附录　MATLAB 绘图

图形功能在 MATLAB 中占据着核心地位。一般的图形显示只需要极少的几条命令即可实现，非常方便。借助一些更复杂的命令，MATLAB 可以将计算结果近乎完美地用图形展示出来。通过图形理解数学语言是一种令人愉快而又非常有效的学习数学的方法。事实上，甚至可以说只有理解了数学表达式的图形意义才是真正的掌握了它们。能够自由自在地绘制出数学函数和数据图形是学习数值分析重要的一步，此附录的目的正是让读者做到这一点。

当前，各个研究领域都会遇到大量数据，这些数据往往难以用数学公式表达。很多情况下可视化是对它们加以分析的唯一途径。这意味着我们需要根据情况选择适当的绘图方式，诸如一维、二维和三维绘图以及运动图象（即四维绘图）等。这是我们鼓励读者学习图形可视化方法的另一个原因。

学习图形命令与学习外语的情形非常类似（当然，前者远比后者容易）。仅仅通过记牢词汇和语法不可能掌握一种语言，练习才是关键。学习图形命令最好的方法即是循序渐进地上机实践。需要特别指出的是，我们不可能罗列 MATLAB 所有的绘图功能，这个附录里介绍的只是实际计算中一些最常用的命令。我们的目的是使读者迅速建立起 Matlab 绘图的基本概念，同时给出进一步学习所必需的提示。我们尽可能用实例进行说明。经常求助 help 命令是有益的，在命令窗口键入“doc”即可进入帮助页面，在那里可以得到所有Matlab 命令的详尽解释。

一．二维绘图

在第一章里，我们已经学习了基本命令”plot”。本节我们介绍二维绘图的其他几个常用命令。

1.1 基本命令

下面的程序结果如图 1.1 所示。

```
clear,  clf,  hold off      %  clf：清除图形窗口内容，相关命令有 cla, reset, hold
x=(0:0.4:10)';
y1=sin(x).*exp(-0.4*x);   y2=cos(x) .*exp(-0.4*x);
plot(x,y1,x,y2,’p‘);      %  p 表示五角星（其他如：d 为菱形，o 为圆形，^为三角形）
axis square      % 使所画图形的纵、横坐标刻度比例相同
xlabel('x'); ylabel('y=sin(x)\times e^{-0.4x}')      %  MATLAB 中可以使用 Latex
grid on
legend('y1=sin(x)\times e^{-0.4x}','y2=cos(x)\times e^{-0.4x}')      %  加入图例
```

figure: 该命令打开一个新的图形窗口，图形窗口按打开先后顺序编号，命令 figure(n)可直接指定创建窗口的编号为 n。如果存在多个图形窗口，必须知道哪一个是当前窗口。所有图形命令都作用于当前窗口。除非特别指出，最后打开的窗口即为当前窗口。键入命令figure(n)可使编号为 n 的图形窗口成为当前窗口。

close: close(n)关闭编号为 n 的图形窗口，close all 关闭所有图形窗口。

axis,axis on,axis off: 坐标轴的刻度范围和刻度值均可自动设定。不过，也可用 axis 命令改变这些参数。

zoom：通过 zoom 命令可以交互式地缩放图形的指定部分，无需使用 axis。在命令窗口

键入 zoom，然后在图形窗口按下鼠标左键，拖动鼠标将需要放大的部分用“取景框”围住。放开鼠标键便立即得到选定部分的局部放大图。再用鼠标点击图形窗口，则图形恢复到放大前的情形。

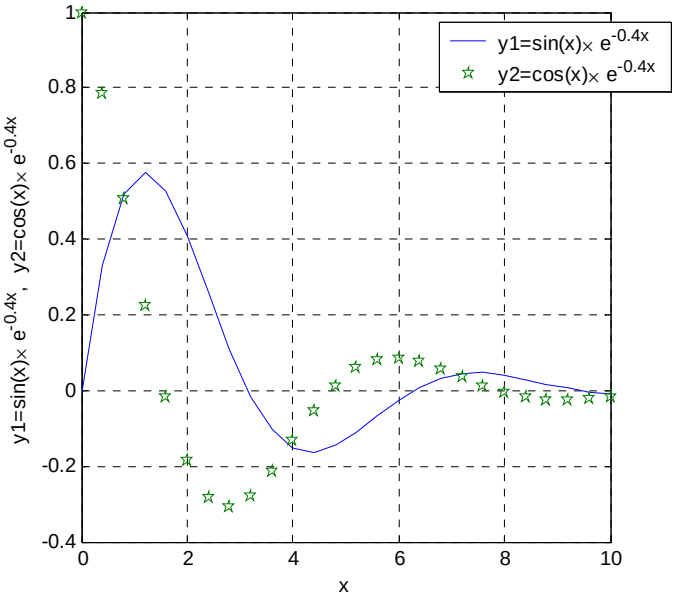

图 1.1 坐标轴标注与注释

1.2 极坐标绘图 命令 polar 绘制极坐标函数的图形。图 1.2 由下面程序画出。

```
t=0:.05:pi+0.01;
y=sin(3*t).*exp(-0.3*t);
polar(t,y)
title('Polar plot')     %   xlabel、ylabel 和 title 给图形加入坐标轴标注和标题
```

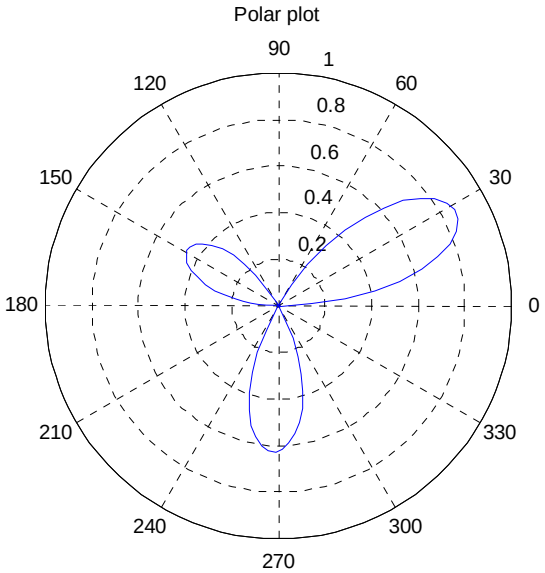

图 1.2 极坐标绘图

1.3 对数和半对数绘图 如图 1.3 所示。

```
t=0.1:0.1:3;
x=exp(t);
y1=exp(t.*sinh(t)); y2=exp(t.*t);
subplot(1,2,1); loglog(x,y1);grid, xlabel('x'); ylabel('y1')
subplot(1,2,2); semilogy(t,y2); xlabel('t'); ylabel('e^{t^2}')
```

