

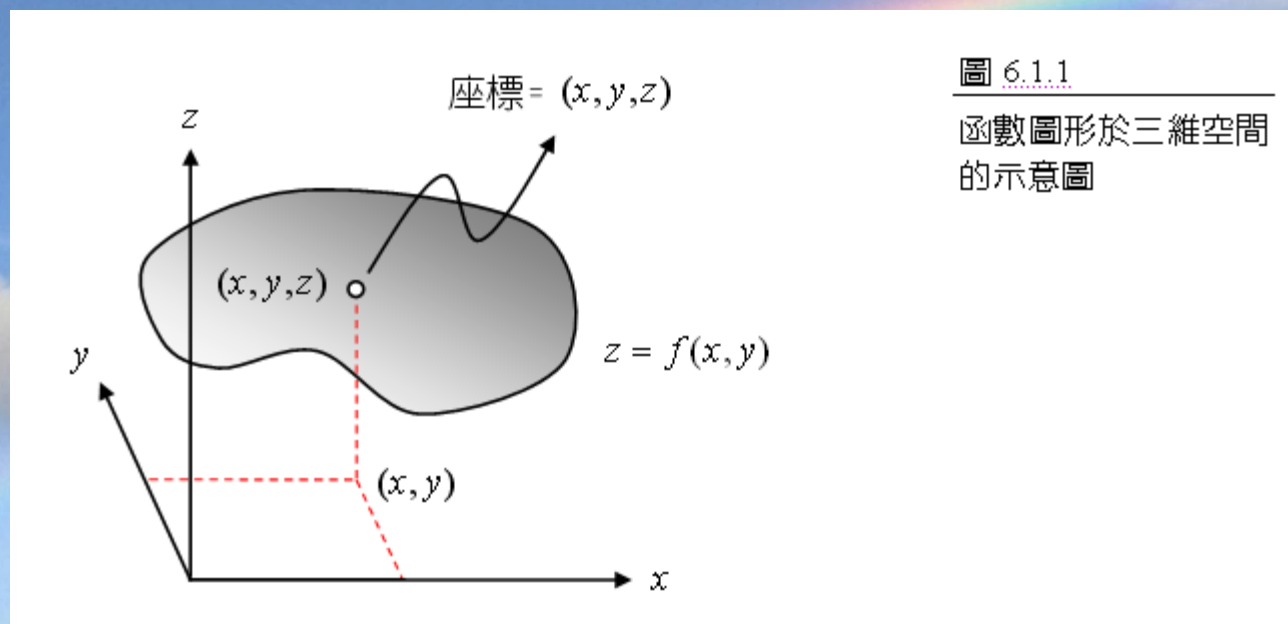
第六章 三維空間繪圖

本章學習目標

- 學習三維繪圖的基本技巧
- 學習peaks() 函數的用法
- 學習二維與三維等高線圖的繪製
- 學習三維圖形的編修

基本三維繪圖

- 對於函數 $f(x,y)$ 而言，每給一組 (x,y) ，便能計算 $z=f(x,y)$
- 只要給予 (x,y) 的組數夠多，即可繪出三維曲面圖



繪製三維的網格圖

表 6.1.1 `mesh()` 函數的使用

函 數	說 明
<code>mesh(x,y,z)</code>	分別以資料點的 x 、 y 與 z 座標之集合所組成的矩陣 <code>xx</code> 、 <code>yy</code> 與 <code>zz</code> 來繪出三維的網格圖
<code>mesh(z)</code>	設二維矩陣 z 的維度為 $m \times n$ ，則 <code>mesh(z)</code> 可繪出 x 座標從 1 到 n ， y 座標從 1 到 m 的三維的網格圖

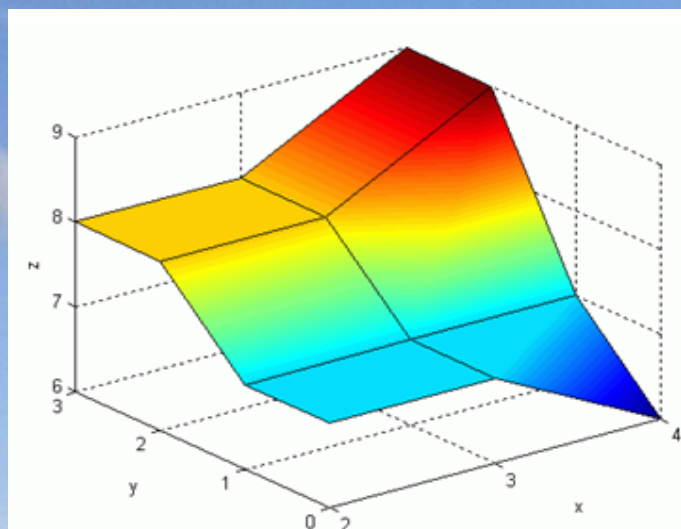
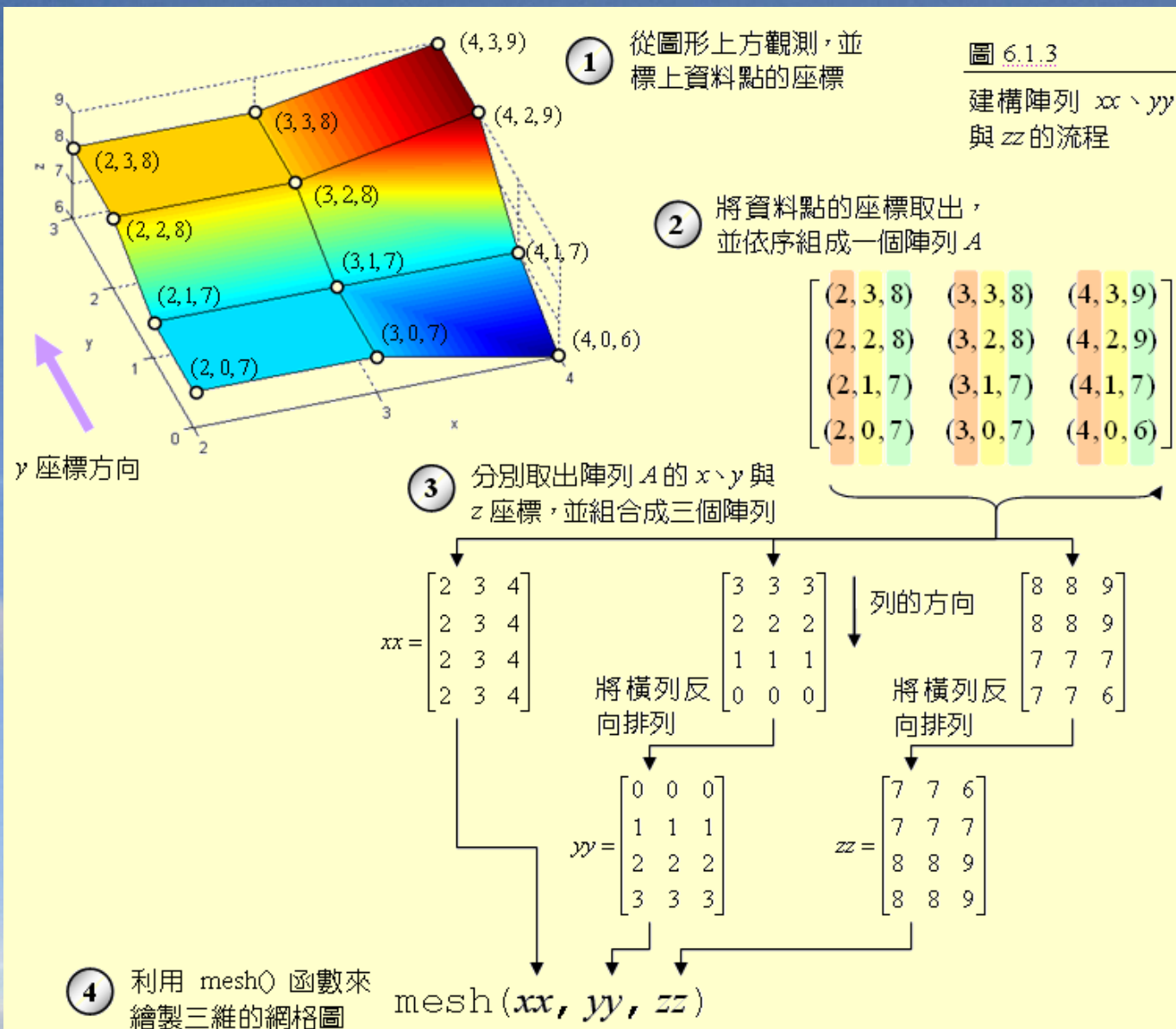


圖 6.1.2

具有 12 個資料點
的三維函數圖

	1	2	3	4	5	6	7	8	9	10	11	12
x	2	2	2	2	3	3	3	3	4	4	4	4
y	0	1	2	3	0	1	2	3	0	1	2	3
z	7	7	8	8	7	7	8	8	6	7	9	9

利用 mesh() 繪製三維的網格圖



mesh() 的使用範例

```
>> xx=[2 3 4;2 3 4;2 3 4;2 3 4]
```

```
xx =
```

2	3	4
2	3	4
2	3	4
2	3	4

```
>> yy=[0 0 0;1 1 1;2 2 2;3 3 3]
```

```
yy =
```

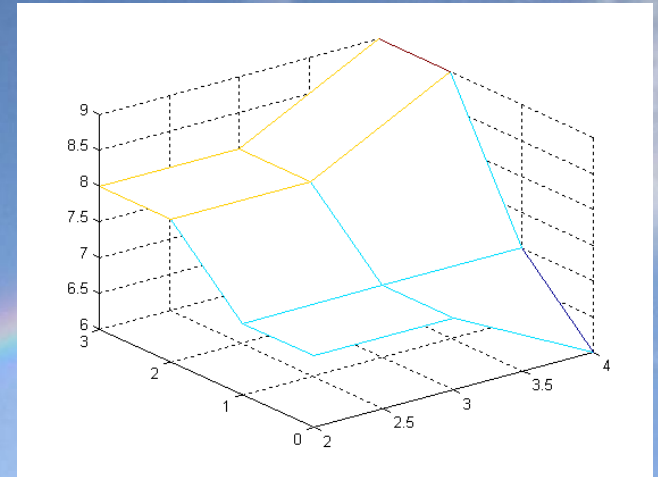
0	0	0
1	1	1
2	2	2
3	3	3

```
>> zz=[7 7 6;7 7 7;8 8 9;8 8 9]
```

