CHAPTER

04 三維立體繪圖

在科學目視表示法（Scientific Visualization）中，隱藏在資料內的資訊是要靠圖形的技巧來呈現，其中三度空間的立體繪圖是一個非常基本且重要的技巧。本章將介紹 MATLAB 在三度空間的各項立體繪圖指令。

4-1 基本的三維繪圖指令

4-2 圖形與圖軸的控制

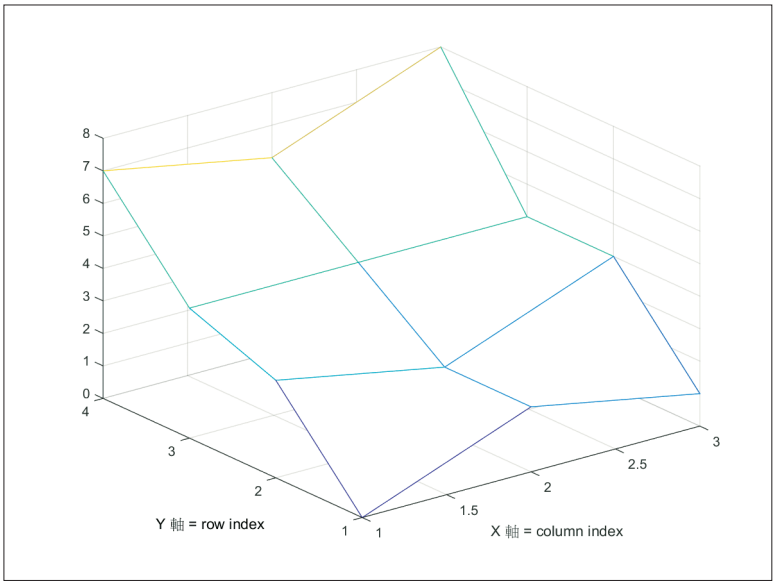
4-3 曲面顏色的控制

4-1 基本的三維繪圖指令

mesh 和 surf 是三度空間立體繪圖的基本指令，mesh 可畫出立體的「網狀圖」（Mesh Plots），surf 則可畫出立體的「曲面圖」（Surface Plots），兩者產生的圖形都會依高度而有不同顏色。下列範例畫出一個二維矩陣的立體網狀圖：

Example 1: 04- 三維立體繪圖 /plotxyz001.m

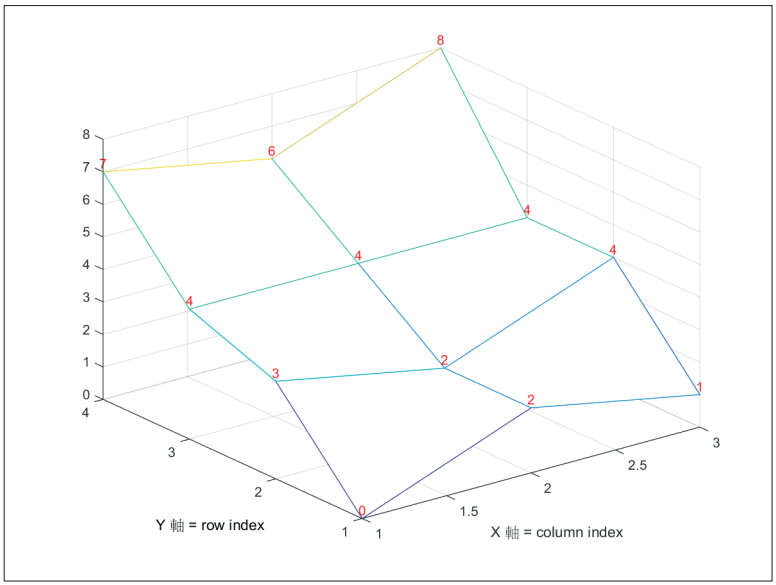
```
z = [0 2 1; 3 2 4; 4 4 4; 7 6 8];
mesh(z);
xlabel('X 軸 = column index');  % X 軸的說明文字
ylabel('Y 軸 = row index');     % Y 軸的說明文字
%colormap(zeros(1,3));          % 以黑色呈現
```



在上述範例中，X 軸對應到矩陣 z 的直行索引，Y 軸則對應到橫列索引，換句話說，在上述圖形中，當 x = j 且 y = i 時，所對應的高度是 z(i, j)，可以驗證如下：

