第6章 在Mathematica 中作图

6.1 二维函数作图

6.1.1 二维函数作图命令 Plot

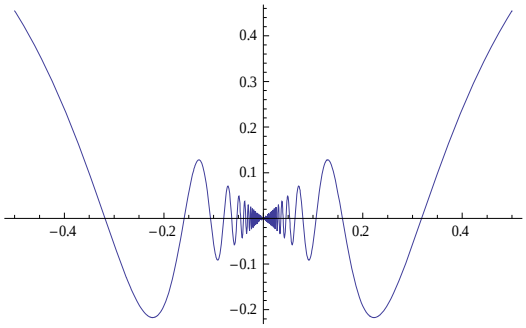
Plot 命令的一般形式：

Plot[f,{x,xmin,xmax}, 选项]

Plot[{f1,f2,…},{x,xmin,xmax}, 选项]

例题：

```
In[1]:=Plot[x Sin[1/x],{x,-0.5, 0.5}]
```




```
In[2]:=Plot[(x^2-x)Sin[x],{x,2,16},AxesLabel->{"x","f(x)"}
       ]
```

（* 给 x、y 坐标轴分别加标记"x", "f (x)" *）

```
In[4]:= Plot[Sin[x],{x,0,3},Frame→True,
            GridLines → Automatic]
```

（* 给图形加上框线和网格 *）

Plot 函数的第一类可选项，告诉系统如何显示图形，以及对坐标轴、刻度等细节的处理。

AspectRatio 图形的高度与宽度的比例 ，默认值是 1/GoldRatio，其中 GoldRatio=0.618。

如果要图形按实际情况显示，设置的选项值是 Automatic。

Axes 是否画坐标轴以及设置坐标轴的中心位置，默认值是 True，画出坐标轴 。

Axes -> None 不设坐标轴；Axes -> {x0,y0} 设置坐标轴中心为{x0,y0} 。

AxesLabel 设置坐标轴上的标记符号。默认值是 None，不做标记。

用{“字符串 1”，“字符串 2”}的形式定义轴的横坐标和纵坐标标记。

Frame 在图形周围是否加框。默认值是 False；Frame -> True 画出边框。

Ticks 设置坐标轴上刻度的位置，默认值是 Automatic，由系统自动定位。

Ticks->None 不标坐标刻度；Ticks-> {xi,yi} 规定 x 轴和 y 轴的刻度值，

FrameLabel 是否在框的周围加标志。默认值是 None。

FrameLabel ->{xmlab, ymlab, xplab, yplab}从底边开始按顺时针方向，设置外框的边缘名称。

PlotLabel　图形的名称标志。默认值是 None，不列标志。
PlotLabel ->lab 则规定图名是 lab。任意输出格式给出的表达式都可作为图名。字符串用"text"的形式给出。
PlotColor　是否产生彩色颜色。默认值是 True。
DisplayFunction　说明用什么机制显示图形。默认值 $DisplayFunction，其意义是立即在屏幕上显示图形。如果要在 Plot 中不输出图形，则再现图形时则需要设置选项 DisplayFunction -> $DisplayFunction 。
PlotRange　指定绘图的范围。系统用默认值时会自动切除区间奇点附近区域的曲线。
PlotRange -> All 画出所有点； PlotRange ->{y0,y1} 画出函数值在[y0,y1]范围内的图；
PlotRange ->{{x0,x1},{y0,y1}} 画出区间在[x0,x1]，函数值在[y0,y1] 的图形。

以上是请上机观看 DisplayFunction 在演示图形中的效果：
In[1]:= Plot [x^3-2, {x,0,10}, DisplayFunction –> Identity]
In[2]:= Show [%, DisplayFunction –>$DisplayFunction]

6.1.2曲线样式

Plot 的第二类选项用于控制图形的生成过程，设置怎样构造图形元素。例如：设置加大画图取样的点数，设置曲线的颜色等特性。下列 Plot 的第二类选项及其意义 ：

选　项	默认值	说明
PlotPoints	25	采样函数的点数，对于函数值变化剧烈的表达式，应设定较大的点数
PlotStyle	Automatic	设置曲线的样式。可设置曲线的颜色、线条的高度和虚实等形式。默认值画出一条黑色实线的曲线。
MaxBend	10	曲线相邻线段之间的最大夹角。当相邻的两段折线之角的折角大于 MaxBend 的值时，系统自动增加一些中间点，使折线变的更加光滑。
PlotDivision	20	对函数取样时细分区间的最大因子，由于有些函数具有无穷振荡的图形，为了避免 MaxBend 一直增加中间点以满足折线的夹角要求而陷入死循环。

1

我们称曲线的颜色、曲线的线形和线的宽度等特性为曲线样式。下列用于设置曲线的样式选项 PlotStyle 的调用形式和选项值。
GrayLevel[g] 灰度比值，g 取 0 到 1 之间的数。
RGBColor[r,g,b]　红、绿、兰的强度, r、g 和 b 取 0 到 1。
Thickness[t]　显示线的宽度为 t，
Dashing[{d1,d2,…}]　用虚线段序列画线。
PointSize[d]　给出一个点的大小 d。
PlotStyle –> s1　为所有曲线规定一种样式 s1。
PlotStyle –>{{s1},{s2},…}　为一曲线序列循环地使用样式 si
请上机观看运行结果：

```
Plot[{Sin[2 x],x},{x,-1.7,1.7}, PlotStyle ->
 {Dashing[{0.01, 0.04, 0.01, 0.04}],
  Dashing[{0.03, 0.01, 0.01, 0.02}]}]
```