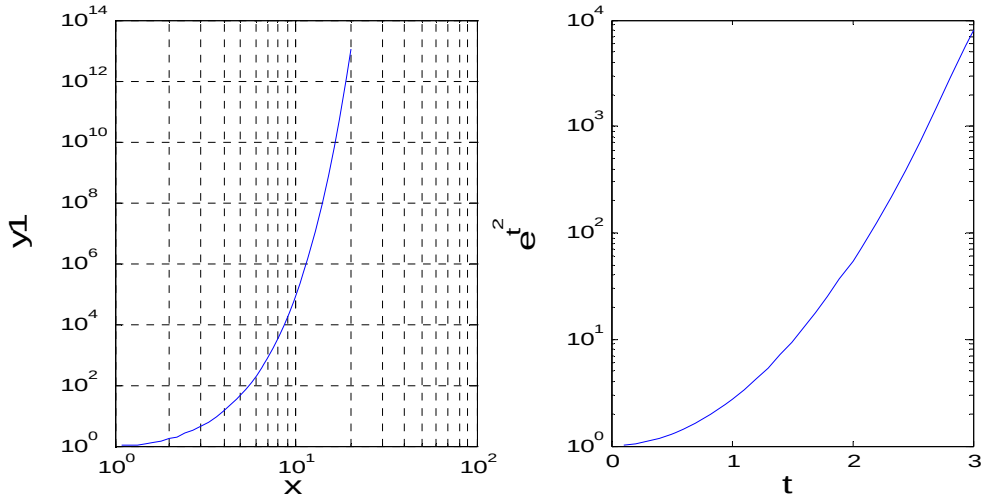

图 1.3 对数坐标图和半对数坐标图

二．三维绘图

2.1 绘制三维图形 绘制三维图形曲线和曲面最常用的命令是 plot3 和 mesh。下面是一个应用 plot3 的实例，如图 2.1 所示。

```
clear, clf
t=0:0.1:20; r=exp(-0.2*t);
th=pi*t*0.5; z=t; x=r.*cos(th); y=r.*sin(th);
plot3(x,y,z),   hold on
plot3([1,1],[-0.5,0],[0,0],'rs')      %   r 表示红色，s 表示正方形
text(1,-0.7,0, 'A');      %   文本标注
n=length(x);
text(x(n),y(n),z(n)+2, 'B')
xlabel('X'); ylabel('Y'), zlabel('Z');
```

二元函数可以通过 mesh 命令用离散点来表示，下面程序作出二元函数 $z = x \cdot e^{-x^2-y^2}$ 在区域$[-2,2]\times[-2,2]$上的图形，如图 2.2 示。

```
clear,clf
x =-2:.2:2;
y=-2:.2:2;
[X,Y]=meshgrid(x, y);      %   生成网格
Z =X.*exp(-X.^2-Y.^2);
```

```
mesh(X,Y,Z);
title('这是函数 z=x*e^{-x.^2-y.^2} 对应的三维图形 ');
xlabel('x'); ylabel('y'); zlabel('z');
```

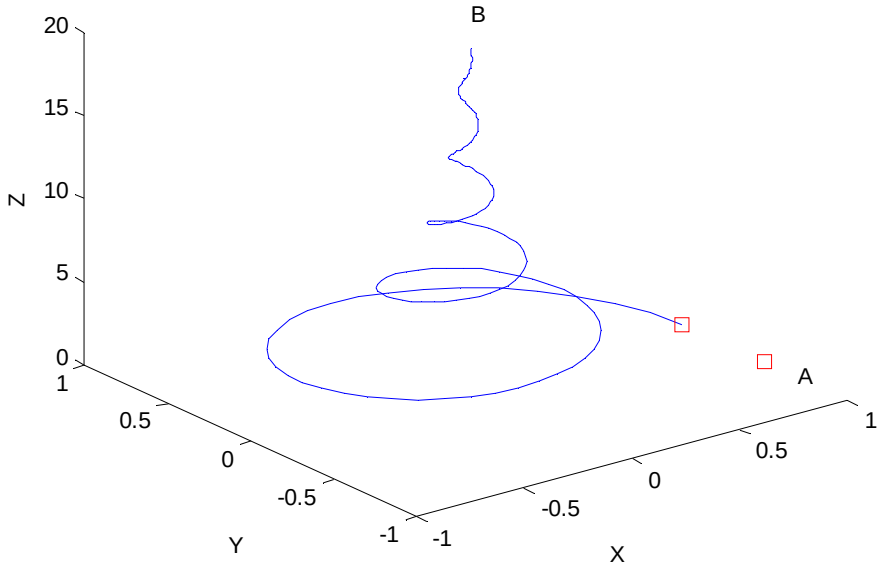

图 2.1　plot3 画的三维曲线图

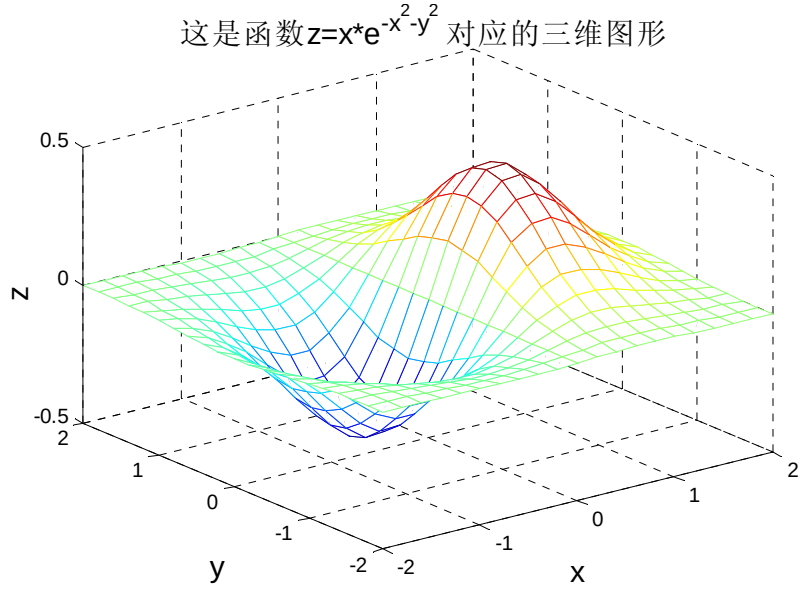

图 2.2 二元函数的网格图

默认颜色：在彩色屏幕上，连接点与点之间的直线由默认颜色设置 hsv（带饱和值的色图）着色。把 z(i,j)的最大值和最小值处均设置为红色，在最大值和最小值之间按照红、黄、绿、青、蓝、紫再到红的顺序均匀着色。以如下程序为例，其结果如图 2.2 所示。

```
clear;clf
for i=1:4
    for j=1:7
        z(j,i)=sqrt(i^2+j^2);
    end
end
mesh(z)
xlabel('x');ylabel('y');zlabel('z')
```