Example 2: 04- 三維立體繪圖 /plotxyz002.m

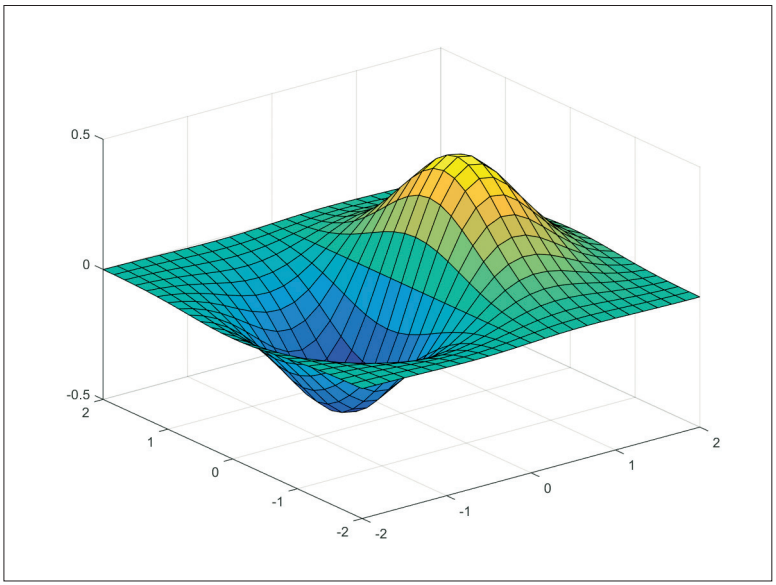
```
z = [0 2 1; 3 2 4; 4 4 4; 7 6 8];
mesh(z);
xlabel('X 軸 = column index');  % X 軸的說明文字
ylabel('Y 軸 = row index');     % Y 軸的說明文字

for i=1:size(z,1)
    for j=1:size(z,2)
        h=text(j, i, z(i,j), num2str(z(i, j)));     % 標示曲面高度
        set(h, 'hori', 'center', 'vertical', 'bottom', 'color', 'r');
                                                    % 改變位置及顏色
    end
end

%colormap(zeros(1,3));          % 以黑色呈現
```



surf 和 mesh 指令的用法類似，但產生的圖形是立體曲面圖：

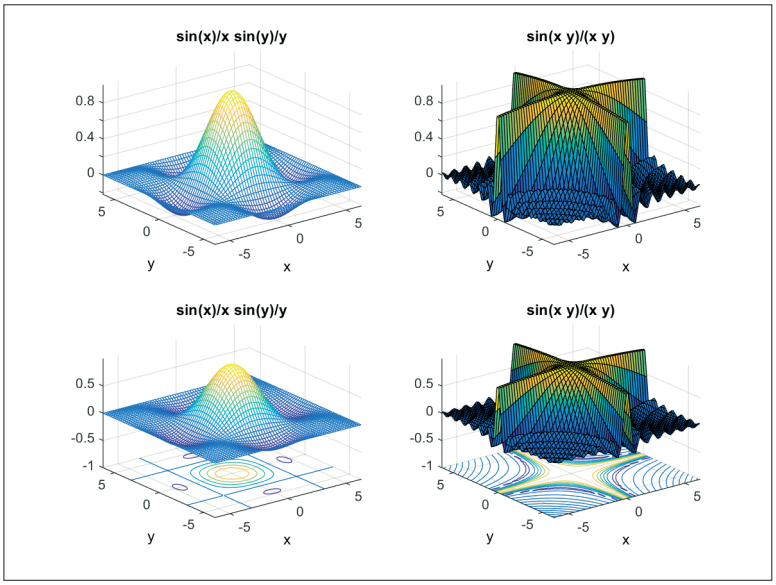
Example 5: 04- 三維立體繪圖 /plotxyz02.m

```
x = linspace(-2, 2, 25);          % 在 x 軸 [-2,2] 之間取 25 點
y = linspace(-2, 2, 25);          % 在 y 軸 [-2,2] 之間取 25 點
[xx,yy] = meshgrid(x, y);         % xx 和 yy 都是 25×25 的矩陣
zz = xx.*exp(-xx.^2-yy.^2);       % zz 也是 25×2 的矩陣
surf(xx, yy, zz);                 % 畫出立體曲面圖
colormap('default')               % 顏色改回預設值
```