Plot[{x,x^2},{x,–10,10}, PlotStyle –> {{GrayLevel[0.5]},
{RGBColor [0,1,1]}}]

Plot[{x,2x},{x,1,3},PlotStyle–>{{Thickness[0.01]}, {Thickness[0.05]}}]

6.1.3 重画和组合图形

✧ **Show 与 GraphicsArray**

Mathematica 在屏幕上显示图形后用 Show 命令再现图形，用 Show 命令重新显示图形时，只允许使用 Plot 第一类可选项。

Show 的常用形式：

Show[pic**]**

Show[pic, 选项名 -> 选项值**]** 设置图形 pic 的各种选项并显示图形

Show[pic1, pic2,…, picn**]**

在 Show 中设置不同的选项产生不同的图形效果，从各种角度观察同一个图形，从中再找出选项的最佳设置值。Show 可用于 Plot3D, ParametricPlot 等几乎所有作图命令的图形再现。

GraphicsArray 组合多个图形成为一个数组，图形数组的数组元素是一幅图。常用形式有：

Show[GraphicsArray[{p1,p2,...}**]]**

依次显示每个图形 pi

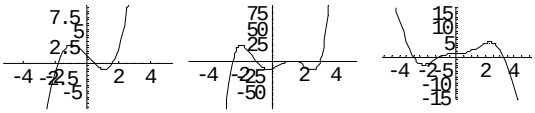
Show[GraphicsArray[{{p11,p12,...},{p21,p22,...},...}**]]**

按矩阵形式显示每个图形

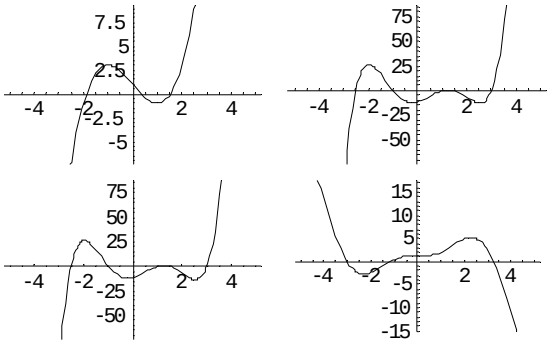
```
p1 = Plot[x^3 -3x +1, {x,-5,5}]
p2 = Plot[(x - 1)(x + 1)(x - 1.5)(x + 2.5)(x - 3),{x, -5, 5}];
p3 =Plot[x^2Sin[x] +1.2,{x, -5, 5}]
```

p1, p2 和 p3 的图略。

```
In[6]:=Show[GraphicsArray[{p1,p2,p3}]]
```



```
Out[6]=-GraphicsArray-
In[7]:=Show[GraphicsArray[{{p1,p2}, {p2,p3}}]]
```



```
Out[7]=-GraphicsArray-
In[8]:= tt =Table[Plot[Sin[x+t],{x,0,2Pi},
            DisplayFunction -> Identity], {t, 0, 8}];
In[9]:=Show[GraphicsArray[Partition[tt,3],
            DisplayFunction -> $DisplayFunction]]
```

✧ **图形表达式**

下列有关图形表达式的操作命令：

Options[pic]　　显示图形 pic 中所用的全部选项

Options[pic, opt]　　显示图形 pic 的 opt 选项值

InputForm [pic]　　显示 pic 的图形表达式

SetOptions [绘图命令, 选项名→选项值]

修改或设置作图命令选项的默认值

如果计算的对象不是显函数。例如：是一个函数表达式的表。*Mathematica* 在作图时要先计算出计算对象的值，然后再计算构造图形所需的 x 和相应的函数值 f(x)。这时计算对象前必须加以 Evaluate，以便对计算对象强行求值。

Plot [Evaluate [f], {x, xmin, xmax}**]**

6.1.4 两维参数绘图ParametricPlot

一般形式是：

```
ParametricPlot [{x[t],y[t]} {t,t0,t1}, 选项]
ParametricPlot[{{x1[t],y1[t]},{x2[t],y2[t]},…},{t,tmin,tmax}, 选项]
```

按照选项， 画一组参数曲线。

```
ParametricPlot [{Sin[t],Sin[2t]},{t,0,2Pi}]
ParametricPlot[{Cos[t],Sin[t]},{t,0,2Pi},
                AspectRatio –>Automatic]
```

6.2 三维函数作图

6.2.1 函数作图命令 Plot3D

一般形式：

Plot3D [f [x,y], {x, x0, x1}, {y, y0, y1}, 选项**]**

在区域 x∈ [x0,x1] 和 y∈[y0,y1]上，按选项画出空间曲面实数值表达式 `f[x,y]`。

Plot3D [{f [x,y],s[x,y]}, {x, x0, x1}, {y, y0, y1}, 选项**]**

按 s[x,y]设置的灰度函数(GrayLavel)或颜色函数(Hue)画函数 f [x,y].

