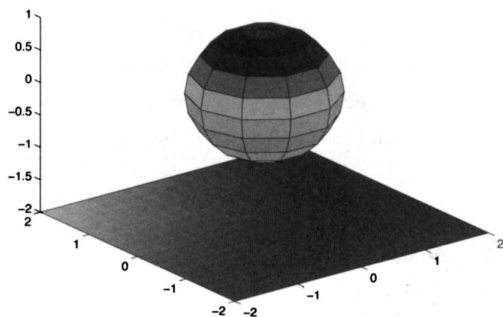


图14-12 用`patch`命令画的球体和多边形

补片对象的外观和属性可以用`set`和`get`来重新设定和获取。补片对象有下列属性和方法。

表14-21 补片对象的属性和方法

Cdata	<p>多边形的颜色，也可参见 CData Mapping</p> <ul style="list-style-type: none"> • 一个RGB三元组，MATLAB预定义的颜色之一或是父对象的颜色表中的下标值 • RGB值的$m \times 3$矩阵或给出多边形每个面的颜色值的向量 • RGB值的$m \times n \times 3$矩阵或给出多边形每个角的颜色值的矩阵 <p>在伪色屏幕中需要抖动 RGB三元组，这有利于父对象图形的属性 Colormap和Dithermap，见表14-8。如果CData含有NaN，那么MATLAB将不对多边形表面着色，也可参见 FaceVertexCData。</p>
CDataMapping	<p>Cdata或FaceVertexCData中 'color index' (颜色下标)可能的情况。下标是整数，则直接或间接表示父图形对象的颜色表colorMap(见表14-18)中的一个位置：'scaled' (缺省值) MATLAB线性地将颜色数据转换成颜色填充父图形对象中的Clim表示的间隔</p>

(续)

	<p>‘direct’下标直接表示父图形对象的颜色表ColorMap中的颜色。超出颜色表的下标映射到第一个下标(如果值小于1)或者最后一个下标</p>
FaceVertexCData	<p>显示多边形表面或者顶点颜色</p> <ul style="list-style-type: none"> • 一个RGB三元组或一个数值表示多边形取得一种颜色,这个数值表示父对象的颜色表的下标(见CDataMapping) • 一个$m \times 3$矩阵,m是Faces中的行数,元素是RGB三元组或$m \times 1$的向量(见CDataMapping)给出多边形的每个顶点的颜色值也可参见Cdata
EdgeColor	<p>显示多边形边缘的颜色</p> <ul style="list-style-type: none"> • 一个RGB三元组或一个MATLAB预定义的颜色,表示所有的边缘线是单一色,缺省值是‘black’(黑色) • ‘none’表示不画边缘线 • ‘flat’表示边缘线颜色和补片顶点的颜色相同 • ‘interp’表示边缘线颜色由补片属性CData或FaceVertexCData的值线性插值得到
EraseMode	见表14-20
FaceColor	<p>多边形表面颜色</p> <ul style="list-style-type: none"> • 一个RGB三元组(缺省值) • ‘none’表示不画表面,但画出边缘 • ‘flat’表示表面颜色和第一个顶点颜色相同 • ‘interp’表示表面颜色通过顶点颜色双线性插值得到
Faces	<p>$m \times n$矩阵,表示顶点的连接状态,每个顶点定义m个曲面。矩阵中的每行表示每个曲面的连接状态,和NaN不同的行中元素的个数表示顶点的个数。见elpdesk可得更多信息</p>
LineStyle	轮廓的线型,见表14-20
LineWidth	轮廓线的宽度,见表14-20
Marker	多边形顶点的记号类型,见表14-20
MarkerEdgeColor	见表14-20,缺省值‘auto’表示给出颜色EdgeColor,另一个属性值是非文件式值‘flat’
MarkerFaceColor	见表14-20,属性值‘auto’表示表面颜色是父轴对象的颜色Color,‘none’表示表面颜色是父图形对象的颜色,还有一个非文件式属性‘flat’
MarkerSize	见表14-20
Vertices	$m \times 3$ 矩阵,表示多边形顶点的 x 、 y 、 z 坐标,也可参见Faces
XData	一个向量或者矩阵,表示顶点的 x 坐标。如果Xdata是一个矩阵,则每列表示多边形中一个单独表面的 x 坐标。这种情况下,要求XData、YData和ZData有相同的维数
YData	和XData相同,但是是 y 坐标
ZData	和XData相同,但是是 z 坐标
Children	空矩阵,补片对象没有子对象

MATLAB可以通过创建光源对象来照明多边形;见表4-26。有下列属性和方法来控制光效果。

表14-22 控制补片对象光效果的属性和方法

FaceLighting	<p>计算多边形表面光效果的算法</p> <ul style="list-style-type: none"> • ‘none’ (缺省值)没有光效果 • ‘flat’ 光从同一个方向照射整个表面。比如,这种方法可用来研究平的表面
--------------	---

(续)

	'gourand'	表示光效果是对整个表面进行插值得到的。比如，这种方法可用来研究曲面
	'phong'	和'gourand'相似，可给出更好的效果，但是需要花费较多时间
EdgeLighting		和FaceLighting相似，但是用于多边形的顶点
BackFaceLighting		当将远离观察点的顶点表面归一化时，多边形的表面光效果；见表 14-12
	'unlit'	表面没有光照
	'lit'	用通常方式照射表面
	'reverse lit' (缺省值)	照射表面，就象表面的法线指向观察点。这对分清对象的内部和外部表面很有用
AmbientStrength		0~1之间的数值，表示多边形背景光照的强度（也可称为周围光）。也可参见轴对象属性 AmbientColor
DiffuseStrength		0~1之间的数值，表示从多边形上漫射光的强度。阴暗对象的属性值较大
SpecularStrength		0~1之间的数值，表示从多边形上反射光的强度。闪耀对象的属性值较大
SpecularExponent		1的数值，表示多边形的'mirror-likeness'。数值通常在1~500之间，大多数情况值在5~20之间
SpecularColor-Reflectance		0~1之间的数值，表示光对象和多边形的颜色对'镜'光颜色的影响程度，数值0表示都受二者影响；数值1表示只受光对象影响。也可参见轴对象属性 AmbientLightColor；表14-11
VertexNormals		$m \times 3$ 矩阵，包含多边形 m 个顶点的曲面法线。矩阵可以用来帮助计算多边形的光效果，但是也可以由用户来设定给出其他的光效果
NormalMode		MATLAB是否自动设置VertexNormals。如果NormalMode设为缺省值'auto'，MATLAB就计算VertexNormals；如果设为'manual'，则直接使用VertexNormals。用户如果设置VertexNormals，那么NormalMode就自动设为'manual'

曲面对象可以在三维视图中创建‘飞毯’。

14.2.12 曲面对象

命令集165 曲面对象

`surface(x,y,Z,c)` 在当前坐标系中创建一个由 x 、 y 和 Z 定义的填充曲面。向量或者矩阵 x 和 y 是可选参数， Z 是一个矩阵，见表 14-23 中 $XData$ 、 $YData$ 和 $ZData$ ，参数 c 表示使用颜色的标量或者矩阵，见表 14-23 中 $CData$ 。如果没有给出 c ，则令 $c=Z$ 。命令返回得到创建的曲面对象的句柄。

`surface(prstr, alt, ...)` 创建一个曲面，同时将它的属性 `prstr` 的值设为 `alt`，可以对多个属性进行设置。返回得到曲面对象的句柄。

曲面对象有下列属性和方法，其中有一些属性和表 14-21 中列出的相同。

表14-23 曲面对象的属性和方法

CData	指定曲面上每个顶点的颜色，也就是ZData中每一点的颜色。如果FaceColor设置为'texturemap'，那么CData的大小就不必和ZData相同，这样CData中包含的图像被映射到ZData所定义的曲面。见表 14-21 可知顶点的颜色是如何指定的
-------	---

(续)

CDataMapping	见表14-21
EdgeColor	见表14-20
EraseMode	见表14-21
FaceColor	见表14-21, 而且属性值可以设为 'texturemap', 见CData
LineStyle	见表14-21
LineWidth	见表14-21
Marker	见表14-21
MarkerEdgeColor	见表14-21
MarkerFaceColor	见表14-21
MarkerSize	见表14-21
MeshStyle	画网格的边缘行和/或列线: 'both' (缺省值)、'row' 或 'column'
XData	$m \times n$ 矩阵或 $m \times 1$ 向量(重复列向量形成 $m \times n$ 矩阵)表示曲面中带内的 x 坐标
YData	$m \times n$ 矩阵或 $m \times 1$ 向量(重复列向量形成 $m \times n$ 矩阵)表示曲面中带内的 y 坐标
ZData	$m \times n$ 矩阵或 $m \times 1$ 向量(重复列向量形成 $m \times n$ 矩阵)表示曲面中带内的 z 坐标
Children	空矩阵, 曲面对象没有子对象