在彩色屏幕上，图 2.3 最大值处的直线是紫色，最小值处是略带红色的黄色。这是因为直线的颜色由其两端点处 z 值的平均值决定。对于当前的网格图，最大值处直线的颜色值接近紫色，最小值处直线的颜色值接近黄色。如果按更细的网格绘制此图，则最大值和最小值处的直线均变为红色。

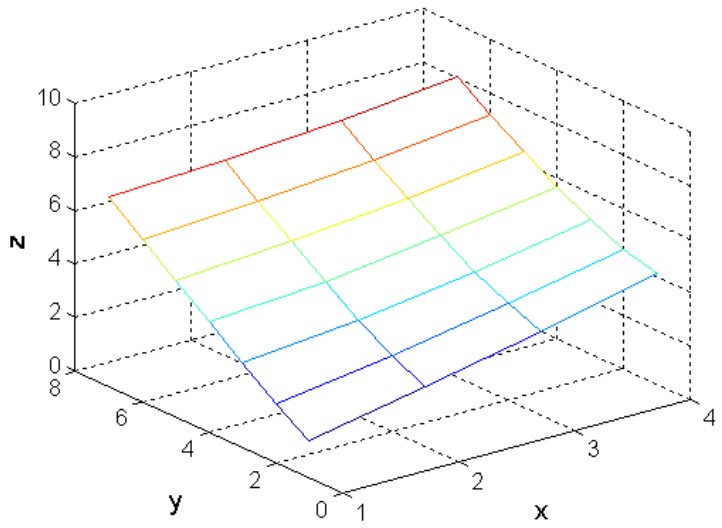

图 2.3 曲面的网格图

2.2 等高线

contour：等高线命令 contour 调用格式为：contour(x,y,z,level)，其中 x、y 和 z 的含义与 mesh 中的相应参数完全一样。level 是表示等高线高度的数组，它也可以是一个整数 m，此整数表示等高线的数目。等高线的高度取为将 z 的值域等分 m-1 份所得到的 m 个等分点的值。图 2.4 显示了函数 $z = x \cdot \exp(-x^2 - y^2)$ 的等高线图。等高线图必须加有注释说明等高线的高度。clabel(h)是自动注释命令，clabel(h,'manual')为手动注释命令。

```
clear, clc, clf, axis square
xm=-2:.2:2; ym=-2:.2:2;
[x,y]=meshgrid(xm, ym);
z=x.*exp(-x.^2-y.^2);
zmax=max(max(z)); zmin=min(min(z));
```

```
dz=(zmax-zmin)/10;
level=(zmin+0.5*dz:dz:zmax);
h=contour(x,y,z,level); clabel(h, 'manual');  %  手动注释，用鼠标选择高度值的位置
title('Contour plot by contour(x,y,z,level)')
xlabel('x'); ylabel('y');
```

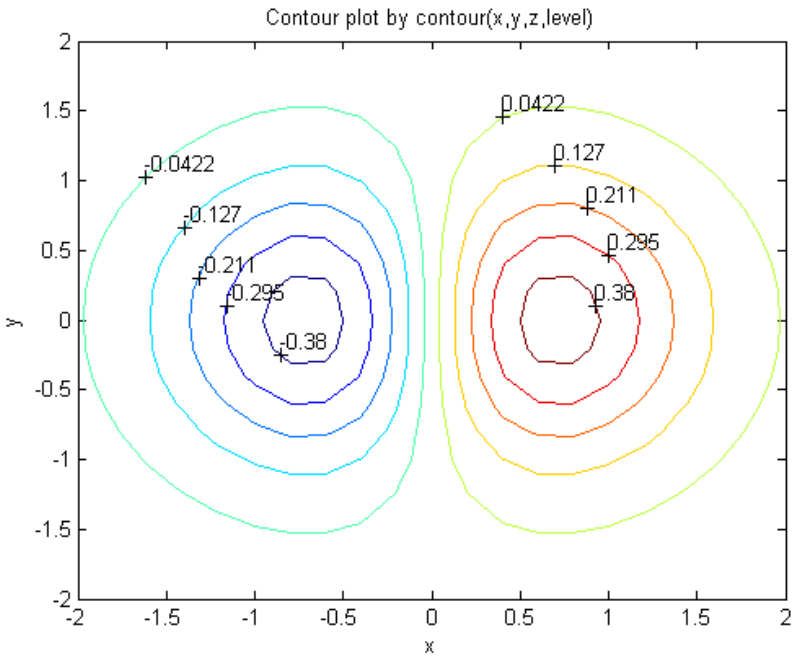

图 2.4　　等高线图

隐函数绘图：contour 命令还可以用来绘制隐函数的图形，例如：$y^3+\exp(y)=\tanh(x)$。为了绘制曲线，将方程重新写为 $f(x,y)=y^3+\exp(y)-\tanh(x)$。只画出对应于 f=0 的等高线图，如图 2.5 所示。

```
clear , clf
xm=-3:0.2:3; ym=-2:0.2:1;
[x,y]=meshgrid(xm,ym);
f=y.^3+exp(y)-tanh(x);
contour(x,y,f,[0,0])
xlabel('x'); ylabel('y');
```

注意到上面的命令文件里，将 contour 命令中表示等高线高度的参数向量设定为[0 0]。虽然我们感兴趣的只是 0 值等高线，但是由于等高线的高度一定要以向量形式表示，所以上述文件中把 0 写两遍变成一个向量进行处理。

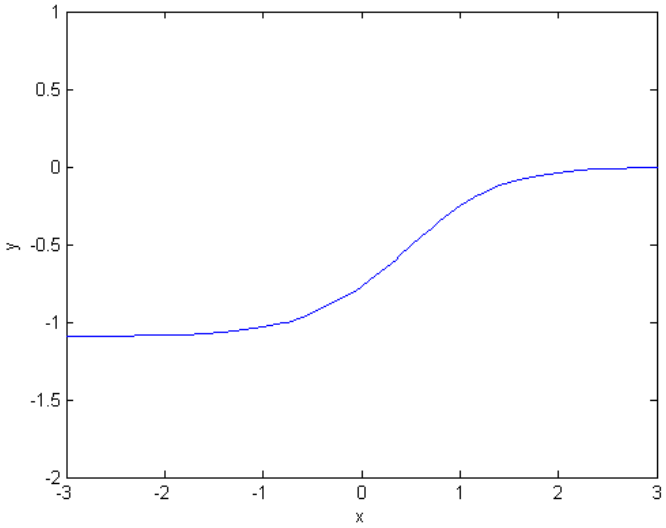

图 2.5　隐函数图

三维网格图的等高线：meshc 在 x-y 平面上绘制 z 的等高线，其对应的三维网格图是 mesh(z)，如图 2.6 所示。

```
clear, clf, hold off
j=1:21; i=1:10;
x=log(i); y=log(j);
[x,y]=meshgrid(x,y);
z=sqrt(0.1*((x-log(5)).^2+(y-log(5)).^2))+1;
meshc(x,y,z)
xlabel('x'); ylabel('y'); zlabel('z');
```