下列 Plot3D 的常用选项：

选项名	默认值	说明
Axes	True	是否包括轴
PlotRange	Automatic	可用 All,{z0,z1}或 {{x0,x1},{y0,y1},{z0,z1}}
PlotLabel	None	在轴上加标志， PlotLabel -> z 设置 z 轴的标志 PlotLabel ->{A,B,C} 设置 x,y,z 轴的标志
AspectRatio	1:1:0.4	图形的高度与宽度之比
ViewPoint	{1.3,–2.4,2}	观察曲面所在的点，可以设定任何观察点
Boxed	True	是否在曲面周围加立体框
BoxRatios	{1,1,0.	三维立体边长比率

	4}	
Mesh	True	是否在曲面上画出 xy 网格。用 False 取消网格
HiddenSurface	True	曲面被挡住的部分是否隐掉。
Shading	True	曲面上是否涂阴影
Light	False	是否设置光源
LightSources	点光源的方向和颜色。说明形式:{光源位置,光源光色}。光源位置 用 {x,y,z} 点坐标表示,光源光色用 RGBColor 等函数表示。 缺省值可用 Option[ ]查看。	
AmbientLight	GrayLevel[0]	漫射光设置。默认值是黑色,表示没有漫射光。 可用灰度或颜色设置任意漫射光。
ClipFill	Automatic	作出图中被切掉的部分的填充方式。
PlotPoint	15	在函数在每个方向上的取样点数。

```
In[1]:= Plot3D[Sin[x y],{x,-Pi,Pi},{y,-2,2},
PlotPoints->45,Axes->False,Boxed->False];
In[2]:=Plot3D[-x y Exp[-x x - y
y],{x,-3,3},{y,-3,3},AspectRatio->Automatic]
```

6.2.2 三维参数作图

ParametricPlot3D 命令的一般形式：

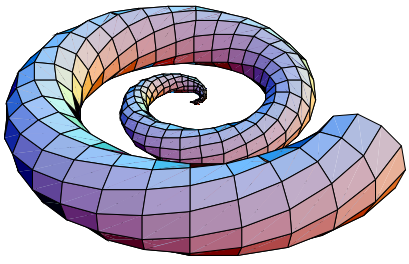
ParametricPlot3D[{x,y,z }, {u,u0,u1,(du)}, {v,v0,v1,(dv)}, 选项**]**

画出三维参数空间**曲面**，其中：x= x (u,v), y= y (u,v), z= z (u,v),

ParametricPlot3D[{x,y,z}, {t,t0,t1}, 选项**]**

画三维参数空间**曲线**，其中：x= x(t), y= y(t), z= z(t),

```
In[1]:=ParametricPlot3D[{u Cos[u](4 + Cos[v + u]),
         u Sin[u](4 + Cos[v + u]),u Sin[v + u]},
     {u,0,4 Pi},{v,0,2 Pi},PlotPoints -> {60, 12}]
In[2]:= Show[%,Boxed -> False,Axes -> False]
```



```
In[3]:=ParametricPlot3D[{Cos[u]Cos[v],Sin[u]
    Sin[v],Sin[v]},{u,0,2Pi}, {v,–Pi/2,Pi/2}]
```

6.3 等值线图和密度图

✧ **等值线图**

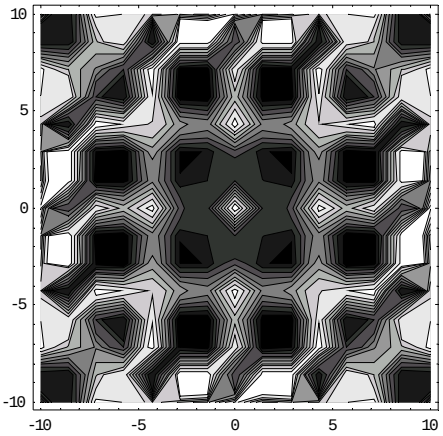
等值线很象地图上的等高线，它们把曲面上高度相等的各点连接起来，等值线序列对应于均匀间隔的 z=f (x,y)值数列。

ContourPlot 命令的一般形式：

ContourPlot [f[x,y],{x,xmin,xmax},{y,ymin,ymax},选项**]**

ContourPlot 具有默认的选项设置 Frame ->True，执行 ContourPlot 以后，*Mathematica* 送回一个 ContourGraphics 目标。如果函数值的网络不够细，等值线图可能会有误差，当函数值变化幅度较大时，ContourPlot 能给出规则的等值线图，当函数值变化太小曲面几乎是平面时，可能给出不规则的等值线图。

```
In[1]:=ContourPlot[Sin[Cos[x^2+y^2]],{x,-10,10},{y,-10,10}]
```




```
Out[1]= -ContourGraphics-
```

对同一函数加大取值点的数目等选项后，等值线的表现也有变化。

```
g[n_] := ContourPlot[Sin[Cos[ x^2 + y^2]],{x, -20, 20},{y, -20,
20},PlotPoints -> n, ContourLines -> False, ColorFunction -> Hue,
ContourSmoothing -> True]
        Table[g[k], {k, 50, 200, 20}]
```

ContourPlot 中常用选项如下。

选项名称	默认值	说 明
Contour	10	等值线的数
PlotPoints	15	每个方向上求值的点数
PlotRange	Automatic	f[x,y]即 z 值的范围，可选{z1,z2},All 等
ContourSpacing	Automatic	是否使用明暗度
ContourSmoothing	None	是否光滑等值线

✧ **密度图**

密度图与等值线图的作用相似。在密度图中，相等的数值用同一灰度表示。画密度图命令形式为：

Densityplot [f[x,y],{x,xmin,xmax},{y,ymin,ymax},选项**]**

```
DensityPlot[Sin[1/(x y)],{x,-0.8,0.8},{y,-0.8,
0.8},ColorFuntion->Hue,PlotPoints -> 25]
     Table[f[k],{k,20,120,10}]
```

```
Out[1]= -DensityGraphics-
```

6.4 数据绘图

6.4.1 二维数据绘图

有时需要绘出给定数据的图形，*Mathematica* 也有直接画出数据的图形命令，还可以使用 Fortran 或 C 等其它语言生成的数据作图。

二维数据的表示形式有：

{{x1,y1},{x2,y2},…}　　数据点{xi, yi},i=1,2,…,n.

