第5章 图形绘制

5.1 绘制二维图

5.1.1 绘制二维线性图

1．plot()函数与线性线条

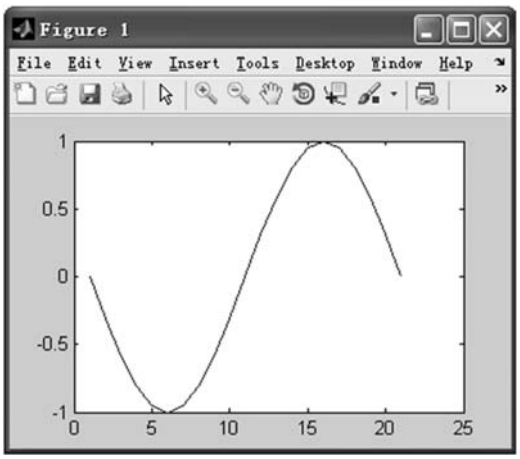
plot()是一个最常用的绘图函数，使用plot()可绘制一个连续的线性图。语法格式如下：

(1) plot(Y)：该命令中的**Y可以是向量、实数矩阵或复数向量**。

如果Y是实数向量，则以向量的索引为横坐标，以向量元素值为纵坐标绘制图形，以直线段顺序连接各点。

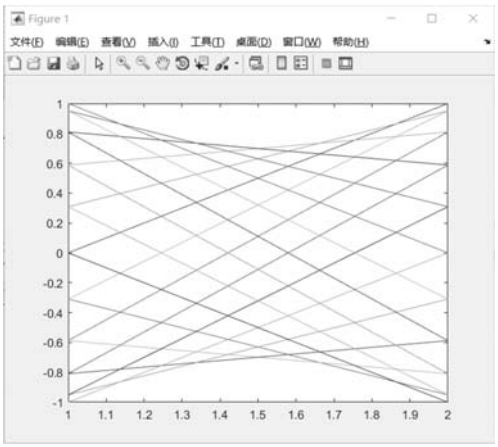
输入下列命令：

```
x = -pi:pi/10:pi;
y = sin(x);
plot(y)
```



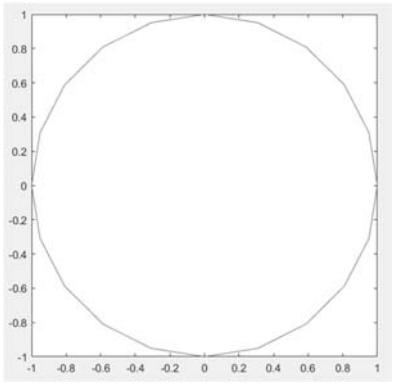
如果Y是矩阵，则绘制**Y**的各列。

```
x=-pi:pi/10:pi;
y=[sin(x);cos(x)];
plot(y)
```



如果Y是复向量，则以复数的**实部为横坐标**，虚部为**纵坐标绘制图形**，即plot(Y)相当于plot(real(Y),imag(Y))，而在其他的绘图格式中复数的虚部会被忽略。

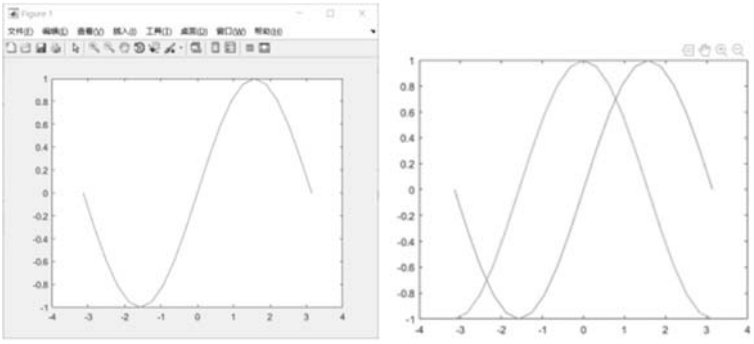
```
x=-pi: pi/10 :pi;
y=sin(x)+i*cos(x);
plot(y)
axis equal
```



(2) plot(X,Y)、plot(X1,Y1,...,Xn,Yn)：该命令中的X和Y可以为向量和矩阵，当X和Y的结构不同时，则有不同的绘制方式：

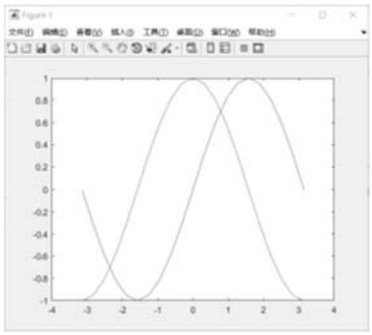
- **X和Y均为n维向量时，以X的元素为横坐标，Y的元素为纵坐标绘制图形**，绘出每个向量Yn对向量Xn的值。

```
x = -pi:pi/10:pi;
y = [sin(x);cos(x)];
plot(x,y(1,:))
pause
plot(x,y(1,:) , x,y(2,:))
```



- 如果**Yn或X之中一个是矩阵，而另一个是向量，则按向量的维数绘制向量对矩阵的行或列的图形**。X为n维向量，Y为m×n或n×m矩阵时，以X的元素为横坐标，绘制Y的m个n维向量。

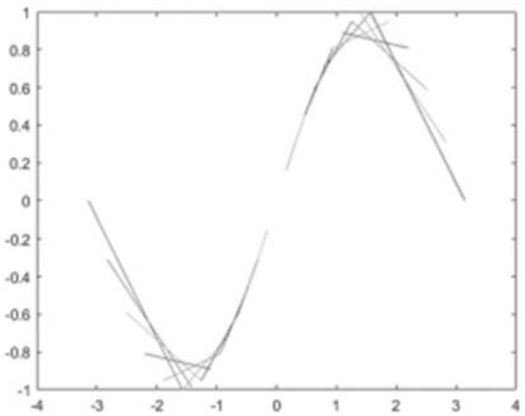
```
x=-pi:pi/10:pi;
y=[sin(x);cos(x)];
plot(x,y)
```



问题： 如果把画图的x和y调换下，plot(y,x) 会怎样？

- X、Y均为m×n矩阵时，以X的各**列**为横坐标，Y的对应**列**为纵坐标绘制图形。

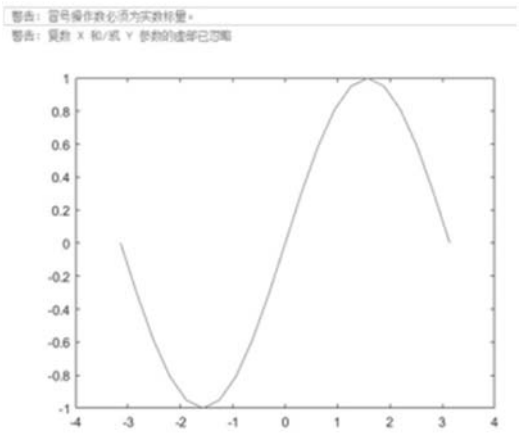
```
x=[-pi:pi/10:pi;-pi/2:pi/20:pi/2];
plot(x,sin(x))
```



思考：如果要根据x矩阵中的两行数据作为横坐标画出2条sin(x)曲线，应该怎么写代码：

- 如果Xn或Yn是复数，则虚部被忽略。

```
x=-pi:pi/10:pi + i*[-pi/2:pi/20:pi/2];
y=sin(x)+i*cos(x);
plot(x,y)
```



(3) plot(X1,Y1,LineSpec,...,Xn,Yn,LineSpec)：LineSpec用于控制图像外观，指定线条的类型(如实线、虚线、点划线等)、标志符号、颜色等属性。

表 5-6　线型和颜色控制符

线　型		点 标 记		颜　色	
-	实线	.	点	y	黄
:	虚线	o	小圆圈	m	棕色
-.	点划线	x	叉子符	c	青色
- -	间断线	+	加号	r	红色
		*	星号	g	绿色
		'square' 或 s	方形	b	蓝色
		'diamond' 或 d	菱形	w	白色
		∧	朝上三角	k	黑色
		∨	朝下三角		
		>	朝右三角		
		<	朝左三角		
		'pentagram'或 p	五角星		
		'hexagram' 或 h	六角星		

plot(X1,Y1,**LineSpec,'PropertyName',PropertyValue**)：使用属性名称和属性值指定线条的特性。还可以设置其中的4种附加的属性

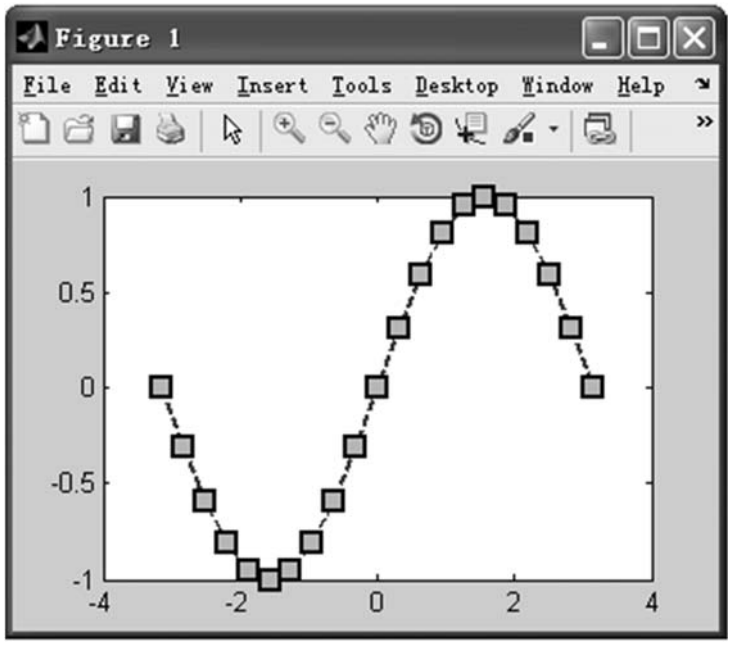
表 5-1 线型的 4 种附加属性

属 性	说 明
LineWidth	用来指定线的宽度
MarkerEdgeColor	用来指定标识表面的颜色
MarkerFaceColor	填充标识的颜色
MarkerSize	指定标识的大小

如果输入下列命令：

```
x = -pi:pi/10:pi;
y = sin(x);
plot(x,y,'--rs','LineWidth',2,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','g',...
    'MarkerSize',10)
```

则绘制出指定属性的线条图形。



第5章 图形绘制

5.1.2 stem()绘制离散图形

stem()函数绘制离散数据的图形。语法如下：

(1) stem(Y)：沿x轴按等间隔绘制序列Y的图形，当Y是矩阵时，使用所有元素的数据绘制。

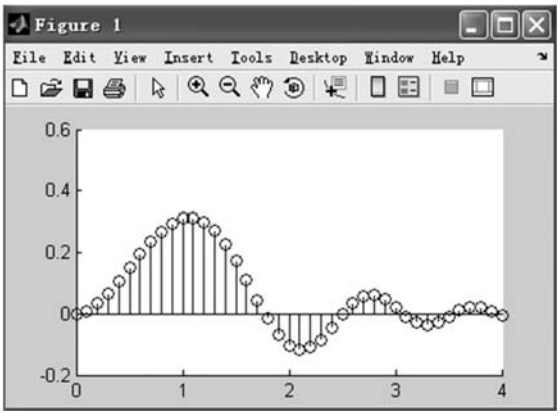
(2) stem(X,Y)：绘制X对Y的图形，X、Y必须是向量或同样大小的矩阵。

(3) stem(...,'fill')：将离散图形末端的小圆圈用当前的颜色填充。

(4) stem(...,LineSpec)：按LineSpec指定的线条属性绘制。

例如，输入下列命令：

```
x = 0:0.1:4;
y = sin(x.^2).*exp(-x);
stem(x,y)
```



5.1.3 对数图

loglog()、semilogx()、semilogy()等函数用于绘制对数坐标图形，其用法与plot相似，这些命令允许数据在不同的图形页面上绘制，例如不同的坐标系统。

1．对数坐标绘图

用log10-log10标度绘图，即x、y轴坐标都是常用对数。语法如下：

- loglog(Y)
- loglog(X1,Y1,...)
- loglog(X1,Y1,LineSpec,...)
- loglog(...,'PropertyName',PropertyValue,...)
- h = loglog(...)