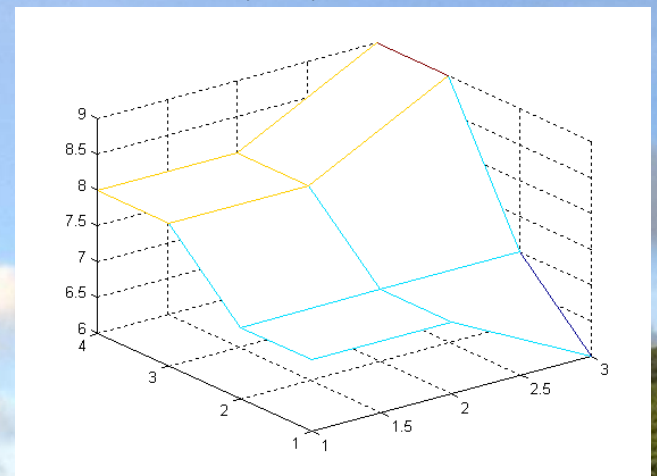
```
zz =
```

7	7	6
7	7	7
8	8	9
8	8	9

```
>> mesh(xx,yy,zz)
```



```
>> mesh(zz)
```



meshgrid() 函數

表 6.1.2 meshgrid() 的使用

函 數	說 明
meshgrid(vx,vy)	以向量 vx 代表 x 軸方向資料點的位置，以 vy 代表 y 軸方向資料點的位置，建構出兩個二維矩陣 xx 與 yy，以供三維繪圖所需

```
>> vx=2:4
```

```
vx =
```

```
2     3     4
```

```
>> vy=0:3
```

```
vy =
```

```
0     1     2     3
```

```
>> [xx,yy]=meshgrid(vx,vy)
```

```
xx =
```

```
2     3     4
```

```
2     3     4
```

```
2     3     4
```

```
2     3     4
```

```
yy =
```

```
0     0     0
```

```
1     1     1
```

```
2     2     2
```

```
3     3     3
```

```
>> zz=sqrt(xx.*yy)
```

```
zz =
```

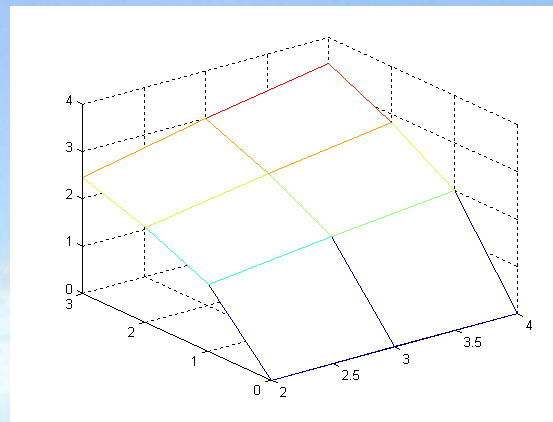
```
0     0     0
```

```
1.4142    1.7321    2.0000
```

```
2.0000    2.4495    2.8284
```

```
2.4495    3.0000    3.4641
```

```
>> mesh(xx,yy,zz)
```



meshc() 與 waterfall() 函數

表 6.1.3 meshc() 與 waterfall() 函數的使用

函 數	說 明
meshc(xx,yy,zz)	繪出網格圖，但在網格圖下方會附帶繪出等高線圖
waterfall(xx,yy,zz)	以切片的方式來繪製三維的立體圖

```
>> x=linspace(-8,8,30);
```

```
>> y=x;
```

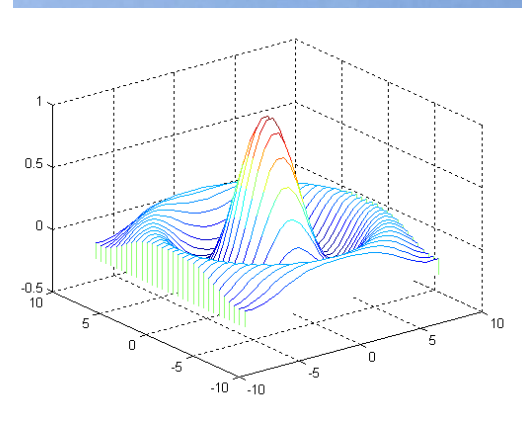
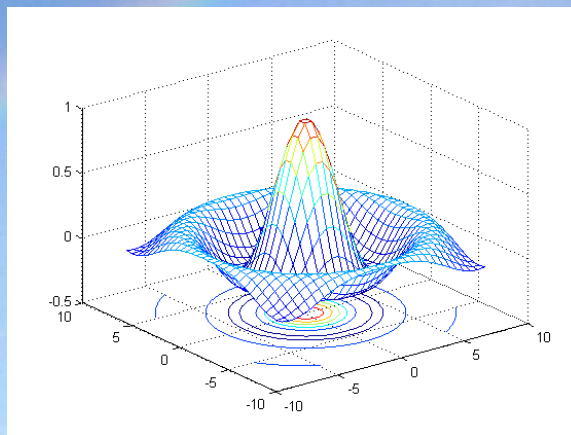
```
>> [xx,yy]=meshgrid(x,y);
```

```
>> expr=sqrt(xx.^2+yy.^2);
```

```
>> zz=sin(expr)./(expr+eps);
```

```
>> meshc(xx,yy,zz)
```

```
>> waterfall(xx,yy,zz)
```



Eps

Floating-point relative accuracy

eps returns the distance from 1.0 to the next largest

double-precision number, that is $\text{eps} = 2^{(-52)}$.

$$f(x, y) = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}, -8 \leq x \leq 8, -8 \leq y \leq 8$$

$$z = f(x, y) = x \cdot e^{-(x^2 + y^2)}, \quad -2 \leq x \leq 2, \quad -2 \leq y \leq 2$$

利用 meshgrid() 和 meshc() 繪出 $f(x, y)$ 三維函數圖形。

```
x = linspace(-2,2,30);  
y = linspace(-2,2,30);  
[xx,yy]=meshgrid(x,y);  
  
zz = xx.*exp(-xx.^2-yy.^2);  
  
meshc(xx,yy,zz)
```


三維曲面圖-surf() 與surfc()

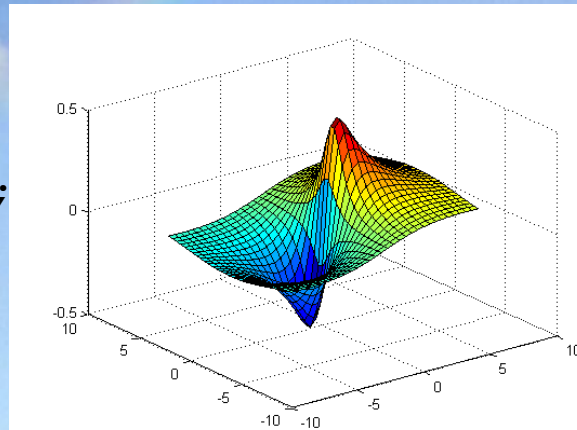
表 6.1.4 surf() 與 surfc() 函數的使用

函數	說明
surf (xx,yy,zz)	分別以資料點的 x 、 y 與 z 座標之集合所組成的矩陣 xx 、 yy 與 zz 來繪出三維的曲面圖
surfc(xx,yy,zz)	同 surf ，但在圖形下方會顯示出函數圖形的等高線圖

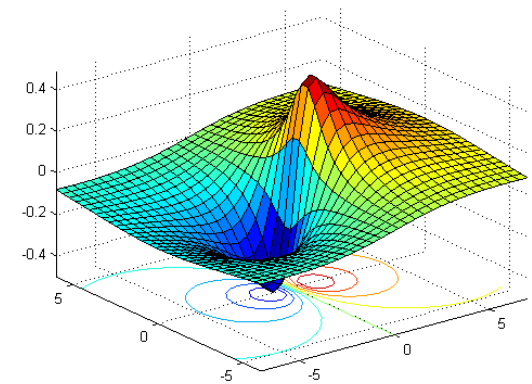
```
>> x=linspace(-7,7,32);  
>> y=linspace(-6,6,32);  
>> [xx,yy]=meshgrid(x,y);  
>> zz=xx./(xx.^2+yy.^2+1);
```

$$f(x,y) = \frac{x}{x^2 + y^2 + 1}, \quad -7 \leq x \leq 7, \quad -6 \leq y \leq 6$$

```
>> surf (xx,yy,zz);
```



```
>> surfc (xx,yy,zz);  
axis tight;
```



