

7 数据和函数的可视化

7.1 引导

7.1.1 离散数据和离散函数的可视化

【*例 7.1.1-1】用图形表示离散函数 $y = |(n-6)|^{-1}$ 。

```
n=0:12; %产生一组自变量数据
y=1./abs(n-6); %计算相应点的函数值
plot(n,y,'r*','MarkerSize',20) %用红花标出数据点
grid on %画坐标方格
Warning: Divide by zero.
```

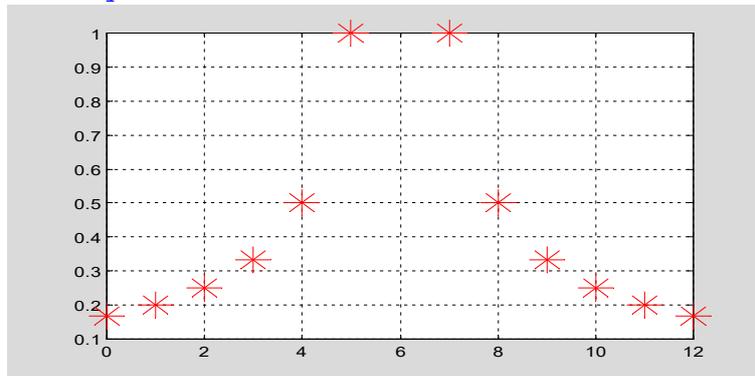


图 7.1.1-1 离散函数的可视化

7.1.2 连续函数的可视化

【*例 7.1.2-1】用图形表示连续调制波形 $y = \sin(t) \sin(9t)$ 。

```
t1=(0:11)/11*pi; % <1>
y1=sin(t1).*sin(9*t1);
t2=(0:100)/100*pi; % <3>
y2=sin(t2).*sin(9*t2);
subplot(2,2,1),plot(t1,y1,'r. '),axis([0,pi,-1,1]),title('子图 (1)')
subplot(2,2,2),plot(t2,y2,'r. '),axis([0,pi,-1,1]),title('子图 (2)')
subplot(2,2,3),plot(t1,y1,t1,y1,'r. ')
axis([0,pi,-1,1]),title('子图 (3)')
subplot(2,2,4),plot(t2,y2)
axis([0,pi,-1,1]),title('子图 (4)')
```

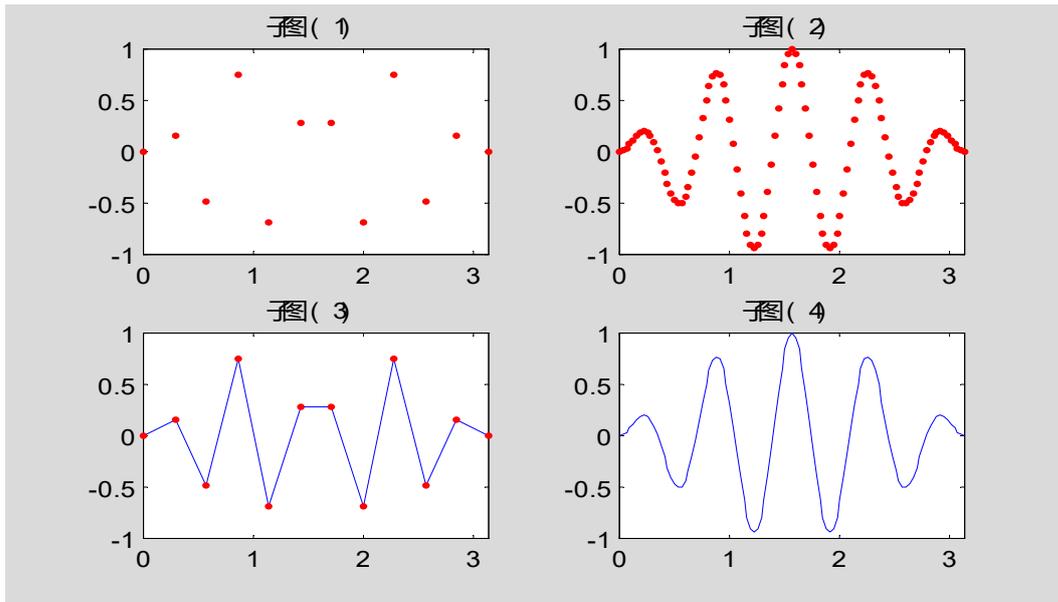


图 7.1.2-1 连续函数的图形表现方法

7.1.3 可视化的一般步骤

7.1.3.1 绘制二维图形的一般步骤

7.1.3.2 绘制三维图形的一般步骤

7.2 二维曲线绘图的基本操作

7.2.1 plot 的基本调用格式

【*例 7.2.1-1】简单例题，比较方便的试验指令。

```
t=(0:pi/50:2*pi)';k=0.4:0.1:1;Y=cos(t)*k;plot(t,Y)
```

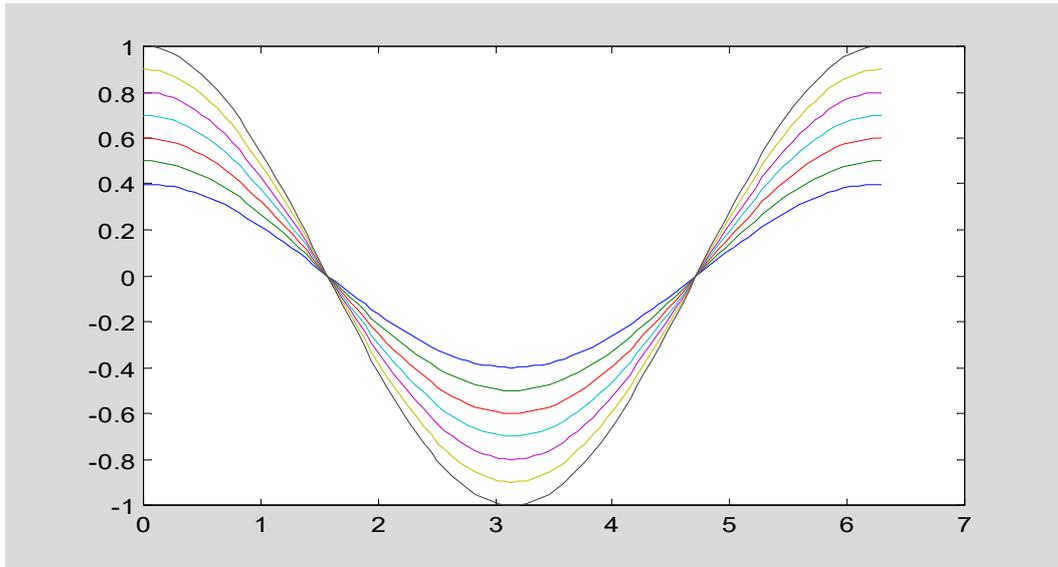


图 7.2.1-1 plot 指令基本操作演示

【*例 7.2.1-2】用图形表示连续调制波形 $y = \sin(t)\sin(9t)$ 及其包络线。

```

t=(0:pi/100:pi)'; %长度为 101 的时间采样列向量 <1>
y1=sin(t)*[1,-1]; %包络线函数值, 是 (101x2) 的矩阵 <2>
y2=sin(t).*sin(9*t); %长度为 101 的调制波列向量 <3>
t3=pi*(0:9)/9; % <4>
y3=sin(t3).*sin(9*t3); plot(t,y1,'r:',t,y2,'b',t3,y3,'bo') % <5>
axis([0,pi,-1,1]) %控制轴的范围 <6>

```

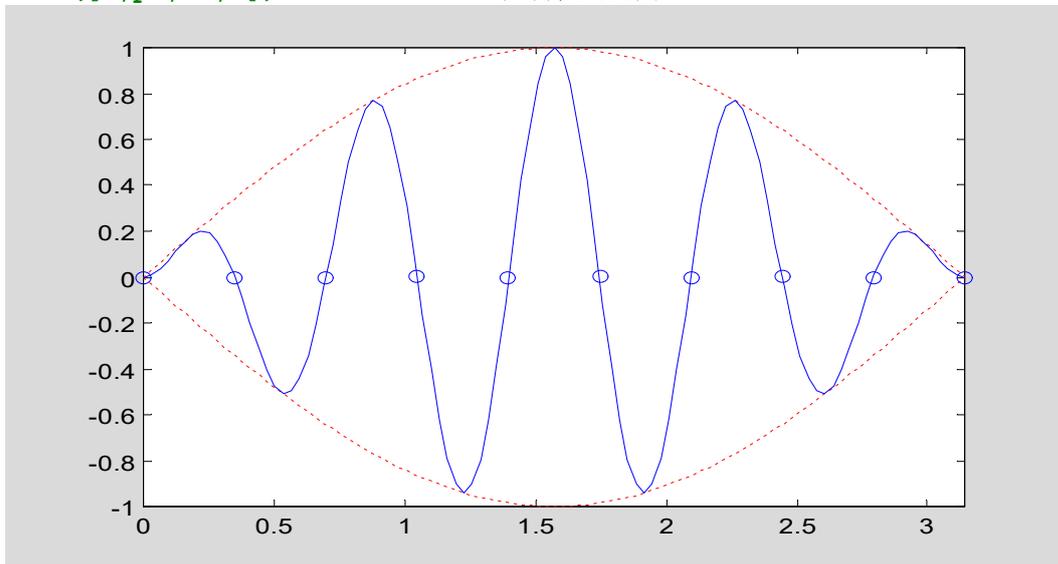


图 7.2.1-2

【*例 7.2.1-3】用复数矩阵形式画 Lissajous 图形。（在模拟信号时代，Lissajous 图形常用来测量信号的频率。）

```

t=linspace(0,2*pi,80)'; % <1>
X=[cos(t),cos(2*t),cos(3*t)]+i*sin(t)*[1,1,1]; % (80x3) 的复数矩阵 <2>
plot(X) % <3>

```

```
axis square
legend('1','2','3')
```

```
%使坐标轴长度相同
%图例
```

<4>

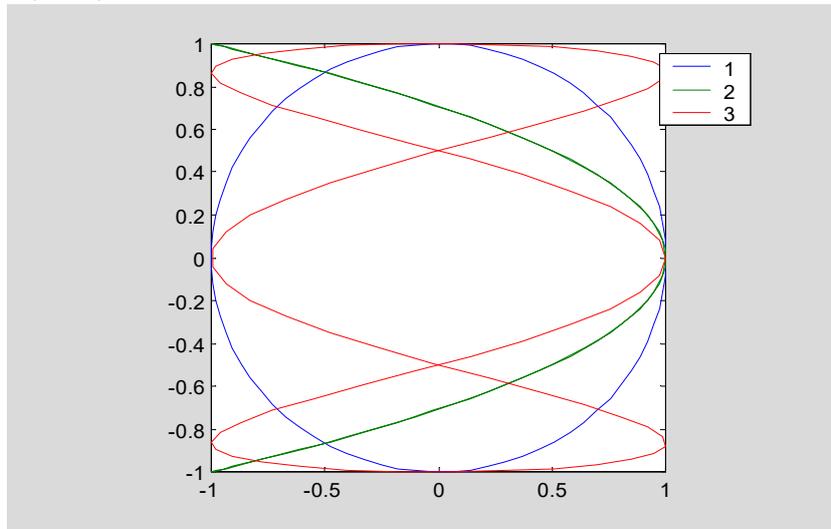


图 7.2.1-3 Lissajous 图

【*例 7.2.1-4】采用模型 $\frac{x^2}{a^2} + \frac{y^2}{25-a^2} = 1$ 画一组椭圆。

```
th = [0:pi/50:2*pi]'; %长度为101的列向量
a = [0.5:.5:4.5]; %长度为9的行向量
X = cos(th)*a; % (101x9) 的矩阵
Y = sin(th)*sqrt(25-a.^2); % (101x9) 的矩阵
plot(X,Y),axis('equal'),xlabel('x'), ylabel('y')
title('A set of Ellipses')
```

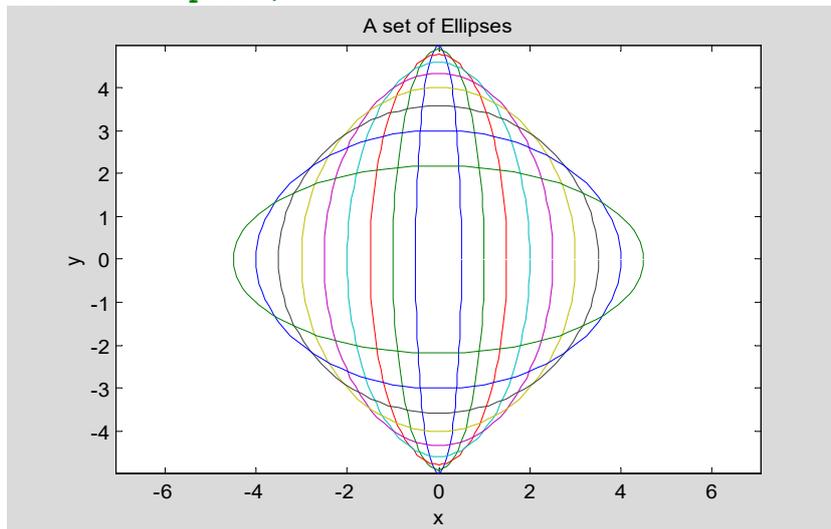


图 7.2.1-4 一组椭圆

7.2.2 曲线的色彩、线型和数据点形

7.2.2.1 色彩和线型

7.2.2.2 数据点形

【*例 7.2.2.2-1】用图形演示平面上一个方块四个顶点在仿射投影（Affine Projection）下的位置、形状变化。

```
%平面上的四个点和它们构成的方块
p1=[-0.5,0,1]';p2=[-0.5,1,1]';p3=[0.5,1,1]';p4=[0.5,0,1]';
Sq=[p1,p2,p3,p4,p1];
%平移投影：沿 x 轴移动 0.5 ，沿 y 轴移动 1 。
dx=0.5;dy=1;T=[1,0,dx;0,1,dy;0,0,1];
%旋转投影：逆时针旋转 30 度。
th=pi/6;R=[cos(th),-sin(th),0;sin(th),cos(th),0;0,0,1];
%刻度投影：x 方向放大到 2 倍，y 方向放大到 3 倍。
alpha=2;beta=3;S=[alpha,0,0;0,beta,0;0,0,1];
E=eye(3,3);%为编程方便，设计一个单位阵。在它投影下，任何形状都不变。 <10>
TRS={E,T,R,S}; %用元胞数组存放四个变换矩阵 <11>
ss={'r^','rd','rp','rh'}; %用元胞数组存放数据点形设置值 <12>
tt={'Original Square','Translation','Rotation','Scaling'}; %用元胞数组存放四张子图的图名 <13>

for i=1:4
    W=TRS{i}*Sq; %进行仿射变换
    subplot(2,2,i)
    for k=1:4
        plot(W(1,k),W(2,k),ss{k}); %用不同点形标志四个顶点 <19>
        axis([-3,3,-1,5]),axis equal
        hold on %使以后图形画在当前子图上 <21>
    end
    plot(W(1,:),W(2,:)) %连接四个顶点使图形封闭 <23>
    grid on %画过坐标格线
    title(tt{i}) %给子图题写图名
    hold off %使以后图形不再画在当前子图上
end
```