例如，输入下列代码：

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

绘制出的对数坐标图形如图5-7所示。

第5章　图形绘制

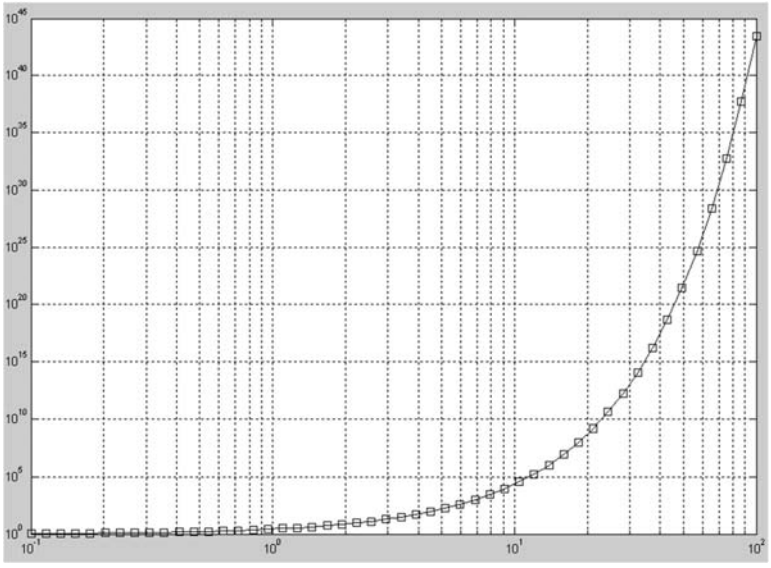


图5-7 对数坐标图形

2．半对数坐标绘图

- semilogx：用半对数坐标绘图，x轴是log10，y是线性的。
- semilogy：用半对数坐标绘图，y轴是log10，x是线性的。

例5-1-1 用半对数坐标绘图。

```
t=0.001:0.002:20;
y=5 + log(t) + t;
semilogx(t,y, 'b')
hold on
semilogx(t,t+5, 'r')
```

绘制出的半对数坐标图形如图5-8所示。

第5章　图形绘制

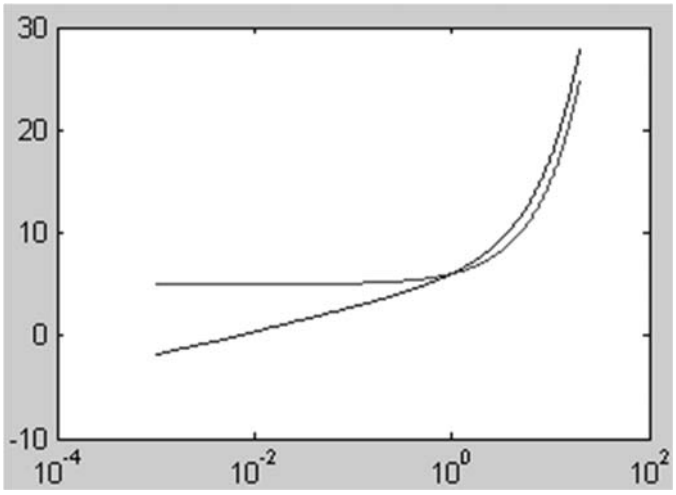


图5-8 半对数坐标图形

5.1.4 polar()绘制极坐标图

MATLAB中的一个重要的函数称做polar()，它用于在极坐标系中画图。这个函数的基本形式如下：

polar(theta, rho)

使用相角theta为极坐标形式绘图；rho代表一个距离数组，为相应的极半径，可使用grid命令画出极坐标网格。用polar()来画以角度为自变量的函数的极坐标图是非常有用的。与其他绘图函数一样，可以设置线型属性，可以返回函数句柄。

例如，输入下列命令：

```
t=0:0.01:2*pi;
polar(t,abs(sin(2*t).*cos(2*t)));
grid on
```

绘制出的极坐标图形如图5-9所示。

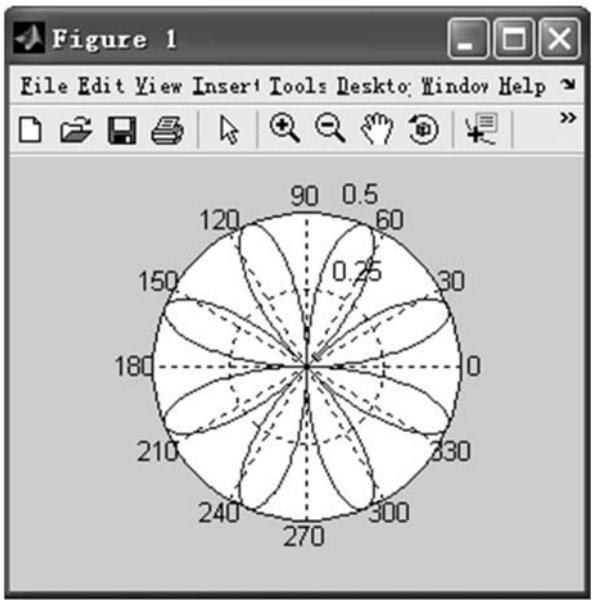


图5-9 极坐标图形

5.2 常用图形的绘制

5.2.1 绘制直线、矩形、圆

1．绘制直线

line()函数用于绘制直线。语法如下：

(1) line：在当前轴中绘制直线，默认值x = [0 1]、y = [0 1]。

(2) line(X,Y)：在当前轴中按照向量X、Y绘制直线，如果X、Y是大小相同的矩阵，则为每一列绘制一条直线。

例如，line([0 1],[3 3])：表示x1 = 0、x2 = 1，y1 = 3、y2 = 3，即从y = 3处绘制0～1的水平线。

```
line([0 1 2;3 4 5],[4 5 6;1 2 3])
```


(3) line(X,Y,Z)：按三维坐标绘制直线。

(4) line(X,Y,Z,'PropertyName',propertyvalue,...)：按三维坐标，根据指定的线型等属性绘制直线。

(5) line('XData',x,'YData',y,'ZData',z,...)：低层绘制直线函数。

(6) h = line(...)：返回图形句柄h。

例如：

```
>> line([.3 .7],[.4 .9],[1 3],'Marker','.','LineStyle','-')
```

在三维坐标中，根据指定的线型等属性绘制直线。

Marker的选项如下：

- '+'：加号；'o'：小圆圈；'*'：星号；'.'：点；'x'：交叉符号。
- 'square' or 's'：方框；'diamond' or 'd'：钻石。
- '^'、'v'：上下三角；'>'、'<'：左右三角。
- 'pentagram' or 'p'：五星；'hexagram' or 'h'：六星。
- 'none'：无标志，默认。

LineStyle的选项如下：

- '–'：实线，默认；'--'：断画线。
- ':'：点线；'–.'：点画线。
- 'none'：无。

2．绘制矩形

使用**rectangle()**函数可以生成2D的rectangle对象，即绘制方形图形。语法如下：

(1) rectangle()：以位置属性Position [0,0,1,1] 和曲线属性Curvature [0,0](代表无曲线)绘制矩形图形。

(2) rectangle**('Position',[x,y,w,h])**：指定位置属性Position，从点(x,y)开始，以width = w、height = h值，按轴的数据单位绘制矩形图形。

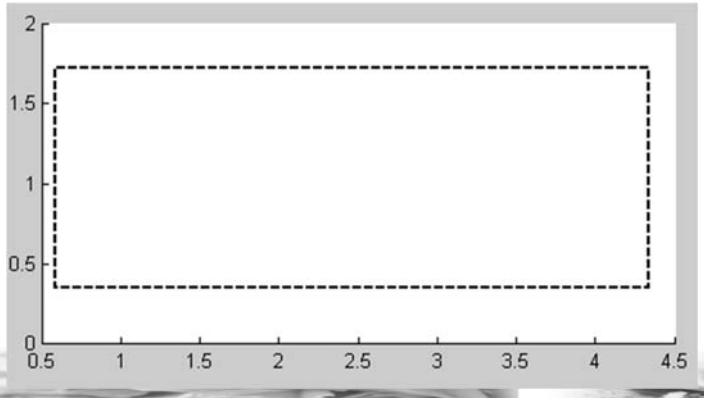
(3) rectangle('Curvature',[x,y])：指定曲线属性curvature，可以使矩形图形变为椭圆图形。**水平的 curvature属性 x 是矩形宽度对曲线从顶部到底部高度的比例，垂直的curvature属性y矩形高度对曲线从左边到右边宽度的比例。**

x和y值可以是0(无曲线)～1(最大曲线)，值[0,0] 生成方形，值[1,1]生成椭圆形。如果只指定Curvature的一个值，在水平和垂直方向以同样长度，按轴的数据单位绘制曲线。

例 根据curvature属性，可改变矩形的不同形状。

解 (1) curvature属性值为[0,0]可生成矩形。程序代码如下：

```
rectangle('Position',[0.59,0.35,3.75,1.37],...
    'Curvature',[0,0],...
    'LineWidth',2,'LineStyle','--')
```



(3) 如果只指定Curvature的单个值，例如“'Curvature',[0.4]”，在水平和垂直方向以同样长度，按轴的数据单位绘制出圆角矩形曲线，如图5-12所示。

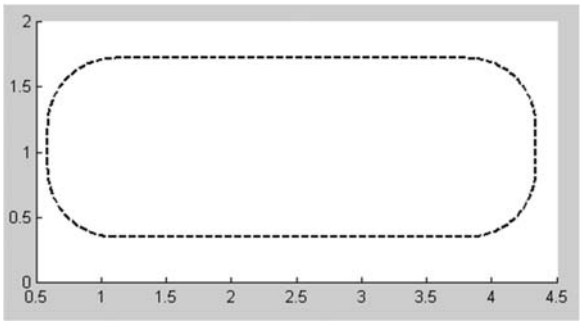


图5-12 绘制圆角矩形曲线

5.2.2 绘制偏差条图形

可使用errorbar()函数沿曲线绘制数据的偏差条，其语法格式如下：

(1) errorbar(Y,E)：绘制Y和Y的每一个元素的偏差E，偏差条E(i)的中心位于曲线上，与曲线上下对称的偏差距离是E(i)，因此偏差条的长度是动态的，并且长度为2 × E(i)。

(2) errorbar(X,Y,E)：以2 × E(i)的对称长度绘制Y 对X 的偏差条，X、Y、E必须是同样大小。当它们是向量时，偏差距离E(i)由(X(i),Y(i))定义；当它们是矩阵时，偏差距离E(i,j)由(X(i,j),Y(i,j)).定义。

(3) errorbar(X,Y,L,U)：绘制X对Y带不对称偏差条的曲线，偏差条长度为L(i)+U(i)，L(i)、U(i)分别指定曲线下部和上部部分偏差条的长度。

(4) errorbar(...,LineSpec)：按LineSpec指定的线条属性绘制。

(5) h = errorbar(...)：返回绘图句柄h。

例如，输入下列命令：

```
x=-2:0.1:2;
y=erf(x);
e = rand(size(x))/10;
errorbar(x,y,e);
```

绘制出偏差条状图形，如图5-18所示。

第5章 图形绘制

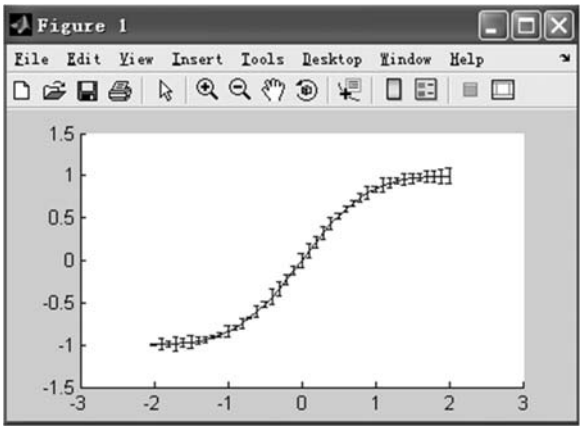


图5-18 偏差条图

5.2.3 绘制直方图与其正态分布曲线

MATLAB中有两个绘制直方图的函数：hist()和rose()，它们分别用于在直角坐标系和极坐标系中绘制直方图。

1．hist()函数

hist()函数以柱状的分布方式显示数据值，可以绘制统计频率直方图。语法如下：

(1) n = hist(Y)：把向量Y的元素分到**10份等间隔**的柱状分格中，绘制Y的直方图。每个柱状分格根据元素号按行向量形式排列。

第5章 图形绘制

(2) n = hist(Y,x)：指定直方图的每个分格，其中x是一个向量，Y按length(x)的数值以柱状的分布方式绘制，中心由x确定。例如，如果x 是一个5元素的向量，hist 函数把Y分布为5个柱，显示为正态分布直方图。

(3) n = hist(Y,nbins)：nbins是一个标量，用于指定分格的数目。Y按该标量分布，显示为平均分布直方图。

例5-2-3 下列代码显示正态分布直方图和平均分布直方图。

解 程序如下：

```
yn=randn(30000,1);      %%正态分布
x=min(yn) : 0.2 : max(yn);
hist(yn, x)
title('正态分布图' )
yu=rand(30000,1);       %%平均分布
figure;
hist(yu, 25)
title('平均分布图' )
```

