

得用拟合或插值。

常用的拟合有多项式拟合 **POLYFIT**

插值有 **INTERP1**, **SPLINE**, **LAGR1** 等。。。

在 **Matlab** 中，用于曲线和曲面平滑的方法与函数很多，曲线平滑可用 **smooth** 和 **smoothts** 等，三维数据可用 **smooth3**，另外样条工具箱中也有不少可用于平滑数据的函数，如三次样条 **csaps** 和 B 样条 **spaps** 等。

matlab 中三维作图功能总结 2007-12-09 11:29 **plot3** 画三维坐标中的点，连线，但只能顺序连接。

surf(X,Y,Z) 用 **X** 和 **Y** 定义 **x-y** 坐标网格，**Z** 定义网格上每一点的高度，来生成三维曲面。
如：**[X,Y,Z] = peaks(30);surf(X,Y,Z)**

mesh，和 **surf** 一样，只不过生成的是网格。

surface 用法也一样。

fill3 只能生成平面。重点在色彩。

[X,Y,Z]=meshgrid(1:3,1:3,1:5) 生成 **3*3*5** 的三维网格，**X,Y,Z** 都是 **3*3*5** 三维矩阵。
这只是生成坐标网格，还需要一个 **V(X,Y,Z)** 定义图形。

ndgrid 生成三维以上网格时用。

smooth3 作用于体数据，使光滑

isosurface

X,Y,Z 如 **meshgrid** 的定义。

V 中元素为 1 则表示存在，即要显示。但要连成片的 1 才会显示。

V 中元素如 $a > 1$ 时，表示要显示的这个点离上方的网格距离是单位距离的 $1/a$
圆滑程度由 **isovalue** 决定，0.9999 是最硬，越接近 0 越圆滑。可同时配合 **isocaps**。

isocaps 生成并显示图形与坐标系交界处的平面。

patch 接收 **isosurface** 返回的参数，生成图形。

Matlab 曲面插值和拟合

附录:

Matlab 样条工具箱(Spline ToolBox)【信息来源教师博客】

Matlab 样条工具箱中的函数提供了样条的建立，操作，绘制等功能；

一. 样条函数的建立

第一步是建立一个样条函数，曲线或者曲面。这里的样条函数，根据前缀，分为 4 类：

cs* 三次样条

pp* 分段多项式样条，系数为 t^n 的系数

sp* B 样条，系数为基函数 $B_n^i(t)$ 的系数

rp* 有理 B 样条

二. 样条操作

样条操作包括：函数操作：求值，算术运算，求导求积分等等

节点操作：主要是节点重数的调节，设定，修改等等

附：样条工具箱函数

1. 三次样条函数

csapi 插值生成三次样条函数

csape 生成给定约束条件下的三次样条函数

csaps 平滑生成三次样条函数

cscvn 生成一条内插参数的三次样条曲线

getcurve 动态生成三次样条曲线

2. 分段多项式样条函数

ppmak 生成分段多项式样条函数

ppual 计算在给定点处的分段多项式样条函数值

3. B 样条函数

spmak 生成 B 样条函数

spcrv 生成均匀划分的 B 样条函数

spapi 插值生成 B 样条函数

spap2 用最小二乘法拟合生成 B 样条函数

spaps 对生成的 B 样条曲线进行光滑处理

spcol 生成 B 样条函数的配置矩阵

4. 有理样条函数

rpmak 生成有理样条函数

rsmak 生成有理样条函数

5. 样条操作函数

fnval 计算在给定点处的样条函数值

fmbrr 返回样条函数的某一部分（如断点或系数等）

fncomb 对样条函数进行算术运算

fn2fm 把一种形式的样条函数转化成另一种形式的样条函数

fnder 求样条函数的微分(即求导数)

fndir 求样条函数的方向导数

fnint 求样条函数的积分

fnjmp 在间断点处求函数值

fnplt 画样条曲线图

fnrfn 在样条曲线中插入断点。

fntlr 生成 taylor 系数或 taylor 多项式

6. 样条曲线端点和节点处理函数

augknt 在已知节点数组中添加一个或多个节点

aveknt 求出节点数组元素的平均值

brk2knt 增加节点数组中节点的重次

knt2brk 从节点数组中求得节点及其重次

knt2mlt 从节点数组中求得节点及其重次

sorted 求出节点数组的元素在另一节点数组中属于第几个分量

aptknt 求出用于生成样条曲线的节点数组

newknt 对分段多项式样条函数进行重分布

optknt 求出用于内插的最优节点数组

chbpnt 求出用于生成样条曲线的合适节点数组

☆☆☆☆☆☆(No Ratings Yet)

 Loading ...

2 Responses to “Matlab 曲面插值和拟合”

1. # *waitingon* 14 Nov 2007 at 2:22 pm

非常之高手，想请教一下，如果我只有三，四个已知三维坐标的点，能不能做到拟合呢？

我用你上面的方法，用 surf 画了一下，

```
t=csvread(' c:/www/dat/lchang.csv' );
```

```
x=[t(1,1,1), t(1,4,1), t(1,7,1), t(1,10,1)];
```

```
y=[t(1,2,1), t(1,5,1), t(1,8,1), t(1,11,1)];
```

```
z=[t(1,3,1), t(1,6,1), t(1,9,1), t(1,12,1)];
```

```
surf(x,y,z)
```

```
title(' Original data Plot' );
```

```

xlabel(' X' ), ylabel(' Y' ), xlabel(' Z' ),

colormap, colorbar;

axis equal

```

结果只出来坐标轴，没有任何关于我的点的东西，不知道为什么，希望给与指点

2. # yangon 14 Nov 2007 at 3:46 pm

并不是数据太少，理论上 3 点就可以确定一个面，所以不是数据多少的问题，而是数据格式不对，你的坐标可能只是离散的坐标，这样子只能画三维曲线出来，而不能画面。你可以试试用 plot3 应该可以画出你的数据来。

如果你想画曲面，要把数据整理成固定的格式，下面一个简单的例子，希望对你有帮助，你可以看出来，只有四个点也是可以画出曲面来的。

```

x=[1:1:2]

y=[1:1:2]

[xx,yy]=meshgrid(x,y)

z=[4.1,4.2;4.3,4.4]

surf(x,y,z)

```

我看到“Matlab 曲面插值和拟合”很受启发，我有一个问题，你可以帮我吗：

```

x = 0:4; y=-2:2; s2 = 1/sqrt(2);
clear v
v(3, :, :) = [0 1 s2 0 -s2 -1 0].'*[1 1 1 1 1];
v(2, :, :) = [1 0 s2 1 s2 0 -1].'*[0 1 0 -1 0];
v(1, :, :) = [1 0 s2 1 s2 0 -1].'*[1 0 -1 0 1];
sph = csape({x,y},v,{'clamped','periodic'});
values = fnval(sph, {0:.1:4, -2:.1:2});
sphere=surf(squeeze(values(1, :, :)), squeeze(values(2, :, :)), squeeze(values(3, :, :)));
axis equal, axis off

```

```

[xx,yy]=ndgrid(-1:.5:1,-1:.5:1);

```

请教：如何求拟合球面sph(图形sphere)上对应点[xx,yy]的z坐标？

十分感谢, 如果可以, 请把你的思路给我发email: zpshh@126.com

插值和拟合都是数据优化的一种方法, 当实验数据不够多时经常需要用到这种方法来画图。在 matlab 中都有特定的函数来完成这些功能。这两种方法的确别在于:

当测量值是准确的, 没有误差时, 一般用插值:

当测量值与真实值有误差时, 一般用数据拟合。

插值:

对于一维曲线的插值, 一般用到的函数 `yi=interp1(X,Y,xi,method)` , 其中 `method` 包括 `nearest`, `linear`, `spline`, `cubic`。

对于二维曲面的插值, 一般用到的函数 `zi=interp2(X,Y,Z,xi,yi,method)`, 其中 `method` 也和上面一样, 常用的是 `cubic`。

拟合:

对于一维曲线的拟合, 一般用到的函数 `p=polyfit(x,y,n)` 和 `yi=polyval(p,xi)`, 这个是最常用的最小二乘法的拟合方法。

对于二维曲面的拟合, 有很多方法可以实现, 但是我这里自己用的是 Spline Toolbox 里面的函数功能。具体使用方法可以看后面的例子。

对于一维曲线的插值和拟合相对比较简单, 这里就不多说了, 对于二维曲面的插值和拟合还是比较有意思的, 而且正好胖子有些数据想让我帮忙处理一下, 就这个机会好好把二维曲面的插值和拟合总结归纳一下, 下面给出实例和讲解。

原始数据

```
x=[1:1:15];
```

```
y=[1:1:5];
```

```
z=[0.2 0.24 0.25 0.26 0.25 0.25 0.25 0.26 0.26 0.29 0.25 0.29;
```

```
0.27 0.31 0.3 0.3 0.26 0.28 0.29 0.26 0.26 0.26 0.26 0.29;
```

```
0.41 0.41 0.37 0.37 0.38 0.35 0.34 0.35 0.35 0.34 0.35 0.35;
```

```
0.41 0.42 0.42 0.41 0.4 0.39 0.39 0.38 0.36 0.36 0.36 0.36;
```

```
0.3 0.36 0.4 0.43 0.45 0.45 0.51 0.42 0.4 0.37 0.37 0.37];
```

z 是一个 5 乘 12 的矩阵。

直接用原始数据画图如下：

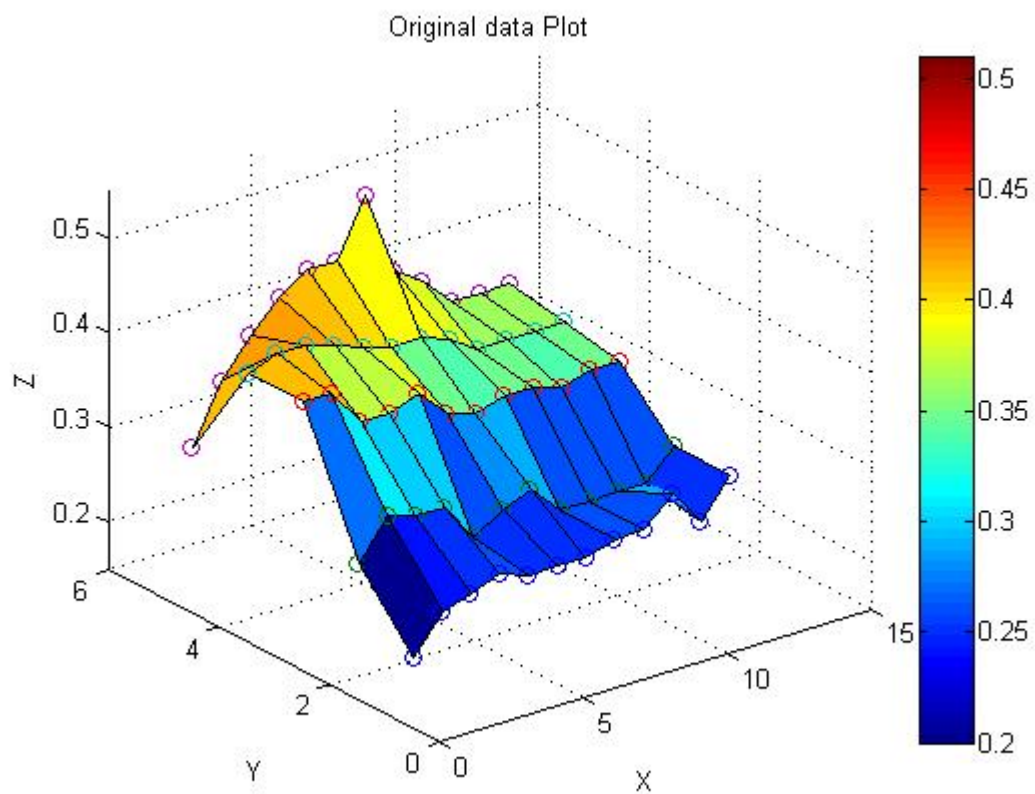
```
surf(x, y, z)
```

```
title(' Original data Plot' );
```

```
xlabel(' X'), ylabel(' Y'), zlabel(' Z'),
```

```
colormap, colorbar;
```

```
axis([0 15 0 6 0.15 0.55])
```



先考虑插值，需要用到的函数 interp2

```
x1=1:0.2:12;
```

```
y1=1:0.2:5;
```

```
[x2,y2]=meshgrid(x1,y1);
```

```

t11=interp2(x,y,z,x2,y2,'cubic');

surf(x1,y1,t11)

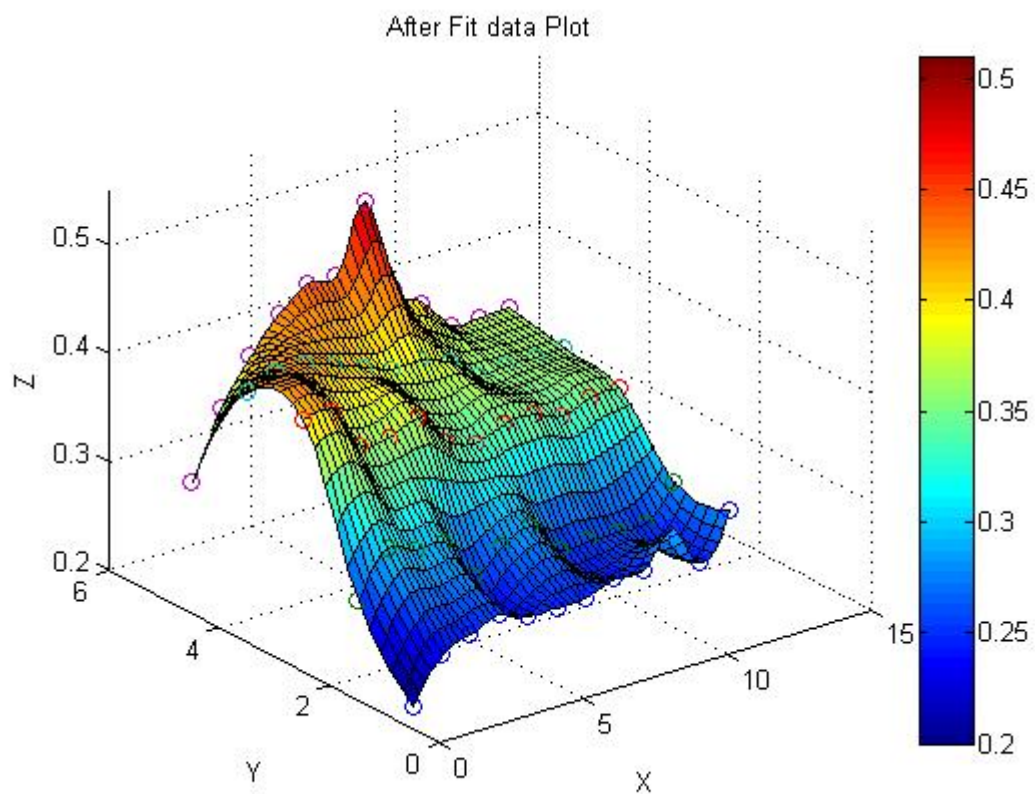
title(' After Fit data Plot' );

xlabel(' X'), ylabel(' Y'), zlabel(' Z'),

colormap, colorbar;

axis([0 15 0 6 0.2 0.55])

```



然后考虑拟合，这个稍微复杂一点：

```

ky = 3; knotsy = augknt([0:2.5:13],ky);

sp = spap2(knotsy,ky,y,z);

yy = 0:.5:12; vals = fnval(sp,yy);

cofsy = fnbrk(sp,'c');

kx = 5; knotsx = augknt([1:4:5],kx);