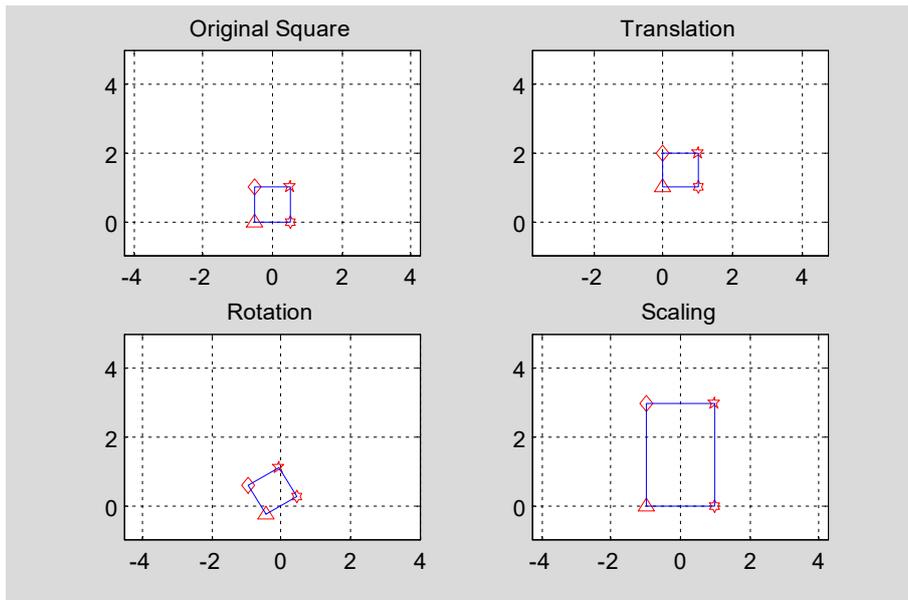


图 7.2.2.2-1 仿射投影演示

7.2.3 坐标、刻度和分格线控制

7.2.3.1 坐标控制

【*例 7.2.3.1-1】观察各种轴控制指令的影响。演示采用长轴为 3.25，短轴为 1.15 的椭圆。注意：采用多子图表现时，图形形状不仅受“控制指令”影响，而且受整个图面“宽高比”及“子图数目”的影响。本书这样处理，是出于篇幅考虑。读者欲想准确体会控制指令的影响，请在全图状态下进行观察。

```
t=0:2*pi/99:2*pi;  
x=1.15*cos(t);y=3.25*sin(t); %y 为长轴, x 为短轴  
subplot(2,3,1),plot(x,y),axis normal,grid on,  
title('Normal and Grid on')  
subplot(2,3,2),plot(x,y),axis equal,grid on,title('Equal')  
subplot(2,3,3),plot(x,y),axis square,grid on,title('Square')  
subplot(2,3,4),plot(x,y),axis image,box off,title('Image and Box off')  
subplot(2,3,5),plot(x,y),axis image fill,box off  
title('Image and Fill')  
subplot(2,3,6),plot(x,y),axis tight,box off,title('Tight')
```

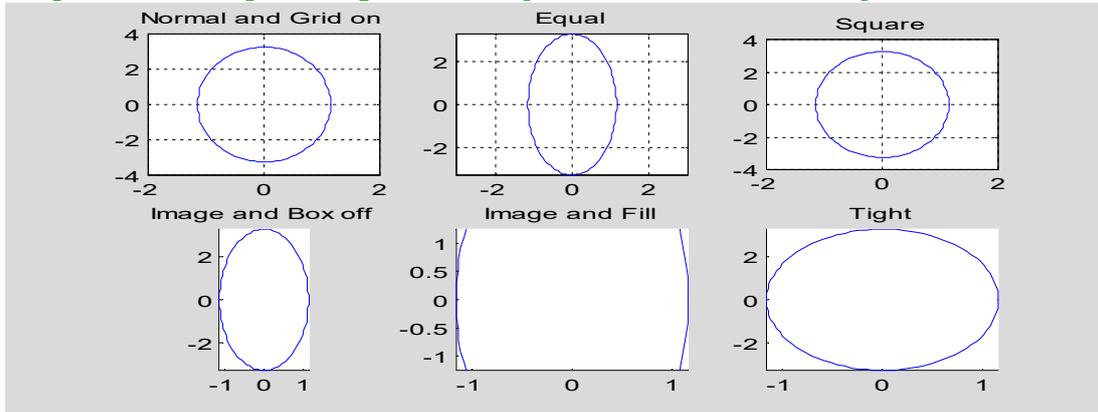


图 7.2.3.1-1 各种轴控制指令的不同影响

7.2.3.2 刻度、分格线和坐标框

【*例 7.2.3.2-1】通过绘制二阶系统阶跃响应，演示 MATLAB 新旧版指令在标识图形上的差别。本例比较综合，涉及的指令知识较广。假如读者能耐心读一下指令、实践操作一遍、再看一下例后的说明，定会有匪浅的收益。

```
clf;t=6*pi*(0:100)/100;y=1-exp(-0.3*t).*cos(0.7*t);  
tt=t(find(abs(y-1)>0.05));ts=max(tt); %<2>  
subplot(1,2,1),plot(t,y,'r-','LineWidth',3),grid on %<3>  
axis([0,6*pi,0.6,max(y)]) %<4>  
title('y=1-exp(-alpha*t)*cos(omega*t)') %<5>  
text(11,1.25,'alpha=0.3');text(11,1.15,'omega=0.7')  
hold on;plot(ts,0.95,'bo','MarkerSize',10);hold off %<7>  
text(ts+1.5,0.95,['ts=' num2str(ts)])  
xlabel('t -->'),ylabel('y -->') %<9>  
subplot(1,2,2),plot(t,y,'r-','LineWidth',3) %<10>  
axis([-inf,6*pi,0.6,inf]) %<11>  
set(gca,'Xtick',[2*pi,4*pi,6*pi],'Ytick',[0.95,1,1.05,max(y)]) %<12>  
grid on %<13>  
title('\it y = 1 - e^{ -\alphanat}cos{\omegat}') %<14>
```

```

text(13.5,1.2,'\fontsize{12}{\alpha}=0.3') %<15>
text(13.5,1.1,'\fontsize{12}{\omega}=0.7') %<16>
hold on;plot(ts,0.95,'bo','MarkerSize',10);hold off %<17>
cell_string{1}='\fontsize{12}\uparrow'; %<18>
cell_string{2}='\fontsize{16} \fontname{隶书}镇定时间'; %<19>
cell_string{3}='\fontsize{6} ' ; %<20>
cell_string{4}=['\fontsize{14}\rmt_{s} = ' num2str(ts)]; %<21>
text(ts,0.85,cell_string) %<22>
xlabel('\fontsize{14} \bft \rightarrow') %<23>
ylabel('\fontsize{14} \bfy \rightarrow') %<24>

```

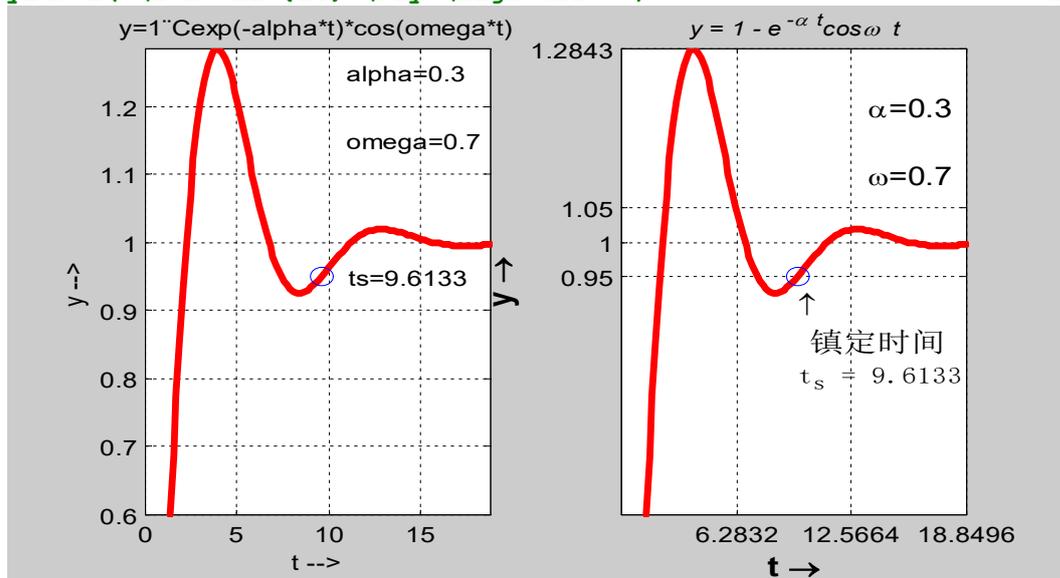


图 7.2.3.2-1 二阶阶跃响应图用 MATLAB4.x 和 5.x 版标识时的差别

7.2.4 图形标识

7.2.4.1 简捷指令形式

7.2.4.2 精细指令形式

【*例 7.2.4.2-1】本例非常简单，专供试验标识用。

```

clf;t=0:pi/50:2*pi;y=sin(t);plot(t,y);axis([0,2*pi,-1.2,1.2])
text(pi/2,1,'\fontsize{16}\leftarrow\itsin(t)\fontname{隶书}极大值')

```

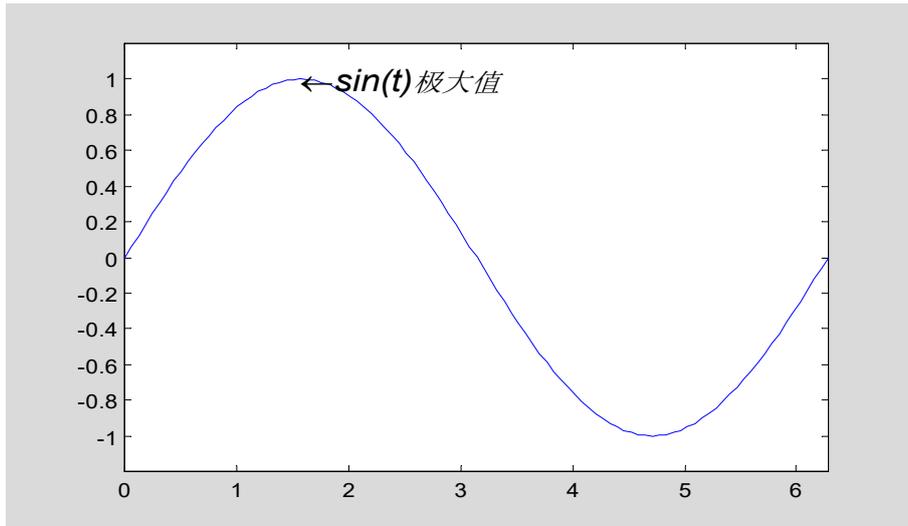


图 7.2.4.2-1 试验标识的图形

7.2.5 多次叠绘、双纵坐标和多子图

7.2.5.1 多次叠绘

【*例 7.2.5.1-1】利用 hold 绘制离散信号通过零阶保持器后产生的波形。

```
t=2*pi*(0:20)/20;y=cos(t).*exp(-0.4*t);
stem(t,y,'g');hold on;stairs(t,y,'r');hold off
```

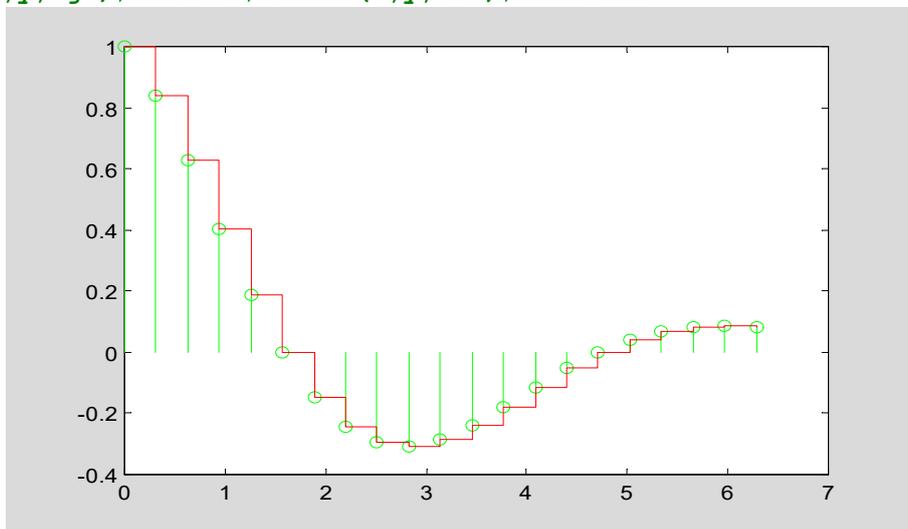


图 7.2.5.1-1 离散信号的重构

7.2.5.2 双纵坐标图

【*例 7.2.5.2-1】画出函数 $y = x \sin x$ 和积分 $s = \int_0^x (x \sin x) dx$ 在区间 $[0,4]$ 上的曲线。

```
clf;dx=0.1;x=0:dx:4;y=x.*sin(x);s=cumtrapz(y)*dx;%梯形法求累计积分
plotyy(x,y,x,s),text(0.5,0,'\fontsize{14}\ity=xsinx')
```

```
sint='\fontsize{16}\int_{\fontsize{8}0}^{\fontsize{8}x}';
text(2.5,3.5,['\fontsize{14}\its=',sint,'\fontsize{14}\itxsinxdx'])
```

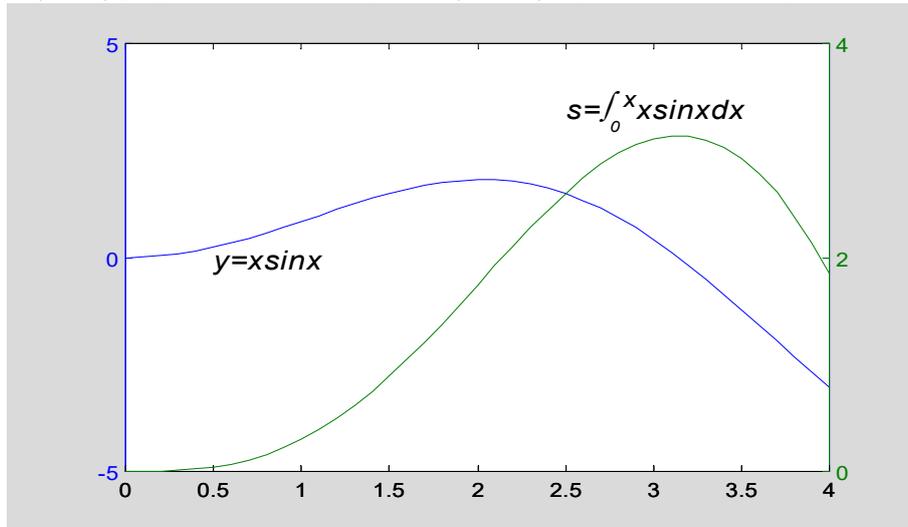


图 7.2.5.2-1 函数和积分

【*例 7.2.5.2-2】受热压力容器的期望温度是 120 度，期望压力是 0.25Mpa。在同一张图上画出它们的阶跃响应曲线。

```
S1=tf([1 1],[1 3 2 1]); %温度的传递函数对象模型。
S2=tf(1,[1 1 1]); %压力的传递函数对象模型。
[Y1,T1]=step(S1); %计算阶跃响应
[Y2,T2]=step(S2); %计算阶跃响应
plotyy(T1,120*Y1,T2,0.25*Y2,'stairs','plot')
```

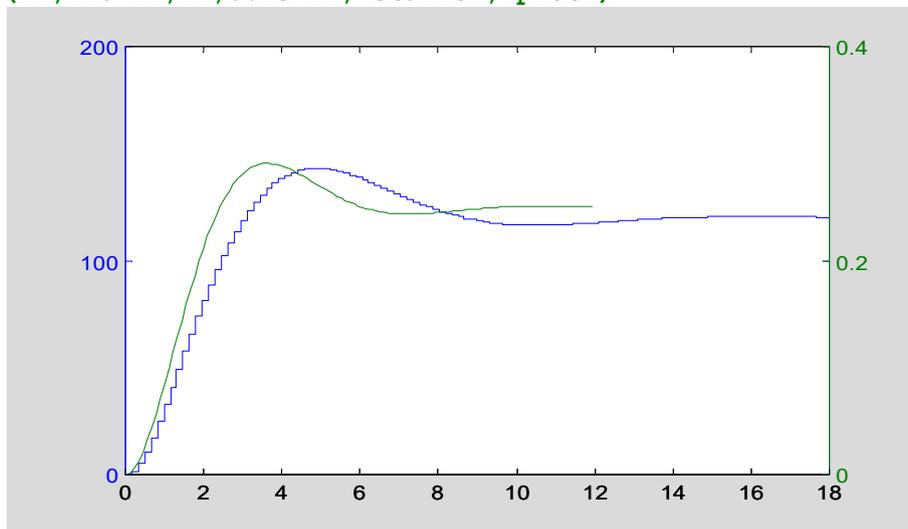


图 7.2.5.2-2 双纵坐标图演示

7.2.5.3 多子图

【*例 7.2.5.3-1】演示 subplot 指令对图形窗的分割。

```
clf;t=(pi*(0:1000)/1000)';
y1=sin(t);y2=sin(10*t);y12=sin(t).*sin(10*t);
subplot(2,2,1),plot(t,y1);axis([0,pi,-1,1])
```

```
subplot(2,2,2),plot(t,y2);axis([0,pi,-1,1])
subplot('position',[0.2,0.05,0.6,0.45])
plot(t,y12,'b-',t,[y1,-y1],'r:');axis([0,pi,-1,1])
```