和多边形一样, 曲面对象可以有光照效果, 有关的属性和方法总结如下 (表14-24); 详细的描述在表14-22中。

表14-24 控制曲面对象光照效果的属性和方法

FaceLighting	EdgeLighting	BackFaceLighting
AmbientStrength	DiffuseStrength	SpecularStrength
SpecularExponent	SpecularColorReflectance	VertexNormals
NormalMode		

例14.13

(a) 在例14.12中用下列方法本可以创建了一个球体:

```
[X, Y, Z]=sphere;      % 取球体坐标
ss=surface(X,Y,Z);     % 创建并画出曲面对象
view(3);
```

可以用set命令将对象属性 'FaceColor' 的属性值 'flat' 改为 'interp':

```
set(ss, 'FaceColor', 'interp')
```

属性 'FaceColor' 的属性值可以为:

```
set(ss, 'FaceColor')
```

```
[ none | {flat} | interp | texturemap ] -or- a ColorSpec.
```

属性 FaceColor 是有效的, 例如当给出命令 patch、surface 或 mesh 时, 就用 'texturemap' 来代替 'interp' 或 'flat'。属性值 'none' 将用在下面的例子中。

(b) 命令 mesh 是一个用来创建曲面的高级命令, 可以得到和用命令 surface 一样的创建曲面的句柄。下面的程序用到了命令 patch 和 mesh, 并且对对象的属性进行了修改。

```
% 向量x1 y1和z1表示三维视图内的一个立方体
```



```

z1 = [1 1 -1 -1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1];
y1 = [1 -1 -1 1 1 1 1 1 1 -1 -1 1 -1 -1 -1 -1];
x1 = [1 1 1 1 1 -1 -1 1 -1 -1 -1 -1 -1 1 1 -1];

clf % 清除当前图形
p = plot3(x1,y1,z1); % 画出立方体的线，并保留句柄

set(p,'LineWidth',3,'Color','b') % 改变线条属性

[XX,YY,ZZ] = sphere(15); % 矩阵XX、YY、ZZ定义一个单位球体
hold on;

h1 = mesh(XX,YY,ZZ); % 画出单位球体，并保留句柄

% 改变球体属性
set(h1,'EdgeColor','b','FaceColor','c')
h2 = mesh(2.*XX,2.*YY,2.*ZZ); % 画一个球体(半径为2)，并保留句柄

% Changes sphere properties.
set(h2,'EdgeColor','r','FaceColor','none')

set(gca,'Visible','off') % 改变当前坐标的可视属性

axis square % 使球体园滑

```

结果如图 14-13 所示。

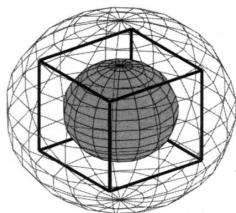


图14-13 大球体中包含一个立方体，立方体中有一个单位球体

14.2.13 文本对象

要将文本显示在图形窗口中，就要用命令 `text` 来创建文本对象。可以指定文本显示的位置，以及改变对象的属性。

命令集166 文本对象

<code>text(x,y,txt)</code>	返回得到文本对象的句柄。在当前二维坐标系中的 (x, y) 位置增加文本字符串 <code>txt</code> ，坐标用当前坐标轴的刻度来指定。
<code>text(x,y,z,txt)</code>	和上个命令一样在当前三维坐标系中增加文本字符串 <code>txt</code> 。
<code>text(prstr, alt, ...)</code>	创建一个文本对象，并将其属性 <code>prstr</code> 的值设置为 <code>alt</code> 。同时可以对多个属性进行设定。

文本对象有下列属性和方法。

表14-25 文本对象的属性和方法

Color	一个RGB三元组或一种MATLAB中预定义的颜色，表示文本的颜色；缺省值为 'white' (白色)，见表14-20
EraseMode	见表14-20
Editing	文本是否可编辑，缺省值 'off' 表示文本不可编辑。如果将属性值设为 'on'，在文本的开始处就有一个光标，这样就可以进行文本编辑。完成后需按下ESC键、点击图形窗口或将Editing设为 'off'。
Extent	有四个元素的只读向量 [left bottom width height] 表示文本位置(在坐标系中)和文本String的大小
FontAngle	指定字体角度的字符串，可选值为 'normal' (缺省值)、'italic' 和 'oblique'。
FontName	见表14-14
FontSize	见表14-14
FontUnits	见表14-14
FontWeight	见表14-14
HorizontalAlignment	表示文本水平对齐，有以下对齐方式：'left' (缺省值)、'center' 和 'right'。
Position	向量 [x y] 或 [x y z]，指出文本在二维或三维空间中的位置
Rotation	以旋转度数表示的文本方向，缺省值为 0
String	细胞向量、字符串矩阵或字符串，指定要显示的文本内容
Units	位置属性值的度量单位，有下列度量单位：'inches'、'centimeters'、'normalized'、'points'、'pixels'、'characters' 和 'data' (缺省值)。缺省值表示屏幕像素。度量单位的选择影响属性 'Position' 和 'Extent'，见上面
Interpreter	String中是否包含 \LaTeX 文本格式命令(缺省为 'tex' 或 'none')。见 helpdesk 可得更多信息
VerticalAlignment	表示文本垂直对齐，有以下对齐方式：'top'、'cap' (同 'top' 但用大写字母)、'middle' (缺省值)、'baseline' (字体的基线在指定的位置)和 'bottom'。
Children	空矩阵[]，文本对象没有子对象

例14.14

在这个例子中，线条和相应的文本颜色相同，另外，改变线条的宽度和文本的字体。

```
clear; clf;

x = linspace(2,10,100);
y1 = sin(x);
y2 = bessell(1,x);
ha = axes('Position',[0.1 0.1 0.6 0.8]);

l1 = line(x,y1);
t1 = text(x(100),y1(100),'Sine');

l2 = line(x,y2);
t2 = text(x(100),y2(100),'Bessel');
```

```

set(l1,'Color'      ,'Blue'   );           % 设置线1的颜色和线条宽度
set(l1,'LineWidth'  ,3        );

set(t1,'Color'      ,'Blue'   );           % 设置文本的颜色、字体和大小
set(t1,'FontWeight' ,'bold'   );
set(t1,'FontSize'   ,18       );

set(l2,'Color'      ,'Red'     );           % 设置线2的颜色和线条宽度
set(l2,'LineWidth'  ,10       );

set(t2,'Color'      ,'Red'     );           % 设置文本2的颜色、角度、字体和大小
set(t2,'FontAngle'  ,'italic' );
set(t2,'FontName'   ,'palatino');
set(t2,'FontSize'   ,30       );

set(ha,'Box','off');                       % 只画出坐标轴，不画出网格

```

结果显示在图14-14中，不幸的是没有显示出颜色来。

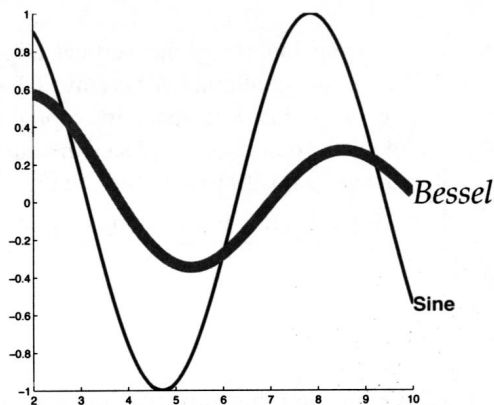


图14-14 有相应文本的线条

14.2.14 光对象

光对象是MATLAB 5中介绍的一种新类型对象，可以用来创建光源。光源并不真正可见，但是它们能影响补片和曲面的亮度。下列属性均对这些对象的亮度有影响：AmbientStrength、DiffuseStrength、EdgeLighting、FaceLighting、BackFaceLighting、SpecularStrength、SpecularExponent、SpecularColorReflectance和VertexNormals。这些属性的详细描述可见表4-22和表4-24。

可以定义整个坐标轴的背景光，见表14-11中属性AmbientLightColor。

命令集167 光对象

`light(prstr, alt,...)` 创建一个光对象，将其属性 `prstr` 的值设置为 `alt`，同时可以对多个属性进行设定，返回得到光对象的句柄。

