

第四章 绘图功能

作为一个功能强大的工具软件，Matlab具有很强的图形处理功能，提供了大量的二维、三维图形函数。由于系统采用面向对象的技术和丰富的矩阵运算，所以在图形处理方面即常方便又高效。

绘制图表的基础步骤:

- 1、准备图表的数据;
- 2、设置显示图表的范围;
- 3、绘图, 并设置相应的参数;
- 4、设置坐标轴属性;
- 5、添加图形注释。

Matlab图形功能简介

Matlab具有强大的图形显示功能

具体介绍：

1、二维图形显示功能

2、二维图形显示功能

二维图形

一、plot函数

函数格式:

plot(x): x下标为横坐标, 元素数值为纵坐标;

plot(x,y): (1)x、y同为向量;

(2)x为向量, y为某维数与x向量相同的矩阵。

函数功能: 以向量x、y为轴, 绘制曲线。

【例1】 在区间 $0 \leq X \leq 2\pi$ 内, 绘制正弦曲线 $Y = \sin(X)$, 其程序为:

```
x=0:pi/100:2*pi;
```

```
y=sin(x);
```

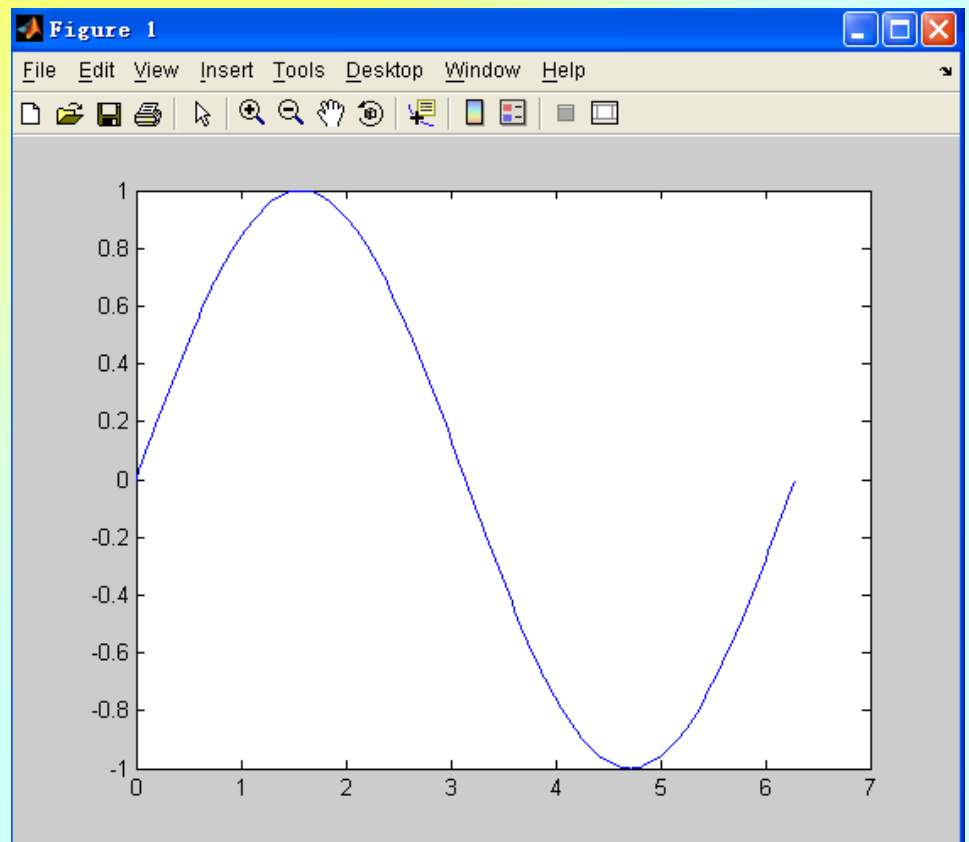
```
plot(x,y)
```

Plot函数示例

```
>> t=0:pi/100:2*pi;
```

```
>> y=sin(t);
```

```
>> plot(t,y)
```



Plot函数示例

自动调用不同颜色

```
t=0:pi/100:2*pi;
```

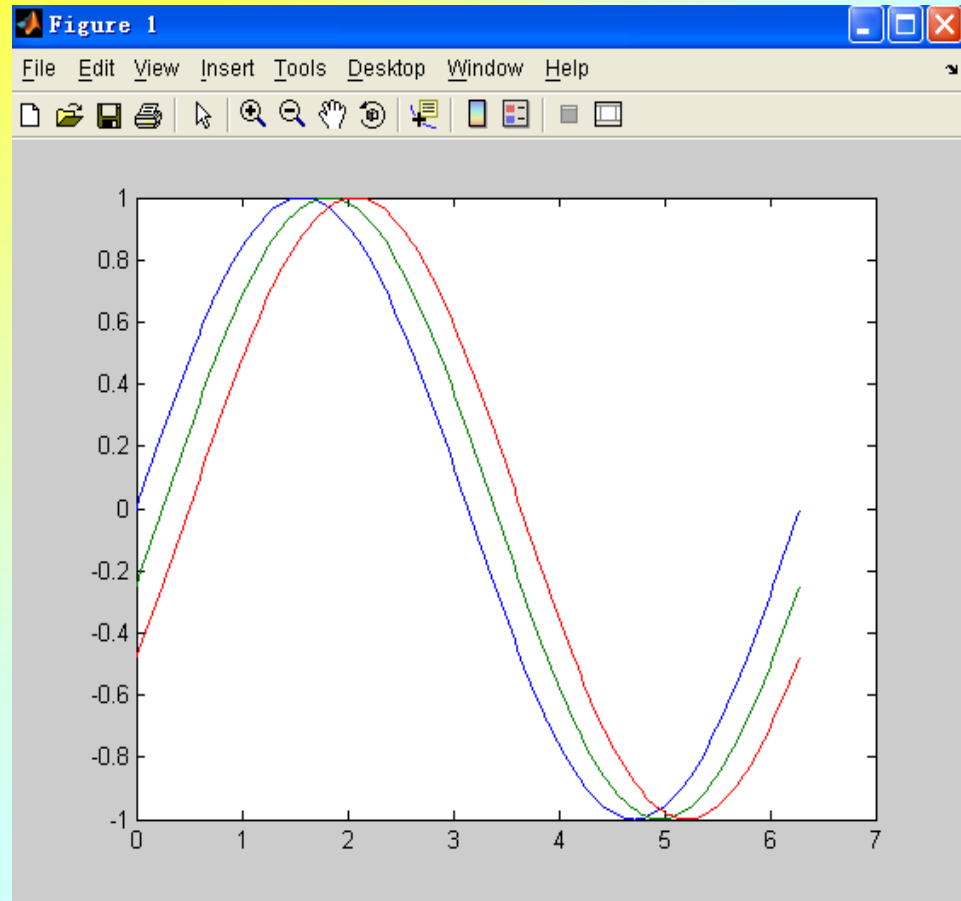
```
y=sin(t);
```

```
plot(t,y)
```

```
y2=sin(t-0.25);
```

```
y3=sin(t-0.5);
```

```
plot(t,y,t,y2,t,y3)
```



指定线型

线型:

实线 “ — ”

点划线 “ —. ”

虚线 “ : ”

破折线 “ —— ”

五角星形 “ p ”

六角星形 “ h ”

线型示例

```
t=0:pi/100:2*pi;
```

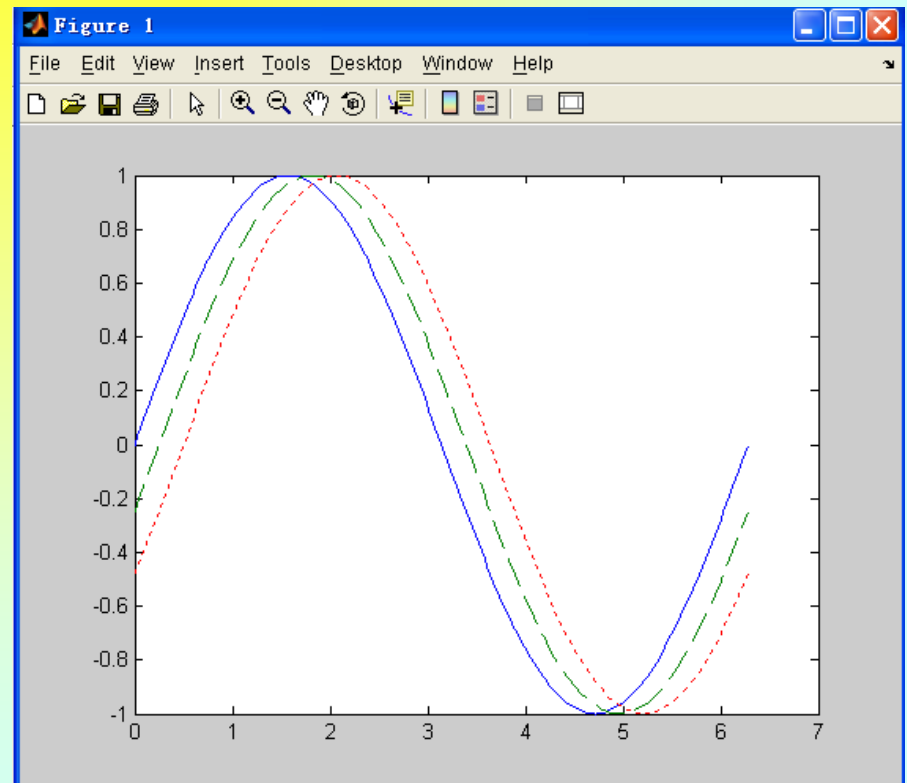
```
y=sin(t);
```

```
plot(t,y)
```

```
y2=sin(t-0.25);
```

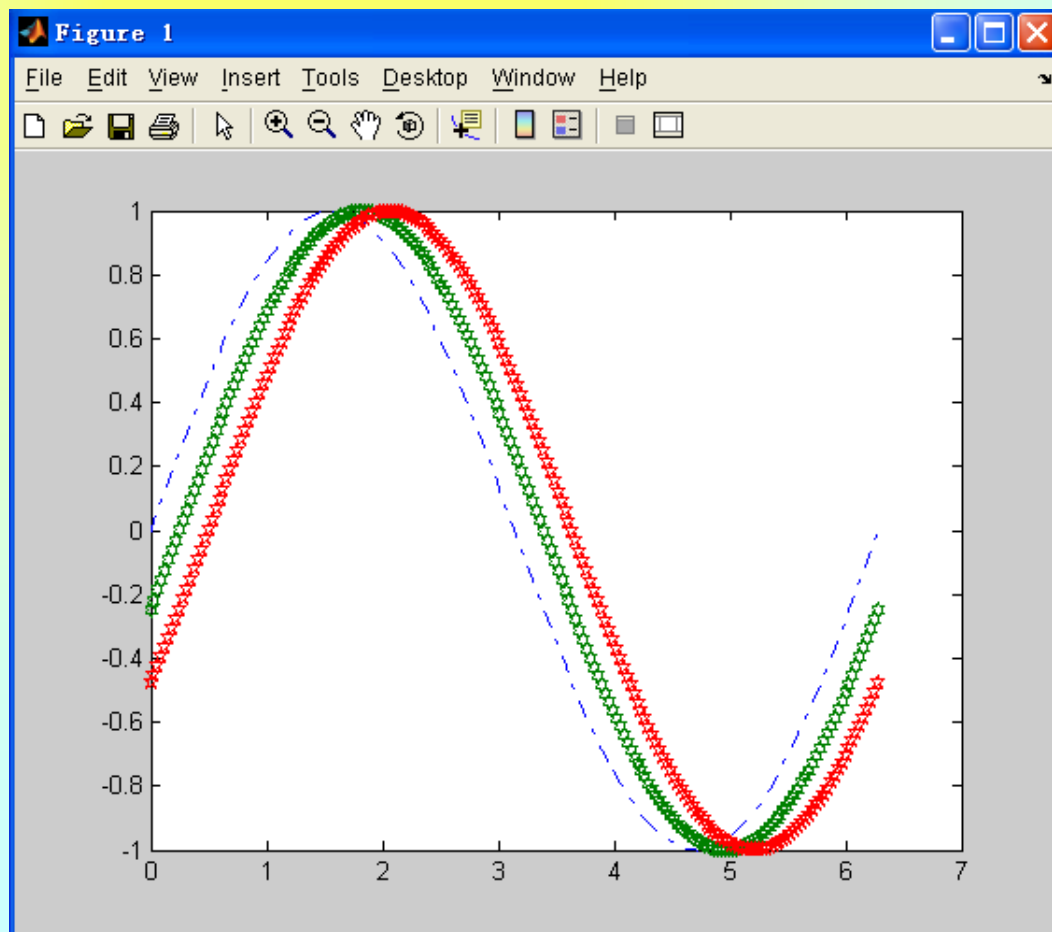
```
y3=sin(t-0.5);
```

```
plot(t,y,'-',t,y2,'--',t,y3,':')
```



线型示例

```
plot(t,y,'-.',t,y2,'h',t,y3,'p')
```



指定线型

标记类型:

圆圈 “ o ”

点 “ . ”

星号 “ * ”

叉号 “ x ”

加号 “ + ”

正方形 “ s ”

菱形 “ d ”

向下三角形 “ v ”

指定线型

预定的颜色标识：

青色 “c”、紫色 “m”

黄色 “y”、黑色 “d”

红色 “r”、绿色 “g”

蓝色 “b”、白色 “w”

图形标记

在绘制图形的同时，可以对图形加上一些说明，如图形名称、图形某一部分的含义、坐标说明等，将这些操作称为添加图形标记。

```
title('加图形标题');
```

```
xlabel('加x轴标记');
```

```
ylabel('加y轴标记');
```

```
text(X,Y,'添加文本');
```

设定坐标轴

用户若对坐标系统不满意，可利用**axis**命令对其重新设定。

axis([xmin xmax ymin ymax]) 设定最大和最小值

Axis auto 将坐标系统返回到自动缺省状态

axis square 将当前图形设置为方形

axis equal 两个坐标因子设成相等

axis off 关闭坐标系统

axis on 显示坐标系统

axis ij 坐标原点在左上角

【例】 在坐标范围 $0 \leq X \leq 2\pi$, $-2 \leq Y \leq 2$ 内重新绘制正弦曲线，其程序为：

```
x=linspace(0,2*pi,60); y=sin(x);  
plot(x,y);  
axis([0 2*pi -2 2]);设定坐标轴范围
```

加图例

给图形加图例命令为legend。该命令把图例放置在图形空白处，用户还可以通过鼠标移动图例，将其放到希望的位置。

格式:**legend('图例说明','图例说明');**

【例5】 为正弦、余弦曲线增加图例，其程序为：

```
x=0:pi/100:2*pi;  
y1=sin(x);  
y2=cos(x);  
plot(x,y1,x,y2, '--', 'LineWidth',2);  
legend('sin(x)', 'cos(x)');
```


【例】 在同一坐标内，分别用不同线型和颜色绘制曲线 $y_1=0.2e^{-0.5x}\cos(4\pi x)$ 和 $y_2=2e^{-0.5x}\cos(\pi x)$ ，标记两曲线交叉点。

程序如下：

```
x=linspace(0,2*pi,1000);  
y1=0.2*exp(-0.5*x).*cos(4*pi*x);  
y2=2*exp(-0.5*x).*cos(pi*x);  
k=find(abs(y1-y2)<1e-2);    %查找y1与y2相等点(近似相等)  
                                %的下标  
x1=x(k);                    %取y1与y2相等点的x坐标  
y3=0.2*exp(-0.5*x1).*cos(4*pi*x1); %求y1与y2值相等点  
                                %的 y坐标  
plot(x,y1,x,y2,'k:',x1,y3,'bp');
```

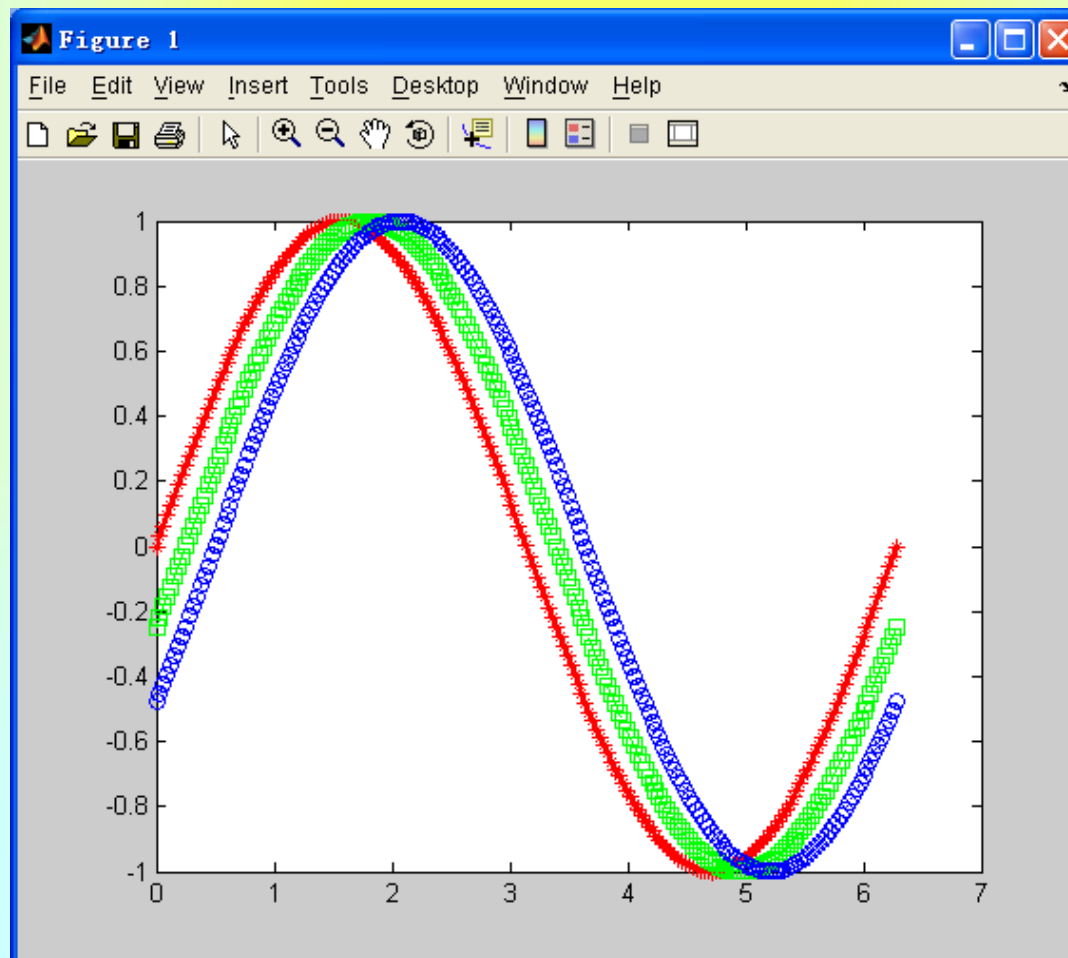
【例】 在同一坐标中，可以绘制3个同心圆，并加坐标控制。

程序如下：

```
t=0:0.01:2*pi;  
x=exp(i*t);  
y=[x;2*x;3*x]';  
plot(y)  
grid on;           %加网格线  
box on;            %加坐标边框  
axis equal         %坐标轴采用等刻度
```