第5章 图形绘制

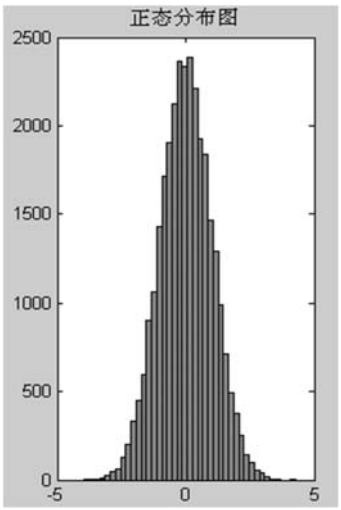


图 正态分布直方图

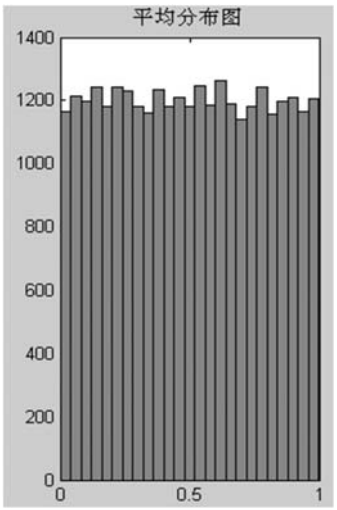


图 平均分布直方图

3．rose()函数

rose()函数用于在极坐标系中绘制直方图，它与hist()函数的调用格式类似。

例如，运行下列程序：

```
theta = 2*pi*rand(1,50);
rose(theta)
```

可以得到极坐标系中绘制的直方图，如图5-22所示。

第5章　图形绘制

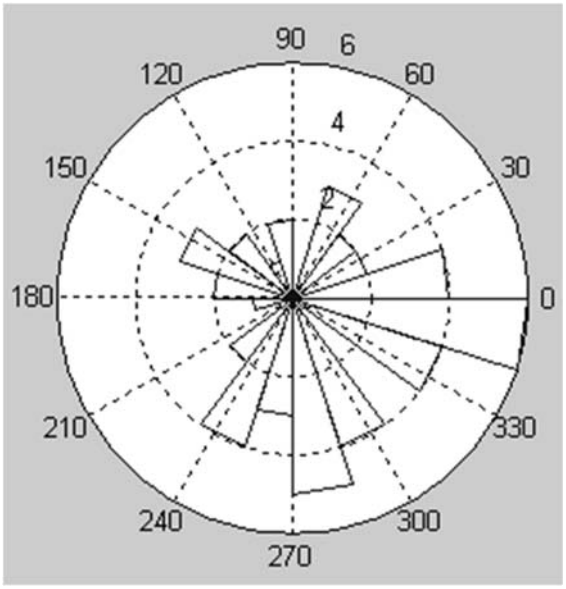


图5-22 极坐标直方图

5.2.4 填充图与面积图

1. fill()函数

使用函数fill()来绘制类似的填充图，产生一个或多个2D多边形的填充区域。

例5-2-5 用函数fill()绘制填充图。

解 程序如下：

```
x=0:pi/60:2*pi;
y=sin(x);
x1=0:pi/60:1;
y1=sin(x1);
plot(x,y,'r');
hold on
fill([x1 1],[y1 0],'g')
```

该程序的绘制结果如图5-24所示。

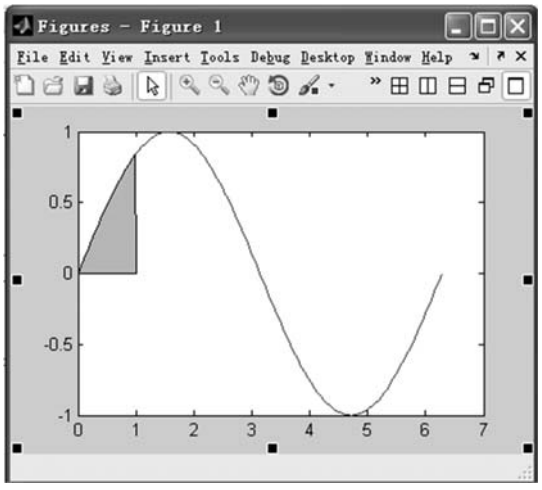


图5-24 用函数fill()绘制填充图

2．area()函数

函数area()用于填充图绘制向量构成的曲线，或者当输入参数为矩阵时，绘制矩阵的每一列为一条曲线，并填充曲线间的区域。填充图可以直观显示向量的每个元素或矩阵的每一列对总和的贡献大小。面积填充图由函数area()绘制。该函数的调用格式为

(1) area(Y)：绘制向量Y或矩阵Y各列的和。

(2) area(X,Y)：若X和Y是向量，则以X中的元素为横坐标，Y中元素为纵坐标绘制图像，并且填充线条和x轴之间的空间；如果Y是矩阵，则绘制Y每一列的和。

(3) area(...,basevalue)：设置填充的底值，默认为0。

例5-2-6 已知Y = [1, 5, 3;3, 2, 7; 1, 5, 3;2, 6, 1]。

5.3 三维图形绘制

三维图形包括三维曲线图和三维曲面图。MATLAB语言提供了三维图形的处理功能，与二维图形相似，绘制三维图形时可以使用MATLAB语言提供的相关函数：

- 三维线图指令：plot3。
- 三维网线图mesh和三维曲面图surf。

5.3.1 plot3()函数

在MATLAB中，plot3()函数用于绘制三维曲线。该函数调用的基本格式与plot()函数的类似。 plot(x,y,z)

```
t=0:pi/10:10*pi;
x=sin(t);
y=cos(t)
z=t;
plot3(x,y,z)
```

例5-3-1 绘制三维的螺旋曲线图。

解 程序如下：

```
%该程序用于绘制三维的螺旋曲线图

t = 0:pi/50:20*pi;
plot3(sin(t),cos(2*t),sin(t)+cos(t))
```

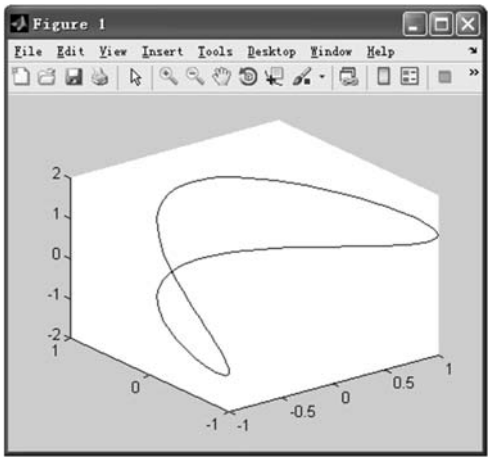


图5-26 plot3绘制三维曲线

5.3.2 mesh()和surf()函数

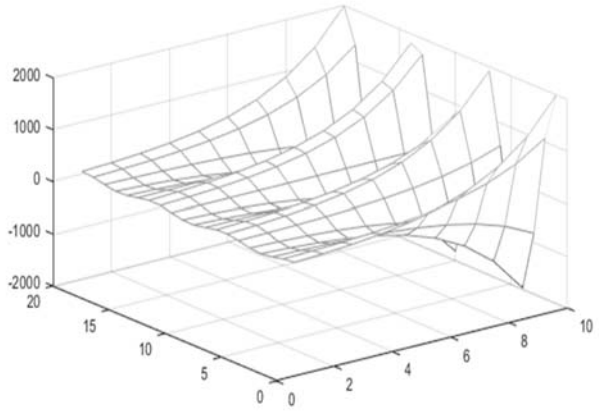
mesh()函数可以绘制出在某一区间内完整的网格曲面，mesh(Z)语句可以给出矩阵Z元素的三维消隐图，网络表面由Z坐标点定义，与前面叙述的x-y平面的线格相同，图形由邻近的点连接而成。它可用来显示用其他方式难以输出的包含大量数据的大型矩阵，也可用来绘制Z变量函数。

surf()函数可以绘制三维曲面图。这两个函数的调用方法基本相同，其格式如下：

(1) mesh(X,Y,Z)，surf (X,Y,Z)：绘制出一个网格图(曲面图)，图像的颜色由 Z 确定，即图像的颜色与高度成正比。如果函数参数中，X和Y是向量，length(X) = n，length(Y) = m，size(Z) = [m,n]，则绘制的图形中，X(j), Y(i), Z(i,j)为图像中的各个节点。

(2) mesh(Z)，surf (Z)：使用X = 1:n和Y = 1:m，[m,n] = size(Z)，高度为Z，它是一个单值函数，图像的颜色与高度Z成正比，即以Z的元素为z坐标，元素对应的矩阵行和列分别为x坐标和y坐标，绘制图像。

```
x=1:0.1:1.9;
y=0:0.1:1.9;
z=cos(10*y')*exp(4*x);
mesh(z)
```



1．mesh()函数的应用

例5-3-2 绘制sin(R)/R函数的三维网格图。

解 程序如下：

```
x=-8:0.5:8;
y=x';
X=ones(size(y))*x;
Y=y*ones(size(y))';
R=sqrt(X.^2+Y.^2)+eps;
Z=sin(R)./R;
mesh(Z)
```

其中，各语句的意义如下：

(1) 首先建立行向量x，列向量y：

第1条语句x的赋值为定义域，在其上估计函数、建立行向量x；

第2条语句建立列向量y。

(2) 生成X矩阵：

ones(size(y))语句按向量y的长度建立1_矩阵(即33个1元素的列向量)；

第3条语句，该1_矩阵与行向量x相乘，建立一个33 × 33重复行的X矩阵，每行都是向量x。

(3) 生成Y矩阵：

建立一个33 × 33重复列的Y矩阵，每列均为向量y：

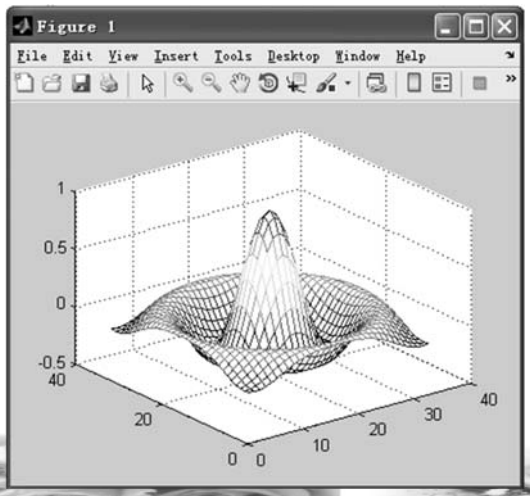
ones(size(y))生成一个33个1元素的1_列向量，ones(size(y))'生成一个33元素的1_行向量(1_矩阵)。

第4条语句产生Y的响应：用列向量y乘以产生的1_矩阵(1_行向量)，建立一个33 × 33重复列的Y矩阵，每列均为向量y。

(4) 生成三维网格曲面图：

- 计算各网格点的半径：第5条语句产生矩阵R(其元素为各网格点到原点的距离)，它们的值对应于x-y坐标平面。
- 生成网格矩阵：最后计算函数值矩阵Z。
- 用mesh(Z)函数即可以得到图形。

该程序运行后得到三维网格曲面图。



2．surf()函数

surf()函数也是MATLAB中常用的三维绘图函数，其一般调用格式如下：

surf(x,y,z)

该函数输入参数的设置与mesh()相同，不同的是mesh()函数绘制的是一网格图，而surf()函数绘制的是着色的三维表面。

第5章 图形绘制

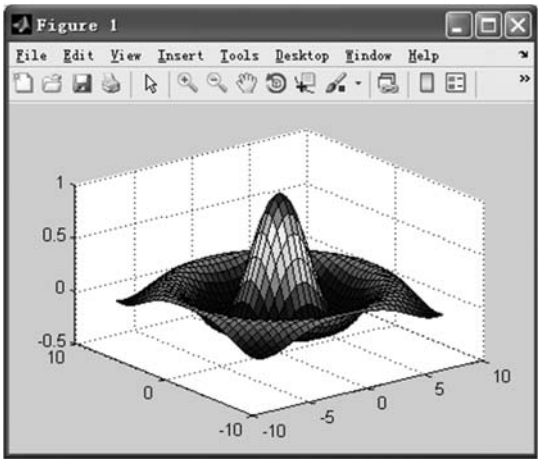


图5-28 三维着色曲面图

5.3.3 meshgrid()函数

meshgrid()函数为3D绘图生成X、Y矩阵。meshgrid()仅限于二维或三维Cartesian空间，meshgrid()更适合在二维或三维Cartesian空间解决问题。meshgrid()函数的语法如下：

[X,Y] = meshgrid(x,y)

把向量x和y指定的域转换成矩阵X、Y，用来实现两个变量和三维mesh()、surface()绘图的功能。输出矩阵X的行复制于向量x，输出矩阵Y的列复制于向量y。

[X,Y] = meshgrid(x)等同于[X,Y] = meshgrid(x,x)。

[X,Y,Z] = meshgrid(x,y,z)

三维矩阵用来实现三个变量和三维立体绘图的功能。

例5-3-3 上例中的前4行用meshgrid()函数代替。

解 程序如下：

```
[X, Y]=meshgrid(-8:0.5:8)
R=sqrt(X.^2+Y.^2)+eps;
Z=sin(R)./R;
mesh(Z)
```

运行结果与图5-27的相同。

绘制函数 $Z-(X-1)^2+(Y-2)^2=0$,

$$-2\le X\le 4,-1\le Y\le 5$$

```
x=-2:4;
y=-1:5;
[X,Y]=meshgrid(x,y);
Z=-(X-1).^2-(Y-2).^2;
mesh(X,Y,Z)
xlabel('x'),ylabel('y'),zlabel('z')
figure
surf(X,Y,Z)
xlabel('x'),ylabel('y'),zlabel('z')
```