在调色板一节中的图P-2和P-8就是用light命令创建的。使用句柄来对对象的属性进行设定。光对象有下列属性和方法。

表14-26 光对象的属性和方法

Position	向量 $[xyz]$ 表示光在其父对象坐标轴中的位置或方向
Color	一个RGB三元组或一种MATLAB中预定义的颜色，表示光的颜色
Style	光源的类型： ' infinite ' (缺省值)表示光源从Position方向发出平行光束 ' local ' 表示光源的位置在Position，并向所有方向发出光束
ButtonDownFcn	无效属性
Children	空矩阵[]，光对象没有子对象
Clipping	无效属性
HitTest	无效属性
Interruptible	对回调函数DeleteFcn没有影响
Selected	无效属性
SelectionHighlight	无效属性
UIContextMenu	无效属性

例14.15

在这个例子中创建一个曲面对象并观察在有光源和没有光源情况下的差别。程序如下：

```
% 用缺省值创建一个曲面对象

subplot(2,1,1)
surf(peaks);
axis('off')
title('unlit surface')

subplot(2,1,2)
sp = surf(peaks);

% 曲面颜色红色，无网格，光照算法'phong'

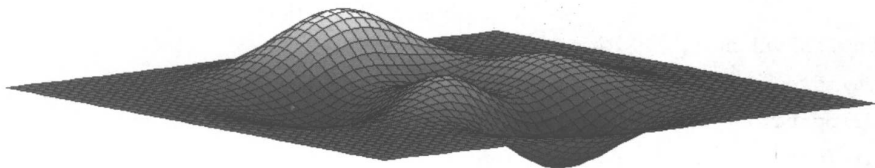
set(sp,'FaceColor','red')
set(sp,'EdgeColor','none')
set(sp,'FaceLighting','phong')

% 使用平行于x轴的光束照曲面

light('Position',[1 0 0],'Style','infinite')
axis('off')
title('lit surface')
```

程序运行得到的结果如图 14-15所示。

无光照曲面



光照曲面

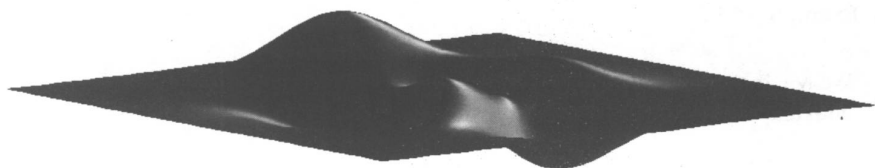


图14-15 有光照和没有光照的曲面

14.3 图形用户界面

MATLAB有强有力的工具来建立图形用户界面，而且有许多预定义窗口，如下：

命令集168 预定义窗口

`dialog(prstr,alt,...)` 创建一个对话框图形对象，并将其属性 **prstr** 设置为 **alt**，同时可以对多个属性进行设置。窗口是对话框模式，见 `msgbox`。返回得到对象的句柄。

`msgbox(message, title, icon, icondata, iconclr, mode)` 创建一个带有信息的窗口，并返回它的句柄，实际上它也是一个图形对象。

- 参数 **message** 是一个字符串、字符串矩阵或细胞向量
- **title** 也是一个字符串表示窗口的标题。
- 字符串 **icon** 表示在窗口上显示的图标类型，有下列几种类型图标：'none'、'error'、'help'、'warn' 或 'custom'。
- 只有图标类型是 'custom'，才能在 **icondata** 中给出用户自定义图标数据，和使用 **iconclr** 中的颜色表。
- 参数 **mode** 表示窗口的模式。有下列几种模式：
'modal' 要求用户反馈信息、'nonmodal' 表示不必用户反馈信息、'replace' 表示无对话模式，并替换有相同标题的窗体。

`helpdlg(hlpstr,title)` 在标题名为 **title** 的窗体中创建一个有帮助文本 **hlpstr** 的对话框。当用鼠标选定一个按钮，窗体就消失。返回窗体的句柄，也是一个图形对象。

`helpwin topic` 显示一个 MATLAB 突出主题 **topic** 的帮助窗口，可以是 M 文件中一个用户自定义命令。在命令帮助中用线条上的 ‘ See Also ’ 将其他相关的命令连接起来。

`helpwin(hlpstr, heading,title,prstr, alt,...)` 在标题 **heading** 的帮助窗口 **title** 中显示帮助文本。这里的 **hlpstr** 是帮助文本，可以是细胞向量、字符串矩阵或字符串，每行用一个 ‘ \n ’ 隔开，也就是 ‘ 回车 ’ 字符。字符串 **prstr** 是文本对象的一个属性，并将它的属性值设为 **alt**，它对帮助文本的显示有影响。可以同时多个属性进行设置。

`helpwin(r1 t1; r2 t2;... ,page, title, prstr,alt,...)` 同上 (**r1**、**r2**... 是标题，**t1**、**t2**... 是帮助文本)，在不同的标题下可能创建出多个帮助页。字符串 **page** 必须和其中的一个标题相同，表示开始显示相应的帮助文本，缺省值为 **r1**。

`warnldg(warn, title)` 创建一个有警告信息的对话框。文本 **warn** 显示在标题为 **title** 的窗口中。当鼠标选中一个按钮时，窗口就会消失。返回得到窗口的句柄，也是一个图形对象。

`errordlg(errstr, title, 'on')` 在标题为 **title** 的窗口中创建一个带有错误信息的对话框。如果给出 ‘ on ’，标题为 **title** 的窗口存在就将它变成当前窗口；如果窗口不存在，就创建一个标题为 **title** 的新窗口。当用鼠标选中一个按钮时，窗口就会消失。返回得到窗口的句柄，也是一个图形对象。

`questdlg(qst, title,alt1,alt2,alt3, default)` 创建一个有提问 **qst** 和标题 **title** 的对话框。如果可选参数没有给出，窗体上就有三个按钮 ‘ Yes ’、‘ No ’ 和 ‘ Cancel ’，也可以用文本 **alt1**、**alt2** 和 **alt3** 给出按钮上的显示文本 (至少给出两个)。字符串 **default** 表示用户按下按钮时返回的内容：如果是使用标准按钮，字符串就等于 ‘ Yes ’、‘ No ’ 或 ‘ Cancel ’；否则等于文本 **alt1**、**alt2** 或 **alt3**。选中一个按钮，窗口就会消失，并且返回按钮上的文本内容。

`inputdlg(legend, title,lineNr,dfltAns)` 创建一个窗口用来输入数据。细胞向量 **legend** 包含答案的图例。字符串 **title** 表示窗口的名字，**lineNr** 是一个包含每个答案行数的向量 (或者是总的行数)。细胞向量 **dfltAns** 包含缺省的答案。命令返回用户给出答案的细胞向量。

`menu(title, alt1, alt2,...)` 创建一个菜单，它的标题是 **title**，有菜单选项 **alt1**、**alt2** 等，等待从键盘或者鼠标输入。返回用户选择的菜单选项，以整数表示。注意：如果其他图形对象调用

```
[selection, ok]=  
listdlg('ListString',  
S,prstr,alt,...)
```

```
[fname, path]=  
uigetfile(filter,  
title, x, y)
```

```
[fname, path]=  
uiputfile(filter,title,x,y)  
pagedlg(fig)
```

```
printdlg(  
'-crossplatform',  
fig)
```

```
uisetcolor(x, title)
```

```
uisetfont(pr, title)
```

```
btngroup(h,prstr,  
val, ...)
```

menu, 那么就必须将那个对象的属性 'Interruptible' 设置为 'yes'。

创建一个列表框。参数S是一个细胞矩阵, 包含列表项。字符串prstr是窗口的属性, 属性值设为alt。可以同时多个属性进行设置, 见helpdesk可得更多关于属性信息。如果给出向量selection, 表示选中的列表项在列表S中的索引。如果给出ok, 用户按下OK按钮, 则ok=1; 否则ok=0(在这种情况下selection是空矩阵)。

创建一个打开文件的窗口。字符串filter决定列出的文件;

字符串可以包含 "wildcards" 或只是一个文件名, 例如 '*.m' 列出所有M文件。字符串title是窗口的标题。参数x和y表示窗口在屏幕上的位置, 有些系统可能不支持这个选项。如果给出fname和path, 它们分别保存的是所选文件的名称字符串和所选文件的路径名字符串。如果用户选中取消按钮或者有错误发生, 就返回0。

同上, 但是是输入到文件中去。

创建一个对话框, 用户可以在其中选择输出当前图形的纸张格式。可选参数fig表示纸张格式应用的图形fig, 而不是当前图形。

创建一个图形打印对话框, 可选参数fig表示要打印的图形数fig, 而不是缺省的当前图形。如果给出 '-crossplatform', 就使用一个MATLAB对话框, 而不是操作系统的标准窗口。