簡易的三維繪圖函數

- 只要給予繪圖的函數與範圍，`ezmesh()` 與 `ezsurf()` 函數便可快速的繪出三維的圖形

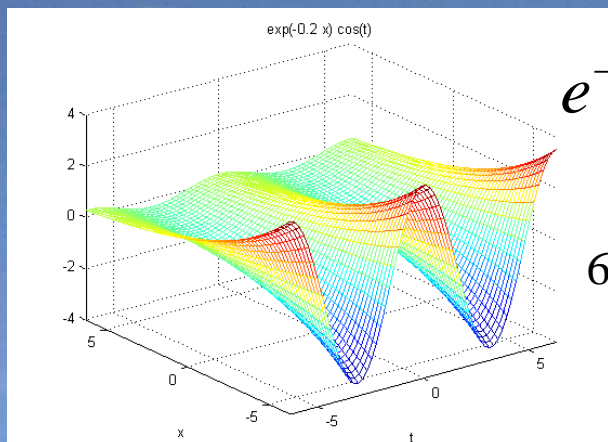
表 6.2.1 簡易三維繪圖函數的使用

函 數	說 明
<code>ezmesh(f, [xmin, xmax, ymin, ymax])</code>	根據函數 f (為一字串) 以 60×60 個網格數繪出 f 的三維圖形，若繪圖範圍省略，則預設 x 與 y 方向的範圍均是從 $-2\pi \sim 2\pi$
<code>ezmeshc(f, [xmin, xmax, ymin, ymax])</code>	同 <code>ezmesh</code> ，但在圖形下方會顯示出圖形的等高線
<code>ezsurf(f, [xmin, xmax, ymin, ymax])</code>	同 <code>ezmesh</code> ，但是網格面會上色
<code>ezsurfc(f, [xmin, xmax, ymin, ymax])</code>	同 <code>ezsurf</code> ，但在圖形下方會顯示出圖形的等高線

ezmesh() 與ezsurf() 的使用範例

$$-2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$$

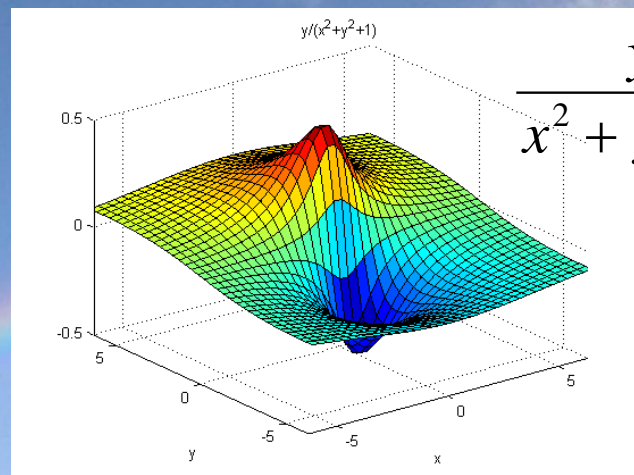
```
>> ezmesh('exp(-0.2*x)*cos(t)')
```



$$e^{-0.2} \cos t$$

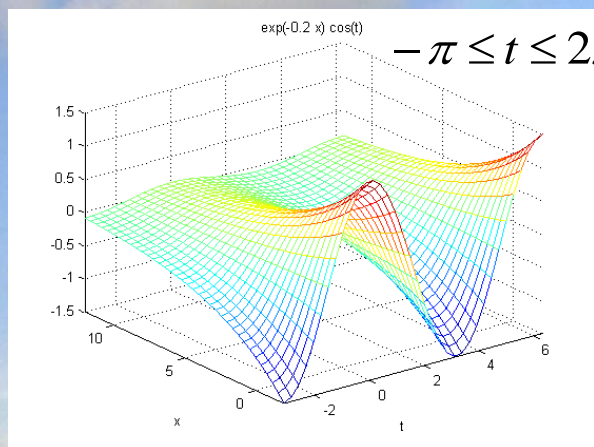
60x60

```
>> ezsurf('y/(x^2+y^2+1)',36)
```



$$\frac{y}{x^2 + y^2 + 1}$$

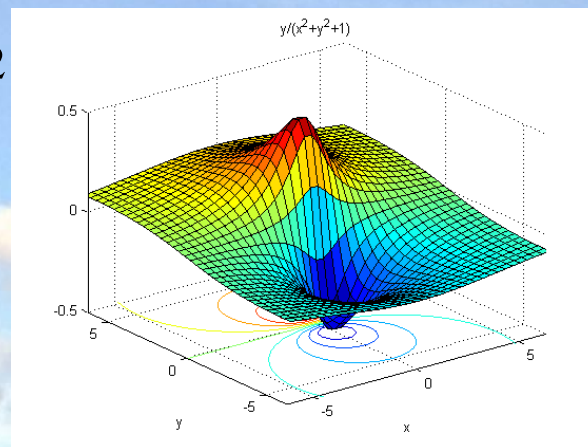
```
>> ezmesh('exp(-0.2*x)*cos(t)',...  
[-pi,2*pi,-2,12],36)
```



$$-\pi \leq t \leq 2\pi, \quad -2 \leq x \leq 12$$

36x36

```
>> ezsurf('y/(x^2+y^2+1)',36)
```



三維圖形展示函數—peaks

$$f(x, y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2}$$

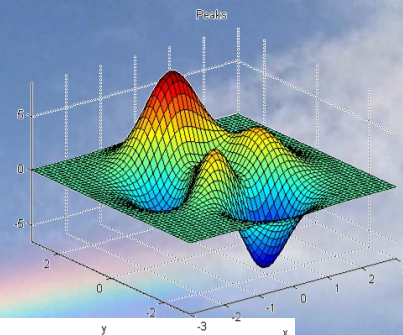


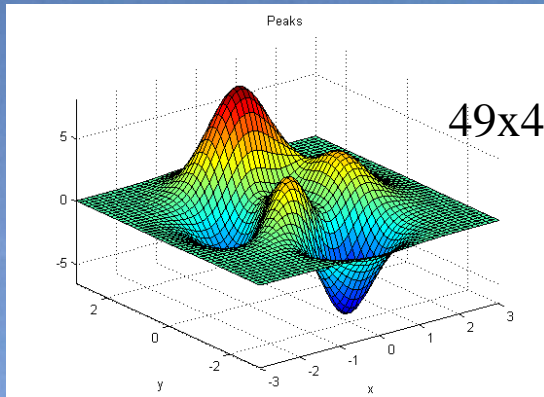
表 6.3.1 使用 **peaks** 函數

函 數	說 明
<code>peaks</code>	以 49×49 個資料點繪製數學函數 peaks ，範圍 x 與 y 方向同為 $-3 \sim 3$
<code>peaks(n)</code>	同 peaks ，但以 $n \times n$ 個資料點來繪圖
<code>zz=peaks</code>	在 x 與 y 方向同為 $-3 \sim 3$ 的範圍內計算 49×49 個數學函數 peaks 的值，並把結果設定給矩陣 zz 存放
<code>zz=peaks(n)</code>	以 $n \times n$ 個資料點計算數學函數 peaks 的值
<code>[xx,yy,zz]=peaks(n)</code>	以 $n \times n$ 個資料點計算數學函數 peaks 的值，並把資料點的 x 、 y 座標值與函數值分別存放在矩陣 xx 、 yy 與 zz 內

Peaks的語法

```
>> peaks;
```

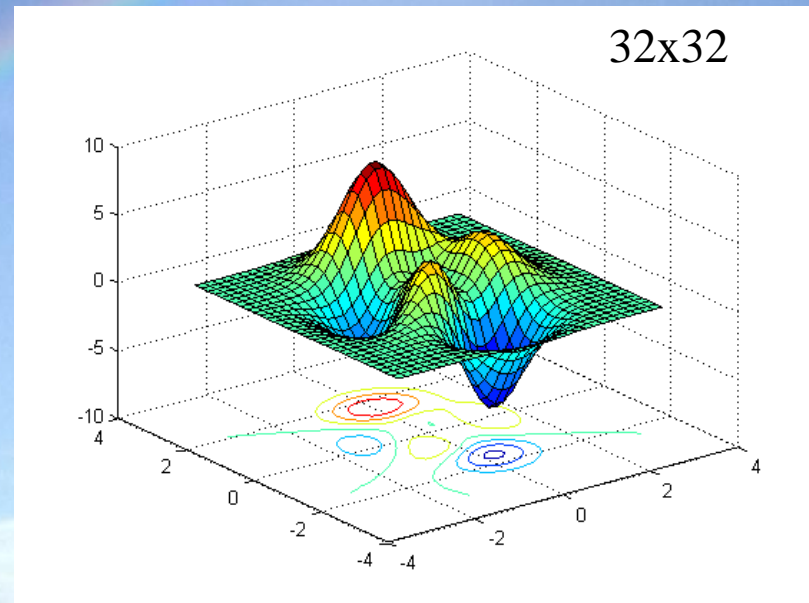
```
z=3*(1-x).^2.*exp(-(x.^2)-(y+1).^2)...  
-10*(x/5-x.^3-y.^5).*exp(-x.^2-y.^2)...  
-1/3*exp(-(x+1).^2-y.^2)
```



49x49

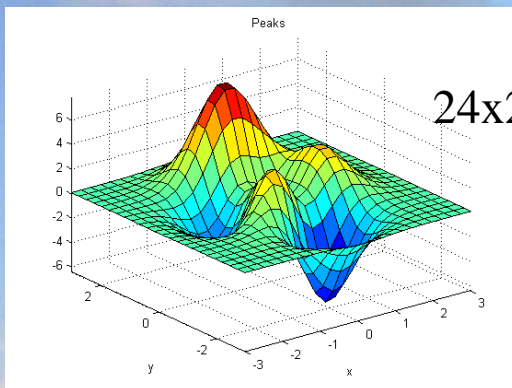
```
>> [xx,yy,zz]=peaks(32);
```

```
>> surfc(xx,yy,zz);
```