%<5>

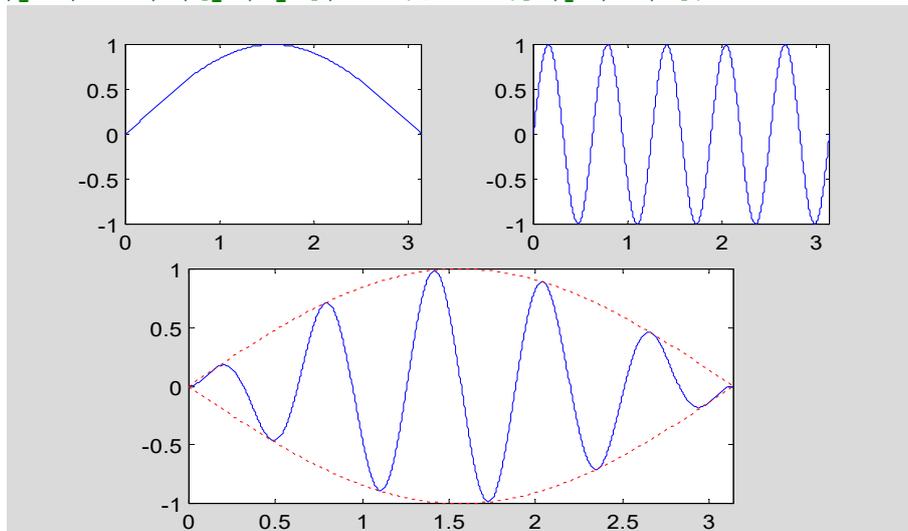


图 7.2.5.3-1 多子图的布置

7.2.6 交互式图形指令

7.2.6.1 ginput

7.2.6.2 gtext

7.2.6.3 legend

7.2.6.4 zoom

7.3 三维绘图的基本操作

7.3.1 三维线图指令 plot3

【*例 7.3.1-1】简单例题。

```
t=(0:0.02:2)*pi;x=sin(t);y=cos(t);z=cos(2*t);
plot3(x,y,z,'b-',x,y,z,'bd'),view([-82,58]),box on,legend('链','宝石')
```

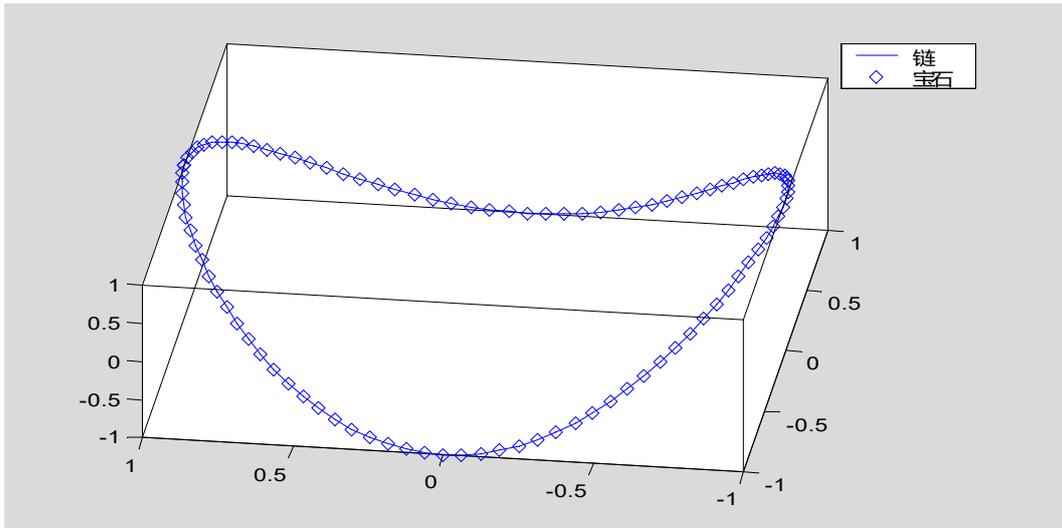


图 7.3.1-1 宝石项链

7.3.2 三维网线图和曲面图

7.3.2.1 三维图形的数据准备

- (1) 确定自变量 x, y 的取值范围和取值间隔。
- (2) 构成 $x-y$ 平面上的自变量采样“格点”矩阵。
- (3) 计算在自变量采样“格点”上的函数值，即 $Z=f(X,Y)$ 。

7.3.2.2 网线图、曲面图基本指令格式

【*例 7.3.2.2-1】用曲面图表现函数 $z = x^2 + y^2$ 。

```

clf,x=-4:4;y=x:[X,Y]=meshgrid(x,y); %生成 x-y 坐标“格点”矩阵
Z=X.^2+Y.^2; %计算格点上的函数值
surf(X,Y,Z);hold on,colormap(hot)
stem3(X,Y,Z,'bo') %用来表现在格点上计算函数值

```

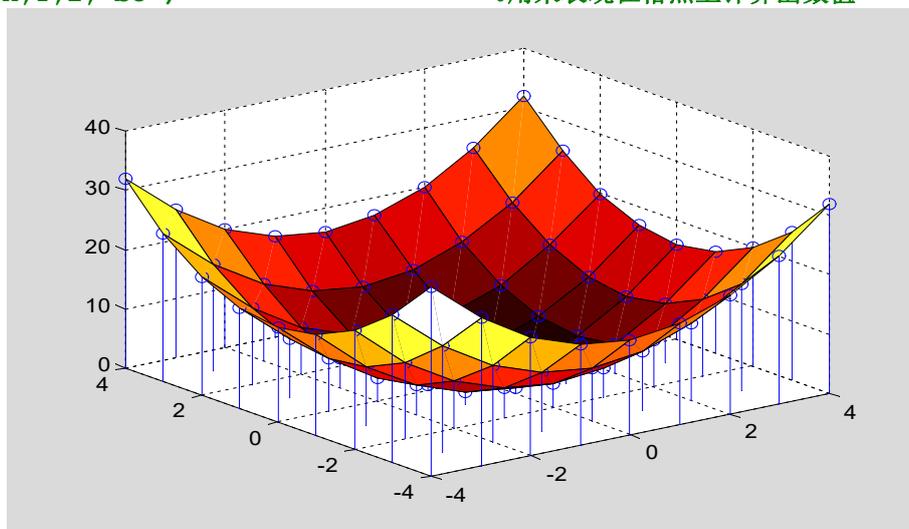


图 7.3.2.2-1 曲面图和格点

7.3.3 透视、镂空和裁切

7.3.3.1 图形的透视

【*例 7.3.3.1-1】透视演示

```
[X0,Y0,Z0]=sphere(30);           %产生单位球面的三维坐标
X=2*X0;Y=2*Y0;Z=2*Z0;           %产生半径为2的球面的三维坐标
clf,surf(X0,Y0,Z0);              %画单位球面
shading interp                    %采用插补明暗处理
hold on,mesh(X,Y,Z),colormap(hot),hold off %采用 hot 色图
hidden off                        %产生透视效果
axis equal,axis off               %不显示坐标轴
```

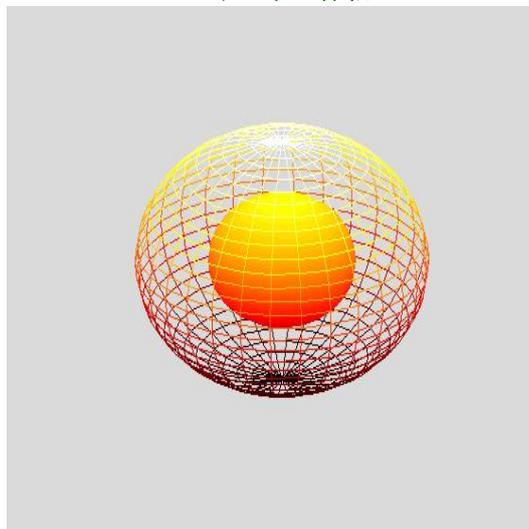


图 7.3.3.1-1 剔透玲珑球

7.3.3.2 图形的镂空

【*例 7.3.3.2-1】演示：如何利用“非数”NaN，对图形进行剪切处理。

```
clf;
t=linspace(0,2*pi,100);r=1-exp(-t/2).*cos(4*t); %旋转母线
[X,Y,Z]=cylinder(r,60); %产生旋转柱面数据
ii=find(X<0&Y<0); %确定 x-y 平面第四象限上的数据下标
Z(ii)=NaN; %剪切
surf(X,Y,Z);colormap(spring),shading interp
light('position',[-3,-1,3],'style','local') %设置光源
material([0.5,0.4,0.3,10,0.3]) %设置表面反射
```

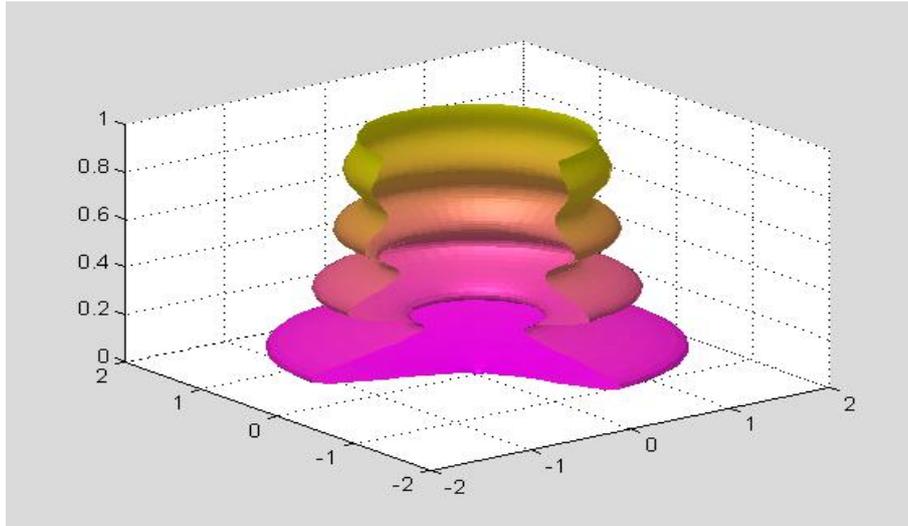


图 7.3.3.2-1 剪切四分之一后的图形

【*例 7.3.3.2-2】演示：如何利用“非数”NaN，对图形进行镂空处理。

```
P=peaks(30);P(18:20,9:15)=NaN;      %镂空
surf(P);colormap(summer)
light('position',[50,-10,5]),lighting flat
material([0.9,0.9,0.6,15,0.4])
```

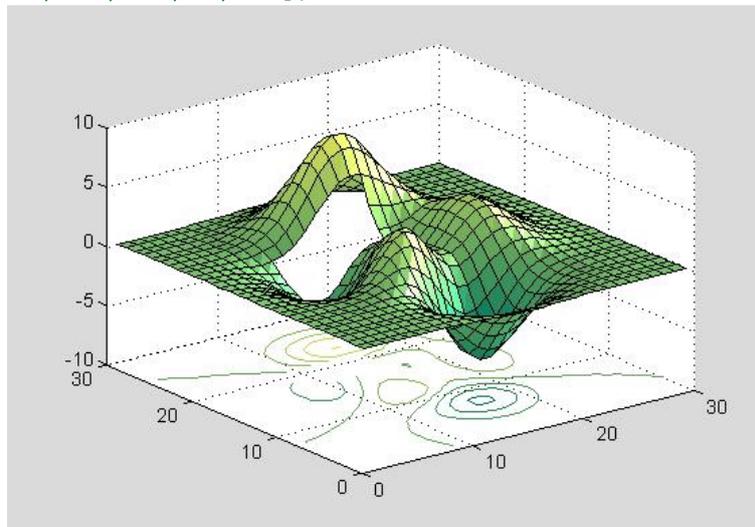


图 7.3.3.2-2 镂空方孔的曲面

7.3.3.3 裁切

【*例 7.3.3.3-1】表现切面

```
clf,x=[-8:0.2:8];y=x;[X,Y]=meshgrid(x,y);ZZ=X.^2-Y.^2;
ii=find(abs(X)>6|abs(Y)>6); %确定超出[-6,6]范围的格点下标
ZZ(ii)=zeros(size(ii)); %强制为0
surf(X,Y,ZZ),shading interp;colormap(copper)
light('position',[0,-15,1]);lighting phong
material([0.8,0.8,0.5,10,0.5])
```

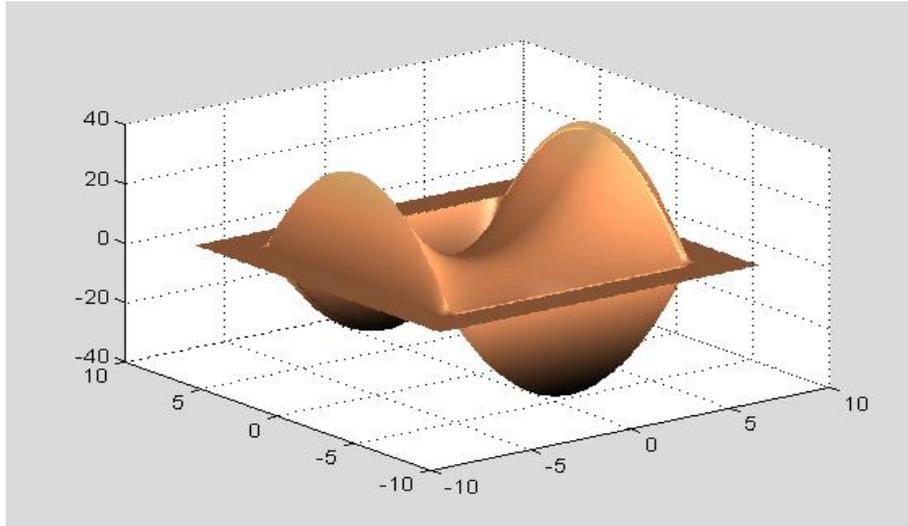


图 7.3.3.3-1 经裁切处理后的图形

7.4 特殊图形和高维可视化

7.4.1 特殊图形指令例示

7.4.1.1 面域图 area

【*例 7.4.1.1-1】面域图指令 **area**。该指令的特点是：在图上绘制多条曲线时，每条曲线（除第一条外）都是把“前”条曲线作基线，再取值绘制而成。因此，该指令所画的图形，能醒目地反映各因素对最终结果的贡献份额。注意：（1）**area** 的第一输入宗量是单调变化的自变量。第二输入宗量是“各因素”的函数值矩阵，且每个“因素”的数据取列向量形式排放。第三输入宗量是绘图的基准线值，只能取标量。当基准值为 0（即以 x 轴为基准线）时，第三输入宗量可以缺省。（2）本例第<4>条指令书写格式 **x', Y'**，强调沿列方向画各条曲线的事实。

```

clf;x=-2:2                                %注意：自变量要单调变化
Y=[3,5,2,4,1;3,4,5,2,1;5,4,3,2,5] %各因素的相对贡献份额
Cum_Sum=cumsum(Y)                          %各曲线在图上的绝对坐标
area(x',Y',0)                               %<4>
legend('因素 A','因素 B','因素 C'),grid on,colormap(spring)
x =
    -2    -1     0     1     2
Y =
     3     5     2     4     1
     3     4     5     2     1
     5     4     3     2     5
Cum_Sum =
     3     5     2     4     1
     6     9     7     6     2
    11    13    10     8     7