指定线型示例

```
plot(t,y,'-*r',t,y2,'--sg',t,y3,':ob')
```



指定线型

需要指出，参数字符串可以按任意顺序排列。

即`linestyle_maker_color`的顺序可以互换。

`plot(x,y,'go-.)`同样可以绘制一条带有圆形标记的点划线。

指定线条的颜色和大小

LineWidth——以点数为单位指定线条宽度

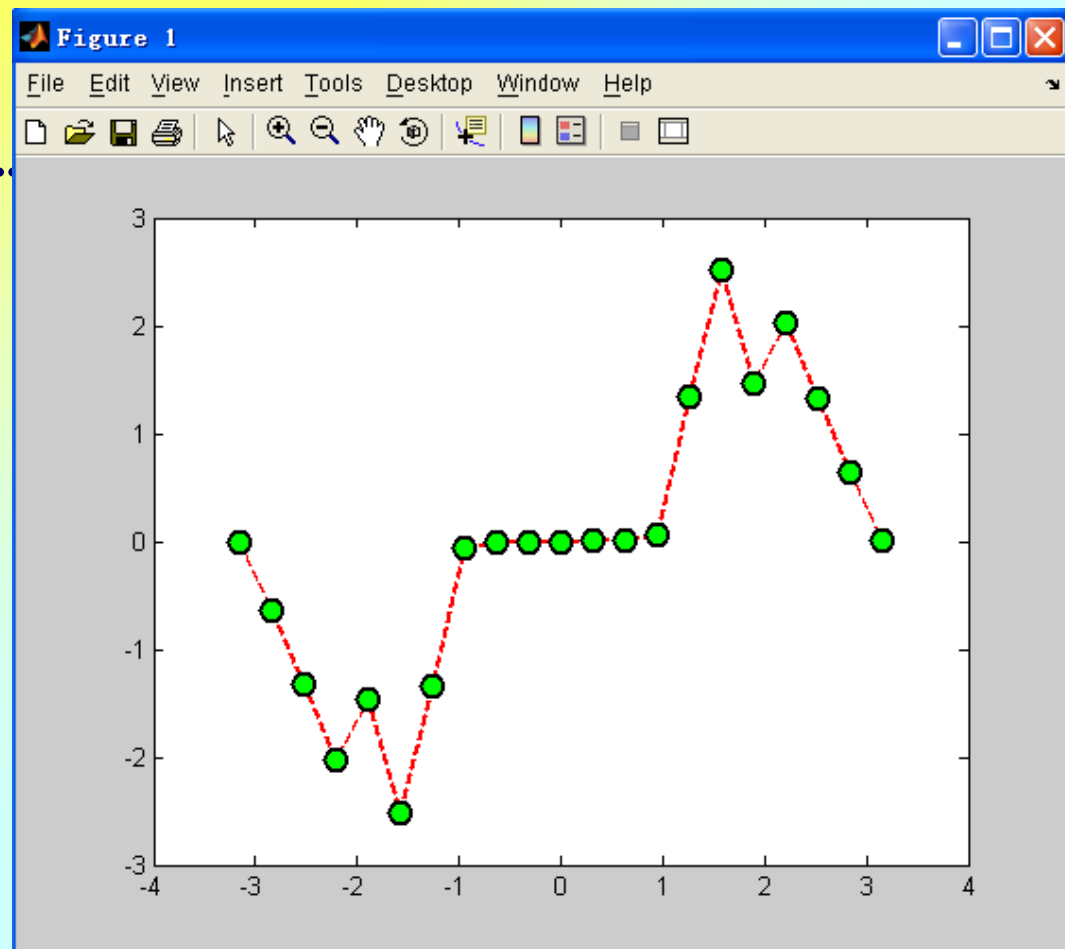
MarkerEdgeColor——指定标记符号的颜色（对封闭标记符号，该属性用于指定边界颜色）

MarkerFaceColor——对封闭标记符号有效，用于指定其填充色。

MarkerSize——以点数为单位指定标记符号的大小。

指定线条的颜色和大小

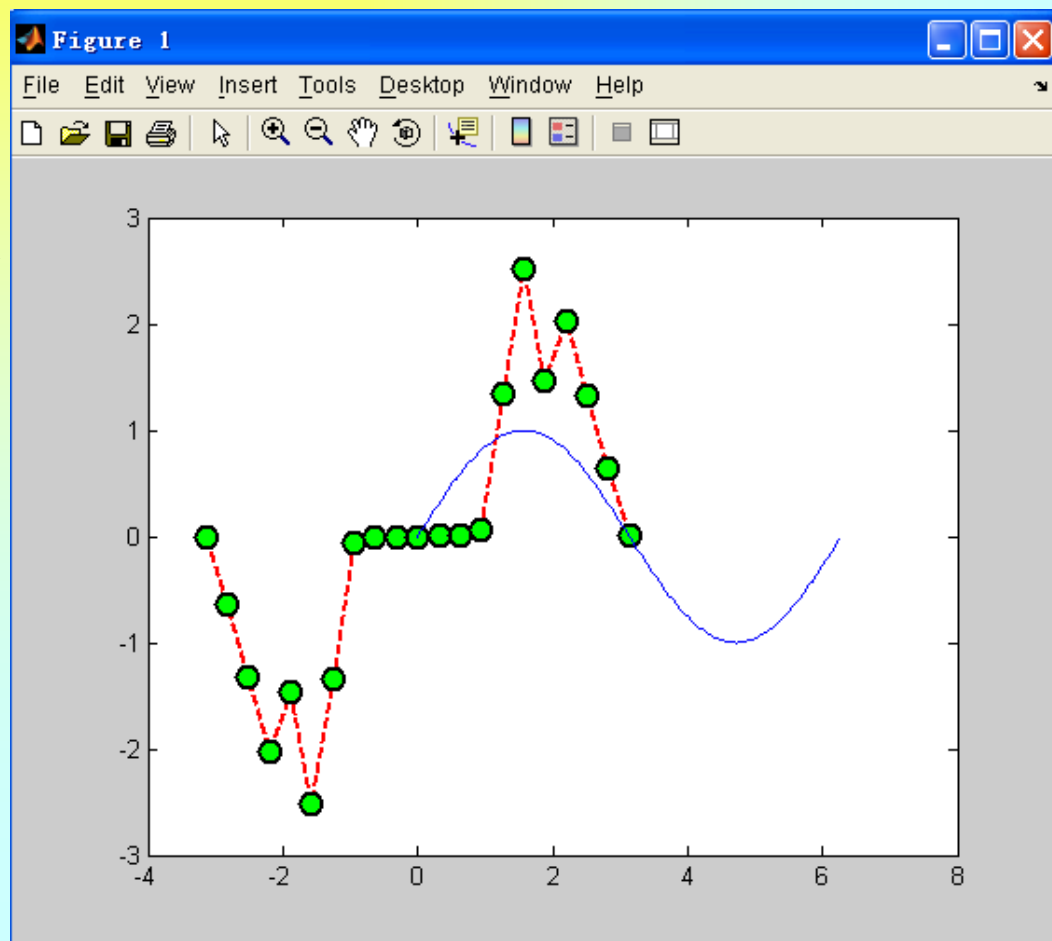
```
x=-pi:pi/10:pi;  
y=tan(sin(x))-sin(tan(x));  
plot(x,y,'--ro','LineWidth',2,...  
      'MarkerEdgeColor','k',...  
      'MarkerFaceColor','g',...  
      'MarkerSize',10)
```



图形叠加

将**hold** 指令设置为**ON**，则再次绘图时将不再清除已经绘制的图形。

```
x=-pi:pi/10:pi;  
y=tan(sin(x))-sin(tan(x));  
plot(x,y,'--ro','LineWidth',2,...  
      'MarkerEdgeColor','k',...  
      'MarkerFaceColor','g',...  
      'MarkerSize',10)  
hold on  
t=0:pi/100:2*pi;  
y=sin(t);  
plot(t,y)
```



离散数据点的绘制

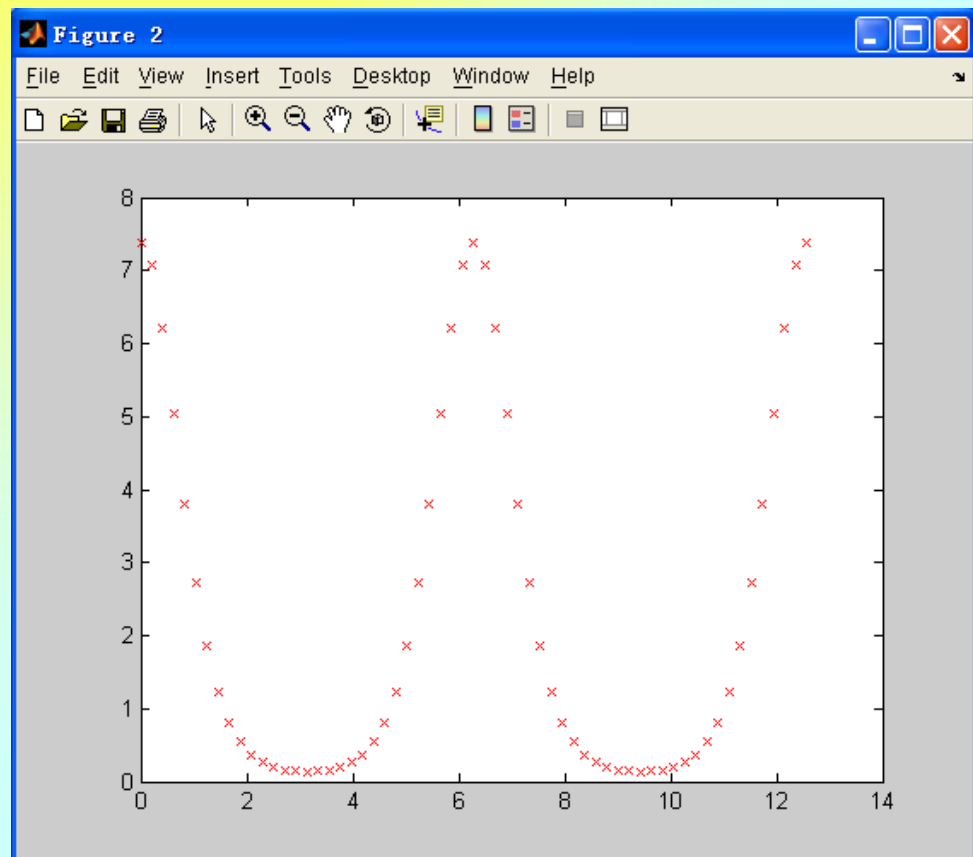
只绘制表示数据的点，而不将其连成完整的线条。

做法：忽略线型属性。

```
x=0:pi/15:4*pi;
```

```
y=exp(2*cos(x));
```

```
plot(x,y,'rx')
```



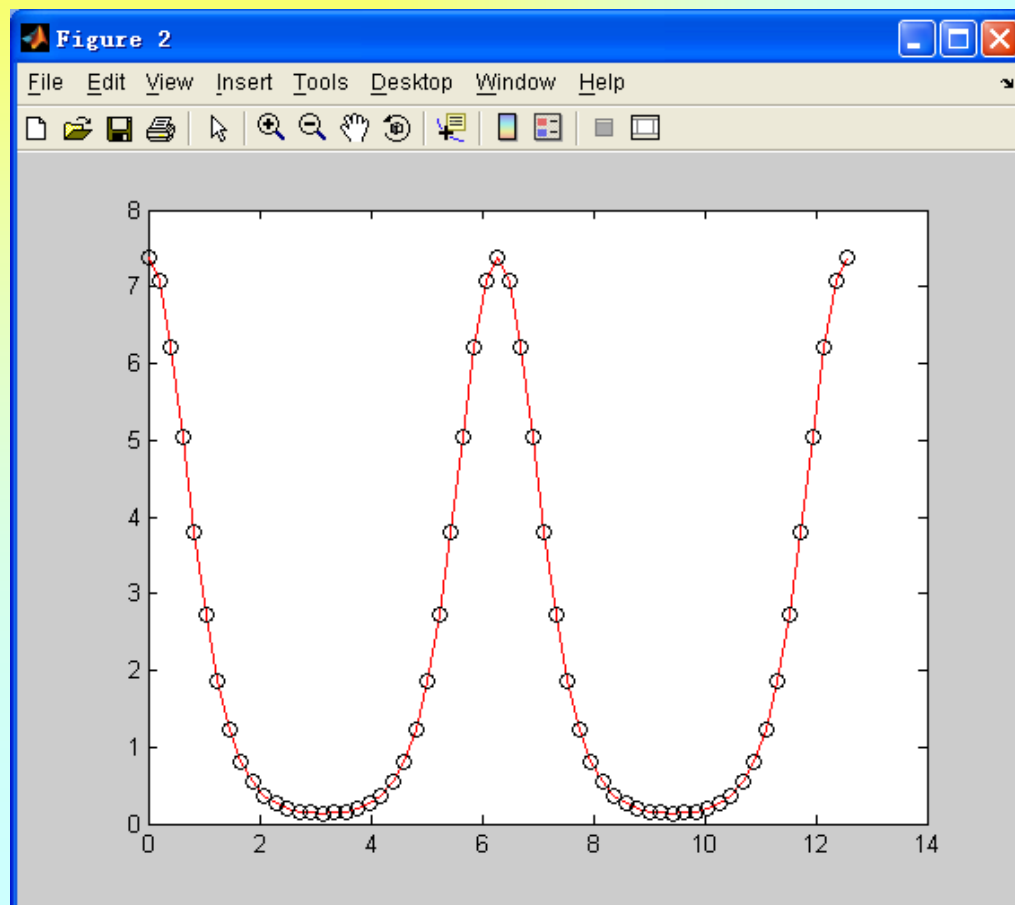
离散数据点的绘制

当然也可以同时绘制数据点和连接这些数据点的线条。

```
x=0:pi/15:4*pi;
```

```
y=exp(2*cos(x));
```

```
plot(x,y,'-r',x,y,'ok')
```



黑白图形的绘制

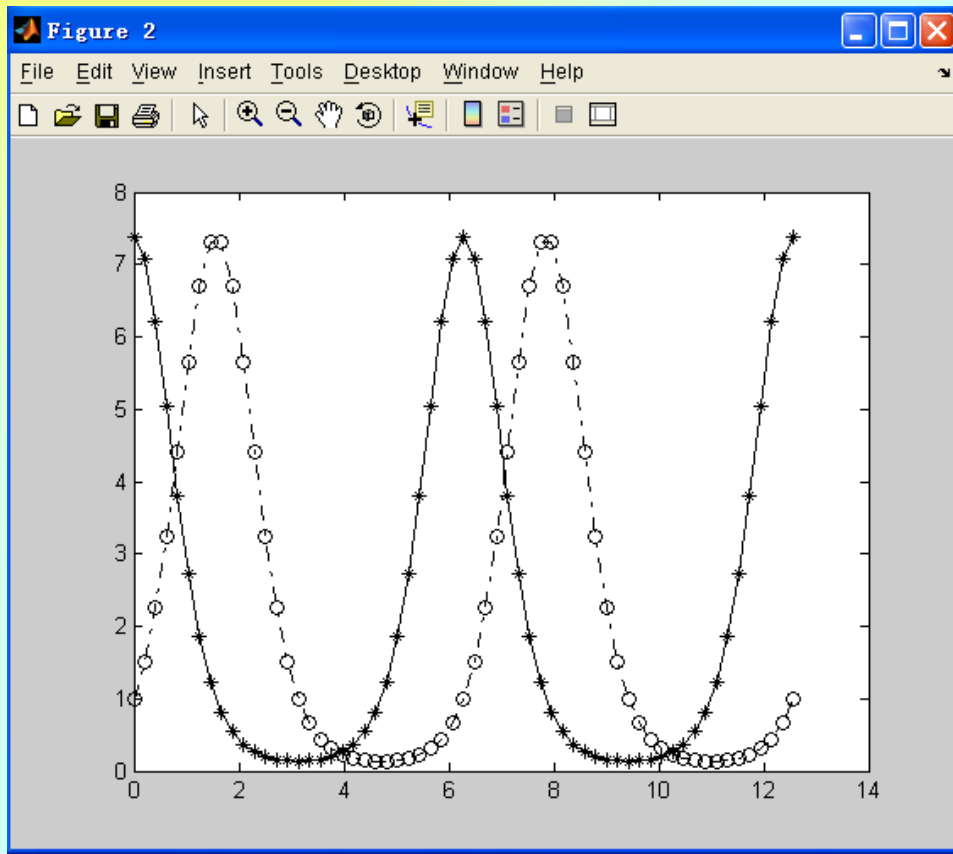
考虑到多数打印机是黑白的，通过线型和标记（而不是颜色）符号来区分不同线条。

```
x=0:pi/15:4*pi;
```

```
y1=exp(2*cos(x));
```

```
y2=exp(2*sin(x));
```

```
plot(x,y1,'-*k',x,y2,'-.ok')
```

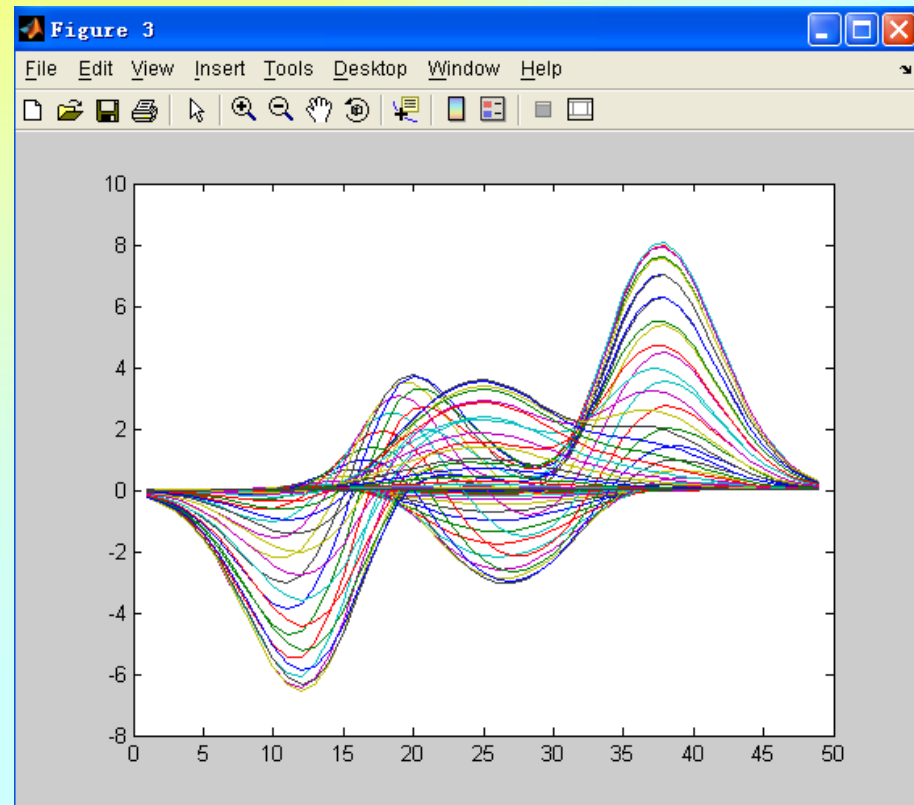


矩阵数据的绘制

如果调用`plot`来绘制一个矩阵，则MATLAB将该矩阵的每一列绘制成一条线，其横坐标为行标`1:m`。

```
Z=peaks;
```

```
plot(Z)
```



绘图的窗口创建

新增绘图窗口: **figure**

`figure(1)`为创建窗口1, `figure(2)`即窗口2...