32x32

```
>> peaks(24);
```



24x24

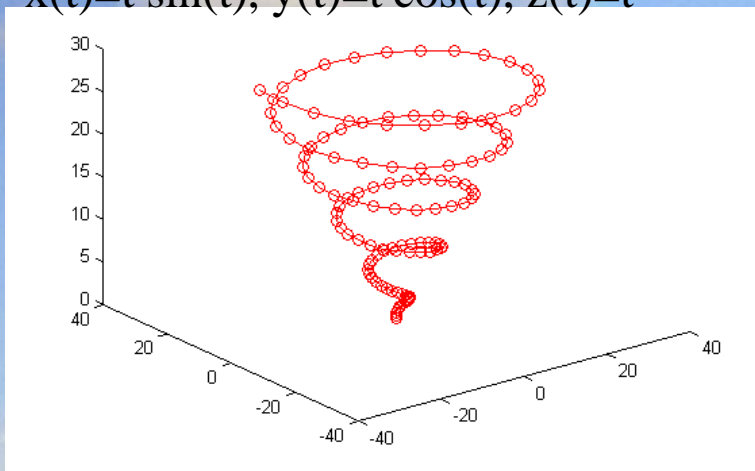
空間曲線繪圖

表 6.4.1 空間曲線繪圖函數

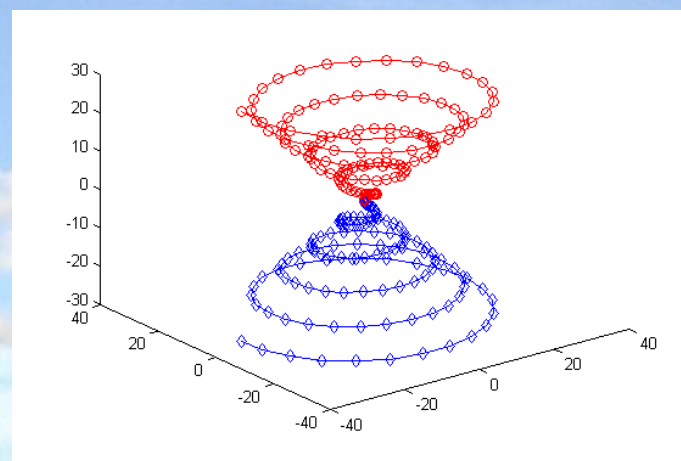
函 數	說 明
<code>plot3(x,y,z)</code>	分別以向量 x, y 與 z 代表資料點在每一個座標軸的位置，繪製三維空間曲線
<code>plot3(x,y,z,'str')</code>	以控制字串 str 所指定的格式繪出三維空間曲線

```
>> t=linspace(0,30,120);  
>> plot3(t.*sin(t),t.*cos(t),t,'-ro');
```

$x(t)=t \sin(t)$, $y(t)=t \cos(t)$, $z(t)=t$



```
>> plot3(t.*sin(t),t.*cos(t),t,'-ro',  
         t.*sin(t),t.*cos(t),-t,'-bd');
```



等高線繪圖

- 等高線圖（contour plot）是把三維的圖中，高度相等的點連接起來，即成等高線圖

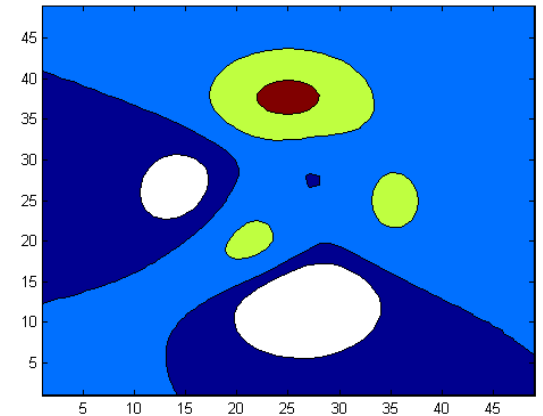
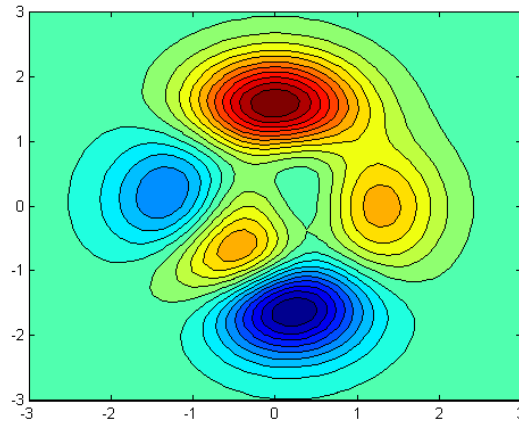
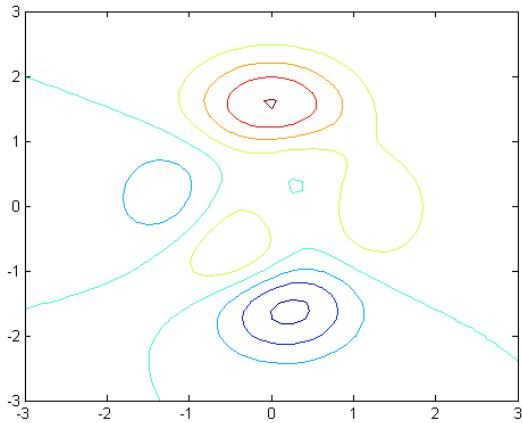
表 6.5.1 二維等高線繪圖函數

函 數	說 明
<code>contour(xx,yy,zz,n)</code>	分別以資料點的 x 、 y 與 z 座標之集合所組成的矩陣 xx 、 yy 與 zz 繪出 n 條等高線。若 n 省略，則 Matlab 會自動視情況調整等高線數。
<code>contour(zz,n)</code>	同上，但 x 方向的座標是從 1 到 n ， y 方向的座標是從 1 到 m （假設矩陣 zz 的維度為 $m \times n$ ）
<code>contour(xx,yy,zz,[z₁,z₂,z₃,...])</code>	繪出高度為 z_1, z_2, z_3, \dots 的等高線圖
<code>contourf(xx,yy,zz,n)</code>	同 <code>contour()</code> ，但會以顏色填滿（ fill ）等高線圖

等高線繪圖的範例

```
>> [xx,yy,zz]=peaks;
```

```
>> contour(xx,yy,zz) >> contourf(xx,yy,zz,20) >> contourf(zz,[-2,0,3,7])
```



標註等高線的值

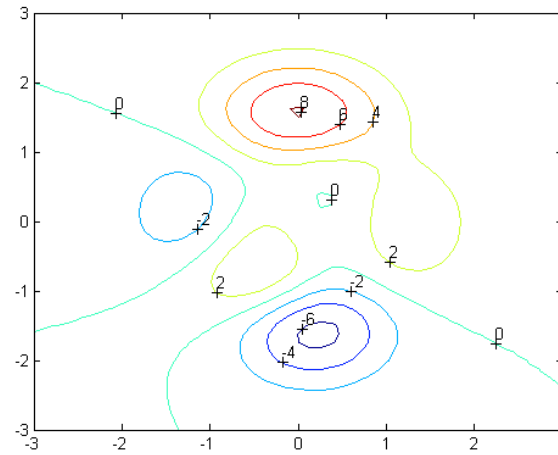
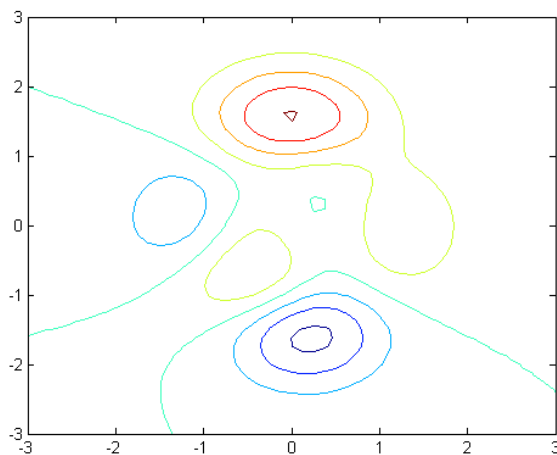
表 6.5.2 將等高線加入高度標記的函數

函 數	說 明
<code>clabel(cmat)</code>	在等高線圖內加上高度的標記，其中 <i>cmat</i> 為繪製等高線圖時， <code>contour()</code> 所傳回的矩陣
<code>clabel(cmat, [z₁, z₂, z₃, ...])</code>	在高度為 [z ₁ , z ₂ , z ₃ , ...] 的等高線上加上高度標記
<code>clabel(cmat, 'manual')</code>	可利用滑鼠點選欲標註之等高線，在該等高線旁即會出現等高線的數值 結束滑鼠操作：按Enter

```
>> [xx,yy,zz]=peaks;
```

```
>> cmat=contour(xx,yy,zz);
```

```
>> clabel(cmat)
```



三維的等高線圖

表 6.5.3 三維等高線繪圖函數

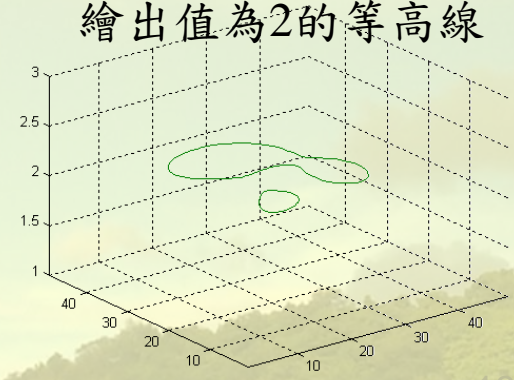
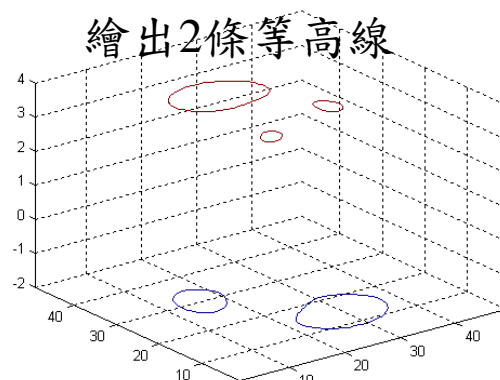
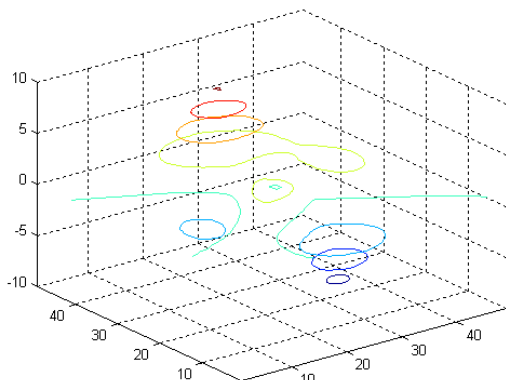
函 數	說 明
<code>contour3 (xx,yy,zz,n)</code>	分別以矩陣 xx 、 yy 與 zz 繪出 n 條三維的等高線。若 n 省略，則 Matlab 會自動視情況調整等高線數。
<code>contour3 (zz,n)</code>	同上，但 x 方向的座標是從 1 到 n ， y 方向的座標是從 1 到 m （假設矩陣 zz 的維度為 $m \times n$ ）
<code>contour3 (xx,yy,zz,[z₁,z₂,z₃,...])</code>	指定繪出高度為 z_1, z_2, z_3, \dots 的三維等高線圖

```
>> zz=peaks;
```

```
>> contour3(zz);
```

```
>> contour3(zz,[2]);
```

```
>> contour3(zz,[2,2]);
```