5.3.4 meshc()和meshz()函数

1．meshc()函数

meshc()与mesh()函数的调用方式相同，只是该函数在mesh()的基础上又增加了绘制相应等高线的功能。

```
[x,y]=meshgrid([-4:.5:4]);
z=sqrt(x.^2+y.^2);
figure
meshc(z)
```

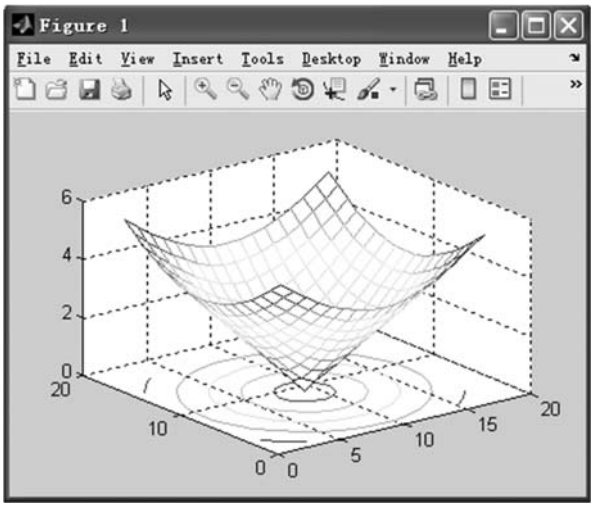


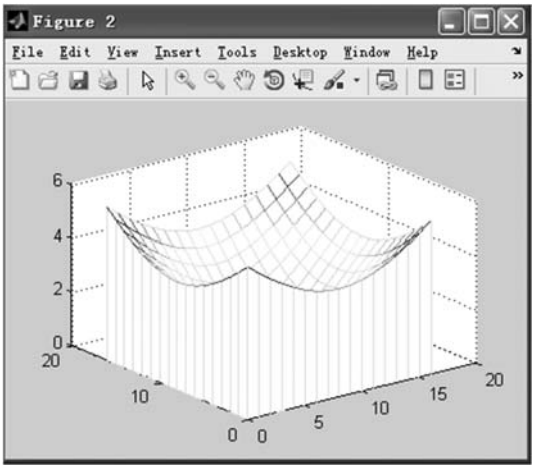
图5-29 meshc图

2．meshz()函数

meshz()与mesh()函数的调用方式也相同，该函数增加了z轴铅垂线。

例5-3-4 已知以下程序：

```
[x,y]=meshgrid([-4:.5:4]);
z=sqrt(x.^2+y.^2);
figure
meshz(z)
```



5.3.5 彗星图

彗星图是一个动画图，其中彗星头(一个圆)跟踪屏幕上的数据点，彗星体是尾随彗星头后动态画出的拖曳线段，是跟踪整个函数的实线。函数comet()、comet3()可用来绘制2D和3D彗星图，其调用格式如下：

- comet(y)：显示向量y的彗星图。
- comet(x,y)：显示向量y相对于向量x的彗星图。
- comet(x,y,p)：指定彗星体的长度为：p*length(y)，p默认为0.1。
- comet3(z)、comet3(x,y,z)、comet3(x,y,z,p)：绘制3D彗星图。

例5-3-5 已知以下程序：

```
t = 0:.01:2*pi;
x = cos(2*t).*(cos(t).^2);
y = sin(2*t).*(sin(t).^2);
comet(y);
pause
comet(x,y);
pause
comet3(x,y,t);
```

运行程序，所绘制的2D彗星图如图5-32和图5-33所示。

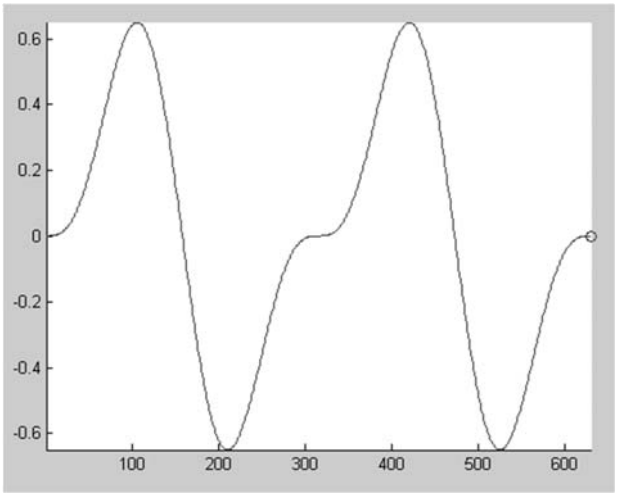


图5-32 comet(y)

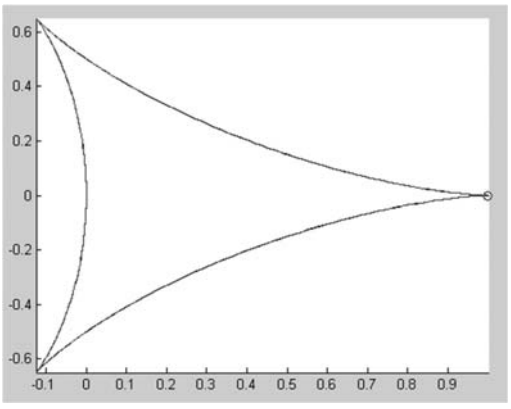


图5-33 comet(x, y)

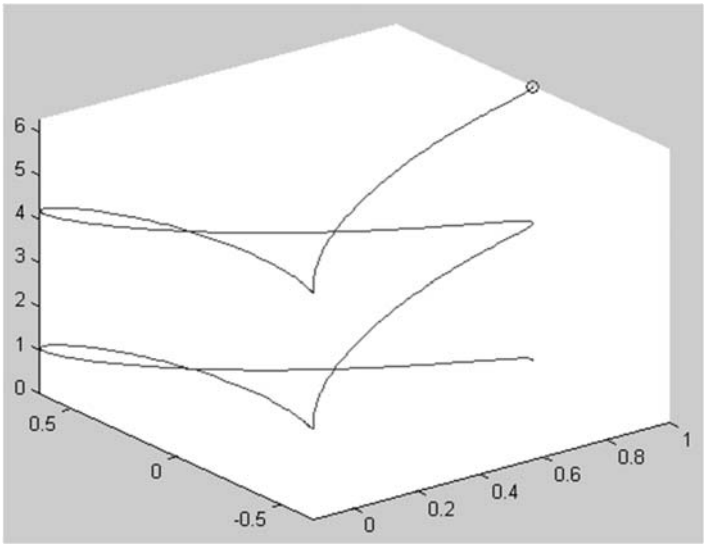


图5-34 comet3(x,y,t)绘制3D彗星图

5.4 绘图控制

5.4.1 图形窗口的创建、控制与figure命令

1．图形窗口的创建

图形窗口的创建使用figure命令。figure命令有以下几种形式：

(1) figure：以默认属性创建一个独立的图形窗口，默认名称和编号为figure 1、……。

(2) figure('PropertyName',PropertyValue,...)：按照指定的属性创建图形窗口。

图形对象figure的属性包含公共属性和特有属性，控制图形的外观和显示特点。一部分属性如表5-3所示。

表 5-3 图形对象 figure 的部分属性

属 性	描 述
BeingDeleted	当对象的 DeleteFcn()函数被调用后，该属性的值为 on
BusyAction	控制 MATLAB 图形对象句柄响应函数点中断方式
ButtonDownFcn	当单击按钮时执行响应函数
Children	该对象所有子对象的句柄
Clipping	打开或关闭剪切功能(只对坐标轴子对象有效)
CreateFcn	当对应类型的对象创建时执行
DeleteFcn	删除对象时执行该函数
HandleVisibility	用于控制句柄是否可以通过命令行或者响应函数访问
HitTest	设置当鼠标点击时是否可以使选中对象成为当前对象

第5章 图形绘制

属 性	描 述
Interruptible	确定当前的响应函数是否可以被后继的响应函数中断
Parent	该对象的上级(父)对象
Selected	表明该对象是否被选中
SelectionHighlight	指定是否显示对象的选中状态
Tag	用户指定的对象标签
Type	该对象的类型
UserData	用户想与该对象关联的任意数据
Visible	设置该对象是否可见
Position	4 元素向量，定义窗口位置：rect = [left, bottom, width, height]
MenuBar	none | {figure}，控制标准主菜单栏的显示，默认为显示(figure)，不控制使用 uimenu 命令建立的自定义菜单
Toolbar	none | {auto} | figure，控制工具栏是否显示。none 为不显示图形窗口 figure 工具栏。auto 为默认，显示图形窗口 figure 工具栏。但是如果添加了 uicontrol 控件，就删除 figure 工具栏。figur 为显示图形窗口 figure 工具栏

(3) figure(h)：如果句柄h对应的窗口已经存在，该命令使得该图形窗口为当前窗口；如果不存在，则创建以h为句柄的窗口。

(4) h = figure(...)：返回图形窗口的句柄。

在命令窗口中输入命令“figure”，按下回车，生成的图形窗口如图5-35所示。

第5章 图形绘制

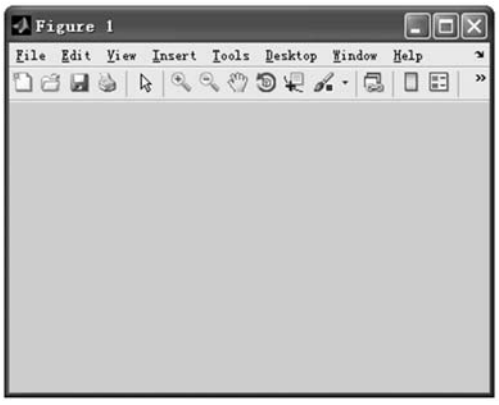


图5-35 图形窗口

2．关闭图形窗口命令

close命令用于关闭指定的图形窗口。

3．设置为当前窗口图形

图形可以在当前窗口输出或在其他窗口输出，当前窗口的图形是输出的目标图形。可使用两种不同的方法设置要输出的图形为当前图形，其结果有以下两种类型：

(1) figure(h)：设置h为当前图形、可视并且在其他窗口前面显示；

(2) set(0,'CurrentFigure',h)：设置h为当前图形，不改变其可视性和其他性质。

4．指定图像位置和大小

例如创建一个figure窗口，位置是在屏幕左上角、大小是1/4，可使用根对象ScreenSize的属性来确定屏幕尺寸的大小。ScreenSize是一个4元素的向量：[left, bottom, width, height]，包含了计算机屏幕的尺寸。

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

在此使用了figure命令的第二种形式，PropertyName= Position，PropertyValue=[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2]，但是显示的只有图像窗口，如果要显示完整的窗口，包括菜单、窗口标题、工具条及边线轮廓等，可使用OuterPosition、Name、NumberTitle等属性设置。

5．编辑图形的属性

创建图形窗口后，用户可以对其属性进行编辑。编辑图形的属性可以通过两种方式进行：一是通过属性编辑器；二是通过set()函数。

在图形窗口中，选择view菜单中的Porperty Editor选项，激活属性编辑器，如图5-36所示。

在该窗口中可以设置标题、颜色表等属性。若要对更多属性进行设置，可以通过点击“More Properties…”项打开属性管理器，在此编辑更多属性，如图5-37所示。

除此之外，还可以通过get()函数和set()函数对图形窗口的属性进行查看和编辑。

第5章 图形绘制

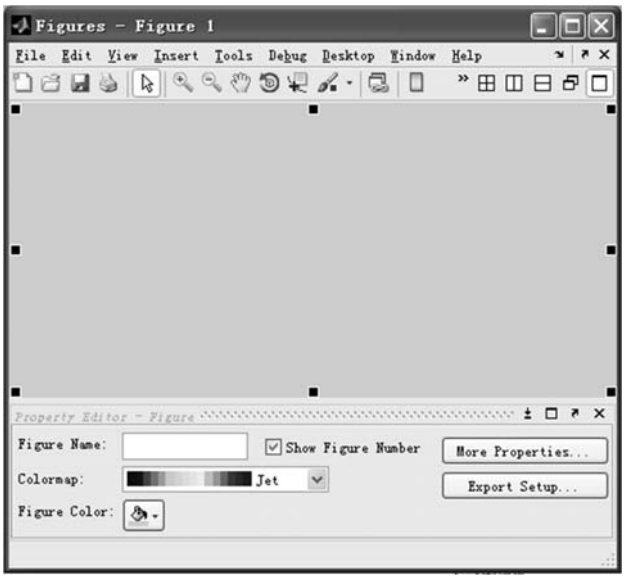


图5-36 属性编辑器

图5-37 属性管理器

6．图形窗口的菜单栏

图形窗口的菜单栏有以下几项：

(1) File菜单。File菜单与Windows系统的其他菜单类似，包括“新建”、“保存”、“打开”等命令。

(2) Edit菜单。包括以下几项：

- Copy Options…：将图形复制到剪切板。
- Figure Properties…：点击该选项，激活属性编辑器。在该窗口中可以设置图形的属性，包括图形窗口的标题、颜色映射表、图形彩色等。另外，点击“More Properties…”项可以设置更多属性，点击“Export Setup…”项可以设置图像导出属性。