在同一窗口中绘制多个图形: **subplot**

`subplot(2,1,2)`, 即该窗口有2行1列, 该指令为绘制第一列第二行的图形。

重新绘制上例7个图形，程序变动后如下：

```
x=linspace(0,2*pi,60);
y=sin(x);
z=cos(x);
t=sin(x)./(cos(x)+eps);
ct=cos(x)./(sin(x)+eps);
H1=figure; 创建新窗口并返回句柄到变量H1
plot(x,y); 绘制图形并设置有关属性
title('sin(x)');
axis([0 2*pi -1 1]);
H2=figure; 创建第二个窗口并返回句柄到变量H2
plot(x,z); 绘制图形并设置有关属性
title('cos(x)');

axis([0 2*pi -1 1]);
H3=figure; 同上
plot(x,t);
title('tangent(x)');
axis([0 2*pi -70 70]);
H7=figure; 同上
plot(x,ct);
title('cotangent(x)');
axis([0 2*pi -70 70]);
```

阅读如下程序：

```
x=linspace(0,2*pi,60);  
y=sin(x);  
z=cos(x);  
plot(x,y,'b');  
hold on;  
plot(x,z,'g');  
axis([0 2*pi -1 1]);  
legend('cos','sin');  
hold off
```

绘制正弦曲线

设置图形保持状态

保持正弦曲线同时绘制余弦曲线

关闭图形保持

绘图的窗口创建

figure

$[X,Y]=\text{meshgrid}(-2:0.1:2);$

$Z=X.*\exp(-X.^2-Y.^2);$

subplot(2,1,1)

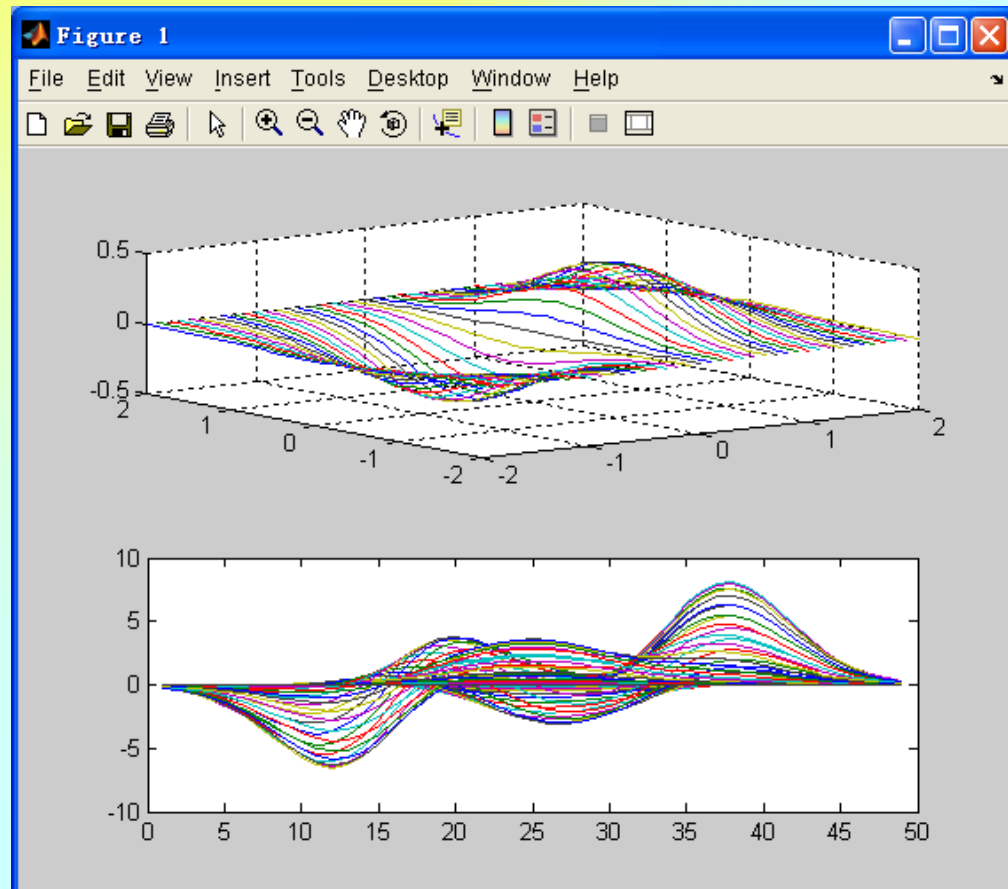
plot3(X,Y,Z)

grid on

$Z=\text{peaks};$

subplot(2,1,2)

plot(Z)



坐标轴标签和图形注释

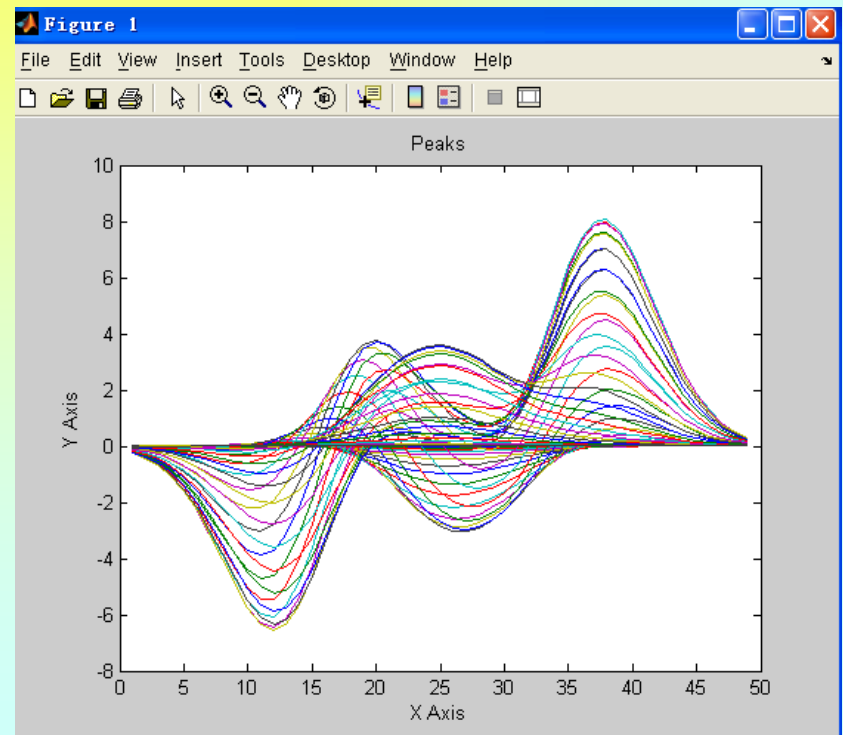
`Z=peaks;`

`plot(Z)`

`xlabel('X Axis')`

`ylabel('Y Axis')`

`title('Peaks')`



创建三维MATLAB曲线图形

三维曲线图的绘制

MATLAB的三维绘图函数主要是`plot3`

函数调用方式`plot3(x,y,z)`

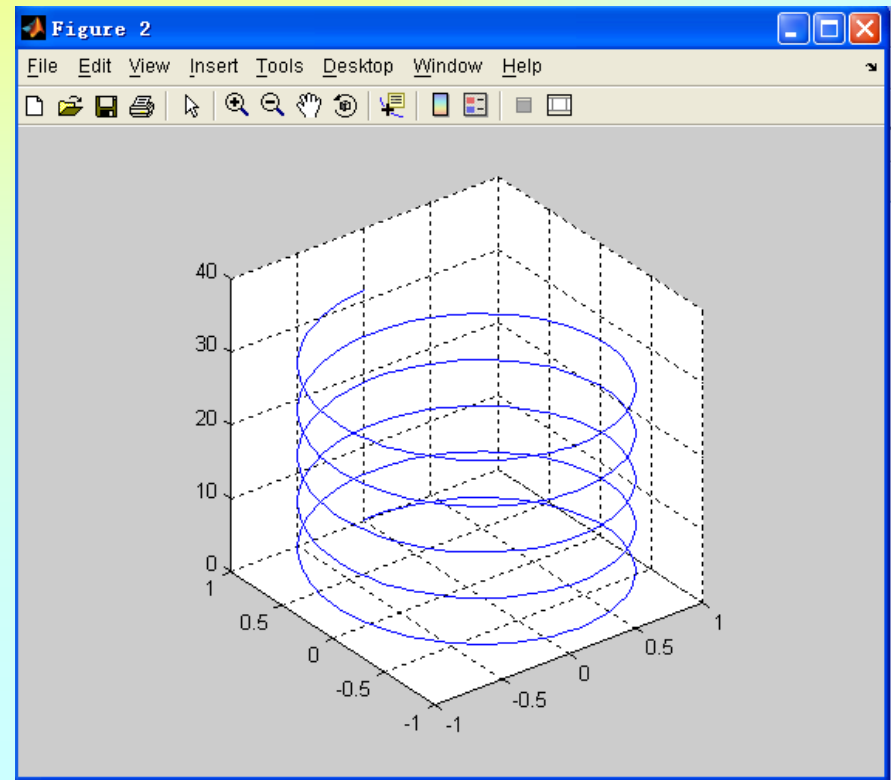
将在三维空间产生一条曲线，线上的点的坐标分别为`x,y,z`的相应元素。

```
t=0:pi/50:10*pi;
```

```
plot3(sin(t),cos(t),t)
```

```
axis square
```

```
grid on
```



三维曲线图的绘制

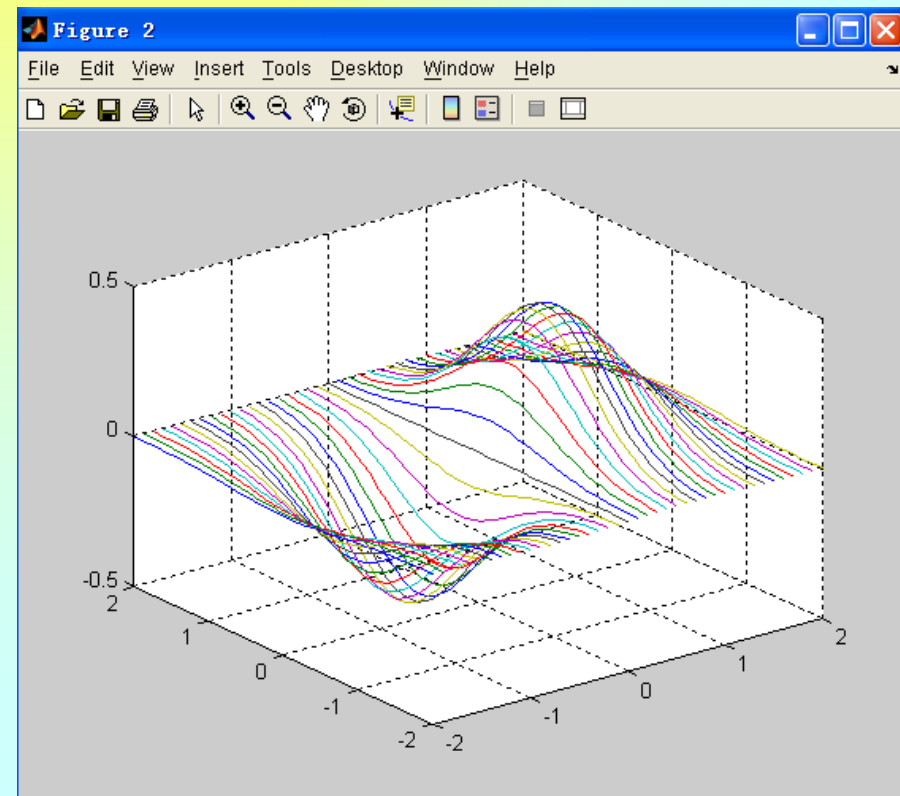
如果plot3函数的参数为3个相同大小的矩阵，则绘制这些矩阵的三维曲线。

```
[X,Y]=meshgrid(-2:0.1:2);
```

```
Z=X.*exp(-X.^2-Y.^2);
```

```
plot3(X,Y,Z)
```

```
grid on
```



创建三维MATLAB曲面图形

mesh函数

mesh函数用于绘制三维网格图。

三维曲面的网格图最**突出的优点**是：它较好地解决了实验数据在三维空间的可视化问题。

函数格式：**mesh(x,y,z,c)**

其中**x**，**y**控制**x**和**y**轴坐标，矩阵**z**是由(**x**，**y**)求得**z**轴坐标，(**x**，**y**，**z**)组成了三维空间的网格点；**c**用于控制网格点颜色。

【例】 下列程序绘制三维网格曲面图

```
x=[0:0.15:2*pi];  
y=[0:0.15:2*pi];  
z=sin(y')*cos(x); 矩阵相乘  
mesh(x,y,z);
```

surf函数

surf用于绘制三维曲面图，各线条之间的补面用颜色填充。surf函数和mesh函数的调用格式一致。

函数格式：**surf (x,y,z)**

其中x，y控制x和y轴坐标，矩阵z是由x，y求得的曲面上z轴坐标。

【例】 下列程序绘制三维曲面图形

```
x=[0:0.15:2*pi];  
y=[0:0.15:2*pi];  
z=sin(y')*cos(x); 矩阵相乘  
surf(x,y,z);  
xlabel('x-axis'),ylabel('y-axis'),zlabel('z-label');  
title('3-D surf');
```

伪彩色图

pcolor: 在平面图中用颜色表示高度，可实现用色彩表达等值线的目的。

函数格式：**pcolor (x,y,z)**

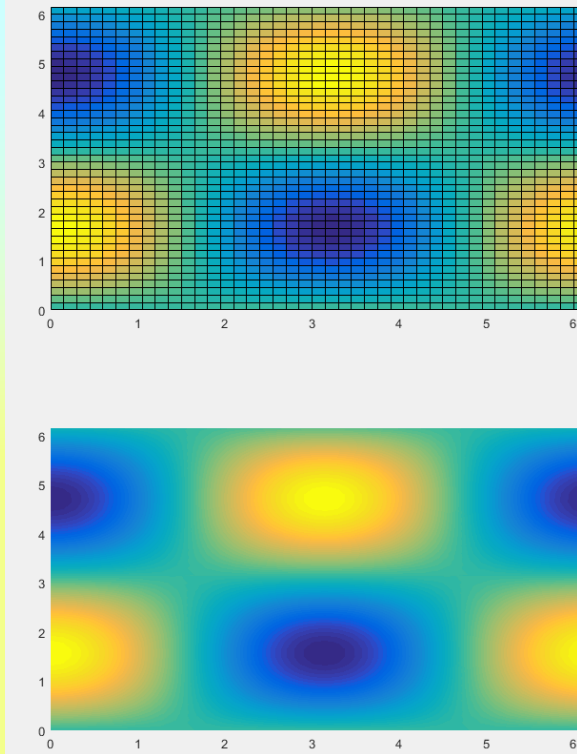
其中**x**，**y**控制**x**和**y**轴坐标，矩阵**z**是由**x**，**y**求得的曲面上**z**轴坐标。

```
x=[0:0.15:2*pi];  
y=[0:0.15:2*pi];  
z=sin(y')*cos(x);  
pcolor(x,y,z);  
shading flat; %(shading interp; shading faceted;)
```

```

x=[0:0.15:2*pi];
y=[0:0.15:2*pi];
z=sin(y')*cos(x);
subplot(2,2,1)
pcolor(x,y,z);
subplot(2,2,2)
pcolor(x,y,z);
shading flat
subplot(2,2,3)
pcolor(x,y,z);
shading interp
subplot(2,2,4)
pcolor(x,y,z);
shading faceted

```



% 颜色用四个顶点数据点下标最小点的颜色

% 颜色用四个顶点处的颜色线性插值

% 在flat基础上，或黑色网线

等高线图

等高线图可通过函数contour或contourf
contour3绘制。

【例】 多峰函数peaks的等高线图

```
[x,y,z]=peaks(30);  
contour3(x,y,z,16);  
xlabel('x-axis'),ylabel('y-axis'),zlabel('z-  
axis');  
title('contour3 of peaks')
```

```
x=[0:0.15:2*pi];  
y=[0:0.15:2*pi];  
z=sin(y')*cos(x);
```

```
subplot(2,2,1)
```

```
contour(x,y,z)
```

```
n=6
```

```
subplot(2,2,2)
```

```
contour(x,y,z,n,'k:')
```

```
subplot(2,2,3)
```

```
c=contourf(x,y,z);
```

```
clabel(c)
```

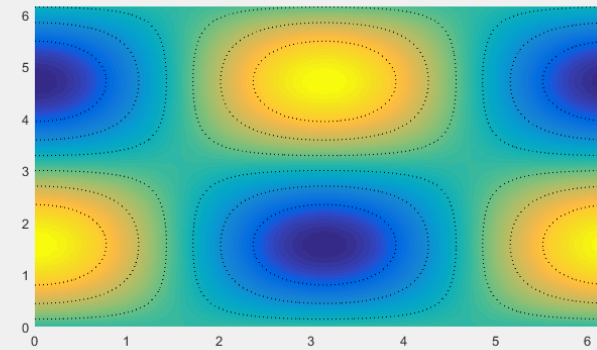
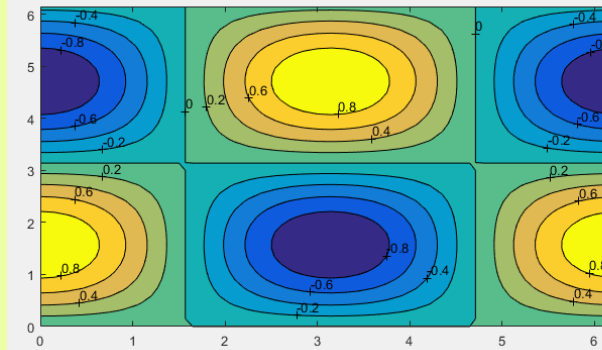
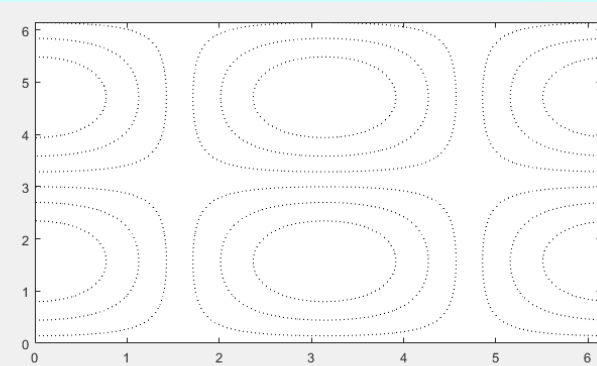
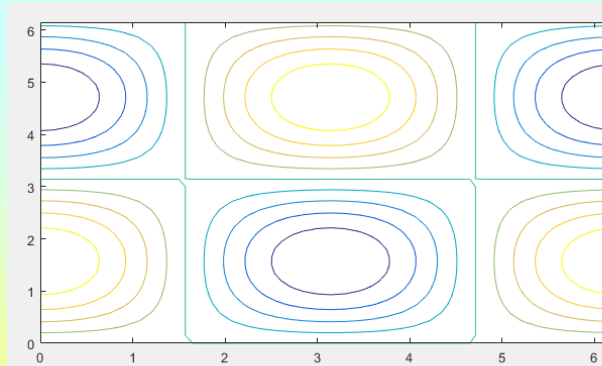
```
subplot(2,2,4)
```

```
pcolor(x,y,z)
```

```
shading interp
```

```
hold on
```

```
contour(x,y,z,n,'k:')
```