创建一个窗口, 用户可以从中选择颜色, 可选参数x可以是图形对象的句柄或者一个RGB三元组。如果没有指定x, 就使用缺省颜色和黑色, 字符串title表示窗口的名字。命令返回一个RGB三元组。如果用户选择取消按钮或者发生错误, 就返回0。

创建一个窗口, 用户可以从中选择字体。字符串title表示窗口的名字, 可选参数pr必须是一个轴对象或文本对象的句柄。选中的字体就应用在句柄为pr的对象上。如果没有指定pr, 就返回一个应用选中字体的新文本对象句柄。如果用户选中取消按钮, 就返回0。

用来在句柄h的图形窗口中创建工具条, 也就是按钮条。如果没有指定h, 则使用当前窗口。可以通过设置工具条的属性来控制它的显示: 将属性prstr设置为alt, 同时可以对多个属性进行设置。用help btngrou可

tabdlg

waitbar(x, title)

得更多关于可用属性的信息。命令btndown、btnup、btnicon、btnpress和btnstate也可用来控制工具条，用help可得更多信息。

创建一个带有标记的对话框。使用help tabdlg可得更多信息。

创建一个进度显示条。参数x表示已完成的部分，字符串title表示窗口的名字。在显示工作进度时反复调用这个函数，同时x的值增长(从0到1)。返回窗口，即图形对象的句柄。当工作完成时可以用关闭按钮关闭这个窗口。

下列工具可以在图形对象中交互式地改变对象属性。

命令集169 交互式控制属性的工具

axlimdlg

一个图形用户界面，用来在轴对象中定义轴的坐标区间，使用help axlimdlg可得更多信息。

edtext

一个图形用户界面，用来在轴对象中交互式地编辑文本对象。使用help edtext可得更多信息。

plottedit

一个图形用户界面，用来添加文本和箭头到轴对象上，也可以用来改变轴对象的一些属性。使用help plottedit可得更多信息。

例14.16

创建一个有三个选项的菜单：

```
choice = menu('Choose:', 'Enter', 'Wait', 'Leave')
```

在XWindows环境中给出的窗口如图14-16所示。

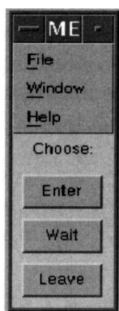


图14-16 用命令menu创建菜单。因为这个菜单是细条形的，所以只能看见窗口标题‘MENU’的前两个字母

可以单击按钮‘Enter’、‘Wait’或‘Leave’进行选择。如果选中中间按钮，也就是‘Wait’，会返回得到：

```
choice =
      2
```

因为选中的是第二个按钮。

在不同的操作系统中，菜单看起来是不一样的。如果没有可用的图形，MATLAB就在命

令窗口中给出菜单选项并等待从键盘输入。

有时需要创建出高级图形用户界面，通常预定义窗口就不够用，必须定义新的窗口。这就要用到**Guide**(图形用户界面开发环境)工具包。这个工具包包含有五个程序用来创建窗口、菜单等。虽然可以使用 `set` 和 `get` 命令(见命令集154)，但是使用 **Guide** 工具包可以更有效地进行图形用户界面的开发。

命令集170 Guide工具包

<code>ctpanel(h)</code>	在 Guide 工具包中显示和其他程序相连接的控制板。在控制程序下给出可选句柄向量 <code>h</code> 表示的图形对象，如果没有给出 <code>h</code> ，则使用当前图形句柄。在这个程序中可以增加用户控制对象，如增加按钮和文本框到图形对象中。
<code>guide(h)</code>	同上。另外， Guide 工具包的所有工具随着可选句柄向量 <code>h</code> 表示的图形对象进行刷新。如果没有给出 <code>h</code> ，则使用当前图形。
<code>propedit(h)</code>	打开属性编辑器，这个工具用来设置上节中列出的所有对象属性。参数 <code>h</code> 是对象句柄或句柄向量。如果没有给出 <code>h</code> ，则使用当前对象句柄。如果句柄向量 <code>h</code> 中的对象类型不同，则只列出它们的共有属性。
<code>align</code>	打开队列工具，这个工具可以用来排列用户控制对象和坐标系，以便它们都在直线上，并且之间有一定距离。这个工具可以分开使用；见 <code>helpdesk</code> 可得更多信息。
<code>cbedit(h)</code>	打开事件过程编辑器，这个工具可以用来决定当用户按下按钮，在文本框中输入文本时运行的回调函数。参数 <code>h</code> 是一个对象句柄或者句柄向量。如果没有指定 <code>h</code> ，则使用当前对象句柄。
<code>menuedit(h)</code>	打开菜单编辑器。这个工具可以用来改变用户自定义菜单的一些属性。参数 <code>h</code> 是对象句柄或者句柄向量。如果没有指定 <code>h</code> ，则使用当前对象句柄。

在设计一个图形用户界面时，开始要做的第一件事情是仔细考虑界面的外观和要完成的功能，最后很容易地使用 **Guide** 工具包中提供的工具来完成设计。

首先要完成程序的设计，用工具包来完成时就象面向对象的程序设计，它画出的窗口上可以带有按钮、弹出式菜单等。

完成界面绘制后的下一步是添加界面函数。当用户按下一个按钮时，希望会触发事件。界面函数可以由回调函数来实现，用 **Guide** 工具包中的事件过程编辑器很容易地将它们放进图形对象中。

以上概括性地给出图形用户界面的开发过程。下面将对 **Guide** 工具包中的每个工具(见命令集170)进行介绍，以及用例程来进行说明。

在这个例程中，所有的屏幕画面都取自 X Windows 环境，同样也可以用在 Windows 和 Macintosh 系统中。实际上，在某个操作系统中创建的用户界面可以不做任何修改就能移植到另一个系统中。

在MATLAB手册《用MATLAB建立GUI》中可以找到更多关于建立图形用户界面的信息。这个手册还给出了在建立图形用户界面时要注意的一些问题。

14.3.1 控制面板

在MATLAB的命令窗口中键入命令 `guide`，将启动控制面板，同时还打开一个被 Guide 工具包控制的图形窗口。网格表示该图形窗口当前的状态为受控状态，见图 14-18。如图形窗口处于打开状态，那么当前图形对象就由控制面板所控制。

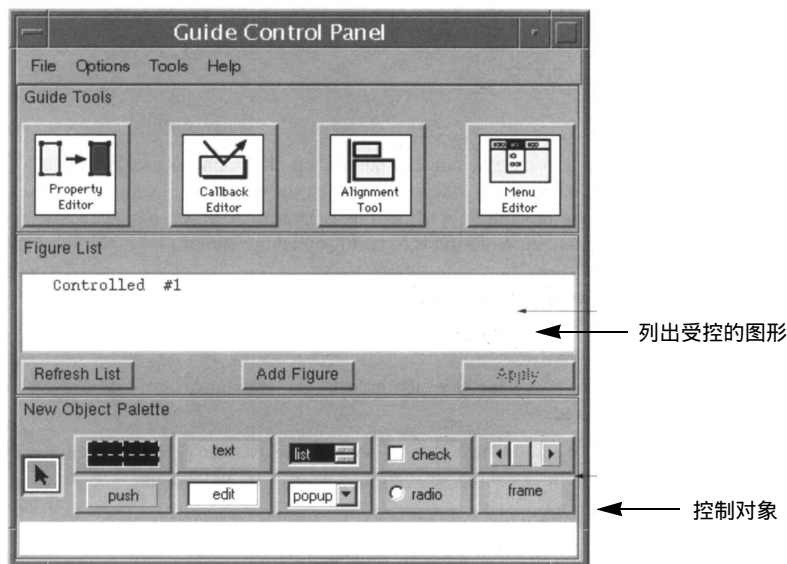


图14-17 控制面板

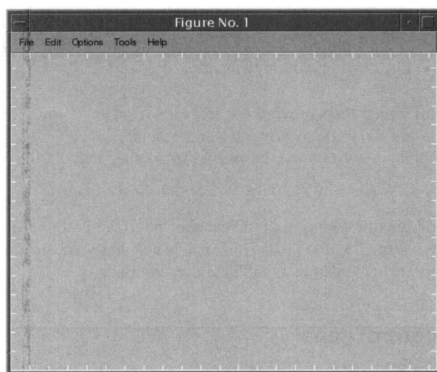


图14-18 Guide工具控制下的图形窗口

从图14-17中可以看出在 Guide 工具包中其他工具的连接关系，而且，有 10 种不同类型的控制对象可以添加到图形窗口中来建立用户界面。

例14.17

这个例子说明了如何来建立一个简单用户界面，创建一个窗口，可以让用户在窗口中选择颜色。这实际上可以用预定义的窗口来完成，`uicontrol`(见命令集168)可用于完成类似的任务。