為了方便測試立體繪圖，MATLAB 提供了一個 peaks 函數，可產生一個凹凸有致的曲面，包含了三個局部極大點（Local Maxima）及三個局部極小點（Local Minima），其方程式為：

$$z = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10(\frac{x}{5} - x^3 - y^5)e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2}$$



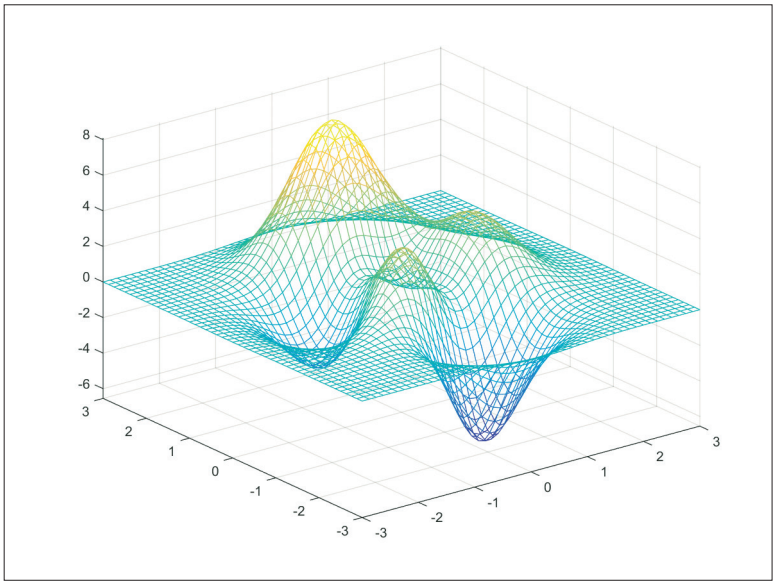
很難想像這些簡單的函數可以產生如此複雜的曲面吧？

4-2 圖形與圖軸的控制

在繪製網狀圖時，MATLAB 會隱藏被遮蓋的網線，若要使被遮蓋的網線亦能呈現出來，可用 hidden off 指令，如下：

Example 1: 04- 三維立體繪圖 /plotxyz10.m

```
[x,y,z] = peaks;
mesh(x,y,z);
hidden off
axis tight
```

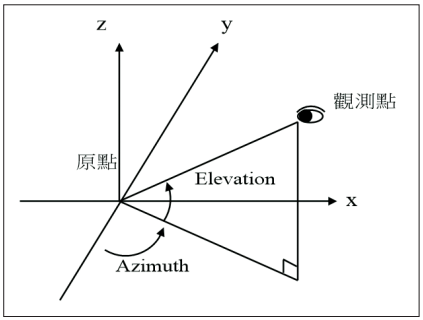


若再鍵入 hidden on，則恢復原先的設定。此種以 on 或 off 來進行兩段式切換的指令可列表如下。

指令	說明
hidden on/off	隱藏（on）或顯示（off）被遮蓋的網線。
zoom on/off	允許（on）或取消（off）以滑鼠點選來放大圖形。
rotate3d on/off	允許（on）或取消（off）以滑鼠點選來旋轉三維圖軸。
axis on/off	顯示（on）或不顯示（off）圖軸。
box on/off	顯示（on）或不顯示（off）圖軸的外框。
hold on/off	在繪製新圖時，保留（on）或消去（off）舊圖。
more on/off	允許（on）或不允許（off）指令視窗的輸出暫停。
echo on/off	允許（on）或不允許（off）指令在檔案內執行時，逐一顯示在視窗。

例如，若要能夠旋轉立體圖形，可以在產生 3D 圖形之後（例如輸入 peaks 之後），再輸入「rotate3d on」，此時您可以壓下滑鼠左鍵來拖曳圖軸，以選取最理想的觀測角度。（你也可以點選圖形視窗上面的 圖示，就可以開始旋轉立體圖形。）