常用网格和表面图形的绘制

```
[X,Y]=meshgrid(-8:.5:8);
```

创建矩阵X和矩阵Y

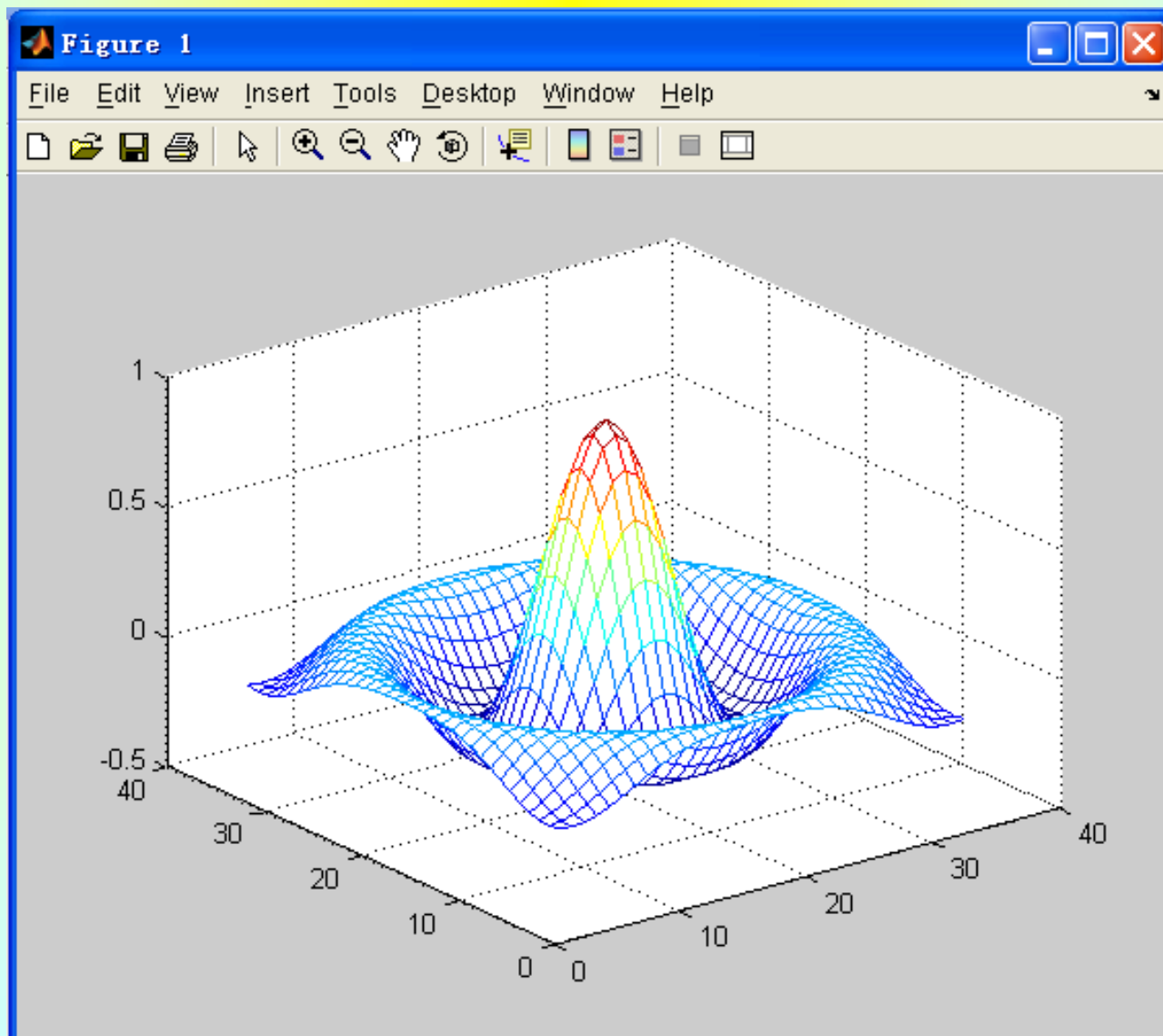
```
R=sqrt(X.^2+Y.^2)+eps;
```

加上eps是为了避免R为0.

```
Z=sin(R)./R;
```

```
figure
```

```
mesh(Z)
```



表面图形的透明处理

默认情况，MATLAB将自动隐藏。表面图形都是实心的，即不透明的。可以用hidden off 指令将图形表面作透明处理。

```
[X,Y]=meshgrid(-8:.5:8);
```

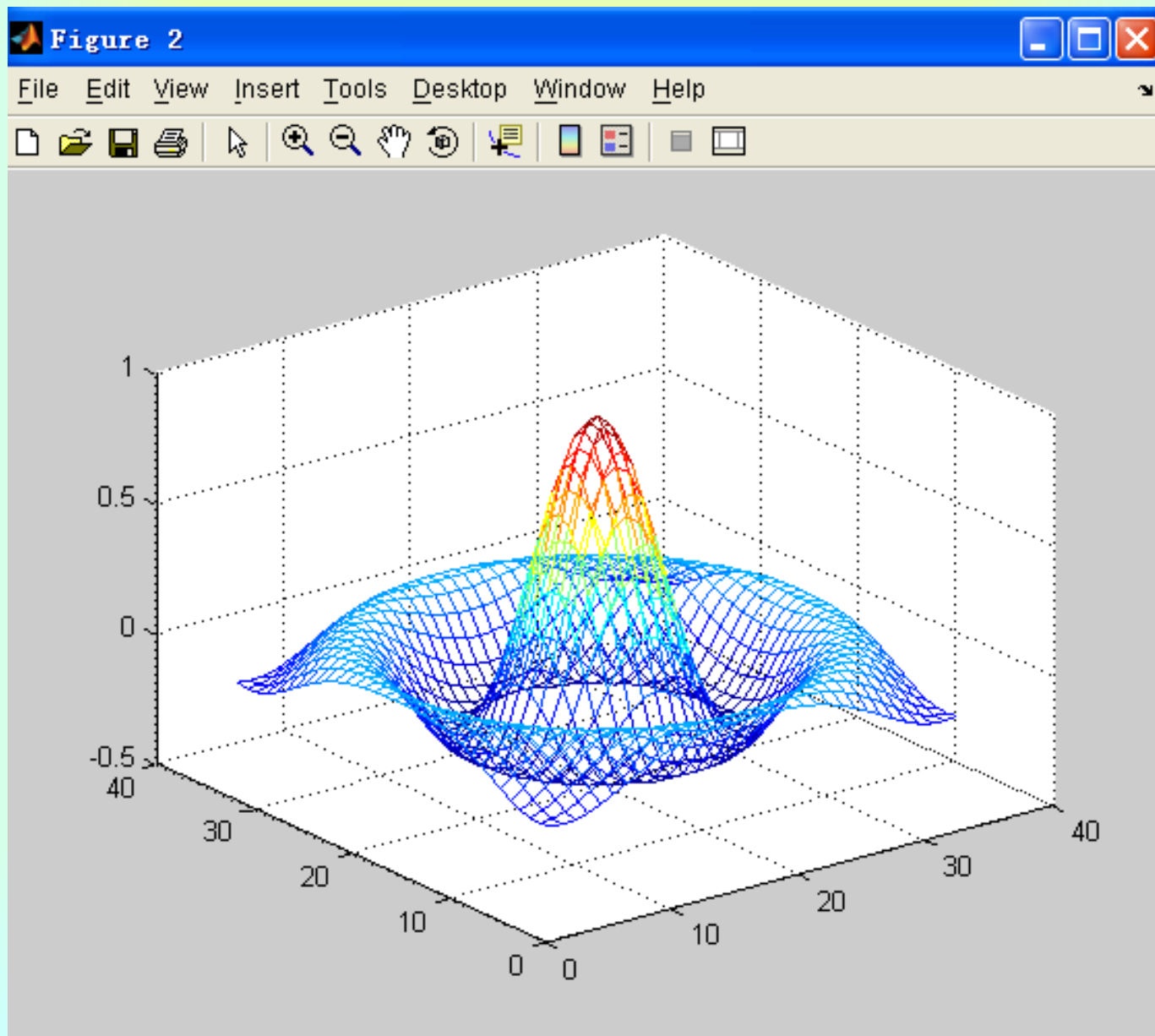
```
R=sqrt(X.^2+Y.^2)+eps;
```

```
Z=sin(R)./R;
```

```
figure
```

```
mesh(Z);
```

```
hidden off
```



颜色映射表

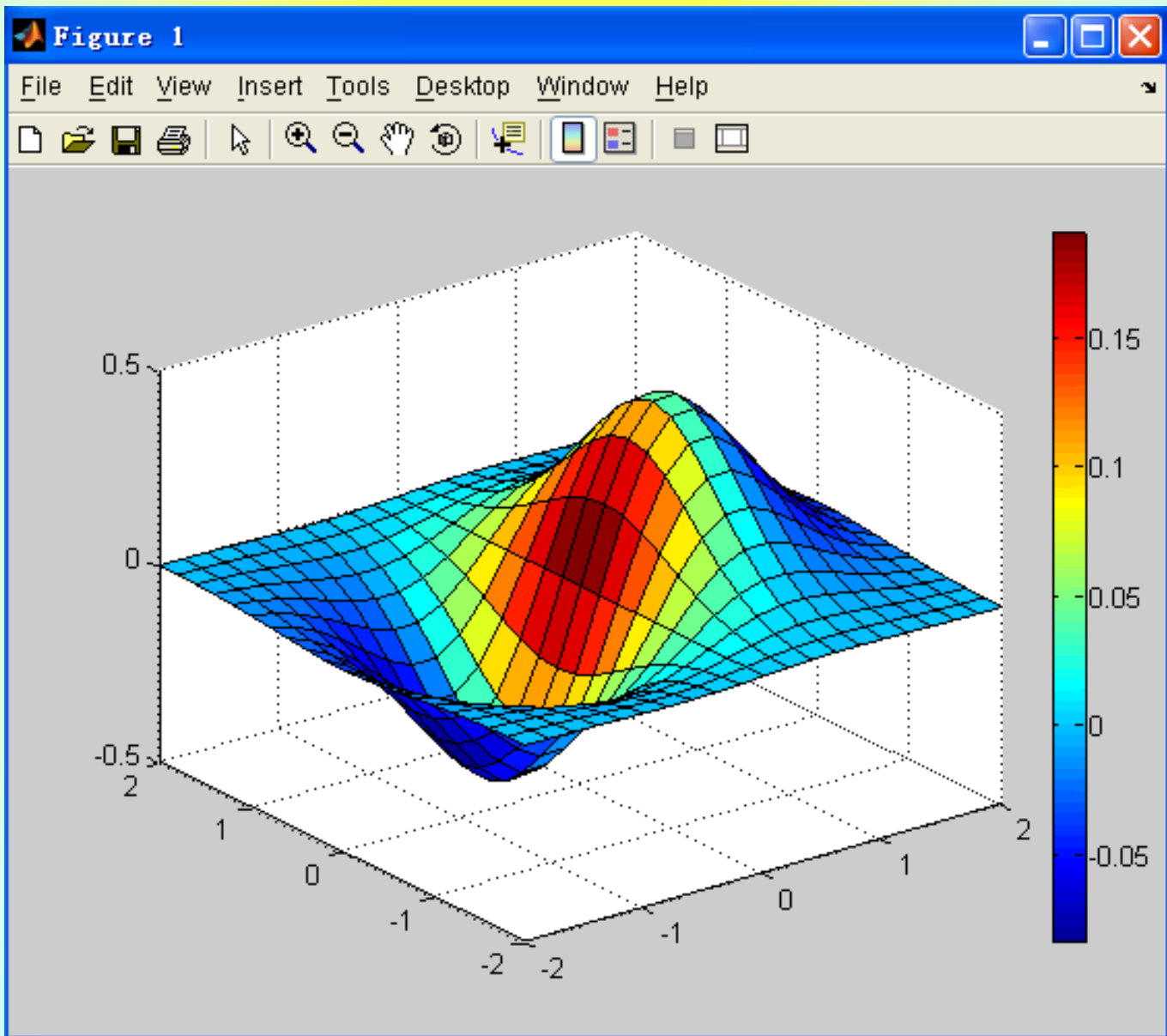
MATLAB中的colorbar函数用于显示当前的颜色映射表。

```
[x,y]=meshgrid([-2:.2:2]);
```

```
Z=x.*exp(-x.^2-y.^2);
```

```
surf(x,y,Z,Gradient(Z))
```

```
colorbar
```