首先决定要用的控制对象，下列对象可能都会用到：

- 三个滑标('slider')，分别用来表示红、绿、蓝的颜色数。
- 三个可编辑的文本框('edit')，用户可以分别给出所需要的三种颜色准确的颜色值。
- 一个帧('frame')，用当前选中的颜色进行填充。
- 七个解释字符串('text')。
- 两个按钮('push button')：CANCEL和OK。

之后可以用控制面板来设计窗体。在控制面板中挑选对象后将它们拖到图形对象上。这一步只是给出窗体的大概设计；对象的大小和位置不必很准确。在图形窗口中可以进行对象的剪切、粘贴和复制。

图14-19中给出了控制对象在窗体的大概布置。

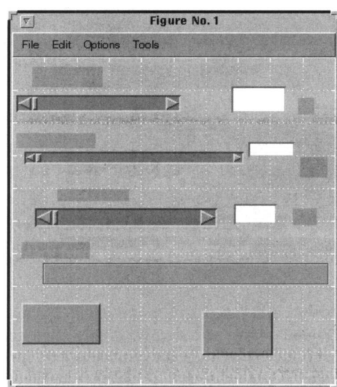


图14-19 控制对象的布置

使用下面两个工具来调整窗体。

14.3.2 属性编辑器

在控制面板中按下属性编辑器按钮，就会显示出下面的窗口来(图14-20)：

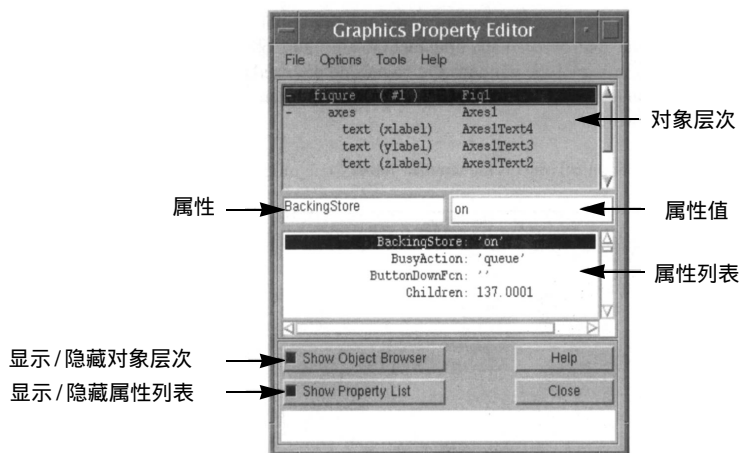


图14-20 属性编辑器，给出了对象的层次关系和属性列表

可以看出属性编辑器有四个部分，在窗口的顶部给出了对象层次关系（见图14-4），表示包含的所有对象以及对象之间的关系。其中左列字符表示对象是否有子对象：加号‘+’表示对象有子对象，但是未列出其子对象名字，如果点击对象名字时会显示出其子对象；减号‘-’表示对象有子对象，但是已经列出其子对象的名字。没有字符表示对象没有任何的子对象。选中一个或多个对象（shift—左键），在窗体的中间的属性列表中显示出它们的属性。可以在层次列表中选中对象，也可以在图形窗口中直接进行选择。

属性列表框中可以列出对象的所有属性。如果选中的对象类型不同，则只列出它们的共有属性。如果选中的对象中有些属性有不同的值，在列表框中以‘Multiple Values’表示出来。选中一个属性，它的名字和相应的值就会在属性列表框上面的属性框和属性值框中分别给出。也可以不选择，而直接在属性框中输入属性名字。不必完全输入，因为有一些字符能唯一确定属性。例如，输入col，然后回车，属性编辑器就会给出属性Color和它的属性值。

显然，可以在属性框和属性值框中修改属性和属性值。如果属性只被赋予几个限定的值或文本区域，那么属性值框也可以是一个弹出式菜单。在文本区域中，属性编辑器忽略第一个空格之后的所有值，因此可以在行的开始处输入新值并按回车来修改属性值。

例14.18

接例14.17，利用属性编辑器可以对下列属性进行修改：

所有对象：

HandleVisibility: 'callback'

窗体('figure')

MenuBar: 'none'

NumberTitle: 'off'

Name: 'Color Picker'

Resize: 'off'

Tag: 'ColPick'

滑标('slider')

Position: [? ? 200 20]

Max: 100

Value: 0

Tag: 'RedSl', 'GreenSl', and 'BlueSl', respectively

滑标上的文本('text')

Position: [? ? 50 16]

FontSize: 12

HorizontalAlignment: left

BackgroundColor: [0.8 0.8 0.8]

ForegroundColor: 'red', 'green', and 'blue', respectively

String: 'Red', 'Green', and 'Blue', respectively

滑标右边的可编辑文本框('edit')

Position: [? ? 50 25]

String: '0'

UserData: 0

FontSize: 12

```
HorizontalAlignment: left
Tag:                  'RedTxt', 'GreenTxt', and 'BlueTxt', respectively
```

可编辑文本框右边的文本(' text ')

```
Position:             [ ? ? 16 16]
FontSize:              12
HorizontalAlignment:  left
BackgroundColor:       [0.8 0.8 0.8]
String:                '%'
```

颜色框(' frame ')

```
Tag:                  'col'
BackgroundColor:       'black'
```

颜色框上的文本(' text ')

```
Position:             [ ? ? 50 16]
FontSize:              12
HorizontalAlignment:  left
BackgroundColor:       [0.8 0.8 0.8]
String:                'Color'
```

按钮(' pushbutton ')

```
Position:             [ ? ? 70 30]
FontSize:              14
FontWeight:            'bold'
String:                'CANCEL' and 'OK' respectively
Tag:                  'cancel' and 'OK' respectively
```

经过这些调整后，窗体如图 14-21所示。

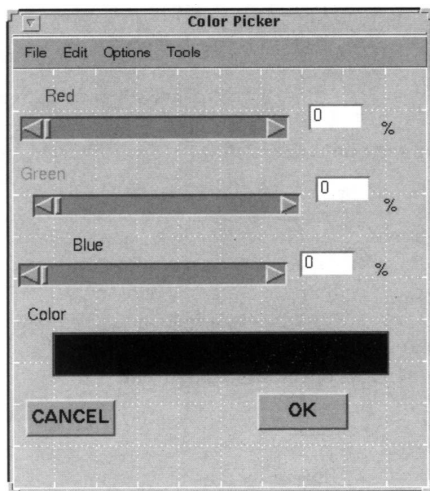


图14-21 用属性编辑器调整后的窗体

14.3.3 对齐工具

按下控制面板中的对齐工具按钮，会显示出如图 14-22的窗体。

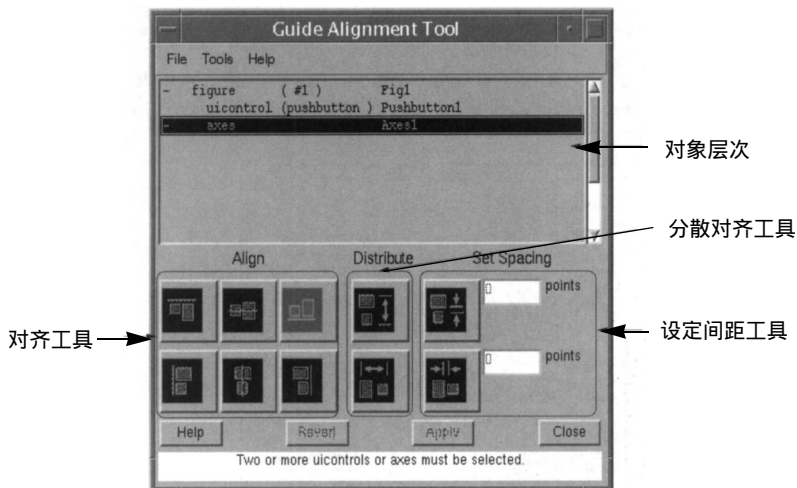


图14-22 对齐工具

在窗体的顶部列出了对象的层次列表，这和属性编辑器是一样的：可以在层次列表中选择对象，也可直接在图形窗口中选择。然而不同的是每次至少要同时选中两个对象。

当选中几个对象时就可以对它们进行相对调整，这可以通过选择调整工具来完成，也就是选择对象层次列表下的按钮。单击‘Apply’按钮进行确认，如果单击‘Revert’按钮，则取消调整设定。