```

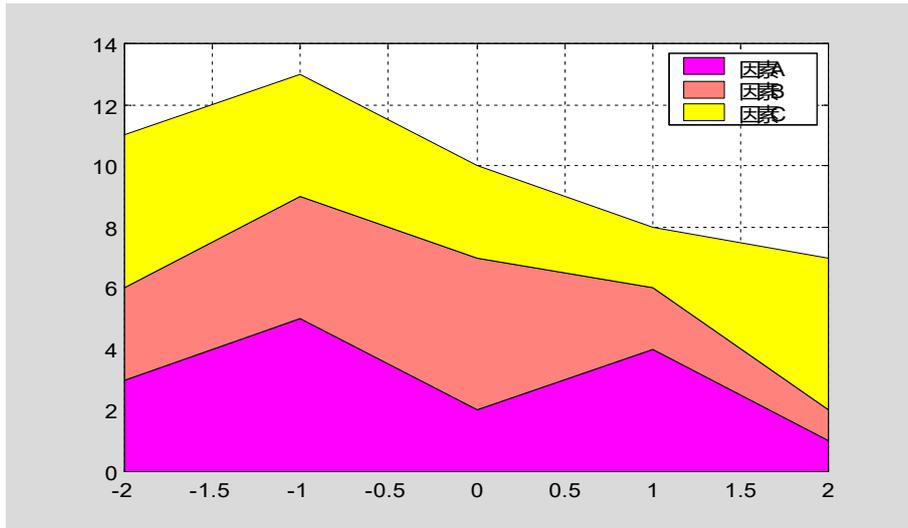


图 7.4.1.1-1 面域图表现各分量的贡献

7.4.1.2 各种直方图 bar, barh, bar3, bar3h

【*例 7.4.1.2-1】二维直方图有两种图型：垂直直方图和水平直方图。而每种图型又有两种表现模式：累计式：分组式。本例选其两种加以表现。

```

x=-2:2; %注意：自变量要单调变化
Y=[3,5,2,4,1;3,4,5,2,1;5,4,3,2,5]; %各因素的相对贡献份额
subplot(1,2,1),bar(x',Y','stacked') %“累计式”直方图
xlabel('x'),ylabel('\Sigma y'),colormap(cool) %控制直方图的用色
legend('因素A','因素B','因素C')
subplot(1,2,2),barh(x',Y','grouped') %“分组式”水平直方图
xlabel('y'),ylabel('x')

```

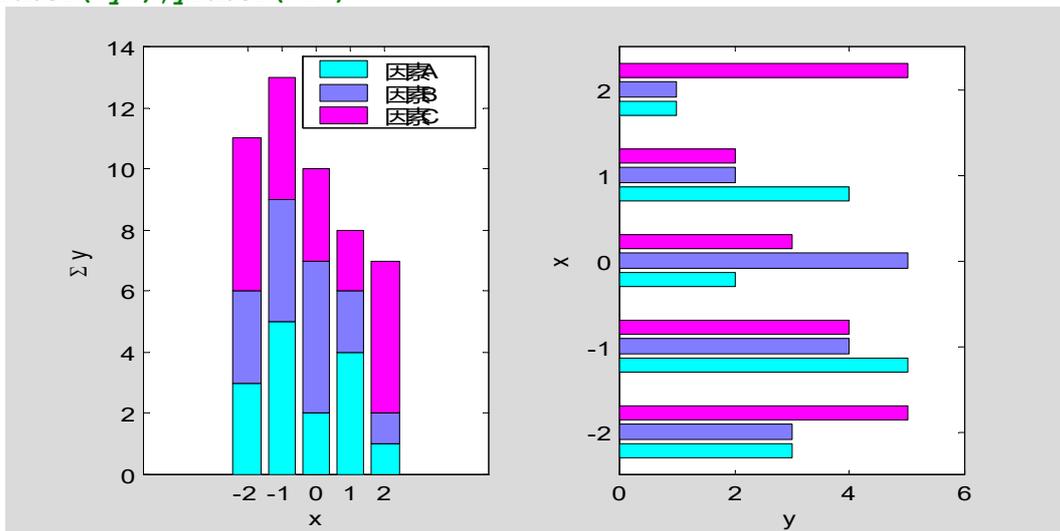


图 7.4.1.2-1 二维直方图

【*例 7.4.1.2-2】用三维直方图表现上例数据。

```

clf;x=-2:2; %注意：自变量要单调变化
Y=[3,5,2,4,1;3,4,5,2,1;5,4,3,2,5]; %各因素的相对贡献份额

```

```

subplot(1,2,1),bar3(x',Y',1)           % “队列式” 直方图
xlabel('因素ABC'),ylabel('x'),zlabel('y')
colormap(summer)                       %控制直方图的用色
subplot(1,2,2),bar3h(x',Y','grouped') % “分组式” 水平直方图
ylabel('y'),zlabel('x')

```

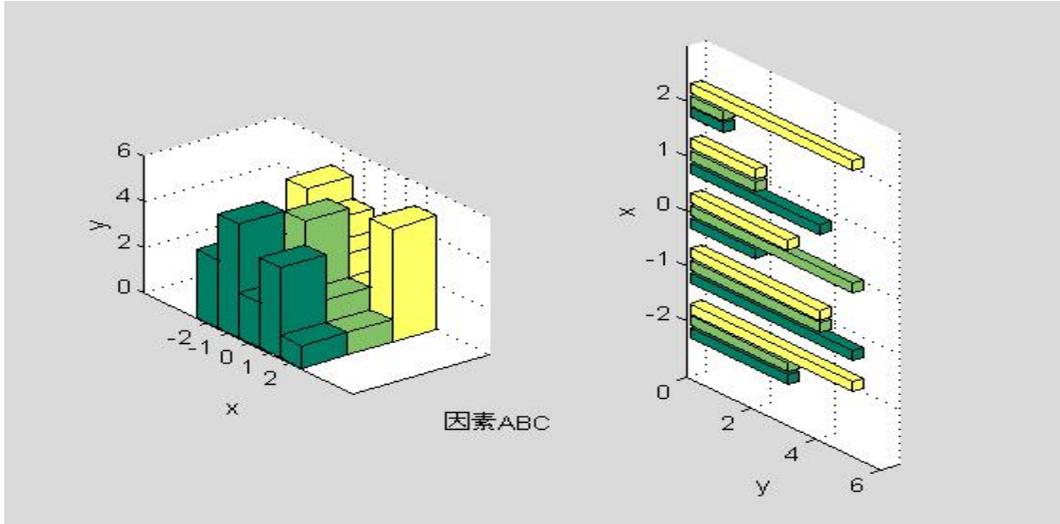


图 7.4.1.2-2 三维直方图

7.4.1.3 饼图 pie, pie3

【*例 7.4.1.3-1】饼图指令 **pie**, **pie3** 用来表示各元素占总和的百分数。该指令第二输入宗量为与第一宗量同长的 0-1 向量，1 使对应扇块突出。

```

a=[1,1.6,1.2,0.8,2.1];
subplot(1,2,1),pie(a,[1 0 1 0 0]),legend({'1','2','3','4','5'})
subplot(1,2,2),pie3(a,a==min(a)),colormap(cool)

```

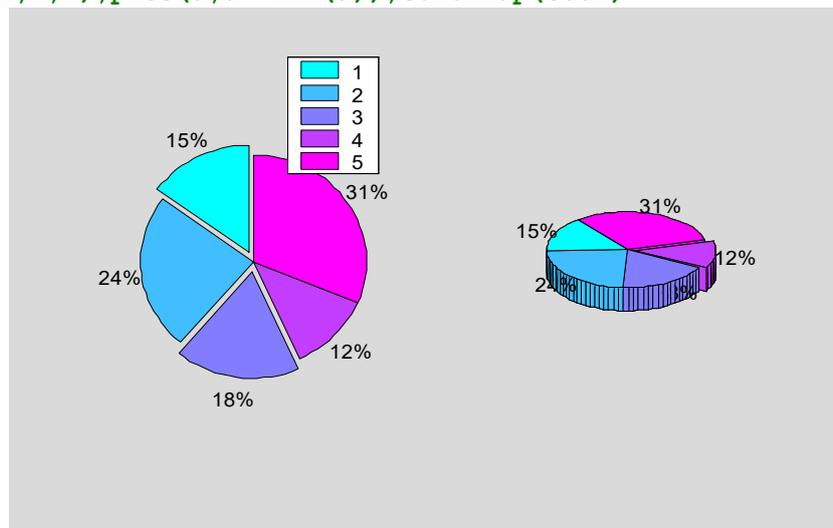


图 7.4.1.3-1 饼形统计图

7.4.1.4 填色图 fill,fill3

【*例 7.4.1.4-1】读者试验本例时，注意三点：MATLAB 画任意多边形的一种方法；保证绘图数据首尾重合，使勾画多边形封闭；使用图柄对图形的属性进行精细设置。

```

clf;n=10; %多边形的边数
dt=2*pi/n;t=0:dt:2*pi
t=[t,t(1)]; %fill 指令要求数据向量的首位重合，使图形封闭。
x=sin(t);y=cos(t);
fill(x,y,'c');axis off %画填色多边形，隐去坐标轴。
ht=text(0,0,'\fontname{隶书}\fontsize{32}十边形');%文字注释，且得图柄。
set(ht,'Color','k','HorizontalAlignment','Center') %依靠图柄设置属性。

```

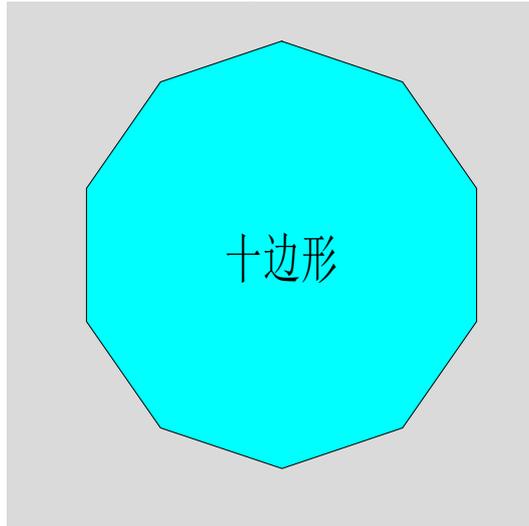


图 7.4.1.4-1 由 fill 产生的填色多边形

【例 7.4.1.4-2】三维填色指令 fill3 演示。注意：（1）X,Y,Z 的相应列元素构成一个三维封闭多边形。本例有 4 列，因此有 4 个多边形。图 7.4.1.4-2 中的“1, 2, 3, 4”号三角形分别由 X,Y,Z 的第 1, 2, 3, 4 列生成。（2）为使多边形封闭，每列的首尾元素应该重合。若不重合，则将默认把最后一点与第一点相连，强行使多边形封闭。（3）该指令的第 4 输入宗量可取定色单字符（如'r','g'等），也可取与 X 同维的数值矩阵。（4）所填色彩受 C 和色图的双重响应。（5）本例图中三角形的编号是通过“图形窗”编辑而生成的。

```

X=[0.5 0.5 0.5 0.5;0.5 0.5 0.5 0.5;0 1 1 0];
Y=[0.5 0.5 0.5 0.5;0.5 0.5 0.5 0.5;0 0 1 1];
Z=[1 1 1 1;0 0 0 0;0 0 0 0];C=[1 0 0 1;0 1 0 1;0 0 1 0];
fill3(X,Y,Z,C),view([-10 55]),colormap cool
xlabel('x'),ylabel('y'),box on;grid on

```

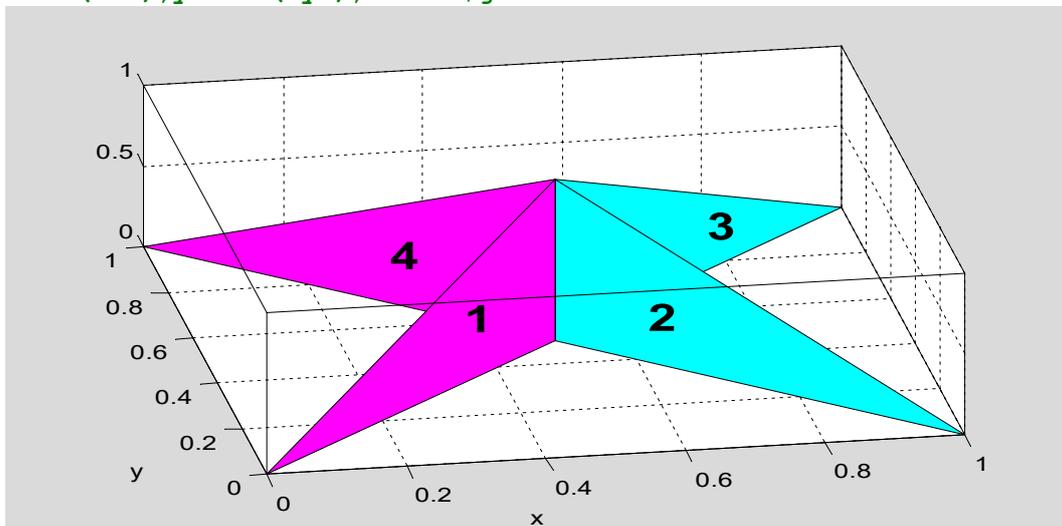


图 7.4.1.4-2 三维填色

7.4.1.5 射线图 compass 和羽毛图 feather

【*例 7.4.1.5-1】 compass 和 feather 指令的区别。

```
t=-pi/2:pi/12:pi/2;    %在[-90°,90°]区间, 每15°取一点。  
r=ones(size(t));      %单位半径  
[x,y]=pol2cart(t,r);  %极坐标转化为直角坐标  
subplot(1,2,1),compass(x,y),title('Compass')  
subplot(1,2,2),feather(x,y),title('Feather')
```

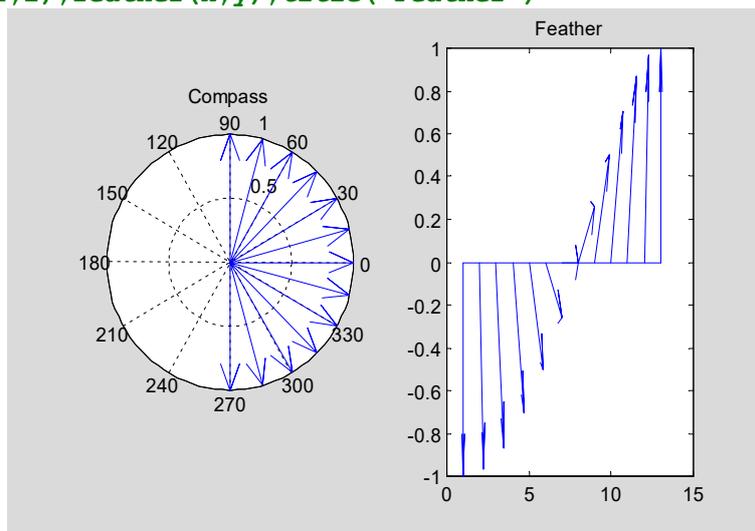


图 7.4.1.5-1 compass 和 feather 指令的区别

7.4.1.6 Voronoi 图和三角剖分

【*例 7.4.1.6-1】用 Voronoi 多边形勾画每个点的最近邻范围。Voronoi 多边形在计算几何、模式识别中有重要应用。从本例图 7.4.1.6-1 中, 可以看到, 三角形顶点所在多边形的三条公共边是剖分三角形边的垂直平分线。

```
clf;rand('state',111)  
n=30;A=rand(n,1)-0.5;B=rand(n,1)-0.5;    %产生 30 个随机点  
T=delaunay(A,B);                          %求相邻三点组  
T=[T T(:,1)];                             %为使三点剖分三角形封闭而采取的措施  
voronoi(A,B)                              %画 Voronoi 图  
hold on;axis square  
fill(A(T(10,:),:),B(T(10,:)), 'y');      %画一个剖分三角形  
voronoi(A,B)                              %重画 Voronoi 图, 避免线被覆盖。
```

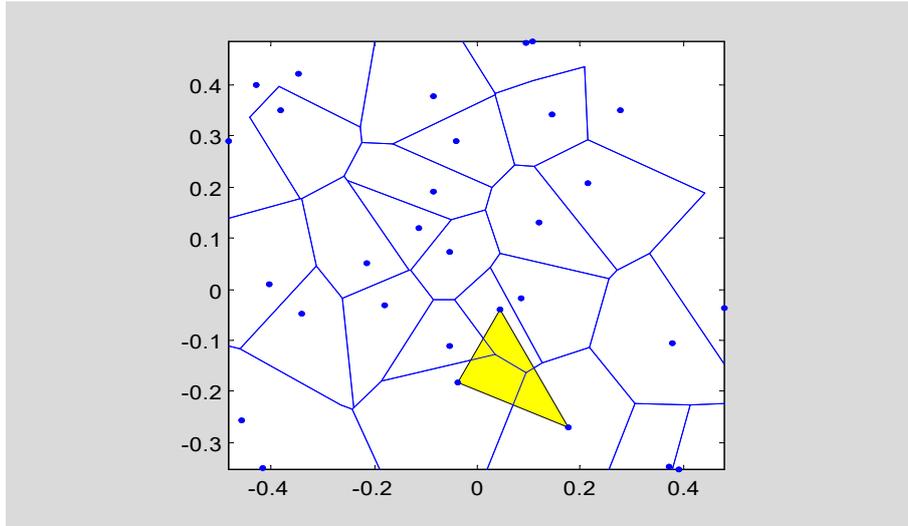


图 7.4.1.6-1 Voronoi 多边形和 Delaunay 三角剖分

7.4.1.7 彩带图 ribbon

【*例 7.4.1.7-1】用彩带绘图指令 **ribbon**，绘制归化二阶系统 $G = \frac{1}{s^2 + 2\zeta s + 1}$ 在不同 ζ 值