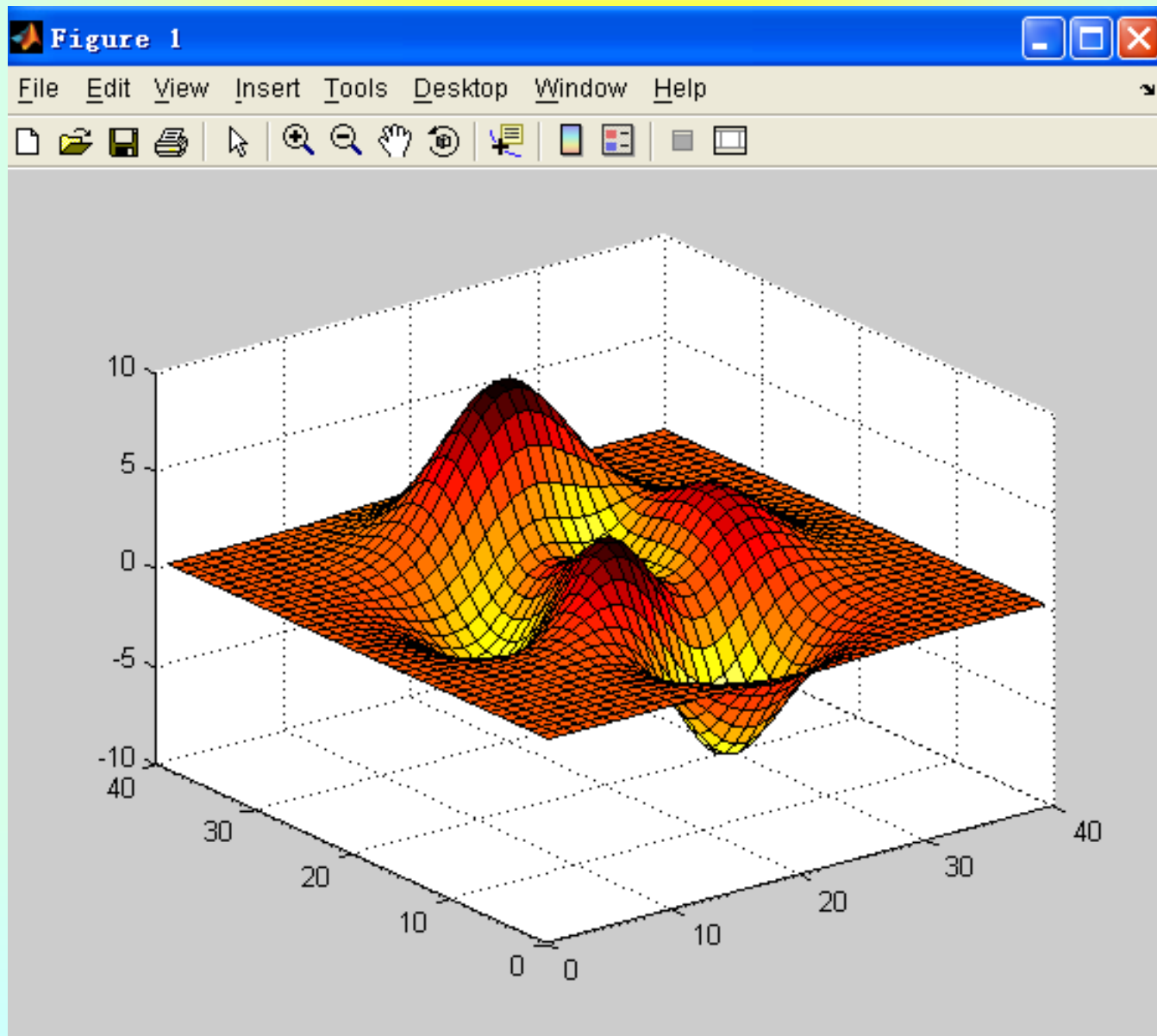
表面曲率的颜色映射

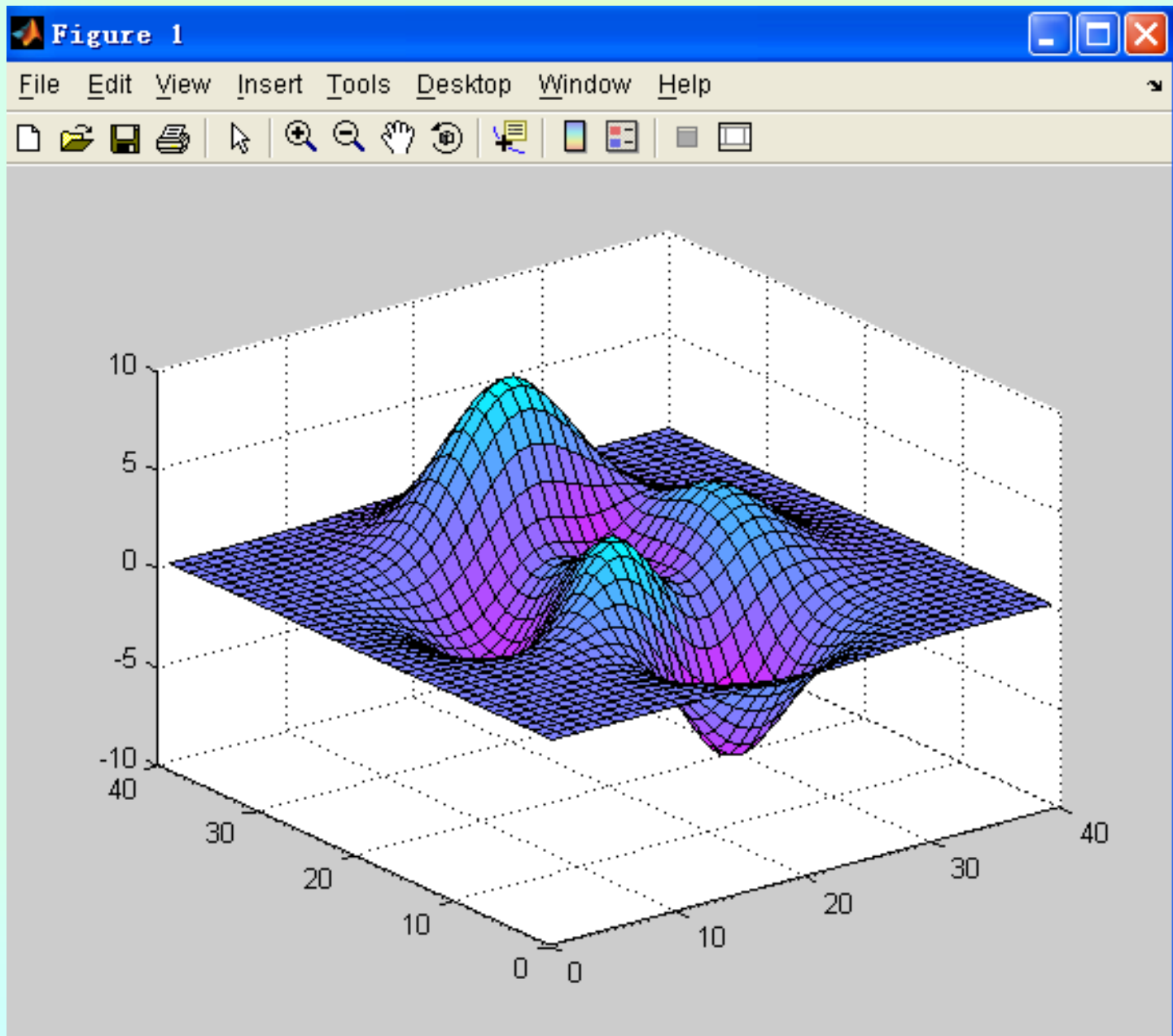
```
P=peaks(40);
```

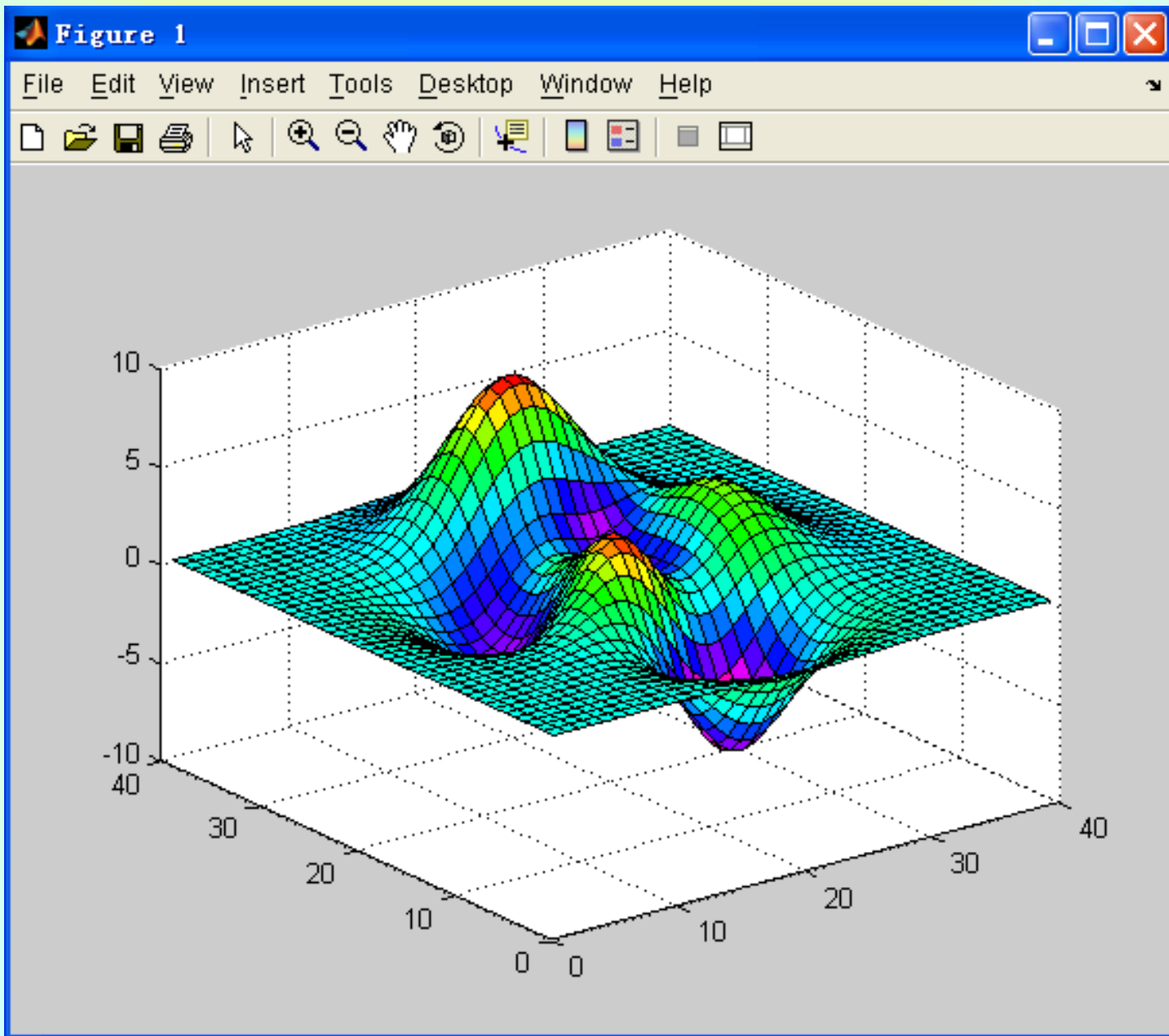
```
surf(P)
```

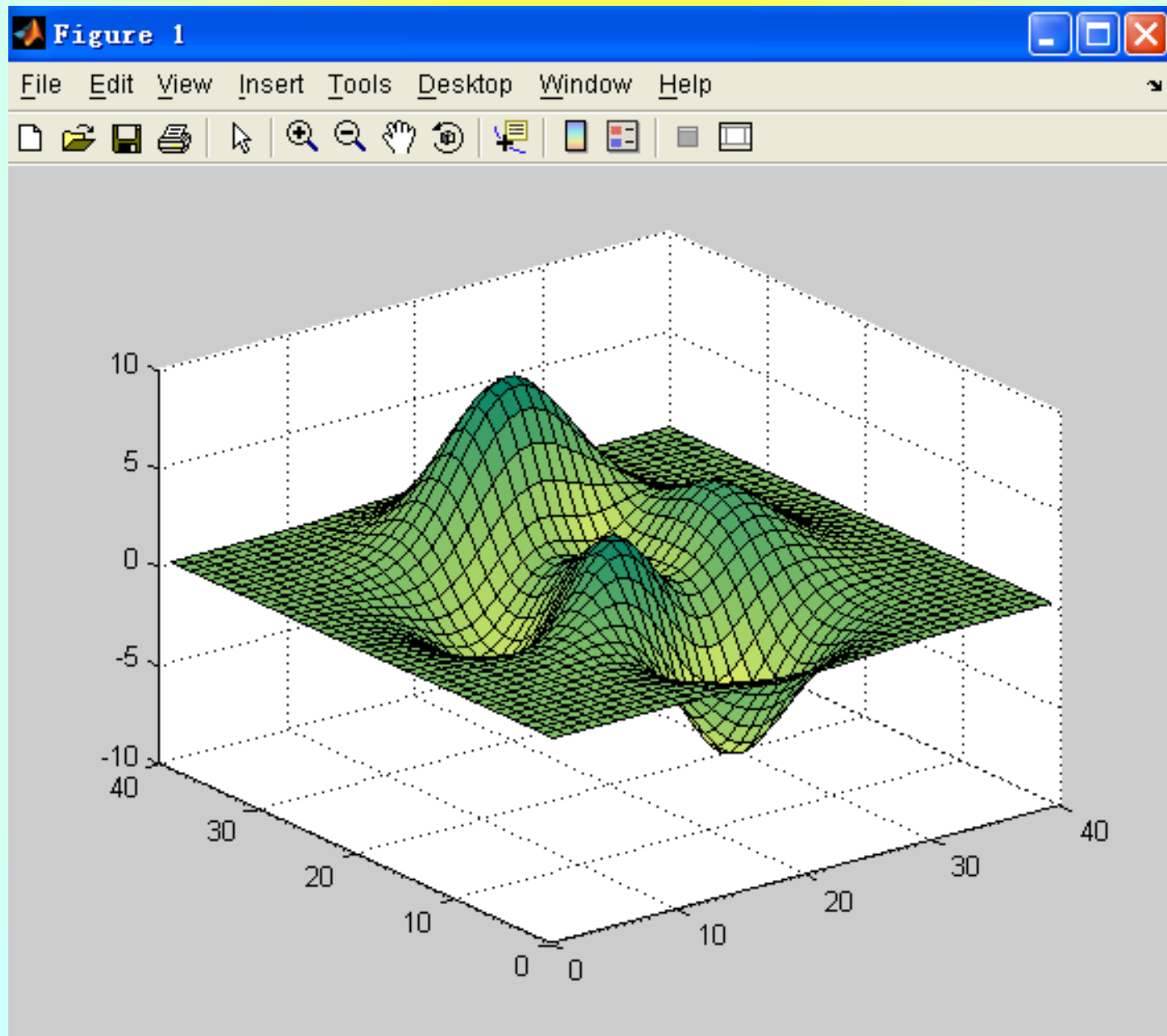
```
colormap hot
```

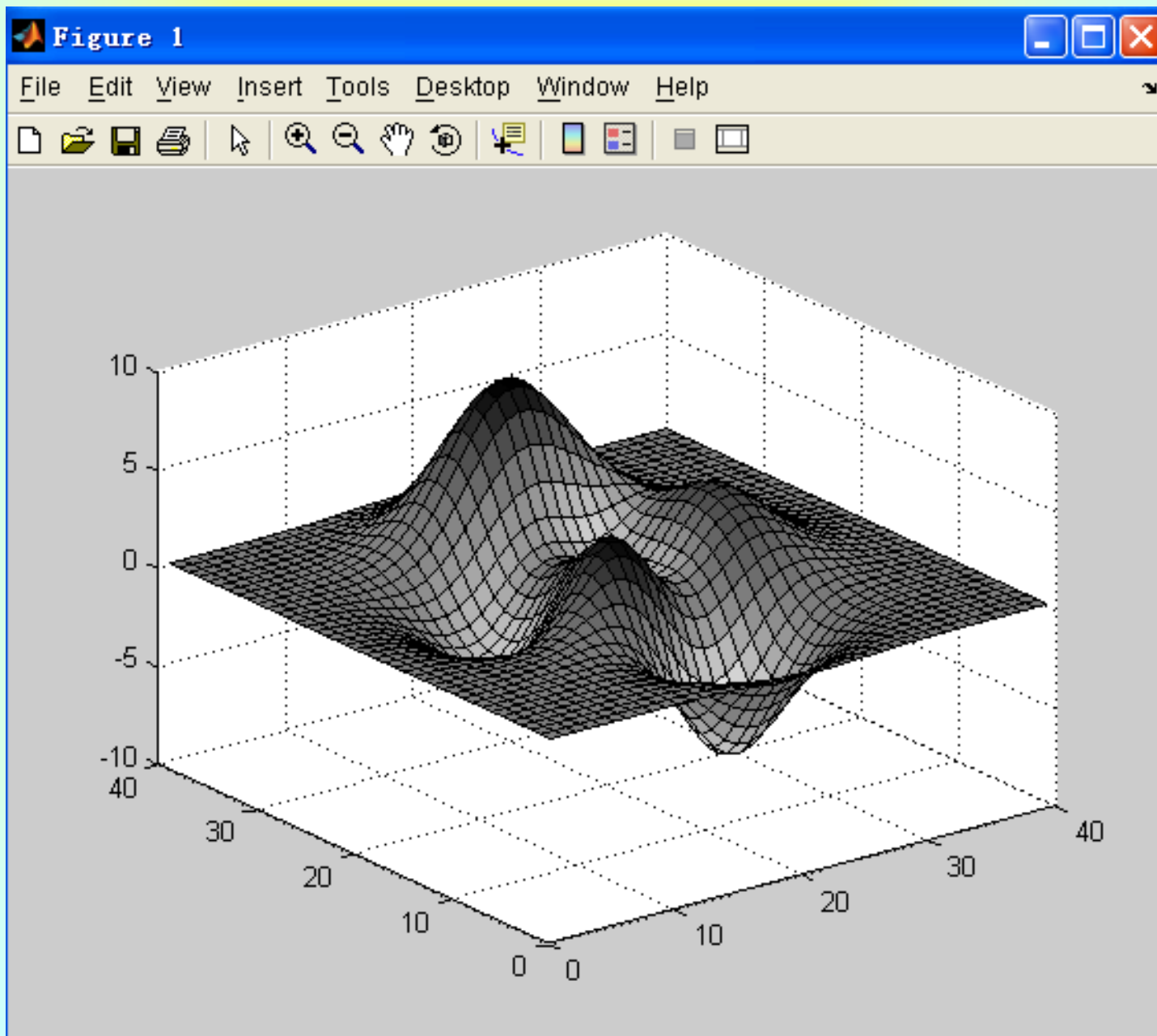
另外还创建有其他颜色映射表，如：hsv、hot、cool、summer、gray等











视点的变换

MATLAB用view命令指定视点相对于坐标原点的水平转角和仰角。

默认情况下：

二维图形的水平转角为0度，仰角为90度。

三维图形的水平转角为-37.5度，仰角为30度。

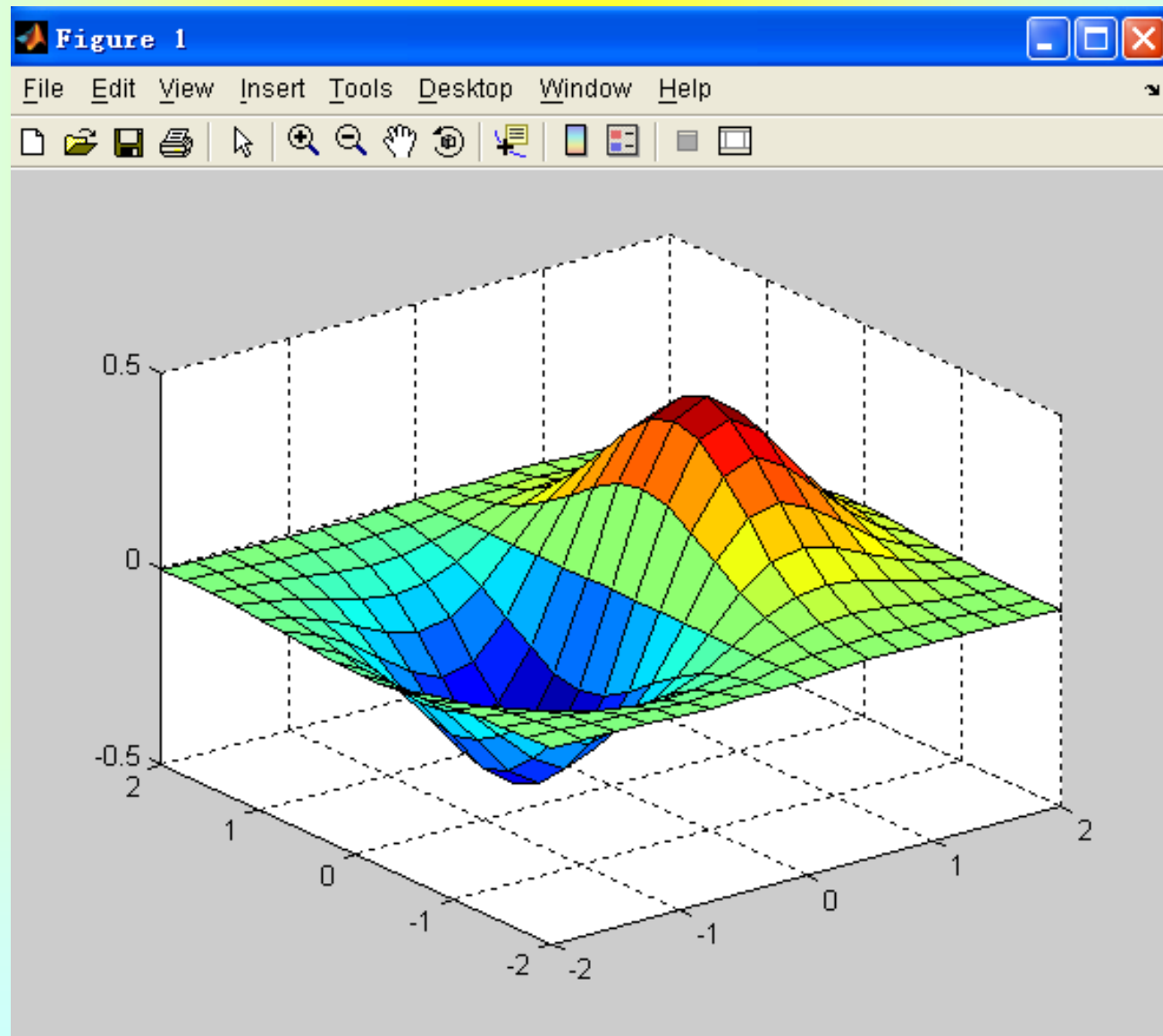
视点的变换

在默认的视点下绘制的图形：

```
[X,Y]=meshgrid([-2:.25:2]);
```

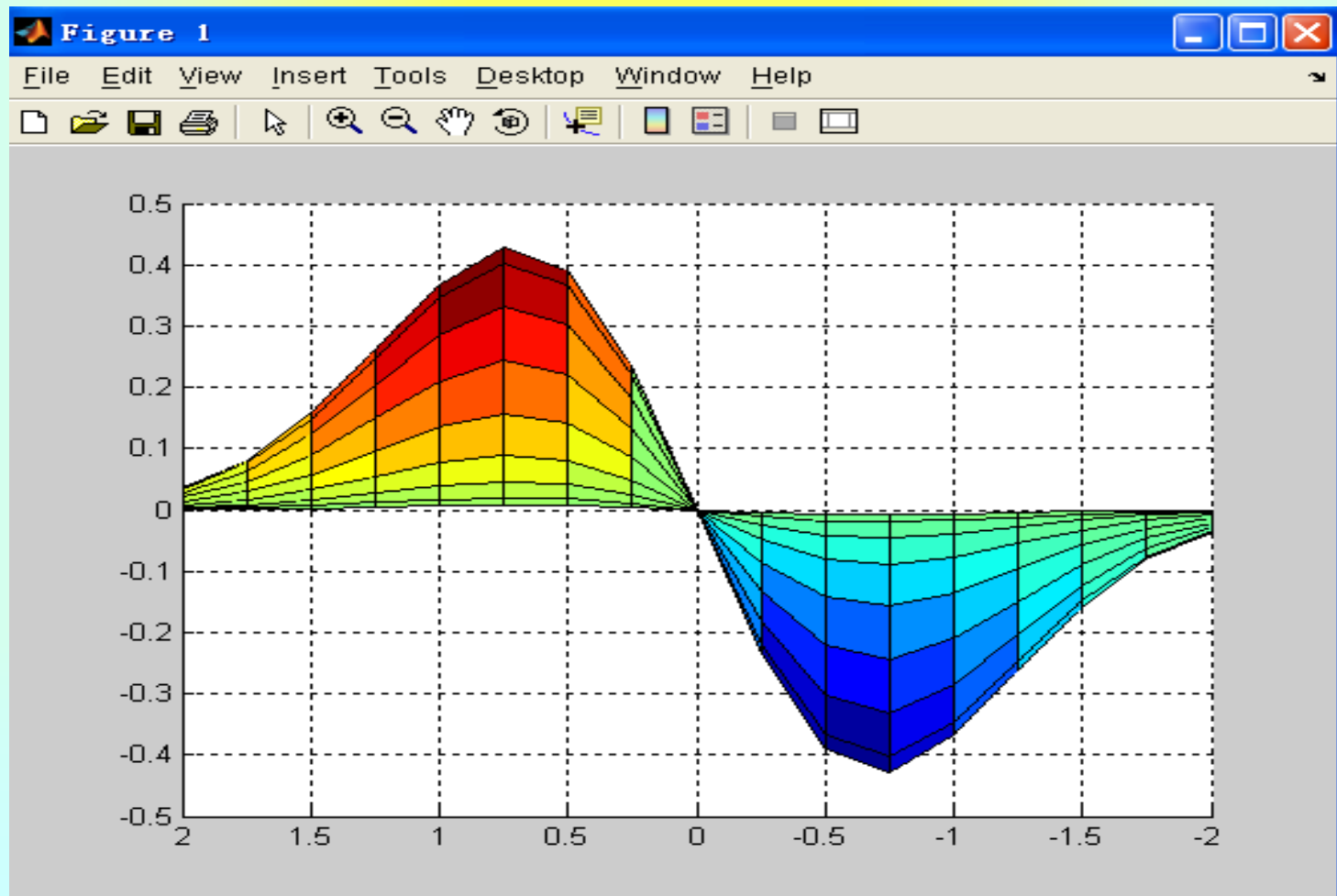
```
Z=X.*exp(-X.^2-Y.^2);
```

```
surf(X,Y,Z)
```