```



```

sp2 = spap2(knotsx,kx,x,coefsy.' ');

coefs = fnbrk(sp2,' c').';

xv = 1:.25:5; yv = 1:.5:12;

values = spcol(knotsx,kx,xv)*coefs*spcol(knotsy,ky,yv).';

surf(yv,xv,values)

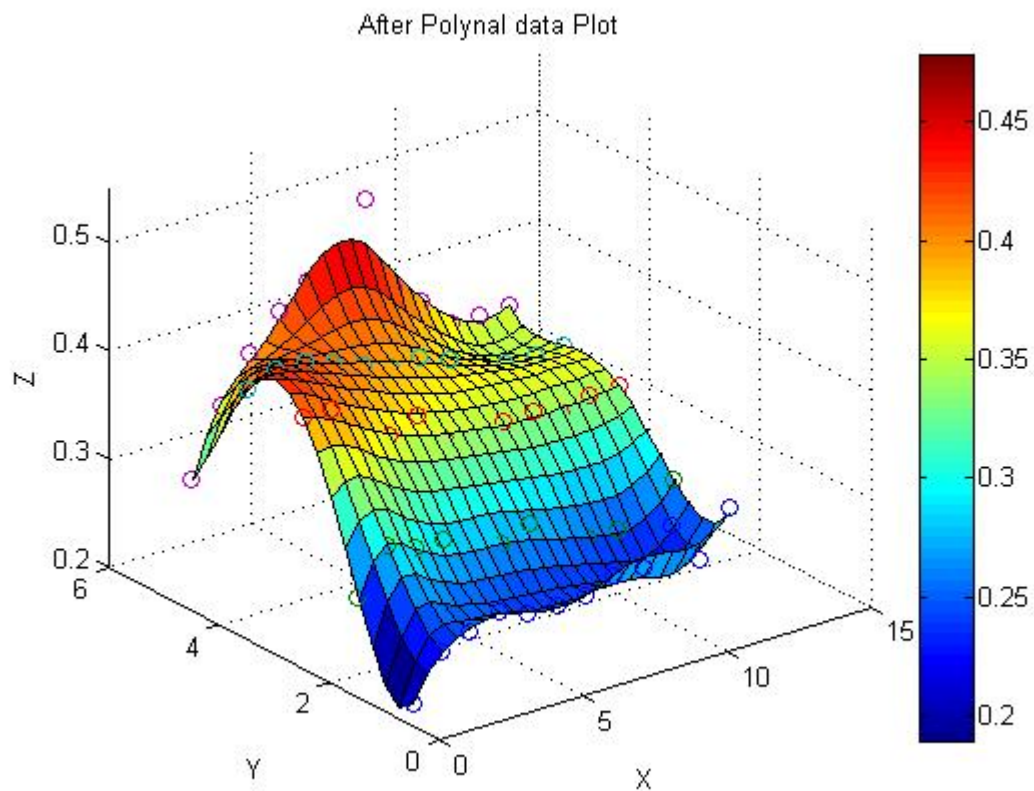
title(' After Polynal data Plot' );

xlabel(' X'), ylabel(' Y'), zlabel(' Z');

colormap, colorbar;

axis([0 15 0 6 0.2 0.55])

```



图上面的小圆点使用 plot3 画出来的原始数据点，具体怎么添加的方法我就不多说了。需要把不等边矩阵转换成等边矩阵然后再画图。

效果从上面三个图上面已经看出来了。基本上满足了现在画图的要求，如果测量值比较精确，我们选择第二种方法插值法来画图，如果觉得测量值不是很精确需要用拟合的方法的话，那当然用第三种方法了。关

于 Spline Toolbox 当然还有更多应用，还有就是里面参数设置为什么要这样设，该怎样设要写下来估计再写一整页都写不完，所以还是大家感兴趣的自己去看帮助文档好了，里面有详细介绍。

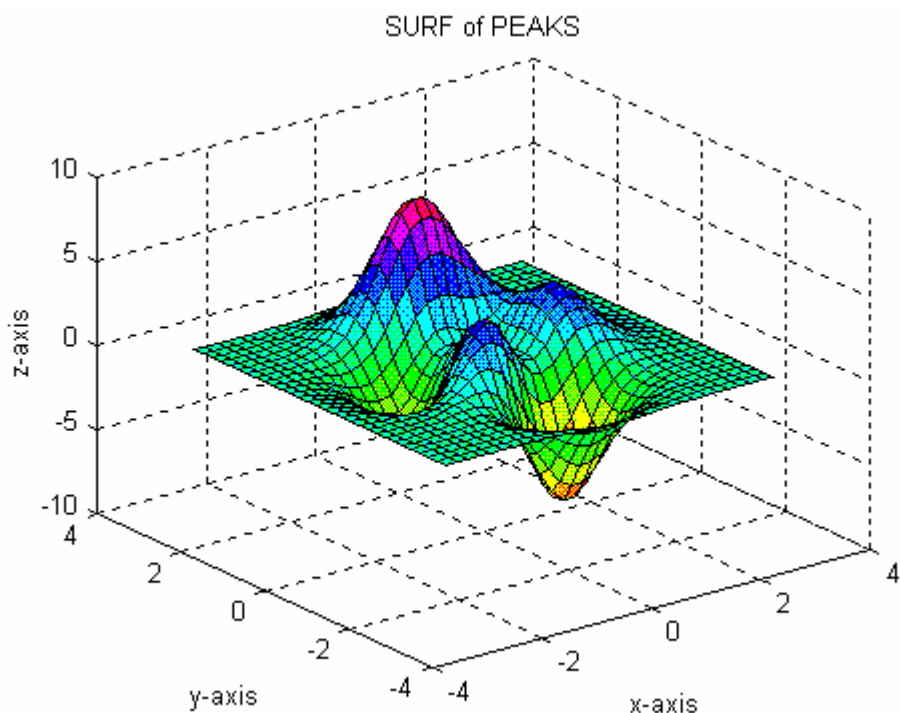
这里只是告诉大家一种可行的办法，而且经过我自己的测试也确实得到了想要达到的结果。Matlab 是个很强的数学工具，前一段时间我也提到了它在 Texture 里面做图的应用。以后会有更多地方用到 Matlab，感兴趣的朋友可以和我交流。^_^

3.3.5 曲面图

曲面图，除了各线条之间的空档（称作**补片**）用颜色填充以外，和网格图看起来是一样的。这种图一般使用函数 **surf** 来绘制。自然，函数 **surf** 使用和函数 **mesh** 相同的调用语法。比如：

```
[X, Y, Z]=peaks(30);  
  
surf(X, Y, Z)  
  
grid, xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')  
  
title('SURF of PEAKS')
```

输出见下图



函数 PEAKS 的曲面图

曲面图的一些特性正好和网格图相反：它的线条是黑色的，线条之间的补片有颜色；而在网格图里，补片是黑色的而线条有颜色。对函数 **mesh**，颜色沿着 z 轴按每一补片变化，而线条颜色不变。

在曲面图里，人们不必考虑象网格图一样隐蔽线条，但要考虑用不同的方法对表面加色彩。在前面的曲面图的例子中，就是分割成**块**，每块就象一块染色玻璃窗口或物体，黑线便是各单色染色玻璃块之间的连接。除此以外，MATLAB 还提供了**平滑**加颜色和**插值**加颜色功能。这可以通过调用函数 **shading** 来实现。

```
[X, Y, Z]=peaks(30);

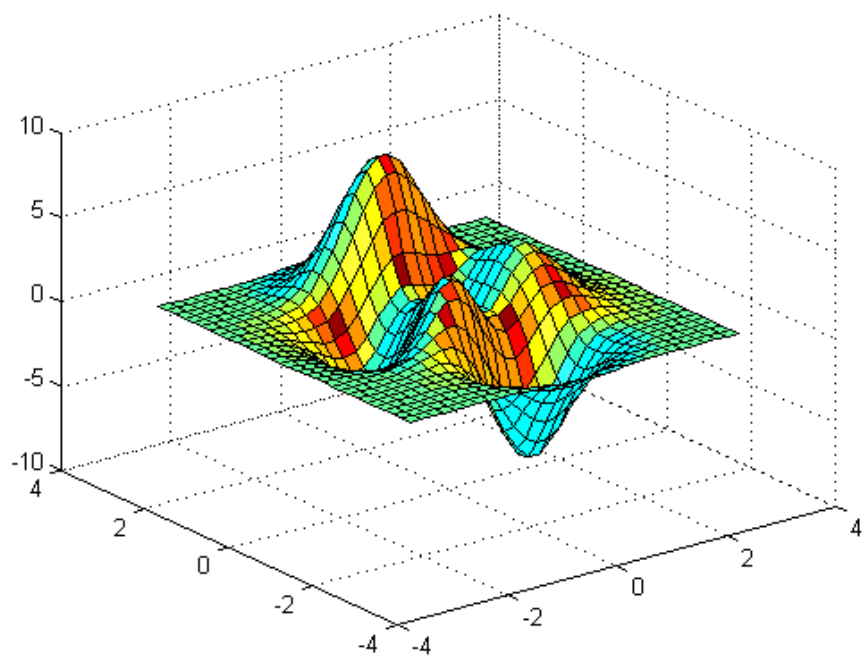
surf(X,Y,Z) % same plot as above

grid,xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis')

title('SURF of PEAKS')

shading flat
```