时的阶跃响应，如图 7.4.1.7-1 所示。对于本例程序，有以下几点值得注意：（1）程序中使用了 Control Toolbox 中的两个指令 **tf** 和 **step**。这 **tf** 是一个（MATLAB5.x 版起用的）“对象”。（2）本例构造的 S 是一个单输入 8 输出系统，作用于该 S 的 **step** 指令也将在一次调用中产生 8 个子系统的阶跃响应。（3）在下段程序运行后，有兴趣的读者可显示 S，以观察系统是如何描述的。（4）本例为了得到较好的表现效果，采用了视角、明暗、色图、光照控制。（5）为使程序有一定通用性，图例采用元胞数组生成。（6）本例产生的图 7.4.1.7-1 中，除“ $t \rightarrow$ ”外，所有标识都是由下段指令产生的。（7）“ $t \rightarrow$ ”中的斜向箭头无法由指令生成，而是直接通过“图形窗”编辑实现的（MATLAB5.3 版起用）。（8）本例程序有通用性。只要修改第<2>条指令对阻尼系数的设定，就可获得响应的彩带图形。

```
clear,clf
zeta2=[0.1 0.2 0.3 0.4 0.5 0.6 0.8 1.0]; %<2>
n=length(zeta2);for k=1:n;Num{k,1}=1;Den{k,1}=[1 2*zeta2(k) 1];end
S=tf(Num,Den); %产生单输入多输出系统
t=(0:0.4:30)'; %时间采样点
[Y,x]=step(S,t); %单输入多输出系统的响应
tt=t*ones(size(zeta2)); %为画彩带图，生成与函数值 y 维数相同的时间矩阵。
ribbon(tt,Y,0.4) %画彩带图
%至此彩带图已经生成。以下指令都是为了使图形效果更好、标识更清楚而用。
view([150,50]),shading interp,colormap(jet)%设置视角、明暗、色图
light,lighting phong,box on %设置光源、照射模式、坐标框
for k=1:n;str_lgd{k,1}=num2str(zeta2(k));end,legend(str_lgd)%图例设置
str1='\itG = (s^{2} + 2\zetas + 1)^{-1}';
str2='\fontsize{16}\fontname{隶书}取不同';
str3='\fontsize{10}\it\zeta';
str4='\fontsize{16}\fontname{隶书}时的阶跃响应';
title([str1,str2,str3,str4]),zlabel('\ity(\zeta,t) \rightarrow')
```

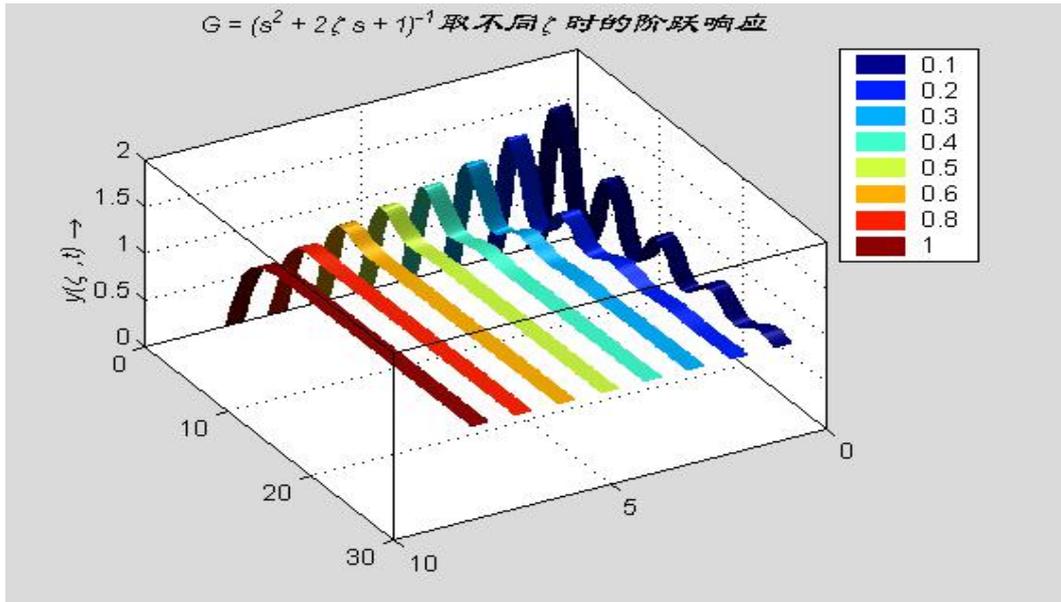


图 7.4.1.7-1 二阶系统在不同阻尼系数时的响应

7.4.1.8 离散杆图 stem, stem3

【*例 7.4.1.8-1】本例表现一个离散方波的快速 Fourier 变换的幅频。本例左图用极坐标指令 **polar** 绘出，右图用三维离散杆图指令 **stem3** 绘出。

MATLAB 的 **polar** 指令是建筑在 **plot** 基础上的。指令执行后，出现的极坐标轴及分度标识也是由 **plot** 以一种“固定”模式产生的。因此，极坐标轴的控制很不灵活，它只能以比较简单的方式表达函数。如对于本例左图，那图形小、线条细、文字太密的缺陷，就较难克服。相比而言，先借助极坐标和直角坐标转换，然后再通过直角坐标图形指令加以表现，往往更显灵活、方便。如本例的右图。

```

th = (0:127)/128*2*pi;           %角度采样点
rho=ones(size(th));              %单位半径
x = cos(th);y = sin(th);
f = abs(fft(ones(10,1),128));    %对离散方波进行FFT变换，并取幅值。
rho=ones(size(th))+f';           %取单位圆为绘制幅频谱的基准。
subplot(1,2,1),polar(th,rho,'r')
subplot(1,2,2),stem3(x,y,f','d','fill') %取菱形离散杆头，并填色。
view([-65 30])                   %控制角度，为表现效果。

```

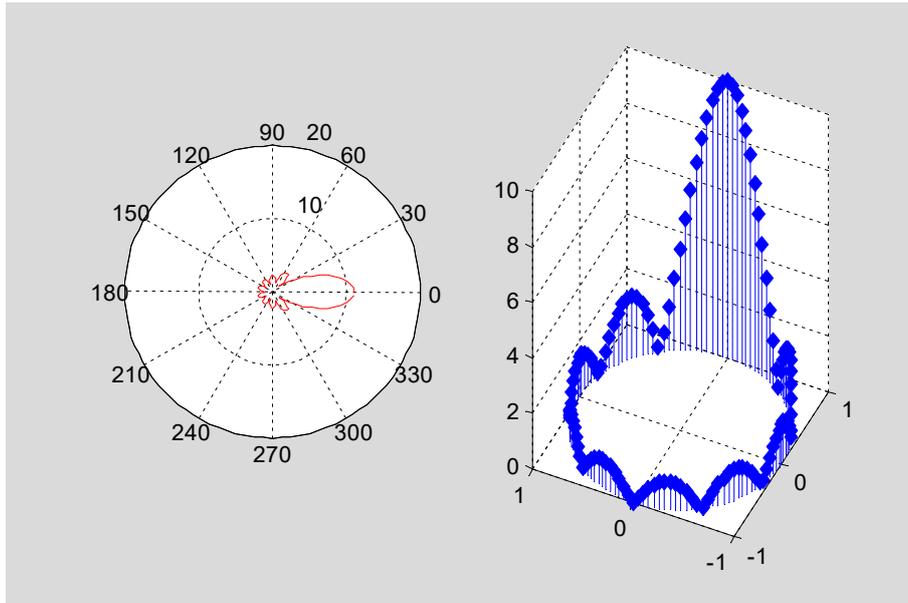


图 7.4.1.8-1 离散方波的幅频谱

7.4.1.9 二维半图指令 `pcolor`, `contour`, `contourf`

【*例 7.4.1.9-1】本例重点演示所谓“二维半”指令：伪彩图 `pcolor`；等位线指令 `contour`、`contourf`；等位线标高指令 `clabel` 的配合使用和区别。练习本例时注意：（1）本例等位线指令中的第 4 输入宗量 `n` 设定高度的等级数，第 5 输入宗量设定等位线的线型、色彩。（2）左右两图的标高方法不同。左图的标识以“+”引导，水平放置。右图沿线布置。这是由 `clabel` 的调用格式不同产生的。（3）左右两图色彩的形成方法不同，色彩效果也不同。（4）在左图中，`colorbar` 画出一根垂直色标尺，而 `caxis` 决定该色标尺的刻度。

```

clf;clear;[X,Y,Z]=peaks(40);      %获得 peaks 图形数据
n=4;                               %等高线分级数
subplot(1,2,1),pcolor(X,Y,Z)      %伪彩图
colormap jet,shading interp
hold on,C=contour(X,Y,Z,n,'k:');  %用黑虚线画等位线，并给出标识数据。
clabel(C)                          %随机标识法
zmax=max(max(Z));zmin=min(min(Z));caxis([zmin,zmax]) %决定色标尺的范围
colorbar                            %画垂直色标尺
hold off,subplot(1,2,2)
[C,h,CF]=contourf(X,Y,Z,n,'k:'); %用黑虚线画填色等位线，并给出标识数据。
clabel(C,h)                        %沿线标识法

```

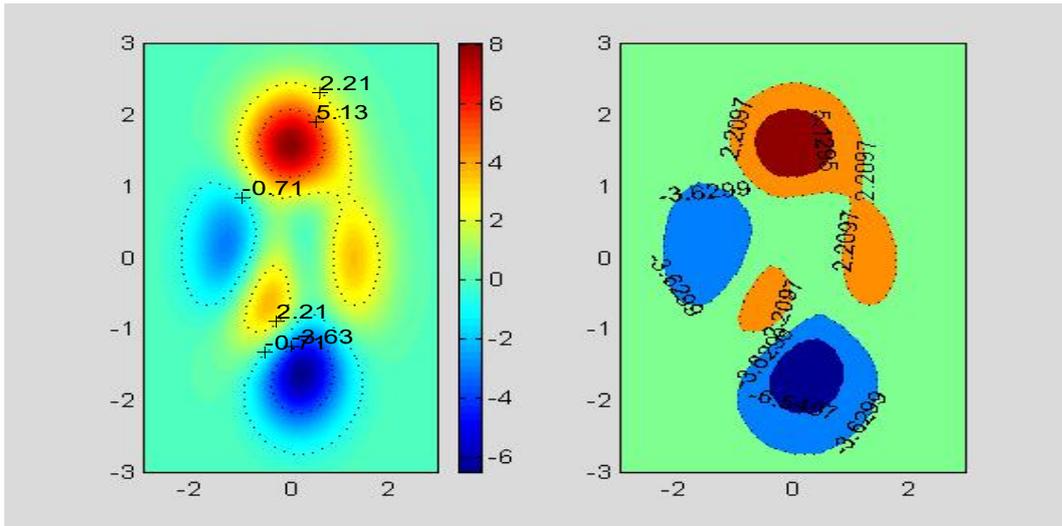


图 7.4.1.9-1 “二维半”指令的演示

7.4.1.10 散点图 scatter, scatter3, plotmatrix

【*例 7.4.1.10-1】表现函数 $z = \frac{\sin R}{R}, R = \sqrt{x^2 + y^2}$ 。请注意本例中的 3 个指令：（1）散点图指令 **scatter3**，标志三维数据点。它的前三个输入宗量必须是同长的向量，如指令<5>。（2）带垂帘的网线图指令 **meshz**，它的调用格式与 **mesh** 没有什么不同。

此外，再次提醒读者注意指令<2>。这样处理的目的是避免 0/0 的不定性。该处理方法是求极限的一种数值方法。

```
x=3*pi*(-1:0.2:1);y=x;[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;Z=sin(R)./R;%引入 eps 避免 0/0, %<2>
C=abs(del2(Z)); %求“五点格式”差分，反映曲面变化。
meshz(X,Y,Z,C) %由曲面变化决定用色
hold on,scatter3(X(:),Y(:),Z(:),'filled') %<5>
hold off,colormap(hot)
```

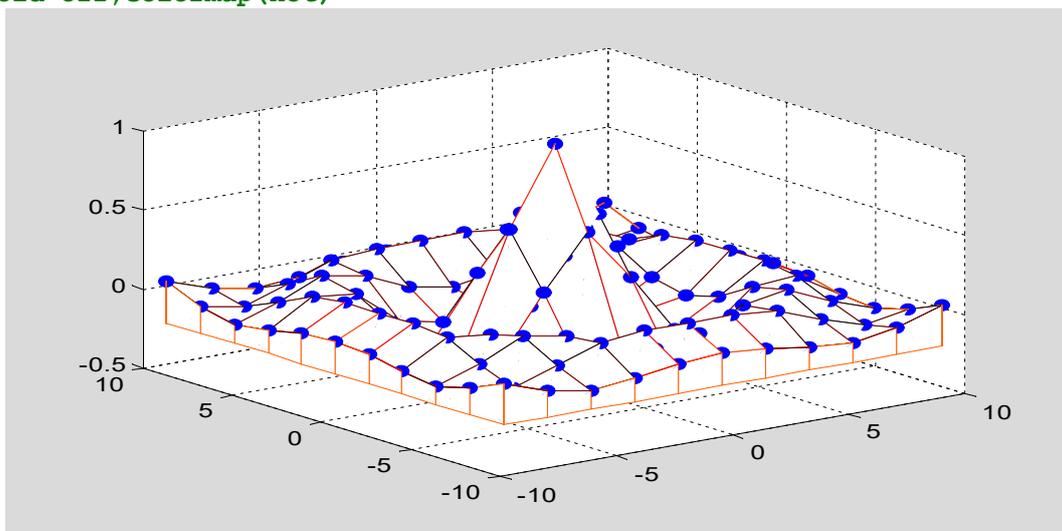


图 7.4.1.10-1 三维散点图

【*例 7.4.1.10-2】指令 **plotmatrix** 有两种基本调用方式：（1）对于数据矩阵 ($p \times n$) 维的 X 和 ($p \times m$) 维的 Y，调用格式 **plotmatrix(X,Y)** 将画出一个分割成 ($m \times n$) 个子散点图。其中第 (i, j) 个子散点图是根据 Y 第 i 列和 X 第 j 列数据画出的。（2）对于数据矩阵 ($p \times n$) 维的 X，调用格式 **plotmatrix(X)** 将画出分割成 ($n \times n$) 个子块的图。该图的对角块，画出的是 X 每列的数据的频数直方图；而其他子块是相应列构成的散点图。

该指令可用于观察数据矩阵（或同一矩阵列向量）间的统计关系。

```
randn('seed',1111),X=randn(100,2);Y=randn(100,2);
subplot(1,3,1),plotmatrix(X)
subplot(1,3,2),plotmatrix(X,X)
subplot(1,3,3),plotmatrix(X,Y)
```

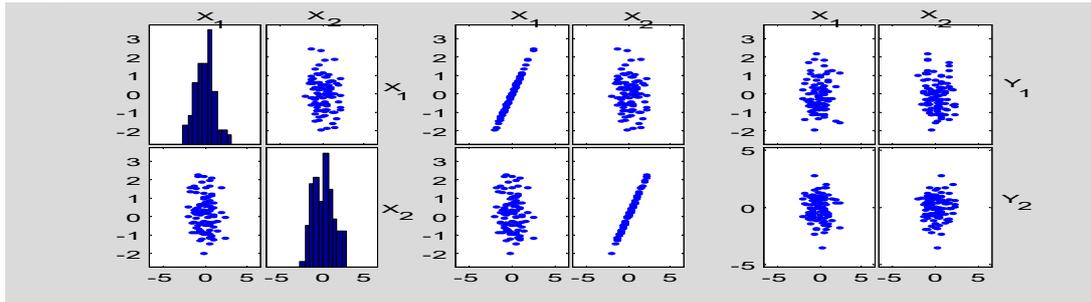


图 7.4.1.10-2 plotmatrix 表现数据统计特性

7.4.1.11 不规则数据的网线图和曲面图

【*例 7.4.1.11-1】用三角网线、曲面图表现函数 $z = \frac{\sin R}{R}, R = \sqrt{x^2 + y^2}$ 。

```
rand('seed',22),X=6*pi*(rand(20,10)-0.5);Y=6*pi*(rand(20,10)-0.5);
R=sqrt(X.^2+Y.^2)+eps;Z=sin(R)./R;
tri=delaunay(X,Y); %进行三角剖分
subplot(1,2,1),trimesh(tri,X,Y,Z)
subplot(1,2,2),trisurf(tri,X,Y,Z)
colormap(jet);brighten(0.5) %增强亮度
```