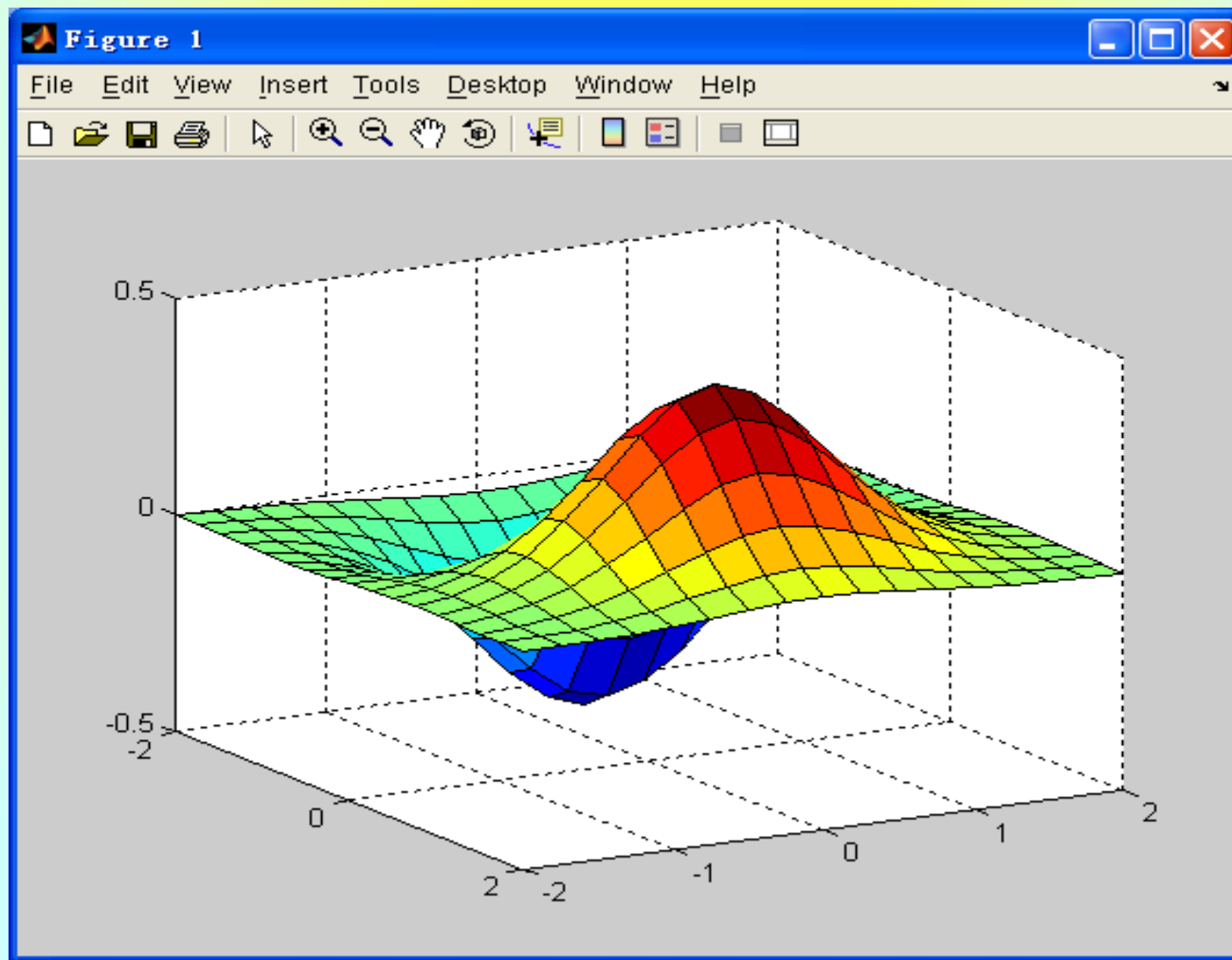
视点的变换

`view([180 0])`



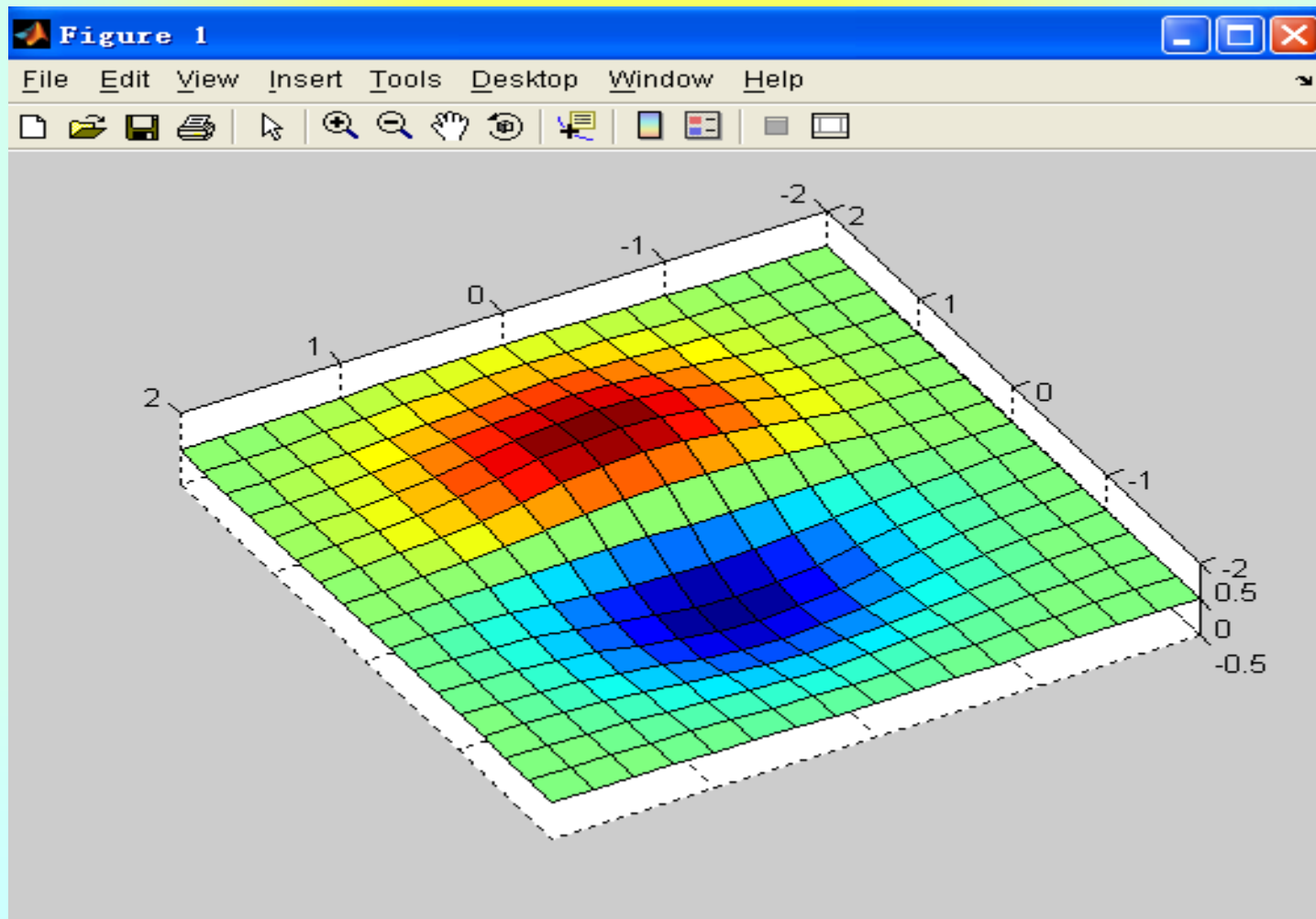
视点的变换

`view([60 20])`



视点的变换

`view([60 100])`



MATLAB中的灯光命令

MATLAB中的灯光对象

light函数可以创建灯光对象，其中：

Color属性用于指定灯光对象的灯光颜色。

Style属性用于指定光源类型:无限远（默认）或本地。

Position属性用于指定光线方向（无限远光源）或光源位置(本地光源)。

向图像场景添加灯光效果

没有灯光的表面图形

利用membrane函数绘制的表面图形：

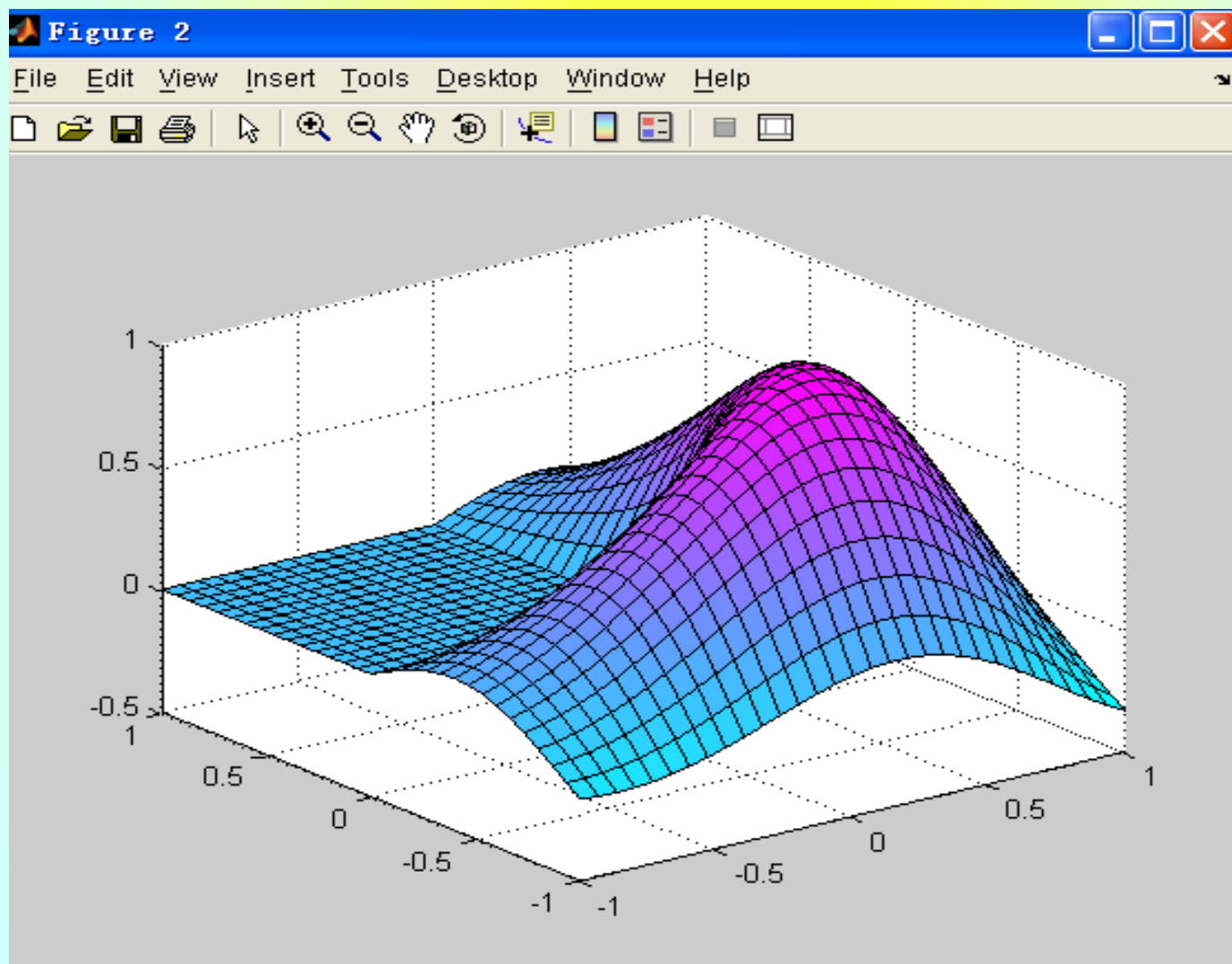
membrane （生成matlab的logo函数）

添加灯光1： `light('position',[0 -2 1])`

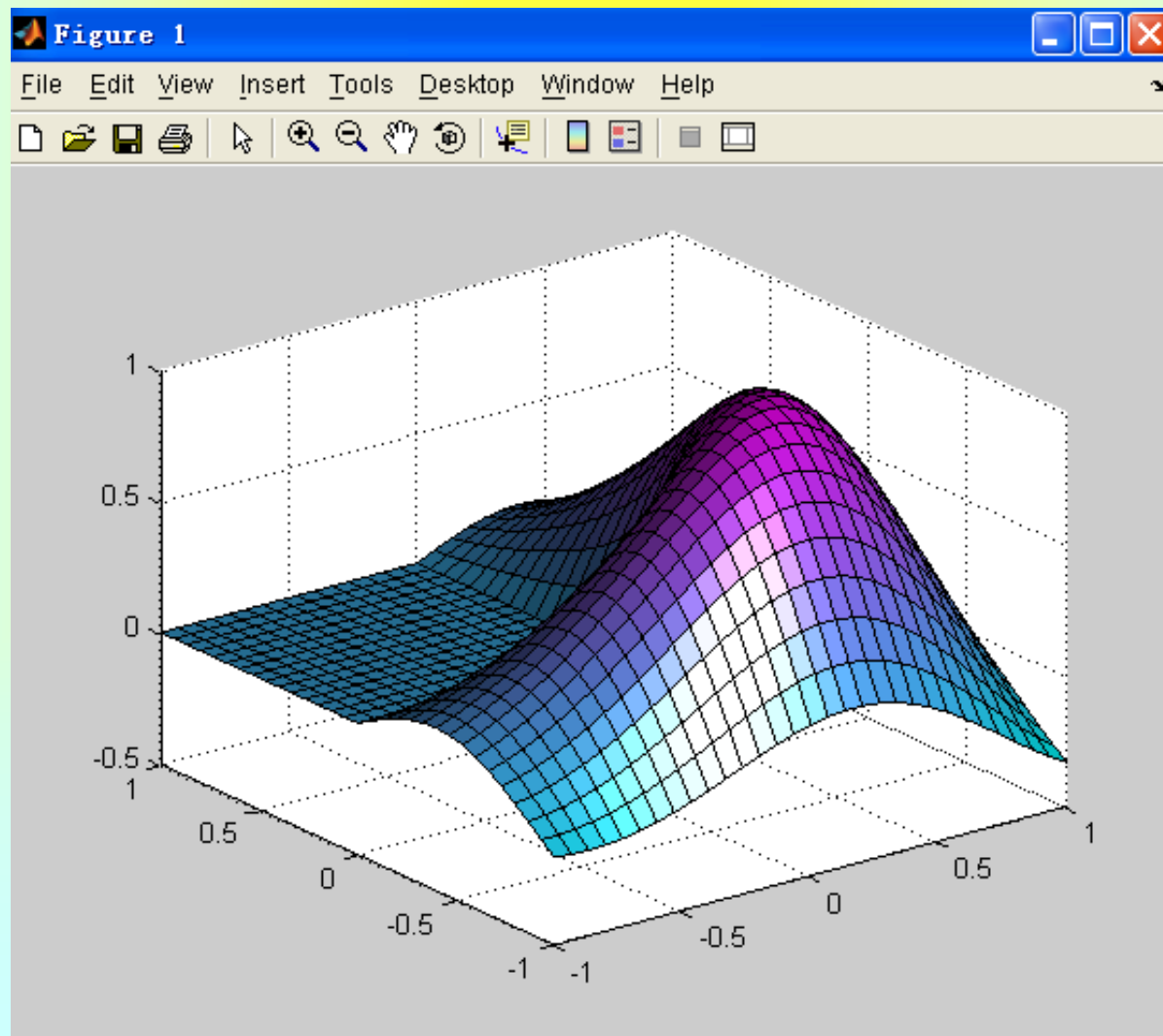
添加灯光2： `light('position',[1 0 1])`

添加灯光3： `light('position',[2 2 10])`

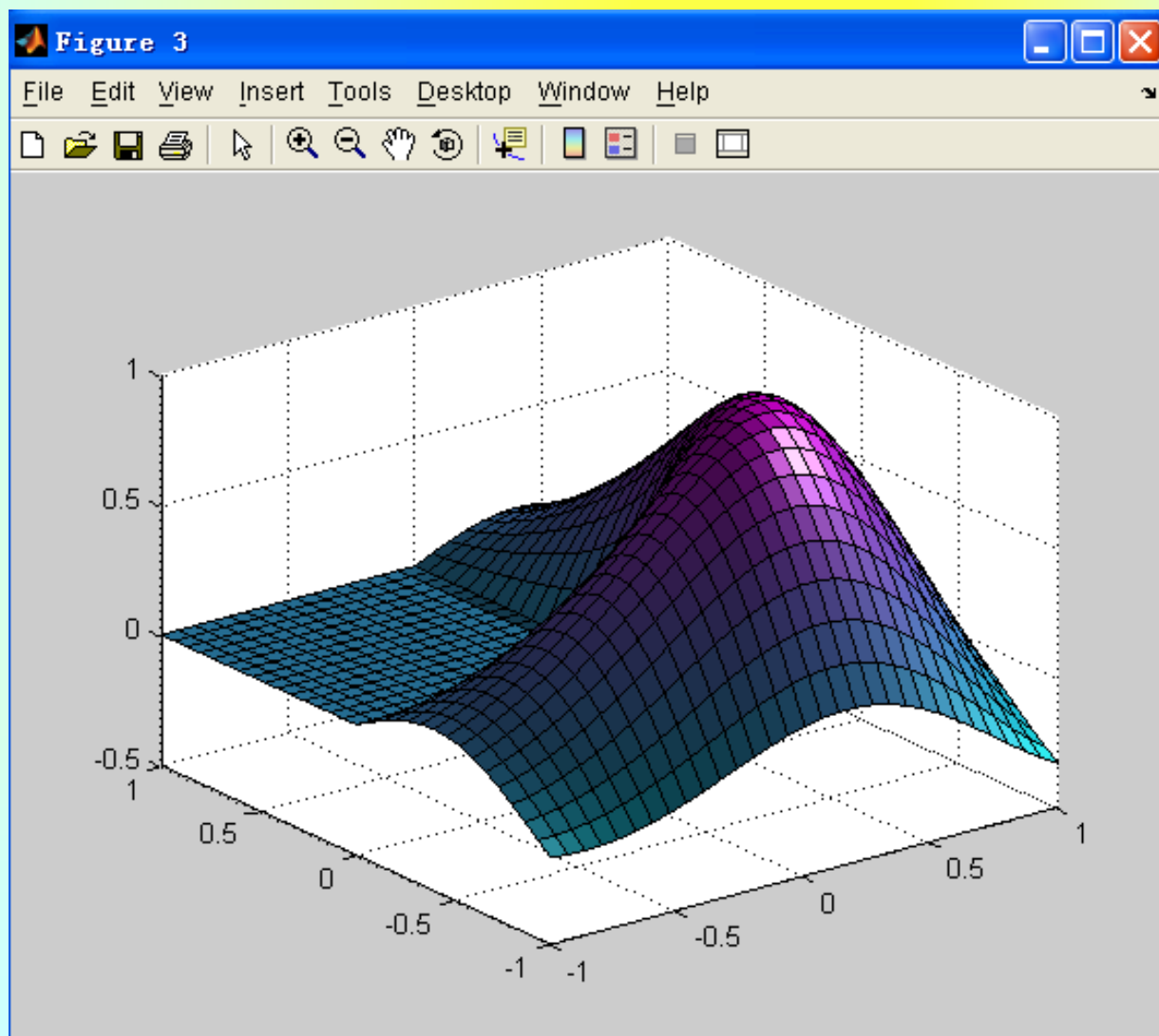
无灯光效果1



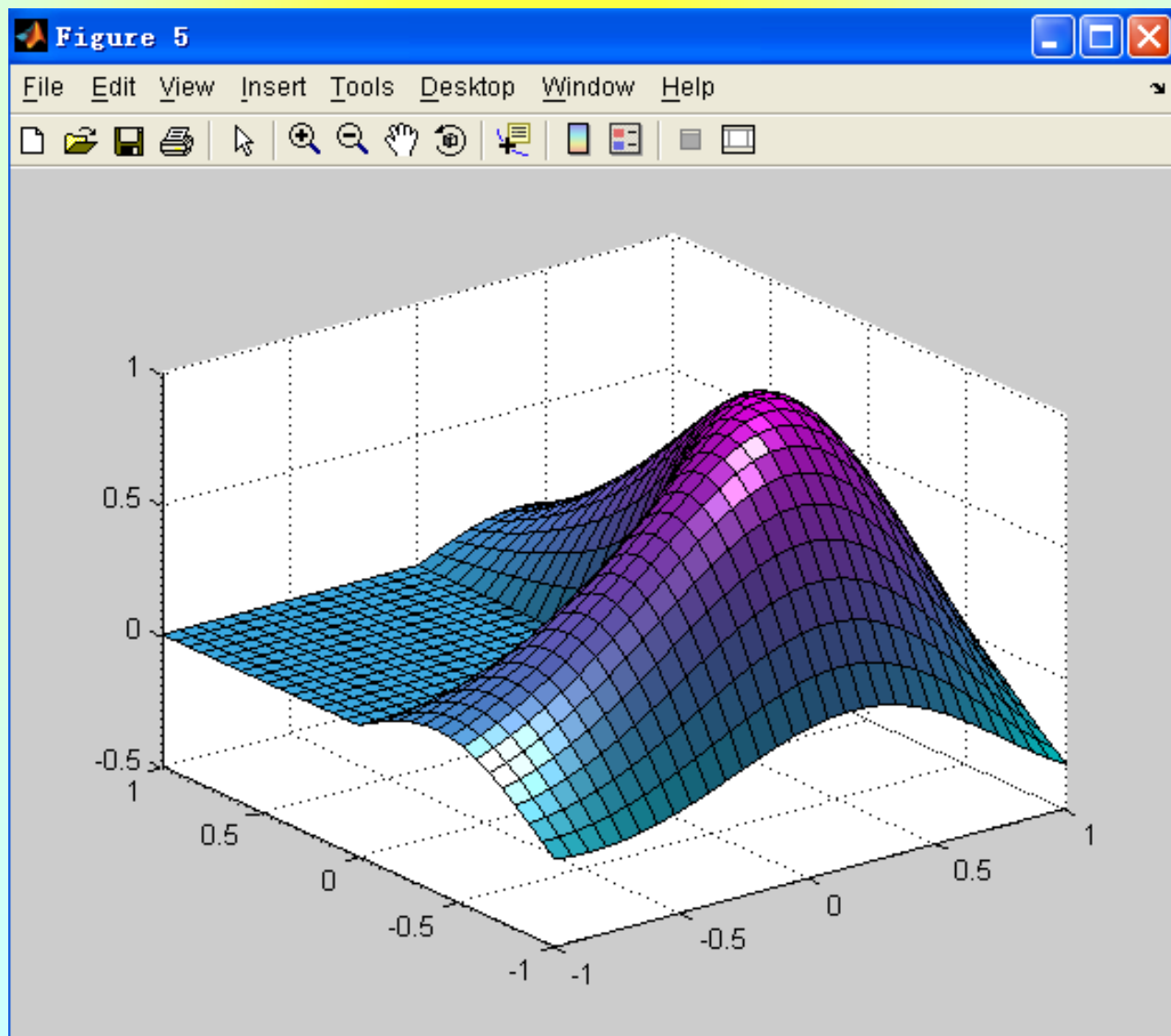
向图像场景添加灯光效果1



向图像场景添加灯光效果2

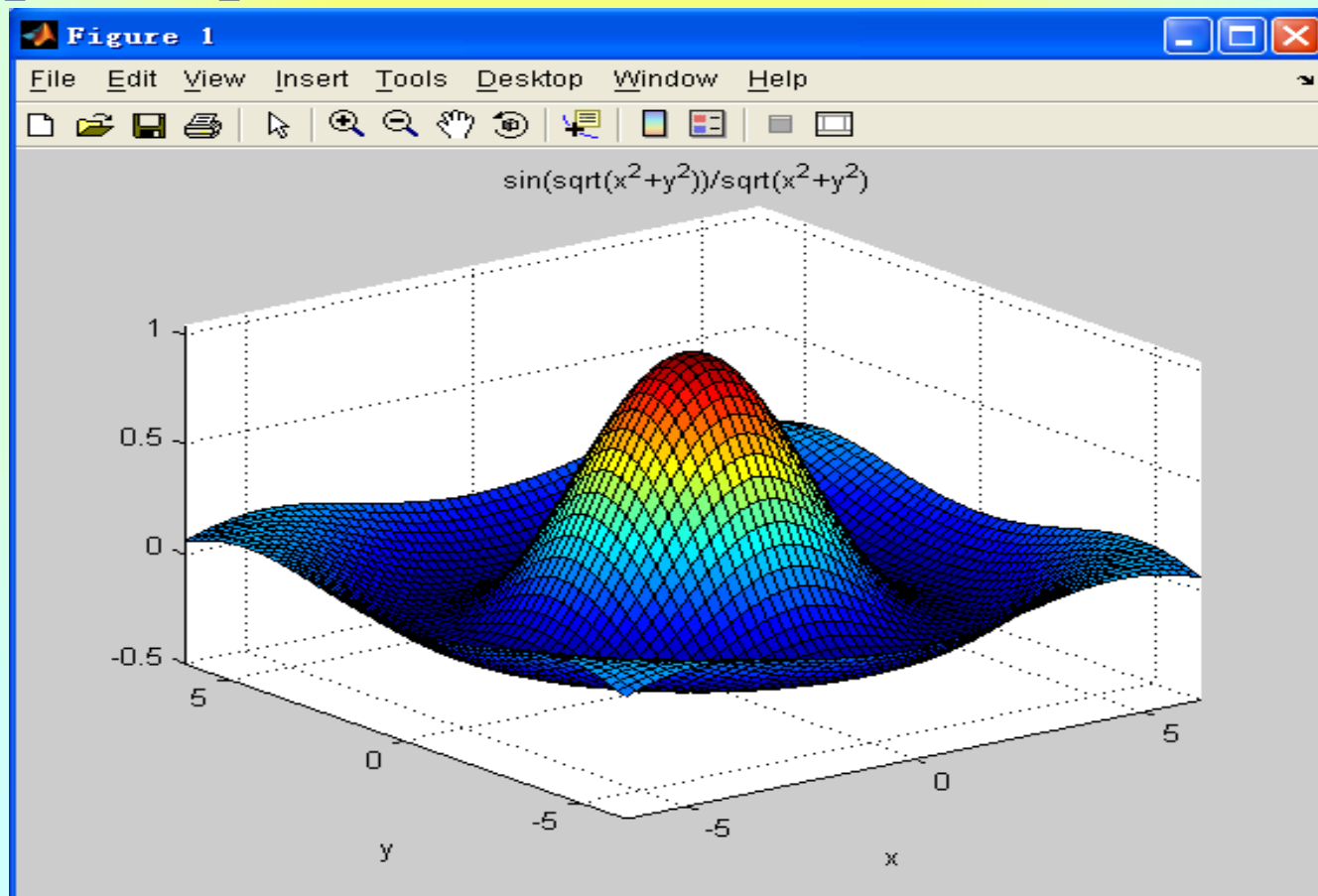


向图像场景添加灯光效果3



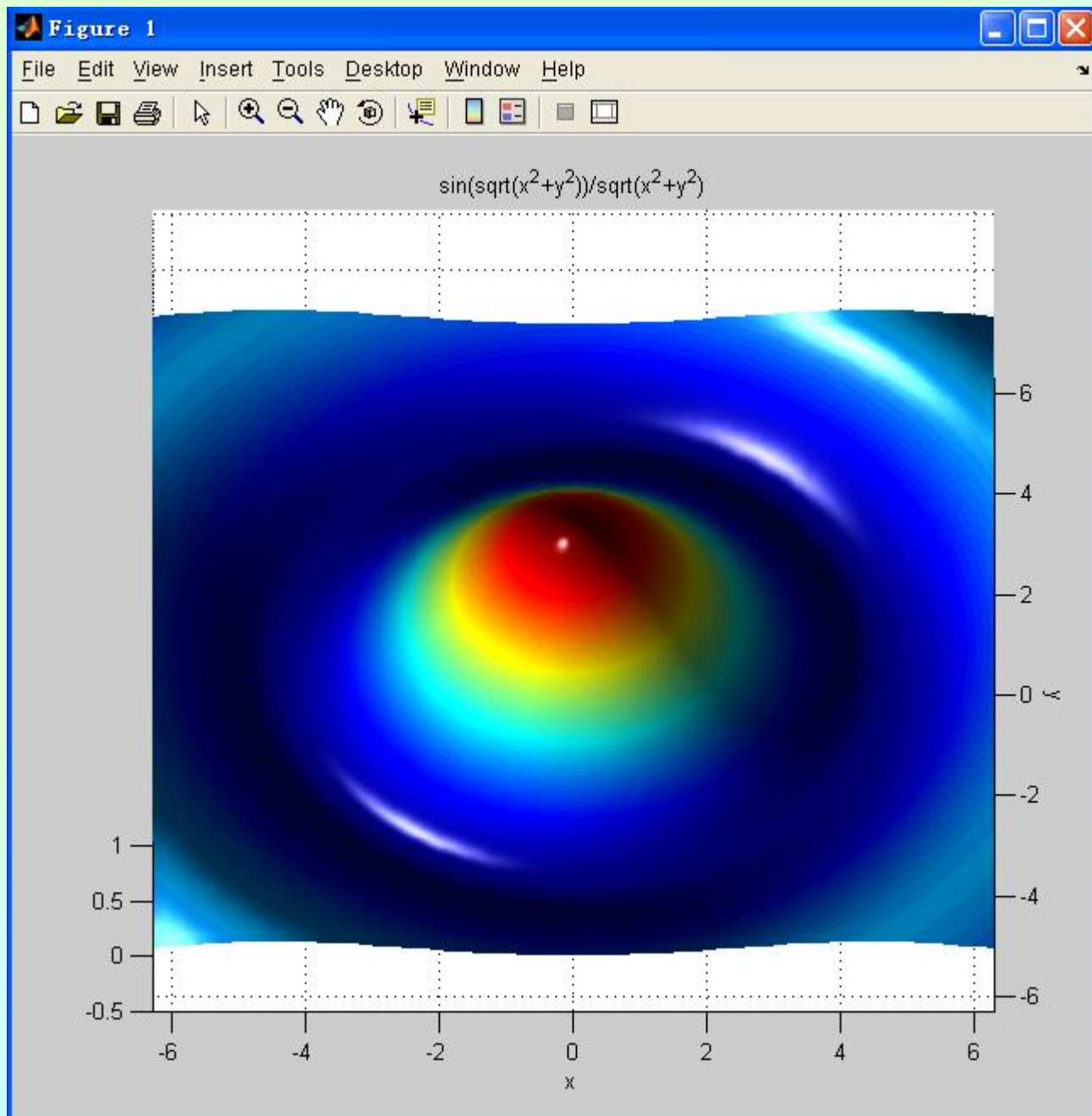
无灯光效果2

`ezsurf('sin(sqrt(x^2+y^2))/sqrt(x^2+y^2)'), ...`
`[-60*pi,60*pi]`



向图像场景添加灯光效果

```
ezsurf('sin(sqrt(x^2+y^2))/sqrt(x^2+y^2)'),[-  
6*pi,6*pi]  
view(0,75)  