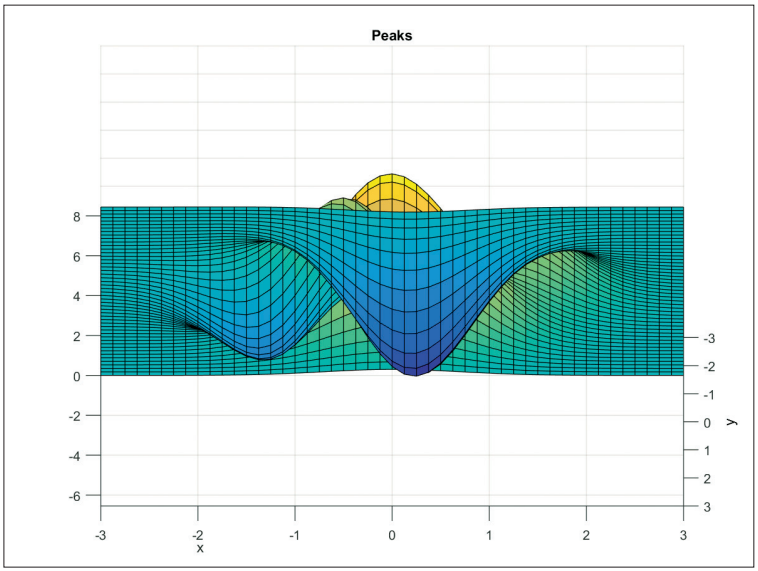
一般而言，三維曲線的觀測角度是由 Azimuth 及 Elevation 來決定，其示意圖如右：



對二維圖形而言，預設值為 Azimuth = 0°，Elevation = 90°；對三維圖形而言，預設值為 Azimuth = -37.5°，Elevation = 30°。若要改變觀測角度，可用 view 指令，如下：

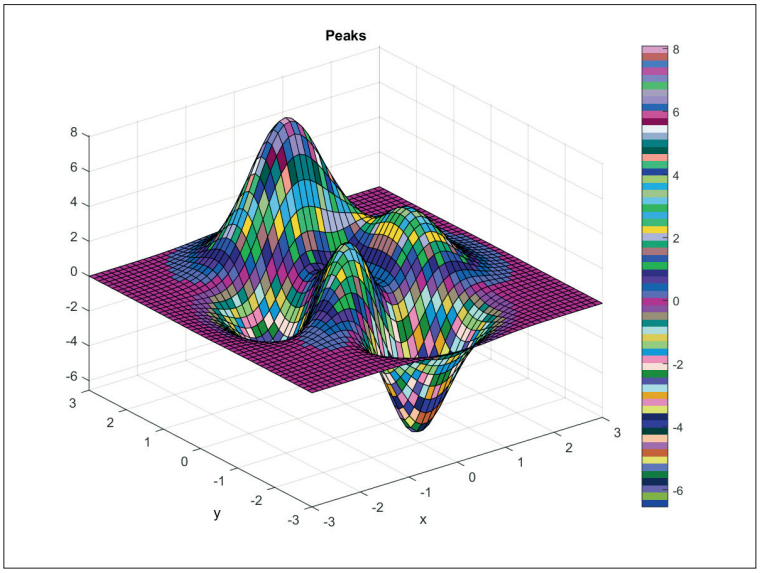
Example 2: 04- 三維立體繪圖 /plotxyz11.m

```
peaks;
colormap('default');          % 顏色改回預設值
view([0,-30]);

z =  3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...
   - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...
   - 1/3*exp(-(x+1).^2 - y.^2)
```



Example 3: 04- 三維立體繪圖 /plotxyz13.m

```
peaks;
colormap(rand(64,3));
colorbar;

z =  3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...
   - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...
   - 1/3*exp(-(x+1).^2 - y.^2)
```



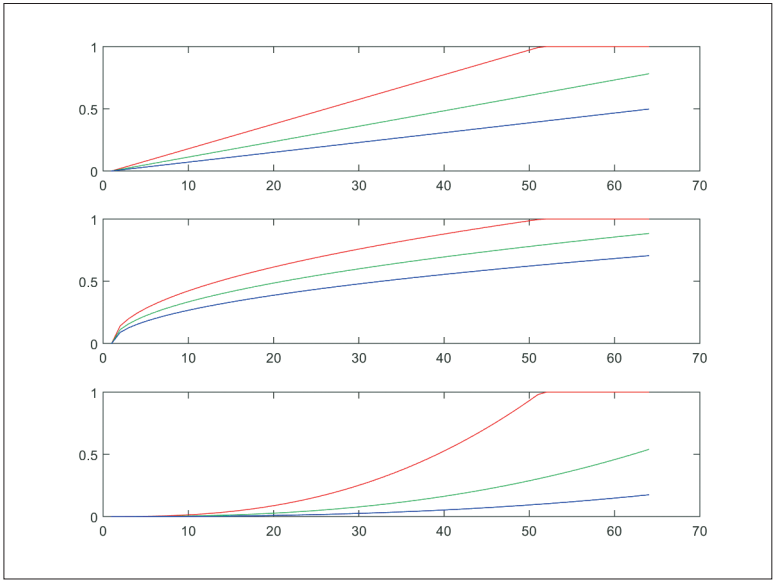
在上述範例中，我們使用亂數來產生一個 64×3 顏色對應表，因此曲面看起來並不是很賞心悅目。事實上，要產生一個好看的顏色對應表並不是件容易的事，因此 MATLAB 有一些現成的顏色對應表，整理如下：