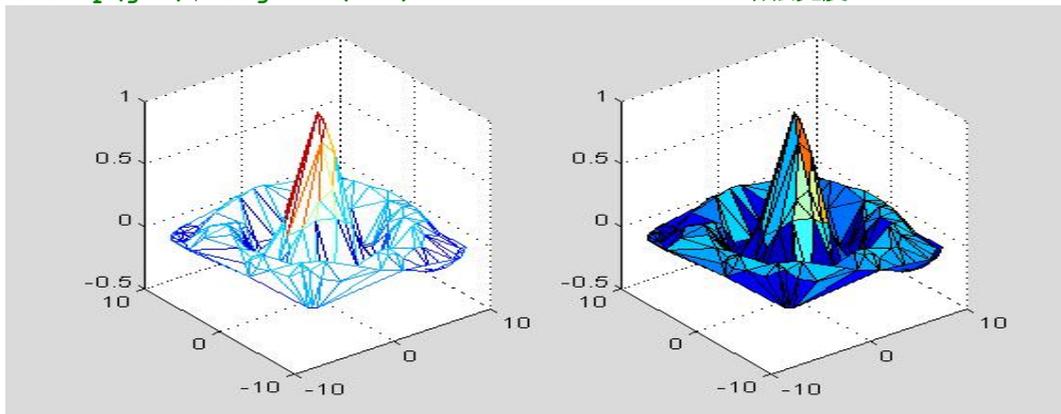


图 7.4.1.11-1 不规则数据的三维表现

7.4.1.12 泛函绘图指令 fplot

【*例 7.4.1.12-1】fplot 与一般绘图指令的绘图效果比较。

```

[x,y]=fplot('cos(tan(pi*x))',[-0.4,1.4],0.2e-3;n=length(x);
subplot(1,2,1),plot(x,y)
title('\fontsize{20}\fontname{隶书}泛函绘图指令效果')
t=(-0.4:1.8/n:1.4)';
subplot(1,2,2),plot(t,cos(tan(pi*t)))
title('\fontsize{20}\fontname{隶书}等分采样绘图')

```

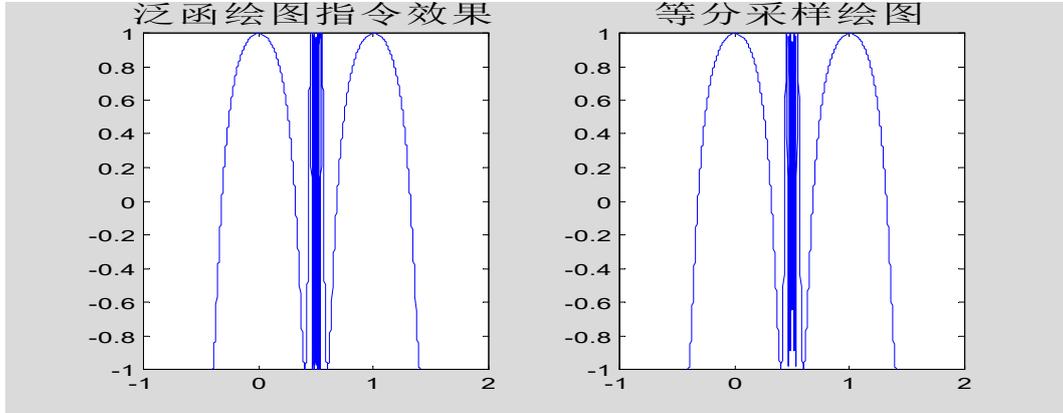


图 7.4.1.12-1 右图曲线中底部有明显绘图缺陷

7.4.2 四维表现

7.4.2.1 用色彩表现函数的特征

【*例 7.4.2.1-1】用色图阵表现函数的不同特征。演示：当三维网线图、曲面图的第四个输入宗量取一些特殊矩阵时，色彩就能表现或加强函数的某特征，如梯度、曲率、方向导数等。

```

x=3*pi*(-1:1/15:1);y=x;[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;Z=sin(R)./R;
[dzdx,dzdy]=gradient(Z);dzdr=sqrt(dzdx.^2+dzdy.^2); %计算对 r 的全导数
dz2=del2(Z); %计算曲率 <4>
subplot(1,2,1),surf(X,Y,Z),title('No. 1 surf(X,Y,Z)')
shading faceted,colorbar('horiz'),brighten(0.2)
subplot(1,2,2),surf(X,Y,Z,R),title('No. 2 surf(X,Y,Z,R)')
shading faceted;colorbar('horiz')

```

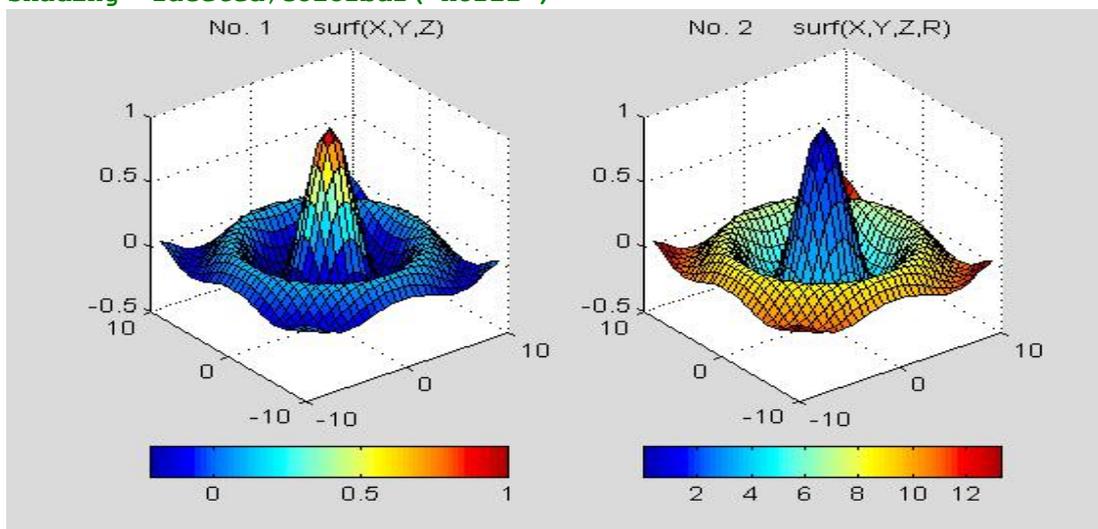


图 7.4.2.1-1 色彩分别表现函数的高度和半径特征

```

clf;subplot(1,2,1),surf(X,Y,Z,dzdx)
shading faceted;brighten(0.1);colorbar('horiz')
title('No. 3 surf(X,Y,Z,dzdx)')
subplot(1,2,2),surf(X,Y,Z,dzdy)
shading faceted;colorbar('horiz')
title('No. 4 surf(X,Y,Z,dzdy)')

```

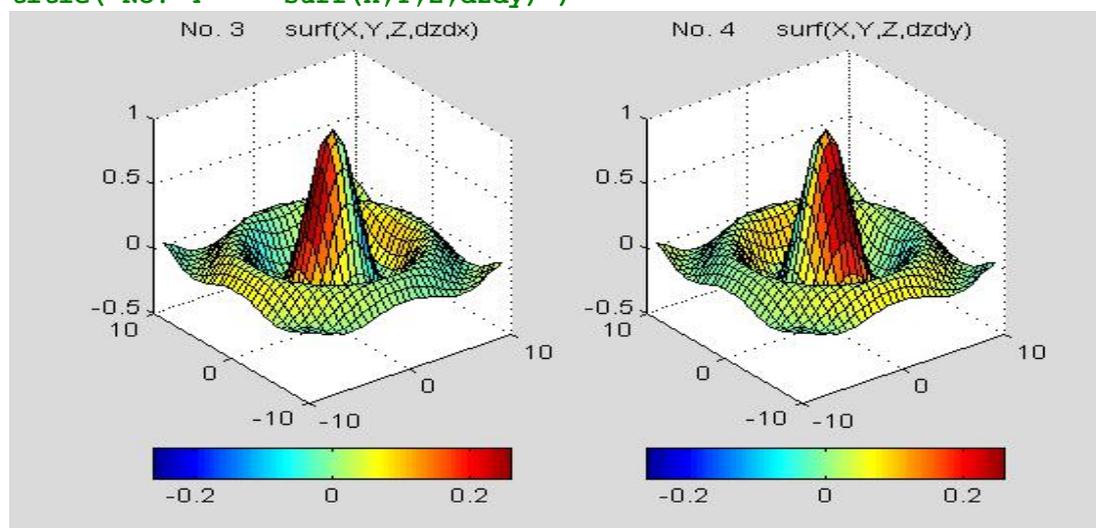


图 7.4.2.1-2 色彩分别表现函数的 x 方向和 y 方向导数特征

```

subplot(1,2,1),surf(X,Y,Z,abs(dzdr))
shading faceted;brighten(0.6);colorbar('horiz')
title('No. 5 surf(X,Y,Z,abs(dzdr))')
subplot(1,2,2),surf(X,Y,Z,abs(dz2))
shading faceted;colorbar('horiz')
title('No. 6 surf(X,Y,Z,abs(dz2))')

```

%<17>

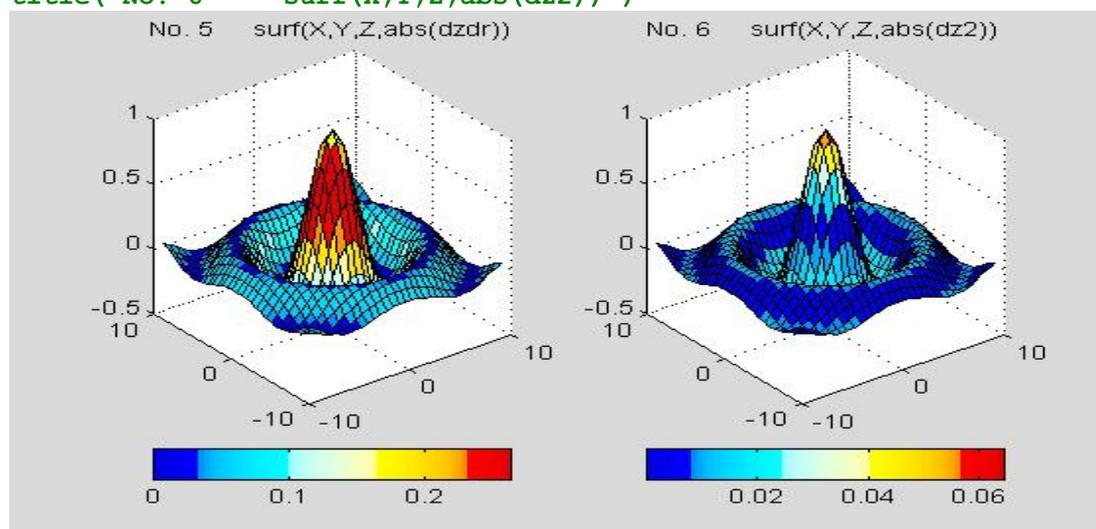


图 7.4.2.1-3 色彩分别表现函数的径向导数和曲率特征

7.4.2.2 切片图和切片等位线图

【*例 7.4.2.2-1】利用 `slice` 和 `contourslice` 表现 MATLAB 提供的无限大水体中水下射流速度

数据 flow。**flow** 是一组定义在三维空间上的函数数据。

在本例中，从图中的色标尺可知，深红色表示“正速度”（向图的左方），深蓝表示“负速度”（向图的右方）。

%以下指令用切面上的色彩表现射流速度

```
clf; [X,Y,Z,V]=flow; %取 4 个 (50×25×25) 的射流数据矩阵，v 是射流速度。
```

```
x1=min(min(min(X))); x2=max(max(max(X))); %取 x 坐标上下限
```

```
y1=min(min(min(Y))); y2=max(max(max(Y))); %取 y 坐标上下限
```

```
z1=min(min(min(Z))); z2=max(max(max(Z))); %取 z 坐标上下限
```

```
sx=linspace(x1+1.2,x2,5); %确定 5 个垂直 x 轴的切面坐标
```

```
sy=0; %在 y=0 处，取垂直 y 轴的切面
```

```
sz=0; %在 z=0 处，取垂直 z 轴的切面
```

```
slice(X,Y,Z,V,sx,sy,sz); %画切片图
```

```
view([-12,30]); shading interp; colormap jet; axis off; colorbar
```

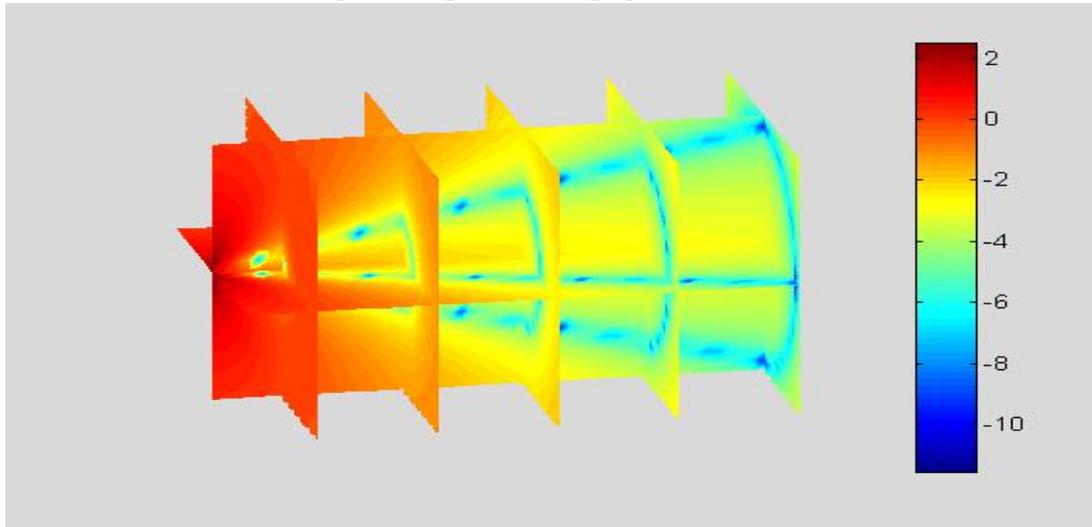


图 7.4.2.2-1 切片图

%以下指令用等位线表现射流速度

```
clf; v1=min(min(min(V))); v2=max(max(max(V))); %射流速度上下限
```

```
cv=linspace(v1,v2,15); %在射流上下限之间取 15 条等位线
```

```
contourslice(X,Y,Z,V,sx,sy,sz,cv); view([-12,30])
```

```
colormap jet; colorbar; box on
```

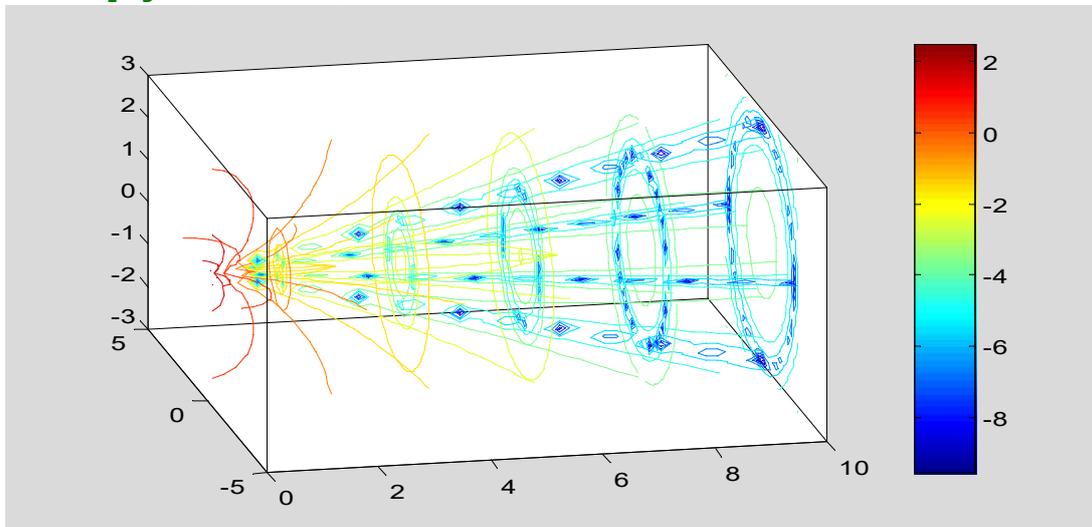


图 7.4.2.2-2 切片等位线图

7.4.3 动态图形

7.4.3.1 彗星状轨迹图

【*例 7.4.3.1-1】简单二维示例。（请读者自己在指令窗中运行以下指令。）

```
shg;n=10;t=n*pi*(0:0.0005:1);x=sin(t);y=cos(t);  
plot(x,y,'g');axis square;hold on  
comet(x,y,0.01);hold off
```