依資料繪出適量的等高線數

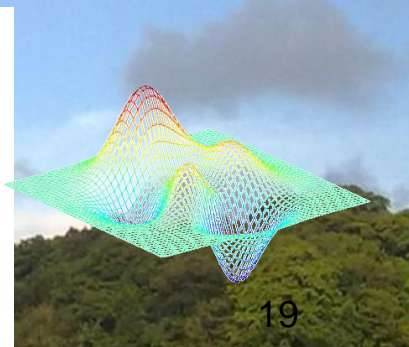
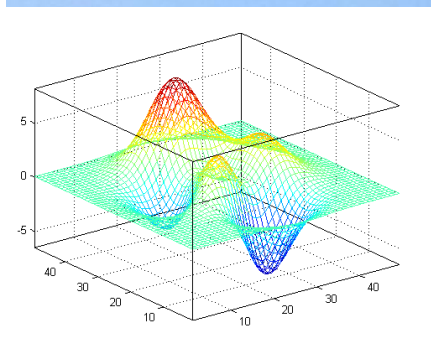
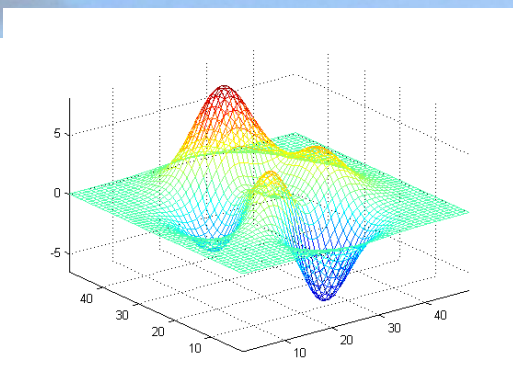
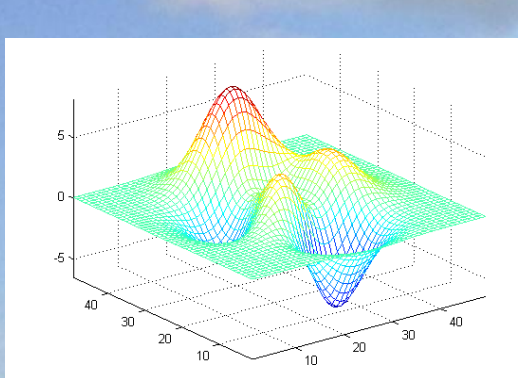
三維圖形的基本編修

表 6.6.1 三維繪圖的基本編修指令

指令	說明
<code>hidden on/off</code>	預設為 on 。設定 off 則會顯示隱藏線，但這個指令只對 <code>mesh()</code> 等函數所繪出的網格圖形有效（隱藏線是指被網格面遮住的線）
<code>axis on/off</code>	預設為 on 。設定 off 則不顯示座標軸與刻度
<code>box on/off</code>	預設為 off 。設定 on 則在圖形的外圍顯示一個外框
<code>hold on/off</code>	預設為 off 。設定 on 時，新產生的圖形不會覆蓋掉原有的圖形
<code>grid on/off</code>	設定 on 則顯示座標的網格線

>> **zz=peaks;**

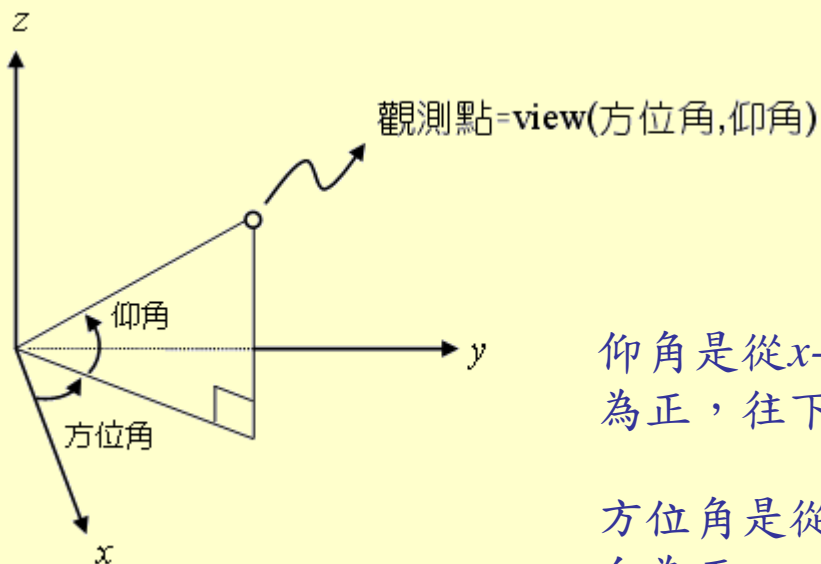
>> **mesh(zz);axis tight; >> hidden off; >> box on; >> axis off;**



改變三維圖形的視角

表 6.6.2 改變三維圖形的視角

函 數	說 明
<code>view(<i>az</i>,<i>el</i>)</code>	設定圖形的視角，其中方位角為 <i>az</i> ，仰角為 <i>el</i> ，單位為度
<code>[<i>az</i>,<i>el</i>]=view</code>	傳回目前所使用的視角

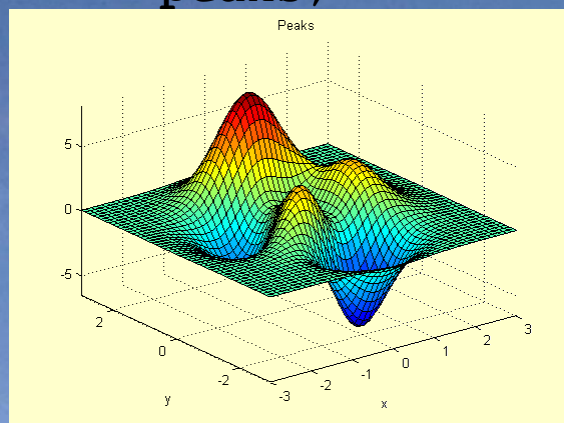


仰角是從 x - y 平面上數起，往上為正，往下為負

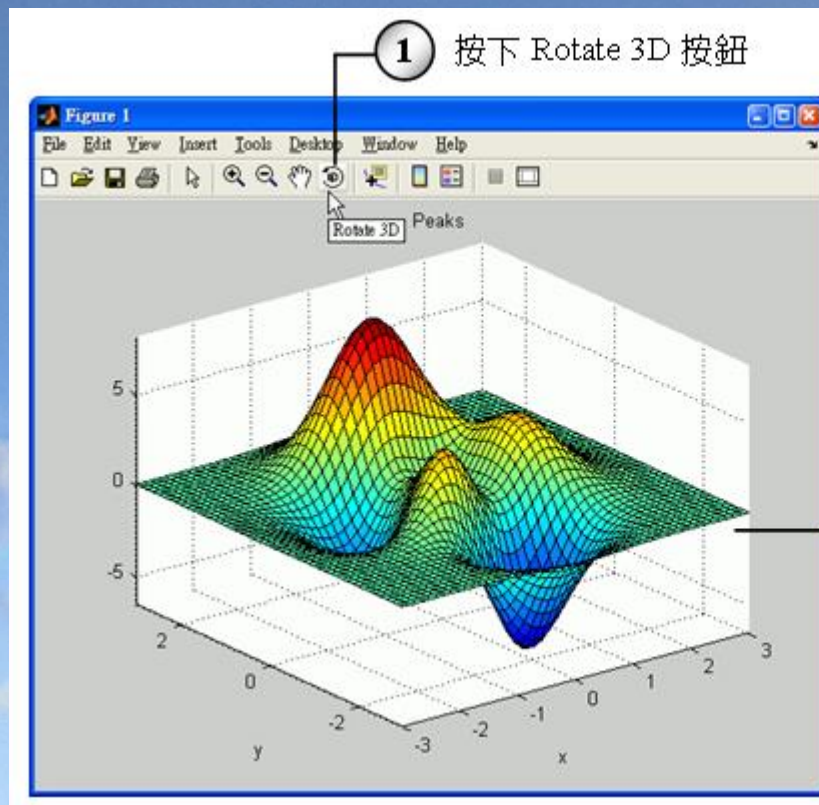
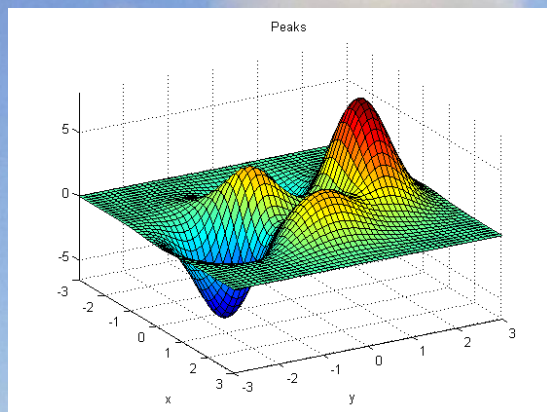
方位角是從 x 軸數起，逆時針方向為正