- Axes Properties…：点击该选项弹出窗口如图所示。在该窗口中可以设置图形坐标系的属性，包括标题、坐标轴标记、范围等。
- Current Object Properties…：设置当前对象的属性，即图形中当前选中的对象，包括坐标轴、曲线、图形等。
- Color Map…：用于设置图形的颜色表。

(3) Insert菜单：在图像中插入对象，如箭头、直线、椭圆、长方形、坐标轴等。

(4) Tools菜单：包括一些常用图形工具，如平移、旋转、缩放、视点控制等；另外，Tools菜单包含了两个数据分析工具，即Basic fitting 工具和Data Statistics工具，用于对图像中的数据进行基本的分析和拟合等。

5.4.2 图形保持与多重线绘制

1．图形保持

当采用绘图命令plot时，MATLAB 默认在当前图形窗口中绘制图像，如果不存在图形窗口，则新建一个图形窗口。如果该窗口中已经存在图像，则将其清除，绘制新的图像。

如果要保持原有图像，并且在原图像中添加新的内容，在同一坐标系内画出多个图像，则可以使用hold命令。该命令的用法为：

- hold on：打开图形保持功能；保留当前图形与当前坐标轴的属性值，后面的图形命令只能在当前存在的坐标轴中增加图形。但是，当新图形的数据范围超出了当前坐标轴的范围，则命令会自动地改变坐标轴的范围，以适应新图形。

- hold off：关闭图形保持功能。
- hold all：当利用函数ColorOrder()和函数LineStyleOrder()设置线型和颜色列表时，该命令用于打开图形保持功能，并保持当前的属性。关闭图形保持时，下一条绘图命令将回到列表的开始处，打开图形保持时，将从当前位置继续循环。
- hold：改变当前的图形保持状态，在打开和关闭中间切换。
- hold(axes_handle,...)：对指定坐标系进行操作。

当hold on命令执行后，所有的新的图像都会叠加在原来存在的图像上。hold off命令可恢复默认情况，用新的图像来替代原来的图像。

例5-4-1 在同一坐标轴内画出sinx和cosx的图像，如图5-38所示，其代码如下：

```
x=-pi:pi/20:pi;
y1=sin(x);
y2=cos(x);
plot(x,y1,'b-');
hold on;
plot(x,y2,'k--');
hold off;
legend ('sin x','cos x');
```

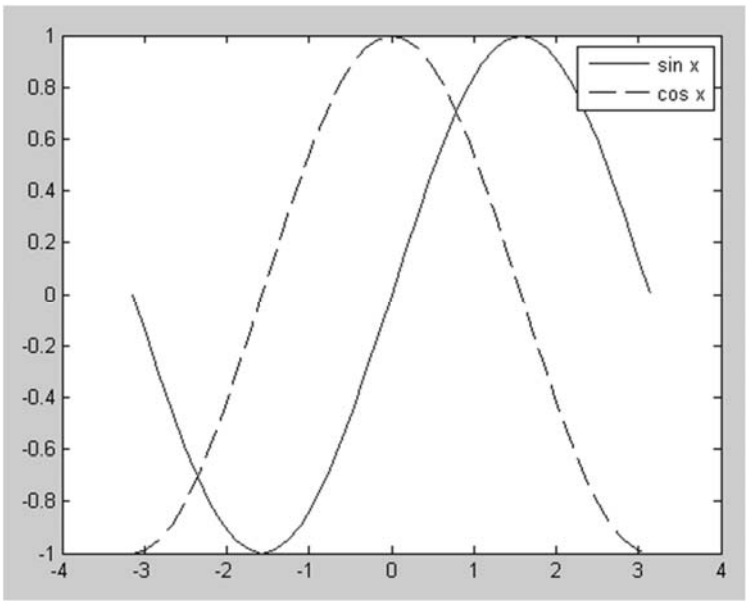


图5-38 在同一坐标轴内画出sinx和cosx的图像

第5章 图形绘制

2．绘制多重线

有三种在一个单线图上绘制多重线的办法。

(1) 利用plot命令的多变量方式绘制。多变量方式绘图是允许不同长度的向量显示在同一图形上，其语法格式如下：

plot(x1,y1,x2,y2,...,xn,yn)

x1,y1,x2,y2,...,xn,yn是成对的向量，每一对x, y在图上产生如上方式的单线。

(2) 使用hold命令。第二种方法也是利用plot命令绘制，但需要hold on/off命令来配合，其语法格式如下：

```
plot(x1,y1)
hold on
plot(x2,y2)
hold off
plot(x1,y1)
```

第5章 图形绘制

(3) 代入矩阵。第三种方法还是利用plot命令绘制，但需代入矩阵。如果plot用于两个变量plot(x,y)，并且x、y是矩阵，则有以下情况：

- 如果y是矩阵、x是向量，plot(x,y)用不同的画线形式绘出y的行或列及相应的x向量，y的行或列的方向与x向量元素的值选择是相同的。
- 如果x是矩阵、y是向量，则除了x向量的线族及相应的y向量外，以上的规则也适用。
- 如果x、y是同样大小的矩阵，plot(x,y)绘制x的列及y相应的列。

5.4.3 子图控制与subplot()函数

在绘图过程中，若要在一个图行窗口中并行显示多幅图像，就需要使用subplot()函数，其调用格式如下：

subplot(m,n,p)

subplot()函数把一个图形窗口分割成m × n个子区域，按m行、n列排列，这些子图像从左向右从上到下编号。用户可以通过参数p调用个各子绘图区域进行操作，并选择子图像p来接受当前所有画图命令。例如，命令subplot(2,3,4)将会创建6个子图像，而且subplot 4是当前子图像。

例5-4-2 绘制子图的程序代码如下：

```
x=0:0.1*pi:2*pi;
subplot(2,2,1)
plot(x,sin(x),'-*');
title('sin(x)');
subplot(2,2,2)
plot(x,cos(x),'--o');
title('cos(x)');
subplot(2,2,3)
plot(x,sin(2*x),'-.*');
title('sin(2x)');
subplot(2,2,4);
plot(x,cos(3*x),':d')
title('cos(3x)')
```

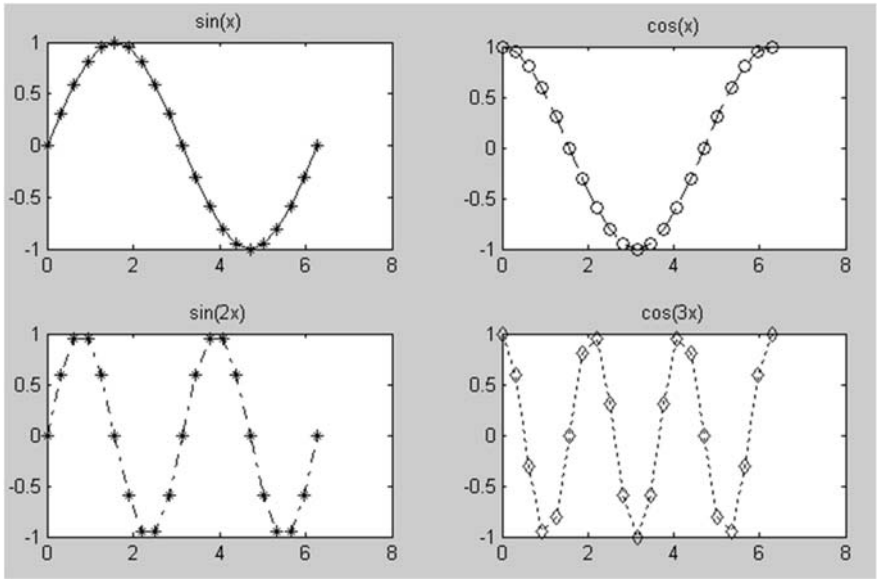


图5-39 绘制子图

5.4.4 图形的注释和标记

图形的注释和标记包括以下内容：

- 图题的标注。
- 坐标轴的标签。
- 文本标注和交互式文本标注。
- 图例的添加。
- 坐标网格的添加。
- 使用矩形或是椭圆在图形中圈出重要部分。

如果图形既没有x轴和y轴的标注，也没有标题，那么用xlabel、ylabel、title命令可以加标注和标题。

表 5-4 MATLAB 图形标注和标题命令

命 令	用 途
title	图形标题
xlabel	x 坐标轴标注
ylabel	y 坐标轴标注
text	标注数据点
grid	给图形加上网格
hold	保持图形窗口的图形

第5章 图形绘制

1．图题的标注

在MATLAB中，标题与文本注释不同，文本注释可以位于图形中的任何部分，标题位于图形的顶部，是一个文本串，并且标题不随图形的改变而改变。

在MATLAB中，通常可以使用三种方式给图形添加图题：

(1) 使用Insert 菜单中的Title命令；

(2) 使用属性编辑器(Property Editor)；

(3) 使用title函数。

title('string')：在图形窗口的顶部中间位置直接输出文本string。

title(fname)：在图形窗口的顶部中间位置，根据文件名称fname指定的文本输出。

2．坐标轴的标签

在MATLAB中，添加坐标轴标注的方法与添加标题的方法基本相同。可以使用如下三种方式给图形的坐标轴添加标签：

(1) 使用Insert菜单下的Label选项；

(2) 使用属性编辑器(Property Editor)添加坐标轴标签：打开Tools菜单，选择Edit Plot命令，激活图形编辑状态。在图形框内双击空白区域，调出属性编辑器；也可以采取在图形框内右击，从弹出的菜单中选择Properties项的方式调出属性编辑器；或者是在View菜单中选择Property Editor项，在xlabel、ylabel选项组中添加标签的文本内容。

(3) 使用MATLAB的添加标签命令xlabel、ylabel、zlabel分别为x轴、y轴、z轴添加标注。

- xlabel('string')：在x轴中间位置直接输出文本string。
- xlabel(fname)：在x轴中间位置，根据文件名称fname指定的文本输出。
- xlabel('标注文本','PropertyName',PropertyValue,...)：根据属性名称和属性值输出文本，这里的属性是标注文本的属性，包括字体大小、字体名、字体粗细等。

ylabel、zlabel使用方法与此相同。

3．文本标注

用户可以在MATLAB图形窗口的任意地方添加文本注释。使用text()函数进行文本标注，其调用格式如下：

- text()函数：它是一个底层函数，用于创建文本图形对象，该函数可以在图形中的指定位置添加文本注释。
- text(x,y,'string')、text(x,y,z,'string')：在二维、三维图形中，在指定位置添加、输出文本string。
- text(x,y,z,'string','PropertyName',PropertyValue....)：在指定位置，根据属性名称和属性值添加、输出文本string。

4．使用legend命令或函数添加图例

图例可以对图像中的各种内容做出注释，每幅图像可以包含一个图例。为了更好地区分所绘制的多条曲线，可以使用图例加以说明，以对它们表示的数据进行更准确的区分。通常，可使用如下方法生成图例。

(1) legend()函数可以在任何图形上添加图例。对于曲线，legend()函数为每条曲线生成一个标志，该标志包括线型示例、标记和颜色；对于填充图，legend()函数的标记为该区域的颜色。

(2) 通过legend()函数来指定图例中的文本，对图例进行显示控制或者编辑图例的属性等。

(3) 利用legend()函数在图例中添加文本的指令有：

legend('string1','string2',...)、

legend(h, 'string1', 'string2', ...)，在(h 指定的)图像中添加图例，图例中的文本通过字符串string1、string2等指定，字符串的顺序与图形对象绘制的顺序对应，字符串的个数对应图例中对象的个数。

例如，legend('cos_x','sin_x',1)表示在右上角用指定的文本显示图例，数字1代表右上角，2代表左上角，3、4分别代表左下角和右下角。

legend(string_matrix)、legend(h,string_matrix)表示在(h指定的)图像中添加图例，图例中的文本由字符串矩阵string_matrix指定；

legend(axes_handle, ...)表示在由坐标系句柄axes_handle指定的坐标系中添加图例。

5．坐标网格的添加

在图形绘制过程中，为了更精确地知道图形上某点的坐标，则需要通过绘制坐标网格来定位。在MATLAB中通过grid()函数来实现这一功能：

- grid off：关闭坐标网格命令；
- grid on：打开坐标网格命令，在图形中绘制坐标网格；
- grid mirror：使用更细化的网格命令；
- grid(AX,…)：使用AX坐标系代替当前坐标系命令。