【*例 7.4.3.1-2】卫星返回地球的运动轨线示意。（请读者自己在指令窗中运行以下指令。）

```
shg;R0=1; %以地球半径为一个单位  
a=12*R0;b=9*R0;T0=2*pi; %T0 是轨道周期  
T=5*T0;dt=pi/100;t=[0:dt:T]';  
f=sqrt(a^2-b^2); %地球与另一焦点的距离  
th=12.5*pi/180; %卫星轨道与 x-y 平面的倾角  
E=exp(-t/20); %轨道收缩率  
x=E.*(a*cos(t)-f);y=E.*(b*cos(th)*sin(t));z=E.*(b*sin(th)*sin(t));  
plot3(x,y,z,'g'); %画全程轨线  
[X,Y,Z]=sphere(30);X=R0*X;Y=R0*Y;Z=R0*Z; %获得单位球坐标  
grid on,hold on,surf(X,Y,Z),shading interp %画地球  
x1=-18*R0;x2=6*R0;y1=-12*R0;y2=12*R0;z1=-6*R0;z2=6*R0;  
axis([x1 x2 y1 y2 z1 z2]) %确定坐标范围  
view([117 37]),comet3(x,y,z,0.02),hold off %设视角、画运动轨线
```

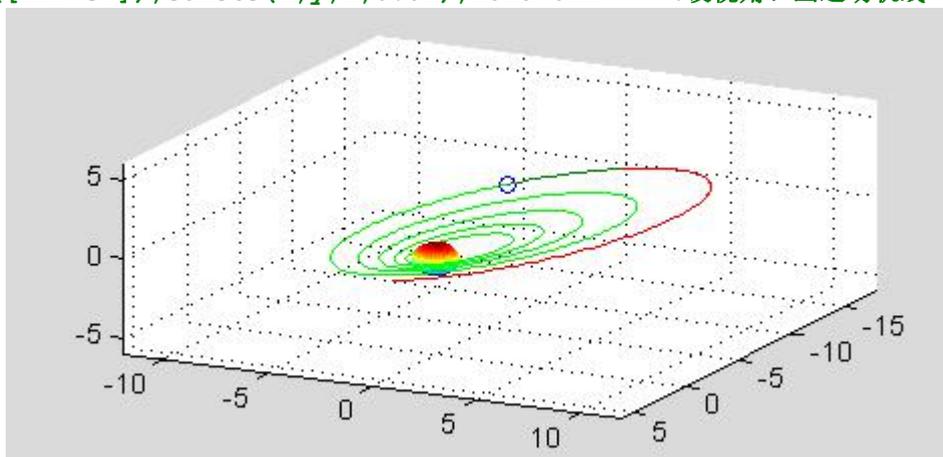


图 7.4.3.1-1 卫星返回地球轨线示意

7.4.3.2 色图的变幻

【例 7.4.3.2-1】色彩变幻（因印刷关系，无法表现。请读者自己在指令窗中运行以下指令。）

注意：在 256 色情况下，才可被正确执行）

```
peaks  
spinmap
```

7.4.3.3 影片动画

【*例 7.4.3.3-1】三维图形的影片动画。（因印刷关系，无法表现本例。请读者自己在指令窗中运行以下指令。）

```
clf;shg,x=3*pi*(-1:0.05:1);y=x;[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps; Z=sin(R)./R;
h=surf(X,Y,Z);colormap(jet);axis off
n=12;mmm=moviein(n); %预设画面矩阵。新版完全可以取消此指令。
for i=1:n
    rotate(h,[0 0 1],25); %是图形绕 z 轴旋转 25 度/每次
    mmm(:,i)=getframe; %捕获画面。新版改为 mmm(i)=getframe。
end
movie(mmm,5,10) %以每秒 10 帧速度，重复播放 5 次。
```

7.5 三维图形的精细控制

7.5.1 视点控制和图形的旋转

7.5.1.1 视点控制 view

7.5.1.2 图形旋转 rotate

【*例 7.5.1.2-1】旋转指令示例。（利用 rotate 制作动画，请看例 7.4.4.3-1）

```
shg;clf;[X,Y] = meshgrid([-2:.2:2]); Z = 4*X.*exp(-X.^2-Y.^2);
G=gradient(Z);subplot(1,2,1),surf(X,Y,Z,G)
subplot(1,2,2),h=surf(X,Y,Z,G);
rotate(h,[-2,-2,0],30,[2,2,0]),colormap(jet)
```

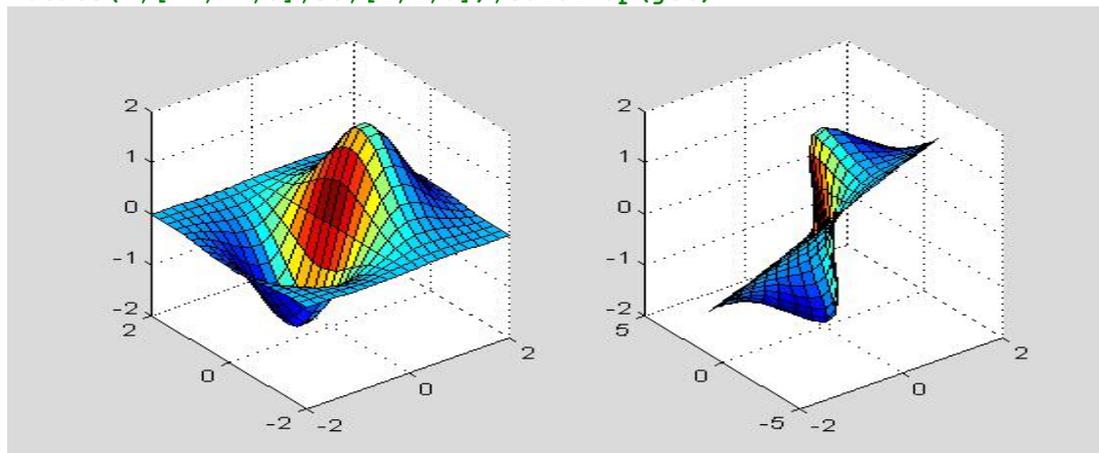


图 7.5.1.2-1 图形对象的旋转

7.5.2 色彩控制

7.5.2.1 用色风格 colordef

7.5.2.2 色图 colormap

【*例 7.5.2.2-1】红绿蓝三色色图。演示：色图和色图矩阵。

```
CM=[1 0 0;0 1 0;0 0 1];m=size(CM,1);Y=[1:m+1;1:m+1]';  
pcolor(Y),colormap(CM)
```

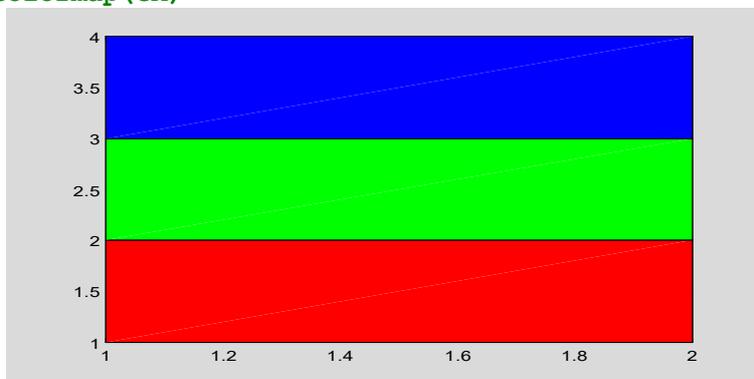


图 7.5.2.2-1 红绿蓝三色色图

【*例 7.5.2.2-2】随机色图。演示：色图和色图矩阵。

```
rand('seed',2);CM=rand(16,3);m=size(CM,1);Y=[1:m+1;1:m+1]';  
pcolor(Y),colormap(CM)
```

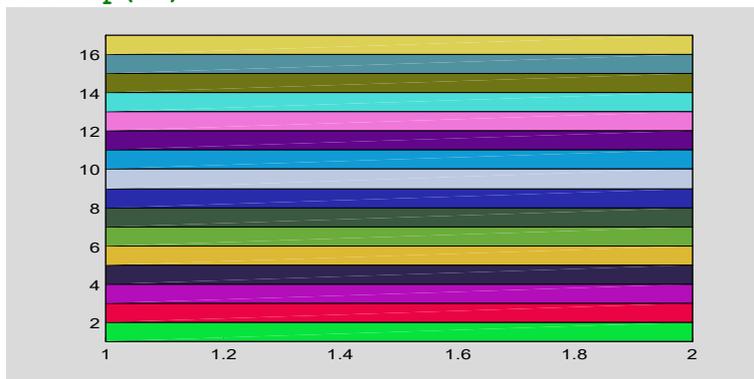


图 7.5.2.2-2 随机 16 色色图

【*例 7.5.2.2-3】演示：（A）用 MATLAB 预定义的两个色图矩阵，构成一个更大的色图阵。见指令<2>。（B）把色轴的范围设置得比着色阵 C 的数据范围小，使色图 CM 的两端色彩展宽，中间色彩压缩。见指令<4>。（C）把色轴的上限增加一个 C 阵数据宽度，使着色时只使用色图 CM 上半阵的色彩。这种方法使同一图形窗中的不同对象实际上可以使用 MATLAB 的不同预定义色图。见指令<6>。（D）为了使色标尺正确反映色轴的设置，指令 **colorbar** 必须在 **caxis** 之后使用。

```
Z=peaks(20);C=Z;Cmin=min(min(C));Cmax=max(max(C));DC=Cmax-Cmin;  
CM=[autumn;winter]; %<2>  
colormap(CM),subplot(1,3,1),surf(Z,C)  
caxis([Cmin+DC*2/5,Cmax-DC*2/5]),colorbar('horiz') %<4>  
subplot(1,3,2),surf(Z,C),colorbar('horiz')  
subplot(1,3,3),surf(Z,C),caxis([Cmin,Cmax+DC]),colorbar('horiz') %<6>
```

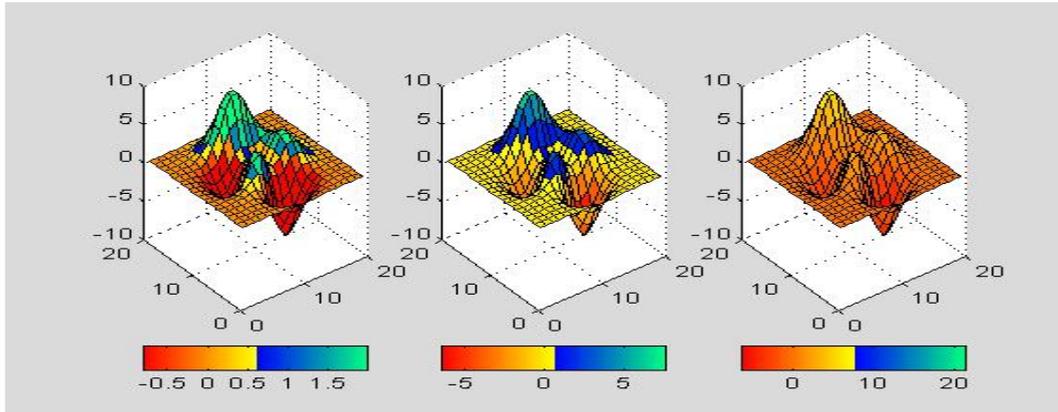


图 7.5.2.2-3 色图、色轴和色标尺的配合使用

7.5.2.3 浓淡处理 shading

【*例 7.5.2.3-1】三种浓淡处理方式比较。

```

clf; Z=peaks(15); colormap(jet)
subplot(1,3,1), surf(Z)
subplot(1,3,2), surf(Z), shading flat
subplot(1,3,3), surf(Z), shading interp

```

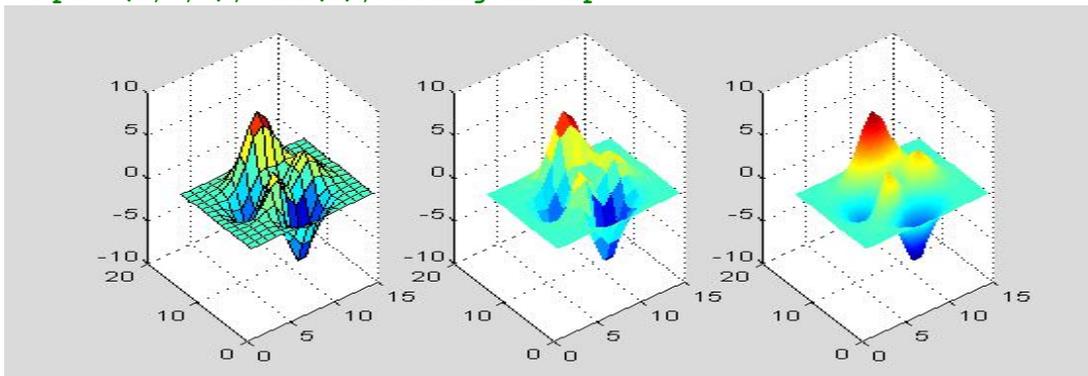


图 7.5.2.3-1 浓淡处理方式比较

7.5.3 照明和材质处理

7.5.3.1 灯光 light

7.5.3.2 照明模式 lighting

7.5.3.3 控制光效果的材质指令 material

7.5.3.4 surf1 指令的精细调用格式

【*例 7.5.3-1】灯光、照明、材质指令所表现的图形。

```

clf;
[X,Y,Z]=sphere(40);

```

```

colormap(jet) %<3>
subplot(1,2,1);surf(X,Y,Z);shading interp %<4>
light('position',[2,-2,2],'style','local') %<5>
lighting phong %<6>
material([0.5,0.3,0.5,10,0.5]) %<7>
subplot(1,2,2);surf(X,Y,Z,-Z);shading flat %<8>
light;lighting flat %<9>
light('position',[-1,-1,-2],'color','y') %<10>
light('position',[-1,0.5,1],'style','local','color','w') %<11>
material([0.4,0.5,0.3,10,0.3]) %<12>

```

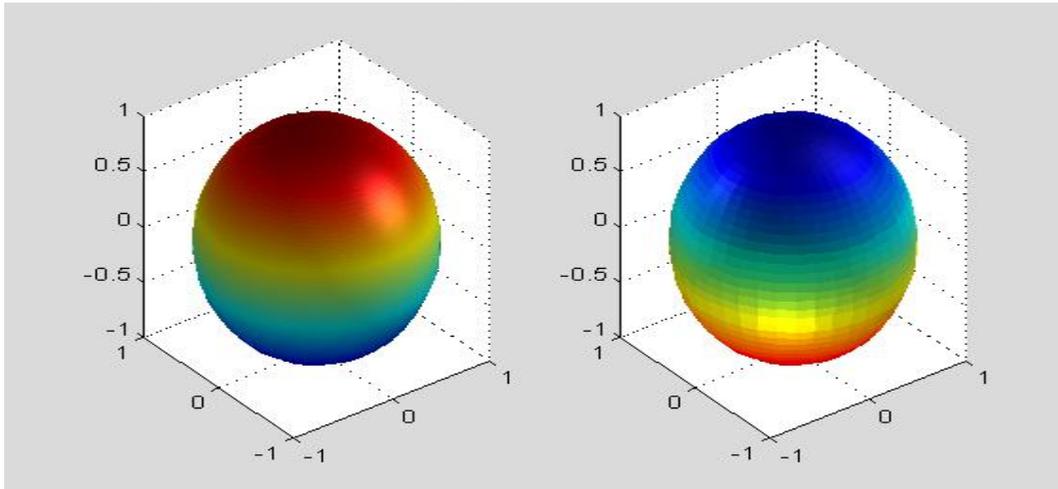


图 7.5.3-1 灯光、照明、材质指令所表现的图形

7.6 图象

7.6.1 图象的类别和显示

7.6.2 图象的读写

7.6.2.1 MAT 文件形式图象数据的保存和提取

7.6.2.2 标准格式图象文件的读写

【*例 7.6.2.2-1】图象文件的读取和图象的显示

(1) 变址图象的读取和显示

```

[X,cmap]=imread('trees.tif'); %读取 TIFF 格式文件
clf
image(X);colormap(cmap);axis image off %显示图象,并保持宽高比。