指令	說明
colormap hsv	HSV 的顏色對應表（預設值）
colormap hot	代表“熱”的顏色對應表
colormap cool	代表“冷”的顏色對應表
colormap summer	代表“夏天”的顏色對應表
colormap gray	代表“灰階”的顏色對應表

我們可用 brighten 指令來使顏色對照表變亮或變暗，如下：

Example 7: 04- 三維立體繪圖 /plotxyz17.m

```
colormap copper
subplot(3, 1, 1); rgbplot(colormap);
brighten(colormap, 0.5)
subplot(3, 1, 2); rgbplot(colormap);
brighten(colormap, -0.8)
subplot(3, 1, 3); rgbplot(colormap);
```

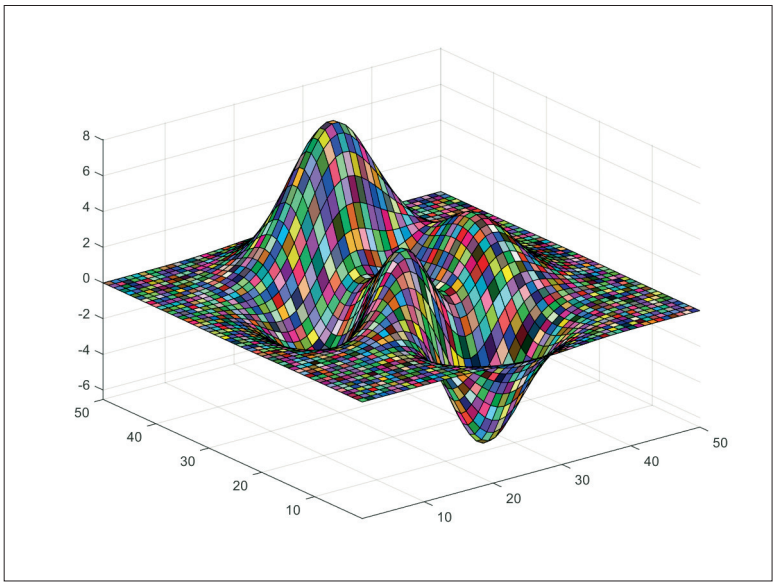


其中 rgbplot 可畫出 R、G、B 三色的比重對列數的變化。

上述 MATLAB 決定顏色的方法稱為「索引顏色」（Indexed Color）法，因為曲面上每個方塊先對應至顏色對應表的索引（即某一特定列），再由此列決定真正要用的顏色。如果您的顯示器能支援 24 位元全彩，則我們可以直接定義一千六百萬（224）種顏色，而不需要再定義顏色對應表。此種方法稱為「真實顏色」（True Color）法，如下：

Example 8: 04- 三維立體繪圖 /plotxyz18.m

```
Z = peaks(50);
C(:, :, 1) = rand(50);  % C(:,:,1) 代表 R（Red，紅色）的份量
C(:, :, 2) = rand(50);  % C(:,:,2) 代表 G（Green，綠色）的份量
C(:, :, 3) = rand(50);  % C(:,:,3) 代表 B（Blue，藍色）的份量
surf(Z, C);
axis tight
```