view() 的使用範例

```
>> peaks;
```



```
>> view(60,30);
```



1 按下 Rotate 3D 按鈕

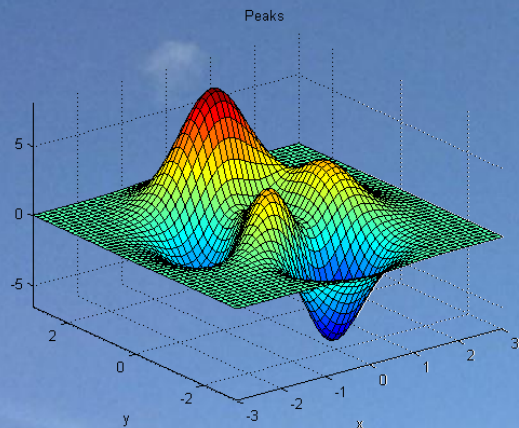
圖 6.6.2

您也可以利用工具列上所提供的按鈕來旋轉圖形

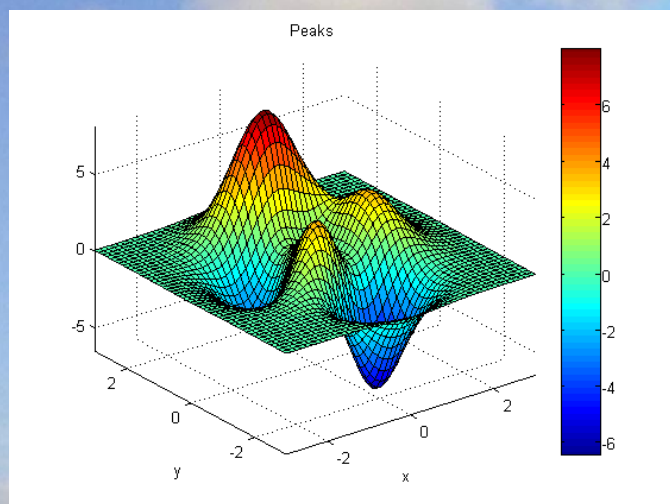
2 利用滑鼠拖曳來旋轉圖形

修改三維圖形的曲面顏色

```
>> peaks;
```



```
>> colorbar;
```



```
>> colormap
```

```
ans =
```

```

      0      0      0.5625
      0      0      0.6250
      0      0      0.6875
      0      0      0.7500
      0      0      0.8125
      0      0      0.8750
      0      0      0.9375
      :      :      :
0.5625      0      0
0.5000      0      0
    
```

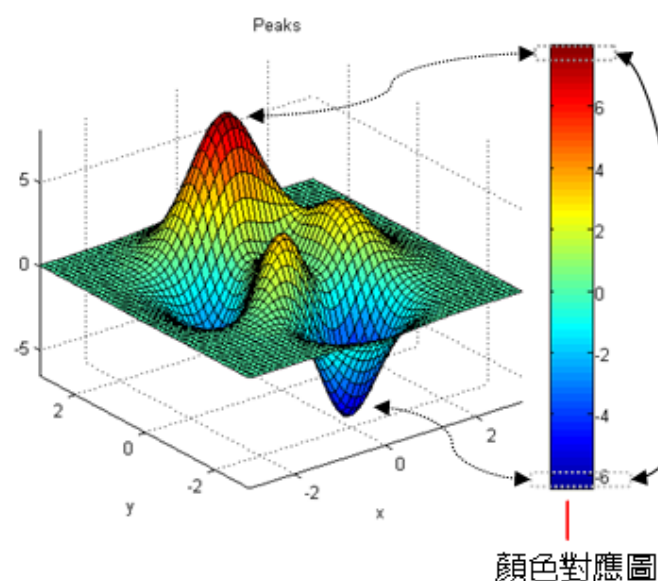
表 6.6.3 常用的 RGB 顏色

顏色	紅色 (red)	綠色 (green)	藍色 (blue)
紅色 (red)	1	0	0
綠色 (green)	0	1	0
藍色 (blue)	0	0	1
黃色 (yellow)	1	1	0
洋紅色 (magenta)	1	0	1
青色 (cyan)	0	1	1
灰色 (gray)	0.5	0.5	0.5
暗紅色 (dark red)	0.5	0	0
黑色 (black)	0	0	0
白色 (white)	1	1	1

colormap() 函數的使用

表 6.6.4 colormap() 函數的使用

函 數	說 明
<code>colormap(map)</code>	使用 <i>map</i> 當成目前配色的顏色對應表
<code>colormap('default')</code>	使用預設的顏色對應表
<code>map=colormap</code>	把目前的顏色對應表設定給變數 <i>map</i>
<code>colorbar</code>	在目前的圖形中顯示顏色對應圖



colorbar 所呈現的即是 colormap 裡每一列 RGB 顏色之混色

圖 6.6.3

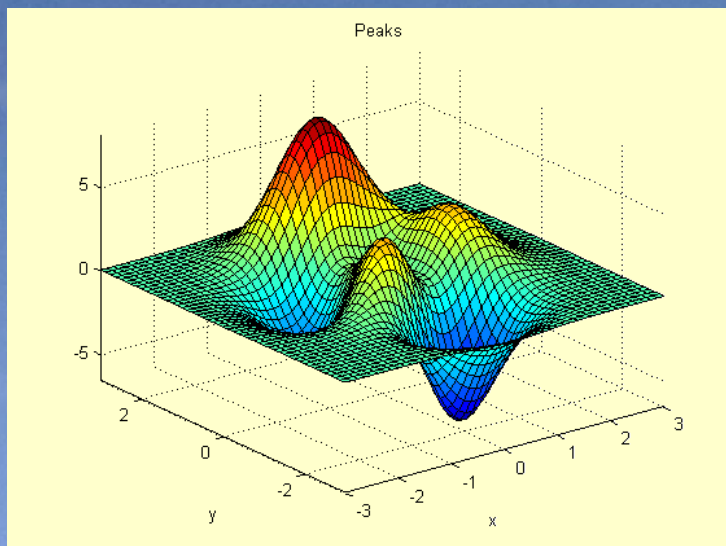
colorbar 與 colormap 之間的對應關係

0	0	0.5625
0	0	0.6250
0	0	0.6875
0	0	0.7500
0	0	0.8125
0	0	0.8750
0	0	0.9375
0	0	1.0000
0	0.0625	1.0000
0	0.1250	1.0000
⋮	⋮	⋮
0.5625	0	0
0.5000	0	0

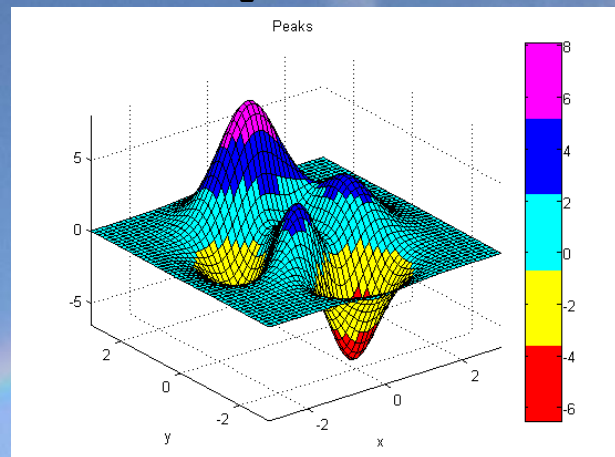
顏色對應表

colormap() 的使用

```
>> peaks;
```



```
>> colormap(cm);colorbar;
```



```
>> cm=[1 0 0;1 1 0;0 1 1;  
       0 0 1;1 0 1]
```

cm =

```
1     0     0  
1     1     0  
0     1     1  
0     0     1  
1     0     1
```

表 6.6.3 常用的 RGB 顏色

顏色	紅色 (red)	綠色 (green)	藍色 (blue)
紅色 (red)	1	0	0
綠色 (green)	0	1	0
藍色 (blue)	0	0	1
黃色 (yellow)	1	1	0
洋紅色 (magenta)	1	0	1
青色 (cyan)	0	1	1
灰色 (gray)	0.5	0.5	0.5
暗紅色 (dark red)	0.5	0	0
黑色 (black)	0	0	0
白色 (white)	1	1	1

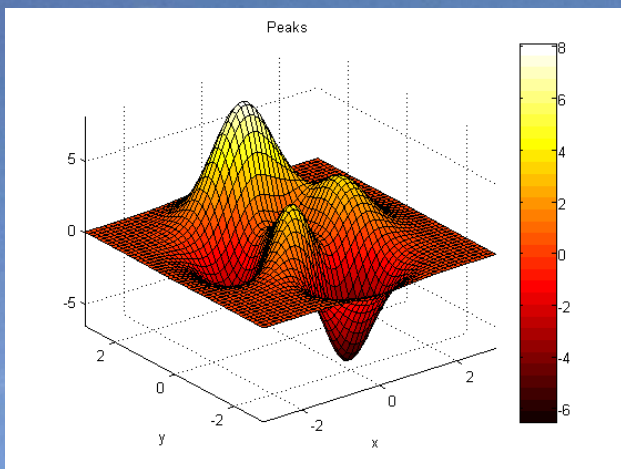
產生顏色對應表的函數

表 6.6.5 產生顏色對應表的函數

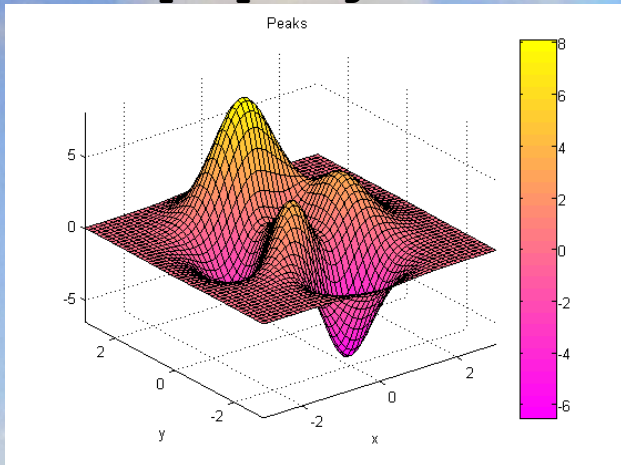
函 數	說 明
<code>hsv(m)</code>	建立一個 $m \times 3$ 的顏色對應矩陣，色系是由紅、橙、黃、綠、藍、靛、紫等循環色彩所組成
<code>jet(m)</code>	建立一個 $m \times 3$ 的顏色對應矩陣，色系是暗紅、紅、橙、黃、綠、藍、靛、紫與暗藍等色彩所組成（ Matlab 預設的顏色對應表）
<code>spring(m)</code>	建立一個 $m \times 3$ 的春天色系矩陣，它是由粉紅與黃色色系所組成
<code>summer(m)</code>	建立一個 $m \times 3$ 的夏天色系矩陣，它是由綠色與黃色色系所組成
<code>autumn(m)</code>	建立一個 $m \times 3$ 的秋天色系矩陣，它是由黃色與紅色色系所組成
<code>winter(m)</code>	建立一個 $m \times 3$ 的冬天色系矩陣，它是由藍色與綠色色系所組成
<code>hot(m)</code>	建立一個 $m \times 3$ 的暖色系矩陣，由黑、紅、黃、白等顏色所組成
<code>cool(m)</code>	建立一個 $m \times 3$ 的冷色系矩陣，由青色和暗紅色等顏色所組成
<code>gray(m)</code>	建立一個 $m \times 3$ 的灰階色系矩陣