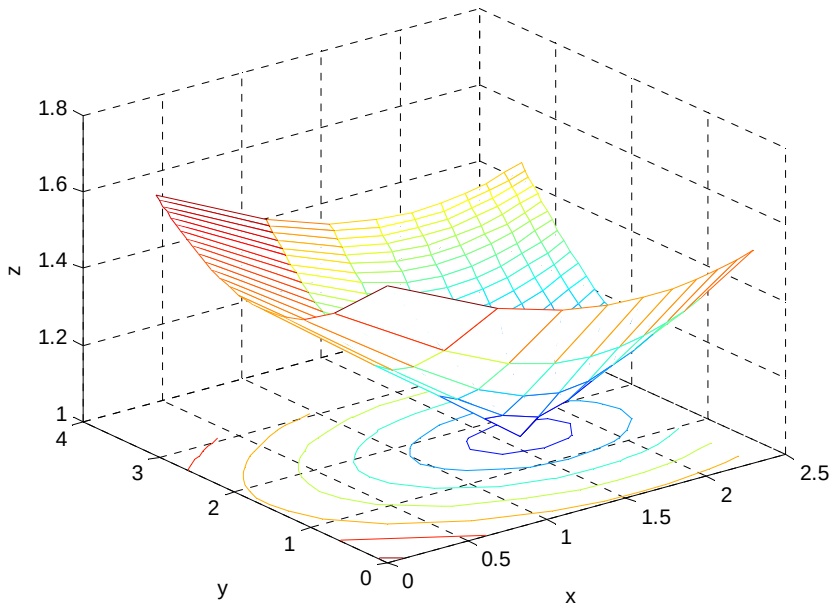

图 2.6　一个矩阵的网格和等高线

三维表面与等高线： surf 和 surfc 命令绘制的图形与 mesh 和 meshc 所绘图形类似。区别仅在于 surf 和 surfc 将所有小的三维网格区域着色，从而得到三维表面图。小网格区域的颜色由其四个顶点处 z 的平均值决定。

带有亮度的三维表面：surf 命令可以用 surf1 来代替，所不同的是后者绘制的三维表面图带有亮度变化。光源的方向可以指定。例如，基本用法 surf1(x,y,z)的亮度采用默认值。特定的光源方向可以用 surf1(x,y,z,s)指定，其中 s 是方向向量。建议将 surf1 与 colormap gray 和 shading flat 或 shading interp 共同使用。

色轴：mesh 和 surf 命令里的第四个参数可以控制图形的颜色；例如：mesh(x,y,z,c)第四个参数 c 指定颜色坐标，它是一个与 z 大小相同的向量。如果 c 省略，则默认 c=z，从而直线和曲面的颜色由 z 的值决定。

色图由 colormap 命令设置，其默认状态为 colormap(hsv)。此时，c(j,i)的最大值点和最小值点均设为红色。最大值和最小值之间的点按照红、黄、绿、青、蓝、紫再到红的顺序均匀着色。不过，用户可以通过定义 c 向量给网格点赋不同的颜色值。命令 caxis([0,100]) 将色图设置为 0 到 100 之间。如果将颜色矩阵的元素都设定为接近 0 的数，则整个图形将呈微红色，如果颜色矩阵的元素都在 50 左右，则整个图形将变成带有点蓝色的紫色。

色图：可以通过 colormap 命令改变色轴上的颜色定义。除了 colormap(hsv)，可供选择的色图定义还有 colormap(hot)、colormap(cold)和 colormap(jet)。读者可以尝试用这些命令改变三维网格或表面图的颜色，以熟悉它们的性质。

shading：surf 创建的图形对象由许多用黑线分隔而成的小四边形组成。这对应着 shading 命令的默认情形，即 shading faceted。shading flat 命令去掉边框线。Shading interp 去掉边框线，同时让表面的颜色均匀过渡。

再谈 hold on：对于非常费时的绘图操作，hold on 命令非常重要。原因是：图形绘制过程中，有些命令会改变一些图形属性，诸如轴的控制设置、色图、视角、色轴等。每用一条这样的命令，整个图形将重绘一遍。所以绘图前先给出所有属性命令，然后用 hold on 保持可以大大节省时间。

三．统计绘图

这一节，我们介绍统计数据的图形表示。

3.1 条形图 共有四条命令绘制 2 维和 3 维条形图，它们是 bar、barh、bar3 和 bar3h。下面程序为二维条形图实例，如图 3.1 所示。

```
clear, close
X=0:5:20;   Y=[7 6 5; 6 8 1; 4 5 9; 2 3 4; 9 7 2];      %  记 Y 的行数为 M=5，列数为 N=3
subplot(2,2,1)
bar(X, Y)      %  按行将 Y 分为 5 组，以直方图显示每一组各元素的取值，向量 X 必
               %  须按升序或降序排列.
title('基本 bar 图')
subplot(2,2,2)
bar(Y)      %  和 bar(1:M, Y)等价
```

```
title('基本 bar 图')
subplot(2,2,3)
bar( Y,'stacked')      %  绘制垛图，即将 Y 的一行元素表示在一个直方条中
title('垛图')
subplot(2,2,4)
barh( Y)      %  水平条形图
title('水平条形图')
```