shading interp  
lightangle(-45,30)  
set(findobj(gca,'type','surface'),...  
    'FaceLighting','phong',...  
    'AmbientStrength',.3,'DiffuseStrength',.8,...  
    'SpecularStrength',.9,'SpecularExponent',25,...  
    'BackFaceLighting','unlit')
```



函数 $f(x)$ 曲线

fplot函数则可自适应地对函数进行采样，能更好地反应函数的变化规律。

fplot函数格式： **fplot(fname, lims, tol)**

其中**fname**为函数名，以字符串形式出现，**lims**为变量取值范围，**tol**为相对允许误差，其系统默认值为 $2e-3$ 。

例： **fplot('sin',[0 2*pi],'-+')**

fplot('[sin(x),cos(x)]',[0 2*pi],1e-3,'·') 同时绘制正弦、余弦曲线

为绘制 $f(x)=\cos(\tan(\pi x))$ 曲线，可先建立函数文件 `fct.m`，其内容为：

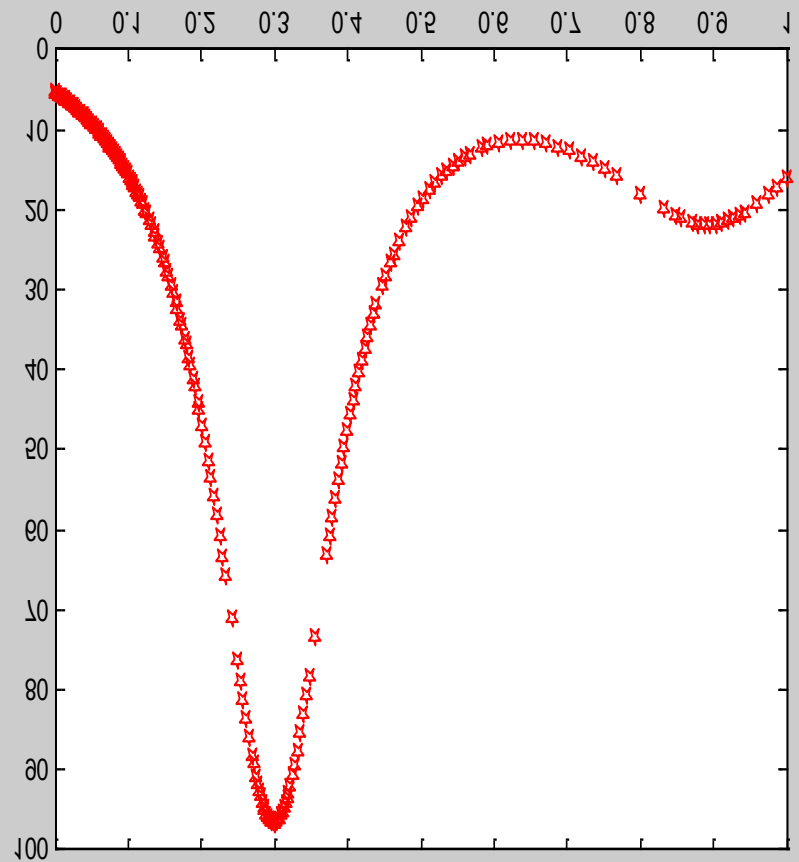
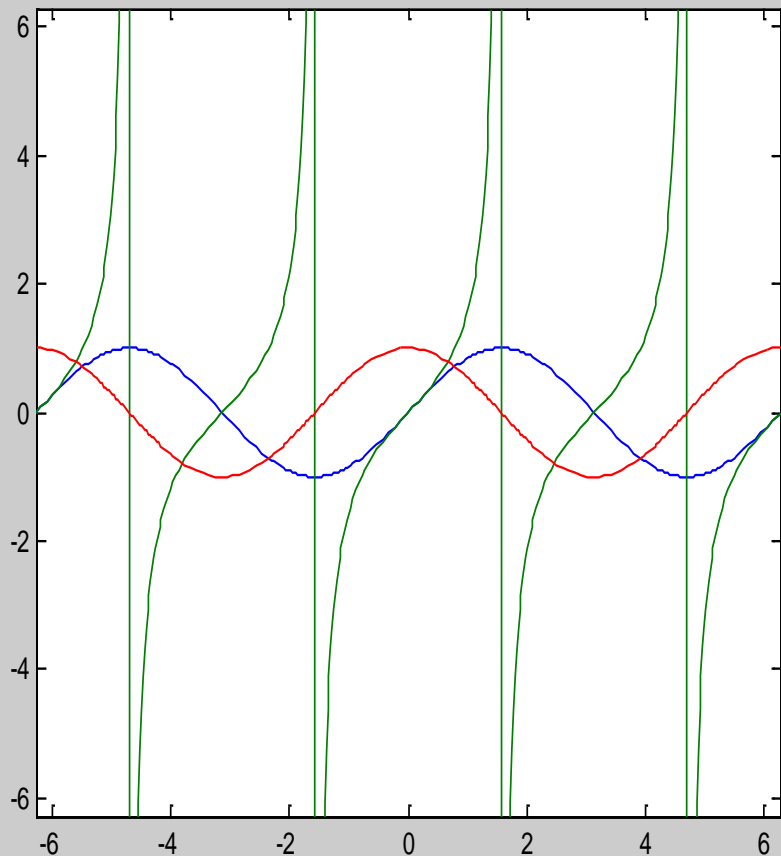
```
function y=fct(x)
```

```
    y=cos(tan(pi*x));
```

用 `fplot` 函数调用 `fct.m` 函数，其命令为：

```
fplot(@fct,[0 1],1e-7)
```

```
fplot('[sin(x),tan(x),cos(x)]',2*pi*[-1 1 -1 1])  
fplot(@(x)[sin(x),tan(x),cos(x)],2*pi*[-1 1])  
fplot('humps',[0 1],'rp')  
fplot('sin(x)',[0 2*pi],'*')
```



ezplot ——符号函数绘图(隐函数)