顏色對應表函數的使用

```
>> colormap(hot(32));colorbar;
```



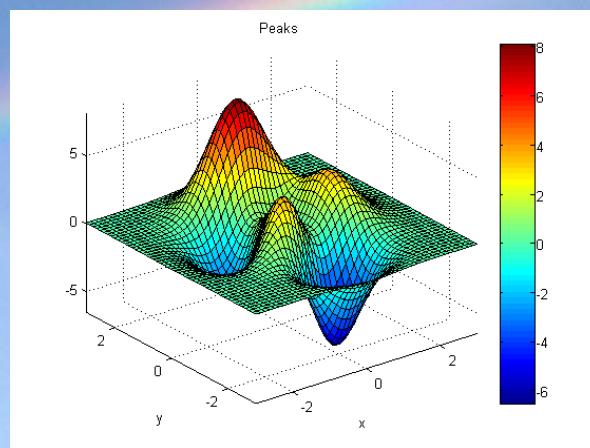
```
>> colormap spring;colorbar;
```



```
>> size(spring)
```

```
ans =  
3
```

```
>> colormap('default');colorbar;
```



```
>> size(colormap)
```

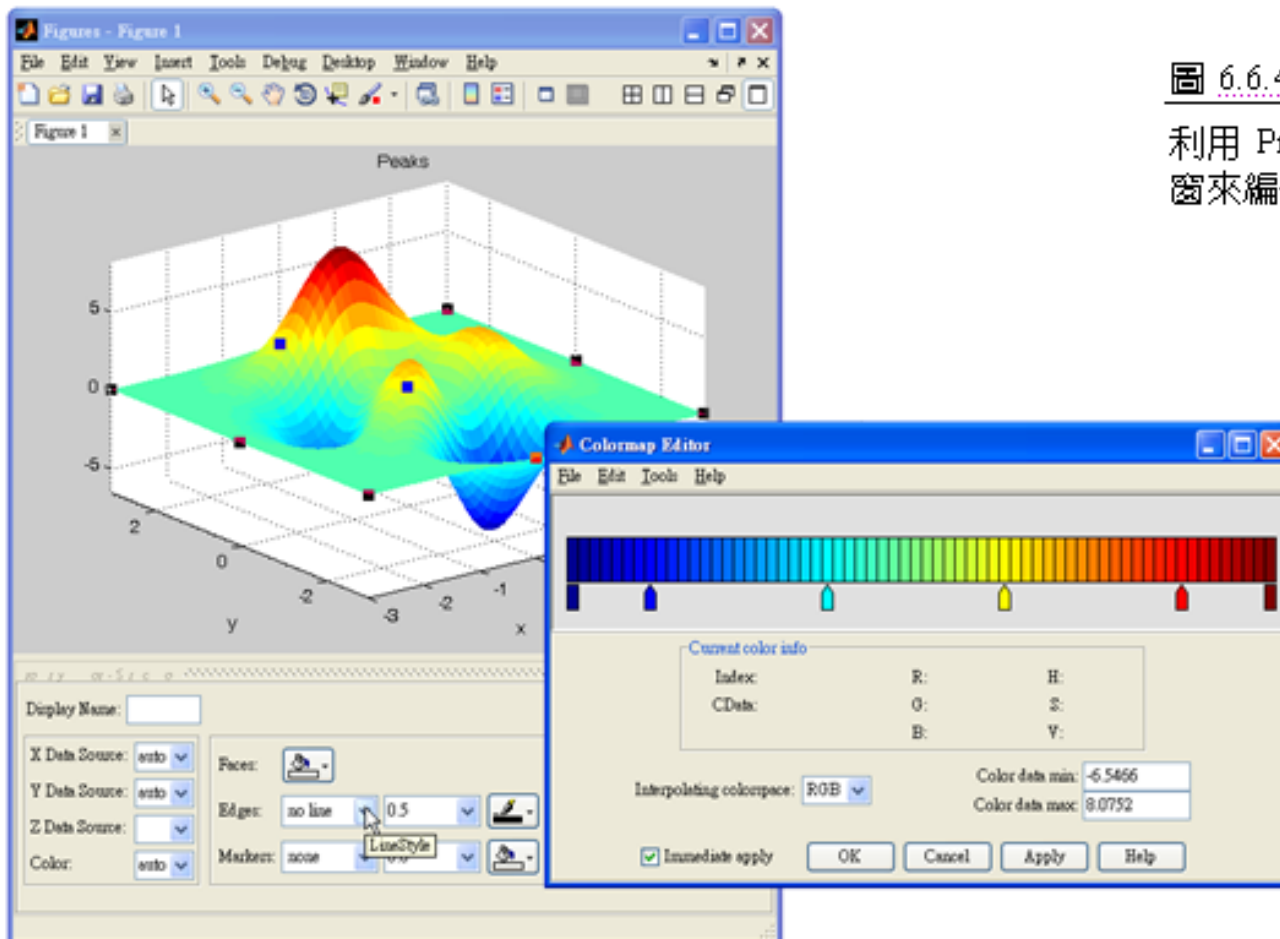
```
ans =  
64    3
```

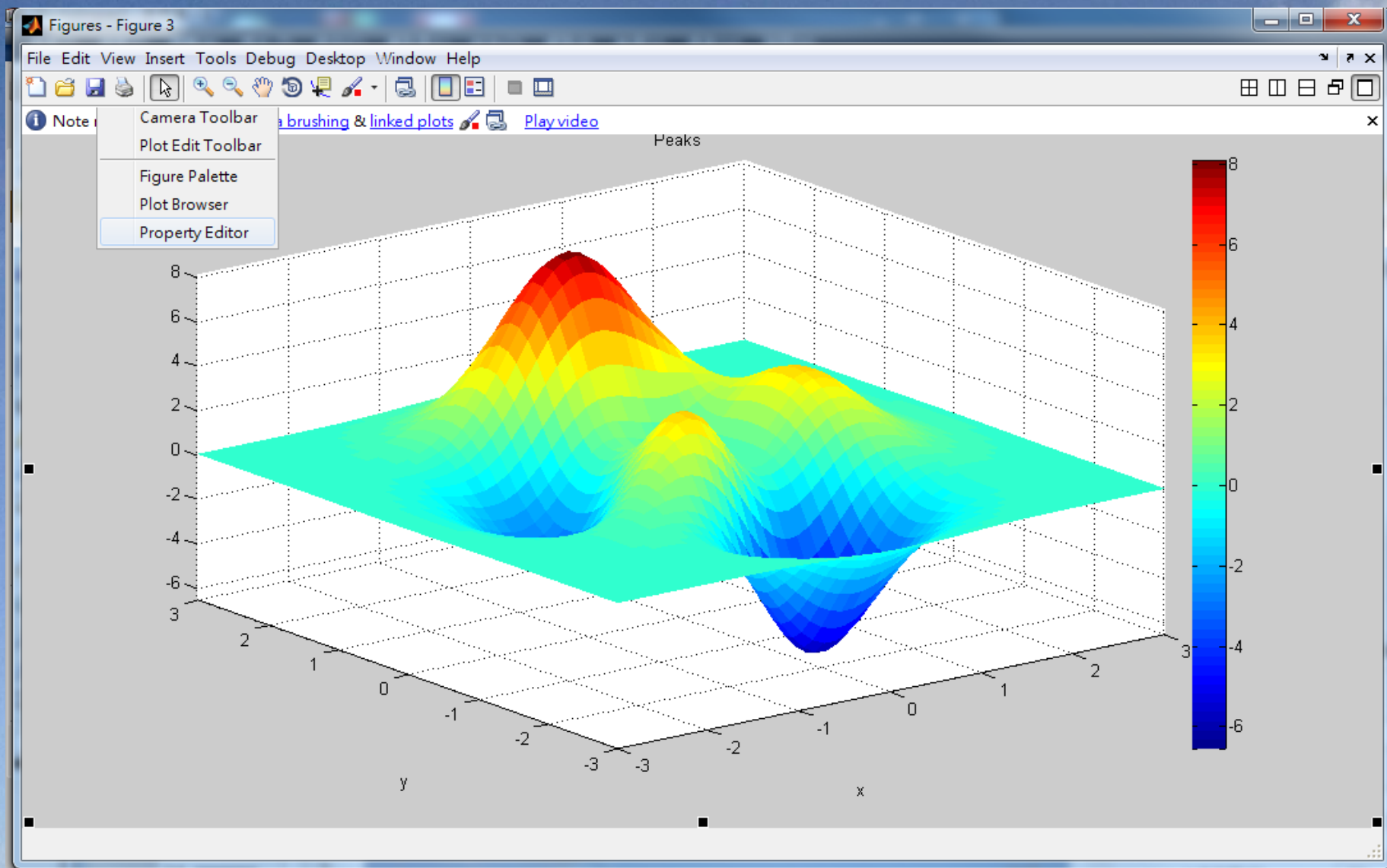

Colormap Editor對話方塊

- 從Edit選單中選擇Colormap來編輯顏色對應表

圖 6.6.4

利用 Property Editor 視窗來編修三維的圖形



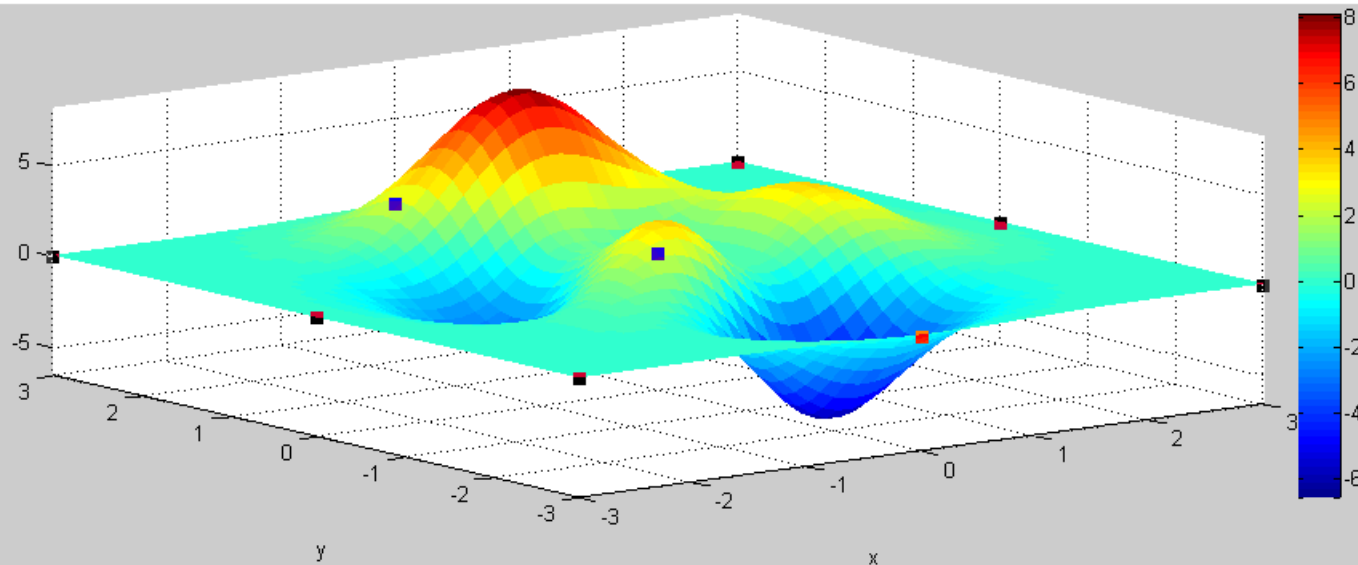


Figures - Figure 3

File Edit View Insert Tools Debug Desktop Window Help



Note new toolbar buttons: [data brushing](#) & [linked plots](#) [Play video](#)