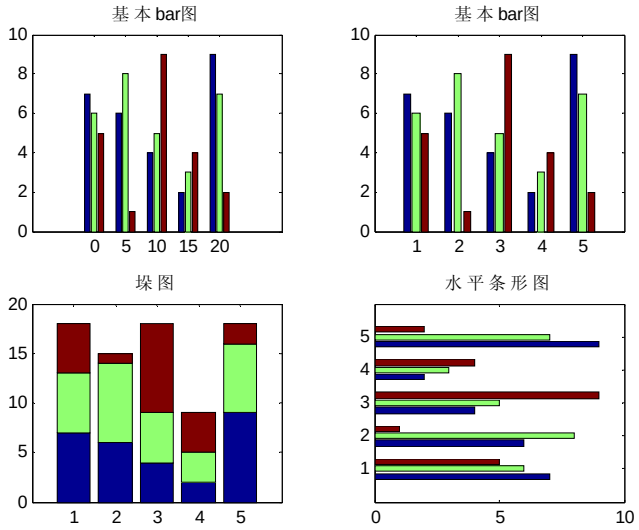

图 3.1　2 维条形图

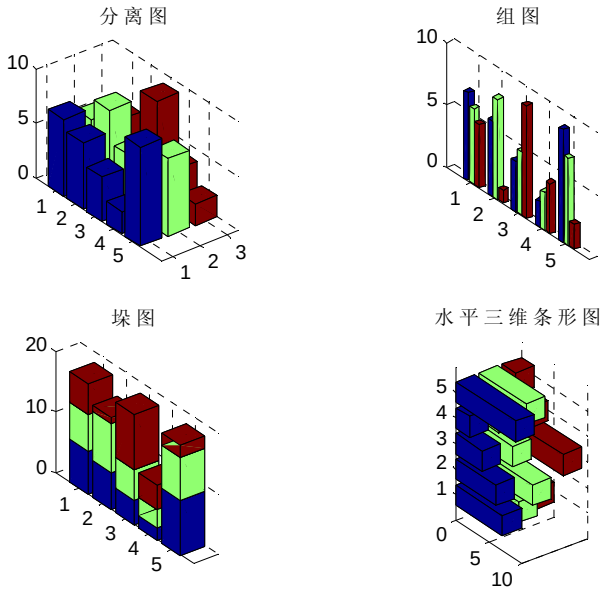

图 3.2　三维条形图实例

图 3.2 是三维条形图实例，由下面的程序实现。

```
clear, close, Y=[7 6 5; 6 8 1; 4 5 9; 2 3 4; 9 7 2];
subplot(2,2,1)   bar3(Y)   title('基本三维条形图')
subplot(2,2,2)   bar3(Y,'grouped')   title('组图')
```

subplot(2,2,3)　bar3(Y,'stacked')　title('垛图')
subplot(2,2,4)　bar3h(Y)　title('水平三维条形图')

3.2 柱状图　柱状图用 hist 命令生成，其功能为图形显示落入指定区间的元素个数。图 3.3 是由下面程序实现的柱状图。

```
randn('state',1)　% 以状态 1 生成正态随机数，设定不同状态以避免随机数列重复
y=exp(randn(1000,1)/3);
subplot(2,2,1);
hist(y);　%　在1 0（默认值）个等分区间内画出x中数据的统计频数直方图。
title('输入数据为 1000×1 的向量, 10 等分区间')
subplot(2,2,2); hist(y,25);　%　指定等分区间个数为 25
title('25 个等分区间')
subplot(2,2,3); hist(y,min(y):.1:max(y));
title('等分区间宽度设为 0.1')
y=exp(randn(1000,3)/3);
subplot(2,2,4); hist(Y); title('输入数据为 1000×3 矩阵');
```

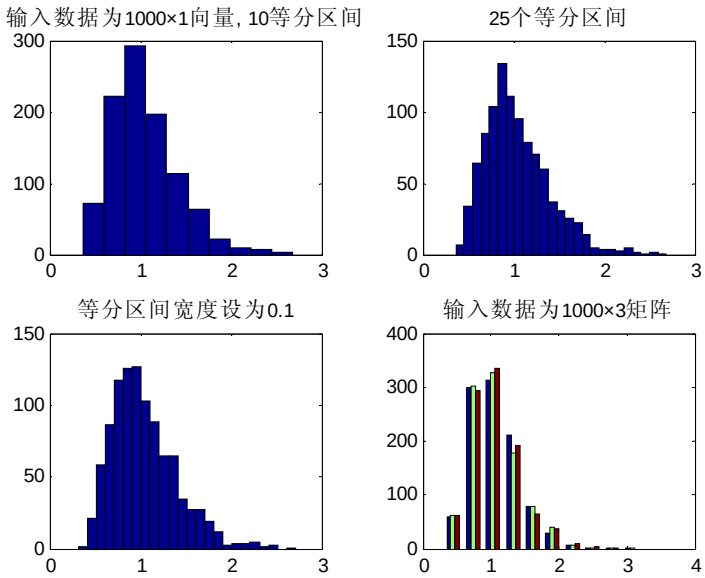

图 3. 3　柱状图

3.3 圆盘图　pie 和 pie3 分别绘制平面和立体的圆盘图，如图 3.4 所示。

```
x=[1.5 3.4 4.2];
subplot(2,2,1); pie(x);　% 按照各分量大小划分圆盘的面积，显示各部分百分比
subplot(2,2,2); pie(x,[0 0 1]);　% 使得 1 位置对应的部分从圆盘中分离
subplot(2,2,3); pie(x,{'Slice 1','Slice 2','Slice 3'});　% 给各部分命名
subplot(2,2,4); pie3(x,[0 1 0])　　%　立体圆盘图
```

3.4 区域图　区域图命令为 area．它的功能和 plot 相似，所不同的是区域图中曲线与 x 坐标轴所夹区域将被填充。

```
randn('state',1); x=[1:12 11:-1:8 10:15]; Y=[x' x'];
```

```
subplot(2,1,1); area(Y+randn(size(Y)));
subplot(2,1,2); Y=Y+5*randn(size(Y));
area(Y,min(min(Y))); axis tight
```

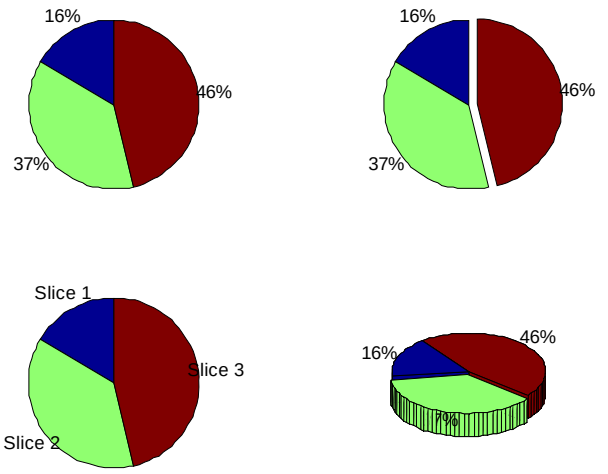

图 3. 4　圆盘图

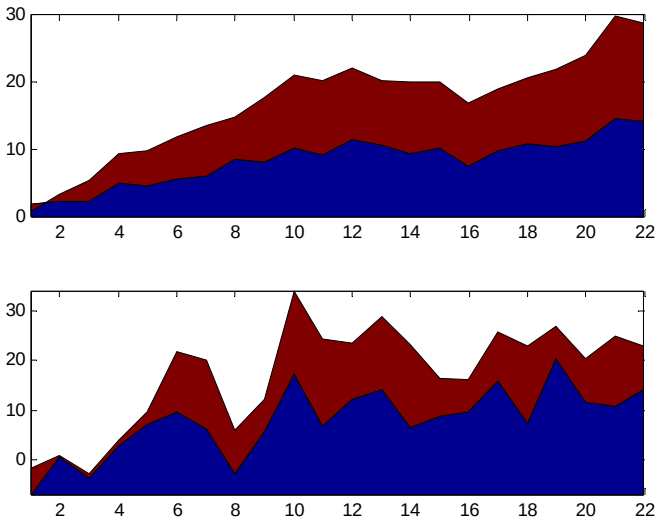

图 3.5　区域图

四．图形句柄

前面几节所介绍的图形函数已经足以应付一般性的需要。如果希望对图形对象做一些底层的操作，则必须借助图形句柄来实现。

Matlab 规定每一个图形对象被赋与一个特定的数值，这个数值即称为图形句柄。

命令　a1=plot(1:10, 'o-')　得到图 4.1（左）。下面将以此图为例仔细考察如何对图形对象进行控制，操作结果如图 4.1（右）所示。

```
>> h=findobj       %   获得所有图形对象的句柄
h =          0     %   Matlab 约定第一个句柄表示根对象，即屏幕，此值永远是 0
     1.0000        %   此句柄表示图形窗口
   101.0009
     3.0010
>> get(h,'type')   %   获得所有句柄的含义
ans =   'root'
        'figure'
        'axes'      %   由此可知 h(3)是坐标轴对象的句柄
        'line'      %   h( 4 )表示线对象。
```