{y1,y2,…}　　当 xi= i 时可省略 xi

二维数据绘图命令	说 明
ListPlot [{{x1,y1},{x2,y2},…}]	画出数据点{x1,y1},{x2,y2},…
ListPlot [{y1,y2,…,yn}]	画出数据点{{1,y1},{2,y2},…,{n,yn}}
ListPlot [数据,PlotJoined -> True]	画一条通过数据点的光滑曲线

表 6 - 5

例如：

```
In[1]:=d = Table[{1./n,Sin[n]},{n,1,2000}];ListPlot[d]
```

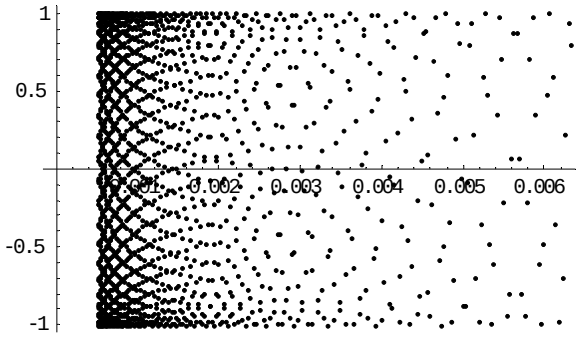


图 6 - 16

```
Out[1]= -Graphics-
```

6.4.2 三维数据绘图

{x,y,z}表示三维空间的一个数据点，{{x1,y1,z1},{x2,y2,z2},…}表示形式三维空间的一个数据点序列。下列相应的绘图函数。

三维数据绘图命令	说 明
ListPlot3D [data]	画出数据 data 的三维图
ListPlot3D[array, shades]	按灰度 shades 画 array 的三维图
ListContourPlot [data]	画出数据 data 的等值线图
ListDensityPlot [data]	画出数据 data 的密度图

例题：

```
In[1]:=tt=Table[Sin[0.01(i+j)] + Cos[0.01(i*j)],
```

```
                {i,1,50},{j,1,50}];
    ListPlot3D[tt,Axes -> False,Boxed ->False,Mesh -> False]
```

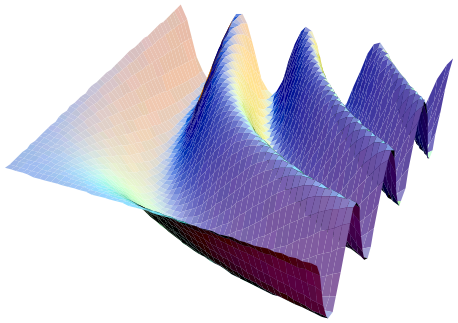


图 6 - 17

```
Out[1]=-Graphics3D-
```

6.5 用图形元素绘图

在 *Mathematica* 中也提供了二维和三维用图形元素作图函数，如点、圆弧和立方体等，使用图形元素适合于画各种结构复杂的图形。在作图中，先用 **Graphics [**图形元素**]**做出平面图形表达式，再用 **Show[**图形表达式**]**的形式演示图形表达式所表示的图形。

下面列出常用的二维图形元素：

	说 明
Point [{x, y}]	点的位置在{x,y}，x 和 y 为坐标值
Line [{{x1,y1}, {x2,y2},…}]	依次连接相邻两点的线段
Rectangle [{xmin,ymin}, {xmax,ymax}]	以{xmin,ymin}和{xmax,ymax}为对角线坐标的填实矩形
Polygon [{x1,y1}, {x2,y2},…]	以{x1,y1},{x2,y2},…为顶点的封闭多边形
Raster [{{a11,a12,…}, {a21,a22,…},…}]	灰度颜色的矩阵
Circle [{x,y}, r]	圆心在{x,y},半径为 r 的圆
Circle [{x,y}, {rx,ry}]]	圆心在{x,y}，长短半轴为 rx 和 ry 的椭圆
Circle [{x,y}, r, {t1,t2}]	从弧度 t1 到弧度 t2 的圆弧
Circle [{x,y}, {rx,rt}, {t1,t2}]	从弧度 t1 到弧度 t2 的椭圆弧
Disk [{x,y}, r]	圆心在{x,y},半径为 r 的填实圆

例题：

```
Graphics[{Line[{{-1.5,-1.5},{1.5,1.5}}],
    PointSize[0.03],Point[{0,1}],Point[{1,0}]}]
```

在 *Mathematica* 中，用 **Graphics3D [**图形元素**]**做出三维图形表达式，与二维作图的方式一样，用 **Show[**图形表达式**]**的形式显示完成的立体图形。

下面列出常用的三维图形元素和示例：

三维图形元素	几何意义
Point[{x,y,z}]	点{x,y,z}
Line[{{x1,y1,z1},{x2,y2,z2},…}]	通过点{x1,y1,z1},{x2,y2,z2},…的线
Polygon[{{x1,y1,z1},{x2,y2,z2},…}]	具有指定角的填实多边形
Cuboid[{x0,y0,z0},{x1,y1,z1}]	以{x0,y0,z0}和{x1,y1,z1}为对角线的立方体
Text [expr,{x,y,z}]	在{x,y,z}处的文本

```
data=[Table[Point[{Random[ ],Random[ ],Random[ ]}],{24}]]
Graphics3D[data]
```

6.6 调用程序包

本节基于 mathematica 4.0.