Property Editor - Surfaceplot

Display Name:

X Data Source: auto

Y Data Source: auto

Z Data Source:

Color: ans

Faces:

Edges: no line

0.5

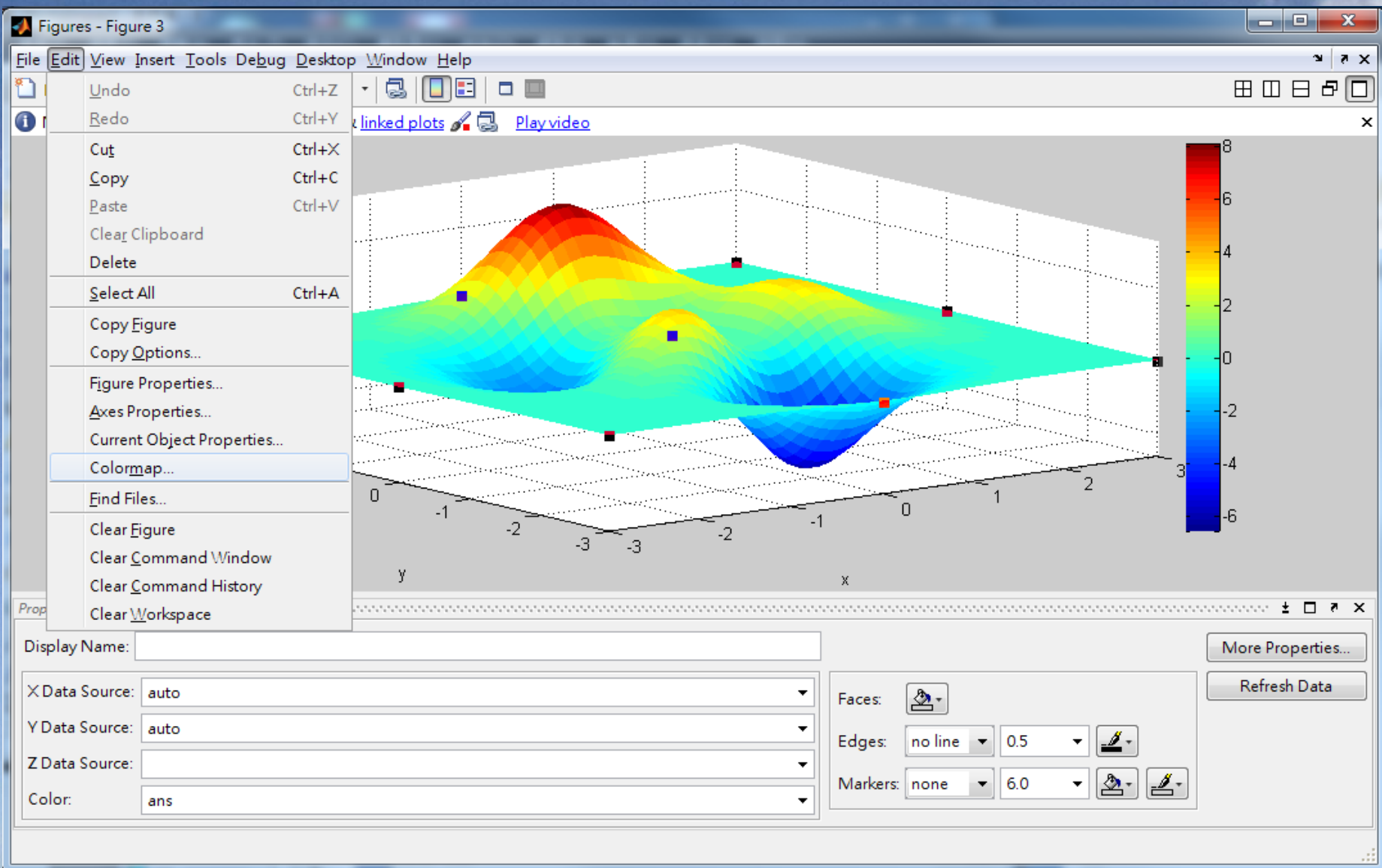
Markers:

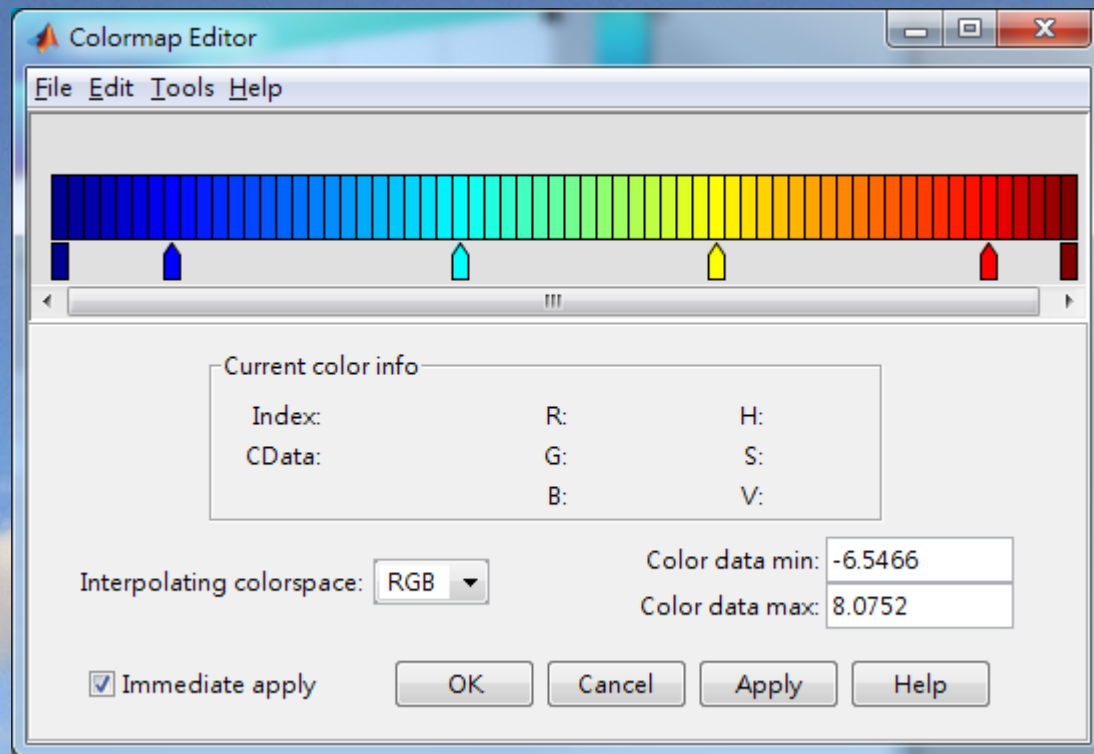
5.0

no line

More Properties...

Refresh Data





練習

試依下列條件繪出 $f(x,y)$ 函數三維函數圖，

$$f(x,y) = d(x^2 + 3y^2) \cdot e^{-x^2-y^2},$$

$$-2.5 \leq x \leq 2.5, -3 \leq y \leq 3$$

(a)資料點數取曲面平滑，(b)在繪圖視窗1繪出 $f(x,y)$ 三維函數圖，並設定圖形的視角，方位角為 36° ，仰角為 18° 。(c)在繪圖視窗2繪出 $f(x,y)$ 的等高線圖，顏色對應表使用6個顏色winter colormap，並在圖右側加入顏色對應表。(d)在所有圖中加入適當的座標軸說明，再將各圖存為圖形格式檔。