第5章 图形绘制

例5-4-3 图形注释和标记演示。

解 程序如下：

```
x=linspace(-3,5,100);
y1=cos(x); y2=sin(x);
plot(x,y1,x,y2)
title('正弦曲线  余弦曲线');
text(3,0.2,' \leftarrow sin(\pi)','FontSize',18)
text(1.5,0,' \leftarrow cos(\pi)','FontSize',18)
text(0, -0.9,'这是文本标注的演示');
xlabel('(x)')
ylabel('y1、y2 ')
legend(' y1=cos(x)',' y2=sin(x)',1)
grid on
```

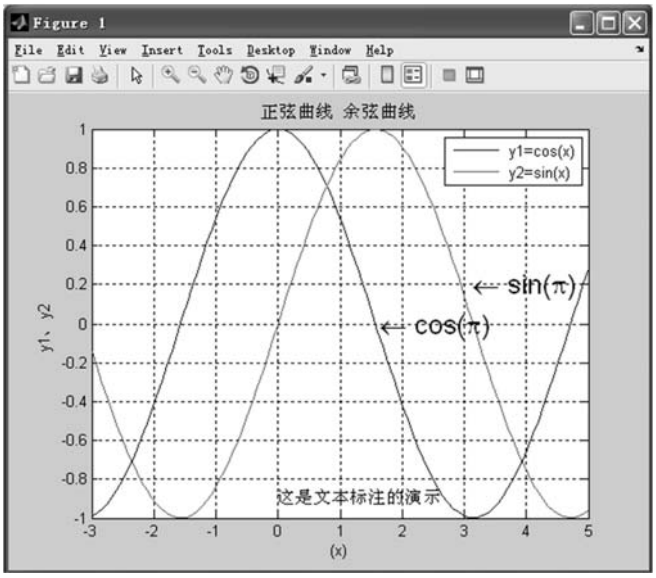


图5-40 图形注释和标记

5.4.5 线型和颜色的控制

如果不指定划线方式和颜色，MATLAB会自动选择点的表示方式及颜色，也可以用不同的符号指定不同的曲线绘制方式。例如：

```
plot(x,y,'*')                    //用'*'作为点绘制的图形
plot(x1,y1,':',x2,y2,'+')        //用':'画第一条线，用'+'
```

画第二条线线型、点标记及颜色的取值有如表5-6所示几种。

表 5-6 线型和颜色控制符

线型		点标记		颜色	
-	实线	.	点	y	黄
:	虚线	o	小圆圈	m	棕色
-.	点划线	x	叉子符	c	青色
- -	间断线	+	加号	r	红色
		*	星号	g	绿色
		'square' 或 s	方形	b	蓝色
		'diamond' 或 d	菱形	w	白色
		∧	朝上三角	k	黑色
		∨	朝下三角		
		>	朝右三角		
		<	朝左三角		
		'pentagram'或 p	五角星		
		'hexagram' 或 h	六角星		

例如：

```
t=-3.14:0.2:3.14
x=sin(t);  y=cos(t);
plot(t,x, '+r',t,y, '-b')
```

由此程序绘制不同线型与颜色的sin及cos图形，如图5-41所示。

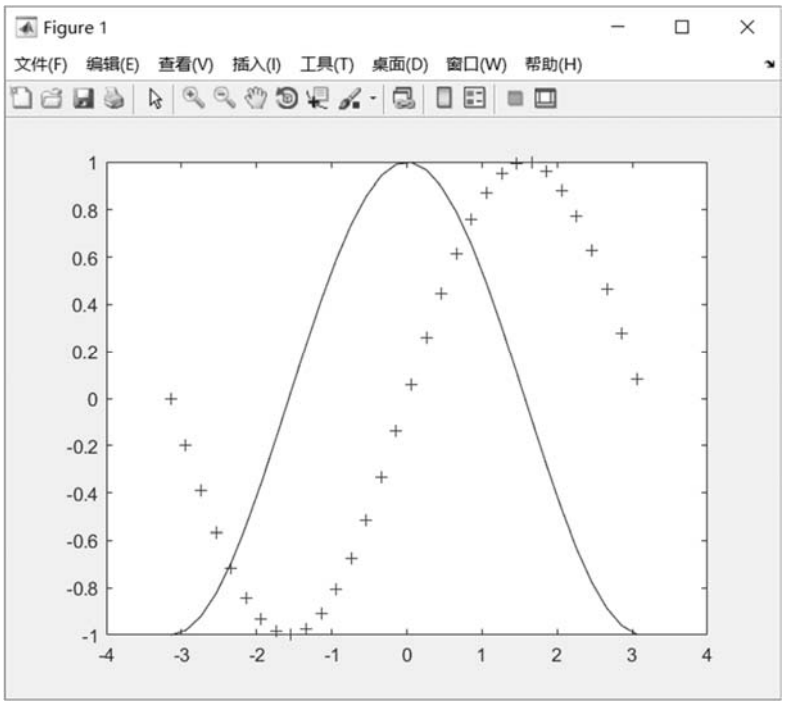


图5-41 不同线型与颜色的sin、cos图形

5.4.6 坐标轴控制

1．坐标轴的控制函数axis()

在默认情况下，MATLAB 会根据绘图命令和数据自动选择坐标轴，图像的X、Y轴的范围宽到能显示输入值的每一个点。但有时只显示这些数据的一部分则更为有利，因为用户可指定坐标轴，以满足特殊的需求。在MATLAB中，使用函数axis()来控制坐标轴，调用格式如下：

- axis([xmin xmax ymin ymax])：此函数将会返回一个4元素行向量[xmin xmax ymin ymax]，其中，xmin、xmax、ymin、ymax指定当前图像中x轴和y轴的上、下限范围。

- axis([xmin xmax ymin ymax zmin zmax cmin cmax])：此函数指定当前图像中x轴、y 轴和z轴的范围。
- v = axis：此函数返回当前图像中x轴、y轴和z轴的范围。当图像是二维时，返回结果有四个元素；当图像是三维时，返回结果有六个元素。
- axis auto：自动模式，使图形的坐标范围满足图中所有的图元素。根据x、y、z轴数据的最大及最小值自动选择坐标轴的范围。用户还可以对指定的坐标轴设置自动选择，如命令“auto x”自动设置x轴，命令“auto y z”自动设置y轴和z轴。

与axis相关的几条常用命令还有以下几个：

- axis equal：严格控制各坐标的分度使其相等，将横轴、纵轴的尺度比例设成相同值。
- axis square：使绘图区为正方形，横轴及纵轴比例是1∶1。
- axis on：恢复对坐标轴的一切设置。
- axis off：取消对坐标轴的一切设置。
- axis manual：以当前的坐标限制图形的绘制。
- axis normal：以预设值画纵轴及横轴。

为了说明axis的应用，我们将画出函数f(x)=sinx从2π到2π之间的图像，然后限定坐标的区域为0≤x≤π，0≤y≤1，其代码如下：

```
x=-2*pi:pi/20:2*pi;
y=sin(x);
plot(x,y);
axis([0 pi 0 1]);
title('sin(x)')
```


2．xlim()、ylim()、zlim()函数

xlim()、ylim()、zlim()函数用于设置或查询轴的限定值。xlim()的语法有以下几种：

- xlim；
- xlim([xmin xmax])；
- xlim('mode')；
- xlim('auto')；
- xlim('manual')；
- xlim(axes_handle, ...)。

```
x = linspace(0,10);
y = sin(x);
plot(x,y)
xlim([0 5])
```

其中，语法xlim([xmin xmax])用于设定x轴的范围。ylim()、zlim()函数的语法与xlim()函数的类似。

应用举例

```
Ts=0.02*pi;t=0:Ts:2*pi;
x=sin(t);ty=t(1:50);y=x(1:50);
subplot(2,4,1),plot(t,x);axis square,title('axis square')
subplot(2,4,2),plot(t,x);axis tight,title('axis tight')
subplot(2,4,3),plot(t,x);axis([0,7,-1.5,1.5]),title('axis defined')
subplot(2,4,4),plot(ty,y);axis equal,title('axis equal')
subplot(2,4,5),plot(t,x);axis image, title('axis image')
subplot(2,4,6),plot(t,x);axis off,title('axis off')
subplot(2,4,7),plot(t,x);axis on,title('axis on')
subplot(2,4,8),plot(t,x);axis auto,title('axis auto')
xlim([0 4]);
```


5.5 特殊图形的绘制

5.5.1 使用bar()函数绘制柱状图

1．绘制二维柱状图

函数bar()和barh()用于绘制二维柱状图，分别绘制纵向和横向图形。在默认情况下，bar()函数绘制的条形图将矩阵中的每个元素表示为“条形”，“条形”的高度表示元素的大小，横坐标上的位置表示不同的行。在图形中，每一行的元素会集中在一起。

MATLAB中主要有四个函数用于绘制条形图，如表5-10所示。

表 5-10 绘制条形图函数

函 数	说 明	函 数	说 明
bar	绘制纵向条形图	bar3	绘制三维纵向条形图
barh	绘制横向条形图	bar3h	绘制三维横向条形图

第5章 图形绘制

bar()函数的调用格式如下：

(1) bar(Y)：使用bar()函数水平或垂直显示、绘制向量或矩阵值，bar()函数不接受多变量。bar(Y)对Y绘制条形图。如果Y为矩阵，Y的每一行聚集在一起。横坐标表示矩阵的行数，纵坐标表示矩阵元素值的大小。

(2) bar(x,Y)：指定绘图的横坐标。x的元素可以非单调，但是x中不能包含相同的值。

(3) bar(...,width)：指定每个条形的相对宽度。条形的默认宽度为0.8。

(4) bar(...,‘style’)：指定条形的样式。style的取值为“grouped”或者“stacked”，如果不指定，则默认为“grouped”。两个取值的意义分别为：

- grouped：绘制的图形共有m组，其中**m为矩阵Y的行数，每一组有n个条形，n为矩阵Y的列数**，Y的每个元素对应一个条形。
- stacked：绘制的图形有m个条形，**每个条形为第m行的n个元素的和，每个条形由多个(n个)色彩构成**，每个色彩对应相应的元素。

第5章 图形绘制

(5) bar(...,‘bar_color’)：指定绘图的色彩，所有条形的色彩由“bar_color”确定，“bar_color”的取值与plot绘图的色彩相同。

bar(x)显示x向量元素的条形图。输入下列命令：

```
x = -2.9:0.2:2.9;
bar(x,exp(-x.*x),'r')
```

绘制出二维条状图形，如图5-54所示。

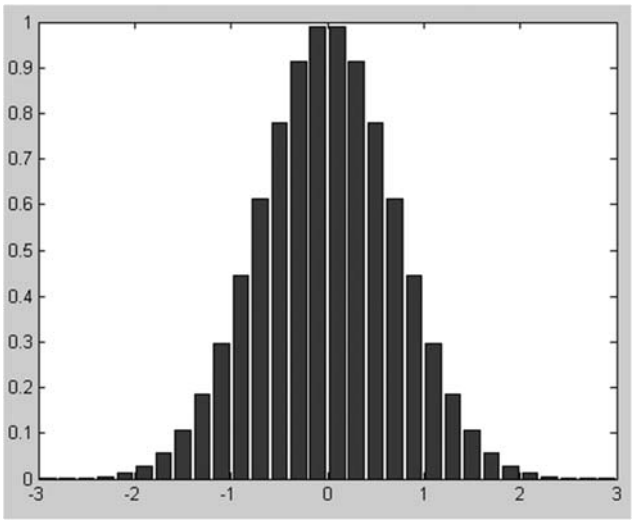


图5-54 二维柱状图形

2．绘制三维柱状图

bar3()和bar3h()用于绘制三维柱状图，分别绘制纵向图形和横向图形。这两个函数的用法相同，并且与函数bar()和barh()的用法类似，读者可以与bar()函数和barh()函数进行比较学习。下面以bar3()函数为例介绍这两个函数的用法。bar3()函数的调用格式如下：

(1) bar3(Y)：绘制三维条形图，Y的每个元素对应一个条形，如果Y为向量，则x轴的范围为[1:length(Y)]，如果Y为矩阵，则x轴的范围为[1:size(Y,2)]，即为矩阵Y的列数，图形中，矩阵每一行的元素聚集在相对集中的位置。

(2) bar3(x,Y)：指定绘制图形的行坐标，规则与bar函数相同。

(3) bar3(...,width)：指定条形的相对宽度，规则与bar函数相同。

(4) bar3(...,'style')：指定图形的类型，“style”的取值可以为“detached”、“grouped”或“stacked”，其意义分别为：

- detached：显示Y的每个元素，在x方向上，Y的每一行为一个相对集中的块；
- grouped：显示m组图形，每组图形包含n个条形，m和n分别对应矩阵Y的行和列；
- stacked：意义与bar中的参数相同，将Y的每一行显示为一个条形，每个条形包括不同的色彩，对应于该行的每个元素。