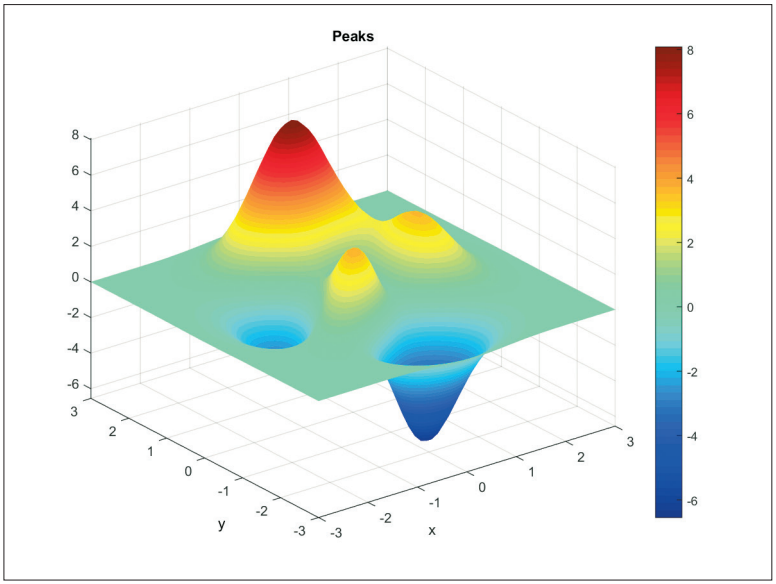


在上圖中，曲面上每一個小方塊的顏色都是由矩陣 C 的 R、G、B 份量來決定，這就是「真實顏色」，而和前述的「索引顏色」有所不同。

在前述的例子中，每一個曲面都可以視為由一塊塊的四角小「瓷磚」所拼成，而且每一塊「瓷磚」表面的顏色是均勻一致的，其顏色值由「瓷磚」所在的曲面高度所決定。若要使「瓷磚」表面的顏色產生連續性的變化，可使用 shading 指令，例如我們可以先輸入「peaks」，再輸入「shading interp」，就可以得到下列顏色漸進變化的圖形：

Example 9: 04- 三維立體繪圖 /shadingInterp01.m

```
peaks;
colormap jet;
colorbar;
shading interp;

z =  3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...
   - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...
   - 1/3*exp(-(x+1).^2 - y.^2)
```



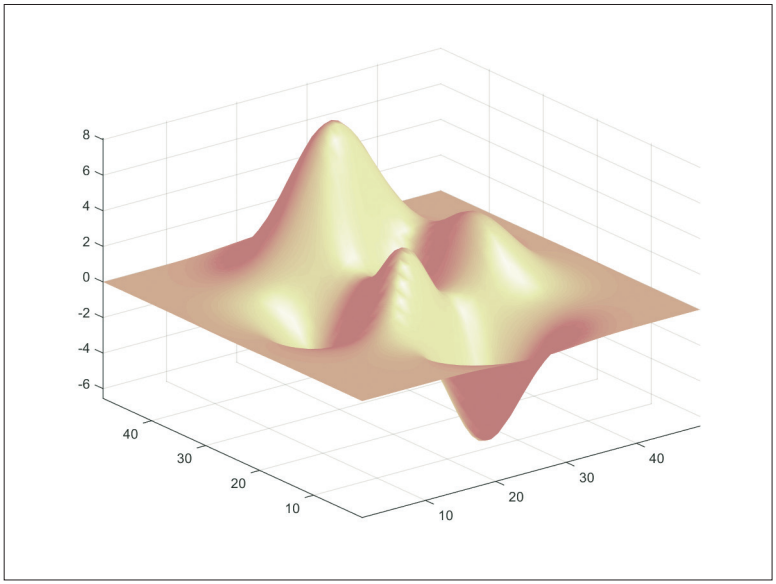
由上圖可看出，經由 shading interp 的作用，曲面的顏色更加柔順平滑。有關 shading 指令的用法，可列表如下：

指令	說明
shading interp	使用 Bilinear Interpolation 來使「瓷磚」根據四頂點的顏色產生連續的變化。
shading flat	「瓷磚」的顏色是均勻一致的。
shading faceted	「瓷磚」的顏色是均勻一致的，而且同時顯示「瓷磚」交接的「邊」。(此為預設值)

使用 colormap 及 shading，可產生意想不到的結果，例如：

Example 10: 04- 三維立體繪圖 /plotxyz19.m

```
surfl(peaks);                    % 曲面圖加上光源
axis tight
colormap(pink);
shading interp
```



上圖的曲面，是否很像一件光滑無瑕的雕塑品呢？！