输出见下图

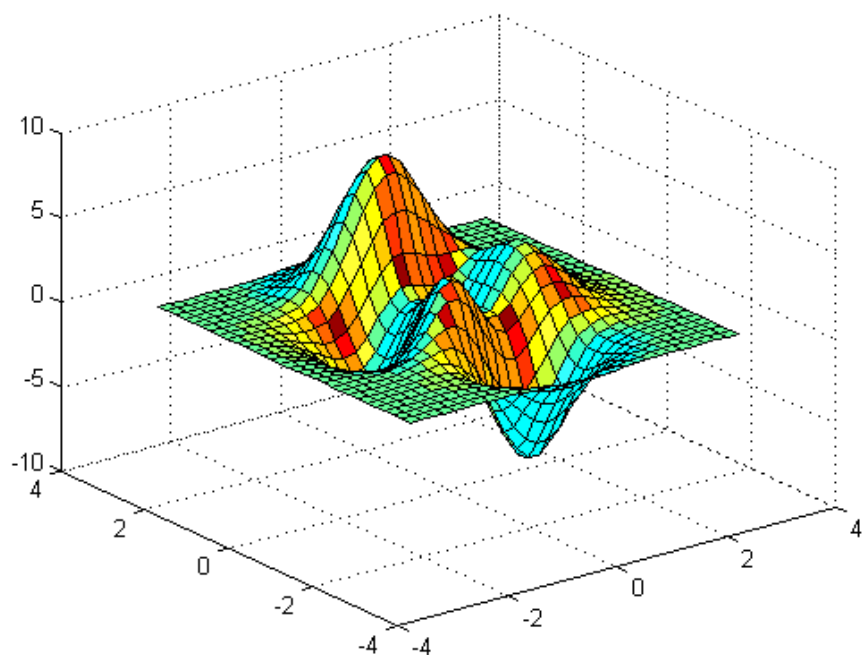


函数 PEAKS 的平滑加彩色曲面图

如上所示平滑加色彩的例子中，每一补片仍保存着单一的颜色，但各块连接处的黑线已去掉。

`shading interp`

输出见下图



函数 PEAKS 的插值加彩色曲面图

如上所示内插加色彩的例子中，同样去掉了线条，但各补片以插值加颜色，即各补片的颜色根据赋予顶点的色值，对其区间进行了插值计算。很明显，插值色彩需要比分块和平滑更多的计算量。在一些计算机系统中，插值色彩会产生非常长的打印延时或打印错误。这问题不在于 PostScript 文件太大，而是由于在打印机上产生沿图形曲面连续变化的阴影所需的巨大计算量。通常对这个问题最简单的解决方法是使用平滑加色彩法来打印。

色彩对 **surf** 作图的视觉效果有着巨大的影响。对网格图也是如此，尽管由于只有线条有颜色，对视觉效果的影响相对要小一些。

因为曲面图不能作成透明，但在一些情况下可以很方便地移走一部分表面以便看到表面以下部分，在 MATLAB 中，这是通过在所期望的洞孔的所在位置，将数据置为特定的 NaN 来实现。由于 NaN 没有任何值，所有的 MATLAB 作图函数都忽略 NaN 的数据点，在该点出现的地方留下一个洞孔。例子如下：

```
[X,Y,Z]=peaks(30);  
  
x=X(1,:); % vector of x axis  
  
y=Y(:,1); % vector of y axis  
  
i=find(y>.8 & y<1.2); % find x-axis indices of hole
```

```

j=find(x>-.6 & x<.5); % find x-axis indices of hole

Z(i,j)=nan*Z(i,j); % set values at hole indices to NaNs

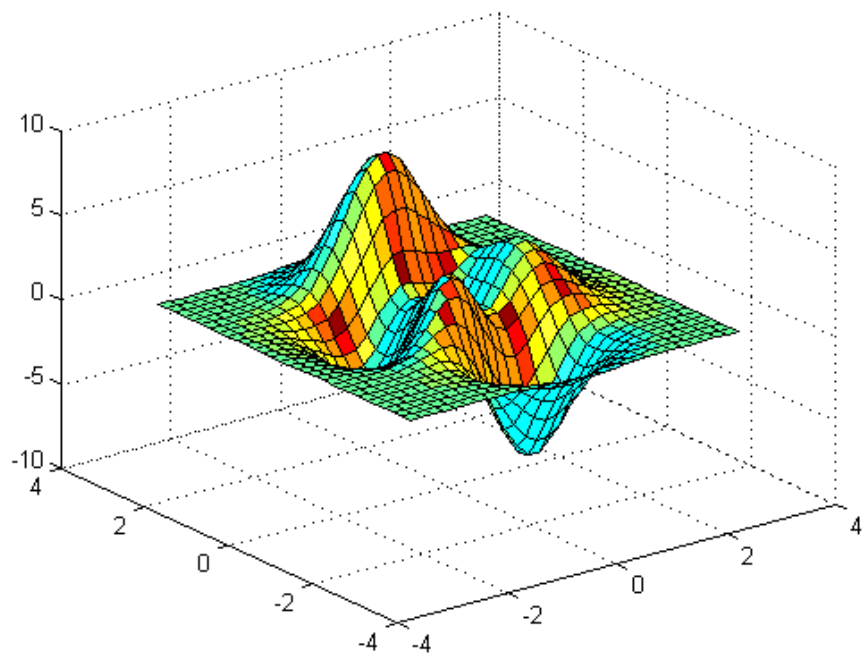
surf(X,Y,Z)

grid,xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis')

title('SURF of PEAKS with a Hole')

```

输出见下图



函数 PEAKS 的带洞孔曲面图

MATLAB 的 **surf** 也有两个同种函数:**surfc**, 它画出具有基本等值线的曲面图;**surfl**, 它画出一个有亮度的曲面图。例如:

```

[X,Y,Z]=peaks(30);

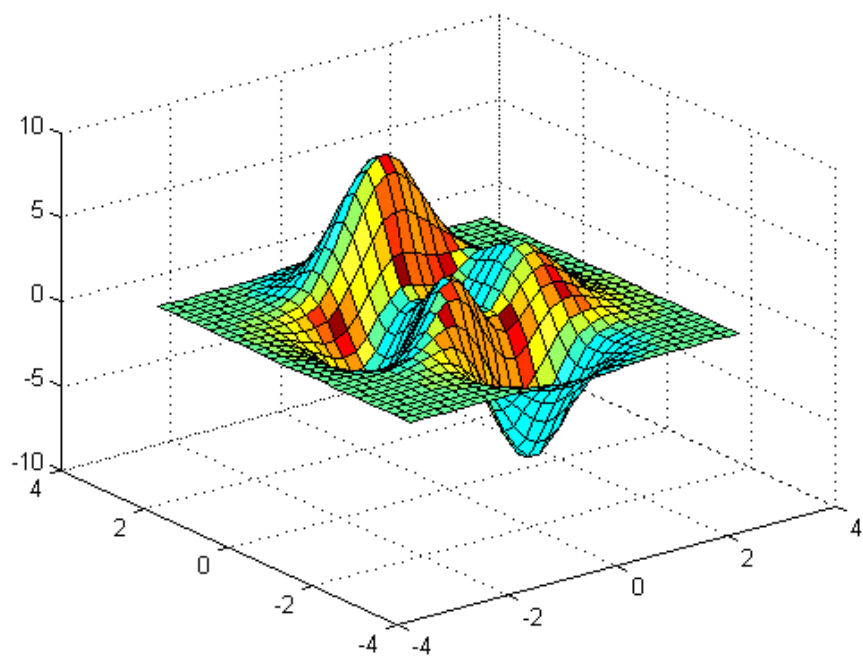
surfc(X,Y,Z) % surf plot with contour plot

grid,xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis')

title('SURFC of PEAKS')