(5) bar3(...,LineSpec)：将所有的条形指定为相同的颜色，颜色的可选值与plot()函数的可选值相同。

例5-5-1 绘制三维柱状图。

解 程序如下：

```
Y = cool(7);
subplot(3,2,1)
bar3(Y,'detached')
title('Detached')
subplot(3,2,2)
bar3(Y,0.25,'detached')
title('Width = 0.25')
subplot(3,2,3)
bar3(Y,'grouped')
title('Grouped')
subplot(3,2,4)
bar3(Y,0.5,'grouped')
title('Width = 0.5')
subplot(3,2,5)
bar3(Y,'stacked')
title('Stacked')
subplot(3,2,6)
bar3(Y,0.3,'stacked')
title('Width = 0.3')
colormap([1 0 0;0 1 0;0 0 1])
```

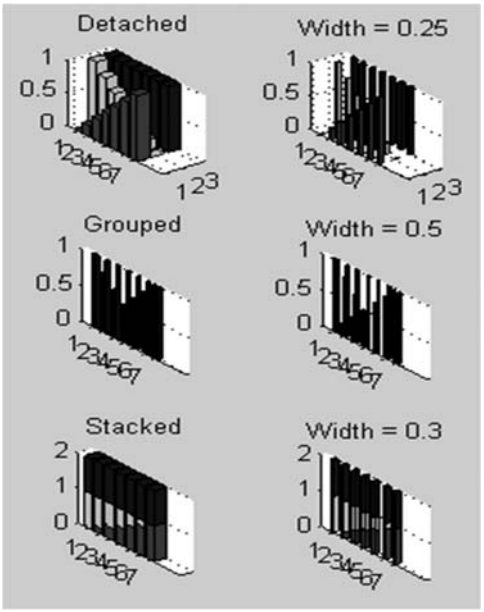


图5-55 三维柱状图

5.5.2 使用stairs()绘制阶梯图形

阶梯图主要用于绘制数字采样数据的时间关系曲线图，使用stairs()函数可以绘制阶梯状图形。stairs()函数的调用格式如下：

- stairs(Y)：绘制Y的元素的阶梯状图形。当Y是向量时，X轴的缩放范围是1～length(Y)，当Y是矩阵时，X轴的缩放范围是1～Y的行数。
- stairs(X,Y)：在X指定的位置绘制Y的元素的阶梯图形。X必须与Y的大小相同，当Y是矩阵时，X可以是行或列向量，例如：length(X) = size(Y,1)。

- stairs(...,LineSpec)：指定线型、符号和颜色等属性。

例如，输入下列命令：

```
x=0:0.25:10;
stairs(x,sin(x));
```

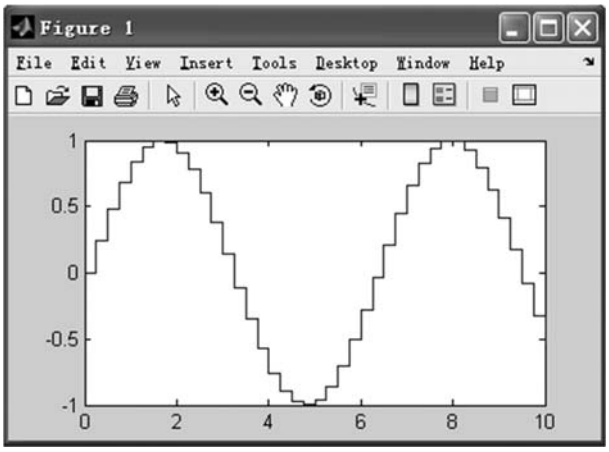



图5-56 阶梯图形

5.5.3 方向和速度矢量图形

MATLAB提供了一些函数用于绘制方向矢量和速度矢量图形，这些函数有compass()、feather()、quiver()和quiver3()。如表5-11所示。

表 5-11 绘制矢量图形的函数

函 数	功 能 描 述
compass	罗盘图，绘制、显示极坐标图形中的极点发散出来的矢量图
feather	羽状图，绘制向量，显示从一条水平线上均匀间隔的点所发散出来的矢量图。向量起点位于与 x 轴平行的直线上，长度相等
quiver	二维矢量图，绘制二维空间中指定点的方向矢量，显示由(u,v)矢量特定的二维矢量图
quiver3	三维矢量图，绘制三维空间中指定点的方向矢量，显示由(u,v,w)矢量特定的三维矢量图

第5章 图形绘制

1．罗盘图的绘制

在MATLAB中，罗盘图由函数compass()绘制，该函数的调用格式如下：

(1) compass(U,V)：绘制罗盘图，数据的x分量和y分量分别由U和V指定；

(2) compass(Z)：绘制罗盘图，数据由Z指定；

(3) compass(...,LineSpec)：绘制罗盘图，指定线型；

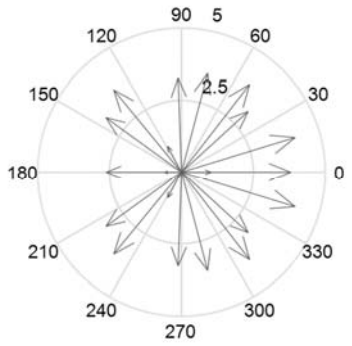
(4) compass(axes_handle,...)：在“axes_handle”指定的坐标系中绘制罗盘图；

(5) h = compass(...)：绘制罗盘图，同时返回图形句柄。

例5-6-2 绘制罗盘图。

解 程序如下：

```
M = randn(20,20);
Z = eig(M);
compass(Z)
```



2．羽状图的绘制

羽状图由函数feather()绘制，该函数的调用格式如下：

(1) feather(U,V)：绘制由U和V指定的向量；

(2) feather(Z)：绘制由Z指定的向量；

(3) feather(...,LineSpec)：指定线型；

(4) feather(axes_handle,...)：在指定的坐标系中绘制羽状图；

(5) h = feather(...)：绘制羽状图，同时返回图像句柄。

```
theta = -pi/2:pi/16:pi/2;
r = 2*ones(size(theta));
[u,v] = pol2cart(theta,r);
feather(u,v)
```

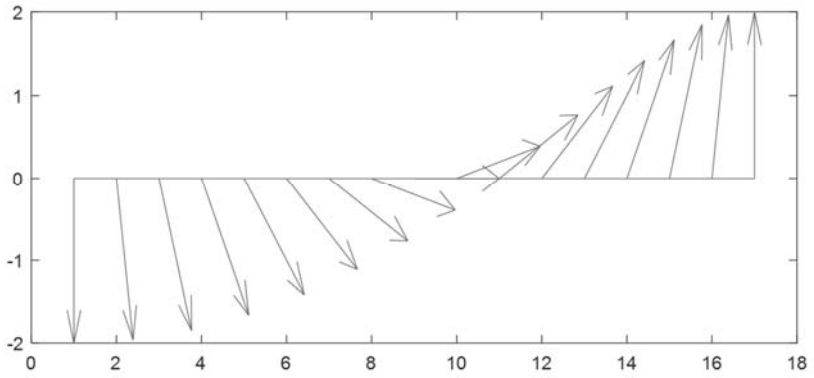


图5-58 绘制羽状图

3. 矢量图的绘制

矢量图在空间中指定点绘制矢量。矢量图通常绘制在其他图形中，显示数据的方向，如在梯度图中绘制矢量图用于显示梯度的方向。

MATLAB用于绘制二维矢量图和三维矢量图的函数，分别为quiver()和quiver3()，两个函数的调用格式基本相同。函数quiver()的主要调用格式如下：

(1) quiver(x,y,u,v)：绘制矢量图，参数x和y用于指定矢量的位置，u和v用于指定待绘制的矢量；

(2) quiver(u,v)：绘制矢量图，矢量的位置采用默认值。

函数quiver3()的主要调用格式如下：

(3) quiver3(x,y,z,u,v,w)：函数quiver3()使用元素(u,v,w)在点(x,y,z)绘制三维矢量图，u,v,w,x,y和z都是实数值，不是复数，并且大小相同。

(4) quiver3(z,u,v,w)：在矩阵z指定的等距离表面的点绘制三维矢量图，quiver3()根据它们之间的距离自动缩放，以防止它们重叠。

(5) quiver3(...,scale)：按照缩放系数scale自动缩放，以防止它们重叠。scale = 2 时，长度放大一倍；cale = 0.5时，长度缩小一倍；scale = 0 时，无缩放。

(6) quiver3(...,LineSpec)：LineSpec指定线型和颜色。

例5-5-4 绘制函数的梯度场。

解 (1) 使用下列程序绘制二维矢量图，如图5-59所示。

```
[X,Y] = meshgrid(-2:0.2:2);
Z = X.*exp(-X.^2 - Y.^2);
[DX,DY] = gradient(Z,.2,.2);
%gradient
hold on
contour(X,Y,Z)
quiver(X,Y,DX,DY)
colormap hsv;
hold off
```

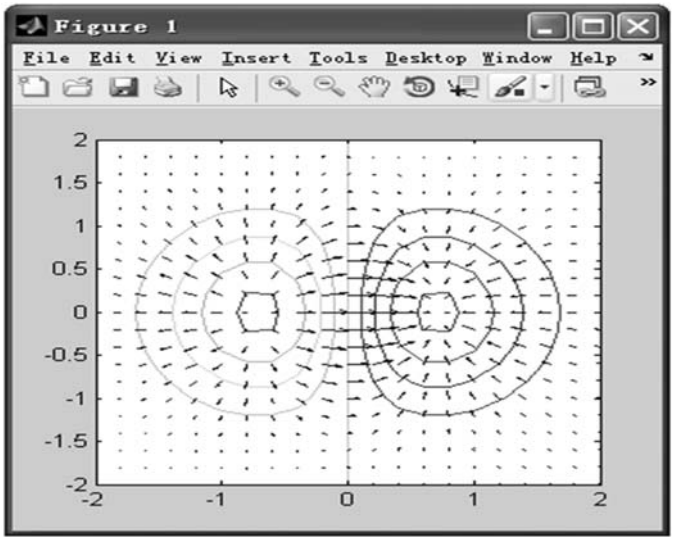


图5-59 二维矢量图

5.5.4 等值线的绘制

表5-12 等值线函数

函数名	功能描述
clabel	使用等值矩阵生成标注，在二维等值线中添加高度值标注，并将标注显示在当前图形
contour	绘制、显示指定数据矩阵Z的二维等高线图
contour3	绘制、显示指定数据矩阵Z的三维等高线图
contourf	绘制、显示矩阵Z的二维等高线图，并在各等高线之间用实体颜色填充
contourc	用于计算等值线矩阵，通常由其他函数调用
meshc	创建一个与二维等高线图匹配的网格线图
surfc	创建一个与二维等高线图匹配的曲面图

1．二维等值线

contour()、contour3()等函数用于绘制二维、三维等值线，其调用格式如下：

(1) contour(Z)：绘制矩阵Z的等值线，绘制时将Z在x-y平面上进行插值，等值线的数量和数值由系统根据Z自动确定；

(2) contour(Z,n)：绘制矩阵Z的等值线，等值线数目为n；

(3) contour(Z,v)：绘制矩阵Z的等值线，等值线的值由向量v确定；

(4) contour(X,Y,Z)、contour(X,Y,Z,n)、contour(X,Y,Z,v)：绘制矩阵Z的等值线，坐标值由矩阵X和Y指定，矩阵X、Y、Z的维数必须相同；

(5) contour(...,LineSpec)：利用指定的线型绘制等值线；

(6) [C,h] = contour(...)：绘制等值线，同时返回等值线矩阵和图形句柄。

例如，上例的函数用contour()函数绘制二维等值线，如图5-61所示。

```
[X,Y] = meshgrid(-2:0.2:2);
Z = X.*exp(-X.^2 -Y.^2);
contour(X,Y,Z)
colormap hsv
```

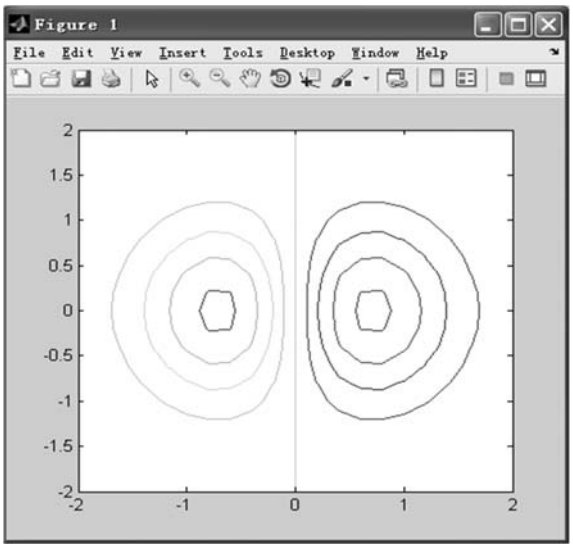


图5-61 二维等值线

2．三维等值线

contour3()函数用于绘制三维等值线，其调用格式与contour()函数的基本相同。

(1) contour3(Z)：绘制矩阵Z的三维等值线，Z看做是相对于x-y平面的高度，Z最少是包含2个不同值的2 × 2的矩阵，contour号和值基于Z的最小和最大值自动选择，x、y轴的范围是[1:n]和[1:m]，[m,n] = size(Z)。