有三组调整工具，第一组是对齐工具，它们将选中的对象在某个轴上对齐，六个工具都指定唯一轴。

第二组调整工具是分散对齐工具，它们将选中的对象在水平方向或垂直方向分散对齐，使得对象间的距离都是固定的。当在对象间调整时，最外面的两个对象保持原来位置不动。

第三组调整工具是设定指定的间距工具，它们可以在水平方向或垂直方向设定每对选中对象指定的间距。选中对象的调整从左下角开始，一直向上到右上角。注意，有些对象可能会超出图形窗口的范围。

例14.19

接例14.17，最后使用对齐工具来美化窗体。通过调整后，窗体如图14-23所示。

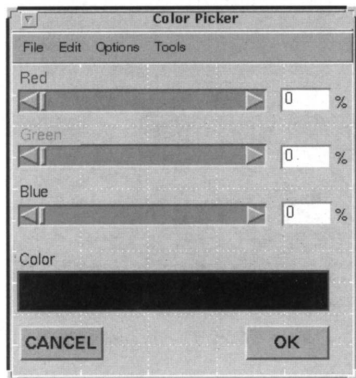


图14-23 经过对齐调整以后的图形

在控制面板中双击文本 ‘ Controlled #1:Color Picker ’ 可以激活这个界面。在界面被激活前，系统会给出提示是否保存界面，可以将它保存在文件 **ColorPicker.m**中，这是一个普通的M文件。在文本编辑器中可以查看文件内容。另外数据MAT文件被保存在文件 **ColorPicker.mat**中。当文件保存后界面被激活，此时可以拖动滑标，点击按钮等，但是没有任何事件发生。这是因为没有编写界面函数，所以要用到下一个工具：事件过程编辑器。

14.3.4 事件过程编辑器

在控制面板中单击 “ Callback Editor ” 按钮，打开如图 14-24所示的事件过程编辑器。

事件过程编辑器的顶部是对象层次列表，和属性编辑器一样。可以在层次列表中或直接在图形中选择要编写回调函数的对象。

对象有对应着不同事件的不同类型回调函数。例如创建或撤消一个对象、用户用鼠标点击对象，这些都是事件。通过回调函数菜单来选择回调函数类型，见图 14-24。

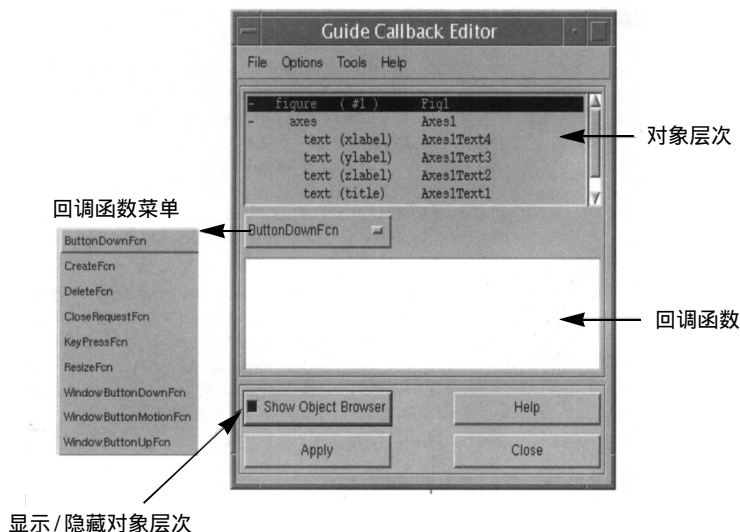


图14-24 事件过程编辑器

当选定某个类型的回调函数后，在事件过程编辑器的中间位置文本区域内书写函数体，和在MATLAB命令窗口中一样。除非函数特别的短，比较方便的做法是调用一个包含事件过程函数代码的M文件。值得注意的是如果在M文件中做了改动，比如调试时，那么这些改动只有在MATLAB命令窗口中输入命令 `clear function`以后才有效。

在编写事件过程函数时有许多很有用的MATLAB命令。

命令集171 事件过程命令

<code>[h,f i g]=gcbo</code> <code>gcbf</code>	<p>返回当前正在执行回调函数的对象句柄 <i>h</i>。如果没有函数在执行，返回空矩阵 []。可选参数 <i>f i g</i> 中是返回的父图形对象句柄。</p> <p>返回当前正在执行回调函数的对象的父对象句柄。如果没有函数在执行，返回空矩阵 []。如果回调函数属于图形对象，</p>
--	--

	则返回这个对象的句柄。
<code>selectmoversize</code>	使用时和 <code>ButtonDownFcn</code> 相同。当执行时对图形对象有下列动作：选择、移动、复制和重定义大小。返回一个结构 <code>A</code> ，其中域 <code>A.Type</code> 可以是 'Select'、'Move'、'Resize' 或 'Copy'，域 <code>A.Handles</code> 可以是被选中对象 ('Select') 的句柄向量，也可以是一个 $m \times 2$ 矩阵。矩阵中的第一列是原始对象的句柄，第二列是复制后新对象的句柄 ('Copy')。
<code>overobj(type)</code>	在鼠标指针下搜索可视的图形对象，字符串 <code>type</code> 给出搜索图形对象的类型。这个命令使用的前提是根对象的属性 <code>Units</code> 设置为 'pixels'，而且只适用于有子对象的图形对象上。
<code>setstatus(h,str)</code>	设置用户控制对象的属性 <code>String</code> ，其中用户对象的属性 <code>Style</code> 设置为 'text'，属性 <code>Tag</code> 设置为 'Status'；见表 14-15。用户控制对象必须是句柄 <code>h</code> 图形对象的子对象。
<code>getstatus(h)</code>	和上个命令相似，但是返回属性 <code>String</code> 。
<code>hidegui(h,status)</code>	设置句柄 <code>h</code> 图形对象的属性 <code>HandleVisibility</code> ；见表 14-2。如果没有指定 <code>h</code> ，则使用当前对象句柄。参数 <code>status</code> 是一个可选字符串，可以为缺省值 'on'、'off' 或 'callback'。这个命令主要用来保护用户界面，防止用户通过命令窗口来撤消它。
<code>status=hidegui(h)</code>	返回得到句柄 <code>h</code> 图形对象的属性 <code>HandleVisibility</code> 。如果没有指定 <code>h</code> ，则使用当前对象句柄。
<code>waitfor(h,prstr,alt)</code>	阻碍函数调用直到撤消句柄 <code>h</code> 图形对象或用户按下 <code>Control C</code> 。同时，如果给出可选属性 <code>prstr</code> ，这个属性将被改变。而且，如果给出 <code>alt</code> ，就将 <code>prstr</code> 的值设为 <code>alt</code> 。如果 <code>h</code> 不是一个句柄， <code>prstr</code> 不是一个合法的属性或 <code>alt</code> 不是一个有效的值，则这个命令不起作用。
<code>uiwait(h)</code>	阻碍函数调用，事件过程函数必须存在于句柄 <code>h</code> 图形对象的一个控制对象中，也就是撤消图形对象或调用 <code>uiresume</code> 。如果没有指定 <code>h</code> ，就用当前图形对象句柄。
<code>uiresume(h)</code>	继续执行被 <code>uiwait</code> 阻碍的函数，见上。
<code>status=uisuspend(h)</code>	挂起所有与句柄 <code>h</code> 图形对象中控制对象的用户交互。例如，单击按钮不再有事件过程函数被调用。返回得到的结构 <code>status</code> 中包含图形对象的属性信息。用 <code>uirestore</code> 可以重复取消挂起。
<code>uirestore(status)</code>	取消被 <code>uisuspend</code> 挂起的用户交互，参数 <code>status</code> 是 <code>uisuspend</code> 返回得到的结构。

例 14.20

接例 14.17。通过双击控制面板中的文本 'Active #1:Color Picker' 来设置 Guide 工具控制

下的图形窗口。在事件过程编辑器中编写下列 ‘ Callback ’ 类型的事件过程函数：

```

红色滑标：      CbColPic redsl;
红色文本区域：  CbColPic redtxt;
绿色滑标：      CbColPic greensl;
绿色文本区域：  CbColPic greentxt;
蓝色滑标：      CbColPic bluesl;
蓝色文本区域：  CbColPic bluetxt;
Cancel按钮：    CbColPic cancel;
Ok按钮：        CbColPic ok;