通过句柄，可以直接得到访问对象的各种属性，从而做到只改变某一特定属性且不对其他对象和属性产生影响。

```
>> set(h(3))       %   获得指定图形对象的所有属性以及每一属性可能的状态
    ALim
    ALimMode: [ {auto} | manual ]      %   方括号中列出所有可能的状态
    AmbientLightColor
    Box: [ on | {off} ]                %   花括号内为该属性的默认状态
    CameraPosition
    CameraPositionMode: [ {auto} | manual ]
            ……                         %   这里省略了大约 80 行内容
    Visible: [ {on} | off ]
```

同样，键入 set(h(4)) 将得到线对象的所有属性信息。

```
>> set(h(3),'XScale')      %   对于指定属性的状态，也可用 set 命令直接得到
[ {linear} | log ]
>> set(h(3),'XScale','log')    %   将 x 坐标轴改为对数坐标
>> set(h( 4 ),'Marker','s','MarkerSize',16)      %   改变线对象的顶点标记符号和大小
```

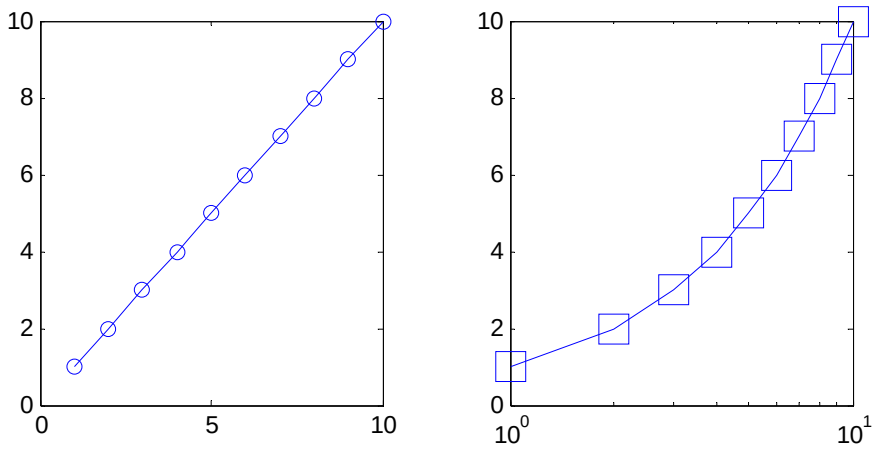

图 4 . 1　句柄操作实例

一般说来，如果打算修改对象的属性，需要在创建它们的时候保存句柄，这样即可利用句柄对图形做底层的操作。不过，即便没有做这样的准备仍有办法使用句柄。对于已经创建的对象，其句柄可以用'gca'、'gcf' 和 'gco' 代替。其中，'gca' 对应当前坐标轴，'gcf' 对应当前图形窗口，'gco' 对应当前的图形对象（即最近一次鼠标点击的对象）。

运行下述命令，结果如图 4.2 所示.。

```
x=linspace(0,2*pi,35);
a1=subplot(2,1,1);          %  坐标轴对象
l1=plot(x,sin(x),'x');      %  线对象
a2=subplot(2,1,2);          %  坐标轴对象
l2=plot(x,cos(x).*sin(x));          %  线对象
tx2=xlabel('x');ty2=ylabel('y');    %  文本对象
```