6.6.1三维常规图形（Graphics3D）

在 Graphics3D 程序包中有画柱形图、散射点图和投影图等函数。下面按类别给出部分函数定义和例题。

✧ **画柱形图函数**

BarChart3D[{{z11,z12,…},{z21,z22,…},…}**]** 高度为 zij 的柱形图

BarChart3D[{{{z11,style11},{z12, style12},…},…}**]** 按样式 style 画柱形图

✧ **画散射点图**

ScatterPlot3D[{{x1,y1,z1},{x2,y2,z2},…}**]** 画三维散射点**{x,y,z}**图

ScatterPlot3D[{{x1,y1,z1},{x2,y2,z2},…},PlotJoined->True**]**

将散射点**{x,y,z}**连成线

ListSurfacePlot3D[{{{x11,y11,z11},{x12,y12,z12},…},…}**]**

画出三维点数组的曲面图

```
In[3]:=pts = Table[{t Cos[t],t Sin[t],t},{t,0,4Pi,Pi/20}];
 ScatterPlot3D[pts,PlotStyle->{PointSize[0.012]},Boxed->False];
```

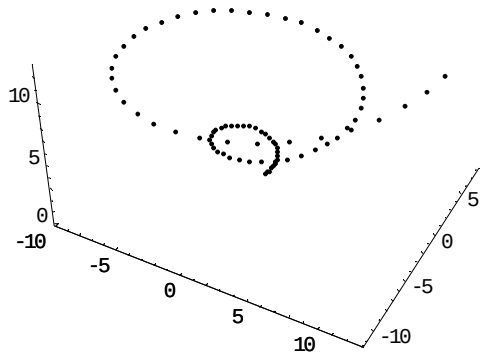


图 6 - 42

✧ **画投影函数**

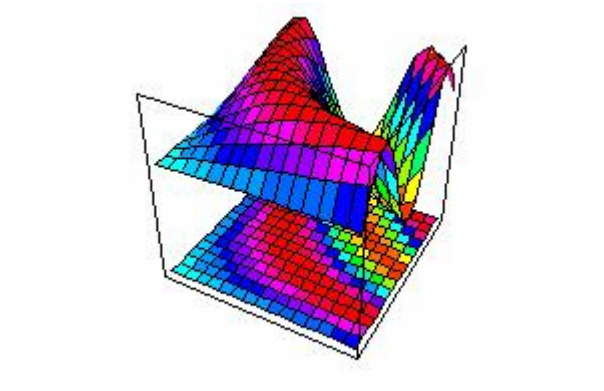
Shadow[g**]** 画出三维图形 g 和在各座标面的投影

ShadowPlot3D[f, {x, xmin, xmax}, {y, ymin, ymax}**]**

画出三维图形 f 在区域{xmin,xmax}，{y,ymin,ymax}内的投影图

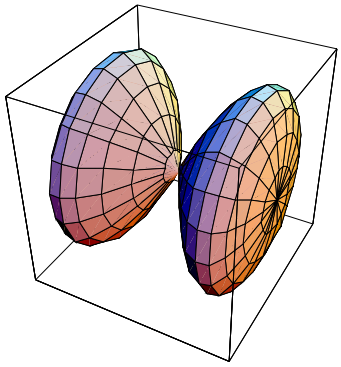
ListShadowPlot3D[array, (opts)**]** 按 opts 选项值用数组数据画投影图，

```
In[4]:=ShadowPlot3D[Sin[x y], {x, 0, 3}, {y, 0, 3}]
```



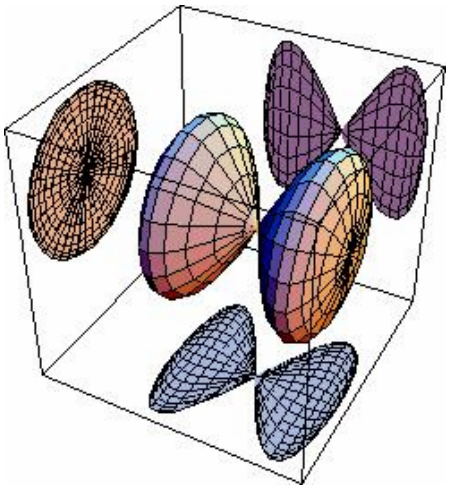
Out[4]=-Graphics3D-

In[5]:=**dbell=ParametricPlot3D[{Sin[t],Sin[2t]Sin[u],Sin[2t]Cos[u]},**
{t,-Pi/2,Pi/2},{u,0,2Pi},Ticks -> None]



Out[5]=-Graphics3D-

In[6]:=**Shadow[dbell]**



Out[6]=-Graphics3D-

In[7]:=**Shadow[dbell, ZShadow -> False]** (* 图略 *)

Project 做图形的投影, 与 Shadow 不同的是 Project 只演示投影图而不演示源图.