```

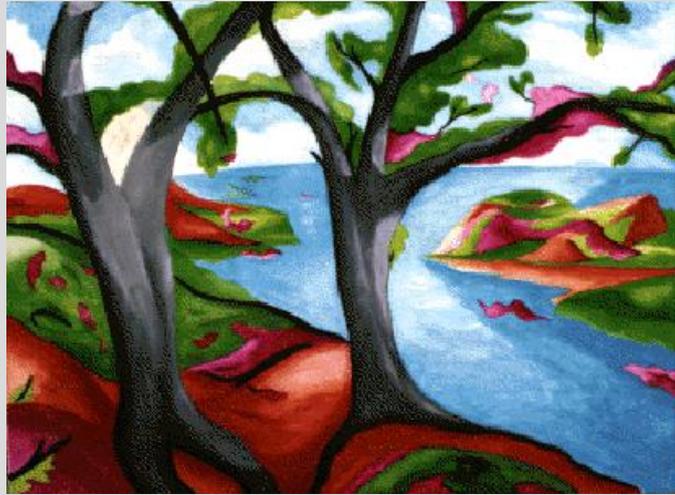


图 7.6.2.2-1-1 变址图象

(2) 亮度图象的读取和显示

```
X=imread('saturn.tif');
imagesc(X);colormap(gray);axis image off
```

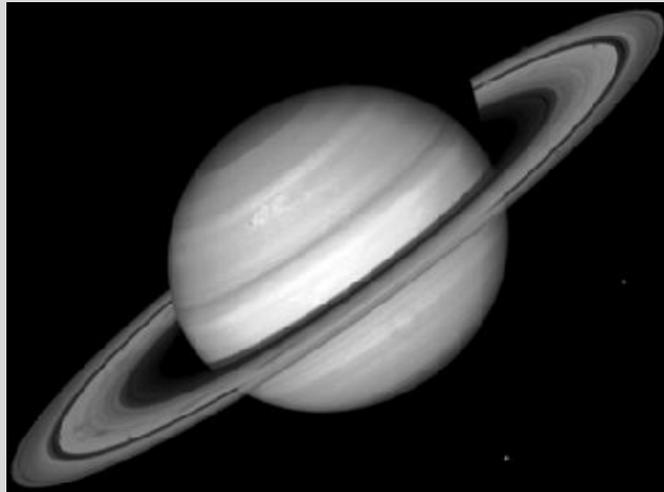


图 7.6.2.2-1-2 亮度图象

(3) 真彩图象的读取、变换格式及显示

```
X=imread('flowers.tif'); %读取 TIFF 格式的图象文件
imwrite(X,'ff.jpg','Quality',100) %把图象以 JPG 格式文件保存
imfinfo('ff.jpg') %读取图象文件特征信息
image(imread('ff.jpg')) %读取 JPG 格式文件，并显示图象。
axis image off %保持宽高比和隐去坐标
ans =
    Filename: 'ff.jpg'
    FileModDate: '10-Mar-2000 19:41:42'
    FileSize: 193220
    Format: 'jpg'
```

```

FormatVersion: ''
      Width: 500
      Height: 362
      BitDepth: 24
      ColorType: 'truecolor'
FormatSignature: ''

```



图 7.6.2.2-1-3 真彩图象

7.6.3 捕获图形生成图象文件

【*例 7.6.3-1】本例目的：（1）加深对 `getframe` 获取图形构架数据的理解。（2）比较原图和再生图，可以发现差异。（请读者在指令窗中，运行以下指令）

```

figure(2);surf(peaks)           %在 2 号图形窗中生成图形
f=getframe(2);                 %捕获 2 号窗的图形数据
figure(1)                       %打开 1 号窗
image(f.cdata);colormap(f.colormap) %在 1 号窗中，重现图形。

```

【*例 7.6.3-2】本例目的：（A）任何图形在 `getframe` 作用下都可成为图象。（B）比较原图和再生图，可以发现差异。（请读者在指令窗中，运行以下指令）

```

figure(2),surf(peaks),[X,cmap]=getframe(2);
imwrite(X,cmap,'fff.tif');figure(1);image(imread('fff.tif'))

```

7.7 图形窗的图形编辑功能

7.7.1 图形窗菜单和工具条简介

【*例 7.7.1-1】运行以下指令产生如图 7.7.1-1 所示的图形窗。

```

clf;shg,t=(pi*(0:1000)/1000)';y1=sin(t);y12=sin(t).*sin(10*t);

```

```
plot(t,y12,'b-',t,[y1,-y1],'r:'),axis([0,pi,-1,1])
```

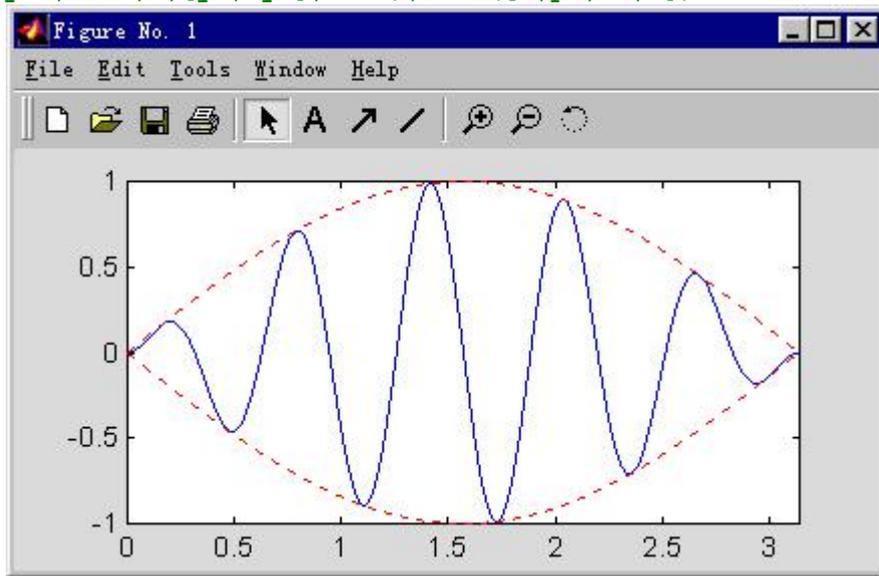


图 7.7.1-1 MATLAB 5.3 版图形窗

7.7.1.1 工具条简介

7.7.1.2 若干重要的菜单选项

7.7.2 二维图形的交互编辑示例

【例 7.7.2-1】把一幅如图 7.7.1-1 所示的简单图形，编辑成图 7.7.2-4 那样。

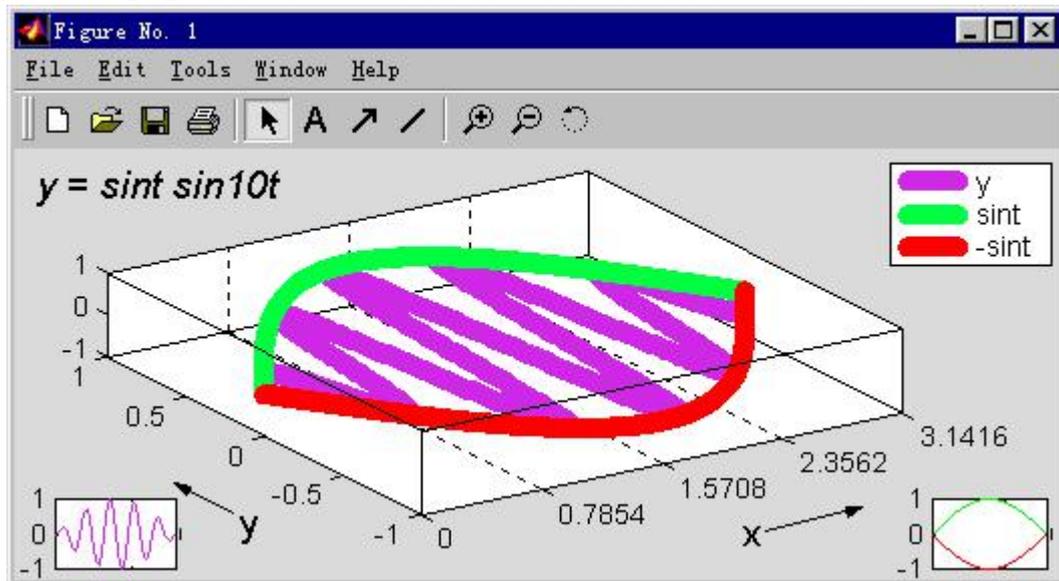


图 7.7.2-4 经交互编辑后的图形

7.8 函数绘图的简捷指令

7.8.1 一元函数简捷绘图指令

【*例 7.8.1-1】绘制 $y = \frac{2}{3}e^{-\frac{t}{2}} \cos \frac{\sqrt{3}}{2}t$ 和它的积分 $s(t) = \int_0^t y(t)dt$ 在 $[0, 4\pi]$ 间的图形。

```
syms t tao;  
y=2/3*exp(-t/2)*cos(sqrt(3)/2*t); %定义符号函数  
s=subs(int(y,t,0,tao),tao,t); %获得积分函数  
subplot(1,2,1),ezplot(y,[0,4*pi]);grid  
subplot(1,2,2),ezplot(s,[0,4*pi]);grid  
title('s = \inty(t)dt') %重写右子图图名
```

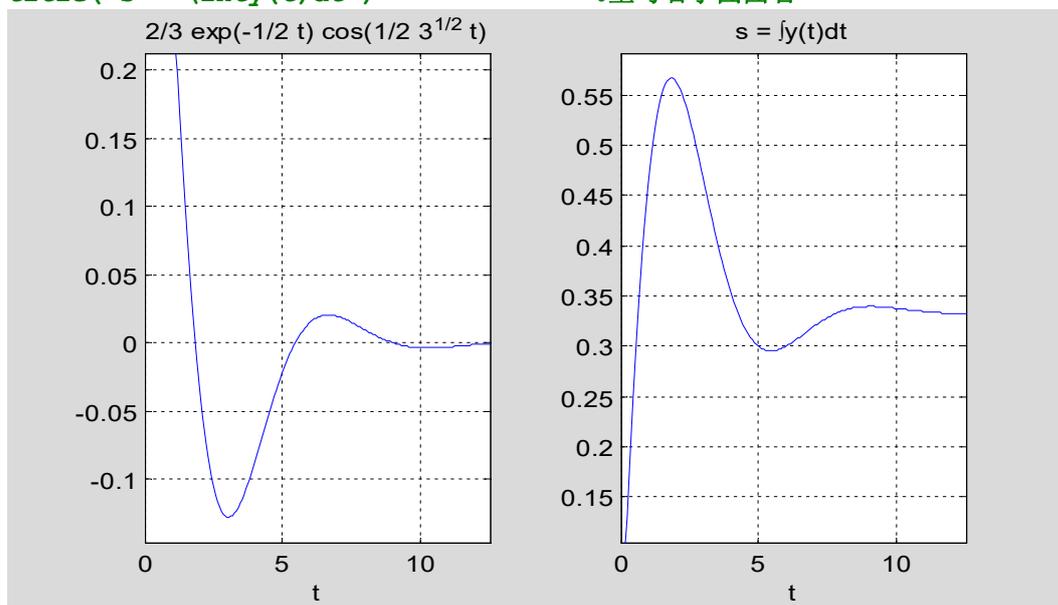


图 7.8.1-1 ezplot 使用示例

7.8.2 二元函数简捷绘图指令

【*例 7.8.2-1】在园域上画 $z = xy$ 的图形。

```
clf,ezsurf('x*y','circ');shading flat;view([-18,28])
```

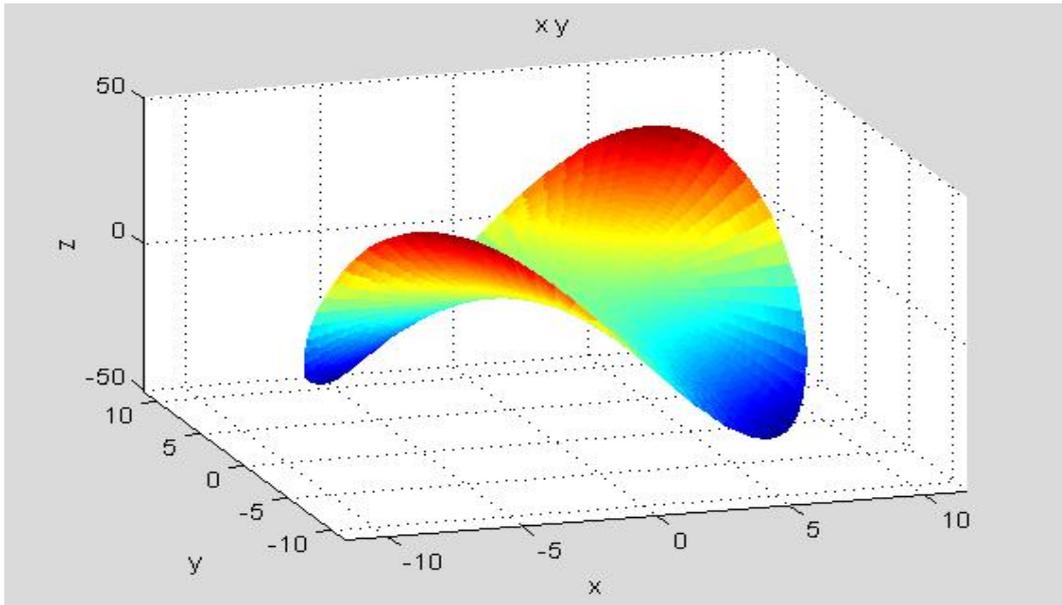


图 7.8.2-1 ezsurf 绘图演示一

【*例 7.8.2-2】使用球坐标参量画部分球壳。

```

x='cos(s)*cos(t)';y='cos(s)*sin(t)';z='sin(s)';
ezsurf(x,y,z,[0,pi/2,0,3*pi/2]) %0 ≤ s ≤ 0.5π,0 ≤ t ≤ 1.5π
view(17,40);shading interp;colormap(spring)
light('position',[0,0,-10],'style','local')
light('position],[-1,-0.5,2],'style','local')
material([0.5,0.5,0.5,10,0.3])

```

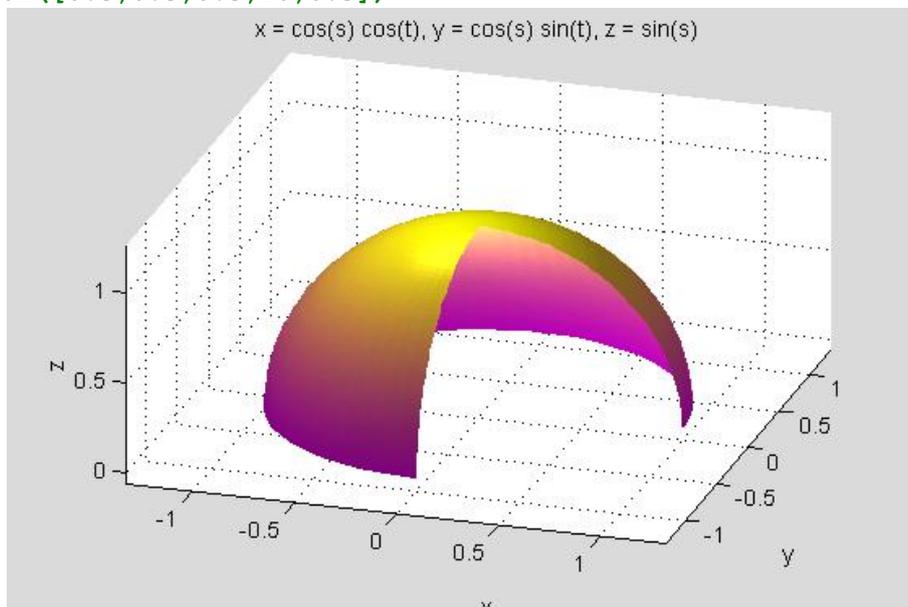


图 7.8.2-2 ezsurf 绘图演示二