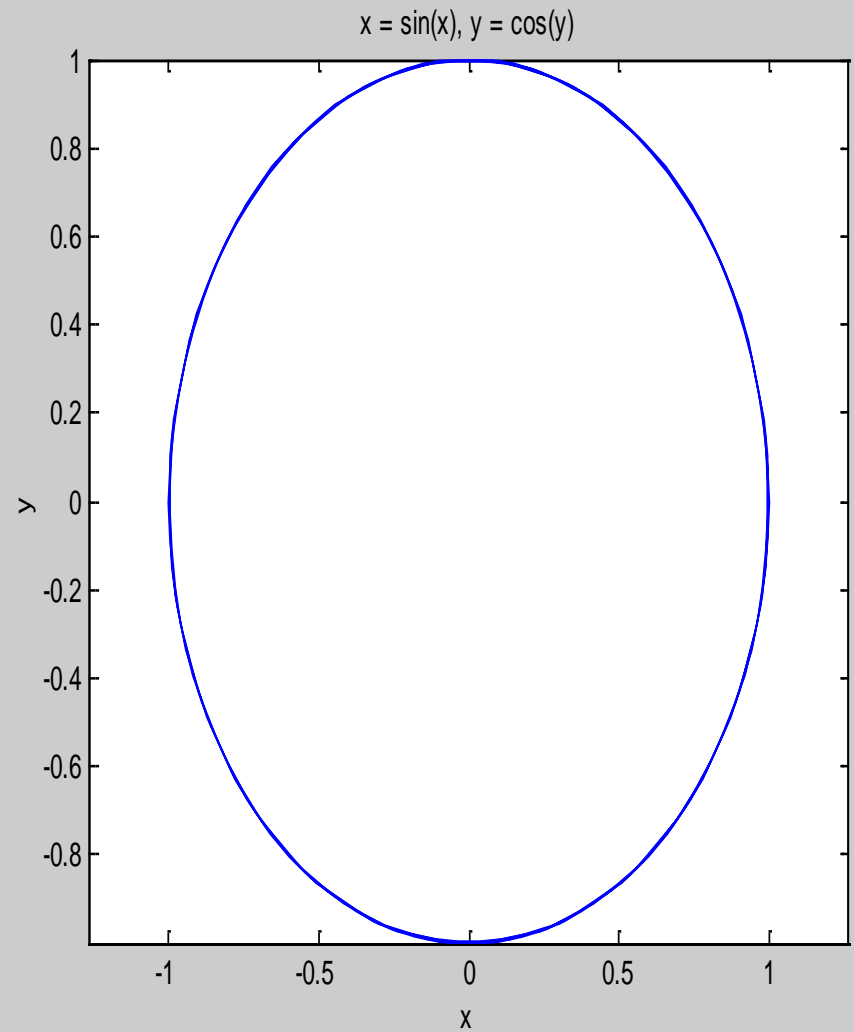
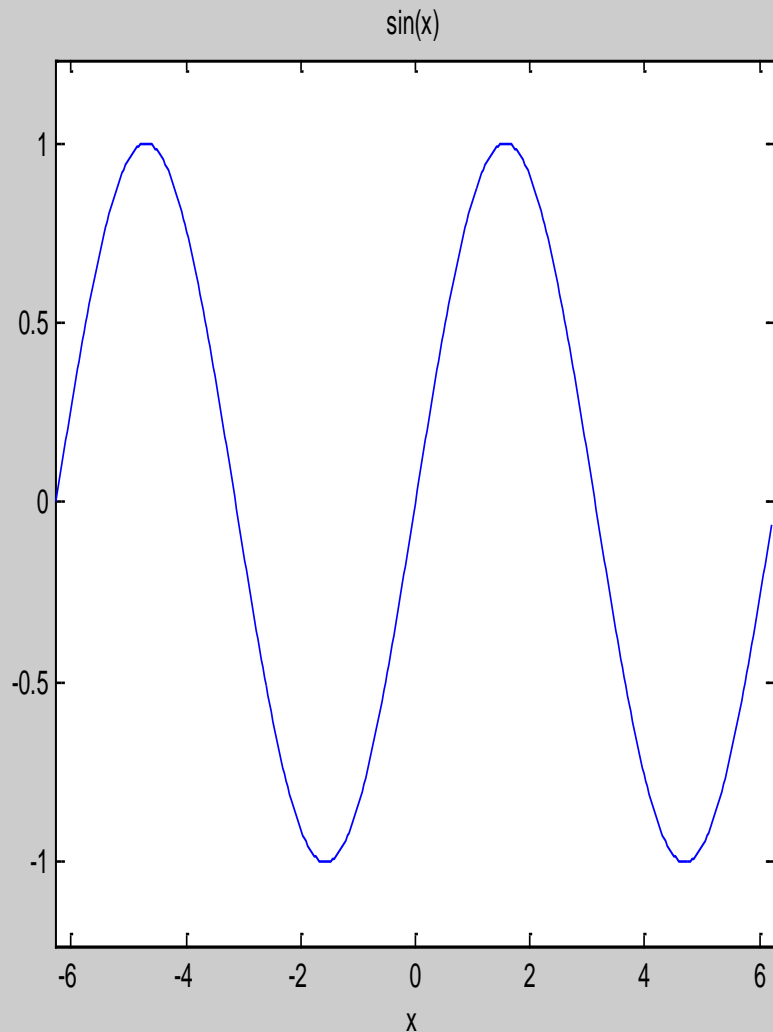
ezplot(f) —这里f为包含单个符号变量
x的符号表达式（不易求解的），在x
轴的默认范围

$[-2\pi, 2\pi]$ 内绘制f(x)的函数图

ezplot(f,xmin,xmax) — 给定区间

ezplot(f,[xmin,xmax],figure(n)) — 指定
绘图窗口绘图。

```
ezplot('sin(x)')      ezplot('x^2-y^2-1')  
ezplot('sin(x)','cos(y)',[-4*pi 4*pi],figure(2))
```



特殊坐标图形

一、对数坐标图形

(一) `loglog(x,y)` 双对数坐标

【例9】 绘制 $y=|1000\sin(7x)|+1$ 的双对数坐标图。程序为：

```
x=[0:0.1:2*pi];
```

```
y=abs(1000*sin(7*x))+1;
```

```
loglog(x,y); %双对数坐标绘图命令
```

(二) 单对数坐标

以X轴为对数重新绘制上述曲线，程序为：

```
x=[0:0.01:2*pi]
```

```
y=abs(1000*sin(7*x))+1
```

```
semilogx(x,y); 单对数x轴绘图命令
```

同样，可以以Y轴为对数重新绘制上述曲线，程序为：

```
x=[0:0.01:2*pi]
```

```
y=abs(1000*sin(7*x))+1
```

```
semilogy(x,y); 单对数y轴绘图命令
```

二、极坐标图

函数**polar(theta,rho)**用来绘制极坐标图，**theta**为极坐标角度，**rho**为极坐标半径

【例10】 绘制 $\sin(2*\theta)*\cos(2*\theta)$ 的极坐标图，程序为：

```
theta=[0:0.01:2*pi];  
rho=sin(2*theta).*cos(2*theta);  
polar(theta,rho); 绘制极坐标图命令  
title('polar plot');
```

其它图形函数

除plot等基本绘图命令外，Matlab系统提供了许多其它特殊绘图函数，这里举一些代表性例子，更详细的信息用户可随时查阅在线帮助，其对应的M-file文件存放在系统\matlab\toolbox\matlab目录下。

一、阶梯图形

函数**stairs(x,y)**可以绘制阶梯图形，如下列程序段：

```
x=[-2.5:0.25:2.5];
```

```
y=exp(-x.*x);
```

```
stairs(x,y); 绘制阶梯图形命令
```

```
title('stairs plot');
```

二、条形图形(直方图)

函数**bar(x,y)**可以绘制二维垂直直方图,如下列程序段将绘制条形图形

```
x=[-2.5:0.25:2.5];
```

```
y=exp(-x.*x);
```

```
bar(x,y); 绘制条形图命令
```

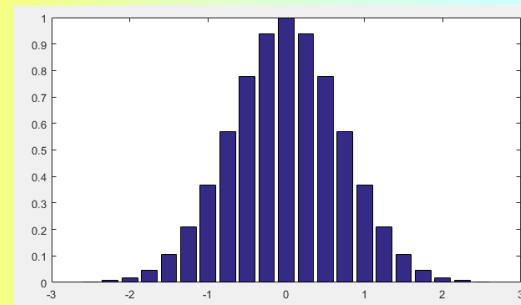
barh:绘制二维的水平直方图

bar3:用于绘制三维垂直直方图

bar3h:用于绘制三维水平直方图

```
bar3(A, 'detached'); bar3(A, 'group');
```

```
    bar3(A, 'stack');
```



三、饼图pie

某一工厂每一季度产品销售额

```
X=[56 78 60 99;66 88 40 112; 65 90 56 130;70 102 70 56]
```

```
Y=sum(X)
```

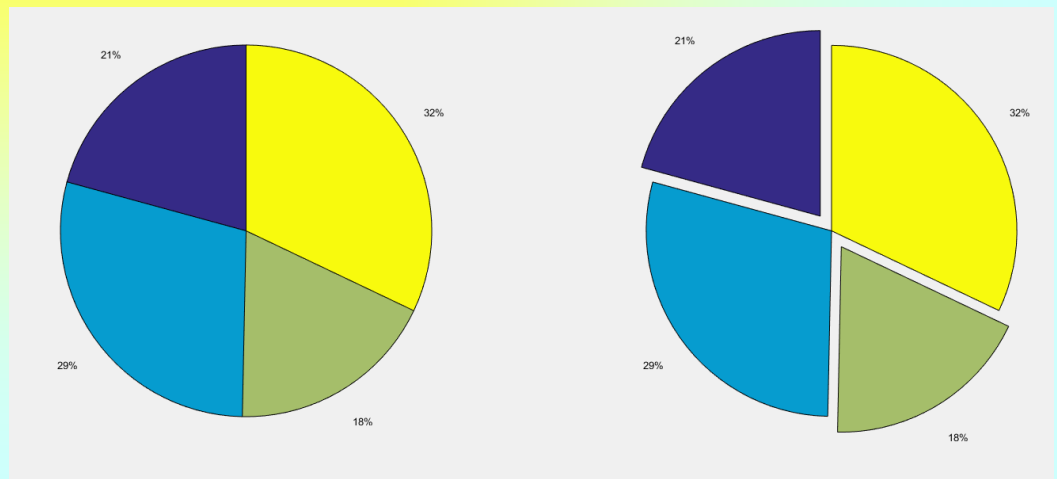
```
subplot(1,2,1)
```

```
pie(Y)
```

```
subplot(1,2,2)
```

```
pie(Y,[1,0,1,0]) %使第一、三块分离出来
```

```
Legend('p1','p2','p3','p4')
```



四、柱状图

柱状图函数首先计算在一定数据范围之内元素个数,然后将每个数据范围在图形窗口中显示为相应的矩形窗。

hist函数在笛卡尔坐标系中显示数据,
rose函数极坐标中显示数据

```
X=randn (1200, 1)  
hist(X)
```

五、枝干图：枝干图将每个离散的数据显示成尾部带有标记符号的线条，称为枝干。

函数： `stem`

单位脉冲序列 $\delta(n)$ 和单位阶跃序列 $u(n)$

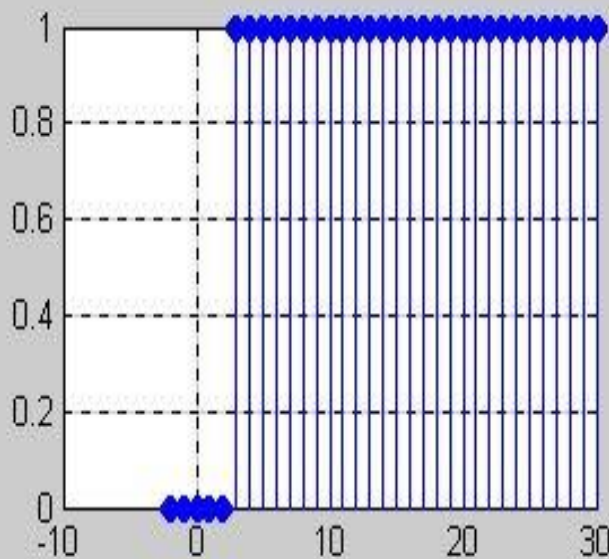
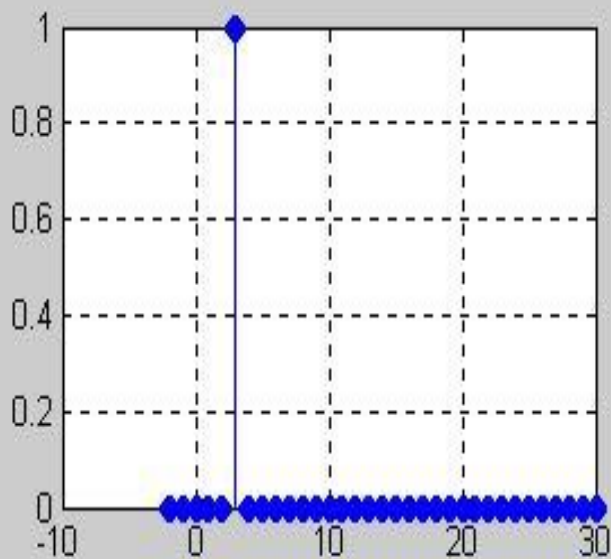
可以用`ones(1,n)`和`zeros(1,n)`来生成单位脉冲序列和单位阶跃序列

`ones(1,n)`命令产生1行n列的1值

`zeros(1,n)`命令产生1行n列的0值

产生单位脉冲序列 $\delta(n-5)$ 和单位阶跃序列 $u(n-5)$

```
n=-2:30;  
x=zeros(1,5),1,zeros(1,27);  
y=zeros(1,5),ones(1,28);  
subplot(2,1,1);  
stem(n,x,'fill');grid on;  
subplot(2,1,2)  
stem(n,y,'fill');grid on;
```



六、填充图形

fill(x,y,'c')函数用来绘制并填充二维多边形，x和y为二维多边形顶点坐标向量。字符 'c' 规定填充颜色。

下述程序段绘制一正方形并以黄色填充：

```
x=[0 1 1 0 0]; 正方形顶点坐标向量  
y=[0 0 1 1 0];  
fill(x,y, 'y');绘制并以黄色填充正方形图  
再如:  
x=[0:0.025:2*pi];  
y=sin(3*x);  
fill(x,y, [0.5 0.3 0.7]); 颜色向量
```

Matlab系统可用向量表示颜色，通常称其为颜色向量。基本颜色向量用[r g b]表示，即RGB颜色组合；以RGB为基本色，通过 r,g,b在0~1范围内的不同取值可以组合出各种颜色。

二维绘图函数小结

<code>plot</code>	二维图形基本函数
<code>fplot</code>	$f(x)$ 函数曲线绘制
<code>fill</code>	填充二维多边图形
<code>polar</code>	极坐标图
<code>bar</code>	条形图
<code>loglog</code>	双对数坐标图
<code>semilogx</code>	x轴为对数的坐标图
<code>semilogy</code>	y轴为对数的坐标图
<code>stairs</code>	阶梯形图
<code>axis</code>	设置坐标轴
<code>clf</code>	清除图形窗口内容
<code>close</code>	关闭图形窗口

<code>figure</code>	创建图形窗口
<code>grid</code>	放置坐标网格线
<code>gtext</code>	用鼠标放置文本
<code>hold</code>	保持当前图形窗口内容
<code>subplot</code>	创建子图
<code>text</code>	放置文本
<code>title</code>	放置图形标题
<code>xlabel</code>	放置x轴坐标标记
<code>ylabel</code>	放置y轴坐标标记

动画设计

如果将Matlab产生的多幅图形保存起来，并利用系统提供的函数进行播放，就可产生动画效果。系统所提供的动画功能函数有getframe、moviein和movie。

getframe函数

getframe函数可将当前图形窗口作为一个画面取下并保存。

格式：A=getframe

将每一帧画面信息数据截取下来整理成列向量。

该函数截取图形的点阵信息，图形窗口的大小，对数据向量的大小影响较大，窗口越大，所需存储容量越大。而图形的复杂性对数据容量要求没有直接的关系。

moviein函数

函数 $m=moviein(n)$ 用来建立一个足够大的 n 列的矩阵 m ，用来保存 n 幅画面的数据，以备播放。

movie函数

$movie(m,p)$ 以每秒 p 幅图形的速度播放由矩阵 m 的列向量所组成的画面。

【例】 播放一个不断变化的眼球程序段。

`m=moviein(20);` 建立一个20个列向量组成的矩阵

`for j=1:20`

`plot(fft(eye(j+10)))`

`%绘制出每一幅眼球图并保存到m矩阵中`

`m(:,j)=getframe;`

`end`

`movie(m,10);` 以每秒10幅的速度播放画面

再如下述程序段播放一个直径不断变化的球体。

```
n=30
[x,y,z]=sphere
m=moviein(n);
for j=1:n
    surf(j*x,j*y,j*z)
    m(:,j)=getframe;
end
movie(m,30);
```

图形句柄

一、句柄

在Matlab系统中，绘图命令产生的每一个部分称为**图形对象**，系统在创建每一个对象时，都为**该对象分配唯一的一个值，称其为句柄**，因此句柄就是图形对象标识符。对象、句柄以及图形对象等概念其实质是统一的，系统将每一个对象按树型层次结构组织起来，这些对象包括根对象，通常为计算机屏幕、图形窗口、坐标系统、线条、曲面、文本串、用户界面控制等。

根对象可包含一个或多个图形窗口对象，而一个图形窗口对象又可包含一组或多组坐标轴子对象，线条、文本等其它对象都是坐标系的子对象。所有创建对象的函数当父对象不存在时，都会自动创建它。

计算机屏幕作为根对象自动建立，其句柄值为0。而`Hf_f=figure`命令则建立图形窗口对象，并返回它的句柄值给变量`Hf_f`。图形窗口的句柄为一整数，并显示在该窗口的标题栏，其它图形对象的句柄为浮点数，Matlab提供了一系列与句柄操作有关的函数，如`gcf`、`gca`等。为便于识别，用大写字母开头的变量表示句柄，如`Hf_f`等。

(`gcf`返回当前Figure对象的句柄值
`gca`返回当前axes对象的句柄值)

二、对象属性

所有图形对象都具有控制对象显示的属性。这些属性既包括对象的一般信息，如对象类型、对象的父对象及子对象等，也包括对象的一些特定信息，如坐标系对象的刻度等。用户可以获取、设置对象属性，以达到控制对象的目的。当创建一个对象时，系统用一组默认属性值定制对象，用户get命令获取这些属性值，同时也可通过set命令重新设置对象属性。

set命令格式为：

set(H,'name',value, ...) 将图形对象H的name属性设置为value

其中H为句柄，name为属性名，value为name的属性值。

用set命令可以方便地设置图形对象属性，如下列程序段就是通过属性来定制图形。

```
x=[0:0.1:7*pi];
```

```
H=plot(x,sin(x)); 返回正弦曲线句柄H
```

```
set(H,'LineStyle','*','LineWidth',0.1); 设置  
正弦曲线线型与线宽
```

其中'**LineStyle**'为线型属性，'**LineWidth**'为线宽属性。

利用`get(H)`命令可以返回当前句柄`H`对象的属性。

键入命令：`get(H)` 系统返回当前图形对象的有关属性：

`H=get(0,'CurrentFigure')` 则返回根对象的'`CurrentFigure`'的属性值，即当前图形窗口的句柄，相当于函数`gcf`。`get(gcf,'Children')`则返回当前坐标系对象的句柄；类似的操作用户可在使用`Matlab`的过程中不断积累。

三、句柄应用

利用句柄操作的有关函数，用户可以查找、访问图形对象，以达到定制对象属性，改变对象显示效果的目的。

```
x=-pi:pi/20:pi;  
y=sin(x);z=cos(x);  
plot(x,y,'r',x,z,'g');  
H1_lines=get(gca,'Children');  
                                获取正、余曲线句柄向量H1_lines  
for k=1:size(H1_lines)  
if get(H1_lines(k),'Color')==[0 1 0]  
                                %[0 1 0]为绿颜色  
H1_green=H1_lines(k)          %返回绿色线条句柄  
end  
end
```

习题

1、在 $[0, 2\pi]$ 范围内绘制二维曲线图 $y = \sin(x) * \cos(5x)$ 。

2、在 $[0, 2\pi]$ 范围内绘制以Y轴为对数的二维曲线图。
 $y = |1000\sin(7x)| + 1$

3、采用图形保持，在同一坐标内绘制曲线 $y_1 = 0.2e^{-0.5x}\cos(4\pi x)$ 和 $y_2 = 2e^{-0.5x}\cos(\pi x)$ 。

4、绘制 $z = \sin(x) * \cos(y)$ 的三维网格和三维曲面图， x, y 变化范围均为 $[0, 2\pi]$ 。