```

输出见下图



函数 PEAKS 的曲面图和基本等值线图

```
[X,Y,Z]=peaks(30);

surf1 (X,Y,Z) % surf plot with lighting

shading interp % surf1 plots look best with interp shading

colormap pink % they also look better with shades of a single color

grid,xlabel('X-axis '),ylabel('Y-axis '),zlabel('Z-axis ')

title('SURFL OF PEAKS ')
```

输出见下图

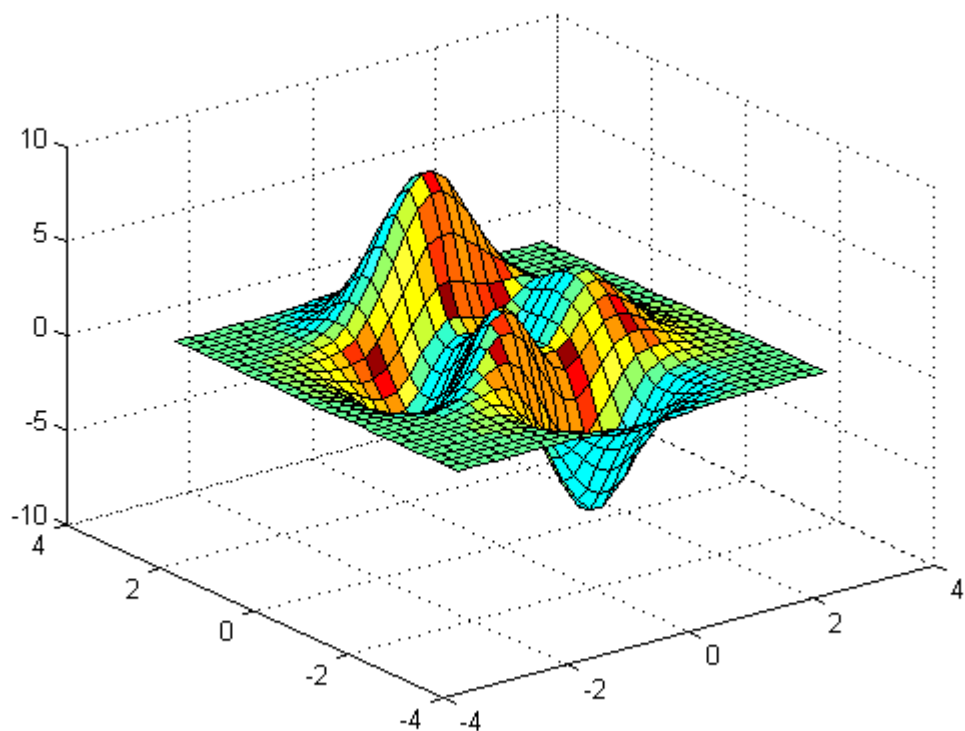


图 18.16 函数 PEAKS 的带光线照明曲面图

关于加到曲面的亮度，函数 **surf1** 作了许多假设。有关设置亮度属性的详细信息请使用在线帮助。

3.3.6 等值线图

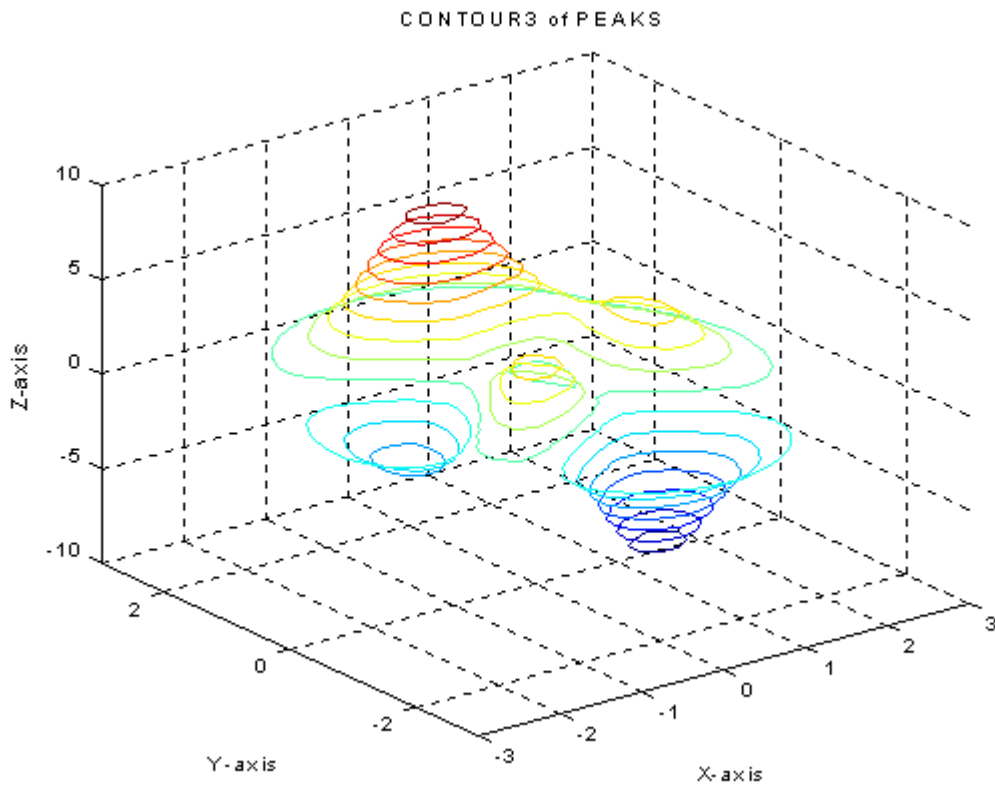
MATLAB 提供了另一种基本的三维图形，即三维等值线图。这种图形通过函数 **contour3** 来绘制。

```
[x, y, z]=peaks(30);  
  
contour3(X, Y, Z, 16); %用 16 种颜色  
  
grid,xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis');
```



```
title('CONTOUR3 of PEAKS')
```

输出见下图



函数 PEAKS 的三维等值线图

可以看到，图形中每一条线的颜色遵循了与二维函数 `plot` 一样的次序。这种颜色次序可以表现出明显的对比，但经常模糊了所代表的数据的一些重要特性。如果能使每一条线遵循在**网格图**和**曲面图**里所用的加色方法，那么效果会好得多。也许在 MATLAB 的下一个版本中，这种颜色设置会成为缺省设置，但使用在下一章要讨论的 MATLAB 图形处理能力，也能解决这个问题。

```
[X, Y, Z]=peaks(30);
```

```
N=16; % 等值线的数目和颜色
```

```
clf %清除当前颜色
```

```
view(3) % 设置视角
```

```
hold on % 保持背景
```

```

set(gca, 'ColorOrder',hsv(N)) % 从 default hsv colormap 获得颜色

contour3(X,Y,Z,N) % 作出 N 条等值线

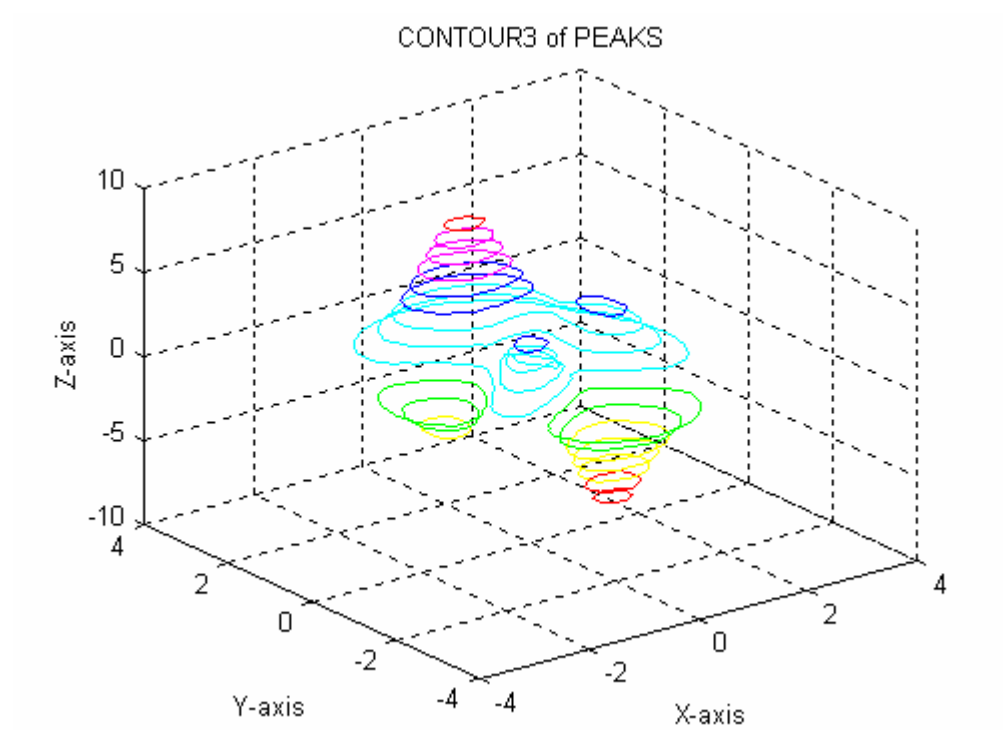
grid,xlabel('X-axis'),ylabel('Y-axis'),zlabel('Z-axis')

title('CONTOUR3 of PEAKS')

hold off