(2) contour3(Z,n)：根据n的值绘制矩阵Z的三维等值线。

例如，上例用contour3()函数绘制三维等值线，如图5-62所示。

```
[X,Y] = meshgrid([-2:.25:2]);
Z = X.*exp(-X.^2-Y.^2);
contour3(X,Y,Z,30)
surface(X,Y,Z,'EdgeColor',[.8 .8 .8],'FaceColor','
none')
grid off
view(-15,25)
colormap cool
```

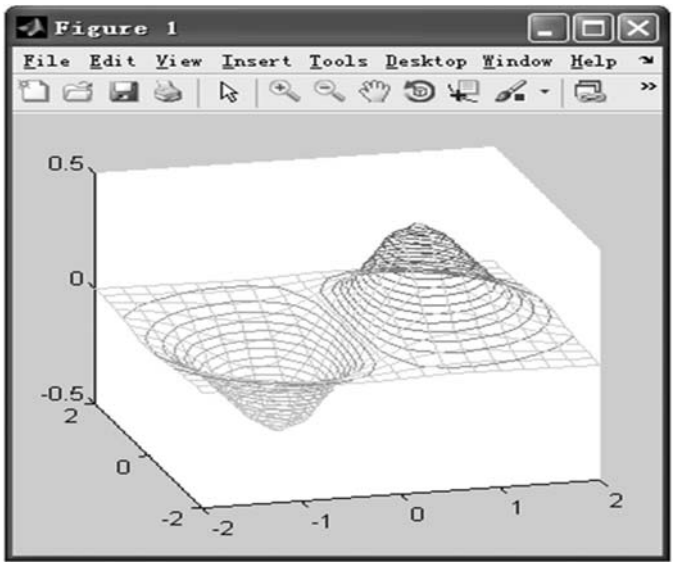



图5-62 三维等值线

5.5.5 饼形图

饼状图是一种统计图形，用于显示每个元素占总体的百分比。在统计学中，人们经常用饼形图来表示各个统计量占总量的份额，饼形图可以显示向量或矩阵中的元素占所有元素总和的百分比。MATLAB提供了pie()函数和pie3()函数，分别用于绘制二维饼形图和三维饼形图。函数pie()的调用格式如下：

(1) pie(X)：绘制X的饼状图，X的每个元素占一个扇形，其顺序为从饼状图上方正中开始，逆时针为序，分别为X的各个元素，如果X为矩阵，则按照各列的顺序排列。

- 在绘制饼状图时，如果X的元素和超过1，则按照每个元素所占有的百分比绘制图形；
- 如果X的元素的和小于1，则按照每个元素的值绘制图形，绘制的图形不是一个完整的圆形。

(2) pie(X,explode)：参数explode设置相应的扇形偏离整体图形，用于突出显示。explode是一个与X维数相同的向量或矩阵，其元素为0或者1，非0元素对应的扇形从图形中偏离。

(3) pie(...,labels)：标注图形，labels为元素为字符串的单元数组，元素个数必须与 X 的个数相同。

pie3()函数的调用方法与pie()函数相同。

例如：

```
x=[2,4,8,3];
explode = [0 1 0 0 ];
labels={'教授','副教授','讲师','助教'};
pie3(x,explode,labels)
```

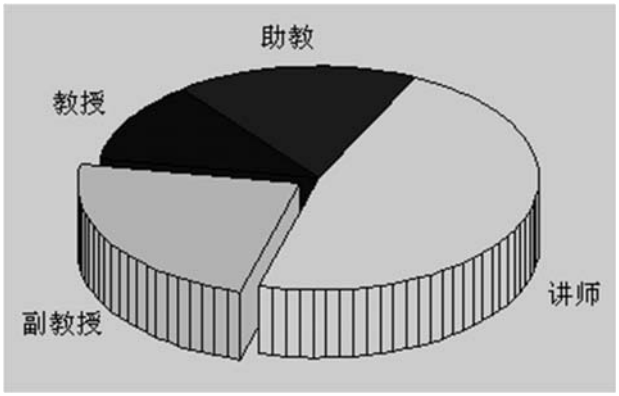


图5-63 带标注的三维饼状图

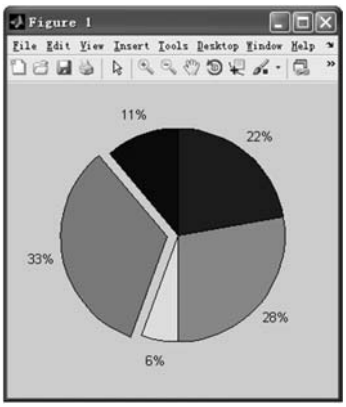
例5-5-5 绘制二、三维饼状图。

解 (1) 下列程序绘制二维饼状图，如图5-64所示。

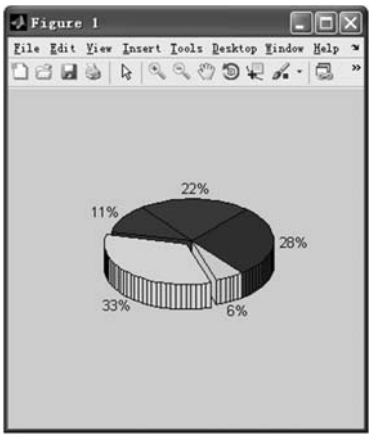
```
x = [1 3 0.5 2.5 2];
explode = [0 1 0 0 0];
pie(x,explode)
colormap jet
```

(2) 下列程序绘制三维饼状图，如图5-65所示。

```
x = [1 3 0.5 2.5 2];
explode = [0 1 0 0 0];
pie3(x,explode)
colormap jet
```



绘制二维饼状图



三维饼状图

自定义三维图像绘制——旋转体的绘制

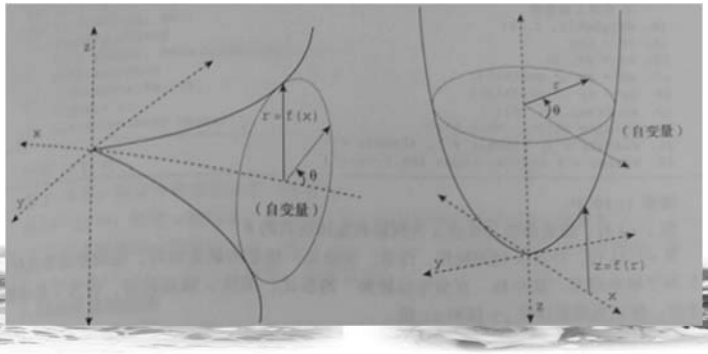
1、旋转连续函数

考虑围绕x轴和z轴旋转连续函数v=f(u)

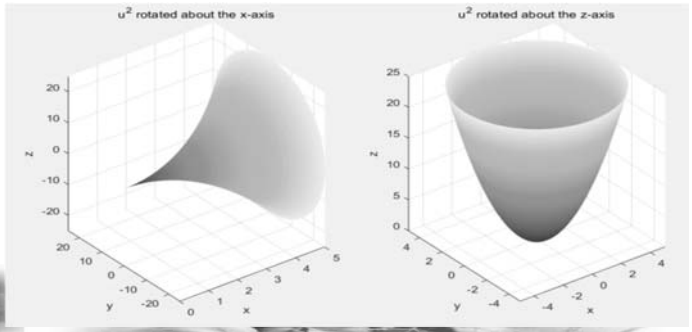
(1)围绕x轴旋转v=f(u)时，可以把方程看作r=f(x)，旋转的逻辑如图所示，x是自变量，y和z的值可通过“极坐标-直角坐标转换”得到：　$y = r\cos(\theta); z = r\sin(\theta)$

（2）围绕z轴旋转v=f(u)时，可以把方程看作z=f(r)，则得到

$$x = r\cos(\theta); y = r\sin(\theta)$$

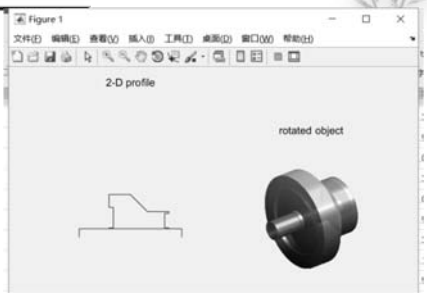


```
facets=100;
u=linspace(0,5,facets);
th=linspace(0,2*pi,facets);
[uu,tth]=meshgrid(u,th);
subplot(1,2,1);
rr = uu.^2;
xx=uu;
yy=rr.*cos(tth);
zz=rr.*sin(tth);
surf(xx,yy,zz,xx);
shading interp, axis tight
xlabel('x'),ylabel('y'),zlabel('z');
title('u^2 rotated about the x-axis')
subplot(1,2,2)
rr=uu;
zz=rr.^2;
xx=rr.*cos(tth);
yy=rr.*sin(tth);
surf(xx,yy,zz);
shading interp, axis tight
xlabel('x'),ylabel('y'),zlabel('z');
title('u^2 rotated about the z-axis')
```



2、旋转离散函数

旋转体的轮廓不是只有连续函数，机器零件的二维轮廓和该轮廓围绕x轴旋转后形成的图行如下图所示。如何画出零件立体图形？



代码实现：

```
u = [0 0 3 3 1.75 1.75 2 2 1.75 1.75 3 4 ...
    5.25 5.25 5 5 5.25 5.25 3 3 6 6];
v = [0 .5 .5 .502 .502 .55 .55 1.75 1.75 ...
    2.5 2.5 1.5 1.5 1.4 1.4 ...
    .55 .55 .502 .502 .5 .5 0];
subplot(1, 2, 1)
plot(u, v, 'k')
axis ([-1 7 -1 3]), axis equal, axis off
title('2-D profile')
facets = 200;
subplot(1, 2, 2)
[xx tth] = meshgrid( u, linspace(0,
2*pi, facets) );
rr = meshgrid( v, 1:facets);
yy = rr .* cos(tth);
zz = rr .* sin(tth);
surf(xx, yy, zz);
shading interp
axis square, axis tight, axis off
colormap bone
lightangle(60, 45)
alpha(0.8)
title('rotated object')
```

3、围绕任意轴旋转

不是只有围绕x轴、y轴或z轴旋转才能生成旋转体。将z=f(x)围绕任意轴旋转的最简单的方法是：

（1）计算将旋转轴沿x轴放置的矩阵；

（2）通过旋转变换u和v；

（3）将变换后的u和v围绕x轴旋转；

（4）在结果曲面上变换回来；

作业：

1、编写一个脚本，该脚本将创建三行两列6个子图。每个子图都要添加标题和x轴、y轴的标签。左上角的子图是y=cos(θ)，右上角的子图是y=cos(2θ),第二行的子图是y=cos(3θ)和y=cos(4θ)，第三行的子图是y=cos(5θ)和y=cos(6θ)，θ的范围都是-2π到2π

2、编写一个名称为thisPlot的脚本，该脚本将执行以下操作：

1）提示用户输入一个大于5的整数N；

2）计算1到N各个数字的阶乘，每个值都被保存到一个向量中；

3）绘制一个标题为‘Logarithmic Growth’的图形，在图中显示每个阶乘的对数。

4）在途中添加方程y=x的连续直线，x的范围是1到N

5）因为这些数字的数量级不同，所以使用plotyy在右边坐标轴绘制直线数据

3、编写一个名称为sineGraph的函数，其功能是在同一个图的区间[start,stop]内绘制4次正弦函数，start和stop是函数的参数。每个区间的点不同，具体为：

1）第一次绘制正弦函数时，应该有2个间隔均匀的点，即start 和stop

2）第二次绘制时应该有4个间隔均匀的点——start、stop和之间的两个点

3）第三次应该有8个间隔均匀的点，第四次应该有256个点；

4）添加图例、标题（标题为‘Multiple graphs on one plot’）和坐标轴的标签确保每条线的颜色不同。

5）该函数应该范围256个点的x值和y值

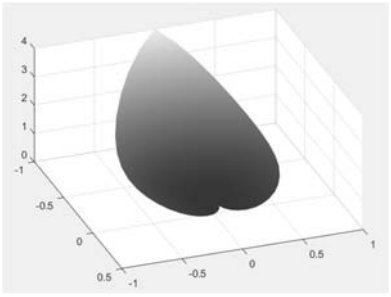
4、一个学校想要建一个钟楼，需要对其进行建模，将方程$z=1/(x^2+y^2)$作为模型，编写一个绘制钟楼的脚本。x、y的取值范围是：-0.75<=x<=0.75，数据间隔为0.05。设置坐标轴，使x、y的所有区域可见，并且z的范围在0到300之间。使用surf（）绘制图像。

5、画一颗爱心送给你的男/女朋友，绘制方法如下：

1）使用x值（范围是0到2π，间隔是0.05π）和y值（范围是0到1，间隔是0.05）创建一个网格[xx,yy]。

2）定义如下变量

```
c=[0.1+0.9*(π-abs(xx-π))/π].*yy
aa=c.*cos(xx)
bb=c.*sin(xx)
zz=(-2)*aa.^3+(3/2)*c.^2+0.5
```



3)使用surf（）函数绘制图形zz vs. aa和zz vs. bb，对色彩进行插值处理