Project[g,pt]

将三维图形 g 投影至一平面，此平面的法线是 g 的中心和 pt 的连线。

Project[g,{e1,e2},pt] 将 g 投影 至 based at pt 由向量 e1, e2 张成的平面上。

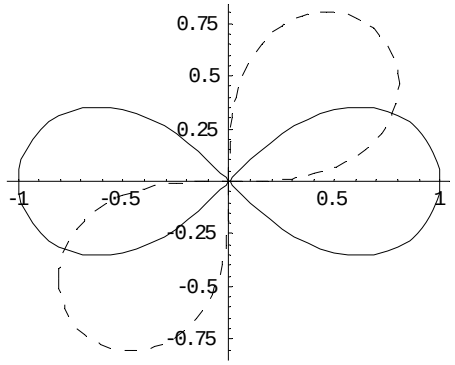
In[7]:=**Show[Project[dbell,{1,0,0}]]**

6.6.2 隐函数图形（ImplicitPlot）

隐函数绘图函数	说 明
ImplicitPlot[eqn,{x,xmin,xmax}]	用 Solve 解方程 eqn 画出隐函数图
ImplicitPlot[eqn,{x,xmin,m1,m2,…,xmax}]	画避开点 mi 的方程 eqn 的图形
ImplicitPlot[eqn,{x,xmin,xmax},{y,ymin,ymax}]	用 ContourPlot 方法画隐函数图
ImplicitPlot[{eqn1,eqn2,…},ranges,options]	画出 egni 的隐函数图

表 6-14

```
In[1]:=<< Graphics`ImplicitPlot`
In[2]:=ImplicitPlot[{(x^2+y^2)^2 ==(x^2- y^2),(x^2 +y^2)^2==2x y},
     {x,-2,2}, PlotStyle->{GrayLevel[0],Dashing[{.03}]}]
```




```
Out[3]=-Graphics-
```

6.6.3多维数据绘图（MultipleListPlot）

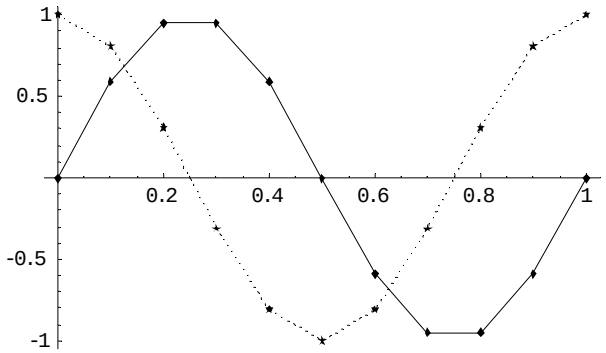
MultipleListPlot[list1,list2,...]

在同一个图上画出数据序列 list1,list2,...

MultipleListPlot[list1,list2,..., PlotJoined -> True]

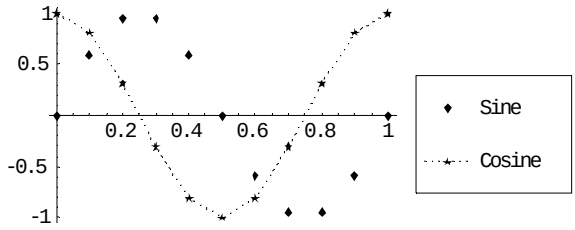
将数据序列 list1,list2,...连线

```
In[1]:=<<Graphics`MultipleListPlot`
In[2]:= (list1 = Table[{x,Sin[2 Pi x]},{x, 0, 1, 0.1}];
      list2 = Table[{x,Cos[2 Pi x]},{x, 0, 1, 0.1}]);
In[3]:=MultipleListPlot[list1, list2, PlotJoined -> True];
```




```
In[4]:=MultipleListPlot[list1,list2,PlotLegend->          {"Sine",
```

```
"Cosine"},
                    PlotJoined -> {False,True}]
```




```
Out[4]=-Graphics-
```

6.6.4空间曲线和曲面(ParametricPlot3D)

ParametricPlot3D 是画三维空间曲线和曲面系统的内嵌函数。ParametricPlot3D 程序包扩充了空间曲线和曲面的功能。其中，PointParametricPlot3D 画一维或二维空间的点。还有画球面和柱面的函数。

PointParametricPlot3D[$\{f_x, f_y, f_z\},\{u,u_0,u_1,du\}$**]**

画出具有一个参数的空间点序列

PointParametricPlot3D[$\{f_x, f_y, f_z\},\{u,u_0,u_1,du\},\{v,v_0,v_1,dv\}$**]**

画出具有两个参数的空间点序列

SphericalPlot3D[r,{theta,to,t1},{phi,p0,p1}**]** 画出球体在区域内的图

CylindricalPlot3D[z,{r,r0,r1},{theta,to,t1}**]** 画出柱体在区域内的图

```
In[1]:= <<Graphics`ParametricPlot3D`
In[2]:=PointParametricPlot3D[ {Cos[u] Cos[v], Sin[u] Cos[v], Sin[v]},
          {u,0,2Pi },{v,-Pi/2,Pi/2 },Boxed->False]
```