```

输出见下图



函数 PEAKS 的三维等值线图

现在，各条等值线的颜色沿着 z 轴的变化和**网格图**和**曲面图**一样。为方便起见，这种策略已体现在精通 *MATLAB* 工具箱的函数 **mmcont3** 中。**mmcont3** 具有和函数 **contour3** 相同的调用语法的变化，并允许选择可用的颜色映象。例如，**mmcont3(X, Y, Z, N, 'hsv')** 复制上面的图形。**mmcont3** 的在线帮助如下：

帮助信息：

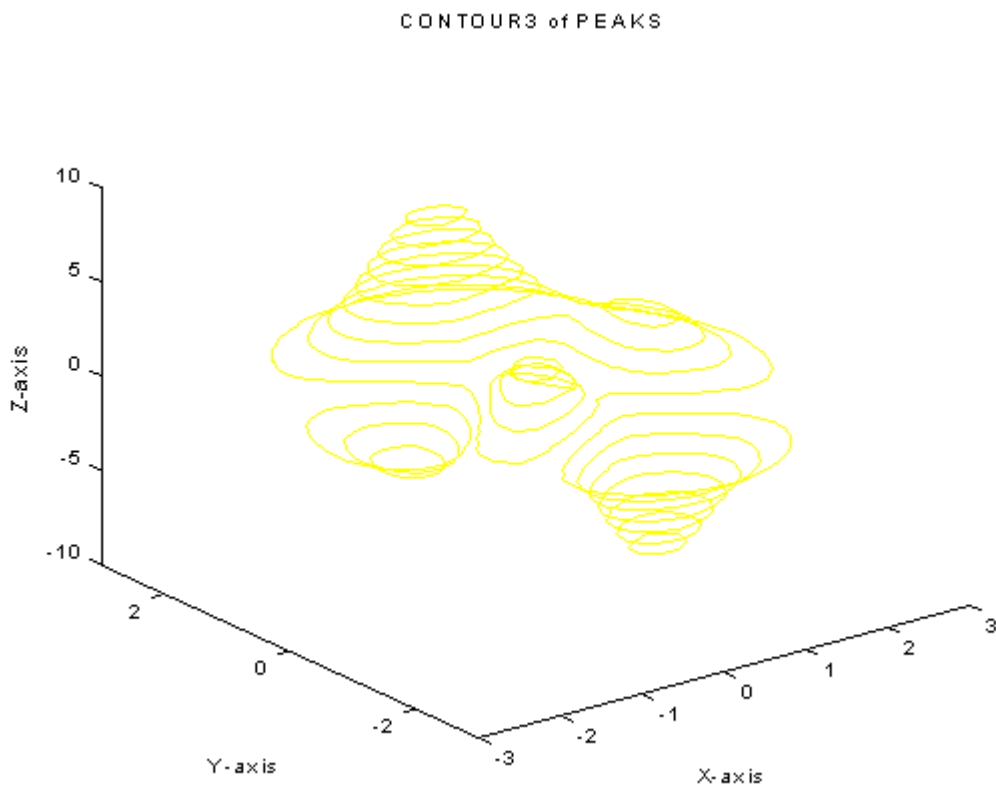
MMCONT3(X, Y, Z, N, C)用由C指定的颜色在三维空间内画N条Z方向的等值线图。C可以是在plot中使用的线形和颜色，例如‘r-’；或者C可以是一个颜色映象的字符串名。X和Y指定了坐标轴的范围。如果未指定参数，缺省的参数值是：N=10，C=‘hot’，X和Y分别是Z的行和列的下标。举例：

MMCONT3(Z)	用暖色映象画10条等值线
MMCONT3(Z, 20)	用暖色映象画20条等值线
MMCONT3(Z, ‘copper’)	用铜黄色映象画10条等值线
MMCONT3(Z, 20, ‘gray’)	用灰色映象画20条等值线
MMCONT3(X, Y, Z, ‘jet’)	用**‘jet’暖色映象画10条等值线
MMCONT3(Z, ‘c--’)	画10条青蓝色的虚划线等值线
MMCONT3(X, Y, Z, 25, ‘pink’)	用粉红色映象画25条等值线
CS=MMCONT3(...)	如在CONTOURC中描述，返回等值线矩阵CS。
[CS, H]=MMCONT3(...)	把句柄的列向量H返回到线条对象。

等值线也可由一种颜色给出：

```
[x, y, z]=peaks(30);  
  
contour3(X, Y, Z, 16, ‘y’) % draw sixteen contour lines in yellow  
  
grid, xlabel(‘x-axis’), ylabel(‘y-axis’), zlabel(‘z-axis’)  
  
title(‘CONTOUR3 of PEAKS’)
```

输出见下图



函数 PEAKS 的黄色三维等值线图

以上函数使用的详细信息请使用在线帮助。

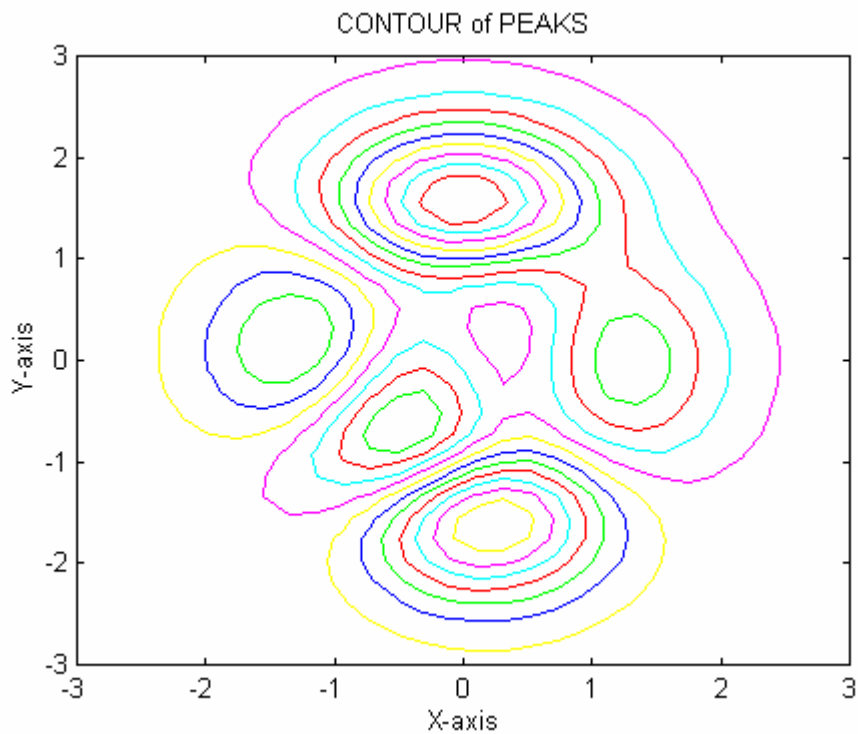
3.3.7 三维数据的二维表示

有些情况下，希望得到三维数据的二维表示。在 MATLAB 里这一点是通过用函数 **view** 设置视角使其中一维不出现来实现的。另外，MATLAB 还提供了两个函数，将 **contour3** 和 **surf** 向下正视到 x-y 平面。例如，函数 **contour3** 的二维图就等价于 **contour**。

```
[X, Y, Z]=peaks(30);  
  
contour(X, Y, Z, 16) % 作 16 条等值线  
  
xlabel('X-axis'), ylabel('Y-axis')
```

```
title('CONTOUR of PEAKS')
```