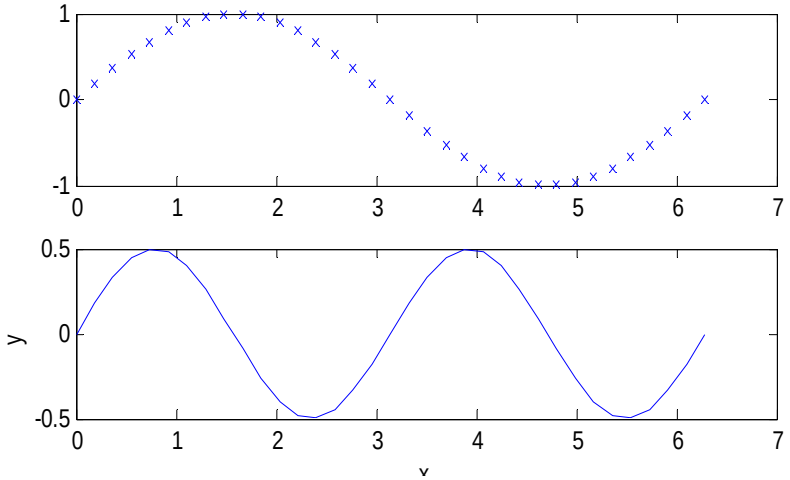

图 4 . 2　直接使用 subplot 所绘图形

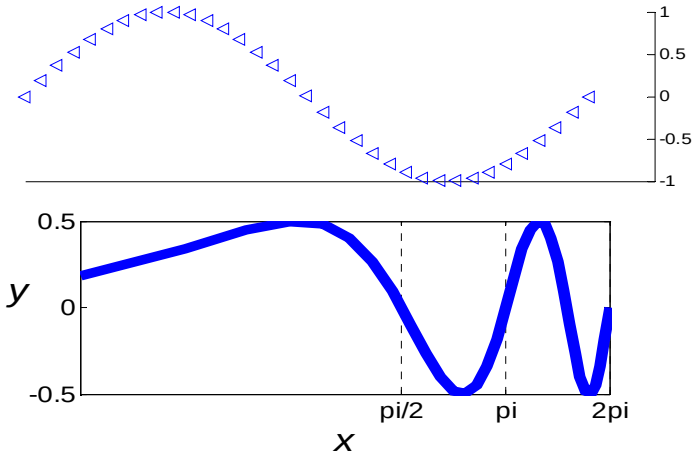

图 4 .3　使用图形句柄对图 4.2 操作后的结果

下面的命令将它们改变为图 4.3 所示的效果。

```
set(a1,'Box','off')      %  去除坐标轴边框
set(a1,'XTick',[])       %  去除坐标轴刻度
set(a1,'YAxisLocation', 'right')      %  将 y 轴移到坐标系的右边
set(a1 ,'TickDir','out')    %  坐标轴刻度标记的指向，,'out'表示指向坐标轴外侧
set(l1,'Marker','<')

set(a2,'Position',[0.2 0.15 0.65 0.35])      %  坐标系所在的矩形区域
set(a2,'XLim',[0 2*pi])      %  x 坐标坐标轴的可见范围
set(a2,'FontSize',14)        %  指定刻度值显示的字号
```

```
set(a2,'XTick',[0 pi/2 pi 2*pi])       %   坐标轴的刻度位置
set(a2,'XtickLabel','0|pi/2|pi|2pi')   %   坐标轴的刻度位置的标识
set(a2,'XGrid','on','XScale','log')    %    XGrid 控制坐标轴的网格线
set(l2,'LineWidth',6)     %   指定线的宽度
set(tx2,'FontAngle','italic','Fontsize',20)   %   指定文本的字体和字号
set(ty2,'Rotation',0)   %   指定文本对象的旋转角度
set(ty2,'FontAngle','italic','FontSize',20)
```

五．其他

5.1 图形的交互式编辑　在 MATLAB6 里，前面提到的所有图形编辑命令都可在图形窗口中通过交互式操作实现。另外，连接图形里标题与文本的箭头和直线也可以交互式地加入。在下面的解释中，假设是在对一个已有图形进行操作。

直线的属性：在一个图形里，可以交互式地改变其颜色、线宽和曲线类型。用鼠标点击菜单栏左上方的箭头符号按钮，移动鼠标使其指向所要编辑的曲线上的任意一点。按下鼠标，曲线变为一个个连续的黑点。在相同的位置双击鼠标，会打开一个属性编辑窗口，曲线的属性可在此窗口中加以改变。如果双击鼠标后没有出现属性窗口，则回到编辑菜单选择 Current Object Properties。

图例：在图形窗口里，点开菜单栏中的下拉菜单 insert 可以交互式地加入图例。点击 legend，图形窗口右上角将出现一个基本的图例框。首先点击图例框中的 text，然后进入 Edit 菜单，拉下此菜单，选择 Current Object Properties。一个属性编辑窗口将被打开，可以在此窗口中改变包括文本标注和字号在内的图例属性。当属性编辑窗口中的编辑工作结束后，点击 Apply 即可。

文本标注：点击菜单 Indert 选择 Test 或按下菜单栏里的按钮 A。将鼠标指针移动到文本标注的开始处，然后点击，将出现一个小的文本框，此时可以立即写入文本标注。点中文本框后拉下菜单栏的 Edit 菜单，可以改变该文本标注中诸如字号等的任何属性。选择 Current Object Properties，在这个属性编辑窗口中，用户可以指定字体、字号以及其它任何希望改变的属性。

坐标轴标注和标题：这些属性的使用过程和 legend 与 text 非常相似。从 Insert 菜单开始，选择适当的项目。

箭头：可以通过交互式操作画出图形窗口里的箭头符号。类似于添加文本标注，此项操作也从 Insert 菜单开始。在下拉菜单中选择 Arrow，然后移动鼠标指针到箭头符号的开始处做点击，拖动鼠标指针到箭头所指的地方，松开鼠标按钮。同样的方法适用于无箭头的直线连接。

轴属性：用户可以改变轴线的宽度和刻度标注的字号。点击图形内部或轴的边缘，进入 Edit 菜单，选择 Current Object Properties。在属性编辑窗口里对刻度标注的字号或轴线的宽度进行修改。

5.2 打印和记录图形　简单地键入 print 即可在当前打印机打印出当前窗口下的图形。要想

创建一个图形文件，以便打印或在其它文档中使用，必须首先指定其格式。例如，创建一个 PS 格式的文件，使用命令：

print -dps filename　或　print filename.ps

此操作将创建一个文件名为 filename.ps 的 PS 文件。再如 JPEG 格式的命令如下：

print -djpeg filename

上述命令创建一个名为 filename.jpg 的文件，此文件可用于 Web 页面、Word 和 PowerPoint。JPEG 格式的图形文件可以转换为 GIF 格式，从而制作出动画图象，这方面的详细情况将在附录 C 介绍。使用 help print 可以得到有关其它图形格式更多的信息。

图形的大小和位置：通过 set 命令可以改变图形打印在纸上的大小和位置。具体做法如下：首先用命令 h=figure 打开一个图形窗口，其中 h 是图形窗口的句柄；然后，在使用 plot 命令之前或之后键入命令 set(h,'PaperPosition',[a,b,c,d])，其中 a、b、c 和 d 是以英寸为单位的数字，a 为打印页的左边空白量，b 为右边空白量，c 和 d 分别为打印页上图形的宽度和高度。'PaperPosition'属性的默认值是[0.25,2.5,8,6]。图形的其他参数也可以通过类似的方法改变。键入命令 get(h)将得到所有图形参数的当前设定值。

5.3 交互式图形功能：交互式图形功能里最基本的内容是鼠标响应功能，即可以通过程序读取鼠标指针所处的位置坐标。ginput 就是其中的一个函数，它有如下几种使用格式：

[x,y]=ginput

[x,y,button]=ginput

[x,y,button]=gingput(n)

假设在屏幕内的所需位置上点击鼠标，则[x,y]=ginput 可以累计任意多的点坐标，直至按下回车键。所以 x 和 y 都是向量，它们的长度与所累计的点数相等。[x,y,button]=ginput 还能记录鼠标的键值，其余功能与[x,y]=ginput 同。鼠标的三个键自左至右键值分别为 1、2 和 3。[x,y,button]=ginput(n)将累计 n 个点，按下回车键可以使其中止。例如：

```
clear, clf, hold off
axis([0 10 10]); hold on; plot([1 2 2 1 1], [2 2 3 3 2]);
text(1,1.6, 'Click inside the box to teminate')
while 1
    [x,y,button]=ginput(1);
    if button= =1, plot(x,y, '+r'), end
    if button= =2, plot(x,y, 'oy'), end
    if button= =3, plot(x,y, '*g'), end
    if x>1 & x<2 & y>2 & y<3, break; end
end
hold off
```

这个例子说明了 ginput 的调用格式。执行此程序，它会等待鼠标点击。如果按下鼠标左键，一个红色的'+'号将出现在屏幕上的点击处。类似地，点击中键或右键时，屏幕上会分别出现黄色的'O'号或绿色的'*'号。用鼠标点击屏幕左下角方框中的光标可使执行中止。图 2.34 显