6.6.5向量场函数 （PlotField和PlotField3D）

✧ **二维向量场程序包 PlotField**

PlotVectorField [{fx,fy},{x,xmin, xmax},{y,ymin,ymax}**]**

画出给定数值函数{fx,fy}在所在区间的x-y平面上的向量场

PlotGradientField [f,{x,xmin, xmax},{y,ymin,ymax}**]**

画出数值函数 f 在给定区间的 Gradient 向量场（梯度），用$\frac{\partial f}{\partial x}$ 和 $\frac{\partial f}{\partial y}$分别定义梯度场的两个分量。

PlotHamiltonianField [f,{x,xmin, xmax},{y,ymin,ymax}**]**

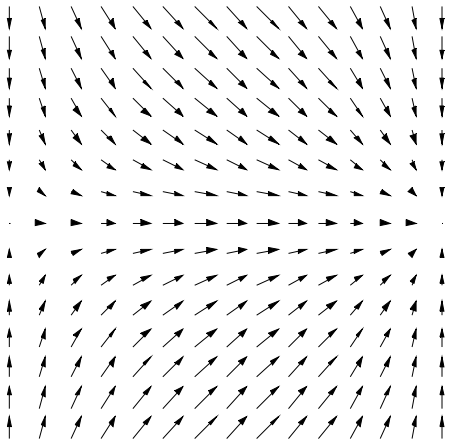
画出数值函数 f 在给定区间的 Hamiltonian 向量场，用$\frac{\partial f}{\partial x}$ 和$-\frac{\partial f}{\partial x}$分别定义梯度场的两个分量。

PlotPolyaField [f[x+I y],{x,x0, x1,(dx)},{y,y0,y1,(dy)},options**]**

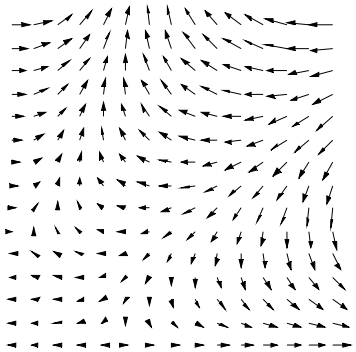
画出复函数 f 在给定区间的复平面的向量场。

ListPlotVectorField[data**]** 绘制平面上点序列表示的向量场。

```
In[1]:= <<Graphics`PlotField`
In[2]:=PlotVectorField[{Sin[x],Cos[y]},{x,0,Pi},{y,0,Pi}]
```



```
Out[2]=-Graphics-
In[5]:=PlotPolyaField[(x + I y)^4 - 1, {x, 0, 3}, {y, 0, 3}]
```



```
Out[5]=-Graphics-
```

✧ **三维向量场程序包 PlotField3D**

PlotVectorField3D [{fx,fy,fz},{x,xmin, xmax},{y,ymin,ymax},{z,zmin,zmax}]

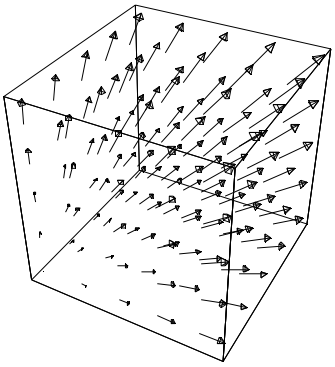
画出向量函数{fx,fy,fz}在给定区间的向量场

PlotGradientField3D [f,{x,xmin, xmax},{y,ymin,ymax},{z,zmin,zmax}]

画出数值函数 f 在给定区间的 Gradient 向量场, 用 $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$,和 $\frac{\partial f}{\partial z}$ 分别定义梯度场的三个分量。

ListPlotVectorField3D[data] 画出三维向量点列 data 的向量场。

```
In[1]:=<<Graphics`PlotField3D`
In[2]:=PlotVectorField3D[{x , y, z}, {x, 0, 2}, {y, 0, 2}, {z, 0, 2},
            PlotPoints -> 5,VectorHeads -> True]
```



```
In[4]:=ListPlotVectorField3D[{{{0,.1,.2},{-.1,.3,.5}}}]
In[5]:=data = Flatten[Table[{{i, j, k},
     {Random[Real,{-1,1}],Random[Real,{-1, 1}],
      Random[Real,{-1,1}]}},{i,3},{j,3},{k,3}],2];
     ListPlotVectorField3D[data]
   （* In[4]和 In[5]的图略*）
```

6.6.7 多边形（Polyhedra）

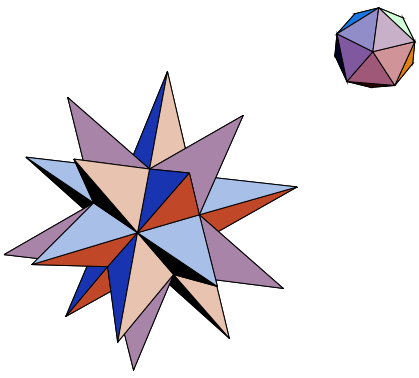
Show[Polyhedron[polyname**]]**　演示多边形 polyname