输出见下图



函数 PEAKS 的等值线图

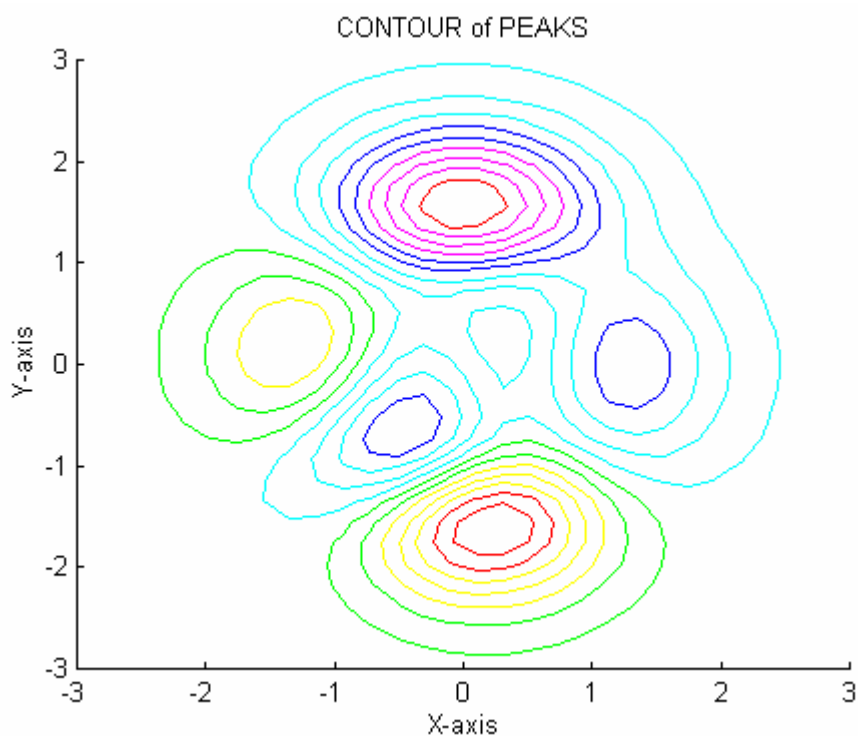
要注意，它如何等效于使用 **contour3** 以及如何改变视点俯视到 $x-y$ 平面。如同 **contour3**，图中的等值线利用 **plot** 命令的六种基本颜色。如前所述，因为颜色不提供视觉效果，这类图形不好使用并且引起混淆。这种缺省行为也许在 MATLAB 的下一版本中会改变，但也可用句柄图形来改变。

```
[X, Y, Z]=peaks(30);  
  
N=16; % number of contour lines and their colors  
  
clf % clear the current figure  
  
hold on % hold blank screen  
  
set(gca, 'ColorOrder',hsv(N)) % use colors from default hsv  
colormap  
  
contour(X,Y,Z,N) % draw N contour lines  
  
xlabel('X-axis'),ylabel('Y-axis'),zlabel('Z-axis')
```

```
title('CONTOUR of PEAKS')
```

```
hold off
```

输出见下图



函数 PEAKS 的等值线图

现在，等值线遵循 **hsv** 颜色映象里的级差，颜色提供了一种有用的效果！为方便起见，上述策略已体现在精通 *MATLAB* 工具箱的函数 **mmcont2** 中。**mmcont2** 接受和函数 **contour** 相同的调用语法变更，并允许选择一个可用的颜色映象。例如，**mmcont2** (X, Y, Z, N, 'hsv') 复制上面的图形。**mmcont2** 的在线帮助如下：

帮助信息：

MMCONT2 (X, Y, Z, N, C) 用由 C 指定的颜色在二维平面内画出关于 Z 方向的 N 条等值线。C 可以是在函数 **plot** 中使用的线形和颜色，例如 'r-'；或者是一个字符串指明颜色映象名。

X,Y 指定了坐标轴的范围。

如果未指定参数，缺省的参数值是：N=10，C='hot'，X和Y分别是Z的行列下标。

例子如下：

MMCONT2(Z)	使用暖色颜色映象画 10 条等值线
MMCONT2(Z, 20)	使用暖色颜色映象画 20 条等值线
MMCONT2(Z, 'copper')	使用铜黄色颜色映象画 10 条等值线
MMCONT2(Z, 20, 'gray')	使用灰色颜色映象画 20 条等值线
MMCONT2(X, Y, Z, 'jet')	使用 'jet' 暖色颜色映象画 10 条等值线
MMCONT2(Z, 'c--')	使用青蓝色颜色映象用虚划线画 10 条等值线
MMCONT2(X, Y, Z, 25, 'pink')	使用粉红色颜色映象画 25 条等值线

CS=MMCONT2(…)返回等值线矩阵 CS，它在 CONTOURC 中描述。

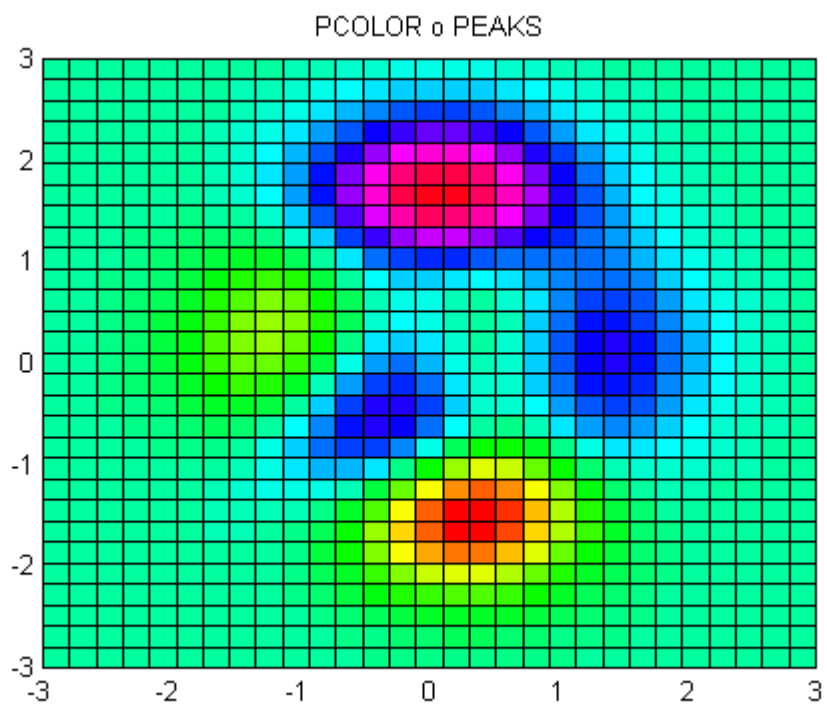
[CS, H]=MMCONT2(…)返回一个包含有线条对象的句柄的列向量。

上面讨论的三维或二维等值线图都假定数据定义在矩形网格或论域内。当不是这种情形时，数据必须被转换到矩形网格上，使等值线图看起来逼真。如本章前面 18.4 节所述，MATLAB 函数 **griddata** 能实现所需的转换，更详细的信息参阅该节。

函数 **surf** 的二维等效函数是 **pcolor**，它代表伪彩色。

```
[X, Y, Z]=peaks(30);  
  
pcolor(X, Y, Z); % surf plot view from above  
  
xlabel('X-axis'), ylabel('Y-axis')  
  
title('PCOLOR of PEAKS')
```