示了 List2.31 绘制的符号。

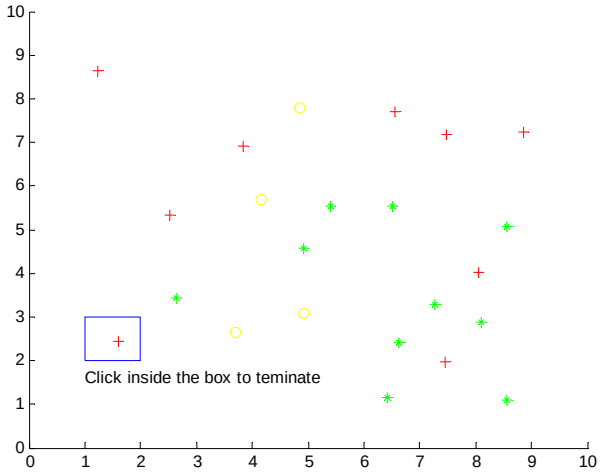

图 5.1　鼠标相应功能实例

5.4　动画　动画的一个很自然应用就是显示与时间有关的现象。动画还可以用来从不同角度并以不同比例显示复杂对象。动画的原理很简单；那就是连续显示一组图形或图像。本节将通过实例说明如何制作和显示动画。用 MATLAB 制作动画简单而有趣。

例 1．用 getframe、moviein 和 movie 等命令实现动画

```
clear
Z=peaks; surf(Z); axis tight; disp('Create the movie');
pause;  % 暂停命令，按下任意键后继续执行后面命令
for j=1:11
    surf(cos(2*pi*(j-1)/10).*Z,Z);   F(j)=getframe;
end
disp('Play the movie')
pause
movie(F)  %  播放动画
```

getframe：getframe 命令从当前图形窗口俘获一帧图片，并将每帧图片写入一个列向量，它有如下两种格式：geframe，geframe(h)。第一种格式俘获当前图形窗口的全部内容。第二种格式里的 h 是根对象、图形窗口或坐标轴的句炳。所以 geframe(h)得到的是句柄 h 所对应的对象或窗口。

moviein：M=moviein(n)创建矩阵 M，此矩阵保存当前图形窗口的 n 帧图片。M=moviein(n,h)创建矩阵 M，此矩阵保存来自句柄 h 所对应的对象或窗口的 n 帧图片。

例 2．用对象的 XData,YData,Zdata 属性实现动画

```
x=linspace(-pi,pi,2000);
y=cos(tan(x))-tan(sin(x));
p=plot(x(1),y(1),'.','EraseMode','none','MarkerSize',5);
axis([min(x) max(x) min(y) max(y)]);
hold on;
```

```
for i=2:length(x)
    set(p,'XData',x(i),'YData',y(i))
    drawnow   %   完成待做的绘图事件，更新图形窗口
end
hold off
```

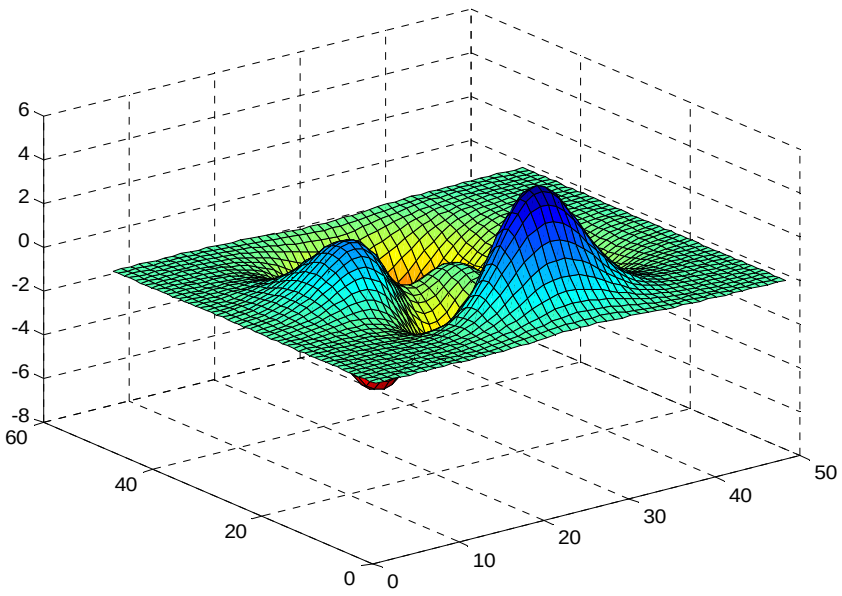

图 5 .2　例 1 动画中一帧图片

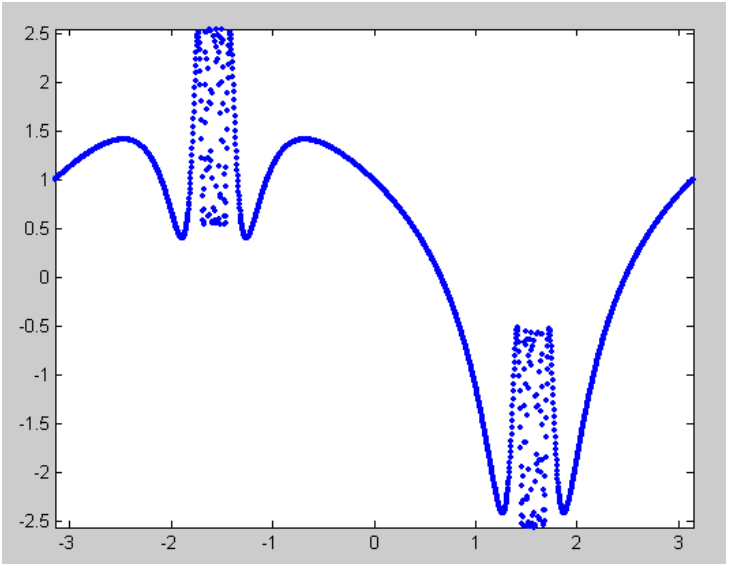

图 5 . 3　例 2 动画中的最终结果

5.5 声音　Matlab可以使用sound命令制作声音向量。

sound (y)　将向量y传送给扬声器。向量确定了最大振幅。

sound (y,f)　与上面的命令相似，而且还设定采样频率为f Hz。该命令不能用于SUN公司

的SPARC工作站。

soundsc(x,f,slim) 采用与sound相同的方式播放向量x，但是shoundsc给出的声音向量，声音可以尽可能地大。f表示采样频率，此项可以缺省。这里slim设定满音量范围，缺省值为[min(x) max(x)]。

例1 正弦波的声音：

```
x=sin(linspace(0, 10000, 10000));      %  正弦波
sound ( x );      % 产生声音。
```

例 2 用load命令载入几个预定义的声音。下面试试其中的两个。

```
load train;   % 装入产生火车汽笛的声音数据
whos ;   % 显示变量y , Fs。向量y表示创建的信号；标量Fs表示以Hz为单位的频率
sound(y);   % 产生声音
load chirp;   % 装入产生鸟儿唧唧喳喳声音的数据
sound(y);   % 产生新的声音
```

在某些系统中还有一些附加的声音命令；用help sounds可以了解这些特殊的系统。