Show[Polyhedron[polyname,{x,y,z},scale**]]**

以{x,y,z}为中心，按 scale 的大小演示多边形 polyname。

Polyhedron 中定义的多面体			
Tetrahedron	四面体	Cube	立方体
Octahedron	八面体	Dodecahedron	十二面体
Icosahedron	二十面体	Hexahedron	六面体

```
In[1]:=<<Graphics`Polyhedra`
In[2]:=Show[ Polyhedron[GreatStellatedDodecahedron],
      Polyhedron[Icosahedron, {3, 3, 3}, 0.7] ]
```



```
Out[3]=-Graphics3D-
```

程序包中还有一些演示多面体的方式。例如按星形演示多面体的 Stellate 函数等。

Show[Stellate[Polyhedron[polyname]]]

Show[Geodesate[Polyhedron[polyname]]]
Show[Truncate[Polyhedron[polyname]]]
Show[OpenTruncate[Polyhedron[polyname]]]
In[4]:= **Show[Stellate[Polyhedron[Octahedron],4.0]]**

6.6.8 几何图元（Shapes）

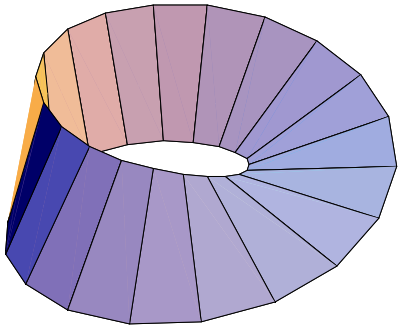
Mathematica 的绘图中最强大的一个方面源于一些初等图形，如多边形和六面体等图形元素。通过这些基本图形，就可以绘制出许多三维物体了。这个软件包给出了一些在绘制一般三维图形时有用的多边形。

Show[Graphics3D[shape]]　　显示几何图元 shape

每个几何图元还可附加参数，例如半径和高度。如果不想给几何图元设置参数，计算中取缺省值，例如， Torus[ ]与 Torus[1,0.5,20,10]效果相同。

程序包中的图元和缺省定义	说 明
Cylinder[r,h,n] Cylinder[1,1,20]	由 n 个多边形组成的半径 r 高度是 2h 的柱体
Cone[r,h,n] Cone[1,1,20]	由 n 个多边形组成半径 r 高度是 2h 的锥体 half height h
Torus[r1,r2,n,m] Torus[1,0.5,20,10]	由 n.m 个多边形组成半径是 r1 和 r2 的圆柱面
Sphere[r,n,m] Sphere[1,20,15]	由 n(m-2)+2 个多边形组成半径 r 的球面
MoebiusStrip[r1,r2,n] MoebiusStrip[1,0.5,20]	由 2n 个多边形组成半径是 r1 和 r2 的
Helix[r,h,m,n] Helix[1,0.5,2,20]	由 n.m 个多边形组成半径是 r，高度是 2h 的螺旋面
DoubleHelix[r,h,m,n] DoubleHelix [1,0.5,2,20]	双螺旋面

In[1]:= <<Graphics`Shapes`
In[2]:=**Show[Graphics3D[MoebiusStrip[]], Boxed -> False]**

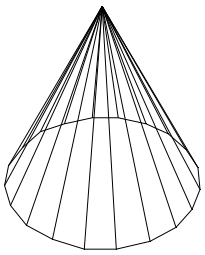


Out[2]=-Graphics3D-

WireFrame[g]　　画出多面体的母线

下面画出锥面的框架线。

```
In[5]:=Show[WireFrame[Graphics3D[Cone[ ]]],Boxed->False]
```



```
Out[5]=-Graphics3D-
```

6.6.9旋转曲面函数（SurfaceOfRevolution ）

函数定义：

SurfaceOfRevolution [f,{x,xmin,xmax},{theta, thetamin,thetamax}]

画出从角度 thetamin 到 thetamax 内绕 z 轴 f 的旋转曲面

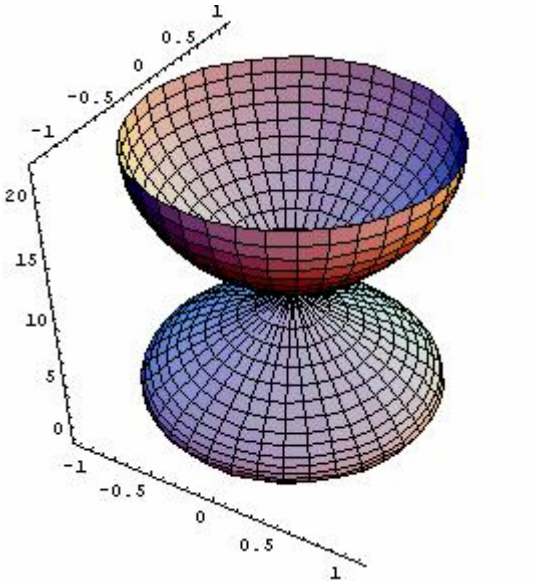
ListSurfaceOfRevolution [{point1,point2,…}]

画出点列曲线的旋转曲面

ListSurfaceOfRevolution [{p1,p2,…},{t,t1,t2}]

画出点列曲线从角度 thetamin 到 thetamax 旋转曲面

```
In[1]:= << Graphics`SurfaceOfRevolution`
In[2]:= SurfaceOfRevolution[{1.1 Sin[u], u^2}, {u, 0, 3 Pi/2},
        BoxRatios ->{1, 1, 1},Boxed -> False, PlotPoints -> 40]
```




```
Out[2]=-Graphics3D-
```