输出见下图



函数 PEAKS 的伪彩色图

由于这是一个 **surf** 图，可以使用函数 **shading**。另外，有时在 **pcolor** 图的上面放一个单色**等值线图**是很有用的。

```
[X, Y, Z]=peaks(30);

pcolor(X, Y, Z);

shading interp

hold on

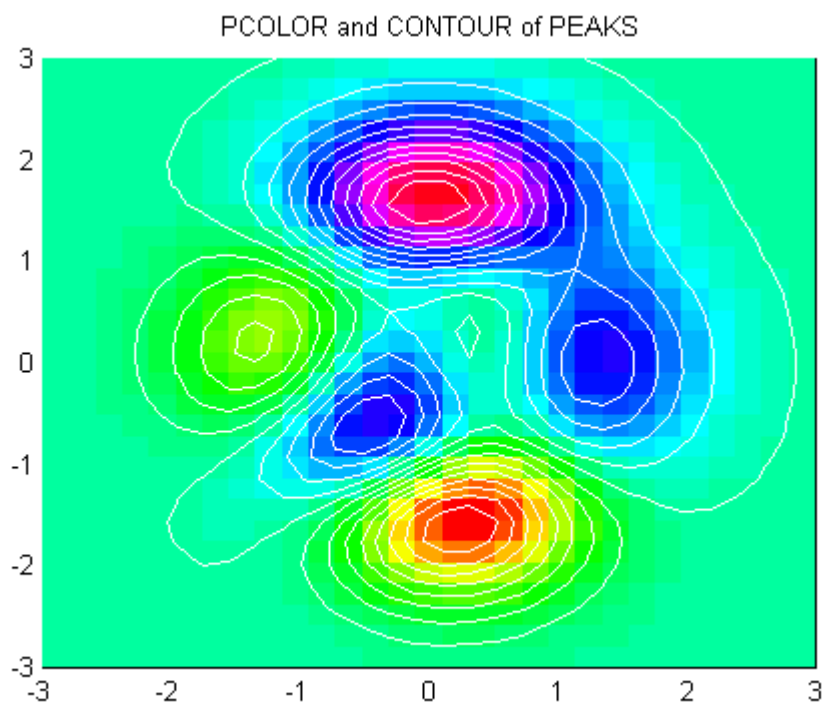
contour(X, Y, Z, 19, 'k') %增加 19 条黑色的等值线

xlabel('X-axis'), ylabel('Y-axis')

title('PCOLOR and CONTOUR of PEAKS')

hold off
```

输出见下图



函数 PEAKS 的伪彩色图和等值线图

4.1.3 格式化输入与输出

`fprintf` 函数的功能是按指定的格式把数据写入一个磁盘文件。其格式为

```
count=fprintf(fid, format, a,b...)
```

`fid` 是要写的文件的标识, 由 `fopen` 命令获得。`a,b...` 为输出数据的变量名。`count` 为输出可选项, 返回成功输出的字节数。如果不指定 `fid`, 则默认 `fid=1`, 即输出到屏幕。

`format` 为控制输出的格式, 是一个字符串, 它与 C 语言中的规定一致。`format` 的格式及各项的含义为:

其中每一项的定义如下:

(1) 格式类型

格式类型是一个大写或小写的字符，定义为：

d (或 i) ——十进制整数

X(或 x) ——十六进制数

o——八进制数

u——无符号十进制数

C——单字符

S——字符串

c (或 E) ——指数型浮点数

f——小数型浮点数

g (或 G) ——e, f 中取短，不输出无效 0

(2) 长度修正

l——长型，如 %ld, %lX, %lu

h——短型，如 %hd, %hx

(3) 域宽及精度描述

m——域宽，小数点占一位，缺省为 10

n——小数位数，缺省为 6

(4) 域宽不满时处理方式

0——数字前添 0

无此项—数字前填充格

(5) 左右对齐

——左对齐

+——右对齐

(6) %为格式说明起始符

(7) /n 为开始新行输出符

下面举例说明中 fprintf 的使用方法。如果有一个 11*2 的矩阵，数据格式为浮点数，把他们写到一个名为 exp. dat 的文件中。程序为：

```
x=[0:0.1:1]';
```

```
y=[x;exp(X)];
```

```
dog=fopen('exp.dat','w');
```

```
fprintf(dog,'%6.2f %12.8f/n',y);
```

该程序运行后，输出一文件，名为 exp. dat，内容为：

```
0.00 1.00000000
```

```
0.10 1.10517092
```

```
1.00 2.71828183
```

从此例可以看出，尽管 fprintf 的命令格式与 C 语言中的类似，但有一个主要的不同点是：

这里变量名只有一个 y，而输出的是 11 行数据。所以说 MATLAB 中的 fprintf 是矢量式的输出。

fscanf

fscanf 函数的功能是按指定格式从文件读取数据。其格式为

```
[A,count]=fscanf(fid,format,size)
```

其中, `fid` 是由 `fopen` 语句获得的文件标识。被读取的数据存放于变量 `A` 中。`count` 是成功读取的字节数, 它为输出可选项。`format` 为指定格式, 其规定与 `printf` 中的 `format` 相同。`size` 为输入可选项, 它是对读取数据量的限制, 定义为

`N` —最多读 `N` 个元素, 形成一列矢量

`inf` —最多可以读取到文件尾

`[m, n]` —最多读 `m*n` 个元素, 放入 `m * n` 矩阵

中, `n` 可为 `inf`, `m` 不可

例如

```
T=fscanf(fish, 's')
```

从标识为 `fish` 的文件中读取一个字符, 放入变量 `T`, `T` 即为字符串变量。

```
A=fscanf(cat, '%5d')
```

从标识为 `cat` 的文件中读取 5 个十进制整数。

`fgetl`

`fgetl` 函数用于从文件中读行, 读时放弃换行符。格式为

```
line=fgetl(fid)
```

该命令把 `fid` 指定的文件中的一行以字符串的形式读入变量 `line`。读入的字符串中不包括换行符。如读到文件末尾, 则返回 -1。该命令只能对文本式文件进行操作。如果读取没有换行符的二进制文件, 则会运行很长时间。

`fgetl` 函数应用实例 (`exmp501.m`)。

假如磁盘中有一数据文件 `ps.dat`, 文件中的数据是按行存储的, 下列程序可以把它们读入变量 `line` 中。

```
fid=fopen('ps.dat','r');
```

```

while 1

line=fgetl(fid);

if ~isstr (line) , brak, end;

disP (line) ;

end

fclose;

```

该程序把文件 ps. dat 一行一行地读入到 line 中，每读一行在屏幕上显示一次，直至文件末尾。注意，程序中结束 while 循环的方法是当读到文件末尾时，根据 fgetl 的规定返回 line=-1，通过 isstr (line) 检测到已非字符串时，用 break 中断循环，从而终止读行操作。

fgets

fgets 函数用于从文件中读行，读时保留换行符。格式为

```

line=fgets (fid)

line=fgets (fid, nchar)

```

该命令的功能与 fgetl 基本相同，不同之处为读入时保留原文件中的换行符。nchar 是整数，为输入可选项，如果指定此项数值，则读入一行时，最多读 nchar 个字符。

fgets 函数应用实例 (exmp502. m) 。

假如磁盘中有一个名为 dout. dat 的数据文件，现在把其中两行分别读入 line1 和 line2, 程序为

```

did=fopen(' dout. dat', 'r');

line1=fgets (fid);

line2=fgets (fid) ;

```

```
fprlntf('start%s end', line1) ;
```

```
fprintf('start%s end',line2);
```

该程序把文件 dout. dat 中的第一行读入 line，第二行读入 line2，然后用两条 fprintf 显示。由于中 fprintf 中未指定输出的文件标识，所以输出到默认的标准文件，即显示器屏幕。dout. dat 文件是按行存放的数据文件，每行后有一换行符。读后显示的结果为

```
start xxx...x
```

```
endstart xxx...x
```

```
end
```

由于两句 fprintf 语句中都没有换行/n，所以第一个 end 和第二个 start 之间不出现换行。但是原文件中每个换行符都保留在 line1 和 line2 中，因此屏幕中显示的数据行结束时都出现了换行。这是 fgets 和 fgetl 的区别。