```

下面编写M文件CbColPic.m：

```

function CbColPic(action)

switch(action)

case 'redsl'                % 红色滑标
                            % 取红色文本框的指针
    RedTxtPtr = findobj(gcf,'Tag','RedTxt');
                            % 读取滑标值
    Val = get(gcbo,'Value');
                            % 设置红色文本框为滑标值
    set(RedTxtPtr,'String',num2str(Val,3));
                            % 刷新框的颜色
    LocalUpdateCol;

case 'greensl'              % 绿色滑标
    GreenTxtPtr = findobj(gcf,'Tag','GreenTxt');
    Val = get(gcbo,'Value');
    set(GreenTxtPtr,'String',num2str(Val,3));
    LocalUpdateCol;

case 'bluesl'               % 蓝色滑标
    BlueTxtPtr = findobj(gcf,'Tag','BlueTxt');
    Val = get(gcbo,'Value');
    set(BlueTxtPtr,'String',num2str(Val,3));
    LocalUpdateCol;

case 'redtxt'               % 红色文本框
                            % 取红色滑标指针
    RedSlPtr = findobj(gcf,'Tag','RedSl');
                            % 取文本框中字符串
    Str = get(gcbo,'String');
                            % 转换字符串
    Val = LocalChStr(Str);
                            % 设置滑标值
    set(RedSlPtr,'Value',Val);
                            % 设置文本框中字符串格式
    set(gcbo,'String',num2str(Val,3));

```

```

                                % 将值保存到UserData中
    set(gcbo,'UserData',Val);

                                % 刷新颜色框
    LocalUpdateCol;

    case 'greentxt'                % 绿色文本框
        GreenSlPtr = findobj(gcbf,'Tag','GreenSl');
        Str = get(gcbo,'String');
        Val = LocalChStr(Str);
        set(GreenSlPtr,'Value',Val);
        set(gcbo,'String',num2str(Val,3));
        set(gcbo,'UserData',Val);
        LocalUpdateCol;

    case 'bluetxt'                % 蓝色文本框
        BlueSlPtr = findobj(gcbf,'Tag','BlueSl');
        Str = get(gcbo,'String');
        Val = LocalChStr(Str);
        set(BlueSlPtr,'Value',Val);
        set(gcbo,'String',num2str(Val,3));
        set(gcbo,'UserData',Val);
        LocalUpdateCol;

end

function OutVal = LocalChStr(InStr)    % 转换字符串

temp = str2num(InStr);                % 将字符串转换为数字

if (isempty(temp))                    % 字符串不是一个数字串
    OutVal = get(gcbo,'UserData');    % 使用前次值
elseif (temp > 100)                    % 数字大于100
    OutVal = 100;
elseif (temp < 0)                      % 数字小于0
    OutVal = 0;
else
    OutVal = temp;
end

function Col = LocalUpdateCol          % 刷新颜色框

% 滑标的指针
RedPtr = findobj(gcbf,'Tag','RedSl');
GreenPtr = findobj(gcbf,'Tag','GreenSl');
BluePtr = findobj(gcbf,'Tag','BlueSl');

% 取滑标值
RedVal = get(RedPtr,'Value')/100;
GreenVal = get(GreenPtr,'Value')/100;
BlueVal = get(BluePtr,'Value')/100;
Col = [RedVal,GreenVal,BlueVal];

```

```
% 取颜色框指针  
FrPtr = findobj(gcf,'Tag','col');  
% 设置它的颜色  
set(FrPtr,'BackgroundColor',Col);
```

在这个窗体中，要让滑标和它相应的文本区表示出相同的颜色数，比如红色，所以必须将它们两者关联起来。这样，当颜色滑标移动时，相应的文本区也得到刷新。

这样必须写两个函数来完成这样的工作：

- LocalChStr(InStr) 将字符串 InStr 转换成数字。这个函数用来转换用户在文本区内输入的数字字符串。
- LocalUpdataCol 设置颜色框的颜色。

通过双击控制面板上的文本 ‘Controlled #1:Color Picker’ 来激活用户界面和进行保存。颜色的设定可以通过拖动滑标或者在文本区内直接输入数值，颜色框上就可以将选中的颜色显示出来！见图 14-25。

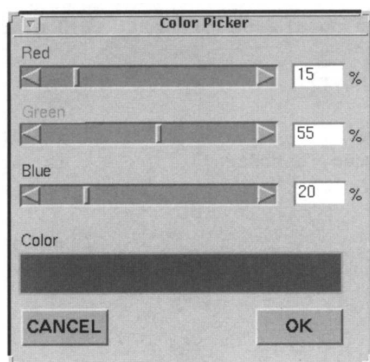


图14-25 通过编写事件过程函数来激活窗体

现在还没有编写鼠标事件，在下一小节中将会介绍。

14.3.5 菜单编辑器

在控制面板上按下 ‘Menu Editor’ 按钮，就会打开如图 14-26 所示的菜单编辑器。

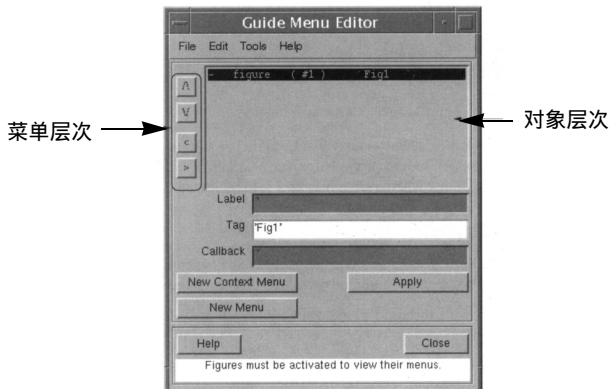


图14-26 菜单编辑器

在菜单编辑器的顶部列出了对象的层次结构关系，它和其他 Guide 工具一样。所不同的是只显示出图形和菜单对象。

菜单编辑器的其他部分及其功能：

- ‘ New Menu ’ 按钮用来添加图形对象菜单或已有菜单的菜单选项。
- ‘ New Context Menu ’ 按钮用来创建对象的快捷菜单；见 14.2.8 节。
- ‘ Label ’ 文本区用来输入菜单或菜单选项标签，这个字符串将会在用户界面中显示。
- ‘ Tag ’ 文本区用来输入分配给菜单的标记符，在编写事件过程函数时要用到这个标记符。
- ‘ Callback ’ 文本区用来输入菜单的事件过程函数，也就是在用户选择菜单时要运行的命令或 M 文件。事件过程函数必须是字符串，和属性编辑器不一样（见前小节）。
- 对象层次结构关系左边的四个按钮是用来在图形窗口中对菜单标记的，可以移动菜单或菜单中的选项。

例 14.21

接例 14.17。借助菜单编辑器来定义按钮 ‘ CANCEL ’ 和 ‘ OK ’ 的事件过程函数，但是首先要添加一个可用的菜单：预定义颜色菜单。

通常在控制面板中双击文本 ‘ Active #1:Color Picker ’ 来调出窗体。创建一个新菜单，分配给它下列数据：

Label: ‘Color’	Tag: ‘’	Callback: ‘’
Label: ‘Pink’	Tag: ‘’	Callback: ‘CbColPic menpink;’
Label: ‘Military Green’	Tag: ‘’	Callback: ‘CbColPic menmilgreen;’
Label: ‘Beige’	Tag: ‘’	Callback: ‘CbColPic menbeg;’

在 M 文件 CbColPic.m 中添加菜单选项的事件过程函数：

```
function CbColPic(action)

switch(action)

...

    case 'menpink';                                % 粉红色
        LocalUpdateAll([1 0.4 0.7]);

    case 'menmilgreen'                             % 绿色
        LocalUpdateAll([0.15 0.55 0.2]);

    case 'menbeg'                                   % 浅褐色
        LocalUpdateAll([1 0.9 0.8]);

end

...

function LocalUpdateAll(Color)                    % 全部刷新

Color = 100.*Color;

% 获取滑标指针并刷新
```

```

RedSlPtr = findobj(gcbf,'Tag','RedSl');
GreenSlPtr = findobj(gcbf,'Tag','GreenSl');
BlueSlPtr = findobj(gcbf,'Tag','BlueSl');
set(RedSlPtr,'Value',Color(1));
set(GreenSlPtr,'Value',Color(2));
set(BlueSlPtr,'Value',Color(3));

% 获取文本框指针并设置

RedTxtPtr = findobj(gcbf,'Tag','RedTxt');
GreenTxtPtr = findobj(gcbf,'Tag','GreenTxt');
BlueTxtPtr = findobj(gcbf,'Tag','BlueTxt');
set(RedTxtPtr,'String',num2str(Color(1),3));
set(GreenTxtPtr,'String',num2str(Color(2),3));
set(BlueTxtPtr,'String',num2str(Color(3),3));

```

% 获取颜色框指针并设置颜色

```

FrPtr = findobj(gcbf,'Tag','col');
set(FrPtr,'BackgroundColor',Color./100);

```

注意，现在已经编写了刷新滑标、文本区和颜色框的函数。

通过双击控制面板上的文本 ‘ Controlled #1:Color Picker ’ 激活动户界面并进行保存。现在也可以用菜单来选择颜色了，见图 14-27。

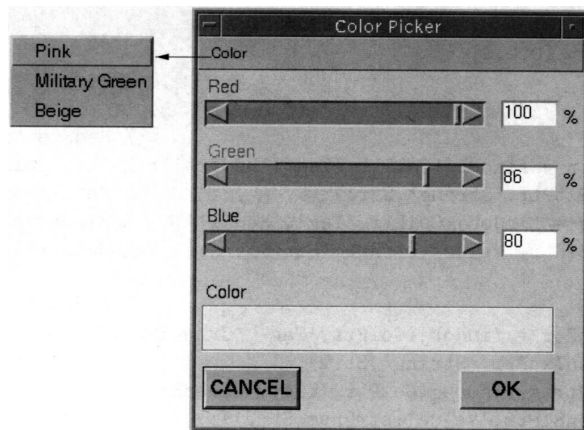


图14-27 添加菜单后的窗体

最后来看一下 ‘ CANCEL ’ 和 ‘ OK ’ 按钮。当用户按下 ‘ OK ’ 按钮时可以返回选定的颜色，所以还得在程序中编写相应的程序代码。而且，希望在开始时有一个初始颜色。如果用户按下 ‘ CANCEL ’ 按钮，就返回初始颜色。这样就要编写一个新函数，保存在 M 文件 uicolor.m 中：

```

function rgbout = uicolor(rgin)

ColorPicker;                                % 打开颜色混合器
set(0,'ShowHiddenHandles','on');            % 获取指针
ColPtr = findobj(0,'Tag','ColPick');

```

```

ColPtr = ColPtr(1); % 最新创建

if (nargin >= 1) % 检查输入内容
    if ((length(rgbin) ~= 3) | (min(rgbin) < 0) | ...
        (max(rgbin) > 1)) % 警告：非法输入
        warndlg('Invalid Colorspecific.','Warning');
        Str1 = '0'; Str2 = '0'; Str3 = '0'; % 用标准值代替
        InCol = [0,0,0];
    else % 输入
        Str1 = num2str(rgbin(1) * 100,3);
        Str2 = num2str(rgbin(2) * 100,3);
        Str3 = num2str(rgbin(3) * 100,3);
        InCol = rgbbin;
    end
else % 如果没有输入，就用标准值

    Str1 = '0'; Str2 = '0'; Str3 = '0';
    InCol = [0,0,0];
end

% 红色文本框
RedTxtPtr = findobj(ColPtr,'Tag','RedTxt'); % 获取指针
set(RedTxtPtr,'String',Str1); % 设置框体
RedSlPtr = findobj(ColPtr,'Tag','RedSl'); % 滑标指针
set(RedSlPtr,'Value',str2num(Str1)); % 设置滑标

% 绿色文本框
GreenTxtPtr = findobj(ColPtr,'Tag','GreenTxt');
set(GreenTxtPtr,'String',Str2);
GreenSlPtr = findobj(ColPtr,'Tag','GreenSl');
set(GreenSlPtr,'Value',str2num(Str2));

% 蓝色文本框
BlueTxtPtr = findobj(ColPtr,'Tag','BlueTxt');
set(BlueTxtPtr,'String',Str3);
BlueSlPtr = findobj(ColPtr,'Tag','BlueSl');
set(BlueSlPtr,'Value',str2num(Str3));

FrPtr = findobj(ColPtr,'Tag','col'); % 颜色框指针
set(FrPtr,'BackgroundColor',InCol); % 设置颜色框

% 将输入保存在UserData中
set(ColPtr,'UserData',InCol);

uiwait(ColPtr); % 等待响应
rgbout = get(ColPtr,'UserData'); % 获取选中的颜色
close(ColPtr); % 关闭颜色混合器
set(0,'ShowHiddenHandles','on'); % 隐藏指针

```

上面的程序可以停下来等用户选择颜色。初始颜色，一个 RGB 三元组，保存在颜色混合

器的UserData中，用户进行颜色选择时也从这里来选取。所以只需调用‘OK’按钮的事件过程函数来覆盖这个颜色就可以，因为如果用户按下‘CANCEL’按钮，返回的是初始颜色。在M文件CbColPin.m中添加两个事件过程函数：

```
function CbColPic(action)

switch(action)

...
case 'cancel'
    uiresume(gcbf);

case 'ok'
    Col = LocalUpdateCol;
    set(gcbf,'UserData',Col)
    uiresume(gcbf);

end

...
```

现在整个程序就全部完成了，可以调用uicolor来使用它(不是ColorPicker，因为用它只定义了窗体本身)。下面的这个例子用来说明如何使用这个程序来改变坐标轴中图形窗口的前景色和背景色。

```
function change

fp = figure;                % 创建一个图形窗口
ap = axes;                  % 创建坐标系

InCol = get(fp,'Color');    % 设置窗口的背景色
OutCol = uicolor(InCol);
set(fp,'Color',OutCol);

InCol = get(ap,'Color');    % 设置坐标系的前景色
OutCol = uicolor(InCol);
```

14.4 动画

14.4.1 介绍性示例

本节开始先介绍一个例子，做一个简短动画来演示例 13.8中的曲线振荡效果。

例14.22

```
x      = -8:0.5:8;          % 定义曲面
[XX,YY] = meshgrid(x);

r = sqrt(XX.^2+YY.^2) + eps;
Z = sin(r)./r;
```



```
surf(Z);                                % 画出帧

% 保存坐标值,使得所有帧都在同一个坐标系中

% 创建一个保存动画的矩阵,保存20帧

theAxes = axis;
fmat = moviein(20);

% 循环创建动画数据

for j = 1:20
    surf(sin(2*pi*j/20)*Z,Z)           % 画出每一步的曲面
    axis(theAxes)                       % 使用相同的坐标
    fmat(:,j) = getframe;               % 拷贝帧到矩阵中
end

movie(fmat,10);                         % 演示动画10次
                                        % 这很有趣!
```

可以看出,动画可以以帧的形式保存在矩阵的每一列中,之后再从矩阵中显示出来。在记录动画前要保存坐标轴的最小和最大值,这样就可以使每一帧都有相同的坐标轴。

14.4.2 拷贝图形窗口

命令 `getframe` 用来拷贝一个图形窗口到一个列向量中。

命令集172 拷贝图形

<code>getframe(p,r)</code>	拷贝句柄 p 对象中的图像。对象可以是屏幕、图形或坐标系。如果没有指定 p , 则使用当前图形窗口。可选参数 r 表示要拷贝对象的矩形区域。 r 是一个位置向量,有四个元素 $[left\ bottom\ width\ height]$ 。
<code>[X,Ftab]=getframe(...)</code>	返回图像矩阵 X 和色表 $Ftab$ 。
<code>[X,Ftab]=frame2im(F)</code>	将动画帧 F 转换成图像矩阵 X , X 的元素在色表 $Ftab$ 中给出。

14.4.3 创建动画

在记录动画前要给它的帧分配空间,这些帧以列的形式存储在矩阵中,用命令 `moviein` 可以创建这样的矩阵。

命令集173 制作动画

<code>moviein(n)</code>	创建一个可以保存 n 帧的矩阵。帧的大小由当前图形窗口的大小来决定,结果是必须先画出其中的一帧;见例 14.22。
-------------------------	---

`F=` 将图像矩阵 `X` 转换成一帧 `F`，`X` 矩阵的元素在色表 `Ftab` 中给出。这
`im2frame(X,Ftab)` 个命令可以用来从一个图形序列中创建动画。

14.4.4 演示动画

动画的演示借助 `movie` 命令，这个命令可以带上参数来表示动画演示的方式和位置。

命令集174 演示动画

`movie(p,Mat,n,fps, pos)` 连续演示 n 次动画矩阵 `Mat`，每秒 fps 帧。如果 `n` 是一个整数向量，帧就按向量中的顺序来显示。可选参数 `p` 是图形或轴对象的句柄，表示动画演示的位置。如果指定可选参数 `pos=(x, y)`，动画就在这个位置显示， x 和 y 是 `Units` 中的单位，相对于对象 `p` 的左下角的位置。